![ETSI logo]

EUROPEAN
TELECOMMUNICATION
STANDARD

DRAFT

pr **ETS 300 974**

November 1996

Source: ETSI TC-SMG

Reference: DE/SMG-030902QR

ICS: 33.020

Key words: Digital cellular telecommunications system, Global System for Mobile communications (GSM)

![GSM logo]

GLOBAL SYSTEM FOR
MOBILE COMMUNICATIONS

Digital cellular telecommunications system (Phase 2+);
Mobile Application Part (MAP) specification
(GSM 09.02)

ETSI

European Telecommunications Standards Institute

ETSI Secretariat

Postal address: F-06921 Sophia Antipolis CEDEX - FRANCE
Office address: 650 Route des Lucioles - Sophia Antipolis - Valbonne - FRANCE
X.400: c=fr, a=atlas, p=etsi, s=secretariat - **Internet:** secretariat@etsi.fr

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

Contents

Foreword

This draft European Telecommunication Standard (ETS) has been produced by the Special Mobile Group (SMG) Technical Committee (TC) of the European Telecommunications Standards Institute (ETSI).

This ETS specifies the Mobile Application Part (MAP), the requirements for the signalling system and procedures within the Digital cellular telecommunications system (Phase 2/Phase2+) at application level.

This ETS is a GSM technical specification version 5, which incorporates GSM Phase 2+ enhancements/features to the version 4 GSM technical specification. This ETS is part of the 1996 release of the GSM Technical Specifications. The ETS from which this Phase 2+ ETS has evolved is Phase 2 GSM ETS 300 599 Edition 5 (GSM 09.02 version 4.15.0).

The contents of this ETS are subject to continuing work within TC-SMG and may change following formal TC-SMG approval. Should TC-SMG modify the contents of this ETS, it will be resubmitted for UAP by ETSI with an identifying change of release date and an increase in version number as follows:

Version 5.x.y

where:

      y   the third digit is incremented when editorial only changes have been incorporated in the specification;

      x   the second digit is incremented for all other types of changes, i.e. technical enhancements, corrections, updates, etc.

Reference is made within this TS to GSM-TSs (note).

    NOTE: TC-SMG has produced documents which give the technical specifications for the implementation of the digital cellular telecommunications system. Historically, these documents have been identified as GSM Technical Specifications (GSM-TSs). These TSs may have subsequently become I-ETSs (Phase 1), or ETSs/ETSI Technical Reports (ETRs) (Phase 2). TC-SMG has also produced ETSI GSM TSs which give the technical specifications for the implementation of Phase 2+ enhancements of the digital cellular telecommunications system. These version 5.x.x GSM Technical Specifications may be referred to as GTSs.

Blank page

1 Scope

It is necessary to transfer between entities of a Public Land Mobile Network (PLMN) information specific to the PLMN in order to deal with the specific behaviour of roaming Mobile Stations (MS)s. The Signalling System No. 7 specified by CCITT is used to transfer this information.

This European Telecommunication Standard (ETS) describes the requirements for the signalling system and the procedures needed at the application level in order to fulfil these signalling needs.

Clauses 1 to 4 are related to general aspects such as terminology, mobile network configuration and other protocols required by MAP.

MAP consists of a set of MAP services which are provided to MAP service-users by a MAP service-provider.
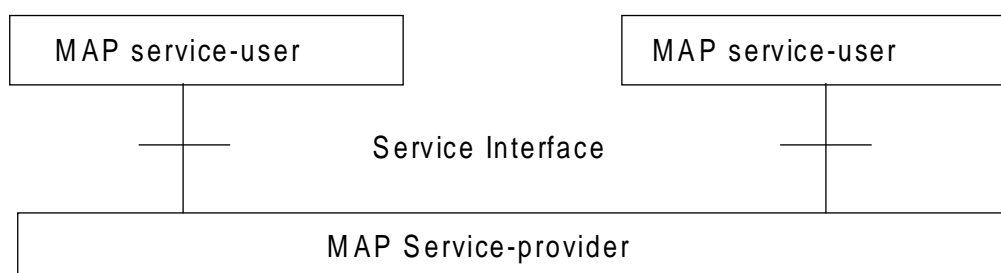
Figure 1.1/1: Modelling principles

Clauses 5 to 10 of this ETS describe the MAP services.

Clauses 11 to 14 define the MAP protocol specification and the behaviour of service provider (protocol elements to be used to provide MAP services, mapping on to TC service primitives, abstract syntaxes...).

Clauses 15 to 21 describe the MAP user procedures which make use of MAP services.

1.1 Normative references

This ETS incorporates by dated and undated reference, provisions from other publications. These normative references are cited at the appropriate places in the text and the publications are listed hereafter. For dated references, subsequent amendments to or revisions of any of these publications apply to this ETS only when incorporated in it by amendment or revision. For undated references, the latest edition of the publication referred to applies.

[1]   GSM 01.04 (ETR 350): "Digital cellular telecommunications system (Phase 2+); Abbreviations and acronyms".

[2]   GSM 02.01: "Digital cellular telecommunications system (Phase 2+); Principles of telecommunication services supported by a GSM Public Land Mobile Network (PLMN)".

[3]   GSM 02.02 (ETS 300 904): "Digital cellular telecommunications system (Phase 2+); Bearer Services (BS) Supported by a GSM Public Land Mobile Network (PLMN)".

[4]   GSM 02.03 (ETS 300 905): "Digital cellular telecommunications system (Phase 2+); Teleservices Supported by a GSM Public Land Mobile Network (PLMN)".

[5]   GSM 02.04 (ETS 300 918): "Digital cellular telecommunications system (Phase 2+); General on supplementary services".

[6]   GSM 02.09 (ETS 300 920): "Digital cellular telecommunications system; Security aspects".

[7]   GSM 02.16: "Digital cellular telecommunications system; International Mobile station Equipment Identities (IMEI)".

[8]   GSM 02.41: "Digital cellular telecommunications system (Phase 2+); Operator determined barring".

[9]   GSM 02.81: "Digital cellular telecommunications system; Line identification supplementary services - Stage 1".

[10]   GSM 02.82: "Digital cellular telecommunications system; Call Forwarding (CF) supplementary services - Stage 1".

[11]   GSM 02.83 : "Digital cellular telecommunications system; Call Waiting (CW) and Call Hold (HOLD) supplementary services - Stage 1".

[12]   GSM 02.84: "Digital cellular telecommunications system; Multi Party (MPTY) supplementary services - Stage 1".

[13]   GSM 02.85: "Digital cellular telecommunications system; Closed User Group (CUG) supplementary services - Stage 1".

[14]   GSM 02.86: "Digital cellular telecommunications system; Advice of charge (AoC) supplementary services - Stage 1".

[15]   GSM 02.88: "Digital cellular telecommunications system; Call Barring (CB) supplementary services - Stage 1".

[16]   GSM 02.90: "Digital cellular telecommunication system (Phase 2+); Unstructured supplementary services operation - Stage 1".

[17]   GSM 03.03 (ETS 300 927): "Digital cellular telecommunications system (Phase 2+); Numbering, addressing and identification".

[18]   GSM 03.04: "Digital cellular telecommunications system; Signalling requirements relating to routeing of calls to mobile subscribers".

[19]   GSM 03.07: "Digital cellular telecommunications system (Phase 2+); Restoration procedures".

[20]   GSM 03.08: "Digital cellular telecommunications system (Phase 2+); Organisation of subscriber data".

[21]   GSM 03.09: "Digital cellular telecommunications system (Phase 2+; Handover procedures".

[22]   GSM 03.11 (ETS 300 928): "Digital cellular telecommunications system; Technical realization of supplementary services".

[23]   GSM 03.12: "Digital cellular telecommunications system; Location registration procedures".

[24]   GSM 03.20 (ETS 300 929): "Digital cellular telecommunications system; Security related network functions".

[25]   GSM 03.38 (ETS 300 900): "Digital cellular telecommunications system (Phase 2+); Alphabets and language specific information for GSM".

[26]   GSM 03.40 (ETS 300 901):   "Digital   cellular   telecommunications   system
             (Phase 2+); Technical realization of the Short Message Service (SMS) Point to
             Point (PP)".

[27]   GSM 03.81: "Digital cellular telecommunications system; Line identification
             supplementary services - Stage 2".

[28]   GSM 03.82: "Digital cellular telecommunications system; Call Forwarding (CF)
             supplementary services - Stage 2".

[29]   GSM 03.83: "Digital cellular telecommunications system; Call Waiting (CW) and
             Call Hold (HOLD) supplementary services - Stage 2".

[30]   GSM 03.84: "Digital cellular telecommunications system; Multi Party (MPTY)
             supplementary services - Stage 2".

[31]   GSM 03.85: "Digital cellular telecommunications system; Closed User Group
             (CUG) supplementary services - Stage 2".

[32]   GSM 03.86 (ETS 300 935): "Digital cellular telecommunications system; Advice
             of Charge (AoC) supplementary services - Stage 2".

[33]   GSM 03.88: "Digital cellular telecommunications system; Call Barring (CB)
             supplementary services - Stage 2".

[34]   GSM 03.90:   "Digital   cellular   telecommunications   system;   Unstructured
             supplementary services operation - Stage 2".

[35]   GSM 04.08 (ETS 300 940):   "Digital   cellular   telecommunications   system
             (Phase 2+); Mobile radio interface layer 3 specification".

[36]   GSM 04.10 (ETS 300 941): "Digital cellular telecommunications system; Mobile
             radio interface layer 3 Supplementary services specification General aspects".

[37]   GSM 04.11 (ETS 300 942):   "Digital   cellular   telecommunications   system
             (Phase 2+); Point-to-Point (PP) Short Message Service (SMS) support on
             mobile radio interface".

[38]   GSM 04.80 (ETS 300 950):   "Digital   cellular   telecommunications   system
             (Phase 2+); Mobile radio interface layer 3 supplementary services specification
             Formats and coding".

[39]   GSM 04.81 (ETS 300 951): "Digital cellular telecommunications system; Line
             identification supplementary services - Stage 3".

[40]   GSM 04.82 (ETS 300 952): "Digital cellular telecommunications system ; Call
             Forwarding (CF) supplementary services - Stage 3".

[41]   GSM 04.83 (ETS 300 953): "Digital cellular telecommunications system; Call
             Waiting (CW) and Call Hold (HOLD) supplementary services - Stage 3".

[42]   GSM 04.84 (ETS 300 954): "Digital cellular telecommunications system; Multi
             Party (MPTY) supplementary services - Stage 3".

[43]   GSM 04.85: "Digital cellular telecommunications system; Closed User Group
             (CUG) supplementary services - Stage 3".

[44]   GSM 04.86 (ETS 300 955): "Digital cellular telecommunications system; Advice
             of Charge (AoC) supplementary services - Stage 3".

[45]   GSM 04.88 (ETS 300 956): "Digital cellular telecommunications system; Call Barring (CB) supplementary services - Stage 3".

[46]   GSM 04.90 (ETS 300 957): "Digital cellular telecommunications system; Unstructured supplementary services operation - Stage 3".

[47]   GSM 08.02: "Digital cellular telecommunications system; Base Station System - Mobile-services Switching Centre (BSS - MSC) interface Interface principles".

[48]   GSM 08.06: "Digital cellular telecommunications system; Signalling transport mechanism specification for the Base Station System - Mobile-services Switching Centre (BSS - MSC) interface".

[49]   GSM 08.08: "Digital cellular telecommunications system (Phase 2+); Mobile Switching Centre - Base Station System (MSC - BSS) interface Layer 3 specification".

[49a]   GSM 08.08: "Digital cellular telecommunications system (Phase 1); Mobile Switching Centre - Base Station System (MSC - BSS) interface Layer 3 specification".

[50]   GSM 09.01 (ETR 359): "Digital cellular telecommunications system; General network interworking scenarios".

[51]   GSM 09.02: "Digital cellular telecommunications system (Phase 1); Mobile Application Part (MAP) specification".

[52]   GSM 09.03: "Digital cellular telecommunications system; Signalling requirements on interworking between the Integrated Services Digital Network (ISDN) or Public Switched Telephone Network (PSTN) and the Public Land Mobile Network (PLMN)".

[53]   GSM 09.04: "Digital cellular telecommunications system; Interworking between the Public Land Mobile Network (PLMN) and the Circuit Switched Public Data Network (CSPDN)".

[54]   GSM 09.05: "Digital cellular telecommunications system; Interworking between the Public Land Mobile Network (PLMN) and the Packet Switched Public Data Network (PSPDN) for Packet Assembly/Disassembly facility (PAD) access".

[55]   GSM 09.06 (ETS 300 975):   "Digital   cellular   telecommunications   system (Phase 2+); Interworking between a Public Land Mobile Network (PLMN) and a Packet Switched Public Data Network/Integrated Services Digital Network (PSPDN/ISDN) for the support of packet switched data transmission services".

[56]   GSM 09.07 (ETS 300 976):   "Digital   cellular   telecommunications   system (Phase 2+); General requirements on interworking between the Public Land Mobile Network (PLMN) and the Integrated Services Digital Network (ISDN) or Public Switched Telephone Network (PSTN)".

[57]   GSM 09.08: "Digital cellular telecommunications system; Application of the Base Station System Application Part (BSSAP) on the E-interface".

[58]   GSM 09.10: "Digital cellular telecommunications system (Phase 2+); Information element mapping between Mobile Station - Base Station System and BSS - Mobile-services Switching Centre (MS - BSS - MSC) Signalling procedures and the Mobile Application Part (MAP)".

[59]   GSM 09.11: "Digital cellular telecommunications system (Phase 2+); Signalling interworking for supplementary services".

[60]   GSM 09.90: "Digital cellular telecommunications system (Phase 2+); Interworking between Phase 1 infrastructure and Phase 2 Mobile Stations (MS)".

[61]   GSM 12.08 (ETS 300 627): "Digital cellular telecommunications system (Phase 2); Subscriber and Equipment Trace".

[62]   ETS 300 102-1 (1990): "Integrated Services Digital Network (ISDN); User-network interface layer 3 specifications for basic call control".

[63]   ETS 300 136 (1992): "Integrated Services Digital Network (ISDN); Closed User Group (CUG) supplementary service description".

[64]   ETS 300 138 (1992): "Integrated Services Digital Network (ISDN); Closed User Group (CUG) supplementary service Digital Subscriber Signalling System No.one (DSS1) protocol".

[65]   ETS 300 287: "Integrated Services Digital Network (ISDN); CCITT Signalling System No.7 Transaction Capabilities Application Part (TCAP)(version 2)".

[66]   ETR 060: "Signalling Protocols and Switching (SPS); Guide-lines for using Abstract Syntax Notation One (ASN.1) in telecommunication application protocols".

[67]   CCITT Recommendation E.164: "Numbering plan for the ISDN era".

[68]   CCITT Recommendation E.212: "Identification plan for land mobile stations".

[69]   CCITT Recommendation E.213: "Telephone and ISDN numbering plan for land mobile stations".

[70]   CCITT Recommendation E.214: "Structuring of the land mobile global title for the signalling connection control part".

[71]   CCITT Recommendation Q.669: "Interworking between the Digital Subscriber Signalling System Layer 3 protocol and the Signalling System No.7 ISDN User part".

[72]   CCITT Recommendation Q.711: "Specifications of Signalling System No.7; Functional description of the signalling connection control part".

[73]   CCITT Recommendation Q.712: "Definition and function of SCCP messages".

[74]   CCITT Recommendation Q.713: "Specifications of Signalling System No.7; SCCP formats and codes".

[75]   CCITT Recommendation Q.714: "Specifications of Signalling System No.7; Signalling connection control part procedures".

[76]   CCITT Recommendation Q.716: "Specifications of Signalling System No.7; Signalling connection control part (SCCP) performances".

[77]   CCITT Recommendation Q.721 (1988): "Specifications of Signalling System No.7; Functional description of the Signalling System No.7 Telephone user part".

[78]   CCITT Recommendation Q.722 (1988): "Specifications of Signalling System No.7; General function of Telephone messages and signals".

[79]   CCITT Recommendation Q.723 (1988): "Specifications of Signalling System No.7; Formats and codes".

[80]   CCITT Recommendation Q.724 (1988): "Specifications of Signalling System No.7; Signalling procedures".

[81]   CCITT Recommendation Q.725 (1988): "Specifications of Signalling System No.7; Signalling performance in the telephone application".

[82]   CCITT Recommendation Q.761 (1988): "Specifications of Signalling System No.7; Functional description of the ISDN user part of Signalling System No.7".

[83]   CCITT Recommendation Q.762 (1988): "Specifications of Signalling System No.7; General function of messages and signals".

[84]   CCITT Recommendation Q.763 (1988): "Specifications of Signalling System No.7; Formats and codes".

[85]   CCITT Recommendation Q.764 (1988): "Specifications of Signalling System No.7; Signalling procedures".

[86]   CCITT Recommendation Q.767: "Specifications of Signalling System No.7; Application of the ISDN user part of CCITT signalling System No.7 for international ISDN interconnections".

[87]   CCITT Recommendation Q.771: "Specifications of Signalling System No.7; Functional description of transaction capabilities".

[88]   CCITT Recommendation Q.772: "Specifications of Signalling System No.7; Transaction capabilities information element definitions".

[89]   CCITT Recommendation Q.773: "Specifications of Signalling System No.7; Transaction capabilities formats and encoding".

[90]   CCITT Recommendation Q.774: "Specifications of Signalling System No.7; Transaction capabilities procedures".

[91]   CCITT Recommendation Q.775: "Specifications of Signalling System No.7; Guide-lines for using transaction capabilities".

[92]   CCITT Recommendation X.200: "Reference Model of Open systems interconnection for CCITT Applications".

[93]   CCITT Recommendation X.208 (1988): "Specification of Abstract Syntax Notation One (ASN.1)".

[94]   CCITT Recommendation X.209 (1988): "Specification of basic encoding rules for Abstract Syntax Notation One (ASN.1)".

[95]   CCITT Recommendation X.210: "Open systems interconnection layer service definition conventions".

1.2   Abbreviations

Abbreviations used in this ETS are listed in GSM 01.04.

2   Configuration of the mobile network

2.1   The entities of the mobile system

To provide the mobile service as it is defined, it is necessary to introduce some specific functions. These functional entities can be implemented in different equipments or integrated. In any case, exchanges of data occur between these entities.

2.1.1   The Home Location Register (HLR)

This functional entity is a data base in charge of the management of mobile subscribers. A PLMN may contain one or several HLRs; it depends on the number of mobile subscribers, on the capacity of the equipment and on the organization of the network. All subscription data are stored there. The main information stored there concerns the location of each MS in order to be able to route calls to the mobile subscribers managed by each HLR. All management interventions occur on this data base. The HLRs have no direct control of MSCs.

Two numbers attached to each mobile subscription are stored in the HLR:

-    IMSI;

-    MSISDN.

The data base contains other information such as:

-    location information (VLR number);

-    basic telecommunication services subscription information;

-    service restrictions (e.g. roaming limitation);

-    supplementary services; the tables contain the parameters attached to these services.

The organization of the subscriber data is detailed in GSM 03.08.

2.1.2   The Visitor Location Register (VLR)

An MS roaming in an MSC area is controlled by the Visitor Location Register in charge of this area. When an MS appears in a location area it starts a location updating procedure. The MSC in charge of that area notices this registration and transfers to the Visitor Location Register the identity of the location area where the MS is situated. A VLR may be in charge of one or several MSC areas.

The VLR also contains the information needed to handle the calls set up or received by the MSs registered in its data base (in some cases the VLR may have to obtain additional information from the HLR); the following elements can be found in its tables:

-    the IMSI;

-    the MSISDN;

-    the TMSI, if applicable;

-    the location area where the MS has been registered. This will be used to call the station;

-    supplementary service parameters.

The information is passed between VLR and HLR by the procedures described in GSM 03.12.

The organization of the subscriber data is detailed in GSM 03.08.

2.1.3   The Mobile-services Switching Centre (MSC)

The Mobile-services Switching Centre is an exchange which performs all the switching functions for MSs located in a geographical area designated as the MSC area. The main difference between an MSC and an exchange in a fixed network is that the MSC has to take into account the impact of the allocation of radio resources and the mobile nature of the subscribers and has to perform, for example, the following procedures:

-       procedures required for the location registration (see GSM 03.12);

-       procedures required for hand-over (see GSM 03.09).

2.1.4   The Base Station System (BSS)

The BSS is the sub-system of Base Station equipment (transceivers, controllers, etc...) which is viewed by the MSC through a single interface (A-interface) with the functionality described in GSM 08.02.

2.1.5   The Gateway MSC (GMSC)

In the case of incoming calls to the PLMN, if the fixed network is unable to interrogate the HLR, the call is routed to an MSC. This MSC will interrogate the appropriate HLR and then route the call to the MSC where the MS is located. The MSC which then performs the routing function to the actual location of the mobile is called the Gateway MSC.

The choice of which MSCs can act as Gateway MSCs is a network operator matter (e.g. all MSCs or some designated MSCs).

See also GSM 03.04.

2.1.6   The SMS Gateway MSC

The SMS GMSC is the interface between the Mobile Network and the network which provides access to the Short Message Service Centre, for short messages to be delivered to MSs.

The choice of which MSCs can act as SMS Gateway MSCs is a network operator matter (e.g. all MSCs or some designated MSCs).

2.1.7   The SMS Interworking MSC

The SMS IWMSC is the interface between the Mobile Network and the network which provides access to the Short Message Service Centre, for short messages submitted by MSs.

The choice of which MSCs can act as SMS Interworking MSCs is a network operator matter (e.g. all MSCs or some designated MSCs).

2.1.8   The Equipment Identity Register (EIR)

This functional unit is a data base in charge of the management of the equipment identities of the MSs; see also GSM 02.16.

2.2   Configuration of a Public Land Mobile Network (PLMN)

The basic configuration of a Public Land Mobile Network is presented in figure 2.2/1. In this figure the most general solution is described in order to define all the possible interfaces which can be found in any PLMN. The specific implementation in each network may be different: some particular functions may be implemented in the same equipment and then some interfaces may become internal interfaces. In any case the configuration of a PLMN must have no impact on the relationship with the other PLMNs.

In this configuration, all the functions are considered implemented in different equipments. Therefore, all the interfaces are external and need the support of the Mobile Application Part of the Signalling System

No. 7 to exchange the data necessary to support the mobile service. From this configuration, all the possible PLMN organizations can be deduced.

2.3 Interconnection between PLMNs

Since the configuration of a PLMN does not have any impact on other PLMNs, the signalling interfaces specified can be implemented both between the entities within a PLMN and between different PLMNs.

2.4 The interfaces within the mobile service

2.4.1 Interface between the HLR and the VLR (D-interface)

This interface is used to exchange the data related to the location of the MS and to the management of the subscriber. The main service provided to the mobile subscriber is the capability to set up or to receive calls within the whole service area. To support that purpose the location registers have to exchange data. The VLR informs the HLR on the registration of a MS managed by the latter and provides it with the relevant location information. The HLR sends to the VLR all the data needed to support the service to the MS. The HLR then calls the previous VLR to inform it that it can cancel the location registration of this station because of the roaming of the mobile.

Exchanges of data may also occur when the mobile subscriber requires a particular service, when he wants to change some data attached to his subscription or when some parameters of the subscription are modified by administrative means.

2.4.2 [Spare]

2.4.3 Interface between the VLR and its associated MSC(s) (B-interface)

The VLR is the location and management data base for the MSs roaming in the area controlled by the associated MSC(s). Whenever the MSC needs data related to a given MS currently located in its area, it interrogates the VLR. When a MS initiates a location updating procedure with an MSC, the MSC informs its VLR which stores the relevant information in its tables. This procedure occurs whenever a mobile roams to another location area. Also, for instance when a subscriber activates a specific supplementary service or modifies some data attached to a service, the MSC transfers (via the VLR) the request to the HLR, which stores these modifications and updates the VLR if required.

However, this interface is not fully operational specified. It is strongly recommended not to implement the B-interface as an external interface.

2.4.4 Interface between VLRs (G-interface)

When an MS initiates a location updating using TMSI, the VLR can fetch the IMSI and authentication set from the previous VLR.

2.4.5 Interface between the HLR and the MSC (C-interface)

When the fixed network is not able to perform the interrogation procedure needed to set up a call to a mobile subscriber, the Gateway MSC has to interrogate the HLR of the called subscriber to obtain the roaming number of the called MS (see GSM 03.04).

To forward a short message to a mobile subscriber, the SMS Gateway MSC has to interrogate the HLR to obtain the MSC number where the MS is located.

2.4.6 Interface between MSCs (E-interface)

When a MS moves from one MSC area to another during a call, a handover procedure has to be performed in order to continue the communication. For that purpose the MSCs involved have to exchange data to initiate and then to realize the operation.

This interface is also used to forward short messages.

2.4.7 Interface between the MSC and Base Station Systems (A-interface)

The description of this interface is contained in the GSM 08-series of MSs.

The BSS-MSC interface carries information concerning:

- BSS management;

- call handling;

- location management.

2.4.8 Interface between MSC and EIR (F-interface)

This interface is used when an MSC wants to check an IMEI.

2.5 Splitting of the data storage

The data attached to each MS management, operation and location are stored in the Location Registers. Some data are duplicated in the HLR and in the VLR, but others may be stored only in one place.

A detailed description of the data organization can be found in GSM 03.08.
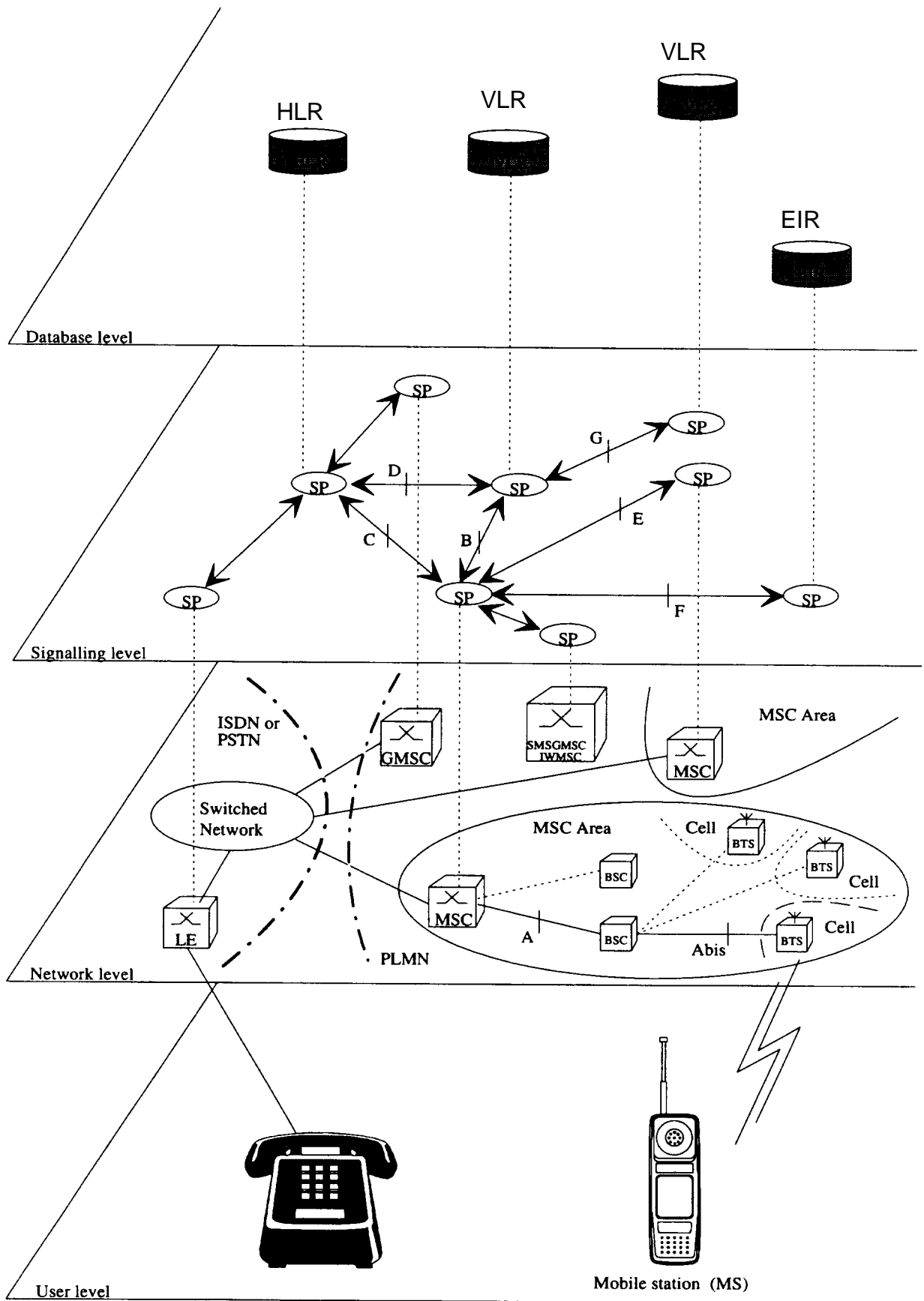
Figure 2.2/1: Configuration of a PLMN

3 Overload and compatibility overview

3.1 Overload control

There is a requirement for an overload/congestion control for all entities of the Public Land Mobile Network and the underlying Signalling System No. 7.

3.1.1 Overload control for MSC (outside MAP)

For the entity MSC the following two procedures (outside MAP) may be applied to control the processor load:

-       ISDN
        CCITT Recommendation Q.764 (Automatic Congestion Control), applicable to reduce the mobile terminating traffic;

-       BSSAP
        GSM 08.08 (A-interface Flow Control), applicable to reduce the mobile originating traffic.

3.1.2 Overload control for MAP entities

For all MAP entities, especially the HLR, the following overload control method is applied:

If overload of a MAP entity is detected requests for certain MAP operations (see tables 3.1/1 and 3.1/2) may be ignored by the responder. The decision as to which MAP Operations may be ignored is made by the MAP service provider and is based upon the priority of the application context.

Since most of the affected MAP operations are supervised in the originating entity by TC timers (medium) an additional delay effect is achieved for the incoming traffic.

If overload levels are applicable in the Location Registers the MAP operations should be discarded taking into account the priority of their application context (see table 3.1/1 for HLR and table 3.1/2 for MSC/VLR; the lowest priority is discarded first).

The ranking of priorities given in the tables 3.1/1 and 3.1/2 is not normative. The tables can only be seen as a proposal which might be changed due to network operator/implementation matters.

	Responder = HLR	Initiating Entity
Priority high		
	Mobility Management	
	networkLocUp	VLR
	(updateLocation),	
	(restoreData/v2),	
	(sendParameters/v1)	
	infoRetrieval	VLR
	(sendAuthenticationInfo/v2),	
	(sendParameters/v1)	
	msPurging	VLR
	(purgeMS/v2)	
	Short Message Service	
	shortMsgGateway	GMSC
	(sendRoutingInfoforSM),	
	(reportSM-DeliveryStatus)	
	mwdMngt	VLR
	(readyForSM/v2),	
	(noteSubscriberPresent/v1)	
	Mobile Terminating Traffic	
	locInfoRetrieval	GMSC
	(sendRoutingInfo)	
	Subscriber Controlled Inputs (Supplementary Services)	
	networkFunctionalSs	VLR
	(registerSS),	
	(eraseSS),	
	(activateSS),	
	(deactivateSS),	
	(interrogateSS),	
	(registerPassword),	
	(processUnstructuredSS-Data/v1),	
	(beginSubscriberActivity/v1)	
	networkUnstructuredSs	VLR
	(processUnstructuredSS-Request/v2)	
	imsiRetrieval	VLR
	(sendIMSI/v2)	
Priority low		

NOTE: The application context name is the last component but one of the object identifier. Operation names are given in brackets for information with "/vn" appended to vn only operations.

Table 3.1/1: Priorities of Application Contexts for HLR as Responder

Responder = MSC/VLR	Initiating Entity
Priority high	
Handover	
handoverControl	MSC
(prepareHandover/v2),	
(performHandover/v1)	
Mobility and Location Register Management	
locationCancel	HLR
(cancelLocation)	
reset	HLR
(reset)	
interVlrInfoRetrieval	VLR
(sendIdentification/v2),	
(sendParameters/v1)	
subscriberDataMngt	HLR
(insertSubscriberData),	
(deleteSubscriberData)	
tracing	HLR
(activateTraceMode),	
(deactivateTraceMode)	
Short Message Service	
shortMsgMO-Relay	MSC
(forwardSM)	
shortMsgMT-Relay	MSC
(forwardSM)	
shortMsgAlert	HLR
(alertServiceCentre/v2),	
(alertServiceCentreWithoutResult/v1)	
Mobile Terminating Traffic	
roamingNbEnquiry	HLR
(provideRoamingNumber)	
Network-Initiated USSD	
networkUnstructuredSs	HLR
(unstructuredSS-Request/v2),	
(unstructuredSS-Notify/v2)	
Priority low	

NOTE: The application context name is the last component but one of the object identifier.
Operation names are given in brackets for information with "/vn" appended to vn only
operations.

Table 3.1/2: Priorities of Application Contexts for MSC/VLR as Responder

3.1.3 Congestion control for Signalling System No. 7

The requirements of SS7 Congestion control have to be taken into account as far as possible.

Means which could be applied to achieve the required traffic reductions are described in subclauses 3.1.1 and 3.1.2.

3.2 Compatibility

3.2.1 General

This ETS of the Mobile Application Part is designed in such a way that an implementation which conforms to it can also conform to the Mobile Application Part operational version 1 specifications, except on the MSC-VLR interface.

A version negotiation mechanism based on the use of an application-context-name is used to negotiate the protocol version used between two entities for supporting a MAP-user signalling procedure.

When starting a signalling procedure, the MAP-user supplies an application-context-name to the MAP-provider. This name refers to the set of application layer communication capabilities required for this dialogue. This refers to the required TC facilities (i.e. version 1 or 2) and the list of operation packages (i.e. set of operations) from which operations can be invoked during the dialogue.

A version one application-context-name may only be transferred to the peer user in a MAP-U-ABORT to an entity of version two or higher (i.e. to trigger a dialogue which involves only communication capabilities defined for MAP operational version 1).

If the proposed application-context-name can be supported by the responding entity the dialogue continues on this basis otherwise the dialogue is refused an the initiating user needs to start a new dialogue which involve another application-context-name which require less communication capabilities but provides similar functionalities (if possible).

When a signalling procedure can be supported by several application contexts which differs by their version number, the MAP-User need to select a name. It can either select the name which corresponds to the highest version it supports or follow a more specific strategy so that the number of protocol fallbacks due to version compatibility problems be minimized.

3.2.2 Strategy for selecting the Application Context (AC) version

A method should be used to minimize the number of protocol fall-backs which would occur sometimes if the highest supported AC-Name were always the one selected by GSM entities when initiating a dialogue. The following method is an example which can be used mainly at transitory phase stage when the network is one of mixed phase entities.

3.2.2.1 Proposed method

A table (table 1) may be set up by administrative action to define the highest application context (AC) version supported by each destination; a destination may be another node within the same or a different PLMN, or another PLMN considered as a single entity. The destination may be defined by an E.164 number or an E.214 number derived from an IMSI. The table also includes the date when each destination is expected to be able to handle at least one AC of the MAP version 2 protocol. When this date is reached, the application context supported by the node is marked as "unknown", which will trigger the use of table 2.

A second table (table 2) contains an entry for each destination which has an entry in table 1. For a given entity, the entry in table 2 may be a single application context version or a vector of different versions applying to different application contexts for that entity. Table 2 is managed as described in subclause 3.2.2.2.

The data for each destination will go through the following states:

a)   the version shown in table 1 is "version 1"; table 2 is not used;

b)   the version shown in table 1 is "unknown"; table 2 is used, and maintained as described in subclause 3.2.2.2;

c)   when the PLMN operator declares that an entity (single node or entire PLMN) has been upgraded to support all the MAP version 2 ACs defined for the relevant interface, the version shown in table 1 is set to "version 2" by administrative action; table 2 is no longer used, and the storage space may be recovered.

3.2.2.2   Managing the version look-up table

WHEN it receives a MAP-OPEN ind the MAP-User determines the originating entity number either using the originating address parameter or the originating reference parameter or retrieving it from the subscriber data using the IMSI or the MSISDN.

IF the entity number is known

> **THEN**
>> It updates (if required) the associated list of highest supported ACs
>
> **ELSE**
>> It creates an entry for this entity and includes the received AC-name in the list of highest supported ACs.

WHEN starting a procedure, the originating MAP-user looks up its version control table.

IF the destination address is known and not timed-out

> **THEN**
> It retrieves the appropriate AC-name and uses it
>
>> **IF** the dialogue is accepted by the peer
>>
>> **THEN**
>>> It does not modify the version control table
>>
>> **ELSE** (this should never occur)
>>> It starts a new dialogue with the common highest version supported (based on information implicitly or explicitly provided by the peer).
>>>
>>> It replace the old AC-name by the new one in the list of associated highest AC supported.
>
> **ELSE**
> It uses the AC-name which corresponds to the highest version it supports.
>
>> **IF** the dialogue is accepted by the peer
>>
>> **THEN**
>>> It adds the destination node in its version control table and includes the AC-Name in the list of associated highest AC supported.

ELSE

It starts a new dialogue with the common highest version supported (based on information implicitly or explicitly provided by the peer).

IF the destination node was not known

THEN

It adds the destination node in its version control table and includes the new AC-Name in the list of associated highest AC supported.

ELSE

It replaces the old AC-name by the new one in the list of highest supported AC and reset the timer.

3.2.2.3   Optimizing the method

A table look-up may be avoided in some cases if both the HLR and the VLR stores for each subscriber the version of the AC-name used at location updating. Then:

- for procedures which make use of the same application-context, the same AC-name (thus the same version) can be selected (without any table look-up) when the procedure is triggered;

- for procedures which make use of a different application-context but which includes one of the packages used by the location updating AC, the same version can be selected (without any table look-up) when the procedure is triggered;

for HLR:

- Subscriber data modification (stand alone);

for VLR:

- Data Restoration.

4 Requirements concerning the use of SCCP and TC

4.1 Use of SCCP

The Mobile Application Part makes use of the services offered by the Signalling Connection Control Part of signalling System No. 7. CCITT Blue Book or ITU-T (03/93) Recommendations Q.711 to Q.716 should be consulted for the full specification of SCCP.

4.1.1 SCCP Class

MAP will only make use of the connectionless classes (0 or 1) of the SCCP.

4.1.2 Sub-System Number (SSN)

The Application Entities (AEs) defined for MAP consist of several Application Service Elements (ASEs) and are addressed by sub-system numbers (SSNs). The SSN for MAP are:

    00000101   for the whole MAP (reserved for possible future use);

    00000110   HLR;

    00000111   VLR;

    00001000   MSC;

    00001001   EIR;

    00001010   is allocated for evolution (possible Authentication centre).

4.1.3 SCCP addressing

4.1.3.1 Introduction

Within the GSM System there will be a need to communicate between entities within the same PLMN and in different PLMNs. Using the Mobile Application Part (MAP) for this function implies the use of Transaction Capabilities (TC) and the Signalling Connection Control Part (SCCP) of CCITT Signalling System No. 7.

Only the entities which should be addressed are described below. The format and coding of address parameters carried by the SCCP for that purpose shall comply with CCITT Recommendation Q.713 with the following restrictions:

1)   Intra-PLMN addressing
      a)   Called Party Address
      -     SSN indicator = 1 (MAP SSNs always included);

      -     a Global title or a point code may also be included;

      b)   Calling Party Address
      -     SSN indicator = 1 (MAP SSNs always included);
      -     a Global title or a point code may also be included; If none of these parameters is present
                the MTP address (i.e. OPC) is used;

2)   Inter-PLMN addressing
   a)   Called Party Address
-    SSN indicator = 1 (MAP SSN always included);

-    Global title indicator = 0100 (Global title includes translation type, numbering plan, encoding scheme and nature of address indicator);

-    the translation type field will be coded "00000000" (Not used);

-    Routing indicator = 0 (Routing on global title);

   b)   Calling Party Address
-    SSN indicator = 1 (MAP SSNs always included);
-    Point code indicator = 0;

-    Global title indicator = 0100 (Global title includes translation type, numbering plan, encoding scheme and nature of address indicator);

-    the translation type field will be coded "00000000" (Not used);

-    Routing indicator = 0 (Routing on Global Title).

When an SCCP UNITDATA message is sent, the Called Party Address is derived from the Called Address of the N-UNITDATA-REQUEST primitive issued by the local TC. The Calling Party Address received in the dialogue initiating SCCP UNITDATA message is forwarded to the local TC of the dialogue responding entity as Calling Address within the N-UNITDATA-INDICATION primitive in order to be stored as Called Address for the current transaction.

In the case of intra-PLMN addressing, it may be necessary to complement the Calling Party Address received in the dialogue initiating SCCP UNITDATA message by the associated Originating Point Code taken from the MTP-TRANSFER-INDICATION primitive as indicated above. In this case, the Signalling Point Code included in the Calling Address forwarded to the local TC on receipt of the dialogue initiating message is used as Destination Point Code for MTP routeing of all messages sent by the dialogue responding entity within one dialogue.

For supporting the GSM protocol for inter-PLMN addressing, SCCP shall not modify the SCCP address information received in an SCCP UNITDATA message before forwarding it to the local TC. In particular, the Calling Party Address received within the dialogue initiating message shall not be complemented by a Signalling Point Code before forwarding it to TC as Calling Address in the N-UNITDATA-INDICATION primitive. Any modification of the SCCP address information received in an SCCP UNITDATA message in the case of inter-PLMN addressing constitutes a PLMN specific option.

For supporting the GSM protocol, the options provided by ITU-T (03/93) SCCP for processing of the Called Address received from the local TC in the N-UNITDATA-REQUEST primitive shall be restricted as follows: If the Called Address received from TC in the N-UNITDATA-REQUEST primitive contains a Global Title, a Destination Point Code shall be established by means of Global Title translation. In the case of inter-PLMN signalling, the Global Title translation shall result in "routing on Global Title". PLMN specific requirements may cause application of a different option provided by ITU-T (03/93) SCCP.

If a Global Title translation is required for obtaining routeing information, one of the numbering plans E.164, E.212 and E.214 is applicable.

- E.212 numbering plan
  An E.212 number must not be included as Global Title in an SCCP UNITDATA message. The translation of an E.212 number into a Mobile Global Title is applicable in a dialogue initiating VLR if the routeing information towards the HLR is derived from the subscriber's IMSI. When an MS moves from one VLR service area to another, the new VLR may derive the address of the previous VLR from the Location Area Identification provided by the MS in the location registration request. The PLMN where the previous VLR is located is identified by the E.212 numbering plan elements of the Location Area Identification, ie the Mobile Country Code (MCC) and the Mobile Network Code (MNC).

- E.214 and E.164 numbering plans
  Only address information belonging to either E.214 or E.164 numbering plan is allowed to be included as Global Title in the Called and Calling Party Address.
  If the Calling Party Address associated with the dialogue initiating message contains a Global Title, the sending network entity shall include its E.164 entity number.
  When receiving an SCCP UNITDATA message, SCCP shall accept either of the valid numbering plans in the Called Party Address and in the Calling Party Address.
  When receiving an N-UNITDATA-REQUEST primitive from TC, SCCP shall accept an E.164 number or an E.214 number in the Called Address and in the Calling Address.

The following subclauses describe the method of SCCP addressing appropriate for each entity both for the simple intra-PLMN case and where an inter-PLMN communication is required. The following entities are considered:

The Mobile-services Switching Centre (MSC);

The Home location Register (HLR);

The Visitor Location Register (VLR);

The Gateway Mobile-services Switching Centre (GMSC);

The Interworking Mobile-services Switching Centre (IWMSC).

4.1.3.2 The Mobile-services Switching Centre (MSC)

There are several cases where it is necessary to address the MSC.

4.1.3.2.1 MSC interaction during handover

The address is derived from the target Cellid.

4.1.3.2.2 MSC for short message routing

When a short message has to be routed to a MS, the GMSC addresses the VMSC by an MSC identity received from the HLR which complies with E.164 rules.

For MS originating short message, the IWMSC address is derived from the Service Centre address.

4.1.3.3 The Home Location Register (HLR)

There are several cases where the HLR has to be addressed:

4.1.3.3.1 During call set-up

When a call is initiated the HLR of the called mobile subscriber will be interrogated to discover the whereabouts of the MS. The addressing required by the SCCP will be derived from the MSISDN dialled by the calling subscriber. The dialled number will be translated into either an SPC, in the case of communications within a PLMN, or a Global Title if other networks are involved (i.e. if the communication is across a PLMN boundary).

If the calling subscriber is a fixed network subscriber, the interrogation can be initiated from the Gateway MSC of the home PLMN in the general case. If the topology of the network allows it, the interrogation could be initiated from any Signalling Point which has MAP capabilities, e.g. local exchange, outgoing International Switching Centre (ISC), etc.

4.1.3.3.2 Before location updating completion

When a MS registers for the first time in a VLR, the VLR has to initiate the update location dialogue with the MS's HLR and a preceding dialogue for authentication information retrieval if the authentication information must be retrieved from the HLR. When initiating either of these dialogues, the only data for addressing the HLR that the VLR has available is contained in the IMSI, and addressing information for SCCP must be derived from it. When continuing the established update location dialogue (as with any other dialogue), the VLR must derive the routeing information towards the HLR from the Calling Party Address received with the first responding CONTINUE message until the dialogue terminating message is received. This means that the VLR must be able to address the HLR based:

- on an E.214 Mobile Global Title originally derived by the VLR from the IMSI; or
- on an E.164 HLR address; or
- in the case of intra-PLMN signalling, on an SPC.

If the HLR is in the same PLMN as the VLR, local translation tables may exist to derive an SPC. For authentication information retrieval and location updating via the international PSTN/ISDN signalling network, the Global title must be derived from the IMSI, using the principles contained in CCITT Recommendation E.214 and the Numbering Plan Indicator (NPI) value referenced by the SCCP Specifications. A summary of the translation from the IMSI (CCITT Recommendation E.212) to Mobile Global Title (described in CCITT Recommendation E.214) is shown below:

- E.212 Mobile Country Code translates to E.164 Country Code;

- E.212 Mobile Network Code translates to E.164 National Destination Code;

- E.212 Mobile Subscriber Identification Number (MSIN) is carried unchanged if within the E.164 number maximum length and terminated by the ST signal (15 digits + ST). If the Mobile Global Title is more than 15 digits the number is truncated to 15 by deleting the least significant digits.

This translation will be done either at the application or at SCCP level in the VLR. The Mobile Global Title thus derived will be used to address the HLR.

If location updating is triggered by an MS that roams from one MSC Area into a different MSC Area served by the same VLR, the VLR shall address the HLR in the same way as if the MS registers for the first time in the VLR.

4.1.3.3.3 After location updating completion

In this case, the subscriber's basic MSISDN has been received from the HLR during the subscriber data retrieval procedure as well as the HLR number constituting a parameter of the MAP message indicating successful completion of the update location dialogue. From either of these E.164 numbers the address information for initiating dialogues with the roaming subscriber's HLR can be derived. Also the subscriber's IMSI may be used for establishing the routeing information towards the HLR. This may apply in particular if the dialogue with the HLR is triggered by subscriber controlled input.

Thus the SCCP address of the roaming subscriber's HLR may be an SPC, or it may be a Global title consisting of the E.164 MSISDN or the E.164 number allocated to the HLR or the E.214 Mobile Global Title derived from the IMSI.

4.1.3.3.4 VLR restoration

If a roaming number is requested by the HLR for an IMSI that has no data record in the interrogated VLR, the VLR provides the roaming number in the dialogue terminating message. Subsequently the VLR must retrieve the authentication data from the MS's HLR, if required, and must then trigger the restore data procedure. For this purpose, the VLR has to initiate in succession two independent dialogues with the MS's HLR. The MTP and SCCP address information needed for routeing towards the HLR can be derived from the IMSI received as a parameter of the MAP message requesting the roaming number. In this case, the IMSI received from the HLR in the roaming number request shall be processed in the same way as the IMSI that is received from an MS that registers for the first time within a VLR. Alternatively to the IMSI, the Calling Party Address associated with the roaming number request may be used to obtain the routeing information towards the HLR.

4.1.3.4 The Visitor Location Register (VLR)

There are several cases when the VLR needs to be addressed:

4.1.3.4.1 Inter-VLR information retrieval

When an MS moves from one VLR service area to another, the new VLR may request the IMSI and authentication sets from the previous VLR. The new VLR derives the address of the previous VLR from the Location Area Identification provided by the MS in the location registration request.

4.1.3.4.2 HLR request

The HLR will only request information from a VLR if it is aware that one of its subscribers is in the VLR's service area. This means that a location updating dialogue initiated by the VLR has been successfully completed, i.e. the HLR has indicated successful completion of the update location procedure to the VLR.

When initiating dialogues towards the VLR after successful completion of location updating, the routeing information used by the HLR is derived from the E.164 VLR number received as a parameter of the MAP message initiating the update location dialogue. If the VLR is in the same PLMN as the HLR, the VLR may be addressed directly by an SPC derived from the E.164 VLR number. For dialogues via the international PSTN/ISDN signalling network, presence of the E.164 VLR number in the Called Party Address is required.

4.1.3.5 The Interworking MSC (IWMSC) for Short Message Service

The IWMSC is the interface between the mobile network and the network to access to the Short Message Service Centre. This exchange has an E.164 address known in the HLR or in the MSC.

4.1.3.6 The Equipment Identity Register (EIR)

The EIR address is either unique or could be derived from the IMEI. The type of address is not defined.

4.1.3.7   Summary table

The following table summarizes the SCCP address used for invoke operations. As a principle, within a PLMN either an SPC or a GT may be used (network operation option), whereas when addressing an entity outside the PLMN the GT must be used. The address type mentioned in the table (e.g. MSISDN) is used as GT or to derive the SPC.

For a response, the originating address passed in the invoke is used. For extra-PLMN addressing the entity number is used as GT; for intra-PLMN addressing an SPC derived from the entity number may be used instead. When using an SPC, the SPC may be taken directly from MTP.

to from	fixed network	HLR	VLR	MSC	EIR
fixed network	---	E:GT T:MSISDN	---	---	---
home location register	---	---	I:SPC/GT E:GT T:VLR NUMBER	---	---
visitor location register	---	I:SPC/GT E:GT T:MGT/MSISDN/ HLR NUMBER (note)	I:SPC/GT E:GT T:VLR NUMBER	---	---
mobile-services switching centre	---	I:SPC/GT E:GT T:MSISDN	I:SPC/GT E:GT T:VLR NUMBER	I:SPC/GT E:GT T:MSC NUMBER	I:SPC/GT E:GT T:EIR NUMBER

I: Intra-PLMN   E: Extra(Inter)-PLMN   T: Address Type
GT: Global Title   MGT: E.214 Mobile Global Title SPC: Signalling Point Code

NOTE: For initiating the location updating procedure and an authentication information retrieval from the HLR preceding it, the VLR has to derive the HLR address from the IMSI of the MS. The result can be an SPC or an E.214 Mobile Global Title. When continuing the established update location dialogue (as with any other dialogue) the VLR must derive the routeing information towards the HLR from the Calling Party Address received with the first responding CONTINUE message until the dialogue terminating message is received.

For transactions invoked by the VLR after update location completion, the VLR may derive the information for addressing the HLR from addresses received in the course of the update location procedure (MSISDN or HLR number) or from the IMSI.

When invoking the Restore Data procedure and an authentication information retrieval from the HLR preceding it, the VLR must derive the information for addressing the HLR from the address information received in association with the roaming number request. This may be either the IMSI received as a parameter of the MAP message requesting the Roaming Number or the Calling Party Address associated with the MAP message requesting the Roaming Number.

Table 4.1/1

4.2   Use of TC

The Mobile Application part makes use of the services offered by the Transaction Capabilities (TC) of signalling system No. 7. ETS 300 287, which is based on CCITT White Book Recommendations Q.771 to Q.775, should be consulted for the full specification of TC.

The MAP uses all the services provided by TC except the ones related to the unstructured dialogue facility.

From a modelling perspective, the MAP is viewed as a single Application Service Element. Further structuring of it is for further study.

Transaction Capabilities refers to a protocol structure above the network layer interface (i.e, the SCCP service interface) up to the application layer including common application service elements but not the specific application service elements using them.

TC is structured as a Component sub-layer above a Transaction sub-layer.

The Component sub-layer provides two types of application services: services for the control of end-to-end dialogues and services for Remote Operation handling. These services are accessed using the TC-Dialogue handling primitives and TC-Component handling primitives respectively.

Services for dialogue control include the ability to exchange information related to application-context negotiation as well as initialization data.

Services for Remote Operation handling provide for the exchange of protocol data units invoking tasks (operations), and reporting their outcomes (results or errors) plus any non-application-specific protocol errors detected by the component sub-layer. The reporting of application-specific protocol errors by the TC user, as distinct from application process errors, is also provided. The Transaction sub-layer provides a simple end-to-end connection association service over which several related protocol data units (i.e. built by the Component Sub-Layer) can be exchanged. A Transaction termination can be prearranged (no indication provided to the TC user) or basic (indication provided).
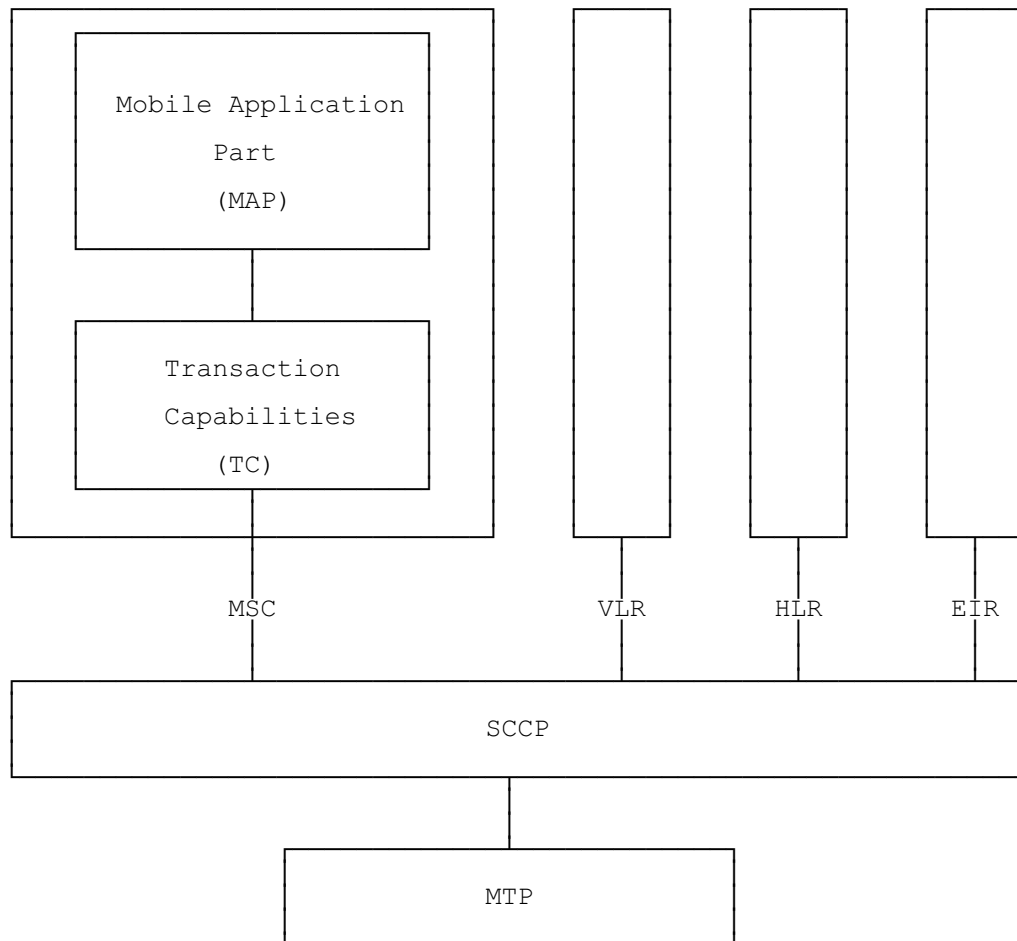
Figure 4.2/1: Facilities for supporting the Mobile Application Part in Signalling System No.7

5   General on MAP services

5.1   Terminology and definitions

The term service is used in clauses 5 to 10 as defined in CCITT Recommendation X.200. The service definition conventions of CCITT Recommendation X.210 are also used.

5.2   Modelling principles

MAP provides its users with a specified set of services and can be viewed by its users as a "black box" or abstract machine representing the MAP service-provider. The service interface can then be depicted as shown in figure 5.2/1.
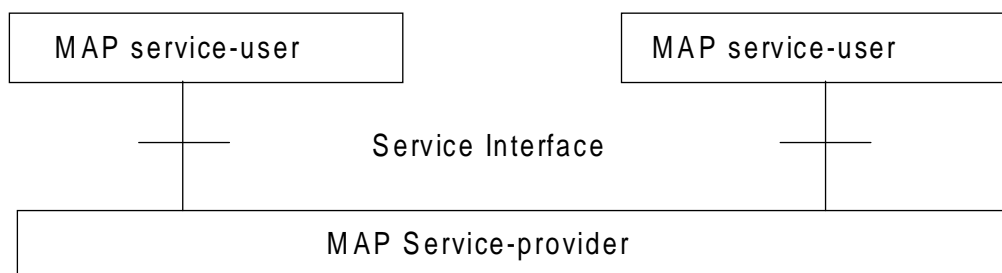
Figure 5.2/1: Modelling principles

The MAP service-users interact with the MAP service-provider by issuing or receiving MAP service-primitives at the service interface.

A MAP service-user may receive services from several instances of the MAP service-provider at the same time. In such cases the overall procedure is synchronised by the service-user.

The MAP service-primitives are named using the following notation:

MAP-ServicePrimitiveName type

where type can be any of: request (req), indication (ind), response (rsp) or confirm (cnf) (In the user arrow diagrams type is not indicated in the case of req/ind and indicated as "ack" in the case of rsp/cnf).

The services are further classified as unconfirmed-service, confirmed-service and provider-initiated-service where the first two categories refer to whether or not the service is confirmed by the service-provider. The confirmation may or may not correspond to a response provided by the other service-user.

MAP services are also classified as common MAP services which are available to all MAP service-users, and MAP service-user specific services which are services available to one or several, but not all, MAP service-users.

A MAP dialogue is defined as an exchange of information between two MAP users in order to perform a common task. A MAP dialogue will consist of one or several MAP services.

5.3 Common MAP services

All MAP service-users require access to services for performing basic application layer functions:

- for establishing and clearing MAP dialogues between peer MAP service-users;
- for accessing functions supported by layers below the applications layer;
- for reporting abnormal situations;
- for handling of different MAP versions;
- for testing whether or not a persistent MAP dialogue is still active at each side.

For these purposes the following common services are defined:

- MAP-OPEN service;
- MAP-CLOSE service;
- MAP-DELIMITER service;
- MAP-U-ABORT service;
- MAP-P-ABORT service;
- MAP-NOTICE service.

In defining the service-primitives the following convention is used for categorising parameters:

M   the inclusion of the parameter is mandatory. The M category can be used for any primitive type and specifies that the corresponding parameter must be present in the indicated primitive type;

O   the inclusion of the parameter is a service-provider option. The O category can be used in indication and confirm type primitives and is used for parameters that may optionally be included by the service-provider;

U   the inclusion of the parameter is a service-user option. The U category can be used in request and response type primitives. The inclusion of the corresponding parameter is the choice of the service-user;

C   the inclusion of the parameter is conditional. The C category can be used for the following purposes:
- to indicate that if the parameter is received from another entity it must be included for the service being considered;

- to indicate that the service user must decide whether to include the parameter, based on the context on which the service is used;

- to indicate that one of a number of mutually exclusive parameters must be included (e.g. parameters indicating a positive result versus parameters indicating a negative result);

- to indicate that a service user optional parameter (marked "U") or a conditional parameter (marked "C") presented by the service user in a request or response type primitive is to be presented to the service user in the corresponding indication or confirm type primitive;

(=)   when appended to one of the above, this symbol means that the parameter takes the same value as the parameter appearing immediately to its left;

blank the parameter is not present.

A primitive type may also be without parameters, i.e. no parameter is required with the primitive type; in this case the corresponding column of the table is empty.

5.3.1   MAP-OPEN service

This service is used for establishing a MAP dialogue between two MAP service-users. The service is a confirmed service with service primitives as shown in table 5.3/1.

Parameters	Request	Indication	Response	Confirm
Application context name	M	M(=)	U	C(=)
Destination address	M	M(=)		
Destination reference	U	C(=)		
Originating address	U	O		
Originating reference	U	C(=)		
Specific information	U	C(=)	U	C(=)
Responding address			U	C(=)
Result			M	M(=)
Refuse-reason			C	C(=)
Provider error				O

Table 5.3/1: Service-primitives for the MAP-OPEN service

Application context name:

This parameter identifies the type of application context being established. If the dialogue is accepted the received application context name shall be echoed. In case of refusal of dialogue this parameter shall indicate the highest version supported.

Destination address:

A valid SCCP address identifying the destination peer entity (see also clause 4). As an implementation option, this parameter may also, in the indication, be implicitly associated with the service access point at which the primitive is issued.

Destination-reference:

This parameter is a reference which refines the identification of the called process. It may be identical to Destination address but its value is to be carried at MAP level. Table 5.3/2 describes the MAP services using this parameter. Only these services are allowed to use it.

MAP service	Reference type	Use of the parameter
MAP-REGISTER-SS	IMSI	Subscriber identity
MAP-ERASE-SS	IMSI	Subscriber identity
MAP-ACTIVATE-SS	IMSI	Subscriber identity
MAP-DEACTIVATE-SS	IMSI	Subscriber identity
MAP-INTERROGATE-SS	IMSI	Subscriber identity
MAP-REGISTER-PASSWORD	IMSI	Subscriber identity
MAP-PROCESS-UNSTRUCTURED-SS-REQUEST	IMSI	Subscriber identity
MAP-UNSTRUCTURED-SS-REQUEST	IMSI	Subscriber identity
MAP-UNSTRUCTURED-SS-NOTIFY	IMSI	Subscriber identity
MAP-FORWARD-SHORT-MESSAGE	IMSI (note)	Subscriber identity

NOTE: Only when the IMSI and the LMSI are received together from the HLR in the mobile terminated short message transfer.

Table 5.3/2: Use of the destination reference

Originating address:

A valid SCCP address identifying the requestor of a MAP dialogue (see also clause 4). As an implementation option, this parameter may also, in the request, be implicitly associated with the service access point at which the primitive is issued.

Originating-reference:

This parameter is a reference which refines the identification of the calling process. It may be identical to the Originating address but its value is to be carried at MAP level. Table 5.3/3 describes the MAP services using the parameter. Only these services are allowed to use it. Processing of the Originating-reference shall be performed according to the supplementary service descriptions and other service descriptions, e.g. operator determined barring.

MAP service	Reference type	Use of the parameter
MAP-REGISTER-SS	ISDN-Address-String	Originated entity address
MAP-ERASE-SS	ISDN-Address-String	Originated entity address
MAP-ACTIVATE-SS	ISDN-Address-String	Originated entity address
MAP-DEACTIVATE-SS	ISDN-Address-String	Originated entity address
MAP-INTERROGATE-SS	ISDN-Address-String	Originated entity address
MAP-REGISTER-PASSWORD	ISDN-Address-String	Originated entity address
MAP-PROCESS-UNSTRUCTURED-SS-REQUEST	ISDN-Address-String	Originated entity address

Table 5.3/3: Use of the originating reference

Specific information:

This parameter may be used for passing any user specific information. Establishment and processing of the Specific information is not specified by GSM and shall be performed according to operator specific requirements.

Responding address:

An address identifying the responding entity. The responding address is included if required by the context (e.g. if it is different from the destination address).

Result:

This parameter indicates whether the dialogue is accepted by the peer.

Refuse reason:

This parameter is only present if the Result parameter indicates that the dialogue is refused. It takes one of the following values:

- Application-context-not-supported;

- Invalid-destination-reference;

- Invalid-originating-reference;

- No-reason-given;

- Remote node not reachable;

- Potential version incompatibility.

5.3.2 MAP-CLOSE service

This service is used for releasing a previously established MAP dialogue. The service may be invoked by either MAP service-user depending on rules defined within the service-user. The service is an unconfirmed service with parameters as shown in table 5.3/4.

Parameters	Request	Indication
Release method	M	
Specific Information	U	C(=)

Table 5.3/4: Service-primitives for the MAP-CLOSE service

Release method:

This parameter can take the following two values:

- normal release; in this case the primitive is mapped onto the protocol and sent to the peer;

- prearranged end; in this case the primitive is not mapped onto the protocol. Prearranged end is managed independently by the two users, i.e. only the request type primitive is required in this case.

Specific information:

This parameter may be used for passing any user specific information. Establishment and processing of the Specific information is not specified by GSM GSM and shall be performed according to operator specific requirements.

5.3.3 MAP-DELIMITER service

This service is used to explicitly request the transfer of the MAP protocol data units to the peer entities.

See also subclause 5.4 and 5.5 for the detailed use of the MAP-DELIMITER service.

The service is an unconfirmed service with service-primitives as shown in table 5.3/5.

Parameters	Request	Indication

Table 5.3/5: Service-primitives for the MAP-DELIMITER service

5.3.4 MAP-U-ABORT service

This service enables the service-user to request the MAP dialogue to be aborted. The service is an unconfirmed service with service-primitives as shown in table 5.3/6.

Parameters	Request	Indication
User reason	M	M(=)
Diagnostic information	U	C(=)
Specific information	U	C(=)

Table 5.3/6: Service-primitives for the MAP-U-ABORT service

User reason:

This parameter can take the following values:

- resource limitation (congestion);
  the requested user resource is unavailable due to congestion;

- resource unavailable;
  the requested user resource is unavailable for reasons other than congestion;

- application procedure cancellation;
  the procedure is cancelled for reason detailed in the diagnostic information parameter;

- procedure error;
  processing of the procedure is terminated for procedural reasons.

Diagnostic information:

This parameter may be used to give additional information for some of the values of the user-reason parameter:

User reason	Diagnostic information
Resource limitation (congestion)	-
Resource unavailable	Short term/long term problem
Application procedure cancellation	Handover cancellation/
	Radio Channel release/
	Network path release/
	Call release/
	Associated procedure failure/
	Tandem dialogue released/
	Remote operations failure
Procedure error	-

Table 5.3/7: User reason and diagnostic information

Specific information:

This parameter may be used for passing any user specific information. Establishment and processing of the Specific information is not specified by GSM and shall be performed according to operator specific requirements.

5.3.5 MAP-P-ABORT service

This service enables the MAP service-provider to abort a MAP dialogue. The service is a provider-initiated service with service-primitive as shown in table 5.3/8.

Parameters		Indication
Provider reason		M
Source		M

Table 5.3/8: Service-primitive for the MAP-P-ABORT service

Provider reason:

This parameter indicates the reason for aborting the MAP dialogue:

- provider malfunction;

- supporting dialogue/transaction released;

- resource limitation;

- maintenance activity;

- version incompatibility;

- abnormal MAP dialogue.

Source:

This parameter indicates the source of the abort. For Transaction Capabilities (TC) applications the parameter may take the following values:

- MAP problem;

- TC problem;

- network service problem.

Provider reason	Source	Corresponding event
Provider malfunction	MAP	Malfunction at MAP level at peer entity
	TC	"Unrecognised message type" or "Badly formatted transaction portion" or "Incorrect transaction portion" received in TC-P-ABORT "Abnormal dialogue"
	Network service	Malfunction at network service level at peer entity
Supporting dialogue/ transaction released	TC	"Unrecognised transaction ID" received in TC-ABORT
Resource limitation	MAP	Congestion towards MAP peer service-user
	TC	"Resource limitation" received in TC-P-ABORT
Maintenance activity	MAP	Maintenance at MAP peer service-user
	Network service	Maintenance at network peer service level
Abnormal MAP dialogue	MAP	MAP dialogue is not in accordance with specified application context
Version incompatibility	TC	A Provider Abort indicating "No common dialogue portion" is received in the dialogue initiated state

Table 5.3/9: Values of provider reason and source parameters and examples of corresponding events

5.3.6 MAP-NOTICE service

This service is used to notify the MAP service-user about protocol problems related to a MAP dialogue not affecting the state of the protocol machines.

The service is a provider-initiated service with service-primitive as shown in table 5.3/10.

Parameters		Indication
Problem diagnostic		M

Table 5.3/10: Service-primitive for the MAP-NOTICE service

Problem diagnostic:

This parameter can take one of the following values:

- abnormal event detected by the peer;
- response rejected by the peer;
- abnormal event received from the peer-   message cannot be delivered to the peer.

5.4 Sequencing of services

The sequencing of services is shown in figure 5.4/1 and is as follows:

Opening:

The MAP-OPEN service is invoked before any user specific service-primitive is accepted. The sequence may contain none, one or several user specific service-primitives. If no user specific service-primitive is contained between the MAP-OPEN and the MAP-DELIMITER primitives, then this will correspond to sending an empty Begin message in TC. If more than one user specific service-primitive is included, all are to be sent in the same Begin message. The sequence ends with a MAP-DELIMITER primitive.

Continuing:

This sequence may not be present in some MAP dialogues. If it is present, it ends with a MAP-DELIMITER primitive. If more than one user specific service-primitive is included, all are to be included in the same Continue message.

Closing:

The sequence can only appear after an opening sequence or a continuing sequence. The sequence may contain none, one or several user specific service-primitives if the MAP-CLOSE primitive specifies normal release. If no user specific service-primitive is included, then this will correspond to sending an empty End message in TC. If more than one user specific service-primitive is included, all are to be sent in the same End message. If prearranged end is specified, the sequence cannot contain any user specific service-primitive. The MAP-CLOSE primitive must be sent after all user specific service-primitives have been delivered to the MAP service-provider.

Aborting:

A MAP service-user can issue a MAP-U-ABORT primitive at any time after the MAP dialogue has been opened or as a response to an attempt to open a MAP dialogue.

The MAP service-provider may issue at any time a MAP-P-ABORT primitive towards a MAP service-user for which a MAP dialogue exists.

MAP-U-ABORT primitives and MAP-P-ABORT primitives terminate the MAP dialogue.

a) Opening

b) Continuing

c) Closing
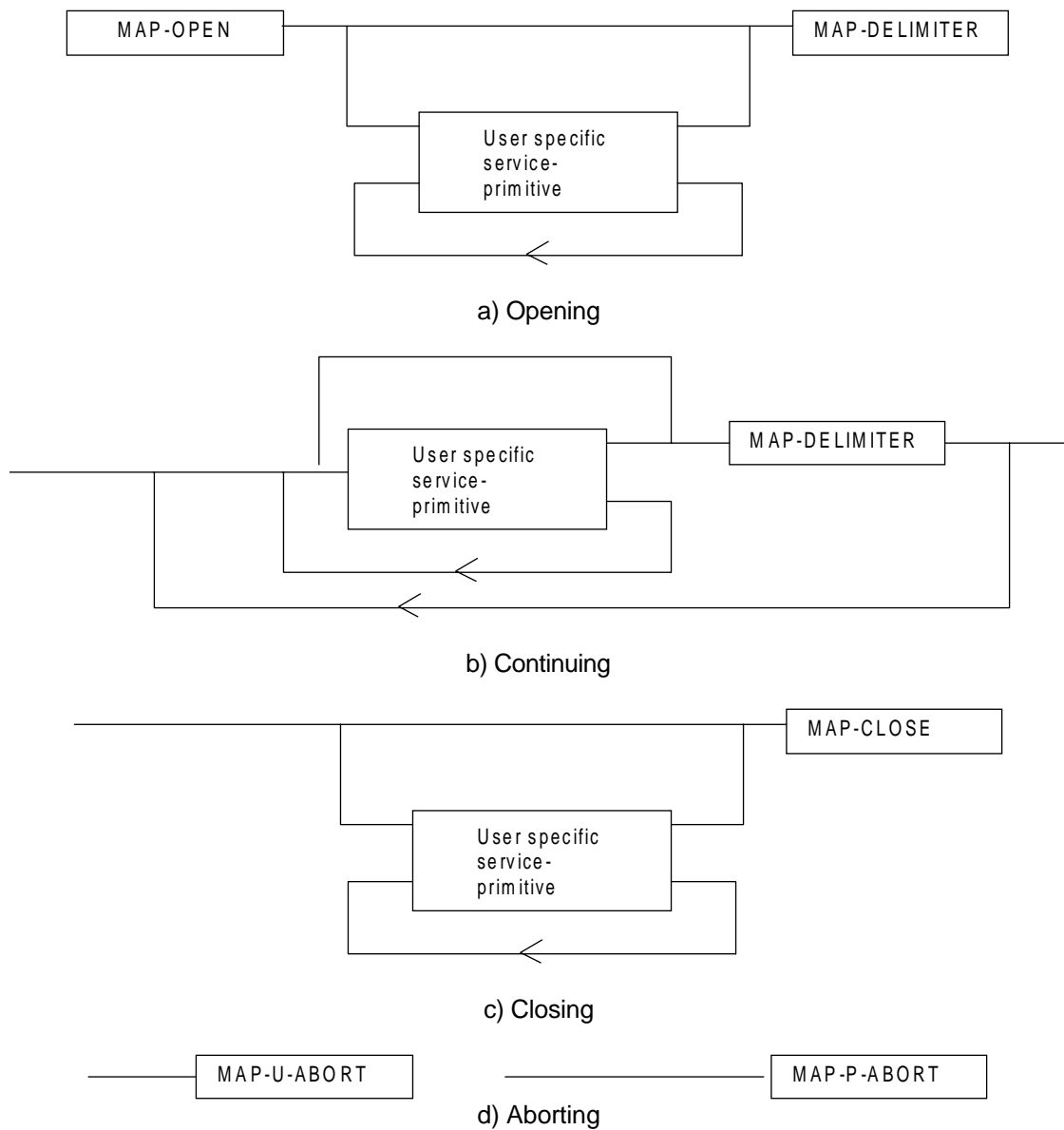
d) Aborting

Figure 5.4/1: Sequencing of services

If the reason "resource unavailable (short term problem)" is indicated in the MAP-U-ABORT indication primitive, the MAP service-user may decide to attempt a new MAP dialogue establishment immediately.

Sequencing of user specific service-primitives is done by the MAP service-user and based on rules applicable for each MAP service-user instance.

A MAP-NOTICE indication primitive may be received at any time during the active period of a MAP dialogue.

5.5 General rules for mapping of services onto TC

5.5.1 Mapping of common services

Table 5.5/1 gives an overview of the mapping rules for mapping of common services onto TC-services. Table 5.5/2 gives the mapping rules for mapping of TC-services onto common services.

Protocol machine description is given in clauses 11 to 14.

MAP service-primitive	TC service-primitive
MAP-OPEN request (+ any user specific service primitives) + MAP-DELIMITER request	TC-BEGIN request (+ component handling primitives)
MAP-OPEN response (+ any user specific service primitives) + MAP-DELIMITER request	TC-CONTINUE request (note) (+ component handling primitives)
(any user specific service primitives) + MAP-DELIMITER request	TC-CONTINUE request (+ component handling primitives)
(any user specific service primitives) + MAP-CLOSE request	TC-END request (+ component handling primitives)
MAP-U-ABORT request	TC-U-ABORT request

NOTE: or TC-END if the MAP-CLOSE request has been received before the MAP-DELIMITER request.

Table 5.5/1: Mapping of common services on to TC services

TC service-primitive	MAP service-primitive
TC-BEGIN indication (+ component handling primitives)	MAP-OPEN indication (+ user specific service primitives) + MAP-DELIMITER indication (note 1)
TC-CONTINUE indication (+ component handling primitives)	First time: MAP-OPEN confirm (+ user specific service primitives) + MAP-DELIMITER indication (note 1) Subsequent times: (user specific service primitives) + MAP-DELIMITER indication (note 1)
TC-END indication (+ component handling primitives)	MAP-OPEN confirm (note 6) (user specific service primitives) + MAP-CLOSE indication
TC-U-ABORT indication	MAP-U-ABORT indication or MAP-P-ABORT indication (note 2) MAP-OPEN confirmation (note 3)
TC-P-ABORT indication	MAP-P-ABORT indication (note 4) MAP-OPEN confirmation (note 5)

NOTE 1: It may not be necessary to present this primitive to the user for MAP version 2 applications.

NOTE 2: The mapping depends on whether the TC-U-ABORT indication primitive contains a MAP-abort-PDU from the remote MAP service-provider or a MAP-user-abort-PDU from the remote MAP service-user.

NOTE 3: Only if the opening sequence is pending and if the "Abort Reason" in the TC-U-ABORT indication is set to "Application Context Not Supported".

NOTE 4: If the "Abort Reason" in the TC-P-ABORT indication is set to a value different from "Incorrect Transaction Portion".

NOTE 5: Only if the opening sequence is pending and if the "Abort Reason" in the TC-P-ABORT indication is set to "Incorrect Transaction Portion".

NOTE 6: Only if opening sequence is pending.

Table 5.5/2: Mapping of TC services on to common service

5.5.2 Mapping of user specific services

Table 5.5/3 gives the general mapping rules which apply to mapping of MAP user specific services onto TC services and table 5.5/4 gives the similar rules for mapping of TC services onto MAP user specific services. Detailed mapping is given in clauses 11 to 14.

MAP service-primitive	TC-service-primitive
MAP-xx request	TC-INVOKE request
MAP-xx response (note 1)	TC-RESULT-L request TC-U-ERROR request TC-U-REJECT request TC-INVOKE request (note 2)

Table 5.5/3: Mapping of MAP user specific services onto TC services

TC-service-primitive	MAP service-primitive
TC-INVOKE indication	MAP-xx indication
TC-RESULT-L indication (note 4) TC-U-ERROR indication TC-INVOKE indication (note 2) TC-L-CANCEL indication	MAP-xx confirm
TC-U-REJECT indication TC-L-REJECT indication TC-R-REJECT indication	MAP-xx confirm or MAP-NOTICE indication (note 3)

Table 5.5/4: Mapping of TC services onto MAP user specific services

Notes to tables 5.10 and 5.11:

NOTE 1: The mapping is determined by parameters contained in the MAP-xx response primitive.

NOTE 2: This applies only to TC class 4 operations where the operation is used to pass a result of another class 2 or class 4 operation.

NOTE 3: The detailed mapping rules are given in clause 13.

NOTE 4: If RESULT-NL components are present they are mapped on to the same MAP-xx confirm.

5.6 Definition of parameters

Following is an alphabetic list of parameters used in the common MAP-services in subclause 5.3:

Application context name	5.3.1	Refuse reason	5.3.1
Destination address	5.3.1	Release method	5.3.2
Destination reference	5.3.1	Responding address	5.3.1
Diagnostic information	5.3.4	Result	5.3.1
Originating address	5.3.1	Source	5.3.5
Originating reference	5.3.1	Specific information	5.3.1/5.3.2/5.3.4
Problem diagnostic	5.3.6	User reason	5.3.4
Provider reason	5.3.5		

Following is an alphabetic list of parameters contained in this clause:

Access connection status	5.6.9.3	MS ISDN	5.6.2.17
Access signalling information	5.6.9.5	MSIsdn-Alert	5.6.2.29
Alert Reason	5.6.8.8	MSC number	5.6.2.11
Authentication set list	5.6.7.1	MWD status	5.6.8.3
Basic Service Group	5.6.3.5	Network resources	5.6.10.1
Bearer service	5.6.3.3	Network signal information	5.6.9.8
BSS-apdu	5.6.9.1	New password	5.6.4.20
Call barring feature	5.6.4.19	No reply condition timer	5.6.4.7
Call barring information	5.6.4.18	ODB General Data	5.6.3.9
Call reference	5.6.5.1	ODB HPLMN Specific Data	5.6.3.10
Called number	5.6.2.24	OMC Id	5.6.2.18
Calling number	5.6.2.25	Originally dialled number	5.6.2.26
Category	5.6.3.1	Originating entity number	5.6.2.10
Ciphering mode	5.6.7.7	Override Category	5.6.4.4
Cksn	5.6.7.5	Previous location area Id	5.6.2.4
CLI Restriction	5.6.4.5	Protocol Id	5.6.9.7
CM service type	5.6.9.2	Provider error	5.6.1.3
CUG feature	5.6.4.7	Rand	5.6.7.2
CUG index	5.6.4.11	Regional Subscription Data	5.6.3.11
CUG info	5.6.4.8	Regional Subscription Response	5.6.3.12
CUG interlock	5.6.4.10	Roaming number	5.6.2.19
CUG Outgoing Access indicator	5.6.3.8	Roaming Restriction Due To Unsupported Feature	5.6.3.13
CUG subscription	5.6.4.9	Service centre address	5.6.2.27
Current location area Id	5.6.2.6	SM Delivery Outcome	5.6.8.6
Current password	5.6.4.21	SM-RP-DA	5.6.8.1
Equipment status	5.6.3.2	SM-RP-OA	5.6.8.2
External Signal Information	5.6.9.4	SM-RP-PRI	5.6.8.5
Forwarded-to number	5.6.2.22	SM-RP-UI	5.6.8.4
Forwarded-to subaddress	5.6.2.23	Sres	5.6.7.3
Forwarding data	5.6.4.17	SS-Code	5.6.4.1
Forwarding feature	5.6.4.16	SS-Data	5.6.4.3
Forwarding information	5.6.4.15	SS-Info	5.6.4.24
Forwarding Options	5.6.4.6	SS-Status	5.6.4.2
GSM bearer capability	5.6.3.6	Stored location area Id	5.6.2.5
Guidance information	5.6.4.22	Subscriber Status	5.6.3.7
Handover number	5.6.2.21	Target cell Id	5.6.2.8
HLR Id	5.6.2.15	Target location area Id	5.6.2.7
HLR number	5.6.2.13	Target MSC number	5.6.2.12
HO-Number Not Required	5.6.6.7	Teleservice	5.6.3.4
IMEI	5.6.2.3	TMSI	5.6.2.2
IMSI	5.6.2.1	Trace reference	5.6.10.2
Inter CUG options	5.6.4.13	Trace type	5.6.10.3
Intra CUG restrictions	5.6.4.14	User error	5.6.1.4
Invoke Id	5.6.1.1	USSD Data Coding Scheme	5.6.4.36
Kc	5.6.7.4	USSD String	5.6.4.37
Linked Id	5.6.1.2	VLR number	5.6.2.14
LMSI	5.6.2.16	Zone Code	5.6.2.28
Location update type	5.6.9.6		
More Messages To Send	5.6.8.7		

5.6.1 Common parameters

The following set of parameters are used in several MAP service-primitives:

5.6.1.1 Invoke Id

This parameter identifies corresponding service primitives. The parameter is supplied by the MAP service-user and must be unique over each service-user/service-provider interface.

5.6.1.2 Linked Id

This parameter us used for linked services and it takes the value of the invoke Id of the service linked to.

5.6.1.3 Provider error

This parameter is used to indicate a protocol related type of error:

- duplicated invoke Id;
- not supported service;
- mistyped parameter;
- resource limitation;
- initiating release, i.e. the peer has already initiated release of the dialogue and the service has to be released;
- unexpected response from the peer;
- service completion failure;
- no response from the peer;
- invalid response received.

5.6.1.4 User error

This parameter can take values as follows:

> NOTE: The values are grouped in order to improve readability; the grouping has no other significance.

a) Generic error:
- system failure, i.e. a task cannot be performed because of a problem in another entity. The type of entity or network resource may be indicated by use of the network resource parameter;

- data missing, i.e. an optional parameter required by the context is missing;

- unexpected data value, i.e. the data type is formally correct but its value or presence is unexpected in the current context;

- resource limitation;

- initiating release, i.e. the receiving entity has started the release procedure;

- facility not supported, i.e. the requested facility is not supported by the PLMN.

b)   Identification or numbering problem:
   -   unknown subscriber, i.e. no such subscription exists;

   -   number changed, i.e. the subscription does not exist for that number any more;

   -   unknown MSC;

   -   unidentified subscriber, i.e. if the subscriber is not contained in the database and it has not or cannot be established whether or not a subscription exists;

   -   unallocated roaming number;

   -   unknown BSS;

   -   unknown equipment;

   -   unknown location area.

c)   Subscription problem:
   -   roaming not allowed, i.e. a location updating attempt is made in an area not covered by the subscription;

   -   illegal subscriber, i.e. illegality of the access has been established by use of authentication procedure;

   -   bearer service not provisioned;

   -   teleservice not provisioned;

   -   illegal equipment, i.e. the IMEI check procedure has shown that the IMEI is blacklisted or not whitelisted.

d)   Handover problem:
   -   invalid target base station, i.e. the identified base station does not exist;

   -   no radio resource available, i.e. the radio path cannot be established due to radio congestion;

   -   no handover number available;

   -   subsequent handover failure, i.e. handover to a third MSC failed for some reason.

e)   Operation and maintenance problem:
   -   tracing buffer full, i.e. tracing cannot be performed because the tracing capacity is exceeded.

f)   Call set-up problem:
   -   no roaming number available, i.e. a roaming number cannot be allocated because all available numbers are in use;

   -   absent subscriber, i.e. the subscriber has activated the detach service or the system detects the absence condition. If this user error value is used for the short message service, an additional indication whether or not the message waiting data flag has been set, may be included;

- busy subscriber, provides additional information indicating one of three possible categories of Busy Subscriber:

  - Network Determined User Busy (NDUB)
    This condition occurs when a call is about to be offered and the information (i.e. Traffic) channel is busy/not available and the maximum number of total calls has been reached. The value of maximum number of calls is 1 for the basic call. When the supplementary service Call Waiting is applicable, the value is n+1, where n is the maximum number of calls that can be waiting (reference GSM 02.01).

  - More Calls Allowed
    This condition occurs when a call is about to be offered and the called MS is already busy but the maximum number of total calls has not yet been reached (reference GSM 02.01).

  - User Determined User Busy (UDUB)
    This condition occurs when a call is offered to an MS and the MS responds "User Busy" because the subscribers resources (terminal or person using them) are busy (reference GSM 02.01).

- no subscriber reply;

- radio congestion;

- impossible call completion, i.e. the call can neither be established nor forwarded and none of the above causes will be reported. A list of SS-Code parameter (see subclause 5.6.1.4) may be appended in order to indicate which supplementary services are active.

- forwarding violation, i.e. the call has already been forwarded the maximum number of times that is allowed;

- CUG reject, i.e. the call does not pass a CUG check; additional information may also be given in order to indicate rejection due to e.g. incoming call barred or non-CUG membership.

- call barred. Optionally, additional information may be included for indicating either that the call meets a barring condition set by the subscriber or that the call is barred for operator reasons.

g)   Supplementary services problem:
   - call barred;
   - illegal SS operation;
   - SS error status;
   - SS not available;
   - SS subscription violation;
   - SS incompatibility;
   - negative password check;
   - password registration failure;
   - Number of Password Attempts;
   - USSD Busy;
   - Unknown Alphabet.

For definition of these errors see GSM 04.80.

h) Short message problem:
- SM delivery failure with detailed reason as follows:
  - memory capacity exceeded;
  - MS protocol error;
  - MS not equipped;
  - unknown service centre (SC);
  - SC congestion;
  - invalid SME address;
  - subscriber is not an SC subscriber;
  - and possibly detailed diagnostic information, coded as specified in GSM 03.40, under SMS-SUBMIT-REPORT and SMS-DELIVERY-REPORT. If the SM entity which returns the SM Delivery Failure error includes detailed diagnostic information, it shall be forwarded in the MAP_FORWARD_SHORT_MESSAGE response.

- message waiting list full, i.e. no further SC address can be added to the message waiting list;

- Subscriber busy for MT SMS, i.e. the mobile terminated short message transfer cannot be completed because another mobile terminated short message transfer is going on.

5.6.2 Numbering and identification parameter

5.6.2.1 IMSI

This parameter is the International Mobile Subscriber Identity defined in GSM 03.03.

5.6.2.2 TMSI

This parameter is the Temporary Mobile Subscriber Identity defined in GSM 03.03.

5.6.2.3 IMEI

This parameter is the International Mobile Equipment Identity defined in GSM 03.03.

5.6.2.4 Previous location area Id

This parameter refers to the identity of the location area from which the subscriber has roamed.

5.6.2.5 Stored location area Id

This parameter refers to the location area where the subscriber is assumed to be located.

5.6.2.6 Current location area Id

This parameter is used to indicate the location area in which the subscriber is currently located.

5.6.2.7 Target location area Id

This parameter refers to the location area into which the subscriber intends to roam.

5.6.2.8 Target cell Id

This parameter refers to the identity of the cell to which a call has to be handed over.

5.6.2.9 [Spare]

5.6.2.10 Originating entity number

This parameter refers to an application layer identification of a system component in terms of its associated ISDN number.

5.6.2.11   MSC number

This parameter refers to the ISDN number of an MSC.

5.6.2.12   Target MSC number

This parameter refers to the ISDN number of an MSC to which a call has to be handed over.

5.6.2.13   HLR number

This parameter refers to the ISDN number of an HLR.

5.6.2.14   VLR number

This parameter refers to the ISDN number of a VLR.

5.6.2.15   HLR Id

This parameter refers to the identity of an HLR derived from the IMSI defined in CCITT Recommendation E.212.

5.6.2.16   LMSI

This parameter refers to a local identity allocated by the VLR to a given subscriber for internal management of data in the VLR.

5.6.2.17   MS ISDN

This parameter refers to one of the ISDN numbers assigned to a mobile subscriber in accordance with CCITT Recommendation E.213.

5.6.2.18   OMC Id

This parameter refers to the identity of an operation and maintenance centre.

5.6.2.19   Roaming number

This parameter refers to the roaming number as defined in CCITT Recommendation E.213.

5.6.2.20   [Spare]

5.6.2.21   Handover number

This parameter refers to the number used for routing a call between MSCs during handover.

5.6.2.22   Forwarded-to number

This parameter refers to the address to which a call is to be forwarded. This may include a subaddress.

5.6.2.23   Forwarded-to subaddress

This parameter refers to the sub-address attached to the address to which a call is to be forwarded.

5.6.2.24   Called number

This parameter refers to a called party number as defined in CCITT Recommendation Q.767.

5.6.2.25   Calling number

This parameter refers to a calling party number as defined in CCITT Recommendation Q.767.

5.6.2.26   Originally dialled number

This parameter refers to the number dialled by the calling party in order to reach a mobile subscriber.

5.6.2.27   Service centre address

This parameter represents the address of a Short Message Service Centre.

5.6.2.28   Zone Code

This parameter is used to define location areas into which the subscriber is allowed or not allowed to roam (regional subscription). With a complete list of Zone Codes the VLR is able to determine for all its location areas whether roaming is allowed or not.

5.6.2.29   MSIsdn-Alert

This parameter refers to the MSISDN stored in a Message Waiting Data File in the HLR. It is used to alert the Service Centre when the MS is again attainable.

5.6.3   Subscriber management parameters

5.6.3.1   Category

This parameter refers to the calling party category as defined in CCITT Recommendation Q.767.

5.6.3.2   Equipment status

This parameter refers to the status of the mobile equipment as defined in GSM 02.16.

5.6.3.3   Bearer service

This parameter may refer to a single bearer service, a set of bearer services or to all bearer services as defined in GSM 02.02.

5.6.3.4   Teleservice

This parameter may refer to a single teleservice, a set of teleservices or to all teleservices as defined in GSM 02.03.

5.6.3.5   Basic Service Group

This parameter refers to the Basic Service Group either as bearer service (see subclause 5.6.3.3) or teleservice (see subclause 5.6.3.4).

5.6.3.6   GSM bearer capability

This parameter refers to the GSM bearer capability information element defined in GSM 04.08.

5.6.3.7   Subscriber Status

This parameter refers to the barring status of the subscriber:

- service granted;
- Operator Determined Barring.

5.6.3.8   CUG Outgoing Access indicator

This parameter represents the Outgoing Access as defined in ETS 300 136.

5.6.3.9 Operator Determined Barring General Data

This parameter refers to the set of subscribers features that the network operator or the service provider can regulate. This set only includes those limitations that can be controlled in the VLR:

- All outgoing calls barred;
- International outgoing calls barred;
- International outgoing calls except those to the home PLMN country barred;
- Premium rate (information) outgoing calls barred;
- Premium rate (entertainment) outgoing calls barred;
- Supplementary service access barred.

5.6.3.10 ODB HPLMN Specific Data

This parameter refers to the set of subscribers features that the network operator or the service provider can regulate only when the subscriber is registered in the HPLMN. This set only includes those limitations that can be controlled in the VLR:

- Operator Determined Barring Type 1;
- Operator Determined Barring Type 2;
- Operator Determined Barring Type 3;
- Operator Determined Barring Type 4.

5.6.3.11 Regional Subscription Data

This parameter defines the regional subscription area in which the subscriber is allowed to roam. It consists of a list of Zone Codes (see subclause 5.6.2.28).

5.6.3.12 Regional Subscription Response

This parameter indicates either that the regional subscription data cannot be handled or that the current MSC area is entirely restricted because of regional subscription.

5.6.3.13 Roaming Restriction Due To Unsupported Feature

This parameter defines that a subscriber is not allowed to roam in the current MSC area. It may be used by the HLR if a feature or service is indicated as unsupported by the VLR.

5.6.4 Supplementary services parameters

5.6.4.1 SS-Code

This parameter may refer to one supplementary service or a set of supplementary services as defined in GSM 02.04. For MAP version 2 this includes:

- Calling Line Identification Presentation service (CLIP);
- Calling Line Identification Restriction service (CLIR);
- Connected Line Identification Presentation service (COLP);
- Connected Line Identification Restriction service (COLR);
- All Call Forwarding services;
- Call Waiting (CW);
- Call Hold (HOLD);
- Multi-Party service (MPTY);
- Closed User Group (CUG);
- All Charging services;
- All Call Restriction services.
- Explicit Call Transfer service (ECT)

5.6.4.2 SS-Status

This parameter refers to the state information of individual supplementary services as defined in GSM 03.11.

5.6.4.3 SS-Data

This parameter refers to the necessary set of information required in order to characterise one supplementary service:

- SS-Code (see subclause 5.6.4.1);
- SS-Status (if applicable) (see subclause 5.6.4.2);
- Override subscription option (see subclause 5.6.4.4);
- CLI Restriction (see subclause 5.6.4.5);
- Basic Service Group Code (see subclause 5.6.3.5).

5.6.4.4 Override Category

This parameter refers to the subscription option Override Category attached to a supplementary service. It can take the following two values:

- Enabled;
- Disabled.

5.6.4.5 CLI Restriction Option

This parameter refers to the subscription option Restriction mode attached to the CLIR supplementary service. It can take the following three values:

- Permanent;
- Temporary (Default Restricted);
- Temporary (Default Allowed).

5.6.4.6 Forwarding Options

This parameter refers to a forwarding option attached to a supplementary service. It can take one of the following values:

- notification to forwarding party (see GSM 02.82 for the meaning of this parameter);
- notification to calling party (see GSM 02.82 for the meaning of this parameter);
- Forwarding reason (see GSM 02.82 for the meaning of this parameter).

5.6.4.7 No reply condition timer

This parameter refers to the no reply condition timer for call forwarding on no reply.

5.6.4.8 CUG info

This parameter refers to the overall information required for operation for each CUG:

- CUG subscriptionlist;
- CUG feature.

5.6.4.9 CUG subscription

This parameter refers to the set of basic information for each CUG defined in that subscription. The following information is stored:

- CUG index;
- CUG interlock;

- Intra CUG restrictions;
- Basic Service Group List.

5.6.4.10 CUG interlock

This parameter represents the CUG interlock code defined in ETS 300 138.

5.6.4.11 CUG index

This parameter represents the CUG index defined in ETS 300 138.

5.6.4.12 CUG feature

This parameter contains two parameter which are associated with the Basic Service Group. If the Basic Service Group Code is not present the features applies to all Basic Services. The following two parameters are included:

- Preferential CUG indicator:
  indicates which CUG index is to be used at outgoing call set-up using the associated Basic Service Group;

- Inter CUG Option:
  describes whether it for the associated Basic Service Group is allowed to make calls outside the CUG and whether incoming calls are allowed;

- Basic Service Group.

See GSM 02.85 for meaning of this parameter.

5.6.4.13 Inter CUG options

This parameter indicates the subscribers ability to make and receive calls outside a specific closed user group. It takes the following values:

- CUG only facility (only calls within CUG are allowed);
- CUG with outgoing access (calls outside CUG allowed);
- CUG with incoming access (calls from outside CUG into CUG allowed);
- CUG with both incoming and outgoing access (all calls allowed).

5.6.4.14 Intra CUG restrictions

This parameter describes whether or not it is allowed to originate calls to or to receive calls from within the CUG. It can take the following values:

- no CUG restrictions;
- CUG incoming calls barred;
- CUG outgoing calls barred.

5.6.4.15 Forwarding information

This parameter represents the information related to each call forwarding service:

- the SS-Code of the relevant call forwarding service   (see subclause 5.6.4.1);

- if required, a list of forwarding feature parameters   (see subclause 5.6.4.16).
  The list may contain one item per Basic Service Group.

5.6.4.16 Forwarding feature

This parameter applies to each combination of call forwarding service and Basic Service Group and contains the following information, as required:

- Basic Service Group (see subclause 5.6.3.5);
- SS-Status (see subclause 5.6.4.2);
- forwarded-to number (see subclause 5.6.2.22);
- forwarded-to subaddress (see subclause 5.6.2.23);
- forwarding options (see subclause 5.6.4.6);
- no reply condition timer (see subclause 5.6.4.7).

5.6.4.17 Forwarding data

This parameter refers to the data required by the network for invoking a call forwarding service:

- forwarded-to number (see subclause 5.6.2.22);
- if required, forwarding options (see subclause 5.6.4.6).

5.6.4.18 Call barring information

This parameter contains for each call barring service:

- SS-Code (see subclause 5.6.4.1);
- a list of call barring feature parameters (see subclauseÿ5.6.4.19).
  The list may contain one item per Basic Service Group.

5.6.4.19 Call barring feature

This parameter gives the status of call barring services as applicable to each Basic Service Group. The parameter contains the following information:

- Basic Service Group (see subclause 5.6.3.5);
- SS-Status (see subclause 5.6.4.2).

5.6.4.20 New password

This parameter refers to the password which the subscriber just registered in the network.
This parameter refers to a password used by the subscriber for supplementary service control.

5.6.4.21 Current password

This parameter refers to a password used by the subscriber for supplementary service control.

5.6.4.22 Guidance information

This parameter refers to guidance information given to a subscriber who is requested to provide a password. One of the following information may be given:

- "enter password";
  This information is used for checking of the old password.

- "enter new password";
  This information is used during password registration for the request of the first new password.

- "enter new password again";
  This information is used during password registration for the request of the new password again for verification.

5.6.4.23   [Spare]

5.6.4.24   SS-Info

This parameter refers to all the information related to a supplementary service and is a choice between:

-     forwarding information   (see subclause 5.6.4.15);
-     call barring information   (see subclause 5.6.4.18);
-     CUG info   (see subclause 5.6.4.8);
-     SS-Data   (see subclause 5.6.4.3).

5.6.4.25-5.6.4.35 [Spare]

5.6.4.36   USSD Data Coding Scheme

This parameter contains the information of the alphabet and the language used for the unstructured information in an Unstructured Supplementary Service Data operation. The coding of this parameter is according to the Cell Broadcast Data Coding Scheme as specified in GSM 03.38.

5.6.4.37   USSD String

This parameter contains a string of unstructured information in an Unstructured Supplementary Service Data operation. The string is sent either by the mobile user or the network. The contents of a string sent by the MS are interpreted by the network as specified in GSM 02.90.

5.6.5   Call parameters

5.6.5.1   Call reference

This parameter refers to a call reference allocated by a call control MSC.

5.6.6   Radio parameters

5.6.6.1-5.6.6.6 [Spare]

5.6.6.7   HO-Number Not Required

This parameter indicates that no handover number allocation is necessary.

5.6.7   Authentication parameters

5.6.7.1   Authentication set list

5.6.7.1   Authentication set list

This parameter represents a list of sets of authentication parameters for a given subscriber:

-     Rand;
-     Sres;
-     Kc.

5.6.7.2   Rand

This parameter represents a random number used for authentication.

5.6.7.3   Sres

This parameter represents the response to an authentication request.

5.6.7.4 Kc

This parameter refers to a key used for ciphering purposes.

5.6.7.5 [Spare]

5.6.7.6 Cksn

This parameter refers to a ciphering key sequence number.

5.6.7.7 Ciphering mode

This parameter refers to the ciphering mode which is associated with a radio channel. It may take values as follows:

- no encryption;
- identification of specific ciphering algorithm.

5.6.8 Short message parameters

5.6.8.1 SM-RP-DA

This parameter represents the destination address used by the short message service relay sub-layer protocol. It can be either of the following:

- IMSI   (see subclause 5.6.2.1);
- LMSI   (see subclause 5.6.2.16);
- MS-ISDN   (see subclause 5.6.2.17);
- roaming number   (see subclause 5.6.2.19);
- service centre address   (see subclause 5.6.2.27).

5.6.8.2 SM-RP-OA

This parameter refers to the originating address used by the short message service relay sub-layer protocol. It can be either of the following:

- MS-ISDN   (see subclause 5.6.2.17);
- service centre address   (see subclause 5.6.2.27).

5.6.8.3 MWD status

This parameter indicates whether or not the address of the originator service centre is already contained in the Message Waiting Data file. In addition, it contains the status of the Memory Capacity Exceeded Flag (MCEF) and the status of the Mobile subscriber Not Reachable Flag (MNRF).

5.6.8.4 SM-RP-UI

This parameter represents the user data field carried by the short message service relay sub-layer protocol.

5.6.8.5 SM-RP-PRI

This parameter is used to indicate whether or not delivery of the short message shall be attempted when a service centre address is already contained in the Message Waiting Data file.

5.6.8.6 SM Delivery Outcome

This parameter indicates the cause for setting the message waiting data. It can take one of the following values:

- Absent subscriber;
- MS memory capacity exceeded;
- Successful transfer.

5.6.8.7 More Messages To Send

This parameter is used to indicate whether or not the service centre has more short messages to send.

5.6.8.8 Alert Reason

This parameter is used to indicate the reason why the service centre is alerted. It can take one of the following values:

- MS present;
- Memory Available.

5.6.9 Access and signalling system related parameters

5.6.9.1 BSS-apdu

This parameter includes one or two concatenated complete 08.06 messages, as described in GSM 03.09 and GSM 09.10. The Protocol ID indicates that the message or messages are according to GSM 08.06. For the coding of the messages see GSM 08.06 and GSM 08.08.

5.6.9.2 CM service type

This parameter identifies the service category being requested by the subscriber:

- mobile originating call;
- emergency call establishment;
- short message service;
- mobile originating call re-establishment;
- mobile terminating call;
- SS request.

5.6.9.3 Access connection status

This parameter represents the following access connection status information:

- RR-connection status (established/not established);
- ciphering mode (on/off);
- authentication status (authenticated/not authenticated).

5.6.9.4 External Signal Information

This parameter contains concatenated information elements (including tag and length) which are defined by a common protocol version, preceded by the associated protocol ID. It is used to transport information of the indicated protocol via MAP interfaces.

5.6.9.5 Access signalling information

This parameter refers to any set of information elements imported from GSM 04.08.

5.6.9.6 Location update type

This parameter refers to the location update type (normal, periodic or IMSI attach) contained in the GSM 04.08 LOCATION REGISTRATION REQUEST message.

5.6.9.7 Protocol ID

This parameter refers to the protocol to which the coding of the content of the associated External Signal Information conforms.

The following values are defined:

- 04.08;
- 08.06;
- ETS300102-1.
  This value indicates the protocol defined by ETS 300 102-1 (EDSS1).

5.6.9.8 Network signal information

This parameter is transported as external signal information. The protocol ID shall be set to "ETS 300 102-1".

The network signal information may include the following information elements as defined in GSM 09.07:

- ISDN BC; the tag and length are defined by ETS 300 102-1.
  For the content, see GSM 09.07.

- HLC; the tag and length are defined by ETS 300 102-1.
  For the content, see GSM 09.07.

- LLC; the tag and length are defined by ETS 300 102-1.
  For the content, see GSM 09.07.

They are contained in the Signal Information parameter according to figure 5.6/1 (irrespective of the order):

```
+-----------------------------+
|  +-----------------------+  |
|  |     ISDN BC TAG       |  |
|  +-----------------------+  |
|  |        LENGTH         |  |
|  +-----------------------+  |
|  |        CONTENT        |  |
|  +-----------------------+  |
|                             |
|  +-----------------------+  |
|  |       HLC TAG         |  |
|  +-----------------------+  |
|  |        LENGTH         |  |
|  +-----------------------+  |
|  |        CONTENT        |  |
|  +-----------------------+  |
|                             |
|  +-----------------------+  |
|  |       LLC TAG         |  |
|  +-----------------------+  |
|  |        LENGTH         |  |
|  +-----------------------+  |
|  |        CONTENT        |  |
|  +-----------------------+  |
+-----------------------------+
```
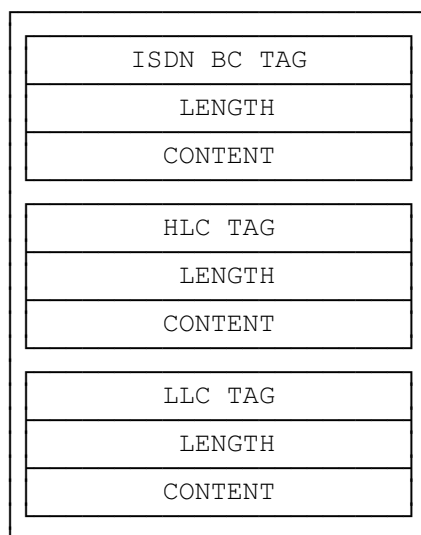
Figure 5.6/1: Network signal information parameter

5.6.10 System operations parameters

5.6.10.1 Network resources

This parameter refers to a class or type of network resource:

- PLMN;
- HLR;
- VLR (current or previous);

- MSC (controlling or current);
- EIR;
- radio sub-system.

5.6.10.2   Trace reference

This parameter represents a reference associated with a tracing request. The parameter is managed by OMC.

5.6.10.3   Trace type

This parameter identifies the type of trace. Trace types are fully defined in GSM 12.08.

5.7   Representation of a list of a basic parameter in service-primitives

In some service-primitives several instances of a basic parameter of subclause 5.6 are required. In the service descriptions such cases will be represented as

ParameterNameLIST

in the tables where ParameterName refers to one of the parameters defined in subclause 5.6. This corresponds to the following construction rule:
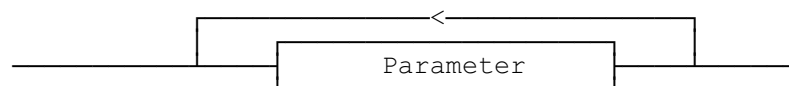
Figure 5.7/1: Construction of Lists

6 Mobility services

6.1   Location management services

6.1.1   MAP_UPDATE_LOCATION_AREA service

6.1.1.1   Definition

This service is used between MSC and VLR to update location information in the network. It is initiated by an MS when changing the location area or at first registration. The detailed conditions are given in GSM 03.12.

The MAP_UPDATE_LOCATION_AREA service is a confirmed service using the primitives from table 6.1/1.

6.1.1.2   Service primitives

Parameter name	Request	Indication	Response	Confirm
Invoke Id	M	M(=)	M(=)	M(=)
Target location area Id	M	M(=)		
Location update type	M	M(=)		
IMSI	C	C(=)		
TMSI	C	C(=)		
Previous location area Id	C	C(=)		
CKSN	C	C(=)		
User error			C	C(=)
Provider error				O

Table 6.1/1: MAP_UPDATE_LOCATION_AREA

6.1.1.3   parameter definitions and use

Invoke Id

See definition in subclause 5.6.1.

Target location area Id

See definition in subclause 5.6.2.

Location update type

See definition in subclause 5.6.9.

IMSI

See definition in subclause 5.6.2. It is up to the MS to provide either IMSI or TMSI, but one shall be present.

TMSI

See definition in subclause 5.6.2. It is up to the MS to provide either IMSI or TMSI, but one shall be present.

Previous location area Id

See definition in subclause 5.6.2. This parameter is provided if the updating is not a first registration.

CKSN

See definition in subclause 5.6.7. The CKSN is given if TMSI is used.

<u>User error</u>

One of the following error causes defined in subclause 5.6.1 is sent by the user in case of location area updating failures, depending on the failure reason:

- unknown subscriber;
  This cause is used if the subscriber is not known in the VLR and even a correlated request to the subscriber's HLR gives a negative result (i.e. the IMSI is not allocated to a subscriber).

- unknown location area;
  This cause is used if the target location area identity given is not known in the VLR.

- roaming not allowed;
  This cause is used if the MS is not allowed to roam into the target location area indicated in the MAP_UPDATE_LOCATION_AREA Req. The cause will be qualified according to the roaming restriction reason, i.e. one of "National Roaming Not Allowed", "PLMN Not Allowed", "Location Area Not Allowed", or "Operator Determined Barring".

- illegal subscriber;
  This error is sent if a correlated authentication procedure has not authenticated the subscriber.

- illegal equipment;
  This error is sent if an IMEI check failed, i.e. the IMEI is blacklisted or not white-listed.

- system failure;

- unexpected data value.

<u>Provider error</u>

For definition of provider errors see subclause 5.6.1.

6.1.2   MAP_UPDATE_LOCATION service

6.1.2.1   Definition

This service is used by the VLR to update the location information stored in the HLR.

The MAP_UPDATE_LOCATION service is a confirmed service using the service primitives given in table 6.1/2.

6.1.2.2   Service primitives

Parameter name	Request	Indication	Response	Confirm
Invoke Id	M	M(=)	M(=)	M(=)
IMSI	M	M(=)		
MSC Address	M	M(=)		
VLR number	M	M(=)		
LMSI	U	C(=)		
HLR number			C	C(=)
User error			C	C(=)
Provider error				O

Table 6.1/2: MAP_UPDATE_LOCATION

6.1.2.3 Parameter definitions and use

Invoke Id

See definition in subclause 5.6.1.

IMSI

See definition in subclause 5.6.2.

MSC Address

See definition in subclause 5.6.2. The MSC address is used for short message delivery only and for each incoming call set-up attempt the MSRN will be requested from the VLR.

VLR number

See definition in subclause 5.6.2.

LMSI

See definition in subclause 5.6.2. It is an operator option to provide the LMSI from the VLR; it is mandatory for the HLR to support the LMSI handling procedures.

HLR number

See definition in subclause 5.6.2. The presence of this parameter is mandatory in case of successful HLR updating.

User error

In case of unsuccessful updating, an error cause shall be returned by the HLR. The following error causes defined in subclause 5.6.1 may be used, depending on the nature of the fault:

- unknown subscriber;

- roaming not allowed;
  This cause will be sent if the MS is not allowed to roam into the PLMN indicated by the VLR number. The cause is qualified by the roaming restriction reason "PLMN Not Allowed" or "Operator Determined Barring". If no qualification is received (HLR with MAP Version 1), "PLMN Not Allowed" is taken as default.

- system failure;

- unexpected data value.

Provider error

For definition of provider errors see subclause 5.6.1.

6.1.3 MAP_CANCEL_LOCATION service

6.1.3.1 Definition

This service is used between HLR and VLR to delete a subscriber record from the VLR. It may be invoked automatically when an MS moves from one VLR area to another, to remove the subscriber record from the old VLR, or by the HLR operator to enforce a location updating from the VLR to the HLR, e.g. on withdrawal of a subscription.

The MAP_CANCEL_LOCATION service is a confirmed service using the primitives defined in table 6.1/3.

6.1.3.2   Service primitives

Parameter name	Request	Indication	Response	Confirm
Invoke Id	M	M(=)	M(=)	M(=)
IMSI	M	M(=)		
LMSI	C	C(=)		
User error			C	C(=)
Provider error				O

Table 6.1/3: MAP_CANCEL_LOCATION

6.1.3.3   Parameter definitions and use

Invoke Id

See definition in subclause 5.6.1.

IMSI

See definition in subclause 5.6.2.

LMSI

See definition in subclause 5.6.2. The LMSI shall be included if it has been received from VLR.

Value 0000 0000 can be used to indicate that the LMSI is not in use.

User error

If the cancellation fails, an error cause is to be returned by the VLR. The following error cause defined in subclause 5.6.1 shall be used:

-        unexpected data value.

Provider error

For definition of provider errors see subclause 5.6.1.

6.1.4   MAP_SEND_IDENTIFICATION service

6.1.4.1   Definition

The MAP_SEND_IDENTIFICATION service is used between a VLR and a previous VLR to retrieve IMSI and authentication sets for a subscriber registering afresh in that VLR.

The MAP_SEND_IDENTIFICATION service is a confirmed service using the service primitives defined in table 6.1/4.

6.1.4.2   Service primitives

Parameter name	Request	Indication	Response	Confirm
Invoke Id	M	M(=)	M(=)	M(=)
TMSI	M	M(=)		
IMSI			C	C(=)
Authentication set			U	C(=)
User error			C	C(=)
Provider error				O

Table 6.1/4: MAP_SEND_IDENTIFICATION

6.1.4.3 Parameter definitions and use

Invoke Id

See definition in subclause 5.6.1.

TMSI

See definition in subclause 5.6.2.

IMSI

See definition in subclause 5.6.2. The IMSI is to be returned if the service succeeds.

Authentication set

See definition in subclause 5.6.7. If the service succeeds a list of up to five authentication sets is returned, if there are any available.

User error

This parameter is mandatory if the service fails. The following error cause defined in subclause 5.6.1 may be used, depending on the nature of the fault:

- unidentified subscriber.

Provider error

For definition of provider errors see subclause 5.6.1.

6.1.5 MAP_DETACH_IMSI service

6.1.5.1 Definition

The MAP_DETACH_IMSI service is used by the MSC to indicate to the VLR that an MS is no longer reachable. The network needs this information e.g. to reject an incoming call without initiating paging on the radio path.

The MAP_DETACH_IMSI service is a non-confirmed service using the service primitives defined in table 6.1/5.

6.1.5.2 Service primitives

Parameter name	Request	Indication
Invoke Id	M	M(=)
IMSI	C	C(=)
TMSI	C	C(=)

Table 6.1/5: MAP_DETACH_IMSI

6.1.5.3 Parameter definitions and use

Invoke Id

See definition in subclause 5.6.1.

IMSI

See definition in subclause 5.6.2. It is up to the MS to provide either IMSI or TMSI as subscriber identity, but one shall be present.

TMSI

See definition in subclause 5.6.2. It is up to the MS to provide either IMSI or TMSI as subscriber identity, but one shall be present.

6.1.6 MAP_PURGE_MS service

6.1.6.1 Definition

This service is used between the VLR and the HLR to cause the HLR to mark its data for an MS so that any request for routing information for a mobile terminated call or a mobile terminated short message will be treated as if the MS is not reachable. It is invoked when the subscriber record for the MS is to be deleted in the VLR, either by MMI interaction or automatically, e.g. because the MS has been inactive for several days.

The MAP_PURGE_MS service is a confirmed service using the primitives defined in table 6.1/6.

6.1.6.2 Service primitives

Parameter name	Request	Indication	Response	Confirm
Invoke Id	M	M(=)	M(=)	M(=)
IMSI	M	M(=)		
VLR number	M	M(=)		
Provider error				O

Table 6.1/6: MAP_PURGE_MS

6.1.6.3 Parameter definitions and use

Invoke ID

See definition in subclause 5.6.1.

IMSI

See definition in subclause 5.6.2.

VLR number

See definition in subclause 5.6.2.

Provider error

See definition of provider errors in subclause 5.6.1.

6.2 Paging and search

6.2.1 MAP_PAGE service

6.2.1.1 Definition

This service is used between VLR and MSC to initiate paging of an MS for mobile terminated call set-up, mobile terminated short message or unstructured SS notification.

The MAP_PAGE service is a confirmed service using the primitives from table 6.2/1.

6.2.1.2 Service primitives

Parameter name	Request	Indication	Response	Confirm
Invoke Id	M	M(=)	M(=)	M(=)
IMSI	M	M(=)		
Stored location area Id	M	M(=)		
TMSI	U	C(=)		
User error			C	C(=)
Provider error				O

Table 6.2/1: MAP_PAGE

6.2.1.3 Parameter definitions and use

Invoke Id

See definition in subclause 5.6.1.

IMSI

See definition in subclause 5.6.2. The IMSI is used to define the paging subgroup. If the TMSI is not supplied, paging on the radio path uses the IMSI as an identifier.

Stored location area Id

See definition in subclause 5.6.2.

TMSI

See definition in subclause 5.6.2. The TMSI is included if paging on the radio channel is to use the TMSI as an identifier.

User error

The following error causes defined in subclause 5.6.1 may be sent by the user in case of a paging error, depending on the failure reason:

- absent subscriber;

- unknown location area;

- busy subscriber;

- system failure;
  This corresponds to the case where there is no call associated with the MAP_PAGE service, i.e. if the call has been released but the dialogue to the VLR has not been aborted.

- unexpected data value.

Provider error

See definition in subclause 5.6.1.

6.2.2   MAP_SEARCH_FOR_MS service

6.2.2.1   Definition

This service is used between VLR and MSC to initiate paging of an MS in all location areas of that VLR. It is used if the VLR does not hold location area information confirmed by radio contact.

The MAP_SEARCH_FOR_MS service is a confirmed service using the primitives from table 6.2/2.

6.2.2.2   Service primitives

Parameter name	Request	Indication	Response	Confirm
Invoke Id	M	M(=)	M(=)	M(=)
IMSI	M	M(=)		
Current location area Id			C	C(=)
User error			C	C(=)
Provider error				O

Table 6.2/2: MAP_SEARCH_FOR_MS

6.2.2.3   Parameter definitions and use

Invoke Id

See definition in subclause 5.6.1.

IMSI

See definition in subclause 5.6.2. The IMSI is used to identify the subscriber when paging on the radio path.

Current location area Id

See definition in subclause 5.6.2. In case of successful outcome of the service, i.e. if the MS responds to paging, the Location Area Id of the area in which the MS responded is given in the response.

User error

The following error causes defined in subclause 5.6.1 shall be sent by the user if the search procedure fails, depending on the failure reason:

-     absent subscriber;
      This error cause is returned by the MSC if the MS does not respond to the paging request.
-     system failure;
      This corresponds to the case where there is no call associated with the MAP_SEARCH_FOR_MS service, i.e. if the call has been released but the dialogue to the VLR has not been aborted.
-     busy subscriber;
-     unexpected data value.

Provider error

See definition in subclause 5.6.1.

6.3 Access management services

6.3.1 MAP_PROCESS_ACCESS_REQUEST service

6.3.1.1 Definition

This service is used between MSC and VLR to initiate processing of an MS access to the network, e.g. in case of mobile originated call set-up or after being paged by the network.

The MAP_PROCESS_ACCESS_REQUEST service is a confirmed service using the primitives from table 6.3/1.

6.3.1.2 Service primitives

Parameter name	Request	Indication	Response	Confirm
Invoke Id	M	M(=)	M(=)	M(=)
CM service type	M	M(=)		
Access connection status	M	M(=)		
Current Location Area Id	M	M(=)		
TMSI	C	C(=)		
Cksn	C	C(=)		
IMSI	C	C(=)	C	C(=)
IMEI	C	C(=)	C	C(=)
MSISDN			U	C(=)
User error			C	C(=)
Provider error				O

Table 6.3/1: MAP_PROCESS_ACCESS_REQUEST

6.3.1.3 Parameter definitions and use

Invoke Id

See definition in subclause 5.6.1.

CM service type

See definition in subclause 5.6.9.

Access connection status

See definition in subclause 5.6.9.

Current Location Area Id

See definition in subclause 5.6.2. This parameter is used to update the VLR in case of previous VLR failure.

TMSI

See definition in subclause 5.6.2. Either TMSI or IMSI as received from the MS are included in the Request/Indication, but one shall be present. In case of CM Service Type "Emergency Call Establishment", the IMEI may replace IMSI/TMSI.

Cksn

See definition in subclause 5.6.7. In case of access with TMSI, the Cksn shall be present.

IMSI

See definition in subclause 5.6.2. Either TMSI or IMSI as received from the MS are included in the Request/Indication, but one shall be present. In case of CM Service Type "Emergency Call Establishment", the IMEI may replace IMSI/TMSI.

In the Response/Confirmation, the IMSI is to be sent in case of successful outcome of the service. In case of CM Service Type "Emergency Call Establishment", IMEI may replace IMSI.

IMEI

See definition in subclause 5.6.2. The IMEI may replace IMSI/TMSI in the Request/Indication and IMSI in the Response/Confirmation only in case the CM Service Type indicates "Emergency Call Establishment".

MSISDN

See definition in subclause 5.6.2. The MSISDN is included in case of successful outcome of the service as an operator option, e.g. if it is needed at the MSC for charging purposes in case of call forwarding.

User error

One of the following error causes defined in subclause 5.6.1 shall be sent by the user if the access request fails, depending on the failure reason:

- unidentified subscriber;

- illegal subscriber;
  This error is sent if a correlated authentication procedure has not authenticated the subscriber.

- illegal equipment;
  This error is sent if an IMEI check failed, i.e. the IMEI is blacklisted or not white-listed.

- roaming not allowed;
  This cause is used after VLR restart if the subscriber has no subscription for the current location area, e.g. due to regional subscription. The cause will be qualified by "location area not allowed" or "national roaming not allowed", respectively.

- unknown location area;

- system failure;

- unexpected data value.

Provider error

For definition of provider errors see subclause 5.6.1.

6.4 Handover services

6.4.1 MAP_PREPARE_HANDOVER service

6.4.1.1 Definition

This service is used between MSC-A and MSC-B (E-interface) when a call is to be handed over from MSC-A to MSC-B.

The MAP_PREPARE_HANDOVER service is a confirmed service using the primitives from table 6.4/1.

6.4.1.2 Service primitives

Parameter name	Request	Indication	Response	Confirm
Invoke Id	M	M(=)	M(=)	M(=)
Target Cell Id	C	C(=)		
HO-NumberNotRequired	C	C(=)		
BSS-APDU	C	C(=)	C	C(=)
Handover Number			C	C(=)
User error			C	C(=)
Provider error				O

Table 6.4/1: MAP_PREPARE_HANDOVER

6.4.1.3 Parameter use

Invoke Id

For definition of this parameter see subclause 5.6.1.

Target Cell Id

For definition of this parameter see subclause 5.6.2. This parameter is only included if the service is not in an ongoing transaction.

HO-Number Not Required

For definition of this parameter see subclause 5.6.6.

BSS-APDU

For definition of this parameter see subclause 5.6.9.

Handover Number

For definition of this parameter see subclause 5.6.2. This parameter shall be returned, unless the parameter HO-NumberNotRequired is sent.

User error

For definition of this parameter see subclause 5.6.1. The following errors defined in subclause 5.6.1 may be used, depending on the nature of the fault:

-       No handover number available;

-       System failure;

-       Unexpected data value;

-       DataMissing.

Provider error

See definition of provider errors in subclause 5.6.1.

6.4.2   MAP_SEND_END_SIGNAL service

6.4.2.1   Definition

This service is used between MSC-B and MSC-A (E-interface) indicating that the radio path has been established by MSC-B to the MS. MSC-A retains then the main control of the call until it clears.

The response is used by MSC-A to inform MSC-B that all resources for the call can be released in MSC-B, either because the call has been released in MSC-A or because the call has been successfully handed over from MSC-B to another MSC.

The MAP_SEND_END_SIGNAL service is a confirmed service using the primitives from table 6.4/2.

6.4.2.2   Service primitives

Parameter name	Request	Indication	Response	Confirm
Invoke Id	M	M(=)	M(=)	M(=)
BSS-APDU	M	M(=)		
Provider error				O

Table 6.4/2: MAP_SEND_END_SIGNAL

6.4.2.3   Parameter use

Invoke Id

For definition of this parameter see subclause 5.6.1.

BSS-APDU

For definition of this parameter see subclause 5.6.9.

Provider error

For definition of this parameter see subclause 5.6.1.

6.4.3   MAP_PROCESS_ACCESS_SIGNALLING service

6.4.3.1   Definition

This service is used between MSC-B and MSC-A (E-interface) to pass information received on the A-interface in MSC-B to MSC-A.

The MAP_PROCESS_ACCESS_SIGNALLING service is a non-confirmed service using the primitives from table 6.4/3.

6.4.3.2   Service primitives

Parameter name	Request	Indication
Invoke Id	M	M(=)
BSS-APDU	M	M(=)

Table 6.4/3: MAP_PROCESS_ACCESS_SIGNALLING

6.4.3.3   Parameter use

Invoke Id

For definition of this parameter see subclause 5.6.1.

BSS-APDU

For definition of this parameter see subclause 5.6.9.

6.4.4   MAP_FORWARD_ACCESS_SIGNALLING service

6.4.4.1   Definition

This service is used between MSC-A and MSC-B (E-interface) to pass information to be forwarded to the A-interface of MSC-B.

The MAP_FORWARD_ACCESS_SIGNALLING service is a non-confirmed service using the primitives from table 6.4/4.

6.4.4.2   Service primitives

Parameter name	Request	Indication
Invoke Id	M	M(=)
BSS-APDU	M	M(=)

Table 6.4/4: MAP_FORWARD_ACCESS_SIGNALLING

6.4.4.3   Parameter use

For the definition and use of all parameters and errors, see subclause 5.6.1

Invoke Id

For definition of this parameter see subclause 5.6.1.

BSS-APDU

For definition of this parameter see subclause 5.6.9.

6.4.5   MAP_PREPARE_SUBSEQUENT_HANDOVER service

6.4.5.1   Definition

This service is used between MSC-B and MSC-A (E-interface) to inform MSC-A that it has been decided that a handover to either MSC-A or a third MSC (MSC-B') is required.

The MAP_PREPARE_SUBSEQUENT_HANDOVER service is a confirmed service using the primitives from table 6.4/5.

6.4.5.2 Service primitives

Parameter name	Request	Indication	Response	Confirm
Invoke Id	M	M(=)	M(=)	M(=)
Target Cell Id	M	M(=)		
Target MSC Number	M	M(=)		
BSS-APDU	M	M(=)	C	C(=)
User error			C	C(=)
Provider error				O

Table 6.4/5: MAP_PREPARE_SUBSEQUENT_HANDOVER

6.4.5.3 Parameter use

Invoke Id

For definition of this parameter see subclause 5.6.1.

Target Cell Id

For definition of this parameter see subclause 5.6.2.

Target MSC Number

For definition of this parameter see subclause 5.6.2.

BSS-APDU

For definition of this parameter see subclause 5.6.9.

User error

For definition of this parameter see subclause 5.6.1. The following error causes defined in subclause 5.6.1 may be used, depending on the nature of the fault:

- Unknown MSC;
- Subsequent handover failure;
- Unexpected data value;
- Data Missing.

Provider error

For definition of this parameter see subclause 5.6.1.

6.4.6 MAP_ALLOCATE_HANDOVER_NUMBER service

6.4.6.1 Definition

This service is used between MSC and VLR (B-interface) to request a handover number.

The MAP_ALLOCATE_HANDOVER_NUMBER service is a confirmed service using the primitives from table 6.4/6.

6.4.6.2 Service primitives

Parameter name	Request	Indication	Response	Confirm
Invoke Id	M	M(=)	M(=)	M(=)
User error			C	C(=)
Provider error				O

Table 6.4/6: MAP_ALLOCATE_HANDOVER_NUMBER

6.4.6.3   Parameter use

Invoke Id

For definition of this parameter see subclause 5.6.1.

User error

For definition of this parameter see subclause 5.6.1. The following errors defined in subclause 5.6.1 may be used, depending on the nature of the fault:

-        No handover number available.

Provider error

For definition of this parameter see subclause 5.6.1.

6.4.7   MAP_SEND_HANDOVER_REPORT service

6.4.7.1   Definition

This service is used between VLR and MSC-B (B-interface) to transfer the handover number to be forwarded to and used by MSC-A.

The MAP_SEND_HANDOVER_REPORT service is a confirmed service using the primitives from table 6.4/7.

6.4.7.2   Service primitives

Parameter name	Request	Indication	Response	Confirm
Invoke Id	M	M(=)	M(=)	M(=)
Handover Number	M	M(=)		Linked Id
M	M(=)		Provider error	
		O		

Table 6.4/7: MAP_SEND_HANDOVER_REPORT

6.4.7.3   Parameter use

Invoke Id

For definition of this parameter see subclause 5.6.1.

Handover Number

For definition of this parameter see subclause 5.6.2.

Linked Id

For   definition   of   this   parameter   see   subclause 5.6.1.   This   service   is   linked   with MAP_ALLOCATE_HANDOVER_NUMBER.

Provider error

For definition of this parameter see subclause 5.6.1.

6.5 Authentication management services

6.5.1 MAP_AUTHENTICATE service

6.5.1.1 Definition

This service is used between the VLR and the MSC when the VLR receives a MAP service indication from the MSC concerning a location registration, call set-up, operation on a supplementary service or a request from the MSC to initiate authentication.

The service is a confirmed service and consists of four service primitives.

6.5.1.2 Service primitives

The service primitives are shown in table 6.5/1

Parameter name	Request	Indication	Response	Confirm
Invoke id	M	M(=)	M(=)	M(=)
RAND	M	M(=)		
CKSN	M	M(=)		
SRES			M	M(=)
Provider error				O

Table 6.5/1: MAP_AUTHENTICATE parameters

6.5.1.3 Parameter use

Invoke id

See subclause 5.6.1 for the use of this parameter.

RAND

See subclause 5.6.7 for the use of this parameter.

CKSN

See subclause 5.6.7 for the use of this parameter.

SRES

See subclause 5.6.7 for the use of this parameter.

Provider error

See subclause 5.6.1 for the use of this parameter.

6.5.2 MAP_SEND_AUTHENTICATION_INFO service

6.5.2.1 Definition

This service is used between the VLR and the HLR for the VLR to retrieve authentication information from the HLR. The VLR requests some sets of RAND/SRES/Kc vectors.

If the HLR cannot provide the VLR with triplets, an empty response is returned. The VLR may then re-use old authentication triplets.

Security related network functions are further described in GSM 03.20.

The service is a confirmed service and consists of four service primitives.

6.5.2.2   Service primitives

The service primitives are shown in table 6.5/2.

Parameter name	Request	Indication	Response	Confirm
Invoke id	M	M(=)	M(=)	M(=)
IMSI	M	M(=)		
AuthenticationSetList			C	C(=)
User error			C	C(=)
Provider error				O

Table 6.5/2: MAP_SEND_AUTHENTICATION_PARAMETERS parameters

6.5.2.3   Parameter use

Invoke id

See subclause 5.6.1 for the use of this parameter.

IMSI

See subclause 5.6.2 for the use of this parameter.

AuthenticationSetList

A set of one to five authentication vectors are transferred from the HLR to the VLR, if the outcome of the service was successful.

User error

One of the following error causes defined in subclause 5.6.1 shall be sent by the user in case of unsuccessful outcome of the service, depending on the respective failure reason:

- unknown subscriber;
- unexpected data value;
- system failure;
- data missing.

Provider error

See subclause 5.6.1 for the use of this parameter.

6.6   Security management services

6.6.1   MAP_SET_CIPHERING_MODE service

6.6.1.1   Definitions

This service is used between the VLR and the MSC to set the ciphering mode and to start ciphering if applicable. It is called when another service requires that information is to be sent on the radio path in encrypted form.

The service is a non-confirmed service and consists of two service primitives.

6.6.1.2   Service primitives

The service primitives are shown in table 6.6/1

Parameter name	Request	Indication
Invoke id	M	M(=)
Ciphering mode	M	M(=)
Kc	C	C(=)

Table 6.6/1: MAP_SET_CIPHERING_MODE parameters

6.6.1.3   Parameter use

Invoke id

See subclause 5.6.1 for the use of this parameter.

Ciphering mode

See subclause 5.6.7 for the use of this parameter.

Kc

The Kc parameter should be included when the ciphering mode parameter indicates that ciphering must be performed.

6.7   International mobile equipment identities management services

6.7.1   MAP_CHECK_IMEI service

6.7.1.1   Definition

This service is used between the VLR and the MSC and between the MSC and the EIR to request check of IMEI. If the IMEI is not available in the MSC, it is requested from the MS and transferred to the EIR in the service request.

The service is a confirmed service and consists of four service primitives.

6.7.1.2   Service primitives

The service primitives are shown in table 6.7/1.

Parameter name	Request	Indication	Response	Confirm
Invoke id	M	M(=)	M(=)	M(=)
IMEI	C	C(=)	C	C(=)
Equipment status			C	C(=)
User error			C	C(=)
Provider error				O

Table 6.7/1: MAP_CHECK_IMEI parameters

6.7.1.3   Parameter use

<u>Invoke id</u>

See subclause 5.6.1 for the use of this parameter.

<u>IMEI</u>

See subclause 5.6.2 for the use of this parameter. The parameter shall not be included in the service request between the VLR and the MSC, but is mandatory in the service request from the MSC to the EIR. It is not included in the service response from the EIR to the MSC, but is mandatory in the service response from the MSC to the VLR on successful outcome.

<u>Equipment status</u>

See subclause 5.6.4 for the use of this parameter. This parameter is sent by the responder in case of successful outcome of the service.

<u>User error</u>

One of the following error causes defined in subclause 5.6.1 shall be sent by the user in case of unsuccessful outcome of the service, depending on the respective failure reason:

-        unknown equipment;
         This error is returned by the responder when the IMEI is not known in the EIR.

-        system failure;

-        unexpected data value.

<u>Provider error</u>

See subclause 5.6.1 for the use of this parameter.

6.7.2   MAP_OBTAIN_IMEI service

6.7.2.1   Definition

This service is used between the VLR and the MSC to request the IMEI. If the IMEI is not available in the MSC, it is requested from the MS.

The service is a confirmed service and consists of four service primitives.

6.7.2.2   Service primitives

The service primitives are shown in table 6.7/2.

Parameter name	Request	Indication	Response	Confirm
Invoke id	M	M(=)	M(=)	M(=)
IMEI			C	C(=)
User error			C	C(=)
Provider error				O

Table 6.7/2: MAP_OBTAIN_IMEI parameters

6.7.2.3 Parameter use

<u>Invoke id</u>

See subclause 5.6.1 for the use of this parameter.

<u>IMEI</u>

See subclause 5.6.2 for the use of this parameter. The parameter IS included in the service response from the MSC to the VLR on successful outcome of the service.

<u>User error</u>

If the service fails, the VLR sends the user error System Failure (see subclause 5.6.1) to the MSC.

<u>Provider error</u>

See subclause 5.6.1 for the use of this parameter.

6.8 Subscriber management services

6.8.1 MAP-INSERT-SUBSCRIBER-DATA service

6.8.1.1 Definition

This service is used by an HLR to update a VLR with certain subscriber data in the following occasions:

- the operator has changed the subscription of one or more supplementary services, basic services or data of a subscriber. Note that in case of withdrawal of a Basic or Supplementary service this primitive shall not be used;

- the operator has applied, changed or removed Operator Determined Barring;

- the subscriber has changed data concerning one or more supplementary services by using a subscriber procedure;

- the HLR provides the VLR with subscriber parameters at location updating of a subscriber or at restoration. In this case, this service is used to indicate explicitly that a supplementary service is not provisioned, if the supplementary service specification requires it. The only supplementary services which have this requirement are the CLIR and COLR services.

It is a confirmed service and consists of the primitives shown in table 6.8/1.

6.8.1.2 Service primitives

Parameter name	Request	Indication	Response	Confirm
Invoke Id	M	M(=)	M(=)	M(=)
IMSI	C	C(=)		
MSISDN	C	C(=)		
Category	C	C(=)		
Subscriber Status	C	C(=)		
Bearer service List	C	C(=)	C	C(=)
Teleservice List	C	C(=)	C	C(=)
Forwarding information List	C	C(=)		
Call barring information List	C	C(=)		
CUG information List	C	C(=)		
SS-Data List	C	C(=)		
Operator Determined Barring General data	C	C(=)	C	C(=)
Operator Determined Barring HPLMN data	C	C(=)		
Roaming Restriction Due To Unsupported Feature	C	C(=)		
Regional Subscription Data	C	C(=)		
SS-Code List			C	C(=)
Regional Subscription Response			C	C(=)
User error			U	C(=)
Provider error				O

Table 6.8/1: MAP-INSERT-SUBSCRIBER-DATA

6.8.1.3 Parameter use

All parameters are described in subclause 5.6. The following clarifications are applicable:

IMSI

It is only included if the service is not used in an ongoing transaction (e.g. location updating).

MSISDN

It is included either at location updating or when it is changed. The MSISDN sent shall be the basic MSISDN.

Category

It is included either at location updating or when it is changed.

Subscriber Status

It is included either at location updating or when it is changed.

To apply, remove or update Operator Determined Barring Categories the Subscriber Status is set to Operator Determined Barring. In this case ODB General Data shall also be present. If the Operator Determined Barring applies and the subscriber is registered in the HPLMN and HPLMN specific Operator Determined Barring applies then ODB HPLMN Specific Data shall also be present.

To remove all Operator Determined Barring Categories the Subscriber Status shall be set to "Service Granted".

Bearer service List

A list of Bearer service parameters (Bearer service is defined in subclause 5.6). A Bearer service parameter must be the code for an individual Bearer service, except in the cases described below.

The codes for the Bearer service groups "allAlternateSpeech-DataCDA" and "allAlternateSpeech-DataCDS" shall, if applicable, be sent from the HLR to the VLR as a pair. The codes for the Bearer service groups "allSpeechFollowedByDataCDA" and "allSpeechFollowedByDataCDS" shall, if applicable, be sent from the HLR to the VLR as a pair.

If it is included in the Request/Indication, it includes either all Bearer services subscribed (at location updating or at restoration) or only the ones changed (at subscriber data modification).

If the VLR receives an Indication containing any Bearer service parameters which it does not support/allocate it returns them in the response to the HLR and therefore discards the unsupported Bearer services (no error is sent back), except in the cases described below.

If the VLR receives the codes for the Bearer service groups "allSpeechFollowedByDataCDA" and "allSpeechFollowedByDataCDS" and supports one or more of the circuit-switched synchronous or asynchronous data rates specified for simple data bearer services, it shall accept the bearer service codes, and not return them in the response to the HLR. If the VLR does not support any of the circuit-switched synchronous or asynchronous data rates specified for simple data bearer services, and receives the pair of codes for "allAlternateSpeech-DataCDA" and "allAlternateSpeech-DataCDS" or the pair of codes for "allSpeechFollowedByDataCDA" and "allSpeechFollowedByDataCDS", it shall reject the pair of codes by returning them in the response to the HLR.

Teleservice List

A list of Teleservice parameters (Teleservice is defined in subclause 5.6). A Teleservice parameter must be the code for an individual Teleservice.

If it is included in the Request/Indication, it contains either all Teleservices subscribed (at location updating or at restoration) or the ones changed (at subscriber data modification).

If the VLR receives an Indication containing any Teleservice parameters which it does not support/allocate it returns them in the response to the HLR and therefore discards the unsupported Teleservices (no error is sent back).

Forwarding information List

A list of Forwarding information parameters (Forwarding information is defined in subclause 5.6). It includes Call Forwarding services either at location updating or at restoration or when they are changed. Each Forwarding information parameter shall be treated independently of all other parameters in the primitive.

The Forwarding information shall include the SS-Code for an individual call forwarding supplementary service. The Forwarding information shall contain one or more Forwarding Features (Forwarding Feature is defined in subclause 5.6).

The Forwarding Feature may include a Basic Service Group. This shall be interpreted according to the rules in subclause 6.8.1.4.

The Forwarding Feature shall contain an SS-Status parameter.

If the SS-Status indicates that call forwarding is registered then (except for call forwarding unconditional) the Forwarding Feature shall contain a forwarded-to number and, if available, the forwarded-to subaddress. In other states the forwarded-to number and, if applicable, the forwarded-to subaddress shall not be included. For call forwarding unconditional the forwarded-to number and, if applicable, the forwarded-to subaddress shall not be included. If the VLR does not receive a forwarded-to subaddress then it shall assume that a forwarded-to subaddress has not been registered.

The Forwarding Feature shall contain the forwarding options (except for call forwarding unconditional where the forwarding options shall not be included). Bits 3 and 4 of the forwarding options shall be ignored by the VLR, and may be set to any value by the HLR.

For call forwarding on no reply: If the SS-Status indicates that call forwarding is registered then the Forwarding Feature shall contain a no reply condition timer. In other states the no reply condition timer shall not be included.

For call forwarding services other than call forwarding on no reply: The Forwarding Feature shall not contain a no reply condition timer.

If the VLR receives an Indication containing any Call Forwarding service codes which it does not support/allocate it returns them to the HLR in the parameter SS-Code List and therefore discards the unsupported Call Forwarding service codes (no error is sent back).

Call barring information List

A list of Call barring information parameters (Call barring information is defined in subclause 5.6). It includes Call Barring services either at location updating or at restoration or when they are changed. Each Call barring information parameter shall be treated independently of all other parameters in the primitive.

The Call barring information shall include the SS-Code for an individual call barring supplementary service. The Call barring information shall contain one or more Call Barring Features (Call Barring Feature is defined in subclause 5.6).

The Call Barring Feature may include a Basic Service Group. This shall be interpreted according to the rules in subclause 6.8.1.4.

The Call Barring Feature shall contain an SS-Status parameter.

If the VLR receives an Indication containing any Call Barring service codes which it does not support/allocate it returns them to the HLR in the parameter SS-Code List and therefore discards the unsupported Call Barring service codes (no error is sent back).

CUG information List

A list of CUG information list parameters (CUG information is defined in subclause 5.6). It includes CUG information either at location updating or at restoration or when it is changed.

At location updating, restoration or when there is a change in CUG data, the HLR shall include the complete CUG-SubscriptionList and, if there are options per basic group, it shall also include the complete CUG-FeatureList. If there are not options per basic service group the CUG-FeatureList shall not be included.

When the VLR receives CUG data it shall replace the stored CUG data with the received data set.

If CUG-FeatureList is omitted in the Insert Subscriber Data operation VLR shall interpret that no options per basic service group exist, and then it shall apply the default values i.e. no outgoing access, no incoming access, no preferential CUG exists.

If CUG-Feature is received without preferential CUG, the VLR shall interpret that no preferential CUG applies.

If the VLR detects that there is overlapping in the information received within a dialogue, it shall send the error Unexpected Data Value.

Note that data consistency between CUG subscription data and CUG feature data is the responsibility of the HLR.

If the VLR does not support the CUG service it returns its code to the HLR in the parameter SS-Code List and therefore discards the received information (no error is sent back).

SS-Data List

A list of SS-Data parameters (SS-Data is defined in subclause 5.6). It is sent for any other supplementary service than Call Forwarding, Call Barring and CUG either at location updating or at restoration or when they are changed. Each SS-Data parameter shall be treated independently of all other parameters in the primitive.

The SS-Data shall include the SS-Code for an individual supplementary service.

The SS-Data shall contain an SS-Status parameter and any subscription options that are applicable to the service defined by the SS-Code.

The SS-Data may include a Basic Service Group List. This shall be interpreted according to the rules in subclause 6.8.1.4.

If the VLR receives an Indication containing any supplementary service codes which it does not support/allocate it returns them to the HLR in the parameter SS-Code List and therefore discards the unsupported service codes received (no error is sent back).

Operator Determined Barring General data

If it is included in a Request/Indication, it includes all the Operator Determined Barring categories that may be applied to a subscriber registered in any PLMN. This parameter is only included in a Request/Indication when the parameter Subscriber Status is set to the value Operator Determined Barring. Note that all General Operator Determined Barring Categories shall be set to their actual status.

If the VLR receives an Indication containing Operator Determined Barring General Data which shows that the subscriber is subject to barring not supported by the VLR, it returns Operator Determined Barring General Data in the response to the HLR to show the barring categories which are not supported by the VLR.

Operator Determined Barring HPLMN data

It includes all the Operator Determined Barring categories that may be applied only to a subscriber registered in the HPLMN. Therefore, it shall only be transferred to the VLR when the subscriber is roaming into the HPLMN and when the parameter Subscriber Status is set to the value Operator Determined Barring. Note that all HPLMN Operator Determined Barring Categories shall be set to their actual status.

If Subscriber Status is set to the value Operator Determined Barring and no Operator Determined Barring HPLMN data is present then the VLR shall not apply any HPLMN specific ODB services to the subscriber.

Roaming Restriction Due To Unsupported Feature

The HLR may decide to include this parameter in the request if certain services or features are indicated as not supported by the MSC/VLR (e.g. Advice of Charge Charging Level).

If this parameter is sent to the VLR the MSC area is restricted by the HLR and the VLR.

Regional Subscription Data

If included in the Insert Subscriber Data request this parameter defines the subscriber's subscription area for the addressed VLR (as defined in subclause 5.6). It contains the complete list of up to 10 Zone Codes that apply to a subscriber in the currently visited PLMN. The HLR shall send only those Zone Codes which are stored against the CC and NDC of the VLR to be updated.

> NOTE: Support of this parameter is a network operator option and it will not be sent to networks which do not support Regional Subscription.

Regional subscription data that have been stored previously in a subscriber data record in the VLR are completely replaced by the regional subscription data received in an Insert Subscriber Data indication during either an Update Location or Restore Data procedure or a stand alone Insert Subscriber data procedure.

After the regional subscription data are inserted the VLR shall derive whether its location areas are allowed or not. If the whole MSC area is restricted it will be reported to HLR by returning the Regional Subscription Response.

The VLR returns a Regional Subscription Response indicating that a problem with the Zone Code has been detected in one of the following cases:

- Too Many Zone Codes: more than 10 Zone Codes are to be stored in the VLR;

- Regional Subscription Not Supported by the VLR;

- Zone Codes Conflict: the VLR detects that the zone codes indicate conflicting service permission for a location area.

Zone codes which have no mapping to location areas shall be ignored.

If a sequence of MAP_INSERT_SUBSCRIBER_DATA services is used during a dialogue, Regional Subscription Data shall be accepted only in one service. Regional Subscription Data received in a subsequent service shall be rejected with the error Unexpected Data Value.

If Regional Subscription Data are not included in any MAP_INSERT_SUBSCRIBER_DATA service, there is no restriction of roaming due to Regional Subscription.

SS-Code List

The list of SS-Code parameters that are provided to a subscriber but are not supported/allocated by the VLR (SS-Code is defined in subclause 5.6). The list can only include individual SS-Codes that were sent in the service request.

Regional Subscription Response

If included in the response this parameter indicates one of:

- MSC Area Restricted entirely because of regional subscription;

- Too Many Zone Codes to be inserted;

- Zone Codes Conflict;

- Regional Subscription not Supported by the VLR.

If the VLR determines after insertion of Regional Subscription Data that the entire MSC area is restricted, the VLR shall respond with a Regional Subscription Response indicating MSC Area Restricted. Otherwise MSC Area Restricted is not sent. The HLR shall check whether the current MSC area is no longer restricted.

User error

Only one of the following values is applicable:

- Unidentified subscriber;

- Data missing;

- Unexpected data value.

6.8.1.4   Basic service information related to supplementary services

A number of parameters that relate to supplementary services can be qualified by a Basic Service Group (or a Basic Service Group List). This subclause explains how this information is to be interpreted. Supplementary service parameters to which this subclause is applicable only apply to the basic service groups described in this subclause, and only those basic service groups shall be overwritten at the VLR.

The Basic Service Group (or Basic Service Group List) is optional.

If present the Basic Service Group (or the elements of the Basic Service Group List) shall be one of:

-    an Elementary Basic Service Group for which the supplementary service is applicable to at least one basic service in the group; and to which the subscriber has a subscription to at least one basic service in the group;

-    the group "All Teleservices" provided that the service is applicable to at least one teleservice and that the subscriber has a subscription to at least one teleservice that is in the same Elementary Basic Service Group as a teleservice to which the service is applicable;

-    the group "All Bearer Services" provided that the service is applicable to at least one bearer service and that the subscriber has a subscription to at least one bearer service that is in the same Elementary Basic Service Group as a basic service to which the service is applicable.

If the Basic Service Group (or Basic Service Group List) is not present then the parameter shall apply to all Basic Service Groups.

If the basic service information is not a single Elementary Basic Service Group then the parameter shall be taken as applying individually to all the Elementary Basic Service Groups for which:

-    the supplementary service is applicable to at least one basic service in the Basic Service Group; and

-    the subscriber has a subscription to at least one basic service in the Basic Service Group.

The VLR is not required to store supplementary services data for Basic Service Groups that are not supported at the VLR.

6.8.2   MAP-DELETE-SUBSCRIBER-DATA service

6.8.2.1   Definition

This service is used by an HLR to remove certain subscriber data from a VLR if the subscription of one or more supplementary   services or basic services is withdrawn. Note that this service is not used in case of erasure or deactivation of supplementary services.

It is a confirmed service and consists of the primitives shown in table 6.8/2.

6.8.2.2   Service primitives

Parameter name	Request	Indication	Response	Confirm
Invoke Id	M	M(=)	M(=)	M(=)
IMSI	M	M(=)		
Basic service List	C	C(=)		
SS-Code List	C	C(=)		
Roaming Restriction Due To Unsupported Feature	C	C(=)		
Regional Subscription Data	C	C(=)		
Regional Subscription Response			C	C(=)
User error			C	C(=)
Provider error				O

Table 6.8/2: MAP-DELETE-SUBSCRIBER-DATA

6.8.2.3   Parameter use

All parameters are described in subclause 5.6. The following clarifications are applicable:

Basic service List

A list of Basic service parameters (Basic service is defined in subclause 5.6). It is used when one, several or all basic services are to be withdrawn from the subscriber.

SS-Code List

A list of SS-Code parameters (SS-Code is defined in subclause 5.6). It is used when several or all supplementary services are to be withdrawn from the subscriber.

There are three possible options:

-       deletion of basic service(s);
        The parameter Basic service List is only included.

-       deletion of supplementary service(s);
        The parameter SS-Code List is only included.

-       deletion of basic and supplementary services;
        Both Basic service List and SS-Code List are included.

Roaming Restriction Due To Unsupported Feature

This parameter is used if Roaming Restriction Due To Unsupported Feature is deleted from the subscriber data. This may occur if unsupported features or services are removed from the subscriber data in the HLR.

If this parameter is sent the VLR shall check if the current Location Area is possibly allowed now.

Regional Subscription Identifier

Contains one single Zone Code (as defined subclause 5.6) and is used if all Zone Codes shall be deleted from the subscriber data. When all the Zone Codes are deleted, the VLR shall check for its location areas whether they are allowed or not. If the whole MSC area is restricted, it will be reported to HLR by returning the Regional Subscription Response "MSC Area Restricted".

The binary coding of the Zone Code value received in a Delete Subscriber Data request shall not be checked by the VLR.

Note that support of this parameter is a network operator option and it shall not be sent to networks which do not support Regional Subscription.

If Regional Subscription is not supported by the VLR, the request for deletion of Zone Codes is refused by sending the Regional Subscription Response "Regional Subscription Not Supported" to the HLR.

If no Zone Codes are stored in the respective subscriber data record, the request for deleting all Zone Code information shall be ignored and no Regional Subscription Response shall be returned.

Regional Subscription Response

If included in the Delete Subscriber Data response this parameter indicates one of:

-       MSC Area Restricted;

-       Regional Subscription Not Supported.

User error

Only one of the following values is applicable:

- Unidentified subscriber;

- Data missing;

- Unexpected data value.

6.9 Identity management services

6.9.1 MAP-PROVIDE-IMSI service

6.9.1.1 Definition

This service is used by a VLR in order to get, via the MSC, the IMSI of a subscriber (e.g. when a subscriber has identified itself with a TMSI not allocated to any subscriber in the VLR).

It is a confirmed service and consists of the primitives shown in table 6.9/1.

6.9.1.2 Service primitives

Parameter name	Request	Indication	Response	Confirm
Invoke Id	M	M(=)	M(=)	M(=)
IMSI			C	C(=)
User error			C	C(=)
Provider error				O

Table 6.9/1: MAP-PROVIDE-IMSI

6.9.1.3 Parameter use

All parameters are described in subclause 5.6. The following clarifications are applicable:

IMSI

This parameter is received when the request is successfully carried out. It contains the requested IMSI.

User error

Only one of the following values is applicable:

- Absent subscriber.

6.9.2 MAP-FORWARD-NEW-TMSI service

6.9.2.1 Definition

This service is used by a VLR to allocate, via MSC, a new TMSI to a subscriber during an ongoing transaction (e.g. call set-up, location updating or supplementary services operation).

It is a confirmed service and consists of the primitives shown in table 6.9/2.

6.9.2.2   Service primitives

Parameter name	Request	Indication	Response	Confirm
Invoke Id	M	M(=)	M(=)	M(=)
TMSI	M	M(=)	Provider error	
		O		

Table 6.9/2: MAP-FORWARD-NEW-TMSI

6.9.2.3   Parameter use

The parameter TMSI is described in subclause 5.6.

6.10   Fault recovery services

6.10.1   MAP_RESET service

6.10.1.1   Definition

This service is used by the HLR, after a restart, to indicate to a list of VLRs that a failure occurred.

The MAP_RESET service is a non-confirmed service using the service primitives defined in table 6.10/1

6.10.1.2   Service primitives

Parameter name	Request	Indication
Invoke Id	M	M(=)
HLR number	M	M(=)
HLR Id LIST	U	C(=)

Table 6.10/1: MAP_RESET

6.10.1.3   Parameter definition and use

Invoke Id

See definition in subclause 5.6.1.

HLR number

See definition in subclause 5.6.2.

HLR Id LIST

The HLR Id List is a list of HLR Id. If the parameter is present in the indication, the VLR may base the retrieval of subscribers to be restored on their IMSI: the subscribers affected by the reset are those whose IMSI leading digits are equal to one of these numbers. If the parameter is absent, subscribers to be restored are those for which the OriginatingEntityNumber received at location updating time matches the equivalent parameter of the Reset Indication.

6.10.2 MAP_FORWARD_CHECK_SS_INDICATION service

6.10.2.1 Definition

This service may be used by an HLR as an implementation option, to indicate to a mobile subscriber that supplementary services parameters may have been altered, e.g. due to a restart. If received from the HLR, the VLR shall forward this indication to the MSC, which in turn forwards it to the MS. The HLR only sends this indication after successful completion of the subscriber data retrieval from HLR to VLR that ran embedded in a MAP_UPDATE_LOCATION procedure.

The MAP_FORWARD_CHECK_SS_INDICATION service is a non-confirmed service using the service primitives defined in table 6.10/2.

6.10.2.2 Service primitives

Parameter name	Request	Indication
Invoke Id	M	M(=)

Table 6.10/2: MAP_FORWARD_CHECK_SS_INDICATION

6.10.2.3 Parameter definition and use

Invoke Id

See definition in subclause 5.6.1.

6.10.3 MAP_RESTORE_DATA service

6.10.3.1 Definition

This service is invoked by the VLR on receipt of a MAP_PROVIDE_ROAMING_NUMBER indication for an unknown IMSI, or for a known IMSI with the indicator "Confirmed by HLR" set to "Not confirmed". The service is used to update the LMSI in the HLR, if provided, and to request the HLR to send all data to the VLR that are to be stored in the subscriber's IMSI record.

The MAP_RESTORE_DATA service is a confirmed service using the service primitives defined in table 6.10/3.

6.10.3.2 Service primitives

Parameter name	Request	Indication	Response	Confirm
Invoke Id	M	M(=)	M(=)	M(=)
IMSI	M	M(=)		
LMSI	U	C(=)		
HLR number			C	C(=)
MS Not Reachable Flag			C	C(=)
User error			C	C(=)
Provider error				O

Table 6.10/3: MAP_RESTORE_DATA

6.10.3.3 Parameter definitions and use

Invoke Id

See definition in subclause 5.6.1.

IMSI

See definition in subclause 5.6.2.

LMSI

See definition in subclause 5.6.2. It is an operator option to provide the LMSI from the VLR; it is mandatory for the HLR to support the LMSI handling procedures.

HLR number

See definition in subclause 5.6.2. The presence of this parameter is mandatory in case of successful outcome of the service.

MS Not Reachable Flag

See definition in subclause 5.6.8. This parameter shall be present in case of successful outcome of the service, if the "MS Not Reachable flag" was set in the HLR.

User error

In case of unsuccessful outcome of the service, an error cause shall be returned by the HLR. The following error causes defined in subclause 5.6.1 may be used, depending on the nature of the fault:

- unknown subscriber;

- system failure;

- unexpected data value;

- data missing.

Provider error

For definition of provider errors see subclause 5.6.1.

7 Operation and maintenance services

7.1 Subscriber tracing services

7.1.1 MAP-ACTIVATE-TRACE-MODE service

7.1.1.1 Definition

This service is used between the HLR and the VLR to activate subscriber tracing in the VLR.

The MAP-ACTIVATE-TRACE-MODE service is a confirmed service using the primitives from table 7.1/1.

7.1.1.2 Service primitives

Parameter name	Request	Indication	Response	Confirm
Invoke id	M	M(=)	M(=)	M(=)
IMSI	C	C(=)		
Trace reference	M	M(=)		
Trace type	M	M(=)		
OMC Id	U	C(=)		
User error			C	C(=)
Provider error				O

Table 7.1/1: MAP-ACTIVATE-TRACE-MODE

7.1.1.3 Parameter use

Invoke id

See definition in subclause 5.6.1.

IMSI

See definition in subclause 5.6.2. The IMSI is a mandatory parameter in a stand-alone operation.

Trace reference

See definition in subclause 5.6.10.

Trace type

See definition in subclause 5.6.10.

OMC Id

See definition in subclause 5.6.2. The use of this parameter is an operator option.

User error

The following errors defined in subclause 5.6.1 may be used, depending on the nature of the fault:

- Unidentified Subscriber;

- Facility Not Supported;

- Tracing Buffer Full;

- System Failure;

- Unexpected Data Value;

- Data missing**.**

<u>Provider error</u>

For definition of provider errors see subclause 5.6.1.

7.1.2 MAP-DEACTIVATE-TRACE-MODE service

7.1.2.1 Definition

This service is used between the VLR and the HLR for deactivating subscriber tracing in the VLR.

The MAP-DEACTIVATE-TRACE-MODE service is a confirmed service using the primitives from table 7.1/2.

7.1.2.2 Service primitives

Parameter name	Request	Indication	Response	Confirm
Invoke id	M	M(=)	M(=)	M(=)
IMSI	C	C(=)		
Trace reference	M	M(=)		
User error			C	C(=)
Provider error				O

Table 7.1/2: MAP-DEACTIVATE-TRACE-MODE

7.1.2.3 Parameter use

<u>Invoke id</u>

See definition in subclause 5.6.1.

<u>IMSI</u>

See definition in subclause 5.6.2. The IMSI is a mandatory parameter in a stand-alone operation.

<u>Trace reference</u>

See definition in subclause 5.6.10.

<u>User error</u>

The following errors defined in subclause 5.6.1 may be used, depending on the nature of the fault:

- Unidentified Subscriber;

- Facility Not Supported;

- System Failure;

- Unexpected Data Value;

- Data missing.

<u>Provider error</u>

For definition of provider errors see subclause 5.6.1.

7.1.3   MAP-TRACE-SUBSCRIBER-ACTIVITY service

7.1.3.1   Definition

This service is used between the VLR and the MSC to activate the subscriber tracing in the MSC.

The MAP-TRACE-SUBSCRIBER-ACTIVITY service is a non-confirmed service using the primitives from table 7.1/3.

7.1.3.2   Service primitives

Parameter name	Request	Indication
Invoke id	M	M(=)
IMSI	C	C(=)
Trace reference	M	M(=)
Trace type	M	M(=)
OMC Id	U	C(=)

Table 7.1/3: MAP-TRACE-SUBSCRIBER-ACTIVITY

7.1.3.3   Parameter use

Invoke id

See definition in subclause 5.6.1.

IMSI

See definition in subclause 5.6.2. The controlling MSC shall provide either the IMSI or the IMEI to the servicing MSC.

Trace reference

See definition in subclause 5.6.10.

Trace type

See definition in subclause 5.6.10.

OMC Id

See definition in subclause 5.6.2. The use of this parameter is an operator option.

7.2 Other operation and maintenance services

7.2.1 MAP-SEND-IMSI service

7.2.1.1 Definition

This service is used by a VLR in order to fetch the IMSI of a subscriber in case of some Operation & Maintenance procedure where subscriber data are needed in the Visited PLMN and MSISDN is the only subscriber's identity known.

It is a confirmed service and consists of the primitive shown in figure 7.2/1.

7.2.1.2 Service primitives

Parameter name	Request	Indication	Response	Confirm
Invoke Id	M	M(=)	M(=)	M(=)
MSISDN	M	M(=)		
IMSI			C	C(=)
User error			C	C(=)
Provider error				O

Table 7.2/1: MAP-SEND-IMSI

7.2.1.3 Parameter use

All parameters are described in subclause 5.6. The following clarifications are applicable:

User error

Only one of the following values is applicable:

- Unknown subscriber;

- Unexpected data value;

- Data missing.

8 Call handling services

8.1 MAP_SEND_INFO_FOR_INCOMING_CALL service

8.1.1 Definition

This service is used between an MSC and VLR. It is invoked by an MSC receiving an incoming call (call to MS) to request the VLR for information to proceed and set up the call.

This is a confirmed service using the primitives described in table 8.1/1.

8.1.2 Service primitives

Parameter name	Request	Indication	Response	Confirm
Invoke ID	M	M(=)	M(=)	M(=)
MSRN	M	C(=)		
Bearer Service	C	C(=)		
Tele Service	C	C(=)		
Dialled Number	C	C(=)		
CUG Interlock	C	C(=)	U	C(=)
CUG Outgoing Access	C	C(=)	C	C(=)
Number of Forwarding	C	C(=)		
Incoming Call Setup Result			C	C(=)
Forwarding Data			C	C(=)
MSISDN			U	C(=)
User error			C	C(=)
Provider error				O

Table 8.1/1: MAP_SEND_INFO_FOR_INCOMING_CALL parameters

8.1.3 Parameter use

See subclause 5.6 for a definition of the parameters used, in addition to the following.

MSRN

This is the Mobile Subscriber Roaming Number. It is the identity of the MS to receive the incoming call.

Bearer Service

If this information is present in the User Service Information element of the ISUP Initial Address Message, it will be passed on to the VLR without change. Note, either a Bearer Service or Tele Service will be provided, not both.

Tele Service

If this information is present in the User Service Information element of the ISUP Initial Address Message it will be passed on to the VLR without change.

Dialled Number

If the information is available, this is the number originally dialled by the calling subscriber.

CUG Interlock

This parameter will only be present in the request and indication if the CUG information is received from ISUP. This parameter will only be present in the response and confirm, if the Incoming Call Setup Result indicates forward and the call is to be treated as a CUG Call, following the CUG Call Authorisation check in the VLR.

CUG Outgoing Access

This parameter may only be present in the request and indication if the CUG Interlock Code is also provided. This information is received from ISUP. This parameter will only be present in the response and confirm, if the Incoming Call Setup Result indicates forward and the call is to be treated as a CUG Call, following the CUG Call Authorisation check in the VLR.

Number of Forwarding

If the call has already been forwarded this parameter will contain a count of the number of times the call has been forwarded, if it is provided by ISUP.

Incoming Call Setup Result

This parameter if present will indicate either success or forward. If the result indicates forward then the Forwarding Data and optionally MSISDN must also be present.

Forwarding Data

This is the address of the subscriber to whom the call should be connected. The parameter will contain the forwarded-to-number and if required the forwarding options.

MSISDN

This is the mobile subscriber ISDN identity of the called party. This is an operator dependant inclusion.

User error

This parameter is sent by the responder when an error is detected and if present, takes one of the following values:

- Unallocated Roaming Number;
  The MS has been identified by an MSRN, and this MSRN is not allocated.

- Unknown Subscriber;

- Absent Subscriber;
  This indicates that there is no call forwarding and either the IMSI detached flag was set in the VLR or paging was unsuccessful.

- Impossible Call Completion;
  This indicates that the call will not be forwarded and shall be released, e.g. call forwarding on busy subscriber and call waiting services are not active.

- Forwarding Violation;
  A call forwarding condition applies but the incoming call has already been forwarded as many times as allowed.

- CUG Reject;
  The value of this error cause will indicate the reason for CUG reject.

- Unexpected Data Value;

- Data Missing;

- System Failure.

See subclause 5.6 for a definition of these reasons.

Provider error

These are defined in subclause 5.6.

8.2 MAP_SEND_INFO_FOR_OUTGOING_CALL service

8.2.1 Definition

This service is used between an MSC and VLR. The service is invoked by an MSC to retrieve the required information from a VLR to enable an outgoing call set up request (call from an MS) to be actioned.

This is a confirmed service using the primitives defined in table 8.2/1.

8.2.2 Service primitives

Parameter name	Request	Indication	Response	Confirm
Invoke Id	M	M(=)	M(=)	M(=)
Called Number	M	M(=)		
Bearer Service	C	C(=)		
Tele Service	C	C(=)		
CUG Index	C	C(=)		
Suppress Pref CUG	C	C(=)		
Suppress CUG OA	C	C(=)		
User error			C	C(=)
Provider error				O

Table 8.2/1: MAP_SEND_INFO_FOR_OUTGOING_CALL parameters

8.2.3 Parameter use

See subclause 5.6 for a definition of the parameters used, in addition to the following.

Bearer Service

If this information is received from the MS, it is passed on unchanged to the VLR. Note, either a Bearer Service or Teleservice will be provided, not both.

Tele Service

If this information is received from the MS, it is passed on unchanged to the VLR.

CUG Index

This parameter will only be present if it is received from the MS, it is passed unchanged to the VLR.

Suppress Pref CUG

This parameter will only be present if it is received from the MS, it is passed unchanged to the VLR.

Suppress CUG OA

This parameter will only be present if it is received from the MS, it is passed unchanged to the VLR.

<u>User error</u>

This parameter is sent by the responder when an error is detected and if present, takes one of the following values:

- Unknown Subscriber;

- Bearer Service Not Provisioned;

- Teleservice Not Provisioned;
  A subscription check has been performed and the call has not passed that check due to incompatibility with regard to the requested service. Depending on the nature of the incompatibility, either of these messages will be returned.

- Call Barred;
  This error will indicate that either outgoing calls are barred for this MS or that calls are barred due to Operator Determined Barring (see GSM 02.41 for a definition of this network feature).

- CUG Reject;
  The value of this error cause will indicate the reason for CUG reject.

- Data Missing;

- Unexpected Data Value;

- System Failure.

See subclause 5.6 for a definition of these reasons.

<u>Provider error</u>

These are defined in subclause 5.6.

8.3   MAP_SEND_ROUTING_INFORMATION service

8.3.1   Definition

This service is used between the Gateway MSC and the HLR. The service is invoked by the Gateway MSC to perform the interrogation of the HLR in order to route a call towards the called MS.

This is a confirmed service using the primitives defined in table 8.3/1.

8.3.2   Service primitives

Parameter name	Request	Indication	Response	Confirm
Invoke Id	M	M(=)	M(=)	M(=)
MSISDN	M	M(=)		
CUG Interlock	C	C(=)	C	C(=)
CUG Outgoing Access	C	C(=)	C	C(=)
Number of Forwarding	C	C(=)		
Network Signal Info	C	C(=)		
IMSI			C	C(=)
MSRN			C	C(=)
Forwarding Data			C	C(=)
User error			C	C(=)
Provider error				O

Table 8.3/1: MAP_SEND_ROUTING_INFORMATION parameters

8.3.3   Parameter use

See subclause 5.6 for a definition of the parameters used in addition to the following.

MSISDN

This is the Mobile Subscriber ISDN number assigned to the called subscriber.

CUG Interlock

If this information is received from ISUP it is transparently passed to the HLR in the request and indication primitives. This parameter will only be present in the response and confirm primitive if the call is to be treated as a CUG Call, following the CUG Call Authorisation check in the HLR.

CUG Outgoing Access

This parameter may only be present if the CUG Interlock code is also provided. This information is received from ISUP and is transparently passed to the HLR in the request and indication primitives. This parameter will only be present in the response and confirm primitive if the call is to be treated as a CUG Call, following the CUG Call Authorisation check in the HLR.

Number of Forwarding

If this information is received from ISUP it is transparently passed to the HLR.

Network Signal Info

This parameter is defined in subclause 5.6.

IMSI

This is the identity of the subscriber to receive the call. If present this indicates success.

MSRN

This parameter may be returned if there are no call forwarding supplementary services active and the information is available. This parameter will not be present if the parameter Forwarding Data is returned.

Forwarding Data

This parameter will be returned if a call forwarding supplementary service is active. The parameter will not be present if the MSRN is returned.

User error

This parameter is sent by the responder when an error is detected and if present, takes one of the following values:

- Unknown Subscriber;

- Number changed;

- Call Barred;
  This error will indicate that either incoming calls are barred for this MS or that calls are barred due to Operator Determined Barring (see GSM 02.41 for a definition of this network feature).

- CUG Reject;
  The value of this error cause will indicate the reason for CUG Reject.

- Bearer Service Not Provisioned;

- Teleservice Not Provisioned;
  A subscription check has been performed and the call has not passed the check due to incompatibility with regard to the requested service. Depending on the nature of the incompatibility, either of these messages will be returned.

- Facility Not Supported;

- Absent Subscriber;
  This indicates that the location of the MS is not known (either the station is not registered and there is no location information available or the Provide Roaming Number procedure fails due to IMSI detached flag being set) and the call forwarding on MS not reachable service is not active.

- Forwarding Violation;

- System Failure;

- Data Missing;

- Unexpected Data Value.

See subclause 5.6 for a definition of these errors.

<u>Provider error</u>

These are defined in subclause 5.6.

8.4   MAP_PROVIDE_ROAMING_NUMBER service

8.4.1   Definition

This service is used between the HLR and VLR. The service is invoked by the HLR to request a VLR to send back a roaming number to enable the HLR to instruct the GMSC to route an incoming call to the called MS.

This is a confirmed service which uses the Primitives described in table 8.4/1.

8.4.2   Service primitives

Parameter name	Request	Indication	Response	Confirm
Invoke Id	M	M(=)	M(=)	M(=)
IMSI	M	M(=)		
MSC Number	M	M(=)		
MSISDN	U	C(=)		
LMSI	C	C(=)		
GSM Bearer Capability	C	C(=)		
Network Signal Info	C	C(=)		
Roaming Number			C	C(=)
User error			C	C(=)
Provider error				O

Table 8.4/1: MAP_PROVIDE_ROAMING_NUMBER parameters

8.4.3   Parameter use

See subclause 5.6 for a definition of the parameters used, in addition to the following.

IMSI

This is the IMSI of the called Subscriber.

MSC Number

This is the ISDN number assigned to the MSC currently serving the MS. The MSC number will have been stored in the HLR as provided at location updating.

MSISDN

If present this is the Mobile Subscriber ISDN identity of the called subscriber. This is the MSISDN contained in the MAP-SEND-ROUTING-INFORMATION request.

LMSI

The Local MS Identity shall be provided if the VLR had previously updated the HLR with this reference.

GSM Bearer Capability

This information is passed according to the rules specified in GSM 09.07.

There may be two GSM Bearer Capabilities supplied.

Network Signal Info

This parameter is defined in subclause 5.6.

The information is taken from the Network Signal Info parameter of the MAP_SEND_ROUTING_INFORMATION primitive indication (see subclause 8.3.3).

Roaming Number

If present this parameter indicates success.

User error

This parameter is sent by the responder when an error is detected and if present, takes one of the following values:

- Absent Subscriber;
  This error will be returned if the IMSI detach flag is set.

- No Roaming Number Available;

- Facility Not Supported;

- System Failure;

- Data Missing;

- Unexpected Data Value.

See subclause 5.6 for a definition of these reasons.

Provider error

These are defined in subclause 5.6.

8.5 MAP_COMPLETE_CALL service

8.5.1 Definition

This service is used between the VLR and MSC. The service is invoked by a VLR to request the MSC to set up a call to an MS in the case of an incoming call, and from an MS for an outgoing call.

This is a confirmed service which uses the Primitives described in table 8.5/1.

8.5.2 Service primitives

Parameter name	Request	Indication	Response	Confirm
Invoke Id	M	M(=)	M(=)	M(=)
MSISDN	C	C(=)		
IMEI	C	C(=)		
Category	C	C(=)		
CUG Index	C	C(=)		
GSM Bearer Capability	C	C(=)		
Network Signal Info	C	C		
No Reply Condition Time	C	C(=)		
SS-Data list	C	C(=)		
CUG Interlock	C	C(=)		
CUG Outgoing Access	C	C(=)		
User error			C	C(=)
Provider error				O

Table 8.5/1: MAP_COMPLETE_CALL parameters

8.5.3 Parameter use

See subclause 5.6 for a definition of the parameters used, in addition to the following.

MSISDN

This is the Mobile Subscriber ISDN identity assigned to the calling subscriber for mobile originating calls or called subscriber for mobile terminating calls. For outgoing Emergency Calls only, this parameter may not be present.

IMEI

This is the International Mobile Equipment Identity of the MS. This parameter will only be present for outgoing Emergency Calls when the calling subscriber cannot be identified. Use of the IMEI in these circumstances is an operator option.

Category

This is the category of the calling subscriber for outgoing call setup only. This information will always be present for an outgoing call.

CUG Index

In the case of an Incoming Call, this parameter will only be present, if CUG information has been received in the incoming call setup request. The CUG Index will be passed to the called subscriber in the call setup signalling. This information may also be present for an outgoing call, this indicates that a CUG has been implicitly invoked. As an operator option, this information may be returned to the calling user but will not be passed to the call destination (reference GSM 03.85).

GSM Bearer Capability

This is the Bearer Capability received by the VLR in the MAP-PROVIDE-ROAMING-NUMBER service. Two GSM Bearer Capabilities may be supplied. This parameter will only be present for incoming call setup and if the information is available.

Network Signal Info

This will only be present if the information is available and GSM BC is not provided. This is the Network Signal Info received by the VLR in the MAP-PROVIDE-ROAMING-NUMBER service. This parameter is only applicable for incoming call setup.

No Reply Condition Time

This parameter is only included for incoming call setup and if the call forwarding on no reply service is active.

SS-Data List

This is a list of SS-Data and will only be passed if the information is available (i.e. Phase 2 supplementary services such as number identification services, CUG and Advice of Charge).

CUG Interlock

This parameter will only be present for outgoing call setup. This is the result of the conversion by the VLR, of the CUG Index originally supplied by the MS.

CUG Outgoing Access

This parameter may be present but only if the CUG Interlock code is also provided.

User error

This parameter is sent by the responder when an error is detected and if present, takes one of the following values:

- No Subscriber Reply;
- Radio Congestion;
- Busy Subscriber;
- Data Missing;
- Unexpected Data Value.

See subclause 5.6 for a definition of these reasons.

Provider error

These are defined in subclause 5.6.

8.6 **MAP_PROCESS_CALL_WAITING service**

8.6.1 **Definition**

This service is used between the VLR and MSC. It is invoked by the VLR to request the MSC to invoke the Call Waiting service.

This is a confirmed service using the primitives described in table 8.6/1.

8.6.2 **Service primitives**

Parameter name	Request	Indication	Response	Confirm
Invoke Id	M	M(=)	M(=)	M(=)
MSISDN	M	M(=)		
NoReply Condition Time	C	C(=)		
SS-Data List	C	C(=)		
CUG Index	C	C(=)		
GSM Bearer Capability	C	C(=)		
Network Signal Info	C	C(=)		
User error			C	C(=)
Provider error				O

Table 8.6/1: MAP_PROCESS_CALL_WAITING parameters

8.6.3 **Parameter use**

See subclause 5.6 for a definition of the parameters used, in addition to the following.

MSISDN

This is the connected called line identification.

No Reply Condition Time

This is the No Reply Condition Timer value. The presence of this parameter indicates whether or not the service Call Forwarding on No Reply is active.

SS-Data List

This is a list of SS-Data and will only be passed if the information is available (i.e. for GSM Phase 2 supplementary services such as number presentation services and Advice of Charge).

CUG Index

This parameter will only be present if the received CUG information in the incoming call setup request from the MSC. It is the result of the translation by the VLR, of the CUG Interlock code.

GSM Bearer Capability

This is the Bearer Capability received by the VLR in the MAP-PROVIDE-ROAMING-NUMBER service. It will only be present if the information is available.

Network Signal Info

This will only be present if the information is available and GSM BC is not provided. This is the Network Signal Info received by the VLR in the MAP-PROVIDE-ROAMING-NUMBER service.

User error

This parameter is sent by the responder when an error is detected, if present, takes one of the following values:

- Unexpected data value;

- Busy Subscriber;

- No Subscriber Reply.

Provider error

These are defined in subclause 5.6.

9 Supplementary services related services

9.1 MAP_REGISTER_SS service

9.1.1 Definition

This service is used between the MSC and the VLR and between the VLR and the HLR to register data related to a supplementary service. The VLR will relay the message to the HLR.

The service is a confirmed service and consists of four service primitives.

9.1.2 Service primitives

The service primitives are shown in table 9.1/1.

Parameter name	Request	Indication	Response	Confirm
Invoke id	M	M(=)	M(=)	M(=)
SS-Code	M	M(=)		
Basic service	C	C(=)		
Forwarded-to number with subaddress	C	C(=)		
No reply condition time	C	C(=)		
Forwarding information			C	C(=)
User error			C	C(=)
Provider error				O

Table 9.1/1: MAP_REGISTER_SS parameters

9.1.3 Parameter use

Invoke id

See subclause 5.6.1 for the use of this parameter.

SS-Code

This parameter indicates the supplementary service which the mobile subscriber wants to register.

Basic service

This parameter indicates for which basic service group the supplementary service is to be registered. If it is not included, the registration request applies to all basic services.

Forwarded-to number with subaddress

This parameter is obligatory if the registration applies to one or more call forwarding supplementary services. It can optionally include a sub-address.

No reply condition time

This parameter is included if the registration applies to the Call Forwarding on No Reply supplementary service (or a superset of this service) and the mobile subscriber supplies a value for this time.

Forwarding information

This parameter is returned by the responder at successful outcome of the service, if the registration request concerned one or a group of Call Forwarding supplementary services.

User error

This parameter is sent by the responder upon unsuccessful outcome of the service, and then takes one of the following values defined in subclause 5.6.1:

- System failure;

- Data missing;

- Unexpected data value;

- Call Barred;

- Bearer service not provisioned;
  This error is returned only if not even a subset of the requested bearer service group has been subscribed to.

- Teleservice not provisioned;
  This error is returned only if not even a subset of the requested teleservice group has been subscribed to.

- Illegal SS operation;

- SS error status;

- SS incompatibility.

Provider error

See subclause 5.6.1 for the use of this parameter.

9.2 MAP_ERASE_SS service

9.2.1 Definition

This service is used between the MSC and the VLR and between the VLR and the HLR to erase data related to a supplementary service. The VLR will relay the message to the HLR.

The service is a confirmed service and consists of four service primitives.

9.2.2 Service primitives

The service primitives are shown in table 9.2/1.

Parameter name	Request	Indication	Response	Confirm
Invoke id	M	M(=)	M(=)	M(=)
SS-Code	M	M(=)		
Basic service	C	C(=)		
Forwarding information			C	C(=)
User error			C	C(=)
Provider error				O

Table 9.2/1: MAP_ERASE_SS parameters

9.2.3 Parameter use

Invoke id

See subclause 5.6.1 for the use of this parameter.

SS-Code

This parameter indicates the supplementary service which the mobile subscriber wants to erase.

Basic service

This parameter indicates for which basic service group the supplementary service should be erased. If it is not included, the erasure request applies to all basic services.

Forwarding information

This parameter is returned by the responder at successful outcome of the service, if the erasure request concerned one or a group of Call Forwarding supplementary services.

<u>User error</u>

This parameter is sent by the responder upon unsuccessful outcome of the service, and then takes one of the following values, defined in subclause 5.6.1:

- System failure;

- Data Missing;

- Unexpected data value;

- Bearer service not provisioned;
  This error is returned only if not even a subset of the requested bearer service group has been subscribed to.

- Teleservice not provisioned;
  This error is returned only if not even a subset of the requested teleservice group has been subscribed to.

- Call Barred;

- Illegal SS operation;

- SS error status.

<u>Provider error</u>

See subclause 5.6.1 for the use of this parameter.

9.3 MAP_ACTIVATE_SS service

9.3.1 Definition

This service is used between the MSC and the VLR and between the VLR and the HLR to activate a supplementary service. The VLR will relay the message to the HLR.

The service is a confirmed service and consists of four service primitives.

9.3.2 Service primitives

The service primitives are shown in table 9.3/1.

Parameter name	Request	Indication	Response	Confirm
Invoke id	M	M(=)	M(=)	M(=)
SS-Code	M	M(=)		
Basic service	C	C(=)		
Forwarding information			C	C(=)
Call barring information			C	C(=)
SS-Data			C	C(=)
User error			C	C(=)
Provider error				O

Table 9.3/1: MAP_ACTIVATE_SS parameters

9.3.3 Parameter use

Invoke id

See subclause 5.6.1 for the use of this parameter.

SS-Code

This parameter indicates the supplementary service which the mobile subscriber wants to activate.

Basic service

This parameter indicates for which basic service groups the requested supplementary service(s) should be activated. If it is not included, the activation request applies to all basic services.

Forwarding information

This parameter is returned by the responder at successful outcome of the service, if the activation request concerned Call Forwarding.

Call barring information

This parameter is returned by the responder at successful outcome of the service, if the activation request concerned Call Barring.

SS-Data

This parameter is returned by the responder at successful outcome of the service, if the activation request concerned for example Call Waiting.

User error

This parameter is sent by the responder upon unsuccessful outcome of the service, and then takes one of the following values, defined in subclause 5.6.1:

- System failure;

- Data Missing;

- Unexpected data value;

- Bearer service not provisioned;
  This error is returned only if not even a subset of the requested bearer service group has been subscribed to.

- Teleservice not provisioned;
  This error is returned only if not even a subset of the requested teleservice group has been subscribed to.

- Call Barred;

- Illegal SS operation;

- SS error status;

- SS subscription violation;

- SS incompatibility;

- Negative PW check;

- Number Of PW Attempts Violation.

Provider error

See subclause 5.6.1 for the use of this parameter.

9.4 MAP_DEACTIVATE_SS service

9.4.1 Definitions

This service is used between the MSC and the VLR and between the VLR and the HLR to deactivate a supplementary service. The VLR will relay the message to the HLR.

The service is a confirmed service and consists of four service primitives.

9.4.2 Service primitives

The service primitives are shown in table 9.4/1.

Parameter name	Request	Indication	Response	Confirm
Invoke id	M	M(=)	M(=)	M(=)
SS-Code	M	M(=)		
Basic service	C	C(=)		
Forwarding information			C	C(=)
Call barring information			C	C(=)
SS-Data			C	C(=)
User error			C	C(=)
Provider error				O

Table 9.4/1: MAP_DEACTIVATE_SS parameters

9.4.3 Parameter use

Invoke id

See subclause 5.6.1 for the use of this parameter.

SS-Code

This parameter indicates the supplementary service which the mobile subscriber wants to deactivate.

Basic service

This parameter indicates for which basic service group the requested supplementary service(s) should be deactivated. If it is not included the deactivation request applies to all basic services.

Forwarding information

This parameter is returned by the responder at successful outcome of the service, if the deactivation request concerned one or a group of Call Forwarding supplementary services.

Call barring information

This parameter is returned by the responder at successful outcome of the service, if the activation request concerned one or a group of Call Barring supplementary services.

SS-Data

This parameter is returned by the responder at successful outcome of the service, for example if the deactivation request concerned the Call Waiting supplementary service.

User error

This parameter is sent by the responder upon unsuccessful outcome of the service, and then takes one of the following values, defined in subclause 5.6.1:

- System failure;

- Data Missing;

- Unexpected data value;

- Bearer service not provisioned;
  This error is returned only if not even a subset of the requested bearer service group has been subscribed to.

- Teleservice not provisioned;
  This error is returned only if not even a subset of the requested teleservice group has been subscribed to.

- Call Barred;

- Illegal SS operation;

- SS error status;

- SS subscription violation;

- Negative PW check;

- Number Of PW Attempts Violation.

Provider error

See subclause 5.6.1 for the use of this parameter.

9.5 MAP_INTERROGATE_SS service

9.5.1 Definitions

This service is used between the MSC and the VLR and between the VLR and the HLR to retrieve information related to a supplementary service. The VLR will relay the message to the HLR if necessary.

The service is a confirmed service and consists of four service primitives.

9.5.2 Service primitives

The service primitives are shown in table 9.5/1.

Parameter name	Request	Indication	Response	Confirm
Invoke id	M	M(=)	M(=)	M(=)
SS-Code	M	M(=)		
Basic service	C	C(=)		
SS-Status			C	C(=)
Basic service Group LIST			C	C(=)
Forwarding feature LIST			C	C(=)
CLI restriction Info			C	C(=)
User error			C	C(=)
Provider error				O

Table 9.5/1: MAP_INTERROGATE_SS parameters

9.5.3 Parameter use

For additional information on parameter use refer to the GSM 04.8x and 04.9x-series of technical specifications.

Invoke id

See subclause 5.6.1 for the use of this parameter.

SS-Code

The mobile subscriber can only interrogate a single supplementary service per service request.

Basic service

This parameter indicates for which basic service group the given supplementary service is interrogated. If it is not included, the interrogation request applies to all basic services.

SS-Status

This parameter is included by the responder if:

- the interrogated supplementary service can only be subscribed for all applicable basic services simultaneously; or

- the interrogated supplementary service is not active for any of the interrogated basic services.

Basic service group LIST

This parameter LIST is used to include one or a series of basic service groups for which the interrogated supplementary service is active. If the interrogated supplementary service is not active for any of the interrogated (and provisioned) basic service groups, the SS-Status parameter is returned.

Forwarding feature LIST

The forwarding feature parameter is described in subclause 5.6.4. A list of one or more forwarding features is returned by the responder when the interrogation request applied to Call Forwarding supplementary service.

If no basic service code parameter is provided within this sequence, the forwarding feature parameter applies to all provisioned basic services.

CLI restriction Info

The CLI-RestrictionInfo parameter is returned by the responder when the interrogation request applies to the CLIR supplementary service.

User error

This error is sent by the responder upon unsuccessful outcome of the interrogation service, and then takes one of the following values, defined in subclause 5.6.1:

- System failure;

- Data Missing;

- Unexpected data value;

- Bearer Service not provisioned;
  This error is returned only if not even a subset of the interrogated bearer services are provided.

- Teleservice not provisioned;
  This error is returned only if not even a subset of the interrogated teleservices are provided.

- Call Barred;

- Illegal SS operation;

- SS not available.

Provider error

See subclause 5.6.1 for the use of this parameter.

9.6 MAP_INVOKE_SS service

9.6.1 Definitions

This service is used between the MSC and the VLR to check the subscriber's subscription to a given supplementary service in the VLR, in connection with in-call invocation of that supplementary service, i.e. after the call set-up phase is finished. For supplementary service invocation during call set-up phase, please refer to the call handling descriptions.

The service is a confirmed service and consists of four service primitives.

9.6.2 Service primitives

The service primitives are shown in table 9.6/1.

Parameter name	Request	Indication	Response	Confirm
Invoke id	M	M(=)	M(=)	M(=)
SS-Code	M	M(=)		
Basic service	C	C(=)		
User error			C	C(=)
Provider error				O

Table 9.6/1: MAP_INVOKE_SS parameters

9.6.3 Parameter use

Invoke id

See subclause 5.6.1 for the use of this parameter.

SS-Code

This SS-Code can only refer to a single supplementary service, e.g. the Call Hold or Multi Party supplementary services.

Basic service

This parameter indicates for which basic service the supplementary service invocation is required.

User error

This parameter is sent by the responder upon unsuccessful outcome of the service, and then takes one of the following values:

- System Failure;
- Data Missing;
- Unexpected data value;
- Call Barred;
- Illegal SS operation;
- SS error status;
- SS not available.

Provider error

See subclause 5.6.1 for the use of this parameter.

9.7 MAP_REGISTER_PASSWORD service

9.7.1 Definitions

This service is used between the MSC and the VLR and between the VLR and the HLR if the mobile subscriber requests to register a new password. The VLR will relay the message to the HLR.

The service is a confirmed service and consists of four service primitives.

9.7.2 Service primitives

The service primitives are shown in table 9.7/1.

Parameter name	Request	Indication	Response	Confirm
Invoke id	M	M(=)	M(=)	M(=)
SS-Code	M	M(=)		
New password			C	C(=)
User error			C	C(=)
Provider error				O

Table 9.7/1: MAP_REGISTER_PASSWORD parameters

9.7.3 Parameter use

Invoke id

See subclause 5.6.1 for the use of this parameter.

SS-Code

This parameter indicates for which supplementary service(s) the password should be registered.

New Password

See subclause 5.6.4 for the use of this parameter.

User error

This parameter is sent by the responder upon unsuccessful outcome of the service, and then takes one of the following values, defined in subclause 5.6.1:

- System failure;
- Data Missing;
- Unexpected data value;
- Call Barred;
- SS subscription violation;
- Password registration failure;
- Negative PW check;
- Number Of PW Attempts Violation.

Provider error

See subclause 5.6.1 for the use of this parameter.

9.8 MAP_GET_PASSWORD service

9.8.1 Definitions

This service is used between the HLR and the VLR and between the VLR and the MSC when the HLR receives a request from the mobile subscriber for an operation on a supplementary service which requires a password from the subscriber. The VLR will relay the message to the MSC.

The service is a confirmed service and consists of four service primitives.

9.8.2 Service primitives

The service primitives are shown in table 9.8/1.

Parameter name	Request	Indication	Response	Confirm
Invoke id	M	M(=)	M(=)	M(=)
Linked id	C	C(=)		
Guidance info	M	M(=)		
Current password			M	M(=)
Provider error				O

Table 9.8/1: MAP_GET_PASSWORD parameters

9.8.3 Parameter use

Invoke id

See subclause 5.6.1 for the use of this parameter.

Linked Id

See subclause 5.6.1 for the use of this parameter. If the MAP GET PASSWORD service is used in conjunction with the MAP REGISTER PASSWORD service, this parameter must be present; otherwise it must be absent.

Guidance info

See subclause 5.6.4 for the use of this parameter.

Current password

See subclause 5.6.4 for the use of this parameter.

Provider error

See subclause 5.6.1 for the use of this parameter.

9.9   MAP_PROCESS_UNSTRUCTURED_SS_REQUEST service

9.9.1   Definitions

This service is used between the MSC and the VLR and between the VLR and the HLR to relay information in order to allow unstructured supplementary service operation.

The MAP_PROCESS_UNSTRUCTURED_SS_REQUEST service is a confirmed service using the primitives from table 9.9/1.

9.9.2   Service primitives

Parameter name	Request	Indication	Response	Confirm
Invoke id	M	M(=)	M(=)	M(=)
USSD Data Coding Scheme	M	M(=)	C	C(=)
USSD String	M	M(=)	C	C(=)
User error			C	C(=)
Provider error				O

Table 9.9/1: MAP_PROCESS_UNSTRUCTURED_SS_REQUEST parameters

9.9.3   Parameter use

Invoke id

See subclause 5.6.1 for the use of this parameter.

USSD Data Coding Scheme:

See subclause 5.6.4 for the use of this parameter. The presence of the parameter in the response is dependent on the unstructured supplementary service application. If this parameter is present, then the USSD String parameter has to be present.

USSD String:

See subclause 5.6.1 for the use of this parameter. The presence of the parameter in the response is dependent on the unstructured supplementary service application. If this parameter is present, then the USSD Data Coding Scheme parameter has to be present.

User error

This parameter is sent by the responder upon unsuccessful outcome of the service, and then takes one of the following values defined in subclause 5.6.1:

- System failure;

- Data missing;

- Unexpected data value;
  This error is returned by the responder if it is not able to deal with the contents of the USSD string.

- Call Barred;

- Unknown Alphabet.

Provider error

See subclause 5.6.1 for the use of this parameter.

9.10 MAP_UNSTRUCTURED_SS_REQUEST service

9.10.1 Definitions

This service is used between the HLR and the VLR and between the VLR and the MSC when the invoking entity requires information from the mobile user, in connection with unstructured supplementary service handling.

The MAP_UNSTRUCTURED_SS_REQUEST service is a confirmed service using the primitives from table 9.10/1.

9.10.2 Service primitives

The service primitives are shown in table 9.10/1.

Parameter name	Request	Indication	Response	Confirm
Invoke id	M	M(=)	M(=)	M(=)
USSD Data Coding Scheme	M	M(=)	C	C(=)
USSD String	M	M(=)	C	C(=)
User error			C	C(=)
Provider error				O

Table 9.10/1: MAP_UNSTRUCTURED_SS_REQUEST parameters

9.10.3 Parameter use

Invoke id

See subclause 5.6.1 for the use of this parameter.

USSD Data Coding Scheme:

See subclause 5.6.4 for the use of this parameter. The presence of the parameter in the response is dependent on the mobile user's MMI input. If this parameter is present, then the USSD String parameter has to be present.

USSD String:

See subclause 5.6.1 for the use of this parameter. The presence of the parameter in the response is dependent on the mobile user's MMI input. If this parameter is present, then the USSD Data Coding Scheme parameter has to be present.

<u>User error</u>

This parameter is sent by the responder upon unsuccessful outcome of the service, and then takes one of the following values defined in subclause 5.6.1:

-        System failure;

-        Data missing;

-        Unexpected data value;
         This error is returned by the responder if it is not able to deal with the contents of the USSD string.

-        Absent Subscriber;

-        Illegal Subscriber;
         This error indicates that delivery of the unstructured supplementary service data failed because the MS failed authentication.

-        Illegal Equipment;

-        USSD Busy;

-        Unknown Alphabet.

<u>Provider error</u>

See subclause 5.6.1 for the use of this parameter.

9.11   MAP_UNSTRUCTURED_SS_NOTIFY service

9.11.1   Definitions

This service is used between the HLR and the VLR and between the VLR and the MSC when the invoking entity requires a notification to be sent to the mobile user, in connection with unstructured supplementary services handling.

The MAP_UNSTRUCTURED_SS_NOTIFY service is a confirmed service using the primitives from table 9.11/1.

9.11.2   Service primitives

The service primitives are shown in table 9.11/1.

Parameter name	Request	Indication	Response	Confirm
Invoke id	M	M(=)	M(=)	M(=)
USSD Data Coding Scheme	M	M(=)		
USSD String	M	M(=)		
User error			C	C(=)
Provider error				O

Table 9.11/1: MAP_UNSTRUCTURED_SS_NOTIFY parameters

9.11.3   Parameter use

Invoke id

See subclause 5.6.1 for the use of this parameter.

USSD Data Coding Scheme:

See subclause 5.6.4 for the use of this parameter.

USSD String:

See subclause 5.6.1 for the use of this parameter.

<u>User error</u>

This parameter is sent by the responder upon unsuccessful outcome of the service, and then takes one of the following values defined in subclause 5.6.1:

- System failure;

- Data missing;

- Unexpected data value;
  This error is returned by the responder if it is not able to deal with the contents of the USSD string.

- Absent Subscriber;

- Illegal Subscriber;
  This error indicates that delivery of the unstructured supplementary service data failed because the MS failed authentication.

- Illegal Equipment;

- USSD Busy;

- Unknown Alphabet.

<u>Provider error</u>

See subclause 5.6.1 for the use of this parameter.

10 Short message service management services

10.1 MAP-SEND-ROUTING-INFO-FOR-SM service

10.1.1 Definition

This service is used between the gateway MSC and the HLR to retrieve the routing information needed for routing the short message to the servicing MSC.

The MAP-SEND-ROUTING-INFO-FOR-SM is a confirmed service using the primitives from table 10.1/1.

10.1.2 Service primitives

The service primitives are shown in table 10.1/1.

Parameter name	Request	Indication	Response	Confirm
Invoke Id	M	M(=)	M(=)	M(=)
MSISDN	M	M(=)		
SM-RP-PRI	M	M(=)		
Service Centre Address	M	M(=)		
IMSI			C	C(=)
MSC Number			C	C(=)
LMSI			C	C(=)
User error			C	C(=)
Provider error				O

Table 10.1/1: MAP-SEND-ROUTING-INFO-FOR-SM

10.1.3 Parameter use

Invoke id:

See definition in subclause 5.6.1.

MSISDN:

See definition in subclause 5.6.2.

SM-RP-PRI:

See definition in subclause 5.6.8.

Service Centre Address:

See definition in subclause 5.6.2.

IMSI:

See definition in subclause 5.6.2. The presence of this parameter is mandatory in a successful case.

MSC Number:

See definition in subclause 5.6.2. This parameter is provided in a successful response.

LMSI:

See definition in subclause 5.6.2. It is an operator option to provide this parameter from the VLR; it is mandatory for the HLR to include the LMSI in a successful response, if the VLR has used the LMSI.

User error:

The following errors defined in subclause 5.6.1 may be used, depending on the nature of the fault:

- Unknown subscriber;

- Call Barred;

- Teleservice Not Provisioned;

- Absent Subscriber;

- Facility Not Supported;

- System failure;

- Unexpected Data Value;

- Data missing.

Provider error:

For definition of provider errors see subclause 5.6.1.

10.2   MAP-FORWARD-SHORT-MESSAGE service

10.2.1   Definition

This service is used between the gateway MSC and the servicing MSC to forward mobile originated or mobile terminated short messages.

The MAP-FORWARD-SHORT-MESSAGE service is a confirmed service using the service primitives given in table 10.2/1.

10.2.2   Service primitives

The service primitives are shown in table 10.2/1.

Parameter name	Request	Indication	Response	Confirm
Invoke Id	M	M(=)	M(=)	M(=)
SM RP DA	M	M(=)		
SM RP OA	M	M(=)		
SM RP UI	M	M(=)		
More Messages To Send	C	C(=)		
User error			C	C(=)
Provider error				O

Table 10.2/1: MAP-FORWARD-SHORT-MESSAGE

10.2.3   Parameter use

Invoke id:

See definition in subclause 5.6.1.

SM RP DA:

See definition in subclause 5.6.8. This parameter can contain either an IMSI or a LMSI in case of the mobile terminated SM. The use of the LMSI is an operator option. The LMSI can be provided if it is received from the HLR. The IMSI is used if the use of the LMSI is not available.

This parameter is omitted in the mobile terminated subsequent SM transfers.

In the mobile originated SM transfer this parameter contains the Service Centre address received from the MS.

SM RP OA:

See definition in subclause 5.6.8. The Service Centre address received from the originating Service Centre is inserted in this parameter in the mobile terminated SM transfer.

This parameter is omitted in the mobile terminated subsequent SM transfers.

The MSISDN received from the VLR is inserted in this parameter in the mobile originated SM transfer.

SM RP UI:

See definition in subclause 5.6.8. The short message transfer protocol data unit received from the Service Centre is inserted in this parameter.

More Messages To Send:

See definition in subclause 5.6.8. The information from MMS indication received from the Service Centre is inserted in this parameter. This is used only in the mobile terminated case.

User error:

The following errors defined in subclause 5.6.1 may be used, depending on the nature of the fault:

- Unidentified subscriber;

- Absent Subscriber;

- Subscriber busy for MT SMS;

- Facility Not Supported;

- Illegal Subscriber indicates that delivery of the mobile terminated short message failed because the MS failed authentication;

- Illegal equipment indicates that delivery of the mobile terminated short message failed because an IMEI check failed, i.e. the IMEI was blacklisted or not white-listed;

- System Failure;

- SM Delivery Failure;

    - The reason of the SM Delivery Failure can be one of the following in the mobile terminated SM:
        - memory capacity exceeded in the mobile equipment;

        - protocol error;

        - mobile equipment does not support the mobile terminated short message service.

    - The reason of the SM Delivery Failure can be one of the following in the mobile originated SM:
        - unknown Service Centre address;

        - Service Centre congestion;

        - invalid Short Message Entity address;

        - subscriber not Service Centre subscriber;

        - protocol error.

- Unexpected Data Value;

- Data Missing.

Provider error:

For definition of provider errors see subclause 5.6.1.

10.3 MAP-REPORT-SM-DELIVERY-STATUS service

10.3.1 Definition

This service is used between the gateway MSC and the HLR. The MAP-REPORT-SM-DELIVERY-STATUS service is used to set the Message Waiting Data into the HLR or to inform the HLR of successful SM transfer after polling. This service is invoked by the gateway MSC.

The MAP-REPORT-SM-DELIVERY-STATUS service is a confirmed service using the service primitives given in table 10.3/1.

10.3.2 Service primitives

The service primitives are shown in table 10.3/1.

Parameter name	Request	Indication	Response	Confirm
Invoke Id	M	M(=)	M(=)	M(=)
MSISDN	M	M(=)		
Service Centre Address	M	M(=)		
SM Delivery Outcome	M	M(=)		
MSIsdn-Alert			C	C(=)
User error			C	C(=)
Provider error				O

Table 10.3/1: MAP-REPORT-SM-DELIVERY-STATUS

10.3.3 Parameter use

Invoke id:

See definition in subclause 5.6.1.

MSISDN:

See definition in subclause 5.6.2.

Service Centre Address:

See definition in subclause 5.6.2.

SM Delivery Outcome:

See definition in subclause 5.6.8. This parameter indicates the status of the mobile terminated SM delivery.

MSIsdn-Alert:

See definition in subclause 5.6.2. The presence of this parameter is mandatory in case of successful outcome of the service.

User error:

The following errors defined in subclause 5.6.1 may be used, depending on the nature of the fault:

- Unknown Subscriber;

- Message Waiting List Full;

- Unexpected Data Value;

- Data missing.

Provider error:

For definition of provider errors see subclause 5.6.1.

10.4 MAP-READY-FOR-SM service

10.4.1 Definition

This service is used between the MSC and VLR and as well between the VLR and the HLR. The MSC initiates this service if a subscriber indicates memory available situation. The VLR uses the service to indicate this to the HLR.

The VLR initiates this service if a subscriber, whose message waiting flag is active in the VLR, has radio contact in the MSC.

The MAP-READY-FOR-SM service is a confirmed service using the primitives from table 10.4/1.

10.4.2 Service primitives

The service primitives are shown in table 10.4/1.

Parameter name	Request	Indication	Response	Confirm
Invoke Id	M	M(=)	M(=)	M(=)
IMSI	C	C(=)		
TMSI	C	C(=)		
Alert Reason	M	M(=)		
User error			C	C(=)
Provider error				O

Table 10.4/1: MAP-READY-FOR-SM

10.4.3 Parameter use

Invoke id:

See definition in subclause 5.6.1.

IMSI:

See definition in subclause 5.6.2. The IMSI is used always between the VLR and the HLR. Between the MSC and the VLR the identification can be either IMSI or TMSI.

TMSI:

See definition in subclause 5.6.2. The identification can be either IMSI or TMSI between MSC and VLR.

Alert Reason:

See definition in subclause 5.6.8. This parameter indicates if the mobile subscriber is present or the MS has memory available.

User error:

The following errors defined in subclause 5.6.1 may be used, depending on the nature of the fault:

- Unknown Subscriber;

- Facility Not Supported:

- System Failure;

- Unexpected Data Value;

- Data missing;

Provider error:

For definition of provider errors see subclause 5.6.1.

10.5 MAP-ALERT-SERVICE-CENTRE service

10.5.1 Definition

This service is used between the HLR and the interworking MSC. The HLR initiates this service, if the HLR detects that a subscriber, whose MSISDN is in the Message Waiting Data file, is active or the MS has memory available.

The MAP-ALERT-SERVICE-CENTRE service is a confirmed service using the primitives from table 10.5/1.

10.5.2 Service primitives

The service primitives are shown in table 10.5/1.

Parameter name	Request	Indication	Response	Confirm
Invoke Id	M	M(=)	M(=)	M(=)
MSIsdn-Alert	M	M(=)		
Service Centre Address	M	M(=)		
User error			C	C(=)
Provider error				O

Table 10.5/1: MAP-ALERT-SERVICE-CENTRE

10.5.3 Parameter use

Invoke id:

See definition in subclause 5.6.1.

MSIsdn-Alert:

See definition in subclause 5.6.2. The provided MSISDN shall be the one which is stored in the Message Waiting Data file.

Service Centre Address:

See definition in subclause 5.6.2.

User error:

The following errors defined in subclause 5.6.1 may be used, depending on the nature of the fault:

- System Failure;

- Unexpected Data Value;

- Data missing.

Provider error:

For definition of provider errors see subclause 5.6.1.

10.6 MAP-INFORM-SERVICE-CENTRE service

10.6.1 Definition

This service is used between the HLR and the gateway MSC to inform the Service Centre which MSISDN number is stored in the Message Waiting Data file. If the stored MSISDN number is not the same than the one received from the gateway MSC in the MAP-SEND-ROUTING-INFO-FOR-SM service primitive the stored MSISDN number is included in the message.

Additionally the status of MCEF and MNRF flags and the inclusion of the particular Service Centre address in the Message Waiting Data list is informed to the gateway MSC when appropriate.

The MAP-INFORM-SERVICE-CENTRE service is a non-confirmed service using the primitives from table 10.6/1.

10.6.2 Service primitives

The service primitives are shown in table 10.6/1.

Parameter name	Request	Indication
Invoke Id	M	M(=)
MSIsdn-Alert	C	C(=)
MWD Status	C	C(=)

Table 10.6/1: MAP-INFORM-SERVICE-CENTRE

10.6.3 Parameter use

Invoke id:

See definition in subclause 5.6.1.

MSIsdn-Alert:

See definition in subclause 5.6.2 This parameter refers to the MSISDN stored in a Message Waiting Data file in the HLR.

MWD Status:

See definition in subclause 5.6.8. This parameter indicates the status of the MCEF and MNRF flags and the status of the particular SC address presence in the Message Waiting Data list.

10.7   MAP-SEND-INFO-FOR-MT-SMS service

10.7.1   Definition

This service is used between the MSC and the VLR. The service is invoked by the MSC receiving an mobile terminated short message to request subscriber related information from the VLR.

The MAP-SEND-INFO-FOR-MT-SMS service is a confirmed service using the primitives from table 10.7/1.

10.7.2   Service primitives

The service primitives are shown in table 10.7/1.

Parameter name	Request	Indication	Response	Confirm
Invoke Id	M	M(=)	M(=)	M(=)
SM RP DA	M	M(=)		
MSISDN			C	C(=)
User error			C	C(=)
Provider error				O

Table 10.7/1: MAP-SEND-INFO-FOR-MT-SMS

10.7.3   Parameter use

Invoke id:

See definition in subclause 5.6.1.

SM RP DA:

See definition in subclause 5.6.8. This parameter shall contain either an IMSI or a LMSI.

MSISDN:

See definition in subclause 5.6.2.

User error:

The following errors defined in subclause 5.6.1 may be used, depending on the nature of the fault:

-        Unknown subscriber;
-        Unidentified Subscriber;
-        Absent subscriber;
-        Unexpected Data Value;
-        Data Missing;
-        Illegal subscriber;
-        Illegal equipment;
-        System Failure.

Provider error:

For definition of provider errors see subclause 5.6.1.

10.8 MAP-SEND-INFO-FOR-MO-SMS service

10.8.1 Definition

This service is used between the MSC and the VLR. The service is invoked by the MSC which has to handle a mobile originated short message request to request the subscriber related information from the VLR.

The MAP-SEND-INFO-FOR-MO-SMS service is a confirmed service using the primitives from table 10.8/1.

10.8.2 Service primitives

The service primitives are shown in table 10.8/1.

Parameter name	Request	Indication	Response	Confirm
Invoke Id	M	M(=)	M(=)	M(=)
Service Centre Address	M	M(=)		
MSISDN			C	C(=)
User error			C	C(=)
Provider error				O

Table 10.8/1: MAP-SEND-INFO-FOR-MO-SMS

10.8.3 Parameter use

Invoke id:

See definition in subclause 5.6.1.

Service Centre Address:

See definition in subclause 5.6.2.

MSISDN:

See definition in subclause 5.6.2.

User error:

The following errors defined in subclause 5.6.1 may be used, depending on the nature of the fault:

- Teleservice Not Provisioned;

- Call Barred;

- Unexpected Data Value;

- Data Missing.

Provider error:

For definition of provider errors see subclause 5.6.1.

11 General

11.1 Overview

Clause 11 to 14 specify the protocol elements to be used to provide the MAP services described in clause 5.

Clause 12 specifies the elements of procedures for the MAP protocol. Clause 13 specifies the mapping on to TC service primitives. Clause 14 specifies the application contexts, operation packages and abstract syntaxes for the MAP protocol as well as the encoding rules to be applied.

11.2 Underlying services

The MAP protocol relies on the services provided by the Transaction Capabilities (TC) of signalling system number 7, as referenced in clause 4.

11.3 Model

The MAP Protocol Machine (MAP PM) can be modelled as a collection of service state machines (SSMs) - one per MAP specific service invoked - coordinated by a MAP dialogue control function with its one state machine: MAP dialogue state machine (DSM). There are two types of Service State Machines: Requesting Service State Machines (RSM) and Performing Service State Machines (PSM).

A new invocation of a MAP PM is employed on the receipt of a MAP-OPEN request primitive or a TC-BEGIN indication primitive. Each invocation controls exactly one MAP dialogue. For each MAP specific service invoked during a dialogue, a MAP RSM is created at the requestor's side and a MAP PSM is created at the performer's side.

This modelling is used only to facilitate understanding and the MAP behaviour descriptions and is not intended to suggest any implementation. SDL descriptions are organized according to this model.

How the MAP-service-user and the MAP refer to a MAP dialogue (i.e. a MAP PM invocation) is a local implementation matter.

How TC dialogue identifiers are assigned to a MAP PM invocation is also a local implementation matter.

11.4 Conventions

The behaviour of the MAP PM depends on the application-context-name associated with the dialogue. One major difference is that the MAP requests the transfer of the application-context-name by TC only for those contexts which do not belong to the so-called "version one context set".

The "version one context set" is a set of application-contexts which model the behaviour of a MAP V1 implementation according to the latest phase 1 version of GSM 09.02. This set is defined in clause 12.

The procedures described in clause 12 are used when the application-context-name does not refer to a dialogue between an MSC and its VLR. When the application-context-name refers to a dialogue between an MSC and its VLR the MAP PM procedures are a local implementation matter.

12 Elements of procedure

12.1 Dialogue establishment

The establishment of a MAP dialogue involves two MAP-service-users, one that is the dialogue-initiator and one that is the dialogue-responder.

This procedure is driven by the following signals:

- a MAP-OPEN request primitive from the dialogue-initiator;

- a TC-BEGIN indication primitive occurring at the responding side;

- a MAP-OPEN response primitive from the dialogue-responder;

- the first TC-CONTINUE indication primitive occurring at the initiating side;

and under specific conditions:

- a TC-END indication primitive occurring at the initiating side;

- a TC-U-ABORT indication primitive occurring at the initiating side;

- a TC-P-ABORT indication primitive occurring at the initiating side.

12.1.1 Receipt of a MAP-OPEN request primitive

On receipt of a MAP-OPEN request primitive the behaviour of the MAP PM shall be as follows:

If the application-context-name belongs to the MAP version 1 contexts set, the MAP PM shall construct an argument for the beginSubscriberActivity operation if a user data parameter is present and request the invocation of this operation, using the TC-INVOKE service (note).

> NOTE: In practice this should only occur when the application-context is related to supplementary service handling.

The MAP PM shall then accept zero, one or several user request primitives until a MAP-DELIMITER request primitive is received.

For each user request primitive, the MAP PM shall request the invocation of the associated operation using the TC-INVOKE service. See subclause 12.6 for a description of the associated SSMs.

On receipt of the MAP-DELIMITER request primitive the MAP PM shall issue a TC-BEGIN request primitive. If the application-context-name does not belong to the MAP version 1 context set, this parameter as well as the user information parameter (if any) shall be mapped to the corresponding TC-BEGIN parameters.

The requesting MAP PM waits for a TC indication primitive and does not accept any other primitive from its user, except a MAP-U-ABORT request or a MAP-CLOSE request.

12.1.2　　Receipt of a TC-BEGIN indication

On receipt of a TC-BEGIN indication primitive, the MAP PM shall:

- if no application-context-name is included in the primitive and if the "Components present" indicator indicates "no components", issue a TC-U-ABORT request primitive (note 2). The local MAP-User is not informed.

- if no application-context-name is included in the primitive and if presence of components is indicated, wait for the first TC-INVOKE primitive, and derive a version 1 application-context-name from the operation code according to table 12.1/1 (note 1).

    NOTE 1:　　In some cases, it may be necessary to analyse the operation argument.

    Then:

    a)　　if no application-context-name can be derived (i.e. the operation code does not exist in MAP V1 specifications), the MAP PM shall issue a TC-U-ABORT request primitive (note 2). The local MAP-User is not informed.

    b)　　if an application-context-name can be derived and if it is acceptable from a load control point of view, the MAP PM shall:

    　　i)　　if this primitive requests the beginSubscriberActivity operation, the MAP PM shall check whether more components have been received associated with this operation. If more components are present, the MAP PM shall issue a MAP-OPEN indication primitive with the version 1 application-context-name "networkFunctionalSsContext-v1". The Destination-reference shall include the IMSI taken from the argument of the beginSubscriberActivity operation; the Originating-reference shall cover the originatingEntityNumber.

    　　　　A beginSubscriberActivity operation that is not associated with any other Component shall be rejected by the MAP PM by issuing a TC-U-ABORT request primitive (note 2). The local MAP-User shall not be informed.

    　　ii)　　otherwise, the MAP PM shall issue a MAP-OPEN indication primitive with the version 1 application-context-name set according to table 12.1/1. DestinationReference and OriginatingReference must not be included in the MAP-OPEN indication primitive.

    Then the MAP PM shall function in a way that the dialogue responding MAP behaves as specified in the GSM phase 1 protocol (latest version of GSM 09.02 phase 1).

    NOTE 2:　　If no AARQ apdu was included in the BEGIN message, TC (Component Sub-layer) will not include an AARE apdu or an ABRT apdu in a TR-U-ABORT request primitive that is to be issued on receipt of a TC-U-ABORT request primitive from the local MAP service provider.

c)   if an application-context-name can be derived but if it is not acceptable from a load control point of view, the MAP PM shall ignore this dialogue request and not inform the MAP-user;

-        if a version 1 application-context-name is included, the MAP PM shall issue a TC-U-ABORT request primitive with abort-reason "User-specific" and user-information "MAP-ProviderAbortInfo" indicating "abnormalDialogue". The local MAP-user shall not be informed.

-        if an application-context-name different from version 1 is included in the primitive and if User-information is present, the User-information must constitute a syntactically correct MAP-OPEN dialogue PDU. Otherwise a TC-U-ABORT request primitive with abort-reason "User-specific" and user-information "MAP-ProviderAbortInfo" indicating "abnormalDialogue" shall be issued and the local MAP-user shall not be informed.

-      if no User-information is present it is checked whether presence of User Information in the TC-BEGIN indication primitive is required for the received application-context-name. If User Information is required but not present, a TC-U-ABORT request primitive with abort-reason "User-specific" and user-information "MAP-ProviderAbortInfo" indicating "abnormalDialogue" shall be issued. The local MAP-user shall not be informed.

-      if an application-context-name different from version 1 is received in a syntactically correct TC-BEGIN indication primitive but is not acceptable from a load control point of view, the MAP PM shall ignore this dialogue request. The MAP-user is not informed.

-      if an application-context-name different from version 1 is received in a syntactically correct TC-BEGIN indication primitive and if it is acceptable from a load control point of view, the MAP PM shall check whether the application-context-name is supported.

NOTE 3:   Unknown application-context-names are treated like unsupported ones.

If it is, the MAP PM shall issue a MAP-OPEN indication primitive with all parameters (application-context-name included) set according to the value of the corresponding parameter of the TC-BEGIN indication primitive.

The MAP PM shall then process any other indication primitives received from TC as described in subclause 12.6. Once all the received components have been processed, the MAP PM shall inform the local MAP service user by a MAP-DELIMITER indication primitive.

If the TC-BEGIN indication primitive is not associated with any component, the MAP PM shall inform the MAP User by a MAP-DELIMITER indication primitive.

Once all the received primitives have been processed, the MAP PM does not accept any primitive from the provider and waits for a MAP-OPEN response primitive from its user.

-      if an application-context-name different from version 1 is received in a syntactically correct TC-BEGIN indication primitive and if it is acceptable from a load control point of view but the application-context-name is not supported, the MAP PM shall issue a TC-U-ABORT request primitive with abort-reason indicating "application-context-not-supported". If an alternative application-context-name cannot be offered, the received application-context-name shall be returned in the TC-U-ABORT Req primitive.

In the following cases an alternative application-context can be offered and its name included in the TC-U-ABORT Req primitive:

a)   if an application-context of version 2 or higher is requested, but only version 1 application-context supported, then the v1 application context shall be returned;

b)   if an application-context of version 3 or higher is requested, but only version 2 application-context supported, then the v2 application context shall be returned.

Operation	Application-context-name (note 1)
updateLocation	networkLocUpContext-v1
cancelLocation	locationCancellationContext-v1
provideRoamingNumber	roamingNumberEnquiryContext-v1
insertSubscriberData	subscriberDataMngtContext-v1
deleteSubscriberData	subscriberDataMngtContext-v1
sendParameters	infoRetrievalContext-v1
	networkLocUpContext-v1 (note 2)
beginSubscriberActivity	networkFunctionalSsContext-v1
sendRoutingInfo	locationInfoRetrievalContext-v1
performHandover	handoverControlContext-v1
reset	resetContext-v1
activateTraceMode	tracingContext-v1
deactivateTraceMode	tracingContext-v1
sendRoutingInfoForSM	shortMsgGatewayContext-v1
forwardSM	shortMsgRelayContext-v1
reportSM-deliveryStatus	shortMsgGatewayContext-v1
noteSubscriberPresent	mwdMngtContext-v1
alertServiceCentreWithoutResult	shortMsgAlertContext-v1
checkIMEI	EquipmentMngtContext-v1

NOTE 1:   These symbolic names refer to object identifier value defined in clause 14 and allocated to each application-context used for the MAP.

NOTE 2:   The choice between the application contexts is based on the parameters received in the operation.

Table 12.1/1: Mapping of V1 operation codes on to application-context-names

12.1.3   Receipt of a MAP-OPEN response

On receipt of a MAP-OPEN response primitive indicating that the dialogue is accepted, the MAP PM shall build a MAP-Accept PDU if the user-information parameter is included in the response primitive and accept any MAP specific service request or service response until a MAP-DELIMITER request or a MAP-CLOSE request is received from the MAP user. The MAP PM shall process the MAP specific primitives as described in subclause 12.6. The MAP PM shall then issue a TC-CONTINUE request primitive after it receives the MAP-DELIMITER request primitive if no MAP-CLOSE request primitive has been received, otherwise it shall issue a TC-END request primitive. In both cases the MAP-Accept PDU (if any) is included in the user-information parameter of the TC primitive.

If the dialogue is not associated with a version 1 application context, the MAP PM shall include the application-context-name in the TC primitive.

If no MAP-CLOSE request has been received, the MAP PM waits for a request primitive from its user or an indication primitive from TC.

On receipt of a MAP-OPEN response primitive indicating that the dialogue is not accepted, the MAP PM shall build a MAP-Refuse PDU and request its transfer using the TC-U-ABORT req primitive (abort reason = user specific).

12.1.4   Receipt of the first TC-CONTINUE ind

On receipt of the first TC-CONTINUE indication primitive for a dialogue, the MAP PM shall check the value of the application-context-name parameter (if any). If this value matches the one used in the MAP-OPEN request primitive, the MAP PM shall issue a MAP-OPEN confirm primitive with the result parameter indicating "accepted", then process the following TC component handling indication primitives as described in subclause 12.6, and then waits for a request primitive from its user or an indication primitive from TC, otherwise it shall issue a TC-U-ABORT request primitive with a MAP-providerAbort PDU indicating "abnormal dialogue" and a MAP-P-ABORT indication primitive with the "provider-reason" parameter indicating "abnormal dialogue".

12.1.5   Receipt of a TC-END ind

On receipt of a TC-END indication primitive in the dialogue initiated state, the MAP PM shall check the value of the application-context-name parameter (if any). If this value does not match the one used in the MAP-OPEN request primitive, the MAP PM shall discard any following component handling primitive and shall issue a MAP-P-ABORT indication primitive with the "provider-reason" parameter indicating "abnormal dialogue".

Otherwise it shall issue a MAP-OPEN confirm primitive with the result parameter set to "accepted" and process the following TC component handling indication primitives as described in subclause 12.6; then it shall issue a MAP-CLOSE indication primitive and return to idle all state machines associated with the dialogue.

12.1.6   Receipt of a TC-U-ABORT ind

On receipt of a TC-U-ABORT indication primitive in the "Dialogue Initiated" state with an abort-reason parameter indicating "ApplicationContextNotSupported", the MAP PM shall issue a MAP-OPEN confirm primitive with the result parameter indicating "Dialogue Refused" and the refuse-reason parameter indicating "ApplicationContextNotSupported".

On receipt of a TC-U-ABORT indication primitive in the "Dialogue Initiated" state with an abort-reason parameter indicating "User Specific" and without user information, the MAP PM shall issue a MAP-OPEN confirm primitive with the result parameter indicating "Dialogue Refused" and the refuse-reason parameter indicating "Potential Version Incompatibility".

On receipt of a TC-U-ABORT indication primitive in the "Dialogue Initiated" state with an abort-reason parameter indicating "User Specific" and a MAP-Refuse PDU included as user information, the MAP PM shall issue a MAP-OPEN confirm primitive with the result set to refused and the refuse reason set as received in the MAP Refuse PDU.

Receipt of a TC-U-ABORT indication primitive with abort-reason "User Specific" and with user information is described as part of abnormal termination (see subclause 12.4.2).

12.1.7   Receipt of a TC-P-ABORT ind

On receipt of a TC-P-ABORT indication primitive in the "Dialogue Initiated" state with a P-abort parameter indicating "Incorrect Transaction Portion", the MAP PM shall issue a MAP-OPEN confirm primitive with the result parameter indicating "Dialogue Refused" and the refuse reason parameter indicating "Potential Version Incompatibility".

On receipt of a TC-P-ABORT indication primitive in the "Dialogue Initiated" state with a P-abort parameter indicating "No Common Dialogue Portion", the MAP PM shall issue a MAP-P-ABORT indication primitive with the provider reason parameter indicating "Version Incompatibility".

Receipt of a TC-P-ABORT indication primitive with another P-abort parameter value is described as part of abnormal termination (see subclause 12.5.2).

12.2 Dialogue continuation

Once established the dialogue is said to be in a continuation phase.

Both MAP users can request the transfer of MAP APDUs until one of them requests the termination of the dialogue.

12.2.1 Sending entity

The MAP PM shall accept any MAP specific service request or response primitives and process them as described in subclause 12.6.

On receipt of a MAP-DELIMITER request primitive, the MAP PM shall issue a TC-CONTINUE request primitive.

12.2.2 Receiving entity

On receipt of a TC-CONTINUE indication primitive the MAP PM shall accept zero, one or several TC component handling indication primitives and process them as described in subclause 12.6.

12.3 Dialogue termination

Both the dialogue-initiator and the dialogue-responder have the ability to request the termination of a dialogue after it has been established.

The dialogue termination procedure is driven by the following events:

- a MAP-CLOSE request primitive;

- a TC-END indication primitive.

12.3.1 Receipt of a MAP-CLOSE request

On receipt of a MAP-CLOSE request primitive, the MAP PM shall issue a TC-END request primitive and, if applicable, return to idle the associated active SSMs. Note that if the release method parameter of the MAP-CLOSE request indicates "normal" the TC-END request primitive will trigger the transmission of components associated with any user specific request or response primitives which may have been issued after the last MAP-DELIMITER request.

12.3.2 Receipt of a TC-END indication

On receipt of a TC-END indication primitive, the MAP shall accept any component handling indication primitives and process them as described in subclause 12.6.

Once all the received primitives have been processed, the MAP PM shall return to idle the associated SSMs and issue a MAP-CLOSE indication primitive.

12.4 User Abort

Both the dialogue-initiator and the dialogue-responder have the ability to abort a dialogue at any time.

The user abort procedure is driven by one of the following events:

- a MAP-U-ABORT request primitive;

- a TC-U-ABORT indication primitive carrying a MAP-user-abort PDU.

12.4.1 MAP-U-ABORT request

On receipt of a MAP-U-ABORT request the MAP PM shall construct a MAP-user-abort PDU from the user-reason and diagnostic parameters and issue a TC-U-ABORT request primitive. All state machines associated with the dialogue are returned to idle.

12.4.2 TC-U-ABORT ind

On receipt of a TC-U-ABORT indication carrying a MAP-user-abort PDU, the MAP PM shall issue a MAP-U-ABORT indication primitive. The user-reason and diagnostic information elements are mapped to the corresponding parameters of the MAP-U-ABORT indication primitive.

All state machines associated with the dialogue are returned to idle.

12.5 Provider Abort

The MAP has the ability to abort a dialogue at both the dialogue-initiator side and the dialogue-responder side.

The provider abort procedure is driven by one of the following events:

- a MAP PM error situation;
- a TC-P-ABORT indication primitive;
- a TC-U-ABORT indication primitive carrying a MAP-abort PDU.

12.5.1 MAP PM error situation

In the case of an abnormal situation detected at the MAP level during an established dialogue, the MAP PM shall:

- issue a MAP-P-ABORT indication primitive with the appropriate value of the provider-reason parameter;

- construct a MAP-abort PDU from the value of these parameters and request its transfer using a TC-U-ABORT request primitive.

12.5.2 TC-P-ABORT ind

On receipt of a TC-P-ABORT indication, the MAP PM shall issue a MAP-P-ABORT indication primitive.

All state machines associated with the dialogue are returned to idle.

12.5.3 TC-U-ABORT ind

On receipt of a TC-U-ABORT indication carrying a MAP-abort PDU, the MAP PM shall issue a MAP-P-ABORT indication primitive, with the appropriate value of the provider-reason parameter. The source parameter shall indicate "MAP-provider".

All state machines associated with the dialogue are returned to idle.

12.6 Procedures for MAP specific services

This subclause describes the MAP procedures for MAP specific services.

These procedures are driven by the following types of events:

- a MAP specific request or a MAP specific MAP response primitive;

- a component handling primitive from TC.

A Service State Machine is activated on receipt of one of the following signals:

- a MAP request primitive, which activates a requesting SSM;

- a TC-INVOKE indication primitive without linked identifier, which activates a responding SSM.

For component handling primitives there are two types of events:

- events which activate a Service State Machine or which can be related to an existing one;
  The procedure elements driven by these events are described in subclauses 12.6.1 to 12.6.4.

- events which cannot be related to a Service State Machine.
  The procedure elements driven by these events are described in subclause 12.6.5.

12.6.1   Service invocation

The MAP specific procedures are initiated by the MAP request primitives.

On receipt of a MAP request primitive, the MAP PM shall build an operation argument from the parameters received in the request primitive and request the invocation of the associated operation using the TC-INVOKE procedure. If a linked ID parameter is inserted in the primitive this indicates a child service and implies that the operation on which the service is mapped is linked to the operation on which the parent service is mapped.

The mapping of MAP specific services on to remote operations is given in table 13.2/1.

12.6.2   Service invocation receipt

On receipt of a TC-INVOKE indication primitive, the MAP PM shall:

- if the invoke ID is already in use by an active service, request the transfer of a reject component using the TC-U-REJECT request primitive with the appropriate problem code (duplicated invokeID) and issue a MAP-NOTICE indication primitive with a diagnostic parameter set to "abnormal event received from the peer";

- if the operation code does not correspond to an operation supported by the application-context, request the transfer of a reject component using the TC-U-REJECT request primitive, with the appropriate problem code (unrecognized operation), and issue a MAP-NOTICE indication primitive with a diagnostic parameter set to "abnormal event received from the peer";

- if a linked ID is included, perform the following checks: If the operation referred to by the linked ID does not allow linked operations or if the operation code does not correspond to a permitted linked operation, issue a TC-U-REJECT request primitive with the appropriate problem code (linked response unexpected or unexpected linked operation) and issue a MAP-NOTICE indication primitive with an appropriate diagnostic "abnormal event received from the peer";

- if the type of the argument is not the one defined for the operation, request the transfer of a reject component using the TC-U-REJECT request primitive, with the appropriate problem code (mistyped parameter), and issue a MAP-NOTICE indication primitive with a diagnostic parameter set to "abnormal event from the peer";

- if the type of the argument is correct but the values of the information elements it contains do not permit the type of MAP service being invoked to be determined, request the transfer of an error component using the TC-U-ERROR request primitive with an error code set to "unexpected data value" and issue a MAP-NOTICE indication primitive with a diagnostic parameter set to "abnormal event from the peer";

  NOTE 1:   These checks are only relevant when there is not a one-to-one mapping between a service and an operation.

- if the type of the argument is correct but information elements required for the service being invoked are missing, request the transfer of an error component using the TC-U-ERROR request primitive with an error code set to "data missing" and issue a MAP-NOTICE indication primitive with a diagnostic parameter set to "abnormal event from the peer";

NOTE 2: These checks are only relevant when there is not a one-to-one mapping between a service and an operation.

- if the type of the argument is correct but contains information elements which are not relevant for the type of MAP service being invoked, request the transfer of an error component using the TC-U-ERROR request primitive with an error code set to "unexpected data value" and issue a MAP-NOTICE indication primitive with a diagnostic parameter set to "abnormal event from the peer";

NOTE 3: These checks are only relevant when there is not a one-to-one mapping between a service and an operation.

- Otherwise, issue the relevant MAP indication primitive to the MAP-service-user. If the service is to be user confirmed, the MAP PM waits for the corresponding response primitive.

12.6.3 Service response

For user confirmed services, the MAP PM shall accept a MAP response primitive and shall:

- if no error indication is included in the primitive and the service maps on to a class 1 or 3 operation, construct a result information element from the parameters received and request its transfer using the TC-RESULT-L service and optionally the TC-RESULT-NL service.

The TC-RESULT-NL services shall be used when the user specific parameters of the response primitives cannot be transferred in a single signalling frame and no segmenting mechanism is available from the underlying layers. The MAP PM shall issue one or several TC-RESULT-NL request primitives followed by a TC-RESULT-L primitive. The user parameters shall be split so that each portion contains sufficient information to construct a value compatible with the type defined for the result of the associated operation.

- if no error indication is included in the primitive and the service response maps on to a class 4 linked operation, construct an operation argument from the parameters received and request its transfer using the TC-INVOKE service for this class 4 linked operation. The operation to be invoked is deduced from the value of the result parameter of the service primitive;

- if an error indication is included in the primitive and the service maps on to a class 1 or 2 operation, either issue a TC-U-REJECT request primitive if the user error parameter indicates "resource limitation" or "initiating release", or construct an error parameter from the parameters received and request its transfer using the TC-U-ERROR request primitive. The error code should be the one associated with the value of the user error parameter of the response primitive.

NOTE: The only user errors that a MAP user can generate in addition to the list of errors attached to the operation which is associated with the service are: resource limitation and initiating release. Any other abnormal situation is detected either by the TC entity or by the MAP entity.

- if an error indication is received and the operation maps on to a class 3 operation, or if no error indication is received but the service maps on to a class 2 operation which has no class 4 linked operation, return the local service state machine to idle without requesting any service from TC.

12.6.4 Receipt of a response

A component handling indication primitive is considered as driving a response for a confirmed service if the invoke ID parameter value matches the one stored for the service, or if the linked ID parameter value matches the one stored for the service and the operation invoked is a class 4 operation. On receipt of a response (except a TC-L-CANCEL indication) for an unconfirmed service the MAP PM shall issue a MAP-NOTICE indication primitive with the appropriate provider error (return result unexpected or return error unexpected).

12.6.4.1 Receipt of a TC-RESULT-NL indication

If the type of the partial result parameter is not compatible with the one defined for the complete result of this operation, request the transfer of a reject component using the TC-U-REJECT request primitive, with the appropriate problem code (mistyped parameter) and issue a confirm primitive with the provider error parameter set to "invalid response received". The MAP PM shall also issue a TC-U-CANCEL request primitive so that all subsequent result components for this operation are discarded by TC.

Otherwise, store the value of the partial result parameter and wait for subsequent TC-RESULT-NL indication primitives until a TC-RESULT-L indication primitive is received.

12.6.4.2 Receipt of a TC-RESULT-L indication

If the type of the result parameter is not the one defined for the result of this operation, request the transfer of a reject component using the TC-U-REJECT request primitive, with the appropriate problem code (mistyped parameter), and issue a confirm primitive with the provider error parameter set to "invalid response received".

If the type of the result parameter is correct but does not contain all the information elements required by the service associated with the invocation, issue a confirm primitive with the provider error parameter set to "invalid response received".

> NOTE 1: These checks are only relevant when there is not a one-to-one mapping between a service and an operation.

If the type of the result parameter is correct but contains information elements which are not relevant for the service associated with the invocation are missing, issue a confirm primitive with the provider error parameter set to "invalid response received".

> NOTE 2: These checks are only relevant when there is not a one-to-one mapping between a service and an operation.

Otherwise, issue a MAP confirm primitive to the MAP-service-user mapping the result parameter of the TC-RESULT-L primitive on to the MAP specific parameters.

If partial results have been previously received, the value of the partial result parameters shall also be taken into account before performing the three previous checks.

12.6.4.3 Receipt of a TC-U-ERROR indication

If the error code is not defined for the MAP or is not one associated with the operation referred to by the invoke identifier, request the transfer of a reject component using the TC-U-REJECT request primitive, with the appropriate problem code (unrecognized error or unexpected error), and issue a confirm primitive with the provider error parameter set to "invalid response received".

If the type of the error parameter is not the one defined for this error, request the transfer of a reject component using the TC-U-REJECT request primitive, with the appropriate problem code (mistyped parameter), and issue a confirm primitive with the provider error parameter set to "invalid response received".

If the type of the error parameter is correct but does not contain all the information elements required by the service associated with the invocation, issue a confirm primitive with the provider error parameter set to "invalid response received".

> NOTE 1: In some cases, it may be necessary to analyse the operation argument.

If the type of the error parameter is correct but its value includes information elements which are not relevant for the service associated with the invocation, issue a confirm primitive with the provider error parameter set to "invalid response received".

NOTE 2: In some cases, it may be necessary to analyse the operation argument.

Otherwise, issue a MAP confirm primitive to the MAP-service-user with the user error parameter set according to the received error code. If applicable the error parameter is mapped to the diagnostic parameter.

12.6.4.4 Receipt of a TC-INVOKE indication

A TC-INVOKE indication primitive is considered as carrying a possible response to a specific service if the linked ID refers to an active specific service and the associated operation is a class 4 operation. Note that the presence of a linked ID parameter in a TC-INVOKE primitive requesting a non class 4 operation indicates a child service whose procedures are the same as the procedures for the parent service.

On receipt of a TC-INVOKE indication confirming an active service, the MAP PM shall:

- if the operation code is not defined for MAP or if the operation referred to by the linked ID does not allow linked operations or if the operation code does not correspond to an allowed linked operation, issue a TC-U-REJECT request primitive with the appropriate problem code (unrecognised operation, linked response unexpected or unexpected linked operation). If the service is confirmed, the MAP shall also issue a Confirm primitive with provider error indication "unexpected response from the peer", otherwise it may issue a MAP-NOTICE indication primitive with an appropriate diagnostic "abnormal event received from the peer".

- otherwise issue a confirm primitive mapping the operation argument parameter to the user specific parameters and setting the result parameter according to the operation code of the linked operation.

12.6.4.5 Receipt of a TC-U-REJECT indication

On receipt of a TC-U-REJECT indication primitive which affects a pending service, the MAP PM shall issue a MAP confirm primitive to the MAP-service-user with the appropriate value of the provider error or user error parameter.

The mapping of TC invoke problem codes on to MAP Provider Error and MAP User Error parameter values is described in clause 13.

12.6.4.6 Receipt of a TC-L-REJECT indication

This event occurs when the local TC detects a protocol error in an incoming component which affects an active specific service.

On receipt of a TC-L-REJECT indicating "return result problem, unexpected return result", the MAP shall issue a confirm primitive with the parameter provider error indicating "unexpected response from the peer".

On receipt of a TC-L-REJECT indicating "return error problem, unexpected error result", the MAP shall issue a confirm primitive with the parameter provider error indicating "unexpected response from the peer".

Note that when the problem code indicates a general problem, it is considered that the event cannot be related to an existing SSM even if the invoke Id is provided by TC. This is because whether the invoke Id refers to a local or remote invocation is ambiguous. The behaviour of the MAP PM in such a case is described in subclause 12.6.5.3.

12.6.4.7   Receipt of a TC-L-CANCEL indication

On receipt of a TC-L-CANCEL indication, the MAP PM shall:

-        if the associated operation is a class 1 operation, issue a confirm primitive with the provider error cause indicating "no response from the peer";

-        if the associated operation is a class 2 operation and no linked operations are defined for this operation, issue a confirm primitive without parameter (i.e. indicating implicitly the successful completion of the service);

-        if the associated operation is a class 2 operation and has linked operations but none of them has been invoked, issue a confirm primitive with the provider error parameter indicating "service completion failure";

-        if the associated operation is a class 2 operation and a linked operation invocation has already been received in response to this operation, ignore the primitive;

-        if the associated operation is a class 3 operation, issue a confirm primitive with the provider error cause indicating "service completion failure";

-        if the associated operation is a class 4 operation, ignore the primitive.

        NOTE: When a TC-L-CANCEL ind primitive is received before the dialogue has been confirmed (i.e. no backward message is received by the dialogue initiator node), the MAP PM shall first issue a MAP-OPEN Cnf primitive with the result parameter indicating "accepted" (which means that the dialogue is considered as being implicitly accepted). Then, as indicated above, the TC-L-CANCEL Indication is interpreted according to the class of the operation to which it refers.

12.6.4.8   Receipt of a TC-NOTICE indication

If a TC-NOTICE indication primitive is received before the dialogue has been confirmed (i.e. no backward message is received by the dialogue initiator node), the MAP PM shall issue a MAP-OPEN Cnf primitive with the result parameter indicating Refused and a refuse reason Remote node not reachable".

If a TC-NOTICE indication primitive is received after the dialogue has been confirmed, the MAP PM shall issue a MAP-NOTICE indication to the user, with a problem diagnostic indicating "message cannot be delivered to the peer".**12.6.5   Other events**

This subclause describes the behaviour of the MAP PM on receipt of a component handling indication primitive which cannot be related to any service or which does not affect a pending one. The MAP user is only informed that an abnormal event occurred during the associated dialogue. It is up to the MAP user to abort, continue or terminate the dialogue.

12.6.5.1   Receipt of a TC-U-REJECT

On receipt of a TC-U-REJECT indication primitive which does not affect an active SSM (i.e. indicating a return result or return error problem), the MAP PM shall issue a MAP-NOTICE indication primitive with the diagnostic parameter set to "response rejected by the peer".

This is also applicable for invoke problems related to a class 4 linked operation.

12.6.5.2 Receipt of a TC-R-REJECT indication

On receipt of a TC-R-REJECT indication (i.e. when a protocol error has been detected by the peer TC entity) which does not affect an active SSM, the MAP PM shall either discard this indication or issue a MAP-NOTICE indication primitive with the provider error indicating "abnormal event detected by the peer".

In case of notification, it is up to the MAP user to continue, abort or terminate the dialogue. Note also that for MAP V1 the reject component is received in an END message and therefore the dialogue is terminated anyway.

12.6.5.3 Receipt of a TC-L-REJECT indication

On receipt of a TC-L-REJECT indication primitive (i.e. when a protocol error has been detected by the local TC entity) which cannot be related to an active SSM, the MAP PM shall either discard this indication or issue a MAP-NOTICE indication primitive with the provider error indicating "abnormal event received from the peer".

In case of notification, it is up to the MAP user to continue, or to terminate the dialogue and implicitly trigger the transmission of the reject component or to abort the dialogue.

12.6.6 Parameter checks

As described in the previous subclauses, the MAP PM performs a set of checks to ensure the correctness of the information elements received; these are:

- check if the syntax and encoding (note) of the operation argument, result or error parameter are correct.

  NOTE: Depending on the implementation, encoding problems on the TC user portion may be detected at TC level or by the MAP user. In the second case the problem is reported in a similar manner to a syntactical problem.

  The syntax shall be considered incorrect if a mandatory information element is missing in any constructed element or if the value of an information element is out of the range defined for the type it is supposed to belong to;

- if there is not a one-to-one mapping between a service and an operation:

  i) check if the value of the information elements (generally a single one) permits the MAP PM to determine the service associated with the operation invocation;

  ii) check that there are no information elements which are irrelevant for the indication or a confirm primitive to be issued;

- check if all the information elements required to built an indication or a confirm primitive are available.

However some additional checks may have to be performed by the MAP user (see clause 15).

12.6.7 Returning state machines to idle

Unlike TC invocation state machines, service state machines exist at both requestor and performer side.

A service state machine at the requestor side is returned to idle when the MAP-specific confirm primitive is issued or when the dialogue terminates.

A service state machine at the performer side is returned to idle on receipt of a MAP-specific response primitive from the MAP user, when the dialogue terminates or at expiry of an implementation dependent watch-dog timer which is started when the state machine is created.

12.6.8 Load control

As stated in the previous subclauses, before issuing a MAP-OPEN indication primitive the MAP PM performs a check to verify if there are sufficient resources to open the dialogue taking into account possible overload conditions.

The decision is based on the priority allocated to the application-context whose name is explicitly included in the TC-BEGIN indication primitive or implied by the first operation invocation when V1 contexts are in use. How a V1 application-context-name is derived from an operation code is described in table 12.1/1.

The priority level allocated to each application-context is described in clause 3 tables 3.1/1 and 3.1/2.

13 Mapping on to TC services

13.1 Dialogue control

Dialogue control services are mapped to TC dialogue handling services. The TC-UNI service is not used by the MAP PM.

13.1.1 Directly mapped parameters

The following parameters of the MAP-OPEN request and indication primitives are directly mapped on to the corresponding parameters of the TC-BEGIN primitives:

- destination address;

- originating address.

13.1.2 Use of other parameters of dialogue handling primitives

13.1.2.1 Dialogue Id

The value of this parameter is associated with the MAP PM invocation in an implementation dependent manner.

13.1.2.2 Application-context-name

If it does not belong to the MAP V1 context set. The application-context-name parameter of a MAP primitive is mapped to the application-context-name parameter of TC dialogue handling primitives according to the rules described in subclause 12.1. Otherwise the application-context-name parameter of TC dialogue handling primitives is never included.

13.1.2.3 User information

The user information parameter of TC dialogue primitives is used to carry the MAP dialogue APDUs.

13.1.2.4 Component present

This parameter is used by the MAP PM as described in CCITT Recommendation Q.771. It is not visible to the MAP user.

13.1.2.5 Termination

The value of this parameter of the TC-END request primitive is set by the MAP PM on the basis of the release method parameter of the MAP-CLOSE request primitive, except when the dialogue state machine is in the state DIALOGUE INITIATED, in which case the Termination parameter shall always indicate "pre-arranged end".

13.1.2.6 P-Abort-Cause

Values of the P-abort-cause parameter are mapped to the values of the provider-reason parameter of the MAP-P-ABORT indication primitive according to table 13.1/1, except in the dialogue initiated phase for the "incorrectTransactionPortion" and "noCommonDialoguePortion" values which are mapped to the "potential incompatibility problem" value of the refuse-reason parameter of the MAP-OPEN cnf primitive. The source parameter in the MAP-P-ABORT ind takes the value "TC problem".

13.1.2.7 Quality of service

The quality of service of TC request primitives is set by the MAP as shown below.

- Return option: "Return message on error" or "Discard message on error" as required by the network operator;

- Sequence control: "Sequence guaranteed" or "Sequence result not guaranteed" as required by the network operator;

    "Sequence guaranteed" shall be used when a segmented result is to be transferred (e.g. subscriber data in response to SendParameters). It may also be appropriate to use Sequence guaranteed when a series of InsertSubscriberData, ProcessAccessSignalling or ForwardAccessSignalling operations is used.

TC P-Abort cause	MAP provider-reason
unrecognized message type	provider malfunction
unrecognized transaction Id	supporting dialogue released
badlyFormattedTransactionPortion	provider malfunction
incorrectTransactionPortion	provider malfunction (note)
resourceLimitation	resource limitation
abnormalDialogue	provider malfunction
noCommonDialoguePortion	version incompatibility

NOTE: Or version incompatibility in the dialogue initiated phase.

Table 13.1/1: Mapping of P-Abort cause in TC-P-ABORT indication on to provider-reason in MAP-P-ABORT indication

13.2 Service specific procedures

Specific services are mapped to TC component handling services.

13.2.1 Directly mapped parameters

The Invoke Id parameter of the MAP request and indication primitive is directly mapped on to the Invoke Id parameter of the component handling primitives.

13.2.2 Use of other parameters of component handling primitives

13.2.2.1 Dialogue Id

The value of this parameter is associated with the MAP PM invocation in an implementation dependent manner.

13.2.2.2 Class

The value of this parameter is set by the MAP PM according to the type of the operation to be invoked.

13.2.2.3   Linked Id

When a service response is mapped to a class 4 operation, the value of this parameter is set by the MAP PM and corresponds to the value assigned by the user to the initial service request (i.e. the value of the invoke ID parameter of the request primitive). Otherwise if such a parameter is included in MAP request/indication primitives it is directly mapped to the linked ID parameter of the associated TC-INVOKE request/indication primitives.

13.2.2.4   Operation

When mapping a request primitive on to a Remote Operations PDU (invoke), the MAP PM shall set the operation code according to the mapping described in table 13.2/1.

When mapping a response primitive on to a Remote Operations service, the MAP PM shall set the operation code of the TC-RESULT-L/NL primitive (if required) to the same value as the one received at invocation time.

MAP-SERVICE	operation
MAP-ACTIVATE-SS	activateSS
MAP-ACTIVATE-TRACE-MODE	activateTraceMode
MAP-ALERT-SERVICE-CENTRE	alertServiceCentre
MAP-CANCEL-LOCATION	cancelLocation
MAP-CHECK-IMEI	checkIMEI
MAP-DEACTIVATE-SS	deactivateSS
MAP-DEACTIVATE-TRACE-MODE	deactivateTraceMode
MAP-DELETE-SUBSCRIBER-DATA	deleteSubscriberData
MAP-ERASE-SS	eraseSS
MAP-FORWARD-ACCESS-SIGNALLING	forwardAccessSignalling
MAP-FORWARD-CHECK-SS-INDICATION	forwardCheckSsIndication
MAP-FORWARD-SHORT-MESSAGE	forwardSM
MAP-GET-PASSWORD	getPassword
MAP-INFORM-SERVICE-CENTRE	informServiceCentre
MAP-INSERT-SUBSCRIBER-DATA	insertSubscriberData
MAP-INTERROGATE-SS	interrogateSs
MAP-PREPARE-HANDOVER	prepareHandover
MAP-PREPARE-SUBSEQUENT-HANDOVER	prepareSubsequentHandover
MAP-PROCESS-ACCESS-SIGNALLING	processAccessSignalling
MAP-PROCESS-UNSTRUCTURED-SS-REQUEST	processUnstructuredSS-Request
MAP-PROVIDE-ROAMING-NUMBER	provideRoamingNumber
MAP-PURGE-MS	purgeMS
MAP-READY-FOR-SM	readyForSM
MAP-REGISTER-PASSWORD	registerPassword
MAP-REGISTER-SS	registerSS
MAP-REPORT-SM-DELIVERY-STATUS	reportSmDeliveryStatus
MAP-RESET	reset
MAP-RESTORE-DATA	restoreData
MAP-SEND-END-SIGNAL	sendEndSignal
MAP-SEND-AUTHENTICATION-INFO	sendAuthenticationInfo
MAP-SEND-IMSI	sendIMSI
MAP-SEND-IDENTIFICATION	sendIdentification
MAP-SEND-ROUTING-INFO-FOR-SM	sendRoutingInfoForSM
MAP-SEND-ROUTING-INFORMATION	sendRoutingInfo
MAP-UNSTRUCTURED-SS-NOTIFY	unstructuredSS-Notify
MAP-UNSTRUCTURED-SS-REQUEST	unstructuredSS-Request
MAP-UPDATE-LOCATION	updateLocation

Table 13.2/1: Mapping of MAP specific services on to MAP operations v2

13.2.2.5 Error

The error parameter in a TC-U-ERROR indication primitive is mapped to the user error parameter in the MAP confirm primitive of the service associated with the operation to which the error is attached.

The user error parameter in MAP response primitives is mapped to the error parameter of the TC-U-ERROR request primitive, except for "initiating-release" and "resource-limitation" which are mapped to the problem code parameter of the TC-U-REJECT request primitive.

13.2.2.6 Parameters

The parameters of MAP specific request and indication primitives are mapped to the argument parameter of TC-INVOKE primitives.

The parameters of MAP specific response and confirm primitives are mapped to the result parameter of TC-RESULT-L primitives, the parameter of TC-U-ERROR primitives or the argument of TC-INVOKE primitives when mapping on linked class 4 operations is used.

13.2.2.7 Time out

The value of this parameter is set by the MAP PM according to the type of operation invoked.

13.2.2.8 Last component

This parameter is used by the MAP PM as described in CCITT Recommendation Q.711. It is not visible from the MAP user.

13.2.2.9 Problem code

13.2.2.9.1 Mapping to MAP User Error

The following values of the user error parameter are mapped as follows to values of the TC problem code parameter. These values are generated by the MAP user. This mapping is valid from the TC-U-REJECT indication primitive to the MAP confirm service primitive and from the MAP response service primitive to the TC-U-REJECT request primitive.

MAP User Error	TC problem code
resource limitation	resource limitation
initiating release	initiating release

Table 13.2/2: Mapping of MAP User Error parameter on to TC problem code in TC-U-REJECT primitives

13.2.2.9.2 Mapping to MAP Provider Error parameter

The following values of the TC problem code parameter of the TC-U-REJECT indication primitive are mapped as follows to values of the MAP Provider Error parameter of the MAP confirm primitive.

TC problem code	MAP Provider Error
duplicated invoke Id	duplicated invoke id
unrecognized operation	service not supported
mistyped parameter	mistyped parameter

Table 13.2/3: Mapping of TC problem code in TC-U-REJECT on to MAP Provider Error parameter

The following values of the problem code parameters of the TC-L-REJECT primitive are mapped to values of the provider error parameter of the MAP confirm primitive as follows:

TC problem code	MAP Provider Error
return result unexpected	unexpected response from the peer
return error unexpected	unexpected response from the peer

Table 13.2/4: Mapping of TC problem code in TC-L-REJECT on to MAP Provider Error parameter

13.2.2.9.3 Mapping to diagnostic parameter

The following values of the problem code parameter of the TC-R-REJECT and TC-U-REJECT primitive are mapped to values of the diagnostic parameter of the MAP-NOTICE indication primitive as follows:

TC problem code	MAP diagnostic
General problem	
abnormal event detected by the peer	
Invoke problem	
- unrecognized linked ID	- abnormal event detected by the peer
- linked response unexpected	- response rejected by the peer
- unexpected linked operation	- response rejected by the peer
Return result problem	
- unrecognized invoke ID	- response rejected by the peer
- return result unexpected	- response rejected by the peer
- mistyped parameter	- response rejected by the peer
Return error problem	
- unrecognized invoke ID	- response rejected by the peer
- return error unexpected	- response rejected by the peer
- unrecognized error	- response rejected by the peer
- unexpected error	- response rejected by the peer
- mistyped parameter	- response rejected by the peer

Table 13.2/5: Mapping of TC problem code of TC-R-REJECT and TC-U-REJECT on to diagnostic parameter

The following values of the problem code parameter of the TC-L-REJECT primitive are mapped to values of the diagnostic parameter of the MAP-NOTICE indication primitive as follows:

TC problem code	MAP diagnostic
General problems:	- abnormal event received from the peer
Invoke problem:	
- unrecognized linked ID	- abnormal event received from the peer
Return result problem:	
- unrecognized invoke ID	- abnormal event received from the peer
Return error problem:	
- unrecognized invoke ID	- abnormal event received from the peer

Table 13.2/6: Mapping of TC problem code of TC-L-REJECT on to diagnostic parameter

13.3 SDL descriptions

The following SDL specification describes a system which includes three blocks: MAP-user, MAP-provider and TC.

Such a system resides in each network component supporting MAP and communicates with its peers via the lower layers of the signalling network which are part of the environment.

Only the MAP-provider is fully described in this subclause. The various type of processes which form the MAP-User block and the TC block are described respectively in clauses 15 to 21 of this ETS and in CCITT Recommendation Q.774.

The MAP-Provider block communicates with the MAP_USER via two channels U1 and U2. Via U1 the MAP-provider receives the MAP request and response primitives. Via U2 it sends the MAP indication and confirm primitives.

The MAP-Provider block communicates with TC via two channels P1 and P2. Via P1 the MAP-Provider sends all the TC request primitives. Via P2 it receives all the TC indication primitives.

The MAP-Provider block is composed of the four following types of processes:

a)   MAP_DSM: This type of process handles a dialogue. There exists one process instance per MAP dialogue.

b)   LOAD_CTRL: This type of process is in charge of load control. There is only one instance of this process in each system.

c)   PERFORMING_MAP_SSM: This type of process handle a MAP service performed during a dialogue. An instance of this process is created by the instance of the MAP_DSM process for each MAP-service to be performed.

d)   REQUESTING_MAP_SSM: This type of process handle a MAP service requested during a dialogue. An instance of this process is created by the instance of the MAP_DSM process for each requested MAP-service.

A process MAP_DSM exchanges external signals with other blocks as well as internal signals with the other processes of the MAP-Provider block. The external signals are either MAP service primitives or TC service primitives.

The signal routes used by the various processes are organized as follows:

a)   A process MAP_DSM receives and sends events from/to the MAP_user via signal route User1/User2. These routes uses respectively channel U1 and U2.

b)   A process MAP_DSM receives and sends events from/to the TC via signal route Tc1/Tc2. These routes uses respectively channel P1 and P2.

c)   A process MAP_DSM receives and sends events from/to the LOAD_CTRL process via signal route Load1/Load2. These routes are internal.

d)   A process MAP_DSM sends events to the PERFORMING_MAP_SSM processes via signal route Intern1. This route is internal.

e)   A process MAP_DSM sends events to the REQUESTING_MAP_SSM processes via signal route Intern2. This route is internal.

f)   A process MAP_PERFORMING_SSM sends events to the MAP_USER via signal route User4. This route uses channel U2.

g)   A process MAP_PERFORMING_SSM sends events to TC via signal route Tc3. This route uses channel P1.

h)   A process MAP_REQUESTING_SSM sends events to the MAP_USER via signal route User5. This route uses channel U2.

j)   A process MAP_REQUESTING_SSM sends events to TC via signal route Tc4. This route uses channel P1.

Figure 13.2/1: System MAP_STACK
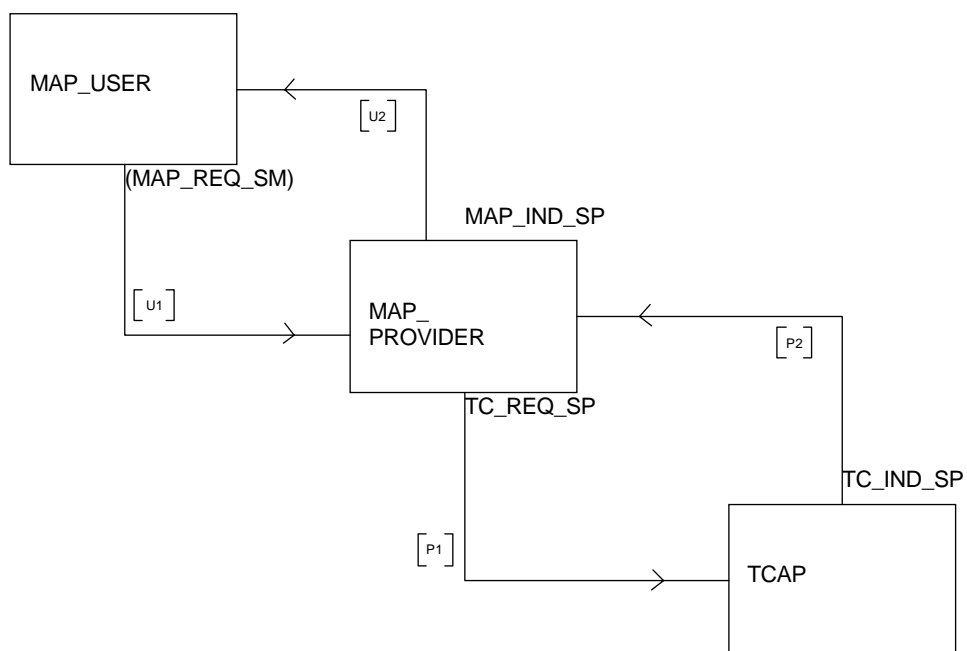
Figure 13.2/2: Block MAPPROVIDER
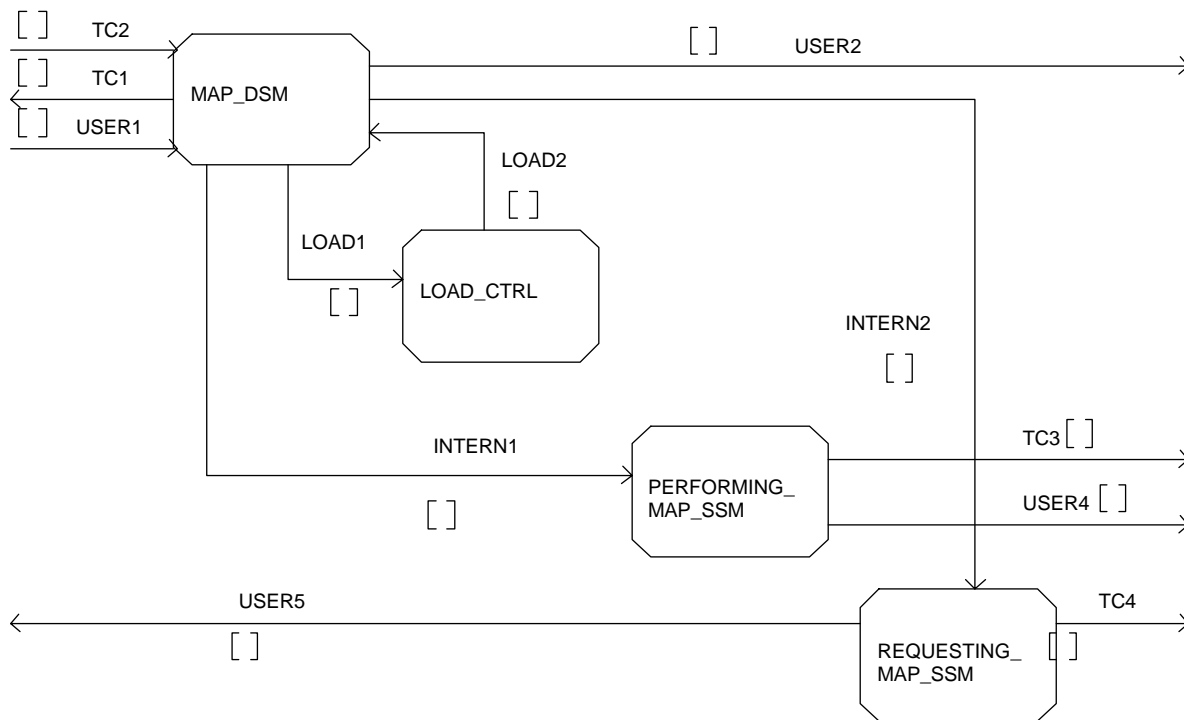
Figure 13.2/3 (sheet 1 of 11): Process MAP_DSM

Comment 'MAP Dialoges State Maschine':
DCL
  COMPONENTS_PRESENT, INVOKEID_ACTIVE, LAST_COMPONENT,OP_EXIST BOOLEAN,
  OP_CODE INTEGER;

IDLE

IDLE

MAP_OPEN_
REQ

MAP_V1
FALSE
TRUE

User_data
included
FALSE
TRUE

TC_INVOKE_
REQ_VIA_TC1

Operation =
Begin Subscriber
Activity

Store_AC_and_
user_data

WAIT_FOR_
USER_REQUESTS
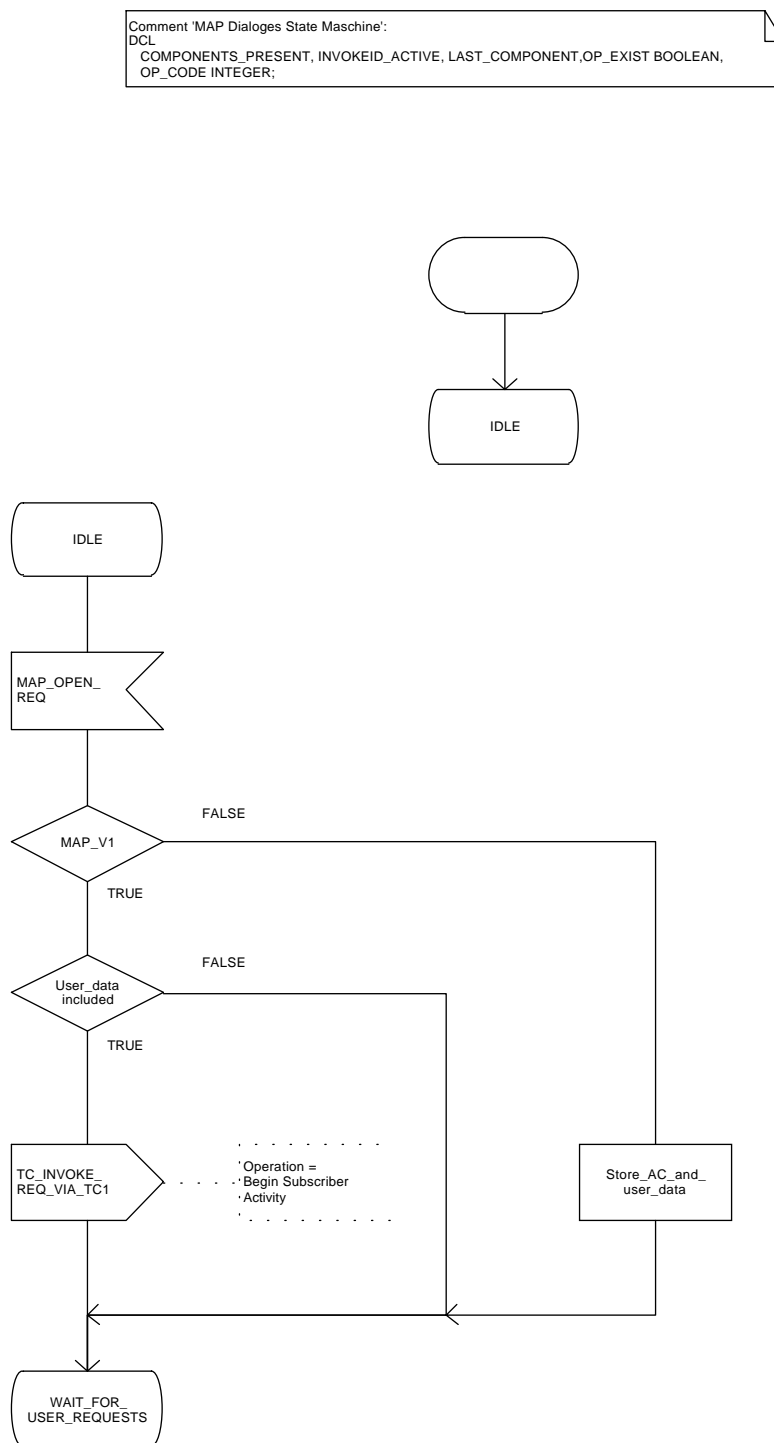
Figure 13.2/3 (sheet 2 of 11): Process MAP_DSM
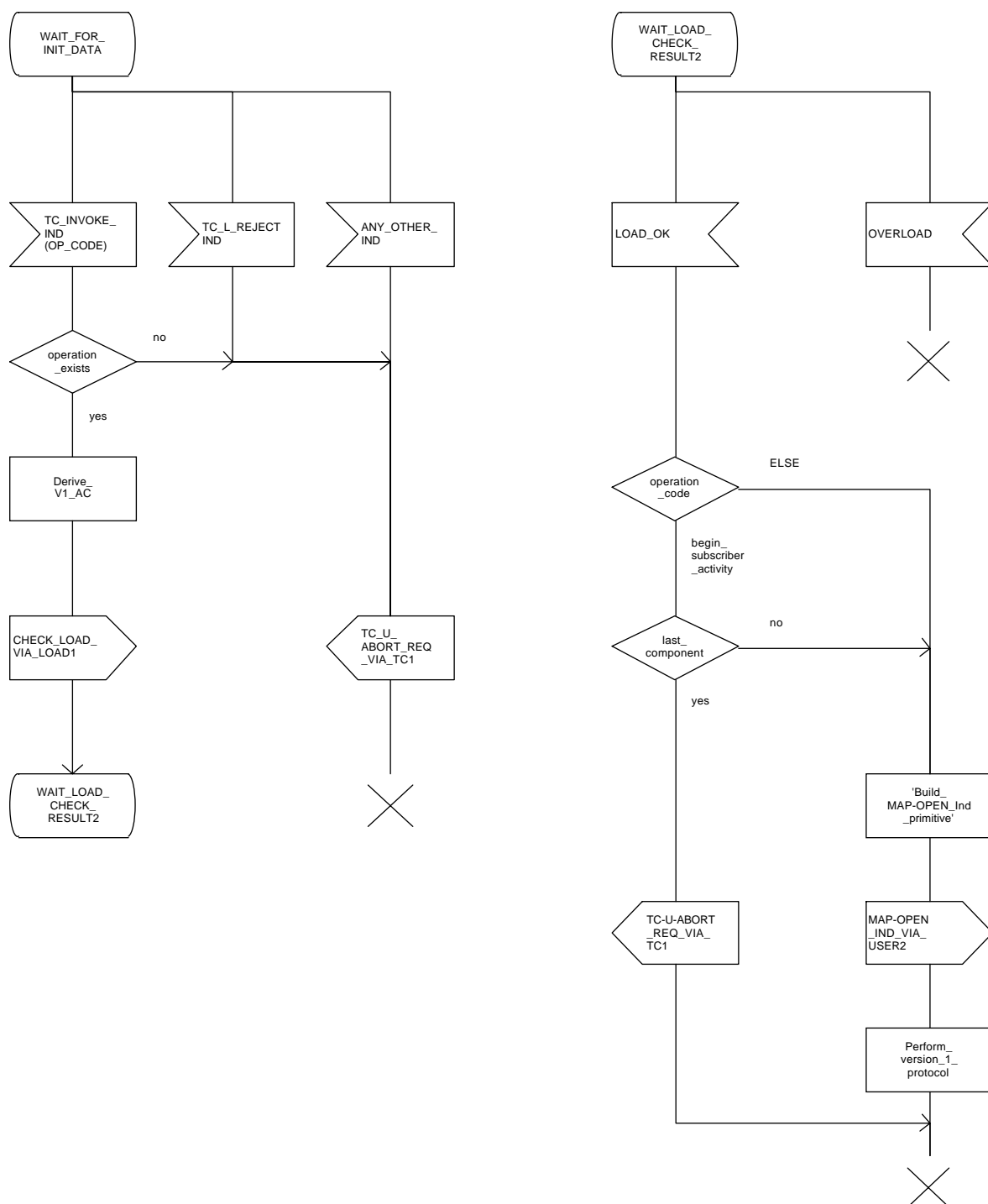
Figure 13.2/3 (sheet 3 of 11): Process MAP_DSM
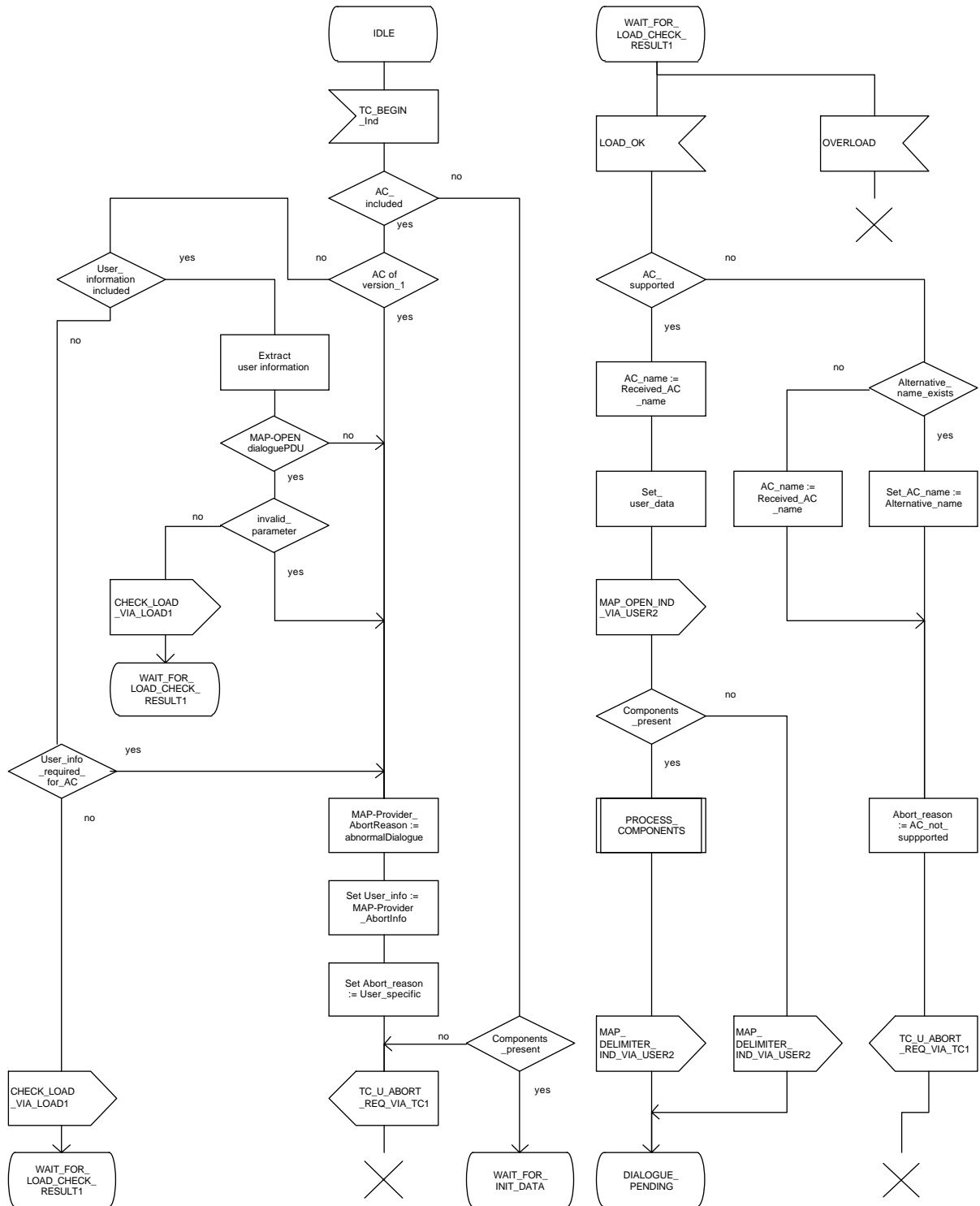
Figure 13.2/3 (sheet 4 of 11): Process MAP_DSM
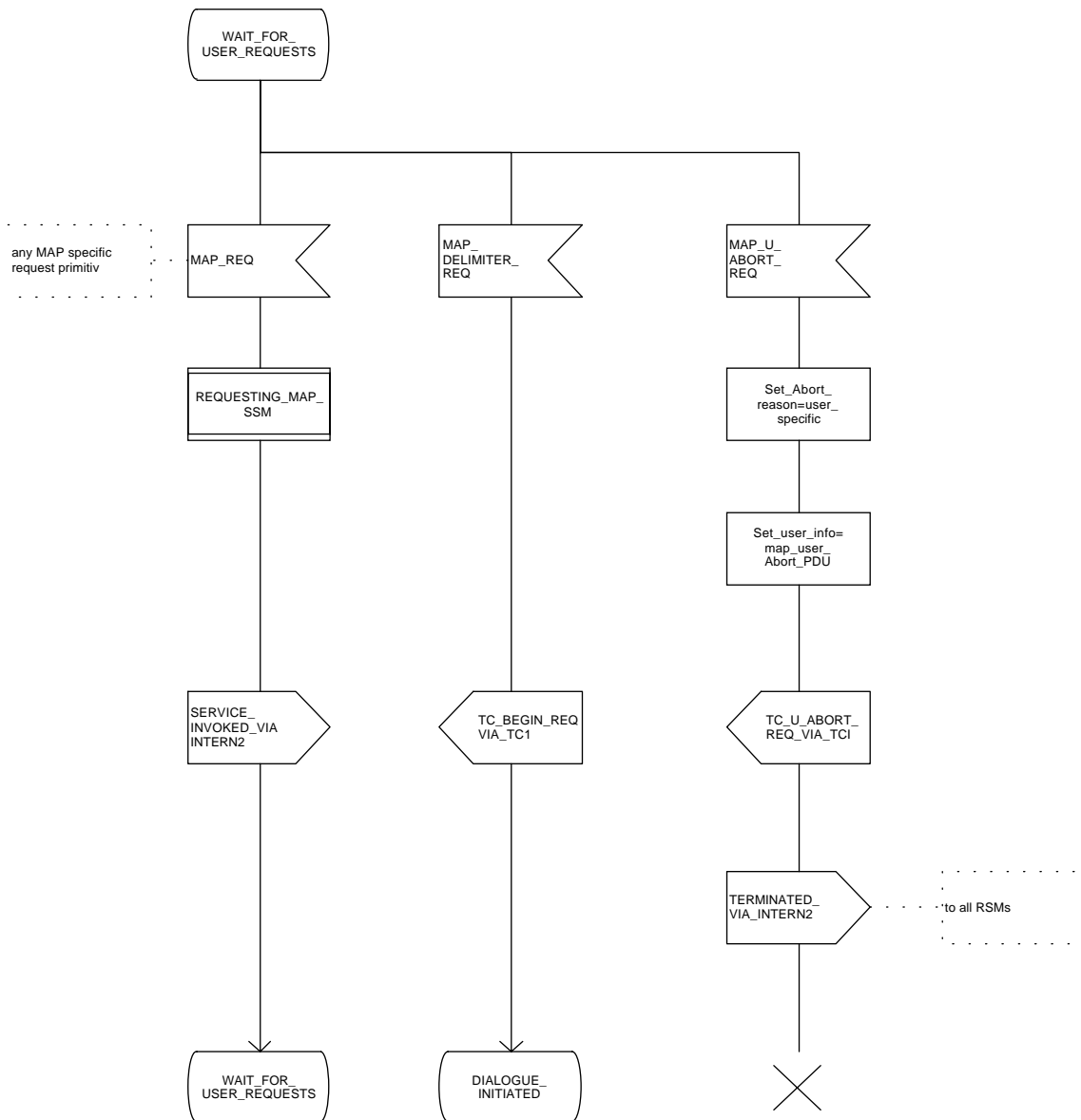
Figure 13.2/3 (sheet 5 of 11): Process MAP_DSM
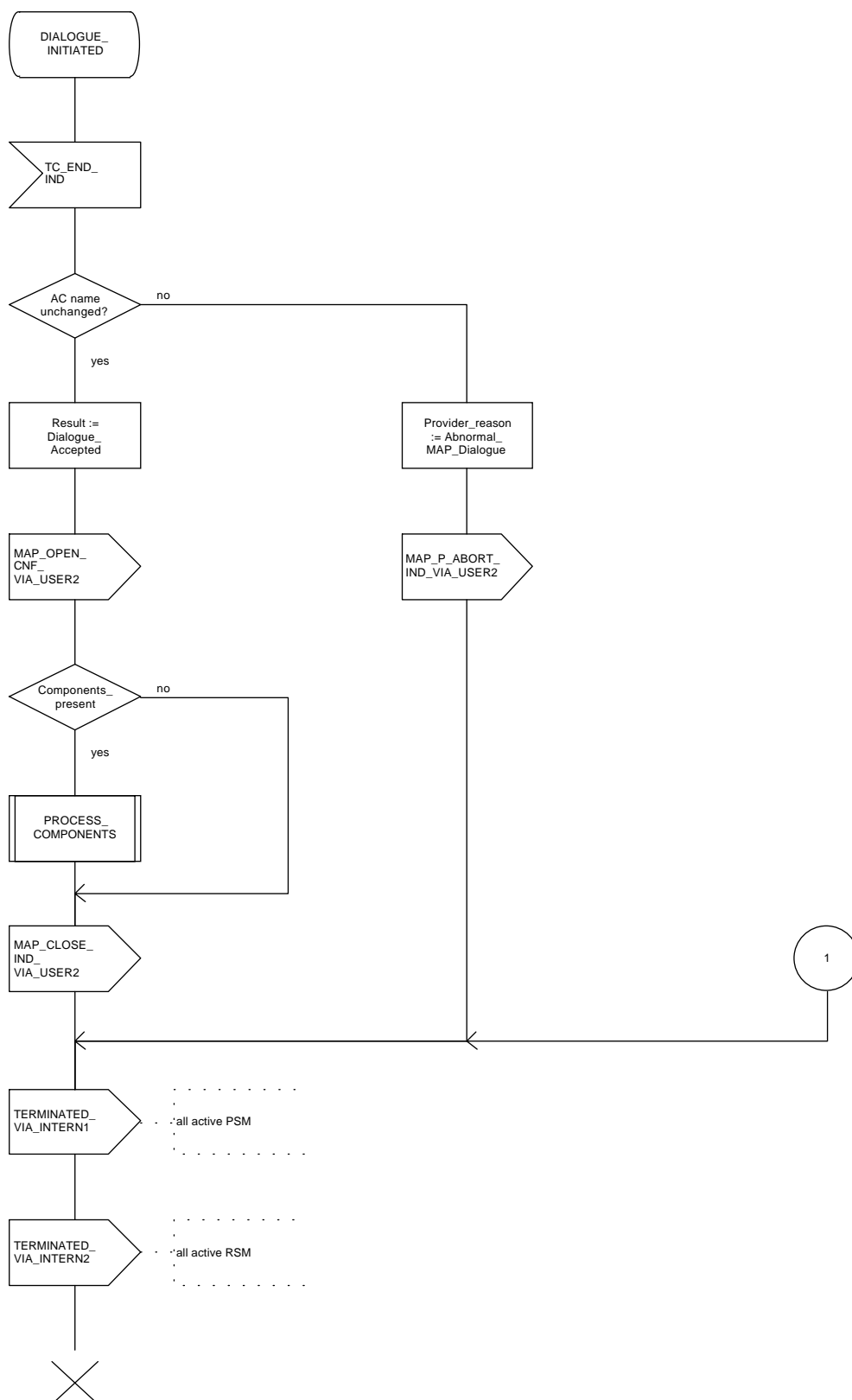
Figure 13.2/3 (sheet 6 of 11): Process MAP_DSM
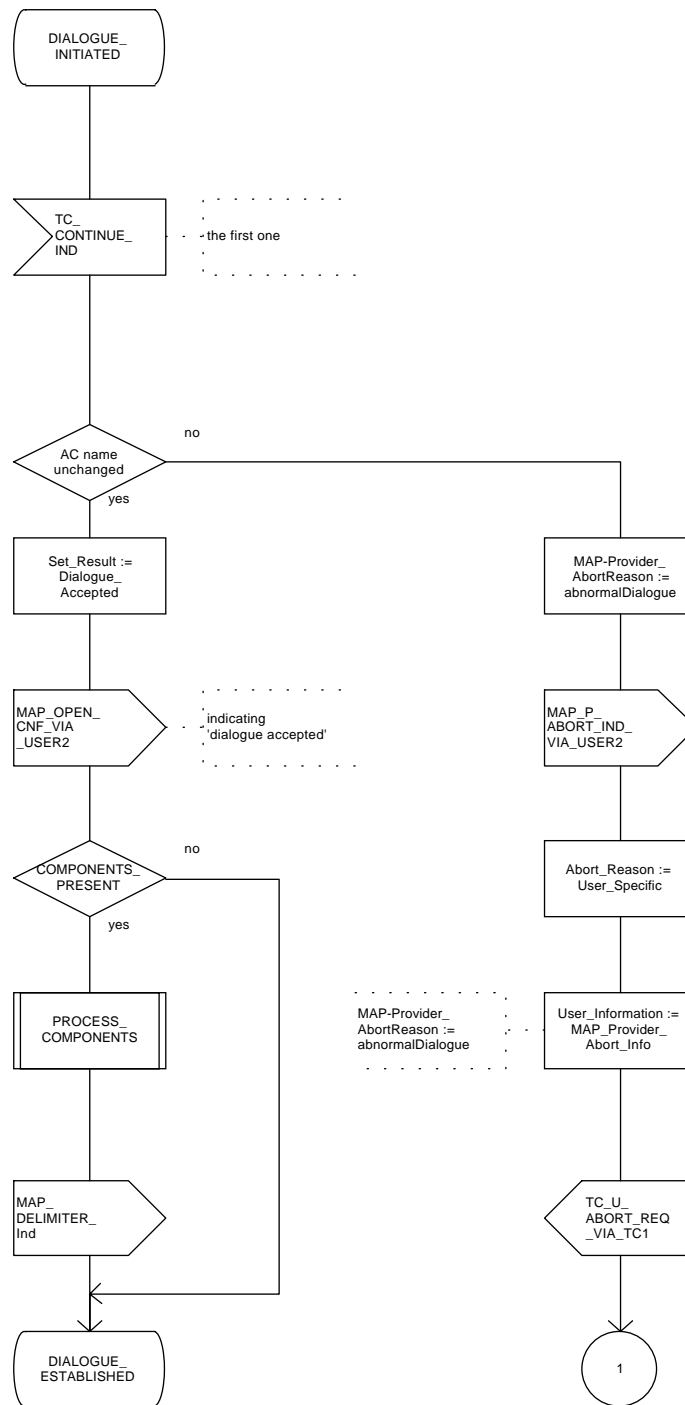
Figure 13.2/3 (sheet 7 of 11): Process MAP_DSM
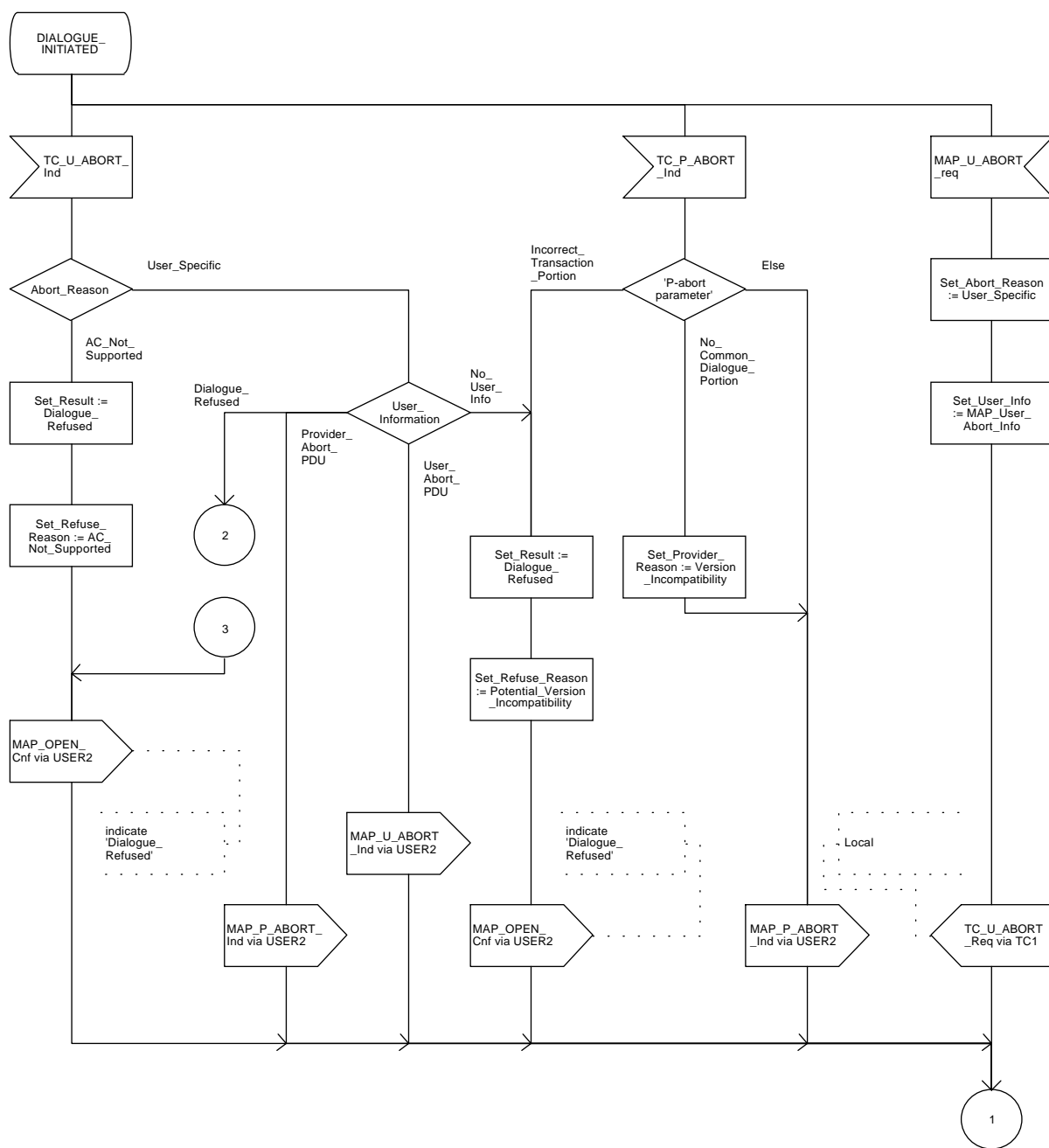
Figure 13.2/3 (sheet 8 of 11): Process MAP_DSM
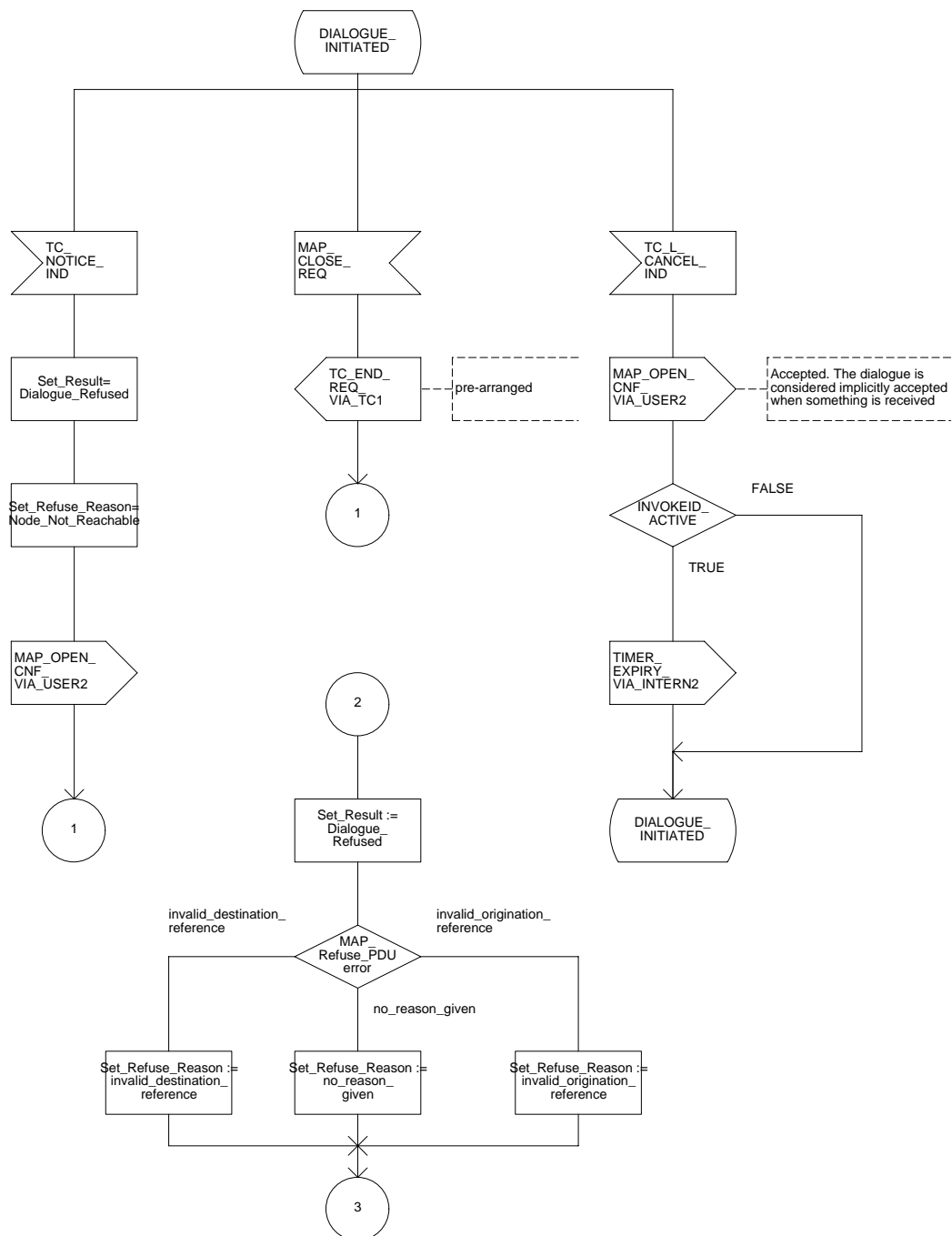
Figure 13.2/3 (sheet 9 of 11): Process MAP_DSM
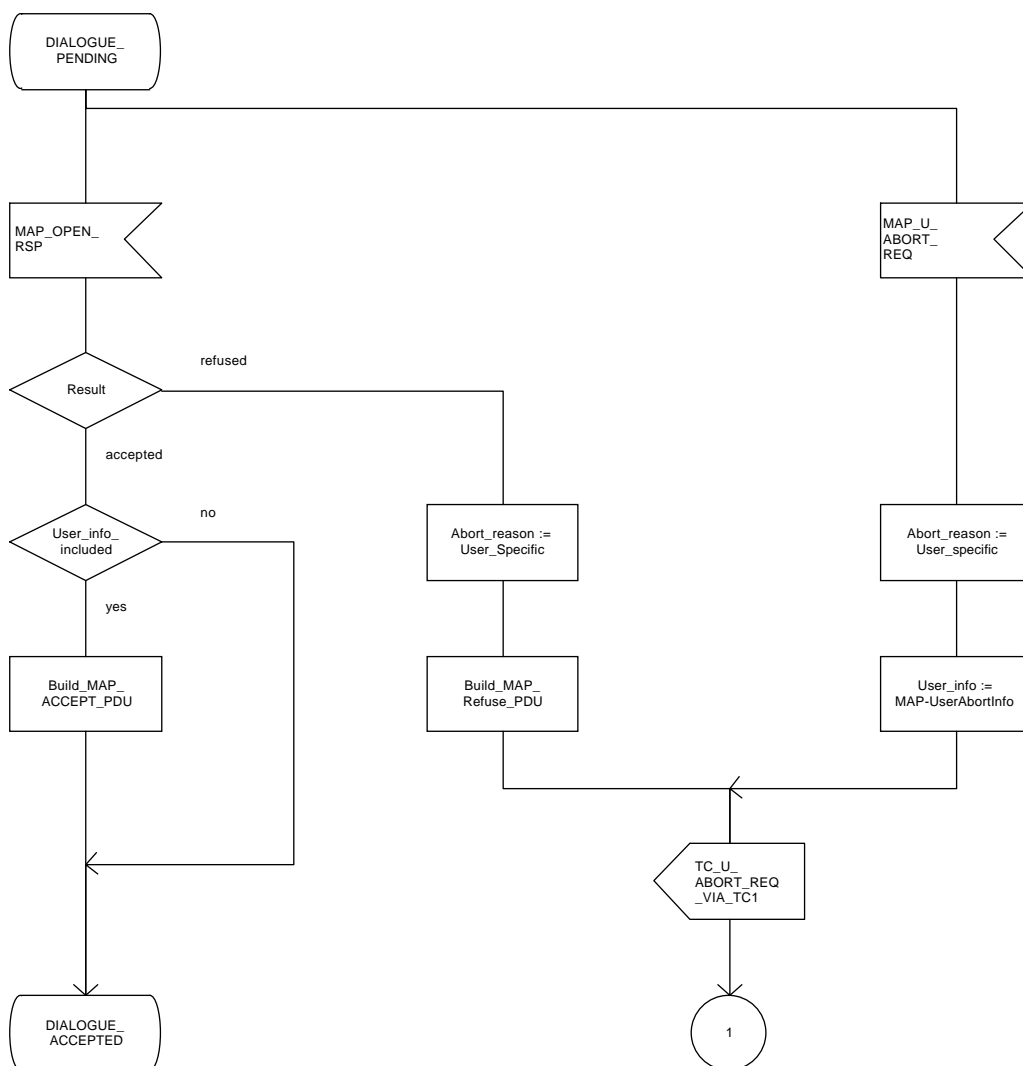
Figure 13.2/3 (sheet 10 of 11): Process MAP_DSM
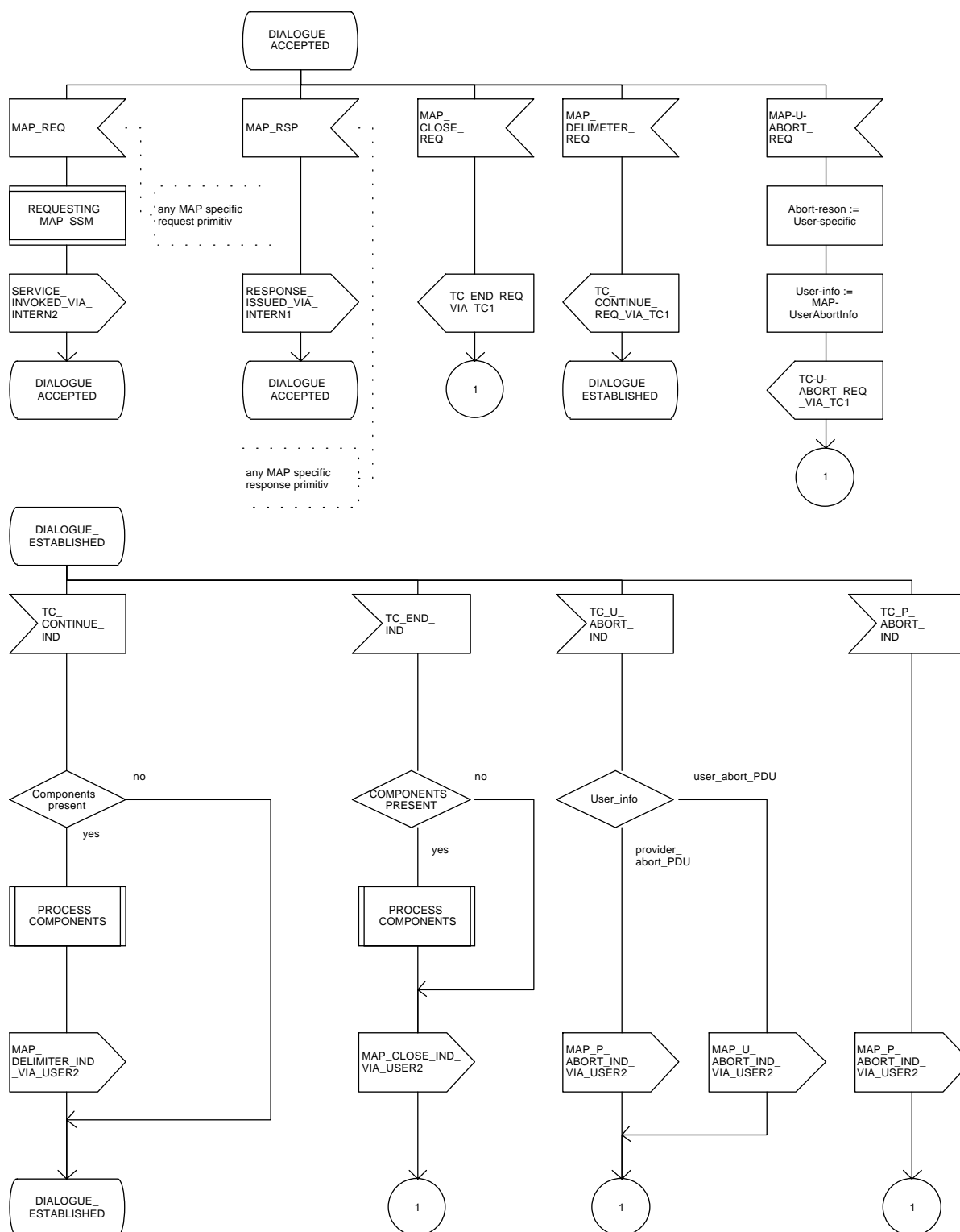
Figure 13.2/3 (sheet 11 of 11): Process MAP_DSM
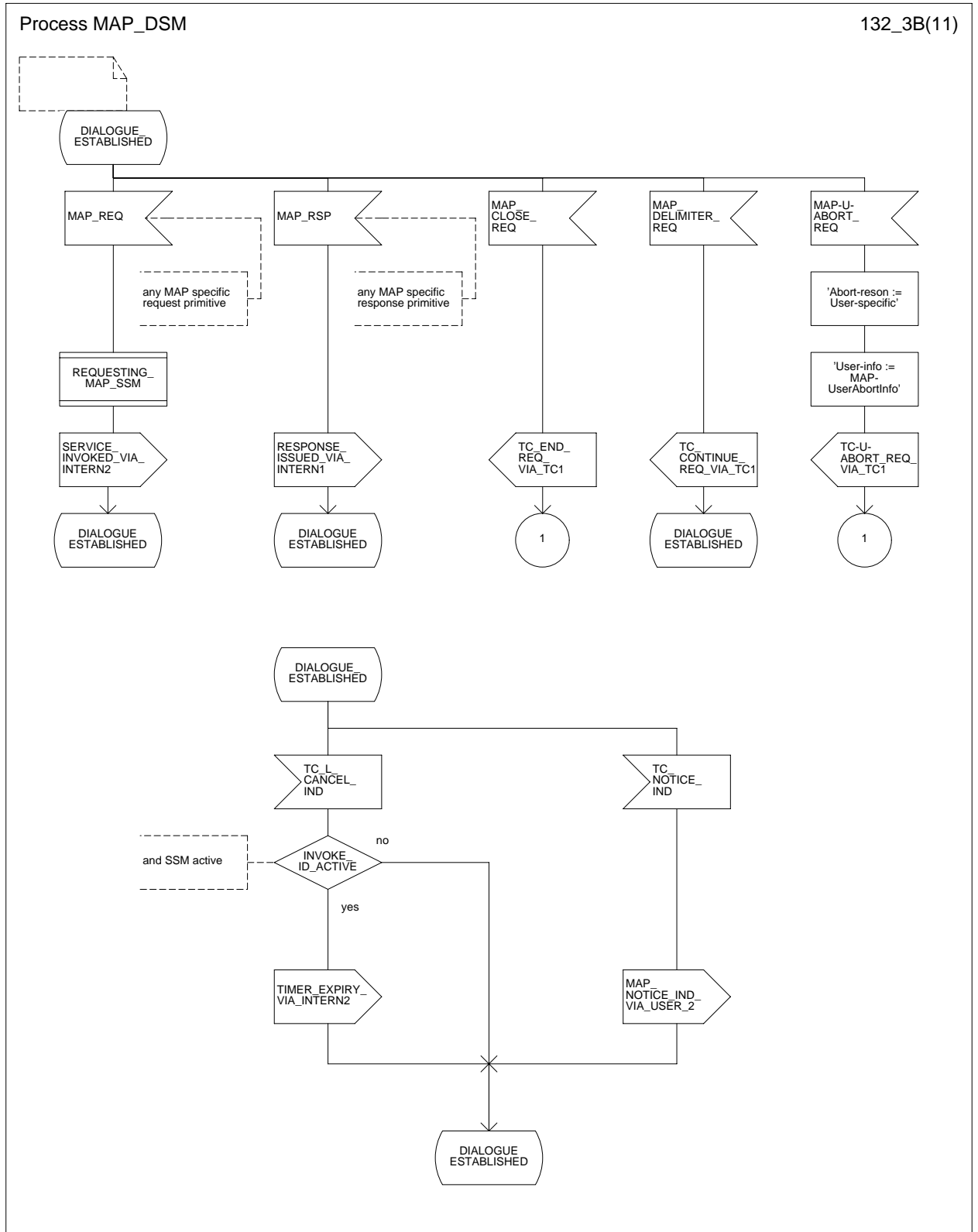
Figure 13.2/4 (sheet 1 of 4): Procedure PROCESS_COMPONENTS

Comments: Components from TCAP:
DCL
  OP_CODE INTERGER,
  OP_EXIST, LAST_COMPONENT, INVOKEID_ASS, LINKEDID_PRES, LINKEDID_ASS BOOLEAN;

1

WAIT_FOR_
COMPONENTS
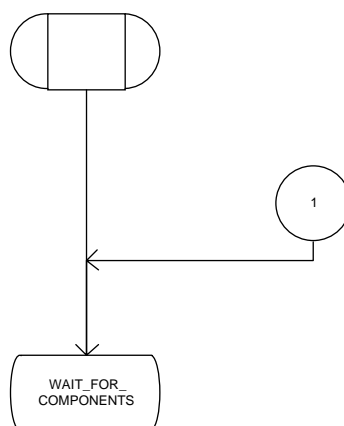
Figure 13.2/4 (sheet 2 of 4): Procedure PROCESS_COMPONENTS
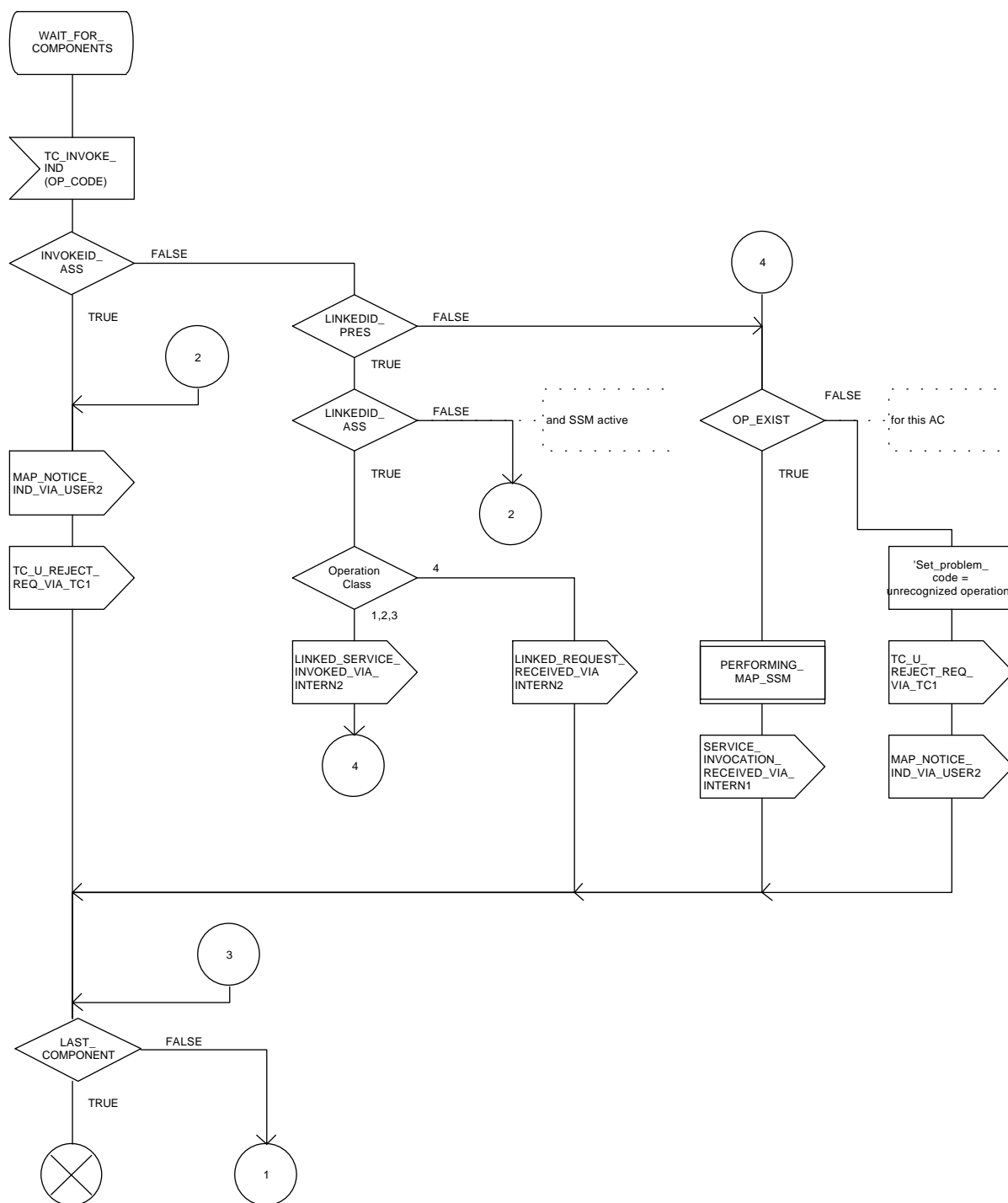
Figure 13.2/4 (sheet 3 of 4): Procedure PROCESS_COMPONENTS
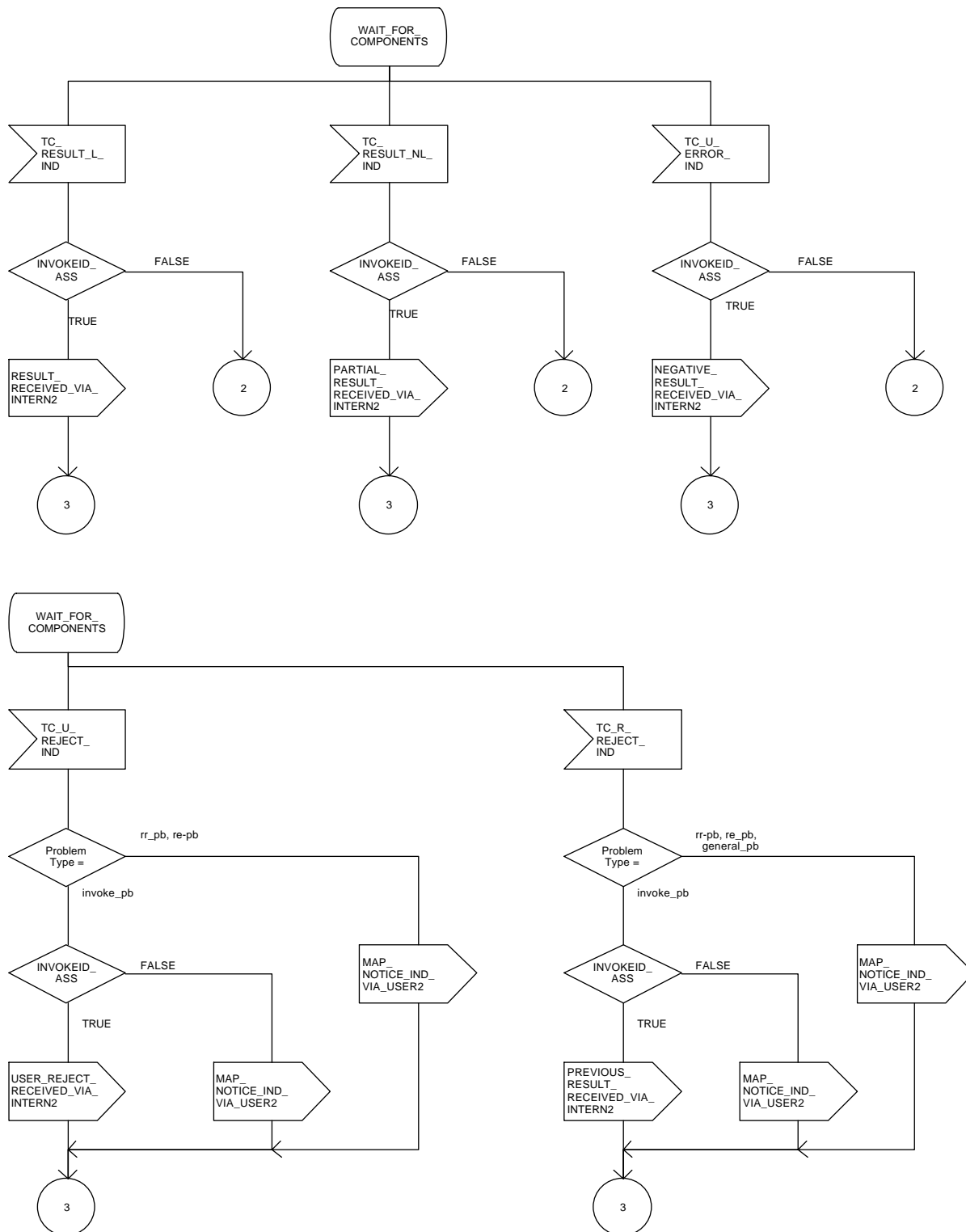
Figure 13.2/4 (sheet 4 of 4): Procedure PROCESS_COMPONENTS
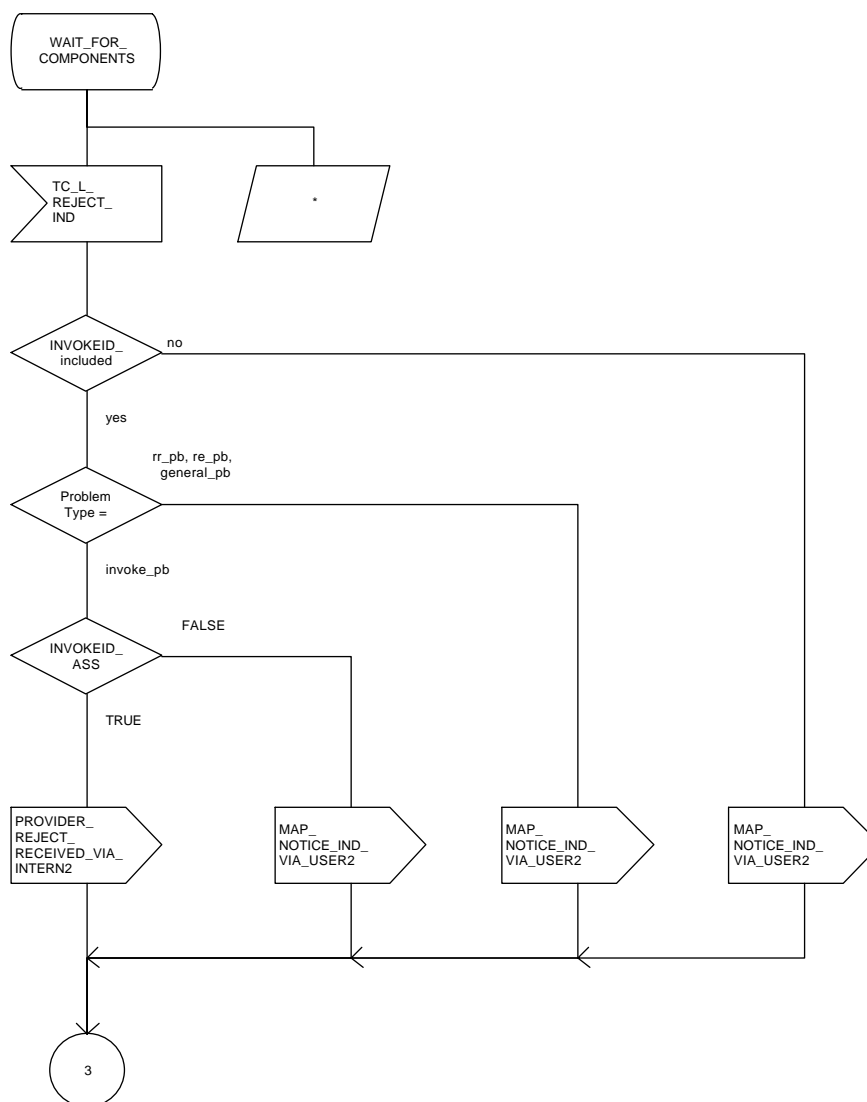
Figure 13.2/5: Process LOAD_CTRL

Comment 'LOAD CONTROL':
DCL
   CONGESTION, DIALOGUE_ACCEPTABLE BOOLEAN;

IDLE

IDLE

IDLE

CHECK_
LOAD

CONGESTION ———— (FALSE)

(TRUE)

'Compare_AC_
priority_with_
load'

LOAD_OK_
VIA_LOAD2

DIALOGUE_
ACCEPTABLE ———— (FALSE)

(TRUE)

LOAD_OK_
VIA_LOAD2

OVERLOAD_
VIA_LOAD2

IDLE
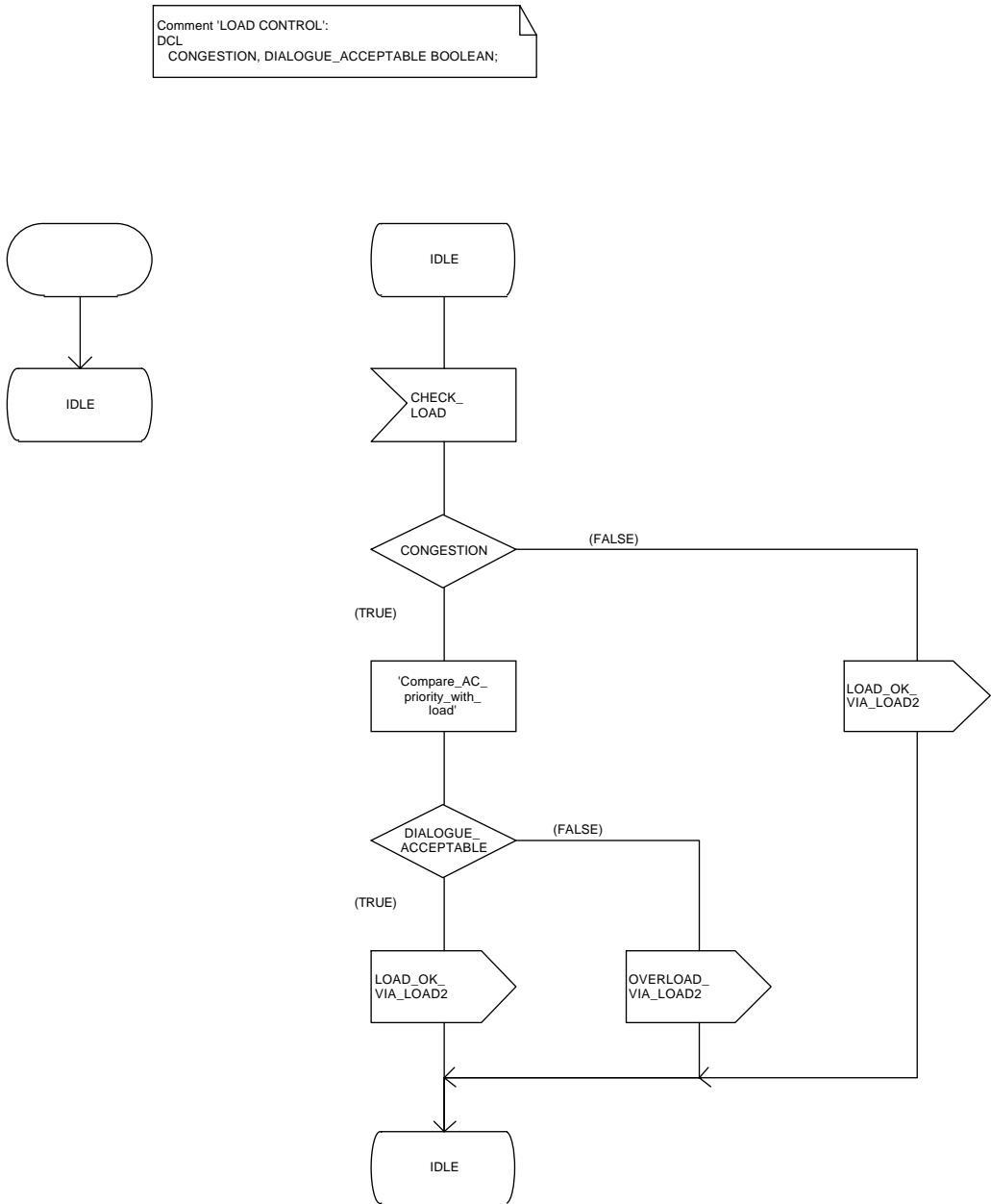
Figure 13.2/6 (sheet 1 of 3): Process PERFORMING_MAP_SSM

Comment 'MAP Service State Machine':
DCL
  ARGUMENT_CORRECT, USER_ERROR_PRESENT,
  SPECIFIC_ERROR_LINKED_REQUEST, CNF BOOLEAN,

  OP_CLASS INTEGER,

  TIMER GUARD_TIMER COMMENT 'expires if MAP user does not respond';

IDLE
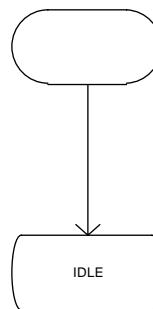
Figure 13.2/6 (sheet 2 of 3): Process PERFORMING_MAP_SSM
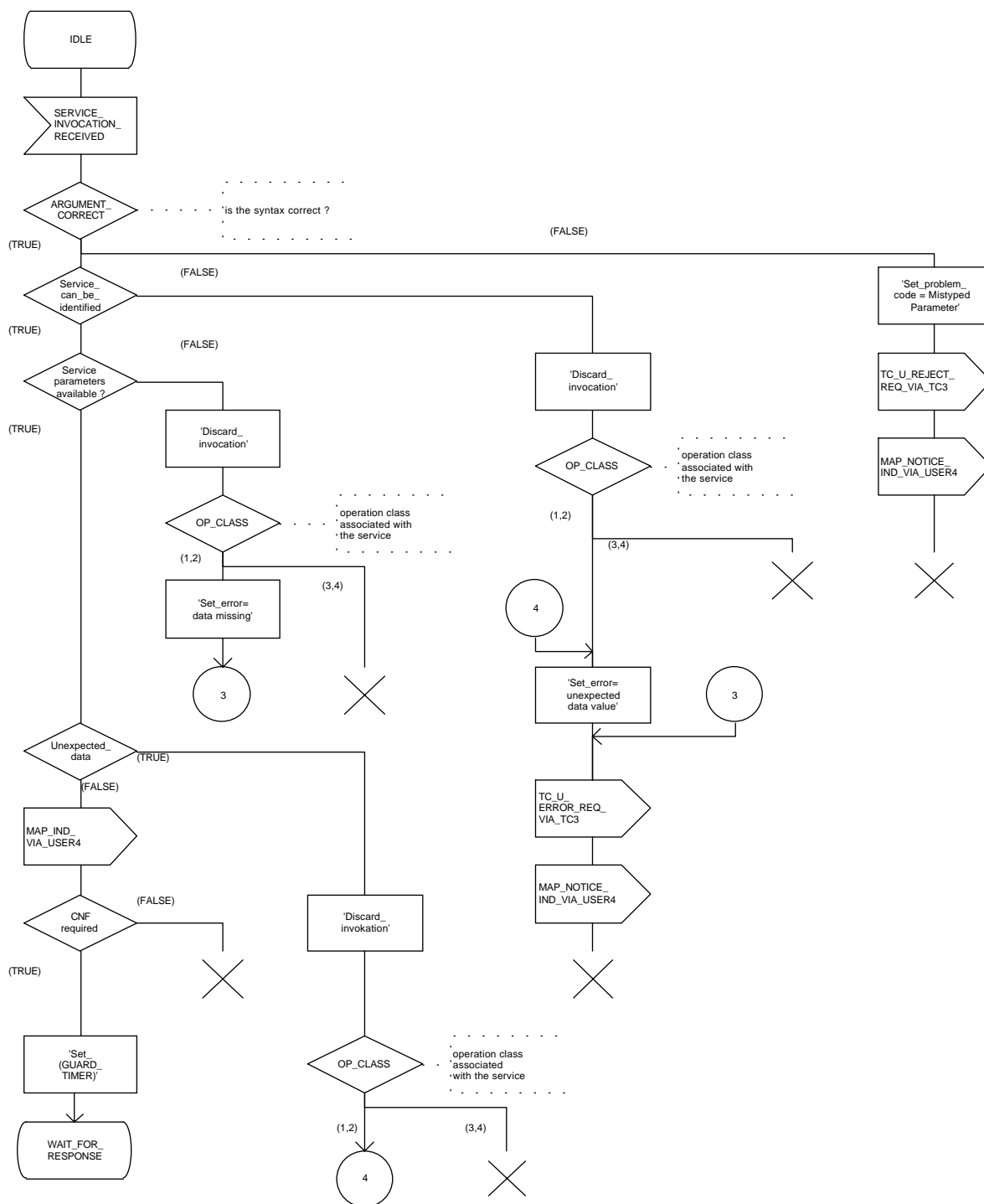
Figure 13.2/6 (sheet 3 of 3): Process PERFORMING_MAP_SSM
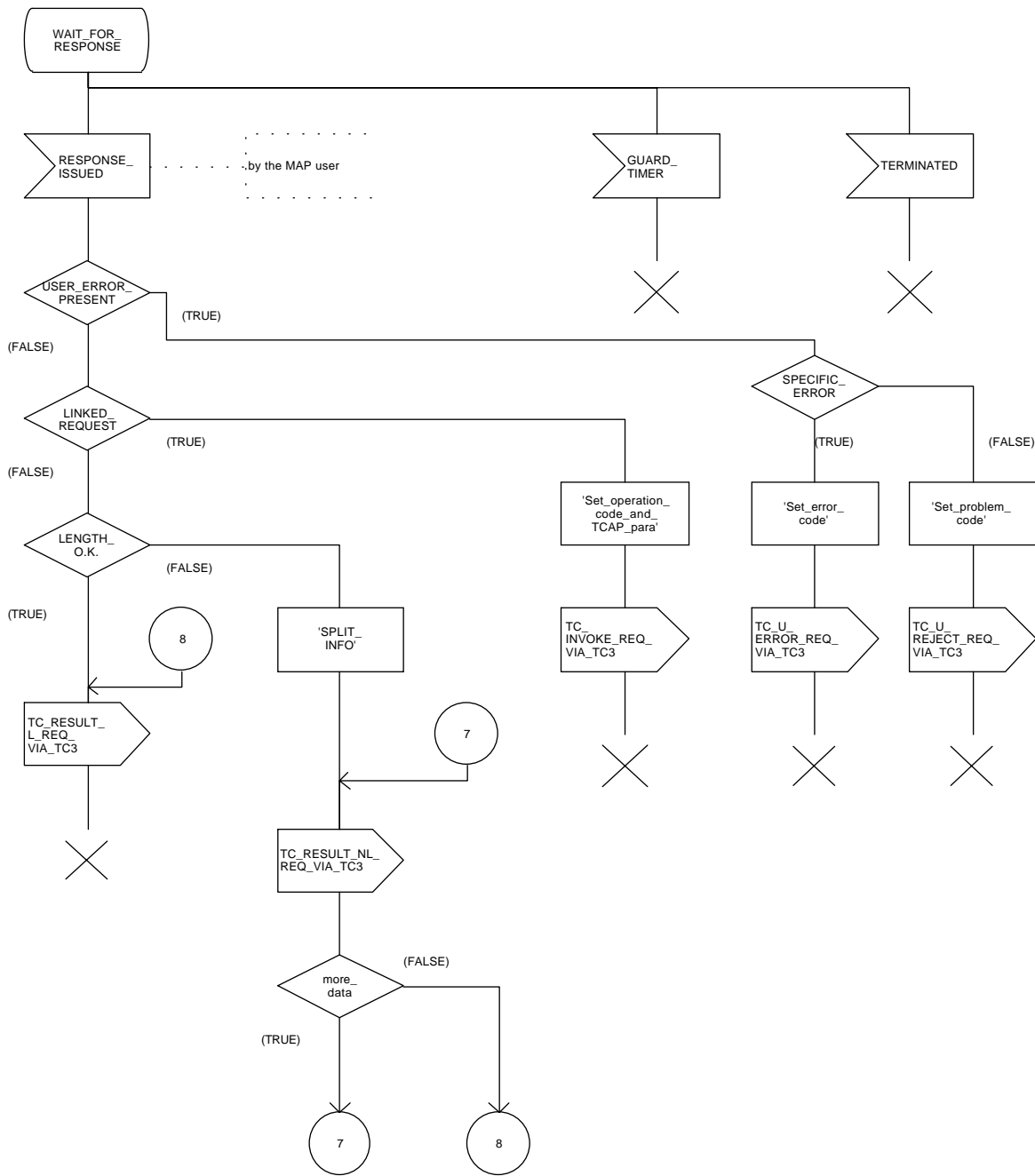
Figure 13.2/7 (sheet 1 of 4): Process REQUESTING_MAP_SSM

Comment 'MAP Service State Maschine':
DCL
  ARGUMENT_CORRECT, ERROR_CODE_CORRECT, LINKED_REQ_DEF, SYNTAX_CORRECT,
  MAP_INITIATED, CNF, LINKED_OPERATION_ALLOWED BOOLEAN,
  OP_CLASS INTEGER;

IDLE

IDLE

SERVICE_
INVOKED

. . . . a service has been invoked
. . . . by the MAP user
. . . .

'Set_Operation__
code_and_TCAP_
parameters'

TC_INVOKE_
REQ_VIA_TC4

Wait_for_
conf ?

(FALSE)

(TRUE)

WAIT_FOR_
CONFIRM
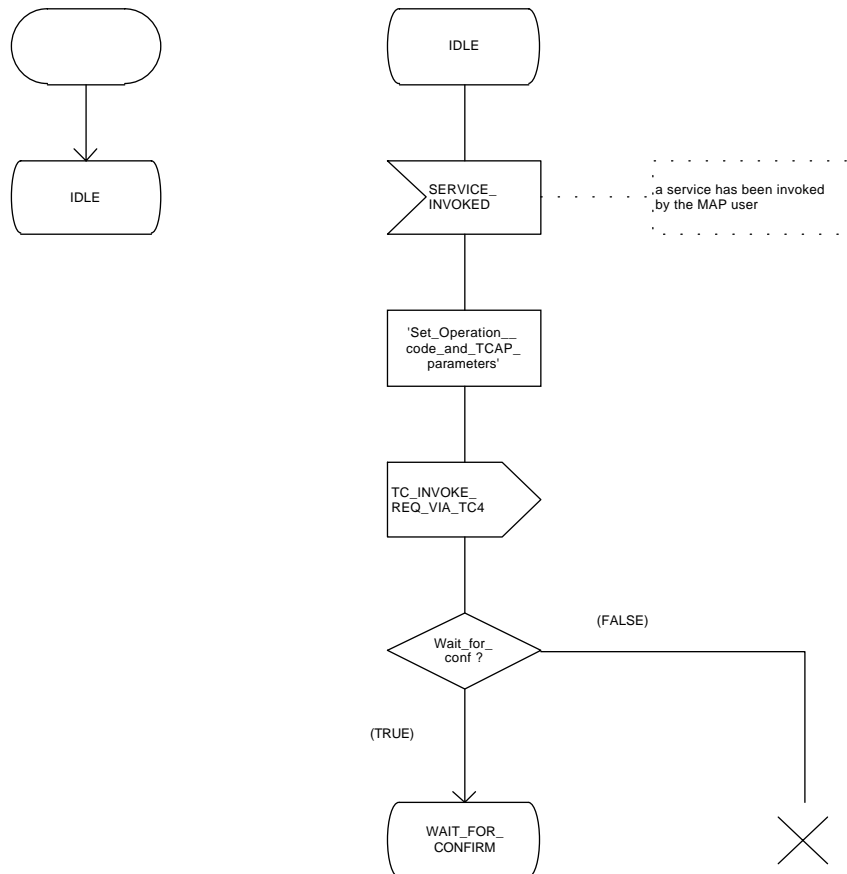
Figure 13.2/7 (sheet 2 of 4): Process REQUESTING_MAP_SSM
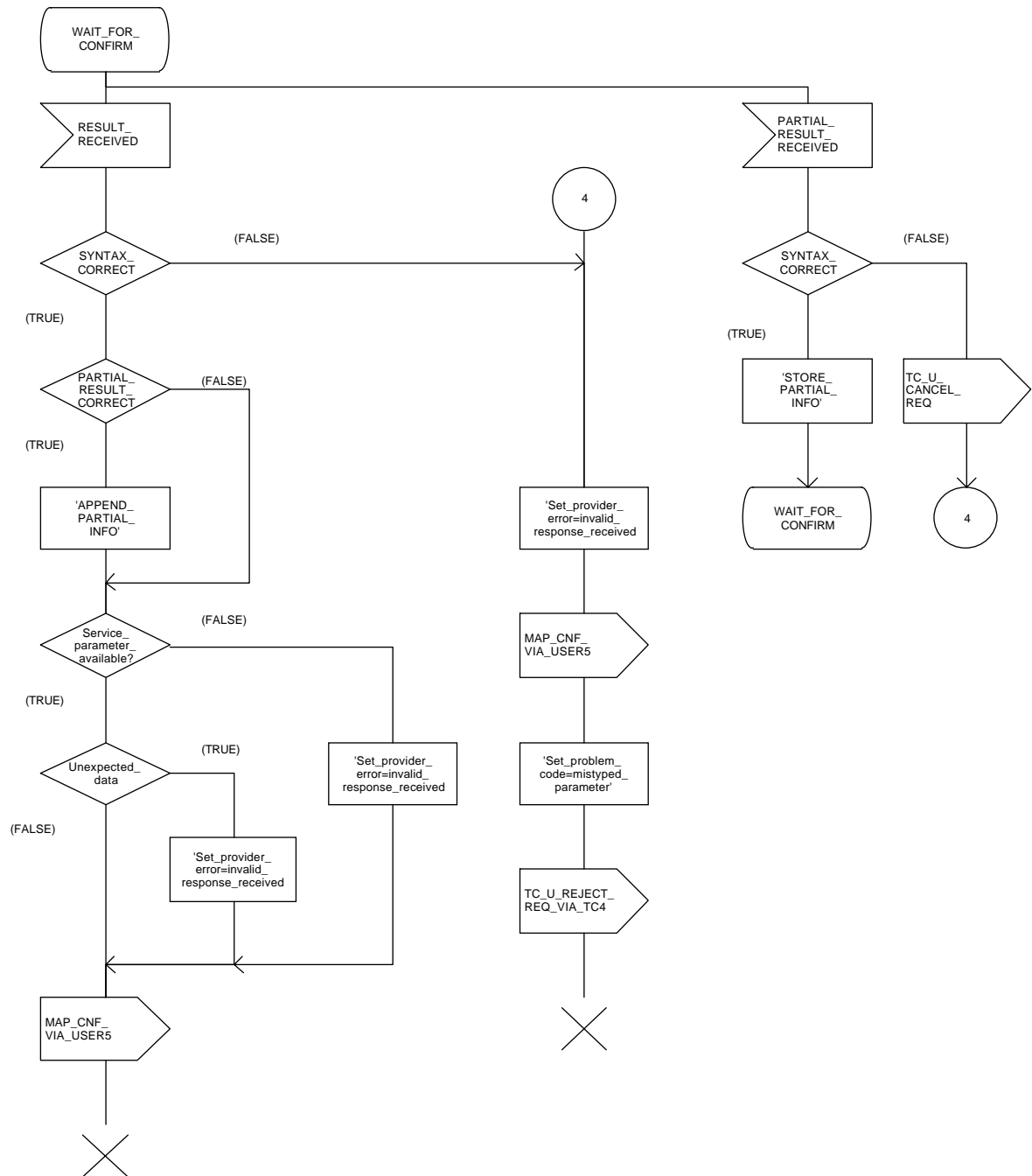
Figure 13.2/7 (sheet 3 of 4): Process REQUESTING_MAP_SSM
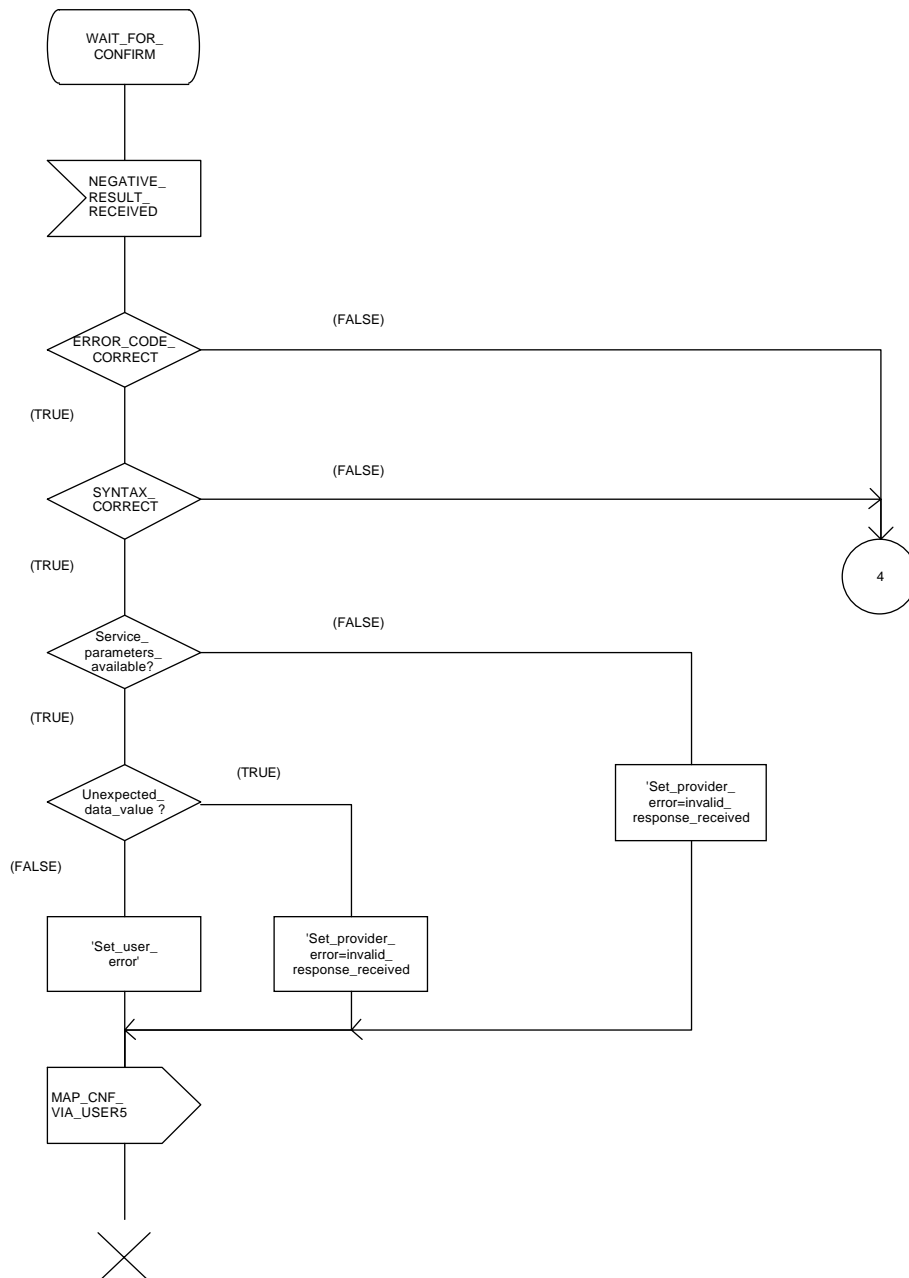
Figure 13.2/7 (sheet 4 of 4): Process REQUESTING_MAP_SSM
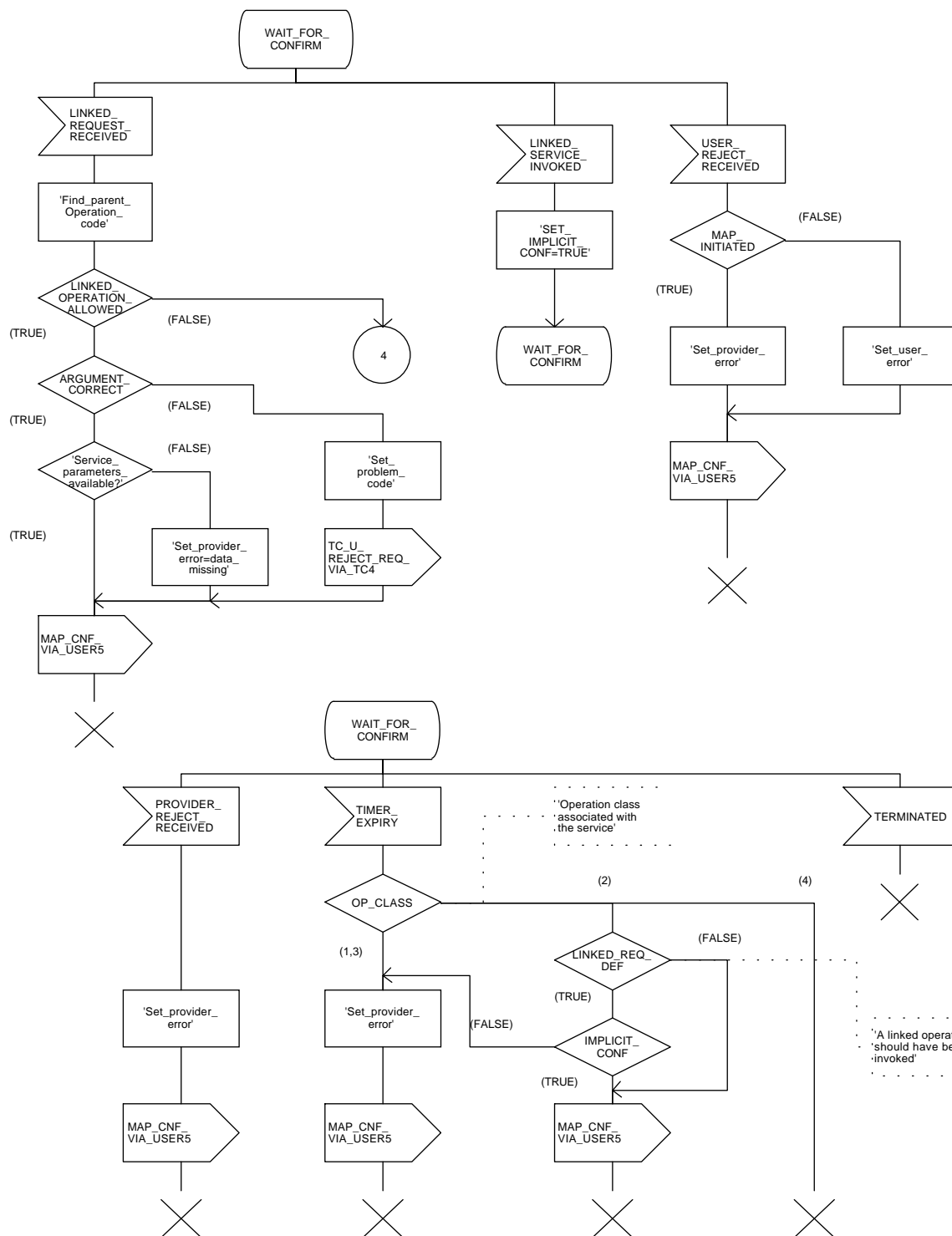
14 Abstract syntax of the MAP protocol

14.1 General

This subclause specifies the Abstract Syntaxes for the Mobile Application Part as well as the associated set of Operations and Errors, using the Abstract Syntax Notation One (ASN.1), defined in CCITT Recommendation X.208 (1988) with additions as defined in subclause 14.1.4 on Compatibility Considerations and the OPERATION and ERROR external MACROs, defined in CCITT Recommendation Q.773.

The Abstract Syntax is defined for all interfaces specified in subclause 2.4 except for the A- and B-interfaces.

The Mobile Application Part protocol is defined by two Abstract Syntaxes:

- one Abstract Syntax which encompass all Operations; and

- Errors identified by the various MAP subsystem numbers.

This Abstract Syntax represents the set of values each of which is a value of the ASN.1 type TCAPMessages. MessageType as defined in CCITT Recommendation Q.773 with the ANY DEFINED BY sections resolved by the operation and error codes included in the ASN.1 module MAP-Protocol. However, only the subset of this abstract syntax which is required by the procedures defined for an entity needs to be supported:

- one Abstract Syntax identified by the OBJECT IDENTIFIER value MAP-DialogueInformation.map-DialogueAS.

This Abstract Syntax represents the set of values each of which is a value of the ASN.1 type MAP-DialogueInformation.MAP-DialoguePDU. Such a value of the ASN.1 single-ASN.1-type element is contained within the user-information element of the TCAPMessages.DialoguePortion ASN.1 type. This Abstract Syntax name is to be used as a direct reference.

14.1.1 Encoding rules

The encoding rules which are applicable to the defined Abstract Syntaxes are the Basic Encoding Rules for Abstract Syntax Notation One, defined in CCITT Recommendation X.209 with the same exceptions as in CCITT Recommendation Q.773 section 4 Message Representation.

When the definite form is used for length encoding, a data value of length less than 128 octets must have the length encoded in the short form.

When the long form is employed to code a length, the minimum number of octets shall be used to code the length field.

OCTET STRING values and BIT STRING values must be encoded in a primitive form.

14.1.2 Use of TC

The mapping of OPERATION and ERROR to TC components is defined in ETS 300 287 (version 2) which is based on CCITT Recommendation Q.773 (1992).

NOTE 1:   The class of an operation is not stated explicitly but is specified as well in the ASN.1 operation type definition.

Class 1: RESULT and ERROR appear in ASN.1 operation type definition.

Class 2: only ERROR appears in ASN.1 operation type definition.

Class 3: only RESULT appears in ASN.1 operation type definition.

Class 4: both RESULT and ERROR do not appear in ASN.1 operation type definition.

The ASN.1 data type which follows the keywords "ARGUMENT", "PARAMETER" or "RESULT" (for OPERATION and ERROR) is always optional from a syntactic point of view. However, except when specifically mentioned, it has to be considered as mandatory from a semantic point of view.

When an optional element is missing in an invoke component or in an inner data structure while it is required by the context, an error component is returned if specified in the operation type; the associated type of error is DataMissing. This holds also when the entire parameter of an invoke component is missing while it is required by the context.

NOTE 2:   When a mandatory element is missing in the parameter or inner data structure of any component, a reject component is returned (if the dialogue still exists). The problem code to be used is "Mistyped parameter".

The Timer Values used in the operation type definitions are indicated as ASN.1 comment. The Timer Value Ranges are:

```
s     = from 3 seconds to 10 seconds;
m     = from 15 seconds to 30 seconds;
ml    = from 1 minute to 10 minutes;
l     = from 28 hours to 38 hours.
```

14.1.3 Use of information elements defined outside MAP

An information element or a set of information elements (messages) transparently carried in the Mobile Application Part but defined in other recommendation/technical specifications are handled in one of the following ways:

i)   The contents of each information element (without the octets encoding the identifier and the length in the recommendation/technical specification where it is defined) is carried as the value of an ASN.1 NamedType derived from the OCTET STRING data type. Additionally, the internal structure may be explained by means of comments. In case of misalignment the referred to recommendation/technical specification takes precedence.

ii)   The complete information element (including the octets encoding the identifier and the length in the recommendation/technical specification where it is defined) or set of information elements and the identity of the associated protocol are carried as the value of the ExternalSignalInfo data type defined in this ETS. Where more than one information element is carried, the information elements are sent contiguously with no filler octets between them.

14.1.4 Compatibility considerations

The following ASN.1 modules are conforming with CCITT Recommendation X.208 (1988), but in addition Ellipsis Notation ("..." - notation) is used as described in ISO 8824-1/PDAM 3.2 wherever future protocol extensions are foreseen.

The "..." construct only applies to a SEQUENCE. An entity supporting a version greater 1 shall not reject unsupported extension following "..." of that SEQUENCE. The Encoding Rules from subclause 14.1.1 apply to every element of the whole Transfer Syntax especially to the ASN.1 type EXTERNAL.

Note that the Ellipses Notation is not supported in a version 1 context. Therefore a SEQUENCE sent in a version 1 context shall not include any element following "...".

NOTE: Every extension has to follow the superset mechanism defined in ETR 060.

Any change other than in the extension area leads to a new Application Context version.

PLMN specific extensions shall follow the extension marker and shall be tagged using PRIVATE tags up to and including 29.

The Abstract Syntaxes of MAP version 2 are in general an extension of the Abstract Syntaxes of MAP version 1. They are therefore (in general) backward compatible with MAP v1 from the Abstract Syntax point of View. Exceptions from this backward compatible changes of the Abstract Syntax were made only for the following reasons:

i)   information element was not used by a GSM phase 1 service due to missing or unstable phase 1 service description (e.g. Closed User Group);

ii)   information element used only on the MSC-VLR Interface (e.g. SendInformationForIncomingCallSetUp);

iii)   changes with impact only in the error situation (e.g. different Cause Values);

iv)   changes on the MAP-AbortInfo.

Changes without impact on the Transfer Syntax are not indicated (e.g. name changes, introduction of extension marker).

Information Elements added in existing Constructs for version 2 or information elements kept only for compatibility with version 1 are indicated by ASN.1 comments These comments state which subset of the abstract syntax must be used in the negotiated application context.

14.1.5   Structure of the Abstract Syntax of MAP

For each MAP parameter which has to be transferred by a MAP Protocol Data Unit (MAP message), there is a PDU field (an ASN.1 NamedType) whose ASN.1 identifier has the same name as the corresponding parameter, except for the differences required by the ASN.1 notation (blanks between words are removed or replaced by hyphen, the first letter of the first word is lower-case and the first letter of the following words are capitalized, e.g. "no reply condition time" is mapped to "noReplyConditionTime"). Additionally some words may be abbreviated as follows:

bs	basic service
ch	call handling
cug	closed user group
ho	handover
ic	incoming call
id	identity
info	information
ms	mobile service
oc	outgoing call
om	operation & maintenance
pw	Password
sm	short message service
ss	supplementary service

The MAP protocol is composed of several ASN.1 modules dealing with either operations, errors, data types, and, if applicable, split into those dealing with mobile services, call handling services, supplementary services and short message services. For operations and errors no values are assigned, but only the operation and error types in order to allow use of the defined types also by other protocols (e.g. GSM 04.80). The values (operation codes and error codes) are defined in a separate module. The ASN.1 source lines are preceded by line-numbers at the left margin in order to enable the usage of the cross-reference in appendix A.

The module containing the definition of the operation packages for MAP is:

> 1.   MAP-OperationPackages.

The module containing the definition of the application contexts for MAP is:

> 2.   MAP-ApplicationContexts.

The module containing the data types for the Abstract Syntax to be used for TCAPMessages.DialoguePortion for MAP is:

> 3.   MAP-DialogueInformation.

The module containing the operation codes and error codes for MAP is:

> 4.   MAP-Protocol.

The modules containing all operation type definitions for MAP are:

> 5.   MAP-MobileServiceOperations;
> 6.   MAP-OperationAndMaintenanceOperations;
> 7.   MAP-CallHandlingOperations;
> 8.   MAP-SupplementaryServiceOperations;
> 9.   MAP-ShortMessageServiceOperations.

The module containing all error type definitions for MAP is:

    10.   MAP-Errors.

Modules containing all data type definitions for MAP are:

    11.   MAP-MS-DataTypes;
    12.   MAP-OM-DataTypes;
    13.   MAP-CH-DataTypes;
    14.   MAP-SS-DataTypes;
    15.   MAP-SS-Code;
    16.   MAP-SM-DataTypes;
    17.   MAP-ER-DataTypes;
    18.   MAP-CommonDataTypes;
    19.   MAP-TS-Code;
    20.   MAP-BS-Code.

References are made also to modules defined outside of this ETS. They are defined in the technical specification Mobile Services Domain and technical specification Transaction Capability respectively:

    MobileDomainDefinitions;
    TCAPMessages;
    DialoguePDUs.

14.2 Operation packages

14.2.1 General aspects

This subclause describes the operation-packages which are used to build the application-contexts defined in subclause 14.3.

Each operation-package is a specification of the roles of a pair of communicating objects (i.e. a pair of MAP-Providers), in term of operations which they can invoke of each other.

The grouping of operations into one or several packages does not necessarily imply any grouping in term of Application Service Elements.

The following ASN.1 MACRO is used to describe operation-packages in this subclause:

```
OPERATION-PACKAGE MACRO ::=

BEGIN

TYPE NOTATION ::= Symmetric | ConsumerInvokes SupplierInvokes |
empty

VALUE NOTATION ::= value(VALUE OBJECT IDENTIFIER)

Symmetric ::= "OPERATIONS" "{" OperationList "}"

ConsumerInvokes ::= "CONSUMER INVOKES" "{" OperationList "}"

SupplierInvokes ::= "SUPPLIER INVOKES" "{" OperationList "}" | empty

OperationList ::= Operation | OperationList "," Operation

Operation ::= value(OPERATION)

END
```

Since the application-context definitions provided in subclause 14.3 uses only an informal description technique, only the type notation is used in the following subclauses to define operation-packages.

The following definitions are used throughout this subclause:

- v1-only operation: An operation which shall only be used in v1 application-contexts;

- v2-only operation: An operation which shall only be used in v2 application-contexts;

- v2-only package: An operation package which contains solely v2-only operations;

- v1-operation: An operation whose specification has not been modified since the MAP v1 specifications or if the modifications are not considered as affecting V1 implementations;

- v1-equivalent operation: The version of an operation which excludes all the information elements and errors which has been added since MAP v1 specifications;

- v1-package: An operation package which contains only v1- operations.

The names of v1-packages are suffixed by "-v1" while other names are suffixed by "-v2".

For each operation package which is not v2-only and which does not include only v1-operations, there is a v1-equivalent package. Except when a definition is explicitly provided in the following subclauses, the v1-equivalent package includes the v1-equivalent operations of the operations which are belong to this package.

14.2.2 Packages specifications

14.2.2.1 Location updating

This operation package includes the operations required for location management procedures between HLR and VLR.

```
LocationUpdatingPackage-v2 ::= OPERATION-PACKAGE
    -- Supplier is HLR if Consumer is VLR
    CONSUMER INVOKES {
        updateLocation}
    SUPPLIER INVOKES {
        forwardCheckSs-Indication}
```

The v1-equivalent package can be determined according to the rules described in subclause 14.2.1.

14.2.2.2 Location cancellation

This operation package includes the operations required for location cancellation and MS purging procedures between HLR and VLR.

```
LocationCancellationPackage-v2 ::= OPERATION-PACKAGE
    -- Supplier is VLR if Consumer is HLR
    CONSUMER INVOKES {
        cancelLocation}
```

The v1-equivalent package can be determined according to the rules described in subclause 14.2.1.

14.2.2.3 Roaming number enquiry

This operation package includes the operations required for roaming number enquiry procedures between HLR and VLR.

```
RoamingNumberEnquiryPackage-v2 ::= OPERATION-PACKAGE
    -- Supplier is VLR if Consumer is HLR
    CONSUMER INVOKES {
        provideRoamingNumber}
```

The v1-equivalent package can be determined according to the rules described in subclause 14.2.1.

14.2.2.4   Information retrieval

This operation package includes the operation required for the authentication information retrieval procedure between HLR and VLR.

```
InfoRetrievalPackage-v2 ::= OPERATION-PACKAGE
    -- Supplier is HLR if Consumer is VLR
    CONSUMER INVOKES {
        sendAuthenticationInfo}
```

The v1-equivalent package is defined as follows:

```
InfoRetrievalPackage-v1 ::= OPERATION-PACKAGE
    -- Supplier is HLR or VLR if Consumer is VLR
    CONSUMER INVOKES {
        sendParameters}
```

14.2.2.5   Inter-VLR information retrieval

This operation package includes the operations required for inter VLR information retrieval procedures.

```
InterVlrInfoRetrievalPackage-v2 ::= OPERATION-PACKAGE
    -- Supplier is VLR if Consumer is VLR
    CONSUMER INVOKES {
        sendIdentification}
```

The v1-equivalent package is : InfoRetrievalPackage-v1

14.2.2.6   IMSI retrieval

This operation package includes the operation required for the IMSI retrieval procedure between HLR and VLR.

```
IMSIRetrievalPackage-v2 ::= OPERATION-PACKAGE
    -- Supplier is HLR if Consumer is VLR
    CONSUMER INVOKES {
        sendIMSI}
```

This package is v2 only.

14.2.2.7 - 14.2.2.9 [spare]

14.2.2.10   Interrogation

This operation package includes the operations required for interrogation procedures between MSC and HLR.

```
InterrogationPackage-v2 ::= OPERATION-PACKAGE
    -- Supplier is HLR if Consumer is MSC
    CONSUMER INVOKES {
        sendRoutingInfo}
```

The v1-equivalent package can be determined according to the rules described in subclause 14.2.1.

14.2.2.11   [spare]

14.2.2.12 Handover Control

This operation package includes the operations required for handover procedures between MSCs.

```
HandoverControlPackage-v2 ::= OPERATION-PACKAGE
    -- Supplier is MSCB if Consumer is MSCA
    CONSUMER INVOKES {
        prepareHandover,
        forwardAccessSignalling}
    SUPPLIER INVOKES {
        sendEndSignal,
        processAccessSignalling,
        prepareSubsequentHandover}
```

The v1-equivalent package is defined as follows.

```
HandoverControlPackage-v1 ::= OPERATION-PACKAGE
    -- Supplier is MSCB if Consumer is MSCA
    CONSUMER INVOKES {
        performHandover,
        forwardAccessSignalling,
        traceSubscriberActivity}
    SUPPLIER INVOKES {
        sendEndSignal,
        noteInternalHandover,
        processAccessSignalling,
        performSubsequentHandover}
```

14.2.2.13 Subscriber Data management stand alone

This operation package includes the operations required for stand alone subscriber data management procedures between HLR and VLR.

```
SubscriberDataMngtStandAlonePackage-v2 ::= OPERATION-PACKAGE
    -- Supplier is VLR if Consumer is HLR
    CONSUMER INVOKES {
        insertSubscriberData,
        deleteSubscriberData}
```

The v1-equivalent package can be determined according to the rules described in subclause 14.2.1.

14.2.2.14 Equipment management

This operation package includes the operations required for equipment management procedures between EIR and MSC.

```
EquipmentMngtPackage-v2 ::= OPERATION-PACKAGE
    -- Supplier is EIR if Consumer is MSC
    CONSUMER INVOKES {
        checkIMEI}
```

The v1-equivalent package can be determined according to the rules described in subclause 14.2.1.

14.2.2.15 Subscriber data management

This operation package includes the operations required for subscriber data management procedures between HLR and VLR.

```
SubscriberDataMngtPackage-v2 ::= OPERATION-PACKAGE
    -- Supplier is VLR if Consumer is HLR
    CONSUMER INVOKES {
        insertSubscriberData}
```

The v1-equivalent package can be determined according to the rules described in subclause 14.2.1.

14.2.2.16 Location register restart

This operation package includes the operations required for location register restart procedures between HLR and VLR.

```
ResetPackage-v2 ::= OPERATION-PACKAGE
    -- Supplier is VLR if Consumer is HLR
    CONSUMER INVOKES {
        reset}
```

The v1-equivalent package can be determined according to the rules described in subclause 14.2.1.

14.2.2.17 Tracing stand-alone

This operation package includes the operations required for stand alone tracing procedures between HLR and VLR.

```
TracingStandAlonePackage-v2 ::= OPERATION-PACKAGE
    -- Supplier is VLR if Consumer is HLR
    CONSUMER INVOKES {
        activateTraceMode,
        deactivateTraceMode}
```

The v1-equivalent package can be determined according to the rules described in subclause 14.2.1.

14.2.2.18 Functional SS handling

This operation package includes the operations required for functional supplementary services procedures between VLR and HLR.

```
FunctionalSsPackage-v2 ::= OPERATION-PACKAGE
    -- Supplier is HLR if Consumer is VLR
    CONSUMER INVOKES {
        registerSS,
        eraseSS,
        activateSS,
        deactivateSS,
        registerPassword,
        interrogateSS}
    SUPPLIER INVOKES {
        getPassword}
```

The v1-equivalent package can be determined according to the rules described in subclause 14.2.1.

14.2.2.19 Tracing

This operation package includes the operations required for tracing procedures between HLR and VLR.

```
TracingPackage-v2 ::= OPERATION-PACKAGE
    -- Supplier is VLR if Consumer is HLR
    CONSUMER INVOKES {
        activateTraceMode}
```

The v1-equivalent package can be determined according to the rules described in subclause 14.2.1.

14.2.2.20 Binding

This operation package includes the operation required to initialize a supplementary service procedure between VLR and HLR.

```
BindingPackage-v1 ::= OPERATION-PACKAGE
    -- Supplier is HLR if Consumer is VLR
    CONSUMER INVOKES {
        beginSubscriberActivity}
```

This package is v1 only.

14.2.2.21 Unstructured SS handling

This operation package includes the operations required for unstructured supplementary services procedures between VLR and HLR.

```
UnstructuredSsPackage-v2 ::= OPERATION-PACKAGE
    -- Supplier is HLR if Consumer is VLR
    CONSUMER INVOKES {
        processUnstructuredSS-Request}
    SUPPLIER INVOKES {
        unstructuredSS-Request,
        unstructuredSS-Notify}
```

The v1-equivalent package is defined as follows:

```
UnstructuredSsPackage-v1 ::= OPERATION-PACKAGE
    -- Supplier is HLR if Consumer is VLR
    CONSUMER INVOKES {
        processUnstructuredSS-Data}
```

14.2.2.22 Short message relay services

This operation package includes the operations required for short message relay service procedures between IWMSC and VMSC or between GMSC and MSC.

```
ShortMsgRelayPackage-v2 ::= OPERATION-PACKAGE
    -- Supplier is IWMSC if Consumer is MSC
    -- Supplier is MSC if Consumer is GMSC
    CONSUMER INVOKES {
        forwardSM}
```

The v1-equivalent package can be determined according to the rules described in subclause 14.2.1.

14.2.2.23 Short message gateway services

This operation package includes the operations required for short message service gateway procedures between MSC and HLR.

```
ShortMsgGatewayPackage-v2 ::= OPERATION-PACKAGE
    -- Supplier is HLR if Consumer is GMSC
    CONSUMER INVOKES {
        sendRoutingInfoForSM,
        reportSM-DeliveryStatus}
    SUPPLIER INVOKES {
        informServiceCentre}
```

The v1-equivalent package is defined as follows:

```
ShortMsgGatewayPackage-v1 ::= OPERATION-PACKAGE
    -- Supplier is HLR if Consumer is GMSC
    CONSUMER INVOKES {
        sendRoutingInfoForSM
        reportSMDeliveryStatus}
```

14.2.2.24 - 14.2.2.25 [spare]

14.2.2.26 Message waiting data management

This operation package includes the operations required for short message waiting data procedures between HLR and VLR.

```
MwdMngtPackage-v2 ::= OPERATION-PACKAGE
    -- Supplier is HLR if Consumer is VLR
    CONSUMER INVOKES {
        readyForSM}
```

The v1-equivalent package is defined as follows:

```
MwdMngtPackage-v1 ::= OPERATION-PACKAGE
    -- Supplier is HLR if Consumer is VLR
    CONSUMER INVOKES {
        noteSubscriberPresent}
```

14.2.2.27   Alerting

This operation package includes the operations required for alerting between HLR and IWMSC.

```
AlertingPackage-v2 ::= OPERATION-PACKAGE
    -- Supplier is IWMSC if Consumer is HLR
    CONSUMER INVOKES {
        alertServiceCentre}
```

The v1-equivalent package is defined as follows.

```
AlertingPackage-v1 ::= OPERATION-PACKAGE
    -- Supplier is IWMSC if Consumer is HLR
    CONSUMER INVOKES {
        alertServiceCentreWithoutResult}
```

14.2.2.28   Data restoration

This operation package includes the operations required for VLR data restoration between HLR and VLR.

```
DataRestorationPackage-v2 ::= OPERATION-PACKAGE
    -- Supplier is HLR if Consumer is VLR
    CONSUMER INVOKES {
        restoreData}
```

The v1-equivalent package is: InfoRetrievalPackage-v1

14.2.2.29   Purging

This operation package includes the operations required for purging between HLR and VLR.

```
PurgingPackage-v2 ::= OPERATION-PACKAGE
    -- Supplier is HLR if Consumer is VLR
    CONSUMER INVOKES {
        purgeMS}
```

This Package is v2 only.

14.3 Application contexts

14.3.1 General aspects

To each dialogue established by a MAP-user is associated an application-context. In this ETS each application-context is assigned a name which is supplied in the MAP-OPEN Req primitive by the MAP-User and transmitted to the peer under certain circumstances.

The following ASN.1 MACRO is used to describe the main aspects of application-contexts in the following subclauses:

```
APPLICATION-CONTEXT MACRO ::=

BEGIN

TYPE NOTATION ::= Symmetric | InitiatorConsumerOf
ResponderConsumerOf | empty

VALUE NOTATION ::= value(VALUE OBJECT IDENTIFIER)

Symmetric ::= "OPERATIONS OF" "{" PackageList "}"

InitiatorConsumerOf ::= "INITIATOR CONSUMER OF" "{" PackageList "}"

ResponderConsumerOf ::= "RESPONDER CONSUMER OF" "{" PackageList "}"
        | empty

PackageList ::= Package | PackageList "," Package

Package ::= value(OPERATION-PACKAGE)
        | type -- shall reference a package type

END
```

The following definitions are used throughout this subclause:

- v1-application-context: An application-context which contains only v1-packages and uses only TC v1 facilities;

- v1-compatible application-context: An application-context which is a v1-application-context or which uses TC V2 facilities with only v1-packages;

- v1 context set: the set of v1-application-contexts defined in this ETS.

The name of v1-application-contexts are suffixed by "-v1" while other names are suffixed by "-v2".

Application-contexts which do not belong to the v1 context set uses v2 TC facilities.

The last component of the application-context-name (i.e. the last component of the object identifier value) assigned to application-context which belong to the v1 context set indicates explicitly "version1".

For each application-context which does not belong to the "v1 context set" there is v1-equivalent application context. This is a v1-application-context which includes the v1-equivalents of the packages included in the original context.

Each application-context uses the abstract-syntax associated with the operation-packages it includes and uses the transfer-syntax derived from it by applying the encoding rules defined in subclause 14.1.1.

Ac which do not belong to the v1 context set require the support of the abstract-syntax identified by the following object identifier value: MAP-DialogueInformation.map-Dialogue-AS defined in subclause 14.4.

14.3.2 Access-context definitions

14.3.2.1 [spare]

14.3.2.2 Location Updating

This application context is used between HLR and VLR for location updating procedures.

```
networkLocUpContext-v2 APPLICATION-CONTEXT
    -- Responder is HLR if Initiator is VLR
    INITIATOR CONSUMER OF {
        LocationUpdatingPackage-v2,
        DataRestorationPackage-v2}
    RESPONDER CONSUMER OF {
        SubscriberDataMngtPackage-v2
        TracingPackage-v2}
::= {map-ac networkLocUp(1) version2(2)}
```

The following application-context-name is assigned to the v1-equivalent application-context:

```
{map-ac networkLocUp(1) version1(1)}
```

14.3.2.3 Location Cancellation

This application context is used between HLR and VLR for location cancellation procedures.

```
locationCancellation-v2 APPLICATION-CONTEXT
    -- Responder is VLR if Initiator is HLR
    INITIATOR CONSUMER OF {
        LocationCancellationPackage-v2}
::= {map-ac locationCancel(2) version2(2)}
```

The following application-context-name is assigned to the v1-equivalent application-context:

```
map-ac locationCancel(2) version1(1)
```

14.3.2.4 Roaming number enquiry

This application context is used between HLR and VLR for roaming number enquiry procedures.

```
roamingNumberEnquiryContext-v2 APPLICATION-CONTEXT
    -- Responder is VLR if Initiator is HLR
    INITIATOR CONSUMER OF {
        RoamingNumberEnquiryPackage-v2}
::= {map-ac roamingNbEnquiry(3) version2(2)}
```

The following application-context-name is assigned to the v1-equivalent application-context:

```
{map-ac roamingNbEnquiry(3) version1(1)}
```

14.3.2.5 [spare]

14.3.2.6 Location Information Retrieval

This application-context is used between GMSC and HLR when retrieving location information.

```
locationInfoRetrievalContext-v2 APPLICATION-CONTEXT
    -- Responder is HLR if Initiator is GMSC
    INITIATOR CONSUMER OF {
        InterrogationPackage-v2}
::= {map-ac locInfoRetrieval(5) version2(2)}
```

The following application-context-name is assigned to the v1-equivalent application-context:

```
{map-ac locInfoRetrieval(5) version1(1)}
```

14.3.2.7 - 14.3.2.10 [spare]

14.3.2.11 Location registers restart

This application context is used for location register restart procedures.

```
resetContext-v2 APPLICATION-CONTEXT
    -- Responder is VLR if Initiator is HLR
    INITIATOR CONSUMER OF {
        ResetPackage-v2}
::= {map-ac reset(10) version2(2)}
```

The following application-context-name is assigned to the v1-equivalent application-context:

```
{map-ac reset(10) version1(1)}
```

14.3.2.12 Handover control

This application context is used for handover procedures between MSCs.

```
handoverControlContext-v2 APPLICATION-CONTEXT
    -- Responder is MSCB if Initiator is MSCA
    INITIATOR CONSUMER OF {
        HandoverControlPackage-v2}
::= {map-ac handoverControl(11) version2(2)}
```

The following application-context-name is assigned to the v1-equivalent application-context:

```
{map-ac handoverControl(11) version1(1)}
```

14.3.2.13 IMSI Retrieval

This application context is used for IMSI retrieval between HLR and VLR.

```
imsiRetrievalContext-v2 APPLICATION-CONTEXT
    -- Responder is HLR if Initiator is VLR
    INITIATOR CONSUMER OF {
        IMSIRetrievalPackage-v2}
::= {map-ac imsiRetrieval(26) version2(2)}
```

This application-context is v2 only.

14.3.2.14 Equipment Management

This application context is used for equipment checking between MSC and EIR:

```
equipmentMngtContext-v2 APPLICATION-CONTEXT
    -- Responder is EIR if Initiator is MSC
    INITIATOR CONSUMER OF {
        EquipmentMngtPackage-v2}
::= {map-ac equipmentMngt(13) version2(2)}
```

The following application-context-name is assigned to the v1-equivalent application-context:

```
{map-ac equipmentMngt(13) version1(1)}
```

14.3.2.15 Information retrieval

This application context is used for authentication information retrieval between HLR and VLR.

```
infoRetrievalContext-v2 APPLICATION-CONTEXT
    -- Responder is HLR if Initiator is VLR
    INITIATOR CONSUMER OF {
        InfoRetrievalPackage-v2}
::= {map-ac infoRetrieval(14) version2(2)}
```

The following application-context-name is assigned to the v1-equivalent application-context:

```
    -- Responder is HLR if Initiator is VLR
{map-ac infoRetrieval(14) version1(1)}
```

14.3.2.16   Inter-VLR information retrieval

This application context is used for information retrieval between VLRs.

```
interVlrInfoRetrievalContext-v2 APPLICATION-CONTEXT
    -- Responder is VLR if Initiator is VLR
    INITIATOR CONSUMER OF {
        InterVlrInfoRetrievalPackage-v2}
::= {map-ac interVlrInfoRetrieval(15) version2(2)}
```

The v1-equivalent application-context is:

```
    -- Responder is VLR if Initiator is VLR
{map-ac infoRetrieval(14) version1(1)}
```

14.3.2.17   Stand Alone Subscriber Data Management

This application context is used for stand alone subscriber data management between HLR and VLR:

```
subscriberDataMngtContext-v2 APPLICATION-CONTEXT
    -- Responder is VLR if Initiator is HLR
    INITIATOR CONSUMER OF {
        SubscriberDataMngtStandAlonePackage-v2}
::= {map-ac subscriberDataMngt(16) version2(2)}
```

The following application-context-name is assigned to the v1-equivalent application-context:

```
{map-ac subscriberDataMngt(16) version1(1)}
```

14.3.2.18   Tracing

This application context is used for stand alone tracing control procedures:

```
tracingContext-v2 APPLICATION-CONTEXT
    -- Responder is VLR if Initiator is HLR
    INITIATOR CONSUMER OF {
        TracingStandAlonePackage-v2}
::= {map-ac tracing(17) version2(2)}
```

The following application-context-name is assigned to the v1-equivalent application-context:

```
{map-ac tracing(17) version1(1)}
```

14.3.2.19 Network functional SS handling

This application context is used for functional-like SS handling procedures between VLR and HLR.

```
networkFunctionalSsContext-v2 APPLICATION-CONTEXT
    -- Responder is HLR, Initiator is VLR
    INITIATOR CONSUMER OF {
        FunctionalSsPackage-v2}
::= {map-ac networkFunctionalSs(18) version2(2)}
```

The v1-equivalent application-context is defined as follows:

```
networkFunctionalSsContext-v1 APPLICATION-CONTEXT
    -- Responder is HLR, Initiator is VLR
    INITIATOR CONSUMER OF {
        FunctionalSsPackage-v1,
        UnstructuredSsPackage-v1,
        BindingPackage-v1}
::= {map-ac networkFunctionalSs(18) version1(1)}
```

14.3.2.20 Network unstructured SS handling

This application context is used for handling stimuli-like procedures between HLR and VLR.

```
networkUnstructuredSsContext-v2 APPLICATION-CONTEXT
    -- Responder is HLR, Initiator is VLR
    -- Responder is VLR, Initiator is HLR
    OPERATIONS OF {
        UnstructuredSsPackage-v2}
::= {map-ac networkUnstructuredSs(19) version2(2)}
```

The v1-equivalent application-context is: networkFunctionalSsContext-v1.

14.3.2.21 Short Message Gateway

This application context is used for short message gateway procedures.

```
shortMsgGatewayContext-v2 APPLICATION-CONTEXT
    -- Responder is HLR if Initiator is GMSC
    INITIATOR CONSUMER OF {
        ShortMsgGatewayPackage-v2}
::= {map-ac shortMsgGateway(20) version2(2)}
```

The following application-context-name is assigned to the v1-equivalent application-context:

```
{map-ac shortMsgGateway(20) version1(1)}
```

14.3.2.22   Mobile originating Short Message Relay

This application context is used for mobile originating short message relay procedures.

```
shortMsgMO-RelayContext-v2 APPLICATION-CONTEXT
    -- Responder is IWMSC if Initiator is MSC
    INITIATOR CONSUMER OF {
        ShortMsgRelayPackage-v2}
::= {map-ac shortMsgMO-Relay(21) version2(2)}
```

The following application-context-name is assigned to the v1-equivalent application-context:

```
{map-ac --shortMsgRelay--21 version1(1)}
```

14.3.2.23   [spare]

14.3.2.24   Short message alert

This application context is used for short message alerting procedures.

```
shortMsgAlertContext-v2 APPLICATION-CONTEXT
    -- Responder is IWMSC if Initiator is HLR
    INITIATOR CONSUMER OF {
        AlertingPackage-v2}
::= {map-ac shortMsgAlert(23) version2(2)}
```

The following application-context-name is symbolically assigned to the v1-equivalent application-context:

```
{map-ac shortMsgAlert(23) version1(1)}
```

14.3.2.25   Short message waiting data management

This application context is used for short message waiting data management procedures.

```
mwdMngtContext-v2 APPLICATION-CONTEXT
    -- Responder is HLR if Initiator is VLR
    INITIATOR CONSUMER OF {
        MwdMngtPackage-v2}
::= {map-ac mwdMngt(24) version2(2)}
```

The following application-context-name is assigned to the v1-equivalent application-context:

```
{map-ac mwdMngt(24) version1(1)}
```

14.3.2.26   Mobile terminating Short Message Relay

This application context is used for mobile terminating short message relay procedures.

```
shortMsgMT-RelayContext-v2 APPLICATION-CONTEXT
    -- Responder is MSC if Initiator is GMSC
    INITIATOR CONSUMER OF {
        ShortMsgRelayPackage-v2}
::= {map-ac shortMsgMT-Relay(25) version2(2)}
```

The v1-equivalent application-context is: shortMsgRelayContext_v1.

14.3.2.27   MS purging

This application context is used between HLR and VLR for MS purging procedures.

```
msPurgingContext-v2 APPLICATION-CONTEXT
    -- Responder is HLR if Initiator is VLR
    INITIATOR CONSUMER OF {
        purgingPackage-v2}
::= {map-ac msPurging(27) version2(2)}
```

This application-context is v2 only.

14.3.3   ASN.1 Module for application-context-names

The following ASN.1 module summarizes the application-context-name assigned to MAP application-contexts.

```
1  MAP-ApplicationContexts {
2      ccitt identified-organization (4) etsi (0) mobileDomain (0)
3      gsm-Network (1) modules (3) map-ApplicationContexts (2) version2 (2)}
4
5  DEFINITIONS
6
7  ::=
8
9  BEGIN
10
11
12 -- EXPORTS everything
13
14
15 IMPORTS
16      gsm-NetworkId,
17      ac-Id
18 FROM MobileDomainDefinitions {
19     ccitt (0) identified-organization (4) etsi (0) mobileDomain (0)
20     mobileDomainDefinitions (0) version1 (1)}
21 ;
22
23 -- application-context-names
24
25 map-ac   OBJECT IDENTIFIER ::= {gsm-NetworkId ac-Id}
26
27 networkLocUpContext-v2  OBJECT IDENTIFIER ::=
28     {map-ac networkLocUp(1) version2(2)}
29 networkLocUpContext-v1  OBJECT IDENTIFIER ::=
30     {map-ac networkLocUp(1) version1(1)}
31
```

```
32  locationCancellation-v2  OBJECT IDENTIFIER ::=
33      {map-ac locationCancel(2) version2(2)}
34  locationCancellation-v1  OBJECT IDENTIFIER ::=
35      {map-ac locationCancel(2) version1(1)}
36
37  roamingNumberEnquiryContext-v2  OBJECT IDENTIFIER ::=
38      {map-ac roamingNbEnquiry(3) version2(2)}
39  roamingNumberEnquiryContext-v1  OBJECT IDENTIFIER ::=
40      {map-ac roamingNbEnquiry(3) version1(1)}
41
42  locationInfoRetrievalContext-v2  OBJECT IDENTIFIER ::=
43      {map-ac locInfoRetrieval(5) version2(2)}
44  locationInfoRetrievalContext-v1  OBJECT IDENTIFIER ::=
45      {map-ac locInfoRetrieval(5) version1(1)}
46
47  resetContext-v2  OBJECT IDENTIFIER ::=
48      {map-ac reset(10) version2(2)}
49  resetContext-v1  OBJECT IDENTIFIER ::=
50      {map-ac reset(10) version1(1)}
51
52  handoverControlContext-v2  OBJECT IDENTIFIER ::=
53      {map-ac handoverControl(11) version2(2)}
54  handoverControlContext-v1  OBJECT IDENTIFIER ::=
55      {map-ac handoverControl(11) version1(1)}
56
57  equipmentMngtContext-v2  OBJECT IDENTIFIER ::=
58      {map-ac equipmentMngt(13) version2(2)}
59  equipmentMngtContext-v1  OBJECT IDENTIFIER ::=
60      {map-ac equipmentMngt(13) version1(1)}
61
62  infoRetrievalContext-v2  OBJECT IDENTIFIER ::=
63      {map-ac infoRetrieval(14) version2(2)}
64  infoRetrievalContext-v1  OBJECT IDENTIFIER ::=
65      {map-ac infoRetrieval(14) version1(1)}
66
67  interVlrInfoRetrievalContext-v2  OBJECT IDENTIFIER ::=
68      {map-ac interVlrInfoRetrieval(15) version2(2)}
69
70  subscriberDataMngtContext-v2  OBJECT IDENTIFIER ::=
71      {map-ac subscriberDataMngt(16) version2(2)}
72  subscriberDataMngtContext-v1  OBJECT IDENTIFIER ::=
73      {map-ac subscriberDataMngt(16) version1(1)}
74
75  tracingContext-v2  OBJECT IDENTIFIER ::=
76      {map-ac tracing(17) version2(2)}
77  tracingContext-v1  OBJECT IDENTIFIER ::=
78      {map-ac tracing(17) version1(1)}
79
80  networkFunctionalSsContext-v2  OBJECT IDENTIFIER ::=
81      {map-ac networkFunctionalSs(18) version2(2)}
82  networkFunctionalSsContext-v1  OBJECT IDENTIFIER ::=
83      {map-ac networkFunctionalSs(18) version1(1)}
```

```
84
85  networkUnstructuredSsContext-v2  OBJECT IDENTIFIER ::=
86      {map-ac networkUnstructuredSs(19) version2(2)}
87
88  shortMsgGatewayContext-v2  OBJECT IDENTIFIER ::=
89      {map-ac shortMsgGateway(20) version2(2)}
90  shortMsgGatewayContext-v1  OBJECT IDENTIFIER ::=
91      {map-ac shortMsgGateway(20) version1(1)}
92
93  shortMsgMO-RelayContext-v2  OBJECT IDENTIFIER ::=
94      {map-ac shortMsgMO-Relay(21) version2(2)}
95  shortMsgRelayContext-v1  OBJECT IDENTIFIER ::=
96      {map-ac --shortMsgRelay--21 version1(1)}
97
98   shortMsgAlertContext-v2  OBJECT IDENTIFIER ::=
99       {map-ac shortMsgAlert(23) version2(2)}
100  shortMsgAlertContext-v1  OBJECT IDENTIFIER ::=
101      {map-ac shortMsgAlert(23) version1(1)}
102
103  mwdMngtContext-v2  OBJECT IDENTIFIER ::=
104      {map-ac mwdMngt(24) version2(2)}
105  mwdMngtContext-v1  OBJECT IDENTIFIER ::=
106      {map-ac mwdMngt(24) version1(1)}
107
108  shortMsgMT-RelayContext-v2  OBJECT IDENTIFIER ::=
109      {map-ac shortMsgMT-Relay(25) version2(2)}
110
111  imsiRetrievalContext-v2  OBJECT IDENTIFIER ::=
112      {map-ac imsiRetrieval(26) version2(2)}
113
114  msPurgingContext-v2  OBJECT IDENTIFIER ::=
115      {map-ac msPurging(27) version2(2)}
116
117  END
```

14.4   MAP Dialogue Information

```
1  MAP-DialogueInformation {
2      ccitt identified-organization (4) etsi (0) mobileDomain (0)
3      gsm-Network (1) modules (3) map-DialogueInformation (3) version2 (2)}
4
5  DEFINITIONS
6
7  IMPLICIT TAGS
8
9  ::=
10
11 BEGIN
12
13 EXPORTS
14     map-DialogueAS,
15     MAP-DialoguePDU
16 ;
17
18 IMPORTS
19     gsm-NetworkId,
20     as-Id
21 FROM MobileDomainDefinitions {
22     ccitt (0) identified-organization (4) etsi (0) mobileDomain (0)
23     mobileDomainDefinitions (0) version1 (1)}
24
25     AddressString
26 FROM MAP-CommonDataTypes {
27     ccitt identified-organization (4) etsi (0) mobileDomain (0)
28     gsm-Network(1) modules (3) map-CommonDataTypes (18) version2 (2)}
29 ;
30
31
32 -- abstract syntax name for MAP-DialoguePDU
33
34 map-DialogueAS   OBJECT IDENTIFIER ::=
35     {gsm-NetworkId as-Id map-DialoguePDU (1) version1 (1)}
36
37 MAP-DialoguePDU ::= CHOICE {
38     map-open                    [0] MAP-OpenInfo,
39     map-accept                  [1] MAP-AcceptInfo,
40     map-close                   [2] MAP-CloseInfo,
41     map-refuse                  [3] MAP-RefuseInfo,
42     map-userAbort               [4] MAP-UserAbortInfo,
43     map-providerAbort           [5] MAP-ProviderAbortInfo}
44
45 MAP-OpenInfo ::= SEQUENCE {
46     destinationReference        [0] AddressString          OPTIONAL,
47     originationReference        [1] AddressString          OPTIONAL,
48     ...}
49
50 MAP-AcceptInfo ::= SEQUENCE {
51     ...}
52
53 MAP-CloseInfo ::= SEQUENCE {
54     ...}
55
```

```
56  MAP-RefuseInfo ::= SEQUENCE {
57      reason   Reason,
58      ...}
59
60  Reason ::= ENUMERATED {
61      noReasonGiven   (0),
62      invalidDestinationReference   (1),
63      invalidOriginatingReference   (2)}
64
65  MAP-UserAbortInfo ::= SEQUENCE {
66      map-UserAbortChoice            MAP-UserAbortChoice,
67      ...}
68
69  MAP-UserAbortChoice ::= CHOICE {
70      userSpecificReason            [0] NULL,
71      userResourceLimitation        [1] NULL,
72      resourceUnavailable           [2] ResourceUnavailableReason,
73      applicationProcedureCancellation    [3] ProcedureCancellationReason}
74
75  ResourceUnavailableReason ::= ENUMERATED {
76      shortTermResourceLimitation   (0),
77      longTermResourceLimitation    (1)}
78
79  ProcedureCancellationReason ::= ENUMERATED {
80      handoverCancellation   (0),
81      radioChannelRelease   (1),
82      networkPathRelease   (2),
83      callRelease   (3),
84      associatedProcedureFailure   (4),
85      tandemDialogueRelease   (5),
86      remoteOperationsFailure   (6)}
87
88  MAP-ProviderAbortInfo ::= SEQUENCE {
89      map-ProviderAbortReason         MAP-ProviderAbortReason,
90      ...}
91
92  MAP-ProviderAbortReason ::= ENUMERATED {
93      abnormalDialogue   (0),
94      invalidPDU   (1)}
95
96  END
```

14.5   MAP operation and error codes

```
1  MAP-Protocol {
2      ccitt identified-organization (4) etsi (0) mobileDomain (0)
3      gsm-Network (1) modules (3) map-Protocol (4) version2 (2)}
4
5  DEFINITIONS
6
7  ::=
8
9  BEGIN
10
11  IMPORTS
12      UpdateLocation,
13      CancelLocation,
14      PurgeMS,
15      SendIdentification,
16      PerformHandover,
17      PrepareHandover,
18      SendEndSignal,
19      ProcessAccessSignalling,
20      ForwardAccessSignalling,
21      PerformSubsequentHandover,
22      PrepareSubsequentHandover,
23      SendAuthenticationInfo,
24      CheckIMEI,
25      SendParameters,
26      InsertSubscriberData,
27      DeleteSubscriberData,
28      Reset,
29      ForwardCheckSS-Indication,
30      RestoreData
31  FROM MAP-MobileServiceOperations {
32      ccitt identified-organization (4) etsi (0) mobileDomain (0)
33      gsm-Network (1) modules (3) map-MobileServiceOperations (5)
34      version2 (2)}
35
36      ActivateTraceMode,
37      DeactivateTraceMode,
38      TraceSubscriberActivity,
39      NoteInternalHandover,
40      SendIMSI
41  FROM MAP-OperationAndMaintenanceOperations {
42      ccitt identified-organization (4) etsi (0) mobileDomain (0)
43      gsm-Network (1) modules (3) map-OperationAndMaintenanceOperations (6)
44      version2 (2)}
45
46      SendRoutingInfo,
47      ProvideRoamingNumber
48  FROM MAP-CallHandlingOperations {
49      ccitt identified-organization (4) etsi (0) mobileDomain (0)
50      gsm-Network (1) modules (3) map-CallHandlingOperations (7)
51      version2 (2)}
52
53      RegisterSS,
54      EraseSS,
55      ActivateSS,
56      DeactivateSS,
57      InterrogateSS,
58      ProcessUnstructuredSS-Data,
59      ProcessUnstructuredSS-Request,
60      UnstructuredSS-Request,
61      UnstructuredSS-Notify,
62      RegisterPassword,
63      GetPassword,
```

```
 64        BeginSubscriberActivity
 65 FROM MAP-SupplementaryServiceOperations {
 66    ccitt identified-organization (4) etsi (0) mobileDomain (0)
 67    gsm-Network (1) modules (3) map-SupplementaryServiceOperations (8)
 68    version2 (2)}
 69
 70        SendRoutingInfoForSM,
 71        ForwardSM,
 72        ReportSM-DeliveryStatus,
 73        NoteSubscriberPresent,
 74        AlertServiceCentreWithoutResult,
 75        AlertServiceCentre,
 76        InformServiceCentre,
 77        ReadyForSM
 78 FROM MAP-ShortMessageServiceOperations {
 79    ccitt identified-organization (4) etsi (0) mobileDomain (0)
 80    gsm-Network (1) modules (3) map-ShortMessageServiceOperations (9)
 81    version2 (2)}
 82
 83        SystemFailure,
 84        DataMissing,
 85        UnexpectedDataValue,
 86        FacilityNotSupported,
 87        UnknownSubscriber,
 88        NumberChanged,
 89        UnknownBaseStation,
 90        UnknownMSC,
 91        UnidentifiedSubscriber,
 92        UnknownEquipment,
 93        RoamingNotAllowed,
 94        IllegalSubscriber,
 95        IllegalEquipment,
 96        BearerServiceNotProvisioned,
 97        TeleserviceNotProvisioned,
 98        InvalidTargetBaseStation,
 99        NoRadioResourceAvailable,
100        NoHandoverNumberAvailable,
101        SubsequentHandoverFailure,
102        TracingBufferFull,
103        NoRoamingNumberAvailable,
104        AbsentSubscriber,
105        CallBarred,
106        ForwardingViolation,
107        CUG-Reject,
108        IllegalSS-Operation,
109        SS-ErrorStatus,
110        SS-NotAvailable,
111        SS-SubscriptionViolation,
112        SS-Incompatibility,
113        UnknownAlphabet,
114        USSD-Busy,
115        PW-RegistrationFailure,
116        NegativePW-Check,
117        NumberOfPW-AttemptsViolation,
118        SubscriberBusyForMT-SMS,
119        SM-DeliveryFailure,
120        MessageWaitingListFull
121 FROM MAP-Errors {
122    ccitt identified-organization (4) etsi (0) mobileDomain (0)
123    gsm-Network (1) modules (3) map-Errors (10) version2 (2)}
124 ;
125
126
127 -- location registration operation codes
128
```

```
129 updateLocation  UpdateLocation ::= localValue 2
130 cancelLocation  CancelLocation ::= localValue 3
131 purgeMS  PurgeMS ::= localValue 67
132 -- purgeMS must not be used in version 1
133 sendIdentification  SendIdentification ::= localValue 55
134 -- sendIdentification must not be used in version 1
135
136
137 -- handover operation codes
138
139 performHandover  PerformHandover ::= localValue 28
140 -- performHandover must not be used in version greater 1
141 prepareHandover  PrepareHandover ::= localValue 68
142 -- prepareHandover must not be used in version 1
143 sendEndSignal  SendEndSignal ::= localValue 29
144 processAccessSignalling  ProcessAccessSignalling ::= localValue 33
145 forwardAccessSignalling  ForwardAccessSignalling ::= localValue 34
146 performSubsequentHandover  PerformSubsequentHandover ::=
147    localValue 30
148 -- performSubsequentHandover must not be used in version greater 1
149 prepareSubsequentHandover  PrepareSubsequentHandover ::=
150    localValue 69
151 -- prepareSubsequentHandover must not be used in version 1
152
153
154 -- authentication operation codes
155
156 sendAuthenticationInfo  SendAuthenticationInfo ::= localValue 56
157 -- sendAuthenticationInfo must not be used in version 1
158
159
160 -- IMEI MANAGEMENT operation codes
161
162 checkIMEI   CheckIMEI ::= localValue 43
163
164
165 -- subscriber management operation codes
166
167 sendParameters  SendParameters ::= localValue 9
168 -- sendParameters must not be used in version greater 1
169 insertSubscriberData  InsertSubscriberData ::= localValue 7
170 deleteSubscriberData  DeleteSubscriberData ::= localValue 8
171
172
173 -- fault recovery operation codes

174
175 reset  Reset ::= localValue 37
176 forwardCheckSS-Indication  ForwardCheckSS-Indication ::=
177    localValue 38
178 restoreData  RestoreData ::= localValue 57
179 -- restoreData must not be used in version 1
180
181
```

```
182 -- operation and maintenance operation codes
183
184 activateTraceMode   ActivateTraceMode ::= localValue 50
185 deactivateTraceMode   DeactivateTraceMode ::= localValue 51
186 traceSubscriberActivity   TraceSubscriberActivity ::= localValue 52
187 -- traceSubscriberActivity must not be used in version greater 1
188 noteInternalHandover   NoteInternalHandover ::= localValue 35
189 -- noteInternalHandover must not be used in version greater 1
190 sendIMSI   SendIMSI ::= localValue 58
191 -- sendIMSI must not be used in version 1
192
193
194 -- call handling operation codes
195
196 sendRoutingInfo   SendRoutingInfo ::= localValue 22
197 provideRoamingNumber   ProvideRoamingNumber ::= localValue 4
198
199
200 -- supplementary service handling operation codes
201
202 registerSS   RegisterSS ::= localValue 10
203 eraseSS   EraseSS ::= localValue 11
204 activateSS   ActivateSS ::= localValue 12
205 deactivateSS   DeactivateSS ::= localValue 13
206 interrogateSS   InterrogateSS ::= localValue 14
207 processUnstructuredSS-Data   ProcessUnstructuredSS-Data ::=
208       localValue 19
209 --  processUnstructuredSS-Data must not be used in version greater 1
210 processUnstructuredSS-Request   ProcessUnstructuredSS-Request ::=
211       localValue 59
212 -- processUnstructuredSS-Request must not be used in version 1
213 unstructuredSS-Request   UnstructuredSS-Request ::= localValue 60
214 -- unstructuredSS-Request must not be used in version 1
215 unstructuredSS-Notify   UnstructuredSS-Notify ::= localValue 61
216 -- unstructuredSS-Notify must not be used in version 1
217 registerPassword   RegisterPassword ::= localValue 17
218 getPassword   GetPassword ::= localValue 18
219 beginSubscriberActivity   BeginSubscriberActivity ::= localValue 54
220 -- beginSubscriberActivity must not be used in version greater 1
221
222
223 -- short message service operation codes
224
225 sendRoutingInfoForSM   SendRoutingInfoForSM ::= localValue 45
226 forwardSM   ForwardSM ::= localValue 46
227 reportSM-DeliveryStatus   ReportSM-DeliveryStatus ::= localValue 47
228 noteSubscriberPresent   NoteSubscriberPresent ::= localValue 48
229 -- noteSubscriberPresent must not be used in version greater 1
230 alertServiceCentreWithoutResult   AlertServiceCentreWithoutResult ::=
231       localValue 49
232 -- alertServiceCentreWithoutResult must not be used in
233 -- version greater 1
234 informServiceCentre   InformServiceCentre ::= localValue 63
235 -- informServiceCentre must not be used in version 1
236 alertServiceCentre   AlertServiceCentre ::= localValue 64
237 -- alertServiceCentre must not be used in version 1
238 readyForSM   ReadyForSM ::= localValue 66
239 -- readyForSM must not be used in version 1
240
241
```

```
242  -- generic error codes
243
244  systemFailure  SystemFailure ::= localValue 34
245  dataMissing  DataMissing ::= localValue 35
246  unexpectedDataValue  UnexpectedDataValue ::= localValue 36
247  facilityNotSupported  FacilityNotSupported ::= localValue 21
248
249
250  -- identification and numbering error codes
251
252  unknownSubscriber  UnknownSubscriber ::= localValue 1
253  numberChanged  NumberChanged ::= localValue 44
254  unknownBaseStation  UnknownBaseStation ::= localValue 2
255  unknownMSC  UnknownMSC ::= localValue 3
256  unidentifiedSubscriber  UnidentifiedSubscriber ::= localValue 5
257  unknownEquipment  UnknownEquipment ::= localValue 7
258
259
260  -- subscription error codes
261
262  roamingNotAllowed  RoamingNotAllowed ::= localValue 8
263  illegalSubscriber  IllegalSubscriber ::= localValue 9
264  illegalEquipment  IllegalEquipment ::= localValue 12
265  bearerServiceNotProvisioned  BearerServiceNotProvisioned ::=
266      localValue 10
267  teleserviceNotProvisioned  TeleserviceNotProvisioned ::=
268      localValue 11
269
270
271  -- handover error codes
272
273  invalidTargetBaseStation  InvalidTargetBaseStation ::= localValue 23
274  noRadioResourceAvailable  NoRadioResourceAvailable ::= localValue 24
275  noHandoverNumberAvailable  NoHandoverNumberAvailable ::=
276      localValue 25
277  subsequentHandoverFailure  SubsequentHandoverFailure ::=
278      localValue 26
279
280
281  -- operation and maintenance error codes
282
283  tracingBufferFull  TracingBufferFull ::= localValue 40
284
285
286  -- call handling error codes

287
288  noRoamingNumberAvailable  NoRoamingNumberAvailable ::= localValue 39
289  absentSubscriber  AbsentSubscriber ::= localValue 27
290  callBarred  CallBarred ::= localValue 13
291  forwardingViolation  ForwardingViolation ::= localValue 14
292  cug-Reject  CUG-Reject ::= localValue 15
293
294
```

```
295  -- supplementary service error codes
296
297  illegalSS-Operation  IllegalSS-Operation ::= localValue 16
298  ss-ErrorStatus  SS-ErrorStatus ::= localValue 17
299  ss-NotAvailable  SS-NotAvailable ::= localValue 18
300  ss-SubscriptionViolation  SS-SubscriptionViolation ::= localValue 19
301  ss-Incompatibility  SS-Incompatibility ::= localValue 20
302  unknownAlphabet  UnknownAlphabet ::= localValue 71
303  -- unknownAlphabet must not be used in version 1
304  ussd-Busy  USSD-Busy ::= localValue 72
305  -- ussd-Busy must not be used in version 1
306  pw-RegistrationFailure  PW-RegistrationFailure ::= localValue 37
307  negativePW-Check  NegativePW-Check ::= localValue 38
308  numberOfPW-AttemptsViolation  NumberOfPW-AttemptsViolation ::=
309       localValue 43
310  -- numberOfPW-AttemptsViolation must not be used in version 1
311
312
313  -- short message service error codes
314
315  subscriberBusyForMT-SMS  SubscriberBusyForMT-SMS ::= localValue 31
316  -- subscriberBusyForMT-SMS must not be used in version 1
317  sm-DeliveryFailure  SM-DeliveryFailure ::= localValue 32
318  messageWaitingListFull  MessageWaitingListFull ::= localValue 33
319
320  END
```

14.6 MAP operation and error types

14.6.1 Mobile Service Operations

```
1  MAP-MobileServiceOperations {
2      ccitt identified-organization (4) etsi (0) mobileDomain (0)
3      gsm-Network (1) modules (3) map-MobileServiceOperations (5)
4      version2 (2)}
5
6  DEFINITIONS
7
8  ::=
9
10 BEGIN
11
12 EXPORTS
13
14     -- location registration operations
15     UpdateLocation,
16     CancelLocation,
17     PurgeMS,
18     SendIdentification,
19
20     -- handover operations
21     PerformHandover,
22     PrepareHandover,
23     SendEndSignal,
24     ProcessAccessSignalling,
25     ForwardAccessSignalling,
26     PerformSubsequentHandover,
27     PrepareSubsequentHandover,
28
29     -- authentication management operations
30     SendAuthenticationInfo,
31
32     -- IMEI management operations
33     CheckIMEI,
34
35     -- subscriber management operations
36     SendParameters,
37     InsertSubscriberData,
38     DeleteSubscriberData,
39
40     -- fault recovery operations
41     Reset,
42     ForwardCheckSS-Indication,
43     RestoreData
44 ;
45
46 IMPORTS
47     OPERATION
48 FROM TCAPMessages {
49    ccitt recommendation q 773 modules (2) messages (1) version2 (2)}
50
51     SystemFailure,
52     DataMissing,
53     UnexpectedDataValue,
54     UnknownSubscriber,
55     UnknownBaseStation,
56     UnknownMSC,
57     UnidentifiedSubscriber,
58     UnknownEquipment,
59     RoamingNotAllowed,
60     InvalidTargetBaseStation,
61     NoRadioResourceAvailable,
```

```
62        NoHandoverNumberAvailable,
63        SubsequentHandoverFailure
64 FROM MAP-Errors {
65    ccitt identified-organization (4) etsi (0) mobileDomain (0)
66    gsm-Network (1) modules (3) map-Errors (10) version2 (2)}
67
68        UpdateLocationArg,
69        UpdateLocationRes,
70        CancelLocationArg,
71        PurgeMS-Arg,
72        SendIdentificationRes,
73        PerformHO-Arg,
74        PrepareHO-Arg,
75        PerformHO-Res,
76        PrepareHO-Res,
77        PerformSubsequentHO-Arg,
78        PrepareSubsequentHO-Arg,
79        SendAuthenticationInfoArg,
80        SendAuthenticationInfoRes,
81        EquipmentStatus,
82        SendParametersArg,
83        SentParameterList,
84        InsertSubscriberDataArg,
85        InsertSubscriberDataRes,
86        DeleteSubscriberDataArg,
87        DeleteSubscriberDataRes,
88        ResetArg,
89        RestoreDataArg,
90        RestoreDataRes
91 FROM MAP-MS-DataTypes {
92    ccitt identified-organization (4) etsi (0) mobileDomain (0)
93    gsm-Network (1) modules (3) map-MS-DataTypes (11) version2 (2)}
94
95        ExternalSignalInfo,
96        TMSI,
97        IMEI
98 FROM MAP-CommonDataTypes {
99    ccitt identified-organization (4) etsi (0) mobileDomain (0)
100    gsm-Network (1) modules (3) map-CommonDataTypes (18) version2 (2)}
101 ;
102
103
104 -- location registration operations
105
106 UpdateLocation ::= OPERATION                                    --Timer m
107     ARGUMENT
108        updateLocationArg           UpdateLocationArg
109     RESULT
110        updateLocationRes           UpdateLocationRes
111     ERRORS {
112        SystemFailure,
113        DataMissing,
114        -- DataMissing must not be used in version 1
115        UnexpectedDataValue,
116        UnknownSubscriber,
117        RoamingNotAllowed}
118
```

```
119  CancelLocation ::= OPERATION                                      --Timer m
120      ARGUMENT
121          cancelLocationArg          CancelLocationArg
122      RESULT
123      ERRORS {
124          DataMissing,
125          -- DataMissing must not be used in version 1
126          UnexpectedDataValue,
127          UnidentifiedSubscriber}
128          -- UnidentifiedSubscriber must not be used in version
129          -- greater 1
130
131  PurgeMS ::= OPERATION                                             --Timer m
132      ARGUMENT
133          purgeMS-Arg                PurgeMS-Arg
134      RESULT
135
136  SendIdentification::= OPERATION                                   --Timer s
137      ARGUMENT
138          tmsi                       TMSI
139      RESULT
140          sendIdentificationRes      SendIdentificationRes
141      ERRORS {
142          DataMissing,
143          UnidentifiedSubscriber}
144
145  -- handover operations
146
147  PrepareHandover ::= OPERATION                                     --Timer m
148      ARGUMENT
149          prepareHO-Arg              PrepareHO-Arg
150      RESULT
151          prepareHO-Res              PrepareHO-Res
152      ERRORS {
153          SystemFailure,
154          DataMissing,
155          UnexpectedDataValue,
156          NoHandoverNumberAvailable}
157
158  PerformHandover ::= OPERATION                                     --Timer s
159      ARGUMENT
160          performHO-Arg              PerformHO-Arg
161      RESULT
162          performHO-Res              PerformHO-Res
163      ERRORS {
164          SystemFailure,
165          UnexpectedDataValue,
166          UnknownBaseStation,
167          InvalidTargetBaseStation,
168          NoRadioResourceAvailable,
169          NoHandoverNumberAvailable}
170
171  SendEndSignal ::= OPERATION                                       --Timer l
172      ARGUMENT
173          bss-APDU                   ExternalSignalInfo
174          -- optional
175          -- bss-APDU must be absent in version 1
176          -- bss-APDU must be present in version greater 1
177      RESULT
```

```
178
179  ProcessAccessSignalling ::= OPERATION                          --Timer s
180      ARGUMENT
181          bss-APDU                    ExternalSignalInfo
182
183  ForwardAccessSignalling ::= OPERATION                          --Timer s
184      ARGUMENT
185          bss-APDU                    ExternalSignalInfo
186
187  PrepareSubsequentHandover ::= OPERATION                        --Timer m
188      ARGUMENT
189          prepareSubsequentHO-Arg     PrepareSubsequentHO-Arg
190      RESULT
191          bss-APDU                    ExternalSignalInfo
192      ERRORS {
193          UnexpectedDataValue,
194          DataMissing,
195          UnknownMSC,
196          SubsequentHandoverFailure}
197
198  PerformSubsequentHandover ::= OPERATION                        --Timer m
199      ARGUMENT
200          performSubsequentHO-Arg     PerformSubsequentHO-Arg
201      RESULT
202          accessSignalInfo            ExternalSignalInfo
203      ERRORS {
204          UnexpectedDataValue,
205          UnknownBaseStation,
206          UnknownMSC,
207          InvalidTargetBaseStation,
208          SubsequentHandoverFailure}
209
210  -- authentication management operations
211
212  SendAuthenticationInfo ::= OPERATION                           --Timer m
213      ARGUMENT
214          sendAuthenticationInfoArg   SendAuthenticationInfoArg
215      RESULT
216          sendAuthenticationInfoRes   SendAuthenticationInfoRes
217          -- optional
218      ERRORS {
219          SystemFailure,
220          DataMissing,
221          UnexpectedDataValue,
222          UnknownSubscriber}
223
```

```
224  -- IMEI management operations
225
226  CheckIMEI ::= OPERATION                                        --Timer m
227      ARGUMENT
228          imei                          IMEI
229      RESULT
230          equipmentStatus               EquipmentStatus
231      ERRORS {
232          SystemFailure,
233          DataMissing,
234          -- DataMissing must not be used in version 1
235          UnexpectedDataValue,
236          -- UnexpectedDataValue must not be used in version greater 1
237          UnknownEquipment}
238
239  -- subscriber management operations
240
241  SendParameters ::= OPERATION                                   --Timer m
242      ARGUMENT
243          sendParametersArg             SendParametersArg
244      RESULT
245          sentParameterList             SentParameterList
246          -- optional
247          -- nothing is returned, if no requested parameter is
248          -- available or exists
249      ERRORS {
250          UnexpectedDataValue,
251          UnknownSubscriber,
252          UnidentifiedSubscriber}
253
254  InsertSubscriberData ::= OPERATION                             --Timer m
255      ARGUMENT
256          insertSubscriberDataArg       InsertSubscriberDataArg
257      RESULT
258          insertSubscriberDataRes       InsertSubscriberDataRes
259          -- optional
260          -- insertSubscriberDataRes must be absent in version 1
261      ERRORS {
262          DataMissing,
263          UnexpectedDataValue,
264          UnidentifiedSubscriber}
265
266  DeleteSubscriberData ::= OPERATION                             --Timer m
267      ARGUMENT
268          deleteSubscriberDataArg       DeleteSubscriberDataArg
269      RESULT
270          deleteSubscriberDataRes       DeleteSubscriberDataRes
271          -- optional
272          -- deleteSubscriberDataRes must be absent in version 1
273      ERRORS {
274          DataMissing,
275          UnexpectedDataValue,
276          UnidentifiedSubscriber}
277
```

```
278  -- fault recovery operations
279
280  Reset ::= OPERATION                                              --Timer m
281      ARGUMENT
282          resetArg                    ResetArg
283
284  ForwardCheckSS-Indication ::= OPERATION      --Timer s
285
286  RestoreData ::= OPERATION                                        --Timer m
287      ARGUMENT
288          restoreDataArg              RestoreDataArg
289      RESULT
290          restoreDataRes              RestoreDataRes
291      ERRORS {
292          SystemFailure,
293          DataMissing,
294          UnexpectedDataValue,
295          UnknownSubscriber}
296
297  END
```

14.6.2 Operation and Maintenance Operations

```
1  MAP-OperationAndMaintenanceOperations {
2      ccitt identified-organization (4) etsi (0) mobileDomain (0)
3      gsm-Network (1) modules (3) map-OperationAndMaintenanceOperations (6)
4      version2 (2)}
5
6  DEFINITIONS
7
8  ::=
9
10 BEGIN
11
12 EXPORTS
13     ActivateTraceMode,
14     DeactivateTraceMode,
15     TraceSubscriberActivity,
16     NoteInternalHandover,
17     SendIMSI
18 ;
19
20 IMPORTS
21     OPERATION
22 FROM TCAPMessages {
23     ccitt recommendation q 773 modules (2) messages (1) version2 (2)}
24
25     SystemFailure,
26     DataMissing,
27     UnexpectedDataValue,
28     FacilityNotSupported,
29     UnknownSubscriber,
30     UnidentifiedSubscriber,
31     TracingBufferFull
32 FROM MAP-Errors {
33     ccitt identified-organization (4) etsi (0) mobileDomain (0)
34     gsm-Network (1) modules (3) map-Errors (10) version2 (2)}
35
36     ActivateTraceModeArg,
37     DeactivateTraceModeArg,
38     TraceSubscriberActivityArg,
39     NoteInternalHO-Arg
40 FROM MAP-OM-DataTypes {
41     ccitt identified-organization (4) etsi (0) mobileDomain (0)
42     gsm-Network (1) modules (3) map-OM-DataTypes (12) version2 (2)}
43
44     ISDN-AddressString,
45     IMSI
46 FROM MAP-CommonDataTypes {
47     ccitt identified-organization (4) etsi (0) mobileDomain (0)
48     gsm-Network (1) modules (3) map-ComonDataTypes (18) version2 (2)}
49 ;
50
51
52 ActivateTraceMode ::= OPERATION                              --Timer m
53     ARGUMENT
54         activateTraceModeArg        ActivateTraceModeArg
55     RESULT
56     ERRORS {
57         SystemFailure,
58         DataMissing,
59         UnexpectedDataValue,
60         FacilityNotSupported,
61         UnidentifiedSubscriber,
62         TracingBufferFull}
```

```
63
64  DeactivateTraceMode ::= OPERATION                                    --Timer m
65      ARGUMENT
66          deactivateTraceModeArg      DeactivateTraceModeArg
67      RESULT
68      ERRORS {
69          SystemFailure,
70          DataMissing,
71          UnexpectedDataValue,
72          FacilityNotSupported,
73          UnidentifiedSubscriber}
74
75  TraceSubscriberActivity ::= OPERATION                                --Timer s
76      ARGUMENT
77          traceSubscriberActivityArg  TraceSubscriberActivityArg
78
79  NoteInternalHandover ::= OPERATION                                   --Timer s
80      ARGUMENT
81          noteInternalHO-Arg          NoteInternalHO-Arg
82
83  SendIMSI ::= OPERATION                                               --Timer m
84      ARGUMENT
85          msisdn                      ISDN-AddressString
86      RESULT
87          imsi                        IMSI
88      ERRORS {
89          DataMissing,
90          UnexpectedDataValue,
91          UnknownSubscriber}
92
93  END
```

14.6.3 Call Handling Operations

```
 1 MAP-CallHandlingOperations {
 2     ccitt identified-organization (4) etsi (0) mobileDomain (0)
 3     gsm-Network (1) modules (3) map-CallHandlingOperations (7)
 4     version2 (2)}
 5
 6 DEFINITIONS
 7
 8 ::=
 9
10 BEGIN
11
12 EXPORTS
13     SendRoutingInfo,
14     ProvideRoamingNumber
15 ;
16
17 IMPORTS
18     OPERATION
19 FROM TCAPMessages {
20     ccitt recommendation q 773 modules (2) messages (1) version2 (2)}
21
22     SystemFailure,
23     DataMissing,
24     UnexpectedDataValue,
25     FacilityNotSupported,
26     UnknownSubscriber,
27     NumberChanged,
28     BearerServiceNotProvisioned,
29     TeleserviceNotProvisioned,
30     NoRoamingNumberAvailable,
31     AbsentSubscriber,
32     CallBarred,
33     ForwardingViolation,
34     CUG-Reject
35 FROM MAP-Errors {
36     ccitt identified-organization (4) etsi (0) mobileDomain (0)
37     gsm-Network (1) modules (3) map-Errors (10) version2 (2)}
38     SendRoutingInfoArg,
39     SendRoutingInfoRes,
40     ProvideRoamingNumberArg
41 FROM MAP-CH-DataTypes {
42     ccitt identified-organization (4) etsi (0) mobileDomain (0)
43     gsm-Network (1) modules (3) map-CH-DataTypes (13) version2 (2)}
44
45     ISDN-AddressString
46 FROM MAP-CommonDataTypes {
47     ccitt identified-organization (4) etsi (0) mobileDomain (0)
48     gsm-Network (1) modules (3) map-CommonDataTypes (18) version2 (2)}
49 ;
50
```

```
51  SendRoutingInfo ::= OPERATION                                    --Timer m
52      ARGUMENT
53          sendRoutingInfoArg           SendRoutingInfoArg
54      RESULT
55          sendRoutingInfoRes           SendRoutingInfoRes
56      ERRORS {
57          SystemFailure,
58          DataMissing,
59          UnexpectedDataValue,
60          FacilityNotSupported,
61          UnknownSubscriber,
62          NumberChanged,
63          -- NumberChanged must not be used in version 1
64      BearerServiceNotProvisioned,
65          TeleserviceNotProvisioned,
66          AbsentSubscriber,
67          CallBarred,
68          CUG-Reject,
69          -- CUG-Reject must not be used in version 1
70          ForwardingViolation}
71
72  ProvideRoamingNumber ::= OPERATION                               --Timer m
73      ARGUMENT
74          provideRoamingNumberArg      ProvideRoamingNumberArg
75      RESULT
76          roamingNumber                ISDN-AddressString
77      ERRORS {
78          SystemFailure,
79          DataMissing,
80          UnexpectedDataValue,
81          FacilityNotSupported,
82          AbsentSubscriber,
83          NoRoamingNumberAvailable}
84
85  END
```

14.6.4 Supplementary service operations

```
1  MAP-SupplementaryServiceOperations {
2      ccitt identified-organization (4) etsi (0) mobileDomain (0)
3      gsm-Network (1) modules (3) map-SupplementaryServiceOperations (8)
4      version2 (2)}
5
6  DEFINITIONS
7
8  ::=
9
10 BEGIN
11
12 EXPORTS
13      RegisterSS,
14      EraseSS,
15      ActivateSS,
16      DeactivateSS,
17      InterrogateSS,
18      ProcessUnstructuredSS-Data,
19      ProcessUnstructuredSS-Request,
20      UnstructuredSS-Request,
21      UnstructuredSS-Notify,
22      RegisterPassword,
23      GetPassword,
24      BeginSubscriberActivity
25 ;
26
27 IMPORTS
28      OPERATION
29 FROM TCAPMessages {
30    ccitt recommendation q 773 modules (2) messages (1) version2 (2)}
31
32      SystemFailure,
33      DataMissing,
34      UnexpectedDataValue,
35      UnknownSubscriber,
36      BearerServiceNotProvisioned,
37      TeleserviceNotProvisioned,
38      CallBarred,
39      IllegalSS-Operation,
40      SS-ErrorStatus,
41      SS-NotAvailable,
42      SS-SubscriptionViolation,
43      SS-Incompatibility,
44      PW-RegistrationFailure,
45      NegativePW-Check,
46      NumberOfPW-AttemptsViolation,
47      UnknownAlphabet,
48      USSD-Busy,
49      AbsentSubscriber,
50      IllegalSubscriber,
51      IllegalEquipment
52 FROM MAP-Errors {
53    ccitt identified-organization (4) etsi (0) mobileDomain (0)
54    gsm-Network (1) modules (3) map-Errors (10) version2 (2)}
55
56      RegisterSS-Arg,
57      SS-Info,
58      SS-ForBS-Code,
59      InterrogateSS-Res,
60      SS-UserData,
61      USSD-Arg,
62      USSD-Res,
63      Password,
```

```
64          GuidanceInfo,
65          BeginSubscriberActivityArg
66  FROM MAP-SS-DataTypes {
67     ccitt identified-organization (4) etsi (0) mobileDomain (0)
68     gsm-Network (1) modules (3) map-SS-DataTypes (14) version2 (2)}
69
70          SS-Code
71  FROM MAP-SS-Code {
72     ccitt identified-organization (4) etsi (0) mobileDomain (0)
73     gsm-Network (1) modules (3) map-SS-Code (15) version2 (2)}
74  ;
75
76
77  -- supplementary service handling operations
78
79  RegisterSS ::= OPERATION                                    --Timer m
80      ARGUMENT
81          registerSS-Arg              RegisterSS-Arg
82      RESULT
83          ss-Info                     SS-Info
84          -- optional
85      ERRORS {
86          SystemFailure,
87          DataMissing,
88          UnexpectedDataValue,
89          UnknownSubscriber,
90          -- UnknownSubscriber must not be used in version greater 1
91          BearerServiceNotProvisioned,
92          TeleserviceNotProvisioned,
93          CallBarred,
94          -- CallBarred must not be used in version 1
95          IllegalSS-Operation,
96          SS-ErrorStatus,
97          SS-SubscriptionViolation,
98          -- SS-SubscriptionViolation must not be used in version
99          -- greater 1
100         SS-Incompatibility}
101
102  EraseSS ::= OPERATION                                       --Timer m
103      ARGUMENT
104          ss-ForBS                    SS-ForBS-Code
105      RESULT
106          ss-Info                     SS-Info
107          -- optional
108      ERRORS {
109          SystemFailure,
110          DataMissing,
111          -- DataMissing must not be used in version 1
112          UnexpectedDataValue,
113          UnknownSubscriber,
114          -- UnknownSubscriber must not be used in version greater 1
115          BearerServiceNotProvisioned,
116          -- BearerServiceNotProvisioned must not be used in version 1
117          TeleserviceNotProvisioned,
118          -- TeleserviceNotProvisioned must not be used in version 1
119          CallBarred,
120          -- CallBarred must not be used in version 1
121          IllegalSS-Operation,
122          SS-ErrorStatus,
123          SS-SubscriptionViolation
124          -- SS-SubscriptionViolation must not be used in version
125          -- greater 1
126          }
127
```

```
128  ActivateSS ::= OPERATION                                              --Timer m
129      ARGUMENT
130          ss-ForBS                    SS-ForBS-Code
131      RESULT
132          ss-Info                     SS-Info
133          -- optional
134      ERRORS {
135          SystemFailure,
136          DataMissing,
137          -- DataMissing must not be used in version 1
138          UnexpectedDataValue,
139          UnknownSubscriber,
140          -- UnknownSubscriber must not be used in version greater 1
141          BearerServiceNotProvisioned,
142          -- BearerServiceNotProvisioned must not be used in version 1
143          TeleserviceNotProvisioned,
144          -- TeleserviceNotProvisioned must not be used in version 1
145          CallBarred,
146          -- CallBarred must not be used in version 1
147          IllegalSS-Operation,
148          SS-ErrorStatus,
149          SS-SubscriptionViolation,
150          SS-Incompatibility,
151          NegativePW-Check,
152          NumberOfPW-AttemptsViolation}
153          -- NumberOfPW-AttemptsViolation must not be used in version 1
154
```

```
155  DeactivateSS ::= OPERATION                                            --Timer m
156      ARGUMENT
157          ss-ForBS                    SS-ForBS-Code
158      RESULT
159          ss-Info                     SS-Info
160          -- optional
161      ERRORS {
162          SystemFailure,
163          DataMissing,
164          -- DataMissing must not be used in version 1
165          UnexpectedDataValue,
166          UnknownSubscriber,
167          -- UnknownSubscriber must not be used in version greater 1
168          BearerServiceNotProvisioned,
169          -- BearerServiceNotProvisioned must not be used in version 1
170          TeleserviceNotProvisioned,
171          -- TeleserviceNotProvisioned must not be used in version 1
172          CallBarred,
173          -- CallBarred must not be used in version 1
174          IllegalSS-Operation,
175          SS-ErrorStatus,
176          SS-SubscriptionViolation,
177          NegativePW-Check,
178          NumberOfPW-AttemptsViolation}
179          -- NumberOfPW-AttemptsViolation must not be used in version 1
180
```

```
181  InterrogateSS ::= OPERATION                                              --Timer m
182      ARGUMENT
183          ss-ForBS                    SS-ForBS-Code
184      RESULT
185          interrogateSS-Res           InterrogateSS-Res
186      ERRORS {
187          SystemFailure,
188          DataMissing,
189          -- DataMissing must not be used in version 1
190          UnexpectedDataValue,
191          UnknownSubscriber,
192          -- UnknownSubscriber must not be used in version greater 1
193          BearerServiceNotProvisioned,
194          -- BearerServiceNotProvisioned must not be used in version 1
195          TeleserviceNotProvisioned,
196          -- TeleserviceNotProvisioned must not be used in version 1
197          CallBarred,
198          -- CallBarred must not be used in version 1
199          IllegalSS-Operation,
200          SS-NotAvailable}
201
202  ProcessUnstructuredSS-Data ::= OPERATION     --Timer m
203      ARGUMENT
204          ss-UserData                 SS-UserData
205      RESULT
206          ss-UserData                 SS-UserData
207          -- optional
208      ERRORS {
209          SystemFailure,
210          UnexpectedDataValue}
211
212  ProcessUnstructuredSS-Request ::= OPERATION  --Timer ml
213      ARGUMENT
214          ussd-Arg                    USSD-Arg
215      RESULT
216          ussd-Res                    USSD-Res
217      ERRORS {
218          SystemFailure,
219          DataMissing,
220          UnexpectedDataValue,
221          UnknownAlphabet,
222          CallBarred}
223
224  UnstructuredSS-Request ::= OPERATION     --Timer 10 minutes
225      ARGUMENT
226          ussd-Arg                    USSD-Arg
227      RESULT
228          ussd-Res                    USSD-Res
229          -- optional
230      ERRORS {
231          SystemFailure,
232          DataMissing,
233          UnexpectedDataValue,
234          AbsentSubscriber,
235          IllegalSubscriber,
236          IllegalEquipment,
237          UnknownAlphabet,
238          USSD-Busy}
239
```

```
240 UnstructuredSS-Notify ::= OPERATION        --Timer ml
241     ARGUMENT
242         ussd-Arg                 USSD-Arg
243     RESULT
244     ERRORS {
245         SystemFailure,
246         DataMissing,
247         UnexpectedDataValue,
248         AbsentSubscriber,
249         IllegalSubscriber,
250         IllegalEquipment,
251         UnknownAlphabet,
252         USSD-Busy}
253
254 RegisterPassword ::= OPERATION              --Timer ml
255     ARGUMENT
256         ss-Code                  SS-Code
257     RESULT
258         newPassword              Password
259     ERRORS {
260         SystemFailure,
261         DataMissing,
262         -- DataMissing must not be used in version 1
263         UnexpectedDataValue,
264         CallBarred,
265         -- CallBarred must not be used in version 1
266         SS-SubscriptionViolation,
267         PW-RegistrationFailure,
268         NegativePW-Check,
269         NumberOfPW-AttemptsViolation}
270         -- NumberOfPW-AttemptsViolation must not be used in version 1
271     LINKED {
272         GetPassword}
273
274 GetPassword ::= OPERATION                                    --Timer m
275     ARGUMENT
276         guidanceInfo             GuidanceInfo
277     RESULT
278         currentPassword          Password
279
280 BeginSubscriberActivity ::= OPERATION                       --Timer m
281     ARGUMENT
282         beginSubscriberActivityArg  BeginSubscriberActivityArg
283
284 END
```

14.6.5 Short message service operations

```
1  MAP-ShortMessageServiceOperations {
2      ccitt identified-organization (4) etsi (0) mobileDomain (0)
3      gsm-Network (1) modules (3) map-ShortMessageServiceOperations (9)
4      version2 (2)}
5
6  DEFINITIONS
7
8  ::=
9
10 BEGIN
11
12 EXPORTS
13      SendRoutingInfoForSM,
14      ForwardSM,
15      ReportSM-DeliveryStatus,
16      NoteSubscriberPresent,
17      AlertServiceCentreWithoutResult,
18      AlertServiceCentre,
19      InformServiceCentre,
20      ReadyForSM
21 ;
22
23 IMPORTS
24      OPERATION
25 FROM TCAPMessages {
26     ccitt recommendation q 773 modules (2) messages (1) version2 (2)}
27
28      SystemFailure,
29      DataMissing,
30      UnexpectedDataValue,
31      FacilityNotSupported,
32      UnknownSubscriber,
33      UnidentifiedSubscriber,
34      IllegalSubscriber,
35      IllegalEquipment,
36      TeleserviceNotProvisioned,
37      AbsentSubscriber,
38      CallBarred,
39      SubscriberBusyForMT-SMS,
40      SM-DeliveryFailure,
41      MessageWaitingListFull
42 FROM MAP-Errors {
43     ccitt identified-organization (4) etsi (0) mobileDomain (0)
44     gsm-Network (1) modules (3) map-Errors (10) version2 (2)}
45
46      RoutingInfoForSM-Arg,
47      RoutingInfoForSM-Res,
48      ForwardSM-Arg,
49      ReportSM-DeliveryStatusArg,
50      AlertServiceCentreArg,
51      InformServiceCentreArg,
52      ReadyForSM-Arg
53 FROM MAP-SM-DataTypes {
54     ccitt identified-organization (4) etsi (0) mobileDomain (0)
55     gsm-Network (1) modules (3) map-SM-DataTypes (16) version2 (2)}
56
57      ISDN-AddressString,
58      IMSI
59 FROM MAP-CommonDataTypes {
60     ccitt identified-organization (4) etsi (0) mobileDomain (0)
61     gsm-Network (1) modules (3) map-CommonDataTypes (18) version2 (2)}
62 ;
63
64
```

```
65  SendRoutingInfoForSM ::= OPERATION                              --Timer m
66      ARGUMENT
67          routingInfoForSM-Arg        RoutingInfoForSM-Arg
68      RESULT
69          routingInfoForSM-Res        RoutingInfoForSM-Res
70      ERRORS {
71          SystemFailure,
72          DataMissing,
73          UnexpectedDataValue,
74          FacilityNotSupported,
75          UnknownSubscriber,
76          TeleserviceNotProvisioned,
77          AbsentSubscriber,
78          CallBarred}
79
80  ForwardSM ::= OPERATION                      --Timer ml
81      ARGUMENT
82          forwardSM-Arg               ForwardSM-Arg
83      RESULT
84      ERRORS {
85          SystemFailure,
86          DataMissing,
87          -- DataMissing must not be used in version 1
88          UnexpectedDataValue,
89          FacilityNotSupported,
90          UnidentifiedSubscriber,
91          IllegalSubscriber,
92          IllegalEquipment,
93          -- IllegalEquipment must not be used in version 1
94          AbsentSubscriber,
95          SubscriberBusyForMT-SMS,
96          -- SubscriberBusyForMT-SMS must not be used in version 1
97          SM-DeliveryFailure}
98
99  ReportSM-DeliveryStatus ::= OPERATION                           --Timer s
100     ARGUMENT
101         reportSM-DeliveryStatusArg  ReportSM-DeliveryStatusArg
102     RESULT
103         storedMSISDN                ISDN-AddressString
104         -- optional
105         -- storedMSISDN must be absent in version 1
106         -- storedMSISDN must be present in version greater 1
107     ERRORS {
108         DataMissing,
109         -- DataMissing must not be used in version 1
110         UnexpectedDataValue,
111         UnknownSubscriber,
112         MessageWaitingListFull}
113
114 NoteSubscriberPresent ::= OPERATION                             --Timer s
115     ARGUMENT
116         imsi                        IMSI
117
118 AlertServiceCentreWithoutResult ::= OPERATION                   --Timer s
119     ARGUMENT
120         alertServiceCentreArg       AlertServiceCentreArg
121
```

```
122  AlertServiceCentre ::= OPERATION                              --Timer s
123      ARGUMENT
124          alertServiceCentreArg         AlertServiceCentreArg
125      RESULT
126      ERRORS {
127          SystemFailure,
128          DataMissing,
129          UnexpectedDataValue}
130
131  InformServiceCentre ::= OPERATION                             --Timer s
132      ARGUMENT
133          informServiceCentreArg        InformServiceCentreArg
134
135  ReadyForSM ::= OPERATION                                      --Timer m
136      ARGUMENT
137          readyForSM-Arg                ReadyForSM-Arg
138      RESULT
139      ERRORS {
140          DataMissing,
141          UnexpectedDataValue,
142          FacilityNotSupported,
143          UnknownSubscriber}
144
145  END
```

14.6.6   Errors

```
1   MAP-Errors {
2       ccitt identified-organization (4) etsi (0) mobileDomain (0)
3       gsm-Network (1) modules (3) map-Errors (10) version2 (2)}
4
5   DEFINITIONS
6
7   ::=
8
9   BEGIN
10
11  EXPORTS
12
13          -- generic errors
14          SystemFailure,
15          DataMissing,
16          UnexpectedDataValue,
17          FacilityNotSupported,
18
19          -- identification and numbering errors
20          UnknownSubscriber,
21          NumberChanged,
22          UnknownBaseStation,
23          UnknownMSC,
24          UnidentifiedSubscriber,
25          UnknownEquipment,
26
27          -- subscription errors
28          RoamingNotAllowed,
29          IllegalSubscriber,
30          IllegalEquipment,
31          BearerServiceNotProvisioned,
32          TeleserviceNotProvisioned,
33
34          -- handover errors
35          InvalidTargetBaseStation,
36          NoRadioResourceAvailable,
37          NoHandoverNumberAvailable,
38          SubsequentHandoverFailure,
39
40          -- operation and maintenance errors
41          TracingBufferFull,
42
43          -- call handling errors
44          NoRoamingNumberAvailable,
45          AbsentSubscriber,
46          CallBarred,
47          ForwardingViolation,
48          CUG-Reject,
49
50          -- supplementary service errors
51          IllegalSS-Operation,
52          SS-ErrorStatus,
53          SS-NotAvailable,
54          SS-SubscriptionViolation,
55          SS-Incompatibility,
56          UnknownAlphabet,
57          USSD-Busy,
58          PW-RegistrationFailure,
59          NegativePW-Check,
60          NumberOfPW-AttemptsViolation,
61
62          -- short message service errors
63          SubscriberBusyForMT-SMS,
64          SM-DeliveryFailure,
65          MessageWaitingListFull
66  ;
```

```
 67
 68  IMPORTS
 69       ERROR
 70  FROM TCAPMessages {
 71     ccitt recommendation q 773 modules (2) messages (1) version2 (2)}
 72
 73       SS-Status,
 74       SS-SubscriptionOption
 75  FROM MAP-SS-DataTypes {
 76     ccitt identified-organization (4) etsi (0) mobileDomain (0)
 77     gsm-Network (1) modules (3) map-SS-DataTypes (14) version2 (2)}
 78
 79       RoamingNotAllowedCause,
 80       CallBarringCause,
 81       CUG-RejectCause,
 82       SS-IncompatibilityCause,
 83       PW-RegistrationFailureCause,
 84       SM-DeliveryFailureCause
 85  FROM MAP-ER-DataTypes {
 86     ccitt identified-organization (4) etsi (0) mobileDomain (0)
 87     gsm-Network (1) modules (3) map-ER-DataTypes (17) version2 (2)}
 88
 89       NetworkResource
 90  FROM MAP-CommonDataTypes {
 91     ccitt identified-organization (4) etsi (0) mobileDomain (0)
 92     gsm-Network (1) modules (3) map-CommonDataTypes (18) version2 (2)}
 93  ;
 94
 95
 96  -- generic errors
 97
 98  SystemFailure ::= ERROR
 99      PARAMETER
100         networkResource              NetworkResource
101         -- optional
102
103  DataMissing ::= ERROR
104
105  UnexpectedDataValue ::= ERROR
106
107  FacilityNotSupported ::= ERROR
108
109
110  -- identification and numbering errors
111
112  UnknownSubscriber ::= ERROR
113
114  NumberChanged ::= ERROR
115
116  UnknownBaseStation ::= ERROR
117
118  UnknownMSC ::= ERROR
119
120  UnidentifiedSubscriber ::= ERROR
121
122  UnknownEquipment ::= ERROR
123
```

```
124
125  -- subscription errors
126
127  RoamingNotAllowed ::= ERROR
128       PARAMETER
129            roamingNotAllowedCause        RoamingNotAllowedCause
130            -- optional
131            -- roamingNotAllowedCause must be absent in version 1
132            -- roamingNotAllowedCause must be present in version greater 1
133
134  IllegalSubscriber ::= ERROR
135
136  IllegalEquipment ::= ERROR
137
138  BearerServiceNotProvisioned ::= ERROR
139
140  TeleserviceNotProvisioned ::= ERROR
141
142
143  -- handover errors
144
145  InvalidTargetBaseStation ::= ERROR
146
147  NoRadioResourceAvailable ::= ERROR
148
149  NoHandoverNumberAvailable ::= ERROR
150
151  SubsequentHandoverFailure ::= ERROR
152
153
154  -- operation and maintenance errors
155
156  TracingBufferFull ::= ERROR
157
158
159  -- call handling errors
160
161  NoRoamingNumberAvailable ::= ERROR
162
163  AbsentSubscriber ::= ERROR
164       PARAMETER
165            mwd-Set                       BOOLEAN
166            -- optional
167            -- mwd-Set must be absent in version greater 1
168
169  CallBarred ::= ERROR
170       PARAMETER
171            callBarringCause         CallBarringCause
172            -- optional
173
174  ForwardingViolation ::= ERROR
175
176  CUG-Reject ::= ERROR
177       PARAMETER
178            cug-RejectCause          CUG-RejectCause
179            -- optional
180
```

```
181
182  -- supplementary service errors
183
184  IllegalSS-Operation ::= ERROR
185
186  SS-ErrorStatus ::= ERROR
187      PARAMETER
188          ss-Status                  SS-Status
189          -- optional
190
191  SS-NotAvailable ::= ERROR
192
193  SS-SubscriptionViolation ::= ERROR
194      PARAMETER
195          ss-SubscriptionOption      SS-SubscriptionOption
196          -- optional
197          -- ss-SubscriptionOption must not be used in version greater 1
198
199  SS-Incompatibility ::= ERROR
200      PARAMETER
201          ss-IncompatibilityCause    SS-IncompatibilityCause
202          -- optional
203
204  UnknownAlphabet ::= ERROR
205
206  USSD-Busy ::= ERROR
207
208  PW-RegistrationFailure ::= ERROR
209      PARAMETER
210          pw-RegistrationFailureCause PW-RegistrationFailureCause
211
212  NegativePW-Check ::= ERROR
213
214  NumberOfPW-AttemptsViolation ::= ERROR
215
216
217  -- short message service errors
218
219  SubscriberBusyForMT-SMS ::= ERROR
220
221  SM-DeliveryFailure ::= ERROR
222      PARAMETER
223          sm-DeliveryFailureCause     SM-DeliveryFailureCause
224
225  MessageWaitingListFull ::= ERROR
226
227  END
```

14.7 MAP constants and data types

14.7.1 Mobile Service data types

```
1  MAP-MS-DataTypes {
2      ccitt identified-organization (4) etsi (0) mobileDomain (0)
3      gsm-Network (1) modules (3) map-MS-DataTypes (11) version2 (2)}
4
5  DEFINITIONS
6
7  IMPLICIT TAGS
8
9  ::=
10
11 BEGIN
12
13 EXPORTS
14
15      -- location registration types
16      UpdateLocationArg,
17      UpdateLocationRes,
18      CancelLocationArg,
19      PurgeMS-Arg,
20      SendIdentificationRes,
21
22      -- handover types
23      PrepareHO-Arg,
24      PrepareHO-Res,
25      PrepareSubsequentHO-Arg,
26      PerformHO-Arg,
27      PerformHO-Res,
28      PerformSubsequentHO-Arg,
29
30      -- authentication management types
31      SendAuthenticationInfoArg,
32      SendAuthenticationInfoRes,
33      Ki,
34
35      -- security management types
36      EquipmentStatus,
37
38      -- subscriber management types
39      SendParametersArg,
40      SentParameterList,
41      InsertSubscriberDataArg,
42      InsertSubscriberDataRes,
43      DeleteSubscriberDataArg,
44      DeleteSubscriberDataRes,
45      SubscriberData,
46      ODB-Data,
47      SubscriberStatus,
48      ZoneCodeList,
49      maxNumOfZoneCodes,
50
51      -- fault recovery types
52      ResetArg,
53      RestoreDataArg,
54      RestoreDataRes
55  ;
56
```

```
57  IMPORTS
58       SS-List,
59       SS-InfoList
60  FROM MAP-SS-DataTypes {
61     ccitt identified-organization (4) etsi (0) mobileDomain (0)
62     gsm-Network (1) modules (3) map-SS-DataTypes (14) version2 (2)}
63
64       ISDN-AddressString,
65       ExternalSignalInfo,
66       IMSI,
67       SubscriberId,
68       IMEI,
69       LocationInfo,
70       HLR-List,
71       LMSI,
72       GlobalCellId,
73       NetworkResource,
74       Category,
75       BearerServiceList,
76       TeleserviceList,
77       BasicServiceList
78  FROM MAP-CommonDataTypes {
79     ccitt identified-organization (4) etsi (0) mobileDomain (0)
80     gsm-Network (1) modules (3) map-CommonDataTypes (18) version2 (2)}
81  ;
82
83
84  -- location registration types
85
86  UpdateLocationArg ::= SEQUENCE {
87      imsi                          IMSI,
88      locationInfo                  LocationInfo,
89      vlr-Number                    ISDN-AddressString,
90      lmsi                          [10] LMSI                      OPTIONAL,
91      ...}
92
93  UpdateLocationRes ::= CHOICE {
94      hlr-Number                    ISDN-AddressString,
95      -- hlr-Number must not be used in version greater 1
96  extensibleUpdateLocationRes       ExtensibleUpdateLocationRes}
97      -- extensibleUpdateLocationRes must not be used in version 1
98
99  ExtensibleUpdateLocationRes ::= SEQUENCE {
100     hlr-Number                    ISDN-AddressString,
101     ...}
102
103 CancelLocationArg ::= CHOICE {
104     imsi                          IMSI,
105     imsi-WithLMSI                 IMSI-WithLMSI}
106
107 PurgeMS-Arg ::= SEQUENCE {
108     imsi                          IMSI,
109     vlr-Number                    ISDN-AddressString,
110     ...}
111
112 IMSI-WithLMSI ::= SEQUENCE {
113     imsi                          IMSI,
114     lmsi                          LMSI,
115     -- a special value 00000000 indicates that the LMSI is not in use
116     ...}
117
118 SendIdentificationRes ::= SEQUENCE {
119     imsi                          IMSI,
120     authenticationSetList         AuthenticationSetList        OPTIONAL,
121     ...}
122
```

```
123 AuthenticationSetList ::= SEQUENCE SIZE (1..5) OF
124                                     AuthenticationSet
125
126 AuthenticationSet ::= SEQUENCE {
127     rand                            RAND,
128     sres                            SRES,
129     kc                              Kc,
130     ...}
131
132 RAND ::= OCTET STRING (SIZE (16))
133
134 SRES ::= OCTET STRING (SIZE (4))
135
136 Kc ::= OCTET STRING (SIZE (8))
137
138 -- handover types
139
140 PrepareHO-Arg ::= SEQUENCE {
141     targetCellId                    GlobalCellId                OPTIONAL,
142     ho-NumberNotRequired            NULL                        OPTIONAL,
143     bss-APDU                        ExternalSignalInfo          OPTIONAL,
144     ...}
145
146 PerformHO-Arg ::= SEQUENCE {
147     targetCellId                    GlobalCellId,
148     servingCellId                   GlobalCellId,
149     channelType                     ChannelType,
150     classmarkInfo                   ClassmarkInfo,
151     handoverPriority                [11] HandoverPriority       OPTIONAL,
152     kc                              [12] Kc                     OPTIONAL}
153
154 ChannelType ::= OCTET STRING (SIZE (1..10))
155     -- concatenation of
156     -- speech data indicator
157     -- channel rate
158     -- speech encoding algorithm/ data rate
159     -- and transparency indicator
160     -- as defined in GSM 08.08
161
162 ClassmarkInfo ::= OCTET STRING (SIZE (1..2))
163     -- classmark information element type 1 or 2 contents as defined
164     -- in GSM 08.08 (phase 1)
165
166 HandoverPriority ::= OCTET STRING (SIZE (1))
167     -- The internal structure is defined in GSM 08.08.
168
169 PrepareHO-Res ::= SEQUENCE {
170     handoverNumber                  ISDN-AddressString          OPTIONAL,
171     bss-APDU                        ExternalSignalInfo          OPTIONAL,
172     ...}
173
174 PerformHO-Res ::= SEQUENCE {
175     handoverNumber                  ISDN-AddressString,
176     accessSignalInfo                ExternalSignalInfo}
177
```

```
178  PrepareSubsequentHO-Arg ::= SEQUENCE {
179      targetCellId                    GlobalCellId,
180      targetMSC-Number                ISDN-AddressString,
181      bss-APDU                        ExternalSignalInfo,
182      ...}
183
184  PerformSubsequentHO-Arg ::= SEQUENCE {
185      targetCellId                    GlobalCellId,
186      servingCellId                   GlobalCellId,
187      targetMSC-Number                ISDN-AddressString,
188      classmarkInfo                   [10] ClassmarkInfo        OPTIONAL}
189
190  -- authentication management types
191
192  SendAuthenticationInfoArg ::= IMSI
193
194  SendAuthenticationInfoRes ::= AuthenticationSetList
195
196  Ki ::= OCTET STRING (SIZE (16))
197
198
199  -- security management types
200
201  EquipmentStatus ::= ENUMERATED {
202      whiteListed  (0),
203      blackListed  (1),
204      greyListed   (2)}
205
206
207  -- subscriber management types
208
209  SendParametersArg ::= SEQUENCE {
210      subscriberId                    SubscriberId,
211      requestParameterList            RequestParameterList}
212
213  RequestParameterList ::= SEQUENCE SIZE (1..2) OF
214                                  RequestParameter
215
216  RequestParameter ::= ENUMERATED {
217      requestIMSI  (0),
218      requestAuthenticationSet  (1),
219      requestSubscriberData  (2),
220      requestKi  (4)}
221
222  SentParameterList ::= SEQUENCE SIZE (1..maxNumOfSentParameter) OF
223                                  SentParameter
224
225  SentParameter ::= CHOICE {
226      imsi                            [0] IMSI,
227      authenticationSet               [1] AuthenticationSet,
228      subscriberData                  [2] SubscriberData,
229      ki                              [4] Ki}
230
231  maxNumOfSentParameter   INTEGER ::= 6
232      -- This NamedValue corresponds to the maximum number of
233      -- authentication set which may be returned by a HLR plus 1
234
```

```
235   InsertSubscriberDataArg ::= SEQUENCE {
236       imsi                          [0] IMSI                       OPTIONAL,
237       COMPONENTS OF                     SubscriberData,
238       ...}
239
240   SubscriberData ::= SEQUENCE {
241       msisdn                        [1] ISDN-AddressString         OPTIONAL,
242       category                      [2] Category                  OPTIONAL,
243       subscriberStatus              [3] SubscriberStatus          OPTIONAL,
244       bearerServiceList             [4] BearerServiceList         OPTIONAL,
245       teleserviceList               [6] TeleserviceList           OPTIONAL,
246       provisionedSS                 [7] SS-InfoList               OPTIONAL,
247       odb-Data                      [8] ODB-Data                  OPTIONAL,
248       -- odb-Data must be absent in version 1
249       roamingRestrictionDueToUnsupportedFeature  [9] NULL         OPTIONAL,
250       -- roamingRestrictionDueToUnsupportedFeature must be absent
251       -- in version 1
252       regionalSubscriptionData      [10] ZoneCodeList             OPTIONAL
253       -- regionalSubscriptionData must be absent in version 1
254       }
255
256   SubscriberStatus ::= ENUMERATED {
257       serviceGranted  (0),
258       operatorDeterminedBarring  (1)}
259
260   ODB-Data ::= SEQUENCE {
261       odb-GeneralData                   ODB-GeneralData,
262       odb-HPLMN-Data                    ODB-HPLMN-Data              OPTIONAL,
263       ...}
264
265   ODB-GeneralData ::= BIT STRING {
266       allOG-CallsBarred  (0),
267       internationalOGCallsBarred  (1),
268       internationalOGCallsNotToHPLMN-CountryBarred  (2),
269       premiumRateInformationOGCallsBarred  (3),
270       premiumRateEntertainementOGCallsBarred  (4),
271       ss-AccessBarred  (5)} (SIZE (6))
272
273   ODB-HPLMN-Data ::= BIT STRING {
274       plmn-SpecificBarringType1  (0),
275       plmn-SpecificBarringType2  (1),
276       plmn-SpecificBarringType3  (2),
277       plmn-SpecificBarringType4  (3)} (SIZE (4))
278
279   ZoneCodeList ::= SEQUENCE SIZE (1..maxNumOfZoneCodes)
280                                   OF ZoneCode
281
282   ZoneCode ::= OCTET STRING (SIZE (2))
283       -- internal structure is defined in GSM 03.03
284
285   maxNumOfZoneCodes    INTEGER ::= 10
286
```

```
287  InsertSubscriberDataRes ::= SEQUENCE {
288      teleserviceList                  [1] TeleserviceList          OPTIONAL,
289      bearerServiceList                [2] BearerServiceList        OPTIONAL,
290      ss-List                          [3] SS-List                  OPTIONAL,
291      odb-GeneralData                  [4] ODB-GeneralData          OPTIONAL,
292      regionalSubscriptionResponse     [5]
293              RegionalSubscriptionResponse                          OPTIONAL,
294      -- regionalSubscriptionResponse must be absent in version 1
295      ...}
296
297  RegionalSubscriptionResponse ::= ENUMERATED {
298      msc-AreaRestricted               (0),
299      tooManyZoneCodes                 (1),
300      zoneCodesConflict                (2),
301      regionalSubscNotSupported        (3)}
302
303  DeleteSubscriberDataArg ::= SEQUENCE {
304      imsi                             [0] IMSI,
305      basicServiceList                 [1] BasicServiceList          OPTIONAL,
306      ss-List                          [2] SS-List                   OPTIONAL,
307      roamingRestrictionDueToUnsupportedFeature [4] NULL             OPTIONAL,
308      -- roamingRestrictionDueToUnsupportedFeature must be absent
309      -- in version 1
310      regionalSubscriptionIdentifier [5] ZoneCode                   OPTIONAL,
311      -- regionalSubscriptionIdentifier must be absent in version 1
312      ...}
313
314  DeleteSubscriberDataRes ::= SEQUENCE {
315      regionalSubscriptionResponse     [0]
316              RegionalSubscriptionResponse                          OPTIONAL,
317      ...}
318
319  -- fault recovery types
320
321  ResetArg ::= SEQUENCE {
322      networkResource                  NetworkResource              OPTIONAL,
323      -- networkResource must be present in version 1
324      -- networkResource must be absent in version greater 1
325      hlr-Number                       ISDN-AddressString,
326      hlr-List                         HLR-List                     OPTIONAL,
327      ...}
328
329  RestoreDataArg ::= SEQUENCE {
330      imsi                             IMSI,
331      lmsi                             LMSI                         OPTIONAL,
332      ...}
333
334  RestoreDataRes ::= SEQUENCE {
335      hlr-Number                       ISDN-AddressString,
336      msNotReachable                   NULL                         OPTIONAL,
337      ...}
338
339  END
```

14.7.2 Operation and maintenance data types

```
1  MAP-OM-DataTypes {
2      ccitt identified-organization (4) etsi (0) mobileDomain (0)
3      gsm-Network (1) modules (3) map-OM-DataTypes (12) version2 (2)}
4
5  DEFINITIONS
6
7  IMPLICIT TAGS
8
9  ::=
10
11 BEGIN
12
13 EXPORTS
14     ActivateTraceModeArg,
15     DeactivateTraceModeArg,
16     TraceSubscriberActivityArg,
17     NoteInternalHO-Arg
18 ;
19
20 IMPORTS
21     AddressString,
22     ExternalSignalInfo,
23     IMSI,
24     GlobalCellId
25 FROM MAP-CommonDataTypes {
26    ccitt identified-organization (4) etsi (0) mobileDomain (0)
27    gsm-Network (1) modules (3) map-CommonDataTypes (18) version2 (2)}
28
29 ;
30
31
32 ActivateTraceModeArg ::= SEQUENCE {
33     imsi                            [0] IMSI                    OPTIONAL,
34     traceReference                  [1] TraceReference,
35     traceType                       [2] TraceType,
36     omc-Id                          [3] AddressString           OPTIONAL,
37     ...}
38
39 TraceReference ::= OCTET STRING (SIZE (1..2))
40
41 TraceType ::= INTEGER
42     (0..255)
43     -- Trace types are fully defined in GSM 12.08.
44
45 DeactivateTraceModeArg ::= SEQUENCE {
46     imsi                            [0] IMSI                    OPTIONAL,
47     traceReference                  [1] TraceReference,
48     ...}
49
50 TraceSubscriberActivityArg ::= SEQUENCE {
51     imsi                            [0] IMSI                    OPTIONAL,
52     traceReference                  [1] TraceReference,
53     traceType                       [2] TraceType,
54     omc-Id                          [3] AddressString           OPTIONAL,
55     callReference                   [4] CallReference           OPTIONAL}
56
57 CallReference ::= OCTET STRING (SIZE (1..3))
```

```
58
59  NoteInternalHO-Arg ::= SEQUENCE {
60      handoverType                    HandoverType,
61      targetCellId                [1] GlobalCellId            OPTIONAL,
62      channelId                   [2] ExternalSignalInfo      OPTIONAL}
63
64  HandoverType ::= ENUMERATED {
65      interBSS   (0),
66      intraBSS   (1)}
67
68  END
```

14.7.3 Call handling data types

```
1  MAP-CH-DataTypes {
2      ccitt identified-organization (4) etsi (0) mobileDomain (0)
3      gsm-Network (1) modules (3) map-CH-DataTypes (13) version2 (2)}
4
5  DEFINITIONS
6
7  IMPLICIT TAGS
8
9  ::=
10
11 BEGIN
12
13 EXPORTS
14     SendRoutingInfoArg,
15     SendRoutingInfoRes,
16     ProvideRoamingNumberArg,
17     NumberOfForwarding
18 ;
19
20 IMPORTS
21     CUG-Interlock,
22     ForwardingData
23 FROM MAP-SS-DataTypes {
24     ccitt identified-organization (4) etsi (0) mobileDomain (0)
25     gsm-Network (1) modules (3) map-SS-DataTypes (14) version2 (2)}
26
27     ISDN-AddressString,
28     ExternalSignalInfo,
29     IMSI,
30     LMSI
31 FROM MAP-CommonDataTypes {
32     ccitt identified-organization (4) etsi (0) mobileDomain (0)
33     gsm-Network (1) modules (3) map-CommonDataTypes (18) version2 (2)}
34 ;
35
36
37 CUG-CheckInfo ::= SEQUENCE {
38     cug-Interlock                  CUG-Interlock,
39     cug-OutgoingAccess             NULL                        OPTIONAL,
40     ...}
41
42 NumberOfForwarding ::= INTEGER (1..5)
43
44 SendRoutingInfoArg ::= SEQUENCE {
45     msisdn                    [0] ISDN-AddressString,
46     cug-CheckInfo             [1] CUG-CheckInfo          OPTIONAL,
47     -- cug-CheckInfo must be absent in version 1
48     numberOfForwarding        [2] NumberOfForwarding     OPTIONAL,
49     networkSignalInfo         [10] ExternalSignalInfo    OPTIONAL,
50     ...}
51
52 SendRoutingInfoRes ::= SEQUENCE {
53     imsi                      IMSI,
54     routingInfo               RoutingInfo,
55     cug-CheckInfo             CUG-CheckInfo              OPTIONAL,
56     -- cug-CheckInfo must be absent in version 1
57     ...}
58
```

```
59  RoutingInfo ::= CHOICE {
60      roamingNumber                       ISDN-AddressString,
61      forwardingData                      ForwardingData}
62
63  ProvideRoamingNumberArg ::= SEQUENCE {
64      imsi                            [0] IMSI,
65      msc-Number                      [1] ISDN-AddressString      OPTIONAL,
66      -- msc-Number must be present in version greater 1
67      msisdn                          [2] ISDN-AddressString      OPTIONAL,
68      previousRoamingNumber           [3] ISDN-AddressString      OPTIONAL,
69      lmsi                            [4] LMSI                    OPTIONAL,
70      gsm-BearerCapability            [5] ExternalSignalInfo      OPTIONAL,
71      networkSignalInfo               [6] ExternalSignalInfo      OPTIONAL,
72      ...}
73
74  END
```

14.7.4 Supplementary service data types

```
1  MAP-SS-DataTypes {
2      ccitt identified-organization (4) etsi (0) mobileDomain (0)
3      gsm-Network (1) modules (3) map-SS-DataTypes (14) version2 (2)}
4
5  DEFINITIONS
6
7  IMPLICIT TAGS
8
9  ::=
10
11 BEGIN
12
13 EXPORTS
14      RegisterSS-Arg,
15      SS-Info,
16      SS-Status,
17      CUG-Index,
18      CUG-Interlock,
19      SS-SubscriptionOption,
20      SS-ForBS-Code,
21      InterrogateSS-Res,
22      SS-UserData,
23      USSD-Arg,
24      USSD-Res,
25      Password,
26      GuidanceInfo,
27      BeginSubscriberActivityArg,
28      SS-List,
29      SS-InfoList,
30      ForwardingData,
31      IntraCUG-Options,
32      InterCUG-Restrictions,
33      OverrideCategory,
34      CliRestrictionOption,
35      NoReplyConditionTime,
36      ForwardingOptions;
37
38 IMPORTS
39      AddressString,
40      ISDN-AddressString,
41      ISDN-SubaddressString,
42      maxSignalInfoLength,
43      IMSI,
44      BasicServiceCode,
45      BasicServiceGroupList,
46      maxNumOfBasicServiceGroups
47 FROM MAP-CommonDataTypes {
48     ccitt identified-organization (4) etsi (0) mobileDomain (0)
49     gsm-Network (1) modules (3) map-CommonDataTypes (18) version2 (2)}
50
51      SS-Code
52 FROM MAP-SS-Code {
53     ccitt identified-organization (4) etsi (0) mobileDomain (0)
54     gsm-Network (1) modules (3) map-SS-Code (15) version2 (2)}
55 ;
56
57
58 RegisterSS-Arg ::= SEQUENCE{
59     ss-Code                        SS-Code,
60     basicService                   BasicServiceCode           OPTIONAL,
61     forwardedToNumber              [4] AddressString          OPTIONAL,
62     forwardedToSubaddress          [6] ISDN-SubaddressString  OPTIONAL,
63     -- forwardedToSubaddress must be absent in version 1
64     noReplyConditionTime           [5] NoReplyConditionTime   OPTIONAL,
65     ...}
66
67 NoReplyConditionTime ::= INTEGER (5..30)
68
69 SS-Info ::= CHOICE {
70     forwardingInfo                 [0] ForwardingInfo,
71     callBarringInfo                [1] CallBarringInfo,
72     cug-Info                       [2] CUG-Info,
73         -- cug-info must not be used in version 1
74     ss-Data                        [3] SS-Data}
75
```

```
76  ForwardingInfo ::= SEQUENCE {
77      ss-Code                         SS-Code                     OPTIONAL,
78      forwardingFeatureList           ForwardingFeatureList,
79      ...}
80
81  ForwardingFeatureList ::=
82      SEQUENCE SIZE (1..maxNumOfBasicServiceGroups) OF
83                                      ForwardingFeature
84
85  ForwardingFeature ::= SEQUENCE {
86      basicService                    BasicServiceCode            OPTIONAL,
87      ss-Status                       [4] SS-Status               OPTIONAL,
88      -- ss-Status must be present in version 1
89      forwardedToNumber               [5] ISDN-AddressString      OPTIONAL,
90      forwardedToSubaddress           [8] ISDN-SubaddressString   OPTIONAL,
91      -- forwardedToSubaddress must be absent in version 1
92      forwardingOptions               [6] ForwardingOptions       OPTIONAL,
93      noReplyConditionTime            [7] NoReplyConditionTime    OPTIONAL,
94      ...}
95
96  SS-Status ::= OCTET STRING (SIZE (1))
97
98      -- bits 8765: 0000 (unused)
99      -- bits 4321: Used to convey the "P bit","R bit","A bit" and "Q bit",
100     --           representing supplementary service state information
101     --           as defined in GSM 03.11
102
103     -- bit 4: "Q bit"
104
105     -- bit 3: "P bit"
106
107     -- bit 2: "R bit"
108
109     -- bit 1: "A bit"
110
111 ForwardingOptions ::= OCTET STRING (SIZE (1))
112
113     -- v1: bits 87: notification to forwarding party
114     --   00  no notification
115     --   01  notification
116     --   10  notification
117
118     -- v2: bit 8: notification to forwarding party
119     --   0  no notification
120     --   1  notification
121
122     -- v2: bit 7: 0 (unused)
123
124     -- v1: bits 65: notification to calling party
125     --   00  no notification
126     --   01  notification
127     --   10  notification
128
129     -- v2: bit 6: v2: notification to calling party
130     --   0  no notification
131     --   1  notification
132
133     -- v2: bit 5: 0 (unused)
134
135     -- v2: bits 43: forwarding reason
136     --   00  ms not reachable
137     --   01  ms busy
138     --   10  no reply
139     --   11  unconditional
140     -- bits 21: 00 (unused)
141
```

```
142  CallBarringInfo ::= SEQUENCE {
143      ss-Code                         SS-Code                 OPTIONAL,
144      -- ss-Code must be present in version 1
145      callBarringFeatureList          CallBarringFeatureList,
146      ...}
147
148  CallBarringFeatureList ::=
149      SEQUENCE SIZE (1..maxNumOfBasicServiceGroups) OF
150                                      CallBarringFeature
151
152  CallBarringFeature ::= SEQUENCE {
153      basicService                    BasicServiceCode        OPTIONAL,
154      ss-Status                       [4] SS-Status           OPTIONAL,
155      -- ss-Status must be present in version 1
156      ...}
157
158  CUG-Info ::= SEQUENCE {
159      cug-SubscriptionList            CUG-SubscriptionList,
160      cug-FeatureList                 CUG-FeatureList         OPTIONAL,
161      ...}
162
163  CUG-SubscriptionList ::= SEQUENCE SIZE (1..maxNumOfCUG) OF
164                                      CUG-Subscription
165
166  CUG-Subscription ::= SEQUENCE {
167      cug-Index                       CUG-Index,
168      cug-Interlock                   CUG-Interlock,
169      intraCUG-Options                IntraCUG-Options,
170      basicServiceGroupList           BasicServiceGroupList   OPTIONAL,
171      ...}
172
173  CUG-Index ::= INTEGER (0..32767)
174      -- The internal structure is defined in ETS 300 138.
175
176  CUG-Interlock ::= OCTET STRING (SIZE (4))
177
178  IntraCUG-Options ::= ENUMERATED {
179      noCUG-Restrictions  (0),
180      cugIC-CallBarred  (1),
181      cugOG-CallBarred  (2)}
182
183  maxNumOfCUG   INTEGER ::= 10
184
185  CUG-FeatureList ::= SEQUENCE SIZE (1..maxNumOfBasicServiceGroups) OF
186                                      CUG-Feature
187
188  CUG-Feature ::= SEQUENCE {
189      basicService                    BasicServiceCode        OPTIONAL,
190      preferentialCUG-Indicator       CUG-Index               OPTIONAL,
191      interCUG-Restrictions           InterCUG-Restrictions,
192      ...}
193
```

```
194  InterCUG-Restrictions::= OCTET STRING (SIZE (1))
195
196      -- bits 876543: 000000 (unused)
197
198      -- bits 21
199      --  00  CUG only facilities
200      --  01  CUG with outgoing access
201      --  10  CUG with incoming access
202      --  11  CUG with both outgoing and incoming access
203
204  SS-Data ::= SEQUENCE {
205      ss-Code                         SS-Code                         OPTIONAL,
206      -- ss-Code must be present in version 1
207      ss-Status                   [4] SS-Status                       OPTIONAL,
208      ss-SubscriptionOption           SS-SubscriptionOption           OPTIONAL,
209      -- ss-SubscriptionOption must be absent in version 1
210      basicServiceGroupList           BasicServiceGroupList           OPTIONAL,
211      -- basicServiceGroupList must be absent in version 1
212      ...}
213
214  SS-SubscriptionOption ::= CHOICE {
215      cliRestrictionOption        [2] CliRestrictionOption,
216      overrideCategory            [1] OverrideCategory}
217
218  CliRestrictionOption ::= ENUMERATED {
219      permanent  (0),
220      temporaryDefaultRestricted  (1),
221      temporaryDefaultAllowed  (2)}
222
223  OverrideCategory ::= ENUMERATED {
224      overrideEnabled  (0),
225      overrideDisabled  (1)}
226
227  SS-ForBS-Code ::= SEQUENCE {
228      ss-Code                     SS-Code,
229      basicService                BasicServiceCode                    OPTIONAL,
230      ...}
231
232  Cli-RestrictionInfo ::= SEQUENCE {
233      ss-Status                   SS-Status,
234      cliRestrictionOption        CliRestrictionOption                OPTIONAL,
235      ...}
236
237  InterrogateSS-Res ::= CHOICE {
238      ss-Status                   [0] SS-Status,
239      forwardedToNumber           [1] ISDN-AddressString,
240      -- forwardedToNumber must not be used in version greater 1
241      basicServiceGroupList       [2] BasicServiceGroupList,
242      forwardingFeatureList       [3] ForwardingFeatureList,
243      cli-RestrictionInfo         [4] Cli-RestrictionInfo}
244      -- cli-RestrictionInfo must not be used in version 1
245
246  SS-UserData ::= IA5String (SIZE (1.. maxSignalInfoLength))
247
248  USSD-Arg ::= SEQUENCE {
249      ussd-DataCodingScheme           USSD-DataCodingScheme,
250      ussd-String                     USSD-String,
251      ...}
252
```

```
253 USSD-Res ::= SEQUENCE {
254     ussd-DataCodingScheme           USSD-DataCodingScheme,
255     ussd-String                     USSD-String,
256     ...}
257
258 USSD-DataCodingScheme ::= OCTET STRING (SIZE (1))
259     -- The structure of the USSD-DataCodingScheme is defined by
260     -- the Cell Broadcast Data Coding Scheme as described in
261     -- GSM 03.38
262
263 USSD-String ::= OCTET STRING (SIZE (1..maxUSSD-StringLength))
264     -- The structure of the contents of the USSD-String is dependent
265     -- on the USSD-DataCodingScheme as described in GSM 03.38.
266
267 maxUSSD-StringLength  INTEGER ::= 160
268
269 Password ::= NumericString
270     (FROM ("0"|"1"|"2"|"3"|"4"|"5"|"6"|"7"|"8"|"9"))
271     (SIZE (4))
272
273 GuidanceInfo ::= ENUMERATED {
274     enterPW  (0),
275     enterNewPW  (1),
276     enterNewPW-Again  (2),
277     badPW-TryAgain  (3),
278     -- badPW-TryAgain must not be used in version greater 1
279     badPW-FormatTryAgain  (4)}
280     -- badPW-FormatTryAgain must not be used in version greater 1
281     -- How this information is really delivered to the subscriber
282     -- (display, announcement, ...) is not part of this
283     -- specification.
284
285 BeginSubscriberActivityArg ::= SEQUENCE {
286     imsi                          IMSI,
287     originatingEntityNumber       ISDN-AddressString}
288
289 SS-List ::= SEQUENCE SIZE (1..maxNumOfSS) OF
290                              SS-Code
291
292 maxNumOfSS   INTEGER ::= 30
293
294 SS-InfoList ::= SEQUENCE SIZE (1..maxNumOfSS) OF
295                              SS-Info
296
297 ForwardingData ::= SEQUENCE {
298     forwardedToNumber             [5] ISDN-AddressString        OPTIONAL,
299     -- forwardedToNumber must be present in version 1
300     forwardedToSubaddress         [4] ISDN-SubaddressString     OPTIONAL,
301     -- forwardedToSubaddress must be absent in version 1
302     forwardingOptions             [6] ForwardingOptions         OPTIONAL,
303     ...}
304
305 END
```

14.7.5   Supplementary service codes

```
1  MAP-SS-Code {
2      ccitt identified-organization (4) etsi (0) mobileDomain (0)
3      gsm-Network (1) modules (3) map-SS-Code (15) version2 (2)}
4
5  DEFINITIONS
6
7  ::=
8
9  BEGIN
10
11 SS-Code ::= OCTET STRING (SIZE (1))
12     -- This type is used to represent the code identifying a single
13     -- supplementary service, a group of supplementary services, or
14     -- all supplementary services. The services and abbreviations
15     -- used are defined in GSM 02.04. The internal structure is
16     -- defined as follows:
17     --
18     -- bits 87654321: group (bits 8765), and specific service
19     -- (bits 4321)
20
21 allSS                           SS-Code ::= '00000000'B
22     -- reserved for possible future use
23     -- all SS
24
25 allLineIdentificationSS         SS-Code ::= '00010000'B
26     -- reserved for possible future use
27     -- all line identification SS
28 clip                            SS-Code ::= '00010001'B
29     -- calling line identification presentation
30 clir                            SS-Code ::= '00010010'B
31     -- calling line identification restriction
32 colp                            SS-Code ::= '00010011'B
33     -- connected line identification presentation
34 colr                            SS-Code ::= '00010100'B
35     -- connected line identification restriction
36 mci         SS-Code ::= '00010101'B
37     -- reserved for possible future use
38     -- malicious call identification
39
40 allForwardingSS                 SS-Code ::= '00100000'B
41     -- all forwarding SS
42 cfu         SS-Code ::= '00100001'B
43     -- call forwarding unconditional
44 allCondForwardingSS             SS-Code ::= '00101000'B
45     -- all conditional forwarding SS
46 cfb         SS-Code ::= '00101001'B
47     -- call forwarding on mobile subscriber busy
48 cfnry                           SS-Code ::= '00101010'B
49     -- call forwarding on no reply
50 cfnrc                           SS-Code ::= '00101011'B
51     -- call forwarding on mobile subscriber not reachable
52
53 allCallOfferingSS               SS-Code ::= '00110000'B
54     -- reserved for possible future use
55     -- all call offering SS includes also all forwarding SS
56 ect                             SS-Code ::= '00110001'B
57         -- explicit call transfer
58 mah         SS-Code ::= '00110010'B
59     -- reserved for possible future use
60     -- mobile access hunting
61
```

```
62  allCallCompletionSS                     SS-Code ::= '01000000'B
63      -- reserved for possible future use
64      -- all Call completion SS
65  cw                                      SS-Code ::= '01000001'B
66      -- call waiting
67  hold                                    SS-Code ::= '01000010'B
68      -- call hold
69  ccbs                                    SS-Code ::= '01000011'B
70      -- reserved for possible future use
71      -- completion of call to busy subscribers
72
73  allMultiPartySS                         SS-Code ::= '01010000'B
74      -- reserved for possible future use
75      -- all multiparty SS
76  multiPTY    SS-Code ::= '01010001'B
77      -- multiparty
78
79  allCommunityOfInterest-SS               SS-Code ::= '01100000'B
80      -- reserved for possible future use
81      -- all community of interest SS
82  cug         SS-Code ::= '01100001'B
83      -- closed user group
84
85  allChargingSS                           SS-Code ::= '01110000'B
86      -- reserved for possible future use
87      -- all charging SS
88  aoci                                    SS-Code ::= '01110001'B
89      -- advice of charge information
90  aocc                                    SS-Code ::= '01110010'B
91      -- advice of charge charging
92
93  allAdditionalInfoTransferSS             SS-Code ::= '10000000'B
94      -- reserved for possible future use
95      -- all additional information transfer SS
96  uus         SS-Code ::= '10000001'B
97      -- reserved for possible future use
98      -- UUS user-to-user signalling
99
100 allBarringSS SS-Code ::= '10010000'B
101     -- all barring SS
102 barringOfOutgoingCalls                  SS-Code ::= '10010001'B
103     -- barring of outgoing calls
104 baoc                                    SS-Code ::= '10010010'B
105     -- barring of all outgoing calls
106 boic                                    SS-Code ::= '10010011'B
107     -- barring of outgoing international calls
108 boicExHC    SS-Code ::= '10010100'B
109     -- barring of outgoing international calls except those directed
110     -- to the home PLMN
111 barringOfIncomingCalls                  SS-Code ::= '10011001'B
112     -- barring of incoming calls
113 baic                                    SS-Code ::= '10011010'B
114     -- barring of all incoming calls
115 bicRoam                                 SS-Code ::= '10011011'B
116     -- barring of incoming calls when roaming outside home PLMN
117     -- Country
118
```

```
119 allPLMN-specificSS                  SS-Code ::= '11110000'B
120 -- *plmn-specific* must not be used in version 1
121 plmn-specificSS-1                    SS-Code ::= '11110001'B
122 plmn-specificSS-2                    SS-Code ::= '11110010'B
123 plmn-specificSS-3                    SS-Code ::= '11110011'B
124 plmn-specificSS-4                    SS-Code ::= '11110100'B
125 plmn-specificSS-5                    SS-Code ::= '11110101'B
126 plmn-specificSS-6                    SS-Code ::= '11110110'B
127 plmn-specificSS-7                    SS-Code ::= '11110111'B
128 plmn-specificSS-8                    SS-Code ::= '11111000'B
129 plmn-specificSS-9                    SS-Code ::= '11111001'B
130 plmn-specificSS-A                    SS-Code ::= '11111010'B
131 plmn-specificSS-B                    SS-Code ::= '11111011'B
132 plmn-specificSS-C                    SS-Code ::= '11111100'B
133 plmn-specificSS-D                    SS-Code ::= '11111101'B
134 plmn-specificSS-E                    SS-Code ::= '11111110'B
135 plmn-specificSS-F                    SS-Code ::= '11111111'B
136
137 END
```

14.7.6 Short message data types

```
1  MAP-SM-DataTypes {
2      ccitt identified-organization (4) etsi (0) mobileDomain (0)
3      gsm-Network (1) modules (3) map-SM-DataTypes (16) version2 (2)}
4
5  DEFINITIONS
6
7  IMPLICIT TAGS
8
9  ::=
10
11 BEGIN
12
13 EXPORTS
14     RoutingInfoForSM-Arg,
15     RoutingInfoForSM-Res,
16     ForwardSM-Arg,
17     ReportSM-DeliveryStatusArg,
18     AlertServiceCentreArg,
19     InformServiceCentreArg,
20     ReadyForSM-Arg,
21     SM-DeliveryOutcome,
22     AlertReason
23 ;
24
25 IMPORTS
26     AddressString,
27     ISDN-AddressString,
28     SignalInfo,
29     IMSI,
30     LocationInfo,
31     LMSI
32 FROM MAP-CommonDataTypes {
33     ccitt identified-organization (4) etsi (0) mobileDomain (0)
34     gsm-Network (1) modules (3) map-CommonDataTypes (18) version2 (2)}
35
36     TeleserviceCode
37 FROM MAP-TS-Code {
38     ccitt identified-organization (4) etsi (0) mobileDomain (0)
39     gsm-Network (1) modules (3) map-TS-Code (19) version2 (2)}
40 ;
41
42
43 RoutingInfoForSM-Arg ::= SEQUENCE {
44     msisdn                      [0] ISDN-AddressString,
45     sm-RP-PRI                   [1] BOOLEAN,
46     serviceCentreAddress        [2] AddressString,
47     teleservice                 [5] TeleserviceCode        OPTIONAL,
48     -- teleservice must be absent in version greater 1
49     ...}
50
51 RoutingInfoForSM-Res::= SEQUENCE {
52     imsi                        IMSI,
53     locationInfoWithLMSI        [0] LocationInfoWithLMSI,
54     mwd-Set                     [2] BOOLEAN                OPTIONAL,
55     -- mwd-Set must be absent in version greater 1
56     ...}
57
58 LocationInfoWithLMSI ::= SEQUENCE {
59         locationInfo            LocationInfo,
60         lmsi                    LMSI                       OPTIONAL,
61     ...}
62
63 ForwardSM-Arg ::= SEQUENCE {
64     sm-RP-DA SM-RP-DA,
65     sm-RP-OA SM-RP-OA,
66     sm-RP-UI SignalInfo,
67     moreMessagesToSend          NULL                       OPTIONAL,
68     -- moreMessagesToSend must be absent in version 1
69     ...}
70
```

```
71  SM-RP-DA ::= CHOICE {
72      imsi                            [0] IMSI,
73      lmsi                            [1] LMSI,
74      roamingNumber                   [3] ISDN-AddressString,
75      -- roaming number must not be used in version greater 1
76      serviceCentreAddressDA          [4] AddressString,
77      noSM-RP-DA                      [5] NULL}
78      -- noSM-RP-DA must not be used in version 1
79
80  SM-RP-OA ::= CHOICE {
81      msisdn                          [2] ISDN-AddressString,
82      serviceCentreAddressOA          [4] AddressString,
83      noSM-RP-OA                      [5] NULL}
84      -- noSM-RP-OA must not be used in version 1
85


86  ReportSM-DeliveryStatusArg ::= SEQUENCE {
87      msisdn                          ISDN-AddressString,
88      serviceCentreAddress            AddressString,
89      sm-DeliveryOutcome              SM-DeliveryOutcome          OPTIONAL,
90      -- sm-DeliveryOutcome must be absent in version 1
91      -- sm-DeliveryOutcome must be present in version greater 1
92      ...}
93
94  SM-DeliveryOutcome ::= ENUMERATED {
95      memoryCapacityExceeded  (0),
96      absentSubscriber  (1),
97      successfulTransfer  (2)}
98
99  AlertServiceCentreArg ::= SEQUENCE {
100     msisdn                          ISDN-AddressString,
101     serviceCentreAddress            AddressString,
102     ...}
103
104 InformServiceCentreArg ::= SEQUENCE {
105     storedMSISDN                    ISDN-AddressString          OPTIONAL,
106     mw-Status                       MW-Status                   OPTIONAL,
107     ...}
108
109 MW-Status ::= BIT STRING {
110     sc-AddressNotIncluded  (0),
111     mnrf-Set  (1),
112     mcef-Set  (2)} (SIZE (6))
113
114 ReadyForSM-Arg ::= SEQUENCE {
115     imsi                            [0] IMSI,
116     alertReason                     AlertReason,
117     ...}
118
119 AlertReason ::= ENUMERATED {
120     ms-Present  (0),
121     memoryAvailable  (1)}
122
123 END
```

14.7.7 Error data types

```
1  MAP-ER-DataTypes {
2      ccitt identified-organization (4) etsi (0) mobileDomain (0)
3      gsm-Network (1) modules (3) map-ER-DataTypes (17) version2 (2)}
4
5  DEFINITIONS
6
7  IMPLICIT TAGS
8
9  ::=
10
11 BEGIN
12
13 EXPORTS
14      RoamingNotAllowedCause,
15      CallBarringCause,
16      CUG-RejectCause,
17      SS-IncompatibilityCause,
18      PW-RegistrationFailureCause,
19      SM-DeliveryFailureCause
20 ;
21
22 IMPORTS
23      SS-Status
24 FROM MAP-SS-DataTypes {
25      ccitt identified-organization (4) etsi (0) mobileDomain (0)
26      gsm-Network (1) modules (3) map-SS-DataTypes (14) version2 (2)}
27
28      SignalInfo,
29      BasicServiceCode
30 FROM MAP-CommonDataTypes {
31      ccitt identified-organization (4) etsi (0) mobileDomain (0)
32      gsm-Network (1) modules (3) map-CommonDataTypes (18) version2 (2)}
33
34      SS-Code
35 FROM MAP-SS-Code {
36      ccitt identified-organization (4) etsi (0) mobileDomain (0)
37      gsm-Network (1) modules (3) map-SS-Code (15) version2 (2)}
38 ;
39
40
41 RoamingNotAllowedCause ::= ENUMERATED {
42      plmnRoamingNotAllowed  (0),
43      operatorDeterminedBarring  (3)}
44
45 CallBarringCause ::= ENUMERATED {
46      barringServiceActive  (0),
47      operatorBarring  (1)}
48
49 CUG-RejectCause ::= ENUMERATED {
50      incomingCallsBarredWithinCUG  (0),
51      subscriberNotMemberOfCUG  (1),
52      requestedBasicServiceViolatesCUG-Constraints  (5),
53      calledPartySS-InteractionViolation  (7)}
54      -- calledPartySS-InteractionViolation must not be used in
55      -- version 1
56
```

```
57  SS-IncompatibilityCause ::= SEQUENCE {
58      ss-Code                          [1] SS-Code                    OPTIONAL,
59      basicService                     BasicServiceCode               OPTIONAL,
60      ss-Status                        [4] SS-Status                  OPTIONAL,
61      ...}
62
63  PW-RegistrationFailureCause ::= ENUMERATED {
64      undetermined  (0),
65      invalidFormat  (1),
66      newPasswordsMismatch  (2)}
67
68  SM-DeliveryFailureCause ::= CHOICE {
69      sm-DeliveryFailureCauseWithDiagnostic
70              SM-DeliveryFailureCauseWithDiagnostic,
71  -- sm-DeliveryFailureCauseWithDiagnostic must not be used in
72  -- version 1
73      sm-EnumeratedDeliveryFailureCause
74              SM-EnumeratedDeliveryFailureCause}
75  -- sm-EnumeratedDeliveryFailureCause must not be used in version
76  -- greater 1
77
78  SM-EnumeratedDeliveryFailureCause ::= ENUMERATED {
79      memoryCapacityExceeded  (0),
80      equipmentProtocolError  (1),
81      equipmentNotSM-Equipped  (2),
82      unknownServiceCentre  (3),
83      sc-Congestion  (4),
84      invalidSME-Address  (5),
85      subscriberNotSC-Subscriber  (6)}
86
87  SM-DeliveryFailureCauseWithDiagnostic ::= SEQUENCE {
88      sm-EnumeratedDeliveryFailureCause    SM-EnumeratedDeliveryFailureCause,
89      diagnosticInfo                       SignalInfo                    OPTIONAL,
90      ...}
91
92  END
```

14.7.8   Common data types

```
1  MAP-CommonDataTypes {
2      ccitt identified-organization (4) etsi (0) mobileDomain (0)
3      gsm-Network (1) modules (3) map-CommonDataTypes (18) version2 (2)}
4
5  DEFINITIONS
6
7  IMPLICIT TAGS
8
9  ::=
10
11 BEGIN
12
13 EXPORTS
14
15      -- general data types and values
16      AddressString,
17      ISDN-AddressString,
18      ISDN-SubaddressString,
19      ExternalSignalInfo,
20      SignalInfo,
21      maxSignalInfoLength,
22
23      -- data types for numbering and identification
24      IMSI,
25      TMSI,
26      SubscriberId,
27      IMEI,
28      LocationInfo,
29      HLR-List,
30      LMSI,
31      GlobalCellId,
32      NetworkResource,
33
34      -- data types for subscriber management
35      Category,
36      BearerServiceList,
37      TeleserviceList,
38      BasicServiceCode,
39      BasicServiceList,
40      BasicServiceGroupList,
41      maxNumOfBasicServiceGroups
42 ;
43
44 IMPORTS
45      TeleserviceCode
46 FROM MAP-TS-Code {
47      ccitt identified-organization (4) etsi (0) mobileDomain (0)
48      gsm-Network (1) modules (3) map-TS-Code (19) version2 (2)}
49
50      BearerServiceCode
51 FROM MAP-BS-Code {
52      ccitt identified-organization (4) etsi (0) mobileDomain (0)
53      gsm-Network (1) modules (3) map-BS-Code (20) version2 (2)}
54 ;
55
56
57 -- general data types
58
```

```
59  TBCD-STRING ::= OCTET STRING
60       -- This type (Telephony Binary Coded Decimal String) is used to
61       -- represent several digits from 0 through 9, *, #, a, b, c, two
62       -- digits per octet, each digit encoded 0000 to 1001 (0 to 9),
63       -- 1010 (*), 1011 (#), 1100 (a), 1101 (b) or 1110 (c); 1111 used
64       -- as filler when there is an odd number of digits.
65
66       -- bits 8765 of octet n encoding digit 2n
67       -- bits 4321 of octet n encoding digit 2(n-1) +1
68
69  AddressString ::= OCTET STRING (SIZE (1..maxAddressLength))
70       -- This type is used to represent a number for addressing
71       -- purposes. It is composed of
72       --  a)   one octet for nature of address, and numbering plan
73       --       indicator.
74       --  b)   digits of an address encoded as TBCD-String.
75
76       -- a)   The first octet includes a one bit extension indicator, a
77       --      3 bits nature of address indicator and a 4 bits numbering
78       --      plan indicator, encoded as follows:
79
80       -- bit 8: 1  (no extension)
81
82       -- bits 765: nature of address indicator
83       --   000  unknown
84       --   001  international number
85       --   010  national significant number
86       --   011  network specific number
87       --   100  subscriber number
88       --   101  reserved
89       --   110  abbreviated number
90       --   111  reserved for extension
91
92       -- bits 4321: numbering plan indicator
93       --   0000   unknown
94       --   0001   ISDN/Telephony Numbering Plan (Rec CCITT E.164)
95       --   0010   spare
96       --   0011   data numbering plan (CCITT Rec X.121)
97       --   0100   telex numbering plan (CCITT Rec F.69)
98       --   0101   spare
99       --   0110   land mobile numbering plan (CCITT Rec E.212)
100      --   0111   spare
101      --   1000   national numbering plan
102      --   1001   private numbering plan
103      --   1111   reserved for extension
104
105      --   all other values are reserved.
106
107      -- b)    The following octets representing digits of an address
108      --       encoded as a TBCD-STRING.
109
110  maxAddressLength   INTEGER ::= 20
111
112  ISDN-AddressString ::=
113          AddressString (SIZE (1..maxISDN-AddressLength))
114      -- This type is used to represent ISDN numbers.
115
116  maxISDN-AddressLength   INTEGER ::= 9
117
```

```
118  ISDN-SubaddressString ::=
119          OCTET STRING (SIZE (1..maxISDN-SubaddressLength))
120     -- This type is used to represent ISDN subaddresses.
121     -- It is composed of
122     --  a)  one octet for type of subaddress and odd/even indicator.
123     --  b)  20 octets for subaddress information.
124
125     --  a)  The first octet includes a one bit extension indicator, a
126     --      3 bits type of subaddress and a one bit odd/even indicator,
127     --      encoded as follows:
128
129     --  bit 8: 1  (no extension)
130
131     --  bits 765: type of subaddress
132     --      000  NSAP (X.213/ISO 8348 AD2)
133     --      010  User Specified
134     --      All other values are reserved
135
136     --  bit 4: odd/even indicator
137     --      0  even number of address signals
138     --      1  odd number of address signals
139     --      The odd/even indicator is used when the type of subaddress
140     --      is "user specified" and the coding is BCD.
141
142     --  bits 321: 000 (unused)
143
144     --  b) Subaddress information.
145     --  The NSAP X.213/ISO8348AD2 address shall be formatted as specified
146     --  by octet 4 which contains the Authority and Format Identifier
147     --  (AFI). The encoding is made according to the "preferred binary
148     --  encoding" as defined in X.213/ISO834AD2. For the definition
149     --  of this type of subaddress, see CCITT Rec I.334.
150
151     --  For User-specific subaddress, this field is encoded according
152     --  to the user specification, subject to a maximum length of 20
153     --  octets. When interworking with X.25 networks BCD coding should
154     --  be applied.
155
156  maxISDN-SubaddressLength  INTEGER ::= 21
157
158  ExternalSignalInfo ::= SEQUENCE {
159      protocolId                    ProtocolId,
160      signalInfo                    SignalInfo,
161      ...}
162      -- Information about the internal structure is given in
163      -- subclause 5.6.9.
164
165  SignalInfo ::= OCTET STRING (SIZE (1..maxSignalInfoLength))
166
167  maxSignalInfoLength  INTEGER ::= 200
168      -- This NamedValue represents the theoretical maximum number of
169      -- octets which are available to carry a single data type,
170      -- without requiring segmentation to cope with the network layer
171      -- service. However, the actual maximum size available for a data
172      -- type may be lower, especially when other information elements
173      -- have to be included in the same component.
174
175  ProtocolId ::= ENUMERATED {
176      gsm-0408   (1),
177      gsm-0806   (2),
178      gsm-BSSMAP  (3),
179      -- Value 3 must not be used in version 1 and
180      -- must not be used in version 2.
181      ets-300102-1   (4)}
182
```

```
183
184  -- data types for numbering and identification
185
186  IMSI ::= TBCD-STRING (SIZE (3..8))
187      -- digits of MCC, MNC, MSIN are concatenated in this order.
188
189  TMSI ::= OCTET STRING (SIZE (1..4))
190
191  SubscriberId ::= CHOICE {
192      imsi                            [0] IMSI,
193      tmsi                            [1] TMSI}
194
195  IMEI ::= TBCD-STRING (SIZE (8))
196      -- Refers to International Mobile Station Equipment Identity
197      -- and Software Version Number (SVN) defined in GSM 03.03.
198      -- If the SVN is not present the last octet shall contain the
199      -- digit 0 and a filler.
200      -- If present the SVN shall be included in the last octet.
201      -- The SVN must not be present in version 1.
202
203  LocationInfo ::= CHOICE {
204      roamingNumber                   [0] ISDN-AddressString,
205      -- roamingNumber must not be used in version greater 1
206      msc-Number                      [1] ISDN-AddressString}
207
208  HLR-Id ::= IMSI
209      -- leading digits of IMSI, i.e. (MCC, MNC, leading digits of
210      -- MSIN) forming HLR Id defined in GSM 03.03.
211
212  HLR-List ::= SEQUENCE SIZE (1..maxNumOfHLR-Id) OF
213                                  HLR-Id
214
215  maxNumOfHLR-Id   INTEGER ::= 50
216
217  LMSI ::= OCTET STRING (SIZE (4))
218
219  GlobalCellId ::= OCTET STRING (SIZE (5..7))
220      -- Refers to Cell Global Identification defined in GSM 03.03.
221      -- Octets are coded according to GSM 04.08.
222      -- The internal structure is defined as follows:
223      -- Mobile Country Code:        3 digits according to CCITT Rec E.212
224      --                             1 digit filler (1111)
225      -- Mobile Network Code:        2 digits according to CCITT Rec E.212
226      -- Location Area Code:         2 octets according to GSM 04.08
227      -- Cell Identity:              2 octets (CI) according to GSM 04.08
228
229  NetworkResource ::= ENUMERATED {
230      plmn  (0),
231      hlr  (1),
232      vlr  (2),
233      pvlr  (3),
234      controllingMSC  (4),
235      vmsc  (5),
236      eir  (6),
237      rss  (7)}
238
239
```

```
240  -- data types for subscriber management
241
242  Category ::= OCTET STRING (SIZE (1))
243      -- The internal structure is defined in CCITT Rec Q.763.
244
245  BearerServiceList ::= SEQUENCE SIZE (1..maxNumOfBearerServices) OF
246                                   BearerServiceCode
247
248  maxNumOfBearerServices   INTEGER ::= 50
249
250  TeleserviceList ::= SEQUENCE SIZE (1..maxNumOfTeleservices) OF
251                                   TeleserviceCode
252
253  maxNumOfTeleservices    INTEGER ::= 20
254
255  BasicServiceCode ::= CHOICE {
256      bearerService              [2] BearerServiceCode,
257      teleservice                [3] TeleserviceCode}
258
259  BasicServiceList ::= SEQUENCE SIZE (1..maxNumOfBasicServices) OF
260                                   BasicServiceCode
261
262  maxNumOfBasicServices   INTEGER ::= 70
263
264  BasicServiceGroupList ::= SEQUENCE SIZE (1..maxNumOfBasicServiceGroups) OF
265                                   BasicServiceCode
266
267  maxNumOfBasicServiceGroups   INTEGER ::= 13
268
269  END
```

14.7.9   Teleservice Codes

```
1  MAP-TS-Code {
2      ccitt identified-organization (4) etsi (0) mobileDomain (0)
3      gsm-Network (1) modules (3) map-TS-Code (19) version2 (2)}
4
5  DEFINITIONS
6
7  ::=
8
9  BEGIN
10
11 TeleserviceCode ::= OCTET STRING (SIZE (1))
12     -- This type is used to represent the code identifying a single
13     -- teleservice, a group of teleservices, or all teleservices. The
14     -- services are defined in GSM 02.03.
15     -- The internal structure is defined as follows:
16
17     -- bits 87654321: group (bits 8765) and specific service
18     -- (bits 4321)
19
20 allTeleservices                        TeleserviceCode ::= '00000000'B
21
22 allSpeechTransmissionServices          TeleserviceCode ::= '00010000'B
23 telephony                              TeleserviceCode ::= '00010001'B
24 emergencyCalls                         TeleserviceCode ::= '00010010'B
25
26 allShortMessageServices                TeleserviceCode ::= '00100000'B
27 shortMessageMT-PP                      TeleserviceCode ::= '00100001'B
28 shortMessageMO-PP                      TeleserviceCode ::= '00100010'B
29
30 allFacsimileTransmissionServices       TeleserviceCode ::= '01100000'B
31 facsimileGroup3AndAlterSpeech          TeleserviceCode ::= '01100001'B
32 automaticFacsimileGroup3               TeleserviceCode ::= '01100010'B
33 facsimileGroup4                        TeleserviceCode ::= '01100011'B
34 -- facsimileGroup4 must not be used in version 1
35
36 -- The following non-hierarchical Compound Teleservice Groups
37 -- are defined in GSM 02.30:
38 allDataTeleservices                    TeleserviceCode ::= '01110000'B
39     -- covers Teleservice Groups 'allFacsimileTransmissionServices'
40     -- and 'allShortMessageServices'
41 allTeleservices-ExeptSMS               TeleserviceCode ::= '10000000'B
42     -- covers Teleservice Groups 'allSpeechTransmissionServices' and
43     -- 'allFacsimileTransmissionServices'
44 --
45 -- Compound Teleservice Group Codes are only used in call
46 -- independent supplementary service operations, i.e. they
47 -- are not used in InsertSubscriberData or in
48 -- DeleteSubscriberData messages. Compound Teleservice Group
49 -- Codes must not be used in version 1.
50
```

```
51  allPLMN-specificTS                      TeleserviceCode ::= '11010000'B
52  -- *plmn-specific* must not be used in version 1
53  plmn-specificTS-1                       TeleserviceCode ::= '11010001'B
54  plmn-specificTS-2                       TeleserviceCode ::= '11010010'B
55  plmn-specificTS-3                       TeleserviceCode ::= '11010011'B
56  plmn-specificTS-4                       TeleserviceCode ::= '11010100'B
57  plmn-specificTS-5                       TeleserviceCode ::= '11010101'B
58  plmn-specificTS-6                       TeleserviceCode ::= '11010110'B
59  plmn-specificTS-7                       TeleserviceCode ::= '11010111'B
60  plmn-specificTS-8                       TeleserviceCode ::= '11011000'B
61  plmn-specificTS-9                       TeleserviceCode ::= '11011001'B
62  plmn-specificTS-A                       TeleserviceCode ::= '11011010'B
63  plmn-specificTS-B                       TeleserviceCode ::= '11011011'B
64  plmn-specificTS-C                       TeleserviceCode ::= '11011100'B
65  plmn-specificTS-D                       TeleserviceCode ::= '11011101'B
66  plmn-specificTS-E                       TeleserviceCode ::= '11011110'B
67  plmn-specificTS-F                       TeleserviceCode ::= '11011111'B
68
69  END
```

14.7.10   Bearer Service Codes

```
 1  MAP-BS-Code {
 2      ccitt identified-organization (4) etsi (0) mobileDomain (0)
 3      gsm-Network (1) modules (3) map-BS-Code (20) version2 (2)}
 4
 5  DEFINITIONS
 6
 7  ::=
 8
 9  BEGIN
10
11  BearerServiceCode ::= OCTET STRING (SIZE (1))
12      -- This type is used to represent the code identifying a single
13      -- bearer service, a group of bearer services, or all bearer
14      -- services. The services are defined in GSM 02.02.
15      -- The internal structure is defined as follows:
16      --
17      -- v1: bit 8: Transparency indicator (If applicable)
18      --  0    transparent (postfix T) or if not applicable
19      --       (no postfix T or NT)
20      --  1    non transparent (postfix NT)
21      --
22      -- v2: bit 8: 0 (unused)
23      -- plmn-specific bearer services:
24      -- bits 87654321: defined by the HPLMN operator
25      -- rest of bearer services:
26      --
27      -- bits 7654321: group (bits 7654), and rate, if applicable
28      -- (bits 321)
29      -- symbolic constants are assigned only for v2
30
31  allBearerServices                   BearerServiceCode ::= '00000000'B
32
33  allDataCDA-Services                 BearerServiceCode ::= '00010000'B
34  dataCDA-300bps                      BearerServiceCode ::= '00010001'B
35  dataCDA-1200bps                     BearerServiceCode ::= '00010010'B
36  dataCDA-1200-75bps                  BearerServiceCode ::= '00010011'B
37  dataCDA-2400bps                     BearerServiceCode ::= '00010100'B
38  dataCDA-4800bps                     BearerServiceCode ::= '00010101'B
39  dataCDA-9600bps                     BearerServiceCode ::= '00010110'B
40
41  allDataCDS-Services                 BearerServiceCode ::= '00011000'B
42  dataCDS-1200bps                     BearerServiceCode ::= '00011010'B
43  dataCDS-2400bps                     BearerServiceCode ::= '00011100'B
44  dataCDS-4800bps                     BearerServiceCode ::= '00011101'B
45  dataCDS-9600bps                     BearerServiceCode ::= '00011110'B
46
47  allPadAccessCA-Services             BearerServiceCode ::= '00100000'B
48  padAccessCA-300bps                  BearerServiceCode ::= '00100001'B
49  padAccessCA-1200bps                 BearerServiceCode ::= '00100010'B
50  padAccessCA-1200-75bps              BearerServiceCode ::= '00100011'B
51  padAccessCA-2400bps                 BearerServiceCode ::= '00100100'B
52  padAccessCA-4800bps                 BearerServiceCode ::= '00100101'B
53  padAccessCA-9600bps                 BearerServiceCode ::= '00100110'B
54
```

```
55  allDataPDS-Services                  BearerServiceCode ::= '00101000'B
56  dataPDS-2400bps                      BearerServiceCode ::= '00101100'B
57  dataPDS-4800bps                      BearerServiceCode ::= '00101101'B
58  dataPDS-9600bps                      BearerServiceCode ::= '00101110'B
59
60  allAlternateSpeech-DataCDA           BearerServiceCode ::= '00110000'B
61
62  allAlternateSpeech-DataCDS           BearerServiceCode ::= '00111000'B
63
64  allSpeechFollowedByDataCDA           BearerServiceCode ::= '01000000'B
65
66  allSpeechFollowedByDataCDS           BearerServiceCode ::= '01001000'B
67
68  -- The following non-hierarchical Compound Bearer Service
69  -- Groups are defined in GSM 02.30:
70  allDataCircuitAsynchronous           BearerServiceCode ::= '01010000'B
71      -- covers "allDataCDA-Services", "allAlternateSpeech-DataCDA" and
72      -- "allSpeechFollowedByDataCDA"
73  allAsynchronousServices              BearerServiceCode ::= '01100000'B
74      -- covers "allDataCDA-Services", "allAlternateSpeech-DataCDA",
75      -- "allSpeechFollowedByDataCDA" and "allPadAccessCDA-Services"
76  allDataCircuitSynchronous            BearerServiceCode ::= '01011000'B
77      -- covers "allDataCDS-Services", "allAlternateSpeech-DataCDS" and
78      -- "allSpeechFollowedByDataCDS"
79  allSynchronousServices               BearerServiceCode ::= '01101000'B
80      -- covers "allDataCDS-Services", "allAlternateSpeech-DataCDS",
81      -- "allSpeechFollowedByDataCDS" and "allDataPDS-Services"
82  --
83  -- Compound Bearer Service Group Codes are only used in call
84  -- independent supplementary service operations, i.e. they
85  -- are not used in InsertSubscriberData or in
86  -- DeleteSubscriberData messages.
87  --   Compound Bearer Service Group Codes must not be used in
88  -- version 1.
89
90  allPLMN-specificBS                   BearerServiceCode ::= '11010000'B
91  -- *plmn-specific* must not be used in version 1
92  plmn-specificBS-1                    BearerServiceCode ::= '11010001'B
93  plmn-specificBS-2                    BearerServiceCode ::= '11010010'B
94  plmn-specificBS-3                    BearerServiceCode ::= '11010011'B
95  plmn-specificBS-4                    BearerServiceCode ::= '11010100'B
96  plmn-specificBS-5                    BearerServiceCode ::= '11010101'B
97  plmn-specificBS-6                    BearerServiceCode ::= '11010110'B
98  plmn-specificBS-7                    BearerServiceCode ::= '11010111'B
99  plmn-specificBS-8                    BearerServiceCode ::= '11011000'B
100 plmn-specificBS-9                    BearerServiceCode ::= '11011001'B
101 plmn-specificBS-A                    BearerServiceCode ::= '11011010'B
102 plmn-specificBS-B                    BearerServiceCode ::= '11011011'B
103 plmn-specificBS-C                    BearerServiceCode ::= '11011100'B
104 plmn-specificBS-D                    BearerServiceCode ::= '11011101'B
105 plmn-specificBS-E                    BearerServiceCode ::= '11011110'B
106 plmn-specificBS-F                    BearerServiceCode ::= '11011111'B
107
108 END
```

15 General on MAP user procedures

15.1 Introduction

Clauses 15 to 21 describe the use of MAP services for GSM signalling procedures. GSM signalling procedures may involve one or several interfaces running one or several application protocols. This ETS addresses only the signalling procedures which require at least the use of one MAP service.

When a signalling procedure takes place in the network, an application process invocation is created in each system component involved. Part of the application process invocation acts as a MAP user and handles one or several MAP dialogues. For each dialogue it employs an instance of the MAP service provider. It may also use other communication services to exchange information on other interfaces, but detailed description of these aspects is outside the scope of this ETS.

15.2 Common aspects of user procedure descriptions

15.2.1 General conventions

For each signalling procedure this ETS provides a brief textual overview accompanied by a flow diagram which represent the functional interactions between system components. Functional interactions are labelled using the MAP service name when the interaction results from a service request or by this service name followed by the symbol "ack" when this interaction results from a service response.

For each of the system components involved, this ETS also provides a detailed textual description of the application process behaviour as well as an SDL diagram. SDL diagrams describe the sequence of events, as seen by the MAP-User, which occurs at MAP service provider boundaries as well as external events which occur at other interfaces and which impact on the previous sequence.

External events do not necessarily correspond to the messages of other protocols used in the system component. The MAP-user procedures are described as if a set of interworking functions (IWF) between the MAP-user and the other protocol entities was implemented (see figure 15.2/1). Such interworking functions are assumed to perform either an identity mapping or some processing or translation as required to eliminate information irrelevant to the MAP-user.

The mapping of service primitives on to protocol elements is described in clauses 11 to 14.

GSM signalling procedures are built from one or more sub-procedures (e.g. authentication, ciphering, ...). Sub-procedures from which signalling procedures are built are represented using SDL MACRO descriptions.

In case of any discrepancy between the textual descriptions and the SDL descriptions, the latter take precedence.

15.2.2 Naming conventions

Events related to MAP are represented by MAP service primitives. The signal names used in the SDL diagrams are derived from the service primitive names defined in clauses 5 to 10, with some lexical transformations for readability and parsability purposes (blanks between words are replaced by underscores, the first letter of each word is capitalized).

Events received and sent on other interfaces are named by appending the message or signal name to a symbol representing the interface type, with some lexical transformations for readability and parsability purposes (blanks between words are replaced by underscores, the first letter of each word is capitalized).

The following symbols are used to represent the interface types:

"I":   For interfaces to the fixed network. "I" stands for ISUP interface.
"A":   For interfaces to BSS (i.e. A-interfaces);
"OM":   For network management interfaces (communication with OMC, MML interface, ...);
"SC":   For interfaces to a Service Centre;
"HO_CA":   For internal interfaces to the Handover Control Application.
"US":   For a local USSD application.

These naming conventions can be summarized by the following BNF description:

<Event_Name>   ::= <MAP_Primitive> | <External_Event>

<MAP_Primitive>   ::= <MAP_Open> | <MAP_Close> | <MAP_U_Abort> | <MAP_P_Abort> |
                      <MAP_Specific> | <MAP_Notice>

<MAP_Open>   ::= MAP_Open_Req | MAP_Open_Ind | MAP_Open_Rsp | MAP_Open_Cnf

<MAP_Close>   ::= MAP_Close_Req | MAP_Close_Ind

<MAP_U_Abort>   ::= MAP_U_Abort_Req | MAP_U_Abort_Ind

<MAP_P_Abort>   ::= MAP_P_Abort_Ind

<MAP_Notice>   ::= MAP_Notice_Ind

<MAP_Specific>   ::= <MAP_Req> | <MAP_Ind> | <MAP_Rsp> | <MAP_Cnf>

<MAP_Req>   ::= MAP_<Service_Name>_Req

<MAP_Ind>   ::= MAP_<Service_Name>_Ind

<MAP_Rsp>   ::= MAP_<Service_Name>_Rsp

<MAP_Cnf>   ::= MAP_<Service_Name>_Cnf

<External_Event>   ::= <Interface_Type>_<External_Signal>

<Interface_Type>   ::= I | A | OM | SC | HO AC | US

<External_Signal>   ::= <Lexical_Unit>

<Service_Name>   ::= <Lexical_Unit>

<Lexical_Unit>   ::= <Lexical_Component> | <Lexical_Unit>_ <Lexical_Component>

<Lexical_Component>   ::= <Upper_Case_Letter><Letter_Or_Digit_List>

<Letter_Or_Digit_List>   ::= <Letter_Or_Digit> | <Letter_Or_Digit_List><Letter_Or_Digit>

<Letter_Or_Digit>   ::= <Letter> | <Digit>

<Letter>   ::= <Lower_Case_Letter> | <Upper_Case_Letter>

<Upper_Case_Letter>   ::= A|B|C|D|E|F|G|H|I|J|K|L|M|N|O|P|Q|R|S|T|U|V|W|X|Y|Z

<Lower_Case_Letter>   ::= a|b|c|d|e|f|g|h|i|j|k|l|m|n|o|p|q|r|s|t|u|v|w|x|y|z

<Digit>   ::= 1|2|3|4|5|6|7|8|9|0

```
                   O&M (GSM 12.xy-series)        SC (GSM 03.40)

              ┌─────────┐                   ┌─────────┐
              │   IWF   │                   │   IWF   │
              └─────────┘                   └─────────┘
                 └─OM_<xx>        ┌─SC_<xx>─┘
                           ┌──────────────────┐
    ──A──┌─────────┐       │                  │       ┌─────────┐─ISUP
  GSM    │   IWF   │─A_<xx>─│    MAP-User      │─I_<xx>─│   IWF   │ (CCITT
  08.08  └─────────┘       │                  │       └─────────┘  Q.763)
  04.08                    └──────────────────┘
  ┌─────────┐                     │      └─US_<xx>─┐  ┌─────────┐
  │Handover │                     │                 │  │  USSD   │
  │Control  │─HO_CA_<xx>─┐  ┌──────────────┐        └──│Application│
  │Applica  │            └──│ MAP-Provider │           └─────────┘
  │  tion   │               └──────────────┘
  └─────────┘
```
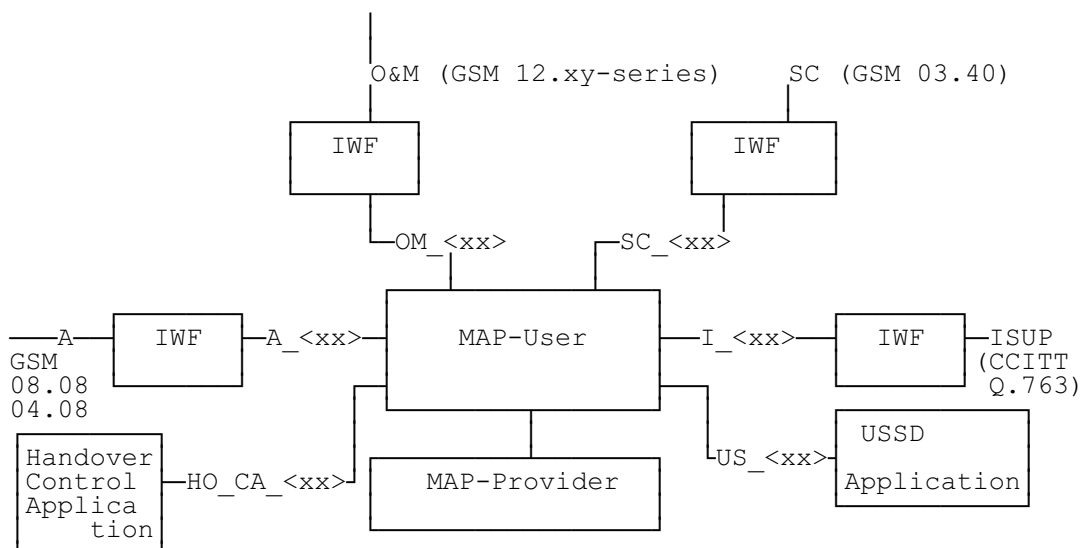
Figure 15.2/1: Interfaces applicable to the MAP-User

15.2.3   Convention on primitives parameters

15.2.3.1   Open service

When the originating and destination reference parameters shall be included in the MAP-OPEN request primitive, their value are indicated as a comment to the signal which represents this primitive.

15.2.3.2   Close service

When a pre-arranged released is requested, a comment is attached to the signal which represents the MAP-CLOSE request primitive. In the absence of comment, a normal release is assumed.

15.2.4   Version handling at dialogue establishment

Unless explicitly indicated in subsequent subclauses, the following principles regarding version handling procedures at dialogue establishment are applied by the MAP-user:

15.2.4.1   Behaviour at the initiating side

When a MAP user signalling procedure has to be executed, the MAP-user issues a MAP-OPEN request primitive with an appropriate application-context-name. If several names are supported (i.e. several versions) a suitable one is selected using the procedures described in clause 3.

If a MAP-CLOSE Confirm primitive to a MAP-OPEN request with a result parameter set to "refused" and a diagnostic parameter indicating "application-context-not-supported" or "potential incompatibility problem", the MAP-User issues a new MAP-OPEN request primitive with the equivalent version one context. This is informally represented in the SDL diagrams by a task symbol indicating "Perform V1 procedure".

15.2.4.2   Behaviour at the responding side

On receipt of a MAP-OPEN indication primitive, the MAP-User analyses the application-context-name.

If it refers to a version one context, the associated V1 procedure is executed, otherwise the associated V2 procedure is executed.

15.2.5 Abort Handling

Unless explicitly indicated in subsequent subclauses, the following principles are applied by the MAP-user regarding abort handling procedures:

On receipt of a MAP-P-ABORT indication or MAP-U-ABORT Indication primitive from any MAP-provider invocation, the MAP-User issues a MAP-U-ABORT Request primitive to each MAP-provider invocation associated with the same user procedure.

If applicable a decision is made to decide if the affected user procedure has to be retried or not.

15.2.6 SDL conventions

The MAP SDLs make use of a number of SDL concepts and conventions, where not all of them may be widely known. Therefore, this subclause outlines the use of a few concepts and conventions to improve understanding of the MAP SDLs.

The MAP User SDLs make use of SDL Processes, Procedures and Macros. Processes are independent from each other even if one process starts another one: The actions of both of them have no ordering in time. SDL Procedures and Macros are just used to ease writing of the specification: They contain parts of a behaviour used in several places, and the corresponding Procedure/Macro definition has to be expanded at the position of the Procedure/Macro call.

All Processes are started at system initialization and live forever, unless process creation/termination is indicated explicitly (i.e. a process is created by some other process).

The direction of Input/Output Signals in the SDL graphs is used to indicate the entity to which/from which communication is directed. If a process A communicates in parallel with processes B and C, all Inputs/Outputs to/from B are directed to one side, whereas communication with C is directed to the other side. However, there has been no formal convention used that communication to a certain entity (e.g. a HLR) will always be directed to a certain side (e.g. right).

In each state all those Input Signals are listed, which result in an action and/or state change. If an Input Signal is not listed in a state, receipt of this input should lead to an implicit consumption without any action or state change (according to the SDL rules). This implicit consumption is mainly used for receipt of the MAP DELIMITER indication and for receipt of a MAP CLOSE indication, except for a premature MAP CLOSE.

15.3 Interaction between MAP Provider and MAP Users

Each MAP User is defined by at least one SDL process. On the dialogue initiating side the MAP User will create a new instance of a MAP Provider implicit by issuing a MAP-OPEN request. This instance corresponds to a TC Dialogue and lives as long as the dialogue exists (see also subclause 11.3). There is a fix relation between MAP User and this Provider instance, i.e. all MAP service primitives from the MAP User for this dialogue are sent to this instance and all TC components received by this MAP Provider are mapped onto service primitives sent to this MAP User.

On the receiving side a MAP Provider instance is created implicit by receipt of a TC BEGIN indication. The corresponding MAP User is determined by the Application Context name included in this primitive, i.e. each Application Context is associated with one and only one MAP User. An instance of this User will be created implicit by receiving a MAP-OPEN indication. Note that in some cases there exist several SDL Processes for one MAP User (Application Context), e.g. the processes Register_SS_HLR, Erase_SS_HLR, Activate_SS_HLR, Deactivate_SS_HLR, Interrogate_SS_HLR, and Register_Password for the AC Network_Functional_SS_Handling. In these cases, a coordinator process is introduced acting as a MAP User, which in turn starts a sub-process depending on the first MAP service primitive received.

16 Mobility procedures

16.1 Location management Procedures

This subclause comprises a number of processes to handle the mobile nature of the subscriber. The processes will be addressed by SCCP Sub-System Number (MSC, VLR or HLR) and the Application Context. The following processes are defined in this subclause:

Process Update Location Area:
      Initiator: Update_Location_Area_MSC, subclause 16.1.1.2;
      Responder: Update_Location_Area_VLR, subclause 16.1.1.3;

Process Update Location:
      Initiator: Update_Location_Area_VLR, subclause 16.1.1.3, or
                      Update_Location_VLR, subclause 16.1.1.6;
      Responder: Update_Location_HLR, subclause 16.1.1.4;

Process Send Identification:
      Initiator: Update_Location_Area_VLR, subclause 16.1.1.3;
      Responder: Send_Identification_VLR, subclause 16.1.1.5;

Process Subscriber Present HLR:
      Initiator: Subscriber_Present_HLR, subclause 16.1.1.7;
      Responder: Short_Message_Alert_IWMSC, subclause 20.4.3;

Process Cancel Location:
      Initiator: Cancel_Location_HLR, subclause 16.1.2.2;
      Responder: Cancel_Location_VLR, subclause 16.1.2.3;

Process Detach IMSI:
      Initiator: Detach_IMSI_MSC, subclause 16.3.2;
      Responder: Detach_IMSI_VLR, subclause 16.3.3.

As both the Update Location Area and the Detach IMSI processes use the same application context name, the MAP Provider cannot distinguish between them. Therefore, a Location Management Coordinator Process will act as one user for this application context. This process (one in MSC, one in VLR) will create the Update Location Area or the Detach IMSI process, depending on the first service primitive received in the respective dialogue.

Additionally, a Location Management Coordinator process in the HLR coordinates the two application processes "Update Location HLR" (subclause 16.1.1.4) and "RESTORE_DATA_HLR" (subclause 16.3.3) that are addressed by the same application context.

Location Management Coordinator MSC

On receipt of a request for location updating from the A-interface, the Location Management Coordinator in the MSC will:

-      create the process Update_Location_Area_MSC in case the updating type indicated in the A-interface primitive indicates normal updating, periodic updating or IMSI Attach;

-      create the process Detach_IMSI_MSC in case the updating type indicated in the A-interface primitive indicates IMSI Detach.

The respective primitive is then forwarded to the created process. Henceforth, the coordinator will relay all service primitives from provider to the user and vice versa, until a request or indication for dialogue termination is received. This last primitive will be relayed, too, before the Coordinator process returns to idle state.

Location Management Coordinator VLR

On receipt of a dialogue request for the Location Management Application Context (see Receive_Open_Ind macro in subclause 21.1), the Location_Management_Coordinator will:

- terminate the procedure in case of parameter problems;

- revert to the MAP version one procedure in case the MSC indicated version one protocol; or

- continue as below, if the dialogue is accepted.

Depending on the first service primitive received from the MAP Provider in this dialogue, the user process is created:

- Update_Location_Area_VLR in case the primitive is a MAP_UPDATE_LOCATION_AREA indication;

- Detach_IMSI_VLR in case the primitive is a MAP_DETACH IMSI indication.

In case a MAP_U_ABORT, MAP_P_ABORT or a premature MAP_CLOSE indication is received instead, the process returns to idle state. If a MAP_NOTICE indication is received, the dialogue towards the MSC is aborted and the process returns to idle state.

After creation of the user process the service primitive received from the provider is passed to the user process. Henceforth, the coordinator will relay all service primitives from provider to the user and vice versa, until a request or indication for dialogue termination is received. This last primitive will be relayed, too, before the Coordinator process returns to idle state.

Location Management Coordinator HLR

On receipt of a dialogue request for the Location Management Application Context (see Receive_Open_Ind macro in subclause 21.1), the Location_Management_Coordinator will:

- terminate the process in case of parameter problems; or

- revert to MAP version 1 protocol if the VLR requests version 1 protocol; or

- continue as described in the following, if the dialogue is accepted.

The user process is created depending on the first service primitive received from the MAP service provider within this dialogue:

- Update_Location_HLR if the primitive is a MAP_UPDATE_LOCATION indication;

- RESTORE_DATA_HLR if the primitive is a MAP_RESTORE_DATA indication.

If a MAP_NOTICE indication is received instead, the dialogue towards the MSC is terminated and the process returns to idle state.

After creation of the user process the service primitive received from the MAP service-provider is passed to the user process. Henceforth, the coordinator will relay all service primitives from MAP service-provider to the MAP service-user and vice versa, until a request or indication for dialogue termination is received. This last primitive will be relayed, too, before the Coordinator process returns to idle state.

Figure 16.1/1: Process Location_Management_Coordinator_MSC

NULL

A_LU_ _Request . . . . Normal Updating, Periodic Updating, IMSI attach

A_LU_ _Request . . . . IMSI detach

Update_ _Location_ _Area_MSC

Detach_IMSI_ _MSC

A_LU_Request_ _to_OFFSPRING

RELAY_INFO

* from provider

MAP_P_ABORT_Ind, MAP_U_ABORT_Ind, MAP_CLOSE_Ind

* from OFFSPRING

MAP_U_ABORT_Req, MAP_CLOSE_Req . . . . from OFFSPRING

- to OFFSPRING

- to OFFSPRING

- to Provider

- to Provider

RELAY_INFO

NULL

RELAY_INFO

NULL
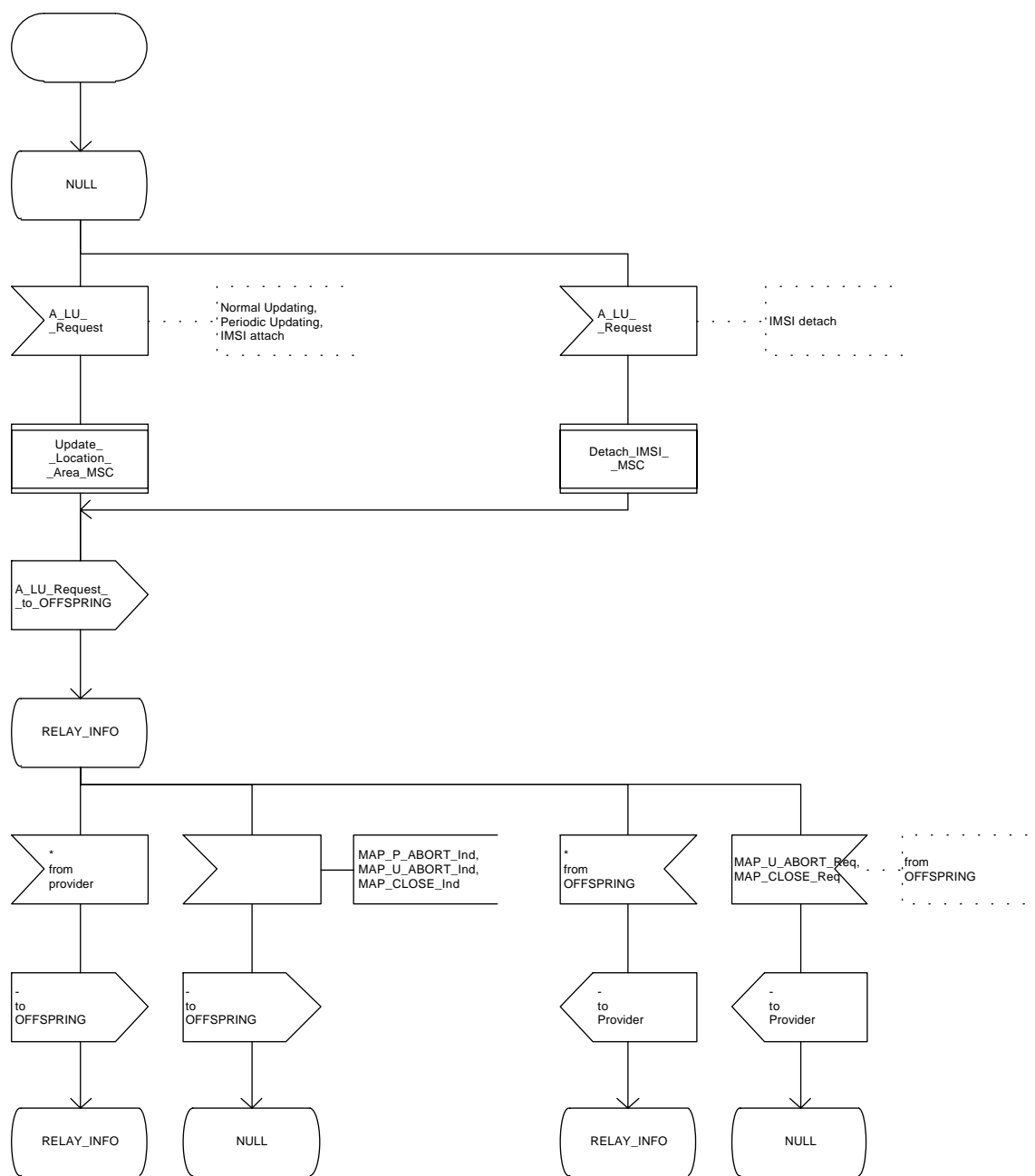
Figure 16.1/1

Figure 16.1/2: Process Location_Management_Coordinator_VLR
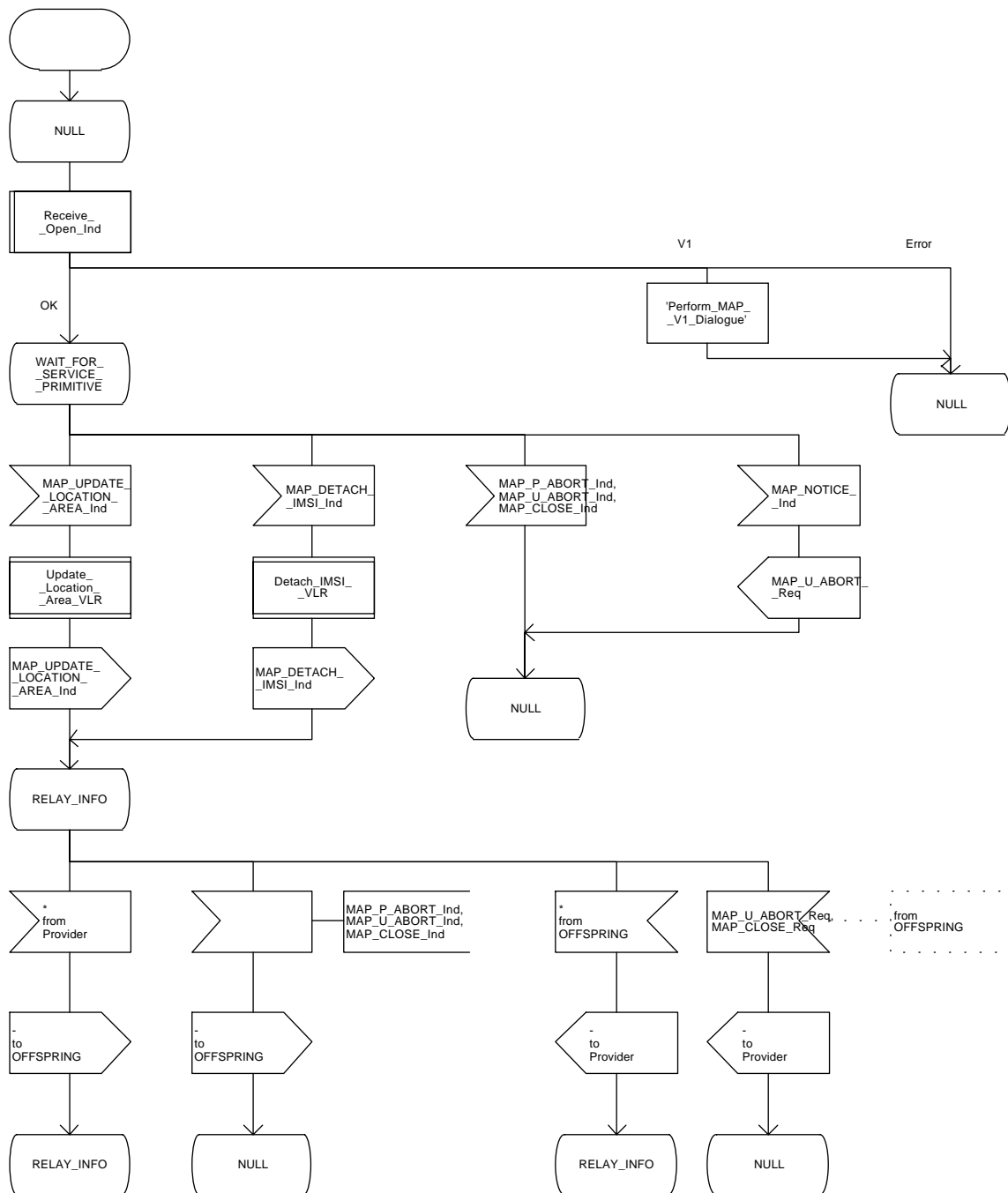
Figure 16.1/3: Process Location_Management_Coordinator_HLR
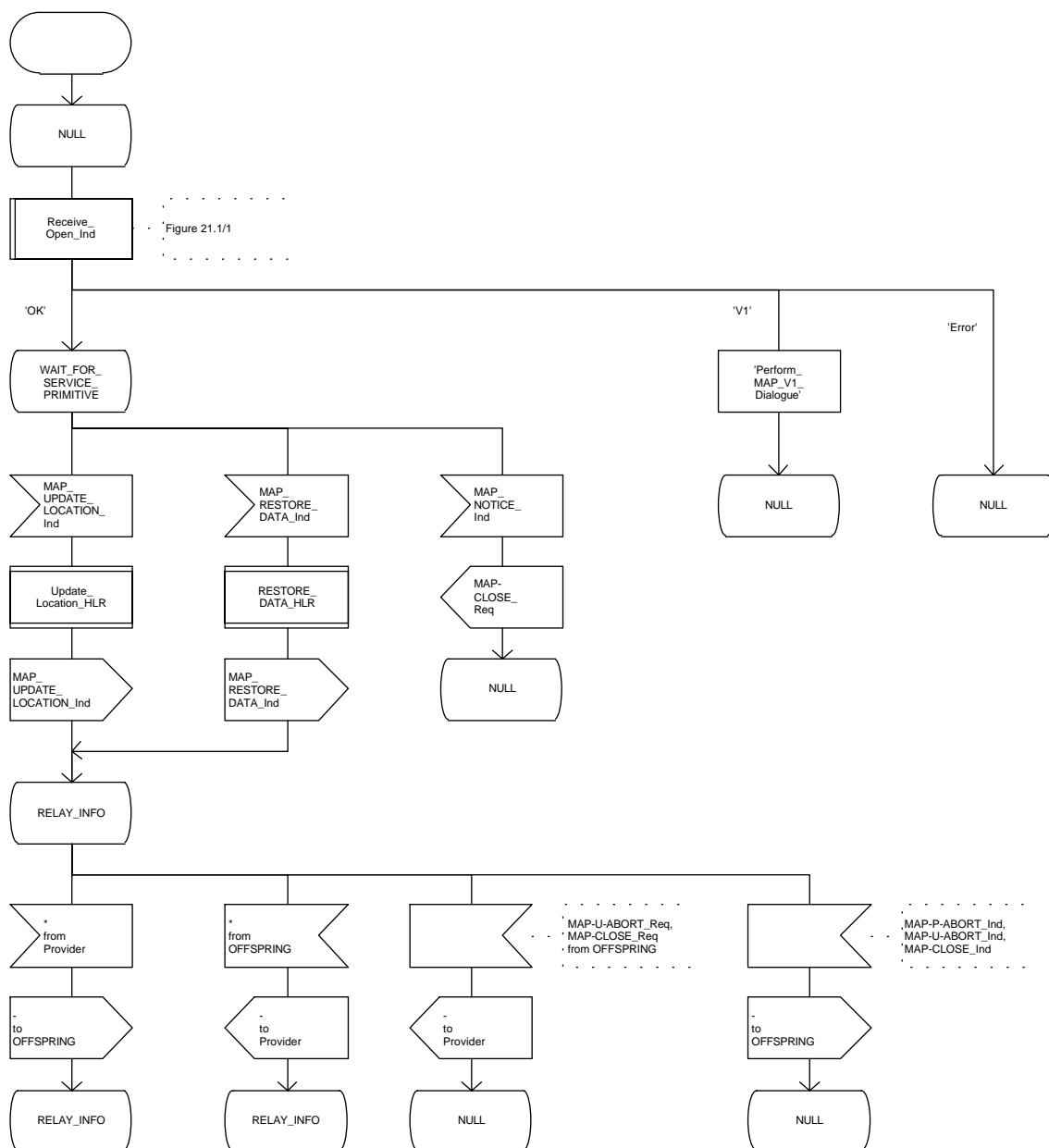
16.1.1 Location updating

16.1.1.1 General

The location updating procedure is used to update the location information held in the network. This location information is used to route incoming calls, short messages and unstructured supplementary service data to the roaming subscriber. Additionally, this procedure is used to provide the VLR with the information that a subscriber already registered, but being detached, is reachable again (IMSI Attach, see GSM 03.12). The use of this Detach / Attach feature is optional for the network operator.

To minimize the updates of the subscriber's HLR, the HLR holds only information about the VLR and MSC the subscriber is attached to. The VLR contains more detailed location information, i.e. the location area the subscriber is actually roaming in. Therefore, the VLR needs to be updated at each location area change (see figure 16.1.1/1 for this procedure), whereas the HLR needs updating only in the following cases:

- when the subscriber registers in a new VLR, i.e. the VLR has no data for that subscriber;

- when the subscriber registers in a new location area of the same VLR and new routing information is to be provided to the HLR (change of MSC area);

- if the indicator "Confirmed by HLR" or the indicator "Location Information Confirmed in HLR" is set to "Not Confirmed" because of HLR or VLR restoration, and the VLR receives an indication that the subscriber is present.

If a mobile subscriber registers in a visitor location register (VLR) not holding any information about this subscriber and is identified by a temporary mobile subscriber identity (TMSI) allocated by a previous visitor location register (PVLR), if the PVLR identity can be derived from LAI the new VLR must obtain the IMSI from PVLR to identify the HLR to be updated (see figure 16.1.1/2). If the IMSI cannot be retrieved from PVLR, it is requested from the MS (see figure 16.1.1/3).

The following MAP services are invoked by the location update procedure:

```
MAP_UPDATE_LOCATION_AREA      (see subclause 6.1);
MAP_UPDATE_LOCATION           (see subclause 6.1);
MAP_CANCEL_LOCATION           (see subclause 6.1);
MAP_INSERT_SUBSCRIBER_DATA    (see subclause 6.8);
MAP_SEND_IDENTIFICATION       (see subclause 6.1);
MAP_PROVIDE_IMSI         (see subclause 6.9);
MAP_AUTHENTICATE         (see subclause 6.5);
MAP_SET_CIPHERING_MODE (see subclause 6.6);
MAP_FORWARD_NEW_TMSI          (see subclause 6.9);
MAP_CHECK_IMEI           (see subclause 6.7);
MAP_ACTIVATE_TRACE_MODE       (see subclause 7.2);
MAP_TRACE_SUBSCRIBER_ACTIVITY      (see subclause 7.2).
```

```
  MS           BS      A      MSC           B          VLR
┌────┐      ┌────┐     │    ┌─────┐         │        ┌─────┐
│ MS │──────│ BS │─────┼────│ MSC │─────────┼────────│ VLR │
└────┘      └────┘     │    └─────┘         │        └─────┘
  │            │       │       │            │            │
  │  A_LU_REQUEST      │       │            │            │
  │------------------------------->│  MAP_UPDATE_         │
  │            │       │       │─────────────────────────>
  │            │       │       │   LOCATION_AREA         │
  │          (note 1)  │       │                         │
  │            │       │       │   MAP_AUTHENTICATE      │
  │<-------------------------------│<─────────────────────
  │            │       │       │   MAP_AUTHENTICATE ack  │
  │------------------------------->│─────────────────────>
  │            │       │       │        (note 2)         │
  │            │       │       │  MAP_SET_CIPHERING_MODE │
  │<-------------------------------│<─────────────────────
  │            │       │       │                         │
  │            │       │       │  MAP_TRACE_SUBSCRIBER_  │
  │            │       │       │         ACTIVITY        │
  │            │       │       │<─────────────────────────
  │            │       │       │                         │
  │            │       │       │     MAP_CHECK_IMEI      │
  │<-------------------------------│<─────────────────────
  │            │       │       │   MAP_CHECK_IMEI ack    │
  │------------------------------->│─────────────────────>
  │            │       │       │                         │
  │            │       │       │  MAP_FORWARD_NEW_TMSI   │
  │<-------------------------------│<─────────────────────
  │            │       │       │                         │
  │  A_LU_CONFIRM      │       │   MAP_UPDATE_LOCATION_  │
  │<-------------------------------│<─────────────────────
  │            │       │       │       AREA ack          │
  │            │       │       │  MAP_FORW._NEW_TMSI ack │
  │------------------------------->│─────────────────────>
  │            │       │       │                         │
```
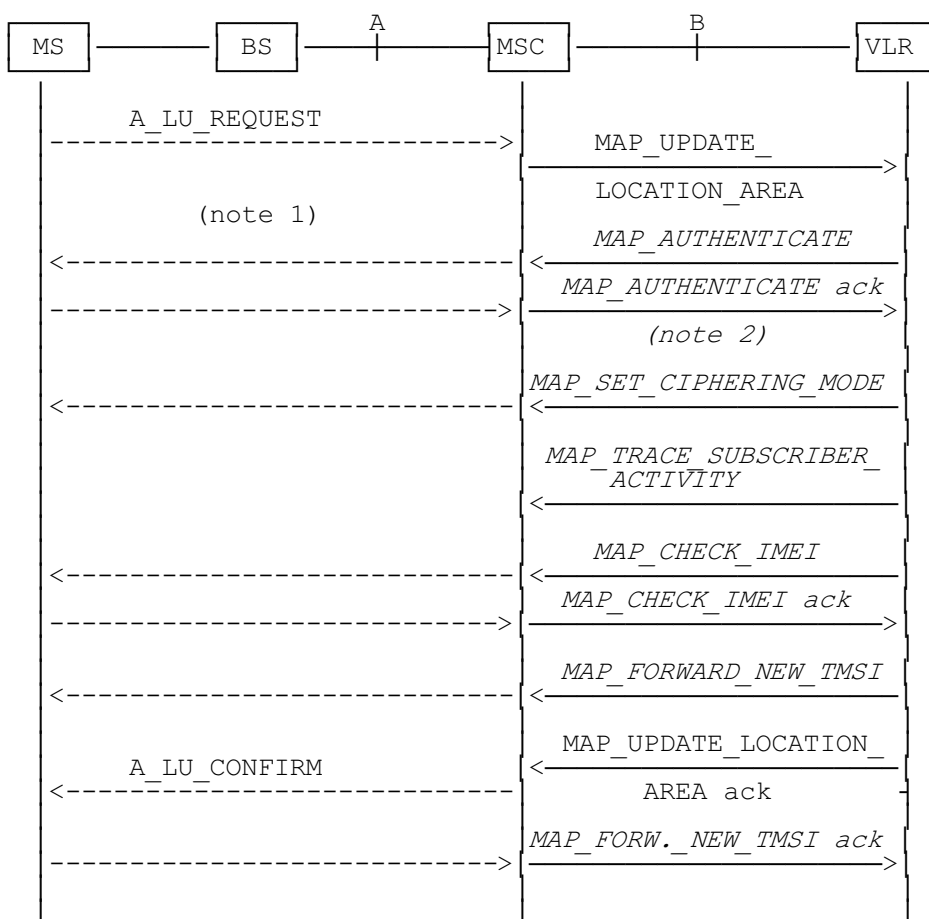
NOTE 1: For details of the procedure on the radio path, see GSM 04.08. The services shown in dotted lines indicate the trigger provided by the signalling on the radio path, and the signalling triggered on the radio path.

NOTE 2: Optional services are printed in *italics*.

Figure 16.1.1/1: Interface and services for location updating when roaming within an visitor location registers area (without need to update HLR)
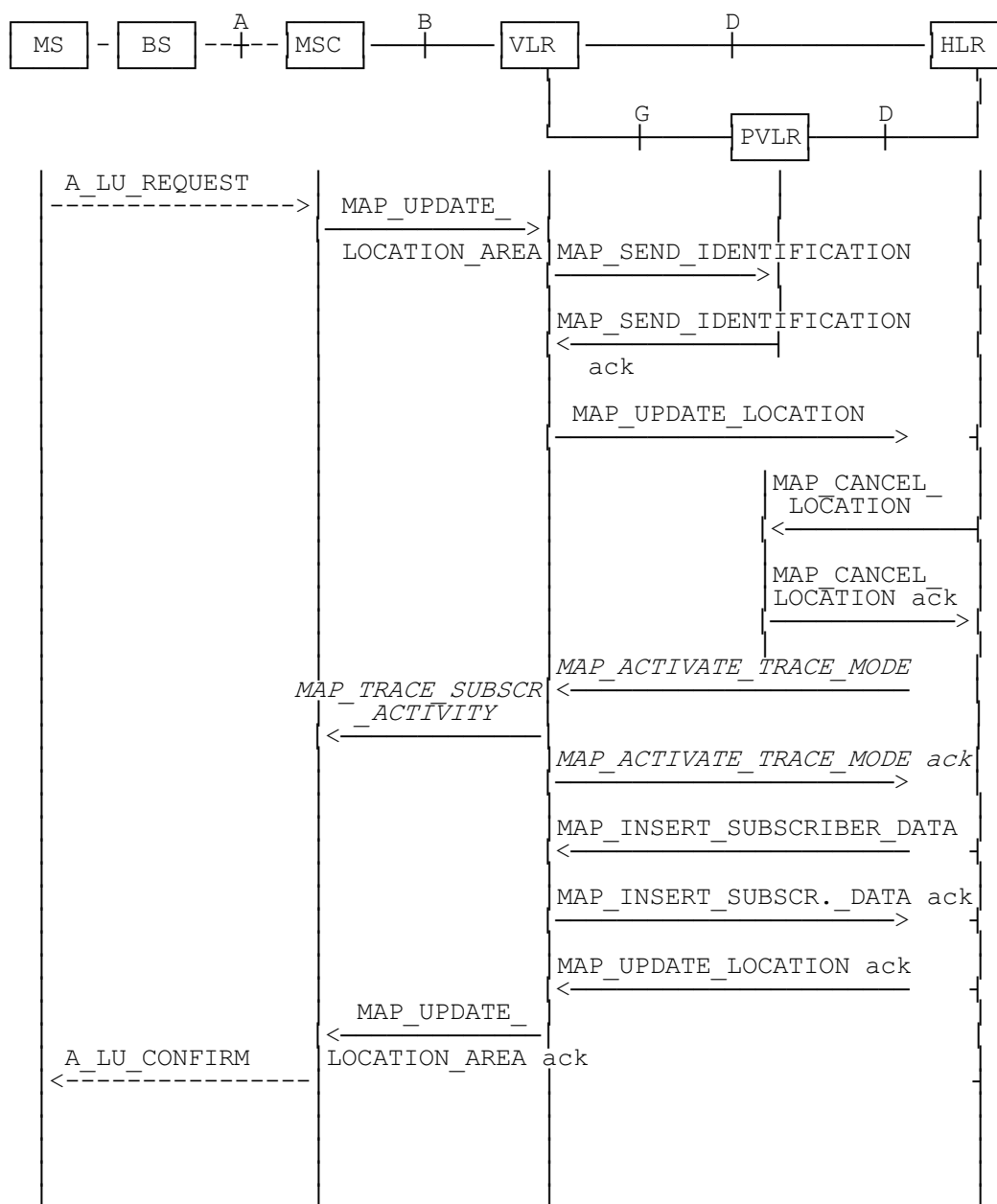
NOTE: The optional procedures in figure 16.1.1/1 apply here respectively.

Figure 16.1.1/2: Interface and services for location updating when changing the VLR area

NOTE: The optional procedures in figure 16.1.1/1 apply here respectively.
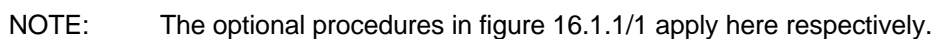
Figure 16.1.1/3: Interface and services for location updating involving both a VLR and an HLR, when IMSI can not be retrieved from the previous VLR

16.1.1.2   Detailed procedure in the MSC

Figure 16.1.1/4 shows the MSC process for location register updating, containing macro calls for:

Receive_Open_Cnf	subclause 21.1;
Authenticate_MSC	subclause 21.5;
Check_IMEI_MSC	subclause 21.6;
Obtain_IMSI_MSC	subclause 21.8;
Trace_Subscriber_Activity_MSC	subclause 21.9.

For structuring purposes, the second part of the process is placed into the macro Update Location Completion MSC, which is specific to this process (see figure 16.1.1/5).

When the VLR receives an A_LU_REQUEST (normal location updating, periodic location updating or IMSI attach) for a subscriber via the radio path, the MSC opens a dialogue to the VLR (MAP_OPEN request without any user specific parameters) and sends a MAP_UPDATE_LOCATION_AREA request, containing the parameters provided in the A_LU_REQUEST by the MS or BSS (for the parameter mapping see GSM 09.10).

If the dialogue is rejected or the VLR indicates a fallback to the version 1 procedure (see Receive_Open_Cnf macro in subclause 21.1), the MSC will send an A_LU_Rej towards the MS and terminate the procedure.

If the dialogue is accepted, the VLR will process this updating request, invoking optionally the MAP_PROVIDE_IMSI,   MAP_TRACE_SUBSCRIBER_ACTIVITY,   MAP_CHECK_IMEI   or   the MAP_AUTHENTICATE services first (see subclause 16.1.1.3 for initiation conditions, subclause 21 for macros defining the handling of services in the MSC). For these macros there are two possible outcomes:

-       a   positive   outcome,   in   which   case   the   process   continues   waiting   for   the MAP_UPDATE_LOCATION_AREA confirmation; or

-       an   error   is   reported,   in   which   case   the   process   terminates   (not   applicable   for Trace_Subscriber_Activity_MSC, which has only a positive outcome).

After receiving the MAP_UPDATE_LOCATION_AREA indication and handling these optional services, the VLR will decide whether a new TMSI need to be allocated to the subscriber or not.

Updating without TMSI reallocation

If the VLR does not reallocate the TMSI, the MSC will receive a MAP_UPDATE_LOCATION_AREA confirmation next (figure 16.1.1/4).

-       if there are no parameters with this primitive, updating was successful and a confirmation will be sent to the MS;

-       if there is an error cause contained in the received primitive, this cause will be mapped to the corresponding cause in the confirmation sent to the MS (see GSM 09.10 for the mapping of messages and causes).

Updating including TMSI reallocation

This case is covered by the macro Update Location Completion MSC given in figure 16.1.1/5. The MSC will upon receipt of a MAP_SET_CIPHERING_MODE request send a ciphering command towards BSS/MS. Thereafter, the MAP_FORWARD_NEW_TMSI indication and the MAP_UPDATE_LOCATION_AREA confirmation are received in arbitrary order, causing a confirmation on the radio path containing both new LAI and new TMSI. If the MAP_UPDATE_LOCATION_AREA confirmation contains any error, the updating request is rejected towards the MS:

- the MS will confirm receipt of the new TMSI, resulting in an empty MAP_FORWARD_NEW_TMSI response terminating the dialogue;

- if there is no confirmation received from the A-interface, the dialogue is terminated locally.

Before receiving a MAP_UPDATE_LOCATION_AREA confirmation, the MSC may receive a MAP_CHECK_IMEI indication. Handling of this indication, comprising IMEI request towards the MS and IMEI checking request towards the EIR, is given in the macro description in subclause 21.6. The result may either be to return to the state Wait for TMSI or to return to terminate.

Forwarding the Check SS Indication

When the VLR receives a MAP_FORWARD_CHECK_SS_INDICATION_Ind during the Update LOCATION Area process, this indication is relayed to the MS (see GSM 09.11 for detailed interworking) and the MSC remains in the current state.

Abort handling

If the VLR receives a MAP_U_ABORT, a MAP_P_ABORT or a premature MAP_CLOSE indication from the VLR during the location update process, the MSC terminates the process by sending an A_LU_CONFIRM containing the error cause Updating Failure to the MS. If the MSC had already confirmed the location update towards the MS, the process terminates without notification towards the A-interface.

If the MSC receives a MAP_NOTICE indication, it issues a MAP_CLOSE and terminates the A-interface dialogue, and the process terminates.

When the procedure is terminated abnormally on the radio path, the dialogue towards the VLR is aborted with the appropriate diagnostic information, and the procedure terminates.

Figure 16.1.1/4: Process Update_Location_Area_MSC

Figure 16.1.1/5: Macro Update_Location_Completion_MSC

16.1.1.3   Detailed procedure in the VLR

Figure 16.1.1/6 shows the process for location updating in the VLR. The following general macros are used:

Receive_Open_Ind	subclause 21.1;
Receive_Open_Cnf	subclause 21.1;
Authenticate_VLR	subclause 21.5;
Check_IMEI_VLR	subclause 21.6;
Insert_Subscriber_Data_VLR	subclause 21.7;
Obtain_IMSI_VLR to request the IMSI for the subscriber	subclause 21.8;
Activate_Tracing_VLR and Trace_Subscriber_Activity_VLR	subclause 21.9,
Subscriber_Present_VLR	subclause 21.10.

Additionally, the process specific macro

Location_Update_Completion_VLR, for optional initiation of Ciphering and TMSI reallocation as for acknowledgement of the MAP_UPDATE_LOCATION_AREA service, see figure 16.1.1/7,

and the optional process specific macro

VLR_Update_HLR to update the HLR and download subscriber data from there, see figure 16.1.1/8,

are invoked by this process.

Process Initiation

The location area updating process will be activated by receiving a MAP_UPDATE_LOCATION_AREA indication from the MSC. If there are parameter errors in the indication, the process is terminated with the appropriate error sent in the MAP_UPDATE_LOCATION_AREA response to the MSC. Else, The behaviour will depend on the subscriber identity received, either an IMSI or an TMSI.

Updating using IMSI

If the subscriber identity is an IMSI, the VLR checks whether the subscriber is unknown (i.e. no IMSI record). If so, the indicator "Location Information Confirmed in HLR" is set to "Not Confirmed" to initiate HLR updating later on. If the IMSI is known, the VLR checks whether the previous location area identification (LAI) provided in the primitive received from the MSC belongs to this VLR. If it does not, the indicator "Location Information Confirmed in HLR" is set to "Not Confirmed" to initiate HLR updating later on. The process may continue in both cases with the authentication check (see below).

Updating using TMSI

If the subscriber identity is a TMSI, the VLR checks whether the previous location area identification (LAI) provided in the primitive received from MSC belongs to an area of this VLR:

- if so, the TMSI will be checked. In case of location area change within a VLR, the TMSI should be known and the process may continue with the authentication check. Additionally, the indicator "Location Information Confirmed in HLR" is set to "Not confirmed" and the trace activity status is checked in case the target Location Area Id belongs to a new MSC.

- if the TMSI is not known or the subscriber data stored are incomplete, e.g. because the new LA belongs to a different VLR or due to VLR restoration, the indicator "Confirmed by VLR" is set to "Not Confirmed" to initiate HLR updating later on.

If the subscriber has not already been registered in the VLR, i.e. the previous LAI belongs to a different VLR, the indicators "Confirmed by HLR" and "Location Information Confirmed in HLR" are set to "Not Confirmed" and the VLR checks whether the identity of the Previous VLR (PVLR) is derivable from the previous LAI:

- if so, the IMSI and authentication parameters are requested from that VLR using the MAP_SEND_IDENTIFICATION service (see sheet 3 of figure 16.1.1/6), containing the subscriber's TMSI.

- if the dialogue is rejected by the PVLR, the process continues requesting the IMSI from the MS. In case the PVLR reverts to the MAP version one dialogue, the VLR will perform the respective procedure of version one, too, with outcomes as for the version two dialogue. Else, the process waits the for the respective MAP_SEND_IDENTIFICATION response from the PVLR:

    - if the IMSI is received in that primitive, the process continues with the authentication check;

    - if the IMSI is not received from the previous VLR for any reason, the dialogue to the PVLR is terminated and the IMSI will be requested from the MS;

    - if a MAP_NOTICE indication is received from the PVLR, the dialogue will be terminated by sending a MAP_CLOSE indication, and the process continues requesting the IMSI from the MS;

    - if a MAP_P_ABORT or MAP_U_ABORT indication is received from the MSC while waiting for the MAP_SEND_IDENTIFICATION response, the process is terminated;

    - if a MAP_NOTICE indication is received from the MSC while waiting for the MAP_SEND_IDENTIFICATION response, the dialogue with the PVLR will be aborted by sending a MAP_U_ABORT indication (Remote Operations Failure), the dialogue with the MSC will be terminated by sending a MAP_CLOSE and the process terminates;

- if the identity of the previous VLR cannot be derived, the process continues by requesting the IMSI from the MS.

Requesting IMSI from the MS

For requesting the IMSI from the MS, the macro Obtain_IMSI_VLR described in subclause 21.8 is invoked (see figure 16.1.1/6 sheet 3). The outcome will be:

- OK, i.e. receipt of IMSI, in which case the process continues with the authentication check described below; or

- receipt of an Absent Subscriber error, indicating that the MS did not respond. In this case the System Failure error is reported in the MAP_UPDATE_LOCATION_AREA response towards the MSC and the updating process is terminated;

- aborted, i.e. the MSC dialogue has been released while waiting for the IMSI. In this case the updating process is terminated, too.

Authentication check

After a subscriber identity has been received, either in the service indication or by an explicit request procedure, the VLR checks whether authentication of this identity is required (see figure 16.1.1/6 sheet 2). If so, the authentication macro described in subclause 21.5 is invoked. The outcome of this macro can be:

- OK, i.e. the subscriber has been authenticated successfully, in which case the process is continued by setting the indicator "Confirmed by Radio Contact" to "Confirmed" and updating the location information held in the register. Thereafter,
    - if one or both of the indicators "Confirmed by HLR" and "Location Information Confirmed in HLR" is set to "Not Confirmed", HLR updating is invoked first;
    - otherwise the process continues with the Location Update Completion VLR macro described below, and the register is updated after successful completion of this macro.

- Illegal subscriber, i.e. there was a mismatch between expected and received SRES. The VLR checks whether authentication had been performed using the TMSI, in which case a new authentication attempt with IMSI may be started (VLR operator option).
    - if so, the process continues by requesting the IMSI from the MS;
    - else, the Illegal Subscriber error is reported in the MAP_UPDATE_LOCATION_AREA response.

- Unknown Subscriber, i.e. the IMSI given is unknown in the HLR. In this case, the subscriber data are deleted in the VLR and the same error is returned in the MAP_UPDATE_LOCATION_AREA response.

- Procedure error, i.e. the authentication process was unsuccessful for some other reason, e.g. because of a failure while requesting authentication information from the HLR. In this case the System Failure error is reported in the MAP_UPDATE_LOCATION_AREA response.

- Null, indicating impossible dialogue continuation (e.g. termination of the radio path), and leading to procedure termination without any further action.

Updating the HLR

If the HLR is to be updated, the VLR_Update_HLR macro described below is performed, with one of the following results (see sheet 4 of figure 16.1.1/6):

- OK, if HLR updating has been completed successfully. The response will contain the HLR number as parameter. Next, the Location_Update_Completion VLR macro is invoked (checking amongst others the roaming restrictions and regional subscription data), and upon successful outcome of this macro the register is updated and the process terminates.

- Roaming Not Allowed, qualified by PLMN Roaming Not Allowed if the location information indicates a PLMN for which the subscriber has no subscription or if the subscribers HLR cannot be reached (e.g. SS7 links to the subscribers HPLMN do not yet exist). In this case, the error Roaming Not Allowed qualified by PLMN Roaming Not Allowed is sent in the MAP_UPDATE_LOCATION_AREA response. The Subscriber Data are deleted in the VLR.

- if Roaming Not Allowed was qualified by the parameter Operator Determined Barring, the same value is sent in the MAP_UPDATE_LOCATION_AREA response to the MSC. The subscriber data are deleted in the VLR.

- Unknown Subscriber, if the subscriber is not known in the HLR. In this case, the subscriber data are deleted in the VLR, and the same error is sent in the MAP_UPDATE_LOCATION_AREA response.

- Procedure error, if there occurs some other error during HLR updating (e.g. abort of the connection to HLR):
  - if the VLR can proceed in stand alone mode (VLR operator option), the Location Update Completion VLR macro is invoked to complete the VLR updating, and the indicator "Confirmed by HLR" remains unchanged;
  - otherwise, the System Failure error is sent in the MAP_UPDATE_LOCATION_AREA response.

- Aborted, indicating that during HLR updating the MSC dialogue has been terminated. In this case, the updating process terminates without any further action.

The macro Location Update Completion VLR

This macro completes the VLR updating process. First, the VLR checks whether there is a roaming restriction for the subscriber (see figure 16.1.1/7):

- if the target LA is not allowed for the subscriber due to national roaming restrictions, the error Roaming Not Allowed with cause National Roaming Not Allowed is returned in the MAP_UPDATE_LOCATION_AREA response towards the MSC.

  The subscriber data are not deleted from VLR, to avoid unnecessary HLR updating when roaming into other LAs of the same MSC. An indication that the subscriber is not allowed to roam is set in the VLR (LA Not Allowed Flag set to not allowed). As a consequence the subscriber is not reachable (checked for MTC, SMS and MT USSD) and cannot perform outgoing actions (checked in Access Management).

- if the target LA is not allowed for the subscriber because of regional subscription data (Zone Code List) or Roaming Restriction Due To Unsupported Feature stored in the VLR, the error Roaming Not Allowed with cause Location Area Not Allowed is returned towards the MSC in the MAP_UPDATE_LOCATION_AREA response.

  Also in this case the subscriber data are not deleted from VLR, to avoid unnecessary HLR updating when roaming into other LAs of the same MSC. The LA Not Allowed Flag is set to not allowed in the VLR.

- if, after check of possible roaming restrictions, the subscriber is allowed to roam in the target LA, the LA Not Allowed Flag is set to allowed (if necessary), the IMSI Detached Flag is set to attached and the process SUBSCRIBER_PRESENT_VLR is started; this may inform the HLR that the subscriber is present again to retry an SMS delivery (see subclause 16.1.1.7). Thereafter, the VLR checks whether TMSI reallocation is required.
  - if so, the VLR sends a MAP_SET_CIPHERING_MODE request containing:
    - Ciphering Mode (version 1 GSM); and
    - Kc, the cipher key to be used.

- if IMEI checking is required by the operator, the VLR will invoke the CHECK_IMEI_VLR macro (see subclause 21.6) to initiate both requesting IMEI from the MS and checking of this IMEI towards the EIR. As result either the service is granted, with process continuation as given below, or the service is rejected, in which case the VLR marks the subscriber as detached and returns an Illegal Equipment error in the MAP_UPDATE_LOCATION_AREA response before the process terminates.
  - the VLR then sends a MAP_FORWARD_NEW_TMSI request containing the new TMSI, and the MAP_UPDATE_LOCATION_AREA response containing no parameters. The process will thereafter wait for the MAP_FORWARD_NEW_TMSI confirm. If this indicates a negative outcome, or if a MAP_P_ABORT or a MAP_U_ABORT primitive is received, the old TMSI is frozen. Subsequent accesses of the MS shall be accepted with both old or new TMSI.

- if TMSI reallocation is not required, the VLR invokes the CHECK_IMEI_VLR macro (see subclause 21.6) to initiate both requesting IMEI from the MS and checking of this IMEI towards the EIR, if IMEI Checking is required by the operator. As a result, either the service is granted, in which case the MAP_UPDATE_LOCATION_AREA response is sent without any parameters, or the service is rejected, in which case an Illegal Equipment error is returned in the MAP_UPDATE_LOCATION_AREA response, before the process terminates.

In all cases where the VLR sends a MAP_UPDATE_LOCATION_AREA response to the MSC, the dialogue towards the MSC is terminated by a MAP_CLOSE request with parameter Release Method indicating Normal Release.

The macro VLR Update HLR

This macro is invoked by the VLR process for location updating or by some other process handling the first subscriber access to the network after a register failure in order to perform HLR updating. If the VLR does not know the subscribers HLR (e.g. no IMSI translation exists as there are not yet any SS7 links to the subscribers HPLMN), the error Roaming Not Allowed with cause PLMN Roaming Not Allowed is returned.

If the subscribers HLR can be reached, the VLR opens a dialogue towards the HLR (see figure 16.1.1/8) by sending a MAP_OPEN request without any user specific parameters, together with a MAP_UPDATE_LOCATION request containing the parameters

- IMSI, identifying the subscriber;

- Location Info, containing the MSC number;

- VLR Number, the E.164 address of the VLR, to be used by the HLR when addressing the VLR henceforth (e.g. when requesting an MSRN);

- the LMSI as an VLR operator option; this is a subscriber identification local to the VLR, used for fast data base access.

In case the HLR rejects dialogue opening (see subclause 21.1), the VLR will terminate the procedure indicating procedure error. If the HLR indicates version one protocol to be used, the VLR will revert to the version one procedure concerning the dialogue with the HLR, with outcomes as for the version two procedure.

If the HLR accepts the dialogue, the HLR will respond with:

- a MAP_INSERT_SUBSCRIBER_DATA indication, handled by the macro Insert_Subs_Data_VLR defined in subclause 21.7;

  NOTE: The HLR may repeat this service several times depending on the amount of data to be transferred to the VLR and to replace subscription data in case they are not supported by the VLR.

- a MAP_ACTIVATE_TRACE_MODE indication, handled by the macro Activate_Tracing_VLR defined in subclause 21.9;

- a MAP_FORWARD_CHECK_SS_INDICATION_ind. This indication will be relayed to the MSC without any change of the current state.

- the MAP_UPDATE_LOCATION confirmation:

  - if this confirmation contains the HLR Number, this indicates that the HLR has passed all information and that updating has been successfully completed. The VLR is updated using the parameters provided in the service and needed by the VLR. If certain parameters are not needed in the VLR, e.g. because some service is not supported, the corresponding data may be discarded. The VLR sets the "Confirmed by HLR" and "Location information confirmed in HLR" indicators to "Confirmed" to indicate successful subscriber data updating;

  - if the confirmation contains an User error cause (Unknown Subscriber, Roaming Not Allowed or some other), the process calling the macro continues accordingly. In the last case, the subscriber data are marked as incomplete by setting the indicators "Confirmed by HLR" and "Location information confirmed in HLR" to "Not Confirmed". The same holds if there is a Provider error or a Data error in the confirmation;

- a MAP_P_ABORT, MAP_U_ABORT, or MAP_CLOSE indication. In these cases, the subscriber data are marked to be incomplete and the process continues as in the case of an error reported by the HLR;

- a MAP_NOTICE indication. Then, the dialogue towards the HLR is terminated, the subscriber data are marked to be incomplete and the process continues as in the case of an error reported by the HLR;

- if during HLR updating the VLR receives a MAP_P_ABORT, MAP_U_ABORT or a MAP_CLOSE indication concerning the MSC dialogue, the process is terminated by sending a MAP_U_ABORT request towards the HLR, and subscriber data are marked to be incomplete;

- if during HLR updating the VLR receives a MAP_NOTICE indication concerning the MSC dialogue, the dialogue with the MSC is terminated by sending a MAP_CLOSE, the dialogue with the HLR is terminated by sending a MAP_U_ABORT, subscriber data are marked to be incomplete and the process is terminated.

Abort Handling

If the VLR receives a MAP_NOTICE indication from the MSC while waiting for a MAP service primitive, the VLR will terminate the MSC dialogue by sending a MAP_CLOSE and any pending HLR dialogue by sending a MAP_U_ABORT (Remote Operations Failure), and the process is terminated.

Figure 16.1.1/6 (sheet 1 of 4): Process Update_Location_Area_VLR
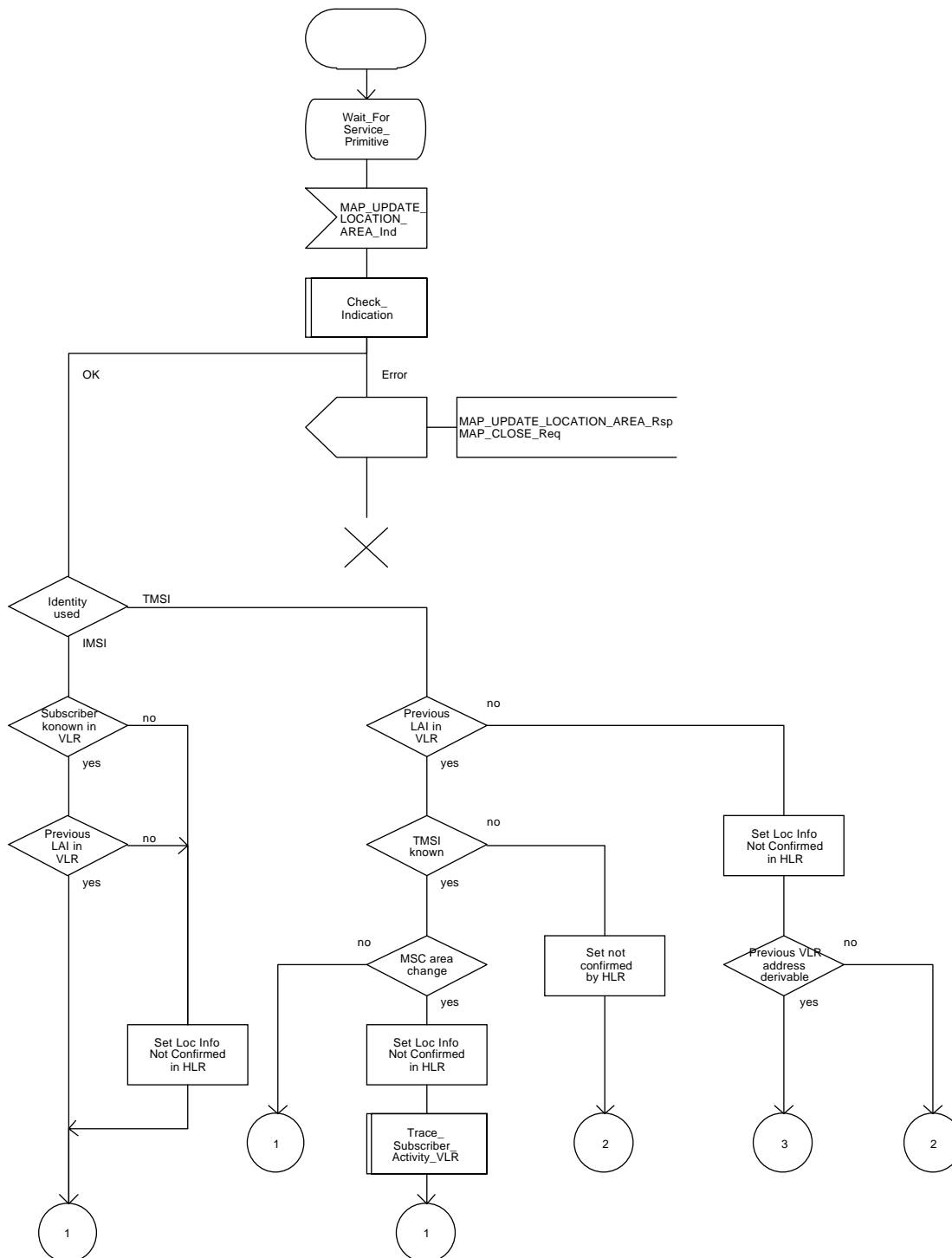
Figure 16.1.1/6 (sheet 2 of 4): Process Update_Location_Area_VLR
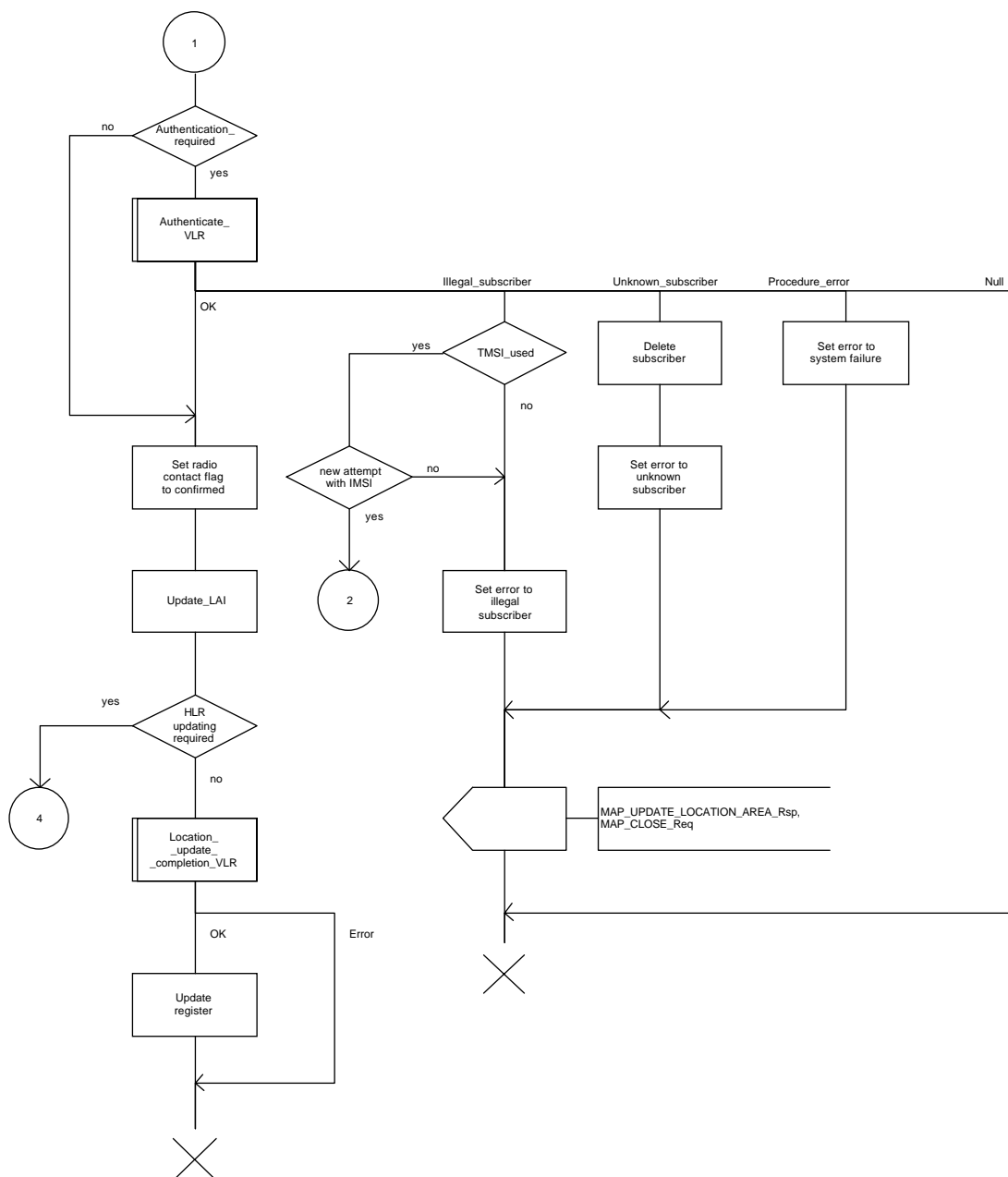
Figure 16.1.1/6 (sheet 3 of 4): Process Update_Location_Area_VLR
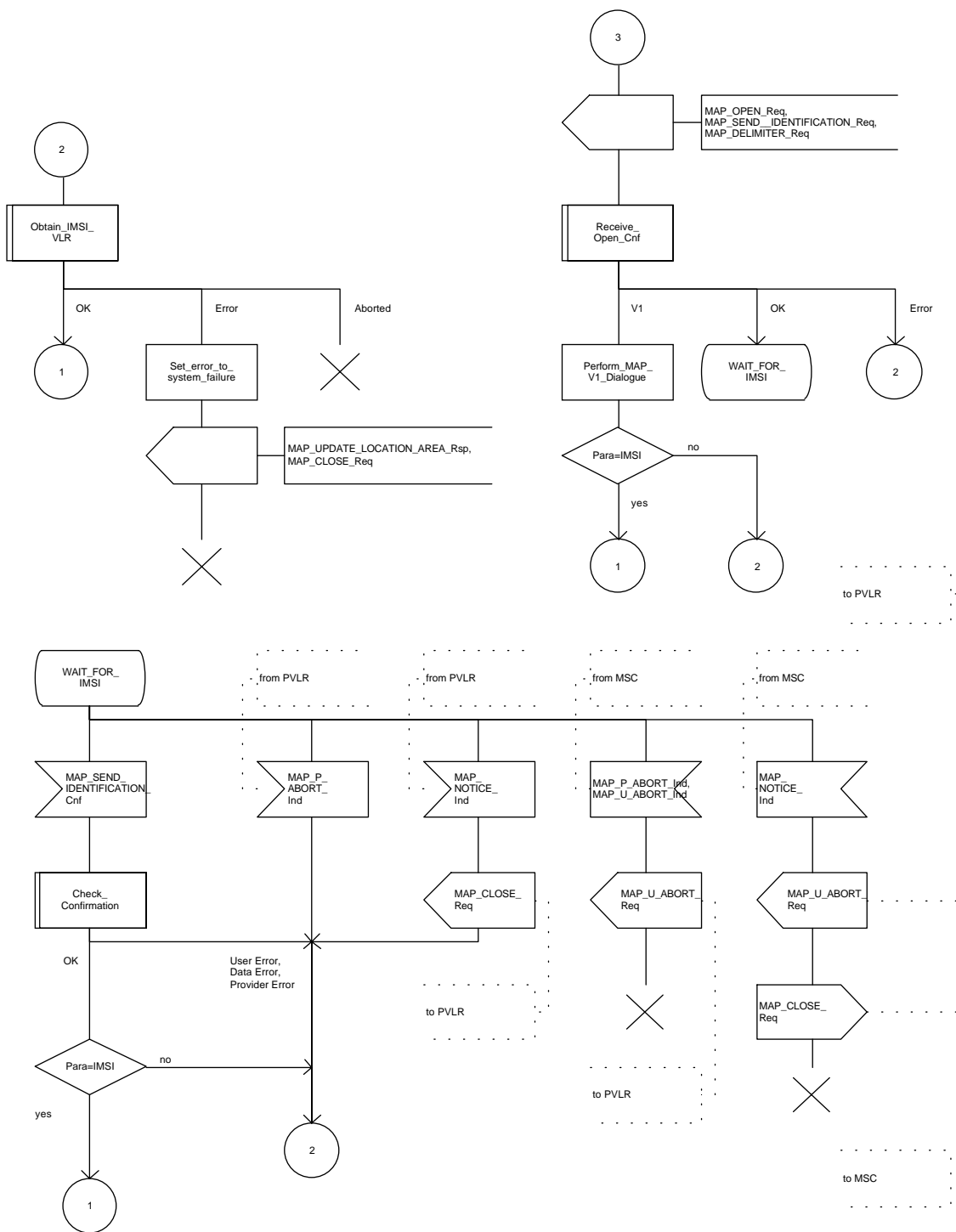
Figure 16.1.1/6 (sheet 4 of 4): Process Update_Location_Area_VLR
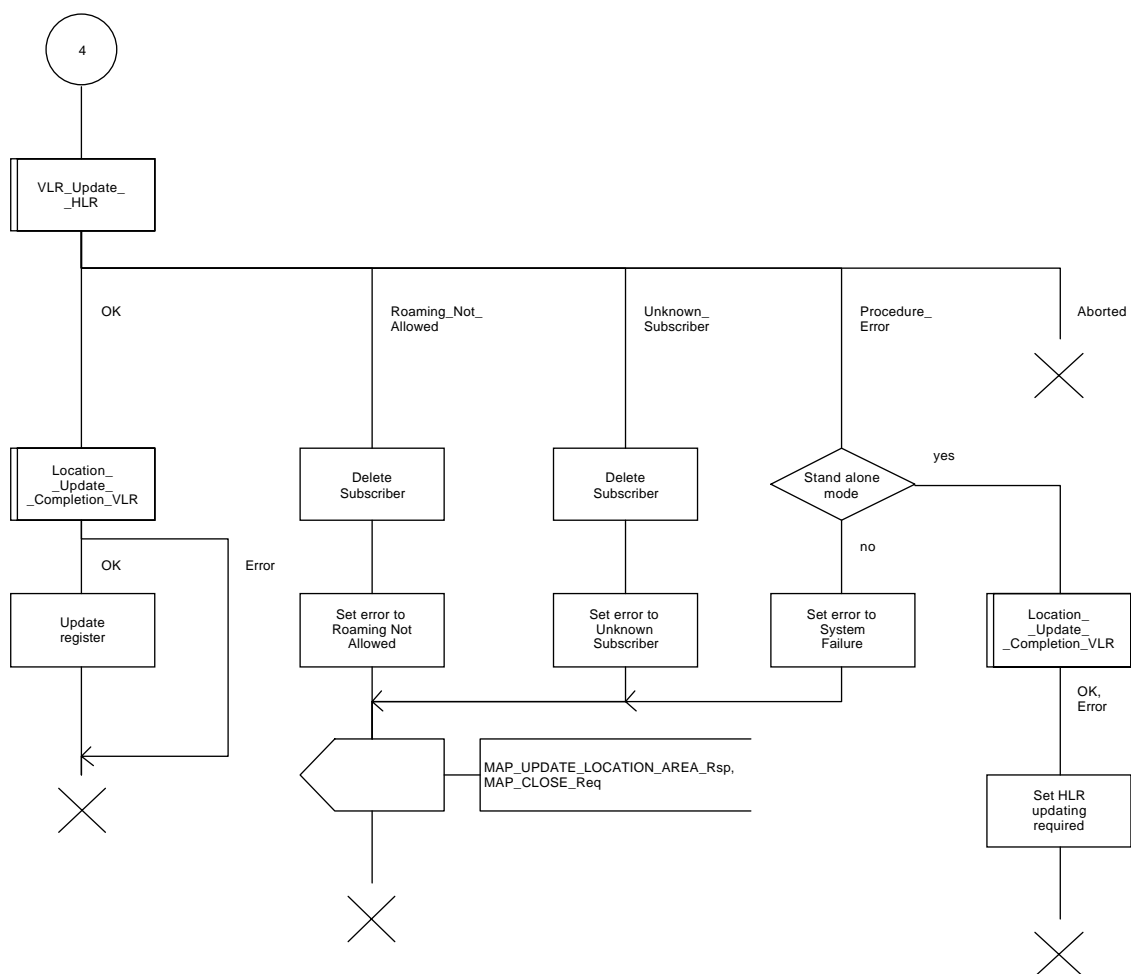
Figure 16.1.1/7: Macro Location_Update_Completion_VLR
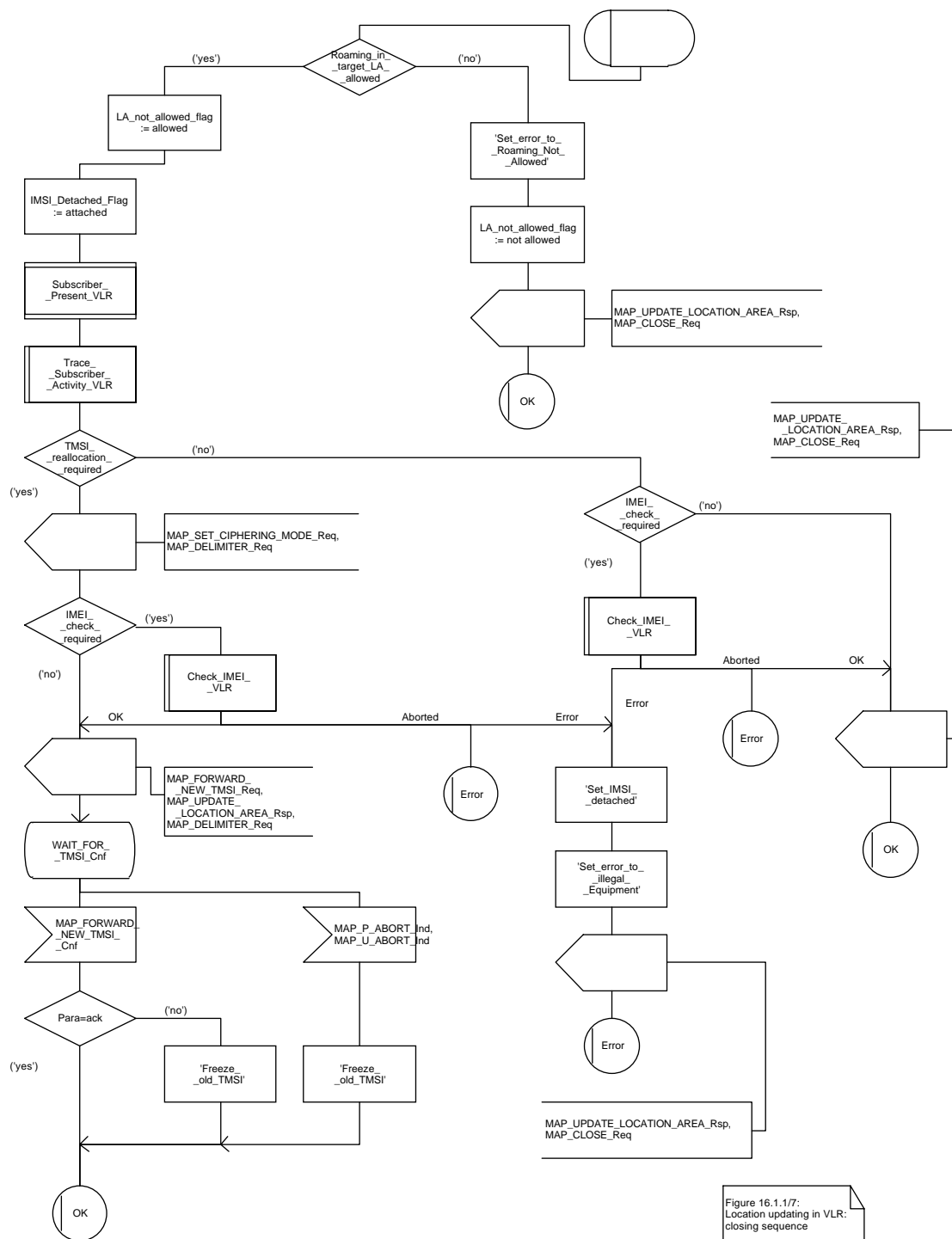
Figure 16.1.1/7:
Location updating in VLR:
closing sequence

Figure 16.1.1/8 (sheet 1 of 2): Macro VLR_Update_HLR
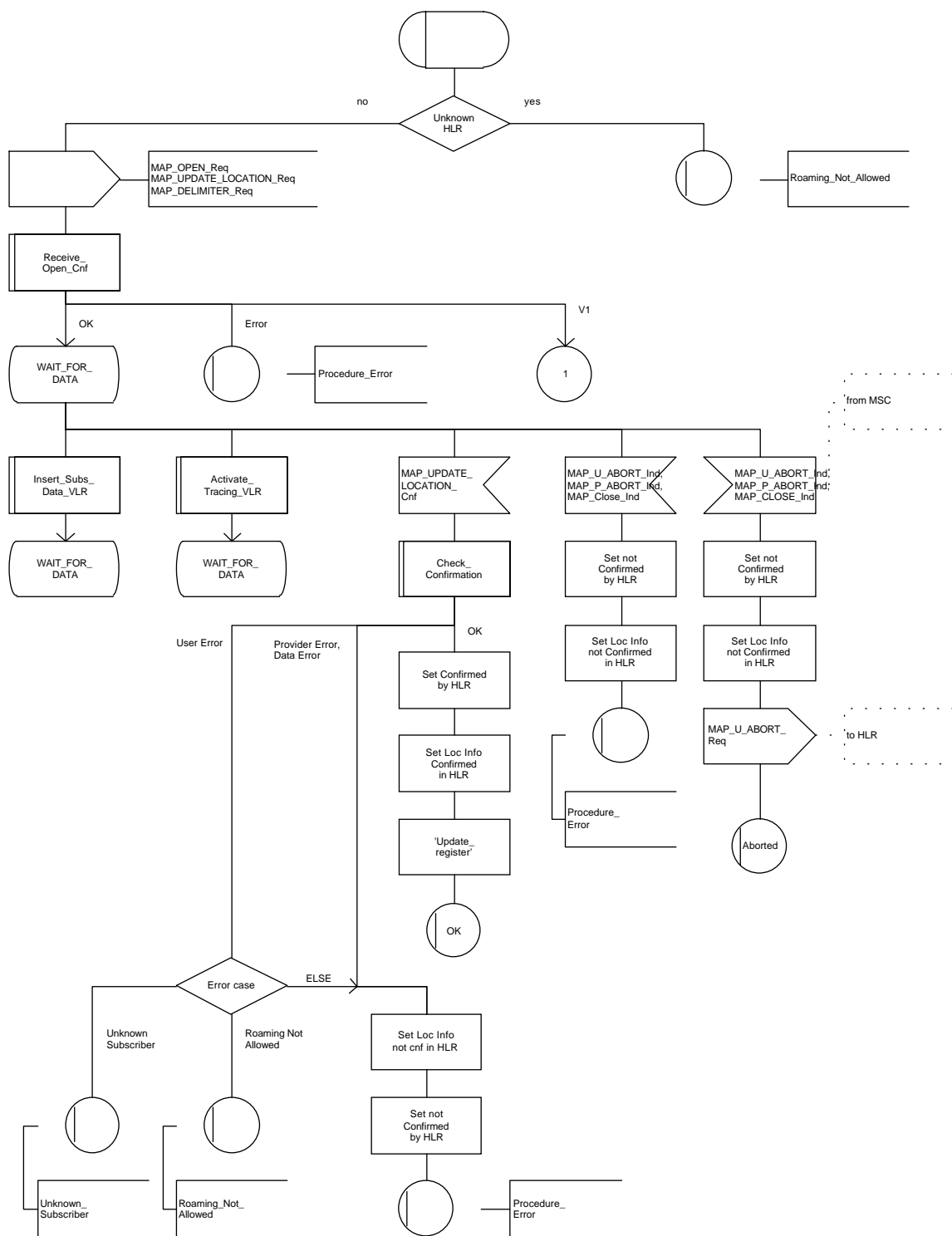
Figure 16.1.1/8 (sheet 2 of 2): Macro VLR_Update_HLR
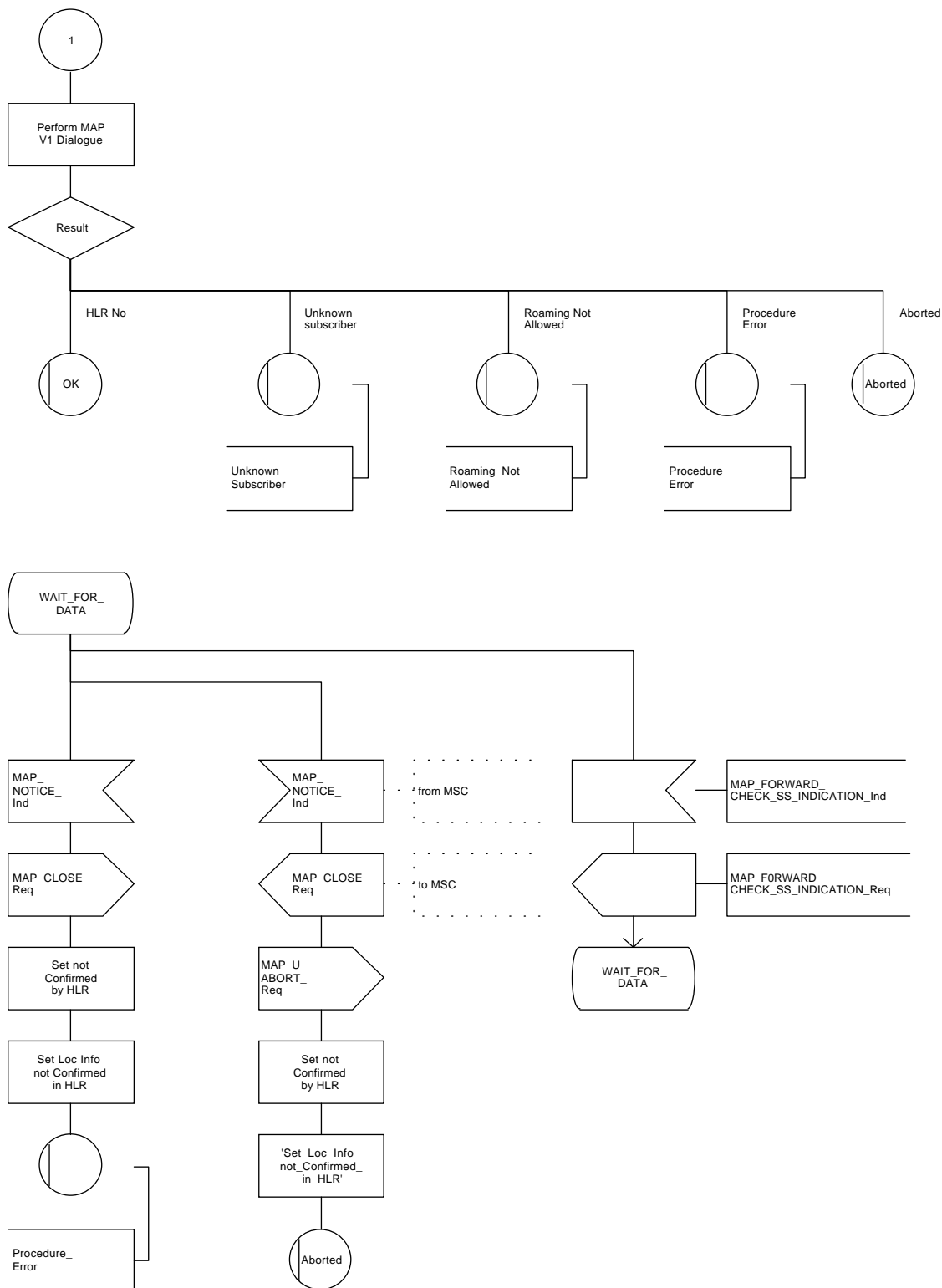
16.1.1.4   Detailed procedure in the HLR

The following macros are used by this process:

- Receive_Open_Ind, defined in subclause 21.1;
- Check_indication, defined in subclause 21.2;
- Insert_Subs_Data_Framed_HLR, described in subclause 16.4.1;
- Control_Tracing_HLR, described in subclause 21.9;

and the processes Cancel_Location_HLR (see subclause 16.1.2) and Subscriber_Present_HLR (see subclause 16.1.1.7) are invoked.

The location updating process in the HLR is activated by receipt of a MAP_UPDATE_LOCATION indication (see figure 16.1.1/9):

- if there is a parameter problem in the indication, the error Unexpected Data Value is returned in the MAP_UPDATE_LOCATION response (see Check_indication macro defined in subclause 21.2); if the subscriber is not known in the HLR, the error Unknown Subscriber is returned in the response. In either case the process terminates;

- tracing shall be set to deactive in the VLR

- if the VLR address received in the MAP_UPDATE_LOCATION indication differs from the one actually stored against the subscriber, the Cancel_Location_HLR process is started to cancel the subscriber data in the stored VLR (see subclause 16.1.2).

The next action will be to check whether the subscriber is allowed to roam into the PLMN indicated by the VLR Number given in the MAP_UPDATE_LOCATION indication:

- if the subscriber is not allowed to roam into the PLMN, the error Roaming not Allowed with cause PLMN Roaming Not Allowed is returned in the MAP_UPDATE_LOCATION response, and the routing information stored (VLR number, MSC Number, LMSI) is deleted (deregistration);

- otherwise the HLR database will be updated with information received in the indication. The HLR sets the "MS purged" flag to False and checks whether tracing is required for that subscriber. This is handled by the macro Control_Tracing_HLR described in subclause 21.9.

Thereafter, the macro Insert_Subs_Data_Framed_HLR described in subclause 16.4.1 is invoked. The outcome of this macro may be:

- aborted, in which case the process terminates;

- error, in which case the error System Failure is returned in the MAP_UPDATE_LOCATION response and the process terminates;

- OK, indicating successful outcome of downloading the subscriber data to the VLR.

The SUBSCRIBER_PRESENT_HLR process is then started to alert the Short Message Service Centre, if required (see subclause 16.1.7). Additionally, the MAP_FORWARD_CHECK_SS_INDICATION request is sent to inform the subscriber about an uncertain state of his SS-Data if this is needed due to previous HLR restoration (use of this service may be omitted as an HLR operator option).

Finally the HLR number is returned in the MAP_UPDATE_LOCATION response.

In all cases where the HLR sends a MAP_UPDATE_LOCATION response to the VLR, the dialogue towards the VLR is terminated by a MAP_CLOSE request with parameter Release Method indicating Normal Release.

Figure 16.1.1/9 (sheet 1 of 2): Process Update_Location_HLR
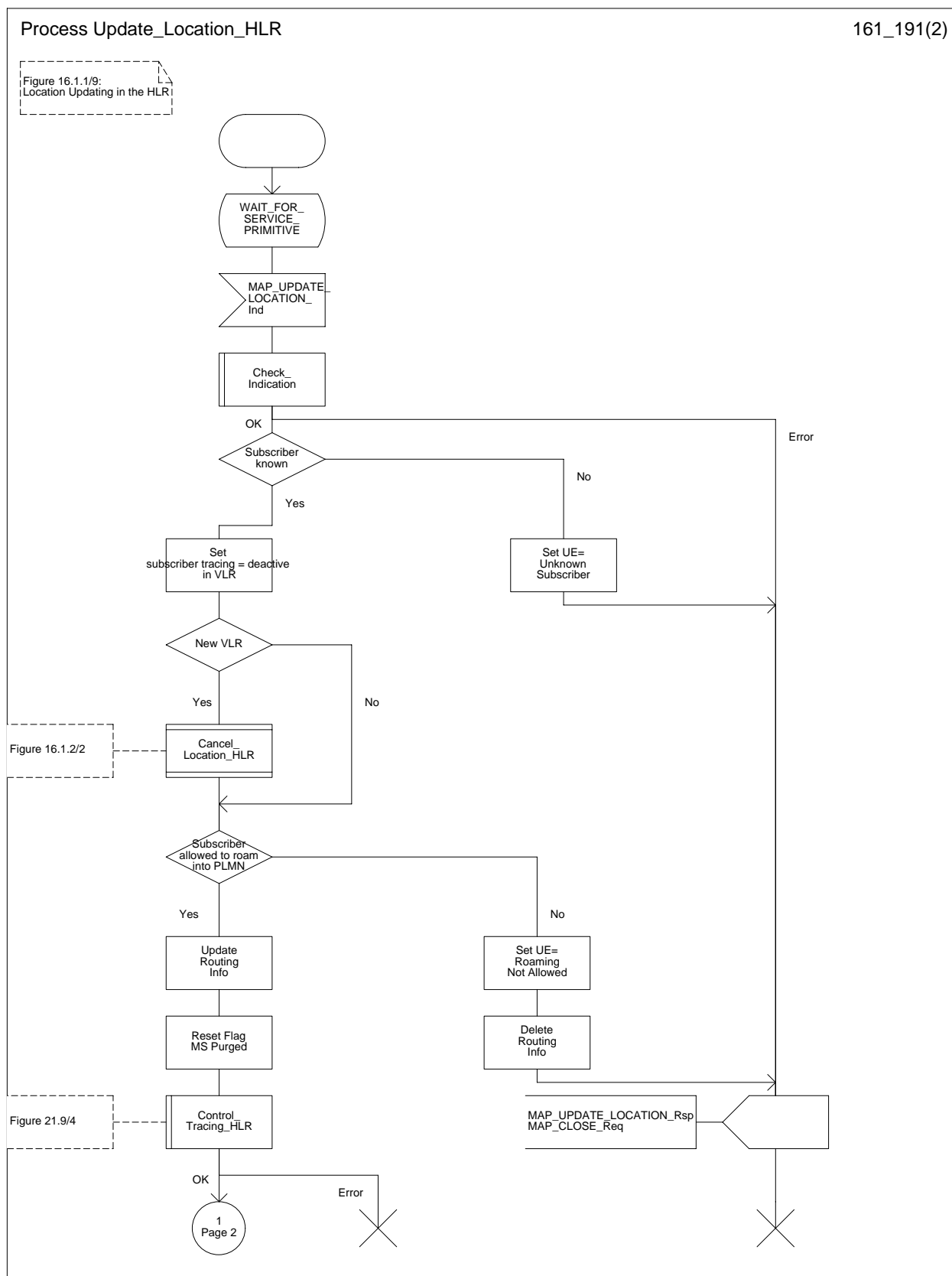
Figure 16.1.1/9 (sheet 2 of 2): Process Update_Location_HLR
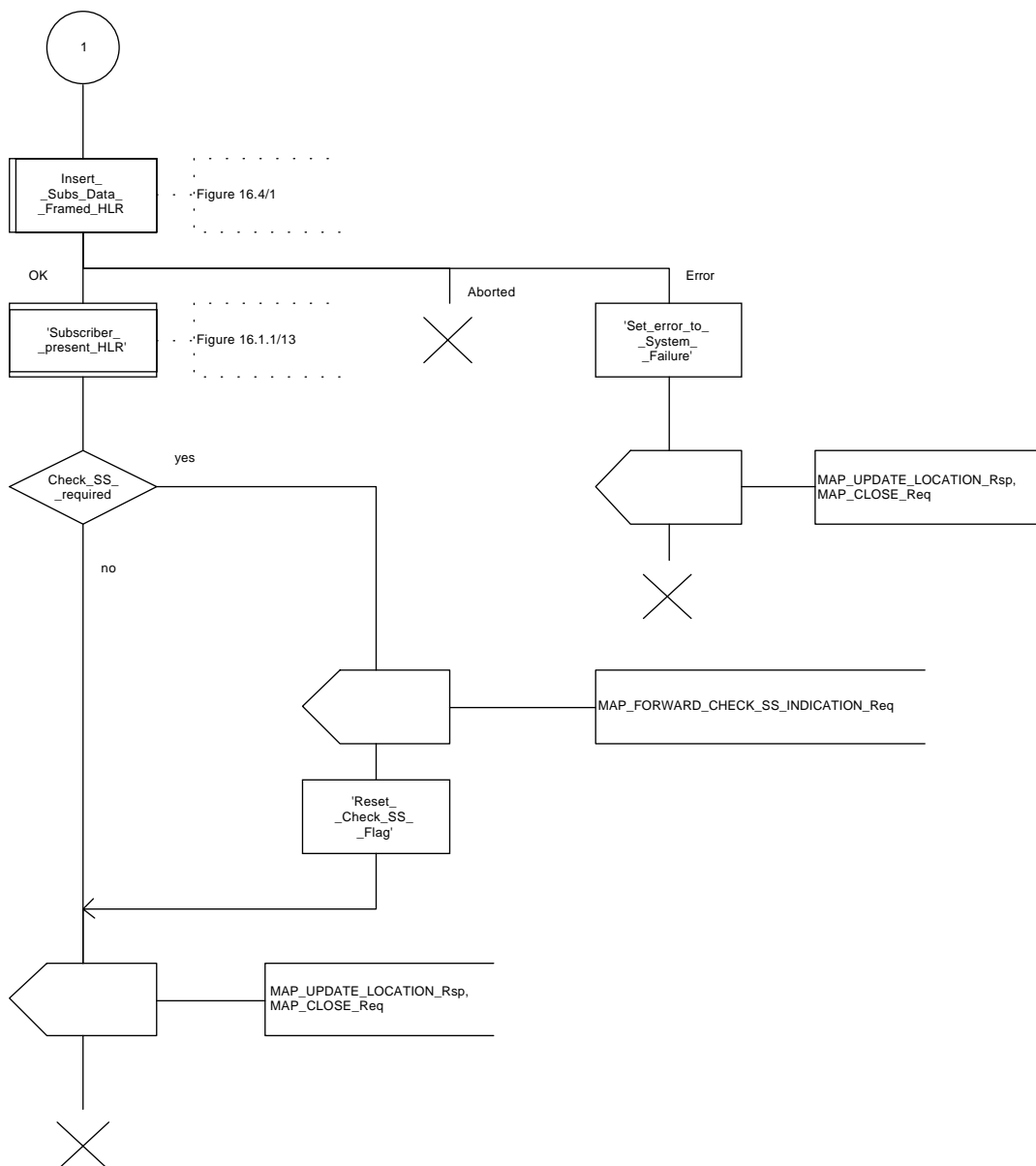
16.1.1.5   Send Identification

16.1.1.5.1   General

This service is invoked by a VLR when it receives a MAP_UPDATE_LOCATION_AREA indication containing a LAI indicating that the subscriber was registered in a different VLR (henceforth called the Previous VLR, PVLR). If the identity of the PVLR is derivable for the VLR (usually if both are within the same network), the IMSI and authentication sets are requested from the PVLR (see subclause 16.1.1.3), using the service described in subclause 6.1.4.
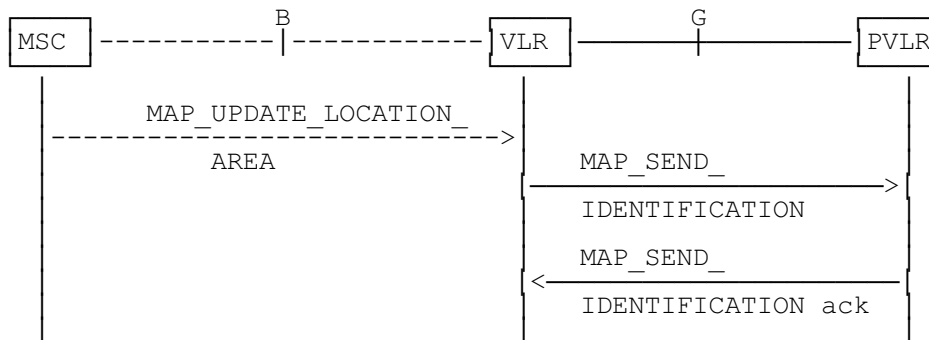
```
       ┌─────┐              B              ┌─────┐              G              ┌──────┐
       │ MSC │ ----------- | ----------- │ VLR │ ─────────── | ─────────── │ PVLR │
       └─────┘                             └─────┘                             └──────┘
          │                                   │                                   │
          │         MAP_UPDATE_LOCATION_       │                                   │
          │ ------------------------------>    │            MAP_SEND_              │
          │               AREA                 │ ─────────────────────────────>   │
          │                                    │          IDENTIFICATION           │
          │                                    │            MAP_SEND_              │
          │                                    │ <───────────────────────────     │
          │                                    │          IDENTIFICATION ack       │
          │                                    │                                   │
```

NOTE: The service shown in dotted lines indicates the trigger provided by other MAP signalling.

Figure 16.1.1/10: Interface and services for Send Identification

16.1.1.5.2   Detailed procedure in the VLR

The VLR procedure is part of the location area updating process described in subclause 16.1.1.3, see also figure 16.1.1/6 sheet 3.

16.1.1.5.3   Detailed procedure in the PVLR

On receipt of a dialogue request for the Send Identification procedure, (see Receive_Open_Ind macro in subclause 21.1), the PVLR will:

-        terminate the procedure in case of parameter problems;

-        revert to the MAP version one procedure (Send Parameters with request for IMSI plus authentication sets) in case the VLR indicated version one protocol; or

-        continue as below, if the dialogue is accepted.

If the PVLR process receives a MAP_NOTICE indication, it terminates the dialogue by sending a MAP_CLOSE request.

If the PVLR process receives a MAP_SEND_IDENTIFICATION indication from the VLR (see figure 16.1.1/11), it checks whether the subscriber identity provided is known:

-        if so, the IMSI and - if available - authentication parameters for the subscriber are returned in the MAP_SEND_IDENTIFICATION response;

-        if not, the error Unidentified Subscriber is returned in the MAP_SEND_IDENTIFICATION response.

In all cases where the PVLR sends a MAP_SEND_IDENTIFICATION response to the VLR, the dialogue towards the VLR is terminated by a MAP_CLOSE request with parameter Release Method indicating Normal Release.

Figure 16.1.1/11: Process Send_Identification_PVLR
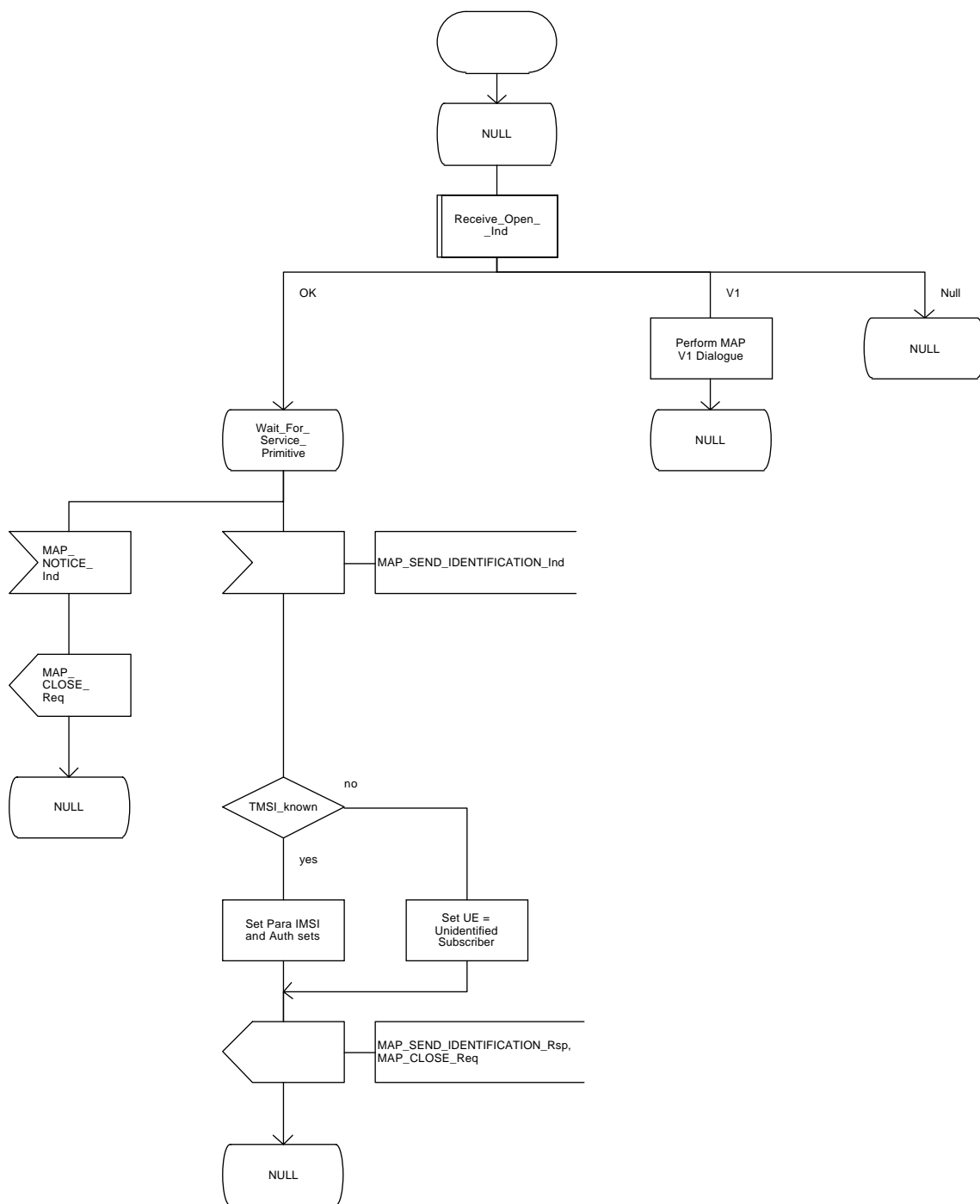
16.1.1.6   The Process Update Location VLR

This process is started by some other MAP user process in case the HLR need to be updated due to previous network failure. It is invoked when the subscriber accesses the network, e.g. for mobile originated call set-up, response to paging or supplementary services handling. Here, location updating consists only of invoking the macro VLR_Update_HLR described above (see subclause 16.1.1.3), which performs HLR updating and downloading of subscriber data.

> If updating is successful (OK) the HLR Number is received in the MAP_UPDATE_LOCATION confirm primitive and the process terminates.

> If one of the errors Roaming not Allowed or Unknown Subscriber is received instead, all subscriber data are deleted from the VLR before the process terminates.

> In case some other error occurs during HLR updating, the process simply terminates. Note, in all error cases the initiating restoration flags in VLR remain false, therefore a new HLR updating attempt will be started later on.

> NOTE: This process will be performed independent from the calling process, no coordination is required.

Figure 16.1.1/12: Process UL_VLR
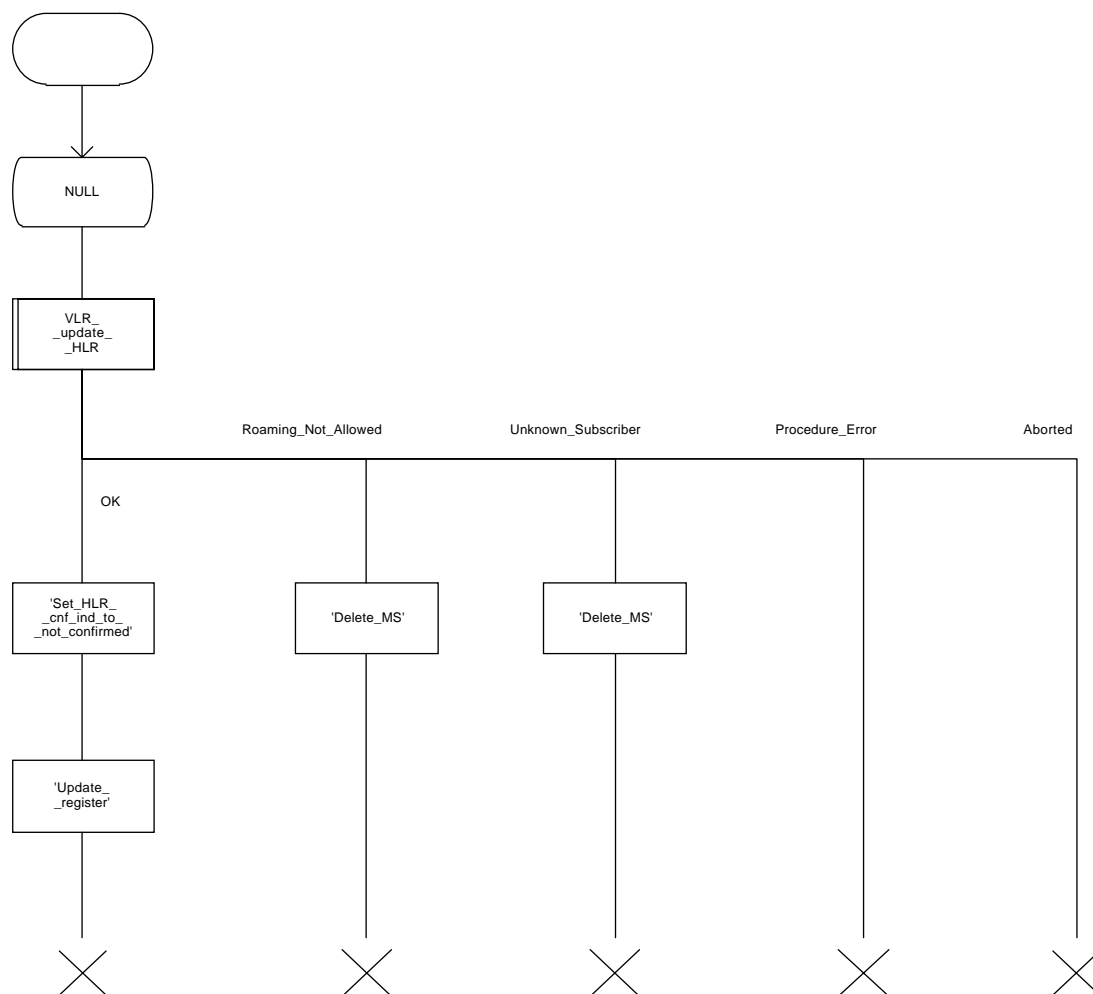
Figure 16.1.1/12:
Location updating initiated
by VLR: VLR process

16.1.1.7 The Process Subscriber Present HLR

The process Subscriber Present HLR is started by the location updating process in HLR to perform actions required for short message alerting. The process checks the Message Waiting Data flag, and if this is set, the macro Alert_Service_Centre_HLR defined in subclause 21.10 is invoked. This macro will alert all service centres from which there are short messages waiting for this subscriber.

Figure 16.1.1/13: Process Subscriber_Present_HLR
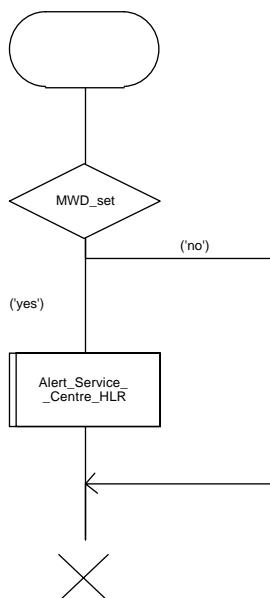
16.1.2   Location Cancellation

16.1.2.1   General

The purpose of this process is to delete a subscriber's record from a previous visitor location register after she has registered with a new visitor location register. The procedure may also be used if the subscriber's record is to be deleted for other operator determined purposes, e.g. withdrawal of subscription, imposition of roaming restrictions or modifications to the subscription which result in roaming restrictions. Location cancellation can be used to enforce location updating including updating of subscriber data in the VLR at the next subscriber access.

In all cases, the process is performed independently of the invoking process (e.g. Location Updating).

The service as described in subclause 6.1.3 is invoked when an HLR receives a MAP_UPDATE_LOCATION indication from a VLR other than that stored in its table for this subscriber. Additionally the service may be invoked by operator intervention. The MAP_CANCEL_LOCATION service is in any case invoked towards the VLR whose identity is contained in the HLR table.
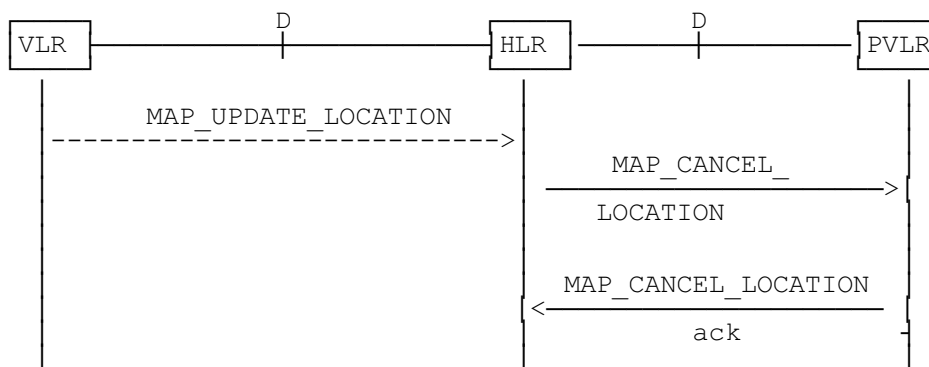
```
  ┌─────┐        D          ┌─────┐        D          ┌──────┐
  │ VLR │────────┼──────────│ HLR │────────┼──────────│ PVLR │
  └─────┘                   └─────┘                   └──────┘
     │                         │                          │
     │    MAP_UPDATE_LOCATION  │                          │
     │-------------------------->                         │
     │                         │                          │
     │                         │      MAP_CANCEL_          │
     │                         │──────────────────────────>
     │                         │      LOCATION            │
     │                         │                          │
     │                         │   MAP_CANCEL_LOCATION    │
     │                         <──────────────────────────│
     │                         │           ack            │
     │                         │                          │
```

NOTE: The service shown in dotted lines indicates the trigger provided by other MAP signalling.

Figure 16.1.2/1: Interface and services for Location Cancellation

16.1.2.2   Detailed procedure in the HLR

The location cancellation process is started by an external process as stated above. The HLR opens a dialogue with the VLR whose identity is contained in the HLR table (MAP_OPEN request without any user specific parameters), sending the MAP_CANCEL_LOCATION request primitive (see figure 16.1.2/2), containing the parameters:

-        IMSI, to identify the subscriber to be deleted from that VLR;
-        LMSI, which is included if available in the HLR.

The HLR then waits for the MAP_OPEN confirmation (see macro Receive_Open_Cnf, subclause 21.1), indicating either:

-        reject of the dialogue (process terminates);

-        reversion to version one (process will be performed according to MAP version one); or

-        dialogue acceptance.

When the VLR accepts the dialogue, it will return a MAP_CANCEL_LOCATION confirmation, containing:

- no parameter, indicating successful outcome of the procedure;

- a user error, provider error or a data error indicating unsuccessful outcome of the procedure.

In case of unsuccessful outcome or if a MAP_P_ABORT indication has been received, the HLR may repeat the MAP_CANCEL_LOCATION request later, where the number of repeat attempts and time in between are HLR operator options, depending on the error returned by the VLR.

16.1.2.3 Detailed procedure in the VLR

Opening of the dialogue is described in the macro Receive_Open_Ind in subclause 21.1, with outcomes:

- reversion to version one procedure;

- procedure termination; or

- dialogue acceptance, with processing as below.

If the VLR process receives a MAP_NOTICE indication, it terminates the dialogue by sending a MAP_CLOSE request.

If the VLR process receives a MAP_CANCEL_LOCATION indication from the HLR (see figure 16.1.2/3), the parameters are checked first (macro Check_Indication, see subclause 21.2). In case of parameter problems the appropriate error is sent in the MAP_CANCEL_LOCATION response.

If the MAP_CANCEL_LOCATION indication contains both the IMSI and the LMSI, the VLR checks whether the stored IMSI matches the received IMSI. If it does not, the VLR attempts to process the request using the IMSI received from the HLR to define the subscriber record to be deleted.

Thereafter the VLR checks whether the subscriber identity provided is known in the VLR:

- if so, the data of the subscriber are deleted from VLR table and a MAP_CANCEL_LOCATION response is returned without any parameters;

- if not, location cancellation is regarded as being successful, too, and the MAP_CANCEL_LOCATION response is returned without any parameters.

In either case, after sending the MAP_CANCEL_LOCATION response the VLR process releases any TMSI which may be associated with the IMSI of the subscriber, terminates the dialogue (MAP_CLOSE with Release Method Normal Release) and returns to the idle state.

Figure 16.1.2/2: Process Cancel_Location_HLR
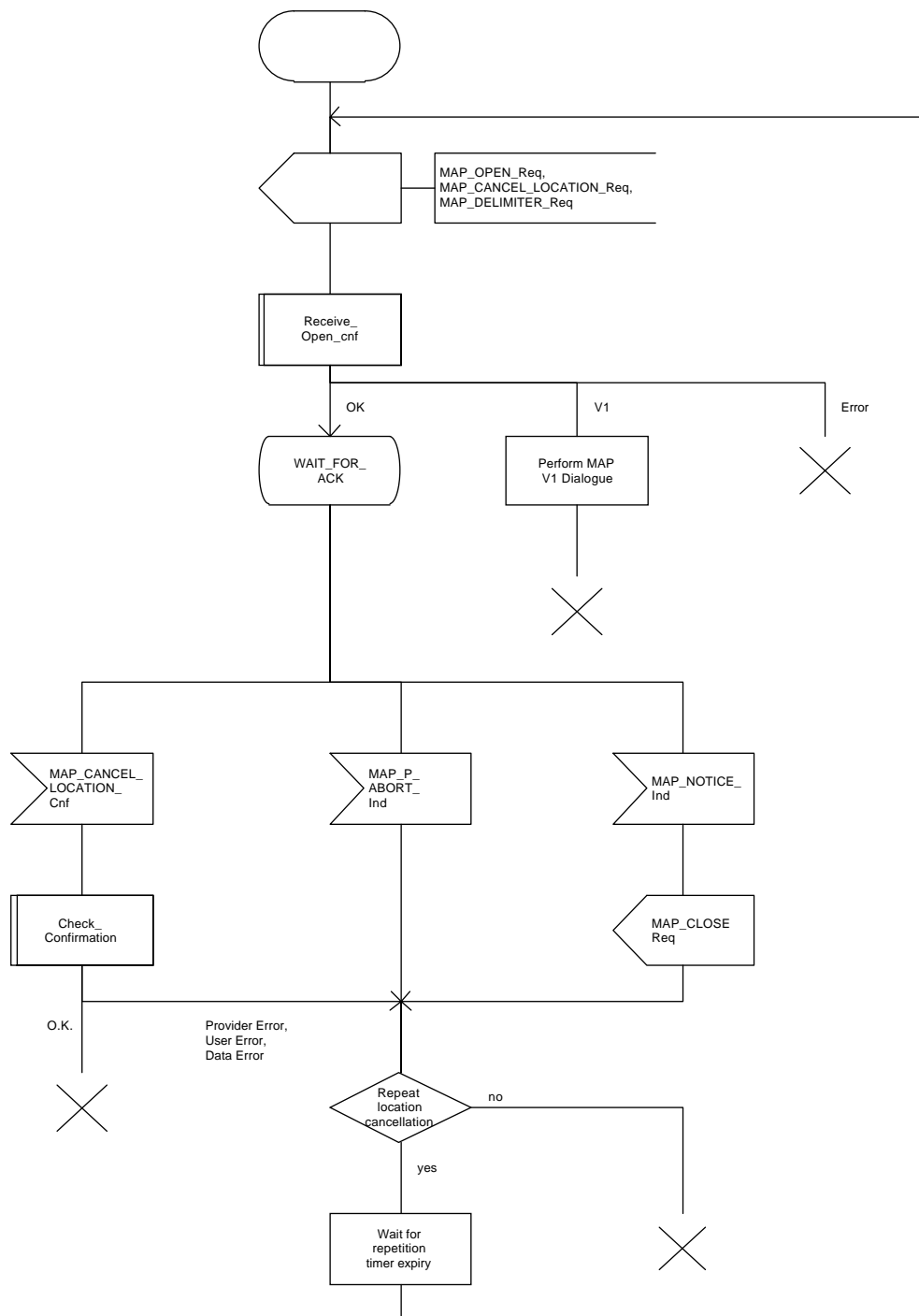
Figure 16.1.2/3: Process Cancel_Location_VLR
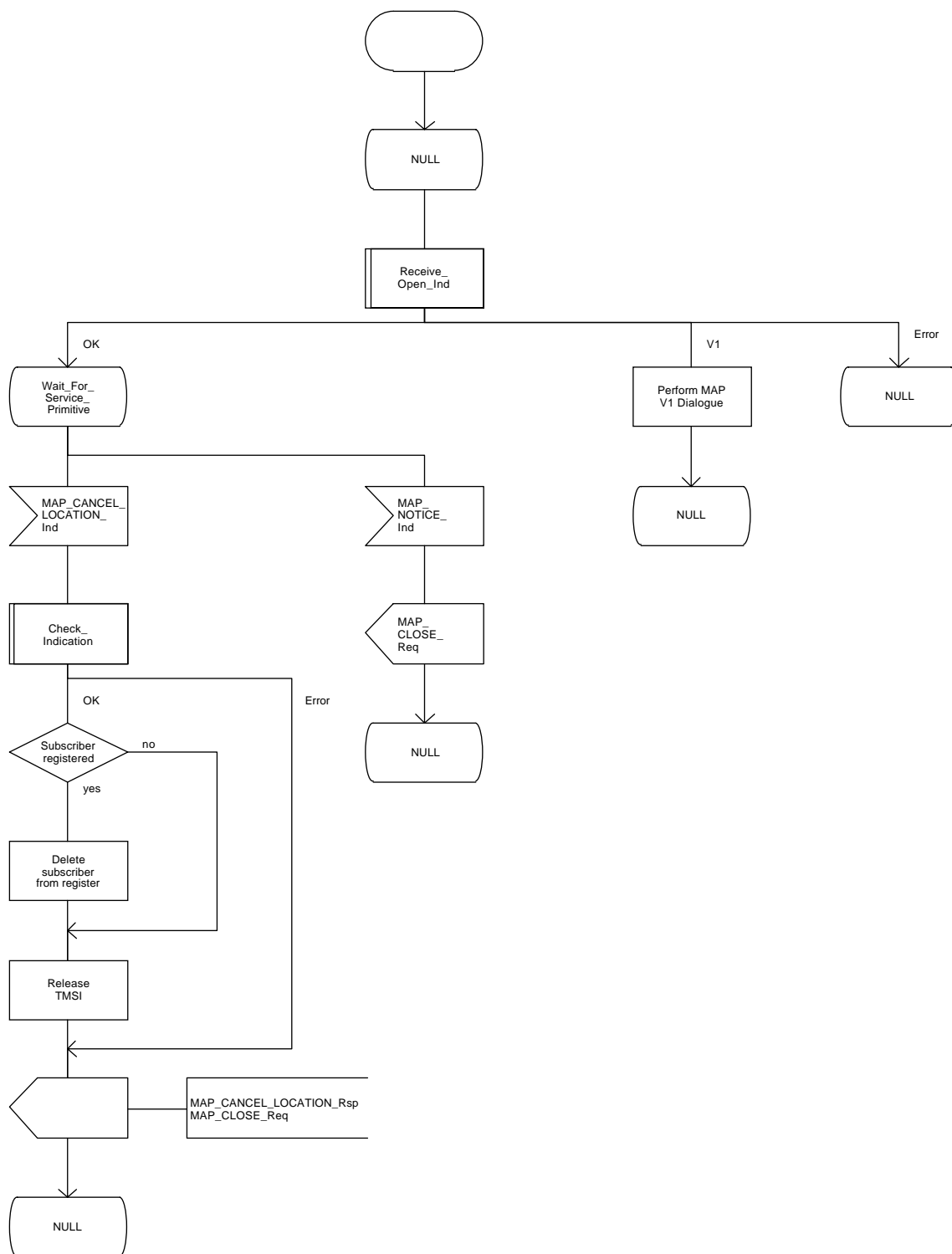
16.1.3 Detach IMSI

16.1.3.1 General

On receipt of an A_LU_REQUEST (DETACH IMSI) indication from the radio interface this procedure invokes the MAP_DETACH_IMSI service described in subclause 6.1.5 in order to inform the visitor location register that a subscriber is no longer reachable (see figure 16.1.3/1), e.g. due to switched off station. This information is used by the VLR to reject mobile terminating calls or short messages without sending page messages on the radio path. The service is unconfirmed as it is likely that the MS is switched off before receiving a confirmation.

The detach IMSI feature is optional for the network operator. The MS is informed by the network whether detach IMSI is to be used or not.

```
                                  A                        B
   +----+        +----+     |  +-----+                 |  +-----+
   | MS |--------| BS |-----|--| MSC |-----------------|--| VLR |
   +----+        +----+     |  +-----+                 |  +-----+
     |                         |                          |
     |      A_LU_Request       |                          |
     |------------------------>|                          |
     |      (DETACH IMSI)       |      MAP_DETACH_IMSI    |
     |                         |------------------------->|
     |                         |                          |
```
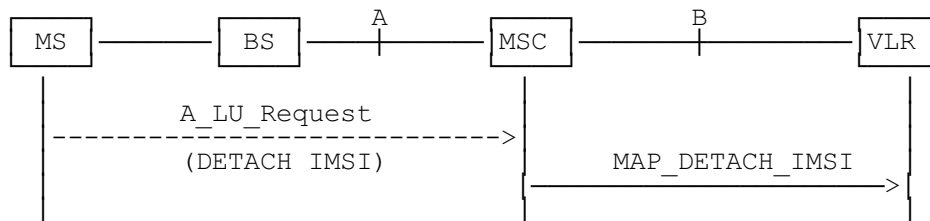
NOTE: The service shown in dotted lines indicates the trigger provided by the radio interface (see GSM 09.10).

Figure 16.1.3/1: Interface and services for MAP_DETACH_IMSI

16.1.3.2 Detailed procedure in the MSC

The MAP_DETACH_IMSI service is invoked by the MSC when receiving an A_LU_Request (DETACH IMSI) for a subscriber (see figure 16.1.3/2).

The MSC will open the dialogue to the VLR with a MAP_OPEN request containing no user specific parameters. The MAP_DETACH_IMSI request will contain the following parameter received from the radio side (for the mapping see GSM 09.10):

-    Subscriber Id, being either a TMSI or an IMSI.

The MSC then waits for the MAP_OPEN confirmation (see macro Receive_Open_Cnf, subclause 21.1), indicating either:

-    reject of dialogue (process terminates);

-    reversion to version one (process will be repeated according to MAP version one); or

-    dialogue acceptance.

Thereafter, the dialogue is terminated locally by the MSC (MAP_CLOSE request with Release Method Prearranged End).

16.1.3.3 Detailed procedure in the VLR

When the VLR receives a MAP_DETACH_IMSI indication (see figure 16.1.3/3), it first checks the indication data (macro Check_Indication, see subclause 21.2). Thereafter it is checked whether the subscriber is known:

- if the subscriber is unknown the VLR ignores the indication;

- if the subscriber is known in the VLR, the IMSI detached flag is set.

The VLR process will terminate the dialogue locally (MAP_CLOSE request with Release Method Prearranged End).

Figure 16.1.3/2: Process Detach_IMSI_MSC

Figure 16.1.3/2:
Detach IMSI in MSC
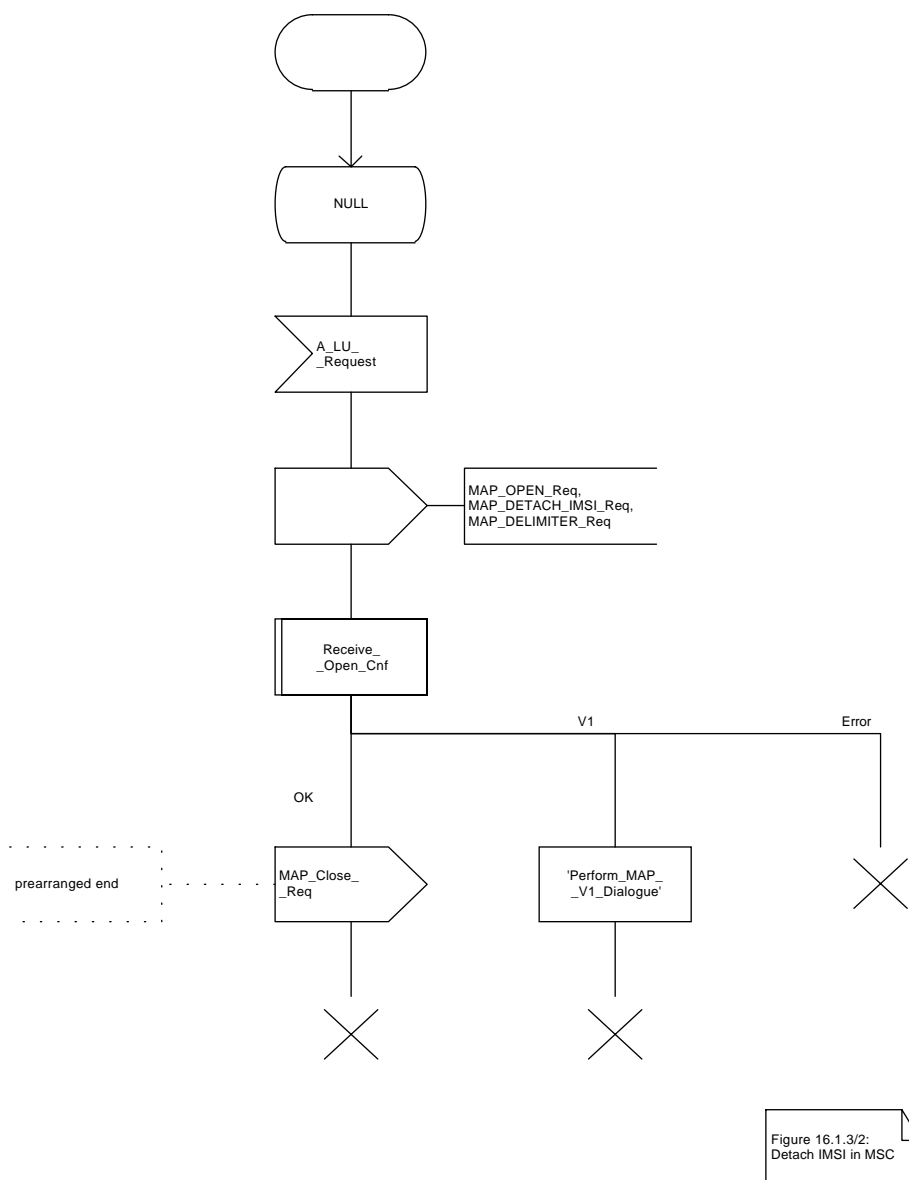
Figure 16.1.3/3: Process Detach_IMSI_VLR
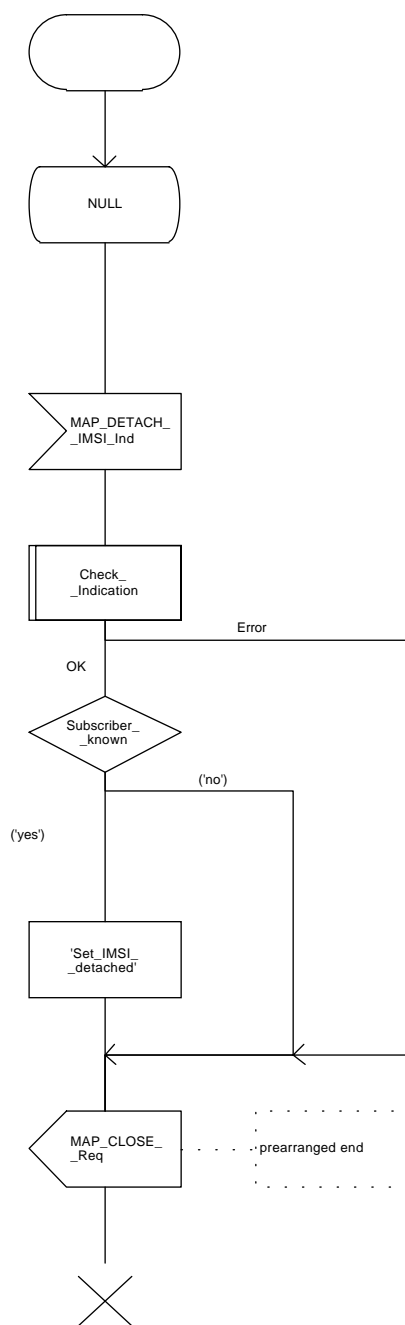
Figure 16.1.3/3:
Detach IMSI in VLR

16.1.4 Purge MS

16.1.4.1 General

When the VLR receives an indication on the O&M interface that the MS record is to be purged (either because of administrative action or because the MS has been inactive for an extended period), this procedure invokes the MAP_PURGE_MS service described in subclause 6.1.6 to request the HLR to set the "MS purged" flag for the MS so that any request for routing information for a mobile terminated call or a mobile terminated short message will be treated as if the MS is not reachable. The message flow is shown in figure 16.1.4/1.

It is optional for the network operator to delete MS records from the VLR, but if the option is used the VLR shall notify the HLR when a record has been deleted.

The O&M process in the VLR must ensure that during the MS purging procedure any other attempt to access the MS record is blocked, to maintain consistency of data.

```
  ┌─────┐                                      ┌─────┐
  │ VLR │──────────────────────────────────────│ HLR │
  └─────┘                                      └─────┘
     │                 MAP_PURGE_MS                │
     │────────────────────────────────────────>   │
     │                                             │
     │               MAP_PURGE_MS_ack              │
     │   <────────────────────────────────────────│
     │                                             │
```
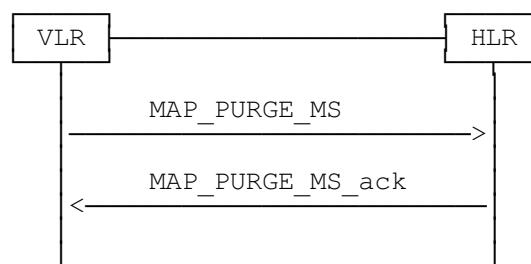
Figure 16.1.4/1: Interface and services for MAP_PURGE_MS

16.1.4.2 Detailed procedure in the VLR

When the VLR receives an indication from O&M that an MS record is to be purged, it invokes the MAP_PURGE_MS service (see figure 16.1.4/2).

The VLR opens the dialogue to the HLR with a MAP_OPEN request containing no user specific parameters. The MAP_PURGE_MS request contains the IMSI of the MS which is to be purged and the VLR number.

The VLR then waits for the MAP_OPEN confirmation (see macro Receive_Open_Cnf, subclause 21.1), indicating one of:

- rejection of the dialogue (process terminates);

- reversion to version one (process terminates);

- dialogue acceptance.

If the HLR accepts the dialogue it returns a MAP_PURGE_MS confirmation, containing no parameter, indicating successful outcome of the procedure.

If a MAP_PURGE_MS confirmation containing a provider error, data error or user error, or a MAP_P_ABORT, MAP_NOTICE or premature MAP_CLOSE indication, has been received, the failure is reported to the O&M interface. Successful outcome of the procedure leads to deletion of the subscriber data and freezing of the TMSI, and is reported to the O&M interface.

16.1.4.3 Detailed procedure in the HLR

Opening of the dialogue is described in the macro Receive_Open_Ind in subclause 21.1. The possible outcomes are:

- termination of the procedure if the AC indicates a version 1 dialogue, as this procedure is defined only for version 2;

- termination of the procedure if there is an error;

- dialogue acceptance, in which case the procedure is as described below.

If the HLR receives a MAP_NOTICE indication, it terminates the dialogue by sending a MAP_CLOSE request.

If the HLR receives a MAP_PURGE_MS indication (see figure 16.1.4/3), it first checks the indication data (macro Check_Indication, see subclause 21.2). If there is a parameter error the HLR terminates the dialogue by sending a MAP_CLOSE request (local termination). If there is no parameter error the HLR then checks whether the subscriber is known.

- if the subscriber is unknown, the HLR reports an error to the O&M interface, and terminates the dialogue by sending a MAP_CLOSE request (local termination);

- if the subscriber is known, the HLR checks whether the purging notification came from the VLR where the MS was last registered:

  - if the received VLR number and the stored VLR number match, the HLR sets the "MS purged" flag for the subscriber and sends a MAP_PURGE_MS response containing an empty result to indicate successful outcome;

  - if the received VLR number and the stored VLR number do not match, the HLR sends a MAP_PURGE_MS response containing an empty result to indicate successful outcome. Since the MS is known by the HLR to be in a different VLR area, it is not appropriate to block mobile terminated calls or short messages to the MS, but the VLR which initiated the purging procedure can safely purge its record for the MS.

In either cases of successful termination the HLR terminates the dialogue by sending a MAP_CLOSE request.

Figure 16.1.4/2: Process Purge_MS_VLR
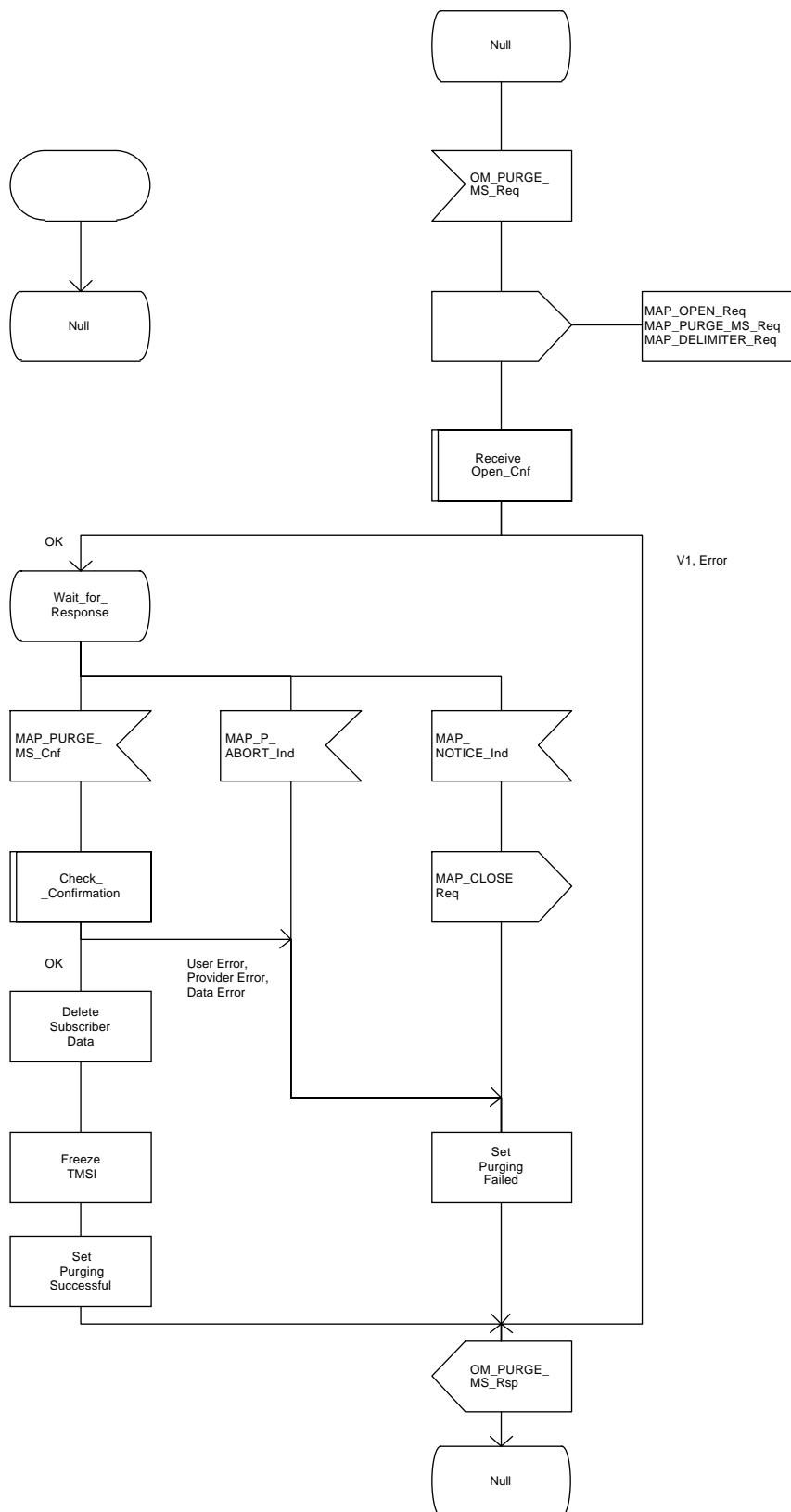
Figure 16.1.4/3: Process Purge_MS_HLR
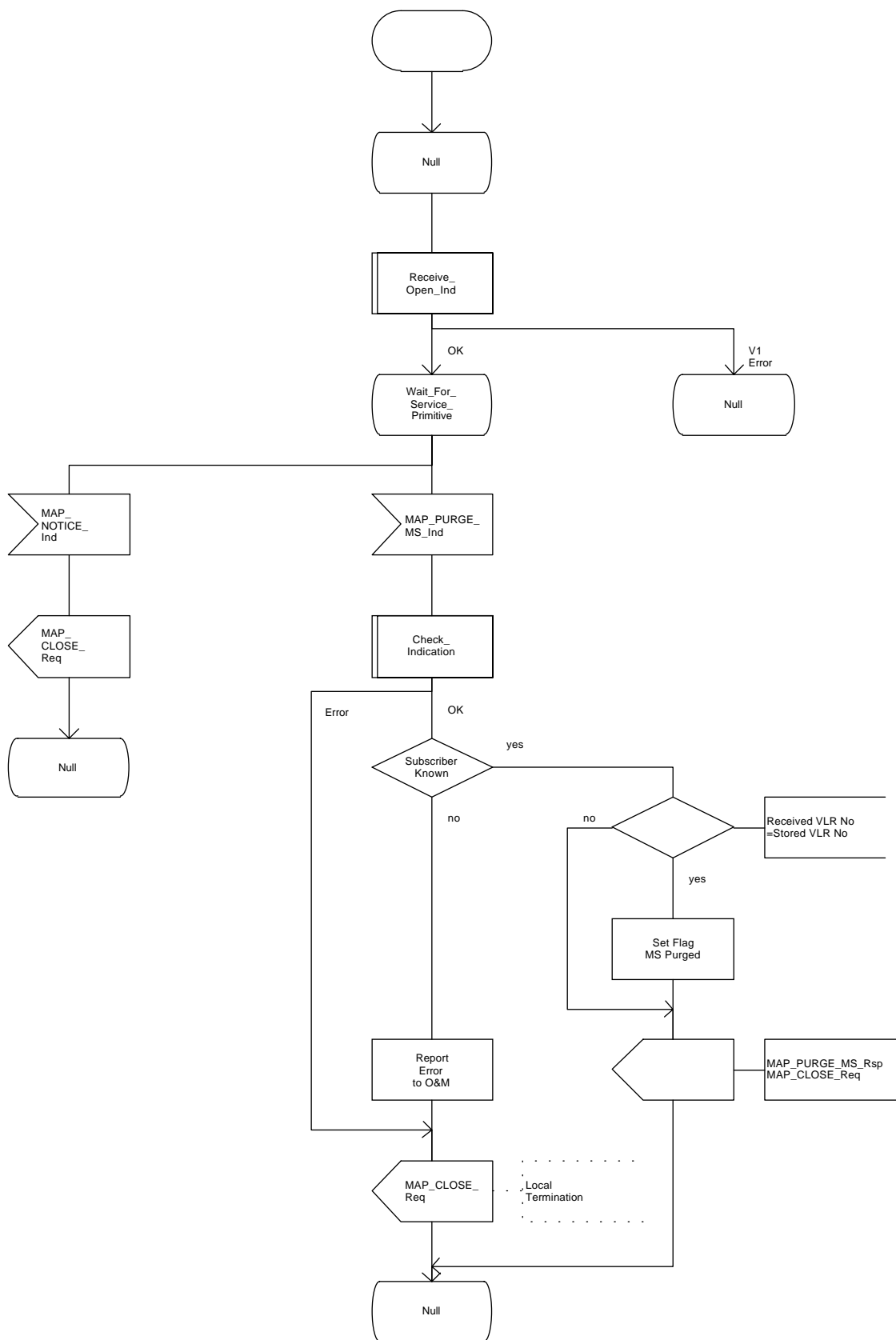
16.2 Handover procedure

16.2.1 General

The handover between different MSCs is called Inter-MSC handover. The interfaces involved for Inter-MSC handover are shown in figure 16.2/1. Following two Inter-MSC handover procedures apply:

1) Basic Inter-MSC handover:

   The call is handed over from the controlling MSC, called MSC-A to another MSC, called MSC-B (figure 16.2/1a).

   Figure 16.2/2 shows a successful handover between MSC-A and MSC-B including a request for handover number allocation by MSC-B to VLR-B.

2) Subsequent Inter-MSC handover:

   After the call has been handed over from MSC-A to MSC-B, a handover to either MSC-A (figure 16.2/1a) or to a third MSC (MSC-B') (figure 16.2/1b) is necessary in order to continue the connection.

   Figure 16.2/3 shows a successful subsequent handover.

```
            E             B
MSC-A -----+----- MSC-B -----+----- VLR-B
```
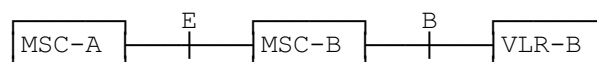
a) Basic handover procedure MSC-A to MSC-B and subsequent handover procedure MSC-B to MSC-A.

```
            E             B
MSC-A -----+----- MSC-B -----+----- VLR-B
      |
      |     E             B
      +---------- MSC-B' -----+----- VLR-B'
```
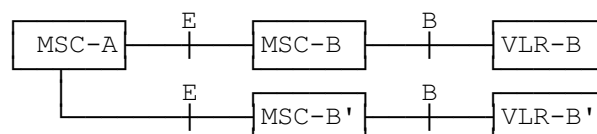
b) Subsequent handover procedure MSC-B to MSC-B'.

Figure 16.2/1: Interface structure for handover

The MAP handover procedures achieve the functionality required to set up an MSC-MSC dialogue, to optionally allocate a handover number and to transport BSSAP messages.

The transported BSSAP messages are controlled and handled by the Handover Control Application in the MSCs. This information will be transparent to the MAP protocol. If the MSC receives via the MAP protocol BSSAP messages, this information will be forwarded to the Handover Control Application (shown in the handover SDL diagrams with the internal HO_CA signalling, it is an internal process in the MSC) and vice versa if the Handover Control Application requires the sending of BSSAP messages via the MAP protocol.

For detailed interworking between the A-interface and MAP procedures, see GSM 03.09 and GSM 09.10.
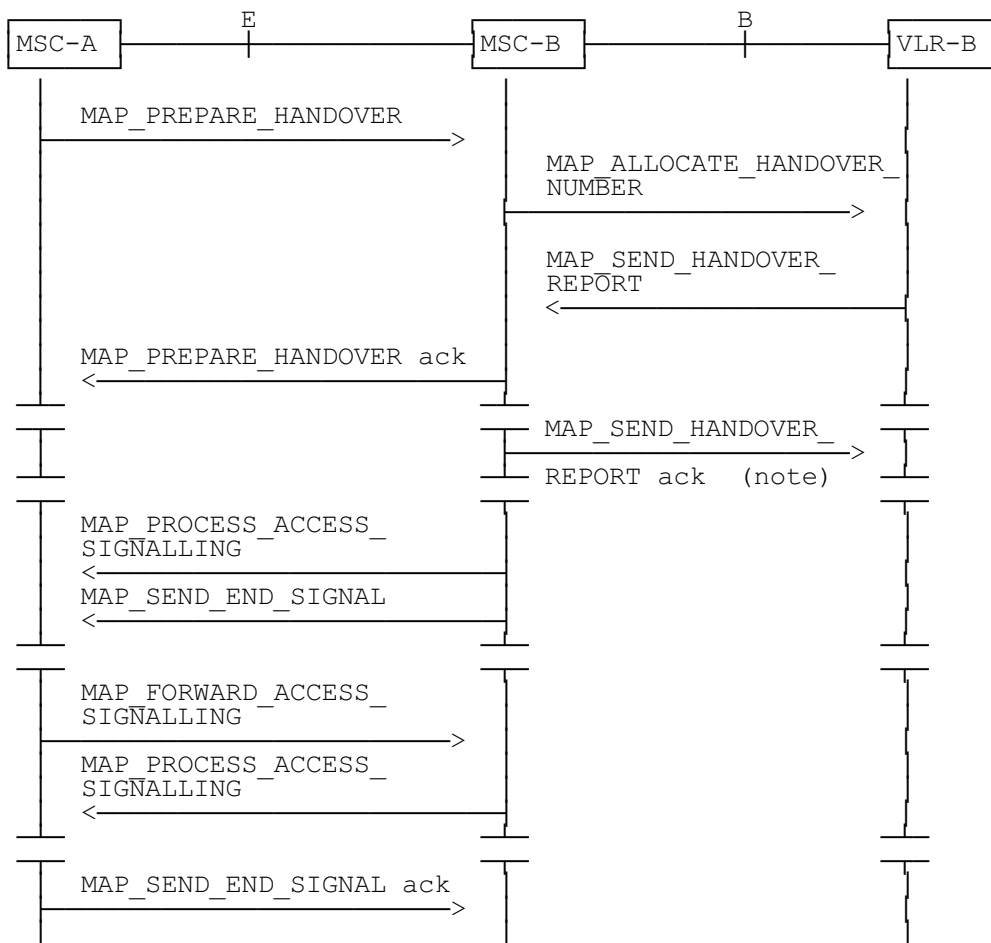
```
 ┌───────┐      E        ┌───────┐      B        ┌───────┐
 │ MSC-A │───────┼───────│ MSC-B │───────┼───────│ VLR-B │
 └───────┘               └───────┘               └───────┘
     │                       │                       │
     │   MAP_PREPARE_HANDOVER │                       │
     │──────────────────────>│                       │
     │                       │  MAP_ALLOCATE_HANDOVER_│
     │                       │  NUMBER                │
     │                       │───────────────────────>│
     │                       │                        │
     │                       │  MAP_SEND_HANDOVER_     │
     │                       │  REPORT                 │
     │                       │<───────────────────────│
     │ MAP_PREPARE_HANDOVER ack                       │
     │<──────────────────────│                        │
     ├───────┤               ├───────┤                ├───────┤
     ├───────┤               ├───────┤ MAP_SEND_HANDOVER_     │
     │       │               │───────────────────────>│       │
     ├───────┤               ├───────┤ REPORT ack  (note)     │
     │ MAP_PROCESS_ACCESS_    ├───────┤                ├───────┤
     │ SIGNALLING             │                        │
     │<──────────────────────│                        │
     │ MAP_SEND_END_SIGNAL    │                        │
     │<──────────────────────│                        │
     ├───────┤               ├───────┤                ├───────┤
     │ MAP_FORWARD_ACCESS_    │                        │
     │ SIGNALLING             │                        │
     │──────────────────────>│                        │
     │ MAP_PROCESS_ACCESS_    │                        │
     │ SIGNALLING             │                        │
     │<──────────────────────│                        │
     ├───────┤               ├───────┤                ├───────┤
     │ MAP_SEND_END_SIGNAL ack│                        │
     │──────────────────────>│                        │
     │                       │                        │
```

NOTE: This can be sent at any time after the connection between MSC-A and MSC-B is established.

Figure 16.2/2: Example of a successful basic handover procedure to MSC-B
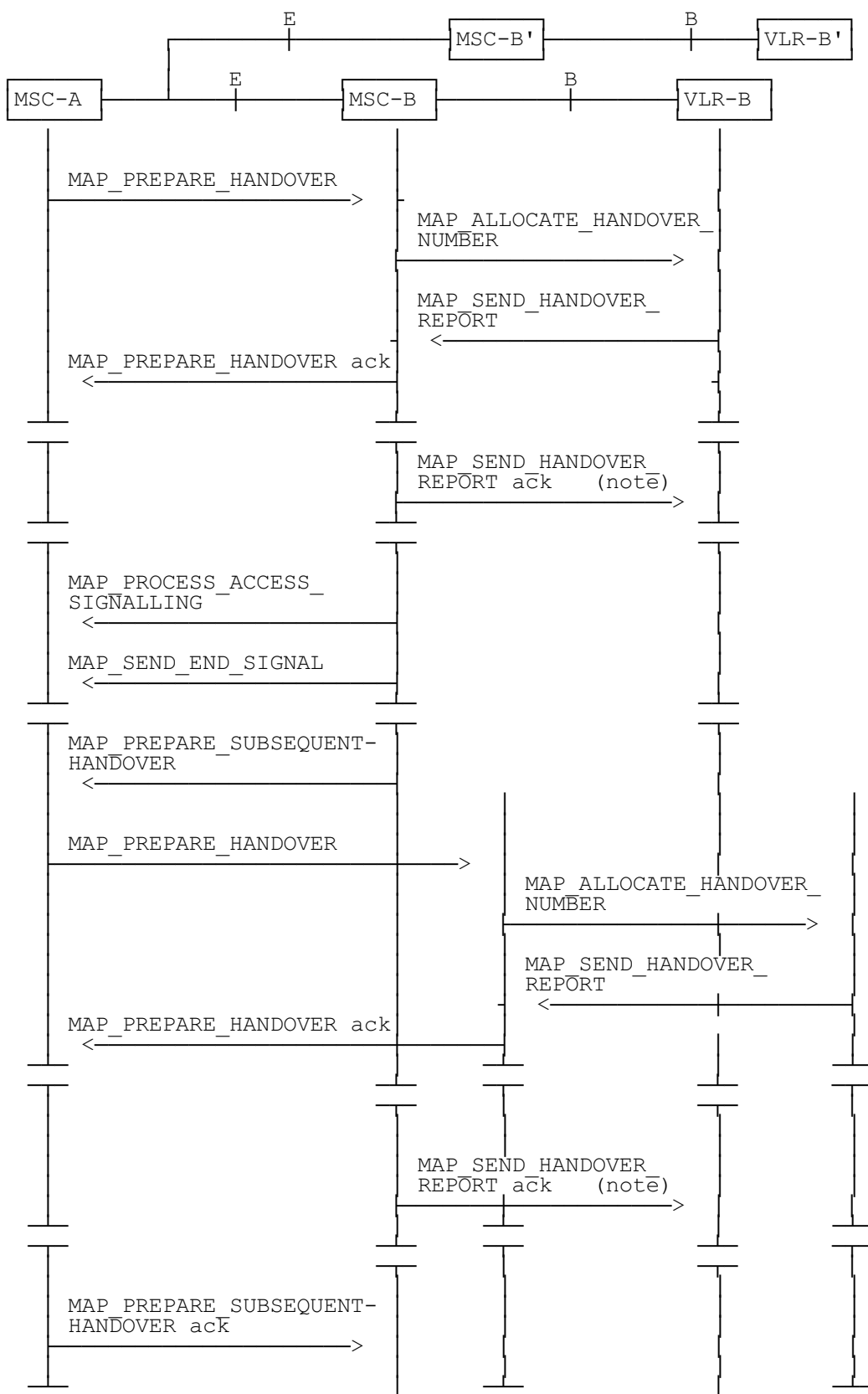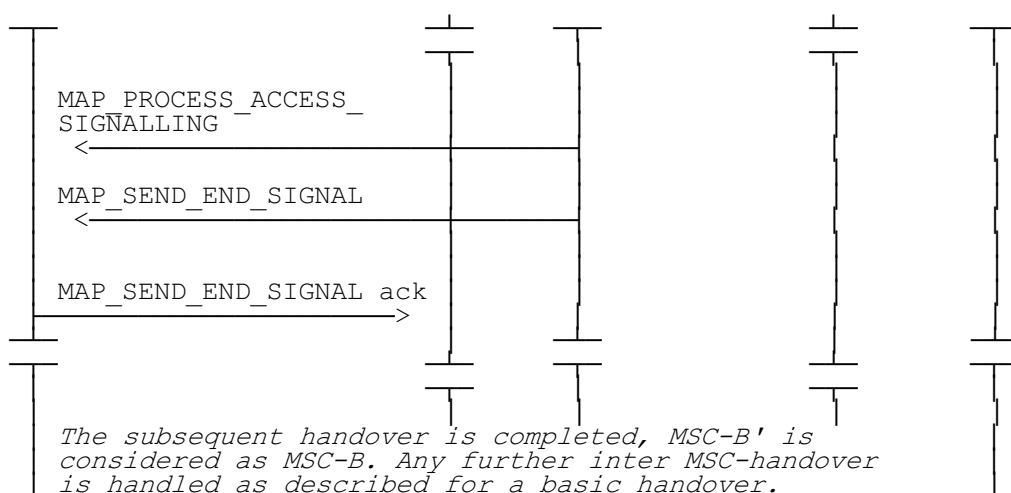
```
              MAP_PROCESS_ACCESS_
              SIGNALLING
                 <
              MAP_SEND_END_SIGNAL
                 <
              MAP_SEND_END_SIGNAL ack
                                    >
```

The subsequent handover is completed, MSC-B' is
considered as MSC-B. Any further inter MSC-handover
is handled as described for a basic handover.

NOTE: This can be sent at any time after the connection between MSC-A and MSC-B is established

Figure 16.2/3: Example of a handover towards a third MSC

16.2.2   **Handover procedure in MSC-A**

This subclause describes the handover procedure in MSC-A, including the request for a basic handover to another MSC (MSC-B), subsequent handover to a third MSC (MSC-B') or back to the controlling MSC (MSC-A).

16.2.2.1   **Basic handover**

When MSC-A has decided that a call has to be handed over to MSC-B, the Handover Control Application in MSC-A requests the MAP application to initiate the MAP_PREPARE_HANDOVER request to MSC-B.

MSC-A opens the dialogue to MSC-B with a MAP_OPEN request containing no user specific parameters and sends a MAP_PREPARE_HANDOVER request. This request may optionally contain an indication that a handover number allocation is not required, targetCellId, for compatibility reasons, and all information required by MSC-B to allocate the necessary radio resources.

If MSC-B accepts the dialogue, it returns a MAP_PREPARE_HANDOVER confirmation containing a handover number, unless the request has included the HO-NumberNotRequired parameter, and BSSAP information which is forwarded to and handled by the Handover Control Application in MSC-A.

Optionally MSC-A can receive, after a MAP_PREPARE_HANDOVER confirmation, a MAP_PROCESS_ACCESS_SIGNALLING indication containing BSSAP information.

When the connection has been established between the MS and MSC-B, MSC-A will be informed by a MAP_SEND_END_SIGNAL indication.

When MSC-A wants to clear the connection with BSS-B, an indication from the Handover Control Application is received in the Map Application to send the MAP_SEND_END-SIGNAL response to MSC-B to close the MAP dialogue.

MSC-A may abort the handover procedure at any time (e.g. if the call is cleared).

16.2.2.2   **Handling of access signalling**

If required, the Handover Control Application in MSC-A requests the MAP application to invoke the MAP_FORWARD_ACCESS_SIGNALLING request containing the information to be transferred to the A-interface of MSC-B (e.g. call control information).

MAP_FORWARD_ACCESS_SIGNALLING is a non-confirmed service.

MSC-B will then forward the required information to the Handover Control Application. The MAP_FORWARD_ACCESS_SIGNALLING is composed in such a way that the information can be passed transparently to the A-interface for call control and mobility management information. Any response received in MSC-B from the A-interface that should be brought to MSC-A will require a new independent request from the Handover Control Application in MSC-B to MSC-A by invoking a MAP_PROCESS_ACCESS_SIGNALLING request.

16.2.2.3   **Other procedures in stable handover situation**

During a call and after handover, a number of procedures between MSC-A and BSS-B controlled by or reported to MSC-A may be initiated in both directions by invoking a MAP_FORWARD_ACCESS_SIGNALLING request and reception of a MAP_PROCESS_ACCESS_SIGNALLING indication.

16.2.2.4   Subsequent handover

When MSC-A receives a MAP_PREPARE_SUBSEQUENT_HANDOVER request, it will start the procedure of handing the call over to a third MSC (MSC-B'), or back to the controlling MSC (MSC-A). If the new handover procedure towards MSC-B' or MSC-A is successful, the handover control application in MSC-A will request the release of the dialogue towards MSC-B by sending the MAP_SEND_END_SIGNAL confirmation.

16.2.2.5   SDL Diagrams

The SDL diagrams on the following pages describe the user processes in MSC-A for the procedures described in this subclause.

The services used are defined in subclause 6.4.

> NOTE: The message primitives HO_CA_MESSAGE used in the SDL-Diagrams are used to show the internal co-ordination between the MAP application and the Handover Control Application. For a detailed description of the co-ordination between the applications for the handover procedure, see GSM 03.09.

Note that in case of reception of errors from the MSCs (see the Handover error handling macro), the MAP user reports them to the Handover Control Application and does not take any action except in cases explicitly mentioned in the SDL diagrams.

Figure 16.2.2/1 (sheet 1 of 12): Process MSC_A_HO
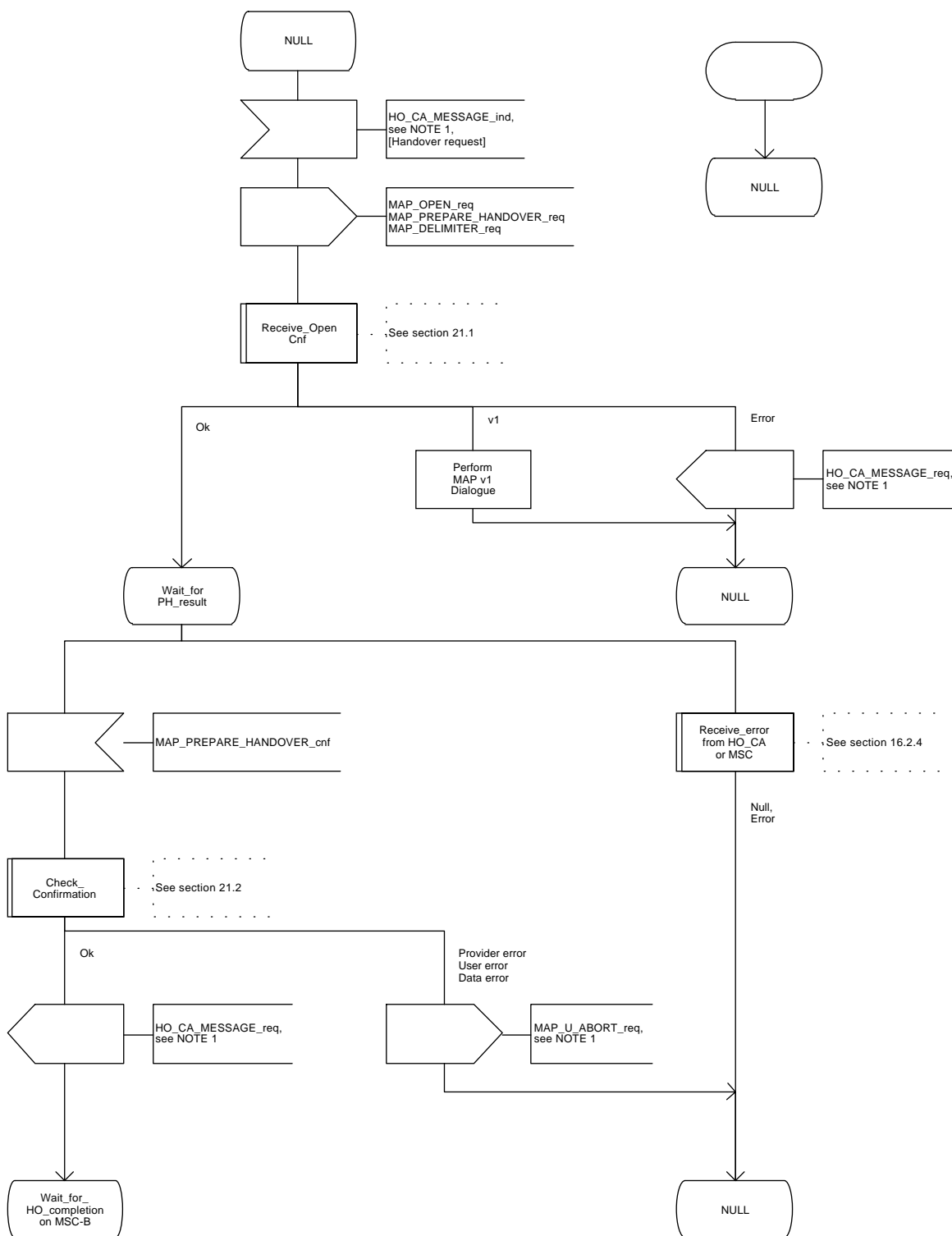
Figure 16.2.2/1 (sheet 2 of 12): Process MSC_A_HO
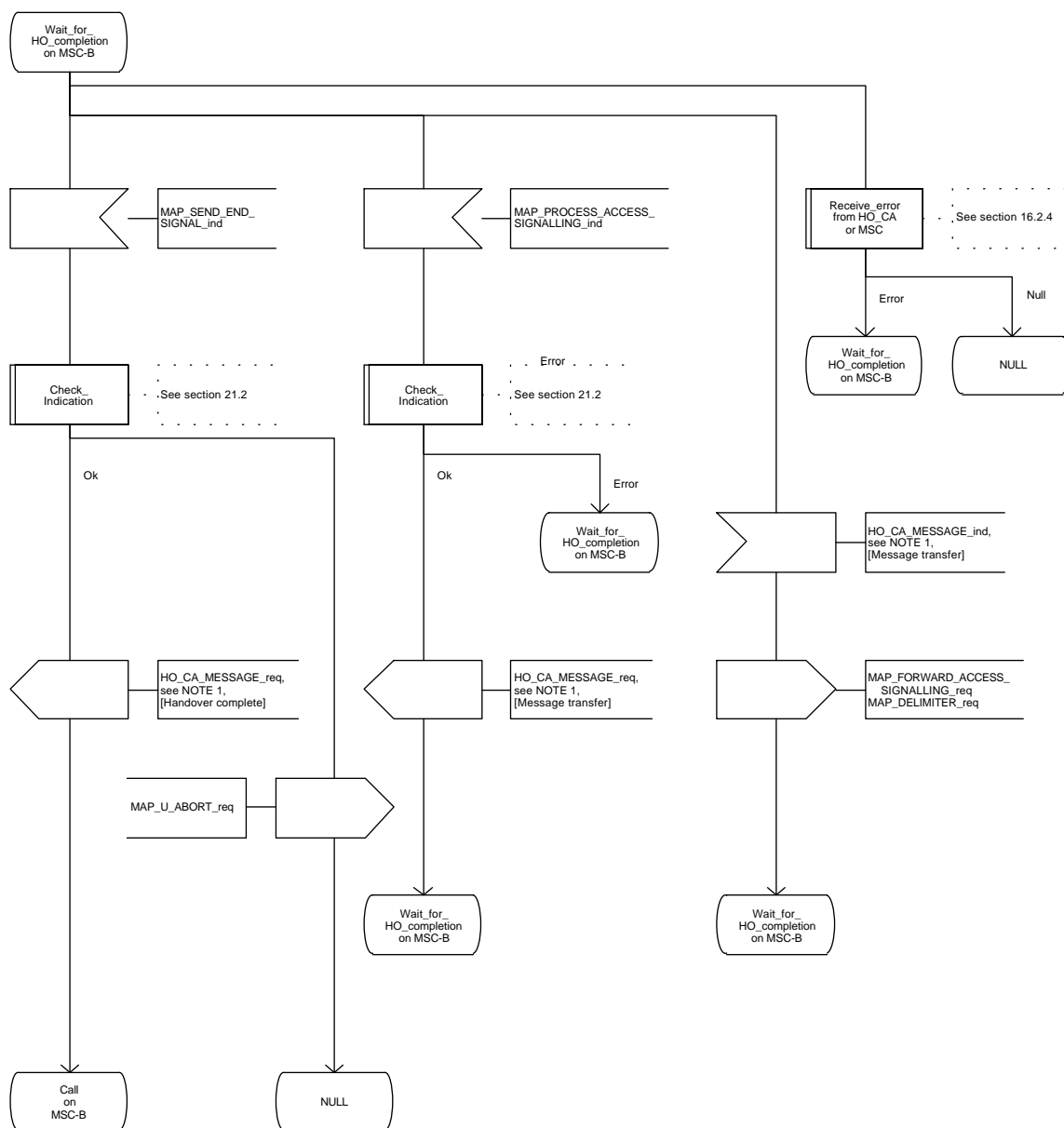
Figure 16.2.2/1 (sheet 3 of 12): Process MSC_A_HO
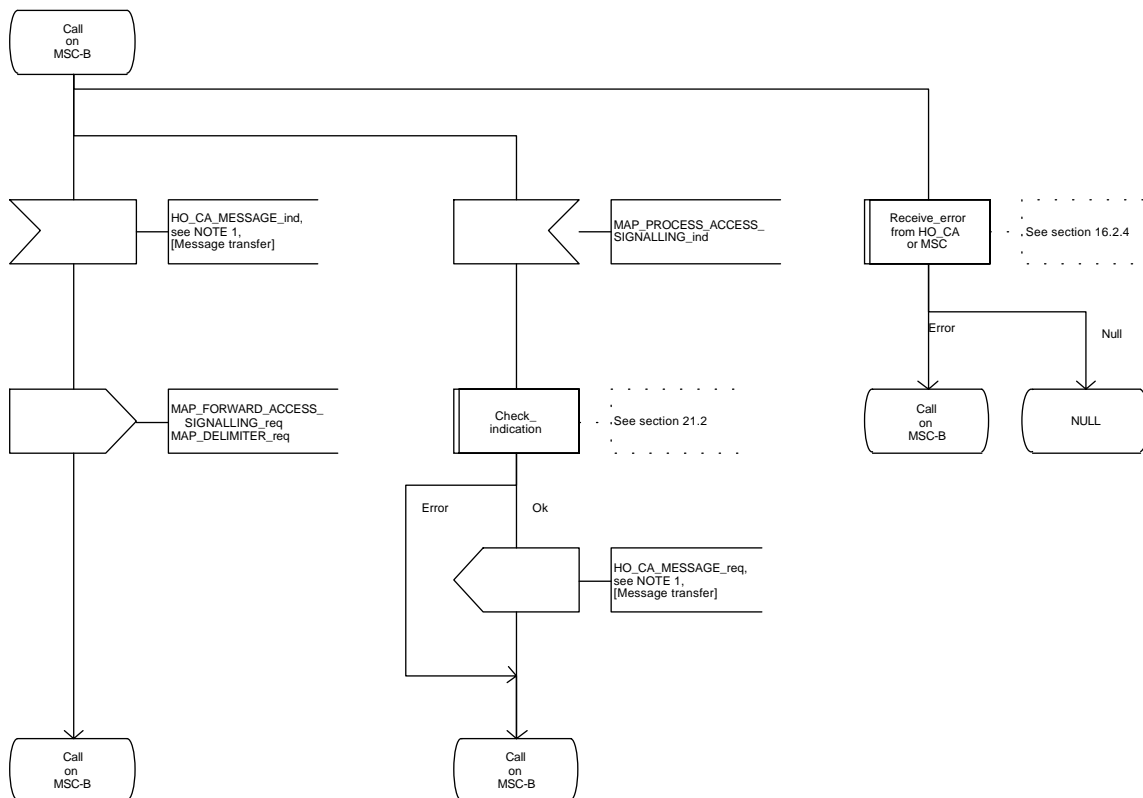
Figure 16.2.2/1 (sheet 4 of 12): Process MSC_A_HO
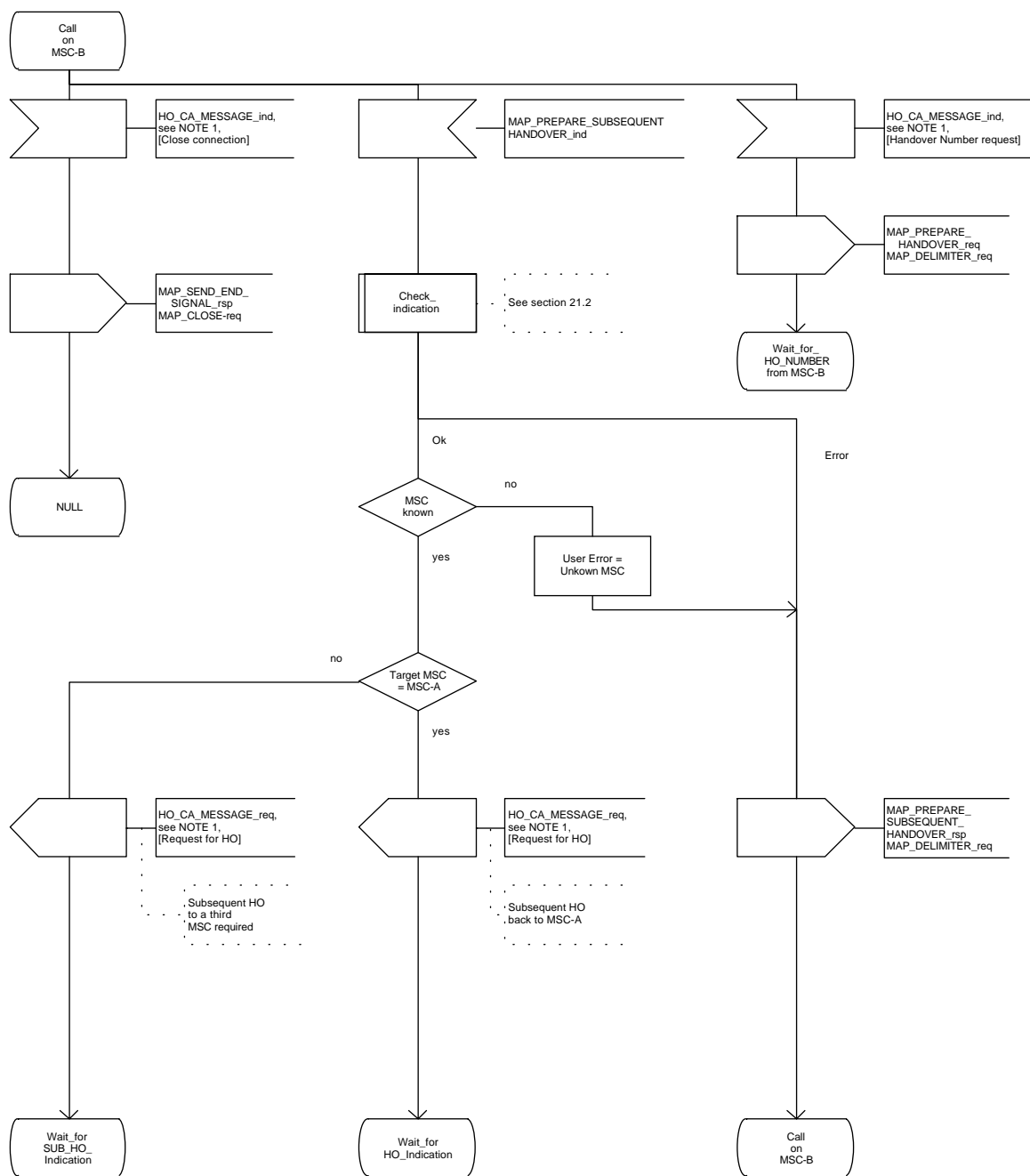
Figure 16.2.2/1 (sheet 5 of 12): Process MSC_A_HO
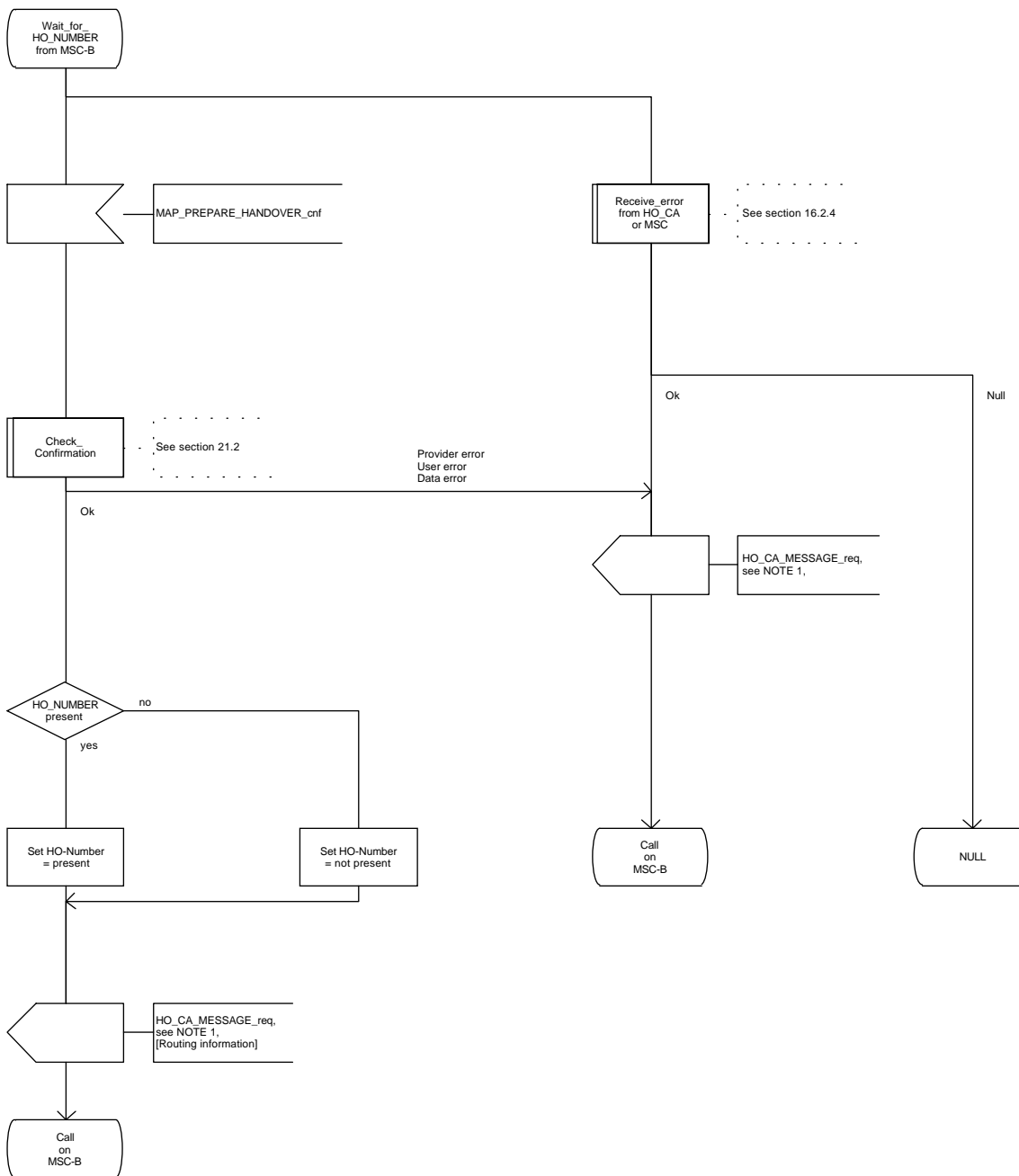
Figure 16.2.2/1 (sheet 6 of 12): Process MSC_A_HO
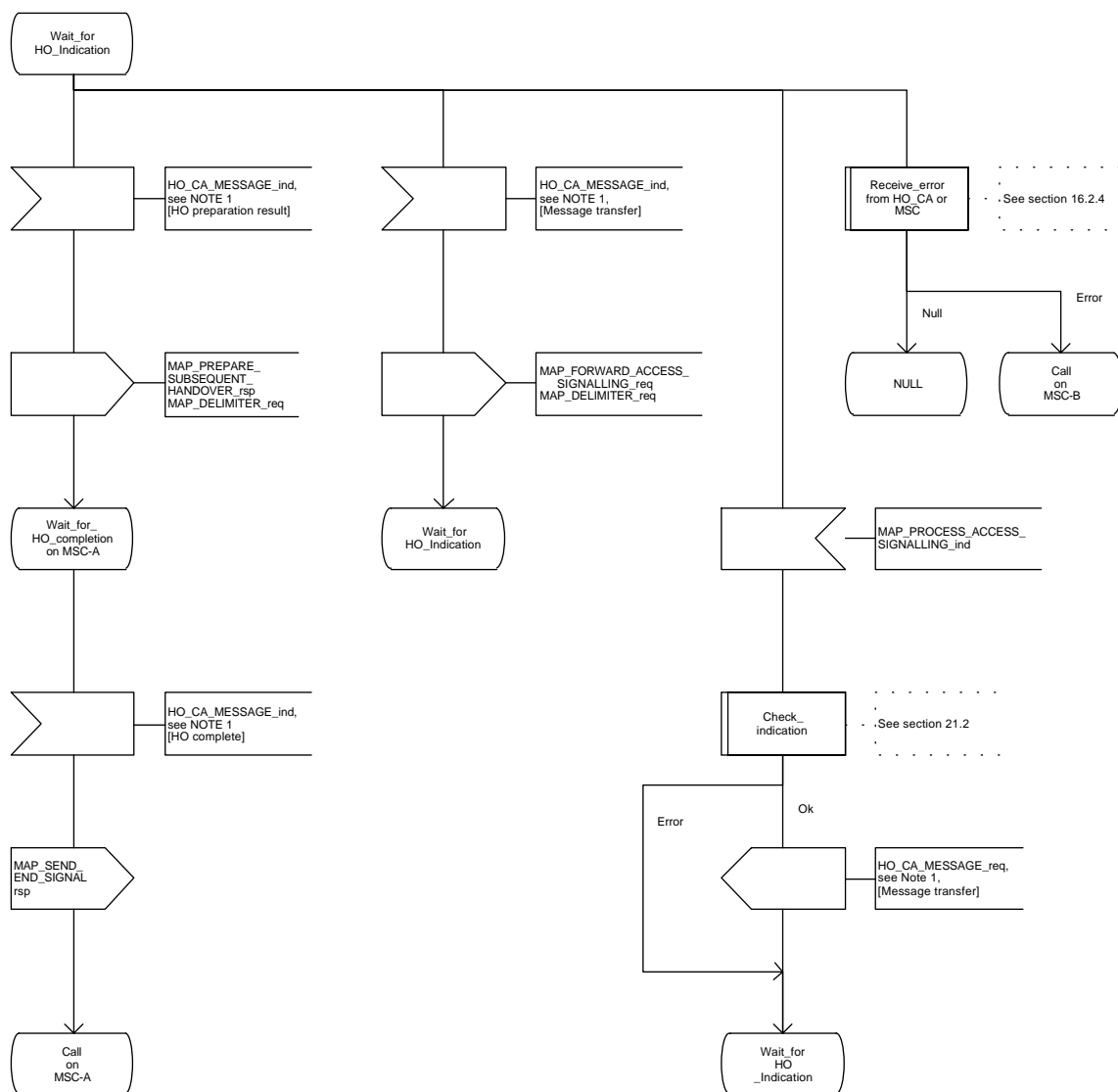
Figure 16.2.2/1 (sheet 7 of 12): Process MSC_A_HO
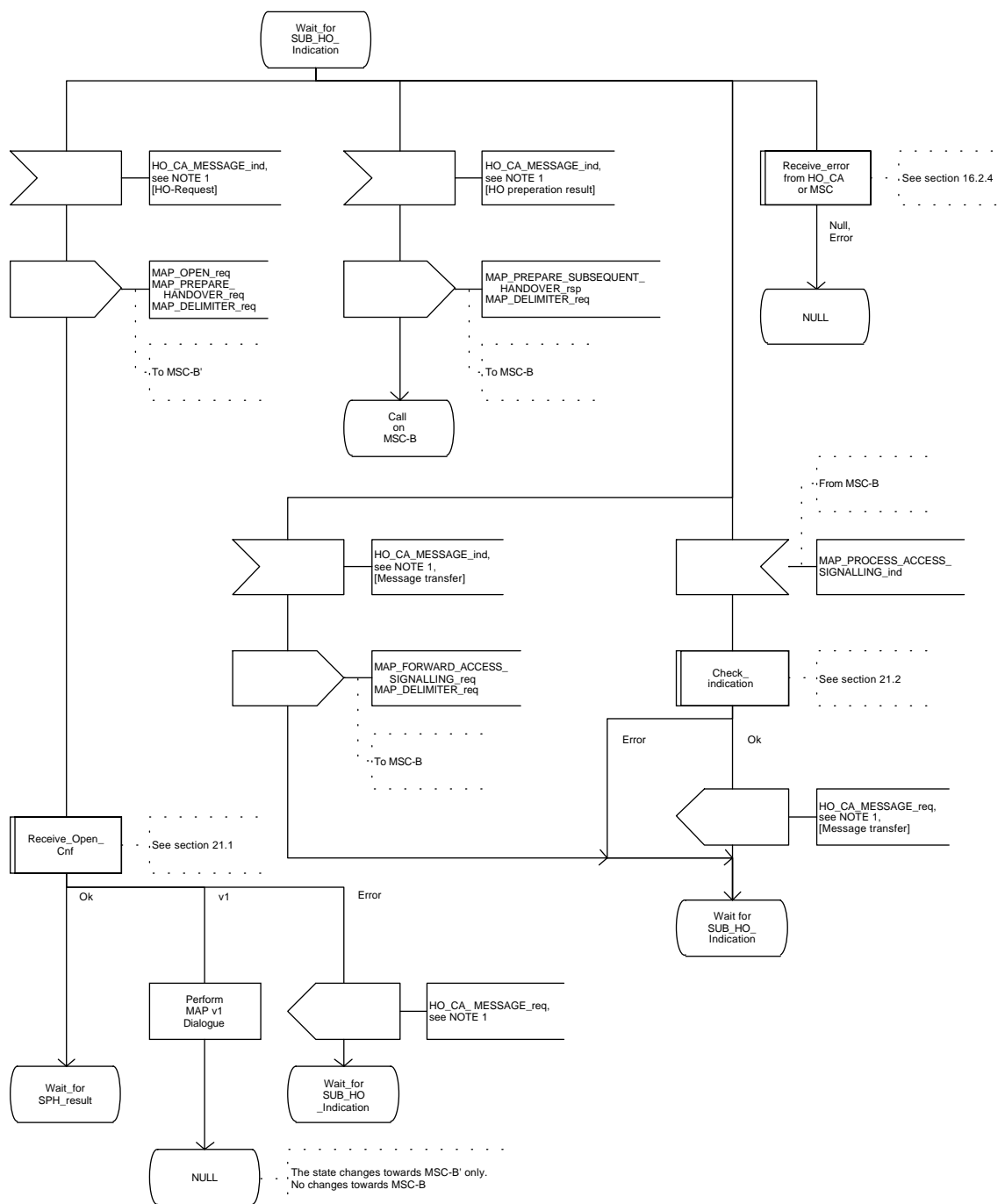
Figure 16.2.2/1 (sheet 8 of 12): Process MSC_A_HO
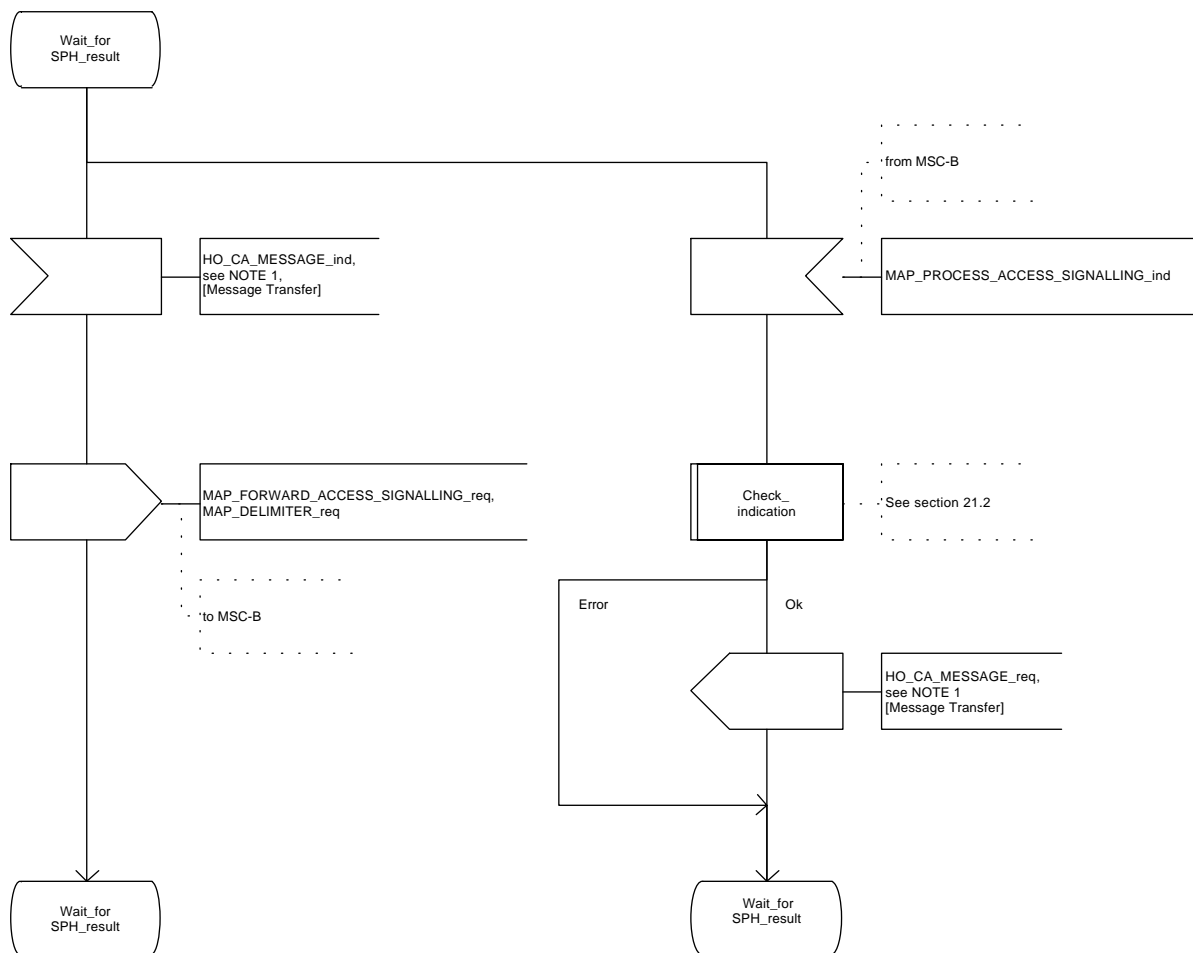
Figure 16.2.2/1 (sheet 9 of 12): Process MSC_A_HO
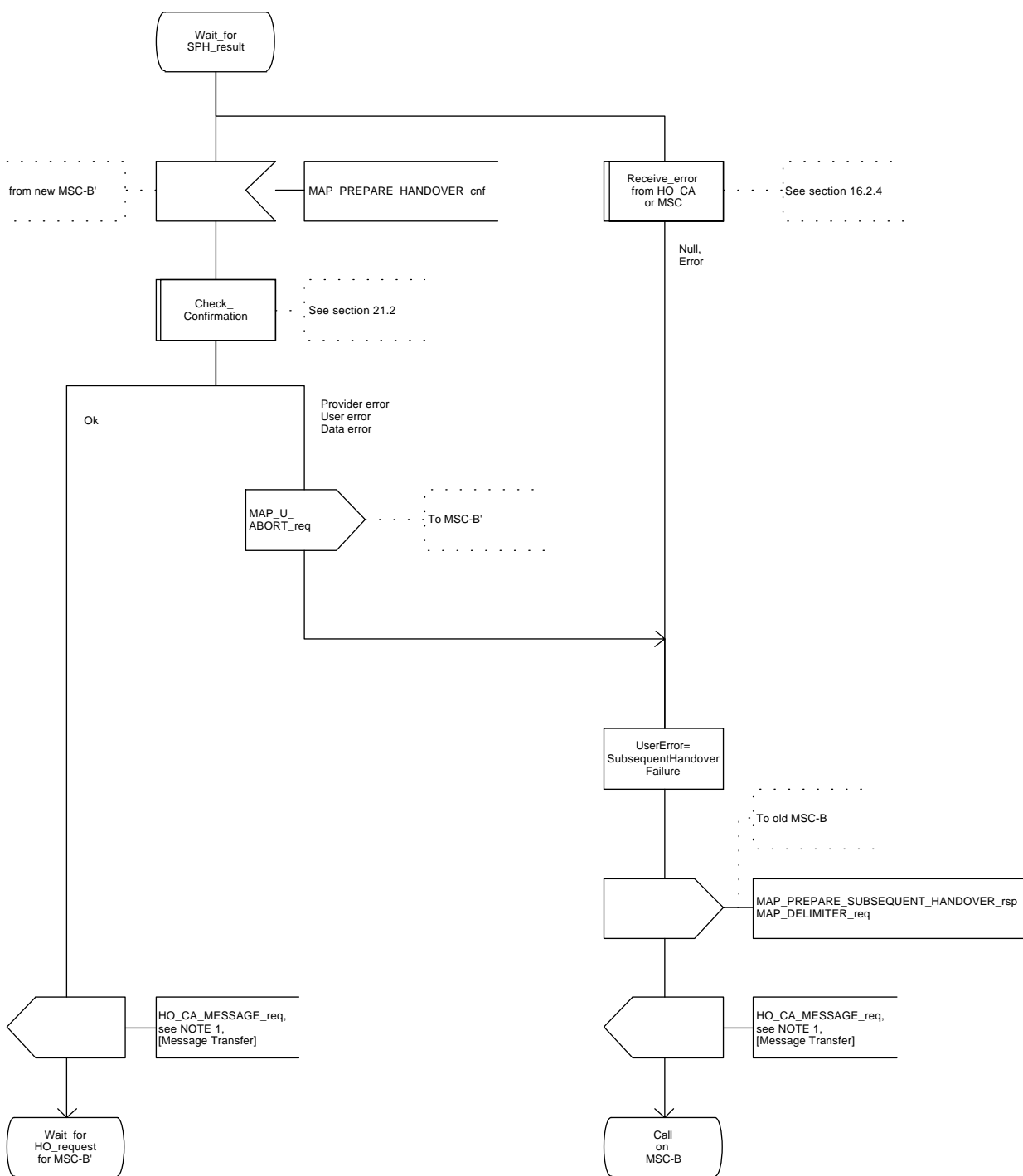
Figure 16.2.2/1 (sheet 10 of 12): Process MSC_A_HO
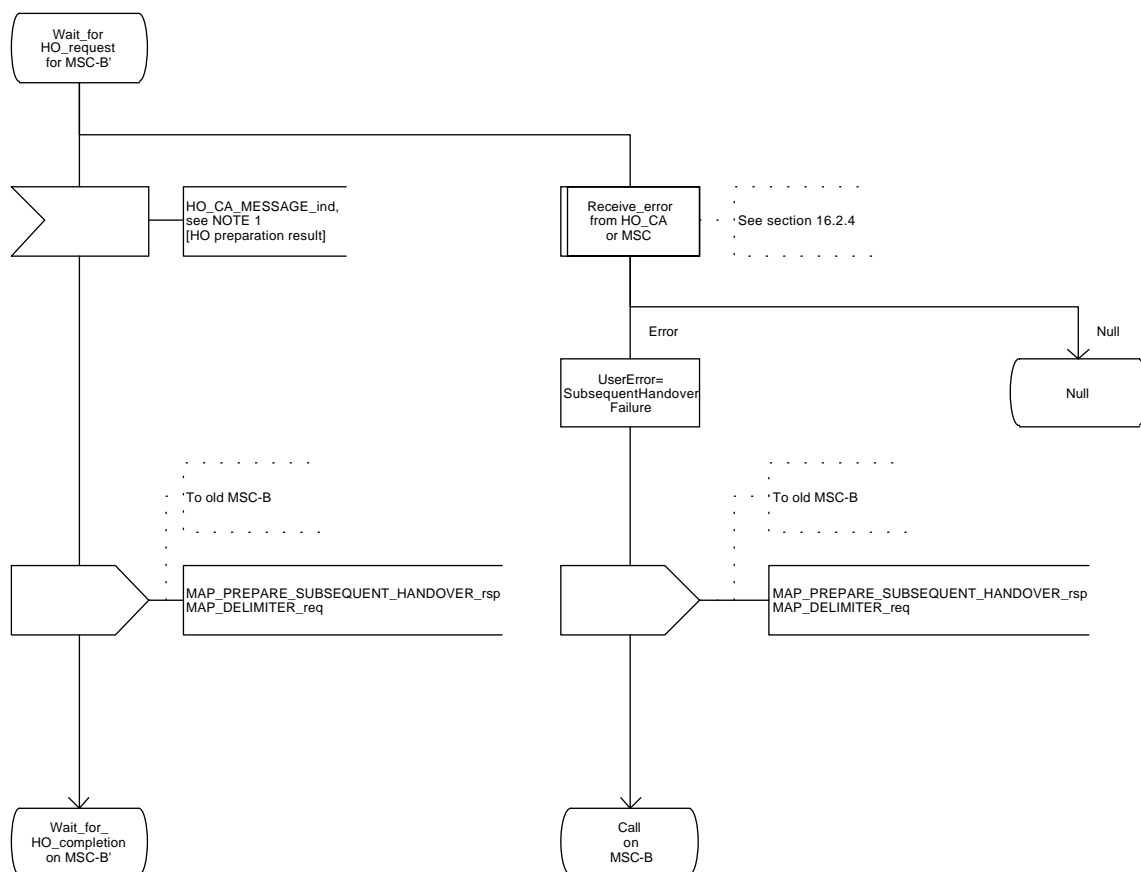
Figure 16.2.2/1 (sheet 11 of 12): Process MSC_A_HO

Wait_for_
HO_completion
on MSC-B'

MAP_SEND_END_SIGNAL_ind

from MSC-B'

Check_
Indication

See section 21.2

OK

Error

HO_CA_MESSAGE_req,
see NOTE 1

MAP_U_
ABORT_req

To MSC-B'

Wait_for_
Call_on
MSC-B'

NULL

MAP_PROCESS_ACCESS_SIGNALLING_ind

Check_
Indication

See section 21.2

Ok

Error

HO_CA_MESSAGE_req,
see NOTE 1
[Message transfer]

Wait_for_
HO_completion
on MSC-B'
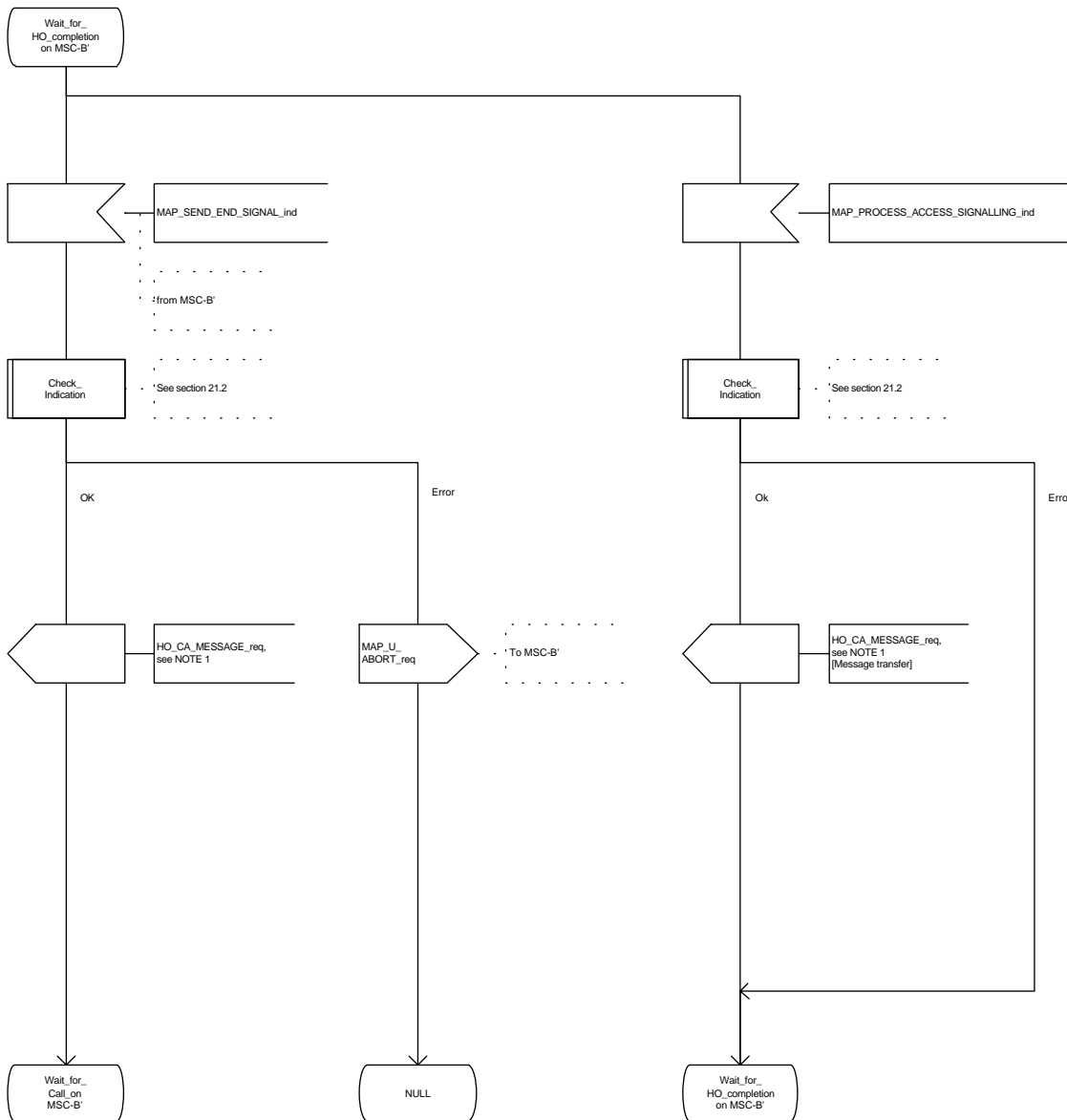
Figure 16.2.2/1 (sheet 12 of 12): Process MSC_A_HO
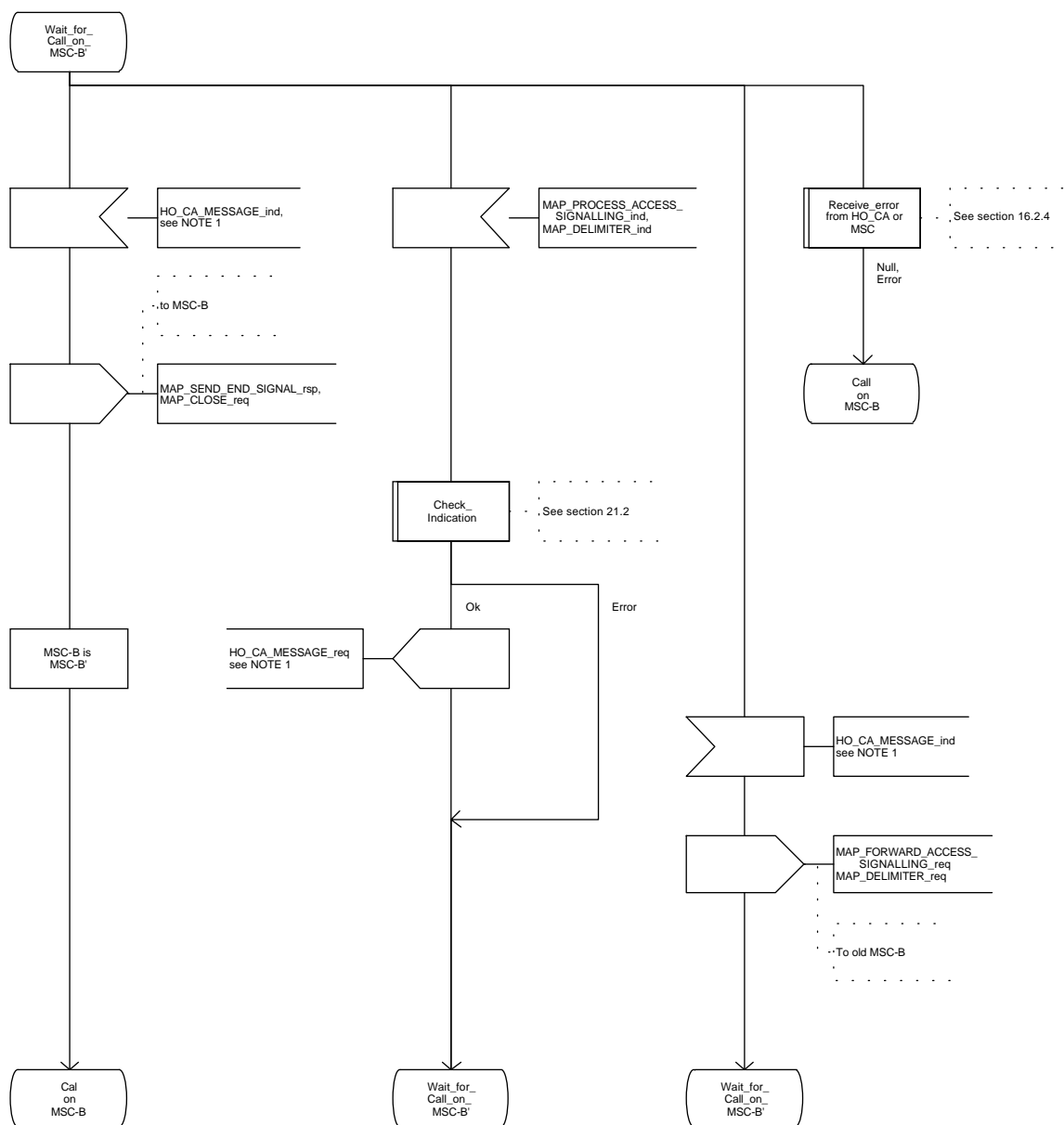
16.2.3   Handover procedure in MSC-B

This subclause describes the handover procedure in MSC-B, including the request for a handover from another MSC (MSC-A), subsequent handover to a third MSC (MSC-B') or back to the controlling MSC (MSC-A).

16.2.3.1   Basic handover

Opening of the dialogue is described in the macro Receive_Open_Ind in subclause 21.1.

When MSC-B process receives a MAP_PREPARE_HANDOVER indication from MSC-A, MSC-B requests its associated VLR to provide a handover number, unless the parameter HO-NumberNotRequired is received in the indication.

When the connection between the MS and MSC-B is established on MSC-B, the Handover Control Application will request the MAP application to indicate this event to MSC-A by invoking the MAP_SEND_END_SIGNAL request. When a call is released, MSC-A will inform MSC-B by MAP_SEND_END_SIGNAL response and the MAP dialogue between MSC-A and MSC-B is closed.

16.2.3.2   Allocation of handover number

When a handover number is required, a MAP_ALLOCATE_HANDOVER_NUMBER request will be sent to the VLR. The handover number is received in the MAP_SEND_HANDOVER_REPORT request, and will be included in the MAP_PREPARE_HANDOVER response to MSC-A.

As soon as the call from MSC-A using the handover number arrives in MSC-B, MSC-B shall release the handover number in the VLR using the MAP_SEND_HANDOVER_REPORT response.

16.2.3.3   Handling of access signalling

If required by the Handover Control Application, MSC-B invokes the MAP_PROCESS_ACCESS_SIGNALLING request containing the information received on the A-interface that should be transferred to MSC-A (e.g. call control information).

MAP_PROCESS_ACCESS_SIGNALLING is a non-confirmed service and any response from MSC-A will require a MAP_FORWARD_ACCESS_SIGNALLING request.

16.2.3.4   Other procedures in stable handover situation

During a call and after handover, a number of procedures between MSC-A and BSS-B controlled by or reported to MSC-A may be initiated by involving access signalling transfer in both directions.

16.2.3.5   Subsequent handover

The procedure is used when the Handover Control Application in MSC-B has decided that a call is to be handed over to another MSC (either back to the controlling MSC (MSC-A) or to a third MSC (MSC-B')).

After the MAP_PREPARE_SUBSEQUENT_HANDOVER response is received from MSC-A, MSC-B will await the disconnection of the call. Once the disconnect is complete, MSC-B will inform its VLR by invoking the MAP_SEND_HANDOVER_REPORT confirmation. VLR-B will then release the allocated handover number.

The subsequent handover procedure is shown in figure 16.2/3.

16.2.3.6   SDL Diagrams

The SDL diagrams on the following pages describe the user process in MSC-B for the procedures described in this subclause.

The services used are defined in subclause 6.4.

NOTE 1:   The message primitives HO_CA_MESSAGE in the SDL-diagrams are used to show the internal co-ordination between the MAP application and the Handover Control Application. For a detailed description of the co-ordination between the applications for the handover procedure, see GSM 03.09.

NOTE 2:   The order in the SDL diagrams to allocate first the handover number and then the radio resources is not binding.

Figure 16.2.3/1 (sheet 1 of 11): Process MSC_B_HO
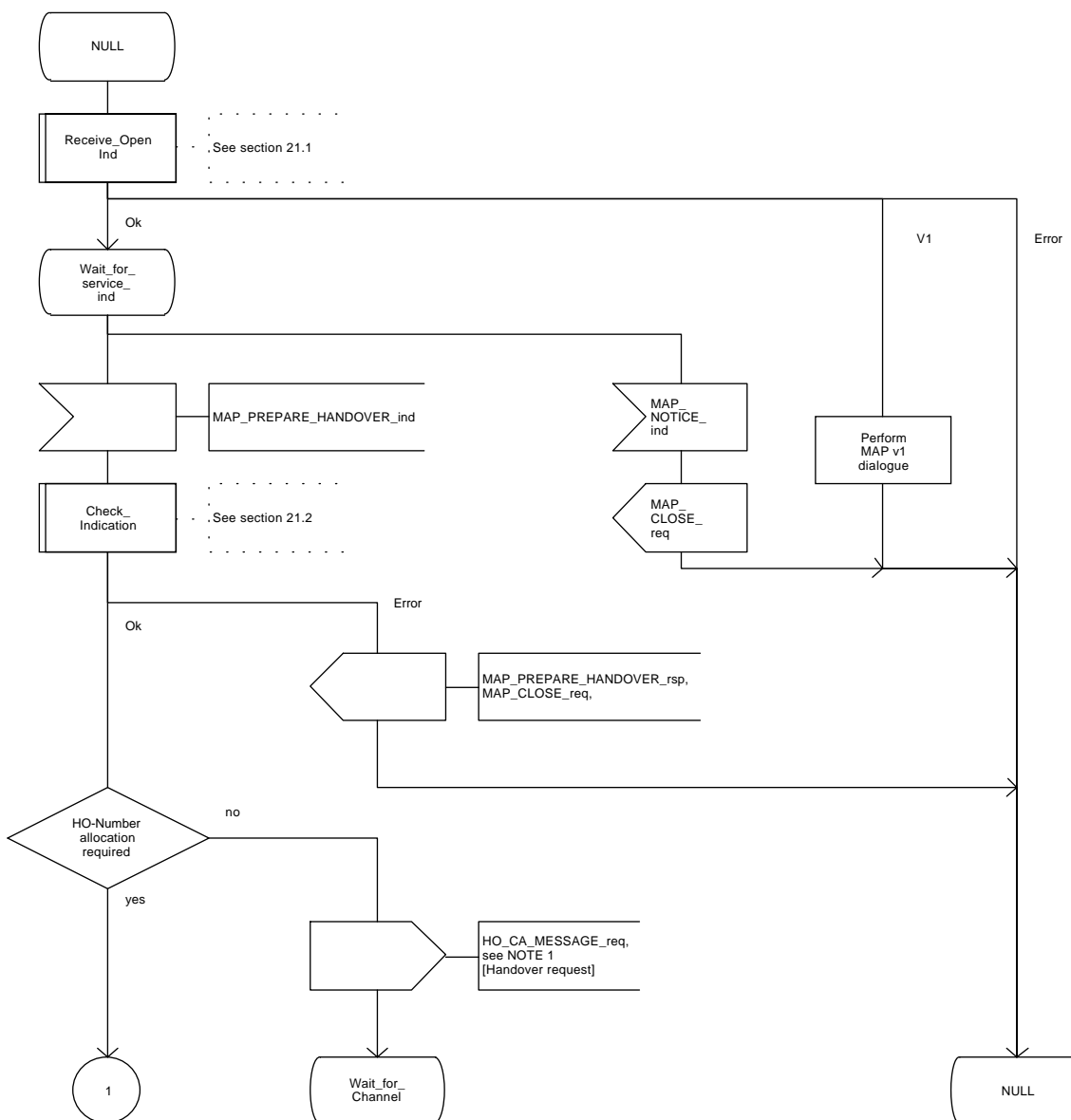
Figure 16.2.3/1 (sheet 2 of 11): Process MSC_B_HO
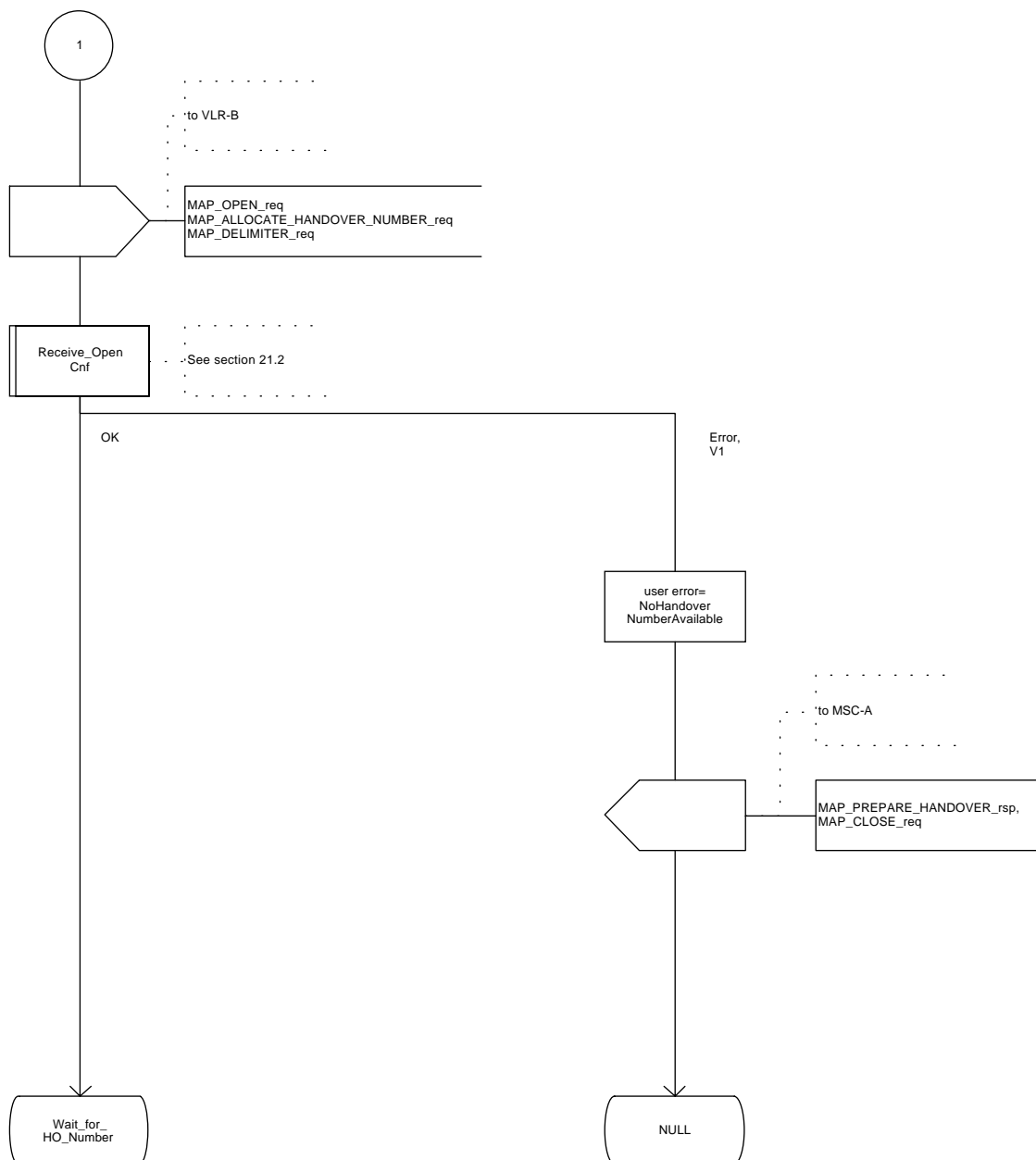
Figure 16.2.3/1 (sheet 3 of 11): Process MSC_B_HO
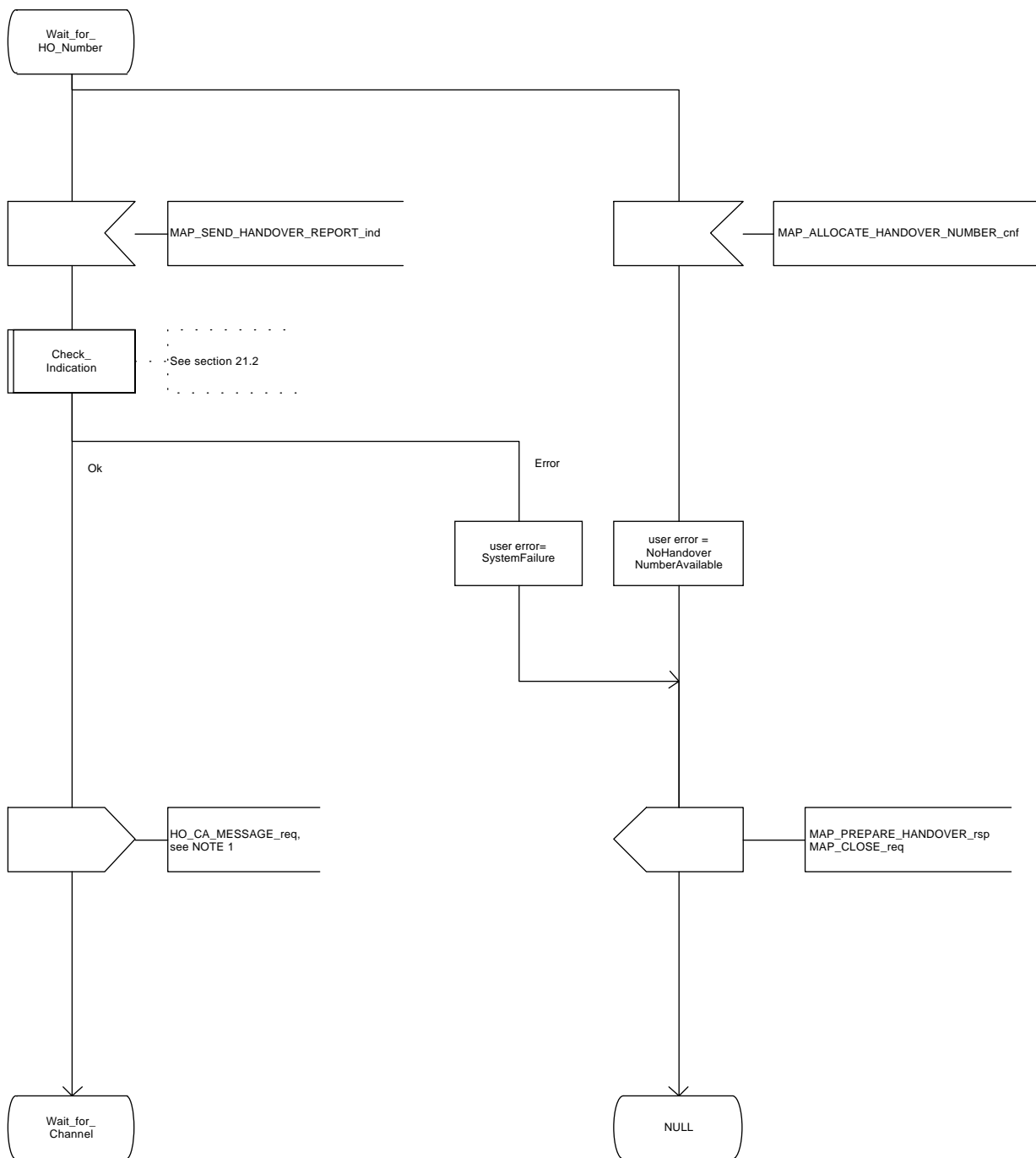
Figure 16.2.3/1 (sheet 4 of 11): Process MSC_B_HO
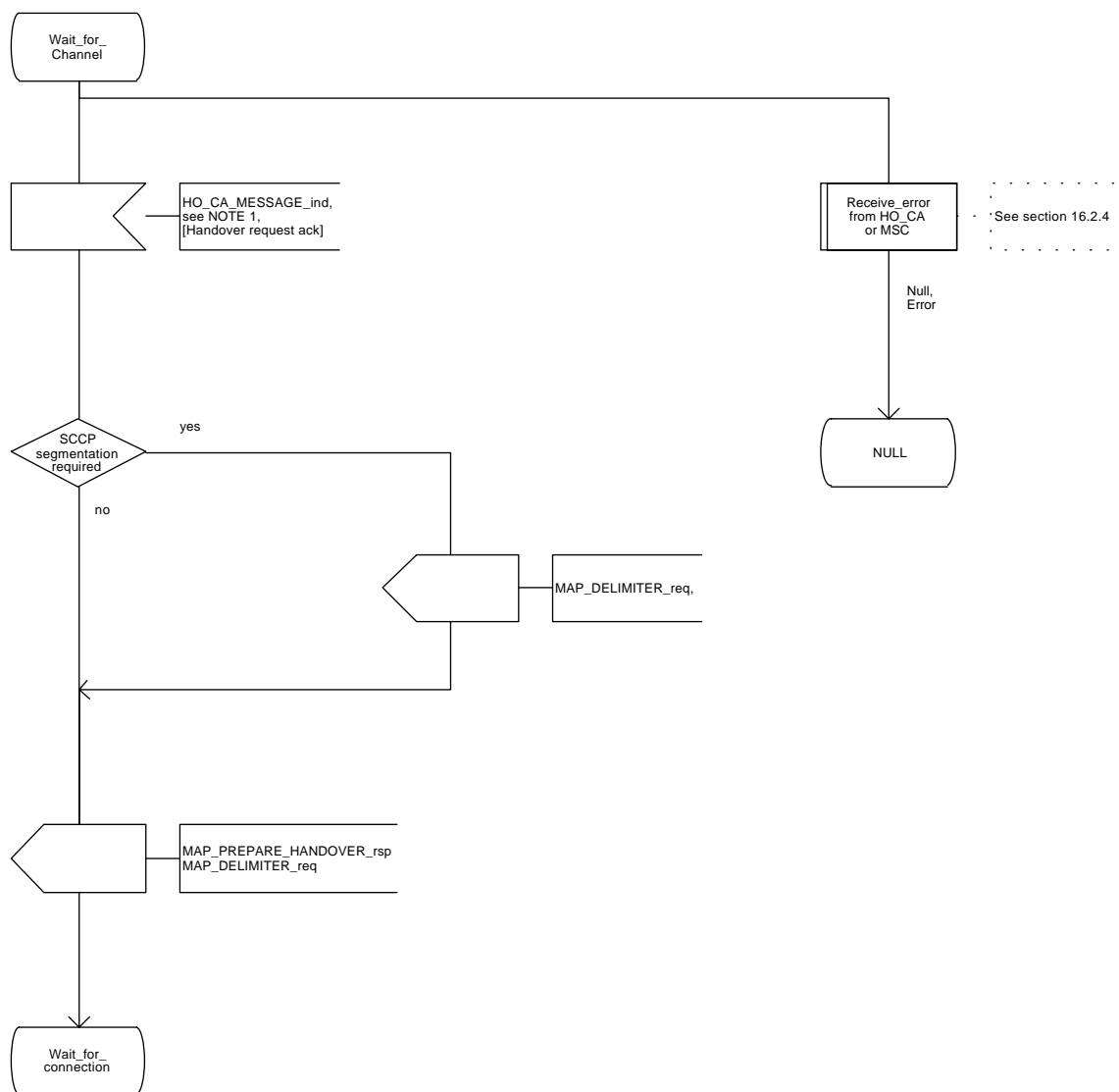
Figure 16.2.3/1 (sheet 5 of 11): Process MSC_B_HO
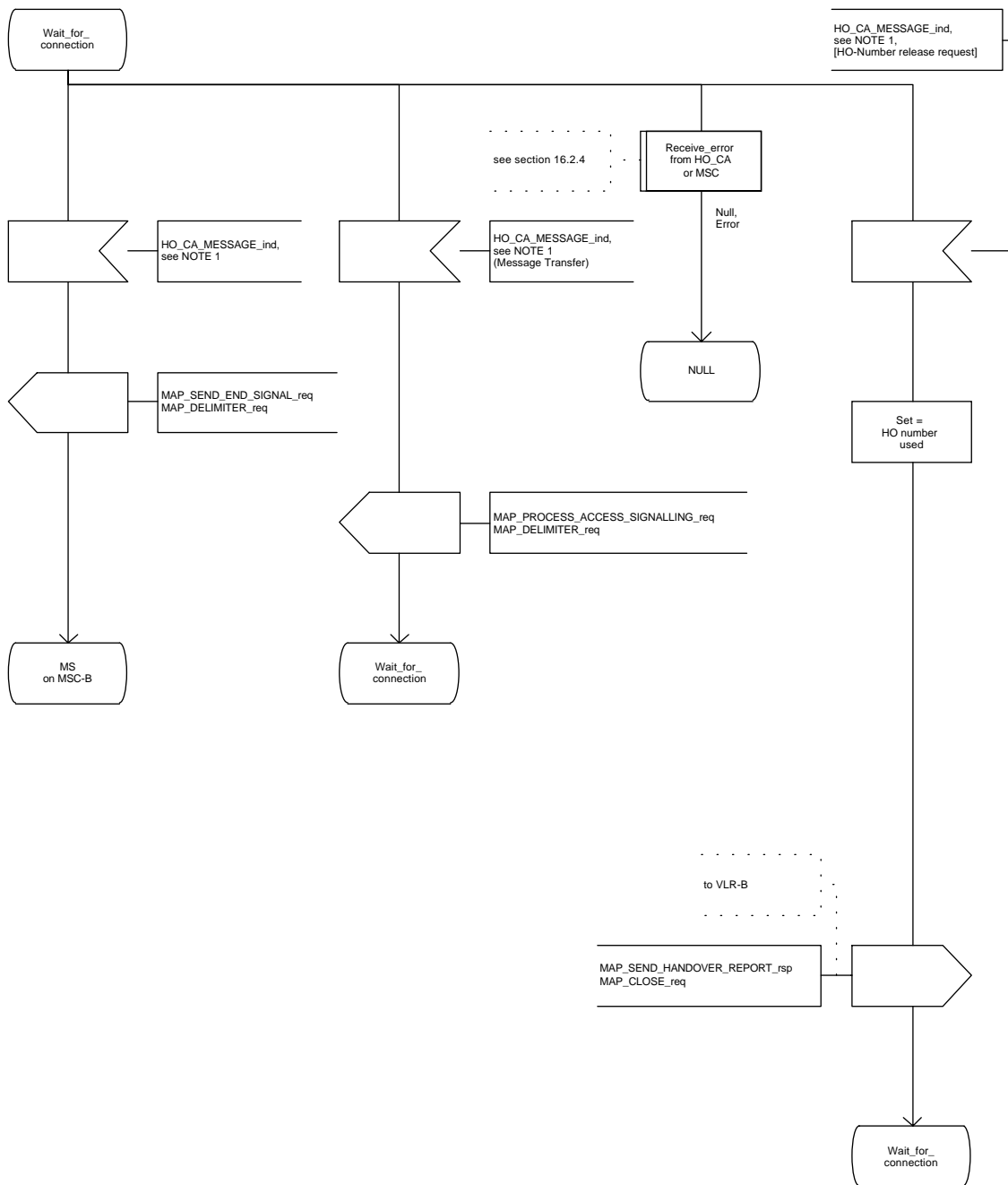
Figure 16.2.3/1 (sheet 6 of 11): Process MSC_B_HO

MS
on MSC-B

HO_CA_MESSAGE_ind,
see NOTE 1

MAP_FORWARD_ACCESS_SIGNALLING_ind
MAP_DELIMITER_ind

MAP_PROCESS_ACCESS_SIGNALLING_req
MAP_DELIMITER_req

HO_CA_MESSAGE_req,
see NOTE 1

MS
on MSC-B
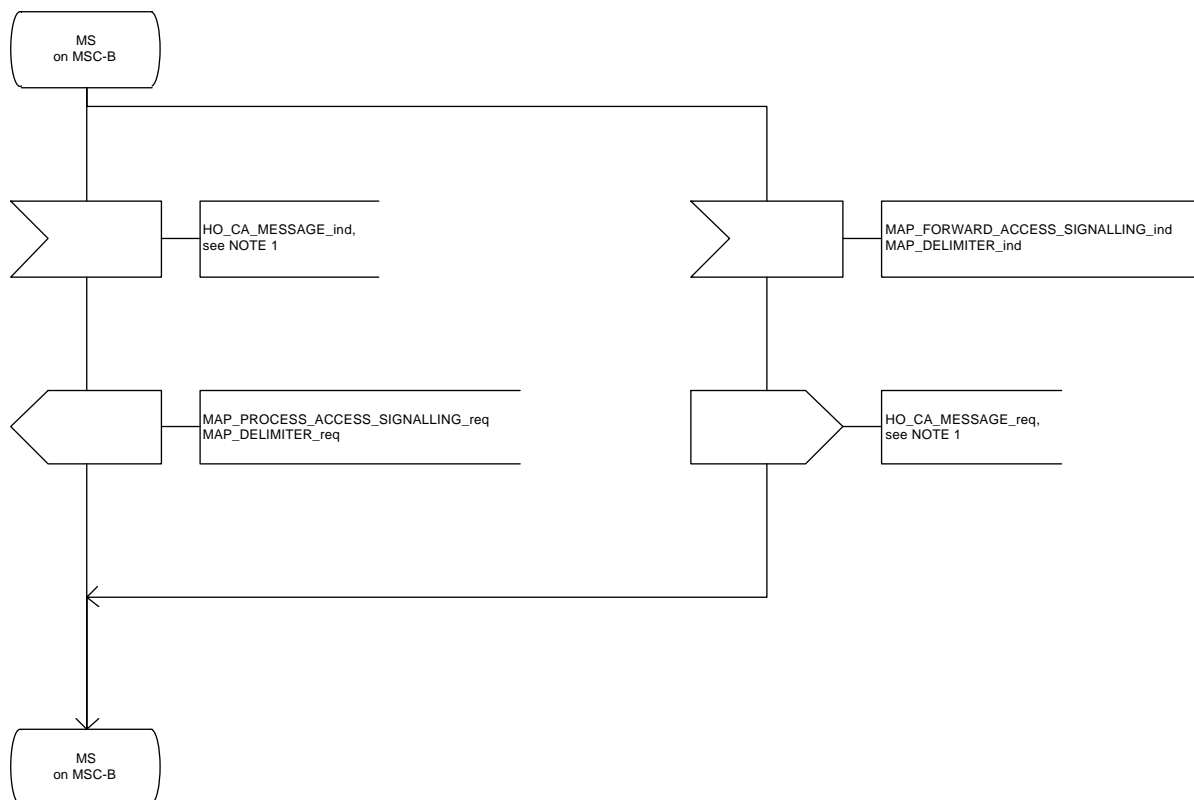
Figure 16.2.3/1 (sheet 7 of 11): Process MSC_B_HO
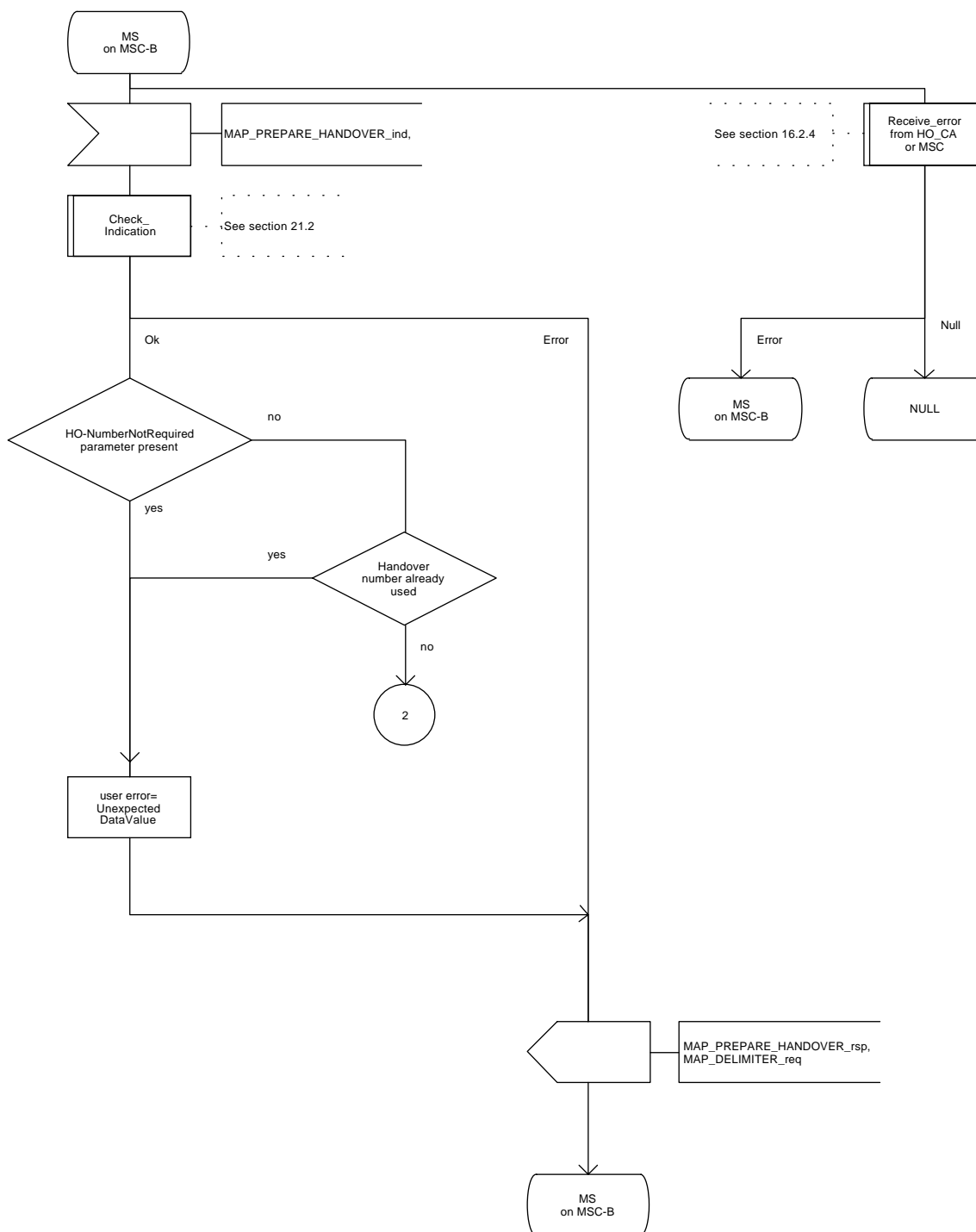
Figure 16.2.3/1 (sheet 8 of 11): Process MSC_B_HO

to VLR-B

MAP_OPEN_req,
MAP_ALLOCATE_HANDOVER_NUMBER_req,
MAP_DELIMITER_req,

Receive_Open
Cnf

See section 21.2

OK

Error,
V1

user error=
NoHandover
NumberAvailable

MAP_PREPARE_HANDOVER_rsp,
MAP_DELIMITER_req

Wait_for_
HO_Number
allocation

NULL
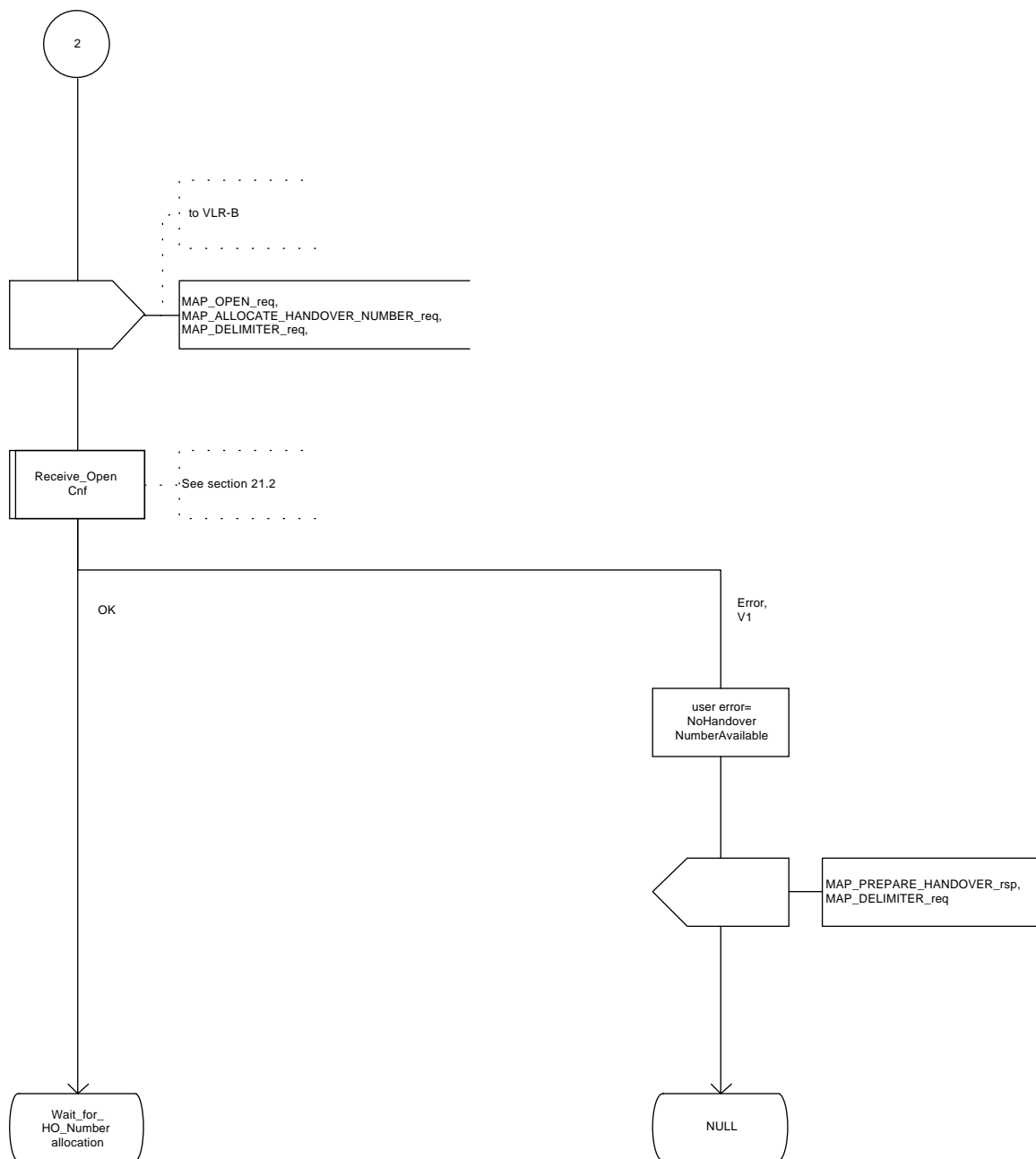
Figure 16.2.3/1 (sheet 9 of 11): Process MSC_B_HO
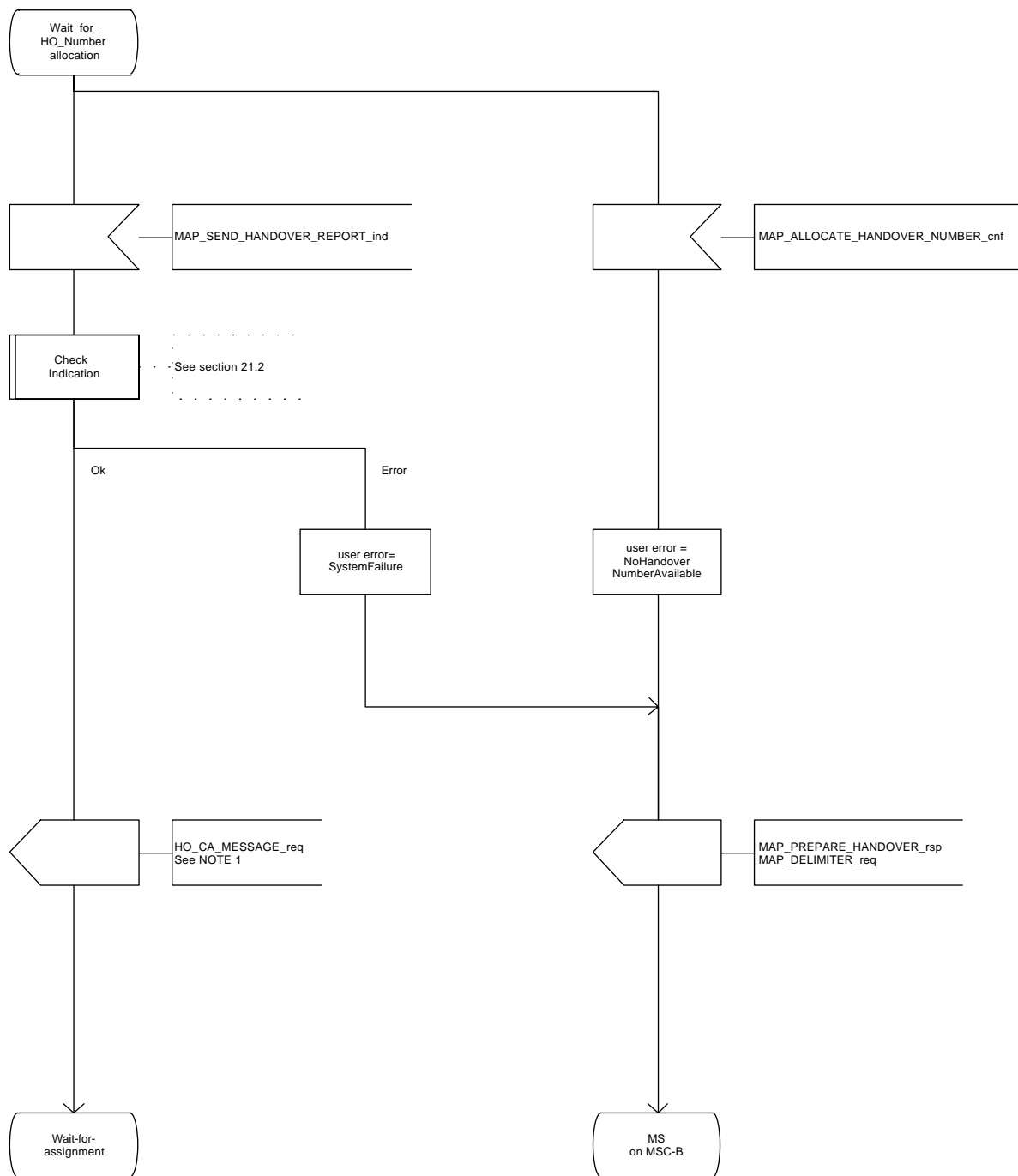
Figure 16.2.3/1 (sheet 10 of 11): Process MSC_B_HO
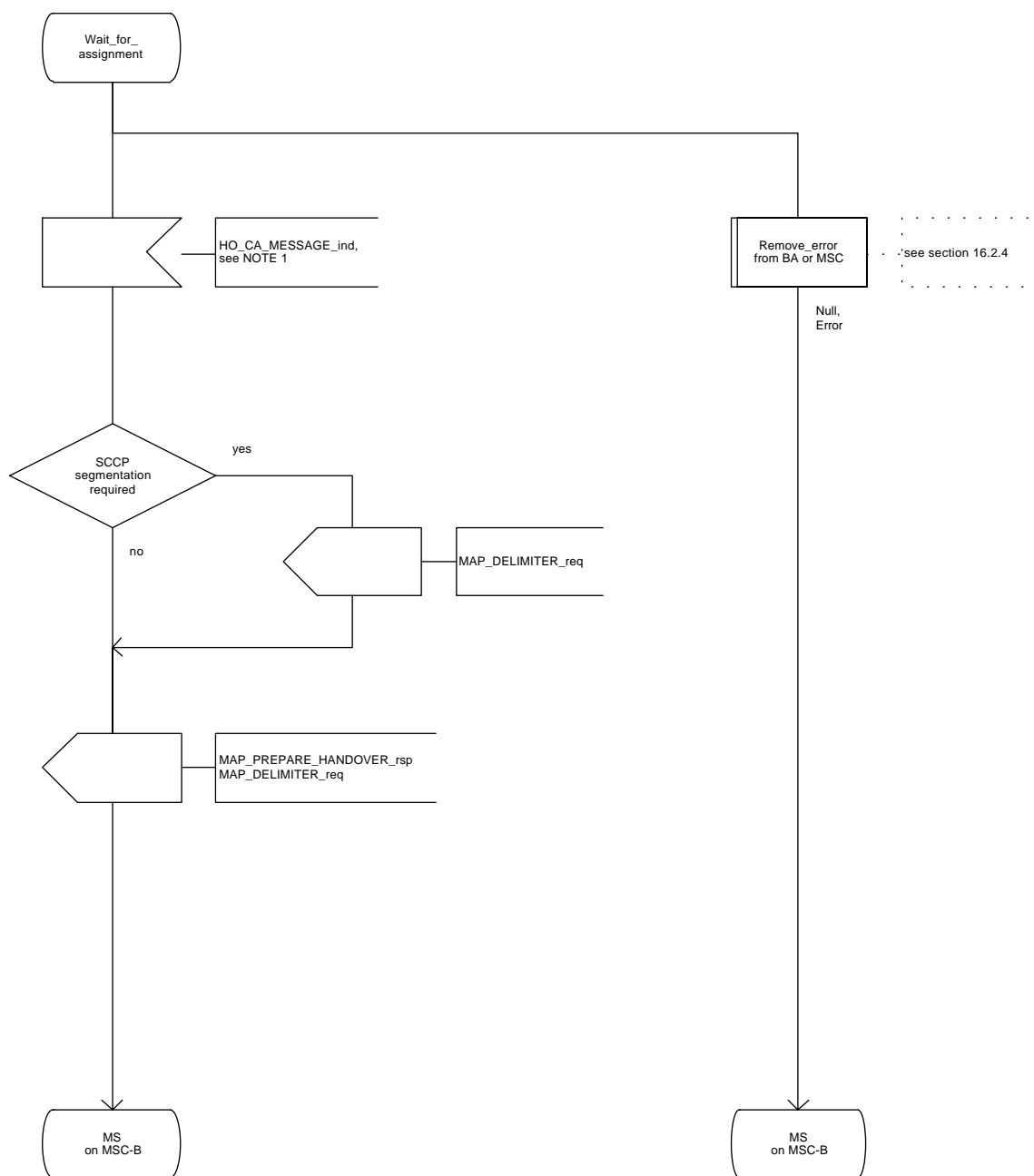
Figure 16.2.3/1 (sheet 11 of 11): Process MSC_B_HO

16.2.4 Handover error handling macro

This macro is used for the handover procedures to receive errors from the MSCs and from the Handover Control Application at any state of a handover process.

If a MAP_NOTICE indication is received, the Handover Control Application is informed and the actual situation is kept and the Handover Control Application decides how the handover process should continue. In all other cases the MSC is returned to a "NULL" state.

Figure 16.2.4/1: Macro Receive_error_from_HO_CA_or_MSC
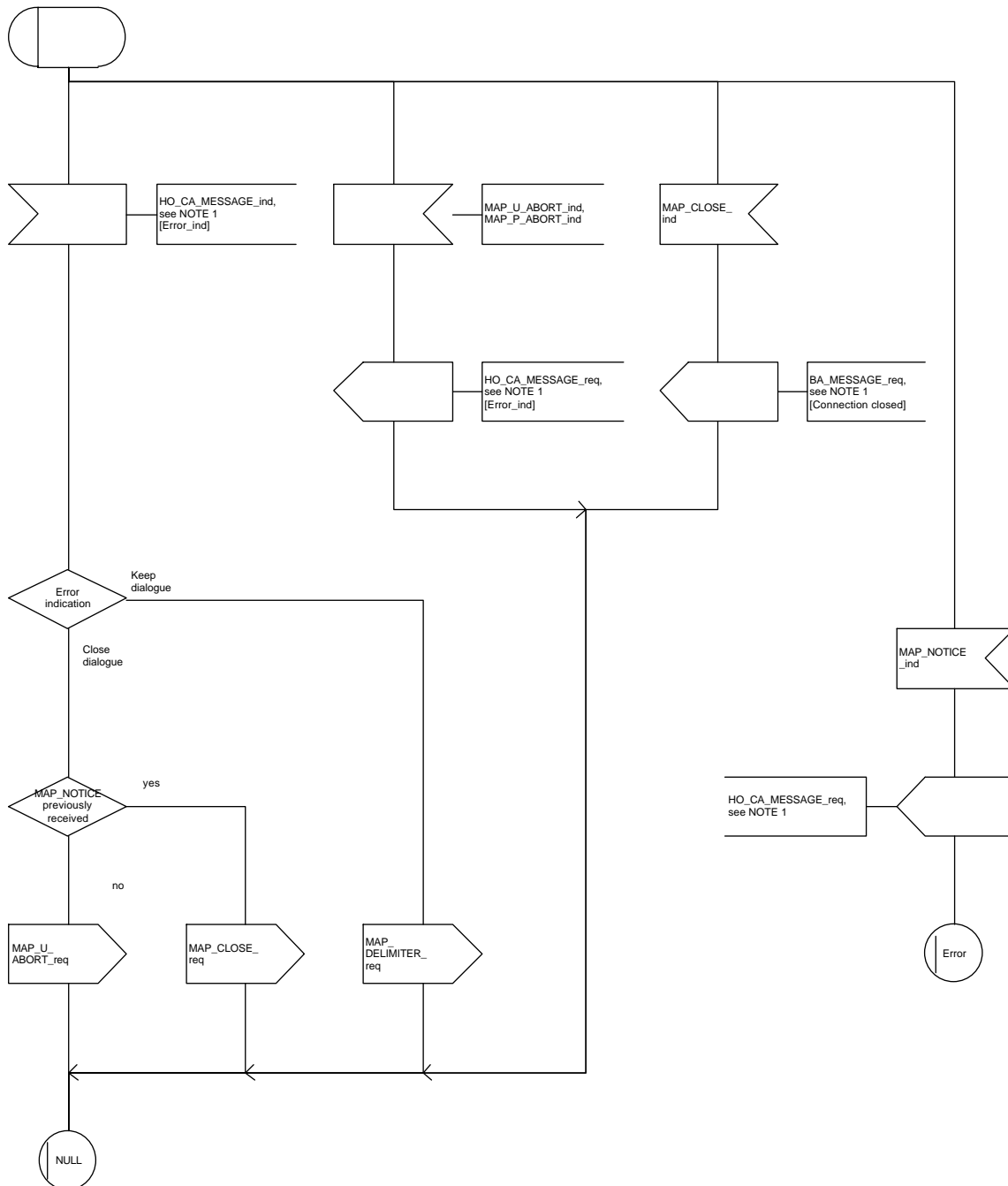
16.2.5 Handover procedure in VLR

16.2.5.1 Allocation of handover number

When receiving the MAP_ALLOCATE_HANDOVER_NUMBER indication, the VLR will determine whether a handover number is available. If no handover number is available, this will be indicated by a MAP_ALLOCATE_HANDOVER_NUMBER response with the appropriate error.

The handover number allocated will otherwise be returned to MSC-B in the MAP_SEND_HANDOVER_REPORT request.

The handover number will be reserved until a MAP_SEND_HANDOVER_REPORT confirmation is received from MSC-B.

16.2.5.2 SDL Diagrams

The SDL diagrams on the following pages describe the user processes in VLR for the procedures described in this subclause.

The services used are defined in subclause 6.4.

Figure 16.2.5/1 (sheet 1 of 2): Process VLR_B_HO
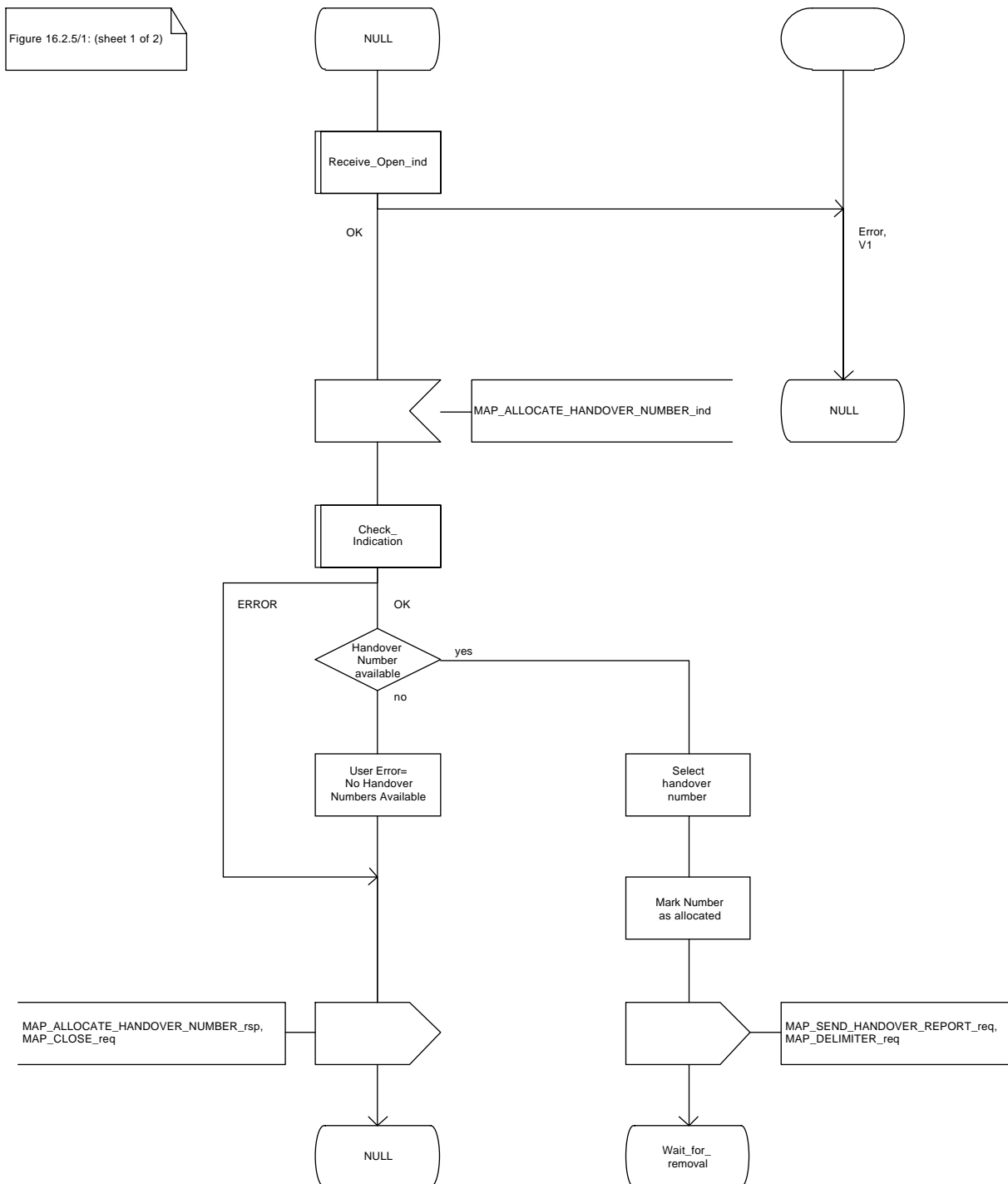
Figure 16.2.5/1 (sheet 2 of 2): Process VLR_B_HO

Figure 16.2.5/1: (sheet 2 of 2)

```
           ┌──────────┐
           │ Wait_for_ │
           │ removal  │
           └────┬─────┘
                │
    ┌───────────┼────────────────────────┬────────────────────────┐
    │           │                         │                        │
  ┌─┴───────────┴─┐                    ┌──┴──────────┐          ┌──┴──────────┐
  │  MAP_SEND_HANDOVER_REPORT_cnf │   │  LOCAL ABORT │         │ MAP_U_ABORT_ind │
  └─┬─────────────┘                   └──┬───────────┘          └──┬──────────┘
    │                                    │                         │
 ┌──┴───────┐                            │                         │
 │ Check_   │                            │                         │
 │ Confirmation │                        │                         │
 └──┬───────┘                            │                         │
    │ OK,                                │                         │
    │ Provider Error,                    │                         │
    │ Data Error,                        │                         │
    │ User Error                         │                         │
 ┌──┴───────┐                       ┌────┴─────────┐               │
 │ Mark handover │                  │ MAP_U_ABORT_req │            │
 │ number as free │                 └────┬─────────┘               │
 └──┬───────┘                            │                         │
    │                                    │                         │
    ├────────────────────────────────────┴─────────────────────────┘
    │
 ┌──┴───────┐
 │  NULL    │
 └──────────┘
```
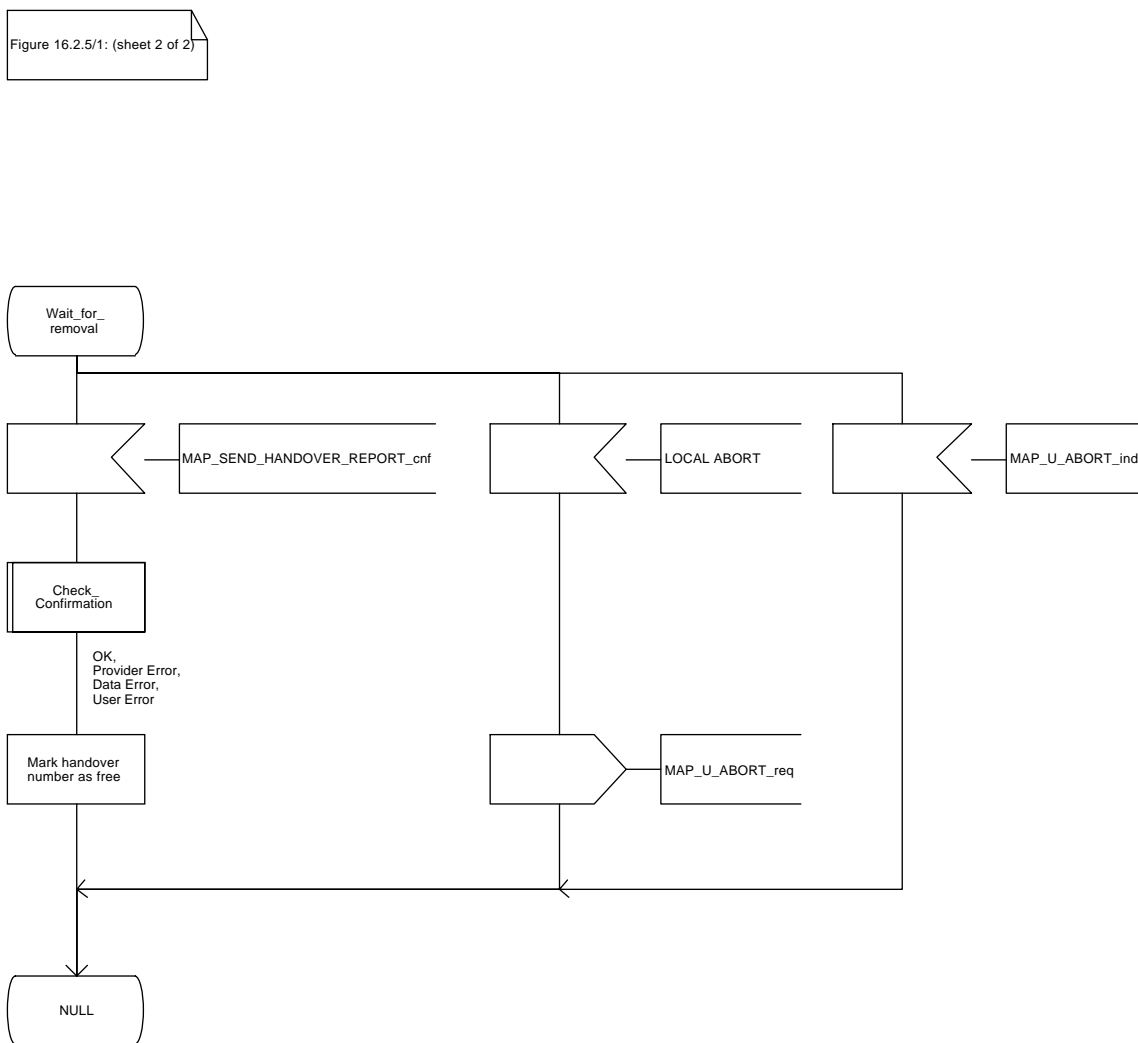
16.3 Fault recovery procedures

After a fault of a location register, the fault recovery procedures ensure that the subscriber data in the VLR become consistent with the subscriber data that are stored in the HLR for the MS concerned and that the location information in HLR and VLR reflect accurately the current location of the MS.

The detailed specification of fault recovery procedures of location registers is given in GSM 03.07.

16.3.1 VLR fault recovery procedures

The following processes are involved with the restoration of one IMSI record in the VLR:

- In case of a location registration request from the MS:
  Update_Location_Area_VLR   subclause 16.1.1.3;
  Update_Location_HLR   subclause 16.1.1.4.

- In case of a mobile terminated call:
  PRN_VLR   subclause 18.2.4;
  RESTORE_DATA_VLR   subclause 18.2.4;
  RESTORE_DATA_HLR   subclause 16.3.3;
  ICS_VLR   subclause 18.3.3.

After a restart, the VLR shall erase all IMSI records affected by the failure and shall cause all affected TMSIs and all affected LMSIs to become invalid. There will be no subscriber data or location information stored for an affected MS until after the VLR has received either a MAP_PROVIDE_ROAMING_NUMBER indication or a MAP_UPDATE_LOCATION_AREA indication for that MS. Restoration of subscriber data in the VLR is triggered individually for each IMSI record by receipt of either of these indications.

Reception of either a MAP_UPDATE_LOCATION_AREA indication or a MAP_PROVIDE_ROAMING_NUMBER indication with an IMSI that is unknown in the VLR causes creation of a skeleton IMSI record that is marked as:

- not confirmed by radio contact by the indicator "Confirmed by Radio Contact" (The function of this indicator is described in GSM 03.07), and

- not confirmed by HLR by the indicator "Confirmed by HLR" (The function of this indicator is described in GSM 03.07).

A third indicator "Location Information Confirmed in HLR" is allocated to each IMSI record in the VLR (The function of this indicator is described in GSM 03.07).

The indicator "Location Information Confirmed in HLR" shall be checked whenever authenticated radio contact with an MS has been established. The status "Not Confirmed" of this indicator shall force the VLR to invoke the MAP_UPDATE_LOCATION service but it shall never cause rejection of a mobile originated request. The status is changed from "Not Confirmed" to "Confirmed" only after successful completion of a MAP_UPDATE_LOCATION procedure for the MS concerned.

If the VLR serves only one MSC, the indicator "Location Information Confirmed in HLR" is only relevant to the HLR restoration procedure and an initial value must be assigned when an IMSI record is created in the VLR:

- if the IMSI record was created due to a roaming number request, the initial value must be set to "Confirmed";

- if reception of a MAP_UPDATE_LOCATION_AREA indication causes creation of the IMSI record, the initial value must be "Not Confirmed".

If the VLR serves more than one MSC, the indicator "Location Information Confirmed in HLR" is used in the VLR restoration procedure as well as in the HLR restoration procedure. When an IMSI record is created in the VLR, the indicator must be set to "Not Confirmed".

VLR restoration triggered by a location registration request

Upon receipt of a MAP_UPDATE_LOCATION_AREA indication, the VLR retrieves authentication data from the HLR by using the MAP_SEND_AUTHENTICATION_INFO service if authentication is required and if no authentication data are available in the VLR for the IMSI concerned (see figure 16.1.1/6).

Receipt of a MAP_UPDATE_LOCATION_AREA indication for an MS whose IMSI is unknown in the VLR or whose data stored in the VLR are marked as "Not Confirmed" by the indicator "Confirmed by HLR" and/or by the indicator "Location Information Confirmed in HLR" forces the VLR to invoke the MAP_UPDATE_LOCATION service after successful authentication, if required. The location updating procedure is performed as described in subclause 16.1.

Any other mobile originated request from an MS whose IMSI is unknown in the VLR or whose subscriber data stored in the VLR are marked as "Not Confirmed" by the indicator "Confirmed by HLR" shall be rejected with error cause "Unidentified Subscriber". This causes the MS to trigger the location registration procedure.

After successful completion of the MAP_UPDATE_LOCATION procedure, the indicators "Confirmed by HLR" and "Location Information Confirmed in HLR" are set to "Confirmed".

The indicator "Confirmed by Radio Contact" is set to "Confirmed" when the radio contact with the MS is authenticated.

VLR restoration triggered by a roaming number request

Figure 16.3/1 illustrates the signalling sequence for restoration of an IMSI record in the VLR triggered by a mobile terminating call set-up.

Upon receipt of a MAP_PROVIDE_ROAMING_NUMBER indication for an IMSI that is unknown in the VLR and for which authentication is required, the VLR retrieves authentication data from the HLR by using the MAP_SEND_AUTHENTICATION_INFO service after an MSRN has been sent to the HLR in the MAP_PROVIDE_ROAMING_NUMBER response.

Receipt of a MAP_PROVIDE_ROAMING_NUMBER indication for an MS whose IMSI is unknown in the VLR or whose data record in the VLR is marked as "Not Confirmed" by the indicator "Confirmed by HLR" forces the VLR to request subscriber data from the HLR by sending a MAP_RESTORE_DATA request which triggers one or more INSERT_SUBSCRIBER_DATA operations from the HLR. The MAP_RESTORE_DATA request may also be used to send the LMSI to the HLR.

The MAP_RESTORE_DATA process in the VLR is described in subclause 18.2.4.

The MAP_RESTORE_DATA process in the HLR is described in subclause 16.3.3.

After successful completion of the MAP_RESTORE_DATA procedure, the indicator "Confirmed by HLR" is set to "Confirmed".

If restoration of an IMSI record was triggered by a MAP_PROVIDE_ROAMING_NUMBER indication (i.e. by a mobile terminating call), the VLR has no valid Location Area Identity information for the MS concerned before successful establishment of the first authenticated radio contact. Upon receipt of a MAP_SEND_INFO_FOR_INCOMING_CALL indication from the MSC (see 5 in figure 16.3/1) for an MS whose subscriber data are marked as "Confirmed" by the indicator "Confirmed by HLR" but not confirmed by radio contact, the VLR shall invoke a "MAP_SEARCH_FOR_MS" instead of a "MAP_PAGE".

A MAP_SEARCH_FOR_MS shall also be performed if the VLR receives a MAP_SEND_INFO_FOR_MT_SMS indication from the MSC for an MS whose IMSI record is marked as "Confirmed" by the indicator "Confirmed by HLR" but not confirmed by radio contact.

The indicator "Confirmed by Radio Contact" is set to "Confirmed" when authenticated radio contact caused by a mobile originated or a mobile terminated activity is established.

```
                        (B)                              D
   ┌─────┐                         ┌─────┐                         ┌─────┐
   │ MSC │─────────────────────────│ VLR │─────────────────────────│ HLR │
   └─────┘        (no external     └─────┘                         └─────┘
    │             interface)        │                                │
    │                               │                                │
    │                               │ MAP_PROVIDE_ROAMING_NUMBER   ) │
    │                               │<───────────────────────────── )│
    │                               │ PROVIDE_ROAMING_NUMBER_ack   )1 │
    │                               │─────────────────────────────> )│
    │                               │                                │
    │                               │ MAP_SEND_AUTHENTICATION_INFO )  │
    │               (note 1)        │─────────────────────────────> ) │
    │                               │ SEND_AUTHENTICATION_INFO_ack )2 │
    │                               │<───────────────────────────── )│
    │                               │                                │
    │                               │      MAP_RESTORE_DATA          │
    │                               │───────────────────────────────>│
  ( │                               │                                │
  ( │                               │ MAP_ACTIVATE_TRACE_MODE        │
  ( │               (note 2)        │<───────────────────────────────│
  ( │                               │ MAP_ACTIVATE_TRACE_MODE_ack    │
  ( │                               │───────────────────────────────>│
  ( │                               │                                │
 4( │                               │ MAP_INSERT_SUBSCRIBER_DATA   ) │
  ( │ 1) see subclause  18.2.4      │<───────────────────────────── )│
  ( │ 2) see subclause  21.5        │ INSERT_SUBSCRIBER_DATA_ack   )3 │
  ( │ 3) see subclauses 16.4.1      │─────────────────────────────> )│
  ( │                   21.7        │                                │
  ( │ 4) see subclauses 16.3.3      │    MAP_RESTORE_DATA_ack         │
  ( │                   18.2.4      │<───────────────────────────────│
    │                               │                                │
  ──┼───────────────────────────────┼───────────────────────────────┼──
  ( │ MAP_SEND_INFO_FOR_INCOMING_CALL or                             │
  ( │─────────────────────────────> │                                │
  ( │ MAP_SEND_INFO_FOR_MT_SMS      │                                │
  ( │                               │                                │
 5( │    MAP_SEARCH_FOR_MS          │                                │
  ( │<───────────────────────────── │                                │
  ( │                               │                                │
  ( │    MAP_SEARCH_FOR_MS_ack      │                                │
  ( │─────────────────────────────> │                                │
  ( │                               │                                │
  ( │ MAP_PROCESS_ACCESS_REQUEST    │                                │
  ( │─────────────────────────────> │ 5) see subclauses 18.3, 20.3   │
    │                               │                                │
```

NOTE 1:   If authentication required.

NOTE 2:   If subscriber tracing active in HLR.

Figure 16.3/1: Procedures related to restoration of VLR in case of mobile terminated call set-up

16.3.2 HLR fault recovery procedures

The following processes are involved with the restart of the HLR:

- HLR_RESTART   subclause 16.3.2;
- REC_RESET_IN_VLR   subclause 16.3.2.

In the case of a location registration request from the MS, the following processes are involved with the HLR restoration procedure:

- Update_Location_Area_VLR   subclause 16.1.1.3;
- Update_Location_HLR   subclause 16.1.1.4.

In the case of a mobile originated service request, the

- Macro Process_Access_Request_VLR   subclause 21.4.2; and the
- Process Update_Location_HLR   subclause 16.1.1.4,

are involved with the HLR restoration procedure.

For the HLR, periodic back-up of data to non-volatile memory is mandatory.

Data that have been changed in the period of time after the last back-up storage and before the restart of the HLR cannot be recovered by reload from the non-volatile memory. Therefore, a restoration procedure is triggered individually for each IMSI record that has been affected by the HLR fault at the first authenticated radio contact that is established with the MS concerned.

The HLR restoration procedure forces updating of MSC number, VLR number and, if provided by the VLR, LMSI in the HLR. Consistency of subscriber data that are stored in the VLR for an MS that has been affected by a HLR fault with the subscriber data stored in the HLR for this MS will be achieved.

As an implementation option, a notification can be forwarded to the MS to alert the subscriber to check the parameters for supplementary services that allow subscriber controlled input (MAP_FORWARD_CHECK_SS_INDICATION service). If the VLR receives this notification from the HLR it shall forward the notification to the MS.

Figure 16.3/2 illustrates the signalling sequence for HLR restoration.

After a restart, the home location register performs the following actions for the subscriber data records that have been affected by the HLR fault (see figure 16.3/3):

- reload all data from the non-volatile back-up;

- if the MAP_FORWARD_CHECK_SS_INDICATION service is implemented, mark each subscriber record "SS Check Required" by setting the "Check SS" indicator;

- set subscriber tracing deactive in the VLR for each of its Mss;

- reset the "MS Purged" flag for each of its MSs;

- send a MAP_RESET request to the VLRs where its MSs are located (see figure 16.3/4).

The MAP_RESET request contains the HLR number and optionally the HLR Identity List.

When receiving a MAP_RESET indication, the VLR will derive all involved MSs of that HLR either from the HLR Identity List (if present), or from the HLR number. The VLR will then mark these MSs with the indicator "Location Information Confirmed in HLR" set to "Not Confirmed" and will deactivate all subscriber tracings for these Mss (see figure 16.3/5).

The status "Not Confirmed" of the indicator "Location Information Confirmed in HLR" forces the VLR to invoke the MAP_UPDATE_LOCATION service after establishment of authenticated radio contact with the MS concerned.

The MAP_UPDATE_LOCATION procedure is performed as described in subclause 16.1.

After receipt of the MAP_UPDATE_LOCATION acknowledge containing the HLR number, the status of the indicator "Location Information Confirmed in HLR" is changed to "Confirmed".

If the MAP_UPDATE_LOCATION procedure is unsuccessful for any reason, the status of the indicator "Location Information Confirmed in HLR" remains unchanged except for the case that the IMSI record in the VLR is deleted because either of the errors "Unknown Subscriber" or "Roaming Not Allowed" has been received from the HLR in response to a MAP_UPDATE_LOCATION request.
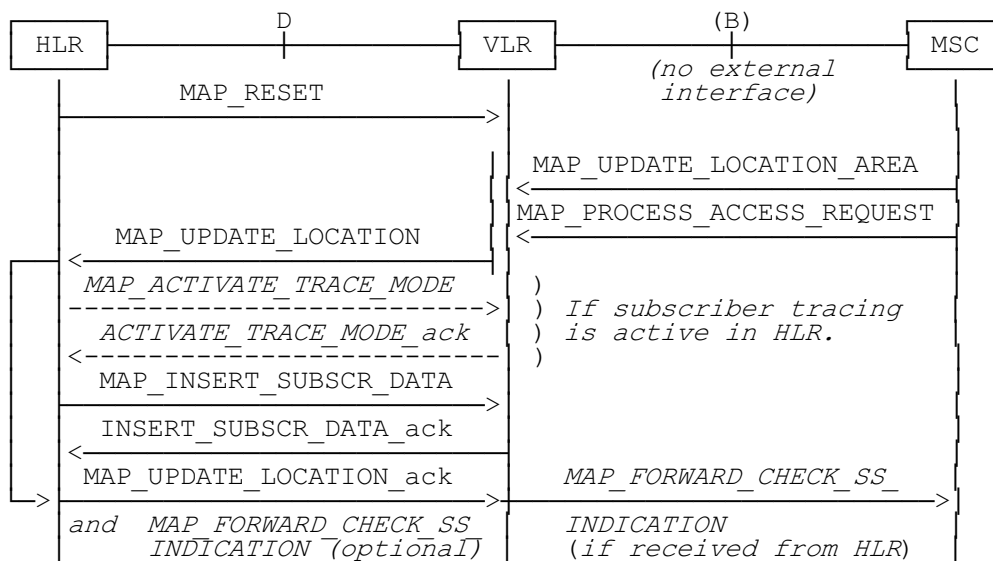
Figure 16.3/2: Procedures related to restoration of HLR

Figure 16.3/3: Process HLR_RESTART

Process HLR_RESTART   163_3(1)

Figure 16.3/3 Restoration of the HLR
               Application process in the HLR for HLR Restart

NULL

OM_
RESTART
_Ind

'Reload_from_
backup'

CHECK
SS_service

'not_implemented'   'implemented'

'CHECK_SS_Indicator   For each MS that is affected
:=   by the HLR fault.
Check_Required'

Set   For each MS that is affected
Subscriber tracing   by the HLR fault
= deactive in VLR

'Reset_Flag   For each MS that is affected
MS_PURGED'   by the HLR fault.

'Read_VLR_
address_list'

SEND_RESET   Figure 16.3/4
_TO_VLR

'Yes'   Another_
             VLR_in_the_
             list

                  'No'

NULL

Figure 16.3/4: Process SEND_RESET_TO_VLR
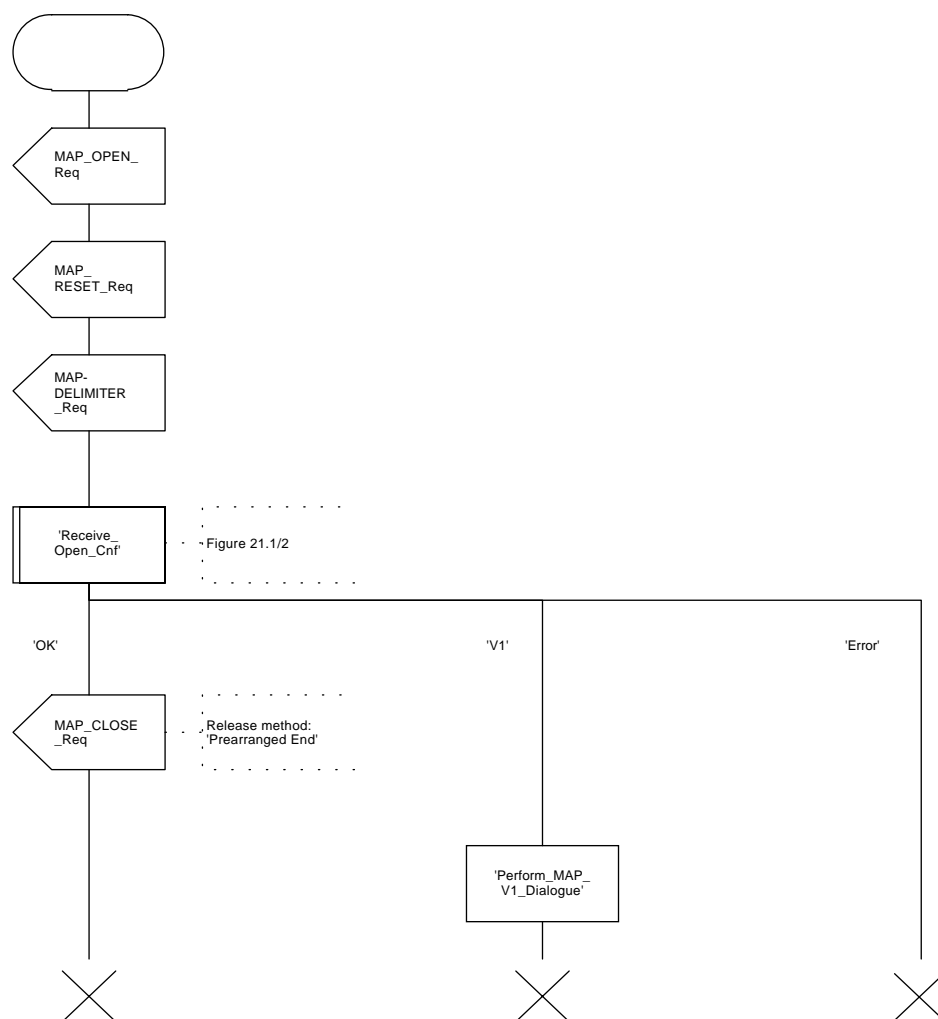
Figure 16.3/5: Process REC_RESET_IN_VLR

Process REC_RESET_IN_VLR — 163_5(1)

Figure 16.3/5 Restoration of the HLR - Application process in the VLR
for reception of the RESET message from HLR

NULL

Receive_
OPEN_Ind — Figure 21.1/1

'OK'   'V1'   'Error'

WAIT

'Perform_MAP_
V1_Dialogue'

MAP_RESET
_Ind — contains HLR_Number or
HLR_Number and
HLR_Id_List

NULL   NULL

MAP_CLOSE
_Req — Release method:
'Prearranged End'

'Identify_IMSI
_Records' — The IMSIs are derived from
the HLR_Number
or from the HLR_Id_List.

'Location_Info_
Conf_in_HLR :=
Not_Confirmed' — For each identified IMSI record.

Deactivate Subscriber Tracing — For each identified IMSI record

NULL

16.3.3   VLR restoration: the restore data procedure in the HLR

The MAP_RESTORE_DATA procedure in the HLR (Process RESTORE_DATA_HLR) is described in this subclause; the corresponding procedure in the VLR (RESTORE_DATA_VLR) is described in subclause 18.2.4.

The process RESTORE_DATA_HLR makes use of the following macros:

- Receive_Open_Ind   subclause 21.1.1;
- Check_Indication   subclause 21.2.1;
- Insert_Subs_Data_Framed_HLR   subclause 16.4.1.

The MAP_RESTORE_DATA service is invoked by the VLR after provision of a roaming number in response to a MAP_PROVIDE_ROAMING_NUMBER indication for an unidentified MS (i.e. IMSI unknown in VLR), or for a known MS whose IMSI record is marked as "Not Confirmed" by the indicator "Confirmed by HLR" (see 4 in figure 16.3/1). The process RESTORE_DATA_VLR is shown in figure 18.2/6.

The restore data process in the HLR is activated by receipt of a MAP_RESTORE_DATA indication from the VLR (see figure 16.3/6). If there is a parameter problem in the indication, either of the errors "Unexpected Data Value" or "Data Missing" is returned in the MAP_RESTORE_DATA response; if the subscriber is not known in the HLR, the error "Unknown Subscriber" is returned in the MAP_RESTORE_DATA response. In all of these cases the process in the HLR terminates.

If the MAP_RESTORE_DATA indication is accepted and if the LMSI is received, the HLR updates the LMSI for the IMSI received in the MAP_RESTORE_DATA indication. For this IMSI the HLR sets "subscriber-tracing-not-active-in-VLR" and checks whether tracing is required. This check is handled by the macro "Control_Tracing_HLR" that is described in subclause 21.9. Thereafter, the macro "Insert_Subs_Data_Framed_HLR" that is described in subclause 16.4.1 is invoked. The outcome of the macro Insert_Subs_Data_Framed_HLR is one of:

- abort, in which case the process terminates;

- error, in which case the HLR returns the error "System Failure" in the MAP_RESTORE_DATA response, and the process terminates;

- OK, indicating successful outcome of downloading the subscriber data to the VLR.

After successful completion of the framed MAP_INSERT_SUBSCRIBER_DATA procedure, the HLR Number and, if applicable, the "MS Not Reachable Flag" which is used for SMS, are provided in the MAP_RESTORE_DATA response.

Upon receipt of the MAP_RESTORE_DATA confirmation, the VLR behaves as described in subclause 18.2.4, figure 18.2/6.

Figure 16.3/6: Process RESTORE_DATA_HLR
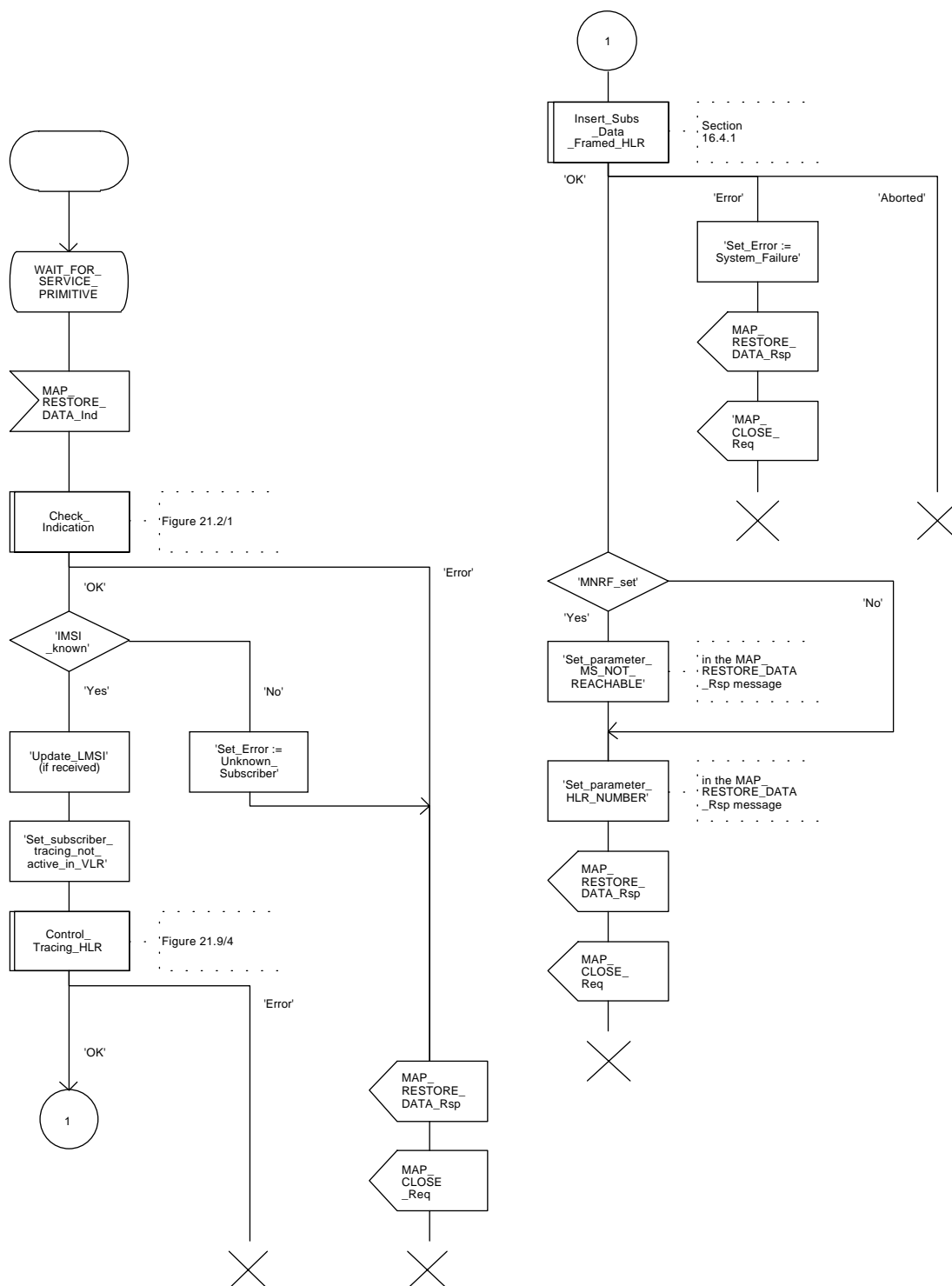
16.4 Macro Insert_Subs_Data_Framed_HLR

This macro is used by any procedure invoked in HLR which requires the transfer of subscriber data by means of the InsertSubscriberData operation (e.g. Update Location or Restore Data).

The invocation of the operation is done in a dialogue already opened by the framing procedure. Therefore the latter is the one that handles the reception of the open indication and sends the dialogue close request.

The macro calls the process "Send_Insert_Subs_Data" (see subclause 21.7.4) as many times as it is needed for transferring all subscriber data. This process call is meant to describe two possible behaviours of HLR to handle service requests and confirmations:

- either the HLR handles requests and confirmations in parallel; or

- the HLR sends the next request only after receiving the confirmation to the previous one.

Another call is done to the macro "Wait_for_Insert_Subscriber_Data" (see subclause 21.7.3). There the reception and handling of the service confirmations is described.

If certain services required for a subscriber are not supported by the VLR (e.g. Advice of Charge Charging Level), this may result in one of the following outcomes:

- The HLR stores and sends "Roaming Restriction Due To Unsupported Feature" in a subsequent MAP_INSERT_SUBSCRIBER_DATA service. If "Roaming Restriction Due To Unsupported Feature" is stored in the HLR, the "MSC Area Restricted Flag" shall be set to "restricted". This will prevent MT calls, MT SM and MT USSD from being forwarded to the MSC/VLR;

- The HLR stores and sends other induced subscriber data (e.g. a specific barring program) in a subsequent MAP_INSERT_SUBSCRIBER_DATA service. This will cause rejection of mobile originated service requests, except emergency calls.

When the VLR receives regional subscription data (Zone Code List) it may respond with "MSC Area Restricted" in the MAP_INSERT_SUBSCRIBER_DATA response. In this case the "MSC Area Restricted Flag" shall be set to "restricted" in the HLR. This will prevent MT calls, MT SM and MT USSD from being forwarded to the MSC/VLR.

If the HLR neither stores "Roaming Restriction Due To Unsupported Feature" nor receives "MSC Area Restricted" in the MAP_INSERT_SUBSCRIBER_DATA response, the "MSC Area Restricted Flag" in the HLR shall be set to "not restricted".

The SDL diagram is shown in figure 16.4/1.

Figure 16.4/1: Macro Insert_Subs_Data_Framed_HLR
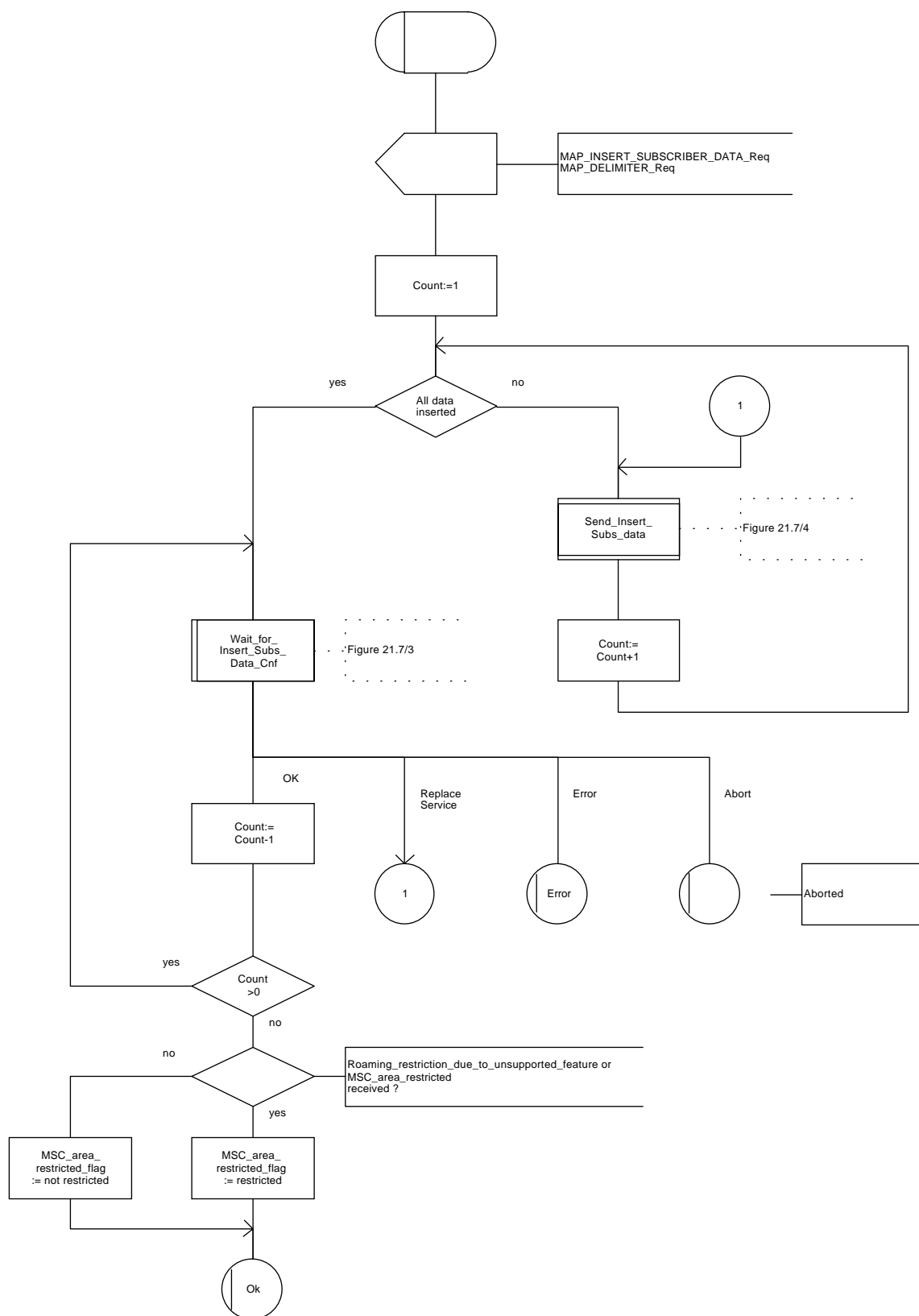
17 Operation and maintenance procedures

17.1 General

The Operation and Maintenance procedures are needed for operating and maintaining the GSM PLMN network.

The following procedures exist for operation and maintenance purposes:

i) Tracing procedures;
ii) Subscriber Data Management procedures;
iii) Subscriber Identity procedures.

The following application contexts refer to complex MAP Users consisting of several processes:

- subscriberDataManagementContext;
- tracingContext.

These two application contexts need a co-ordinating process in the VLR as described in the following subclauses.

17.1.1 Tracing Co-ordinator for the VLR

The MAP_OPEN indication opens the dialogue for the stand-alone tracing procedure when the application context tracingContext is received. If that service is successful, the Co-ordinator can receive the firs service primitive from the MAP_PM. Depending on the received primitive, the user process is created as follows:

- if the MAP_ACTIVATE_TRACE_MODE indication is received, the process ATM_VLR_Standalone is created;

- if the MAP_DEACTIVATE_TRACE_MODE indication is received, the process DTM_VLR_Standalone is created.

After creation of the user process the Co-ordinator relays the messages between the MAP_PM and the invoked process until a request or an indication for dialogue termination is received.

The Tracing Co-ordinator is shown in the figure 17.1/1.

Figure 17.1/1: Process Co_Tracing_VLR
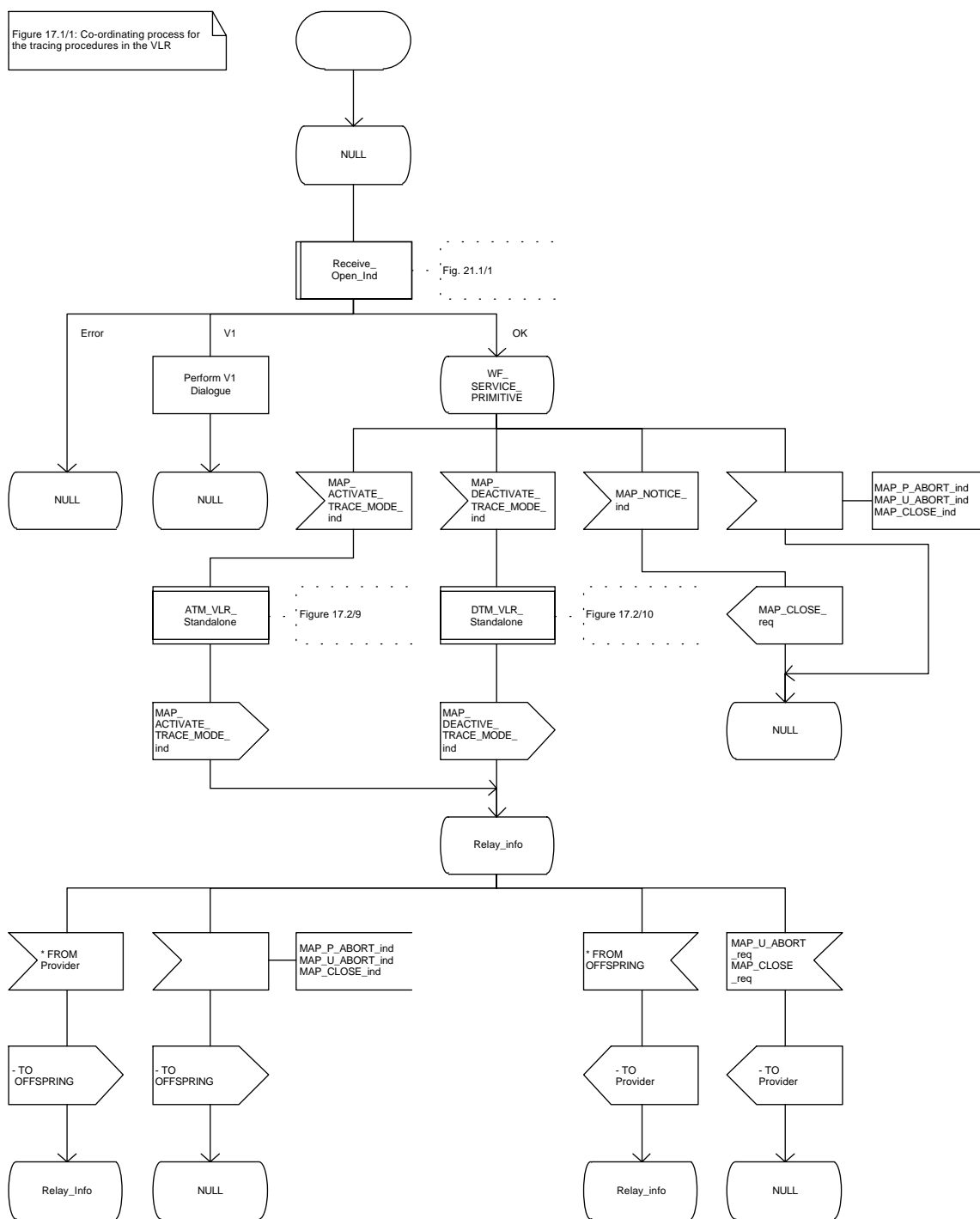
17.1.2 Subscriber Data Management Co-ordinator for the VLR

The MAP_OPEN indication opens the dialogue for the stand-alone subscriber data management procedure when the application context subscriberDataManagementContex is received. If that service is successful, the Co-ordinator can receive the firs service primitive from the MAP_PM. Depending on the received primitive, the user process is created as follows:

- if the MAP_INSERT_SUBSCRIBER_DATA indication is received, the process INS_SUBS_DATA_VLR is created;

- if the MAP_DELETE_SUBSCRIBER_DATA indication is received, the process Delete_Subscriber_Data_VLR is created.

After creation of the user process the Co-ordinator relays the messages between the MAP_PM and the invoked process until a request or an indication for dialogue termination is received.

The Subscriber_Data_Management Co-ordinator is shown in the figure 17.1/2.

Figure 17.1/2: Process Co_Data_Management_VLR
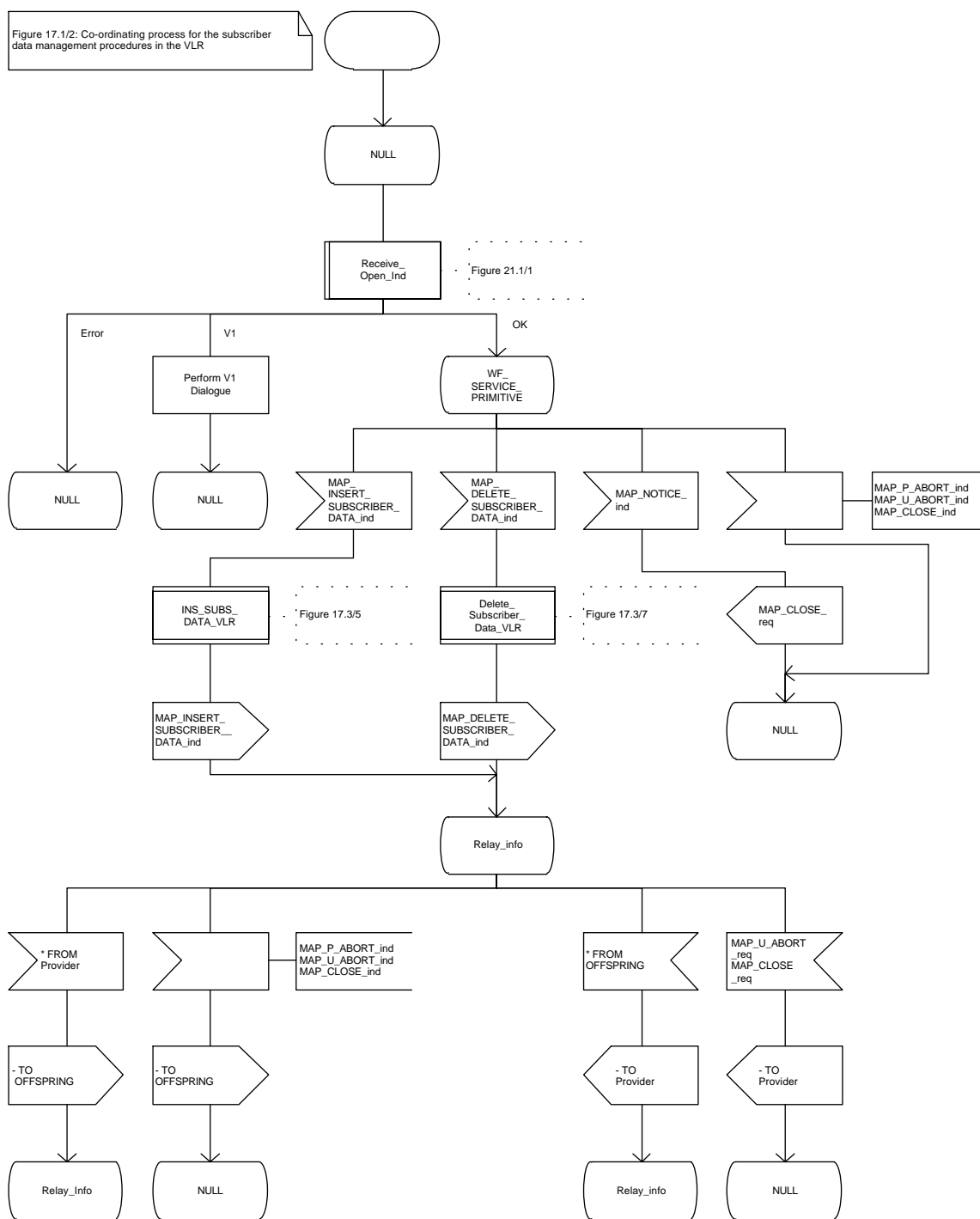
17.2 Tracing procedures

Three type of tracing procedures exist:

i)   Subscriber tracing management procedures;
ii) Subscriber tracing procedures;
iii) Event tracing procedures.

The subscriber tracing management procedures are used for management of the status and the type of the tracing. The subscriber tracing activation procedure is used at location updating or data restoration when the trace mode of a subscriber is set active in the HLR or, as a stand alone procedure, when the subscriber is already registered and the trace mode becomes active in the HLR. The procedures for providing a trace request to the VLR are shown in figures 17.2/1 and 17.2/2.
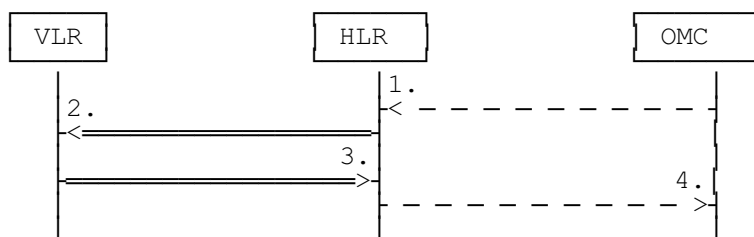
1)   Subscriber Tracing Activation
2)   MAP_ACTIVATE_TRACE_MODE
3)   MAP_ACTIVATE_TRACE_MODE_ACK
4)   Subscriber Tracing Activation Accepted

Figure 17.2/1: Stand alone subscriber tracing activation procedure
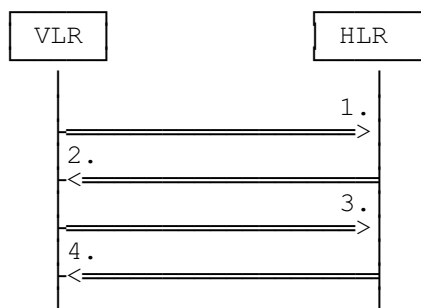
1)   MAP_UPDATE_LOCATION or MAP_RESTORE_DATA
2)   MAP_ACTIVATE_TRACE_MODE
3)   MAP_ACTIVATE_TRACE_MODE_ACK
4)   MAP_UPDATE_LOCATION_ACK or MAP_RESTORE_DATA_ACK

Figure 17.2/2: Subscriber tracing activation procedure at location updating or data restoration

The HLR sends the trace request (IMSI, trace reference, trace type and identity of the OMC) to the VLR in a MAP_ACTIVATE_TRACE_MODE request. The receipt of this primitive is acknowledged. The acknowledge primitive will indicate that the trace request is accepted by the VLR. If the request is not accepted, the reason will be reported to the HLR.

The subscriber tracing deactivation procedure is used when the trace request of a subscriber is to be cancelled in the VLR. The procedure is shown in figure 17.2/3.

1)   Subscriber Tracing Deactivation
2)   MAP_DEACTIVATE_TRACE_MODE
3)   MAP_DEACTIVATE_TRACE_MODE_ACK
4)   Subscriber Tracing Deactivation Accepted
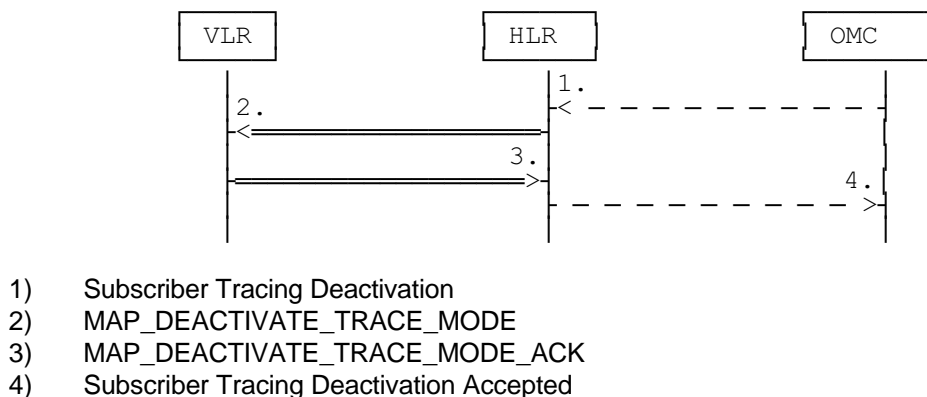
Figure 17.2/3: Subscriber tracing deactivation procedure

The HLR sends a MAP_DEACTIVATE_TRACE_MODE request to the VLR. The VLR will acknowledge the deactivation. The acknowledge primitive will indicate that the trace request has been deleted by the VLR. If the deactivation is not accepted, the reason will be reported to the HLR.

The subscriber tracing procedures are used when the VLR detects any subscriber related activity for which the trace mode is activated, e.g. receives the MAP_PROCESS_ACCESS_REQUEST indication. The procedure is shown in figure 17.2/4.

1)   MAP_PROCESS_ACCESS_REQUEST, MAP_UPDATE_LOCATION_AREA,
2)   MAP_TRACE_SUBSCRIBER_ACTIVITY
3)   Subscriber tracing information
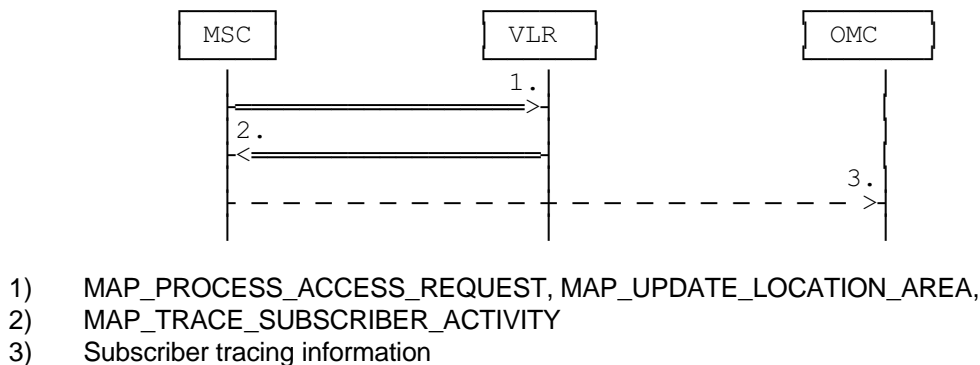
Figure 17.2/4: Subscriber tracing procedure in the servicing MSC

The VLR will generate the MAP_TRACE_SUBSCRIBER_ACTIVITY indication. The receiving MSC will send the trace record to the OMC.

[Figure numbers 17.2/5 and 17.2/6 are spare.]

17.2.1 Procedures in the HLR

17.2.1.1 Subscriber tracing activation procedure

When receiving the subscriber tracing mode activation command for a subscriber from the OMC, the HLR will activate tracing, if the subscriber is known and registered in the HLR and the subscriber is roaming in the home PLMN area. The MAP_ACTIVATE_TRACE_MODE request is sent to the VLR where the subscriber is registered.

If the MAP_ACTIVATE_TRACE_MODE confirmation is received indicating an error situation, the errors are mapped to the OMC interface. The activation request may also be repeated; the number of repeat attempts and the time in between are HLR operator options, depending on the error returned by the VLR.

If the subscriber is known in the HLR, but is deregistered or roaming outside the home PLMN area, the subscriber tracing status is activated in the HLR, but the VLR is not updated.

When receiving a request for location updating or data restoration while the subscriber trace mode is active, the macro Control_Tracing_HLR (see figure 21.9/4) shall be initiated by the location updating process in the HLR.

The subscriber tracing activation process in the HLR is shown in figure 17.2/7.

Figure 17.2/7 (sheet 1 of 2): Process ATM_HLR

Process ATM_HLR   172_71(2)

Figure 17.2/7: The subscriber tracing activation process in the HLR

- NULL → OM_Subscriber_Tracing_Activation_req
- Subscriber Known?
  - no → Set Error Unknown Subscriber
  - yes → Tracing already active?
    - yes → Set Error Tracing already active
    - No → Set Subscriber Tracing = Active in HLR
      - Deregistered Subscriber?
        - yes →
        - no → Subscriber in Home PLMN area?
          - no →
          - yes →

- NULL → NULL

- Set Subscriber Tracing = Not Active in VLR → OM_Subscriber_Tracing_Activation_cnf → NULL

- 1 Page 1
  - Repeat attempt?
    - yes → Increment attempt counter → Repeat attempt delay →
    - no → Set subscriber tracing = deactive in VLR → OM_Subscriber_Tracing_Activation_cnf → Set subscriber tracing activation failure → NULL

- MAP_OPEN_req / MAP_ACTIVATE_TRACE_MODE_req / MAP_DELIMITER → Receive_Open_Cnf
  - Error → 1 Page 1
  - V1 → Perform_MAP_V1_Dialogue → NULL
  - OK → WF_ATM_RESULT

Figure 17.2/7 (sheet 2 of 2): Process ATM_HLR

Figure 17.2/7: The subscriber tracing activation process in the HLR
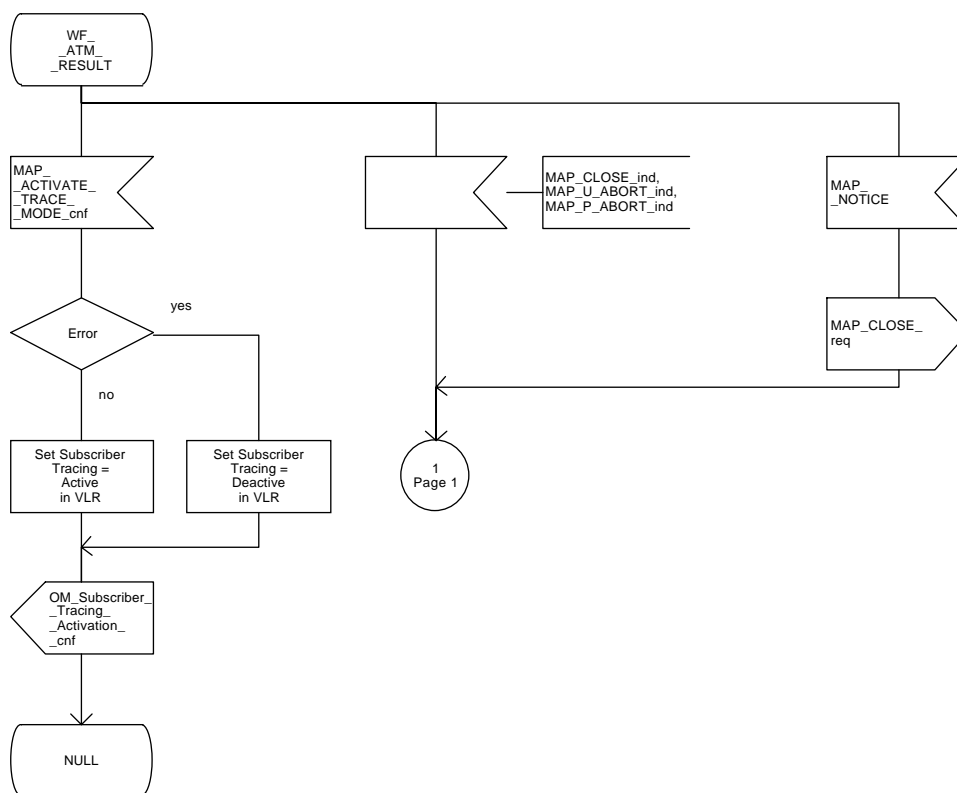
17.2.1.2 Subscriber tracing deactivation procedure

When receiving the subscriber trace mode deactivation command for a subscriber from the OMC, the HLR will send the MAP_DEACTIVATE_TRACE_MODE request to the VLR where the subscriber is registered, if the trace mode activation has been carried out. The subscriber tracing in HLR is set to a deactive state.

If the operation is successful, the HLR will set the subscriber tracing in VLR to a deactive state.

If the MAP_DEACTIVATE_TRACE_MODE confirmation is received indicating an error situation, the errors are mapped to the OMC interface. The deactivation request may be also repeated; the number of repeat attempts and the time in between are HLR operator options, depending on the error returned by the VLR.

The subscriber tracing deactivation procedure is shown in figure 17.2/8.

Figure 17.2/8 (sheet 1 of 2): Process DTM_HLR

Figure 17.2/8: The subscriber tracing deactivation process in the HLR
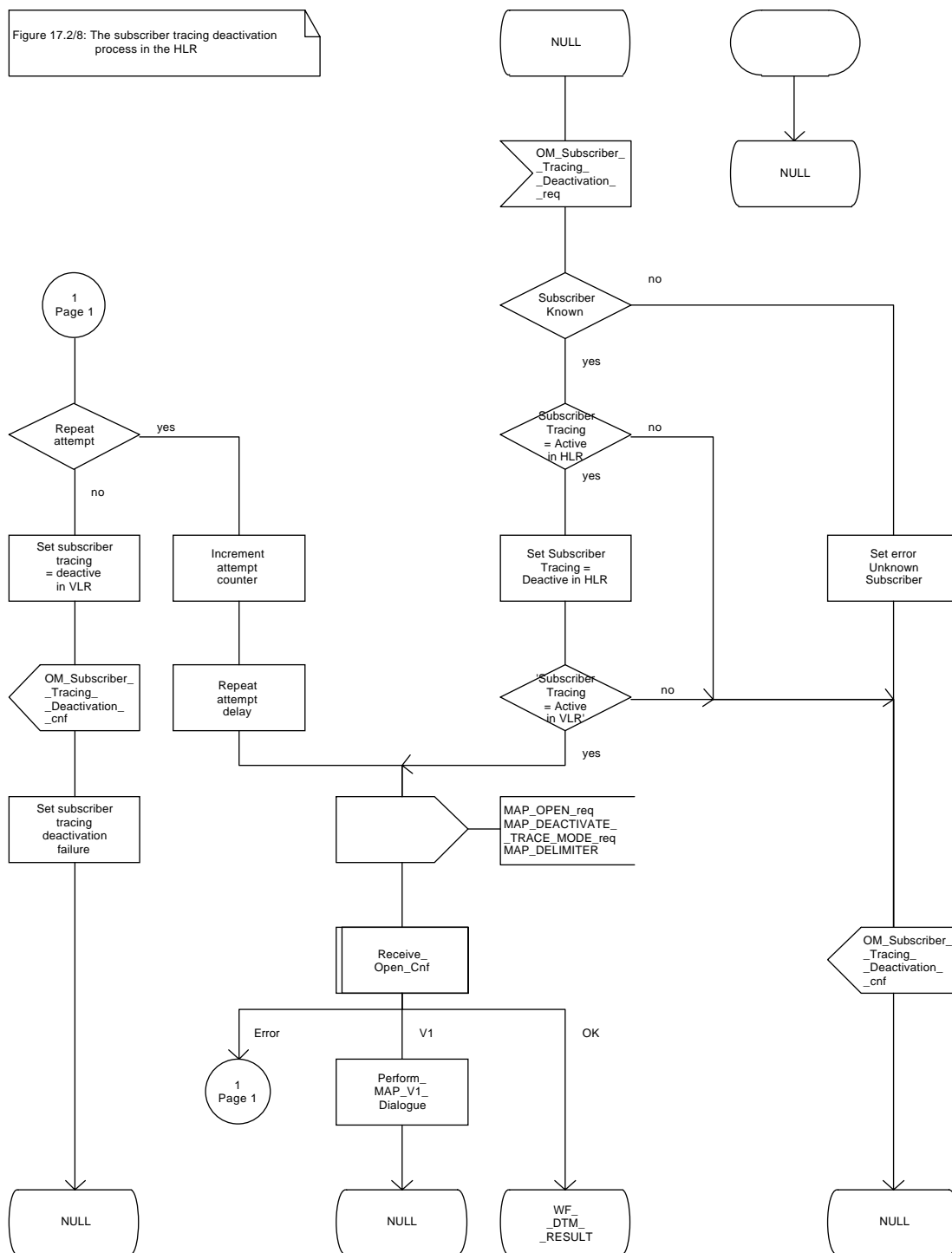
Figure 17.2/8 (sheet 2 of 2): Process DTM_HLR

Figure 17.2/8: The subscriber tracing deactivation process in the HLR
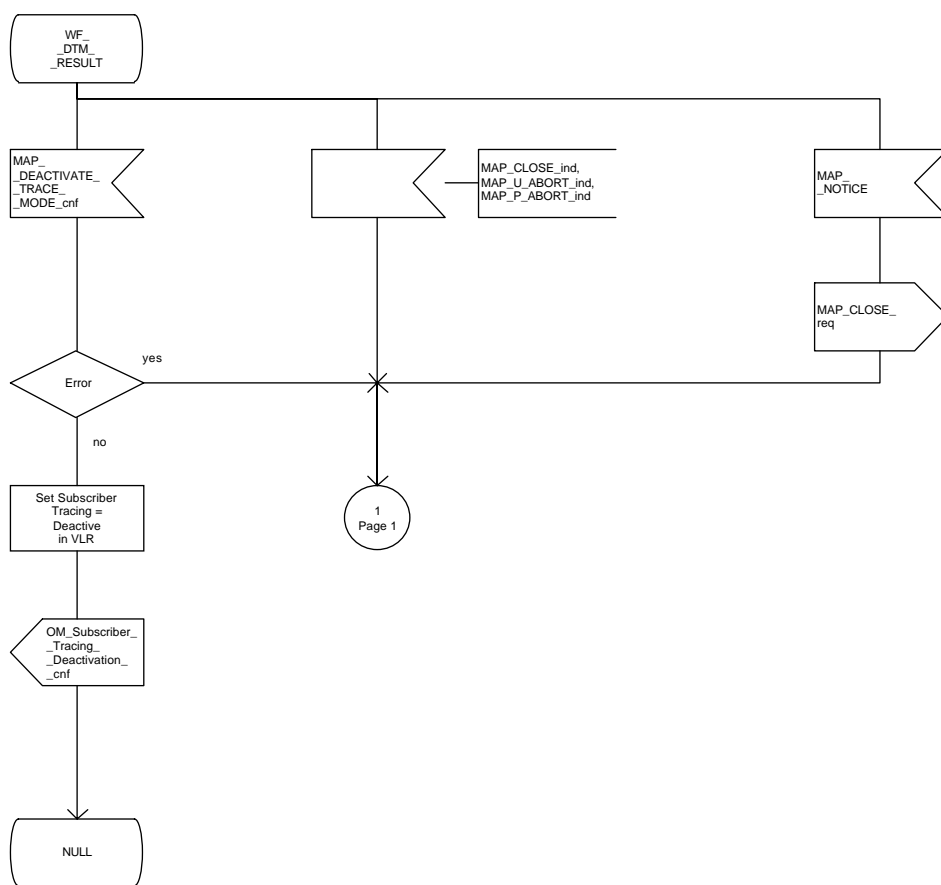
17.2.2 Procedures in the VLR

The VLR is involved in the following tracing procedures:

i)   Subscriber tracing activation procedure;
ii)   Subscriber tracing deactivation procedure;
iii) Subscriber tracing procedure.

17.2.2.1 Subscriber tracing activation procedure

When receiving a MAP_ACTIVATE_TRACE_MODE indication, the VLR will check the parameters and data in the primitive. Data errors are reported as an unexpected data value error or as a data missing error depending on the nature of the error.

If the subscriber is known, the tracing facility is supported and the tracing capacity is not exceeded, the successful report is sent in the MAP_ACTIVATE_TRACE_MODE response primitive.

The MAP_ACTIVATE_TRACE_MODE indication primitive may be received during a location updating or data restoration procedure, so the location updating or restore data process shall use the macro Activate_Tracing_VLR (see figure 21.9/3).

The subscriber tracing activation process in the VLR is shown in figure 17.2/9.

Figure 17.2/9: Process ATM_VLR_Standalone

17.2.2.2   Subscriber tracing deactivation procedure

When receiving a MAP_DEACTIVATE_TRACE_MODE indication, the VLR will check the parameters and data in the primitive. Data errors are reported as an unexpected data value error or as a data missing error depending on the nature of the error.

If the subscriber is known and the tracing facility is supported, the successful report is sent in the MAP_DEACTIVATE_TRACE_MODE response primitive.

The subscriber tracing deactivation procedure in the VLR is shown in figure 17.2/10.

Figure 17.2/10: Process DTM_VLR_Standalone

Figure 17.2/10: The subscriber tracing deactivation process in the VLR
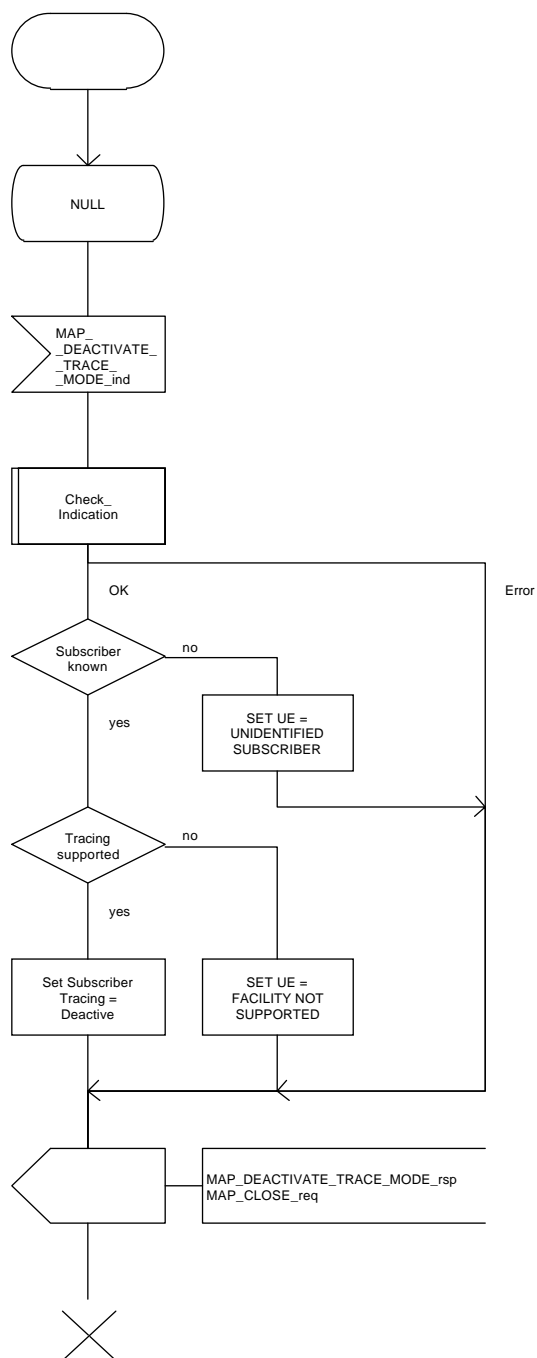
17.2.2.3   Subscriber tracing procedure

When the VLR receives a MAP_PROCESS_ACCESS_REQUEST, MAP_UPDATE_LOCATION_AREA or MAP_IMSI_ATTACH indication related to any subscriber activity from the MSC, the subscriber tracing procedure may be carried out. The macro Trace_Subscriber_Activity_VLR is shown in figure 21.9/2.

17.2.3   Procedures in the MSC

The MSC is involved in the following tracing procedure:

i)   Subscriber tracing procedure.

17.2.3.1   Subscriber tracing procedure

When receiving the MAP_TRACE_SUBSCRIBER_ACTIVITY indication from the VLR, the MSC stores trace reference, trace type and the identity of the OMC in charge of the trace, and the MSC starts to collect the trace information. The MSC will send the trace record to the OMC.

The macro Trace_Subscriber_Activity_MSC is shown in figure 21.9/1.

17.3   Subscriber data management procedures

Two types of subscriber data management procedures exist in the Mobile Application Part

i)   Subscriber Deletion;
ii)   Subscriber Data Modification.

No requirements have been identified for the Subscriber creation and subscriber data interrogation procedures.

The subscriber deletion and subscriber data modification procedures are initiated by the OMC (see figures 17.3/1 and 17.3/2).

1)   Delete Subscriber
2)   MAP_CANCEL_LOCATION
3)   MAP_CANCEL_LOCATION_ACK
4)   Subscriber Deleted
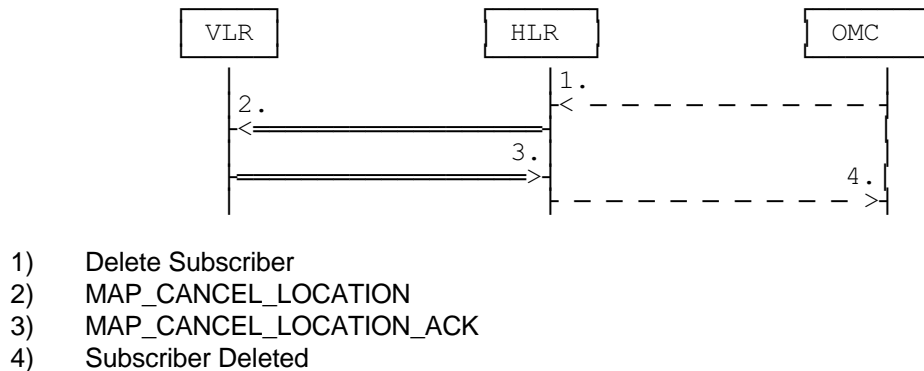
Figure 17.3/1: Subscriber deletion procedure

In the subscriber deletion procedure the subscriber data should be removed from the VLR and from the HLR. The HLR uses the MAP_CANCEL_LOCATION service.

1)   Modify Subscriber Data
2)   MAP_CANCEL_LOCATION, MAP_INSERT_SUBSCRIBER_DATA or MAP_DELETE_SUBSCRIBER_DATA
3)   MAP_CANCEL_LOCATION_ACK, MAP_INSERT_SUBSCRIBER_DATA_ACK or MAP_DELETE_SUBSCRIBER_DATA_ACK
4)   Subscriber Data Modified
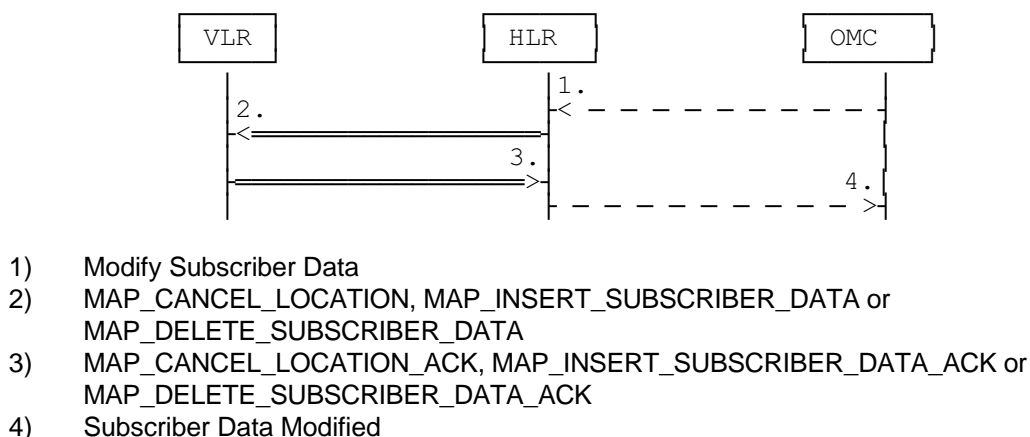
Figure 17.3/2: Subscriber data modification procedure

In the subscriber data modification procedure the subscriber data is modified in the HLR and when necessary also in the VLR. The HLR initiates either the MAP_INSERT_SUBSCRIBER_DATA, MAP_DELETE_SUBSCRIBER_DATA or MAP_CANCEL_LOCATION service depending on the modified data.

17.3.1   Procedures in the HLR

17.3.1.1   Subscriber deletion procedure

When the subscriber deletion request is received from the OMC, the HLR shall delete the subscriber data from the HLR and initiate the MAP_CANCEL_LOCATION request to the VLR where the subscriber is registered.

The subscriber deletion procedure in the HLR is shown in the figure 17.3/3.

Figure 17.3/3: Process Delete_Subscriber_HLR

Figure 17.3/3: The subscriber deletion process in the HLR

NULL

NULL

OM_Delete_
Subscriber
_req

NULL

Subscriber
Known — no

yes

Subscriber
Registered
in a VLR — no

Set error
Unknown
Subscriber

yes

Cancel_
Location_
HLR

Figure 16.1.2/2

Delete
Subscriber
Data

OM_
Subscriber
Deletion
_cnf

NULL
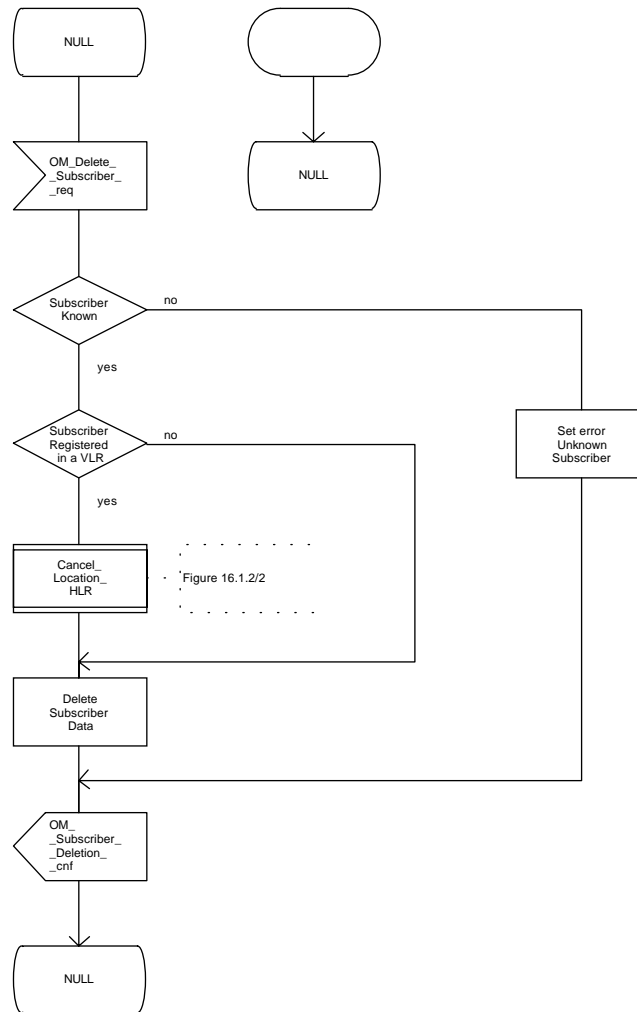
17.3.1.2 Subscriber data modification procedure

The OMC can modify the subscriber data in several different ways. The modifications can be categorized in five groups:

a) HLR internal modification, no effect in the VLR;

b) data shall be modified both in the HLR and VLR;

c) withdrawal of a basic service or a supplementary service;

d) modification affects on the roaming of the subscriber and the subscriber shall be removed from the VLR data base;

e) authentication algorithm or authentication key of the subscriber is modified.

In case "b" the MAP_INSERT_SUBSCRIBER_DATA service is initiated in the HLR.

In case "c" the MAP_DELETE_SUBSCRIBER_DATA service is initiated in the HLR.

In cases "d" and "e" the MAP_CANCEL_LOCATION service is initiated in the HLR.

If the result of a primitive received from the VLR is unsuccessful, the HLR may initiate re-attempts; the number of repeat attempts and the time in between are HLR operator options, depending on the error returned by the VLR.

The subscriber data modification procedure in the HLR is shown in the figures 17.3/4, 17.3/5 and 21.7/2.

Figure 17.3/4: Process Modify_Data_HLR
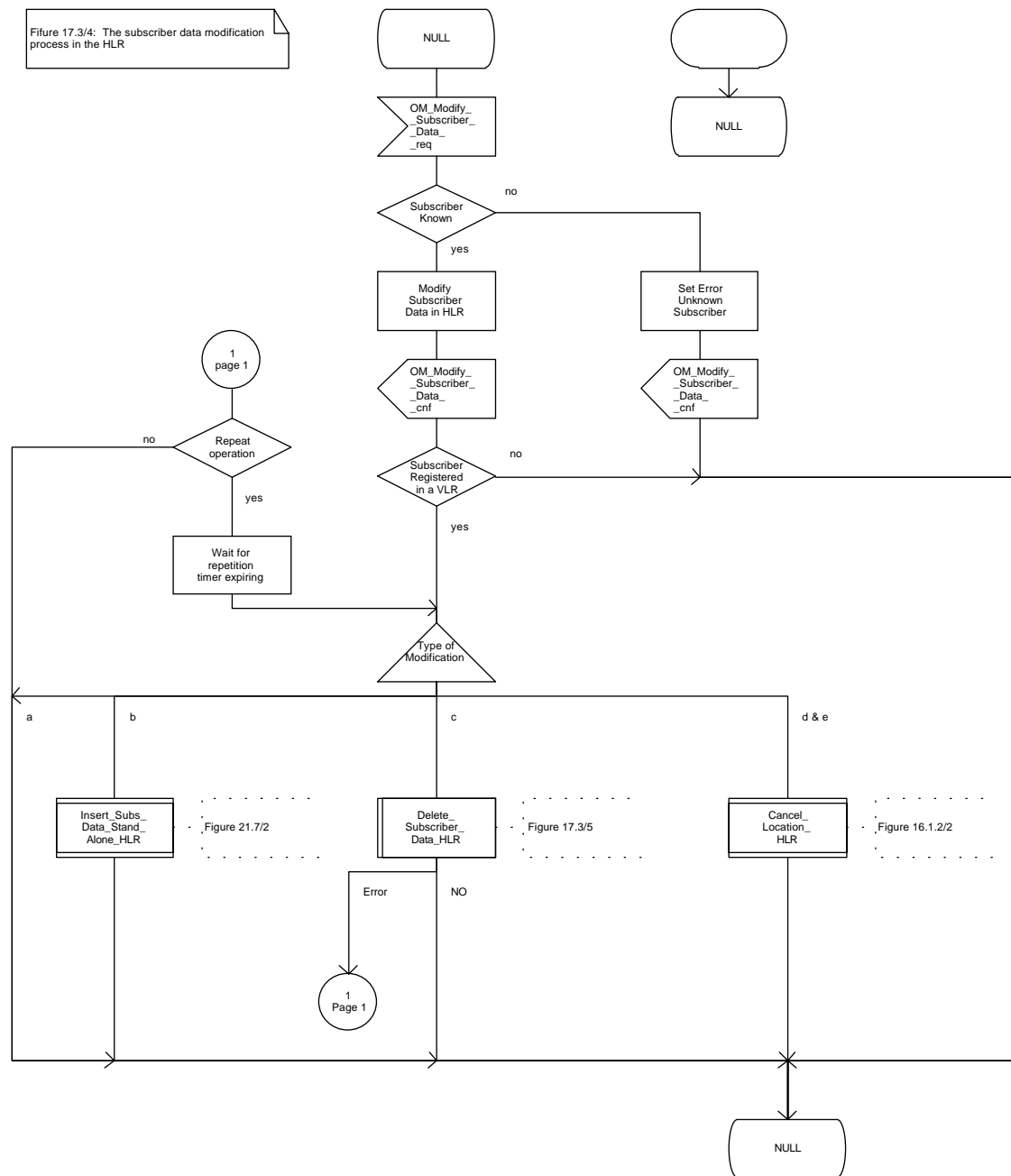
Figure 17.3/5: Macro Delete_Subscriber_Data_HLR

Figure 17.3/5: The delete subscriber data macro in the HLR
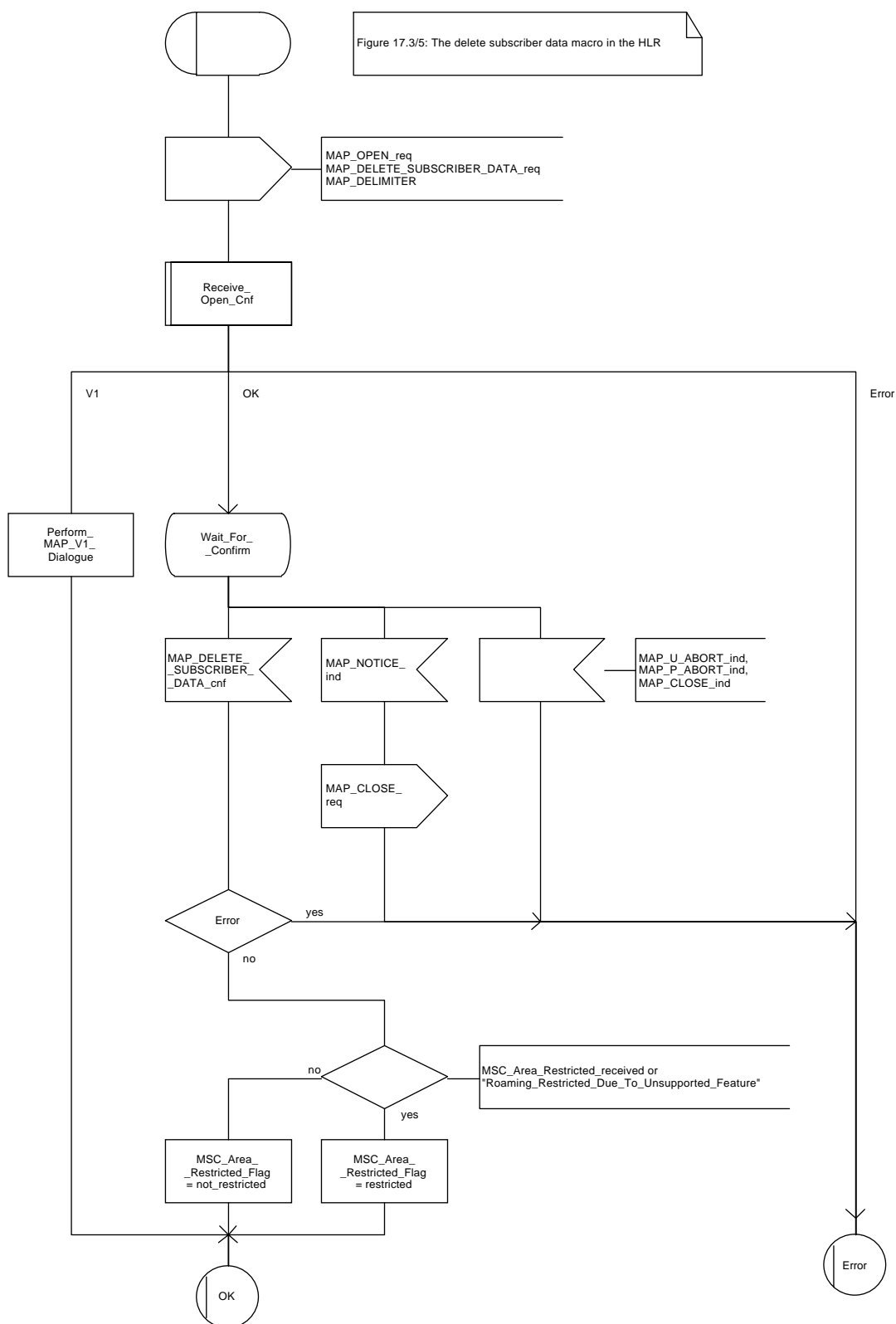
17.3.2 Procedures in the VLR

17.3.2.1 Subscriber deletion procedure

The subscriber deletion procedure in the VLR is described in the subclause 16.1.

17.3.2.2 Subscriber data modification procedure

When receiving either the MAP_INSERT_SUBSCRIBER_DATA indication or the MAP_DELETE_SUBSCRIBER_DATA indication, the VLR check the parameters and data in the primitive. Data errors are reported as an unexpected data value error or a data missing error depending on the nature of the error.

After receiving the first MAP_INSERT_SUBSCRIBER_DATA indication, the VLR will check the IMSI that is included in the primitive. If the IMSI is unknown, the error "Unidentified subscriber" is returned.

If the VLR does not support received basic or supplementary services or the network feature Operator Determined Barring, or there is a problem with Regional Subscription Data then it reports it to the HLR.

If the entire MSC area is restricted due to regional subscription, this is reported to the HLR.

If the updating of the subscriber data is not possible, the VLR will initiate the MAP_U_ABORT request primitive. If the updating is successful, the MAP_CLOSE indication is received from the HLR.

The subscriber data modification procedure in the VLR is shown in the figures 17.3/6, 17.3/7 and 21.7/1.

Figure 17.3/6: Process INS_SUBS_DATA_VLR
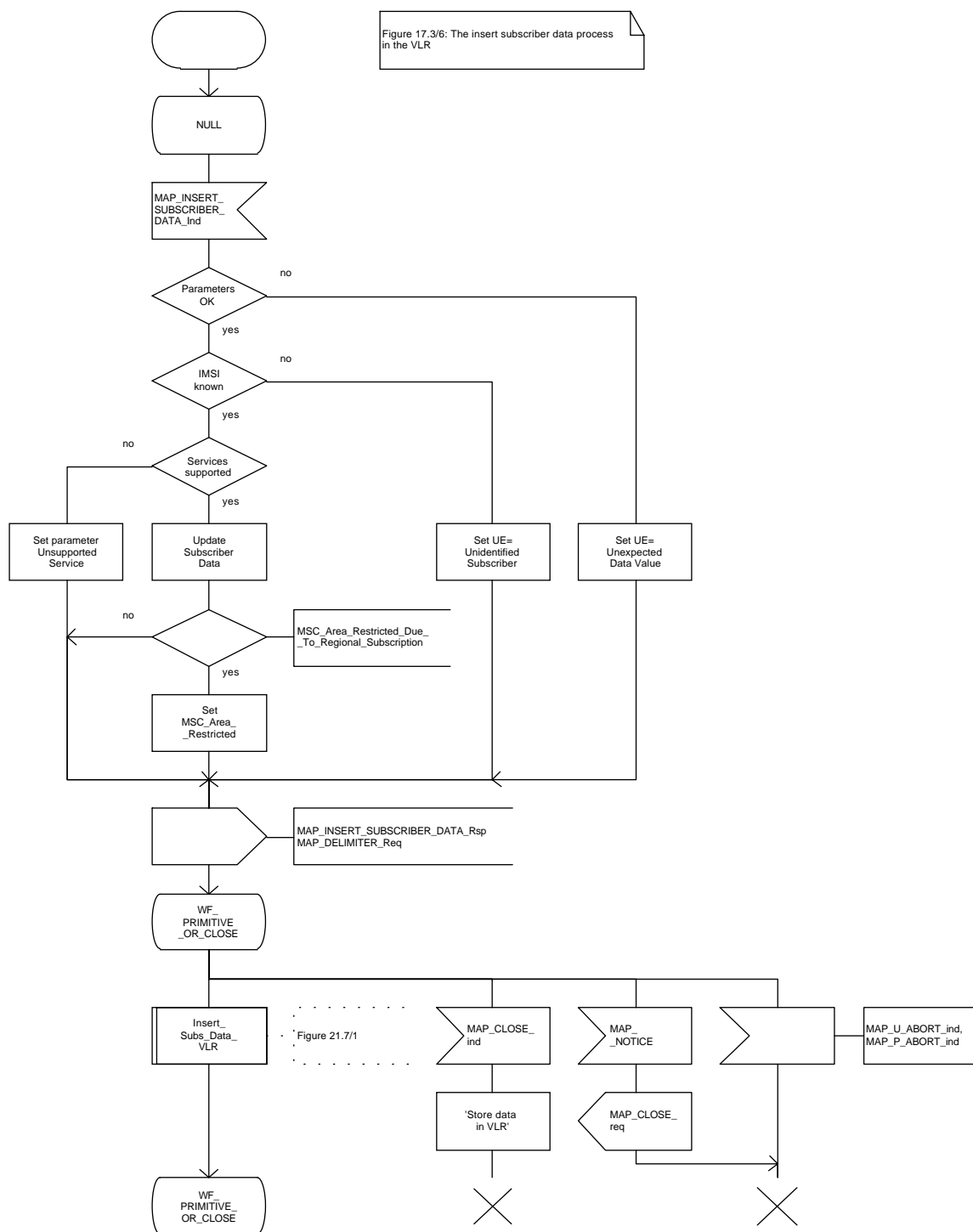
Figure 17.3/7: Process Delete_Subscriber_Data_VLR

Figure 17.3/7: The delete subscriber data process in the VLR

NULL

MAP_DELETE_
SUBSCRIBER
_DATA_ind

Check_
Indication

OK   Error

Subscriber
known   no

yes

Error   Delete
data   'SET UE =
                                        UNIDENTIFIED
OK   SUBSCRIBER'

no   MSC_Area_Restricted_Due_
            _To_Regional_Subscription

yes

Set
MSC_Area_
_Restricted

MAP_U_   MAP_DELETE_SUBSCRIBER_DATA_rsp,
ABORT_req   MAP_CLOSE_req
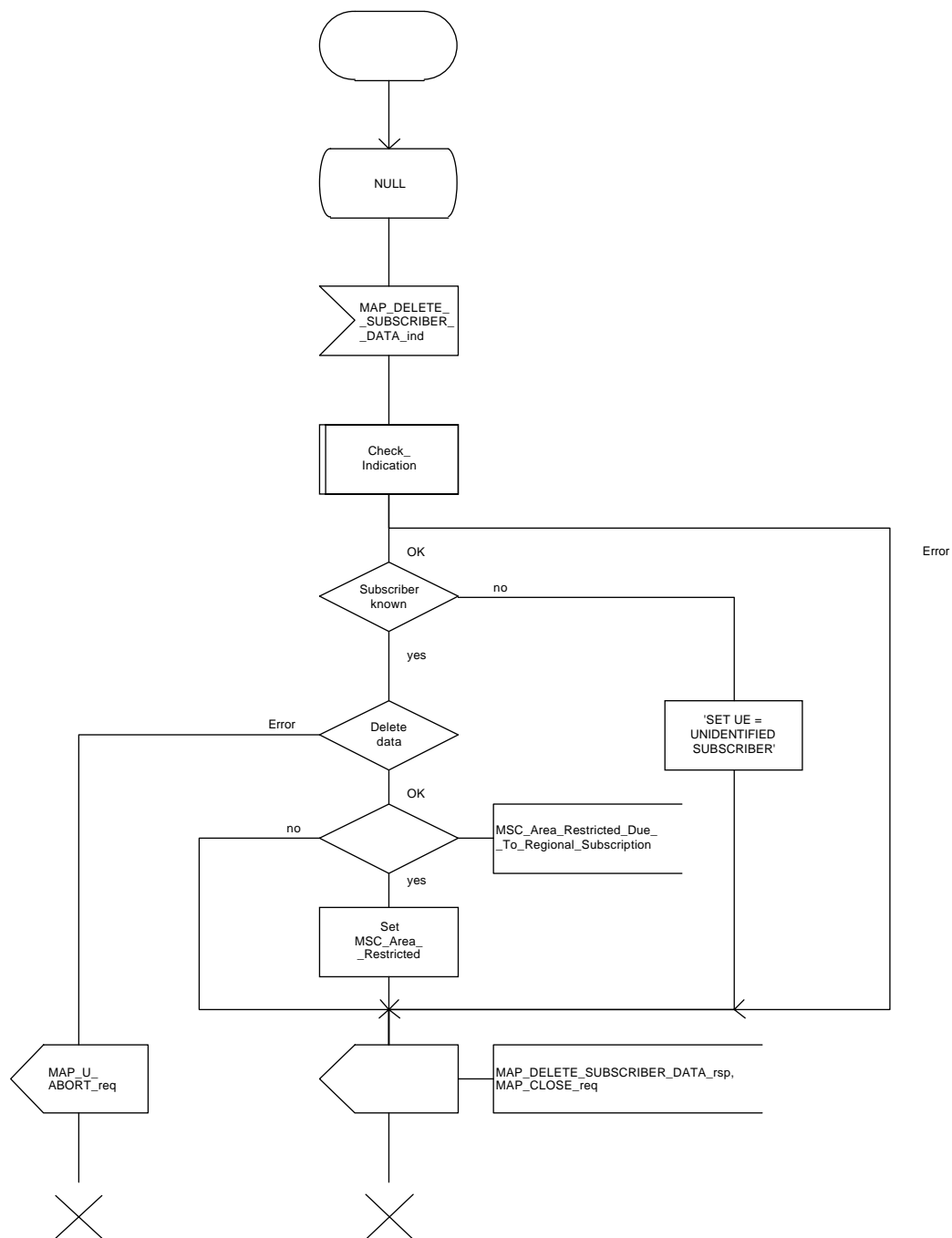
17.4 Subscriber Identity procedure

In the subscriber identity procedure the IMSI of the subscriber is retrieved from the HLR. The procedure is shown in figure 17.4/1.

```
        ┌───────┐      ┌───────┐                      ┌───────┐
        │  OMC  │      │  VLR  │                      │  HLR  │
        └───────┘      └───────┘                      └───────┘
            │      1.  │                                   │
            │- - - - - >│                                  │
            │          │         2.                        │
            │          │═════════════════════════════════>│
            │          │         3.                        │
            │    4.    │<═════════════════════════════════ │
            │< - - - - │                                   │
            │          │                                   │
```

1) Identity request
2) MAP_SEND_IMSI
3) MAP_SEND_IMSI_ACK
4) Identity confirm
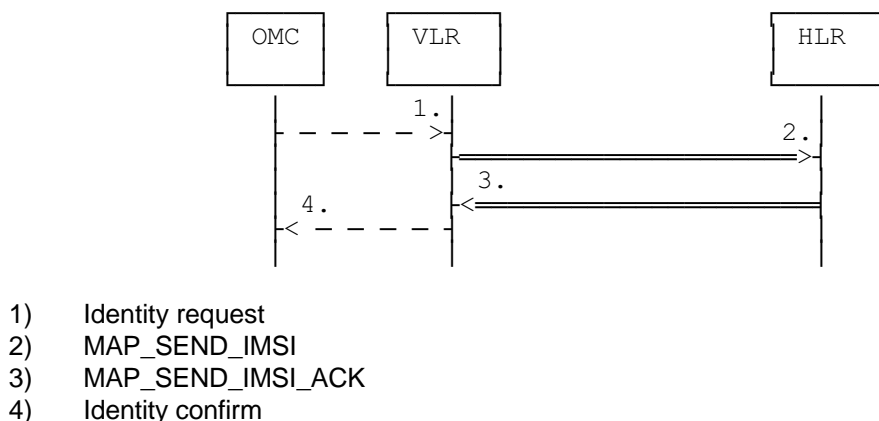
Figure 17.4/1: The subscriber identity procedure

17.4.1 Subscriber identity procedure in the HLR

Opening of the dialogue is described in the macro Receive_Open_Ind in subclause 21.1, with outcomes:

- reversion to version one procedure;

- procedure termination; or

- dialogue acceptance, with proceeding as below.

When receiving the MAP_SEND_IMSI indication, the HLR will check the parameters and data in the primitive. Data errors are reported as an unexpected data value error or a data missing error depending on the nature of the error.

If the subscriber is known in the HLR, the IMSI is fetched from the database and sent to the VLR. If the MSISDN cannot be identified, unknown subscriber indication is passed to the VLR.

The subscriber identity procedure in the HLR is shown in figure 17.4/2.

Figure 17.4/2: Process Send_IMSI_HLR

Figure 17.4/2: The send IMSI process in the HLR
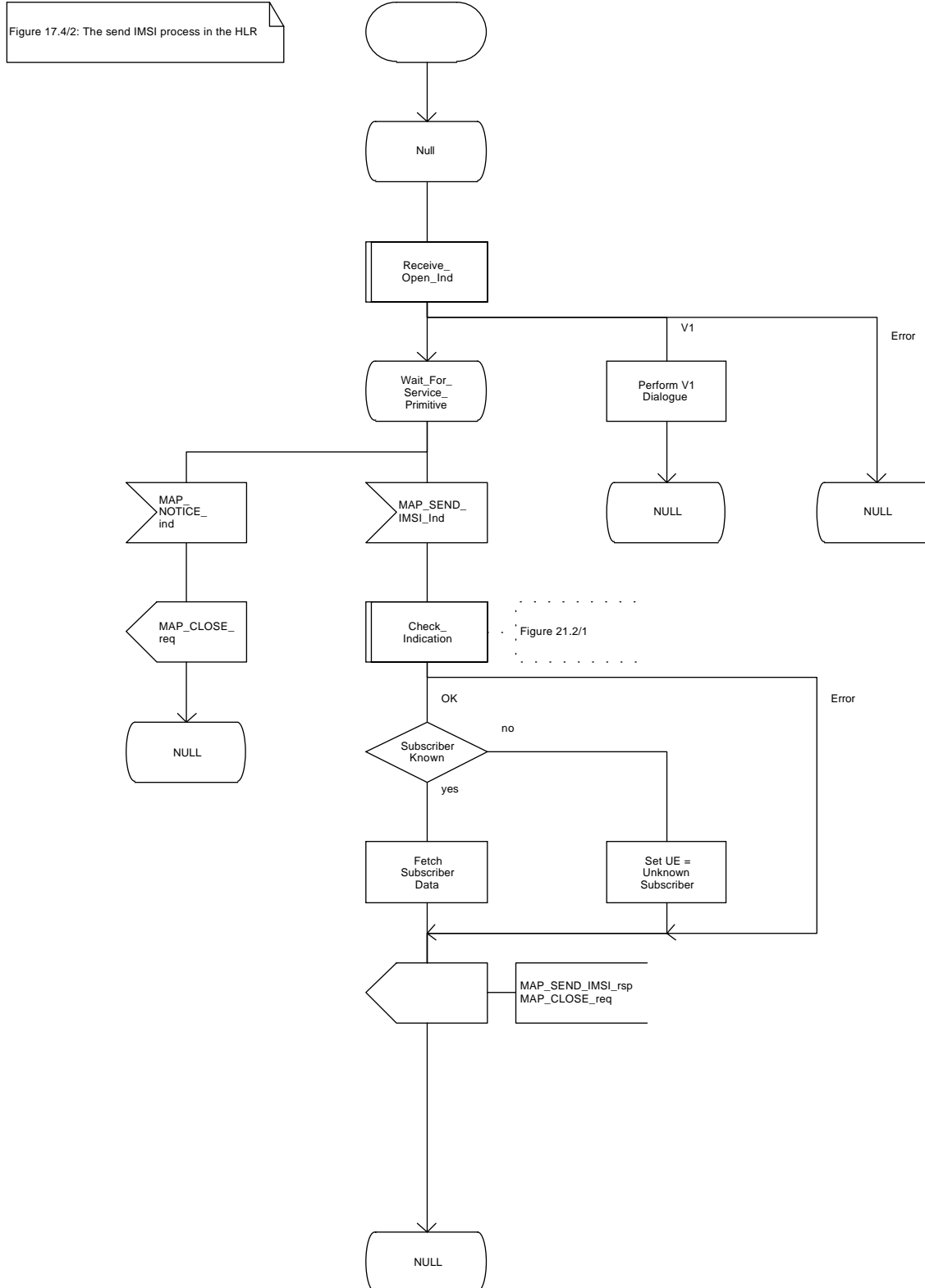
17.4.2   Subscriber identity procedure in the VLR

When the IMSI request is received from the OMC, the VLR will send the MAP_SEND_IMSI request to the HLR. The contents of the response is sent to the OMC.

The subscriber identity procedure in the VLR is shown in figure 17.4/3.

Figure 17.4/3: Process Send_IMSI_VLR
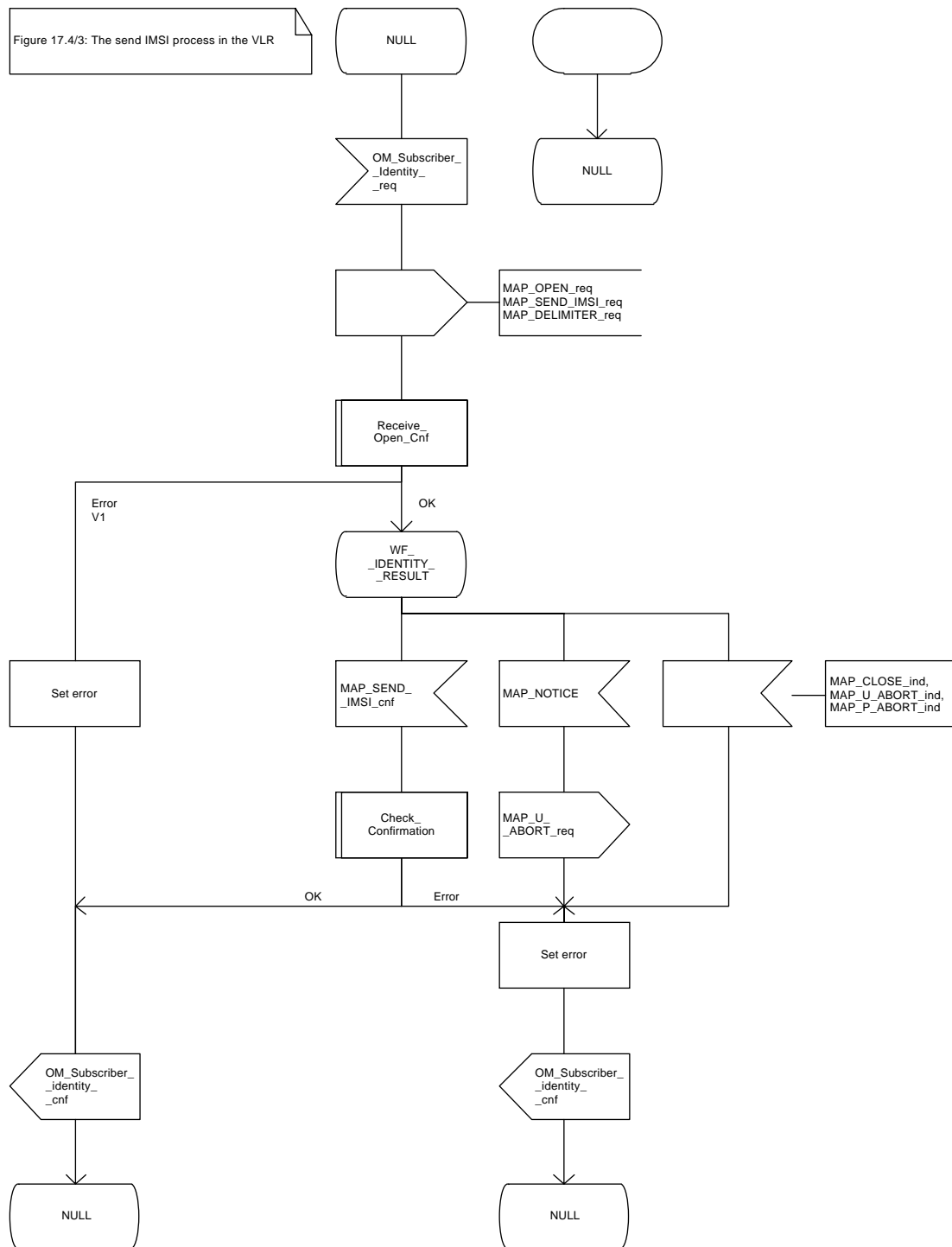
18 Call handling procedures

18.1 General

The Call Handling Procedures are used for the following:

i)   for MS terminating or MS originating calls where the MSC (acting as a VMSC) can obtain the required information directly from the visitor location register;

ii)   for MS terminating calls where the MSC (acting as gateway MSC) has to interrogate the home location register in order to obtain routing information. The HLR must retrieve the routing information from the VLR.

To provide a complete description of the Call Handling procedures, reference is made to the associated Call Handling Procedures in the BSS, MS and Fixed Network. These are only shown to indicate the interworking possibilities. Where inconsistency exists then the relevant procedures described in the following recommendations take precedence:

BSS to MSC procedures   GSM 08.08;
MSC to MS procedures   GSM 04.08;
ISUP procedures   CCITT Q.761-764 (1988);
TUP procedures   CCITT Q.721-725 (1988).

It should be noted that although the call handling procedures for TUP and ISUP are referenced, other national/proprietary user parts may be used in the fixed network.

The Call Handling co-ordination procedures are described in figures 18.1/1, 18.1/2 and 18.1/3. These procedures perform a co-ordination and distribution function within the MAP User for Call Handling.

The MSC (see figure 18.1/1) process activates the appropriate process depending on the message received and the function performed by the MSC (either MSC or Gateway MSC). The MSC can receive only one message from the MS, a CM Service Request (A_CM_SERV_REQ) for an outgoing call. Receipt of this message will result in the Outgoing Call procedures being activated, this procedure is described in subclause 18.4. If the MSC is acting as a Gateway MSC then an Initial Address Message (I_IAM) received from the network will activate the Send Routing Information (SRI_MSC) procedure as described in subclause 18.2. If the MSC is not acting as a Gateway then the incoming Initial Address Message (I_IAM) will activate the incoming call set-up procedure (ICS_MSC) as described in subclause 18.3.

The Call Handling co-ordination procedures in the VLR (figure 18.1/2) and HLR (figure 18.1/3) will be activated when a MAP_OPEN_ind is received which requests one of the Call Handling Application Contexts, the Application Contexts assigned to Call Handling are described in subclause 14.2. If the Application Context requested is not a MAP Version 1 context and there is no error detected in the Open indication, the appropriate process will be activated. For further details of the process activated refer to the following subclauses:

Incoming Call Set Up   subclause 18.3;
Outgoing Call Set Up   subclause 18.4;
Provide Roaming Number subclause 18.2;
Send Routing Information   subclause 18.2.

NOTE: Where reference is made in this subclause to a supplementary service being active, the service must be active for the required basic service. If no basic service is present in a construct then all basic services are covered by the construct.

Figure 18.1/1 (sheet 1 of 2): Process CH_COORD_MSC
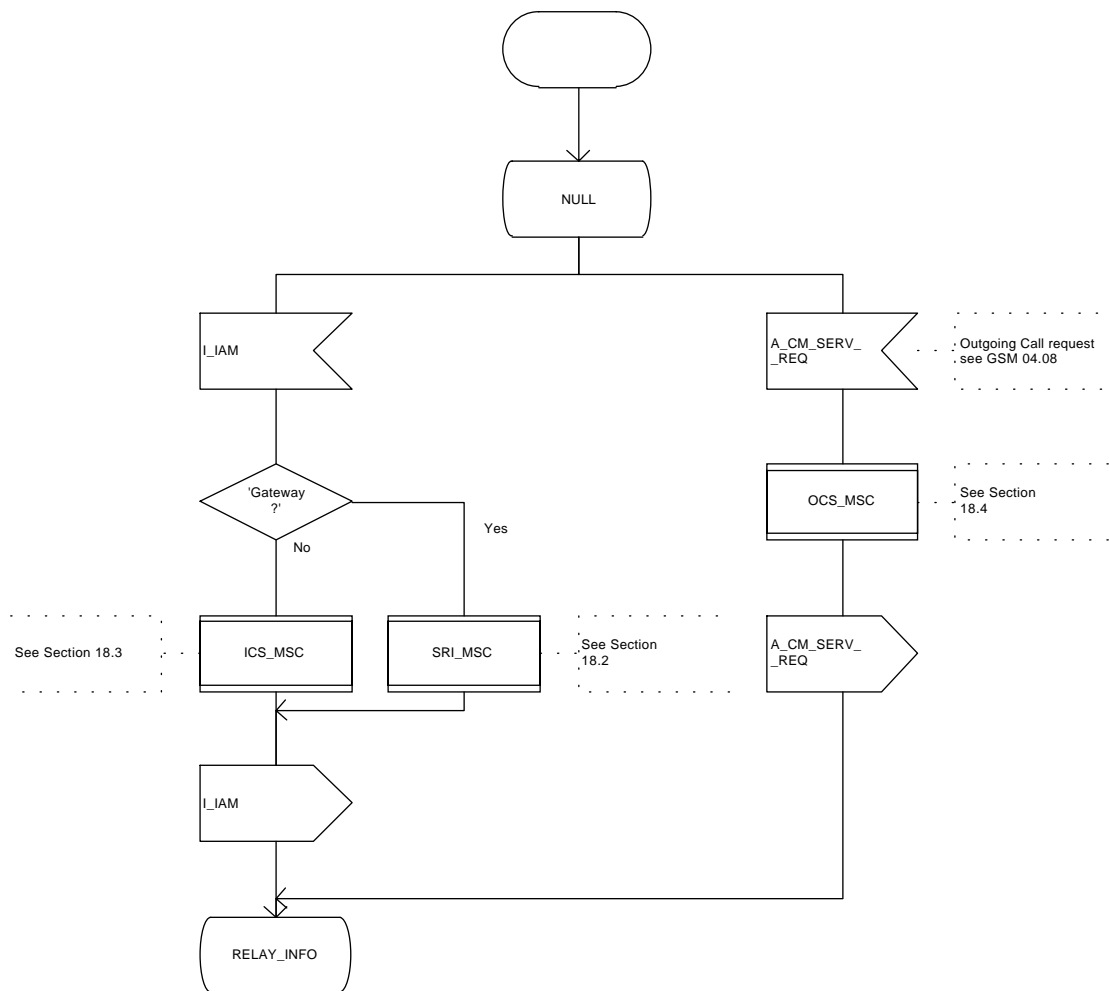
Figure 18.1/1 (sheet 2 of 2): Process CH_COORD_MSC
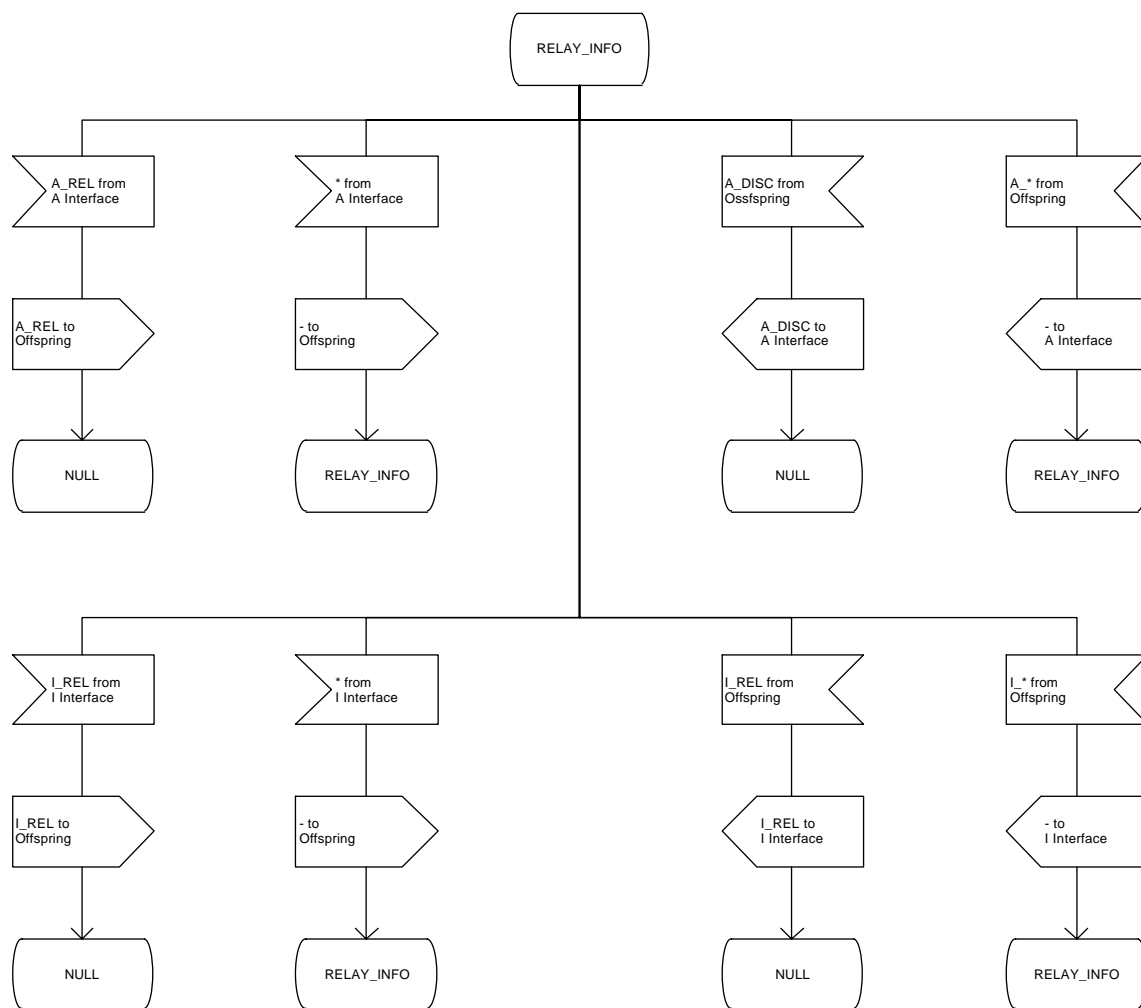
Figure 18.1/2 (sheet 1 of 2): Process CH_COORD_VLR
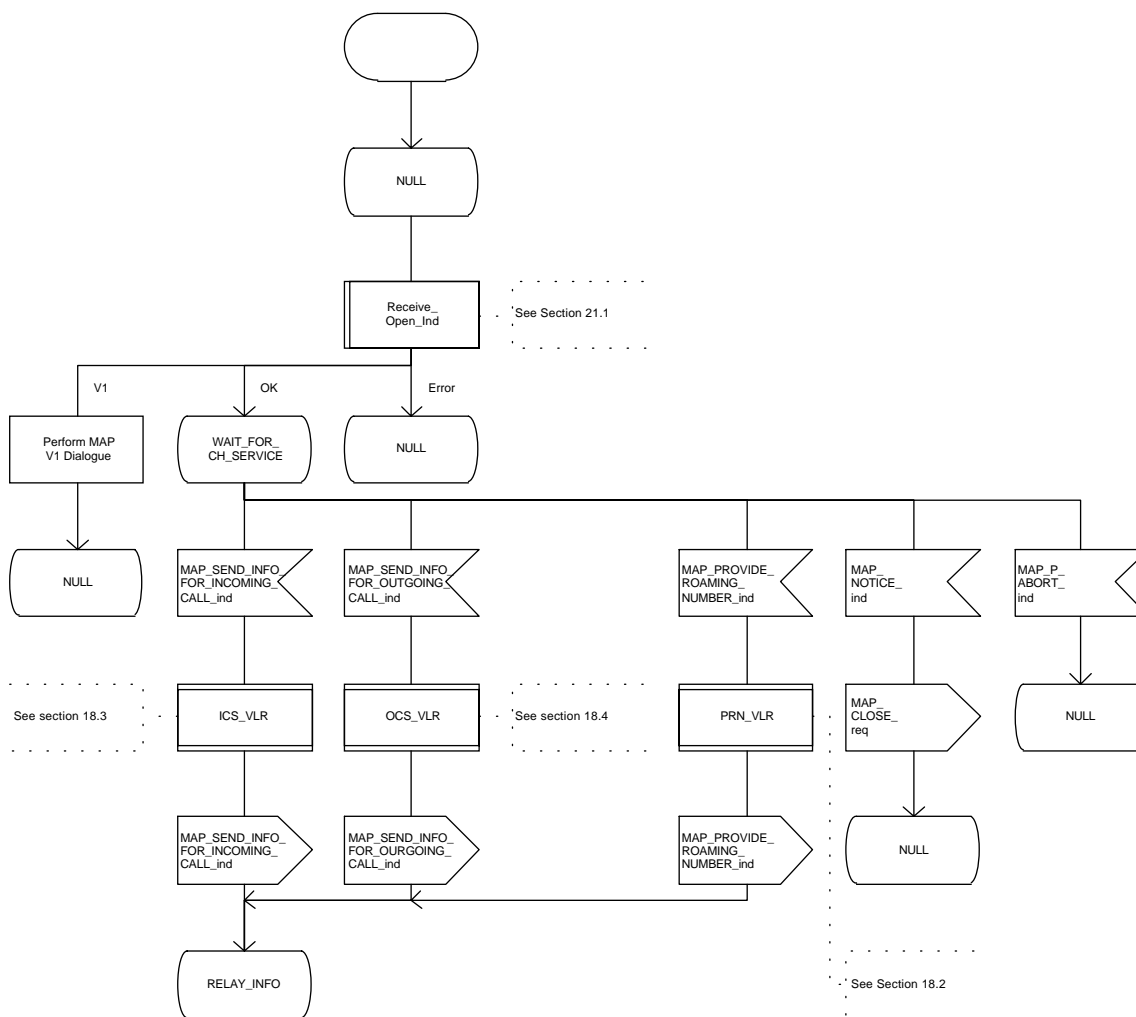
Figure 18.1/2 (sheet 2 of 2): Process CH_COORD_VLR
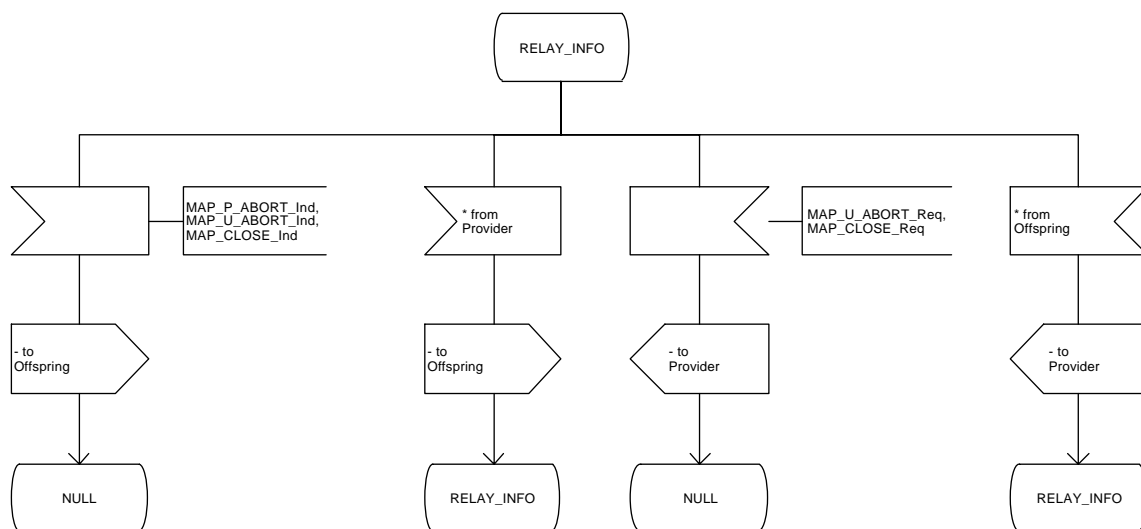
Figure 18.1/3 (sheet 1 of 2): Process CH_COORD_HLR
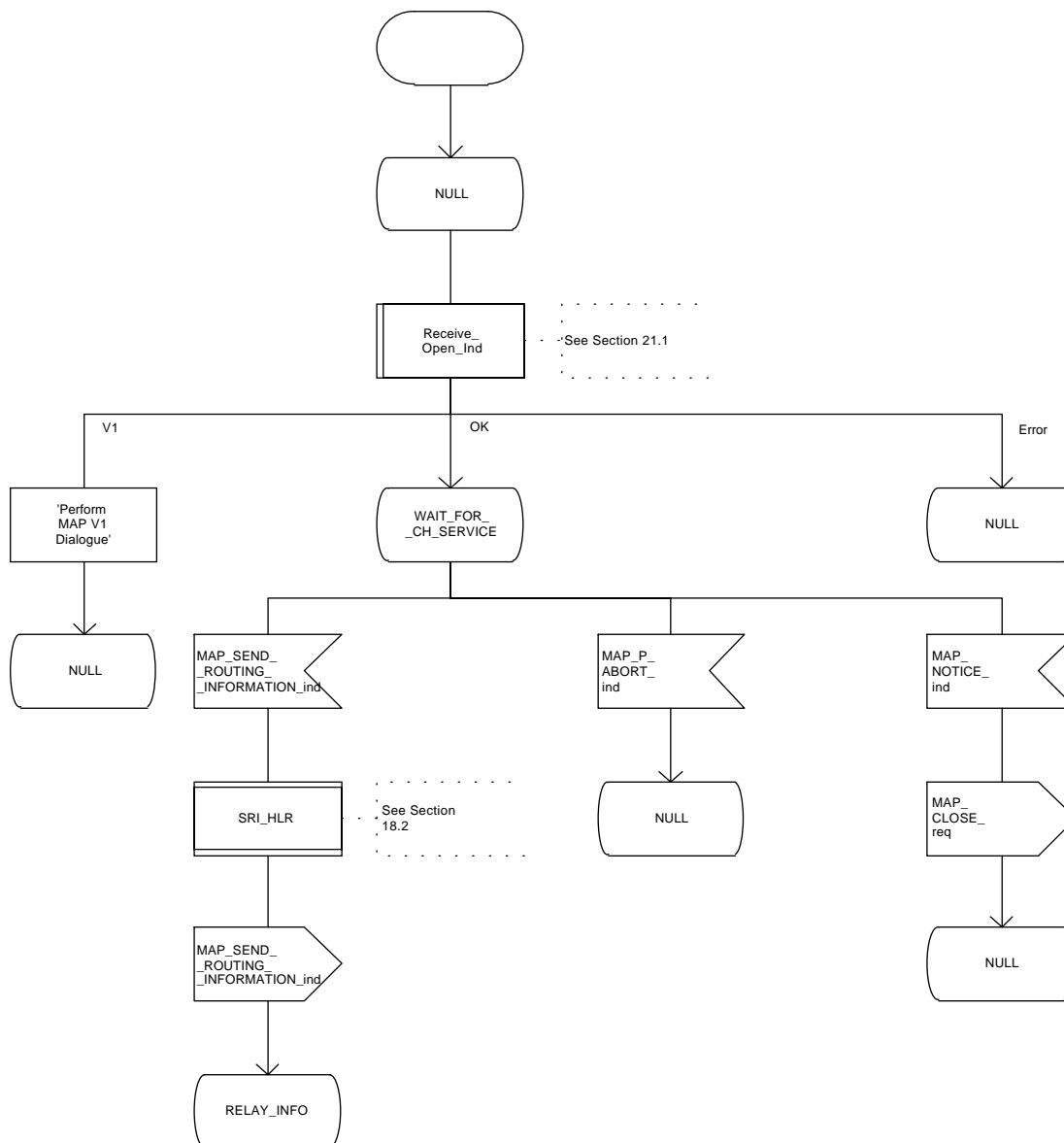
Figure 18.1/3 (sheet 2 of 2): Process CH_COORD_HLR
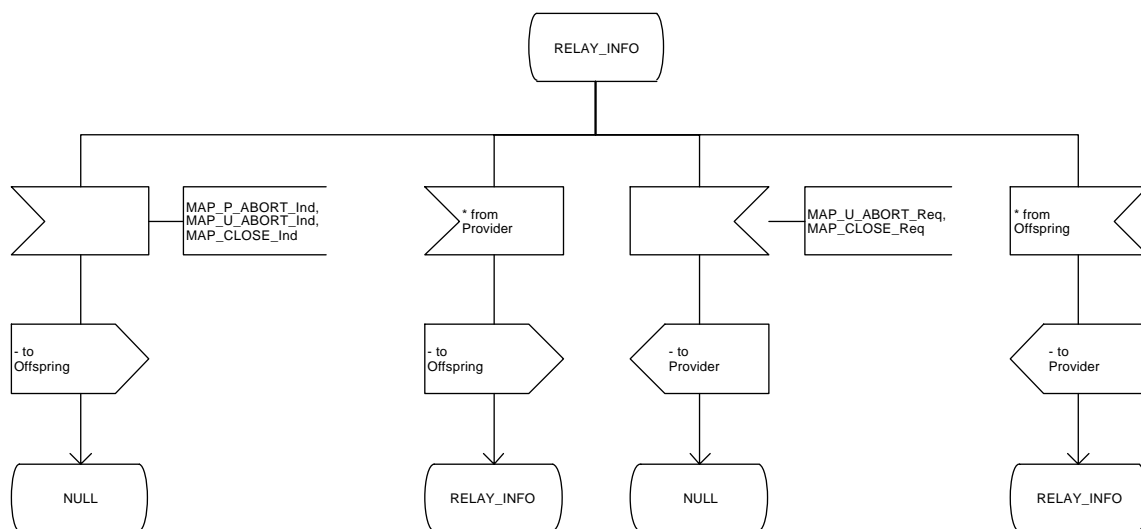
18.2 Retrieval of routing information

18.2.1 General

The procedure is shown in figure 18.2/1; the sequence only assumes successful operation.
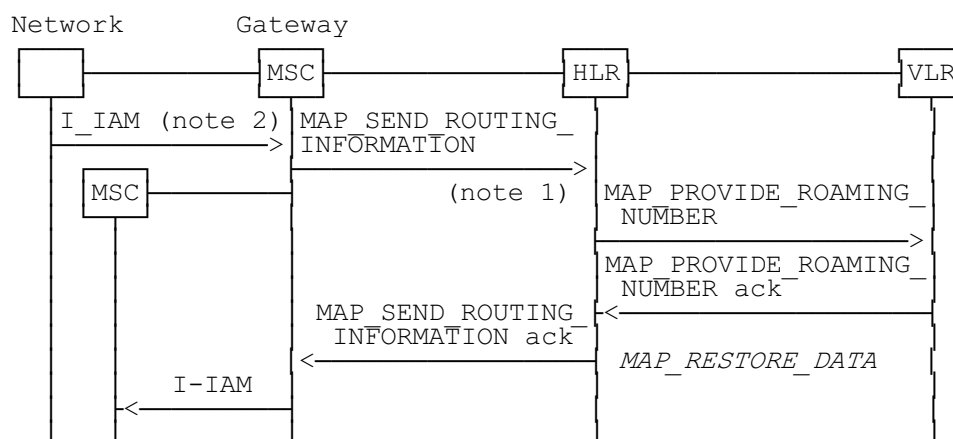
Notes:

    xxx = Optional Procedure

NOTE 1: This service may also be used by an ISDN exchange for obtaining routing information from the HLR.

NOTE 2: TUP or ISUP may be used in signalling between MSCs, depending on the network type between the MSCs. For further details on the TUP and ISUP procedures refer to the following CCITT Recommendations:

    Q.721-725 - Telephone User Part (TUP);
    Q.761-764 - ISDN User Part (ISUP).

Figure 18.2/1: Procedure for retrieval of routing information

It is assumed that signalling towards the fixed network uses the messages IAM (Initial Address Message) and REL (Release). Examples of other interworking cases are not shown in this standard (for further information see CCITT Recommendation Q.699).

The following MAP services are used to retrieve routing information:

MAP_SEND_ROUTING_INFORMATION   see subclause 8;
MAP_PROVIDE_ROAMING_NUMBER   see subclause 8;
MAP_RESTORE_DATA   see subclause 6.

18.2.2   Procedures in the MSC

The procedure is described in figure 18.2/2. These procedures make use of a number of processes and macros not described in this subclause. The description of these can be found in the following subclauses:

       Receive_Open_Cnf   see subclause 21.1;
       Check_Confirmation   see subclause 21.2.

       NOTE: If CUG Information is provided in the Initial Address Message (I_IAM) and the PLMN does not provide a CUG capability, the call will be released if the CUG does not have Outgoing Access. In this case the ISUP Cause indicator will be set to #87 (user not member of CUG) and a Release message (I_REL) returned to the originating network. If the CUG has Outgoing Access the call will be treated as an ordinary call, in this case the CUG Information in the Initial Address Message may need to be modified. See Annex E of CCITT Recommendation Q.767 (Approved 1990) for a description of the procedures for a CUG call interworking with a network which does not support the CUG facility. It has been assumed that these checks will be performed by the Gateway MSC prior to invoking MAP procedures.

Successful Outcome

The Gateway MSC will receive an Initial Address Message (I-IAM) indicating an incoming call. To be able to route the call toward the mobile subscriber, via the appropriate MSC, the Gateway MSC will initiate a MAP-SEND-ROUTING-INFORMATION service request to the HLR of the MS to receive the call.

If a routing address can be provided by the HLR, it is returned to the MSC in the MAP-SEND-ROUTING-INFORMATION service response. The routing address can take two forms: MSRN or forwarded-to-number. If a forwarded-to-number is provided, the Gateway MSC will increment the forwarding counter. The Gateway MSC will then construct an Initial Address Message (I-IAM) and send this to the MSC of the area where the MS is currently located. The procedure in the Gateway MSC is then terminated.

Data or Parameter errors

If any parameter or data errors are detected by the MSC, the procedure for this call will be terminated and a release message (I_REL) returned to the network to release the call.

Service Provider Failure

Once the dialogue between the Gateway MSC and the HLR is established a MAP_P_ABORT service indication may be received at any stage from the MAP Service Provider. This aborts the dialogue, see clause 5 for a definition of the abort reasons.

The call will be released by the MSC sending a Release message (I_REL) to the Network. The procedure in the Gateway MSC is then terminated.

Call Released by Calling Subscriber

This may occur at any stage during the call set up procedure. Call release is indicated to the MSC by a Release Message (I_REL). The procedure in the Gateway MSC is then terminated. Any response subsequently received from the HLR will be discarded.

Routing Failure

Failure to provide routing information is reported by the HLR to the Gateway MSC in the MAP_SEND_ROUTING_INFORMATION service confirm. The MAP User Error parameter will indicate one of the following conditions:

MAP User Error		
Unknown Subscriber	ISUP Error	= 1
Absent Subscriber	" "	= 18
Call Barred	" "	= 21
Forwarding Violation	" "	= 21
Number Changed		= 22
CUG Reject	" "	= 55
Teleservice/Bearer Service not provisioned		= 57
Facility Not Supported	" "	= 69
Data Missing	" "	=111
Unexpected Data Value	" "	=111
System Failure	" "	=111

> NOTE: A suggested mapping is shown above between the MAP User Error value and the ISUP Error Cause, this is only provided to aid understanding. CCITT Q.6xx-series of recommendations provide further guidance for the mapping of Error Cause values (specifically ISUP and TUP errors).

Table 18.2/1: Suggested mapping between ISUP and MAP Errors

Regardless of the reason for routing failure the Gateway MSC will construct a Release message (I_REL), selecting the appropriate Error Cause value, and return this to the network. The procedure in the Gateway MSC will then be terminated.

Figure 18.2/2 (sheet 1 of 2): Process SRI_MSC
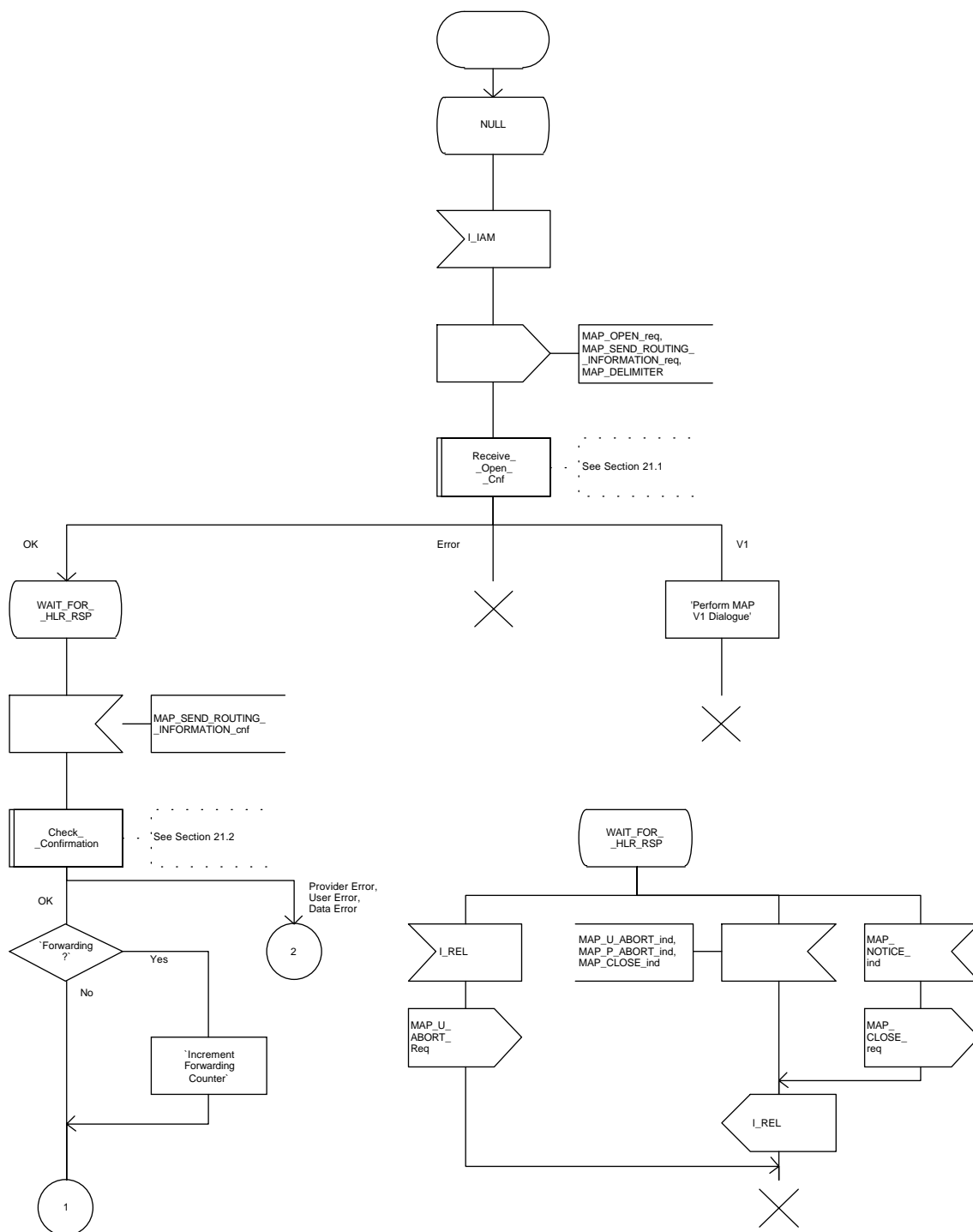
Figure 18.2/2 (sheet 2 of 2): Process SRI_MSC

```
        ( 1 )
          |
     +----------+
     | `Construct|
     |   IAM`   |
     +----------+
          |
      /---------\
     <  I_IAM    |
      \---------/
          |
          X
```

```
        ( 2 )
          |
     +----------+        . . . . . . . . . . . . . . . .
     | `Construct| . . . :ERROR CAUSE MAPPING (MAP/ISUP)
     |   REL`   |        :Refer to table 18.2/1
     +----------+        . . . . . . . . . . . . . . . .
          |
      /---------\
     <  I_REL    |
      \---------/
          |
          X
```
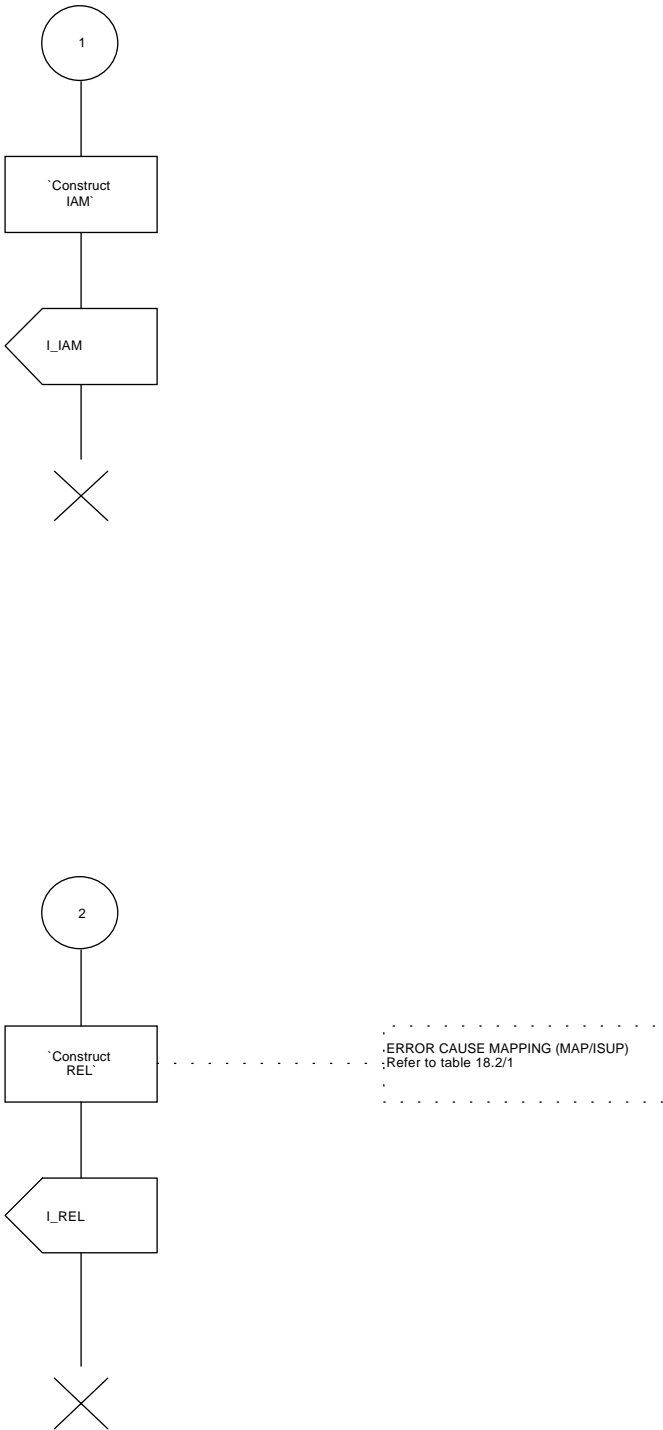
18.2.3   Procedures in the HLR

The procedure is shown in figure 18.2/3 and 18.2/4. These procedures make use of a number of processes and macros not described in this subclause. The description of these can be found in the following subclauses:

Receive_Open_Cnf	subclause 21.1;
Check_Confirmation	subclause 21.2;
Receive_Open_Ind	subclause 21.1;
Check_Indication	subclause 21.2.

Successful Outcome

When a MAP_SEND_ROUTING_INFORMATION service indication is received from the Gateway MSC, the HLR will first check for parameter or data errors. If the MS is known to the HLR, the HLR will then check if incoming calls are barred for this MS.

The HLR will then proceed with a subscription check if the Network Signal Information is provided.

If CUG Interlock and optionally, CUG Outgoing Access, information is provided by the Gateway MSC and/or the called subscriber is a CUG subscriber the HLR will then perform the mobile terminated CUG call authorisation check. The purpose of this check is to identify a match between calling and called party CUG attributes, whilst enforcing intra_CUG communication restrictions. The procedures for authorising a CUG call is described in more detail in GSM 03.85.

If Call Forwarding Unconditional service is active and the call has not been forwarded more times than allowed, and a CUG call authorisation check is not required the HLR will increment the forwarding counter and provide the routing information for forwarding this call. This information is returned to the Gateway MSC in the MAP_SEND_ROUTING_INFORMATION service response. The procedure in the HLR is then terminated.

If Call Forwarding Unconditional service is active and CUG information is provided and/or the forwarding subscriber is a CUG subscriber then the HLR will perform a CUG call authorisation check on the forwarding party's outgoing access. The procedures for authorising a CUG check with Call Forwarding interaction are described in more detail in GSM 03.85. On successful completion of the authorisation check, the HLR will increment the forwarding counter and provide the routing information for forwarding this call. The new CUG Information (i.e. output of the CUG call authorisation check) to be used on the forwarding leg of the call will also be provided. This information is returned to the Gateway MSC in the MAP_SEND_ROUTING_INFORMATION service response. The procedure in the HLR is then terminated.

If the MS is reachable, i.e. if the MS is not purged, the VLR is not restricted and the HLR has a VLR number for the MS, the HLR requests the current VLR to provide a roaming number to be used for further call routing. The HLR will request this information from the VLR by initiating the MAP_PROVIDE_ROAMING_NUMBER service. If this operation was successful, the MSRN is returned to the Gateway MSC in the MAP_SEND_ROUTING_INFORMATION service response. The procedure is then terminated in the HLR.

In the SDL diagram in figure 18.2/3, the task "Perform MAP V1 Dialogue" includes:

-        the relay of the roaming number returned from the VLR to the GMSC in the MAP_SEND_ROUTING_INFORMATION response if the outcome of the dialogue with the VLR was successful;

-        the possible invocation of call forwarding on not reachable if the VLR returned the "absent subscriber" error;

-        the appropriate mapping of an error received from the VLR into a user error in the MAP_SEND_ROUTING_INFO response;

as described in the MAP version 1 specification.

Parameter or data error

If the HLR detects any parameter or data errors in the MAP_SEND_ROUTING_INFORMATION service indication, the user error "Data Missing" or "Unexpected Data Value" is returned to the Gateway MSC. If the error is detected in the response to the HLR request for a roaming number (MAP_PROVIDE_ROAMING_NUMBER), the user error "System Failure" is returned to the Gateway MSC in the MAP_SEND_ROUTING_INFORMATION service response.

Service Provider Failure

A MAP_P_Abort service indication may be received from the MAP Service Provider. This will abort the dialogue, see clause 5 for a definition of the abort reasons. Depending on the reason for the abort and where the abort originated, the associated dialogue with the MSC may be terminated. Otherwise the procedure in the HLR is terminated.

MS Not Known

If the MS is not known to the HLR, the user error "Unknown Subscriber" is returned to the Gateway MSC. This is contained in the MAP_SEND_ROUTING_INFORMATION service response. The procedure in the HLR is then terminated.

Number Changed

If the number of the subscriber is changed inside the PLMN, the user error "Number Changed" is returned to the Gateway MSC. This is contained in the MAP_SEND_ROUTING_INFORMATION service response. The procedure in the HLR is then terminated.

Calls Barred

If the MS is barred from receiving incoming calls, the user error "Call Barred" is returned to the Gateway MSC. The MS may be barred from receiving a call due to Operator Determined Barring (as defined in GSM 02.41) in which case, the Call Barring cause "Operator Barring" is returned to the Gateway MSC. The user error is contained in the MAP_SEND_ROUTING_INFORMATION service response. The procedure in the HLR is then terminated.

Subscription Check Failure

If Network Signal Information is provided, the HLR will perform a service subscription check on this information. If this check fails, depending on the reason for failure, the user error "Bearer Service Not Provisioned" or "Teleservice Not Provisioned" is returned to the Gateway MSC. This is contained in the MAP_SEND_ROUTING_INFORMATION service response. The procedure in the HLR is then terminated.

CUG Reject

If the call fails the CUG call authorisation check (this applies to incoming access check and outgoing access if the call has been forwarded - see GSM 03.85 for conditions), the user error "CUG Reject" is returned to the Gateway MSC. This is contained in the MAP_SEND_ROUTING_INFORMATION service response. The procedure in the HLR is then terminated.

MS Not Reachable

The MS may not be reachable for the following reasons:

- there is no location information available for the MS;

- the MSC Area Restricted Flag is set, indicating that roaming in the MSC area is not allowed;

- the "MS purged" flag is set, indicating that the MS data have been purged from the VLR;

- the Provide Roaming Number procedure fails and returns the error Absent Subscriber;

- the Provide Roaming Number procedure fails and returns the error "No Roaming Number Available".

The HLR will then check if the service Call Forwarding on MS Not Reachable is active. If the service is not active the appropriate user error ("System Failure" if the VLR returned "No Roaming Number Available", otherwise "Absent Subscriber") is returned to the Gateway MSC. This is contained in the MAP_SEND_ROUTING_INFORMATION service response. The procedure in the HLR is then terminated.

If the Call Forwarding on MS Not Reachable service is active, the call has not been forwarded more times than allowed and a CUG call authorisation check is not required, the HLR will increment the forwarding counter and provide the routing information for forwarding this call.

If the Call Forwarding on MS Not Reachable service is active and CUG information is provided and/or the forwarding subscriber is a CUG subscriber then the HLR will perform a CUG call authorisation check on the forwarding party's outgoing access. The procedures for authorising a CUG check with Call Forwarding interaction are described in more detail in GSM 03.85. On successful completion of the authorisation check, the HLR will increment the forwarding counter and provide the routing information for forwarding this call. The new CUG Information (i.e. output of the CUG call authorisation check) to be used on the forwarding leg of the call will also be provided.

This forwarding information is returned to the Gateway MSC in the MAP_SEND_ROUTING_INFORMATION service response. The procedure in the HLR is then terminated.

Forwarding Violation

If a Call Forwarding service is active, but the call has already been forwarded as many times as allowed, the user error "Forwarding Violation" is returned to the Gateway MSC. This is contained in the MAP_SEND_ROUTING_INFORMATION service response. The procedure in the HLR is then terminated.

No Roaming Number Allocated

The HLR request to the VLR to provide a roaming number may fail for a number of reasons (see subclause 8.4). If the reason is "Facility Not Supported", the user error is set to "Facility Not Supported"; otherwise, if the reason for failure is anything other than "Absent Subscriber" or "No Roaming Number Available", the user error is set to "System Failure". The error is returned to the Gateway MSC in the MAP_SEND_ROUTING_INFORMATION service response. The procedure in the HLR is then terminated.

If the reason for failure is "Absent Subscriber" or "No Roaming Number Available", then the procedure continues as described in the MS Not Reachable subclause above.

Figure 18.2/3 (sheet 1 of 3): Process SRI_HLR
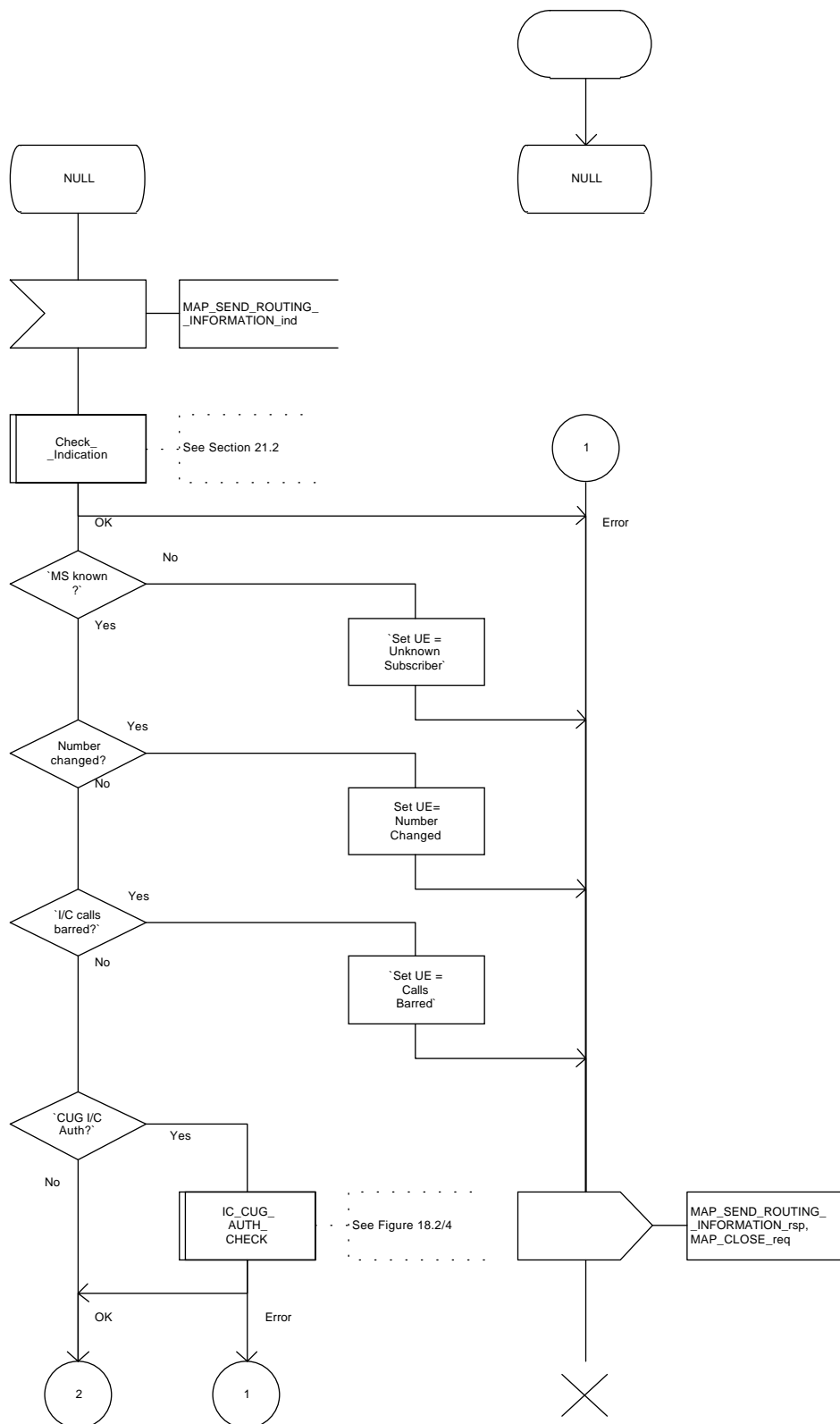
Figure 18.2/3 (sheet 2 of 3): Process SRI_HLR
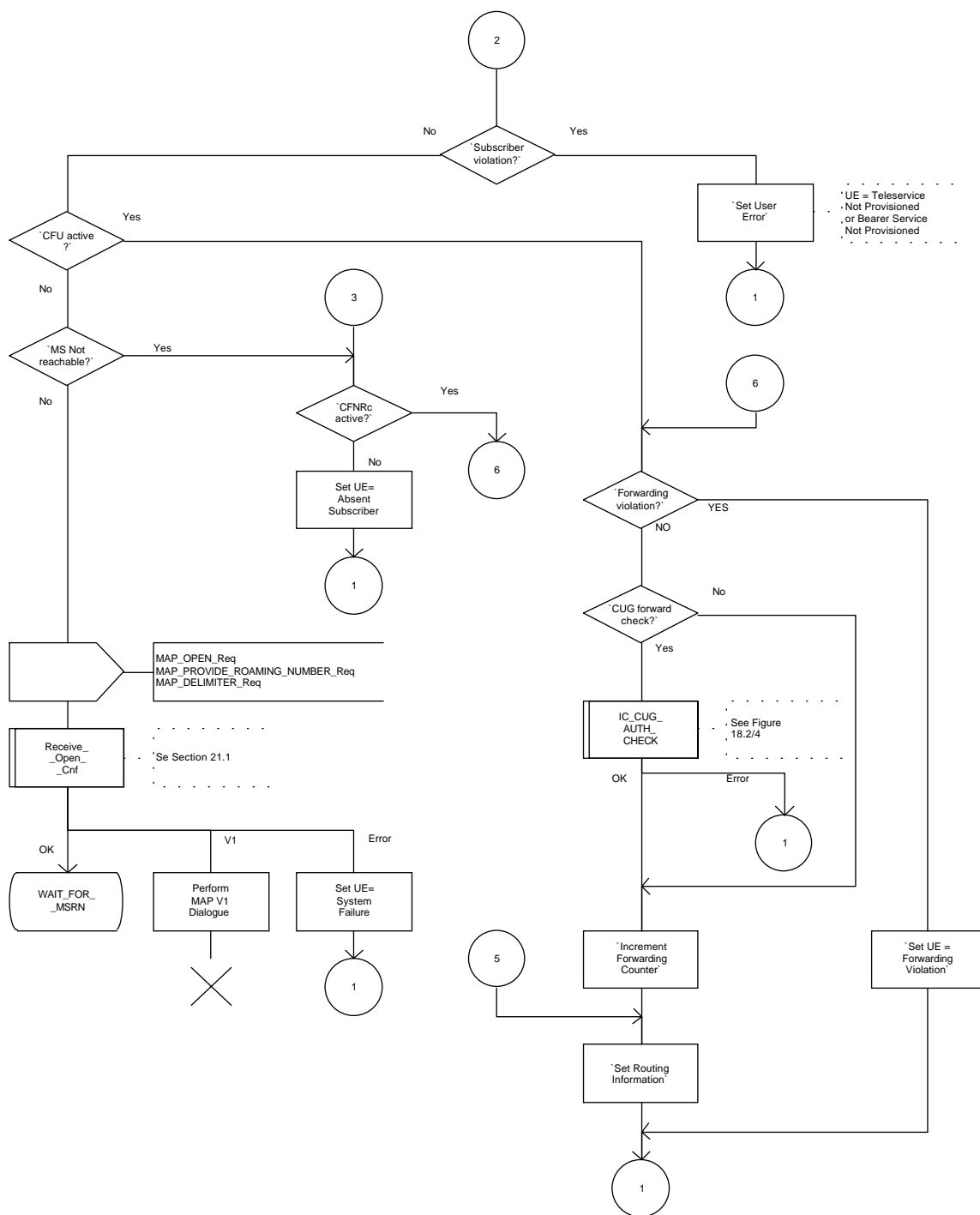
Figure 18.2/3 (sheet 3 of 3): Process SRI_HLR
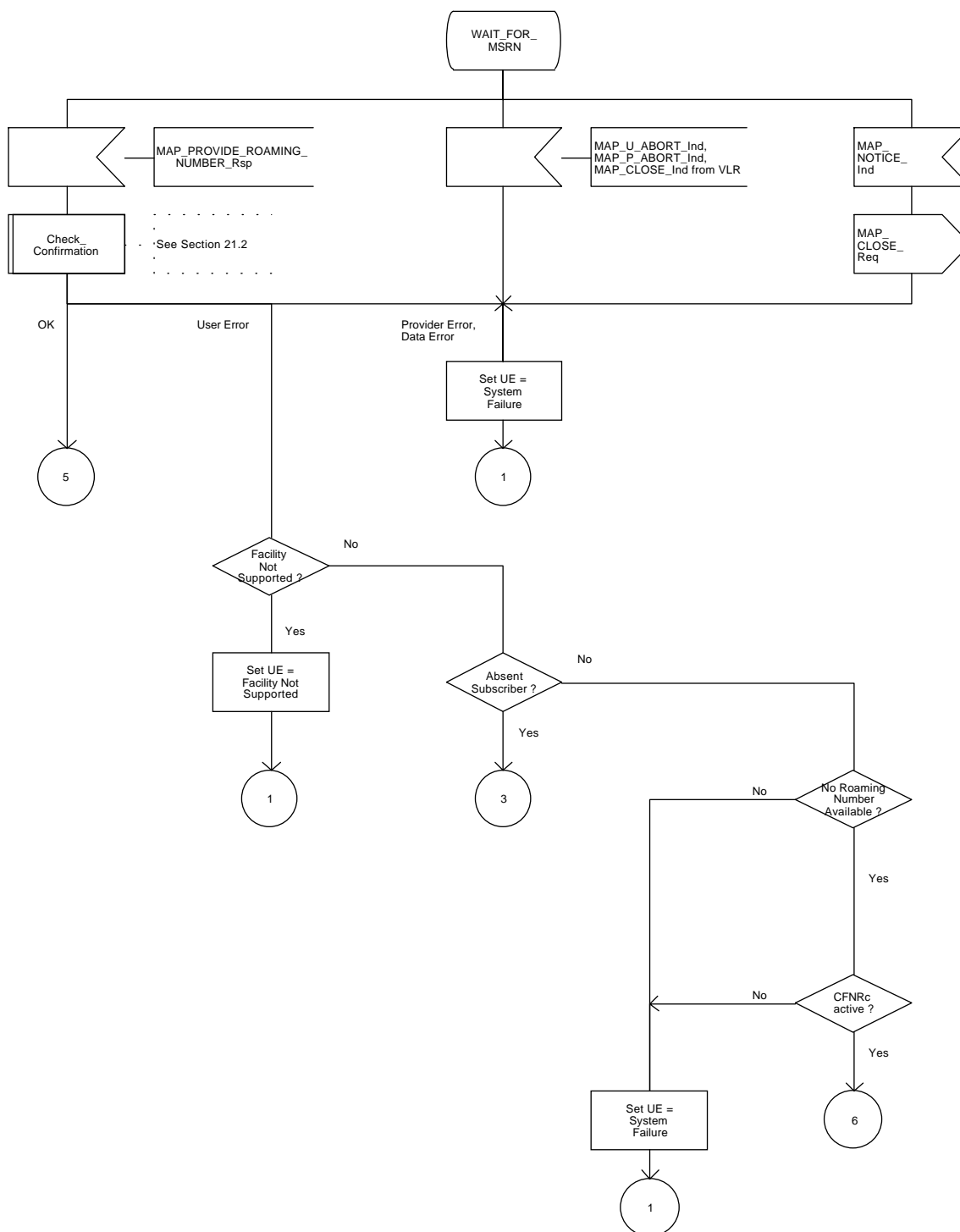
Figure 18.2/4: Macro IC_CUG_AUTH_CHECK
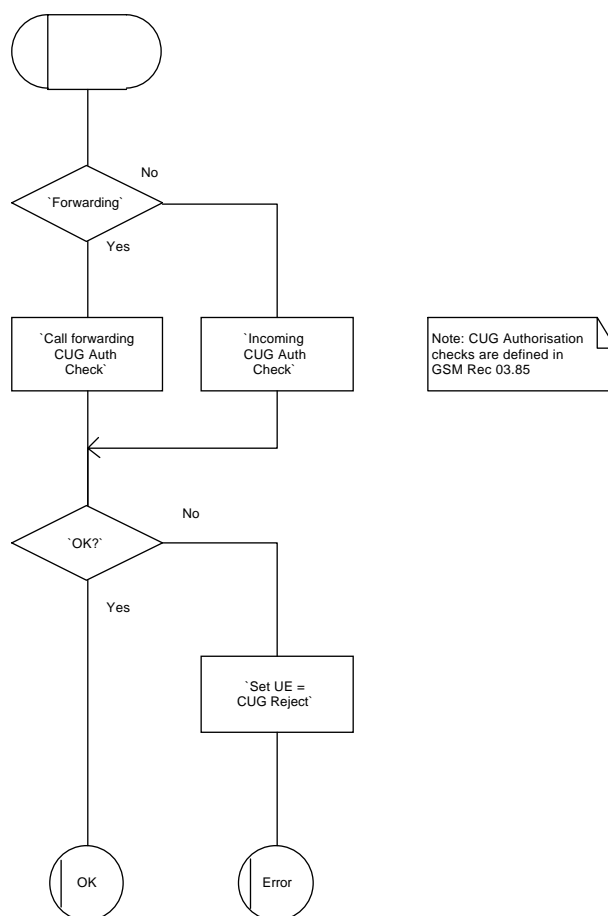
18.2.4 Procedures in the VLR

The procedures are described in figures 18.2/5, 18.2/6 and 18.2/7. These procedures make use of a number of processes and macros not described in this subclause. The description of these can be found in the following subclauses:

Receive_Open_Cnf	subclause 21.1;
Check_Confirmation	subclause 21.2;
Receive_Open_Ind	subclause 21.1;
Check_Indication	subclause 21.2;
Obtain_Authent_Sets_VLR	subclause 21.5;
Insert_Subs_Data_VLR	subclause 21.7.

Successful Outcome

On reception of the MAP_PROVIDE_ROAMING_NUMBER service indication from the HLR, the VLR will first perform a service provision check to confirm that the service requested is supported by the PLMN.

If the service is provided and the MS is also known, the VLR will check the IMSI detached flag for the MS. If the IMSI detached flag was not set then an MSRN will be allocated. An MSC number will already be stored in the VLR as well as the HLR.

The MSC number provided in the MAP_PROVIDE_ROAMING_NUMBER service indication will be used with the RAD_CONF_IND for the creation of the MSRN. That is, if the MSC number provided by the HLR does not match the MSC number stored against the IMSI concerned the indicator "Confirmed by Radio Contact" (RAD_CONF_IND) should be checked to establish if the number has been confirmed by radio contact.

If the RAD_CONF_IND is set to "NOT_CONF", this means that the MSC number is out of date as it has not been confirmed by radio contact, therefore the MSC number provided by the HLR can be used for creation of the MSRN.

If the RAD_CONF_IND is set to "CONF", this means that the MSC number has already been confirmed by radio contact and therefore, is more likely to be correct than the MSC number held by the HLR. The MSC number held by the VLR can be used for creation of the MSRN.

If no IMSI record exists for the IMSI received in the MAP_PROVIDE_ROAMING_NUMBER service indication or if no MSC number is available in the IMSI record concerned, the VLR will use the MSC number provided by the HLR for creation of the MSRN.

> NOTE: MAP version 1 described 2 options for allocating an MSRN by the VLR, either at location updating or per terminating call set up. MAP Version 2 only supports one of these options, that of allocating the MSRN per terminating call set-up. The option to allocate MSRN at location updating has been deleted.

The roaming number is returned to the HLR in the MAP_PROVIDE_ROAMING_NUMBER service response. The VLR associates the MSRN with a service indication, if such an indication was contained in the roaming number request from the HLR. This roaming number will be used to set up the call to the serving MSC. The procedure is then terminated in the VLR.

If the indicator "Confirmed by HLR" (HLR_CONF_IND) is set to "Not Confirmed", the VLR will also need to request authentication and subscriber data. This is described in the procedure for MS Not Known, below.

Parameter or data errors

If the MAP_PROVIDE_ROAMING_NUMBER service indication contains both the LMSI and the IMSI, the VLR checks whether the stored IMSI matches the received IMSI. If it does not, the VLR attempts to process the request using the IMSI received from the HLR to define the called subscriber.

If any parameter or data errors are detected in the MAP_PROVIDE_ROAMING_NUMBER service indication, the VLR will return "Unexpected Data Value" in the MAP_PROVIDE_ROAMING_NUMBER service response. The procedure is then terminated in the VLR.

Called Subscriber Not Reachable

If the called subscriber is not reachable, either because the IMSI Detached Flag is set or because the called subscriber is not allowed to roam into the current location area, the error "Absent Subscriber" is returned to the HLR. This procedure is then terminated in the VLR.

No Roaming Number Available

If the VLR cannot assign a Roaming Number, the user error "No Roaming Number Available" is returned to the HLR in the MAP_PROVIDE_ROAMING_NUMBER service response. The procedure is then terminated in the VLR.

Service Not Supported

If the Service Provision check shows that the service is not provided, the error "Facility Not Supported" is returned to the HLR in the MAP_PROVIDE_ROAMING_NUMBER service response. The procedure is then terminated in the VLR.

MS not Known

If the MS is not known by the VLR, this indicates that either a VLR restart has occurred after a VLR failure or the HLR is in restoration phase and the VLR addressed by the HLR is not the one where the called MS is currently located. The VLR will still proceed to allocate an MSRN (if available) and, as an operator option, an LMSI. In addition, a skeleton IMSI record for this MS will be created and the indicators "Confirmed by HLR" and "Confirmed by Radio Contact" set to "Not Confirmed". If the VLR serves two or more MSCs, the VLR will also set the restoration indicator "Location Information Confirmed in HLR" (LOC_CONF_IND) to "Not confirmed". These indicators will be set to "Confirmed" when a procedure with the relevant entity has been successfully completed. The roaming number is then returned to the HLR in the MAP_PROVIDE_ROAMING_NUMBER service response.

To be able to process the MAP_SEND_INFO_FOR_INCOMING_CALL indication later, the VLR will request separately, for the IMSI concerned, the following:

- authentication data with a MAP_SEND_AUTHENTICATION_INFO (if authentication is to be performed), this procedure is described in clause 21;

- subscriber data with a MAP_RESTORE_DATA request, this procedure is described in the subclause below.

These additional procedures are run independently of the provide roaming number procedure, no result is expected in this procedure.

VLR Restoration

The restore data procedure is invoked by the VLR to update subscriber data in the HLR and VLR after VLR restart, when the VLR receives a MAP_PROVIDE_ROAMING_NUMBER indication from the HLR for an unidentified MS, or for a known MS with the indicator "Confirmed by HLR" set to "Not Confirmed". The procedure in the VLR is described in figure 18.2/6.

If the dialogue with the HLR is opened successfully, the process waits for a MAP_ACTIVATE_TRACE_MODE indication primitive or one or more MAP_INSERT_SUBSCRIBER_DATA indication primitives containing the subscriber data. When the tracing activation has been completed if necessary and all the subscriber data have been received, the dialogue with the HLR is terminated by a MAP_RESTORE_DATA confirmation. If the confirmation contains no errors, the VLR stores the subscriber data in the IMSI record, updates the HLR number and the MS Not Reachable Flag which is used for SMS (if appropriate), and sets the indicator "Confirmed by HLR" to "Confirmed". The status of the indicators "Confirmed by radio contact" and "Location Information Confirmed in HLR" remains unchanged.

The procedure in the VLR is then terminated. For a description of the procedures in the HLR refer to subclause 16.3.

Figure 18.2/5 (sheet 1 of 3): Process PRN_VLR
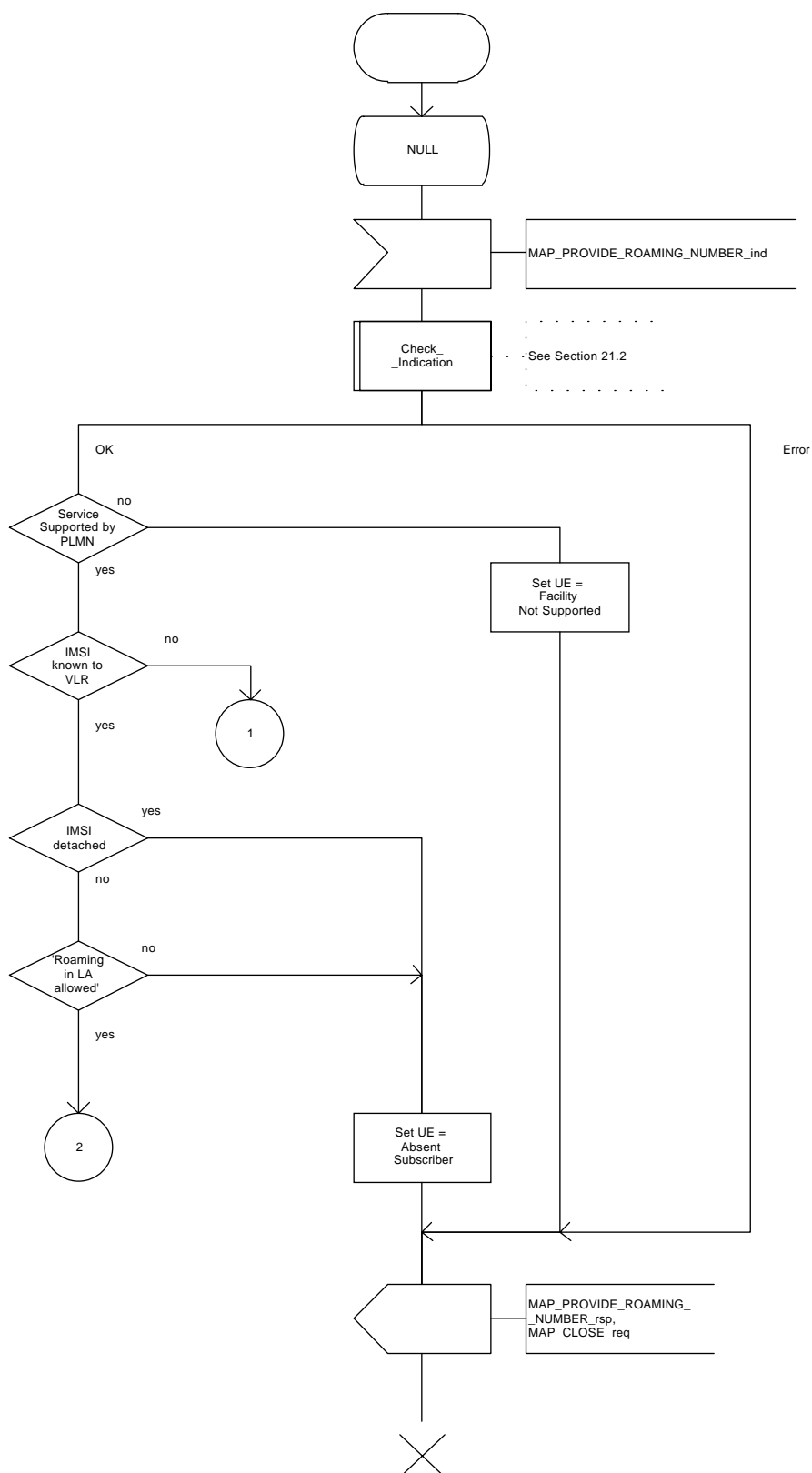
Figure 18.2/5 (sheet 2 of 3): Process PRN_VLR
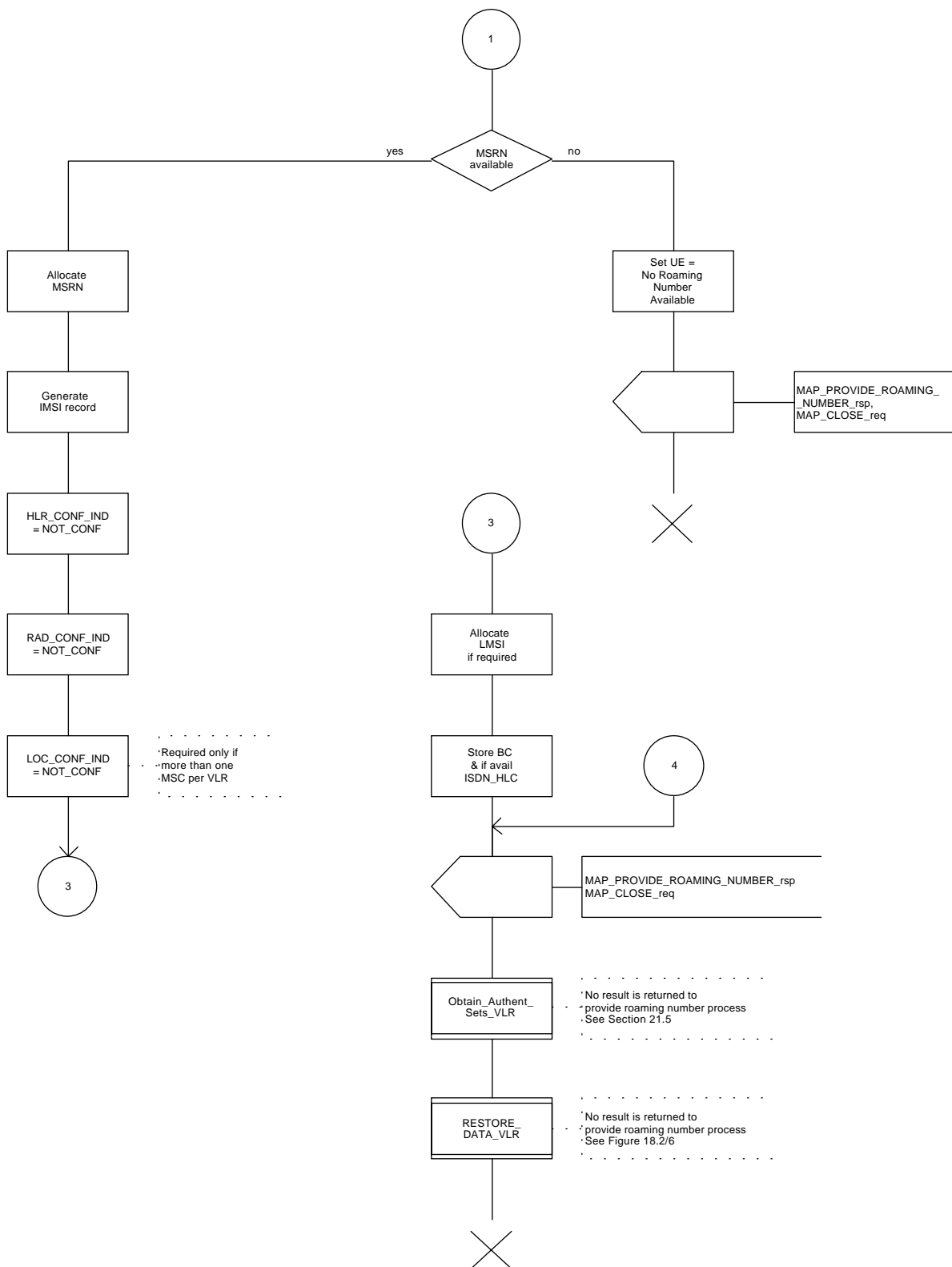
Figure 18.2/5 (sheet 3 of 3): Process PRN_VLR
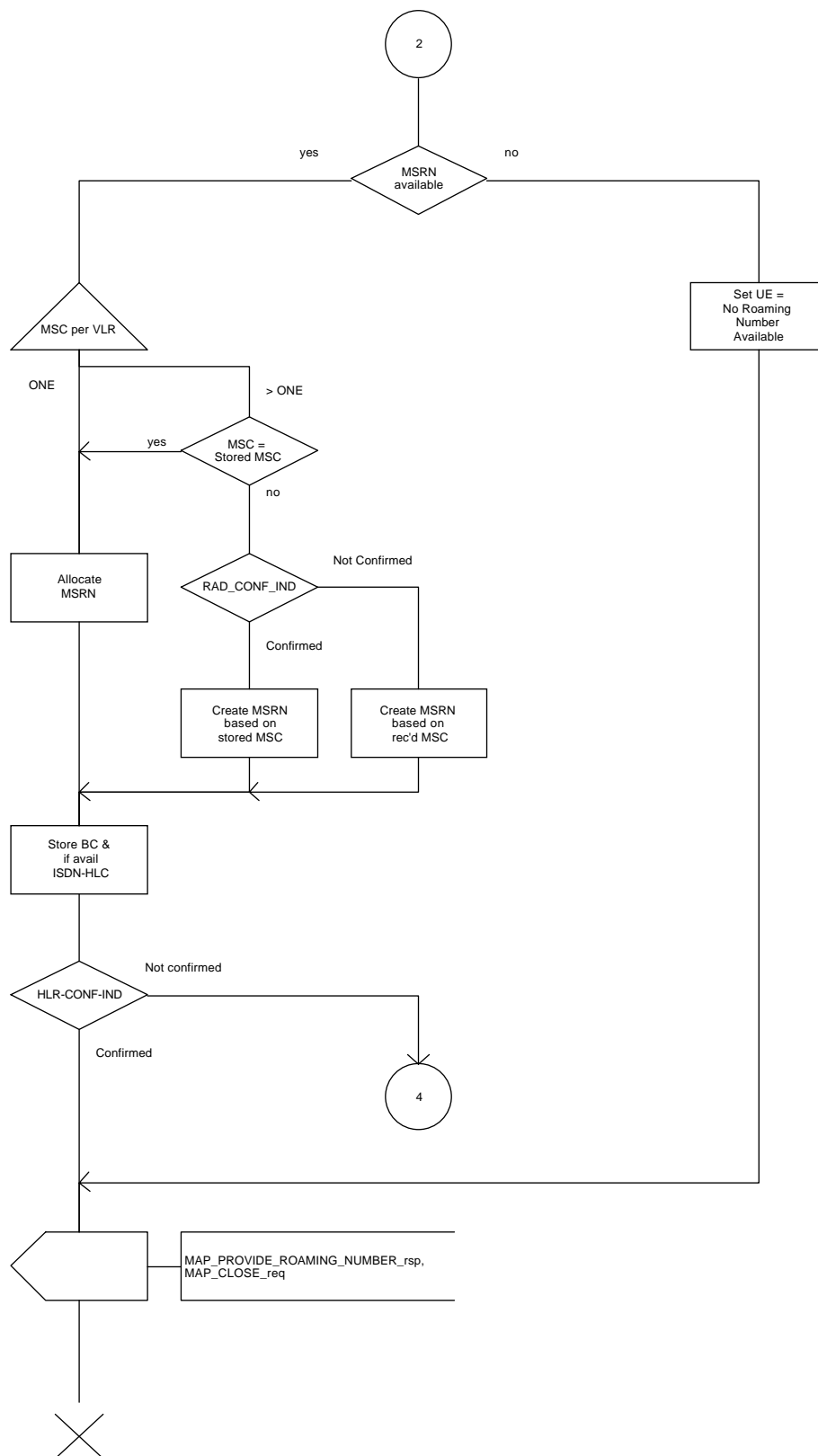
Figure 18.2/6: Process RESTORE_DATA_VLR
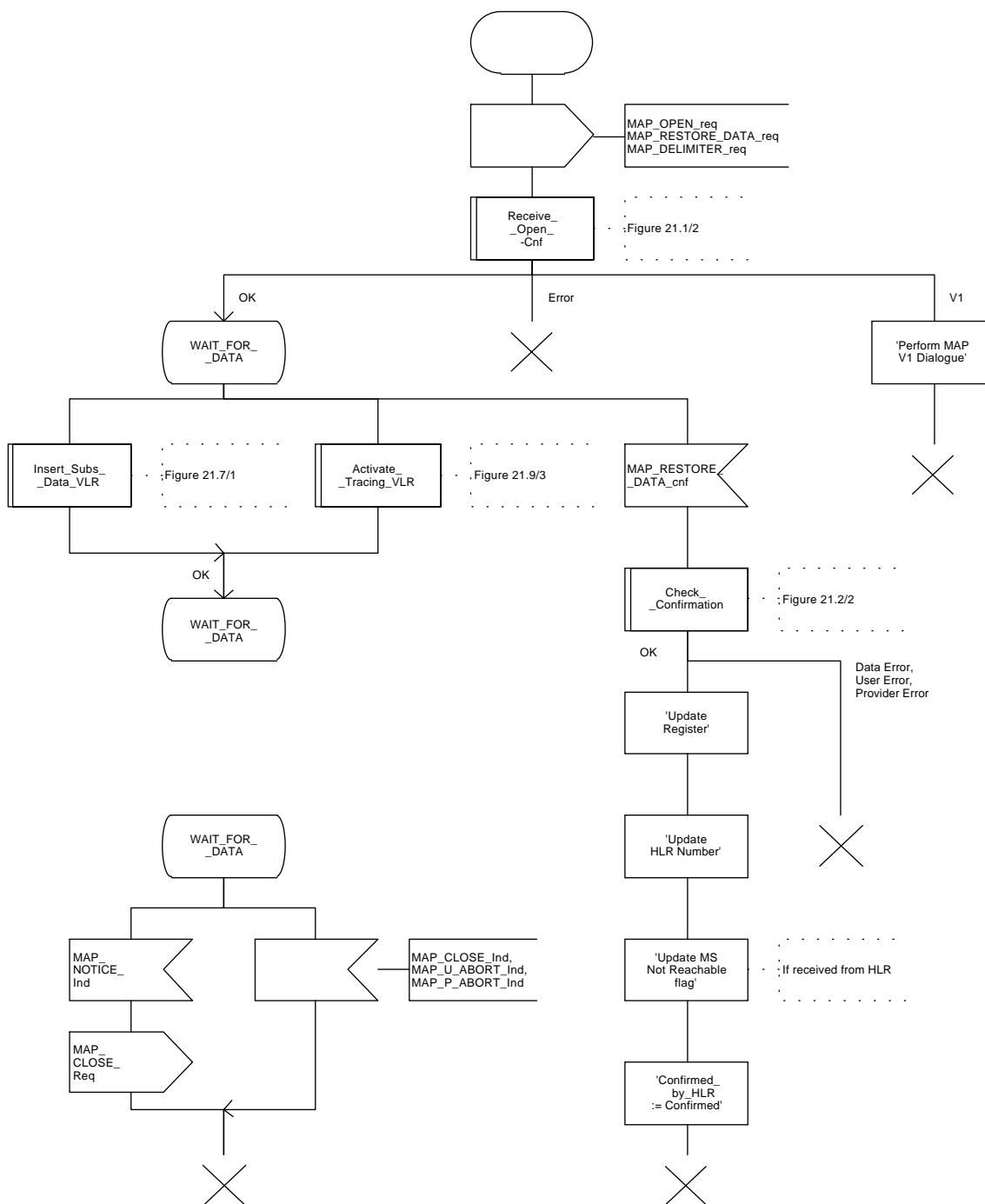
18.3 MS terminating calls

18.3.1 General

The procedure is shown in figure 18.3/1, this is an example sequence which assumes successful operation. It is also assumed that the location of the subscriber has already been determined.

```
                                        Gateway
 ┌───┐                                   ┌───┐                              ┌───┐
 │MSC│                                   │MSC│                              │VLR│
 └─┬─┘    I_IAM (note 1)                 └─┬─┘                              └─┬─┘
   │───────────────────────────────────────>│ MAP_SEND_INFO_FOR_INCOMING_    │
   │ ┌──┐          ┌───┐                     │              CALL              │
   │ │MS│ (note 3) │BSS│ (note 2)            │────────────────────────────────>│
   │ └┬─┘          └─┬─┘                     │ MAP_PAGE or MAP_SEARCH_FOR_    │
   │               │         Paging          │     MOBILE_SUBSCRIBER          │
   │               │<────────────────────────│<────────────────────────────────│
   │               │      Layer 3 info       │                                │
   │               │────────────────────────>│                                │
   │               │                          │   MAP_PROCESS_CALL_WAITING     │
   │               │                          │                                │
   │               │            MAP_PROCESS_ACCESS_REQUEST                     │
   │               │                          │────────────────────────────────>│
   │               │                          │    MAP_PROVIDE_IMSI            │
   │               │                          │    MAP_AUTHENTICATION          │
   │               │                          │    MAP_SET_CIPHERING_MODE      │
   │               │                          │    MAP_CHECK_IMEI              │
   │               │            MAP_PROCESS_ACCESS_REQUEST ack                 │
   │               │                          │<────────────────────────────────│
   │               │                          │   MAP_FORWARD_NEW_TMSI         │
   │               │                          │                                │
   │               │             MAP_COMPLETE_CALL                            │
   │               │                          │<────────────────────────────────│
   │          A_SETUP                         │                                │
   │<─────────────────────────────────────────│                                │
   │          A_CALL_CONF                      │                                │
   │─────────────────────────────────────────>│                                │
   │               │      A_ASS_REQ           │                                │
   │               │<────────────────────────│                                 │
   │               │      A_ASS_COM           │                                │
   │               │────────────────────────>│                                 │
   │          A_ALERT                         │                                │
   │─────────────────────────────────────────>│                                │
   │      I_ACM (note 1)                      │                                │
   │<─────────────────────────────────────────│                                │
   │          A_CONN                          │                                │
   │─────────────────────────────────────────>│                                │
   │          A_CONN_ACK                      │                                │
   │<─────────────────────────────────────────│                                │
   │      I_ANM (note 1)                      │ MAP_SEND_INFO_FOR_INCOMING_    │
   │<─────────────────────────────────────────│            CALL ack            │
   │                                          │<────────────────────────────────│
```

xxx = Optional Procedure

Figure 18.3/1: Procedure for MS terminating call set-up

Notes to figure 18.3/1:

NOTE 1: TUP or ISUP may be used in signalling between MSCs, depending on the network type between the MSCs. For further details on the TUP and ISUP procedures refer to the following CCITT Recommendations:

Q.721-Q.725 - Telephone User Part (TUP);
Q.761-Q.764 - ISDN User Part (ISUP).

It is assumed that signalling towards the fixed network only uses 3 messages IAM (Initial Address Message), ACM (Address Complete Message) and ANM (Answer Message). Examples of other interworking cases are not shown in this standard (for further information see e.g. CCITT Recommendation Q.699).

NOTE 2: The BSS and MSC message flow shown in this diagram is only an example of the interworking possibilities. For further details of the procedures between the MSC and BSS refer to GSM 08.08 (MSC to BSS Interface, Layer 3).

NOTE 3: The MS and MSC message flow shown in this diagram is only an example of the interworking possibilities. For further details on the procedures between the MSC and MS refer to GSM 04.08 (Mobile Radio Interface Layer 3 Specification).

The following MAP services are used during MS terminating call set up:

MAP_SEND_INFO_FOR_INCOMING_CALL see clause 8;
MAP_PROCESS_ACCESS_REQUEST   see clause 6;
MAP_PAGE   see clause 6;
MAP_SEARCH_FOR_MS   see clause 6;
MAP_COMPLETE_CALL   see clause 8;
MAP_PROCESS_CALL_WAITING   see clause 8;
MAP_PROVIDE_IMSI   see clause 6;
MAP_AUTHENTICATION   see clause 6;
MAP_SET_CIPHERING_MODE   see clause 6;
MAP_FORWARD_NEW_TMSI   see clause 6;
MAP_CHECK_IMEI   see clause 6.

18.3.2   Procedures in the MSC

The procedures are shown in figures 18.3/2, 18.3/3 and 18.3/4. These procedures make use of a number of processes and macros not described in this subclause. The description of these can be found in the following subclauses:

Receive_Open_Cnf   subclause 21.1;
Check_Confirmation   subclause 21.2;
Check_Indication   subclause 21.2;
Process_Access_Request_MSC   subclause 21.4;
Page_MSC   subclause 21.3;
Search_For_MS_MSC   subclause 21.3.

Successful Outcome

The Gateway MSC will pass the IAM (Initial Address Message) to the appropriate MSC. The Gateway MSC knows the location of the called subscriber from the roaming number.

On receipt of an IAM the MSC will initiate the MAP_SEND_INFO_FOR_INCOMING_CALL service request to request from the VLR, the relevant parameters for the called subscriber to enable the call to be set up. The response to this from the VLR will be a request to the MSC to commence Paging. This request will either be in the form of a MAP_PAGE service indication or a MAP_SEARCH_FOR_MS service indication. If the MSC does not detect that the called MS is in a busy state (see GSM 02.01 for a definition of these states), the paging procedure will be activated.

Successful response to paging from the BSS will be complete layer 3 information as specified in GSM 08.08, this indicates that the addressed MS has responded to the paging. The MSC will then initiate a MAP_PROCESS_ACCESS_REQUEST service request to the VLR to initiate processing of a subscriber access to the network.

On successful completion of the Process Access Request procedure the VLR will send the required parameters for the call set up to proceed. This information will be contained in a MAP_COMPLETE_CALL service indication.

To start call establishment the MSC will send an A_SETUP message to the MS. Following compatibility checking the MS will respond with an A_CALL_CONF message. These messages and the procedures associated with them are described in more detail in GSM 04.08. To complete call establishment the following events occur:

- channel assignment (refer GSM 08.08);

- called user alerted (refer GSM 04.08), this will be followed by the MSC sending an Address Complete Message (I_ACM) to the Gateway MSC;

- called user accepts the call, this will be followed by the MSC sending an Answer Message (I_ANM) to the Gateway MSC;

- the VLR will close the dialogue with the MSC by returning a MAP_SEND_INFO_FOR_INCOMING_CALL service confirm with the result parameter indicating success.

Parameter or data errors

If any parameter or data errors are detected by the MSC, the procedure for this call will be terminated and a release message (I_REL) returned to the gateway MSC to release the call.

Service Provider Failure

Once the dialogue between the MSC and the VLR is established a MAP_P_ABORT service indication may be received at any stage from the MAP Service Provider. This aborts the dialogue, see clause 5 for a definition of the abort reasons.

The call will be released by the MSC sending a Release message (I_REL) to the Gateway MSC. If at the time the procedure is terminated the call has already been confirmed by the MS, i.e. the MS has responded to the set up message (A_SETUP) by returning a call confirmed message (A_CALL_CONF), then the resources associated with the mobile terminating side of the call should also be released (i.e. MS and base station resources). The procedure in the MSC is then terminated.

Call Released by Calling Subscriber

This may occur at any stage during the call set up procedure. Call release is indicated to the MSC by a Release Message (I_REL). The MSC will then terminate the procedure towards the VLR using the MAP_U_ABORT service. The User Reason will indicate Application Procedure Cancellation and the Diagnostic Information will indicate Call Release. If at the time the procedure is terminated the call has already been confirmed by the MS, i.e. the MS has responded to the set up message (A_SETUP) by returning a call confirmed message (A_CALL_CONF), then the resources associated with the mobile terminating side of the call should also be released (i.e. MS and base station resources). The procedure in the MSC is then terminated.

No Paging or Search Response

If the timer for paging expires, paging may be repeated n times. The number of successive paging attempts and the time intervals are a network dependent choice. If the paging fails the MSC informs the VLR in the MAP service response (either MAP_PAGE or MAP_SEARCH_FOR_MS) with the user error set to "Absent Subscriber". If the Call Forwarding on MS Not Reachable service is not active, the VLR will terminate the procedure by returning a MAP_SEND_INFO_FOR_INCOMING_CALL service confirm with the user error set to "Impossible Call Completion". The MSC will then send a Release message (I_REL) to the Gateway MSC to release the call.

If Call Forwarding on MS Not Reachable is active, then the call will be forwarded as described below in the Call Forwarding subclause.

Called Subscriber Not Reachable

In response to the MAP_SEND_INFO_FOR_INCOMING_CALL service request the VLR may inform the MSC that the subscriber cannot be reached. If Call Forwarding on MS Not Reachable service is not active, this notification is contained in a MAP_SEND_INFO_FOR_INCOMING_CALL service confirm with the user error set to "Absent Subscriber". To terminate the procedure, the MSC will send a Release message (I_REL) to the Gateway MSC to release the call.

If Call Forwarding on MS Not Reachable service is active, then the call will be forwarded as described below in the Call Forwarding subclause.

Access to Called Subscriber Not Possible

The VLR, in response to the request from the MSC to provide information for the called subscriber (MAP_SEND_INFO_FOR_INCOMING_CALL service request) may detect an error condition. This is reported in the MAP_SEND_INFO_FOR_INCOMING_CALL service confirm with the user error indicating one of the following:

　　Unallocated Roaming Number;
　　Unknown Subscriber;
　　Unexpected Data Value;
　　Data Missing;
　　System Failure.

To terminate the procedure for this call the MSC will send a Release Message (I_REL) to the Gateway MSC to release the call.

Called Subscriber is Busy

Two scenarios need to be considered here:

1)   If, following the paging request from the VLR (either MAP_PAGE or MAP_SEARCH_FOR_MS service indication), the MSC has knowledge of the "busy" state of the subscriber (see GSM 02.01 for a definition of these states), the MSC will set the user error to "Busy Subscriber". Depending on which state is detected the Busy Subscriber error will indicate either More Calls Allowed or Network Determined User Busy (no more calls allowed). This error will be returned to the VLR in the page response (either MAP_PAGE or MAP_SEARCH_FOR_MS service response).

        Depending on the Busy Subscriber state, the VLR may request Call Waiting or Call Forwarding on Busy to be invoked if these services are active. The procedures are described in the subclauses below.

        If neither Call Forwarding on Busy nor Call Waiting service is active, the VLR will terminate the procedure by returning a MAP_SEND_INFO_FOR_INCOMING_CALL service confirm with the user error set to "Impossible Call Completion". The MSC will then send a Release message (I_REL) to the Gateway MSC to release the call.

2)   If the Called MS enters a busy state after the paging but before the Set up request is sent to the MS (A_SETUP), the MS may respond to a second set up request with either Call Confirmed message (A_CALL_CONF) User Busy (Error Cause 17) or a Release Complete message (A_REL_COM) also indicating User Busy. In either case the MSC will generate a response to the VLR, a MAP_COMPLETE_CALL service confirm, with the user error indicating "Busy Subscriber" (User Determined User Busy).

        If the Call Forwarding on Busy service is active, then the call will proceed as described in the Call Forwarding subclause below.

        If Call Forwarding on Busy is not active, the VLR will terminate the procedure by returning a MAP_SEND_INFO_FOR_INCOMING_CALL service confirm with the user error set to "Impossible Call Completion". The MSC will then send a Release message (I_REL) message to the Gateway MSC to release the call.

Call Rejected by Called Subscriber

Following the generation of a Set up message (A_SETUP) the called MS responds with a Release Complete Message (A_REL_COM) with an Error Cause indicating the nature of the error. For Error Cause #17 (User Busy) the handling of this cause is described in the Call Subscriber is Busy subclause above. For all other Error Causes the MSC will terminate the dialogue in the direction of the VLR sending a MAP_U_ABORT and then release the call sending a Release Message (I_REL) to the Gateway MSC. The procedure in the MSC is then terminated.

Assignment Failure

The BSS may fail to seize a radio channel for one of the following reasons:

- use of the terrestrial channel selected by the MSC is impossible;

    NOTE: It is possible another try for radio channel assignment can be attempted after reselection of a new terrestrial channel.

- Radio Congestion.

The MSC will be informed of this failure by the Assignment Failure message (A_ASS_FAIL). No further action can be taken at this stage, other than return the error Radio Congestion to the VLR in the MAP_COMPLETE_CALL service response. If the VLR confirms that the Call Forwarding on MS Not Reachable service is active, then the call will be forwarded as described below in the Call Forwarding subclause. Otherwise the VLR will instruct the MSC to terminate the procedure and release the call. The MSC releases the call by sending a Disconnect message (A_DISC) to the MS and a Release message (I_REL) to the Gateway MSC.

No Answer from Called Subscriber

Following the generation of Address Complete Message (I_ACM) there is no response from the Called MS, i.e. the Connect message (A_CONN) is not received within the given time period.

The MSC will generate a response to the VLR, a MAP_COMPLETE_CALL service confirm, with the user error indicating "No Subscriber Reply". If the Call Forwarding on No Reply service is active then the call will proceed as described in the Call Forwarding subclause below. If the service is not active, the VLR will terminate the procedure by returning a MAP_SEND_INFO_FOR_INCOMING_CALL service confirm with the user error set to "Impossible Call completion". The MSC will then send a Release message (I_REL) to the Gateway MSC to release the call.

Call Forwarding

If the VLR is informed that it is not possible to complete the call set up (e.g. absent subscriber, busy subscriber, No Subscriber Reply, Radio Congestion), the VLR will check if any of the appropriate Call Forwarding services are active. If the appropriate Call Forwarding service is active, and the call has not been forwarded more times than allowed, the VLR will provide the relevant forwarding data in the MAP_SEND_INFO_FOR_INCOMING_CALL service confirm. To terminate the procedure, the MSC will set the Redirection Information in the Initial Address message (I_IAM) from the VLR supplied data. If the call is a CUG call, then the CUG Information supplied by the VLR should replace the CUG Information provided in the original Initial Address Message (I_IAM). The reason for this is because it is possible that, as a result of the CUG Call Authorisation Check performed by the VLR, the CUG Information to be provided in the forwarding leg of the call may be different (see GSM 03.85 for details). This Initial Address Message will then be sent to the Gateway MSC for the call to be forwarded.

If the call has already been forwarded as many times as allowed the VLR will return MAP_SEND_INFO_FOR_INCOMING_CALL service confirm with the user error set to "Forwarding violation". In this instance the MSC will terminate the procedure by sending a Release message (I_REL) to the calling subscriber to release the call.

Call Waiting

If the VLR is informed that it is not possible to complete the call set up because the subscriber is busy, the VLR will check if the Call Waiting service is active. If the service is active, the VLR will instruct the MSC to commence Call Waiting by sending a set up message to the MS (A_SETUP), this indicates to the MS that another call is waiting. One of the following events will then occur:

-   T1 Timer Expiry
    Timer T1 (this is the network timer controlling time between sending of A_SETUP message and receipt of an acknowledge from the MS) times out. The MSC will release the call in the direction of the Calling Subscriber by sending a release message (I_REL) with the cause value set to User Busy (#17), to the Gateway MSC. The Call Waiting procedure in the MSC is then terminated and, as the call has already been released, no error is reported to the VLR. To terminate the Call Handling procedures for this call, the VLR will return to the MSC the MAP_SEND_INFO_FOR_INCOMING_CALL service confirm indicating success.

-   Call Confirmed
    The MS responds to the A_SETUP message with a Call Confirmed message (A_CALL_CONF), the cause will indicate User Busy (#17). Once this is received channel assignment (if required) is performed. In the case of assignment failure, the MSC will release the call in the direction of the Calling Subscriber by sending a release message (I_REL) with the cause value set to User Busy (#17), to the Gateway MSC. The Call Waiting procedure in the MSC is then terminated and, as the call has already been released, no error is reported to the VLR. To terminate the Call Handling procedures for this call, the VLR will return to the MSC the MAP_SEND_INFO_FOR_INCOMING_CALL service confirm indicating success.

    The call confirm message from the MS will be followed by an Alerting message (A_ALERT) from the MS, this indicates that called user Alerting has commenced. Failure to receive and Alerting message will result in termination of the Call Waiting procedure as described for Assignment Failure in the previous paragraph.

-   Disconnect
    The called MS may decide to reject the call, this is indicated to the MSC by the Disconnect Message (A_DISC) with the Error Cause set to #17 (User Busy). In this instance the MSC will terminate the Call Waiting procedure by returning the error Busy Subscriber (User Determined User Busy) to the VLR in the MAP_PROCESS_CALL_WAITING service response. Also, any other error cause returned in the Disconnect message will be handled in the same way.

If the Call Waiting is to proceed, i.e. the call has been confirmed and the Alerting message has been received from the called MS, then one of the following events will occur:

-   Called MS accepts the new call
    The called MS may elect to either place the existing call on hold or release the existing call, alternatively the A subscriber may release the call. Regardless of the scenario invoked to free the existing call, acceptance of the call waiting is signalled by the Called MS sending a Connect message (A_CONN) to the MSC. To acknowledge the connection the MSC will respond with a connect acknowledge message (A_CONN_ACK). The MSC also informs the calling user that the call has been answered by returning an Answer message (I_ANM). The Call Waiting procedure has now been completed successfully. As no error is reported, the VLR will terminate the Call Set up procedure for this call by returning MAP_SEND_INFO_FOR_INCOMING_CALL service confirm indicating success.

- CWT Timer expiry
  Timer CWT (this is the network timer controlling the time between receipt of the A_ALERT message and receipt of the A_CONN during waiting call) times out. If a No Reply Condition Timer (NRCT) value is not supplied by the VLR in the Call Waiting request then the procedure can be terminated by the MSC sending a Disconnect message (A_DISC) to the MS. The MSC will release the call in the direction of the calling subscriber by sending a Release message (I_REL), with the cause value set to User Busy (#17), to the Gateway MSC. The Call Waiting procedure is then terminated and, as the call has already been released, no error is reported to the VLR. To terminate the Call Handling procedures for this call, the VLR returns to the MSC the MAP_SEND_INFO_FOR_INCOMING_CALL service confirm indicating success.

  If a No Reply Condition Timer value is supplied by the VLR, this indicates that the Call Forwarding on No Reply service is active. In this case the MSC will terminate the Call Waiting procedure by returning MAP_PROCESS_CALL_WAITING service response with the error value set to No Subscriber Reply and await instructions from the VLR for forwarding the call.

- Disconnect
  The Called MS may release the new call, this will be indicated by a Disconnect message (A_DISC). The Call Waiting procedure is terminated and the error Busy Subscriber (User Determined User Busy) is returned to the VLR in the MAP_PROCESS_CALL_WAITING service confirm.

Alerting Failure

A Call Confirmed message (A_CALL_CONF) has been received from the MS but an Alerting message (A_ALERT) has not been received from the MS. The timer associated with the alerting procedure has expired. The MSC will then release the call by sending a Release message (I_REL) to the calling subscriber. The resources associated with the mobile terminating side of the call should also be released (i.e. MS and base station resources). To terminate the procedure in the VLR a MAP_U_ABORT indication is sent. The call handling procedures are then terminated in the MSC for this call.

Access Request Failure

The Access Request procedure may fail for several reasons, either the access request is not accepted (i.e. the subscriber is not recognised or roaming is not allowed) or one of the associated procedures fails (i.e. Authentication, Ciphering, IMEI Check or TMSI reallocation). Failure of this procedure is reported to the MSC in the MAP_PROCESS_ACCESS_REQUEST service confirm (see subclause 6.3 for a definition of the possible user errors). The VLR will terminate the call set up procedure by returning MAP_SEND_INFO_FOR_INCOMING_CALL service confirm with the user error set either to "System Failure" or "Absent Subscriber". The MSC will then terminate the procedure by sending a Release Message (I_REL) to the calling subscriber to release the call.

CUG Failure

If the VLR cannot convert the CUG Interlock Code into an Index the user error "CUG Reject" is returned to the MSC in the MAP_SEND_INFO_FOR_INCOMING_CALL service confirm. In this instance the MSC will terminate the procedure by sending a Release Message (I_REL) to the Gateway MSC, with the ISUP Cause Indicator set to the appropriate value. The mapping between the GSM and ISUP reject causes are defined in GSM 09 series of technical specifications.

Figure 18.3/2 (sheet 1 of 2): Process ICS_MSC
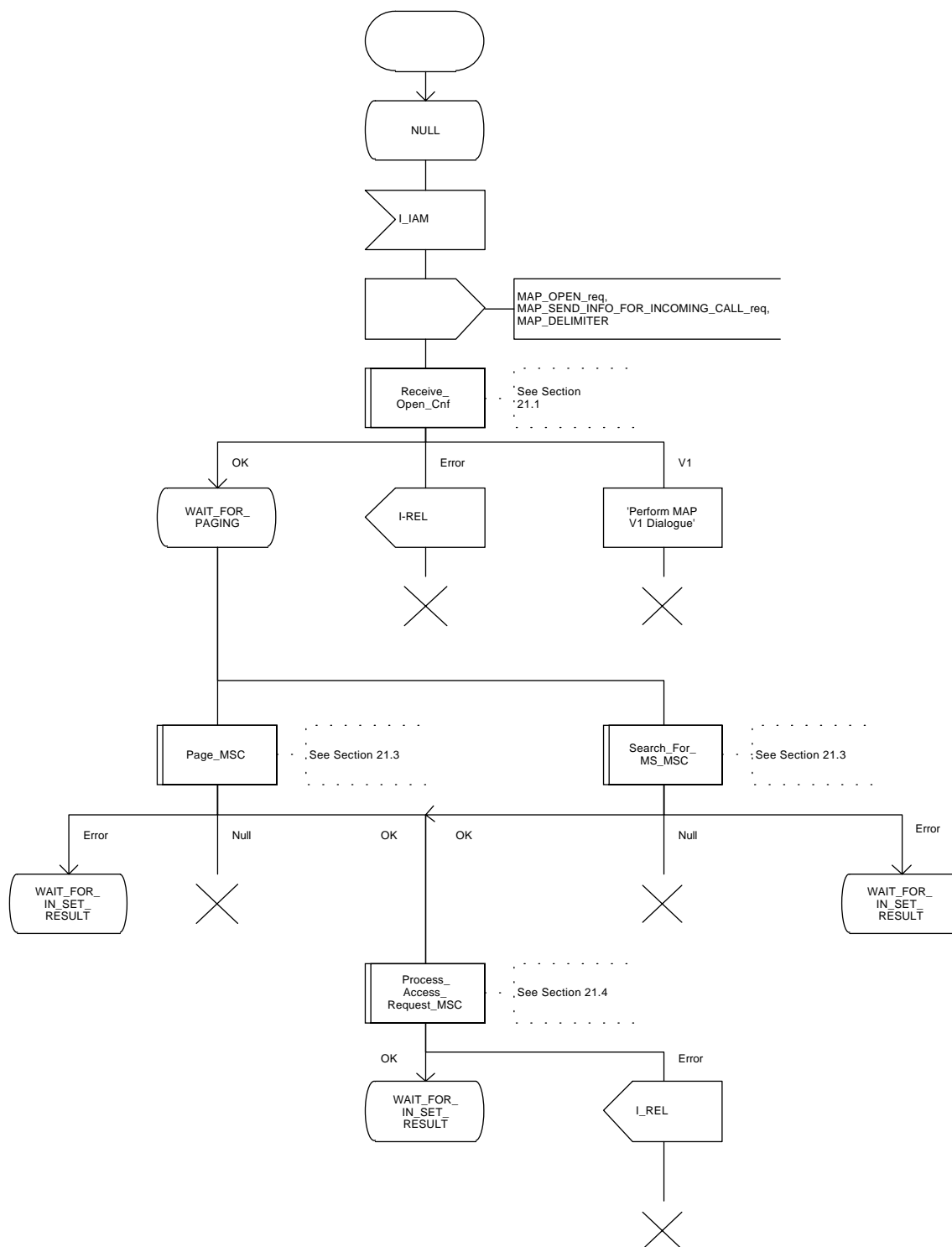
Figure 18.3/2 (sheet 2 of 2): Process ICS_MSC
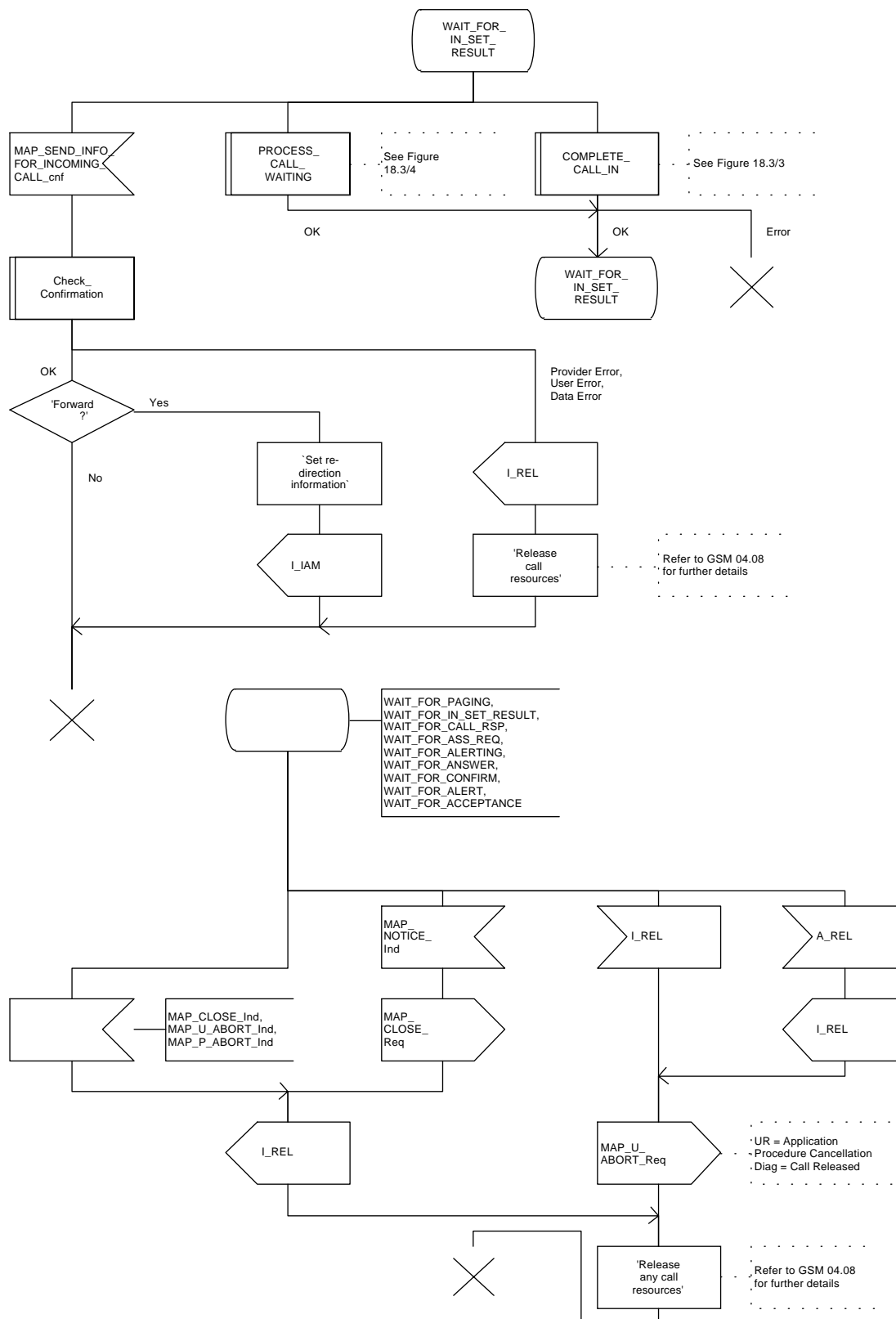
Figure 18.3/3 (sheet 1 of 2): Macro COMPLETE_CALL_IN
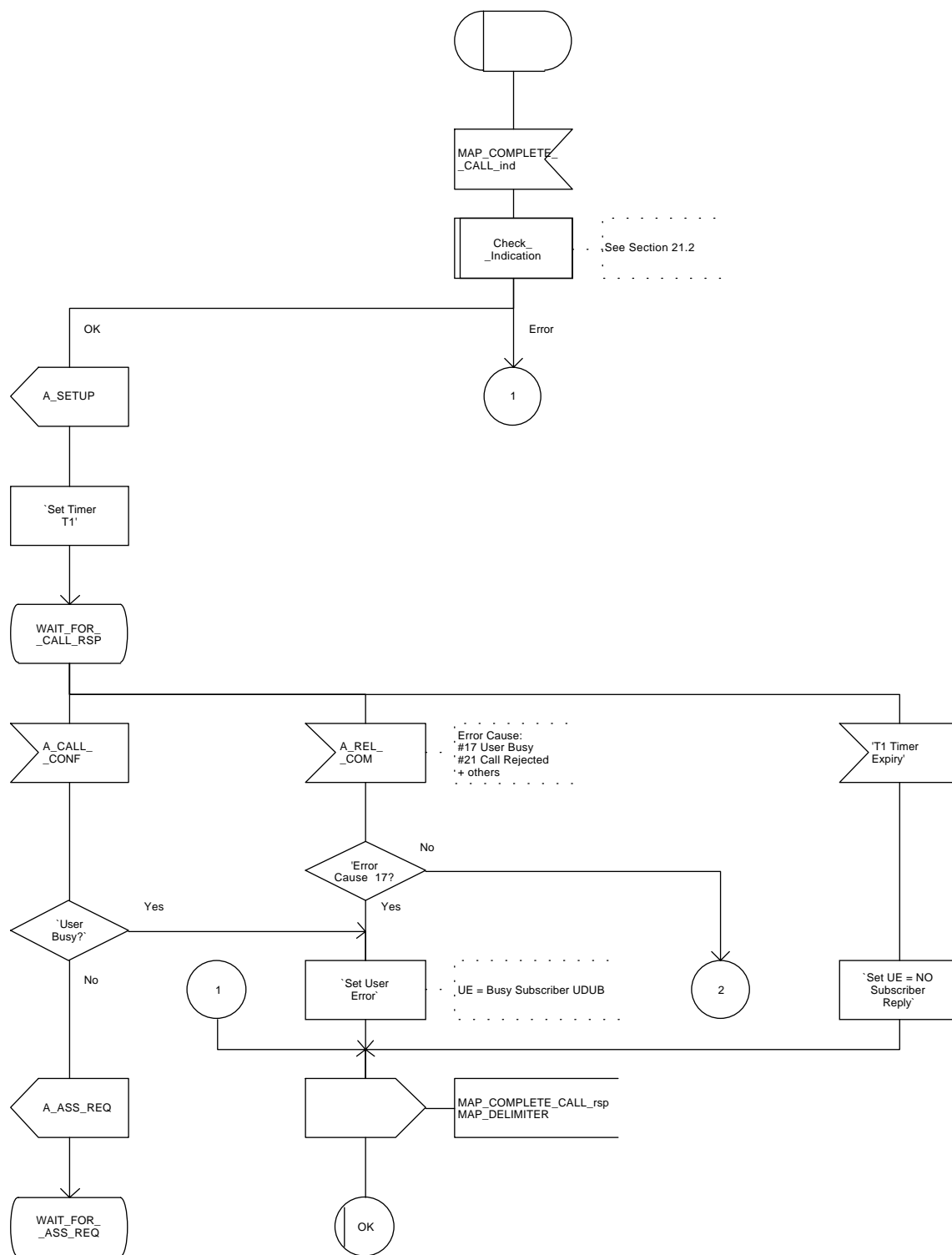
Figure 18.3/3 (sheet 2 of 2): Macro COMPLETE_CALL_IN
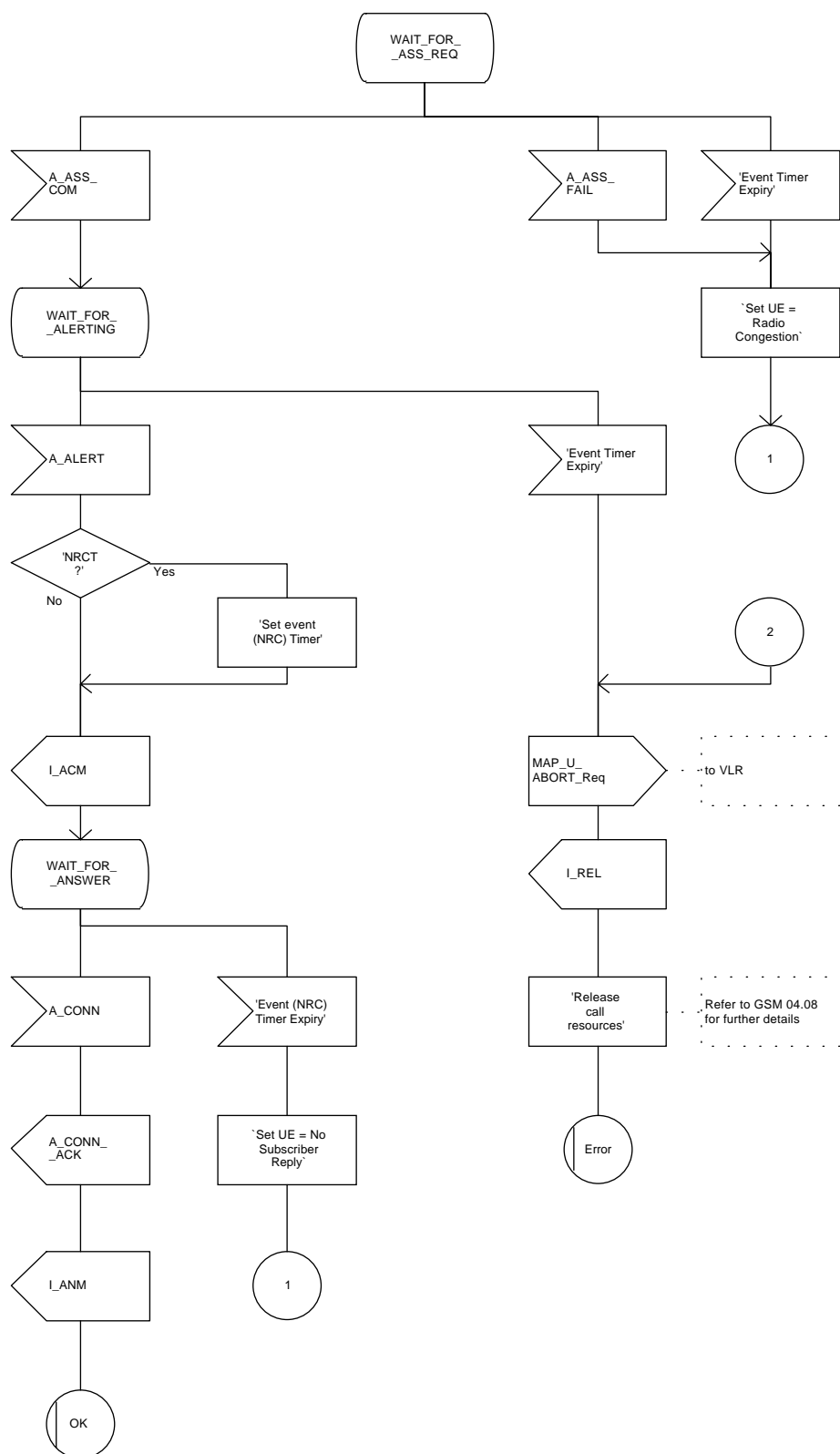
Figure 18.3/4 (sheet 1 of 3): Macro PROCESS_CALL_WAITING
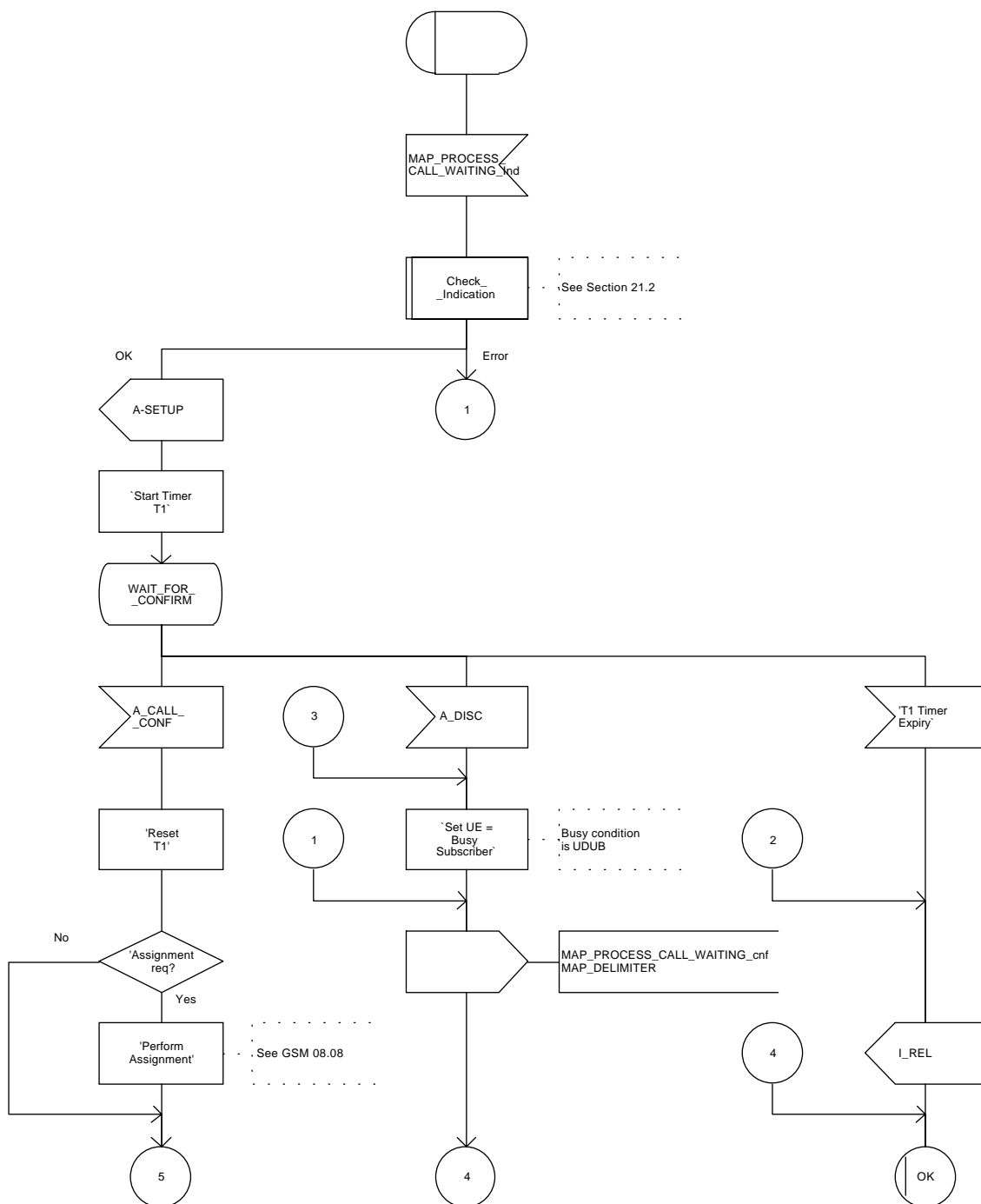
Figure 18.3/4 (sheet 2 of 3): Macro PROCESS_CALL_WAITING
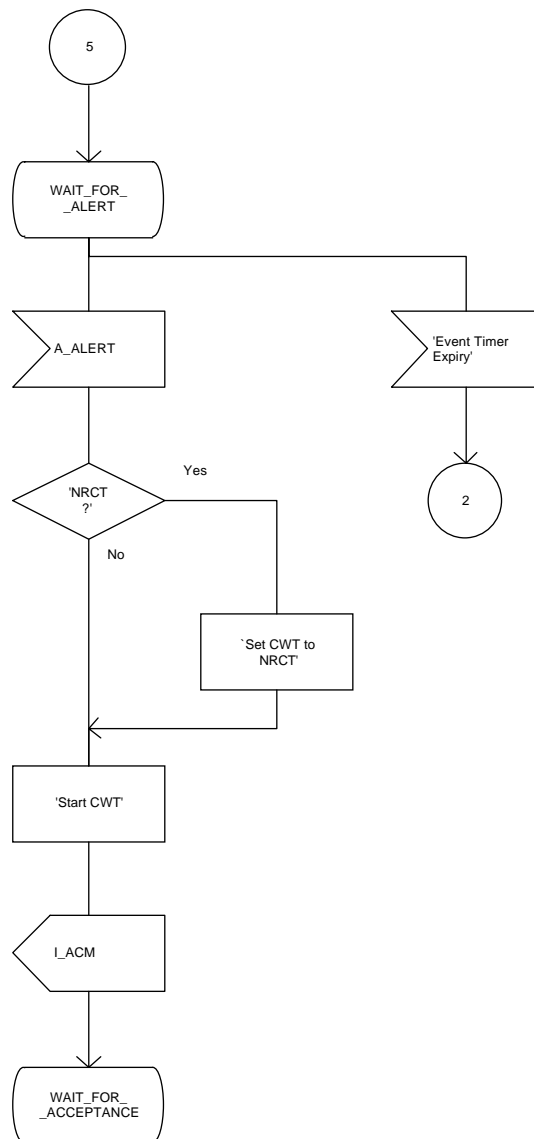
Figure 18.3/4 (sheet 3 of 3): Macro PROCESS_CALL_WAITING
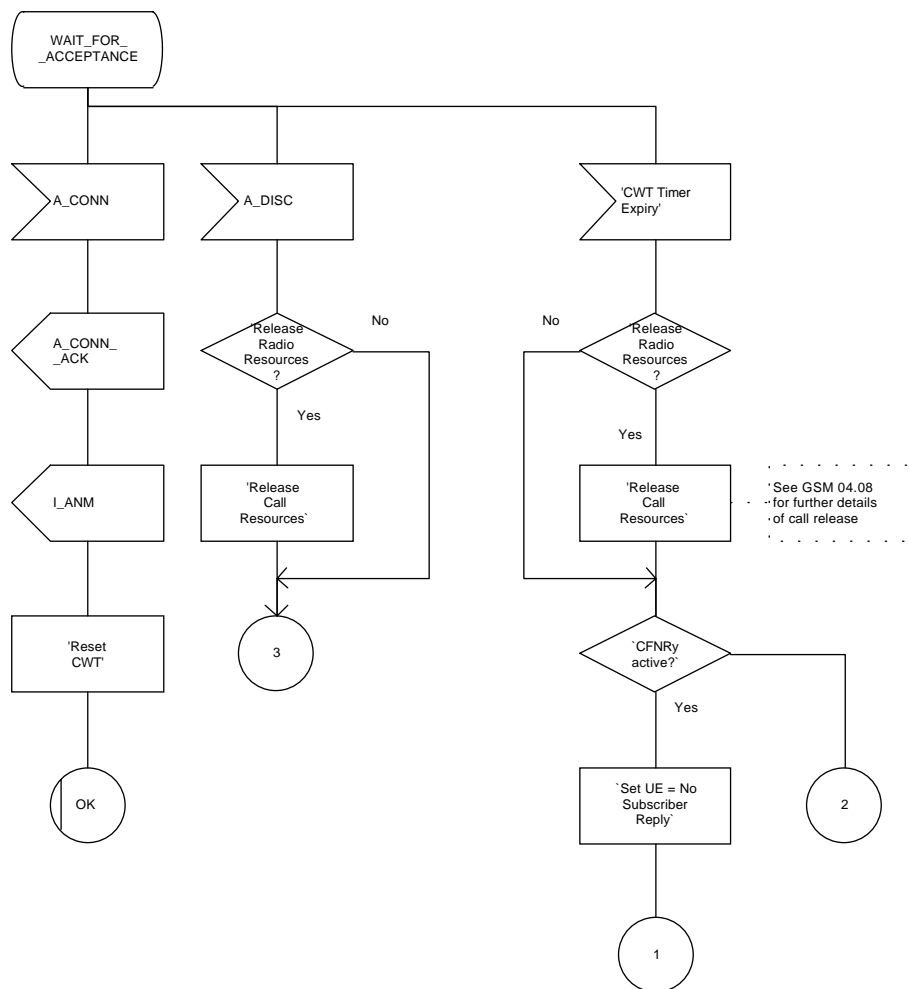
18.3.3   Procedures in the VLR

The procedure is shown in figures 18.3/5, 18.3/6 and 18.2/4. These procedures make use of a number of processes and macros not described in this subclause. The description of these can be found in the following subclauses:

Receive_Open_Cnf	subclause 21.1;
Check_Confirmation	subclause 21.2;
Check_Indication	subclause 21.2;
Process_Access_Request_VLR	subclause 21.4.

Successful Outcome

When the MAP_SEND_INFO_FOR_INCOMING_CALL service indication is received from the MSC, the VLR will first check the parameters and data for errors. If there are no errors the VLR will check to see if the MS is known to it, the MS will be identified by MSRN. If the MS is known, the Indicator "Confirmed by HLR" (HLR_CONF_IND) is set to "Confirmed", roaming in the current location area is allowed and the IMSI detached flag is not set, the VLR will initiate paging procedures to locate the MS.

If the VLR holds location information for the MS, the service MAP_PAGE is invoked, containing the IMSI of the subscriber, the Stored Location Area Id and the subscribers TMSI (if available in the VLR). The VLR then waits for the MAP_PROCESS_ACCESS_REQUEST service request, which is confirmation that paging was successful.

If no location information is available in the VLR, e.g. due to previous VLR recovery, the service MAP_SEARCH_FOR_MS is invoked, this request will just contain the IMSI of the subscriber. The VLR then waits for the MAP_SEARCH_FOR_MS service confirm, in the case of success this will contain the Current Location Area Id of the subscribers actual location, causing updating of the Location Area Information in the VLR. In the case of error only an error cause is returned.

Completion of the Paging procedures is followed by the MAP_PROCESS_ACCESS_REQUEST service indication to the VLR, this is to initiate subscriber access to the network for the called subscriber. On receipt of this indication the VLR will commence the Access Request procedure (this is described in subclause 21). On successful completion of the Process Access Request procedure, the MAP_PROCESS_ACCESS_REQUEST service indication is confirmed to the MSC.

If CUG Interlock code is present (and optionally CUG Outgoing Access indicator) in the MAP_SEND_INFO_FOR_INCOMING_CALL service indication, the VLR will translate the CUG Interlock code to the CUG Index (see GSM 03.85 for the rules regarding the provision of CUG Index).

In addition, if the Call Forwarding on No Reply service is active the NRCT value will be supplied along with the other parameters of the MAP_COMPLETE_CALL service request.

The other parameters for call set up will be sent to the MSC using MAP_COMPLETE_CALL service request. There will be no response from the MSC for successful completion of Complete Call. If no errors are reported by the MSC the VLR will set the result parameter of MAP_SEND_INFO_FOR_INCOMING_CALL service response to success and return this to the MSC. This terminates the procedures in the VLR.

Data or parameter errors

If any parameter or data errors are detected by the VLR during the MS terminating call set up procedures, the MAP service user error is set (either "Data Error" or "Unexpected Data Value") and returned. The procedure is then terminated in the VLR.

Service Provider Failure

Once a dialogue between the MSC and VLR is established, a MAP_P_ABORT service indication may be received, at any stage, from the MAP Service Provider. This will abort the dialogue (see clause 5 for a definition of the abort reasons) and the procedure in the VLR terminated.

Call Released by Calling Subscriber

The call may be released by the calling subscriber at any stage during the MS terminating call set up procedure. The VLR will be informed of this event by the MSC sending a MAP_U_ABORT service indication. The VLR will then terminate all procedures associated with this call, this may include sending a MAP_U_ABORT to the HLR if CUG information has been requested.

MS Data Not Confirmed by HLR

If the MSRN is known to the VLR but the indicator "Confirmed by HLR" (HLR_CONF_IND) is set to "Not Confirmed", the user error will be set to "System Failure" and the Result parameter set to Error. This information is returned to the MSC in the MAP_SEND_INFO_FOR_INCOMING_CALL and the procedure in the VLR terminated.

Called Subscriber Not Reachable

If the called subscriber is not reachable, either because the IMSI Detached Flag is set or because the called subscriber is not allowed to roam into the current location area, the VLR will check if the call forwarding on MS not reachable service is active. If the service is active and the call has not been forwarded as many times as allowed, the forwarding address is returned to the MSC in the MAP_SEND_INFO_FOR_INCOMING_CALL service response. The procedure is then terminated in the VLR.

If the call forwarding service is not active the user error is set to "Absent Subscriber" and this is returned to the MSC in the MAP_SEND_INFO_FOR_INCOMING_CALL service response. The procedure is then terminated in the VLR.

Forwarding Violation

If a call forwarding service is active but the call has already been forwarded as many times as allowed the MAP_SEND_INFO_FOR_INCOMING_CALL service response user error will be set to "Forwarding Violation". The result parameter is also set to Error. This information is returned to the MSC and the procedure in the VLR is terminated.

Access Request Failure

The Access Request procedure may fail for several reasons, either the access request is not accepted (i.e. subscriber is not recognised or roaming is not allowed) or one of the associated procedures fails (i.e. Authentication, Ciphering, IMEI Check or TMSI reallocation). The VLR will then terminate the procedure for this call by setting the MAP_SEND_INFO_FOR_INCOMING_CALL service parameter user error to "System Failure" or "Absent Subscriber" (depending on the error returned by the Access Request Procedure) and the Result parameter to Error. This is then returned to the MSC and the procedure in the VLR terminated.

Paging or Call Set up Failure

These procedures may fail for a number of reasons:

1)   Absent Subscriber

If either procedure fails due to absent subscriber the VLR will then determine whether the Call Forwarding on MS Not Reachable service is active. If the service is active, the call has not been forwarded more times than allowed, and a CUG call authorisation check is not required, the VLR will provide the forwarding address to the MSC in MAP_SEND_INFO_FOR_INCOMING_CALL service response. The procedure is then terminated in the VLR.

If CUG information is provided and/or the forwarding subscriber is a CUG subscriber then the VLR will perform a CUG call authorisation check on the forwarding party's outgoing access. The procedures for authorising a CUG call with call forwarding interaction are described in more detail in GSM 03.85. On successful completion of the CUG call authorisation check the VLR will provide the forwarding address and the new CUG Information (i.e. output of the CUG Call Authorisation check) to be used on the forwarding leg of the call, to the MSC in MAP_SEND_INFO_FOR_INCOMING_CALL service response. The procedure is then terminated in the VLR.

If the Call Forwarding on MS Not Reachable service is not active, the MAP_SEND_INFO_FOR_INCOMING_CALL service user error parameter is set to "Absent Subscriber". This is then returned to the MSC and the procedure in the VLR terminated.

2)   Radio congestion or No Subscriber Reply

The failure Radio Congestion will only occur with the Call Set up procedure.

The VLR will determine if a Call Forwarding service is active (e.g. Call Forwarding on MS Not Reachable in the case of radio congestion and Call Forwarding on No Reply for No Subscriber Reply). If the appropriate Call Forwarding service is active, the call has not been forwarded more times than allowed, and a CUG authorisation check is not required, the VLR will provide the forwarding address to the MSC in MAP_SEND_INFO_FOR_INCOMING_CALL service response. The procedure is then terminated in the VLR.

If CUG information is provided and/or the forwarding subscriber is a CUG subscriber then the VLR will perform a CUG call authorisation check on the forwarding party's outgoing access. The procedures for authorising a CUG call with call forwarding interaction are described in more detail in GSM 03.85. On successful completion of the CUG call authorisation check the VLR will provide the forwarding address to the MSC in MAP_SEND_INFO_FOR_INCOMING_CALL service response. The procedure is then terminated in the VLR.

If a Call Forwarding service is not active, the MAP_SEND_INFO_FOR_INCOMING_CALL service user error parameter is set to "Impossible Call Completion". This is then returned to the MSC and the procedure in the VLR terminated.

3)   Busy Subscriber

Three scenarios need to be considered in this category of failure:

-    Network Determined User Busy (NDUB)
     This condition occurs when a call is about to be offered and the information (i.e., traffic) channel (Bm or Lm) is busy/not available and the maximum number of total calls has been reached. The value of the maximum number of calls is 1 for the basic call. When the supplementary service Call Waiting is applicable, the value is n+1, where n is the maximum number of calls that can be waiting (reference GSM 02.01).

If this error is reported by the MSC, the VLR will check if the Call Forwarding on Busy service is active. If the service is active, and the call has not been forwarded more times than allowed, and a CUG authorisation check is not required, the VLR will provide the forwarding address to the MSC in MAP_SEND_INFO_FOR_INCOMING_CALL service response. The procedure is then terminated in the VLR.

If CUG information is provided and/or the forwarding subscriber is a CUG subscriber then the VLR will perform a CUG call authorisation check on the forwarding party's outgoing access. The procedures for authorising a CUG call with call forwarding interaction are described in more detail in GSM 03.85. On successful completion of the CUG call authorisation check the VLR will provide the forwarding address to the MSC in MAP_SEND_INFO_FOR_INCOMING_CALL service response. The procedure is then terminated in the VLR.

If Call Forwarding on Busy is not active, the MAP_SEND_INFO_FOR_INCOMING_CALL service user error is set to "Impossible Call Completion". This is then returned to the MSC and the procedure in the VLR terminated.

-    More Calls Allowed
     This condition occurs when a call is about to be offered and the called MS is already busy but the maximum number of total calls has not yet been reached. (reference GSM 02.01).

     If this error is reported by the MSC, the VLR will first check if the Call Waiting service is active. If the service is not active the VLR will then check if the Call Forwarding on Busy service is active. If this service is also not active then the procedure in the VLR for this call will be terminated and MAP_SEND_INFO_FOR_INCOMING_CALL service response is returned to the MSC with the Error Cause set to Impossible Call Completion.

     If the Call Waiting service is not active but the Call Forwarding on Busy service is active, then the procedure will continue as if the Network Determined User Busy error had been returned, see above.

     If the Call Waiting service is active, the VLR will construct a Call Waiting request. If CUG Information is provided, the VLR will first translate the interlock code to the CUG Index. If there is no match, the MAP_SEND_INFO_FOR_INCOMING_CALL service response is returned to the MSC with the user error set to "System Failure". The procedure in the VLR is then terminated.

     If there is a match, the VLR will then check if the Call Forwarding on No Reply service is active. If this service is active and the call has not been forwarded more times that allowed, the No Reply Condition Timer will be provided in the Call Waiting request (this indicates to the MSC that Call Forwarding on No Reply service is active). Should the call have already been forwarded as many times as allowed then a No Reply Condition Timer value will not be supplied.

     On successful completion of the CUG and call forwarding checks, the VLR will instruct the MSC to commence Call Waiting by sending MAP_PROCESS_CALL_WAITING service request. If no errors are returned by the MSC, the VLR will return MAP_SEND_INFO_FOR_INCOMING_CALL service response, with the result set to Success. The procedure in the VLR is then terminated.

     If Call Waiting was not completed successfully the MSC will respond with MAP_PROCESS_CALL_WAITING service confirm. If the user error indicates "Busy Subscriber", the procedure continues as described for Network Determined User Busy above. If the error No Subscriber Reply is returned, the procedure continues as described under No Subscriber Reply (item 2 above). All other errors returned by the MSC will result in the VLR returning Impossible Call Completion user error in MAP_SEND_INFO_FOR_INCOMING_CALL service response. The procedure in the VLR is then terminated.

-        User Determined User Busy (UDUB)
         This condition occurs when a call is offered to an MS and the MS responds "User Busy" because
         the subscribers resources (terminal or person using them) are busy (reference GSM 02.01). This
         busy condition is treated in exactly the same way as Network Determined User Busy, see above.

If the Busy Subscriber error cause does not indicate one of the above states then the VLR will treat this
error as More Calls Allowed.

CUG Failure

Failure will be due either to the VLR being unable to convert the Interlock code into an Index, or the CUG
Call Authorisation check for call forwarding interaction may not be successful. This will result in VLR
terminating the procedure for this call by setting MAP_SEND_INFO_FOR_INCOMING_CALL service user
error to "CUG Reject" in the case of CUG authorisation failure, and "System Failure" for failure to convert
the CUG Interlock Code to an Index. This is then sent to the MSC, the procedure in the VLR is then
terminated.

Figure 18.3/5 (sheet 1 of 6): Process ICS_VLR
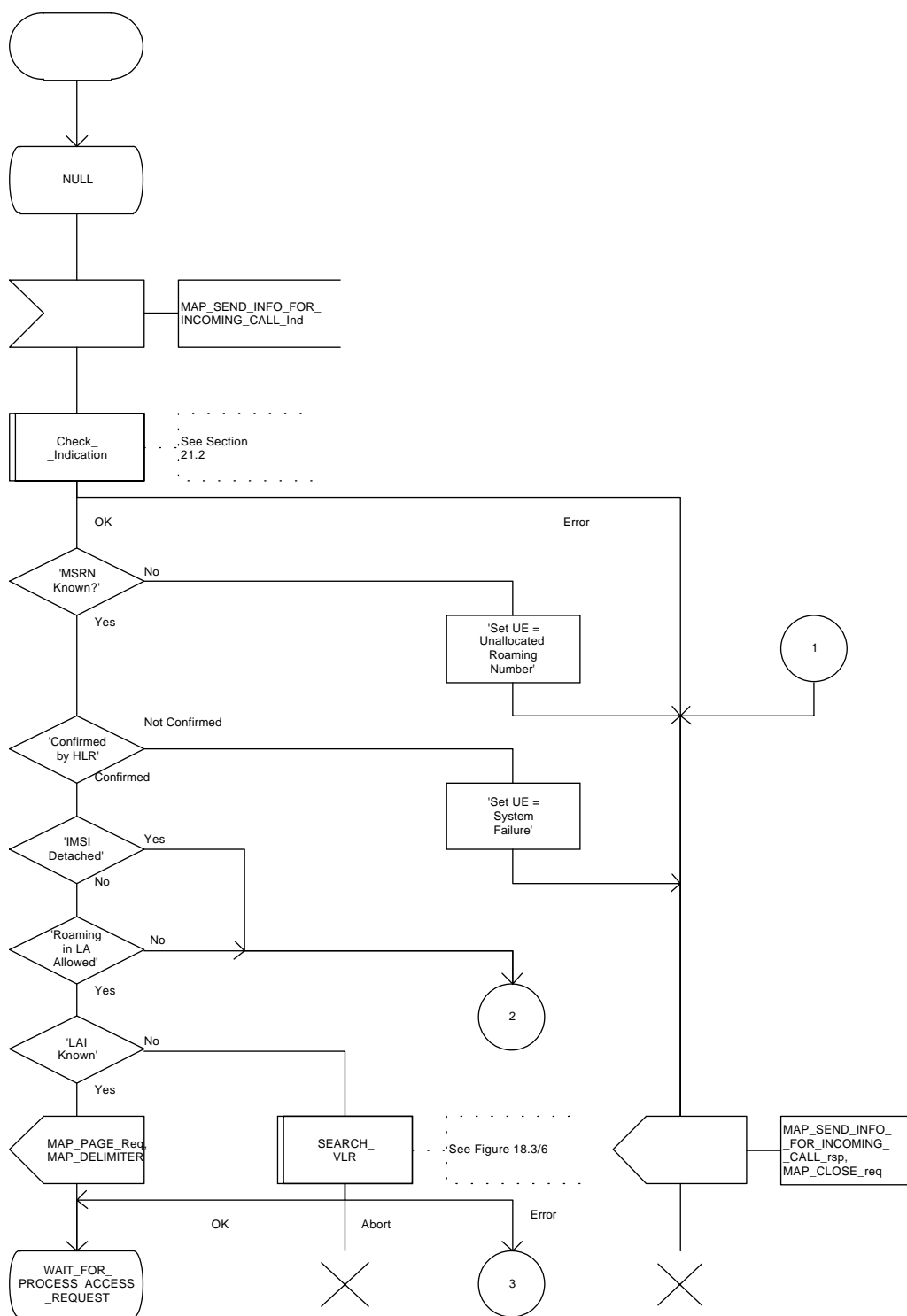
Figure 18.3/5 (sheet 2 of 6): Process ICS_VLR
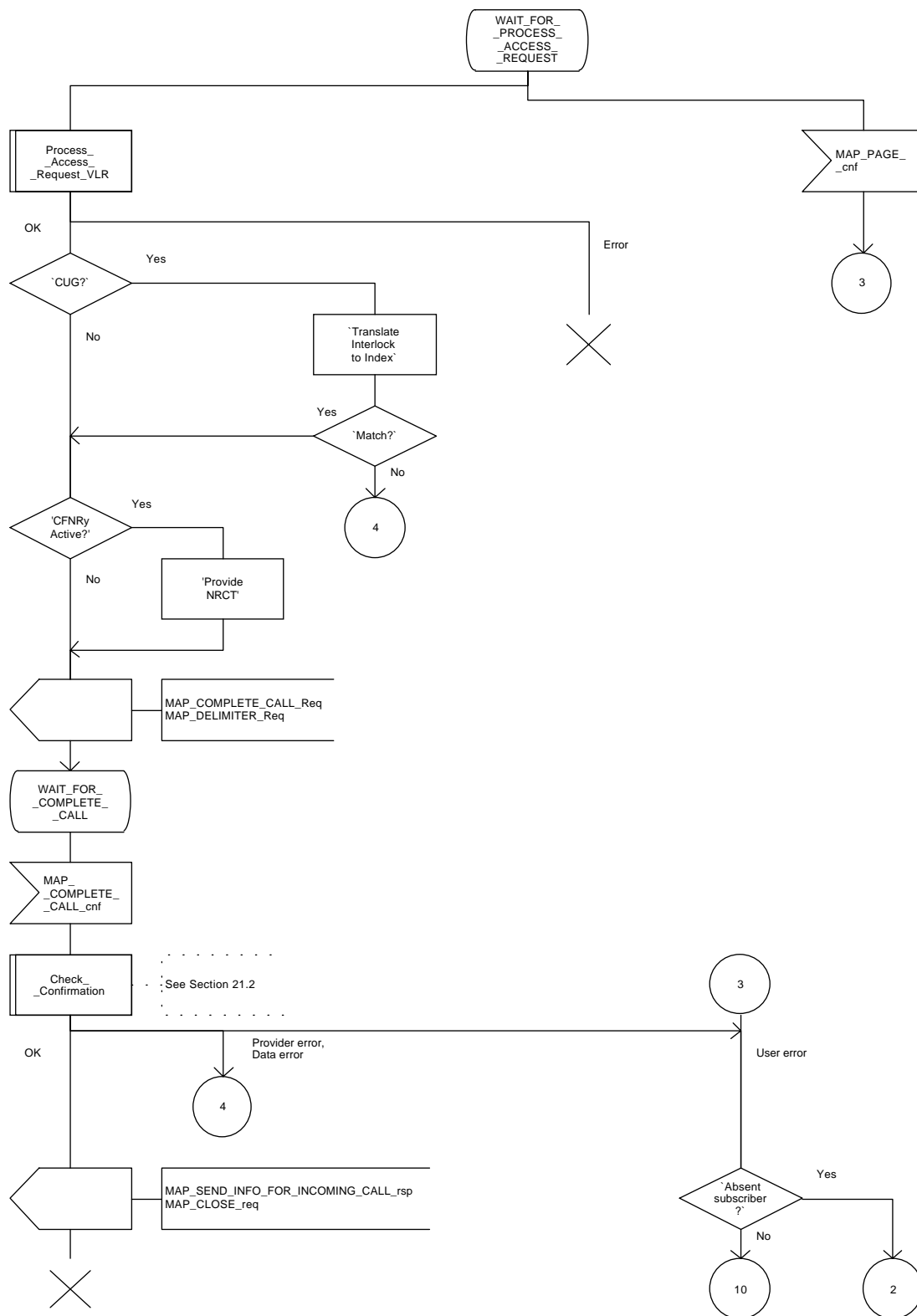
Figure 18.3/5 (sheet 3 of 6): Process ICS_VLR
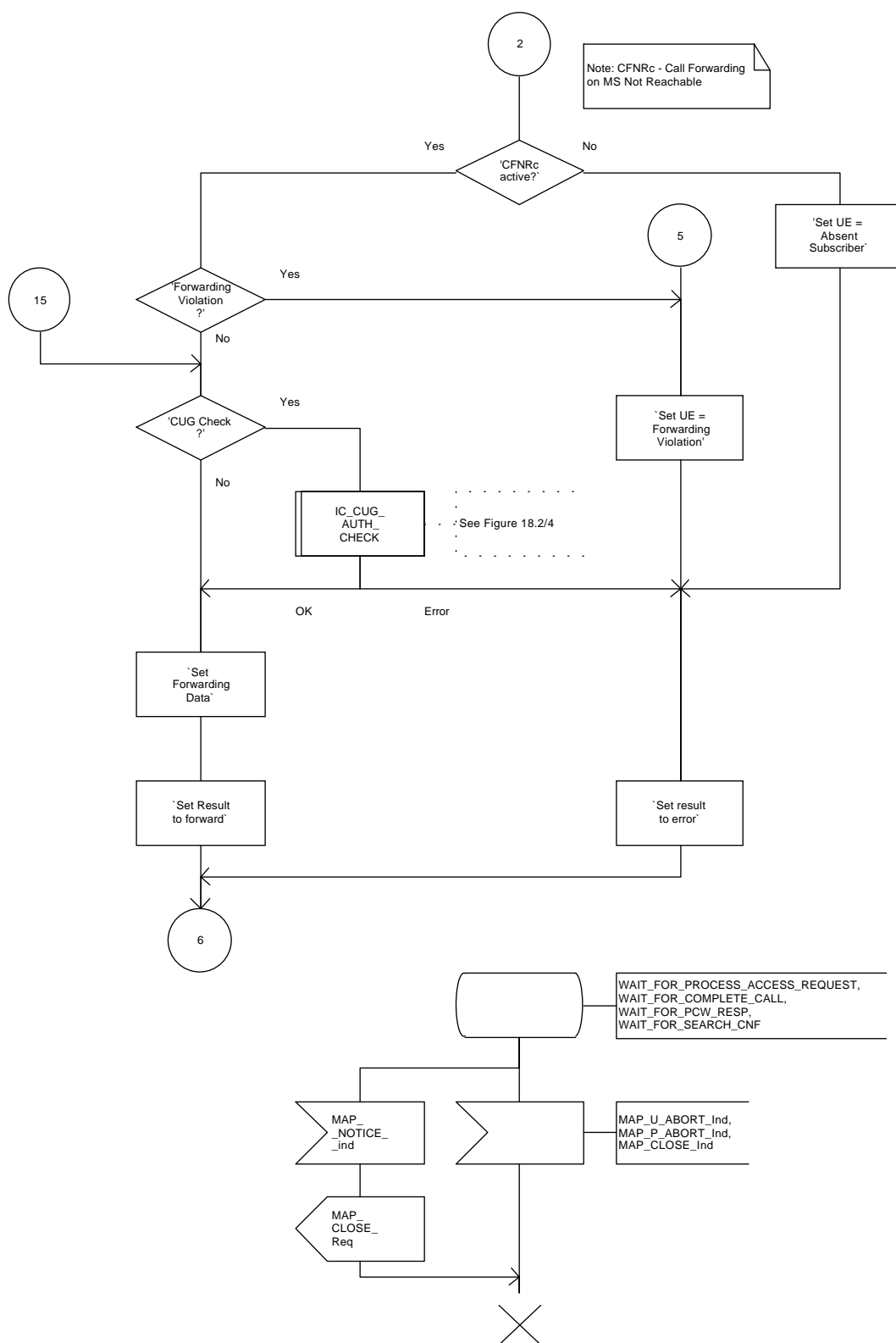
Figure 18.3/5 (sheet 4 of 6): Process ICS_VLR
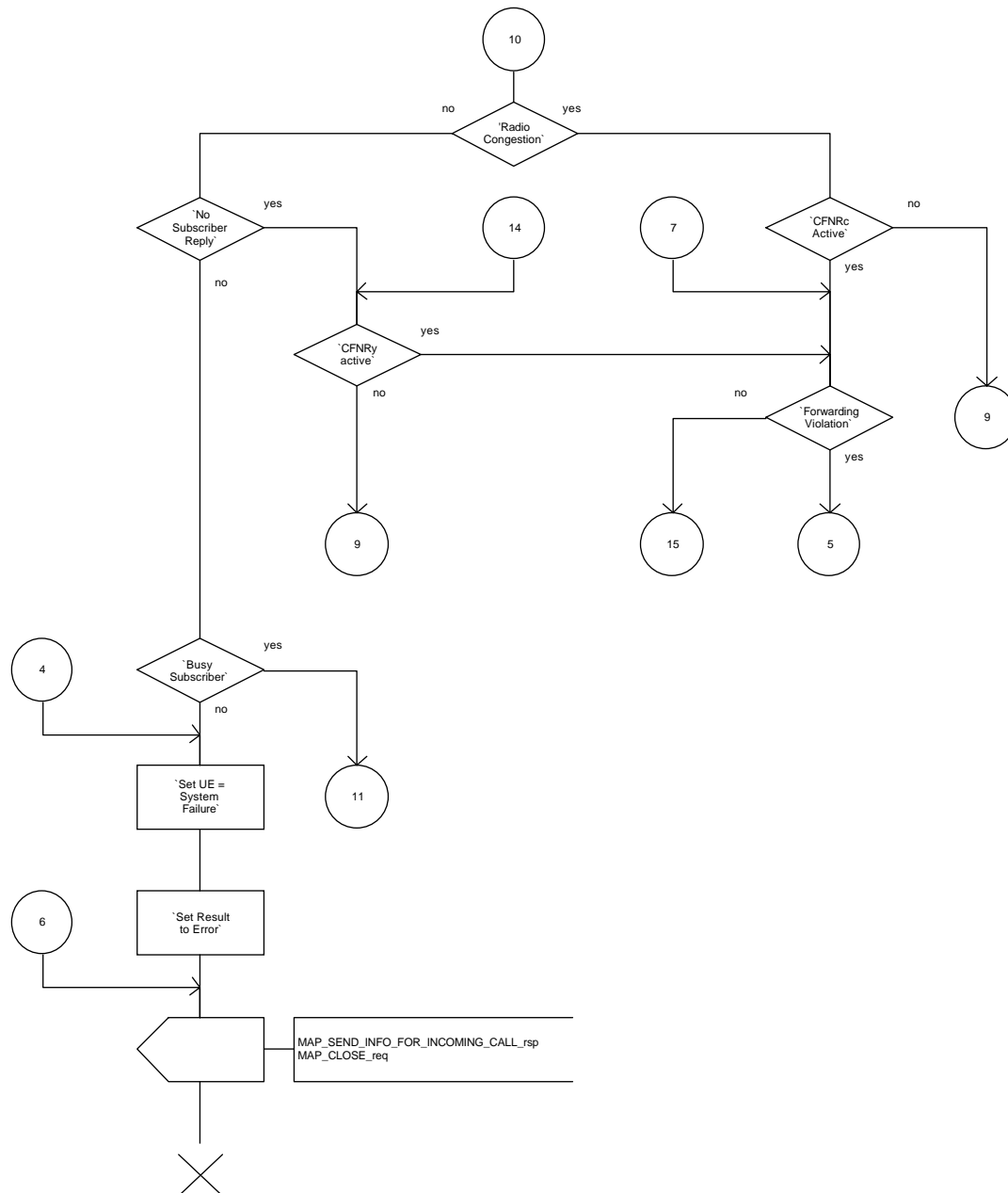
Figure 18.3/5 (sheet 5 of 6): Process ICS_VLR
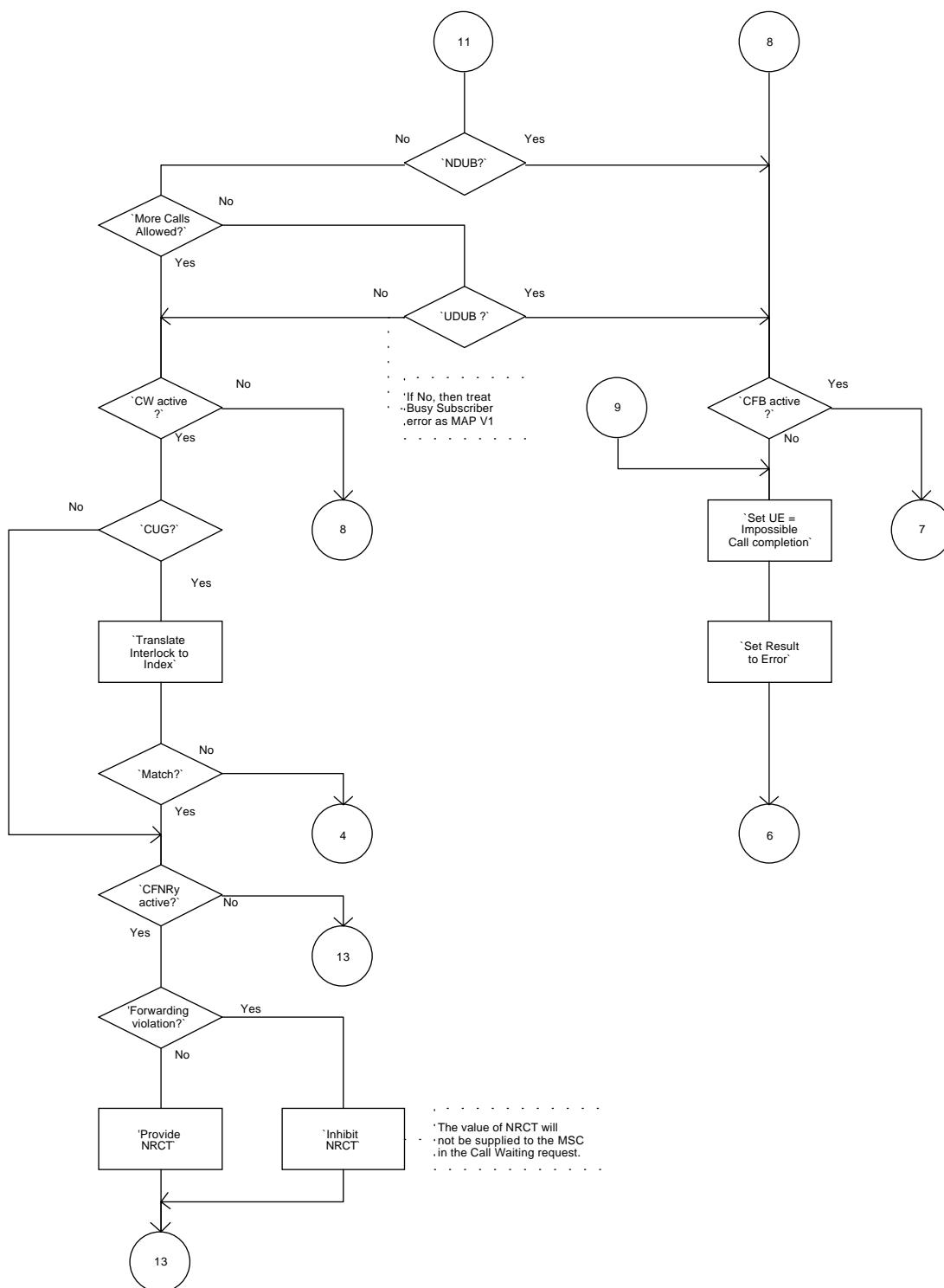
Figure 18.3/5 (sheet 6 of 6): Process ICS_VLR
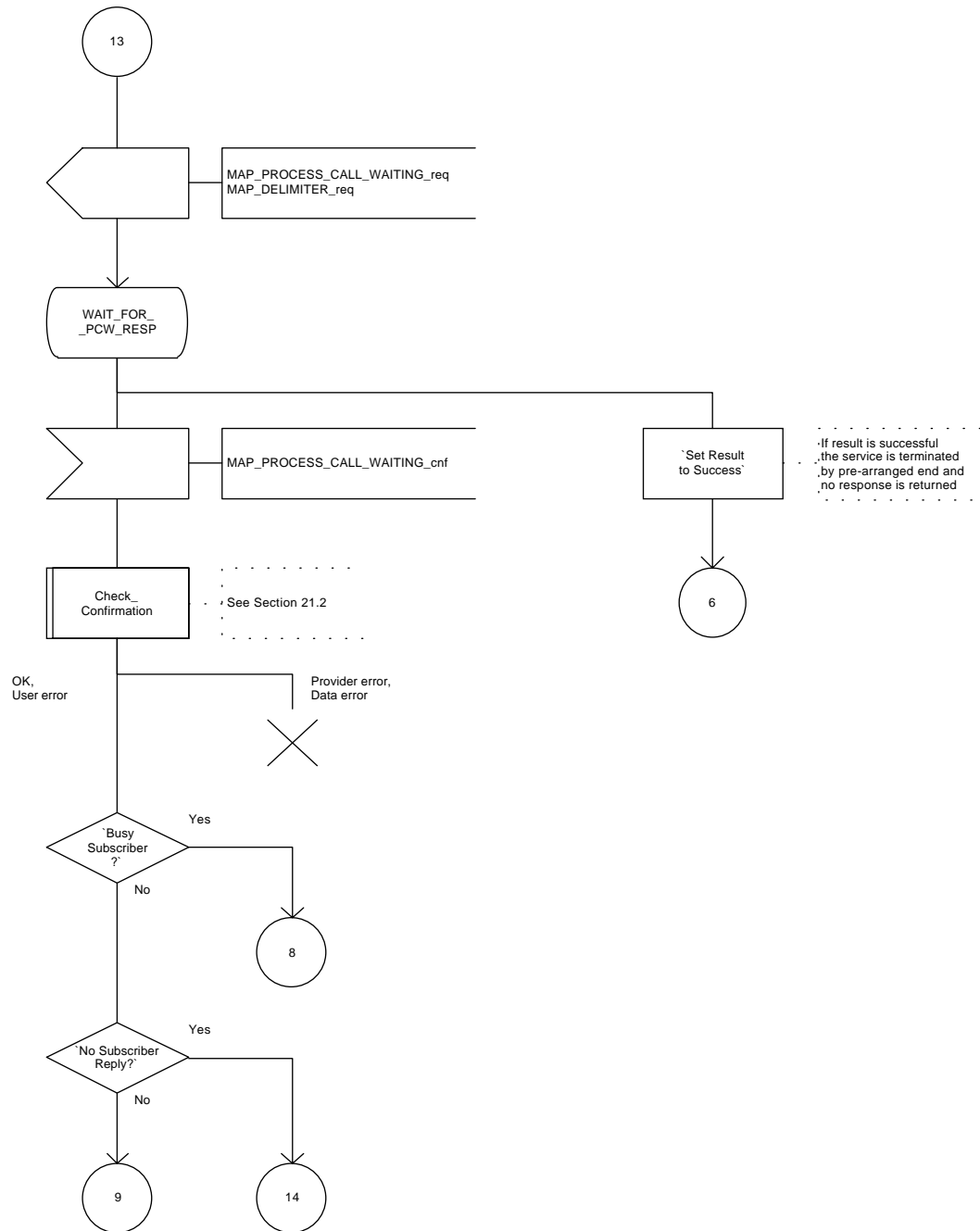
Figure 18.3/6: Macro SEARCH_VLR
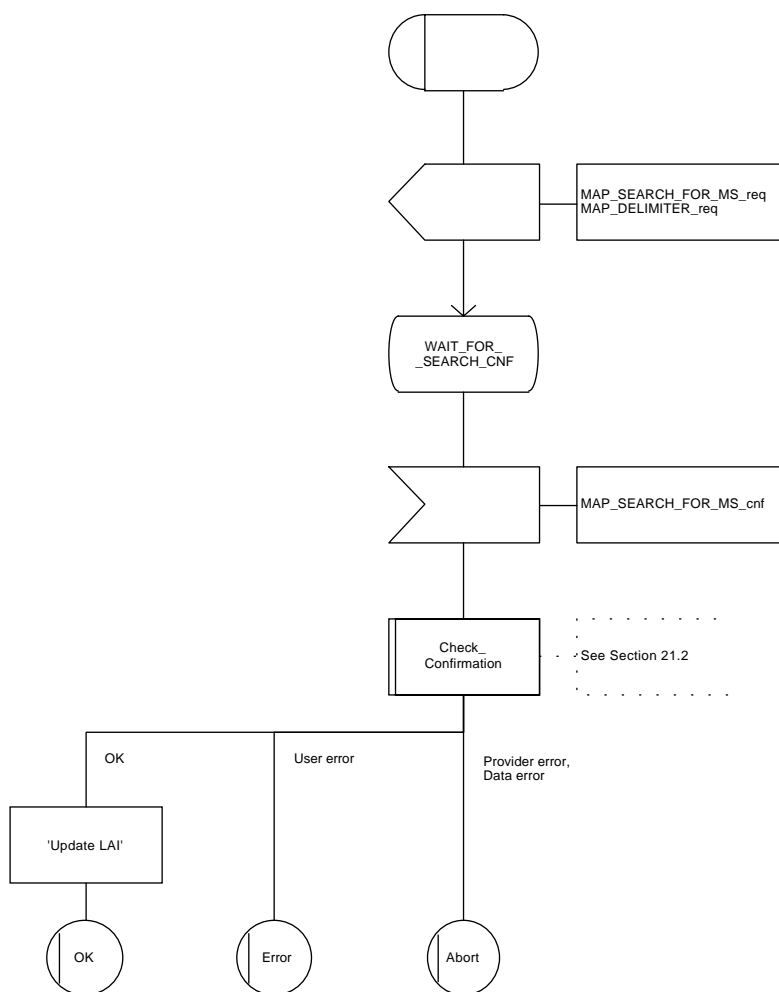
18.4   MS originated calls

18.4.1   General

The call can be established in two ways in the GSM. Either with the early assignment or with the off air call set up (OACSU). The difference between these is that in the latter case the radio channel is reserved at the last possible moment, i.e. when the called subscriber answers. In this subclause only the procedure with the early assignment is shown. The procedure is shown in figure 18.4/1, this is an example sequence which assumes successful operation.
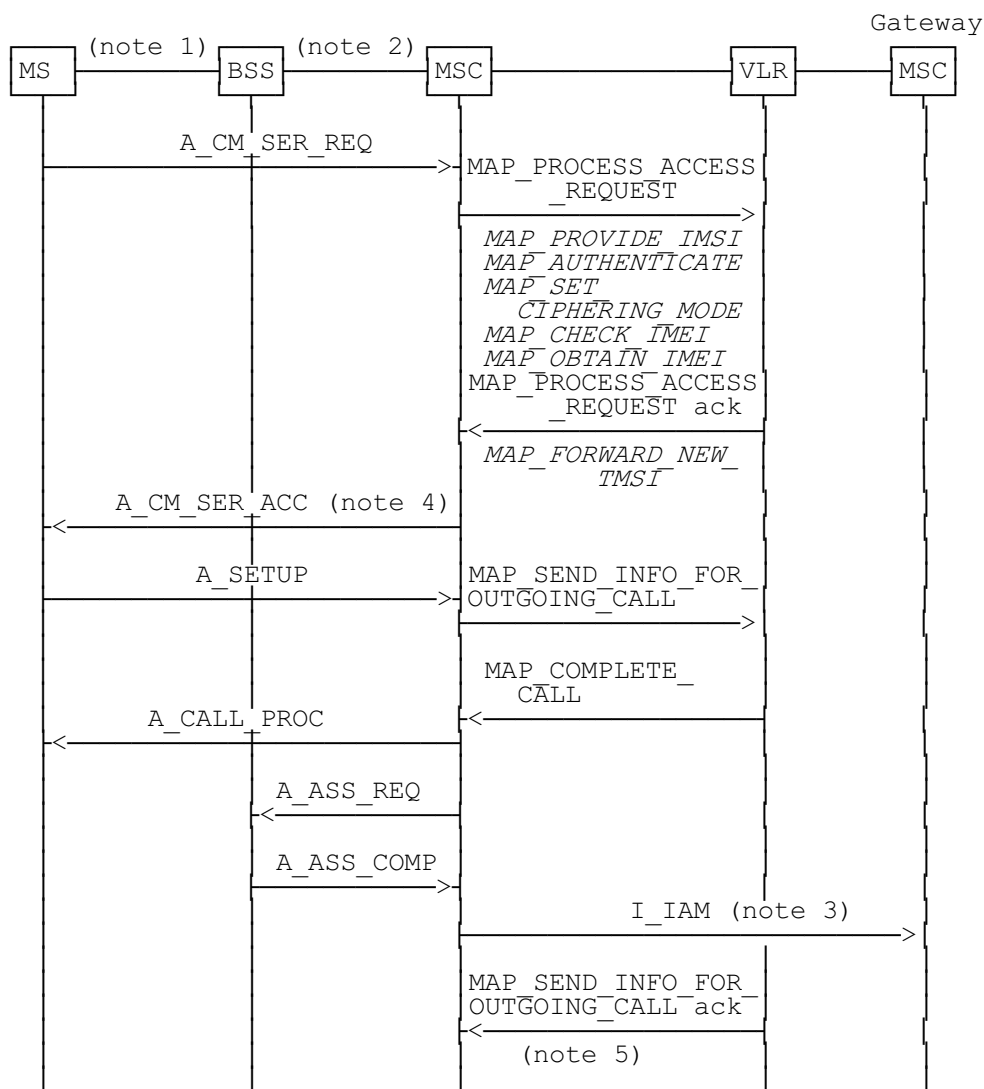
xxx = Optional Procedure

Figure 18.4/1: Procedure for MS originating call set-up

Notes to figure 18.4/1:

NOTE 1: The BSS and MSC message flow shown in this diagram is only an example of the interworking possibilities. For further details of the procedures between the MSC and MS refer to GSM 04.08 (Mobile Radio Interface Layer 3 Specification).

NOTE 2: The MS and MSC message flow shown in this diagram is only an example of the interworking possibilities. For further details of the procedures between the MSC and BSS refer to GSM 08.08 (base station to switch Interface, Layer 3).

NOTE 3: TUP and ISUP may be used in signalling between MSCs, depending on the network type between the MSCs. For further details on the TUP and ISUP procedures refer to the following CCITT Recommendations:

Q.721-Q.725 Telephone User Part (TUP);
Q.761-Q.764 ISDN User Part (ISUP).

It is assumed that signalling towards the fixed network only uses one message, IAM (Initial Address Message). Examples of other interworking cases are not shown in this standard (for further information see e.g. CCITT Recommendation Q.699).

NOTE 4: Acknowledgement of A-CM-SER-REQ can also be performed by invocation of ciphering mode setting (A-CIPH-MOD-CMD).

NOTE 5: In some circumstances negotiation of bearer service or teleservice capabilities may be required to set up an MS originated call. The interworking between the VLR and MSC to control this negotiation is not specified in the MAP.

The following MAP services are used during outgoing call set up:

MAP_PROCESS_ACCESS_REQUEST	see clause 6;
MAP_SEND_INFO_FOR_OUTGOING_CALL	see clause 8;
MAP_COMPLETE_CALL	see clause 8;
MAP_PROVIDE_IMSI	see clause 6;
MAP_SET_CIPHERING_MODE	see clause 6;
MAP_AUTHENTICATE	see clause 6;
MAP_FORWARD_NEW_TMSI	see clause 6;
MAP_CHECK_IMEI	see clause 6;
MAP_OBTAIN_IMEI	see clause 6.

18.4.2 Procedures in the MSC

The procedure is described in figures 18.4/2, 18.4/3 and 18.4/4. These procedures make use of a number of processes and macros not described in this subclause. The description of these can be found in the following subclauses:

Check_Confirmation	subclause 21.2;
Process_Access_Request_MSC	subclause 21.4.

Successful Outcome

On receipt of an A_CM_SERV_REQ indicating an outgoing circuit switched call set up from the MS, the MSC will initiate a MAP_PROCESS_ACCESS_REQUEST service request to the VLR. The VLR will then invoke the Access Request procedures, this is described in clause 21.

Once the MAP_PROCESS_ACCESS_REQUEST service indication is accepted the MS will then send to the MSC the call control information (e.g. called party address and bearer capability), this is contained in an A_SETUP message. From this information the MSC will construct a MAP_SEND_INFO_FOR_OUTGOING_CALL service request and send it to the VLR, to request the subscriber parameters for handling the call.

If the request was successful the VLR will respond with a MAP_COMPLETE_CALL service request. This is the instruction to the MSC to proceed with the call set up. To indicate to the calling MS that the call is being proceeded, the MSC will send an A_CALL_PROC message. This message and the procedures associated with it are described in more detail in GSM 04.08.

To complete the call set up procedures, the MSC will commence the connection establishment phase of the call by sending an Initial Address Message (I_IAM) to the Gateway MSC. If this call is to be treated as a CUG call, then the CUG information (Interlock Code and optionally the Outgoing Access indicator) provided in the MAP_COMPLETE_CALL service indication, should be used to construct the CUG information in the Initial Address Message (I_IAM). The reason for this, is because it is possible that the CUG Information returned by the VLR may be different to the CUG Information originally supplied by the MSC, due to the changes required by the CUG Call Authorisation check. The CUG Index may also be returned by the VLR, the presence of this parameter in the MAP_COMPLETE_CALL service indication is confirmation that an implicit CUG has been invoked. This information will not be passed onto the called user but as an operator option, this information may be returned to the Calling User. See GSM 03.85 for further details of the CUG options and checks.

Service Provider Failure

Once the dialogue between the MSC and the VLR is established a MAP_P_ABORT service indication may be received at any stage from the MAP Service Provider. This aborts the dialogue, see clause 5 for a definition of the abort reason.

The MSC then clears the call by sending a Disconnect message (A_DISC) to the MS and a Clear Command (A_CLR_CMD) to the BSS to release the associated dedicated resource. The clear procedures are described in more detail in GSM 04.08 (MS) and GSM 08.08 (BSS). The procedure in the MSC is then terminated.

Call Released by Calling Subscriber

The call may be released at any time by the calling subscriber, this is indicated to the MSC by the MS sending a Release Message (either A_REL, A_REL_COM or A_DISC, see GSM 04.08 for a more detailed description of the options available). To terminate the call set up procedures for this call the MSC will send a MAP_U_ABORT service request to the VLR, this will indicate a User Reason of Application Procedure Cancelled and a Diagnostic value of Call Released. The procedure in the MSC is then terminated.

Process Access Request Failure

The VLR may reject the Process Access Request, either the Access Request is not accepted or one of the associated procedures called by the Access Request procedure may fail (i.e. Authentication, Ciphering, IMEI check or TMSI reallocation). Regardless of the reason, the call will be cleared by the MSC rejecting the MS Service Request. This is indicated to the MS by the MSC sending a CM Service Reject message (A_CM_SERV_REJ). The procedure in the MSC is then terminated.

Calling Subscriber Barred

The calling subscriber may be barred from making outgoing calls or roaming (see GSM 02.41 for a description of the call barring conditions), the MSC is informed of this condition by the VLR in the MAP_SEND_INFO_FOR_OUTGOING_CALL service confirm, the user error will be set to "Call Barred" with the barring cause indicating either "Call Barred" or "Call Barred for Administrative Reasons". The MSC will instruct the MS to release the call by sending a Release Complete message (A_REL_COMP). The procedure in the MSC is then terminated.

Assignment Failure

The BSS may fail to seize a radio channel for one of the following reasons:

- Use of the terrestrial channel selected by the MSC is impossible;

- Radio Congestion.

The MSC will be informed of this failure by the Assignment Failure message (A_ASS_FAIL). The call is then released by the MSC sending a Disconnect message (A_DISC) to the MS to initiate the clear procedure.

The MSC will then inform the VLR of this failure by setting the MAP_COMPLETE_CALL service user error to "Radio Congestion". The VLR will return MAP_SEND_INFO_FOR_OUTGOING_CALL service response with the user error set to "System Failure". The procedure in the MSC is then terminated.

CUG Failure

The serving VLR will perform the outgoing call CUG authorisation, if the call fails this authorisation the user error "CUG Reject" is returned by the VLR in the MAP_SEND_INFO_FOR_OUTGOING_CALL service response. The MSC will instruct the MS to release the call by sending a Release Complete message (A_REL_COMP). The procedure in the MSC is then terminated.

Figure 18.4/2 (sheet 1 of 2): Process OCS_MSC
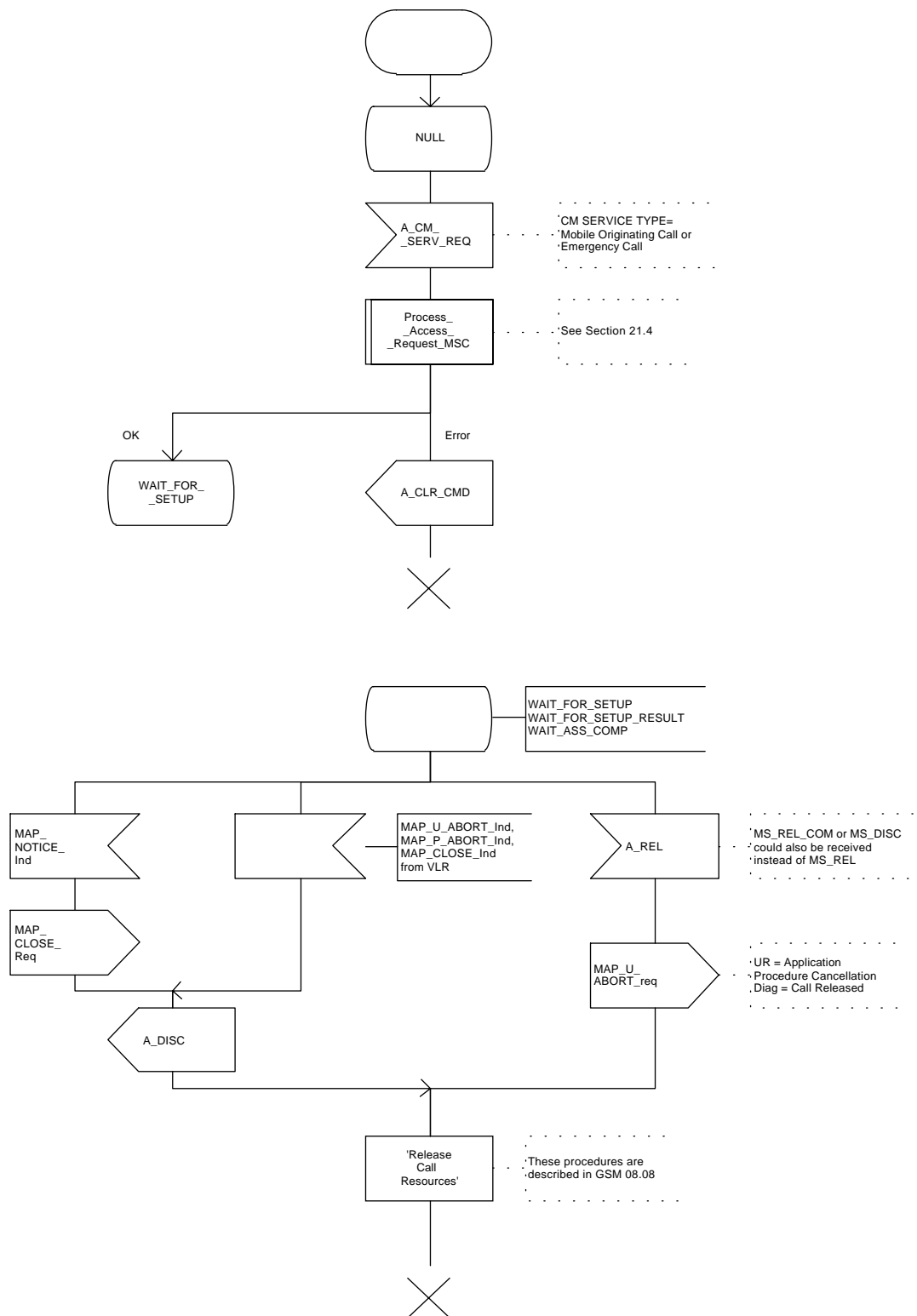
Figure 18.4/2 (sheet 2 of 2): Process OCS_MSC
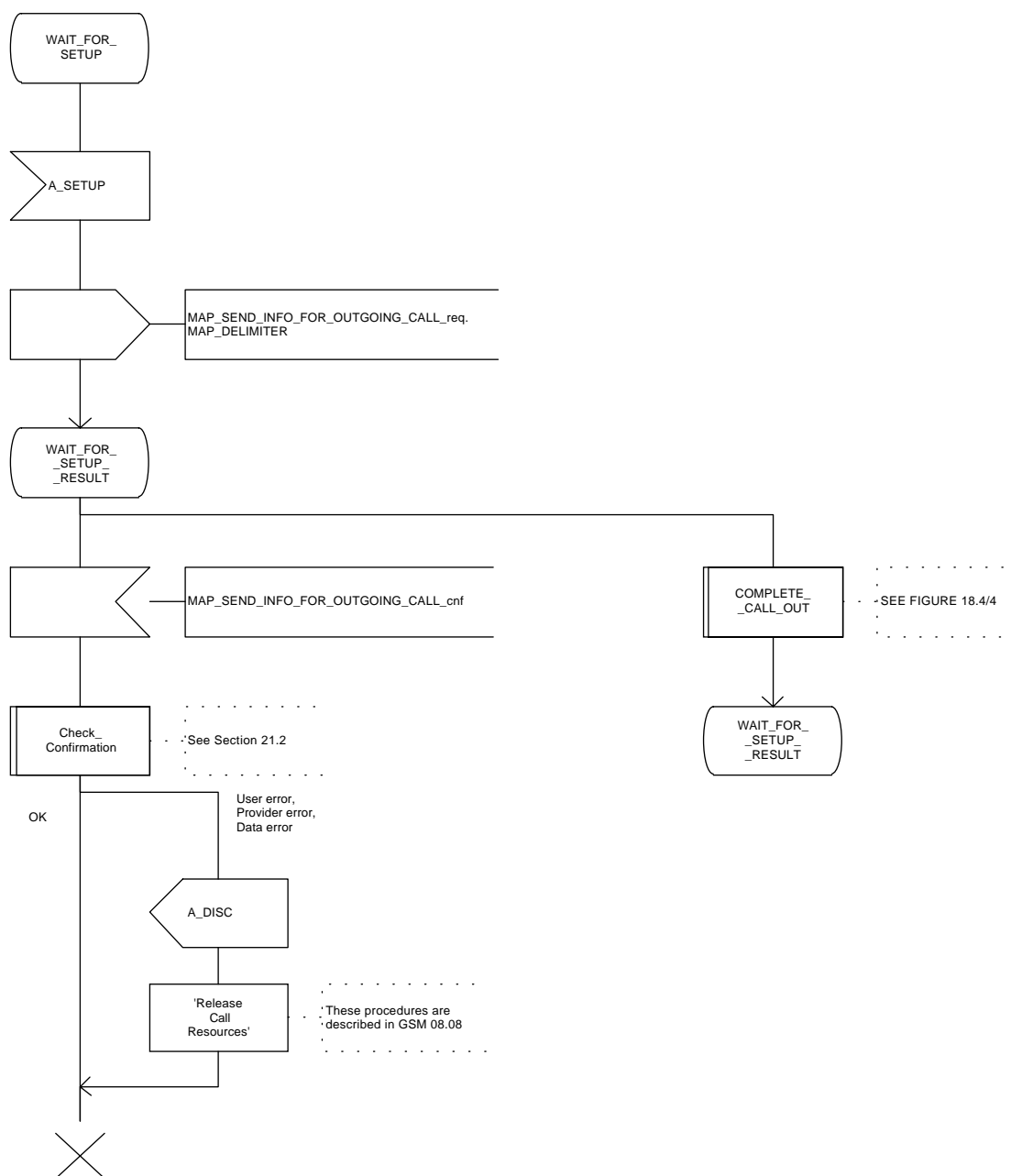
Figure 18.4/4: Macro COMPLETE_CALL_OUT
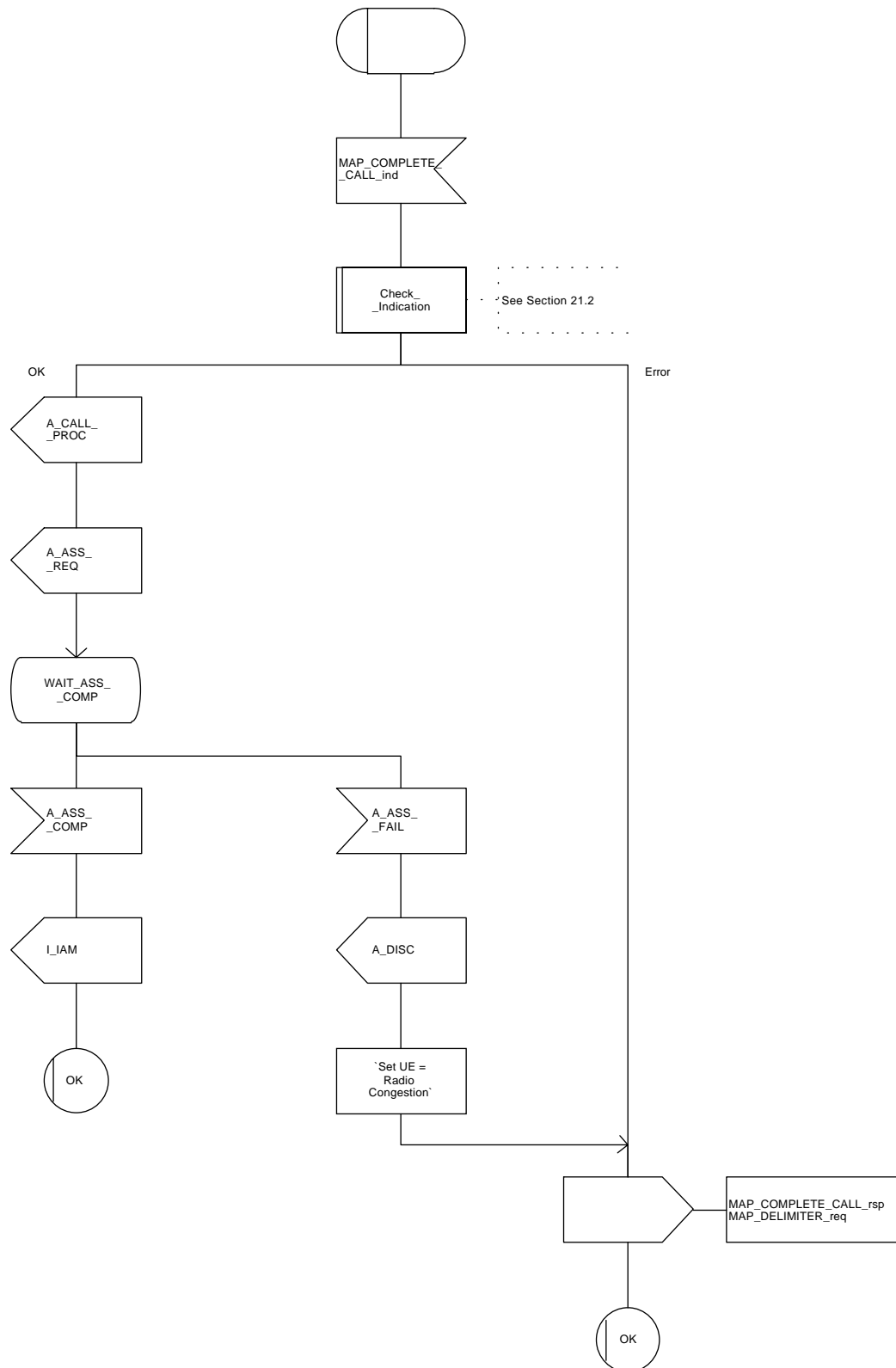
18.4.3 Procedures in the VLR

The procedure is shown in figures 18.4/5 and 18.4/6. These procedures make use of a number of processes and macros not described in this subclause. The description of these can be found in the following subclauses:

Check_Confirmation   subclause 21.2;
Process_Access_Request_VLR   subclause 21.4;
Receive_Open_Ind   subclause 21.1;
Check_Indication   subclause 21.2.

Successful Outcome

Once the procedures initiated by the MAP_PROCESS_ACCESS_REQUEST service request have been successfully completed, the MSC will then initiate a MAP_SEND_INFO_FOR_OUTGOING_CALL service request. This is to request the VLR to provide information to allow the MSC to handle an outgoing call set up request from an MS. The VLR will check the parameters and data for errors.

If a specific Bearer Service or Teleservice has been requested by the MS, the VLR will then perform the subscription check for the provision of the service. The VLR will also check the activation status of the supplementary services and the status of call barring services (i.e. barring of outgoing calls, conditional barring of outgoing calls or administrative reasons such as bad payer).

If the calling MS is part of a closed user group (CUG), the MSC will have passed any CUG information (received from the MS) to the VLR in the MAP_SEND_INFO_FOR_OUTGOING_CALL service request. The VLR will perform the authorisation checks on the call, this authorisation is determined by the information received from the MSC, the subscription information held by the VLR and the call scenario. It is possible that a CUG call may be implicitly invoked without CUG information being passed to the VLR in the MAP_SEND_INFO_FOR_OUTGOING_CALL service indication. In this instance, the VLR will just return the CUG_Index value to the MSC. The procedures for authorising a CUG call and the information to be returned to the serving MSC are described in more detail in GSM 03.85.

If the calling MS has passed all checks the VLR will instruct the MSC to set up the call by sending a MAP_COMPLETE_CALL service request to the MSC. This request may contain parameters related to the subscription, e.g. supplementary services and basic services. On successful completion of the MAP_COMPLETE_CALL service, this procedure is then terminated in the VLR.

Parameter or data errors

If the VLR detects any parameter or data errors in MAP_SEND_INFO_FOR_OUTGOING_CALL service indication from the MSC, the VLR will reject the service request. The MAP_SEND_INFO_FOR_OUTGOING_CALL service parameter user error will be set to either "Data Missing" or "Unexpected Data Value". This response is returned to the MSC and the procedures in the VLR will be terminated.

Service Provider Failure

Once a dialogue between the MSC and VLR or VLR and HLR is established, a MAP_P_ABORT service indication may be received, at any stage, from the MAP Service Provider. This will abort the dialogue, see clause 5 for a definition of the abort reasons.

Call Released by Calling MS

The call may be released by the calling subscriber at any stage during the outgoing call set up procedure. The VLR will be informed of this event by the MSC sending a MAP_U_ABORT service indication. The VLR will then terminate all procedures associated with the call.

Process Access Request Failure

If any of this procedure fails (e.g. Authentication, IMEI Check, Ciphering or TMSI reallocation) the VLR will terminate the outgoing call set up procedure for this call. The MSC will have already been informed of the failure by the Process Access Request Procedure (see clause 21).

Service Subscription Check Failure

If the Bearer Service or Teleservice requested by the MS for this call is not available. The VLR will set the MAP_SEND_INFO_FOR_OUTGOING_CALL service parameter user error to either "Teleservice Not Provisioned" or 'Bearer Service Not Provisioned". This confirmation is then returned to the MSC and the procedures for this call terminated in the VLR.

Calls Barred

If calls for this MS are barred (e.g. all outgoing calls, conditional barring of outgoing calls, roaming barred or calls barred due to Operator Determined Barring as defined in GSM 02.41), the VLR will terminate outgoing call set up procedures. The MAP_SEND_INFO_FOR_OUTGOING_CALL service parameter user error will be set to "Call Barred", this is then returned to the MSC.

CUG Failure

If the call fails the CUG Authorisation check (the procedures for authorising an outgoing CUG call are defined in GSM 03.85), CUG failure will be reported back to the MSC, the MAP_SEND_INFO_FOR_OUTGOING_CALL service parameter user error will be set to "CUG Reject". Outgoing call set up procedures for this call will then be terminated in the VLR.

Call Set up Failure

If, following a MAP_COMPLETE_CALL service request from the VLR, the MSC fails to set up the call this will be reported as an error in MAP_COMPLETE_CALL service confirm. Regardless of the reason for failure, the VLR will set the user error parameter of MAP_SEND_INFO_FOR_OUTGOING_CALL to "System Failure". This service response is then returned to the MSC and the procedures in the VLR terminated.

Figure 18.4/5 (sheet 1 of 2): Process OCS_VLR
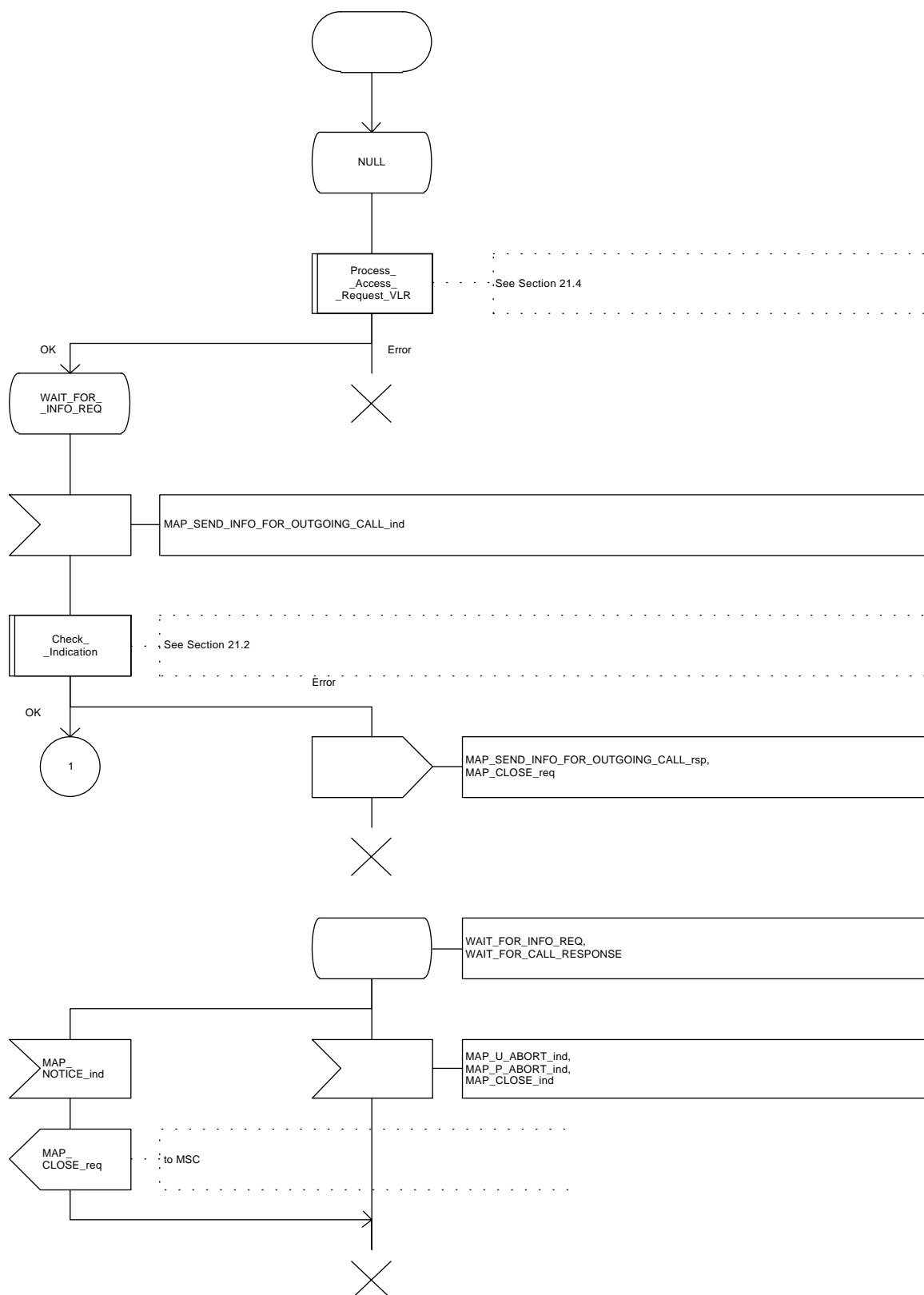
Figure 18.4/5 (sheet 2 of 2): Process OCS_VLR
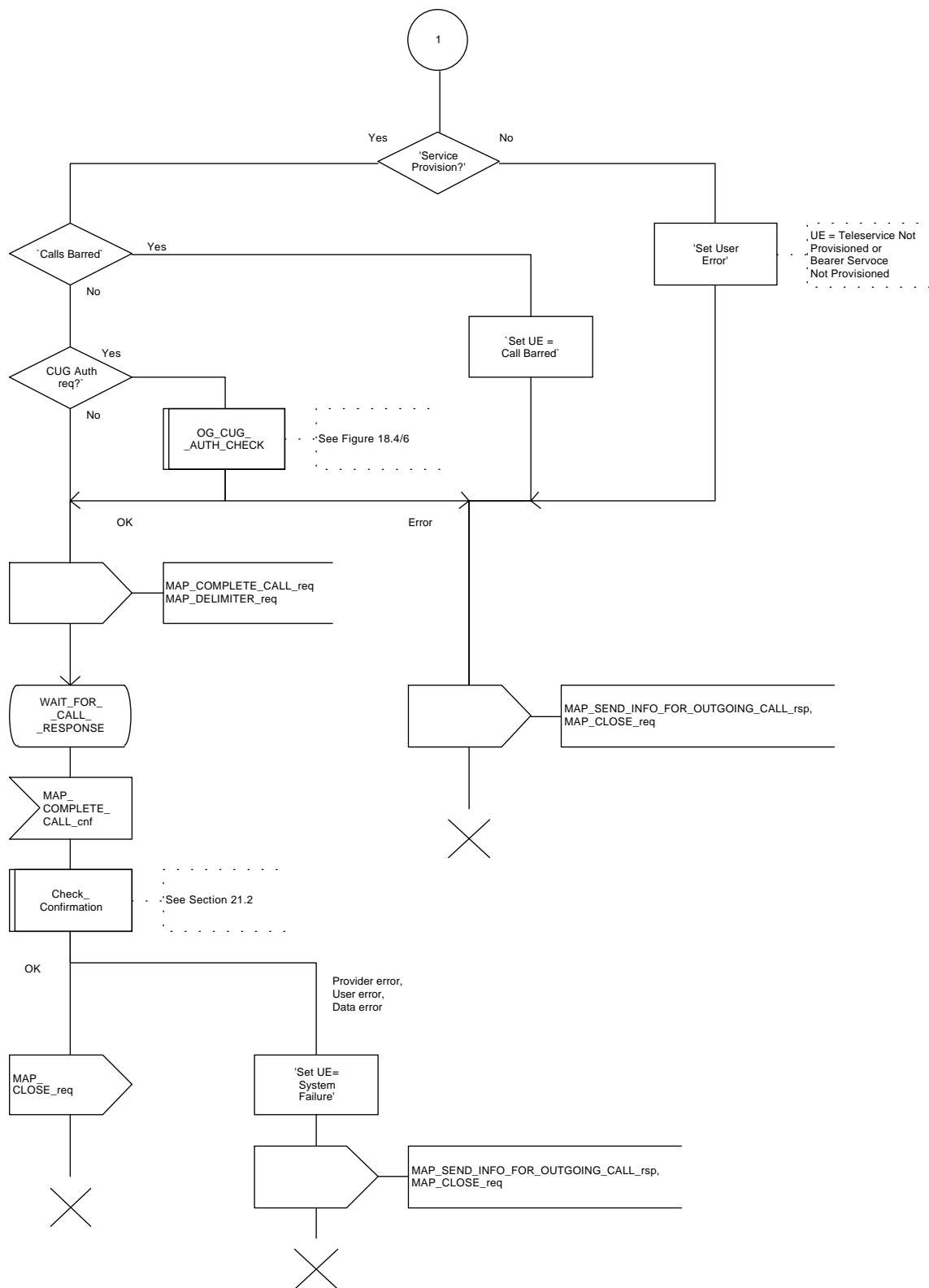
Figure 18.4/6: Macro OG_CUG_AUTH_CHECK
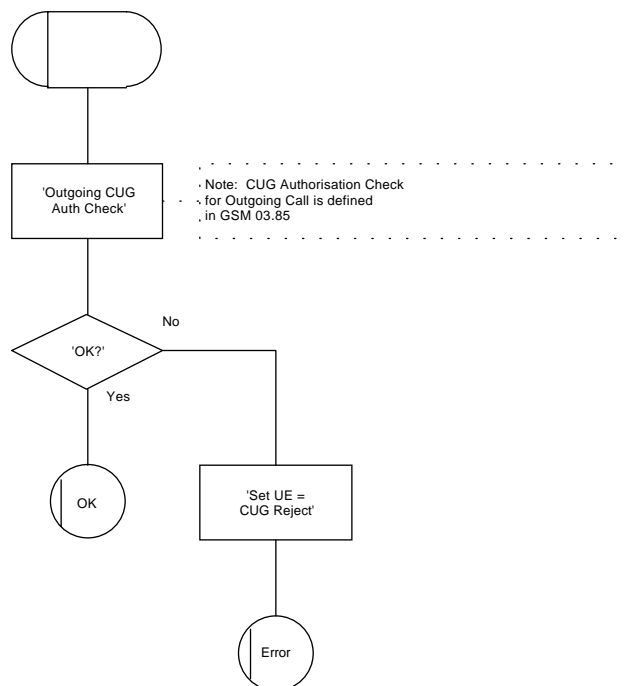
19 Supplementary services procedures

The following application contexts exist for handling of supplementary services:

- accessUnstructuredSsContext;

- accessFunctionalSsContext.

The accessUnstructuredSsContext refers to a simple MAP users, for which the corresponding MAP process can be identified by the MAP-Provider directly.

However, the accessFunctionalSsContext refers to a complex MAP-User consisting of several processes. For this user, a process co-ordinator is defined for each network entity, in order to identify the correct process to invoke. These processes open and validate the dialogue, then invoke the necessary operation-specific process. These processes are described below.

19.0 Functional supplementary service processes

19.0.1 Functional supplementary service process co-ordinator for MSC

Upon receipt of a CM-Service request with CM-service type = SS, the MSC initiates the process access request procedure towards the VLR as described in clause 21 of this ETS.

Once a CM connection is established, the MSC can handle supplementary service indications from the MS. Table 19.0/1 shows the co-ordinating process' reaction on receipt of specific SS service indications on the air interface. After the relevant process is invoked, the received air interface service indication is sent to that process. The creation of service requests on the basis of air interface messages is described in GSM 09.11.

Service indication received	Process invoked
A_REGISTER_SS_ind	REGISTER_SS_MSC
A_ERASE_SS_ind	ERASE_SS_MSC
A_ACTIVATE_SS_ind	ACTIVATE_SS_MSC
A_DEACTIVATE_SS_ind	DEACTIVATE_SS_MSC
A_INTERROGATE_SS_ind	INTERROGATE_SS_MSC
A_REGISTER_PASSWORD	REGISTER_PASSWORD_MSC

Table 19.0/1: Relationship between received service indication and invoked process in the MSC

Figure 19.0/1 shows the co-ordinating process in the MSC.

Figure 19.0/1

Figure 19.01/1: Supplementary Service Coordination process in the MSC, to identify which functional supplementary service process shall be invoked.
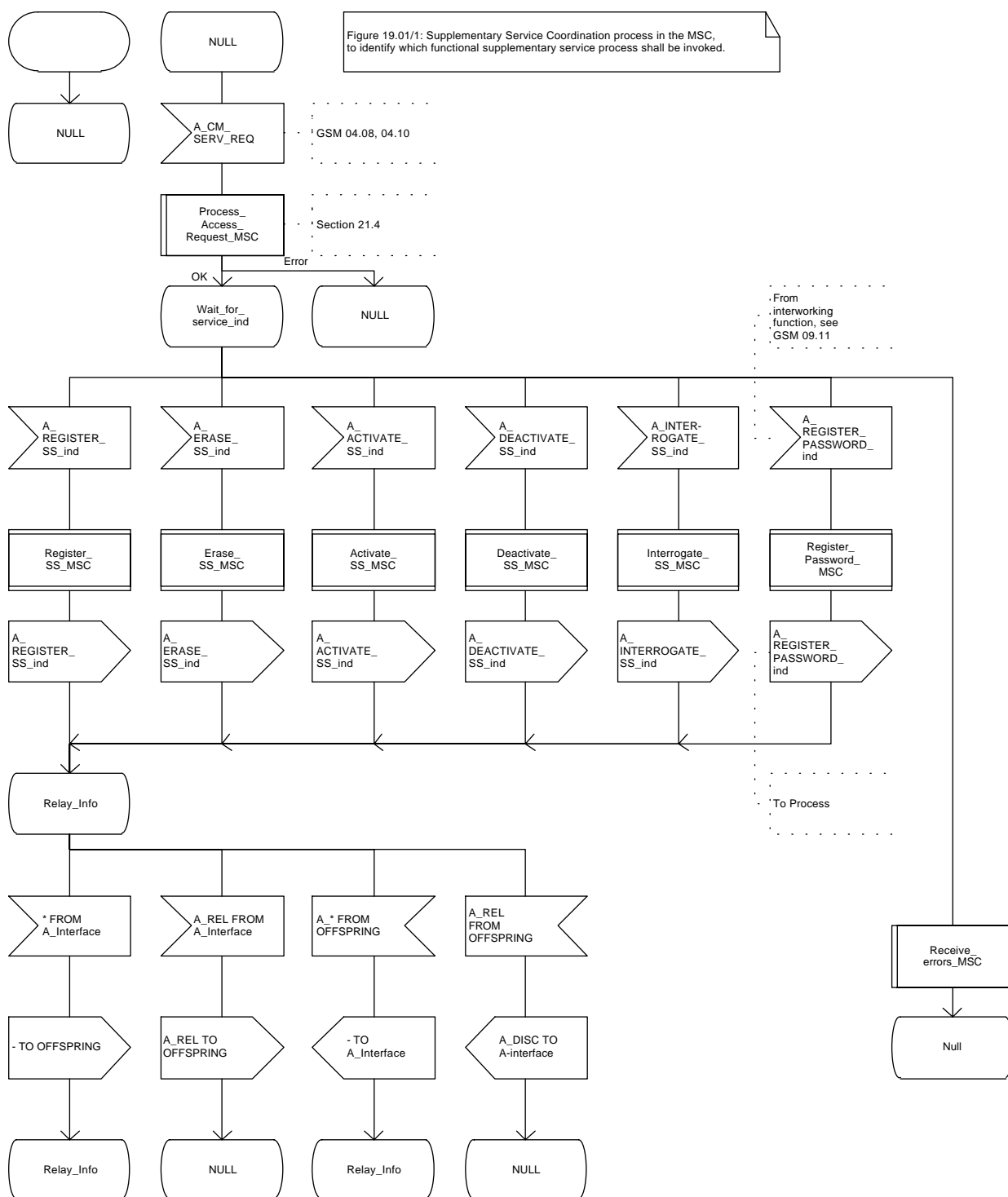
19.0.2   Functional supplementary service process co-ordinator for VLR

Any functional SS process in the VLR starts by the VLR receiving the MAP_PROCESS_ACCESS_REQUEST indication. The VLR then acts as described in clause 21 of this ETS.

If the Process Access Request was successful, the VLR can handle supplementary service indications from the MSC. Table 19.0/2 shows the co-ordinating process' reaction on receipt of specific SS service indications from the MSC. After the relevant process is invoked, the received service indication is sent to that process, and the co-ordinating process terminates.

Service indication received	Process invoked
MAP_REGISTER_SS_ind	REGISTER_SS_VLR
MAP_ERASE_SS_ind	ERASE_SS_VLR
MAP_ACTIVATE_SS_ind	ACTIVATE_SS_VLR
MAP_DEACTIVATE_SS_ind	DEACTIVATE_SS_VLR
MAP_INTERROGATE_SS_ind	INTERROGATE_SS_VLR
MAP_REGISTER_PASSWORD	REGISTER_PASSWORD_VLR

Table 19.0/2: Relationship between received service indication and invoked process in the VLR

Figure 19.0/2 shows the co-ordinating process in the VLR.

Figure 19.0/2 (sheet 1 of 2)

Figure 19.0/2: Supplementary Service Coordination process in the VLR, to open and process the access request from the MSC, and then identify which functional supplementary service process shall be invoked.
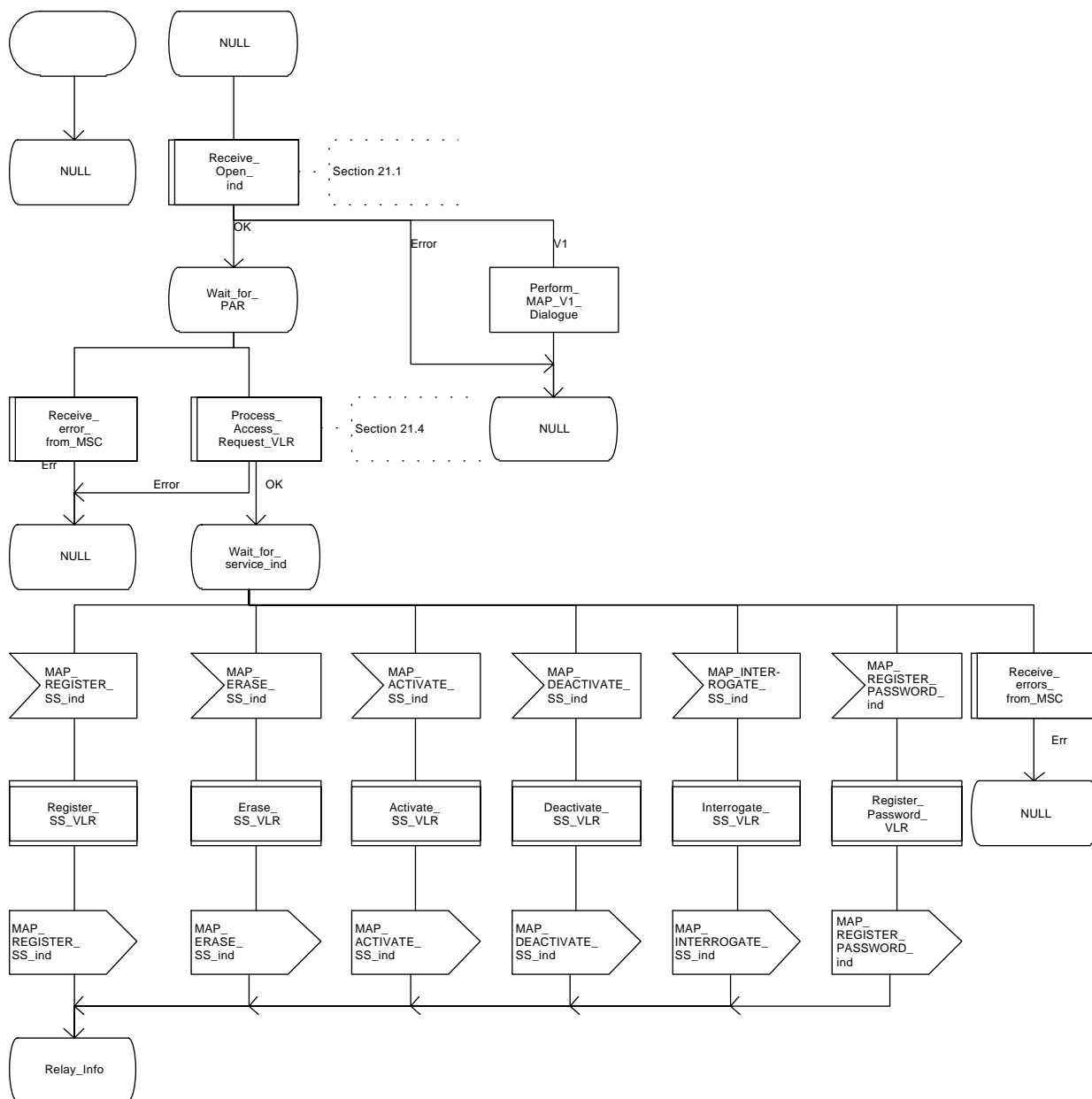
Figure 19.0/2 (sheet 2 of 2)
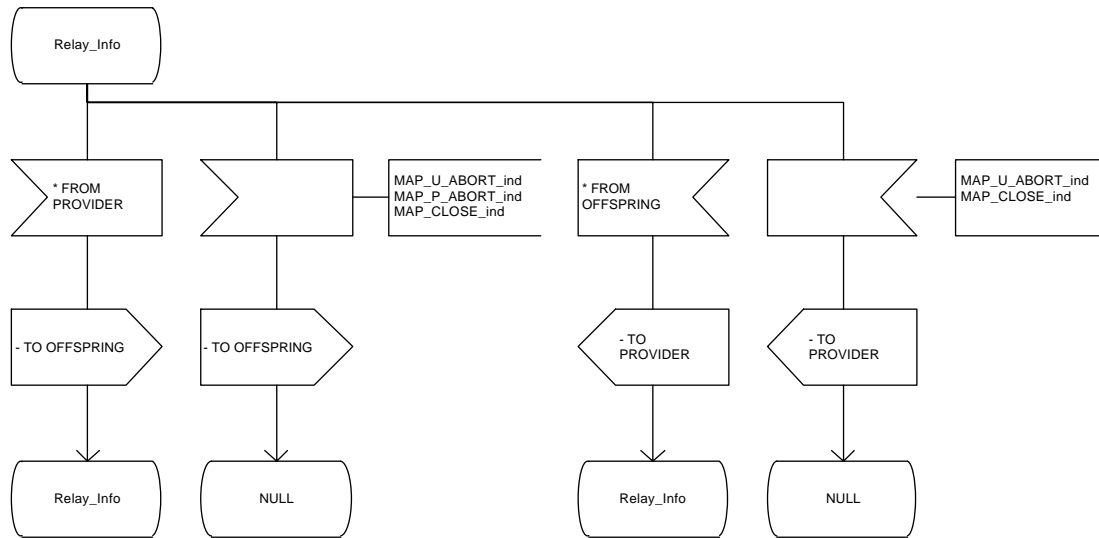
19.0.3   Functional supplementary service process co-ordinator for HLR

Any functional SS process in the HLR starts by the HLR receiving a MAP-OPEN service indication. If that service is successful, the HLR can handle supplementary service indications from the VLR. Table 19.0/3 shows the co-ordinating process' reaction on receipt of specific SS service indications from the VLR. After the relevant process is invoked, the received service indication is sent to that process, and the co-ordinating process terminates.

Service indication received	Process invoked
MAP_REGISTER_SS_ind	REGISTER_SS_HLR
MAP_ERASE_SS_ind	ERASE_SS_HLR
MAP_ACTIVATE_SS_ind	ACTIVATE_SS_HLR
MAP_DEACTIVATE_SS_ind	DEACTIVATE_SS_HLR
MAP_INTERROGATE_SS_ind	INTERROGATE_SS_HLR
MAP_REGISTER_PASSWORD	REGISTER_PASSWORD_HLR

Table 19.0/3: Relationship between received service indication and invoked process in the HLR.

Figure 19.0/3 shows the co-ordinating process in the HLR.

Figure 19.0/3 (sheet 1 of 2)
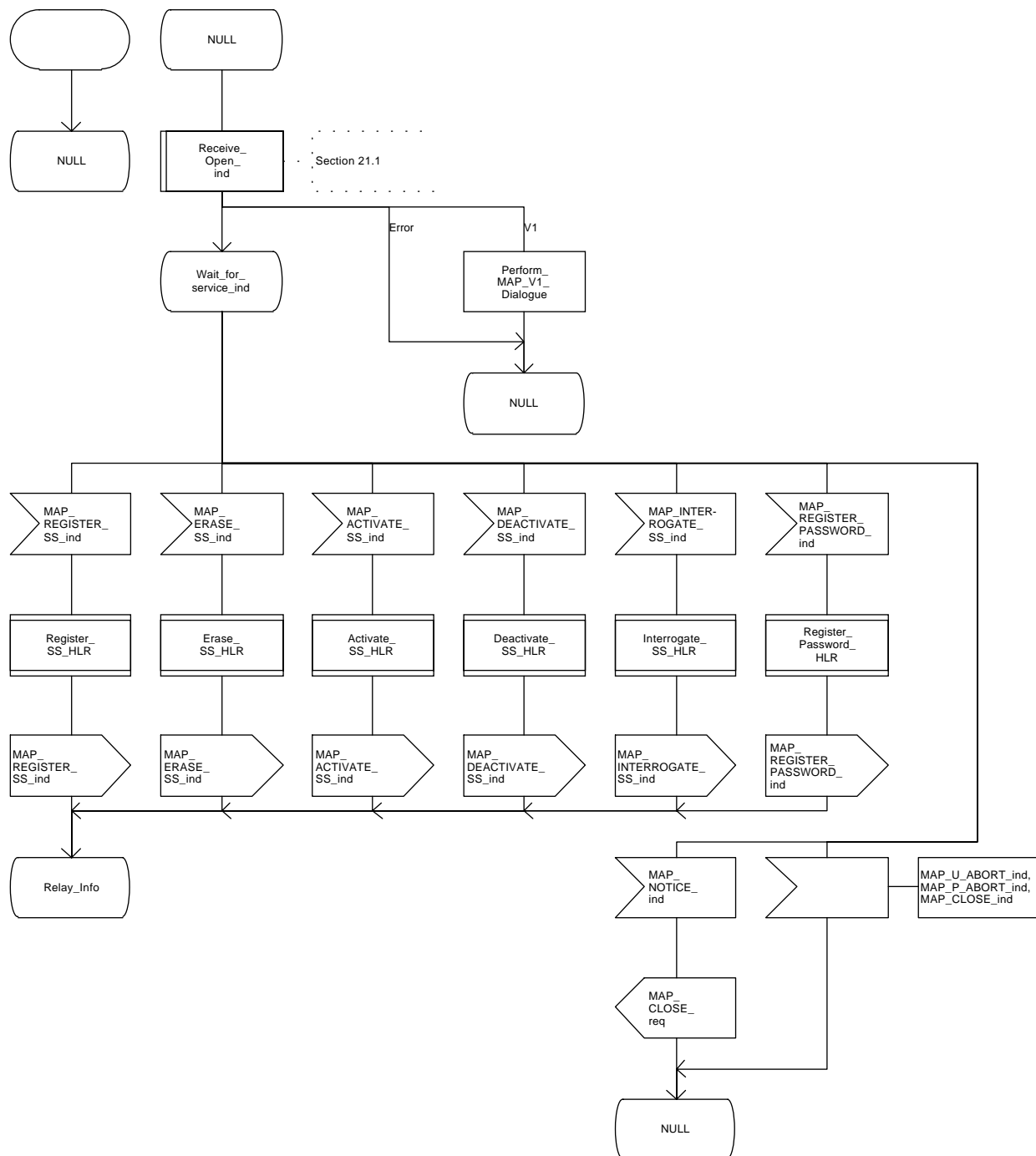
Figure 19.0/3 (sheet 2 of 2)

19.1   Registration procedure

19.1.1   General

The registration procedure is used to register data related to a supplementary service in the HLR. The registration procedure is a fully transparent communication between the MS and the HLR, except that some services may be invoked as a result of the procedure, as described in the subclauses below.

The registration procedure is shown in figure 19.1.1/1.

The following services may be used:

    MAP_PROCESS_ACCESS_REQUEST   (defined in clauses 6 and 21);
    MAP_TRACE_SUBSCRIBER_ACTIVITY   (defined in clauses 7 and 21);
    MAP_PROVIDE_IMSI   (defined in clauses 6 and 21);
    MAP_FORWARD_NEW_TMSI   (defined in clauses 6 and 21);
    MAP_AUTHENTICATE   (defined in clauses 6 and 21);
    MAP_SET_CIPHERING_MODE   (defined in clauses 6 and 21);
    MAP_CHECK_IMEI   (defined in clauses 6 and 21);
    MAP_READY_FOR_SM   (defined in clauses 10 and 21);
    MAP_INSERT_SUBSCRIBER_DATA   (defined in clauses 6 and 21);
    MAP_REGISTER_SS   (defined in clause 9).
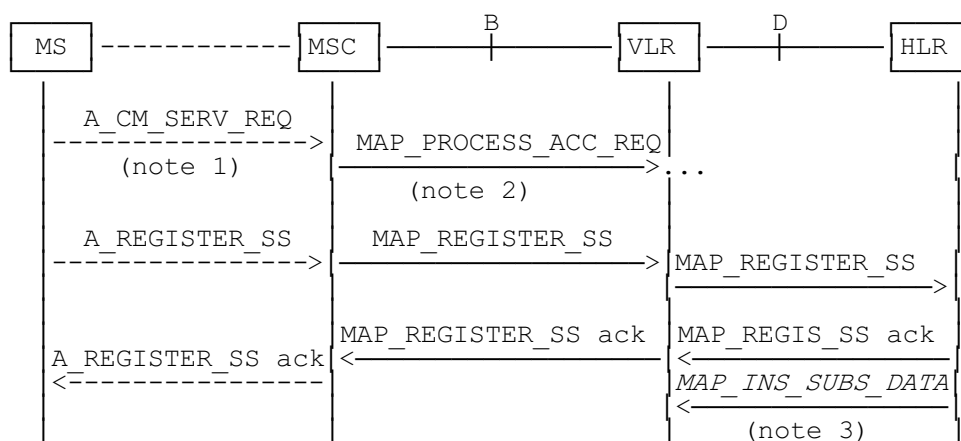
NOTE 1:   For details of the procedure on the radio path, see GSM 04.08, 04.10, 04.8x and 04.9x. Services shown in dotted lines indicate the trigger provided by the signalling on the radio path, and the signalling triggered on the radio path.

NOTE 2:   For details on the Process Access Request procedure, please refer to clause 21 in this ETS.

NOTE 3:   Services printed in *italics* are optional.

        Figure 19.1.1/1: Interfaces and services for supplementary service registration

19.1.2   Procedures in the MSC

Supplementary service registration

The A_REGISTER_SS service indication received by the MAP user in the MSC contains the SS-Code and any parameters that are related to the supplementary service.

The MAP user transfers the received information to the VLR in the MAP_REGISTER_SS request without checking the contents of the service indication. Rules for the mapping are described in GSM 09.11.

The MSC then awaits the receipt of the MAP_REGISTER_SS confirm from the VLR. The outcome of the procedure is reported to the MS in the A_REGISTER_SS response message as described in GSM 04.8x, 04.9x and 09.11. Finally the SS-connection is released.

For call independent SS operations, each message shall only contain a single component. Messages which contain more than one component will be stopped at the air interface (as specified in GSM 09.11).

Error handling

If at any time during the supplementary service part of this procedure a MAP_P_ABORT, MAP_U_ABORT, MAP_NOTICE or unexpected MAP_CLOSE indication is received from the VLR concerning the process, a CM_RELEASE_COMPLETE indication is sent to the MS (as specified in GSM 09.11). Upon receipt of a MAP_NOTICE indication from the VLR, the MSC must close the VLR dialogue by sending a MAP_CLOSE request. The process is then terminated.

If an A_CM_RELEASE indication is received from the MS, all open transactions shall be released using the MAP_U_ABORT request indicating application procedure cancellation, and the process is terminated.

The registration procedure in the MSC is shown in figure 19.1.2/1.

Figure 19.1.2/1
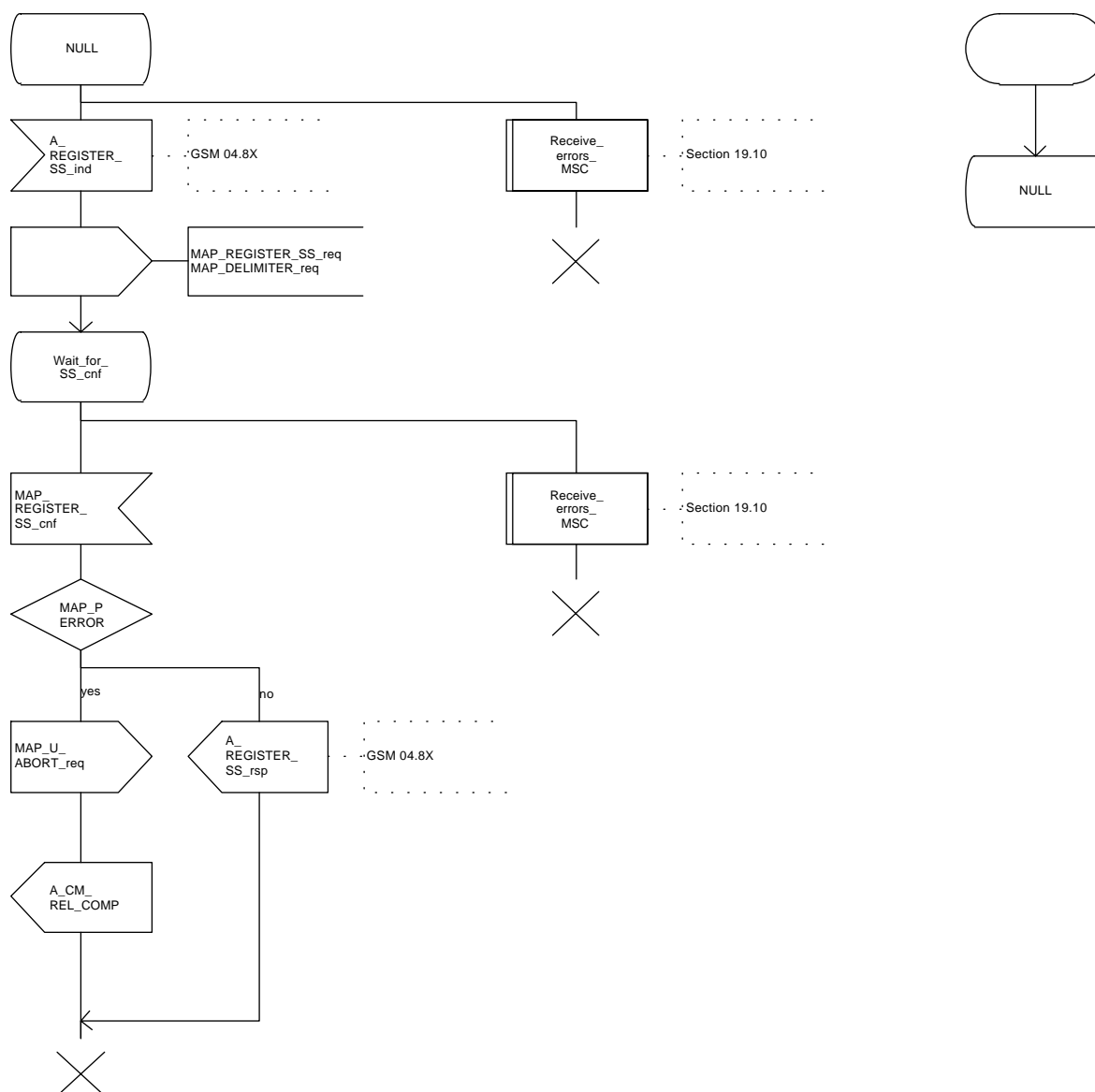
19.1.3 Procedures in the VLR

Supplementary service registration

When receiving the MAP_REGISTER_SS indication, the MAP user in the VLR transfers the information to the HLR in the MAP_REGISTER_SS request without checking the contents of the service indication.

The VLR then awaits the receipt of the MAP_REGISTER_SS confirm from the HLR. The MAP user in the VLR shall transfer the information contained in this primitive to the MSC in the MAP_REGISTER_SS response without checking its contents.

For call independent SS operations, each message shall only contain a single component. Messages which contain more than one component will be stopped at the air interface (as specified in GSM 09.11).

Error handling

If at any time during this procedure a MAP_P_ABORT, MAP_U_ABORT, MAP_NOTICE or unexpected MAP_CLOSE indication is received from the MSC concerning the process, a MAP_U_ABORT request indicating application procedure cancellation is sent to the HLR (if a connection exists). If a MAP_NOTICE indication was received from the MSC, that dialogue must be closed by sending a MAP_CLOSE request towards the MSC. The process is terminated.

If a MAP_P_ABORT, MAP_U_ABORT or MAP_CLOSE indication is received from the HLR, a MAP_U_ABORT request shall be sent to the MSC terminating the process. If a MAP_NOTICE indication was received from the HLR, that dialogue must be closed by sending a MAP_CLOSE request towards the HLR. The process terminates.

The registration procedure in the VLR is shown in figure 19.1.3/1.

Figure 19.1.3/1 (sheet 1 of 2)
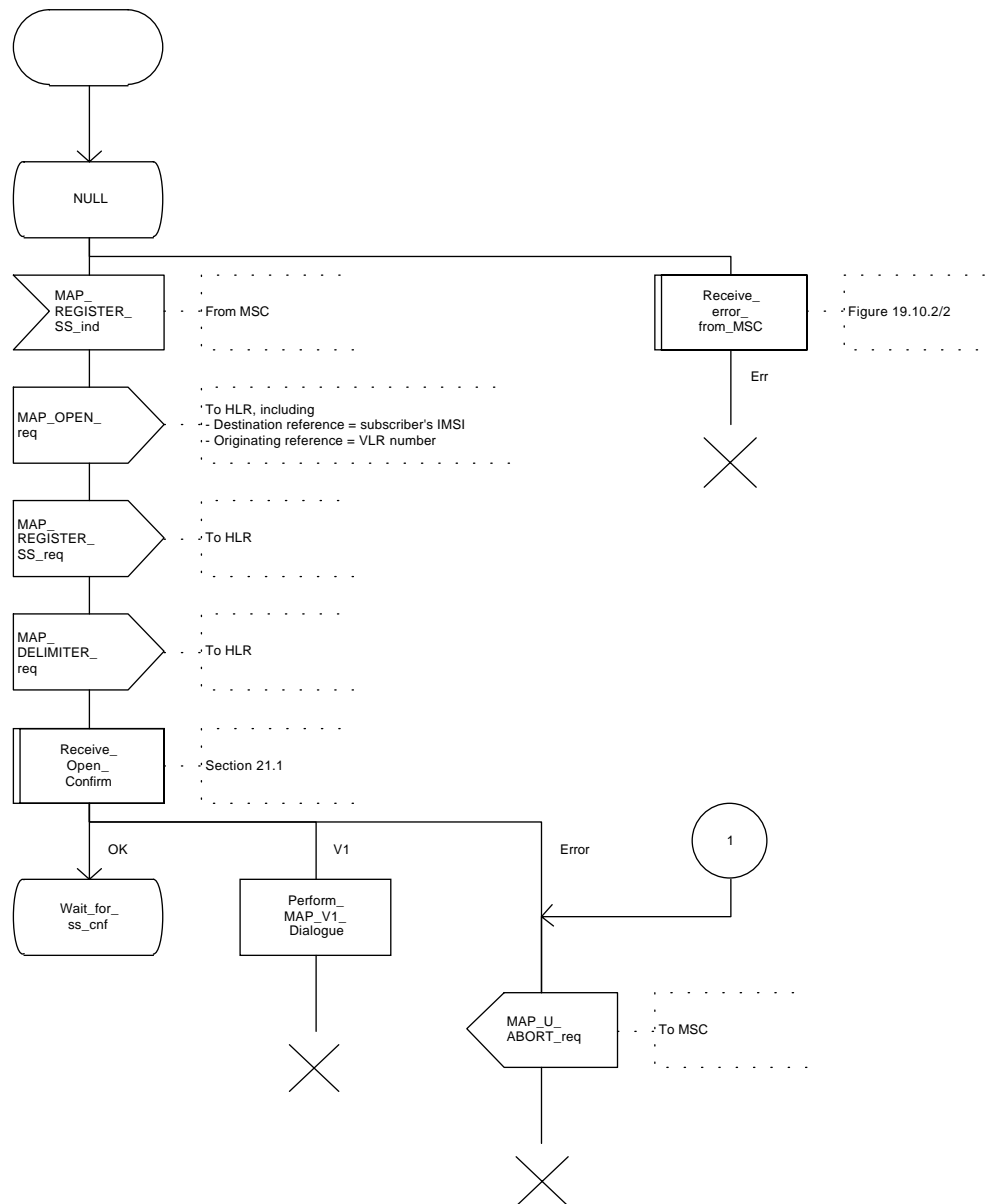
Figure 19.1.3/1 (sheet 2 of 2)
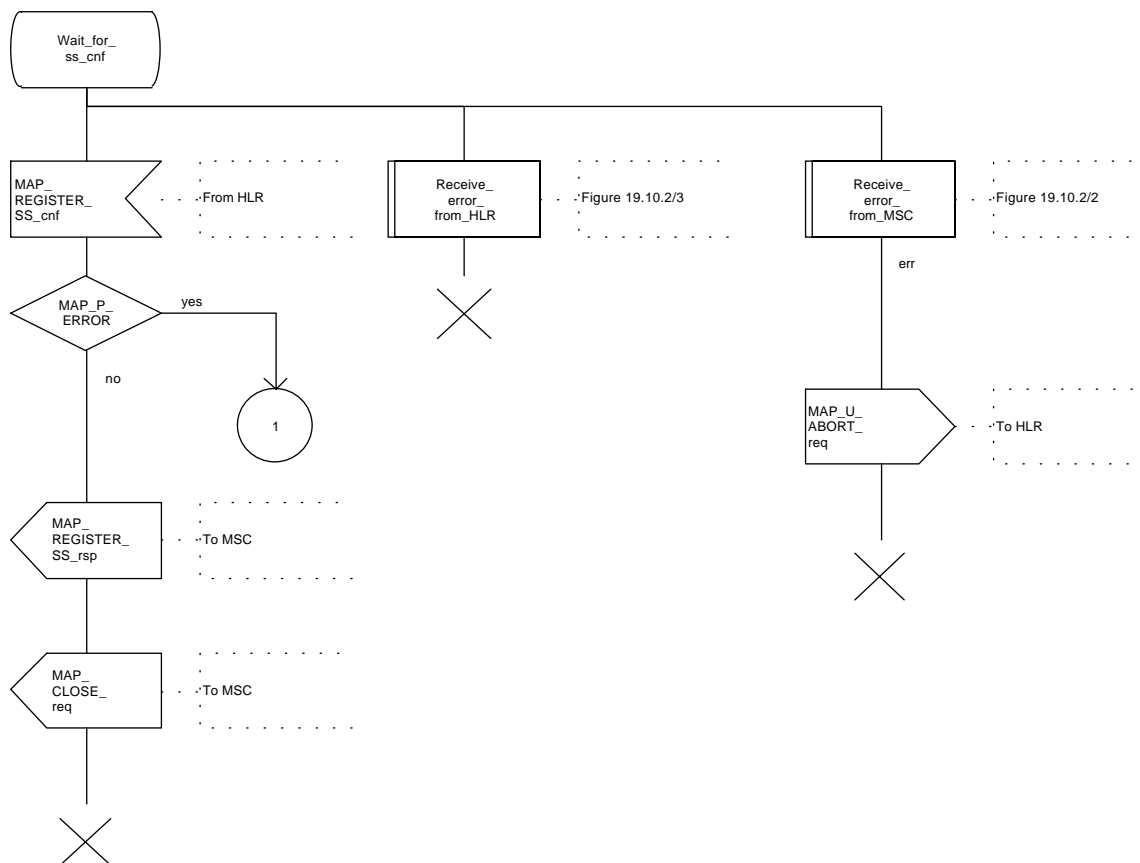
19.1.4 Procedures in the HLR

The procedure in the HLR is initiated when it receives a MAP_REGISTER_SS indication.

The HLR acts as follows:

- if the operator has barred the subscriber from access to supplementary services, the Call Barred error should be returned to the VLR. The parameter "operatorBarring" shall be included with the error.

The supplementary service request shall then be processed according to GSM 03.11 and the 03.8x and 03.9x-series of technical specifications. This handling may lead to either a successful result, a partially successful result, or an error being returned.

For call independent SS operations, each message shall only contain a single component. Messages which contain more than one component will be stopped at the air interface (as specified in GSM 09.11):

- if the VLR is to be updated after the supplementary service registration, the MAP_INSERT_SUBS_DATA_HLR process shall be initiated;

- if at any time during this procedure a MAP_P_ABORT, MAP_U_ABORT or MAP_CLOSE indication concerning the process is received from the VLR, the process is terminated. If a MAP_NOTICE indication is received, a MAP_CLOSE request indicating sent towards the VLR.

The registration procedure in the HLR is shown in figure 19.1.4/1.

Figure 19.1.4/1 (sheet 1 of 2)
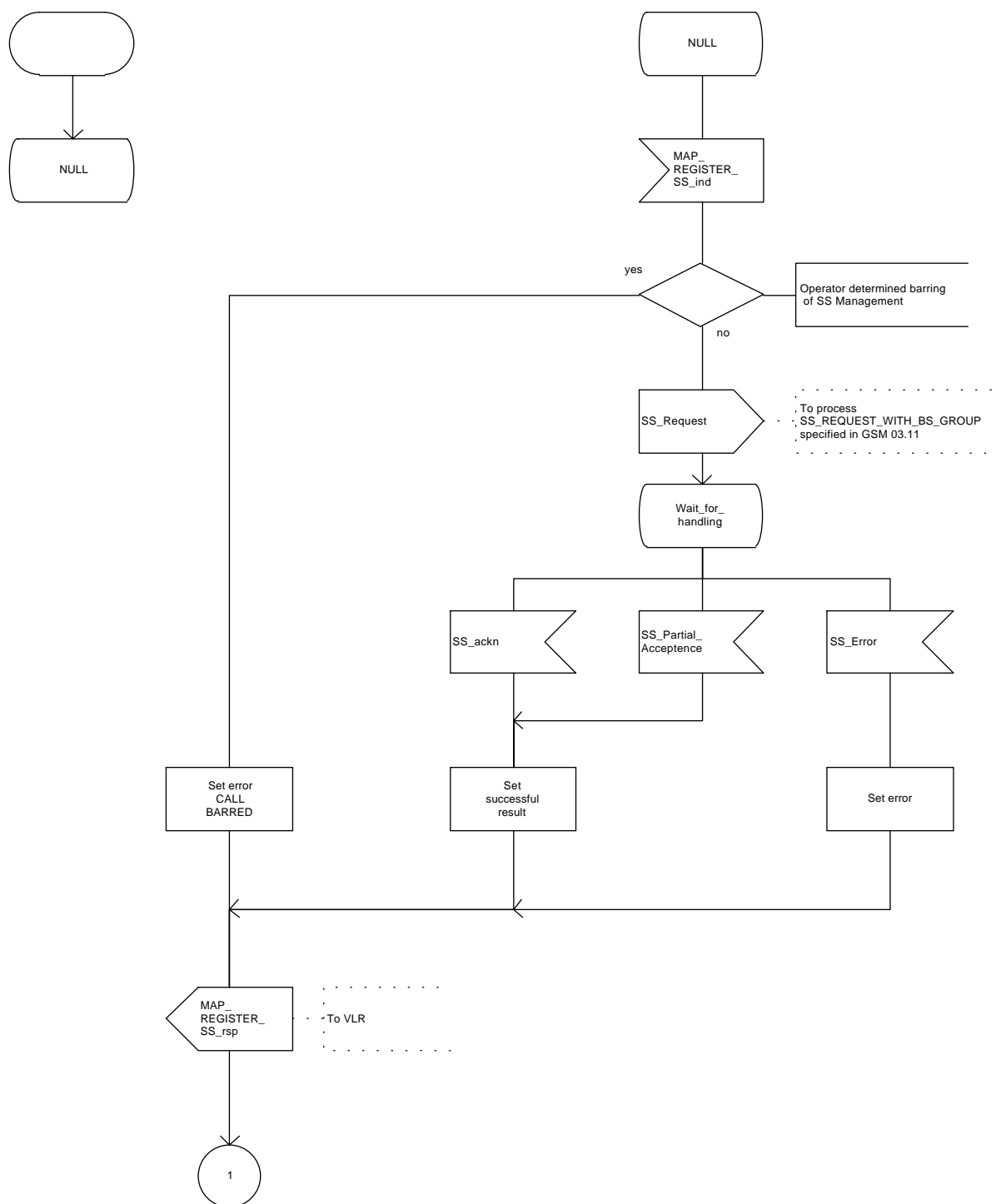
Figure 19.1.4/1 (sheet 2 of 2)
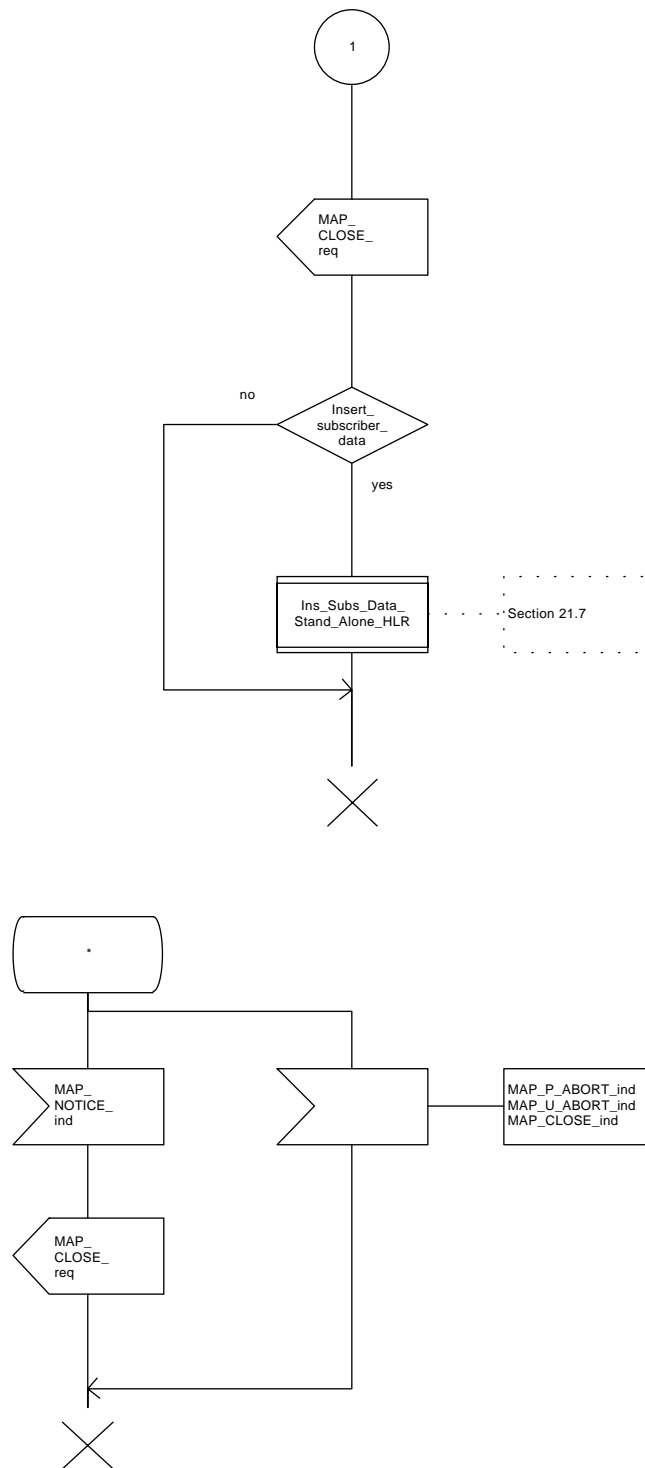
19.2 Erasure procedure

19.2.1 General

The erasure procedure is used to erase data related to a supplementary service in the HLR. The erasure procedure is a fully transparent communication between the MS and the HLR, except that some services may be invoked as a result of the procedure, as described in the subclauses below.

The erasure procedure is shown in figure 19.2.1/1.

The following services may be used:

MAP_PROCESS_ACCESS_REQUEST	(defined in subclauses 6 and 21);
MAP_TRACE_SUBSCRIBER_ACTIVITY	(defined in clauses 7 and 21);
MAP_PROVIDE_IMSI	(defined in clauses 6 and 21);
MAP_FORWARD_NEW_TMSI	(defined in clauses 6 and 21);
MAP_AUTHENTICATE	(defined in clauses 6 and 21);
MAP_SET_CIPHERING_MODE	(defined in clauses 6 and 21);
MAP_CHECK_IMEI	(defined in clauses 6 and 21);
MAP_READY_FOR_SM	(defined in clauses 10 and 21);
MAP_INSERT_SUBSCRIBER_DATA	(defined in clauses 6 and 21);
MAP_ERASE_SS	(defined in clause 9).
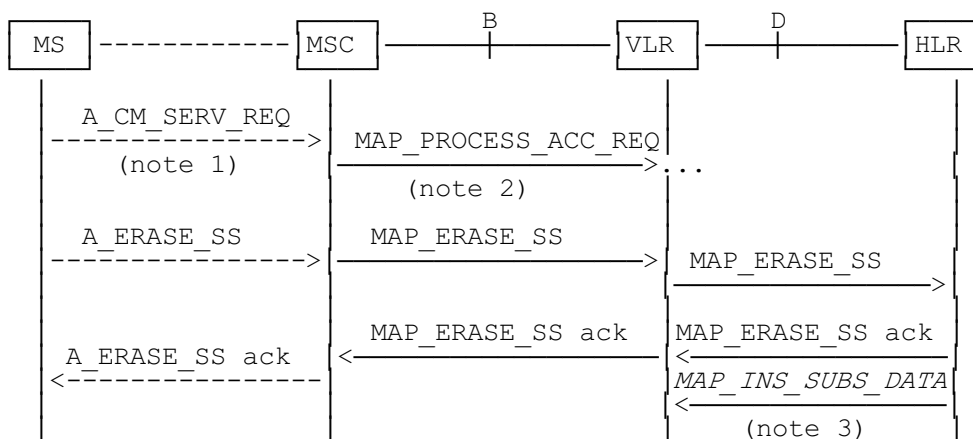
NOTE 1:   For details of the procedure on the radio path, see GSM 04.08, 04.10, 04.8x and 04.9x. Services shown in dotted lines indicate the trigger provided by the signalling on the radio path, and the signalling triggered on the radio path.

NOTE 2:   For details on the Process Access Request procedure, please refer to clause 21 in this ETS.

NOTE 3:   Services printed in *italics* are optional.

Figure 19.2.1/1: Interfaces and services for supplementary service erasure

19.2.2 Procedures in the MSC

The MSC procedures for erasure are identical to those specified for registration in subclause 19.1.2. The text and diagrams in subclause 19.1.2 apply with all references to registration changed to erasure.

19.2.3   Procedures in the VLR

The VLR procedures for erasure are identical to those specified for registration in subclause 19.1.3. The text and diagrams in subclause 19.1.3 apply with all references to registration changed to erasure.

19.2.4   Procedures in the HLR

The HLR procedure for erasure is identical to those specified for registration in subclause 19.1.4. The text and diagrams in subclause 19.1.4 apply with all references to registration changed to erasure.

19.3 Activation procedure

19.3.1 General

The activation procedure is used to activate a supplementary service in the HLR. The activation procedure is a fully transparent communication between the MS and the HLR, except that some services may be invoked as a result of the procedure, as described in the subclauses below.

The activation procedure is shown in figure 19.3.1/1.

The following services may be used:

MAP_PROCESS_ACCESS_REQUEST	(defined in clauses 6 and 21);
MAP_TRACE_SUBSCRIBER_ACTIVITY	(defined in clauses 7 and 21);
MAP_PROVIDE_IMSI	(defined in clauses 6 and 21);
MAP_FORWARD_NEW_TMSI	(defined in clauses 6 and 21);
MAP_AUTHENTICATE	(defined in clauses 6 and 21);
MAP_SET_CIPHERING_MODE	defined in clauses 6 and 21);
MAP_CHECK_IMEI	(defined in clauses 6 and 21);
MAP_READY_FOR_SM	(defined in clauses 10 and 21);
MAP_GET_PASSWORD	(defined in clause 9);
MAP_INSERT_SUBSCRIBER_DATA	(defined in clauses 6 and 21);
MAP_ACTIVATE_SS	defined in clause 9).
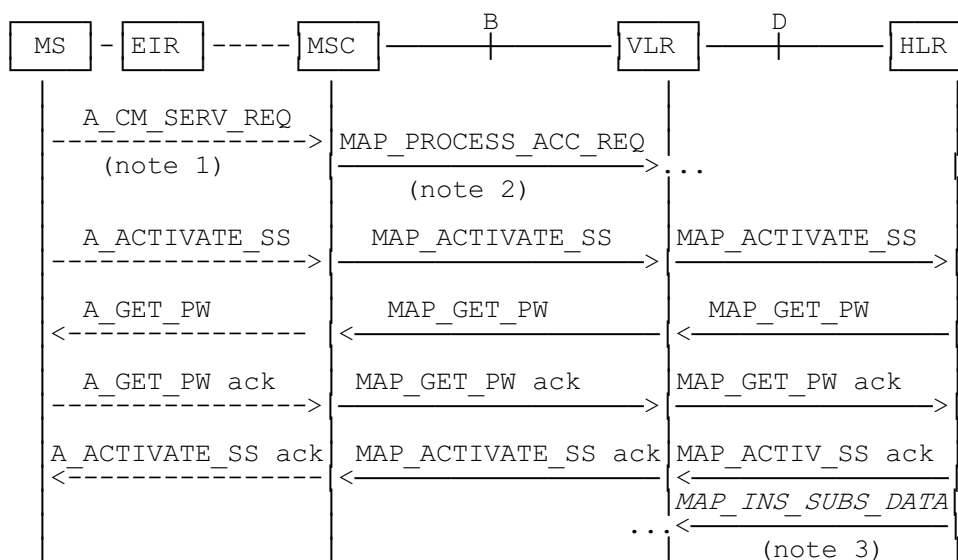
NOTE 1: For details of the procedure on the radio path, see GSM 04.08, 04.10, 04.8x and 04.9x. Services shown in dotted lines indicate the trigger provided by the signalling on the radio path, and the signalling triggered on the radio path.

NOTE 2: For details on the Process Access Request procedure, please refer to clause 21 in this ETS.

NOTE 3: Services printed in italics are optional.

Figure 19.3.1/1: Interfaces and services for supplementary service activation

19.3.2   Procedures in the MSC

The A_ACTIVATE_SS service indication received by the MAP user in the MSC contains the SS-Code and any parameters related to the supplementary service.

The MSC transfers the received information to the VLR in the MAP_ACTIVATE_SS request without checking the contents of the service indication. Rules for the mapping are described in GSM 09.11.

The MAP user may subsequently receive the MAP_GET_PASSWORD indication from the VLR. Upon receipt of this indication, the MSC sends the A_GET_PASSWORD message towards the MS and then awaits the response from the MS. When an A_GET_PASSWORD confirm message is received from the MS, the MSC initiates the MAP_GET_PASSWORD response towards the VLR without checking further the contents of the indication. Also see GSM 09.11.

The MSC will receive a MAP_ACTIVATE_SS confirm from the VLR. The outcome of the procedure is reported to the MS in the A_ACTIVATE_SS response message, see GSM 04.8x, 04.9x and 09.11. Finally the SS connection is released.

For call independent SS operations, each message shall only contain a single component. Messages which contain more than one component will be stopped at the air interface (as specified in GSM 09.11).

The handling of MAP_P_ABORT, MAP_U_ABORT, MAP_NOTICE and unexpected MAP_CLOSE or A_CM_RELEASE in this procedure is identical to the handling in the Registration procedure in the MSC, see subclause 19.1.2 of this ETS.

The activation procedure in the MSC is shown in figure 19.3.2/1.

Figure 19.3.2/1

19.3.3 Procedures in the VLR

Supplementary service activation

When receiving the MAP_ACTIVATE_SS indication, the MAP user in the VLR transfers the information to the HLR in the MAP_ACTIVATE_SS request without checking the contents of the service indication.

The VLR may then receive the MAP_GET_PASSWORD indication. This information is transferred to the MSC in the MAP_GET_PASSWORD request. If a MAP_GET_PASSWORD confirm primitive is received from the MSC, the VLR initiates the MAP_GET_PASSWORD response towards the HLR.

The VLR will receive the MAP_ACTIVATE_SS confirm from the HLR. The MAP user in the VLR shall transfer the information contained in this primitive to the MSC in the MAP_ACTIVATE_SS response without checking its contents.

For call independent SS operations, each message shall only contain a single component. Messages which contain more than one component will be stopped at the air interface (as specified in GSM 09.11).

Error handling

The handling of MAP_P_ABORT, MAP_U_ABORT, MAP_NOTICE and unexpected MAP_CLOSE in this procedure is identical to the handling in the Registration procedure in the VLR, see subclause 19.1.3 of this ETS.

The activation procedure in the VLR is shown in figure 19.3.3/1.

Figure 19.3.3/1 (sheet 1 of 2)
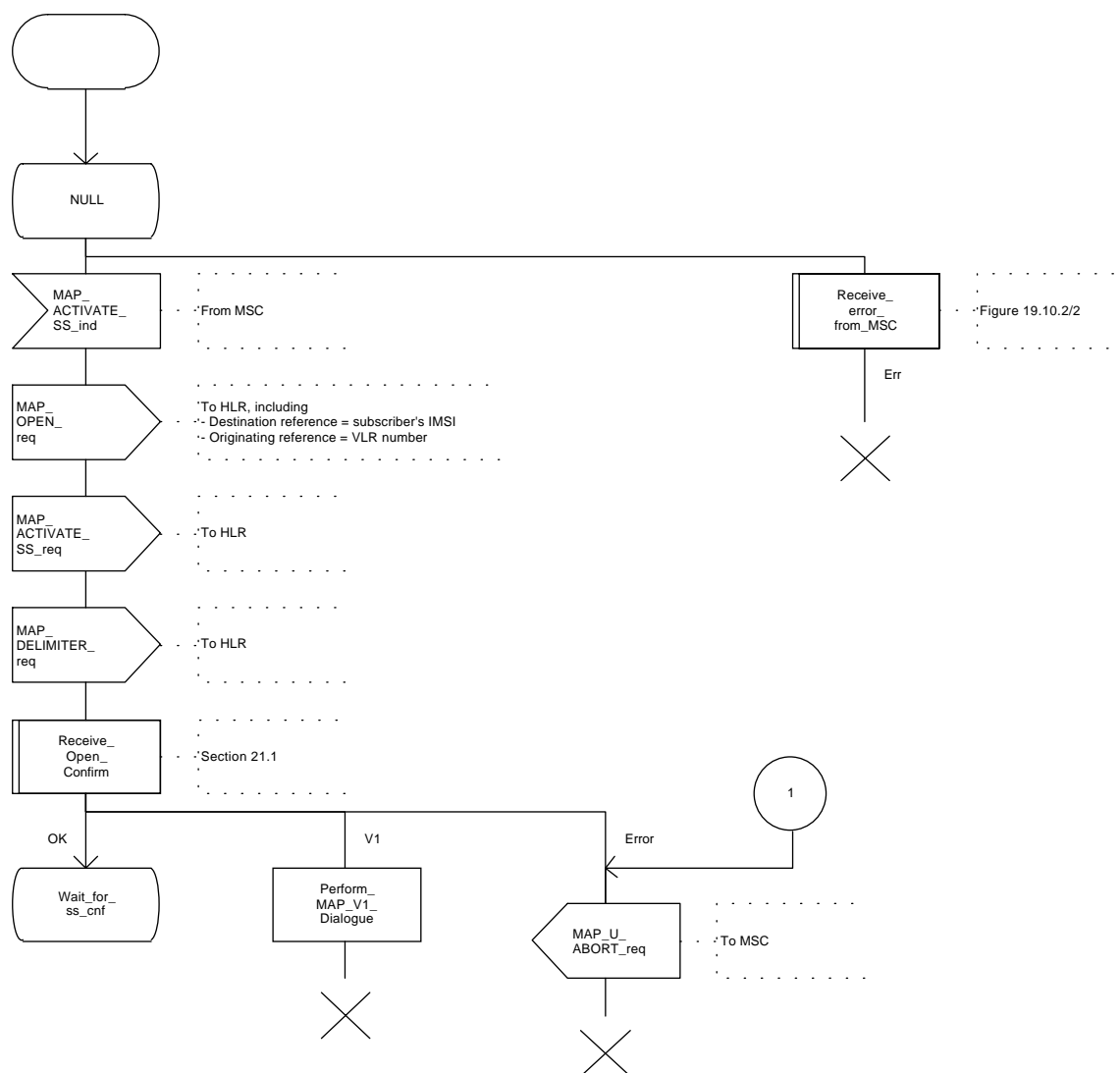
Figure 19.3.3/1 (sheet 2 of 2)
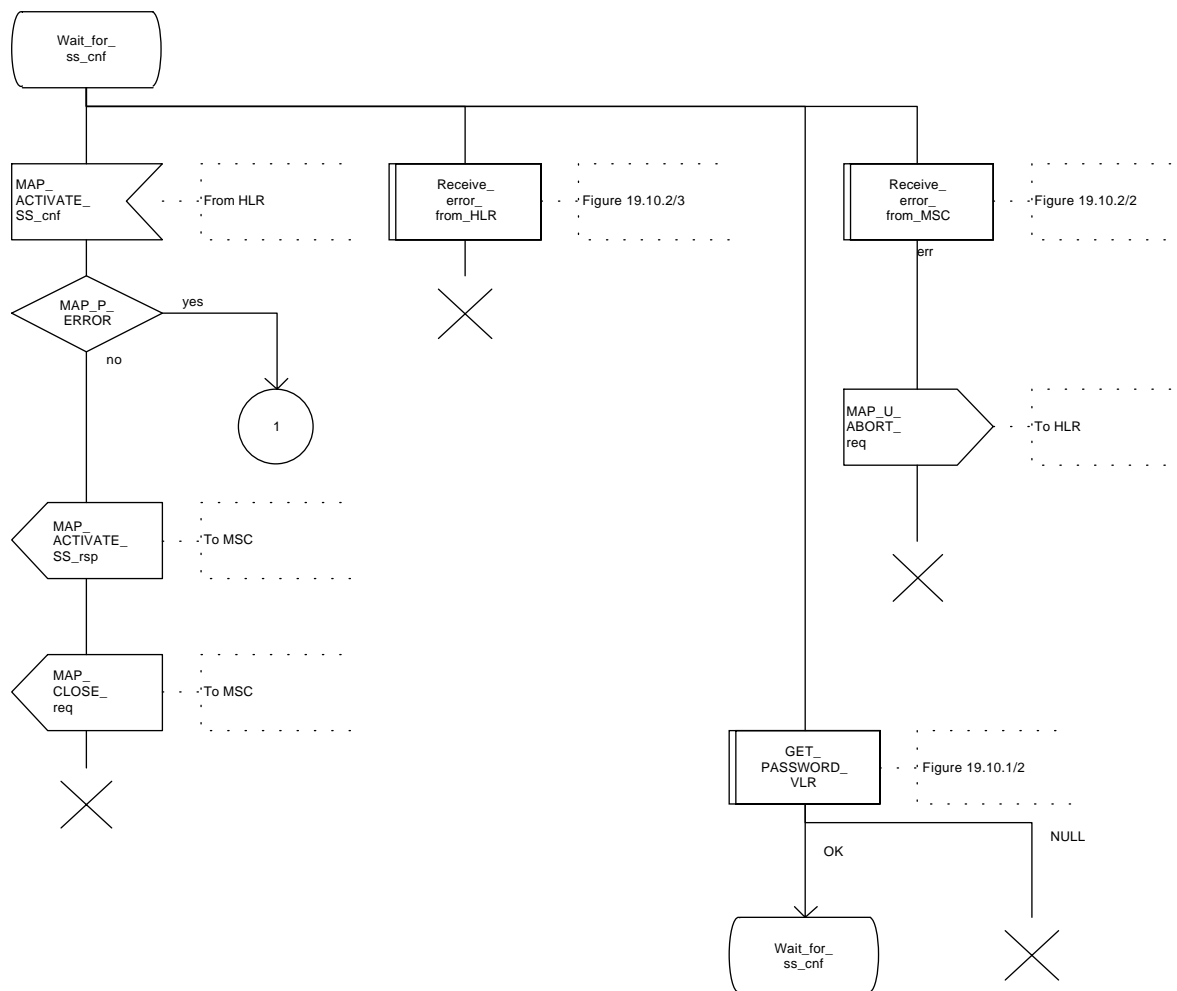
19.3.4 Procedures in the HLR

The procedure in the HLR is initiated when it receives a MAP_ACTIVATE_SS indication.

The HLR acts as follows:

- if the operator has barred the subscriber from access to supplementary services, the Call Barred error should be returned to the VLR. The parameter "operatorBarring" shall be included with the error.

The supplementary service request shall then be processed according to GSM 03.11 and the 03.8x and 03.9x-series of technical specifications. This handling may lead to either a successful result, a partially successful result, or an error being returned.

During the handling of activation, the get password procedure may be initiated (as specified in GSM 03.11). This will involve the sending of a MAP_GET_PASSWORD request to the VLR.

For call independent SS operations, each message shall only contain a single component. Messages which contain more than one component will be stopped at the air interface (as specified in GSM 09.11):

- if the VLR is to be updated after the supplementary service activation, the MAP_INSERT_SUBS_DATA_HLR process is initiated;

- handling of receipt of MAP_P_ABORT, MAP_U_ABORT or MAP_CLOSE indications from the VLR is identical to their handling in the registration procedure, see subclause 19.1.4 above.

The activation procedure in the HLR is shown in figure 19.3.4/1.

Figure 19.3.4/1 (sheet 1 of 2)
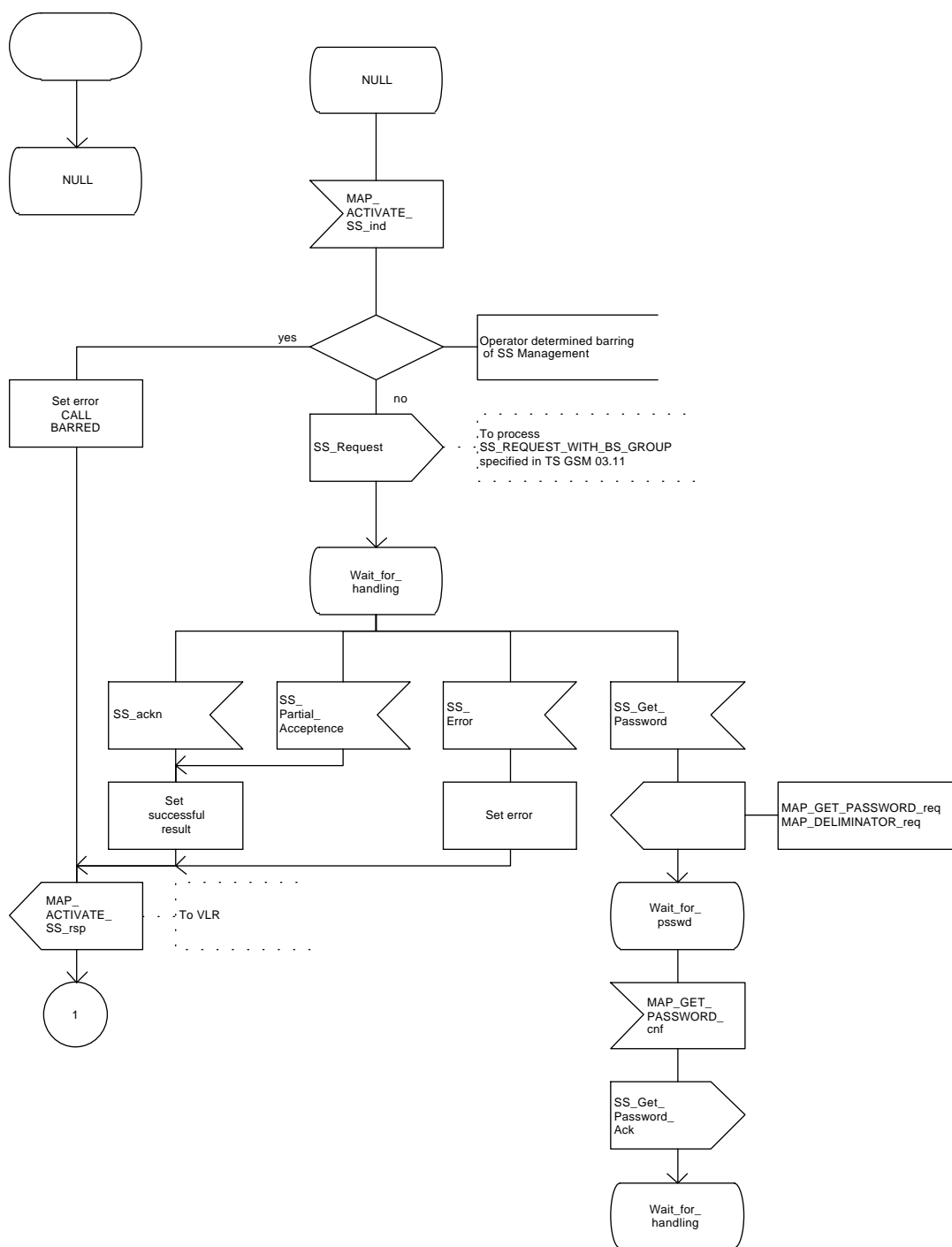
Figure 19.3.4/1 (sheet 2 of 2)
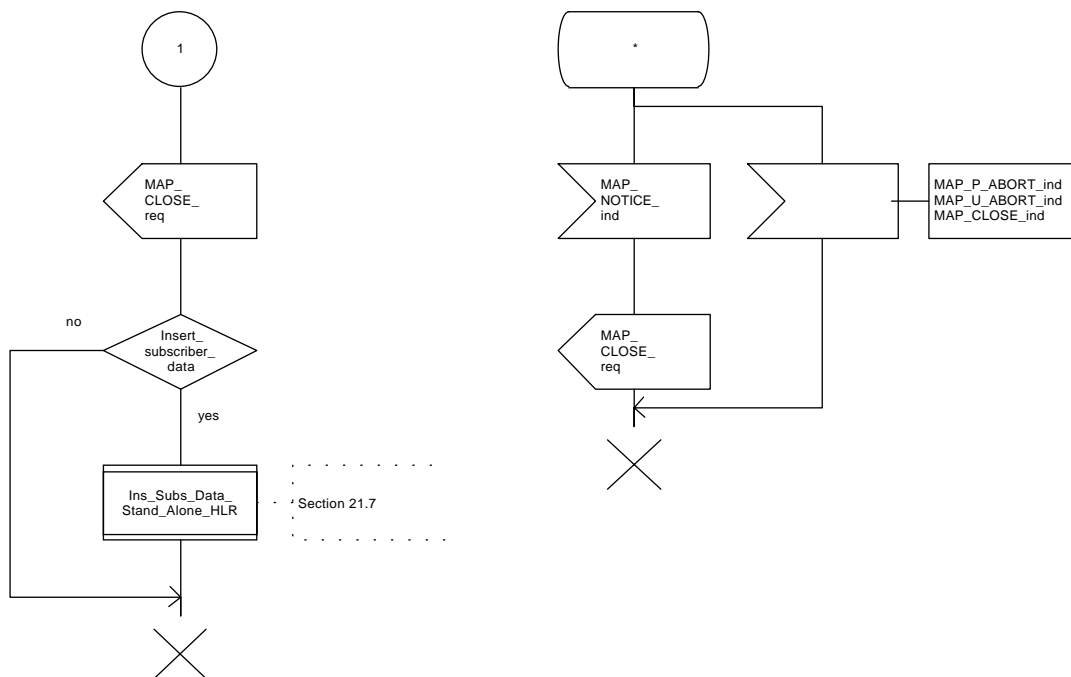
19.4 Deactivation procedure

19.4.1 General

The deactivation procedure is used to deactivate a supplementary service in the HLR. The deactivation procedure is a fully transparent communication between the MS and the HLR, except that some services may be invoked as a result of the procedure, as described in the subclauses below.

The deactivation procedure is shown in figure 19.4.1/1.

The following services may be used:

MAP_PROCESS_ACCESS_REQUEST	(defined in clauses 6 and 21);
MAP_TRACE_SUBSCRIBER_ACTIVITY	(defined in clauses 7 and 21);
MAP_PROVIDE_IMSI	(defined in clauses 6 and 21);
MAP_FORWARD_NEW_TMSI	defined in clauses 6 and 21);
MAP_AUTHENTICATE	(defined in clauses 6 and 21);
MAP_SET_CIPHERING_MODE	(defined in clauses 6 and 21);
MAP_CHECK_IMEI	(defined in clauses 6 and 21);
MAP_READY_FOR_SM	(defined in clauses 10 and 21);
MAP_GET_PASSWORD	(defined in clause 9);
MAP_INSERT_SUBSCRIBER_DATA	(defined in clauses 6 and 21);
MAP_DEACTIVATE_SS	(defined in clause 9).
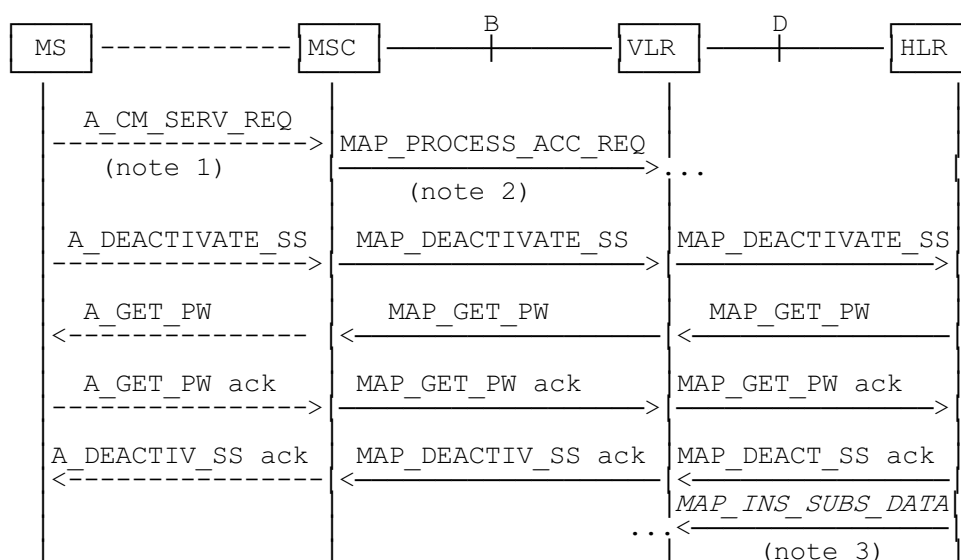
NOTE 1:   For details of the procedure on the radio path, see GSM 04.08, 04.10, 04.8x and 04.9x. Services shown in dotted lines indicate the trigger provided by the signalling on the radio path, and the signalling triggered on the radio path.

NOTE 2:   For details on the Process Access Request procedure, please refer to clause 21 in this ETS.

NOTE 3:   Services printed in *italics* are optional.

Figure 19.4.1/1: Interfaces and services for supplementary service deactivation

19.4.2   Procedures in the MSC

The MSC procedures for deactivation are identical to those specified for activation in subclause 19.3.2. The text and diagrams in subclause 19.3.2 apply with all references to activation changed to deactivation.

19.4.3   Procedures in the VLR

The VLR procedures for deactivation are identical to those specified for activation in subclause 19.3.3. The text and diagrams in subclause 19.3.3 apply with all references to activation changed to deactivation.

19.4.4   Procedures in the HLR

The HLR procedures for deactivation are identical to those specified for activation in subclause 19.3.4. The text and diagrams in subclause 19.3.4 apply with all references to activation changed to deactivation.

19.5 Interrogation procedure

19.5.1 General

The interrogation procedure is used to retrieve information related to a supplementary service from the VLR or the HLR. It is the VLR which decides whether an interrogation request should be forwarded to the HLR or not. Some non-supplementary service related services may be invoked as a result of the procedure, as described in the subclauses below.

The interrogation procedure is shown in figure 19.5.1/1.

The following services may be used:

MAP_PROCESS_ACCESS_REQUEST	(defined in clauses 6 and 21);
MAP_TRACE_SUBSCRIBER_ACTIVITY	(defined in clauses 7 and 21);
MAP_PROVIDE_IMSI	(defined in clauses 6 and 21);
MAP_FORWARD_NEW_TMSI	(defined in clauses 6 and 21);
MAP_AUTHENTICATE	(defined in clauses 6 and 21);
MAP_SET_CIPHERING_MODE	(defined in clauses 6 and 21);
MAP_CHECK_IMEI	(defined in clauses 6 and 21);
MAP_READY_FOR_SM	(defined in clauses 10 and 21);
MAP_INTERROGATE_SS	(defined in clause 9).
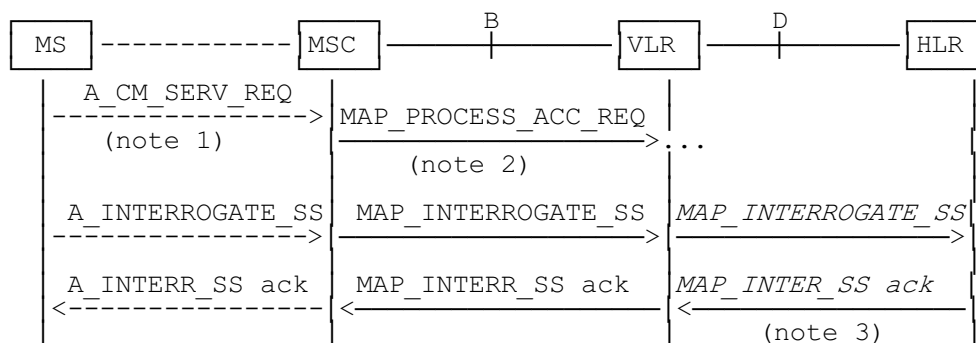
```
 ┌────┐              ┌────┐    B     ┌────┐    D     ┌────┐
 │ MS │ ------------ │MSC │────────┼─│VLR │────────┼─│HLR │
 └────┘              └────┘          └────┘          └────┘
   │  A_CM_SERV_REQ    │                │                │
   │ ----------------> │MAP_PROCESS_ACC_REQ             │
   │    (note 1)       │───────────────>│...            │
   │                   │   (note 2)     │               │
   │ A_INTERROGATE_SS  │MAP_INTERROGATE_SS  MAP_INTERROGATE_SS
   │ ----------------> │───────────────>│──────────────>│
   │ A_INTERR_SS ack   │MAP_INTERR_SS ack   MAP_INTER_SS ack
   │ <---------------- │<───────────────│<──────────────│
   │                   │                │   (note 3)    │
```

NOTE 1:   For details of the procedure on the radio path, see GSM 04.08, 04.10, 04.8x and 04.9x. Services shown in dotted lines indicate the trigger provided by the signalling on the radio path, and the signalling triggered on the radio path.

NOTE 2:   For details on the Process Access Request procedure, please refer to clause 21 in this ETS.

NOTE 3:   Services printed in italics are optional.

Figure 19.5.1/1: Interfaces and services for supplementary service interrogation

19.5.2 Procedures in the MSC

The MSC procedures for interrogation are identical to those specified for registration in subclause 19.1.2. The text and diagrams in subclause 19.1.2 apply with all references to registration changed to interrogation.

19.5.3 Procedures in the VLR

Supplementary service interrogation

When receiving the MAP_INTERROGATE_SS indication, the MAP user acts as follows:

- if the operator has barred the subscriber from access to supplementary services, the error Call Barred is returned to the MSC. The parameter "operatorBarring" shall be included with the error.

The interrogation is either answered by the VLR or by the HLR, depending on the service interrogated.

a) Interrogation to be handled by the VLR

The supplementary service request shall then be processed according to GSM 03.11 and the 03.8x and 03.9x-series of technical specifications. This handling may lead to either a successful result, a partially successful result, or an error being returned.

For call independent SS operations, each message shall only contain a single component. Messages which contain more than one component will be stopped at the air interface (as specified in GSM 09.11).

b) Interrogation to be handled by HLR

If the interrogation is to be handled by the HLR, on receiving the MAP_INTERROGATE_SS indication, the MAP user in the VLR transfers the information to the HLR in the MAP_INTERROGATE_SS request without further checking the contents of the service indication.

The VLR will receive the MAP_INTERROGATE_SS confirm from the HLR. The MAP user in the VLR shall transfer the information contained in this primitive to the MSC in the MAP_INTERROGATE_SS response without checking its contents.

For call independent SS operations, each message shall only contain a single component. Messages which contain more than one component will be stopped at the air interface (as specified in GSM 09.11).

Error handling

Handling of MAP_P_ABORT, MAP_U_ABORT, MAP_NOTICE and unexpected MAP_CLOSE in this procedure is identical to the handling in the Registration procedure in the VLR, subclause 19.1.3. The Interrogation procedure is described in figure 19.5.3/1.

Figure 19.5.3/1 (sheet 1 of 3)
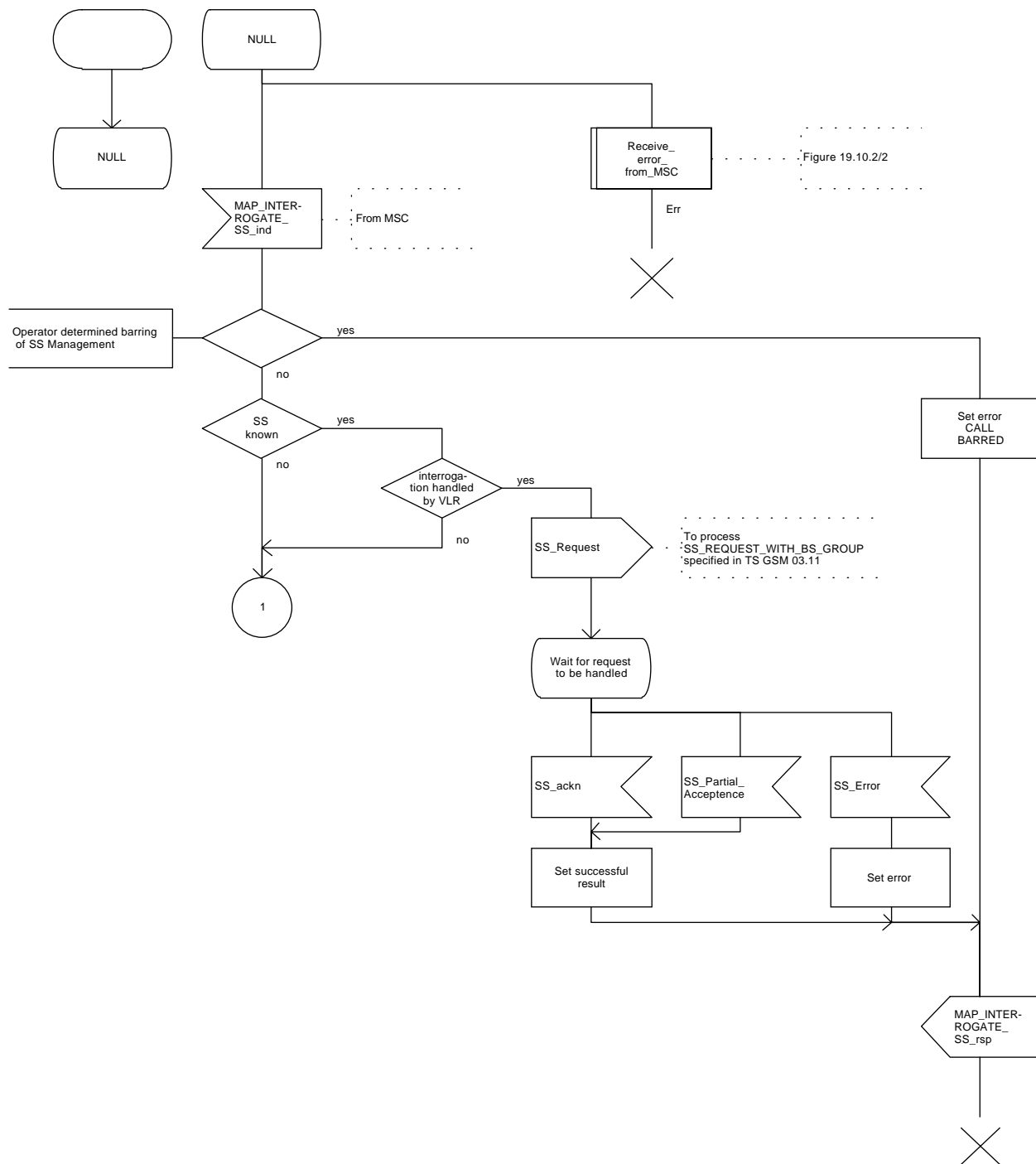
Figure 19.5.3/1 (sheet 2 of 3)

①

MAP_OPEN_req	To HLR, including
	- Destination reference = subscriber's IMSI
	- Originating reference = VLR number

| MAP_INTERROGATE_SS_req | To HLR |

| MAP_DELIMITER_req | To HLR |

| Receive_Open_Confirm | Section 21.1 |

②

OK · · · V1 · · · Error

Wait_for_ss_cnf

Perform_MAP_V1_Dialogue

MAP_U_ABORT_req · To MSC
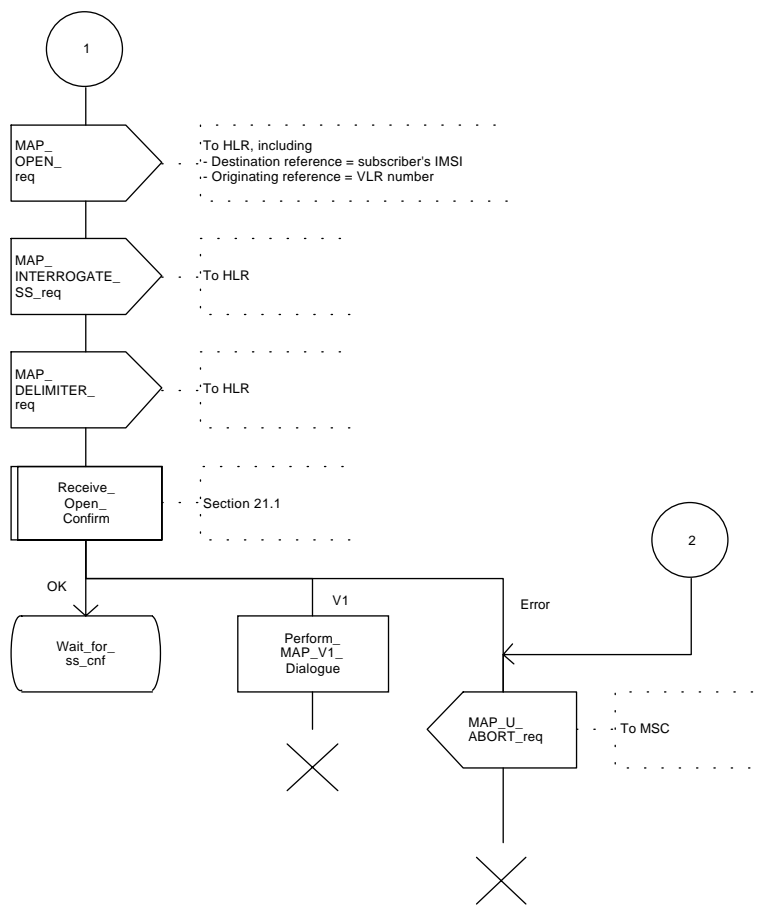
Figure 19.5.3/1 (sheet 3 of 3)
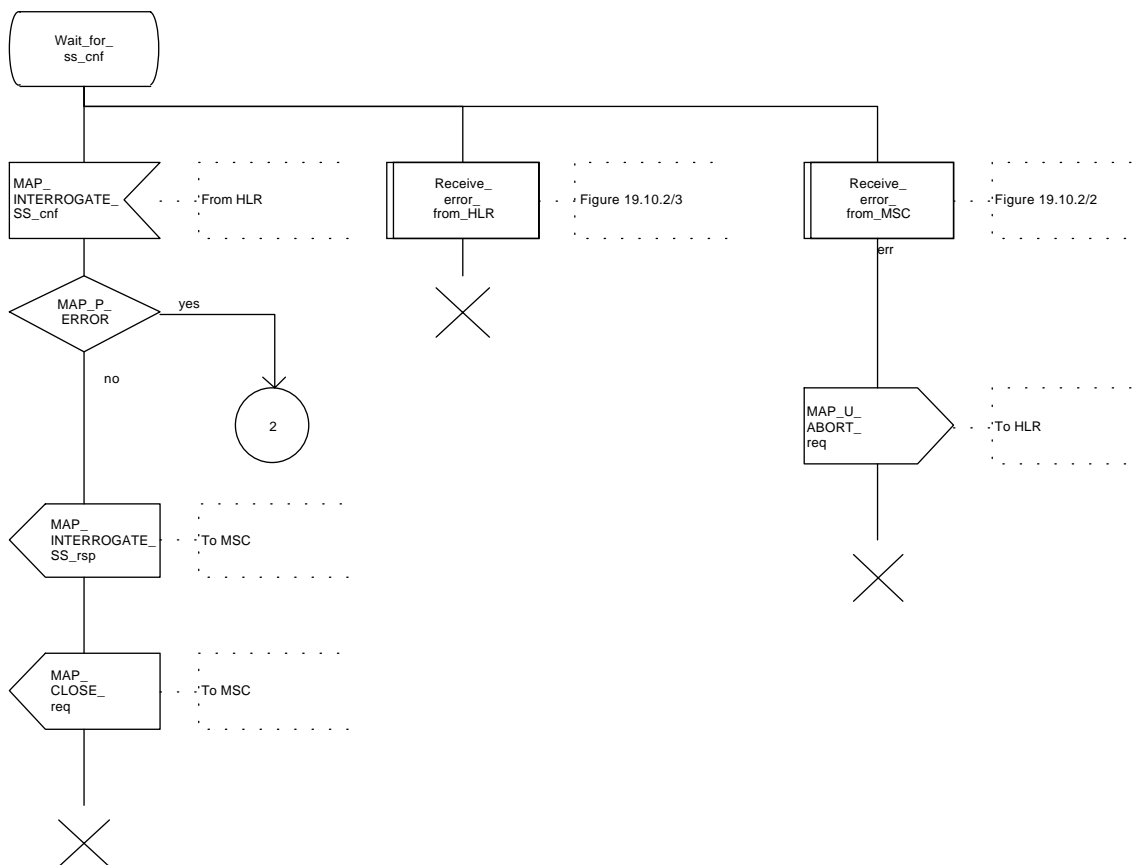
19.5.4   Procedures in the HLR

When receiving the MAP_INTERROGATE_SS indication, the MAP user acts as follows:

- if the operator has barred the subscriber from access to supplementary services, the error Call Barred is returned to the MSC. The parameter "operatorBarring" shall be included with the error;

- if the supplementary service is not supported in HLR the error Unexpected Data Value is returned to the VLR.

The interrogation is either answered by the VLR or by the HLR, depending on the service interrogated.

a)   Interrogation to be handled by the VLR

If the interrogation procedure should have been answered by the VLR, then the HLR assumes that the VLR does not support the interrogated supplementary service, and returns the SS Not Available error to the VLR.

b)   Interrogation to be handled by HLR

The supplementary service request shall be processed according to GSM 03.11 and the 03.8x and 03.9x-series of technical specifications. This handling may lead to either a successful result or an error being returned.

For call independent SS operations, each message shall only contain a single component.

Error handling

Handling of MAP_P_ABORT, MAP_U_ABORT, MAP_NOTICE and unexpected MAP_CLOSE in this procedure is identical to the handling in the Registration procedure in the VLR, subclause 19.1.3. The Interrogation procedure is described in figure 19.5.4/1.

Figure 19.5.4/1
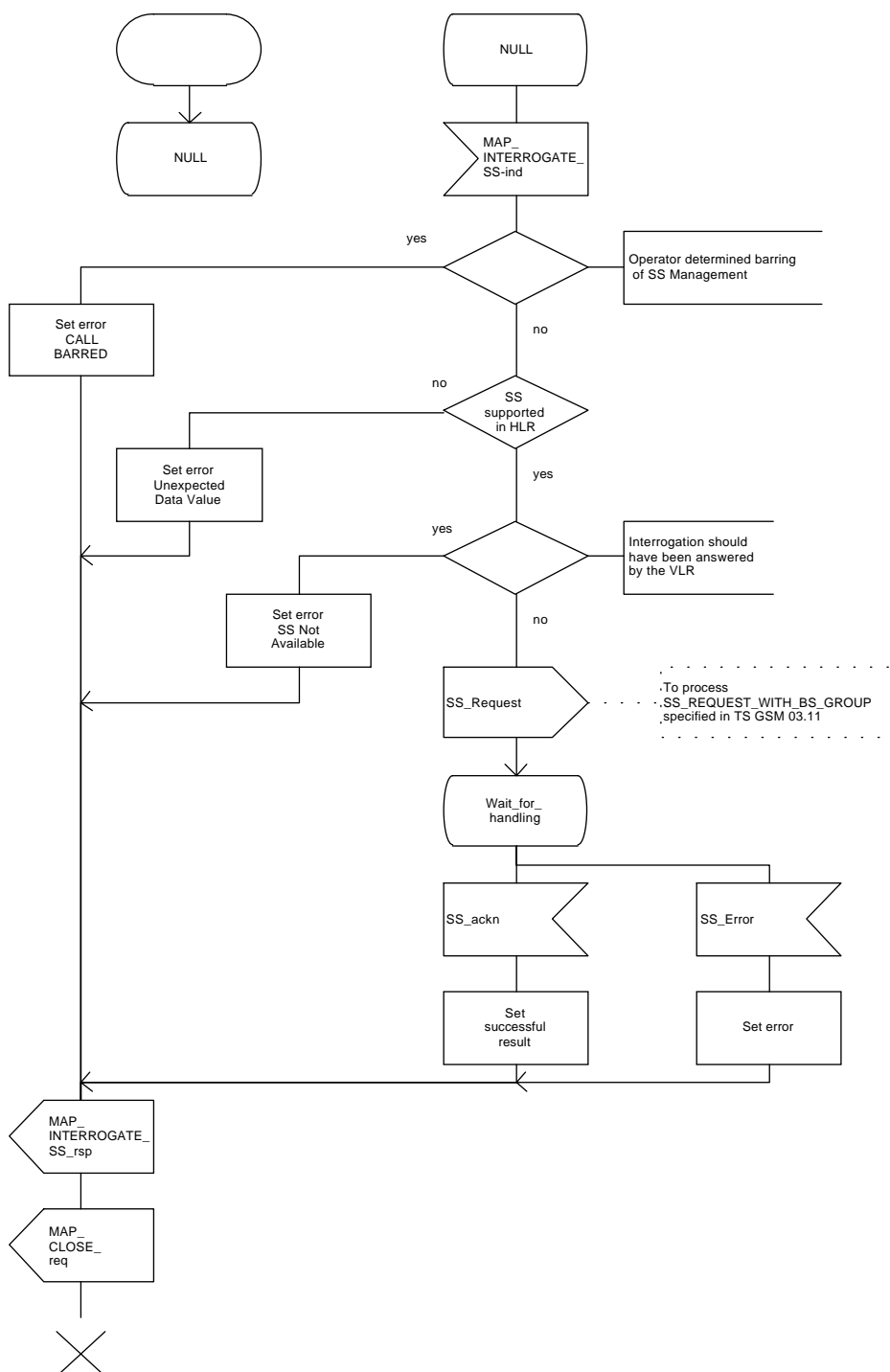
19.6   Invocation procedure

19.6.1   General

The invocation procedure is used to check subscription data in the VLR for certain supplementary services which are invoked after the call set-up phase is finished. For invocation of supplementary services which are invoked during the call set-up phase, please refer to the Call Handling procedure descriptions.

The invocation procedure is shown in figure 19.6.1/1. Note that some optional services may be invoked in connection with this procedure, as described in the subclause below.

The following services are used:

```
MAP_PROCESS_ACCESS_REQUEST      (defined in clauses 6 and 21);
MAP_TRACE_SUBSCRIBER_ACTIVITY   (defined in clauses 7 and 21);
MAP_PROVIDE_IMSI                (defined in clauses 6 and 21);
MAP_FORWARD_NEW_TMSI            (defined in clauses 6 and 21);
MAP_AUTHENTICATE               (defined in clauses 6 and 21);
MAP_SET_CIPHERING_MODE        (defined in clauses 6 and 21);
MAP_CHECK_IMEI                 (defined in clauses 6 and 21);
MAP_READY_FOR_SM               (defined in clauses 10 and 21);
MAP_INVOKE_SS                  (defined in clause 9).
```
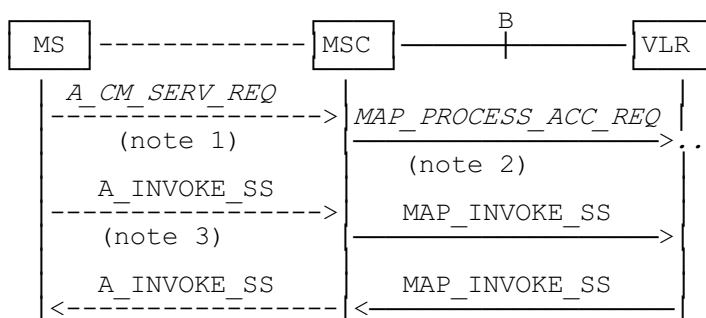
```
                                           B
          ┌─────┐             ┌─────┐       │      ┌─────┐
          │ MS  │-------------│ MSC │───────┼──────│ VLR │
          └─────┘             └─────┘       │      └─────┘
            │  A_CM_SERV_REQ     │                   │
            │------------------->│ MAP_PROCESS_ACC_REQ│
            │    (note 1)        │──────────────────> ...
            │                    │    (note 2)        │
            │   A_INVOKE_SS      │                    │
            │------------------->│  MAP_INVOKE_SS     │
            │    (note 3)        │──────────────────> │
            │                    │                    │
            │   A_INVOKE_SS      │  MAP_INVOKE_SS     │
            │<-------------------│<───────────────────│
            │                    │                    │
```

NOTE 1:   For details of the procedure on the radio path, see GSM 04.08, 04.10, 04.8x and 04.9x. Services shown in dotted lines indicate the trigger provided by the signalling on the radio path, and the signalling triggered on the radio path.

NOTE 2:   For details on the Process Access Request procedure, please refer to clause 21 in this ETS.

NOTE 3:   A_INVOKESS is a generic message to illustrate any supplementary service invocation request message on the air interface, e.g. BuildMPTY, see GSM 04.80.

Figure 19.6.1/1: Interfaces and services for supplementary service invocation

19.6.2 Procedures in the MSC

Process access request

Before the Call Hold or Multi-Party supplementary services can be invoked, a CC connection must be established between the MS and the MSC as described in GSM 04.08 and the Call Handling procedure descriptions within this ETS.

When an A_INVOKE_SS request message arrives at the MSC during a call (as described in GSM 04.10, 04.8x and 04.9x-series of technical specifications), then if control of subscription to the invoked supplementary service is required, the MSC initiates the process access request procedure towards the VLR as described in clause 21 of this ETS.

Supplementary service invocation

If the Process Access Request procedure towards the VLR is successful, the MSC shall forward a MAP_INVOKE_SS service request towards the VLR. This request shall contain the SS-Code of the supplementary service to be invoked, and possibly the Basic service code. Mapping from the A_INVOKE_SS to this service request is described in GSM 09.11.

The MSC will receive a MAP_INVOKE_SS confirm from the VLR. If the outcome of the service is successful (i.e. the service confirm is empty), the MSC will invoke the requested supplementary service as described in GSM 02.8x-series, 03.8x and 03.9x-series of technical specifications. If the outcome of the service is unsuccessful, the MSC shall send an appropriate A_INVOKE_SS response towards the MS. The structure of this message is described in GSM 09.11 and 04.8x and 04.9x-series of technical specifications.

Error handling

If at any time during this procedure a MAP_P_ABORT, MAP_U_ABORT, MAP_NOTICE or MAP_CLOSE indication concerning the process is received from the VLR, the process is terminated. If a MAP_NOTICE indication was received from the VLR, the VLR dialogue must also be aborted by sending a MAP_U_ABORT request indicating Procedure error towards the VLR. Possible signalling to the MS is described in GSM 04.10.

If an A_CM_RELEASE indication is received from the MS, all open transactions are released using the MAP_U_ABORT request indicating application procedure cancellation; the process terminates.

The invocation procedure in the MSC is shown in figure 19.6.2/1.

Figure 19.6.2/1 (sheet 1 of 2)
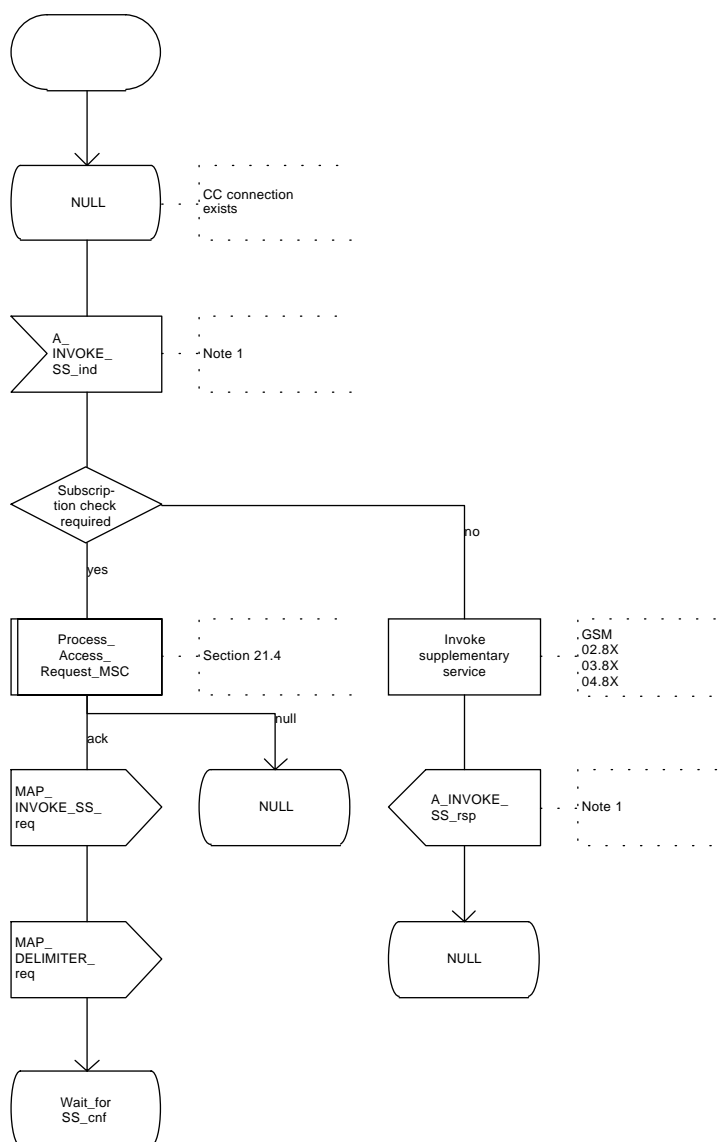
Note 1: Fictitious signal to indicate receipt/sending of SS invocation invoke component on the air interface (eg. BuildMPTY). Described in GSM 04.8X and 09.11.

Figure 19.6.2/1 (sheet 2 of 2)
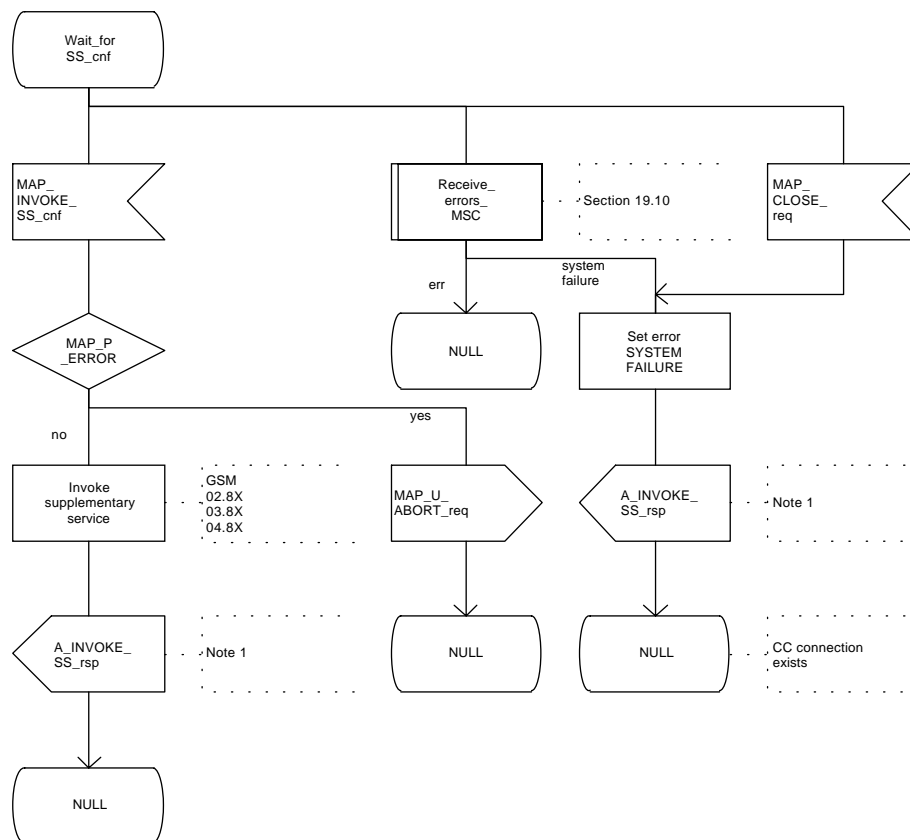
Note 1: Fictitious signal to indicate receipt/ sending of SS invocation invoke component on the air interface, (eg. BuildMPTY). Described in GSM 04.8X and 09.11.

19.6.3 Procedures in the VLR

Process Access Request

When receiving the MAP_PROCESS_ACCESS_REQUEST indication, the VLR acts as described in clause 21 of this ETS.

Supplementary service invocation

When receiving the MAP_INVOKE_SS indication, the MAP user acts as follows:

- if the operator has barred the subscriber from access to supplementary services, the error "Call Barred" is returned to the MSC. The parameter "operatorBarring" shall be included with the error;

- if any irrelevant information elements (according to the service description) or invalid information element values are present in the service request, then the unexpected data value error is returned to the MSC in the MAP_INVOKE_SS response;

- if the VLR does not support the invoked supplementary service then the VLR shall respond with the SS Not Available error;

- if the requested supplementary service cannot be invoked by subscriber actions, then the VLR shall respond with the Illegal SS Operation error;

- if the subscriber is not provided with (i.e. subscribed to) the requested supplementary service, then the SS error status error (possibly including the SS-Status as parameter) is returned to the MSC in the MAP_INVOKE_SS response.

If all checks are passed the VLR returns an empty MAP_INVOKE_SS response to the MSC, thus indicating that the invocation request was accepted.

If at any time during this procedure a MAP_P_ABORT, MAP_U_ABORT, MAP_NOTICE or unexpected MAP_CLOSE indication concerning the process is received from the MSC, the process terminates. If a MAP_NOTICE indication was received from the MSC, that dialogue must be aborted by sending a MAP_U_ABORT request indicating Procedure error towards the MSC. The process terminates.

The invocation procedure in the VLR is shown in figure 19.6.3/1.

Figure 19.6.3/1
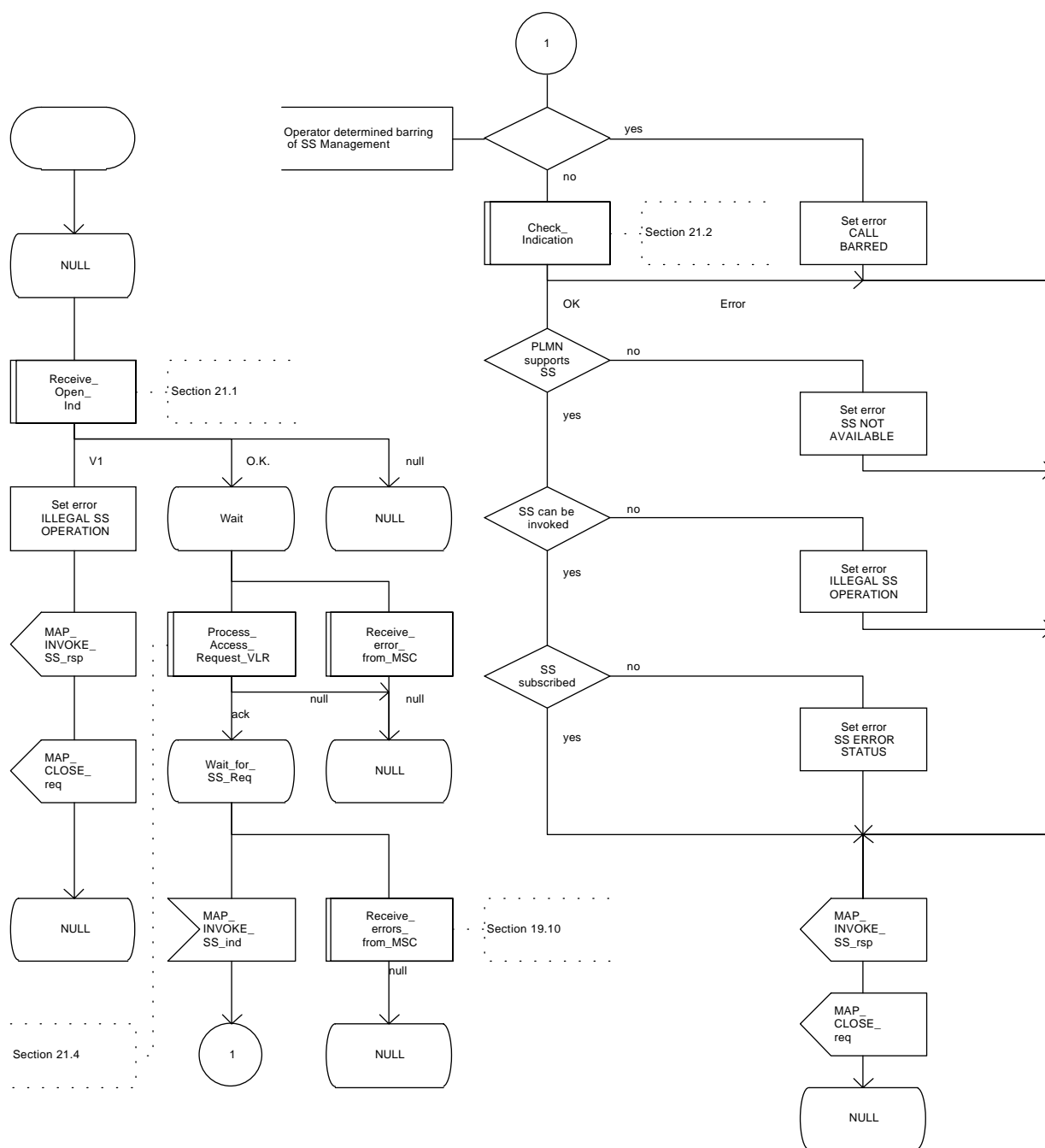
19.7   Password registration procedure

19.7.1   General

The password registration procedure is used to register a password in the HLR. The password registration procedure is a fully transparent communication between the MS and the HLR, except that some services may be invoked as a result of the procedure, as described below.

The password registration procedure is shown in figure 19.7.1/1.

The following services may be used:

MAP_PROCESS_ACCESS_REQUEST	(defined in clauses 6 and 21);
MAP_TRACE_SUBSCRIBER_ACTIVITY	(defined in clauses 7 and 21);
MAP_PROVIDE_IMSI	(defined in clauses 6 and 21);
MAP_FORWARD_NEW_TMSI	(defined in clauses 6 and 21);
MAP_AUTHENTICATE	(defined in clauses 6 and 21);
MAP_SET_CIPHERING_MODE	(defined in clauses 6 and 21);
MAP_CHECK_IMEI	(defined in clauses 6 and 21);
MAP_READY_FOR_SM	(defined in clauses 10 and 21);
MAP_GET_PASSWORD	(defined in clause 9).

NOTE 1:   For details of the procedure on the radio path, see GSM 04.08, 04.10, 04.8x and 04.9x. Services shown in dotted lines are triggers/ triggered signalling on the radio path.

NOTE 2:   For details on the Process Access Request procedure, please refer to clause 21 in this ETS.

NOTE 3:   Use of each of the three MAP_GET_PASSWORD operations is described in subclause 19.7.4.

Figure 19.7.1/1: Interfaces and services for supplementary service password registration

19.7.2   Procedures in the MSC

The password registration procedure in the MSC is identical to that for activation specified in subclause 19.3.2. All the text and diagrams in subclause 19.3.2 apply with all references to activation changed to password registration.

19.7.3   Procedures in the VLR

The password registration procedure in the VLR is identical to that for activation specified in subclause 19.3.3. All the text and diagrams in subclause 19.3.3 apply with all references to activation changed to password registration.

19.7.4   Procedures in the HLR

The procedure in the HLR is initiated when it receives a MAP_REGISTER_PASSWORD indication.

The HLR acts as follows:

-       if the operator has barred the subscriber for access to supplementary services, the Call Barred error is returned to the VLR. The parameter "operatorBarring" shall be included with the error;

-       if any irrelevant information elements (according to the service description) or invalid information element values are present, then the unexpected data value error is returned to the VLR in the response. This error should thus be returned if the SS-Code provided by the mobile subscriber is not allocated.

The HLR shall then process the MAP_REGISTER_PASSWORD indication as specified in GSM 03.11. During the handling of password registration, the password procedure will be initiated (as specified in GSM 03.11) This will involve the sending of MAP_GET_PASSWORD requests to the VLR.

-       Handling of receipt of MAP_P_ABORT, MAP_U_ABORT or MAP_CLOSE indications from the VLR is identical to their handling in the registration procedure, see subclause 19.1.4 above.

The password registration procedure in the HLR is shown in figure 19.7.4/1.

Figure 19.7.4/1 (sheet 1 of 2)
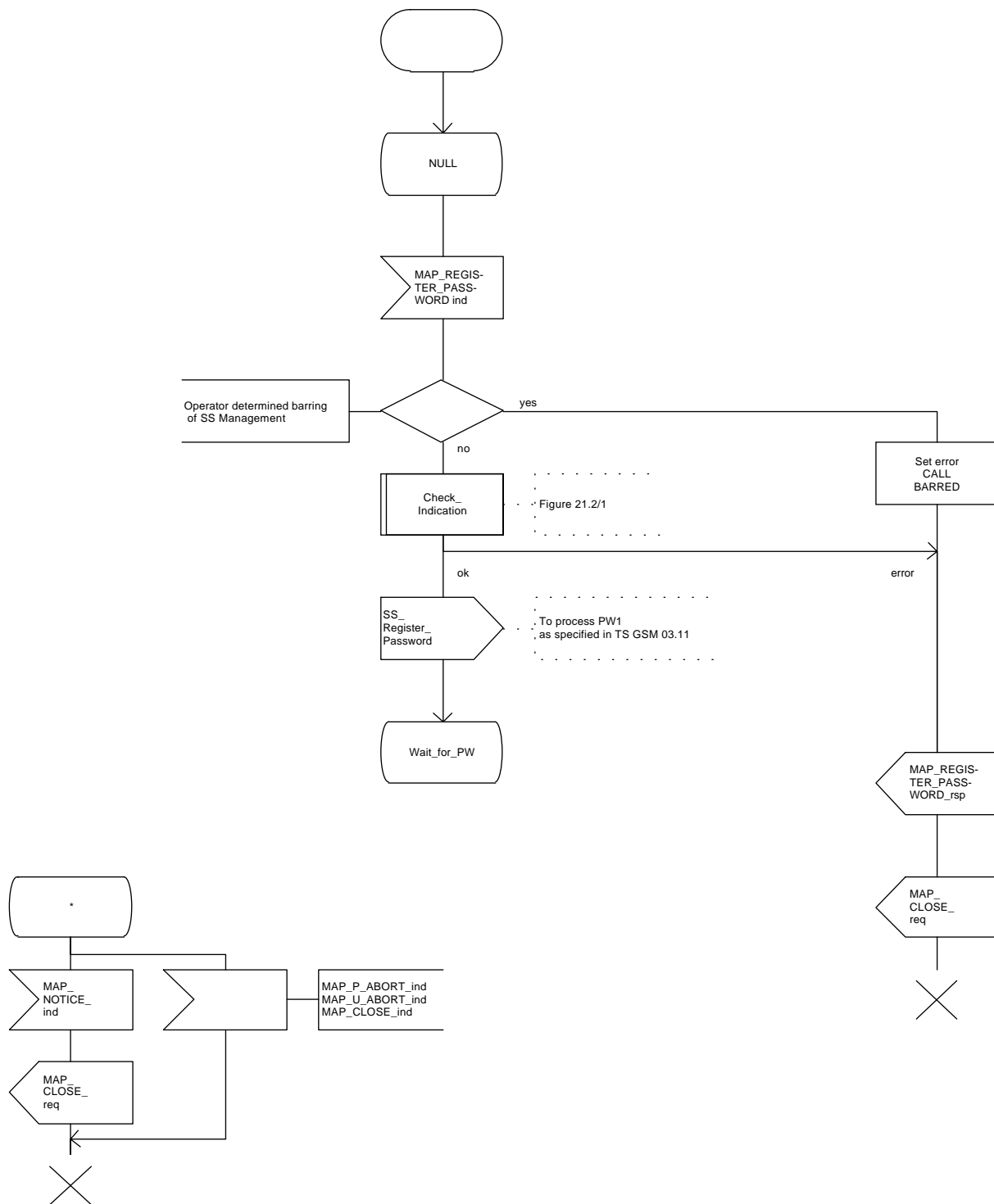
Figure 19.7.4/1 (sheet 2 of 2)
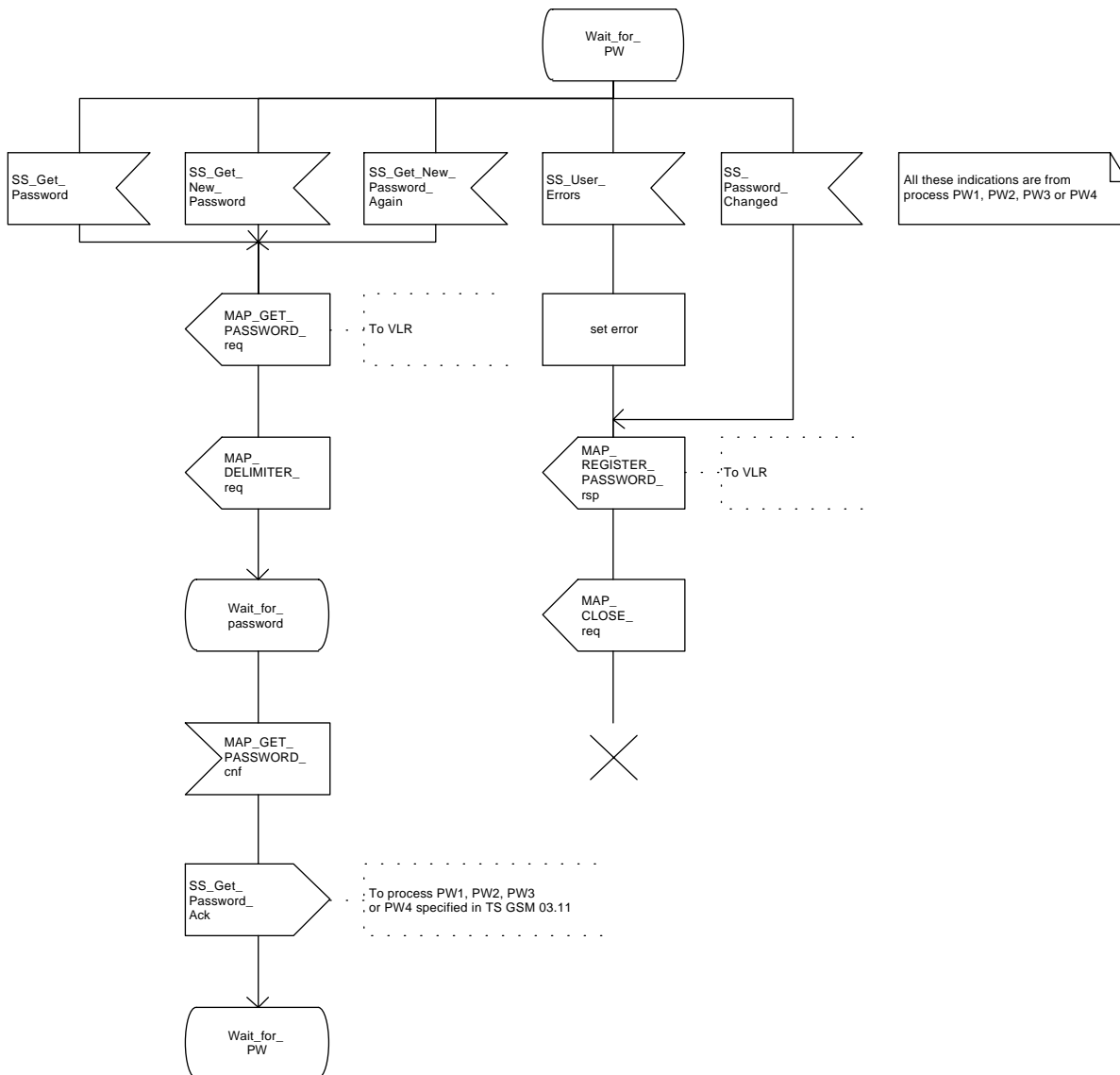
19.8   Mobile Initiated USSD procedure

19.8.1   General

The procedure supports supplementary service signalling procedures which can allow PLMN specific services to be introduced.

The message flow for the procedure can be found in GSM 03.90.

The following services may be used:

MAP_PROCESS_ACCESS_REQUEST	(defined in clauses 6 and 21);
MAP_TRACE_SUBSCRIBER_ACTIVITY	(defined in clauses 7 and 21);
MAP_PROVIDE_IMSI	(defined in clauses 6 and 21);
MAP_FORWARD_NEW_TMSI	(defined in clauses 6 and 21);
MAP_AUTHENTICATE	(defined in clauses 6 and 21);
MAP_SET_CIPHERING_MODE	(defined in clauses 6 and 21);
MAP_CHECK_IMEI	(defined in clauses 6 and 21);
MAP_READY_FOR_SM	(defined in clauses 10 and 21);
MAP_UNSTRUCTURED_SS_REQUEST	(defined in clause 9);
MAP_UNSTRUCTURED_SS_NOTIFY	(defined in clause 9).

The following service is certainly used:

MAP_PROCESS_UNSTRUCTURED_SS_REQUEST   (defined in clause 9).

19.8.2   Procedures in the MSC

Before the Process Unstructured SS Request service can be invoked, a call independent CM connection must be created between the MS and the MSC.

Once a CM-connection is established, the MSC may handle the A_PROCESS_UNSTRUCTURED_SS_REQUEST from the MS. This message contains information input by the user, the message may be fed to an application contained locally in the MSC or to the VLR. The rules for determining this are specified in GSM 03.90.

1)   Message Destined for VLR

If the message is destined for the VLR then the MSC shall transfer the message to the VLR using the mapping specified in detail in GSM 09.11.

The MSC may subsequently receive one or more MAP_UNSTRUCTURED_SS_REQUEST or MAP_UNSTRUCTURED_SS_NOTIFY indications from the VLR. These shall be sent transparently to the MS. When a confirmation is received from the MS this shall be returned to the VLR.

When the MSC receives a MAP_PROCESS_UNSTRUCTURED_SS_REQUEST confirmation from the VLR then it shall pass this to the MS and initiate release of the CM connection.

2)   Message Destined for Local Application

If the message is destined for the local USSD application then the MSC shall transfer the message to the application.

The MSC may subsequently receive one or more requests from the application which correspond to the MAP_UNSTRUCTURED_SS_REQUEST or MAP_UNSTRUCTURED_SS_NOTIFY indications. These shall be sent transparently to the MS. When a confirmation is received from the MS this shall be returned to the application.

When the MSC receives the result of the original operation from the application then it shall pass this to the MS and initiate release of the CM connection.

Error Handling

Both the MS and the VLR or USSD Application may initiate release of the CM-connection at any time. This is handled as shown in the diagrams.

The procedure in the MSC is shown in figure 19.8.2/1.

Figure 19.8.2/1 (sheet 1 of 2)

Figure 19.8.2/1 (sheet 2 of 2)
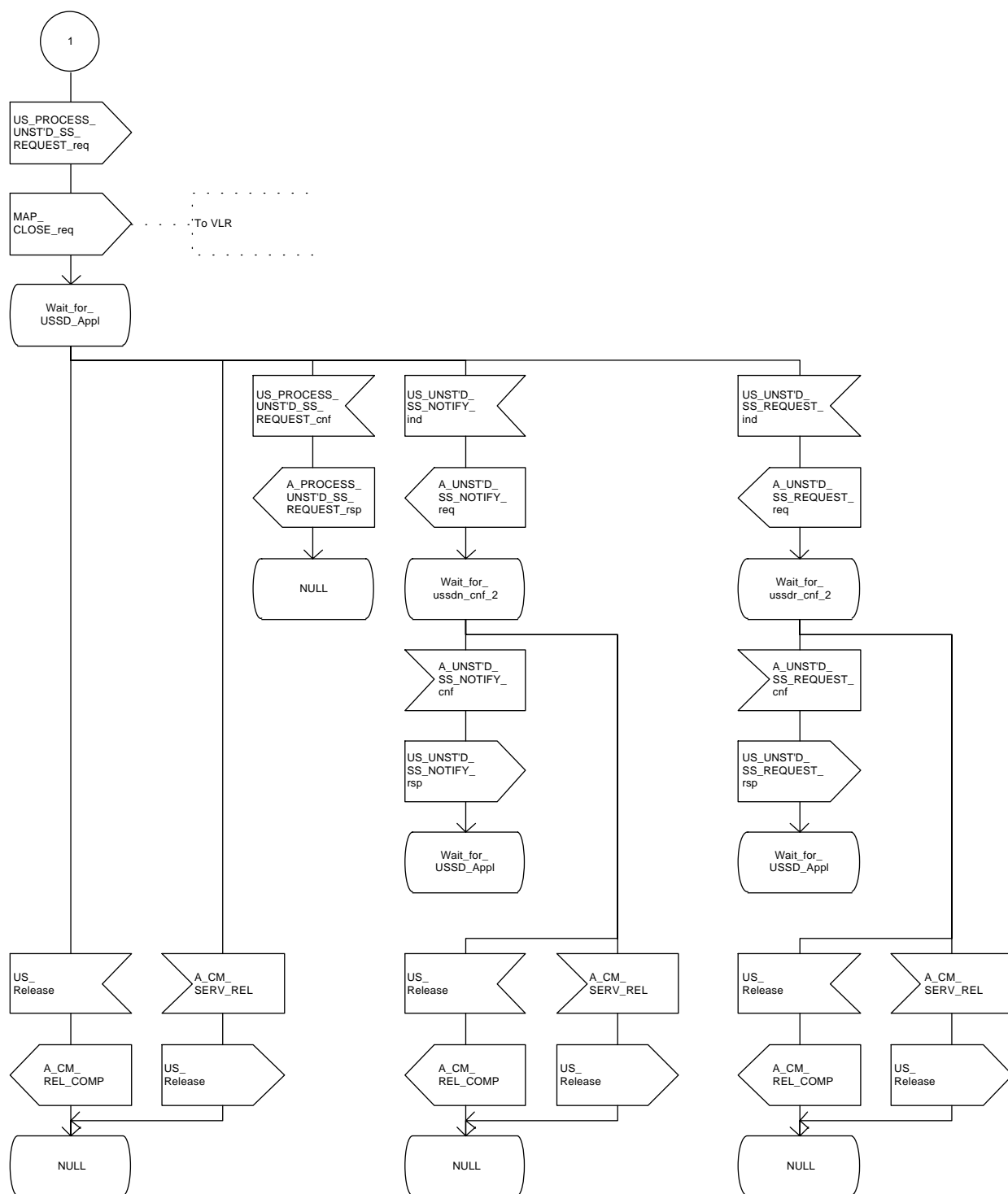
19.8.3   Procedures in the VLR

The initiation of the process is shown in subclause 19.0.2.

Once a MAP dialogue is established, the VLR may handle the MAP_PROCESS_UNSTRUCTURED_SS_REQUEST from the MSC. This message contains information input by the user, the message may be fed to an application contained locally in the VLR or to the HLR. The rules for determining this are specified in GSM 03.90.

Message Destined for HLR

If the message is destined for the HLR then the VLR shall transfer the message transparently to the HLR.

The VLR may subsequently receive one or more MAP_UNSTRUCTURED_SS_REQUEST or MAP_UNSTRUCTURED_SS_NOTIFY indications from the HLR. These shall be sent transparently to the MSC. When a confirmation is received from the MSC this shall be returned to the HLR.

When the VLR receives a MAP_PROCESS_UNSTRUCTURED_SS_REQUEST confirmation from the HLR then it shall pass this to the MS and close the MAP provider service.

Message Destined for Local Application

If the message is destined for the local USSD application then the VLR shall transfer the message to the application.

The VLR may subsequently receive one or more requests from the application which correspond to the MAP_UNSTRUCTURED_SS_REQUEST or MAP_UNSTRUCTURED_SS_NOTIFY indications. These shall be sent transparently to the MSC. When a confirmation is received from the MSC this shall be returned to the application.

When the VLR receives the result of the original operation from the application then it shall pass this to the MSC and initiate release of the CM connection.

Error Handling

Both the MSC and the HLR or USSD Application may initiate release of the MAP service at any time. This is handled as shown in the diagrams.

The procedure in the VLR is shown in figures 19.8.3/1 and 19.8.3/2.

Figure 19.8.3/1 (sheet 1 of 3)
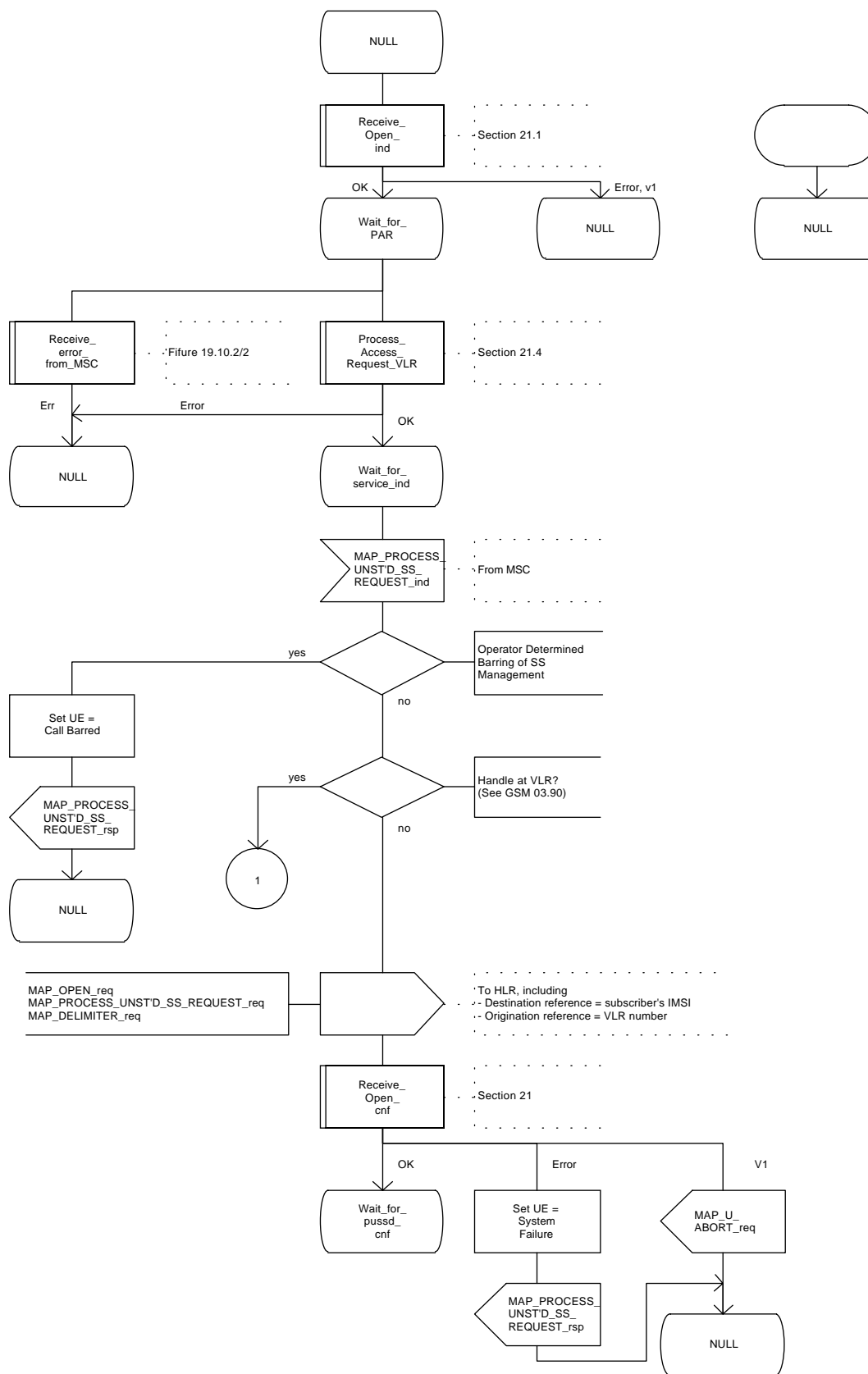
Figure 19.8.3/1 (sheet 2 of 3)
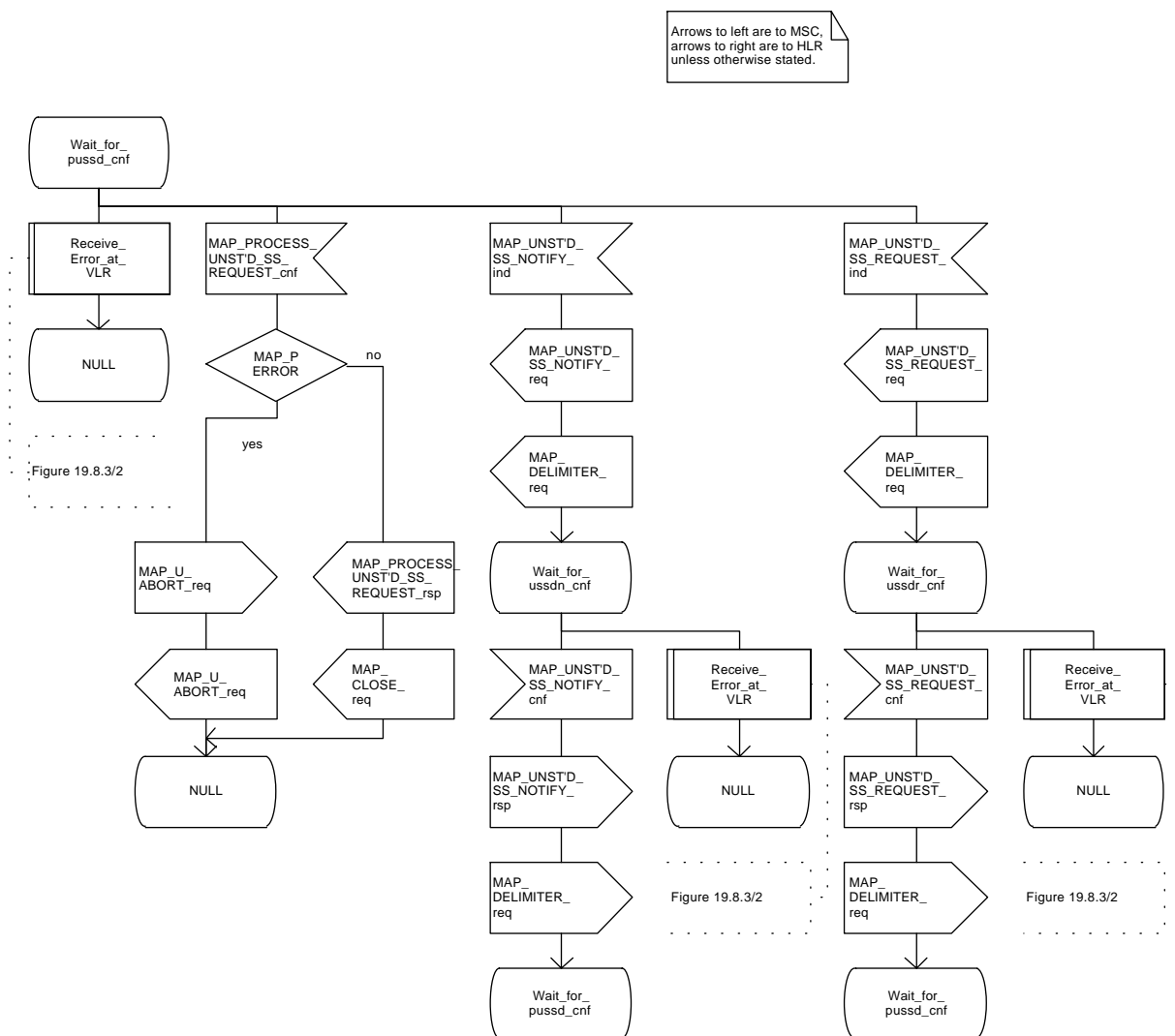
Arrows to left are to MSC,
arrows to right are to HLR
unless otherwise stated.

Figure 19.8.3/1 (sheet 3 of 3)
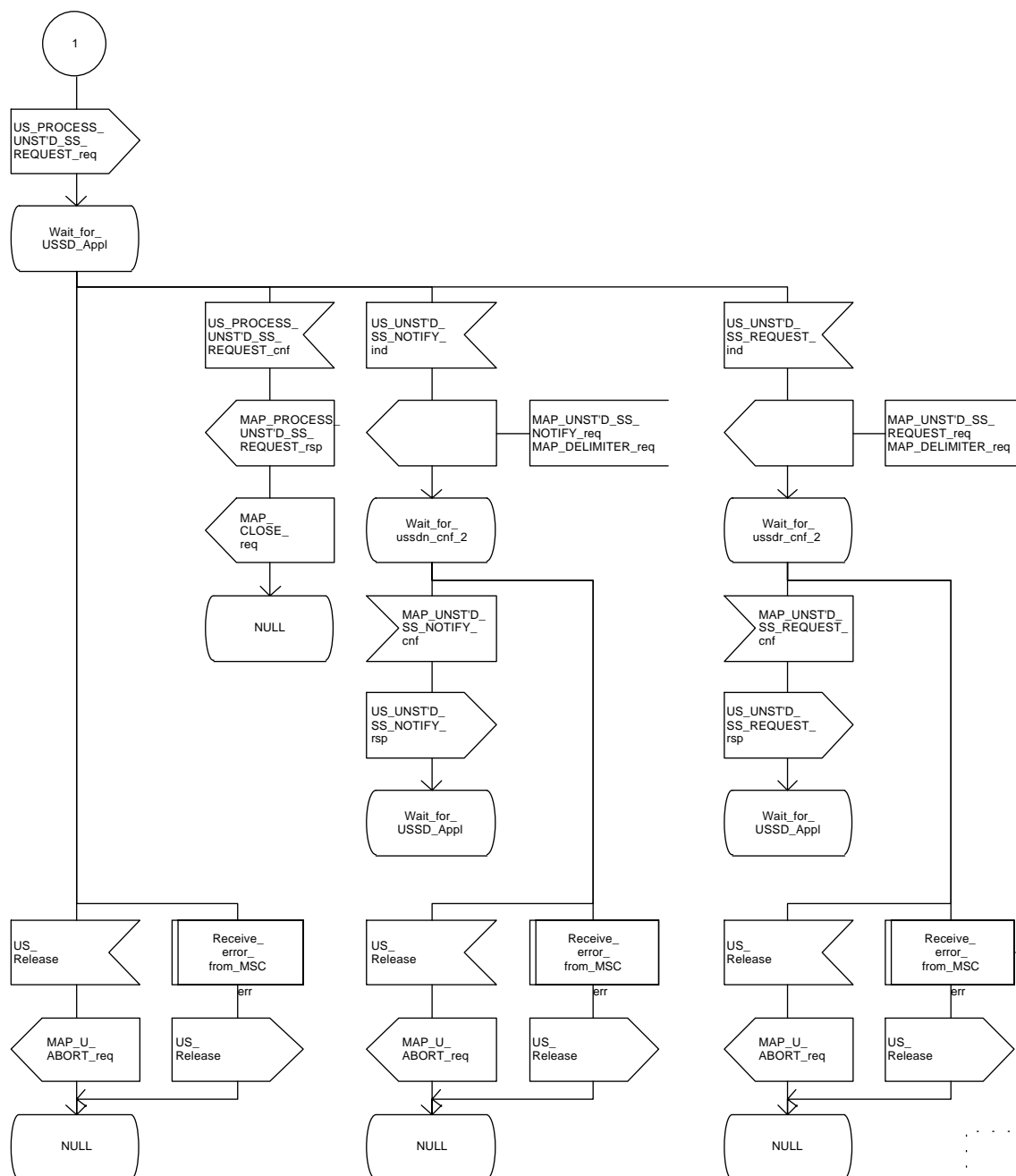
Figure 19.8.3/2
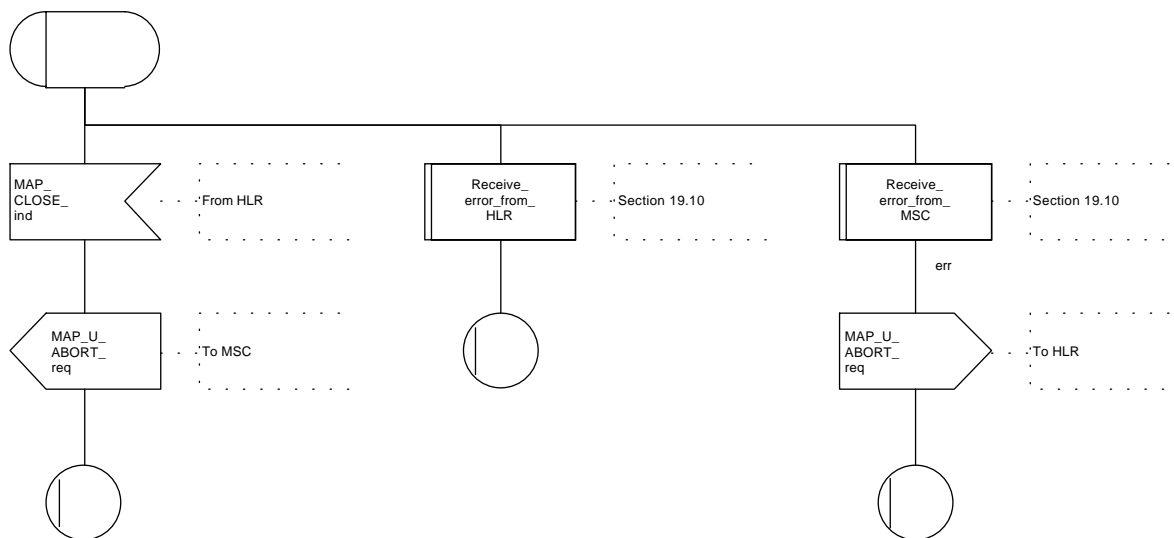
19.8.4 Procedures in the HLR

The initiation of the process is shown in subclause 19.0.3.

Once a MAP dialogue is established, the HLR may handle the MAP_PROCESS_UNSTRUCTURED_SS_REQUEST from the VLR. This message contains information input by the user. If the alphabet used for the message is understood then the message shall be fed to an application contained locally in the HLR. If the alphabet is not understood then the error "UnknownAlphabet" shall be returned.

The HLR may subsequently receive one or more requests from the application which correspond to the MAP_UNSTRUCTURED_SS_REQUEST or MAP_UNSTRUCTURED_SS_NOTIFY indications. These shall be sent transparently to the VLR. When a confirmation is received from the VLR this shall be returned to the application.

When the HLR receives the result of the original operation from the application then it shall pass this to the VLR and initiate release of the CM connection.

Error Handling

Both the VLR and the USSD Application may initiate release of the MAP service at any time. This is handled as shown in the diagrams.

The procedure in the HLR is shown in figure 19.8.4/1.

Figure 19.8.4/1 (sheet 1 of 2)

Arrows to left are to VLR
unless otherwise stated.

NULL

Receive_
Open_
ind . . . .
. . Section 21.1
. . . .

OK

Error, V1

Wait_for_
service_ind

NULL

MAP_PROCESS_
UNST'D_SS_
REQUEST_ind

Operator Determined
Barring of SS
Management

no

yes

*

alphabet
supported

no

MAP_
NOTICE_
ind

MAP_P_ABORT_ind
MAP_U_ABORT_ind
MAP_CLOSE_ind

yes

US_PROCESS_
UNST'D_SS_
REQUEST_req

Set UE =
Unknown
Alphabet

Set UE =
Call
Barred

MAP_
CLOSE_
req

Wait_for_
USSD_Appl

MAP_PROCESS_
UNST'D_SS_
REQUEST_rsp

SS_
Release

NULL

NULL

Figure 19.8.4/1 (sheet 2 of 2)

Arrows to left are to VLR,
arrows to right are to USSD application
unless otherwise stated.

19.9 Network initiated USSD procedure

19.9.1 General

The procedure supports supplementary service signalling procedures which can allow PLMN specific services to be introduced.

The message flow for the procedure can be found in GSM 03.90.

The following services may be used:

MAP_PAGE	(defined in clauses 6 and 21);
MAP_SEARCH_FOR_MOBILE_SUBSCRIBER	(defined in clauses 6 and 21);
MAP_PROCESS_ACCESS_REQUEST	(defined in clauses 6 and 21);
MAP_AUTHENTICATE	(defined in clauses 6 and 21);
MAP_SET_CIPHERING_MODE	(defined in clauses 6 and 21);
MAP_FORWARD_NEW_TMSI	(defined in clauses 6 and 21);
MAP_READY_FOR_SM	(defined in clauses 10 and 21).

At least one of the following services will certainly be used, and both may be used:

MAP_UNSTRUCTURED_SS_REQUEST	(defined in clause 9);
MAP_UNSTRUCTURED_SS_NOTIFY	(defined in clause 9).

19.9.2 Procedure in the MSC

The procedure may be invoked either by the VLR or by a USSD application local to the MSC. They may start by using either the MAP_UNSTRUCTURED_SS_REQUEST or MAP_UNSTRUCTURED_SS_NOTIFY service. If the request is initiated by a local USSD application then the MSC will open a dialogue with the HLR.

In both cases the MSC will initiate a CM connection to the MS (using the page or search macros defined in subclause 21.3). Once the connection is successfully established the message received from the VLR or USSD application will be sent to the MS using the mapping specified in GSM 09.11.

Following transfer of the message the MSC will wait for a confirmation from the MS. This will be sent to the VLR or USSD application as appropriate.

Following this, the MSC may receive further uses of the MAP_UNSTRUCTURED_SS_REQUEST or MAP_UNSTRUCTURED_SS_NOTIFY services, or may receive an indication to release the connection to the MS.

In the event of an error, the connection to the MS shall be released, and the MAP process with the VLR shall be aborted as shown in the diagram.

The procedure in the MSC is shown in figure 19.9.2/1.

Figure 19.9.2/1 (sheet 1 of 4)

Figure 19.9.2/1 (sheet 2 of 4)

Arrows to left are to MS,
arrows to right are to VLR
unless otherwise stated.

Wait_for_
USSD_ind

Receive_
Errors_MSC

NULL

Section
19.10

MAP_UNST'D_
SS_NOTIFY_
ind

A_UNST'D_
SS_NOTIFY_
req

Wait_for_
ussdn_cnf

A_UNST'D_
SS_NOTIFY_
cnf

Receive_
Errors_MSC

MAP_UNST'D_
SS_NOTIFY_
rsp

NULL

MAP_
DELIMITER_
req

Section
19.10

Wait_for_
USSD_ind

MAP_UNST'D_
SS_REQUEST_
ind

A_UNST'D_
SS_REQUEST_
req

Wait_for_
ussdr_cnf

A_UNST'D_
SS_REQUEST_
cnf

Receive_
Errors_MSC

MAP_UNST'D_
SS_REQUEST_
rsp

NULL

MAP_
DELIMITER_
req

Section
19.10

Wait_for_
USSD_ind

Figure 19.9.2/1 (sheet 3 of 4)

Figure 19.9.2/1 (sheet 4 of 4)
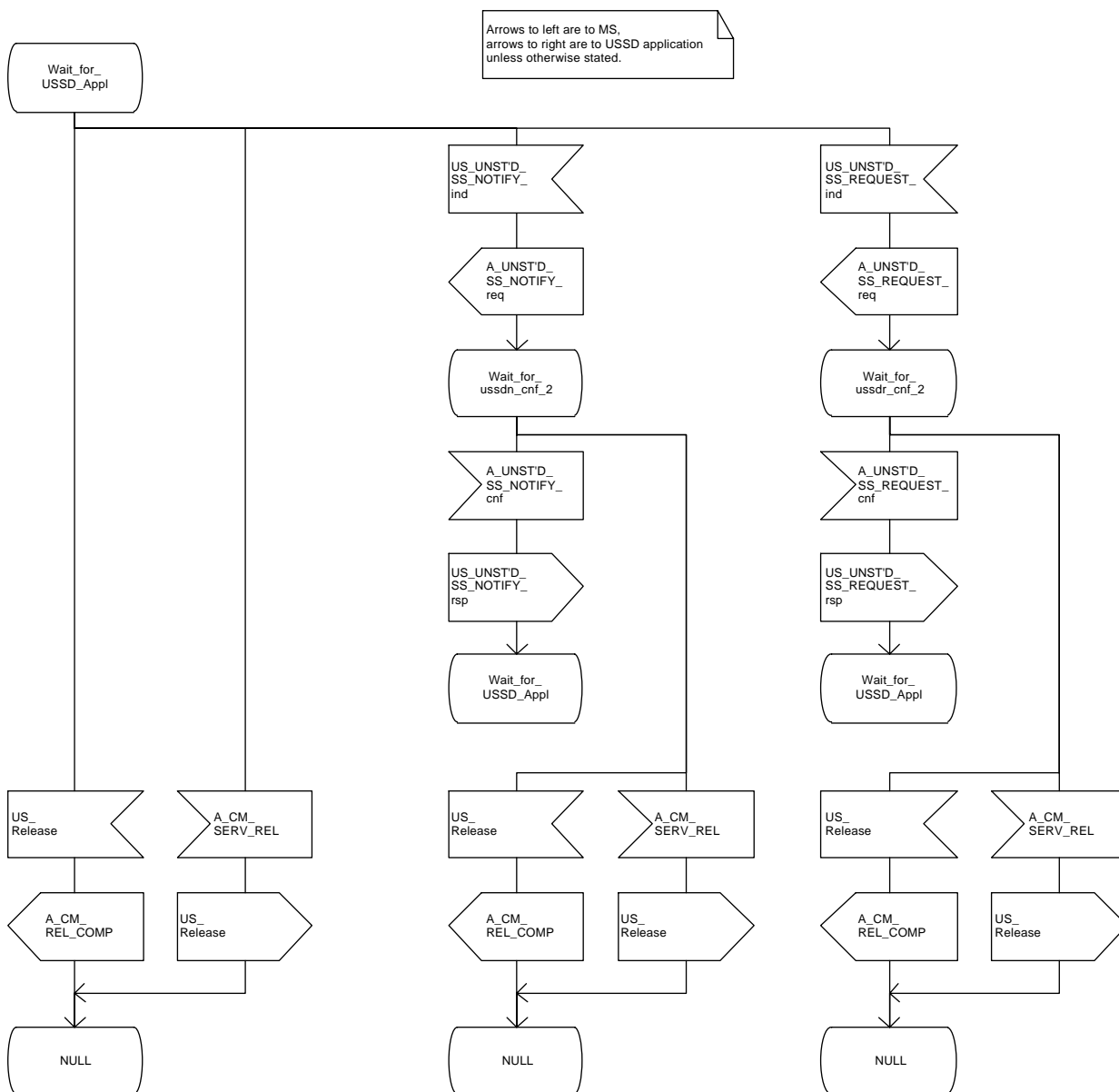
19.9.3   Procedure in the VLR

The procedure may be invoked either by the HLR or by a USSD application local to the VLR. They may start by using either the MAP_UNSTRUCTURED_SS_REQUEST or MAP_UNSTRUCTURED_SS_NOTIFY service.

In both cases the VLR will first initiate a MAP dialogue with the MSC. When the indication for the unstructured SS request or notify is received then the macro Start_USSD_VLR will be used to page the MS and open a CM connection. Once the CM connection is successfully established the indication received from the HLR or USSD application will be sent to the MSC.

Following transfer of the message the VLR will wait for a confirmation from the MSC. This will be sent to the HLR or USSD application as appropriate.

Following this, the VLR may receive further uses of the MAP_UNSTRUCTURED_SS_REQUEST or MAP_UNSTRUCTURED_SS_NOTIFY services, or may receive a MAP_CLOSE_ind.

In the event of an error, the MAP process with the MSC shall be released, and if necessary the MAP process with the HLR shall be aborted as shown in the diagram.

The procedure in the VLR is shown in figure 19.9.3/1.

MSC Initiated USSD

If a USSD application in the MSC wishes to use the network initiated USSD procedure, and a connection to the MS does not exist then it shall open a dialogue to the VLR. This dialogue will automatically lead to the VLR performing page and search using the macro Start_USSD_VLR.

Macro Start_USSD_VLR

This macro is used to initiate a CM connection with the MS for transfer of network initiated unstructured SS data.

It first checks for correct data in the VLR. If a problem is found then "Err" is returned.

A page or search procedure (as appropriate) will then be used to contact the MS. Following successful page or search the macro Process_Access_Request_VLR specified in subclause 21.4 will be used to handle the CM connection establishment.

The macro is shown in figure 19.9.3/2.

Figure 19.9.3/1 (sheet 1 of 4)

Figure 19.9.3/1 (sheet 2 of 4)

Arrows to left are to MSC,
arrows to right are to HLR
unless otherwise stated.
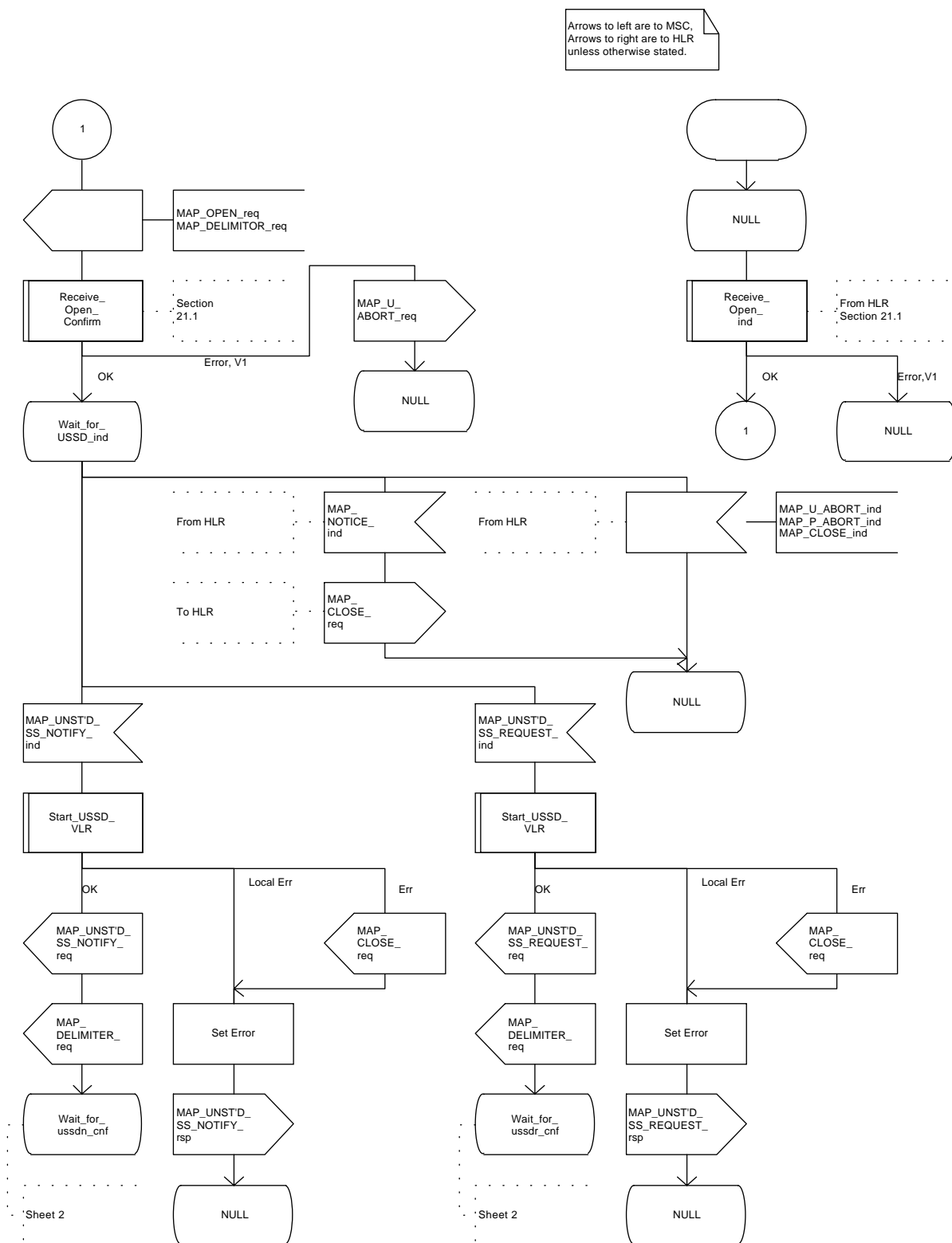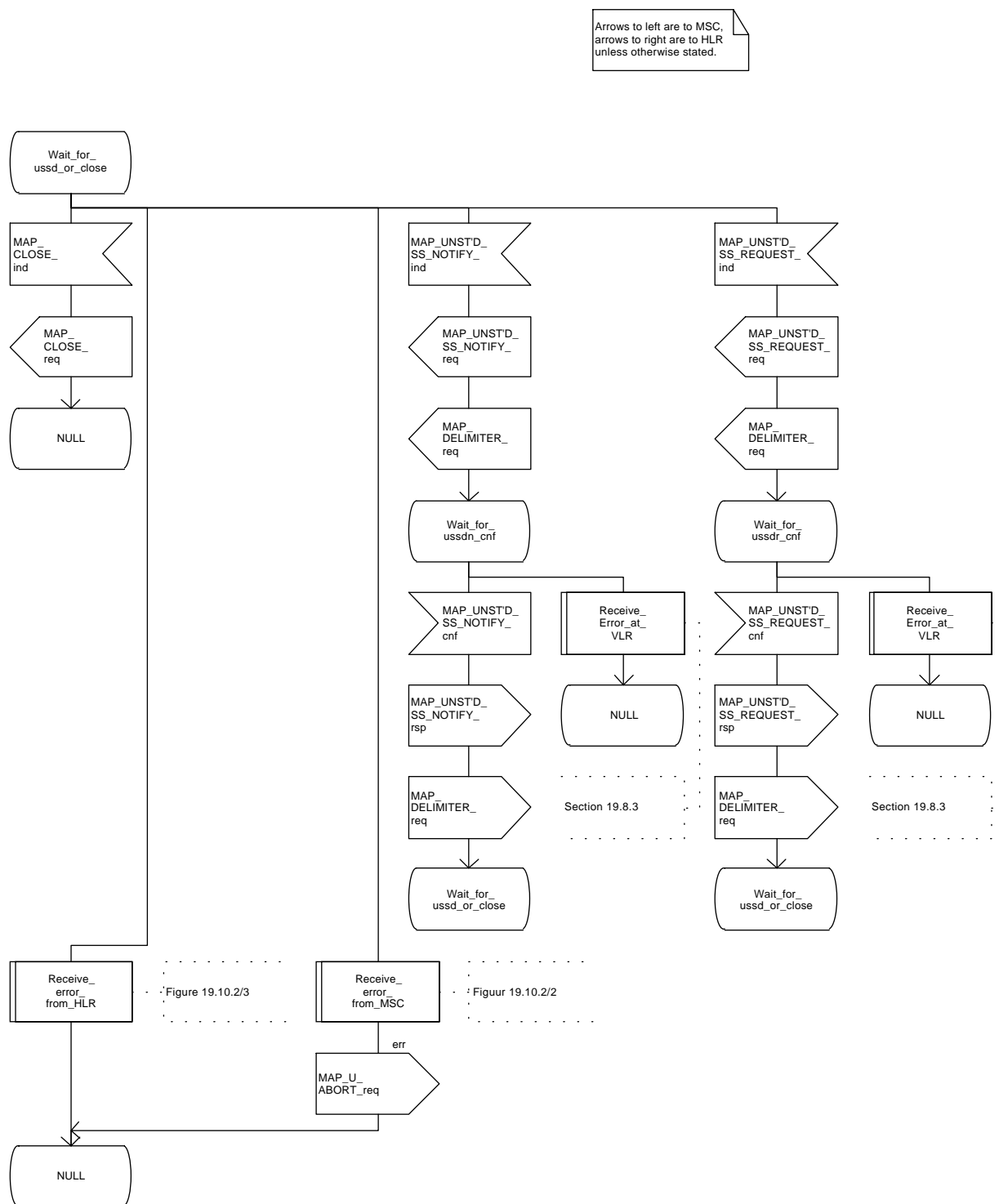
Figure 19.9.3/1 (sheet 3 of 4)
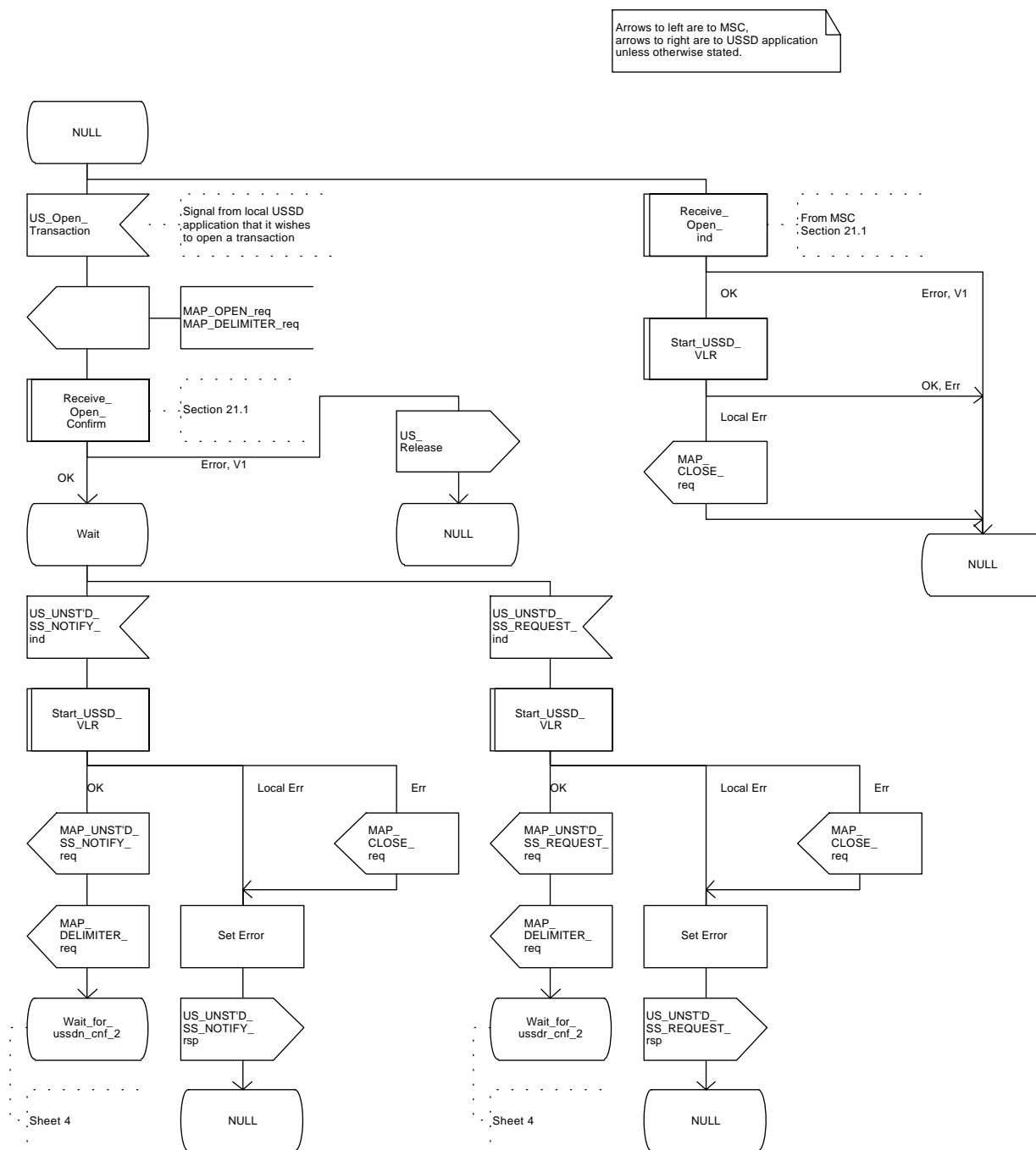
Figure 19.9.3/1 (sheet 4 of 4)

Figure 19.9.3/2 (sheet 1 of 2)

```
                    ┌──────────┐
                    │          │
                    └──────────┘
                         │
                         │              No
                    ◇ Subscriber ◇──────────────┐
                    ◇  Known?   ◇                │
                         │ Yes            ┌─────────────┐
                         │                │   Set UE=   │
                         │                │ Unidentified│
                         │                │  Subscriber │
                         │                └─────────────┘
                         │              No        │
                    ◇ Confirmed ◇────────┐        │
                    ◇  by HLR?  ◇        │        │
                         │ Yes    ┌─────────────┐ │
                         │        │   Set UE=   │ │
                         │        │   System    │ │
                         │        │   Failure   │ │
                         │        └─────────────┘ │
                         │             Yes  │      │
                    ◇   IMSI   ◇─────────────┐     │
                    ◇ Detached? ◇            │     │
                         │ No               │     │
                         │                  │     │
                         │              No  │     │
                    ◇ Roaming  ◇────────┐   │     │
                    ◇  in LA   ◇        │   │     │
                    ◇ allowed? ◇        │   │     │
                         │ Yes   ┌─────────────┐  │
                         │       │   Set UE=   │  │
                         │       │   Abscent   │  │
                         │       │  Subscriber │  │
                         │       └─────────────┘  │
                         ▼                  ▼      ▼
                       ( 2 )              ( Err )
```
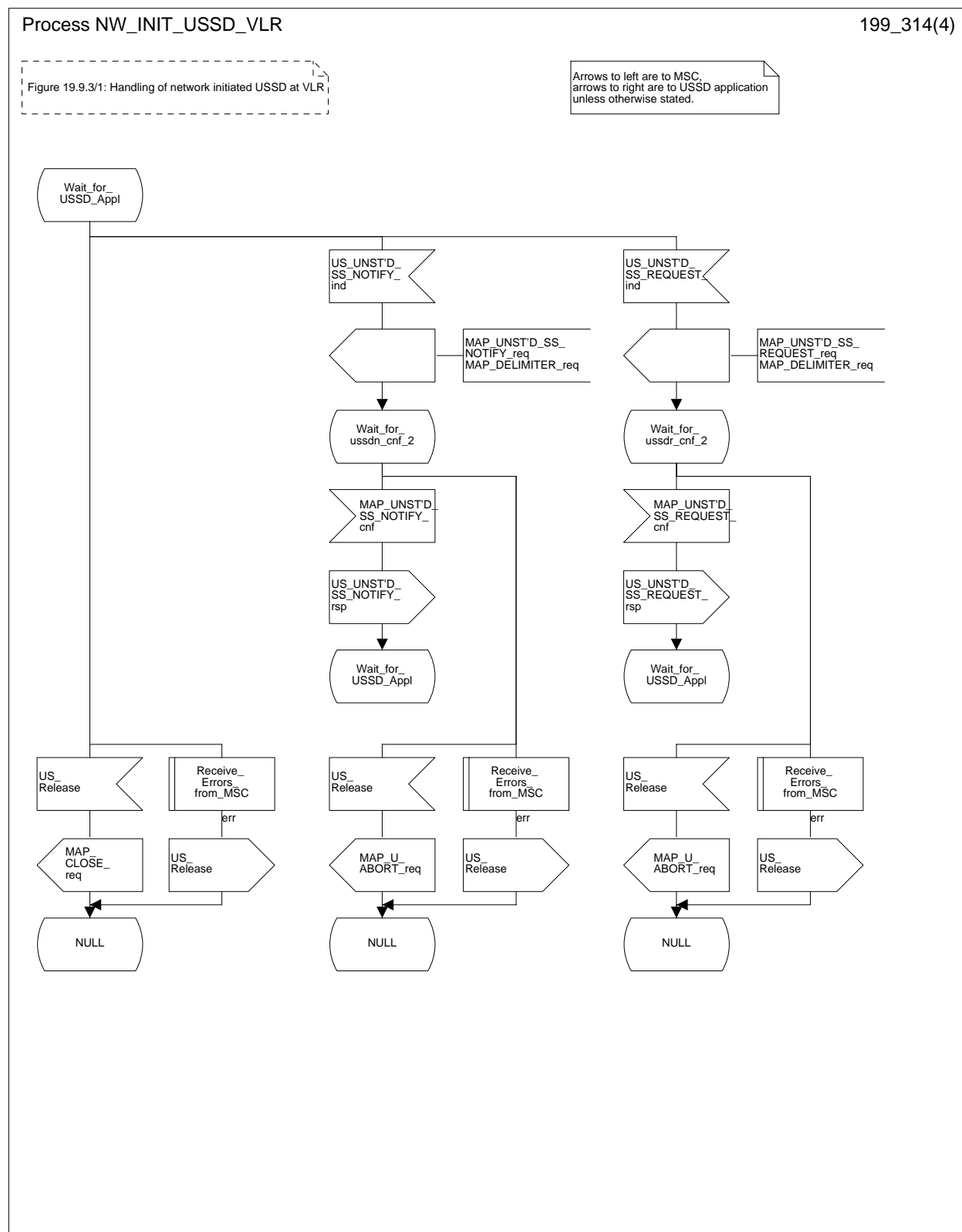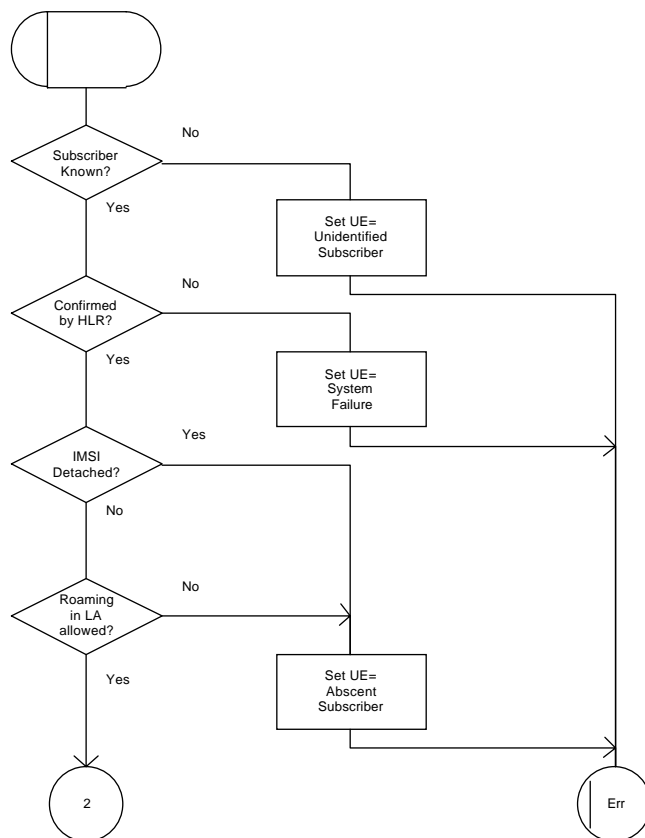
Figure 19.9.3/2 (sheet 2 of 2)
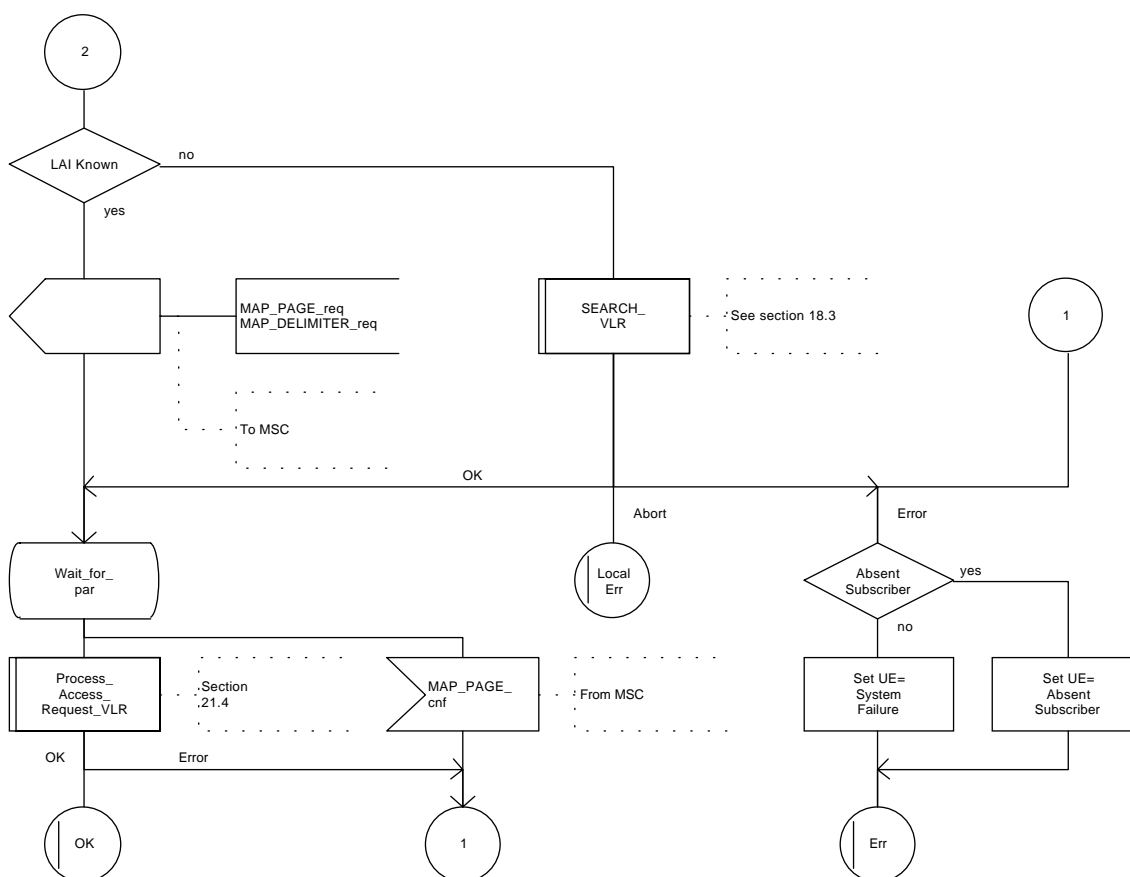
19.9.4   Procedure in the HLR

The procedure may be invoked by the USSD application local to the HLR. It may start by using either the MAP_UNSTRUCTURED_SS_REQUEST or MAP_UNSTRUCTURED_SS_NOTIFY service.

In both cases the HLR will first check whether the MS is reachable (i.e. there is a VLR identity stored in the subscriber record, the MS record is not marked as purged and the MS record is not marked "MSC Area Restricted").

If the MS is reachable, the HLR will initiate a MAP dialogue with the VLR. Once the dialogue is successfully established the message received from the USSD application will be sent to the MSC.

Following transfer of the message the HLR will wait for a confirmation from the MSC. This will be sent to the USSD application.

Following this, the HLR may receive further uses of the MAP_UNSTRUCTURED_SS_REQUEST or MAP_UNSTRUCTURED_SS_NOTIFY services, or may receive a MAP_CLOSE_ind.

In the event of an error, the MAP process with the VLR shall be released as shown in the diagram.

The procedure in the HLR is shown in figure 19.9.4/1.

Figure 19.9.4/1 (sheet 1 of 2)

Arrows to left are to VLR,
arrows to right are to USSD application
unless otherwise stated.

Figure 19.9.4/1 (sheet 2 of 2)

Arrows to left are to VLR,
arrows to right are to USSD application
unless otherwise stated.
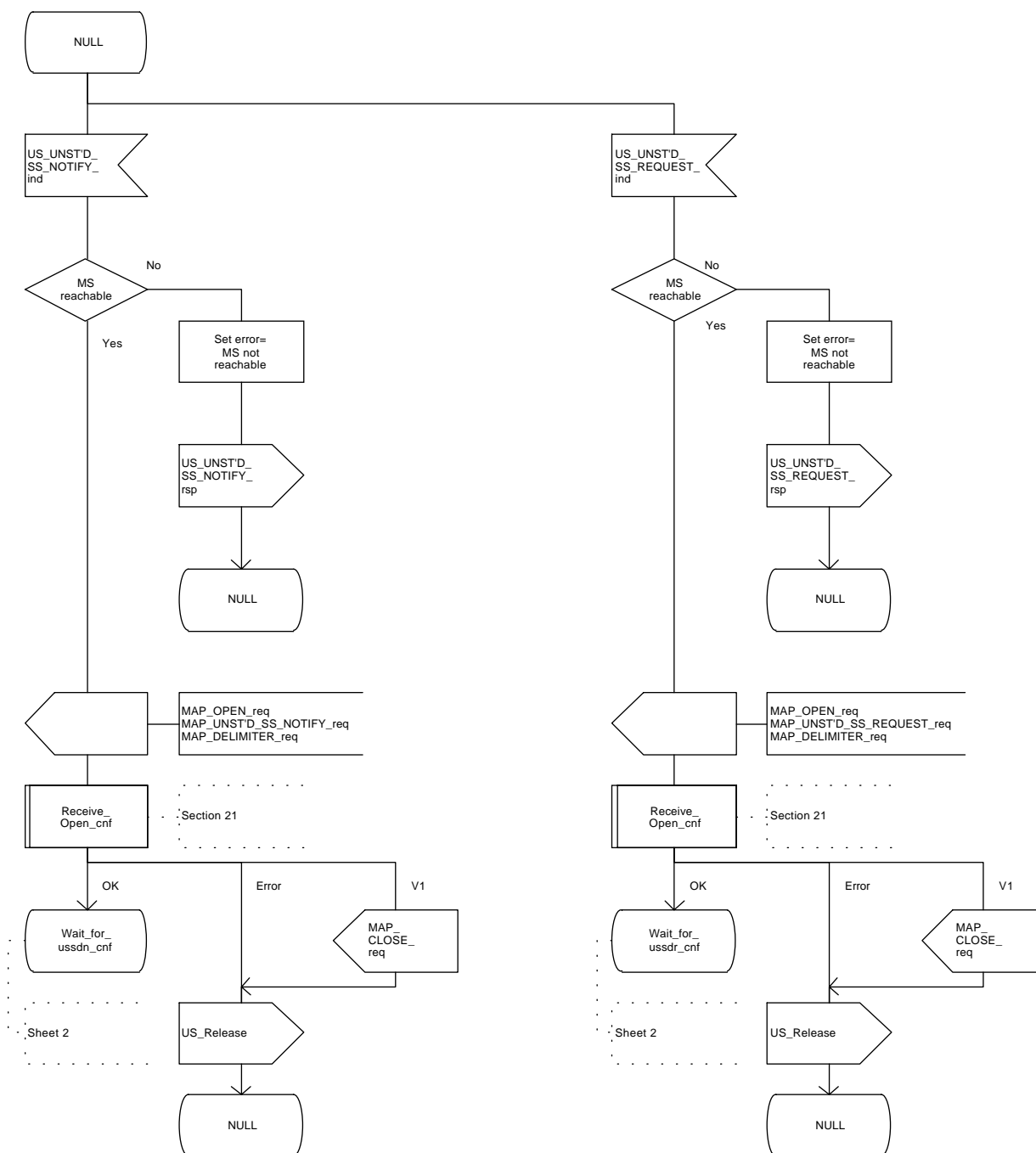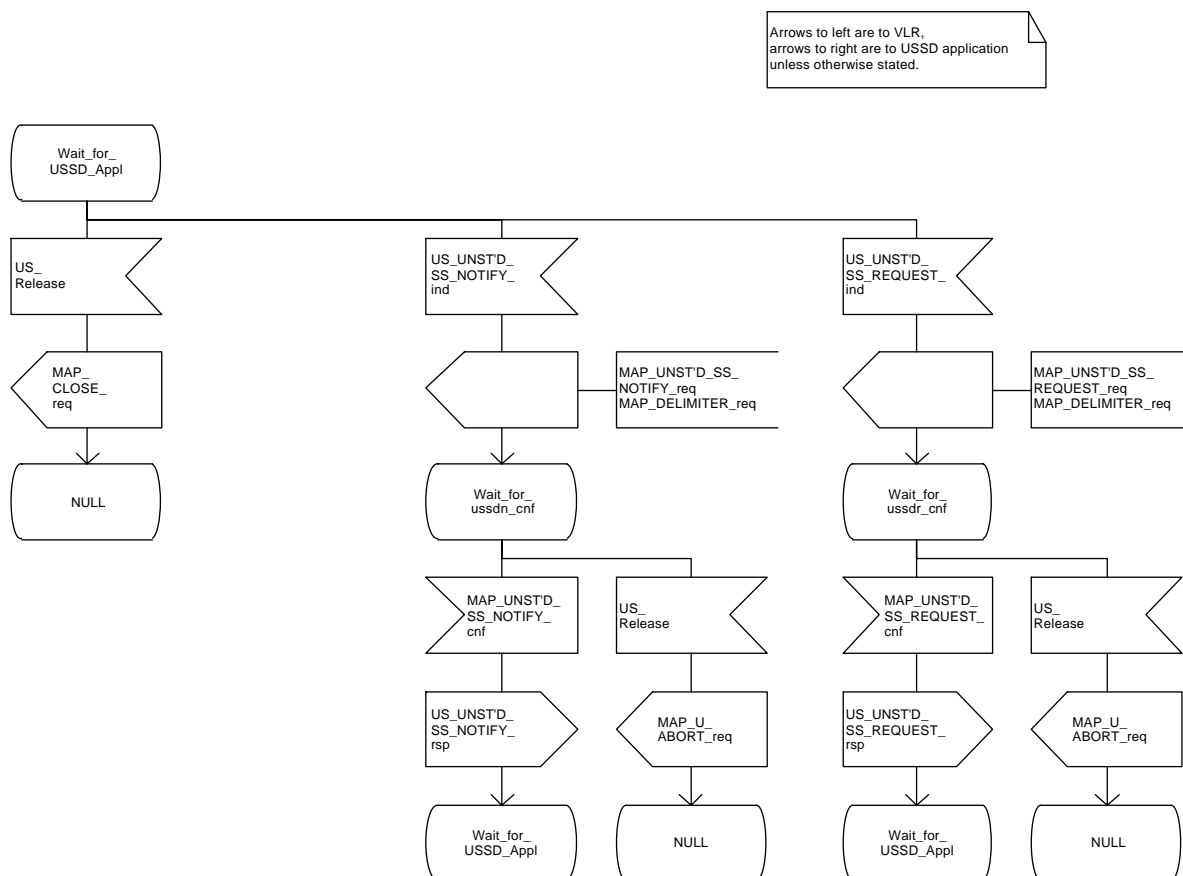
19.10   Common macros for clause 19

The following macros are used for the description of more than one of the supplementary service processes described in clause 19:

19.10.1   SS Password handling macros

Macro Get_Password_MSC

This macro is used by the MSC to relay a request for password from the VLR to the MS, and to relay a response from the MS back to the VLR. The macro is described in figure 19.10.1/1.

Macro Get_Password_VLR

This macro is used by the VLR to relay a request for password from the HLR to the MSC, and to relay a response from the MSC back to the HLR. The macro is described in figure 19.10.1/2.

Figure 19.10.1/1
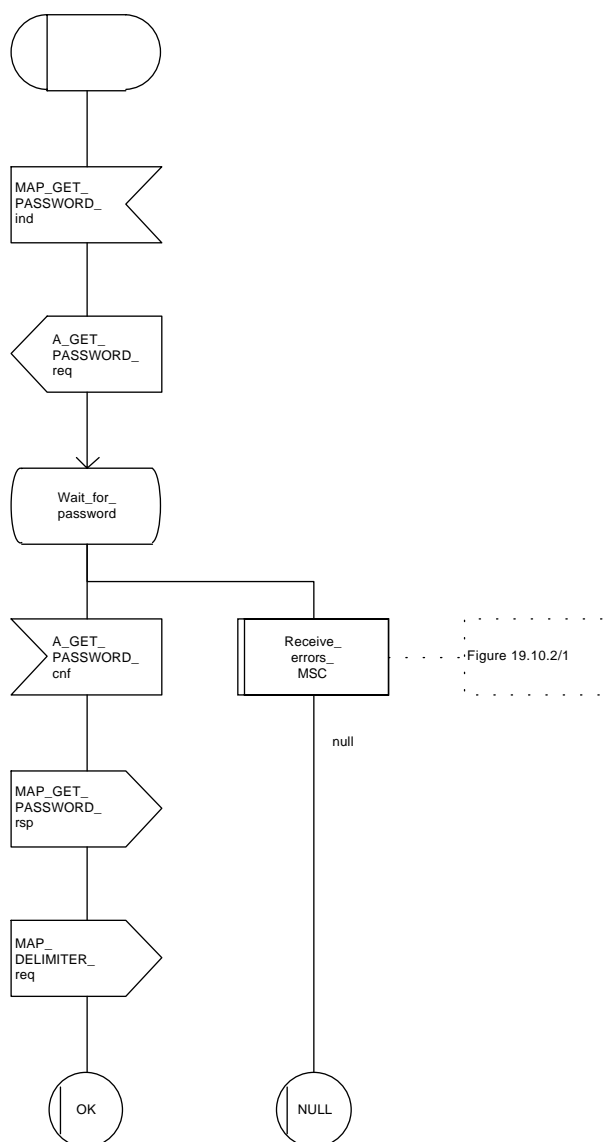
Figure 19.10.1/2
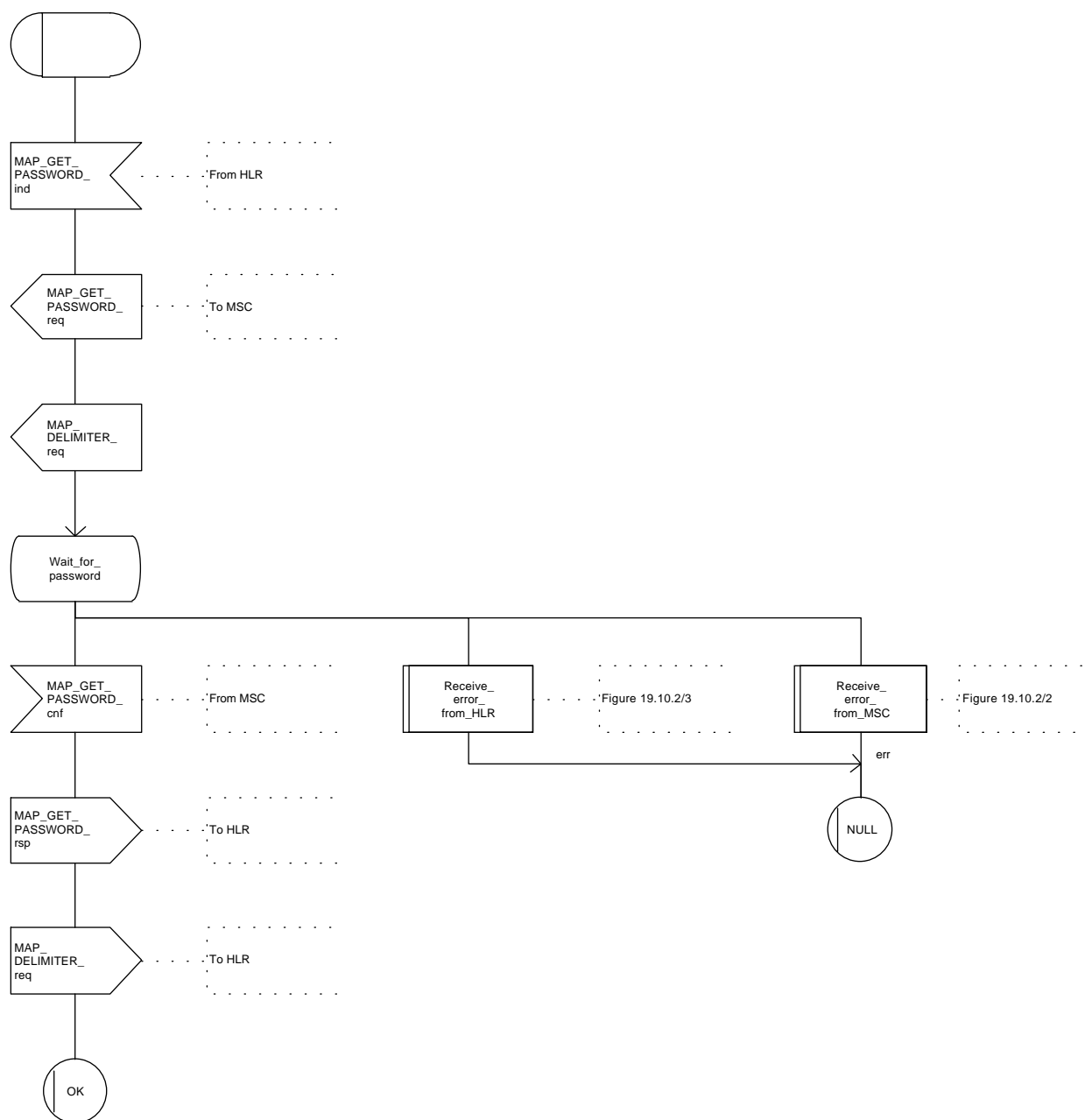
19.10.2   SS Error handling macros

Macro Receive_errors_MSC

This macro is used by the MSC to receive signals which should lead to failure if received in any state of a supplementary service process. If the air interface connection is released by the MS, the communication towards the VLR is aborted, and the MSC should return to a stable "NULL" state. If a MAP_NOTICE indication is received from the VLR, or the VLR aborts or unexpectedly closes the connection, then the air interface connection shall be released. The macro is described in figure 19.10.2/1.

Macro Receive_error_from_MSC

This macro is used by the VLR to receive signals from the MSC which should lead to failure if received in any state of a supplementary service process. If a MAP_NOTICE indication is received from the MSC, that connection is closed before the only outcome of the macro, "err" is reported back to the calling process. The macro is described in figure 19.10.2/2.

Macro Receive_error_from_HLR

This macro is used by the VLR to receive signals from the HLR which should lead to failure if received in any state of a supplementary service process. If a MAP_NOTICE indication is received from the MSC, that connection is closed. The macro is described in figure 19.10.2/3.

Figure 19.10.2/1
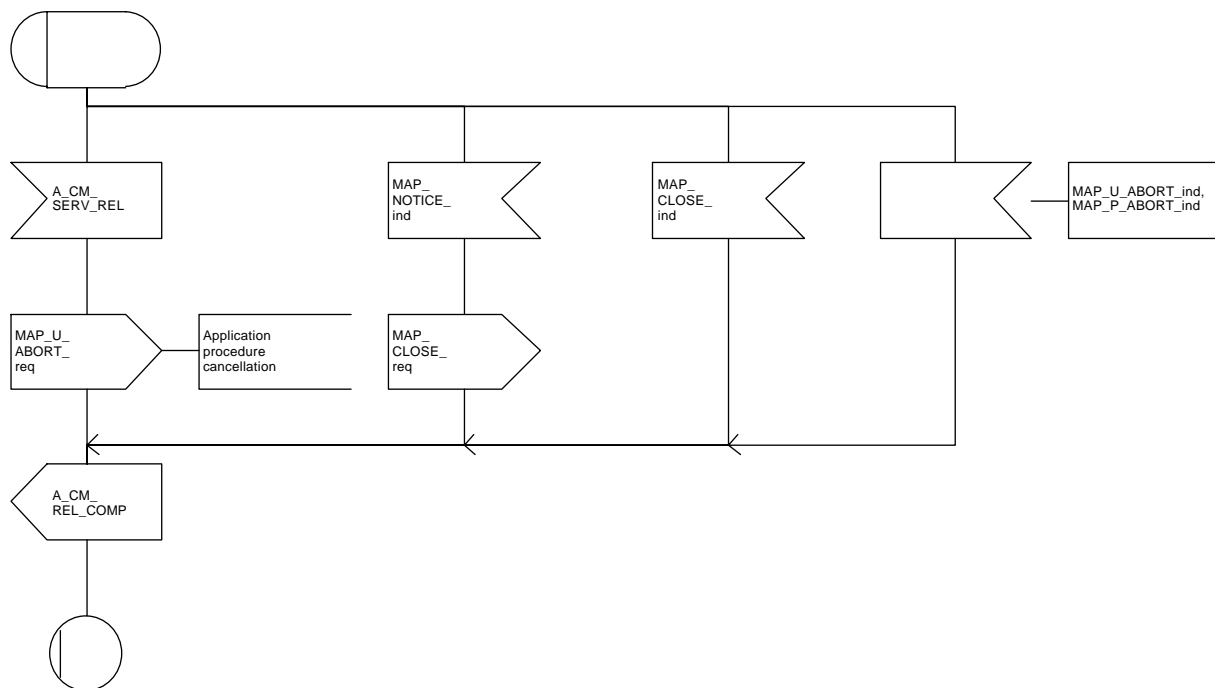
Figure 19.10.2/2
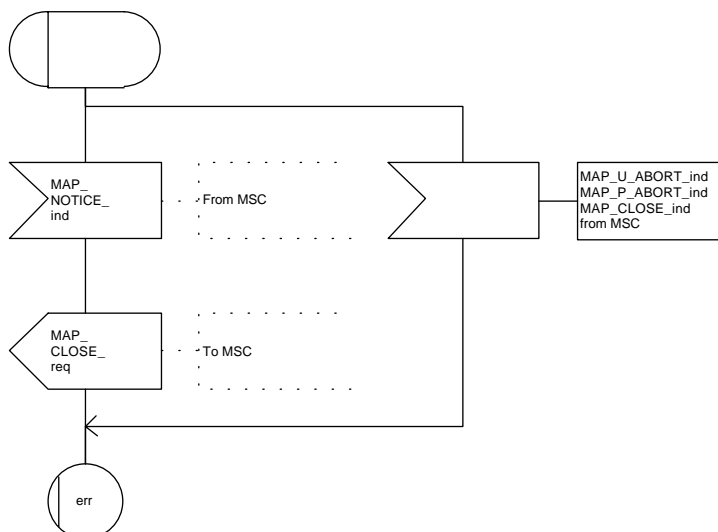
Figure 19.10.2/3
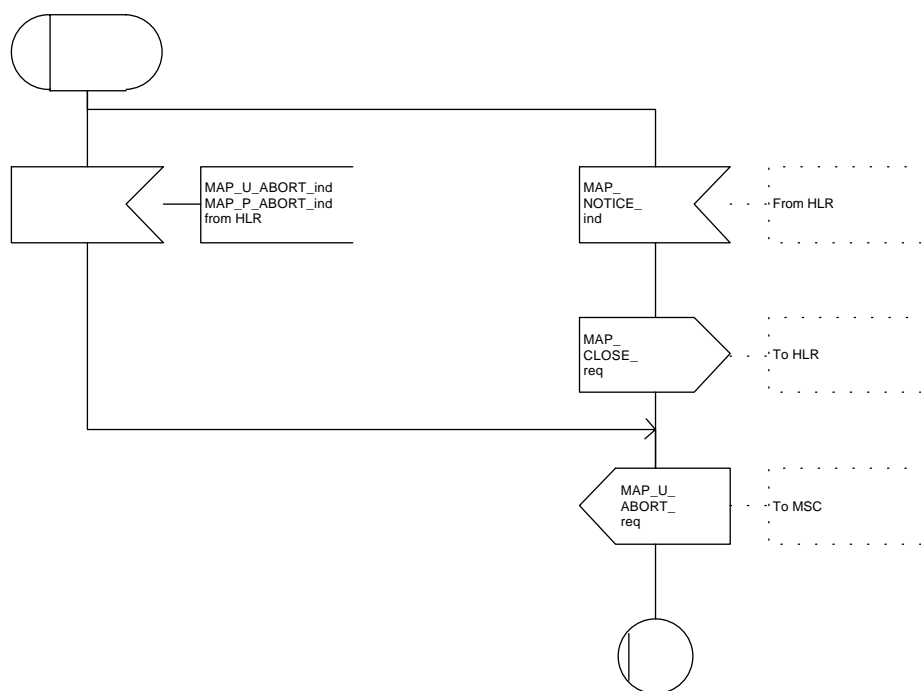
20   Short message service procedures

20.1   General

The short message service procedures are used to control both mobile originated and mobile terminated short message transfer.

Four procedures exist for short message services:

-    mobile originated short message service transfer;

-    mobile terminated short message service transfer;

-    short message alert procedure;

-    short message waiting data set procedure.

The following application context refers to a complex MAP user consisting of several processes:

-    shortMessageGatewayContext.

This application context needs a co-ordinating process in the HLR. Additionally a Co-ordinator has to be defined for the mobile originated situation in the MSC, because the A_CM_SERV_REQ message does not distinguish between mobile originated short message transfer and the short message alert procedures.

20.1.1   Mobile originated short message service Co-ordinator for the MSC

The A_CM_SERV_REQ message (GSM 04.08) is received from the A-interface containing the CM service type. This parameter indicates mobile originated short message service. The service MAP_PROCESS_ACCESS_REQUEST is started.

If the MAP_PROCESS_ACCESS_REQUEST service ends successfully, the MS initiates mobile originated short message transfer or alerting indication. Depending on the situation, the appropriate process is initiated as follows:

-    if the A_RP_MO_DATA indication is received, the process MOSM_MSC is initiated (see subclause 20.2.1);

-    if the A_RP_SM_MEMORY_AVAILABLE indication is received, the process SC_Alert_MSC is initiated (see subclause 20.4.1).

After creation of the user process the Co-ordinator relays the messages between the A-interface and the invoked process until a request or an indication for dialogue termination is received.

The SMS process Co-ordinator is shown in the figure 20.1/1.

Figure 20.1/1
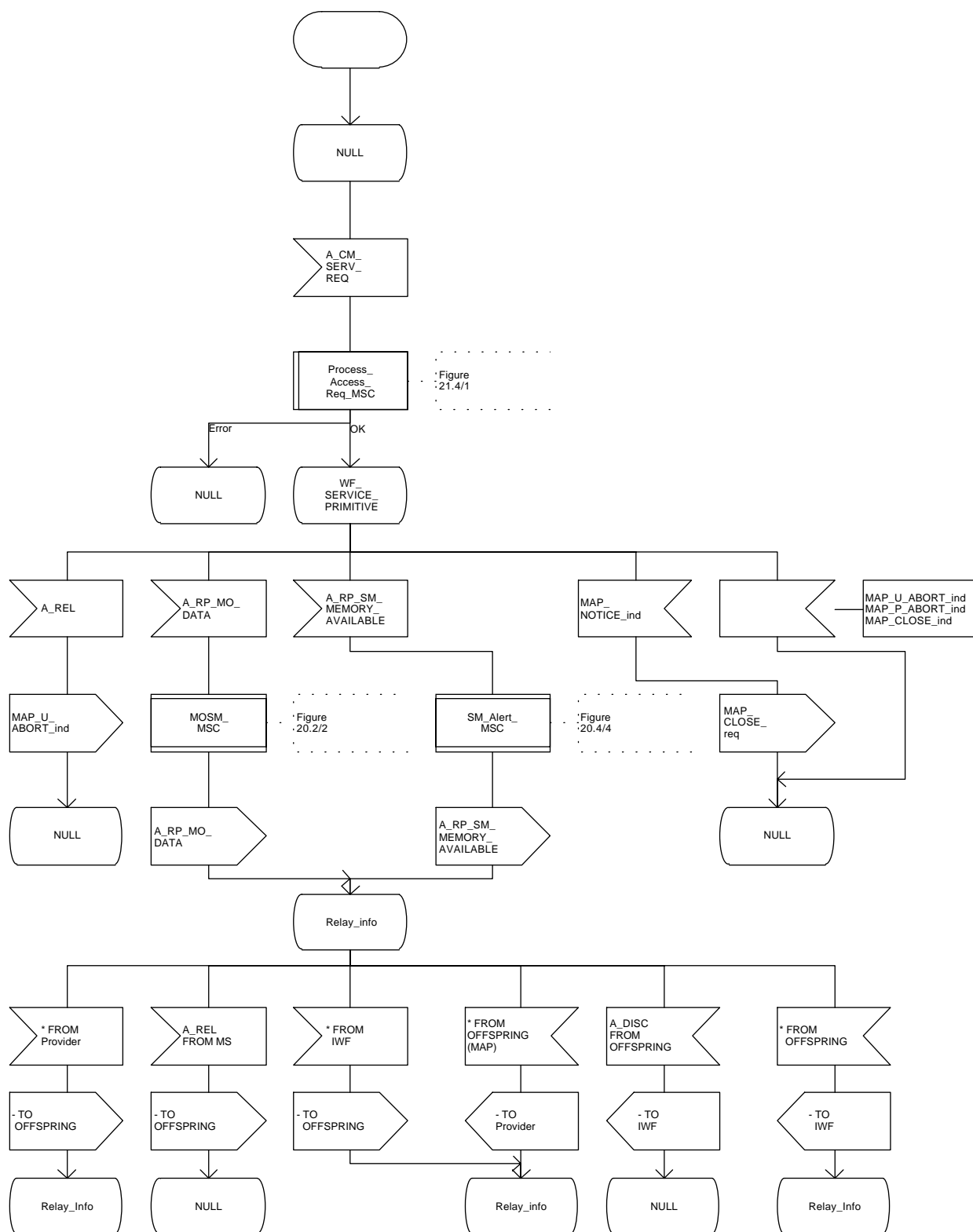
20.1.2 Short message Gateway Co-ordinator for the HLR

The MAP_OPEN indication opens a dialogue for the short message procedure between the gateway MSC and the HLR when the application context shortMessageGatewayContext is received. If that service is successful, the Co-ordinator can receive the first service primitive from the MAP_PM. Depending on the received primitive, the user process is created as follows:

- if the MAP_SEND_ROUTING_INFO_FOR_SM indication is received, the process Mobile_Terminated_MS_HLR is created;

- if the MAP_REPORT_SM_DELIVERY_STATUS indication is received, the process Report_SM_delivery_stat_HLR is created.

After creation of the user processs the Co-ordinator relays the messages between the MAP_PM and the invoked process until a request or an indication for dialogue termination is received.

The SM Gateway Co-ordinator is shown in the figure 20.1/2.

Figure 20.1/2
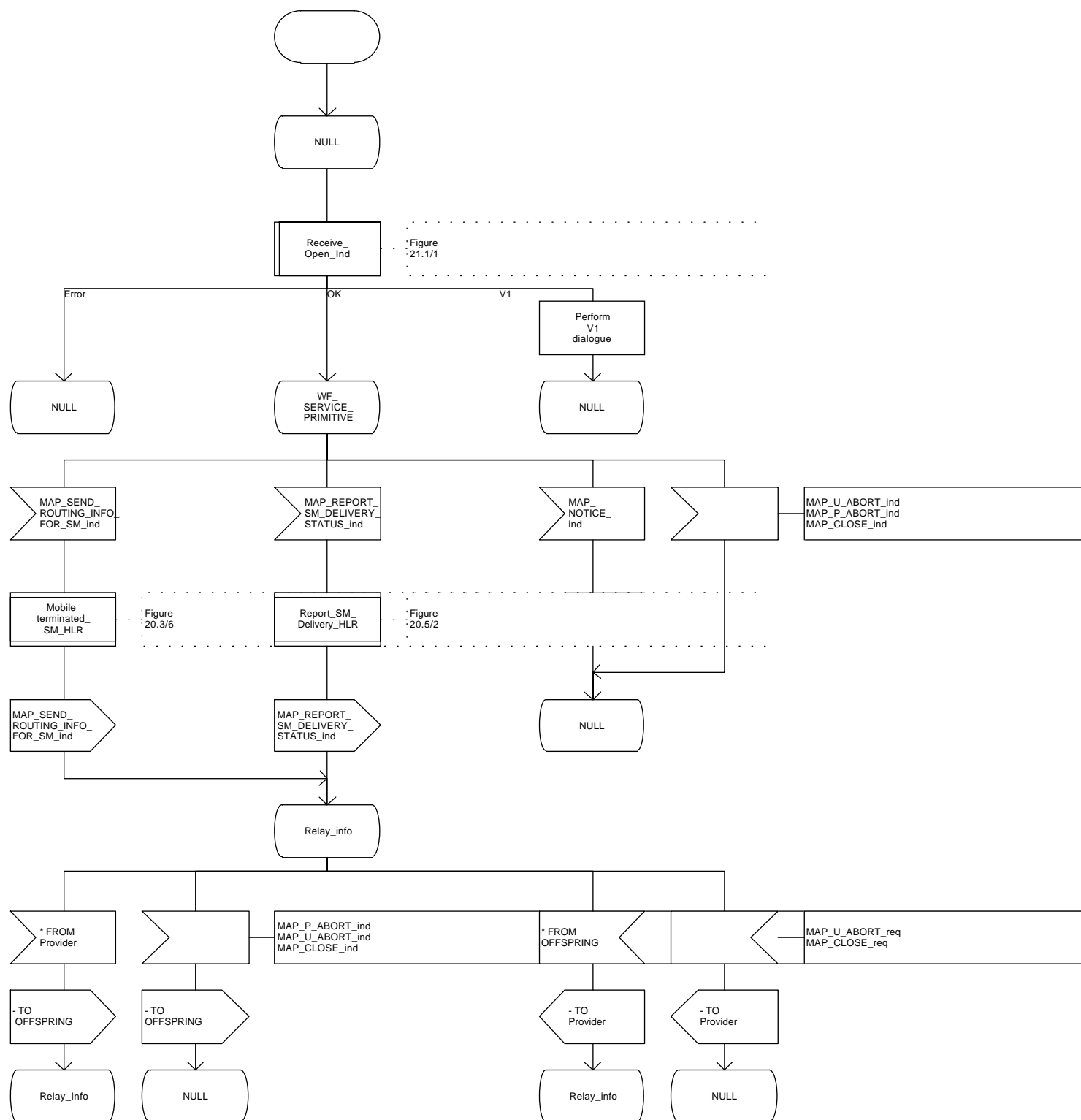
20.2 The mobile originated short message transfer procedure

The mobile originated short message service procedure is used to forward short message from a mobile subscriber to a Service Centre. The mobile originated short message service procedure is shown in figure 20.2/1.

1) Short Message (BSSAP; GSM 04.11)
2) MAP_SEND_INFO_FOR_MO_SMS
3) MAP_SEND_INFO_FOR_MO_SMS_ACK
4) MAP_FORWARD_SHORT_MESSAGE
5) Short message (GSM 03.40)
6) Short message Acknowledgement (GSM 03.40)
7) MAP_FORWARD_SHORT_MESSAGE_ACK
8) Short Message Acknowledgment (BSSAP; GSM 04.11)
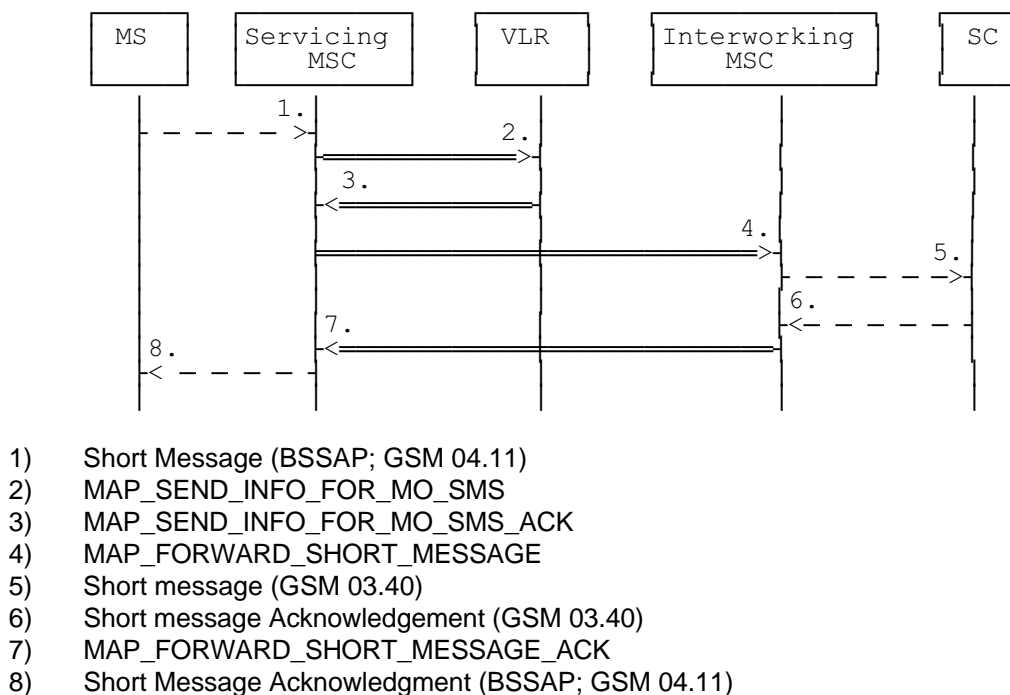
Figure 20.2/1: Mobile originated short message transfer

In addition the following MAP services are used:

MAP_PROCESS_ACCESS_REQUEST   (see subclause 6.3);
MAP_AUTHENTICATE   (see subclause 6.5);
MAP_SET_CIPHERING_MODE   (see subclause 6.6);
MAP_PROVIDE_IMSI   (see subclause 6.9);
MAP_CHECK_IMEI   (see subclause 6.7);
MAP_FORWARD_NEW_TMSI   (see subclause 6.9);
MAP_TRACE_SUBSCRIBER_ACTIVITY   (see subclause 7.1);
MAP_READY_FOR_SM   (see subclause 10.4).

20.2.1   Procedure in the servicing MSC

The activation of the MAP_PROCESS_ACCESS_REQUEST service is described in the subclause 20.6.2.

When receiving the short message from the A-interface, the MSC sends the MAP_SEND_INFO_FOR_MO_SMS request to the VLR. As a response the MSC will receive the MAP_SEND_INFO_FOR_MO_SMS confirmation from VLR indicating that:

- the service ends successfully. If the MSC is not itself the IWMSC, the short message transmission towards the IWMSC is initiated using the MAP_FORWARD_SHORT_MESSAGE request;

- the service ends unsuccessfully. The error cause in the MAP_SEND_INFO_FOR_MO_SMS confirmation indicates the reason for the unsuccessful end. The mapping between MAP error causes and RP_ERROR causes is described in GSM 09.10.

If there are data errors in the MAP_SEND_INFO_FOR_MO_SMS confirmation, or there is an operation failure in MAP, the RP_ERROR cause network out of order is forwarded to the MS.

If the service MAP_FORWARD_SHORT_MESSAGE is started, the MSC will check whether the grouping of MAP_OPEN request and MAP_FORWARD_SHORT_MESSAGE request needs segmentation. If this is the case then the MAP_OPEN request primitive shall be sent first without any associated MAP service request primitive and the dialogue confirmation must be received before the MAP_FORWARD_SHORT_MESSAGE request is sent. As a response to the procedure, the servicing MSC will receive the MAP_FORWARD_SHORT_MESSAGE confirmation from the IWMSC indicating that:

- the short message has been successfully delivered to the Service Centre. The acknowledgement is sent to the MS;

- one of several error cases has occurred. The mapping between MAP error causes and RP_ERROR causes is described in GSM 09.10. The appropriate indication is provided to the MS.

If the procedure failed, a provider error or an abort indication is received. The RP_ERROR cause network out of order is provided to the MS.

If the MSC itself is the interworking MSC, the short message is forwarded to the Service Centre. In that case the service MAP_FORWARD_SHORT_MESSAGE is not initiated. The acknowledge message from the Service Centre is forwarded to the MS (GSM 3.40, GSM 4.11).

The mobile originated short message service procedure is shown in figure 20.2/2.

Figure 20.2/2 (sheet 1 of 3)
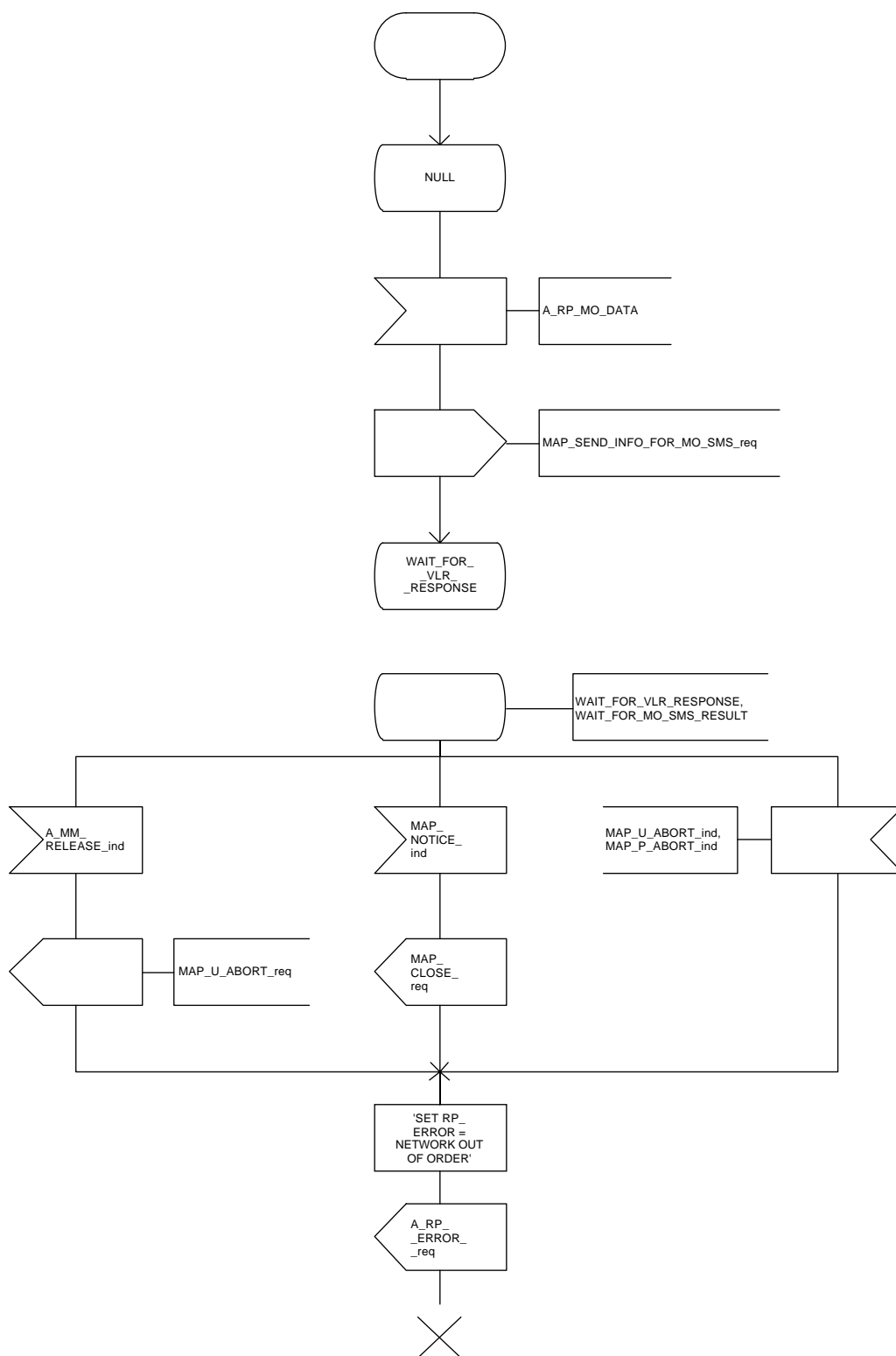
Figure 20.2/2 (sheet 2 of 3)
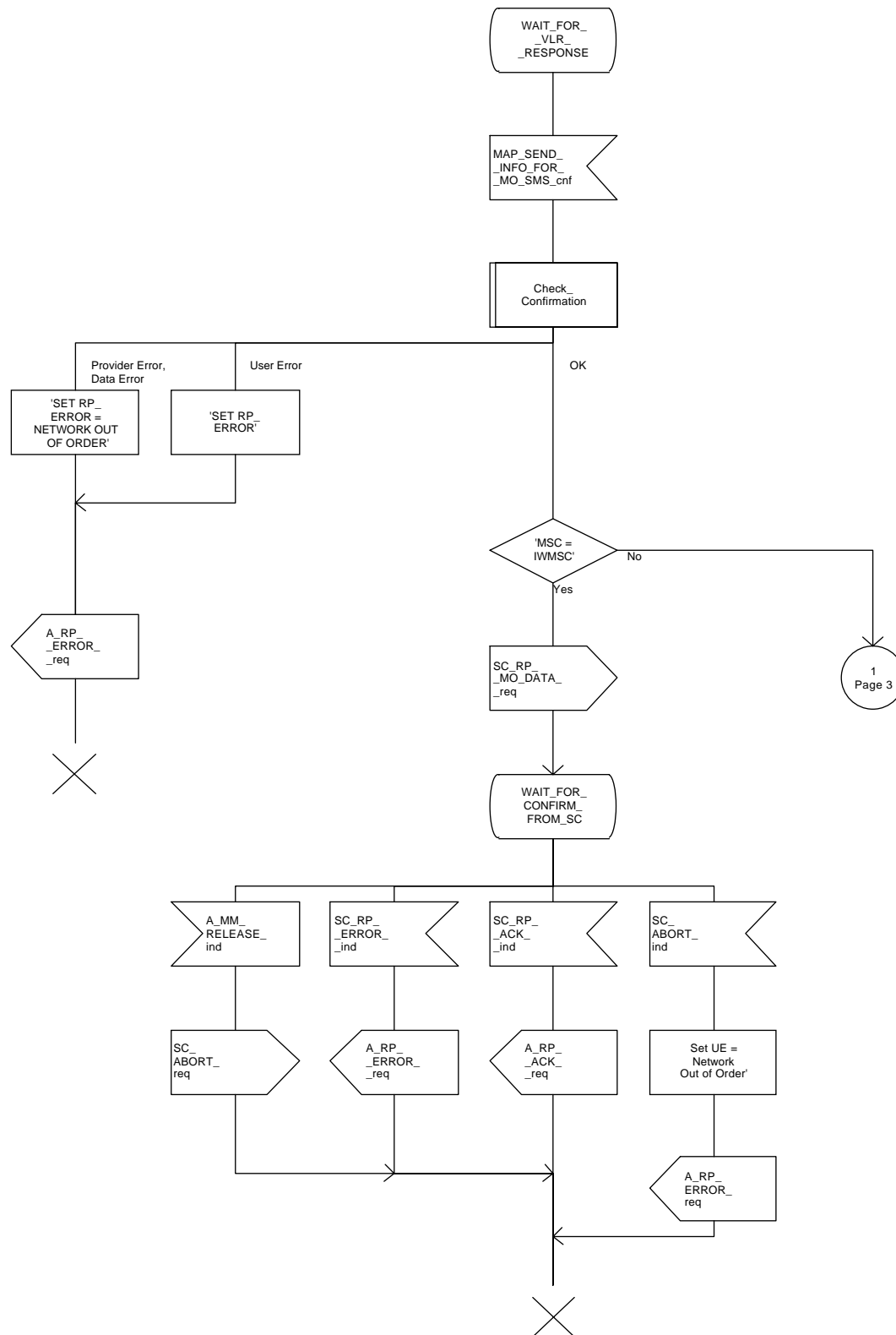
Figure 20.2/2 (sheet 3 of 3)
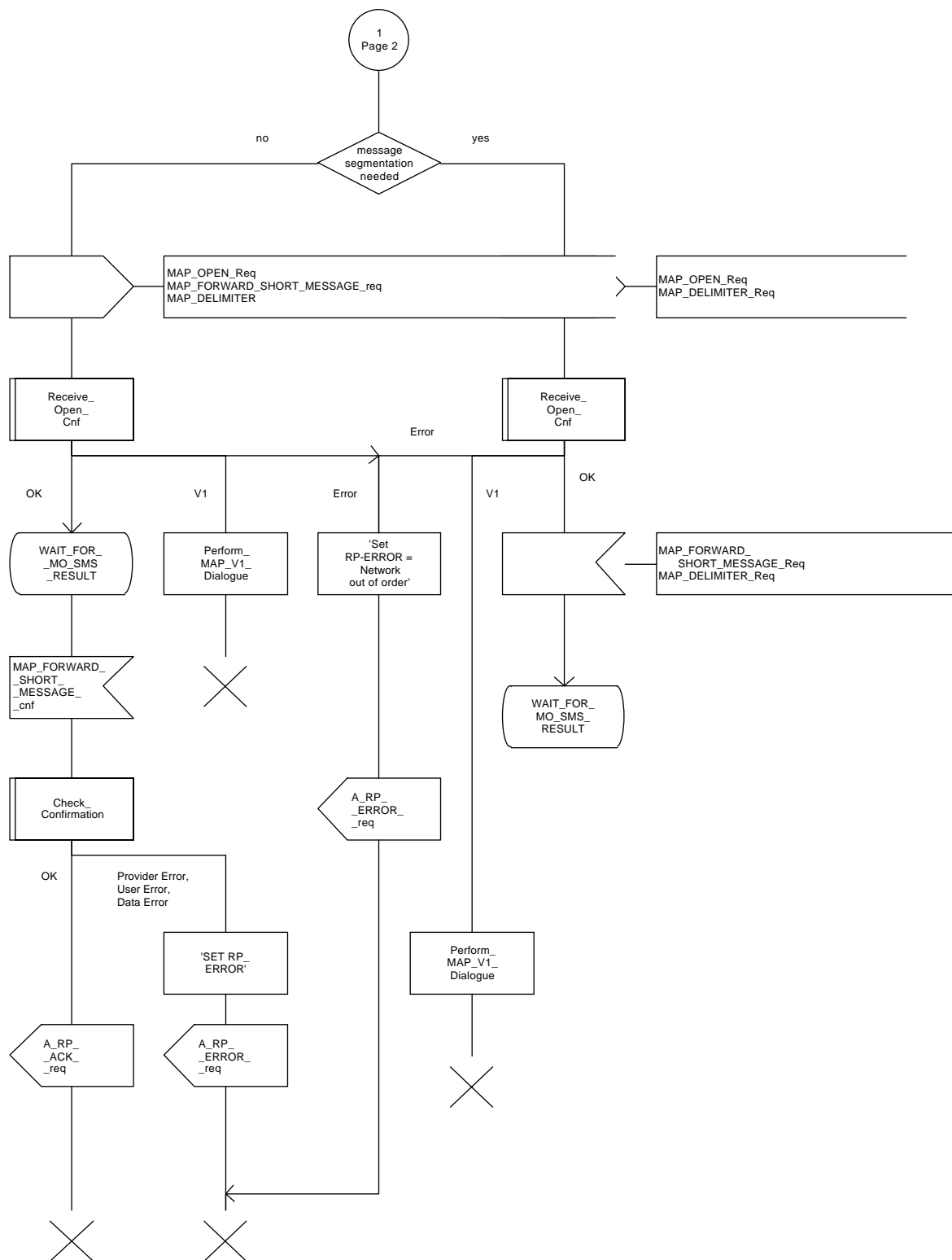
20.2.2 Procedure in the VLR

The MAP_PROCESS_ACCESS_REQUEST indication starts the MAP_PROCESS_ACCESS_REQUEST service in the VLR. The application context in the MAP_OPEN indication is mobile originated short message transfer.

If the service MAP_PROCESS_ACCESS_REQUEST is successful, the VLR waits for the next message from the MSC. When receiving the MAP_SEND_INFO_FOR_MO_SMS indication, the VLR acts as follows:

- if there is incompatibility in the subscription check, the error teleservice not provisioned is returned to the MSC;

- if the short message transfer would contravene operator determined barring, the call barred error with cause operator barring is returned;

- if the short message transfer would contravene the supplementary service call barring conditions in the VLR, the call barred error with cause barring service active is returned.

When the mobile subscriber has passed all checks, the MAP_SEND_INFO_FOR_MO_SMS response is initiated and the procedure is terminated in the VLR. The mobile originated short message transfer procedure in the VLR is shown in figure 20.2/3.

Figure 20.2/3
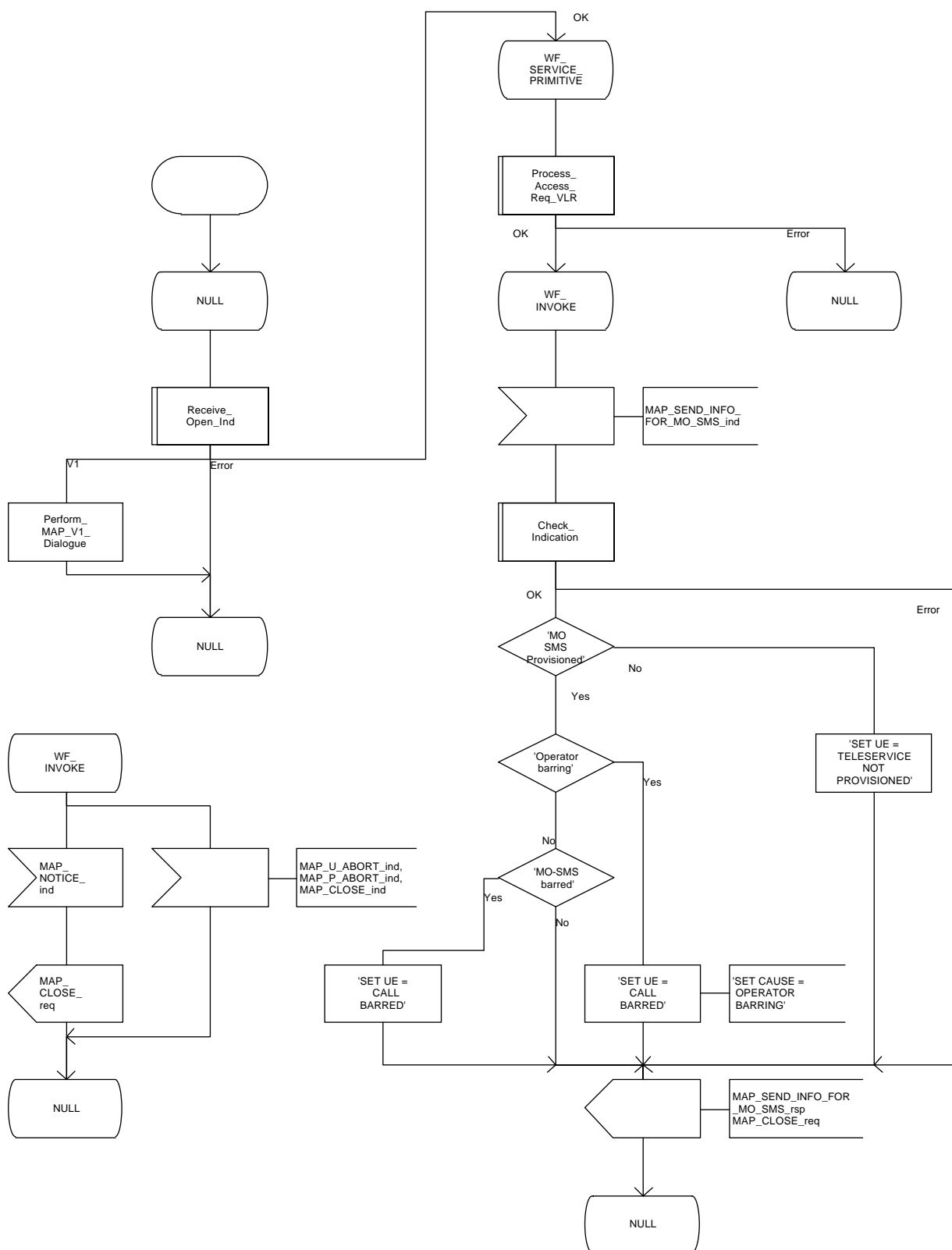
20.2.3 Procedure in the interworking MSC

This procedure applies only when the IWMSC is not the servicing MSC.

When receiving a MAP_OPEN indication primitive that is not associated with any MAP service indication primitive and if the dialogue is accepted, the MAP service-user in the interworking MSC issues a MAP_DELIMITER request primitive in order to trigger the local MAP service-provider to confirm the dialogue. Then a MAP_FORWARD_SHORT_MESSAGE indication shall be received.

When a MAP_FORWARD_SHORT_MESSAGE indication is correctly received, the Interworking MSC invokes forwarding of the short message to the Service Centre. If invalid data content is detected, an unexpected data value error or a data missing error is returned to the servicing MSC.

The outcome of the procedure with the Service Centre is awaited before a MAP_FORWARD_SHORT_MESSAGE response is given back to the servicing MSC:

- if a short message is accepted by the Service Centre, an acknowledgement is sent back to the servicing MSC;

- if the Service Centre is not identified, the SM Delivery Failure error is returned to the servicing MSC;

- if the Service Centre returns an error indication, the SM Delivery Failure error is returned to the servicing MSC with the error cause and any diagnostic information received from the Service Centre;

- if the short message cannot be forwarded to the Service Centre or the procedure towards the Service Centre fails for some reason, a system failure error is sent to the servicing MSC.

The mobile originated short message service transfer in the IWMSC is shown in figure 20.2/4.

Figure 20.2/4
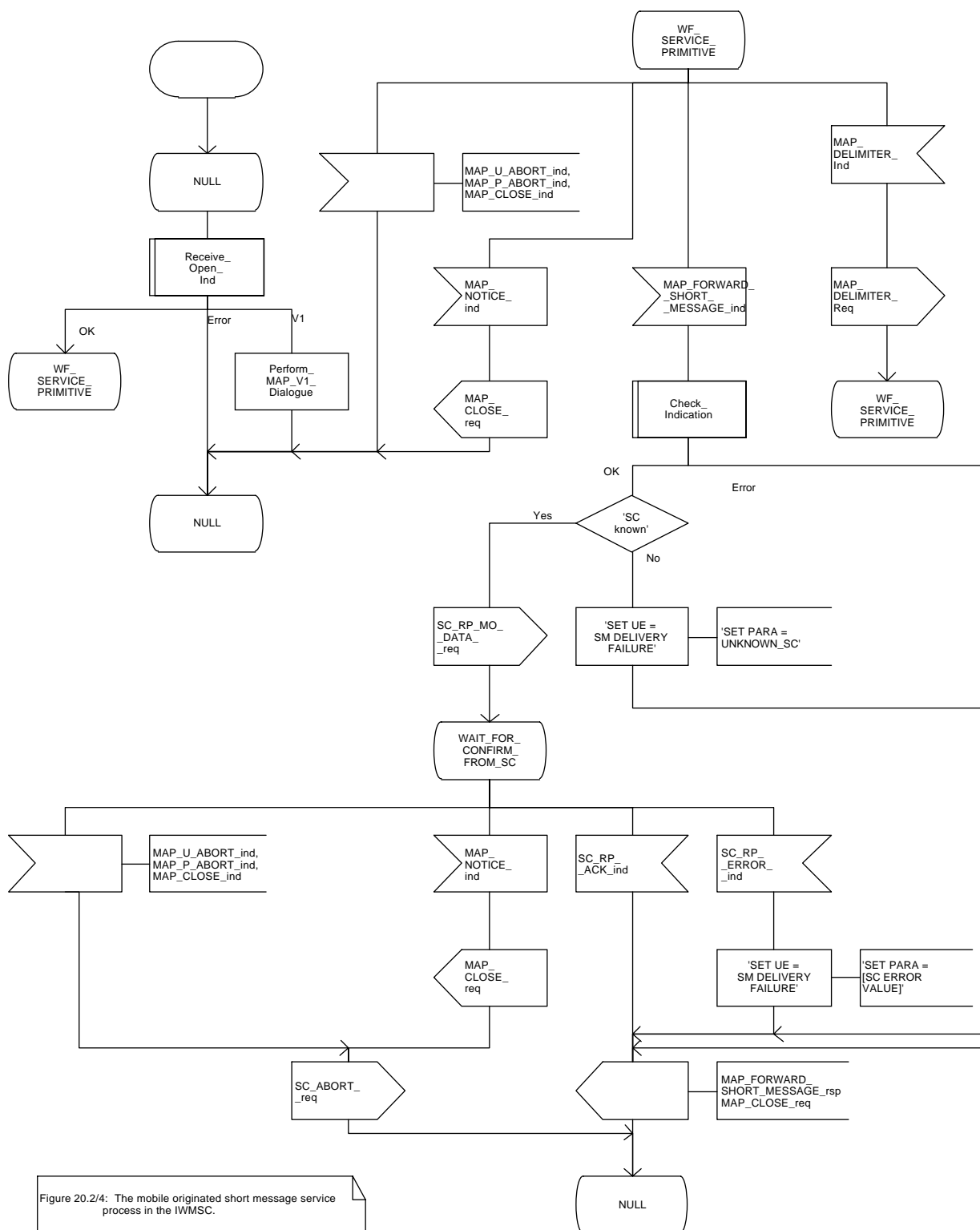
Figure 20.2/4: The mobile originated short message service process in the IWMSC.

20.3 The mobile terminated short message transfer procedure

The mobile terminated short message transfer procedure is used for forwarding a short message or several short messages from a Service Centre to a mobile subscriber. The mobile terminated short message procedure for a single short message transfer is shown in figure 20.3/1.
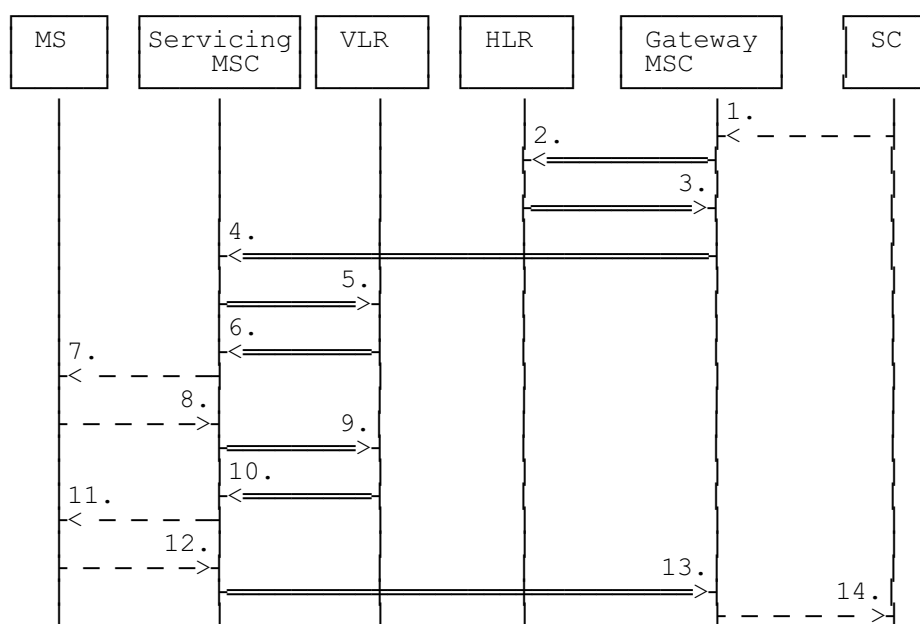
1) Short Message (GSM 03.40)
2) MAP_SEND_ROUTING_INFO_FOR_SM
3) MAP_SEND_ROUTING_INFO_FOR_SM_ACK
4) MAP_FORWARD_SHORT_MESSAGE
5) MAP_SEND_INFO_FOR_MT_SMS
6) MAP_PAGE/MAP_SEARCH_FOR_MOBILE_SUBSCRIBER
7) Page (BSSAP; GSM 04.08)
8) Page response (BSSAP; GSM 04.08)
9) MAP_PROCESS_ACCESS_REQUEST_ACK and
   MAP_SEARCH_FOR_MOBILE_SUBSCRIBER_ACK
10) MAP_SEND_INFO_FOR_MT_SMS_ACK
11) Short Message (BSSAP; GSM 04.11)
12) Short Message Acknowledgement (BSSAP; GSM 04.11)
13) MAP_FORWARD_SHORT_MESSAGE_ACK
14) Short Message Acknowledgment (GSM 03.40)

Figure 20.3/1: Mobile terminated short message service procedures

The mobile terminated short message procedure for multiple short message transfer is shown in figure 20.3/2.
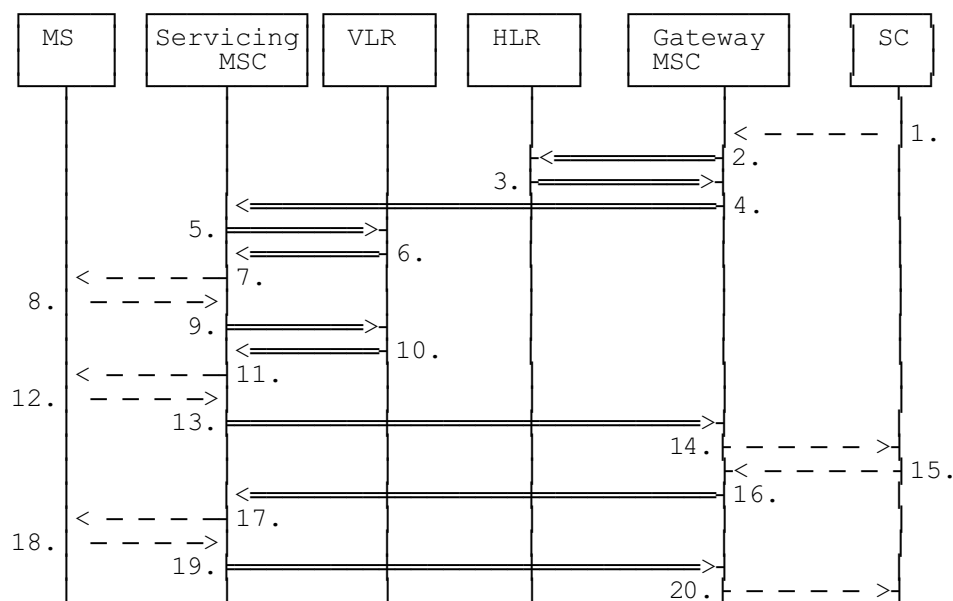
1) Short Message (GSM 03.40)
2) MAP_SEND_ROUTING_INFO_FOR_SM
3) MAP_SEND_ROUTING_INFO_FOR_SM_ACK
4) MAP_FORWARD_SHORT_MESSAGE (note 1)
5) MAP_SEND_INFO_FOR_MT_SMS
6) MAP_PAGE/MAP_SEARCH_FOR_MOBILE_SUBSCRIBER
7) Page (BSSAP; GSM 08.08)
8) Page response (BSSAP; GSM 04.08)
9) MAP_PROCESS_ACCESS_REQUEST_ACK and
   MAP_SEARCH_FOR_MOBILE_SUBSCRIBER_ACK
10) MAP_SEND_INFO_FOR_MT_SMS_ACK
11) Short Message (BSSAP; GSM 04.11)
12) Short Message Acknowledgement (BSSAP; GSM 04.11)
13) MAP_FORWARD_SHORT_MESSAGE_ACK
14) Short Message Acknowledgment (GSM 03.40)
15) Short Message (GSM 03.40)
16) MAP_FORWARD_SHORT_MESSAGE (note 2)
17) Short Message (BSSAP; GSM 04.11)
18) Short Message Acknowledgement (BSSAP; GSM 04.11)
19) MAP_FORWARD_SHORT_MESSAGE_ACK
20) Short Message Acknowledgement (GSM 03.40)

NOTE 1:   The More Messages To Send flag is TRUE.

NOTE 2:   The More Messages To Send flag is FALSE

Figure 20.3/2: Mobile terminated short message procedure for multiple short message transfer.

In the multiple short message transfer the service MAP_FORWARD_SHORT_MESSAGE can be used several times. However, the short message transfer is always acknowledged to the Service Centre before the next short message is sent.

In addition the following MAP services are used:

MAP_PROCESS_ACCESS_REQUEST	(see subclause 6.3);
MAP_PAGE	(see subclause 6.2);
MAP_SEARCH_FOR_MS	(see subclause 6.2);
MAP_AUTHENTICATE	(see subclause 6.5);
MAP_SET_CIPHERING_MODE	(see subclause 6.6);
MAP_CHECK_IMEI	(see subclause 6.7);
MAP_FORWARD_NEW_TMSI	(see subclause 6.9);
MAP_REPORT_SM_DELIVERY_STATUS	(see subclause 10.3);
MAP_INFORM_SERVICE_CENTRE	see subclause 10.6);
MAP_TRACE_SUBSCRIBER_ACTIVITY	(see subclause 7.1);
MAP_READY_FOR_SM	(see subclause 10.4).

20.3.1   Procedure in the Servicing MSC

When initiating the dialogue with the servicing MSC, the SMS Gateway MSC must provide the IMSI of the subscriber to whom the short message is directed.

The IMSI can be included either in the Destination Reference of the MAP_OPEN indication received from the SMS Gateway MSC or in the sm-RP-DA information field of the MAP_FORWARD_SHORT_MESSAGE indication.

When receiving a MAP_OPEN indication primitive that is not associated with any MAP service indication primitive and if the dialogue is accepted, the MAP service-user in the servicing MSC issues a MAP_DELIMITER request primitive in order to trigger the local MAP service-provider to confirm the dialogue.

When receiving the first MAP_FORWARD_SHORT_MESSAGE indication from the gateway MSC, the servicing MSC sends the MAP_SEND_INFO_FOR_MT_SMS request primitive to the VLR, if the MAP service primitive is accepted and if short message service is supported in the servicing MSC.

The MAP_FORWARD_SHORT_MESSAGE indication primitive is checked by the macro "Check_Indication". If the received MAP service primitive contains errors, the service is aborted and an unexpected data value error or data missing error is returned to the GMSC.

If the MSC does not support the short message service, the service is aborted in the servicing MSC and the error "Facility Not Supported" is returned to the GMSC.

The subscriber identity information that may be included in the MAP_OPEN indication primitive and in the MAP service indication primitive is checked by the macro "Check_Subscr_Identity_For_MT_SMS" as follows.

If a Destination Reference has been received in the MAP_OPEN indication, an LMSI must be present in the sm-RP-DA information field of the MAP_FORWARD_SHORT_MESSAGE indication. The LMSI shall be included in the sm-RP-DA information field of the MAP_SEND_INFO_FOR_MT_SMS request sent to the VLR; the associated MAP_OPEN request must contain a Destination Reference that carries an IMSI.

Otherwise, if the IMSI is included in the sm-RP-DA information field of the MAP_FORWARD_SHORT_MESSAGE indication, it is mapped into the sm-RP-DA information field of the MAP_SEND_INFO_FOR_MT_SMS request that is sent to the VLR. In this case, the IMSI is not accompanied by an LMSI and neither the MAP_OPEN indication received from the gateway MSC nor the MAP_OPEN request sent to the VLR shall include a Destination Reference.

If a Destination Reference has been received in the servicing MSC and the sm-RP-DA information field of the MAP_FORWARD_SHORT_MESSAGE indication does not include an LMSI or if no Destination Reference has been received and the sm-RP-DA information field does not cover an IMSI the service is aborted in the servicing MSC and the error "Unexpected Data Value" is returned to the SMS GMSC.

The following responses to the MAP_SEND_INFO_FOR_MT_SMS request may be received from the VLR:

- absent subscriber, unidentified subscriber or system failure error. The error code is forwarded to the GMSC;

- unknown subscriber error. The system failure indication is provided to the GMSC;

- data missing or unexpected data value error. The system failure indication is provided to the GMSC;

- a provider error or an abort indication. The system failure indication is provided to the GMSC;

- paging procedure invocation (see subclause 21.3) reporting the successful outcome of the procedure;

- search procedure invocation (see subclause 21.3) reporting the successful outcome of the procedure.

The result of the paging or the search procedure is processed as follows:

- if the procedure is completed successfully, the MSC will send the MAP_PROCESS_ACCESS_REQUEST request to the VLR (see subclause 21.4);

- if the procedure is completed successfully, but the MS has no mobile terminated short message transfer capability, the procedure is terminated and SM delivery failure indication with cause "equipment not SM equipped" is provided to the GMSC;

- if the procedure ends unsuccessfully, the termination of the procedure is awaited from the VLR. The absent subscriber error is forwarded to the GMSC, but the other error causes are reported as a system failure indication.

If the short message transfer is aborted for any reason, the dialogue with the VLR is aborted. If the procedure with the VLR is aborted by the VLR or by the provider, a system failure indication is provided to the GMSC.

The unsuccessful outcome of the MAP_PROCESS_ACCESS_REQUEST service is reported by using the system failure error to the GMSC.

When the service MAP_PROCESS_ACCESS_REQUEST is carried out, the MSC will receive the MAP_SEND_INFO_FOR_MT_SMS confirmation indicating:

- the unsuccessful outcome of the procedure. The error indication received from the VLR is forwarded to the GMSC;

- the successful outcome of the procedure. The MSC initiates forwarding of the short message to the MS.

If the primitive itself is badly formatted or data is missing, the system failure error is sent to the GMSC.

If forwarding of the short message is initiated, the MSC awaits the result before one of the following responses is sent back to the GMSC:

- an acknowledge if the short message has been successfully delivered to the mobile subscriber;

- an SM delivery failure error containing a parameter indicating either of the following: there is a MS protocol error or the MS memory capacity is exceeded; detailed diagnostic information (see subclause 5.6.1.4) may also be carried;

- a system failure error if the delivery procedure is aborted.

If the More Messages To Send flag was FALSE or the service MAP_FORWARD_SHORT_MESSAGE ends unsuccessfully, the transaction to the gateway MSC is terminated. Otherwise, the servicing MSC waits for the next short message from the Service Centre.

When receiving the next MAP_FORWARD_SHORT_MESSAGE indication from the gateway MSC the servicing MSC will act as follows:

- if the received primitive contains errors, the unexpected data value error or data missing error is provided to the gateway MSC;

- if the More Messages To Send flag is FALSE, the servicing MSC will start the short message transfer procedure to the mobile subscriber. The successful or unsuccessful outcome of this procedure is reported to the gateway MSC and the transaction is terminated.

- if the More Messages To Send flag is TRUE, the servicing MSC will start the short message transfer to the mobile subscriber. If the outcome of this procedure is unsuccessful, the reason is reported to the gateway MSC and the procedure is terminated. If the procedure is successful, it is acknowledged to the gateway MSC and more short messages can be received.

The tracing procedure may be activated. It is described in detail in the clause 17.

The mobile terminated short message transfer procedure in the servicing MSC is shown in figures 20.3/3 and 20.3/4. The page and search procedures are shown in figure 21.3/1.

Figure 20.3/3 (sheet 1 of 3)
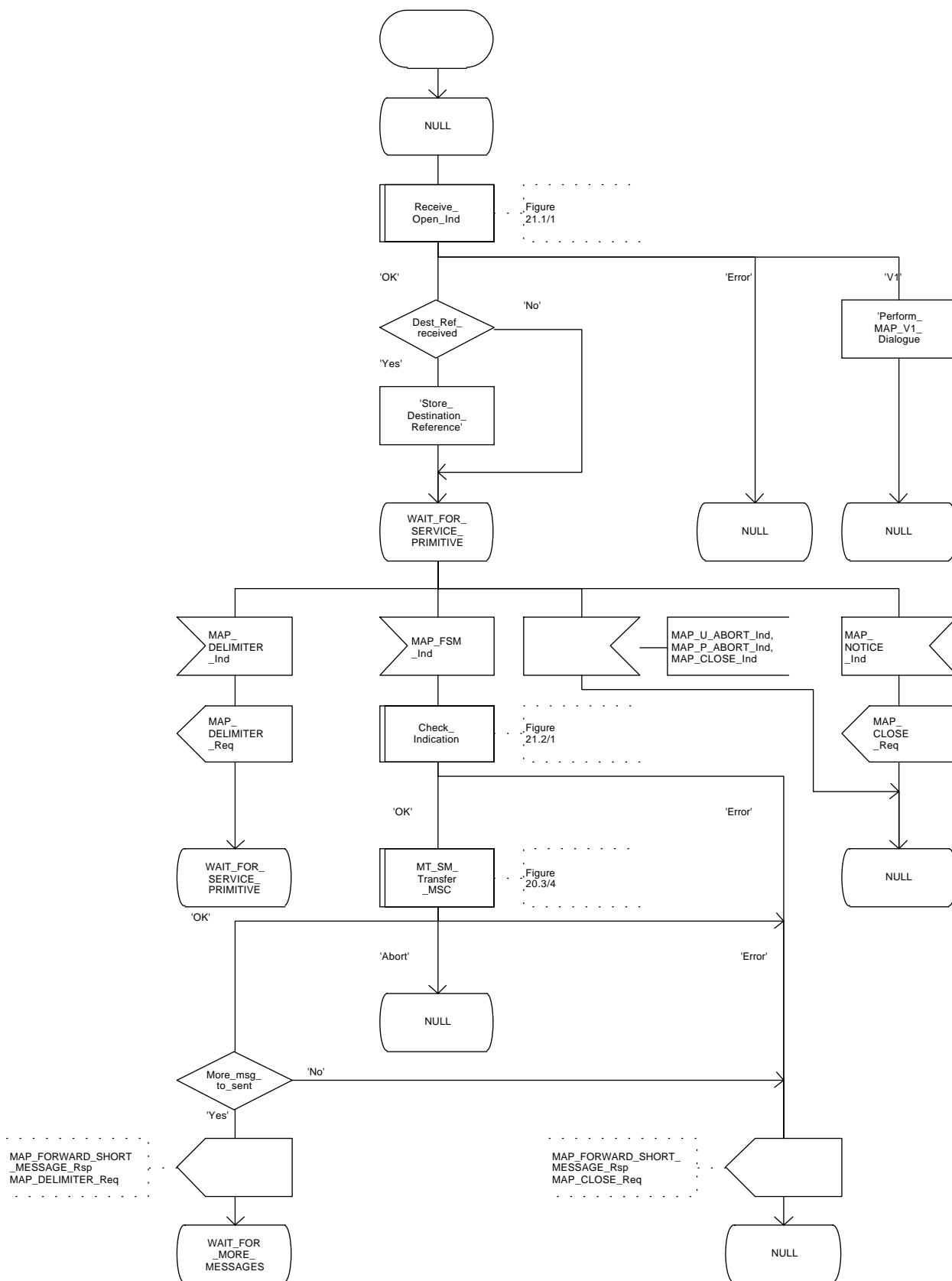
Figure 20.3/3 (sheet 2 of 3)
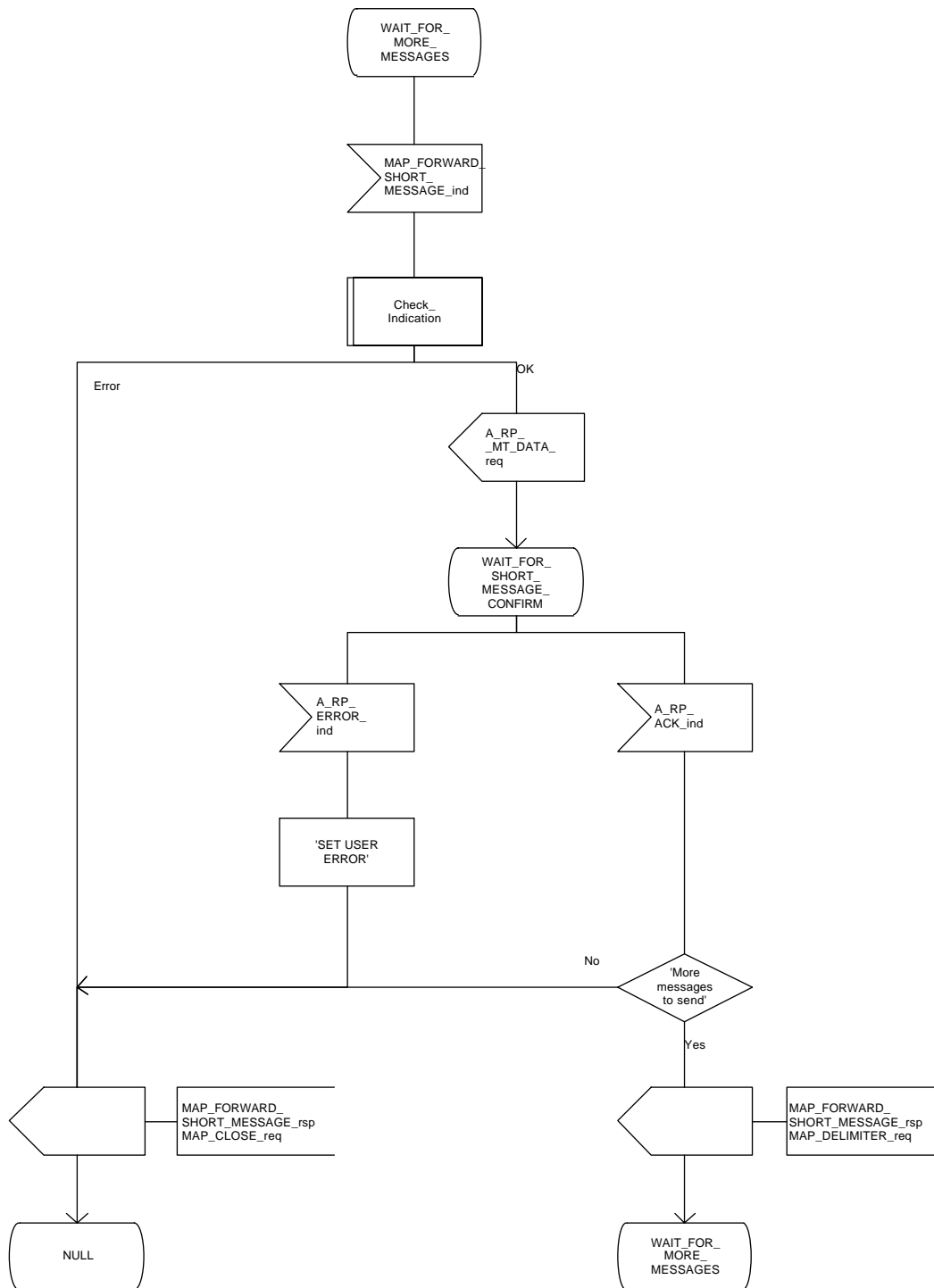
Figure 20.3/3 (sheet 3 of 3)
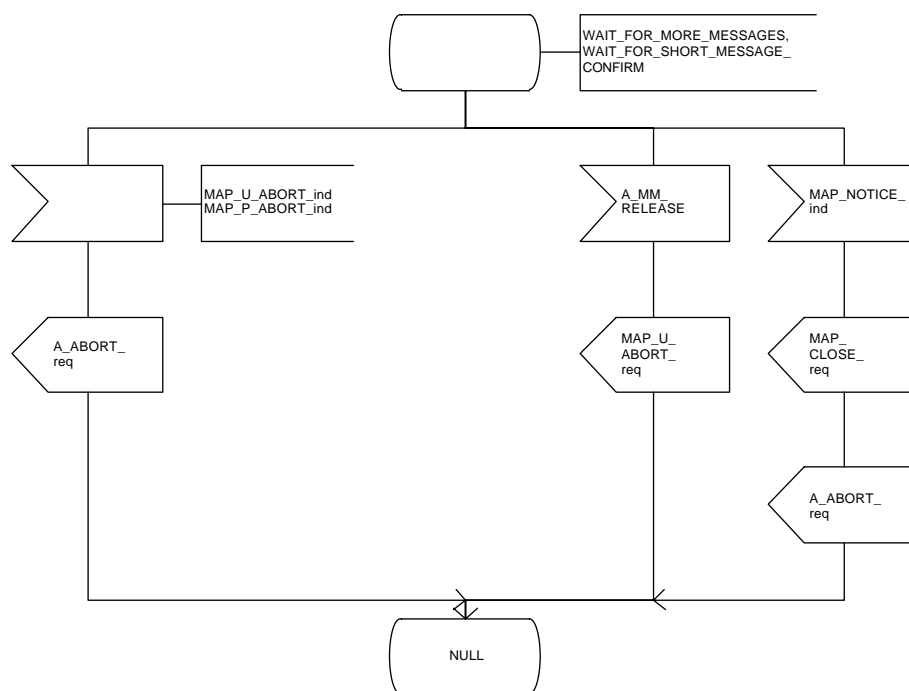
Figure 20.3/4 (sheet 1 of 3)
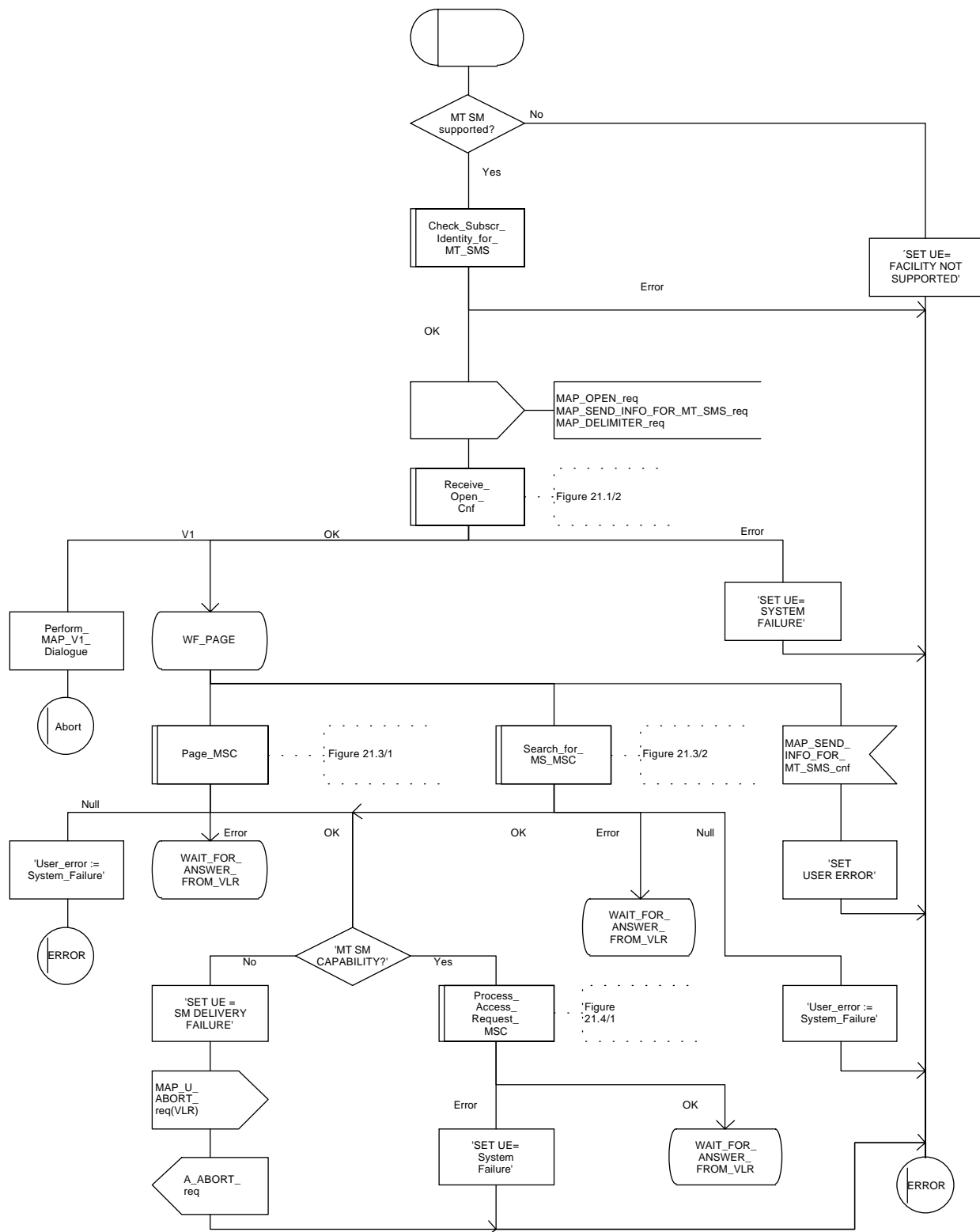
Figure 20.3/4 (sheet 2 of 3)
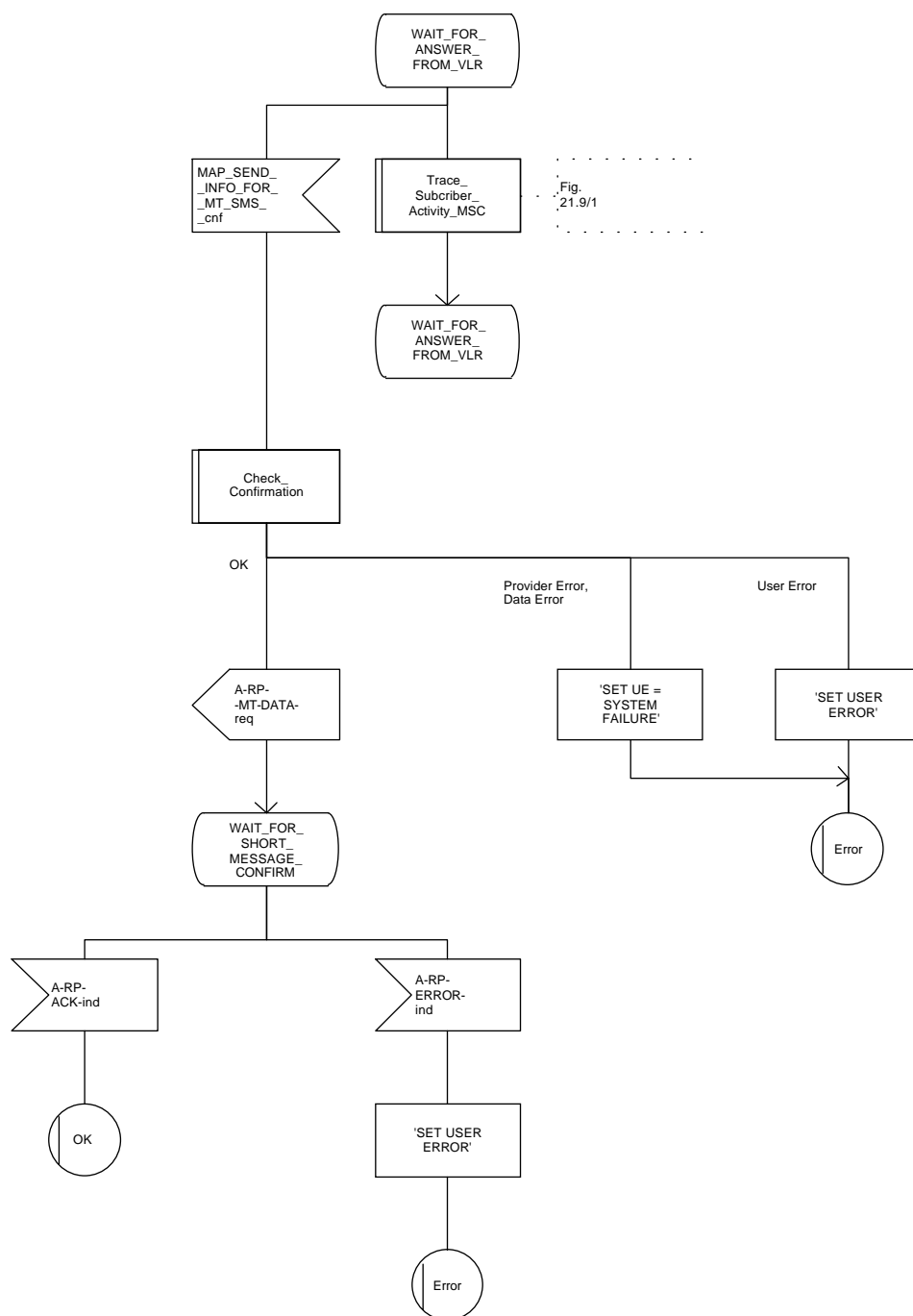
Figure 20.3/4 (sheet 3 of 3)
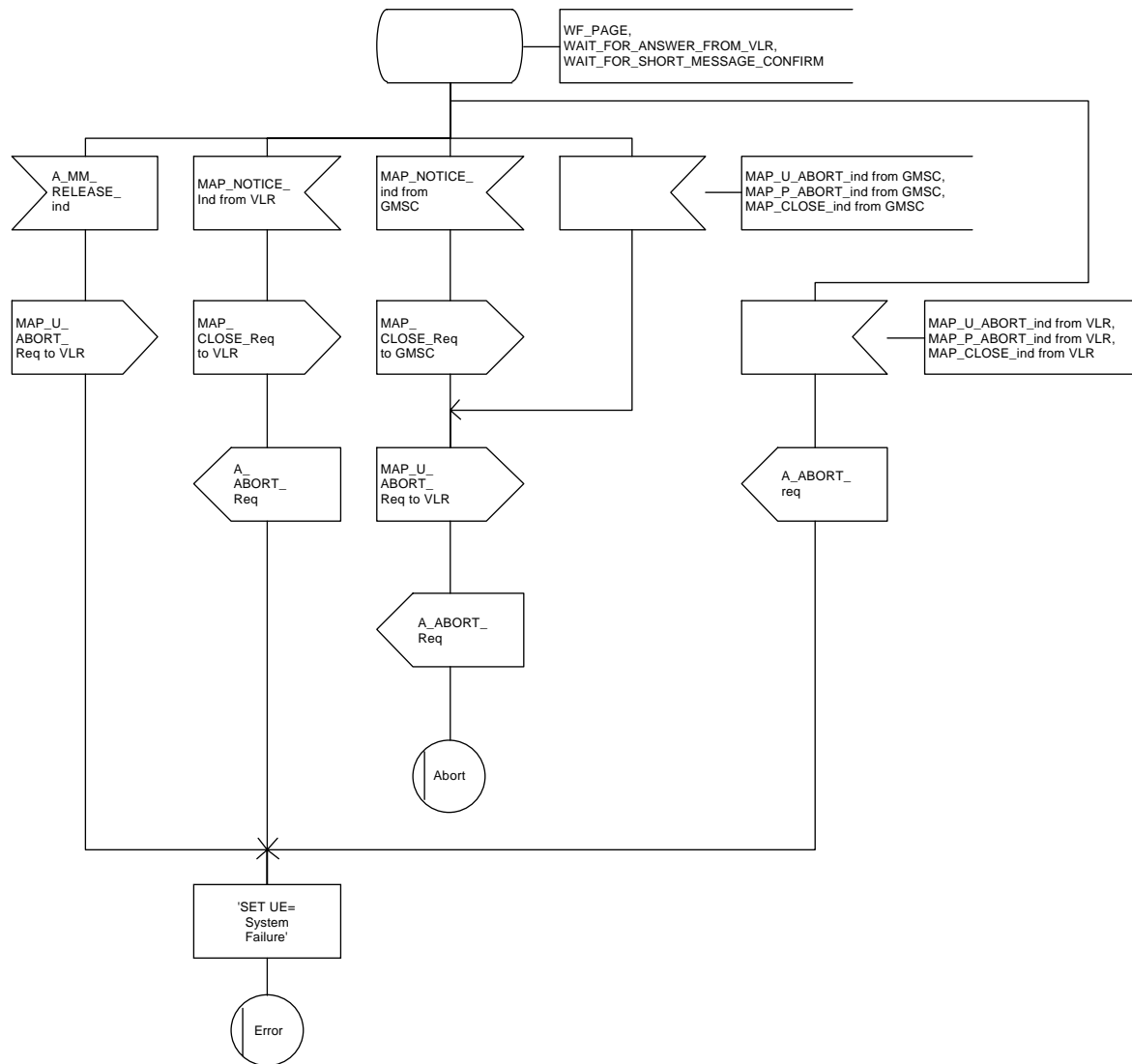
20.3.2 Procedures in the VLR

When receiving the MAP_SEND_INFO_FOR_MT_SMS indication, the VLR will act as follows:

- the parameters and data in the primitive are checked by the macro "Check_Indication". A data failure is reported as an unexpected data value error or a data missing error depending on the nature of the failure;

- for mobile terminated short message the mobile subscriber is identified either by the IMSI only or by the IMSI accompanied by the LMSI. The subscriber identity information that may be included in the MAP_OPEN indication primitive and in the MAP service indication primitive is checked by the macro "Check_Subscr_Identity_For_MT_SMS". In the first case, the IMSI is included in the sm-RP-DA information field and the Destination Reference must not be present in the MAP_OPEN primitive. In the latter case the IMSI must be obtained from the Destination Reference of the MAP_OPEN indication primitive and an LMSI must be present in the sm-RP-DA information field of the MAP_SEND_INFO_FOR_MT_SMS indication. If the mobile subscriber is unknown, the unidentified subscriber error is returned;

- if the "Confirmed by HLR" indicator is set to "Not Confirmed", the unidentified subscriber error is returned;

- if the IMSI Detached Flag is set to detached or the LA Not Allowed Flag is set to not allowed in the VLR, an absent subscriber error is returned and the MS not reachable flag (MNRF) is set;

- if the MAP_SEND_INFO_FOR_MT_SMS indication has passed all the tests, the VLR will initiate the paging procedure. If the location area identification is known and the "Confirmed by Radio Contact" indicator is set to "Confirmed", the MAP_PAGE service is used. Otherwise the MAP_SEARCH_FOR_MOBILE_SUBSCRIBER service is started.

The following responses to the paging procedure may be received from the MSC:

- the MAP_SEARCH_FOR_MOBILE_SUBSCRIBER confirmation indicating a successful outcome, if the search procedure is used. After that the VLR awaits the MAP_PROCESS_ACCESS_REQUEST indication from the MSC;

- the MAP_PAGE confirmation or MAP_SEARCH_FOR_MOBILE_SUBSCRIBER confirmation indicating unsuccessful outcome. If an absent subscriber error is received, the MS not reachable flag (MNRF) is set in the VLR. The errors are forwarded to the MSC in the MAP_SEND_INFO_FOR_MT_SMS response. If the unexpected data value, busy subscriber or unknown location area error is received, the system failure indication is given to the MSC;

- the MAP_PROCESS_ACCESS_REQUEST indication telling that the outcome of the service MAP_PAGE is successful.

If the paging procedure or process access request procedure or any other procedure invoked fails, the appropriate error is reported to the MSC.

If the process access request procedure is successful, the VLR will send the MAP_SEND_INFO_FOR_MT_SMS response to the MSC and the transaction is terminated in the VLR.

The mobile terminated short message transfer procedure in the VLR is shown in figure 20.3/5.

Figure 20.3/5 (sheet 1 of 3)
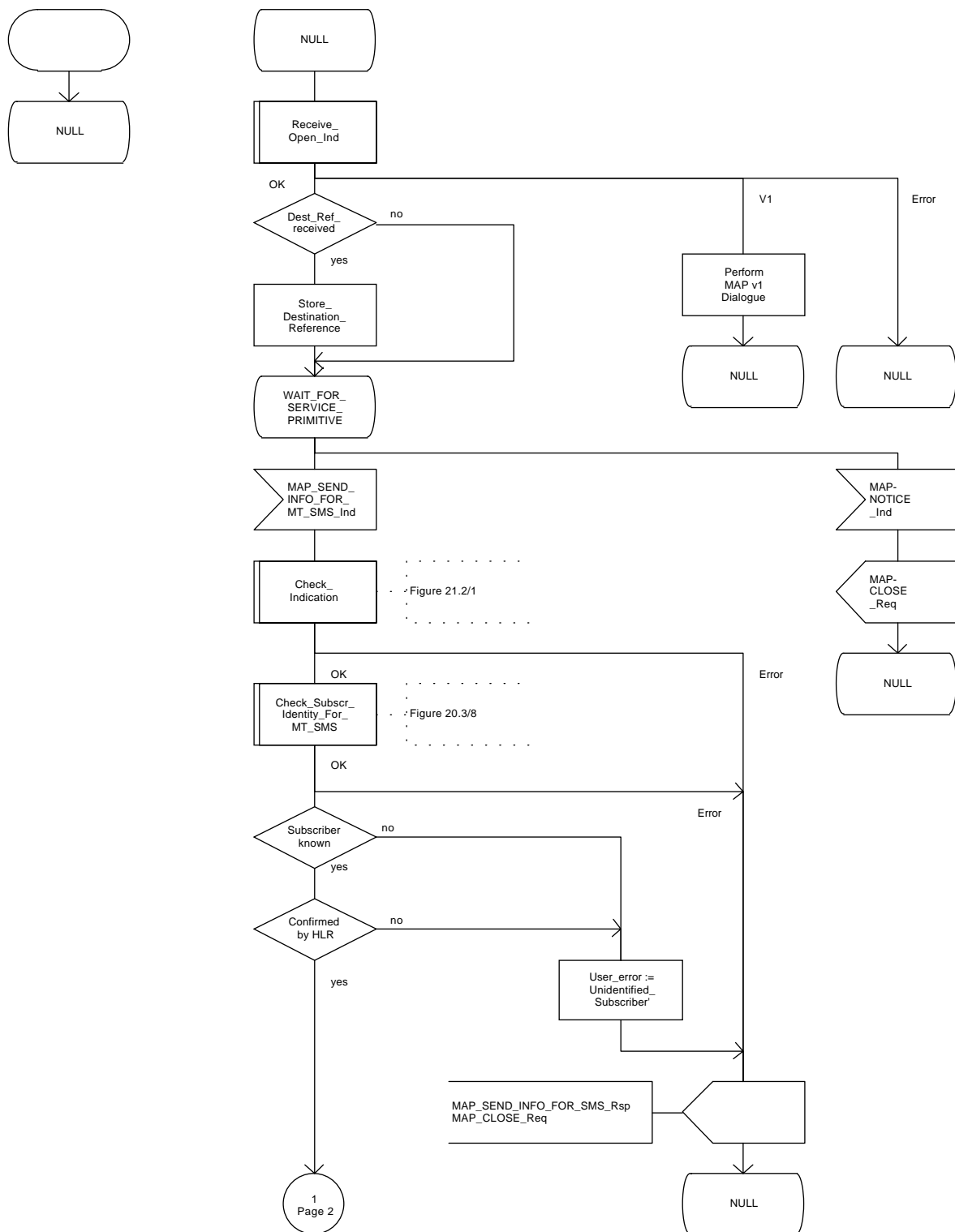
Figure 20.3/5 (sheet 2 of 3)
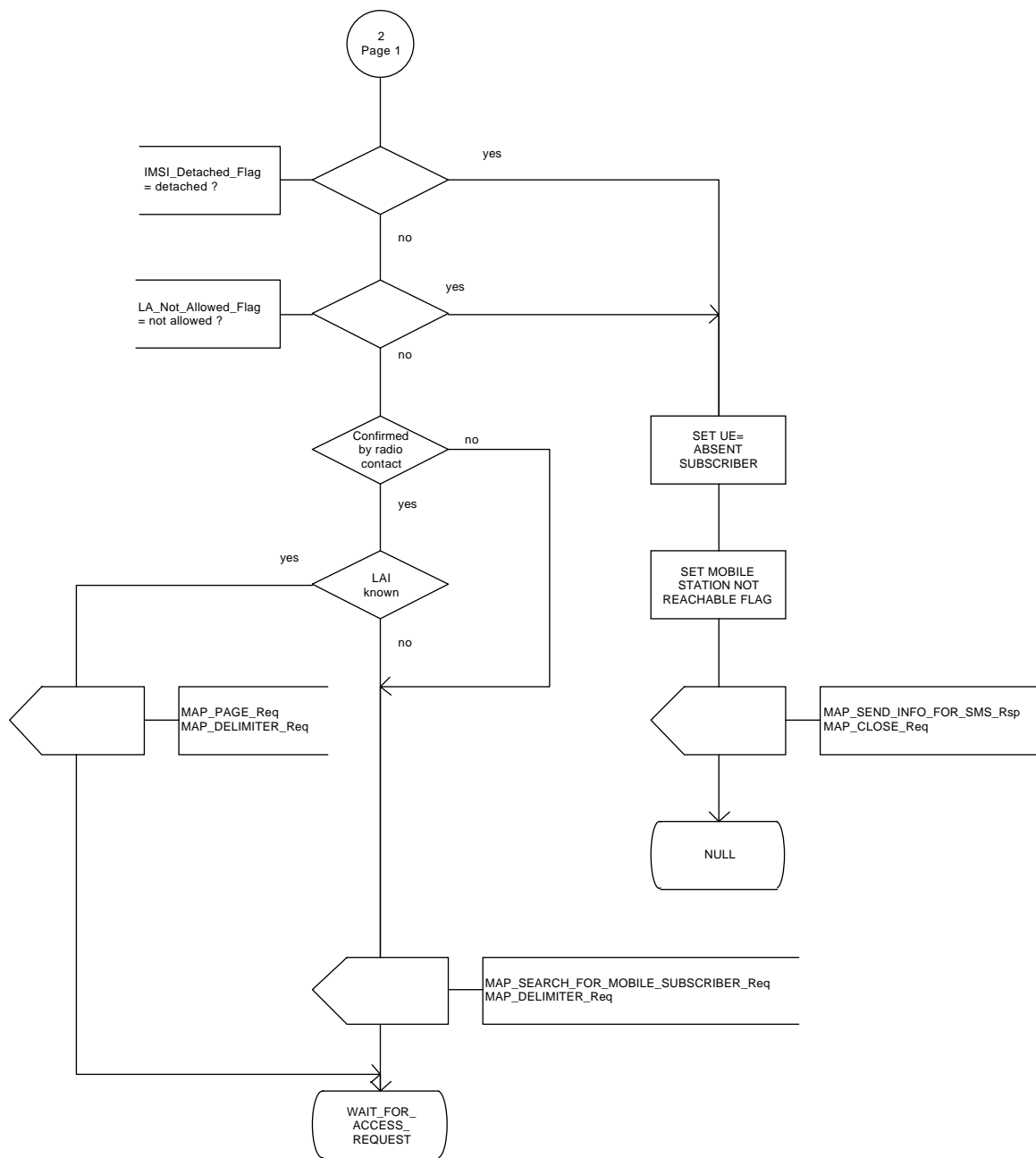
Figure 20.3/5 (sheet 3 of 3)
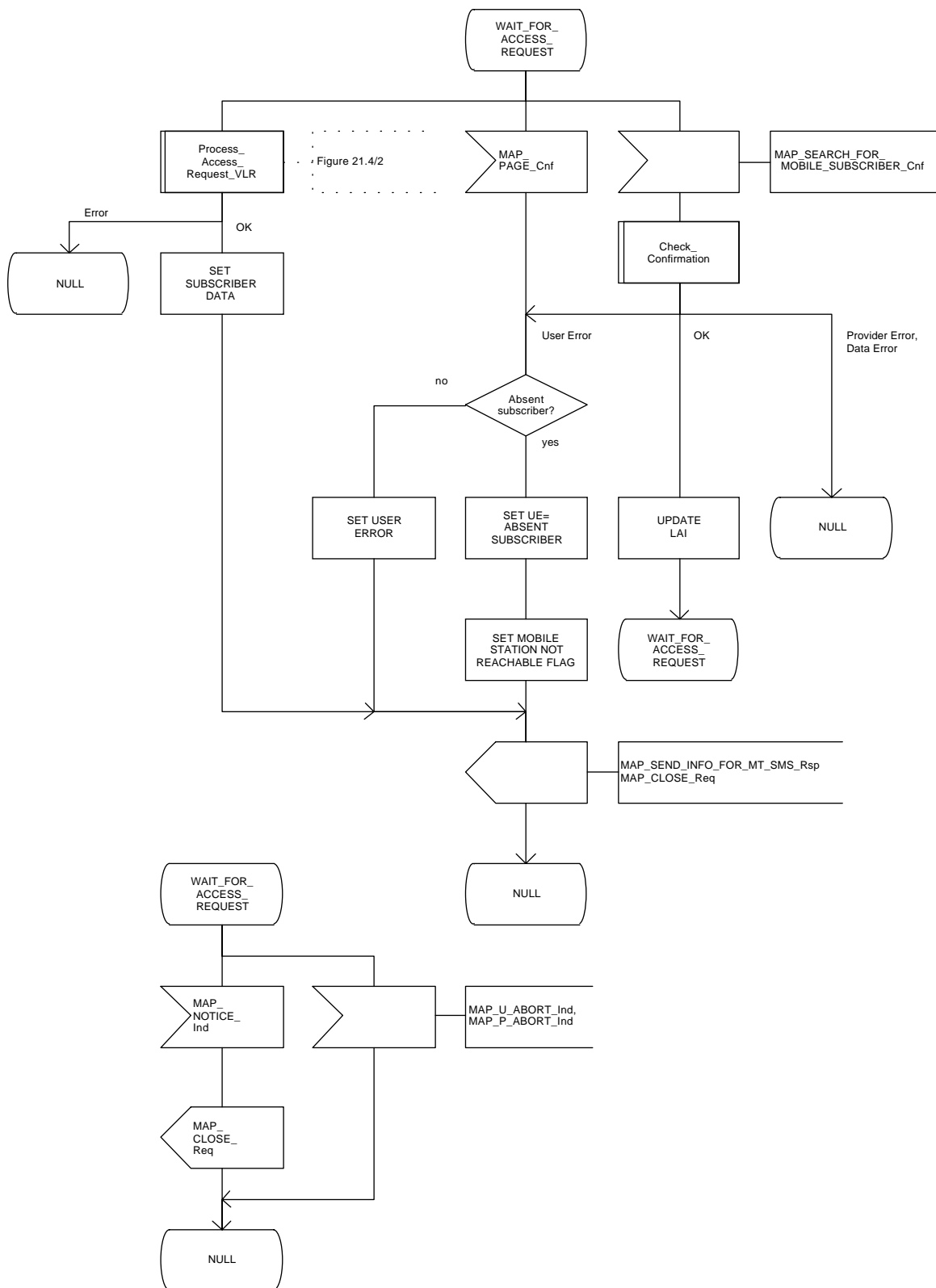
20.3.3   Procedures in the HLR

The MAP_SEND_ROUTING_INFO_FOR_SM indication is received from the GMSC. The following error cases are reported to the GMSC in the MAP_SEND_ROUTING_INFO_FOR_SM response as an unsuccessful outcome of the procedure:

-       if the necessary parameters and data are not present in the primitive or they are badly formatted, the data missing or unexpected data value error is returned;

-       if the mobile subscriber is unknown, i.e. it cannot be identified from the MSISDN given, an unknown subscriber error is returned;

-       if the short message transfer would contravene operator determined barring, the call barred error with cause operator barring is returned;

-       if the short message transfer would contravene the supplementary service barring, the call barred error with cause barring service active is returned;

-       if the mobile subscription identified by the given MSISDN number does not include the short message service, the teleservice not provisioned error is returned;

-       if the location registration of the mobile subscriber shows that the visited PLMN does not support the MT short message service, the facility not supported error is returned;

-       if no MSC identity is stored for the mobile subscriber or the "MSC Area Restricted Flag" is set or the "MS purged" flag is set, i.e. the MS is not reachable, the MSISDN-Alert and the SC address are included in the MWD (if possible) and the "Absent Subscriber" error is returned.

The priority parameter (SM_RP_PRI) is processed as follows:

-       if the priority is low (SM_RP_PRI = False) and the MS not reachable flag (MNRF) is set, an absent subscriber error is returned. The SC-address given in the request will be included in the MWD if possible. The service MAP_INFORM_SERVICE_CENTRE including the parameter MW Status is invoked to indicate whether or not the SC address has been included in the MWD list.

-       if the priority is low (SM_RP_PRI = False), and the MNRF is clear, the routing information is retrieved as described below;

-       if the priority is high (SM_RP_PRI = True) and the MNRF is set, the HLR will send the acknowledge primitive containing the routing information to the gateway MSC. In addition the service MAP_INFORM_SERVICE_CENTRE including the parameter MW Status is invoked to indicate whether or not the SC address is already included in the MWD list.

If the MSISDN-Alert number of the mobile subscriber stored in the MWD is not the same as that received in the MAP_SEND_ROUTING_INFO_FOR_SM indication, the HLR will include in the MAP_INFORM_SERVICE_CENTRE request to the GMSC the MSISDN-Alert number stored.

The MAP_INFORM_SERVICE_CENTRE request is sent also when the MCEF and/or MNRF are set but the routing information is still sent to the GMSC. The status of the flags is indicated in the parameter MW Status.

The routing information is included in a MAP_SEND_ROUTING_INFO_FOR_SM response as follows:

-       the IMSI will be returned to the GMSC together with the MSC number and may be optionally accompanied by the LMSI.

The mobile terminated short message transfer procedure in the HLR is shown in figure 20.3/6.

Figure 20.3/6 (sheet 1 of 3)

Process Mobile_terminated_SM_HLR   203_62(3)

Figure 20.3/6: The mobile terminated short message service process in the HLR
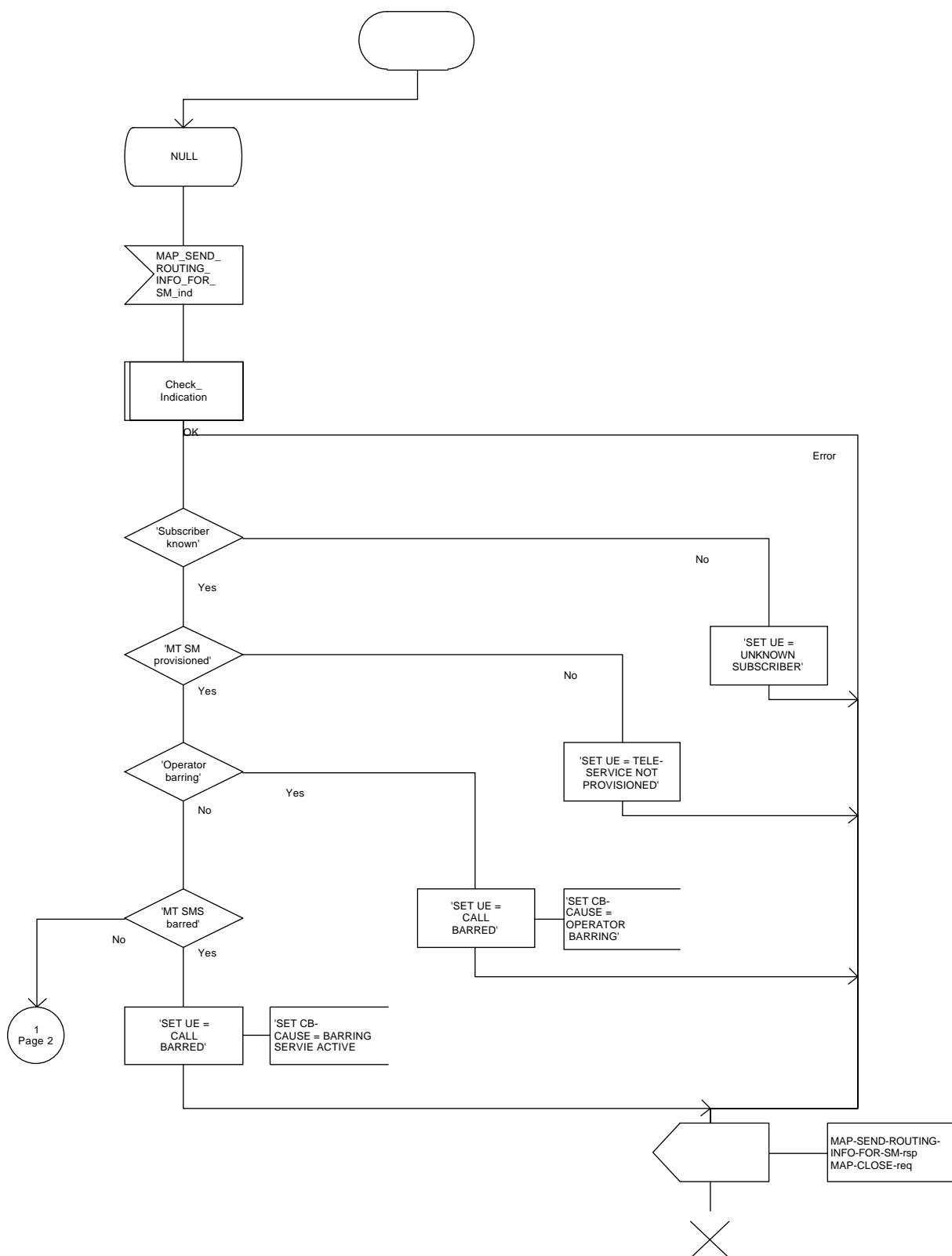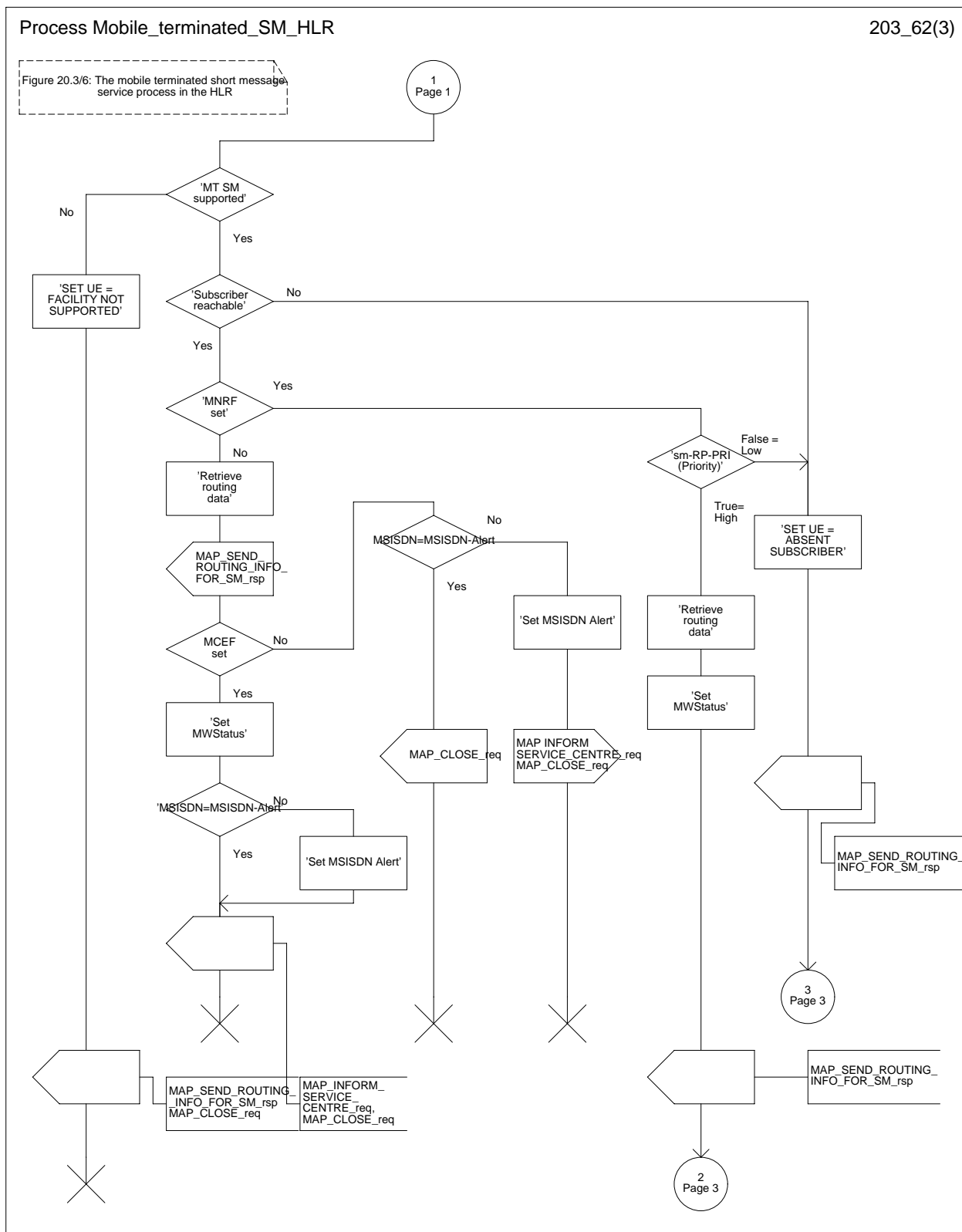
Figure 20.3/6 (sheet 3 of 3)
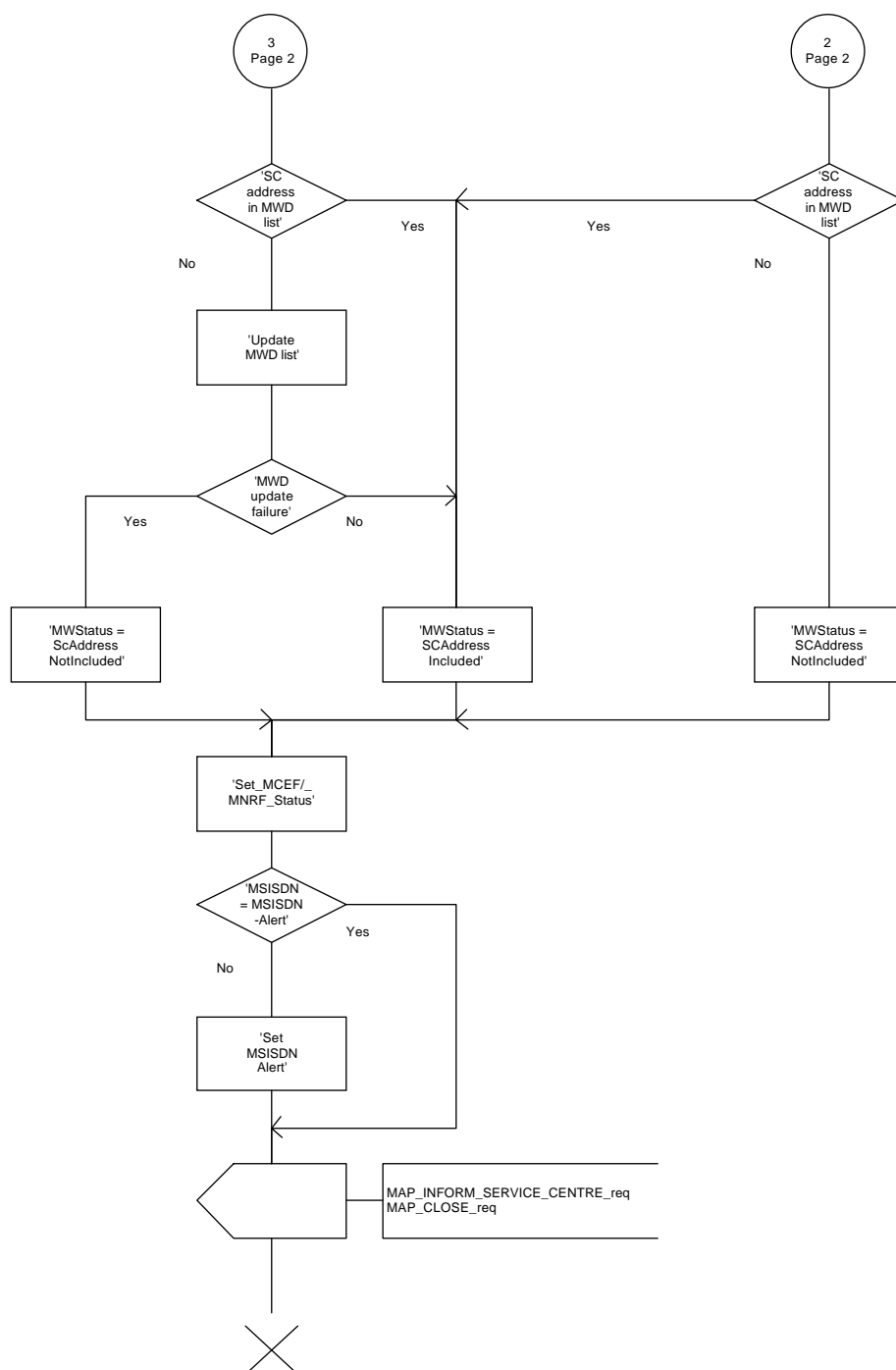
20.3.4 Procedures in the gateway MSC

The short message handling function of the GMSC will request routing information when a mobile terminated short message is received from a Service Centre. The GMSC sends the MAP_SEND_ROUTING_INFO_FOR_SM request to the HLR containing the subscriber data of the mobile subscriber.

As an outcome of the procedure the MAP_SEND_ROUTING_INFO_FOR_SM confirmation is received indicating:

- an unsuccessful event indication containing an error;
  The mapping between the MAP error causes and the RP_ERROR causes is explained in GSM 03.40.

- a successful event indication containing following parameters:
  - an IMSI optionally accompanied by an LMSI; and
  - a routing address (a servicing MSC address).

The GMSC may also receive a MAP_INFORM_SERVICE_CENTRE indication after the MAP_SEND_ROUTING_INFO_FOR_SM confirmation. The parameter MW Status in the message indicates whether or not the Service Centre address is stored in the Message Waiting Data. It also indicates the status of the MCEF and MNRF flags in the HLR.

If the MSISDN-Alert stored in the MWD data is not the same as the one sent to the HLR, the MSISDN-Alert is received in the MAP_INFORM_SERVICE_CENTRE indication. This MSISDN number shall be transferred in a delivery failure report to the SC.

In the abnormal end or in the provider error case the system failure error is provided to the SC.

The forward short message procedure is initiated when the GMSC has obtained the routing information needed to forward a mobile terminated short message to the servicing MSC. If an LMSI has been provided in the MAP_SEND_ROUTING_INFO_FOR_SM confirmation, it can be included in the sm-RP-DA information field of the first MAP_FORWARD_SHORT_MESSAGE request sent to the servicing MSC. In this case, the IMSI must be included in the Destination Reference of the MAP_OPEN request. If the LMSI is not sent by the SMS Gateway MSC, the sm-RP-DA information field in the first MAP_FORWARD_SHORT_MESSAGE request sent to the servicing MSC shall contain the IMSI and the Destination Reference in the MAP_OPEN request shall not be present. The Service Centre address is sent in the parameter SM_RP_OA. The More Messages To Send flag is set to TRUE or FALSE depending on the information received from the Service Centre.

If the GMSC is the servicing MSC then the MAP service is not initiated. The procedure in the Servicing MSC is described in subclause 20.3.1 and in the figure 20.3/4.

If the grouping of MAP_OPEN request and MAP_FORWARD_SHORT_MESSAGE request together would need segmenting, these primitives must not be grouped together. The MAP_OPEN request primitive is sent first without any associated MAP service request primitive and the dialogue confirmation must be received before the MAP_FORWARD_SHORT_MESSAGE request is sent.

As a response to the procedure, the GMSC will receive the MAP_FORWARD_SHORT_MESSAGE confirmation indicating:

- a successful forwarding of the short message. This indication is passed to the SC;

- unsuccessful forwarding of the short message. The mapping of the MAP error causes to the RP_ERROR causes is explained in GSM 09.10 and GSM 04.11. The appropriate error indication is sent to the SC.

A provider error is indicated as a system failure error to the SC.

The GMSC invokes the procedure MAP_REPORT_SM_DELIVERY_STATUS, if an absent subscriber, an unidentified subscriber or SM delivery failure with error cause MS memory capacity exceeded indication is received from the servicing MSC, and the corresponding flags received in the MAP_INFORM_SC are not already set or the SC address is not yet included in the MWD set. The gateway MSC may also invoke the procedure when the transfer was successful, if the MNRF and/or MCEF flags were set in the HLR. This procedure is described in detail in subclause 20.5.

Unexpected data value, system failure and unidentified subscriber errors are indicated as a system failure to the SC. Other errors are indicated using appropriate cause values and diagnostic information between the GMSC and the SC as described in GSM 03.40 and GSM 04.11.

If there are more short messages to send in the Service Centre and the previous short message transfer succeeded, then the gateway MSC awaits the next short message.

When receiving the next short message from the SC, the gateway MSC sets the More Messages To Send flag according to the information received and starts the service MAP_FORWARD_SHORT_MESSAGE again.

If the gateway MSC is the servicing MSC, then the short message transfer to mobile subscriber is started as described in the subclause 20.3.1.

The mobile terminated short message transfer procedure in the gateway MSC is shown in figure 20.3/7.

Figure 20.3/7 (sheet 1 of 5)
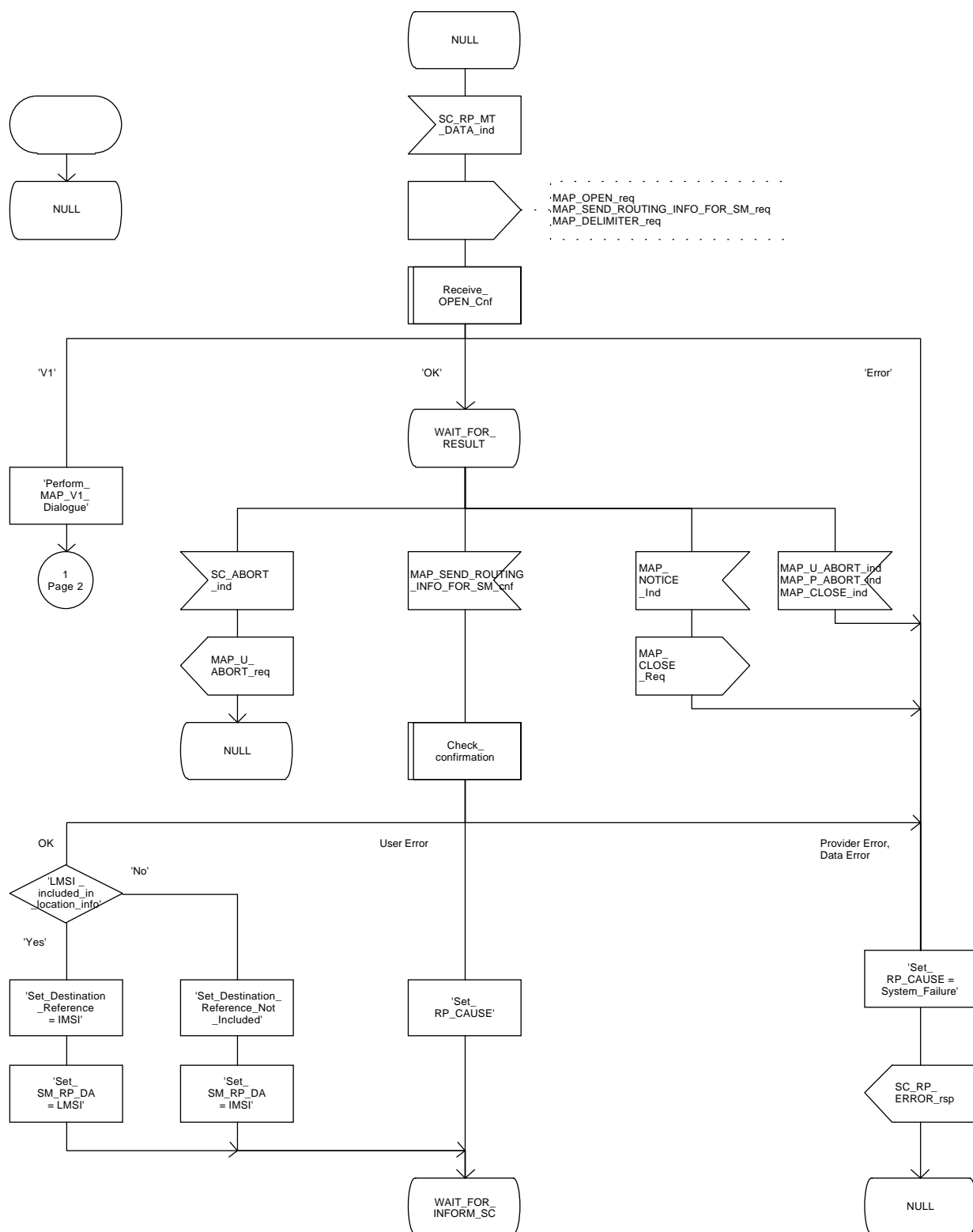
Figure 20.3/7 (sheet 2 of 5)
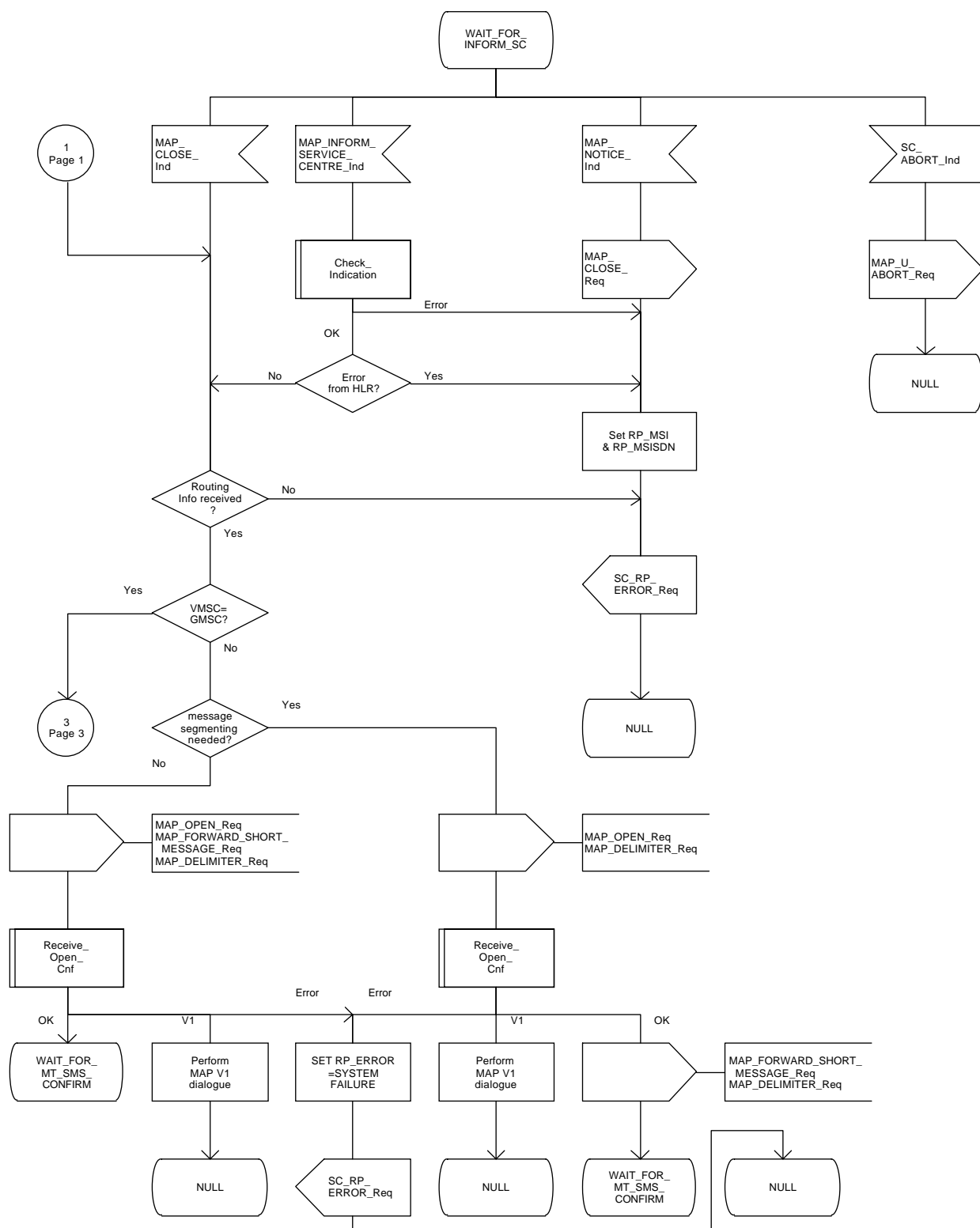
Figure 20.3/7 (sheet 3 of 5)
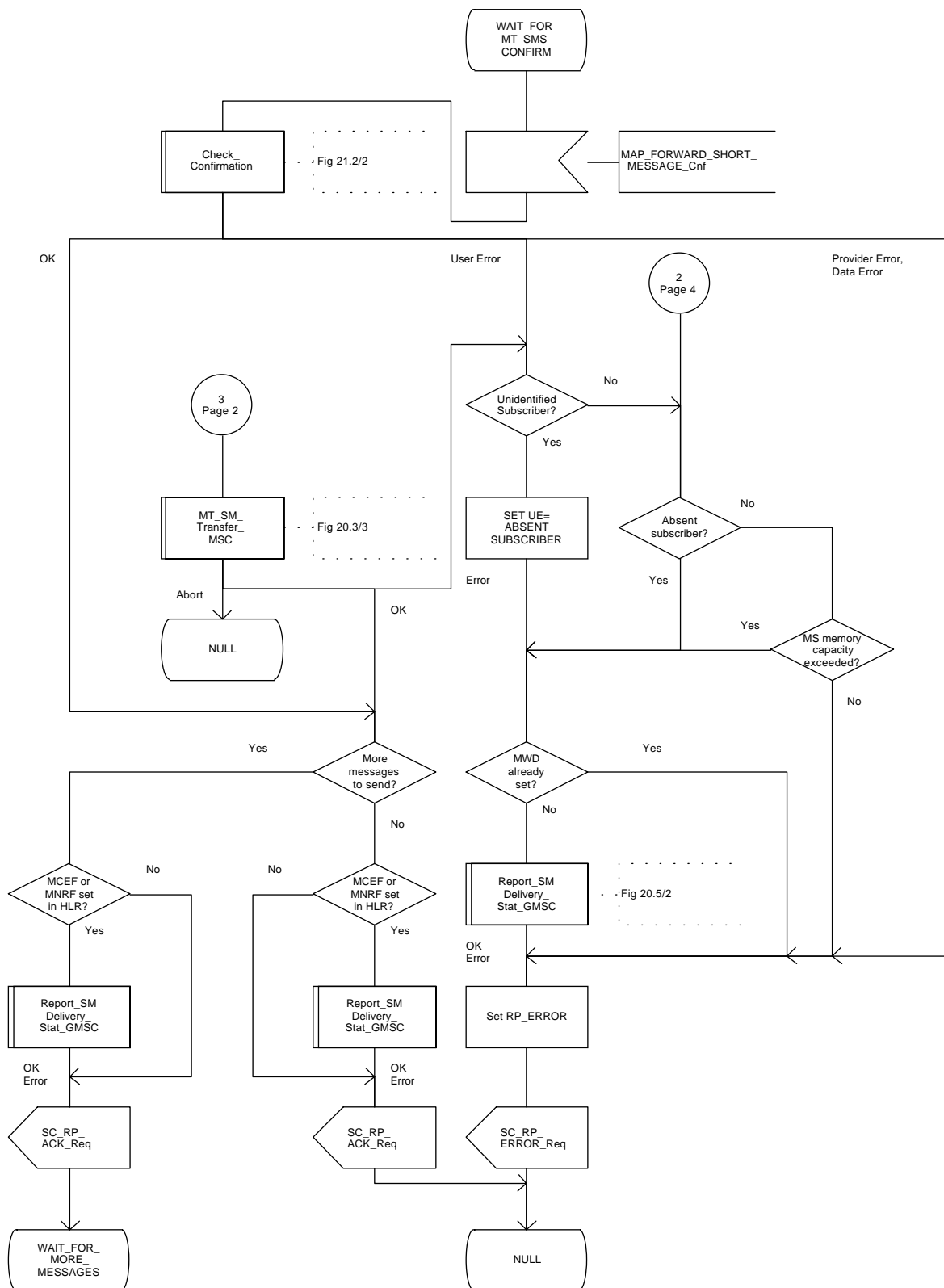
Figure 20.3/7 (sheet 4 of 5)
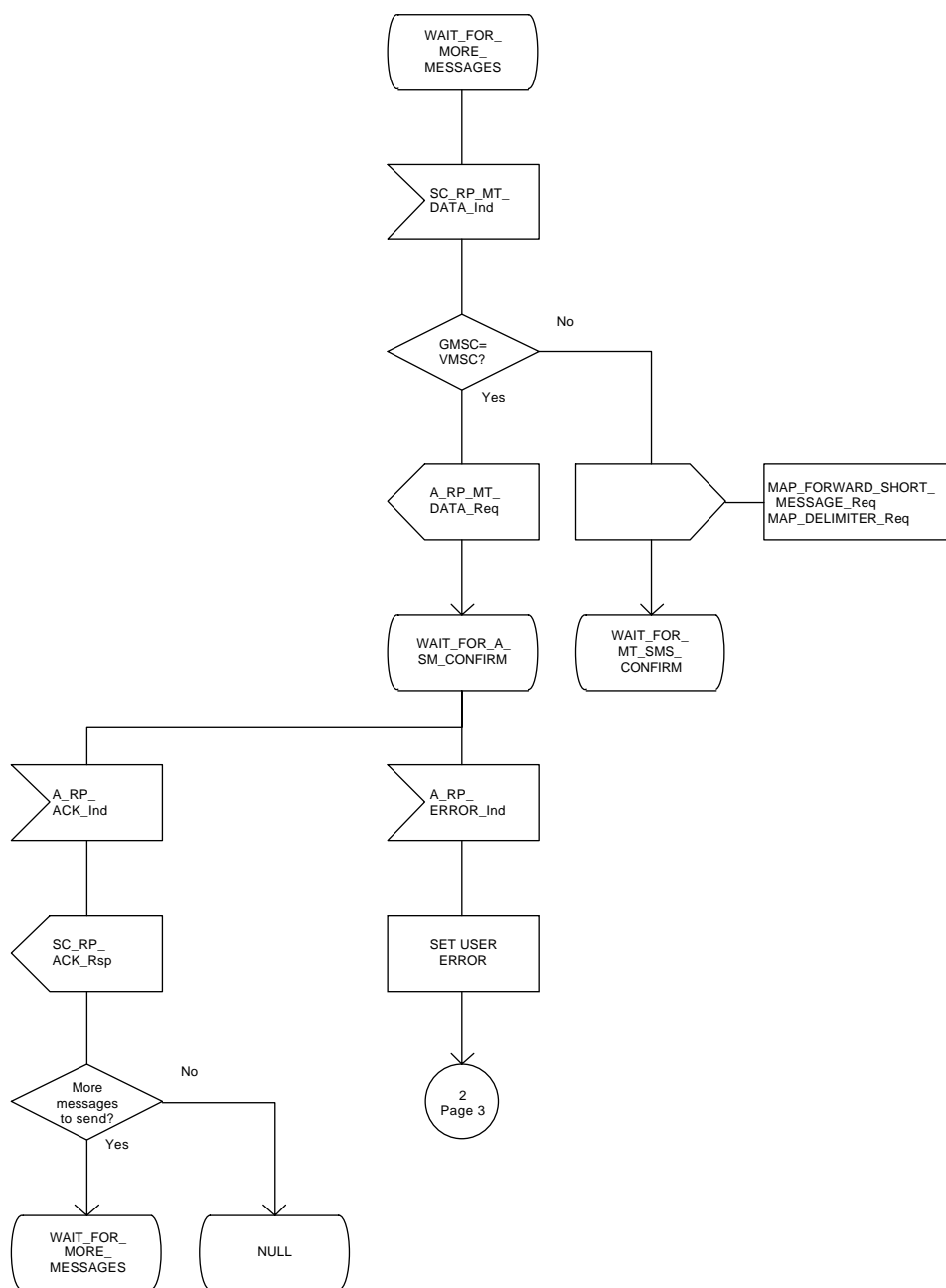
Figure 20.3/7 (sheet 5 of 5)
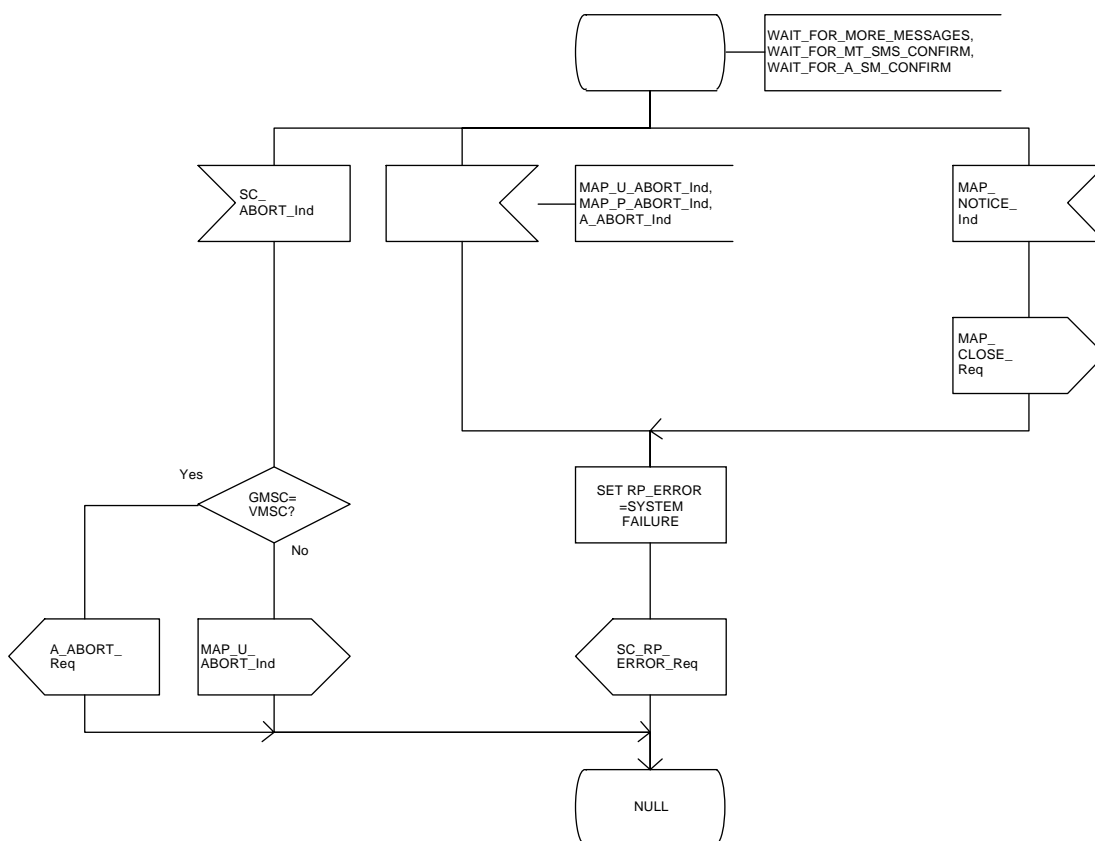
Figure 20.3/8
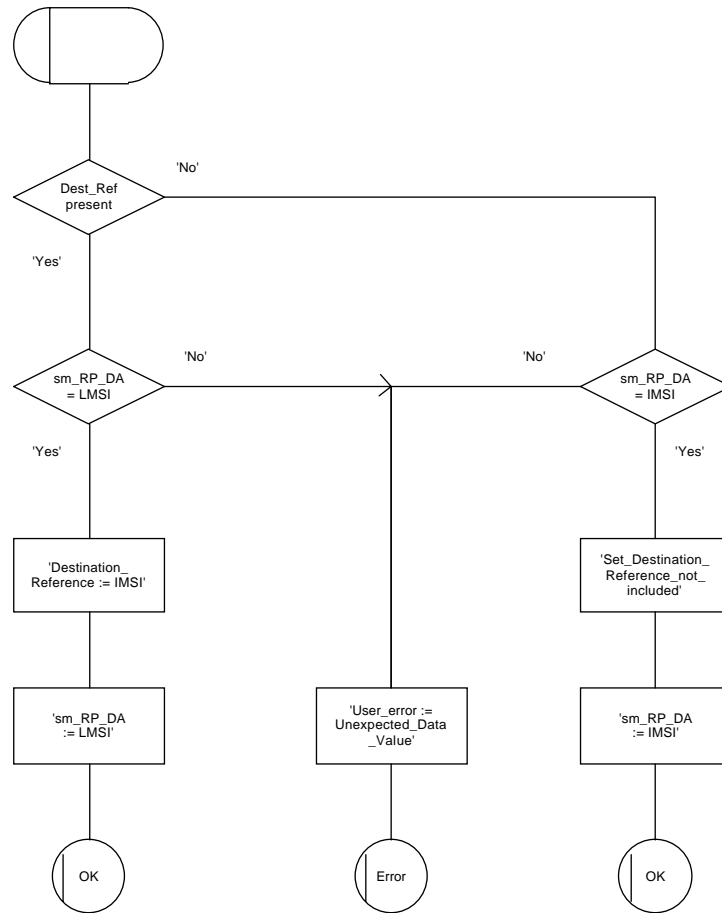
20.4 The Short Message Alert procedure

The Short Message Alert procedure is used for alerting the Service Centre when the mobile subscriber is active after a short message transfer has failed because the mobile subscriber is not reachable or when the MS has indicated that it has memory capacity to accept a short message.

The Short Message Alert procedure for the case when the mobile subscriber was not reachable is shown in figure 20.4/1.
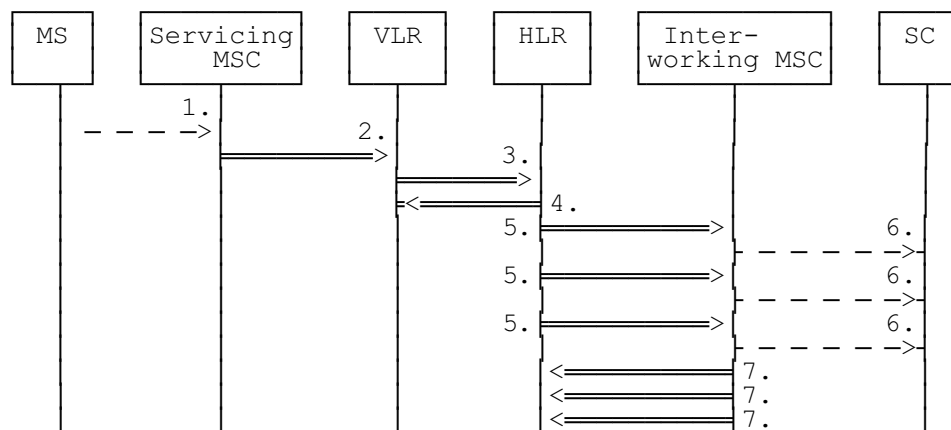
1) CM Service Request, Page response or Location Updating (BSSAP; GSM 04.08)
2) MAP_PROCESS_ACCESS_REQUEST / MAP_UPDATE_LOCATION_AREA
3) MAP_READY_FOR_SM (Mobile Present) / MAP_UPDATE_LOCATION /
   Supplementary Service Control Request
4) MAP_READY_FOR_SM_ACK
5) MAP_ALERT_SERVICE_CENTRE (notes 1 and 2)
6) Alert Service Centre (GSM 03.40)
7) MAP_ALERT_SERVICE_CENTRE_ACK

NOTE 1: To all Service Centres in the Message Waiting List.

NOTE 2: The HLR initiates the MAP_ALERT_SERVICE_CENTRE service only if the MS Memory Capacity Exceeded flag is clear.

Figure 20.4/1: Short message alert procedure (Mobile is present)

The Short Message Alert procedure for the case where the MS indicates that it has memory capacity to accept one or more short messages is shown in figure 20.4/2.
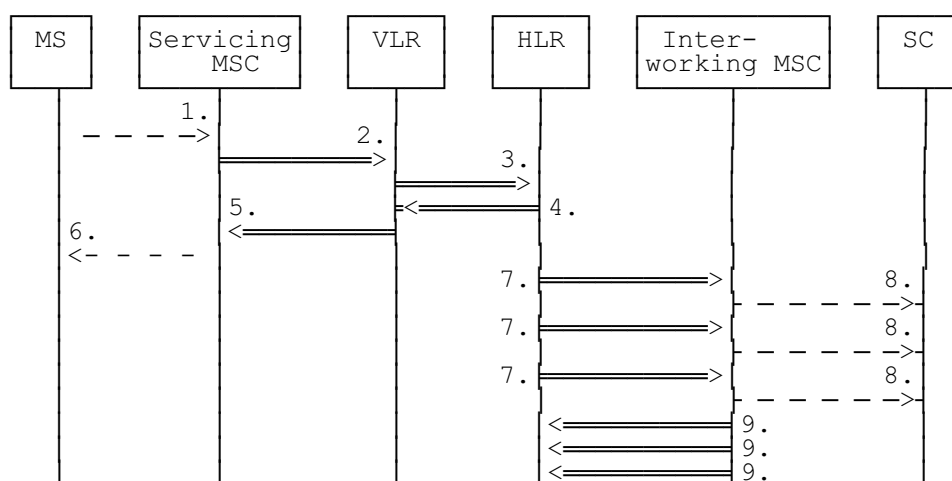
1) SM memory capacity available (BSSAP; GSM 04.11)
2) MAP_READY_FOR_SM (Memory Available)
3) MAP_READY_FOR_SM (Memory Available)
4) MAP_READY_FOR_SM_ACK
5) MAP_READY_FOR_SM_ACK
6) SM memory capacity available (Acknowledge) (BSSAP; GSM 04.11)
7) MAP_ALERT_SERVICE_CENTRE (note)
8) Alert Service Centre (GSM 03.40)
9) MAP_ALERT_SERVICE_CENTRE_ACK

NOTE: To all Service Centres in the Message Waiting List.

Figure 20.4/2: Short message alert procedure (MS memory capacity available)

In addition the following MAP services are used in the MS memory available case:

MAP_PROCESS_ACCESS_REQUEST   (see subclause 6.3);
MAP_AUTHENTICATE   (see subclause 6.5);
MAP_SET_CIPHERING_MODE   (see subclause 6.6);
MAP_PROVIDE_IMSI   (see subclause 6.9);
MAP_CHECK_IMEI   (see subclause 6.7);
MAP_FORWARD_NEW_TMSI   (see subclause 6.9);
MAP_TRACE_SUBSCRIBER_ACTIVITY   (see subclause 7.1).

The Short Message Alert procedure when the MS indicates successful transfer after polling is shown in figure 20.4/3.
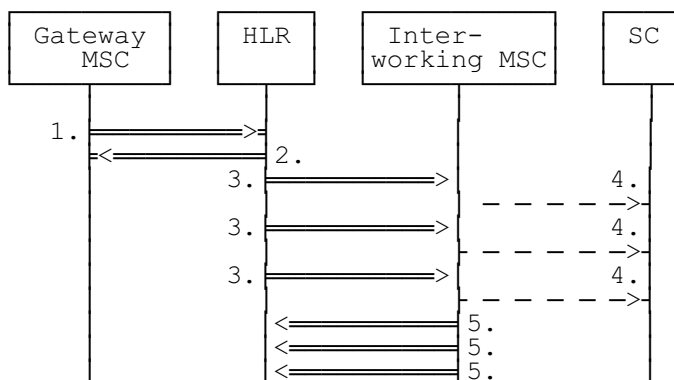
1) MAP_REPORT_SM_DELIVERY_STATUS (Successful Transfer)
2) MAP_REPORT_SM_DELIVERY_STATUS_ACK
3) MAP_ALERT_SERVICE_CENTRE (note)
4) Alert Service Centre (GSM 03.40)
5) MAP_ALERT_SERVICE_CENTRE_ACK

NOTE: To all Service Centres in the Message Waiting List.

Figure 20.4/3: Short message alert procedure (Successful transfer after polling)

20.4.1   Procedures in the Servicing MSC

The activation of the MAP_PROCESS_ACCESS_REQUEST service is described in the subclause 20.6.2.

After receiving the SM memory capacity available indication, the servicing MSC sends the MAP_READY_FOR_SM request to the VLR indicating memory available. The outcome of that procedure is one of the following:

- successful acknowledgment. The MSC sends the corresponding message to the MS;

- negative acknowledgment, where the error causes are treated as follows:
    - unexpected data value, data missing and system failure errors are reported as network out of order error to the MS;

    - facility not supported is reported as requested facility not implemented error to the MS;

- procedure failure, which is reported as network out of order error to the MS if a connection to the MS still exists.

The short message alert procedure in the MSC for the MS memory capacity available case is shown in figure 20.4/4.

Figure 20.4/4
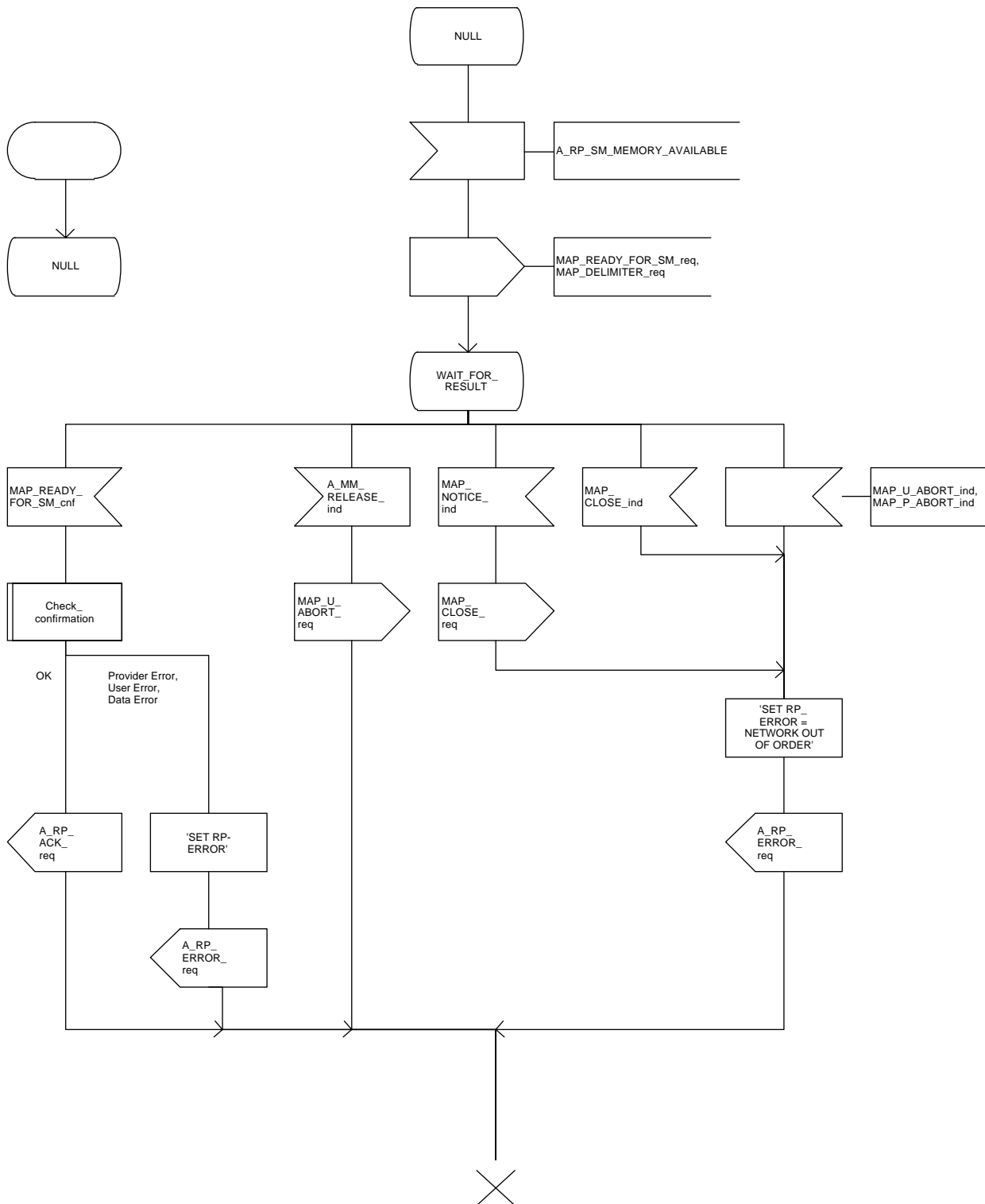
20.4.2   Procedures in the VLR

20.4.2.1   The Mobile Subscriber is present

When receiving the MAP_PROCESS_ACCESS_REQUEST indication, MAP_UPDATE_LOCATION_AREA indication while the MS not reachable flag (MNRF) is set, the VLR will send the MAP_READY_FOR_SM request towards the HLR. The Alert Reason is set to indicate that the mobile subscriber is present. If the authentication procedure is initiated and it fails, the VLR will not initiate the service. The process in VLR is described in detail in the subclause 21.10.

20.4.2.2   The Mobile Equipment has memory available

The MAP_PROCESS_ACCESS_REQUEST indication starts the MAP_PROCESS_ACCESS_REQUEST service in the VLR. The application context in the MAP_OPEN indication refers to the short message alerting procedure.

If the service MAP_PROCESS_ACCESS_REQUEST is successful, the VLR waits for the next message from the MSC. When receiving the MAP_READY_FOR_SM indication from the MSC, the VLR will check the contents. Data errors are reported to the MSC as an unexpected data value or data missing error, depending on the error. If the primitive passes the data check, the VLR forwards it to the HLR and awaits an acknowledgment.

When receiving the MAP_READY_FOR_SM confirmation from the HLR and the Alert Reason is MS memory available, the VLR will act as follows:

-        the MAP_READY_FOR_SM response is sent to the MSC as follows:
    -        an acknowledge in the positive case;

    -        system failure error, if unexpected data value, data missing, or unknown subscriber errors are received, otherwise the error cause received from the HLR;

    -        a facility not supported error, if the HLR supports MAP V1 only;

    -        procedure failure is reported as a system failure error.

The short message alert procedure in the VLR is shown in figures 20.4/5.

Figure 20.4/5
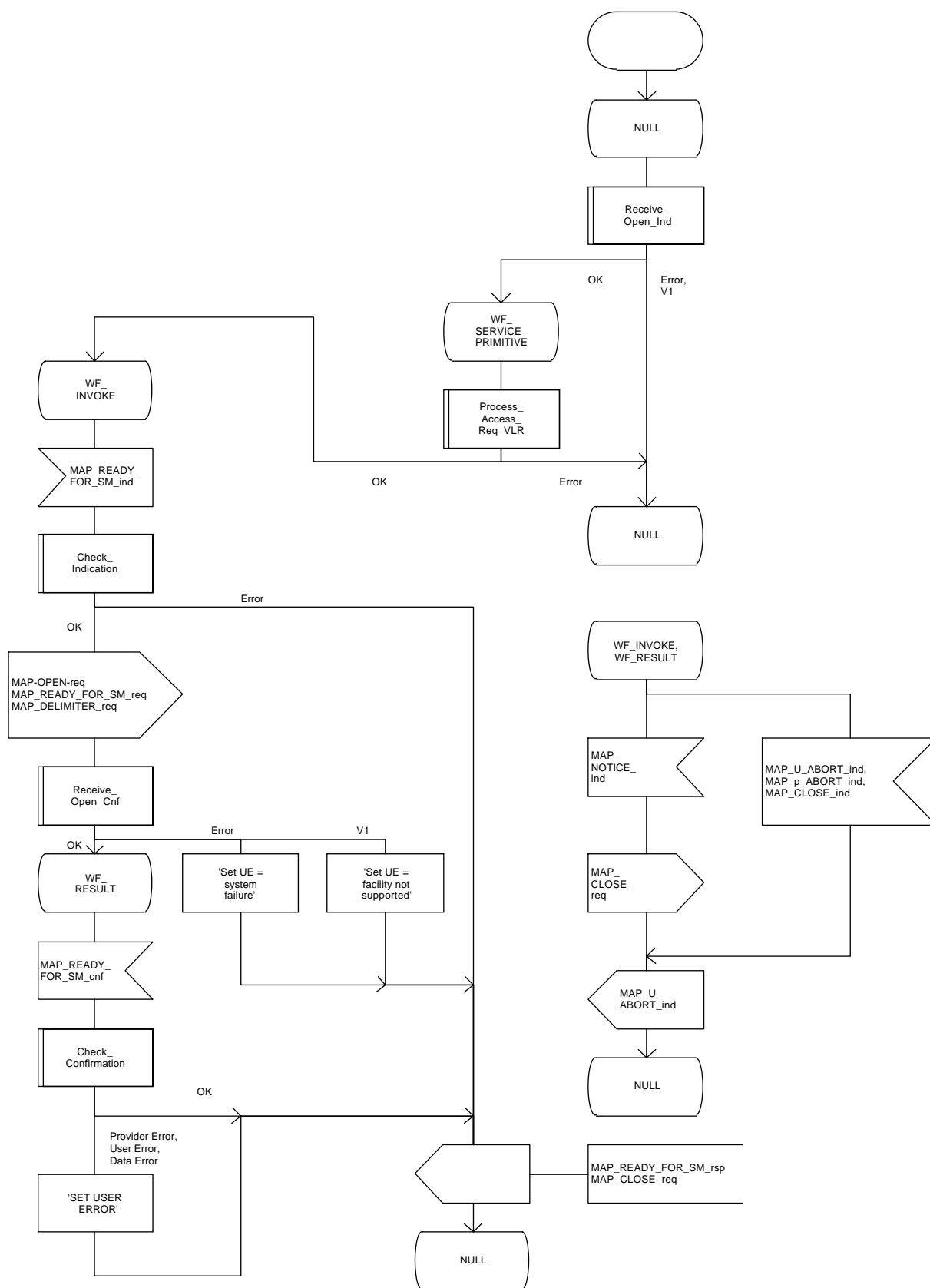
20.4.3   Procedures in the HLR

When receiving the MAP_READY_FOR_SM indication, the HLR will check the contents. Data errors are reported to the VLR as an unexpected data value or a data missing error depending on the error. If the HLR does not support the MNRF, MCEF, and MWD a facility not supported error is reported to the VLR. If the IMSI is unknown an unknown subscriber error is reported to the VLR. Otherwise an acknowledgement is returned to the VLR.

If neither the MS not reachable flag (MNRF) or the memory capacity exceeded flag (MCEF) is set, the HLR sets a timer and waits for it to expire. This ensures that in the race situation the MAP_REPORT_SM_DELIVERY_STATUS service (as described in the subclause 20.6) for the same subscriber can be carried out when delayed in the GMSC.

If the Alert Reason indicates the mobile present situation, or when the update location procedure has been successfully completed or Supplementary Service Control request is received, the MS not reachable flag is cleared and the service centre alert procedure is initiated. If the memory capacity exceeded flag is set, the MS not reachable flag is cleared but the alert procedure is not started.

If the Alert Reason indicates the memory available situation, the HLR initiates the alert procedure. The MS not reachable and memory capacity available flags are cleared.

If the MAP_REPORT_SM_DELIVERY_STATUS indication is received and it indicates the successful transfer of the mobile terminated short message, the HLR initiates the alert procedure described in the subclause 21.10 and clears MCEF and MNRF flags.

The short message alert procedure in the HLR is shown in figures 20.4/6 and 21.10/2.

Figure 20.4/6
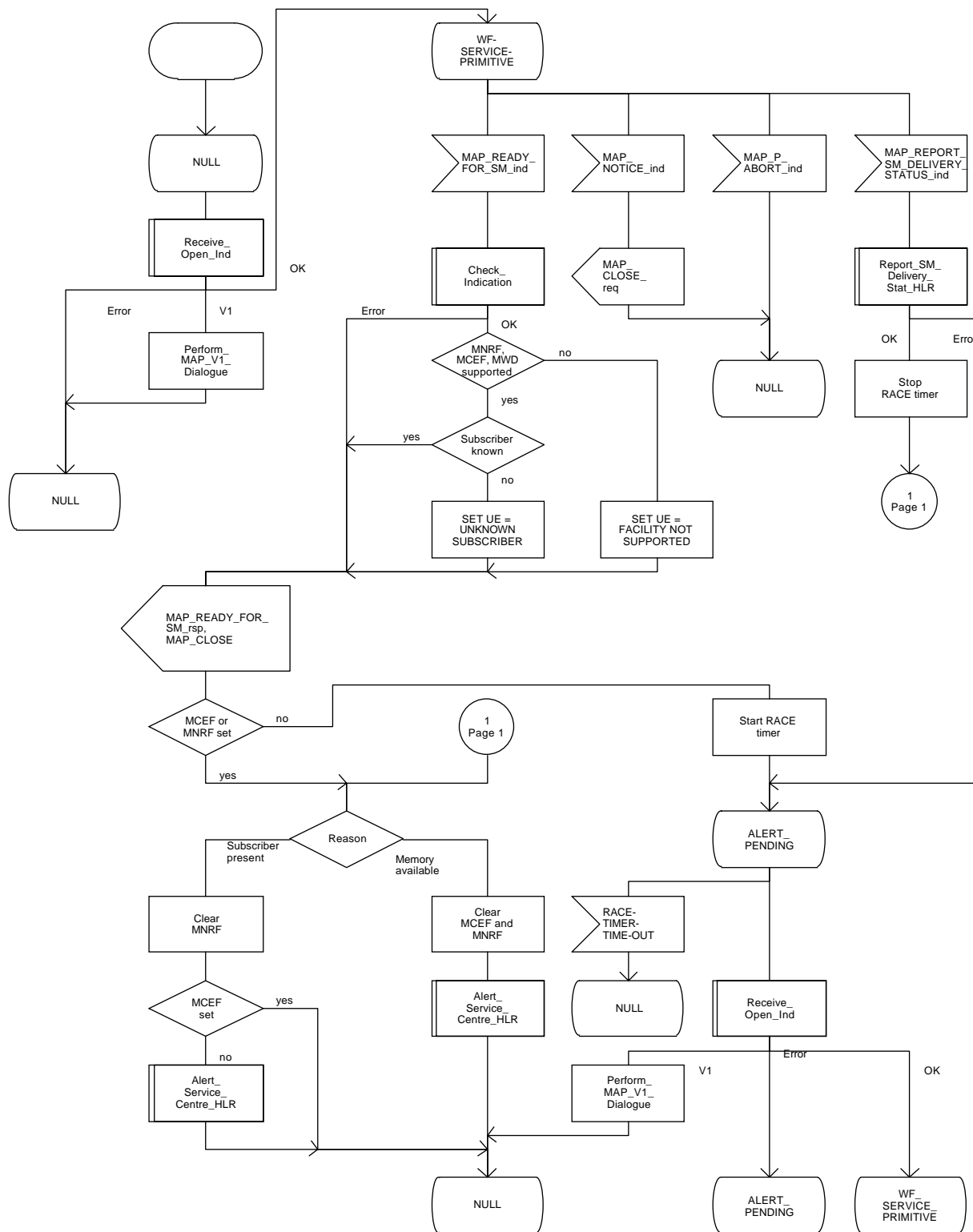
20.4.4   Procedures in the Interworking MSC

When a MAP_ALERT_SERVICE_CENTRE indication is correctly received by the IWMSC, the IWMSC will forward the alerting to the given Service Centre if possible.

Data errors are reported to the HLR as an unexpected data value or a data missing error depending on the error.

The short message alert procedure is shown in figure 20.4/7.

Figure 20.4/7
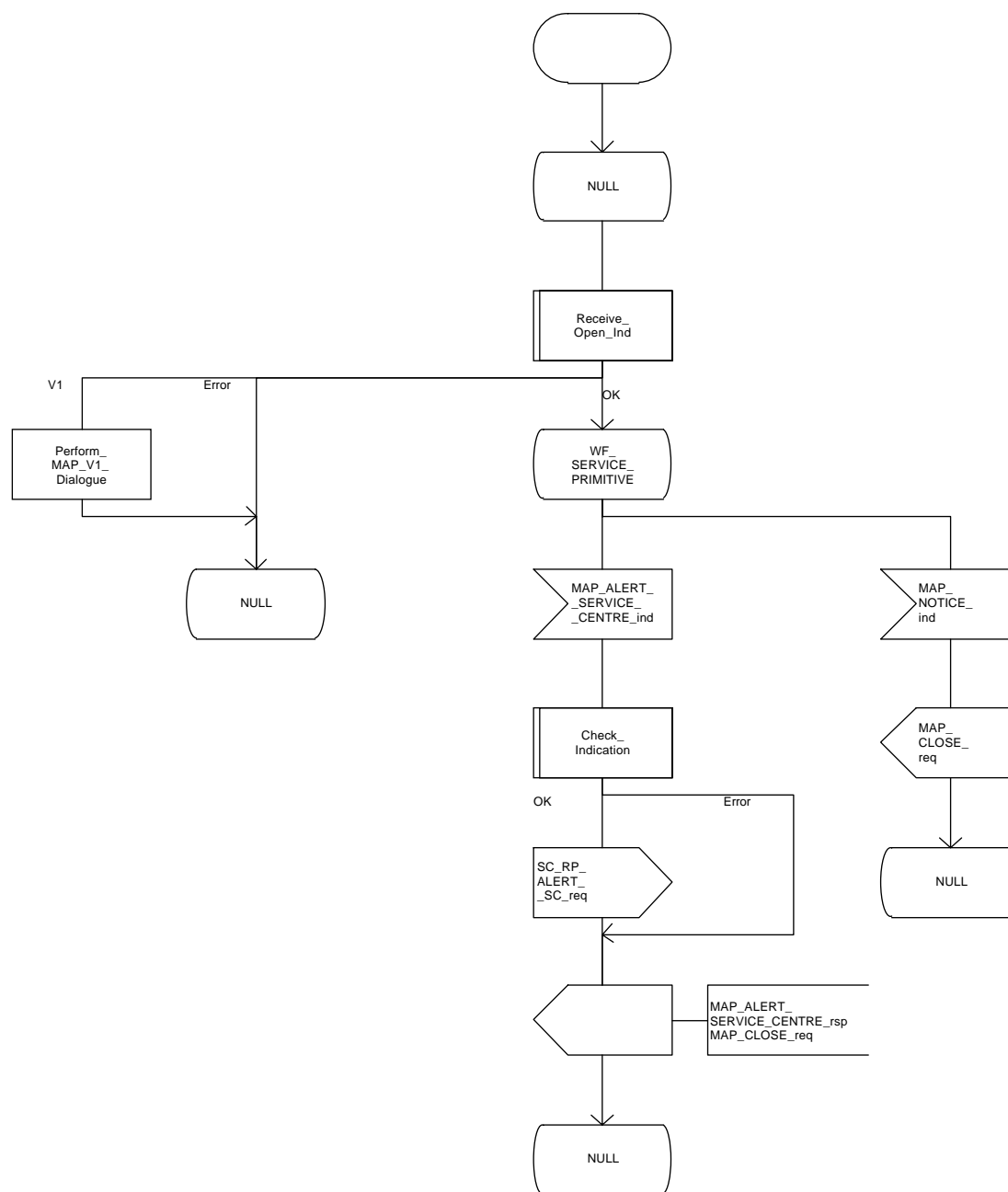
20.5   The SM delivery status report procedure

The SM delivery status report procedure is used to set the Service Centre address into the message waiting list in the HLR because the subscriber is absent or unidentified or the memory capacity is exceeded. The procedure sets the memory capacity exceeded flag in the HLR if the MS memory does not have room for more messages or the MS not reachable flag in the case of unidentified or absent subscriber.

Additionally the procedure is used to report the HLR about the successful transfer after the Service Centre has polled the subscriber. This procedure is described also in the subclause 20.4.

The SM delivery status report procedure is shown in figure 20.5/1.

1)   MAP_FORWARD_SHORT_MESSAGE_ACK/_NACK (Absent subscriber,
      unidentified subscriber or memory capacity exceeded)
2)   MAP_REPORT_SM_DELIVERY_STATUS
3)   MAP_REPORT_SM_DELIVERY_STATUS_ACK
4)   Short Message Negative Acknowledgement (GSM 03.40)
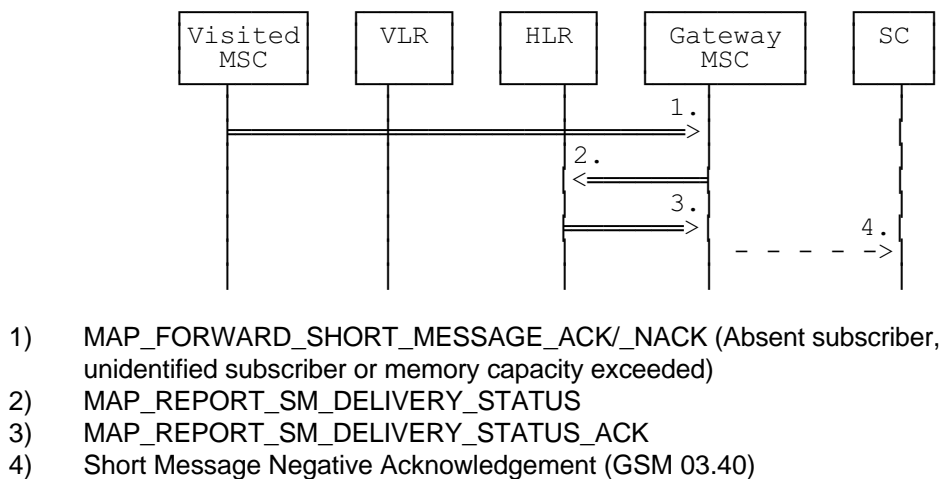
Figure 20.5/1: Short message delivery status report procedure

20.5.1   Procedures in the HLR

When the HLR receives a MAP_REPORT_SM_DELIVERY_STATUS indication, it acts as described in the subclause 20.6, macro Report_SM_Delivery_Stat_HLR.

The short message delivery status report process in the HLR is shown in figure 20.5/2.

Figure 20.5/2
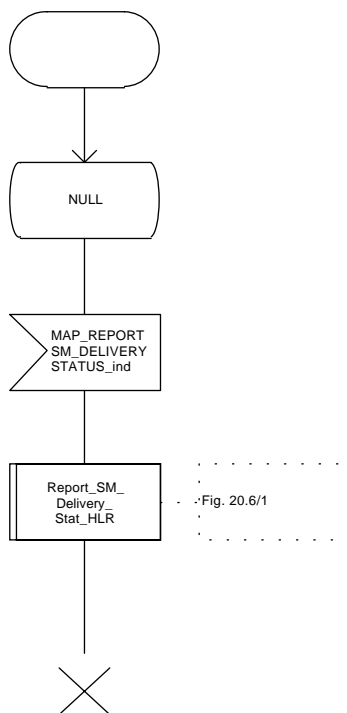
20.5.2 Procedures in the gateway MSC

The GMSC invokes the short message delivery status report procedure if an absent subscriber indication or unidentified subscriber indication or SM delivery failure error indicating MS memory capacity exceeded is received from the servicing MSC during a mobile terminated short message transfer, and the HLR has not indicated that the SC address is included in the MWD. The unidentified subscriber indication is however processed as the absent subscriber indication.

The service is invoked also when the HLR has indicated that either of the flags MCEF or MNRF is set and the SM delivery was successful.

The reason for unsuccessful or successful delivery of the short message is included in the SM Delivery Outcome in the MAP_REPORT_SM_DELIVERY_STATUS request.

The GMSC sends the MAP_REPORT_SM_DELIVERY_STATUS request to the HLR. As a response the GMSC will receive the MAP_REPORT_SM_DELIVERY_STATUS confirmation reporting:

-    successful outcome of the procedure. The acknowledge primitive may contain the MSISDN-Alert number which is stored in the MWD List in the HLR;

-    unsuccessful outcome of the procedure. The system failure indication is forwarded to the SC.

A provider error is indicated as a system failure to the SC.

The procedure towards the Service Centre may also be aborted. If so the operation towards the HLR is also aborted.

The short message delivery status report procedure in the GMSC is shown in figure 20.5/3.

Figure 20.5/3
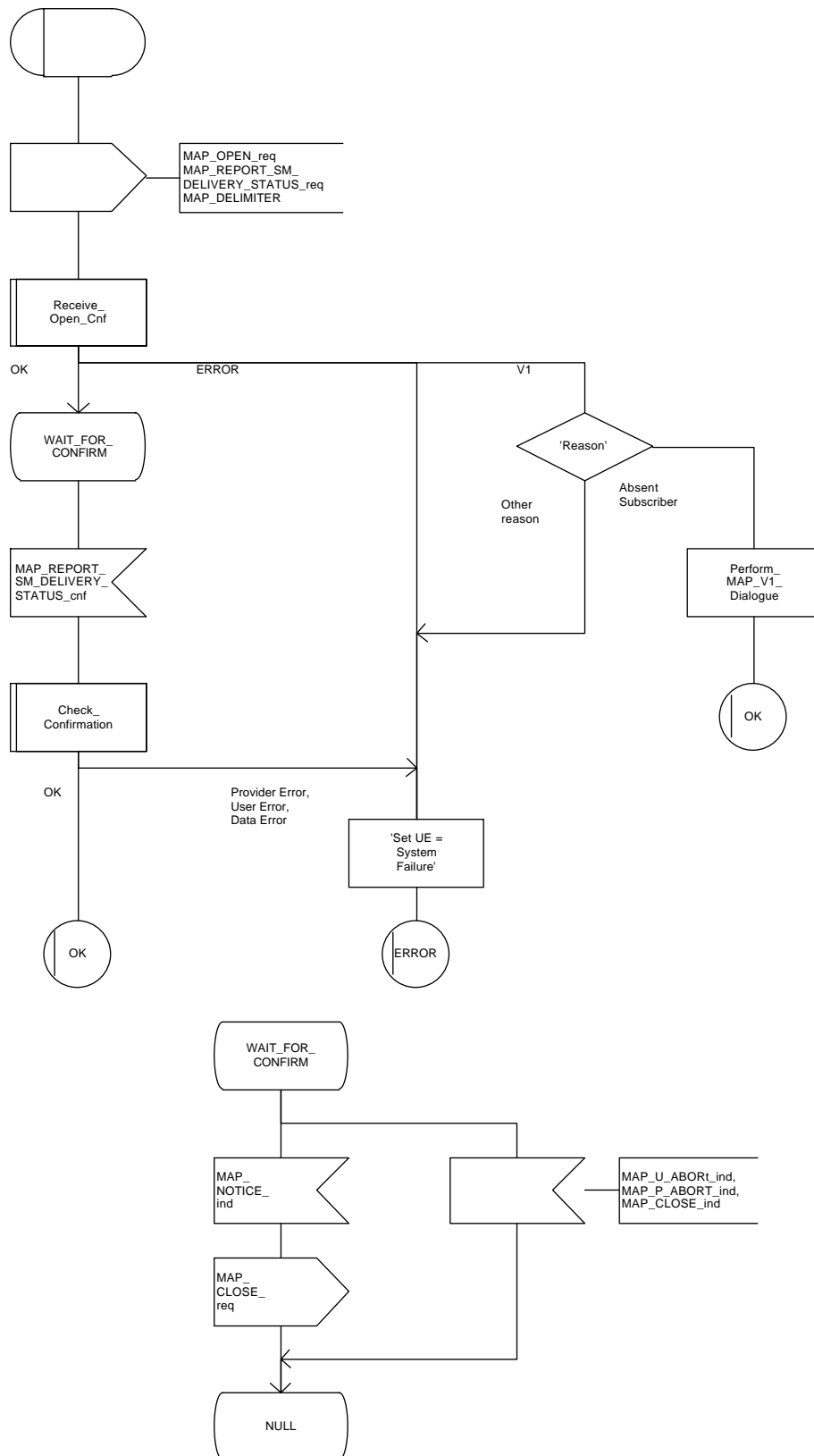
20.6   Common procedures for the short message clause

20.6.1   The macro Report_SM_Delivery_Stat_HLR

This macro is used when the HLR receives a MAP_REPORT_SM_DELIVERY_STATUS indication from the GMSC. The HLR responses to the indication as follows:

-        if invalid data content is detected, an unexpected data value error or a data missing error is returned to the GMSC;

-        if the MSISDN number provided is not recognized by the HLR, an unknown subscriber error is returned to the GMSC;

-        if the MAP_REPORT_SM_DELIVERY_STATUS indication reports a successful SM delivery, the Service Centres in the Message Waiting list are alerted as described in the subclause 21.10;

-        if the SM Delivery Outcome reports unsuccessful delivery and the inclusion of the SC address in the MWD is not possible, a message waiting list full error is returned to the GMSC;

-        if the SM Delivery Outcome reports unsuccessful delivery and the message waiting list is not full, the given Service Centre address is inserted and an acknowledgement is sent to the GMSC. If the MSISDN-Alert stored in the subscriber data is not the same as that received in the MAP_REPORT_SM_DELIVERY_STATUS indication, the MSISDN-Alert is sent in a response primitive to the GMSC;

-        if the SM Delivery Outcome is MS memory capacity exceeded the HLR sets the memory capacity exceeded flag in the subscriber data and resets the MRNF;

-        if the SM Delivery Outcome is absent subscriber the HLR sets the MS not reachable flag in the subscriber data.

The short message delivery status report macro in the HLR is shown in figure 20.6/1.

Figure 20.6/1
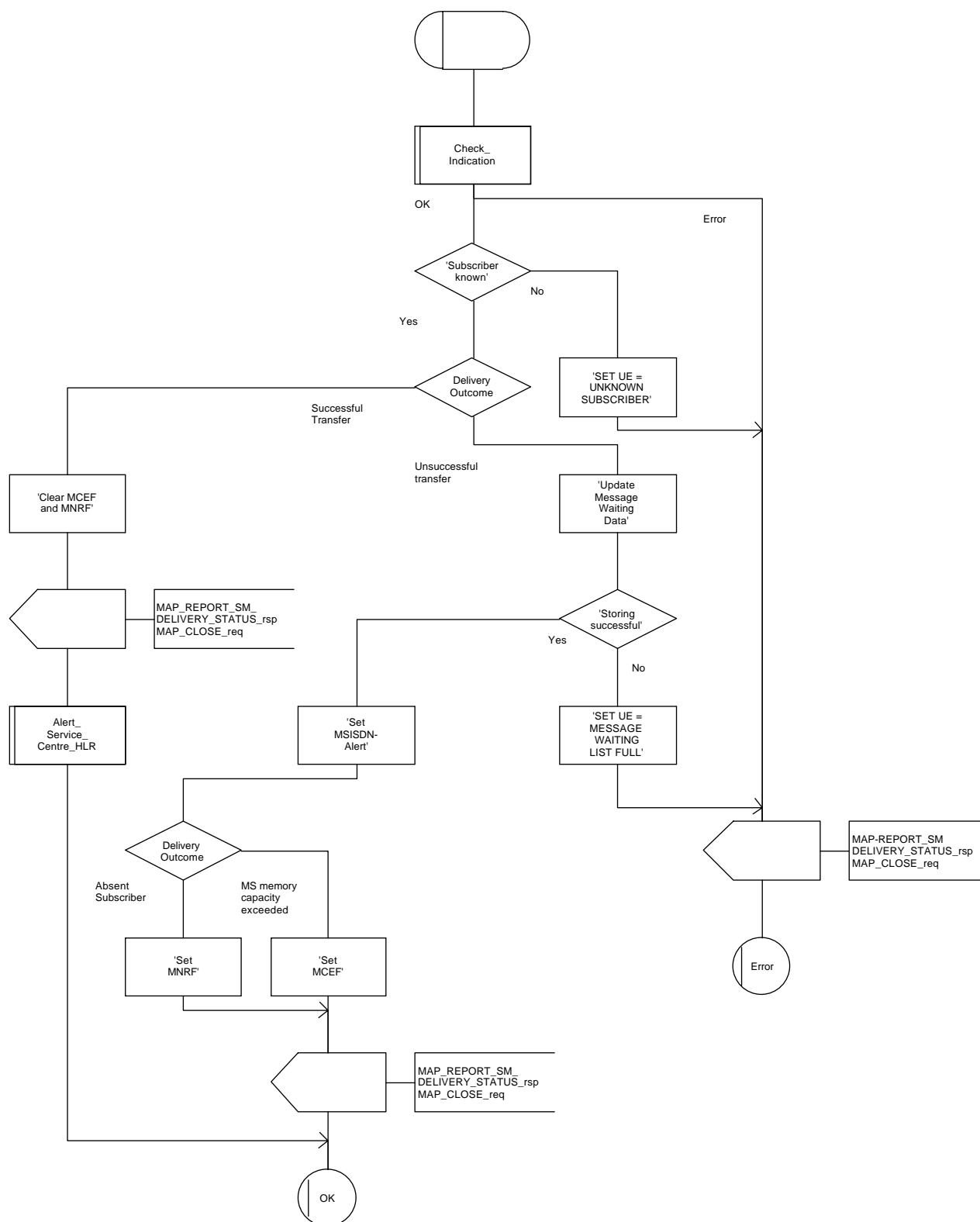
21   General macro description

21.1   MAP open macros

21.1.1   Macro Receive_Open_Ind

This macro is used by a MAP service-user procedure when a peer entity requests opening of a dialogue.

If the application context received in the MAP-OPEN indication primitive indicates a context name of the MAP version one context set, the macro terminates respectively.

If an application-context different from version 1 is received, the presence of MAP_OPEN information is checked. If no MAP_OPEN information has been received, the MAP_OPEN response with:

-       Result set to Dialogue Accepted; and

-       Application Context Name set to the received value,

is returned and the macro terminates successfully.

If MAP_OPEN information is received, the macro "CHECK_REFERENCE" is called in order to check whether the received values for Destination Reference and Originating Reference correspond with the requirements of the received application-context-name. The outcome of this check is an error, the MAP_OPEN response with:

-       Result set to Dialogue Refused;

-       Refuse Reason set to Invalid Destination Reference or Invalid Originating Reference;

-       Application Context Name set to the highest version supported,

is returned and the macro terminates with unsuccessful outcome.

If the data values received for Destination Reference and Originating Reference are accepted for the associated application-context-name it is checked whether the Destination Reference is known if this check is required by the process that calls the macro.

If the Destination Reference (e.g. a subscribers IMSI) is unknown, the MAP_OPEN response with

-       Result set to Dialogue Refused;

-       Refuse Reason set to Invalid Destination Reference;

-       Application Context Name set to the highest version supported,

is returned and the macro terminates with unsuccessful outcome.

Else, if the Destination Reference is accepted or if no check is required, the MAP_OPEN response with

-       Result set to Dialogue Accepted; and

-       Application Context Name set to the received value,

is returned and the macro terminates successfully.

21.1.2 Macro Receive_Open_Confirmation

This macro is used by a user procedures after it requested opening of a dialogue towards a peer entity.

On receipt of a MAP_OPEN Confirmation with a "Result" parameter indicating "Dialogue Accepted", the macro terminates successfully.

If the "Result" parameter indicates "Dialogue Refused", the "Refuse-reason" parameter is examined. If the "Refuse-reason" parameter indicates "Potential Version Incompatibility", the macro terminates in a way that causes restart of the dialogue by using the version 1 protocol.

If the "Refuse-reason" parameter indicates "Application Context Not Supported" and if the received Application Context Name indicates "Version 1", the macro terminates in a way that causes restart of the dialogue by using the version 1 protocol. If the received Application Context Name does not indicate "Version 1", the dialogue is terminated with unsuccessful outcome.

If the "Refuse-reason" parameter indicates neither "Potential Version Incompatibility" nor "Application Context Not Supported", the dialogue is terminated with unsuccessful outcome.

If a MAP_U_ABORT, a MAP_P_ABORT or a MAP_NOTICE Indication is received, the dialogue is terminated with unsuccessful outcome.

Figure 21.1/1
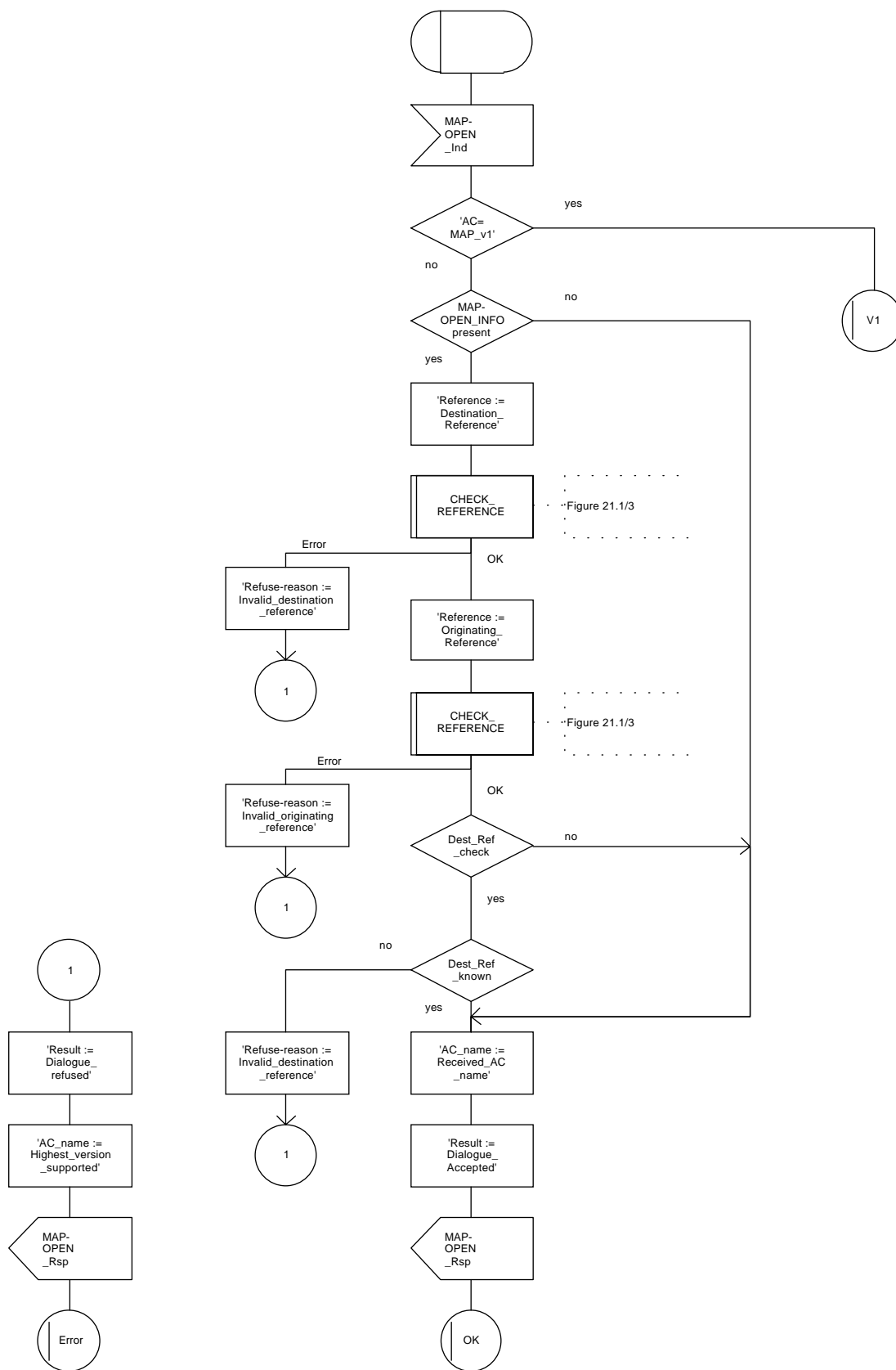
Figure 21.1/2
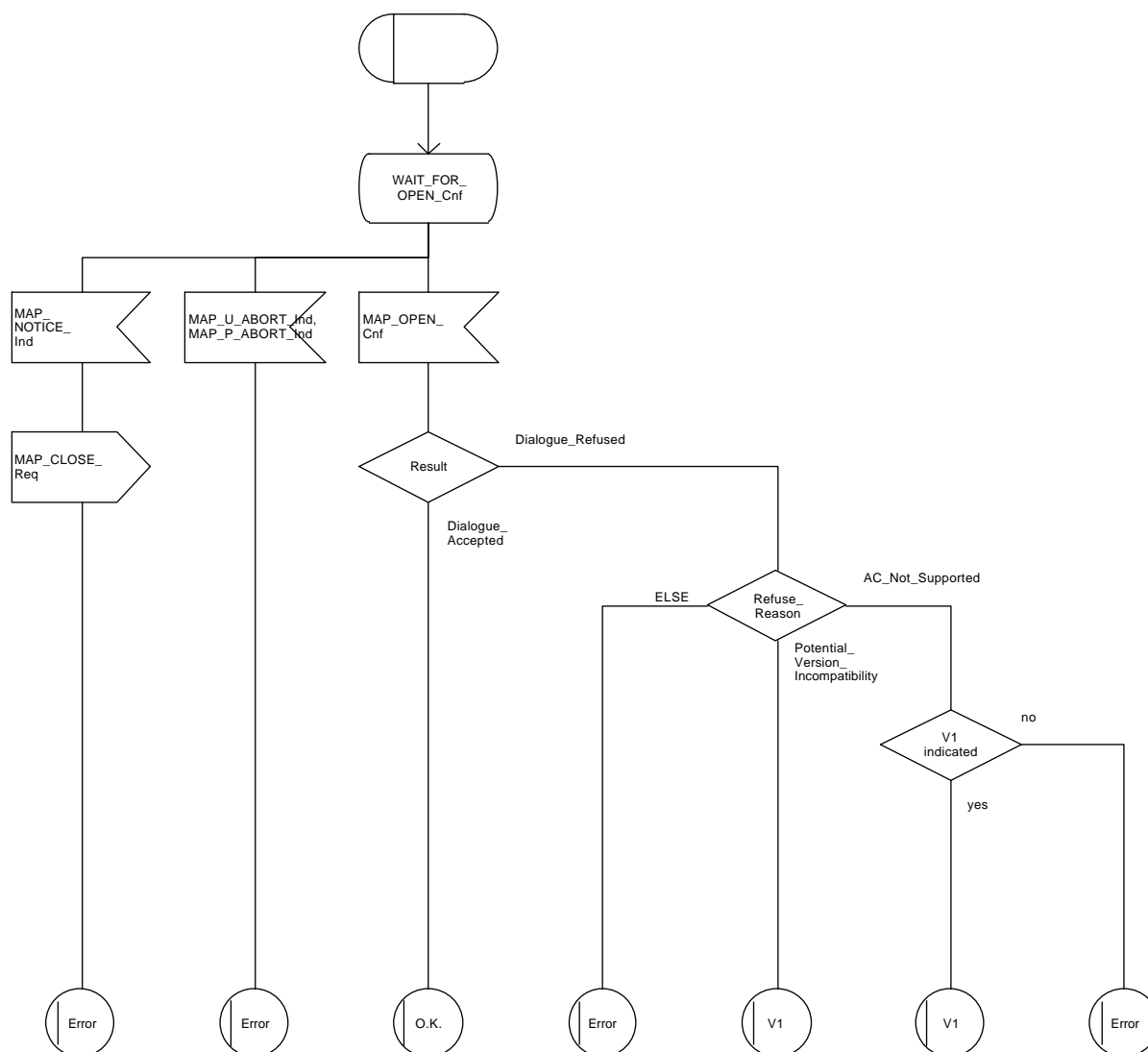
Figure 21.1/3
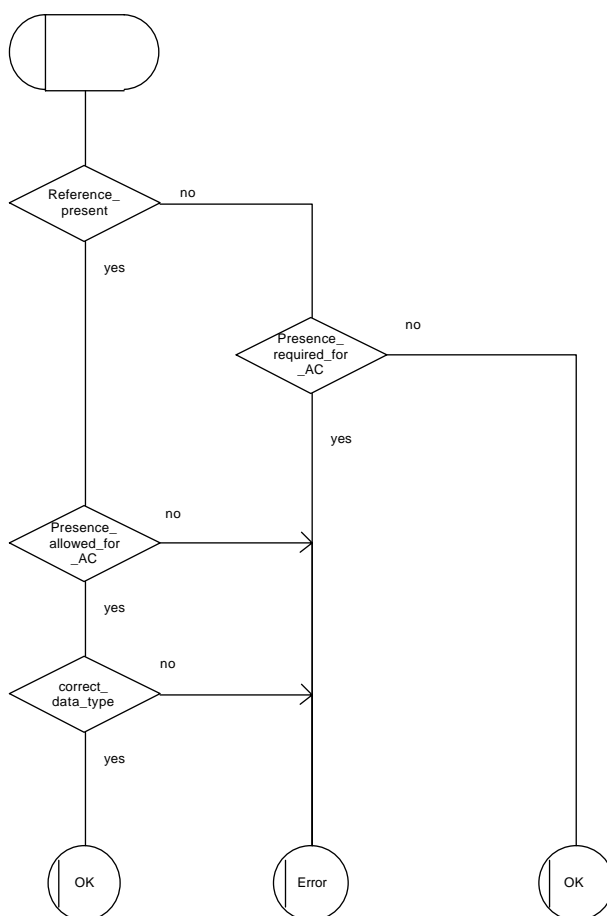
21.2 Macros to check the content of indication and confirmation primitives

21.2.1 Macro Check_Indication

If a parameter required by the application is missing from the indication, the macro takes the error exit, with a user error of "Data Missing".

If a parameter not expected by the application is present in the indication, or an expected parameter has a value not in the set of values permitted by the application, the macro takes the error exit, with a user error of "Unexpected Data Value".

Otherwise the macro takes the "OK" exit.

The macro is shown in figure 21.2/1.

21.2.2 Macro Check_Confirmation

If the confirmation contains a provider error the macro issues a MAP CLOSE request and takes the provider error exit.

Otherwise, if the confirmation contains a user error the macro takes the user error exit.

Otherwise, if a parameter required by the application is missing from the confirmation, or a parameter not expected by the application is present in the confirmation, or an expected parameter has a value not in the set of values permitted by the application, the macro takes the data error exit.

Otherwise the macro takes the "OK" exit.

The macro is shown in figure 21.2/2.

Figure 21.2/1
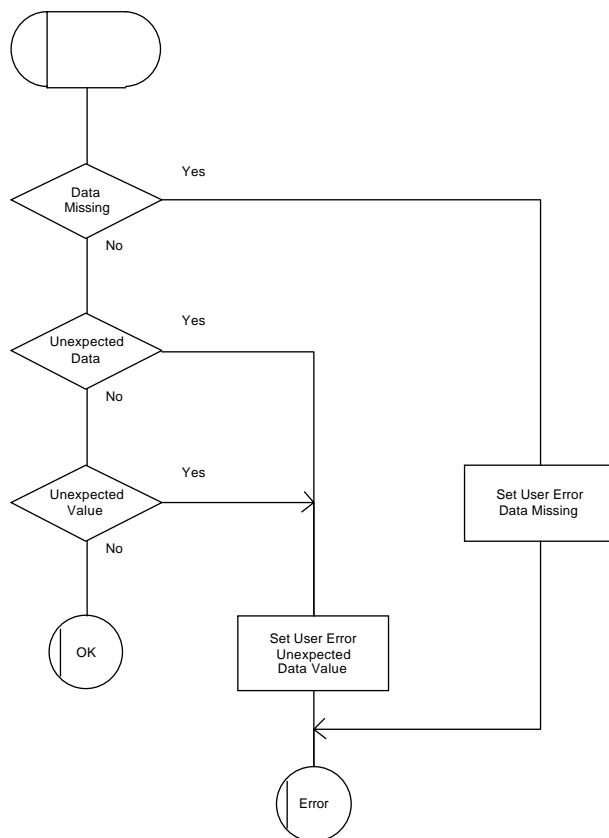
Figure 21.2/2
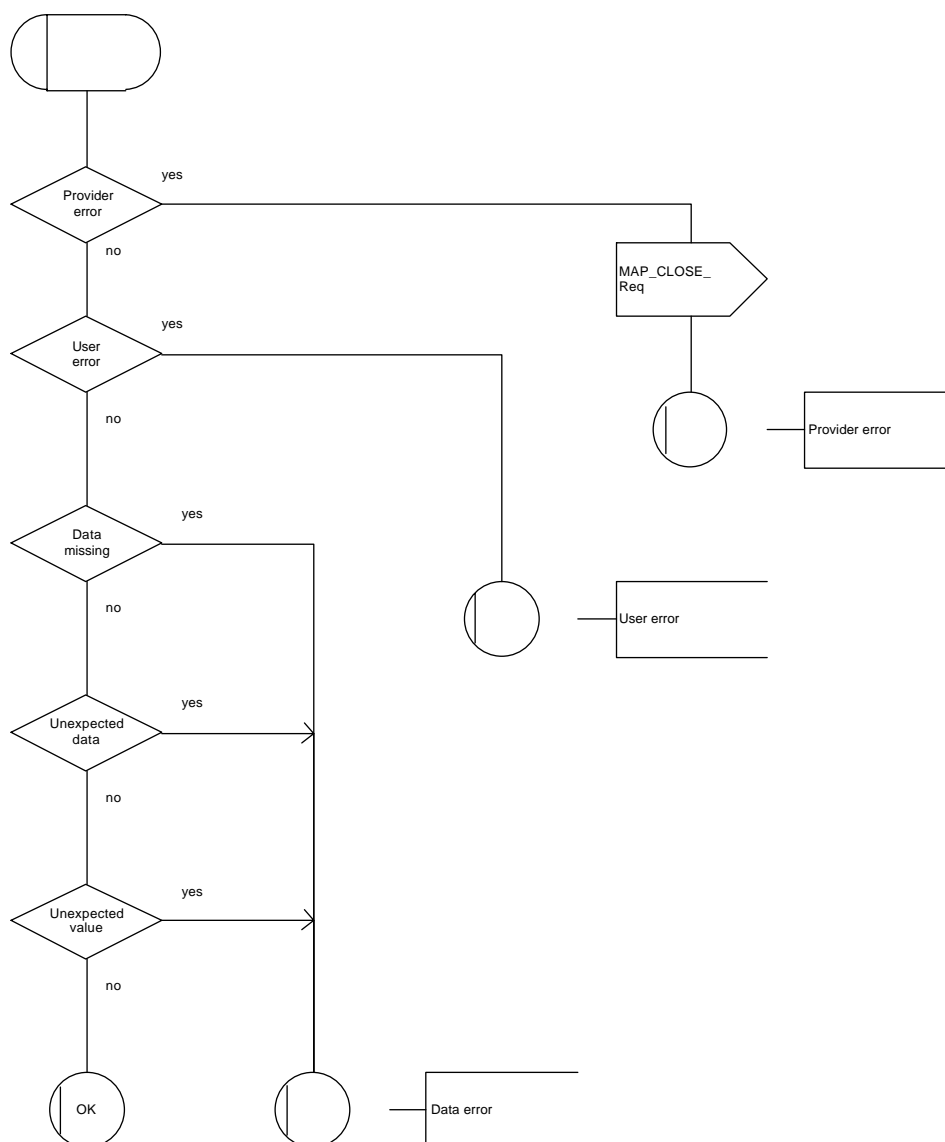
21.3 The page and search macros

21.3.1 Macro PAGE_MSC

This macro (see figure 21.3/1) is called if a mobile terminating call set-up, an unstructured SS notification, a network-initiated unstructured SS request or a mobile terminating short message is to be delivered to the MS and the current location area identity of the MS is known in the VLR.

When the MSC receives a MAP_PAGE indication, parameter checks are performed first (macro Check_Indication, see subclause 21.2). If parameter errors are detected, the MSC returns a MAP_PAGE response containing the appropriate error cause and the macro terminates with unsuccessful outcome.

Thereafter, several checks on the indication content are performed. The macro terminates by returning the MAP_PAGE response with error:

Unknown Location Area if the LAI is not known in the MSC;

System Failure if the call has been released by the calling subscriber or the SMS or SS transaction for this subscriber has been released by the originating entity in the meantime.

Next, the MSC checks if an MM-connection over the radio link already exists for the given IMSI. If so,

- in the case of mobile terminating call set-up the MSC determines whether the busy condition can be established (see GSM 02.01 for a definition of busy states). If the MSC determines that the MS is busy, it returns a MAP_PAGE response with error Busy Subscriber, qualified by either More Calls Allowed or No More Calls Allowed. The macro then terminates with unsuccessful outcome.

- if the service requested is short message service or an unstructured SS notification or network-initiated unstructured SS request, or if the service is mobile terminating call set-up, but the existing connection is for signalling purposes only (i.e. a service different from call set-up), the access connection status is set according to the characteristics of the existing connection (i.e. RR-connection established, ciphering mode on/off, MM-connection existing and authenticated or not), and the macro terminates with successful outcome.

If no MM-connection for the given IMSI exists, paging is initiated at the radio interface within all cells of the location area indicated by the VLR. If the VLR provided the TMSI, the MSC uses it to identify the MS at the radio interface; otherwise the MSC uses the IMSI. The IMSI will also be used to determine the page group (see GSM 04.08). There are several possible outcomes of paging:

- the MS responds to paging, causing the access connection status to be set accordingly (i.e. no RR-connection, in which case other values are not significant), and the macro terminates with successful outcome;

- the MS responds with a channel request containing an establishment cause which is not "answer to paging". The MSC sends a MAP_PAGE response primitive with user error Busy Subscriber before the macro terminates with unsuccessful outcome. This will give priority to the mobile originating request. Alternatively, as an implementation option, the MSC may treat this as a response to paging, which will give priority to the mobile terminating request.

- there is no response from the MS. The MSC sends a MAP_PAGE response primitive with user error Absent Subscriber before the macro terminates with unsuccessful outcome;

- the call handling connection or MAP transaction on which the call, SMS or unstructured SS transaction is waiting for delivery, is released before a response is received from the MS (indicated in the SDL by the input signal I-REL). The MAP transaction with the VLR will be released in this case by a MAP_U_ABORT request, and the unsuccessful macro termination will indicate transaction termination.

- the MAP transaction with the VLR may be released by receiving a MAP_U_ABORT or MAP_P_ABORT indication. The call handling connection or MAP transaction on which the call, SMS or unstructured SS transaction is waiting for delivery, is released (indicated in the SDL by the output signal I-REL), and the unsuccessful macro termination will indicate transaction termination.

21.3.2   Macro Search_For_MS_MSC

This macro (see figure 21.3/2) is called if a mobile terminating call set-up, an unstructured SS notification, a network-initiated unstructured SS request or a mobile terminating short message is to be delivered to the MS and the current location area identity of the MS is not known in VLR.

When the MSC receives a MAP_SEARCH_FOR_MS Indication, parameter checks are performed first (macro Check_indication, see subclause 21.2). If parameter errors are detected, the MSC returns a MAP_SEARCH_FOR_MS response containing the appropriate error cause and the macro terminates with unsuccessful outcome.

Thereafter, the MSC checks whether the call or the SMS or SS transaction still exists in the MSC. If the call or the SMS or SS transaction has been released, the MSC returns a MAP_SEARCH_FOR_MS response with error System Failure and the macro terminates with unsuccessful outcome.

Next, the MSC checks if an MM-connection over the radio link already exists for the given IMSI. If so,

- in the case of mobile terminating call set-up the MSC determines whether the busy condition can be established (see GSM 02.01 for a definition of busy states). If the MSC determines that the MS is busy, it returns a MAP_SEARCH_FOR_MS response with error Busy Subscriber, qualified by either More Calls Allowed or No More Calls Allowed. The macro then terminates with unsuccessful outcome.

- if the service requested is short message service or an unstructured SS notification or network-initiated unstructured SS request, or if the service is mobile terminating call set-up, but the existing connection is for signalling purposes only (i.e. a service different from call set-up), a MAP_SEARCH_FOR_MS response containing the IMSI and current location area identification of the called MS is returned to the VLR. The access connection status is set according to the characteristics of the existing connection (i.e. RR-connection established, ciphering mode on/off, MM-connection existing and authenticated or not), and the macro terminates with successful outcome.

If no MM-connection for the given IMSI exists, paging is initiated at the radio interface within all cells of all location areas of the VLR, using the IMSI to identify the subscriber and the page group (see GSM 04.08). There are several possible outcomes of paging:

- the MS responds to paging, causing a MAP_SEARCH_FOR_MS response containing the IMSI and current location area identification of the called MS to be returned to the VLR. The access connection status will be set accordingly (i.e. no RR-connection, in which case other values are not significant), and the macro terminates with successful outcome.

- the MS responds with a channel request containing an establishment cause which is not "answer to paging". The MSC sends a MAP_SEARCH_FOR_MS response primitive with user error "Busy Subscriber" before the macro terminates with unsuccessful outcome. This will give priority to the mobile originating request. Alternatively, as an implementation option, the MSC may treat this as a response to paging, which will give priority to the mobile terminating request.

- there is no response from the MS. The MSC sends a MAP_SEARCH_FOR_MS response primitive with user error "Absent Subscriber" before the macro terminates with unsuccessful outcome.

- the call handling connection or MAP transaction on which the call, SMS or unstructured SS transaction is waiting for delivery, is released before a response is received from the MS (indicated in the SDL by the input signal I-REL). The MAP transaction with the VLR will be released in this case by a MAP_U_ABORT request, and the unsuccessful macro termination will indicate transaction termination.

- the MAP transaction with the VLR may be released by receiving a MAP_U_ABORT or MAP_P_ABORT indication. The call handling connection or MAP transaction on which the call, SMS or unstructured SS transaction is waiting for delivery, is released (indicated in the SDL by the output signal I-REL), and the unsuccessful macro termination will indicate transaction termination.

Figure 21.3/1
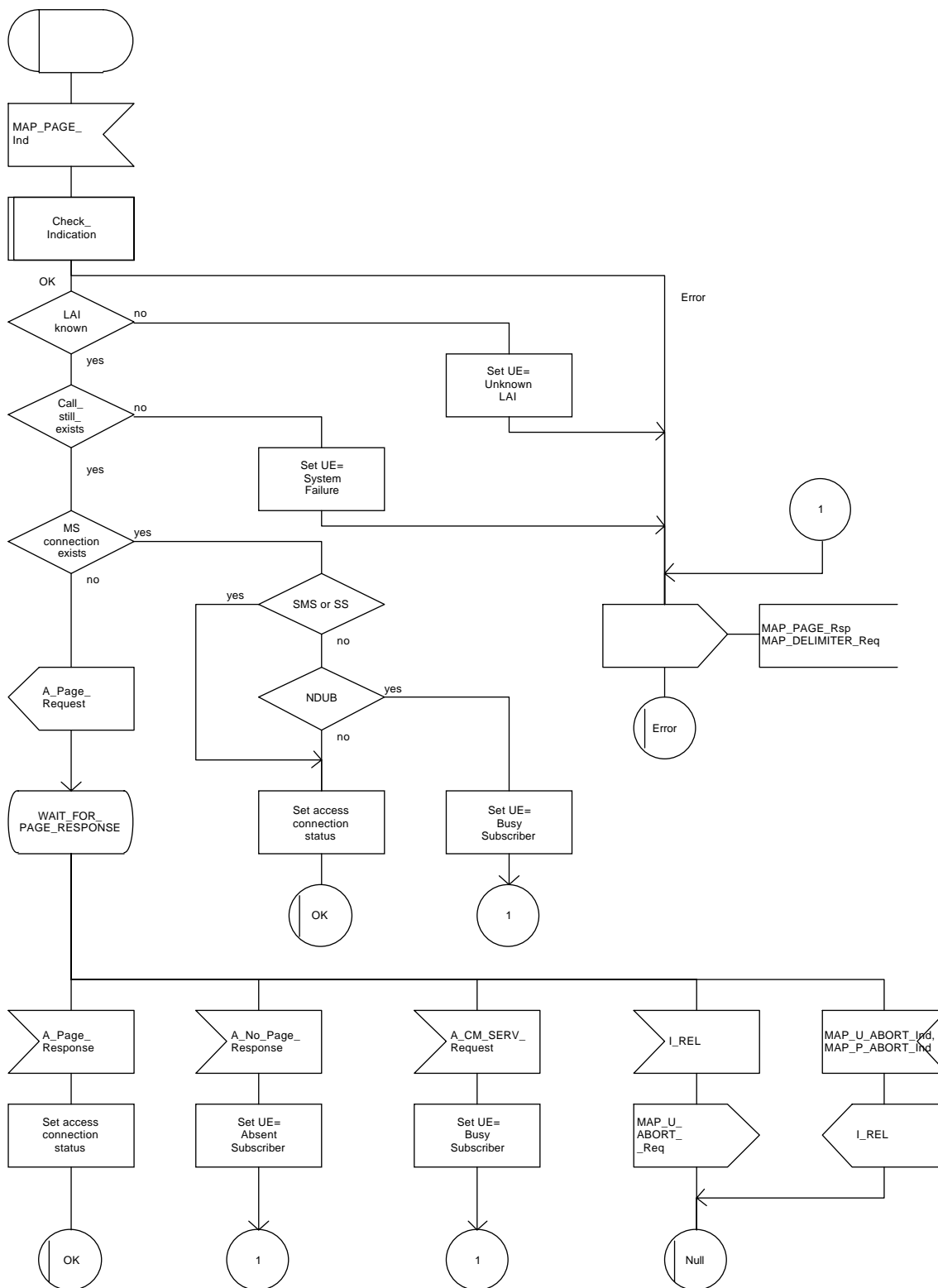
Figure 21.3/2
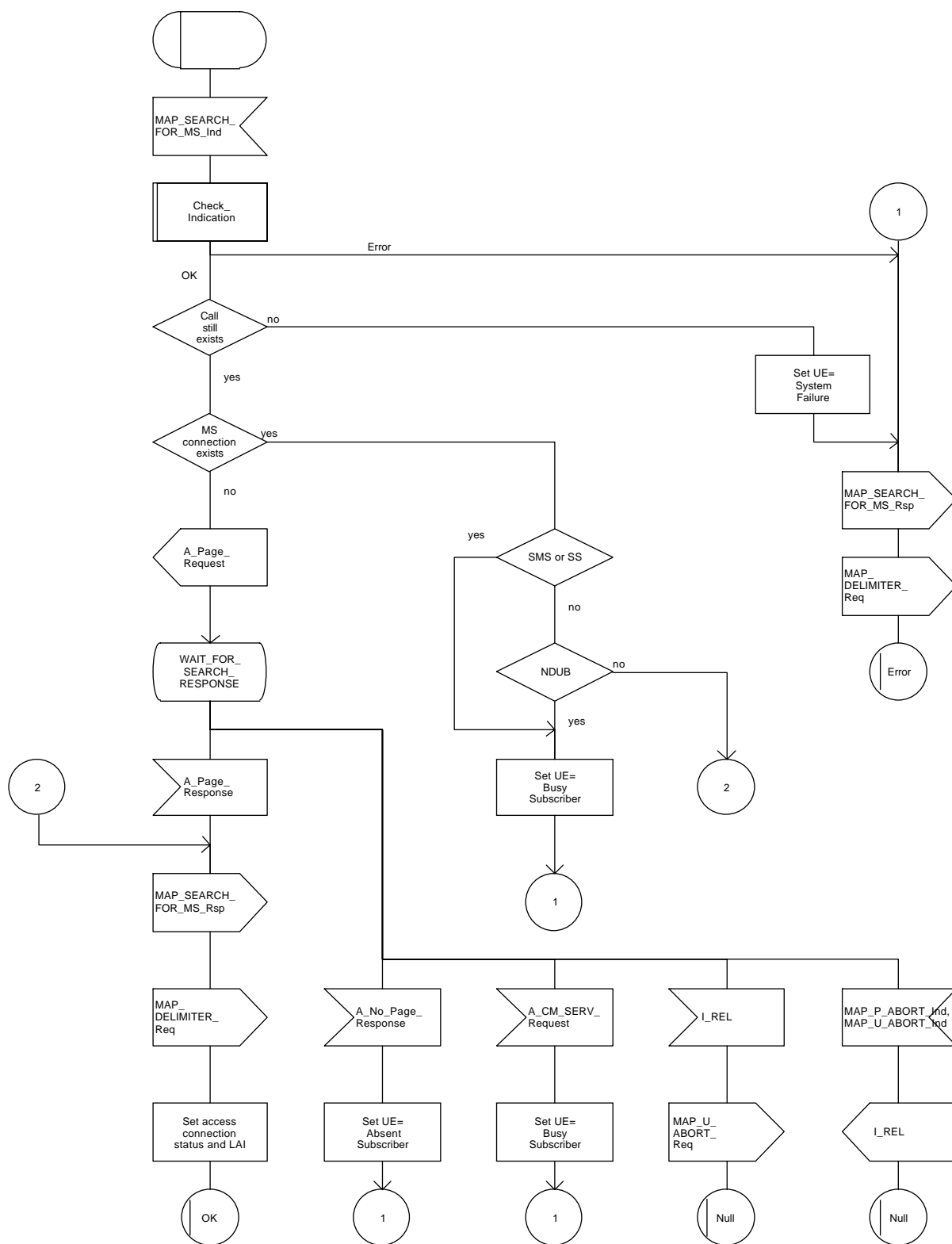
21.4 Macros for handling an Access Request

These macros are invoked when a MS accesses the network, e.g. to set up an outgoing call or when responding to paging. The macro handles identification and authentication of the mobile subscriber as well as invocation of security related features (see GSM 02.09).

21.4.1 Macro Process_Access_Request_MSC

This macro is invoked by any procedure receiving an access request from the MS, e.g. the page response at mobile terminating call set-up or the request for outgoing call set-up.

If no dialogue with the VLR exists (e.g. within the procedure for outgoing call set-up), the MSC will open a dialogue towards the VLR by sending a MAP_OPEN request without any user specific parameters.

In any case, the parameters received from the MS are mapped to a MAP_PROCESS_ACCESS_REQUEST request primitive, containing:

- the received subscriber identification (IMSI, TMSI) or - in case of emergency call set-up - an IMEI;

- the CM service type, indicating the type of request;

- the status of the access connection, i.e. whether a connection to this MS already exists and if so, whether it is already authenticated and ciphered;

- the current location area id of the MS; and

- the CKSN received from the MS.

If opening of the dialogue was required, the MSC will wait for the dialogue confirmation (see macro Receive_Open_Confirmation, subclause 21.1), leading either to:

- immediate unsuccessful exit from the macro, in case no dialogue is possible;

- reversion to MAP version one dialogue if indicated by the VLR. The macro terminates with unsuccessful outcome, as the complete dialogue will be covered by the version one procedure, so that no further action from the calling process is required;

- continuation as given below, if the dialogue is accepted by the VLR.

The MSC waits then for the MAP_PROCESS_ACCESS_REQUEST confirmation. In between, several other indications may be received from the VLR:

- the MSC may receive a MAP_PROVIDE_IMSI indication, handled by the macro Obtain_IMSI_MSC defined in subclause 21.8. In case of positive outcome, the procedure continues waiting for the MAP_PROCESS_ACCESS_REQUEST confirmation, else the macro terminates with unsuccessful outcome;

- the MSC may receive a MAP_AUTHENTICATE indication, handled by the macro Authenticate_MSC defined in subclause 21.5. In case of positive outcome, the procedure continues waiting for the MAP_PROCESS_ACCESS_REQUEST confirmation, else the macro terminates with unsuccessful outcome;

- the MSC may receive a MAP_TRACE_SUBSCRIBER_ACTIVITY indication, handled by the macro Trace_Subscriber_Activity_MSC defined in subclause 21.9;

- the MSC may receive a MAP_SET_CIPHERING_MODE indication, which will be stored for initiating ciphering later on;

- the MSC may receive a MAP_CHECK_IMEI indication, handled by the macro Check_IMEI_MSC defined in subclause 21.6. In case of positive outcome, the procedure continues waiting for the MAP_PROCESS_ACCESS_REQUEST confirmation, else the macro terminates with unsuccessful outcome;

- the MSC may receive a MAP_Obtain_IMEI indication, handled by the macro Obtain_IMEI_MSC defined in subclause 21.6. In case of positive outcome, the procedure continues waiting for the MAP_PROCESS_ACCESS_REQUEST confirmation, else the macro terminates with unsuccessful outcome;

- the MSC may receive a MAP_U_ABORT or MAP_P_ABORT indication, or a premature MAP_CLOSE indication from the VLR. In all these cases, the macro terminates with unsuccessful outcome, after sending the appropriate reject towards the MS (see GSM 09.10);

- the MSC may receive a MAP_NOTICE indication from the VLR. In this case, the dialogue towards the VLR is terminated by a MAP_CLOSE primitive, the appropriate reject is sent towards the MS (see GSM 09.10), and the macro terminates with unsuccessful outcome;

- the MSC may receive an indication for release of the radio path, in which case the dialogue towards the VLR will be terminated by a MAP_U_ABORT primitive, containing the diagnostic information Radio Channel Release.

When the MAP_PROCESS_ACCESS_REQUEST confirmation is received, the parameters of this primitive are checked first. In case of unsuccessful outcome of the service, the MAP User Error received is mapped onto the appropriate radio interface message (see GSM 09.10), before the macro terminates with unsuccessful outcome.

In case of positive outcome of the service, ciphering is initiated on the radio path, if this had been requested by the VLR (see above). Otherwise, if the access request was not triggered by a page response from the MS, the access request is accepted explicitly by sending a CM_Service_Accept message to the MS. If the access request was triggered by a page response from the MS then no CM Service Accept message is sent.

After ciphering has been initiated, the MSC will wait for the MAP_FORWARD_NEW_TMSI indication from the VLR. While waiting, the MSC may receive:

- a MAP_U_ABORT or MAP_P_ABORT indication, or a premature MAP_CLOSE indication from the VLR. In these cases, the macro terminates with unsuccessful outcome, after sending a release request towards the MS (see GSM 09.10);

- a MAP_NOTICE indication from the VLR. In this case, the dialogue towards the VLR is terminated by a MAP_CLOSE primitive, the appropriate reject is sent towards the MS (see GSM 09.10), and the macro terminates with unsuccessful outcome;

- an indication for release of the radio path, in which case the dialogue towards the VLR will be terminated by a MAP_U_ABORT primitive, containing the diagnostic information Radio Channel Release;

- a MAP_DELIMITER request from the VLR. This will be taken as a successful outcome of the macro (i.e. the VLR did not require TMSI reallocation), and it terminates successfully;

- an A_SETUP request from the MS. This will be saved for handling by the procedure which invoked the macro Process_Access_Request_MSC after the macro has terminated.

When the MAP_FORWARD_NEW_TMSI indication is received in the MSC, the TMSI Reallocation Command is sent to the MS, and the MSC waits for an acknowledgement from the MS. In case a positive acknowledgement is received, the MSC sends an empty MAP_FORWARD_NEW_TMSI response primitive to the VLR and terminates successfully. Else, the dialogue is terminated locally (MAP_CLOSE_Req with Release method Prearranged End) without any further action.

If the MSC receives an A_SETUP request while it is waiting for the TMSI acknowledgement from the MS, the A_SETUP is saved for handling by the procedure which invoked the macro Process_Access_Request_MSC after the macro has terminated.

If the dialogue is aborted by the VLR while waiting for the TMSI acknowledgement from the MS, the MSC regards the access request to be failed and terminates with unsuccessful outcome, after sending a release request towards the MS (see GSM 09.10).

Figure 21.4/1 (sheet 1 of 3)
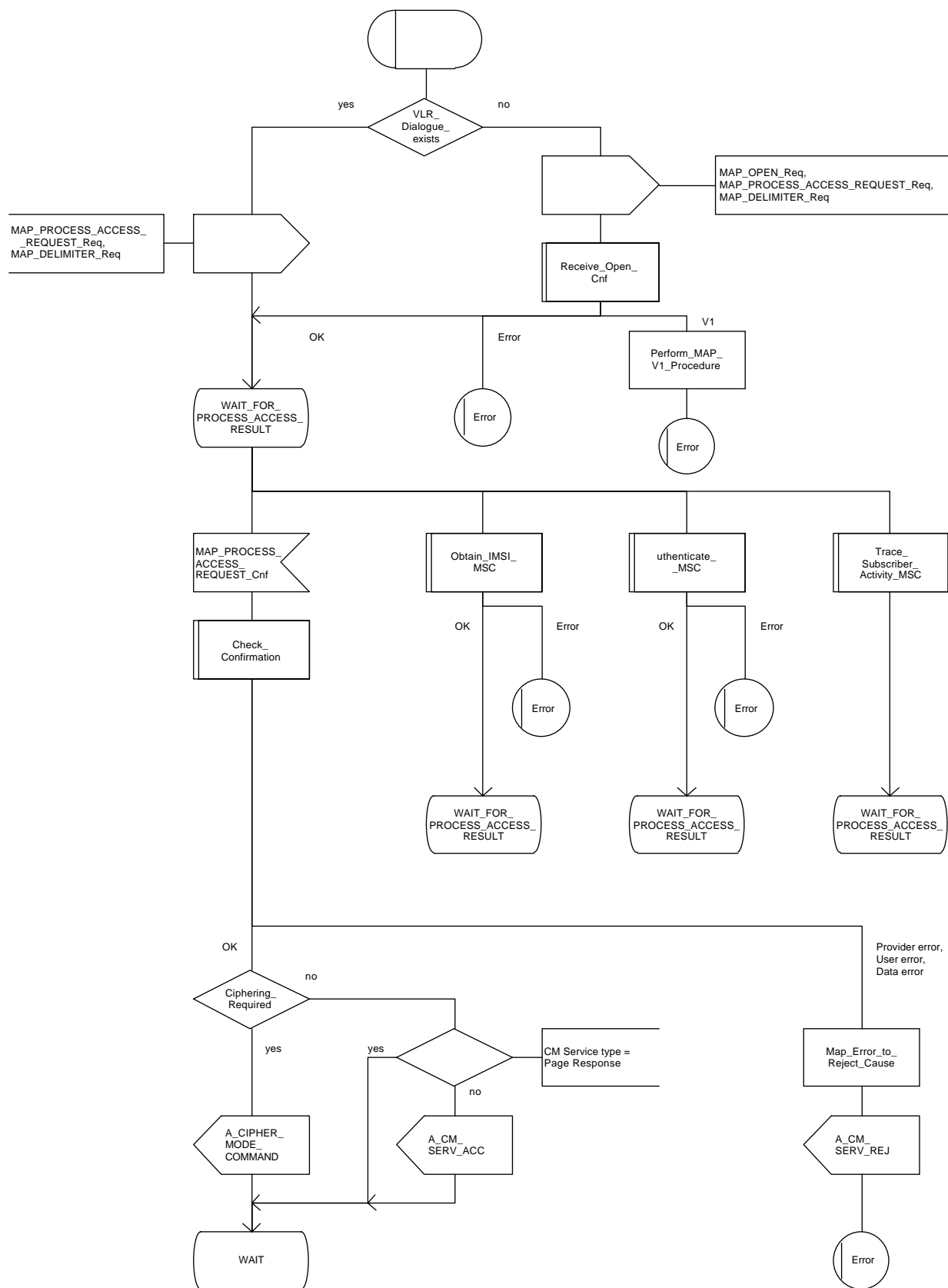
Figure 21.4/1 (sheet 2 of 3)
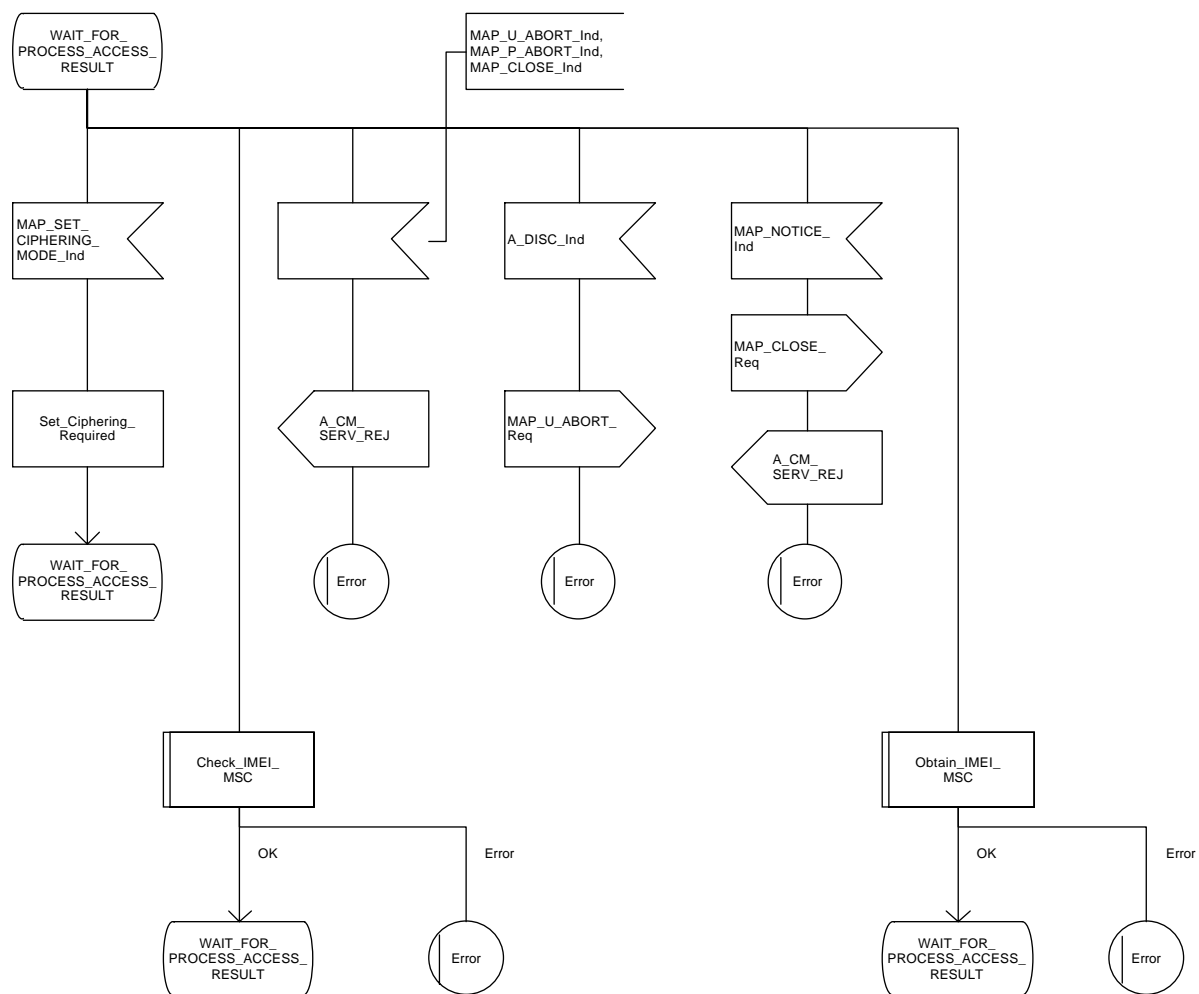
Figure 21.4/1 (sheet 3 of 3)
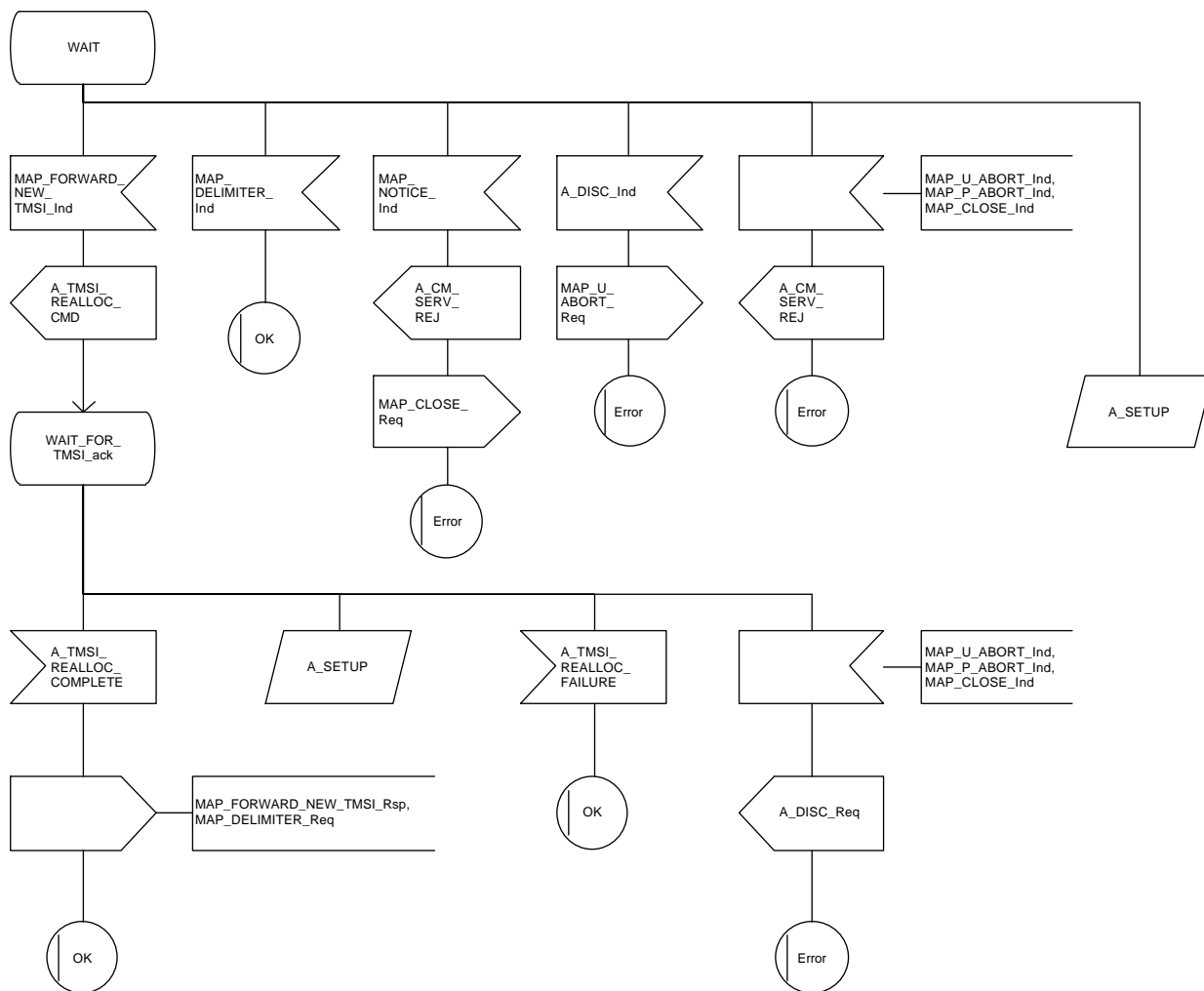
21.4.2 Macro Process_Access_Request_VLR

When the VLR receives a MAP_PROCESS_ACCESS_REQUEST indication, the VLR will check this indication first (macro Check_Indication, see subclause 21.2). In case of negative outcome, the macro will proceed with the error handling described below.

If the indication data are correct, it is checked first whether the subscriber identification (IMSI or TMSI) is known if included:

- if the identification is not known, the IMSI may be requested from the MS, described in the macro Identification_Procedure (see below) with outcome:
    - OK, if a IMSI known in the VLR has been received;

    - Error, if the VLR did not recognize the subscriber's identity. The macro will proceed with the error handling described below;

    - Aborted, if the transaction to the MSC is released. The macro will terminate immediately with unsuccessful.

In case the identity received is an IMEI, the error System Failure is set and the macro proceeds with the error handling described below.

> NOTE: Emergency Call with IMEI may be accepted within the error handling phase.

For a known subscriber the authentication check is performed next (see macro Authenticate_VLR, subclause 21.5), if required. If a negative result is received, the VLR proceeds on receipt of user error:

- illegal subscriber depending on the identity used for authentication;

    In case IMSI is already used or no new authentication attempt with IMSI shall not be performed (operator option), the error Illegal Subscriber is set and the macro proceeds with the error handling described below.

    If a new authentication attempt with IMSI shall be performed, the IMSI is requested from the MS (macro Obtain_IMSI_VLR, see subclause 21.8):

    - the authentication will be performed again if a IMSI known in the VLR is received;

    - the error Unidentified Subscriber is set and the macro proceeds with the error handling described below, if the IMSI received is unknown in VLR;

    - if the IMSI request procedure fails for any other reason, the error System Failure is set and the macro proceeds with the error handling described below;

    - if the dialogue has been aborted during the IMSI request, the macro terminates immediately with unsuccessful outcome;

- unknown subscriber by setting the error Unidentified Subscriber and proceeding with the error handling described below.

> NOTE: This can occur only in case of data inconsistency between HLR and VLR;

- procedure error by setting the error System Failure and proceeding with the error handling described below;

- null (i.e. the dialogue towards the MSC is terminated) by terminating immediately with unsuccessful outcome.

The MS access is accepted if no authentication is required or after successful authentication. Then, the indicator "Confirmed by Radio Contact" is set to "Confirmed". If the indicator "Location Information Confirmed in HLR" is set to "Not Confirmed", HLR updating will be started as an independent process (Update_Location_VLR, see subclause 16.1.1.6).

If the indicator "Confirmed by HLR" is set to "Not Confirmed", the error Unidentified Subscriber is set and the macro proceeds with the error handling described below.

If roaming is not allowed in the location area indicated in the Current Location Area Id parameter, the error Roaming Not Allowed qualified by the roaming restriction reason is set and the macro proceeds with the error handling described below.

In case roaming is allowed, the IMSI is set to attached and the process for notifying the HLR that the subscriber is present is started if required (Subscriber Present VLR, see subclause 21.10).

At next, tracing is invoked if required by the operator (macro Trace_Subscriber_Activity_VLR, see subclause 21.9). Thereafter,

> if ciphering is not required, IMEI checking is invoked if required by the operator (see macro Check_IMEI_VLR defined in subclause 21.6).
>
> > The error Illegal Equipment is set in case of unsuccessful outcome of the IMEI check, the subscriber is marked as detached and the macro proceeds with the error handling described below.
> >
> > The macro terminates immediately with unsuccessful outcome if the MSC dialogue has been released during the IMEI check.
> >
> > Else, the macro terminates successfully by returning the MAP_PROCESS_ACCESS_REQUEST response containing the IMSI to indicate acceptance of the MS access.
>
> if ciphering is required, the MAP_SET_CIPHERING_MODE request containing:
>
> > - the cipher mode indicating the cipher algorithm required; and
> > - the cipher key to be used;
>
> is sent to the MSC.

As a further operator option, IMEI checking may be performed next.

> The error Illegal Equipment is set in case of unsuccessful outcome of the IMEI check, the subscriber is marked as detached and the macro proceeds with the error handling described below.
>
> The macro terminates immediately with unsuccessful outcome if the MSC dialogue has been released during the IMEI check.
>
> Else, the macro terminates successfully by returning the MAP_PROCESS_ACCESS_REQUEST response containing the IMSI to indicate acceptance of the MS access.

IF no TMSI reallocation is required (again an operator option), the macro terminates thereafter. Else, TMSI reallocation is performed by sending a MAP_FORWARD_NEW_TMSI request, containing the new TMSI as parameter. The old TMSI will be frozen until an acknowledgement from the MS has been received. Before the macro terminates, the VLR will wait for the MAP_FORWARD_NEW_TMSI response, containing no parameters if reallocation has been confirmed by the MS, or a Provider Error, otherwise, in which case the old TMSI is kept frozen to avoid double allocation. In this case, both the old as the new TMSI are subsequently regarded valid when used by the MS.

Error handling

In case some error is detected during handling the access request, a respective error has been set. Before returning this error cause to the MSC in a MAP_PROCESS_ACCESS_REQUEST response, it need to be checked whether this access is for emergency call set-up, as this will require extra treatment.

If the CM Service type given in the MAP_PROCESS_ACCESS_REQUEST indication is emergency call set-up, it is checked whether EC set-up in the particular error situation is permitted (operator option). If so, it is checked whether the IMEI is required, and if so the IMEI is requested from the MS (macro Obtain_IMEI_VLR, see subclause 21.6).

> The macro will terminate immediately with unsuccessful outcome if the MSC transaction has been aborted during the IMEI retrieval.

> In case of an error reported back from IMEI retrieval, MAP_PROCESS_ACCESS_REQUEST response containing the error cause set previously is returned to the MSC, the dialogue is closed (MAP_CLOSE request indicating normal release) and the macro terminates with unsuccessful outcome.

When a subscriber identity required by the operator (IMSI or IMEI) is available, the user error set previously is deleted, the respective identity is returned in the MAP_PROCESS_ACCESS_REQUEST response to indicate acceptance of emergency call, and the macro terminates with successful outcome.

In all other cases, the MAP_PROCESS_ACCESS_REQUEST response containing the error cause set previously is returned to the MSC, the dialogue is closed (MAP_CLOSE request indicating normal release) and the macro terminates with unsuccessful outcome.

21.4.3 Macro Identification Procedure

This macro is invoked by the macro Process_Access_Request_VLR in case the subscribers identity is not known in the VLR.

If the identity received from the MS is an IMSI, the error Unidentified Subscriber will be set and reported back to the calling macro (to be sent in the MAP_PROCESS_ACCESS_REQUEST response). The same error is used in case a TMSI was received from the MS, but the operator does not allow open identification of the MS.

If open identification of the MS is allowed, the macro Obtain_IMSI_VLR is invoked, requesting the subscribers IMSI from the MS (see subclause 21.8), with outcome

> OK, in which case it is checked whether for the IMSI received there exists a subscriber record in the VLR. If so, the macro terminates successfully, else the error Unidentified Subscriber will be set and reported back to the calling macro.

> Error, in which case the error System Failure will be set and reported back to the calling macro.

> Aborted, i.e. the MSC transaction is released, in which the macro terminates accordingly.

Figure 21.4/2 (sheet 1 of 3)

Figure 21.4/2: Macro for processing on access request in VLR
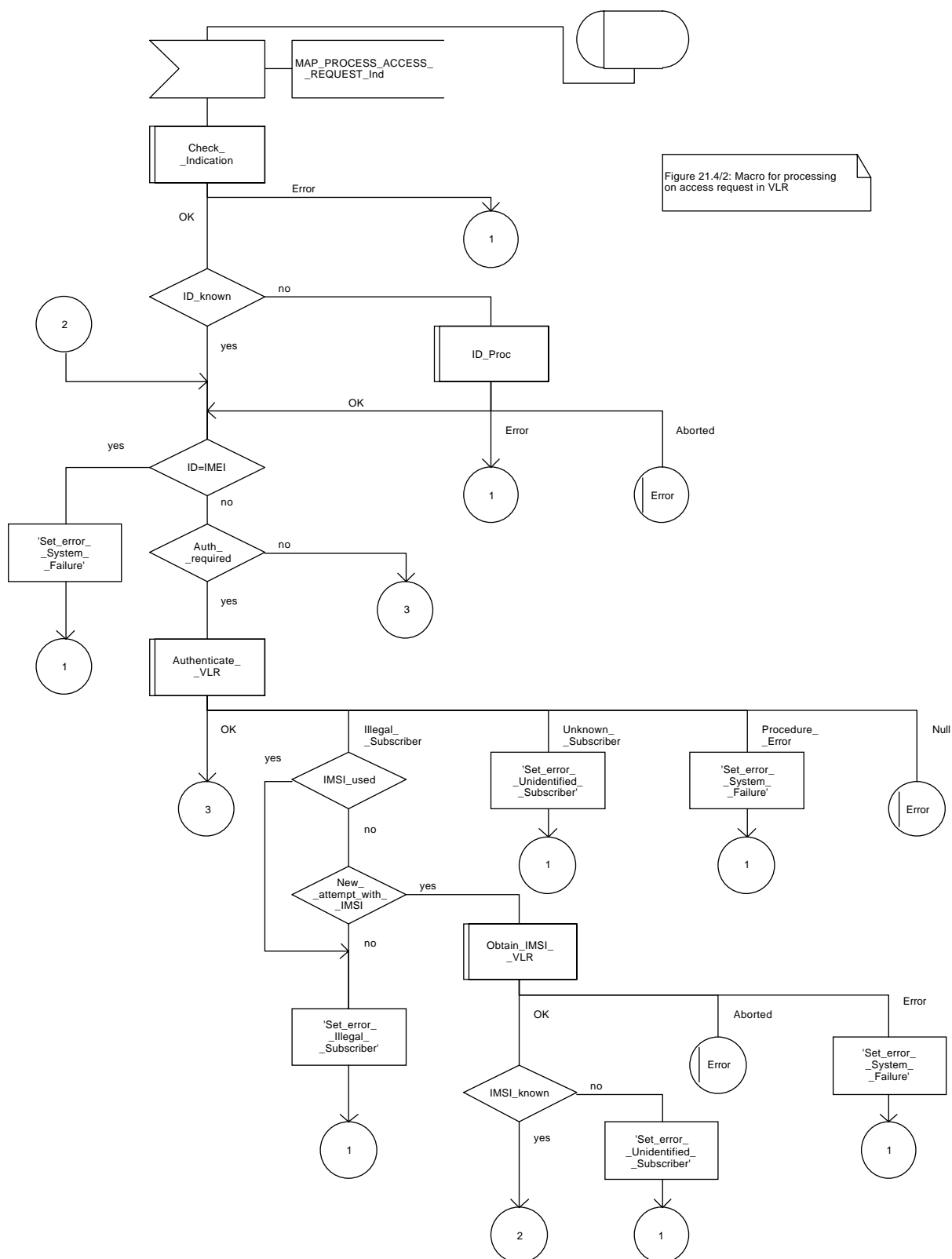
Figure 21.4/2 (sheet 2 of 3)
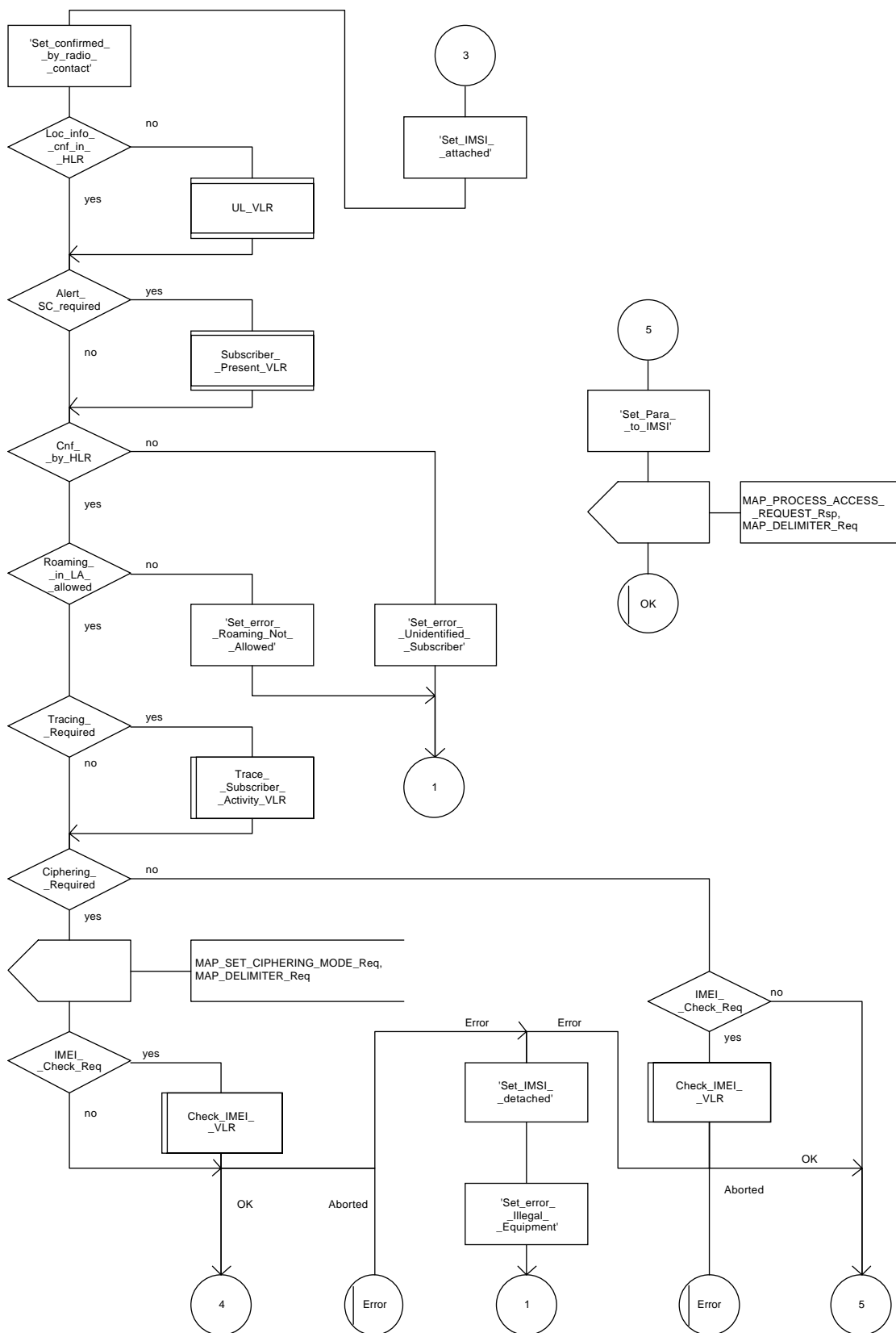
Figure 21.4/2 (sheet 3 of 3)
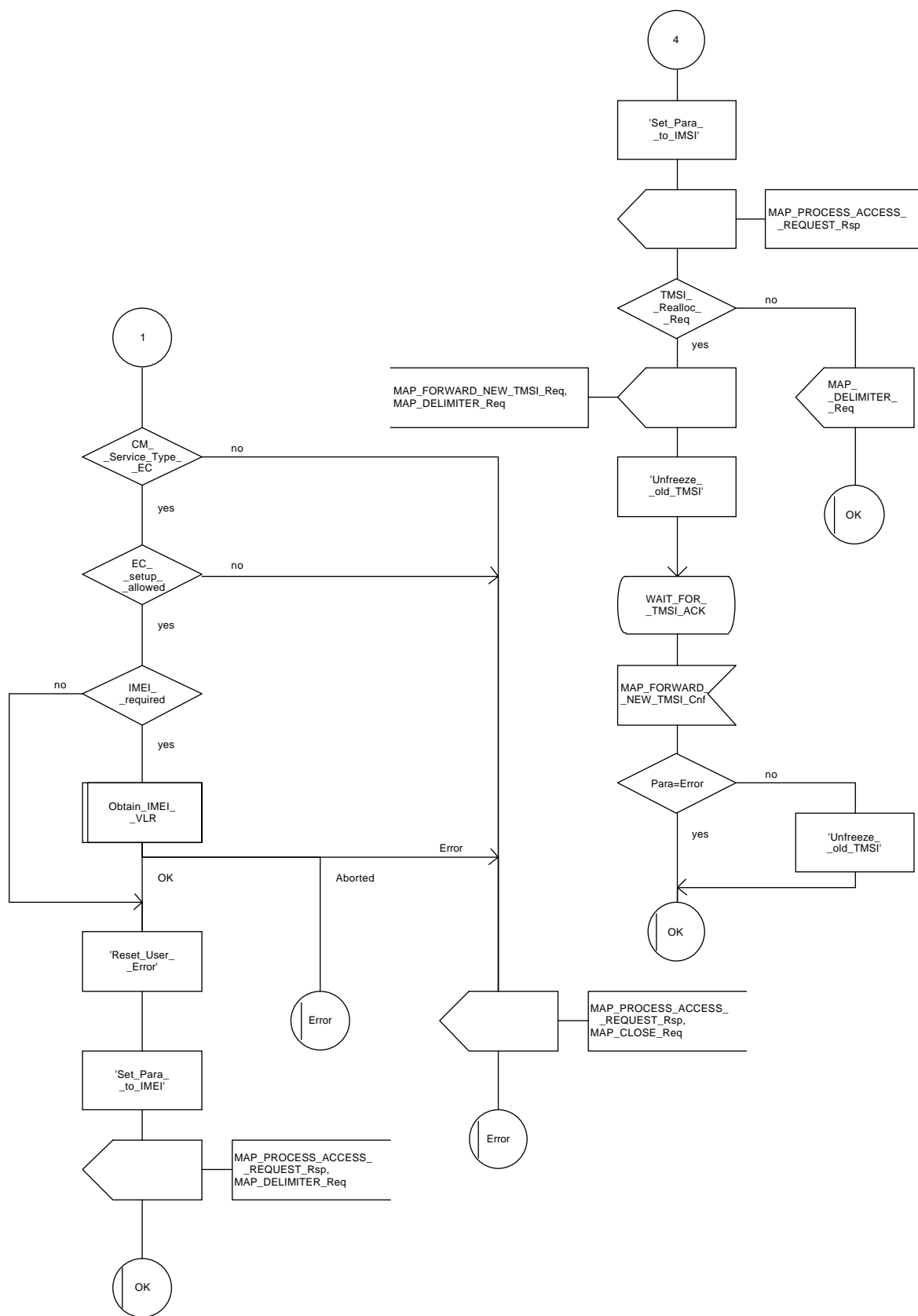
Figure 21.4/3
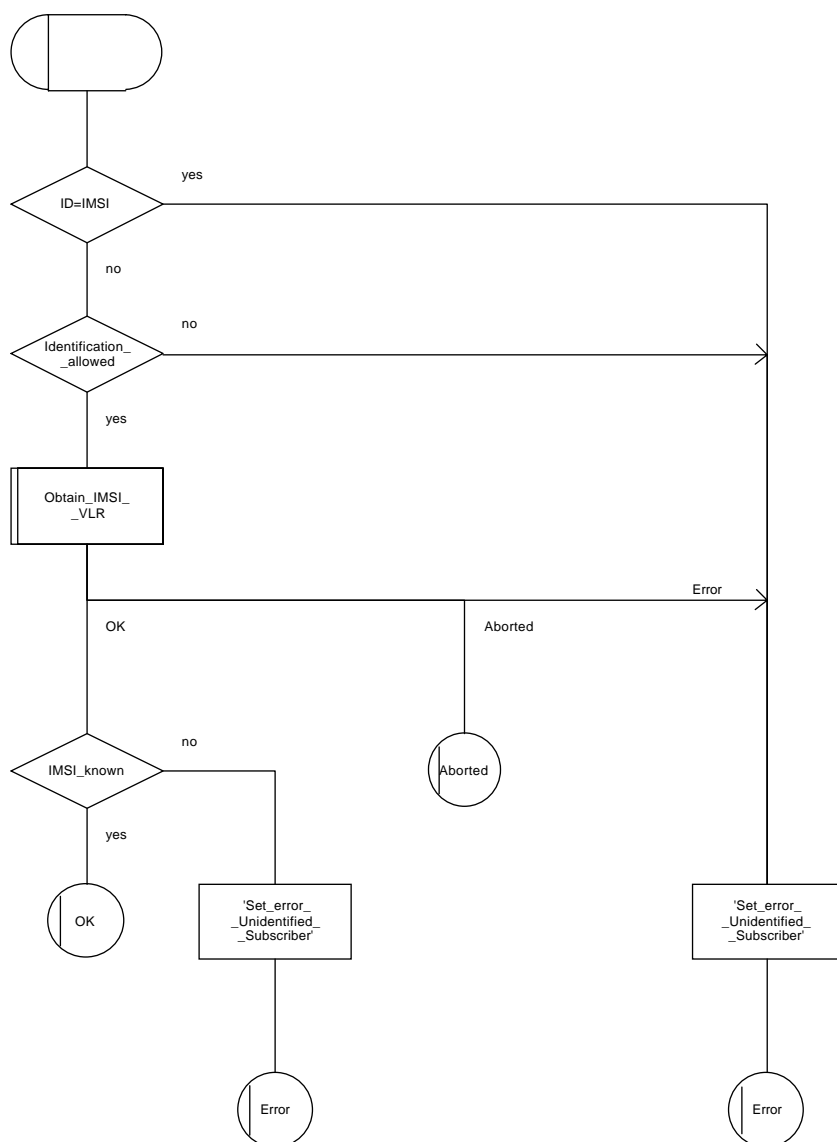
21.5 Authentication macros and processes

The following macros are used in the GSM network in order to enable authentication of a mobile subscriber.

21.5.1 Macro Authenticate_MSC

This macro is used by the MSC to relay a request for authentication transparently from the VLR to the MS, wait for a response from the MS and to relay the response from the MS back to the VLR. If, while the MSC is waiting for the authentication response, the air interface connection is released or a MAP_U_ABORT, MAP_P_ABORT or MAP_CLOSE indication is received from the VLR, then necessary connections are released and the "Error" exit is used. The macro is described in figure 21.5/1.

21.5.2 Macro Authenticate_VLR

This macro is used by the VLR to control the authentication of a subscriber. The macro proceeds as follows:

- if there are not enough authentication triplets in the VLR to perform the authentication, then the macro "Obtain_Authent_Para_VLR" described below is invoked. If this macro fails, then the corresponding error (Unknown Subscriber or Procedure Error) is returned to the calling process;

- if there are enough authentication triplets in the VLR, or the Obtain_Authent_Para_VLR macro was successful, then a MAP_AUTHENTICATE request is sent to the MSC. This request contains the RAND and CKSN parameters as indicated in the service description;

- the VLR then waits for a response from the MSC;

- if a MAP_U_ABORT, MAP_P_ABORT or MAP_CLOSE indication is received from the MSC in this wait state, the VLR checks whether authentication sets are available. If no sets are available the process Obtain_Authent_Sets_VLR is invoked to fetch authentication sets from the HLR. The "Null" exit is then used;

- if a MAP_NOTICE indication is received from the MSC in this wait state, the VLR closes the dialogue with the MSC, then checks whether authentication sets are available. If no sets are available the process Obtain_Authent_Sets_VLR is invoked to fetch authentication sets from the HLR. The "Null" exit is then used;

- if a MAP_AUTHENTICATE confirmation is received by the VLR, it checks whether the received Signed Result (SRES) is identical to the stored one (see GSM 03.20). If this is not the case, the "Illegal Subscriber" exit is used. If the SRES values are identical, then the "OK" exit is used;

- before exit, the VLR may fetch a new set of triplets from the HLR. This is done by initiating a separate Obtain_Authent_Sets_VLR process described below.

The macro is described in figure 21.5/2.

21.5.3 Process Obtain_Authentication_Sets_VLR

This process is initiated by the VLR to fetch triplets from a subscriber's HLR in a stand-alone, independent manner. The Obtain_Authent_Para_VLR macro described below is simply called; the process is described in figure 21.5/3.

21.5.4   Macro Obtain_Authent_Para_VLR

This macro is used by the VLR to request authentication triplets from the HLR. The macro proceeds as follows:

- a connection is opened, and a MAP_SEND_AUTHENTICATION_INFO request sent to the HLR;

- if the HLR indicates that a MAP version 1 dialogue is to be used, the VLR performs the equivalent MAP version 1 dialogue. which can return a positive result containing authentication sets, an empty positive result, or an error;

- if the dialogue opening fails, the "Procedure Error" exit is used. Otherwise, the VLR waits for the response from the HLR;

- if a MAP_SEND_AUTHENTICATION_INFO confirmation is received from the HLR, the VLR checks the received data.

One of the following positive responses may be received from a MAP version 1 or MAP version 2 dialogue with the HLR:

- Authentication triplets, in which case the outcome is successful;

- Empty response, in which case the VLR may re-use old triplets, if allowed by the PLMN operator.

If the VLR cannot re-use old triplets (or no such triplets are available) then the "Procedure Error" exit is used.

If the outcome was successful or re-use of old parameters in the VLR is allowed, then the "OK" exit is used.

If an "Unknown Subscriber" error is included in the MAP_SEND_AUTHENTICATION_INFO confirm or is returned by the MAP version 1 dialogue, then the "Unknown Subscriber" exit is used.

- if a MAP-U-ABORT, MAP_P_ABORT, MAP_NOTICE or unexpected MAP_CLOSE service indication is received from the MSC, then open connections are terminated, and the macro takes the "Null" exit;

- if a MAP-U-ABORT, MAP_P_ABORT or unexpected MAP_CLOSE service indication is received from the HLR, then the VLR checks whether old authentication parameters can be re-used. If old parameters cannot be re-used the macro takes the "Procedure Error" exit; otherwise it takes the "OK" exit;

- if a MAP_NOTICE service indication is received from the HLR, then the dialogue with the HLR is closed. The VLR then checks whether old authentication parameters can be re-used. If old parameters cannot be re-used the macro takes the "Procedure Error" exit; otherwise it takes the "OK" exit.

The macro is described in figure 21.5/4.

21.5.5   Process Obtain_Auth_Sets_HLR

Opening of the dialogue is described in the macro Receive_Open_Ind in subclause 21.1, with outcomes:

-        reversion to version one procedure;

-        procedure termination; or

-        dialogue acceptance, with proceeding as below.

This process is used by the HLR to obtain authentication triplets from the AuC, upon request from the VLR. The process acts as follows:

-        a MAP_SEND_AUTHENTICATION_INFO indication is received by the HLR;

-        the HLR checks the service indication for errors. If any, they are reported to the VLR in the MAP_SEND_AUTHENTICATION_INFO response. If no errors are detected, authentication triplets are fetched from the AuC. Further details are found in GSM 03.20;

-        if errors are detected they are reported to the VLR in the MAP_SEND_AUTHENTICATION_INFO response. Otherwise the authentication triplets are returned.

The process is described in figure 21.5/5.

Figure 21.5/1

Figure 21.5/2

Figure 21.5/3

Macro Obtain_Authent_Para_VLR   215_41(2)

Figure 21.5/4: Macro to obtain authentication parameters from the HLR to the VLR

MAP_OPEN_Req
MAP_SEND_AUTHENTICATION_INFO_Req
MAP_DELIMITER_Req

Receive_Open_Cnf — Section 21.1

OK   V1   Error

WAIT_FOR_AUTHENTICATION_INFO   Perform MAP V1 Dialogue   Procedure_Error

MAP_SEND_AUTHENTICATION_INFO_Cnf

Positive result — yes

no

Check confirmation — Figure 21.2/2

Empty result — no

OK   Provider Error, Data Error   User Error

Empty result — yes   Re-use old parameters — yes

no   no

Error= Unknown Subscriber — no   OK

yes   yes

Re-use old parameters — yes

no

OK   Procedure_Error   Unknown_Subscriber   Procedure_Error

Figure 21.5/4 (sheet 2 of 2)

Macro Obtain_Authent_Para_VLR
215_42(2)

Figure 21.5/4: Macro to obtain authentication parameters from the HLR to the VLR

WAIT_FOR_
AUTHENTICATION_
INFO

MAP_P_ABORT_ind,
MAP_U_ABORT_ind,
MAP_CLOSE_ind
from MSC

MAP_U_
ABORT_req

To HLR

Null

MAP_
NOTICE_
ind

From MSC

MAP_
CLOSE_
req

To MSC

MAP_U_
ABORT_req

To HLR

Null

MAP_P_ABORT_ind,
MAP_U_ABORT_ind,
MAP_CLOSE_ind
from HLR

MAP_
NOTICE_
ind

From HLR

MAP_
CLOSE_
Req

To HLR

Re-use old
parameters

no

yes

OK

Procedure_Error
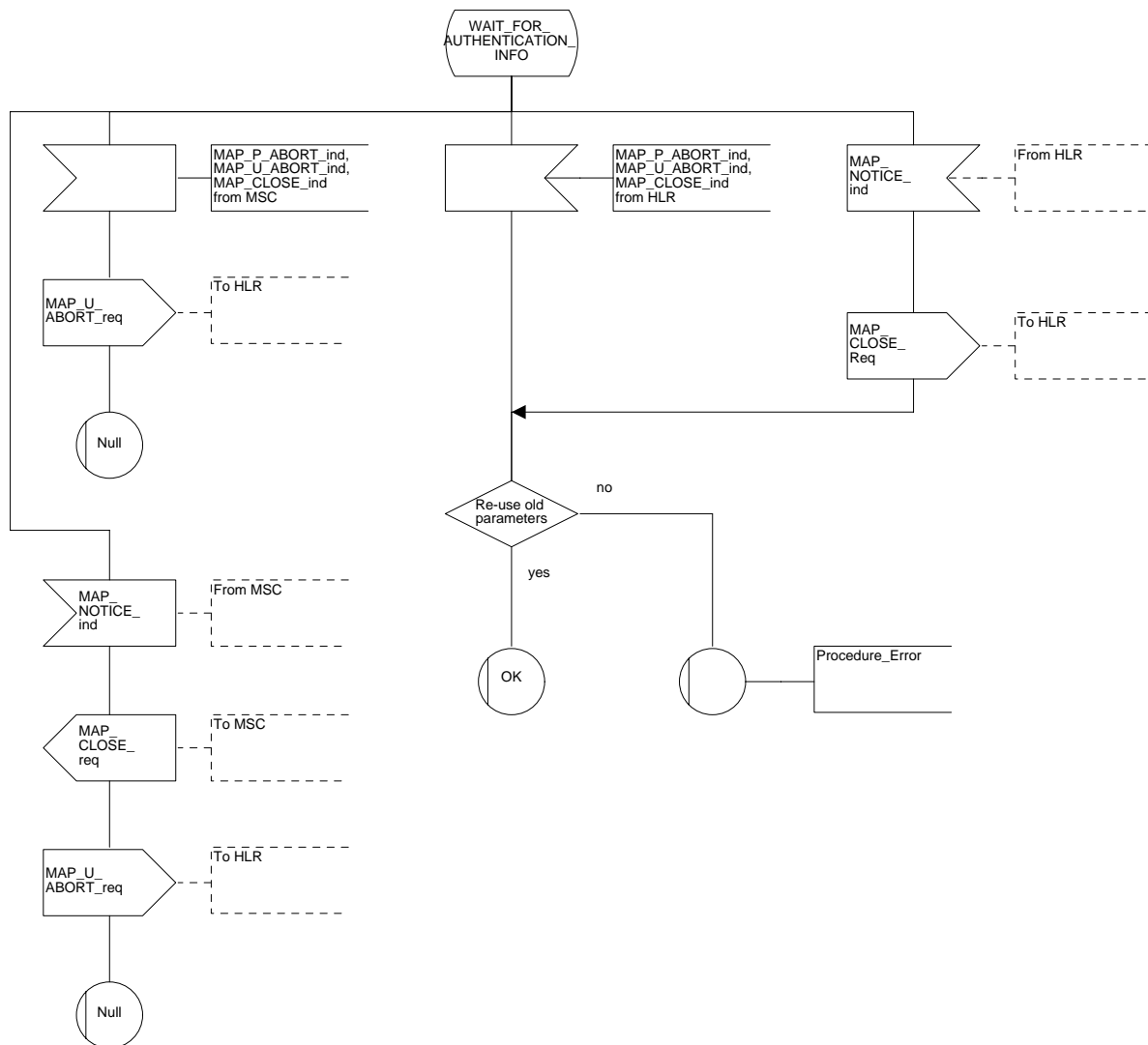
Figure 21.5/5
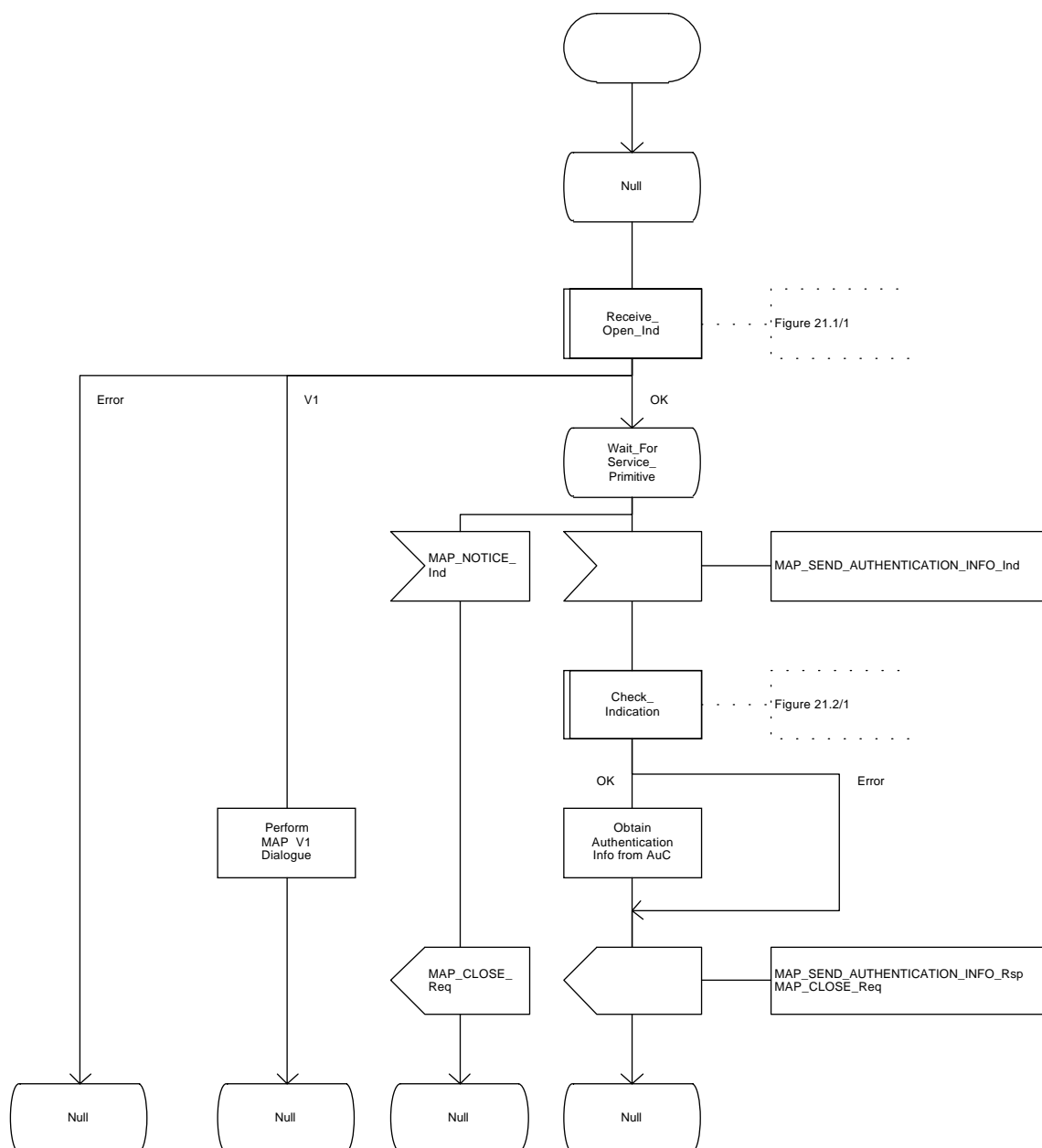
21.6   IMEI Handling Macros

The following macros are used in the GSM network in order to enable handling and checking of the mobile equipment identity.

21.6.1   Macro Check_IMEI_MSC

This macro is used by the MSC to receive a request from the VLR, relay it to the EIR, and pass the result from the EIR back to the VLR. The macro proceeds as follows:

-       a MAP_CHECK_IMEI service indication containing only the Invoke Id is received from the VLR;

-       if the IMEI is not available in the MSC, it is requested from the MS using the IDENTITY REQUEST message;

-       if the MS releases the radio resources, a MAP_U_ABORT request indicating "Application procedure Cancellation" is sent to the VLR, and the "Error" exit of the macro is used;

-       when the IMEI is known, a connection is set up towards the EIR, and a MAP_CHECK_IMEI service request is sent including the IMEI;

-       if the opening of the dialogue fails, a System Failure is reported to the VLR. Otherwise, the MSC waits for a response from the EIR;

-       when the MAP_CHECK_IMEI service confirm is received, it is checked for errors. Any errors discovered in the MSC lead to the System Failure error to be reported to the VLR in the MAP_CHECK_IMEI response. Any errors reported from the EIR are sent directly to the VLR in the MAP_CHECK_IMEI service response. If no errors are detected by or reported to the MSC, the IMEI is added to the MAP_CHECK_IMEI service response returned to the VLR. The "OK" exit is used in all cases;

-       if a MAP_P_ABORT, MAP_U_ABORT, MAP_CLOSE or MAP_NOTICE service indication is received from the EIR, the MSC closes the transaction with the EIR (if necessary), reports a System Failure error back to the VLR in the MAP_CHECK_IMEI response, and uses the macro's "OK" exit;

-       if a MAP_P_ABORT, MAP_U_ABORT, MAP_CLOSE or MAP_NOTICE indication is received from the VLR, the MSC closes the transaction with the VLR (if necessary) and aborts the connections towards the EIR and the MS; the macro takes the "Error" exit.

If the dialogue with the EIR drops back to version 1, the result or error returned by the EIR is checked. The use of the "Check_Confirmation" macro in the SDL diagram indicates that the checks carried out on the result returned by the EIR in a MAP v1 dialogue are functionally equivalent to those carried out on the parameters of the MAP_CHECK_IMEI confirm received from the EIR in a MAP v2 dialogue.

The macro is described in figure 21.6/1.

21.6.2   Macro Check_IMEI_VLR

This macro is used by the VLR to control the check of a mobile equipment's IMEI. The macro proceeds as follows:

-       a MAP_CHECK_IMEI service request is sent to the MSC, including only the Invoke Id;

-       the VLR then waits for the response from the MSC;

- if a MAP_CHECK_IMEI service confirm including either:
  - the IMEI and the Equipment Status; or
  - an error;
  is received, the VLR checks whether the response requires that an alarm be generated on the Operation and Maintenance interface. The criteria for such alarms are PLMN operator dependent;

- the VLR then checks whether the response from the MSC means that service is granted to the MS. The criteria for granting service depending on the equipment status or errors received in the MAP_CHECK_IMEI service response are also PLMN operator dependent;

- if a MAP_P_ABORT, MAP_U_ABORT, MAP_CLOSE or MAP_NOTICE indication is received from the MSC, then the MSC connection is closed (if necessary) and the macro takes the "Aborted" exit.

The macro is described in figure 21.6/2.

21.6.3   Process Check_IMEI_EIR

This process is used by the EIR to obtain the status of a piece of mobile equipment, upon request from the MSC. The process acts as follows:

- a MAP_OPEN service indication is received (macro Receive_Open_Ind, subclause 21.1.1). If the dialogue opening fails, the process terminates;

- otherwise, a MAP_CHECK_IMEI indication is received by the EIR, containing the IMEI to be checked;

- the EIR checks the service indication for errors. If there are any, they are reported to the MSC in the MAP-CHECK_IMEI response. If no errors are detected, the EIR data base function is interrogated for the status of the given equipment. Further details are found in GSM 02.16;

- the status of the equipment (white-listed, grey-listed, black-listed or unknown) is returned to the MSC in the MAP_CHECK_IMEI service response;

- if a MAP_U_ABORT, MAP_P_ABORT, MAP_NOTICE or MAP_CLOSE indication is received from the MSC at any time during this process, the process in the EIR terminates.

The process is described in figure 21.6/3.

21.6.4   Macro Obtain_IMEI_MSC

This macro is used by the MSC to respond to a request from the VLR to provide the IMEI. The macro proceeds as follows:

- a MAP_OBTAIN_IMEI service indication containing only the Invoke Id is received from the VLR;

- if the IMEI is not available in the MSC, it is requested from the MS using the IDENTITY REQUEST message;

- when the IMEI is known, it is returned to the VLR in the MAP_OBTAIN_IMEI service response. The macro terminates at the "OK" exit;

- if the IMEI cannot be obtained by the MSC, the System Failure error is reported back to the VLR in the MAP_OBTAIN_IMEI service response. The macro terminates at the "OK" exit;

- if a MAP_P_ABORT, MAP_U_ABORT or MAP_CLOSE indication is received from the VLR, the macro terminates at the "Error" exit.

The macro is described in figure 21.6/4.

21.6.5   Macro Obtain_IMEI_VLR

This macro is used by the VLR to obtain the IMEI from the MSC, e.g. to enable handling of emergency calls in case of authentication failure (in which case the IMEI may be used by some operators as an alternative to the IMSI). It proceeds as follows:

-       the MAP_OBTAIN_IMEI service request is sent to the MSC, including only the Invoke Id;

-       the VLR then waits for the response from the MSC;

-       if the IMEI is received in the MAP_OBTAIN_IMEI service response, the macro terminates at the "OK" exit;

-       if the System Failure error is reported in the MAP_OBTAIN_IMEI service response, the "Error" exit is used;

-       if the MSC terminates the dialogue using a MAP_P_ABORT, MAP_U_ABORT, MAP_CLOSE or MAP_NOTICE service indication, the necessary connections are released, and the "Aborted" exit is used for termination of the macro.

The macro is shown in figure 21.6/5.

Figure 21.6/1 (sheet 1 of 2)
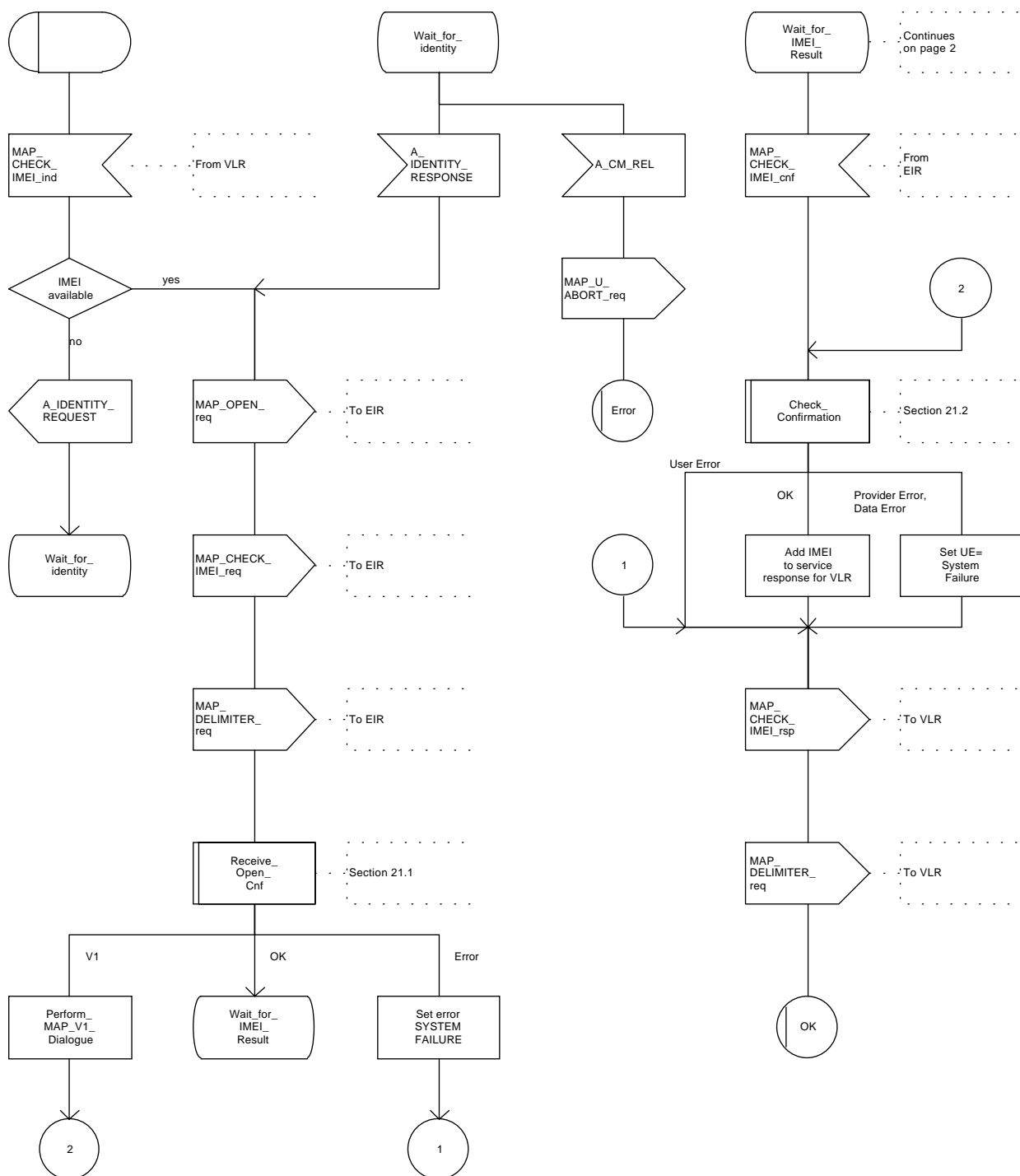
Figure 21.6/1 (sheet 2 of 2)
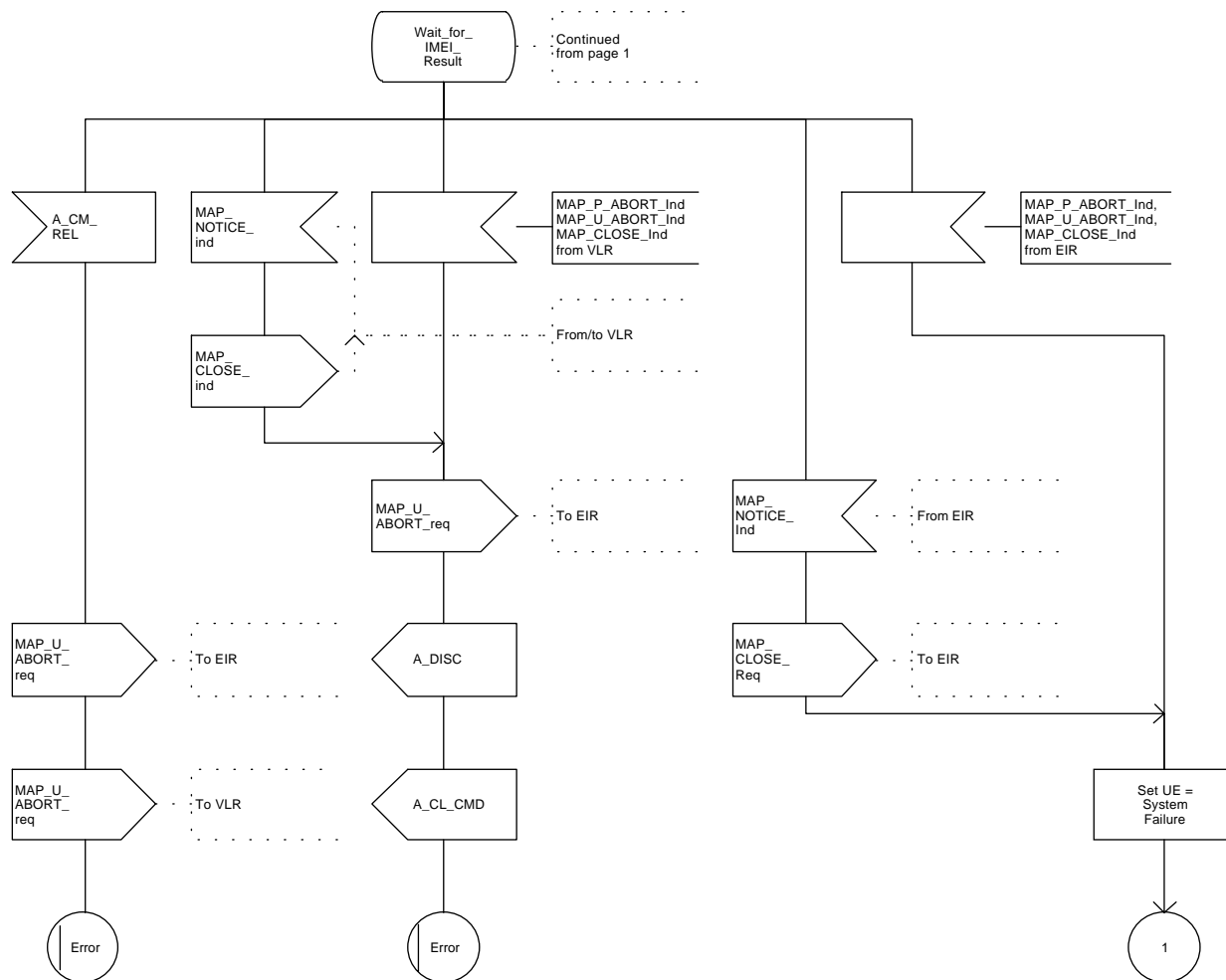
Figure 21.6/2
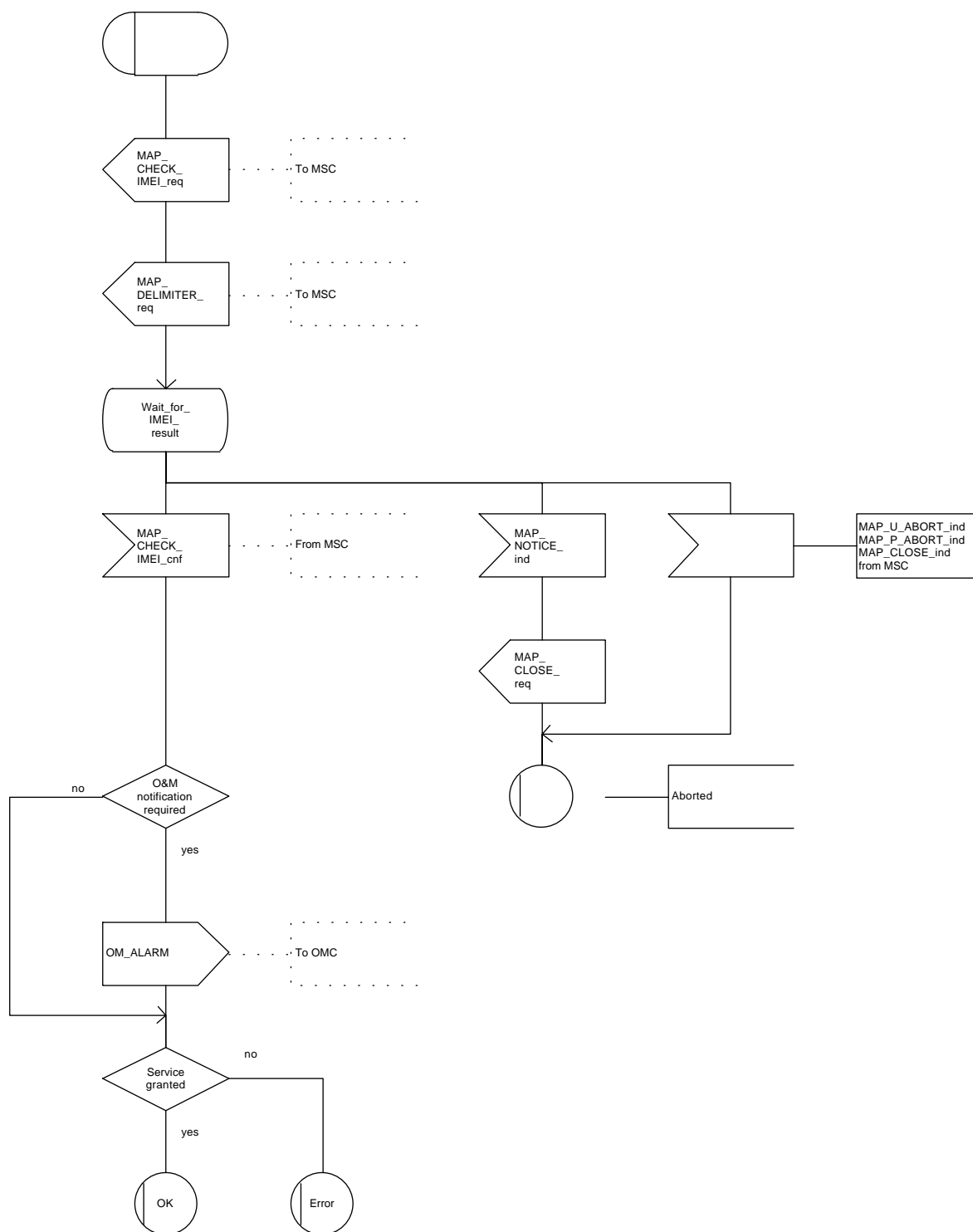
Figure 21.6/3
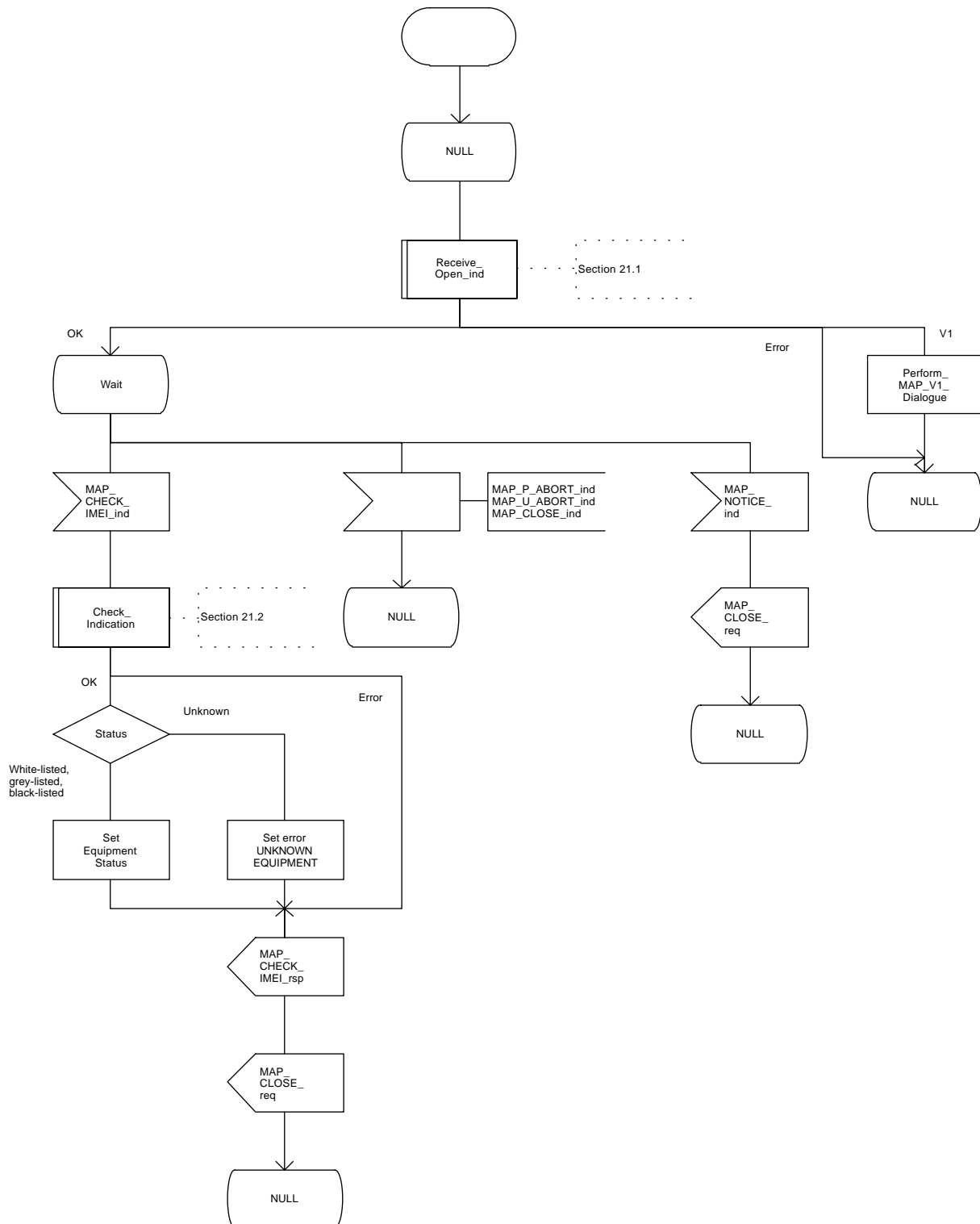
Figure 21.6/4
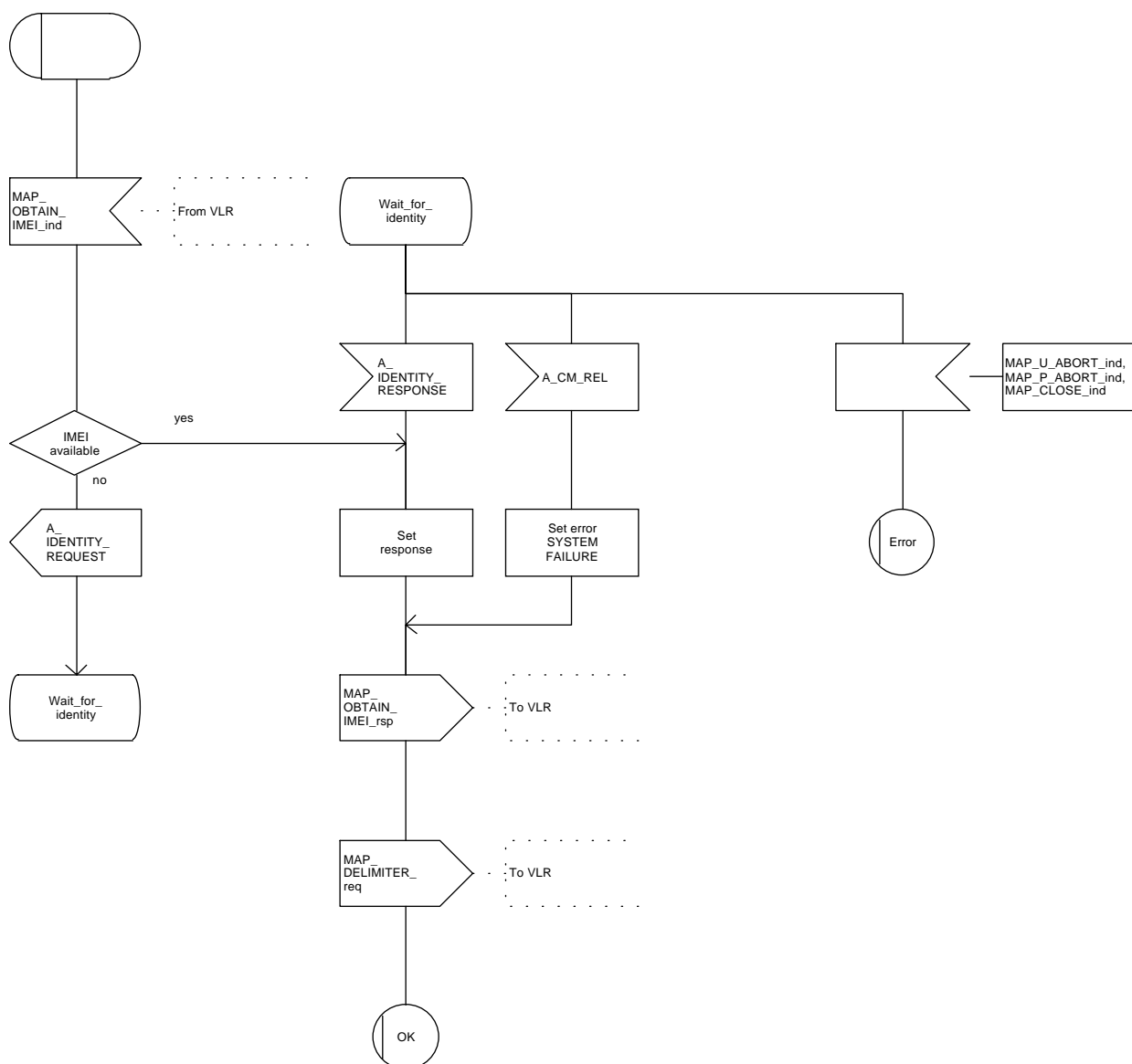
Figure 21.6/5
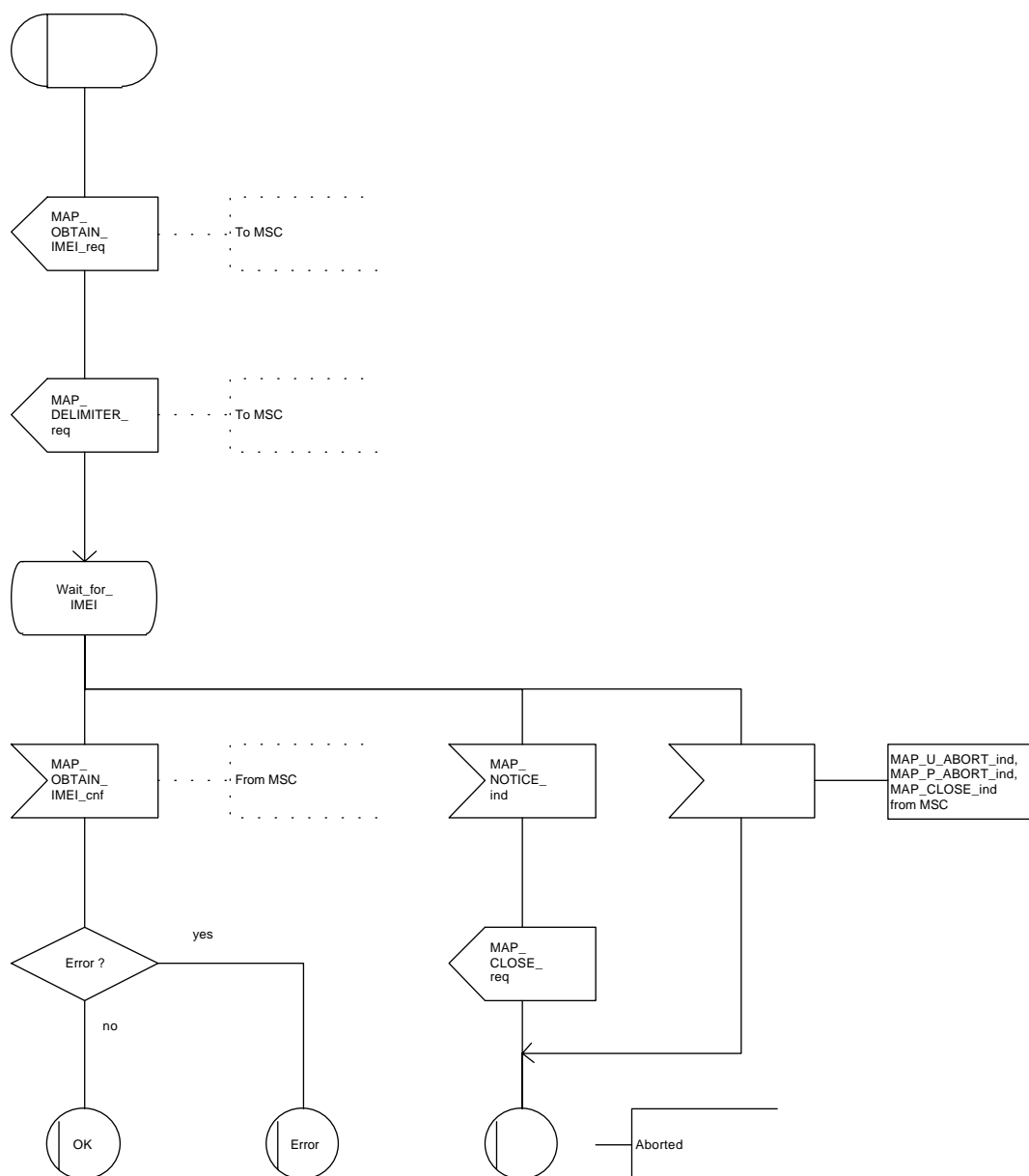
21.7 Insert Subscriber Data Macros

21.7.1 Macro Insert_Subs_Data_VLR

This macro describes the reception of the InsertSubscriberData service indication. This macro is used by any procedure that triggers the reception of subscriber data (e.g. Update Location or Restore Data).

If the VLR does not support any basic or supplementary service or the network feature Operator Determined Barring, or there is a problem with Regional Subscription Data then it reports it to the HLR.

If the entire MSC area is restricted due to regional subscription this is reported to the HLR.

The SDL diagram is shown in figure 21.7/1.

Figure 21.7/1
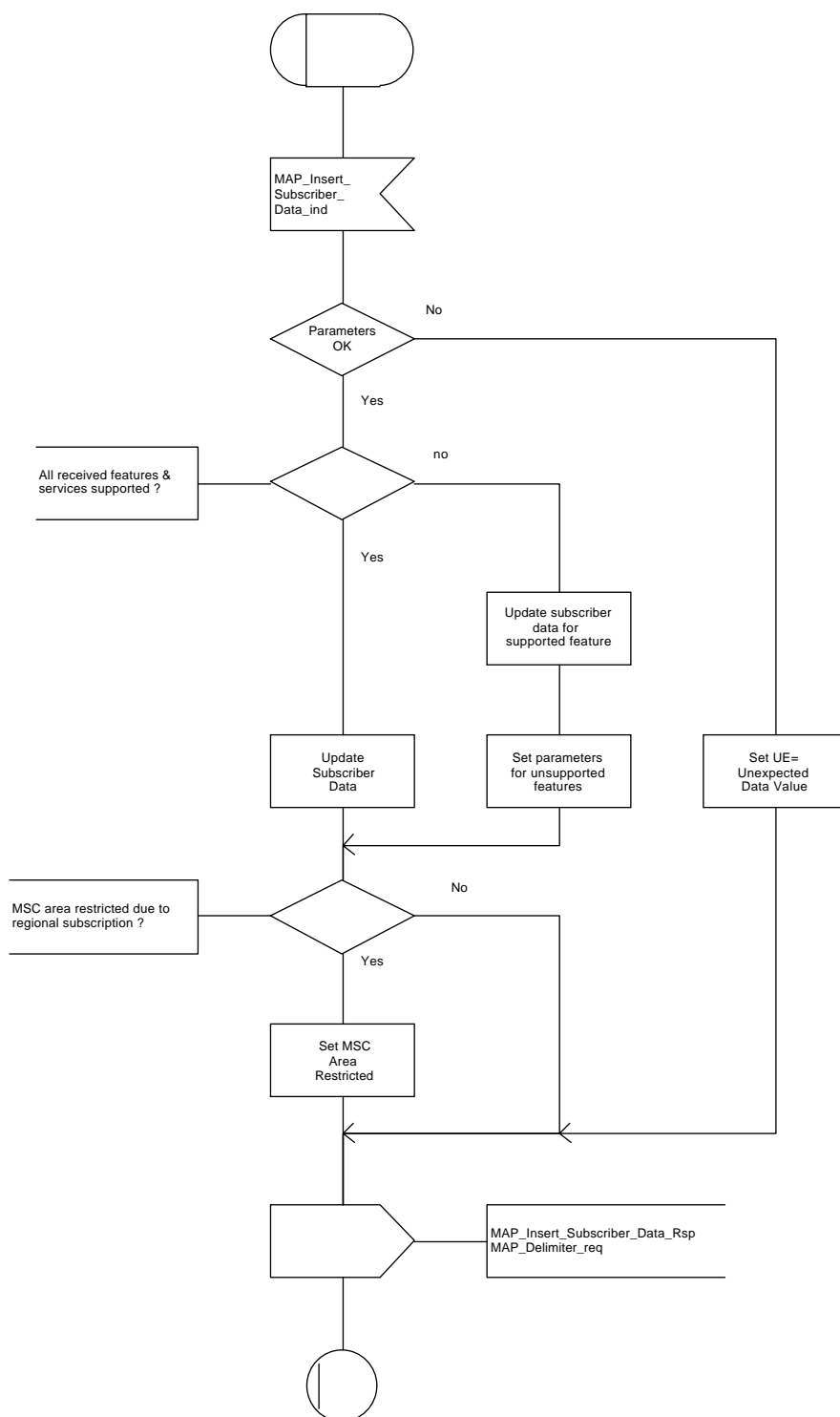
21.7.2 Process Insert_Subs_Data_Stand_Alone_HLR

This process is used by HLR to transfer subscriber data to VLR in a stand alone mode, i.e. in its own dialogue. this is done whenever a change of subscriber data is performed either by the operator or by the subscriber and this change has to be reported to VLR.

The process, after opening the dialogue with VLR, sends as many requests of the InsertSubscriberData service as necessary to transfer the subscriber data. The call to the process "Send_Insert_Subs_Data" (see subclause 21.7.4) is meant to describe two possible behaviours of the HLR when more than one service request has to be sent:

- either the HLR handles the requests and the confirmations in parallel; or

- the HLR sends every request after receiving the confirmation to the previous one.

The macro "Wait_for_Insert_Subs_Data_Cnf" (see subclause 21.7.3) is also called in order to handle every single confirmation.

If the result of a primitive received from the VLR is unsuccessful, the HLR may initiate re-attempts; the number of repeat attempts and the time in between are HLR operator options, depending on the error returned by the VLR.

If certain services required for a subscriber are not supported by the VLR (e.g. Advice of Charge Charging Level), this may result in one of the following outcomes:

- the HLR stores and sends "Roaming Restriction Due To Unsupported Feature" in a subsequent MAP_INSERT_SUBSCRIBER_DATA service. If "Roaming Restriction Due To Unsupported Feature" is stored in the HLR, the "MSC Area Restricted Flag" shall be set to "restricted". This will prevent MT calls, MT SM and MT USSD from being forwarded to the MSC/VLR.

- the HLR stores and sends other induced subscriber data (e.g. a specific barring program) in a subsequent MAP_INSERT_SUBSCRIBER_DATA service. This will cause rejection of mobile originated service requests, except emergency calls.

When the VLR receives regional subscription data (Zone Code List) it may respond with "MSC Area Restricted" in the MAP_INSERT_SUBSCRIBER_DATA response. In this case the "MSC Area Restricted Flag" shall be set to "restricted" in the HLR. This will prevent MT calls, MT SM and MT USSD from being forwarded to the MSC/VLR.

If the HLR does not store "Roaming Restriction Due To Unsupported Feature" as a consequence of the stand alone Insert Subscriber Data procedure and the HLR does not receive "MSC Area Restricted" in the MAP_INSERT_SUBSCRIBER_DATA response and "Roaming Restriction Due To Unsupported Feature" has not been stored in the HLR in the course of a previous subscriber data retrieval procedure, the "MSC Area Restricted Flag" in the HLR shall be set to "not restricted".

The SDL diagram is shown in figure 21.7/2.

Figure 21.7/2 (sheet 1 of 2)
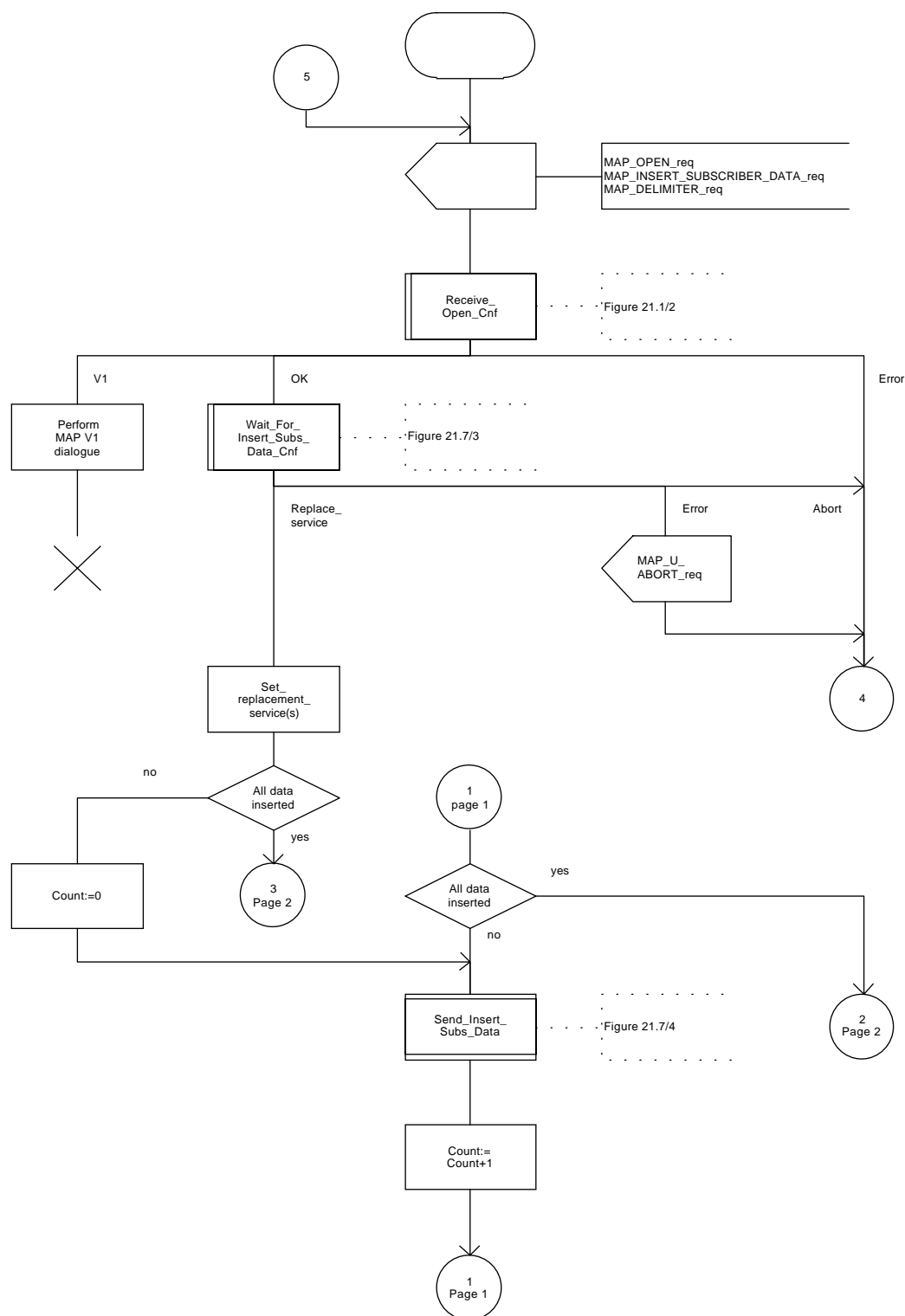
Figure 21.7/2 (sheet 2 of 2)
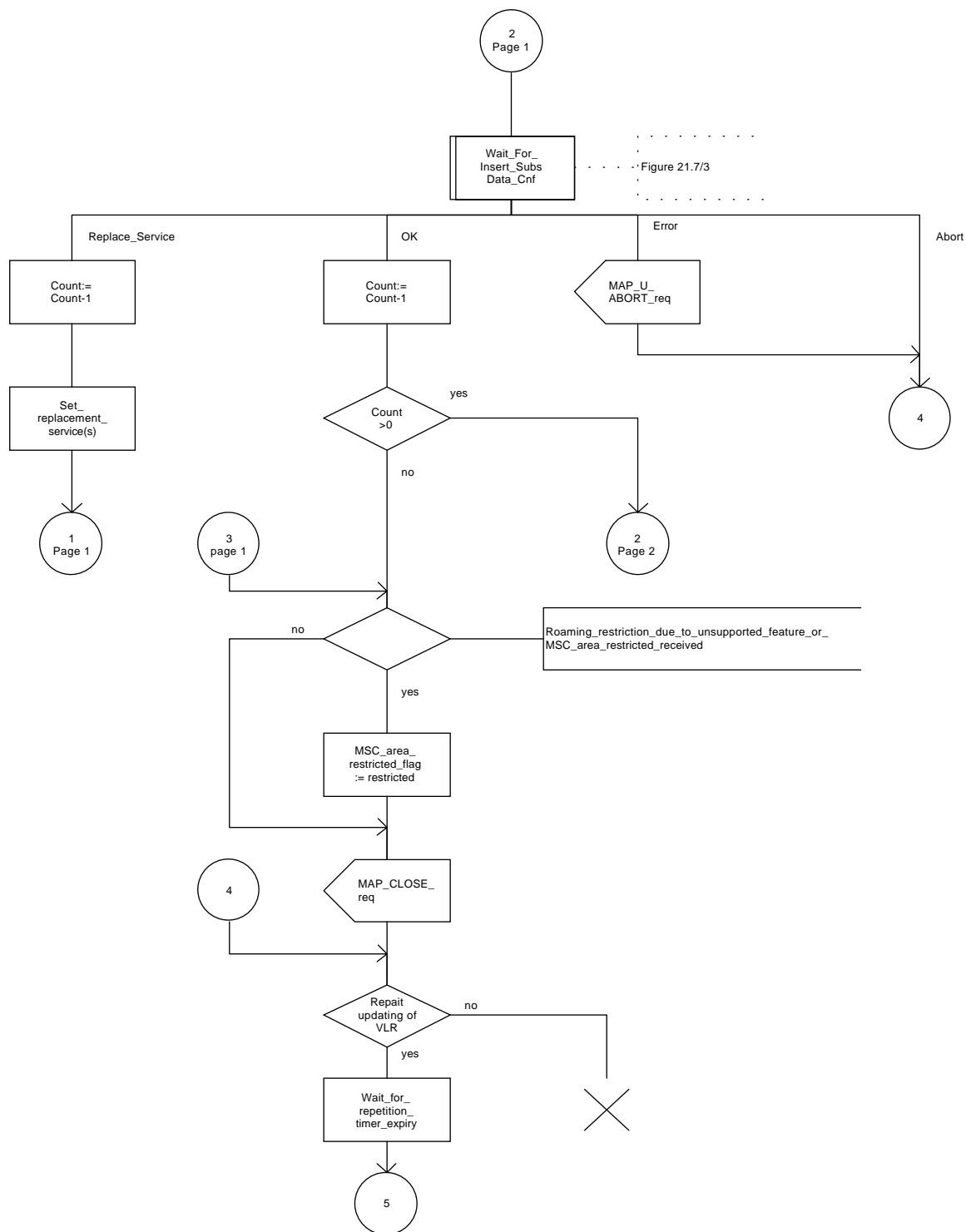
21.7.3   Macro Wait_for_Insert_Subs_Data_Cnf

This macro is used by any process or macro that describes the handling of the reception of the Insert_Subscriber_Data service in HLR (e.g. Update Location or Restore Data).

If the VLR reports the non-support of some basic or supplementary service or the network feature Operator Determined Barring then three actions are possible:

-        to ignore the information received;

-        to replace the not supported service;

-        or to perform any other internal action.

The SDL diagram is shown in figure 21.7/3.

Figure 21.7/3
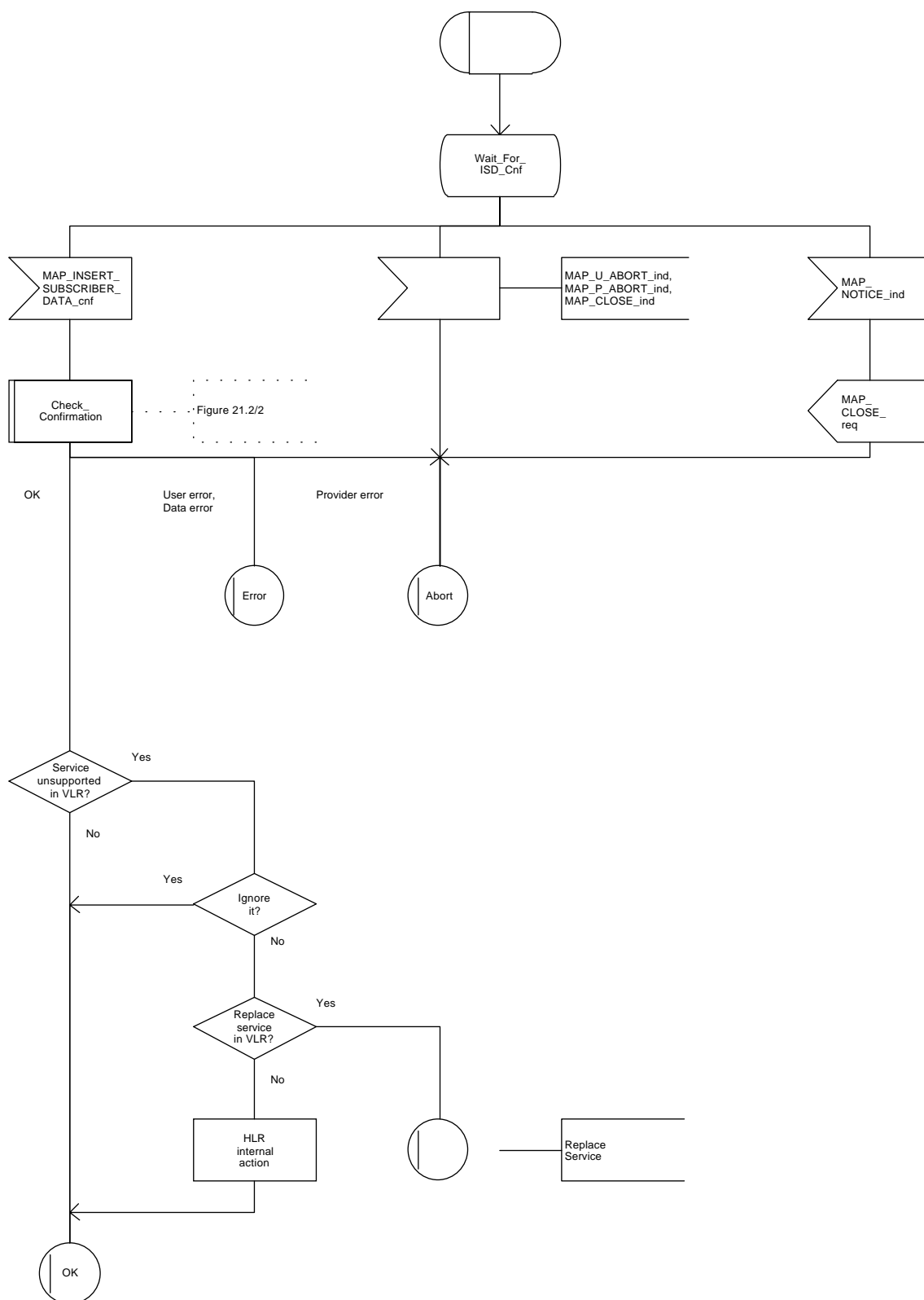
21.7.4 Process Send_Insert_Subs_Data

This process is used by any process or macro where the Insert_Subscriber_Data request is sent to VLR.

The SDL diagram is shown in figure 21.7/4.

Figure 21.7/4

MAP_INSERT_SUBSCRIBER_DATA_req
MAP_DELIMITER_req
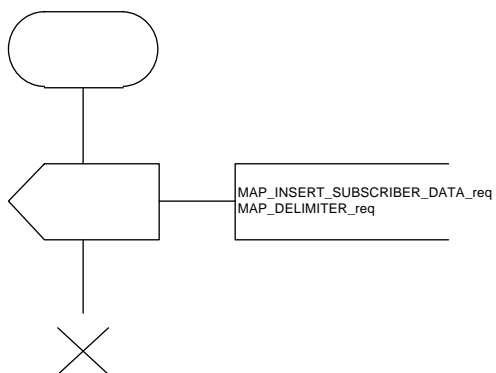
21.8 Request IMSI Macros

21.8.1 Macro Obtain_IMSI_MSC

This macro describes the handling of the request received from the VLR to provide the IMSI of a subscriber (e.g. at Location Updating).

The SDL diagram is shown in figure 21.8/1.

Figure 21.8/1
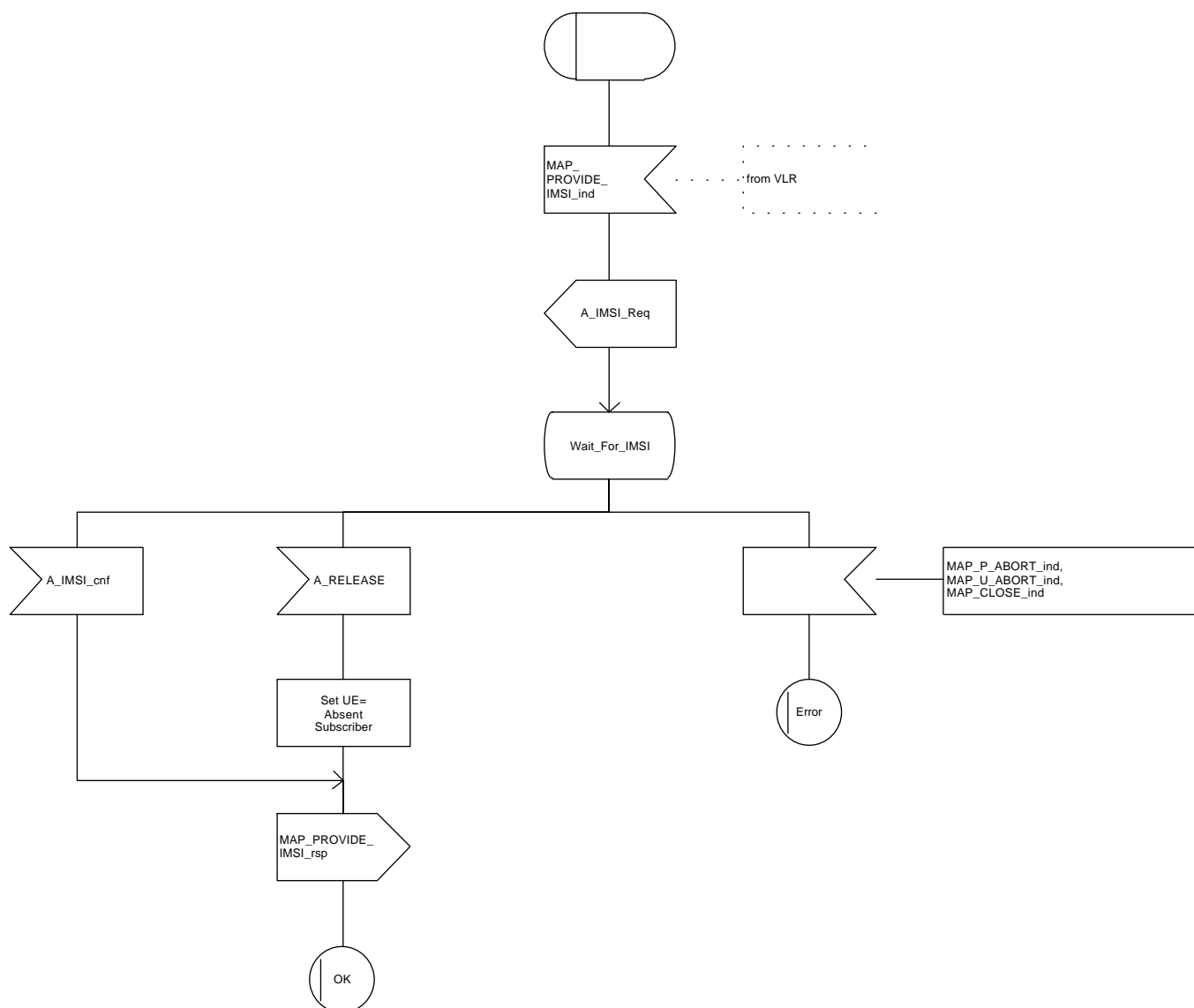
21.8.2 Macro Obtain_IMSI_VLR

This macro describes the way VLR requests the MSC the IMSI of a subscriber (e.g. at Location Updating).

The SDL diagram is shown in figure 21.8/2.

Figure 21.8/2
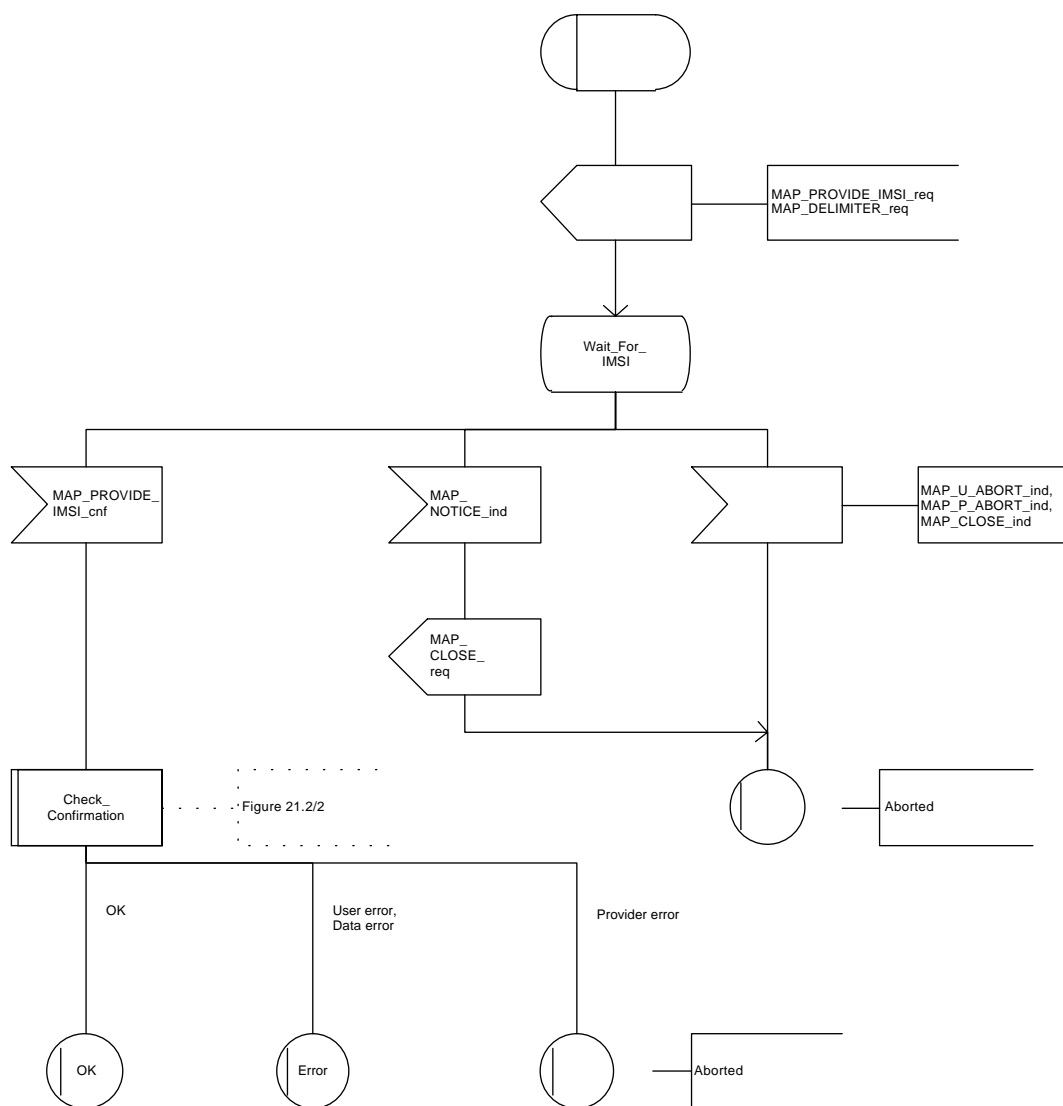
21.9   Tracing macros

21.9.1   Macro Trace_Subscriber_Activity_MSC

The Trace_Subscriber_Activity_MSC is invoked in the MSC, when the MSC receives the MAP_TRACE_SUBSCRIBER_ACTIVITY indication from the VLR. The data of the primitive is checked and the tracing in the MSC is started if the content includes no errors. No response is returned to the VLR.

The Trace_Subscriber_Activity_MSC macro is described in the figure 21.9/1.

Figure 21.9/1
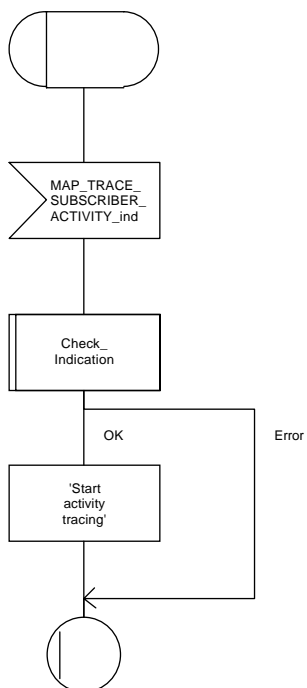
21.9.2   Macro Trace_Subscriber_Activity_VLR

The macro Trace_Subscriber_Activity_VLR is invoked, if the subscriber activity is detected by the VLR and the tracing is active. The VLR sends MAP_TRACE_SUBSCRIBER_ACTIVITY request to the MSC. No answer is awaited from the MSC.

The Trace_Subscriber_Activity_VLR macro is shown in the figure 21.9/2.

Figure 21.9/2

no

Subscriber
Tracing =
Active'

yes

MAP_TRACE_SUBSCRIBER_ACTIVITY_req
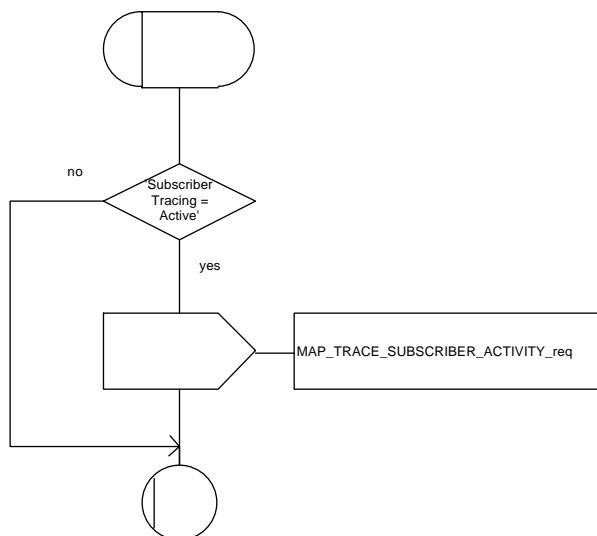
21.9.3 Macro Activate_Tracing_VLR

The Activate_Tracing_VLR macro is invoked, when the MAP_ACTIVATE_TRACE_MODE indication is received from the HLR. The primitive is processed in the VLR as follows:

- if the data contains errors, a data missing or unexpected data value indication is returned to the HLR;

- if the tracing is not supported, a facility not supported indication is returned to the HLR;

- if the tracing buffer does not have any space left for the data, a tracing buffer full indication is returned to the HLR;

- if no errors is detected, the tracing is set active and a positive acknowledge is returned to the HLR.

The Activate_Tracing_VLR macro is described in the figure 21.9/3.

Figure 21.9/3
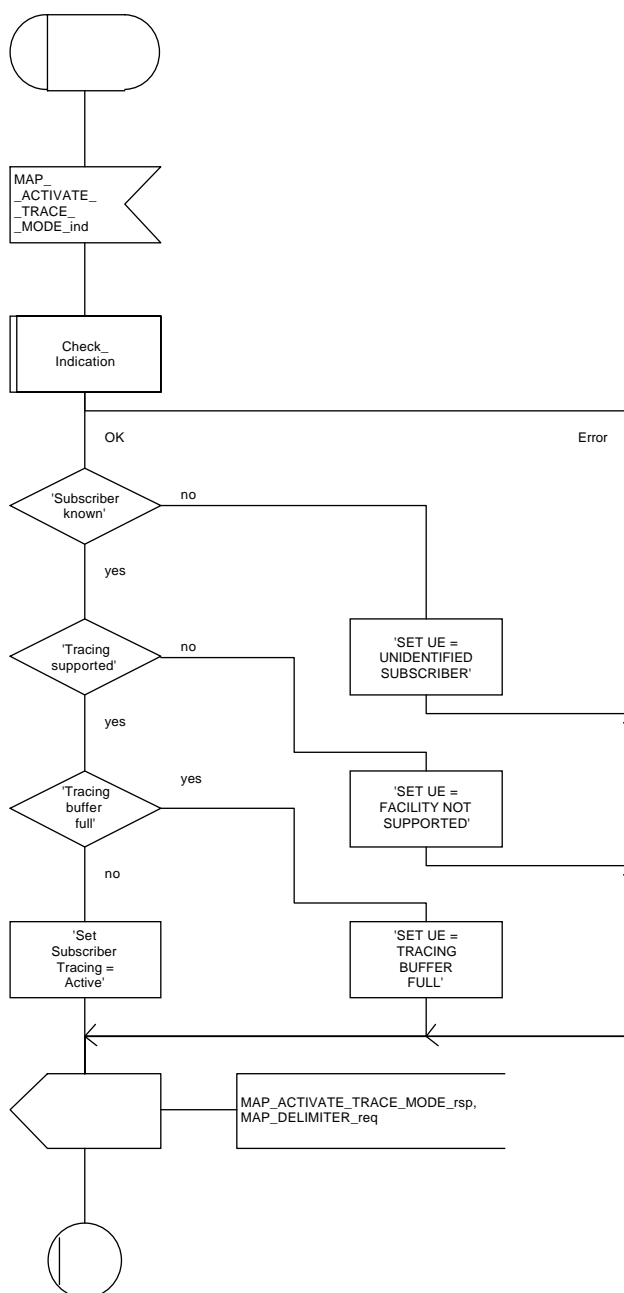
21.9.4   Macro Control_Tracing_HLR

The Control_Tracing_HLR macro may be invoked in the HLR, if subscriber related activity is detected. If the tracing is active in the HLR and not active in the VLR, the MAP_ACTIVATE_TRACE_MODE request is sent to the VLR.

The MAP_ACTIVATE_TRACE_MODE confirmation from the VLR is processed as follows:

-       if the primitive contains a successful acknowledge, the tracing in VLR is set active;

-       if the primitive contains errors, the tracing in VLR is set deactive.

The Control_Tracing_HLR macro is shown in the figure 21.9/4.

Figure 21.9/4
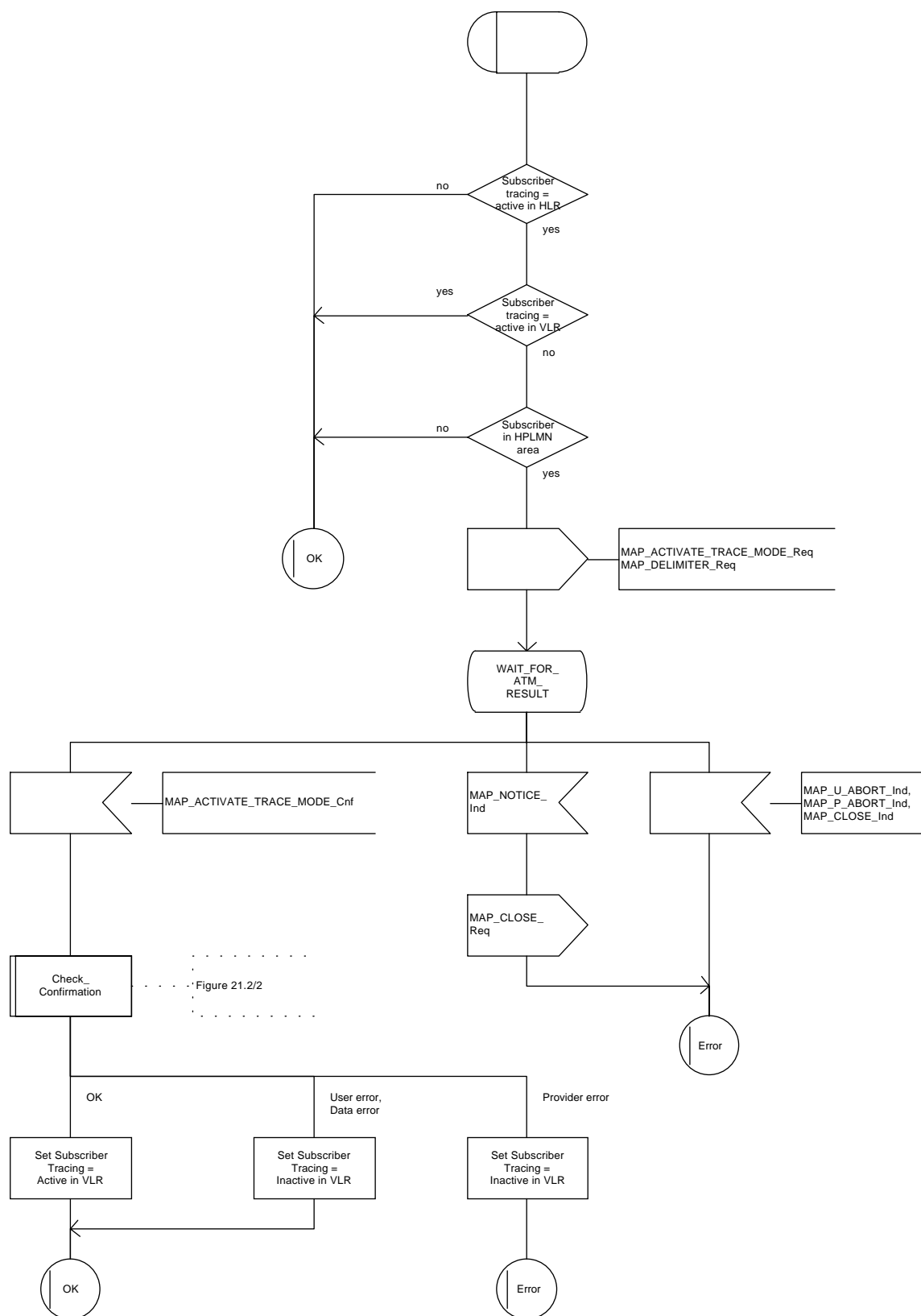
21.10 Short Message Alert procedures

21.10.1 Subscriber_Present_VLR process

The Subscriber_Present_VLR process is invoked by the VLR, when the mobile subscriber becomes active and the MNRF flag is set. The general description of the short message alert procedures is in the subclause 20.4.

The VLR sends the MAP_READY_FOR_SM request to the HLR and waits for the HLR to answer. When receiving the answer, the VLR will act as follows:

- the MNRF flag is cleared if the procedure is successful;

- the MNRF flag is not cleared if the procedure is not successful.

The Subscriber_Present_VLR process is shown in the figure 21.10/1.

Figure 21.10/1
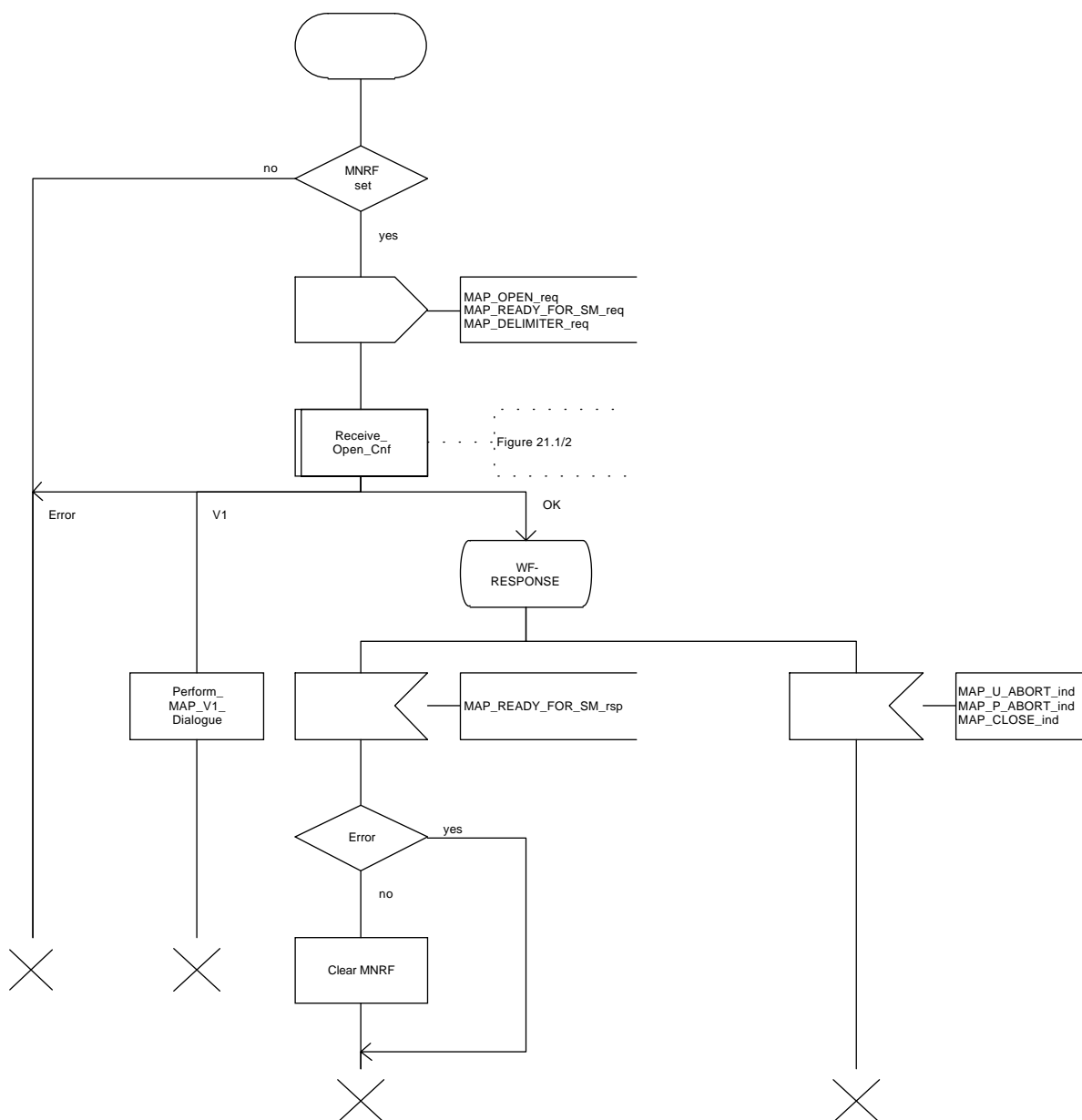
21.10.2 Macro Alert_Service_Centre_HLR

The Alert_Service_Centre_HLR macro is initiated when the HLR notices that the Service Centre(s) shall be alerted. The macro starts process Alert_Service_Centre_HLR for every SC address in the MWD list.

In the process Alert_Service_Centre_HLR the HLR sends MAP_ALERT_SERVICE_CENTRE request to the appropriate IWMSC. The MWD entry is deleted when the positive acknowledge is received from the IWMSC. The unsuccessful alert may be repeated. The MWD entry should be purged in the unsuccessful case, at least when a suitable time period has expired.

The Alert_Service_Centre_HLR macro is shown in the figure 21.10/2 and the Alert_Service_Centre_HLR process is shown in the figure 21.10/3.

Figure 21.10/2
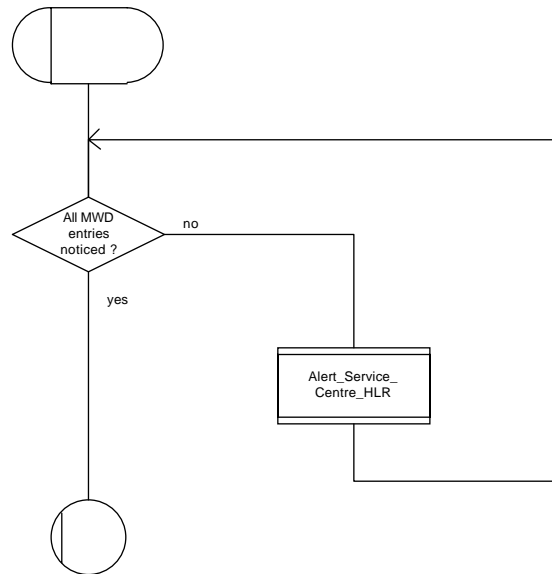
Figure 21.10/3
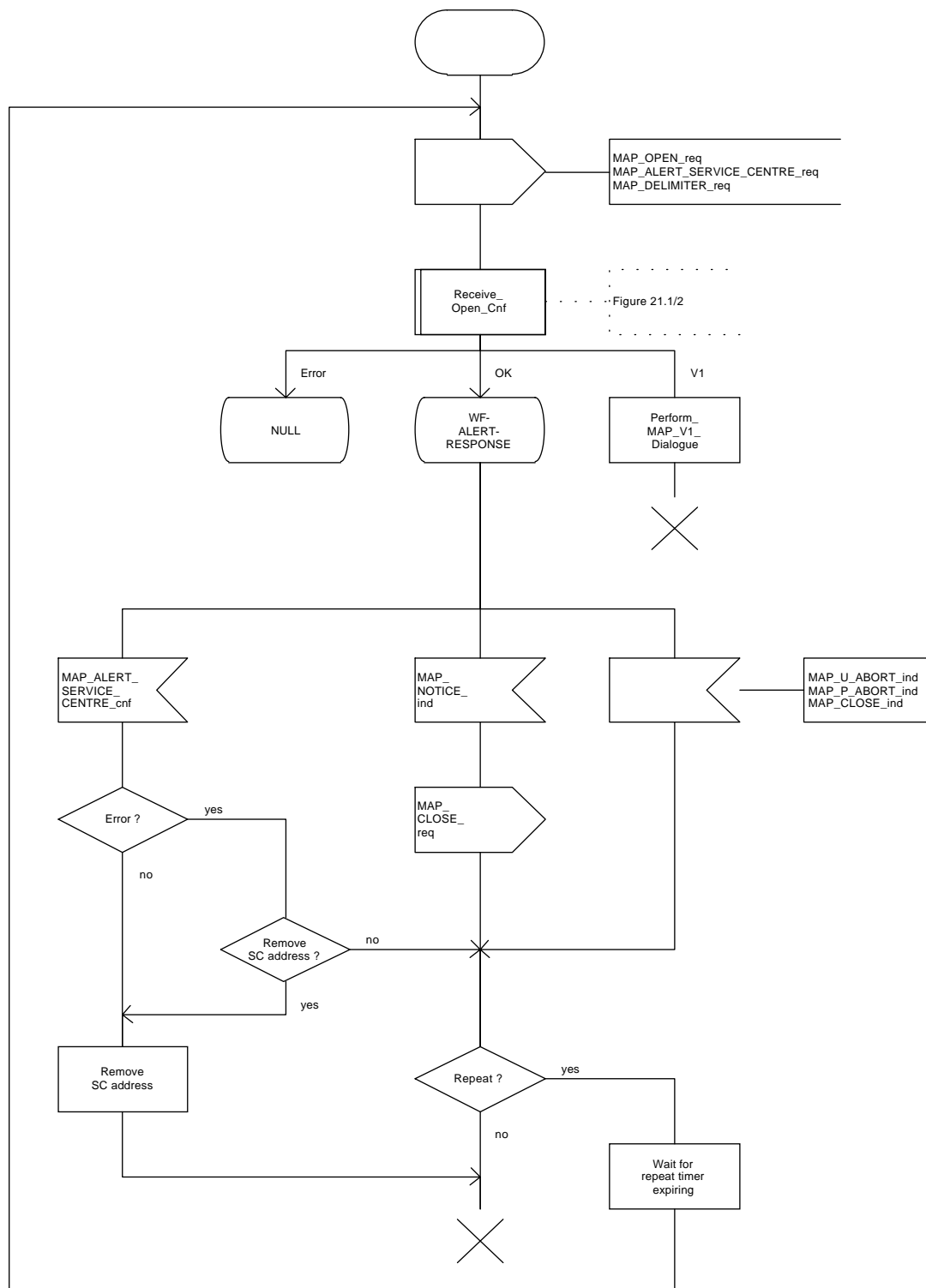
Annex A (informative): Cross-reference for abstract syntaxes of MAP

Annex A is not part of the standard, it is included for information purposes only.

For every ASN.1 item such as identifier, type-reference or value-reference the cross-reference allows to locate all occurrences by means of module-name and line numbers. For that purpose line numbers are printed at the left margin in front of each ASN.1 source line starting with 1 for every module.

The items are sorted alphabetically in the cross-reference in a case-insensitive manner. Occurrences of an item are its definition and all its usages such as in exports, imports or within a type or value assignment.

For every item additional information is provided such as kind of item (identifier, value reference, type reference), and tag, associated type and value if applicable.

The cross-reference for a root module includes all modules referred to directly or indirectly via imports. The cross-references for the root modules MAP-Protocol/TCAPMessages and MAP-DialoguePDU are included.

```
abort.......................identifier of [APPLICATION 7] IMPLICIT Abort
    DEFINED in TCAPMessages          :     56

Abort.......................type reference SEQUENCE
    DEFINED in TCAPMessages          :     74
       USED in TCAPMessages          :     56

absentSubscriber..............value reference AbsentSubscriber, CHOICE VALUE
    DEFINED in MAP-Protocol          :    290

AbsentSubscriber..............type reference ERROR
    DEFINED in MAP-Errors            :    163
       USED in MAP-Protocol          :    104    290
       USED in MAP-CallHandlingOperat :    31    66    82
       USED in MAP-SupplementaryServi :    50   235   249
       USED in MAP-ShortMessageServic :    37    77    94
       USED in MAP-Errors            :     45

absentSubscriber..............identifier of Named Number, 1
    DEFINED in MAP-SM-DataTypes      :     95

accessSignalInfo..............identifier of ExternalSignalInfo
    DEFINED in MAP-MobileServiceOpera :   202

accessSignalInfo..............identifier of ExternalSignalInfo
    DEFINED in MAP-MS-DataTypes      :    169

activateSS....................value reference ActivateSS, CHOICE VALUE
    DEFINED in MAP-Protocol          :    205

ActivateSS....................type reference OPERATION
    DEFINED in MAP-SupplementaryServi :   129
       USED in MAP-Protocol          :     55   205
       USED in MAP-SupplementaryServi :    15

activateTraceMode.............value reference ActivateTraceMode, CHOICE VALUE
    DEFINED in MAP-Protocol          :    184

ActivateTraceMode.............type reference OPERATION
    DEFINED in MAP-OperationAndMainte :    52
       USED in MAP-Protocol          :     36   184
       USED in MAP-OperationAndMainte :    13

activateTraceModeArg..........identifier of ActivateTraceModeArg
    DEFINED in MAP-OperationAndMainte :    54

ActivateTraceModeArg..........type reference SEQUENCE
    DEFINED in MAP-OM-DataTypes      :     32
       USED in MAP-OperationAndMainte :    36    54
       USED in MAP-OM-DataTypes      :     14

AddressString.................type reference OCTET STRING
    DEFINED in MAP-CommonDataTypes   :     69
       USED in MAP-CommonDataTypes   :     16   113
       USED in MAP-OM-DataTypes      :     21    36    54
       USED in MAP-SS-DataTypes      :     33    55
```

```
        USED in MAP-SM-DataTypes       :     24    44    75    81    87   100

alertReason....................identifier of AlertReason
   DEFINED in MAP-SM-DataTypes        :    115

AlertReason....................type reference ENUMERATED
   DEFINED in MAP-SM-DataTypes        :    118
      USED in MAP-SM-DataTypes        :    115

alertServiceCentre............value reference AlertServiceCentre, CHOICE VALUE
   DEFINED in MAP-Protocol            :    237

AlertServiceCentre............type reference OPERATION
   DEFINED in MAP-ShortMessageServic  :    122
      USED in MAP-Protocol            :     75   237
      USED in MAP-ShortMessageServic  :     18

alertServiceCentreArg.........identifier of AlertServiceCentreArg
   DEFINED in MAP-ShortMessageServic  :    120

alertServiceCentreArg.........identifier of AlertServiceCentreArg
   DEFINED in MAP-ShortMessageServic  :    124

AlertServiceCentreArg.........type reference SEQUENCE
   DEFINED in MAP-SM-DataTypes        :     98
      USED in MAP-ShortMessageServic  :     50   120   124
      USED in MAP-SM-DataTypes        :     18

alertServiceCentreWithoutResult...value reference AlertServiceCentreWithoutResult, CHOICE
VALUE
   DEFINED in MAP-Protocol            :    231

AlertServiceCentreWithoutResult.........type reference OPERATION
   DEFINED in MAP-ShortMessageServic  :    118
      USED in MAP-Protocol            :     74   231
      USED in MAP-ShortMessageServic  :     17

allAdditionalInfoTransferSS...value reference SS-Code, '10000000'B
   DEFINED in MAP-SS-Code             :     94

allAlternateSpeech-DataCDA....value reference BearerServiceCode, '00110000'B
   DEFINED in MAP-BS-Code             :     60

allAlternateSpeech-DataCDS....value reference BearerServiceCode, '00111000'B
   DEFINED in MAP-BS-Code             :     62

allAsynchronousServices.......value reference BearerServiceCode, '01100000'B
   DEFINED in MAP-BS-Code             :     73

allBearerServices.............value reference BearerServiceCode, '00000000'B
   DEFINED in MAP-BS-Code             :     31

allCallCompletionSS...........value reference SS-Code, '01000000'B
   DEFINED in MAP-SS-Code             :     63

allCallOfferingSS.............value reference SS-Code, '00110000'B
   DEFINED in MAP-SS-Code             :     53

allCallRestrictionSS..........value reference SS-Code, '10010000'B
   DEFINED in MAP-SS-Code             :    102

allChargingSS.................value reference SS-Code, '01110000'B
   DEFINED in MAP-SS-Code             :     86

allCommunityOfInterest-SS.....value reference SS-Code, '01100000'B
   DEFINED in MAP-SS-Code             :     80

allCondForwardingSS...........value reference SS-Code, '00101000'B
   DEFINED in MAP-SS-Code             :     44

allDataCDA-Services...........value reference BearerServiceCode, '00010000'B
   DEFINED in MAP-BS-Code             :     33

allDataCDS-Services...........value reference BearerServiceCode, '00011000'B
   DEFINED in MAP-BS-Code             :     41

allDataCircuitAsynchronous....value reference BearerServiceCode, '01010000'B
   DEFINED in MAP-BS-Code             :     70

allDataCircuitSynchronous.............value reference BearerServiceCode, '01011000'B
   DEFINED in MAP-BS-Code             :     76
```

```
allDataPDS-Services...........value reference BearerServiceCode, '00101000'B
   DEFINED in MAP-BS-Code            :      55

allDataTeleservices...........value reference TeleserviceCode, '01110000'B
   DEFINED in MAP-TS-Code            :      38

allFacsimileTransmissionServices.value reference TeleserviceCode, '01100000'B
   DEFINED in MAP-TS-Code            :      30

allForwardingSS...............value reference SS-Code, '00100000'B
   DEFINED in MAP-SS-Code            :      40

allLineIdentificationSS.......value reference SS-Code, '00010000'B
   DEFINED in MAP-SS-Code            :      25

allMultiPartySS...............value reference SS-Code, '01010000'B
   DEFINED in MAP-SS-Code            :      74

allOG-CallsBarred.............identifier of Named Number, 0
   DEFINED in MAP-MS-DataTypes       :     260

allPadAccessCA-Services.......value reference BearerServiceCode, '00100000'B
   DEFINED in MAP-BS-Code            :      47

allPLMN-specificBS............value reference BearerServiceCode, '11010000'B
   DEFINED in MAP-BS-Code            :      90

allPLMN-specificSS............value reference SS-Code, '11110000'B
   DEFINED in MAP-SS-Code            :     121

allPLMN-specificTS............value reference TeleserviceCode, '11010000'B
   DEFINED in MAP-TS-Code            :      51

allShortMessageServices.......value reference TeleserviceCode, '00100000'B
   DEFINED in MAP-TS-Code            :      26

allSpeechFollowedByDataCDA....value reference BearerServiceCode, '01000000'B
   DEFINED in MAP-BS-Code            :      64

allSpeechFollowedByDataCDS....value reference BearerServiceCode, '01001000'B
   DEFINED in MAP-BS-Code            :      66

allSpeechTransmissionServices.value reference TeleserviceCode, '00010000'B
   DEFINED in MAP-TS-Code            :      22

allSS.........................value reference SS-Code, '00000000'B
   DEFINED in MAP-SS-Code            :      21

allSynchronousServices........value reference BearerServiceCode, '01101000'B
   DEFINED in MAP-BS-Code            :      79

allTeleservices...............value reference TeleserviceCode, '00000000'B
   DEFINED in MAP-TS-Code            :      20

allTeleservices-ExeptSMS......value reference TeleserviceCode, '10000000'B
   DEFINED in MAP-TS-Code            :      41

aocc..........................value reference SS-Code, '01110010'B
   DEFINED in MAP-SS-Code            :      91

aoci..........................value reference SS-Code, '01110001'B
   DEFINED in MAP-SS-Code            :      89

AuthenticationSet.............type reference SEQUENCE
   DEFINED in MAP-MS-DataTypes       :     119
      USED in MAP-MS-DataTypes       :     117    221

authenticationSet.............identifier of [1] AuthenticationSet
   DEFINED in MAP-MS-DataTypes       :     221

authenticationSetList.........identifier of AuthenticationSetList
   DEFINED in MAP-MS-DataTypes       :     113

AuthenticationSetList.........type reference SEQUENCE OF
   DEFINED in MAP-MS-DataTypes       :     116
      USED in MAP-MS-DataTypes       :     113    187

automaticFacsimileGroup3......value reference TeleserviceCode, '01100010'B
   DEFINED in MAP-TS-Code            :      32
```

```
badlyFormattedTransactionPortion........identifier of Named Number, 2
   DEFINED in TCAPMessages          :    105

badlyStructuredComponent......identifier of Named Number, 2
   DEFINED in TCAPMessages          :    181

badPW-FormatTryAgain..........identifier of Named Number, 4
   DEFINED in MAP-SS-DataTypes       :    274

badPW-TryAgain...............identifier of Named Number, 3
   DEFINED in MAP-SS-DataTypes       :    272

baic.........................value reference SS-Code, '10011010'B
   DEFINED in MAP-SS-Code           :    115

baoc.........................value reference SS-Code, '10010010'B
   DEFINED in MAP-SS-Code           :    106

barringOfIncomingCalls........value reference SS-Code, '10011001'B
   DEFINED in MAP-SS-Code           :    113

barringOfOutgoingCalls........value reference SS-Code, '10010001'B
   DEFINED in MAP-SS-Code           :    104

barringServiceActive..........identifier of Named Number, 0
   DEFINED in MAP-ER-DataTypes       :    46

basicService..................identifier of BasicServiceCode
   DEFINED in MAP-SS-DataTypes       :    54

basicService..................identifier of BasicServiceCode
   DEFINED in MAP-SS-DataTypes       :    80

basicService..................identifier of BasicServiceCode
   DEFINED in MAP-SS-DataTypes       :    147

basicService..................identifier of BasicServiceCode
   DEFINED in MAP-SS-DataTypes       :    183

basicService..................identifier of BasicServiceCode
   DEFINED in MAP-SS-DataTypes       :    223

basicService..................identifier of BasicServiceCode
   DEFINED in MAP-ER-DataTypes       :    59

BasicServiceCode..............type reference CHOICE
   DEFINED in MAP-CommonDataTypes    :    255
      USED in MAP-CommonDataTypes    :    38    260    265
      USED in MAP-SS-DataTypes       :    38    54    80    147    183    223
      USED in MAP-ER-DataTypes       :    29    59

BasicServiceGroupList.........type reference SEQUENCE OF
   DEFINED in MAP-CommonDataTypes    :    264
      USED in MAP-CommonDataTypes    :    40
      USED in MAP-SS-DataTypes       :    39    164    204    235

basicServiceGroupList.........identifier of BasicServiceGroupList
   DEFINED in MAP-SS-DataTypes       :    164

basicServiceGroupList.........identifier of BasicServiceGroupList
   DEFINED in MAP-SS-DataTypes       :    204

basicServiceGroupList.........identifier of [2] BasicServiceGroupList
   DEFINED in MAP-SS-DataTypes       :    235

basicServiceList..............identifier of [1] BasicServiceList
   DEFINED in MAP-MS-DataTypes       :    299

BasicServiceList..............type reference SEQUENCE OF
   DEFINED in MAP-CommonDataTypes    :    259
      USED in MAP-MS-DataTypes        :    71    299
      USED in MAP-CommonDataTypes    :    39

bearerService.................identifier of [2] BearerServiceCode
   DEFINED in MAP-CommonDataTypes    :    256

BearerServiceCode.............type reference OCTET STRING
   DEFINED in MAP-BS-Code           :    11
      USED in MAP-CommonDataTypes    :    50    246    256
      USED in MAP-BS-Code           :    31    33    34    35    36    37    38    39    41
                                          42    43    44    45    47    48    49    50    51
```

```
                                         52    53    55    56    57    58    60    62    64
                                         66    70    73    76    79    90    92    93    94
                                         95    96    97    98    99   100   101   102   103
                                        104   105   106
```

bearerServiceList.............identifier of [4] BearerServiceList
    DEFINED in MAP-MS-DataTypes   :   238

bearerServiceList.............identifier of [2] BearerServiceList
    DEFINED in MAP-MS-DataTypes   :   283

BearerServiceList.............type reference SEQUENCE OF
    DEFINED in MAP-CommonDataTypes   :   245
        USED in MAP-MS-DataTypes   :   69   238   283
        USED in MAP-CommonDataTypes :   36

bearerServiceNotProvisioned..........value reference BearerServiceNotProvisioned, CHOICE
VALUE
    DEFINED in MAP-Protocol   :   266

BearerServiceNotProvisioned...type reference ERROR
    DEFINED in MAP-Errors   :   138
        USED in MAP-Protocol   :   96   266
        USED in MAP-CallHandlingOperat :   28   64
        USED in MAP-SupplementaryServi :   37   92   116   142   169   194
        USED in MAP-Errors   :   31

begin........................identifier of [APPLICATION 2] IMPLICIT Begin
    DEFINED in TCAPMessages   :   53

Begin........................type reference SEQUENCE
    DEFINED in TCAPMessages   :   61
        USED in TCAPMessages   :   53

beginSubscriberActivity.value reference BeginSubscriberActivity, CHOICE VALUE
    DEFINED in MAP-Protocol   :   220

BeginSubscriberActivity.......type reference OPERATION
    DEFINED in MAP-SupplementaryServi :   281
        USED in MAP-Protocol   :   64   220
        USED in MAP-SupplementaryServi :   25

beginSubscriberActivityArg....identifier of BeginSubscriberActivityArg
    DEFINED in MAP-SupplementaryServi :   283

BeginSubscriberActivityArg....type reference SEQUENCE
    DEFINED in MAP-SS-DataTypes   :   280
        USED in MAP-SupplementaryServi :   66   283
        USED in MAP-SS-DataTypes   :   27

bicRoam......................value reference SS-Code, '10011011'B
    DEFINED in MAP-SS-Code   :   117

blackListed..................identifier of Named Number, 1
    DEFINED in MAP-MS-DataTypes   :   196

boic.........................value reference SS-Code, '10010011'B
    DEFINED in MAP-SS-Code   :   108

boicExHC.....................value reference SS-Code, '10010100'B
    DEFINED in MAP-SS-Code   :   110

bss-APDU.....................identifier of ExternalSignalInfo
    DEFINED in MAP-MobileServiceOpera :   173

bss-APDU.....................identifier of ExternalSignalInfo
    DEFINED in MAP-MobileServiceOpera :   181

bss-APDU.....................identifier of ExternalSignalInfo
    DEFINED in MAP-MobileServiceOpera :   185

bss-APDU.....................identifier of ExternalSignalInfo
    DEFINED in MAP-MobileServiceOpera :   191

bss-APDU.....................identifier of ExternalSignalInfo
    DEFINED in MAP-MS-DataTypes   :   136

bss-APDU.....................identifier of ExternalSignalInfo
    DEFINED in MAP-MS-DataTypes   :   164

bss-APDU.....................identifier of ExternalSignalInfo

```
         DEFINED in MAP-MS-DataTypes       :      174

callBarred...................value reference CallBarred, CHOICE VALUE
    DEFINED in MAP-Protocol          :      291

CallBarred...................type reference ERROR
    DEFINED in MAP-Errors            :      169
       USED in MAP-Protocol          :      105    291
       USED in MAP-CallHandlingOperat :      32     67
       USED in MAP-SupplementaryServi :      39     94    120    146    173    198
                                            223    265
       USED in MAP-ShortMessageServic :      38     78
       USED in MAP-Errors            :      46

callBarringCause.............identifier of CallBarringCause
    DEFINED in MAP-Errors            :      171

CallBarringCause.............type reference ENUMERATED
    DEFINED in MAP-ER-DataTypes      :      45
       USED in MAP-Errors            :      80     171
       USED in MAP-ER-DataTypes      :      15

CallBarringFeature...........type reference SEQUENCE
    DEFINED in MAP-SS-DataTypes      :      146
       USED in MAP-SS-DataTypes      :      144

callBarringFeatureList.......identifier of CallBarringFeatureList
    DEFINED in MAP-SS-DataTypes      :      139

CallBarringFeatureList.......type reference SEQUENCE OF
    DEFINED in MAP-SS-DataTypes      :      142
       USED in MAP-SS-DataTypes      :      139

callBarringInfo..............identifier of [1] CallBarringInfo
    DEFINED in MAP-SS-DataTypes      :      65

CallBarringInfo..............type reference SEQUENCE
    DEFINED in MAP-SS-DataTypes      :      136
       USED in MAP-SS-DataTypes      :      65

calledPartySS-InteractionViolation......identifier of Named Number, 7
    DEFINED in MAP-ER-DataTypes      :      53

callReference................identifier of [4] CallReference
    DEFINED in MAP-OM-DataTypes      :      55

CallReference................type reference OCTET STRING
    DEFINED in MAP-OM-DataTypes      :      57
       USED in MAP-OM-DataTypes      :      55

cancelLocation...............value reference CancelLocation, CHOICE VALUE
    DEFINED in MAP-Protocol          :      130

CancelLocation...............type reference OPERATION
    DEFINED in MAP-MobileServiceOpera :      119
       USED in MAP-Protocol          :      13     130
       USED in MAP-MobileServiceOpera :      16

cancelLocationArg............identifier of CancelLocationArg
    DEFINED in MAP-MobileServiceOpera :      121

CancelLocationArg............type reference CHOICE
    DEFINED in MAP-MS-DataTypes      :      97
       USED in MAP-MobileServiceOpera :      70     121
       USED in MAP-MS-DataTypes      :      18

category.....................identifier of [2] Category
    DEFINED in MAP-MS-DataTypes      :      236

Category.....................type reference OCTET STRING
    DEFINED in MAP-CommonDataTypes   :      242
       USED in MAP-MS-DataTypes      :      68     236
       USED in MAP-CommonDataTypes   :      35

ccbs.........................value reference SS-Code, '01000011'B
    DEFINED in MAP-SS-Code           :      70

cfb..........................value reference SS-Code, '00101001'B
    DEFINED in MAP-SS-Code           :      46

cfnrc........................value reference SS-Code, '00101011'B
```

```
                DEFINED in MAP-SS-Code         :     50


cfnry........................value reference SS-Code, '00101010'B
   DEFINED in MAP-SS-Code         :     48


cfu..........................value reference SS-Code, '00100001'B
   DEFINED in MAP-SS-Code         :     42


channelId....................identifier of [2] ExternalSignalInfo
   DEFINED in MAP-OM-DataTypes    :     62


channelType..................identifier of ChannelType
   DEFINED in MAP-MS-DataTypes    :    142


ChannelType..................type reference OCTET STRING
   DEFINED in MAP-MS-DataTypes    :    147
      USED in MAP-MS-DataTypes    :    142


checkIMEI....................value reference CheckIMEI, CHOICE VALUE
   DEFINED in MAP-Protocol        :    162


CheckIMEI....................type reference OPERATION
   DEFINED in MAP-MobileServiceOpera :  226
      USED in MAP-Protocol        :     24    162
      USED in MAP-MobileServiceOpera :    33


classmarkInfo................identifier of ClassmarkInfo
   DEFINED in MAP-MS-DataTypes    :    143


ClassmarkInfo................type reference OCTET STRING
   DEFINED in MAP-MS-DataTypes    :    155
      USED in MAP-MS-DataTypes    :    143    181


classmarkInfo................identifier of [10] ClassmarkInfo
   DEFINED in MAP-MS-DataTypes    :    181


clip.........................value reference SS-Code, '00010001'B
   DEFINED in MAP-SS-Code         :     28


clir.........................value reference SS-Code, '00010010'B
   DEFINED in MAP-SS-Code         :     30


cliRestrictionOption.........identifier of [2] CliRestrictionOption
   DEFINED in MAP-SS-DataTypes    :    209


CliRestrictionOption.........type reference ENUMERATED
   DEFINED in MAP-SS-DataTypes    :    212
      USED in MAP-SS-DataTypes    :    209    228


cliRestrictionOption.........identifier of CliRestrictionOption
   DEFINED in MAP-SS-DataTypes    :    228


Cli-RestrictionInfo..........type reference SEQUENCE
   DEFINED in MAP-SS-DataTypes    :    226
      USED in MAP-SS-DataTypes    :    237


cli-RestrictionInfo..........identifier of [4] Cli-RestrictionInfo
   DEFINED in MAP-SS-DataTypes    :    237


colp.........................value reference SS-Code, '00010011'B
   DEFINED in MAP-SS-Code         :     32


colr.........................value reference SS-Code, '00010100'B
   DEFINED in MAP-SS-Code         :     34


Component....................type reference CHOICE
   DEFINED in TCAPMessages        :    124
      USED in TCAPMessages        :     47    115


ComponentPortion.............type reference  [APPLICATION 12] IMPLICIT SEQUENCE OF
   DEFINED in TCAPMessages        :    115
      USED in TCAPMessages        :     59    63    67    72


components...................identifier of ComponentPortion
   DEFINED in TCAPMessages        :     59


components...................identifier of ComponentPortion
   DEFINED in TCAPMessages        :     63


components...................identifier of ComponentPortion
   DEFINED in TCAPMessages        :     67
```

```
components....................identifier of ComponentPortion
    DEFINED in TCAPMessages        :      72

Continue.....................type reference SEQUENCE
    DEFINED in TCAPMessages        :      69
      USED in TCAPMessages         :      55

continue-ME..................identifier of [APPLICATION 5] IMPLICIT Continue
    DEFINED in TCAPMessages        :      55

controllingMSC...............identifier of Named Number, 4
    DEFINED in MAP-CommonDataTypes :     234

ct...........................value reference SS-Code, '00110001'B
    DEFINED in MAP-SS-Code         :      56

cug..........................value reference SS-Code, '01100001'B
    DEFINED in MAP-SS-Code         :      83

cugIC-CallBarred.............identifier of Named Number, 1
    DEFINED in MAP-SS-DataTypes    :     174

cugOG-CallBarred.............identifier of Named Number, 2
    DEFINED in MAP-SS-DataTypes    :     175

CUG-CheckInfo................type reference SEQUENCE
    DEFINED in MAP-CH-DataTypes    :      36
      USED in MAP-CH-DataTypes     :      45    54

cug-CheckInfo................identifier of [1] CUG-CheckInfo
    DEFINED in MAP-CH-DataTypes    :      45

cug-CheckInfo................identifier of CUG-CheckInfo
    DEFINED in MAP-CH-DataTypes    :      54

CUG-Feature..................type reference SEQUENCE
    DEFINED in MAP-SS-DataTypes    :     182
      USED in MAP-SS-DataTypes     :     180

cug-FeatureList..............identifier of CUG-FeatureList
    DEFINED in MAP-SS-DataTypes    :     154

CUG-FeatureList..............type reference SEQUENCE OF
    DEFINED in MAP-SS-DataTypes    :     179
      USED in MAP-SS-DataTypes     :     154

cug-Index....................identifier of CUG-Index
    DEFINED in MAP-SS-DataTypes    :     161

CUG-Index....................type reference INTEGER
    DEFINED in MAP-SS-DataTypes    :     167
      USED in MAP-SS-DataTypes     :      17   161   184

cug-Info.....................identifier of [2] CUG-Info
    DEFINED in MAP-SS-DataTypes    :      66

CUG-Info.....................type reference SEQUENCE
    DEFINED in MAP-SS-DataTypes    :     152
      USED in MAP-SS-DataTypes     :      66

cug-Interlock................identifier of CUG-Interlock
    DEFINED in MAP-CH-DataTypes    :      37

cug-Interlock................identifier of CUG-Interlock
    DEFINED in MAP-SS-DataTypes    :     162

CUG-Interlock................type reference OCTET STRING
    DEFINED in MAP-SS-DataTypes    :     170
      USED in MAP-CH-DataTypes     :      20    37
      USED in MAP-SS-DataTypes     :      18   162

cug-OutgoingAccess...........identifier of NULL
    DEFINED in MAP-CH-DataTypes    :      38

cug-Reject...................value reference CUG-Reject, CHOICE VALUE
    DEFINED in MAP-Protocol        :     293

CUG-Reject...................type reference ERROR
    DEFINED in MAP-Errors          :     176
      USED in MAP-Protocol         :     107   293
```

```
            USED in MAP-CallHandlingOperat :      34     68
            USED in MAP-Errors             :      48

cug-RejectCause...............identifier of CUG-RejectCause
      DEFINED in MAP-Errors            :     178

CUG-RejectCause...............type reference ENUMERATED
      DEFINED in MAP-ER-DataTypes      :      49
            USED in MAP-Errors             :      81    178
            USED in MAP-ER-DataTypes       :      16

CUG-Subscription..............type reference SEQUENCE
      DEFINED in MAP-SS-DataTypes      :     160
            USED in MAP-SS-DataTypes       :     158

cug-SubscriptionList..........identifier of CUG-SubscriptionList
      DEFINED in MAP-SS-DataTypes      :     153

CUG-SubscriptionList..........type reference SEQUENCE OF
      DEFINED in MAP-SS-DataTypes      :     157
            USED in MAP-SS-DataTypes       :     153

currentPassword...............identifier of Password
      DEFINED in MAP-SupplementaryServi :    279

cw............................value reference SS-Code, '01000001'B
      DEFINED in MAP-SS-Code           :      66

dataCDA-1200bps...............value reference BearerServiceCode, '00010010'B
      DEFINED in MAP-BS-Code           :      35

dataCDA-1200-75bps............value reference BearerServiceCode, '00010011'B
      DEFINED in MAP-BS-Code           :      36

dataCDA-2400bps...............value reference BearerServiceCode, '00010100'B
      DEFINED in MAP-BS-Code           :      37

dataCDA-300bps................value reference BearerServiceCode, '00010001'B
      DEFINED in MAP-BS-Code           :      34

dataCDA-4800bps...............value reference BearerServiceCode, '00010101'B
      DEFINED in MAP-BS-Code           :      38

dataCDA-9600bps...............value reference BearerServiceCode, '00010110'B
      DEFINED in MAP-BS-Code           :      39

dataCDS-1200bps...............value reference BearerServiceCode, '00011010'B
      DEFINED in MAP-BS-Code           :      42

dataCDS-2400bps...............value reference BearerServiceCode, '00011100'B
      DEFINED in MAP-BS-Code           :      43

dataCDS-4800bps...............value reference BearerServiceCode, '00011101'B
      DEFINED in MAP-BS-Code           :      44

dataCDS-9600bps...............value reference BearerServiceCode, '00011110'B
      DEFINED in MAP-BS-Code           :      45

dataMissing...................value reference DataMissing, CHOICE VALUE
      DEFINED in MAP-Protocol          :     246

DataMissing...................type reference ERROR
      DEFINED in MAP-Errors            :     103
            USED in MAP-Protocol           :      84    246
            USED in MAP-MobileServiceOpera :      52    113    124    142    154    194
                                                  220    233    262
                                                  274    293
            USED in MAP-OperationAndMainte :      26     58     70     89
            USED in MAP-CallHandlingOperat :      23     58     79
            USED in MAP-SupplementaryServi :      34     88    111    137    164    189
                                                  220    233    247
                                                  262
            USED in MAP-ShortMessageServic :      29     72     86    108    128    140
            USED in MAP-Errors             :      15

dataPDS-2400bps...............value reference BearerServiceCode, '00101100'B
      DEFINED in MAP-BS-Code           :      56

dataPDS-4800bps...............value reference BearerServiceCode, '00101101'B
      DEFINED in MAP-BS-Code           :      57
```

```
dataPDS-9600bps...............value reference BearerServiceCode, '00101110'B
   DEFINED in MAP-BS-Code         :     58

deactivateSS..................value reference DeactivateSS, CHOICE VALUE
   DEFINED in MAP-Protocol        :    206

DeactivateSS..................type reference OPERATION
   DEFINED in MAP-SupplementaryServi :  156
      USED in MAP-Protocol        :     56    206
      USED in MAP-SupplementaryServi :    17

deactivateTraceMode...........value reference DeactivateTraceMode, CHOICE VALUE
   DEFINED in MAP-Protocol        :    185

DeactivateTraceMode...........type reference OPERATION
   DEFINED in MAP-OperationAndMainte :   64
      USED in MAP-Protocol        :     37    185
      USED in MAP-OperationAndMainte :    14

deactivateTraceModeArg........identifier of DeactivateTraceModeArg
   DEFINED in MAP-OperationAndMainte :   66

DeactivateTraceModeArg........type reference SEQUENCE
   DEFINED in MAP-OM-DataTypes    :     45
      USED in MAP-OperationAndMainte :   37    66
      USED in MAP-OM-DataTypes    :     15

deleteSubscriberData..........value reference DeleteSubscriberData, CHOICE VALUE
   DEFINED in MAP-Protocol        :    170

DeleteSubscriberData..........type reference OPERATION
   DEFINED in MAP-MobileServiceOpera :  266
      USED in MAP-Protocol        :     27    170
      USED in MAP-MobileServiceOpera :    38

deleteSubscriberDataArg.......identifier of DeleteSubscriberDataArg
   DEFINED in MAP-MobileServiceOpera :  268

DeleteSubscriberDataArg.......type reference SEQUENCE
   DEFINED in MAP-MS-DataTypes    :    297
      USED in MAP-MobileServiceOpera :   86    268
      USED in MAP-MS-DataTypes    :     42

deleteSubscriberDataRes.......identifier of DeleteSubscriberDataRes
   DEFINED in MAP-MobileServiceOpera :  270

DeleteSubscriberDataRes.......type reference SEQUENCE
   DEFINED in MAP-MS-DataTypes    :    308
      USED in MAP-MobileServiceOpera :   87    270
      USED in MAP-MS-DataTypes    :     43

derivable.....................identifier of InvokeIdType
   DEFINED in TCAPMessages        :    167

DestTransactionID.............type reference  [APPLICATION 9] IMPLICIT TransactionID
   DEFINED in TCAPMessages        :     98
      USED in TCAPMessages        :     65    70    74

diagnosticInfo................identifier of SignalInfo
   DEFINED in MAP-ER-DataTypes    :     89

dialoguePortion...............identifier of DialoguePortion
   DEFINED in TCAPMessages        :     58

dialoguePortion...............identifier of DialoguePortion
   DEFINED in TCAPMessages        :     62

dialoguePortion...............identifier of DialoguePortion
   DEFINED in TCAPMessages        :     66

dialoguePortion...............identifier of DialoguePortion
   DEFINED in TCAPMessages        :     71

dialoguePortion...............identifier of DialoguePortion
   DEFINED in TCAPMessages        :     77

DialoguePortion...............type reference  [APPLICATION 11] EXTERNAL
   DEFINED in TCAPMessages        :     82
      USED in TCAPMessages        :     58    62    66    71    77

dtid..........................identifier of DestTransactionID
```

```
     DEFINED in TCAPMessages        :     65

dtid.........................identifier of DestTransactionID
     DEFINED in TCAPMessages        :     70

dtid.........................identifier of DestTransactionID
     DEFINED in TCAPMessages        :     74

duplicateInvokeID.............identifier of Named Number, 0
     DEFINED in TCAPMessages        :    183

eir..........................identifier of Named Number, 6
     DEFINED in MAP-CommonDataTypes :    236

emergencyCalls...............value reference TeleserviceCode, '00010010'B
     DEFINED in MAP-TS-Code         :     24

End..........................type reference SEQUENCE
     DEFINED in TCAPMessages        :     65
        USED in TCAPMessages        :     54

end-ME.......................identifier of [APPLICATION 4] IMPLICIT End
     DEFINED in TCAPMessages        :     54

enterNewPW...................identifier of Named Number, 1
     DEFINED in MAP-SS-DataTypes    :    270

enterNewPW-Again.............identifier of Named Number, 2
     DEFINED in MAP-SS-DataTypes    :    271

enterPW......................identifier of Named Number, 0
     DEFINED in MAP-SS-DataTypes    :    269

equipmentNotSM-Equipped.......identifier of Named Number, 2
     DEFINED in MAP-ER-DataTypes    :     81

equipmentProtocolError........identifier of Named Number, 1
     DEFINED in MAP-ER-DataTypes    :     80

equipmentStatus..............identifier of EquipmentStatus
     DEFINED in MAP-MobileServiceOpera :  230

EquipmentStatus..............type reference ENUMERATED
     DEFINED in MAP-MS-DataTypes    :    194
        USED in MAP-MobileServiceOpera :   81    230
        USED in MAP-MS-DataTypes    :     35

eraseSS......................value reference EraseSS, CHOICE VALUE
     DEFINED in MAP-Protocol        :    204

EraseSS......................type reference OPERATION
     DEFINED in MAP-SupplementaryServi :  103
        USED in MAP-Protocol        :     54    204
        USED in MAP-SupplementaryServi :   14

errorCode....................identifier of ERROR
     DEFINED in TCAPMessages        :    158

ets-300102-1.................identifier of Named Number, 4
     DEFINED in MAP-CommonDataTypes :    181

extensibleUpdateLocationRes...identifier of ExtensibleUpdateLocationRes
     DEFINED in MAP-MS-DataTypes    :     90

ExtensibleUpdateLocationRes...type reference SEQUENCE
     DEFINED in MAP-MS-DataTypes    :     93
        USED in MAP-MS-DataTypes    :     90

ExternalSignalInfo............type reference SEQUENCE
     DEFINED in MAP-CommonDataTypes :    158
        USED in MAP-MobileServiceOpera :   95   173   181   185   191   202
        USED in MAP-MS-DataTypes    :     59   136   164   169   174
        USED in MAP-CommonDataTypes :     19
        USED in MAP-OM-DataTypes    :     22    62
        USED in MAP-CH-DataTypes    :     27    48    69    70

facilityNotSupported.......value reference FacilityNotSupported, CHOICE VALUE
     DEFINED in MAP-Protocol        :    248

FacilityNotSupported..........type reference ERROR
     DEFINED in MAP-Errors          :    107
```

```
        USED in MAP-Protocol           :    86    248
        USED in MAP-OperationAndMainte :    28    60    72
        USED in MAP-CallHandlingOperat :    25    60    81
        USED in MAP-ShortMessageServic :    31    74    89    142
        USED in MAP-Errors             :    17

facsimileGroup3AndAlterSpeech.value reference TeleserviceCode, '01100001'B
    DEFINED in MAP-TS-Code             :    31

facsimileGroup4...............value reference TeleserviceCode, '01100011'B
    DEFINED in MAP-TS-Code             :    33

forwardAccessSignalling.value reference ForwardAccessSignalling, CHOICE VALUE
    DEFINED in MAP-Protocol            :   145

ForwardAccessSignalling.......type reference OPERATION
    DEFINED in MAP-MobileServiceOpera  :   183
        USED in MAP-Protocol           :    20   145
        USED in MAP-MobileServiceOpera :    25

forwardCheckSS-Indication.....value reference ForwardCheckSS-Indication, CHOICE VALUE
    DEFINED in MAP-Protocol            :   176

ForwardCheckSS-Indication.....type reference OPERATION
    DEFINED in MAP-MobileServiceOpera  :   284
        USED in MAP-Protocol           :    29   176
        USED in MAP-MobileServiceOpera :    42

forwardedToNumber.............identifier of [4] AddressString
    DEFINED in MAP-SS-DataTypes        :    55

forwardedToNumber.............identifier of [5] ISDN-AddressString
    DEFINED in MAP-SS-DataTypes        :    83

forwardedToNumber.............identifier of [1] ISDN-AddressString
    DEFINED in MAP-SS-DataTypes        :   233

forwardedToNumber.............identifier of [5] ISDN-AddressString
    DEFINED in MAP-SS-DataTypes        :   293

forwardedToSubaddress.........identifier of [6] ISDN-SubaddressString
    DEFINED in MAP-SS-DataTypes        :    56

forwardedToSubaddress.........identifier of [8] ISDN-SubaddressString
    DEFINED in MAP-SS-DataTypes        :    84

forwardedToSubaddress.........identifier of [4] ISDN-SubaddressString
    DEFINED in MAP-SS-DataTypes        :   295

forwardingData................identifier of ForwardingData
    DEFINED in MAP-CH-DataTypes        :    60

ForwardingData................type reference SEQUENCE
    DEFINED in MAP-SS-DataTypes        :   292
        USED in MAP-CH-DataTypes       :    21    60
        USED in MAP-SS-DataTypes       :    30

ForwardingFeature.............type reference SEQUENCE
    DEFINED in MAP-SS-DataTypes        :    79
        USED in MAP-SS-DataTypes       :    77

forwardingFeatureList.........identifier of ForwardingFeatureList
    DEFINED in MAP-SS-DataTypes        :    72

ForwardingFeatureList.........type reference SEQUENCE OF
    DEFINED in MAP-SS-DataTypes        :    75
        USED in MAP-SS-DataTypes       :    72   236

forwardingFeatureList.........identifier of [3] ForwardingFeatureList
    DEFINED in MAP-SS-DataTypes        :   236

forwardingInfo................identifier of [0] ForwardingInfo
    DEFINED in MAP-SS-DataTypes        :    64

ForwardingInfo................type reference SEQUENCE
    DEFINED in MAP-SS-DataTypes        :    70
        USED in MAP-SS-DataTypes       :    64

forwardingOptions.............identifier of [6] ForwardingOptions
    DEFINED in MAP-SS-DataTypes        :    86
```

```
ForwardingOptions.............type reference OCTET STRING
    DEFINED in MAP-SS-DataTypes      :     105
       USED in MAP-SS-DataTypes      :      86    297


forwardingOptions.............identifier of [6] ForwardingOptions
    DEFINED in MAP-SS-DataTypes      :     297


forwardingViolation.........value reference ForwardingViolation, CHOICE VALUE
    DEFINED in MAP-Protocol          :     292


ForwardingViolation..........type reference ERROR
    DEFINED in MAP-Errors            :     174
       USED in MAP-Protocol          :     106    292
       USED in MAP-CallHandlingOperat :      33     70
       USED in MAP-Errors            :      47


forwardSM.....................value reference ForwardSM, CHOICE VALUE
    DEFINED in MAP-Protocol          :     227


ForwardSM.....................type reference OPERATION
    DEFINED in MAP-ShortMessageServic :      80
       USED in MAP-Protocol          :      71    227
       USED in MAP-ShortMessageServic :      14


forwardSM-Arg.................identifier of ForwardSM-Arg
    DEFINED in MAP-ShortMessageServic :      82


ForwardSM-Arg.................type reference SEQUENCE
    DEFINED in MAP-SM-DataTypes      :      62
       USED in MAP-ShortMessageServic :      48     82
       USED in MAP-SM-DataTypes      :      16


generalProblem...............identifier of [0] IMPLICIT GeneralProblem
    DEFINED in TCAPMessages          :     170


GeneralProblem...............type reference INTEGER
    DEFINED in TCAPMessages          :     179
       USED in TCAPMessages          :     170


getPassword...................value reference GetPassword, CHOICE VALUE
    DEFINED in MAP-Protocol          :     219


GetPassword...................type reference OPERATION
    DEFINED in MAP-SupplementaryServi :     275
       USED in MAP-Protocol          :      63    219
       USED in MAP-SupplementaryServi :      24    273


GlobalCellId..................type reference OCTET STRING
    DEFINED in MAP-CommonDataTypes    :     219
       USED in MAP-MS-DataTypes      :      66    134    140    141    172    178    179
       USED in MAP-CommonDataTypes    :      31
       USED in MAP-OM-DataTypes      :      24     61


greyListed....................identifier of Named Number, 2
    DEFINED in MAP-MS-DataTypes      :     197


gsm-0408......................identifier of Named Number, 1
    DEFINED in MAP-CommonDataTypes    :     176


gsm-0806......................identifier of Named Number, 2
    DEFINED in MAP-CommonDataTypes    :     177


gsm-BearerCapability..........identifier of [5] ExternalSignalInfo
    DEFINED in MAP-CH-DataTypes      :      69


gsm-BSSMAP....................identifier of Named Number, 3
    DEFINED in MAP-CommonDataTypes    :     178


guidanceInfo..................identifier of GuidanceInfo
    DEFINED in MAP-SupplementaryServi :     277


GuidanceInfo..................type reference ENUMERATED
    DEFINED in MAP-SS-DataTypes      :     268
       USED in MAP-SupplementaryServi :      65    277
       USED in MAP-SS-DataTypes      :      26


handoverNumber...............identifier of ISDN-AddressString
    DEFINED in MAP-MS-DataTypes      :     163


handoverNumber...............identifier of ISDN-AddressString
    DEFINED in MAP-MS-DataTypes      :     168
```

```
handoverPriority..............identifier of [11] HandoverPriority
    DEFINED in MAP-MS-DataTypes       :     144

HandoverPriority..............type reference OCTET STRING
    DEFINED in MAP-MS-DataTypes       :     159
       USED in MAP-MS-DataTypes       :     144

handoverType..................identifier of HandoverType
    DEFINED in MAP-OM-DataTypes       :      60

HandoverType..................type reference ENUMERATED
    DEFINED in MAP-OM-DataTypes       :      64
       USED in MAP-OM-DataTypes       :      60

hlr...........................identifier of Named Number, 1
    DEFINED in MAP-CommonDataTypes    :     231

HLR-Id........................type reference IMSI
    DEFINED in MAP-CommonDataTypes    :     208
       USED in MAP-CommonDataTypes    :     213

hlr-List......................identifier of HLR-List
    DEFINED in MAP-MS-DataTypes       :     320

HLR-List......................type reference SEQUENCE OF
    DEFINED in MAP-CommonDataTypes    :     212
       USED in MAP-MS-DataTypes       :      64    320
       USED in MAP-CommonDataTypes    :      29

hlr-Number....................identifier of ISDN-AddressString
    DEFINED in MAP-MS-DataTypes       :      88

hlr-Number....................identifier of ISDN-AddressString
    DEFINED in MAP-MS-DataTypes       :      94

hlr-Number....................identifier of ISDN-AddressString
    DEFINED in MAP-MS-DataTypes       :     319

hlr-Number....................identifier of ISDN-AddressString
    DEFINED in MAP-MS-DataTypes       :     329

hold..........................value reference SS-Code, '01000010'B
    DEFINED in MAP-SS-Code            :      68

ho-NumberNotRequired..........identifier of NULL
    DEFINED in MAP-MS-DataTypes       :     135

illegalEquipment..............value reference IllegalEquipment, CHOICE VALUE
    DEFINED in MAP-Protocol           :     265

IllegalEquipment..............type reference ERROR
    DEFINED in MAP-Errors             :     136
       USED in MAP-Protocol           :      95    265
       USED in MAP-SupplementaryServi :      52    237    251
       USED in MAP-ShortMessageServic :      35     92
       USED in MAP-Errors             :      30

illegalSS-Operation...........value reference IllegalSS-Operation, CHOICE VALUE
    DEFINED in MAP-Protocol           :     298

IllegalSS-Operation...........type reference ERROR
    DEFINED in MAP-Errors             :     184
       USED in MAP-Protocol           :     108    298
       USED in MAP-SupplementaryServi :      40     96    122    148    175    200
       USED in MAP-Errors             :      51

illegalSubscriber.............value reference IllegalSubscriber, CHOICE VALUE
    DEFINED in MAP-Protocol           :     264

IllegalSubscriber.............type reference ERROR
    DEFINED in MAP-Errors             :     134
       USED in MAP-Protocol           :      94    264
       USED in MAP-SupplementaryServi :      51    236    250
       USED in MAP-ShortMessageServic :      34     91
       USED in MAP-Errors             :      29

imei..........................identifier of IMEI
    DEFINED in MAP-MobileServiceOpera :     228

IMEI..........................type reference TBCD-STRING
```

```
       DEFINED in MAP-CommonDataTypes    :    195
         USED in MAP-MobileServiceOpera :    97    228
         USED in MAP-MS-DataTypes       :    62
         USED in MAP-CommonDataTypes    :    27


imsi.........................identifier of IMSI
     DEFINED in MAP-OperationAndMainte :    87


imsi.........................identifier of IMSI
     DEFINED in MAP-ShortMessageServic :   116


imsi.........................identifier of IMSI
     DEFINED in MAP-MS-DataTypes       :    81


imsi.........................identifier of IMSI
     DEFINED in MAP-MS-DataTypes       :    98


imsi.........................identifier of IMSI
     DEFINED in MAP-MS-DataTypes       :   102


imsi.........................identifier of IMSI
     DEFINED in MAP-MS-DataTypes       :   107


imsi.........................identifier of IMSI
     DEFINED in MAP-MS-DataTypes       :   112


imsi.........................identifier of [0] IMSI
     DEFINED in MAP-MS-DataTypes       :   220


imsi.........................identifier of [0] IMSI
     DEFINED in MAP-MS-DataTypes       :   230


imsi.........................identifier of [0] IMSI
     DEFINED in MAP-MS-DataTypes       :   298


imsi.........................identifier of IMSI
     DEFINED in MAP-MS-DataTypes       :   324


IMSI.........................type reference TBCD-STRING
     DEFINED in MAP-CommonDataTypes    :   186
       USED in MAP-OperationAndMainte  :    45    87
       USED in MAP-ShortMessageServic  :    58   116
       USED in MAP-MS-DataTypes        :    60    81    98   102   107   112
                                            185   220   230   298   324
       USED in MAP-CommonDataTypes     :    24   192   208
       USED in MAP-OM-DataTypes        :    23    33    46    51
       USED in MAP-CH-DataTypes        :    28    52    63
       USED in MAP-SS-DataTypes        :    37   281
       USED in MAP-SM-DataTypes        :    27    50    71   114


imsi.........................identifier of [0] IMSI
     DEFINED in MAP-CommonDataTypes    :   192


imsi.........................identifier of [0] IMSI
     DEFINED in MAP-OM-DataTypes       :    33


imsi.........................identifier of [0] IMSI
     DEFINED in MAP-OM-DataTypes       :    46


imsi.........................identifier of [0] IMSI
     DEFINED in MAP-OM-DataTypes       :    51


imsi.........................identifier of IMSI
     DEFINED in MAP-CH-DataTypes       :    52


imsi.........................identifier of [0] IMSI
     DEFINED in MAP-CH-DataTypes       :    63


imsi.........................identifier of IMSI
     DEFINED in MAP-SS-DataTypes       :   281


imsi.........................identifier of IMSI
     DEFINED in MAP-SM-DataTypes       :    50


imsi.........................identifier of [0] IMSI
     DEFINED in MAP-SM-DataTypes       :    71


imsi.........................identifier of [0] IMSI
     DEFINED in MAP-SM-DataTypes       :   114


imsi-WithLMSI................identifier of IMSI-WithLMSI
```

```
          DEFINED in MAP-MS-DataTypes     :      99


IMSI-WithLMSI.................type reference SEQUENCE
     DEFINED in MAP-MS-DataTypes     :     106
        USED in MAP-MS-DataTypes     :      99


incomingCallsBarredWithinCUG..identifier of Named Number, 0
   DEFINED in MAP-ER-DataTypes     :      50


incorrectTransactionPortion...identifier of Named Number, 3
   DEFINED in TCAPMessages         :     106


informServiceCentre.........value reference InformServiceCentre, CHOICE VALUE
   DEFINED in MAP-Protocol         :     235


InformServiceCentre...........type reference OPERATION
     DEFINED in MAP-ShortMessageServic :     131
        USED in MAP-Protocol         :      76    235
        USED in MAP-ShortMessageServic :      19


informServiceCentreArg........identifier of InformServiceCentreArg
   DEFINED in MAP-ShortMessageServic :     133


InformServiceCentreArg........type reference SEQUENCE
     DEFINED in MAP-SM-DataTypes     :     103
        USED in MAP-ShortMessageServic :      51    133
        USED in MAP-SM-DataTypes     :      19


initiatingRelease.............identifier of Named Number, 4
   DEFINED in TCAPMessages         :     187


insertSubscriberData..........value reference InsertSubscriberData, CHOICE VALUE
   DEFINED in MAP-Protocol         :     169


InsertSubscriberData..........type reference OPERATION
     DEFINED in MAP-MobileServiceOpera :     254
        USED in MAP-Protocol         :      26    169
        USED in MAP-MobileServiceOpera :      37


insertSubscriberDataArg.......identifier of InsertSubscriberDataArg
   DEFINED in MAP-MobileServiceOpera :     256


InsertSubscriberDataArg.......type reference SEQUENCE
     DEFINED in MAP-MS-DataTypes     :     229
        USED in MAP-MobileServiceOpera :      84    256
        USED in MAP-MS-DataTypes     :      40


insertSubscriberDataRes.......identifier of InsertSubscriberDataRes
   DEFINED in MAP-MobileServiceOpera :     258


InsertSubscriberDataRes.......type reference SEQUENCE
     DEFINED in MAP-MS-DataTypes     :     281
        USED in MAP-MobileServiceOpera :      85    258
        USED in MAP-MS-DataTypes     :      41


interBSS.....................identifier of Named Number, 0
   DEFINED in MAP-OM-DataTypes     :      65


interCUG-Restrictions.........identifier of InterCUG-Restrictions
   DEFINED in MAP-SS-DataTypes     :     185


InterCUG-Restrictions.........type reference OCTET STRING
     DEFINED in MAP-SS-DataTypes     :     188
        USED in MAP-SS-DataTypes     :     185


internationalOGCallsBarred....identifier of Named Number, 1
   DEFINED in MAP-MS-DataTypes     :     261


internationalOGCallsNotToHPLMN-CountryBa identifier of Named Number, 2
   DEFINED in MAP-MS-DataTypes     :     262


interrogateSS.......value reference InterrogateSS, CHOICE VALUE
   DEFINED in MAP-Protocol         :     207


InterrogateSS................type reference OPERATION
     DEFINED in MAP-SupplementaryServi :     182
        USED in MAP-Protocol         :      57    207
        USED in MAP-SupplementaryServi :      18


interrogateSS-Res.............identifier of InterrogateSS-Res
   DEFINED in MAP-SupplementaryServi :     186
```

```
InterrogateSS-Res.............type reference CHOICE
    DEFINED in MAP-SS-DataTypes    :    231
       USED in MAP-SupplementaryServi :    60    186
       USED in MAP-SS-DataTypes    :    21


intraBSS......................identifier of Named Number, 1
    DEFINED in MAP-OM-DataTypes    :    66


intraCUG-Options..............identifier of IntraCUG-Options
    DEFINED in MAP-SS-DataTypes    :    163


IntraCUG-Options..............type reference ENUMERATED
    DEFINED in MAP-SS-DataTypes    :    172
       USED in MAP-SS-DataTypes    :    163


invalidFormat.................identifier of Named Number, 1
    DEFINED in MAP-ER-DataTypes    :    65


invalidSME-Address............identifier of Named Number, 5
    DEFINED in MAP-ER-DataTypes    :    84


invalidTargetBaseStation.     value reference InvalidTargetBaseStation, CHOICE VALUE
    DEFINED in MAP-Protocol    :    274


InvalidTargetBaseStation......type reference ERROR
    DEFINED in MAP-Errors    :    145
       USED in MAP-Protocol    :    98    274
       USED in MAP-MobileServiceOpera :    60    167    207
       USED in MAP-Errors    :    35


invoke........................identifier of [1] IMPLICIT Invoke
    DEFINED in TCAPMessages    :    125


Invoke........................type reference SEQUENCE
    DEFINED in TCAPMessages    :    133
       USED in TCAPMessages    :    125


invokeID......................identifier of InvokeIdType
    DEFINED in TCAPMessages    :    134


invokeID......................identifier of InvokeIdType
    DEFINED in TCAPMessages    :    145


invokeID......................identifier of InvokeIdType
    DEFINED in TCAPMessages    :    157


invokeID......................identifier of CHOICE
    DEFINED in TCAPMessages    :    166


InvokeIdType..................type reference INTEGER
    DEFINED in TCAPMessages    :    175
       USED in TCAPMessages    :    47    134    135    145    157    167


invokeProblem.................identifier of [1] IMPLICIT InvokeProblem
    DEFINED in TCAPMessages    :    171


InvokeProblem.................type reference INTEGER
    DEFINED in TCAPMessages    :    183
       USED in TCAPMessages    :    171


ISDN-AddressString............type reference AddressString
    DEFINED in MAP-CommonDataTypes    :    112
       USED in MAP-OperationAndMainte :    44    85
       USED in MAP-CallHandlingOperat :    45    76
       USED in MAP-ShortMessageServic :    57    103
       USED in MAP-MS-DataTypes    :    58    83    88    94    103    163
                                          168    173    180    235    319    329
       USED in MAP-CommonDataTypes    :    17    204    206
       USED in MAP-CH-DataTypes    :    26    44    59    64    66    67
       USED in MAP-SS-DataTypes    :    34    83    233    282    293
       USED in MAP-SM-DataTypes    :    25    42    73    80    86    99
                                          104


ISDN-SubaddressString.........type reference OCTET STRING
    DEFINED in MAP-CommonDataTypes    :    118
       USED in MAP-CommonDataTypes    :    18
       USED in MAP-SS-DataTypes    :    35    56    84    295


kc............................identifier of Kc
    DEFINED in MAP-MS-DataTypes    :    122
```

```
Kc...........................type reference OCTET STRING
    DEFINED in MAP-MS-DataTypes      :    129
      USED in MAP-MS-DataTypes       :    122   145

kc...........................identifier of [12] Kc
    DEFINED in MAP-MS-DataTypes      :    145

Ki...........................type reference OCTET STRING
    DEFINED in MAP-MS-DataTypes      :    189
      USED in MAP-MS-DataTypes       :    223

ki...........................identifier of [4] Ki
    DEFINED in MAP-MS-DataTypes      :    223

linkedID.....................identifier of [0] IMPLICIT InvokeIdType
    DEFINED in TCAPMessages          :    135

linkedResponseUnexpected......identifier of Named Number, 6
    DEFINED in TCAPMessages          :    189

lmsi.........................identifier of [10] LMSI
    DEFINED in MAP-MS-DataTypes      :     84

lmsi.........................identifier of LMSI
    DEFINED in MAP-MS-DataTypes      :    108

lmsi.........................identifier of LMSI
    DEFINED in MAP-MS-DataTypes      :    325

LMSI.........................type reference OCTET STRING
    DEFINED in MAP-CommonDataTypes   :    217
      USED in MAP-MS-DataTypes       :     65    84    108   325
      USED in MAP-CommonDataTypes    :     30
      USED in MAP-CH-DataTypes       :     29    68
      USED in MAP-SM-DataTypes       :     29    59    72

lmsi.........................identifier of [4] LMSI
    DEFINED in MAP-CH-DataTypes      :     68

lmsi.........................identifier of LMSI
    DEFINED in MAP-SM-DataTypes      :     59

lmsi.........................identifier of [1] LMSI
    DEFINED in MAP-SM-DataTypes      :     72

locationInfo.................identifier of LocationInfo
    DEFINED in MAP-MS-DataTypes      :     82

LocationInfo.................type reference CHOICE
    DEFINED in MAP-CommonDataTypes   :    203
      USED in MAP-MS-DataTypes       :     63    82
      USED in MAP-CommonDataTypes    :     28
      USED in MAP-SM-DataTypes       :     28    58

locationInfo.................identifier of LocationInfo
    DEFINED in MAP-SM-DataTypes      :     58

locationInfoWithLMSI.........identifier of [0] LocationInfoWithLMSI
    DEFINED in MAP-SM-DataTypes      :     51

LocationInfoWithLMSI.........type reference SEQUENCE
    DEFINED in MAP-SM-DataTypes      :     57
      USED in MAP-SM-DataTypes       :     51

mah..........................value reference SS-Code, '00110010'B
    DEFINED in MAP-SS-Code           :     59

MAP-BS-Code..................module reference
    DEFINED in MAP-BS-Code           :      1
      USED in MAP-CommonDataTypes    :     51

MAP-CallHandlingOperations....module reference
    DEFINED in MAP-CallHandlingOperat :      1
      USED in MAP-Protocol           :     48

MAP-CH-DataTypes.............module reference
    DEFINED in MAP-CH-DataTypes      :      1
      USED in MAP-CallHandlingOperat :     41

MAP-CommonDataTypes..........module reference
```

```
        DEFINED in MAP-CommonDataTypes    :      1
            USED in MAP-MobileServiceOpera  :     98
            USED in MAP-OperationAndMainte  :     46
            USED in MAP-CallHandlingOperat  :     46
            USED in MAP-ShortMessageServic  :     59
            USED in MAP-Errors              :     90
            USED in MAP-MS-DataTypes        :     72
            USED in MAP-OM-DataTypes        :     25
            USED in MAP-CH-DataTypes        :     30
            USED in MAP-SS-DataTypes        :     41
            USED in MAP-SM-DataTypes        :     30
            USED in MAP-ER-DataTypes        :     30

MAP-Errors...................module reference
        DEFINED in MAP-Errors              :      1
            USED in MAP-Protocol            :    121
            USED in MAP-MobileServiceOpera  :     64
            USED in MAP-OperationAndMainte  :     32
            USED in MAP-CallHandlingOperat  :     35
            USED in MAP-SupplementaryServi  :     53
            USED in MAP-ShortMessageServic  :     42

MAP-ER-DataTypes..............module reference
        DEFINED in MAP-ER-DataTypes        :      1
            USED in MAP-Errors              :     85

MAP-MobileServiceOperations...module reference
        DEFINED in MAP-MobileServiceOpera  :      1
            USED in MAP-Protocol            :     31

MAP-MS-DataTypes..............module reference
        DEFINED in MAP-MS-DataTypes        :      1
            USED in MAP-MobileServiceOpera  :     91

MAP-OM-DataTypes..............module reference
        DEFINED in MAP-OM-DataTypes        :      1
            USED in MAP-OperationAndMainte  :     40

MAP-OperationAndMaintenanceOperations...module reference
        DEFINED in MAP-OperationAndMainte  :      1
            USED in MAP-Protocol            :     41

MAP-Protocol..................module reference
        DEFINED in MAP-Protocol            :      1

MAP-ShortMessageServiceOperations.......module reference
        DEFINED in MAP-ShortMessageServic  :      1
            USED in MAP-Protocol            :     78

MAP-SM-DataTypes..............module reference
        DEFINED in MAP-SM-DataTypes        :      1
            USED in MAP-ShortMessageServic  :     53

MAP-SS-Code...................module reference
        DEFINED in MAP-SS-Code             :      1
            USED in MAP-SupplementaryServi  :     72
            USED in MAP-SS-DataTypes        :     46
            USED in MAP-ER-DataTypes        :     35

MAP-SS-DataTypes..............module reference
        DEFINED in MAP-SS-DataTypes        :      1
            USED in MAP-SupplementaryServi  :     67
            USED in MAP-Errors              :     75
            USED in MAP-MS-DataTypes        :     54
            USED in MAP-CH-DataTypes        :     22
            USED in MAP-ER-DataTypes        :     24

MAP-SupplementaryServiceOperations......module reference
        DEFINED in MAP-SupplementaryServi  :      1
            USED in MAP-Protocol            :     65

MAP-TS-Code...................module reference
        DEFINED in MAP-TS-Code             :      1
            USED in MAP-CommonDataTypes     :     46
            USED in MAP-SM-DataTypes        :     35

maxAddressLength..............value reference INTEGER, 20
        DEFINED in MAP-CommonDataTypes     :    110
            USED in MAP-CommonDataTypes     :     69

maxISDN-AddressLength.........value reference INTEGER, 9
```

```
        DEFINED in MAP-CommonDataTypes   :     116
          USED in MAP-CommonDataTypes   :     113

maxISDN-SubaddressLength......value reference INTEGER, 21
        DEFINED in MAP-CommonDataTypes   :     156
          USED in MAP-CommonDataTypes   :     119

maxNumOfBasicServiceGroups....value reference INTEGER, 13
        DEFINED in MAP-CommonDataTypes   :     267
          USED in MAP-CommonDataTypes   :      41    264
          USED in MAP-SS-DataTypes       :      40     76    143    179

maxNumOfBasicServices.........value reference INTEGER, 70
        DEFINED in MAP-CommonDataTypes   :     262
          USED in MAP-CommonDataTypes   :     259

maxNumOfBearerServices........value reference INTEGER, 50
        DEFINED in MAP-CommonDataTypes   :     248
          USED in MAP-CommonDataTypes   :     245

maxNumOfCUG...................value reference INTEGER, 10
        DEFINED in MAP-SS-DataTypes      :     177
          USED in MAP-SS-DataTypes       :     157

maxNumOfHLR-Id...............value reference INTEGER, 50
        DEFINED in MAP-CommonDataTypes   :     215
          USED in MAP-CommonDataTypes   :     212

maxNumOfSentParameter.........value reference INTEGER, 6
        DEFINED in MAP-MS-DataTypes      :     225
          USED in MAP-MS-DataTypes       :     215

maxNumOfSS...................value reference INTEGER, 30
        DEFINED in MAP-SS-DataTypes      :     287
          USED in MAP-SS-DataTypes       :     284    289

maxNumOfTeleservices..........value reference INTEGER, 20
        DEFINED in MAP-CommonDataTypes   :     253
          USED in MAP-CommonDataTypes   :     250

maxNumOfZoneCodes.............value reference INTEGER, 10
        DEFINED in MAP-MS-DataTypes      :     279
          USED in MAP-MS-DataTypes       :     273

maxSignalInfoLength...........value reference INTEGER, 200
        DEFINED in MAP-CommonDataTypes   :     167
          USED in MAP-CommonDataTypes   :      21    165
          USED in MAP-SS-DataTypes       :      36    240

maxUSSD-StringLength..........value reference INTEGER, 160
        DEFINED in MAP-SS-DataTypes      :     262
          USED in MAP-SS-DataTypes       :     257

mcef-Set......................identifier of Named Number, 2
        DEFINED in MAP-SM-DataTypes      :     111

mci...........................value reference SS-Code, '00010101'B
        DEFINED in MAP-SS-Code           :      36

memoryAvailable...............identifier of Named Number, 1
        DEFINED in MAP-SM-DataTypes      :     120

memoryCapacityExceeded........identifier of Named Number, 0
        DEFINED in MAP-SM-DataTypes      :      94

memoryCapacityExceeded........identifier of Named Number, 0
        DEFINED in MAP-ER-DataTypes      :      79

MessageType...................type reference CHOICE
        DEFINED in TCAPMessages          :      51
          USED in TCAPMessages           :      47

messageWaitingListFull...value reference MessageWaitingListFull, CHOICE VALUE
        DEFINED in MAP-Protocol          :     319

MessageWaitingListFull........type reference ERROR
        DEFINED in MAP-Errors            :     225
          USED in MAP-Protocol           :     120    319
          USED in MAP-ShortMessageServic :      41    112
          USED in MAP-Errors             :      65
```

```
networkResource...............identifier of NetworkResource
   DEFINED in MAP-Errors              :     100

networkResource...............identifier of NetworkResource
   DEFINED in MAP-MS-DataTypes        :     316

NetworkResource...............type reference ENUMERATED
   DEFINED in MAP-CommonDataTypes  :     229
      USED in MAP-Errors            :      89    100
      USED in MAP-MS-DataTypes      :      67    316
      USED in MAP-CommonDataTypes   :      32

networkSignalInfo.............identifier of [10] ExternalSignalInfo
   DEFINED in MAP-CH-DataTypes        :      48

networkSignalInfo.............identifier of [6] ExternalSignalInfo
   DEFINED in MAP-CH-DataTypes        :      70

newPassword...................identifier of Password
   DEFINED in MAP-SupplementaryServi :     259

newPasswordsMismatch..........identifier of Named Number, 2
   DEFINED in MAP-ER-DataTypes        :      66

noCUG-Restrictions............identifier of Named Number, 0
   DEFINED in MAP-SS-DataTypes        :     173

noHandoverNumberAvailable.....value reference NoHandoverNumberAvailable, CHOICE VALUE
   DEFINED in MAP-Protocol            :     276

NoHandoverNumberAvailable.....type reference ERROR
   DEFINED in MAP-Errors             :     149
      USED in MAP-Protocol           :     100    276
      USED in MAP-MobileServiceOpera :      62    156    169
      USED in MAP-Errors             :      37

noRadioResourceAvailable......value reference NoRadioResourceAvailable, CHOICE VALUE
   DEFINED in MAP-Protocol            :     275

NoRadioResourceAvailable......type reference ERROR
   DEFINED in MAP-Errors             :     147
      USED in MAP-Protocol           :      99    275
      USED in MAP-MobileServiceOpera :      61    168
      USED in MAP-Errors             :      36

noReplyConditionTime..........identifier of [5] NoReplyConditionTime
   DEFINED in MAP-SS-DataTypes        :      58

NoReplyConditionTime..........type reference INTEGER
   DEFINED in MAP-SS-DataTypes       :      61
      USED in MAP-SS-DataTypes       :      58     87

noReplyConditionTime..........identifier of [7] NoReplyConditionTime
   DEFINED in MAP-SS-DataTypes        :      87

noRoamingNumberAvailable......value reference NoRoamingNumberAvailable, CHOICE VALUE
   DEFINED in MAP-Protocol            :     289

NoRoamingNumberAvailable......type reference ERROR
   DEFINED in MAP-Errors             :     161
      USED in MAP-Protocol           :     103    289
      USED in MAP-CallHandlingOperat :      30     83
      USED in MAP-Errors             :      44

noSM-RP-DA....................identifier of [5] NULL
   DEFINED in MAP-SM-DataTypes        :      76

noSM-RP-OA....................identifier of [5] NULL
   DEFINED in MAP-SM-DataTypes        :      82

noteInternalHandover..........value reference NoteInternalHandover, CHOICE VALUE
   DEFINED in MAP-Protocol            :     188

NoteInternalHandover..........type reference OPERATION
   DEFINED in MAP-OperationAndMainte :      79
      USED in MAP-Protocol           :      39    188
      USED in MAP-OperationAndMainte :      16

noteInternalHO-Arg............identifier of NoteInternalHO-Arg
   DEFINED in MAP-OperationAndMainte :      81
```

```
NoteInternalHO-Arg............type reference SEQUENCE
    DEFINED in MAP-OM-DataTypes      :    59
        USED in MAP-OperationAndMainte :    39   81
        USED in MAP-OM-DataTypes     :    17


noteSubscriberPresent.........value reference NoteSubscriberPresent, CHOICE VALUE
    DEFINED in MAP-Protocol          :   229


NoteSubscriberPresent.........type reference OPERATION
    DEFINED in MAP-ShortMessageServic :   114
        USED in MAP-Protocol         :    73   229
        USED in MAP-ShortMessageServic :   16


not-derivable.................identifier of NULL
    DEFINED in TCAPMessages          :   168


numberChanged.................value reference NumberChanged, CHOICE VALUE
    DEFINED in MAP-Protocol          :   254


NumberChanged.................type reference ERROR
    DEFINED in MAP-Errors            :   114
        USED in MAP-Protocol         :    88   254
        USED in MAP-CallHandlingOperat :   27   62
        USED in MAP-Errors           :    21


NumberOfForwarding............type reference INTEGER
    DEFINED in MAP-CH-DataTypes      :    41
        USED in MAP-CH-DataTypes     :    47


numberOfForwarding............identifier of [2] NumberOfForwarding
    DEFINED in MAP-CH-DataTypes      :    47


numberOfPW-AttemptsViolation.........value reference NumberOfPW-AttemptsViolation, CHOICE
VALUE
    DEFINED in MAP-Protocol          :   309


NumberOfPW-AttemptsViolation..type reference ERROR
    DEFINED in MAP-Errors            :   214
        USED in MAP-Protocol         :   117   309
        USED in MAP-SupplementaryServi :   47  153   179   270
        USED in MAP-Errors           :    60


odb-Data......................identifier of [8] ODB-Data
    DEFINED in MAP-MS-DataTypes      :   241


ODB-Data......................type reference SEQUENCE
    DEFINED in MAP-MS-DataTypes      :   254
        USED in MAP-MS-DataTypes     :   241


odb-GeneralData...............identifier of ODB-GeneralData
    DEFINED in MAP-MS-DataTypes      :   255


ODB-GeneralData...............type reference BIT STRING
    DEFINED in MAP-MS-DataTypes      :   259
        USED in MAP-MS-DataTypes     :   255   285


odb-GeneralData...............identifier of [4] ODB-GeneralData
    DEFINED in MAP-MS-DataTypes      :   285


odb-HPLMN-Data................identifier of ODB-HPLMN-Data
    DEFINED in MAP-MS-DataTypes      :   256


ODB-HPLMN-Data................type reference BIT STRING
    DEFINED in MAP-MS-DataTypes      :   267
        USED in MAP-MS-DataTypes     :   256


omc-Id........................identifier of [3] AddressString
    DEFINED in MAP-OM-DataTypes      :    36


omc-Id........................identifier of [3] AddressString
    DEFINED in MAP-OM-DataTypes      :    54


operationCode.................identifier of OPERATION
    DEFINED in TCAPMessages          :   136


operationCode.................identifier of OPERATION
    DEFINED in TCAPMessages          :   147


operatorBarring...............identifier of Named Number, 1
    DEFINED in MAP-ER-DataTypes      :    47
```

```
operatorDeterminedBarring.....identifier of Named Number, 1
   DEFINED in MAP-MS-DataTypes       :    252

operatorDeterminedBarring.....identifier of Named Number, 3
   DEFINED in MAP-ER-DataTypes       :     43

originatingEntityNumber.......identifier of ISDN-AddressString
   DEFINED in MAP-SS-DataTypes       :    282

OrigTransactionID...type reference  [APPLICATION 8] IMPLICIT TransactionID
   DEFINED in TCAPMessages           :     97
      USED in TCAPMessages           :     61    69

otid..........................identifier of OrigTransactionID
   DEFINED in TCAPMessages           :     61

otid..........................identifier of OrigTransactionID
   DEFINED in TCAPMessages           :     69

overrideCategory..............identifier of [1] OverrideCategory
   DEFINED in MAP-SS-DataTypes       :    210

OverrideCategory..............type reference ENUMERATED
   DEFINED in MAP-SS-DataTypes       :    217
      USED in MAP-SS-DataTypes       :    210

overrideDisabled..............identifier of Named Number, 1
   DEFINED in MAP-SS-DataTypes       :    219

overrideEnabled...............identifier of Named Number, 0
   DEFINED in MAP-SS-DataTypes       :    218

padAccessCA-1200bps...........value reference BearerServiceCode, '00100010'B
   DEFINED in MAP-BS-Code            :     49

padAccessCA-1200-75bps........value reference BearerServiceCode, '00100011'B
   DEFINED in MAP-BS-Code            :     50

padAccessCA-2400bps...........value reference BearerServiceCode, '00100100'B
   DEFINED in MAP-BS-Code            :     51

padAccessCA-300bps............value reference BearerServiceCode, '00100001'B
   DEFINED in MAP-BS-Code            :     48

padAccessCA-4800bps...........value reference BearerServiceCode, '00100101'B
   DEFINED in MAP-BS-Code            :     52

padAccessCA-9600bps...........value reference BearerServiceCode, '00100110'B
   DEFINED in MAP-BS-Code            :     53

parameter.....................identifier of ANY DEFINED BY operationCode
   DEFINED in TCAPMessages           :    137

parameter.....................identifier of ANY DEFINED BY operationCode
   DEFINED in TCAPMessages           :    148

parameter.....................identifier of ANY DEFINED BY errorCode
   DEFINED in TCAPMessages           :    159

Password......................type reference NumericString
   DEFINED in MAP-SS-DataTypes       :    264
      USED in MAP-SupplementaryServi :     64    259    279
      USED in MAP-SS-DataTypes       :     25

performHandover...............value reference PerformHandover, CHOICE VALUE
   DEFINED in MAP-Protocol           :    139

PerformHandover...............type reference OPERATION
   DEFINED in MAP-MobileServiceOpera :    158
      USED in MAP-Protocol           :     16    139
      USED in MAP-MobileServiceOpera :     21

performHO-Arg.................identifier of PerformHO-Arg
   DEFINED in MAP-MobileServiceOpera :    160

PerformHO-Arg.................type reference SEQUENCE
   DEFINED in MAP-MS-DataTypes       :    139
      USED in MAP-MobileServiceOpera :     73    160
      USED in MAP-MS-DataTypes       :     26
```

```
plmn-specificBS-D.............value reference BearerServiceCode, '11011101'B
   DEFINED in MAP-BS-Code       :     104

plmn-specificBS-E.............value reference BearerServiceCode, '11011110'B
   DEFINED in MAP-BS-Code       :     105

plmn-specificBS-F.............value reference BearerServiceCode, '11011111'B
   DEFINED in MAP-BS-Code       :     106

plmn-specificSS-1.............value reference SS-Code, '11110001'B
   DEFINED in MAP-SS-Code       :     123

plmn-specificSS-2.............value reference SS-Code, '11110010'B
   DEFINED in MAP-SS-Code       :     124

plmn-specificSS-3.............value reference SS-Code, '11110011'B
   DEFINED in MAP-SS-Code       :     125

plmn-specificSS-4.............value reference SS-Code, '11110100'B
   DEFINED in MAP-SS-Code       :     126

plmn-specificSS-5.............value reference SS-Code, '11110101'B
   DEFINED in MAP-SS-Code       :     127

plmn-specificSS-6.............value reference SS-Code, '11110110'B
   DEFINED in MAP-SS-Code       :     128

plmn-specificSS-7.............value reference SS-Code, '11110111'B
   DEFINED in MAP-SS-Code       :     129

plmn-specificSS-8.............value reference SS-Code, '11111000'B
   DEFINED in MAP-SS-Code       :     130

plmn-specificSS-9.............value reference SS-Code, '11111001'B
   DEFINED in MAP-SS-Code       :     131

plmn-specificSS-A.............value reference SS-Code, '11111010'B
   DEFINED in MAP-SS-Code       :     132

plmn-specificSS-B.............value reference SS-Code, '11111011'B
   DEFINED in MAP-SS-Code       :     133

plmn-specificSS-C.............value reference SS-Code, '11111100'B
   DEFINED in MAP-SS-Code       :     134

plmn-specificSS-D.............value reference SS-Code, '11111101'B
   DEFINED in MAP-SS-Code       :     135

plmn-specificSS-E.............value reference SS-Code, '11111110'B
   DEFINED in MAP-SS-Code       :     136

plmn-specificSS-F.............value reference SS-Code, '11111111'B
   DEFINED in MAP-SS-Code       :     137

plmn-specificTS-1.............value reference TeleserviceCode, '11010001'B
   DEFINED in MAP-TS-Code       :      53

plmn-specificTS-2.............value reference TeleserviceCode, '11010010'B
   DEFINED in MAP-TS-Code       :      54

plmn-specificTS-3.............value reference TeleserviceCode, '11010011'B
   DEFINED in MAP-TS-Code       :      55

plmn-specificTS-4.............value reference TeleserviceCode, '11010100'B
   DEFINED in MAP-TS-Code       :      56

plmn-specificTS-5.............value reference TeleserviceCode, '11010101'B
   DEFINED in MAP-TS-Code       :      57

plmn-specificTS-6.............value reference TeleserviceCode, '11010110'B
   DEFINED in MAP-TS-Code       :      58

plmn-specificTS-7.............value reference TeleserviceCode, '11010111'B
   DEFINED in MAP-TS-Code       :      59

plmn-specificTS-8.............value reference TeleserviceCode, '11011000'B
   DEFINED in MAP-TS-Code       :      60

plmn-specificTS-9.............value reference TeleserviceCode, '11011001'B
   DEFINED in MAP-TS-Code       :      61
```

```
plmn-specificTS-A.............value reference TeleserviceCode, '11011010'B
   DEFINED in MAP-TS-Code          :      62

plmn-specificTS-B.............value reference TeleserviceCode, '11011011'B
   DEFINED in MAP-TS-Code          :      63

plmn-specificTS-C.............value reference TeleserviceCode, '11011100'B
   DEFINED in MAP-TS-Code          :      64

plmn-specificTS-D.............value reference TeleserviceCode, '11011101'B
   DEFINED in MAP-TS-Code          :      65

plmn-specificTS-E.............value reference TeleserviceCode, '11011110'B
   DEFINED in MAP-TS-Code          :      66

plmn-specificTS-F.............value reference TeleserviceCode, '11011111'B
   DEFINED in MAP-TS-Code          :      67

preferentialCUG-Indicator.....identifier of CUG-Index
   DEFINED in MAP-SS-DataTypes     :     184

premiumRateEntertainementOGCallsBarred..identifier of Named Number, 4
   DEFINED in MAP-MS-DataTypes     :     264

premiumRateInformationOGCallsBarred.....identifier of Named Number, 3
   DEFINED in MAP-MS-DataTypes     :     263

prepareHandover...............value reference PrepareHandover, CHOICE VALUE
   DEFINED in MAP-Protocol         :     141

PrepareHandover...............type reference OPERATION
   DEFINED in MAP-MobileServiceOpera :   147
      USED in MAP-Protocol         :      17     141
      USED in MAP-MobileServiceOpera :     22

prepareHO-Arg.................identifier of PrepareHO-Arg
   DEFINED in MAP-MobileServiceOpera :   149

PrepareHO-Arg.................type reference SEQUENCE
   DEFINED in MAP-MS-DataTypes     :     133
      USED in MAP-MobileServiceOpera :     74     149
      USED in MAP-MS-DataTypes     :      23

prepareHO-Res.................identifier of PrepareHO-Res
   DEFINED in MAP-MobileServiceOpera :   151

PrepareHO-Res.................type reference SEQUENCE
   DEFINED in MAP-MS-DataTypes     :     162
      USED in MAP-MobileServiceOpera :     76     151
      USED in MAP-MS-DataTypes     :      24

prepareSubsequentHandover.....value reference PrepareSubsequentHandover, CHOICE VALUE
   DEFINED in MAP-Protocol         :     149

PrepareSubsequentHandover.....type reference OPERATION
   DEFINED in MAP-MobileServiceOpera :   187
      USED in MAP-Protocol         :      22     149
      USED in MAP-MobileServiceOpera :     27

prepareSubsequentHO-Arg.......identifier of PrepareSubsequentHO-Arg
   DEFINED in MAP-MobileServiceOpera :   189

PrepareSubsequentHO-Arg.......type reference SEQUENCE
   DEFINED in MAP-MS-DataTypes     :     171
      USED in MAP-MobileServiceOpera :     78     189
      USED in MAP-MS-DataTypes     :      25

previousRoamingNumber.........identifier of [3] ISDN-AddressString
   DEFINED in MAP-CH-DataTypes     :      67

problem.......................identifier of CHOICE
   DEFINED in TCAPMessages         :     169

processAccessSignalling.......value reference ProcessAccessSignalling, CHOICE VALUE
   DEFINED in MAP-Protocol         :     144

ProcessAccessSignalling.......type reference OPERATION
   DEFINED in MAP-MobileServiceOpera :   179
      USED in MAP-Protocol         :      19     144
      USED in MAP-MobileServiceOpera :     24
```

```
processUnstructuredSS-Data....value reference ProcessUnstructuredSS-Data, CHOICE VALUE
    DEFINED in MAP-Protocol          :     208

ProcessUnstructuredSS-Data....type reference OPERATION
    DEFINED in MAP-SupplementaryServi :    203
        USED in MAP-Protocol          :     58     208
        USED in MAP-SupplementaryServi :    19

processUnstructuredSS-Request.......value reference ProcessUnstructuredSS-Request, CHOICE
VALUE
    DEFINED in MAP-Protocol          :     211

ProcessUnstructuredSS-Request.....type reference OPERATION
    DEFINED in MAP-SupplementaryServi :    213
        USED in MAP-Protocol          :     59     211
        USED in MAP-SupplementaryServi :    20

protocolId....................identifier of ProtocolId
    DEFINED in MAP-CommonDataTypes   :     159

ProtocolId....................type reference ENUMERATED
    DEFINED in MAP-CommonDataTypes   :     175
        USED in MAP-CommonDataTypes   :     159

provideRoamingNumber.......value reference ProvideRoamingNumber, CHOICE VALUE
    DEFINED in MAP-Protocol          :     198

ProvideRoamingNumber..........type reference OPERATION
    DEFINED in MAP-CallHandlingOperat :    72
        USED in MAP-Protocol          :     47     198
        USED in MAP-CallHandlingOperat :    14

provideRoamingNumberArg.......identifier of ProvideRoamingNumberArg
    DEFINED in MAP-CallHandlingOperat :    74

ProvideRoamingNumberArg.......type reference SEQUENCE
    DEFINED in MAP-CH-DataTypes      :     62
        USED in MAP-CallHandlingOperat :    40     74
        USED in MAP-CH-DataTypes      :     16

provisionedSS.................identifier of [7] SS-InfoList
    DEFINED in MAP-MS-DataTypes      :     240

purgeMS.......................value reference PurgeMS, CHOICE VALUE
    DEFINED in MAP-Protocol          :     131

PurgeMS.......................type reference OPERATION
    DEFINED in MAP-MobileServiceOpera :    131
        USED in MAP-Protocol          :     14     131
        USED in MAP-MobileServiceOpera :    17

purgeMS-Arg...................identifier of PurgeMS-Arg
    DEFINED in MAP-MobileServiceOpera :    133

PurgeMS-Arg...................type reference SEQUENCE
    DEFINED in MAP-MS-DataTypes      :     101
        USED in MAP-MobileServiceOpera :    71     133
        USED in MAP-MS-DataTypes      :     19

pvlr..........................identifier of Named Number, 3
    DEFINED in MAP-CommonDataTypes   :     233

pw-RegistrationFailure...value reference PW-RegistrationFailure, CHOICE VALUE
    DEFINED in MAP-Protocol          :     307

PW-RegistrationFailure........type reference ERROR
    DEFINED in MAP-Errors            :     208
        USED in MAP-Protocol          :     115     307
        USED in MAP-SupplementaryServi :    45     268
        USED in MAP-Errors            :     58

pw-RegistrationFailureCause...identifier of PW-RegistrationFailureCause
    DEFINED in MAP-Errors            :     210

PW-RegistrationFailureCause...type reference ENUMERATED
    DEFINED in MAP-ER-DataTypes      :     63
        USED in MAP-Errors            :     83     210
        USED in MAP-ER-DataTypes      :     18

p-abortCause..................identifier of P-AbortCause
    DEFINED in TCAPMessages          :     76
```

```
P-AbortCause..................type reference  [APPLICATION 10] IMPLICIT INTEGER
    DEFINED in TCAPMessages         :    102
       USED in TCAPMessages         :     76


rand..........................identifier of RAND
    DEFINED in MAP-MS-DataTypes     :    120


RAND..........................type reference OCTET STRING
    DEFINED in MAP-MS-DataTypes     :    125
       USED in MAP-MS-DataTypes     :    120


readyForSM....................value reference ReadyForSM, CHOICE VALUE
    DEFINED in MAP-Protocol         :    239


ReadyForSM....................type reference OPERATION
    DEFINED in MAP-ShortMessageServic :  135
       USED in MAP-Protocol         :     77   239
       USED in MAP-ShortMessageServic :   20


readyForSM-Arg................identifier of ReadyForSM-Arg
    DEFINED in MAP-ShortMessageServic :  137


ReadyForSM-Arg................type reference SEQUENCE
    DEFINED in MAP-SM-DataTypes     :    113
       USED in MAP-ShortMessageServic :   52   137
       USED in MAP-SM-DataTypes     :     20


reason........................identifier of CHOICE
    DEFINED in TCAPMessages         :     75


regionalSubscNotSupported.....identifier of Named Number, 3
    DEFINED in MAP-MS-DataTypes     :    295


regionalSubscriptionData......identifier of [10] ZoneCodeList
    DEFINED in MAP-MS-DataTypes     :    246


regionalSubscriptionIdentifier..........identifier of [5] ZoneCode
    DEFINED in MAP-MS-DataTypes     :    304


regionalSubscriptionResponse...identifier of [5] RegionalSubscriptionResponse
    DEFINED in MAP-MS-DataTypes     :    286


RegionalSubscriptionResponse............type reference ENUMERATED
    DEFINED in MAP-MS-DataTypes     :    291
       USED in MAP-MS-DataTypes     :    287   310


regionalSubscriptionResponse...identifier of [0] RegionalSubscriptionResponse
    DEFINED in MAP-MS-DataTypes     :    309


registerPassword..............value reference RegisterPassword, CHOICE VALUE
    DEFINED in MAP-Protocol         :    218


RegisterPassword..............type reference OPERATION
    DEFINED in MAP-SupplementaryServi :  255
       USED in MAP-Protocol         :     62   218
       USED in MAP-SupplementaryServi :   23


registerSS....................value reference RegisterSS, CHOICE VALUE
    DEFINED in MAP-Protocol         :    203


RegisterSS....................type reference OPERATION
    DEFINED in MAP-SupplementaryServi :   80
       USED in MAP-Protocol         :     53   203
       USED in MAP-SupplementaryServi :   13


registerSS-Arg................identifier of RegisterSS-Arg
    DEFINED in MAP-SupplementaryServi :   82


RegisterSS-Arg................type reference SEQUENCE
    DEFINED in MAP-SS-DataTypes     :     52
       USED in MAP-SupplementaryServi :   57    82
       USED in MAP-SS-DataTypes     :     14


reject........................identifier of [4] IMPLICIT Reject
    DEFINED in TCAPMessages         :    128


Reject........................type reference SEQUENCE
    DEFINED in TCAPMessages         :    165
       USED in TCAPMessages         :    128
```

```
reportSM-DeliveryStatus.value reference ReportSM-DeliveryStatus, CHOICE VALUE
   DEFINED in MAP-Protocol            :     228

ReportSM-DeliveryStatus.......type reference OPERATION
   DEFINED in MAP-ShortMessageServic :      99
      USED in MAP-Protocol           :      72    228
      USED in MAP-ShortMessageServic :      15

reportSM-DeliveryStatusArg....identifier of ReportSM-DeliveryStatusArg
   DEFINED in MAP-ShortMessageServic :     101

ReportSM-DeliveryStatusArg....type reference SEQUENCE
   DEFINED in MAP-SM-DataTypes        :      85
      USED in MAP-ShortMessageServic :      49    101
      USED in MAP-SM-DataTypes       :      17

requestAuthenticationSet......identifier of Named Number, 1
   DEFINED in MAP-MS-DataTypes        :     211

requestedBasicServiceViolatesCUG-Constraidentifier of Named Number, 5
   DEFINED in MAP-ER-DataTypes        :      52

requestIMSI...................identifier of Named Number, 0
   DEFINED in MAP-MS-DataTypes        :     210

requestKi.....................identifier of Named Number, 4
   DEFINED in MAP-MS-DataTypes        :     213

RequestParameter..............type reference ENUMERATED
   DEFINED in MAP-MS-DataTypes        :     209
      USED in MAP-MS-DataTypes       :     207

requestParameterList..........identifier of RequestParameterList
   DEFINED in MAP-MS-DataTypes        :     204

RequestParameterList..........type reference SEQUENCE OF
   DEFINED in MAP-MS-DataTypes        :     206
      USED in MAP-MS-DataTypes       :     204

requestSubscriberData.........identifier of Named Number, 2
   DEFINED in MAP-MS-DataTypes        :     212

reset.........................value reference Reset, CHOICE VALUE
   DEFINED in MAP-Protocol            :     175

Reset.........................type reference OPERATION
   DEFINED in MAP-MobileServiceOpera :     280
      USED in MAP-Protocol           :      28    175
      USED in MAP-MobileServiceOpera :      41

resetArg......................identifier of ResetArg
   DEFINED in MAP-MobileServiceOpera :     282

ResetArg......................type reference SEQUENCE
   DEFINED in MAP-MS-DataTypes        :     315
      USED in MAP-MobileServiceOpera :      88    282
      USED in MAP-MS-DataTypes       :      46

resourceLimitation............identifier of Named Number, 4
   DEFINED in TCAPMessages            :     107

resourceLimitation............identifier of Named Number, 3
   DEFINED in TCAPMessages            :     186

restoreData...................value reference RestoreData, CHOICE VALUE
   DEFINED in MAP-Protocol            :     178

RestoreData...................type reference OPERATION
   DEFINED in MAP-MobileServiceOpera :     286
      USED in MAP-Protocol           :      30    178
      USED in MAP-MobileServiceOpera :      43

restoreDataArg................identifier of RestoreDataArg
   DEFINED in MAP-MobileServiceOpera :     288

RestoreDataArg................type reference SEQUENCE
   DEFINED in MAP-MS-DataTypes        :     323
      USED in MAP-MobileServiceOpera :      89    288
      USED in MAP-MS-DataTypes       :      47

restoreDataRes................identifier of RestoreDataRes
```

```
    DEFINED in MAP-MobileServiceOpera :     290

RestoreDataRes................type reference SEQUENCE
    DEFINED in MAP-MS-DataTypes    :     328
      USED in MAP-MobileServiceOpera :     90    290
      USED in MAP-MS-DataTypes    :     48

result-RR.....................identifier of SEQUENCE
    DEFINED in TCAPMessages        :     146

returnError...................identifier of [3] IMPLICIT ReturnError
    DEFINED in TCAPMessages        :     127

ReturnError...................type reference SEQUENCE
    DEFINED in TCAPMessages        :     156
      USED in TCAPMessages        :     127

returnErrorProblem............identifier of [3] IMPLICIT ReturnErrorProblem
    DEFINED in TCAPMessages        :     173

ReturnErrorProblem............type reference INTEGER
    DEFINED in TCAPMessages        :     196
      USED in TCAPMessages        :     173

returnErrorUnexpected.........identifier of Named Number, 1
    DEFINED in TCAPMessages        :     197

ReturnResult..................type reference SEQUENCE
    DEFINED in TCAPMessages        :     144
      USED in TCAPMessages        :     126    129

returnResultLast..............identifier of [2] IMPLICIT ReturnResult
    DEFINED in TCAPMessages        :     126

returnResultNotLast...........identifier of [7] IMPLICIT ReturnResult
    DEFINED in TCAPMessages        :     129

returnResultProblem...........identifier of [2] IMPLICIT ReturnResultProblem
    DEFINED in TCAPMessages        :     172

ReturnResultProblem...........type reference INTEGER
    DEFINED in TCAPMessages        :     192
      USED in TCAPMessages        :     172

returnResultUnexpected........identifier of Named Number, 1
    DEFINED in TCAPMessages        :     193

roamingNotAllowed.............value reference RoamingNotAllowed, CHOICE VALUE
    DEFINED in MAP-Protocol        :     263

RoamingNotAllowed.............type reference ERROR
    DEFINED in MAP-Errors          :     127
      USED in MAP-Protocol         :     93    263
      USED in MAP-MobileServiceOpera :     59    117
      USED in MAP-Errors           :     28

roamingNotAllowedCause........identifier of RoamingNotAllowedCause
    DEFINED in MAP-Errors          :     129

RoamingNotAllowedCause........type reference ENUMERATED
    DEFINED in MAP-ER-DataTypes    :     41
      USED in MAP-Errors           :     79    129
      USED in MAP-ER-DataTypes     :     14

roamingNumber.................identifier of ISDN-AddressString
    DEFINED in MAP-CallHandlingOperat :     76

roamingNumber.................identifier of [0] ISDN-AddressString
    DEFINED in MAP-CommonDataTypes :     204

roamingNumber.................identifier of ISDN-AddressString
    DEFINED in MAP-CH-DataTypes    :     59

roamingNumber.................identifier of [3] ISDN-AddressString
    DEFINED in MAP-SM-DataTypes    :     73

roamingRestrictionDueToUnsupportedFeatur identifier of [9] NULL
    DEFINED in MAP-MS-DataTypes    :     243

roamingRestrictionDueToUnsupportedFeatur identifier of [4] NULL
    DEFINED in MAP-MS-DataTypes    :     301
```

```
routingInfo...................identifier of RoutingInfo
    DEFINED in MAP-CH-DataTypes      :     53

RoutingInfo...................type reference CHOICE
    DEFINED in MAP-CH-DataTypes      :     58
        USED in MAP-CH-DataTypes      :     53

routingInfoForSM-Arg..........identifier of RoutingInfoForSM-Arg
    DEFINED in MAP-ShortMessageServic :     67

RoutingInfoForSM-Arg..........type reference SEQUENCE
    DEFINED in MAP-SM-DataTypes      :     41
        USED in MAP-ShortMessageServic :     46     67
        USED in MAP-SM-DataTypes      :     14

routingInfoForSM-Res..........identifier of RoutingInfoForSM-Res
    DEFINED in MAP-ShortMessageServic :     69

RoutingInfoForSM-Res..........type reference SEQUENCE
    DEFINED in MAP-SM-DataTypes      :     49
        USED in MAP-ShortMessageServic :     47     69
        USED in MAP-SM-DataTypes      :     15

rss...........................identifier of Named Number, 7
    DEFINED in MAP-CommonDataTypes   :     237

sc-AddressNotIncluded.........identifier of Named Number, 0
    DEFINED in MAP-SM-DataTypes      :     109

sc-Congestion.................identifier of Named Number, 4
    DEFINED in MAP-ER-DataTypes      :     83

sendAuthenticationInfo...value reference SendAuthenticationInfo, CHOICE VALUE
    DEFINED in MAP-Protocol          :     156

SendAuthenticationInfo........type reference OPERATION
    DEFINED in MAP-MobileServiceOpera :     212
        USED in MAP-Protocol          :     23     156
        USED in MAP-MobileServiceOpera :     30

sendAuthenticationInfoArg.....identifier of SendAuthenticationInfoArg
    DEFINED in MAP-MobileServiceOpera :     214

SendAuthenticationInfoArg.....type reference IMSI
    DEFINED in MAP-MS-DataTypes      :     185
        USED in MAP-MobileServiceOpera :     79     214
        USED in MAP-MS-DataTypes      :     31

sendAuthenticationInfoRes.....identifier of SendAuthenticationInfoRes
    DEFINED in MAP-MobileServiceOpera :     216

SendAuthenticationInfoRes.....type reference AuthenticationSetList
    DEFINED in MAP-MS-DataTypes      :     187
        USED in MAP-MobileServiceOpera :     80     216
        USED in MAP-MS-DataTypes      :     32

sendEndSignal.................value reference SendEndSignal, CHOICE VALUE
    DEFINED in MAP-Protocol          :     143

SendEndSignal.................type reference OPERATION
    DEFINED in MAP-MobileServiceOpera :     171
        USED in MAP-Protocol          :     18     143
        USED in MAP-MobileServiceOpera :     23

sendIdentification...........value reference SendIdentification, CHOICE VALUE
    DEFINED in MAP-Protocol          :     133

SendIdentification............type reference OPERATION
    DEFINED in MAP-MobileServiceOpera :     136
        USED in MAP-Protocol          :     15     133
        USED in MAP-MobileServiceOpera :     18

sendIdentificationRes.........identifier of SendIdentificationRes
    DEFINED in MAP-MobileServiceOpera :     140

SendIdentificationRes.........type reference SEQUENCE
    DEFINED in MAP-MS-DataTypes      :     111
        USED in MAP-MobileServiceOpera :     72     140
        USED in MAP-MS-DataTypes      :     20
```

```
sendIMSI......................value reference SendIMSI, CHOICE VALUE
    DEFINED in MAP-Protocol         :     190


SendIMSI......................type reference OPERATION
    DEFINED in MAP-OperationAndMainte :   83
       USED in MAP-Protocol         :     40    190
       USED in MAP-OperationAndMainte :   17


sendParameters................value reference SendParameters, CHOICE VALUE
    DEFINED in MAP-Protocol         :     167


SendParameters................type reference OPERATION
    DEFINED in MAP-MobileServiceOpera :   241
       USED in MAP-Protocol         :     25    167
       USED in MAP-MobileServiceOpera :   36


sendParametersArg.............identifier of SendParametersArg
    DEFINED in MAP-MobileServiceOpera :   243


SendParametersArg.............type reference SEQUENCE
    DEFINED in MAP-MS-DataTypes      :     202
       USED in MAP-MobileServiceOpera :   82    243
       USED in MAP-MS-DataTypes      :     38


sendRoutingInfo...............value reference SendRoutingInfo, CHOICE VALUE
    DEFINED in MAP-Protocol         :     197


SendRoutingInfo...............type reference OPERATION
    DEFINED in MAP-CallHandlingOperat :   51
       USED in MAP-Protocol         :     46    197
       USED in MAP-CallHandlingOperat :   13


sendRoutingInfoArg............identifier of SendRoutingInfoArg
    DEFINED in MAP-CallHandlingOperat :   53


SendRoutingInfoArg............type reference SEQUENCE
    DEFINED in MAP-CH-DataTypes      :     43
       USED in MAP-CallHandlingOperat :   38    53
       USED in MAP-CH-DataTypes      :     14


sendRoutingInfoForSM.......value reference SendRoutingInfoForSM, CHOICE VALUE
    DEFINED in MAP-Protocol         :     226


SendRoutingInfoForSM..........type reference OPERATION
    DEFINED in MAP-ShortMessageServic :   65
       USED in MAP-Protocol         :     70    226
       USED in MAP-ShortMessageServic :   13


sendRoutingInfoRes............identifier of SendRoutingInfoRes
    DEFINED in MAP-CallHandlingOperat :   55


SendRoutingInfoRes............type reference SEQUENCE
    DEFINED in MAP-CH-DataTypes      :     51
       USED in MAP-CallHandlingOperat :   39    55
       USED in MAP-CH-DataTypes      :     15


SentParameter.................type reference CHOICE
    DEFINED in MAP-MS-DataTypes      :     219
       USED in MAP-MS-DataTypes      :     216


sentParameterList.............identifier of SentParameterList
    DEFINED in MAP-MobileServiceOpera :   245


SentParameterList.............type reference SEQUENCE OF
    DEFINED in MAP-MS-DataTypes      :     215
       USED in MAP-MobileServiceOpera :   83    245
       USED in MAP-MS-DataTypes      :     39


serviceCentreAddress..........identifier of [2] AddressString
    DEFINED in MAP-SM-DataTypes      :     44


serviceCentreAddress..........identifier of AddressString
    DEFINED in MAP-SM-DataTypes      :     87


serviceCentreAddress..........identifier of AddressString
    DEFINED in MAP-SM-DataTypes      :     100


serviceCentreAddressDA........identifier of [4] AddressString
    DEFINED in MAP-SM-DataTypes      :     75


serviceCentreAddressOA........identifier of [4] AddressString
```

```
      DEFINED in MAP-SM-DataTypes       :      81

serviceGranted................identifier of Named Number, 0
   DEFINED in MAP-MS-DataTypes       :     251

servingCellId.................identifier of GlobalCellId
   DEFINED in MAP-MS-DataTypes       :     141

servingCellId.................identifier of GlobalCellId
   DEFINED in MAP-MS-DataTypes       :     179

shortMessageMO-PP.............value reference TeleserviceCode, '00100010'B
   DEFINED in MAP-TS-Code            :      28

shortMessageMT-PP.............value reference TeleserviceCode, '00100001'B
   DEFINED in MAP-TS-Code            :      27

signalInfo....................identifier of SignalInfo
   DEFINED in MAP-CommonDataTypes    :     160

SignalInfo....................type reference OCTET STRING
   DEFINED in MAP-CommonDataTypes    :     165
      USED in MAP-CommonDataTypes    :      20    160
      USED in MAP-SM-DataTypes       :      26     65
      USED in MAP-ER-DataTypes       :      28     89

sm-DeliveryFailure............value reference SM-DeliveryFailure, CHOICE VALUE
   DEFINED in MAP-Protocol           :     318

SM-DeliveryFailure............type reference ERROR
   DEFINED in MAP-Errors             :     221
      USED in MAP-Protocol           :     119    318
      USED in MAP-ShortMessageServic :      40     97
      USED in MAP-Errors             :      64

sm-DeliveryFailureCause.......identifier of SM-DeliveryFailureCause
   DEFINED in MAP-Errors             :     223

SM-DeliveryFailureCause.......type reference CHOICE
   DEFINED in MAP-ER-DataTypes       :      68
      USED in MAP-Errors             :      84    223
      USED in MAP-ER-DataTypes       :      19

sm-DeliveryFailureCauseWithDiagnostic...identifier of SM-DeliveryFailureCauseWithDiagnostic
   DEFINED in MAP-ER-DataTypes       :      69

SM-DeliveryFailureCauseWithDiagnostic...type reference SEQUENCE
   DEFINED in MAP-ER-DataTypes       :      87
      USED in MAP-ER-DataTypes       :      69

sm-DeliveryOutcome............identifier of SM-DeliveryOutcome
   DEFINED in MAP-SM-DataTypes       :      88

SM-DeliveryOutcome............type reference ENUMERATED
   DEFINED in MAP-SM-DataTypes       :      93
      USED in MAP-SM-DataTypes       :      88

sm-EnumeratedDeliveryFailureCause......identifier of SM-EnumeratedDeliveryFailureCause
   DEFINED in MAP-ER-DataTypes       :      73

SM-EnumeratedDeliveryFailureCause.......type reference ENUMERATED
   DEFINED in MAP-ER-DataTypes       :      78
      USED in MAP-ER-DataTypes       :      73     88

sm-EnumeratedDeliveryFailureCause......identifier of SM-EnumeratedDeliveryFailureCause
   DEFINED in MAP-ER-DataTypes       :      88

sm-RP-DA......................identifier of SM-RP-DA
   DEFINED in MAP-SM-DataTypes       :      63

SM-RP-DA......................type reference CHOICE
   DEFINED in MAP-SM-DataTypes       :      70
      USED in MAP-SM-DataTypes       :      63

sm-RP-OA......................identifier of SM-RP-OA
   DEFINED in MAP-SM-DataTypes       :      64

SM-RP-OA......................type reference CHOICE
   DEFINED in MAP-SM-DataTypes       :      79
      USED in MAP-SM-DataTypes       :      64
```

```
sm-RP-PRI....................identifier of [1] BOOLEAN
     DEFINED in MAP-SM-DataTypes        :      43

sm-RP-UI.....................identifier of SignalInfo
     DEFINED in MAP-SM-DataTypes        :      65

sres.........................identifier of SRES
     DEFINED in MAP-MS-DataTypes        :     121

SRES.........................type reference OCTET STRING
     DEFINED in MAP-MS-DataTypes        :     127
        USED in MAP-MS-DataTypes        :     121

ss-AccessBarred..............identifier of Named Number, 5
     DEFINED in MAP-MS-DataTypes        :     265

ss-Code......................identifier of SS-Code
     DEFINED in MAP-SupplementaryServi :     257

ss-Code......................identifier of SS-Code
     DEFINED in MAP-SS-DataTypes        :      53

ss-Code......................identifier of SS-Code
     DEFINED in MAP-SS-DataTypes        :      71

ss-Code......................identifier of SS-Code
     DEFINED in MAP-SS-DataTypes        :     137

ss-Code......................identifier of SS-Code
     DEFINED in MAP-SS-DataTypes        :     199

ss-Code......................identifier of SS-Code
     DEFINED in MAP-SS-DataTypes        :     222

SS-Code......................type reference OCTET STRING
     DEFINED in MAP-SS-Code             :      11
        USED in MAP-SupplementaryServi :      71    257
        USED in MAP-SS-DataTypes        :      45     53     71    137    199    222    285
        USED in MAP-SS-Code             :      21     25     28     30     32     34     36     40     42
                                               44     46     48     50     53     56     59     63     66
                                               68     70     74     77     80     83     86     89     91
                                               94     98    102    104    106    108    110    113    115
                                              117    121    123    124    125    126    127    128    129
                                              130    131    132    133    134    135    136    137
        USED in MAP-ER-DataTypes        :      34     58

ss-Code......................identifier of [1] SS-Code
     DEFINED in MAP-ER-DataTypes        :      58

ss-Data......................identifier of [3] SS-Data
     DEFINED in MAP-SS-DataTypes        :      68

SS-Data......................type reference SEQUENCE
     DEFINED in MAP-SS-DataTypes        :     198
        USED in MAP-SS-DataTypes        :      68

ss-ErrorStatus...............value reference SS-ErrorStatus, CHOICE VALUE
     DEFINED in MAP-Protocol            :     299

SS-ErrorStatus...............type reference ERROR
     DEFINED in MAP-Errors              :     186
        USED in MAP-Protocol            :     109    299
        USED in MAP-SupplementaryServi :      41     97    123    149    176
        USED in MAP-Errors              :      52

ss-ForBS.....................identifier of SS-ForBS-Code
     DEFINED in MAP-SupplementaryServi :     105

ss-ForBS.....................identifier of SS-ForBS-Code
     DEFINED in MAP-SupplementaryServi :     131

ss-ForBS.....................identifier of SS-ForBS-Code
     DEFINED in MAP-SupplementaryServi :     158

ss-ForBS.....................identifier of SS-ForBS-Code
     DEFINED in MAP-SupplementaryServi :     184

SS-ForBS-Code................type reference SEQUENCE
     DEFINED in MAP-SS-DataTypes        :     221
        USED in MAP-SupplementaryServi :      59    105    131    158    184
        USED in MAP-SS-DataTypes        :      20
```

```
ss-Incompatibility...........value reference SS-Incompatibility, CHOICE VALUE
    DEFINED in MAP-Protocol        :    302

SS-Incompatibility...........type reference ERROR
    DEFINED in MAP-Errors          :    199
        USED in MAP-Protocol       :    112    302
        USED in MAP-SupplementaryServi :    44    101    151
        USED in MAP-Errors         :     55

ss-IncompatibilityCause.......identifier of SS-IncompatibilityCause
    DEFINED in MAP-Errors          :    201

SS-IncompatibilityCause.......type reference SEQUENCE
    DEFINED in MAP-ER-DataTypes    :     57
        USED in MAP-Errors         :     82    201
        USED in MAP-ER-DataTypes   :     17

ss-Info......................identifier of SS-Info
    DEFINED in MAP-SupplementaryServi :    84

ss-Info......................identifier of SS-Info
    DEFINED in MAP-SupplementaryServi :   107

ss-Info......................identifier of SS-Info
    DEFINED in MAP-SupplementaryServi :   133

ss-Info......................identifier of SS-Info
    DEFINED in MAP-SupplementaryServi :   160

SS-Info......................type reference CHOICE
    DEFINED in MAP-SS-DataTypes    :     63
        USED in MAP-SupplementaryServi :    58    84    107    133    160
        USED in MAP-SS-DataTypes   :     15    290

SS-InfoList..................type reference SEQUENCE OF
    DEFINED in MAP-SS-DataTypes    :    289
        USED in MAP-MS-DataTypes   :     53    240
        USED in MAP-SS-DataTypes   :     29

ss-List......................identifier of [3] SS-List
    DEFINED in MAP-MS-DataTypes    :    284

ss-List......................identifier of [2] SS-List
    DEFINED in MAP-MS-DataTypes    :    300

SS-List......................type reference SEQUENCE OF
    DEFINED in MAP-SS-DataTypes    :    284
        USED in MAP-MS-DataTypes   :     52    284    300
        USED in MAP-SS-DataTypes   :     28

ss-NotAvailable..............value reference SS-NotAvailable, CHOICE VALUE
    DEFINED in MAP-Protocol        :    300

SS-NotAvailable..............type reference ERROR
    DEFINED in MAP-Errors          :    191
        USED in MAP-Protocol       :    110    300
        USED in MAP-SupplementaryServi :    42    201
        USED in MAP-Errors         :     53

ss-Status....................identifier of SS-Status
    DEFINED in MAP-Errors          :    188

ss-Status....................identifier of [4] SS-Status
    DEFINED in MAP-SS-DataTypes    :     81

SS-Status....................type reference OCTET STRING
    DEFINED in MAP-SS-DataTypes    :     90
        USED in MAP-Errors         :     73    188
        USED in MAP-SS-DataTypes   :     16    81    148    201    227    232
        USED in MAP-ER-DataTypes   :     23    60

ss-Status....................identifier of [4] SS-Status
    DEFINED in MAP-SS-DataTypes    :    148

ss-Status....................identifier of [4] SS-Status
    DEFINED in MAP-SS-DataTypes    :    201

ss-Status....................identifier of SS-Status
    DEFINED in MAP-SS-DataTypes    :    227
```

```
ss-Status...................identifier of [0] SS-Status
    DEFINED in MAP-SS-DataTypes    :    232


ss-Status...................identifier of [4] SS-Status
    DEFINED in MAP-ER-DataTypes    :     60


ss-SubscriptionOption.........identifier of SS-SubscriptionOption
    DEFINED in MAP-Errors          :    195


ss-SubscriptionOption.........identifier of SS-SubscriptionOption
    DEFINED in MAP-SS-DataTypes    :    202


SS-SubscriptionOption.........type reference CHOICE
    DEFINED in MAP-SS-DataTypes    :    208
       USED in MAP-Errors          :     74    195
       USED in MAP-SS-DataTypes    :     19    202


ss-SubscriptionViolation......value reference SS-SubscriptionViolation, CHOICE VALUE
    DEFINED in MAP-Protocol        :    301


SS-SubscriptionViolation......type reference ERROR
    DEFINED in MAP-Errors          :    193
       USED in MAP-Protocol        :    111    301
       USED in MAP-SupplementaryServi :   43    98    124    150    177    267
       USED in MAP-Errors          :     54


ss-UserData...................identifier of SS-UserData
    DEFINED in MAP-SupplementaryServi :  205


ss-UserData...................identifier of SS-UserData
    DEFINED in MAP-SupplementaryServi :  207


SS-UserData...................type reference IA5String
    DEFINED in MAP-SS-DataTypes    :    240
       USED in MAP-SupplementaryServi :   61    205    207
       USED in MAP-SS-DataTypes    :     22


storedMSISDN..................identifier of ISDN-AddressString
    DEFINED in MAP-ShortMessageServic :  103


storedMSISDN..................identifier of ISDN-AddressString
    DEFINED in MAP-SM-DataTypes    :    104


subscriberBusyForMT-SMS.value reference SubscriberBusyForMT-SMS, CHOICE VALUE
    DEFINED in MAP-Protocol        :    316


SubscriberBusyForMT-SMS.......type reference ERROR
    DEFINED in MAP-Errors          :    219
       USED in MAP-Protocol        :    118    316
       USED in MAP-ShortMessageServic :   39     95
       USED in MAP-Errors          :     63


subscriberData................identifier of [2] SubscriberData
    DEFINED in MAP-MS-DataTypes    :    222


SubscriberData................type reference SEQUENCE
    DEFINED in MAP-MS-DataTypes    :    234
       USED in MAP-MS-DataTypes    :    222    231


subscriberId..................identifier of SubscriberId
    DEFINED in MAP-MS-DataTypes    :    203


SubscriberId..................type reference CHOICE
    DEFINED in MAP-CommonDataTypes :    191
       USED in MAP-MS-DataTypes    :     61    203
       USED in MAP-CommonDataTypes :     26


subscriberNotMemberOfCUG......identifier of Named Number, 1
    DEFINED in MAP-ER-DataTypes    :     51


subscriberNotSC-Subscriber....identifier of Named Number, 6
    DEFINED in MAP-ER-DataTypes    :     85


subscriberStatus..............identifier of [3] SubscriberStatus
    DEFINED in MAP-MS-DataTypes    :    237


SubscriberStatus..............type reference ENUMERATED
    DEFINED in MAP-MS-DataTypes    :    250
       USED in MAP-MS-DataTypes    :    237


subsequentHandoverFailure.....value reference SubsequentHandoverFailure, CHOICE VALUE
```

```
        DEFINED in MAP-Protocol         :     278


SubsequentHandoverFailure.....type reference ERROR
        DEFINED in MAP-Errors           :     151
           USED in MAP-Protocol         :     101    278
           USED in MAP-MobileServiceOpera :    63    196    208
           USED in MAP-Errors           :      38


successfulTransfer............identifier of Named Number, 2
        DEFINED in MAP-SM-DataTypes      :      96


systemFailure.................value reference SystemFailure, CHOICE VALUE
        DEFINED in MAP-Protocol         :     245


SystemFailure.................type reference ERROR
        DEFINED in MAP-Errors           :      98
           USED in MAP-Protocol         :      83    245
           USED in MAP-MobileServiceOpera :    51    112    153    164    219    232
                                                292
           USED in MAP-OperationAndMainte :    25     57     69
           USED in MAP-CallHandlingOperat :    22     57     78
           USED in MAP-SupplementaryServi :    33     87    110    136    163    188
                                               210    219    232    246    261
           USED in MAP-ShortMessageServic :    28     71     85    127
           USED in MAP-Errors           :      14


targetCellId..................identifier of GlobalCellId
        DEFINED in MAP-MS-DataTypes      :     134


targetCellId..................identifier of GlobalCellId
        DEFINED in MAP-MS-DataTypes      :     140


targetCellId..................identifier of GlobalCellId
        DEFINED in MAP-MS-DataTypes      :     172


targetCellId..................identifier of GlobalCellId
        DEFINED in MAP-MS-DataTypes      :     178


targetCellId..................identifier of [1] GlobalCellId
        DEFINED in MAP-OM-DataTypes      :      61


targetMSC-Number..............identifier of ISDN-AddressString
        DEFINED in MAP-MS-DataTypes      :     173


targetMSC-Number..............identifier of ISDN-AddressString
        DEFINED in MAP-MS-DataTypes      :     180


TBCD-STRING...................type reference OCTET STRING
        DEFINED in MAP-CommonDataTypes   :      59
           USED in MAP-CommonDataTypes   :     186    195


TCAPMessages..................module reference
        DEFINED in TCAPMessages         :       1
           USED in MAP-MobileServiceOpera :    48
           USED in MAP-OperationAndMainte :    22
           USED in MAP-CallHandlingOperat :    19
           USED in MAP-SupplementaryServi :    30
           USED in MAP-ShortMessageServic :    25
           USED in MAP-Errors           :      70


telephony.....................value reference TeleserviceCode, '00010001'B
        DEFINED in MAP-TS-Code          :      23


teleservice...................identifier of [3] TeleserviceCode
        DEFINED in MAP-CommonDataTypes   :     257


teleservice...................identifier of [5] TeleserviceCode
        DEFINED in MAP-SM-DataTypes      :      45


TeleserviceCode...............type reference OCTET STRING
        DEFINED in MAP-TS-Code          :      11
           USED in MAP-CommonDataTypes   :      45    251    257
           USED in MAP-SM-DataTypes      :      34     45
           USED in MAP-TS-Code          :      20     22     23     24     26     27
                                                28     30     31     32     33     38
                                                41     51     53     54     55     56
                                                57     58     59     60     61     62
                                                63     64     65     66     67


teleserviceList...............identifier of [6] TeleserviceList
        DEFINED in MAP-MS-DataTypes      :     239
```

```
teleserviceList...............identifier of [1] TeleserviceList
   DEFINED in MAP-MS-DataTypes      :    282


TeleserviceList...............type reference SEQUENCE OF
   DEFINED in MAP-CommonDataTypes   :    250
      USED in MAP-MS-DataTypes      :    70   239   282
      USED in MAP-CommonDataTypes   :    37


teleserviceNotProvisioned.....value reference TeleserviceNotProvisioned, CHOICE VALUE
   DEFINED in MAP-Protocol          :    268


TeleserviceNotProvisioned.....type reference ERROR
   DEFINED in MAP-Errors            :    140
      USED in MAP-Protocol          :    97   268
      USED in MAP-CallHandlingOperat :   29    65
      USED in MAP-SupplementaryServi :   38    93   118   144   171   196
      USED in MAP-ShortMessageServic :   36    76
      USED in MAP-Errors            :    32


temporaryDefaultAllowed.......identifier of Named Number, 2
   DEFINED in MAP-SS-DataTypes      :    215


temporaryDefaultRestricted....identifier of Named Number, 1
   DEFINED in MAP-SS-DataTypes      :    214


tmsi..........................identifier of TMSI
   DEFINED in MAP-MobileServiceOpera :   138


TMSI..........................type reference OCTET STRING
   DEFINED in MAP-CommonDataTypes   :    189
      USED in MAP-MobileServiceOpera :   96   138
      USED in MAP-CommonDataTypes   :    25   193


tmsi..........................identifier of [1] TMSI
   DEFINED in MAP-CommonDataTypes   :    193


tooManyZoneCodes..............identifier of Named Number, 1
   DEFINED in MAP-MS-DataTypes      :    293


traceReference................identifier of [1] TraceReference
   DEFINED in MAP-OM-DataTypes      :    34


TraceReference................type reference OCTET STRING
   DEFINED in MAP-OM-DataTypes      :    39
      USED in MAP-OM-DataTypes      :    34    47    52


traceReference................identifier of [1] TraceReference
   DEFINED in MAP-OM-DataTypes      :    47


traceReference................identifier of [1] TraceReference
   DEFINED in MAP-OM-DataTypes      :    52


traceSubscriberActivity.value reference TraceSubscriberActivity, CHOICE VALUE
   DEFINED in MAP-Protocol          :    186


TraceSubscriberActivity.......type reference OPERATION
   DEFINED in MAP-OperationAndMainte :   75
      USED in MAP-Protocol          :    38   186
      USED in MAP-OperationAndMainte :   15


traceSubscriberActivityArg....identifier of TraceSubscriberActivityArg
   DEFINED in MAP-OperationAndMainte :   77


TraceSubscriberActivityArg....type reference SEQUENCE
   DEFINED in MAP-OM-DataTypes      :    50
      USED in MAP-OperationAndMainte :   38    77
      USED in MAP-OM-DataTypes      :    16


traceType.....................identifier of [2] TraceType
   DEFINED in MAP-OM-DataTypes      :    35


TraceType.....................type reference INTEGER
   DEFINED in MAP-OM-DataTypes      :    41
      USED in MAP-OM-DataTypes      :    35    53


traceType.....................identifier of [2] TraceType
   DEFINED in MAP-OM-DataTypes      :    53


tracingBufferFull.............value reference TracingBufferFull, CHOICE VALUE
   DEFINED in MAP-Protocol          :    284
```

```
TracingBufferFull.............type reference ERROR
    DEFINED in MAP-Errors          :     156
        USED in MAP-Protocol       :     102    284
        USED in MAP-OperationAndMainte :  31     62
        USED in MAP-Errors         :      41

TransactionID................type reference OCTET STRING
    DEFINED in TCAPMessages        :     100
        USED in TCAPMessages       :      47     97     98

undetermined.................identifier of Named Number, 0
    DEFINED in MAP-ER-DataTypes    :      64

unexpectedDataValue..........value reference UnexpectedDataValue, CHOICE VALUE
    DEFINED in MAP-Protocol        :     247

UnexpectedDataValue..........type reference ERROR
    DEFINED in MAP-Errors          :     105
        USED in MAP-Protocol       :      85    247
        USED in MAP-MobileServiceOpera :  53    115    126    155    165    193
                                         204    221    235    250    263    275
                                         294
        USED in MAP-OperationAndMainte :  27     59     71     90
        USED in MAP-CallHandlingOperat :  24     59     80
        USED in MAP-SupplementaryServi :  35     89    113    139    166    191
                                         211    221    234    248    264
        USED in MAP-ShortMessageServic :  30     73     88    110    129    141
        USED in MAP-Errors         :      16

unexpectedError..............identifier of Named Number, 3
    DEFINED in TCAPMessages        :     199

unexpectedLinkedOperation.....identifier of Named Number, 7
    DEFINED in TCAPMessages        :     190

unidentifiedSubscriber........value reference UnidentifiedSubscriber, CHOICE VALUE
    DEFINED in MAP-Protocol        :     257

UnidentifiedSubscriber........type reference ERROR
    DEFINED in MAP-Errors          :     120
        USED in MAP-Protocol       :      91    257
        USED in MAP-MobileServiceOpera :  57    127    143    252    264    276
        USED in MAP-OperationAndMainte :  30     61     73
        USED in MAP-ShortMessageServic :  33     90
        USED in MAP-Errors         :      24

unidirectional......identifier of [APPLICATION 1] IMPLICIT Unidirectional
    DEFINED in TCAPMessages        :      52

Unidirectional...............type reference SEQUENCE
    DEFINED in TCAPMessages        :      58
        USED in TCAPMessages       :      52

unknownAlphabet..............value reference UnknownAlphabet, CHOICE VALUE
    DEFINED in MAP-Protocol        :     303

UnknownAlphabet..............type reference ERROR
    DEFINED in MAP-Errors          :     204
        USED in MAP-Protocol       :     113    303
        USED in MAP-SupplementaryServi :  48    222    238    252
        USED in MAP-Errors         :      56

unknownBaseStation...........value reference UnknownBaseStation, CHOICE VALUE
    DEFINED in MAP-Protocol        :     255

UnknownBaseStation...........type reference ERROR
    DEFINED in MAP-Errors          :     116
        USED in MAP-Protocol       :      89    255
        USED in MAP-MobileServiceOpera :  55    166    205
        USED in MAP-Errors         :      22

unknownEquipment.............value reference UnknownEquipment, CHOICE VALUE
    DEFINED in MAP-Protocol        :     258

UnknownEquipment.............type reference ERROR
    DEFINED in MAP-Errors          :     122
        USED in MAP-Protocol       :      92    258
        USED in MAP-MobileServiceOpera :  58    237
        USED in MAP-Errors         :      25
```

```
unknownMSC....................value reference UnknownMSC, CHOICE VALUE
    DEFINED in MAP-Protocol        :      256

UnknownMSC....................type reference ERROR
    DEFINED in MAP-Errors          :      118
        USED in MAP-Protocol       :       90     256
        USED in MAP-MobileServiceOpera :   56    195    206
        USED in MAP-Errors         :       23

unknownServiceCentre..........identifier of Named Number, 3
    DEFINED in MAP-ER-DataTypes    :       82

unknownSubscriber.............value reference UnknownSubscriber, CHOICE VALUE
    DEFINED in MAP-Protocol        :      253

UnknownSubscriber.............type reference ERROR
    DEFINED in MAP-Errors          :      112
        USED in MAP-Protocol       :       87    253
        USED in MAP-MobileServiceOpera :   54    116    222    251    295
        USED in MAP-OperationAndMainte :   29     91
        USED in MAP-CallHandlingOperat :   26     61
        USED in MAP-SupplementaryServi :   36     90    114    140    167    192
        USED in MAP-ShortMessageServic :   32     75    111    143
        USED in MAP-Errors         :       20

unrecognizedComponent.........identifier of Named Number, 0
    DEFINED in TCAPMessages        :      179

unrecognizedError.............identifier of Named Number, 2
    DEFINED in TCAPMessages        :      198

unrecognizedInvokeID..........identifier of Named Number, 0
    DEFINED in TCAPMessages        :      192

unrecognizedInvokeID..........identifier of Named Number, 0
    DEFINED in TCAPMessages        :      196

unrecognizedLinkedID..........identifier of Named Number, 5
    DEFINED in TCAPMessages        :      188

unrecognizedMessageType.......identifier of Named Number, 0
    DEFINED in TCAPMessages        :      103

unrecognizedOperation.........identifier of Named Number, 1
    DEFINED in TCAPMessages        :      184

unrecognizedTransactionID.....identifier of Named Number, 1
    DEFINED in TCAPMessages        :      104

unstructuredSS-Notify.....value reference UnstructuredSS-Notify, CHOICE VALUE
    DEFINED in MAP-Protocol        :      216

UnstructuredSS-Notify.........type reference OPERATION
    DEFINED in MAP-SupplementaryServi :   241
        USED in MAP-Protocol       :       61    216
        USED in MAP-SupplementaryServi :   22

unstructuredSS-Request...value reference UnstructuredSS-Request, CHOICE VALUE
    DEFINED in MAP-Protocol        :      214

UnstructuredSS-Request........type reference OPERATION
    DEFINED in MAP-SupplementaryServi :   225
        USED in MAP-Protocol       :       60    214
        USED in MAP-SupplementaryServi :   21

updateLocation................value reference UpdateLocation, CHOICE VALUE
    DEFINED in MAP-Protocol        :      129

UpdateLocation................type reference OPERATION
    DEFINED in MAP-MobileServiceOpera :   106
        USED in MAP-Protocol       :       12    129
        USED in MAP-MobileServiceOpera :   15

updateLocationArg.............identifier of UpdateLocationArg
    DEFINED in MAP-MobileServiceOpera :   108

UpdateLocationArg.............type reference SEQUENCE
    DEFINED in MAP-MS-DataTypes    :       80
        USED in MAP-MobileServiceOpera :   68    108
        USED in MAP-MS-DataTypes   :       16
```

```
updateLocationRes..............identifier of UpdateLocationRes
    DEFINED in MAP-MobileServiceOpera :    110


UpdateLocationRes..............type reference CHOICE
    DEFINED in MAP-MS-DataTypes      :      87
       USED in MAP-MobileServiceOpera :    69    110
       USED in MAP-MS-DataTypes      :      17


ussd-Arg......................identifier of USSD-Arg
    DEFINED in MAP-SupplementaryServi :    215


ussd-Arg......................identifier of USSD-Arg
    DEFINED in MAP-SupplementaryServi :    227


ussd-Arg......................identifier of USSD-Arg
    DEFINED in MAP-SupplementaryServi :    243


USSD-Arg......................type reference SEQUENCE
    DEFINED in MAP-SS-DataTypes      :      242
       USED in MAP-SupplementaryServi :    62    215    227    243
       USED in MAP-SS-DataTypes      :      23


ussd-Busy....................value reference USSD-Busy, CHOICE VALUE
    DEFINED in MAP-Protocol          :      305


USSD-Busy....................type reference ERROR
    DEFINED in MAP-Errors            :      206
       USED in MAP-Protocol          :      114    305
       USED in MAP-SupplementaryServi :    49    239    253
       USED in MAP-Errors            :      57


ussd-DataCodingScheme.........identifier of USSD-DataCodingScheme
    DEFINED in MAP-SS-DataTypes      :      243


ussd-DataCodingScheme.........identifier of USSD-DataCodingScheme
    DEFINED in MAP-SS-DataTypes      :      248


USSD-DataCodingScheme.........type reference OCTET STRING
    DEFINED in MAP-SS-DataTypes      :      252
       USED in MAP-SS-DataTypes      :      243    248


ussd-Res......................identifier of USSD-Res
    DEFINED in MAP-SupplementaryServi :    217


ussd-Res......................identifier of USSD-Res
    DEFINED in MAP-SupplementaryServi :    229


USSD-Res......................type reference SEQUENCE
    DEFINED in MAP-SS-DataTypes      :      247
       USED in MAP-SupplementaryServi :    63    217    229
       USED in MAP-SS-DataTypes      :      24


ussd-String...................identifier of USSD-String
    DEFINED in MAP-SS-DataTypes      :      244


ussd-String...................identifier of USSD-String
    DEFINED in MAP-SS-DataTypes      :      249


USSD-String...................type reference OCTET STRING
    DEFINED in MAP-SS-DataTypes      :      257
       USED in MAP-SS-DataTypes      :      244    249


uus...........................value reference SS-Code, '10000001'B
    DEFINED in MAP-SS-Code           :      98


vlr...........................identifier of Named Number, 2
    DEFINED in MAP-CommonDataTypes   :      232


vlr-Number....................identifier of ISDN-AddressString
    DEFINED in MAP-MS-DataTypes      :      83


vlr-Number....................identifier of ISDN-AddressString
    DEFINED in MAP-MS-DataTypes      :      103


vmsc..........................identifier of Named Number, 5
    DEFINED in MAP-CommonDataTypes   :      235


whiteListed...................identifier of Named Number, 0
    DEFINED in MAP-MS-DataTypes      :      195


ZoneCode......................type reference OCTET STRING
```

```
    DEFINED in MAP-MS-DataTypes       :    276
       USED in MAP-MS-DataTypes       :    274   304


ZoneCodeList..................type reference SEQUENCE OF
    DEFINED in MAP-MS-DataTypes       :    273
       USED in MAP-MS-DataTypes       :    246


zoneCodesConflict.............identifier of Named Number, 2
    DEFINED in MAP-MS-DataTypes       :    294
```

Annex B (informative): Fully expanded ASN.1 sources for abstract syntaxes of MAP

Annex B is not part of the standard, it is included for information purposes only.

For every (Value)Assignment in the root ASN.1 module all the used defined types and defined values, which are defined within the ASN.1 module or imported from ASN.1 modules, are replaced by the constructs this type or value is composed of.

The fully expanded ASN.1 root module is itself a correct and equivalent representation of the MAP-Protocol.

It allows to see at all the parameters, including all nested ones for a specific operationcode or errorcode at once.

Note that for those operations which use a result without parameters the keyword RESULT is not shown. Empty results are only defined in the ASN.1 description in clause 14.

B.1 Fully Expanded ASN.1 Source of MAP-Protocol/TCAPMessages

```
--          Expanded ASN.1 Module 'MAP-Protocol'
--SIEMENS ASN.1 Compiler      P2.10  (94-10-21 10:03:57)
--          Date: 94-10-21 Time: 10:49:06

MAP-Protocol { 0 identified-organization (4) etsi (0) mobileDomain (0) gsm-Network (1)
modules (3)
map-Protocol (4) version2 (2) }

DEFINITIONS

::=

BEGIN

updateLocation OPERATION
   ARGUMENT
      updateLocationArg SEQUENCE {
         imsi        OCTET STRING (SIZE (3..8)),
         locationInfo CHOICE {
            roamingNumber [0] IMPLICIT OCTET STRING (SIZE (1..9)),
            msc-Number [1] IMPLICIT OCTET STRING (SIZE (1..9))},
         vlr-Number OCTET STRING (SIZE (1..9)),
         lmsi        [10] IMPLICIT OCTET STRING (SIZE (4)) OPTIONAL,
         ... }
   RESULT
      updateLocationRes CHOICE {
         hlr-Number OCTET STRING (SIZE (1..9)),
         extensibleUpdateLocationRes SEQUENCE {
            hlr-Number OCTET STRING (SIZE (1..9)),
            ... }}
   ERRORS {
      -- systemFailure -- localValue 34,
      -- dataMissing -- localValue 35,
      -- unexpectedDataValue -- localValue 36,
      -- unknownSubscriber -- localValue 1,
      -- roamingNotAllowed -- localValue 8}
 ::= localValue 2

cancelLocation OPERATION
   ARGUMENT
      cancelLocationArg CHOICE {
         imsi        OCTET STRING (SIZE (3..8)),
         imsi-WithLMSI SEQUENCE {
            imsi        OCTET STRING (SIZE (3..8)),
            lmsi        OCTET STRING (SIZE (4)),
            ... }}
   ERRORS {
      -- dataMissing -- localValue 35,
      -- unexpectedDataValue -- localValue 36,
      -- unidentifiedSubscriber -- localValue 5}
 ::= localValue 3

purgeMS    OPERATION
```

```
    ARGUMENT
       purgeMS-Arg SEQUENCE {
           imsi       OCTET STRING (SIZE (3..8)),
           vlr-Number OCTET STRING (SIZE (1..9)),
           ... }
  ::= localValue 67

sendIdentification OPERATION
    ARGUMENT
       tmsi       OCTET STRING (SIZE (1..4))
    RESULT
       sendIdentificationRes SEQUENCE {
           imsi       OCTET STRING (SIZE (3..8)),
           authenticationSetList SEQUENCE SIZE (1..5) OF
              SEQUENCE {
                  rand       OCTET STRING (SIZE (16)),
                  sres       OCTET STRING (SIZE (4)),
                  kc         OCTET STRING (SIZE (8)),
                  ... } OPTIONAL,
           ... }
    ERRORS {
       -- dataMissing -- localValue 35,
       -- unidentifiedSubscriber -- localValue 5}
  ::= localValue 55

performHandover OPERATION
    ARGUMENT
       performHO-Arg SEQUENCE {
           targetCellId OCTET STRING (SIZE (5..7)),
           servingCellId OCTET STRING (SIZE (5..7)),
           channelType OCTET STRING (SIZE (1..10)),
           classmarkInfo OCTET STRING (SIZE (1..2)),
           handoverPriority [11] IMPLICIT OCTET STRING (SIZE (1)) OPTIONAL,
           kc          [12] IMPLICIT OCTET STRING (SIZE (8)) OPTIONAL}
    RESULT
       performHO-Res SEQUENCE {
           handoverNumber OCTET STRING (SIZE (1..9)),
           accessSignalInfo SEQUENCE {
              protocolId ENUMERATED {
                  gsm-0408    (1),
                  gsm-0806    (2),
                  gsm-BSSMAP  (3),
                  ets-300102-1  (4)},
              signalInfo OCTET STRING (SIZE (1..200)),
              ... }}
    ERRORS {
       -- systemFailure -- localValue 34,
       -- unexpectedDataValue -- localValue 36,
       -- unknownBaseStation -- localValue 2,
       -- invalidTargetBaseStation -- localValue 23,
       -- noRadioResourceAvailable -- localValue 24,
       -- noHandoverNumberAvailable -- localValue 25}
  ::= localValue 28

prepareHandover OPERATION
    ARGUMENT
       prepareHO-Arg SEQUENCE {
           targetCellId OCTET STRING (SIZE (5..7)) OPTIONAL,
           ho-NumberNotRequired NULL OPTIONAL,
           bss-APDU    SEQUENCE {
              protocolId ENUMERATED {
                  gsm-0408    (1),
                  gsm-0806    (2),
                  gsm-BSSMAP  (3),
                  ets-300102-1  (4)},
              signalInfo OCTET STRING (SIZE (1..200)),
              ... } OPTIONAL,
           ... }
    RESULT
       prepareHO-Res SEQUENCE {
           handoverNumber OCTET STRING (SIZE (1..9)) OPTIONAL,
           bss-APDU    SEQUENCE {
              protocolId ENUMERATED {
                  gsm-0408    (1),
                  gsm-0806    (2),
                  gsm-BSSMAP  (3),
                  ets-300102-1  (4)},
              signalInfo OCTET STRING (SIZE (1..200)),
              ... } OPTIONAL,
           ... }
    ERRORS {
```

```
        -- systemFailure -- localValue 34,
        -- dataMissing -- localValue 35,
        -- unexpectedDataValue -- localValue 36,
        -- noHandoverNumberAvailable -- localValue 25}
  ::= localValue 68

sendEndSignal OPERATION
   ARGUMENT
      bss-APDU   SEQUENCE {
         protocolId ENUMERATED {
            gsm-0408    (1),
            gsm-0806    (2),
            gsm-BSSMAP  (3),
            ets-300102-1  (4)},
         signalInfo OCTET STRING (SIZE (1..200)),
         ... }
  ::= localValue 29

processAccessSignalling OPERATION
   ARGUMENT
      bss-APDU   SEQUENCE {
         protocolId ENUMERATED {
            gsm-0408    (1),
            gsm-0806    (2),
            gsm-BSSMAP  (3),
            ets-300102-1  (4)},
         signalInfo OCTET STRING (SIZE (1..200)),
         ... }
  ::= localValue 33

forwardAccessSignalling OPERATION
   ARGUMENT
      bss-APDU   SEQUENCE {
         protocolId ENUMERATED {
            gsm-0408    (1),
            gsm-0806    (2),
            gsm-BSSMAP  (3),
            ets-300102-1  (4)},
         signalInfo OCTET STRING (SIZE (1..200)),
         ... }
  ::= localValue 34

performSubsequentHandover OPERATION
   ARGUMENT
      performSubsequentHO-Arg SEQUENCE {
         targetCellId OCTET STRING (SIZE (5..7)),
         servingCellId OCTET STRING (SIZE (5..7)),
         targetMSC-Number OCTET STRING (SIZE (1..9)),
         classmarkInfo [10] IMPLICIT OCTET STRING (SIZE (1..2)) OPTIONAL}
   RESULT
      accessSignalInfo SEQUENCE {
         protocolId ENUMERATED {
            gsm-0408    (1),
            gsm-0806    (2),
            gsm-BSSMAP  (3),
            ets-300102-1  (4)},
         signalInfo OCTET STRING (SIZE (1..200)),
         ... }
   ERRORS {
      -- unexpectedDataValue -- localValue 36,
      -- unknownBaseStation -- localValue 2,
      -- unknownMSC -- localValue 3,
      -- invalidTargetBaseStation -- localValue 23,
      -- subsequentHandoverFailure -- localValue 26}
  ::= localValue 30

prepareSubsequentHandover OPERATION
   ARGUMENT
      prepareSubsequentHO-Arg SEQUENCE {
         targetCellId OCTET STRING (SIZE (5..7)),
         targetMSC-Number OCTET STRING (SIZE (1..9)),
         bss-APDU   SEQUENCE {
            protocolId ENUMERATED {
               gsm-0408    (1),
               gsm-0806    (2),
               gsm-BSSMAP  (3),
               ets-300102-1  (4)},
            signalInfo OCTET STRING (SIZE (1..200)),
            ... },
         ... }
   RESULT
```

```
        bss-APDU    SEQUENCE {
            protocolId ENUMERATED {
                gsm-0408    (1),
                gsm-0806    (2),
                gsm-BSSMAP  (3),
                ets-300102-1  (4)},
            signalInfo OCTET STRING (SIZE (1..200)),
            ... }
    ERRORS {
        -- unexpectedDataValue -- localValue 36,
        -- dataMissing -- localValue 35,
        -- unknownMSC -- localValue 3,
        -- subsequentHandoverFailure -- localValue 26}
  ::= localValue 69

sendAuthenticationInfo OPERATION
    ARGUMENT
        sendAuthenticationInfoArg OCTET STRING (SIZE (3..8))
    RESULT
        sendAuthenticationInfoRes SEQUENCE SIZE (1..5) OF
            SEQUENCE {
                rand        OCTET STRING (SIZE (16)),
                sres        OCTET STRING (SIZE (4)),
                kc          OCTET STRING (SIZE (8)),
                ... }
    ERRORS {
        -- systemFailure -- localValue 34,
        -- dataMissing -- localValue 35,
        -- unexpectedDataValue -- localValue 36,
        -- unknownSubscriber -- localValue 1}
  ::= localValue 56

checkIMEI  OPERATION
    ARGUMENT
        imei        OCTET STRING (SIZE (8))
    RESULT
        equipmentStatus ENUMERATED {
            whiteListed  (0),
            blackListed  (1),
            greyListed   (2)}
    ERRORS {
        -- systemFailure -- localValue 34,
        -- dataMissing -- localValue 35,
        -- unexpectedDataValue -- localValue 36,
        -- unknownEquipment -- localValue 7}
  ::= localValue 43

sendParameters OPERATION
    ARGUMENT
        sendParametersArg SEQUENCE {
            subscriberId CHOICE {
                imsi    [0] IMPLICIT OCTET STRING (SIZE (3..8)),
                tmsi    [1] IMPLICIT OCTET STRING (SIZE (1..4))},
            requestParameterList SEQUENCE SIZE (1..2) OF
                ENUMERATED {
                    requestIMSI  (0),
                    requestAuthenticationSet  (1),
                    requestSubscriberData  (2),
                    requestKi  (4)}}
    RESULT
        sentParameterList SEQUENCE SIZE (1..6) OF
            CHOICE {
                imsi    [0] IMPLICIT OCTET STRING (SIZE (3..8)),
                authenticationSet [1] IMPLICIT SEQUENCE {
                    rand        OCTET STRING (SIZE (16)),
                    sres        OCTET STRING (SIZE (4)),
                    kc          OCTET STRING (SIZE (8)),
                    ... },
                subscriberData [2] IMPLICIT SEQUENCE {
                    msisdn    [1] IMPLICIT OCTET STRING (SIZE (1..9)) OPTIONAL,
                    category  [2] IMPLICIT OCTET STRING (SIZE (1)) OPTIONAL,
                    subscriberStatus [3] IMPLICIT ENUMERATED {
                        serviceGranted  (0),
                        operatorDeterminedBarring  (1)} OPTIONAL,
                    bearerServiceList [4] IMPLICIT SEQUENCE SIZE (1..50) OF
                        OCTET STRING (SIZE (1)) OPTIONAL,
                    teleserviceList [6] IMPLICIT SEQUENCE SIZE (1..20) OF
                        OCTET STRING (SIZE (1)) OPTIONAL,
                    provisionedSS [7] IMPLICIT SEQUENCE SIZE (1..30) OF
                        CHOICE {
                            forwardingInfo [0] IMPLICIT SEQUENCE {
```

```
                        ss-Code    OCTET STRING (SIZE (1)) OPTIONAL,
                        forwardingFeatureList SEQUENCE SIZE (1..13) OF
                           SEQUENCE {
                              basicService CHOICE {
                                 bearerService [2] IMPLICIT OCTET STRING (SIZE (1)),
                                 teleservice [3] IMPLICIT OCTET STRING (SIZE (1))} OPTIONAL,
                                 ss-Status  [4] IMPLICIT OCTET STRING (SIZE (1)) OPTIONAL,
                                 forwardedToNumber [5] IMPLICIT OCTET STRING (SIZE (1..9))
OPTIONAL,
                                 forwardedToSubaddress [8] IMPLICIT OCTET STRING (SIZE (1..21))
OPTIONAL,
                                 forwardingOptions [6] IMPLICIT OCTET STRING (SIZE (1))
OPTIONAL,
                                 noReplyConditionTime [7] IMPLICIT INTEGER (5..30) OPTIONAL,
                                 ... },
                           ... },
                     callBarringInfo [1] IMPLICIT SEQUENCE {
                        ss-Code    OCTET STRING (SIZE (1)) OPTIONAL,
                        callBarringFeatureList SEQUENCE SIZE (1..13) OF
                           SEQUENCE {
                              basicService CHOICE {
                                 bearerService [2] IMPLICIT OCTET STRING (SIZE (1)),
                                 teleservice [3] IMPLICIT OCTET STRING (SIZE (1))} OPTIONAL,
                                 ss-Status  [4] IMPLICIT OCTET STRING (SIZE (1)) OPTIONAL,
                                 ... },
                           ... },
                     cug-Info  [2] IMPLICIT SEQUENCE {
                        cug-SubscriptionList SEQUENCE SIZE (1..10) OF
                           SEQUENCE {
                              cug-Index  INTEGER (0..32767),
                              cug-Interlock OCTET STRING (SIZE (4)),
                              intraCUG-Options ENUMERATED {
                                 noCUG-Restrictions  (0),
                                 cugIC-CallBarred  (1),
                                 cugOG-CallBarred  (2)},
                              basicServiceGroupList SEQUENCE SIZE (1..13) OF
                                 CHOICE {
                                    bearerService [2] IMPLICIT OCTET STRING (SIZE (1)),
                                    teleservice [3] IMPLICIT OCTET STRING (SIZE (1))}
OPTIONAL,
                              ... },
                        cug-FeatureList SEQUENCE SIZE (1..13) OF
                           SEQUENCE {
                              basicService CHOICE {
                                 bearerService [2] IMPLICIT OCTET STRING (SIZE (1)),
                                 teleservice [3] IMPLICIT OCTET STRING (SIZE (1))} OPTIONAL,
                              preferentialCUG-Indicator INTEGER (0..32767) OPTIONAL,
                              interCUG-Restrictions OCTET STRING (SIZE (1)),
                              ... } OPTIONAL,
                        ... },
                     ss-Data   [3] IMPLICIT SEQUENCE {
                        ss-Code    OCTET STRING (SIZE (1)) OPTIONAL,
                        ss-Status [4] IMPLICIT OCTET STRING (SIZE (1)) OPTIONAL,
                        ss-SubscriptionOption CHOICE {
                           cliRestrictionOption [2] IMPLICIT ENUMERATED {
                              permanent  (0),
                              temporaryDefaultRestricted  (1),
                              temporaryDefaultAllowed  (2)},
                           overrideCategory [1] IMPLICIT ENUMERATED {
                              overrideEnabled  (0),
                              overrideDisabled  (1)}} OPTIONAL,
                        basicServiceGroupList SEQUENCE SIZE (1..13) OF
                           CHOICE {
                              bearerService [2] IMPLICIT OCTET STRING (SIZE (1)),
                              teleservice [3] IMPLICIT OCTET STRING (SIZE (1))} OPTIONAL,
                        ... }} OPTIONAL,
               odb-Data   [8] IMPLICIT SEQUENCE {
                  odb-GeneralData BIT STRING {
                     allOG-CallsBarred  (0),
                     internationalOGCallsBarred  (1),
                     internationalOGCallsNotToHPLMN-CountryBarred  (2),
                     premiumRateInformationOGCallsBarred  (3),
                     premiumRateEntertainementOGCallsBarred  (4),
                     ss-AccessBarred  (5)} (SIZE (6)),
                  odb-HPLMN-Data BIT STRING {
                     plmn-SpecificBarringType1  (0),
                     plmn-SpecificBarringType2  (1),
                     plmn-SpecificBarringType3  (2),
                     plmn-SpecificBarringType4  (3)} (SIZE (4)) OPTIONAL,
                  ... } OPTIONAL,
            roamingRestrictionDueToUnsupportedFeature [9] IMPLICIT NULL OPTIONAL,
```

```
                    regionalSubscriptionData [10] IMPLICIT SEQUENCE SIZE (1..10) OF
                        OCTET STRING (SIZE (2)) OPTIONAL},
                ki          [4] IMPLICIT OCTET STRING (SIZE (16))}
        ERRORS {
            -- unexpectedDataValue -- localValue 36,
            -- unknownSubscriber -- localValue 1,
            -- unidentifiedSubscriber -- localValue 5}
     ::= localValue 9


    insertSubscriberData OPERATION
        ARGUMENT
            insertSubscriberDataArg SEQUENCE {
                imsi        [0] IMPLICIT OCTET STRING (SIZE (3..8)) OPTIONAL,
                msisdn      [1] IMPLICIT OCTET STRING (SIZE (1..9)) OPTIONAL,
                category    [2] IMPLICIT OCTET STRING (SIZE (1)) OPTIONAL,
                subscriberStatus [3] IMPLICIT ENUMERATED {
                    serviceGranted  (0),
                    operatorDeterminedBarring  (1)} OPTIONAL,
                bearerServiceList [4] IMPLICIT SEQUENCE SIZE (1..50) OF
                    OCTET STRING (SIZE (1)) OPTIONAL,
                teleserviceList [6] IMPLICIT SEQUENCE SIZE (1..20) OF
                    OCTET STRING (SIZE (1)) OPTIONAL,
                provisionedSS [7] IMPLICIT SEQUENCE SIZE (1..30) OF
                    CHOICE {
                        forwardingInfo [0] IMPLICIT SEQUENCE {
                            ss-Code    OCTET STRING (SIZE (1)) OPTIONAL,
                            forwardingFeatureList SEQUENCE SIZE (1..13) OF
                                SEQUENCE {
                                    basicService CHOICE {
                                        bearerService [2] IMPLICIT OCTET STRING (SIZE (1)),
                                        teleservice [3] IMPLICIT OCTET STRING (SIZE (1))} OPTIONAL,
                                    ss-Status  [4] IMPLICIT OCTET STRING (SIZE (1)) OPTIONAL,
                                    forwardedToNumber [5] IMPLICIT OCTET STRING (SIZE (1..9)) OPTIONAL,
                                    forwardedToSubaddress [8] IMPLICIT OCTET STRING (SIZE (1..21))
    OPTIONAL,
                                    forwardingOptions [6] IMPLICIT OCTET STRING (SIZE (1)) OPTIONAL,
                                    noReplyConditionTime [7] IMPLICIT INTEGER (5..30) OPTIONAL,
                                    ... },
                            ... },
                        callBarringInfo [1] IMPLICIT SEQUENCE {
                            ss-Code    OCTET STRING (SIZE (1)) OPTIONAL,
                            callBarringFeatureList SEQUENCE SIZE (1..13) OF
                                SEQUENCE {
                                    basicService CHOICE {
                                        bearerService [2] IMPLICIT OCTET STRING (SIZE (1)),
                                        teleservice [3] IMPLICIT OCTET STRING (SIZE (1))} OPTIONAL,
                                    ss-Status  [4] IMPLICIT OCTET STRING (SIZE (1)) OPTIONAL,
                                    ... },
                            ... },
                        cug-Info   [2] IMPLICIT SEQUENCE {
                            cug-SubscriptionList SEQUENCE SIZE (1..10) OF
                                SEQUENCE {
                                    cug-Index  INTEGER (0..32767),
                                    cug-Interlock OCTET STRING (SIZE (4)),
                                    intraCUG-Options ENUMERATED {
                                        noCUG-Restrictions  (0),
                                        cugIC-CallBarred  (1),
                                        cugOG-CallBarred  (2)},
                                    basicServiceGroupList SEQUENCE SIZE (1..13) OF
                                        CHOICE {
                                            bearerService [2] IMPLICIT OCTET STRING (SIZE (1)),
                                            teleservice [3] IMPLICIT OCTET STRING (SIZE (1))} OPTIONAL,
                                    ... },
                            cug-FeatureList SEQUENCE SIZE (1..13) OF
                                SEQUENCE {
                                    basicService CHOICE {
                                        bearerService [2] IMPLICIT OCTET STRING (SIZE (1)),
                                        teleservice [3] IMPLICIT OCTET STRING (SIZE (1))} OPTIONAL,
                                    preferentialCUG-Indicator INTEGER (0..32767) OPTIONAL,
                                    interCUG-Restrictions OCTET STRING (SIZE (1)),
                                    ... } OPTIONAL,
                            ... },
                        ss-Data    [3] IMPLICIT SEQUENCE {
                            ss-Code    OCTET STRING (SIZE (1)) OPTIONAL,
                            ss-Status  [4] IMPLICIT OCTET STRING (SIZE (1)) OPTIONAL,
                            ss-SubscriptionOption CHOICE {
                                cliRestrictionOption [2] IMPLICIT ENUMERATED {
                                    permanent  (0),
                                    temporaryDefaultRestricted  (1),
                                    temporaryDefaultAllowed  (2)},
                                overrideCategory [1] IMPLICIT ENUMERATED {
```

```
                              overrideEnabled  (0),
                              overrideDisabled  (1)}} OPTIONAL,
                   basicServiceGroupList SEQUENCE SIZE (1..13) OF
                        CHOICE {
                           bearerService [2] IMPLICIT OCTET STRING (SIZE (1)),
                           teleservice [3] IMPLICIT OCTET STRING (SIZE (1))} OPTIONAL,
                   ... }} OPTIONAL,
            odb-Data   [8] IMPLICIT SEQUENCE {
               odb-GeneralData BIT STRING {
                  allOG-CallsBarred  (0),
                  internationalOGCallsBarred  (1),
                  internationalOGCallsNotToHPLMN-CountryBarred  (2),
                  premiumRateInformationOGCallsBarred  (3),
                  premiumRateEntertainementOGCallsBarred  (4),
                  ss-AccessBarred  (5)} (SIZE (6)),
               odb-HPLMN-Data BIT STRING {
                  plmn-SpecificBarringType1  (0),
                  plmn-SpecificBarringType2  (1),
                  plmn-SpecificBarringType3  (2),
                  plmn-SpecificBarringType4  (3)} (SIZE (4)) OPTIONAL,
               ... } OPTIONAL,
            roamingRestrictionDueToUnsupportedFeature [9] IMPLICIT NULL OPTIONAL,
            regionalSubscriptionData [10] IMPLICIT SEQUENCE SIZE (1..10) OF
               OCTET STRING (SIZE (2)) OPTIONAL,
            ... }
   RESULT
      insertSubscriberDataRes SEQUENCE {
         teleserviceList [1] IMPLICIT SEQUENCE SIZE (1..20) OF
            OCTET STRING (SIZE (1)) OPTIONAL,
         bearerServiceList [2] IMPLICIT SEQUENCE SIZE (1..50) OF
            OCTET STRING (SIZE (1)) OPTIONAL,
         ss-List    [3] IMPLICIT SEQUENCE SIZE (1..30) OF
            OCTET STRING (SIZE (1)) OPTIONAL,
         odb-GeneralData [4] IMPLICIT BIT STRING {
            allOG-CallsBarred  (0),
            internationalOGCallsBarred  (1),
            internationalOGCallsNotToHPLMN-CountryBarred  (2),
            premiumRateInformationOGCallsBarred  (3),
            premiumRateEntertainementOGCallsBarred  (4),
            ss-AccessBarred  (5)} (SIZE (6)) OPTIONAL,
         regionalSubscriptionResponse [5] IMPLICIT ENUMERATED {
            msc-AreaRestricted  (0),
            tooManyZoneCodes  (1),
            zoneCodesConflict  (2),
            regionalSubscNotSupported  (3)} OPTIONAL,
         ... }
   ERRORS {
      -- dataMissing -- localValue 35,
      -- unexpectedDataValue -- localValue 36,
      -- unidentifiedSubscriber -- localValue 5}
 ::= localValue 7

deleteSubscriberData OPERATION
   ARGUMENT
      deleteSubscriberDataArg SEQUENCE {
         imsi       [0] IMPLICIT OCTET STRING (SIZE (3..8)),
         basicServiceList [1] IMPLICIT SEQUENCE SIZE (1..70) OF
            CHOICE {
               bearerService [2] IMPLICIT OCTET STRING (SIZE (1)),
               teleservice [3] IMPLICIT OCTET STRING (SIZE (1))} OPTIONAL,
         ss-List    [2] IMPLICIT SEQUENCE SIZE (1..30) OF
            OCTET STRING (SIZE (1)) OPTIONAL,
         roamingRestrictionDueToUnsupportedFeature [4] IMPLICIT NULL OPTIONAL,
         regionalSubscriptionIdentifier [5] IMPLICIT OCTET STRING (SIZE (2)) OPTIONAL,
         ... }
   RESULT
      deleteSubscriberDataRes SEQUENCE {
         regionalSubscriptionResponse [0] IMPLICIT ENUMERATED {
            msc-AreaRestricted  (0),
            tooManyZoneCodes  (1),
            zoneCodesConflict  (2),
            regionalSubscNotSupported  (3)} OPTIONAL,
         ... }
   ERRORS {
      -- dataMissing -- localValue 35,
      -- unexpectedDataValue -- localValue 36,
      -- unidentifiedSubscriber -- localValue 5}
 ::= localValue 8

reset      OPERATION
   ARGUMENT
```

```
    resetArg    SEQUENCE {
       networkResource ENUMERATED {
          plmn        (0),
          hlr         (1),
          vlr         (2),
          pvlr        (3),
          controllingMSC  (4),
          vmsc        (5),
          eir         (6),
          rss         (7)} OPTIONAL,
       hlr-Number OCTET STRING (SIZE (1..9)),
       hlr-List   SEQUENCE SIZE (1..50) OF
          OCTET STRING (SIZE (3..8)) OPTIONAL,
       ... }
 ::= localValue 37

forwardCheckSS-Indication OPERATION
 ::= localValue 38

restoreData OPERATION
   ARGUMENT
      restoreDataArg SEQUENCE {
         imsi       OCTET STRING (SIZE (3..8)),
         lmsi       OCTET STRING (SIZE (4)) OPTIONAL,
         ... }
   RESULT
      restoreDataRes SEQUENCE {
         hlr-Number OCTET STRING (SIZE (1..9)),
         msNotReachable NULL OPTIONAL,
         ... }
   ERRORS {
      -- systemFailure -- localValue 34,
      -- dataMissing -- localValue 35,
      -- unexpectedDataValue -- localValue 36,
      -- unknownSubscriber -- localValue 1}
 ::= localValue 57

activateTraceMode OPERATION
   ARGUMENT
      activateTraceModeArg SEQUENCE {
         imsi      [0] IMPLICIT OCTET STRING (SIZE (3..8)) OPTIONAL,
         traceReference [1] IMPLICIT OCTET STRING (SIZE (1..2)),
         traceType [2] IMPLICIT INTEGER (0..255),
         omc-Id    [3] IMPLICIT OCTET STRING (SIZE (1..20)) OPTIONAL,
         ... }
   ERRORS {
      -- systemFailure -- localValue 34,
      -- dataMissing -- localValue 35,
      -- unexpectedDataValue -- localValue 36,
      -- facilityNotSupported -- localValue 21,
      -- unidentifiedSubscriber -- localValue 5,
      -- tracingBufferFull -- localValue 40}
 ::= localValue 50

deactivateTraceMode OPERATION
   ARGUMENT
      deactivateTraceModeArg SEQUENCE {
         imsi      [0] IMPLICIT OCTET STRING (SIZE (3..8)) OPTIONAL,
         traceReference [1] IMPLICIT OCTET STRING (SIZE (1..2)),
         ... }
   ERRORS {
      -- systemFailure -- localValue 34,
      -- dataMissing -- localValue 35,
      -- unexpectedDataValue -- localValue 36,
      -- facilityNotSupported -- localValue 21,
      -- unidentifiedSubscriber -- localValue 5}
 ::= localValue 51

traceSubscriberActivity OPERATION
   ARGUMENT
      traceSubscriberActivityArg SEQUENCE {
         imsi      [0] IMPLICIT OCTET STRING (SIZE (3..8)) OPTIONAL,
         traceReference [1] IMPLICIT OCTET STRING (SIZE (1..2)),
         traceType [2] IMPLICIT INTEGER (0..255),
         omc-Id    [3] IMPLICIT OCTET STRING (SIZE (1..20)) OPTIONAL,
         callReference [4] IMPLICIT OCTET STRING (SIZE (1..3)) OPTIONAL}
 ::= localValue 52

noteInternalHandover OPERATION
   ARGUMENT
      noteInternalHO-Arg SEQUENCE {
```

```
            handoverType ENUMERATED {
                interBSS    (0),
                intraBSS    (1)},
            targetCellId [1] IMPLICIT OCTET STRING (SIZE (5..7)) OPTIONAL,
            channelId [2] IMPLICIT SEQUENCE {
                protocolId ENUMERATED {
                    gsm-0408    (1),
                    gsm-0806    (2),
                    gsm-BSSMAP  (3),
                    ets-300102-1  (4)},
                signalInfo OCTET STRING (SIZE (1..200)),
                ... } OPTIONAL}
     ::= localValue 35

sendIMSI   OPERATION
    ARGUMENT
        msisdn     OCTET STRING (SIZE (1..9))
    RESULT
        imsi       OCTET STRING (SIZE (3..8))
    ERRORS {
        -- dataMissing -- localValue 35,
        -- unexpectedDataValue -- localValue 36,
        -- unknownSubscriber -- localValue 1}
     ::= localValue 58

sendRoutingInfo OPERATION
    ARGUMENT
        sendRoutingInfoArg SEQUENCE {
            msisdn     [0] IMPLICIT OCTET STRING (SIZE (1..9)),
            cug-CheckInfo [1] IMPLICIT SEQUENCE {
                cug-Interlock OCTET STRING (SIZE (4)),
                cug-OutgoingAccess NULL OPTIONAL,
                ... } OPTIONAL,
            numberOfForwarding [2] IMPLICIT INTEGER (1..5) OPTIONAL,
            networkSignalInfo [10] IMPLICIT SEQUENCE {
                protocolId ENUMERATED {
                    gsm-0408    (1),
                    gsm-0806    (2),
                    gsm-BSSMAP  (3),
                    ets-300102-1  (4)},
                signalInfo OCTET STRING (SIZE (1..200)),
                ... } OPTIONAL,
            ... }
    RESULT
        sendRoutingInfoRes SEQUENCE {
            imsi       OCTET STRING (SIZE (3..8)),
            routingInfo CHOICE {
                roamingNumber OCTET STRING (SIZE (1..9)),
                forwardingData SEQUENCE {
                    forwardedToNumber [5] IMPLICIT OCTET STRING (SIZE (1..9)) OPTIONAL,
                    forwardedToSubaddress [4] IMPLICIT OCTET STRING (SIZE (1..21)) OPTIONAL,
                    forwardingOptions [6] IMPLICIT OCTET STRING (SIZE (1)) OPTIONAL,
                    ... }},
            cug-CheckInfo SEQUENCE {
                cug-Interlock OCTET STRING (SIZE (4)),
                cug-OutgoingAccess NULL OPTIONAL,
                ... } OPTIONAL,
            ... }
    ERRORS {
        -- systemFailure -- localValue 34,
        -- dataMissing -- localValue 35,
        -- unexpectedDataValue -- localValue 36,
        -- facilityNotSupported -- localValue 21,
        -- unknownSubscriber -- localValue 1,
        -- numberChanged -- localValue 44,
        -- bearerServiceNotProvisioned -- localValue 10,
        -- teleserviceNotProvisioned -- localValue 11,
        -- absentSubscriber -- localValue 27,
        -- callBarred -- localValue 13,
        -- cug-Reject -- localValue 15,
        -- forwardingViolation -- localValue 14}
     ::= localValue 22

provideRoamingNumber OPERATION
    ARGUMENT
        provideRoamingNumberArg SEQUENCE {
            imsi       [0] IMPLICIT OCTET STRING (SIZE (3..8)),
            msc-Number [1] IMPLICIT OCTET STRING (SIZE (1..9)) OPTIONAL,
            msisdn     [2] IMPLICIT OCTET STRING (SIZE (1..9)) OPTIONAL,
            previousRoamingNumber [3] IMPLICIT OCTET STRING (SIZE (1..9)) OPTIONAL,
            lmsi       [4] IMPLICIT OCTET STRING (SIZE (4)) OPTIONAL,
```

```
                gsm-BearerCapability [5] IMPLICIT SEQUENCE {
                    protocolId ENUMERATED {
                        gsm-0408    (1),
                        gsm-0806    (2),
                        gsm-BSSMAP  (3),
                        ets-300102-1  (4)},
                    signalInfo OCTET STRING (SIZE (1..200)),
                    ... } OPTIONAL,
                networkSignalInfo [6] IMPLICIT SEQUENCE {
                    protocolId ENUMERATED {
                        gsm-0408    (1),
                        gsm-0806    (2),
                        gsm-BSSMAP  (3),
                        ets-300102-1  (4)},
                    signalInfo OCTET STRING (SIZE (1..200)),
                    ... } OPTIONAL,
                ... }
        RESULT
            roamingNumber OCTET STRING (SIZE (1..9))
        ERRORS {
            -- systemFailure -- localValue 34,
            -- dataMissing -- localValue 35,
            -- unexpectedDataValue -- localValue 36,
            -- facilityNotSupported -- localValue 21,
            -- absentSubscriber -- localValue 27,
            -- noRoamingNumberAvailable -- localValue 39}
     ::= localValue 4

    registerSS OPERATION
        ARGUMENT
            registerSS-Arg SEQUENCE {
                ss-Code    OCTET STRING (SIZE (1)),
                basicService CHOICE {
                    bearerService [2] IMPLICIT OCTET STRING (SIZE (1)),
                    teleservice [3] IMPLICIT OCTET STRING (SIZE (1))} OPTIONAL,
                forwardedToNumber [4] IMPLICIT OCTET STRING (SIZE (1..20)) OPTIONAL,
                forwardedToSubaddress [6] IMPLICIT OCTET STRING (SIZE (1..21)) OPTIONAL,
                noReplyConditionTime [5] IMPLICIT INTEGER (5..30) OPTIONAL,
                ... }
        RESULT
            ss-Info    CHOICE {
                forwardingInfo [0] IMPLICIT SEQUENCE {
                    ss-Code    OCTET STRING (SIZE (1)) OPTIONAL,
                    forwardingFeatureList SEQUENCE SIZE (1..13) OF
                        SEQUENCE {
                            basicService CHOICE {
                                bearerService [2] IMPLICIT OCTET STRING (SIZE (1)),
                                teleservice [3] IMPLICIT OCTET STRING (SIZE (1))} OPTIONAL,
                            ss-Status  [4] IMPLICIT OCTET STRING (SIZE (1)) OPTIONAL,
                            forwardedToNumber [5] IMPLICIT OCTET STRING (SIZE (1..9)) OPTIONAL,
                            forwardedToSubaddress [8] IMPLICIT OCTET STRING (SIZE (1..21)) OPTIONAL,
                            forwardingOptions [6] IMPLICIT OCTET STRING (SIZE (1)) OPTIONAL,
                            noReplyConditionTime [7] IMPLICIT INTEGER (5..30) OPTIONAL,
                            ... },
                    ... },
                callBarringInfo [1] IMPLICIT SEQUENCE {
                    ss-Code    OCTET STRING (SIZE (1)) OPTIONAL,
                    callBarringFeatureList SEQUENCE SIZE (1..13) OF
                        SEQUENCE {
                            basicService CHOICE {
                                bearerService [2] IMPLICIT OCTET STRING (SIZE (1)),
                                teleservice [3] IMPLICIT OCTET STRING (SIZE (1))} OPTIONAL,
                            ss-Status  [4] IMPLICIT OCTET STRING (SIZE (1)) OPTIONAL,
                            ... },
                    ... },
                cug-Info   [2] IMPLICIT SEQUENCE {
                    cug-SubscriptionList SEQUENCE SIZE (1..10) OF
                        SEQUENCE {
                            cug-Index  INTEGER (0..32767),
                            cug-Interlock OCTET STRING (SIZE (4)),
                            intraCUG-Options ENUMERATED {
                                noCUG-Restrictions  (0),
                                cugIC-CallBarred  (1),
                                cugOG-CallBarred  (2)},
                            basicServiceGroupList SEQUENCE SIZE (1..13) OF
                                CHOICE {
                                    bearerService [2] IMPLICIT OCTET STRING (SIZE (1)),
                                    teleservice [3] IMPLICIT OCTET STRING (SIZE (1))} OPTIONAL,
                            ... },
                    cug-FeatureList SEQUENCE SIZE (1..13) OF
                        SEQUENCE {
```

```
                    basicService CHOICE {
                        bearerService [2] IMPLICIT OCTET STRING (SIZE (1)),
                        teleservice [3] IMPLICIT OCTET STRING (SIZE (1))} OPTIONAL,
                    preferentialCUG-Indicator INTEGER (0..32767) OPTIONAL,
                    interCUG-Restrictions OCTET STRING (SIZE (1)),
                    ... } OPTIONAL,
            ... },
        ss-Data   [3] IMPLICIT SEQUENCE {
            ss-Code   OCTET STRING (SIZE (1)) OPTIONAL,
            ss-Status [4] IMPLICIT OCTET STRING (SIZE (1)) OPTIONAL,
            ss-SubscriptionOption CHOICE {
                cliRestrictionOption [2] IMPLICIT ENUMERATED {
                    permanent   (0),
                    temporaryDefaultRestricted  (1),
                    temporaryDefaultAllowed  (2)},
                overrideCategory [1] IMPLICIT ENUMERATED {
                    overrideEnabled  (0),
                    overrideDisabled  (1)}} OPTIONAL,
            basicServiceGroupList SEQUENCE SIZE (1..13) OF
                CHOICE {
                    bearerService [2] IMPLICIT OCTET STRING (SIZE (1)),
                    teleservice [3] IMPLICIT OCTET STRING (SIZE (1))} OPTIONAL,
            ... }}
    ERRORS {
        -- systemFailure -- localValue 34,
        -- dataMissing -- localValue 35,
        -- unexpectedDataValue -- localValue 36,
        -- unknownSubscriber -- localValue 1,
        -- bearerServiceNotProvisioned -- localValue 10,
        -- teleserviceNotProvisioned -- localValue 11,
        -- callBarred -- localValue 13,
        -- illegalSS-Operation -- localValue 16,
        -- ss-ErrorStatus -- localValue 17,
        -- ss-SubscriptionViolation -- localValue 19,
        -- ss-Incompatibility -- localValue 20}
    ::= localValue 10

eraseSS   OPERATION
    ARGUMENT
        ss-ForBS   SEQUENCE {
            ss-Code    OCTET STRING (SIZE (1)),
            basicService CHOICE {
                bearerService [2] IMPLICIT OCTET STRING (SIZE (1)),
                teleservice [3] IMPLICIT OCTET STRING (SIZE (1))} OPTIONAL,
            ... }
    RESULT
        ss-Info    CHOICE {
            forwardingInfo [0] IMPLICIT SEQUENCE {
                ss-Code    OCTET STRING (SIZE (1)) OPTIONAL,
                forwardingFeatureList SEQUENCE SIZE (1..13) OF
                    SEQUENCE {
                        basicService CHOICE {
                            bearerService [2] IMPLICIT OCTET STRING (SIZE (1)),
                            teleservice [3] IMPLICIT OCTET STRING (SIZE (1))} OPTIONAL,
                        ss-Status  [4] IMPLICIT OCTET STRING (SIZE (1)) OPTIONAL,
                        forwardedToNumber [5] IMPLICIT OCTET STRING (SIZE (1..9)) OPTIONAL,
                        forwardedToSubaddress [8] IMPLICIT OCTET STRING (SIZE (1..21)) OPTIONAL,
                        forwardingOptions [6] IMPLICIT OCTET STRING (SIZE (1)) OPTIONAL,
                        noReplyConditionTime [7] IMPLICIT INTEGER (5..30) OPTIONAL,
                        ... },
                ... },
            callBarringInfo [1] IMPLICIT SEQUENCE {
                ss-Code    OCTET STRING (SIZE (1)) OPTIONAL,
                callBarringFeatureList SEQUENCE SIZE (1..13) OF
                    SEQUENCE {
                        basicService CHOICE {
                            bearerService [2] IMPLICIT OCTET STRING (SIZE (1)),
                            teleservice [3] IMPLICIT OCTET STRING (SIZE (1))} OPTIONAL,
                        ss-Status  [4] IMPLICIT OCTET STRING (SIZE (1)) OPTIONAL,
                        ... },
                ... },
            cug-Info   [2] IMPLICIT SEQUENCE {
                cug-SubscriptionList SEQUENCE SIZE (1..10) OF
                    SEQUENCE {
                        cug-Index  INTEGER (0..32767),
                        cug-Interlock OCTET STRING (SIZE (4)),
                        intraCUG-Options ENUMERATED {
                            noCUG-Restrictions  (0),
                            cugIC-CallBarred  (1),
                            cugOG-CallBarred  (2)},
                        basicServiceGroupList SEQUENCE SIZE (1..13) OF
```

```
                        CHOICE {
                            bearerService [2] IMPLICIT OCTET STRING (SIZE (1)),
                            teleservice [3] IMPLICIT OCTET STRING (SIZE (1))} OPTIONAL,
                        ... },
                cug-FeatureList SEQUENCE SIZE (1..13) OF
                    SEQUENCE {
                        basicService CHOICE {
                            bearerService [2] IMPLICIT OCTET STRING (SIZE (1)),
                            teleservice [3] IMPLICIT OCTET STRING (SIZE (1))} OPTIONAL,
                        preferentialCUG-Indicator INTEGER (0..32767) OPTIONAL,
                        interCUG-Restrictions OCTET STRING (SIZE (1)),
                        ... } OPTIONAL,
                ... },
            ss-Data   [3] IMPLICIT SEQUENCE {
                ss-Code   OCTET STRING (SIZE (1)) OPTIONAL,
                ss-Status [4] IMPLICIT OCTET STRING (SIZE (1)) OPTIONAL,
                ss-SubscriptionOption CHOICE {
                    cliRestrictionOption [2] IMPLICIT ENUMERATED {
                        permanent   (0),
                        temporaryDefaultRestricted  (1),
                        temporaryDefaultAllowed  (2)},
                    overrideCategory [1] IMPLICIT ENUMERATED {
                        overrideEnabled  (0),
                        overrideDisabled  (1)}} OPTIONAL,
                basicServiceGroupList SEQUENCE SIZE (1..13) OF
                    CHOICE {
                        bearerService [2] IMPLICIT OCTET STRING (SIZE (1)),
                        teleservice [3] IMPLICIT OCTET STRING (SIZE (1))} OPTIONAL,
                ... }}
    ERRORS {
        -- systemFailure -- localValue 34,
        -- dataMissing -- localValue 35,
        -- unexpectedDataValue -- localValue 36,
        -- unknownSubscriber -- localValue 1,
        -- bearerServiceNotProvisioned -- localValue 10,
        -- teleserviceNotProvisioned -- localValue 11,
        -- callBarred -- localValue 13,
        -- illegalSS-Operation -- localValue 16,
        -- ss-ErrorStatus -- localValue 17,
        -- ss-SubscriptionViolation -- localValue 19}
    ::= localValue 11

activateSS OPERATION
    ARGUMENT
        ss-ForBS   SEQUENCE {
            ss-Code   OCTET STRING (SIZE (1)),
            basicService CHOICE {
                bearerService [2] IMPLICIT OCTET STRING (SIZE (1)),
                teleservice [3] IMPLICIT OCTET STRING (SIZE (1))} OPTIONAL,
            ... }
    RESULT
        ss-Info   CHOICE {
            forwardingInfo [0] IMPLICIT SEQUENCE {
                ss-Code   OCTET STRING (SIZE (1)) OPTIONAL,
                forwardingFeatureList SEQUENCE SIZE (1..13) OF
                    SEQUENCE {
                        basicService CHOICE {
                            bearerService [2] IMPLICIT OCTET STRING (SIZE (1)),
                            teleservice [3] IMPLICIT OCTET STRING (SIZE (1))} OPTIONAL,
                        ss-Status [4] IMPLICIT OCTET STRING (SIZE (1)) OPTIONAL,
                        forwardedToNumber [5] IMPLICIT OCTET STRING (SIZE (1..9)) OPTIONAL,
                        forwardedToSubaddress [8] IMPLICIT OCTET STRING (SIZE (1..21)) OPTIONAL,
                        forwardingOptions [6] IMPLICIT OCTET STRING (SIZE (1)) OPTIONAL,
                        noReplyConditionTime [7] IMPLICIT INTEGER (5..30) OPTIONAL,
                        ... },
                ... },
            callBarringInfo [1] IMPLICIT SEQUENCE {
                ss-Code   OCTET STRING (SIZE (1)) OPTIONAL,
                callBarringFeatureList SEQUENCE SIZE (1..13) OF
                    SEQUENCE {
                        basicService CHOICE {
                            bearerService [2] IMPLICIT OCTET STRING (SIZE (1)),
                            teleservice [3] IMPLICIT OCTET STRING (SIZE (1))} OPTIONAL,
                        ss-Status [4] IMPLICIT OCTET STRING (SIZE (1)) OPTIONAL,
                        ... },
                ... },
            cug-Info   [2] IMPLICIT SEQUENCE {
                cug-SubscriptionList SEQUENCE SIZE (1..10) OF
                    SEQUENCE {
                        cug-Index  INTEGER (0..32767),
                        cug-Interlock OCTET STRING (SIZE (4)),
```

```
                        intraCUG-Options ENUMERATED {
                            noCUG-Restrictions  (0),
                            cugIC-CallBarred  (1),
                            cugOG-CallBarred  (2)},
                        basicServiceGroupList SEQUENCE SIZE (1..13) OF
                            CHOICE {
                                bearerService [2] IMPLICIT OCTET STRING (SIZE (1)),
                                teleservice [3] IMPLICIT OCTET STRING (SIZE (1))} OPTIONAL,
                        ... },
                cug-FeatureList SEQUENCE SIZE (1..13) OF
                    SEQUENCE {
                        basicService CHOICE {
                            bearerService [2] IMPLICIT OCTET STRING (SIZE (1)),
                            teleservice [3] IMPLICIT OCTET STRING (SIZE (1))} OPTIONAL,
                        preferentialCUG-Indicator INTEGER (0..32767) OPTIONAL,
                        interCUG-Restrictions OCTET STRING (SIZE (1)),
                        ... } OPTIONAL,
                ... },
            ss-Data    [3] IMPLICIT SEQUENCE {
                ss-Code    OCTET STRING (SIZE (1)) OPTIONAL,
                ss-Status  [4] IMPLICIT OCTET STRING (SIZE (1)) OPTIONAL,
                ss-SubscriptionOption CHOICE {
                    cliRestrictionOption [2] IMPLICIT ENUMERATED {
                    permanent    (0),
                    temporaryDefaultRestricted  (1),
                    temporaryDefaultAllowed  (2)},
                    overrideCategory [1] IMPLICIT ENUMERATED {
                    overrideEnabled  (0),
                    overrideDisabled  (1)}} OPTIONAL,
                basicServiceGroupList SEQUENCE SIZE (1..13) OF
                    CHOICE {
                        bearerService [2] IMPLICIT OCTET STRING (SIZE (1)),
                        teleservice [3] IMPLICIT OCTET STRING (SIZE (1))} OPTIONAL,
                ... }}
    ERRORS {
        -- systemFailure -- localValue 34,
        -- dataMissing -- localValue 35,
        -- unexpectedDataValue -- localValue 36,
        -- unknownSubscriber -- localValue 1,
        -- bearerServiceNotProvisioned -- localValue 10,
        -- teleserviceNotProvisioned -- localValue 11,
        -- callBarred -- localValue 13,
        -- illegalSS-Operation -- localValue 16,
        -- ss-ErrorStatus -- localValue 17,
        -- ss-SubscriptionViolation -- localValue 19,
        -- ss-Incompatibility -- localValue 20,
        -- negativePW-Check -- localValue 38,
        -- numberOfPW-AttemptsViolation -- localValue 43}
    ::= localValue 12

deactivateSS OPERATION
    ARGUMENT
        ss-ForBS   SEQUENCE {
            ss-Code    OCTET STRING (SIZE (1)),
            basicService CHOICE {
                bearerService [2] IMPLICIT OCTET STRING (SIZE (1)),
                teleservice [3] IMPLICIT OCTET STRING (SIZE (1))} OPTIONAL,
            ... }
    RESULT
        ss-Info    CHOICE {
            forwardingInfo [0] IMPLICIT SEQUENCE {
                ss-Code    OCTET STRING (SIZE (1)) OPTIONAL,
                forwardingFeatureList SEQUENCE SIZE (1..13) OF
                    SEQUENCE {
                        basicService CHOICE {
                            bearerService [2] IMPLICIT OCTET STRING (SIZE (1)),
                            teleservice [3] IMPLICIT OCTET STRING (SIZE (1))} OPTIONAL,
                        ss-Status  [4] IMPLICIT OCTET STRING (SIZE (1)) OPTIONAL,
                        forwardedToNumber [5] IMPLICIT OCTET STRING (SIZE (1..9)) OPTIONAL,
                        forwardedToSubaddress [8] IMPLICIT OCTET STRING (SIZE (1..21)) OPTIONAL,
                        forwardingOptions [6] IMPLICIT OCTET STRING (SIZE (1)) OPTIONAL,
                        noReplyConditionTime [7] IMPLICIT INTEGER (5..30) OPTIONAL,
                        ... },
                ... },
            callBarringInfo [1] IMPLICIT SEQUENCE {
                ss-Code    OCTET STRING (SIZE (1)) OPTIONAL,
                callBarringFeatureList SEQUENCE SIZE (1..13) OF
                    SEQUENCE {
                        basicService CHOICE {
                            bearerService [2] IMPLICIT OCTET STRING (SIZE (1)),
                            teleservice [3] IMPLICIT OCTET STRING (SIZE (1))} OPTIONAL,
```

```
                          ss-Status  [4] IMPLICIT OCTET STRING (SIZE (1)) OPTIONAL,
                          ... },
                  ... },
              cug-Info   [2] IMPLICIT SEQUENCE {
                  cug-SubscriptionList SEQUENCE SIZE (1..10) OF
                      SEQUENCE {
                          cug-Index  INTEGER (0..32767),
                          cug-Interlock OCTET STRING (SIZE (4)),
                          intraCUG-Options ENUMERATED {
                              noCUG-Restrictions  (0),
                              cugIC-CallBarred  (1),
                              cugOG-CallBarred  (2)},
                          basicServiceGroupList SEQUENCE SIZE (1..13) OF
                              CHOICE {
                                  bearerService [2] IMPLICIT OCTET STRING (SIZE (1)),
                                  teleservice [3] IMPLICIT OCTET STRING (SIZE (1))} OPTIONAL,
                          ... },
                  cug-FeatureList SEQUENCE SIZE (1..13) OF
                      SEQUENCE {
                          basicService CHOICE {
                              bearerService [2] IMPLICIT OCTET STRING (SIZE (1)),
                              teleservice [3] IMPLICIT OCTET STRING (SIZE (1))} OPTIONAL,
                          preferentialCUG-Indicator INTEGER (0..32767) OPTIONAL,
                          interCUG-Restrictions OCTET STRING (SIZE (1)),
                          ... } OPTIONAL,
                  ... },
              ss-Data    [3] IMPLICIT SEQUENCE {
                  ss-Code    OCTET STRING (SIZE (1)) OPTIONAL,
                  ss-Status  [4] IMPLICIT OCTET STRING (SIZE (1)) OPTIONAL,
                  ss-SubscriptionOption CHOICE {
                      cliRestrictionOption [2] IMPLICIT ENUMERATED {
                          permanent   (0),
                          temporaryDefaultRestricted  (1),
                          temporaryDefaultAllowed  (2)},
                      overrideCategory [1] IMPLICIT ENUMERATED {
                          overrideEnabled  (0),
                          overrideDisabled  (1)}} OPTIONAL,
                  basicServiceGroupList SEQUENCE SIZE (1..13) OF
                      CHOICE {
                          bearerService [2] IMPLICIT OCTET STRING (SIZE (1)),
                          teleservice [3] IMPLICIT OCTET STRING (SIZE (1))} OPTIONAL,
                  ... }}
      ERRORS {
          -- systemFailure -- localValue 34,
          -- dataMissing -- localValue 35,
          -- unexpectedDataValue -- localValue 36,
          -- unknownSubscriber -- localValue 1,
          -- bearerServiceNotProvisioned -- localValue 10,
          -- teleserviceNotProvisioned -- localValue 11,
          -- callBarred -- localValue 13,
          -- illegalSS-Operation -- localValue 16,
          -- ss-ErrorStatus -- localValue 17,
          -- ss-SubscriptionViolation -- localValue 19,
          -- negativePW-Check -- localValue 38,
          -- numberOfPW-AttemptsViolation -- localValue 43}
       ::= localValue 13

  interrogateSS OPERATION
      ARGUMENT
          ss-ForBS   SEQUENCE {
              ss-Code    OCTET STRING (SIZE (1)),
              basicService CHOICE {
                  bearerService [2] IMPLICIT OCTET STRING (SIZE (1)),
                  teleservice [3] IMPLICIT OCTET STRING (SIZE (1))} OPTIONAL,
              ... }
      RESULT
          interrogateSS-Res CHOICE {
              ss-Status  [0] IMPLICIT OCTET STRING (SIZE (1)),
              forwardedToNumber [1] IMPLICIT OCTET STRING (SIZE (1..9)),
              basicServiceGroupList [2] IMPLICIT SEQUENCE SIZE (1..13) OF
                  CHOICE {
                      bearerService [2] IMPLICIT OCTET STRING (SIZE (1)),
                      teleservice [3] IMPLICIT OCTET STRING (SIZE (1))},
              forwardingFeatureList [3] IMPLICIT SEQUENCE SIZE (1..13) OF
                  SEQUENCE {
                      basicService CHOICE {
                          bearerService [2] IMPLICIT OCTET STRING (SIZE (1)),
                          teleservice [3] IMPLICIT OCTET STRING (SIZE (1))} OPTIONAL,
                      ss-Status  [4] IMPLICIT OCTET STRING (SIZE (1)) OPTIONAL,
                      forwardedToNumber [5] IMPLICIT OCTET STRING (SIZE (1..9)) OPTIONAL,
                      forwardedToSubaddress [8] IMPLICIT OCTET STRING (SIZE (1..21)) OPTIONAL,
```

```
                    forwardingOptions [6] IMPLICIT OCTET STRING (SIZE (1)) OPTIONAL,
                    noReplyConditionTime [7] IMPLICIT INTEGER (5..30) OPTIONAL,
                    ... },
                cli-RestrictionInfo [4] IMPLICIT SEQUENCE {
                    ss-Status  OCTET STRING (SIZE (1)),
                    cliRestrictionOption ENUMERATED {
                       permanent   (0),
                       temporaryDefaultRestricted  (1),
                       temporaryDefaultAllowed  (2)} OPTIONAL,
                    ... }}
        ERRORS {
            -- systemFailure -- localValue 34,
            -- dataMissing -- localValue 35,
            -- unexpectedDataValue -- localValue 36,
            -- unknownSubscriber -- localValue 1,
            -- bearerServiceNotProvisioned -- localValue 10,
            -- teleserviceNotProvisioned -- localValue 11,
            -- callBarred -- localValue 13,
            -- illegalSS-Operation -- localValue 16,
            -- ss-NotAvailable -- localValue 18}
     ::= localValue 14

processUnstructuredSS-Data OPERATION
        ARGUMENT
            ss-UserData IA5String   (SIZE (1..200))
        RESULT
            ss-UserData IA5String   (SIZE (1..200))
        ERRORS {
            -- systemFailure -- localValue 34,
            -- unexpectedDataValue -- localValue 36}
     ::= localValue 19

processUnstructuredSS-Request OPERATION
        ARGUMENT
            ussd-Arg   SEQUENCE {
               ussd-DataCodingScheme OCTET STRING (SIZE (1)),
               ussd-String OCTET STRING (SIZE (1..160)),
               ... }
        RESULT
            ussd-Res   SEQUENCE {
               ussd-DataCodingScheme OCTET STRING (SIZE (1)),
               ussd-String OCTET STRING (SIZE (1..160)),
               ... }
        ERRORS {
            -- systemFailure -- localValue 34,
            -- dataMissing -- localValue 35,
            -- unexpectedDataValue -- localValue 36,
            -- unknownAlphabet -- localValue 71,
            -- callBarred -- localValue 13}
     ::= localValue 59

unstructuredSS-Request OPERATION
        ARGUMENT
            ussd-Arg   SEQUENCE {
               ussd-DataCodingScheme OCTET STRING (SIZE (1)),
               ussd-String OCTET STRING (SIZE (1..160)),
               ... }
        RESULT
            ussd-Res   SEQUENCE {
               ussd-DataCodingScheme OCTET STRING (SIZE (1)),
               ussd-String OCTET STRING (SIZE (1..160)),
               ... }
        ERRORS {
            -- systemFailure -- localValue 34,
            -- dataMissing -- localValue 35,
            -- unexpectedDataValue -- localValue 36,
            -- absentSubscriber -- localValue 27,
            -- illegalSubscriber -- localValue 9,
            -- illegalEquipment -- localValue 12,
            -- unknownAlphabet -- localValue 71,
            -- ussd-Busy -- localValue 72}
     ::= localValue 60

unstructuredSS-Notify OPERATION
        ARGUMENT
            ussd-Arg   SEQUENCE {
               ussd-DataCodingScheme OCTET STRING (SIZE (1)),
               ussd-String OCTET STRING (SIZE (1..160)),
               ... }
        ERRORS {
            -- systemFailure -- localValue 34,
```

```
      -- dataMissing -- localValue 35,
      -- unexpectedDataValue -- localValue 36,
      -- absentSubscriber -- localValue 27,
      -- illegalSubscriber -- localValue 9,
      -- illegalEquipment -- localValue 12,
      -- unknownAlphabet -- localValue 71,
      -- ussd-Busy -- localValue 72}
 ::= localValue 61

registerPassword OPERATION
   ARGUMENT
      ss-Code   OCTET STRING (SIZE (1))
   RESULT
      newPassword NumericString  (FROM ("0"|"1"|"2"|"3"|"4"|"5"|"6"|"7"|"8"|"9")|SIZE (4))
   ERRORS {
      -- systemFailure -- localValue 34,
      -- dataMissing -- localValue 35,
      -- unexpectedDataValue -- localValue 36,
      -- callBarred -- localValue 13,
      -- ss-SubscriptionViolation -- localValue 19,
      -- pw-RegistrationFailure -- localValue 37,
      -- negativePW-Check -- localValue 38,
      -- numberOfPW-AttemptsViolation -- localValue 43}
   LINKED {
      -- getPassword -- localValue 18}
 ::= localValue 17

getPassword OPERATION
   ARGUMENT
      guidanceInfo ENUMERATED {
         enterPW     (0),
         enterNewPW  (1),
         enterNewPW-Again  (2),
         badPW-TryAgain  (3),
         badPW-FormatTryAgain  (4)}
   RESULT
      currentPassword NumericString  (FROM ("0"|"1"|"2"|"3"|"4"|"5"|"6"|"7"|"8"|"9")|SIZE
(4))
 ::= localValue 18

beginSubscriberActivity OPERATION
   ARGUMENT
      beginSubscriberActivityArg SEQUENCE {
         imsi       OCTET STRING (SIZE (3..8)),
         originatingEntityNumber OCTET STRING (SIZE (1..9))}
 ::= localValue 54

sendRoutingInfoForSM OPERATION
   ARGUMENT
      routingInfoForSM-Arg SEQUENCE {
         msisdn    [0] IMPLICIT OCTET STRING (SIZE (1..9)),
         sm-RP-PRI [1] IMPLICIT BOOLEAN,
         serviceCentreAddress [2] IMPLICIT OCTET STRING (SIZE (1..20)),
         teleservice [5] IMPLICIT OCTET STRING (SIZE (1)) OPTIONAL,
         ... }
   RESULT
      routingInfoForSM-Res SEQUENCE {
         imsi       OCTET STRING (SIZE (3..8)),
         locationInfoWithLMSI [0] IMPLICIT SEQUENCE {
            locationInfo CHOICE {
               roamingNumber [0] IMPLICIT OCTET STRING (SIZE (1..9)),
               msc-Number [1] IMPLICIT OCTET STRING (SIZE (1..9))},
            lmsi       OCTET STRING (SIZE (4)) OPTIONAL,
            ... },
         mwd-Set    [2] IMPLICIT BOOLEAN OPTIONAL,
         ... }
   ERRORS {
      -- systemFailure -- localValue 34,
      -- dataMissing -- localValue 35,
      -- unexpectedDataValue -- localValue 36,
      -- facilityNotSupported -- localValue 21,
      -- unknownSubscriber -- localValue 1,
      -- teleserviceNotProvisioned -- localValue 11,
      -- absentSubscriber -- localValue 27,
      -- callBarred -- localValue 13}
 ::= localValue 45

forwardSM  OPERATION
   ARGUMENT
      forwardSM-Arg SEQUENCE {
         sm-RP-DA   CHOICE {
```

```
            imsi       [0] IMPLICIT OCTET STRING (SIZE (3..8)),
            lmsi       [1] IMPLICIT OCTET STRING (SIZE (4)),
            roamingNumber [3] IMPLICIT OCTET STRING (SIZE (1..9)),
            serviceCentreAddressDA [4] IMPLICIT OCTET STRING (SIZE (1..20)),
            noSM-RP-DA [5] IMPLICIT NULL},
        sm-RP-OA   CHOICE {
            msisdn     [2] IMPLICIT OCTET STRING (SIZE (1..9)),
            serviceCentreAddressOA [4] IMPLICIT OCTET STRING (SIZE (1..20)),
            noSM-RP-OA [5] IMPLICIT NULL},
        sm-RP-UI   OCTET STRING (SIZE (1..200)),
        moreMessagesToSend NULL OPTIONAL,
        ... }
    ERRORS {
        -- systemFailure -- localValue 34,
        -- dataMissing -- localValue 35,
        -- unexpectedDataValue -- localValue 36,
        -- facilityNotSupported -- localValue 21,
        -- unidentifiedSubscriber -- localValue 5,
        -- illegalSubscriber -- localValue 9,
        -- illegalEquipment -- localValue 12,
        -- absentSubscriber -- localValue 27,
        -- subscriberBusyForMT-SMS -- localValue 31,
        -- sm-DeliveryFailure -- localValue 32}
 ::= localValue 46

reportSM-DeliveryStatus OPERATION
    ARGUMENT
        reportSM-DeliveryStatusArg SEQUENCE {
            msisdn     OCTET STRING (SIZE (1..9)),
            serviceCentreAddress OCTET STRING (SIZE (1..20)),
            sm-DeliveryOutcome ENUMERATED {
                memoryCapacityExceeded  (0),
                absentSubscriber  (1),
                successfulTransfer  (2)} OPTIONAL,
            ... }
    RESULT
        storedMSISDN OCTET STRING (SIZE (1..9))
    ERRORS {
        -- dataMissing -- localValue 35,
        -- unexpectedDataValue -- localValue 36,
        -- unknownSubscriber -- localValue 1,
        -- messageWaitingListFull -- localValue 33}
 ::= localValue 47

noteSubscriberPresent OPERATION
    ARGUMENT
        imsi        OCTET STRING (SIZE (3..8))
 ::= localValue 48

alertServiceCentreWithoutResult OPERATION
    ARGUMENT
        alertServiceCentreArg SEQUENCE {
            msisdn     OCTET STRING (SIZE (1..9)),
            serviceCentreAddress OCTET STRING (SIZE (1..20)),
            ... }
 ::= localValue 49

informServiceCentre OPERATION
    ARGUMENT
        informServiceCentreArg SEQUENCE {
            storedMSISDN OCTET STRING (SIZE (1..9)) OPTIONAL,
            mw-Status  BIT STRING {
                sc-AddressNotIncluded  (0),
                mnrf-Set     (1),
                mcef-Set     (2)} (SIZE (6)) OPTIONAL,
            ... }
 ::= localValue 63

alertServiceCentre OPERATION
    ARGUMENT
        alertServiceCentreArg SEQUENCE {
            msisdn     OCTET STRING (SIZE (1..9)),
            serviceCentreAddress OCTET STRING (SIZE (1..20)),
            ... }
    ERRORS {
        -- systemFailure -- localValue 34,
        -- dataMissing -- localValue 35,
        -- unexpectedDataValue -- localValue 36}
 ::= localValue 64

readyForSM OPERATION
```

```
  ARGUMENT
     readyForSM-Arg SEQUENCE {
        imsi      [0] IMPLICIT OCTET STRING (SIZE (3..8)),
        alertReason ENUMERATED {
           ms-Present  (0),
           memoryAvailable  (1)},
        ... }
  ERRORS {
     -- dataMissing -- localValue 35,
     -- unexpectedDataValue -- localValue 36,
     -- facilityNotSupported -- localValue 21,
     -- unknownSubscriber -- localValue 1}
 ::= localValue 66

systemFailure ERROR
  PARAMETER
     networkResource ENUMERATED {
        plmn       (0),
        hlr        (1),
        vlr        (2),
        pvlr       (3),
        controllingMSC  (4),
        vmsc       (5),
        eir        (6),
        rss        (7)}
 ::= localValue 34

dataMissing ERROR
 ::= localValue 35

unexpectedDataValue ERROR
 ::= localValue 36

facilityNotSupported ERROR
 ::= localValue 21

unknownSubscriber ERROR
 ::= localValue 1

numberChanged ERROR
 ::= localValue 44

unknownBaseStation ERROR
 ::= localValue 2

unknownMSC ERROR
 ::= localValue 3

unidentifiedSubscriber ERROR
 ::= localValue 5

unknownEquipment ERROR
 ::= localValue 7

roamingNotAllowed ERROR
  PARAMETER
     roamingNotAllowedCause ENUMERATED {
        plmnRoamingNotAllowed  (0),
        operatorDeterminedBarring  (3)}
 ::= localValue 8

illegalSubscriber ERROR
 ::= localValue 9

illegalEquipment ERROR
 ::= localValue 12

bearerServiceNotProvisioned ERROR
 ::= localValue 10

teleserviceNotProvisioned ERROR
 ::= localValue 11

invalidTargetBaseStation ERROR
 ::= localValue 23

noRadioResourceAvailable ERROR
 ::= localValue 24

noHandoverNumberAvailable ERROR
 ::= localValue 25
```

```
subsequentHandoverFailure ERROR
 ::= localValue 26

tracingBufferFull ERROR
 ::= localValue 40

noRoamingNumberAvailable ERROR
 ::= localValue 39

absentSubscriber ERROR
    PARAMETER
       mwd-Set    BOOLEAN
 ::= localValue 27

callBarred ERROR
    PARAMETER
       callBarringCause ENUMERATED {
          barringServiceActive  (0),
          operatorBarring  (1)}
 ::= localValue 13

forwardingViolation ERROR
 ::= localValue 14

cug-Reject ERROR
    PARAMETER
       cug-RejectCause ENUMERATED {
          incomingCallsBarredWithinCUG  (0),
          subscriberNotMemberOfCUG  (1),
          requestedBasicServiceViolatesCUG-Constraints  (5),
          calledPartySS-InteractionViolation  (7)}
 ::= localValue 15

illegalSS-Operation ERROR
 ::= localValue 16

ss-ErrorStatus ERROR
    PARAMETER
       ss-Status  OCTET STRING (SIZE (1))
 ::= localValue 17

ss-NotAvailable ERROR
 ::= localValue 18

ss-SubscriptionViolation ERROR
    PARAMETER
       ss-SubscriptionOption CHOICE {
          cliRestrictionOption [2] IMPLICIT ENUMERATED {
             permanent  (0),
             temporaryDefaultRestricted  (1),
             temporaryDefaultAllowed  (2)},
          overrideCategory [1] IMPLICIT ENUMERATED {
             overrideEnabled  (0),
             overrideDisabled  (1)}}
 ::= localValue 19

ss-Incompatibility ERROR
    PARAMETER
       ss-IncompatibilityCause SEQUENCE {
          ss-Code    [1] IMPLICIT OCTET STRING (SIZE (1)) OPTIONAL,
          basicService CHOICE {
             bearerService [2] IMPLICIT OCTET STRING (SIZE (1)),
             teleservice [3] IMPLICIT OCTET STRING (SIZE (1))} OPTIONAL,
          ss-Status [4] IMPLICIT OCTET STRING (SIZE (1)) OPTIONAL,
          ... }
 ::= localValue 20

unknownAlphabet ERROR
 ::= localValue 71

ussd-Busy  ERROR
 ::= localValue 72

pw-RegistrationFailure ERROR
    PARAMETER
       pw-RegistrationFailureCause ENUMERATED {
          undetermined  (0),
          invalidFormat  (1),
          newPasswordsMismatch  (2)}
 ::= localValue 37
```

```
negativePW-Check ERROR
 ::= localValue 38

numberOfPW-AttemptsViolation ERROR
 ::= localValue 43

subscriberBusyForMT-SMS ERROR
 ::= localValue 31

sm-DeliveryFailure ERROR
   PARAMETER
      sm-DeliveryFailureCause CHOICE {
          sm-DeliveryFailureCauseWithDiagnostic SEQUENCE {
              sm-EnumeratedDeliveryFailureCause ENUMERATED {
                 memoryCapacityExceeded  (0),
                 equipmentProtocolError  (1),
                 equipmentNotSM-Equipped  (2),
                 unknownServiceCentre  (3),
                 sc-Congestion  (4),
                 invalidSME-Address  (5),
                 subscriberNotSC-Subscriber  (6)},
              diagnosticInfo OCTET STRING (SIZE (1..200)) OPTIONAL,
              ... },
          sm-EnumeratedDeliveryFailureCause ENUMERATED {
             memoryCapacityExceeded  (0),
             equipmentProtocolError  (1),
             equipmentNotSM-Equipped  (2),
             unknownServiceCentre  (3),
             sc-Congestion  (4),
             invalidSME-Address  (5),
             subscriberNotSC-Subscriber  (6)}}
 ::= localValue 32

messageWaitingListFull ERROR
 ::= localValue 33

END
```

B.2   Fully Expanded ASN.1 Source of MAP-DialogueInformation

```
--     Expanded ASN.1 Module 'MAP-DialogueInformation'
--SIEMENS ASN.1 Compiler      P2.00 (94-01-31 14:46:00)
--             Date: 94-02-15 Time: 12:22:04



MAP-DialogueInformation { ccitt identified-organization(4) etsi(0) mobileDomain(0)
gsm-Network(1) modules(3) map-DialogueInformation(3) version2(2) }

DEFINITIONS

::=

BEGIN

EXPORTS
   map-DialogueAS,
   MAP-DialoguePDU
;


map-DialogueAS OBJECT IDENTIFIER ::= { ccitt(0) identified-organization(4)
etsi(0) mobileDomain(0) gsm-Network(1) 1 map-DialoguePDU(1) version1(1) }

MAP-DialoguePDU ::= CHOICE {
   map-open   [0] IMPLICIT SEQUENCE {
      destinationReference [0] IMPLICIT OCTET STRING (SIZE (1..20)) OPTIONAL,
      originationReference [1] IMPLICIT OCTET STRING (SIZE (1..20)) OPTIONAL,
      ... },
   map-accept [1] IMPLICIT SEQUENCE {
      ... },
   map-close  [2] IMPLICIT SEQUENCE {
      ... },
   map-refuse [3] IMPLICIT SEQUENCE {
      reason     ENUMERATED {
         noReasonGiven  (0),
         invalidDestinationReference  (1),
         invalidOriginatingReference  (2)},
      ... },
   map-userAbort [4] IMPLICIT SEQUENCE {
      map-UserAbortChoice CHOICE {
         userSpecificReason [0] IMPLICIT NULL,
         userResourceLimitation [1] IMPLICIT NULL,
         resourceUnavailable [2] IMPLICIT ENUMERATED {
            shortTermResourceLimitation  (0),
            longTermResourceLimitation  (1)},
         applicationProcedureCancellation [3] IMPLICIT ENUMERATED {
            handoverCancellation  (0),
            radioChannelRelease  (1),
            networkPathRelease  (2),
            callRelease  (3),
            associatedProcedureFailure  (4),
            tandemDialogueRelease  (5),
            remoteOperationsFailure  (6)}},
      ... },
   map-providerAbort [5] IMPLICIT SEQUENCE {
      map-ProviderAbortReason ENUMERATED {
         abnormalDialogue  (0),
         invalidPDU  (1)},
      ... }}

MAP-OpenInfo ::= SEQUENCE {
   destinationReference [0] IMPLICIT OCTET STRING (SIZE (1..20)) OPTIONAL,
   originationReference [1] IMPLICIT OCTET STRING (SIZE (1..20)) OPTIONAL,
   ... }

MAP-AcceptInfo ::= SEQUENCE {
   ... }

MAP-CloseInfo ::= SEQUENCE {
   ... }

MAP-RefuseInfo ::= SEQUENCE {
   reason     ENUMERATED {
      noReasonGiven  (0),
      invalidDestinationReference  (1),
      invalidOriginatingReference  (2)},
   ... }
```

```
Reason      ::= ENUMERATED {
   noReasonGiven  (0),
   invalidDestinationReference  (1),
   invalidOriginatingReference  (2)}

MAP-UserAbortInfo ::= SEQUENCE {
   map-UserAbortChoice CHOICE {
      userSpecificReason [0] IMPLICIT NULL,
      userResourceLimitation [1] IMPLICIT NULL,
      resourceUnavailable [2] IMPLICIT ENUMERATED {
         shortTermResourceLimitation  (0),
         longTermResourceLimitation  (1)},
      applicationProcedureCancellation [3] IMPLICIT ENUMERATED {
         handoverCancellation  (0),
         radioChannelRelease  (1),
         networkPathRelease  (2),
         callRelease  (3),
         associatedProcedureFailure  (4),
         tandemDialogueRelease  (5),
         remoteOperationsFailure  (6)}},
   ... }

MAP-UserAbortChoice ::= CHOICE {
   userSpecificReason [0] IMPLICIT NULL,
   userResourceLimitation [1] IMPLICIT NULL,
   resourceUnavailable [2] IMPLICIT ENUMERATED {
      shortTermResourceLimitation  (0),
      longTermResourceLimitation  (1)},
   applicationProcedureCancellation [3] IMPLICIT ENUMERATED {
      handoverCancellation  (0),
      radioChannelRelease  (1),
      networkPathRelease  (2),
      callRelease  (3),
      associatedProcedureFailure  (4),
      tandemDialogueRelease  (5),
      remoteOperationsFailure  (6)}}

ResourceUnavailableReason ::= ENUMERATED {
   shortTermResourceLimitation  (0),
   longTermResourceLimitation  (1)}

ProcedureCancellationReason ::= ENUMERATED {
   handoverCancellation  (0),
   radioChannelRelease  (1),
   networkPathRelease  (2),
   callRelease  (3),
   associatedProcedureFailure  (4),
   tandemDialogueRelease  (5),
   remoteOperationsFailure  (6)}

MAP-ProviderAbortInfo ::= SEQUENCE {
   map-ProviderAbortReason ENUMERATED {
      abnormalDialogue  (0),
      invalidPDU  (1)},
   ... }

MAP-ProviderAbortReason ::= ENUMERATED {
   abnormalDialogue  (0),
   invalidPDU  (1)}

END
```

Annex C (informative): Formal protocol incompatibilities between versions 1 & 2 of MAP

C.1 Introduction

Annex C is not normative; it presents for information those parts of the MAP version 2 protocol which are not backward compatible with (i.e. a true superset of) the MAP version 1 protocol. For each incompatibility there is a commentary on the impact on the interworking of MAP version 1 and MAP version 2 entities.

SMG have decided that the MAP specification should include the operations and procedures used on the B interface (MSC/VLR) only for modelling purposes; the B interface cannot be implemented as an open interface. Hence any incompatibilities which affect operations used only on the B interface have no impact on the interworking of MAP version 1 and MAP version 2 entities.

C.2 Deletion of operations and errors

This subclause lists the operations and errors which have been completely removed from the MAP protocol.

C.2.1 Deletion of operation DeregisterMobileSubscriber

Although it is defined in the protocol in the MAP version 1 specification, this operation is not used (see subclause 1.2 of the latest phase 1 version of GSM 09.02).

C.2.2 Deletion of operation RegisterChargingInfo

There is no known implementation of MAP version 1 which supports this operation. The deletion has been approved by SMG.

C.2.3 Deletion of operation ForwardSS-Notification

There is no known implementation of MAP version 1 which supports this operation. The deletion has been approved by SMG.

C.2.4 Deletion of operations used only on the B-interface

The following operations (listed in alphabetical order) are not defined in the MAP version 2 protocol, because they are used only on the B-interface:

AllocateHandoverNumber; AttachIMSI; Authenticate; CompleteCall; DetachIMSI; ForwardNewTMSI; InvokeSS; Page; ProcessAccessRequest; ProvideIMSI; SearchForMobileSubscriber; SendHandoverReport; SendInfoForIncomingCall; SendInfoForOutgoingCall; SetCipheringMode; UpdateLocationArea.

C.2.5 Deletion of error InsufficientBearerCapabilities

This error is defined in the MAP version 1 protocol, but it is not specified for use with any operation.

C.3 Deletion of errors for operations

This subclause lists the cases where an error which is specified for use with an operation in the MAP version 1 specification is not specified for use with the same operation in the MAP version 2 specification.

C.3.1 Error NegativePW-Check for operation RegisterSS

Password checking is not used for the supplementary services to which registration applies.

C.3.2 Error NegativePW-Check for operation EraseSS

Password checking is not used for the supplementary services to which erasure applies.

C.3.3 Error NegativePW-Check for operation InterrogateSS

Password checking is not used for the interrogation of supplementary services.

C.3.4 Error CUG-Reject for operation SendRoutingInfoForSM

Closed User Group does not apply to the short message service.

C.4 Changes to definitions of data types

This subclause lists in alphabetical order those data types whose definitions have been directly or indirectly changed. For constructed data types, only the components which have been changed are mentioned. The commentary on the end effect of each change is given in subclauses C.6 (parameters of operations), C.7 (results of operations) or C.8 (errors of operations).

C.4.1 CUG-Feature

The type CUG-Feature was a sequence of four components; these have been deleted and replaced by three new components. It is used for the components of the type CUG-FeatureList.

C.4.2 CUG-FeatureList

The type CUG-FeatureList is a sequence of components of type CUG-Feature. It is used for a component of the type CUG-Info.

C.4.3 CUG-Info

The type CUG-Info is a sequence. One component of the sequence has been replaced by a component of a new type; the other component was a choice between a cug-Feature and a cug-FeatureList, and is now an optional cug-FeatureList. The type CUG-Info is used for a component of the type SS-Info.

C.4.4 CUG-RejectCause

The range of permitted values of the enumerated type CUG-RejectCause has been extended. The type is used for the parameter of the error CUG-Reject.

C.4.5 IMSI

The lower limit of the length of an IMSI has been increased from 2 octets to 3 octets. It is not possible to code a minimum length IMSI (MCC+MNC+MSIN) in 2 octets, so the theoretical lower limit of 2 octets should never be used by a MAP version 1 entity to send an IMSI; if it is, a MAP version 2 entity will treat it as a protocol error. Since this change has no practical impact it is not discussed further.

C.4.6 ISDN-AddressString

The upper limit of the length of an ISDN-AddressString has been reduced from 10 octets to 9 octets. The maximum length of an E.164 number is 15 digits; this can be encoded as a TBCD-string in 8 octets, plus a further octet to hold the type of number and number plan indicator. The cases where the ISDN-AddressString type was used in MAP version 1 to carry anything other than an E.164 number are described in subclause C.6; the other cases are not discussed further.

C.4.7 Password

In MAP version 1 the type Password was a choice between a printable string of length 4 to 8 octets or a numeric string of length 4 octets. It is now a numeric string of length 4 octets. The type Password is used for the result of the operation GetPassword.

C.4.8 RequestParameter

The enumerated type RequestParameter is no longer allowed to take the value requestCUG-Info. It is used as a component of the type RequestParameterList.

C.4.9 RequestParameterList

The type RequestParameterList is a sequence of components of type RequestParameter. The parameter of the operation SendParameters is a sequence of which one component is of type RequestParameterList.

C.4.10 SentParameter

The type SentParameter is a choice of which one component is of type SubscriberData. It is used as a component of the type SentParameterList.

C.4.11 SentParameterList

The type SentParameterList is a sequence whose components are of type SentParameter. The maximum number of components in the sequence has been reduced from 10 to 6.

The type SentParameterList is used for the result of the operation SendParameters.

C.4.12 SS-Data

The type SS-Data is a sequence of which one component is of type SS-SubscriptionOption. It is used for a component of the type SS-Info.

C.4.13 SS-Info

The type SS-Info is a choice of which one component is of type CUG-Info and another component is of type SS-Data. It is used for the result of the operations RegisterSS, EraseSS, ActivateSS and DeactivateSS, and for components of the type SS-InfoList.

C.4.14 SS-InfoList

The type SS-InfoList is a sequence of components of type SS-Info. It is used for a component of the type SubscriberData.

C.4.15 SS-SubscriptionOption

The type SS-SubscriptionOption was a choice from five components: perCallBasis (used for the CLIR supplementary service); notificationToHeldRetrievedParty (used for the Call Transfer supplementary service); userToUserServiceIndicator (used for the User to User Signalling supplementary service); maximumConfereesNumber (used for the Conference Calling supplementary service); and huntGroupAccessSelectionOrder (used for the Mobile Access Hunting supplementary service. It has been replaced by a choice from two components: cliRestrictionOption (used for the CLIR supplementary service); and overrideCategory (used for the CLIP and COLP supplementary services).

The Call Transfer, User to User Signalling, Conference Calling and Mobile Access Hunting supplementary services are not specified for GSM Phase 1 or GSM Phase 2, so data for these services should not be transferred in a dialogue involving a MAP version 1 entity. These cases will therefore not be discussed further.

The type SS-SubscriptionOption is used for a component of SS-Data and for the parameter of the error SS-SubscriptionViolation.

C.4.16   SubscriberData

The type SubscriberData is a sequence of which one component is of type SS-InfoList. Components of SubscriberData are used as a component of the parameter of the operation InsertSubscriberData; the type is also used for a component of the type SentParameter.

C.5   Changes to parameters of errors

This subclause lists in alphabetical order the errors whose parameters have changed.

C.5.1   CUG-Reject

The error CUG-Reject has an optional parameter of type CUG-RejectCause. The error CUG-Reject is used for the operation SendRoutingInfo.

C.5.2   SS-SubscriptionViolation

The error SS-SubscriptionViolation has an optional parameter of type SS-SubscriptionOption. The error SS-SubscriptionViolation is used for the operations ActivateSS, DeactivateSS, EraseSS and RegisterSS.

C.6   Changes to parameters of operations

This subclause lists in alphabetical order the operations whose parameters have changed, and gives a commentary on the effect of the changes on each operation.

C.6.1   InsertSubscriberData

The parameter of the operation InsertSubscriberData is a sequence of which one component is a sequence of components of SubscriberData. The components of SubscriberData which are affected by the changes listed in subclause C.4 are cug-Info and ss-SubscriptionOption.

The CUG supplementary service is not supported by MAP version 1; CUG-Info should therefore not be used as a component of SubscriberData in a dialogue involving a MAP version 1 entity.

The replacement of the perCallBasis (type BOOLEAN) subscription option by the cliRestrictionOption (type ENUMERATED) for the CLIR supplementary service means that full support for the CLIR supplementary service is not possible if either entity involved can support only MAP version 1.

C.6.2   RegisterSS

The forwardedToNumber component of the parameter of the operation RegisterSS had a maximum length of 10 octets in MAP version 1, as it was of the type ISDN-AddressString. In MAP version 2 the maximum length is 20 octets, as the type is AddressString. The maximum length (9 octets) of the ISDN-AddressString type in MAP version 2 may not be adequate to hold the forwardedToNumber, which is not necessarily an E.164 number; the user may enter the number using the digits for international access rather than the "+" key.

C.6.3 SendParameters

The operation SendParameters uses as its parameter a sequence of which one component is of type RequestParameter. The value requestCUG-Info can no longer be used for this component. The SendParameters operation is used only when interworking with a MAP version 1 entity, and MAP version 1 does not support the GSM Phase 2 CUG supplementary service, so the SendParameters operation should in any case not be used to request CUG information.

C.6.4 SendRoutingInfoForSM

The cug-Interlock component of the parameter of the operation SendRoutingInfoForSM has been deleted. Closed User Group does not apply to the short message service.

C.7 Changes to results of operations

This subclause lists in alphabetical order the operations whose results have changed, and gives a commentary on the effect of the changes on each operation.

C.7.1 ActivateSS

The result of the operation ActivateSS is of type SS-Info. Two data types used for components of SS-Info have suffered incompatible changes: CUG-Info and SS-SubscriptionOption.

The ActivateSS operation does not apply to the CUG supplementary service, so the cug-Info component of SS-Info should never be present in the result of the operation ActivateSS.

The ActivateSS operation does not apply to the CLIP, CLIR or COLP supplementary services, for which the type SS-SubscriptionOption is used, so the ss-SubscriptionOption component of SS-Info should never be present in the result of the operation ActivateSS.

C.7.2 DeactivateSS

The result of the operation DeactivateSS is of type SS-Info. Two data types used for components of SS-Info have suffered incompatible changes: CUG-Info and SS-SubscriptionOption.

The DeactivateSS operation does not apply to the CUG supplementary service, so the cug-Info component of SS-Info should never be present in the result of the operation DeactivateSS.

The DeactivateSS operation does not apply to the CLIP, CLIR or COLP supplementary services, for which the type SS-SubscriptionOption is used, so the ss-SubscriptionOption component of SS-Info should never be present in the result of the operation DeactivateSS.

C.7.3 EraseSS

The result of the operation EraseSS is of type SS-Info. Two data types used for components of SS-Info have suffered incompatible changes: CUG-Info and SS-SubscriptionOption.

The EraseSS operation does not apply to the CUG supplementary service, so the cug-Info component of SS-Info should never be present in the result of the operation EraseSS.

The EraseSS operation does not apply to the CLIP, CLIR or COLP supplementary services, for which the type SS-SubscriptionOption is used, so the ss-SubscriptionOption component of SS-Info should never be present in the result of the operation EraseSS.

C.7.4   GetPassword

The result of the operation GetPassword is of type Password. In MAP version 1 this was a choice between a printable string of length 4 to 8 octets or a numeric string of length 4 octets. It is now a numeric string of length 4 octets. The printable string option was never used in MAP version 1, as indicated by a comment in the ASN.1 in the latest phase 1 version of GSM 09.02.

C.7.5   InterrogateSS

The result of the InterrogateSS operation is a CHOICE; one of the components of the CHOICE is a list of basic services to which the supplementary service applies, which is used for the Call Barring supplementary service. In MAP version 1 this list can in principle have up to 70 members, the number of individual basic services. However Call Barring can apply to only 13 basic service groups. In MAP version 2 the length of the list of basic service codes which can be returned in the result of the InterrogateSS operation is reduced to 13 to reflect this.

C.7.6   RegisterSS

The result of the operation RegisterSS is of type SS-Info. Two data types used for components of SS-Info have suffered incompatible changes: CUG-Info and SS-SubscriptionOption.

The RegisterSS operation does not apply to the CUG supplementary service, so the cug-Info component of SS-Info should never be present in the result of the operation RegisterSS.

The RegisterSS operation does not apply to the CLIP, CLIR or COLP supplementary services, for which the type SS-SubscriptionOption is used, so the ss-SubscriptionOption component of SS-Info should never be present in the result of the operation RegisterSS.

C.7.7   SendParameters

The result of the operation SendParameters is of type SentParameterList, which is a sequence of components of type SentParameter. The maximum number of components in the sequence has been reduced from 10 to 6. MAP version 1 could in principle send 10 sets of CUG-Information, but the supplementary service Closed User Group is not defined for GSM Phase 1, and the MAP version 1 signalling protocol will not support Closed User Group as defined for GSM Phase 2, so a MAP version 1 entity should never request parameters for CUG. The maximum number of sent parameters therefore consists of an IMSI and 5 AuthenticationSets - a total of 6.

The type SentParameter is a choice of which one component is of type SubscriberData. The components of SubscriberData which are affected by the changes listed in subclause C.4 are CUG-Info and ss-SubscriptionOption.

The CUG supplementary service is not supported by MAP version 1; CUG-Info should therefore not be used as a component of SubscriberData in a dialogue involving a MAP version 1 entity.

The replacement of the perCallBasis (type BOOLEAN) subscription option by the cliRestrictionOption (type ENUMERATED) for the CLIR supplementary service means that full support for the CLIR supplementary service is not possible if either entity involved can support only MAP version 1.

C.7.8   SendRoutingInfoForSM

The result of the operation SendRoutingInfoForSM is a sequence of which one component was a choice between location information (optionally with an associated LMSI) and forwarding data; the choice of forwarding data has been removed. Call Forwarding does not apply to the short message service.

C.8 Changes to errors of operations

This subclause lists in alphabetical order the operations whose errors have changed, and gives a commentary on the effect of the changes on each operation.

C.8.1 ActivateSS

The definition of the type SS-SubscriptionOption used for the optional parameter of the error SS-SubscriptionViolation has been changed. However the only use defined for the error SS-SubscriptionViolation is when the user attempts to activate or deactivate a Call Barring supplementary service and the subscription option "Control by Service Provider" has been taken. The MAP version 1 protocol does not define this subscription option, so there is no case when the error SS-SubscriptionViolation will be used with the optional parameter.

C.8.2 DeactivateSS

The definition of the type SS-SubscriptionOption used for the optional parameter of the error SS-SubscriptionViolation has been changed. However the only use defined for the error SS-SubscriptionViolation is when the user attempts to activate or deactivate a Call Barring supplementary service and the subscription option "Control by Service Provider" has been taken. The MAP version 1 protocol does not define this subscription option, so there is no case when the error SS-SubscriptionViolation will be used with the optional parameter.

C.8.3 EraseSS

The definition of the type SS-SubscriptionOption used for the optional parameter of the error SS-SubscriptionViolation has been changed. However the only use defined for the error SS-SubscriptionViolation is when the user attempts to activate or deactivate a Call Barring supplementary service and the subscription option "Control by Service Provider" has been taken, so there is no case when the error SS-SubscriptionViolation will be used for the operation EraseSS.

C.8.4 RegisterSS

The definition of the type SS-SubscriptionOption used for the optional parameter of the error SS-SubscriptionViolation has been changed. However the only use defined for the error SS-SubscriptionViolation is when the user attempts to activate or deactivate a Call Barring supplementary service and the subscription option "Control by Service Provider" has been taken, so there is no case when the error SS-SubscriptionViolation will be used for the operation RegisterSS.

C.8.5 SendRoutingInfo

The definition of the type (CUG-RejectCause) used for the optional parameter of the error CUG-Reject has been changed. However the supplementary service Closed User Group is not defined for GSM Phase 1, and the MAP version 1 signalling protocol will not support Closed User Group as defined for GSM Phase 2, so the error CUG-Reject should not be used in a dialogue involving a MAP version 1 entity.

History

Document history		
August 1996	Publication of GSM 09.02 version 5.3.0	
November 1996	Unified Approval procedure	UAP 59: 1996-11-25 to 1997-03-21