

ETSI TS 129 002 V3.3.0 (2000-01)

Technical Specification

**Digital cellular telecommunications system (Phase 2+) (GSM);
Universal Mobile Telecommunications System (UMTS);
Mobile Application Part (MAP) specification
(3G TS 29.002 version 3.3.0 Release 1999)**



Reference

DTS/TSGN-0229002U

Keywords

GSM, UMTS

ETSI

Postal address

F-06921 Sophia Antipolis Cedex - FRANCE

Office address

650 Route des Lucioles - Sophia Antipolis
Valbonne - FRANCE
Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16
Siret N° 348 623 562 00017 - NAF 742 C
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° 7803/88

Internet

secretariat@etsi.fr
Individual copies of this ETSI deliverable
can be downloaded from
<http://www.etsi.org>
If you find errors in the present document, send your
comment to: editor@etsi.fr

Important notice

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

Copyright Notification

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

© European Telecommunications Standards Institute 2000.
All rights reserved.

Intellectual Property Rights

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

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

Foreword

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

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

For 3GPP documents:

3G TS | TR nn.nnn "<title>" (with or without the prefix 3G)

is equivalent to

ETSI TS | TR 1nn nnn "[Digital cellular telecommunications system (Phase 2+) (GSM);] Universal Mobile Telecommunications System; <title>

For GSM document identities of type "GSM xx.yy", e.g. GSM 01.04, the corresponding ETSI document identity may be found in the Cross Reference List on www.etsi.org/key

Contents

Foreword.....	25
1 Scope.....	27
2 References	27
3 Abbreviations.....	33
4 Configuration of the mobile network	33
4.1 The entities of the mobile system.....	33
4.1.1 The Home Location Register (HLR).....	33
4.1.2 The Visitor Location Register (VLR).....	34
4.1.3 The Mobile-services Switching Centre (MSC)	34
4.1.4 The Base Station System (BSS).....	34
4.1.5 The Gateway MSC (GMSC).....	35
4.1.6 The SMS Gateway MSC	35
4.1.7 The SMS Interworking MSC	35
4.1.8 The VBS/VGCS Anchor MSC.....	35
4.1.9 The Equipment Identity Register (EIR).....	35
4.1.10 The GSM Service Control Function (gsmSCF).....	35
4.1.11 The VBS/VGCS Relay MSC	35
4.1.12 The Group Call Register (GCR).....	35
4.1.13 The Shared InterWorking Function Server (SIWFS)	36
4.1.14 The Serving GPRS Support Node (SGSN)	36
4.1.15 The Gateway GPRS Support Node (GGSN).....	36
4.1.17 The Serving Mobile Location Center (SMLC).....	37
4.1.18 The Gateway Mobile Location Center (GMLC)	37
4.1.19 The Location Measurement Unit (LMU)	37
4.3 Interconnection between PLMNs.....	39
4.4 The interfaces within the mobile service.....	39
4.4.1 Interface between the HLR and the VLR (D-interface).....	39
4.4.2 Interface between the HLR and the gsmSCF (J-interface)	39
4.4.3 Interface between the VLR and its associated MSC(s) (B-interface)	39
4.4.4 Interface between VLRs (G-interface)	39
4.4.5 Interface between the HLR and the MSC (C-interface)	39
4.4.6 Interface between the MSC and the gsmSCF (L-interface).....	39
4.4.6A Interface between the VLR and the gsmSCF (M-interface)	40
4.4.7 Interface between MSCs (E-interface)	40
4.4.8 Interface between the MSC and Base Station Systems (A-interface)	40
4.4.9 Interface between MSC and EIR (F-interface).....	40
4.4.10 Interface between VBS/VGCS Anchor MSC and GCR (I-interface).....	40
4.4.11 Interface between the MSC and the SIWF server (K-interface).....	40
4.4.12 Interface between SGSN and HLR (Gr-interface).....	40
4.4.13 Interface between SGSN and SMS-GMSC or SMS-IWMSC (Gd-interface).....	40
4.4.14 Interface between GGSN and HLR (Gc-interface).....	40
4.4.15 Interface between SGSN and EIR (Gf-interface)	41
4.4.16 Interface between SGSN and BSC (Gb-interface)	41
4.4.17 Interface between SGSN and MSC/VLR (Gs-interface)	41
4.4.18 Interface between SMLC and MSC (Ls interface)	41
4.4.19 Interface between SMLC and VLR (Lv interface)	41
4.4.20 Interface between GMLC and HLR (Lh interface).....	41
4.4.21 Interface between GMLC and MSC (Lg interface).....	41
4.4.22 Interface between LCS Client and GMLC (Le interface).....	41
4.4.23 Interface between the gsmSCF and the GMLC (Lc-interface)	41
4.5 Splitting of the data storage	41
5 Overload and compatibility overview	42
5.1 Overload control	42

5.1.1	Overload control for MSC (outside MAP).....	42
5.1.2	Overload control for MAP entities.....	42
5.1.3	Congestion control for Signalling System No. 7.....	45
5.2	Compatibility.....	46
5.2.1	General.....	46
5.2.2	Strategy for selecting the Application Context (AC) version.....	46
5.2.2.1	Proposed method.....	46
5.2.2.2	Managing the version look-up table.....	47
5.2.2.3	Optimizing the method.....	48
6	Requirements concerning the use of SCCP and TC.....	48
6.1	Use of SCCP.....	48
6.1.1	SCCP Class.....	48
6.1.2	Sub-System Number (SSN).....	48
6.1.3	SCCP addressing.....	49
6.1.3.1	Introduction.....	49
6.1.3.2	The Mobile-services Switching Centre (MSC).....	51
6.1.3.2.1	MSC interaction during handover.....	51
6.1.3.2.2	MSC for short message routing.....	51
6.1.3.2.3	MSC for location request routing.....	51
6.1.3.2.4	MSC for LMU Control.....	51
6.1.3.3	The Home Location Register (HLR).....	51
6.1.3.3.1	During call set-up.....	51
6.1.3.3.2	Before location updating completion.....	52
6.1.3.3.3	After location updating completion.....	52
6.1.3.3.4	VLR restoration.....	52
6.1.3.3.5	During Network-Requested PDP Context Activation.....	53
6.1.3.3.6	Before GPRS location updating completion.....	53
6.1.3.3.7	After GPRS location updating completion.....	54
6.1.3.3.8	Query for a Location Request.....	54
6.1.3.4	The Visitor Location Register (VLR).....	54
6.1.3.4.1	Inter-VLR information retrieval.....	54
6.1.3.4.2	HLR request.....	54
6.1.3.5	The Interworking MSC (IWMSC) for Short Message Service.....	54
6.1.3.6	The Equipment Identity Register (EIR).....	54
6.1.3.7	The Shared Inter Working Function (SIWF).....	54
6.1.3.8	The Serving GPRS Support Node (SGSN).....	55
6.1.3.9	The Gateway GPRS Support Node (GGSN).....	55
6.1.3.10	The Gateway MSC (GMSC) for Short Message Service.....	55
6.1.3.10A	Void.....	55
6.1.3.10A.1	Void.....	55
6.1.3.10A.2	Void.....	55
6.1.3.10B	The Gateway Mobile Location Center (GMLC).....	55
6.1.3.11	Summary table.....	55
6.2	Use of TC.....	57
7	General on MAP services.....	58
7.1	Terminology and definitions.....	58
7.2	Modelling principles.....	59
7.3	Common MAP services.....	59
7.3.1	MAP-OPEN service.....	60
7.3.2	MAP-CLOSE service.....	63
7.3.3	MAP-DELIMITER service.....	63
7.3.4	MAP-U-ABORT service.....	63
7.3.5	MAP-P-ABORT service.....	64
7.3.6	MAP-NOTICE service.....	65
7.4	Sequencing of services.....	66
7.5	General rules for mapping of services onto TC.....	67
7.5.1	Mapping of common services.....	67
7.5.2	Mapping of user specific services.....	69
7.6	Definition of parameters.....	69

7.6.1	Common parameters.....	71
7.6.1.1	Invoke Id.....	71
7.6.1.2	Linked Id	71
7.6.1.3	Provider error	71
7.6.1.4	User error.....	72
7.6.2	Numbering and identification parameter	75
7.6.2.1	IMSI	75
7.6.2.2	TMSI	75
7.6.2.3	IMEI	75
7.6.2.4	Previous location area Id	75
7.6.2.5	Stored location area Id.....	75
7.6.2.6	Current location area Id.....	75
7.6.2.7	Target location area Id.....	76
7.6.2.8	Target cell Id	76
7.6.2.9	Void.....	76
7.6.2.10	Originating entity number.....	76
7.6.2.11	MSC number	76
7.6.2.12	Target MSC number	76
7.6.2.13	HLR number	76
7.6.2.14	VLR number	76
7.6.2.15	HLR Id.....	76
7.6.2.16	LMSI	76
7.6.2.17	MS ISDN	76
7.6.2.18	OMC Id	76
7.6.2.19	Roaming number	76
7.6.2.20	Void.....	77
7.6.2.21	Handover number	77
7.6.2.22	Forwarded-to number	77
7.6.2.23	Forwarded-to subaddress.....	77
7.6.2.24	Called number	77
7.6.2.25	Calling number	77
7.6.2.26	Originally dialled number.....	77
7.6.2.27	Service centre address	77
7.6.2.28	Zone Code	77
7.6.2.29	MSIsdn-Alert.....	77
7.6.2.30	Location Information.....	77
7.6.2.31	GMSC Address.....	77
7.6.2.32	VMSC Address.....	77
7.6.2.33	Group Id	78
7.6.2.34	North American Equal Access preferred Carrier Id.....	78
7.6.2.35	SIWFS Number	78
7.6.2.36	B-subscriber address.....	78
7.6.2.37	Serving cell Id.....	78
7.6.2.38	SGSN number.....	78
7.6.2.39	SGSN address	78
7.6.2.40	GGSN address	78
7.6.2.41	GGSN number	78
7.6.2.42	APN	78
7.6.2.43	Network Node number	78
7.6.2.44	PDP-Type	78
7.6.2.45	PDP-Address	79
7.6.2.46	Additional number	79
7.6.2.47	P-TMSI.....	79
7.6.2.48	B-subscriber number.....	79
7.6.2.49	B-subscriber subaddress	79
7.6.2.50	LMU Number	79
7.6.2.51	MLC Number	79
7.6.3	Subscriber management parameters	79
7.6.3.1	Category	79
7.6.3.2	Equipment status.....	79

7.6.3.3	Extensible Bearer service	79
7.6.3.4	Extensible Teleservice	79
7.6.3.5	Extensible Basic Service Group	79
7.6.3.6	GSM bearer capability	80
7.6.3.7	Subscriber Status	80
7.6.3.8	CUG Outgoing Access indicator	80
7.6.3.9	Operator Determined Barring General Data	80
7.6.3.10	ODB HPLMN Specific Data	80
7.6.3.11	Regional Subscription Data	81
7.6.3.12	Regional Subscription Response	81
7.6.3.13	Roaming Restriction Due To Unsupported Feature	81
7.6.3.14	Extensible SS-Info	81
7.6.3.15	Extensible Forwarding information	81
7.6.3.16	Extensible Forwarding feature	81
7.6.3.17	Extensible SS-Status	81
7.6.3.18	Extensible Forwarding Options	82
7.6.3.19	Extensible No reply condition timer	82
7.6.3.20	Extensible Call barring information	82
7.6.3.21	Extensible Call barring feature	82
7.6.3.22	CUG info	82
7.6.3.23	CUG subscription	82
7.6.3.24	CUG interlock	82
7.6.3.25	CUG index	83
7.6.3.26	CUG feature	83
7.6.3.27	Inter CUG options	83
7.6.3.28	Intra CUG restrictions	83
7.6.3.29	Extensible SS-Data	83
7.6.3.30	Subscriber State	83
7.6.3.31	Requested Info	84
7.6.3.32	Suppression of Announcement	84
7.6.3.33	Suppress T-CSI	84
7.6.3.34	GMSC CAMEL Subscription Info	84
7.6.3.35	VLR CAMEL Subscription Info	84
7.6.3.36	Supported CAMEL Phases in the VLR	84
7.6.3.36A	Supported CAMEL Phases in the SGSN	84
7.6.3.37	CUG Subscription Flag	84
7.6.3.38	CAMEL Subscription Info Withdraw	84
7.6.3.39	Voice Group Call Service (VGCS) Data	84
7.6.3.40	Voice Broadcast Service (VBS) Data	84
7.6.3.41	ISDN bearer capability	84
7.6.3.42	Lower layer Compatibility	84
7.6.3.43	High Layer Compatibility	84
7.6.3.44	Alerting Pattern	85
7.6.3.45	GPRS Subscription Data Withdraw	85
7.6.3.46	GPRS Subscription Data	85
7.6.3.47	QoS-Subscribed	85
7.6.3.48	VPLMN address allowed	85
7.6.3.49	Roaming Restricted In SGSN Due To Unsupported Feature	85
7.6.3.50	Network Access Mode	85
7.6.3.51	Mobile Not Reachable Reason	85
7.6.3.52	Cancellation Type	85
7.6.3.53	All GPRS Data	85
7.6.3.54	Complete Data List Included	85
7.6.3.55	PDP Context Identifier	85
7.6.3.56	LSA Information	85
7.6.3.57	SoLSA support indicator	86
7.6.3.58	LSA Information Withdraw	86
7.6.3.59	LMU Indicator	86
7.6.3.60	LCS Information	86
7.6.3.61	GMLC List	86

7.6.3.62	LCS Privacy Exception List	86
7.6.3.63	LCS Privacy Exception Parameters	86
7.6.3.64	External Client List.....	86
7.6.3.65	Internal Client List.....	87
7.6.3.65A	MO-LR List	87
7.6.3.65B	Privacy Verification By MS User.....	87
7.6.3.65C	GMLC List Withdraw.....	87
7.6.3.66	IST Alert Timer	87
7.6.3.67	Call Termination Indicator	87
7.6.3.68	IST Information Withdraw	87
7.6.3.69	IST Support Indicator	87
7.6.3.70	Super-Charger Supported In HLR	87
7.6.3.71	Super-Charger Supported In Serving Network Entity	87
7.6.3.72	Age Indicator	88
7.6.3.73	GPRS enhancements support indicator.....	88
7.6.3.74	Extensible QoS-Subscribed	88
7.6.3.75	SGSN Camel Subscription Info.....	88
7.6.3.76	SMS-CSI	88
7.6.3.77	GPRS-CSI	88
7.6.4	Supplementary services parameters.....	89
7.6.4.1	SS-Code.....	89
7.6.4.2	SS-Status.....	90
7.6.4.3	SS-Data.....	90
7.6.4.4	Override Category	90
7.6.4.5	CLI Restriction Option	90
7.6.4.6	Forwarding Options.....	90
7.6.4.7	No reply condition timer.....	90
7.6.4.8 - 7.6.4.14	Void	91
7.6.4.15	Forwarding information.....	91
7.6.4.16	Forwarding feature	91
7.6.4.17	Void.....	91
7.6.4.18	Call barring information	91
7.6.4.19	Call barring feature.....	91
7.6.4.20	New password.....	91
7.6.4.21	Current password.....	91
7.6.4.22	Guidance information	92
7.6.4.23	Void.....	92
7.6.4.24	SS-Info.....	92
7.6.4.25 - 7.6.4.35	Void.....	92
7.6.4.36	USSD Data Coding Scheme	92
7.6.4.37	USSD String	92
7.6.4.38	Bearer service	92
7.6.4.39	Teleservice.....	92
7.6.4.40	Basic Service Group	93
7.6.4.41	eMLPP information	93
7.6.4.42	SS-event.....	93
7.6.4.43	SS-event data	93
7.6.4.44	LCS Privacy Exceptions	93
7.6.4.44A	Mobile Originating Location Request (MO-LR)	93
7.6.5	Call parameters.....	94
7.6.5.1	Call reference number.....	94
7.6.5.2	Interrogation type	94
7.6.5.3	OR interrogation.....	94
7.6.5.4	OR capability.....	94
7.6.5.5	Forwarding reason	94
7.6.5.6	Forwarding interrogation required.....	94
7.6.5.7	O-CSI.....	94
7.6.5.7A	D-CSI.....	94
7.6.5.7B	T-CSI.....	94
7.6.5.7C	VT-CSI.....	94

7.6.5.8	Call Direction	95
7.6.5.9	Channel Type.....	95
7.6.5.10	Chosen Channel.....	95
7.6.5.11	CCBS Feature.....	95
7.6.5.12	UU Data.....	95
7.6.5.14	Number Portability Status.....	95
7.6.5.15	Pre-paging supported.....	95
7.6.6	Radio parameters.....	95
7.6.6.1 - 7.6.6.6	Void	95
7.6.6.7	HO-Number Not Required	95
7.6.7	Authentication parameters.....	96
7.6.7.1	Authentication set list	96
7.6.7.2	Rand	96
7.6.7.3	Sres	96
7.6.7.4	Kc	96
7.6.7.5	Xres	96
7.6.7.5A	Ck 96	
7.6.7.5B	Ik 96	
7.6.7.5C	Autn	96
7.6.7.6	Cksn	96
7.6.7.6A	Ksi96	
7.6.7.6B	Auts	96
7.6.7.7	Ciphering mode	96
7.6.8	Short message parameters	97
7.6.8.1	SM-RP-DA	97
7.6.8.2	SM-RP-OA	97
7.6.8.3	MWD status.....	97
7.6.8.4	SM-RP-UI	97
7.6.8.5	SM-RP-PRI	97
7.6.8.6	SM Delivery Outcome.....	97
7.6.8.7	More Messages To Send	97
7.6.8.8	Alert Reason.....	97
7.6.8.9	Absent Subscriber Diagnostic SM.....	98
7.6.8.10	Alert Reason Indicator.....	98
7.6.8.11	Additional SM Delivery Outcome.....	98
7.6.8.12	Additional Absent Subscriber Diagnostic SM.....	98
7.6.8.13	Delivery Outcome Indicator	98
7.6.8.14	GPRS Node Indicator	98
7.6.8.15	GPRS Support Indicator.....	98
7.6.8.16	SM-RP-MTI	98
7.6.8.17	SM-RP-SMEA.....	98
7.6.9	Access and signalling system related parameters	98
7.6.9.1	BSS-apdu.....	98
7.6.9.2	CM service type.....	98
7.6.9.3	Access connection status	99
7.6.9.4	External Signal Information.....	99
7.6.9.5	Access signalling information.....	99
7.6.9.6	Location update type	99
7.6.9.7	Protocol ID	99
7.6.9.8	Network signal information	99
7.6.9.9	Call Info.....	100
7.6.9.10	Additional signal info	100
7.6.10	System operations parameters	101
7.6.10.1	Network resources	101
7.6.10.2	Trace reference	101
7.6.10.3	Trace type	101
7.6.11	Location Service Parameters	101
7.6.11.1	Age of Location Estimate	101
7.6.11.2	Void.....	101
7.6.11.3	Void.....	101

7.6.11.4	LCS Client ID	101
7.6.11.5	LCS Event	101
7.6.11.6	LCS MLC Data.....	102
7.6.11.7	LCS Priority.....	102
7.6.11.8	LCS QoS.....	102
7.6.11.9	Void.....	102
7.6.11.10	Void.....	102
7.6.11.11	Location Estimate	102
7.6.11.12	Location Type.....	102
7.6.11.13	NA-ESRD.....	103
7.6.11.14	NA-ESRK.....	103
7.6.11.15	Void.....	103
7.6.11.16	Privacy Override.....	103
7.6.11.17	Void.....	103
7.6.11.18	Void.....	103
7.6.11.19	Void.....	103
7.7	Representation of a list of a basic parameter in service-primitives	103
8	Mobility services	104
8.1	Location management services	104
8.1.1	Void	104
8.1.1.1	Void.....	104
8.1.1.2	Void.....	104
8.1.1.3	Void8.1.2 MAP_UPDATE_LOCATION service.....	104
8.1.2.1	Definition.....	104
8.1.2.2	Service primitives	104
8.1.2.3	Parameter definitions and use	104
8.1.3	MAP_CANCEL_LOCATION service.....	106
8.1.3.1	Definition.....	106
8.1.3.2	Service primitives	106
8.1.3.3	Parameter definitions and use	106
8.1.4	MAP_SEND_IDENTIFICATION service.....	107
8.1.4.1	Definition.....	107
8.1.4.2	Service primitives	107
8.1.4.3	Parameter definitions and use	107
8.1.5	Void	108
8.1.5.1	Void.....	108
8.1.5.2	Void.....	108
8.1.5.3	Void.....	108
8.1.6	MAP_PURGE_MS service.....	108
8.1.6.1	Definition.....	108
8.1.6.2	Service primitives	108
8.1.6.3	Parameter definitions and use	108
8.1.7	MAP_UPDATE_GPRS_LOCATION service.....	109
8.1.7.1	Definition.....	109
8.1.7.2	Service primitives	109
8.1.7.3	Parameter definitions and use	109
8.1.8	MAP-NOTE-MM-EVENT	111
8.1.8.1	Definition	111
8.1.8.2	Service primitives.....	111
8.1.8.3	Parameter use	111
8.2	Paging and search	112
8.2.1	MAP_PAGE service	112
8.2.1.1	Definition.....	112
8.2.1.2	Service primitives	112
8.2.1.3	Parameter definitions and use	113
8.2.2	MAP_SEARCH_FOR_MS service.....	113
8.2.2.1	Definition.....	113
8.2.2.2	Service primitives	113
8.2.2.3	Parameter definitions and use	114
8.3	Access management services	114

8.3.1	MAP_PROCESS_ACCESS_REQUEST service.....	114
8.3.1.1	Definition.....	114
8.3.1.2	Service primitives.....	115
8.3.1.3	Parameter definitions and use.....	115
8.4	Handover services.....	116
8.4.1	MAP_PREPARE_HANOVER service.....	116
8.4.1.1	Definition.....	116
8.4.1.2	Service primitives.....	116
8.4.1.3	Parameter use.....	117
8.4.2	MAP_SEND_END_SIGNAL service.....	117
8.4.2.1	Definition.....	117
8.4.2.2	Service primitives.....	117
8.4.2.3	Parameter use.....	118
8.4.3	MAP_PROCESS_ACCESS_SIGNALLING service.....	118
8.4.3.1	Definition.....	118
8.4.3.2	Service primitives.....	118
8.4.3.3	Parameter use.....	118
8.4.4	MAP_FORWARD_ACCESS_SIGNALLING service.....	118
8.4.4.1	Definition.....	118
8.4.4.2	Service primitives.....	119
8.4.4.3	Parameter use.....	119
8.4.5	MAP_PREPARE_SUBSEQUENT_HANOVER service.....	119
8.4.5.1	Definition.....	119
8.4.5.2	Service primitives.....	119
8.4.5.3	Parameter use.....	119
8.4.6	MAP_ALLOCATE_HANOVER_NUMBER service.....	120
8.4.6.1	Definition.....	120
8.4.6.2	Service primitives.....	120
8.4.6.3	Parameter use.....	120
8.4.7	MAP_SEND_HANOVER_REPORT service.....	120
8.4.7.1	Definition.....	120
8.4.7.2	Service primitives.....	121
8.4.7.3	Parameter use.....	121
8.5	Authentication management services.....	121
8.5.1	MAP_AUTHENTICATE service.....	121
8.5.1.1	Definition.....	121
8.5.1.2	Service primitives.....	121
8.5.1.3	Parameter use.....	122
8.5.2	MAP_SEND_AUTHENTICATION_INFO service.....	122
8.5.2.1	Definition.....	122
8.5.2.2	Service primitives.....	122
8.5.2.3	Parameter use.....	123
8.6	Security management services.....	124
8.6.1	MAP_SET_CIPHERING_MODE service.....	124
8.6.1.1	Definitions.....	124
8.6.1.2	Service primitives.....	124
8.6.1.3	Parameter use.....	124
8.7	International mobile equipment identities management services.....	124
8.7.1	MAP_CHECK_IMEI service.....	124
8.7.1.1	Definition.....	124
8.7.1.2	Service primitives.....	124
8.7.1.3	Parameter use.....	125
8.7.2	MAP_OBTAIN_IMEI service.....	125
8.7.2.1	Definition.....	125
8.7.2.2	Service primitives.....	125
8.7.2.3	Parameter use.....	126
8.8	Subscriber management services.....	126
8.8.1	MAP-INSERT-SUBSCRIBER-DATA service.....	126
8.8.1.1	Definition.....	126
8.8.1.2	Service primitives.....	127

8.8.1.3	Parameter use.....	127
8.8.1.4	Basic service information related to supplementary services	135
8.8.2	MAP-DELETE-SUBSCRIBER-DATA service.....	135
8.8.2.1	Definition.....	135
8.8.2.2	Service primitives	136
8.8.2.3	Parameter use.....	136
8.9	Identity management services	138
8.9.1	MAP-PROVIDE-IMSI service.....	138
8.9.1.1	Definition.....	138
8.9.1.2	Service primitives	138
8.9.1.3	Parameter use.....	138
8.9.2	MAP-FORWARD-NEW-TMSI service.....	139
8.9.2.1	Definition.....	139
8.9.2.2	Service primitives	139
8.9.2.3	Parameter use.....	139
8.10	Fault recovery services	139
8.10.1	MAP_RESET service.....	139
8.10.1.1	Definition.....	139
8.10.1.2	Service primitives	139
8.10.1.3	Parameter definition and use	139
8.10.2	MAP_FORWARD_CHECK_SS_INDICATION service.....	140
8.10.2.1	Definition.....	140
8.10.2.2	Service primitives	140
8.10.2.3	Parameter definition and use	140
8.10.3	MAP_RESTORE_DATA service	140
8.10.3.1	Definition.....	140
8.10.3.2	Service primitives	141
8.10.3.3	Parameter definitions and use.....	141
8.11	Subscriber Information services.....	142
8.11.1	MAP-ANY-TIME-INTERROGATION service	142
8.11.1.1	Definition.....	142
8.11.1.2	Service primitives	142
8.11.1.3	Parameter definition and use	142
8.11.2	MAP-PROVIDE-SUBSCRIBER-Info service.....	143
8.11.2.1	Definition.....	143
8.11.2.2	Service primitives	143
8.11.2.3	Parameter definition and use	143
8.11.3	MAP-ANY-TIME-SUBSCRIPTION-INTERROGATION service.....	143
8.11.3.1	Definition.....	143
8.11.3.2	Service primitives	144
8.11.3.3	Parameter definition and use	144
8.11.4	MAP-ANY-TIME-MODIFICATION service	145
8.11.4.1	Definition.....	145
8.11.4.2	Service primitives	145
8.11.4.3	Parameter definition and use	145
8.11.5	MAP-NOTE-SUBSCRIBER-DATA-MODIFIED service	146
8.11.5.1	Definition.....	146
8.11.5.2	Service primitives	146
8.11.5.3	Parameter definition and use	146
9	Operation and maintenance services	148
9.1	Subscriber tracing services	148
9.1.1	MAP-ACTIVATE-TRACE-MODE service	148
9.1.1.1	Definition.....	148
9.1.1.2	Service primitives	148
9.1.1.3	Parameter use.....	148
9.1.2	MAP-DEACTIVATE-TRACE-MODE service	149
9.1.2.1	Definition.....	149
9.1.2.2	Service primitives	149
9.1.2.3	Parameter use.....	149
9.1.3	MAP-TRACE-SUBSCRIBER-ACTIVITY service.....	150

9.1.3.1	Definition.....	150
9.1.3.2	Service primitives.....	150
9.1.3.3	Parameter use.....	150
9.2	Other operation and maintenance services.....	150
9.2.1	MAP-SEND-IMSI service	150
9.2.1.1	Definition.....	150
9.2.1.2	Service primitives.....	151
9.2.1.3	Parameter use.....	151
10	Call handling services.....	151
10.1	MAP_SEND_ROUTING_INFORMATION service	151
10.1.1	Definition	151
10.1.2	Service primitives.....	152
10.1.3	Parameter use	152
10.2	MAP_PROVIDE_ROAMING_NUMBER service	156
10.2.1	Definition	156
10.2.2	Service primitives.....	157
10.2.3	Parameter use	157
10.3	MAP_RESUME_CALL_HANDLING service	159
10.3.1	Definition	159
10.3.2	Service primitives.....	159
10.3.3	Parameter use	159
10.4	MAP_PREPARE_GROUP_CALL service	161
10.4.1	Definition	161
10.4.2	Service primitives.....	161
10.4.3	Parameter definitions and use.....	161
10.5	MAP_PROCESS_GROUP_CALL_SIGNALLING service.....	162
10.5.1	Definitions.....	162
10.5.2	Service primitives.....	162
10.5.3	Parameter definitions and use.....	162
10.6	MAP_FORWARD_GROUP_CALL_SIGNALLING service	163
10.6.1	Definitions.....	163
10.6.2	Service primitives.....	163
10.6.3	Parameter definitions and use.....	163
10.7	MAP_SEND_GROUP_CALL_END_SIGNAL service.....	164
10.7.1	Definitions.....	164
10.7.2	Service primitives.....	164
10.7.3	Parameter definitions and use.....	164
10.8	MAP_Provide_SIWFS_Number.....	164
10.8.1	Definition	164
10.8.2	Service primitive	165
10.8.3	Parameter use	165
10.9	MAP_SIWFS_Signalling_Modify.....	166
10.9.1	Definition	166
10.9.2	Service primitive	166
10.9.3	Parameter use	166
10.10	MAP_SET_REPORTING_STATE service.....	167
10.10.1	Definition	167
10.10.2	Service primitives.....	167
10.10.3	Parameter use	167
10.11	MAP_STATUS_REPORT service	168
10.11.1	Definition	168
10.11.2	Service primitives.....	168
10.11.3	Parameter use	168
10.12	MAP_REMOTE_USER_FREE service	169
10.12.1	Definition	169
10.12.2	Service primitives.....	169
10.12.3	Parameter use	169
10.13	MAP_IST_ALERT service.....	170
10.13.1	Definition	170
10.13.2	Service primitives.....	170

10.13.3	Parameter use	171
10.14	MAP_IST_COMMAND service	171
10.14.1	Definition	171
10.14.2	Service primitives.....	171
10.14.3	Parameter use	172
11	Supplementary services related services	172
11.1	MAP_REGISTER_SS service	172
11.1.1	Definition	172
11.1.2	Service primitives.....	172
11.1.3	Parameter use	172
11.2	MAP_ERASE_SS service	173
11.2.1	Definition	173
11.2.2	Service primitives.....	174
11.2.3	Parameter use	174
11.3	MAP_ACTIVATE_SS service	175
11.3.1	Definition	175
11.3.2	Service primitives.....	175
11.3.3	Parameter use	175
11.4	MAP_DEACTIVATE_SS service.....	176
11.4.1	Definitions.....	176
11.4.2	Service primitives.....	176
11.4.3	Parameter use	177
11.5	MAP_INTERROGATE_SS service	178
11.5.1	Definitions.....	178
11.5.2	Service primitives.....	178
11.5.3	Parameter use	178
11.6	MAP_INVOKE_SS service.....	179
11.6.1	Definitions.....	179
11.6.2	Service primitives.....	180
11.6.3	Parameter use	180
11.7	MAP_REGISTER_PASSWORD service.....	180
11.7.1	Definitions.....	180
11.7.2	Service primitives.....	181
11.7.3	Parameter use	181
11.8	MAP_GET_PASSWORD service.....	181
11.8.1	Definitions.....	181
11.8.2	Service primitives.....	182
11.8.3	Parameter use	182
11.9	MAP_PROCESS_UNSTRUCTURED_SS_REQUEST service	182
11.9.1	Definitions.....	182
11.9.2	Service primitives.....	182
11.9.3	Parameter use	183
11.10	MAP_UNSTRUCTURED_SS_REQUEST service	183
11.10.1	Definitions.....	183
11.10.2	Service primitives.....	184
11.10.3	Parameter use	184
11.11	MAP_UNSTRUCTURED_SS_NOTIFY service.....	185
11.11.1	Definitions.....	185
11.11.2	Service primitives.....	185
11.11.3	Parameter use	185
11.12	MAP_SS_INVOCATION_NOTIFY	186
11.12.1	Definition	186
11.12.2	Service primitives.....	186
11.12.3	Parameter use	186
11.13	MAP_REGISTER_CC_ENTRY service.....	186
11.13.1	Definition	186
11.13.2	Service primitives.....	187
11.13.3	Parameter use	187
11.14	MAP_ERASE_CC_ENTRY service	188
11.14.1	Definition	188

11.14.2	Service primitives.....	188
11.14.3	Parameter use	188
12	Short message service management services.....	189
12.1	MAP-SEND-ROUTING-INFO-FOR-SM service	189
12.1.1	Definition	189
12.1.2	Service primitives.....	189
12.1.3	Parameter use	190
12.2	MAP-MO-FORWARD-SHORT-MESSAGE service.....	191
12.2.1	Definition	191
12.2.2	Service primitives.....	191
12.2.3	Parameter use	191
12.3	MAP-REPORT-SM-DELIVERY-STATUS service	192
12.3.1	Definition	192
12.3.2	Service primitives.....	192
12.3.3	Parameter use	193
12.4	MAP-READY-FOR-SM service	194
12.4.1	Definition	194
12.4.2	Service primitives.....	194
12.4.3	Parameter use	194
12.5	MAP-ALERT-SERVICE-CENTRE service.....	195
12.5.1	Definition	195
12.5.2	Service primitives.....	195
12.5.3	Parameter use	195
12.6	MAP-INFORM-SERVICE-CENTRE service	196
12.6.1	Definition	196
12.6.2	Service primitives.....	196
12.6.3	Parameter use	196
12.7	MAP-SEND-INFO-FOR-MT-SMS service.....	197
12.7.1	Definition	197
12.7.2	Service primitives.....	197
12.7.3	Parameter use	197
12.8	MAP-SEND-INFO-FOR-MO-SMS service	198
12.8.1	Definition	198
12.8.2	Service primitives.....	198
12.8.3	Parameter use	198
12.9	MAP-MT-FORWARD-SHORT-MESSAGE service.....	199
12.9.1	Definition	199
12.9.2	Service primitives.....	199
12.9.3	Parameter use	199
13	Network-Requested PDP Context Activation services.....	200
13.1	MAP_SEND_ROUTING_INFO_FOR_GPRS service	200
13.1.1	Definition	200
13.1.2	Service primitives.....	200
13.1.3	Parameter definition and use	201
13.2	MAP_FAILURE_REPORT service	201
13.2.1	Definition	201
13.2.2	Service primitives.....	202
13.2.3	Parameter definition and use	202
13.3	MAP_NOTE_MS_PRESENT_FOR_GPRS service	202
13.3.1	Definition	202
13.3.2	Service primitives.....	203
13.3.3	Parameter definition and use	203
13A	Location Service Management Services.....	204
13A.1	MAP-SEND-ROUTING-INFO-FOR-LCS Service	204
13A.1.1	Definition	204
13A.1.2	Service Primitives	204
13A.1.3	Parameter Use	204
13A.2	MAP-PROVIDE-SUBSCRIBER-LOCATION Service	206

13A.2.1	Definition	206
13A.2.2	Service Primitives	206
13A.2.3	Parameter Definition and Use	207
13A.3	MAP-SUBSCRIBER-LOCATION-REPORT Service	208
13A.3.1	Definition	208
13A.3.2	Service Primitives	208
13A.3.3	Parameter Definition and Use	208
13A.4	Void	210
13A.4.1	Void	210
13A.4.2	Void	210
13A.4.3	Void	210
13A.5	Void	210
13A.5.1	Void	210
13A.5.2	Void	210
13A.5.3	Void	210
13A.6	Void	210
13A.6.1	Void	211
13A.6.2	Void	211
13A.6.3	Void	211
13A.7	Void	211
13A.7.1	Void	211
13A.7.2	Void	211
13A.7.3	Void	211
13A.8	Void	211
13A.8.1	Void	211
13A.8.2	Void	211
13A.8.3	Void	211
13A.9	Void	211
13A.9.1	Void	211
13A.9.2	Void	211
13A.9.3	Void	211
14	General.....	211
14.1	Overview.....	211
14.2	Underlying services	211
14.3	Model.....	212
14.4	Conventions	212
15	Elements of procedure	212
15.1	Dialogue establishment.....	212
15.1.1	Handling of unknown operations	213
15.1.2	Receipt of a MAP-OPEN request primitive	213
15.1.3	Receipt of a TC-BEGIN indication.....	214
15.1.4	Receipt of a MAP-OPEN response	216
15.1.5	Receipt of the first TC-CONTINUE ind	216
15.1.6	Receipt of a TC-END ind.....	216
15.1.7	Receipt of a TC-U-ABORT ind	217
15.1.8	Receipt of a TC-P-ABORT ind	217
15.2	Dialogue continuation.....	217
15.2.1	Sending entity.....	217
15.2.2	Receiving entity.....	217
15.3	Dialogue termination	218
15.3.1	Receipt of a MAP-CLOSE request.....	218
15.3.2	Receipt of a TC-END indication.....	218
15.4	User Abort	218
15.4.1	MAP-U-ABORT request.....	218
15.4.2	TC-U-ABORT ind	218
15.5	Provider Abort.....	218
15.5.1	MAP PM error situation.....	219
15.5.2	TC-P-ABORT ind	219
15.5.3	TC-U-ABORT ind	219

15.6	Procedures for MAP specific services	219
15.6.1	Service invocation.....	219
15.6.2	Service invocation receipt.....	220
15.6.3	Service response.....	220
15.6.4	Receipt of a response	221
15.6.4.1	Receipt of a TC-RESULT-NL indication	221
15.6.4.2	Receipt of a TC-RESULT-L indication.....	221
15.6.4.3	Receipt of a TC-U-ERROR indication	222
15.6.4.4	Receipt of a TC-INVOKE indication	222
15.6.4.5	Receipt of a TC-U-REJECT indication	222
15.6.4.6	Receipt of a TC-L-REJECT indication.....	223
15.6.4.7	Receipt of a TC-L-CANCEL indication	223
15.6.4.8	Receipt of a TC-NOTICE indication.....	223
15.6.5	Other events	223
15.6.5.1	Receipt of a TC-U-REJECT	224
15.6.5.2	Receipt of a TC-R-REJECT indication	224
15.6.5.3	Receipt of a TC-L-REJECT indication.....	224
15.6.6	Parameter checks.....	224
15.6.7	Returning state machines to idle.....	224
15.6.8	Load control	225
16	Mapping on to TC services.....	225
16.1	Dialogue control	225
16.1.1	Directly mapped parameters.....	225
16.1.2	Use of other parameters of dialogue handling primitives	225
16.1.2.1	Dialogue Id.....	225
16.1.2.2	Application-context-name	225
16.1.2.3	User information.....	225
16.1.2.4	Component present	225
16.1.2.5	Termination	225
16.1.2.6	P-Abort-Cause	226
16.1.2.7	Quality of service.....	226
16.2	Service specific procedures	226
16.2.1	Directly mapped parameters.....	226
16.2.2	Use of other parameters of component handling primitives	226
16.2.2.1	Dialogue Id.....	226
16.2.2.2	Class	227
16.2.2.3	Linked Id	227
16.2.2.4	Operation	227
16.2.2.5	Error	229
16.2.2.6	Parameters	229
16.2.2.7	Time out	229
16.2.2.8	Last component.....	229
16.2.2.9	Problem code.....	229
16.2.2.9.1	Mapping to MAP User Error	229
16.2.2.9.2	Mapping to MAP Provider Error parameter	229
16.2.2.9.3	Mapping to diagnostic parameter.....	230
16.3	SDL descriptions	231
17	Abstract syntax of the MAP protocol.....	257
17.1	General.....	257
17.1.1	Encoding rules.....	257
17.1.2	Use of TC.....	257
17.1.2.1	Use of Global Operation and Error codes defined outside MAP.....	258
17.1.3	Use of information elements defined outside MAP.....	258
17.1.4	Compatibility considerations.....	259
17.1.5	Structure of the Abstract Syntax of MAP.....	259
17.1.6	Application Contexts.....	261
17.2	Operation packages.....	262
17.2.1	General aspects	262
17.2.2	Packages specifications	263

17.2.2.1	Location updating	263
17.2.2.2	Location cancellation	263
17.2.2.3	Roaming number enquiry	264
17.2.2.4	Information retrieval	264
17.2.2.5	Inter-VLR information retrieval	264
17.2.2.6	IMSI retrieval	264
17.2.2.7	Call control transfer	265
17.2.2.8 - 17.2.2.9	Void	265
17.2.2.10	Interrogation	265
17.2.2.11	Void	265
17.2.2.12	Handover Control	265
17.2.2.13	Subscriber Data management stand alone	265
17.2.2.14	Equipment management	266
17.2.2.15	Subscriber data management	266
17.2.2.16	Location register restart	266
17.2.2.17	Tracing stand-alone	266
17.2.2.18	Functional SS handling	266
17.2.2.19	Tracing	267
17.2.2.20	Binding	267
17.2.2.21	Unstructured SS handling	267
17.2.2.22	MO Short message relay services	267
17.2.2.23	Short message gateway services	268
17.2.2.24	MT Short message relay services	268
17.2.2.25	Void	268
17.2.2.26	Message waiting data management	268
17.2.2.27	Alerting	268
17.2.2.28	Data restoration	269
17.2.2.29	Purging	269
17.2.2.30	Subscriber information enquiry	269
17.2.2.31	Any time information enquiry	269
17.2.2.32	Group Call Control	270
17.2.2.33	Provide SIWFS number	270
17.2.2.34	SIWFS Signalling Modify	270
17.2.2.35	Gprs location updating	270
17.2.2.36	Gprs Interrogation	270
17.2.2.37	Failure reporting	271
17.2.2.38	GPRS notifying	271
17.2.2.39	Supplementary Service invocation notification	271
17.2.2.40	Set Reporting State	271
17.2.2.41	Status Report	271
17.2.2.42	Remote User Free	272
17.2.2.43	Call Completion	272
17.2.2.44	Location service gateway services	272
17.2.2.45	Location service enquiry	272
17.2.2.46	Void	272
17.2.2.47	Void	272
17.2.2.48	Void	272
17.2.2.49	IST Alerting	273
17.2.2.50	Service Termination	273
17.2.2.51	Mobility Management event notification	273
17.2.2.53	Subscriber Data modification notification	273
17.3	Application contexts	274
17.3.1	General aspects	274
17.3.2	Application context definitions	275
17.3.2.1	Void	275
17.3.2.2	Location Updating	275
17.3.2.3	Location Cancellation	275
17.3.2.4	Roaming number enquiry	275
17.3.2.5	Void	276
17.3.2.6	Location Information Retrieval	276

17.3.2.7	Call control transfer	276
17.3.2.8 - 17.3.2.10	Void	276
17.3.2.11	Location registers restart	276
17.3.2.12	Handover control	276
17.3.2.13	IMSI Retrieval	277
17.3.2.14	Equipment Management	277
17.3.2.15	Information retrieval	277
17.3.2.16	Inter-VLR information retrieval	277
17.3.2.17	Stand Alone Subscriber Data Management	278
17.3.2.18	Tracing	278
17.3.2.19	Network functional SS handling	278
17.3.2.20	Network unstructured SS handling	279
17.3.2.21	Short Message Gateway	279
17.3.2.22	Mobile originating Short Message Relay	279
17.3.2.23	Void	280
17.3.2.24	Short message alert	280
17.3.2.25	Short message waiting data management	280
17.3.2.26	Mobile terminating Short Message Relay	280
17.3.2.27	MS purging	281
17.3.2.28	Subscriber information enquiry	281
17.3.2.29	Any time information enquiry	281
17.3.2.30	Group Call Control	281
17.3.2.31	Provide SIWFS Number	281
17.3.2.32	Gprs Location Updating	282
17.3.2.33	Gprs Location Information Retrieval	282
17.3.2.34	Failure Reporting	282
17.3.2.35	GPRS Notifying	282
17.3.2.36	Supplementary Service invocation notification	282
17.3.2.37	Reporting	283
17.3.2.38	Call Completion	283
17.3.2.39	Location Service Gateway	284
17.3.2.40	Location Service Enquiry	284
17.3.2.41	Void	284
17.3.2.42	Void	284
17.3.2.43	Void	284
17.3.2.44	IST Alerting	284
17.3.2.45	Service Termination	284
17.3.2.46	Mobility Management event notification	284
17.3.2.48	Subscriber Data modification notification	285
17.3.3	ASN.1 Module for application-context-names	285
17.4	MAP Dialogue Information	287
17.5	MAP operation and error codes	289
17.6	MAP operation and error types	295
17.6.1	Mobile Service Operations	295
17.6.2	Operation and Maintenance Operations	301
17.6.3	Call Handling Operations	302
17.6.4	Supplementary service operations	305
17.6.5	Short message service operations	309
17.6.6	Errors	311
17.6.7	Group Call operations	317
17.6.8	Location service operations	318
17.7	MAP constants and data types	319
17.7.1	Mobile Service data types	319
17.7.2	Operation and maintenance data types	341
17.7.3	Call handling data types	342
17.7.4	Supplementary service data types	348
17.7.5	Supplementary service codes	352
17.7.6	Short message data types	355
17.7.7	Error data types	357
17.7.8	Common data types	363

17.7.9	Teleservice Codes	368
17.7.10	Bearer Service Codes	369
17.7.11	Extension data types	371
17.7.12	Group Call data types	372
17.7.13	Location service data types	374
18	General on MAP user procedures	378
18.1	Introduction	378
18.2	Common aspects of user procedure descriptions	378
18.2.1	General conventions	378
18.2.2	Naming conventions	378
18.2.3	Convention on primitives parameters	380
18.2.3.1	Open service	380
18.2.3.2	Close service	381
18.2.4	Version handling at dialogue establishment	381
18.2.4.1	Behaviour at the initiating side	381
18.2.4.2	Behaviour at the responding side	381
18.2.5	Abort Handling	381
18.2.6	SDL conventions	381
18.3	Interaction between MAP Provider and MAP Users	382
19	Mobility procedures	382
19.1	Location management Procedures	382
19.1.1	Location updating	385
19.1.1.1	General	385
19.1.1.3	Detailed procedure in the VLR	391
19.1.1.4	Detailed procedure in the HLR	400
19.1.1.5	Send Identification	410
19.1.1.5.1	General	410
19.1.1.5.2	Detailed procedure in the VLR	410
19.1.1.5.3	Detailed procedure in the PVLRL	410
19.1.1.6	Process Update Location VLR	415
19.1.1.8	Detailed procedure in the SGSN	417
19.1.2	Location Cancellation	420
19.1.2.1	General	420
19.1.2.2	Detailed procedure in the HLR	420
19.1.2.3	Detailed procedure in the VLR	421
19.1.2.4	Detailed procedure in the SGSN	424
19.1.3	Void	428
19.1.3.1	Void	428
19.1.3.2	Void	428
19.1.3.3	Void	428
19.1.4	Purge MS	428
19.1.4.1	General	428
19.1.4.2	Void	429
19.1.4.3	Void	429
19.1.4.4	Detailed procedure in the SGSN	429
19.2	Handover procedure	436
19.2.1	General	436
19.2.2	Handover procedure in MSC-A	439
19.2.2.1	Basic handover	439
19.2.2.2	Handling of access signalling	439
19.2.2.3	Other procedures in stable handover situation	439
19.2.2.4	Subsequent handover	439
19.2.2.5	SDL Diagrams	440
19.2.3	Handover procedure in MSC-B	453
19.2.3.1	Basic handover	453
19.2.3.2	Allocation of handover number	453
19.2.3.3	Handling of access signalling	453
19.2.3.4	Other procedures in stable handover situation	453
19.2.3.5	Subsequent handover	453

19.2.3.6	SDL Diagrams	453
19.2.4	Handover error handling macro	466
19.2.5	Handover procedure in VLR	468
19.2.5.1	Allocation of handover number	468
19.2.5.2	SDL Diagrams	468
19.3	Fault recovery procedures	471
19.3.1	VLR fault recovery procedures	471
19.3.2	HLR fault recovery procedures	473
19.3.3	VLR restoration: the restore data procedure in the HLR	481
19.4	Macro Insert_Subst_Data_Framed_HLR	483
19.5	Mobility Management Event notification procedure	487
19.5.1	General	487
19.5.2	Process in the VLR	487
19.5.3	Process in the gsmSCF	489
20	Operation and maintenance procedures	490
20.1	General	490
20.1.1	Tracing Co-ordinator for the VLR	491
20.1.2	Subscriber Data Management Co-ordinator for the VLR	493
20.1.3	Tracing Co-ordinator for the SGSN	495
20.1.4	Subscriber Data Management Co-ordinator for the SGSN	497
20.2	Tracing procedures	499
20.2.1	Procedures in the HLR	502
20.2.1.1	Subscriber tracing activation procedure	502
20.2.1.2	Subscriber tracing deactivation procedure	507
20.2.2	Procedures in the VLR	512
20.2.2.1	Subscriber tracing activation procedure	512
20.2.2.2	Subscriber tracing deactivation procedure	514
20.2.2.3	Subscriber tracing procedure	516
20.2.3	Procedures in the MSC	516
20.2.3.1	Subscriber tracing procedure	516
20.2.4	Procedures in the SGSN	516
20.2.4.1	Subscriber tracing activation procedure	516
20.2.4.2	Subscriber tracing deactivation procedure in SGSN	516
20.3	Subscriber data management procedures	519
20.3.1	Procedures in the HLR	520
20.3.1.1	Subscriber deletion procedure	520
20.3.1.2	Subscriber data modification procedure	522
20.3.2	Procedures in the VLR	527
20.3.2.1	Subscriber deletion procedure	527
20.3.2.2	Subscriber data modification procedure	527
20.3.3	Procedures in the SGSN	530
20.3.3.1	Subscriber deletion procedure	530
20.3.3.2	Subscriber data modification procedure	530
20.4	Subscriber Identity procedure	533
20.4.1	Subscriber identity procedure in the HLR	533
20.4.2	Subscriber identity procedure in the VLR	535
21	Call handling procedures	537
21.1	General	537
21.2	Retrieval of routing information	538
21.2.1	General	538
21.2.2	Process in the GMSC	539
21.2.3	Procedures in the HLR	543
21.2.4	Process in the VLR to provide a roaming number	549
21.2.5	Process in the VLR to restore subscriber data	551
21.2.6	Process in the VLR to provide subscriber information	553
21.2.7	Process in the HLR for Any Time Interrogation	555
21.2.7.1	Process in the gsmSCF	555
21.2.7.2	Process in the HLR	555
21.2.8	Process in the GMLC for Any Time Interrogation	559

21.2.8.1	Process in the gsmSCF	559
21.2.8.2	Process in the GMLC	559
21.3	Transfer of call handling.....	562
21.3.1	General.....	562
21.3.2	Process in the VMSC	562
21.3.3	Process in the GMSC	565
21.4	Inter MSC Group Call Procedures.....	567
21.4.1	General.....	567
21.4.2	Process in the Anchor MSC	567
21.4.3	Process in the Relay MSC	573
21.5	Allocation and modifications of resources in an SIWFS	578
21.5.1	General.....	578
21.5.2	Process in the VMSC	582
21.5.2.1	Allocation of SIWFS resources	582
21.5.2.2	Modification of SIWFS resources initiated by the user.....	583
21.5.2.3	Modification of SIWFS resources initiated by the SIWFS	584
21.5.3	Process in the SIWFS.....	591
21.5.3.1	Procedures for allocation of SIWFS resources.....	591
21.5.3.2	Process for modification of SIWFS resources initiated by the user.....	592
21.5.3.3	Process for modification of SIWFS resources initiated by the SIWFS.....	592
21.6	Setting of Reporting State	598
21.6.1	General.....	598
21.6.2	Process in the HLR for Set Reporting State stand-alone	598
21.6.3	Reporting co-ordinator process in the VLR	601
21.6.4	Process in the VLR to set the reporting state.....	603
21.7	Status Reporting.....	606
21.7.1	General.....	606
21.7.2	Process in the VLR for Status Reporting.....	607
21.7.3	Process in the HLR for Status Reporting.....	610
21.8	Remote User Free	615
21.8.1	General.....	615
21.8.2	Process in the HLR for Remote User Free	615
21.8.3	Process in the VLR for Remote User Free	618
21.9	IST Alert.....	621
21.9.1	General.....	621
21.9.2	Procedure in the MSC.....	621
21.9.3	Procedure in the HLR.....	624
21.10	IST Command.....	626
21.10.1	General.....	626
21.10.2	Procedure in the HLR.....	626
21.10.3	Procedure in the MSC	629
22	Supplementary services procedures.....	631
22.1	Functional supplementary service processes.....	631
22.1.1	Functional supplementary service process co-ordinator for MSC.....	631
22.1.2	Functional supplementary service process co-ordinator for VLR	633
22.1.3	Functional supplementary service process co-ordinator for HLR	636
22.1.4	Call completion supplementary service process co-ordinator for HLR.....	639
22.2	Registration procedure.....	641
22.2.1	General.....	641
22.2.2	Procedures in the MSC.....	641
22.2.3	Procedures in the VLR	644
22.2.4	Procedures in the HLR	647
22.3	Erasure procedure	650
22.3.1	General.....	650
22.3.2	Procedures in the MSC.....	650
22.3.3	Procedures in the VLR.....	651
22.3.4	Procedures in the HLR.....	651
22.4	Activation procedure.....	651
22.4.1	General.....	651
22.4.2	Procedures in the MSC.....	652

22.4.3	Procedures in the VLR	654
22.4.4	Procedures in the HLR	657
22.5	Deactivation procedure	660
22.5.1	General	660
22.5.2	Procedures in the MSC.....	661
22.5.3	Procedures in the VLR	661
22.5.4	Procedures in the HLR	661
22.6	Interrogation procedure	661
22.6.1	General.....	661
22.6.2	Procedures in the MSC.....	662
22.6.3	Procedures in the VLR.....	662
22.6.4	Procedures in the HLR.....	667
22.7	Invocation procedure	669
22.7.1	General.....	669
22.7.2	Procedures in the MSC.....	669
22.7.3	Procedures in the VLR.....	673
22.8	Password registration procedure	675
22.8.1	General.....	675
22.8.2	Procedures in the MSC.....	676
22.8.3	Procedures in the VLR.....	676
22.8.4	Procedures in the HLR.....	676
22.9	Mobile Initiated USSD procedure	679
22.9.1	General.....	679
22.9.2	Procedures in the MSC.....	679
22.9.3	Procedures in the VLR.....	683
22.9.4	Procedures in the HLR.....	688
22.10	Network initiated USSD procedure	692
22.10.1	General.....	692
22.10.2	Procedure in the MSC.....	693
22.10.3	Procedure in the VLR.....	698
22.10.4	Procedure in the HLR.....	705
22.11	Common macros for clause 22.....	711
22.11.1	SS Password handling macros.....	712
22.11.2	SS Error handling macros	715
22.12	Supplementary Service Invocation Notification procedure	721
22.12.1	General.....	721
22.12.2	Procedures in the MSC.....	721
22.12.3	Procedures in the gsmSCF	723
22.13	Activation of a CCBS request.....	725
22.13.1	General.....	725
22.13.2	Procedure in the VLR.....	725
22.13.3	Procedure in the HLR.....	728
22.14	Deactivation of a CCBS request	730
22.14.1	General.....	730
22.14.2	Procedure in the VLR.....	730
22.14.3	Procedure in the HLR.....	733
23	Short message service procedures	735
23.1	General.....	735
23.1.1	Mobile originated short message service Co-ordinator for the MSC	735
23.1.2	Short message Gateway Co-ordinator for the HLR.....	737
23.1.3	Mobile originated short message service Co-ordinator for the SGSN.....	739
23.2	The mobile originated short message transfer procedure.....	741
23.2.1	Procedure in the servicing MSC.....	742
23.2.2	Procedure in the VLR.....	747
23.2.3	Procedure in the interworking MSC.....	748
23.2.4	Procedure in the servicing SGSN.....	751
23.3	The mobile terminated short message transfer procedure.....	755
23.3.1	Procedure in the Servicing MSC.....	757
23.3.2	Procedures in the VLR.....	766
23.3.3	Procedures in the HLR.....	770

23.3.4	Procedures in the gateway MSC.....	779
23.3.5	Procedure in the Servicing SGSN	789
23.4	The Short Message Alert procedure	797
23.4.1	Procedures in the Servicing MSC	799
23.4.2	Procedures in the VLR	801
23.4.2.1	The Mobile Subscriber is present	801
23.4.2.2	The Mobile Equipment has memory available	801
23.4.3	Procedures in the HLR	803
23.4.4	Procedures in the Interworking MSC	806
23.4.5	Procedures in the Servicing SGSN.....	808
23.4.5.1	The Mobile Subscriber is present	808
23.4.5.2	The Mobile Equipment has memory available	808
23.5	The SM delivery status report procedure	810
23.5.1	Procedures in the HLR	810
23.5.2	Procedures in the gateway MSC.....	812
23.6	Common procedures for the short message clause.....	814
23.6.1	The macro Report_SM_Delivery_Stat_HLR.....	814
24	GPRS process description	816
24.1	General.....	816
24.1.1	Process in the HLR for Send Routing Information for GPRS	816
24.1.2	Process in the GGSN for Send Routing Information for GPRS	819
24.2.1	Process in the HLR for Failure Report	821
24.2.2	Process in the GGSN for Failure Report	824
24.3.1	Process in the GGSN for Note Ms Present For Gprs	826
24.3.2	Process in the HLR for Note Ms Present For Gprs	829
24A	CSE control of subscriber data	831
24A.1	Any Time Subscription Interrogation procedure	831
24A.1.1	General.....	831
24A.1.2	Process in the gsmSCF.....	831
24A.1.3	Process in the HLR.....	831
24A.2	Any Time Modification procedure	834
24A.2.1	General.....	834
24A.2.2	Process in the gsmSCF.....	834
24A.2.3	Process in the HLR.....	834
24A.3	Subscriber Data Modification Notification procedure.....	837
24A.3.1	General.....	837
24A.3.2	Processes in the MAP Entities.....	838
24A.3.2.1	Process in the HLR.....	838
24A.3.2.2	Process in the gsmSCF	839
25	General macro description.....	840
25.1	MAP open macros	840
25.1.1	Macro Receive_Open_Ind	840
25.1.2	Macro Receive_Open_Cnf.....	841
25.2	Macros to check the content of indication and confirmation primitives	845
25.2.1	Macro Check_Indication.....	845
25.2.2	Macro Check_Confirmation.....	845
25.3	The page and search macros	848
25.3.1	Macro PAGE_MSC	848
25.3.2	Macro Search_For_MS_MSC.....	849
25.4	Macros for handling an Access Request	852
25.4.1	Macro Process_Access_Request_MSC.....	852
25.4.2	Macro Process_Access_Request_VLR	857
25.4.3	Macro Identification Procedure.....	859
25.5	Authentication macros and processes	864
25.5.1	Macro Authenticate_MSC.....	864
25.5.2	Macro Authenticate_VLR	864
25.5.3	Process Obtain_Authentication_Sets_VLR.....	864
25.5.4	Macro Obtain_Authent_Para_VLR.....	865

25.5.5	Process Obtain_Auth_Sets_HLR	866
25.5.6	Process Obtain_Authent_Para_SGSN.....	876
25.6	IMEI Handling Macros.....	878
25.6.1	Macro Check_IMEI_MSC	879
25.6.2	Macro Check_IMEI_VLR.....	879
25.6.3	Process Check_IMEI_EIR	880
25.6.4	Macro Obtain_IMEI_MSC	880
25.6.5	Macro Obtain_IMEI_VLR.....	880
25.6.6	Process Check_IMEI_SGSN.....	887
25.7	Insert Subscriber Data Macros.....	890
25.7.1	Macro Insert_Subs_Data_VLR.....	890
25.7.2	Process Insert_Subs_Data_Stand_Alone_HLR.....	892
25.7.3	Macro Wait_for_Insert_Subs_Data_Cnf.....	898
25.7.4	Process Send_Insert_Subs_Data	900
25.7.5	Macro Insert_Subs_Data_SGSN.....	902
25.7.6	Macro Wait_for_Insert_GPRS_Subs_Data_Cnf.....	904
25.8	Request IMSI Macros	906
25.8.1	Macro Obtain_IMSI_MSC.....	906
25.8.2	Macro Obtain_IMSI_VLR	908
25.9	Tracing macros	910
25.9.1	Macro Trace_Subscriber_Activity_MSC.....	910
25.9.2	Macro Trace_Subscriber_Activity_VLR	912
25.9.3	Macro Activate_Tracing_VLR.....	914
25.9.4	Macro Control_Tracing_HLR.....	916
25.9.5	Macro Trace_Subscriber_Activity_SGSN	919
25.9.6	Macro Activate_Tracing_SGSN	921
25.10	Short Message Alert procedures	923
25.10.1	Subscriber_Present_VLR process	923
25.10.2	Macro Alert_Service_Centre_HLR.....	925
25.10.3	The Mobile Subscriber is present.....	928
Annex A (informative): Cross-reference for abstract syntaxes of MAP		930
Annex B (informative): Fully expanded ASN.1 sources for abstract syntaxes of MAP.....		1127
B.1	Fully Expanded ASN.1 Source of MAP-Protocol/TCAPMessages	1127
B.2	Fully Expanded ASN.1 Source of MAP-DialogueInformation.....	1209
Annex C (informative): Formal protocol incompatibilities between versions 1 & 2 of MAP		1213
C.1	Introduction.....	1213
C.2	Deletion of operations and errors	1213
C.2.1	Deletion of operation DeregisterMobileSubscriber	1213
C.2.2	Deletion of operation RegisterChargingInfo.....	1213
C.2.3	Deletion of operation ForwardSS-Notification	1213
C.2.4	Deletion of operations used only on the B-interface.....	1213
C.2.5	Deletion of error InsufficientBearerCapabilities.....	1213
C.3	Deletion of errors for operations	1214
C.3.1	Error NegativePW-Check for operation RegisterSS	1214
C.3.2	Error NegativePW-Check for operation EraseSS	1214
C.3.3	Error NegativePW-Check for operation InterrogateSS.....	1214
C.3.4	Error CUG-Reject for operation SendRoutingInfoForSM.....	1214
C.4	Changes to definitions of data types.....	1214
C.4.1	CUG-Feature.....	1214
C.4.2	CUG-FeatureList	1214
C.4.3	CUG-Info.....	1214
C.4.4	CUG-RejectCause.....	1214
C.4.5	IMSI.....	1215
C.4.6	ISDN-AddressString.....	1215

C.4.7	Password.....	1215
C.4.8	RequestParameter.....	1215
C.4.9	RequestParameterList.....	1215
C.4.10	SentParameter.....	1215
C.4.11	SentParameterList.....	1215
C.4.12	SS-Data.....	1215
C.4.13	SS-Info.....	1215
C.4.14	SS-InfoList.....	1216
C.4.15	SS-SubscriptionOption.....	1216
C.4.16	SubscriberData.....	1216
C.5	Changes to parameters of errors.....	1216
C.5.1	CUG-Reject.....	1216
C.5.2	SS-SubscriptionViolation.....	1216
C.6	Changes to parameters of operations.....	1217
C.6.1	InsertSubscriberData.....	1217
C.6.2	RegisterSS.....	1217
C.6.3	SendParameters.....	1217
C.6.4	SendRoutingInfoForSM.....	1217
C.7	Changes to results of operations.....	1217
C.7.1	ActivateSS.....	1217
C.7.2	DeactivateSS.....	1218
C.7.3	EraseSS.....	1218
C.7.4	GetPassword.....	1218
C.7.5	InterrogateSS.....	1218
C.7.6	RegisterSS.....	1218
C.7.7	SendParameters.....	1219
C.7.8	SendRoutingInfoForSM.....	1219
C.8	Changes to errors of operations.....	1219
C.8.1	ActivateSS.....	1219
C.8.2	DeactivateSS.....	1219
C.8.3	EraseSS.....	1219
C.8.4	RegisterSS.....	1220
C.8.5	SendRoutingInfo.....	1220
Annex D (informative):	Clause mapping table.....	1221
D.1	Mapping of Clause numbers.....	1221
Annex E (informative):	Change History.....	1222
History.....		1225

1 Foreword

- 2 This Technical Specification has been produced by the 3GPP.
- 3 This TS specifies the Mobile Application Part (MAP), the requirements for the signalling system and procedures within
- 4 the 3GPP system at application level.
- 5 The contents of the present document are subject to continuing work within the TSG and may change following formal
- 6 TSG approval. Should the TSG modify the contents of this TS, it will be re-released by the TSG with an identifying
- 7 change of release date and an increase in version number as follows:
- 8 Version 3.y.z
- 9 where:
- 10 x the first digit:

- 11 1 presented to TSG for information;
- 12 2 presented to TSG for approval;
- 13 3 indicates TSG approved document under change control.
- 14 y the second digit is incremented for all changes of substance, i.e. technical enhancements, corrections, updates,
15 etc.
- 16 z the third digit is incremented when editorial only changes have been incorporated in the specification.

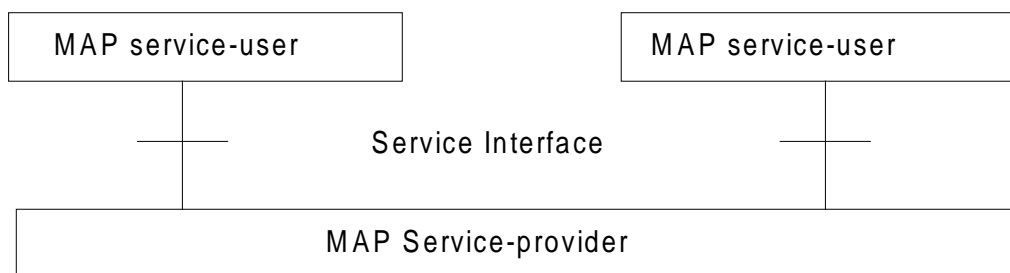
17 1 Scope

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

21 This Technical Specification (TS) describes the requirements for the signalling system and the procedures needed at the
22 application level in order to fulfil these signalling needs.

23 Clauses 1 to 6 are related to general aspects such as terminology, mobile network configuration and other protocols
24 required by MAP.

25 MAP consists of a set of MAP services which are provided to MAP service-users by a MAP service-provider.



26

27

Figure 1.1/1: Modelling principles

28 Clauses 7 to 12 of the present document describe the MAP services.

29 Clauses 14 to 17 define the MAP protocol specification and the behaviour of service provider (protocol elements to be
30 used to provide MAP services, mapping on to TC service primitives, abstract syntaxes, etc.).

31 Clauses 18 to 25 describe the MAP user procedures which make use of MAP services.

32 2 References

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

- 35 • References are either specific (identified by date of publication, edition number, version number, etc.) or
36 non-specific.
- 37 • For a specific reference, subsequent revisions do not apply.
- 38 • For a non-specific reference, the latest version applies.
- 39 • A non-specific reference to an ETS shall also be taken to refer to later versions published as an EN with the same
40 number.
- 41 • For this Release 1999 document, references to GSM documents are for Release 1999 versions (version 3.x.y).

42 [1] 3G TS 21.905: "3G Vocabulary".

43 [2] GSM 02.01: "Digital cellular telecommunications system (Phase 2+); Principles of
44 telecommunication services supported by a GSM Public Land Mobile Network (PLMN)".

45 [3] 3G TS 22.002: "Bearer Services Supported by a GSM Public Land Mobile Network (PLMN)".

46 [4] GSM 02.03: "Digital cellular telecommunications system (Phase 2+); Teleservices Supported by a
47 GSM Public Land Mobile Network (PLMN)".

- 48 [5] 3G TS 22.004: "General on Supplementary Services".
- 49 [6] GSM 02.09: "Digital cellular telecommunications system (Phase 2+); Security aspects".
- 50 [7] 3G TS 22.016: "International Mobile station Equipment Identities (IMEI)".
- 51 [8] 3G TS 22.041: "Operator Determined Barring".
- 52 [9] 3G TS 22.081: "Line identification supplementary services - Stage 1".
- 53 [10] 3G TS 22.082: "Call Forwarding (CF) supplementary services - Stage 1".
- 54 [11] 3G TS 22.083 : "Call Waiting (CW) and Call Hold (HOLD) Supplementary Services - Stage 1".
- 55 [12] 3G TS 22.084: "Multi Party (MPTY) Supplementary Services - Stage 1".
- 56 [13] 3G TS 22.085: "Closed User Group (CUG) supplementary services - Stage 1".
- 57 [14] 3G TS 22.086: "Advice of charge (AoC) Supplementary Services - Stage 1".
- 58 [15] 3G TS 22.088: "Call Barring (CB) supplementary services - Stage 1".
- 59 [16] 3G TS 22.090: "Unstructured Supplementary Service Data (USSD); - Stage 1".
- 60 [17] 3G TS 23.003: "Numbering, addressing and identification".
- 61 [18] GSM 03.04: "Digital cellular telecommunications system (Phase 2+); Signalling requirements
62 relating to routeing of calls to mobile subscribers".
- 63 [19] 3G TS 23.007: "Restoration procedures".
- 64 [20] 3G TS 23.008: "Organisation of subscriber data".
- 65 [21] 3G TS 23.009: "Handover procedures".
- 66 [22] 3G TS 23.011: "Technical realization of Supplementary Services - General Aspects".
- 67 [23] 3G TS 23.012: "Location registration procedures".
- 68 [24] GSM 03.20: "Digital cellular telecommunications system (Phase 2+); Security related network
69 functions".
- 70 [25] 3G TS 23.038: "Alphabets and language".
- 71 [26] 3G TS 23.040: "Technical realization of the Short Message Service (SMS) Point to Point (PP)".
- 72 [26a] GSM 03.71: "Digital cellular telecommunications system (Phase 2+); Location Services (LCS);
73 Functional Description; Stage 2".
- 74 [27] 3G TS 23.081: "Line Identification Supplementary Services - Stage 2".
- 75 [28] 3G TS 23.082: "Call Forwarding (CF) Supplementary Services - Stage 2".

- 76 [29] 3G TS 23.083: "Call Waiting (CW) and Call Hold (HOLD) Supplementary Services - Stage 2".
- 77 [30] 3G TS 23.084: "Multi Party (MPTY) Supplementary Services - Stage 2".
- 78 [31] 3G TS 23.085: "Closed User Group (CUG) Supplementary Services - Stage 2".
- 79 [32] 3G TS 23.086: "Advice of Charge (AoC) Supplementary Services - Stage 2".
- 80 [33] 3G TS 23.088: "Call Barring (CB) Supplementary Services - Stage 2".
- 81 [34] 3G TS 23.090: "Unstructured Supplementary Services Data (USSD) - Stage 2".
- 82 [35] 3G TS 24.008: "Mobile Radio Interface Layer 3 specification; Core Network Protocols - Stage 3".
- 83 [36] 3G TS 24.010: "Mobile radio interface layer 3 Supplementary Services specification - General
84 aspects".
- 85 [37] 3G TS 24.011: "Point-to-Point (PP) Short Message Service (SMS) support on mobile radio
86 interface".
- 87 [37a] GSM 04.71: "Digital cellular telecommunications system (Phase 2+); Mobile radio interface layer
88 3 location services specification.
- 89 [38] 3G TS 24.080: "Mobile radio interface layer 3 supplementary services specification - Formats and
90 coding".
- 91 [39] 3G TS 24.081: "Line identification supplementary services - Stage 3".
- 92 [40] 3G TS 24.082: "Call Forwarding (CF) Supplementary Services - Stage 3".
- 93 [41] 3G TS 24.083: "Call Waiting (CW) and Call Hold (HOLD) supplementary services - Stage 3".
- 94 [42] 3G TS 24.084: "Multi Party (MPTY) Supplementary Services - Stage 3".
- 95 [43] 3G TS 24.085: "Closed User Group (CUG) Supplementary Services - Stage 3".
- 96 [44] 3G TS 24.086: "Advice of Charge (AoC) Supplementary Services - Stage 3".
- 97 [45] 3G TS 24.088: "Call Barring (CB) Supplementary Services - Stage 3".
- 98 [46] 3G TS 24.090: "Unstructured Supplementary Services Data - Stage 3".
- 99 [47] GSM 08.02: "Digital cellular telecommunications system (Phase 2+); Base Station System -
100 Mobile-services Switching Centre (BSS - MSC) interface Interface principles".
- 101 [48] GSM 08.06: "Digital cellular telecommunications system (Phase 2+); Signalling transport
102 mechanism specification for the Base Station System - Mobile-services Switching Centre (BSS -
103 MSC) interface".
- 104 [49] GSM 08.08: "Digital cellular telecommunications system (Phase 2+); Mobile Switching Centre -
105 Base Station System (MSC - BSS) interface Layer 3 specification".

- 106 [49a] GSM 08.08: "Digital cellular telecommunications system (Phase 2+); Mobile Switching Centre -
107 Base Station System (MSC - BSS) interface Layer 3 specification".
- 108 [49a1] GSM 08.31: "Digital cellular telecommunications system (Phase 2+); Location Services (LCS);
109 Serving Mobile Location Center (SMLC) – Serving Mobile Location Center (SMLC); SMLC Peer
110 Protocol (SMLCPP)."
- 111 [49b] GSM 08.71: "Digital cellular telecommunications system (Phase 2+); Location Services (LCS);
112 Serving Mobile Location Centre - Base Station System (SMLC - BSS) interface Layer 3
113 specification".
- 114 [50] GSM 09.01: "Digital cellular telecommunications system (Phase 2+); General network
115 interworking scenarios".
- 116 [51] 3G TS 29.002: "Mobile Application Part (MAP) specification".
- 117 [52] GSM 09.03: "Digital cellular telecommunications system (Phase 2+); Signalling requirements on
118 interworking between the Integrated Services Digital Network (ISDN) or Public Switched
119 Telephone Network (PSTN) and the Public Land Mobile Network (PLMN)".
- 120 [53] GSM 09.04: "Digital cellular telecommunications system (Phase 2+); Interworking between the
121 Public Land Mobile Network (PLMN) and the Circuit Switched Public Data Network (CSPDN)".
- 122 [54] GSM 09.05: "Digital cellular telecommunications system (Phase 2+); Interworking between the
123 Public Land Mobile Network (PLMN) and the Packet Switched Public Data Network (PSPDN) for
124 Packet Assembly/Disassembly facility (PAD) access".
- 125 [55] 3G TS 29.006: "Interworking between a Public Land Mobile Network (PLMN) and a Packet
126 Switched Public Data Network/Integrated Services Digital Network (PSPDN/ISDN) for the
127 support of Packet Switched data transmission services".
- 128 [56] 3G TS 29.007: "Digital cellular telecommunications system (Phase 2+); General requirements on
129 interworking between the Public Land Mobile Network (PLMN) and the Integrated Services
130 Digital Network (ISDN) or Public Switched Telephone Network (PSTN)".
- 131 [57] GSM 09.08: "Digital cellular telecommunications system (Phase 2+); Application of the Base
132 Station System Application Part (BSSAP) on the E-interface".
- 133 [58] 3G TS 29.010: "Information element mapping between Mobile Station - Base Station System and
134 BSS - Mobile-services Switching Centre (MS - BSS - MSC) Signalling procedures and the Mobile
135 Application Part (MAP)".
- 136 [59] 3G TS 29.011: "Signalling interworking for Supplementary Services".
- 137 [59a] GSM 09.31: "Digital cellular telecommunications system (Phase 2+); Location Services (LCS);
138 Base Station System Application Part LCS Extension (BSSAP-LE)".
- 139 [60] GSM 09.90: "Digital cellular telecommunications system (Phase 2+); Interworking between Phase
140 1 infrastructure and Phase 2 Mobile Stations (MS)".
- 141 [61] GSM 12.08: "Digital cellular telecommunications system (Phase 2); Subscriber and Equipment
142 Trace".
- 143 [62] ETS 300 102-1 (1990): "Integrated Services Digital Network (ISDN); User-network interface layer
144 3 specifications for basic call control".

- 145 [63] ETS 300 136 (1992): "Integrated Services Digital Network (ISDN); Closed User Group (CUG)
146 supplementary service description".
- 147 [64] ETS 300 138 (1992): "Integrated Services Digital Network (ISDN); Closed User Group (CUG)
148 supplementary service Digital Subscriber Signalling System No.one (DSS1) protocol".
- 149 [65] ETS 300 287: "Integrated Services Digital Network (ISDN); Signalling System No.7; Transaction
150 Capabilities (TC) version 2".
- 151 [66] ETR 060: "Signalling Protocols and Switching (SPS); Guide-lines for using Abstract Syntax
152 Notation One (ASN.1) in telecommunication application protocols".
- 153 [67] ITU-T Recommendation E.164: "Numbering plan for the ISDN era".
- 154 [68] ITU-T Recommendation E.212: "Identification plan for land mobile stations".
- 155 [69] ITU-T Recommendation E.213: "Telephone and ISDN numbering plan for land mobile stations".
- 156 [70] ITU-T Recommendation E.214: "Structuring of the land mobile global title for the signalling
157 connection control part".
- 158 [71] CCITT Recommendation Q.669: "Interworking between the Digital Subscriber Signalling System
159 Layer 3 protocol and the Signalling System No.7 ISDN User part".
- 160 [72] ITU-T Recommendation Q.711: "Specifications of Signalling System No.7; Functional description
161 of the Signalling Connection Control Part".
- 162 [73] ITU-T Recommendation Q.712: "Definition and function of SCCP messages".
- 163 [74] ITU-T Recommendation Q.713: "Specifications of Signalling System No.7; SCCP formats and
164 codes".
- 165 [75] ITU-T Recommendation Q.714: "Specifications of Signalling System No.7; Signalling Connection
166 Control Part procedures".
- 167 [76] ITU-T Recommendation Q.716: "Specifications of Signalling System No.7; Signalling connection
168 control part (SCCP) performances".
- 169 [77] ITU-T Recommendation Q.721 (1988): "Specifications of Signalling System No.7; Functional
170 description of the Signalling System No.7 Telephone user part".
- 171 [78] ITU-T Recommendation Q.722 (1988): "Specifications of Signalling System No.7; General
172 function of Telephone messages and signals".
- 173 [79] ITU-T Recommendation Q.723 (1988): "Specifications of Signalling System No.7; Formats and
174 codes".
- 175 [80] ITU-T Recommendation Q.724 (1988): "Specifications of Signalling System No.7; Signalling
176 procedures".
- 177 [81] ITU-T Recommendation Q.725 (1988): "Specifications of Signalling System No.7; Signalling
178 performance in the telephone application".

- 179 [82] ITU-T Recommendation Q.761 (1988): "Specifications of Signalling System No.7; Functional
180 description of the ISDN user part of Signalling System No.7".
- 181 [83] ITU-T Recommendation Q.762 (1988): "Specifications of Signalling System No.7; General
182 function of messages and signals".
- 183 [84] ITU-T Recommendation Q.763 (1988): "Specifications of Signalling System No.7; Formats and
184 codes".
- 185 [85] ITU-T Recommendation Q.764 (1988): "Specifications of Signalling System No.7; Signalling
186 procedures".
- 187 [86] ITU-T Recommendation Q.767: "Specifications of Signalling System No.7; Application of the
188 ISDN user part of CCITT signalling System No.7 for international ISDN interconnections".
- 189 [87] ITU-T Recommendation Q.771: "Specifications of Signalling System No.7; Functional description
190 of transaction capabilities".
- 191 [88] ITU-T Recommendation Q.772: "Specifications of Signalling System No.7; Transaction
192 capabilities information element definitions".
- 193 [89] ITU-T Recommendation Q.773: "Specifications of Signalling System No.7; Transaction
194 capabilities formats and encoding".
- 195 [90] ITU-T Recommendation Q.774: "Specifications of Signalling System No.7; Transaction
196 capabilities procedures".
- 197 [91] ITU-T Recommendation Q.775: "Specifications of Signalling System No.7; Guide-lines for using
198 transaction capabilities".
- 199 [92] ITU-T Recommendation X.200: "Reference Model of Open systems interconnection for CCITT
200 Applications".
- 201 [93] ITU-T Recommendation X.208 (1988): "Specification of Abstract Syntax Notation One (ASN.1)".
- 202 [94] ITU-T Recommendation X.209 (1988): "Specification of basic encoding rules for Abstract Syntax
203 Notation One (ASN.1)".
- 204 [95] ITU-T Recommendation X.210: "Open systems interconnection layer service definition
205 conventions".
- 206 [97] 3G TS 23.018: "Basic Call Handling".
- 207 [98] 3G TS 23.078: " Customised Applications for Mobile network Enhanced Logic (CAMEL) Phase 3
208 - Stage 2".
- 209 [99] 3G TS 23.079: "Support of Optimal Routeing (SOR) - Stage 2".
- 210 [100] GSM 03.68: "Digital cellular telecommunications system (Phase 2+); - Stage 2".
- 211 [101] GSM 03.69: "Digital cellular telecommunications system (Phase 2+); - Stage 2".
- 212 [102] ANSI T1.113: "Signaling System No. 7 (SS7) - ISDN User Part".

- 213 [103] 3G TS 23.054 "Shared Inter Working Function (SIWF) - Stage 2".
- 214 [104] 3G TS 23.060: "General Packet Radio Service (GPRS) Description; Stage 2".
- 215 [105] 3G TS 29.060: "General Packet Radio Service (GPRS); GPRS Tunnelling Protocol (GTP) across
216 the Gn and Gp Interface".
- 217 [106] 3G TS 29.018: "General Packet Radio Service (GPRS); Serving GPRS Support Node (SGSN) -
218 Visitors Location Register (VLR); Gs interface layer 3 specification".
- 219 [107] 3G TS 23.093: "Technical Realization of Completion of Calls to Busy Subscriber (CCBS); Stage
220 2".
- 221 [108] 3G TS 23.066: "Support of Mobile Number Portability (MNP); Technical Realisation Stage 2".
- 222 [109] ANSI T1.112 (1996): "Telecommunication – Signalling No. 7 – Signaling Connection Control
223 Part (SCCP)".
- 224 [110] 3G TS 23.116: "Super-Charger Technical Realisation; Stage 2."
- 225 [111] ITU-T Recommendation Q.711: "Specifications of Signalling System No.7; Signalling System No.
226 7 – Functional Description of the Signalling Connection Control Part".
- 227 [112] ITU-T Recommendation Q.712: "Specifications of Signalling System No.7; Signalling System No.
228 7 – Definition and Function of SCCP Messages".
- 229 [113] ITU-T Recommendation Q.713: "Specifications of Signalling System No.7; Signalling System No.
230 7 – SCCP formats and codes".
- 231 [114] ITU-T Recommendation Q.714: "Specifications of Signalling System No.7; Signalling System No.
232 7 – Signalling Connection Control Part Procedures".
- 233 [115] ITU-T Recommendation Q.716: "Specifications of Signalling System No.7; Signalling System No.
234 7 – Signalling Connection Control Part (SCCP) Performance".
- 235 [116] ITU-T Q.850, May 1998: "Usage of cause and location in the Digital Subscriber Signalling System
236 No. 1 and the Signalling System No. 7 ISDN User Part".
- 237

238 3 Abbreviations

239 Abbreviations used in the present document are listed in 3G TS 21.905.

240 4 Configuration of the mobile network

241 4.1 The entities of the mobile system

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

245 4.1.1 The Home Location Register (HLR)

246 This functional entity is a data base in charge of the management of mobile subscribers. A PLMN may contain one or
247 several HLRs; it depends on the number of mobile subscribers, on the capacity of the equipment and on the organization
248 of the network. All subscription data are stored there. The main information stored there concerns the location of each

249 MS in order to be able to route calls to the mobile subscribers managed by each HLR. All management interventions
250 occur on this data base. The HLRs have no direct control of MSCs.

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

- 252 - IMSI;
- 253 - MSISDN.

254 The data base contains other information such as:

- 255 - location information (VLR number);
- 256 - basic telecommunication services subscription information;
- 257 - service restrictions (e.g. roaming limitation);
- 258 - supplementary services; the tables contain the parameters attached to these services.
- 259 - GPRS subscription data and routing information.

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

261 4.1.2 The Visitor Location Register (VLR)

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

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

- 269 - the IMSI;
- 270 - the MSISDN;
- 271 - the TMSI, if applicable;
- 272 - the location area where the MS has been registered. This will be used to call the station;
- 273 - supplementary service parameters.

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

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

276 4.1.3 The Mobile-services Switching Centre (MSC)

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

- 281 - procedures required for the location registration (see GSM 03.12);
- 282 - procedures required for hand-over (see GSM 03.09).

283 4.1.4 The Base Station System (BSS)

284 The BSS is the sub-system of Base Station equipment (transceivers, controllers, etc...) which is viewed

285 - by the MSC through an interface (A-interface) with the functionality described in GSM 08.02;

286 - by the SGSN through an interface (Gb-interface) with the functionality described in GSM 03.60.

287 4.1.5 The Gateway MSC (GMSC)

288 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
289 MSC. This MSC will interrogate the appropriate HLR and then route the call to the MSC where the MS is located. The
290 MSC which then performs the routing function to the actual location of the mobile is called the Gateway MSC.

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

293 If the call is a voice group/broadcast call it is routed directly from the GMSC to the VBS/VGCS Anchor MSC, based on
294 information (VBS/VGCS call reference) contained in the dialled number. See also GTSs 03.68 and 03.69.

295 See also GSM 03.04.

296 4.1.6 The SMS Gateway MSC

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

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

301 4.1.7 The SMS Interworking MSC

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

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

306 4.1.8 The VBS/VGCS Anchor MSC

307 The voice broadcast/group call anchor MSC obtains from the associated GCR all relevant attributes and controls in turn
308 all cells in its area, VBS/VGCS Relay-MSCs and dispatchers belonging to a given group call.

309 4.1.9 The Equipment Identity Register (EIR)

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

312 4.1.10 The GSM Service Control Function (gsmSCF)

313 This functional entity contains the CAMEL service logic to implement OSS. It interfaces with the gsmSSF, gprsSSF,
314 the HLR, VLR and MSC; see also 3G TS 23.078.

315 4.1.11 The VBS/VGCS Relay MSC

316 The voice broadcast/group call relay MSC obtains from the associated anchor MSC all relevant attributes and controls
317 in turn all cells in its area belonging to a given group call.

318 4.1.12 The Group Call Register (GCR)

319 This functional unit is a data base in charge of the management of attributes related to the establishment of Voice
320 Broadcast Calls and Voice Group Calls

321 4.1.13 The Shared InterWorking Function Server (SIWFS)

322 A Shared Inter Working Function is a network function that may be used by any MSC in the same PLMN to provide
323 interworking for a data/fax call. Whereas an IWF can only be used by its MSC, the SIWF can be used by several other
324 network nodes e.g. any MSC within the same PLMN (the concept is not limited to a certain number of MSCs). SIWF is
325 applied to data services in GSM Phase 2 and GSM Phase 2+ (as defined in GSM 02.02, GSM 02.03 and GSM 02.34).

326 The usage of an SIWF requires no additional manipulation at the MS.

327 An IWF provides specific functions associated with the visited MSC for the interworking with other networks. It
328 comprises signalling and traffic channel related functions. The traffic channel related functions are provided by an Inter
329 Working Unit (IWU).

330 The SIWF concept is that it provides specific functions for the interworking with other networks. It comprises signalling
331 and traffic channel related functions. Whereas the signalling related functions are associated with the visited MSC, the
332 IWU providing the traffic channel related functions has another physical location.

333 The entity that contains all additional functions needed in the visited MSC to provide the SIWF is called SIWF
334 Controller (SIWFC). The entity where the IWU is located is called SIWF Server (SIWFS). The Interface between a
335 visited MSC and a SIWFS is called the K Interface.

336 SIWFS can be provided by a MSC (MSC/SIWFS) or by another network entity (stand alone SIWFS).

337 4.1.14 The Serving GPRS Support Node (SGSN)

338 This functional unit keeps track of the individual MSs' location and performs security functions and access control; see
339 also GSM 03.60.

340 4.1.15 The Gateway GPRS Support Node (GGSN)

341 This functional unit provides interworking with external packet-switched networks, network screens and routing of the
342 Network Requested PDP-context activation; see also GSM 03.60.4.2 "Configuration of a Public Land Mobile Network
343 (PLMN)".

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

349 In this configuration, all the functions are considered implemented in different equipments. Therefore, all the interfaces
350 are external and need the support of the Mobile Application Part of the Signalling System No. 7 to exchange the data
351 necessary to support the mobile service. From this configuration, all the possible PLMN organizations can be deduced.

352 4.1.16 The Number Portability Location Register (NPLR)

353 This functional unit provides routing information necessary in some Mobile Number Portability environments in order
354 to route calls for ported mobile subscribers. For details see also GSM 03.66 [108].

355 4.1.17 The Serving Mobile Location Center (SMLC)

356 An SMLC is a database and processing entity that manages the procedures for obtaining the geographic location of a
357 target MS in the coverage area served by the SMLC. In managing the location procedures, the SMLC chooses the
358 positioning method and provides data and instructions to the LMUs or target MS that perform the actual location
359 measurements associated with the chosen method. The SMLC also verifies any location estimate computed by the target
360 MS or computes a location itself from measurements provided to it by the target MS or LMUs.

361 An SMLC also manages a set of LMUs in its coverage area whose purpose is to provide location measurements and
362 location assistance data to the SMLC to compute, or assist in computing, location estimates for target MSs. Management
363 functions performed by an SMLC on behalf of its LMUs include maintaining the status and current serving MSC of each
364 LMU and supporting O&M procedures,

365 The database in an SMLC contains data necessary for choosing an appropriate position method and any parameters
366 associated with this method for a target MS in any serving cell, for computing or verifying location estimates and for
367 managing its LMUs.

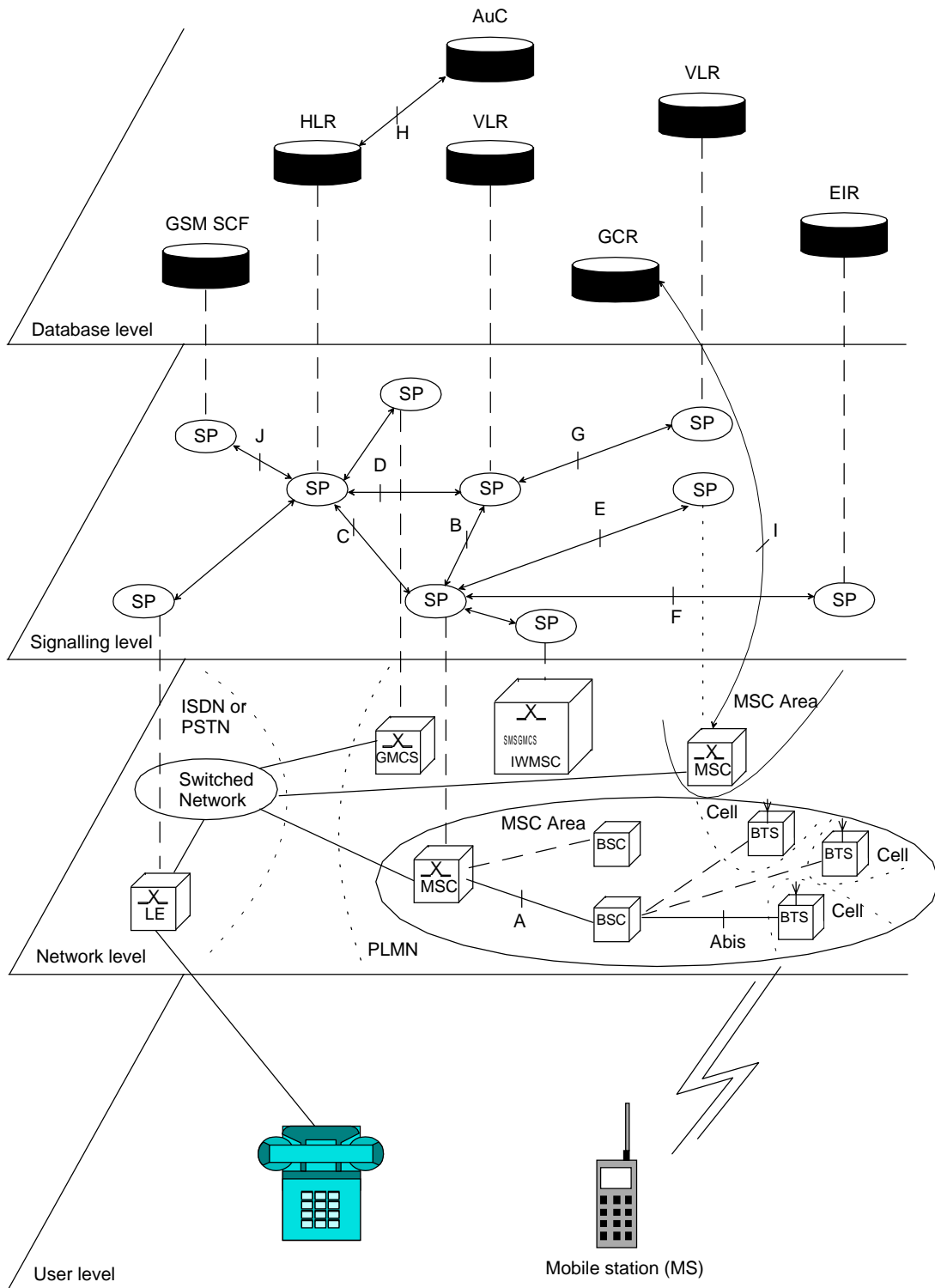
368 4.1.18 The Gateway Mobile Location Center (GMLC)

369 The GMLC provides access to location services (LCS) for LCS clients external to a PLMN. A GMLC may also support
370 access to location services from LCS clients internal to its own PLMN. The GMLC allows an LCS client to issue
371 location requests for certain target MSs; it then conveys these requests to the VMSC currently serving each target MS
372 and passes back the location results to the LCSclient. Any target MS whose location is requested may belong to either
373 the GMLC's own PLMN or another PLMN and may currently be served by either the GMLC's own PLMN or another
374 PLMN.

375 4.1.19 The Location Measurement Unit (LMU)

376 The LMU is the logical network entity that performs location measurements in the VPLMN in order to either position a
377 target MS or provide assistance data to be used in conjunction with other location measurements. An LMU is controlled
378 by an SMLC in the VPLMN from which location commands can be received and to which any location measurements
379 are returned.

380



381

382

Figure 4.2/1: Configuration of a PLMN

383 4.3 Interconnection between PLMNs

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

386 4.4 The interfaces within the mobile service

387 4.4.1 Interface between the HLR and the VLR (D-interface)

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

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

397 4.4.2 Interface between the HLR and the gsmSCF (J-interface)

398 This interface is used by the gsmSCF to request information from the HLR (via the Any-time Interrogation function) or
399 to allow call independent related network- or user-initiated interaction between an MS and the gsmSCF (via the USSD
400 function). Support of the gsmSCF-HLR interface is a network operator option. As a network operator option, the HLR
401 may refuse to provide the information requested by the gsmSCF.

402 4.4.3 Interface between the VLR and its associated MSC(s) (B-interface)

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

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

411 4.4.4 Interface between VLRs (G-interface)

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

414 4.4.5 Interface between the HLR and the MSC (C-interface)

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

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

420 4.4.6 Interface between the MSC and the gsmSCF (L-interface)

421 When one of the following Supplementary Services, CD, ECT or MPTY, is invoked in the MSC a notification shall be
422 sent towards the gsmSCF.

423 4.4.6A Interface between the VLR and the gsmSCF (M-interface)

424 This interface is used by the VLR to notify the gsmSCF about the occurrence of a Mobility Management event.

425 4.4.7 Interface between MSCs (E-interface)

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

429 This interface is also used to forward short messages, to perform location for a target MS for which handover has
430 occurred on an established call and to transfer LCS messages to and from an LMU for which handover of a signalling
431 channel has occurred.

432 This interface is also used to transfer information for inter-MSC VBS/VGCS calls .

433 4.4.8 Interface between the MSC and Base Station Systems (A-interface)

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

435 The BSS-MSC interface carries information concerning:

- 436 - BSS management;
- 437 - call handling;
- 438 - location management.

439 4.4.9 Interface between MSC and EIR (F-interface)

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

441 4.4.10 Interface between VBS/VGCS Anchor MSC and GCR (I-interface)

442 This is an internal interface.

443 4.4.11 Interface between the MSC and the SIWF server (K-interface)

444 When a MSC detects that it can not provide the requested IW function, resources from an SIWF server can be used.
445 This interface is used to allocate resources in that SIWF server and establish required physical connections to that
446 server.

447 4.4.12 Interface between SGSN and HLR (Gr-interface)

448 The description of this interface is contained in the GSM 03.60.

449 4.4.13 Interface between SGSN and SMS-GMSC or SMS-IW MSC (Gd- 450 interface)

451 The description of this interface is contained in the GSM 03.60.

452 4.4.14 Interface between GGSN and HLR (Gc-interface)

453 The description of this interface is contained in the GSM 03.60.

454 4.4.15 Interface between SGSN and EIR (Gf-interface)

455 The description of this interface is contained in the GSM 03.60.

456 4.4.16 Interface between SGSN and BSC (Gb-interface)

457 The description of this interface is contained in the GSM 03.60.

458 4.4.17 Interface between SGSN and MSC/VLR (Gs-interface)

459 The description of this interface is contained in the GSM 09.18.

460 4.4.18 Interface between SMLC and MSC (Ls interface)

461 This interface is used by the MSC to request either the initiation of location procedures or the retrieval of location
462 assistance data for a particular target MS in the coverage area served by the SMLC. The interface is also used to
463 transfer LCS measurement and O&M information between an SMLC and LMU or BSC via the MSC.

464 4.4.19 Interface between SMLC and VLR (Lv interface)

465 This interface is used by the VLR to register or deregister an LMU in the SMLC.

466 4.4.20 Interface between GMLC and HLR (Lh interface)

467 This interface is used by the GMLC to request the address of the visited MSC for a particular target MS whose location
468 has been requested.

469 4.4.21 Interface between GMLC and MSC (Lg interface)

470 This interface is used by the GMLC to convey a location request to the MSC currently serving a particular target MS
471 whose location was requested. The interface is used by the MSC to return location results to the GMLC.

472 4.4.22 Interface between LCS Client and GMLC (Le interface)

473 This interface is used by a client of the Location Services (LCS) to request location information from a GMLC for
474 certain target MSs. The interface is used by the GMLC to provide location information to an LCS client. This interface
475 is external to a PLMN and is not defined within GSM.

476 4.4.23 Interface between the gsmSCF and the GMLC (Lc-interface)

477 This interface is used by the gsmSCF to request information from the GMLC (via the Any-time Interrogation function).

478 As a network operator option, the GMLC may refuse to provide the information requested by the gsmSCF.

479 4.5 Splitting of the data storage

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

482 The data associated with any client that uses a particular GMLC to access location services is stored in the GMLC.

483 A detailed description of the data organization can be found in GSM 03.08.

484 5 Overload and compatibility overview

485 5.1 Overload control

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

488 5.1.1 Overload control for MSC (outside MAP)

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

- 490 - ISDN
- 491 CCITT Recommendation Q.764 (Automatic Congestion Control), applicable to reduce the mobile terminating
492 traffic;
- 493 - BSSAP
- 494 GSM 08.08 (A-interface Flow Control), applicable to reduce the mobile originating traffic.

495 5.1.2 Overload control for MAP entities

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

497 If overload of a MAP entity is detected requests for certain MAP operations (see tables 5.1/1, 5.1/2, 5.1/3 and 5.1/4)
498 may be ignored by the responder. The decision as to which MAP Operations may be ignored is made by the MAP
499 service provider and is based upon the priority of the application context.

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

502 If overload levels are applicable in the Location Registers the MAP operations should be discarded taking into account
503 the priority of their application context (see table 5.1/1 for HLR, table 5.1/2 for MSC/VLR, table 5.1/3 for the SGSN
504 and table 5.1/4 for the SMLC; the lowest priority is discarded first).

505 The ranking of priorities given in the tables 5.1/1, 5.1/2, 5.1/3 and 5.1/4 is not normative. The tables can only be seen as
506 a proposal which might be changed due to network operator/implementation matters.

507

Table 5.1/1: Priorities of Application Contexts for HLR as Responder

	Responder = HLR	Initiating Entity
508		
509	<i>Priority high</i>	
510	<u>Mobility Management</u>	
511	networkLocUp	VLR
512	(updateLocation),	
513	(restoreData/v2),	
514	(sendParameters/v1)	
515	gprsLocationUpdate	SGSN
516	(updateGPRSLocation/v3),	
517	infoRetrieval	VLR/SGSN
518	(sendAuthenticationInfo/v2/v3),	
519	(sendParameters/v1)	
520	istAlerting	MSC
521	(istAlert/v3)	msPurging VLR
522	(purgeMS/v2/v3)	
523		
524	msPurging	SGSN
525	(purgeMS/v3)	
526		
527	<u>Short Message Service</u>	
528	shortMsgGateway	GMSC
529	(sendRoutingInfoforSM),	
530	(reportSM-DeliveryStatus)	
531	mwdMngt VLR/SGSN	
532	(readyForSM/v2/v3),	
533	(noteSubscriberPresent/v1)	
534		
535	<u>Mobile Terminating Traffic</u>	
536	locInfoRetrieval	GMSC
537	(sendRoutingInfo)	
538	anyTimeEnquiry	gsmSCF
539	(anyTimeInterrogation)	
540	reporting	VLR
541	(statusReport)	
542		
543	<u>Location Services</u>	
544	locationSvcGateway	GMLC
545	(sendRoutingInfoforLCS/v3)	
546		
547	<u>Subscriber Controlled Inputs (Supplementary Services)</u>	
548	networkFunctionalSs	VLR
549	(registerSS),	
550	(eraseSS),	
551	(activateSS),	
552	(deactivateSS),	
553	(interrogateSS),	
554	(registerPassword),	
555	(processUnstructuredSS-Data/v1),	
556	(beginSubscriberActivity/v1)	
557	callCompletion	VLR
558	(registerCCEnter),	
559	(eraseCCEnter)	
560	networkUnstructuredSs	VLR
561	(processUnstructuredSS-Request/v2)	
562		
563	imsiRetrieval	VLR
564	(sendIMSI/v2)	
565	gprsLocationInfoRetrieval	GGSN/SGSN
566	(sendRoutingInfoForGprs/v3)	
567	failureReport	GGSN/SGSN
568	(failureReport/v3)	
569		
570	<i>Priority low</i>	

571 NOTE: The application context name is the last component but one of the object identifier.
 572 Operation names are given in brackets for information with "/vn" appended to vn only operations.
 573

574 **Table 5.1/3: Priorities of Application Contexts for SGSN as Responder**

575 Responder = SGSN	575 Initiating Entity
576 <i>Priority high</i>	
577 <u>Mobility and Location Register Management</u>	
578 locationCancel	HLR
579 (cancelLocation v3)	
580 reset	HLR
581 (reset)	
582 subscriberDataMngt	HLR
583 (insertSubscriberData v3),	
584 (deleteSubscriberData v3)	
585 tracing	HLR
586 (activateTraceMode),	
587 (deactivateTraceMode)	
588	
589 <u>Short Message Service</u>	
590 shortMsgMT-Relay	MSC
591 (MT-ForwardSM v3)	
592 (forwardSM v1/v2)	
593	
594 <u>Network-Requested PDP context activation</u>	
595 gprsNotify HLR	
596 (noteMsPresentForGprs v3),	
597	
598 <i>Priority low</i>	

599 NOTE: The application context name is the last component but one of the object identifier.
 600 Operation names are given in brackets for information with "/vn" appended to vn.
 601

602 **Table 5.1/2: Priorities of Application Contexts for MSC/VLR as Responder**

Responder = MSC/VLR	Initiating Entity
<i>Priority high</i>	
<u>Handover</u>	
handoverControl (prepareHandover/v2), (performHandover/v1)	MSC
<u>Mobility and Location Register Management</u>	
locationCancel (cancelLocation)	HLR
reset (reset)	HLR
immediateTermination (istCommand/v3)	HLR
interVlrInfoRetrieval (sendIdentification/v2/v3), (sendParameters/v1)	VLR
subscriberDataMngt (insertSubscriberData), (deleteSubscriberData)	HLR
tracing (activateTraceMode), (deactivateTraceMode)	HLR
<u>Short Message Service</u>	
shortMsgMO-Relay (MO-ForwardSM v3) (forwardSM v1/v2)	MSC/SGSN
shortMsgMT-Relay (MT-ForwardSM v3) (forwardSM v1/v2)	MSC
shortMsgAlert (alertServiceCentre/v2), (alertServiceCentreWithoutResult/v1)	HLR
<u>Mobile Terminating Traffic</u>	
roamingNbEnquiry (provideRoamingNumber)	HLR
callControlTransfer (resumeCallHandling)	MSC
subscriberInfoEnquiry (provideSubscriberInformation)	HLR
reporting (remoteUserFree) (SetReportingState)	HLR
<u>Location Services</u>	
locationSvcEnquiry (provideSubscriberLocation v3)	GMLC
<u>Network-Initiated USSD</u>	
networkUnstructuredSs (unstructuredSS-Request/v2), (unstructuredSS-Notify/v2)	HLR
<i>Priority low</i>	

603 NOTE: The application context name is the last component but one of the object identifier.
604 Operation names are given in brackets for information with "/vn" appended to vn only operations.
605

606 5.1.3 Congestion control for Signalling System No. 7

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

608 Means which could be applied to achieve the required traffic reductions are described in subclauses 5.1.1 and 5.1.2.

609 5.2 Compatibility

610 5.2.1 General

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

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

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

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

623 If the proposed application-context-name can be supported by the responding entity the dialogue continues on this basis
624 otherwise the dialogue is refused and the initiating user needs to start a new dialogue, which involves another
625 application-context-name which requires less communication capabilities but provides similar functionalities (if
626 possible).

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

631 5.2.2 Strategy for selecting the Application Context (AC) version

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

635 5.2.2.1 Proposed method

636 A table (table 1) may be set up by administrative action to define the highest application context (AC) version supported
637 by each destination; a destination may be another node within the same or a different PLMN, or another PLMN
638 considered as a single entity. The destination may be defined by an E.164 number or an E.214 number derived from an
639 IMSI or in North America (World Zone 1) by an E.164 number or an IMSI (E.212 number). The table also includes the
640 date when each destination is expected to be able to handle at least one AC of the latest version of the MAP protocol.
641 When this date is reached, the application context supported by the node is marked as "unknown", which will trigger the
642 use of table 2.

643 A second table (table 2) contains an entry for each destination which has an entry in table 1. For a given entity, the entry
644 in table 2 may be a single application context version or a vector of different versions applying to different application
645 contexts for that entity. Table 2 is managed as described in subclause 5.2.2.2.

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

- 647 a) the version shown in table 1 is "version n-1", where 'n' is the highest version existing in this specification; table 2
648 is not used;
- 649 b) the version shown in table 1 is "unknown"; table 2 is used, and maintained as described in subclause 5.2.2.2;
- 650 c) when the PLMN operator declares that an entity (single node or entire PLMN) has been upgraded to support all
651 the MAP version n ACs defined for the relevant interface, the version shown in table 1 is set to "version n" by
652 administrative action; table 2 is no longer used, and the storage space may be recovered.

653 5.2.2.2 Managing the version look-up table

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

657 **IF** the entity number is known

658 **THEN**

659 It updates (if required) the associated list of highest supported ACs

660 **ELSE**

661 It creates an entry for this entity and includes the received AC-name in the list of highest supported ACs.

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

663 **IF** the destination address is known and not timed-out

664 **THEN**

665 It retrieves the appropriate AC-name and uses it

666 **IF** the dialogue is accepted by the peer

667 **THEN**

668 It does not modify the version control table

669 **ELSE** (this should never occur)

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

672 It replace the old AC-name by the new one in the list of associated highest AC supported.

673 **ELSE**

674 It uses the AC-name which corresponds to the highest version it supports.

675 **IF** the dialogue is accepted by the peer

676 **THEN**

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

679 **ELSE**

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

682 **IF** the destination node was not known

683 **THEN**

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

686 **ELSE**

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

688 5.2.2.3 Optimizing the method

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

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

696 **for HLR:**

- 697 - Subscriber data modification (stand alone);

698 **for VLR:**

- 699 - Data Restoration.

700 6 Requirements concerning the use of SCCP and TC

701 6.1 Use of SCCP

702 The Mobile Application Part (MAP) makes use of the services offered by the Signalling Connection Control Part
703 (SCCP).

704 MAP supports the following SCCP versions;

- 705 • Signalling Connection Control Part , Signalling System no. 7 CCITT ('Blue Book SCCP')
- 706 • Signalling Connection Control Part , Signalling System no. 7 ITU-T Recommendation (07/96) Q.711 to Q.716
707 ('White Book SCCP'). Support of White Book SCCP at the receiving side shall be mandated from 00:01hrs, 1st
708 July 2002(UTC).

709 A White Book SCCP message will fail if any signalling point used in the transfer of the message does not support White
710 Book SCCP. Therefore it is recommended that the originator of the White Book SCCP message supports a drop back
711 mechanism or route capability determination mechanism to interwork with signalling points that are beyond the control
712 of GSM/UMTS network operators.

713 In North America (World Zone 1) the national version of SCCP is used as specified in ANSI T1.112. Interworking
714 between a PLMN in North America and a PLMN outside North America will involve an STP to translate between ANSI
715 SCCP and ITU-T/CCITT SCCP.

716 6.1.1 SCCP Class

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

718 6.1.2 Sub-System Number (SSN)

719 The Application Entities (AEs) defined for MAP consist of several Application Service Elements (ASEs) and are
720 addressed by sub-system numbers (SSNs). The SSN for MAP are specified in GSM 03.03 [17].

721 When the SGSN emulates MSC behavior for processing messages (MAP-MO-FORWARD-SHORT-MESSAGE,
722 MAP_CHECK_IMEI) towards entities which do not support interworking to SGSNs, it shall use the MSC SSN in the
723 calling party address instead of the SGSN SSN.

724 6.1.3 SCCP addressing

725 6.1.3.1 Introduction

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

729 Only the entities which should be addressed are described below. If the CCITT or ITU-T SCCP is used, the format and
730 coding of address parameters carried by the SCCP for that purpose shall comply with CCITT Recommendation Q.713
731 with the following restrictions:

732 1) Intra-PLMN addressing

733 For communication between entities within the same PLMN, a MAP SSN shall always be included in the called
734 and calling party addresses. All other aspects of SCCP addressing are network specific.

735 2) Inter-PLMN addressing

736 a) Called Party Address

- 737 - SSN indicator = 1 (MAP SSN always included);
- 738 - Global title indicator = 0100 (Global title includes translation type, numbering plan, encoding scheme and
739 nature of address indicator);
- 740 - the translation type field will be coded "00000000" (Not used). For call related messages for non-optimal
741 routed calls (as described in GSM 03.66) directed to another PLMN the translation type field may be coded
742 "10000000" (CRMNP);
- 743 - Routing indicator = 0 (Routing on global title);

744 b) Calling Party Address

- 745 - SSN indicator = 1 (MAP SSNs always included);
- 746 - Point code indicator = 0;
- 747 - Global title indicator = 0100 (Global title includes translation type, numbering plan, encoding scheme and
748 nature of address indicator);
- 749 - Numbering Plan = 0001 (ISDN Numbering Plan, E.164; In Case of Inter-PLMN Signalling, the dialogue
750 initiating entity and dialogue responding entity shall always include its own E.164 Global Title as Calling
751 Party Address);
- 752 - the translation type field will be coded "00000000" (Not used);
- 753 - Routing indicator = 0 (Routing on Global Title).

754 If ANSI T1.112 SCCP is used, the format and coding of address parameters carried by the SCCP for that purpose shall
755 comply with ANSI specification T1.112 with the following restrictions:

756 1) Intra-PLMN addressing

757 For communication between entities within the same PLMN, a MAP SSN shall always be included in the called
758 and calling party addresses. All other aspects of SCCP addressing are network specific.

- 759 2) Inter-PLMN addressing
- 760 a) Called Party Address
- 761 - SSN indicator = 1 (MAP SSN always included);
- 762 - Global title indicator = 0010 (Global title includes translation type);
- 763 - the Translation Type (TT) field will be coded as follows:
- 764 TT = 9, if IMSI is included,
- 765 TT = 14, if MSISDN is included,
- 766 Or TT = 10, if Network Element is included. (If TT=10, then Number Portability GTT is not invoked, if
- 767 TT=14, then Number Portability GTT may be invoked.)
- 768 - Routing indicator = 0 (Routing on global title);
- 769 b) Calling Party Address
- 770 - SSN indicator = 1 (MAP SSNs always included);
- 771 - Point code indicator = 0;
- 772 - Global Title indicator = 0010 (Global title includes translation type);
- 773 TT = 9, if IMSI is included,
- 774 TT = 14, if MSISDN is included,
- 775 Or TT = 10, if Network Element is included. (If TT=10, then Number Portability GTT is not invoked, if
- 776 TT=14, then Number Portability GTT may be invoked.)
- 777 Routing indicator = 0 (Routing on Global Title).
- 778 If a Global Title translation is required for obtaining routing information, one of the numbering plans E.164, E.212 and
- 779 E.214 is applicable.
- 780 - E.212 numbering plan
- 781 When CCITT or ITU-T SCCP is used, an E.212 number must not be included as Global Title in an SCCP
- 782 UNITDATA message. The translation of an E.212 number into a Mobile Global Title is applicable in a dialogue
- 783 initiating VLR, SGSN or GGSN if the routing information towards the HLR is derived from the subscriber's
- 784 IMSI. In World Zone 1 when ANSI SCCP is used, the IMSI (E.212 number) is used as a Global Title to address
- 785 the HLR. When an MS moves from one VLR service area to another, the new VLR may derive the address of the
- 786 previous VLR from the Location Area Identification provided by the MS in the location registration request. The
- 787 PLMN where the previous VLR is located is identified by the E.212 numbering plan elements of the Location
- 788 Area Identification, ie the Mobile Country Code (MCC) and the Mobile Network Code (MNC).
- 789 - E.214 and E.164 numbering plans
- 790 When CCITT or ITU-T SCCP is used, , only address information belonging to either E.214 or E.164 numbering
- 791 plan is allowed to be included as Global Title in the Called and Calling Party Address. In World Zone 1 when
- 792 ANSI SCCP is used, the IMSI (E.212 number) is used as a Global Title to address the HLR.
- 793 If the Calling Party Address associated with the dialogue initiating message contains a Global Title, the sending
- 794 network entity shall include its E.164 entity number.
- 795 When receiving an SCCP UNITDATA message, SCCP shall accept either of the valid numbering plans in the
- 796 Called Party Address and in the Calling Party Address.
- 797 When CCITT or ITU-T SCCP is used and an N-UNITDATA-REQUEST primitive from TC is received, SCCP
- 798 shall accept an E.164 number or an E.214 number in the Called Address and in the Calling Address. In World
- 799 Zone 1 when ANSI SCCP is used, the IMSI (E.212 number) is used instead of E.214 number.

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

- 802 - the Mobile-services Switching Centre (MSC);
- 803 - the Home location Register (HLR);
- 804 - the Visitor Location Register (VLR);
- 805 - the Gateway Mobile-services Switching Centre (GMSC);
- 806 - the GSM Service Control Function (gsmSCF);
- 807 - the Interworking Mobile-services Switching Centre (IWMSC);
- 808 - the Shared Inter Working Function (SIWF);
- 809 - the Serving GPRS Support Node (SGSN);
- 810 - the Gateway GPRS Support Node (GGSN);
- 811 - the Gateway Mobile Location Center (GMLC).

812 6.1.3.2 The Mobile-services Switching Centre (MSC)

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

814 6.1.3.2.1 MSC interaction during handover

815 The address is derived from the target Cellid.

816 6.1.3.2.2 MSC for short message routing

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

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

820 6.1.3.2.3 MSC for location request routing

821 When a location request for a particular MS needs to be sent to the MS's VMSC, the GMLC addresses the VMSC using
822 an E.164 address received from the MS's HLR.

823

824 6.1.3.2.4 MSC for LMU Control

825 When a control message has to be routed to an LMU from an SMLC, the SMLC addresses the serving MSC for the
826 LMU using an E.164 address.

827 6.1.3.3 The Home Location Register (HLR)

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

829 6.1.3.3.1 During call set-up

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

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

837 6.1.3.3.2 Before location updating completion

838 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
839 and a preceding dialogue for authentication information retrieval if the authentication information must be retrieved
840 from the HLR. When initiating either of these dialogues, the only data for addressing the HLR that the VLR has
841 available is contained in the IMSI, and addressing information for SCCP must be derived from it. When continuing the
842 established update location dialogue (as with any other dialogue), the VLR must derive the routing information towards
843 the HLR from the Calling Party Address received with the first responding CONTINUE message until the dialogue
844 terminating message is received. This means that the VLR must be able to address the HLR based:

- 845 - on an E.214 Mobile Global Title originally derived by the VLR from the IMSI (when CCITT or ITU-T SCCP is
846 used), or an E.212 number originally derived from IMSI (when ANSI SCCP is used, an IMSI); or
- 847 - on an E.164 HLR address; or
- 848 - in the case of intra-PLMN signalling, on an SPC.

849 When answering with Global Title to the VLR, the HLR shall insert its E.164 address in the Calling Party Address of the
850 SCCP message containing the first responding CONTINUE message.

851 If the HLR is in the same PLMN as the VLR, local translation tables may exist to derive an SPC. For authentication
852 information retrieval and location updating via the international PSTN/ISDN signalling network that requires the use of
853 CCITT or ITU-T SCCP, the Global title must be derived from the IMSI, using the principles contained in CCITT
854 Recommendation E.214 and the Numbering Plan Indicator (NPI) value referenced by the SCCP Specifications. In
855 World Zone 1 where the ANSI SCCP is used, IMSI (E.212 number) is used as Global Title. A summary of the
856 translation from the IMSI (CCITT Recommendation E.212) to Mobile Global Title (described in CCITT
857 Recommendation E.214) is shown below:

- 858 - E.212 Mobile Country Code translates to E.164 Country Code;
- 859 - E.212 Mobile Network Code translates to E.164 National Destination Code;
- 860 - E.212 Mobile Subscriber Identification Number (MSIN) is carried unchanged if within the E.164 number
861 maximum length (15 digits). If the Mobile Global Title is more than 15 digits the number is truncated to 15 by
862 deleting the least significant digits.

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

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

867 6.1.3.3.3 After location updating completion

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

874 Thus the SCCP address of the roaming subscriber's HLR may be an SPC, or it may be a Global title consisting of the
875 E.164 MSISDN or the E.164 number allocated to the HLR or either the E.214 Mobile Global Title derived from the
876 IMSI if CCITT or ITU-T SCCP is used, or the IMSI if ANSI SCCP is used (ANSI SCCP is used in World Zone 1).

877 6.1.3.3.4 VLR restoration

878 If a roaming number is requested by the HLR for an IMSI that has no data record in the interrogated VLR, the VLR
879 provides the roaming number in the dialogue terminating message. Subsequently the VLR must retrieve the
880 authentication data from the MS's HLR, if required, and must then trigger the restore data procedure. For this purpose,
881 the VLR has to initiate in succession two independent dialogues with the MS's HLR. The MTP and SCCP address
882 information needed for routing towards the HLR can be derived from the IMSI received as a parameter of the MAP
883 message requesting the roaming number. In this case, the IMSI received from the HLR in the roaming number request

884 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.
885 Alternatively to the IMSI, the Calling Party Address associated with the roaming number request may be used to obtain
886 the routing information towards the HLR.

887 6.1.3.3.5 During Network-Requested PDP Context Activation

888 When receiving a PDP PDU the GGSN may interrogate the HLR of the MS for information retrieval. When initiating
889 such a dialogue, the only data for addressing the HLR that the GGSN has available is contained in the IMSI, and
890 addressing information must be derived from it. The IMSI is obtained from the IP address or the X.25 address in the
891 incoming IP message by means of a translation table. This means that the GGSN shall be able to address the HLR based
892 on an E.214, (if CCITT or ITU-T SCCP is used), or E.212 (if ANSI SCCP is used), Mobile Global Title originally
893 derived by the GGSN from the IMSI in the case of inter-PLMN signalling. In the case of intra-PLMN signalling, an
894 SPC may also be used.

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

- 900 - E.212 Mobile Country Code translates to E.164 Country Code;
- 901 - E.212 Mobile Network Code translates to E.164 National Destination Code;
- 902 - E.212 Mobile Subscriber Identification Number (MSIN) is carried unchanged if within the E.164 number
903 maximum length (15 digits). If the Mobile Global Title is more than 15 digits the number is truncated to 15 by
904 deleting the least significant digits.

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

907 6.1.3.3.6 Before GPRS location updating completion

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

- 915 - on an E.214 (if CCITT or ITU-T SCCP is used) or E.212 (if ANSI SCCP is used) Mobile Global Title originally
916 derived by the SGSN from the IMSI; or
- 917 - on an E.164 HLR address; or
- 918 - in the case of intra-PLMN signalling, on an SPC.

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

- 924 - E.212 Mobile Country Code translates to E.164 Country Code;
- 925 - E.212 Mobile Network Code translates to E.164 National Destination Code;
- 926 - E.212 Mobile Subscriber Identification Number (MSIN) is carried unchanged if within the E.164 number
927 maximum length (15 digits). If the Mobile Global Title is more than 15 digits the number is truncated to 15 by
928 deleting the least significant digits.

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

931 6.1.3.3.7 After GPRS location updating completion

932 In this case, the subscriber's Basic MSISDN has been received from the HLR during the subscriber data retrieval
933 procedure as well as the HLR number constituting a parameter of the MAP message indicating successful completion of
934 the update location dialogue. From either of these E.164 numbers the address information for initiating dialogues with
935 the roaming subscriber's HLR can be derived. Also the subscriber's IMSI may be used for establishing the routing
936 information towards the HLR.

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

939 6.1.3.3.8 Query for a Location Request

940 For a location request from an external client, the GMLC needs to address the home HLR of the target MS to obtain the
941 address of the target MS's serving MSC. The GMLC uses either the international E.164 MSISDN, the international
942 E.214 number (if CCITT or ITU-T SCCP is used) or the international E.212 number (if ANSI SCCP is used) of the MS
943 as means to route a query to the HLR.

944 6.1.3.4 The Visitor Location Register (VLR)

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

946 6.1.3.4.1 Inter-VLR information retrieval

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

950 6.1.3.4.2 HLR request

951 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.
952 This means that a location updating dialogue initiated by the VLR has been successfully completed, i.e. the HLR has
953 indicated successful completion of the update location procedure to the VLR.

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

959 6.1.3.5 The Interworking MSC (IWMSC) for Short Message Service

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

962 6.1.3.6 The Equipment Identity Register (EIR)

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

964 6.1.3.7 The Shared Inter Working Function (SIWF)

965 When the Visited MSC detects a data or fax call and the IWF in the V-MSC can not handle the required service an
966 SIWF can be invoked. The SIWF is addressed with an E.164 number.

967 6.1.3.8 The Serving GPRS Support Node (SGSN)

968 The HLR will initiate dialogues towards the SGSN if it is aware that one of its subscribers is in the SGSN's serving area.
969 This means that a GPRS location updating has been successfully completed, i.e, the HLR has indicated successful
970 completion of the GPRS location update to the SGSN. The routing information used by the HLR is derived from the
971 E.164 SGSN number received as parameter of the MAP message initiating the GPRS update location procedure. If the
972 SGSN is in the same PLMN as the HLR, the SGSN may be addressed directly by an SPC derived from the E.164 SGSN
973 number. For dialogues via the international PSTN/ISDN signalling network, the presence of the E.164 SGSN number in
974 the Called Party Address is required.

975 When the GMSC initiates dialogues towards the SGSN the SGSN (MAP) SSN (See GSM 03.03) shall be included in
976 the called party address. The routing information used by the GMSC is derived from the E.164 SGSN number received
977 as a parameter of the MAP message initiating the forward short message procedure. If the GMSC does not support the
978 GPRS functionality the MSC (MAP) SSN value shall be included in the called party address.

979 Note: Every VMSC and SGSN shall have uniquely identifiable application using E.164 numbers, for the
980 purpose of SMS over GPRS when the GMSC does not support the GPRS functionality.

981 6.1.3.9 The Gateway GPRS Support Node (GGSN)

982 The GGSN provides interworking with external packet-switched networks, network screens and routing of the Network-
983 Requested PDP Context activation. If a Network-Requested PDP Context activation fails, the HLR will alert the GGSN
984 when the subscriber becomes reachable. The HLR will use the E.164 GGSN number received as parameter of the MAP
985 message reporting the failure.

986 6.1.3.10 The Gateway MSC (GMSC) for Short Message Service

987 The GMSC provides interworking with the network to access the Short Message Service Centre, the mobile network and
988 routing of Send Routing Info For SM. The GMSC has on E.164 address known in the HLR, SGSN or MSC

989 6.1.3.10A Void

990 6.1.3.10A.1 Void

991 6.1.3.10A.2 Void

992 6.1.3.10B The Gateway Mobile Location Center (GMLC)

993 The GMLC initiates location requests on behalf of external clients. The E.164 address of the GMLC is provided to an
994 HLR when the GMLC requests a serving MSC address from the HLR for a target MS. The E.164 address of the GMLC
995 is also provided to a serving MSC when the GMLC requests the location of a target MS served by this MSC.

996 6.1.3.11 Summary table

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

1000 For a response, the originating address passed in the invoke is used as SCCP Called Party Address. For extra-PLMN
1001 addressing the own E.164 entity address is used as SCCP Calling Party Address; for intra-PLMN addressing an SPC
1002 derived from the entity number may be used instead. When using an SPC, the SPC may be taken directly from MTP.

1003

Table 6.1/1

to from	fixed net work	HLR	VLR	MSC	EIR	gsmSCF	SIWF	SGSN	GGSN
fixed network	---	E:GT T:MSISDN	---	---	---	---	---	---	---
home location register	---	---	I:SPC/GT E:GT T:VLR NUMBER	---	---	I:SPC/GT E:GT T:gsmSCF NUMBER	---	I:SPC/GT E:GT T:SGSN NUMBER	I:SPC/GT E:GT T:GGSN NUMBER
visitor location register	---	I:SPC/GT E:GT T:MGT (outside World Zone 1)/MSISDN (World Zone 1)/HLR NUMBER (note)	I:SPC/GT E:GT T:VLR NUMBER	---	---	I:SPC/GT E:GT T:gsmSCF 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:SPC/GT E:GT T:gsmSCF NUMBER	I:SPC/GT E:GT T:SIWF NUMBER	I:SPC/GT E:GT T:SGSN NUMBER	---
gsm Service Control Function	---	I:SPC/GT E:GT T:MSISDN	---	---	---	---	---	---	---
Shared Inter Working Function	---	---	---	I:SPC/GT E:GT T:MSC NUMBER	---	---	---	---	---
Serving GPRS Support Node	---	I:SPC/GT E:GT T:MGT/ MSISDN/HL R NUMBER	---	I:SPC/GT E:GT T:MSC NUMBER	I:SPC/GT E:GT T:EIR NUMBER	---	---	---	---
Gateway GPRS Support Node	---	I:SPC/GT E:GT T:MGT	---	---	---	---	---	---	---
Gateway Mobile Location Center	---	I:SPC/GT E:GT T:MSISDN, MGT (outside World Zone 1) or IMSI (World Zone 1) (note)	---	I:SPC/GT E:GT T:MSC NUMBER	---	---	---	---	---

1004

1005

1006

1007

1008

1009

1010

1011

1012

1013

1014

1015

1016

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 if CCITT or ITU-T SCCP is used, or IMSI itself if ANSI SCCP is used (ANSI SCCP is used in World Zone 1).. When continuing the established update location dialogue (as with any other dialogue) the VLR must derive the routing 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.

1017 When invoking the Restore Data procedure and an authentication information retrieval from the HLR
 1018 preceding it, the VLR must derive the information for addressing the HLR from the address information
 1019 received in association with the roaming number request. This may be either the IMSI received as a
 1020 parameter of the MAP message requesting the Roaming Number or the Calling Party Address associated
 1021 with the MAP message requesting the Roaming Number.
 1022 The gsmSCF shall be addressed using more than one Global Title number. The first Global Title number
 1023 is used to address a gsmSCF for MAP. The second Global Title number is used to address a gsmSCF for
 1024 CAP.
 1025 For querying the HLR to obtain the VMSC address to support location services, the GMLC has to derive
 1026 the HLR address from either the MSISDN or IMSI of the target MS. When using the IMSI, the result can
 1027 be an SPC or an E.214 Mobile Global Title if CCITT or ITU-T SCCP is used, or IMSI itself if ANSI
 1028 SCCP is used (ANSI SCCP is used in World Zone 1).
 1029

1030

Table 6.1/2

to from		GMLC
fixed network		---
home location register		---
visitor location register		---
mobile-services switching centre		---
gsm Service Control Function		I:SPC/GT E:GT T:MSISDN
Shared Inter Working Function		---
Serving GPRS Support Node		---
Gateway GPRS Support Node		---
Gateway Mobile Location Center		

1031

1032 I: Intra-PLMN E: Extra(Inter)-PLMN T: Address Type
 1033 GT: Global Title MGT: E.214 Mobile Global Title SPC: Signalling Point Code

1034 6.2 Use of TC

1035 The Mobile Application part makes use of the services offered by the Transaction Capabilities (TC) of signalling system
 1036 No. 7. ETS 300 287, which is based on CCITT White Book Recommendations Q.771 to Q.775, should be consulted for
 1037 the full specification of TC.

1038 The MAP uses all the services provided by TC except the ones related to the unstructured dialogue facility.

1039 From a modelling perspective, the MAP is viewed as a single Application Service Element. Further structuring of it is
 1040 for further study.

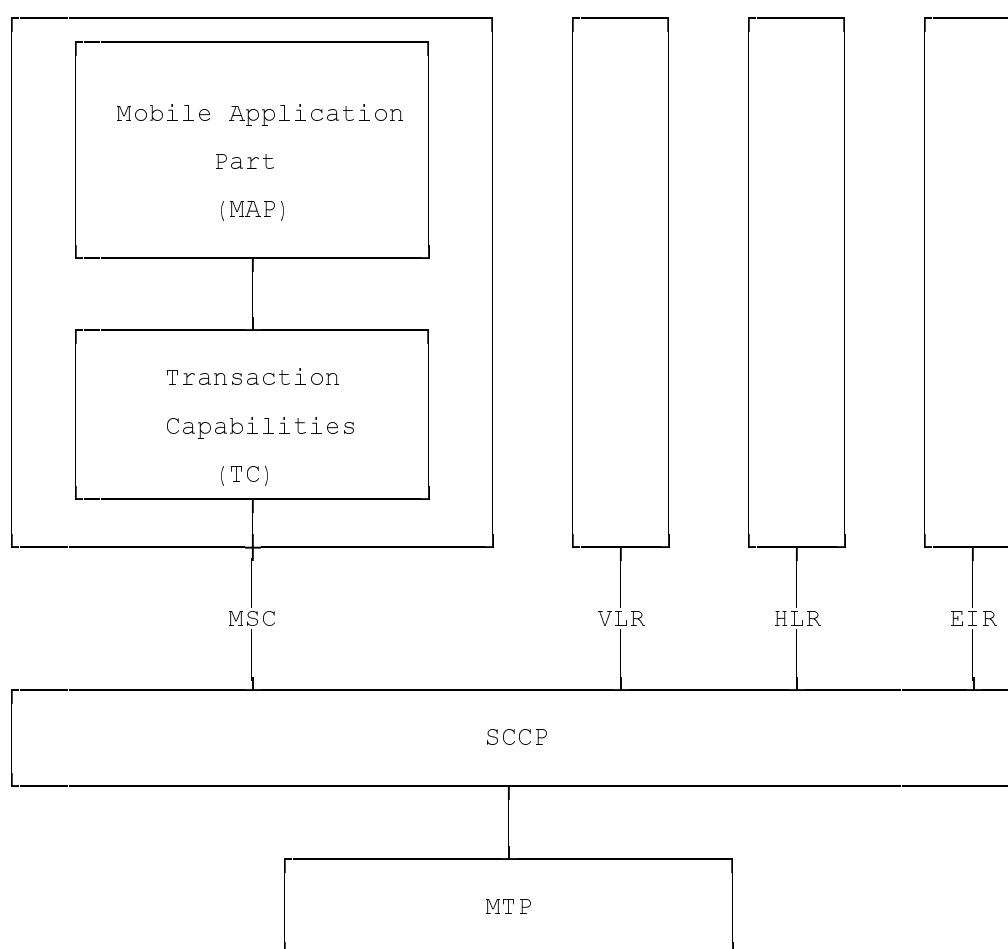
1041 Transaction Capabilities refers to a protocol structure above the network layer interface (i.e. the SCCP service interface)
 1042 up to the application layer including common application service elements but not the specific application service
 1043 elements using them.

1044 TC is structured as a Component sub-layer above a Transaction sub-layer.

1045 The Component sub-layer provides two types of application services: services for the control of end-to-end dialogues
 1046 and services for Remote Operation handling. These services are accessed using the TC-Dialogue handling primitives
 1047 and TC-Component handling primitives respectively.

1048 Services for dialogue control include the ability to exchange information related to application-context negotiation as
 1049 well as initialization data.

1050 Services for Remote Operation handling provide for the exchange of protocol data units invoking tasks (operations), and
 1051 reporting their outcomes (results or errors) plus any non-application-specific protocol errors detected by the component
 1052 sub-layer. The reporting of application-specific protocol errors by the TC user, as distinct from application process
 1053 errors, is also provided. The Transaction sub-layer provides a simple end-to-end connection association service over
 1054 which several related protocol data units (i.e. built by the Component Sub-Layer) can be exchanged. A Transaction
 1055 termination can be prearranged (no indication provided to the TC user) or basic (indication provided).



1056

1057 **Figure 6.2/1: Facilities for supporting the Mobile Application Part in Signalling System No.7**

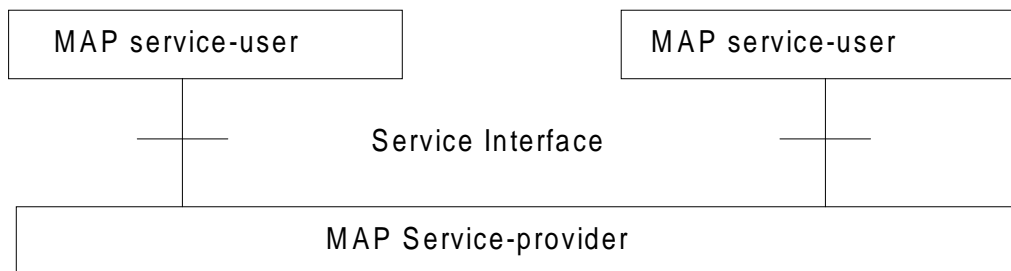
1058 7 General on MAP services

1059 7.1 Terminology and definitions

1060 The term service is used in clauses 7 to 12 as defined in CCITT Recommendation X.200. The service definition
 1061 conventions of CCITT Recommendation X.210 are also used.

1062 7.2 Modelling principles

1063 MAP provides its users with a specified set of services and can be viewed by its users as a "black box" or abstract
1064 machine representing the MAP service-provider. The service interface can then be depicted as shown in figure 7.2/1.



1065

1066

Figure 7.2/1: Modelling principles

1067 The MAP service-users interact with the MAP service-provider by issuing or receiving MAP service-primitives at the
1068 service interface.

1069 A MAP service-user may receive services from several instances of the MAP service-provider at the same time. In such
1070 cases the overall procedure is synchronised by the service-user.

1071 The MAP service-primitives are named using the following notation:

1072 MAP-ServicePrimitiveName **type**

1073 where **type** can be any of: request (req), indication (ind), response (rsp) or confirm (cnf) (In the user arrow diagrams
1074 type is not indicated in the case of req/ind and indicated as "ack" in the case of rsp/cnf).

1075 The services are further classified as unconfirmed-service, confirmed-service and provider-initiated-service where the
1076 first two categories refer to whether or not the service is confirmed by the service-provider. The confirmation may or
1077 may not correspond to a response provided by the other service-user.

1078 MAP services are also classified as common MAP services which are available to all MAP service-users, and MAP
1079 service-user specific services which are services available to one or several, but not all, MAP service-users.

1080 A MAP dialogue is defined as an exchange of information between two MAP users in order to perform a common task.

1081 A MAP dialogue will consist of one or several MAP services.

1082 7.3 Common MAP services

1083 All MAP service-users require access to services for performing basic application layer functions:

- 1084 - for establishing and clearing MAP dialogues between peer MAP service-users;
- 1085 - for accessing functions supported by layers below the applications layer;
- 1086 - for reporting abnormal situations;
- 1087 - for handling of different MAP versions;
- 1088 - for testing whether or not a persistent MAP dialogue is still active at each side.

1089 For these purposes the following common services are defined:

- 1090 - MAP-OPEN service;
- 1091 - MAP-CLOSE service;
- 1092 - MAP-DELIMITER service;
- 1093 - MAP-U-ABORT service;

1094 - MAP-P-ABORT service;

1095 - MAP-NOTICE service.

1096 In defining the service-primitives the following convention is used for categorising parameters:

1097 M the inclusion of the parameter is mandatory. The M category can be used for any primitive type and specifies
1098 that the corresponding parameter must be present in the indicated primitive type;

1099 O the inclusion of the parameter is a service-provider option. The O category can be used in indication and
1100 confirm type primitives and is used for parameters that may optionally be included by the service-provider;

1101 U the inclusion of the parameter is a service-user option. The U category can be used in request and response
1102 type primitives. The inclusion of the corresponding parameter is the choice of the service-user;

1103 C the inclusion of the parameter is conditional. The C category can be used for the following purposes:

1104 - to indicate that if the parameter is received from another entity it must be included for the service being
1105 considered;

1106 - to indicate that the service user must decide whether to include the parameter, based on the context on
1107 which the service is used;

1108 - to indicate that one of a number of mutually exclusive parameters must be included (e.g. parameters
1109 indicating a positive result versus parameters indicating a negative result);

1110 - to indicate that a service user optional parameter (marked "U") or a conditional parameter (marked "C")
1111 presented by the service user in a request or response type primitive is to be presented to the service user in
1112 the corresponding indication or confirm type primitive;

1113 (=) when appended to one of the above, this symbol means that the parameter takes the same value as the
1114 parameter appearing immediately to its left;

1115 blank the parameter is not present.

1116 A primitive type may also be without parameters, i.e. no parameter is required with the primitive type; in this case the
1117 corresponding column of the table is empty.

1118 7.3.1 MAP-OPEN service

1119 This service is used for establishing a MAP dialogue between two MAP service-users. The service is a confirmed
1120 service with service primitives as shown in table 7.3/1.

1121 **Table 7.3/1: Service-primitives for the MAP-OPEN service**

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

1122

1123 Application context name:

1124 This parameter identifies the type of application context being established. If the dialogue is accepted the received
1125 application context name shall be echoed. In case of refusal of dialogue this parameter shall indicate the highest version
1126 supported.

1127 Destination address:

1128 A valid SCCP address identifying the destination peer entity (see also clause 6). As an implementation option, this
 1129 parameter may also, in the indication, be implicitly associated with the service access point at which the primitive is
 1130 issued.

1131 Destination-reference:

1132 This parameter is a reference which refines the identification of the called process. It may be identical to Destination
 1133 address but its value is to be carried at MAP level. Table 7.3/2 describes the MAP services using this parameter. Only
 1134 these services are allowed to use it.

1135 **Table 7.3/2: Use of the destination reference**

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
MAP-REGISTER-CC-ENTRY	IMSI	Subscriber identity
MAP-ERASE-CC-ENTRY	IMSI	Subscriber identity

1136
 1137 NOTE: Only when the IMSI and the LMSI are received together from the HLR in the mobile terminated short
 1138 message transfer.

1139 Originating address:

1140 A valid SCCP address identifying the requestor of a MAP dialogue (see also clause 6). As an implementation option,
 1141 this parameter may also, in the request, be implicitly associated with the service access point at which the primitive is
 1142 issued.

1143 Originating-reference:

1144 This parameter is a reference which refines the identification of the calling process. It may be identical to the
 1145 Originating address but its value is to be carried at MAP level. Table 7.3/3 describes the MAP services using the
 1146 parameter. Only these services are allowed to use it. Processing of the Originating-reference shall be performed
 1147 according to the supplementary service descriptions and other service descriptions, e.g. operator determined barring.

1148

Table 7.3/3: Use of the originating reference

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
MAP-REGISTER-CC-ENTRY	ISDN-Address-String	Originated entity address
MAP-ERASE-CC-ENTRY	ISDN-Address-String	Originated entity address

1149

1150 Specific information:

1151 This parameter may be used for passing any user specific information. Establishment and processing of the Specific
1152 information is not specified by GSM and shall be performed according to operator specific requirements.

1153 Responding address:

1154 An address identifying the responding entity. The responding address is included if required by the context (e.g. if it is
1155 different from the destination address).

1156 Result:

1157 This parameter indicates whether the dialogue is accepted by the peer.

1158 Refuse reason:

1159 This parameter is only present if the Result parameter indicates that the dialogue is refused. It takes one of the following
1160 values:

- 1161 - Application-context-not-supported;
- 1162 - Invalid-destination-reference;
- 1163 - Invalid-originating-reference;
- 1164 - No-reason-given;
- 1165 - Remote node not reachable;
- 1166 - Potential version incompatibility.

1167 7.3.2 MAP-CLOSE service

1168 This service is used for releasing a previously established MAP dialogue. The service may be invoked by either MAP
1169 service-user depending on rules defined within the service-user. The service is an unconfirmed service with parameters
1170 as shown in table 7.3/4.

1171 **Table 7.3/4: Service-primitives for the MAP-CLOSE service**

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

1172

1173 Release method:

1174 This parameter can take the following two values:

- 1175 - normal release; in this case the primitive is mapped onto the protocol and sent to the peer;
- 1176 - prearranged end; in this case the primitive is not mapped onto the protocol. Prearranged end is managed
1177 independently by the two users, i.e. only the request type primitive is required in this case.

1178 Specific information:

1179 This parameter may be used for passing any user specific information. Establishment and processing of the Specific
1180 information is not specified by GSM GSM and shall be performed according to operator specific requirements.

1181 7.3.3 MAP-DELIMITER service

1182 This service is used to explicitly request the transfer of the MAP protocol data units to the peer entities.

1183 See also subclause 7.4 and 7.5 for the detailed use of the MAP-DELIMITER service.

1184 The service is an unconfirmed service with service-primitives as shown in table 7.3/5.

1185 **Table 7.3/5: Service-primitives for the MAP-DELIMITER service**

Parameters	Request	Indication

1186

1187 7.3.4 MAP-U-ABORT service

1188 This service enables the service-user to request the MAP dialogue to be aborted. The service is an unconfirmed service
1189 with service-primitives as shown in table 7.3/6.

1190 **Table 7.3/6: Service-primitives for the MAP-U-ABORT service**

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

1191

1192 User reason:

1193 This parameter can take the following values:

- 1194 - resource limitation (congestion);
- 1195 the requested user resource is unavailable due to congestion;

- 1196 - resource unavailable;
 1197 the requested user resource is unavailable for reasons other than congestion;
 1198 - application procedure cancellation;
 1199 the procedure is cancelled for reason detailed in the diagnostic information parameter;
 1200 - procedure error;
 1201 processing of the procedure is terminated for procedural reasons.

1202 Diagnostic information:

1203 This parameter may be used to give additional information for some of the values of the user-reason parameter:

1204 **Table 7.3/7: User reason and diagnostic information**

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	-

- 1205
 1206 Specific information:
 1207 This parameter may be used for passing any user specific information. Establishment and processing of the Specific
 1208 information is not specified by GSM and shall be performed according to operator specific requirements.

1209 7.3.5 MAP-P-ABORT service

1210 This service enables the MAP service-provider to abort a MAP dialogue. The service is a provider-initiated service with
 1211 service-primitive as shown in table 7.3/8.

1212 **Table 7.3/8: Service-primitive for the MAP-P-ABORT service**

Parameters	Indication
Provider reason	M
Source	M

- 1213
 1214 Provider reason:
 1215 This parameter indicates the reason for aborting the MAP dialogue:
 1216 - provider malfunction;
 1217 - supporting dialogue/transaction released;
 1218 - resource limitation;
 1219 - maintenance activity;
 1220 - version incompatibility;
 1221 - abnormal MAP dialogue.

1222 Source:

1223 This parameter indicates the source of the abort. For Transaction Capabilities (TC) applications the parameter may take
1224 the following values:

- 1225 - MAP problem;
- 1226 - TC problem;
- 1227 - network service problem.

1228 **Table 7.3/9: Values of provider reason and source parameters and examples of corresponding events**

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

1229

1230 7.3.6 MAP-NOTICE service

1231 This service is used to notify the MAP service-user about protocol problems related to a MAP dialogue not affecting the
1232 state of the protocol machines.

1233 The service is a provider-initiated service with service-primitive as shown in table 7.3/10.

1234 **Table 7.3/10: Service-primitive for the MAP-NOTICE service**

Parameters	Indication
Problem diagnostic	M

1235

1236 Problem diagnostic:

1237 This parameter can take one of the following values:

- 1238 - abnormal event detected by the peer;
- 1239 - response rejected by the peer;
- 1240 - abnormal event received from the peer
- 1241 - message cannot be delivered to the peer.

1242 7.4 Sequencing of services

1243 The sequencing of services is shown in figure 7.4/1 and is as follows:

1244 Opening:

1245 The MAP-OPEN service is invoked before any user specific service-primitive is accepted. The sequence may
1246 contain none, one or several user specific service-primitives. If no user specific service-primitive is contained
1247 between the MAP-OPEN and the MAP-DELIMITER primitives, then this will correspond to sending an
1248 empty Begin message in TC. If more than one user specific service-primitive is included, all are to be sent in
1249 the same Begin message. The sequence ends with a MAP-DELIMITER primitive.

1250 Continuing:

1251 This sequence may not be present in some MAP dialogues. If it is present, it ends with a MAP-DELIMITER
1252 primitive. If more than one user specific service-primitive is included, all are to be included in the same
1253 Continue message.

1254 Closing:

1255 The sequence can only appear after an opening sequence or a continuing sequence. The sequence may contain
1256 none, one or several user specific service-primitives if the MAP-CLOSE primitive specifies normal release. If
1257 no user specific service-primitive is included, then this will correspond to sending an empty End message in
1258 TC. If more than one user specific service-primitive is included, all are to be sent in the same End message. If
1259 prearranged end is specified, the sequence cannot contain any user specific service-primitive. The MAP-
1260 CLOSE primitive must be sent after all user specific service-primitives have been delivered to the MAP
1261 service-provider.

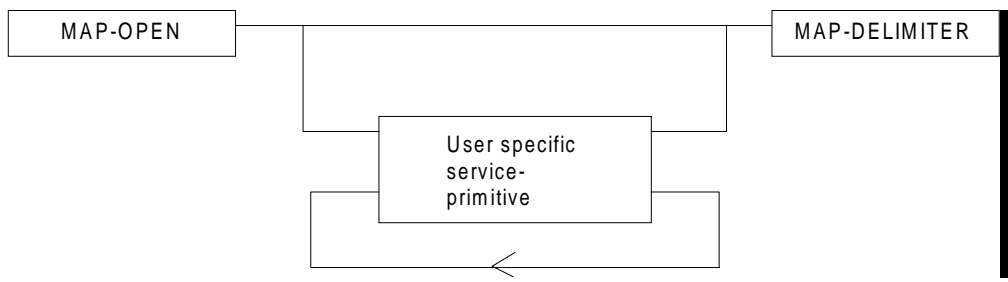
1262 Aborting:

1263 A MAP service-user can issue a MAP-U-ABORT primitive at any time after the MAP dialogue has been
1264 opened or as a response to an attempt to open a MAP dialogue.

1265 The MAP service-provider may issue at any time a MAP-P-ABORT primitive towards a MAP service-user for which a
1266 MAP dialogue exists.

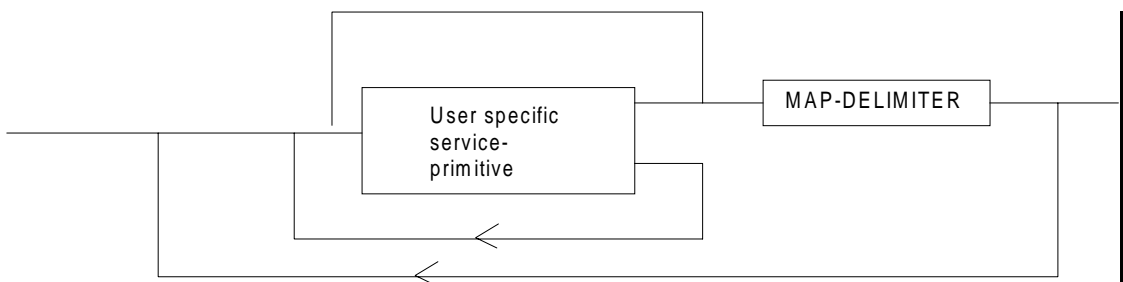
1267 MAP-U-ABORT primitives and MAP-P-ABORT primitives terminate the MAP dialogue.

1268
1269
1270
1271



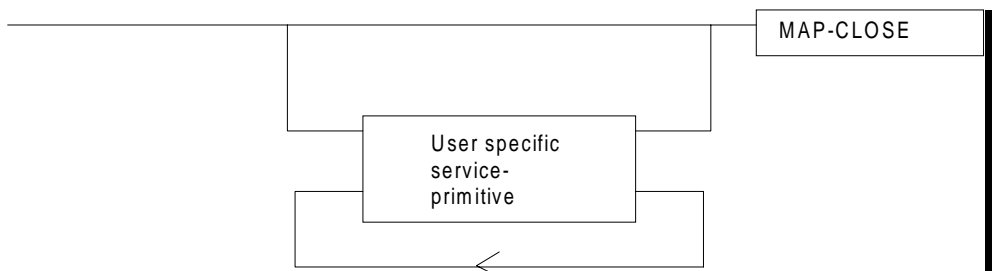
a) Opening

1272
1273
1274
1275



b) Continuing

1276
1277
1278
1279



c) Closing

1280
1281
1282
1283



d) Aborting

Figure 7.4/1: Sequencing of services

1284 If the reason "resource unavailable (short term problem)" is indicated in the MAP-U-ABORT indication primitive, the
1285 MAP service-user may decide to attempt a new MAP dialogue establishment immediately.

1286 Sequencing of user specific service-primitives is done by the MAP service-user and based on rules applicable for each
1287 MAP service-user instance.

1288 A MAP-NOTICE indication primitive may be received at any time during the active period of a MAP dialogue.

1289 7.5 General rules for mapping of services onto TC

1290 7.5.1 Mapping of common services

1291 Table 7.5/1 gives an overview of the mapping rules for mapping of common services onto TC-services. Table 7.5/2
1292 gives the mapping rules for mapping of TC-services onto common services.

1293 Protocol machine description is given in clauses 14 to 17.

1294

Table 7.5/1: Mapping of common services on to TC services

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

1295

1296 NOTE: or TC-END if the MAP-CLOSE request has been received before the MAP-DELIMITER request.

1297

1298

Table 7.5/2: Mapping of TC services on to common service

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)

1299

1300 NOTE 1: It may not be necessary to present this primitive to the user for MAP version 2 applications.

1301 NOTE 2: The mapping depends on whether the TC-U-ABORT indication primitive contains a MAP-abort-PDU
1302 from the remote MAP service-provider or a MAP-user-abort-PDU from the remote MAP service-user.1303 NOTE 3: Only if the opening sequence is pending and if the "Abort Reason" in the TC-U-ABORT indication is set
1304 to "Application Context Not Supported".1305 NOTE 4: If the "Abort Reason" in the TC-P-ABORT indication is set to a value different from "Incorrect
1306 Transaction Portion".1307 NOTE 5: Only if the opening sequence is pending and if the "Abort Reason" in the TC-P-ABORT indication is set
1308 to "Incorrect Transaction Portion".

1309 NOTE 6: Only if opening sequence is pending.

1310 7.5.2 Mapping of user specific services

1311 Table 7.5/3 gives the general mapping rules which apply to mapping of MAP user specific services onto TC services
 1312 and table 7.5/4 gives the similar rules for mapping of TC services onto MAP user specific services. Detailed mapping is
 1313 given in clauses 14 to 17.

1314 **Table 7.5/3: Mapping of MAP user specific services onto TC services**

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)

1315

1316 **Table 7.5/4: Mapping of TC services onto MAP user specific 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)

1317

1318 Notes to tables 7.5/3 and 7.5/4:

1319 NOTE 1: The mapping is determined by parameters contained in the MAP-xx response primitive.

1320 NOTE 2: This applies only to TC class 4 operations where the operation is used to pass a result of another class 2 or
 1321 class 4 operation.

1322 NOTE 3: The detailed mapping rules are given in clause 16.

1323 NOTE 4: If RESULT-NL components are present they are mapped on to the same MAP-xx confirm.

1324 7.6 Definition of parameters

1325 Following is an alphabetic list of parameters used in the common MAP-services in subclause 7.3:

Application context name	7.3.1	Refuse reason	7.3.1
Destination address	7.3.1	Release method	7.3.2
Destination reference	7.3.1	Responding address	7.3.1
Diagnostic information	7.3.4	Result	7.3.1
Originating address	7.3.1	Source	7.3.5
Originating reference	7.3.1	Specific information	7.3.1/7.3.2/7.3.4
Problem diagnostic	7.3.6	User reason	7.3.4
Provider reason	7.3.5		

1326

1327 Following is an alphabetic list of parameters contained in this clause:

Absent Subscriber Diagnostic SM	7.6.8.9	Invoke Id	7.6.1.1
Access connection status	7.6.9.3	ISDN Bearer Capability	7.6.3.41
		IST Alert Timer	7.6.3.66
		IST Information Withdrawn	7.6.3.68
		IST Support Indicator	7.6.3.69
Access signalling information	7.6.9.5	Kc	7.6.7.4
Additional Absent Subscriber Diagnostic SM	7.6.8.12	Linked Id	7.6.1.2
Additional number	7.6.2.46	LMSI	7.6.2.16
Additional signal info	7.6.9.10	Location Information	7.6.2.30
Additional SM Delivery Outcome	7.6.8.11		
Age Indicator	7.6.3.72	Location update type	7.6.9.6
Alert Reason	7.6.8.8	Lower Layer Compatibility	7.6.3.42
		LSA Information	7.6.3.56
		LSA Information Withdraw	7.6.3.58
Alert Reason Indicator	7.6.8.10	Mobile Not Reachable Reason	7.6.3.51
Alerting Pattern	7.6.3.44	Modification request for CSI	7.6.3.81
All GPRS Data	7.6.3.53	Modification request for SS Information	7.6.3.82
All Information Sent	7.6.1.5	More Messages To Send	7.6.8.7
APN	7.6.2.42	MS ISDN	7.6.2.17
Authentication set list	7.6.7.1	MSC number	7.6.2.11
B-subscriber Address	7.6.2.36	MSISdn-Alert	7.6.2.29
B subscriber Number	7.6.2.48	MWD status	7.6.8.3
B subscriber subaddress	7.6.2.49	Network Access Mode	7.6.3.50
Basic Service Group	7.6.4.40	Network node number	7.6.2.43
Bearer service	7.6.4.38	Network resources	7.6.10.1
BSS-apdu	7.6.9.1	Network signal information	7.6.9.8
Call Barring Data	7.6.3.83	New password	7.6.4.20
Call barring feature	7.6.4.19	No reply condition timer	7.6.4.7
Call barring information	7.6.4.18	North American Equal Access preferred Carrier Id	7.6.2.34
		Number Portability Status	7.6.5.14
Call Direction	7.6.5.8	ODB Data	7.6.3.85
Call Forwarding Data	7.6.3.84	ODB General Data	7.6.3.9
Call Info	7.6.9.9	ODB HPLMN Specific Data	7.6.3.10
Call reference	7.6.5.1		
Call Termination Indicator	7.6.3.67	OMC Id	7.6.2.18
Called number	7.6.2.24	Originally dialled number	7.6.2.26
Calling number	7.6.2.25	Originating entity number	7.6.2.10
CAMEL Subscription Info	7.6.3.78	Override Category	7.6.4.4
CAMEL Subscription Info Withdraw	7.6.3.38	P-TMSI	7.6.2.47
Cancellation Type	7.6.3.52	PDP-Address	7.6.2.45
Category	7.6.3.1	PDP-Context identifier	7.6.3.55
CCBS Feature	7.6.5.8	PDP-Type	7.6.2.44
Channel Type	7.6.5.9	Pre-paging supported	7.6.5.15
Chosen Channel	7.6.5.10	Previous location area Id	7.6.2.4
Ciphering mode	7.6.7.7	Protocol Id	7.6.9.7
Cksn	7.6.7.5	Provider error	7.6.1.3
CLI Restriction	7.6.4.5	QoS-Subscribed	7.6.3.47
CM service type	7.6.9.2	Rand	7.6.7.2
Complete Data List Included	7.6.3.54	Regional Subscription Data	7.6.3.11
CUG feature	7.6.3.26	Regional Subscription Response	7.6.3.12
CUG index	7.6.3.25	Requested Info	7.6.3.31
CUG info	7.6.3.22	Requested Subscription Info	7.6.3.86
CUG interlock	7.6.3.24	Roaming number	7.6.2.19
CUG Outgoing Access indicator	7.6.3.8	Roaming Restricted In SGSN Due To	7.6.3.49
CUG subscription	7.6.3.23	Unsupported Feature	
		Roaming Restriction Due To	7.6.3.13
CUG Subscription Flag	7.6.3.37	Unsupported Feature	
		Service centre address	7.6.2.27
Current location area Id	7.6.2.6	Serving Cell Id	7.6.2.37
Current password	7.6.4.21	SGSN address	7.6.2.39
eMLPP Information	7.6.4.41	SGSN CAMEL Subscription Info	7.6.3.75
Equipment status	7.6.3.2	SGSN number	7.6.2.38
Extensible Basic Service Group	7.6.3.5	SIWF Number	7.6.2.35
Extensible Bearer service	7.6.3.3	SoLSA Support Indicator	7.6.3.57
		SM Delivery Outcome	7.6.8.6
Extensible Call barring feature	7.6.3.21	SM-RP-DA	7.6.8.1
Extensible Call barring information	7.6.3.20		

Extensible Call barring information for CSE	7.6.3.79	SM-RP-MTI	7.6.8.16
Extensible Forwarding feature	7.6.3.16	SM-RP-OA	7.6.8.2
Extensible Forwarding info	7.6.3.15	SM-RP-PRI	7.6.8.5
Extensible Forwarding information for CSE	7.6.3.80	SM-RP-SMEA	7.6.8.17
Extensible Forwarding Options	7.6.3.18	SM-RP-UI	7.6.8.4
Extensible No reply condition timer	7.6.3.19	Sres	7.6.7.3
Extensible QoS-Subscribed	7.6.3.74	SS-Code	7.6.4.1
Extensible SS-Data	7.6.3.29	SS-Data	7.6.4.3
Extensible SS-Info	7.6.3.14	SS-Event	7.6.4.42
Extensible SS-Status	7.6.3.17	SS-Event-Data	7.6.4.43
Extensible Teleservice	7.6.3.4	SS-Info	7.6.4.24
External Signal Information	7.6.9.4	SS-Status	7.6.4.2
Forwarded-to number	7.6.2.22	Stored location area Id	7.6.2.5
Forwarded-to subaddress	7.6.2.23	Subscriber State	7.6.3.30
Forwarding feature	7.6.4.16	Subscriber Status	7.6.3.7
Forwarding information	7.6.4.15	Super-Charger Supported in HLR	7.6.3.70
Forwarding Options	7.6.4.6	Super-Charger Supported in Serving Network Entity	7.6.3.71
GGSN address	7.6.2.40	Supported CAMEL Phases in VLR	7.6.3.36
GGSN number	7.6.2.41	Supported CAMEL Phases in SGSN	7.6.3.36A
GMSC CAMEL Subscription Info	7.6.3.34	Suppress T-CSI	7.6.3.33
GPRS enhancements support indicator	7.6.3.73	Suppression of Announcement	7.6.3.32
GPRS Node Indicator	7.6.8.14	Target cell Id	7.6.2.8
GPRS Subscription Data	7.6.3.46	Target location area Id	7.6.2.7
GPRS Subscription Data Withdraw	7.6.3.45	Target MSC number	7.6.2.12
GPRS Support Indicator	7.6.8.15	Teleservice	7.6.4.39
Group Id	7.6.2.33	TMSI	7.6.2.2
GSM bearer capability	7.6.3.6	Trace reference	7.6.10.2
Guidance information	7.6.4.22	Trace type	7.6.10.3
Handover number	7.6.2.21	User error	7.6.1.4
High Layer Compatibility	7.6.3.43	USSD Data Coding Scheme	7.6.4.36
HLR Id	7.6.2.15	USSD String	7.6.4.37
HLR number	7.6.2.13	UU Data	7.6.5.12
HO-Number Not Required	7.6.6.7	UUS CF Interaction	7.6.5.13
IMEI	7.6.2.3	VBS Data	7.6.3.40
IMSI	7.6.2.1	VGCS Data	7.6.3.39
Inter CUG options	7.6.3.27	VLR CAMEL Subscription Info	7.6.3.35
Intra CUG restrictions	7.6.3.28	VLR number	7.6.2.14
		VPLMN address allowed	7.6.3.48
		Zone Code	7.6.2.28

1328

1329 7.6.1 Common parameters

1330 The following set of parameters are used in several MAP service-primitives:

1331 7.6.1.1 Invoke Id

1332 This parameter identifies corresponding service primitives. The parameter is supplied by the MAP service-user and must
1333 be unique over each service-user/service-provider interface.

1334 7.6.1.2 Linked Id

1335 This parameter is used for linked services and it takes the value of the invoke Id of the service linked to.

1336 7.6.1.3 Provider error

1337 This parameter is used to indicate a protocol related type of error:

1338 - duplicated invoke Id;

1339 - not supported service;

- 1340 - mistyped parameter;
- 1341 - resource limitation;
- 1342 - initiating release, i.e. the peer has already initiated release of the dialogue and the service has to be released;
- 1343 - unexpected response from the peer;
- 1344 - service completion failure;
- 1345 - no response from the peer;
- 1346 - invalid response received.

1347 7.6.1.4 User error

1348 This parameter can take values as follows:

1349 NOTE: The values are grouped in order to improve readability; the grouping has no other significance.

1350 a) Generic error:

- 1351 - system failure, i.e. a task cannot be performed because of a problem in another entity. The type of entity or
1352 network resource may be indicated by use of the network resource parameter;
- 1353 - data missing, i.e. an optional parameter required by the context is missing;
- 1354 - unexpected data value, i.e. the data type is formally correct but its value or presence is unexpected in the
1355 current context;
- 1356 - resource limitation;
- 1357 - initiating release, i.e. the receiving entity has started the release procedure;
- 1358 - facility not supported, i.e. the requested facility is not supported by the PLMN;
- 1359 - incompatible terminal, i.e. the requested facility is not supported by the terminal.

1360 b) Identification or numbering problem:

- 1361 - unknown subscriber, i.e. no such subscription exists;
- 1362 - number changed, i.e. the subscription does not exist for that number any more;
- 1363 - unknown MSC;
- 1364 - unidentified subscriber, i.e. if the subscriber is not contained in the database and it has not or cannot be
1365 established whether or not a subscription exists;
- 1366 - unallocated roaming number;
- 1367 - unknown equipment;
- 1368 - unknown location area.

1369 c) Subscription problem:

- 1370 - roaming not allowed, i.e. a location updating attempt is made in an area not covered by the subscription;
- 1371 - illegal subscriber, i.e. illegality of the access has been established by use of authentication procedure;
- 1372 - bearer service not provisioned;
- 1373 - teleservice not provisioned;
- 1374 - illegal equipment, i.e. the IMEI check procedure has shown that the IMEI is blacklisted or not whitelisted.

- 1375 d) Handover problem:
- 1376 - no handover number available;
- 1377 - subsequent handover failure, i.e. handover to a third MSC failed for some reason.
- 1378 e) Operation and maintenance problem:
- 1379 - tracing buffer full, i.e. tracing cannot be performed because the tracing capacity is exceeded.
- 1380 f) Call set-up problem:
- 1381 - no roaming number available, i.e. a roaming number cannot be allocated because all available numbers are in
1382 use;
- absent subscriber, i.e. the subscriber has activated the detach service or the system detects the absence condition. This error may be qualified to indicate whether the subscriber was IMSI detached, in a restricted area or did not respond to paging;
- 1383 - busy subscriber. This error may be qualified to indicate that the subscriber was busy due to CCBS and that
1384 CCBS is possible;
- 1385 - no subscriber reply;
- 1386 - forwarding violation, i.e. the call has already been forwarded the maximum number of times that is allowed;
- 1387 - CUG reject, i.e. the call does not pass a CUG check; additional information may also be given in order to
1388 indicate rejection due to e.g. incoming call barred or non-CUG membership.
- 1389 - call barred. Optionally, additional information may be included for indicating either that the call meets a
1390 barring condition set by the subscriber or that the call is barred for operator reasons. In case of barring of
1391 Mobil Terminating Short Message, the additional information may indicate a barring condition due to
1392 « unauthorised Message Originator».
- 1393 - optimal routeing not allowed, i.e. the entity which sends the error does not support optimal routeing, or the
1394 HLR will not accept an optimal routeing interrogation from the GMSC, or the call cannot be optimally routed
1395 because it would contravene optimal routeing constraints.
- 1396 - forwarding failed, i.e. the GMSC interrogated the HLR for forwarding information but the HLR returned an
1397 error.
- 1398 g) Supplementary services problem:
- 1399 - call barred;
- 1400 - illegal SS operation;
- 1401 - SS error status;
- 1402 - SS not available;
- 1403 - SS subscription violation;
- 1404 - SS incompatibility;
- 1405 - negative password check;
- 1406 - password registration failure;
- 1407 - Number of Password Attempts;
- 1408 - USSD Busy;
- 1409 - Unknown Alphabet.
- 1410 - short term denial;

- 1411 - long term denial.
- 1412 For definition of these errors see GSM 04.80.
- 1413 h) Short message problem:
- 1414 - SM delivery failure with detailed reason as follows:
- 1415 - memory capacity exceeded;
- 1416 - MS protocol error;
- 1417 - MS not equipped;
- 1418 - unknown service centre (SC);
- 1419 - SC congestion;
- 1420 - invalid SME address;
- 1421 - subscriber is not an SC subscriber;
- 1422 - and possibly detailed diagnostic information, coded as specified in TS GSM 03.40, under SMS-SUBMIT-
1423 REPORT and SMS-DELIVERY-REPORT. If the SM entity which returns the SM Delivery Failure error
1424 includes detailed diagnostic information, it shall be forwarded in the
1425 MAP_MO_FORWARD_SHORT_MESSAGE and in the MAP_MT_FORWARD_SHORT_MESSAGE
1426 response.
- 1427 - message waiting list full, i.e. no further SC address can be added to the message waiting list;
- 1428 - Subscriber busy for MT SMS, i.e. the mobile terminated short message transfer cannot be completed because:
- 1429 - another mobile terminated short message transfer is going on and the delivery node does not support
1430 message buffering; or
- 1431 - another mobile terminated short message transfer is going on and it is not possible to buffer the message
1432 for later delivery; or
- 1433 - the message was buffered but it is not possible to deliver the message before the expiry of the buffering
1434 time defined in GSM 03.40;
- Absent Subscriber SM, i.e. the mobile terminated short message transfer cannot be completed because the network cannot contact the subscriber. Diagnostic information regarding the reason for the subscriber's absence may be included with this error.

- 1435 i) Location services problem:
- 1436 - Unauthorized Requesting Network
 - 1437 - Unauthorized LCS Client with detailed reason as follows
 - 1438 - Unauthorized Privacy Class
 - 1439 - Unauthorized Call Unrelated External Client
 - 1440 - Unauthorized Call Related External Client
 - 1441 - Privacy override not applicable
 - 1442 - Position method failure with detailed reason as follows:
 - 1443 - Congestion
 - 1444 - Insufficient resources
 - 1445 - Insufficient Measurement Data
 - 1446 - Inconsistent Measurement Data
 - 1447 - Location procedure not completed
 - 1448 - Location procedure not supported by target MS
 - 1449 - QoS not attainable
 - 1450 - Unknown or unreachable LCS Client

1451 7.6.1.5 All Information Sent

1452 This parameter indicates to the receiving entity when the sending entity has sent all necessary information.

1453 7.6.2 Numbering and identification parameter

1454 7.6.2.1 IMSI

1455 This parameter is the International Mobile Subscriber Identity defined in GSM 03.03.

1456 7.6.2.2 TMSI

1457 This parameter is the Temporary Mobile Subscriber Identity defined in GSM 03.03.

1458 7.6.2.3 IMEI

1459 This parameter is the International Mobile Equipment Identity defined in GSM 03.03.

1460 7.6.2.4 Previous location area Id

1461 This parameter refers to the identity of the location area from which the subscriber has roamed.

1462 7.6.2.5 Stored location area Id

1463 This parameter refers to the location area where the subscriber is assumed to be located.

1464 7.6.2.6 Current location area Id

1465 This parameter is used to indicate the location area in which the subscriber is currently located.

1466 7.6.2.7 Target location area Id

1467 This parameter refers to the location area into which the subscriber intends to roam.

1468 7.6.2.8 Target cell Id

1469 This parameter refers to the identity of the cell to which a call has to be handed over.

1470 7.6.2.9 Void

1471 7.6.2.10 Originating entity number

1472 This parameter refers to an application layer identification of a system component in terms of its associated ISDN
1473 number.

1474 7.6.2.11 MSC number

1475 This parameter refers to the ISDN number of an MSC.

1476 7.6.2.12 Target MSC number

1477 This parameter refers to the ISDN number of an MSC to which a call has to be handed over.

1478 7.6.2.13 HLR number

1479 This parameter refers to the ISDN number of an HLR.

1480 7.6.2.14 VLR number

1481 This parameter refers to the ISDN number of a VLR.

1482 7.6.2.15 HLR Id

1483 This parameter refers to the identity of an HLR derived from the IMSI defined in CCITT Recommendation E.212.

1484 7.6.2.16 LMSI

1485 This parameter refers to a local identity allocated by the VLR to a given subscriber for internal management of data in
1486 the VLR. LMSI shall not be sent to the SGSN.

1487 7.6.2.17 MS ISDN

1488 This parameter refers to one of the ISDN numbers assigned to a mobile subscriber in accordance with CCITT
1489 Recommendation E.213.

1490 7.6.2.18 OMC Id

1491 This parameter refers to the identity of an operation and maintenance centre.

1492 7.6.2.19 Roaming number

1493 This parameter refers to the roaming number as defined in CCITT Recommendation E.213.

- 1494 **7.6.2.20** **Void**
- 1495 **7.6.2.21** **Handover number**
- 1496 This parameter refers to the number used for routing a call between MSCs during handover.
- 1497 **7.6.2.22** **Forwarded-to number**
- 1498 This parameter refers to the address to which a call is to be forwarded. This may include a subaddress. For subscribers
1499 having an originating CAMEL Phase 2 or higher subscription, this address need not be in non-E.164 international
1500 format.
- 1501 **7.6.2.23** **Forwarded-to subaddress**
- 1502 This parameter refers to the sub-address attached to the address to which a call is to be forwarded.
- 1503 **7.6.2.24** **Called number**
- 1504 This parameter refers to a called party number as defined in CCITT Recommendation Q.767.
- 1505 **7.6.2.25** **Calling number**
- 1506 This parameter refers to a calling party number as defined in CCITT Recommendation Q.767.
- 1507 **7.6.2.26** **Originally dialled number**
- 1508 This parameter refers to the number dialled by the calling party in order to reach a mobile subscriber.
- 1509 **7.6.2.27** **Service centre address**
- 1510 This parameter represents the address of a Short Message Service Centre.
- 1511 **7.6.2.28** **Zone Code**
- 1512 This parameter is used to define location areas into which the subscriber is allowed or not allowed to roam (regional
1513 subscription). With a complete list of Zone Codes the VLR or the SGSN is able to determine for all its location areas
1514 whether roaming is allowed or not.
- 1515 **7.6.2.29** **MSISdn-Alert**
- 1516 This parameter refers to the MSISDN stored in a Message Waiting Data File in the HLR. It is used to alert the Service
1517 Centre when the MS is again attainable.
- 1518 **7.6.2.30** **Location Information**
- 1519 This parameter indicates the location of the served subscriber as defined in GSM 03.18.
- 1520 **7.6.2.31** **GMSC Address**
- 1521 This parameter refers to the E.164 address of a GMSC.
- 1522 **7.6.2.32** **VMSC Address**
- 1523 This parameter refers to the E.164 address of a VMSC.

1524 7.6.2.33 Group Id

1525 This parameter is used to describe groups a subscriber can be member of. A subscriber can partake in all group calls
1526 (VBS/VGCS) where he subscribed to the respective groups.

1527 7.6.2.34 North American Equal Access preferred Carrier Id

1528 This parameter refers to the carrier identity preferred by the subscriber for calls requiring routing via an interexchange
1529 carrier. This identity is used at:

- 1530 - outgoing calls: when the subscriber does not specify at call setup a carrier identity;
- 1531 - forwarded calls: when a call is forwarded by the subscriber;
- 1532 - incoming calls: applicable to the roaming leg of the call.

1533 7.6.2.35 SIWFS Number

1534 This parameter refers to the number used for routing a call between the MSC and the SIWFS (used by ISUP).

1535 7.6.2.36 B-subscriber address

1536 This parameter refers to the address used by the SIWFS to route the outgoing call from the SIWFS to either the B-
1537 subscriber in case the non-loop method or back to the VMSC in case of the loop method.

1538 7.6.2.37 Serving cell Id

1539 This parameter indicates the cell currently being used by the served subscriber.

1540 7.6.2.38 SGSN number

1541 This parameter refers to the ISDN number of a SGSN.

1542 7.6.2.39 SGSN address

1543 This parameter refers to the IP-address of a SGSN. This parameter is defined in GSM 03.03.

1544 7.6.2.40 GGSN address

1545 This parameter refers to the IP-address of a GGSN. This parameter is defined in GSM 03.03.

1546 7.6.2.41 GGSN number

1547 This parameter refers to the ISDN number of a GGSN or the ISDN number of the protocol-converter if a protocol-
1548 converting GSN is used between the GGSN and the HLR..

1549 7.6.2.42 APN

1550 This parameter refers to the DNS name of a GGSN. This parameter is defined in GSM 03.60.

1551 7.6.2.43 Network Node number

1552 This parameter refers either to the ISDN number of SGSN or to the ISDN number of MSC.

1553 7.6.2.44 PDP-Type

1554 This parameter indicates which type of protocol is used by the MS as defined in GSM 03.60.

1555 7.6.2.45 PDP-Address

1556 This parameter indicates the address of the data protocol as defined in GSM 03.60.

1557 7.6.2.46 Additional number

1558 This parameter can refer either to the SGSN number or to the MSC number.

1559 7.6.2.47 P-TMSI

1560 This parameter is the Packet Temporary Mobile Subscriber Identity defined in GSM 03.03.

1561 7.6.2.48 B-subscriber number

1562 This parameter refers to the number of the destination B dialled by the A user. This may include a subaddress.

1563 7.6.2.49 B-subscriber subaddress

1564 This parameter refers to the sub-address attached to the destination B dialled by the A user.

1565 7.6.2.50 LMU Number

1566 This parameter refers to a local number assigned to an LMU by an SMLC.

1567 7.6.2.51 MLC Number

1568 This parameter refers to the ISDN (E.164) number of an MLC.

1569 7.6.3 Subscriber management parameters**1570 7.6.3.1 Category**

1571 This parameter refers to the calling party category as defined in CCITT Recommendation Q.767.

1572 7.6.3.2 Equipment status

1573 This parameter refers to the status of the mobile equipment as defined in GSM 02.16.

1574 7.6.3.3 Extensible Bearer service

1575 This parameter may refer to a single bearer service, a set of bearer services or to all bearer services as defined in TS
1576 GSM 02.02. This parameter is used only for subscriber profile management. Extensible Bearer service values include all
1577 values defined for a Bearer service parameter (7.6.4.38).

1578 7.6.3.4 Extensible Teleservice

1579 This parameter may refer to a single teleservice, a set of teleservices or to all teleservices as defined in TS GSM 02.03.
1580 This parameter is used only for subscriber profile management. Extensible Teleservice values include all values defined
1581 for a Teleservice parameter (7.6.4.39).

1582 7.6.3.5 Extensible Basic Service Group

1583 This parameter refers to the Basic Service Group either as an extensible bearer service (see subclause 7.6.3.3) or an
1584 extensible teleservice (see subclause 7.6.3.4). This parameter is used only for subscriber profile management. The null
1585 value (i.e. neither extensible bearer service nor extensible teleservice) is used to denote the group containing all
1586 extensible bearer services and all extensible teleservices.

1587 7.6.3.6 GSM bearer capability

1588 This parameter refers to the GSM bearer capability information element defined in GSM 04.08.

1589 7.6.3.7 Subscriber Status

1590 This parameter refers to the barring status of the subscriber:

1591 - service granted;

1592 - Operator Determined Barring.

1593 7.6.3.8 CUG Outgoing Access indicator

1594 This parameter represents the Outgoing Access as defined in ETS 300 136.

1595 7.6.3.9 Operator Determined Barring General Data

1596 This parameter refers to the set of subscribers features that the network operator or the service provider can regulate.

1597 This set only includes those limitations that can be controlled in the VLR or in the SGSN:

1598 - All outgoing calls barred; (*)

1599 - International outgoing calls barred; (*)

1600 - International outgoing calls except those to the home PLMN country barred; (*)

1601 - Interzonal outgoing calls barred; (*)

1602 - Interzonal outgoing calls except those to the home PLMN country barred; (*)

1603 - Interzonal outgoing calls AND international outgoing calls except those directed to the home PLMN country
1604 barred; (*)

1605 - Premium rate (information) outgoing calls barred;

1606 - Premium rate (entertainment) outgoing calls barred;

1607 - Supplementary service access barred;

1608 - Invocation of call transfer barred;

1609 - Invocation of chargeable call transfer barred;

1610 - Invocation of internationally chargeable call transfer barred;

1611 - Invocation of interzonally chargeable call transfer barred;

1612 - Invocation of call transfer where both legs are chargeable barred.

1613 (*) Only these ODBs are supported by the SGSN. The SGSN applies them only for short message transfer.

1614 7.6.3.10 ODB HPLMN Specific Data

1615 This parameter refers to the set of subscribers features that the network operator or the service provider can regulate

1616 only when the subscriber is registered in the HPLMN. This set only includes those limitations that can be controlled in
1617 the VLR or in the SGSN:

1618 - Operator Determined Barring Type 1;

1619 - Operator Determined Barring Type 2;

1620 - Operator Determined Barring Type 3;

1621 - Operator Determined Barring Type 4.

1622 7.6.3.11 Regional Subscription Data

1623 This parameter defines the regional subscription area in which the subscriber is allowed to roam. It consists of a list of
1624 Zone Codes (see subclause 7.6.2.28).

1625 7.6.3.12 Regional Subscription Response

1626 This parameter indicates either that the regional subscription data cannot be handled or that the current MSC or SGSN
1627 area is entirely restricted because of regional subscription.

1628 7.6.3.13 Roaming Restriction Due To Unsupported Feature

1629 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
1630 feature or service is indicated as unsupported by the VLR.

1631 7.6.3.14 Extensible SS-Info

1632 This parameter refers to all the information related to a supplementary service and is a choice between:

- 1633 - extensible forwarding information (see subclause 7.6.3.15);
- 1634 - extensible call barring information (see subclause 7.6.3.20);
- 1635 - CUG info (see subclause 7.6.3.22);
- 1636 - extensible SS-Data (see subclause 7.6.3.29).

1637 7.6.3.15 Extensible Forwarding information

1638 This parameter represents the information related to each call forwarding service:

- 1639 - the SS-Code of the relevant call forwarding service (see subclause 7.6.4.1);
- 1640 - if required, a list of extensible forwarding feature parameters (see subclause 7.6.3.16).

1641 The list may contain one item per Basic Service Group.

1642 7.6.3.16 Extensible Forwarding feature

1643 This parameter applies to each combination of call forwarding service and Basic Service Group and contains the
1644 following information, as required:

- 1645 - extensible Basic Service Group (see subclause 7.6.3.5);
- 1646 - extensible SS-Status (see subclause 7.6.3.17);
- 1647 - forwarded-to number (see subclause 7.6.2.22);
- 1648 - forwarded-to subaddress (see subclause 7.6.2.23);
- 1649 - extensible forwarding options (see subclause 7.6.3.18);
- 1650 - extensible no reply condition timer (see subclause 7.6.4.19).

1651 7.6.3.17 Extensible SS-Status

1652 This parameter refers to the state information of individual supplementary services as defined in TS GSM 03.11.

1653 7.6.3.18 Extensible Forwarding Options

1654 This parameter refers to a set of forwarding options attached to a supplementary service. It contains the following
1655 informations:

- 1656 - notification to forwarding party (see TS GSM 02.82 for the meaning of this parameter);
- 1657 - redirection notification to the forwarded-to party (see TS GSM 02.82 for the meaning of this parameter);
- 1658 - notification to calling party (see TS GSM 02.82 for the meaning of this parameter);
- 1659 - redirecting presentation (see TS GSM 02.82 for the meaning of this parameter);
- 1660 - Forwarding reason (see TS GSM 02.82 for the meaning of this parameter).

1661 7.6.3.19 Extensible No reply condition timer

1662 This parameter refers to the extensible no reply condition timer for call forwarding on no reply.

1663 7.6.3.20 Extensible Call barring information

1664 This parameter contains for each call barring service:

- 1665 - SS-Code (see subclause 7.6.4.1);
 - 1666 - a list of extensible call barring feature parameters (see subclause 7.6.3.21).
- 1667 The list may contain one item per Basic Service Group.

1668 7.6.3.21 Extensible Call barring feature

1669 This parameter gives the status of call barring services as applicable to each Basic Service Group. The parameter
1670 contains the following information:

- 1671 - Extensible Basic Service Group (see subclause 7.6.3.5);
- 1672 - provisioned SS-Status (see subclause 7.6.3.17).

1673 7.6.3.22 CUG info

1674 This parameter refers to the overall information required for operation for each CUG:

- 1675 - CUG subscriptionList;
- 1676 - CUG featureList.

1677 7.6.3.23 CUG subscription

1678 This parameter refers to the set of basic information for each CUG defined in that subscription. The following
1679 information is stored:

- 1680 - CUG index;
- 1681 - CUG interlock;
- 1682 - Intra CUG restrictions;
- 1683 - Basic Service Group List.

1684 7.6.3.24 CUG interlock

1685 This parameter represents the CUG interlock code defined in ETS 300 138.

1686 **7.6.3.25 CUG index**

1687 This parameter represents the CUG index defined in ETS 300 138.

1688 **7.6.3.26 CUG feature**

1689 This parameter contains two parameters which are associated with the Basic Service Group. If the Basic Service Group
1690 Code is not present the feature applies to all Basic Services. The following parameters are included:

1691 - Preferential CUG indicator:

1692 indicates which CUG index is to be used at outgoing call set-up using the associated Basic Service Group;

1693 - Inter CUG Option:

1694 describes whether it for the associated Basic Service Group is allowed to make calls outside the CUG and
1695 whether incoming calls are allowed;

1696 - Basic Service Group.

1697 See TS GSM 02.85 for meaning of this parameter.

1698 **7.6.3.27 Inter CUG options**

1699 This parameter indicates the subscribers ability to make and receive calls outside a specific closed user group. It takes
1700 any of the following values:

1701 - CUG only facility (only calls within CUG are allowed);

1702 - CUG with outgoing access (calls outside CUG allowed);

1703 - CUG with incoming access (calls from outside CUG into CUG allowed);

1704 - CUG with both incoming and outgoing access (all calls allowed).

1705 **7.6.3.28 Intra CUG restrictions**

1706 This parameter describes whether or not the subscriber is allowed to originate calls to or to receive calls from within the
1707 CUG. It can take any of the following values:

1708 - no CUG restrictions;

1709 - CUG incoming calls barred;

1710 - CUG outgoing calls barred.

1711 **7.6.3.29 Extensible SS-Data**

1712 This parameter refers to the necessary set of information required in order to characterise one supplementary service:

1713 - SS-Code (see subclause 7.6.4.1);

1714 - Extensible SS-Status (if applicable) (see subclause 7.6.3.17);

1715 - Extensible Override subscription option (if applicable) (see subclause 7.6.3.30);

1716 - Extensible CLI Restriction (if applicable) (see subclause 7.6.3.31);

1717 - Extensible Basic Service Group Code (see subclause 7.6.3.5).

1718 **7.6.3.30 Subscriber State**

1719 This parameter indicates the state of the MS as defined in GSM 03.18.

1720 7.6.3.31 Requested Info

1721 This parameter indicates the subscriber information being requested as defined in GSM 03.18.

1722 7.6.3.32 Suppression of Announcement

1723 This parameter indicates if the announcement or tones shall be suppressed as defined in 3G TS 23.078.

1724 7.6.3.33 Suppress T-CSI

1725 This parameter is used to suppress the invocation of terminating CAMEL services.

1726 7.6.3.34 GMSC CAMEL Subscription Info

1727 This parameter contains CAMEL subscription information, i.e.O-CSI and/or D-CSI and/or T-CSI, which indicates to the
1728 GMSC that originating and/or terminating CAMEL services shall be invoked for the incoming call.

1729 7.6.3.35 VLR CAMEL Subscription Info

1730 This parameter identifies the subscriber as having CAMEL services which are invoked in the MSC or VLR.

1731 7.6.3.36 Supported CAMEL Phases in the VLR

1732 This parameter indicates which phases of CAMEL are supported in the VLR.

1733 7.6.3.36A Supported CAMEL Phases in the SGSN

1734 This parameter indicates which phases of CAMEL are supported in the SGSN.

1735 7.6.3.37 CUG Subscription Flag

1736 This parameter indicates a that a subscriber with a T-CSI also has a CUG subscription. It is defined in 3G TS 23.078.

1737 7.6.3.38 CAMEL Subscription Info Withdraw

1738 This parameter indicates that CAMEL Subscription Info shall be deleted from the VLR or SGSN.

1739 7.6.3.39 Voice Group Call Service (VGCS) Data

1740 This parameter refers to one or more groups a subscriber may be member of for voice group calls.

1741 7.6.3.40 Voice Broadcast Service (VBS) Data

1742 This parameter refers to one or more groups a subscriber may be member of for the voice broadcast service. Per group it
1743 is further indicated whether the subscriber is only allowed to listen to respective group calls or whether he is in addition
1744 entitled to initiate respective voice broadcast calls.

1745 7.6.3.41 ISDN bearer capability

1746 This parameter refers to the ISDN bearer capability information element defined in GSM 09.07.

1747 7.6.3.42 Lower layer Compatibility

1748 This parameter refers to the lower layer compatibility information element defined in GSM 04.08.

1749 7.6.3.43 High Layer Compatibility

1750 This parameter refers to the high layer compatibility information element defined in GSM 04.08.

1751 7.6.3.44 Alerting Pattern

1752 This parameter is an indication that can be used by the MS to alert the user in a specific manner in case of mobile
1753 terminating traffic (switched call or USSD). That indication can be an alerting level or an alerting category.

1754 7.6.3.45 GPRS Subscription Data Withdraw

1755 This parameter indicates that GPRS Subscription Data shall be deleted from the SGSN.

1756 7.6.3.46 GPRS Subscription Data

1757 This parameter refers to the list of PDP-Contexts that subscriber has subscribed to.

1758 7.6.3.47 QoS-Subscribed

1759 This parameter indicates the quality of service subscribed for a certain service. It is defined in GSM 03.60.

1760 7.6.3.48 VPLMN address allowed

1761 This parameter specifies whether the MS is allowed to use a dynamic address allocated in the VPLMN. It is defined in
1762 GSM 03.60.

1763 7.6.3.49 Roaming Restricted In SGSN Due To Unsupported Feature

1764 This parameter defines that a subscriber is not allowed to roam in the current SGSN area. It may be used by the HLR if a
1765 feature or service is indicated as unsupported by the SGSN.

1766 7.6.3.50 Network Access Mode

1767 This parameter is defined in GSM 03.08.

1768 7.6.3.51 Mobile Not Reachable Reason

1769 This parameter stores the reason for the MS being absent when an attempt to deliver a short message to an MS fails at
1770 the MSC, SGSN or both. It is defined in TS GSM 03.40.

1771 7.6.3.52 Cancellation Type

1772 This parameter indicates the reason of location cancellation. It is defined in TS GSM 03.60.

1773 7.6.3.53 All GPRS Data

1774 This parameter indicates to the SGSN that all GPRS Subscription Data shall be deleted for the subscriber.

1775 7.6.3.54 Complete Data List Included

1776 This parameter indicates to the SGSN that the complete GPRS Subscription Data stored for the Subscriber shall be
1777 replaced with the GPRS Subscription Data received.

1778 7.6.3.55 PDP Context Identifier

1779 This parameter is used to identify a PDP context for the subscriber.

1780 7.6.3.56 LSA Information

1781 This parameter refers to one or more localised service areas a subscriber may be a member of, together with the priority
1782 of each localised service area. The access right outside these localised service areas is also indicated.

1783 7.6.3.57 SoLSA support indicator

1784 This parameter indicates that the VLR or the SGSN supports SoLSA subscription.

1785 7.6.3.58 LSA Information Withdraw

1786 This parameter indicates that LSA information shall be deleted from the VLR or the SGSN.

1787 7.6.3.59 LMU Indicator

1788 This parameter indicates the presence of an LMU.

1789 7.6.3.60 LCS Information

1790 This parameter defines the LCS related information for an MS subscriber and contains the following components:

- 1791 - GMLC List (see subclause 7.6.3.61)
- 1792 - LCS Privacy Exception List (see subclause 7.6.3.62)
- 1793 - MO-LR List (see subclause 7.6.3.65A)

1794 7.6.3.61 GMLC List

1795 This parameter contains the addresses of all GMLCs in the MS subscriber's HPLMN that are permitted to issue a non-call related MT-LR location request for this MS. Usage of this parameter is defined in GSM 03.71.

1797 7.6.3.62 LCS Privacy Exception List

1798 This parameter defines the classes of LCS Client that are allowed to locate any target MS. For each class, the following information is provided:

- 1800 - SS-Code (see subclause 7.6.4.1);
- 1801 - a list of LCS privacy exception parameters (see subclause 7.6.3.63).

1802 7.6.3.63 LCS Privacy Exception Parameters

1803 This parameter gives the status of each LCS privacy exception class and any additional parameters relevant to this class.
1804 The parameter contains the following information:

- 1805 - provisioned SS-Status (see subclause 7.6.3.17);
- 1806 - privacy verification by MS user (see subclause 7.6.3.65B);
- 1807 - external client List (see subclause 7.6.3.64);
- 1808 - internal client List (see subclause 7.6.3.65)

1809 7.6.3.64 External Client List

1810 This parameter is only applicable to the non-call related privacy class and gives the identities of the external clients that
1811 are allowed to locate a target MS for a non-call related MT-LR. Each identity is an international (e.g.E.164) address. For
1812 each identified external client, GMLC restrictions may be defined. It may also be indicated if the MS shall be notified of
1813 a non-restricted MT-LR from each identified LCS client.and, if so, whether notification only or notification with privacy
1814 verification shall apply.Usage of this parameter is defined in GSM 03.71.

1815 7.6.3.65 Internal Client List

1816 This parameter is only applicable to the PLMN operator privacy class and gives the identities of the internal PLMN
1817 operator clients that are allowed to locate a target MS for an NI-LR or MT-LR. Usage of this parameter is defined in
1818 GSM 03.71.

1819 7.6.3.65A MO-LR List

1820 This parameter defines the classes of MO-LR for which a subscription exists for a particular MS. For each class, the
1821 following information is provided:

1822 - SS-Code (see subclause 7.6.4.1);

1823 7.6.3.65B Privacy Verification By MS User

1824 This parameter is applicable to the non-call related privacy class and indicates whether the MS user shall be notified for
1825 a non-call related MT-LR from any value added LCS client when the MT-LR is restricted and be enabled to accept or
1826 override the restriction.

1827 7.6.3.65C GMLC List Withdraw

1828 This parameter indicates whether the subscriber's LCS GMLC list shall be deleted from the VLR. The parameter does
1829 not apply to, and shall be ignored if received by, an SGSN.

1830 7.6.3.66 IST Alert Timer

1831 This parameter indicates the IST Alert Timer value that must be used in the MSC to inform the HLR about the call
1832 activities that the subscriber performs. Units are minutes.

1833 7.6.3.67 Call Termination Indicator

1834 This parameter indicates whether the MSC shall terminate a specific ongoing call, or all the call activities related to a
1835 specified subscriber.

1836 7.6.3.68 IST Information Withdraw

1837 This parameter indicates that IST information shall be deleted from the VMSC.

1838 7.6.3.69 IST Support Indicator

1839 This parameter indicates the degree of IST functionality supported by the MSC (Visited MSC or Gateway MSC). It can
1840 take one of the following values:

1841 - Basic IST functionality

1842 - IST command service (in addition to the basic IST functionality and including the ability to terminate all calls
1843 being carried for the identified subscriber).

1844 7.6.3.70 Super-Charger Supported In HLR

1845 This parameter is used by the HLR to indicate support of the Super-Charger functionality and an indication of the age of
1846 the subscription data stored in the HLR.

1847 7.6.3.71 Super-Charger Supported In Serving Network Entity

1848 This parameter is used to indicate support of the Super-Charger functionality by the originating entity and to indicate
1849 either that subscription data is required or the date and time of the last know subscriber data modification.

1850 **7.6.3.72 Age Indicator**

1851 This parameter is used by the HLR to determine the validity of the subscription data retained by the serving network
1852 entity in a Super-Charged network.

1853 **7.6.3.73 GPRS enhancements support indicator**

1854 This parameter indicates to the HLR that the SGSN supports GPRS enhancements.

1855 **7.6.3.74 Extensible QoS-Subscribed**

1856 This parameter indicates the enhanced QoS subscribed for a certain service. It is defined in 3G TS 23.060.

1857 **7.6.3.75 SGSN Camel Subscription Info**

1858 This parameter identifies the subscriber as having CAMEL services which are invoked in the SGSN.

1859 **7.6.3.76 SMS-CSI**

1860 This parameter identifies the subscriber as having SMS CAMEL services as defined in 3G TS 23.078

1861 **7.6.3.77 GPRS-CSI**

1862 This parameter identifies the subscriber as having GPRS CAMEL services as defined in 3G TS 23.078

1863 **7.6.3.78 CAMEL subscription info**

1864 This parameter indicates the CSI that can be controlled by CSE.

1865 **7.6.3.79 Extensible Call barring information for CSE**

1866 This parameter contains for each call barring service for CSE:

- 1867 - SS-Code;
- 1868 - a list of extensible call barring feature parameters.
1869 The list may contain one item per Basic Service Group.
- 1870 - password.
- 1871 - wrong password attempt counter.
- 1872 - notification-to-CSE flag.

1873 **7.6.3.80 Extensible Forwarding information for CSE**

1874 This parameter represents the information for CSE related to each call forwarding service:

- 1875 - the SS-Code of the relevant call forwarding service;
- 1876 - if required, a list of extensible forwarding feature parameters.
1877 The list may contain one item per Basic Service Group.
- 1878 - notification-to-CSE flag.

1879 **7.6.3.81 Modification Request for CSI**

1880 This parameter indicates the CAMEL subscription information to be modified by CSE.

1881 **7.6.3.82 Modification Request for SS Information**

1882 This parameter indicates the call forwarding and call barring supplementary service data to be modified by CSE.

1883 **7.6.3.83 Call Barring Data**

1884 This parameter contains the extensible call barring feature list (see subclause 7.6.3.21) and Notification to CSE flag.

1885 **7.6.3.84 Call Forwarding Data**

1886 This parameter contains the extensible call forwarding feature list (see subclause 7.6.3.16) and Notification to CSE flag.

1887 **7.6.3.85 ODB Data**

1888 This parameter contains the ODB general data, ODB HPLMN specific data and Notification to CSE flag.

1889 **7.6.3.86 Requested Subscription Info**

1890 This parameter indicates the subscription information being requested.

1891 **7.6.4 Supplementary services parameters**

1892 **7.6.4.1 SS-Code**

1893 This parameter may refer to one supplementary service or a set of supplementary services as defined in TS GSM 02.04.

1894 For MAP Release '98 this includes:

- 1895 - Calling Line Identification Presentation service (CLIP);
- 1896 - Calling Line Identification Restriction service (CLIR);
- 1897 - Connected Line Identification Presentation service (COLP);
- 1898 - Connected Line Identification Restriction service (COLR);
- 1899 - Calling Name Presentation (CNAP)
- 1900 - All Call Forwarding services;
- 1901 - Call Waiting (CW);
- 1902 - Call Hold (HOLD);
- 1903 - Multi-Party service (MPTY);
- 1904 - Closed User Group (CUG);
- 1905 - All Charging services;
- 1906 - All Call Restriction services;
- 1907 - Explicit Call Transfer service (ECT);
- 1908 - enhanced Multi-Level Precedence and Pre-emption service (eMLPP);
- 1909 - Completion of Calls to Busy Subscriber, originating side (CCBS-A);
- 1910 - Completion of Calls to Busy Subscriber, destination side (CCBS-B);

1911 - All LCS privacy exceptions (see subclause 7.6.4.44);

1912 - Mobile Originating Location Request (MO-LR) (see subclause 7.6.4.44A).

1913 7.6.4.2 SS-Status

1914 This parameter refers to the state information of individual supplementary services as defined in GSM 03.11.

1915 7.6.4.3 SS-Data

1916 This parameter refers to the necessary set of information required in order to characterise one supplementary service:

1917 - SS-Code (see subclause 7.6.4.1);

1918 - SS-Status (if applicable) (see subclause 7.6.4.2);

1919 - Override subscription option (see subclause 7.6.4.4);

1920 - CLI Restriction (see subclause 7.6.4.5);

1921 - Basic Service Group Code (see subclause 7.6.4.40).

1922 7.6.4.4 Override Category

1923 This parameter refers to the subscription option Override Category attached to a supplementary service. It can take the
1924 following two values:

1925 - Enabled;

1926 - Disabled.

1927 7.6.4.5 CLI Restriction Option

1928 This parameter refers to the subscription option Restriction mode attached to the CLIR supplementary service. It can
1929 take the following three values:

1930 - Permanent;

1931 - Temporary (Default Restricted);

1932 - Temporary (Default Allowed).

1933 7.6.4.6 Forwarding Options

1934 This parameter refers to a forwarding option attached to a supplementary service. It can take one of the following
1935 values:

1936 - notification to forwarding party (see GSM 02.82 for the meaning of this parameter);

1937 - notification to calling party (see GSM 02.82 for the meaning of this parameter);

1938 - redirecting presentation (see GSM 02.82 for the meaning of this parameter);

1939 - Forwarding reason (see GSM 02.82 for the meaning of this parameter).

1940 7.6.4.7 No reply condition timer

1941 This parameter refers to the no reply condition timer for call forwarding on no reply.

1942 7.6.4.8 - 7.6.4.14 Void

1943 7.6.4.15 Forwarding information

1944 This parameter represents the information related to each call forwarding service:

- 1945 - the SS-Code of the relevant call forwarding service (see subclause 7.6.4.1);
 - 1946 - if required, a list of forwarding feature parameters (see subclause 7.6.4.16).
- 1947 The list may contain one item per Basic Service Group.

1948 7.6.4.16 Forwarding feature

1949 This parameter applies to each combination of call forwarding service and Basic Service Group and contains the
1950 following information, as required:

- 1951 - Basic Service Group (see subclause 7.6.4.40);
- 1952 - SS-Status (see subclause 7.6.4.2);
- 1953 - forwarded-to number (see subclause 7.6.2.22);
- 1954 - forwarded-to subaddress (see subclause 7.6.2.23);
- 1955 - forwarding options (see subclause 7.6.4.6);
- 1956 - no reply condition timer (see subclause 7.6.4.7).

1957 7.6.4.17 Void

1958 7.6.4.18 Call barring information

1959 This parameter contains for each call barring service:

- 1960 - SS-Code (see subclause 7.6.4.1);
 - 1961 - a list of call barring feature parameters (see subclause 7.6.4.19).
- 1962 The list may contain one item per Basic Service Group.

1963 7.6.4.19 Call barring feature

1964 This parameter gives the status of call barring services as applicable to each Basic Service Group. The parameter
1965 contains the following information:

- 1966 - Basic Service Group (see subclause 7.6.4.40);
- 1967 - SS-Status (see subclause 7.6.4.2).

1968 7.6.4.20 New password

1969 This parameter refers to the password which the subscriber just registered in the network.

1970 This parameter refers to a password used by the subscriber for supplementary service control.

1971 7.6.4.21 Current password

1972 This parameter refers to a password used by the subscriber for supplementary service control.

1973 7.6.4.22 Guidance information

1974 This parameter refers to guidance information given to a subscriber who is requested to provide a password. One of the
1975 following information may be given:

1976 - "enter password";

1977 This information is used for checking of the old password.

1978 - "enter new password";

1979 This information is used during password registration for the request of the first new password.

1980 - "enter new password again";

1981 This information is used during password registration for the request of the new password again for verification.

1982 7.6.4.23 Void**1983 7.6.4.24 SS-Info**

1984 This parameter refers to all the information related to a supplementary service and is a choice between:

1985 - forwarding information (see subclause 7.6.4.15);

1986 - call barring information (see subclause 7.6.4.18);

1987 - CUG info (see subclause 7.6.4.8);

1988 - SS-Data (see subclause 7.6.4.3).

1989 - eMLPP information (see subclause 7.6.4.41).

1990 7.6.4.25 - 7.6.4.35 Void**1991 7.6.4.36 USSD Data Coding Scheme**

1992 This parameter contains the information of the alphabet and the language used for the unstructured information in an
1993 Unstructured Supplementary Service Data operation. The coding of this parameter is according to the Cell Broadcast
1994 Data Coding Scheme as specified in GSM 03.38.

1995 7.6.4.37 USSD String

1996 This parameter contains a string of unstructured information in an Unstructured Supplementary Service Data operation.
1997 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
1998 network as specified in GSM 02.90.

1999 7.6.4.38 Bearer service

2000 This parameter may refer to a single bearer service, a set of bearer services or to all bearer services as defined in TS
2001 GSM 02.02. This parameter is used only for supplementary service management.

2002 7.6.4.39 Teleservice

2003 This parameter may refer to a single teleservice, a set of teleservices or to all teleservices as defined in TS GSM 02.03.
2004 This parameter is used only for supplementary service management.

2005 7.6.4.40 Basic Service Group

2006 This parameter refers to the Basic Service Group either as a bearer service (see subclause 7.6.4.38) or a teleservice (see
2007 subclause 7.6.4.39). This parameter is used only for supplementary service management. The null value (i.e. neither
2008 bearer service nor teleservice) is used to denote the group containing all bearer services and all teleservices.

2009 7.6.4.41 eMLPP information

2010 This parameter contains two parameters which are associated with the eMLPP service. The following two parameters
2011 are included:

2012 - maximum entitled priority:

2013 indicates the highest priority level the subscriber is allowed to apply for an outgoing call set-up;

2014 - default priority:

2015 defines the priority level which shall be assigned to a call if no explicit priority is indicated during call set-up.

2016 7.6.4.42 SS-event

2017 This parameter indicates the Supplementary Service for which an invocation notification is sent towards the gsmSCF. It
2018 can indicate one of the following services:

2019 - Explicit Call Transfer (ECT)

2020 - Call Deflection (CD)

2021 - Multi-Party call (MPTY)

2022 - Completion of Calls to Busy Subscriber (CCBS)

2023 7.6.4.43 SS-event data

2024 This parameter contains additional information related to Supplementary Service invocation. Depending on the service
2025 invoked it can contain the following information:

2026 ECT A list with all Called Party Numbers involved.

2027 CDThe called Party number involved.

2028 7.6.4.44 LCS Privacy Exceptions

2029 Distinct SS codes are assigned to the following classes of LCS client in a target MS subscriber's privacy exception list.

2030 - Universal Class

2031 - Call related value added class

2032 - Non-Call related value added class

2033 - PLMN operator class

2034 7.6.4.44A Mobile Originating Location Request (MO-LR)

2035 Distinct SS codes are assigned to the following classes of MO-LR:

2036 - Basic Self Location

2037 - Autonomous Self Location

2038 - Transfer to Third Party

2039 7.6.5 Call parameters

2040 7.6.5.1 Call reference number

2041 This parameter refers to a call reference number allocated by a call control MSC.

2042 7.6.5.2 Interrogation type

2043 This parameter refers to the type of interrogation for routing information which is sent from a GMSC to an HLR. It can
2044 take either of two values:

- 2045 - basic call (for information to route a call before the call has been extended to the VMSC of the called party);
- 2046 - forwarding (for information to route the call to the forwarded-to destination after the VMSC of the forwarding
2047 party has requested the GMSC to resume handling of the call.

2048 7.6.5.3 OR interrogation

2049 This parameter indicates that the GMSC which interrogated the HLR for routing information is not in the same PLMN
2050 as the HLR, and therefore that the call will potentially be optimally routed.

2051 7.6.5.4 OR capability

2052 This parameter indicates the phase of OR which the GMSC supports.

2053 7.6.5.5 Forwarding reason

2054 This parameter indicates the reason for which the call is to be forwarded. It can take one of three values:

- 2055 - busy subscriber;
- 2056 - mobile subscriber not reachable;
- 2057 - no subscriber reply.

2058 7.6.5.6 Forwarding interrogation required

2059 This parameter indicates that if the VMSC of the forwarding subscriber requests the GMSC to resume handling of the
2060 call the GMSC shall interrogate the HLR for forwarding information.

2061 7.6.5.7 O-CSI

2062 This parameter identifies the subscriber as having originating CAMEL services as defined in TS 3G TS 23.078.

2063 7.6.5.7A D-CSI

2064 This parameter identifies the subscriber as having originating CAMEL dialled services as defined in TS 3G TS 23.078.

2065 7.6.5.7B T-CSI

2066 This parameter identifies the subscriber as having terminating CAMEL services in the GMSC, as defined in TS 3G TS
2067 23.078.

2068 7.6.5.7C VT-CSI

2069 This parameter identifies the subscriber as having terminating CAMEL services in the VMSC, as defined in TS 3G TS
2070 23.078.

2071 7.6.5.8 Call Direction

2072 This parameter is used to indicate the direction of the call.

2073 7.6.5.9 Channel Type

2074 This parameter is the result of a Channel Mode Modification for TS61/62. It contains the changed Air Interface User
2075 Rate. The information is sent from the SIWFS to the MSC to assign the correct radio resource. This parameter is defined
2076 in GSM 08.08.

2077 7.6.5.10 Chosen Channel

2078 This parameter is sent from the MSC to the SIWFS to adjust the interworking unit to the assigned radio resources. This
2079 parameter is defined in GSM 08.08.

2080 7.6.5.11 CCBS Feature

2081 This parameter corresponds to the 'CCBS Description' parameter in 3G TS 23.093. It refers to the necessary set of
2082 information required in order to characterise a certain CCBS request. The parameter may contain the following
2083 information:

2084 - CCBS Index (see 3G TS 23.093 for the use of this parameter);

2085 - B-subscriber number (see subclause 7.6.2.48);

2086 - B-subscriber subaddress (see subclause 7.6.2.49);

2087 - Basic Service Group Code (see subclause 7.6.4.40).

2088 7.6.5.12 UU Data

2089 This parameter includes User-To-User Data. It is defined in GSM 03.87.

2090 7.6.5.13 UUS CF Interaction

2091 This parameter indicates if the call forwarding or call deflection has been activated after UUS1 request has been
2092 accepted. It is defined in GSM 03.87.

2093 7.6.5.14 Number Portability Status

2094 This parameter indicates the number portability status of subscriber. See GSM 03.66.

2095 7.6.5.15 Pre-paging supported

2096 This parameter indicates that the entity which sent it supports pre-paging.

2097 7.6.6 Radio parameters**2098 7.6.6.1 - 7.6.6.6 Void****2099 7.6.6.7 HO-Number Not Required**

2100 This parameter indicates that no handover number allocation is necessary.

2101 7.6.7 Authentication parameters

2102 7.6.7.1 Authentication set list

2103 This parameter represents a list of sets of authentication parameters for a given subscriber.

2104 The list either contains Authentication Triplets (Rand, Sres, Kc) or Authentication Quintuplets (Rand, Xres, Ck, Ik,

2105 Autn). If the list contains Authentication Quintuplets, the order of sequence in this list is chronological, the first

2106 quintuplet in the list is the oldest one.

2107 7.6.7.2 Rand

2108 This parameter represents a random number used for authentication.

2109 7.6.7.3 Sres

2110 This parameter represents the response to an authentication request.

2111 7.6.7.4 Kc

2112 This parameter refers to a key used for ciphering purposes.

2113 7.6.7.5 Xres

2114 This parameter represents the response to an UMTS authentication request.

2115 7.6.7.5A Ck

2116 This parameter refers to a key used for UMTS ciphering purposes.

2117 7.6.7.5B Ik

2118 This parameter refers to the Integrity Key.

2119 7.6.7.5C Autn

2120 This parameter refers to the Authentication Token.

2121 7.6.7.6 Cksn

2122 This parameter refers to a ciphering key sequence number.

2123 7.6.7.6A Ksi

2124 This parameter refers to a key set identifier.

2125 7.6.7.6B Auts

2126 This parameter refers to the resynchronisation token.

2127 7.6.7.7 Ciphering mode

2128 This parameter refers to the ciphering mode which is associated with a radio channel. It may take values as follows:

2129 - no encryption;

2130 - identification of specific ciphering algorithm.

2131 7.6.8 Short message parameters

2132 7.6.8.1 SM-RP-DA

2133 This parameter represents the destination address used by the short message service relay sub-layer protocol. It can be
2134 either of the following:

- 2135 - IMSI (see subclause 7.6.2.1);
- 2136 - LMSI (see subclause 7.6.2.16);
- 2137 - MS-ISDN (see subclause 7.6.2.17);
- 2138 - roaming number (see subclause 7.6.2.19);
- 2139 - service centre address (see subclause 7.6.2.27).

2140 7.6.8.2 SM-RP-OA

2141 This parameter refers to the originating address used by the short message service relay sub-layer protocol. It can be
2142 either of the following:

- 2143 - MS-ISDN (see subclause 7.6.2.17);
- 2144 - service centre address (see subclause 7.6.2.27).

2145 7.6.8.3 MWD status

2146 This parameter indicates whether or not the address of the originator service centre is already contained in the Message
2147 Waiting Data file. In addition, it contains the status of the Memory Capacity Exceeded Flag (MCEF), the status of the
2148 Mobile subscriber Not Reachable Flag (MNRF) and the status of the Mobile station Not Reachable for GPRS flag
2149 (MNRG).

2150 7.6.8.4 SM-RP-UI

2151 This parameter represents the user data field carried by the short message service relay sub-layer protocol.

2152 7.6.8.5 SM-RP-PRI

2153 This parameter is used to indicate whether or not delivery of the short message shall be attempted when a service centre
2154 address is already contained in the Message Waiting Data file.

2155 7.6.8.6 SM Delivery Outcome

2156 This parameter indicates the cause for setting the message waiting data. It can take one of the following values:

- 2157 - Absent subscriber;
- 2158 - MS memory capacity exceeded;
- 2159 - Successful transfer.

2160 7.6.8.7 More Messages To Send

2161 This parameter is used to indicate whether or not the service centre has more short messages to send.

2162 7.6.8.8 Alert Reason

2163 This parameter is used to indicate the reason why the service centre is alerted. It can take one of the following values:

- 2164 - MS present;

2165 - Memory Available.

2166 7.6.8.9 Absent Subscriber Diagnostic SM

2167 This parameter is used to indicate the reason why the subscriber is absent. For the values for this parameter see TS GSM
2168 03.40.

2169 7.6.8.10 Alert Reason Indicator

2170 This parameter indicates that the alert reason is sent to the HLR due to GPRS activity.

2171 7.6.8.11 Additional SM Delivery Outcome

2172 This parameter is used to indicate the GPRS delivery outcome in case a combination between delivery outcome for
2173 GPRS and non-GPRS are sent to the HLR.

2174 7.6.8.12 Additional Absent Subscriber Diagnostic SM

2175 This parameter indicates the reason of the additional SM Delivery Outcome.

2176 7.6.8.13 Delivery Outcome Indicator

2177 This parameter indicates that the delivery outcome sent to the HLR is for GPRS.

2178 7.6.8.14 GPRS Node Indicator

2179 This parameter indicates that the Network Node Number sent by the HLR is the SGSN number.

2180 7.6.8.15 GPRS Support Indicator

2181 This parameter indicates that the SMS-GMSC supports GPRS specific procedure of combine delivery of Short
2182 Message via MSC and/or via the SGSN.

2183 7.6.8.16 SM-RP-MTI

2184 This parameter represents the RP-Message Type Indicator of the Short Message. It is used to distinguish a SM sent to
2185 the mobile station in order to acknowledge an MO-SM initiated by the mobile from a normal MT-SM. This parameter is
2186 formatted according to the formatting rules of address fields as described in GSM 03.40.

2187 7.6.8.17 SM-RP-SMEA

2188 This parameter represents the RP-Originating SME-address of the Short Message Entity that has originated the SM.
2189 This parameter is used by the short message service relay sub-layer protocol and is formatted according to the
2190 formatting rules of address fields as described in GSM 03.40.

2191 7.6.9 Access and signalling system related parameters

2192 7.6.9.1 BSS-apdu

2193 This parameter includes one or two concatenated complete 08.06 messages, as described in GSM 03.09 and
2194 GSM 09.10. The Protocol ID indicates that the message or messages are according to GSM 08.06. For the coding of the
2195 messages see GSM 08.06 and GSM 08.08.

2196 7.6.9.2 CM service type

2197 This parameter identifies the service category being requested by the subscriber:

- 2198 - mobile originating call;
- 2199 - emergency call establishment;
- 2200 - short message service;
- 2201 - mobile originating call re-establishment;
- 2202 - mobile terminating call;
- 2203 - SS request;
- 2204 - Voice group call setup;
- 2205 - Voice broadcast setup.

2206 7.6.9.3 Access connection status

2207 This parameter represents the following access connection status information:

- 2208 - RR-connection status (established/not established);
- 2209 - ciphering mode (on/off);
- 2210 - authentication status (authenticated/not authenticated).

2211 7.6.9.4 External Signal Information

2212 This parameter contains concatenated information elements (including tag and length) which are defined by a common
2213 protocol version, preceded by the associated protocol ID. It is used to transport information of the indicated protocol via
2214 MAP interfaces.

2215 7.6.9.5 Access signalling information

2216 This parameter refers to any set of information elements imported from GSM 04.08.

2217 7.6.9.6 Location update type

2218 This parameter refers to the location update type (normal, periodic or IMSI attach) contained in the GSM 04.08
2219 LOCATION REGISTRATION REQUEST message.

2220 7.6.9.7 Protocol ID

2221 This parameter refers to the protocol to which the coding of the content of the associated External Signal Information
2222 conforms.

2223 The following values are defined:

- 2224 - 04.08;
- 2225 - 08.06;
- 2226 - ETS 300 102-1.

2227 This value indicates the protocol defined by ETS 300 102-1 (EDSS1).

2228 7.6.9.8 Network signal information

2229 This parameter is transported as external signal information. The protocol ID shall be set to "ETS 300 102-1".

2230 The network signal information may include the following information elements as defined in GSM 09.07:

- 2231 - ISDN BC; the tag and length are defined by ETS 300 102-1.

2232 For the content, see GSM 09.07.

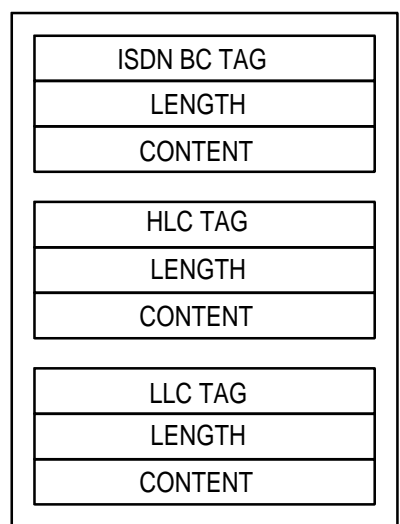
2233 - HLC; the tag and length are defined by ETS 300 102-1.

2234 For the content, see GSM 09.07.

2235 - LLC; the tag and length are defined by ETS 300 102-1.

2236 For the content, see GSM 09.07.

2237 They are contained in the Signal Information parameter according to figure 7.6/1 (irrespective of the order):



2238

2239

Figure 7.6/1: Network signal information parameter

2240 7.6.9.9 Call Info

2241 This parameter is transported as external signal information. The protocol ID shall be set to "GSM 04.08".

2242 The Call Info includes the set of information elements from the original SETUP message and is imported from
2243 GSM 04.08.

2244 7.6.9.10 Additional signal info

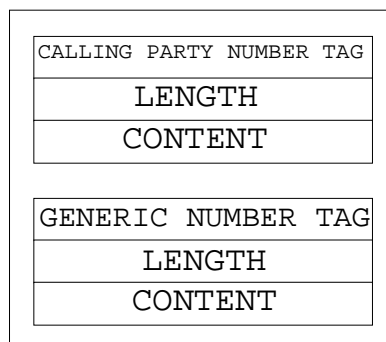
2245 This parameter is transported as ext-external signal information. The protocol ID shall be set to "ETS 300 356".

2246 The additional signal information may include the following information elements:

2247 - Calling Party Number as defined by ETS 300 356.

2248 - Generic Number as defined by ETS 300 356.

2249 They are contained in the Signal Information parameter according to figure 7.6/2 (irrespective of the order):



2250

2251

Figure 7.6/2: Additional signal information parameter2252 **7.6.10 System operations parameters**2253 **7.6.10.1 Network resources**

2254 This parameter refers to a class or type of network resource:

- 2255 - PLMN;
- 2256 - HLR;
- 2257 - VLR (current or previous);
- 2258 - MSC (controlling or current);
- 2259 - EIR;
- 2260 - radio sub-system.

2261 **7.6.10.2 Trace reference**

2262 This parameter represents a reference associated with a tracing request. The parameter is managed by OMC.

2263 **7.6.10.3 Trace type**

2264 This parameter identifies the type of trace. Trace types are fully defined in GSM 12.08.

2265 **7.6.11 Location Service Parameters**2266 **7.6.11.1 Age of Location Estimate**

2267 This parameter indicates how long ago the location estimate was obtained.

2268 **7.6.11.2 Void**2269 **7.6.11.3 Void**2270 **7.6.11.4 LCS Client ID**

2271 This parameter provides information related to the identity of an LCS client.

2272 **7.6.11.5 LCS Event**

2273 This parameter identifies an event associated with the triggering of a location estimate.

2274 7.6.11.6 LCS MLC Data

2275 This parameter provides the identities of any authorized GMLCs for a target MS. Only these GMLCs are allowed to
2276 send a location request for an external client when location requests are restricted to these GMLCs.

2277 7.6.11.7 LCS Priority

2278 This parameter gives the priority of the location request.

2279 7.6.11.8 LCS QoS

2280 This parameter defines the Quality of Service (QoS) for any location request. It is composed of the following elements.

2281 1) Response Time

2282 Indicates the category of response time – “low delay” or “delay tolerant”

2283 2) Horizontal Accuracy

2284 Indicates the required horizontal accuracy of the location estimate.

2285 3) Vertical Coordinate

2286 Indicates if a vertical coordinate is required (in addition to horizontal coordinates)

2287 4) Vertical Accuracy

2288 Indicates the required vertical accuracy of the location estimate (inclusion is optional).

2289 7.6.11.9 Void**2290 7.6.11.10 Void****2291 7.6.11.11 Location Estimate**

2292 This parameter gives an estimate of the location of an MS in universal coordinates and the accuracy of the estimate.

2293 7.6.11.12 Location Type

2294 This parameter indicates the type of location estimate required by the LCS client. Possible location estimate types
2295 include:

2296 • current location

2297 • current or last known location

2298 • initial location for an emergency services call

2299 7.6.11.13 NA-ESRD

2300 This parameter only applies to location for an emergency services call in North America and gives the North American
2301 Emergency Services Routing Digits.

2302 7.6.11.14 NA-ESRK

2303 This parameter only applies to location for an emergency services call in North America and gives the North American
2304 Emergency Services Routing Key.

2305 7.6.11.15 Void

2306 7.6.11.16 Privacy Override

2307 This parameter indicates if MS privacy is overridden by the LCS client when the GMLC and VMSC for an MR-LR are
2308 in the same country.

2309 7.6.11.17 Void

2310 7.6.11.18 Void

2311 7.6.11.19 Void

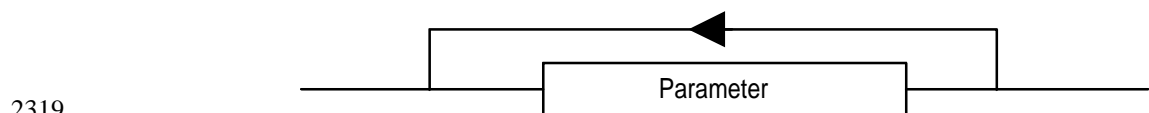
2312 7.7 Representation of a list of a basic parameter in service- 2313 primitives

2314 In some service-primitives several instances of a basic parameter of subclause 7.6 are required. In the service
2315 descriptions such cases will be represented as

2316

ParameterNameLIST

2317 in the tables where ParameterName refers to one of the parameters defined in subclause 7.6. This corresponds to the
2318 following construction rule:



2320 **Figure 7.7/1: Construction of Lists**

2321 **8** **Mobility services**2322 **8.1** **Location management services**2323 **8.1.1** **Void**2324 **8.1.1.1** **Void**2325 **8.1.1.2** **Void**2326 **8.1.1.3** **Void** **8.1.2** **MAP_UPDATE_LOCATION service**2327 **8.1.2.1** **Definition**

2328 This service is used by the VLR to update the location information stored in the HLR.

2329 The MAP_UPDATE_LOCATION service is a confirmed service using the service primitives given in table 6.1/2.

2330 **8.1.2.2** **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(=)		
Supported CAMEL Phases	C	C(=)		
SoLSA Support Indicator	C	C(=)		
IST Support Indicator	C	C(=)		
Super-Charger Supported in Serving	C	C(=)		
Network Entity				
HLR number			C	C(=)
User error			C	C(=)
Provider error				O

2331

2332

Table 8.1/2: MAP_UPDATE_LOCATION2333 **8.1.2.3** **Parameter definitions and use**2334 Invoke Id

2335 See definition in subclause 5.6.1.

2336 IMSI

2337 See definition in subclause 5.6.2.

2338 MSC Address

2339 See definition in subclause 5.6.2. The MSC address is used for short message delivery only and for each incoming call

2340 set-up attempt the MSRN will be requested from the VLR.

2341 VLR number

2342 See definition in subclause 5.6.2.

2343 LMSI

2344 See definition in subclause 5.6.2. It is an operator option to provide the LMSI from the VLR; it is mandatory for the
2345 HLR to support the LMSI handling procedures.

2346 Supported CAMEL Phases

2347 This parameter indicates which phases of CAMEL are supported. Must be present if a CAMEL phase different from
2348 phase 1 is supported. Otherwise may be absent.

2349 HLR number

2350 See definition in subclause 5.6.2. The presence of this parameter is mandatory in case of successful HLR updating.

2351 SoLSA Support Indicator

2352 This parameter is used by the VLR to indicate to the HLR in the Update Location indication that SoLSA is supported. If
2353 this parameter is not included in the Update Location indication and the Subscriber is marked as only allowed to roam in
2354 Subscribed LSAs, then the HLR shall reject the roaming and indicate to the VLR that roaming is not allowed to that
2355 Subscriber in the VLR.

2356 This SoLSA Support Indicator shall be stored by the HLR per VLR where there are Subscribers roaming. If a Subscriber
2357 is marked as only allowed to roam in Subscribed LSAs while roaming in a VLR and no SoLSA Support indicator is
2358 stored for that VLR, the location status of that Subscriber shall be set to Restricted.

2359 IST Support Indicator

2360 This parameter is used to indicate to the HLR that the VMSC supports basic IST functionality, that is, the VMSC is able
2361 to terminate the Subscriber Call Activity that originated the IST Alert when it receives the IST alert response indicating
2362 that the call(s) shall be terminated. If this parameter is not included in the Update Location indication and the Subscriber
2363 is marked as an IST Subscriber, then the HLR may limit the service for the subscriber (by inducing an Operator
2364 Determined barring of Roaming, Incoming or Outgoing calls), or allow service assuming the associated risk of not
2365 having the basic IST mechanism available.

2366 This parameter can also indicate that the VMSC supports the IST Command service, including the ability to terminate
2367 all calls being carried for the identified subscriber by using the IMSI as a key. If this additional capability is not included
2368 in the Update Location indication and the HLR supports the IST Command capability, then the HLR may limit the
2369 service for the subscriber (by inducing an Operator Determined barring of Roaming, Incoming or Outgoing calls), or
2370 allow service assuming the associated risk of not having the IST Command mechanism available.

2371 Super-Charger Supported in Serving Network Entity

2372 This parameter is used by the VLR to indicate to the HLR that the VLR supports the Super-Charger functionality and
2373 whether subscription data has been retained by the VLR. If subscription data has been retained by the VLR the age
2374 indicator shall be included. Otherwise the VLR shall indicate that subscriber data is required.

2375 If this parameter is absent then the VLR does not support the Super-Charger functionality.

2376 User error

2377 In case of unsuccessful updating, an error cause shall be returned by the HLR. The following error causes defined in
2378 subclause 5.6.1 may be used, depending on the nature of the fault:

2379 - unknown subscriber;

2380 - roaming not allowed;

2381 This cause will be sent if the MS is not allowed to roam into the PLMN indicated by the VLR number. The cause
2382 is qualified by the roaming restriction reason "PLMN Not Allowed" or "Operator Determined Barring". If no
2383 qualification is received (HLR with MAP Version 1), "PLMN Not Allowed" is taken as default.

2384 - system failure;

2385 - unexpected data value.

2386 Provider error

2387 For definition of provider errors see subclause 5.6.1.

2388 8.1.3 MAP_CANCEL_LOCATION service

2389 8.1.3.1 Definition

2390 This service is used between HLR and VLR to delete a subscriber record from the VLR. It may be invoked
2391 automatically when an MS moves from one VLR area to another, to remove the subscriber record from the old VLR, or
2392 by the HLR operator to enforce a location updating from the VLR to the HLR, e.g. on withdrawal of a subscription.

2393 Also this service is used between HLR and SGSN to delete a subscriber record from the SGSN. It may be invoked
2394 automatically when an MS moves from one SGSN area to another, to remove the subscriber record from the old SGSN,
2395 or by the HLR operator to enforce a location updating from the SGSN to the HLR.

2396 The MAP_CANCEL_LOCATION service is a confirmed service using the primitives defined in table 8.1/3.

2397 8.1.3.2 Service primitives

2398 **Table 8.1/3: MAP_CANCEL_LOCATION**

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

2399

2400 8.1.3.3 Parameter definitions and use

2401 Invoke Id

2402 See definition in subclause 7.6.1.

2403 IMSI

2404 See definition in subclause 7.6.2.

2405 LMSI

2406 See definition in subclause 7.6.2. The LMSI shall be included if it has been received from VLR. LMSI is not applicable
2407 between SGSN and HLR.

2408 Value 0000 0000 can be used to indicate that the LMSI is not in use.

2409 Cancellation Type

2410 See definition in subclause 5.6.3. The presence of this parameter is mandatory when the Cancel Location is sent to the
2411 SGSN. If the VLR receives this parameter and do not understand it the VLR shall ignore it.

2412 User error

2413 If the cancellation fails, an error cause is to be returned by the VLR or by the SGSN. The one of the following error
2414 causes defined in subclause 5.6.1 shall be used:

2415 - unexpected data value;

2416 - data missing.

2417 Provider error

2418 For definition of provider errors see subclause 7.6.1.

2419 8.1.4 MAP_SEND_IDENTIFICATION service

2420 8.1.4.1 Definition

2421 The MAP_SEND_IDENTIFICATION service is used between a VLR and a previous VLR to retrieve IMSI and
2422 authentication sets for a subscriber registering afresh in that VLR.

2423 The MAP_SEND_IDENTIFICATION service is a confirmed service using the service primitives defined in table 8.1/4.

2424 8.1.4.2 Service primitives

2425 **Table 8.1/4: MAP_SEND_IDENTIFICATION**

Parameter name	Request	Indication	Response	Confirm
Invoke Id	M	M(=)	M(=)	M(=)
TMSI	M	M(=)		
Number of requested vectors	M	M(=)		
Segmentation prohibited indicator	C	C(=)		
IMSI			C	C(=)
Authentication set			U	C(=)
User error			C	C(=)
Provider error				O

2426

2427 8.1.4.3 Parameter definitions and use

2428 Invoke Id

2429 See definition in subclause 7.6.1.

2430 TMSI

2431 See definition in subclause 7.6.2.

2432 Number of requested vectors

2433 A number indicating how many authentication vectors the new VLR is prepared to receive.

2434 Segmentation prohibited indicator

2435 This parameter indicates if the new VLR or SGSN allows message segmentation.

2436 IMSI

2437 See definition in subclause 7.6.2. The IMSI is to be returned if the service succeeds.

2438 Authentication set

2439 See definition in subclause 7.6.7. If the service succeeds a list of up to five authentication sets is returned, if there are
2440 any available.

2441 User error

2442 This parameter is mandatory if the service fails. The following error cause defined in subclause 7.6.1 may be used,
2443 depending on the nature of the fault:

2444 - unidentified subscriber.

2445 Provider error

2446 For definition of provider errors see subclause 7.6.1.

2447 8.1.5 Void

2448 8.1.5.1 Void

2449 8.1.5.2 Void

2450 8.1.5.3 Void

2451 8.1.6 MAP_PURGE_MS service

2452 8.1.6.1 Definition

2453 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
2454 routing information for a mobile terminated call or a mobile terminated short message will be treated as if the MS is not
2455 reachable. It is invoked when the subscriber record for the MS is to be deleted in the VLR, either by MMI interaction or
2456 automatically, e.g. because the MS has been inactive for several days. This service shall not be used if both the VLR and
2457 HLR support the Super-Charger functionality.

2458 Also this service is used between the SGSN and the HLR to cause the HLR to mark its data for an MS so that any
2459 request for routing information for a mobile terminated short message or a network requested PDP-context activation
2460 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
2461 SGSN, either by MMI interaction or automatically, e.g. because the MS has been inactive for several days. This service
2462 shall not be used if both the SGSN and HLR support the Super-Charger functionality.

2463 The MAP_PURGE_MS service is a confirmed service using the primitives defined in table 8.1/6.

2464 8.1.6.2 Service primitives

2465

Table 8.1/6: MAP_PURGE_MS

Parameter name	Request	Indication	Response	Confirm
Invoke Id	M	M(=)	M(=)	M(=)
IMSI	M	M(=)		
VLR number	C	C(=)		
Freeze TMSI			C	C(=)
Freeze P-TMSI			C	C(=)
SGSN number	C	C(=)		
User error			C	C(=)
Provider error				O

2466

2467 8.1.6.3 Parameter definitions and use

2468 Invoke ID

2469 See definition in subclause 7.6.1.

2470 IMSI

2471 See definition in subclause 7.6.2.

2472 VLR number

2473 Shall be present if the sender is VLR. See definition in subclause 7.6.2.

2474 SGSN number

2475 Shall be present if the sender is SGSN. See definition in subclause 7.6.2

2476 Freeze TMSI

2477 This parameter is sent to the VLR to indicate that the TMSI has to be frozen. It shall be present if the received VLR
2478 number matches the stored VLR number.

2479 Freeze P-TMSI

2480 This parameter is sent to the SGSN to indicate that the P-TMSI has to be frozen. It shall be present if the received SGSN
2481 number matches the stored SGSN number.

2482 User error

2483 This parameter is sent by the responder when an error is detected and if present, takes one of the following values:

2484 - Data Missing;

2485 - Unexpected Data Value;

2486 - UnknownSubscriber.

2487 Provider error

2488 See definition of provider errors in subclause 7.6.1.

2489 **8.1.7 MAP_UPDATE_GPRS_LOCATION service**2490 **8.1.7.1 Definition**

2491 This service is used by the SGSN to update the location information stored in the HLR.

2492 The MAP_UPDATE_GPRS_LOCATION service is a confirmed service using the service primitives given in
2493 table 8.1/7.

2494 **8.1.7.2 Service primitives**2495 **Table 8.1/7: MAP_UPDATE_GPRS_LOCATION**

Parameter name	Request	Indication	Response	Confirm
Invoke Id	M	M(=)	M(=)	M(=)
IMSI	M	M(=)		
SGSN number	M	M(=)		
SGSN address	M	M(=)		
Supported CAMEL Phases	C	C(=)		
SoLSA Support Indicator	C	C(=)		
Super-Charger Supported in Serving Network Entity	C	C(=)		
GPRS enhancements support indicator	C	C(=)		
HLR number			C	C(=)
User error			C	C(=)
Provider error				O

2496

2497 **8.1.7.3 Parameter definitions and use**2498 Invoke Id

2499 See definition in subclause 7.6.1.

2500 IMSI

2501 See definition in subclause 7.6.2.

2502 SGSN number

2503 See definition in subclause 7.6.2.

2504 SGSN address

2505 See definition in subclause 7.6.2.

2506 Supported CAMEL Phases

2507 This parameter indicates which phases of CAMEL are supported. The SGSN can only support CAMEL phase 3 or
2508 greater.

2509 SoLSA Support Indicator

2510 This parameter is used by the SGSN to indicate to the HLR in the Update GPRS Location indication that SoLSA is
2511 supported. If this parameter is not included in the Update GPRS Location indication and the Subscriber is marked as
2512 only allowed to roam in Subscribed LSAs, then the HLR shall reject the roaming and indicate to the SGSN that roaming
2513 is not allowed to that Subscriber in the SGSN.

2514 This SoLSA Support Indicator shall be stored by the HLR per SGSN where there are Subscribers roaming. If a
2515 Subscriber is marked as only allowed to roam in Subscribed LSAs while roaming in a SGSN and no SoLSA Support
2516 indicator is stored for that SGSN, the location status of that Subscriber has to be set to Restricted.

2517 Super-Charger Supported in Serving Network Entity

2518 This parameter is used by the SGSN to indicate to the HLR that the SGSN supports the Super-Charger functionality and
2519 whether subscription data has been retained by the SGSN. If subscription data has been retained by the SGSN the age
2520 indicator shall be included. Otherwise the SGSN shall indicate that subscriber data is required.

2521 If this parameter is absent then the SGSN does not support the Super-Charger functionality.

2522 GPRS enhancements support indicator

2523 This parameter is used by the SGSN to indicate to the HLR in the Update GPRS Location indication that GPRS
2524 enhancements are supported. If this parameter is included in the Update GPRS Location indication the HLR may send
2525 the extensible QoS in the PDP contexts to the SGSN.

2526 HLR number

2527 See definition in subclause 7.6.2. The presence of this parameter is mandatory in case of successful HLR updating.

2528 User error

2529 In case of unsuccessful updating, an error cause shall be returned by the HLR. The following error causes defined in
2530 subclause 7.6.1 may be used, depending on the nature of the fault:

2531 - unknown subscriber;

2532 - roaming not allowed;

2533 This cause will be sent if the MS is not allowed to roam into the PLMN indicated by the SGSN number. The
2534 cause is qualified by the roaming restriction reason "PLMN Not Allowed" or "Operator Determined Barring".

2535 - system failure;

2536 - unexpected data value.

2537 The diagnostic in the Unknown Subscriber may indicate "Imsi Unknown" or "Gprs Subscription Unknown".

2538 Provider error

2539 For definition of provider errors see subclause 7.6.1.

2540 **8.1.8 MAP-NOTE-MM-EVENT**2541 **8.1.8.1 Definition**

2542 This service is used between the VLR and the gsmSCF when for a subscriber a mobility management event has been
 2543 processed successfully, that subscriber is provisioned with M-CSI and the relevant mobility management event is
 2544 marked for reporting.
 2545

2546 **8.1.8.2 Service primitives**

2547 The service primitives are shown in table 8.1/8.

2548 **Table 8.1/8: MAP_NOTE_MM_EVENT parameters**

Parameter name	Request	Indication	Response	Confirm
Invoke id	M	M(=)	M(=)	M(=)
Event Met	M	M(=)		
Service Key	M	M(=)		
IMSI	M	M(=)		
Basic MSISDN	M	M(=)		
Location Information	C	C(=)		
LSA Identity	C	C(=)		
Supported CAMEL Phases	M	M(=)		
User error			C	C(=)
Provider error				O

2549

2550 **8.1.8.3 Parameter use**2551 Event Met

2552 This parameter indicates the mobility management event that has lead to the notification. It shall have one of the
 2553 following values:

2554

2555 - Location update in the same VLR service area

2556

2557 - Location update to ananother VLR service area

2558

2559 - IMSI attach

2560

2561 - MS initiated IMSI detach (explicit detach)

2562

2563 - Network initiated IMSI detach (implicit detach)

2564

2565 Service Key

2566 See subclause 7.6.x.

2567

2568 IMSI

2569 See subclause 7.6.x.

2570

2571 Basic MSISDN

2572 See subclause 7.6.x.

2573

2574 Location Information

2575 See subclause 7.6.x. This information shall be sent, if available.

2576

2577 LSA Identity

2578 See subclause 7.6.x. This information shall be sent, if available.

2579

2580 Supported CAMEL Phases

2581 See subclause 7.6.x. This information shall always be sent.

2582

2583 User error

2584 This parameter is sent by the receiving entity when an error is detected. It shall have one of the following values:

2585 - Data Missing;

2586 - Unexpected Data Value;

2587 - Unknown Subscriber;

2588 - MM-EventNotSupported;

2589

2590 Provider error

2591 This is defined in subclause 7.6.1.

2592 **8.2 Paging and search**2593 **8.2.1 MAP_PAGE service**2594 **8.2.1.1 Definition**

2595 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.

2597 The MAP_PAGE service is a confirmed service using the primitives from table 8.2/1.

2598 **8.2.1.2 Service primitives**

2599

Table 8.2/1: MAP_PAGE

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

2600

2601 8.2.1.3 Parameter definitions and use

2602 Invoke Id

2603 See definition in subclause 7.6.1.

2604 IMSI

2605 See definition in subclause 7.6.2. The IMSI is used to define the paging subgroup. If the TMSI is not supplied, paging
2606 on the radio path uses the IMSI as an identifier.

2607 Stored location area Id

2608 See definition in subclause 7.6.2.

2609 TMSI

2610 See definition in subclause 7.6.2. The TMSI is included if paging on the radio channel is to use the TMSI as an
2611 identifier.

2612 User error

2613 The following error causes defined in subclause 7.6.1 may be sent by the user in case of a paging error, depending on
2614 the failure reason:

2615 - absent subscriber;

2616 - unknown location area;

2617 - busy subscriber;

2618 - system failure;

2619 This corresponds to the case where there is no call associated with the MAP_PAGE service, i.e. if the call has
2620 been released but the dialogue to the VLR has not been aborted.

2621 - unexpected data value.

2622 Provider error

2623 See definition in subclause 7.6.1.

2624 8.2.2 MAP_SEARCH_FOR_MS service

2625 8.2.2.1 Definition

2626 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
2627 VLR does not hold location area information confirmed by radio contact.

2628 The MAP_SEARCH_FOR_MS service is a confirmed service using the primitives from table 8.2/2.

2629 8.2.2.2 Service primitives

2630

Table 8.2/2: MAP_SEARCH_FOR_MS

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

2631

2632 8.2.2.3 Parameter definitions and use

2633 Invoke Id

2634 See definition in subclause 7.6.1.

2635 IMSI

2636 See definition in subclause 7.6.2. The IMSI is used to identify the subscriber when paging on the radio path.

2637 Current location area Id

2638 See definition in subclause 7.6.2. In case of successful outcome of the service, i.e. if the MS responds to paging, the
2639 Location Area Id of the area in which the MS responded is given in the response.

2640 User error

2641 The following error causes defined in subclause 7.6.1 shall be sent by the user if the search procedure fails, depending
2642 on the failure reason:

2643 - absent subscriber;

2644 This error cause is returned by the MSC if the MS does not respond to the paging request.

2645 - system failure;

2646 This corresponds to the case where there is no call associated with the MAP_SEARCH_FOR_MS service, i.e. if
2647 the call has been released but the dialogue to the VLR has not been aborted.

2648 - busy subscriber;

2649 - unexpected data value.

2650 Provider error

2651 See definition in subclause 7.6.1.

2652 8.3 Access management services

2653 8.3.1 MAP_PROCESS_ACCESS_REQUEST service

2654 8.3.1.1 Definition

2655 This service is used between MSC and VLR to initiate processing of an MS access to the network, e.g. in case of mobile
2656 originated call set-up or after being paged by the network.

2657 The MAP_PROCESS_ACCESS_REQUEST service is a confirmed service using the primitives from table 8.3/1.

2658 8.3.1.2 Service primitives

2659 **Table 8.3/1: MAP_PROCESS_ACCESS_REQUEST**

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(=)		
Serving cell 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

2660

2661 8.3.1.3 Parameter definitions and use

2662 Invoke Id

2663 See definition in subclause 7.6.1.

2664 CM service type

2665 See definition in subclause 7.6.9.

2666 Access connection status

2667 See definition in subclause 7.6.9.

2668 Current Location Area Id

2669 See definition in subclause 7.6.2. This parameter is used to update the VLR in case of previous VLR failure.

2670 Serving cell id

2671 See definition in subclause 7.6.2.

2672 TMSI

2673 See definition in subclause 7.6.2. Either TMSI or IMSI as received from the MS are included in the Request/Indication,
 2674 but one shall be present. In case of CM Service Type "Emergency Call Establishment", the IMEI may replace
 2675 IMSI/TMSI.

2676 Cksn

2677 See definition in subclause 7.6.7. In case of access with TMSI, the Cksn shall be present.

2678 IMSI

2679 See definition in subclause 7.6.2. Either TMSI or IMSI as received from the MS are included in the Request/Indication,
 2680 but one shall be present. In case of CM Service Type "Emergency Call Establishment", the IMEI may replace
 2681 IMSI/TMSI.

2682 In the Response/Confirmation, the IMSI is to be sent in case of successful outcome of the service. In case of CM Service
 2683 Type "Emergency Call Establishment", IMEI may replace IMSI.

2684 IMEI

2685 See definition in subclause 7.6.2. The IMEI may replace IMSI/TMSI in the Request/Indication and IMSI in the
 2686 Response/Confirmation only in case the CM Service Type indicates "Emergency Call Establishment".

2687 MSISDN

2688 See definition in subclause 7.6.2. 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.

2690 User error

2691 One of the following error causes defined in subclause 7.6.1 shall be sent by the user if the access request fails, depending on the failure reason:

2693 - unidentified subscriber;

2694 - illegal subscriber;

2695 This error is sent if a correlated authentication procedure has not authenticated the subscriber.

2696 - illegal equipment;

2697 This error is sent if an IMEI check failed, i.e. the IMEI is blacklisted or not white-listed.

2698 - roaming not allowed;

2699 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.

2700

2702 - unknown location area;

2703 - system failure;

2704 - unexpected data value.

2705 Provider error

2706 For definition of provider errors see subclause 7.6.1.

2707 **8.4 Handover services**2708 **8.4.1 MAP_PREPARE_HANOVER service**2709 **8.4.1.1 Definition**

2710 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.

2712 The MAP_PREPARE_HANOVER service is a confirmed service using the primitives from table 8.4/1.

2713 **8.4.1.2 Service primitives**2714 **Table 8.4/1: MAP_PREPARE_HANOVER**

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

2715

2716 8.4.1.3 Parameter use

2717 Invoke Id

2718 For definition of this parameter see subclause 7.6.1.

2719 Target Cell Id

2720 For definition of this parameter see subclause 7.6.2. This parameter is only included if the service is not in an ongoing transaction.

2722 HO-Number Not Required

2723 For definition of this parameter see subclause 7.6.6.

2724 BSS-APDU

2725 For definition of this parameter see subclause 7.6.9.

2726 Handover Number

2727 For definition of this parameter see subclause 7.6.2. This parameter shall be returned, unless the parameter HO-NumberNotRequired is sent.

2729 User error

2730 For definition of this parameter see subclause 7.6.1. The following errors defined in subclause 7.6.1 may be used, depending on the nature of the fault:

2732 - No handover number available;

2733 - System failure;

2734 - Unexpected data value;

2735 - DataMissing.

2736 Provider error

2737 See definition of provider errors in subclause 7.6.1.

2738 8.4.2 MAP_SEND_END_SIGNAL service

2739 8.4.2.1 Definition

2740 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.

2742 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.

2745 The MAP_SEND_END_SIGNAL service is a confirmed service using the primitives from table 8.4/2.

2746 8.4.2.2 Service primitives

2747 **Table 8.4/2: MAP_SEND_END_SIGNAL**

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

2748

2749 8.4.2.3 Parameter use

2750 Invoke Id

2751 For definition of this parameter see subclause 7.6.1.

2752 BSS-APDU

2753 For definition of this parameter see subclause 7.6.9.

2754 Provider error

2755 For definition of this parameter see subclause 7.6.1.

2756 8.4.3 MAP_PROCESS_ACCESS_SIGNALLING service

2757 8.4.3.1 Definition

2758 This service is used between MSC-B and MSC-A (E-interface) to pass information received on the A-interface in
 2759 MSC-B to MSC-A.

2760 The MAP_PROCESS_ACCESS_SIGNALLING service is a non-confirmed service using the primitives from
 2761 table 8.4/3.

2762 8.4.3.2 Service primitives

2763 **Table 8.4/3: MAP_PROCESS_ACCESS_SIGNALLING**

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

2764

2765 8.4.3.3 Parameter use

2766 Invoke Id

2767 For definition of this parameter see subclause 7.6.1.

2768 BSS-APDU

2769 For definition of this parameter see subclause 7.6.9.

2770 8.4.4 MAP_FORWARD_ACCESS_SIGNALLING service

2771 8.4.4.1 Definition

2772 This service is used between MSC-A and MSC-B (E-interface) to pass information to be forwarded to the A-interface of
 2773 MSC-B.

2774 The MAP_FORWARD_ACCESS_SIGNALLING service is a non-confirmed service using the primitives from
 2775 table 8.4/4.

2776 8.4.4.2 Service primitives

2777 **Table 8.4/4: MAP_FORWARD_ACCESS_SIGNALLING**

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

2778

2779 8.4.4.3 Parameter use

2780 For the definition and use of all parameters and errors, see subclause 7.6.1

2781 Invoke Id

2782 For definition of this parameter see subclause 7.6.1.

2783 BSS-APDU

2784 For definition of this parameter see subclause 7.6.9.

2785 8.4.5 MAP_PREPARE_SUBSEQUENT_HANOVER service

2786 8.4.5.1 Definition

2787 This service is used between MSC-B and MSC-A (E-interface) to inform MSC-A that it has been decided that a
2788 handover to either MSC-A or a third MSC (MSC-B') is required.2789 The MAP_PREPARE_SUBSEQUENT_HANOVER service is a confirmed service using the primitives from
2790 table 8.4/5.

2791 8.4.5.2 Service primitives

2792 **Table 8.4/5: MAP_PREPARE_SUBSEQUENT_HANOVER**

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

2793

2794 8.4.5.3 Parameter use

2795 Invoke Id

2796 For definition of this parameter see subclause 7.6.1.

2797 Target Cell Id

2798 For definition of this parameter see subclause 7.6.2.

2799 Target MSC Number

2800 For definition of this parameter see subclause 7.6.2.

2801 BSS-APDU

2802 For definition of this parameter see subclause 7.6.9.

2803 User error

2804 For definition of this parameter see subclause 7.6.1. The following error causes defined in subclause 7.6.1 may be used,
2805 depending on the nature of the fault:

- 2806 - Unknown MSC;
- 2807 - Subsequent handover failure;
- 2808 - Unexpected data value;
- 2809 - Data Missing.

2810 Provider error

2811 For definition of this parameter see subclause 7.6.1.

2812 **8.4.6 MAP_ALLOCATE_HANOVER_NUMBER service**2813 **8.4.6.1 Definition**

2814 This service is used between MSC and VLR (B-interface) to request a handover number.

2815 The MAP_ALLOCATE_HANOVER_NUMBER service is a confirmed service using the primitives from table 8.4/6.

2816 **8.4.6.2 Service primitives**2817 **Table 8.4/6: MAP_ALLOCATE_HANOVER_NUMBER**

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

2818

2819 **8.4.6.3 Parameter use**2820 Invoke Id

2821 For definition of this parameter see subclause 7.6.1.

2822 User error

2823 For definition of this parameter see subclause 7.6.1. The following errors defined in subclause 7.6.1 may be used,
2824 depending on the nature of the fault:

- 2825 - No handover number available.

2826 Provider error

2827 For definition of this parameter see subclause 7.6.1.

2828 **8.4.7 MAP_SEND_HANOVER_REPORT service**2829 **8.4.7.1 Definition**

2830 This service is used between VLR and MSC-B (B-interface) to transfer the handover number to be forwarded to and
2831 used by MSC-A.

2832 The MAP_SEND_HANOVER_REPORT service is a confirmed service using the primitives from table 8.4/7.

2833 8.4.7.2 Service primitives

2834 **Table 8.4/7: MAP_SEND_HANOVER_REPORT**

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

2835

2836 8.4.7.3 Parameter use

2837 Invoke Id

2838 For definition of this parameter see subclause 7.6.1.

2839 Handover Number

2840 For definition of this parameter see subclause 7.6.2.

2841 Linked Id

2842 For definition of this parameter see subclause 7.6.1. This service is linked with

2843 MAP_ALLOCATE_HANOVER_NUMBER.

2844 Provider error

2845 For definition of this parameter see subclause 7.6.1.

2846 8.5 Authentication management services

2847 8.5.1 MAP_AUTHENTICATE service

2848 8.5.1.1 Definition

2849 This service is used between the VLR and the MSC when the VLR receives a MAP service indication from the MSC
 2850 concerning a location registration, call set-up, operation on a supplementary service or a request from the MSC to
 2851 initiate authentication.

2852 The service is a confirmed service and consists of four service primitives.

2853 8.5.1.2 Service primitives

2854 The service primitives are shown in table 8.5/1

2855 **Table 8.5/1: MAP_AUTHENTICATE parameters**

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

2856

2857 8.5.1.3 Parameter use

2858 Invoke id

2859 See subclause 7.6.1 for the use of this parameter.

2860 RAND

2861 See subclause 7.6.7 for the use of this parameter.

2862 CKSN

2863 See subclause 7.6.7 for the use of this parameter.

2864 SRES

2865 See subclause 7.6.7 for the use of this parameter.

2866 Provider error

2867 See subclause 7.6.1 for the use of this parameter.

2868 8.5.2 MAP_SEND_AUTHENTICATION_INFO service

2869 8.5.2.1 Definition

2870 This service is used between the VLR and the HLR for the VLR to retrieve authentication information from the HLR.

2871 The VLR requests up to five authentication vectors.

2872 Also this service is used between the SGSN and the HLR for the SGSN to retrieve authentication information from the

2873 HLR. The SGSN requests up to five authentication vectors.

2874 If the HLR cannot provide the VLR or the SGSN with triplets, an empty response is returned. The VLR or the SGSN
2875 may then re-use old authentication triplets, except where this is forbidden under the conditions specified in GSM 03.20
2876 [24].

2877 If the HLR cannot provide the VLR or the SGSN with quintuplets, an empty response is returned. The VLR or the
2878 SGSN shall not re-use old authentication quintuplets.

2879 If the VLR or SGSN receives a MAP-Send_AUTHENTICATION_INFO response containing a User Error parameter as
2880 part of the handling of an authentication procedure, the authentication procedure in the VLR or SGSN shall fail.

2881 Security related network functions are further described in GSM 03.20 and 3G TS 33.102.

2882 The service is a confirmed service and consists of four service primitives.

2883 8.5.2.2 Service primitives

2884 The service primitives are shown in table 8.5/2.

2885

Table 8.5/2: MAP_SEND_AUTHENTICATION_PARAMETERS parameters

Parameter name	Request	Indication	Response	Confirm
Invoke id	M	M(=)	M(=)	M(=)
IMSI	M	M(=)		
Number of requested vectors	M	M(=)		
Re-synchronisation Info	C	C(=)		
Segmentation prohibited indicator	C	C(=)		
Immediate response preferred indicator	C	C(=)		
AuthenticationSetList			C	C(=)
User error			C	C(=)
Provider error				O

2886

2887 **8.5.2.3 Parameter use**2888 Invoke id

2889 See subclause 7.6.1 for the use of this parameter.

2890 IMSI

2891 See subclause 7.6.2 for the use of this parameter.

2892 Number of requested vectors

2893 A number indicating how many authentication vectors the VLR or SGSN is prepared to receive.

2894 Re-synchronisation Info

2895 For definition and use of this parameter see 3G TS 33.102.

2896 Segmentation prohibited indicator

2897 This parameter indicates if the VLR or SGSN allows message segmentation.

2898 Immediate response preferred indicator

2899 This parameter indicates that the VLR or SGSN requests that the HLR immediately sends back the available
 2900 authentication vectors. It shall be ignored if the number of available vectors is less than the number of requested vectors
 2901 and if the VLR or SGSN or the HLR does not support message segmentation.

2902 AuthenticationSetList

2903 A set of one to five authentication vectors are transferred from the HLR to the VLR or from the HLR to the SGSN, if the
 2904 outcome of the service was successful.

2905 User error

2906 One of the following error causes defined in subclause 7.6.1 shall be sent by the user in case of unsuccessful outcome of
 2907 the service, depending on the respective failure reason:

2908 - unknown subscriber;

2909 - unexpected data value;

2910 - system failure;

2911 - data missing.

2912 Provider error

2913 See subclause 7.6.1 for the use of this parameter.

2914 8.6 Security management services

2915 8.6.1 MAP_SET_CIPHERING_MODE service

2916 8.6.1.1 Definitions

2917 This service is used between the VLR and the MSC to set the ciphering mode and to start ciphering if applicable. It is
2918 called when another service requires that information is to be sent on the radio path in encrypted form.

2919 The service is a non-confirmed service and consists of two service primitives.

2920 8.6.1.2 Service primitives

2921 The service primitives are shown in table 8.6/1

2922 **Table 8.6/1: MAP_SET_CIPHERING_MODE parameters**

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

2923

2924 8.6.1.3 Parameter use

2925 Invoke id

2926 See subclause 7.6.1 for the use of this parameter.

2927 Ciphering mode

2928 See subclause 7.6.7 for the use of this parameter.

2929 Kc

2930 The Kc parameter should be included when the ciphering mode parameter indicates that ciphering must be performed.

2931 8.7 International mobile equipment identities management 2932 services

2933 8.7.1 MAP_CHECK_IMEI service

2934 8.7.1.1 Definition

2935 This service is used between the VLR and the MSC and between the MSC and the EIR and between the SGSN and EIR
2936 to request check of IMEI. If the IMEI is not available in the MSC or in the SGSN, it is requested from the MS and
2937 transferred to the EIR in the service request.

2938 The service is a confirmed service and consists of four service primitives.

2939 8.7.1.2 Service primitives

2940 The service primitives are shown in table 8.7/1.

2941

Table 8.7/1: MAP_CHECK_IMEI parameters

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

2942

2943 **8.7.1.3 Parameter use**2944 Invoke id

2945 See subclause 7.6.1 for the use of this parameter.

2946 IMEI

2947 See subclause 7.6.2 for the use of this parameter. The parameter shall not be included in the service request between the
 2948 VLR and the MSC, but is mandatory in the service request from the MSC to the EIR and from the SGSN to the EIR. It
 2949 is not included in the service response from the EIR to the MSC or to the SGSN, but is mandatory in the service
 2950 response from the MSC to the VLR on successful outcome.

2951 Equipment status

2952 See subclause 7.6.4 for the use of this parameter. This parameter is sent by the responder in case of successful outcome
 2953 of the service.

2954 User error

2955 One of the following error causes defined in subclause 7.6.1 shall be sent by the user in case of unsuccessful outcome of
 2956 the service, depending on the respective failure reason:

2957 - unknown equipment;

2958 This error is returned by the responder when the IMEI is not known in the EIR.

2959 - system failure;

2960 - unexpected data value.

2961 Provider error

2962 See subclause 7.6.1 for the use of this parameter.

2963 **8.7.2 MAP_OBTAIN_IMEI service**2964 **8.7.2.1 Definition**

2965 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
 2966 requested from the MS.

2967 The service is a confirmed service and consists of four service primitives.

2968 **8.7.2.2 Service primitives**

2969 The service primitives are shown in table 8.7/2.

2970

Table 8.7/2: MAP_OBTAIN_IMEI parameters

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

2971

2972 **8.7.2.3** Parameter use2973 Invoke id

2974 See subclause 7.6.1 for the use of this parameter.

2975 IMEI2976 See subclause 7.6.2 for the use of this parameter. The parameter IS included in the service response from the MSC to the
2977 VLR on successful outcome of the service.2978 User error

2979 If the service fails, the VLR sends the user error System Failure (see subclause 7.6.1) to the MSC.

2980 Provider error

2981 See subclause 7.6.1 for the use of this parameter.

2982 **8.8** Subscriber management services2983 **8.8.1** MAP-INSERT-SUBSCRIBER-DATA service2984 **8.8.1.1** Definition

2985 This service is used by an HLR to update a VLR with certain subscriber data in the following occasions:

- 2986 - the operator has changed the subscription of one or more supplementary services, basic services or data of a
2987 subscriber. Note that in case of withdrawal of a Basic or Supplementary service this primitive shall not be used;
- 2988 - the operator has applied, changed or removed Operator Determined Barring;
- 2989 - the subscriber has changed data concerning one or more supplementary services by using a subscriber procedure;
- 2990 - the HLR provides the VLR with subscriber parameters at location updating of a subscriber or at restoration. In
2991 this case, this service is used to indicate explicitly that a supplementary service is not provisioned, if the
2992 supplementary service specification requires it. The only supplementary services which have this requirement are
2993 the CLIR and COLR services. Network access mode is provided only in restoration. If the Super-Charger
2994 functionality is supported the HLR may not need to provide the VLR with subscriber parameters at location
2995 updating of a subscriber. See TS 23.116.

2996 Also this service is used by an HLR to update a SGSN with certain subscriber data in the following occasions:

- 2997 - if the GPRS subscription has changed;
- 2998 - if the network access mode is changed;
- 2999 - the operator has applied, changed or removed Operator Determined Barring;
- 3000 - the HLR provides the SGSN with subscriber parameters at GPRS location updating of a subscriber. If the Super-
3001 Charger functionality is supported the HLR may not need to provide the SGSN with subscriber parameters. See
3002 TS 23.116.

3003 It is a confirmed service and consists of the primitives shown in table 6.8/1.

3004 8.8.1.2 Service primitives

3005 Table 8.8/1: MAP-INSERT-SUBSCRIBER-DATA

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(=)		
eMLPP Subscription Data	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(=)		
VLR CAMEL Subscription Info	C	C(=)		
Voice Broadcast Data	C	C(=)		
Voice Group Call Data	C	C(=)		
Network access mode	C	C(=)		
GPRS Subscription Data	C	C(=)		
Roaming Restricted In SGSN Due To Unsupported Feature	C	C(=)		
North American Equal Access preferred Carrier Id List	U	C(=)		
SGSN Camel Subscription Info	C	C(=)		
LSA Information	C	C(=)		
IST Alert Timer	C	C(=)		
SS-Code List			C	C(=)
LMU Identifier	C	C(=)		
LCS Information	C	C(=)		
Super-Charger Supported In HLR	C	C(=)		
Regional Subscription Response			C	C(=)
Supported CAMEL Phases			C	C(=)
User error			U	C(=)
Provider error				O

3006

3007 8.8.1.3 Parameter use

3008 Network access mode

3009 This parameter defines if the subscriber has access to MSC/VLR and/or to SGSN. This parameter is used by SGSN and
3010 MSC/VLR. In VLR, the parameter is used only as part of Restore Data Procedure and the parameter is not stored in the
3011 VLR.

3012 All parameters are described in subclause 7.6. The following clarifications are applicable:

3013 IMSI

3014 It is only included if the service is not used in an ongoing transaction (e.g. location updating). This parameter is used by
3015 the VLR and the SGSN.

3016 MSISDN

3017 It is included either at location updating or when it is changed. The MSISDN sent shall be the basic MSISDN. This
3018 parameter is used by the VLR and the SGSN.

3019 Category

3020 It is included either at location updating or when it is changed. This parameter is used only by the VLR and if the SGSN
3021 receives this parameter it shall ignore it.

3022 Subscriber Status

3023 It is included either at location updating or when it is changed.

3024 To apply, remove or update Operator Determined Barring Categories the Subscriber Status is set to Operator
3025 Determined Barring. In this case ODB General Data shall also be present. If the Operator Determined Barring applies
3026 and the subscriber is registered in the HPLMN and HPLMN specific Operator Determined Barring applies then ODB
3027 HPLMN Specific Data shall also be present.

3028 To remove all Operator Determined Barring Categories the Subscriber Status shall be set to "Service Granted". This
3029 parameter is used by the VLR and the SGSN.

3030 Bearer service List

3031 A list of Extensible Bearer service parameters (Extensible Bearer service is defined in subclause 7.6). An Extensible
3032 Bearer service parameter must be the code for an individual Bearer service, except in the cases described below.

3033 The codes for the Bearer service groups "allAlternateSpeech-DataCDA" and "allAlternateSpeech-DataCDS" shall, if
3034 applicable, be sent from the HLR to the VLR as a pair. The codes for the Bearer service groups
3035 "allSpeechFollowedByDataCDA" and "allSpeechFollowedByDataCDS" shall, if applicable, be sent from the HLR to
3036 the VLR as a pair.

3037 If it is included in the Request/Indication, it includes either all Extensible Bearer services subscribed (at location
3038 updating or at restoration) or only the ones added (at subscriber data modification).

3039 If the VLR receives an Indication containing any Extensible Bearer service parameters which it does not
3040 support/allocate it returns them in the response to the HLR and discards the unsupported Extensible Bearer services (no
3041 error is sent back), except in the cases described below.

3042 If the VLR receives the codes for the Bearer service groups "allSpeechFollowedByDataCDA" and
3043 "allSpeechFollowedByDataCDS" and supports one or more of the circuit-switched synchronous or asynchronous data
3044 rates specified for simple data bearer services, it shall accept the bearer service codes, and not return them in the
3045 response to the HLR. If the VLR does not support any of the circuit-switched synchronous or asynchronous data rates
3046 specified for simple data bearer services, and receives the pair of codes for "allAlternateSpeech-DataCDA" and
3047 "allAlternateSpeech-DataCDS" or the pair of codes for "allSpeechFollowedByDataCDA" and
3048 "allSpeechFollowedByDataCDS", it shall reject the pair of codes by returning them in the response to the HLR. This
3049 parameter is used only by the VLR and if the SGSN receives this parameter it shall ignore it.

3050 Teleservice List

3051 A list of Extensible Teleservice parameters (Extensible Teleservice is defined in subclause 7.6). An Extensible
3052 Teleservice parameter must be the code for an individual Teleservice.

3053 If it is included in the Request/Indication, it contains either all Extensible Teleservices subscribed (at location updating
3054 or at restoration) or the ones added (at subscriber data modification). Only the Extensible Teleservices that are relevant
3055 to the node at which the message is received should be included in the Teleservice List.

3056 If the VLR or the SGSN receives an Indication containing any Extensible Teleservice parameters which it does not
3057 support/allocate it returns them in the response to the HLR and discards the unsupported Extensible Teleservices (no
3058 error is sent back). This parameter is used by the VLR and the SGSN.

3059 Forwarding information List

3060 A list of Extensible Forwarding information parameters (Extensible Forwarding information is defined in subclause 7.6).
3061 It includes Call Forwarding services either at location updating or at restoration or when they are changed. Each
3062 Extensible Forwarding information parameter shall be treated independently of all other parameters in the primitive.

3063 The Extensible Forwarding information shall include the SS-Code for an individual call forwarding supplementary
3064 service. The Extensible Forwarding information shall contain one or more Extensible Forwarding Features (Extensible
3065 Forwarding Feature is defined in subclause 7.6).

- 3066 The Extensible Forwarding Feature may include an Extensible Basic Service Group. This shall be interpreted according
3067 to the rules in subclause 8.8.1.4.
- 3068 The Extensible Forwarding Feature shall contain an Extensible SS-Status parameter.
- 3069 If the Extensible SS-Status indicates that call forwarding is registered then (except for call forwarding unconditional) the
3070 Extensible Forwarding Feature shall contain a forwarded-to number and, if available, the forwarded-to subaddress. In
3071 other states the forwarded-to number and, if applicable, the forwarded-to subaddress shall not be included. For call
3072 forwarding unconditional the forwarded-to number and, if applicable, the forwarded-to subaddress shall not be included.
3073 If the VLR does not receive a forwarded-to subaddress then it shall assume that a forwarded-to subaddress has not been
3074 registered.
- 3075 The Extensible Forwarding Feature shall contain the extensible forwarding options (except for call forwarding
3076 unconditional where the extensible forwarding options shall not be included). Bits 3 and 4 of the extensible forwarding
3077 options shall be ignored by the VLR, and may be set to any value by the HLR.
- 3078 For call forwarding on no reply: If the extensible SS-Status indicates that call forwarding is registered then the
3079 Extensible Forwarding Feature shall contain an extensible no reply condition timer. In other states the no reply condition
3080 timer shall not be included.
- 3081 For call forwarding services other than call forwarding on no reply: The Extensible Forwarding Feature shall not contain
3082 a no reply condition timer.
- 3083 If the VLR receives an Indication containing any Call Forwarding service codes which it does not support/allocate it
3084 returns them to the HLR in the parameter SS-Code List and discards the unsupported Call Forwarding service codes (no
3085 error is sent back). This parameter is used only by the VLR and if the SGSN receives this parameter it shall ignore it.
- 3086 Call barring information List
- 3087 A list of Extensible Call barring information parameters (Extensible Call barring information is defined in
3088 subclause 7.6). It includes Call Barring services either at location updating or at restoration or when they are changed.
3089 Each Extensible Call barring information parameter shall be treated independently of all other parameters in the
3090 primitive.
- 3091 The Extensible Call barring information shall include the SS-Code for an individual call barring supplementary service.
3092 The Extensible Call barring information shall contain one or more Extensible Call Barring Features (Extensible Call
3093 Barring Feature is defined in subclause 7.6).
- 3094 The Extensible Call Barring Feature may include an Extensible Basic Service Group. This shall be interpreted according
3095 to the rules in subclause 8.8.1.4.
- 3096 The Extensible Call Barring Feature shall contain an extensible SS-Status parameter.
- 3097 If the VLR receives an Indication containing any Extensible Call Barring service codes which it does not
3098 support/allocate it returns them to the HLR in the parameter SS-Code List and discards the unsupported Extensible Call
3099 Barring service codes (no error is sent back). This parameter is used only by the VLR and if the SGSN receives this
3100 parameter it shall ignore it.
- 3101 CUG information List
- 3102 A list of CUG information list parameters (CUG information is defined in subclause 7.6). It includes CUG information
3103 either at location updating or at restoration or when it is changed.
- 3104 At location updating, restoration or when there is a change in CUG data, the HLR shall include the complete CUG-
3105 SubscriptionList and, if there are options per basic group, it shall also include the complete CUG-FeatureList. If there
3106 are not options per extensible basic service group the CUG-FeatureList shall not be included.
- 3107 In any dialogue, the first insertSubscriberData message which contains CUG information shall include a non-empty
3108 CUG-SubscriptionList.
- 3109 When the VLR receives CUG data it shall replace the stored CUG data with the received data set.

- 3110 If CUG-FeatureList is omitted in the Insert Subscriber Data operation VLR shall interpret that no options per extensible
3111 basic service group exist, and then it shall apply the default values i.e. no outgoing access, no incoming access, no
3112 preferential CUG exists.
- 3113 If CUG-Feature is received without preferential CUG, the VLR shall interpret that no preferential CUG applies.
- 3114 If the VLR detects that there is overlapping in the information received within a dialogue, it shall send the error
3115 Unexpected Data Value.
- 3116 Note that data consistency between CUG subscription data and CUG feature data is the responsibility of the HLR.
- 3117 If the VLR does not support the CUG service it returns its code to the HLR in the parameter SS-Code List and discards
3118 the received information (no error is sent back). This parameter is used only by the VLR and if the SGSN receives this
3119 parameter it shall ignore it.
- 3120 SS-Data List
- 3121 A list of Extensible SS-Data parameters (Extensible SS-Data is defined in subclause 7.6). It is sent for any other
3122 supplementary service than Call Forwarding, Call Barring, CUG and eMLPP either at location updating or at
3123 restoration or when they are changed. Each SS-Data parameter shall be treated independently of all other parameters in
3124 the primitive.
- 3125 The Extensible SS-Data shall include the SS-Code for an individual supplementary service.
- 3126 The Extensible SS-Data shall contain an Extensible SS-Status parameter and any subscription options that are applicable
3127 to the service defined by the SS-Code.
- 3128 The SS-Data may include a Basic Service Group List. This shall be interpreted according to the rules in
3129 subclause 8.8.1.4.
- 3130 If the VLR receives an Indication containing any supplementary service codes which it does not support/allocate it
3131 returns them to the HLR in the parameter SS-Code List and therefore discards the unsupported service codes received
3132 (no error is sent back). This parameter is used only by the VLR and if the SGSN receives this parameter it shall ignore
3133 it.
- 3134 Operator Determined Barring General data
- 3135 If it is included in a Request/Indication, it includes all the Operator Determined Barring categories that may be applied
3136 to a subscriber registered in any PLMN. This parameter is only included in a Request/Indication when the parameter
3137 Subscriber Status is set to the value Operator Determined Barring. Note that all General Operator Determined Barring
3138 Categories shall be set to their actual status.
- 3139 If the VLR or the SGSN receives an Indication containing Operator Determined Barring General Data which shows that
3140 the subscriber is subject to barring not supported / not allocated by the VLR or by the SGSN, it returns Operator
3141 Determined Barring General Data in the response to the HLR to show the barring categories which are not supported /
3142 not allocated by the VLR or by the SGSN. This parameter is used by the VLR and the SGSN.
- 3143 Operator Determined Barring HPLMN data
- 3144 It includes all the Operator Determined Barring categories that may be applied only to a subscriber registered in the
3145 HPLMN. Therefore, it shall only be transferred to the VLR or to the SGSN when the subscriber is roaming into the
3146 HPLMN and when the parameter Subscriber Status is set to the value Operator Determined Barring. Note that all
3147 HPLMN Operator Determined Barring Categories shall be set to their actual status.
- 3148 If Subscriber Status is set to the value Operator Determined Barring and no Operator Determined Barring HPLMN data
3149 is present then the VLR or the SGSN shall not apply any HPLMN specific ODB services to the subscriber. This
3150 parameter is used by the VLR and the SGSN.
- 3151 eMLPP Subscription Data
- 3152 If included in the Insert Subscriber Data request this parameter defines the priorities the subscriber might apply for a call
3153 (as defined in subclause 7.6). It contains both subparameters of eMLPP.
- 3154 If the VLR does not support the eMLPP service it returns its code to the HLR in the parameter SS-Code List and
3155 therefore discards the received information (no error is sent back).

3156 eMLPP subscription data that have been stored previously in a subscriber data record in the VLR are completely
3157 replaced by the new eMLPP subscription data received in a MAP_INSERT_SUBSCRIBER_DATA during either an
3158 Update Location or Restore Data procedure or a stand alone Insert Subscriber data procedure. This parameter is used
3159 only by the VLR and if the SGSN receives this parameter it shall ignore it.

3160 Roaming Restriction Due To Unsupported Feature

3161 The HLR may decide to include this parameter in the request if certain services or features are indicated as not
3162 supported by the MSC/VLR (e.g. Advice of Charge Charging Level).

3163 If this parameter is sent to the VLR the MSC area is restricted by the HLR and the VLR. This parameter is used only by
3164 the VLR and if the SGSN receives this parameter it shall ignore it.

3165 Regional Subscription Data

3166 If included in the Insert Subscriber Data request this parameter defines the subscriber's subscription area for the
3167 addressed VLR or for the addressed SGSN (as defined in subclause 7.6). It contains the complete list of up to 10 Zone
3168 Codes that apply to a subscriber in the currently visited PLMN. The HLR shall send only those Zone Codes which are
3169 stored against the CC and NDC of the VLR or the CC and NDC of the SGSN to be updated.

3170 NOTE: Support of this parameter is a network operator option and it will not be sent to networks which do not
3171 support Regional Subscription.

3172 Regional subscription data that have been stored previously in a subscriber data record in the VLR or in the SGSN are
3173 completely replaced by the regional subscription data received in an Insert Subscriber Data indication during either an
3174 Update Location or Restore Data procedure or a stand alone Insert Subscriber data procedure.

3175 After the regional subscription data are inserted the VLR or the SGSN shall derive whether its location areas are allowed
3176 or not. If the whole MSC or SGSN area is restricted it will be reported to HLR by returning the Regional Subscription
3177 Response.

3178 The VLR or the SGSN returns a Regional Subscription Response indicating that a problem with the Zone Code has been
3179 detected in one of the following cases:

- 3180 - Too Many Zone Codes: more than 10 Zone Codes are to be stored in the VLR or in the SGSN;
- 3181 - Regional Subscription Not Supported by the VLR or the SGSN;
- 3182 - Zone Codes Conflict: the VLR or the SGSN detects that the zone codes indicate conflicting service permission
3183 for a location area.

3184 Zone codes which have no mapping to location areas shall be ignored.

3185 If a sequence of MAP_INSERT_SUBSCRIBER_DATA services is used during a dialogue, Regional Subscription Data
3186 shall be accepted only in one service. Regional Subscription Data received in a subsequent service shall be rejected with
3187 the error Unexpected Data Value.

3188 If Regional Subscription Data are not included in any MAP_INSERT_SUBSCRIBER_DATA service, there is no
3189 restriction of roaming due to Regional Subscription. This parameter is used by the VLR and the SGSN.

3190 Voice Broadcast Data

3191 This parameter contains a list of group id's a user might have subscribed to; (VBS-Data is defined in subclause 7.6). It
3192 includes VBS information either at location updating or at restoration or when it is changed.

3193 At location updating, restoration or when there is a change in VBS data, the HLR shall include the complete VBS-Data.

3194 When the VLR receives VBS-Data within a dialogue it shall replace the stored VBS-data with the received data set. All
3195 subsequent VBS-dta received within this dialogue shall be interpreted as add-on data.

3196 If VBS-data is omitted in the Insert Subscriber Data operation the VLR shall keep the previously stored VBS data.

3197 If the VLR detects that there is overlapping in the information received within a dialogue, it shall send the error
3198 Unexpected Data Value. . This parameter is used only by the VLR and if the SGSN receives this parameter it shall
3199 ignore it.

3200 Voice Group Call Data

3201 This parameter contains a list of group id's a user might have subscribed to; see subclause 7.6.

3202 At location updating, restoration or when there is a change in VGCS data, the HLR shall include the complete VGCS-
3203 Data.

3204 When the VLR receives VGCS-Data within a dialogue it shall replace the stored VGCS-Data with the received data set.
3205 All VGCS-Data received within this dialogue shall be interpreted as add-on data.

3206 If VBCS-Data is omitted in the Insert Subscriber Data operation the VLR shall keep the previously stored VGCS-Data.

3207 If the VLR detects that there is overlapping in the information received within a dialogue, it shall send the error
3208 Unexpected Data Value. This parameter is used only by the VLR and if the SGSN receives this parameter it shall ignore
3209 it.

3210 North American Equal Access preferred Carrier Id List

3211 A list of the preferred carrier identity codes that are subscribed to.

3212 When the VLR receives this parameter from the HLR, it shall replace the previously stored preferred carrier identity
3213 codes with the received ones. It is not possible to delete all the preferred carrier identity codes from the VLR using this
3214 service. To delete all the preferred carrier identity codes from the VLR, the HLR shall use the
3215 MAP_CANCEL_LOCATION service.

3216 LSA Information

3217 If included in the ISD request, this parameter contains a list of localised service area identities a user might have
3218 subscribed to together with the priority of each localised service area; see subclause 7.6. The access right outside these
3219 localised service areas is also indicated. In all cases mentioned below, the LSA information shall only include LSA Data
3220 applicable to the VPLMN where the Subscriber is located. The VLR number, received in the MAP-
3221 UPDATE_LOCATION primitive, or the SGSN number, received in the MAP_UPDATE_GPRS_LOCATION
3222 primitive, can be used, alongside data stored in the HLR, to determine the LSA Data applicable to the VPLMN.

3223 At restoration, location updating or GPRS location updating the HLR shall include the complete set of applicable LSA
3224 Information.

3225 When there is a change in LSA data the HLR shall include at least the new and/or modified LSA data.

3226 When there is a change in the access right outside the localised service areas the HLR shall include the LSA only access
3227 indicator.

3228 When the SGSN or the VLR receives LSA information within a dialogue it shall check if the received data has to be
3229 considered as the entire LSA information. If so, it shall replace the stored LSA information with the received data set,
3230 otherwise it shall replace the data only for the modified LSA data (if any) and/or access right, and add the new LSA data
3231 (if any) to the stored LSA Information.

3232 If the entire LSA information is received, it shall always include the LSA only access indicator value together with the
3233 LSA data applicable for the PLMN (if any).

3234 If LSA Information is omitted in the Insert Subscriber Data operation the SGSN or the VLR shall keep the previously
3235 stored LSA Information.

3236 If the SGSN or the VLR detects that there is overlapping in the information received within a dialogue, it shall send the
3237 error Unexpected Data Value. This parameter is used by the VLR and the SGSN.

3238 IST Alert Timer

3239 This parameter contains the IST Alert timer value that must be used to inform the HLR about the call activities that the
3240 subscriber performs.

3241 At Location Updating, restoration, or when there is a change in the IST data defined for the Subscriber, the HLR shall
3242 include the IST Alert timer.

3243 LMU Identifier

3244 This parameter indicates the presence of an LMU. This parameter is used only by the VLR and shall be ignored if
3245 received by an SGSN.

3246 LCS Information

3247 This parameter provides the following LCS related information for an MS subscriber:

3248 - list of GMLCs in the HPLMN

3249 - privacy exception list

3250 - MO-LR list

3251 At restoration and location updating, the HLR shall include the complete LCS data of the subscriber.

3252 When there is a change in LCS subscriber data the HLR shall include at least the new and/or modified LCS data. LCS
3253 data that is not modified need not be included.

3254 The VLR shall keep any previously stored LCS Information that is not included in an Insert Subscriber Data operation.

3255 If the VLR detects that there is overlapping in the LCS information received within a dialogue, it shall send the error
3256 Unexpected Data Value.

3257 This parameter is used only by the VLR and shall be ignored if received by an SGSN.

3258 Super-Charger Supported In HLR

3259 This parameter is used by the HLR to indicate support for the Super-Charger functionality. If this parameter is present it
3260 shall include an indication of the age of the subscription data stored in the HLR.

3261 If this parameter is absent then the HLR does not support the Super-Charger functionality.

3262 SS-Code List

3263 The list of SS-Code parameters that are provided to a subscriber but are not supported/allocated by the VLR (SS-Code
3264 is defined in subclause 7.6). The list can only include individual SS-Codes that were sent in the service request. This
3265 parameter is used only by the VLR.

3266 Regional Subscription Response

3267 If included in the response this parameter indicates one of:

3268 - MSC Area Restricted entirely because of regional subscription;

3269 - SGSN Area Restricted entirely because of regional subscription;

3270 - Too Many Zone Codes to be inserted;

3271 - Zone Codes Conflict;

3272 - Regional Subscription not Supported by the VLR or by the SGSN.

3273 If the VLR determines after insertion of Regional Subscription Data that the entire MSC area is restricted, the VLR shall
3274 respond with a Regional Subscription Response indicating MSC Area Restricted. Otherwise MSC Area Restricted is not
3275 sent. The HLR shall check whether the current MSC area is no longer restricted.

3276 If the SGSN determines after insertion of Regional Subscription Data that the entire SGSN area is restricted, the SGSN
3277 shall respond with a Regional Subscription Response indicating SGSN Area Restricted. Otherwise SGSN Area
3278 Restricted is not sent. The HLR shall check whether the current SGSN area is no longer restricted. This parameter is
3279 used by the VLR and by the SGSN.

3280 VLR CAMEL Subscription Info

3281 This parameter is sent for subscribers who have CAMEL services which are invoked in the MSC. In CAMEL phase 1,
3282 this parameter contains only the O-CSI. In CAMEL Phase 2, this parameter may contain OCSI and SS-CSI. In CAMEL

3283 Phase 3, this parameter may contain O-CSI, D-CSI, SS-CSI, VT-CSI, SMS-CSI and M-CSI. If an O-CSI and/or VT-CSI
3284 is contained, TDP-Criteria may also be present in CAMEL Phase 2 or 3. The VLR CAMEL Subscription Info is sent at
3285 location updating or when any information in the applicable CAMEL Subscription Info in the HLR has been changed.
3286 The entire set of CAMEL Subscription Info is sent within one dialogue. If a set of CAMEL Subscription Info is already
3287 stored in the VLR it is replaced by the received data. If the VLR CAMEL Subscription Info is omitted in the Insert
3288 Subscriber Data operation the VLR shall keep the previously stored VLR CAMEL Subscription Info. Within one
3289 dialogue subsequent received data are interpreted as add-on data. If the VLR detects that there is overlapping in the
3290 information received within a dialogue, it shall send the error Unexpected Data Value. This parameter is used only by
3291 the VLR and if the SGSN receives this parameter it shall ignore it.

3292 The VLR CAMEL Subscription Info may contain the TIF-CSI (Translation Information Flag). for CAMEL Phase 2 and
3293 3 See 3G TS 23.072 for the use of this parameter and the conditions for its presence.

3294 Supported CAMEL Phases

3295 The use of this parameter and the requirements for its presence are specified in 3G TS 23.078. This parameter is used
3296 by the VLR and SGSN.

3297 A VLR or SGSN not supporting any CAMEL.Phase may omit this parameter.

3298 GPRS Subscription Data

3299 This parameter contains a list of PDP-contexts a user has subscribed to; see subclause 7.6.

3300 At GPRS location updating the HLR shall include the complete GPRS Subscription Data.

3301 When there is a change in GPRS subscriber data the HLR shall include only the new and/or modified PDP contexts.

3302 When the SGSN receives GPRS Subscription Data within a dialogue it shall check if the received data has to be
3303 considered as the entire GPRS subscription data. If so, it shall replace the stored GPRS Subscription Data with the
3304 received data set, otherwise it shall replace the data only for the modified PDP contexts (if any) and add the new PDP
3305 contexts (if any) to the stored GPRS Subscription Data.

3306 If GPRS Subscription Data is omitted in the Insert Subscriber Data operation the SGSN shall keep the previously stored
3307 GPRS Subscription Data.

3308 If the SGSN detects that there is overlapping in the information received within a dialogue, it shall send the error
3309 Unexpected Data Value. This parameter is used only by the SGSN and if the VLR receives this parameter it shall ignore
3310 it.

3311 SGSN CAMEL Subscription Info

3312 The SGSN CAMEL Subscription Info is sent at GPRS location updating or when any information in the applicable
3313 SGSN CAMEL Subscription Info in the HLR has been changed. In CAMEL Phase 3, this parameter may contain GPRS-
3314 CSI or/and SMS-CSI. The entire set of SGSN CAMEL Subscription Info is sent. If a set of SGSN CAMEL Subscription
3315 Info is already stored in the SGSN it is replaced by the received data. This parameter is used only by the SGSN and if
3316 the VLR receives this parameter it shall ignore it.

3317 Roaming Restricted In SGSN Due To Unsupported Feature

3318 The HLR may decide to include this parameter in the request if certain services or features are indicated as not
3319 supported by the SGSN. This parameter is used only by the SGSN and if the VLR receives this parameter it shall ignore
3320 it.

3321 User error

3322 Only one of the following values is applicable:

- 3323 - Unidentified subscriber;
- 3324 - Data missing;
- 3325 - Unexpected data value.

3326 8.8.1.4 Basic service information related to supplementary services

3327 A number of parameters that relate to supplementary services can be qualified by a Basic Service Group (or a Basic
3328 Service Group List). This subclause explains how this information is to be interpreted. Supplementary service
3329 parameters to which this subclause is applicable only apply to the basic service groups described in this subclause, and
3330 only those basic service groups shall be overwritten at the VLR.

3331 The Basic Service Group (or Basic Service Group List) is optional.

3332 If present the Basic Service Group (or the elements of the Basic Service Group List) shall be one of:

- 3333 - an Elementary Basic Service Group for which the supplementary service is applicable to at least one basic
3334 service in the group; and to which the subscriber has a subscription to at least one basic service in the group;
- 3335 - the group "All Teleservices" provided that the service is applicable to at least one teleservice and that the
3336 subscriber has a subscription to at least one teleservice that is in the same Elementary Basic Service Group as a
3337 teleservice to which the service is applicable;
- 3338 - the group "All Bearer Services" provided that the service is applicable to at least one bearer service and that the
3339 subscriber has a subscription to at least one bearer service that is in the same Elementary Basic Service Group as
3340 a basic service to which the service is applicable.

3341 If the Basic Service Group (or Basic Service Group List) is not present then the parameter shall apply to all Basic
3342 Service Groups.

3343 If the basic service information is not a single Elementary Basic Service Group then the parameter shall be taken as
3344 applying individually to all the Elementary Basic Service Groups for which:

- 3345 - the supplementary service is applicable to at least one basic service in the Basic Service Group; and
- 3346 - the subscriber has a subscription to at least one basic service in the Basic Service Group.

3347 The VLR is not required to store supplementary services data for Basic Service Groups that are not supported at the
3348 VLR.

3349 8.8.2 MAP-DELETE-SUBSCRIBER-DATA service

3350 8.8.2.1 Definition

3351 This service is used by an HLR to remove certain subscriber data from a VLR if the subscription of one or more
3352 supplementary services or basic services is withdrawn. Note that this service is not used in case of erasure or
3353 deactivation of supplementary services.

3354 Also this service is used by an HLR to remove GPRS subscription data from a SGSN.

3355 It is a confirmed service and consists of the primitives shown in table 8.8/2.

3356 8.8.2.2 Service primitives

3357 **Table 8.8/2: MAP-DELETE-SUBSCRIBER-DATA**

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(=)		
Camel Subscription Info Withdraw	C	C(=)		
Regional Subscription Data	C	C(=)		
VBS Group Indication	C	C(=)		
VGCS Group Indication	C	C(=)		
GPRS Subscription Data Withdraw	C	C(=)		
Roaming Restricted In SGSN Due To Unsupported Feature	C	C(=)		
LSA Information Withdraw	C	C(=)		
IST Information Withdraw	C	C(=)		
Regional Subscription Response			C	C(=)
GMLC List Withdraw	C	C(=)		
User error			C	C(=)
Provider error				O

3358

3359 8.8.2.3 Parameter use

3360 All parameters are described in subclause 7.6. The following clarifications are applicable:

3361 Basic service List

3362 A list of Extensible Basic service parameters (Extensible Basic service is defined in subclause 7.6). It is used when one,
3363 several or all basic services are to be withdrawn from the subscriber. If the VLR or the SGSN receives a value for an
3364 Extensible Basic Service which it does not support, it shall ignore that value. This parameter is used by the VLR and by
3365 the SGSN.

3366 SS-Code List

3367 A list of SS-Code parameters (SS-Code is defined in subclause 7.6). It is used when several or all supplementary
3368 services are to be withdrawn from the subscriber.

3369 There are three possible options:

3370 - deletion of basic service(s);

3371 The parameter Basic service List is only included.

3372 - deletion of supplementary service(s);

3373 The parameter SS-Code List is only included.

3374 - deletion of basic and supplementary services;

3375 Both Basic service List and SS-Code List are included.

3376 This parameter is used only by the VLR and if the SGSN receives this parameter it shall ignore it.

3377 Roaming Restriction Due To Unsupported Feature

3378 This parameter is used if Roaming Restriction Due To Unsupported Feature is deleted from the subscriber data. This
3379 may occur if unsupported features or services are removed from the subscriber data in the HLR.

3380 If this parameter is sent the VLR shall check if the current Location Area is possibly allowed now. This parameter is
3381 used only by the VLR and if the SGSN receives this parameter it shall ignore it.

3382 CAMEL Subscription Info Withdraw

3383 This parameter is used to indicate that CAMEL Subscription Info shall be deleted from the VLR or from the SGSN. All
3384 CAMEL Subscription Info for the subscriber shall be deleted. This parameter is used by the VLR and by the SGSN .

3385 Regional Subscription Identifier

3386 Contains one single Zone Code (as defined subclause 7.6) and is used if all Zone Codes shall be deleted from the
3387 subscriber data. When all the Zone Codes are deleted, the VLR or the SGSN shall check for its location areas whether
3388 they are allowed or not. If the whole MSC area is restricted, VLR will report it to HLR by returning the Regional
3389 Subscription Response "MSC Area Restricted". If the whole SGSN area is restricted, SGSN will report it to HLR by
3390 returning the Regional Subscription Response "SGSN Area Restricted".

3391 The binary coding of the Zone Code value received in a Delete Subscriber Data request shall not be checked by the
3392 VLR or by the SGSN.

3393 Note that support of this parameter is a network operator option and it shall not be sent to networks which do not
3394 support Regional Subscription.

3395 If Regional Subscription is not supported by the VLR or by the SGSN, the request for deletion of Zone Codes is refused
3396 by sending the Regional Subscription Response "Regional Subscription Not Supported" to the HLR.

3397 If no Zone Codes are stored in the respective subscriber data record, the request for deleting all Zone Code information
3398 shall be ignored and no Regional Subscription Response shall be returned. This parameter is used by the VLR and by
3399 the SGSN.

3400 VBS Group Indication

3401 Contains an indication (flag) which is used if all Group Id's shall be deleted from the subscriber data for the Voice
3402 Broadcast teleservice.

3403 If VBS is not supported in the VLR or no Group Ids are stored for VBS in the respective subscriber record, the request
3404 for deletion of all Group Ids shall be ignored. This parameter is used only by the VLR and if the SGSN receives this
3405 parameter it shall ignore it.

3406 VGCS Group Indication

3407 Contains an indication (flag) which is used if all Group Id's shall be deleted from the subscriber data for the Voice
3408 Group Call teleservice. This parameter is used only by the VLR and if the SGSN receives this parameter it shall ignore
3409 it.

3410 If VGCS is not supported in the VLR or no Group Ids are stored for VGCS in the respective subscriber record, the
3411 request for deletion of all Group Ids shall be ignored.

3412 GPRS Subscription Data Withdraw

3413 This parameter is used to indicate whether all GPRS Subscription Data for the subscriber shall be deleted or if only a
3414 subset of the stored GPRS Subscription Data for the subscriber shall be deleted. In the latter case only those PDP
3415 context whose identifiers are included in the subsequent identifier list will be deleted. This parameter is used only by the
3416 SGSN and if the VLR receives this parameter it shall ignore it.

3417 Roaming Restricted In SGSN Due To Unsupported Feature

3418 This parameter is used if Roaming Restricted In SGSN Due To Unsupported Feature is deleted from the GPRS
3419 subscriber data. This may occur if unsupported features or services are removed from the GPRS subscriber data in the
3420 HLR.

3421 If this parameter is sent the SGSN shall check if the current Location Area is possibly allowed now. This parameter is
3422 used only by the SGSN and if the VLR receives this parameter it shall ignore it.

3423 LSA Information Withdraw

3424 This parameter is used to indicate whether all LSA Information for the subscriber shall be deleted or if only a subset of
3425 the stored LSA Information for the subscriber shall be deleted. In the latter case only the LSA data whose LSA identities
3426 are included in the subsequent LSA data list will be deleted. This parameter is used by the VLR and the SGSN.

3427 IST Information Withdraw

3428 This parameter is used to indicate that the IST condition has been removed for the subscriber. See GSM 03.35 for the
3429 use of this parameter.

3430 Regional Subscription Response

3431 If included in the Delete Subscriber Data response this parameter indicates one of:

- 3432 - MSC Area Restricted
- 3433 - SGSN Area Restricted;
- 3434 - Regional Subscription Not Supported.

3435 This parameter is used by the VLR and by the SGSN.

3436 GMLC List Withdraw

3437 This parameter indicates that the subscriber's LCS GMLC List shall be deleted from the VLR.

3438 This parameter is used only by the VLR and shall be ignored if received by an SGSN.

3439 User error

3440 Only one of the following values is applicable:

- 3441 - Unidentified subscriber;
- 3442 - Data missing;
- 3443 - Unexpected data value.

3444 **8.9 Identity management services**3445 **8.9.1 MAP-PROVIDE-IMSI service**3446 **8.9.1.1 Definition**

3447 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
3448 identified itself with a TMSI not allocated to any subscriber in the VLR).

3449 It is a confirmed service and consists of the primitives shown in table 8.9/1.

3450 **8.9.1.2 Service primitives**

3451

Table 8.9/1: MAP-PROVIDE-IMSI

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

3452

3453 **8.9.1.3 Parameter use**

3454 All parameters are described in subclause 7.6. The following clarifications are applicable:

3455 IMSI

3456 This parameter is received when the request is successfully carried out. It contains the requested IMSI.

3457 User error

3458 Only one of the following values is applicable:

3459 - Absent subscriber.

3460 8.9.2 MAP-FORWARD-NEW-TMSI service

3461 8.9.2.1 Definition

3462 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).

3464 It is a confirmed service and consists of the primitives shown in table 8.9/2.

3465 8.9.2.2 Service primitives

3466 **Table 8.9/2: MAP-FORWARD-NEW-TMSI**

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

3467

3468 8.9.2.3 Parameter use

3469 The parameter TMSI is described in subclause 7.6.

3470 8.10 Fault recovery services

3471 8.10.1 MAP_RESET service

3472 8.10.1.1 Definition

3473 This service is used by the HLR, after a restart, to indicate to a list of VLRs or SGSNs that a failure occurred.

3474 The MAP_RESET service is a non-confirmed service using the service primitives defined in table 8.10/1

3475 8.10.1.2 Service primitives

3476 **Table 8.10/1: MAP_RESET**

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

3477

3478 8.10.1.3 Parameter definition and use

3479 Invoke Id

3480 See definition in subclause 7.6.1.

3481 HLR number

3482 See definition in subclause 7.6.2.

3483 HLR Id LIST

3484 The HLR Id List is a list of HLR Id. If the parameter is present in the indication, the VLR or SGSN may base the
 3485 retrieval of subscribers to be restored on their IMSI: the subscribers affected by the reset are those whose IMSI leading
 3486 digits are equal to one of these numbers. If the parameter is absent, subscribers to be restored are those for which the
 3487 OriginatingEntityNumber received at location updating time matches the equivalent parameter of the Reset Indication.

3488 **8.10.2 MAP_FORWARD_CHECK_SS_INDICATION service**3489 **8.10.2.1 Definition**

3490 This service may be used by an HLR as an implementation option, to indicate to a mobile subscriber that supplementary
 3491 services parameters may have been altered, e.g. due to a restart. If received from the HLR, the VLR shall forward this
 3492 indication to the MSC, which in turn forwards it to the MS. The HLR only sends this indication after successful
 3493 completion of the subscriber data retrieval from HLR to VLR that ran embedded in a MAP_UPDATE_LOCATION
 3494 procedure.

3495 The MAP_FORWARD_CHECK_SS_INDICATION service is a non-confirmed service using the service primitives
 3496 defined in table 8.10/2.

3497 **8.10.2.2 Service primitives**3498 **Table 8.10/2: MAP_FORWARD_CHECK_SS_INDICATION**

Parameter name	Request	Indication
Invoke Id	M	M(=)

3499

3500 **8.10.2.3 Parameter definition and use**3501 Invoke Id

3502 See definition in subclause 7.6.1.

3503 **8.10.3 MAP_RESTORE_DATA service**3504 **8.10.3.1 Definition**

3505 This service is invoked by the VLR on receipt of a MAP_PROVIDE_ROAMING_NUMBER indication for an unknown
 3506 IMSI, or for a known IMSI with the indicator "Confirmed by HLR" set to "Not confirmed". The service is used to
 3507 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
 3508 subscriber's IMSI record.

3509 The MAP_RESTORE_DATA service is a confirmed service using the service primitives defined in table 6.10/3.

3510 8.10.3.2 Service primitives

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

3511
3512

Table 8.10/3: MAP_RESTORE_DATA

3513 8.10.3.3 Parameter definitions and use

3514 Invoke Id

3515 See definition in subclause 5.6.1.

3516 IMSI

3517 See definition in subclause 5.6.2.

3518 LMSI3519 See definition in subclause 5.6.2. It is an operator option to provide the LMSI from the VLR; it is mandatory for the
3520 HLR to support the LMSI handling procedures.3521 Supported CAMEL Phases3522 This parameter indicates which phases of CAMEL are supported. Must be present if a CAMEL phase different from
3523 phase 1 is supported. Otherwise may be absent.3524 HLR number3525 See definition in subclause 5.6.2. The presence of this parameter is mandatory in case of successful outcome of the
3526 service.3527 MS Not Reachable Flag3528 See definition in subclause 5.6.8. This parameter shall be present in case of successful outcome of the service, if the
3529 "MS Not Reachable flag" was set in the HLR.3530 User error3531 In case of unsuccessful outcome of the service, an error cause shall be returned by the HLR. The following error causes
3532 defined in subclause 5.6.1 may be used, depending on the nature of the fault:

3533 - unknown subscriber;

3534 - system failure;

3535 - unexpected data value;

3536 - data missing.

3537 Provider error

3538 For definition of provider errors see subclause 5.6.1.

3539 8.11 Subscriber Information services

3540 8.11.1 MAP-ANY-TIME-INTERROGATION service

3541 8.11.1.1 Definition

3542 This service is used by the gsmSCF, to request information (e.g. subscriber state and location) from the HLR or the
3543 GMLC at any time.

3544 When this service is used to the HLR, the subscriber state or location may be requested.

3545 When this service is used to the GMLC, only the location may be requested.

3546 8.11.1.2 Service primitives

3547 **Table 8.11/1: Any_Time_Interrogation**

Parameter name	Request	Indication	Response	Confirm
Invoke id	M	M(=)	M(=)	M(=)
Requested Info	M	M(=)		
gsmSCF-Address	M	M(=)		
IMSI	C	C(=)		
MSISDN	C	C(=)		
Location Information			C	C(=)
Subscriber State			C	C(=)
User error			C	C(=)
Provider error				O

3548

3549 8.11.1.3 Parameter definition and use

3550 All parameters are described in subclause 7.6.

3551 The HLR or GMLC may be able to use the value of the parameter gsmSCF-address to screen a
3552 MAP_Any_Time_Interrogation indication.

3553 The use of the parameters and the requirements for their presence are specified in 3G TS 23.078.

3554 User error

3555 This parameter is sent by the responder when an error is detected and if present, takes one of the following values:

3556 - System Failure;

3557 - Any Time Interrogation Not Allowed;

3558 - Data Missing;

3559 - Unexpected Data Value;

3560 - Unknown Subscriber.

3561 Provider error

3562 These are defined in subclause 7.6.1.

3563 8.11.2 MAP-PROVIDE-SUBSCRIBER-Info service

3564 8.11.2.1 Definition

3565 This service is used to request information (e.g. subscriber state and location) from the VLR at any time.

3566 8.11.2.2 Service primitives

3567 **Table 8.11/2: Provide_Subscriber_Information**

Parameter name	Request	Indication	Response	Confirm
Invoke id	M	M(=)	M(=)	M(=)
Requested Info	M	M(=)		
IMSI	M	M(=)		
LMSI	U	O		
Location Information			C	C(=)
Subscriber State			C	C(=)
User error			C	C(=)
Provider error				O

3568

3569 8.11.2.3 Parameter definition and use

3570 All parameters are defined in section 7.6. The use of these parameters and the requirements for their presence are
3571 specified in GSM 03.18

3572 User error

3573 This parameter is sent by the responder when an error is detected and if present, takes one of the following values:

3574 - Data Missing;

3575 - Unexpected Data Value.

3576 Provider error

3577 These are defined in subclause 7.6.1.

3578 8.11.3 MAP-ANY-TIME-SUBSCRIPTION-INTERROGATION service

3579 8.11.3.1 Definition

3580 This service is used by the gsmSCF, to request subscription information (e.g. call forwarding supplementary service data
3581 or CSI) from the HLR at any time.

3582 8.11.3.2 Service primitives

3583 **Table 8.11/3: Any_Time_Subscription_Interrogation**

Parameter name	Request	Indication	Response	Confirm
Invoke id	M	M(=)	M(=)	M(=)
Requested Subscription Info	M	M(=)		
GsmSCF-Address	M	M(=)		
IMSI	C	C(=)		
MSISDN	C	C(=)		
Call Forwarding Data			C	C(=)
Call Barring Data			C	C(=)
ODB Data			C	C(=)
CAMEL Subscription Info			C	C(=)
Supported CAMEL phases in VLR			C	C(=)
Supported CAMEL phases in SGSN			C	C(=)
User error			C	C(=)
Provider error				O

3584

3585 8.11.3.3 Parameter definition and use

3586 All parameters are described in subclause 7.6.

3587 The HLR may be able to use the value of the parameter gsmSCF-address to screen a
 3588 MAP_Any_Time_Subscription_Interrogation indication.

3589 The use of the parameters and the requirements for their presence are specified in 3G TS 23.078.

3590 User error

3591 This parameter is sent by the responder when an error is detected and if present, takes one of the following values:

- 3592 - Unexpected Data Value;
- 3593 - Unknown Subscriber;
- 3594 - BearerServiceNotProvisioned;
- 3595 - TeleserviceNotProvisioned;
- 3596 - CallBarred;
- 3597 - IllegalSS-Operation;
- 3598 - SS-NotAvailable;
- 3599 - InformationNotAvailable
- 3600 - Any Time Subscription Interrogation Not Allowed;
- 3601 - Data Missing;

3602 Provider error

3603 These are defined in subclause 7.6.1.

3604 **8.11.4 MAP-ANY-TIME-MODIFICATION service**3605 **8.11.4.1 Definition**

3606 This service is used by the gsmSCF, to modify information of the HLR at any time.

3607 **8.11.4.2 Service primitives**3608 **Table 8.11/4: Any_Time_Modification**

Parameter name	Request	Indication	Response	Confirm
Invoke id	M	M(=)	M(=)	M(=)
gsmSCF-Address	M	M(=)		
IMSI	C	C(=)		
MSISDN	C	C(=)		
Modification request for SS information	C	C(=)		
Modification request for CSI	C	C(=)		
Ext Forwarding information-for-CSE			C	C(=)
Ext Call barring information-for-CSE			C	C(=)
CAMEL subscription info			C	C(=)
User error			C	C(=)
Provider error				O

3609

3610 **8.11.4.3 Parameter definition and use**

3611 All parameters are described in subclause 7.6.

3612 The HLR may be able to use the value of the parameter gsmSCF-address to screen an MAP_Any_Time_Modification indication.

3614 The use of these parameters and the requirements for their presence are specified in 3G TS 23.078.

3615 User error

3616 This parameter is sent by the responder when an error is detected and if present, takes one of the following values:

3617 - Any Time Modification Not Allowed;

3618 - Data Missing;

3619 - Unexpected Data Value;

3620 - Unknown Subscriber;

3621 - Bearer service not provisioned;

3622 This error is returned only if not even a subset of the requested bearer service group has been subscribed to.

3623 - Teleservice not provisioned;

3624 This error is returned only if not even a subset of the requested teleservice group has been subscribed to.

3625 - Call Barred;

3626 - Illegal SS operation;

3627 - SS error status;

- 3628 - SS incompatibility;
 3629 - SS subscription violation;
 3630 - Information Not Available.

3631 Provider error

3632 These are defined in subclause 7.6.1.

8.11.5 MAP-NOTE-SUBSCRIBER-DATA-MODIFIED service

8.11.5.1 Definition

This service is used by the HLR to inform the gsmSCF that subscriber data have been modified.

8.11.5.2 Service primitives

Table 8.11/5: Note_Subscriber_Data_Modified

Parameter name	Request	Indication	Response	Confirm
Invoke id	M	M(=)	M(=)	M(=)
IMSI	M	M(=)		
MSISDN	M	M(=)		
Type of Modification	M	M(=)		
User error			C	C(=)
Provider error				O

8.11.5.3 Parameter definition and use

Invoke id

See subclause 7.6.1 for the use of this parameter.

IMSI

See subclause 7.6.2 for the use of this parameter.

MSISDN

See subclause 7.6.2 for the use of this parameter.

Type of Modification

This parameter indicates which subscriber data have been modified

It takes one of the following values:

- Call Forwarding SS Data modified;
- Call Barring SS Data modified;
- Operator Determined Barring Data modified;
- CAMEL Subscription Information modified.

User error

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

- Unexpected Data Value;
- Unknown Subscriber.

Provider error

These are defined in subclause 7.6.1.

The use of the parameters and the requirements for their presence are specified in 3G TS 23.078.

3633 9 Operation and maintenance services

3634 9.1 Subscriber tracing services

3635 9.1.1 MAP-ACTIVATE-TRACE-MODE service

3636 9.1.1.1 Definition

3637 This service is used between the HLR and the VLR to activate subscriber tracing in the VLR.

3638 Also this service is used between the HLR and the SGSN to activate subscriber tracing in the SGSN.

3639 The MAP-ACTIVATE-TRACE-MODE service is a confirmed service using the primitives from table 9.1/1.

3640 9.1.1.2 Service primitives

3641 **Table 9.1/1: MAP-ACTIVATE-TRACE-MODE**

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

3642

3643 9.1.1.3 Parameter use

3644 Invoke id

3645 See definition in subclause 7.6.1.

3646 IMSI

3647 See definition in subclause 7.6.2. The IMSI is a mandatory parameter in a stand-alone operation.

3648 Trace reference

3649 See definition in subclause 7.6.10.

3650 Trace type

3651 See definition in subclause 7.6.10.

3652 OMC Id

3653 See definition in subclause 7.6.2. The use of this parameter is an operator option.

3654 User error

3655 The following errors defined in subclause 7.6.1 may be used, depending on the nature of the fault:

3656 - Unidentified Subscriber;

3657 - Facility Not Supported;

3658 - Tracing Buffer Full;

- 3659 - System Failure;
 3660 - Unexpected Data Value;
 3661 - Data missing.

3662 Provider error

3663 For definition of provider errors see subclause 7.6.1.

3664 9.1.2 MAP-DEACTIVATE-TRACE-MODE service

3665 9.1.2.1 Definition

3666 This service is used between the VLR and the HLR for deactivating subscriber tracing in the VLR.

3667 Also this service is used between the SGSN and the HLR for deactivating subscriber tracing in the SGSN.

3668 The MAP-DEACTIVATE-TRACE-MODE service is a confirmed service using the primitives from table 9.1/2.

3669 9.1.2.2 Service primitives

3670

Table 9.1/2: MAP-DEACTIVATE-TRACE-MODE

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

3671

3672 9.1.2.3 Parameter use

3673 Invoke id

3674 See definition in subclause 7.6.1.

3675 IMSI

3676 See definition in subclause 7.6.2. The IMSI is a mandatory parameter in a stand-alone operation.

3677 Trace reference

3678 See definition in subclause 7.6.10.

3679 User error

3680 The following errors defined in subclause 7.6.1 may be used, depending on the nature of the fault:

- 3681 - Unidentified Subscriber;
 3682 - Facility Not Supported;
 3683 - System Failure;
 3684 - Unexpected Data Value;
 3685 - Data missing.

3686 Provider error

3687 For definition of provider errors see subclause 7.6.1.

3688 9.1.3 MAP-TRACE-SUBSCRIBER-ACTIVITY service

3689 9.1.3.1 Definition

3690 This service is used between the VLR and the MSC to activate the subscriber tracing in the MSC.

3691 The MAP-TRACE-SUBSCRIBER-ACTIVITY service is a non-confirmed service using the primitives from table 9.1/3.

3692 9.1.3.2 Service primitives

3693 **Table 9.1/3: MAP-TRACE-SUBSCRIBER-ACTIVITY**

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

3694

3695 9.1.3.3 Parameter use

3696 Invoke id

3697 See definition in subclause 7.6.1.

3698 IMSI

3699 See definition in subclause 7.6.2. The controlling MSC shall provide either the IMSI or the IMEI to the servicing MSC.

3700 Trace reference

3701 See definition in subclause 7.6.10.

3702 Trace type

3703 See definition in subclause 7.6.10.

3704 OMC Id

3705 See definition in subclause 7.6.2. The use of this parameter is an operator option.

3706 9.2 Other operation and maintenance services

3707 9.2.1 MAP-SEND-IMSI service

3708 9.2.1.1 Definition

3709 This service is used by a VLR in order to fetch the IMSI of a subscriber in case of some Operation & Maintenance
3710 procedure where subscriber data are needed in the Visited PLMN and MSISDN is the only subscriber's identity known.

3711 It is a confirmed service and consists of the primitive shown in table 9.2/1.

3712 9.2.1.2 Service primitives

3713

Table 9.2/1: MAP-SEND-IMSI

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

3714

3715 9.2.1.3 Parameter use

3716 All parameters are described in subclause 7.6. The following clarifications are applicable:

3717 User error

3718 Only one of the following values is applicable:

- 3719 - Unknown subscriber;
- 3720 - Unexpected data value;
- 3721 - Data missing.

3722 **10 Call handling services**3723 **10.1 MAP_SEND_ROUTING_INFORMATION service**3724 **10.1.1 Definition**

3725 This service is used between the Gateway MSC and the HLR. The service is invoked by the Gateway MSC to perform
 3726 the interrogation of the HLR in order to route a call towards the called MS.

3727 This is a confirmed service using the primitives listed in table 10.1/1.

3728 This service is also used between the GMSC and the NPLR.

3729 10.1.2 Service primitives

3730 Table 10.1/1: MAP_SEND_ROUTING_INFORMATION parameters

Parameter name	Request	Indication	Response	Confirm
Invoke Id	M	M(=)	M(=)	M(=)
Interrogation Type	M	M(=)		
GMSC Address	M	M(=)		
MSISDN	M	M(=)	C	C(=)
OR Interrogation	C	C(=)		
OR Capability	C	C(=)		
CUG Interlock	C	C(=)	C	C(=)
CUG Outgoing Access	C	C(=)	C	C(=)
Number of Forwarding	C	C(=)		
Network Signal Info	C	C(=)		
Supported CAMEL Phases	C	C(=)		
Suppress T-CSI	C	C(=)		
Suppression of Announcement	C	C(=)		
Call Reference Number	C	C(=)		
Forwarding Reason	C	C(=)		
Basic Service Group	C	C(=)		
Alerting Pattern	C	C(=)		
CCBS Call	C	C(=)		
Supported CCBS Phase	C	C(=)		
Additional Signal Info	C	C(=)		
IST Support Indicator	C	C(=)		
Pre-paging supported	C	C(=)		
Call Diversion Treatment Indicator	C	C(=)		
IMSI			C	C(=)
MSRN			C	C(=)
Forwarding Data			C	C(=)
Forwarding Interrogation Required			C	C(=)
VMSC address			C	C(=)
GMSC Camel Subscription Info			C	C(=)
Location Information			C	C(=)
Subscriber State			C	C(=)
Basic Service Code			C	C(=)
CUG Subscription Flag			C	C(=)
North American Equal Access preferred			U	C(=)
Carrier Id				
User error			C	C(=)
SS-List			U	C(=)
CCBS Target			C	C(=)
Keep CCBS Call Indicator			C	C(=)
IST Alert Timer			C	C(=)
Number Portability Status			U	C(=)
Provider error				O

3731

3732 10.1.3 Parameter use

3733 See subclause 7.6 for a definition of the parameters used in addition to the following. Note that:

3734 - a conditional parameter whose use is defined only in 3G TS 23.078 shall be absent if the sending entity does not
3735 support CAMEL;

3736 - a conditional parameter whose use is defined only in GSM 03.79 shall be absent if the sending entity does not
3737 support optimal routing;

3738 - a conditional parameter whose use is defined only in 3G TS 23.078 & GSM 03.79 shall be absent if the sending
3739 entity supports neither CAMEL nor optimal routing.

3740 Interrogation Type

3741 See GSM 03.79 [99] for the use of this parameter.

- 3742 GMSC address
- 3743 The E.164 address of the GMSC.
- 3744 MSISDN
- 3745 This is the Mobile Subscriber ISDN number assigned to the called subscriber. In the Request & Indication it is the
3746 number received by the GMSC in the IAM. If the call is to be forwarded and the HLR supports determination of the
3747 redirecting number, the HLR inserts the basic MSISDN in the Response.
- 3748 See GSM 03.66 [108] for the use of this parameter and the conditions for its presence in the response.
- 3749 OR Interrogation
- 3750 See GSM 03.79 [99] for the use of this parameter and the conditions for its presence.
- 3751 OR Capability
- 3752 See GSM 03.79 [99] for the use of this parameter and the conditions for its presence.
- 3753 CUG Interlock
- 3754 See GSM 03.18 [97] for the use of this parameter and the conditions for its presence.
- 3755 CUG Outgoing Access
- 3756 See GSM 03.18 [97] for the use of this parameter and the conditions for its presence.
- 3757 Number of Forwarding
- 3758 See GSM 03.18 [97] for the use of this parameter and the conditions for its presence.
- 3759 Network Signal Info
- 3760 See GSM 03.18 [97] for the conditions for the presence of the components of this parameter.
- 3761 Supported CAMEL Phases
- 3762 The use of this parameter and the requirements for its presence are specified in 3G TS 23.078
- 3763 T-CSI Suppression
- 3764 The use of this parameter and the requirements for its presence are specified in 3G TS 23.078
- 3765 Suppression Of Announcement
- 3766 The use of this parameter and the requirements for its presence are specified in 3G TS 23.078
- 3767 Call Reference Number
- 3768 The use of this parameter and the conditions for its presence are specified in 3G TS 23.078 [98] and GSM 03.79 [99].
- 3769 Forwarding Reason
- 3770 See GSM 03.79 [99] for the use of this parameter and the conditions for its presence.
- 3771 Basic Service Group
- 3772 See GSM 03.79 [99] for the use of this parameter and the conditions for its presence.
- 3773 Alerting Pattern
- 3774 See GSM 03.18 [97] for the use of this parameter and the conditions for its presence.
- 3775 CCBS Call
- 3776 See 3G TS 23.093 [107] for the use of this parameter and the conditions for its presence.

3777 Supported CCBS Phase3778 Additional Signal Info

3779 See 3G TS 23.081 [27] for the conditions for the presence of the components of this parameter.

3780 This parameter indicates by its presence that CCBS is supported and the phase of CCBS which is supported.

3781 IST Support Indicator

3782 This parameter is used to indicate to the HLR that the GMSC supports basic IST functionality, that is, the GMSC is able
3783 to terminate the subscriber call activity that originated the IST Alert when it receives the IST Alert response indicating
3784 that the call(s) shall be terminated. If this parameter is not included in the Send Routing Information indication and the
3785 subscriber is marked as an IST subscriber, then the HLR may limit the service for the call (by barring the incoming call
3786 if it is not subject to forwarding, or suppressing Call Forwarding from the GMSC), or allow the call assuming the
3787 associated risk of not having the basic IST mechanism available.

3788 This parameter can also indicate that the GMSC supports the IST Command, including the ability to terminate all calls
3789 being carried for the identified subscriber by using the IMSI as a key. If this additional capability is not included in the
3790 Send Routing Information indication and the subscriber is marked as an IST subscriber, then the HLR may limit the
3791 service for the subscriber (by barring the incoming calls if they are not subject to forwarding, or suppressing Call
3792 Forwarding from the GMSC), or allow the incoming calls assuming the associated risk of not having the IST Command
3793 mechanism available.

3794 Pre-paging supported

3795 See 3G TS 23.018 for the use of this parameter and the conditions for its presence.

3796 Call Diversion Treatment Indicator

3797 This parameter indicates whether or not call diversion is allowed.

3798 IMSI

3799 See GSM 03.18 [97] and GSM 03.66 [108] for the use of this parameter and the conditions for its presence.

3800 MSRN

3801 See GSM 03.18 [97], GSM 03.66 [108] and GSM 03.79 [99] for the use of this parameter and the conditions for its
3802 presence. If the NPLR returns only the MSISDN-number without Routeing Number to the GMSC, the MSISDN-number
3803 shall be returned as MSRN.

3804 Forwarding Data

3805 This parameter includes the forwarded-to number, the forwarding reason and the forwarding options Notification to
3806 calling party and Redirecting presentation, and can include the forwarded-to subaddress. See GSM 03.18 [97] and GSM
3807 03.79 [99] for the conditions for the presence of its components.

3808 Forwarding Interrogation Required

3809 See GSM 03.79 [99] for the use of this parameter and the conditions for its presence.

3810 VMSC address

3811 See GSM 03.79 [99] for the use of this parameter and the conditions for its presence.

3812 GMSC CAMEL Subscription Info

3813 The use of this parameter and the requirements for its presence are specified in 3G TS 23.078

3814 Location Information

3815 The use of this parameter and the requirements for its presence are specified in 3G TS 23.078

3816 Subscriber State

3817 The use of this parameter and the requirements for its presence are specified in 3G TS 23.078

3818 CUG Subscription Flag

3819 The use of this parameter and the requirements for its presence are specified in 3G TS 23.078.

3820 North American Equal Access preferred Carrier Id

3821 This parameter is returned to indicate the preferred carrier identity to be used to setup the call (i.e. forwarding the call or
3822 establishing the roaming leg).

3823 SS-List

3824 This parameter includes SS-codes and will be returned as an operator option. The HLR shall not send PLMN-specific
3825 SS-codes across PLMN boundaries. However if the GMSC receives PLMN-specific SS-codes from a foreign PLMN's
3826 HLR the GMSC may ignore it. If the GMSC attempts to process the PLMN specific SS codes, this may lead to
3827 unpredictable behaviour but the GMSC shall continue call processing.

3828 Basic Service Code

3829 The use of this parameter and the requirements for its presence are specified in 3G TS 23.078.

3830 If the CAMEL service is not involved, this parameter includes the basic service code and will be returned as an operator
3831 option. The HLR shall not send a PLMN-specific Basic Service Code across PLMN boundaries. However if the GMSC
3832 receives a PLMN-specific Basic Service Code from a foreign PLMN's HLR the GMSC may ignore it. If the GMSC
3833 attempts to process the PLMN specific Basic Service codes, this may lead to unpredictable behaviour but the GMSC
3834 shall continue call processing.

3835 CCBS Target

3836 See GSM 03.93 for the use of this parameter and the conditions for its presence.

3837 Keep CCBS Call Indicator

3838 See GSM 03.93 for the use of this parameter and the conditions for its presence.

3839 IST Alert Timer

3840 It includes the IST Alert timer value that must be used to inform the HLR about the call activities that the subscriber
3841 performs. This parameter is only sent to the GMSC in response to a Send Routing Information request which indicates
3842 the the GMSC supports IST.

3843 Number Portability Status

3844 This parameter indicates the number portability status of the subscriber. This parameter may be present if the sender of
3845 SRIack is NPLR.

3846 User error

3847 This parameter is sent by the responder when an error is detected and if present, takes one of the following values:

3848 - Unknown Subscriber;

3849 - Number changed;

3850 - Call Barred;

3851 This error will indicate that either incoming calls are barred for this MS or that calls are barred due to Operator
3852 Determined Barring (see GSM 02.41 for a definition of this network feature).

3853 - CUG Reject;

3854 The value of this error cause will indicate the reason for CUG Reject.

3855 - Bearer Service Not Provisioned;

- 3856 - Teleservice Not Provisioned;
- 3857 A subscription check has been performed and the call has not passed the check due to incompatibility with regard
3858 to the requested service. Depending on the nature of the incompatibility, either of these messages will be
3859 returned.
- 3860 - Facility Not Supported;
- 3861 - Absent Subscriber;
- 3862 This indicates that the location of the MS is not known (either the station is not registered and there is no location
3863 information available or the Provide Roaming Number procedure fails due to IMSI detached flag being set), or
3864 the GMSC requested forwarding information with a forwarding reason of not reachable, and the call forwarding
3865 on MS not reachable service is not active.
- 3866 - Busy Subscriber;
- 3867 This indicates that Call Forwarding on Busy was not active for the specified basic service group when the GMSC
3868 requested forwarding information with a forwarding reason of busy.
- 3869 The error may also indicate that the subscriber is busy due to an outstanding CCBS recall. In the error data it may
3870 then be specified that CCBS is possible for the busy encountered call.
- 3871 - No Subscriber Reply;
- 3872 This indicates that Call Forwarding on No Reply was not active for the specified basic service group when the
3873 GMSC requested forwarding information with a forwarding reason of no reply.
- 3874 - OR Not Allowed;
- 3875 This indicates that the HLR is not prepared to accept an OR interrogation from the GMSC, or that calls to the
3876 specified subscriber are not allowed to be optimally routed.
- 3877 - Forwarding Violation;
- 3878 - System Failure;
- 3879 - Data Missing;
- 3880 - Unexpected Data Value.
- 3881 See subclause 7.6 for a definition of these errors.
- 3882 Provider error
- 3883 These are defined in subclause 7.6.

3884 10.2 MAP_PROVIDE_ROAMING_NUMBER service

3885 10.2.1 Definition

3886 This service is used between the HLR and VLR. The service is invoked by the HLR to request a VLR to send back a
3887 roaming number to enable the HLR to instruct the GMSC to route an incoming call to the called MS.

3888 This is a confirmed service which uses the Primitives described in table 10.2/1.

3889 10.2.2 Service primitives

3890 Table 10.2/1: MAP_PROVIDE_ROAMING_NUMBER parameters

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(=)		
Suppression Of Announcement	C	C(=)		
Call Reference Number	C	C(=)		
GMSC Address	C	C(=)		
OR Interrogation	C	C(=)		
OR Not Supported in GMSC	C	C(=)		
Alerting Pattern	C	C(=)		
CCBS Call	C	C(=)		
Supported CAMEL Phases in GMSC	C	C(=)		
Additional Signal Info	C	C(=)		
Pre-paging supported	C	C(=)		
Roaming Number			C	C(=)
User error			C	C(=)
Provider error				O

3891

3892 10.2.3 Parameter use

3893 See subclause 7.6 for a definition of the parameters used, in addition to the following. Note that:

- 3894 - a conditional parameter whose use is defined only in 3G TS 23.078 shall be absent if the sending entity does not
3895 support CAMEL;
- 3896 - a conditional parameter whose use is defined only in GSM 03.79 shall be absent if the sending entity does not
3897 support optimal routing;
- 3898 - a conditional parameter whose use is defined only in 3G TS 23.078 & GSM 03.79 shall be absent if the sending
3899 entity supports neither CAMEL nor optimal routing.

3900 IMSI

3901 This is the IMSI of the called Subscriber.

3902 MSC Number

3903 This is the ISDN number assigned to the MSC currently serving the MS. The MSC number will have been stored in the
3904 HLR as provided at location updating.

3905 MSISDN

3906 See GSM 03.18 [97] for the use of this parameter and the conditions for its presence.

3907 LMSI

3908 See GSM 03.18 [97] for the use of this parameter and the conditions for its presence.

3909 GSM Bearer Capability

3910 See GSM 03.18 [97] for the use of this parameter and the conditions for its presence.

3911 This information is passed according to the rules specified in TS GSM 09.07.

3912 There may be two GSM Bearer Capabilities supplied.

3913 Network Signal Info

3914 See GSM 03.18 [97] for the conditions for the presence of the components of this parameter.

3915 Suppression Of Announcement

3916 The use of this parameter and the requirements for its presence are specified in 3G TS 23.078.

3917 Call Reference Number

3918 The use of this parameter and the conditions for its presence are specified in 3G TS 23.078 [98] and GSM 03.79 [99].

3919 GMSC Address

3920 The use of this parameter and the conditions for its presence are specified in 3G TS 23.078 [98] and GSM 03.79 [99].

3921 OR Interrogation

3922 See GSM 03.79 [99] for the use of this parameter and the conditions for its presence.

3923 OR Not Supported in GMSC

3924 See GSM 03.79 [99] for the use of this parameter and the conditions for its presence.

3925 Supported CAMEL Phases in GMSC

3926 See 3G TS 23.078 [98] for the use of this parameter and the conditions for its presence.

3927 Alerting Pattern

3928 See 3G TS 23.078 [98] for the use of this parameter and the conditions for its presence.

3929 CCBS Call

3930 See 3G TS 23.093 [107] for the use of this parameter and the conditions for its presence.

3931 Additional Signal Info

3932 See GSM 03.81 [27] for the conditions for the presence of the components of this parameter.

3933 Pre-paging supported

3934 See 3G 23.018 for the use of this parameter and the conditions for its presence.

3935 Roaming Number

3936 See GSM 03.18 [97] for the use of this parameter and the conditions for its presence.

3937 User error

3938 This parameter is sent by the responder when an error is detected and if present, takes one of the following values:

3939 - Absent Subscriber;

3940 This error will be returned if the IMSI detach flag is set.

3941 - No Roaming Number Available;

3942 - OR Not Allowed;

3943 This indicates that the MAP_PROVIDE_ROAMING_NUMBER indication included the OR interrogation
3944 indicator, but the VLR does not support optimal routing.

3945 - Facility Not Supported;

3946 - System Failure;

3947 - Data Missing;

3948 - Unexpected Data Value.

3949 See subclause 7.6 for a definition of these reasons.

3950 Provider error

3951 These are defined in subclause 7.6.

3952 10.3 MAP_RESUME_CALL_HANDLING service

3953 10.3.1 Definition

3954 This service is used between the terminating VMSC and the GMSC. The service is invoked by the terminating VMSC to
3955 request the GMSC to resume handling the call and forward it to the specified destination.

3956 This is a confirmed service which uses the Primitives listed in table 10.3/1.

3957 10.3.2 Service primitives

3958 **Table 10.3/1: MAP_RESUME_CALL_HANDLING parameters**

Parameter name	Request	Indication	Response	Confirm
Invoke Id	M	M(=)	M(=)	M(=)
Call Reference Number	C	C(=)		
Basic Service Group	C	C(=)		
IMSI	C	C(=)		
Forwarding Data	C	C(=)		
CUG Interlock	C	C(=)		
CUG Outgoing Access	C	C(=)		
O-CSI	C	C(=)		
D-CSI	C	C(=)		
CCBS Target	C	C(=)		
UU Data	C	C(=)		
UUS CF Interaction	C	C(=)		
All Information Sent	C	C(=)		
MSISDN	C	C(=)		
User error			C	C(=)
Provider error				O

3959

3960 10.3.3 Parameter use

3961 Information received in subsequent segment of a segmented dialogue shall not overwrite information received in an
3962 earlier segment.

3963 See subclause 7.6 for a definition of the parameters used, in addition to the following.

3964 Call Reference Number

3965 See GSM 03.79 [99] for the use of this parameter. This parameter shall be present in a first segment of the dialogue

3966 Basic Service Group

3967 See GSM 03.79 [99] for the use of this parameter. This parameter shall be present in a first segment of the dialogue

3968 IMSI

3969 This is the IMSI of the forwarding Subscriber. This parameter shall be present in a first segment of the dialogue

3970 Forwarding Data

3971 This parameter includes the forwarded-to number, the forwarding reason and the forwarding options Notification to
3972 calling party and Redirecting presentation, and can include the forwarded-to subaddress. See GSM 03.79 [99] for the
3973 conditions for the presence of its components. This parameter shall be present in a first segment of the dialogue

3974 CUG Interlock

3975 See GSM 03.79 [99] for the use of this parameter and the conditions for its presence.

3976 CUG Outgoing Access

3977 See GSM 03.79 [99] for the use of this parameter and the conditions for its presence.

3978 O-CSI

3979 See GSM 03.79 [99] for the use of this parameter and the conditions for its presence.

3980 For CAMEL phases 1 & 2, the O-CSI shall contain only one set of O-BCSM TDP data.

3981 D-CSI

3982 The Dialed Services-CSI.

3983 See 3G TS 23.078 for the use of this parameter and the conditions for its presence.

3984 CCBS Target

3985 See GSM 03.93 [107] for the use of this parameter and the conditions for its presence.

3986 UU Data

3987 See GSM 03.87 for the use of this parameter and the conditions for its presence.

3988 UUS CF Interaction

3989 See GSM 03.87 for the use of this parameter and the conditions for its presence.

3990 All Information Sent

3991 This parameter is set when the VMSC has sent all information to GMSC.

3992 MSISDN

3993 This parameter is the basic MSISDN of the forwarding subscriber. It shall be present if the VMSC supports
3994 determination of the redirecting number.

3995 User error

3996 This parameter is sent by the responder when an error is detected and if present, takes one of the following values:

3997 - Optimal Routeing not allowed;

3998 - Forwarding failed;

3999 - Unexpected Data Value;

4000 - Data Missing.

4001 Provider error

4002 These are defined in subclause 7.6.

4003 **10.4 MAP_PREPARE_GROUP_CALL service**4004 **10.4.1 Definition**

4005 This service is used by the Anchor_MSC to inform the Relay_MSC about a group call setup.

4006 The MAP_PREPARE_GROUP_CALL service is a confirmed service using the service primitives given in table 10.4

4007 **10.4.2 Service primitives**4008 **Table 10.4/1: MAP_PREPARE_GROUP_CALL service**

Parameter name	Request	Indication	Response	Confirm
Invoke Id	M	M(=)	M(=)	M(=)
Teleservice	M	M(=)		
ASCI Call Reference	M	M(=)		
Ciphering Algorithm	M	M(=)		
Group Key Number	C	C(=)		
Group Key	C	C(=)		
Priority	C	C(=)		
CODEC-Information	M	M(=)		
Uplink Free Indicator	M	M(=)		
Group Call Number			M	M(=)
User Error			C	C(=)
Provider Error				O

4009

4010 **10.4.3 Parameter definitions and use**4011 Invoke Id

4012 See definition in section 7.6.1

4013 Teleservice

4014 Voice Broadcast Service or Voice Group Call Service

4015 ASCI Call Reference

4016 Broadcast call reference or group call reference. This item is used to access the VBS-GCR or VGCS-GCR within the
4017 Relay_MSC.

4018 Ciphering Algorithm

4019 The ciphering algorithm to be used for the group call.

4020 Group Key Number

4021 This number has to be broadcasted and is used by the mobile station to select the chosen group key.

4022 Shall be present if the ciphering applies.

4023 Group Key

4024 This key is used for ciphering on the radio interface.

4025 Shall be present if the ciphering applies.

4026 Priority

4027 Default priority level related to the call if eMLPP applies.

4028 CODEC-Information

4029 Information on the codecs allowed for this call.

4030 Uplink Free Indicator

4031 A flag indicating whether the call is initiated from a dispatcher.

4032 Group Call Number

4033 This temporary allocated E.164 number is used for routing the call from the Anchor MSC to the Relay MSC.

4034 User Error4035 For definition of this parameter see section 7.6.1 The following errors defined in section 7.6.1 may be used, depending
4036 on the nature of the fault:

4037 - No Group Call Number available

4038 - System Failure

4039 - Unexpected Data Value.

4040 Provider Error

4041 See definition of provider error in section 7.6.1.

4042 **10.5 MAP_PROCESS_GROUP_CALL_SIGNALLING service**4043 **10.5.1 Definitions**

4044 This service is used between Relay MSC and Anchor MSC for transmission of Group Call notifications.

4045 The MAP_PROCESS_GROUP_CALL_SIGNALLING service is a non-confirmed service using the service primitives
4046 given in table 10.54047 **10.5.2 Service primitives**4048 **Table 10.5/1: MAP_PROCESS_GROUP_CALL_SIGNALLING service**

Parameter name	Request	Indication
Invoke Id	M	M(=)
Uplink Request	C	C(=)
Uplink Release Indication	C	C(=)
Release Group Call	C	C(=)

4049

4050 **10.5.3 Parameter definitions and use**4051 Invoke Id

4052 See definition in section 7.6.1.

4053 Uplink Request4054 This information element indicates to the anchor MSC that a service subscriber roaming in the relay MSC area requests
4055 access to the uplink.4056 Uplink Release Indication

4057 This information element if included by the Relay MSC indicates to the Anchor MSC that the uplink has become free.

4058 Release Group Call

4059 This information element if included by the Relay MSC indicates to the Anchor MSC that the service subscriber who
4060 has initiated the call and who currently has access to the uplink terminates the call.

4061 **10.6 MAP_FORWARD_GROUP_CALL_SIGNALLING service**4062 **10.6.1 Definitions**

4063 This service is used between Anchor MSC and Relay MSC for transmission of Group Call notifications.

4064 The MAP_FORWARD_GROUP_CALL_SIGNALLING service is a non-confirmed service using the service primitives
4065 given in table 10.6.

4066 **10.6.2 Service primitives**4067 **Table 10.6: MAP_FORWARD_GROUP_CALL_SIGNALLING service**

Parameter name	Request	Indication
Invoke Id	M	M(=)
IMSI	C	C(=)
Uplink Request Acknowledgement	C	C(=)
Uplink Release Indication	C	C(=)
Uplink Reject Command	C	C(=)
Uplink Seized Command	C	C(=)
Uplink Release Command	C	C(=)

4068

4069 **10.6.3 Parameter definitions and use**4070 IMSI

4071 Identity of the service subscriber who has established the call and who is allowed to terminate the call.

4072 Invoke Id

4073 See definition in section 7.6.1

4074 Uplink Request Acknowledgement

4075 This information element is used for positive acknowledgement of an uplink request

4076 Uplink Release Indication

4077 This information element if included by the Anchor MSC indicates to the Relay MSC that the uplink has become free.

4078 Uplink Reject Command

4079 This information element is used for negative acknowledgement of an uplink request

4080 Uplink Seized Command

4081 This information element if included by the Anchor MSC indicates to the Relay MSC that the uplink is no longer free.

4082 Uplink Release Command

4083 This information element if included by the Anchor MSC indicates to the Relay MSC that the uplink which is granted to
4084 a MS in the relay MSC area shall be released.

4085 10.7 MAP_SEND_GROUP_CALL_END_SIGNAL service

4086 10.7.1 Definitions

4087 This service is used between the Relay MSC and the Anchor MSC indicating that VGCS / VBS channels have been
 4088 established in the Relay MSC area. The response is used by the Anchor MSC to inform Relay MSC that all resources for
 4089 the call can be released in Relay MSC because the call has been released in the Anchor MSC.

4090 The MAP_SEND_GROUP_CALL_END_SIGNAL service is a confirmed service using the service primitives given in
 4091 table 10.7

4092 10.7.2 Service primitives

4093 **Table 10.7: MAP_SEND_GROUP_CALL_END_SIGNAL service**

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

4094

4095 10.7.3 Parameter definitions and use

4096 IMSI

4097 Identity of the service subscriber who has established the call and who is allowed to terminate the call.

4098 Shall be present if the call was established by a service subscriber roaming in the relay MSC area.

4099 Invoke Id

4100 See definition in section 7.6.1

4101 Provider Error

4102 See definition of provider error in section 7.6.1.

4103 10.8 MAP_Provide_SIWFS_Number

4104 10.8.1 Definition

4105 This service is used between an MSC and SIWFS. It is invoked by an MSC receiving an incoming call (call to or from
 4106 MS) to request the SIWFS to allocate IWU resources. The service is defined in GSM 03.54.

4107 This is a confirmed service using the primitives described in table 10.8.

4108 10.8.2 Service primitive

4109 Table 10.8: MAP_Provide_SIWFS_Number service

Parameter name	Request	Indication	Response	Confirm
Invoke ID	M	M(=)	M(=)	M(=)
GSM Bearer Capability	M	M(=)		
ISDN Bearer Capability	M	M(=)		
Call Direction	M	M(=)		
B-subscriber address	M	M(=)		
Chosen Channel	M	M(=)		
Lower Layer Compatibility	C	C(=)		
High Layer Compatibility	C	C(=)		
SIWFS number			C	C(=)
User error			C	C(=)
Provider error				O

4110

4111 10.8.3 Parameter use

4112 See subclause 7.6 for a definition of the parameter used, in addition to the following.

4113 GSM Bearer Capability

4114 This information is the result from the negotiation with the mobile station. The information is sent from the MSC to the
4115 SIWFS to allocate the correct IWU.

4116 ISDN Bearer Capability

4117 This parameter refers to the ISDN Bearer Capability information element. For the MTC this parameter is received in the
4118 ISUP User Service Information parameter. For the MOC call this parameter is mapped from the GSM BC parameter
4119 according to GSM 09.07. The parameter is used by the SIWFS to route the call and to allocate the outgoing circuit.

4120 Call Direction

4121 This parameter indicates the direction of the call (mobile originated or mobile terminated) at call set-up.

4122 B-subscriber address

4123 This parameter is sent from the MSC to the SIWFS to inform the SIWFS where to route the call i.e. where to send the
4124 IAM. If the loop method is used this parameter will indicate the address to the VMSC. This address is allocated by the
4125 VMSC in the same way as a MSRN and is used to correlate the incoming IAM to the corresponding MAP dialogue. If
4126 the non-loop method is used this parameter will indicate the address to the B-subscriber.

4127 Chosen Channel

4128 This parameter is sent from the MSC to the SIWFS to adjust the interworking unit to the assigned radio resources. This
4129 parameter is defined in GSM 08.08.

4130 Lower Layer Compatibility

4131 This parameter is sent from the MSC to the SIWF to allow the interworking unit to perform a compatibility check. This
4132 parameter is handled as specified in GSM 09.07. This parameter is defined in GSM 04.08.

4133 High Layer Compatibility

4134 This parameter is sent from the MSC to the SIWF to allow the interworking unit to perform a compatibility check. This
4135 parameter is handled as specified in GSM 09.07. This parameter is defined in GSM 04.08.

4136 SIWFS number

4137 This parameter is sent from the SIWFS to the MSC. This address is used by the visited MSC to route the call, i.e. the
4138 IAM to the SIWFS (similar to MSRN) and will be used by the SIWFS to correlate the incoming IAM to the
4139 corresponding MAP message. This parameter must always be sent from the SIWFS when a successful allocation of
4140 SIWFS resources has been made.

4141 User error

4142 This parameter is sent by the responder when an error is detected and if present, takes one of the following values:

- 4143 - Resource limitation;
- 4144 - Facility Not Supported;
- 4145 - Unexpected Data Value;
- 4146 - System Failure.

4147 See subclause 7.6 for a definition of these reasons.

4148 Provider error

4149 These are defined in subclause 7.6.

4150 **10.9 MAP_SIWFS_Signalling_Modify**4151 **10.9.1 Definition**

4152 This service is used to transport signalling information between an MSC and an SIWFS in the case of a request to
 4153 modify the configuration (e.g. HSCSD). It is invoked either by an MSC or by the SIWFS. The service is defined in
 4154 GSM 03.54.

4155 This is a confirmed service using the primitives described in table 10.9.

4156 **10.9.2 Service primitive**4157 **Table 10.9: MAP_SIWFS_Signalling_Modify service**

Parameter name	Request	Indication	Response	Confirm
Invoke ID	M	M(=)	M(=)	M(=)
Channel Type	C	C(=)		
Chosen Channel	C	C(=)	C(=)	C(=)
User error			C	C(=)
Provider error				O

4158

4159 **10.9.3 Parameter use**

4160 See subclause 7.6 for a definition of the parameter used, in addition to the following.

4161 Channel Type

4162 This parameter is the result of a Channel Mode Modification for TS61/62. It contains the changed Air Interface User
 4163 Rate. The information is sent from the SIWFS to the MSC to assign the correct radio resource. This parameter is defined
 4164 in GSM 08.08.

4165 Chosen Channel

4166 This parameter is sent from the MSC to the SIWFS to adjust the interworking unit to the assigned radio resources. This
 4167 parameter is defined in GSM 08.08.

4168 User error

4169 This parameter is sent by the responder when an error is detected and if present, takes one of the following values:

- 4170 - Resource limitation;
- 4171 - Facility Not Supported;

- 4172 - Data Missing;
- 4173 - Unexpected Data Value;
- 4174 - System Failure.
- 4175 See subclause 7.6 for a definition of these reasons.
- 4176 Provider error
- 4177 These are defined in subclause 7.6.

4178 10.10 MAP_SET_REPORTING_STATE service

4179 10.10.1 Definition

- 4180 This service is used between the HLR and the VLR to set the reporting state for a requested service. It is a confirmed
4181 service using the service primitives shown in table 10.10/1.

4182 10.10.2 Service primitives

- 4183 The service primitives are shown in table 10.10/1.

4184 **Table 10.10/1: MAP_SET_REPORTING_STATE parameters**

Parameter name	Request	Indication	Response	Confirm
Invoke id	M	M(=)	M(=)	M(=)
IMSI	C	C(=)		
LMSI	C	C(=)		
CCBS Monitoring	C	C(=)		
CCBS Subscriber Status			C	C(=)
User error			C	C(=)
Provider error				O

4185

4186 10.10.3 Parameter use

- 4187 See subclause 7.6 for a definition of the parameters used, in addition to the following.

4188 IMSI

- 4189 The IMSI is a mandatory parameter if the service is used as the only one in a dialogue.

4190 CCBS Monitoring

- 4191 This parameter indicates whether monitoring for CCBS shall be started or stopped. If it indicates that monitoring shall
4192 be started this service corresponds to the message 'Start Reporting' in GSM 03.93; if it indicates that monitoring shall be
4193 stopped this service corresponds to the message 'Stop Reporting' in GSM 03.93.

4194 CCBS Subscriber Status

- 4195 See GSM 03.93 for the use of this parameter and the conditions for its presence.

4196 User error

- 4197 This parameter is sent by the responder upon unsuccessful outcome of the service, and then takes one of the following
4198 values defined in subclause 7.6.1:

- 4199 - System Failure;
- 4200 - Unidentified Subscriber;

- 4201 - Unexpected Data Value;
 4202 - Data Missing;
 4203 - Resource Limitation;
 4204 - Facility Not Supported.

4205 NOTE: This error is reserved for future use.

4206 Provider error

4207 These are defined in subclause 7.6.

4208 10.11 MAP_STATUS_REPORT service

4209 10.11.1 Definition

4210 This service is used by the VLR to report an event or call outcome to the HLR. It is a confirmed service using the service
 4211 primitives shown in table 10.11/1.

4212 10.11.2 Service primitives

4213 The service primitives are shown in table 10.11/1.

4214

Table 10.11/1: MAP_STATUS_REPORT parameters

Parameter name	Request	Indication	Response	Confirm
Invoke id	M	M(=)	M(=)	M(=)
IMSI	M	M(=)		
CCBS Subscriber Status	C	C(=)		
Monitoring Mode	C	C(=)		
Call Outcome	C	C(=)		
User error			C	C(=)
Provider error				O

4215

4216 10.11.3 Parameter use

4217 See subclause 7.6 for a definition of the parameters used, in addition to the following.

4218 CCBS Subscriber Status

4219 If this parameter is present without Monitoring Mode and Call Outcome this service corresponds to the message 'Event
 4220 Report' in GSM 03.93 [107]. See GSM 03.93 [107] for the use of this parameter and the conditions for its presence.

4221 Monitoring Mode

4222 If this parameter is present with CCBS Call Outcome this service corresponds to the message 'CCBS Call Report' in
 4223 GSM 03.93. See GSM 03.93 for the use of this parameter and the conditions for its presence.

4224 Call Outcome

4225 See GSM 03.93 for the use of this parameter and the conditions for its presence.

4226 User error

4227 This parameter is sent by the responder upon unsuccessful outcome of the service, and then takes one of the following
 4228 values defined in subclause 7.6.1:

- 4229 - Unknown Subscriber ;

- 4230 - System Failure;
 4231 - Unexpected Data Value;
 4232 - Data Missing.

4233 Provider error

4234 These are defined in subclause 7.6.

4235 10.12 MAP_REMOTE_USER_FREE service

4236 10.12.1 Definition

4237 This service is used between the HLR and the VLR to report that the B subscriber is now idle and that the A subscriber
 4238 can be notified. It is a confirmed service using the service primitives shown in table 10.12/1.

4239 10.12.2 Service primitives

4240 The service primitives are shown in table 10.12/1.

4241 **Table 10.12/1: MAP_REMOTE_USER_FREE parameters**

Parameter name	Request	Indication	Response	Confirm
Invoke id	M	M(=)	M(=)	M(=)
IMSI	M	M(=)		
Call Info	M	M(=)		
CCBS Feature	M	M(=)		
Translated B Number	M	M(=)		
Replace B Number	C	C(=)		
Alerting Pattern	C	C(=)		
RUF Outcome			C	C(=)
User error			C	C(=)
Provider error				O

4242

4243 10.12.3 Parameter use

4244 See subclause 7.6 for a definition of the parameters used, in addition to the following.

4245 Call Info

4246 See GSM 03.93 for the use of this parameter.

4247 CCBS Feature

4248 See GSM 03.93 for the conditions for the presence of the parameters included in the CCBS feature.

4249 Translated B Number

4250 See GSM 03.93 for the use of this parameter.

4251 Replace B Number

4252 See GSM 03.93 for the use of this parameter and the conditions for its presence.

4253 Alerting Pattern

4254 See GSM 03.93 for the use of this parameter and the conditions for its presence.

4255 RUF Outcome

4256 See GSM 03.93 for the use of this parameter and the conditions for its presence.

4257 User error

4258 This parameter is sent by the responder upon unsuccessful outcome of the service, and then takes one of the following
4259 values defined in subclause 7.6.1:

4260 - Unexpected Data Value;

4261 - Data Missing;

4262 - Incompatible Terminal;

4263 This error is returned by the responder when the terminal used for CCBS activation is not compatible with the
4264 terminal used for the CCBS recall. For details refer to GSM 04.08.

4265 - Absent Subscriber (IMSI Detach; Restricted Area; No Page Response);

4266 - System Failure;

4267 - Busy Subscriber (CCBS Busy).

4268 Provider error

4269 These are defined in subclause 7.6.

4270 **10.13 MAP_IST_ALERT service**4271 **10.13.1 Definition**

4272 This service is used between the MSC (Visited MSC or Gateway MSC) and the HLR, to report that the IST timer
4273 running for a call for the Subscriber has expired. It is a confirmed service using the service primitives shown in table
4274 10.13/1.

4275 **10.13.2 Service primitives**

4276 The service primitives are shown in table 10.13/1.

4277 **Table 10.13/1: MAP_IST_ALERT parameters**

Parameter name	Request	Indication	Response	Confirm
Invoke Id	M	M(=)	M(=)	M(=)
IMSI	M	M(=)		
IST Alert Timer			C	C(=)
IST Information Withdraw			C	C(=)
Call termination Indicator			C	C(=)
User error			C	C(=)
Provider error				O

4278

4279 **10.13.3 Parameter use**

4280 All parameters are described in subclause 7.6. The following clarifications are applicable:

4281 IST Alert Timer

4282 If included in the IST Alert response, it includes the new IST Alert timer value that must be used to inform the HLR
4283 about the call activities that the subscriber performs.

4284 IST Information Withdraw

4285 If included in the IST Alert response, this parameter is used to indicate that the IST condition has been removed for the
4286 subscriber. When the MSC receives this parameter, IST control for that call shall be terminated.

4287 Call termination Indicator

4288 If included in the IST Alert response, this parameter is used to indicate whether the MSC shall terminate the call activity
4289 that had previously triggered the IST Alert procedure, or it shall also release all other call activities for the specified
4290 subscriber (outgoing call activities if the IST Alert is initiated by the VMSC, or incoming call activities if the IST Alert
4291 is initiated by the GMSC). Release of all other call activities is possible only if the MSC has the capability to link the
4292 call activities for the Subscriber by using the IMSI as key.

4293 User error

4294 This parameter is sent by the responder when an error is detected and if present, takes one of the following values:

- 4295 - System Failure;
- 4296 - Unexpected Data Value;
- 4297 - Resource Limitation;
- 4298 - Facility Not Supported;
- 4299 - Unknown Subscriber.

4300 **10.14 MAP_IST_COMMAND service**4301 **10.14.1 Definition**

4302 This service is used by the HLR to instruct the MSC (Visited MSC or Gateway MSC) to terminate ongoing call
4303 activities for a specific subscriber. It is a confirmed service using the service primitives shown in table 10.14/1.

4304 **10.14.2 Service primitives**

4305 The service primitives are shown in table 10.14/1.

4306 **Table 10.14/1: MAP_IST_COMMAND parameters**

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

4307

4308 **10.14.3 Parameter use**

4309 All parameters are described in subclause 7.6. The following clarifications are applicable:

4310 User error

4311 This parameter is sent by the responder when an error is detected and if present, takes one of the following values:

- 4312 - System Failure;
- 4313 - Unexpected Data Value;
- 4314 - Resource Limitation;
- 4315 - Facility Not Supported;
- 4316 - Unknown Subscriber.

4317 **11 Supplementary services related services**4318 **11.1 MAP_REGISTER_SS service**4319 **11.1.1 Definition**4320 This service is used between the MSC and the VLR and between the VLR and the HLR to register data related to a
4321 supplementary service. The VLR will relay the message to the HLR.

4322 The service is a confirmed service and consists of four service primitives.

4323 **11.1.2 Service primitives**

4324 The service primitives are shown in table 11.1/1.

4325 **Table 11.1/1: MAP_REGISTER_SS parameters**

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(=)		
EMLPP default priority	C	C(=)	C	C(=)
Forwarding information			C	C(=)
User error			C	C(=)
Provider error				O

4326

4327 **11.1.3 Parameter use**4328 Invoke id

4329 See subclause 7.6.1 for the use of this parameter.

4330 SS-Code

4331 This parameter indicates the supplementary service which the mobile subscriber wants to register.

4332 Basic service

4333 This parameter indicates for which basic service group the supplementary service is to be registered. If it is not included,
4334 the registration request applies to all basic services.

4335 Forwarded-to number with subaddress

4336 This parameter is obligatory if the registration applies to one or more call forwarding supplementary services. It can
4337 optionally include a sub-address.

4338 No reply condition time

4339 This parameter is included if the registration applies to the Call Forwarding on No Reply supplementary service (or a
4340 superset of this service) and the mobile subscriber supplies a value for this time.

4341 EMLPP default priority

4342 This parameter is sent by the initiator to register the eMLPP default priority level and is returned by the responder at
4343 successful outcome of the service.

4344 Forwarding information

4345 This parameter is returned by the responder at successful outcome of the service, if the registration request concerned
4346 one or a group of Call Forwarding supplementary services.

4347 User error

4348 This parameter is sent by the responder upon unsuccessful outcome of the service, and then takes one of the following
4349 values defined in subclause 7.6.1:

4350 - System failure;

4351 - Data missing;

4352 - Unexpected data value;

4353 - Call Barred;

4354 - Bearer service not provisioned;

4355 This error is returned only if not even a subset of the requested bearer service group has been subscribed to.

4356 - Teleservice not provisioned;

4357 This error is returned only if not even a subset of the requested teleservice group has been subscribed to.

4358 - Illegal SS operation;

4359 - SS error status;

4360 - SS incompatibility.

4361 Provider error

4362 See subclause 7.6.1 for the use of this parameter.

4363 **11.2 MAP_ERASE_SS service**4364 **11.2.1 Definition**

4365 This service is used between the MSC and the VLR and between the VLR and the HLR to erase data related to a
4366 supplementary service. The VLR will relay the message to the HLR.

4367 The service is a confirmed service and consists of four service primitives.

4368 **11.2.2 Service primitives**

4369 The service primitives are shown in table 11.2/1.

4370

Table 11.2/1: MAP_ERASE_SS parameters

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

4371

4372 **11.2.3 Parameter use**4373 Invoke id

4374 See subclause 7.6.1 for the use of this parameter.

4375 SS-Code

4376 This parameter indicates the supplementary service which the mobile subscriber wants to erase.

4377 Basic service4378 This parameter indicates for which basic service group the supplementary service should be erased. If it is not included,
4379 the erasure request applies to all basic services.4380 Forwarding information4381 This parameter is returned by the responder at successful outcome of the service, if the erasure request concerned one or
4382 a group of Call Forwarding supplementary services.4383 User error4384 This parameter is sent by the responder upon unsuccessful outcome of the service, and then takes one of the following
4385 values, defined in subclause 7.6.1:

4386 - System failure;

4387 - Data Missing;

4388 - Unexpected data value;

4389 - Bearer service not provisioned;

4390 This error is returned only if not even a subset of the requested bearer service group has been subscribed to.

4391 - Teleservice not provisioned;

4392 This error is returned only if not even a subset of the requested teleservice group has been subscribed to.

4393 - Call Barred;

4394 - Illegal SS operation;

4395 - SS error status.

4396 Provider error

4397 See subclause 7.6.1 for the use of this parameter.

4398 **11.3 MAP_ACTIVATE_SS service**4399 **11.3.1 Definition**

4400 This service is used between the MSC and the VLR and between the VLR and the HLR to activate a supplementary
4401 service. The VLR will relay the message to the HLR.

4402 The service is a confirmed service and consists of four service primitives.

4403 **11.3.2 Service primitives**

4404 The service primitives are shown in table 11.3/1.

4405 **Table 11.3/1: MAP_ACTIVATE_SS parameters**

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

4406

4407 **11.3.3 Parameter use**4408 Invoke id

4409 See subclause 7.6.1 for the use of this parameter.

4410 SS-Code

4411 This parameter indicates the supplementary service which the mobile subscriber wants to activate.

4412 Basic service

4413 This parameter indicates for which basic service groups the requested supplementary service(s) should be activated. If it
4414 is not included, the activation request applies to all basic services.

4415 Forwarding information

4416 This parameter is returned by the responder at successful outcome of the service, if the activation request concerned Call
4417 Forwarding.

4418 Call barring information

4419 This parameter is returned by the responder at successful outcome of the service, if the activation request concerned Call
4420 Barring.

4421 SS-Data

4422 This parameter is returned by the responder at successful outcome of the service, if the activation request concerned for
4423 example Call Waiting.

4424 User error

4425 This parameter is sent by the responder upon unsuccessful outcome of the service, and then takes one of the following
4426 values, defined in subclause 7.6.1:

4427 - System failure;

4428 - Data Missing;

4429 - Unexpected data value;

4430 - Bearer service not provisioned;

4431 This error is returned only if not even a subset of the requested bearer service group has been subscribed to.

4432 - Teleservice not provisioned;

4433 This error is returned only if not even a subset of the requested teleservice group has been subscribed to.

4434 - Call Barred;

4435 - Illegal SS operation;

4436 - SS error status;

4437 - SS subscription violation;

4438 - SS incompatibility;

4439 - Negative PW check;

4440 - Number Of PW Attempts Violation.

4441 Provider error

4442 See subclause 7.6.1 for the use of this parameter.

4443 **11.4 MAP_DEACTIVATE_SS service**4444 **11.4.1 Definitions**

4445 This service is used between the MSC and the VLR and between the VLR and the HLR to deactivate a supplementary
4446 service. The VLR will relay the message to the HLR.

4447 The service is a confirmed service and consists of four service primitives.

4448 **11.4.2 Service primitives**

4449 The service primitives are shown in table 11.4/1.

4450 **Table 11.4/1: MAP_DEACTIVATE_SS parameters**

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

4451

4452 11.4.3 Parameter use

4453 Invoke id

4454 See subclause 7.6.1 for the use of this parameter.

4455 SS-Code

4456 This parameter indicates the supplementary service which the mobile subscriber wants to deactivate.

4457 Basic service

4458 This parameter indicates for which basic service group the requested supplementary service(s) should be deactivated. If
4459 it is not included the deactivation request applies to all basic services.

4460 Forwarding information

4461 This parameter is returned by the responder at successful outcome of the service, if the deactivation request concerned
4462 one or a group of Call Forwarding supplementary services.

4463 Call barring information

4464 This parameter is returned by the responder at successful outcome of the service, if the activation request concerned one
4465 or a group of Call Barring supplementary services.

4466 SS-Data

4467 This parameter is returned by the responder at successful outcome of the service, for example if the deactivation request
4468 concerned the Call Waiting supplementary service.

4469 User error

4470 This parameter is sent by the responder upon unsuccessful outcome of the service, and then takes one of the following
4471 values, defined in subclause 7.6.1:

4472 - System failure;

4473 - Data Missing;

4474 - Unexpected data value;

4475 - Bearer service not provisioned;

4476 This error is returned only if not even a subset of the requested bearer service group has been subscribed to.

4477 - Teleservice not provisioned;

4478 This error is returned only if not even a subset of the requested teleservice group has been subscribed to.

4479 - Call Barred;

4480 - Illegal SS operation;

4481 - SS error status;

4482 - SS subscription violation;

4483 - Negative PW check;

4484 - Number Of PW Attempts Violation.

4485 Provider error

4486 See subclause 7.6.1 for the use of this parameter.

4487 11.5 MAP_INTERROGATE_SS service

4488 11.5.1 Definitions

4489 This service is used between the MSC and the VLR and between the VLR and the HLR to retrieve information related to
4490 a supplementary service. The VLR will relay the message to the HLR if necessary.

4491 The service is a confirmed service and consists of four service primitives.

4492 11.5.2 Service primitives

4493 The service primitives are shown in table 11.5/1.

4494 **Table 11.5/1: MAP_INTERROGATE_SS parameters**

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(=)
EMLPP Info			C	C(=)
CCBS Feature LIST			C	C(=)
User error			C	C(=)
Provider error				O

4495

4496 11.5.3 Parameter use

4497 For additional information on parameter use refer to the GSM 04.8x and 04.9x-series of technical specifications.

4498 Invoke id

4499 See subclause 7.6.1 for the use of this parameter.

4500 SS-Code

4501 The mobile subscriber can only interrogate a single supplementary service per service request.

4502 Basic service

4503 This parameter indicates for which basic service group the given supplementary service is interrogated. If it is not
4504 included, the interrogation request applies to all basic services.

4505 SS-Status

4506 This parameter is included by the responder if:

- 4507 - the interrogated supplementary service can only be subscribed for all applicable basic services simultaneously; or
- 4508 - the interrogated supplementary service is not active for any of the interrogated basic services, or
- 4509 - the interrogation was for the CCBS supplementary service and no CCBS request is active or the service is not
4510 provisioned.

4511 Basic service group LIST

4512 This parameter LIST is used to include one or a series of basic service groups for which the interrogated supplementary
4513 service is active. If the interrogated supplementary service is not active for any of the interrogated (and provisioned)
4514 basic service groups, the SS-Status parameter is returned.

4515 Forwarding feature LIST

4516 The forwarding feature parameter is described in subclause 7.6.4. A list of one or more forwarding features is returned
4517 by the responder when the interrogation request applied to Call Forwarding supplementary service.

4518 If no basic service code parameter is provided within this sequence, the forwarding feature parameter applies to all
4519 provisioned basic services.

4520 CLI restriction Info

4521 The CLI-RestrictionInfo parameter is returned by the responder when the interrogation request applies to the CLIR
4522 supplementary service.

4523 EMLPP Info

4524 The eMLPP info (maximum entitled priority and default priority) is returned by the responder if the interrogation
4525 request applies to the eMLPP supplementary service.

4526 CCBS Feature LIST

4527 The CCBS feature parameter is described in subclause 7.6. A list of one or more CCBS features is returned by the
4528 responder when the interrogation request applied to the CCBS supplementary service. See GSM 03.93 [107] for the
4529 conditions for the presence of the parameters included in the CCBS feature.

4530 User error

4531 This error is sent by the responder upon unsuccessful outcome of the interrogation service, and then takes one of the
4532 following values, defined in subclause 7.6.1:

4533 - System failure;

4534 - Data Missing;

4535 - Unexpected data value;

4536 - Bearer Service not provisioned;

4537 This error is returned only if not even a subset of the interrogated bearer services are provided.

4538 - Teleservice not provisioned;

4539 This error is returned only if not even a subset of the interrogated teleservices are provided.

4540 - Call Barred;

4541 - Illegal SS operation;

4542 - SS not available.

4543 Provider error

4544 See subclause 7.6.1 for the use of this parameter.

4545 11.6 MAP_INVOKE_SS service

4546 11.6.1 Definitions

4547 This service is used between the MSC and the VLR to check the subscriber's subscription to a given supplementary
4548 service in the VLR, in connection with in-call invocation of that supplementary service, i.e. after the call set-up phase is
4549 finished. For supplementary service invocation during call set-up phase, please refer to the call handling descriptions.

4550 The service is a confirmed service and consists of four service primitives.

4551 11.6.2 Service primitives

4552 The service primitives are shown in table 11.6/1.

4553 **Table 11.6/1: MAP_INVOKE_SS parameters**

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

4554

4555 11.6.3 Parameter use

4556 Invoke id

4557 See subclause 7.6.1 for the use of this parameter.

4558 SS-Code

4559 This SS-Code can only refer to a single supplementary service, e.g. the Call Hold or Multi Party supplementary services.

4560 Basic service

4561 This parameter indicates for which basic service the supplementary service invocation is required.

4562 User error

4563 This parameter is sent by the responder upon unsuccessful outcome of the service, and then takes one of the following values:

- 4565 - System Failure;
- 4566 - Data Missing;
- 4567 - Unexpected data value;
- 4568 - Call Barred;
- 4569 - Illegal SS operation;
- 4570 - SS error status;
- 4571 - SS not available.

4572 Provider error

4573 See subclause 7.6.1 for the use of this parameter.

4574 11.7 MAP_REGISTER_PASSWORD service

4575 11.7.1 Definitions

4576 This service is used between the MSC and the VLR and between the VLR and the HLR if the mobile subscriber requests
4577 to register a new password. The VLR will relay the message to the HLR.

4578 The service is a confirmed service and consists of four service primitives.

4579 **11.7.2 Service primitives**

4580 The service primitives are shown in table 11.7/1.

4581 **Table 11.7/1: MAP_REGISTER_PASSWORD parameters**

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

4582

4583 **11.7.3 Parameter use**4584 Invoke id

4585 See subclause 7.6.1 for the use of this parameter.

4586 SS-Code

4587 This parameter indicates for which supplementary service(s) the password should be registered.

4588 New Password

4589 See subclause 7.6.4 for the use of this parameter.

4590 User error4591 This parameter is sent by the responder upon unsuccessful outcome of the service, and then takes one of the following
4592 values, defined in subclause 7.6.1:

- 4593 - System failure;
- 4594 - Data Missing;
- 4595 - Unexpected data value;
- 4596 - Call Barred;
- 4597 - SS subscription violation;
- 4598 - Password registration failure;
- 4599 - Negative PW check;
- 4600 - Number Of PW Attempts Violation.

4601 Provider error

4602 See subclause 7.6.1 for the use of this parameter.

4603 **11.8 MAP_GET_PASSWORD service**4604 **11.8.1 Definitions**

4605 This service is used between the HLR and the VLR and between the VLR and the MSC when the HLR receives a
4606 request from the mobile subscriber for an operation on a supplementary service which requires a password from the
4607 subscriber. The VLR will relay the message to the MSC.

4608 The service is a confirmed service and consists of four service primitives.

4609 **11.8.2 Service primitives**

4610 The service primitives are shown in table 11.8/1.

4611 **Table 11.8/1: MAP_GET_PASSWORD parameters**

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

4612

4613 **11.8.3 Parameter use**4614 Invoke id

4615 See subclause 7.6.1 for the use of this parameter.

4616 Linked Id4617 See subclause 7.6.1 for the use of this parameter. If the MAP GET PASSWORD service is used in conjunction with the
4618 MAP REGISTER PASSWORD service, this parameter must be present; otherwise it must be absent.4619 Guidance info

4620 See subclause 7.6.4 for the use of this parameter.

4621 Current password

4622 See subclause 7.6.4 for the use of this parameter.

4623 Provider error

4624 See subclause 7.6.1 for the use of this parameter.

4625 **11.9 MAP_PROCESS_UNSTRUCTURED_SS_REQUEST**
4626 **service**4627 **11.9.1 Definitions**4628 This service is used between the MSC and the VLR, between the VLR and the HLR, between the HLR and gsmSCF and
4629 between the HLR and HLR to relay information in order to allow unstructured supplementary service operation.4630 The MAP_PROCESS_UNSTRUCTURED_SS_REQUEST service is a confirmed service using the primitives from
4631 table 11.9/1.4632 **11.9.2 Service primitives**4633 **Table 11.9/1: MAP_PROCESS_UNSTRUCTURED_SS_REQUEST parameters**

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(=)
MSISDN	U	C(=)		
User error			C	C(=)
Provider error				O

4634

4635 11.9.3 Parameter use

4636 Invoke id

4637 See subclause 7.6.1 for the use of this parameter.

4638 USSD Data Coding Scheme:

4639 See subclause 7.6.4 for the use of this parameter. The presence of the parameter in the response is dependent on the
4640 unstructured supplementary service application. If this parameter is present, then the USSD String parameter has to be
4641 present.

4642 USSD String:

4643 See subclause 7.6.1 for the use of this parameter. The presence of the parameter in the response is dependent on the
4644 unstructured supplementary service application. If this parameter is present, then the USSD Data Coding Scheme
4645 parameter has to be present.

4646 MSISDN:

4647 The subscriber's basic MSISDN.

4648 See definition in subclause 7.6.2. The MSISDN is included as an operator option, e.g. to allow addressing the
4649 subscriber's data in the gsmSCF with the MSISDN.

4650 User error

4651 This parameter is sent by the responder upon unsuccessful outcome of the service, and then takes one of the following
4652 values defined in subclause 7.6.1:

4653 - System failure;

4654 - Data missing;

4655 - Unexpected data value;

4656 This error is returned by the responder if it is not able to deal with the contents of the USSD string.

4657 - Call Barred;

4658 - Unknown Alphabet.

4659 Provider error

4660 See subclause 7.6.1 for the use of this parameter.

4661 11.10 MAP_UNSTRUCTURED_SS_REQUEST service

4662 11.10.1 Definitions

4663 This service is used between the gsmSCF and the HLR, the HLR and the VLR and between the VLR and the MSC when
4664 the invoking entity requires information from the mobile user, in connection with unstructured supplementary service
4665 handling.

4666 The MAP_UNSTRUCTURED_SS_REQUEST service is a confirmed service using the primitives from table 11.10/1.

4667 **11.10.2 Service primitives**

4668 The service primitives are shown in table 11.10/1.

4669 **Table 11.10/1: MAP_UNSTRUCTURED_SS_REQUEST parameters**

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(=)
Alerting Pattern	C	C(=)		
User error			C	C(=)
Provider error				O

4670

4671 **11.10.3 Parameter use**4672 Invoke id

4673 See subclause 7.6.1 for the use of this parameter.

4674 USSD Data Coding Scheme:4675 See subclause 7.6.4 for the use of this parameter. The presence of the parameter in the response is dependent on the
4676 mobile user's MMI input. If this parameter is present, then the USSD String parameter has to be present.4677 USSD String:4678 See subclause 7.6.1 for the use of this parameter. The presence of the parameter in the response is dependent on the
4679 mobile user's MMI input. If this parameter is present, then the USSD Data Coding Scheme parameter has to be present.4680 Alerting Pattern

4681 See subclause 7.6.3 for the use of this parameter.

4682 User error4683 This parameter is sent by the responder upon unsuccessful outcome of the service, and then takes one of the following
4684 values defined in subclause 7.6.1:

4685 - System failure;

4686 - Data missing;

4687 - Unexpected data value;

4688 This error is returned by the responder if it is not able to deal with the contents of the USSD string.

4689 - Absent Subscriber;

4690 - Illegal Subscriber;

4691 This error indicates that delivery of the unstructured supplementary service data failed because the MS failed
4692 authentication.

4693 - Illegal Equipment;

4694 - USSD Busy;

4695 - Unknown Alphabet.

4696 Provider error

4697 See subclause 7.6.1 for the use of this parameter.

4698 **11.11 MAP_UNSTRUCTURED_SS_NOTIFY service**4699 **11.11.1 Definitions**

4700 This service is used between the gsmSCF and the HLR, the HLR and the VLR and between the VLR and the MSC when
 4701 the invoking entity requires a notification to be sent to the mobile user, in connection with unstructured supplementary
 4702 services handling.

4703 The MAP_UNSTRUCTURED_SS_NOTIFY service is a confirmed service using the primitives from table 11.11/1.

4704 **11.11.2 Service primitives**

4705 The service primitives are shown in table 11.11/1.

4706 **Table 11.11/1: MAP_UNSTRUCTURED_SS_NOTIFY parameters**

Parameter name	Request	Indication	Response	Confirm
Invoke id	M	M(=)	M(=)	M(=)
USSD Data Coding Scheme	M	M(=)		
USSD String	M	M(=)		
Alerting Pattern	C	C(=)		
User error			C	C(=)
Provider error				O

4707

4708 **11.11.3 Parameter use**4709 Invoke id

4710 See subclause 7.6.1 for the use of this parameter.

4711 USSD Data Coding Scheme:

4712 See subclause 7.6.4 for the use of this parameter.

4713 USSD String:

4714 See subclause 7.6.1 for the use of this parameter.

4715 Alerting Pattern

4716 See subclause 7.6.3 for the use of this parameter.

4717 User error

4718 This parameter is sent by the responder upon unsuccessful outcome of the service, and then takes one of the following
 4719 values defined in subclause 7.6.1:

4720 - System failure;

4721 - Data missing;

4722 - Unexpected data value;

4723 This error is returned by the responder if it is not able to deal with the contents of the USSD string.

4724 - Absent Subscriber;

4725 - Illegal Subscriber;

4726 This error indicates that delivery of the unstructured supplementary service data failed because the MS failed
 4727 authentication.

- 4728 - Illegal Equipment;
 4729 - USSD Busy;
 4730 - Unknown Alphabet.

4731 Provider error

4732 See subclause 7.6.1 for the use of this parameter.

4733 11.12 MAP_SS_INVOCATION_NOTIFY

4734 11.12.1 Definition

4735 This service is used between the MSC and the gsmSCF when the subscriber invokes one of the following supplementary
 4736 services; Call Deflection (CD), Explicit Call Transfer (ECT) or Multi Party (MPTY).

4737 This service is used between the HLR and the gsmSCF when the subscriber invokes the CCBS supplementary service.

4738 11.12.2 Service primitives

4739 The service primitives are shown in table 11.12/1.

4740 **Table 11.12/1: SS_INVOCATION_NOTIFY parameters**

Parameter name	Request	Indication	Response	Confirm
Invoke id	M	M(=)	M(=)	M(=)
MSISDN	M	M(=)		
IMSI	M	M(=)		
SS- event	M	M(=)		
SS- event data	C	C(=)		
User error			C	C(=)
Provider error				O

4741

4742 11.12.3 Parameter use

4743 All parameters are described in subclause 7.6. The use of these parameters and the requirements for their presence are
 4744 specified in 3G TS 23.078.

4745 User error

4746 This parameter is sent by the responder when an error is detected and if present, takes one of the following values:

- 4747 - Data Missing;
 4748 - Unexpected Data Value;
 4749 - Unknown Subscriber.

4750 Provider error

4751 This is defined in subclause 7.6.1.

4752 11.13 MAP_REGISTER_CC_ENTRY service

4753 11.13.1 Definition

4754 This service is used between the MSC and the VLR and between the VLR and the HLR to register data for a requested
 4755 call completion supplementary service. The VLR will relay the message to the HLR.

4756 The service is a confirmed service and uses the service primitives shown in table 11.13/1.

4757 11.13.2 Service primitives

4758 The service primitives are shown in table 11.13/1.

4759 **Table 11.13/1: MAP_REGISTER_CC_ENTRY parameters**

Parameter name	Request	Indication	Response	Confirm
Invoke id	M	M(=)	M(=)	M(=)
SS Code	M	M(=)		
CCBS Feature	C	C(=)	C	C(=)
Translated B number	C	C(=)		
Service Indicator	C	C(=)		
Call Info	C	C(=)		
Network Signal Info	C	C(=)		
User error			C	C(=)
Provider error				O

4760

4761 11.13.3 Parameter use

4762 See subclause 7.6 for a definition of the parameters used, in addition to the following.

4763 SS-Code

4764 This parameter indicates the call completion supplementary service for which the mobile subscriber wants to register an
4765 entry.

4766 CCBS Feature

4767 See GSM 03.93 for the conditions for the presence of the parameters included in the CCBS feature.

4768 Translated B Number

4769 See GSM 03.93 for the use of this parameter and the conditions for its presence.

4770 Service Indicator

4771 This parameter corresponds to the parameters 'Presentation Indicator' and 'CAMEL Invoked' in GSM 03.93 [107]. It
4772 indicates which services have been invoked for the original call (e.g. CLIR, CAMEL). See GSM 03.93 [107] for the use
4773 of this parameter and the conditions for its presence.

4774 Call Info

4775 See GSM 03.93 [107] for the use of this parameter and the conditions for its presence.

4776 Network Signal Info

4777 See GSM 03.93 [107] for the use of this parameter and the conditions for its presence.

4778 User error

4779 This parameter is sent by the responder upon unsuccessful outcome of the service, and then takes one of the following
4780 values, defined in subclause 7.6.1:

- 4781 - System failure;
- 4782 - Data missing;
- 4783 - Unexpected data value;
- 4784 - Call Barred;
- 4785 - Illegal SS operation;

- 4786 - SS error status;
 4787 - SS incompatibility.
 4788 - Short Term Denial;
 4789 - Long Term Denial;

4790 - Facility Not Supported;

4791 Note: This error is reserved for future use.

4792 Private Extensions shall not be sent with these user errors for this operation.

4793 Provider error

4794 See subclause 7.6.1 for the use of this parameter.

4795 11.14 MAP_ERASE_CC_ENTRY service

4796 11.14.1 Definition

4797 This service is used between the MSC and the VLR and between the VLR and the HLR to erase data related to a call completion supplementary service. The VLR will relay the message to the HLR.

4799 The service is a confirmed service and uses the service primitives shown in table 11.14/1.

4800 11.14.2 Service primitives

4801 The service primitives are shown in table 11.14/1.

4802 **Table 11.14/1: MAP_ERASE_CC_ENTRY parameters**

Parameter name	Request	Indication	Response	Confirm
Invoke id	M	M(=)	M(=)	M(=)
SS-Code	M	M(=)	C(=)	C(=)
CCBS Index	C	C(=)		
SS-Status			C	C(=)
User error			C	C(=)
Provider error				O

4803

4804 11.14.3 Parameter use

4805 See subclause 7.6 for a definition of the parameters used, in addition to the following.

4806 SS-Code

4807 This parameter indicates the call completion supplementary service for which the mobile subscriber wants to erase an entry/entries.

4809 CCBS Index

4810 See GSM 03.93 for the use of this parameter and the condition for its presence.

4811 SS-Status

4812 Depending on the outcome of the service request this parameter may indicate either provisioned and active or not provisioned.

4813

4814 User error

4815 This parameter is sent by the responder upon unsuccessful outcome of the service, and then takes one of the following
4816 values, defined in subclause 7.6.1:

- 4817 - System failure;
- 4818 - Data Missing;
- 4819 - Unexpected data value;
- 4820 - Call Barred;
- 4821 - Illegal SS operation;
- 4822 - SS error status.

4823 Private Extensions shall not be sent with these user errors for this operation.

4824 Provider error

4825 See subclause 7.6.1 for the use of this parameter.

4826 **12 Short message service management services**4827 **12.1 MAP-SEND-ROUTING-INFO-FOR-SM service**4828 **12.1.1 Definition**

4829 This service is used between the gateway MSC and the HLR to retrieve the routing information needed for routing the
4830 short message to the servicing MSC.

4831 The MAP-SEND-ROUTING-INFO-FOR-SM is a confirmed service using the primitives from table 12.1/1.

4832 **12.1.2 Service primitives**

4833 The service primitives are shown in table 12.1/1.

4834

Table 12.1/1: MAP-SEND-ROUTING-INFO-FOR-SM

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(=)		
SM-RP-MTI	C	C(=)		
SM-RP-SMEA	C	C(=)		
GPRS Support Indicator	C	C(=)		
IMSI			C	C(=)
Network Node Number			C	C(=)
LMSI			C	C(=)
GPRS Node Indicator			C	C(=)
Additional Number			C	C(=)
User error			C	C(=)
Provider error				O

4835

4836 12.1.3 Parameter use

4837 Invoke id:

4838 See definition in subclause 7.6.1.

4839 MSISDN:

4840 See definition in subclause 7.6.2.

4841 SM-RP-PRI:

4842 See definition in subclause 7.6.8.

4843 Service Centre Address:

4844 See definition in subclause 7.6.2.

4845 SM-RP-MTI:

4846 See definition in subclause 7.6.8. This parameter shall be present when the feature « SM filtering by the HPLMN » is
4847 supported by the SMS-GMSC and when the equivalent parameter is received from the short message service relay sub-
4848 layer protocol.

4849 SM-RP-SMEA:

4850 See definition in subclause 7.6.8. This parameter shall be present when the feature « SM filtering by the HPLMN » is
4851 supported by the SMS-GMSC and when the equivalent parameter is received from the short message service relay sub-
4852 layer protocol.

4853 GPRS Support Indicator:

4854 See definition in subclause 7.6.8. The presence of this parameter is mandatory if the SMS-GMSC supports receiving of
4855 the two numbers from the HLR.

4856 IMSI:

4857 See definition in subclause 7.6.2. The presence of this parameter is mandatory in a successful case.

4858 Network Node Number:

4859 See definition in subclause 7.6.2. This parameter is provided in a successful response.

4860 LMSI:

4861 See definition in subclause 7.6.2. It is an operator option to provide this parameter from the VLR; it is mandatory for the
4862 HLR to include the LMSI in a successful response, if the VLR has used the LMSI.

4863 GPRS Node Indicator:

4864 See definition in subclause 7.6.8. The presence of this parameter is mandatory if only the SGSN number is sent in the
4865 Network Node Number.

4866 Additional Number:

4867 See definition in subclause 7.6.2. This parameter is provided in a successful response.

4868 User error:

4869 The following errors defined in subclause 7.6.1 may be used, depending on the nature of the fault:

- 4870 - Unknown subscriber;
- 4871 - Call Barred;
- 4872 - Teleservice Not Provisioned;

- 4873 - Absent Subscriber_SM;
 4874 - Facility Not Supported;
 4875 - System failure;
 4876 - Unexpected Data Value;
 4877 - Data missing.

4878 **Provider error:**

4879 For definition of provider errors see subclause 7.6.1.

4880 12.2 MAP-MO-FORWARD-SHORT-MESSAGE service

4881 12.2.1 Definition

4882 This service is used between the serving MSC or the SGSN and the gateway MSC to forward mobile originated short
 4883 messages.

4884 The MAP-MO-FORWARD-SHORT-MESSAGE service is a confirmed service using the service primitives given in
 4885 table 12.2/1.

4886 12.2.2 Service primitives

4887 The service primitives are shown in table 12.2/1.

4888

Table 12.2/1: MAP-MO-FORWARD-SHORT-MESSAGE

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(=)	C	C(=)
IMSI	C	C(=)		
User error			C	C(=)
Provider error				O

4889

4890 12.2.3 Parameter use

4891 **Invoke id:**

4892 See definition in subclause 7.6.1.

4893 **SM RP DA:**

4894 See definition in subclause 7.6.8.

4895 In the mobile originated SM transfer this parameter contains the Service Centre address received from the mobile
 4896 station.

4897 **SM RP OA:**

4898 See definition in subclause 7.6.8.

4899 The MSISDN received from the VLR or from the SGSN is inserted in this parameter in the mobile originated SM
 4900 transfer.

4901 **SM RP UI:**

4902 See definition in subclause 7.6.8. The short message transfer protocol data unit received from the Service Centre is
4903 inserted in this parameter.

4904 **IMSI**

4905 See definition in subclause 7.6.2.1. The IMSI of the originating subscriber is inserted in this parameter in the mobile
4906 originated SM transfer.

4907 This parameter shall be included if the sending entity, whether MSC or SGSN, supports mobile number portability.

4908 **User error:**

4909 The following errors defined in subclause 7.6.1 may be used, depending on the nature of the fault:

4910 - Facility Not Supported;

4911 - System Failure;

4912 - SM Delivery Failure;

4913 - The reason of the SM Delivery Failure can be one of the following in the mobile originated SM:

4914 - unknown Service Centre address;

4915 - Service Centre congestion;

4916 - invalid Short Message Entity address;

4917 - subscriber not Service Centre subscriber;

4918 - protocol error.

4919 - Unexpected Data Value

4920 **Provider error:**

4921 For definition of provider errors see subclause 7.6.1.

4922 **12.3 MAP-REPORT-SM-DELIVERY-STATUS service**

4923 **12.3.1 Definition**

4924 This service is used between the gateway MSC and the HLR. The MAP-REPORT-SM-DELIVERY-STATUS service is
4925 used to set the Message Waiting Data into the HLR or to inform the HLR of successful SM transfer after polling. This
4926 service is invoked by the gateway MSC.

4927 The MAP-REPORT-SM-DELIVERY-STATUS service is a confirmed service using the service primitives given in
4928 table 12.3/1.

4929 **12.3.2 Service primitives**

4930 The service primitives are shown in table 12.3/1.

4931

Table 12.3/1: MAP-REPORT-SM-DELIVERY-STATUS

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(=)		
Absent Subscriber	C	C(=)		
Diagnostic SM				
GPRS Support Indicator	C	C(=)		
Delivery Outcome Indicator	C	C(=)		
Additional SM Delivery Outcome	C	C(=)		
Additional Absent Subscriber	C	C(=)		
Diagnostic SM				
MSISdn-Alert			C	C(=)
User error			C	C(=)
Provider error				O

4932

4933 **12.3.3 Parameter use**4934 **Invoke id:**

4935 See definition in subclause 7.6.1.

4936 **MSISDN:**

4937 See definition in subclause 7.6.2.

4938 **Service Centre Address:**

4939 See definition in subclause 7.6.2.

4940 **SM Delivery Outcome:**

4941 See definition in subclause 7.6.8. This parameter indicates the status of the mobile terminated SM delivery.

4942 **Absent Subscriber Diagnostic SM:**

4943 See definition in subclause 7.6.8.

4944 **GPRS Support Indicator:**4945 See definition in subclause 7.6.8. The presence of this parameter is mandatory if the SMS-GMSC supports handling of
4946 two delivery outcomes.4947 **Delivery Outcome Indicator:**

4948 See definition in subclause 7.6.8.

4949 **Additional SM Delivery Outcome:**

4950 See definition in subclause 7.6.8.

4951 **Additional Absent Subscriber Diagnostic SM:**

4952 See definition in subclause 7.6.8.

4953 **MSISdn-Alert:**4954 See definition in subclause 7.6.2. This parameter shall be present in case of unsuccessful delivery, when the MSISDN
4955 received in the operation is different from the stored MSISdn-Alert; the stored MSISdn-Alert is the value that is returned
4956 to the gateway MSC.4957 **User error:**

4958 The following errors defined in subclause 7.6.1 may be used, depending on the nature of the fault:

- 4959 - Unknown Subscriber;
- 4960 - Message Waiting List Full;
- 4961 - Unexpected Data Value;
- 4962 - Data missing.

4963 **Provider error:**

4964 For definition of provider errors see subclause 7.6.1.

4965 12.4 MAP-READY-FOR-SM service

4966 12.4.1 Definition

4967 This service is used between the MSC and VLR and as well between the VLR and the HLR. The MSC initiates this
4968 service if a subscriber indicates memory available situation. The VLR uses the service to indicate this to the HLR.

4969 The VLR initiates this service if a subscriber, whose message waiting flag is active in the VLR, has radio contact in the
4970 MSC.

4971 Also this service is used between the SGSN and the HLR. The SGSN initiates this service if a subscriber indicates
4972 memory available situation. The SGSN uses the service to indicate this to the HLR.

4973 The SGSN initiates this service if a subscriber, whose message waiting flag is active in the SGSN, has radio contact in
4974 the GPRS.

4975 The MAP-READY-FOR-SM service is a confirmed service using the primitives from table 12.4/1.

4976 12.4.2 Service primitives

4977 The service primitives are shown in table 12.4/1.

4978

Table 12.4/1: MAP-READY-FOR-SM

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

4979

4980 12.4.3 Parameter use

4981 **Invoke id:**

4982 See definition in subclause 7.6.1.

4983 See definition in subclause 7.6.2. The IMSI is used always between the VLR and the HLR and between the SGSN and
4984 the HLR. Between the MSC and the VLR the identification can be either IMSI or TMSI.

4985 **TMSI:**

4986 See definition in subclause 7.6.2. The identification can be either IMSI or TMSI between MSC and VLR.

4987 **Alert Reason:**

4988 See definition in subclause 7.6.8. This parameter indicates if the mobile subscriber is present or the MS has memory
4989 available.

4990 **Alert Reason Indicator:**

4991 See definition in subclause 7.6.8.

4992 **User error:**

4993 The following errors defined in subclause 7.6.1 may be used, depending on the nature of the fault:

- 4994 - Unknown Subscriber;
- 4995 - Facility Not Supported;
- 4996 - System Failure;
- 4997 - Unexpected Data Value;
- 4998 - Data missing;

4999 **Provider error:**

5000 For definition of provider errors see subclause 7.6.1.

5001 **12.5 MAP-ALERT-SERVICE-CENTRE service**5002 **12.5.1 Definition**

5003 This service is used between the HLR and the interworking MSC. The HLR initiates this service, if the HLR detects that
5004 a subscriber, whose MSISDN is in the Message Waiting Data file, is active or the MS has memory available.

5005 The MAP-ALERT-SERVICE-CENTRE service is a confirmed service using the primitives from table 12.5/1.

5006 **12.5.2 Service primitives**

5007 The service primitives are shown in table 12.5/1.

5008

Table 12.5/1: MAP-ALERT-SERVICE-CENTRE

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

5009

5010 **12.5.3 Parameter use**5011 **Invoke id:**

5012 See definition in subclause 7.6.1.

5013 **MSISdn-Alert:**

5014 See definition in subclause 7.6.2. The provided MSISDN shall be the one which is stored in the Message Waiting Data
5015 file.

5016 **Service Centre Address:**

5017 See definition in subclause 7.6.2.

5018 **User error:**

5019 The following errors defined in subclause 7.6.1 may be used, depending on the nature of the fault:

5020 - System Failure;

5021 - Unexpected Data Value;

5022 - Data missing.

5023 **Provider error:**

5024 For definition of provider errors see subclause 7.6.1.

5025 **12.6 MAP-INFORM-SERVICE-CENTRE service**5026 **12.6.1 Definition**

5027 This service is used between the HLR and the gateway MSC to inform the Service Centre which MSISDN number is
 5028 stored in the Message Waiting Data file. If the stored MSISDN number is not the same than the one received from the
 5029 gateway MSC in the MAP-SEND-ROUTING-INFO-FOR-SM service primitive the stored MSISDN number is included
 5030 in the message.

5031 Additionally the status of MCEF, MNRF and MNRG flags and the inclusion of the particular Service Centre address in
 5032 the Message Waiting Data list is informed to the gateway MSC when appropriate.

5033 The MAP-INFORM-SERVICE-CENTRE service is a non-confirmed service using the primitives from table 12.6/1.

5034 **12.6.2 Service primitives**

5035 The service primitives are shown in table 12.6/1.

5036

Table 12.6/1: MAP-INFORM-SERVICE-CENTRE

Parameter name	Request	Indication
Invoke Id	M	M(=)
MSIsdn-Alert	C	C(=)
MWD Status	C	C(=)

5037

5038 **12.6.3 Parameter use**5039 **Invoke id:**

5040 See definition in subclause 7.6.1.

5041 **MSIsdn-Alert:**

5042 See definition in subclause 7.6.2 This parameter refers to the MSISDN stored in a Message Waiting Data file in the
 5043 HLR.

5044 **MWD Status:**

5045 See definition in subclause 7.6.8. This parameter indicates the status of the MCEF, MNRF and MNRG flags and the
 5046 status of the particular SC address presence in the Message Waiting Data list.

5047 **12.7 MAP-SEND-INFO-FOR-MT-SMS service**5048 **12.7.1 Definition**

5049 This service is used between the MSC and the VLR. The service is invoked by the MSC receiving an mobile terminated
5050 short message to request subscriber related information from the VLR.

5051 The MAP-SEND-INFO-FOR-MT-SMS service is a confirmed service using the primitives from table 12.7/1.

5052 **12.7.2 Service primitives**

5053 The service primitives are shown in table 12.7/1.

5054 **Table 12.7/1: MAP-SEND-INFO-FOR-MT-SMS**

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

5055

5056 **12.7.3 Parameter use**5057 **Invoke id:**

5058 See definition in subclause 7.6.1.

5059 **SM RP DA:**

5060 See definition in subclause 7.6.8. This parameter shall contain either an IMSI or a LMSI.

5061 **MSISDN:**

5062 See definition in subclause 7.6.2.

5063 **User error:**

5064 The following errors defined in subclause 7.6.1 may be used, depending on the nature of the fault:

- 5065 - Unknown subscriber;
- 5066 - Unidentified Subscriber;
- 5067 - Absent subscriber;
- 5068 - Unexpected Data Value;
- 5069 - Data Missing;
- 5070 - Illegal subscriber;
- 5071 - Illegal equipment;
- 5072 - Subscriber busy for MT SMS;
- 5073 - System Failure.

5074 **Provider error:**

5075 For definition of provider errors see subclause 7.6.1.

5076 12.8 MAP-SEND-INFO-FOR-MO-SMS service

5077 12.8.1 Definition

5078 This service is used between the MSC and the VLR. The service is invoked by the MSC which has to handle a mobile
5079 originated short message request to request the subscriber related information from the VLR.

5080 The MAP-SEND-INFO-FOR-MO-SMS service is a confirmed service using the primitives from table 12.8/1.

5081 12.8.2 Service primitives

5082 The service primitives are shown in table 12.8/1.

5083 **Table 12.8/1: MAP-SEND-INFO-FOR-MO-SMS**

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

5084

5085 12.8.3 Parameter use

5086 **Invoke id:**

5087 See definition in subclause 7.6.1.

5088 **Service Centre Address:**

5089 See definition in subclause 7.6.2.

5090 **MSISDN:**

5091 See definition in subclause 7.6.2.

5092 **User error:**

5093 The following errors defined in subclause 7.6.1 may be used, depending on the nature of the fault:

5094 - Teleservice Not Provisioned;

5095 - Call Barred;

5096 - Unexpected Data Value;

5097 - Data Missing.

5098 **Provider error:**

5099 For definition of provider errors see subclause 7.6.1.

5100 12.9 MAP-MT-FORWARD-SHORT-MESSAGE service

5101 12.9.1 Definition

5102 This service is used between the gateway MSC and the servicing MSC or the SGSN to forward mobile mobile
5103 terminated short messages.

5104 The MAP-MT-FORWARD-SHORT-MESSAGE service is a confirmed service using the service primitives given in
5105 table 12.9/1.

5106 12.9.2 Service primitives

5107 The service primitives are shown in table 12.9/1.

5108 **Table 12.9/1: MAP-MT-FORWARD-SHORT-MESSAGE**

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(=)	C	C(=)
More Messages To Send	C	C(=)		
User error			C	C(=)
Provider error				O

5109

5110 12.9.3 Parameter use

5111 **Invoke id:**

5112 See definition in subclause 7.6.1.

5113 **SM RP DA:**

5114 See definition in subclause 7.6.8. This parameter can contain either an IMSI or a LMSI. The use of the LMSI is an
5115 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
5116 not available.

5117 This parameter is omitted in the mobile terminated subsequent SM transfers.

5118 **SM RP OA:**

5119 See definition in subclause 7.6.8. The Service Centre address received from the originating Service Centre is inserted in
5120 this parameter .

5121 This parameter is omitted in the mobile terminated subsequent SM transfers.

5122 **SM RP UI:**

5123 See definition in subclause 7.6.8. The short message transfer protocol data unit received from the Service Centre is
5124 inserted in this parameter. A short message transfer protocol data unit may also be inserted in this parameter in the
5125 message delivery acknowledgement from the MSC or from the SGSN to the Service Centre.

5126 **More Messages To Send:**

5127 See definition in subclause 7.6.8. The information from the MMS indication received from the Service Centre is inserted
5128 in this parameter.

5129 **User error:**

5130 The following errors defined in subclause 7.6.1 may be used, depending on the nature of the fault:

- 5131 - Unidentified subscriber;
 - Absent Subscriber_SM;
- 5132 - Subscriber busy for MT SMS;
- 5133 - Facility Not Supported;
- 5134 - Illegal Subscriber indicates that delivery of the mobile terminated short message failed because the mobile station
 5135 failed authentication;
- 5136 - Illegal equipment indicates that delivery of the mobile terminated short message failed because an IMEI check
 5137 failed, i.e. the IMEI was blacklisted or not white-listed;
- 5138 - System Failure;
- 5139 - SM Delivery Failure;
- 5140 - The reason of the SM Delivery Failure can be one of the following in the mobile terminated SM:
 5141 - memory capacity exceeded in the mobile equipment;
 5142 - protocol error;
 5143 - mobile equipment does not support the mobile terminated short message service.
- 5144 - Unexpected Data Value;
- 5145 - Data Missing.
- 5146 **Provider error:**
- 5147 For definition of provider errors see subclause 7.6.1.

5148 13 Network-Requested PDP Context Activation services

5149 13.1 MAP_SEND_ROUTING_INFO_FOR_GPRS service

5150 13.1.1 Definition

5151 This service is used by the GGSN to request GPRS routing information from the HLR.

5152 13.1.2 Service primitives

5153 **Table 13.1/1: MAP_SEND_ROUTING_INFO_FOR_GPRS**

Parameter name	Request	Indication	Response	Confirm
Invoke id	M	M(=)	M(=)	M(=)
IMSI	M	M(=)		
GGSN address	C	C(=)	C	C(=)
GGSN number	M	M(=)		
SGSN address			C	C(=)
Mobile Not Reachable Reason			C	C(=)
User error			C	C(=)
Provider error				O

5154

5155 13.1.3 Parameter definition and use

5156 Invoke Id

5157 See definition in subclause 7.6.1.

5158 IMSI

5159 See definition in subclause 7.6.2.

5160 GGSN address

5161 This parameter shall be present if the protocol-converting GSN is used between the GGSN and the HLR.

5162 GGSN number

5163 See definition in subclause 7.6.2.

5164 SGSN address

5165 This parameter shall be present if the outcome of the Send Routing Info For GPRS request to the GPRS application
5166 process in the HLR is positive.

5167 Mobile Not Reachable Reason

5168 This parameter shall be present if the outcome of the Send Routing Info For GPRS request to the GPRS application
5169 process in the HLR is positive and the MNRG flag in the HLR is set. See definition in subclause 7.6.3.51.

5170 User error

5171 This parameter is sent by the responder when an error is detected and if present, takes one of the following values:

5172 - Absent Subscriber;

5173 - System Failure;

5174 - Data Missing;

5175 - Unexpected Data Value;

5176 - UnknownSubscriber.

5177 The diagnostic in the Unknown Subscriber may indicate "Imsi Unknown" or "Gprs Subscription Unknown".

5178 Provider error

5179 These are defined in subclause 7.6.1.

5180 13.2 MAP_FAILURE_REPORT service

5181 13.2.1 Definition

5182 This service is used by the GGSN to inform the HLR that network requested PDP-context activation has failed.

5183 13.2.2 Service primitives

5184 Table 13.2/1: MAP_FAILURE_REPORT

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

5185

5186 13.2.3 Parameter definition and use

5187 Invoke Id

5188 See definition in subclause 7.6.1.

5189 IMSI

5190 See definition in subclause 7.6.2.

5191 GGSN address

5192 This parameter shall be present if the protocol-converting GSN is used between the GGSN and the HLR.

5193 GGSN number

5194 See definition in subclause 7.6.2.

5195 User error

5196 This parameter is sent by the responder when an error is detected and if present, takes one of the following values:

- 5197 - System Failure;
- 5198 - Data Missing;
- 5199 - Unexpected Data Value;
- 5200 - UnknownSubscriber.

5201 Provider error

5202 These are defined in subclause 7.6.1.

5203 13.3 MAP_NOTE_MS_PRESENT_FOR_GPRS service

5204 13.3.1 Definition

5205 This service is used by the HLR to inform the GGSN that the MS is present for GPRS again.

5206 13.3.2 Service primitives

5207 Table 13.3/1: MAP_NOTE_MS_PRESENT_FOR_GPRS

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

5208

5209 13.3.3 Parameter definition and use

5210 Invoke Id

5211 See definition in subclause 7.6.1.

5212 IMSI

5213 See definition in subclause 7.6.2.

5214 GGSN address

5215 This parameter shall be present if the protocol-converting GSN is used between the GGSN and the HLR.

5216 SGSN address

5217 See definition in subclause 7.6.2.

5218 User error

5219 This parameter is sent by the responder when an error is detected and if present, takes one of the following values:

5220 - System Failure;

5221 - Data Missing;

5222 - Unexpected Data Value;

5223 - UnknownSubscriber.

5224 Provider error

5225 These are defined in subclause 7.6.1.

5226 13A Location Service Management Services

5227 13A.1 MAP-SEND-ROUTING-INFO-FOR-LCS Service

5228 13A.1.1 Definition

5229 This service is used between the GMLC and the HLR to retrieve the routing information needed for routing a location
 5230 service request to the servicing VMSC. The MAP-SEND-ROUTING-INFO-FOR-LCS is a confirmed service using the
 5231 primitives from table A.1/1.

5232 13A.1.2 Service Primitives

5233 The service primitives are shown in table 13A.1/1.

5234

5235 **Table 13A.1/1: MAP-SEND-ROUTING-INFO-FOR-LCS**

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

5236

5237 13A.1.3 Parameter Use

5238 Invoke id:

5239 See definition in subclause 7.6.1.

5240 MLC Number:

5241 See definition in subclause 7.6.2.

5242 MSISDN:

5243 See definition in subclause 7.6.2. The request shall carry either the IMSI or MSISDN. The response shall carry
 5244 whichever of these was not included in the request (see GSM 03.71 for details).

5245 IMSI:

5246 See definition in subclause 7.6.2.

5247 LMSI:

5248 See definition in subclause 7.6.2. It is an operator option to provide this parameter from the VLR; it is mandatory for the
5249 HLR to include the LMSI in a successful response, if the VLR has used the LMSI.

5250 MSC Number:

5251 See definition in subclause 7.6.2. This parameter is provided in a successful response.

5252 User error:

5253 The following errors defined in subclause 7.6.1 may be used, depending on the nature of the fault:

- 5254 - Unknown subscriber;
- 5255 - Absent Subscriber;
- 5256 - Facility Not Supported;
- 5257 - System failure;
- 5258 - Unexpected Data Value;
- 5259 - Data missing;
- 5260 - Unauthorized requesting network

5261 Provider error:

5262 For definition of provider errors see subclause 7.6.1.

5263 13A.2 MAP-PROVIDE-SUBSCRIBER-LOCATION Service

5264 13A.2.1 Definition

5265 This service is used by a GMLC to request the location of a target MS from the visited MSC at any time. This is a
5266 confirmed service using the primitives from table 13A.2/1.

5267 13A.2.2 Service Primitives

5268 **Table 13A.2/1: Provide_Subscriber_Location**

Parameter name	Request	Indication	Response	Confirm
Invoke id	M	M(=)	M(=)	M(=)
Location Type	M	M(=)		
MLC Number	M	M(=)		
LCS Client ID	M	M(=)		
Privacy Override	U	C(=)		
IMSI	C	C(=)		
MSISDN	C	C(=)		
LMSI	C	C(=)		
LCS Priority	C	C(=)		
LCS QoS	C	C(=)		
IMEI	U	C(=)		
Location Estimate			M	M(=)
Age of Location Estimate			C	C(=)
User error			C	C(=)
Provider error				O

5269

5270 13A.2.3 Parameter Definition and Use

5271 All parameters are defined in section 7.6. The use of these parameters and the requirements for their presence are
5272 specified in GSM 03.71.

5273 Location Type

5274 This parameter identifies the type of location information requested

5275 MLC Number

5276 This is the E.164 number of the requesting GMLC.

5277 LCS Client ID

5278 This parameter provides information related to the identity of an LCS client.

5279 Privacy Override

5280 This parameter indicates if MS privacy is overridden by the LCS client when the GMLC and VMSC for an MR-LR are
5281 in the same country.

5282 IMSI

5283 The IMSI is provided to identify the target MS. At least one of the IMSI or MSISDN is mandatory.

5284 MSISDN

5285 The MSISDN is provided to identify the target MS. At least one of the IMSI or MSISDN is mandatory.

5286 LMSI

5287 The LMSI shall be provided if previously supplied by the HLR

5288 LCS Priority

5289 This parameter indicates the priority of the location request.

5290 LCS QoS

5291 This parameter indicates the required quality of service in terms of response time and accuracy.

5292 IMEI

5293 Inclusion of the IMEI is optional.

5294 Location Estimate

5295 This parameter provides the location estimate.

5296 Age of Location Estimate

5297 This parameter indicates how long ago the location estimate was obtained.

5298 User error

5299 This parameter is sent by the responder when the location request has failed or cannot proceed and if present, takes one
5300 of the following values defined in section 7.6.1.

5301 - System Failure;

5302 - Data Missing;

5303 - Unexpected Data Value;

5304 - Facility Not Supported;

- 5305 - Unidentified Subscriber;
- 5306 - Illegal Subscriber;
- 5307 - Illegal Equipment;
- 5308 - Absent Subscriber (diagnostic information may also be provided);
- 5309 - Unauthorized requesting network;
- 5310 - Unauthorized LCS Client with detailed reason;
- 5311 - Position method failure with detailed reason.

5312 Provider error

5313 These are defined in subclause 7.6.1.

5314 13A.3 MAP-SUBSCRIBER-LOCATION-REPORT Service

5315 13A.3.1 Definition

5316 This service is used by a VMSC to provide the location of a target MS to a GMLC when a request for location is either
 5317 implicitly administered or made at some earlier time. This is a confirmed service using the primitives from
 5318 table 13A.3/1.

5319 13A.3.2 Service Primitives

5320 **Table 13A.3/1: Subscriber_Location_Report**

Parameter name	Request	Indication	Response	Confirm
Invoke id	M	M(=)	M(=)	M(=)
LCS Event	M	M(=)		
LCS Client ID	M	M(=)		
MSC Number	M	M(=)		
IMSI	C	C(=)		
MSISDN	C	C(=)		
NA-ESRD	C	C(=)		
NA-ESRK	C	C(=)		
IMEI	U	C(=)		
Location Estimate	C	C(=)		
Age of Location Estimate	C	C(=)		
LMSI	U	C(=)		
User error			C	C(=)
Provider error				O

5321

5322 13A.3.3 Parameter Definition and Use

5323 All parameters are defined in section 7.6. The use of these parameters and the requirements for their presence are
5324 specified in GSM 03.71.

5325 LCS Event

5326 This parameter indicates the event that triggered the Subscriber Location Report.

5327 LCS Client ID

5328 This parameter provides information related to the identity of the recipient LCS client.

5329 MSC Number:

5330 See definition in subclause 7.6.2. This parameter provides the address of the visited MSC for target MS.

5331 IMSI

5332 The IMSI shall be provided if available to the VMSC

5333 MSISDN

5334 The MSISDN shall be provided if available to the VMSC

5335 NA-ESRD

5336 If the target MS has originated an emergency service call in North America, the NA-ESRD shall be provided by the
5337 VMSC if available.

5338 NA-ESRK

5339 If the target MS has originated an emergency service call in North America, the NA-ESRK shall be provided by the
5340 VMSC if assigned.

5341 IMEI

5342 Inclusion of the IMEI is optional.

5343 Location Estimate

5344 This parameter provides the location estimate. The absence of this parameter implies that a location estimate was not
5345 available or could not be successfully obtained.

5346 Age of Location Estimate

5347 This parameter indicates how long ago the location estimate was obtained.

5348 LMSI

5349 The LMSI may be provided if assigned by the VLR.

5350 User error

5351 This parameter is sent by the responder when the received message contains an error, cannot be forwarded or stored for
5352 an LCS client or cannot be accepted for some other reason and if present, takes one of the following values defined in
5353 section 7.6.1.

5354 - System Failure;

5355 - Data Missing;

5356 - Unexpected Data Value;

5357 - Resource Limitation;

5358 - Unknown Subscriber;

5359 - Unauthorized requesting network;

5360 - Unknown or unreachable LCS Client.

5361 Provider error

5362 These are defined in subclause 7.6.1.

5363 **13A.4 Void**

5364 **13A.4.1 Void**

5365 **13A.4.2 Void**

5366 **13A.4.3 Void**

5367 **13A.5 Void**

5368 **13A.5.1 Void**

5369 **13A.5.2 Void**

5370 **13A.5.3 Void**

5371 **13A.6 Void**

5372 13A.6.1 Void

5373 13A.6.2 Void

5374 13A.6.3 Void

5375 13A.7 Void

5376 13A.7.1 Void

5377 13A.7.2 Void

5378 13A.7.3 Void

5379 13A.8 Void

5380 13A.8.1 Void

5381 13A.8.2 Void

5382 13A.8.3 Void

5383 13A.9 Void

5384 13A.9.1 Void

5385 13A.9.2 Void

5386 13A.9.3 Void

5387 14 General

5388 14.1 Overview

5389 Clause 14 to 17 specify the protocol elements to be used to provide the MAP services described in clause 7.

5390 Clause 15 specifies the elements of procedures for the MAP protocol. Clause 16 specifies the mapping on to TC service
5391 primitives. Clause 17 specifies the application contexts, operation packages and abstract syntaxes for the MAP protocol
5392 as well as the encoding rules to be applied.

5393 14.2 Underlying services

5394 The MAP protocol relies on the services provided by the Transaction Capabilities (TC) of signalling system number 7,
5395 as referenced in clause 6.

5396 14.3 Model

5397 The MAP Protocol Machine (MAP PM) can be modelled as a collection of service state machines (SSMs) - one per
5398 MAP specific service invoked - coordinated by a MAP dialogue control function with its one state machine: MAP
5399 dialogue state machine (DSM). There are two types of Service State Machines: Requesting Service State Machines
5400 (RSM) and Performing Service State Machines (PSM).

5401 A new invocation of a MAP PM is employed on the receipt of a MAP-OPEN request primitive or a TC-BEGIN
5402 indication primitive. Each invocation controls exactly one MAP dialogue. For each MAP specific service invoked
5403 during a dialogue, a MAP RSM is created at the requestor's side and a MAP PSM is created at the performer's side.

5404 This modelling is used only to facilitate understanding and the MAP behaviour descriptions and is not intended to
5405 suggest any implementation. SDL descriptions are organized according to this model.

5406 How the MAP-service-user and the MAP refer to a MAP dialogue (i.e. a MAP PM invocation) is a local implementation
5407 matter.

5408 How TC dialogue identifiers are assigned to a MAP PM invocation is also a local implementation matter.

5409 14.4 Conventions

5410 The behaviour of the MAP PM depends on the application-context-name associated with the dialogue. One major
5411 difference is that the MAP requests the transfer of the application-context-name by TC only for those contexts which do
5412 not belong to the so-called "version one context set".

5413 The "version one context set" is a set of application-contexts which model the behaviour of a MAP V1 implementation
5414 according to the latest phase 1 version of GSM 09.02. This set is defined in clause 15.

5415 The procedures described in clause 15 are used when the application-context-name does not refer to a dialogue between
5416 an MSC and its VLR. When the application-context-name refers to a dialogue between an MSC and its VLR the MAP
5417 PM procedures are a local implementation matter.

5418 15 Elements of procedure

5419 15.1 Dialogue establishment

5420 The establishment of a MAP dialogue involves two MAP-service-users, one that is the dialogue-initiator and one that is
5421 the dialogue-responder.

5422 This procedure is driven by the following signals:

- 5423 - a MAP-OPEN request primitive from the dialogue-initiator;
- 5424 - a TC-BEGIN indication primitive occurring at the responding side;
- 5425 - a MAP-OPEN response primitive from the dialogue-responder;
- 5426 - the first TC-CONTINUE indication primitive occurring at the initiating side;

5427 and under specific conditions:

- 5428 - a TC-END indication primitive occurring at the initiating side;
- 5429 - a TC-U-ABORT indication primitive occurring at the initiating side;
- 5430 - a TC-P-ABORT indication primitive occurring at the initiating side.

5431 15.1.1 Handling of unknown operations

5432 Unknown operations (i.e. a standard operation introduced in a later version of 09.02 or a private operation) can be
5433 introduced in MAP in a backwards compatible way. This means, that the receiver of an unknown operation shall, if the
5434 dialogue state allows it, send a TC-REJECT component to the sender of the operation indicating 'unrecognised
5435 operation' and continue with the processing of further components or messages exchanged within the dialogue as if the
5436 unknown operation had not been received.

5437 The standardised structure of a MAP dialogue shall not be affected by the invocation of unknown operations, i.e. if a
5438 dialogue uses only a TC-BEGIN message which is acknowledged by a TC-END message, a TC-CONTINUE message
5439 shall not be used to invoke an unknown operation. However the standardised structure of a MAP dialogue may be
5440 affected by the rejection of unknown operations, i.e. if a dialogue uses only a TC-BEGIN message which is
5441 acknowledged by a TC-END message, a TC-CONTINUE message followed by a TC-END message may be used to
5442 carry the rejection of an unknown operation and the response to the standardised operation. The entity which initiated a
5443 dialogue whose standardised structure is a TC-BEGIN message which is acknowledged by a TC-END message shall not
5444 send any messages in that dialogue after the TC-BEGIN.

5445 Note that if the dialogue structure is affected as described in this paragraph the TC-CONTINUE shall include the
5446 dialogue portion required to confirm the acceptance of the dialogue.

5447 Unknown operations can be invoked in the following types of messages (there is no restriction as to how many unknown
5448 operations can be invoked in a message):

- 5449 - TC-BEGIN the component to invoke the unknown operation shall follow the component of the standard
5450 operation that is included in this message.
- 5451 - TC-CONTINUE: the component to invoke the unknown operation may be transported as the only component
5452 in a stand-alone message or can be grouped with existing operations. In the latter case a specific sequencing
5453 of components is not required.
- 5454 - TC-END: if the component to invoke the unknown operation is grouped with an existing operation a specific
5455 sequencing of components is not required

5456 The TC-REJECT component may be sent in the following messages:

- 5457 - TC-CONTINUE or TC-END: either as the only component of the message or grouped with an existing
5458 component. The choice is up to the MAP-Service User.

5459 If the received message contains only unknown operations the MAP-Service User shall send the TC-REJECT
5460 components in a TC-CONTINUE message to the peer entity, if the dialogue state allows it.

5461 If the received message contains unknown operations and standard operations and the standardised structure
5462 of the dialogue requires the response to the standard operation to be sent within a TC-END message, then the
5463 MAP-Service User may send the response to the standard operations and the TC-REJECT components for the
5464 unknown operations in a TC-CONTINUE message followed by a TC-END message. A specific distribution
5465 of the components to the TC messages or a specific sequencing of components is not required.

5466 Note that SDLs of chapters 19 - 25 do not show the report to the MAP-Service User about the reception of the
5467 unknown operation. This has been done for the sake of simplicity of description; the MAP PM may inform the MAP-
5468 Service User.

5469 The sender of the unknown operation shall ensure that there is enough room in the used message for the unknown
5470 operation.

5471 15.1.2 Receipt of a MAP-OPEN request primitive

5472 On receipt of a MAP-OPEN request primitive the behaviour of the MAP PM shall be as follows:

5473 The MAP PM shall accept zero, one or several user request primitives until a MAP-DELIMITER request primitive is
5474 received.

5475 For each user request primitive, the MAP PM shall request the invocation of the associated operation using the TC-
5476 INVOKE service. See subclause 15.6 for a description of the associated SSMs.

5477 On receipt of the MAP-DELIMITER request primitive the MAP PM shall issue a TC-BEGIN request primitive. The
 5478 application-context-name as well as the user information parameter (if any) shall be mapped to the corresponding TC-
 5479 BEGIN parameters.

5480 The requesting MAP PM waits for a TC indication primitive and does not accept any other primitive from its user,
 5481 except a MAP-U-ABORT request or a MAP-CLOSE request.

5482 15.1.3 Receipt of a TC-BEGIN indication

5483 On receipt of a TC-BEGIN indication primitive, the MAP PM shall:

5484 - if no application-context-name is included in the primitive and if the "Components present" indicator indicates
 5485 "no components", issue a TC-U-ABORT request primitive (note 2). The local MAP-User is not informed.

5486 - if no application-context-name is included in the primitive and if presence of components is indicated, wait for
 5487 the first TC-INVOKE primitive, and derive a version 1 application-context-name from the operation code
 5488 according to table 15.1/1 (note 1).

5489 NOTE 1: In some cases, it may be necessary to analyse the operation argument.

5490 Then:

5491 a) if no application-context-name can be derived (i.e. the operation code does not exist in MAP V1
 5492 specifications), the MAP PM shall issue a TC-U-ABORT request primitive (note 2). The local MAP-User is
 5493 not informed.

5494 b) if an application-context-name can be derived and if it is acceptable from a load control point of view, the
 5495 MAP PM shall:

5496 i) if this primitive requests the beginSubscriberActivity operation, the MAP PM shall check whether more
 5497 components have been received associated with this operation. If more components are present, the MAP
 5498 PM shall issue a MAP-OPEN indication primitive with the version 1 application-context-name
 5499 "networkFunctionalSsContext-v1". The Destination-reference shall include the IMSI taken from the
 5500 argument of the beginSubscriberActivity operation; the Originating-reference shall cover the
 5501 originatingEntityNumber.

5502 A beginSubscriberActivity operation that is not associated with any other Component shall be rejected by
 5503 the MAP PM by issuing a TC-U-ABORT request primitive (note 2). The local MAP-User shall not be
 5504 informed.

5505 ii) otherwise, the MAP PM shall issue a MAP-OPEN indication primitive with the version 1 application-
 5506 context-name set according to table 15.1/1. DestinationReference and OriginatingReference must not be
 5507 included in the MAP-OPEN indication primitive.

5508 Then the MAP PM shall function in a way that the dialogue responding MAP behaves as specified in the
 5509 GSM phase 1 protocol (latest version of TS GSM 09.02 phase 1).

5510 NOTE 2: If no AARQ apdu was included in the BEGIN message, TC (Component Sub-layer) will not include an
 5511 AARE apdu or an ABRT apdu in a TR-U-ABORT request primitive that is to be issued on receipt of a
 5512 TC-U-ABORT request primitive from the local MAP service provider.

5513 c) if an application-context-name can be derived but if it is not acceptable from a load control point of view, the
 5514 MAP PM shall ignore this dialogue request and not inform the MAP-user;

5515 - if a version 1 application-context-name is included, the MAP PM shall issue a TC-U-ABORT request primitive
 5516 with abort-reason "User-specific" and user-information "MAP-ProviderAbortInfo" indicating
 5517 "abnormalDialogue". The local MAP-user shall not be informed.

5518 - if an application-context-name different from version 1 is included in the primitive and if User-information is
 5519 present, the User-information must constitute a syntactically correct MAP-OPEN dialogue PDU. Otherwise a
 5520 TC-U-ABORT request primitive with abort-reason "User-specific" and user-information "MAP-
 5521 ProviderAbortInfo" indicating "abnormalDialogue" shall be issued and the local MAP-user shall not be
 5522 informed.

- 5523 - if no User-information is present it is checked whether presence of User Information in the TC-BEGIN
5524 indication primitive is required for the received application-context-name. If User Information is required but
5525 not present, a TC-U-ABORT request primitive with abort-reason "User-specific" and user-information
5526 "MAP-ProviderAbortInfo" indicating "abnormalDialogue" shall be issued. The local MAP-user shall not be
5527 informed.
- 5528 - if an application-context-name different from version 1 is received in a syntactically correct TC-BEGIN
5529 indication primitive but is not acceptable from a load control point of view, the MAP PM shall ignore this
5530 dialogue request. The MAP-user is not informed.
- 5531 - if an application-context-name different from version 1 is received in a syntactically correct TC-BEGIN
5532 indication primitive and if it is acceptable from a load control point of view, the MAP PM shall check whether
5533 the application-context-name is supported.

5534 NOTE 3: Unknown application-context-names are treated like unsupported ones.

5535 If it is, the MAP PM shall issue a MAP-OPEN indication primitive with all parameters (application-context-
5536 name included) set according to the value of the corresponding parameter of the TC-BEGIN indication
5537 primitive.

5538 The MAP PM shall then process any other indication primitives received from TC as described in
5539 subclause 15.6. Once all the received components have been processed, the MAP PM shall inform the local MAP
5540 service user by a MAP-DELIMITER indication primitive.

5541 If the TC-BEGIN indication primitive is not associated with any component, the MAP PM shall inform the MAP
5542 User by a MAP-DELIMITER indication primitive.

5543 Once all the received primitives have been processed, the MAP PM does not accept any primitive from the
5544 provider and waits for a MAP-OPEN response primitive from its user.

- 5545 - if an application-context-name different from version 1 is received in a syntactically correct TC-BEGIN
5546 indication primitive and if it is acceptable from a load control point of view but the application-context-name
5547 is not supported, the MAP PM shall issue a TC-U-ABORT request primitive with abort-reason indicating
5548 "application-context-not-supported". If an alternative application-context-name cannot be offered, the
5549 received application-context-name shall be returned in the TC-U-ABORT Req primitive.

5550 In the following cases an alternative application-context can be offered and its name included in the TC-U-
5551 ABORT Req primitive:

- 5552 a) if an application-context of version 2 or higher is requested, but only version 1 application-context supported,
5553 then the v1 application context shall be returned;
- 5554 b) if an application-context of version 3 or higher is requested, but only version 2 application-context supported,
5555 then the v2 application context shall be returned.
- 5556 c) if an application-context of version 4 or higher is requested, but only version 3 application-context supported,
5557 then the v3 application context shall be returned.

5558

Table 15.1/1: Mapping of V1 operation codes on to application-context-names

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

5559

5560 NOTE 1: These symbolic names refer to the object identifier value defined in clause 17 and allocated to each
5561 application-context used for the MAP.

5562 NOTE 2: The choice between the application contexts is based on the parameters received in the operation.

5563 15.1.4 Receipt of a MAP-OPEN response

5564 On receipt of a MAP-OPEN response primitive indicating that the dialogue is accepted, the MAP PM shall build a
5565 MAP-Accept PDU if the user-information parameter is included in the response primitive and accept any MAP specific
5566 service request or service response until a MAP-DELIMITER request or a MAP-CLOSE request is received from the
5567 MAP user. The MAP PM shall process the MAP specific primitives as described in subclause 15.6. The MAP PM shall
5568 then issue a TC-CONTINUE request primitive after it receives the MAP-DELIMITER request primitive if no MAP-
5569 CLOSE request primitive has been received, otherwise it shall issue a TC-END request primitive. In both cases the
5570 MAP-Accept PDU (if any) is included in the user-information parameter of the TC primitive.

5571 If the dialogue is not associated with a version 1 application context, the MAP PM shall include the application-context-
5572 name in the TC primitive.

5573 If no MAP-CLOSE request has been received, the MAP PM waits for a request primitive from its user or an indication
5574 primitive from TC.

5575 On receipt of a MAP-OPEN response primitive indicating that the dialogue is not accepted, the MAP PM shall build a
5576 MAP-Refuse PDU and request its transfer using the TC-U-ABORT req primitive (abort reason = user specific).

5577 15.1.5 Receipt of the first TC-CONTINUE ind

5578 On receipt of the first TC-CONTINUE indication primitive for a dialogue, the MAP PM shall check the value of the
5579 application-context-name parameter. If this value matches the one used in the MAP-OPEN request primitive, the MAP
5580 PM shall issue a MAP-OPEN confirm primitive with the result parameter indicating "accepted", then process the
5581 following TC component handling indication primitives as described in subclause 15.6, and then waits for a request
5582 primitive from its user or an indication primitive from TC, otherwise it shall issue a TC-U-ABORT request primitive
5583 with a MAP-providerAbort PDU indicating "abnormal dialogue" and a MAP-P-ABORT indication primitive with the
5584 "provider-reason" parameter indicating "abnormal dialogue".

5585 15.1.6 Receipt of a TC-END ind

5586 On receipt of a TC-END indication primitive in the dialogue initiated state, the MAP PM shall check the value of the
5587 application-context-name parameter. If this value does not match the one used in the MAP-OPEN request primitive, the

5588 MAP PM shall discard any following component handling primitive and shall issue a MAP-P-ABORT indication
5589 primitive with the "provider-reason" parameter indicating "abnormal dialogue".

5590 Otherwise it shall issue a MAP-OPEN confirm primitive with the result parameter set to "accepted" and process the
5591 following TC component handling indication primitives as described in subclause 15.6; then it shall issue a MAP-
5592 CLOSE indication primitive and return to idle all state machines associated with the dialogue.

5593 15.1.7 Receipt of a TC-U-ABORT ind

5594 On receipt of a TC-U-ABORT indication primitive in the "Dialogue Initiated" state with an abort-reason parameter
5595 indicating "ApplicationContextNotSupported", the MAP PM shall issue a MAP-OPEN confirm primitive with the result
5596 parameter indicating "Dialogue Refused" and the refuse-reason parameter indicating
5597 "ApplicationContextNotSupported".

5598 On receipt of a TC-U-ABORT indication primitive in the "Dialogue Initiated" state with an abort-reason parameter
5599 indicating "User Specific" and without user information, the MAP PM shall issue a MAP-OPEN confirm primitive with
5600 the result parameter indicating "Dialogue Refused" and the refuse-reason parameter indicating "Potential Version
5601 Incompatibility".

5602 On receipt of a TC-U-ABORT indication primitive in the "Dialogue Initiated" state with an abort-reason parameter
5603 indicating "User Specific" and a MAP-Refuse PDU included as user information, the MAP PM shall issue a MAP-
5604 OPEN confirm primitive with the result set to refused and the refuse reason set as received in the MAP Refuse PDU.

5605 Receipt of a TC-U-ABORT indication primitive with abort-reason "User Specific" and with user information is
5606 described as part of abnormal termination (see subclause 15.4.2).

5607 15.1.8 Receipt of a TC-P-ABORT ind

5608 On receipt of a TC-P-ABORT indication primitive in the "Dialogue Initiated" state with a P-abort parameter indicating
5609 "Incorrect Transaction Portion", the MAP PM shall issue a MAP-OPEN confirm primitive with the result parameter
5610 indicating "Dialogue Refused" and the refuse reason parameter indicating "Potential Version Incompatibility".

5611 On receipt of a TC-P-ABORT indication primitive in the "Dialogue Initiated" state with a P-abort parameter indicating
5612 "No Common Dialogue Portion", the MAP PM shall issue a MAP-P-ABORT indication primitive with the provider
5613 reason parameter indicating "Version Incompatibility".

5614 Receipt of a TC-P-ABORT indication primitive with another P-abort parameter value is described as part of abnormal
5615 termination (see subclause 15.5.2).

5616 15.2 Dialogue continuation

5617 Once established the dialogue is said to be in a continuation phase.

5618 Both MAP users can request the transfer of MAP APDUs until one of them requests the termination of the dialogue.

5619 15.2.1 Sending entity

5620 The MAP PM shall accept any MAP specific service request or response primitives and process them as described in
5621 subclause 15.6.

5622 On receipt of a MAP-DELIMITER request primitive, the MAP PM shall issue a TC-CONTINUE request primitive.

5623 15.2.2 Receiving entity

5624 On receipt of a TC-CONTINUE indication primitive the MAP PM shall accept zero, one or several TC component
5625 handling indication primitives and process them as described in subclause 15.6.

5626 15.3 Dialogue termination

5627 Both the dialogue-initiator and the dialogue-responder have the ability to request the termination of a dialogue after it
5628 has been established.

5629 The dialogue termination procedure is driven by the following events:

5630 - a MAP-CLOSE request primitive;

5631 - a TC-END indication primitive.

5632 15.3.1 Receipt of a MAP-CLOSE request

5633 On receipt of a MAP-CLOSE request primitive, the MAP PM shall issue a TC-END request primitive and, if applicable,
5634 return to idle the associated active SSMs. Note that if the release method parameter of the MAP-CLOSE request
5635 indicates "normal" the TC-END request primitive will trigger the transmission of components associated with any user
5636 specific request or response primitives which may have been issued after the last MAP-DELIMITER request.

5637 15.3.2 Receipt of a TC-END indication

5638 On receipt of a TC-END indication primitive, the MAP shall accept any component handling indication primitives and
5639 process them as described in subclause 15.6.

5640 Once all the received primitives have been processed, the MAP PM shall return to idle the associated SSMs and issue a
5641 MAP-CLOSE indication primitive.

5642 15.4 User Abort

5643 Both the dialogue-initiator and the dialogue-responder have the ability to abort a dialogue at any time.

5644 The user abort procedure is driven by one of the following events:

5645 - a MAP-U-ABORT request primitive;

5646 - a TC-U-ABORT indication primitive carrying a MAP-user-abort PDU.

5647 15.4.1 MAP-U-ABORT request

5648 On receipt of a MAP-U-ABORT request the MAP PM shall construct a MAP-user-abort PDU from the user-reason and
5649 diagnostic parameters and issue a TC-U-ABORT request primitive. All state machines associated with the dialogue are
5650 returned to idle.

5651 15.4.2 TC-U-ABORT ind

5652 On receipt of a TC-U-ABORT indication carrying a MAP-user-abort PDU, the MAP PM shall issue a MAP-U-ABORT
5653 indication primitive. The user-reason and diagnostic information elements are mapped to the corresponding parameters
5654 of the MAP-U-ABORT indication primitive.

5655 All state machines associated with the dialogue are returned to idle.

5656 15.5 Provider Abort

5657 The MAP has the ability to abort a dialogue at both the dialogue-initiator side and the dialogue-responder side.

5658 The provider abort procedure is driven by one of the following events:

5659 - a MAP PM error situation;

5660 - a TC-P-ABORT indication primitive;

5661 - a TC-U-ABORT indication primitive carrying a MAP-abort PDU.

5662 15.5.1 MAP PM error situation

5663 In the case of an abnormal situation detected at the MAP level during an established dialogue, the MAP PM shall:

- 5664 - issue a MAP-P-ABORT indication primitive with the appropriate value of the provider-reason parameter;
- 5665 - construct a MAP-abort PDU from the value of these parameters and request its transfer using a TC-U-ABORT
- 5666 request primitive.

5667 15.5.2 TC-P-ABORT ind

5668 On receipt of a TC-P-ABORT indication, the MAP PM shall issue a MAP-P-ABORT indication primitive.

5669 All state machines associated with the dialogue are returned to idle.

5670 15.5.3 TC-U-ABORT ind

5671 On receipt of a TC-U-ABORT indication carrying a MAP-abort PDU, the MAP PM shall issue a MAP-P-ABORT
5672 indication primitive, with the appropriate value of the provider-reason parameter. The source parameter shall indicate
5673 "MAP-provider".

5674 All state machines associated with the dialogue are returned to idle.

5675 15.6 Procedures for MAP specific services

5676 This subclause describes the MAP procedures for MAP specific services.

5677 These procedures are driven by the following types of events:

- 5678 - a MAP specific request or a MAP specific MAP response primitive;
- 5679 - a component handling primitive from TC.

5680 A Service State Machine is activated on receipt of one of the following signals:

- 5681 - a MAP request primitive, which activates a requesting SSM;
- 5682 - a TC-INVOKE indication primitive without linked identifier, which activates a responding SSM.

5683 For component handling primitives there are two types of events:

- 5684 - events which activate a Service State Machine or which can be related to an existing one;
- 5685 The procedure elements driven by these events are described in subclauses 15.6.1 to 15.6.4.
- 5686 - events which cannot be related to a Service State Machine.

5687 The procedure elements driven by these events are described in subclause 15.6.5.

5688 15.6.1 Service invocation

5689 The MAP specific procedures are initiated by the MAP request primitives.

5690 On receipt of a MAP request primitive, the MAP PM shall build an operation argument from the parameters received in
5691 the request primitive and request the invocation of the associated operation using the TC-INVOKE procedure. If a
5692 linked ID parameter is inserted in the primitive this indicates a child service and implies that the operation on which the
5693 service is mapped is linked to the operation on which the parent service is mapped.

5694 The mapping of MAP specific services on to remote operations is given in table 16.2/1.

5695 15.6.2 Service invocation receipt

5696 On receipt of a TC-INVOKE indication primitive, the MAP PM shall:

- 5697 - if the invoke ID is already in use by an active service, request the transfer of a reject component using the TC-U-
5698 REJECT request primitive with the appropriate problem code (duplicated invokeID) and issue a MAP-NOTICE
5699 indication primitive with a diagnostic parameter set to "abnormal event received from the peer";
- 5700 - if the operation code does not correspond to an operation supported by the application-context, request the
5701 transfer of a reject component using the TC-U-REJECT request primitive, with the appropriate problem code
5702 (unrecognized operation), and -if the dialogue version is lower than 3- issue a MAP-NOTICE indication
5703 primitive with a diagnostic parameter set to „abnormal event received from the peer“;
- 5704 - if a linked ID is included, perform the following checks: If the operation referred to by the linked ID does not
5705 allow linked operations or if the operation code does not correspond to a permitted linked operation, issue a TC-
5706 U-REJECT request primitive with the appropriate problem code (linked response unexpected or unexpected
5707 linked operation);
- 5708 - if the type of the argument is not the one defined for the operation, request the transfer of a reject component
5709 using the TC-U-REJECT request primitive, with the appropriate problem code (mistyped parameter), and issue a
5710 MAP-NOTICE indication primitive with a diagnostic parameter set to "abnormal event from the peer";
- 5711 - if the type of the argument is correct but the values of the information elements it contains do not permit the type
5712 of MAP service being invoked to be determined, request the transfer of an error component using the TC-U-
5713 ERROR request primitive with an error code set to "unexpected data value" and issue a MAP-NOTICE
5714 indication primitive with a diagnostic parameter set to "abnormal event from the peer";

5715 NOTE 1: These checks are only relevant when there is not a one-to-one mapping between a service and an
5716 operation.

- 5717 - if the type of the argument is correct but information elements required for the service being invoked are missing,
5718 request the transfer of an error component using the TC-U-ERROR request primitive with an error code set to
5719 "data missing" and issue a MAP-NOTICE indication primitive with a diagnostic parameter set to "abnormal
5720 event from the peer";

5721 NOTE 2: These checks are only relevant when there is not a one-to-one mapping between a service and an
5722 operation.

- 5723 - if the type of the argument is correct but contains information elements which are not relevant for the type of
5724 MAP service being invoked, request the transfer of an error component using the TC-U-ERROR request
5725 primitive with an error code set to "unexpected data value" and issue a MAP-NOTICE indication primitive with a
5726 diagnostic parameter set to "abnormal event from the peer";

5727 NOTE 3: These checks are only relevant when there is not a one-to-one mapping between a service and an
5728 operation.

- 5729 - Otherwise, issue the relevant MAP indication primitive to the MAP-service-user. If the service is to be user
5730 confirmed, the MAP PM waits for the corresponding response primitive.

5731 15.6.3 Service response

5732 For user confirmed services, the MAP PM shall accept a MAP response primitive and shall:

- 5733 - if no error indication is included in the primitive and the service maps on to a class 1 or 3 operation, construct a
5734 result information element from the parameters received and request its transfer using the TC-RESULT-L service
5735 and optionally the TC-RESULT-NL service.

5736 The TC-RESULT-NL services shall be used when the user specific parameters of the response primitives cannot be
5737 transferred in a single signalling frame and no segmenting mechanism is available from the underlying layers. The MAP
5738 PM shall issue one or several TC-RESULT-NL request primitives followed by a TC-RESULT-L primitive. The user
5739 parameters shall be split so that each portion contains sufficient information to construct a value compatible with the
5740 type defined for the result of the associated operation.

- 5741 - if no error indication is included in the primitive and the service response maps on to a class 4 linked operation,
 5742 construct an operation argument from the parameters received and request its transfer using the TC-INVOKE
 5743 service for this class 4 linked operation. The operation to be invoked is deduced from the value of the result
 5744 parameter of the service primitive;
- 5745 - if an error indication is included in the primitive and the service maps on to a class 1 or 2 operation, either issue a
 5746 TC-U-REJECT request primitive if the user error parameter indicates "resource limitation" or "initiating release",
 5747 or construct an error parameter from the parameters received and request its transfer using the TC-U-ERROR
 5748 request primitive. The error code should be the one associated with the value of the user error parameter of the
 5749 response primitive.
- 5750 NOTE: The only user errors that a MAP user can generate in addition to the list of errors attached to the operation
 5751 which is associated with the service are: resource limitation and initiating release. Any other abnormal
 5752 situation is detected either by the TC entity or by the MAP entity.
- 5753 - if an error indication is received and the operation maps on to a class 3 operation, or if no error indication is
 5754 received but the service maps on to a class 2 operation which has no class 4 linked operation, return the local
 5755 service state machine to idle without requesting any service from TC.

5756 15.6.4 Receipt of a response

5757 A component handling indication primitive is considered as driving a response for a confirmed service if the invoke ID
 5758 parameter value matches the one stored for the service, or if the linked ID parameter value matches the one stored for
 5759 the service and the operation invoked is a class 4 operation. On receipt of a response (except a TC-L-CANCEL
 5760 indication) for an unconfirmed service the MAP PM shall issue a MAP-NOTICE indication primitive with the
 5761 appropriate provider error (return result unexpected or return error unexpected).

5762 15.6.4.1 Receipt of a TC-RESULT-NL indication

5763 If the type of the partial result parameter is not compatible with the one defined for the complete result of this operation,
 5764 request the transfer of a reject component using the TC-U-REJECT request primitive, with the appropriate problem code
 5765 (mistyped parameter) and issue a confirm primitive with the provider error parameter set to "invalid response received".
 5766 The MAP PM shall also issue a TC-U-CANCEL request primitive so that all subsequent result components for this
 5767 operation are discarded by TC.

5768 Otherwise, store the value of the partial result parameter and wait for subsequent TC-RESULT-NL indication primitives
 5769 until a TC-RESULT-L indication primitive is received.

5770 15.6.4.2 Receipt of a TC-RESULT-L indication

5771 If the type of the result parameter is not the one defined for the result of this operation, request the transfer of a reject
 5772 component using the TC-U-REJECT request primitive, with the appropriate problem code (mistyped parameter), and
 5773 issue a confirm primitive with the provider error parameter set to "invalid response received".

5774 If the type of the result parameter is correct but does not contain all the information elements required by the service
 5775 associated with the invocation, issue a confirm primitive with the provider error parameter set to "invalid response
 5776 received".

5777 NOTE 1: These checks are only relevant when there is not a one-to-one mapping between a service and an
 5778 operation.

5779 If the type of the result parameter is correct but contains information elements which are not relevant for the service
 5780 associated with the invocation are missing, issue a confirm primitive with the provider error parameter set to "invalid
 5781 response received".

5782 NOTE 2: These checks are only relevant when there is not a one-to-one mapping between a service and an
 5783 operation.

5784 Otherwise, issue a MAP confirm primitive to the MAP-service-user mapping the result parameter of the TC-RESULT-L
 5785 primitive on to the MAP specific parameters.

5786 If partial results have been previously received, the value of the partial result parameters shall also be taken into account
5787 before performing the three previous checks.

5788 15.6.4.3 Receipt of a TC-U-ERROR indication

5789 If the error code is not defined for the MAP or is not one associated with the operation referred to by the invoke
5790 identifier, request the transfer of a reject component using the TC-U-REJECT request primitive, with the appropriate
5791 problem code (unrecognized error or unexpected error), and issue a confirm primitive with the provider error parameter
5792 set to "invalid response received".

5793 If the type of the error parameter is not the one defined for this error, request the transfer of a reject component using the
5794 TC-U-REJECT request primitive, with the appropriate problem code (mistyped parameter), and issue a confirm
5795 primitive with the provider error parameter set to "invalid response received".

5796 If the type of the error parameter is correct but does not contain all the information elements required by the service
5797 associated with the invocation, issue a confirm primitive with the provider error parameter set to "invalid response
5798 received".

5799 NOTE 1: In some cases, it may be necessary to analyse the operation argument.

5800 If the type of the error parameter is correct but its value includes information elements which are not relevant for the
5801 service associated with the invocation, issue a confirm primitive with the provider error parameter set to "invalid
5802 response received".

5803 NOTE 2: In some cases, it may be necessary to analyse the operation argument.

5804 Otherwise, issue a MAP confirm primitive to the MAP-service-user with the user error parameter set according to the
5805 received error code. If applicable the error parameter is mapped to the diagnostic parameter.

5806 15.6.4.4 Receipt of a TC-INVOKE indication

5807 A TC-INVOKE indication primitive is considered as carrying a possible response to a specific service if the linked ID
5808 refers to an active specific service and the associated operation is a class 4 operation. Note that the presence of a linked
5809 ID parameter in a TC-INVOKE primitive requesting a non class 4 operation indicates a child service whose procedures
5810 are the same as the procedures for the parent service.

5811 On receipt of a TC-INVOKE indication confirming an active service, the MAP PM shall:

5812 - if the operation code is not defined for MAP and the dialogue version is at least 3, issue a TC-U-REJECT request
5813 primitive with the appropriate problem code (unrecognized operation).

5814 - if the operation code is not defined for MAP and the dialogue version is lower than 3, or if the operation referred
5815 to by the linked ID does not allow linked operations or if the operation code does not correspond to an allowed
5816 linked operation, issue a TC-U-REJECT request primitive with the appropriate problem code (unrecognized
5817 operation, linked response unexpected or unexpected linked operation). If the service is confirmed, the MAP
5818 shall also issue a Confirm primitive with provider error indication "unexpected response from the peer",
5819 otherwise it may issue a MAP-NOTICE indication primitive with an appropriate diagnostic "abnormal event
5820 received from the peer".

5821 - otherwise issue a confirm primitive mapping the operation argument parameter to the user specific parameters
5822 and setting the result parameter according to the operation code of the linked operation.

5823 15.6.4.5 Receipt of a TC-U-REJECT indication

5824 On receipt of a TC-U-REJECT indication primitive which affects a pending service, the MAP PM shall issue a MAP
5825 confirm primitive to the MAP-service-user with the appropriate value of the provider error or user error parameter.

5826 The mapping of TC invoke problem codes on to MAP Provider Error and MAP User Error parameter values is
5827 described in clause 16.

5828 15.6.4.6 Receipt of a TC-L-REJECT indication

5829 This event occurs when the local TC detects a protocol error in an incoming component which affects an active specific
5830 service.

5831 On receipt of a TC-L-REJECT indicating "return result problem, unexpected return result", the MAP shall issue a
5832 confirm primitive with the parameter provider error indicating "unexpected response from the peer".

5833 On receipt of a TC-L-REJECT indicating "return error problem, unexpected error result", the MAP shall issue a confirm
5834 primitive with the parameter provider error indicating "unexpected response from the peer".

5835 Note that when the problem code indicates a general problem, it is considered that the event cannot be related to an
5836 existing SSM even if the invoke Id is provided by TC. This is because whether the invoke Id refers to a local or remote
5837 invocation is ambiguous. The behaviour of the MAP PM in such a case is described in subclause 15.6.5.3.

5838 15.6.4.7 Receipt of a TC-L-CANCEL indication

5839 On receipt of a TC-L-CANCEL indication, the MAP PM shall:

5840 - if the associated operation is a class 1 operation, issue a confirm primitive with the provider error cause
5841 indicating "no response from the peer";

5842 - if the associated operation is a class 2 operation and no linked operations are defined for this operation, issue a
5843 confirm primitive without parameter (i.e. indicating implicitly the successful completion of the service);

5844 - if the associated operation is a class 2 operation and has linked operations but none of them has been invoked,
5845 issue a confirm primitive with the provider error parameter indicating "service completion failure";

5846 - if the associated operation is a class 2 operation and a linked operation invocation has already been received in
5847 response to this operation, ignore the primitive;

5848 - if the associated operation is a class 3 operation, issue a confirm primitive with the provider error cause
5849 indicating "service completion failure";

5850 - if the associated operation is a class 4 operation, ignore the primitive.

5851 NOTE: When a TC-L-CANCEL ind primitive is received before the dialogue has been confirmed (i.e. no
5852 backward message is received by the dialogue initiator node), the MAP PM shall first issue a MAP-OPEN
5853 Cnf primitive with the result parameter indicating "accepted" (which means that the dialogue is considered
5854 as being implicitly accepted). Then, as indicated above, the TC-L-CANCEL Indication is interpreted
5855 according to the class of the operation to which it refers.

5856 15.6.4.8 Receipt of a TC-NOTICE indication

5857 If a TC-NOTICE indication primitive is received before the dialogue has been confirmed (i.e. no backward message is
5858 received by the dialogue initiator node), the MAP PM shall issue a MAP-OPEN Cnf primitive with the result parameter
5859 indicating Refused and a refuse reason Remote node not reachable".

5860 If a TC-NOTICE indication primitive is received after the dialogue has been confirmed, the MAP PM shall issue a
5861 MAP-NOTICE indication to the user, with a problem diagnostic indicating "message cannot be delivered to the peer".

5862 15.6.5 Other events

5863 This subclause describes the behaviour of the MAP PM on receipt of a component handling indication primitive which
5864 cannot be related to any service or which does not affect a pending one. The MAP user is only informed that an
5865 abnormal event occurred during the associated dialogue. It is up to the MAP user to abort, continue or terminate the
5866 dialogue.

5867 15.6.5.1 Receipt of a TC-U-REJECT

5868 On receipt of a TC-U-REJECT indication primitive which does not affect an active SSM (i.e. indicating a return result
5869 or return error problem), the MAP PM shall issue a MAP-NOTICE indication primitive with the diagnostic parameter
5870 set to "response rejected by the peer".

5871 This is also applicable for invoke problems related to a class 4 linked operation.

5872 15.6.5.2 Receipt of a TC-R-REJECT indication

5873 On receipt of a TC-R-REJECT indication (i.e. when a protocol error has been detected by the peer TC entity) which
5874 does not affect an active SSM, the MAP PM shall either discard this indication or issue a MAP-NOTICE indication
5875 primitive with the provider error indicating "abnormal event detected by the peer".

5876 In case of notification, it is up to the MAP user to continue, abort or terminate the dialogue. Note also that for MAP V1
5877 the reject component is received in an END message and therefore the dialogue is terminated anyway.

5878 15.6.5.3 Receipt of a TC-L-REJECT indication

5879 On receipt of a TC-L-REJECT indication primitive (i.e. when a protocol error has been detected by the local TC entity)
5880 which cannot be related to an active SSM, the MAP PM shall either discard this indication or issue a MAP-NOTICE
5881 indication primitive with the provider error indicating "abnormal event received from the peer".

5882 In case of notification, it is up to the MAP user to continue, or to terminate the dialogue and implicitly trigger the
5883 transmission of the reject component or to abort the dialogue.

5884 15.6.6 Parameter checks

5885 As described in the previous subclauses, the MAP PM performs a set of checks to ensure the correctness of the
5886 information elements received; these are:

5887 - check if the syntax and encoding (note) of the operation argument, result or error parameter are correct.

5888 NOTE: Depending on the implementation, encoding problems on the TC user portion may be detected at TC level
5889 or by the MAP user. In the second case the problem is reported in a similar manner to a syntactical
5890 problem.

5891 The syntax shall be considered incorrect if a mandatory information element is missing in any constructed
5892 element or if the value of an information element is out of the range defined for the type it is supposed to belong
5893 to;

5894 - if there is not a one-to-one mapping between a service and an operation:

5895 i) check if the value of the information elements (generally a single one) permits the MAP PM to determine the
5896 service associated with the operation invocation;

5897 ii) check that there are no information elements which are irrelevant for the indication or a confirm primitive to
5898 be issued;

5899 - check if all the information elements required to built an indication or a confirm primitive are available.

5900 However some additional checks may have to be performed by the MAP user (see clause 18).

5901 15.6.7 Returning state machines to idle

5902 Unlike TC invocation state machines, service state machines exist at both requestor and performer side.

5903 A service state machine at the requestor side is returned to idle when the MAP-specific confirm primitive is issued or
5904 when the dialogue terminates.

5905 A service state machine at the performer side is returned to idle on receipt of a MAP-specific response primitive from
5906 the MAP user, when the dialogue terminates or at expiry of an implementation dependent watch-dog timer which is
5907 started when the state machine is created.

5908 15.6.8 Load control

5909 As stated in the previous subclauses, before issuing a MAP-OPEN indication primitive the MAP PM performs a check
5910 to verify if there are sufficient resources to open the dialogue taking into account possible overload conditions.

5911 The decision is based on the priority allocated to the application-context whose name is explicitly included in the TC-
5912 BEGIN indication primitive or implied by the first operation invocation when V1 contexts are in use. How a V1
5913 application-context-name is derived from an operation code is described in table 15.1/1.

5914 The priority level allocated to each application-context is described in clause 3 tables 5.1/1 and 5.1/2.

5915 16 Mapping on to TC services

5916 16.1 Dialogue control

5917 Dialogue control services are mapped to TC dialogue handling services. The TC-UNI service is not used by the MAP
5918 PM.

5919 16.1.1 Directly mapped parameters

5920 The following parameters of the MAP-OPEN request and indication primitives are directly mapped on to the
5921 corresponding parameters of the TC-BEGIN primitives:

5922 - destination address;

5923 - originating address.

5924 16.1.2 Use of other parameters of dialogue handling primitives

5925 16.1.2.1 Dialogue Id

5926 The value of this parameter is associated with the MAP PM invocation in an implementation dependent manner.

5927 16.1.2.2 Application-context-name

5928 The application-context-name parameter of a MAP primitive is mapped to the application-context-name parameter of
5929 TC dialogue handling primitives according to the rules described in subclause 15.1.

5930 16.1.2.3 User information

5931 The user information parameter of TC dialogue primitives is used to carry the MAP dialogue APDUs.

5932 16.1.2.4 Component present

5933 This parameter is used by the MAP PM as described in CCITT Recommendation Q.771. It is not visible to the MAP
5934 user.

5935 16.1.2.5 Termination

5936 The value of this parameter of the TC-END request primitive is set by the MAP PM on the basis of the release method
5937 parameter of the MAP-CLOSE request primitive, except when the dialogue state machine is in the state DIALOGUE
5938 INITIATED, in which case the Termination parameter shall always indicate "pre-arranged end".

5939 16.1.2.6 P-Abort-Cause

5940 Values of the P-abort-cause parameter are mapped to the values of the provider-reason parameter of the MAP-P-
 5941 ABORT indication primitive according to table 16.1/1, except in the dialogue initiated phase for the
 5942 "incorrectTransactionPortion" and "noCommonDialoguePortion" values which are mapped to the "potential
 5943 incompatibility problem" value of the refuse-reason parameter of the MAP-OPEN cnf primitive. The source parameter
 5944 in the MAP-P-ABORT ind takes the value "TC problem".

5945 16.1.2.7 Quality of service

5946 The quality of service of TC request primitives is set by the MAP as shown below.

5947 - Return option: "Return message on error" or "Discard message on error" as required by the network operator;

5948 - Sequence control: "Sequence guaranteed" or "Sequence result not guaranteed" as required by the network
 5949 operator;

5950 "Sequence guaranteed" shall be used when a segmented result is to be transferred (e.g. subscriber data in
 5951 response to SendParameters). It may also be appropriate to use Sequence guaranteed when a series of
 5952 InsertSubscriberData, ProcessAccessSignalling or ForwardAccessSignalling operations is used.

5953 It is essential that the TC message which indicates acceptance of a dialogue opening request is received by the dialogue
 5954 initiator before any subsequent message in that dialogue; otherwise the dialogue opening will fail. The dialogue
 5955 responder shall ensure that this requirement is met by:

5956 - Sending the dialogue acceptance message in a TC-END, if the dialogue structure requires it; or

5957 - Using "Sequence guaranteed", if the dialogue acceptance message is sent in a TC-CONTINUE; or

5958 - Waiting until the dialogue acceptance message has been acknowledged by the dialogue initiator before sending a
 5959 subsequent message, if the dialogue acceptance message is sent in a TC-CONTINUE.

5960 **Table 16.1/1: Mapping of P-Abort cause in TC-P-ABORT indication on to provider-reason in MAP-P-
 5961 ABORT indication**

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

5962

5963 NOTE: Or version incompatibility in the dialogue initiated phase.

5964 16.2 Service specific procedures

5965 Specific services are mapped to TC component handling services.

5966 16.2.1 Directly mapped parameters

5967 The Invoke Id parameter of the MAP request and indication primitive is directly mapped on to the Invoke Id parameter
 5968 of the component handling primitives.

5969 16.2.2 Use of other parameters of component handling primitives

5970 16.2.2.1 Dialogue Id

5971 The value of this parameter is associated with the MAP PM invocation in an implementation dependent manner.

5972 16.2.2.2 Class

5973 The value of this parameter is set by the MAP PM according to the type of the operation to be invoked.

5974 16.2.2.3 Linked Id

5975 When a service response is mapped to a class 4 operation, the value of this parameter is set by the MAP PM and
5976 corresponds to the value assigned by the user to the initial service request (i.e. the value of the invoke ID parameter of
5977 the request primitive). Otherwise if such a parameter is included in MAP request/indication primitives it is directly
5978 mapped to the linked ID parameter of the associated TC-INVOKE request/indication primitives.

5979 16.2.2.4 Operation

5980 When mapping a request primitive on to a Remote Operations PDU (invoke), the MAP PM shall set the operation code
5981 according to the mapping described in table 16.2/1.

5982 When mapping a response primitive on to a Remote Operations service, the MAP PM shall set the operation code of the
5983 TC-RESULT-L/NL primitive (if required) to the same value as the one received at invocation time.

5984

Table 16.2/1: Mapping of MAP specific services on to MAP operations

MAP-SERVICE	operation
MAP-ACTIVATE-SS	activateSS
MAP-ACTIVATE-TRACE-MODE	activateTraceMode
MAP-ALERT-SERVICE-CENTRE	alertServiceCentre
MAP-ANY-TIME-INTERROGATION	anyTimeInterrogaton
MAP-ANY-TIME-MODIFICATION	anyTimeModification
MAP-ANY-TIME-SUBSCRIPTION-INTERROGATION	anyTimeSubscriptionInterrogaton
MAP-CANCEL-LOCATION	cancelLocation
MAP-CHECK-IMEI	checkIMEI
MAP-DEACTIVATE-SS	deactivateSS
MAP-DEACTIVATE-TRACE-MODE	deactivateTraceMode
MAP-DELETE-SUBSCRIBER-DATA	deleteSubscriberData
MAP-ERASE-CC-ENTRY	eraseCC-Entry
MAP-ERASE-SS	eraseSS
MAP-FAILURE-REPORT	failureReport
MAP-FORWARD-ACCESS-SIGNALLING	forwardAccessSignalling
MAP-FORWARD-CHECK-SS-INDICATION	forwardCheckSsIndication
MAP-FORWARD-GROUP-CALL-SIGNALLING	forwardGroupCallSignalling
MAP-MT-FORWARD-SHORT-MESSAGE	mt-forwardSM
MAP-MO-FORWARD-SHORT-MESSAGE	mo-forwardSM
MAP-GET-PASSWORD	getPassword
MAP-INFORM-SERVICE-CENTRE	informServiceCentre
MAP-INSERT-SUBSCRIBER-DATA	insertSubscriberData
MAP-INTERROGATE-SS	interrogateSs
MAP-IST-ALERT	istAlert
MAP-IST-COMMAND	istCommand
MAP-NOTE-MS-PRESENT-FOR-GPRS	noteMsPresentForGprs
MAP-NOTE-SUBSCRIBER-DATA-MODIFIED	noteSubscriberDataModified
MAP-PREPARE-GROUP-CALL	prepareGroupCall
MAP-PREPARE-HANDOVER	prepareHandover
MAP-PREPARE-SUBSEQUENT-HANDOVER	prepareSubsequentHandover
MAP-PROCESS-ACCESS-SIGNALLING	processAccessSignalling
MAP-PROCESS-GROUP-CALL-SIGNALLING	processGroupCallSignalling
MAP-PROCESS-UNSTRUCTURED-SS-REQUEST	processUnstructuredSS-Request
MAP-PROVIDE-ROAMING-NUMBER	provideRoamingNumber
MAP-PROVIDE-SIWFS-NUMBER	provideSIWFSNumber
MAP-PROVIDE-SUBSCRIBER-LOCATION	provideSubscriberLocation
MAP-PROVIDE-SUBSCRIBER-INFO	provideSubscriberInfo
MAP-PURGE-MS	purgeMS
MAP-READY-FOR-SM	readyForSM
MAP-REGISTER-CC-ENTRY	registerCC-Entry
MAP-REGISTER-PASSWORD	registerPassword
MAP-REGISTER-SS	registerSS
MAP-REMOTE-USER-FREE	remoteUserFree
MAP-REPORT-SM-DELIVERY-STATUS	reportSmDeliveryStatus
MAP-RESET	reset
MAP-RESTORE-DATA	restoreData
MAP-SEND_GROUP-CALL_END_SIGNAL	sendGroupCallEndSignal
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-INFO-FOR-GPRS	sendRoutingInfoForGprs
MAP-SEND-ROUTING-INFO-FOR-LCS	sendRoutingInfoForLCS
MAP-SEND-ROUTING-INFORMATION	sendRoutingInfo
MAP-SET-REPORTING-STATE	setReportingState
MAP-SIWFS-SIGNALLING-MODIFY	SIWFSsignallingModify
MAP-STATUS-REPORT	statusReport
MAP-SUBSCRIBER-LOCATION-REPORT	subscriberLocationReport
MAP-SUPPLEMENTARY-SERVICE-INVOCATION-NOTIFICATION	ss-Invocation-Notification
MAP-UNSTRUCTURED-SS-NOTIFY	unstructuredSS-Notify

MAP-UNSTRUCTURED-SS-REQUEST	unstructuredSS-Request
MAP-UPDATE-GPRS-LOCATION	updateGprsLocation
MAP-UPDATE-LOCATION	updateLocation
MAP-NOTE-MM-EVENT	NoteMM-Event

5985

5986 **16.2.2.5 Error**

5987 The error parameter in a TC-U-ERROR indication primitive is mapped to the user error parameter in the MAP confirm
5988 primitive of the service associated with the operation to which the error is attached.

5989 The user error parameter in MAP response primitives is mapped to the error parameter of the TC-U-ERROR request
5990 primitive, except for "initiating-release" and "resource-limitation" which are mapped to the problem code parameter of
5991 the TC-U-REJECT request primitive.

5992 **16.2.2.6 Parameters**

5993 The parameters of MAP specific request and indication primitives are mapped to the argument parameter of TC-
5994 INVOKE primitives.

5995 The parameters of MAP specific response and confirm primitives are mapped to the result parameter of TC-RESULT-L
5996 primitives, the parameter of TC-U-ERROR primitives or the argument of TC-INVOKE primitives when mapping on
5997 linked class 4 operations is used.

5998 **16.2.2.7 Time out**

5999 The value of this parameter is set by the MAP PM according to the type of operation invoked.

6000 **16.2.2.8 Last component**

6001 This parameter is used by the MAP PM as described in CCITT Recommendation Q.711. It is not visible from the MAP
6002 user.

6003 **16.2.2.9 Problem code**6004 **16.2.2.9.1 Mapping to MAP User Error**

6005 The following values of the user error parameter are mapped as follows to values of the TC problem code parameter.
6006 These values are generated by the MAP user. This mapping is valid from the TC-U-REJECT indication primitive to the
6007 MAP confirm service primitive and from the MAP response service primitive to the TC-U-REJECT request primitive.

6008 **Table 16.2/2: Mapping of MAP User Error parameter on to TC problem code in TC-U-REJECT**
6009 **primitives**

MAP User Error	TC problem code
resource limitation	resource limitation
initiating release	initiating release

6010

6011 **16.2.2.9.2 Mapping to MAP Provider Error parameter**

6012 The following values of the TC problem code parameter of the TC-U-REJECT indication primitive are mapped as
6013 follows to values of the MAP Provider Error parameter of the MAP confirm primitive.

6014 **Table 16.2/3: Mapping of TC problem code in TC-U-REJECT on to MAP Provider Error parameter**

TC problem code	MAP Provider Error
duplicated invoke Id	duplicated invoke id
unrecognized operation	service not supported
mistyped parameter	mistyped parameter

6015

6016 The following values of the problem code parameters of the TC-L-REJECT primitive are mapped to values of the
6017 provider error parameter of the MAP confirm primitive as follows:

6018 **Table 16.2/4: Mapping of TC problem code in TC-L-REJECT on to MAP Provider Error parameter**

TC problem code	MAP Provider Error
return result unexpected	unexpected response from the peer
return error unexpected	unexpected response from the peer

6019

6020 **16.2.2.9.3 Mapping to diagnostic parameter**

6021 The following values of the problem code parameter of the TC-R-REJECT and TC-U-REJECT primitive are mapped to
6022 values of the diagnostic parameter of the MAP-NOTICE indication primitive as follows:

6023 **Table 16.2/5: Mapping of TC problem code of TC-R-REJECT and TC-U-REJECT on to diagnostic
6024 parameter**

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

6025

6026 The following values of the problem code parameter of the TC-L-REJECT primitive are mapped to values of the
6027 diagnostic parameter of the MAP-NOTICE indication primitive as follows:

6028 **Table 16.2/6: Mapping of TC problem code of TC-L-REJECT on to diagnostic parameter**

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

6029

6030 16.3 SDL descriptions

6031 The following SDL specification describes a system which includes three blocks: MAP-user, MAP-provider and TC.

6032 Such a system resides in each network component supporting MAP and communicates with its peers via the lower layers
6033 of the signalling network which are part of the environment.

6034 Only the MAP-provider is fully described in this subclause. The various type of processes which form the MAP-User
6035 block and the TC block are described respectively in clauses 18 to 25 of the present document and in CCITT
6036 Recommendation Q.774.

6037 The MAP-Provider block communicates with the MAP_USER via two channels U1 and U2. Via U1 the MAP-provider
6038 receives the MAP request and response primitives. Via U2 it sends the MAP indication and confirm primitives.

6039 The MAP-Provider block communicates with TC via two channels P1 and P2. Via P1 the MAP-Provider sends all the
6040 TC request primitives. Via P2 it receives all the TC indication primitives.

6041 The MAP-Provider block is composed of the four following types of processes:

6042 a) MAP_DSM: This type of process handles a dialogue. There exists one process instance per MAP dialogue.

6043 b) LOAD_CTRL: This type of process is in charge of load control. There is only one instance of this process in
6044 each system.

6045 c) PERFORMING_MAP_SSM: This type of process handle a MAP service performed during a dialogue. An
6046 instance of this process is created by the instance of the MAP_DSM process for each MAP-service to be
6047 performed.

6048 d) REQUESTING_MAP_SSM: This type of process handle a MAP service requested during a dialogue. An
6049 instance of this process is created by the instance of the MAP_DSM process for each requested MAP-service.

6050 A process MAP_DSM exchanges external signals with other blocks as well as internal signals with the other processes
6051 of the MAP-Provider block. The external signals are either MAP service primitives or TC service primitives.

6052 The signal routes used by the various processes are organized as follows:

6053 a) A process MAP_DSM receives and sends events from/to the MAP_user via signal route User1/User2. These
6054 routes uses respectively channel U1 and U2.

6055 b) A process MAP_DSM receives and sends events from/to the TC via signal route Tc1/Tc2. These routes uses
6056 respectively channel P1 and P2.

6057 c) A process MAP_DSM receives and sends events from/to the LOAD_CTRL process via signal route
6058 Load1/Load2. These routes are internal.

6059 d) A process MAP_DSM sends events to the PERFORMING_MAP_SSM processes via signal route Intern1.
6060 This route is internal.

6061 e) A process MAP_DSM sends events to the REQUESTING_MAP_SSM processes via signal route Intern2.
6062 This route is internal.

6063 f) A process MAP_PERFORMING_SSM sends events to the MAP_USER via signal route User4. This route
6064 uses channel U2.

6065 g) A process MAP_PERFORMING_SSM sends events to TC via signal route Tc3. This route uses channel P1.

6066 h) A process MAP_REQUESTING_SSM sends events to the MAP_USER via signal route User5. This route
6067 uses channel U2.

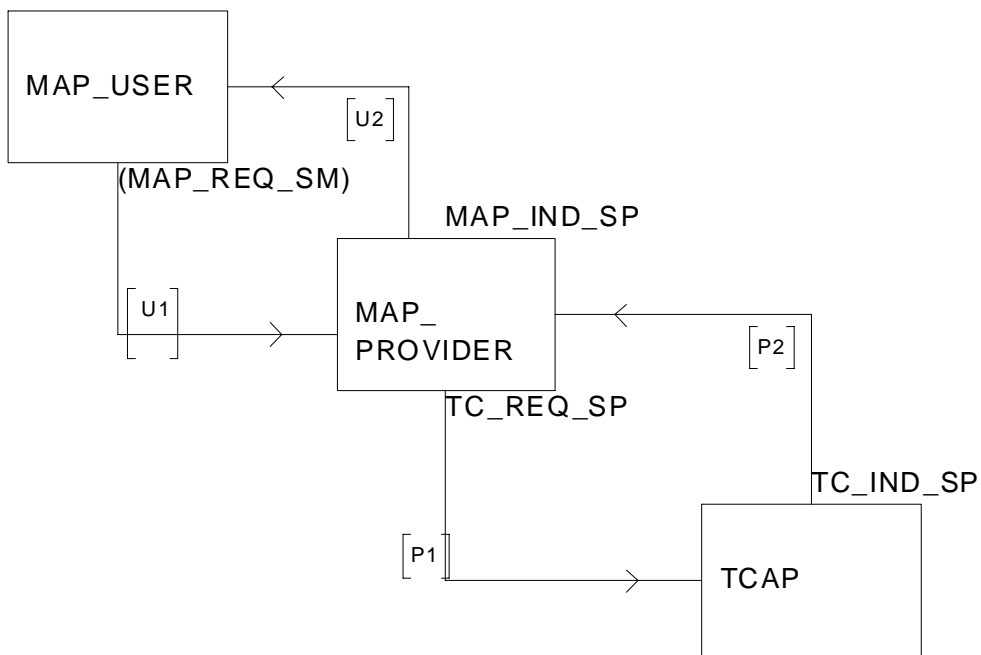
6068 j) A process MAP_REQUESTING_SSM sends events to TC via signal route Tc4. This route uses channel P1.

09.02 version 6.6.0

System MAP_STACK

16.2_1(1)

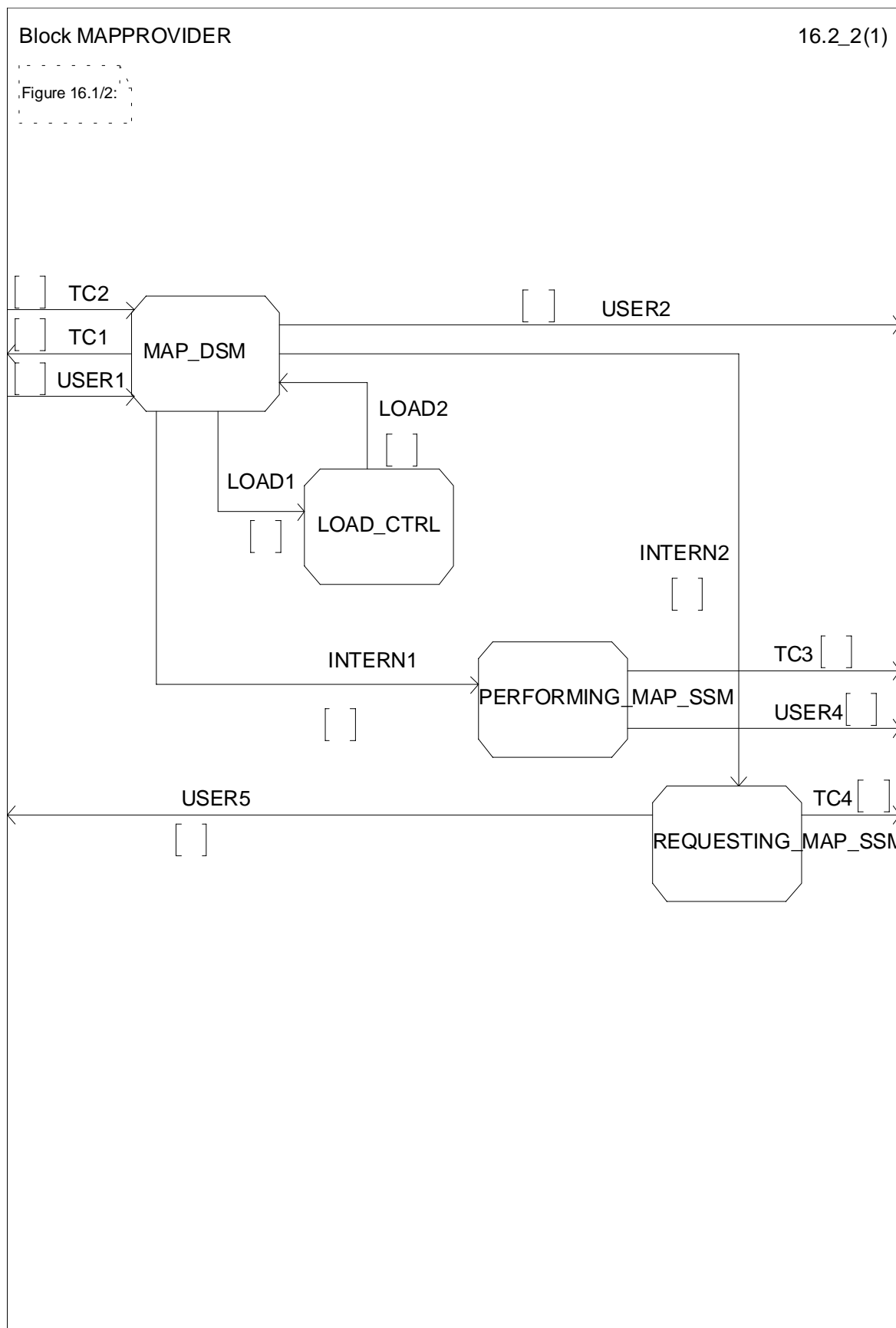
Figure 16.2/1:



6069

6070

Figure 16.2/1: System MAP_STACK



6071

6072

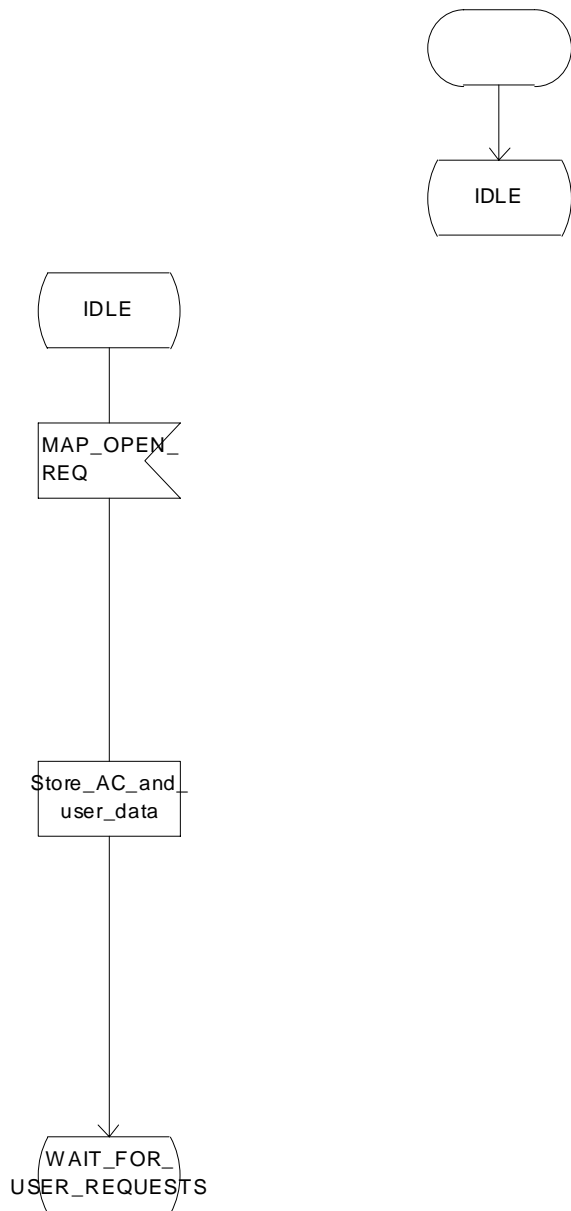
Figure 16.2/2: Block MAPPROVIDER

Process MAP_DSM

16.2_3.1(11)

Figure 16.2/3:

Comment 'MAP Dialoges State Maschine':
DCL
COMPONENTS_PRESENT, INVOKEID_ACTIVE, LAST_COMPONENT, OP_EXIST BOOLEAN,
OP_CODE INTEGER;



6073

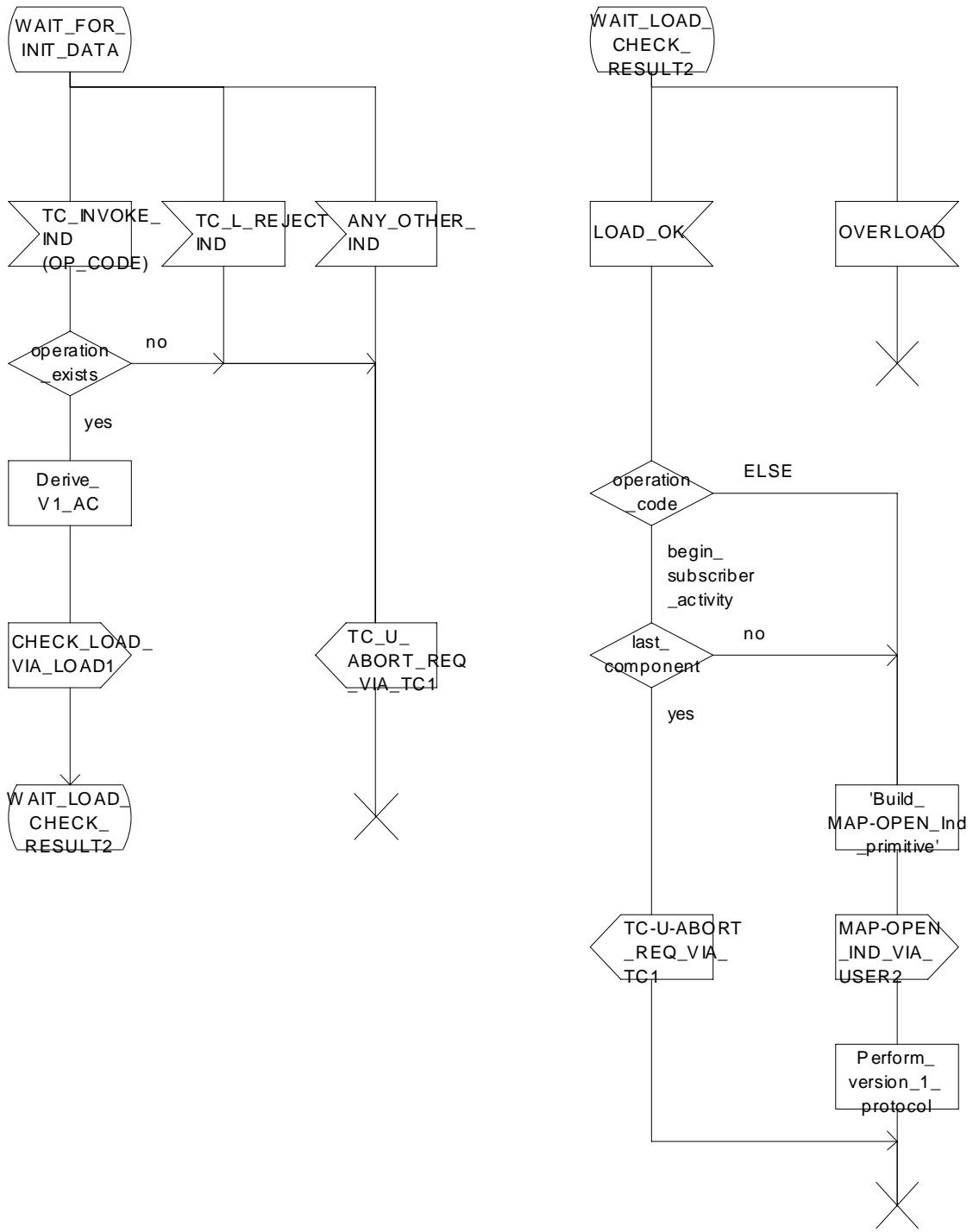
6074

Figure 16.2/3 (sheet 1 of 11): Process MAP_DSM

Process MAP_DSM

16.2_3.2(11)

Figure 16.2/3:



6075

6076

Figure 16.2/3 (sheet 2 of 11): Process MAP_DSM

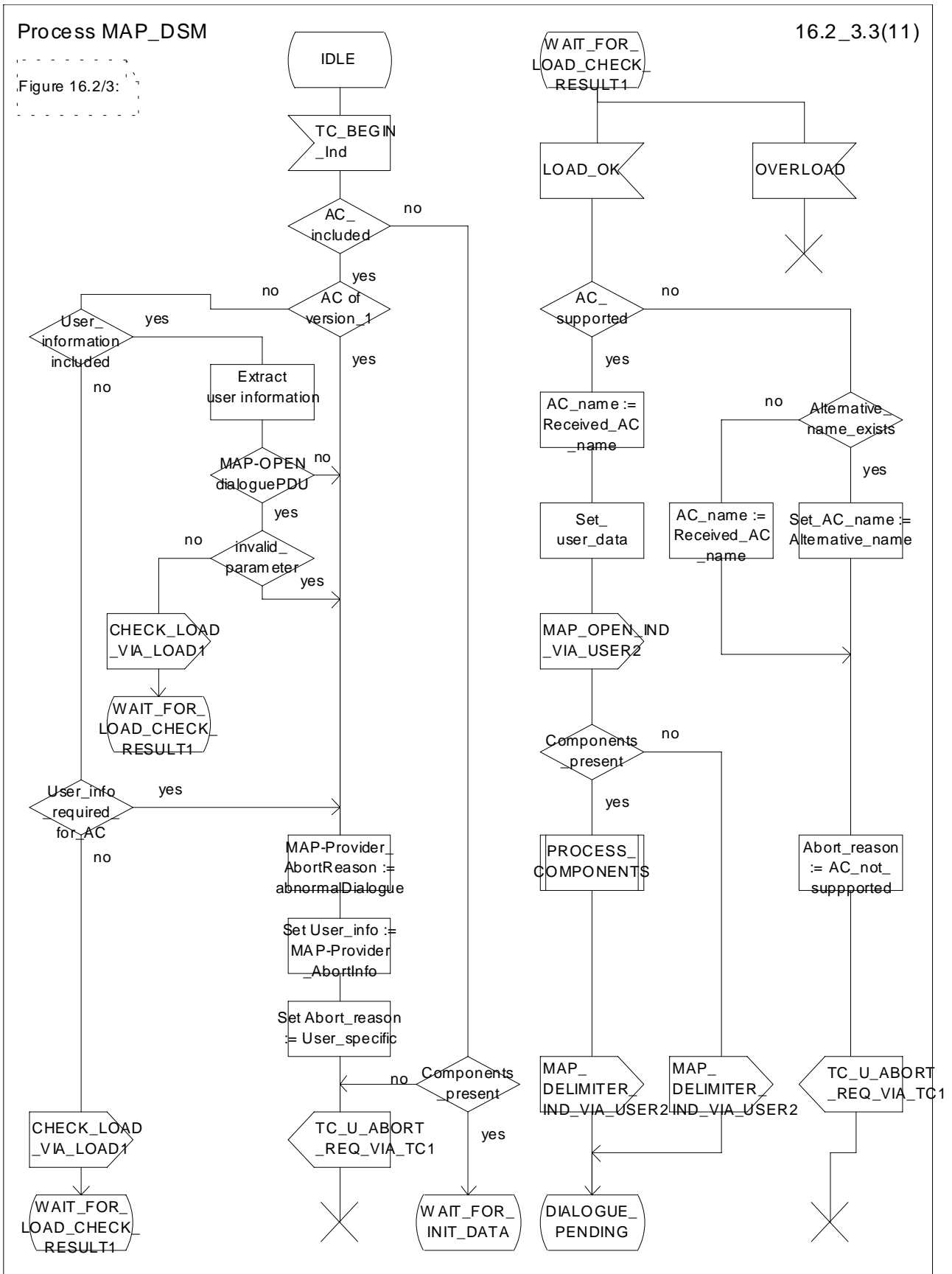
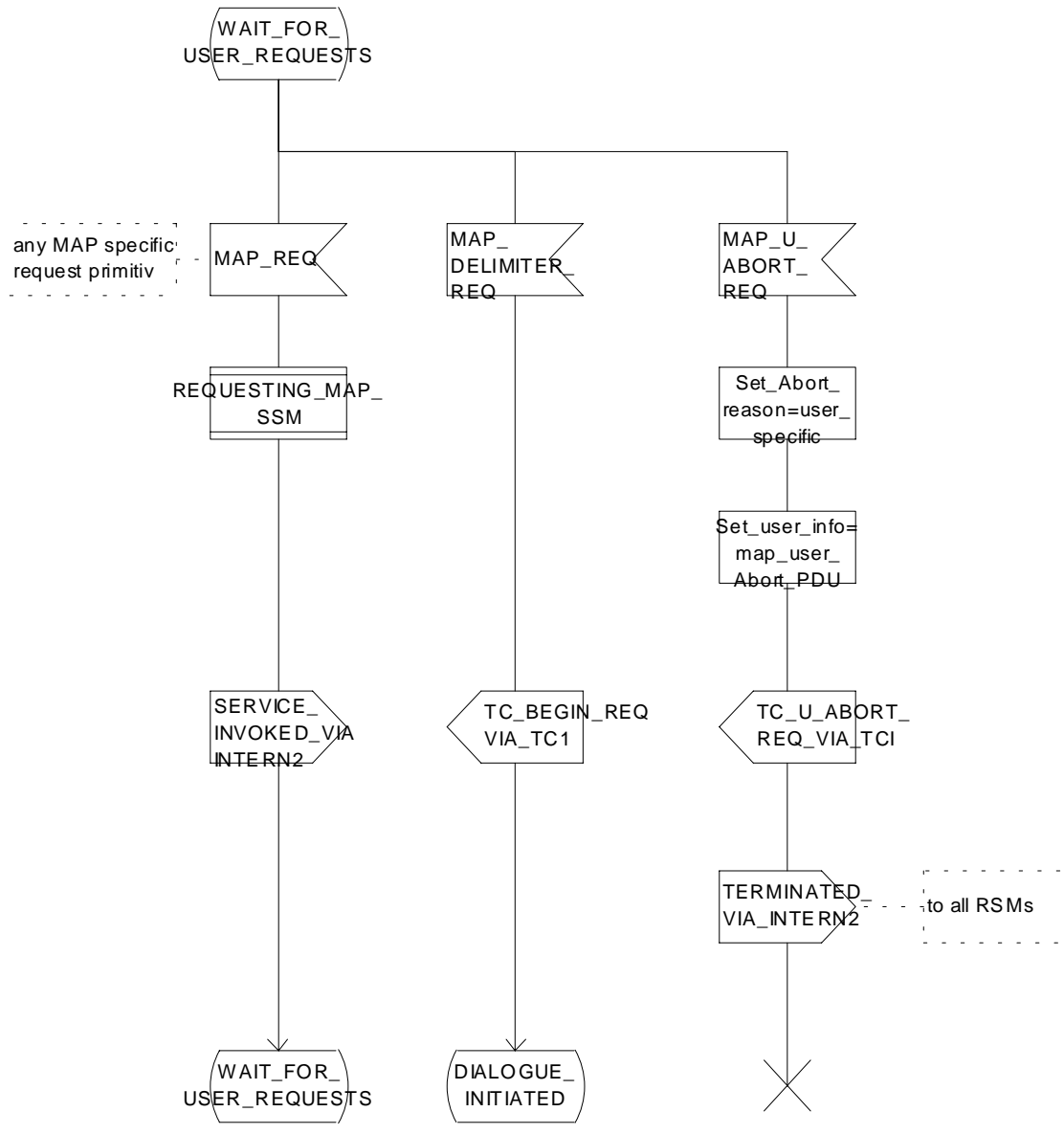


Figure 16.2/3 (sheet 3 of 11): Process MAP_DSM

Process MAP_DSM

16.2_3.4(11)

Figure 16.2/3:



6079

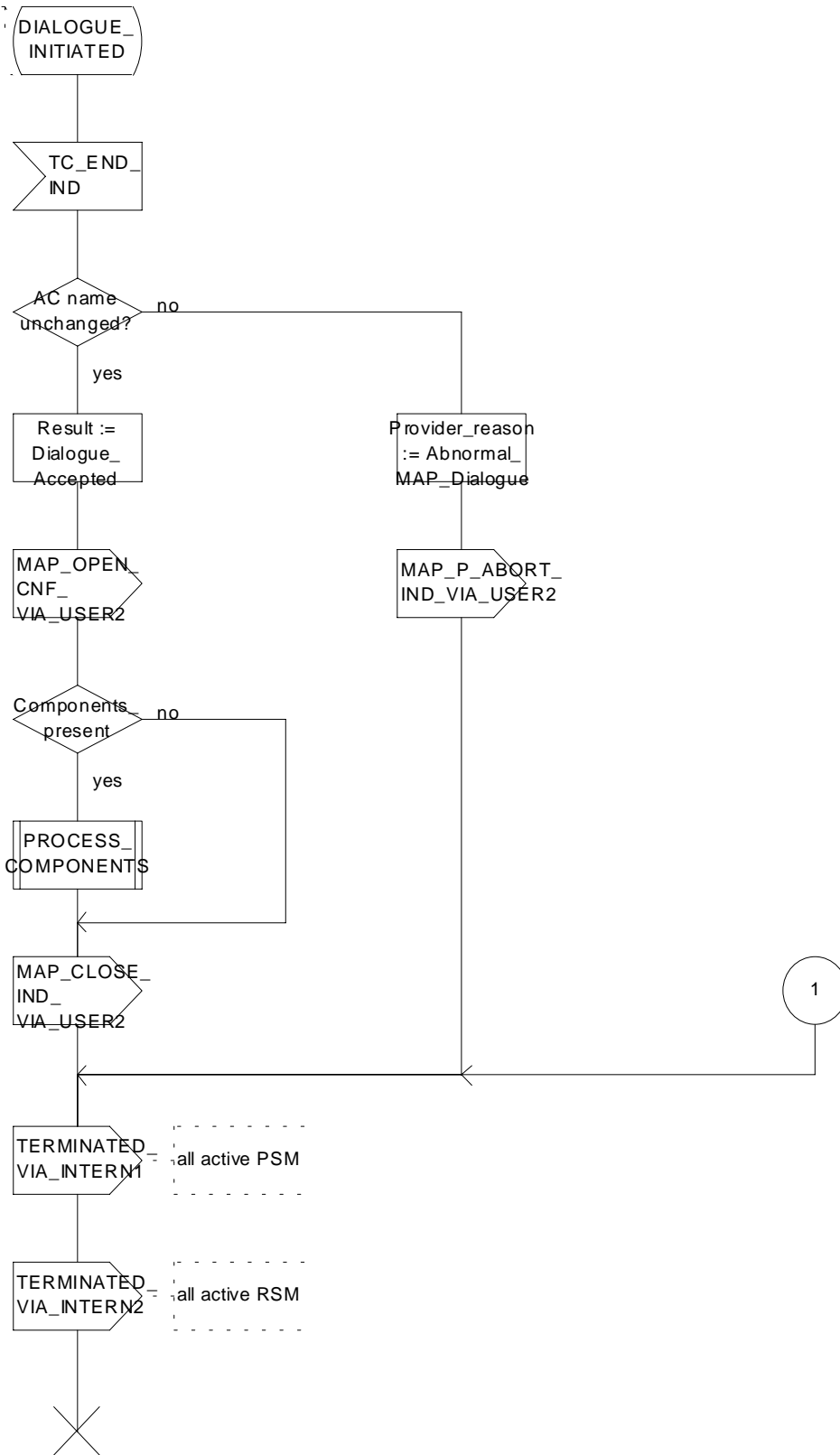
6080

Figure 16.2/3 (sheet 4 of 11): Process MAP_DSM

Process MAP_DSM

16.2_3.5(11)

Figure 16.2/3:



6081

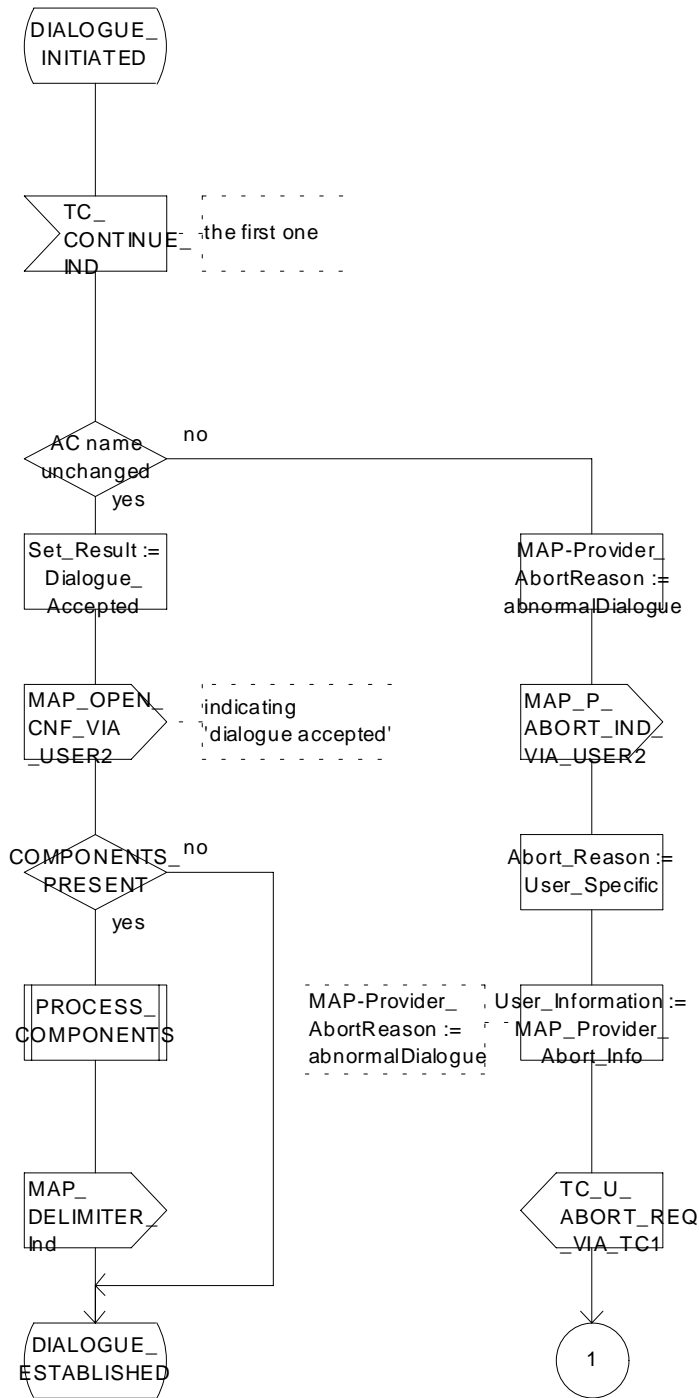
6082

Figure 16.2/3 (sheet 5 of 11): Process MAP_DSM

Process MAP_DSM

16.2_3.6(11)

Figure 16.2/3:



6083

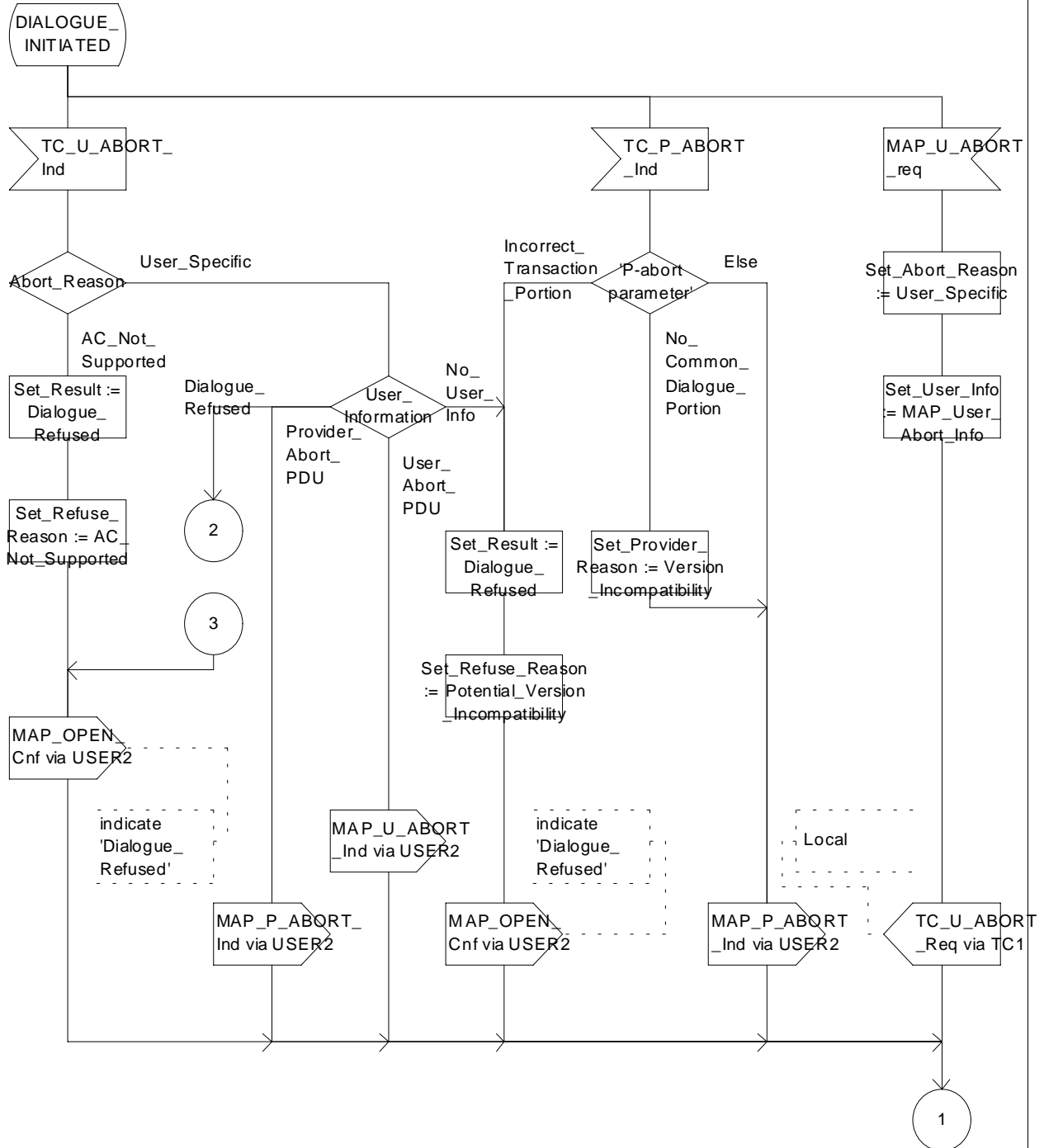
6084

Figure 16.2/3 (sheet 6 of 11): Process MAP_DSM

Process MAP_DSM

16.2_3.7(11)

Figure 16.2/3:



6085

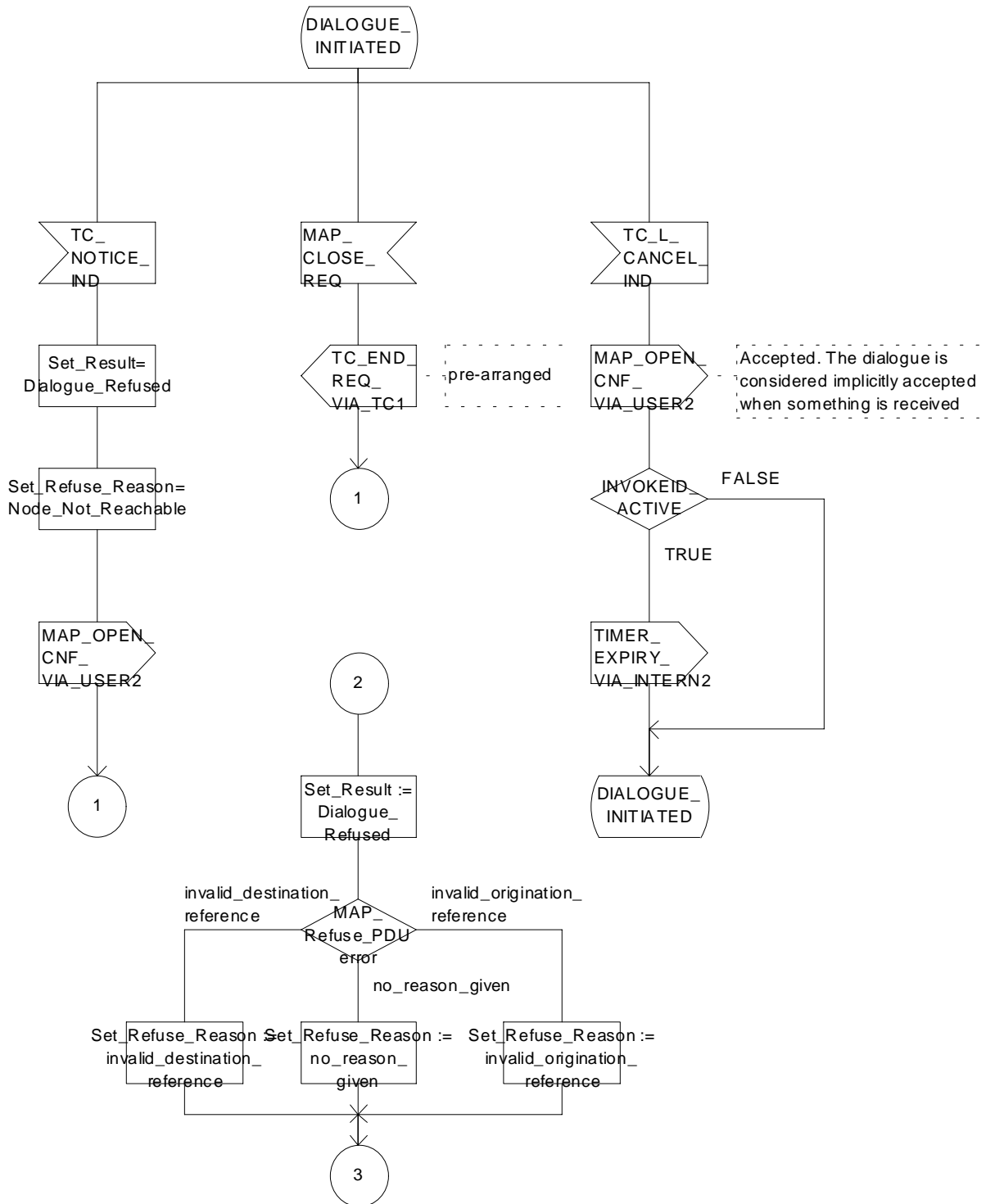
6086

Figure 16.2/3 (sheet 7 of 11): Process MAP_DSM

Process MAP_DSM

16.2_3.8(11)

Figure 16.2/3:



6087

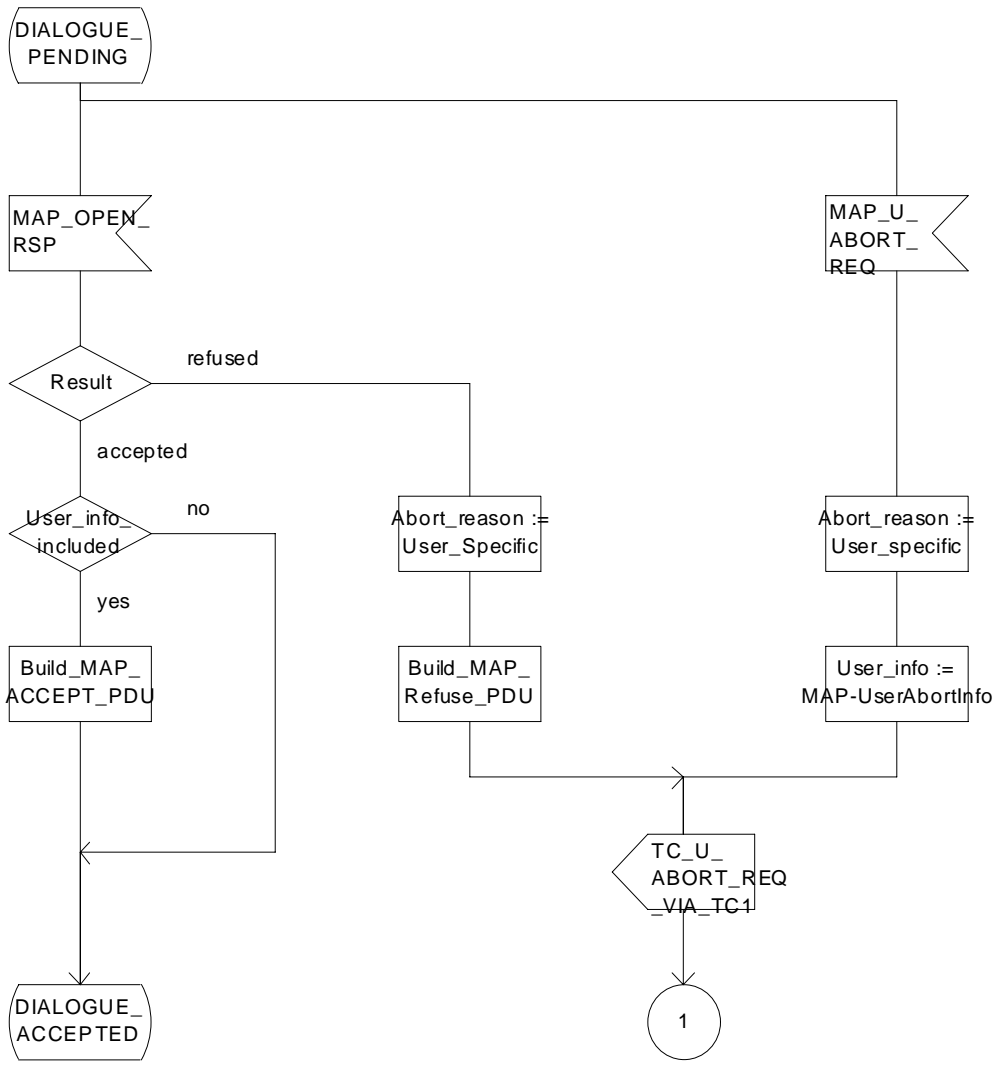
6088

Figure 16.2/3 (sheet 8 of 11): Process MAP_DSM

Process MAP_DSM

16.2_3.9(11)

Figure 16.2/3:



6089

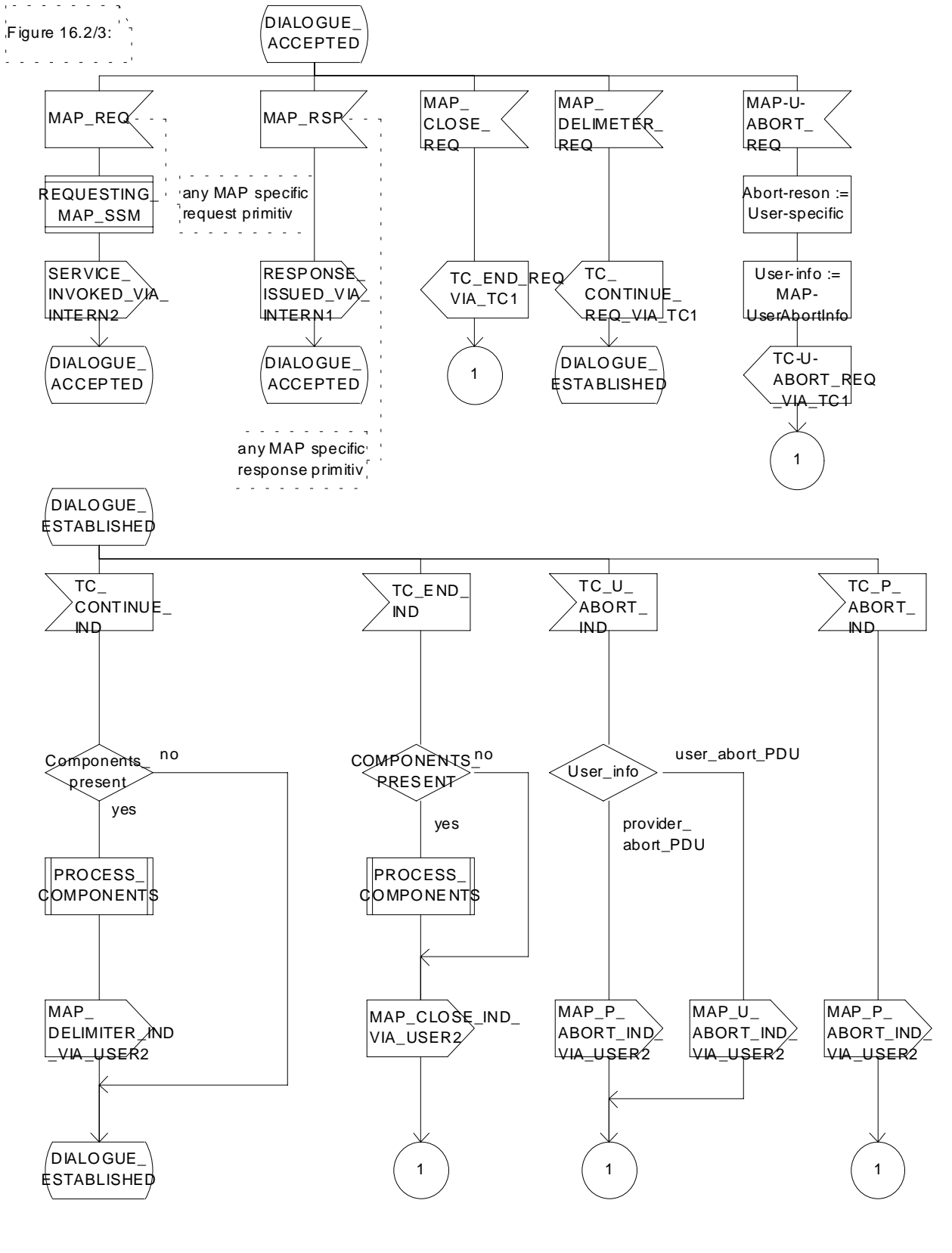
6090

Figure 16.2/3 (sheet 9 of 11): Process MAP_DSM

Process MAP_DSM

16.2_3.10(11)

Figure 16.2/3:



6091

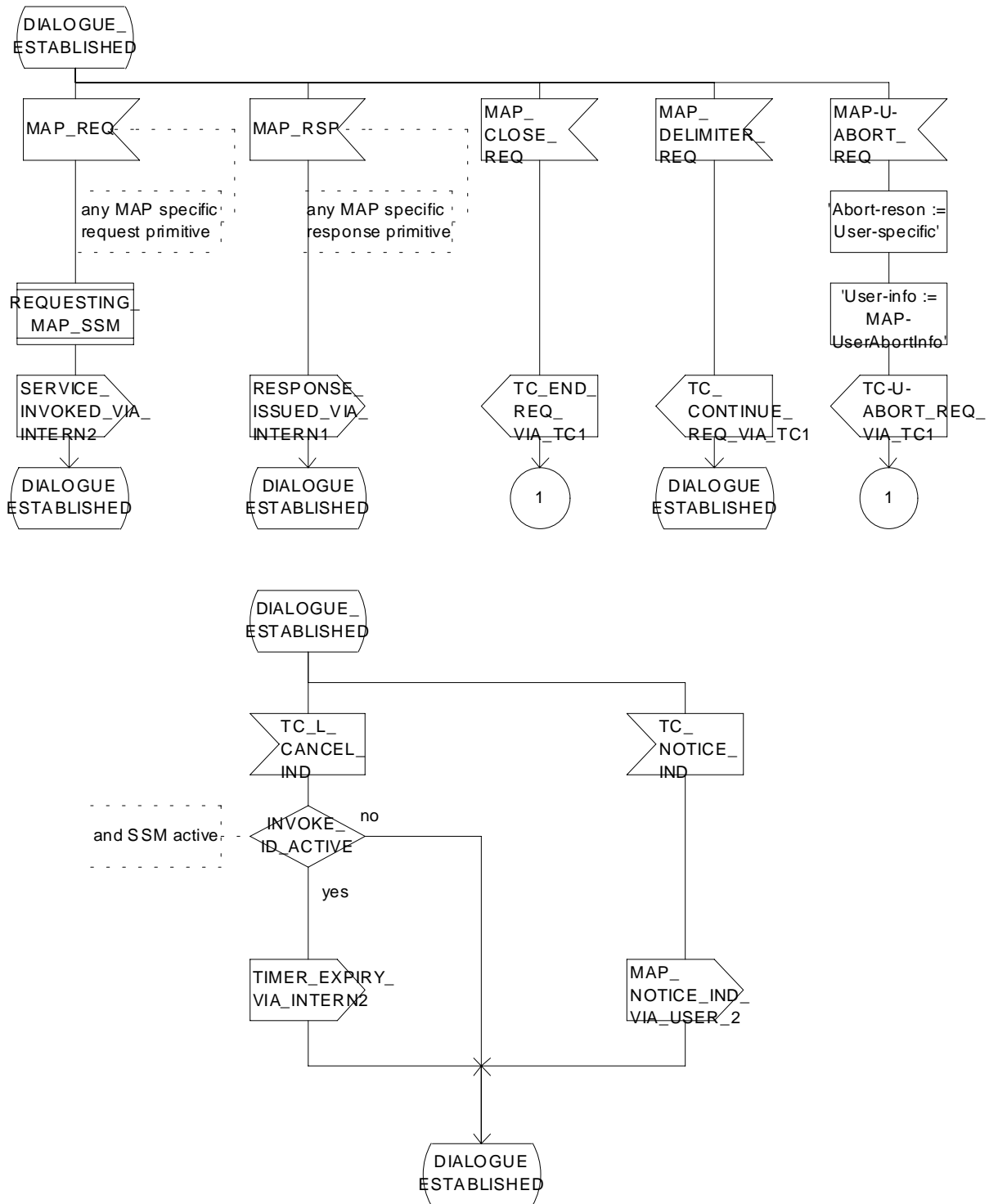
6092

Figure 16.2/3 (sheet 10 of 11): Process MAP_DSM

Process MAP_DSM

16.2_3.11(11)

Figure 16.2/3:



6093

6094

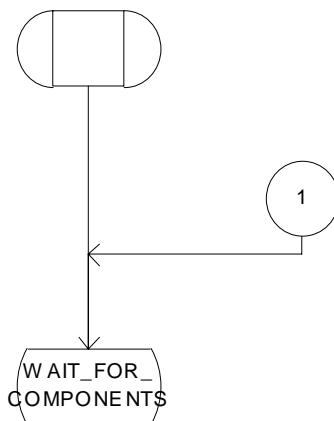
Figure 16.2/3 (sheet 11 of 11): Process MAP_DSM

Procedure PROCESS_COMPONENTS

16.2_4.1(4)

Figure 16.2/4:

Comments: Components from TCAP:
DCL
OP_CODE INTERGER,
OP_EXIST, LAST_COMPONENT, INVOKEID_ASS, LINKEDID_PRES, LINKEDID_ASS BOOLEAN;



6095

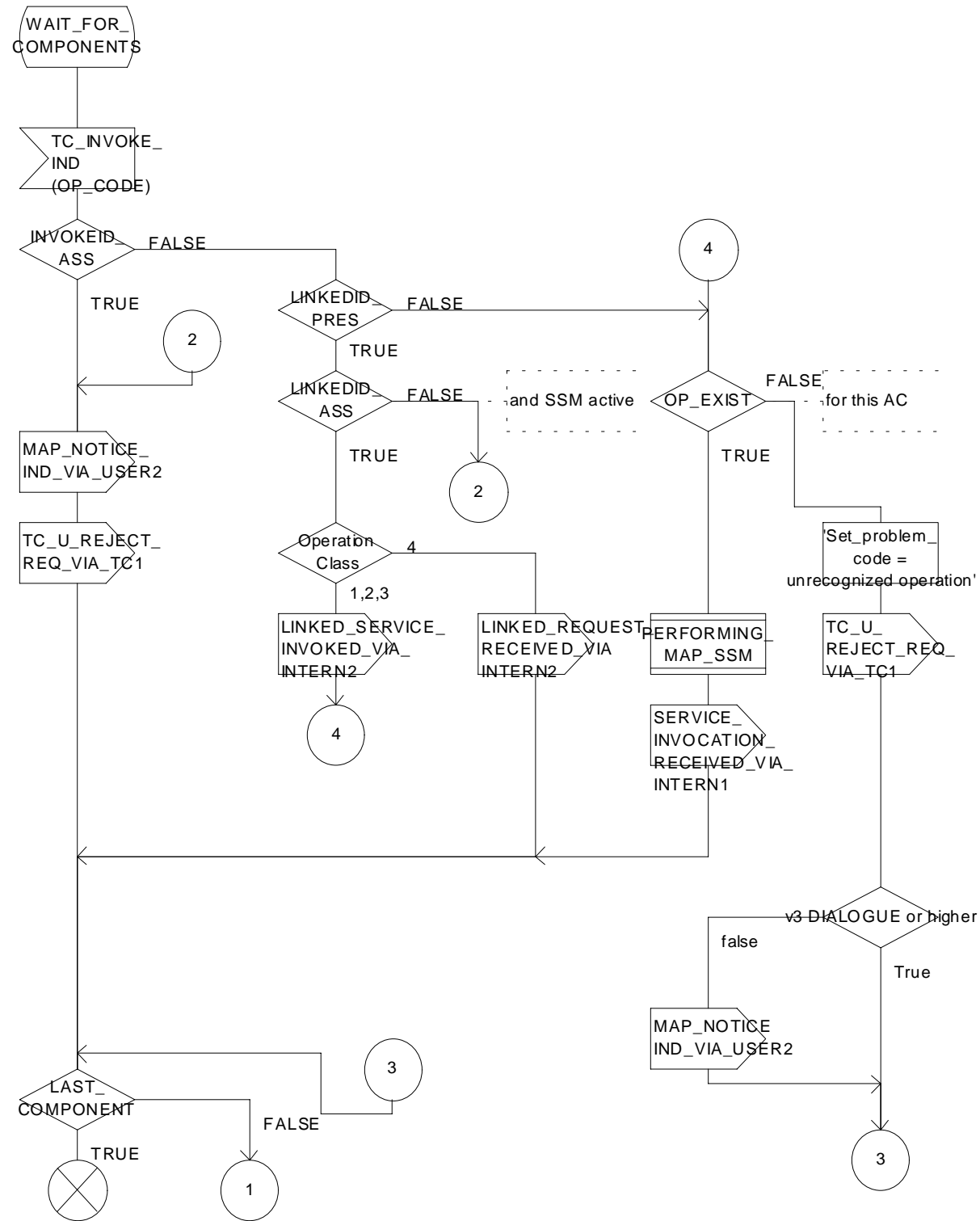
6096

Figure 16.2/4 (sheet 1 of 4): Procedure PROCESS_COMPONENTS

Procedure PROCESS_COMPONENTS

16.2_4.2(4)

Figure 16.2/4:



6097

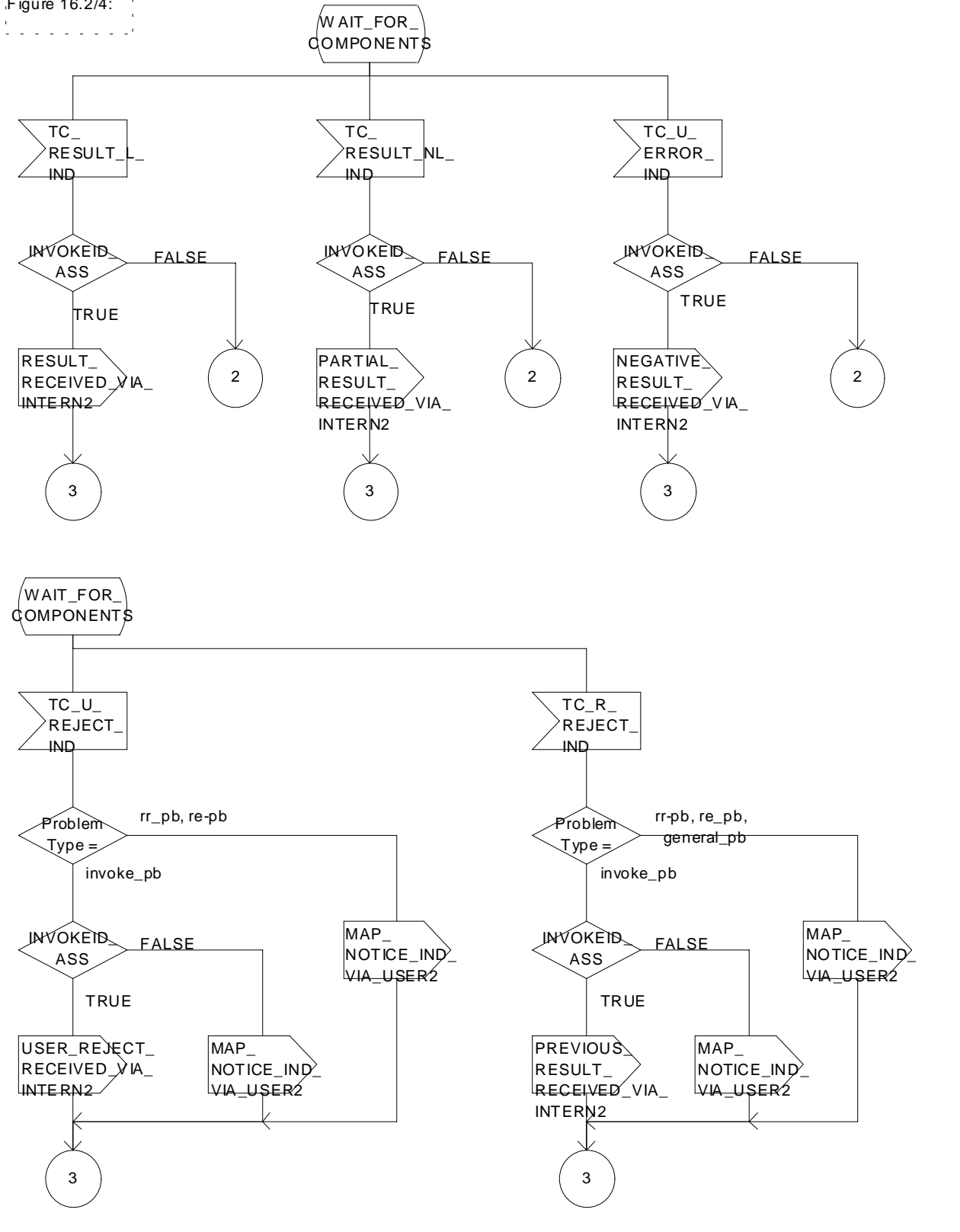
6098

Figure 16.2/4 (sheet 2 of 4): Procedure PROCESS_COMPONENTS

Procedure PROCESS_COMPONENTS

16.2_4.3(4)

Figure 16.2/4:



6099

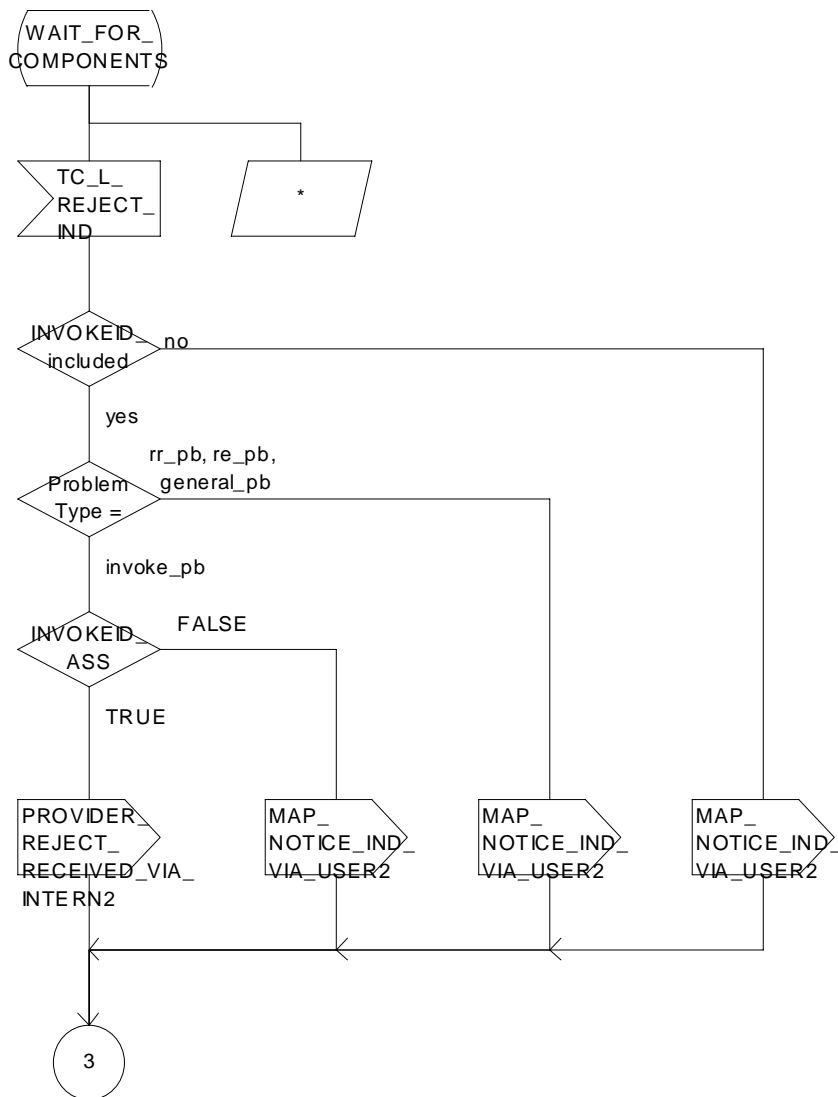
6100

Figure 16.2/4 (sheet 3 of 4): Procedure PROCESS_COMPONENTS

Procedure PROCESS_COMPONENTS

16.2_4.4(4)

Figure 16.2/4:



6101

6102

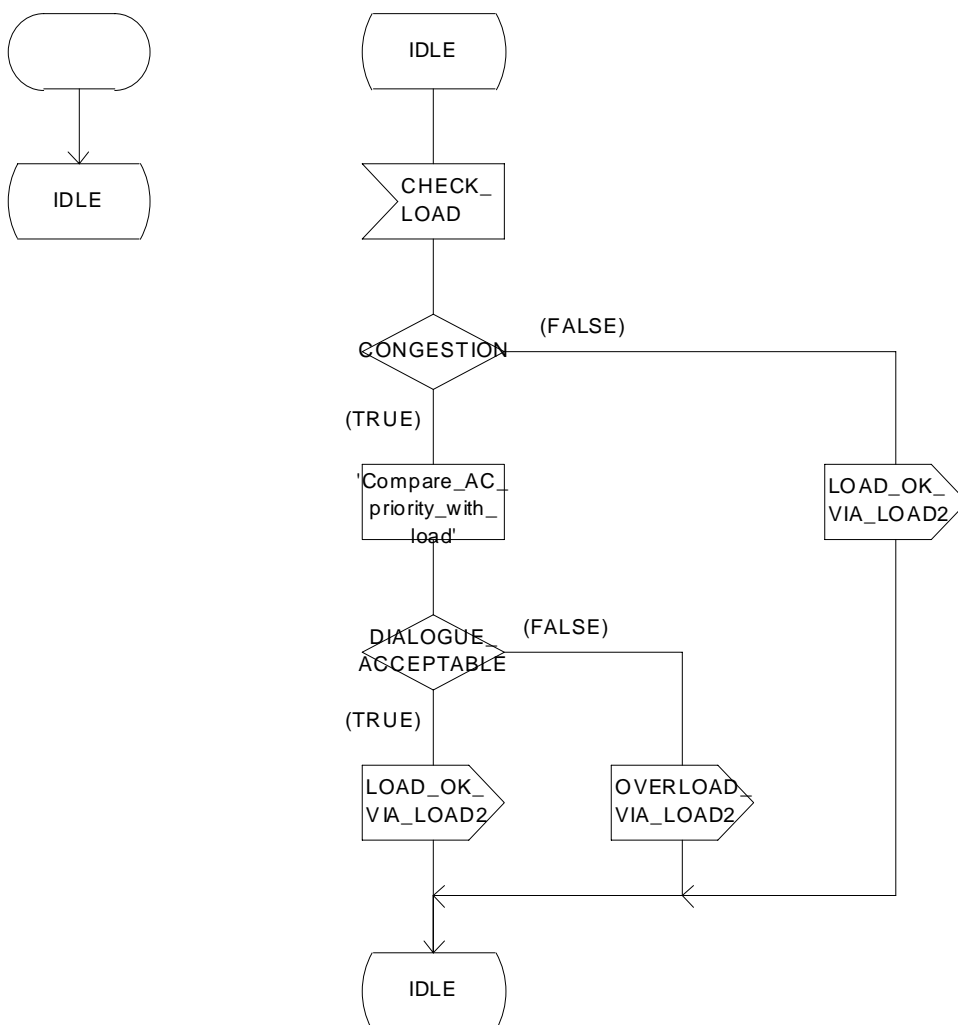
Figure 16.2/4 (sheet 4 of 4): Procedure PROCESS_COMPONENTS

Process LOAD_CTRL

16.2_5(1)

Figure 16.2/5:

Comment 'LOAD CONTROL':
 DCL
 CONGESTION, DIALOGUE_ACCEPTABLE BOOLEAN



6103

6104

Figure 16.2/5: Process LOAD_CTRL

Process PERFORMING_MAP_SSM

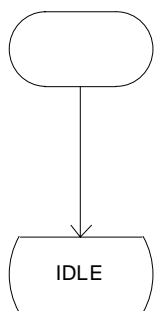
16.2_6.1(3)

Figure 16.2/6:

Comment 'MAP Service State Machine':
DCL
ARGUMENT_CORRECT, USER_ERROR_PRESENT,
SPECIFIC_ERROR_LINKED_REQUEST, CNF BOOLEAN,

OP_CLASS INTEGER,

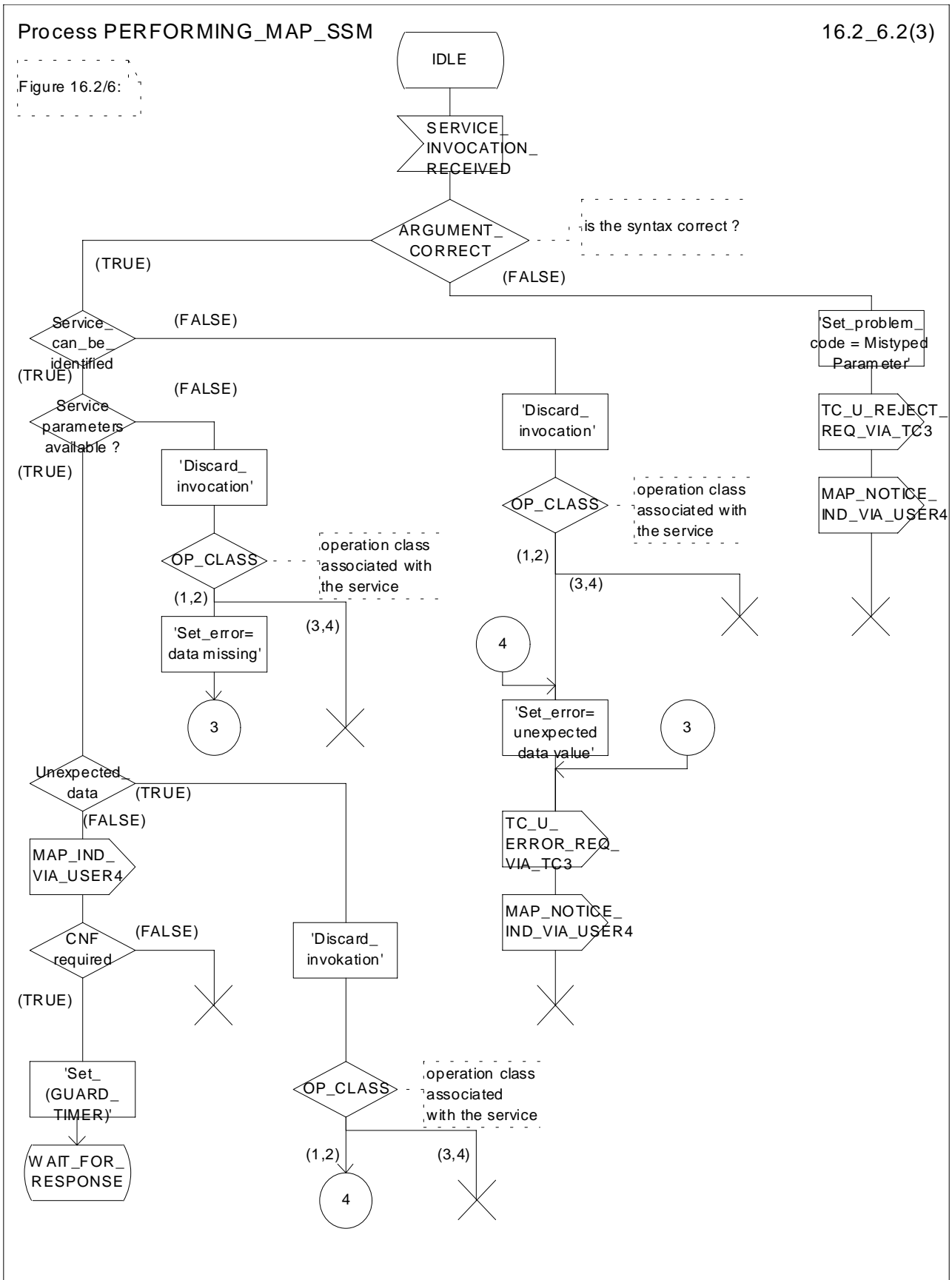
TIMER GUARD_TIMER COMMENT 'expires if MAP user does not respond';



6105

6106

Figure 16.2/6 (sheet 1 of 3): Process PERFORMING_MAP_SSM



6107

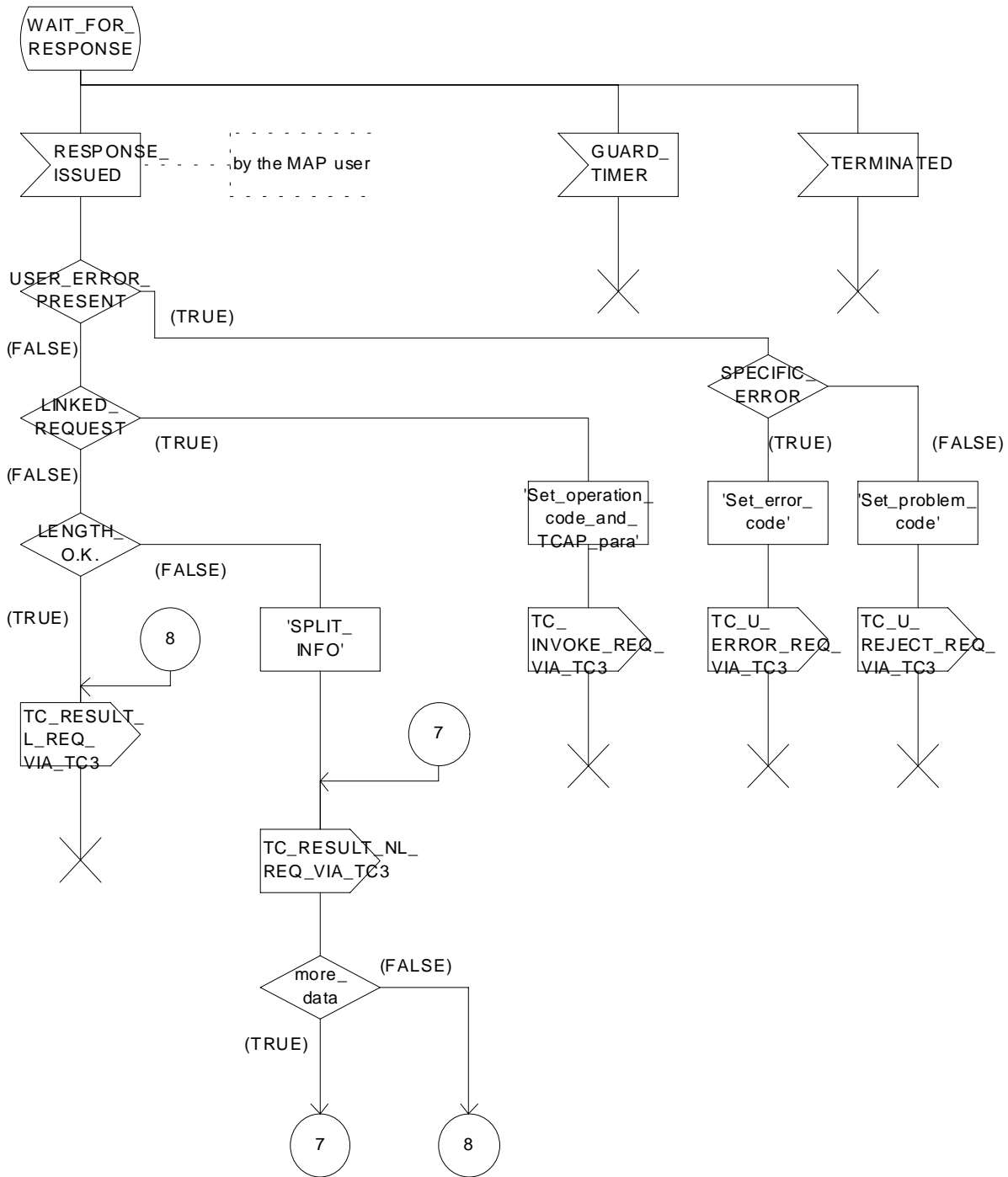
6108

Figure 16.2/6 (sheet 2 of 3): Process PERFORMING_MAP_SSM

Process PERFORMING_MAP_SSM

16.2_6.3(3)

Figure 16.2/6:



6109

6110

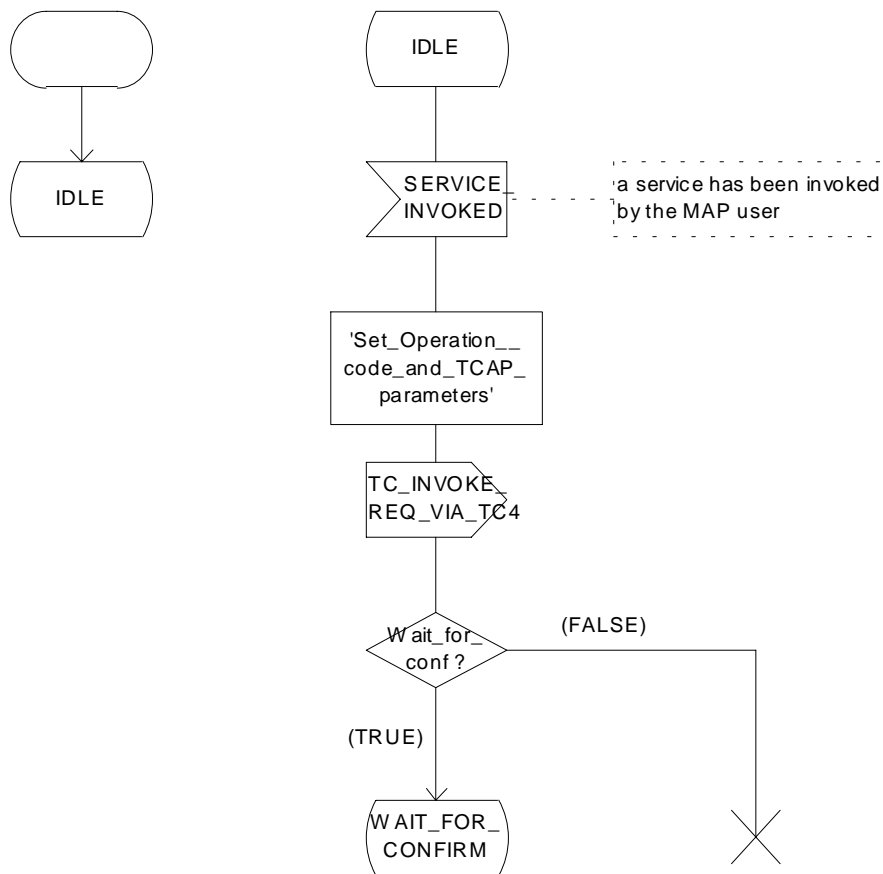
Figure 16.2/6 (sheet 3 of 3): Process PERFORMING_MAP_SSM

Process REQUESTING_MAP_SSM

16.2_7.1(4)

Figure 16.2/7:

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;



6111

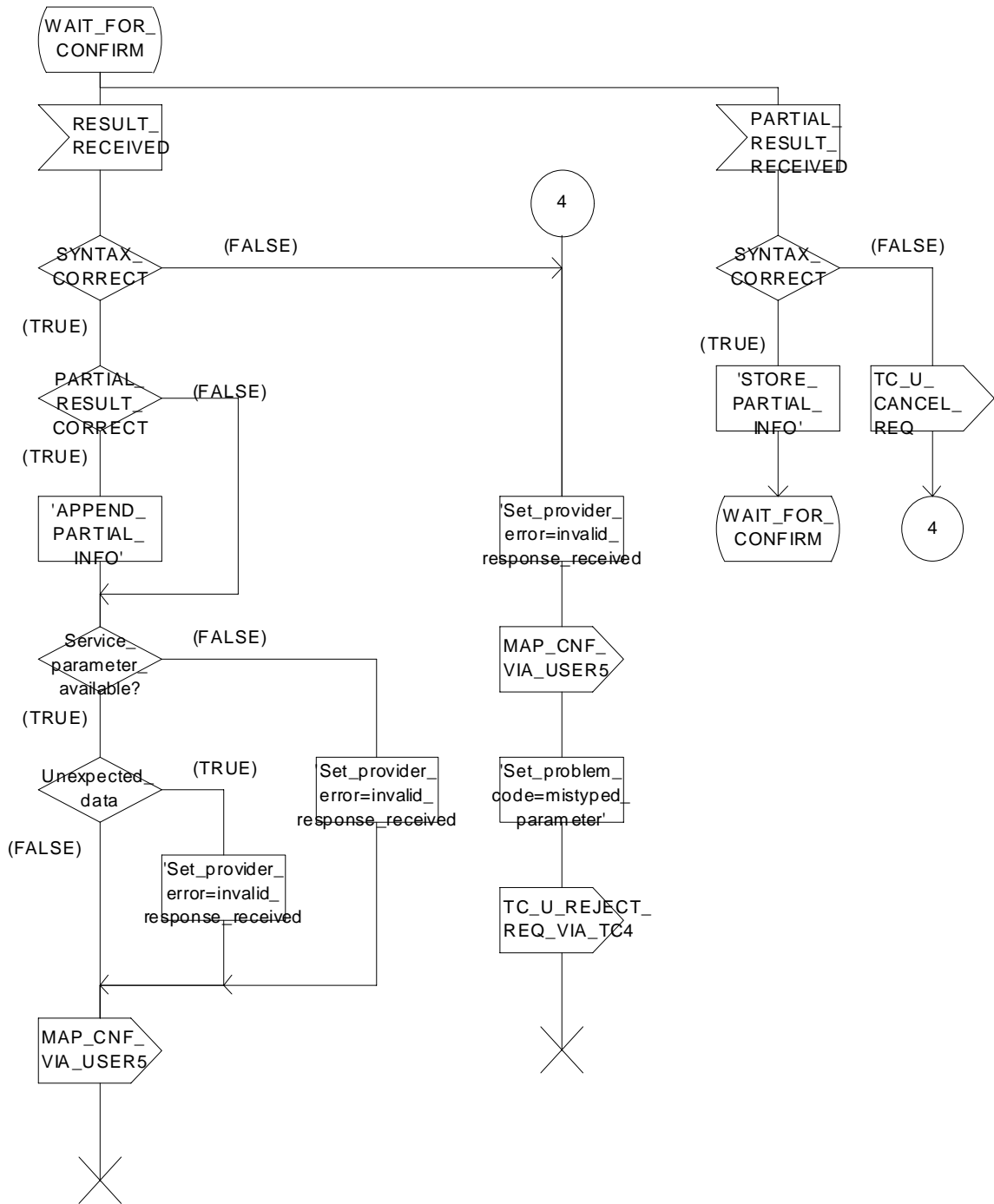
6112

Figure 16.2/7 (sheet 1 of 4): Process REQUESTING_MAP_SSM

Process REQUESTING_MAP_SSM

16.2_7.2(4)

Figure 16.2/7:



6113

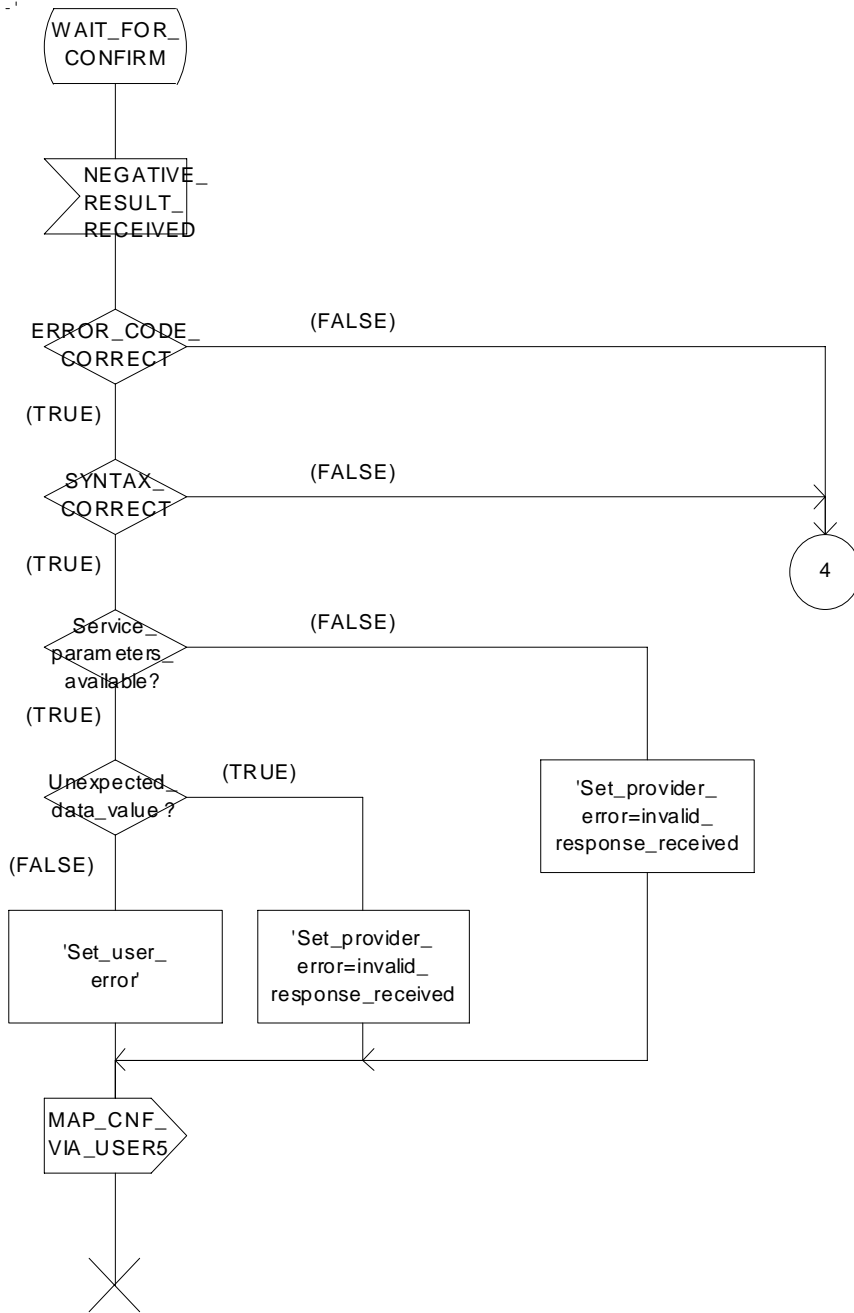
6114

Figure 16.2/7 (sheet 2 of 4): Process REQUESTING_MAP_SSM

Process REQUESTING_MAP_SSM

16.2_7.3(4)

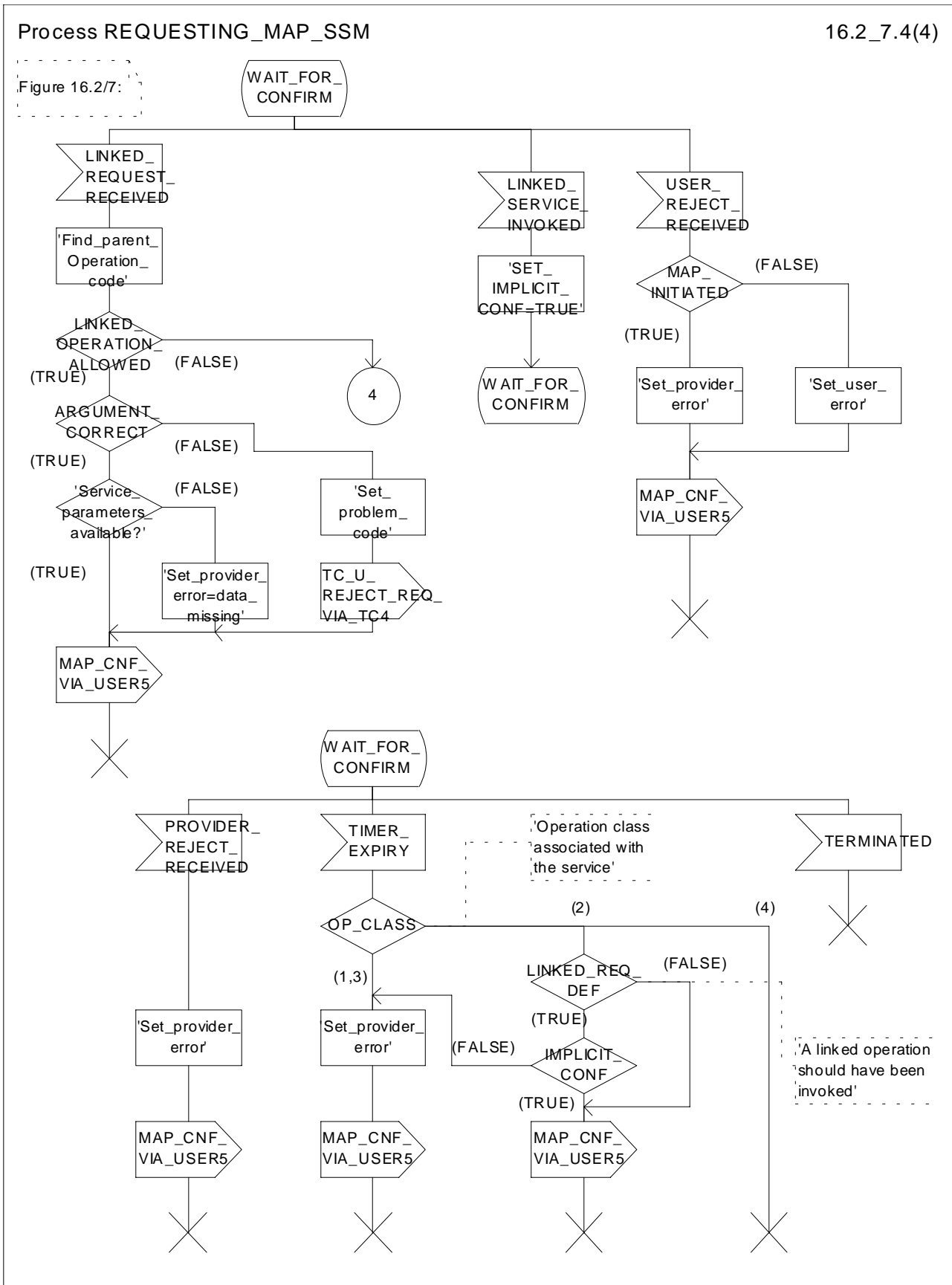
Figure 16.2/7:



6115

6116

Figure 16.2/7 (sheet 3 of 4): Process REQUESTING_MAP_SSM



6117

6118

Figure 16.2/7 (sheet 4 of 4): Process REQUESTING_MAP_SSM

6119 17 Abstract syntax of the MAP protocol

6120 17.1 General

6121 This subclause specifies the Abstract Syntaxes for the Mobile Application Part as well as the associated set of
6122 Operations and Errors, using the Abstract Syntax Notation One (ASN.1), defined in CCITT Recommendation X.208
6123 (1988) or X.680 (1994) with additions as defined in subclause 17.1.4 on Compatibility Considerations and the
6124 OPERATION and ERROR external MACROs, defined in CCITT Recommendation Q.773.

6125 The Abstract Syntax is defined for all interfaces specified in subclause 4.4 except for the A- and B-interfaces.

6126 The Mobile Application Part protocol is defined by two Abstract Syntaxes:

- 6127 - one Abstract Syntax which encompass all Operations; and
- 6128 - Errors identified by the various MAP subsystem numbers.

6129 This Abstract Syntax represents the set of values each of which is a value of the ASN.1 type TCAPMessages.
6130 MessageType as defined in CCITT Recommendation Q.773 with the ANY DEFINED BY sections resolved by the
6131 operation and error codes included in the ASN.1 module MAP-Protocol. However, only the subset of this abstract
6132 syntax which is required by the procedures defined for an entity needs to be supported:

- 6133 - one Abstract Syntax identified by the OBJECT IDENTIFIER value MAP-DialogueInformation.map-
6134 DialogueAS.

6135 This Abstract Syntax represents the set of values each of which is a value of the ASN.1 type MAP-
6136 DialogueInformation.MAP-DialoguePDU. Such a value of the ASN.1 single-ASN.1-type element is contained within
6137 the user-information element of the TCAPMessages.DialoguePortion ASN.1 type. This Abstract Syntax name is to be
6138 used as a direct reference.

6139 17.1.1 Encoding rules

6140 The encoding rules which are applicable to the defined Abstract Syntaxes are the Basic Encoding Rules for Abstract
6141 Syntax Notation One, defined in CCITT Recommendation X.690 with the same exceptions as in CCITT
6142 Recommendation Q.773 section 4 Message Representation.

6143 When the definite form is used for length encoding, a data value of length less than 128 octets must have the length
6144 encoded in the short form.

6145 When the long form is employed to code a length, the minimum number of octets shall be used to code the length field.

6146 OCTET STRING values and BIT STRING values must be encoded in a primitive form.

6147 There is no restriction to the use of empty constructors (e.g. an empty SEQUENCE type). That is, the encoding of the
6148 content of any data value shall consist of zero, one or more octets.

6149 17.1.2 Use of TC

6150 The mapping of OPERATION and ERROR to TC components is defined in ETS 300 287 (version 2) which is based on
6151 CCITT Recommendation Q.773 (1992).

6152 NOTE 1: The class of an operation is not stated explicitly but is specified as well in the ASN.1 operation type
6153 definition.

6154 Class 1: RESULT and ERROR appear in ASN.1 operation type definition.

6155 Class 2: only ERROR appears in ASN.1 operation type definition.

6156 Class 3: only RESULT appears in ASN.1 operation type definition.

6157 Class 4: both RESULT and ERROR do not appear in ASN.1 operation type definition.

6158 The ASN.1 data type which follows the keywords "ARGUMENT", "PARAMETER" or "RESULT" (for OPERATION
6159 and ERROR) is always optional from a syntactic point of view. However, except when specifically mentioned with the
6160 ASN.1 comment «-- optional», the «parameter» part of a component has to be considered as mandatory from a
6161 semantic point of view.

6162 When an optional element is missing in an invoke component or in an inner data structure while it is required by the
6163 context, an error component is returned if specified in the operation type; the associated type of error is DataMissing.
6164 This holds also when the entire parameter of an invoke component is missing while it is required by the context.

6165 NOTE 2: When a mandatory element is missing in the parameter or inner data structure of any component, a reject
6166 component is returned (if the dialogue still exists). The problem code to be used is "Mistyped parameter".

6167 The Timer Values used in the operation type definitions are indicated as ASN.1 comment. The Timer Value Ranges are:

6168 s = from 3 seconds to 10 seconds;

6169 m = from 15 seconds to 30 seconds;

6170 ml = from 1 minute to 10 minutes;

6171 l = from 28 hours to 38 hours.

6172 17.1.2.1 Use of Global Operation and Error codes defined outside MAP

6173 An entity supporting an application context greater than 2 shall be capable of receiving an operation or error code,
6174 within an application context defined in GSM 09.02, encoded as either an Object Identifier (as defined in CCITT
6175 Recommendation X.690 (1994)) or an integer value (as defined in section 17.5). Related restrictions regarding the use
6176 of Object Identifiers are as follows:

6177 - The length of the Object Identifier shall not exceed 16 octets and the number of components of the Object
6178 Identifier shall not exceed 16.

6179 - Object Identifiers shall be used only for operations or errors defined outside of GSM 09.02.

6180 - Global error codes may be sent only in response to a global operation. If a standard operation is received then
6181 a global error code shall not be sent in response.

6182 Handling of an unknown operation codes by the receiving entity is defined in section 15.1.1

6183 17.1.3 Use of information elements defined outside MAP

6184 An information element or a set of information elements (messages) transparently carried in the Mobile Application Part
6185 but defined in other recommendation/technical specifications are handled in one of the following ways:

6186 i) The contents of each information element (without the octets encoding the identifier and the length in the
6187 recommendation/technical specification where it is defined) is carried as the value of an ASN.1 NamedType
6188 derived from the OCTET STRING data type. Additionally, the internal structure may be explained by means of
6189 comments. In case of misalignment the referred to recommendation/technical specification takes precedence.

6190 ii) The complete information element (including the octets encoding the identifier and the length in the
6191 recommendation/technical specification where it is defined) or set of information elements and the identity of the
6192 associated protocol are carried as the value of the ExternalSignalInfo data type defined in the present document.
6193 Where more than one information element is carried, the information elements are sent contiguously with no
6194 filler octets between them.

6195 17.1.4 Compatibility considerations

6196 The following ASN.1 modules conform to CCITT Recommendation X.208 (1988) or X.680 (1994) (the only module
6197 which makes use of X.680 is MAP-ExtensionDataTypes), but in addition Ellipsis Notation ("..." - notation) is used as
6198 described in ITU-T Recommendation X.680 Amendment 1 (1995) wherever future protocol extensions are foreseen.

6199 The "..." construct applies only to SEQUENCE and ENUMERATED data types. An entity supporting a version greater
6200 than 1 shall not reject an unsupported extension following "..." of that SEQUENCE or ENUMERATED data type. The
6201 Encoding Rules from subclause 17.1.1 apply to every element of the whole Transfer Syntax especially to the ASN.1
6202 type EXTERNAL.

6203 Private extensions shall:

6204 1) if included in operations of an AC of V2, follow the extension marker and be tagged using PRIVATE tags up
6205 to and including 29.

6206 NOTE: This type of extension is in most cases used only within a PLMN.

6207 2) if included in operations of an AC of V3 or higher: be included only in the Private Extension Container that is
6208 defined in the specification.

6209 NOTE: This type of extension can be used between PLMNs.

6210 Private extensions shall not be included in v2 supplementary service operations.

6211 Private extensions shall not be included within user error for RegisterCCEntry and EraseCCEntry operations.

6212 PCS extensions shall be included in the PCS Extension Container that is defined in this specification.

6213 In order to improve extensibility, a few error parameters have been defined as a CHOICE between the version 2
6214 description and a SEQUENCE including the version 2 description and an extension container. Operations used in a v2-
6215 application-context must consider only the first alternative while operations used in a vn-application-context (n>2) must
6216 consider only the second alternative.

6217 17.1.5 Structure of the Abstract Syntax of MAP

6218 For each MAP parameter which has to be transferred by a MAP Protocol Data Unit (MAP message), there is a PDU
6219 field (an ASN.1 NamedType) whose ASN.1 identifier has the same name as the corresponding parameter, except for the
6220 differences required by the ASN.1 notation (blanks between words are removed or replaced by hyphen, the first letter of
6221 the first word is lower-case and the first letter of the following words are capitalized, e.g. "no reply condition time" is
6222 mapped to "noReplyConditionTime"). Additionally some words may be abbreviated as follows:

6223 bs basic service

6224 ch call handling

6225 cug closed user group

6226 ho handover

6227 ic incoming call

6228 id identity

6229 info information

6230 mm mobility management

6231 lcs location services

6232 ms mobile service

6233 oc outgoing call

6234 om operation & maintenance

6235 pw Password

6236 sm short message service

6237 ss supplementary service

6238 The MAP protocol is composed of several ASN.1 modules dealing with either operations, errors, data types, and, if
6239 applicable, split into those dealing with mobile services, call handling services, supplementary services and short
6240 message services. For operations and errors no values are assigned, but only the operation and error types in order to
6241 allow use of the defined types also by other protocols (e.g. TS GSM 04.80). The values (operation codes and error
6242 codes) are defined in a separate module. The ASN.1 source lines are preceded by line-numbers at the left margin in
6243 order to enable the usage of the cross-reference in annex A.

6244 The module containing the definition of the operation packages for MAP is:

6245 1. MAP-OperationPackages.

6246 The module containing the definition of the application contexts for MAP is:

6247 2. MAP-ApplicationContexts.

6248 The module containing the data types for the Abstract Syntax to be used for TCAPMessages.DialoguePortion for MAP
6249 is:

6250 3. MAP-DialogueInformation.

6251 The module containing the operation codes and error codes for MAP is:

6252 4. MAP-Protocol.

6253 The modules containing all operation type definitions for MAP are:

6254 5. MAP-MobileServiceOperations;

6255 6. MAP-OperationAndMaintenanceOperations;

6256 7. MAP-CallHandlingOperations;

6257 8. MAP-SupplementaryServiceOperations;

6258 9. MAP-ShortMessageServiceOperations;

6259 10. MAP-Group-Call-Operations.

6260 11. MAP-LocationServiceOperations

6261 The module containing all error type definitions for MAP is:

6262 12. MAP-Errors.

6263 Modules containing all data type definitions for MAP are:

6264 13. MAP-MS-DataTypes;

6265 14. MAP-OM-DataTypes;

6266 15. MAP-CH-DataTypes;

6267 16. MAP-SS-DataTypes;

6268 17. MAP-SS-Code;

6269 18. MAP-SM-DataTypes;

6270 19. MAP-ER-DataTypes;

6271 20. MAP-CommonDataTypes;

6272 21. MAP-TS-Code;

- 6273 22. MAP-BS-Code;
 6274 23. MAP-ExtensionDataTypes;
 6275 24. MAP-GR-DataTypes;
 6276 25. MAP-LCS-DataTypes.

6277 References are made also to modules defined outside of the present document. They are defined in the technical
 6278 specification Mobile Services Domain and technical specification Transaction Capability respectively:

- 6279 MobileDomainDefinitions;
 6280 TCAPMessages;
 6281 DialoguePDUs.

6282 17.1.6 Application Contexts

6283 The following informative table lists the latest versions of the Application Contexts used in this specification, with the
 6284 operations used by them and, where applicable, whether or not the operation description is exactly the same as for
 6285 previous versions. Information in sections 17.6 & 17.7 relates only to the ACs in this table.

AC Name	AC Version	Operations Used	Comments *
locationCancellationContext	v3	cancelLocation	
equipmentMngtContext	v2	checkIMEI	
imsiRetrievalContext	v2	sendIMSI	
infoRetrievalContext	v3	sendAuthenticationInfo	
interVlrInfoRetrievalContext	v3	sendIdentification	
handoverControlContext	v2	prepareHandover forwardAccessSignalling sendEndSignal processAccessSignalling prepareSubsequentHandover	
mwdMngtContext	v3	readyForSM	
msPurgingContext	v3	purgeMS	
shortMsgAlertContext	v2	alertServiceCentre	
resetContext	v2	reset	
networkUnstructuredSsContext	v2	processUnstructuredSS-Request unstructuredSS-Request unstructuredSS-Notify	
tracingContext	v3	activateTraceMode deactivateTraceMode	
networkFunctionalSsContext	v2	registerSS eraseSS activateSS deactivateSS registerPassword interrogateSS getPassword	
shortMsgMO-RelayContext	v3	mo-forwardSM	
shortMsgMT-RelayContext	v3	mt-forwardSM	
shortMsgGatewayContext	v3	sendRoutingInfoForSM reportSM-DeliveryStatus InformServiceCentre	the syntax of this operation has been extended in comparison with release 96 version
networkLocUpContext	v3	updateLocation forwardCheckSs-Indication restoreData insertSubscriberData	the syntax is the same in v1 & v2

		activateTraceMode	
gprsLocationUpdateContext	v3	updateGprsLocation insertSubscriberData activateTraceMode	
subscriberDataMngtContext	v3	insertSubscriberData deleteSubscriberData	
roamingNumberEnquiryContext	v3	provideRoamingNumber	
locationInfoRetrievalContext	v3	sendRoutingInfo	
gprsNotifyContext	v3	noteMsPresentForGprs	
gprsLocationInfoRetrievalContext	v3	sendRoutingInfoForGprs	
failureReportContext	v3	failureReport	
callControlTransferContext	v4	resumeCallHandling	
subscriberInfoEnquiryContext	v3	provideSubscriberInfo	
anyTimeEnquiryContext	v3	anyTimeInterrogation	
anyTimeInfoHandlingContext	v3	anyTimeSubscriptionInterrogation anyTimeModification	
ss-InvocationNotificationContext	v3	ss-InvocationNotification	
slWFSAlocationContext	v3	provideSIWFSSNumber slWFSSignallingModify	
groupCallControlContext	v3	prepareGroupCall processGroupCallSignalling forwardGroupCallSignalling sendGroupCallEndSignal	
reportingContext	v3	setReportingState statusReport remoteUserFree	
callCompletionContext	v3	registerCC-Entry eraseCC-Entry	
istAlertingContext	v3	istAlert	
ImmediateTerminationContext	v3	istCommand	
locationSvcEnquiryContext	v3	provideSubscriberLocation subscriberLocationReport	
locationSvcGatewayContext	v3	sendRoutingInfoForLCS	
mm-EventReportingContext	v3	noteMM-Event	
subscriberDataModificationNotificationContext	v3	noteSubscriberDataModified	

6286

6287 NOTE (*): The syntax of the operations is not the same as in previous versions unless explicitly stated

6288

17.2 Operation packages

6289

17.2.1 General aspects

6290 This subclause describes the operation-packages which are used to build the application-contexts defined in
6291 subclause 17.3.6292 Each operation-package is a specification of the roles of a pair of communicating objects (i.e. a pair of MAP-Providers),
6293 in term of operations which they can invoke of each other.6294 The grouping of operations into one or several packages does not necessarily imply any grouping in term of Application
6295 Service Elements.

6296 The following ASN.1 MACRO is used to describe operation-packages in this subclause:

```

6297 OPERATION-PACKAGE MACRO ::=
6298
6299 BEGIN
6300
6301 TYPE NOTATION ::= Symmetric | ConsumerInvokes SupplierInvokes |
6302 empty
6303
6304 VALUE NOTATION ::= value(VALUE OBJECT IDENTIFIER)
6305 Symmetric ::= "OPERATIONS" "{" OperationList "}"
6306 ConsumerInvokes ::= "CONSUMER INVOKES" "{" OperationList "}"
6307 SupplierInvokes ::= "SUPPLIER INVOKES" "{" OperationList "}" | empty
6308 OperationList ::= Operation | OperationList "," Operation
6309 Operation ::= value(OPERATION)
6310
6311 END
6312

```

6313 Since the application-context definitions provided in subclause 17.3 use only an informal description technique, only the
6314 type notation is used in the following subclauses to define operation-packages.

6315 The following definitions are used throughout this subclause ($n \geq 2$):

- 6316 - v1-only operation: An operation which shall be used only in v1 application-contexts;
- 6317 - vn-only operation: An operation which shall be used only in vn application-contexts;
- 6318 - v(n-1)-operation: An operation whose specification has not been modified since the MAP v(n-1) specifications or
6319 if the modifications are considered as not affecting v(n-1) implementations;
- 6320 - v(n-1)-equivalent operation: The version of an operation which excludes all the information elements and errors
6321 which have been added since the MAP v(n-1) specification;
- 6322 - vn-only package: An operation package which contains only vn-only operations;
- 6323 - v(n-1)-package: An operation package which contains only v(n-1)- operations.

6324 The names of vn-packages are suffixed by "-vn" where $n \geq 2$.

6325 For each operation package which is not vn-only ($n \geq 2$) and which does not include only v(n-1)-operations, there is a
6326 v(n-1)-equivalent package. Except when a definition is explicitly provided in the following subclauses, the v(n-1)-
6327 equivalent package includes the v(n-1)-equivalent operations of the operations which belong to this package.

6328 17.2.2 Packages specifications

6329 17.2.2.1 Location updating

6330 This operation package includes the operations required for location management procedures between HLR and VLR.

```

6331 LocationUpdatingPackage-v3 ::= OPERATION-PACKAGE
6332 -- Supplier is HLR if Consumer is VLR
6333 CONSUMER INVOKES {
6334     updateLocation}
6335 SUPPLIER INVOKES {
6336     forwardCheckSs-Indication}
6337

```

6338 The v1-equivalent and v2-equivalent packages can be determined according to the rules described in subclause 17.2.1.

6339 17.2.2.2 Location cancellation

6340 This operation package includes the operations required for location cancellation and MS purging procedures between
6341 HLR and VLR and between HLR and SGSN.

```

6342 LocationCancellationPackage-v3 ::= OPERATION-PACKAGE
6343 -- Supplier is VLR or SGSN if Consumer is HLR
6344 CONSUMER INVOKES {
6345     cancelLocation}
6346

```

6347 The v1-equivalent and v2-equivalent packages can be determined according to the rules described in subclause 17.2.1.

6348 17.2.2.3 Roaming number enquiry

6349 This operation package includes the operations required for roaming number enquiry procedures between HLR and
6350 VLR.

```
6351 RoamingNumberEnquiryPackage-v3 ::= OPERATION-PACKAGE
6352     -- Supplier is VLR if Consumer is HLR
6353     CONSUMER INVOKES {
6354         provideRoamingNumber}
6355
```

6356 The v1-equivalent and v2-equivalent packages can be determined according to the rules described in subclause 17.2.1.

6357 17.2.2.4 Information retrieval

6358 This operation package includes the operation required for the authentication information retrieval procedure between
6359 HLR and VLR and between HLR and SGSN.

```
6360 InfoRetrievalPackage-v3 ::= OPERATION-PACKAGE
6361     -- Supplier is HLR if Consumer is VLR
6362     -- Supplier is HLR if Consumer is SGSN
6363     CONSUMER INVOKES {
6364         sendAuthenticationInfo}
6365
```

6366 The v2-equivalent package is defined as follows:

```
6367 InfoRetrievalPackage-v2 ::= OPERATION-PACKAGE
6368     -- Supplier is HLR if Consumer is VLR
6369     -- Supplier is HLR if Consumer is SGSN
6370     CONSUMER INVOKES {
6371         sendAuthenticationInfo}
6372
```

6373 The v1-equivalent package is defined as follows:

```
6374 InfoRetrievalPackage-v1 ::= OPERATION-PACKAGE
6375     -- Supplier is HLR or VLR if Consumer is VLR
6376     -- Supplier is HLR if Consumer is SGSN
6377     CONSUMER INVOKES {
6378         sendParameters}
6379
```

6380 17.2.2.5 Inter-VLR information retrieval

6381 This operation package includes the operations required for inter VLR information retrieval procedures.

```
6382 InterVlrInfoRetrievalPackage-v3 ::= OPERATION-PACKAGE
6383     -- Supplier is VLR if Consumer is VLR
6384     CONSUMER INVOKES {
6385         sendIdentification}
6386
```

6387 The v2-equivalent package is defined as follows:

```
6388 InterVlrInfoRetrievalPackage-v2 ::= OPERATION-PACKAGE
6389     -- Supplier is VLR if Consumer is VLR
6390     CONSUMER INVOKES {
6391         sendIdentification}
6392
```

6393 The v1-equivalent package is : InfoRetrievalPackage-v1

6394 17.2.2.6 IMSI retrieval

6395 This operation package includes the operation required for the IMSI retrieval procedure between HLR and VLR.

```
6396 IMSIRetrievalPackage-v2 ::= OPERATION-PACKAGE
6397     -- Supplier is HLR if Consumer is VLR
6398     CONSUMER INVOKES {
6399         sendIMSI}
6400
```

6401 This package is v2 only.

6402 17.2.2.7 Call control transfer

6403 This operation package includes the operation required for the call control transfer procedure between VMSC and
6404 GMSC.

```
6405 CallControlTransferPackage-v4 ::= OPERATION-PACKAGE
6406     -- Supplier is GMSC if Consumer is VMSC
6407     CONSUMER INVOKES {
6408         resumeCallHandling}
6409
```

6410 The v3-equivalent package can be determined according to the rules described in subclause 17.2.1.

6411 17.2.2.8 - 17.2.2.9 Void

6412 17.2.2.10 Interrogation

6413 This operation package includes the operations required for interrogation procedures between MSC and HLR or NPLR.

```
6414 InterrogationPackage-v3 ::= OPERATION-PACKAGE
6415     -- Supplier is HLR or NPLR if Consumer is MSC
6416     CONSUMER INVOKES {
6417         sendRoutingInfo}
6418
```

6419 The v1-equivalent and v2-equivalent packages can be determined according to the rules described in subclause 17.2.1.

6420 17.2.2.11 Void

6421 17.2.2.12 Handover Control

6422 This operation package includes the operations required for handover procedures between MSCs.

```
6423 HandoverControlPackage-v2 ::= OPERATION-PACKAGE
6424     -- Supplier is MSCB if Consumer is MSCA
6425     CONSUMER INVOKES {
6426         prepareHandover,
6427         forwardAccessSignalling}
6428     SUPPLIER INVOKES {
6429         sendEndSignal,
6430         processAccessSignalling,
6431         prepareSubsequentHandover}
6432
```

6433 The v1-equivalent package is defined as follows.

```
6434 HandoverControlPackage-v1 ::= OPERATION-PACKAGE
6435     -- Supplier is MSCB if Consumer is MSCA
6436     CONSUMER INVOKES {
6437         performHandover,
6438         forwardAccessSignalling,
6439         traceSubscriberActivity}
6440     SUPPLIER INVOKES {
6441         sendEndSignal,
6442         noteInternalHandover,
6443         processAccessSignalling,
6444         performSubsequentHandover}
6445
```

6446 17.2.2.13 Subscriber Data management stand alone

6447 This operation package includes the operations required for stand alone subscriber data management procedures
6448 between HLR and VLR or between HLR and SGSN.

```
6449 SubscriberDataMngtStandAlonePackage-v3 ::= OPERATION-PACKAGE
6450     -- Supplier is VLR or SGSN if Consumer is HLR
6451     CONSUMER INVOKES {
6452         insertSubscriberData,
6453         deleteSubscriberData}
6454
```

6455 The v1-equivalent and v2-equivalent packages can be determined according to the rules described in subclause 17.2.1.

6456 17.2.2.14 Equipment management

6457 This operation package includes the operations required for equipment management procedures between EIR and MSC
6458 or between EIR and SGSN.

```
6459 EquipmentMngtPackage-v2 ::= OPERATION-PACKAGE
6460     -- Supplier is EIR if Consumer is MSC
6461     -- Supplier is EIR if Consumer is SGSN
6462     CONSUMER INVOKES {
6463         checkIMEI}
6464
```

6465 The v1-equivalent package can be determined according to the rules described in subclause 17.2.1.

6466 17.2.2.15 Subscriber data management

6467 This operation package includes the operations required for subscriber data management procedures between HLR and
6468 VLR or between HLR and SGSN.

```
6469 SubscriberDataMngtPackage-v3 ::= OPERATION-PACKAGE
6470     -- Supplier is VLR or SGSN if Consumer is HLR
6471     CONSUMER INVOKES {
6472         insertSubscriberData}
6473
```

6474 The v1-equivalent and v2-equivalent packages can be determined according to the rules described in subclause 17.2.1.

6475 17.2.2.16 Location register restart

6476 This operation package includes the operations required for location register restart procedures between HLR and VLR
6477 or between HLR and SGSN.

```
6478 ResetPackage-v2 ::= OPERATION-PACKAGE
6479     -- Supplier is VLR or SGSN if Consumer is HLR
6480     CONSUMER INVOKES {
6481         reset}
6482
```

6483 The v1-equivalent package can be determined according to the rules described in subclause 17.2.1.

6484 17.2.2.17 Tracing stand-alone

6485 This operation package includes the operations required for stand alone tracing procedures between HLR and VLR or
6486 between HLR and SGSN.

```
6487 TracingStandAlonePackage-v3 ::= OPERATION-PACKAGE
6488     -- Supplier is VLR or SGSN if Consumer is HLR
6489     CONSUMER INVOKES {
6490         activateTraceMode,
6491         deactivateTraceMode}
6492
```

6493 The v1-equivalent and v2-equivalent packages can be determined according to the rules described in subclause 17.2.1.

6494 17.2.2.18 Functional SS handling

6495 This operation package includes the operations required for functional supplementary services procedures between VLR
6496 and HLR.

```
6497 FunctionalSsPackage-v2 ::= OPERATION-PACKAGE
6498     -- Supplier is HLR if Consumer is VLR
6499     CONSUMER INVOKES {
6500         registerSS,
6501         eraseSS,
6502         activateSS,
6503         deactivateSS,
6504         registerPassword,
6505         interrogateSS}
6506     SUPPLIER INVOKES {
6507         getPassword}
6508
```

6509 The v1-equivalent package can be determined according to the rules described in subclause 17.2.1.

6510 17.2.2.19 Tracing

6511 This operation package includes the operations required for tracing procedures between HLR and VLR or between HLR
6512 and SGSN.

```
6513 TracingPackage-v3 ::= OPERATION-PACKAGE
6514     -- Supplier is VLR or SGSN if Consumer is HLR
6515     CONSUMER INVOKES {
6516         activateTraceMode}
6517
```

6518 The v1-equivalent and v2-equivalent packages can be determined according to the rules described in subclause 17.2.1.

6519 17.2.2.20 Binding

6520 This operation package includes the operation required to initialize a supplementary service procedure between VLR
6521 and HLR or between gsmSCF and HLR.

```
6522 BindingPackage-v1 ::= OPERATION-PACKAGE
6523     -- Supplier is HLR if Consumer is VLR
6524     -- Supplier is gsmSCF if Consumer is HLR
6525     CONSUMER INVOKES {
6526         beginSubscriberActivity}
6527
```

6527 This package is v1 only.

6528 17.2.2.21 Unstructured SS handling

6529 This operation package includes the operations required for unstructured supplementary services procedures between
6530 VLR and HLR, between the HLR and the gsmSCF, and between HLR and HLR.

```
6531 UnstructuredSsPackage-v2 ::= OPERATION-PACKAGE
6532     -- Supplier is HLR if Consumer is VLR
6533     -- Supplier is gsmSCF or HLR if Consumer is HLR
6534     CONSUMER INVOKES {
6535         processUnstructuredSS-Request}
6536     SUPPLIER INVOKES {
6537         unstructuredSS-Request,
6538         unstructuredSS-Notify}
6539
```

6540 The v1-equivalent package is defined as follows:

```
6541 UnstructuredSsPackage-v1 ::= OPERATION-PACKAGE
6542     -- Supplier is HLR if Consumer is VLR
6543     -- Supplier is gsmSCF if Consumer is HLR
6544     CONSUMER INVOKES {
6545         processUnstructuredSS-Data}
6546
```

6547 17.2.2.22 MO Short message relay services

6548 This operation package includes the operations required for short message relay service procedures between IWMSC
6549 and VMSC or between GMSC and MSC or between SGSN and IWMSC.

```
6550 MOShortMsgRelayPackage-v3 ::= OPERATION-PACKAGE
6551     -- Supplier is IWMSC if Consumer is MSC
6552     -- Supplier is IWMSC if Consumer is SGSN
6553     CONSUMER INVOKES {
6554         MO-forwardSM}
6555
```

6556 The v2-equivalent package is defined as follows:

```
6557 ShortMsgRelayPackage-v2 ::= OPERATION-PACKAGE
6558     -- Supplier is IWMSC if Consumer is MSC
6559     -- Supplier is MSC or SGSN if Consumer is GMSC
6560     -- Supplier is IWMSC if Consumer is SGSN
6561     CONSUMER INVOKES {
6562         forwardSM}
6563
```

6564 The v1-equivalent package can be determined according to the rules described in subclause 17.2.1.

6565 17.2.2.23 Short message gateway services

6566 This operation package includes the operations required for short message service gateway procedures between MSC
6567 and HLR.

```
6568 ShortMsgGatewayPackage-v3 ::= OPERATION-PACKAGE
6569 -- Supplier is HLR if Consumer is GMSC
6570 CONSUMER INVOKES {
6571     sendRoutingInfoForSM,
6572     reportSM-DeliveryStatus}
6573 SUPPLIER INVOKES {
6574     informServiceCentre}
```

6575
6576 The v2-equivalent package can be determined according to the rules described in subclause 17.2.1

6577 The v1-equivalent package is defined as follows:

```
6578 ShortMsgGatewayPackage-v1 ::= OPERATION-PACKAGE
6579 -- Supplier is HLR if Consumer is GMSC
6580 CONSUMER INVOKES {
6581     sendRoutingInfoForSM
6582     reportSMDeliveryStatus}
```

6584 17.2.2.24 MT Short message relay services

6585 This operation package includes the operations required for short message relay service procedures between GMSC and
6586 MSC or between GMSC and SGSN.

```
6587 MTShortMsgRelayPackage-v3 ::= OPERATION-PACKAGE
6588 -- Supplier is MSC or SGSN if Consumer is GMSC
6589 CONSUMER INVOKES {
6590     MT-forwardSM}
```

6591
6592 The v2-equivalent package is: **ShortMsgRelayPackage-v2**

6593 17.2.2.25 Void

6594 17.2.2.26 Message waiting data management

6595 This operation package includes the operations required for short message waiting data procedures between HLR and
6596 VLR, between HLR and SGSN.

```
6597 MwdMngtPackage-v3 ::= OPERATION-PACKAGE
6598 -- Supplier is HLR if Consumer is SGSN
6599 -- Supplier is HLR if Consumer is VLR
6600 CONSUMER INVOKES {
6601     readyForSM}
```

6602
6603 The v2-equivalent package can be determined according to the rules described in subclause 17.2.1.

6604

6605 The v1-equivalent package is defined as follows:

```
6606 MwdMngtPackage-v1 ::= OPERATION-PACKAGE
6607 -- Supplier is HLR if Consumer is VLR
6608 CONSUMER INVOKES {
6609     noteSubscriberPresent}
```

6611 17.2.2.27 Alerting

6612 This operation package includes the operations required for alerting between HLR and IW MSC.

```
6613 AlertingPackage-v2 ::= OPERATION-PACKAGE
6614 -- Supplier is IW MSC if Consumer is HLR
6615 CONSUMER INVOKES {
6616     alertServiceCentre}
```

6617

6618 The v1-equivalent package is defined as follows.

```
6619 AlertingPackage-v1 ::= OPERATION-PACKAGE
6620     -- Supplier is IWMSC if Consumer is HLR
6621     CONSUMER INVOKES {
6622         alertServiceCentreWithoutResult}
6623
```

6624 17.2.2.28 Data restoration

6625 This operation package includes the operations required for VLR data restoration between HLR and VLR.

```
6626 DataRestorationPackage-v3 ::= OPERATION-PACKAGE
6627     -- Supplier is HLR if Consumer is VLR
6628     CONSUMER INVOKES {
6629         restoreData}
6630
```

6631 The v2-equivalent package can be determined according to the rules described in subclause 17.2.1.

6632 The v1-equivalent package is: InfoRetrievalPackage-v1

6633 17.2.2.29 Purging

6634 This operation package includes the operations required for purging between HLR and VLR or between HLR and
6635 SGSN.

```
6636 PurgingPackage-v3 ::= OPERATION-PACKAGE
6637     -- Supplier is HLR if Consumer is VLR
6638     -- Supplier is HLR if Consumer is SGSN
6639     CONSUMER INVOKES {
6640         purgeMS}
6641
```

6642 The v2-equivalent package can be determined according to the rules described in subclause 17.2.1.

6643 17.2.2.30 Subscriber information enquiry

6644 This operation package includes the operations required for subscriber information enquiry procedures between HLR
6645 and VLR.

```
6646 SubscriberInformationEnquiryPackage-v3 ::= OPERATION-PACKAGE
6647     -- Supplier is VLR if Consumer is HLR
6648     CONSUMER INVOKES {
6649         provideSubscriberInfo}
6650
```

6651 This package is v3 only.

6652 17.2.2.31 Any time information enquiry

6653 This operation package includes the operations required for any time information enquiry procedures between gsmSCF
6654 and HLR or GMLC.

```
6655 AnyTimeInformationEnquiryPackage-v3 ::= OPERATION-PACKAGE
6656     -- Supplier is HLR or GMLC if Consumer is gsmSCF
6657     CONSUMER INVOKES {
6658         anyTimeInterrogation}
6659
```

6660 This package is v3 only.

6661 17.2.2.32 Group Call Control

6662 This operation package includes the operations required for group call and broadcast call procedures between MSCs.

```

6663 GroupCallControlPackage-v3 ::= OPERATION-PACKAGE
6664   -- Supplier is relay MSC if Consumer is anchor MSC
6665   CONSUMER INVOKES {
6666     prepareGroupCall,
6667     forwardGroupCallSignalling}
6668   SUPPLIER INVOKES {
6669     sendGroupCallEndSignal,
6670     processGroupCallSignalling}
6671

```

6672 This package is v3 only.

6673 17.2.2.33 Provide SIWFS number

6674 This operation package includes the operations required between VMSC and SIWF for requesting resources from an SIWF.

```

6676 ProvideSIWFSNumberPackage-v3 ::= OPERATION-PACKAGE
6677   -- Supplier is SIWF if Consumer is VMSC
6678   CONSUMER INVOKES {
6679     provideSIWFSNumber}
6680

```

6681 This package is v3 only.

6682 17.2.2.34 SIWFS Signalling Modify

6683 This operation package includes the operations required for the modification of the resources in an SIWF between the VMSC and SIWF.

```

6685 SIWFSSignallingModifyPackage-v3 ::= OPERATION-PACKAGE
6686   -- Supplier is SIWF if Consumer is VMSC
6687   CONSUMER INVOKES {
6688     siWFSSignallingModify}
6689

```

6690 This package is v3 only.

6691 17.2.2.35 Gprs location updating

6692 This operation package includes the operations required for the gprs location management procedures between HLR and SGSN.

```

6694 GprsLocationUpdatingPackage-v3 ::= OPERATION-PACKAGE
6695   -- Supplier is HLR if Consumer is SGSN
6696   CONSUMER INVOKES {
6697     updateGprsLocation}
6698

```

6699 This package is v3 only.

6700 17.2.2.36 Gprs Interrogation

6701 This operation package includes the operations required for interrogation procedures between HLR and GGSN.

```

6702 GprsInterrogationPackage-v3 ::= OPERATION-PACKAGE
6703   -- Supplier is HLR if Consumer is GGSN
6704   CONSUMER INVOKES {
6705     sendRoutingInfoForGprs}
6706

```

6707 This package is v3 only.

6708 17.2.2.37 Failure reporting

6709 This operation package includes the operations required for failure reporting between HLR and GGSN.

```
6710 FailureReportingPackage-v3 ::= OPERATION-PACKAGE
6711     -- Supplier is HLR if Consumer is GGSN
6712     CONSUMER INVOKES {
6713         failureReport}
```

6714

6715 This package is v3 only.

6716 17.2.2.38 GPRS notifying

6717 This operation package includes the operations required for notifying that GPRS subscriber is present between HLR and
6718 GGSN.

```
6719 GprsNotifyingPackage-v3 ::= OPERATION-PACKAGE
6720     -- Supplier is GGSN if Consumer is HLR
6721     CONSUMER INVOKES {
6722         noteMsPresentForGprs}
```

6723

6724 This package is v3 only.

6725 17.2.2.39 Supplementary Service invocation notification

6726 This operation package includes the operations required for Supplementary Service invocation notification procedures
6727 between the MSC and the gsmSCF and between the HLR and the gsmSCF.

```
6728 SS-InvocationNotificationPackage-v3 ::= OPERATION-PACKAGE
6729     -- Supplier is gsmSCF if Consumer is MSC
6730     -- Supplier is gsmSCF if Consumer is HLR
6731     CONSUMER INVOKES {
6732         ss-InvocationNotification}
```

6733

6734 This package is v3 only.

6735 17.2.2.40 Set Reporting State

6736 This operation package includes the operation required for procedures between HLR and VLR to set the reporting state.

```
6737 SetReportingStatePackage-v3 ::= OPERATION-PACKAGE
6738     -- Supplier is VLR if Consumer is HLR
6739     CONSUMER INVOKES {
6740         setReportingState}
```

6741

6742 This package is v3 only.

6743 17.2.2.41 Status Report

6744 This operation package includes the operation required for procedures between VLR and HLR to report call results and
6745 events.

```
6746 StatusReportPackage-v3 ::= OPERATION-PACKAGE
6747     -- Supplier is HLR if Consumer is VLR
6748     CONSUMER INVOKES {
6749         statusReport}
```

6750

6751 This package is v3 only.

6752 17.2.2.42 Remote User Free

6753 This operation package includes the operation required by the HLR to indicate to the VLR that the remote user is free.

```
6754 RemoteUserFreePackage-v3 ::= OPERATION-PACKAGE
6755 -- Supplier is VLR if Consumer is HLR
6756 CONSUMER INVOKES {
6757     remoteUserFree}
6758
```

6759 This package is v3 only.

6760 17.2.2.43 Call Completion

6761 This operation package includes the operations required for procedures between VLR and HLR for subscriber control of
6762 call completion services.

```
6763 CallCompletionPackage-v3 ::= OPERATION-PACKAGE
6764 -- Supplier is HLR if Consumer is VLR
6765 CONSUMER INVOKES {
6766     registerCC-Entry,
6767     eraseCC-Entry}
6768
```

6769 This package is v3 only.

6770 17.2.2.44 Location service gateway services

6771 This operation package includes the operations required for location service gateway procedures between GMLC and
6772 HLR.

```
6773 LocationSvcGatewayPackage-v3 ::= OPERATION-PACKAGE
6774 -- Supplier is HLR if Consumer is GMLC
6775 CONSUMER INVOKES {
6776     sendRoutingInfoForLCS}
6777
```

6778 This package is v3 only.

6779 17.2.2.45 Location service enquiry

6780 This operation package includes the operations required for the location service enquiry procedures between GMLC and
6781 MSC.

```
6782 LocationSvcEnquiryPackage-v3 ::= OPERATION-PACKAGE
6783 -- Supplier is MSC if Consumer is GMLC
6784 CONSUMER INVOKES {
6785     provideSubscriberLocation}
6786 SUPPLIER INVOKES {
6787     subscriberLocationReport}
6788
```

6789 This package is v3 only.

6790 17.2.2.46 Void

6791 17.2.2.47 Void

6792 17.2.2.48 Void

6793 17.2.2.49 IST Alerting

6794 This operation package includes the operation required for alerting procedures between the MSC (Visited MSC or
6795 Gateway MSC) and HLR.

```
6796 IST-AlertingPackage-v3 ::= OPERATION-PACKAGE
6797     -- Supplier is HLR if Consumer is VMSC
6798     -- Supplier is HLR if Consumer is GMSC
6799     CONSUMER INVOKES {
6800         istAlert}
```

6801
6802 This package is v3 only.

6803 17.2.2.50 Service Termination

6804 This operation package includes the operation required for immediate service termination procedures between the HLR
6805 and the Visited MSC or between the HLR and the Gateway MSC.

```
6806 ServiceTerminationPackage-v3 ::= OPERATION-PACKAGE
6807     -- Supplier is VMSC or GMSC if Consumer is HLR
6808     CONSUMER INVOKES {
6809         istCommand}
```

6810
6811 This package is v3 only.

6812 17.2.2.51 Mobility Management event notification

6813 This operation package includes the operations required for Mobility Management event notification procedures
6814 between VLR and gsmSCF.

```
6815 MM-EventReportingPackage-v3 ::= OPERATION-PACKAGE
6816     -- Supplier is gsmSCF if Consumer is VLR
6817     CONSUMER INVOKES {
6818         noteMM-Event}
```

6819 This package is v3 only.

6820 17.2.2.52 Any time information handling

6821 This operation package includes the operations required for any time information handling procedures between gsmSCF
6822 and HLR.

```
6823 AnyTimeInformationHandlingPackage-v3 ::= OPERATION-PACKAGE
6824     -- Supplier is HLR if Consumer is gsmSCF
6825     CONSUMER INVOKES {
6826         anyTimeSubscriptionInterrogation,
6827         anyTimeModification}
```

6828
6829 This package is v3 only.

17.2.2.53 Subscriber Data modification notification

This operation package includes the operations required for Subscriber Data modification notification procedures
between HLR and gsmSCF.

```
SubscriberDataModificationNotificationPackage-v3 ::= OPERATION-PACKAGE
    -- Supplier is gsmSCF if Consumer is HLR
    CONSUMER INVOKES {
        noteSubscriberDataModified}
```

This package is v3 only.

6830 17.3 Application contexts

6831 17.3.1 General aspects

6832 An application-context is assigned for each dialogue established by a MAP-user. In the present document each
 6833 application-context is assigned a name which is supplied in the MAP-OPEN Req primitive by the MAP-User and
 6834 transmitted to the peer under certain circumstances.

6835 The following ASN.1 MACRO is used to describe the main aspects of application-contexts in the following subclauses:

```

6836 APPLICATION-CONTEXT MACRO ::=
6837 BEGIN
6838 BEGIN
6839 TYPE NOTATION ::= Symmetric | InitiatorConsumerOf
6840 ResponderConsumerOf | empty
6841 ResponderConsumerOf | empty
6842 VALUE NOTATION ::= value(VALUE OBJECT IDENTIFIER)
6843 VALUE NOTATION ::= value(VALUE OBJECT IDENTIFIER)
6844 Symmetric ::= "OPERATIONS OF" "{" PackageList "}"
6845 Symmetric ::= "OPERATIONS OF" "{" PackageList "}"
6846 InitiatorConsumerOf ::= "INITIATOR CONSUMER OF" "{" PackageList "}"
6847 InitiatorConsumerOf ::= "INITIATOR CONSUMER OF" "{" PackageList "}"
6848 ResponderConsumerOf ::= "RESPONDER CONSUMER OF" "{" PackageList "}"
6849 ResponderConsumerOf ::= "RESPONDER CONSUMER OF" "{" PackageList "}"
6850 | empty
6851 | empty
6852 PackageList ::= Package | PackageList "," Package
6853 PackageList ::= Package | PackageList "," Package
6854 Package ::= value(OPERATION-PACKAGE)
6855 Package ::= value(OPERATION-PACKAGE)
6856 | type -- shall reference a package type
6857 | type -- shall reference a package type
6858 END
  
```

6859 The following definitions are used throughout this subclause:

- 6860 - v1-application-context: An application-context which contains only v1-packages and uses only TC v1 facilities;
- 6861 - v1 context set: the set of v1-application-contexts defined in the present document.
- 6862 - vn-application-context (n>=2): An application-context which contains only vn-packages;

6863 The names of v1-application-contexts are suffixed by "-v1" while other names are suffixed by "-vn" where n>=2.

6864 Application-contexts which do not belong to the v1 context set use v2 TC facilities.

6865 The last component of each application-context-name (i.e. the last component of the object identifier value) assigned to
 6866 an application-context which belongs to the v1 context set indicates explicitly "version1".

6867 For each application-context which does not belong to the "v1 context set" there is a v1-equivalent application context.

6868 This is a v1-application-context which includes the v1-equivalents of the packages included in the original context.

6869 Each application-context uses the abstract-syntax associated with the operation-packages it includes and uses the
 6870 transfer-syntax derived from it by applying the encoding rules defined in subclause 17.1.1.

6871 ACs which do not belong to the v1 context set require the support of the abstract-syntax identified by the object
 6872 identifier value: MAP-DialogueInformation.map-Dialogue-AS defined in subclause 17.4.

6873 17.3.2 Application context definitions

6874 17.3.2.1 Void

6875 17.3.2.2 Location Updating

6876 This application context is used between HLR and VLR for location updating procedures.

```
6877 networkLocUpContext-v3 APPLICATION-CONTEXT
6878 -- Responder is HLR if Initiator is VLR
6879 INITIATOR CONSUMER OF {
6880     LocationUpdatingPackage-v3,
6881     DataRestorationPackage-v3}
6882 RESPONDER CONSUMER OF {
6883     SubscriberDataMngtPackage-v3
6884     TracingPackage-v3}
6885 ::= {map-ac networkLocUp(1) version3(3)}
```

6886
6887 The following application-context-name is assigned to the v2-equivalent application-context:

```
6888 {map-ac networkLocUp(1) version2(2)}
```

6889

6890 The following application-context-name is assigned to the v1-equivalent application-context:

```
6891 {map-ac networkLocUp(1) version1(1)}
```

6892

6893 17.3.2.3 Location Cancellation

6894 This application context is used between HLR and VLR or between HLR and SGSN for location cancellation
6895 procedures. For the HLR - SGSN interface only version 3 of this application context is applicable.

```
6896 locationCancellationContext-v3 APPLICATION-CONTEXT
6897 -- Responder is VLR or SGSN if Initiator is HLR
6898 INITIATOR CONSUMER OF {
6899     LocationCancellationPackage-v3}
6900 ::= {map-ac locationCancel(2) version3(3)}
```

6901

6902 The following application-context-name is assigned to the v2-equivalent application-context:

```
6903 map-ac locationCancel(2) version2(2)
```

6904

6905 The following application-context-name is assigned to the v1-equivalent application-context:

```
6906 map-ac locationCancel(2) version1(1)
```

6907

6908 17.3.2.4 Roaming number enquiry

6909 This application context is used between HLR and VLR for roaming number enquiry procedures.

```
6910 roamingNumberEnquiryContext-v3 APPLICATION-CONTEXT
6911 -- Responder is VLR if Initiator is HLR
6912 INITIATOR CONSUMER OF {
6913     RoamingNumberEnquiryPackage-v3}
6914 ::= {map-ac roamingNbEnquiry(3) version3(3)}
```

6915

6916 The following application-context-name is assigned to the v2-equivalent application-context:

```
6917 {map-ac roamingNbEnquiry(3) version2(2)}
```

6918

6919 The following application-context-name is assigned to the v1-equivalent application-context:

```
6920 {map-ac roamingNbEnquiry(3) version1(1)}
```

6921

6922 17.3.2.5 Void

6923 17.3.2.6 Location Information Retrieval

6924 This application-context is used between GMSC and HLR or between GMSC and NPLR when retrieving location
6925 information. For the GMSC - NPLR interface version 1, version 2 and version 3 of this application context are
6926 applicable.

```
6927 locationInfoRetrievalContext-v3 APPLICATION-CONTEXT
6928 -- Responder is HLR or NPLR if Initiator is GMSC
6929 INITIATOR CONSUMER OF {
6930     InterrogationPackage-v3}
6931 ::= {map-ac locInfoRetrieval(5) version3(3)}
```

6932
6933 The following application-context-name is assigned to the v2-equivalent application-context:

```
6934 {map-ac locInfoRetrieval(5) version2(2)}
```

6935
6936

6937 The following application-context-name is assigned to the v1-equivalent application-context:

```
6938 {map-ac locInfoRetrieval(5) version1(1)}
```

6939

6940 17.3.2.7 Call control transfer

6941 This application context is used for the call control transfer procedure between the VMSC and the GMSC.

```
6942 callControlTransferContext-v4 APPLICATION-CONTEXT
6943 -- Responder is GMSC if Initiator is VMSC
6944 INITIATOR CONSUMER OF {
6945     CallControlTransferPackage-v4}
6946 ::= {map-ac callControlTransfer(6) version4(4)}
```

6947

6948 The following application-context-name is assigned to the v3-equivalent application-context:

```
6949 {map-ac callControlTransfer(6) version3(3)}
```

6950 17.3.2.8 - 17.3.2.10 Void

6951 17.3.2.11 Location registers restart

6952 This application context is used between HLR and VLR or between HLR and SGSN for location register restart
6953 procedures. For the HLR - SGSN interface version 1 and version 2 of this application context are applicable.

```
6954 resetContext-v2 APPLICATION-CONTEXT
6955 -- Responder is VLR or SGSN if Initiator is HLR
6956 INITIATOR CONSUMER OF {
6957     ResetPackage-v2}
6958 ::= {map-ac reset(10) version2(2)}
```

6959

6960 The following application-context-name is assigned to the v1-equivalent application-context:

```
6961 {map-ac reset(10) version1(1)}
```

6962

6963 17.3.2.12 Handover control

6964 This application context is used for handover procedures between MSCs.

```
6965 handoverControlContext-v2 APPLICATION-CONTEXT
6966 -- Responder is MSCB if Initiator is MSCA
6967 INITIATOR CONSUMER OF {
6968     HandoverControlPackage-v2}
6969 ::= {map-ac handoverControl(11) version2(2)}
```

6970

6971 The following application-context-name is assigned to the v1-equivalent application-context:

```
6972 {map-ac handoverControl(11) version1(1)}
```

6973

6974 17.3.2.13 IMSI Retrieval

6975 This application context is used for IMSI retrieval between HLR and VLR.

```

6976 imsiRetrievalContext-v2 APPLICATION-CONTEXT
6977   -- Responder is HLR if Initiator is VLR
6978   INITIATOR CONSUMER OF {
6979     IMSIRetrievalPackage-v2}
6980 ::= {map-ac imsiRetrieval(26) version2(2)}
6981

```

6982 This application-context is v2 only.

6983 17.3.2.14 Equipment Management

6984 This application context is used for equipment checking between MSC and EIR or between SGSN and EIR. For the
6985 SGSN - EIR interface version 1 and version 2 of this application context are applicable:

```

6986 equipmentMngtContext-v2 APPLICATION-CONTEXT
6987   -- Responder is EIR if Initiator is MSC
6988   -- Responder is EIR if Initiator is SGSN
6989   INITIATOR CONSUMER OF {
6990     EquipmentMngtPackage-v2}
6991 ::= {map-ac equipmentMngt(13) version2(2)}
6992

```

6993 The following application-context-name is assigned to the v1-equivalent application-context:

```

6994 {map-ac equipmentMngt(13) version1(1)}
6995

```

6996 17.3.2.15 Information retrieval

6997 This application context is used for authentication information retrieval between HLR and VLR or between HLR and
6998 SGSN. For the HLR - SGSN interface version 1 and version 2 and version 3 of this application context are applicable.

```

6999 infoRetrievalContext-v3 APPLICATION-CONTEXT
7000   -- Responder is HLR if Initiator is VLR
7001   -- Responder is HLR if Initiator is SGSN
7002   INITIATOR CONSUMER OF {
7003     InfoRetrievalPackage-v3}
7004 ::= {map-ac infoRetrieval(14) version3(3)}
7005

```

7006 The following application-context-name is assigned to the v2-equivalent application-context:

```

7007 infoRetrievalContext-v2 APPLICATION-CONTEXT
7008   -- Responder is HLR if Initiator is VLR
7009   -- Responder is HLR if Initiator is SGSN
7010   INITIATOR CONSUMER OF {
7011     InfoRetrievalPackage-v2}
7012 ::= {map-ac infoRetrieval(14) version2(2)}
7013

```

7014 The following application-context-name is assigned to the v1-equivalent application-context:

```

7015   -- Responder is HLR if Initiator is VLR
7016 {map-ac infoRetrieval(14) version1(1)}
7017

```

7018 17.3.2.16 Inter-VLR information retrieval

7019 This application context is used for information retrieval between VLRs.

```

7020 interVlrInfoRetrievalContext-v3 APPLICATION-CONTEXT
7021   -- Responder is VLR if Initiator is VLR
7022   INITIATOR CONSUMER OF {
7023     InterVlrInfoRetrievalPackage-v3}
7024 ::= {map-ac interVlrInfoRetrieval(15) version3(3)}
7025

```

7026 The v2-equivalent application-context is:

```
7027 interVlrInfoRetrievalContext-v2 APPLICATION-CONTEXT
7028   -- Responder is VLR if Initiator is VLR
7029   INITIATOR CONSUMER OF {
7030     InterVlrInfoRetrievalPackage-v2}
7031 ::= {map-ac interVlrInfoRetrieval(15) version2(2)}
```

7033 The v1-equivalent application-context is:

```
7034   -- Responder is VLR if Initiator is VLR
7035 {map-ac infoRetrieval(14) version1(1)}
```

7037 17.3.2.17 Stand Alone Subscriber Data Management

7038 This application context is used for stand alone subscriber data management between HLR and VLR or between HLR
7039 and SGSN. For the HLR - SGSN interface only version 3 of this application context is applicable:

```
7040 subscriberDataMngtContext-v3 APPLICATION-CONTEXT
7041   -- Responder is VLR or SGSN if Initiator is HLR
7042   INITIATOR CONSUMER OF {
7043     SubscriberDataMngtStandAlonePackage-v3}
7044 ::= {map-ac subscriberDataMngt(16) version3(3)}
```

7046 The following application-context-name is assigned to the v2-equivalent application-context:

```
7047 {map-ac subscriberDataMngt(16) version2(2)}
```

7049 The following application-context-name is assigned to the v1-equivalent application-context:

```
7050 {map-ac subscriberDataMngt(16) version1(1)}
```

7052 17.3.2.18 Tracing

7053 This application context is used between HLR and VLR or between HLR and SGSN for stand alone tracing control
7054 procedures: For the HLR - SGSN interface version 1, version 2 and version 3 of this application context are applicable.

```
7055 tracingContext-v3 APPLICATION-CONTEXT
7056   -- Responder is VLR or SGSN if Initiator is HLR
7057   INITIATOR CONSUMER OF {
7058     TracingStandAlonePackage-v3}
7059 ::= {map-ac tracing(17) version3(3)}
```

7061 The following application-context-name is assigned to the v2-equivalent application-context:

```
7062 {map-ac tracing(17) version2(2)}
```

7064 The following application-context-name is assigned to the v1-equivalent application-context:

```
7065 {map-ac tracing(17) version1(1)}
```

7067 17.3.2.19 Network functional SS handling

7068 This application context is used for functional-like SS handling procedures between VLR and HLR.

```
7069 networkFunctionalSsContext-v2 APPLICATION-CONTEXT
7070   -- Responder is HLR, Initiator is VLR
7071   INITIATOR CONSUMER OF {
7072     FunctionalSsPackage-v2}
7073 ::= {map-ac networkFunctionalSs(18) version2(2)}
```

7074

7075 The v1-equivalent application-context is defined as follows:

```

7076 networkFunctionalSsContext-v1 APPLICATION-CONTEXT
7077   -- Responder is HLR, Initiator is VLR
7078   INITIATOR CONSUMER OF {
7079     FunctionalSsPackage-v1,
7080     UnstructuredSsPackage-v1,
7081     BindingPackage-v1}
7082 ::= {map-ac networkFunctionalSs(18) version1(1)}
7083

```

7084 17.3.2.20 Network unstructured SS handling

7085 This application context is used for handling stimuli-like procedures between HLR and VLR, between the HLR and
7086 gsmSCF, and between HLR and HLR.

```

7087 networkUnstructuredSsContext-v2 APPLICATION-CONTEXT
7088   -- Responder is HLR, Initiator is VLR
7089   -- Responder is VLR, Initiator is HLR
7090   -- Responder is gsmSCF, Initiator is HLR
7091   -- Responder is HLR, Initiator is gsmSCF
7092   -- Responder is HLR, Initiator is HLR
7093   OPERATIONS OF {
7094     UnstructuredSsPackage-v2}
7095 ::= {map-ac networkUnstructuredSs(19) version2(2)}
7096

```

7097 The following application-context-name is assigned to the v1-equivalent application-context:

```

7098 {map-ac networkFunctionalSs(18) version1(1)}
7099

```

7100 17.3.2.21 Short Message Gateway

7101 This application context is used for short message gateway procedures.

```

7102 shortMsgGatewayContext-v3 APPLICATION-CONTEXT
7103   -- Responder is HLR if Initiator is GMSC
7104   INITIATOR CONSUMER OF {
7105     ShortMsgGatewayPackage-v3}
7106 ::= {map-ac shortMsgGateway(20) version3(3)}
7107

```

7108 The following application-context-name is assigned to the v2-equivalent application-context:

```

7109 {map-ac shortMsgGateway(20) version2(2)}
7110

```

7111 The following application-context-name is assigned to the v1-equivalent application-context:

```

7112 {map-ac shortMsgGateway(20) version1(1)}
7113

```

7114 17.3.2.22 Mobile originating Short Message Relay

7115 This application context is used between MSC and IWMSC or between SGSN and IWMSC for mobile originating short
7116 message relay procedures. For the SGSN - IWMSC interface version 1, version 2 and version 3 of this application
7117 context are applicable.

```

7118 shortMsgMO-RelayContext-v3 APPLICATION-CONTEXT
7119   -- Responder is IWMSC if Initiator is MSC
7120   -- Responder is IWMSC if Initiator is SGSN
7121   INITIATOR CONSUMER OF {
7122     MOShortMsgRelayPackage-v3}
7123 ::= {map-ac shortMsgMO-Relay(21) version3(3)}
7124

```

7125 The following application-context-name is assigned to the v2-equivalent application-context:

```

7126 {map-ac shortMsgMO-Relay(21) version2(2)}
7127

```

7128 The following application-context-name is assigned to the v1-equivalent application-context:

```

7129 {map-ac shortMsg-Relay(21) version1(1)}
7130

```


7131 17.3.2.23 Void

7132 17.3.2.24 Short message alert

7133 This application context is used for short message alerting procedures.

```
7134 shortMsgAlertContext-v2 APPLICATION-CONTEXT
7135 -- Responder is IWMSC if Initiator is HLR
7136 INITIATOR CONSUMER OF {
7137     AlertingPackage-v2}
7138 ::= {map-ac shortMsgAlert(23) version2(2)}
```

7139
7140 The following application-context-name is symbolically assigned to the v1-equivalent application-context:

```
7141 {map-ac shortMsgAlert(23) version1(1)}
```

7142

7143 17.3.2.25 Short message waiting data management

7144 This application context is used between VLR and HLR or between SGSN and HLR for short message waiting data
7145 management procedures. For the SGSN - HLR interface only version 3 of this application context is applicable.

```
7146 mwdMngtContext-v3 APPLICATION-CONTEXT
7147 -- Responder is HLR if Initiator is SGSN
7148 -- Responder is HLR if Initiator is VLR
7149 INITIATOR CONSUMER OF {
7150     MwdMngtPackage-v3}
7151 ::= {map-ac mwdMngt(24) version3(3)}
```

7152

7153 The following application-context-name is assigned to the v2-equivalent application-context:

```
7154 {map-ac mwdMngt(24) version2(2)}
```

7155

7156 The following application-context-name is assigned to the v1-equivalent application-context:

```
7157 {map-ac mwdMngt(24) version1(1)}
```

7158

7159 17.3.2.26 Mobile terminating Short Message Relay

7160 This application context is used between GMSC and MSC or between GMSC and SGSN for mobile terminating short
7161 message relay procedures. For the GMSC - SGSN interface version 2 and version 3 of this application context and the
7162 equivalent version 1 application context are applicable.

```
7163 shortMsgMT-RelayContext-v3 APPLICATION-CONTEXT
7164 -- Responder is MSC or SGSN if Initiator is GMSC
7165 INITIATOR CONSUMER OF {
7166     MTShortMsgRelayPackage-v3}
7167 ::= {map-ac shortMsgMT-Relay(25) version3(3)}
```

7168

7169 The following application-context-name is assigned to the v2-equivalent application-context:

```
7170 {map-ac shortMsgMT-Relay(25) version2(2)}
```

7171

7172 The following application-context-name is assigned to the v1-equivalent application-context:

```
7173 {map-ac shortMsgMO-Relay(21) version1(1)}
```

7174

7175 17.3.2.27 MS purging

7176 This application context is used between HLR and VLR or between HLR and SGSN for MS purging procedures. For the
7177 SGSN - HLR interface only version 3 of this application context is applicable.

```
7178 msPurgingContext-v3 APPLICATION-CONTEXT
7179   -- Responder is HLR if Initiator is VLR
7180   -- Responder is HLR if Initiator is SGSN
7181   INITIATOR CONSUMER OF {
7182     purgingPackage-v3}
7183 ::= {map-ac msPurging(27) version3(3)}
```

7184
7185 The following application-context-name is assigned to the v2-equivalent application-context:

```
7186 {map-ac msPurging(27) version2(2)}
```

7187

7188 17.3.2.28 Subscriber information enquiry

7189 This application context is used between HLR and VLR for subscriber information enquiry procedures.

```
7190 subscriberInfoEnquiryContext-v3 APPLICATION-CONTEXT
7191   -- Responder is VLR if Initiator is HLR
7192   INITIATOR CONSUMER OF {
7193     SubscriberInformationEnquiryPackage-v3}
7194 ::= {map-ac subscriberInfoEnquiry(28) version3(3)}
```

7195

7196 This application-context is v3 only.

7197 17.3.2.29 Any time information enquiry

7198 This application context is used between gsmSCF and HLR or GMLC for any time information enquiry procedures.

```
7199 anyTimeInfoEnquiryContext-v3 APPLICATION-CONTEXT
7200   -- Responder is HLR or GMLC if Initiator is gsmSCF
7201   INITIATOR CONSUMER OF {
7202     AnyTimeInformationEnquiryPackage-v3}
7203 ::= {map-ac anyTimeInfoEnquiry(29) version3(3)}
```

7204

7205 This application-context is v3 only.

7206 17.3.2.30 Group Call Control

7207 This application context is used between anchor MSC and relay MSC for group call and broadcast call procedures.

```
7208 groupCallControlContext-v3 APPLICATION-CONTEXT
7209   -- Responder is relay MSC if Initiator is anchor MSC
7210   INITIATOR CONSUMER OF {
7211     GroupCallControlPackage-v3}
7212 ::= {map-ac groupCallControl(31) version3(3)}
```

7213

7214 This application-context is v3 only.

7215 17.3.2.31 Provide SIWFS Number

7216 This application context is used for activation or modification of SIWF resources.

```
7217 sIWFSAllocationContext-v3 APPLICATION-CONTEXT
7218   -- Responder is SIWF if Initiator is VMSC
7219   INITIATOR CONSUMER OF {
7220     ProvideSIWFSNumberPackage-v3,
7221     SIWFSSignallingModifyPackage-v3}
7222 ::= {map-ac sIWFSAllocation(12) version3(3)}
```

7223

7224 This application-context is v3 only.

7225 17.3.2.32 Gprs Location Updating

7226 This application context is used between HLR and SGSN for gprs location updating procedures.

```
7227 gprsLocationUpdateContext-v3 APPLICATION-CONTEXT
7228 -- Responder is HLR if Initiator is SGSN
7229 INITIATOR CONSUMER OF {
7230     GprsLocationUpdatingPackage-v3}
7231 RESPONDER CONSUMER OF {
7232     SubscriberDataMngtPackage-v3
7233     TracingPackage-v3}
7234 ::= {map-ac gprsLocationUpdate(32) version3(3)}
```

7235
7236 This application-context is v3 only.

7237 17.3.2.33 Gprs Location Information Retrieval

7238 This application context is used between HLR and GGSN when retrieving gprs location information.

```
7239 gprsLocationInfoRetrievalContext-v3 APPLICATION-CONTEXT
7240 -- Responder is HLR if Initiator is GGSN
7241 INITIATOR CONSUMER OF {
7242     GprsInterrogationPackage-v3}
7243 ::= {map-ac gprsLocationInfoRetrieval(33) version3(3)}
```

7244
7245 This application-context is v3 only.

7246 17.3.2.34 Failure Reporting

7247 This application context is used between HLR and GGSN to inform that network requested PDP-context activation has failed.

```
7249 failureReportContext-v3 APPLICATION-CONTEXT
7250 -- Responder is HLR if Initiator is GGSN
7251 INITIATOR CONSUMER OF {
7252     FailureReportingPackage-v3}
7253 ::= {map-ac failureReport(34) version3(3)}
```

7254
7255 This application-context is v3 only.

7256 17.3.2.35 GPRS Notifying

7257 This application context is used between HLR and GGSN for notifying that GPRS subscriber is present again.

```
7258 gprsNotifyContext-v3 APPLICATION-CONTEXT
7259 -- Responder is GGSN if Initiator is HLR
7260 INITIATOR CONSUMER OF {
7261     GprsNotifyingPackage-v3}
7262 ::= {map-ac gprsNotify(35) version3(3)}
```

7263
7264 This application-context is v3 only.

7265 17.3.2.36 Supplementary Service invocation notification

7266 This application context is used between the MSC and the gsmSCF and between the HLR and the gsmSCF for Supplementary Service invocation notification procedures.

```
7268 ss-InvocationNotificationContext-v3 APPLICATION-CONTEXT
7269 -- Responder is gsmSCF, Initiator is MSC
7270 -- Responder is gsmSCF, Initiator is HLR
7271 INITIATOR CONSUMER OF {
7272     SS-InvocationNotificationPackage-v3}
7273 ::= {map-ac ss-InvocationNotification(36) version3(3)}
```

7274
7275 This application-context is v3 only.

7276 17.3.2.37 Reporting

7277 This application context is used between HLR and VLR for reporting procedures.

```
7278 reportingContext-v3 APPLICATION-CONTEXT
7279   -- Responder is VLR if Initiator is HLR
7280   -- Responder is HLR if Initiator is VLR
7281   INITIATOR CONSUMER OF {
7282     SetReportingStatePackage-v3,
7283     StatusReportPackage-v3,
7284     RemoteUserFreePackage-v3}
7285   RESPONDER CONSUMER OF {
7286     SetReportingStatePackage-v3,
7287     StatusReportPackage-v3}
7288 ::= {map-ac reporting(7) version3(3)}
7289
```

7290 This application-context is v3 only.

7291 17.3.2.38 Call Completion

7292 This application context is used between VLR and the HLR for subscriber control of call completion services.

```
7293 callCompletionContext-v3 APPLICATION-CONTEXT
7294   -- Responder is HLR if Initiator is VLR
7295   INITIATOR CONSUMER OF {
7296     CallCompletionPackage-v3}
7297 ::= {map-ac callCompletion(8) version3(3)}
7298
```

7299 This application-context is v3 only.

7300 17.3.2.39 Location Service Gateway

7301 This application context is used for location service gateway procedures.

```

7302 locationSvcGatewayContext-v3 APPLICATION-CONTEXT
7303 -- Responder is HLR if Initiator is GMLC
7304 INITIATOR CONSUMER OF {
7305     locationSvcGatewayPackage-v3}
7306 ::= {map-ac locationSvcGateway(37) version3(3)}

```

7307

7308 17.3.2.40 Location Service Enquiry

7309 This application context is used for location service enquiry procedures.

```

7310 locationSvcEnquiryContext-v3 APPLICATION-CONTEXT
7311 -- Responder is MSC if Initiator is GMLC
7312 INITIATOR CONSUMER OF {
7313     locationSvcEnquiryPackage-v3}
7314 ::= {map-ac locationScvEnquiry(38) version3 (3)}

```

7315

7316 17.3.2.41 Void

7317 17.3.2.42 Void

7318 17.3.2.43 Void

7319 17.3.2.44 IST Alerting

7320 This application context is used between MSC (Visited MSC or Gateway MSC) and HLR for for alerting services within
7321 IST procedures.

```

7322 istAlertingContext-v3 APPLICATION-CONTEXT
7323 -- Responder is HLR if Initiator is VMSC
7324 -- Responder is HLR if Initiator is GMSC
7325 INITIATOR CONSUMER OF {
7326     IST-AlertingPackage-v3}
7327 ::= {map-ac alerting (4) version3(3)}

```

7328

7329 This application-context is v3 only.

7330 17.3.2.45 Service Termination

7331 This application context is used between HLR and MSC (Visited MSC or Gateway MSC) for service termination
7332 services within IST procedures.

```

7333 serviceTerminationContext-v3 APPLICATION-CONTEXT
7334 -- Responder is VMSC or GMSC if Initiator is HLR
7335 INITIATOR CONSUMER OF {
7336     ServiceTerminationPackage-v3}
7337 ::= {map-ac serviceTermination (9) version3(3)}

```

7338

7339 This application-context is v3 only.

7340 17.3.2.46 Mobility Management event notification

7341 This application context is used between VLR and gsmSCF for Mobility Management event notification procedures.

```

7342 mm-EventReportingContext-v3 APPLICATION-CONTEXT
7343 -- Responder is gsmSCF, Initiator is VLR
7344 INITIATOR CONSUMER OF {
7345     MM-EventReportingPackage-v3}
7346 ::= {map-ac mm-EventReporting(42) version3(3)}

```

7347

7348 This application-context is v3 only.

7349 17.3.2.47 Any time information handling

7350 This application context is used between gsmSCF and HLR for any time information handling procedures.

```

7351 anyTimeInfohandlingContext-v3 APPLICATION-CONTEXT
7352   -- Responder is HLR if Initiator is gsmSCF
7353   INITIATOR CONSUMER OF {
7354     AnyTimeInformationHandlingPackage-v3}
7355 ::= {map-ac anyTimeInfoHandling(43) version3(3)}
7356

```

7357 This application-context is v3 only.

17.3.2.48 Subscriber Data modification notification

This application context is used between HLR and gsmSCF for Subscriber Data modification notification procedures.

```

subscriberDataModificationNotificationContext-v3 APPLICATION-CONTEXT
  -- Responder is gsmSCF, Initiator is HLR
  INITIATOR CONSUMER OF {
    SubscriberDataModificationNotificationPackage-v3}
  ::= {map-ac subscriberDataModificationNotification(22) version3(3)}

```

This application-context is v3 only.

17.3.3 ASN.1 Module for application-context-names

The following ASN.1 module summarizes the application-context-name assigned to MAP application-contexts.

```

7358 MAP-ApplicationContexts {
7359   ccitt identified-organization (4) etsi (0) mobileDomain (0)
7360   gsm-Network (1) modules (3) map-ApplicationContexts (2) version6 (6)}
7361
7362 DEFINITIONS
7363 ::=
7364 BEGIN
7365
7366 -- EXPORTS everything
7367
7368 IMPORTS
7369   gsm-NetworkId,
7370   ac-Id
7371 FROM MobileDomainDefinitions {
7372   ccitt (0) identified-organization (4) etsi (0) mobileDomain (0)
7373   mobileDomainDefinitions (0) version1 (1)}
7374 ;
7375
7376 -- application-context-names
7377
7378 map-ac OBJECT IDENTIFIER ::= {gsm-NetworkId ac-Id}
7379
7380 networkLocUpContext-v3 OBJECT IDENTIFIER ::=
7381   {map-ac networkLocUp(1) version3(3)}
7382
7383 locationCancellationContext-v3 OBJECT IDENTIFIER ::=
7384   {map-ac locationCancel(2) version3(3)}
7385
7386 roamingNumberEnquiryContext-v3 OBJECT IDENTIFIER ::=
7387   {map-ac roamingNbEnquiry(3) version3(3)}
7388
7389 locationInfoRetrievalContext-v3 OBJECT IDENTIFIER ::=
7390   {map-ac locInfoRetrieval(5) version3(3)}
7391
7392 resetContext-v2 OBJECT IDENTIFIER ::=
7393   {map-ac reset(10) version2(2)}
7394
7395 handoverControlContext-v2 OBJECT IDENTIFIER ::=
7396   {map-ac handoverControl(11) version2(2)}
7397
7400
7401

```

7402	equipmentMngtContext-v2 OBJECT IDENTIFIER ::=
7403	{map-ac equipmentMngt(13) version2(2)}
7404	
7405	infoRetrievalContext-v3 OBJECT IDENTIFIER ::=
7406	{map-ac infoRetrieval(14) version3(3)}
7407	
7408	interVlrInfoRetrievalContext-v3 OBJECT IDENTIFIER ::=
7409	{map-ac interVlrInfoRetrieval(15) version3(3)}
7410	
7411	subscriberDataMngtContext-v3 OBJECT IDENTIFIER ::=
7412	{map-ac subscriberDataMngt(16) version3(3)}
7413	
7414	tracingContext-v3 OBJECT IDENTIFIER ::=
7415	{map-ac tracing(17) version3(3)}
7416	
7417	networkFunctionalSsContext-v2 OBJECT IDENTIFIER ::=
7418	{map-ac networkFunctionalSs(18) version2(2)}
7419	
7420	networkUnstructuredSsContext-v2 OBJECT IDENTIFIER ::=
7421	{map-ac networkUnstructuredSs(19) version2(2)}
7422	
7423	shortMsgGatewayContext-v3 OBJECT IDENTIFIER ::=
7424	{map-ac shortMsgGateway(20) version3(3)}
7425	
7426	shortMsgMO-RelayContext-v3 OBJECT IDENTIFIER ::=
7427	{map-ac shortMsgMO-Relay(21) version3(3)}
7428	
7429	shortMsgAlertContext-v2 OBJECT IDENTIFIER ::=
7430	{map-ac shortMsgAlert(23) version2(2)}
7431	
7432	mwdMngtContext-v3 OBJECT IDENTIFIER ::=
7433	{map-ac mwdMngt(24) version3(3)}
7434	
7435	shortMsgMT-RelayContext-v3 OBJECT IDENTIFIER ::=
7436	{map-ac shortMsgMT-Relay(25) version3(3)}
7437	
7438	imsiRetrievalContext-v2 OBJECT IDENTIFIER ::=
7439	{map-ac imsiRetrieval(26) version2(2)}
7440	
7441	msPurgingContext-v3 OBJECT IDENTIFIER ::=
7442	{map-ac msPurging(27) version3(3)}
7443	
7444	subscriberInfoEnquiryContext-v3 OBJECT IDENTIFIER ::=
7445	{map-ac subscriberInfoEnquiry(28) version3(3)}
7446	
7447	anyTimeInfoEnquiryContext-v3 OBJECT IDENTIFIER ::=
7448	{map-ac anyTimeInfoEnquiry(29) version3(3)}
7449	
7450	callControlTransferContext-v4 OBJECT IDENTIFIER ::=
7451	{map-ac callControlTransfer(6) version4(4)}
7452	
7453	ss-InvocationNotificationContext-v3 OBJECT IDENTIFIER ::=
7454	{map-ac ss-InvocationNotification(36) version3(3)}
7455	
7456	sIWFSAllocationContext-v3 OBJECT IDENTIFIER ::=
7457	{map-ac sIWFSAllocation(12) version3(3)}
7458	
7459	groupCallControlContext-v3 OBJECT IDENTIFIER ::=
7460	{map-ac groupCallControl(31) version3(3)}
7461	
7462	gprsLocationUpdateContext-v3 OBJECT IDENTIFIER ::=
7463	{map-ac gprsLocationUpdate(32) version3(3)}
7464	
7465	gprsLocationInfoRetrievalContext-v3 OBJECT IDENTIFIER ::=
7466	{map-ac gprsLocationInfoRetrieval(33) version3(3)}
7467	
7468	failureReportContext-v3 OBJECT IDENTIFIER ::=
7469	{map-ac failureReport(34) version3(3)}
7470	
7471	gprsNotifyContext-v3 OBJECT IDENTIFIER ::=
7472	{map-ac gprsNotify(35) version3(3)}
7473	
7474	reportingContext-v3 OBJECT IDENTIFIER ::=
7475	{map-ac reporting(7) version3(3)}
7476	

```

7477 callCompletionContext-v3 OBJECT IDENTIFIER ::=
7478     {map-ac callCompletion(8) version3(3)}
7479
7480 istAlertingContext-v3 OBJECT IDENTIFIER ::=
7481     {map-ac istAlerting(4) version3(3)}
7482
7483 serviceTerminationContext-v3 OBJECT IDENTIFIER ::=
7484     {map-ac immediateTermination(9) version3(3)}
7485
7486 locationSvcGatewayContext-v3 OBJECT IDENTIFIER ::=
7487     {map-ac locationSvcGateway(37) version3(3)}
7488
7489 locationSvcEnquiryContext-v3 OBJECT IDENTIFIER ::=
7490     {map-ac locationSvcEnquiry(38) version3(3)}
7491
7492
7493 mm-EventReportingContext-v3 OBJECT IDENTIFIER ::=
7494     {map-ac mm-EventReporting(42) version3(3)}
7495
7496 anyTimeInfoHandlingContext-v3 OBJECT IDENTIFIER ::=
7497     {map-ac anyTimeInfoHandling(43) version3(3)}
7498
7499 subscriberDataModificationNotificationContext-v3 OBJECT IDENTIFIER ::=
7500     {map-ac subscriberDataModificationNotification(22) version3(3)}
7501
7502 -- The following Object Identifiers are reserved for application-
7503 -- contexts existing in previous versions of the protocol
7504
7505 -- AC Name & Version                Object Identifier
7506 --
7507 -- networkLocUpContext-v1           map-ac networkLocUp (1)          version1 (1)
7508 -- networkLocUpContext-v2           map-ac networkLocUp (1)          version2 (2)
7509 -- locationCancellationContext-v1    map-ac locationCancellation (2)   version1 (1)
7510 -- locationCancellationContext-v2    map-ac locationCancellation (2)   version2 (2)
7511 -- roamingNumberEnquiryContext-v1    map-ac roamingNumberEnquiry (3)   version1 (1)
7512 -- roamingNumberEnquiryContext-v2    map-ac roamingNumberEnquiry (3)   version2 (2)
7513 -- locationInfoRetrievalContext-v1    map-ac locationInfoRetrieval (5)  version1 (1)
7514 -- locationInfoRetrievalContext-v2    map-ac locationInfoRetrieval (5)  version2 (2)
7515 -- resetContext-v1                   map-ac reset (10)                 version1 (1)
7516 -- handoverControlContext-v1         map-ac handoverControl (11)       version1 (1)
7517 -- equipmentMngtContext-v1           map-ac equipmentMngt (13)         version1 (1)
7518 -- infoRetrievalContext-v1           map-ac infoRetrieval (14)         version1 (1)
7519 -- infoRetrievalContext-v2           map-ac infoRetrieval (14)         version2 (2)
7520 -- interVlrlInfoRetrievalContext-v2  map-ac interVlrlInfoRetrieval (15) version2 (2)
7521 -- subscriberDataMngtContext-v1      map-ac subscriberDataMngt (16)    version1 (1)
7522 -- subscriberDataMngtContext-v2      map-ac subscriberDataMngt (16)    version2 (2)
7523 -- tracingContext-v1                 map-ac tracing (17)               version1 (1)
7524 -- tracingContext-v2                 map-ac tracing (17)               version2 (2)
7525 -- networkFunctionalSsContext-v1      map-ac networkFunctionalSs (18)    version1 (1)
7526 -- shortMsgGatewayContext-v1         map-ac shortMsgGateway (20)       version1 (1)
7527 -- shortMsgGatewayContext-v2         map-ac shortMsgGateway (20)       version2 (2)
7528 -- shortMsgRelayContext-v1           map-ac shortMsgRelay (21)         version1 (1)
7529 -- shortMsgAlertContext-v1           map-ac shortMsgAlert (23)         version1 (1)
7530 -- mwdMngtContext-v1                 map-ac mwdMngt (24)               version1 (1)
7531 -- mwdMngtContext-v2                 map-ac mwdMngt (24)               version2 (2)
7532 -- shortMsgMT-RelayContext-v2        map-ac shortMsgMT-Relay (25)      version2 (2)
7533 -- msPurgingContext-v2               map-ac msPurging (27)             version2 (2)
7534 -- callControlTransferContext-v3     map-ac callControlTransferContext (6) version3 (3)
7535
7536
7537 END

```

7538 17.4 MAP Dialogue Information

```

7539 MAP-DialogueInformation {
7540     ccitt identified-organization (4) etsi (0) mobileDomain (0)
7541     gsm-Network (1) modules (3) map-DialogueInformation (3) version6 (6)}
7542
7543 DEFINITIONS
7544
7545 IMPLICIT TAGS
7546
7547 ::=
7548
7549 BEGIN
7550

```



```

7551 EXPORTS
7552     map-DialogueAS,
7553     MAP-DialoguePDU
7554 ;
7555
7556 IMPORTS
7557     gsm-NetworkId,
7558     as-Id
7559 FROM MobileDomainDefinitions {
7560     ccitt identified-organization (4) etsi (0) mobileDomain (0)
7561     mobileDomainDefinitions (0) version1 (1)}
7562
7563     AddressString
7564 FROM MAP-CommonDataTypes {
7565     ccitt identified-organization (4) etsi (0) mobileDomain (0)
7566     gsm-Network(1) modules (3) map-CommonDataTypes (18) version6 (6)}
7567
7568     ExtensionContainer
7569 FROM MAP-ExtensionDataTypes {
7570     ccitt identified-organization (4) etsi (0) mobileDomain (0)
7571     gsm-Network (1) modules (3) map-ExtensionDataTypes (21) version6 (6)}
7572
7573 ;
7574
7575
7576 -- abstract syntax name for MAP-DialoguePDU
7577

```

```

7578 map-DialogueAS OBJECT IDENTIFIER ::=
7579     {gsm-NetworkId as-Id map-DialoguePDU (1) version1 (1)}
7580

```

```

7581 MAP-DialoguePDU ::= CHOICE {
7582     map-open [0] MAP-OpenInfo,
7583     map-accept [1] MAP-AcceptInfo,
7584     map-close [2] MAP-CloseInfo,
7585     map-refuse [3] MAP-RefuseInfo,
7586     map-userAbort [4] MAP-UserAbortInfo,
7587     map-providerAbort [5] MAP-ProviderAbortInfo}
7588

```

```

7589 MAP-OpenInfo ::= SEQUENCE {
7590     destinationReference [0] AddressString OPTIONAL,
7591     originationReference [1] AddressString OPTIONAL,
7592     ...,
7593     extensionContainer ExtensionContainer OPTIONAL
7594     -- extensionContainer must not be used in version 2
7595 }
7596

```

```

7597 MAP-AcceptInfo ::= SEQUENCE {
7598     ...,
7599     extensionContainer ExtensionContainer OPTIONAL
7600     -- extensionContainer must not be used in version 2
7601 }
7602

```

```

7603 MAP-CloseInfo ::= SEQUENCE {
7604     ...,
7605     extensionContainer ExtensionContainer OPTIONAL
7606     -- extensionContainer must not be used in version 2
7607 }
7608

```

```

7609 MAP-RefuseInfo ::= SEQUENCE {
7610     reason Reason,
7611     ...,
7612     extensionContainer ExtensionContainer OPTIONAL
7613     -- extensionContainer must not be used in version 2
7614 }
7615

```

```

7616 Reason ::= ENUMERATED {
7617     noReasonGiven (0),
7618     invalidDestinationReference (1),
7619     invalidOriginatingReference (2)}
7620

```

```

7621 MAP-UserAbortInfo ::= SEQUENCE {
7622     map-UserAbortChoice MAP-UserAbortChoice,
7623     ...,
7624     extensionContainer ExtensionContainer OPTIONAL
7625     -- extensionContainer must not be used in version 2
7626 }
7627

```

```

7628 MAP-UserAbortChoice ::= CHOICE {
7629     userSpecificReason          [0] NULL,
7630     userResourceLimitation      [1] NULL,
7631     resourceUnavailable         [2] ResourceUnavailableReason,
7632     applicationProcedureCancellation [3] ProcedureCancellationReason}
7633
7634 ResourceUnavailableReason ::= ENUMERATED {
7635     shortTermResourceLimitation (0),
7636     longTermResourceLimitation (1)}
7637
7638 ProcedureCancellationReason ::= ENUMERATED {
7639     handoverCancellation (0),
7640     radioChannelRelease (1),
7641     networkPathRelease (2),
7642     callRelease (3),
7643     associatedProcedureFailure (4),
7644     tandemDialogueRelease (5),
7645     remoteOperationsFailure (6)}
7646
7647 MAP-ProviderAbortInfo ::= SEQUENCE {
7648     map-ProviderAbortReason          MAP-ProviderAbortReason,
7649     ...,
7650     extensionContainer              ExtensionContainer              OPTIONAL
7651     -- extensionContainer must not be used in version 2
7652 }
7653
7654 MAP-ProviderAbortReason ::= ENUMERATED {
7655     abnormalDialogue (0),
7656     invalidPDU (1)}
7657
7658 END

```

17.5 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) version6 (6)}
4
5  DEFINITIONS
6
7  ::=
8
9  BEGIN
10
11  IMPORTS
12  UpdateLocation,
13  CancelLocation,
14  PurgeMS,
15  SendIdentification,
16  UpdateGprsLocation,
17  PrepareHandover,
18  SendEndSignal,
19  ProcessAccessSignalling,
20  ForwardAccessSignalling,
21  PrepareSubsequentHandover,
22  SendAuthenticationInfo,
23  CheckIMEI,
24  InsertSubscriberData,
25  DeleteSubscriberData,
26  Reset,
27  ForwardCheckSS-Indication,
28  RestoreData,
29  ProvideSubscriberInfo,
30  AnyTimeInterrogation,
31  AnyTimeSubscriptionInterrogation,
32  AnyTimeModification,
33  SendRoutingInfoForGprs,
34  FailureReport,
35  NoteMsPresentForGprs,
36  NoteMM-Event,
37  NoteSubscriberDataModified
38
39
40  FROM MAP-MobileServiceOperations {
41  ccitt identified-organization (4) etsi (0) mobileDomain (0)
42  gsm-Network (1) modules (3) map-MobileServiceOperations (5)
43  version6 (6)}

```

```
44
45   ActivateTraceMode,
46   DeactivateTraceMode,
47   SendIMSI
48 FROM MAP-OperationAndMaintenanceOperations {
49   ccitt identified-organization (4) etsi (0) mobileDomain (0)
50   gsm-Network (1) modules (3) map-OperationAndMaintenanceOperations (6)
51   version6 (6)}
52
53   SendRoutingInfo,
54   ProvideRoamingNumber,
55   ResumeCallHandling,
56   ProvideSIWFSSNumber,
57   SIWFSSignallingModify,
58   SetReportingState,
59   StatusReport,
60   RemoteUserFree,
61   IST-Alert,
62   IST-Command
63 FROM MAP-CallHandlingOperations {
64   ccitt identified-organization (4) etsi (0) mobileDomain (0)
65   gsm-Network (1) modules (3) map-CallHandlingOperations (7)
66   version6 (6)}
67
68   RegisterSS,
69   EraseSS,
70   ActivateSS,
71   DeactivateSS,
72   InterrogateSS,
73   ProcessUnstructuredSS-Request,
74   UnstructuredSS-Request,
75   UnstructuredSS-Notify,
76   RegisterPassword,
77   GetPassword,
78   SS-InvocationNotification,
79   RegisterCC-Entry,
80   EraseCC-Entry
81 FROM MAP-SupplementaryServiceOperations {
82   ccitt identified-organization (4) etsi (0) mobileDomain (0)
83   gsm-Network (1) modules (3) map-SupplementaryServiceOperations (8)
84   version6 (6)}
85
86   SendRoutingInfoForSM,
87   MO-ForwardSM,
88   MT-ForwardSM,
89   ReportSM-DeliveryStatus,
90   AlertServiceCentre,
91   InformServiceCentre,
92   ReadyForSM
93 FROM MAP-ShortMessageServiceOperations {
94   ccitt identified-organization (4) etsi (0) mobileDomain (0)
95   gsm-Network (1) modules (3) map-ShortMessageServiceOperations (9)
96   version6 (6)}
97
98   PrepareGroupCall,
99   ProcessGroupCallSignalling,
100  ForwardGroupCallSignalling,
101  SendGroupCallEndSignal
102 FROM MAP-Group-Call-Operations {
103   ccitt identified-organization (4) etsi (0) mobileDomain (0)
104   gsm-Network (1) modules (3) map-Group-Call-Operations (22)
105   version6 (6)}
106
107   ProvideSubscriberLocation,
108   SendRoutingInfoForLCS,
109   SubscriberLocationReport
110 FROM MAP-LocationServiceOperations {
111   ccitt identified-organization (4) etsi (0) mobileDomain (0)
112   gsm-Network (1) modules (3) map-LocationServiceOperations (24)
113   version6 (6)}
114
115   SystemFailure,
116   DataMissing,
117   UnexpectedDataValue,
118   FacilityNotSupported,
119   UnknownSubscriber,
120   NumberChanged,
121   UnknownMSC,
122   UnidentifiedSubscriber,
```

```

123     UnknownEquipment,
124     RoamingNotAllowed,
125     IllegalSubscriber,
126     IllegalEquipment,
127     BearerServiceNotProvisioned,
128     TeleserviceNotProvisioned,
129     NoHandoverNumberAvailable,
130     SubsequentHandoverFailure,
131     TracingBufferFull,
132     OR-NotAllowed,
133     NoRoamingNumberAvailable,
134     AbsentSubscriber,
135     BusySubscriber,
136     NoSubscriberReply,
137     CallBarred,
138     ForwardingViolation,
139     ForwardingFailed,
140     CUG-Reject,
141     ATI-NotAllowed,
142     IllegalSS-Operation,
143     SS-ErrorStatus,
144     SS-NotAvailable,
145     SS-SubscriptionViolation,
146     SS-Incompatibility,
147     UnknownAlphabet,
148     USSD-Busy,
149     PW-RegistrationFailure,
150     NegativePW-Check,
151     NumberOfPW-AttemptsViolation,
152     SubscriberBusyForMT-SMS,
153     SM-DeliveryFailure,
154     MessageWaitingListFull,
155     AbsentSubscriberSM,
156     ResourceLimitation,
157     NoGroupCallNumberAvailable,
158     ShortTermDenial,
159     LongTermDenial,
160     IncompatibleTerminal,
161     UnauthorizedRequestingNetwork,
162     UnauthorizedLCSCClient,
163     PositionMethodFailure,
164     UnknownOrUnreachableLCSCClient,
165     ATSI-NotAllowed,
166     ATM-NotAllowed,
167     InformationNotAvailable,
168     MM-EventNotSupported
169
170 FROM MAP-Errors {
171     ccitt identified-organization (4) etsi (0) mobileDomain (0)
172     gsm-Network (1) modules (3) map-Errors (10) version6 (6)}
173 ;
174
175
176 -- location registration operation codes
177
178 updateLocation UpdateLocation ::= localValue 2
179 cancelLocation CancelLocation ::= localValue 3
180 purgeMS PurgeMS ::= localValue 67
181 sendIdentification SendIdentification ::= localValue 55
182
183
184 -- handover operation codes
185
186 prepareHandover PrepareHandover ::= localValue 68
187 sendEndSignal SendEndSignal ::= localValue 29
188 processAccessSignalling ProcessAccessSignalling ::= localValue 33
189 forwardAccessSignalling ForwardAccessSignalling ::= localValue 34
190 prepareSubsequentHandover PrepareSubsequentHandover ::=
191     localValue 69
192
193
194 -- authentication operation codes
195
196 sendAuthenticationInfo SendAuthenticationInfo ::= localValue 56
197
198
199 -- IMEI MANAGEMENT operation codes
200

```

```

201 checkIMEI CheckIMEI ::= localValue 43
202
203
204 -- subscriber management operation codes
205
206 insertSubscriberData InsertSubscriberData ::= localValue 7
207 deleteSubscriberData DeleteSubscriberData ::= localValue 8
208
209
210 -- fault recovery operation codes
211
212 reset Reset ::= localValue 37
213 forwardCheckSS-Indication ForwardCheckSS-Indication ::=
214     localValue 38
215 restoreData RestoreData ::= localValue 57
216
217
218 -- operation and maintenance operation codes
219
220 activateTraceMode ActivateTraceMode ::= localValue 50
221 deactivateTraceMode DeactivateTraceMode ::= localValue 51
222 sendIMSI SendIMSI ::= localValue 58
223
224
225 -- call handling operation codes
226
227 sendRoutingInfo SendRoutingInfo ::= localValue 22
228 provideRoamingNumber ProvideRoamingNumber ::= localValue 4
229 resumeCallHandling ResumeCallHandling ::= localValue 6
230 provideSIWFSNumber ProvideSIWFSNumber ::= localValue 31
231 siWFSSignallingModify SIWFSSignallingModify ::= localValue 32
232 setReportingState SetReportingState ::= localValue 73
233 statusReport StatusReport ::= localValue 74
234 remoteUserFree RemoteUserFree ::= localValue 75
235 istAlert IST-Alert ::= localValue 87
236 istCommand IST-Command ::= localValue 88
237
238
239 -- supplementary service handling operation codes
240
241 registerSS RegisterSS ::= localValue 10
242 eraseSS EraseSS ::= localValue 11
243 activateSS ActivateSS ::= localValue 12
244 deactivateSS DeactivateSS ::= localValue 13
245 interrogateSS InterrogateSS ::= localValue 14
246 processUnstructuredSS-Request ProcessUnstructuredSS-Request ::=
247     localValue 59
248 unstructuredSS-Request UnstructuredSS-Request ::= localValue 60
249 unstructuredSS-Notify UnstructuredSS-Notify ::= localValue 61
250 registerPassword RegisterPassword ::= localValue 17
251 getPassword GetPassword ::= localValue 18
252 registerCC-Entry RegisterCC-Entry ::= localValue 76
253 eraseCC-Entry EraseCC-Entry ::= localValue 77
254
255
256 -- short message service operation codes
257
258 sendRoutingInfoForSM SendRoutingInfoForSM ::= localValue 45
259 mo-forwardSM MO-ForwardSM ::= localValue 46
260 mt-forwardSM MT-ForwardSM ::= localValue 44
261 reportSM-DeliveryStatus ReportSM-DeliveryStatus ::= localValue 47
262 informServiceCentre InformServiceCentre ::= localValue 63
263 alertServiceCentre AlertServiceCentre ::= localValue 64
264 readyForSM ReadyForSM ::= localValue 66
265
266 -- provide subscriber info operation codes
267
268 provideSubscriberInfo ProvideSubscriberInfo ::= localValue 70
269
270 -- any time interrogation operation codes
271
272 anyTimeInterrogation AnyTimeInterrogation ::= localValue 71
273
274 -- any time information handling operation codes
275
276 anyTimeSubscriptionInterrogation AnyTimeSubscriptionInterrogation ::= localValue 62
277 anyTimeModification AnyTimeModification ::= localValue 65

```

```

278
279 -- subscriber data modification notification operation codes
280
281 noteSubscriberDataModified NoteSubscriberDataModified ::= localValue 5
282
283 -- supplementary service invocation notification operation codes
284
285 ss-InvocationNotification SS-InvocationNotification ::= localValue 72
286
287
288 --Group Call operation codes
289
290 prepareGroupCall PrepareGroupCall ::= localValue 39
291 sendGroupCallEndSignal SendGroupCallEndSignal ::= localValue 40
292 processGroupCallSignalling ProcessGroupCallSignalling ::= localValue 41
293 forwardGroupCallSignalling ForwardGroupCallSignalling ::= localValue 42
294
295
296 -- gprs location updating operation codes
297
298 updateGprsLocation UpdateGprsLocation ::= localValue 23
299
300 -- gprs location information retrieval operation codes
301
302 sendRoutingInfoForGprs SendRoutingInfoForGprs ::= localValue 24
303
304 -- failure reporting operation codes
305
306 failureReport FailureReport ::= localValue 25
307
308 -- GPRS notification operation codes
309
310 noteMsPresentForGprs NoteMsPresentForGprs ::= localValue 26
311
312 -- Location service operation codes
313
314 provideSubscriberLocation ProvideSubscriberLocation ::= localValue 83
315 sendRoutingInfoForLCS SendRoutingInfoForLCS ::= localValue 85
316 subscriberLocationReport SubscriberLocationReport ::= localValue 86
317
318
319 -- Mobility Management operation codes
320
321 noteMM-Event NoteMM-Event ::= localValue 89
322
323
324 -- generic error codes
325
326 systemFailure SystemFailure ::= localValue 34
327 dataMissing DataMissing ::= localValue 35
328 unexpectedDataValue UnexpectedDataValue ::= localValue 36
329 facilityNotSupported FacilityNotSupported ::= localValue 21
330 incompatibleTerminal IncompatibleTerminal ::= localValue 28
331 resourceLimitation ResourceLimitation ::= localValue 51
332
333
334 -- identification and numbering error codes
335
336 unknownSubscriber UnknownSubscriber ::= localValue 1
337 numberChanged NumberChanged ::= localValue 44
338 unknownMSC UnknownMSC ::= localValue 3
339 unidentifiedSubscriber UnidentifiedSubscriber ::= localValue 5
340 unknownEquipment UnknownEquipment ::= localValue 7
341
342
343 -- subscription error codes
344
345 roamingNotAllowed RoamingNotAllowed ::= localValue 8
346 illegalSubscriber IllegalSubscriber ::= localValue 9
347 illegalEquipment IllegalEquipment ::= localValue 12
348 bearerServiceNotProvisioned BearerServiceNotProvisioned ::=
349     localValue 10
350 teleserviceNotProvisioned TeleserviceNotProvisioned ::=
351     localValue 11
352
353

```

354 -- handover error codes

355

356 **noHandoverNumberAvailable** NoHandoverNumberAvailable ::=

localValue 25

358 **subsequentHandoverFailure** SubsequentHandoverFailure ::=

localValue 26

360

361 -- operation and maintenance error codes

362

364 **tracingBufferFull** TracingBufferFull ::= localValue 40

365

366

367 -- call handling error codes

368

369 **noRoamingNumberAvailable** NoRoamingNumberAvailable ::= localValue 39

370 **absentSubscriber** AbsentSubscriber ::= localValue 27

371 **busySubscriber** BusySubscriber ::= localValue 45

372 **noSubscriberReply** NoSubscriberReply ::= localValue 46

373 **callBarred** CallBarred ::= localValue 13

374 **forwardingFailed** ForwardingFailed ::= localValue 47

375 **or-NotAllowed** OR-NotAllowed ::= localValue 48

376 **forwardingViolation** ForwardingViolation ::= localValue 14

377 **cug-Reject** CUG-Reject ::= localValue 15

378

379

380 -- any time interrogation error codes

381

382 **ati-NotAllowed** ATI-NotAllowed ::= localValue 49

383

384 -- any time information handling error codes

385 **atsi-NotAllowed** ATSI-NotAllowed ::= localValue 60

386 **atm-NotAllowed** ATM-NotAllowed ::= localValue 61

387 **informationNotAvailable** InformationNotAvailable ::= localValue 62

388

389

390 -- Group Call error codes

391

391 **noGroupCallNumberAvailable** NoGroupCallNumberAvailable ::= localValue 50

392

393

394 -- supplementary service error codes

395

396 **illegalSS-Operation** IllegalSS-Operation ::= localValue 16

397 **ss-ErrorStatus** SS-ErrorStatus ::= localValue 17

398 **ss-NotAvailable** SS-NotAvailable ::= localValue 18

399 **ss-SubscriptionViolation** SS-SubscriptionViolation ::= localValue 19

400 **ss-Incompatibility** SS-Incompatibility ::= localValue 20

401 **unknownAlphabet** UnknownAlphabet ::= localValue 71

402 **ussd-Busy** USSD-Busy ::= localValue 72

403 **pw-RegistrationFailure** PW-RegistrationFailure ::= localValue 37

404 **negativePW-Check** NegativePW-Check ::= localValue 38

405 **numberOfPW-AttemptsViolation** NumberOfPW-AttemptsViolation ::=

localValue 43

407 **shortTermDenial** ShortTermDenial ::= localValue 29

408 **longTermDenial** LongTermDenial ::= localValue 30

409

410

411 -- short message service error codes

412

413 **subscriberBusyForMT-SMS** SubscriberBusyForMT-SMS ::= localValue 31

414 **sm-DeliveryFailure** SM-DeliveryFailure ::= localValue 32

415 **messageWaitingListFull** MessageWaitingListFull ::= localValue 33

416 **absentsubscriberSM** AbsentSubscriberSM ::= localValue 6

417

418 -- location service error codes

419

420 **unauthorizedRequestingNetwork** UnauthorizedRequestingNetwork ::= localValue 52

421 **unauthorizedLCSCClient** UnauthorizedLCSCClient ::= localValue 53

422 **positionMethodFailure** PositionMethodFailure ::= localValue 54

423 **unknownOrUnreachableLCSCClient** UnknownOrUnreachableLCSCClient ::= localValue 58

424

425

426 -- Mobility Management error codes

427 **mm-EventNotSupported** MM-EventNotSupported ::= localValue 59

428

429 -- The following operation codes are reserved for operations
 430 -- existing in previous versions of the protocol

431

432 -- Operation Name	AC used	Oper. Code
433 --		
434 -- sendParameters	map-ac infoRetrieval (14) version1 (1)	localValue 9
435 -- processUnstructuredSS-Data	map-ac networkFunctionalSs (18) version1 (1)	localValue 19
436 -- performHandover	map-ac handoverControl (11) version1 (1)	localValue 28
437 -- performSubsequentHandover	map-ac handoverControl (11) version1 (1)	localValue 30
438 -- noteInternalHandover	map-ac handoverControl (11) version1 (1)	localValue 35
439 -- noteSubscriberPresent	map-ac mwdMngt (24) version1 (1)	localValue 48
440 -- alertServiceCentreWithoutResult	map-ac shortMsgAlert (23) version1 (1)	localValue 49
441 -- traceSubscriberActivity	map-ac handoverControl (11) version1 (1)	localValue 52
442 -- beginSubscriberActivity	map-ac networkFunctionalSs (18) version1 (1)	localValue 54

443

444 -- The following error codes are reserved for errors
 445 -- existing in previous versions of the protocol

446

447 -- Error Name	AC used	Error Code
448 --		
449 -- unknownBaseStation	map-ac handoverControl (11) version1 (1)	localValue 2
450 -- invalidTargetBaseStation	map-ac handoverControl (11) version1 (1)	localValue 23
451 -- noRadioResourceAvailable	map-ac handoverControl (11) version1 (1)	localValue 24

452

453

454 END

17.6 MAP operation and error types

17.6.1 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    version6 (6)}
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   -- gprs location registration operations
21   UpdateGprsLocation,
22
23   -- subscriber information enquiry operations
24   ProvideSubscriberInfo,
25
26   -- any time information enquiry operations
27   AnyTimeInterrogation,
28
29   -- any time information handling operations
30   AnyTimeSubscriptionInterrogation,
31   AnyTimeModification,
32
33   -- subscriber data modification notification operations
34   NoteSubscriberDataModified,
35
36
37   -- handover operations
38   PrepareHandover,
39   SendEndSignal,
40   ProcessAccessSignalling,
41   ForwardAccessSignalling,
42   PrepareSubsequentHandover,
43
44   -- authentication management operations
45   SendAuthenticationInfo,

```



```

46
47 -- IMEI management operations
48 CheckIMEI,
49
50 -- subscriber management operations
51 InsertSubscriberData,
52 DeleteSubscriberData,
53
54 -- fault recovery operations
55 Reset,
56 ForwardCheckSS-Indication,
57 RestoreData,
58
59 -- gprs location information retrieval operations
60 SendRoutingInfoForGprs,
61
62 -- failure reporting operations
63 FailureReport,
64
65 -- gprs notification operations
66 NoteMsPresentForGprs,
67
68 -- Mobility Management operations
69 NoteMM-Event
70
71
72
73
74 ;
75
76 IMPORTS
77 OPERATION
78 FROM TCAPMessages {
79 ccitt recommendation q 773 modules (2) messages (1) version2 (2)}
80
81 SystemFailure,
82 DataMissing,
83 UnexpectedDataValue,
84 UnknownSubscriber,
85 UnknownMSC,
86 UnidentifiedSubscriber,
87 UnknownEquipment,
88 RoamingNotAllowed,
89 ATI-NotAllowed,
90 NoHandoverNumberAvailable,
91 SubsequentHandoverFailure,
92 AbsentSubscriber,
93 MM-EventNotSupported,
94 ATSI-NotAllowed,
95 ATM-NotAllowed,
96 BearerServiceNotProvisioned,
97 TeleserviceNotProvisioned,
98 CallBarred,
99 IllegalSS-Operation,
100 SS-ErrorStatus,
101 SS-NotAvailable,
102 SS-Incompatibility,
103 SS-SubscriptionViolation,
104 InformationNotAvailable
105
106
107 FROM MAP-Errors {
108 ccitt identified-organization (4) etsi (0) mobileDomain (0)
109 gsm-Network (1) modules (3) map-Errors (10) version6 (6)}
110
111 UpdateLocationArg,
112 UpdateLocationRes,
113 CancelLocationArg,
114 CancelLocationRes,
115 PurgeMS-Arg,
116 PurgeMS-Res,
117 SendIdentificationArg,
118 SendIdentificationRes,
119 UpdateGprsLocationArg,
120 UpdateGprsLocationRes,
121 PrepareHO-Arg,
122 PrepareHO-Res,
123 PrepareSubsequentHO-Arg,
124 SendAuthenticationInfoArg,

```

```

125     SendAuthenticationInfoRes,
126     EquipmentStatus,
127     InsertSubscriberDataArg,
128     InsertSubscriberDataRes,
129     DeleteSubscriberDataArg,
130     DeleteSubscriberDataRes,
131     ResetArg,
132     RestoreDataArg,
133     RestoreDataRes,
134     ProvideSubscriberInfoArg,
135     ProvideSubscriberInfoRes,
136     AnyTimeSubscriptionInterrogationArg,
137     AnyTimeSubscriptionInterrogationRes,
138     AnyTimeModificationArg,
139     AnyTimeModificationRes,
140     NoteSubscriberDataModifiedArg,
141     NoteSubscriberDataModifiedRes,
142     AnyTimeInterrogationArg,
143     AnyTimeInterrogationRes,
144     SendRoutingInfoForGprsArg,
145     SendRoutingInfoForGprsRes,
146     FailureReportArg,
147     FailureReportRes,
148     NoteMsPresentForGprsArg,
149     NoteMsPresentForGprsRes,
150     NoteMM-EventArg,
151     NoteMM-EventRes
152
153
154 FROM MAP-MS-DataTypes {
155     ccitt identified-organization (4) etsi (0) mobileDomain (0)
156     gsm-Network (1) modules (3) map-MS-DataTypes (11) version6 (6)}
157
158     ExternalSignalInfo,
159     IMEI
160 FROM MAP-CommonDataTypes {
161     ccitt identified-organization (4) etsi (0) mobileDomain (0)
162     gsm-Network (1) modules (3) map-CommonDataTypes (18) version6 (6)}
163 ;
164
165
166 -- location registration operations
167

```

168	UpdateLocation ::= OPERATION	--Timer m
169	ARGUMENT	
170	updateLocationArg	UpdateLocationArg
171	RESULT	
172	updateLocationRes	UpdateLocationRes
173	ERRORS {	
174	SystemFailure,	
175	DataMissing,	
176	UnexpectedDataValue,	
177	UnknownSubscriber,	
178	RoamingNotAllowed}	

180	CancelLocation ::= OPERATION	--Timer m
181	ARGUMENT	
182	cancelLocationArg	CancelLocationArg
183	RESULT	
184	cancelLocationRes	CancelLocationRes
185	-- optional	
186	ERRORS {	
187	DataMissing,	
188	UnexpectedDataValue}	

190	PurgeMS ::= OPERATION	--Timer m
191	ARGUMENT	
192	purgeMS-Arg	PurgeMS-Arg
193	RESULT	
194	purgeMS-Res	PurgeMS-Res
195	-- optional	
196	ERRORS{	
197	DataMissing,	
198	UnexpectedDataValue,	
199	UnknownSubscriber}	

200

```

201 SendIdentification ::= OPERATION --Timer s
202     ARGUMENT
203         sendIdentificationArg          SendIdentificationArg
204     RESULT
205         sendIdentificationRes          SendIdentificationRes
206     ERRORS {
207         DataMissing,
208         UnidentifiedSubscriber}
209
210 -- gprs location registration operations
211
212 UpdateGprsLocation ::= OPERATION --Timer m
213     ARGUMENT
214         updateGprsLocationArg          UpdateGprsLocationArg
215     RESULT
216         updateGprsLocationRes          UpdateGprsLocationRes
217     ERRORS {
218         SystemFailure,
219         UnexpectedDataValue,
220         UnknownSubscriber,
221         RoamingNotAllowed}
222
223 -- subscriber information enquiry operations
224
225 ProvideSubscriberInfo ::= OPERATION --Timer m
226     ARGUMENT
227         provideSubscriberInfoArg        ProvideSubscriberInfoArg
228     RESULT
229         provideSubscriberInfoRes        ProvideSubscriberInfoRes
230     ERRORS {
231         DataMissing,
232         UnexpectedDataValue}
233
234 -- any time information enquiry operations
235
236 AnyTimeInterrogation ::= OPERATION --Timer m
237     ARGUMENT
238         anyTimeInterrogationArg        AnyTimeInterrogationArg
239     RESULT
240         anyTimeInterrogationRes        AnyTimeInterrogationRes
241     ERRORS {
242         SystemFailure,
243         ATI-NotAllowed,
244         DataMissing,
245         UnexpectedDataValue,
246         UnknownSubscriber}
247
248 -- any time information handling operations
249
250 AnyTimeSubscriptionInterrogation ::= OPERATION --Timer m
251     ARGUMENT
252         anyTimeSubscriptionInterrogationArg AnyTimeSubscriptionInterrogationArg
253     RESULT
254         anyTimeSubscriptionInterrogationRes AnyTimeSubscriptionInterrogationRes
255     ERRORS {
256         ATSI-NotAllowed,
257         DataMissing,
258         UnexpectedDataValue,
259         UnknownSubscriber,
260         BearerServiceNotProvisioned,
261         TeleserviceNotProvisioned,
262         CallBarred,
263         IllegalSS-Operation,
264         SS-NotAvailable,
265         InformationNotAvailable}
266

```

```

267 AnyTimeModification ::= OPERATION --Timer m
268     ARGUMENT
269         anyTimeModificationArg         AnyTimeModificationArg
270     RESULT
271         anyTimeModificationRes         AnyTimeModificationRes
272     ERRORS {
273         ATM-NotAllowed,
274         DataMissing,
275         UnexpectedDataValue,
276         UnknownSubscriber,
277         BearerServiceNotProvisioned,
278         TeleserviceNotProvisioned,
279         CallBarred,
280         IllegalSS-Operation,
281         SS-SubscriptionViolation,
282         SS-ErrorStatus,
283         SS-Incompatibility,
284         InformationNotAvailable}
285
286 -- subscriber data modification notification operations
287
288 NoteSubscriberDataModified ::= OPERATION --Timer m
289     ARGUMENT
290         noteSubscriberDataModifiedArg   NoteSubscriberDataModifiedArg
291     RESULT
292         noteSubscriberDataModifiedRes   NoteSubscriberDataModifiedRes
293         -- optional
294     ERRORS {
295         UnexpectedDataValue,
296         UnknownSubscriber}
297
298 -- handover operations
299
300 PrepareHandover ::= OPERATION --Timer m
301     ARGUMENT
302         prepareHO-Arg                   PrepareHO-Arg
303     RESULT
304         prepareHO-Res                   PrepareHO-Res
305     ERRORS {
306         SystemFailure,
307         DataMissing,
308         UnexpectedDataValue,
309         NoHandoverNumberAvailable}
310
311 SendEndSignal ::= OPERATION --Timer l
312     ARGUMENT
313         bss-APDU                         ExternalSignalInfo
314     RESULT
315
316 ProcessAccessSignalling ::= OPERATION --Timer s
317     ARGUMENT
318         bss-APDU                         ExternalSignalInfo
319
320 ForwardAccessSignalling ::= OPERATION --Timer s
321     ARGUMENT
322         bss-APDU                         ExternalSignalInfo
323
324 PrepareSubsequentHandover ::= OPERATION --Timer m
325     ARGUMENT
326         prepareSubsequentHO-Arg         PrepareSubsequentHO-Arg
327     RESULT
328         bss-APDU                         ExternalSignalInfo
329     ERRORS {
330         UnexpectedDataValue,
331         DataMissing,
332         UnknownMSC,
333         SubsequentHandoverFailure}
334
335 -- authentication management operations
336
337

```

```

338 SendAuthenticationInfo ::= OPERATION                                --Timer m
339     ARGUMENT
340         sendAuthenticationInfoArg      SendAuthenticationInfoArg
341     RESULT
342         sendAuthenticationInfoRes      SendAuthenticationInfoRes
343         -- optional
344     ERRORS {
345         SystemFailure,
346         DataMissing,
347         UnexpectedDataValue,
348         UnknownSubscriber}
349
350 -- IMEI management operations
351
352 CheckIMEI ::= OPERATION                                           --Timer m
353     ARGUMENT
354         imei                            IMEI
355     RESULT
356         equipmentStatus                 EquipmentStatus
357     ERRORS {
358         SystemFailure,
359         DataMissing,
360         UnknownEquipment}
361
362 -- subscriber management operations
363
364 InsertSubscriberData ::= OPERATION                                --Timer m
365     ARGUMENT
366         insertSubscriberDataArg        InsertSubscriberDataArg
367     RESULT
368         insertSubscriberDataRes        InsertSubscriberDataRes
369         -- optional
370     ERRORS {
371         DataMissing,
372         UnexpectedDataValue,
373         UnidentifiedSubscriber}
374
375 DeleteSubscriberData ::= OPERATION                                --Timer m
376     ARGUMENT
377         deleteSubscriberDataArg        DeleteSubscriberDataArg
378     RESULT
379         deleteSubscriberDataRes        DeleteSubscriberDataRes
380         -- optional
381     ERRORS {
382         DataMissing,
383         UnexpectedDataValue,
384         UnidentifiedSubscriber}
385
386 -- fault recovery operations
387
388 Reset ::= OPERATION                                               --Timer m
389     ARGUMENT
390         resetArg                        ResetArg
391
392 ForwardCheckSS-Indication ::= OPERATION                          --Timer s
393
394 RestoreData ::= OPERATION                                         --Timer m
395     ARGUMENT
396         restoreDataArg                 RestoreDataArg
397     RESULT
398         restoreDataRes                 RestoreDataRes
399     ERRORS {
400         SystemFailure,
401         DataMissing,
402         UnexpectedDataValue,
403         UnknownSubscriber}
404
405 -- gprs location information retrieval operations
406

```

```

407 SendRoutingInfoForGprs ::= OPERATION --Timer m
408     ARGUMENT
409         sendRoutingInfoForGprsArg      SendRoutingInfoForGprsArg
410     RESULT
411         sendRoutingInfoForGprsRes      SendRoutingInfoForGprsRes
412     ERRORS {
413         AbsentSubscriber,
414         SystemFailure,
415         DataMissing,
416         UnexpectedDataValue,
417         UnknownSubscriber}
418
419 -- failure reporting operations
420
421 FailureReport ::= OPERATION --Timer m
422     ARGUMENT
423         failureReportArg                FailureReportArg
424     RESULT
425         failureReportRes                FailureReportRes
426         -- optional
427     ERRORS {
428         SystemFailure,
429         DataMissing,
430         UnexpectedDataValue,
431         UnknownSubscriber}
432
433 -- gprs notification operations
434
435 NoteMsPresentForGprs ::= OPERATION --Timer m
436     ARGUMENT
437         noteMsPresentForGprsArg         NoteMsPresentForGprsArg
438     RESULT
439         noteMsPresentForGprsRes         NoteMsPresentForGprsRes
440         -- optional
441     ERRORS {
442         SystemFailure,
443         DataMissing,
444         UnexpectedDataValue,
445         UnknownSubscriber}
446
447
448 NoteMM-Event ::= OPERATION --Timer m
449     ARGUMENT
450         noteMM-EventArg                 NoteMM-EventArg
451     RESULT
452         noteMM-EventRes                 NoteMM-EventRes
453     ERRORS {
454         DataMissing,
455         UnexpectedDataValue,
456         UnknownSubscriber,
457         MM-EventNotSupported}
458
459 END

```

17.6.2 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      version6 (6)}
5
6  DEFINITIONS
7
8  ::=
9
10 BEGIN
11
12 EXPORTS
13     ActivateTraceMode,
14     DeactivateTraceMode,
15     SendIMSI
16 ;
17
18 IMPORTS
19     OPERATION
20 FROM TCAPMessages {
21     ccitt recommendation q 773 modules (2) messages (1) version2 (2)}
22

```

```

23     SystemFailure,
24     DataMissing,
25     UnexpectedDataValue,
26     FacilityNotSupported,
27     UnknownSubscriber,
28     UnidentifiedSubscriber,
29     TracingBufferFull
30 FROM MAP-Errors {
31     ccitt identified-organization (4) etsi (0) mobileDomain (0)
32     gsm-Network (1) modules (3) map-Errors (10) version6 (6)}
33
34     ActivateTraceModeArg,
35     ActivateTraceModeRes,
36     DeactivateTraceModeArg,
37     DeactivateTraceModeRes
38 FROM MAP-OM-DataTypes {
39     ccitt identified-organization (4) etsi (0) mobileDomain (0)
40     gsm-Network (1) modules (3) map-OM-DataTypes (12) version6 (6)}
41
42     ISDN-AddressString,
43     IMSI
44 FROM MAP-CommonDataTypes {
45     ccitt identified-organization (4) etsi (0) mobileDomain (0)
46     gsm-Network (1) modules (3) map-CommonDataTypes (18) version6 (6)}
47 ;
48
49

```

```

50 ActivateTraceMode ::= OPERATION --Timer m
51     ARGUMENT
52         activateTraceModeArg          ActivateTraceModeArg
53     RESULT
54         activateTraceModeRes          ActivateTraceModeRes
55         -- optional
56     ERRORS {
57         SystemFailure,
58         DataMissing,
59         UnexpectedDataValue,
60         FacilityNotSupported,
61         UnidentifiedSubscriber,
62         TracingBufferFull}

```

```

64 DeactivateTraceMode ::= OPERATION --Timer m
65     ARGUMENT
66         deactivateTraceModeArg        DeactivateTraceModeArg
67     RESULT
68         deactivateTraceModeRes        DeactivateTraceModeRes
69         -- optional
70     ERRORS {
71         SystemFailure,
72         DataMissing,
73         UnexpectedDataValue,
74         FacilityNotSupported,
75         UnidentifiedSubscriber}

```

```

77 SendIMSI ::= OPERATION --Timer m
78     ARGUMENT
79         msisdn                        ISDN-AddressString
80     RESULT
81         imsi                          IMSI
82     ERRORS {
83         DataMissing,
84         UnexpectedDataValue,
85         UnknownSubscriber}

```

```

86
87 END

```

17.6.3 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     version6 (6)}
5
6 DEFINITIONS
7
8 ::=
9
10 BEGIN

```

```

11
12 EXPORTS
13     SendRoutingInfo,
14     ProvideRoamingNumber,
15     ResumeCallHandling,
16     ProvideSIWFSNumber,
17     SIWFSSignallingModify,
18     SetReportingState,
19     StatusReport,
20     RemoteUserFree,
21     IST-Alert,
22     IST-Command
23 ;
24
25 IMPORTS
26     OPERATION
27 FROM TCAPMessages {
28     ccitt recommendation q 773 modules (2) messages (1) version2 (2)}
29
30     SystemFailure,
31     DataMissing,
32     UnexpectedDataValue,
33     FacilityNotSupported,
34     OR-NotAllowed,
35     UnknownSubscriber,
36     NumberChanged,
37     BearerServiceNotProvisioned,
38     TeleserviceNotProvisioned,
39     NoRoamingNumberAvailable,
40     AbsentSubscriber,
41     BusySubscriber,
42     NoSubscriberReply,
43     CallBarred,
44     ForwardingViolation,
45     ForwardingFailed,
46     CUG-Reject,
47     ResourceLimitation,
48     IncompatibleTerminal,
49     UnidentifiedSubscriber
50
51 FROM MAP-Errors {
52     ccitt identified-organization (4) etsi (0) mobileDomain (0)
53     gsm-Network (1) modules (3) map-Errors (10) version6 (6)}
54     SendRoutingInfoArg,
55     SendRoutingInfoRes,
56     ProvideRoamingNumberArg,
57     ProvideRoamingNumberRes,
58     ResumeCallHandlingArg,
59     ResumeCallHandlingRes,
60     ProvideSIWFSNumberArg,
61     ProvideSIWFSNumberRes,
62     SIWFSSignallingModifyArg,
63     SIWFSSignallingModifyRes,
64     SetReportingStateArg,
65     SetReportingStateRes,
66     StatusReportArg,
67     StatusReportRes,
68     RemoteUserFreeArg,
69     RemoteUserFreeRes,
70     IST-AlertArg,
71     IST-AlertRes,
72     IST-CommandArg,
73     IST-CommandRes
74 FROM MAP-CH-DataTypes {
75     ccitt identified-organization (4) etsi (0) mobileDomain (0)
76     gsm-Network (1) modules (3) map-CH-DataTypes (13) version6 (6)}
77
78 ;
79

```



```

80 SendRoutingInfo ::= OPERATION --Timer m
81 -- The timer is set to the upper limit of the range if the GMSC supports pre-paging.
82 ARGUMENT
83     sendRoutingInfoArg          SendRoutingInfoArg
84 RESULT
85     sendRoutingInfoRes          SendRoutingInfoRes
86 ERRORS {
87     SystemFailure,
88     DataMissing,
89     UnexpectedDataValue,
90     FacilityNotSupported,
91     OR-NotAllowed,
92     UnknownSubscriber,
93     NumberChanged,
94     BearerServiceNotProvisioned,
95     TeleserviceNotProvisioned,
96     AbsentSubscriber,
97     BusySubscriber,
98     NoSubscriberReply,
99     CallBarred,
100    CUG-Reject,
101    ForwardingViolation}
102
103 ProvideRoamingNumber ::= OPERATION --Timer m
104 -- The timer is set to the upper limit of the range if the HLR supports pre-paging.
105 ARGUMENT
106     provideRoamingNumberArg      ProvideRoamingNumberArg
107 RESULT
108     provideRoamingNumberRes      ProvideRoamingNumberRes
109 ERRORS {
110     SystemFailure,
111     DataMissing,
112     UnexpectedDataValue,
113     FacilityNotSupported,
114     OR-NotAllowed,
115     AbsentSubscriber,
116     NoRoamingNumberAvailable}
117
118 ResumeCallHandling ::= OPERATION --Timer m
119 ARGUMENT
120     resumeCallHandlingArg        ResumeCallHandlingArg
121 RESULT
122     resumeCallHandlingRes        ResumeCallHandlingRes
123     -- optional
124 ERRORS {
125     ForwardingFailed,
126     OR-NotAllowed,
127     UnexpectedDataValue,
128     DataMissing }
129
130 ProvideSIWFSNumber ::= OPERATION --Timer m
131 ARGUMENT
132     provideSIWFSNumberArg        ProvideSIWFSNumberArg
133 RESULT
134     provideSIWFSNumberRes        ProvideSIWFSNumberRes
135 ERRORS {
136     ResourceLimitation,
137     DataMissing,
138     UnexpectedDataValue,
139     SystemFailure}
140
141 SIWFSSignallingModify ::= OPERATION --Timer m
142 ARGUMENT
143     siWFSSignallingModifyArg      SIWFSSignallingModifyArg
144 RESULT
145     siWFSSignallingModifyRes      SIWFSSignallingModifyRes
146     -- optional
147 ERRORS {
148     ResourceLimitation,
149     DataMissing,
150     UnexpectedDataValue,
151     SystemFailure}
152

```

```

153 SetReportingState ::= OPERATION --Timer m
154     ARGUMENT
155         setReportingStateArg          SetReportingStateArg
156     RESULT
157         setReportingStateRes          SetReportingStateRes
158         -- optional
159     ERRORS {
160         SystemFailure,
161         UnidentifiedSubscriber,
162         UnexpectedDataValue,
163         DataMissing,
164         ResourceLimitation,
165         FacilityNotSupported}
166
167 StatusReport ::= OPERATION --Timer m
168     ARGUMENT
169         statusReportArg              StatusReportArg
170     RESULT
171         statusReportRes              StatusReportRes
172         -- optional
173     ERRORS {
174         UnknownSubscriber,
175         SystemFailure,
176         UnexpectedDataValue,
177         DataMissing}
178
179 RemoteUserFree ::= OPERATION --Timer m1
180     ARGUMENT
181         remoteUserFreeArg            RemoteUserFreeArg
182     RESULT
183         remoteUserFreeRes            RemoteUserFreeRes
184     ERRORS {
185         UnexpectedDataValue,
186         DataMissing,
187         IncompatibleTerminal,
188         AbsentSubscriber,
189         SystemFailure,
190         BusySubscriber}
191
192 IST-Alert ::= OPERATION --Timer m
193     ARGUMENT
194         istAlertArg                  IST-AlertArg
195     RESULT
196         istAlertRes                  IST-AlertRes
197         -- optional
198     ERRORS {
199         UnexpectedDataValue,
200         ResourceLimitation,
201         UnknownSubscriber,
202         SystemFailure,
203         FacilityNotSupported}
204
205 IST-Command ::= OPERATION --Timer m
206     ARGUMENT
207         istCommandArg                IST-CommandArg
208     RESULT
209         istCommandRes                IST-CommandRes
210         -- optional
211     ERRORS {
212         UnexpectedDataValue,
213         ResourceLimitation,
214         UnknownSubscriber,
215         SystemFailure,
216         FacilityNotSupported}
217
218 END

```

17.6.4 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     version6 (6)}
5
6 DEFINITIONS
7
8 ::=
9

```

```

10 BEGIN
11
12 EXPORTS
13     RegisterSS,
14     EraseSS,
15     ActivateSS,
16     DeactivateSS,
17     InterrogateSS,
18     ProcessUnstructuredSS-Request,
19     UnstructuredSS-Request,
20     UnstructuredSS-Notify,
21     RegisterPassword,
22     GetPassword,
23     SS-InvocationNotification,
24     RegisterCC-Entry,
25     EraseCC-Entry
26 ;
27
28 IMPORTS
29     OPERATION
30 FROM TCAPMessages {
31     ccitt recommendation q 773 modules (2) messages (1) version2 (2)}
32
33     SystemFailure,
34     DataMissing,
35     UnexpectedDataValue,
36     UnknownSubscriber,
37     BearerServiceNotProvisioned,
38     TeleserviceNotProvisioned,
39     CallBarred,
40     IllegalSS-Operation,
41     SS-ErrorStatus,
42     SS-NotAvailable,
43     SS-SubscriptionViolation,
44     SS-Incompatibility,
45     PW-RegistrationFailure,
46     NegativePW-Check,
47     NumberOfPW-AttemptsViolation,
48     UnknownAlphabet,
49     USSD-Busy,
50     AbsentSubscriber,
51     IllegalSubscriber,
52     IllegalEquipment,
53     ShortTermDenial,
54     LongTermDenial,
55     FacilityNotSupported
56 FROM MAP-Errors {
57     ccitt identified-organization (4) etsi (0) mobileDomain (0)
58     gsm-Network (1) modules (3) map-Errors (10) version6 (6)}
59
60     RegisterSS-Arg,
61     SS-Info,
62     SS-ForBS-Code,
63     InterrogateSS-Res,
64     USSD-Arg,
65     USSD-Res,
66     Password,
67     GuidanceInfo,
68     SS-InvocationNotificationArg,
69     SS-InvocationNotificationRes,
70     RegisterCC-EntryArg,
71     RegisterCC-EntryRes,
72     EraseCC-EntryArg,
73     EraseCC-EntryRes
74 FROM MAP-SS-DataTypes {
75     ccitt identified-organization (4) etsi (0) mobileDomain (0)
76     gsm-Network (1) modules (3) map-SS-DataTypes (14) version6 (6)}
77
78     SS-Code
79 FROM MAP-SS-Code {
80     ccitt identified-organization (4) etsi (0) mobileDomain (0)
81     gsm-Network (1) modules (3) map-SS-Code (15) version6 (6)}
82 ;
83
84
85 -- supplementary service handling operations
86

```

```

87 RegisterSS ::= OPERATION --Timer m
88 ARGUMENT
89     registerSS-Arg          RegisterSS-Arg
90 RESULT
91     ss-Info                SS-Info
92     -- optional
93 ERRORS {
94     SystemFailure,
95     DataMissing,
96     UnexpectedDataValue,
97     BearerServiceNotProvisioned,
98     TeleserviceNotProvisioned,
99     CallBarred,
100    IllegalSS-Operation,
101    SS-ErrorStatus,
102    SS-Incompatibility}

```

```

104 EraseSS ::= OPERATION --Timer m
105 ARGUMENT
106     ss-ForBS              SS-ForBS-Code
107 RESULT
108     ss-Info              SS-Info
109     -- optional
110 ERRORS {
111     SystemFailure,
112     DataMissing,
113     UnexpectedDataValue,
114     BearerServiceNotProvisioned,
115     TeleserviceNotProvisioned,
116     CallBarred,
117     IllegalSS-Operation,
118     SS-ErrorStatus
119     }

```

```

121 ActivateSS ::= OPERATION --Timer m
122 ARGUMENT
123     ss-ForBS              SS-ForBS-Code
124 RESULT
125     ss-Info              SS-Info
126     -- optional
127 ERRORS {
128     SystemFailure,
129     DataMissing,
130     UnexpectedDataValue,
131     BearerServiceNotProvisioned,
132     TeleserviceNotProvisioned,
133     CallBarred,
134     IllegalSS-Operation,
135     SS-ErrorStatus,
136     SS-SubscriptionViolation,
137     SS-Incompatibility,
138     NegativePW-Check,
139     NumberOfPW-AttemptsViolation}

```

```

141 DeactivateSS ::= OPERATION --Timer m
142 ARGUMENT
143     ss-ForBS              SS-ForBS-Code
144 RESULT
145     ss-Info              SS-Info
146     -- optional
147 ERRORS {
148     SystemFailure,
149     DataMissing,
150     UnexpectedDataValue,
151     BearerServiceNotProvisioned,
152     TeleserviceNotProvisioned,
153     CallBarred,
154     IllegalSS-Operation,
155     SS-ErrorStatus,
156     SS-SubscriptionViolation,
157     NegativePW-Check,
158     NumberOfPW-AttemptsViolation}

```

159

```

160 InterrogateSS ::= OPERATION --Timer m
161     ARGUMENT
162         ss-ForBS                SS-ForBS-Code
163     RESULT
164         interrogateSS-Res        InterrogateSS-Res
165     ERRORS {
166         SystemFailure,
167         DataMissing,
168         UnexpectedDataValue,
169         BearerServiceNotProvisioned,
170         TeleserviceNotProvisioned,
171         CallBarred,
172         IllegalSS-Operation,
173         SS-NotAvailable}
174

```

```

175 ProcessUnstructuredSS-Request ::= OPERATION --Timer 10 minutes
176     ARGUMENT
177         ussd-Arg                USSD-Arg
178     RESULT
179         ussd-Res                USSD-Res
180     ERRORS {
181         SystemFailure,
182         DataMissing,
183         UnexpectedDataValue,
184         UnknownAlphabet,
185         CallBarred}
186

```

```

187 UnstructuredSS-Request ::= OPERATION --Timer ml
188     ARGUMENT
189         ussd-Arg                USSD-Arg
190     RESULT
191         ussd-Res                USSD-Res
192         -- optional
193     ERRORS {
194         SystemFailure,
195         DataMissing,
196         UnexpectedDataValue,
197         AbsentSubscriber,
198         IllegalSubscriber,
199         IllegalEquipment,
200         UnknownAlphabet,
201         USSD-Busy}
202

```

```

203 UnstructuredSS-Notify ::= OPERATION --Timer ml
204     ARGUMENT
205         ussd-Arg                USSD-Arg
206     RESULT
207     ERRORS {
208         SystemFailure,
209         DataMissing,
210         UnexpectedDataValue,
211         AbsentSubscriber,
212         IllegalSubscriber,
213         IllegalEquipment,
214         UnknownAlphabet,
215         USSD-Busy}
216

```

```

217 RegisterPassword ::= OPERATION --Timer ml
218     ARGUMENT
219         ss-Code                SS-Code
220     RESULT
221         newPassword            Password
222     ERRORS {
223         SystemFailure,
224         DataMissing,
225         UnexpectedDataValue,
226         CallBarred,
227         SS-SubscriptionViolation,
228         PW-RegistrationFailure,
229         NegativePW-Check,
230         NumberOfPW-AttemptsViolation}
231     LINKED {
232         GetPassword}
233

```

```

234 GetPassword ::= OPERATION --Timer m
235     ARGUMENT
236         guidanceInfo           GuidanceInfo
237     RESULT
238         currentPassword        Password
239
240 SS-InvocationNotification ::= OPERATION --Timer m
241     ARGUMENT
242         ss-InvocationNotificationArg  SS-InvocationNotificationArg
243     RESULT
244         ss-InvocationNotificationRes  SS-InvocationNotificationRes
245         -- optional
246     ERRORS {
247         DataMissing,
248         UnexpectedDataValue,
249         UnknownSubscriber}
250
251 RegisterCC-Entry ::= OPERATION --Timer m
252     ARGUMENT
253         registerCC-EntryArg          RegisterCC-EntryArg
254     RESULT
255         registerCC-EntryRes          RegisterCC-EntryRes
256     ERRORS {
257         SystemFailure,
258         DataMissing,
259         UnexpectedDataValue,
260         CallBarred,
261         IllegalSS-Operation,
262         SS-ErrorStatus,
263         SS-Incompatibility,
264         ShortTermDenial,
265         LongTermDenial,
266         FacilityNotSupported}
267
268 EraseCC-Entry ::= OPERATION --Timer m
269     ARGUMENT
270         eraseCC-EntryArg              EraseCC-EntryArg
271     RESULT
272         eraseCC-EntryRes              EraseCC-EntryRes
273     ERRORS {
274         SystemFailure,
275         DataMissing,
276         UnexpectedDataValue,
277         CallBarred,
278         IllegalSS-Operation,
279         SS-ErrorStatus}
280
281 END

```

17.6.5 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      version6 (6)}
5
6  DEFINITIONS
7
8  ::=
9
10 BEGIN
11
12 EXPORTS
13     SendRoutingInfoForSM,
14     MO-ForwardSM,
15     MT-ForwardSM,
16     ReportSM-DeliveryStatus,
17     AlertServiceCentre,
18     InformServiceCentre,
19     ReadyForSM
20 ;
21
22 IMPORTS
23     OPERATION
24 FROM TCAPMessages {
25     ccitt recommendation q 773 modules (2) messages (1) version2 (2)}
26
27     SystemFailure,

```

```

28 DataMissing,
29 UnexpectedDataValue,
30 FacilityNotSupported,
31 UnknownSubscriber,
32 UnidentifiedSubscriber,
33 IllegalSubscriber,
34 IllegalEquipment,
35 TeleserviceNotProvisioned,
36 CallBarred,
37 SubscriberBusyForMT-SMS,
38 SM-DeliveryFailure,
39 MessageWaitingListFull,
40 AbsentSubscriberSM
41 FROM MAP-Errors {
42 ccitt identified-organization (4) etsi (0) mobileDomain (0)
43 gsm-Network (1) modules (3) map-Errors (10) version6 (6)}
44
45 RoutingInfoForSM-Arg,
46 RoutingInfoForSM-Res,
47 MO-ForwardSM-Arg,
48 MO-ForwardSM-Res,
49 MT-ForwardSM-Arg,
50 MT-ForwardSM-Res,
51 ReportSM-DeliveryStatusArg,
52 ReportSM-DeliveryStatusRes,
53 AlertServiceCentreArg,
54 InformServiceCentreArg,
55 ReadyForSM-Arg,
56 ReadyForSM-Res
57 FROM MAP-SM-DataTypes {
58 ccitt identified-organization (4) etsi (0) mobileDomain (0)
59 gsm-Network (1) modules (3) map-SM-DataTypes (16) version6 (6)}
60
61
62
63 ;
64
65

```

66	SendRoutingInfoForSM ::= OPERATION	--Timer m
67	ARGUMENT	
68	routingInfoForSM-Arg	RoutingInfoForSM-Arg
69	RESULT	
70	routingInfoForSM-Res	RoutingInfoForSM-Res
71	ERRORS {	
72	SystemFailure,	
73	DataMissing,	
74	UnexpectedDataValue,	
75	FacilityNotSupported,	
76	UnknownSubscriber,	
77	TeleserviceNotProvisioned,	
78	CallBarred,	
79	AbsentSubscriberSM}	

81	MO-ForwardSM ::= OPERATION	--Timer ml
82	ARGUMENT	
83	mo-forwardSM-Arg	MO-ForwardSM-Arg
84	RESULT	
85	mo-forwardSM-Res	MO-ForwardSM-Res
86	-- optional	
87	ERRORS {	
88	SystemFailure,	
89	UnexpectedDataValue,	
90	FacilityNotSupported,	
91	SM-DeliveryFailure}	

92

```

93 MT-ForwardSM ::= OPERATION --Timer ml
94 ARGUMENT
95     mt-ForwardSM-Arg          MT-ForwardSM-Arg
96 RESULT
97     mt-forwardSM-Res          MT-ForwardSM-Res
98     -- optional
99 ERRORS {
100     SystemFailure,
101     DataMissing,
102     UnexpectedDataValue,
103     FacilityNotSupported,
104     UnidentifiedSubscriber,
105     IllegalSubscriber,
106     IllegalEquipment,
107     SubscriberBusyForMT-SMS,
108     SM-DeliveryFailure,
109     AbsentSubscriberSM}
110
111 ReportSM-DeliveryStatus ::= OPERATION --Timer s
112 ARGUMENT
113     reportSM-DeliveryStatusArg ReportSM-DeliveryStatusArg
114 RESULT
115     reportSM-DeliveryStatusRes ReportSM-DeliveryStatusRes
116     -- optional
117 ERRORS {
118     DataMissing,
119     UnexpectedDataValue,
120     UnknownSubscriber,
121     MessageWaitingListFull}
122
123 AlertServiceCentre ::= OPERATION --Timer s
124 ARGUMENT
125     alertServiceCentreArg      AlertServiceCentreArg
126 RESULT
127 ERRORS {
128     SystemFailure,
129     DataMissing,
130     UnexpectedDataValue}
131
132 InformServiceCentre ::= OPERATION --Timer s
133 ARGUMENT
134     informServiceCentreArg     InformServiceCentreArg
135
136 ReadyForSM ::= OPERATION --Timer m
137 ARGUMENT
138     readyForSM-Arg            ReadyForSM-Arg
139 RESULT
140     readyForSM-Res            ReadyForSM-Res
141     -- optional
142 ERRORS {
143     DataMissing,
144     UnexpectedDataValue,
145     FacilityNotSupported,
146     UnknownSubscriber}
147
148 END

```

17.6.6 Errors

```

1 MAP-Errors {
2     ccitt identified-organization (4) etsi (0) mobileDomain (0)
3     gsm-Network (1) modules (3) map-Errors (10) version6 (6)}
4
5 DEFINITIONS
6
7 ::=
8
9 BEGIN
10
11 EXPORTS
12
13     -- generic errors
14     SystemFailure,
15     DataMissing,
16     UnexpectedDataValue,
17     FacilityNotSupported,
18     IncompatibleTerminal,
19     ResourceLimitation,

```



```
20
21 -- identification and numbering errors
22 UnknownSubscriber,
23 NumberChanged,
24 UnknownMSC,
25 UnidentifiedSubscriber,
26 UnknownEquipment,
27
28 -- subscription errors
29 RoamingNotAllowed,
30 IllegalSubscriber,
31 IllegalEquipment,
32 BearerServiceNotProvisioned,
33 TeleserviceNotProvisioned,
34
35 -- handover errors
36 NoHandoverNumberAvailable,
37 SubsequentHandoverFailure,
38
39 -- operation and maintenance errors
40 TracingBufferFull,
41
42 -- call handling errors
43 OR-NotAllowed,
44 NoRoamingNumberAvailable,
45 BusySubscriber,
46 NoSubscriberReply,
47 AbsentSubscriber,
48 CallBarred,
49 ForwardingViolation,
50 ForwardingFailed,
51 CUG-Reject,
52
53 -- any time interrogation errors
54 ATI-NotAllowed,
55
56 -- any time information handling errors
57 ATSI-NotAllowed,
58 ATM-NotAllowed,
59 InformationNotAvailable,
60
61 -- supplementary service errors
62 IllegalSS-Operation,
63 SS-ErrorStatus,
64 SS-NotAvailable,
65 SS-SubscriptionViolation,
66 SS-Incompatibility,
67 UnknownAlphabet,
68 USSD-Busy,
69 PW-RegistrationFailure,
70 NegativePW-Check,
71 NumberOfPW-AttemptsViolation,
72 ShortTermDenial,
73 LongTermDenial,
74
75 -- short message service errors
76 SubscriberBusyForMT-SMS,
77 SM-DeliveryFailure,
78 MessageWaitingListFull,
79 AbsentSubscriberSM,
80
81 -- Group Call errors
82 NoGroupCallNumberAvailable,
83
84 -- location service errors
85 UnauthorizedRequestingNetwork,
86 UnauthorizedLCSCClient,
87 PositionMethodFailure,
88 UnknownOrUnreachableLCSCClient,
89
90 -- Mobility Management errors
91 MM-EventNotSupported
92
93
94 ;
95
96 IMPORTS
97 ERROR
98 FROM TCAPMessages {
```

```

99     ccitt recommendation q 773 modules (2) messages (1) version2 (2)}
100
101     SS-Status
102 FROM MAP-SS-DataTypes {
103     ccitt identified-organization (4) etsi (0) mobileDomain (0)
104     gsm-Network (1) modules (3) map-SS-DataTypes (14) version6 (6)}
105
106     SS-IncompatibilityCause,
107     PW-RegistrationFailureCause,
108     SM-DeliveryFailureCause,
109     SystemFailureParam,
110     DataMissingParam,
111     UnexpectedDataParam,
112     FacilityNotSupParam,
113     UnknownSubscriberParam,
114     NumberChangedParam,
115     UnidentifiedSubParam,
116     RoamingNotAllowedParam,
117     IllegalSubscriberParam,
118     IllegalEquipmentParam,
119     BearerServNotProvParam,
120     TeleservNotProvParam,
121     TracingBufferFullParam,
122     NoRoamingNbParam,
123     OR-NotAllowedParam,
124     AbsentSubscriberParam,
125     BusySubscriberParam,
126     NoSubscriberReplyParam,
127     CallBarredParam,
128     ForwardingViolationParam,
129     ForwardingFailedParam,
130     CUG-RejectParam,
131     ATI-NotAllowedParam,
132     SubBusyForMT-SMS-Param,
133     MessageWaitListFullParam,
134     AbsentSubscriberSM-Param,
135     ResourceLimitationParam,
136     NoGroupCallNbParam,
137     IncompatibleTerminalParam,
138     ShortTermDenialParam,
139     LongTermDenialParam,
140     UnauthorizedRequestingNetwork-Param,
141     UnauthorizedLCSCClient-Param,
142     PositionMethodFailure-Param,
143     UnknownOrUnreachableLCSCClient-Param,
144     MM-EventNotSupported-Param,
145     ATSI-NotAllowedParam,
146     ATM-NotAllowedParam,
147     IllegalSS-OperationParam,
148     SS-NotAvailableParam,
149     SS-SubscriptionViolationParam,
150     InformationNotAvailableParam
151
152
153
154 FROM MAP-ER-DataTypes {
155     ccitt identified-organization (4) etsi (0) mobileDomain (0)
156     gsm-Network (1) modules (3) map-ER-DataTypes (17) version6 (6)}
157 ;
158
159
160 -- generic errors
161
162 

|                                                                                                                          |
|--------------------------------------------------------------------------------------------------------------------------|
| <b>SystemFailure ::= ERROR</b><br>PARAMETER<br>systemFailureParam                      SystemFailureParam<br>-- optional |
|--------------------------------------------------------------------------------------------------------------------------|


163
164
165
166
167 

|                                                                                                                                                                            |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>DataMissing ::= ERROR</b><br>PARAMETER<br>dataMissingParam                        DataMissingParam<br>-- optional<br>-- dataMissingParam must not be used in version <3 |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|


168
169
170
171
172

```

173	UnexpectedDataValue ::= ERROR
174	PARAMETER
175	unexpectedDataParam UnexpectedDataParam
176	-- optional
177	-- unexpectedDataParam must not be used in version <3
178	
179	FacilityNotSupported ::= ERROR
180	PARAMETER
181	facilityNotSupParam FacilityNotSupParam
182	-- optional
183	-- facilityNotSupParam must not be used in version <3
184	
185	IncompatibleTerminal ::= ERROR
186	PARAMETER
187	incompatibleTerminalParam IncompatibleTerminalParam
188	-- optional
189	
190	ResourceLimitation ::= ERROR
191	PARAMETER
192	resourceLimitationParam ResourceLimitationParam
193	-- optional
194	
195	-- identification and numbering errors
196	
197	UnknownSubscriber ::= ERROR
198	PARAMETER
199	unknownSubscriberParam UnknownSubscriberParam
200	-- optional
201	-- unknownSubscriberParam must not be used in version <3
202	
203	NumberChanged ::= ERROR
204	PARAMETER
205	numberChangedParam NumberChangedParam
206	-- optional
207	
208	UnknownMSC ::= ERROR
209	
210	UnidentifiedSubscriber ::= ERROR
211	PARAMETER
212	unidentifiedSubParam UnidentifiedSubParam
213	-- optional
214	-- unidentifiedSubParam must not be used in version <3
215	
216	UnknownEquipment ::= ERROR
217	
218	
219	-- subscription errors
220	
221	RoamingNotAllowed ::= ERROR
222	PARAMETER
223	roamingNotAllowedParam RoamingNotAllowedParam
224	
225	IllegalSubscriber ::= ERROR
226	PARAMETER
227	illegalSubscriberParam IllegalSubscriberParam
228	-- optional
229	-- illegalSubscriberParam must not be used in version <3
230	
231	IllegalEquipment ::= ERROR
232	PARAMETER
233	illegalEquipmentParam IllegalEquipmentParam
234	-- optional
235	-- illegalEquipmentParam must not be used in version <3
236	
237	BearerServiceNotProvisioned ::= ERROR
238	PARAMETER
239	bearerServNotProvParam BearerServNotProvParam
240	-- optional
241	-- bearerServNotProvParam must not be used in version <3
242	
243	TeleserviceNotProvisioned ::= ERROR
244	PARAMETER
245	teleservNotProvParam TeleservNotProvParam
246	-- optional
247	-- teleservNotProvParam must not be used in version <3
248	
249	

250 -- handover errors

251
252 **NoHandoverNumberAvailable** ::= ERROR

253

254 **SubsequentHandoverFailure** ::= ERROR

255

256
257 -- operation and maintenance errors

258

259 **TracingBufferFull** ::= ERROR

260 PARAMETER

261 tracingBufferFullParam TracingBufferFullParam

262 -- optional

263

264

265 -- call handling errors

266

267 **NoRoamingNumberAvailable** ::= ERROR

268 PARAMETER

269 noRoamingNbParam NoRoamingNbParam

270 -- optional

271

272 **AbsentSubscriber** ::= ERROR

273 PARAMETER

274 absentSubscriberParam AbsentSubscriberParam

275 -- optional

276

277 -- absentSubscriberParam must not be used in version <3

278

279 **BusySubscriber** ::= ERROR

280 PARAMETER

281 busySubscriberParam BusySubscriberParam

282 -- optional

283

284 **NoSubscriberReply** ::= ERROR

285 PARAMETER

286 noSubscriberReplyParam NoSubscriberReplyParam

287 -- optional

288

289 **CallBarred** ::= ERROR

290 PARAMETER

291 callBarredParam CallBarredParam

292 -- optional

293

294 **ForwardingViolation** ::= ERROR

295 PARAMETER

296 forwardingViolationParam ForwardingViolationParam

297 -- optional

298

299 **ForwardingFailed** ::= ERROR

300 PARAMETER

301 forwardingFailedParam ForwardingFailedParam

302 -- optional

303

304 **CUG-Reject** ::= ERROR

305 PARAMETER

306 cug-RejectParam CUG-RejectParam

307 -- optional

308

309 **OR-NotAllowed** ::= ERROR

310 PARAMETER

311 or-NotAllowedParam OR-NotAllowedParam

312 -- optional

313

314

315 -- any time interrogation errors

316 **ATI-NotAllowed** ::= ERROR

317 PARAMETER

318 ati-NotAllowedParam ATI-NotAllowedParam

319 -- optional

320

321 -- any time information handling errors

322 **ATSI-NotAllowed** ::= ERROR

323 PARAMETER

324 atsi-NotAllowedParam ATSI-NotAllowedParam

325 -- optional

326

327	ATM-NotAllowed ::= ERROR	
328	PARAMETER	
329	atm-NotAllowedParam	ATM-NotAllowedParam
330	<i>-- optional</i>	
331		
332	InformationNotAvailable ::= ERROR	
333	PARAMETER	
334	informationNotAvailableParam	InformationNotAvailableParam
335	<i>-- optional</i>	
336		
337		
338	<i>-- supplementary service errors</i>	
339		
340	IllegalSS-Operation ::= ERROR	
341	PARAMETER	
342	illegalSS-OperationParam	IllegalSS-OperationParam
343	<i>-- optional</i>	
344	<i>-- illegalSS-OperationParam must not be used in version <3</i>	
345		
346	SS-ErrorStatus ::= ERROR	
347	PARAMETER	
348	ss-Status	SS-Status
349	<i>-- optional</i>	
350		
351	SS-NotAvailable ::= ERROR	
352	PARAMETER	
353	ss-NotAvailableParam	SS-NotAvailableParam
354	<i>-- optional</i>	
355	<i>-- ss-NotAvailableParam must not be used in version <3</i>	
356		
357	SS-SubscriptionViolation ::= ERROR	
358	PARAMETER	
359	ss-SubscriptionViolationParam	SS-SubscriptionViolationParam
360	<i>-- optional</i>	
361	<i>-- ss-NotAvailableParam must not be used in version <3</i>	
362		
363	SS-Incompatibility ::= ERROR	
364	PARAMETER	
365	ss-IncompatibilityCause	SS-IncompatibilityCause
366	<i>-- optional</i>	
367		
368	UnknownAlphabet ::= ERROR	
369		
370	USSD-Busy ::= ERROR	
371		
372	PW-RegistrationFailure ::= ERROR	
373	PARAMETER	
374	pw-RegistrationFailureCause	PW-RegistrationFailureCause
375		
376	NegativePW-Check ::= ERROR	
377		
378	NumberOfPW-AttemptsViolation ::= ERROR	
379		
380	ShortTermDenial ::= ERROR	
381	PARAMETER	
382	shortTermDenialParam	ShortTermDenialParam
383	<i>-- optional</i>	
384		
385	LongTermDenial ::= ERROR	
386	PARAMETER	
387	longTermDenialParam	LongTermDenialParam
388	<i>-- optional</i>	
389		
390		
391	<i>-- short message service errors</i>	
392		
393	SubscriberBusyForMT-SMS ::= ERROR	
394	PARAMETER	
395	subBusyForMT-SMS-Param	SubBusyForMT-SMS-Param
396	<i>-- optional</i>	
397		
398	SM-DeliveryFailure ::= ERROR	
399	PARAMETER	
400	sm-DeliveryFailureCause	SM-DeliveryFailureCause
401		

```

402 MessageWaitingListFull ::= ERROR
403     PARAMETER
404         messageWaitListFullParam      MessageWaitListFullParam
405         -- optional
406
407 AbsentSubscriberSM ::= ERROR
408     PARAMETER
409         absentSubscriberSM-Param      AbsentSubscriberSM-Param
410         -- optional
411
412 -- Group Call errors
413
414 NoGroupCallNumberAvailable ::= ERROR
415     PARAMETER
416         noGroupCallNbParam            NoGroupCallNbParam
417         -- optional
418
419 -- location service errors
420
421 UnauthorizedRequestingNetwork ::= ERROR
422     PARAMETER
423         unauthorizedRequestingNetwork-Param  UnauthorizedRequestingNetwork-Param
424         -- optional
425
426 UnauthorizedLCSCClient ::= ERROR
427     PARAMETER
428         unauthorizedLCSCClient-Param      UnauthorizedLCSCClient-Param
429         -- optional
430
431 PositionMethodFailure ::= ERROR
432     PARAMETER
433         positionMethodFailure-Param      PositionMethodFailure-Param
434         -- optional
435
436 UnknownOrUnreachableLCSCClient ::= ERROR
437     PARAMETER
438         unknownOrUnreachableLCSCClient-Param  UnknownOrUnreachableLCSCClient-Param
439         -- optional
440
441 MM-EventNotSupported ::= ERROR
442     PARAMETER
443         mm-EventNotSupported-Param      MM-EventNotSupported-Param
444         -- optional
445
446 END

```

17.6.7 Group Call operations

```

1  MAP-Group-Call-Operations {
2      ctt identified-organization (4) etsi (0) mobileDomain (0)
3      gsm-Network (1) modules (3) map-Group-Call-Operations (22)
4      version6 (6)}
5
6  DEFINITIONS
7
8  ::=
9
10 BEGIN
11
12 EXPORTS
13     PrepareGroupCall,
14     SendGroupCallEndSignal,
15     ForwardGroupCallSignalling,
16     ProcessGroupCallSignalling
17 ;
18
19 IMPORTS
20     OPERATION
21 FROM TCAPMessages {
22     ctt recommendation q 773 modules (2) messages (1) version2 (2)}
23
24     SystemFailure,
25     UnexpectedDataValue,
26     NoGroupCallNumberAvailable
27 FROM MAP-Errors {
28     ctt identified-organization (4) etsi (0) mobileDomain (0)
29     gsm-Network (1) modules (3) map-Errors (10) version6 (6)}

```

```

30
31     PrepareGroupCallArg,
32     PrepareGroupCallRes,
33     SendGroupCallEndSignalArg,
34     SendGroupCallEndSignalRes,
35     ForwardGroupCallSignallingArg,
36     ProcessGroupCallSignallingArg
37 FROM MAP-GR-DataTypes {
38     ccitt identified-organization (4) etsi (0) mobileDomain (0)
39     gsm-Network (1) modules (3) map-GR-DataTypes (23) version6 (6)}
40
41
42
43 ;
44
45
46 PrepareGroupCall ::= OPERATION --Timer m
47     ARGUMENT
48         prepareGroupCallArg         PrepareGroupCallArg
49     RESULT
50         prepareGroupCallRes         PrepareGroupCallRes
51     ERRORS {
52         SystemFailure,
53         NoGroupCallNumberAvailable,
54         UnexpectedDataValue}
55
56 SendGroupCallEndSignal ::= OPERATION --Timer l
57     ARGUMENT
58         sendGroupCallEndSignalArg     SendGroupCallEndSignalArg
59     RESULT
60         sendGroupCallEndSignalRes     SendGroupCallEndSignalRes
61
62
63 ProcessGroupCallSignalling ::= OPERATION --Timer s
64     ARGUMENT
65         processGroupCallSignallingArg ProcessGroupCallSignallingArg
66
67 ForwardGroupCallSignalling ::= OPERATION --Timer s
68     ARGUMENT
69         forwardGroupCallSignallingArg ForwardGroupCallSignallingArg
70
71 END

```

17.6.8 Location service operations

```

1 MAP-LocationServiceOperations {
2     ccitt identified-organization (4) etsi (0) mobileDomain (0)
3     gsm-Network (1) modules (3) map-LocationServiceOperations (24)
4     version6 (6)}
5
6 DEFINITIONS
7
8 ::=
9
10 BEGIN
11
12 EXPORTS
13     ProvideSubscriberLocation,
14     SendRoutingInfoForLCS,
15     SubscriberLocationReport
16 ;
17
18 IMPORTS
19     OPERATION
20 FROM TCAPMessages {
21     ccitt recommendation q 773 modules (2) messages (1) version2 (2)}
22
23     SystemFailure,
24     DataMissing,
25     UnexpectedDataValue,
26     FacilityNotSupported,
27     UnknownSubscriber,
28     AbsentSubscriber,
29     UnauthorizedRequestingNetwork,
30     UnauthorizedLCSClient,
31     PositionMethodFailure,
32     ResourceLimitation,
33     UnknownOrUnreachableLCSClient,

```

```

34   UnidentifiedSubscriber,
35   IllegalEquipment,
36   IllegalSubscriber
37 FROM MAP-Errors {
38   ccitt identified-organization (4) etsi (0) mobileDomain (0)
39   gsm-Network (1) modules (3) map-Errors (10) version6 (6)}
40
41   RoutingInfoForLCS-Arg,
42   RoutingInfoForLCS-Res,
43   ProvideSubscriberLocation-Arg,
44   ProvideSubscriberLocation-Res,
45   SubscriberLocationReport-Arg,
46   SubscriberLocationReport-Res
47 FROM MAP-LCS-DataTypes {
48   ccitt identified-organization (4) etsi (0) mobileDomain (0)
49   gsm-Network (1) modules (3) map-LCS-DataTypes (25) version6 (6)}
50 ;
51

```

```

52 SendRoutingInfoForLCS ::= OPERATION          --Timer m
53   ARGUMENT
54     routingInfoForLCS-Arg          RoutingInfoForLCS-Arg
55   RESULT
56     routingInfoForLCS-Res          RoutingInfoForLCS-Res
57   ERRORS {
58     SystemFailure,
59     DataMissing,
60     UnexpectedDataValue,
61     FacilityNotSupported,
62     UnknownSubscriber,
63     AbsentSubscriber,
64     UnauthorizedRequestingNetwork }
65

```

```

66 ProvideSubscriberLocation ::= OPERATION      --Timer ml
67   ARGUMENT
68     provideSubscriberLocation-Arg  ProvideSubscriberLocation-Arg
69   RESULT
70     provideSubscriberLocation-Res ProvideSubscriberLocation-Res
71   ERRORS {
72     SystemFailure,
73     DataMissing,
74     UnexpectedDataValue,
75     FacilityNotSupported,
76     UnidentifiedSubscriber,
77     IllegalSubscriber,
78     IllegalEquipment,
79     AbsentSubscriber,
80     UnauthorizedRequestingNetwork,
81     UnauthorizedLCSClient,
82     PositionMethodFailure }
83

```

```

84 SubscriberLocationReport ::= OPERATION      --Timer m
85   ARGUMENT
86     subscriberLocationReport-Arg   SubscriberLocationReport-Arg
87   RESULT
88     subscriberLocationReport-Res   SubscriberLocationReport-Res
89   ERRORS {
90     SystemFailure,
91     DataMissing,
92     ResourceLimitation,
93     UnexpectedDataValue,
94     UnknownSubscriber,
95     UnauthorizedRequestingNetwork,
96     UnknownOrUnreachableLCSClient }
97

```

```

98
99 END
100

```

17.7 MAP constants and data types

17.7.1 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) version6 (6)}
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     CancelLocationRes,
20     PurgeMS-Arg,
21     PurgeMS-Res,
22     SendIdentificationArg,
23     SendIdentificationRes,
24     UpdateGprsLocationArg,
25     UpdateGprsLocationRes,
26     IST-SupportIndicator,
27
28
29     -- handover types
30     PrepareHO-Arg,
31     PrepareHO-Res,
32     PrepareSubsequentHO-Arg,
33
34     -- authentication management types
35     SendAuthenticationInfoArg,
36     SendAuthenticationInfoRes,
37
38     -- security management types
39     EquipmentStatus,
40     Kc,
41
42     -- subscriber management types
43     InsertSubscriberDataArg,
44     InsertSubscriberDataRes,
45     DeleteSubscriberDataArg,
46     DeleteSubscriberDataRes,
47     SubscriberData,
48     ODB-Data,
49     SubscriberStatus,
50     ZoneCodeList,
51     maxNumOfZoneCodes,
52     O-CSI,
53     D-CSI,
54     O-BcsmCamelTDPCriteriaList,
55     T-BCSM-CAMEL-TDP-CriteriaList,
56     SS-CSI,
57     ServiceKey,
58     DefaultCallHandling,
59     CamelCapabilityHandling,
60     BasicServiceCriteria,
61     SupportedCamelPhases,
62     maxNumOfCamelTDPData,
63     CUG-Index,
64     CUG-Interlock,
65     InterCUG-Restrictions,
66     IntraCUG-Options,
67     IST-AlertTimerValue,
68     T-CSI,
69     T-BcsmTriggerDetectionPoint,
70
71     -- fault recovery types
72     ResetArg,
73     RestoreDataArg,
74     RestoreDataRes,
75
76     -- subscriber information enquiry types
77     ProvideSubscriberInfoArg,
78     ProvideSubscriberInfoRes,
79     SubscriberInfo,
80     LocationInformation,
81     SubscriberState,
82
```

```

83  -- any time information enquiry types
84  AnyTimeInterrogationArg,
85  AnyTimeInterrogationRes,
86
87  -- any time information handling types
88  AnyTimeSubscriptionInterrogationArg,
89  AnyTimeSubscriptionInterrogationRes,
90  AnyTimeModificationArg,
91  AnyTimeModificationRes,
92
93  -- subscriber data modification notification types
94  NoteSubscriberDataModifiedArg,
95  NoteSubscriberDataModifiedRes,
96
97  -- gprs location information retrieval types
98  SendRoutingInfoForGprsArg,
99  SendRoutingInfoForGprsRes,
100
101  -- failure reporting types
102  FailureReportArg,
103  FailureReportRes,
104
105  -- gprs notification types
106  NoteMsPresentForGprsArg,
107  NoteMsPresentForGprsRes,
108
109  -- Mobility Management types
110  NoteMM-EventArg,
111  NoteMM-EventRes
112
113
114
115 ;
116
117 IMPORTS
118     maxNumOfSS,
119     SS-SubscriptionOption,
120     SS-List,
121     SS-ForBS-Code,
122     Password
123 FROM MAP-SS-DataTypes {
124     ccitt identified-organization (4) etsi (0) mobileDomain (0)
125     gsm-Network (1) modules (3) map-SS-DataTypes (14) version6 (6)}
126
127     SS-Code
128 FROM MAP-SS-Code {
129     ccitt identified-organization (4) etsi (0) mobileDomain (0)
130     gsm-Network (1) modules (3) map-SS-Code (15) version6 (6)}
131
132     Ext-BearerServiceCode
133 FROM MAP-BS-Code {
134     ccitt identified-organization (4) etsi (0) mobileDomain (0)
135     gsm-Network (1) modules (3) map-BS-Code (20) version6 (6)}
136
137     Ext-TeleserviceCode
138 FROM MAP-TS-Code {
139     ccitt identified-organization (4) etsi (0) mobileDomain (0)
140     gsm-Network (1) modules (3) map-TS-Code (19) version6 (6)}
141
142
143     AddressString,
144     ISDN-AddressString,
145     ISDN-SubaddressString,
146     ExternalSignalInfo,
147     IMSI,
148     TMSI,
149     HLR-List,
150     LMSI,
151     Identity,
152     GlobalCellId,
153     CellIdOrLAI,
154     Ext-BasicServiceCode,
155     NAEA-PreferredCI,
156     EMLPP-Info,
157     SubscriberIdentity,
158     AgeOfLocationInformation,
159     LCSClientExternalID,
160     LCSClientInternalID
161

```

```

162
163
164 FROM MAP-CommonDataTypes {
165     ccitt identified-organization (4) etsi (0) mobileDomain (0)
166     gsm-Network (1) modules (3) map-CommonDataTypes (18) version6 (6)}
167
168     ExtensionContainer
169 FROM MAP-ExtensionDataTypes {
170     ccitt identified-organization (4) etsi (0) mobileDomain (0)
171     gsm-Network (1) modules (3) map-ExtensionDataTypes (21) version6 (6)}
172
173     AbsentSubscriberDiagnosticSM
174 FROM MAP-ER-DataTypes {
175     ccitt identified-organization (4) etsi (0) mobileDomain (0)
176     gsm-Network (1) modules (3) map-ER-DataTypes (17) version6 (6)}
177
178 ;
179 ;
180
181
182 -- location registration types
183
184 UpdateLocationArg ::= SEQUENCE {
185     imsi                               IMSI,
186
187     msc-Number                         [1] ISDN-AddressString,
188     vlr-Number                         ISDN-AddressString,
189     lmsi                               [10] LMSI OPTIONAL,
190     extensionContainer                 ExtensionContainer           OPTIONAL,
191     ... ,
192     vlr-Capability                    [6] VLR-Capability           OPTIONAL }
193
194 VLR-Capability ::= SEQUENCE{
195     supportedCamelPhases                [0] SupportedCamelPhases           OPTIONAL,
196     extensionContainer                 ExtensionContainer           OPTIONAL,
197     ... ,
198     solsaSupportIndicator               [2] NULL                               OPTIONAL,
199     istSupportIndicator                 [1] IST-SupportIndicator         OPTIONAL,
200     superChargerSupportedInServingNetworkEntity [3] SuperChargerInfo       OPTIONAL }
201
202 SuperChargerInfo ::= CHOICE {
203     sendSubscriberData                  [0] NULL,
204     subscriberDataStored                [1] AgeIndicator }
205
206 AgeIndicator ::= OCTET STRING (SIZE (1..6))
207     -- The internal structure of this parameter is implementation specific.
208
209
210 IST-SupportIndicator ::= ENUMERATED {
211     basicISTSupported                  (0),
212     istCommandSupported                (1), ...}
213 -- exception handling:
214 -- reception of values > 1 shall be mapped to ' istCommandSupported '
215
216
217 UpdateLocationRes ::= SEQUENCE {
218     hlr-Number                         ISDN-AddressString,
219
220     extensionContainer                 ExtensionContainer           OPTIONAL,
221     ... }
222
223 CancelLocationArg ::= [3] SEQUENCE {
224     identity                           Identity,
225     cancellationType                   CancellationType           OPTIONAL,
226     extensionContainer                 ExtensionContainer           OPTIONAL,
227     ...}
228
229
230 CancellationType ::= ENUMERATED {
231     updateProcedure                    (0),
232     subscriptionWithdraw                (1),
233     ...}
234 -- The HLR shall not send values other than listed above
235
236

```

237	CancelLocationRes ::= SEQUENCE {		
238	extensionContainer	ExtensionContainer	OPTIONAL,
239	...}		
240			
241	PurgeMS-Arg ::= [3] SEQUENCE {		
242	imsi	IMSI,	
243	vlr-Number	[0] ISDN-AddressString	OPTIONAL,
244	sgsn-Number	[1] ISDN-AddressString	OPTIONAL,
245	extensionContainer	ExtensionContainer	OPTIONAL,
246	...}		
247			
248	PurgeMS-Res ::= SEQUENCE {		
249	freezeTMSI	[0] NULL	OPTIONAL,
250	freezeP-TMSI	[1] NULL	OPTIONAL,
251	extensionContainer	ExtensionContainer	OPTIONAL,
252	...}		
253			
254	SendIdentificationArg ::= SEQUENCE {		
255	tmsi	TMSI,	
256	numberOfRequestedVectors	NumberOfRequestedVectors,	
257	segmentationProhibited	NULL	OPTIONAL,
258	-- if segmentation is prohibited the previous VLR shall not send the result		
259	-- within a TC-CONTINUE message.		
260	extensionContainer	ExtensionContainer	OPTIONAL,
261	...}		
262			
263	SendIdentificationRes ::= [3] SEQUENCE {		
264	imsi	IMSI	OPTIONAL,
265	-- IMSI must be present if SendIdentificationRes is not segmented.		
266	-- If the TC-Continue segmentation option is taken the IMSI must be		
267	-- present in one segmented transmission of SendIdentificationRes.		
268	authenticationSetList	AuthenticationSetList	OPTIONAL,
269	extensionContainer	[2] ExtensionContainer	OPTIONAL,
270	...}		
271			
272	AuthenticationSetList ::= CHOICE {		
273	tripletList	[0] TripletList,	
274	quintupletList	[1] QuintupletList }	
275			
276	TripletList ::= SEQUENCE SIZE (1..5) OF		
277		AuthenticationTriplet	
278			
279	QuintupletList ::= SEQUENCE SIZE (1..5) OF		
280		AuthenticationQuintuplet	
281			
282	AuthenticationTriplet ::= SEQUENCE {		
283	rand	RAND,	
284	sres	SRES,	
285	kc	Kc,	
286	...}		
287			
288	AuthenticationQuintuplet ::= SEQUENCE {		
289	rand	RAND,	
290	xres	XRES,	
291	ck	CK,	
292	ik	IK,	
293	autn	AUTN,	
294	...}		
295			
296	RAND ::= OCTET STRING (SIZE (16))		
297			
298	SRES ::= OCTET STRING (SIZE (4))		
299			
300	Kc ::= OCTET STRING (SIZE (8))		
301			
302	XRES ::= OCTET STRING (SIZE (4..16))		
303			
304	CK ::= OCTET STRING (SIZE (16))		
305			
306	IK ::= OCTET STRING (SIZE (16))		
307			
308	AUTN ::= OCTET STRING (SIZE (14..18))		
309			
310	AUTS ::= OCTET STRING (SIZE (12..16))		
311			

312 -- gprs location registration types

```
313
314 UpdateGprsLocationArg ::= SEQUENCE {
315     imsi                IMSI,
316     sgsn-Number         ISDN-AddressString,
317     sgsn-Address        GSN-Address,
318     extensionContainer  ExtensionContainer          OPTIONAL,
319     ... ,
320     sgsn-Capability     [0] SGSN-Capability        OPTIONAL }
```

```
321
322 SGSN-Capability ::= SEQUENCE{
323     solsaSupportIndicator  NULL                OPTIONAL,
324     extensionContainer    [1] ExtensionContainer  OPTIONAL,
325     ... ,
326     superChargerSupportedInServingNetworkEntity [2] SuperChargerInfo  OPTIONAL ,
327     gprsEnhancementsSupportIndicator [3] NULL                OPTIONAL,
328     supportedCamelPhases [4] SupportedCamelPhases  OPTIONAL }
```

```
329
330 GSN-Address ::= OCTET STRING (SIZE (5..17))
331 -- Octets are coded according to TS GSM 03.03
```

```
332
333 UpdateGprsLocationRes ::= SEQUENCE {
334     hlr-Number           ISDN-AddressString,
335     extensionContainer  ExtensionContainer          OPTIONAL,
336     ... }
```

337 -- handover types

```
338
339 PrepareHO-Arg ::= SEQUENCE {
340     targetCellId        GlobalCellId              OPTIONAL,
341     ho-NumberNotRequired NULL                    OPTIONAL,
342     bss-APDU            ExternalSignalInfo        OPTIONAL,
343     ... }
```

```
344
345 PrepareHO-Res ::= SEQUENCE {
346     handoverNumber      ISDN-AddressString        OPTIONAL,
347     bss-APDU            ExternalSignalInfo        OPTIONAL,
348     ... }
```

```
349
350 PrepareSubsequentHO-Arg ::= SEQUENCE {
351     targetCellId        GlobalCellId,
352     targetMSC-Number    ISDN-AddressString,
353     bss-APDU            ExternalSignalInfo,
354     ... }
```

355 -- authentication management types

```
356
357 SendAuthenticationInfoArg ::= SEQUENCE {
358     imsi                [0] IMSI,
359     numberOfRequestedVectors  NumberOfRequestedVectors,
360     segmentationProhibited  NULL                OPTIONAL,
361     -- if segmentation is prohibited the HLR shall not send the result within
362     -- a TC-CONTINUE message.
363     immediateResponsePreferred [1] NULL                OPTIONAL,
364     -- if present, the HLR may send an immediate response with the available authentication
365     -- vectors (see § 8.5.2 for more information).
366     re-synchronisationInfo  Re-synchronisationInfo  OPTIONAL,
367     extensionContainer      [2] ExtensionContainer    OPTIONAL,
368     ... }
```

```
369
370 NumberOfRequestedVectors ::= INTEGER (1..5)
```

```
371
372 Re-synchronisationInfo ::= SEQUENCE {
373     rand                RAND,
374     rand-ms             RAND,
375     auts                AUTS,
376     ... }
```

```
377
378 SendAuthenticationInfoRes ::= [3] SEQUENCE {
379     authenticationSetList  AuthenticationSetList  OPTIONAL,
380     extensionContainer     ExtensionContainer          OPTIONAL,
381     ... }
```

382 -- security management types

383

```

388 EquipmentStatus ::= ENUMERATED {
389     whiteListed (0),
390     blackListed (1),
391     greyListed (2)}
392
393
394 -- subscriber management types
395
396 InsertSubscriberDataArg ::= SEQUENCE {
397     imsi [0] IMSI OPTIONAL,
398     COMPONENTS OF SubscriberData,
399     extensionContainer [14] ExtensionContainer OPTIONAL,
400     ... ,
401     naea-PreferredCI [15] NAEA-PreferredCI OPTIONAL,
402     -- naea-PreferredCI is included at the discretion of the HLR operator.
403     gprsSubscriptionData [16] GPRSSubscriptionData OPTIONAL,
404     roamingRestrictedInSgsnDueToUnsupportedFeature [23]
405     NULL OPTIONAL,
406     networkAccessMode [24] NetworkAccessMode OPTIONAL,
407     lsaInformation [25] LSAInformation OPTIONAL,
408     lmu-Indicator [21] NULL OPTIONAL,
409     lcsInformation [22] LCSInformation OPTIONAL,
410     istAlertTimer [26] IST-AlertTimerValue OPTIONAL,
411     superChargerSupportedInHLR [27] AgeIndicator OPTIONAL
412 }
413 -- If the Network Access Mode parameter is sent, it shall be present only in
414 -- the first sequence if the segmentation is used
415
416 IST-AlertTimerValue ::= INTEGER (15..255)
417
418 LCSInformation ::= SEQUENCE {
419     gmlc-List [0] GMLC-List OPTIONAL,
420     lcs-PrivacyExceptionList [1] LCS-PrivacyExceptionList OPTIONAL,
421     molr-List [2] MOLR-List OPTIONAL,
422     ...}
423
424 GMLC-List ::= SEQUENCE SIZE (1..maxNumOfGMLC) OF
425     ISDN-AddressString
426 -- if segmentation is used, the complete GMLC-List shall be sent in one segment
427
428 maxNumOfGMLC INTEGER ::= 5
429
430
431 NetworkAccessMode ::= ENUMERATED {
432     bothMSCAndSGSN (0),
433     onlyMSC (1),
434     onlySGSN (2),
435     ...}
436 -- if unknown values are received in NetworkAccessMode
437 -- they shall be discarded.
438
439 GPRSDataList ::= SEQUENCE SIZE (1..maxNumOfPDP-Contexts) OF
440     PDP-Context
441
442 maxNumOfPDP-Contexts INTEGER ::= 50
443
444 PDP-Context ::= SEQUENCE {
445     pdp-ContextId ContextId,
446     pdp-Type [16] PDP-Type,
447     pdp-Address [17] PDP-Address OPTIONAL,
448     qos-Subscribed [18] QoS-Subscribed,
449     vplmnAddressAllowed [19] NULL OPTIONAL,
450     apn [20] APN ,
451     extensionContainer [21] ExtensionContainer OPTIONAL,
452     ... ,
453     ext-QoS-Subscribed [0] Ext-QoS-Subscribed OPTIONAL }
454 -- qos-Subscribed shall be discarded if ext-QoS-Subscribed is received and supported
455
456 ContextId ::= INTEGER (1..maxNumOfPDP-Contexts)
457

```

```

458 GPRSSubscriptionData ::= SEQUENCE {
459     completeDataListIncluded          NULL                OPTIONAL,
460
461     -- If segmentation is used, completeDataListIncluded may only be present in the
462     -- first segment.
463     gprsDataList                      [1] GPRSDataList,
464     extensionContainer                 [2] ExtensionContainer OPTIONAL,
465     ... ,
466     sgsn-CAMEL-SubscriptionInfo       [3] SGSN-CAMEL-SubscriptionInfo OPTIONAL }
467

```

```

468 SGSN-CAMEL-SubscriptionInfo ::= SEQUENCE {
469     gprs-CSI                          [0] GPRS-CSI        OPTIONAL,
470     sms-CSI                            [1] SMS-CSI        OPTIONAL,
471     extensionContainer                 [2] ExtensionContainer OPTIONAL,
472     ... }
473

```

```

474 GPRS-CSI ::= SEQUENCE {
475     gprs-CamelTDPDataList             [0] GPRS-CamelTDPDataList,
476     camelCapabilityHandling           [1] CamelCapabilityHandling,
477     extensionContainer                 [2] ExtensionContainer   OPTIONAL,
478     notificationToCSE                 [3] NULL              OPTIONAL,
479     csiActive                          [4] NULL              OPTIONAL,
480     ... }
481 -- notificationToCSE and csiActive shall not be present when GPRS-CSI is sent to SGSN.
482 -- They may only be included in ATSI/ATM Ack message.
483

```

```

484 GPRS-CamelTDPDataList ::= SEQUENCE SIZE (1..maxNumOfCamelTDPData) OF
485     GPRS-CamelTDPData
486 -- GPRS-CamelTDPDataList shall not contain more than one instance of
487 -- GPRS-CamelTDPData containing the same value for gprs-TriggerDetectionPoint.
488

```

```

489 GPRS-CamelTDPData ::= SEQUENCE {
490     gprs-TriggerDetectionPoint        [0] GPRS-TriggerDetectionPoint,
491     serviceKey                        [1] ServiceKey,
492     gsmSCF-Address                   [2] ISDN-AddressString,
493     defaultSessionHandling            [3] DefaultGPRS-Handling,
494     extensionContainer                 [4] ExtensionContainer   OPTIONAL,
495     ...
496 }
497

```

```

498 DefaultGPRS-Handling ::= ENUMERATED {
499     continueTransaction (0) ,
500     releaseTransaction (1) ,
501     ... }
502 -- exception handling:
503 -- reception of values in range 2-31 shall be treated as "continueTransaction"
504 -- reception of values greater than 31 shall be treated as "releaseTransaction"
505

```

```

506 GPRS-TriggerDetectionPoint ::= ENUMERATED {
507     attach                            (1),
508     attachChangeOfPosition            (2),
509     pdp-ContextEstablishment          (11),
510     pdp-ContextEstablishmentAcknowledgement (12),
511     pdp-ContextChangeOfPosition      (14),
512     ... }
513 -- exception handling:
514 -- For GPRS-CamelTDPData sequences containing this parameter with any
515 -- other value than the ones listed the receiver shall ignore the whole
516 -- GPRS-CamelTDPData sequence.
517

```

```

518 APN ::= OCTET STRING (SIZE (2..63))
519     -- Octets are coded according to TS GSM 03.03
520
521

```

```

522 PDP-Type ::= OCTET STRING (SIZE (2))
523     -- Octets are coded according to TS GSM 09.60
524

```

```

525 PDP-Address ::= OCTET STRING (SIZE (1..16))
526 -- Octets are coded according to TS GSM 09.60
527
528 -- The possible size values are:
529 -- 1-7 octets X.25 address type
530 -- 4 octets IPv4 address type
531 -- 16 octets Ipv6 address type
532

```

```

533 QoS-Subscribed ::= OCTET STRING (SIZE (3))
534 -- Octets are coded according to TS GSM 04.08.
535

```

```

536 Ext-QoS-Subscribed ::= OCTET STRING (SIZE (3..15))
537 -- Octets are coded according to 3G TS 24.008.
538

```

```

539 LSAOnlyAccessIndicator ::= ENUMERATED {
540     accessOutsideLSAsAllowed (0),
541     accessOutsideLSAsRestricted (1)}
542

```

```

543 LSADataList ::= SEQUENCE SIZE (1..maxNumOfLSAs) OF
544     LSAData
545

```

```

546 maxNumOfLSAs INTEGER ::= 20
547

```

```

548 LSAData ::= SEQUENCE {
549     lsaIdentity                [0] LSAIdentity,
550     lsaPriority                [1] LSAPriority,
551     lsaActiveModeIndicator    [2] NULL                OPTIONAL,
552     lsaActiveModeSupportIndicator [3] NULL                OPTIONAL,
553     extensionContainer        [4] ExtensionContainer  OPTIONAL,
554     ...}
555

```

```

556 LSAInformation ::= SEQUENCE {
557     completeDataListIncluded    NULL                OPTIONAL,
558
559     -- If segmentation is used, completeDataListIncluded may only be present in the
560     -- first segment.
561     lsaOnlyAccessIndicator      [1] LSAOnlyAccessIndicator  OPTIONAL,
562     lsaDataList                [2] LSADataList            OPTIONAL,
563     extensionContainer          [3] ExtensionContainer  OPTIONAL,
564     ...}
565

```

```

566 LSAIdentity ::= OCTET STRING (SIZE (3))
567 -- Octets are coded according to TS GSM 03.03
568

```

```

569 LSAPriority ::= OCTET STRING (SIZE (1))
570 -- Octets are coded according to TS GSM 08.08
571

```

```

572
573 SubscriberData ::= SEQUENCE {
574     msisdn                    [1] ISDN-AddressString  OPTIONAL,
575     category                  [2] Category              OPTIONAL,
576     subscriberStatus          [3] SubscriberStatus      OPTIONAL,
577     bearerServiceList         [4] BearerServiceList      OPTIONAL,
578     -- The exception handling for reception of unsupported / not allocated
579     -- bearerServiceCodes is defined in section 6.8.1
580     teleserviceList           [6] TeleserviceList        OPTIONAL,
581     -- The exception handling for reception of unsupported / not allocated
582     -- teleserviceCodes is defined in section 6.8.1
583     provisionedSS             [7] Ext-SS-InfoList        OPTIONAL,
584     odb-Data                  [8] ODB-Data              OPTIONAL,
585     roamingRestrictionDueToUnsupportedFeature [9] NULL                OPTIONAL,
586     regionalSubscriptionData   [10] ZoneCodeList         OPTIONAL,
587     vbsSubscriptionData        [11] VBSDataList          OPTIONAL,
588     vgcsSubscriptionData       [12] VGCSDataList         OPTIONAL,
589     vlrCamelSubscriptionInfo   [13] VlrCamelSubscriptionInfo OPTIONAL,
590     }
591

```

```

592 Category ::= OCTET STRING (SIZE (1))
593 -- The internal structure is defined in CCITT Rec Q.763.
594

```

```

595 SubscriberStatus ::= ENUMERATED {
596     serviceGranted (0),
597     operatorDeterminedBarring (1)}
598

```



```

599 BearerServiceList ::= SEQUENCE SIZE (1..maxNumOfBearerServices) OF
600     Ext-BearerServiceCode
601
602 maxNumOfBearerServices INTEGER ::= 50
603
604 TeleserviceList ::= SEQUENCE SIZE (1..maxNumOfTeleservices) OF
605     Ext-TeleserviceCode
606
607 maxNumOfTeleservices INTEGER ::= 20
608
609 ODB-Data ::= SEQUENCE {
610     odb-GeneralData          ODB-GeneralData,
611     odb-HPLMN-Data          ODB-HPLMN-Data          OPTIONAL,
612     extensionContainer      ExtensionContainer      OPTIONAL,
613     ...}
614
615 ODB-GeneralData ::= BIT STRING {
616     allOG-CallsBarred (0),
617     internationalOGCallsBarred (1),
618     internationalOGCallsNotToHPLMN-CountryBarred (2),
619     interzonalOGCallsBarred (6),
620     interzonalOGCallsNotToHPLMN-CountryBarred (7),
621     interzonalOGCallsAndInternationalOGCallsNotToHPLMN-CountryBarred (8),
622     premiumRateInformationOGCallsBarred (3),
623     premiumRateEntertainmentOGCallsBarred (4),
624     ss-AccessBarred (5),
625     allECT-Barred (9),
626     chargeableECT-Barred (10),
627     internationalECT-Barred (11),
628     interzonalECT-Barred (12),
629     doublyChargeableECT-Barred (13),
630     multipleECT-Barred (14)} (SIZE (15..32))
631 -- exception handling: reception of unknown bit assignments in the
632 -- ODB-GeneralData type shall be treated like unsupported ODB-GeneralData
633
634 ODB-HPLMN-Data ::= BIT STRING {
635     plmn-SpecificBarringType1 (0),
636     plmn-SpecificBarringType2 (1),
637     plmn-SpecificBarringType3 (2),
638     plmn-SpecificBarringType4 (3)} (SIZE (4..32))
639 -- exception handling: reception of unknown bit assignments in the
640 -- ODB-HPLMN-Data type shall be treated like unsupported ODB-HPLMN-Data
641
642 Ext-SS-InfoList ::= SEQUENCE SIZE (1..maxNumOfSS) OF
643     Ext-SS-Info
644
645 Ext-SS-Info ::= CHOICE {
646     forwardingInfo          [0] Ext-ForwInfo,
647     callBarringInfo        [1] Ext-CallBarInfo,
648     cug-Info                [2] CUG-Info,
649     ss-Data                 [3] Ext-SS-Data,
650     emlpp-Info              [4] EMLPP-Info}
651
652
653 Ext-ForwInfo ::= SEQUENCE {
654     ss-Code                  SS-Code,
655     forwardingFeatureList    Ext-ForwFeatureList,
656     extensionContainer      [0] ExtensionContainer      OPTIONAL,
657     ...}
658
659 Ext-ForwFeatureList ::= SEQUENCE SIZE (1..maxNumOfExt-BasicServiceGroups) OF
660     Ext-ForwFeature
661
662 Ext-ForwFeature ::= SEQUENCE {
663     basicService             Ext-BasicServiceCode          OPTIONAL,
664     ss-Status [4] Ext-SS-Status,
665     forwardedToNumber        [5] ISDN-AddressString        OPTIONAL,
666     -- When this data type is sent from an HLR which supports CAMEL Phase 2
667     -- to a VLR that supports CAMEL Phase 2 the VLR shall not check the
668     -- format of the number
669     forwardedToSubaddress    [8] ISDN-SubaddressString     OPTIONAL,
670     forwardingOptions        [6] Ext-ForwOptions            OPTIONAL,
671     noReplyConditionTime     [7] Ext-NoRepCondTime          OPTIONAL,
672     extensionContainer      [9] ExtensionContainer          OPTIONAL,
673     ...}
674

```

675	Ext-SS-Status ::= OCTET STRING (SIZE (1..5))
676	
677	-- OCTET 1:
678	--
679	-- bits 8765: 0000 (unused)
680	-- bits 4321: Used to convey the "P bit", "R bit", "A bit" and "Q bit",
681	-- representing supplementary service state information
682	-- as defined in TS GSM 03.11
683	
684	-- bit 4: "Q bit"
685	
686	-- bit 3: "P bit"
687	
688	-- bit 2: "R bit"
689	
690	-- bit 1: "A bit"
691	
692	-- OCTETS 2-5: reserved for future use. They shall be discarded if
693	-- received and not understood.
694	
695	
696	Ext-ForwOptions ::= OCTET STRING (SIZE (1..5))
697	
698	-- OCTET 1:
699	
700	-- bit 8: notification to forwarding party
701	-- 0 no notification
702	-- 1 notification
703	
704	-- bit 7: redirecting presentation
705	-- 0 no presentation
706	-- 1 presentation
707	
708	-- bit 6: notification to calling party
709	-- 0 no notification
710	-- 1 notification
711	
712	-- bit 5: 0 (unused)
713	
714	-- bits 43: forwarding reason
715	-- 00 ms not reachable
716	-- 01 ms busy
717	-- 10 no reply
718	-- 11 unconditional
719	
720	-- bits 21: 00 (unused)
721	
722	-- OCTETS 2-5: reserved for future use. They shall be discarded if
723	-- received and not understood.
724	
725	Ext-NoRepCondTime ::= INTEGER (1..100)
726	-- Only values 5-30 are used.
727	-- Values in the ranges 1-4 and 31-100 are reserved for future use
728	-- If received:
729	-- values 1-4 shall be mapped on to value 5
730	-- values 31-100 shall be mapped on to value 30
731	
732	Ext-CallBarInfo ::= SEQUENCE {
733	ss-Code SS-Code,
734	callBarringFeatureList Ext-CallBarFeatureList,
735	extensionContainer ExtensionContainer OPTIONAL,
736	...}
737	
738	Ext-CallBarFeatureList ::= SEQUENCE SIZE (1..maxNumOfExt-BasicServiceGroups) OF
739	Ext-CallBarringFeature
740	
741	Ext-CallBarringFeature ::= SEQUENCE {
742	basicService Ext-BasicServiceCode OPTIONAL,
743	ss-Status [4] Ext-SS-Status,
744	extensionContainer ExtensionContainer OPTIONAL,
745	...}
746	
747	CUG-Info ::= SEQUENCE {
748	cug-SubscriptionList CUG-SubscriptionList,
749	cug-FeatureList CUG-FeatureList OPTIONAL,
750	extensionContainer [0] ExtensionContainer OPTIONAL,
751	...}
752	

753	CUG-SubscriptionList ::= SEQUENCE SIZE (0..maxNumOfCUG) OF
754	CUG-Subscription
755	
756	CUG-Subscription ::= SEQUENCE {
757	cug-Index CUG-Index,
758	cug-Interlock CUG-Interlock,
759	intraCUG-Options IntraCUG-Options,
760	basicServiceGroupList Ext-BasicServiceGroupList OPTIONAL,
761	extensionContainer [0] ExtensionContainer OPTIONAL,
762	...}
763	
764	CUG-Index ::= INTEGER (0..32767)
765	-- The internal structure is defined in ETS 300 138.
766	
767	CUG-Interlock ::= OCTET STRING (SIZE (4))
768	
769	IntraCUG-Options ::= ENUMERATED {
770	noCUG-Restrictions (0),
771	cugIC-CallBarred (1),
772	cugOG-CallBarred (2)}
773	
774	maxNumOfCUG INTEGER ::= 10
775	
776	CUG-FeatureList ::= SEQUENCE SIZE (1..maxNumOfExt-BasicServiceGroups) OF
777	CUG-Feature
778	
779	Ext-BasicServiceGroupList ::= SEQUENCE SIZE (1..maxNumOfExt-BasicServiceGroups) OF
780	Ext-BasicServiceCode
781	
782	maxNumOfExt-BasicServiceGroups INTEGER ::= 32
783	
784	CUG-Feature ::= SEQUENCE {
785	basicService Ext-BasicServiceCode OPTIONAL,
786	preferentialCUG-Indicator CUG-Index OPTIONAL,
787	interCUG-Restrictions InterCUG-Restrictions,
788	extensionContainer ExtensionContainer OPTIONAL,
789	...}
790	
791	InterCUG-Restrictions ::= OCTET STRING (SIZE (1))
792	
793	-- bits 876543: 000000 (unused)
794	-- Exception handling:
795	-- bits 876543 shall be ignored if received and not understood
796	
797	-- bits 21
798	-- 00 CUG only facilities
799	-- 01 CUG with outgoing access
800	-- 10 CUG with incoming access
801	-- 11 CUG with both outgoing and incoming access
802	
803	Ext-SS-Data ::= SEQUENCE {
804	ss-Code SS-Code,
805	ss-Status [4] Ext-SS-Status,
806	ss-SubscriptionOption SS-SubscriptionOption OPTIONAL,
807	basicServiceGroupList Ext-BasicServiceGroupList OPTIONAL,
808	extensionContainer [5] ExtensionContainer OPTIONAL,
809	...}
810	
811	LCS-PrivacyExceptionList ::= SEQUENCE SIZE (1..maxNumOfPrivacyClass) OF
812	LCS-PrivacyClass
813	
814	maxNumOfPrivacyClass INTEGER ::= 4

```

815
816 LCS-PrivacyClass ::= SEQUENCE {
817     ss-Code                SS-Code,
818     ss-Status              Ext-SS-Status,
819     privacyVerificationByMSuser [0] NULL OPTIONAL,
820     -- privacyVerificationByMSUser is expected only for SS-code = callunrelated
821     externalClientList     [1] ExternalClientList OPTIONAL,
822     -- externalClientList is expected only for SS-code = callunrelated
823     plmnClientList        [2] PLMNClientList OPTIONAL,
824     -- plmnClientList is expected only for SS-code = plmn
825     extensionContainer     [3] ExtensionContainer OPTIONAL,
826     -- if segmentation is used, the complete LCS-PrivacyClass shall be sent in one segment
827     ...}
828
829 ExternalClientList ::= SEQUENCE SIZE (0..maxNumOfExternalClient) OF
830     ExternalClient
831
832 maxNumOfExternalClient INTEGER ::= 5
833
834 PLMNClientList ::= SEQUENCE SIZE (1..maxNumOfPLMNClient) OF
835     LCSCClientInternalID
836
837 maxNumOfPLMNClient INTEGER ::= 5
838
839 ExternalClient ::= SEQUENCE {
840     clientIdentity          LCSCClientExternalID,
841     gmlc-Restriction       [0] GMLC-Restriction OPTIONAL,
842     notificationToMSUser   [1] NotificationToMSUser OPTIONAL,
843     extensionContainer     [2] ExtensionContainer OPTIONAL,
844     ...}
845
846 GMLC-Restriction ::= ENUMERATED {
847     gmlc-List              (0),
848     home-Country          (1)}
849
850 NotificationToMSUser ::= ENUMERATED {
851     notification           (0),
852     notificationWithPrivacyVerification (1)}
853
854 MOLR-List ::= SEQUENCE SIZE (1..maxNumOfMOLR-Class) OF
855     MOLR-Class
856
857 maxNumOfMOLR-Class INTEGER ::= 3
858
859 MOLR-Class ::= SEQUENCE {
860     ss-Code                SS-Code,
861     ss-Status              Ext-SS-Status,
862     extensionContainer     [0] ExtensionContainer OPTIONAL,
863     ...}
864
865 ZoneCodeList ::= SEQUENCE SIZE (1..maxNumOfZoneCodes)
866     OF ZoneCode
867
868 ZoneCode ::= OCTET STRING (SIZE (2))
869     -- internal structure is defined in TS GSM 03.03
870
871 maxNumOfZoneCodes INTEGER ::= 10
872
873 InsertSubscriberDataRes ::= SEQUENCE {
874     teleserviceList        [1] TeleserviceList OPTIONAL,
875     bearerServiceList     [2] BearerServiceList OPTIONAL,
876     ss-List                [3] SS-List OPTIONAL,
877     odb-GeneralData       [4] ODB-GeneralData OPTIONAL,
878     regionalSubscriptionResponse [5] RegionalSubscriptionResponse OPTIONAL,
879     supportedCamelPhases  [6] SupportedCamelPhases OPTIONAL,
880     extensionContainer     [7] ExtensionContainer OPTIONAL,
881     ...}
882
883
884 RegionalSubscriptionResponse ::= ENUMERATED {
885     networkNode-AreaRestricted (0),
886     tooManyZoneCodes          (1),
887     zoneCodesConflict         (2),
888     regionalSubscNotSupported (3)}
889

```

890	DeleteSubscriberDataArg ::= SEQUENCE {		
891	imsi	[0] IMSI,	
892	basicServiceList	[1] BasicServiceList	OPTIONAL,
893	-- The exception handling for reception of unsupported/not allocated		
894	-- basicServiceCodes is defined in section 6.8.2		
895	ss-List	[2] SS-List	OPTIONAL,
896	roamingRestrictionDueToUnsupportedFeature	[4] NULL	OPTIONAL,
897	regionalSubscriptionIdentifier	[5] ZoneCode	OPTIONAL,
898	vbsGroupIndication	[7] NULL	OPTIONAL,
899	vgcsGroupIndication	[8] NULL	OPTIONAL,
900	camelSubscriptionInfoWithdraw	[9] NULL	OPTIONAL,
901	extensionContainer	[6] ExtensionContainer	OPTIONAL,
902	...		
903	gprsSubscriptionDataWithdraw	[10] GPRSSubscriptionDataWithdraw	OPTIONAL,
904	roamingRestrictedInSgsnDueToUnsupportedFeature	[11] NULL	OPTIONAL,
905	lsaInformationWithdraw	[12] LSAInformationWithdraw	OPTIONAL,
906	gmlc-ListWithdraw	[13] NULL	OPTIONAL,
907	istInformationWithdraw	[14] NULL	OPTIONAL }
908			
909	GPRSSubscriptionDataWithdraw ::= CHOICE {		
910	allGPRSData	NULL,	
911	contextIdList	ContextIdList}	
912			
913	ContextIdList ::= SEQUENCE SIZE (1..maxNumOfPDP-Contexts) OF		
914	ContextId		
915			
916	LSAInformationWithdraw ::= CHOICE {		
917	allLSAData	NULL,	
918	lsaIdentityList	LSAIdentityList }	
919			
920	LSAIdentityList ::= SEQUENCE SIZE (1..maxNumOfLSAs) OF		
921	LSAIdentity		
922			
923	BasicServiceList ::= SEQUENCE SIZE (1..maxNumOfBasicServices) OF		
924	Ext-BasicServiceCode		
925			
926	maxNumOfBasicServices INTEGER ::= 70		
927			
928	DeleteSubscriberDataRes ::= SEQUENCE {		
929	regionalSubscriptionResponse	[0]	
930		RegionalSubscriptionResponse	OPTIONAL,
931	extensionContainer	ExtensionContainer	OPTIONAL,
932	...}		
933			
934	VlrCamelSubscriptionInfo ::= SEQUENCE {		
935	o-CSI	[0] O-CSI	OPTIONAL,
936	extensionContainer	[1] ExtensionContainer	OPTIONAL,
937	...		
938	ss-CSI	[2] SS-CSI	OPTIONAL,
939	o-BcsmCamelTDP-CriteriaList	[4] O-BcsmCamelTDPCriteriaList	OPTIONAL,
940	tif-CSI	[3] NULL	OPTIONAL,
941	m-CSI	[5] M-CSI	OPTIONAL,
942	sms-CSI	[6] SMS-CSI	OPTIONAL,
943	vt-CSI	[7] T-CSI	OPTIONAL,
944	t-BCSM-CAMEL-TDP-CriteriaList	[8] T-BCSM-CAMEL-TDP-CriteriaList	OPTIONAL,
945	d-CSI	[9] D-CSI	OPTIONAL
946	}		
947			
948	D-CSI ::= SEQUENCE {		
949	dp-AnalysedInfoCriteriaList	DP-AnalysedInfoCriteriaList,	
950	camelCapabilityHandling	CamelCapabilityHandling,	
951	extensionContainer	ExtensionContainer	OPTIONAL,
952	...}		
953			
954	DP-AnalysedInfoCriteriaList ::= SEQUENCE SIZE (1..maxNumOfDP-AnalysedInfoCriteria) OF		
955	DP-AnalysedInfoCriterium		
956			
957	maxNumOfDP-AnalysedInfoCriteria INTEGER ::= 10		
958			
959	DP-AnalysedInfoCriterium ::= SEQUENCE {		
960	dialledNumber	ISDN-AddressString,	
961	serviceKey	ServiceKey,	
962	gsmSCF-Address	ISDN-AddressString,	
963	defaultCallHandling	DefaultCallHandling,	
964	extensionContainer	ExtensionContainer	OPTIONAL,
965	...}		
966			

```

967 SS-CSI ::= SEQUENCE {
968     ss-CamelData                SS-CamelData,
969     extensionContainer           ExtensionContainer           OPTIONAL,
970     ... }
971
972 SS-CamelData ::= SEQUENCE {
973     ss-EventList                SS-EventList,
974     gsmSCF-Address              ISDN-AddressString,
975     extensionContainer           [0] ExtensionContainer           OPTIONAL,
976     ... ,
977     notificationToCSE           [1] NULL                          OPTIONAL,
978     csiActive                    [2] NULL                          OPTIONAL
979 }
980 -- notificationToCSE and csiActive shall not be present when SS-CSI is sent to VLR.
981 -- They may only be included in ATSI/ATM Ack message.
982
983 SS-EventList ::= SEQUENCE SIZE (1..maxNumOfCamelSSEvents) OF SS-Code
984 -- Actions for the following SS-Code values are defined in CAMEL Phase 3:
985 -- ect                SS-Code ::= '00110001'B
986 -- multiPTY           SS-Code ::= '01010001'B
987 -- cd                 SS-Code ::= '00100100'B
988 -- ccbs              SS-Code ::= '01000100'B
989 -- all other SS codes shall be ignored
990
991 maxNumOfCamelSSEvents INTEGER ::= 10
992
993 O-CSI ::= SEQUENCE {
994     o-BcsmCamelTDPDataList      O-BcsmCamelTDPDataList,
995     extensionContainer           ExtensionContainer           OPTIONAL,
996     ... ,
997     camelCapabilityHandling     [0] CamelCapabilityHandling  OPTIONAL,
998     notificationToCSE           [1] NULL                          OPTIONAL,
999     csiActive                    [2] NULL                          OPTIONAL
1000 }
1001 -- notificationtoCSE and csiActive shall not be present when O-CSI is sent to VLR/GMSC.
1002 -- They may only be included in ATSI/ATM Ack message.
1003
1004 O-BcsmCamelTDPDataList ::= SEQUENCE SIZE (1..maxNumOfCamelTDPData) OF
1005     O-BcsmCamelTDPData
1006 -- O-BcsmCamelTDPDataList shall not contain more than one instance of
1007 -- O-BcsmCamelTDPData containing the same value for o-BcsmTriggerDetectionPoint.
1008 -- For CAMEL Phase 2, this means that only one instance of O-BcsmCamelTDPData is allowed
1009 -- with o-BcsmTriggerDetectionPoint being equal to DP2.
1010
1011 maxNumOfCamelTDPData INTEGER ::= 10
1012
1013 O-BcsmCamelTDPData ::= SEQUENCE {
1014     o-BcsmTriggerDetectionPoint O-BcsmTriggerDetectionPoint,
1015     serviceKey                  ServiceKey,
1016     gsmSCF-Address              [0] ISDN-AddressString,
1017     defaultCallHandling         [1] DefaultCallHandling,
1018     extensionContainer           [2] ExtensionContainer           OPTIONAL,
1019     ...
1020 }
1021
1022 ServiceKey ::= INTEGER (0..2147483647)
1023
1024 O-BcsmTriggerDetectionPoint ::= ENUMERATED {
1025     collectedInfo (2),
1026     ... ,
1027     routeSelectFailure (4) }
1028 -- exception handling:
1029 -- For O-BcsmCamelTDPData sequences containing this parameter with any
1030 -- other value than the ones listed the receiver shall ignore the whole
1031 -- O-BcsmCamelTDPDatasequence.
1032 -- For O-BcsmCamelTDP-Criteria sequences containing this parameter with any
1033 -- other value than the ones listed the receiver shall ignore the whole
1034 -- O-BcsmCamelTDP-Criteria sequence.
1035
1036 O-BcsmCamelTDPCriteriaList ::= SEQUENCE SIZE (1..maxNumOfCamelTDPData) OF
1037     O-BcsmCamelTDP-Criteria
1038
1039 T-BCSM-CAMEL-TDP-CriteriaList ::= SEQUENCE SIZE (1..maxNumOfCamelTDPData) OF
1040     T-BCSM-CAMEL-TDP-Criteria
1041

```

```

1042 O-BcsmCamelTDP-Criteria ::= SEQUENCE {
1043   o-BcsmTriggerDetectionPoint      O-BcsmTriggerDetectionPoint,
1044   destinationNumberCriteria        [0] DestinationNumberCriteria   OPTIONAL,
1045   basicServiceCriteria              [1] BasicServiceCriteria       OPTIONAL,
1046   callTypeCriteria                  [2] CallTypeCriteria           OPTIONAL,
1047   ... ,
1048   o-CauseValueCriteria              [3] O-CauseValueCriteria       OPTIONAL,
1049   extensionContainer                [4] ExtensionContainer         OPTIONAL }
1050
1051 T-BCSM-CAMEL-TDP-Criteria ::= SEQUENCE {
1052   t-BCSM-TriggerDetectionPoint      T-BcsmTriggerDetectionPoint,
1053   basicServiceCriteria              [0] BasicServiceCriteria       OPTIONAL,
1054   t-CauseValueCriteria              [1] T-CauseValueCriteria       OPTIONAL,
1055   ... }
1056
1057 DestinationNumberCriteria ::= SEQUENCE {
1058   matchType                         [0] MatchType,
1059   destinationNumberList              [1] DestinationNumberList     OPTIONAL,
1060   destinationNumberLengthList       [2] DestinationNumberLengthList OPTIONAL,
1061   -- one or both of destinationNumberList and destinationNumberLengthList
1062   -- shall be present
1063   ... }
1064
1065 DestinationNumberList ::= SEQUENCE SIZE (1..maxNumOfCamelDestinationNumbers) OF
1066   ISDN-AddressString
1067 -- The receiving entity shall not check the format of a number in
1068 -- the dialled number list
1069
1070 DestinationNumberLengthList ::= SEQUENCE SIZE (1..maxNumOfCamelDestinationNumberLengths) OF
1071   INTEGER(1..maxNumOfISDN-AddressDigits)
1072
1073 BasicServiceCriteria ::= SEQUENCE SIZE(1..maxNumOfCamelBasicServiceCriteria) OF
1074   Ext-BasicServiceCode
1075
1076 maxNumOfISDN-AddressDigits INTEGER ::= 15
1077
1078 maxNumOfCamelDestinationNumbers INTEGER ::= 10
1079
1080 maxNumOfCamelDestinationNumberLengths INTEGER ::= 3
1081
1082 maxNumOfCamelBasicServiceCriteria INTEGER ::= 5
1083
1084 CallTypeCriteria ::= ENUMERATED {
1085   forwarded (0),
1086   notForwarded (1)}
1087
1088 MatchType ::= ENUMERATED {
1089   inhibiting (0),
1090   enabling (1)}
1091
1092 O-CauseValueCriteria ::= SEQUENCE SIZE(1..maxNumOfCAMEL-O-CauseValueCriteria) OF
1093   CauseValue
1094
1095 T-CauseValueCriteria ::= SEQUENCE SIZE(1..maxNumOfCAMEL-T-CauseValueCriteria) OF
1096   CauseValue
1097
1098 maxNumOfCAMEL-O-CauseValueCriteria INTEGER ::= 5
1099
1100 maxNumOfCAMEL-T-CauseValueCriteria INTEGER ::= 5
1101
1102 CauseValue ::= OCTET STRING (SIZE(1))
1103 -- Type extracted from Cause parameter in ITU-T Recommendation Q.763.
1104 -- For the use of cause value refer to ITU-T Recommendation Q.850.
1105
1106
1107 DefaultCallHandling ::= ENUMERATED {
1108   continueCall (0) ,
1109   releaseCall (1) ,
1110   ...}
1111 -- exception handling:
1112 -- reception of values in range 2-31 shall be treated as "continueCall"
1113 -- reception of values greater than 31 shall be treated as "releaseCall"
1114

```

```

1115 CamelCapabilityHandling ::= INTEGER(1..16)
1116 -- value 1 = CAMEL phase 1,
1117 -- value 2 = CAMEL phase 2,
1118 -- value 3 = CAMEL Phase 3:
1119 -- reception of values greater than 3 shall be treated as CAMEL phase 3.
1120
1121 SupportedCamelPhases ::= BIT STRING {
1122     phase1 (0),
1123     phase2 (1) ,
1124     phase3 (2) } (SIZE (1..16))
1125 -- A node shall mark in the BIT STRING all CAMEL Phases it supports.
1126 -- Other values than listed above shall be discarded.
1127
1128 SMS-CSI ::= SEQUENCE {
1129     sms-CAMEL-TDP-DataList          [0] SMS-CAMEL-TDP-DataList,
1130     camelCapabilityHandling         [1] CamelCapabilityHandling
1131     ,
1132     extensionContainer              [2] ExtensionContainer          OPTIONAL,
1133     notificationToCSE               [3] NULL                      OPTIONAL,
1134     csiActive                       [4] NULL                      OPTIONAL,
1135     ...}
1136 -- notificationToCSE and csiActive shall not be present when SMS-CSI is sent to VLR/SGSN.
1137 -- They may only be included in ATSI/ATM Ack message.
1138
1139 SMS-CAMEL-TDP-DataList ::= SEQUENCE SIZE (1..maxNumOfCamelTDPData) OF
1140     SMS-CAMEL-TDP-Data
1141 -- SMS-CAMEL-TDP-DataList shall not contain more than one instance of
1142 -- SMS-CAMEL-TDP-Data containing the same value for sms-TriggerDetectionPoint.
1143
1144 SMS-CAMEL-TDP-Data ::= SEQUENCE {
1145     sms-TriggerDetectionPoint       [0] SMS-TriggerDetectionPoint,
1146     serviceKey                     [1] ServiceKey,
1147     gsmSCF-Address                 [2] ISDN-AddressString,
1148     defaultSMS-Handling            [3] DefaultSMS-Handling,
1149     extensionContainer              [4] ExtensionContainer          OPTIONAL,
1150     ...}
1151
1152 SMS-TriggerDetectionPoint ::= ENUMERATED {
1153     sms-CollectedInfo (1),
1154     ... }
1155 -- exception handling:
1156 -- For SMS-CAMEL-TDP-Data sequences containing this parameter with any
1157 -- other value than the ones listed the receiver shall ignore the whole
1158 -- SMS-CAMEL-TDP-Data sequence.
1159
1160 DefaultSMS-Handling ::= ENUMERATED {
1161     continueTransaction (0) ,
1162     releaseTransaction (1) ,
1163     ...}
1164 -- exception handling:
1165 -- reception of values in range 2-31 shall be treated as "continueTransaction"
1166 -- reception of values greater than 31 shall be treated as "releaseTransaction"
1167
1168 M-CSI ::= SEQUENCE {
1169     mobilityTriggers               MobilityTriggers,
1170     serviceKey                    ServiceKey,
1171     gsmSCF-Address                [0] ISDN-AddressString,
1172     extensionContainer             [1] ExtensionContainer          OPTIONAL,
1173     notificationToCSE             [2] NULL                      OPTIONAL,
1174     csiActive                     [3] NULL                      OPTIONAL,
1175     ...}
1176 -- notificationToCSE and csiActive shall not be present when M-CSI is sent to VLR.
1177 -- They may only be included in ATSI/ATM Ack message.
1178
1179 MobilityTriggers ::= SEQUENCE SIZE (1..maxNumOfMobilityTriggers) OF
1180     MM-Code
1181
1182 maxNumOfMobilityTriggers INTEGER ::= 10
1183

```



```

1184 MM-Code ::= OCTET STRING (SIZE (1))
1185 -- This type is used to indicate a Mobility Management event.
1186 -- Actions for the following M-Code values are defined in CAMEL Phase 3:
1187 --
1188 -- Location-update-in-same-VLR          MM-Code ::= '00000000'B
1189 -- Location-update-to-other-VLR        MM-Code ::= '00000001'B
1190 -- IMSI-Attach                          MM-Code ::= '00000010'B
1191 -- MS-initiated-IMSI-Detach             MM-Code ::= '00000011'B
1192 -- Network-initiated-IMSI-Detach       MM-Code ::= '00000100'B
1193 --
1194 -- If any other MM-code is received in M-CSI, then that MM-code shall be
1195 -- ignored.
1196
1197 T-CSI ::= SEQUENCE {
1198     t-BcsmCamelTDPDataList          T-BcsmCamelTDPDataList,
1199     extensionContainer                ExtensionContainer          OPTIONAL,
1200     ...,
1201     camelCapabilityHandling          [0] CamelCapabilityHandling  OPTIONAL,
1202     notificationToCSE                [1] NULL                    OPTIONAL,
1203     csi-Active                       [2] NULL                    OPTIONAL,
1204 }
1205 -- notificationToCSE and csi-Active shall not be present when T-CSI is sent to VLR/GMSC.
1206 -- They may only be included in ATSI/ATM Ack message.
1207
1208 T-BcsmCamelTDPDataList ::= SEQUENCE SIZE (1..maxNumOfCamelTDPData) OF
1209     T-BcsmCamelTDPData
1210 --- T-BcsmCamelTDPDataList shall not contain more than one instance of
1211 --- T-BcsmCamelTDPData containing the same value for t-BcsmTriggerDetectionPoint.
1212 --- For CAMEL Phase 2, this means that only one instance of T-BcsmCamelTDPData is allowed
1213 --- with t-BcsmTriggerDetectionPoint being equal to DP12.
1214 --- For CAMEL Phase 3, more TDP's are allowed.
1215
1216 T-BcsmCamelTDPData ::= SEQUENCE {
1217     t-BcsmTriggerDetectionPoint      T-BcsmTriggerDetectionPoint,
1218     serviceKey                       ServiceKey,
1219     gsmSCF-Address                   [0] ISDN-AddressString,
1220     defaultCallHandling               [1] DefaultCallHandling,
1221     extensionContainer                [2] ExtensionContainer      OPTIONAL,
1222     ...}
1223
1224 T-BcsmTriggerDetectionPoint ::= ENUMERATED {
1225     termAttemptAuthorized (12),
1226     ...,
1227     tBusy (13),
1228     tNoAnswer (14)}
1229 -- exception handling:
1230 -- For T-BcsmCamelTDPData sequences containing this parameter with any other
1231 -- value than the ones listed above, the receiver shall ignore the whole
1232 -- T-BcsmCamelTDPData sequence.
1233
1234 -- gprs location information retrieval types
1235
1236
1237 SendRoutingInfoForGprsArg ::= SEQUENCE {
1238     imsi                             [0] IMSI,
1239     ggsn-Address                      [1] GSN-Address          OPTIONAL,
1240     ggsn-Number                      [2] ISDN-AddressString,
1241     extensionContainer                [3] ExtensionContainer  OPTIONAL,
1242     ...}
1243
1244 SendRoutingInfoForGprsRes ::= SEQUENCE {
1245     sgsn-Address                     [0] GSN-Address,
1246     ggsn-Address                     [1] GSN-Address          OPTIONAL,
1247     mobileNotReachableReason         [2] AbsentSubscriberDiagnosticSM  OPTIONAL,
1248     extensionContainer                [3] ExtensionContainer  OPTIONAL,
1249     ...}
1250
1251 -- failure report types
1252
1253 FailureReportArg ::= SEQUENCE {
1254     imsi                             [0] IMSI,
1255     ggsn-Number                      [1] ISDN-AddressString  ,
1256     ggsn-Address                     [2] GSN-Address          OPTIONAL,
1257     extensionContainer                [3] ExtensionContainer  OPTIONAL,
1258     ...}
1259

```

```

1260 FailureReportRes ::= SEQUENCE {
1261     ggsn-Address          [0] GSN-Address          OPTIONAL,
1262     extensionContainer    [1] ExtensionContainer    OPTIONAL,
1263     ...}
1264
1265 -- gprs notification types
1266
1267 NoteMsPresentForGprsArg ::= SEQUENCE {
1268     imsi                  [0] IMSI,
1269     sgsn-Address          [1] GSN-Address,
1270     ggsn-Address          [2] GSN-Address          OPTIONAL,
1271     extensionContainer    [3] ExtensionContainer    OPTIONAL,
1272     ...}
1273
1274 NoteMsPresentForGprsRes ::= SEQUENCE {
1275     extensionContainer    [0] ExtensionContainer    OPTIONAL,
1276     ...}
1277
1278 -- fault recovery types
1279
1280
1281 ResetArg ::= SEQUENCE {
1282     hlr-Number            ISDN-AddressString,
1283     hlr-List              HLR-List                  OPTIONAL,
1284     ...}
1285
1286 RestoreDataArg ::= SEQUENCE {
1287     imsi                  IMSI,
1288     lmsi                  LMSI                      OPTIONAL,
1289     extensionContainer    ExtensionContainer        OPTIONAL,
1290     ... ,
1291     vlr-Capability        [6] VLR-Capability        OPTIONAL }
1292
1293 RestoreDataRes ::= SEQUENCE {
1294     hlr-Number            ISDN-AddressString,
1295     msNotReachable        NULL                      OPTIONAL,
1296     extensionContainer    ExtensionContainer        OPTIONAL,
1297     ...}
1298
1299 -- VBS/VGCS types
1300 VBSDDataList ::= SEQUENCE SIZE (1..maxNumOfVBSSGroupIds) OF
1301     VoiceBroadcastData
1302
1303 VGCSDataList ::= SEQUENCE SIZE (1..maxNumOfVGCSGroupIds) OF
1304     VoiceGroupCallData
1305
1306 maxNumOfVBSSGroupIds INTEGER ::= 50
1307
1308 maxNumOfVGCSGroupIds INTEGER ::= 50
1309
1310 VoiceGroupCallData ::= SEQUENCE {
1311     groupId                GroupId,
1312     extensionContainer      ExtensionContainer      OPTIONAL,
1313     ...}
1314
1315 VoiceBroadcastData ::= SEQUENCE {
1316     groupid                GroupId,
1317     broadcastInitEntitlement NULL                      OPTIONAL,
1318     extensionContainer      ExtensionContainer      OPTIONAL,
1319     ...}
1320
1321 GroupId ::= OCTET STRING (SIZE (3))
1322     -- Refers to the Group Identification as specified in GSM TS 03.03
1323     -- and 03.68/ 03.69
1324
1325 -- provide subscriber info types
1326
1327 ProvideSubscriberInfoArg ::= SEQUENCE {
1328     imsi                  [0] IMSI,
1329     lmsi                  [1] LMSI                  OPTIONAL,
1330     requestedInfo         [2] RequestedInfo,
1331     extensionContainer    [3] ExtensionContainer    OPTIONAL,
1332     ...}
1333

```

1334	ProvideSubscriberInfoRes ::= SEQUENCE {		
1335	subscriberInfo	SubscriberInfo,	
1336	extensionContainer	ExtensionContainer	OPTIONAL,
1337	...}		
1338			
1339	SubscriberInfo ::= SEQUENCE {		
1340	locationInformation	[0] LocationInformation	OPTIONAL,
1341	subscriberState	[1] SubscriberState	OPTIONAL,
1342	extensionContainer	[2] ExtensionContainer	OPTIONAL,
1343	...}		
1344			
1345	RequestedInfo ::= SEQUENCE {		
1346	locationInformation	[0] NULL	OPTIONAL,
1347	subscriberState	[1] NULL	OPTIONAL,
1348	extensionContainer	[2] ExtensionContainer	OPTIONAL,
1349	...}		
1350			
1351	LocationInformation ::= SEQUENCE {		
1352	ageOfLocationInformation	AgeOfLocationInformation	OPTIONAL,
1353	geographicalInformation	[0] GeographicalInformation	OPTIONAL,
1354	vlr-number	[1] ISDN-AddressString	OPTIONAL,
1355	locationNumber	[2] LocationNumber	OPTIONAL,
1356	cellIdOrLAI	[3] CellIdOrLAI	OPTIONAL,
1357	extensionContainer	[4] ExtensionContainer	OPTIONAL,
1358	... ,		
1359	selectedLSA-Id	[5] LSAIdentity	OPTIONAL,
1360	msc-Number	[6] ISDN-AddressString	OPTIONAL,
1361	geodeticInformation	[7] GeodeticInformation	OPTIONAL }
1362			
1363	GeographicalInformation ::= OCTET STRING (SIZE (8))		
1364	-- Refers to geographical Information defined in GSM 03.32.		
1365	-- Only the description of an ellipsoid point with uncertainty circle		
1366	-- as specified in GSM 03.32 is allowed to be used		
1367	-- The internal structure according to GSM 03.32 is as follows:		
1368	-- Type of shape (ellipsoid point with uncertainty circle)		1 octet
1369	-- Degrees of Latitude		3 octets
1370	-- Degrees of Longitude		3 octets
1371	-- Uncertainty code		1 octet
1372			
	GeodeticInformation ::= OCTET STRING (SIZE (10))		
	-- Refers to Calling Geodetic Location defined in Q.763 (1999).		
	-- Only the description of an ellipsoid point with uncertainty circle		
	-- as specified in Q.763 (1999) is allowed to be used		
	-- The internal structure according to Q.763 (1999) is as follows:		
	-- Screening and presentation indicators		1 octet
	-- Type of shape (ellipsoid point with uncertainty circle)		1 octet
	-- Degrees of Latitude		3 octets
	-- Degrees of Longitude		3 octets
	-- Uncertainty code		1 octet
	-- Confidence		1 octet
1373			
1374	LocationNumber ::= OCTET STRING (SIZE (2..10))		
1375	-- the internal structure is defined in CCITT Rec Q.763		
1376			
1377	SubscriberState ::= CHOICE {		
1378	assumedIdle	[0] NULL,	
1379	camelBusy [1] NULL,		
1380	netDetNotReachable	NotReachableReason,	
1381	notProvidedFromVLR	[2] NULL}	
1382			
1383	NotReachableReason ::= ENUMERATED {		
1384	msPurged (0),		
1385	imsiDetached (1),		
1386	restrictedArea (2),		
1387	notRegistered (3)}		
1388			
1389	-- any time interrogation info types		
1390			
1391	AnyTimeInterrogationArg ::= SEQUENCE {		
1392	subscriberIdentity	[0] SubscriberIdentity,	
1393	requestedInfo	[1] RequestedInfo,	
1394	gsmSCF-Address	[3] ISDN-AddressString,	
1395	extensionContainer	[2] ExtensionContainer	OPTIONAL,
1396	...}		
1397			

1398	AnyTimeInterrogationRes ::= SEQUENCE {		
1399	subscriberInfo	SubscriberInfo,	
1400	extensionContainer	ExtensionContainer	OPTIONAL,
1401	...}		
1402			
1403			
1404	<i>-- any time information handling types</i>		
1405			
1406	AnyTimeSubscriptionInterrogationArg ::= SEQUENCE {		
1407	subscriberIdentity	[0] SubscriberIdentity,	
1408	requestedSubscriptionInfo	[1] RequestedSubscriptionInfo,	
1409	gsmSCF-Address	[2] ISDN-AddressString,	
1410	extensionContainer	[3] ExtensionContainer	OPTIONAL,
1411	...}		
1412			
1413	AnyTimeSubscriptionInterrogationRes ::= SEQUENCE {		
1414	callForwardingData	[1] CallForwardingData	OPTIONAL,
1415	callBarringData	[2] CallBarringData	OPTIONAL,
1416	odb-Info	[3] ODB-Info	OPTIONAL,
1417	camel-SubscriptionInfo	[4] CAMEL-SubscriptionInfo	OPTIONAL,
1418	supportedVLR-CAMEL-Phases	[5] SupportedCamelPhases	OPTIONAL,
1419	supportedSGSN-CAMEL-Phases	[6] SupportedCamelPhases	OPTIONAL,
1420	extensionContainer	[7] ExtensionContainer	OPTIONAL,
1421	...}		
1422			
1423	RequestedSubscriptionInfo ::= SEQUENCE {		
1424	requestedSS-Info	[1] SS-ForBS-Code	OPTIONAL,
1425	odb	[2] NULL	OPTIONAL,
1426	requestedCAMEL-SubscriptionInfo	[3] RequestedCAMEL-SubscriptionInfo	OPTIONAL,
1427	supportedVLR-CAMEL-Phases	[4] NULL	OPTIONAL,
1428	supportedSGSN-CAMEL-Phases	[5] NULL	OPTIONAL,
1429	extensionContainer	[6] ExtensionContainer	OPTIONAL,
1430	...}		
1431			
1432	RequestedCAMEL-SubscriptionInfo ::= ENUMERATED {		
1433	o-CSI	(0),	
1434	t-CSI	(1),	
1435	vt-CSI	(2),	
1436	tif-CSI	(3),	
1437	gprs-CSI	(4),	
1438	sms-CSI	(5),	
1439	ss-CSI	(6),	
1440	m-CSI	(7),	
1441	d-csi	(8)}	
1442			
1443	CallForwardingData ::= SEQUENCE {		
1444	forwardingFeatureList	Ext-ForwFeatureList,	
1445	notificationToCSE	NULL	OPTIONAL,
1446	extensionContainer	[0] ExtensionContainer	OPTIONAL,
1447	...}		
1448			
1449	CallBarringData ::= SEQUENCE {		
1450	callBarringFeatureList	Ext-CallBarFeatureList,	
1451	password	Password,	
1452	wrongPasswordAttemptsCounter	WrongPasswordAttemptsCounter,	
1453	notificationToCSE	NULL	OPTIONAL,
1454	extensionContainer	ExtensionContainer	OPTIONAL,
1455	...}		
1456			
1457	WrongPasswordAttemptsCounter ::= INTEGER (0..4)		
1458			
1459	ODB-Info ::= SEQUENCE {		
1460	odb-Data	ODB-Data,	
1461	notificationToCSE	NULL	OPTIONAL,
1462	extensionContainer	ExtensionContainer	OPTIONAL,
1463	...}		
1464			

```

1465 CAMEL-SubscriptionInfo ::= SEQUENCE {
1466   o-CSI [0] O-CSI OPTIONAL,
1467   o-BcsmCamelTDP-CriteriaList [1] O-BcsmCamelTDPCriteriaList OPTIONAL,
1468   t-CSI [2] T-CSI OPTIONAL,
1469   t-BCSM-CAMEL-TDP-CriteriaList [3] T-BCSM-CAMEL-TDP-CriteriaList OPTIONAL,
1470   vt-CSI [4] T-CSI OPTIONAL,
1471   vt-BCSM-CAMEL-TDP-CriteriaList [5] T-BCSM-CAMEL-TDP-CriteriaList OPTIONAL,
1472   tif-CSI [6] NULL OPTIONAL,
1473   tif-CSI-NotificationToCSE [7] NULL OPTIONAL,
1474   gprs-CSI [8] GPRS-CSI OPTIONAL,
1475   sms-CSI [9] SMS-CSI OPTIONAL,
1476   ss-CSI [10] SS-CSI OPTIONAL,
1477   m-CSI [11] M-CSI OPTIONAL,
1478   extensionContainer [12] ExtensionContainer OPTIONAL,
1479   ...}
1480
1481 AnyTimeModificationArg ::= SEQUENCE {
1482   subscriberIdentity [0] SubscriberIdentity,
1483   gsmSCF-Address [1] ISDN-AddressString,
1484   modificationRequestFor-SS-Info [2] ModificationRequestFor-SS-Info OPTIONAL,
1485   modificationRequestFor-CSI [3] ModificationRequestFor-CSI OPTIONAL,
1486   extensionContainer [4] ExtensionContainer OPTIONAL,
1487   ...}
1488
1489 AnyTimeModificationRes ::= SEQUENCE {
1490   ss-InfoFor-CSE [0] Ext-SS-InfoFor-CSE OPTIONAL,
1491   camel-SubscriptionInfo [1] CAMEL-SubscriptionInfo OPTIONAL,
1492   extensionContainer [2] ExtensionContainer OPTIONAL,
1493   ...}
1494
1495 ModificationRequestFor-SS-Info ::= SEQUENCE {
1496   ss-Code [0] SS-Code,
1497   basicService [1] Ext-BasicServiceCode OPTIONAL,
1498   ss-Status [2] Ext-SS-Status OPTIONAL,
1499   forwardedToNumber [3] AddressString OPTIONAL,
1500   forwardedToSubaddress [4] ISDN-SubaddressString OPTIONAL,
1501   noReplyConditionTime [5] Ext-NoRepCondTime OPTIONAL,
1502   modifyNotificationToCSE [6] ModificationInstruction OPTIONAL,
1503   extensionContainer [7] ExtensionContainer OPTIONAL,
1504   ...}
1505
1506 ModificationRequestFor-CSI ::= SEQUENCE {
1507   requestedCamelSubscriptionInfo [0] RequestedCAMEL-SubscriptionInfo OPTIONAL,
1508   modifyNotificationToCSE [1] ModificationInstruction OPTIONAL,
1509   modifyCSI-State [2] ModificationInstruction OPTIONAL,
1510   extensionContainer [3] ExtensionContainer OPTIONAL,
1511   ...}
1512
1513 ModificationInstruction ::= ENUMERATED {
1514   deactivate (0),
1515   activate (1)}
1516
1517 -- subscriber data modification notification types
1518
1519 NoteSubscriberDataModifiedArg ::= SEQUENCE {
1520   imsi IMSI,
1521   msisdn ISDN-AddressString,
1522   typeOfModification TypeOfModification,
1523   extensionContainer ExtensionContainer OPTIONAL,
1524   ...}
1525
1526 NoteSubscriberDataModifiedRes ::= SEQUENCE {
1527   extensionContainer ExtensionContainer OPTIONAL,
1528   ...}
1529
1530 TypeOfModification ::= ENUMERATED {
1531   callForwardingSS-Data (0),
1532   callBarringSS-Data (1),
1533   operatorDeterminedBarringData (2),
1534   camelSubscriptionInformation (3),
1535   ...}
1536 -- exception handling:
1537 -- reception of other values shall be treated as unexpected data
1538
1539
1540 -- mobility management event notification info types
1541

```

```

1542 NoteMM-EventArg ::= SEQUENCE {
1543     serviceKey                ServiceKey,
1544     eventMet                   [0] MM-Code,
1545     imsi                       [1] IMSI,
1546     msisdn                     [2] ISDN-AddressString,
1547     locationInformation        [3] LocationInformation OPTIONAL,
1548     lsaIdentity                [4] LSAIdentity OPTIONAL,
1549     supportedCAMELPhases       [5] SupportedCamelPhases OPTIONAL,
1550     extensionContainer         [6] ExtensionContainer OPTIONAL,
1551     ...}
1552
1553 NoteMM-EventRes ::= SEQUENCE {
1554     extensionContainer         ExtensionContainer OPTIONAL,
1555     ...}
1556
1557 Ext-SS-InfoFor-CSE ::= CHOICE {
1558     forwardingInfoFor-CSE     [0] Ext-ForwardingInfoFor-CSE,
1559     callBarringInfoFor-CSE    [1] Ext-CallBarringInfoFor-CSE
1560 }
1561
1562 Ext-ForwardingInfoFor-CSE ::= SEQUENCE {
1563     ss-Code                    [0] SS-Code,
1564     forwardingFeatureList      [1] Ext-ForwFeatureList,
1565     notificationToCSE          [2] NULL,
1566     extensionContainer         [3] ExtensionContainer OPTIONAL,
1567     ...}
1568
1569 Ext-CallBarringInfoFor-CSE ::= SEQUENCE {
1570     ss-Code                    [0] SS-Code,
1571     callBarringFeatureList     [1] Ext-CallBarFeatureList,
1572     password                   [2] Password,
1573     wrongPasswordAttemptsCounter [3] WrongPasswordAttemptsCounter,
1574     notificationToCSE          [4] NULL,
1575     extensionContainer         [5] ExtensionContainer OPTIONAL,
1576     ...}
1577
1578 END

```

17.7.2 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) version6 (6)}
4
5  DEFINITIONS
6
7  IMPLICIT TAGS
8
9  ::=
10
11  BEGIN
12
13  EXPORTS
14     ActivateTraceModeArg,
15     ActivateTraceModeRes,
16     DeactivateTraceModeArg,
17     DeactivateTraceModeRes
18  ;
19
20  IMPORTS
21     AddressString,
22     IMSI
23  FROM MAP-CommonDataTypes {
24     ccitt identified-organization (4) etsi (0) mobileDomain (0)
25     gsm-Network (1) modules (3) map-CommonDataTypes (18) version6 (6)}
26
27     ExtensionContainer
28  FROM MAP-ExtensionDataTypes {
29     ccitt identified-organization (4) etsi (0) mobileDomain (0)
30     gsm-Network (1) modules (3) map-ExtensionDataTypes (21) version6 (6)}
31
32
33  ;
34
35

```

```

36 ActivateTraceModeArg ::= SEQUENCE {
37     imsi [0] IMSI OPTIONAL,
38     traceReference [1] TraceReference,
39     traceType [2] TraceType,
40     omc-Id [3] AddressString OPTIONAL,
41     extensionContainer [4] ExtensionContainer OPTIONAL,
42     ...}
43
44 TraceReference ::= OCTET STRING (SIZE (1..2))
45
46 TraceType ::= INTEGER
47     (0..255)
48     -- Trace types are fully defined in TS GSM 12.08.
49
50 ActivateTraceModeRes ::= SEQUENCE {
51     extensionContainer [0] ExtensionContainer OPTIONAL,
52     ...}
53
54 DeactivateTraceModeArg ::= SEQUENCE {
55     imsi [0] IMSI OPTIONAL,
56     traceReference [1] TraceReference,
57     extensionContainer [2] ExtensionContainer OPTIONAL,
58     ...}
59
60 DeactivateTraceModeRes ::= SEQUENCE {
61     extensionContainer [0] ExtensionContainer OPTIONAL,
62     ...}
63
64 END

```

17.7.3 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) version6 (6)}
4
5 DEFINITIONS
6
7 IMPLICIT TAGS
8
9 ::=
10
11 BEGIN
12
13 EXPORTS
14     SendRoutingInfoArg,
15     SendRoutingInfoRes,
16     ProvideRoamingNumberArg,
17     ProvideRoamingNumberRes,
18     ResumeCallHandlingArg,
19     ResumeCallHandlingRes,
20     NumberOfForwarding,
21     SuppressionOfAnnouncement,
22     CallReferenceNumber,
23     ProvideSIWFSNumberArg,
24     ProvideSIWFSNumberRes,
25     SIWFSSignallingModifyArg,
26     SIWFSSignallingModifyRes,
27     SetReportingStateArg,
28     SetReportingStateRes,
29     StatusReportArg,
30     StatusReportRes,
31     RemoteUserFreeArg,
32     RemoteUserFreeRes,
33     IST-AlertArg,
34     IST-AlertRes,
35     IST-CommandArg,
36     IST-CommandRes
37 ;
38
39 IMPORTS
40     SubscriberInfo,
41     SupportedCamelPhases,
42     CUG-Interlock,
43     O-CSI,
44     D-CSI,
45     O-BcsmCamelTDPCriteriaList,

```

```

46 T-BCSM-CAMEL-TDP-CriteriaList,
47 IST-SupportIndicator,
48 IST-AlertTimerValue,
49 T-CSI
50
51
52 FROM MAP-MS-DataTypes {
53   ccitt identified-organization (4) etsi (0) mobileDomain (0)
54   gsm-Network (1) modules (3) map-MS-DataTypes (11) version6 (6)}
55
56   ForwardingOptions,
57   SS-List,
58   CCBS-Feature
59 FROM MAP-SS-DataTypes {
60   ccitt identified-organization (4) etsi (0) mobileDomain (0)
61   gsm-Network (1) modules (3) map-SS-DataTypes (14) version6 (6)}
62
63   ISDN-AddressString,
64   ISDN-SubaddressString,
65   ExternalSignalInfo,
66   Ext-ExternalSignalInfo,
67   IMSI,
68   LMSI,
69   Ext-BasicServiceCode,
70   AlertingPattern,
71   NAEA-PreferredCI
72
73
74 FROM MAP-CommonDataTypes {
75   ccitt identified-organization (4) etsi (0) mobileDomain (0)
76   gsm-Network (1) modules (3) map-CommonDataTypes (18) version6 (6)}
77
78   ExtensionContainer
79 FROM MAP-ExtensionDataTypes {
80   ccitt identified-organization (4) etsi (0) mobileDomain (0)
81   gsm-Network (1) modules (3) map-ExtensionDataTypes (21) version6 (6)}
82 ;
83
84

```

CUG-CheckInfo ::= SEQUENCE {			
cug-Interlock	CUG-Interlock,		
cug-OutgoingAccess	NULL		OPTIONAL,
extensionContainer	ExtensionContainer		OPTIONAL,
...			

NumberOfForwarding ::= INTEGER (1..5)
--

SendRoutingInfoArg ::= SEQUENCE {			
msisdn	[0] ISDN-AddressString,		
cug-CheckInfo	[1] CUG-CheckInfo		OPTIONAL,
numberOfForwarding	[2] NumberOfForwarding		OPTIONAL,
interrogationType	[3] InterrogationType,		
or-Interrogation	[4] NULL		OPTIONAL,
or-Capability	[5] OR-Phase		OPTIONAL,
gmsc-Address	[6] ISDN-AddressString,		
callReferenceNumber	[7] CallReferenceNumber		OPTIONAL,
forwardingReason	[8] ForwardingReason		OPTIONAL,
basicServiceGroup	[9] Ext-BasicServiceCode		OPTIONAL,
networkSignalInfo	[10] ExternalSignalInfo		OPTIONAL,
camelInfo	[11] CamelInfo		OPTIONAL,
suppressionOfAnnouncement	[12] SuppressionOfAnnouncement		OPTIONAL,
extensionContainer	[13] ExtensionContainer		OPTIONAL,
...			
alertingPattern	[14] AlertingPattern		OPTIONAL,
ccbs-Call	[15] NULL		OPTIONAL,
supportedCCBS-Phase	[16] SupportedCCBS-Phase		OPTIONAL,
additionalSignalInfo	[17] Ext-ExternalSignalInfo		OPTIONAL,
istSupportIndicator	[18] IST-SupportIndicator		OPTIONAL,
pre-pagingSupported	[19] NULL		OPTIONAL,
callDiversionTreatmentIndicator	[20] CallDiversionTreatmentIndicator		OPTIONAL }

SuppressionOfAnnouncement ::= NULL

InterrogationType ::= ENUMERATED {	
basicCall (0),	
forwarding (1)}	

OR-Phase ::= INTEGER (1..127)

```

124
125 CallReferenceNumber ::= OCTET STRING (SIZE (1..8))
126
127 ForwardingReason ::= ENUMERATED {
128     notReachable (0),
129     busy (1),
130     noReply (2)}
131
132 SupportedCCBS-Phase ::= INTEGER (1..127)
133 -- exception handling:
134 -- Only value 1 is used.
135 -- Values in the ranges 2-127 are reserved for future use.
136 -- If received values 2-127 shall be mapped on to value 1.
137
138 CallDiversionTreatmentIndicator ::= OCTET STRING (SIZE(1))
139 -- callDiversionAllowed (xxxx xx01)
140 -- callDiversionNotAllowed (xxxx xx10)
141 -- network default is call diversion allowed
142
143 SendRoutingInfoRes ::= [3] SEQUENCE {
144     imsi [9] IMSI OPTIONAL,
145     -- IMSI must be present if SendRoutingInfoRes is not segmented.
146     -- If the TC-Result-NL segmentation option is taken the IMSI must be
147     -- present in one segmented transmission of SendRoutingInfoRes.
148     extendedRoutingInfo ExtendedRoutingInfo OPTIONAL,
149     cug-CheckInfo [3] CUG-CheckInfo OPTIONAL,
150     cugSubscriptionFlag [6] NULL OPTIONAL,
151     subscriberInfo [7] SubscriberInfo OPTIONAL,
152     ss-List [1] SS-List OPTIONAL,
153     basicService [5] Ext-BasicServiceCode OPTIONAL,
154     forwardingInterrogationRequired [4] NULL OPTIONAL,
155     vmsc-Address [2] ISDN-AddressString OPTIONAL,
156     extensionContainer [0] ExtensionContainer OPTIONAL,
157     ... ,
158     naea-PreferredCI [10] NAEA-PreferredCI OPTIONAL,
159     -- naea-PreferredCI is included at the discretion of the HLR operator.
160     ccbs-Indicators [11] CCBS-Indicators OPTIONAL,
161     msisdn [12] ISDN-AddressString OPTIONAL,
162     numberPortabilityStatus [13] NumberPortabilityStatus OPTIONAL,
163     istAlertTimer [14] IST-AlertTimerValue OPTIONAL
164 }
165
166 NumberPortabilityStatus ::= ENUMERATED {
167     notKnownToBePorted (0),
168     ownNumberPortedOut (1),
169     foreignNumberPortedToForeignNetwork (2),
170     ...}
171 -- exception handling:
172 -- reception of other values than the ones listed the receiver shall ignore the
173 -- whole NumberPortabilityStatus
174
175 CCBS-Indicators ::= SEQUENCE {
176     ccbs-Possible [0] NULL OPTIONAL,
177     keepCCBS-CallIndicator [1] NULL OPTIONAL,
178     extensionContainer [2] ExtensionContainer OPTIONAL,
179     ...}
180
181 RoutingInfo ::= CHOICE {
182     roamingNumber ISDN-AddressString,
183     forwardingData ForwardingData}
184
185 ForwardingData ::= SEQUENCE {
186     forwardedToNumber [5] ISDN-AddressString OPTIONAL,
187     -- When this datatype is sent from an HLR which supports CAMEL Phase 2
188     -- to a GMSC which supports CAMEL Phase 2 the GMSC shall not check the
189     -- format of the number
190     forwardedToSubaddress [4] ISDN-SubaddressString OPTIONAL,
191     forwardingOptions [6] ForwardingOptions OPTIONAL,
192     extensionContainer [7] ExtensionContainer OPTIONAL,
193     ...}
194

```

195	ProvideRoamingNumberArg ::= SEQUENCE {		
196	imsi	[0] IMSI,	
197	msc-Number	[1] ISDN-AddressString,	
198	msisdn	[2] ISDN-AddressString	OPTIONAL,
199	lmsi	[4] LMSI	OPTIONAL,
200	gsm-BearerCapability	[5] ExternalSignalInfo	OPTIONAL,
201	networkSignalInfo	[6] ExternalSignalInfo	OPTIONAL,
202	suppressionOfAnnouncement	[7] SuppressionOfAnnouncement	OPTIONAL,
203	gmsc-Address	[8] ISDN-AddressString	OPTIONAL,
204	callReferenceNumber	[9] CallReferenceNumber	OPTIONAL,
205	or-Interrogation	[10] NULL	OPTIONAL,
206	extensionContainer	[11] ExtensionContainer	OPTIONAL,
207	...		
208	alertingPattern	[12] AlertingPattern	OPTIONAL,
209	ccbs-Call	[13] NULL	OPTIONAL,
210	supportedCamelPhasesInGMSC	[15] SupportedCamelPhases	OPTIONAL,
211	additionalSignalInfo	[14] Ext-ExternalSignalInfo	OPTIONAL,
212	orNotSupportedInGMSC	[16] NULL	OPTIONAL,
213	pre-pagingSupported	[17] NULL	OPTIONAL }
214			
215	ProvideRoamingNumberRes ::= SEQUENCE {		
216	roamingNumber	ISDN-AddressString,	
217	extensionContainer	ExtensionContainer	OPTIONAL,
218	...		
219			
220	ResumeCallHandlingArg ::= SEQUENCE {		
221	callReferenceNumber	[0] CallReferenceNumber	OPTIONAL,
222	basicServiceGroup	[1] Ext-BasicServiceCode	OPTIONAL,
223	forwardingData	[2] ForwardingData	OPTIONAL,
224	imsi	[3] IMSI	OPTIONAL,
225	cug-CheckInfo	[4] CUG-CheckInfo	OPTIONAL,
226	o-CSI	[5] O-CSI	OPTIONAL,
227	extensionContainer	[7] ExtensionContainer	OPTIONAL,
228	ccbs-Possible	[8] NULL	OPTIONAL,
229	msisdn	[9] ISDN-AddressString	OPTIONAL,
230	uu-Data	[10] UU-Data	OPTIONAL,
231	allInformationSent	[11] NULL	OPTIONAL,
232	d-csi	[12] D-CSI	OPTIONAL,
233	...		
234			
235	UU-Data ::= SEQUENCE {		
236	uuIndicator	[0] UUIndicator	OPTIONAL,
237	uui	[1] UUI	OPTIONAL,
238	uusCFInteraction	[2] NULL	OPTIONAL,
239	extensionContainer	[3] ExtensionContainer	OPTIONAL,
240	...		
241			
242	UUIndicator ::= OCTET STRING (SIZE (1))		
243	-- Octets are coded according to ETS 300 356		
244			
245	UUI ::= OCTET STRING (SIZE (1..131))		
246	-- Octets are coded according to ETS 300 356		
247			
248	ResumeCallHandlingRes ::= SEQUENCE {		
249	extensionContainer	ExtensionContainer	OPTIONAL,
250	...		
251			
252	CamelInfo ::= SEQUENCE {		
253	supportedCamelPhases	SupportedCamelPhases,	
254	suppress-T-CSI	NULL	OPTIONAL,
255	extensionContainer	ExtensionContainer	OPTIONAL,
256	...		
257			
258	ExtendedRoutingInfo ::= CHOICE {		
259	routingInfo	RoutingInfo,	
260	camelRoutingInfo	[8] CamelRoutingInfo}	
261			
262	CamelRoutingInfo ::= SEQUENCE {		
263	forwardingData	ForwardingData	OPTIONAL,
264	gmscCamelSubscriptionInfo	[0] GmscCamelSubscriptionInfo,	
265	extensionContainer	[1] ExtensionContainer	OPTIONAL,
266	...		
267			

```

268 GmscCamelSubscriptionInfo ::= SEQUENCE {
269     t-CSI [0] T-CSI OPTIONAL,
270     o-CSI [1] O-CSI OPTIONAL,
271     extensionContainer [2] ExtensionContainer OPTIONAL,
272     ... ,
273     o-BcsmCamelTDP-CriteriaList [3] O-BcsmCamelTDPCriteriaList OPTIONAL,
274     t-BCSM-CAMEL-TDP-CriteriaList [4] T-BCSM-CAMEL-TDP-CriteriaList OPTIONAL,
275     d-csi [5] D-CSI OPTIONAL
276 }
277 ProvideSIWFSNumberArg ::= SEQUENCE {
278     gsm-BearerCapability [0] ExternalSignalInfo,
279     isdn-BearerCapability [1] ExternalSignalInfo,
280     call-Direction [2] CallDirection,
281     b-Subscriber-Address [3] ISDN-AddressString,
282     chosenChannel [4] ExternalSignalInfo,
283     lowerLayerCompatibility [5] ExternalSignalInfo OPTIONAL,
284     highLayerCompatibility [6] ExternalSignalInfo OPTIONAL,
285     extensionContainer [7] ExtensionContainer OPTIONAL,
286     ...}
287
288 CallDirection ::= OCTET STRING (SIZE (1))
289 -- OCTET 1
290
291 -- bit 1 (direction of call)
292 -- 0 Mobile Originated Call (MOC)
293 -- 1 Mobile Terminated Call (MTC)
294
295
296 ProvideSIWFSNumberRes ::= SEQUENCE {
297     siWFSNumber [0] ISDN-AddressString,
298     extensionContainer [1] ExtensionContainer OPTIONAL,
299     ...}
300
301 SIWFSsignallingModifyArg ::= SEQUENCE {
302     channelType [0] ExternalSignalInfo OPTIONAL,
303     chosenChannel [1] ExternalSignalInfo OPTIONAL,
304     extensionContainer [2] ExtensionContainer OPTIONAL,
305     ...}
306
307 SIWFSsignallingModifyRes ::= SEQUENCE {
308     chosenChannel [0] ExternalSignalInfo OPTIONAL,
309     extensionContainer [1] ExtensionContainer OPTIONAL,
310     ...}
311
312 SetReportingStateArg ::= SEQUENCE {
313     imsi [0] IMSI OPTIONAL,
314     lmsi [1] LMSI OPTIONAL,
315     ccbs-Monitoring [2] ReportingState OPTIONAL,
316     extensionContainer [3] ExtensionContainer OPTIONAL,
317     ...}
318
319 ReportingState ::= ENUMERATED {
320     stopMonitoring (0),
321     startMonitoring (1),
322     ...}
323 -- exception handling:
324 -- reception of values 2-10 shall be mapped to 'stopMonitoring'
325 -- reception of values > 10 shall be mapped to 'startMonitoring'
326
327 SetReportingStateRes ::= SEQUENCE{
328     ccbs-SubscriberStatus [0] CCBS-SubscriberStatus OPTIONAL,
329     extensionContainer [1] ExtensionContainer OPTIONAL,
330     ...}
331
332 CCBS-SubscriberStatus ::= ENUMERATED {
333     ccbsNotIdle (0),
334     ccbsIdle (1),
335     ccbsNotReachable (2),
336     ...}
337 -- exception handling:
338 -- reception of values 3-10 shall be mapped to 'ccbsNotIdle'
339 -- reception of values 11-20 shall be mapped to 'ccbsIdle'
340 -- reception of values > 20 shall be mapped to 'ccbsNotReachable'
341

```

```

342 StatusReportArg ::= SEQUENCE{
343     imsi                [0] IMSI,
344     eventReportData    [1] EventReportData    OPTIONAL,
345     callReportData     [2] CallReportData     OPTIONAL,
346     extensionContainer [3] ExtensionContainer  OPTIONAL,
347     ...}
348
349 EventReportData ::= SEQUENCE{
350     ccbs-SubscriberStatus [0] CCBS-SubscriberStatus    OPTIONAL,
351     extensionContainer    [1] ExtensionContainer    OPTIONAL,
352     ...}
353
354 CallReportData ::= SEQUENCE{
355     monitoringMode      [0] MonitoringMode    OPTIONAL,
356     callOutcome        [1] CallOutcome        OPTIONAL,
357     extensionContainer [2] ExtensionContainer  OPTIONAL,
358     ...}
359
360 MonitoringMode ::= ENUMERATED {
361     a-side              (0),
362     b-side              (1),
363     ...}
364 -- exception handling:
365 -- reception of values 2-10 shall be mapped 'a-side'
366 -- reception of values > 10 shall be mapped to 'b-side'
367
368 CallOutcome ::= ENUMERATED {
369     success              (0),
370     failure              (1),
371     busy                 (2),
372     ...}
373 -- exception handling:
374 -- reception of values 3-10 shall be mapped to 'success'
375 -- reception of values 11-20 shall be mapped to 'failure'
376 -- reception of values > 20 shall be mapped to 'busy'
377
378 StatusReportRes ::= SEQUENCE {
379     extensionContainer [0] ExtensionContainer    OPTIONAL,
380     ...}
381
382 RemoteUserFreeArg ::= SEQUENCE{
383     imsi                [0] IMSI,
384     callInfo            [1] ExternalSignalInfo,
385     ccbs-Feature        [2] CCBS-Feature,
386     translatedB-Number [3] ISDN-AddressString,
387     replaceB-Number    [4] NULL                OPTIONAL,
388     alertingPattern    [5] AlertingPattern    OPTIONAL,
389     extensionContainer [6] ExtensionContainer  OPTIONAL,
390     ...}
391
392 RemoteUserFreeRes ::= SEQUENCE{
393     ruf-Outcome         [0] RUF-Outcome,
394     extensionContainer [1] ExtensionContainer    OPTIONAL,
395     ...}
396
397 RUF-Outcome ::= ENUMERATED{
398     accepted (0),
399     rejected (1),
400     noResponseFromFreeMS (2), -- T4 Expiry
401     noResponseFromBusyMS (3), -- T10 Expiry
402     udubFromFreeMS (4),
403     udubFromBusyMS (5),
404     ...}
405 -- exception handling:
406 -- reception of values 6-20 shall be mapped to 'accepted'
407 -- reception of values 21-30 shall be mapped to 'rejected'
408 -- reception of values 31-40 shall be mapped to 'noResponseFromFreeMS'
409 -- reception of values 41-50 shall be mapped to 'noResponseFromBusyMS'
410 -- reception of values 51-60 shall be mapped to 'udubFromFreeMS'
411 -- reception of values > 60 shall be mapped to 'udubFromBusyMS'
412
413 IST-AlertArg ::= SEQUENCE{
414     imsi                [0] IMSI,
415     extensionContainer [1] ExtensionContainer    OPTIONAL,
416     ...}
417

```

```

418 IST-AlertRes ::= SEQUENCE{
419     istAlertTimer                [0] IST-AlertTimerValue      OPTIONAL,
420     istInformationWithdraw        [1] NULL                        OPTIONAL,
421     callTerminationIndicator     [2] CallTerminationIndicator  OPTIONAL,
422     extensionContainer            [3] ExtensionContainer         OPTIONAL,
423     ...}
424
425 IST-CommandArg ::= SEQUENCE{
426     imsi                          [0] IMSI,
427     extensionContainer            [1] ExtensionContainer         OPTIONAL,
428     ...}
429
430 IST-CommandRes ::= SEQUENCE{
431     extensionContainer            ExtensionContainer              OPTIONAL,
432     ...}
433
434 CallTerminationIndicator ::= ENUMERATED {
435     terminateCallActivityReferred (0),
436     terminateAllCallActivities    (1),
437     ...}
438 -- exception handling:
439 -- reception of values 2-10 shall be mapped to ' terminateCallActivityReferred '
440 -- reception of values > 10 shall be mapped to ' terminateAllCallActivities '
441
442 -- In MSCs not supporting linkage of all call activities, any value received shall
443 -- be interpreted as ' terminateCallActivityReferred '
444
445 END

```

17.7.4 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) version6 (6)}
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     SS-SubscriptionOption,
18     SS-ForBS-Code,
19     InterrogateSS-Res,
20     USSD-Arg,
21     USSD-Res,
22     USSD-DataCodingScheme,
23     USSD-String,
24     Password,
25     GuidanceInfo,
26     SS-List,
27     SS-InfoList,
28     OverrideCategory,
29     CliRestrictionOption,
30     NoReplyConditionTime,
31     ForwardingOptions,
32     maxNumOfSS,
33     SS-Data,
34     SS-InvocationNotificationArg,
35     SS-InvocationNotificationRes,
36     CCBS-Feature,
37     RegisterCC-EntryArg,
38     RegisterCC-EntryRes,
39     EraseCC-EntryArg,
40     EraseCC-EntryRes
41 ;
42
43 IMPORTS
44     AddressString,
45     ISDN-AddressString,
46     ISDN-SubaddressString,
47     IMSI,

```

```

48     BasicServiceCode,
49     AlertingPattern,
50     EMLPP-Priority,
51     ExternalSignalInfo
52
53 FROM MAP-CommonDataTypes {
54     ccitt-identified-organization (4) etsi (0) mobileDomain (0)
55     gsm-Network (1) modules (3) map-CommonDataTypes (18) version6 (6)}
56
57     ExtensionContainer
58 FROM MAP-ExtensionDataTypes {
59     ccitt-identified-organization (4) etsi (0) mobileDomain (0)
60     gsm-Network (1) modules (3) map-ExtensionDataTypes (21) version6 (6)}
61
62     SS-Code
63 FROM MAP-SS-Code {
64     ccitt-identified-organization (4) etsi (0) mobileDomain (0)
65     gsm-Network (1) modules (3) map-SS-Code (15) version6 (6)}
66 ;
67
68

```

RegisterSS-Arg ::= SEQUENCE{			
ss-Code	SS-Code,		
basicService	BasicServiceCode		OPTIONAL,
forwardedToNumber	[4] AddressString		OPTIONAL,
forwardedToSubaddress	[6] ISDN-SubaddressString		OPTIONAL,
noReplyConditionTime	[5] NoReplyConditionTime		OPTIONAL,
...			
defaultPriority	[7] EMLPP-Priority		OPTIONAL }

NoReplyConditionTime ::= INTEGER (5..30)

SS-Info ::= CHOICE {	
forwardingInfo	[0] ForwardingInfo,
callBarringInfo	[1] CallBarringInfo,
ss-Data	[3] SS-Data}

ForwardingInfo ::= SEQUENCE {		
ss-Code	SS-Code	OPTIONAL,
forwardingFeatureList	ForwardingFeatureList,	
...		

ForwardingFeatureList ::=	
SEQUENCE SIZE (1..maxNumOfBasicServiceGroups) OF	ForwardingFeature

ForwardingFeature ::= SEQUENCE {			
basicService	BasicServiceCode		OPTIONAL,
ss-Status [4] SS-Status			OPTIONAL,
forwardedToNumber	[5] ISDN-AddressString		OPTIONAL,
forwardedToSubaddress	[8] ISDN-SubaddressString		OPTIONAL,
forwardingOptions	[6] ForwardingOptions		OPTIONAL,
noReplyConditionTime	[7] NoReplyConditionTime		OPTIONAL,
...			

SS-Status ::= OCTET STRING (SIZE (1))	
-- bits 8765: 0000 (unused)	
-- bits 4321: Used to convey the "P bit", "R bit", "A bit" and "Q bit",	
-- representing supplementary service state information	
-- as defined in TS GSM 03.11	
-- bit 4: "Q bit"	
-- bit 3: "P bit"	
-- bit 2: "R bit"	
-- bit 1: "A bit"	

118	ForwardingOptions ::= OCTET STRING (SIZE (1))
119	
120	-- bit 8: notification to forwarding party
121	-- 0 no notification
122	-- 1 notification
123	
124	-- bit 7: redirecting presentation
125	-- 0 no presentation
126	-- 1 presentation
127	
128	-- bit 6: notification to calling party
129	-- 0 no notification
130	-- 1 notification
131	
132	-- bit 5: 0 (unused)
133	
134	-- bits 43: forwarding reason
135	-- 00 ms not reachable
136	-- 01 ms busy
137	-- 10 no reply
138	-- 11 unconditional when used in a SRI Result,
139	-- or call deflection when used in a RCH Argument
140	-- bits 21: 00 (unused)
141	
142	CallBarringInfo ::= SEQUENCE {
143	ss-Code SS-Code OPTIONAL,
144	callBarringFeatureList CallBarringFeatureList,
145	...}
146	
147	CallBarringFeatureList ::= SEQUENCE SIZE (1..maxNumOfBasicServiceGroups) OF
148	CallBarringFeature
149	
150	CallBarringFeature ::= SEQUENCE {
151	basicService BasicServiceCode OPTIONAL,
152	ss-Status [4] SS-Status OPTIONAL,
153	...}
154	
155	SS-Data ::= SEQUENCE {
156	ss-Code SS-Code OPTIONAL,
157	ss-Status [4] SS-Status OPTIONAL,
158	ss-SubscriptionOption SS-SubscriptionOption OPTIONAL,
159	basicServiceGroupList BasicServiceGroupList OPTIONAL,
160	...,
161	defaultPriority EMLPP-Priority OPTIONAL
162	}
163	
164	SS-SubscriptionOption ::= CHOICE {
165	cliRestrictionOption [2] CliRestrictionOption,
166	overrideCategory [1] OverrideCategory}
167	
168	CliRestrictionOption ::= ENUMERATED {
169	permanent (0),
170	temporaryDefaultRestricted (1),
171	temporaryDefaultAllowed (2)}
172	
173	OverrideCategory ::= ENUMERATED {
174	overrideEnabled (0),
175	overrideDisabled (1)}
176	
177	SS-ForBS-Code ::= SEQUENCE {
178	ss-Code SS-Code,
179	basicService BasicServiceCode OPTIONAL,
180	...}
181	
182	GenericServiceInfo ::= SEQUENCE {
183	ss-Status SS-Status,
184	cliRestrictionOption CliRestrictionOption OPTIONAL,
185	...,
186	maximumEntitledPriority [0] EMLPP-Priority OPTIONAL,
187	defaultPriority [1] EMLPP-Priority OPTIONAL,
188	ccbs-FeatureList [2] CCBS-FeatureList OPTIONAL }
189	
190	CCBS-FeatureList ::= SEQUENCE SIZE (1..maxNumOfCCBS-Requests) OF
191	CCBS-Feature
192	
193	maxNumOfCCBS-Requests INTEGER ::= 5
194	

195	CCBS-Feature ::= SEQUENCE {		
196	ccbs-Index	[0] CCBS-Index	OPTIONAL,
197	b-subscriberNumber	[1] ISDN-AddressString	OPTIONAL,
198	b-subscriberSubaddress	[2] ISDN-SubaddressString	OPTIONAL,
199	basicServiceGroup	[3] BasicServiceCode	OPTIONAL,
200	...}		
201			
202	CCBS-Index ::= INTEGER (1..maxNumOfCCBS-Requests)		
203			
204	InterrogateSS-Res ::= CHOICE {		
205	ss-Status [0] SS-Status,		
206	basicServiceGroupList	[2] BasicServiceGroupList,	
207	forwardingFeatureList	[3] ForwardingFeatureList,	
208	genericServiceInfo	[4] GenericServiceInfo }	
209			
210	USSD-Arg ::= SEQUENCE {		
211	ussd-DataCodingScheme	USSD-DataCodingScheme,	
212	ussd-String	USSD-String,	
213	... ,		
214	alertingPattern	AlertingPattern	OPTIONAL,
215	msisdn	[0] ISDN-AddressString	OPTIONAL }
216			
217	USSD-Res ::= SEQUENCE {		
218	ussd-DataCodingScheme	USSD-DataCodingScheme,	
219	ussd-String	USSD-String,	
220	...}		
221			
222	USSD-DataCodingScheme ::= OCTET STRING (SIZE (1))		
223	-- The structure of the USSD-DataCodingScheme is defined by		
224	-- the Cell Broadcast Data Coding Scheme as described in		
225	-- TS GSM 03.38		
226			
227	USSD-String ::= OCTET STRING (SIZE (1..maxUSSD-StringLength))		
228	-- The structure of the contents of the USSD-String is dependent		
229	-- on the USSD-DataCodingScheme as described in TS GSM 03.38.		
230			
231	maxUSSD-StringLength INTEGER ::= 160		
232			
233	Password ::= NumericString		
234	(FROM ("0" "1" "2" "3" "4" "5" "6" "7" "8" "9"))		
235	(SIZE (4))		
236			
237	GuidanceInfo ::= ENUMERATED {		
238	enterPW (0),		
239	enterNewPW (1),		
240	enterNewPW-Again (2)}		
241	-- How this information is really delivered to the subscriber		
242	-- (display, announcement, ...) is not part of this		
243	-- specification.		
244			
245	SS-List ::= SEQUENCE SIZE (1..maxNumOfSS) OF		
246	SS-Code		
247			
248	maxNumOfSS INTEGER ::= 30		
249			
250	SS-InfoList ::= SEQUENCE SIZE (1..maxNumOfSS) OF		
251	SS-Info		
252			
253	BasicServiceGroupList ::= SEQUENCE SIZE (1..maxNumOfBasicServiceGroups) OF		
254	BasicServiceCode		
255			
256	maxNumOfBasicServiceGroups INTEGER ::= 13		
257			
258	SS-InvocationNotificationArg ::= SEQUENCE {		
259	imsi	[0] IMSI,	
260	msisdn	[1] ISDN-AddressString,	
261	ss-Event	[2] SS-Code,	
262	-- The following SS-Code values are allowed :		
263	-- ect	SS-Code ::= '00110001'B	
264	-- multiPTY	SS-Code ::= '01010001'B	
265	-- cd	SS-Code ::= '00100100'B	
266	-- ccbs	SS-Code ::= '01000100'B	
267	ss-EventSpecification	[3] SS-EventSpecification	OPTIONAL,
268	extensionContainer	[4] ExtensionContainer	OPTIONAL,
269	...}		
270			


```

271 SS-InvocationNotificationRes ::= SEQUENCE {
272     extensionContainer      ExtensionContainer      OPTIONAL,
273     ...
274 }
275
276 SS-EventSpecification ::= SEQUENCE SIZE (1..maxEventSpecification) OF
277     AddressString
278
279 maxEventSpecification INTEGER ::= 2
280
281 RegisterCC-EntryArg ::= SEQUENCE {
282     ss-Code                [0] SS-Code,
283     ccbs-Data [1]          CCBS-Data OPTIONAL,
284     ...}
285
286 CCBS-Data ::= SEQUENCE {
287     ccbs-Feature           [0] CCBS-Feature,
288     translatedB-Number    [1] ISDN-AddressString,
289     serviceIndicator      [2] ServiceIndicator      OPTIONAL,
290     callInfo              [3] ExternalSignalInfo,
291     networkSignalInfo     [4] ExternalSignalInfo,
292     ...}
293
294 ServiceIndicator ::= BIT STRING {
295     clir-invoked (0),
296     camel-invoked (1)} (SIZE(2..32))
297 -- exception handling:
298 -- bits 2 to 31 shall be ignored if received and not understood
299
300 RegisterCC-EntryRes ::= SEQUENCE {
301     ccbs-Feature           [0] CCBS-Feature      OPTIONAL,
302     ...}
303
304 EraseCC-EntryArg ::= SEQUENCE {
305     ss-Code                [0] SS-Code,
306     ccbs-Index             [1] CCBS-Index      OPTIONAL,
307     ...}
308
309 EraseCC-EntryRes ::= SEQUENCE {
310     ss-Code                [0] SS-Code,
311     ss-Status [1] SS-Status      OPTIONAL,
312     ...}
313
314 END

```

17.7.5 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) version6 (6)}
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 TS 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	allNameIdentificationSS	SS-Code ::= '00011000'B
41	-- all name identification SS	
42	cnap	SS-Code ::= '00011001'B
43	-- calling name presentation	
44		
45	-- SS-Codes '00011010'B to '00011111'B are reserved for future	
46	-- NameIdentification Supplementary Service use.	
47		
48	allForwardingSS	SS-Code ::= '00100000'B
49	-- all forwarding SS	
50	cfu	SS-Code ::= '00100001'B
51	-- call forwarding unconditional	
52	allCondForwardingSS	SS-Code ::= '00101000'B
53	-- all conditional forwarding SS	
54	cfb	SS-Code ::= '00101001'B
55	-- call forwarding on mobile subscriber busy	
56	cfnry	SS-Code ::= '00101010'B
57	-- call forwarding on no reply	
58	cfnrc	SS-Code ::= '00101011'B
59	-- call forwarding on mobile subscriber not reachable	
60	cd	SS-Code ::= '00100100'B
61	-- call deflection	
62		
63	allCallOfferingSS	SS-Code ::= '00110000'B
64	-- reserved for possible future use	
65	-- all call offering SS includes also all forwarding SS	
66	ect	SS-Code ::= '00110001'B
67	-- explicit call transfer	
68	mah	SS-Code ::= '00110010'B
69	-- reserved for possible future use	
70	-- mobile access hunting	
71		
72	allCallCompletionSS	SS-Code ::= '01000000'B
73	-- reserved for possible future use	
74	-- all Call completion SS	
75	cw	SS-Code ::= '01000001'B
76	-- call waiting	
77	hold	SS-Code ::= '01000010'B
78	-- call hold	
79	ccbs-A	SS-Code ::= '01000011'B
80	-- completion of call to busy subscribers, originating side	
81	ccbs-B	SS-Code ::= '01000100'B
82	-- completion of call to busy subscribers, destination side	
83	-- this SS-Code is used only in InsertSubscriberData	
84		
85	allMultiPartySS	SS-Code ::= '01010000'B
86	-- reserved for possible future use	
87	-- all multiparty SS	
88	multiPTY	SS-Code ::= '01010001'B
89	-- multiparty	
90		
91	allCommunityOfInterest-SS	SS-Code ::= '01100000'B
92	-- reserved for possible future use	
93	-- all community of interest SS	
94	cug	SS-Code ::= '01100001'B
95	-- closed user group	
96		

97	allChargingSS	SS-Code ::= '01110000'B
98	-- reserved for possible future use	
99	-- all charging SS	
100	aoci	SS-Code ::= '01110001'B
101	-- advice of charge information	
102	aocc	SS-Code ::= '01110010'B
103	-- advice of charge charging	
104		
105	allAdditionalInfoTransferSS	SS-Code ::= '10000000'B
106	-- reserved for possible future use	
107	-- all additional information transfer SS	
108	uus1	SS-Code ::= '10000001'B
109	-- UUS1 user-to-user signalling	
110	uus2	SS-Code ::= '10000010'B
111	-- UUS2 user-to-user signalling	
112	uus3	SS-Code ::= '10000011'B
113	-- UUS3 user-to-user signalling	
114		
115	allBarringSS	SS-Code ::= '10010000'B
116	-- all barring SS	
117	barringOfOutgoingCalls	SS-Code ::= '10010001'B
118	-- barring of outgoing calls	
119	baoc	SS-Code ::= '10010010'B
120	-- barring of all outgoing calls	
121	boic	SS-Code ::= '10010011'B
122	-- barring of outgoing international calls	
123	boicExHC	SS-Code ::= '10010100'B
124	-- barring of outgoing international calls except those directed	
125	-- to the home PLMN	
126	barringOfIncomingCalls	SS-Code ::= '10011001'B
127	-- barring of incoming calls	
128	baic	SS-Code ::= '10011010'B
129	-- barring of all incoming calls	
130	bicRoam	SS-Code ::= '10011011'B
131	-- barring of incoming calls when roaming outside home PLMN	
132	-- Country	
133		
134	allPLMN-specificSS	SS-Code ::= '11110000'B
135	plmn-specificSS-1	SS-Code ::= '11110001'B
136	plmn-specificSS-2	SS-Code ::= '11110010'B
137	plmn-specificSS-3	SS-Code ::= '11110011'B
138	plmn-specificSS-4	SS-Code ::= '11110100'B
139	plmn-specificSS-5	SS-Code ::= '11110101'B
140	plmn-specificSS-6	SS-Code ::= '11110110'B
141	plmn-specificSS-7	SS-Code ::= '11110111'B
142	plmn-specificSS-8	SS-Code ::= '11111000'B
143	plmn-specificSS-9	SS-Code ::= '11111001'B
144	plmn-specificSS-A	SS-Code ::= '11111010'B
145	plmn-specificSS-B	SS-Code ::= '11111011'B
146	plmn-specificSS-C	SS-Code ::= '11111100'B
147	plmn-specificSS-D	SS-Code ::= '11111101'B
148	plmn-specificSS-E	SS-Code ::= '11111110'B
149	plmn-specificSS-F	SS-Code ::= '11111111'B
150		
151	allCallPrioritySS	SS-Code ::= '10100000'B
152	-- reserved for possible future use	
153	-- all call priority SS	
154	emlpp	SS-Code ::= '10100001'B
155	-- enhanced Multilevel Precedence Pre-emption (EMLPP) service	
156		
157	allLCSPrivacyException	SS-Code ::= '10110000'B
158	-- all LCS Privacy Exception Classes	
159	universal	SS-Code ::= '10110001'B
160	-- allow location by any LCS client	
161	callrelated	SS-Code ::= '10110010'B
162	-- allow location by any value added LCS client to which a call	
163	-- is established from the target MS	
164	callunrelated	SS-Code ::= '10110011'B
165	-- allow location by designated external value added LCS clients	
166	plmnoperator	SS-Code ::= '10110100'B
167	-- allow location by designated PLMN operator LCS clients	
168		

```

169 allMOLR-SS                SS-Code ::= '11000000'B
170     -- all Mobile Originating Location Request Classes
171 basicSelfLocation         SS-Code ::= '11000001'B
172     -- allow an MS to request its own location
173 autonomousSelfLocation   SS-Code ::= '11000010'B
174     -- allow an MS to perform self location without interaction
175     -- with the PLMN for a predetermined period of time
176 transferToThirdParty      SS-Code ::= '11000011'B
177     -- allow an MS to request transfer of its location to another LCS client
178
179 END

```

17.7.6 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) version6 (6)}
4
5  DEFINITIONS
6
7  IMPLICIT TAGS
8
9  ::=
10
11 BEGIN
12
13 EXPORTS
14     RoutingInfoForSM-Arg,
15     RoutingInfoForSM-Res,
16     MO-ForwardSM-Arg,
17     MO-ForwardSM-Res,
18     MT-ForwardSM-Arg,
19     MT-ForwardSM-Res,
20     ReportSM-DeliveryStatusArg,
21     ReportSM-DeliveryStatusRes,
22     AlertServiceCentreArg,
23     InformServiceCentreArg,
24     ReadyForSM-Arg,
25     ReadyForSM-Res,
26     SM-DeliveryOutcome,
27     AlertReason
28 ;
29
30 IMPORTS
31     AddressString,
32     ISDN-AddressString,
33     SignalInfo,
34     IMSI,
35     LMSI
36 FROM MAP-CommonDataTypes {
37     ccitt identified-organization (4) etsi (0) mobileDomain (0)
38     gsm-Network (1) modules (3) map-CommonDataTypes (18) version6 (6)}
39
40     AbsentSubscriberDiagnosticSM
41 FROM MAP-ER-DataTypes {
42     ccitt identified-organization (4) etsi (0) mobileDomain (0)
43     gsm-Network (1) modules (3) map-ER-DataTypes (17) version6 (6)}
44
45     ExtensionContainer
46 FROM MAP-ExtensionDataTypes {
47     ccitt identified-organization (4) etsi (0) mobileDomain (0)
48     gsm-Network (1) modules (3) map-ExtensionDataTypes (21) version6 (6)}
49 ;
50
51
52 RoutingInfoForSM-Arg ::= SEQUENCE {
53     msisdn                [0] ISDN-AddressString,
54     sm-RP-PRI [1] BOOLEAN,
55     serviceCentreAddress  [2] AddressString,
56     extensionContainer    [6] ExtensionContainer           OPTIONAL,
57     ... ,
58     gprsSupportIndicator  [7] NULL                        OPTIONAL,
59     -- gprsSupportIndicator is set only if the SMS-GMSC supports
60     -- receiving of two numbers from the HLR
61     sm-RP-MTI [8] SM-RP-MTI                               OPTIONAL,
62     sm-RP-SMEA           [9] SM-RP-SMEA                   OPTIONAL }
63

```

64	SM-RP-MTI ::= INTEGER (0..10)		
65	-- 0 SMS Deliver		
66	-- 1 SMS Status Report		
67	-- other values are reserved for future use and shall be discarded if		
68	-- received		
69			
70			
71	SM-RP-SMEA ::= OCTET STRING (SIZE (1..12))		
72	-- this parameter contains an address field which is encoded		
73	-- as defined in GSM 03.40. An address field contains 3 elements :		
74	-- address-length		
75	-- type-of-address		
76	-- address-value		
77			
78			
79	RoutingInfoForSM-Res ::= SEQUENCE {		
80	imsi	IMSI,	
81	locationInfoWithLMSI	[0] LocationInfoWithLMSI,	
82	extensionContainer	[4] ExtensionContainer	OPTIONAL,
83	...}		
84			
85	LocationInfoWithLMSI ::= SEQUENCE {		
86	networkNode-Number	[1] ISDN-AddressString,	
87	lmsi	LMSI	OPTIONAL,
88	extensionContainer	ExtensionContainer	OPTIONAL,
89	...		
90	gprsNodeIndicator	[5] NULL	OPTIONAL,
91	-- gprsNodeIndicator is set only if the SGSN number is sent as the		
92	-- Network Node Number		
93	additional-Number	[6] Additional-Number	OPTIONAL
94	-- NetworkNode-number can be either msc-number or sgsn-number		
95	}		
96			
97	Additional-Number ::= CHOICE {		
98	msc-Number	[0] ISDN-AddressString,	
99	sgsn-Number	[1] ISDN-AddressString}	
100	-- additional-number can be either msc-number or sgsn-number		
101	-- if received networkNode-number is msc-number then the		
102	-- additional number is sgsn-number		
103	-- if received networkNode-number is sgsn-number then the		
104	-- additional number is msc-number		
105			
106	MO-ForwardSM-Arg ::= SEQUENCE {		
107	sm-RP-DA	SM-RP-DA,	
108	sm-RP-OA	SM-RP-OA,	
109	sm-RP-UI	SignalInfo,	
110	extensionContainer	ExtensionContainer	OPTIONAL,
111	...		
112	imsi	IMSI	OPTIONAL }
113			
114	MO-ForwardSM-Res ::= SEQUENCE {		
115	sm-RP-UI	SignalInfo	OPTIONAL,
116	extensionContainer	ExtensionContainer	OPTIONAL,
117	...}		
118			
119	MT-ForwardSM-Arg ::= SEQUENCE {		
120	sm-RP-DA	SM-RP-DA,	
121	sm-RP-OA	SM-RP-OA,	
122	sm-RP-UI	SignalInfo,	
123	moreMessagesToSend	NULL	OPTIONAL,
124	extensionContainer	ExtensionContainer	OPTIONAL,
125	...}		
126			
127	MT-ForwardSM-Res ::= SEQUENCE {		
128	sm-RP-UI	SignalInfo	OPTIONAL,
129	extensionContainer	ExtensionContainer	OPTIONAL,
130	...}		
131			
132	SM-RP-DA ::= CHOICE {		
133	imsi	[0] IMSI,	
134	lmsi	[1] LMSI,	
135	serviceCentreAddressDA	[4] AddressString,	
136	noSM-RP-DA	[5] NULL}	
137			

```

138 SM-RP-OA ::= CHOICE {
139     msisdn                [2] ISDN-AddressString,
140     serviceCentreAddressOA [4] AddressString,
141     noSM-RP-OA           [5] NULL}
142
143 ReportSM-DeliveryStatusArg ::= SEQUENCE {
144     msisdn                ISDN-AddressString,
145     serviceCentreAddress AddressString,
146     sm-DeliveryOutcome   SM-DeliveryOutcome,
147     absentSubscriberDiagnosticSM [0] AbsentSubscriberDiagnosticSM
148
149     extensionContainer   [1] ExtensionContainer           OPTIONAL,
150
151     ...,
152     gprsSupportIndicator [2] NULL                       OPTIONAL,
153     -- gprsSupportIndicator is set only if the SMS-GMSC supports
154     -- handling of two delivery outcomes
155     deliveryOutcomeIndicator [3] NULL                   OPTIONAL,
156     -- DeliveryOutcomeIndicator is set when the SM-DeliveryOutcome
157     -- is for GPRS
158     additionalSM-DeliveryOutcome [4] SM-DeliveryOutcome OPTIONAL,
159     -- If received, additionalSM-DeliveryOutcome is for GPRS
160     additionalAbsentSubscriberDiagnosticSM [5] AbsentSubscriberDiagnosticSM OPTIONAL
161     }
162
163 SM-DeliveryOutcome ::= ENUMERATED {
164     memoryCapacityExceeded (0),
165     absentSubscriber (1),
166     successfulTransfer (2)}
167
168 ReportSM-DeliveryStatusRes ::= SEQUENCE {
169     storedMSISDN          ISDN-AddressString           OPTIONAL,
170     extensionContainer    ExtensionContainer            OPTIONAL,
171     ...}
172
173
174 AlertServiceCentreArg ::= SEQUENCE {
175     msisdn                ISDN-AddressString,
176     serviceCentreAddress AddressString,
177     ...}
178
179 InformServiceCentreArg ::= SEQUENCE {
180     storedMSISDN          ISDN-AddressString           OPTIONAL,
181     mw-Status MW-Status  OPTIONAL,
182     extensionContainer    ExtensionContainer            OPTIONAL,
183     ...}
184
185 MW-Status ::= BIT STRING {
186     sc-AddressNotIncluded (0),
187     mnrf-Set (1),
188     mcef-Set (2),
189     mnrg-Set (3)} (SIZE (6..16))
190     -- exception handling:
191     -- bits 4 to 15 shall be ignored if received and not understood
192
193 ReadyForSM-Arg ::= SEQUENCE {
194     imsi                [0] IMSI,
195     alertReason         AlertReason,
196     alertReasonIndicator NULL                       OPTIONAL,
197     -- alertReasonIndicator is set only when the alertReason
198     -- sent to HLR is for GPRS
199     extensionContainer  ExtensionContainer            OPTIONAL,
200     ...}
201
202 ReadyForSM-Res ::= SEQUENCE {
203     extensionContainer  ExtensionContainer            OPTIONAL,
204     ...}
205
206
207 AlertReason ::= ENUMERATED {
208     ms-Present (0),
209     memoryAvailable (1)}
210
211 END

```

17.7.7 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) version6 (6)}
4
5 DEFINITIONS
6
7 IMPLICIT TAGS
8
9 ::=
10
11 BEGIN
12
13 EXPORTS
14     RoamingNotAllowedParam,
15     CallBarredParam,
16     CUG-RejectParam,
17     SS-IncompatibilityCause,
18     PW-RegistrationFailureCause,
19     SM-DeliveryFailureCause,
20     SystemFailureParam,
21     DataMissingParam,
22     UnexpectedDataParam,
23     FacilityNotSupParam,
24     OR-NotAllowedParam,
25     UnknownSubscriberParam,
26     NumberChangedParam,
27     UnidentifiedSubParam,
28     IllegalSubscriberParam,
29     IllegalEquipmentParam,
30     BearerServNotProvParam,
31     TeleservNotProvParam,
32     TracingBufferFullParam,
33     NoRoamingNbParam,
34     AbsentSubscriberParam,
35     BusySubscriberParam,
36     NoSubscriberReplyParam,
37     ForwardingViolationParam,
38     ForwardingFailedParam,
39     ATI-NotAllowedParam,
40     SubBusyForMT-SMS-Param,
41     MessageWaitListFullParam,
42     AbsentSubscriberSM-Param,
43     AbsentSubscriberDiagnosticSM,
44     ResourceLimitationParam,
45     NoGroupCallNbParam,
46     IncompatibleTerminalParam,
47     ShortTermDenialParam,
48     LongTermDenialParam,
49     UnauthorizedRequestingNetwork-Param,
50     UnauthorizedLCSCClient-Param,
51     PositionMethodFailure-Param,
52     UnknownOrUnreachableLCSCClient-Param,
53     MM-EventNotSupported-Param,
54     ATSI-NotAllowedParam,
55     ATM-NotAllowedParam,
56     IllegalSS-OperationParam,
57     SS-NotAvailableParam,
58     SS-SubscriptionViolationParam,
59     InformationNotAvailableParam
60
61
62
63
64 ;
65
66 IMPORTS
67     SS-Status
68 FROM MAP-SS-DataTypes {
69     ccitt identified-organization (4) etsi (0) mobileDomain (0)
70     gsm-Network (1) modules (3) map-SS-DataTypes (14) version6 (6)}
71
72     SignalInfo,
73     BasicServiceCode,
74     NetworkResource
75 FROM MAP-CommonDataTypes {
76     ccitt identified-organization (4) etsi (0) mobileDomain (0)
77     gsm-Network (1) modules (3) map-CommonDataTypes (18) version6 (6)}
78
79     SS-Code

```

```

80 FROM MAP-SS-Code {
81   ccitt identified-organization (4) etsi (0) mobileDomain (0)
82   gsm-Network (1) modules (3) map-SS-Code (15) version6 (6)}
83
84   ExtensionContainer
85 FROM MAP-ExtensionDataTypes {
86   ccitt identified-organization (4) etsi (0) mobileDomain (0)
87   gsm-Network (1) modules (3) map-ExtensionDataTypes (21) version6 (6)}
88 ;
89
90 RoamingNotAllowedParam ::= SEQUENCE {
91   roamingNotAllowedCause          RoamingNotAllowedCause,
92   extensionContainer              ExtensionContainer          OPTIONAL,
93   ...}
94
95 RoamingNotAllowedCause ::= ENUMERATED {
96   plmnRoamingNotAllowed (0),
97   operatorDeterminedBarring (3)}
98
99 CallBarredParam ::= CHOICE {
100  callBarringCause                CallBarringCause,
101  -- call BarringCause must not be used in version 3
102  extensibleCallBarredParam       ExtensibleCallBarredParam
103  -- extensibleCallBarredParam must not be used in version <3
104  }
105
106 CallBarringCause ::= ENUMERATED {
107  barringServiceActive (0),
108  operatorBarring (1)}
109
110 ExtensibleCallBarredParam ::= SEQUENCE {
111  callBarringCause                CallBarringCause          OPTIONAL,
112  extensionContainer              ExtensionContainer          OPTIONAL,
113  ... ,
114  unauthorisedMessageOriginator  [1] NULL              OPTIONAL }
115
116 CUG-RejectParam ::= SEQUENCE {
117  cug-RejectCause                CUG-RejectCause          OPTIONAL,
118  extensionContainer              ExtensionContainer          OPTIONAL,
119  ...}
120
121 CUG-RejectCause ::= ENUMERATED {
122  incomingCallsBarredWithinCUG (0),
123  subscriberNotMemberOfCUG (1),
124  requestedBasicServiceViolatesCUG-Constraints (5),
125  calledPartySS-InteractionViolation (7)}
126
127 SS-IncompatibilityCause ::= SEQUENCE {
128  ss-Code                        [1] SS-Code          OPTIONAL,
129  basicService                   BasicServiceCode        OPTIONAL,
130  ss-Status [4] SS-Status         OPTIONAL,
131  ...}
132
133 PW-RegistrationFailureCause ::= ENUMERATED {
134  undetermined (0),
135  invalidFormat (1),
136  newPasswordsMismatch (2)}
137
138
139 SM-EnumeratedDeliveryFailureCause ::= ENUMERATED {
140  memoryCapacityExceeded (0),
141  equipmentProtocolError (1),
142  equipmentNotSM-Equipped (2),
143  unknownServiceCentre (3),
144  sc-Congestion (4),
145  invalidSME-Address (5),
146  subscriberNotSC-Subscriber (6)}
147
148 SM-DeliveryFailureCause ::= SEQUENCE {
149  sm-EnumeratedDeliveryFailureCause SM-EnumeratedDeliveryFailureCause,
150  diagnosticInfo                    SignalInfo              OPTIONAL,
151  extensionContainer                ExtensionContainer          OPTIONAL,
152  ...}
153

```



```

154 AbsentSubscriberSM-Param ::= SEQUENCE {
155     absentSubscriberDiagnosticSM      AbsentSubscriberDiagnosticSM      OPTIONAL,
156     -- AbsentSubscriberDiagnosticSM can be either for non-GPRS
157     -- or for GPRS
158     extensionContainer                ExtensionContainer                OPTIONAL,
159     ...,
160     additionalAbsentSubscriberDiagnosticSM [0] AbsentSubscriberDiagnosticSM OPTIONAL }
161 -- if received, additionalAbsentSubscriberDiagnosticSM
162 -- is for GPRS and absentSubscriberDiagnosticSM is
163 -- for non-GPRS
164
165 AbsentSubscriberDiagnosticSM ::= INTEGER (0..255)
166 -- AbsentSubscriberDiagnosticSM values are defined in ETS 300 536 (GSM 03.40)
167
168 SystemFailureParam ::= CHOICE {
169     networkResource                    NetworkResource,
170     -- networkResource must not be used in version 3
171     extensibleSystemFailureParam      ExtensibleSystemFailureParam
172     -- extensibleSystemFailureParam must not be used in version <3
173 }
174
175 ExtensibleSystemFailureParam ::= SEQUENCE {
176     networkResource                    NetworkResource                    OPTIONAL,
177     extensionContainer                ExtensionContainer                OPTIONAL,
178     ...}
179
180 DataMissingParam ::= SEQUENCE {
181     extensionContainer                ExtensionContainer                OPTIONAL,
182     ...}
183
184 UnexpectedDataParam ::= SEQUENCE {
185     extensionContainer                ExtensionContainer                OPTIONAL,
186     ...}
187
188 FacilityNotSupParam ::= SEQUENCE {
189     extensionContainer                ExtensionContainer                OPTIONAL,
190     ...}
191
192 OR-NotAllowedParam ::= SEQUENCE {
193     extensionContainer                ExtensionContainer                OPTIONAL,
194     ...}
195
196 UnknownSubscriberParam ::= SEQUENCE {
197     extensionContainer                ExtensionContainer                OPTIONAL,
198     ...,
199     unknownSubscriberDiagnostic      UnknownSubscriberDiagnostic      OPTIONAL}
200
201 UnknownSubscriberDiagnostic ::= ENUMERATED {
202     imsiUnknown (0),
203     gprsSubscriptionUnknown (1),
204     ...}
205 -- if unknown values are received in
206 -- unknownSubscriberDiagnostic they shall be discarded
207
208
209 NumberChangedParam ::= SEQUENCE {
210     extensionContainer                ExtensionContainer                OPTIONAL,
211     ...}
212
213 UnidentifiedSubParam ::= SEQUENCE {
214     extensionContainer                ExtensionContainer                OPTIONAL,
215     ...}
216
217 IllegalSubscriberParam ::= SEQUENCE {
218     extensionContainer                ExtensionContainer                OPTIONAL,
219     ...}
220
221 IllegalEquipmentParam ::= SEQUENCE {
222     extensionContainer                ExtensionContainer                OPTIONAL,
223     ...}
224
225 BearerServNotProvParam ::= SEQUENCE {
226     extensionContainer                ExtensionContainer                OPTIONAL,
227     ...}
228

```

229	TeleservNotProvParam ::= SEQUENCE {		
230	extensionContainer	ExtensionContainer	OPTIONAL,
231	...}		
232			
233	TracingBufferFullParam ::= SEQUENCE {		
234	extensionContainer	ExtensionContainer	OPTIONAL,
235	...}		
236			
237	NoRoamingNbParam ::= SEQUENCE {		
238	extensionContainer	ExtensionContainer	OPTIONAL,
239	...}		
240			
241	AbsentSubscriberParam ::= SEQUENCE {		
242	extensionContainer	ExtensionContainer	OPTIONAL,
243	...,		
244	absentSubscriberReason	[0] AbsentSubscriberReason	OPTIONAL}
245			
246	AbsentSubscriberReason ::= ENUMERATED {		
247	imsiDetach (0),		
248	restrictedArea (1),		
249	noPageResponse (2),		
250	... ,		
251	purgedMS (3)}		
252	<i>-- exception handling: at reception of other values than the ones listed the</i>		
253	<i>-- AbsentSubscriberReason shall be ignored.</i>		
254	<i>-- The AbsentSubscriberReason: purgedMS is defined for the Super-Charger feature</i>		
255	<i>-- (see TS 23.116). If this value is received in a Provide Roaming Number response</i>		
256	<i>-- it shall be mapped to the AbsentSubscriberReason: imsiDetach in the Send Routing</i>		
257	<i>-- Information response</i>		
258			
259	BusySubscriberParam ::= SEQUENCE {		
260	extensionContainer	ExtensionContainer	OPTIONAL,
261	...,		
262	ccbs-Possible	[0] NULL	OPTIONAL,
263	ccbs-Busy [1] NULL	OPTIONAL}	
264			
265	NoSubscriberReplyParam ::= SEQUENCE {		
266	extensionContainer	ExtensionContainer	OPTIONAL,
267	...}		
268			
269	ForwardingViolationParam ::= SEQUENCE {		
270	extensionContainer	ExtensionContainer	OPTIONAL,
271	...}		
272			
273	ForwardingFailedParam ::= SEQUENCE {		
274	extensionContainer	ExtensionContainer	OPTIONAL,
275	...}		
276			
277	ATI-NotAllowedParam ::= SEQUENCE {		
278	extensionContainer	ExtensionContainer	OPTIONAL,
279	...}		
280			
281	ATSI-NotAllowedParam ::= SEQUENCE {		
282	extensionContainer	ExtensionContainer	OPTIONAL,
283	...}		
284			
285	ATM-NotAllowedParam ::= SEQUENCE {		
286	extensionContainer	ExtensionContainer	OPTIONAL,
287	...}		
288			
289	IllegalSS-OperationParam ::= SEQUENCE {		
290	extensionContainer	ExtensionContainer	OPTIONAL,
291	...}		
292			
293	SS-NotAvailableParam ::= SEQUENCE {		
294	extensionContainer	ExtensionContainer	OPTIONAL,
295	...}		
296			
297	SS-SubscriptionViolationParam ::= SEQUENCE {		
298	extensionContainer	ExtensionContainer	OPTIONAL,
299	...}		
300			
301	InformationNotAvailableParam ::= SEQUENCE {		
302	extensionContainer	ExtensionContainer	OPTIONAL,
303	...}		
304			

```

305 SubBusyForMT-SMS-Param ::= SEQUENCE {
306     extensionContainer          ExtensionContainer          OPTIONAL,
307     ... ,
308     gprsConnectionSuspended    NULL          OPTIONAL }
309 -- If GprsConnectionSuspended is not understood it shall
310 -- be discarded
311
312 MessageWaitListFullParam ::= SEQUENCE {
313     extensionContainer          ExtensionContainer          OPTIONAL,
314     ... }
315
316 ResourceLimitationParam ::= SEQUENCE {
317     extensionContainer          ExtensionContainer          OPTIONAL,
318     ... }
319
320 NoGroupCallNbParam ::= SEQUENCE {
321     extensionContainer          ExtensionContainer          OPTIONAL,
322     ... }
323
324 IncompatibleTerminalParam ::= SEQUENCE {
325     extensionContainer          ExtensionContainer          OPTIONAL,
326     ... }
327
328 ShortTermDenialParam ::= SEQUENCE {
329     ... }
330
331 LongTermDenialParam ::= SEQUENCE {
332     ... }
333
334 UnauthorizedRequestingNetwork-Param ::= SEQUENCE {
335     extensionContainer          ExtensionContainer          OPTIONAL,
336     ... }
337
338 UnauthorizedLCSCClient-Param ::= SEQUENCE {
339     unauthorizedLCSCClient-Diagnostic [0] UnauthorizedLCSCClient-Diagnostic OPTIONAL,
340     extensionContainer                [1] ExtensionContainer          OPTIONAL,
341     ... }
342
343 UnauthorizedLCSCClient-Diagnostic ::= ENUMERATED {
344     noAdditionalInformation (0),
345     clientNotInMSPrivacyExceptionList (1),
346     callToClientNotSetup (2),
347     privacyOverrideNotApplicable (3),
348     disallowedByLocalRegulatoryRequirements (4),
349     ... }
350 -- exception handling:
351 -- any unrecognized value shall be ignored
352
353 PositionMethodFailure-Param ::= SEQUENCE {
354     positionMethodFailure-Diagnostic [0] PositionMethodFailure-Diagnostic OPTIONAL,
355     extensionContainer                [1] ExtensionContainer          OPTIONAL,
356     ... }
357
358 PositionMethodFailure-Diagnostic ::= ENUMERATED {
359     congestion (0),
360     insufficientResources (1),
361     insufficientMeasurementData (2),
362     inconsistentMeasurementData (3),
363     locationProcedureNotCompleted (4),
364     locationProcedureNotSupportedByTargetMS (5),
365     qosNotAttainable (6),
366     ... }
367 -- exception handling:
368 -- any unrecognized value shall be ignored
369
370 UnknownOrUnreachableLCSCClient-Param ::= SEQUENCE {
371     extensionContainer          ExtensionContainer          OPTIONAL,
372     ... }
373
374 MM-EventNotSupported-Param ::= SEQUENCE {
375     extensionContainer          ExtensionContainer          OPTIONAL,
376     ... }
377
378
379 END

```

17.7.8 Common data types

```

1  MAP-CommonDataTypes {
2      ccitt identified-organization (4) etsi (0) mobileDomain (0)
3      gsm-Network (1) modules (3) map-CommonDataTypes (18) version6 (6)}
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     maxISDN-AddressLength,
19     ISDN-SubaddressString,
20     ExternalSignalInfo,
21     Ext-ExternalSignalInfo,
22     SignalInfo,
23     maxSignalInfoLength,
24     AlertingPattern,
25
26     -- data types for numbering and identification
27     IMSI,
28     TMSI,
29     Identity,
30     SubscriberId,
31     IMEI,
32     HLR-List,
33     LMSI,
34     GlobalCellId,
35     NetworkResource,
36     NAEA-PreferredCI,
37     NAEA-CIC,
38     ASCI-CallReference,
39     SubscriberIdentity,
40
41     -- data types for CAMEL
42     CellIdOrLAI,
43
44     -- data types for subscriber management
45     BasicServiceCode,
46     Ext-BasicServiceCode,
47     EMLPP-Info,
48     EMLPP-Priority,
49
50     -- data types for geographic location
51     AgeOfLocationInformation,
52     LCSClientExternalID,
53     LCSClientInternalID
54 ;
55
56 IMPORTS
57     TeleserviceCode,
58     Ext-TeleserviceCode
59 FROM MAP-TS-Code {
60     ccitt identified-organization (4) etsi (0) mobileDomain (0)
61     gsm-Network (1) modules (3) map-TS-Code (19) version6 (6)}
62
63     BearerServiceCode,
64     Ext-BearerServiceCode
65 FROM MAP-BS-Code {
66     ccitt identified-organization (4) etsi (0) mobileDomain (0)
67     gsm-Network (1) modules (3) map-BS-Code (20) version6 (6)}
68
69     ExtensionContainer
70 FROM MAP-ExtensionDataTypes {
71     ccitt identified-organization (4) etsi (0) mobileDomain (0)
72     gsm-Network (1) modules (3) map-ExtensionDataTypes (21) version6 (6)}
73 ;
74
75

```

```

76 -- general data types
77
78 TBCD-STRING ::= OCTET STRING
79 -- This type (Telephony Binary Coded Decimal String) is used to
80 -- represent several digits from 0 through 9, *, #, a, b, c, two
81 -- digits per octet, each digit encoded 0000 to 1001 (0 to 9),
82 -- 1010 (*), 1011 (#), 1100 (a), 1101 (b) or 1110 (c); 1111 used
83 -- as filler when there is an odd number of digits.
84
85 -- bits 8765 of octet n encoding digit 2n
86 -- bits 4321 of octet n encoding digit 2(n-1) +1
87
88 AddressString ::= OCTET STRING (SIZE (1..maxAddressLength))
89 -- This type is used to represent a number for addressing
90 -- purposes. It is composed of
91 -- a) one octet for nature of address, and numbering plan
92 -- indicator.
93 -- b) digits of an address encoded as TBCD-String.
94
95 -- a) The first octet includes a one bit extension indicator, a
96 -- 3 bits nature of address indicator and a 4 bits numbering
97 -- plan indicator, encoded as follows:
98
99 -- bit 8: 1 (no extension)
100
101 -- bits 765: nature of address indicator
102 -- 000 unknown
103 -- 001 international number
104 -- 010 national significant number
105 -- 011 network specific number
106 -- 100 subscriber number
107 -- 101 reserved
108 -- 110 abbreviated number
109 -- 111 reserved for extension
110
111 -- bits 4321: numbering plan indicator
112 -- 0000 unknown
113 -- 0001 ISDN/Telephony Numbering Plan (Rec CCITT E.164)
114 -- 0010 spare
115 -- 0011 data numbering plan (CCITT Rec X.121)
116 -- 0100 telex numbering plan (CCITT Rec F.69)
117 -- 0101 spare
118 -- 0110 land mobile numbering plan (CCITT Rec E.212)
119 -- 0111 spare
120 -- 1000 national numbering plan
121 -- 1001 private numbering plan
122 -- 1111 reserved for extension
123
124 -- all other values are reserved.
125
126 -- b) The following octets representing digits of an address
127 -- encoded as a TBCD-STRING.
128
129 maxAddressLength INTEGER ::= 20
130
131 ISDN-AddressString ::=
132 AddressString (SIZE (1..maxISDN-AddressLength))
133 -- This type is used to represent ISDN numbers.
134
135 maxISDN-AddressLength INTEGER ::= 9
136

```

```

137 ISDN-SubaddressString ::=
138     OCTET STRING (SIZE (1..maxISDN-SubaddressLength))
139     -- This type is used to represent ISDN subaddresses.
140     -- It is composed of
141     -- a) one octet for type of subaddress and odd/even indicator.
142     -- b) 20 octets for subaddress information.
143
144     -- a) The first octet includes a one bit extension indicator, a
145     --     3 bits type of subaddress and a one bit odd/even indicator,
146     --     encoded as follows:
147
148     -- bit 8: 1 (no extension)
149
150     -- bits 765: type of subaddress
151     --     000 NSAP (X.213/ISO 8348 AD2)
152     --     010 User Specified
153     --     All other values are reserved
154
155     -- bit 4: odd/even indicator
156     --     0 even number of address signals
157     --     1 odd number of address signals
158     --     The odd/even indicator is used when the type of subaddress
159     --     is "user specified" and the coding is BCD.
160
161     -- bits 321: 000 (unused)
162
163     -- b) Subaddress information.
164     -- The NSAP X.213/ISO8348AD2 address shall be formatted as specified
165     -- by octet 4 which contains the Authority and Format Identifier
166     -- (AFI). The encoding is made according to the "preferred binary
167     -- encoding" as defined in X.213/ISO834AD2. For the definition
168     -- of this type of subaddress, see CCITT Rec I.334.
169
170     -- For User-specific subaddress, this field is encoded according
171     -- to the user specification, subject to a maximum length of 20
172     -- octets. When interworking with X.25 networks BCD coding should
173     -- be applied.
174
175 maxISDN-SubaddressLength INTEGER ::= 21
176
177 ExternalSignalInfo ::= SEQUENCE {
178     protocolId          ProtocolId,
179     signalInfo          SignalInfo,
180     -- Information about the internal structure is given in
181     -- subclause 7.6.9.
182     extensionContainer  ExtensionContainer OPTIONAL,
183     -- extensionContainer must not be used in version 2
184     ...}
185
186 SignalInfo ::= OCTET STRING (SIZE (1..maxSignalInfoLength))
187
188 maxSignalInfoLength INTEGER ::= 200
189     -- This NamedValue represents the theoretical maximum number of
190     -- octets which are available to carry a single data type,
191     -- without requiring segmentation to cope with the network layer
192     -- service. However, the actual maximum size available for a data
193     -- type may be lower, especially when other information elements
194     -- have to be included in the same component.
195
196 ProtocolId ::= ENUMERATED {
197     gsm-0408 (1),
198     gsm-0806 (2),
199     gsm-BSSMAP (3),
200     -- Value 3 is reserved and must not be used
201     ets-300102-1 (4)}
202
203 Ext-ExternalSignalInfo ::= SEQUENCE {
204     ext-ProtocolId      Ext-ProtocolId,
205     signalInfo          SignalInfo,
206     -- Information about the internal structure is given in
207     -- subclause 7.6.9.10
208     extensionContainer  ExtensionContainer OPTIONAL,
209     ...}
210

```

```

211 Ext-ProtocolId ::= ENUMERATED {
212     ets-300356 (1),
213     ...
214 }
215 -- exception handling:
216 -- For Ext-ExternalSignalInfo sequences containing this parameter with any
217 -- other value than the ones listed the receiver shall ignore the whole
218 -- Ext-ExternalSignalInfo sequence.
219
220 AlertingPattern ::= OCTET STRING (SIZE (1) )
221 -- This type is used to represent Alerting Pattern
222
223 -- bits 8765 : 0000 (unused)
224
225 -- bits 43 : type of Pattern
226 --     00 level
227 --     01 category
228 --     10 category
229 --     all other values are reserved.
230
231 -- bits 21 : type of alerting
232
233 alertingLevel-0 AlertingPattern ::= '00000000'B
234 alertingLevel-1 AlertingPattern ::= '00000001'B
235 alertingLevel-2 AlertingPattern ::= '00000010'B
236 -- all other values of Alerting level are reserved
237 -- Alerting Levels are defined in GSM 02.07
238
239 alertingCategory-1 AlertingPattern ::= '00000100'B
240 alertingCategory-2 AlertingPattern ::= '00000101'B
241 alertingCategory-3 AlertingPattern ::= '00000110'B
242 alertingCategory-4 AlertingPattern ::= '00000111'B
243 alertingCategory-5 AlertingPattern ::= '00001000'B
244 -- all other values of Alerting Category are reserved
245 -- Alerting categories are defined in GSM 02.07
246
247
248 -- data types for numbering and identification
249
250 IMSI ::= TBCD-STRING (SIZE (3..8))
251 -- digits of MCC, MNC, MSIN are concatenated in this order.
252
253 Identity ::= CHOICE {
254     imsi                               IMSI,
255     imsi-WithLMSI                     IMSI-WithLMSI}
256
257 IMSI-WithLMSI ::= SEQUENCE {
258     imsi                               IMSI,
259     lmsi                               LMSI,
260     -- a special value 00000000 indicates that the LMSI is not in use
261     ...}
262
263 ASCII-CallReference ::= TBCD-STRING (SIZE (1..8))
264 -- digits of VGCS/VBC-area,Group-ID are concatenated in this order.
265
266
267 TMSI ::= OCTET STRING (SIZE (1..4))
268
269 SubscriberId ::= CHOICE {
270     imsi                               [0] IMSI,
271     tmsi                               [1] TMSI}
272
273 IMEI ::= TBCD-STRING (SIZE (8))
274 -- Refers to International Mobile Station Equipment Identity
275 -- and Software Version Number (SVN) defined in TS GSM 03.03.
276 -- If the SVN is not present the last octet shall contain the
277 -- digit 0 and a filler.
278 -- If present the SVN shall be included in the last octet.
279
280 HLR-Id ::= IMSI
281 -- leading digits of IMSI, i.e. (MCC, MNC, leading digits of
282 -- MSIN) forming HLR Id defined in TS GSM 03.03.
283
284 HLR-List ::= SEQUENCE SIZE (1..maxNumOfHLR-Id) OF
285     HLR-Id
286
287 maxNumOfHLR-Id INTEGER ::= 50

```

```

288
289 IMSI ::= OCTET STRING (SIZE (4))
290
291 GlobalCellId ::= OCTET STRING (SIZE (5..7))
292   -- Refers to Cell Global Identification defined in TS GSM 03.03.
293   -- The internal structure is defined as follows:
294   -- octet 1 bits 4321      Mobile Country Code 1st digit
295   --           bits 8765      Mobile Country Code 2nd digit
296   -- octet 2 bits 4321      Mobile Country Code 3rd digit
297   --           bits 8765      Mobile Network Code 3rd digit
298   --                               or filler (1111) for 2 digit MNCs
299   -- octet 3 bits 4321      Mobile Network Code 1st digit
300   --           bits 8765      Mobile Network Code 2nd digit
301   -- octets 4 and 5        Location Area Code according to TS GSM 04.08
302   -- octets 6 and 7        Cell Identity (CI) according to TS GSM 04.08
303
304 NetworkResource ::= ENUMERATED {
305   plmn (0),
306   hlr (1),
307   vlr (2),
308   pvlr (3),
309   controllingMSC (4),
310   vmsc (5),
311   eir (6),
312   rss (7)}
313
314 NAEA-PreferredCI ::= SEQUENCE {
315   naea-PreferredCIC [0] NAEA-CIC,
316   extensionContainer [1] ExtensionContainer OPTIONAL,
317   ...}
318
319 NAEA-CIC ::= OCTET STRING (SIZE (3))
320   -- The internal structure is defined by the Carrier Identification
321   -- parameter in ANSI T1.113.3. Carrier codes between "000" and "999" may
322   -- be encoded as 3 digits using "000" to "999" or as 4 digits using
323   -- "0000" to "0999". Carrier codes between "1000" and "9999" are encoded
324   -- using 4 digits.
325
326 SubscriberIdentity ::= CHOICE {
327   imsi [0] IMSI,
328   msisdn [1] ISDN-AddressString
329 }
330
331 LCSCClientExternalID ::= SEQUENCE {
332   externalAddress [0] AddressString OPTIONAL,
333   extensionContainer [1] ExtensionContainer OPTIONAL,
334   ... }
335
336 LCSCClientInternalID ::= ENUMERATED {
337   broadcastService (0),
338   o-andM-HPLMN (1),
339   o-andM-VPLMN (2),
340   anonymousLocation (3),
341   targetMSSubscribedService (4),
342   ... }
343
344 -- data types for CAMEL
345
346 CellIdOrLAI ::= CHOICE {
347   cellIdFixedLength [0] CellIdFixedLength,
348   laiFixedLength [1] LAIFixedLength}
349
350 CellIdFixedLength ::= OCTET STRING (SIZE (7))
351   -- Refers to Cell Global Identification defined in TS GSM 03.03.
352   -- The internal structure is defined as follows:
353   -- octet 1 bits 4321      Mobile Country Code 1st digit
354   --           bits 8765      Mobile Country Code 2nd digit
355   -- octet 2 bits 4321      Mobile Country Code 3rd digit
356   --           bits 8765      Mobile Network Code 3rd digit
357   --                               or filler (1111) for 2 digit MNCs
358   -- octet 3 bits 4321      Mobile Network Code 1st digit
359   --           bits 8765      Mobile Network Code 2nd digit
360   -- octets 4 and 5        Location Area Code according to TS GSM 04.08
361   -- octets 6 and 7        Cell Identity (CI) according to TS GSM 04.08
362
363

```



```

364 LAIFixedLength ::= OCTET STRING (SIZE (5))
365 -- Refers to Location Area Identification defined in TS GSM 03.03.
366 -- The internal structure is defined as follows:
367 -- octet 1 bits 4321      Mobile Country Code 1st digit
368 --           bits 8765      Mobile Country Code 2nd digit
369 -- octet 2 bits 4321      Mobile Country Code 3rd digit
370 --           bits 8765      Mobile Network Code 3rd digit
371 --                               or filler (1111) for 2 digit MNCs
372 -- octet 3 bits 4321      Mobile Network Code 1st digit
373 --           bits 8765      Mobile Network Code 2nd digit
374 -- octets 4 and 5        Location Area Code according to TS GSM 04.08
375
376
377 -- data types for subscriber management
378
379 BasicServiceCode ::= CHOICE {
380     bearerService          [2] BearerServiceCode,
381     teleservice            [3] TeleserviceCode}
382
383 Ext-BasicServiceCode ::= CHOICE {
384     ext-BearerService       [2] Ext-BearerServiceCode,
385     ext-Teleservice         [3] Ext-TeleserviceCode}
386
387 EMLPP-Info ::= SEQUENCE {
388     maximumentitledPriority  EMLPP-Priority,
389     defaultPriority          EMLPP-Priority,
390     extensionContainer       ExtensionContainer          OPTIONAL,
391     ...}
392
393 EMLPP-Priority ::= INTEGER (0..15)
394 -- The mapping from the values A,B,0,1,2,3,4 to the integer-value is
395 -- specified as follows where A is the highest and 4 is the lowest
396 -- priority level
397 -- the integer values 7-15 are spare and shall be mapped to value 4
398
399 priorityLevelA      EMLPP-Priority ::= 6
400 priorityLevelB      EMLPP-Priority ::= 5
401 priorityLevel0      EMLPP-Priority ::= 0
402 priorityLevel1      EMLPP-Priority ::= 1
403 priorityLevel2      EMLPP-Priority ::= 2
404 priorityLevel3      EMLPP-Priority ::= 3
405 priorityLevel4      EMLPP-Priority ::= 4
406
407
408 -- data types for geographic location
409
410 AgeOfLocationInformation ::= INTEGER (0..32767)
411 -- the value represents the elapsed time in minutes since the last
412 -- network contact of the mobile station (i.e. the actuality of the
413 -- location information).
414 -- value "0" indicates that the MS is currently in contact with the
415 -- network
416 -- value "32767" indicates that the location information is at least
417 -- 32767 minutes old
418
419 END

```

17.7.9 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) version6 (6)}
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 TS 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 Ext-TeleserviceCode ::= OCTET STRING (SIZE (1..5))
21 -- This type is used to represent the code identifying a single
22 -- teleservice, a group of teleservices, or all teleservices. The
23 -- services are defined in TS GSM 02.03.
24 -- The internal structure is defined as follows:
25
26 -- OCTET 1:
27 -- bits 87654321: group (bits 8765) and specific service
28 -- (bits 4321)
29
30 -- OCTETS 2-5: reserved for future use. If received the
31 -- Ext-TeleserviceCode shall be
32 -- treated according to the exception handling defined for the
33 -- operation that uses this type.
34
35 -- Ext-TeleserviceCode includes all values defined for TeleserviceCode.
36
37
38 allTeleservices TeleserviceCode ::= '00000000'B
39
40 allSpeechTransmissionServices TeleserviceCode ::= '00010000'B
41 telephony TeleserviceCode ::= '00010001'B
42 emergencyCalls TeleserviceCode ::= '00010010'B
43
44 allShortMessageServices TeleserviceCode ::= '00100000'B
45 shortMessageMT-PP TeleserviceCode ::= '00100001'B
46 shortMessageMO-PP TeleserviceCode ::= '00100010'B
47
48 allFacsimileTransmissionServices TeleserviceCode ::= '01100000'B
49 facsimileGroup3AndAlterSpeech TeleserviceCode ::= '01100001'B
50 automaticFacsimileGroup3 TeleserviceCode ::= '01100010'B
51 facsimileGroup4 TeleserviceCode ::= '01100011'B
52
53 -- The following non-hierarchical Compound Teleservice Groups
54 -- are defined in TS GSM 02.30:
55 allDataTeleservices TeleserviceCode ::= '01110000'B
56 -- covers Teleservice Groups 'allFacsimileTransmissionServices'
57 -- and 'allShortMessageServices'
58 allTeleservices-ExeptSMS TeleserviceCode ::= '10000000'B
59 -- covers Teleservice Groups 'allSpeechTransmissionServices' and
60 -- 'allFacsimileTransmissionServices'
61 --
62 -- Compound Teleservice Group Codes are only used in call
63 -- independent supplementary service operations, i.e. they
64 -- are not used in InsertSubscriberData or in
65 -- DeleteSubscriberData messages.
66
67 allVoiceGroupCallServices TeleserviceCode ::= '10010000'B
68
69 voiceGroupCall TeleserviceCode ::= '10010001'B
70 voiceBroadcastCall TeleserviceCode ::= '10010010'B
71
72 allPLMN-specificTS TeleserviceCode ::= '11010000'B
73 plmn-specificTS-1 TeleserviceCode ::= '11010001'B
74 plmn-specificTS-2 TeleserviceCode ::= '11010010'B
75 plmn-specificTS-3 TeleserviceCode ::= '11010011'B
76 plmn-specificTS-4 TeleserviceCode ::= '11010100'B
77 plmn-specificTS-5 TeleserviceCode ::= '11010101'B
78 plmn-specificTS-6 TeleserviceCode ::= '11010110'B
79 plmn-specificTS-7 TeleserviceCode ::= '11010111'B
80 plmn-specificTS-8 TeleserviceCode ::= '11011000'B
81 plmn-specificTS-9 TeleserviceCode ::= '11011001'B
82 plmn-specificTS-A TeleserviceCode ::= '11011010'B
83 plmn-specificTS-B TeleserviceCode ::= '11011011'B
84 plmn-specificTS-C TeleserviceCode ::= '11011100'B
85 plmn-specificTS-D TeleserviceCode ::= '11011101'B
86 plmn-specificTS-E TeleserviceCode ::= '11011110'B
87 plmn-specificTS-F TeleserviceCode ::= '11011111'B
88
89 END

```

17.7.10 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) version6 (6)}

```

```

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 TS GSM 02.02.
15   -- The internal structure is defined as follows:
16   --
17   -- plmn-specific bearer services:
18   -- bits 87654321: defined by the HPLMN operator
19
20   -- rest of bearer services:
21   -- bit 8: 0 (unused)
22   -- bits 7654321: group (bits 7654), and rate, if applicable
23   -- (bits 321)
24
25 Ext-BearerServiceCode ::= OCTET STRING (SIZE (1..5))
26   -- This type is used to represent the code identifying a single
27   -- bearer service, a group of bearer services, or all bearer
28   -- services. The services are defined in TS GSM 02.02.
29   -- The internal structure is defined as follows:
30   --
31   -- OCTET 1:
32   -- plmn-specific bearer services:
33   -- bits 87654321: defined by the HPLMN operator
34   --
35   -- rest of bearer services:
36   -- bit 8: 0 (unused)
37   -- bits 7654321: group (bits 7654), and rate, if applicable
38   -- (bits 321)
39
40   -- OCTETS 2-5: reserved for future use. If received the
41   -- Ext-TeleserviceCode shall be
42   -- treated according to the exception handling defined for the
43   -- operation that uses this type.
44
45
46   -- Ext-BearerServiceCode includes all values defined for BearerServiceCode.
47
48
49 allBearerServices          BearerServiceCode ::= '00000000'B
50
51 allDataCDA-Services       BearerServiceCode ::= '00010000'B
52 dataCDA-300bps            BearerServiceCode ::= '00010001'B
53 dataCDA-1200bps           BearerServiceCode ::= '00010010'B
54 dataCDA-1200-75bps        BearerServiceCode ::= '00010011'B
55 dataCDA-2400bps           BearerServiceCode ::= '00010100'B
56 dataCDA-4800bps           BearerServiceCode ::= '00010101'B
57 dataCDA-9600bps           BearerServiceCode ::= '00010110'B
58 general-dataCDA           BearerServiceCode ::= '00010111'B
59
60 allDataCDS-Services       BearerServiceCode ::= '00011000'B
61 dataCDS-1200bps           BearerServiceCode ::= '00011010'B
62 dataCDS-2400bps           BearerServiceCode ::= '00011100'B
63 dataCDS-4800bps           BearerServiceCode ::= '00011101'B
64 dataCDS-9600bps           BearerServiceCode ::= '00011110'B
65 general-dataCDS           BearerServiceCode ::= '00011111'B
66
67 allPadAccessCA-Services   BearerServiceCode ::= '00100000'B
68 padAccessCA-300bps        BearerServiceCode ::= '00100001'B
69 padAccessCA-1200bps       BearerServiceCode ::= '00100010'B
70 padAccessCA-1200-75bps    BearerServiceCode ::= '00100011'B
71 padAccessCA-2400bps       BearerServiceCode ::= '00100100'B
72 padAccessCA-4800bps       BearerServiceCode ::= '00100101'B
73 padAccessCA-9600bps       BearerServiceCode ::= '00100110'B
74 general-padAccessCA       BearerServiceCode ::= '00100111'B
75
76 allDataPDS-Services       BearerServiceCode ::= '00101000'B
77 dataPDS-2400bps           BearerServiceCode ::= '00101100'B
78 dataPDS-4800bps           BearerServiceCode ::= '00101101'B
79 dataPDS-9600bps           BearerServiceCode ::= '00101110'B
80 general-dataPDS           BearerServiceCode ::= '00101111'B
81

```

```

82 allAlternateSpeech-DataCDA      BearerServiceCode ::= '00110000'B
83
84 allAlternateSpeech-DataCDS      BearerServiceCode ::= '00111000'B
85
86 allSpeechFollowedByDataCDA     BearerServiceCode ::= '01000000'B
87
88 allSpeechFollowedByDataCDS     BearerServiceCode ::= '01001000'B
89
90 -- The following non-hierarchical Compound Bearer Service
91 -- Groups are defined in TS GSM 02.30:
92 allDataCircuitAsynchronous     BearerServiceCode ::= '01010000'B
93 -- covers "allDataCDA-Services", "allAlternateSpeech-DataCDA" and
94 -- "allSpeechFollowedByDataCDA"
95 allAsynchronousServices        BearerServiceCode ::= '01100000'B
96 -- covers "allDataCDA-Services", "allAlternateSpeech-DataCDA",
97 -- "allSpeechFollowedByDataCDA" and "allPadAccessCDA-Services"
98 allDataCircuitSynchronous      BearerServiceCode ::= '01011000'B
99 -- covers "allDataCDS-Services", "allAlternateSpeech-DataCDS" and
100 -- "allSpeechFollowedByDataCDS"
101 allSynchronousServices         BearerServiceCode ::= '01101000'B
102 -- covers "allDataCDS-Services", "allAlternateSpeech-DataCDS",
103 -- "allSpeechFollowedByDataCDS" and "allDataPDS-Services"
104 --
105 -- Compound Bearer Service Group Codes are only used in call
106 -- independent supplementary service operations, i.e. they
107 -- are not used in InsertSubscriberData or in
108 -- DeleteSubscriberData messages.
109
110 allPLMN-specificBS             BearerServiceCode ::= '11010000'B
111 plmn-specificBS-1              BearerServiceCode ::= '11010001'B
112 plmn-specificBS-2              BearerServiceCode ::= '11010010'B
113 plmn-specificBS-3              BearerServiceCode ::= '11010011'B
114 plmn-specificBS-4              BearerServiceCode ::= '11010100'B
115 plmn-specificBS-5              BearerServiceCode ::= '11010101'B
116 plmn-specificBS-6              BearerServiceCode ::= '11010110'B
117 plmn-specificBS-7              BearerServiceCode ::= '11010111'B
118 plmn-specificBS-8              BearerServiceCode ::= '11011000'B
119 plmn-specificBS-9              BearerServiceCode ::= '11011001'B
120 plmn-specificBS-A              BearerServiceCode ::= '11011010'B
121 plmn-specificBS-B              BearerServiceCode ::= '11011011'B
122 plmn-specificBS-C              BearerServiceCode ::= '11011100'B
123 plmn-specificBS-D              BearerServiceCode ::= '11011101'B
124 plmn-specificBS-E              BearerServiceCode ::= '11011110'B
125 plmn-specificBS-F              BearerServiceCode ::= '11011111'B
126
127 END

```

17.7.11 Extension data types

```

1  MAP-ExtensionDataTypes {
2    ccitt identified-organization (4) etsi (0) mobileDomain (0)
3    gsm-Network (1) modules (3) map-ExtensionDataTypes (21) version6 (6)}
4
5  DEFINITIONS
6
7  IMPLICIT TAGS
8
9  ::=
10
11  BEGIN
12
13  EXPORTS
14
15    PrivateExtension,
16    ExtensionContainer;
17
18
19  -- IOC for private MAP extensions
20
21
22  MAP-EXTENSION ::= CLASS {
23    &ExtensionType                                OPTIONAL,
24    &extensionId                                  OBJECT IDENTIFIER }
25    -- The length of the Object Identifier shall not exceed 16 octets and the
26    -- number of components of the Object Identifier shall not exceed 16
27
28

```

```

29
30 -- data types
31
32 ExtensionContainer ::= SEQUENCE {
33     privateExtensionList      [0]PrivateExtensionList      OPTIONAL,
34     pcs-Extensions            [1]PCS-Extensions            OPTIONAL,
35     ...}
36
37 PrivateExtensionList ::= SEQUENCE SIZE (1..maxNumOfPrivateExtensions) OF
38     PrivateExtension
39
40 PrivateExtension ::= SEQUENCE {
41     extId                      MAP-EXTENSION.&extensionId
42                               ({ExtensionSet}),
43     extType                    MAP-EXTENSION.&ExtensionType
44                               ({ExtensionSet}@extId)      OPTIONAL}
45
46 maxNumOfPrivateExtensions INTEGER ::= 10
47
48 ExtensionSet MAP-EXTENSION ::=
49     {...
50     -- ExtensionSet is the set of all defined private extensions
51     }
52
53     -- Unsupported private extensions shall be discarded if received.
54
55
56 PCS-Extensions ::= SEQUENCE {
57     ...}
58
59 END
60

```

17.7.12 Group Call data types

```

1  MAP-GR-DataTypes {
2      ccitt identified-organization (4) etsi (0) mobileDomain (0)
3      gsm-Network (1) modules (3) map-GR-DataTypes (23) version6 (6)}
4
5  DEFINITIONS
6
7  IMPLICIT TAGS
8
9  ::=
10
11 BEGIN
12
13 EXPORTS
14     PrepareGroupCallArg,
15     PrepareGroupCallRes,
16     SendGroupCallEndSignalArg,
17     SendGroupCallEndSignalRes,
18     ForwardGroupCallSignallingArg,
19     ProcessGroupCallSignallingArg
20 ;
21
22 IMPORTS
23     ISDN-AddressString,
24     IMSI,
25     EMLPP-Priority,
26     ASCII-CallReference
27 FROM MAP-CommonDataTypes {
28     ccitt identified-organization (4) etsi (0) mobileDomain (0)
29     gsm-Network (1) modules (3) map-CommonDataTypes (18) version6 (6)}
30
31     Ext-TeleserviceCode
32 FROM MAP-TS-Code {
33     ccitt identified-organization (4) etsi (0) mobileDomain (0)
34     gsm-Network (1) modules (3) map-TS-Code (19) version6 (6)}
35
36     Kc
37 FROM MAP-MS-DataTypes {
38     ccitt identified-organization (4) etsi (0) mobileDomain (0)
39     gsm-Network (1) modules (3) map-MS-DataTypes (11) version6 (6)}
40
41
42     ExtensionContainer

```

```

43 FROM MAP-ExtensionDataTypes {
44     ccitt identified-organization (4) etsi (0) mobileDomain (0)
45     gsm-Network (1) modules (3) map-ExtensionDataTypes (21) version6 (6)}
46 ;
47
48

```

```

49 PrepareGroupCallArg ::= SEQUENCE {
50     teleservice                Ext-TeleserviceCode,
51     asciCallReference          ASCII-CallReference,
52     codec-Info                 CODEC-Info,
53     cipheringAlgorithm         CipheringAlgorithm,
54     groupKeyNumber             [0]GroupKeyNumber           OPTIONAL,
55     groupKey                   [1]Kc                       OPTIONAL,
56     priority                   [2]EMLPP-Priority          OPTIONAL,
57     uplinkFree                 [3] NULL                   OPTIONAL,
58     extensionContainer         [4] ExtensionContainer      OPTIONAL,
59     ...}
60

```

```

61 PrepareGroupCallRes ::= SEQUENCE {
62     groupCallNumber            ISDN-AddressString,
63     extensionContainer         ExtensionContainer           OPTIONAL,
64     ...}
65

```

```

66 SendGroupCallEndSignalArg ::= SEQUENCE {
67     imsi                       IMSI                       OPTIONAL,
68     extensionContainer         ExtensionContainer           OPTIONAL,
69     ...}
70

```

```

71 SendGroupCallEndSignalRes ::= SEQUENCE {
72     extensionContainer         ExtensionContainer           OPTIONAL,
73     ...}
74

```

```

75 ForwardGroupCallSignallingArg ::= SEQUENCE {
76     imsi                       IMSI                       OPTIONAL,
77     uplinkRequestAck           [0] NULL            OPTIONAL,
78     uplinkReleaseIndication    [1] NULL            OPTIONAL,
79     uplinkRejectCommand        [2] NULL            OPTIONAL,
80     uplinkSeizedCommand        [3] NULL            OPTIONAL,
81     uplinkReleaseCommand       [4] NULL            OPTIONAL,
82     extensionContainer         ExtensionContainer           OPTIONAL,
83     ...}
84

```

```

85 ProcessGroupCallSignallingArg ::= SEQUENCE {
86     uplinkRequest              [0] NULL            OPTIONAL,
87     uplinkReleaseIndication    [1] NULL            OPTIONAL,
88     releaseGroupCall           [2] NULL            OPTIONAL,
89     extensionContainer         ExtensionContainer           OPTIONAL,
90     ...}
91

```

```

92 GroupKeyNumber ::= INTEGER (0..15)
93

```

```

94 CODEC-Info ::= OCTET STRING (SIZE (5..10))
95     -- Refers to channel type
96     -- coded according to GSM 08.08
97

```

```

98
99 CipheringAlgorithm ::= OCTET STRING (SIZE (1))
100     -- Refers to 'permitted algorithms' in 'encryption information'
101     -- coded according to GSM 08.08:
102
103     -- Bits 8-1
104     -- 8765 4321
105     -- 0000 0001           No encryption
106     -- 0000 0010           GSM A5/1
107     -- 0000 0100           GSM A5/2
108     -- 0000 1000           GSM A5/3
109     -- 0001 0000           GSM A5/4
110     -- 0010 0000           GSM A5/5
111     -- 0100 0000           GSM A5/6
112     -- 1000 0000           GSM A5/7
113
114
115

```

```

116
117
118
119 END

```

17.7.13 Location service data types

```

1  MAP-LCS-DataTypes {
2      ccitt identified-organization (4) etsi (0) mobileDomain (0)
3      gsm-Network (1) modules (3) map-LCS-DataTypes (25) version6 (6)}
4
5  DEFINITIONS
6  IMPLICIT TAGS
7  ::=
8  BEGIN
9
10 EXPORTS
11     RoutingInfoForLCS-Arg,
12     RoutingInfoForLCS-Res,
13     ProvideSubscriberLocation-Arg,
14     ProvideSubscriberLocation-Res,
15     SubscriberLocationReport-Arg,
16     SubscriberLocationReport-Res,
17     LocationType,
18     LCSClientName,
19     LCS-QoS,
20     Horizontal-Accuracy,
21     ResponseTime,
22     Ext-GeographicalInformation
23 ;
24
25 IMPORTS
26     AddressString,
27     ISDN-AddressString,
28     IMEI,
29     IMSI,
30     LMSI,
31     SubscriberIdentity,
32     AgeOfLocationInformation,
33     LCSClientExternalID,
34     LCSClientInternalID
35 FROM MAP-CommonDataTypes {
36     ccitt identified-organization (4) etsi (0) mobileDomain (0)
37     gsm-Network (1) modules (3) map-CommonDataTypes (18) version6 (6)}
38
39     ExtensionContainer
40 FROM MAP-ExtensionDataTypes {
41     ccitt identified-organization (4) etsi (0) mobileDomain (0)
42     gsm-Network (1) modules (3) map-ExtensionDataTypes (21) version6 (6)}
43
44     USSD-DataCodingScheme,
45     USSD-String
46 FROM MAP-SS-DataTypes {
47     ccitt identified-organization (4) etsi (0) mobileDomain (0) gsm-Network (1) modules (3)
48     map-SS-DataTypes (14) version6 (6)}
49 ;
50
51

```

<pre> 52 RoutingInfoForLCS-Arg ::= SEQUENCE { 53 mlcNumber 54 targetMS 55 extensionContainer 56 ...} </pre>	<pre> [0] ISDN-AddressString, [1] SubscriberIdentity, [2] ExtensionContainer </pre>	<pre> OPTIONAL, </pre>
---	---	------------------------

<pre> 58 RoutingInfoForLCS-Res ::= SEQUENCE { 59 targetMS 60 lcsLocationInfo 61 extensionContainer 62 ...} </pre>	<pre> [0] SubscriberIdentity, [1] LCSLocationInfo, [2] ExtensionContainer </pre>	<pre> OPTIONAL, </pre>
---	--	------------------------

<pre> 64 LCSLocationInfo ::= SEQUENCE { 65 msc-Number 66 lmsi 67 extensionContainer 68 ...} </pre>	<pre> ISDN-AddressString, [0] LMSI [1] ExtensionContainer </pre>	<pre> OPTIONAL, OPTIONAL, </pre>
--	--	----------------------------------

69

```

70 ProvideSubscriberLocation-Arg ::= SEQUENCE {
71     locationType                LocationType,
72     mlc-Number                  ISDN-AddressString,
73     lcs-ClientID                [0] LCS-ClientID                OPTIONAL,
74     privacyOverride             [1] NULL                    OPTIONAL,
75     imsi                        [2] IMSI                    OPTIONAL,
76     msisdn                      [3] ISDN-AddressString        OPTIONAL,
77     lmsi                        [4] LMSI                    OPTIONAL,
78     imei                        [5] IMEI                    OPTIONAL,
79     lcs-Priority                [6] LCS-Priority            OPTIONAL,
80     lcs-QoS                     [7] LCS-QoS                OPTIONAL,
81     extensionContainer          [8] ExtensionContainer        OPTIONAL,
82     ... }
83
84 -- one of imsi or msisdn is mandatory
85
86 LocationType ::= SEQUENCE {
87     locationEstimateType        [0] LocationEstimateType,
88     ... }
89
90 LocationEstimateType ::= ENUMERATED {
91     currentLocation              (0),
92     currentOrLastKnownLocation  (1),
93     initialLocation              (2),
94     ... }
95 -- exception handling:
96 -- a ProvideSubscriberLocation-Arg containing an unrecognized LocationEstimateType
97 -- shall be rejected by the receiver with a return error cause of unexpected data value
98
99 LCS-ClientID ::= SEQUENCE {
100    lcsClientType                [0] LCSClientType,
101    lcsClientExternalID          [1] LCSClientExternalID    OPTIONAL,
102    lcsClientDialedByMS         [2] AddressString          OPTIONAL,
103    lcsClientInternalID         [3] LCSClientInternalID    OPTIONAL,
104    lcsClientName                [4] LCSClientName          OPTIONAL,
105    ... }
106
107 LCSClientType ::= ENUMERATED {
108    emergencyServices            (0),
109    valueAddedServices           (1),
110    plmnOperatorServices         (2),
111    lawfulInterceptServices     (3),
112    ... }
113 -- exception handling:
114 -- unrecognized values may be ignored if the LCS client uses the privacy override
115 -- otherwise, an unrecognized value shall be treated as unexpected data by a receiver
116 -- a return error shall then be returned if received in a MAP invoke
117
118 LCSClientName ::= SEQUENCE {
119    dataCodingScheme             [0] USSD-DataCodingScheme,
120    nameString                   [2] NameString,
121    ... }
122
123 -- The USSD-DataCodingScheme shall indicate use of the default alphabet through the
124 -- following encoding
125 -- bit 7 6 5 4 3 2 1 0
126 --      0 0 0 0 1 1 1 1
127
128 NameString ::= USSD-String (SIZE (1..maxNameStringLength))
129
130 maxNameStringLength INTEGER ::= 63
131
132 LCS-Priority ::= OCTET STRING (SIZE (1))
133 -- 0 = highest priority
134 -- 1 = normal priority
135 -- all other values treated as 1
136
137 LCS-QoS ::= SEQUENCE {
138    horizontal-accuracy          [0] Horizontal-Accuracy    OPTIONAL,
139    verticalCoordinateRequest     [1] NULL                    OPTIONAL,
140    vertical-accuracy            [2] Vertical-Accuracy        OPTIONAL,
141    responseTime                 [3] ResponseTime            OPTIONAL,
142    extensionContainer           [4] ExtensionContainer        OPTIONAL,
143    ... }
144

```



```

145 Horizontal-Accuracy ::= OCTET STRING (SIZE (1))
146     -- bit 8 = 0
147     -- bits 7-1 = 7 bit Uncertainty Code defined in GSM 03.32
148
149 Vertical-Accuracy ::= OCTET STRING (SIZE (1))
150     -- bit 8 = 0
151     -- bits 7-1 = 7 bit Vertical Uncertainty Code defined in GSM 03.32
152
153 ResponseTime ::= SEQUENCE {
154     responseTimeCategory          ResponseTimeCategory,
155     ... }
156 -- note: an expandable SEQUENCE simplifies later addition of a numeric response time.
157
158 ResponseTimeCategory ::= ENUMERATED {
159     lowdelay (0),
160     delaytolerant (1),
161     ... }
162 -- exception handling:
163 -- an unrecognized value shall be treated the same as value 1 (delaytolerant)
164
165 ProvideSubscriberLocation-Res ::= SEQUENCE {
166     locationEstimate              Ext-GeographicalInformation,
167     ageOfLocationEstimate         [0] AgeOfLocationInformation      OPTIONAL,
168     extensionContainer            [1] ExtensionContainer           OPTIONAL,
169     ... }
170
171 Ext-GeographicalInformation ::= OCTET STRING (SIZE (1..maxExt-GeographicalInformation))
172 -- Refers to geographical information defined in GSM 03.32.
173 -- This is composed of 1 or more octets with an internal structure according to GSM 03.32
174 -- Octet 1: Type of shape, only the following shapes in GSM 03.32 are allowed:
175 --     (a) Ellipsoid point with uncertainty circle
176 --     (b) Ellipsoid point with uncertainty ellipse
177 --     (c) Ellipsoid point with altitude and uncertainty ellipsoid
178 --     (d) Ellipsoid Arc
179 -- Any other value in octet 1 shall be treated as invalid
180 -- Octets 2 to 8 for case (a) - Ellipsoid point with uncertainty circle
181 --     Degrees of Latitude                3 octets
182 --     Degrees of Longitude              3 octets
183 --     Uncertainty code                  1 octet
184 -- Octets 2 to 11 for case (b) - Ellipsoid point with uncertainty ellipse:
185 --     Degrees of Latitude                3 octets
186 --     Degrees of Longitude              3 octets
187 --     Uncertainty semi-major axis       1 octet
188 --     Uncertainty semi-minor axis       1 octet
189 --     Angle of major axis               1 octet
190 --     Confidence                         1 octet
191 -- Octets 2 to 14 for case (c) - Ellipsoid point with altitude and uncertainty ellipsoid
192 --     Degrees of Latitude                3 octets
193 --     Degrees of Longitude              3 octets
194 --     Altitude                          2 octets
195 --     Uncertainty semi-major axis       1 octet
196 --     Uncertainty semi-minor axis       1 octet
197 --     Angle of major axis               1 octet
198 --     Uncertainty altitude              1 octet
199 --     Confidence                         1 octet
200 -- Octets 2 to 13 for case (d) - Ellipsoid Arc
201 --     Degrees of Latitude                3 octets
202 --     Degrees of Longitude              3 octets
203 --     Inner radius                       2 octets
204 --     Uncertainty radius                 1 octet
205 --     Offset angle                       1 octet
206 --     Included angle                     1 octet
207 --     Confidence                         1 octet
208 --
209 --
210 -- An Ext-GeogrphicalInformation parameter containing any other shape or an incorrect number
211 -- of octets or coding according to GSM 03.32 shall be treated as invalid data by a receiver
212
213 maxExt-GeographicalInformation INTEGER ::= 20
214 -- the maximum length allows for further shapes in GSM 03.32 to be included in later versions
215 -- of GSM 09.02
216

```

```

217 SubscriberLocationReport-Arg ::= SEQUENCE {
218     lcs-Event                LCS-Event,
219     lcs-ClientID             LCS-ClientID,
220     lcsLocationInfo         LCSLocationInfo,
221     msisdn                   [0] ISDN-AddressString           OPTIONAL,
222     imsi                     [1] IMSI                       OPTIONAL,
223     imei                     [2] IMEI                       OPTIONAL,
224     na-ESRD                  [3] ISDN-AddressString           OPTIONAL,
225     na-ESRK                  [4] ISDN-AddressString           OPTIONAL,
226     locationEstimate         [5] Ext-GeographicalInformation  OPTIONAL,
227     ageOfLocationEstimate    [6] AgeOfLocationInformation  OPTIONAL,
228     extensionContainer       [7] ExtensionContainer           OPTIONAL,
229     ...}
230
231 -- one of msisdn or imsi is mandatory
232
233 LCS-Event ::= ENUMERATED {
234     emergencyCallOrigination (0),
235     emergencyCallRelease (1),
236     mo-lr (2),
237     ... }
238 -- exception handling:
239 -- a SubscriberLocationReport-Arg containing an unrecognized LCS-Event
240 -- shall be rejected by a receiver with a return error cause of unexpected data value
241
242 SubscriberLocationReport-Res ::= SEQUENCE {
243     extensionContainer       ExtensionContainer           OPTIONAL,
244     ...}
245
246
247
248 END
249

```

18 General on MAP user procedures

18.1 Introduction

Clauses 18 to 25 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. The present document 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 the present document.

18.2 Common aspects of user procedure descriptions

18.2.1 General conventions

For each signalling procedure the present document 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, the present document 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 18.2/1). 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 14 to 17.

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.

18.2.2 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 7 to 12, 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

```

Figure 18.2/1: Interfaces applicable to the MAP-User

18.2.3 Convention on primitives parameters

18.2.3.1 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.

18.2.3.2 Close service

When a pre-arranged release 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.

18.2.4 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:

18.2.4.1 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 5.

If version 2 is selected and a MAP-CLOSE Confirm primitive in response to the MAP-OPEN request is received 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 Vr procedure".

If version 3 is selected and a MAP-CLOSE Confirm primitive in response to the MAP-OPEN request is received 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 or version two context. This is informally represented in the SDL diagrams by task symbols indicating "Perform Vr procedure" .

18.2.4.2 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; if it refers to a version two context, the associated V2 procedure is executed, otherwise the associated V3 procedure is executed.

18.2.5 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.

18.2.6 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.

18.3 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 14.3). 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.

19 Mobility procedures

19.1 Location management Procedures

For non-GPRS subscribers, 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:

Update_Location_Area_VLR, subclause 19.1.1.3;

Process Update Location:

Initiator: Update_Location_Area_VLR, subclause 19.1.1.3;

Responder: Update_Location_HLR, subclause 19.1.1.4;

Process Send Identification:

Initiator: Send_Identification_VLR, subclause 19.1.1.x;

Responder: Send_Identification_PVLR, subclause 19.1.1.5;

Process Cancel Location:

Initiator: Cancel_Location_HLR, subclause 19.1.2.2;

Responder: Cancel_Location_VLR, subclause 19.1.2.3;

Process Purge MS:

Initiator: Purge_MS_VLR, subclause 19.1.4.2;

Responder: Purge_MS_HLR, subclause 19.1.4.3.

For GPRS subscribers, this subclause comprises a number of other processes to handle the mobile nature of the subscriber. The processes will be addressed by SCCP Sub-System Number (SGSN or HLR) and the Application Context. The following processes are defined in this subclause:

Process GPRS Update Location:

Initiator: GPRS_Update_Location_Area_VLR, subclause 19.1.1.3, or
SGSN_Update_HLR, subclause 19.1.1.8,

Responder: Update_GPRS_Location_HLR, subclause 19.1.1.4;

Process Cancel Location:

Initiator: Cancel_GPRS_Location_HLR, subclause 19.1.2.2;

Responder: Cancel_Location_SGSN, subclause 19.1.2.4;

Process Purge MS:

Initiator: Purge_MS_SGSN, subclause 19.1.4.4;

Responder: Purge_MS_HLR, subclause 19.1.4.3.

The following existing process is also used for GPRS subscribers :

Process Subscriber Present HLR:

Initiator: Subscriber_Present_HLR, subclause 19.1.1.7;

Responder: Short_Message_Alert_IWMSC, subclause 23.4.3;

Location Management Coordinator HLR

Sheet 1: 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.

Process Location_Management_Coordinator_HLR

19.1_3(1)

Figure 19.1/3
Location management coordination process in the HLR

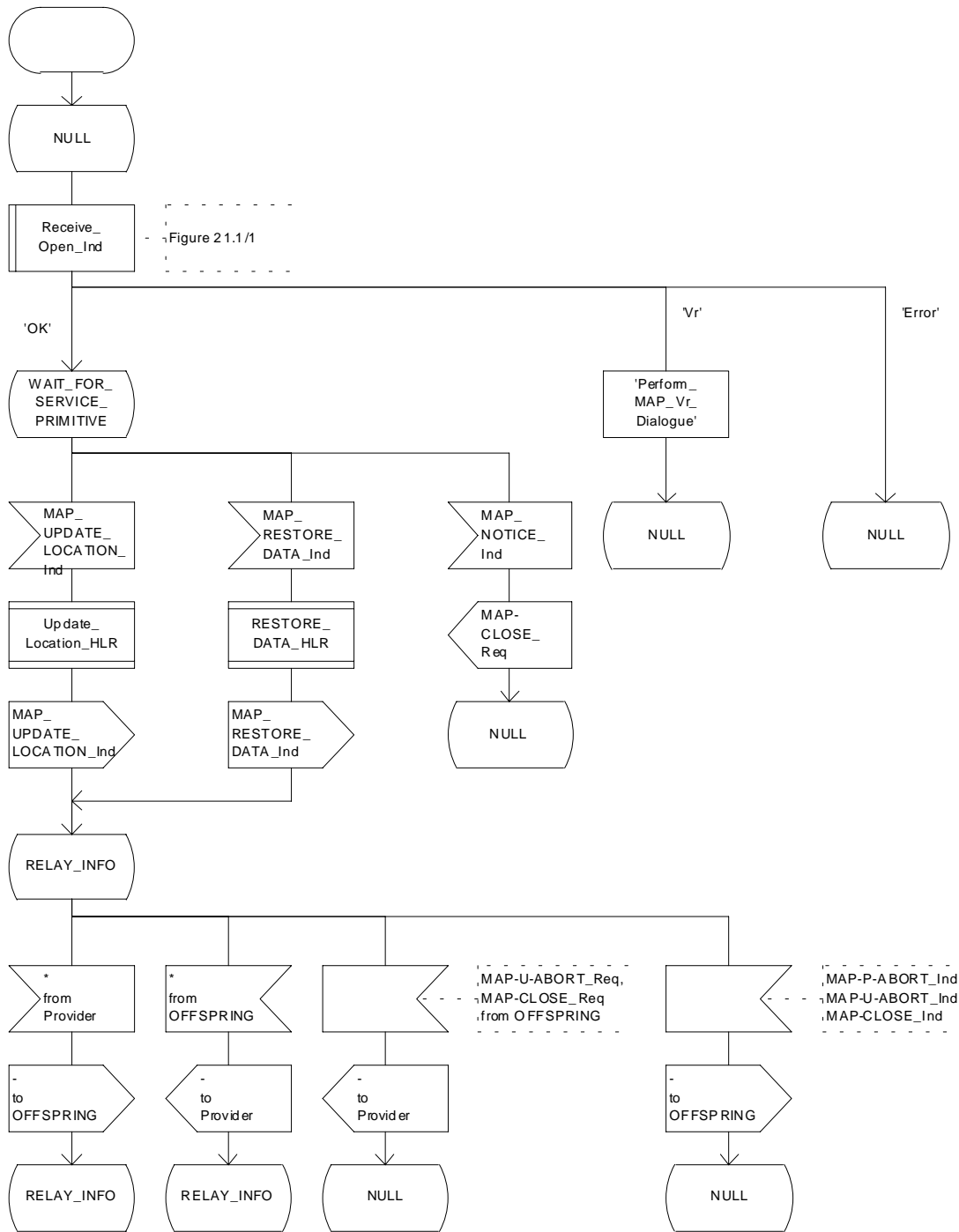


Figure 19.1/3: Process Location_Management_Coordinator_HLR

19.1.1 Location updating

19.1.1.1 General

The location updating procedure is used to update the location information held in the network. For GPRS subscribers, this procedure describes also updating of the SGSN and, if Gs interface is installed, updating of the VLR in combination with an attach/routing area updating in the SGSN. This location information is used to route incoming calls, packet data, short messages and unstructured supplementary service data to the roaming subscriber. Additionally, this procedure is used to provide the VLR and/or the SGSN with the information that a subscriber already registered, but being detached, is reachable again (IMSI Attach and/or GPRS Attach, see GSM 03.12 and GSM 03.60). The use of the IMSI 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 and, for GPRS subscribers, the SGSN the subscriber is attached to. The VLR and the SGSN contain more detailed location information, i.e. the location area the subscriber is actually roaming in (for the VLR) and the routing area (RA) where the GPRS subscriber is located (for SGSN). Therefore, the VLR needs to be updated at each location area change (see figure 19.1.1/1 for this procedure) and the SGSN needs to be updated at each routing area change. The HLR needs updating only in the following cases:

- when the subscriber registers in a new VLR or SGSN, i.e. the VLR or SGSN 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, VLR or SGSN restoration, and the VLR or SGSN 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 19.1.1/2). If the IMSI cannot be retrieved from PVLR, it is requested from the MS (see figure 19.1.1/3).

The stage 2 specification for GPRS is in GSM 03.60. The interworking between the MAP signalling procedures and the GPRS procedures in the SGSN is shown by the transfer of signals between these procedures (see subclause 19.1.1.8).

The message flow for successful GPRS Attach/ RA update procedure (with Gs interface not installed) is shown in figure 19.1.1/4.

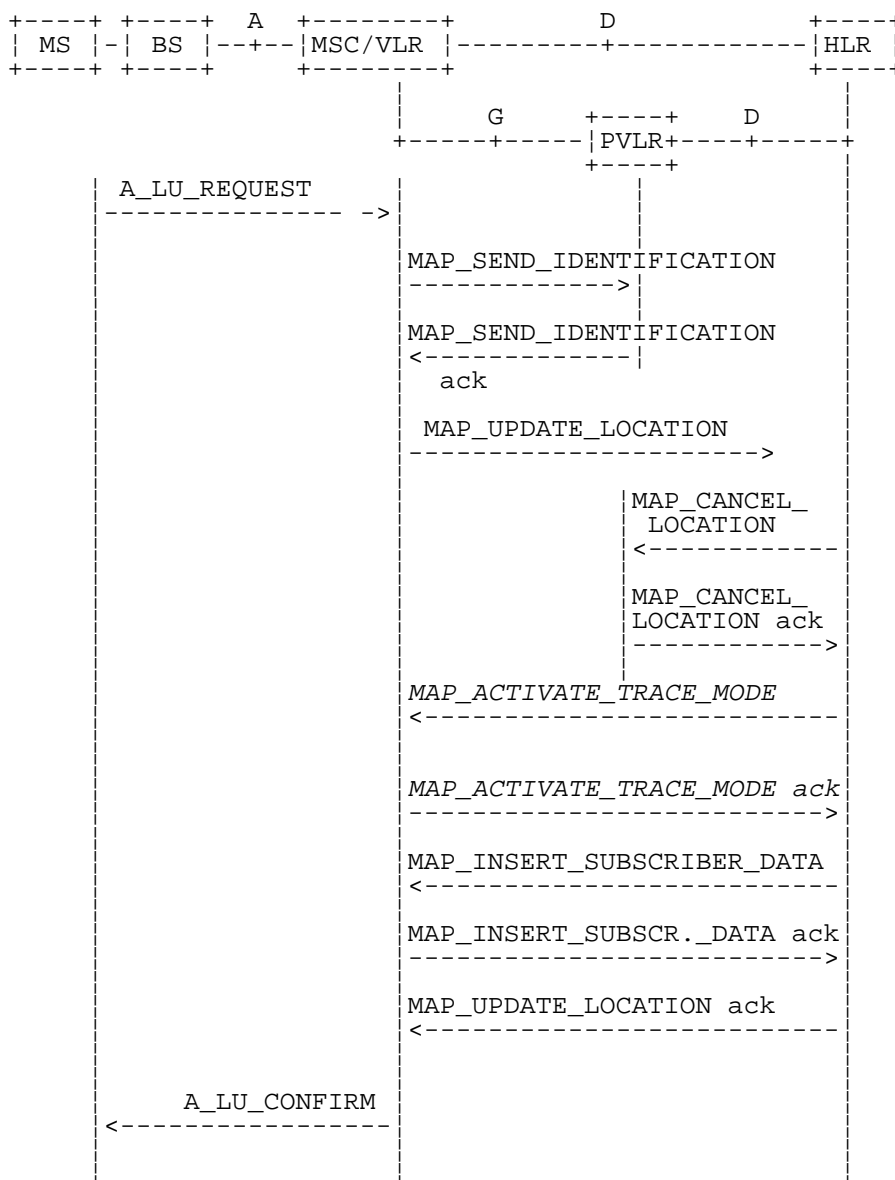
The message flow for successful GPRS Attach/ RA update procedure combined with a successful VLR location updating (Gs interface installed) is shown in figure 19.1.1/5.

The following MAP services are invoked by the location update procedure:

MAP_UPDATE_LOCATION_AREA (see subclause 8.1);(**)
MAP_UPDATE_LOCATION (see subclause 8.1);(**)
MAP_UPDATE_GPRS_LOCATION (see subclause 8.1) (*);
MAP_CANCEL_LOCATION (see subclause 8.1);
MAP_INSERT_SUBSCRIBER_DATA (see subclause 8.8);
MAP_SEND_IDENTIFICATION (see subclause 8.1) (**);
MAP_PROVIDE_IMSI (see subclause 8.9) (**);
MAP_AUTHENTICATE (see subclause 8.5) (**);
MAP_SET_CIPHERING_MODE (see subclause 8.6) (**);
MAP_FORWARD_NEW_TMSI (see subclause 8.9) (**);
MAP_CHECK_IMEI (see subclause 8.7) (**);
MAP_ACTIVATE_TRACE_MODE (see subclause 9.2);
MAP_TRACE_SUBSCRIBER_ACTIVITY (see subclause 9.2) (**).

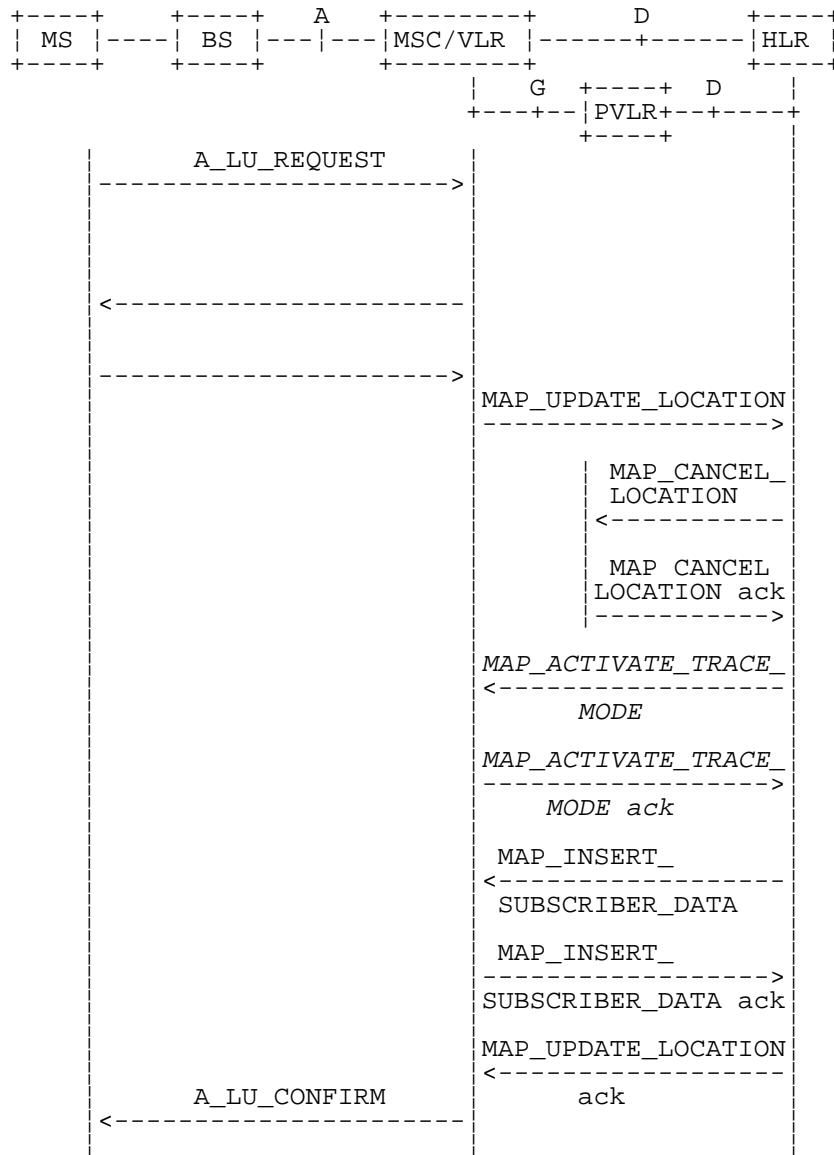
(*): only used in SGSN and HLR for GPRS

(**): not used in SGSN



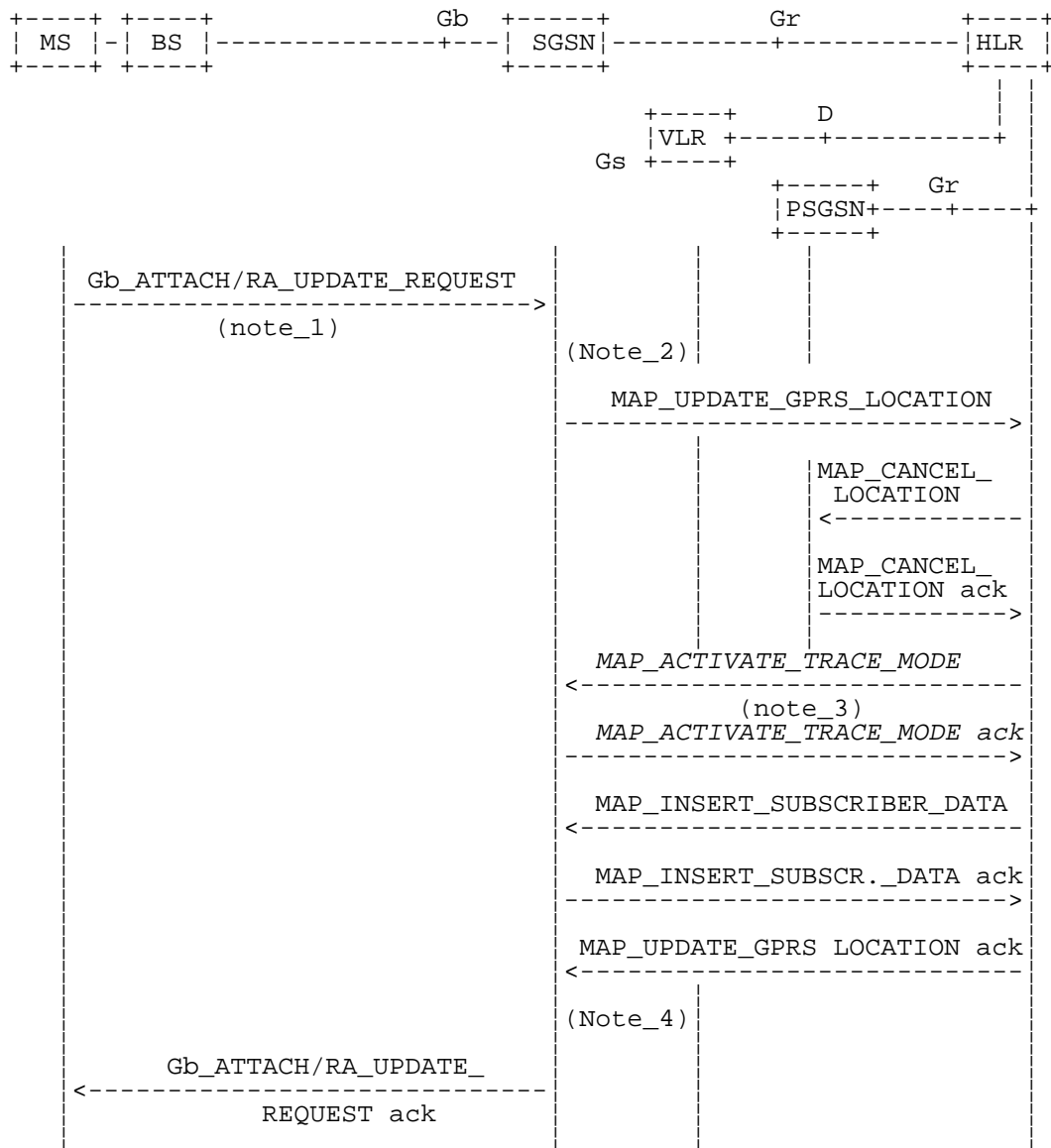
NOTE: Procedures shown in italics are optional.

Figure 19.1.1/2: Interface and services for location updating when changing the VLR area



NOTE: Procedures shown in italics are optional.

Figure 19.1.1/3: Interface and services for location updating involving both a VLR and an HLR, when IMSI can not be retrieved from the previous VLR



PSGSN = Previous SGSN

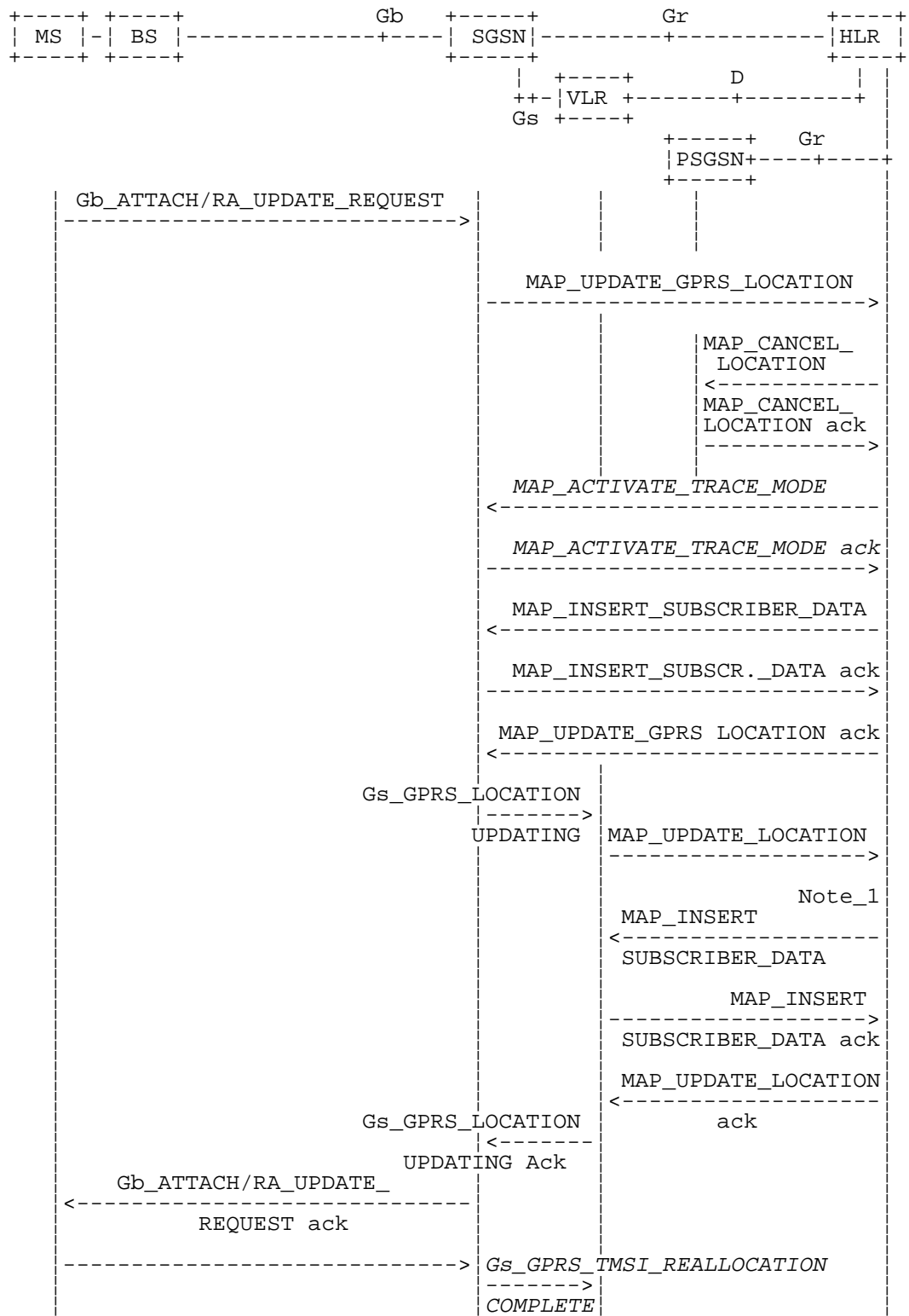
NOTE 1: For details of the procedure on the radio path, see GSM 08.18. The services shown in dotted lines indicate the trigger provided by the signalling on the radio path, and the signalling triggered on the radio path.

NOTE 2: For security functions (authentication, ciphering, IMEI check) triggering refer to GSM 03.60. MAP processes invoked for those procedures are described in section 25.

NOTE 3: Optional services are printed in *italics*.

NOTE 4: Refer to GSM 03.60 for termination of the procedure and triggering of the signalling on the Gb interface.

Figure 19.1.1/4: Interface and services for GPRS location updating (Gs-interface not installed)



NOTE: The optional procedures in figure 19.1.1/14 apply here respectively. For details of the procedure on the Gs-interface, see GSM 09.18.

NOTE 1: Location Cancellation procedure toward the old VLR and optional tracing activation toward the new VLR are not represented on this figure.

Figure 19.1.1/5: Interface and services for GPRS location updating (Gs-interface installed)

19.1.1.3 Detailed procedure in the VLR

Updating request via the Gs interface (optional for GPRS)

If Gs-interface is installed, the VLR may receive the Gs_GPRS_LOCATION_UPDATING_Request message from the SGSN for triggering an IMSI Attach or Location Updating procedure (see GSM 03.60 and 09.18).

Figure 19.1.1/16 shows the process for handling this Gs interface message.

The process specific macro

« GPRS_Location_Update_Completion_VLR » for optional initiation of TMSI reallocation as for acknowledgement of the Gs_GPRS_LOCATION_UPDATING_Request message (see figure 19.1.1/17),

and the optional process specific macro

« VLR_Update_GPRS_HLR » to update the HLR and download subscriber data from there (see figure 19.1.1/18), are invoked by this process.

On receipt of the Gs_GPRS_LOCATION_UPDATING_Request message, 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. The indicator "Confirmed by Radio Contact" is set to "Confirmed" and the location information held in the register is updated. If no VLR/SGSN association exists it is created (storage of SGSN address received) otherwise it is updated.

If the HLR is to be updated, the VLR_Update_GPRS_HLR macro described below is performed, with one of the following results (see sheet 2 of figure 19.1.1/18):

- OK, if HLR updating has been completed successfully. The response will contain the HLR number as parameter. Next, the GPRS_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 appropriate error (see GSM 09.18) is sent to the SGSN in the Gs_GPRS_LOCATION_UPDATING Reject. The Subscriber Data are deleted in the VLR.
- if Roaming Not Allowed was qualified by the parameter Operator Determined Barring, the appropriate error (see GSM 09.18) is sent in the Gs_GPRS_LOCATION_UPDATING Reject to the SGSN. 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 appropriate error (see GSM 09.18) is sent in the Gs_GPRS_LOCATION_UPDATING Reject.
- Procedure error, if there occurs some other error during HLR updating (e.g. abort of the connection to HLR). In this case the appropriate error (see GSM 09.18) is sent in the Gs_GPRS_LOCATION_UPDATING Reject.

The macro GPRS 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 19.1.1/17):

- if the target LA is not allowed for the subscriber due to national roaming restrictions, the appropriate error (see GSM 09.18) is sent in the Gs_GPRS_LOCATION_UPDATING Reject towards the SGSN.

The subscriber data are not deleted from VLR, to avoid unnecessary HLR updating when roaming into other LAs of the same MSC/VLR. 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 appropriate error (see GSM 09.18) is returned to the SGSN in the Gs_GPRS_LOCATION_UPDATING Reject.

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 19.1.1.7). Thereafter, the VLR checks whether TMSI reallocation is required.
 - if so, the VLR sends the TMSI within the Gs_GPRS_LOCATION_UPDATING Accept message and Gs_GPRS_TMSI_REALLOCATION_Complete is expected.
- if TMSI reallocation is not required, the VLR sends the Gs_GPRS_LOCATION_UPDATING Accept message to the SGSN.

The macro VLR Update GPRS HLR

This macro is invoked by the VLR process for location updating (see GSM 03.60). 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 19.1.1/18) 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 25.1), the VLR will terminate the procedure indicating procedure error. If the HLR indicates version Vr protocol to be used, the VLR will revert to the version Vr procedure concerning the dialogue with the HLR, with outcomes as for the current MAP version procedure.

If the HLR accepts the dialogue, the HLR will respond with:

- a MAP_INSERT_SUBSCRIBER_DATA indication, handled by the macro Insert_Sub_Data_VLR defined in subclause 25.7;

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 25.9;
- a MAP_FORWARD_CHECK_SS_INDICATION_ind. This indication will not be relayed to the SGSN.
- 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.

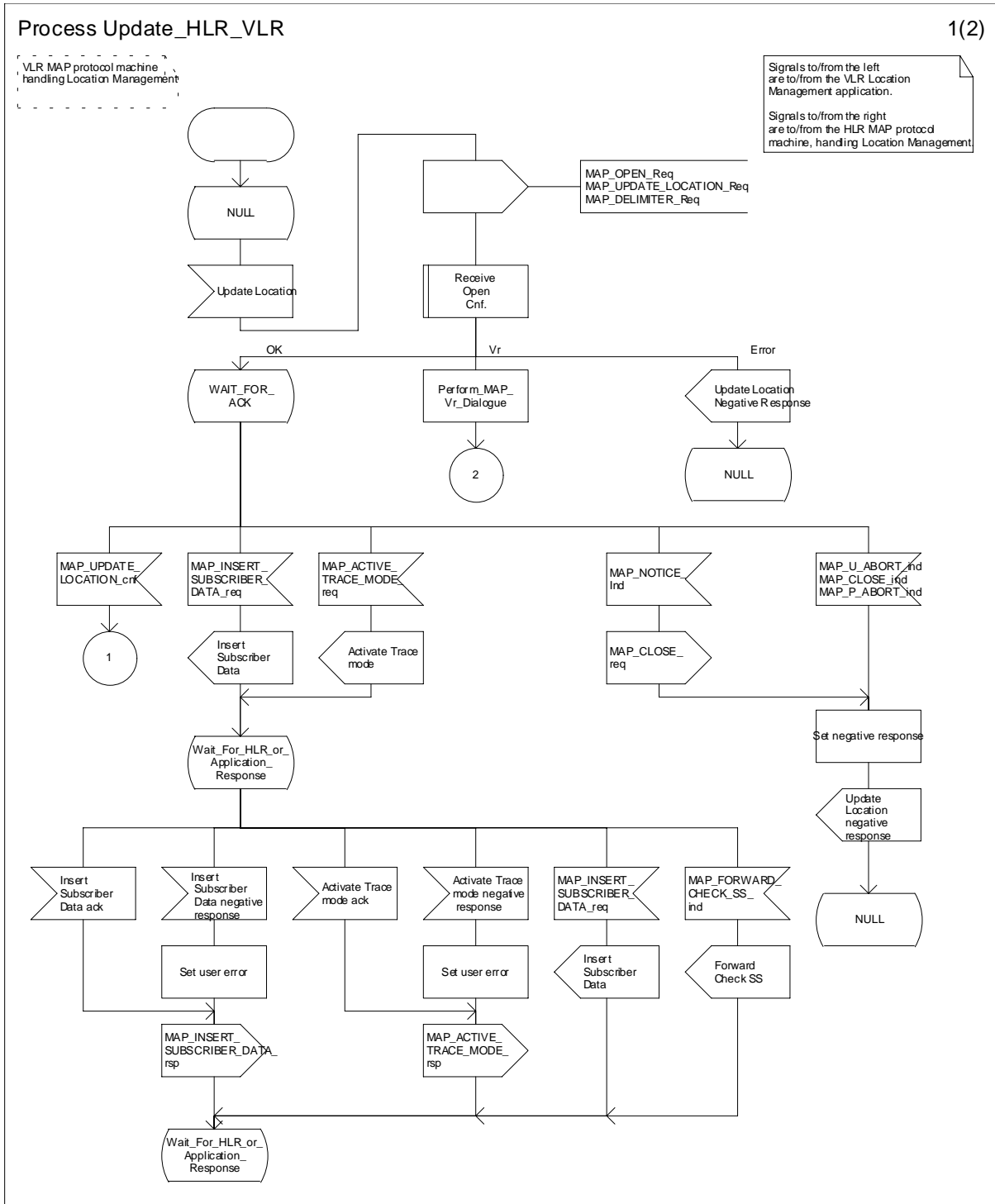


Figure 19.1.1/6 (sheet 1 of 2): Process Update_HLR_VLR

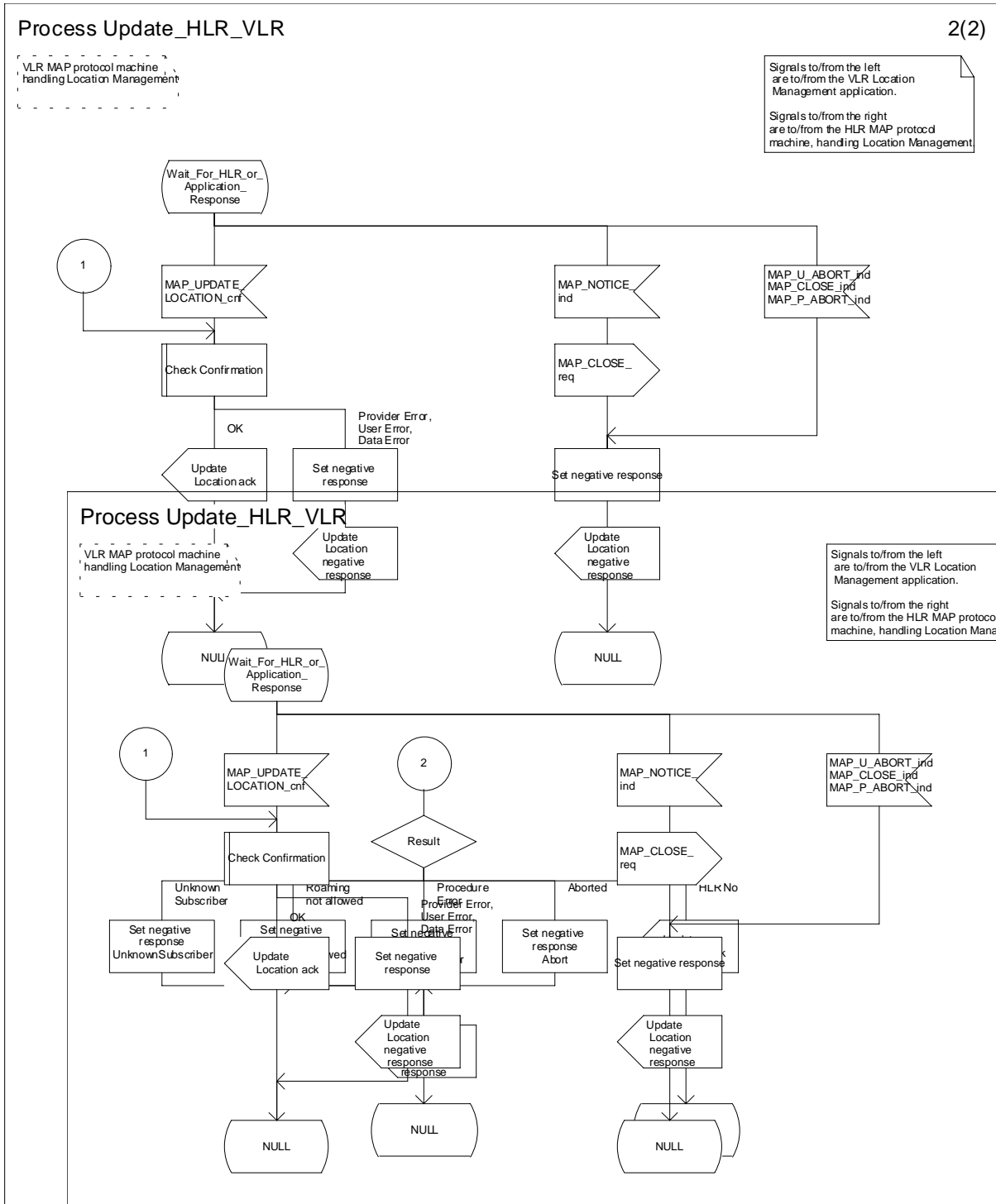
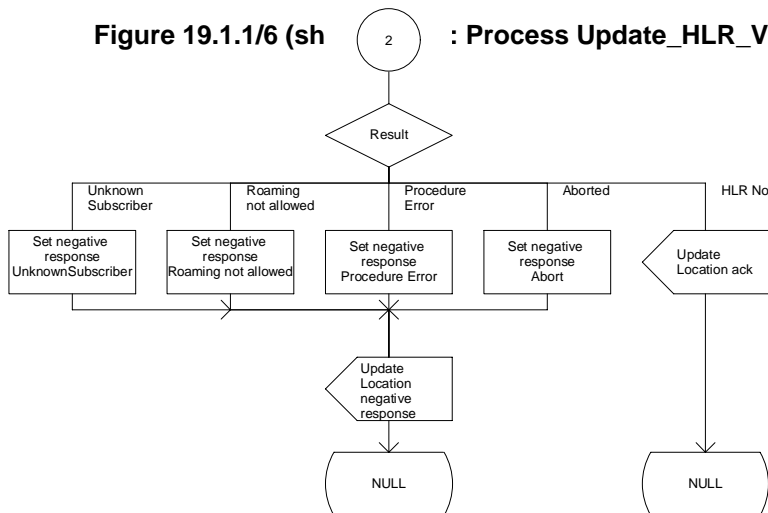


Figure 19.1.1/6 (sh 2 : Process Update_HLR_VLR



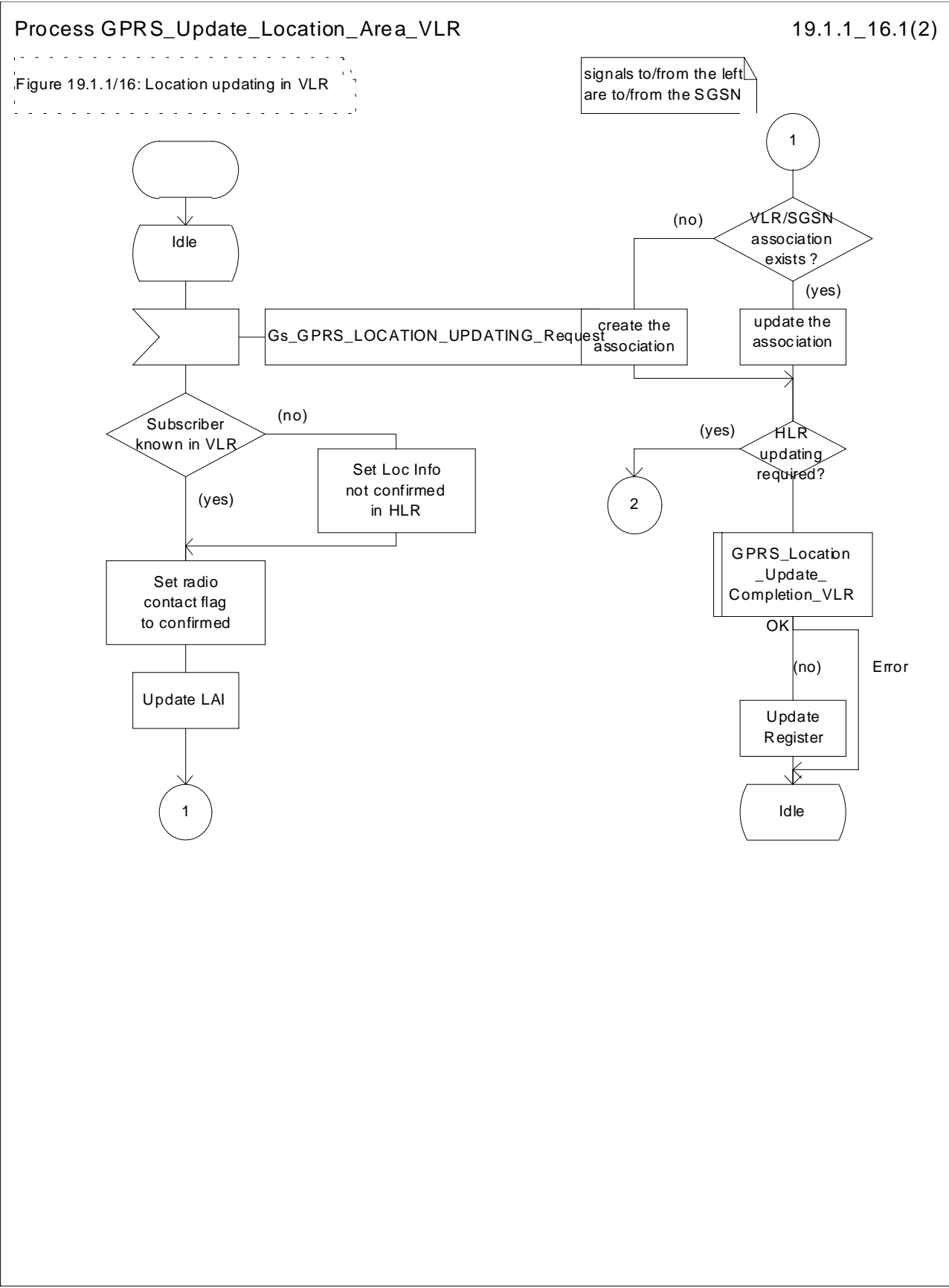


Figure 19.1.1/16 (sheet 1 of 2): Process GPRS_Update_Location_Area_VLR

Process GPRS_Update_Location_Area_VLR

19.1.1_16.2(2)

Figure 19.1.1/16: Location updating in VLR

Signals to/from the left are to/from the SGSN

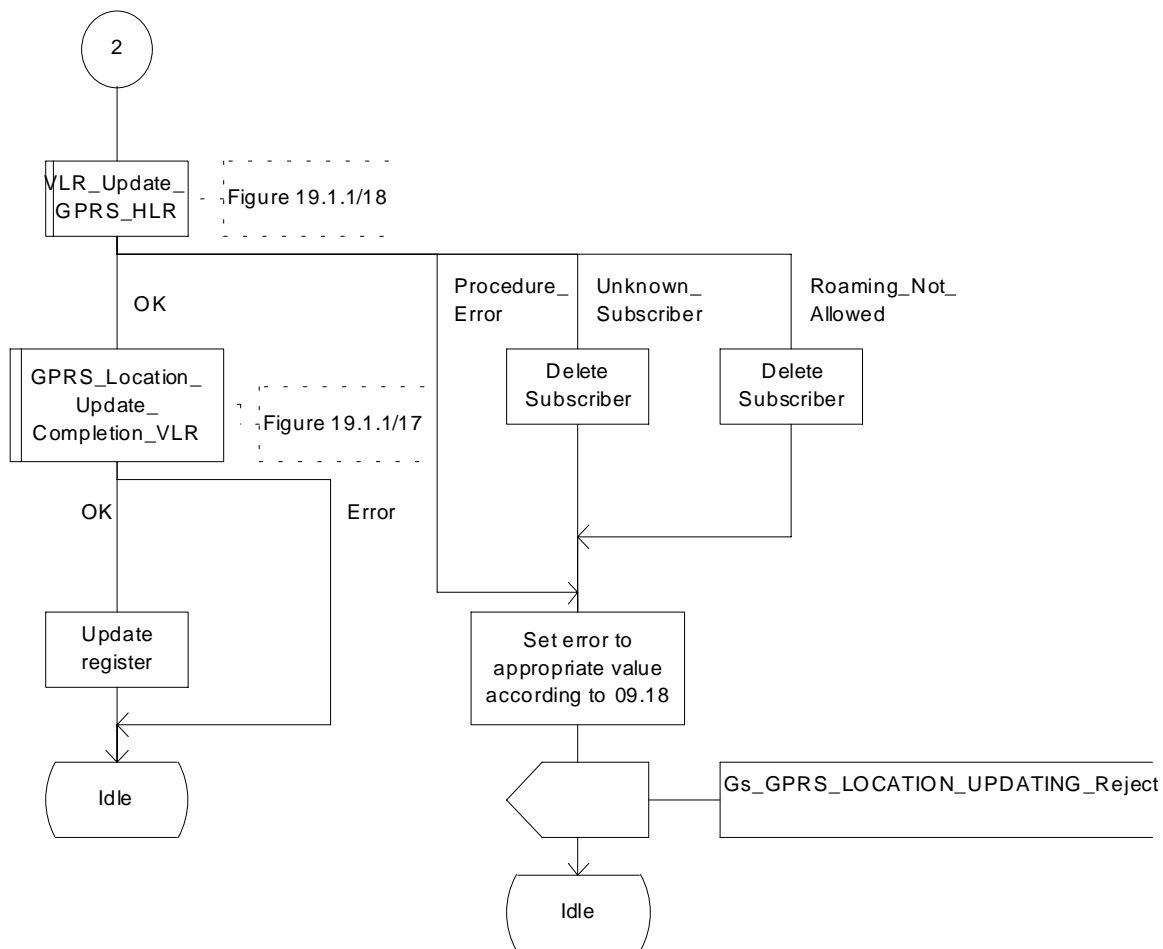


Figure 19.1.1/16 (sheet 2 of 2): Process GPRS_Update_Location_Area_VLR

Macrodefinition GPRS_Location_Update_Completion_VLR

19.1.1_17(1)

Figure 19.1.1/17:
'Location updating in VLR for GPRS:
closing sequence

Signals to/from the left
are to/from the SGSN

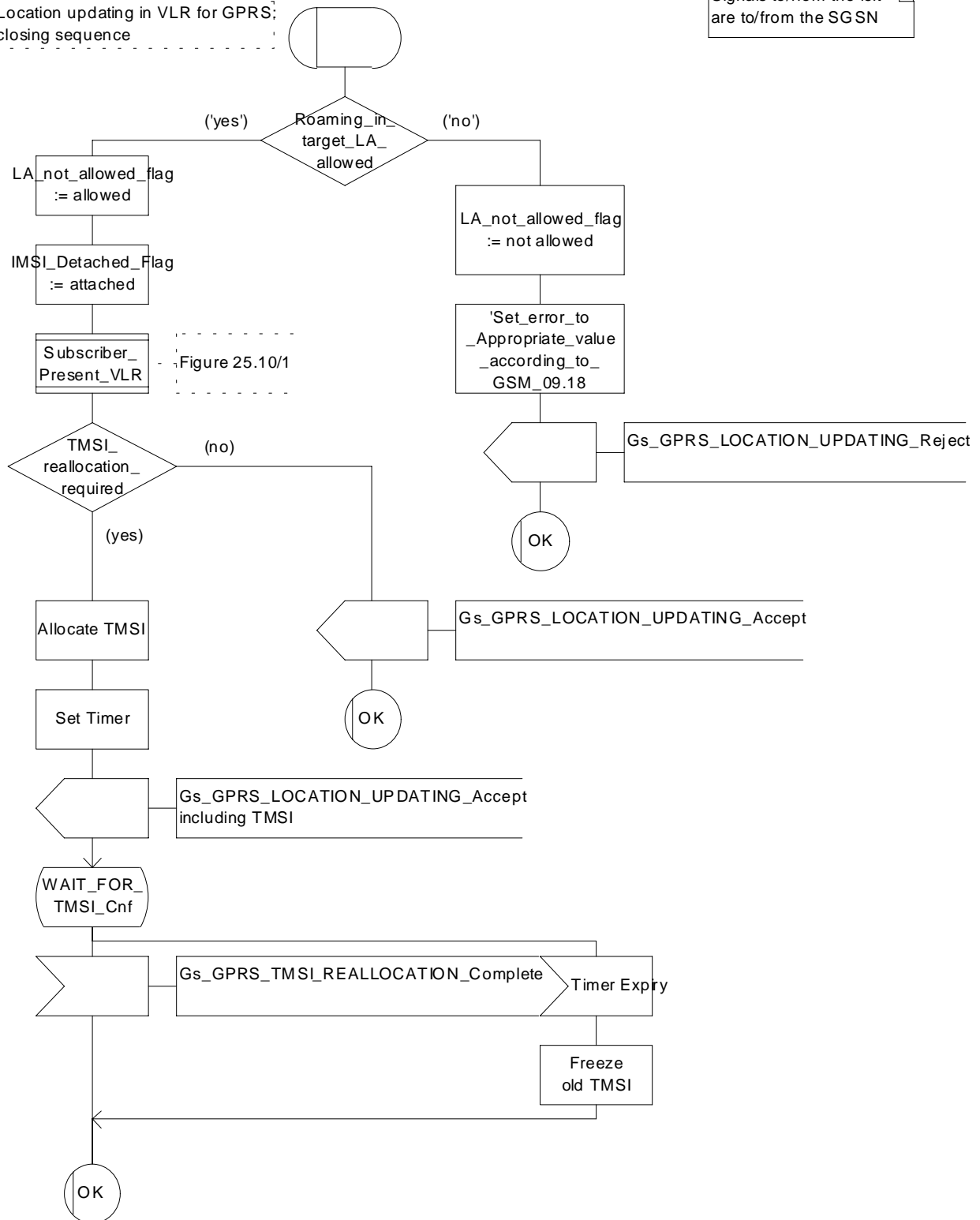


Figure 19.1.1/17: Macro GPRS_Location_Update_Completion_VLR

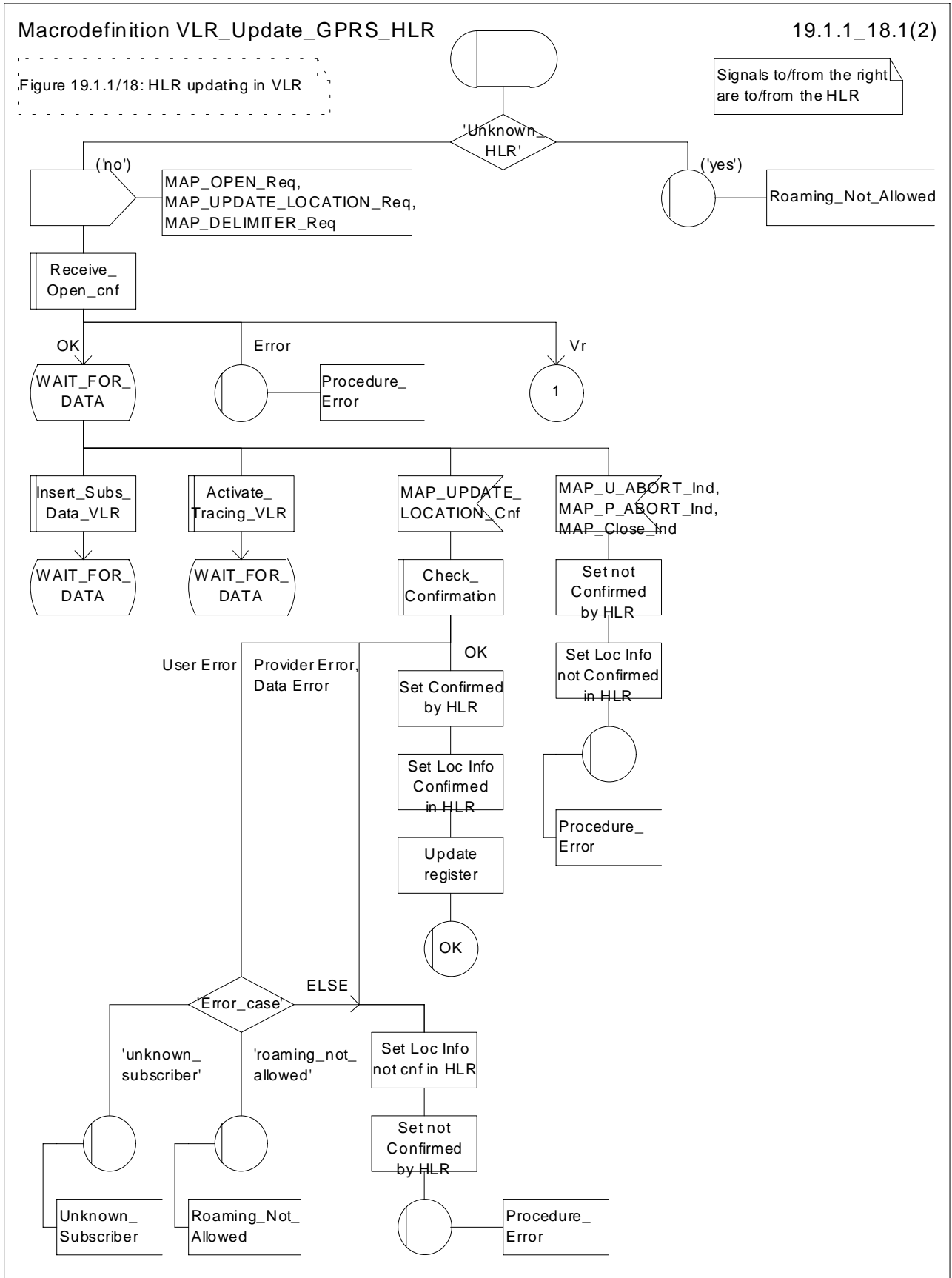


Figure 19.1.1/18 (sheet 1 of 2): Macro VLR_Update_GPRS_HLR

Macrodefinition VLR_Update_GPRS_HLR

19.1.1_18.2(2)

Figure 19.1.1/18: HLR updating in VLR

Signals to/from the right are to/from the HLR

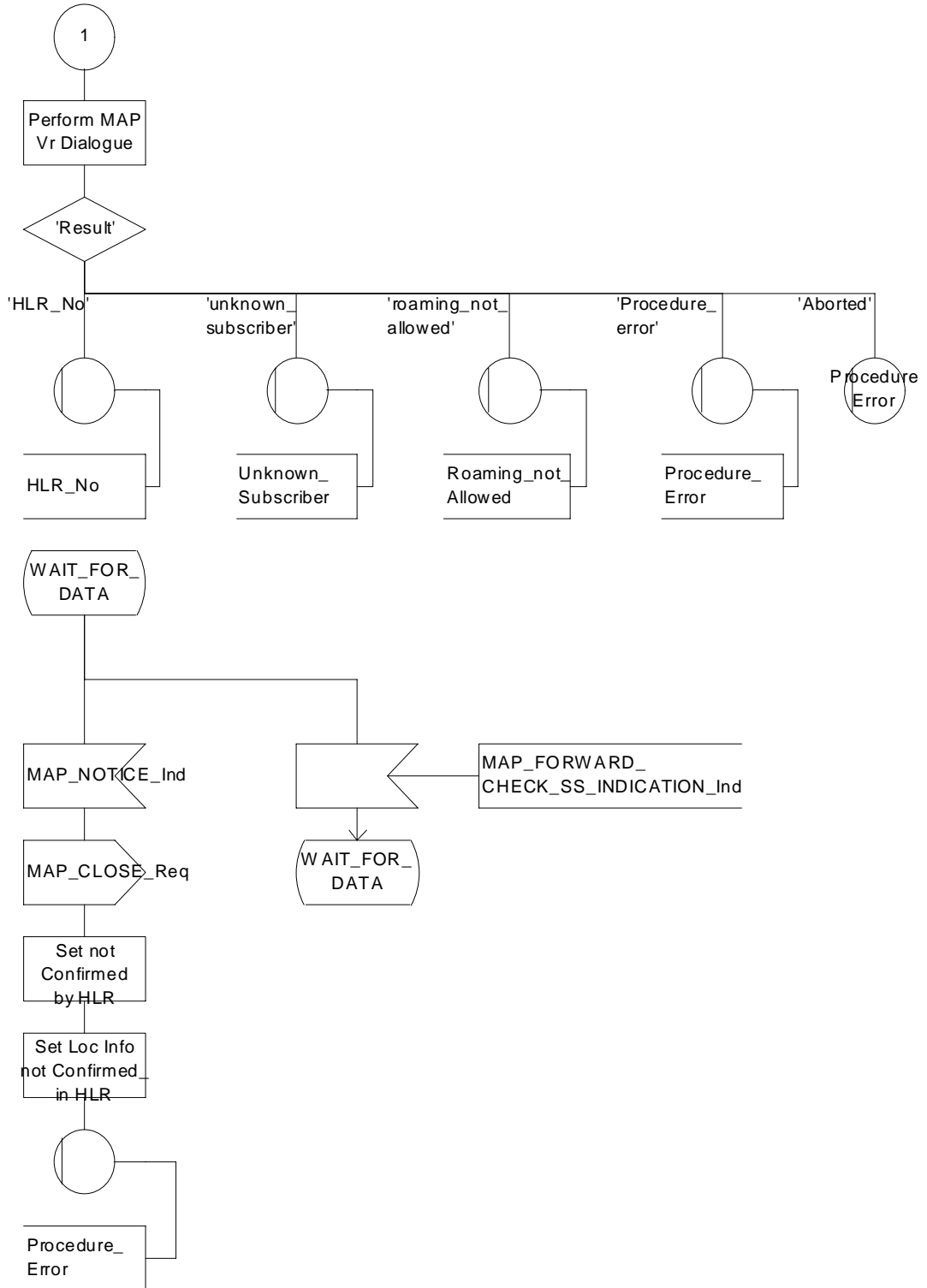


Figure 19.1.1/18 (sheet 2 of 2): Macro VLR_Update_GPRS_HLR

19.1.1.4 Detailed procedure in the HLR

Sheet 1: The procedure Super_Charged_Cancel_Location_HLR is specific to Super-Charger; it is specified in TS 23.116 [110]. If the previous SGSN and the originating HLR support the Super-Charger functionality, processing continues from the "Yes" exit of the test "Result=Pass?".

Sheet 2: The procedure Super_Charged_Location_Updating_HLR is specific to Super-Charger; it is specified in TS 23.116 [110]. If subscription data needs to be sent to the SGSN, processing continues from the "No" exit of the test "Result=Pass?".

When addressed by the SGSN, the following macros are used by the process Update_GPRS_Location_HLR:

- Receive_Open_indication, defined in subclause 25.1;
- Check_indication, defined in subclause 25.2;
- Insert_Subs_Data_In_SGSN_Framed_HLR, described in subclause 19.4.x;
- Control_Tracing_HLR_with_SGSN, described in subclause 25.9;

and the processes Cancel_Location_HLR (see subclause 19.1.2) and Subscriber_Present_HLR (see subclause 19.1.1.7) are invoked.

The location updating process in the HLR is activated by receipt of a MAP_UPDATE_GPRS_LOCATION indication (see figure 19.1.1/19):

- if there is a parameter problem in the indication, the error Unexpected Data Value is returned in the MAP_UPDATE_GPRS_LOCATION response (see Check_indication macro defined in subclause 25.2); if the subscriber is not known in the HLR, the error Unknown Subscriber (with diagnostic value set to "Imsi Unknown") is returned in the response. In either case the process terminates;
- if Network Access Mode is set to "non-GPRS only" the error Unknown Subscriber (with diagnostic value set to "Gprs Subscription Unknown") is returned in the response. The process terminates;
- tracing shall be set to deactivate in the SGSN.
- if the SGSN number received in the MAP_UPDATE_GPRS_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 SGSN (see subclause 19.1.2).

The next action will be to check whether the subscriber is allowed to roam into the PLMN indicated by the SGSN Number given in the MAP_UPDATE_GPRS_LOCATION indication:

- if the subscriber is not allowed to roam into the PLMN, the error Roaming not Allowed with cause PLMN Roaming Not Allowed or 'Operator determined Barring', depending on the case, is returned in the MAP_UPDATE_GPRS_LOCATION response, and the routing information stored (SGSN number) is deleted (deregistration);
- otherwise the HLR database will be updated with information received in the indication. The HLR sets the "MS purged for GPRS" flag to False and checks whether tracing is required for that subscriber. This is handled by the macro Control_Tracing_HLR-with_SGSN described in subclause 25.9.

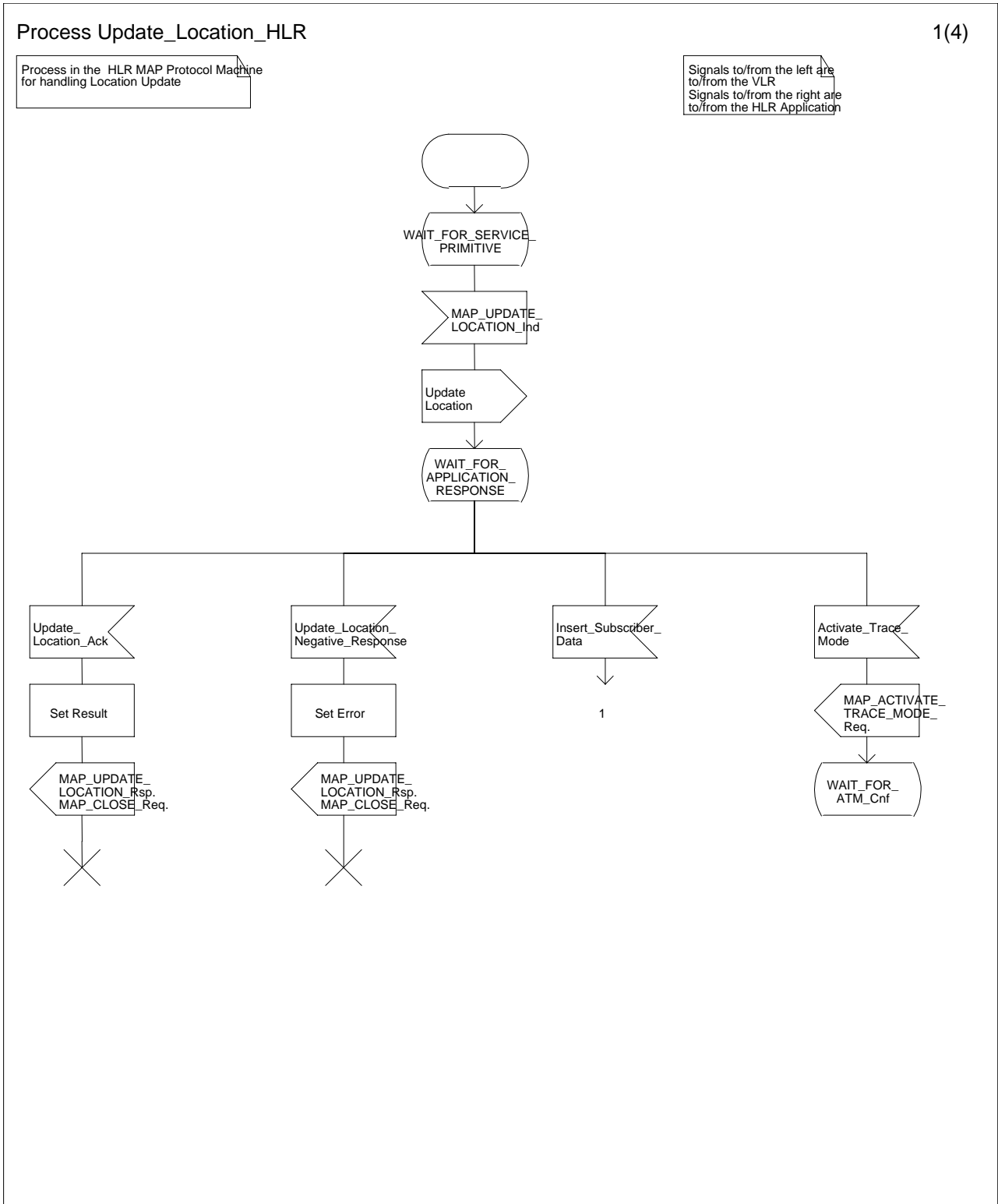
Thereafter, the macro Insert_Subs_Data_In_SGSN_Framed_HLR described in subclause 19.4.x is invoked. The outcome of this macro may be:

- aborted, in which case the process terminates;
- error, in which case the error System Failure is returned in the MAP_UPDATE_GPRS_LOCATION response and the process terminates;
- OK, indicating successful outcome of downloading the subscriber data to the SGSN.

The SUBSCRIBER_PRESENT_HLR process is then started to alert the Short Message Service Centre, if required (see subclause 19.1.7).

Finally the HLR number is returned in the MAP_UPDATE_GPRS_LOCATION response.

In all cases where the HLR sends a MAP_UPDATE_GPRS_LOCATION response to the SGSN, the dialogue towards the SGSN is terminated by a MAP_CLOSE request with parameter Release Method indicating Normal Release.



Process Update_Location_HLR

1(4)

Process in the HLR MAP Protocol Machine, for handling Location Update

Signals to/from the left are to/from the VLR
Signals to/from the right are to/from the HLR Application

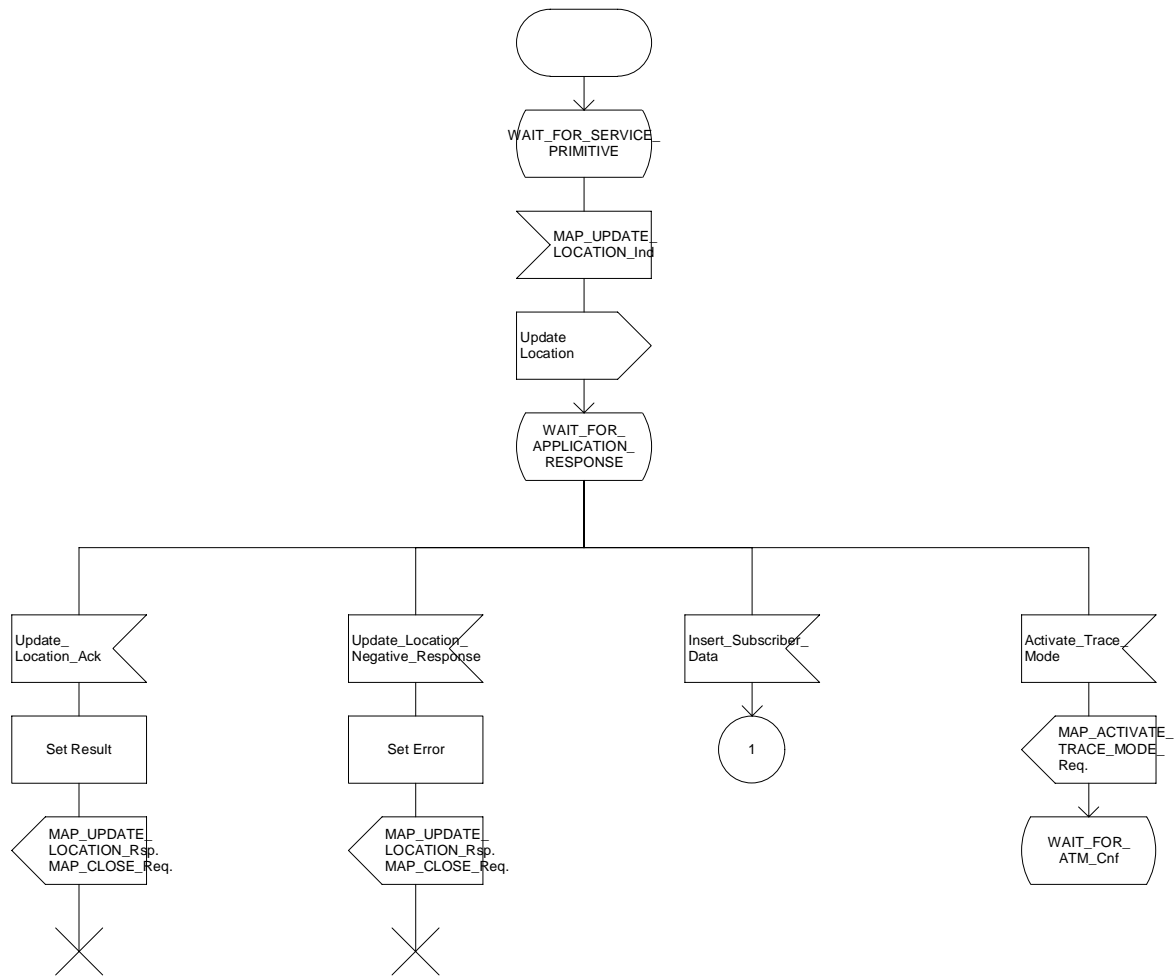


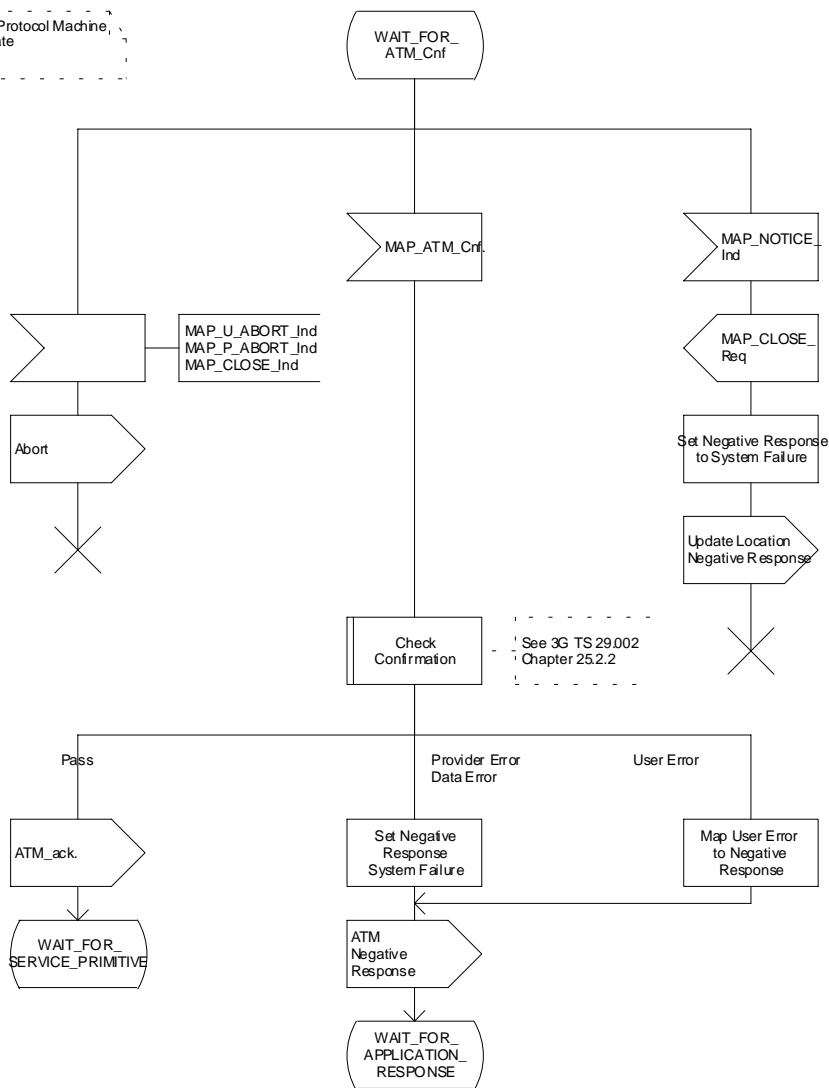
Figure 19.1.1/9 (sheet 1 of 4): Process Update_Location_HLR

Process Update_Location_HLR

2(4)

Process in the HLR MAP Protocol Machine, for handling Location Update

Signals to/from the left are to/from the VLR
Signals to/from the right are to/from the HLR Application



Process Update_Location_HLR

2(4)

Process in the HLR MAP Protocol Machine for handling Location Update

Signals to/from the left are to/from the VLR
Signals to/from the right are to/from the HLR Application

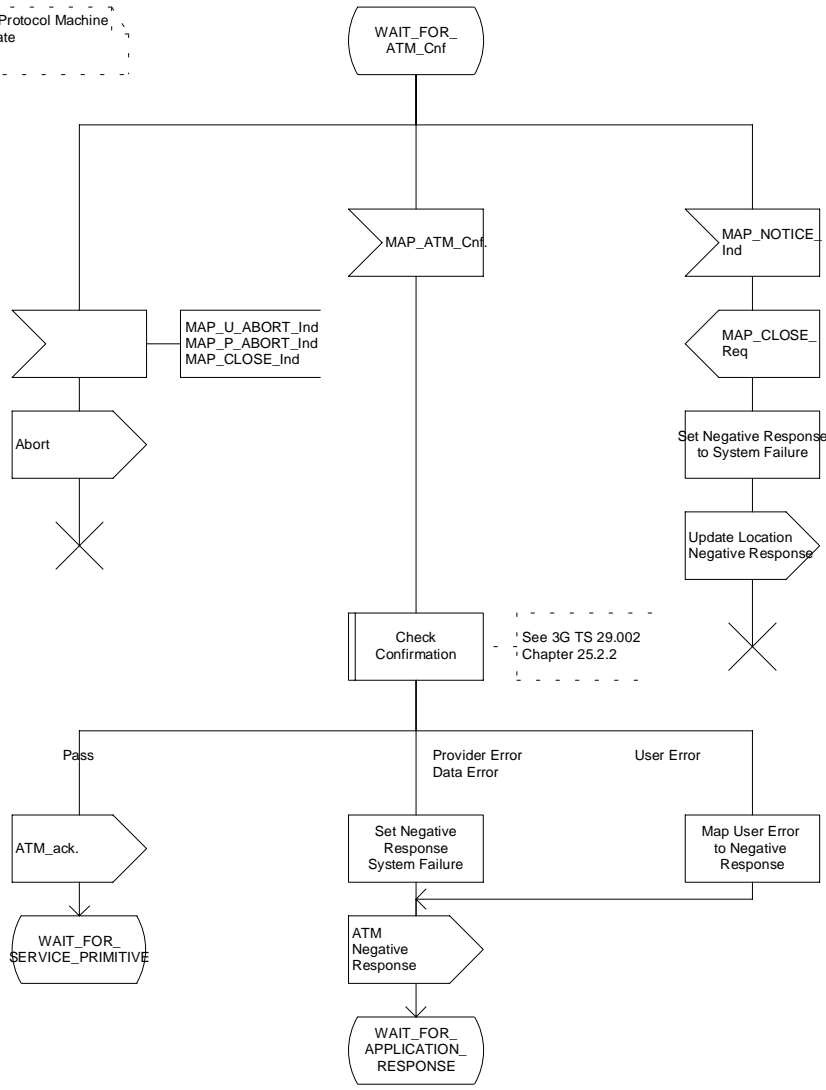


Figure 19.1.1/9 (sheet 2 of 4): Process Update_Location_HLR

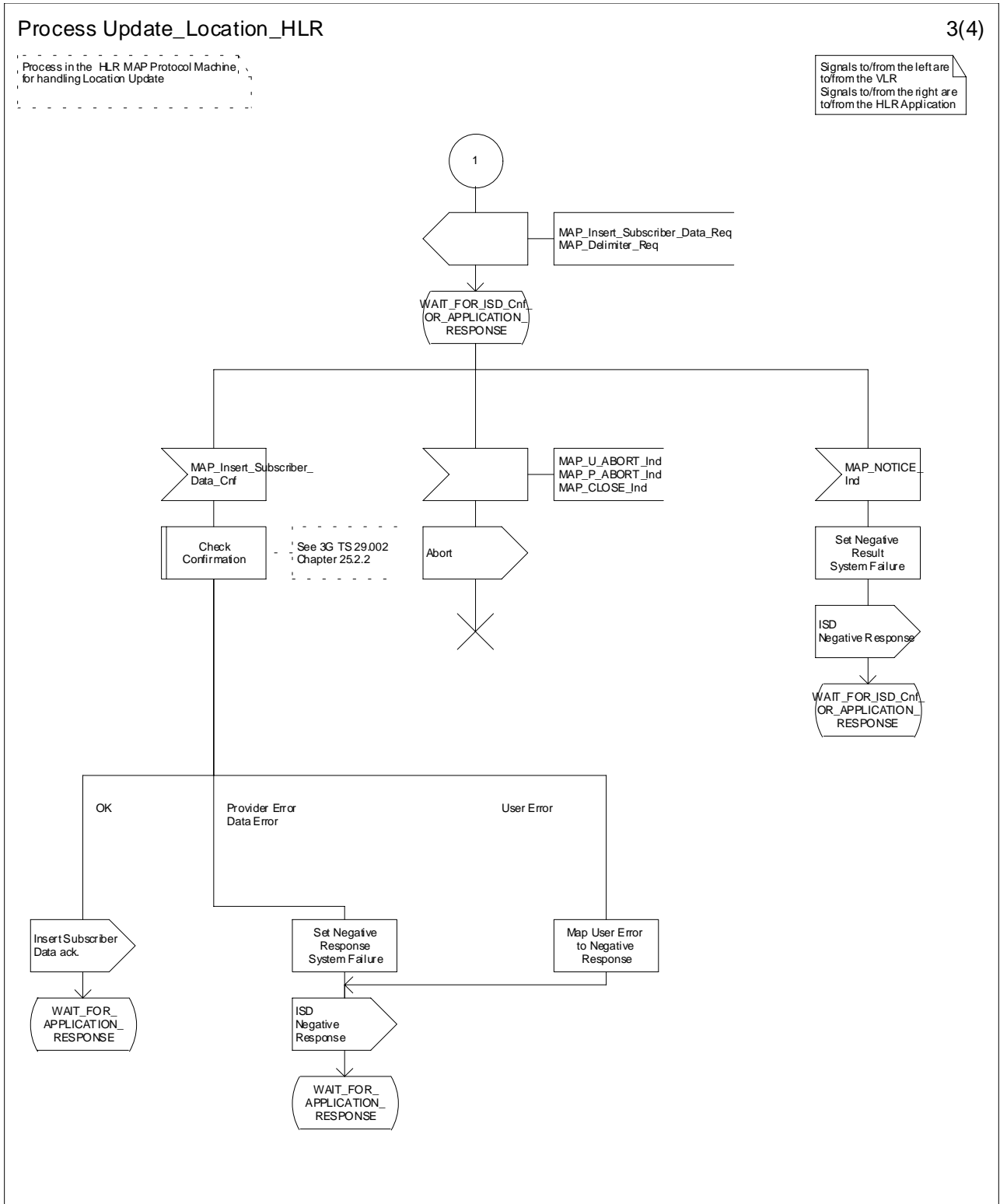


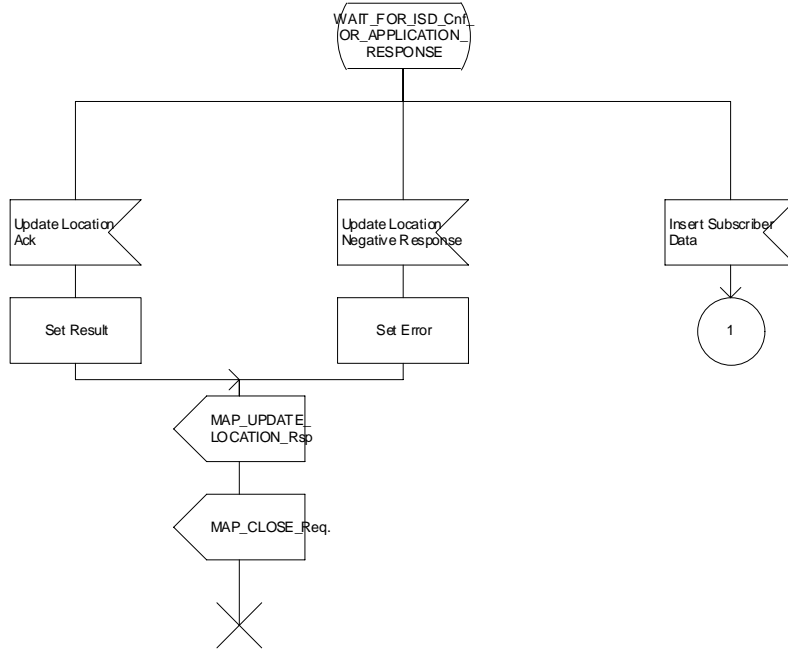
Figure 19.1.1/9 (sheet 3 of 4): Process Update_Location_HLR

Process Update_Location_HLR

4(4)

Process in the HLR MAP Protocol Machine, for handling Location Update

Signals to/from the left are to/from the VLR
Signals to/from the right are to/from the HLR Application



Process Update_Location_HLR

4(4)

Process in the HLR MAP Protocol Machine for handling Location Update

Signals to/from the left are to/from the VLR
Signals to/from the right are to/from the HLR Application

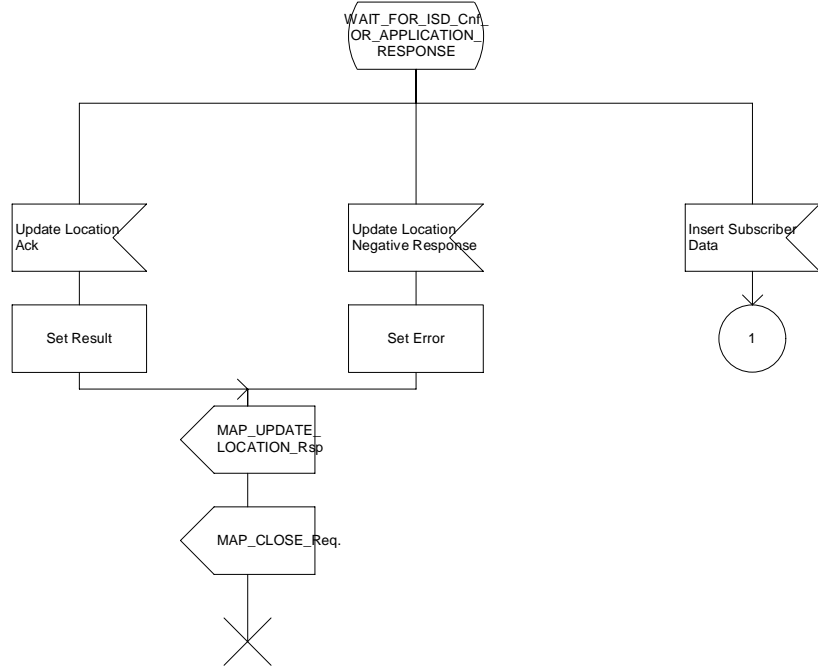


Figure 19.1.1/9 (sheet 4 of 4): Process Update_Location_HLR

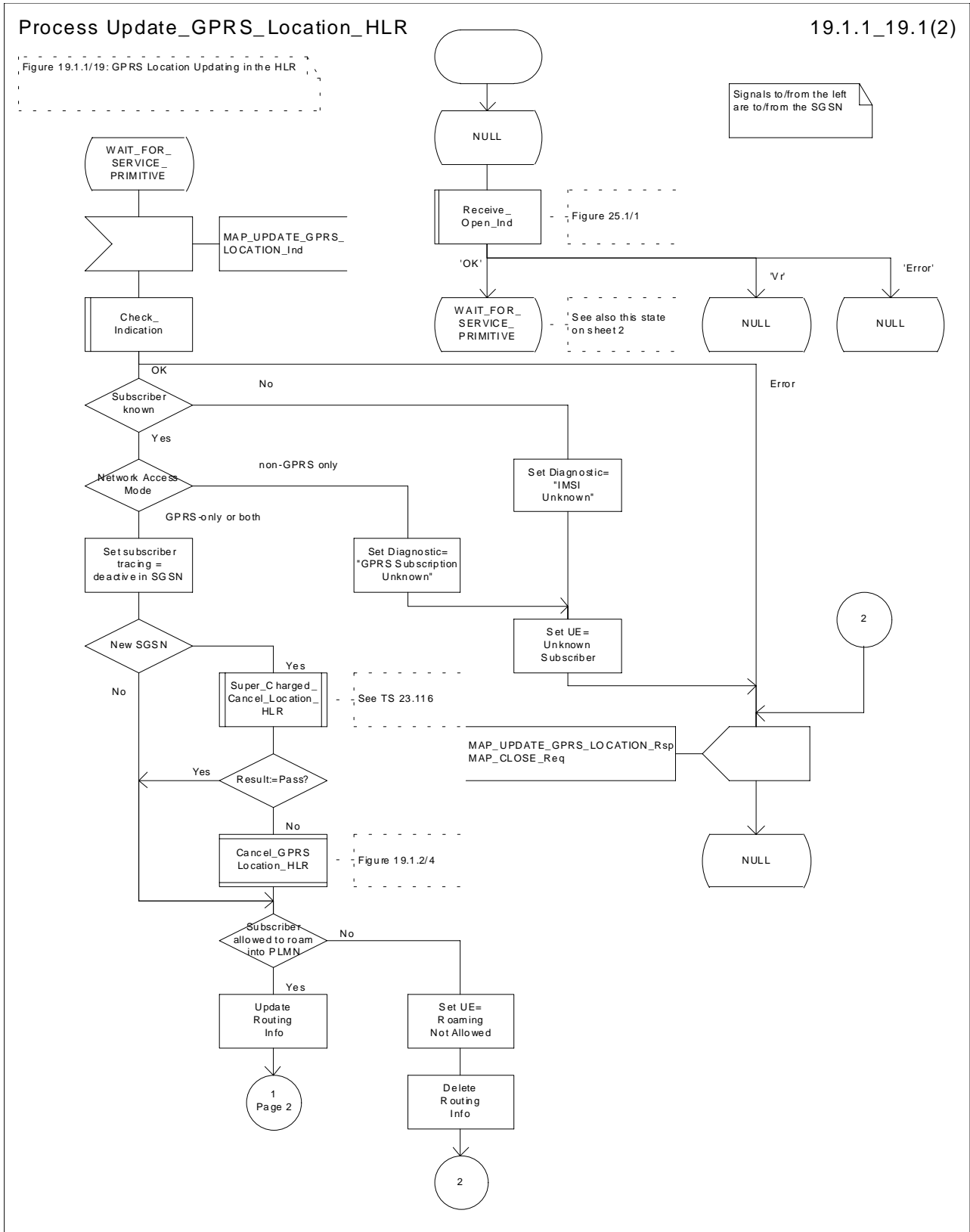


Figure 19.1.1/19 (sheet 1 of 2): Process Update_GPRS_Location_HLR

Process Update_GPRS_Location_HLR

19.1.1_19.2(2)

Figure 19.1.1/19: GPRS Location Updating in the HLR

Signals to/from the left are to/from the HLR

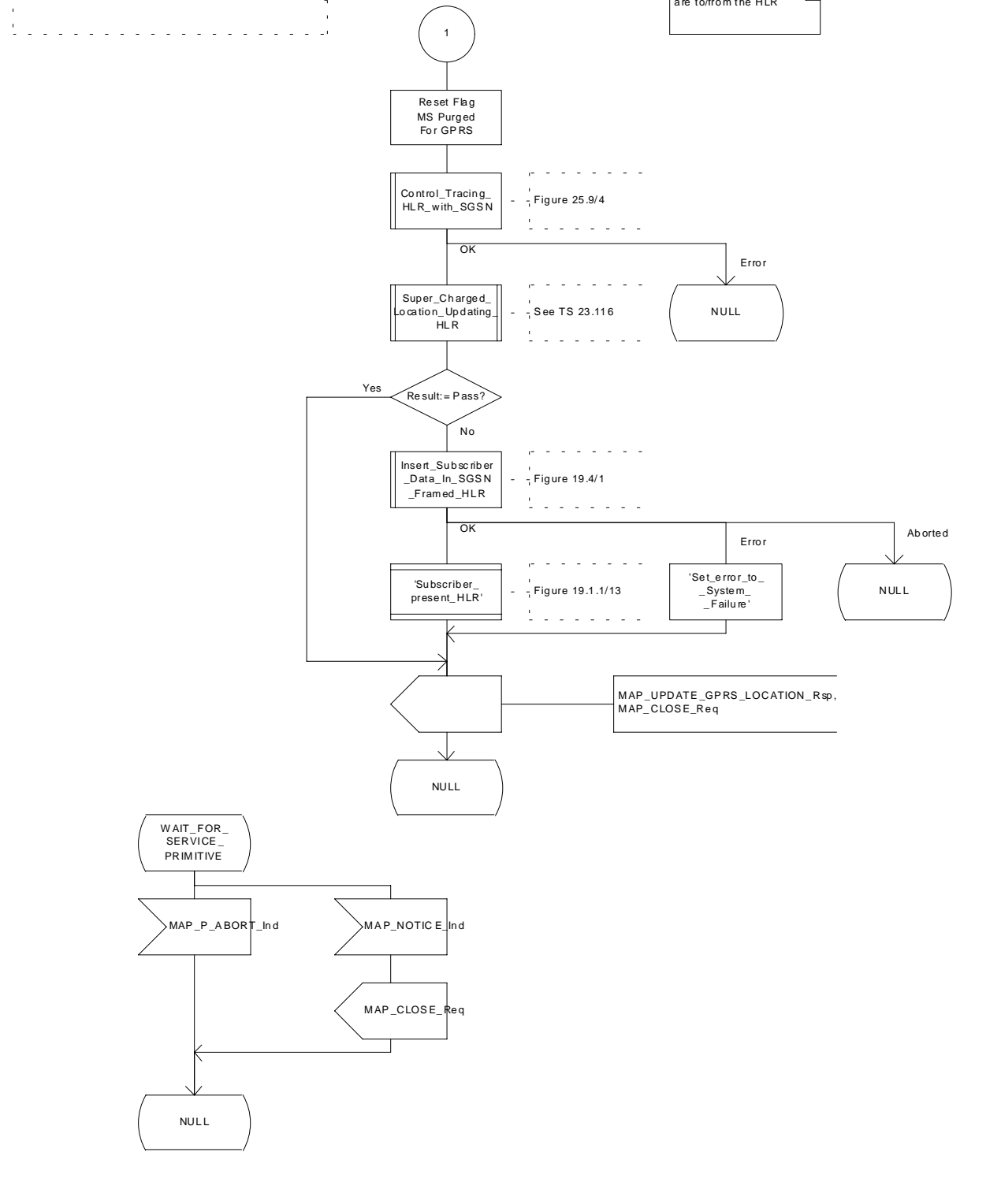
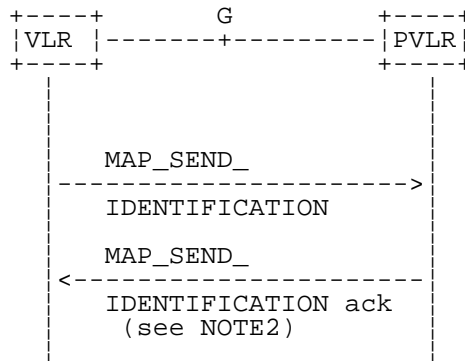


Figure 19.1.1/19 (sheet 2 of 2): Process Update_GPRS_Location_HLR

19.1.1.5 Send Identification

19.1.1.5.1 General

This service is invoked by the VLR when it receives Update location from the MSC 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 19.1.1.3), using the service described in subclause 8.1.4.



NOTE1: The service shown in dotted lines indicates the trigger provided by other MAP signalling.

NOTE2: Several MAP_SEND_IDENTIFICATION request/response may be used if message segmentation is required.

Figure 19.1.1/10: Interface and services for Send Identification

19.1.1.5.2 Detailed procedure in the VLR

The VLR procedure is part of the location area updating process described in subclause 19.1.1.X.

19.1.1.5.3 Detailed procedure in the PVLR

On receipt of a dialogue request for the Send Identification procedure, (see Receive_Open_Ind macro in subclause 25.1), the PVLR will:

- terminate the procedure in case of parameter problems;
- revert to the MAP version Vr procedure in case the VLR indicated version Vr 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 19.1.1/11), 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.

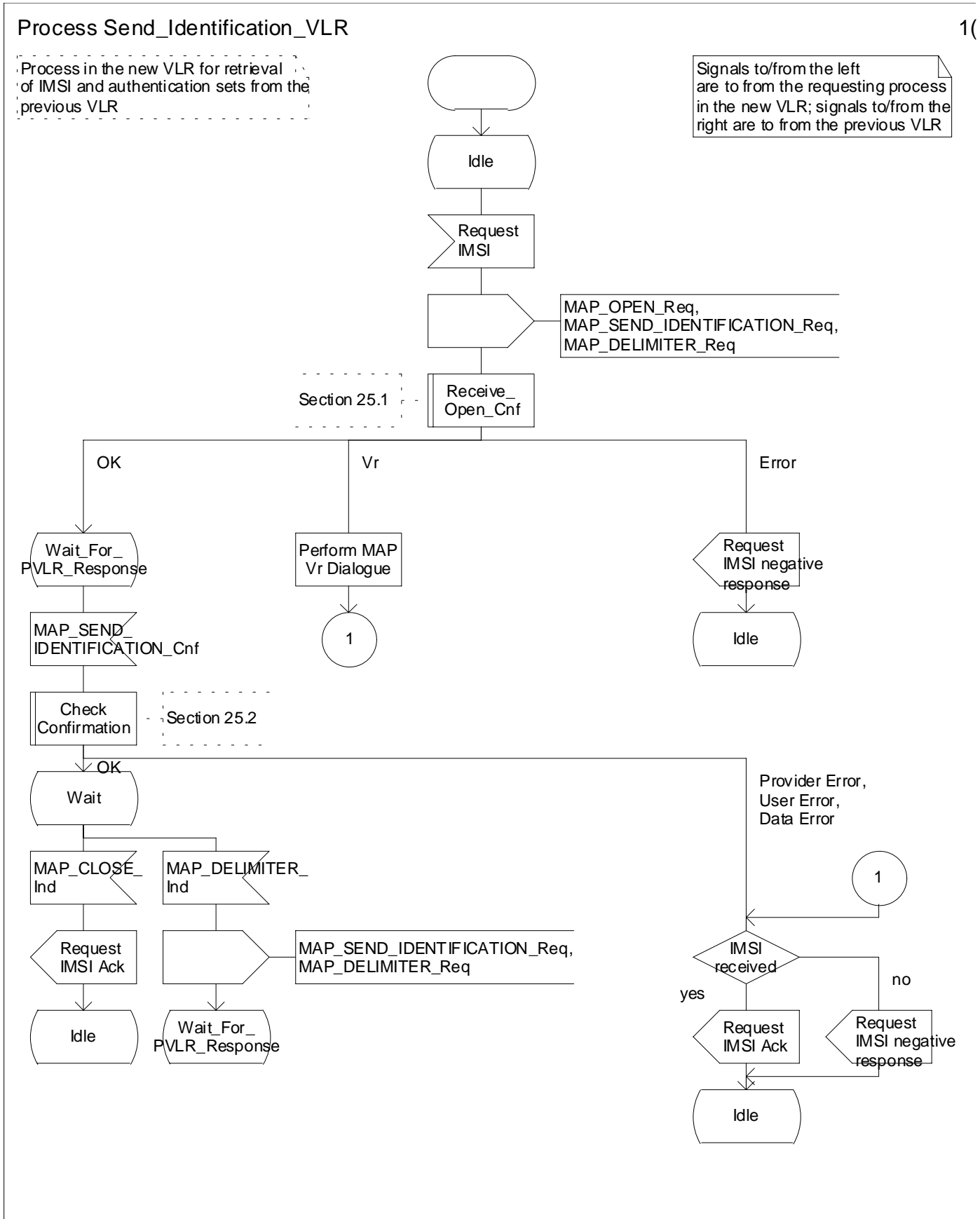


Figure 19.1.1/XX (sheet 1 of 2): Process Send_Identification_VLR

Process Send_Identification_VLR

Process in the new VLR for retrieval of IMSI and authentication sets from the previous VLR

Signals to/from the left are to from the requesting process in the new VLR; signals to/from the right are to from the previous VLR

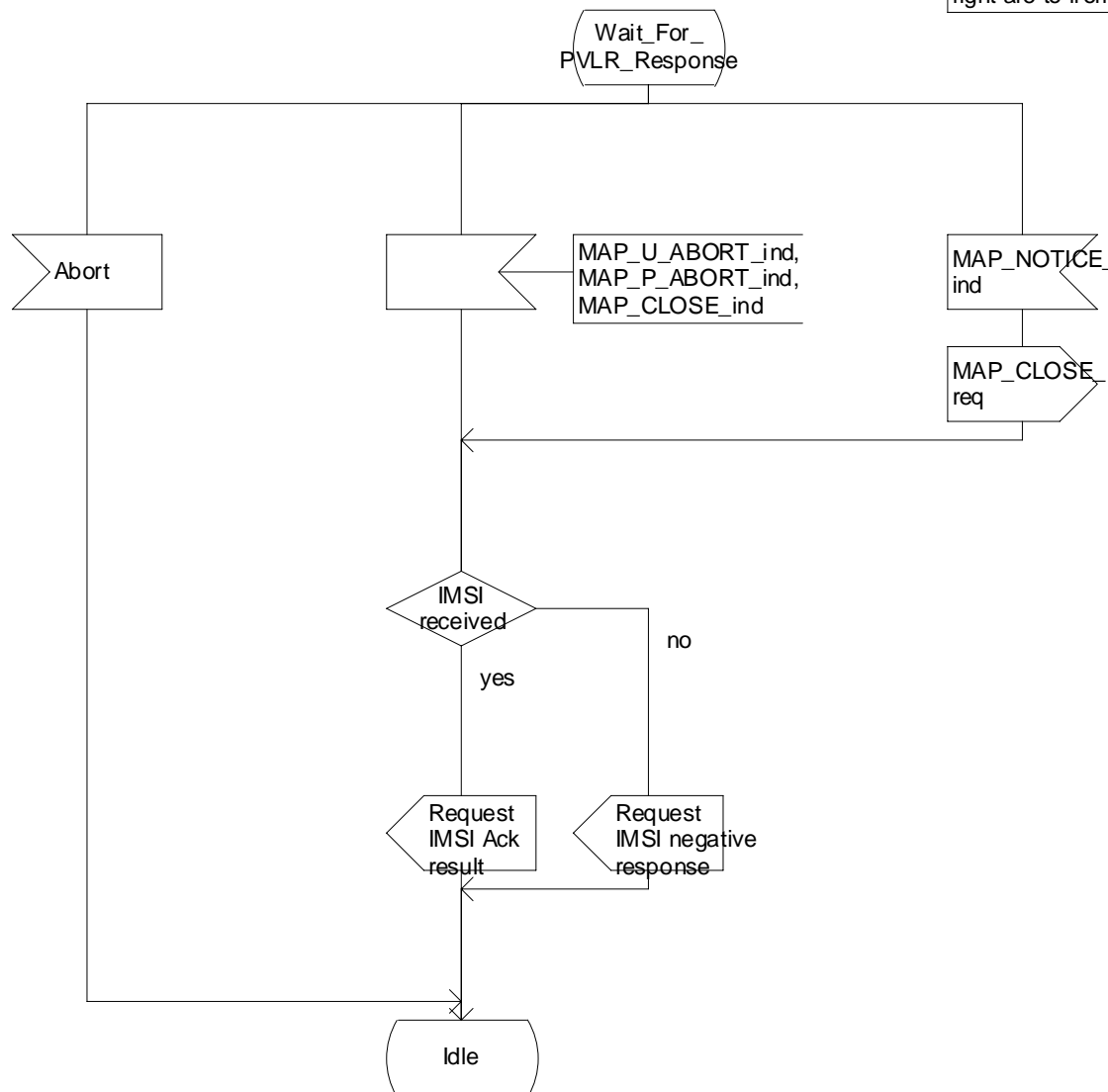


Figure 19.1.1/XX (sheet 2 of 2): Process Send_Identification_VLR

Process Send_Identification_PVLR

1(

Figure 19.1/11: Process in the Previous VLR to handle an identification request

Signals to/from the left are to from the new VLR. Signals to/from the right are to/from the PVLR Location Management application

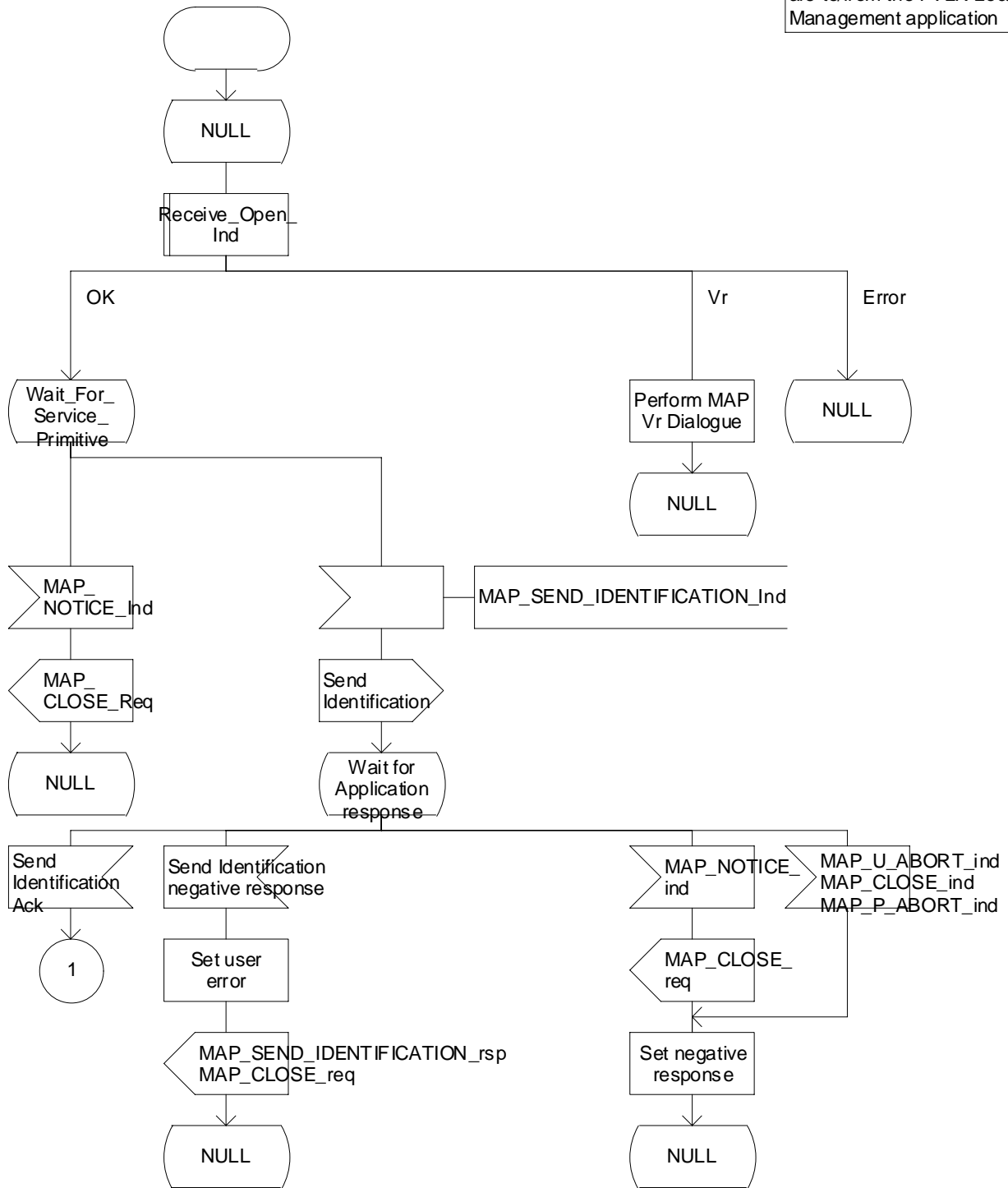


Figure 19.1/XX (sheet 1 of 2): Process Send_Identification_PVLR

Process Send_Identification_PVLR

Figure 19.1/11: Process in the Previous VLR to handle an identification request

Signals to/from the left are to/from the new VLR

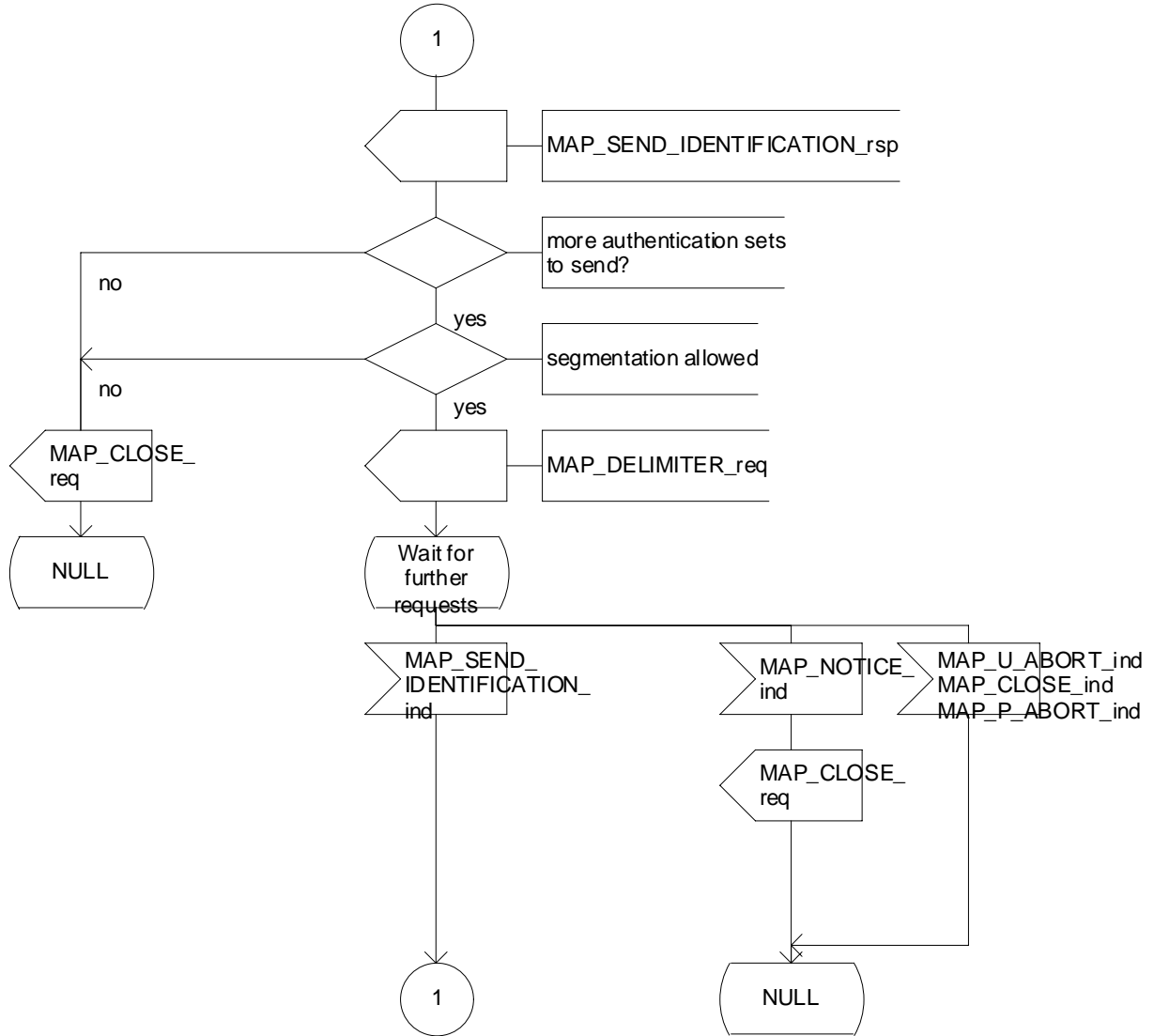


Figure 19.1.1/XX (sheet 2 of 2): Process Send_Identification_PVLR

19.1.1.6 Process Update Location VLR

This process is started by some other MAP user process in the 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 19.1.1.3), 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; the register will be updated and the SCP will be informed about the Mobility Management event. The process then terminates.

In the above case, the notification sent to the gsmSCF shall be '*Location Update to new VLR Service Area*'.

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 the 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.

NOTE: The procedure Notify_gsmSCF is specified in 3G TS 23.078.

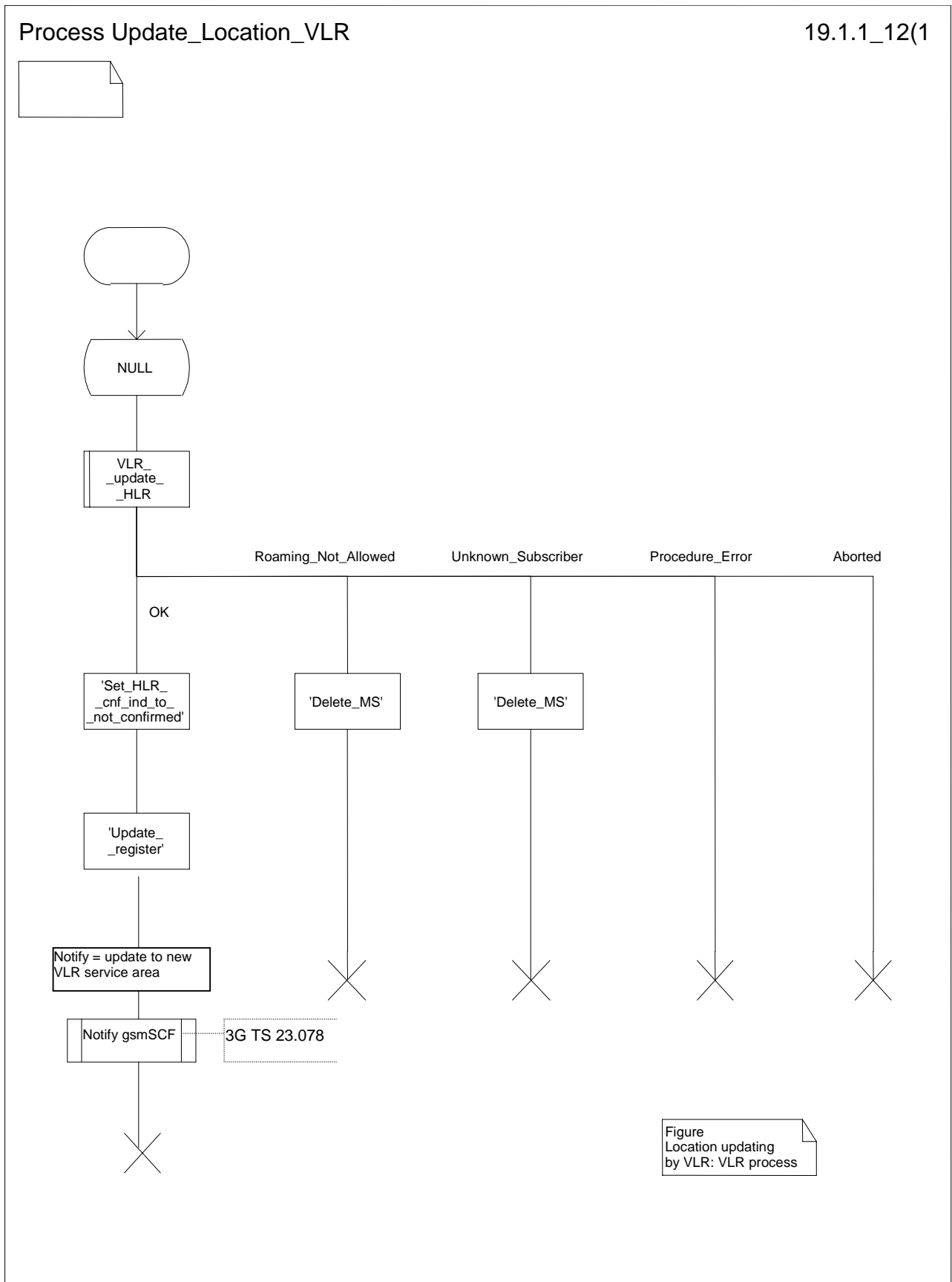


Figure 19.1.1/12: Process UL_VLR

19.1.1.8 Detailed procedure in the SGSN

Figure 19.1.1/20 shows the MAP process for updating of the SGSN. The following general macros are used:

Receive_Open_Cnf	subclause 25.1;
Insert_Subscriber_Data_SGSN	subclause 25.7;
Activate_Tracing_SGSN	subclause 25.9;

Sheet 2: The procedure Check_User_Error_In_Serving_Network_Entity is specific to Super-Charger; it is specified in TS 23.116 [110].

The location updating process

The MAP process receives an « Update HLR request » from the relevant process in the SGSN (see GSM 03.60) to perform HLR updating. If the SGSN 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 « Update HLR negative response » with error Roaming Not Allowed (cause PLMN Roaming Not Allowed) is returned to the requesting process.

If the subscribers HLR can be reached, the SGSN opens a dialogue towards the HLR by sending a MAP_OPEN request without any user specific parameters, together with a MAP_UPDATE_GPRS_LOCATION request containing the parameters

- IMSI, identifying the subscriber;
- SGSN Address and SGSN number;

In case the HLR rejects dialogue opening (see subclause 25.1) or indicates version Vr protocol to be used, the SGSN will terminate the process indicating « Update HLR negative response » to the requesting process.

If the HLR accepts the dialogue, the HLR will respond with:

- a MAP_INSERT_SUBSCRIBER_DATA indication, handled by the macro Insert_Subs_Data_SGSN defined in subclause 25.7;

NOTE: The HLR may repeat this service several times depending on the amount of data to be transferred to the SGSN and to replace subscription data in case they are not supported by the SGSN.

- a MAP_ACTIVATE_TRACE_MODE indication, handled by the macro Activate_Tracing_SGSN defined in subclause 25.9;
- the MAP_UPDATE_GPRS_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 « Update HLR response » message is returned to the requesting process for completion of the SGSN updating (see GSM 03.60).
 - if the confirmation contains an User error cause (Unknown Subscriber, Roaming Not Allowed or some other), the corresponding error is returned to the requesting process in the « Update HLR negative response ».
- a MAP_P_ABORT, MAP_U_ABORT, or MAP_CLOSE indication. In these cases, the corresponding error is returned to the requesting process in the « Update HLR negative response ».
- a MAP_NOTICE indication. Then, the dialogue towards the HLR is terminated, and the « HLR Update negative response » with the appropriate error is returned to the requesting process.

Process SGSN_Update_HLR

19.1.1_20.1(2)

Figure 19.1.1/20: HLR updating in SGSN

Signals from/to the left are from/to requesting process in SGSN
 Signals to/from the right are to/from the HLR

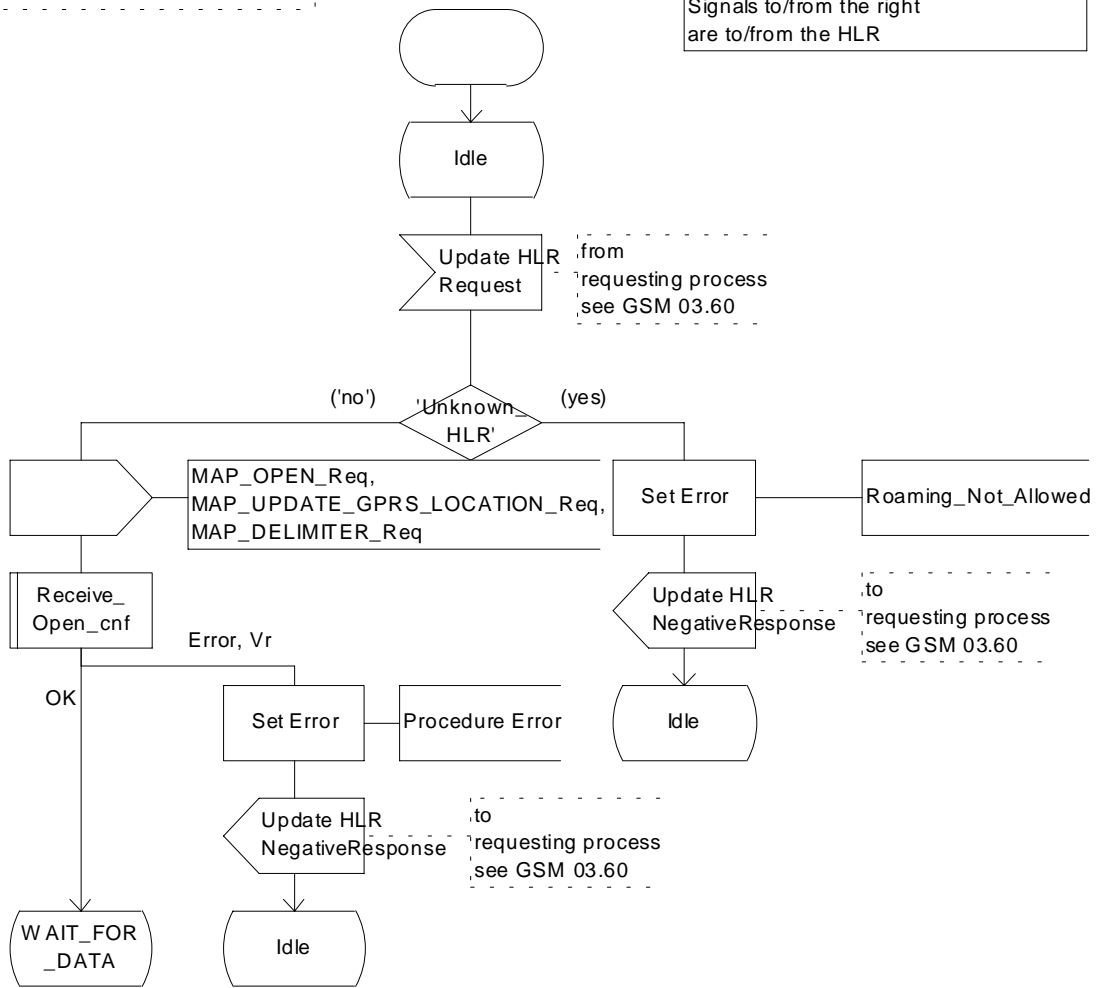


Figure 19.1.1/20 (sheet 1 of 2): Process SGSN_Update_HLR

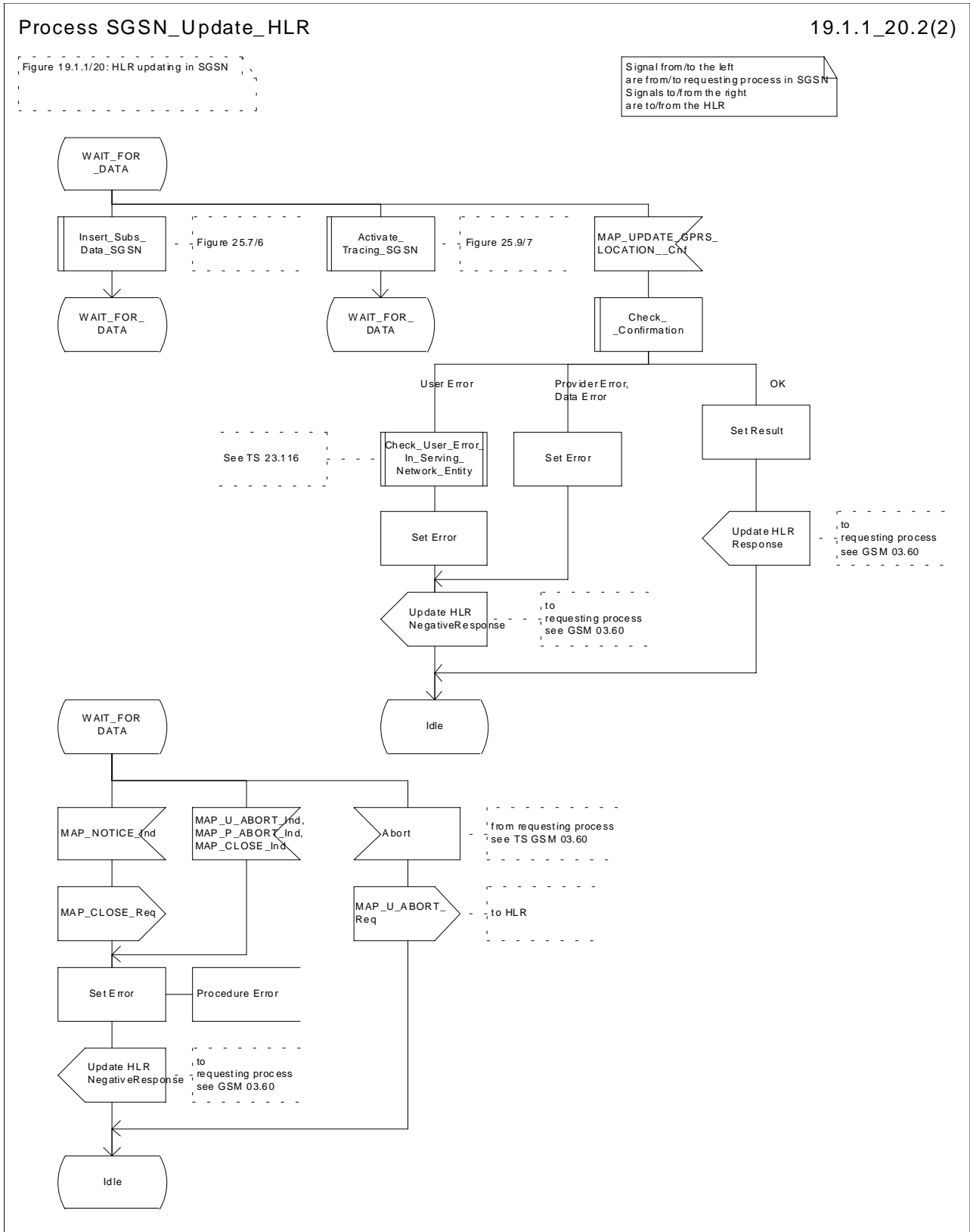


Figure 19.1.1/20 (sheet 2 of 2): Process SGSN_Update_HLR

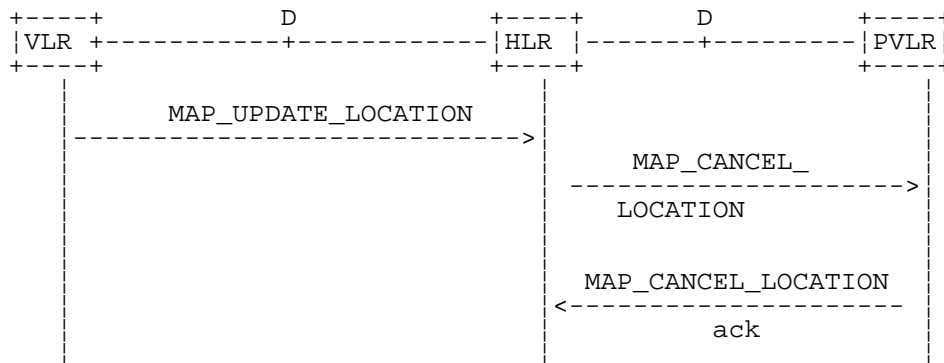
19.1.2 Location Cancellation

19.1.2.1 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. Also this process is used to delete a subscriber's record from a old SGSN after she has registered with a SGSN. 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 or in the SGSN 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 8.1.3 is invoked when an HLR receives a MAP_UPDATE_LOCATION indication from a VLR other than that stored in its table for this subscriber. Also the MAP_CANCEL_LOCATION service is invoked when the HLR receives a MAP_UPDATE_GPRS_LOCATION indication from a SGSN other than 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 or the SGSN whose identity is contained in the HLR table.



NOTE: The service shown in dotted lines indicates the trigger provided by other MAP signalling.

Figure 19.1.2/1: Interface and services for Location Cancellation

NOTE: The service shown in dotted lines indicates the trigger provided by other MAP signalling.

Figure 19.1.2/6: Interface and services for Location Cancellation in GPRS

19.1.2.2 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 or with the SGSN whose identity is contained in the HLR table (MAP_OPEN request without any user specific parameters), sending the MAP_CANCEL_LOCATION request primitive (see figures 16.1.2/2 and 16.1.2/4), containing the parameters:

- IMSI, to identify the subscriber to be deleted from that VLR or SGSN;
- LMSI, which is included if available in the HLR. LMSI is not applicable between HLR and SGSN;
- Cancellation Type if the Cancel Location is sent to SGSN. Cancellation Type is not applicable between HLR and VLR. If the VLR receives this parameter and do not understand it this parameter shall be ignored.

19.1.2.3 Detailed procedure in the VLR

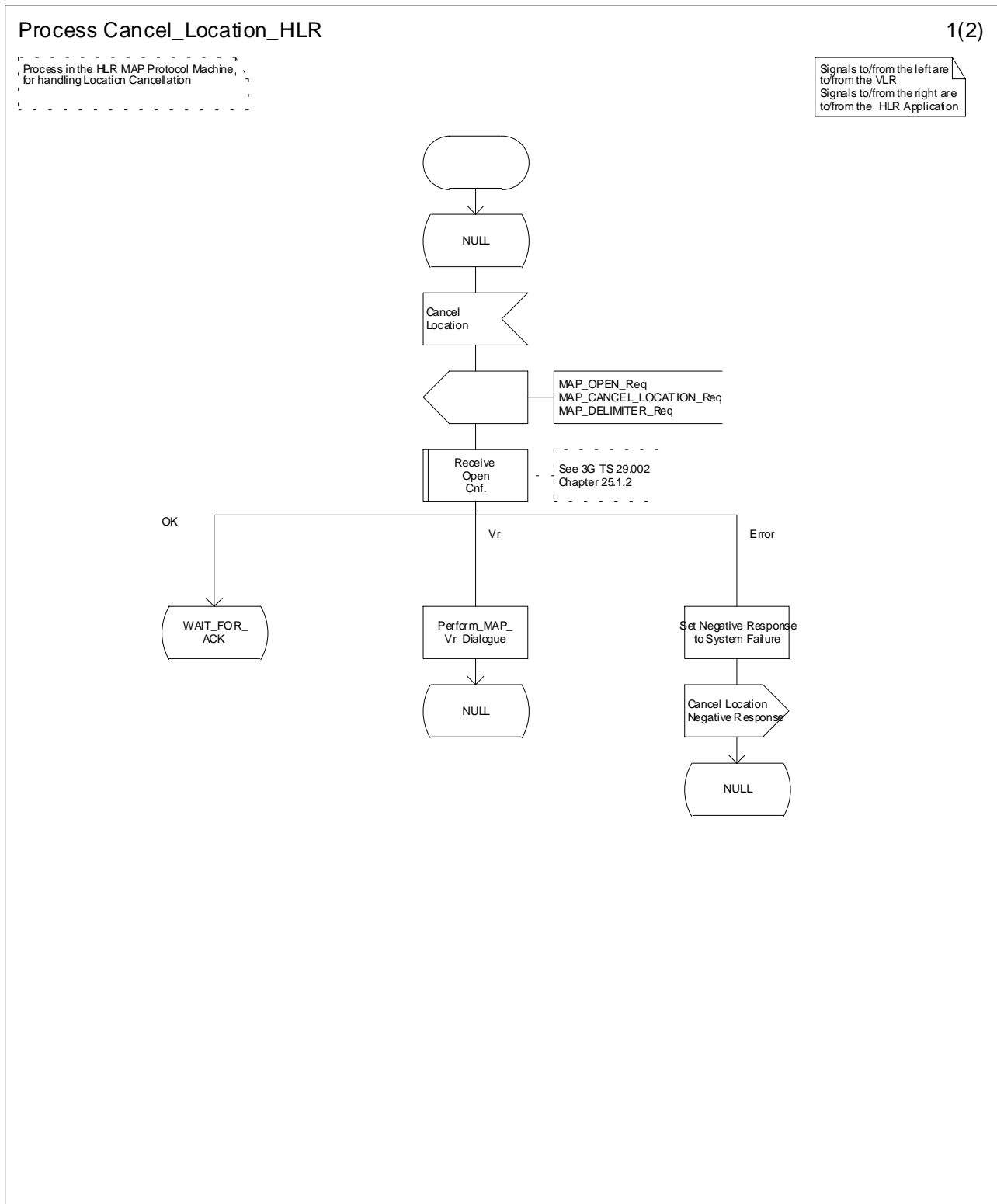


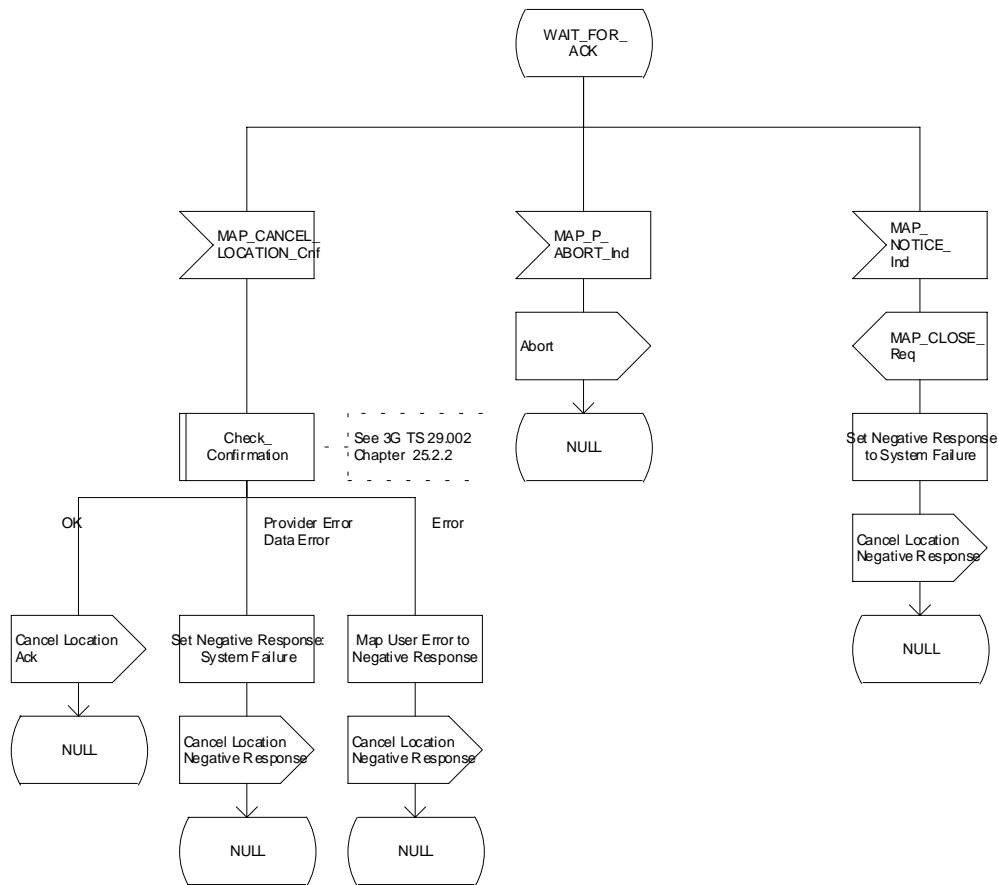
Figure 19.1.2/2 (Sheet 1 of 2): Process Cancel_Location_HLR

Process Cancel_Location_HLR

2(2)

Process in the HLR MAP Protocol Machine, for handling Location Cancellation

Signals to/from the left are to/from the VLR
Signals to/from the right are to/from the HLR Application



See 3G TS 29.002 Chapter 25.2.2

Process Cancel_Location_HLR

2(2)

Process in the HLR MAP Protocol Machine for handling Location Cancellation

Signals to/from the left are to/from the VLR
Signals to/from the right are to/from the HLR Application

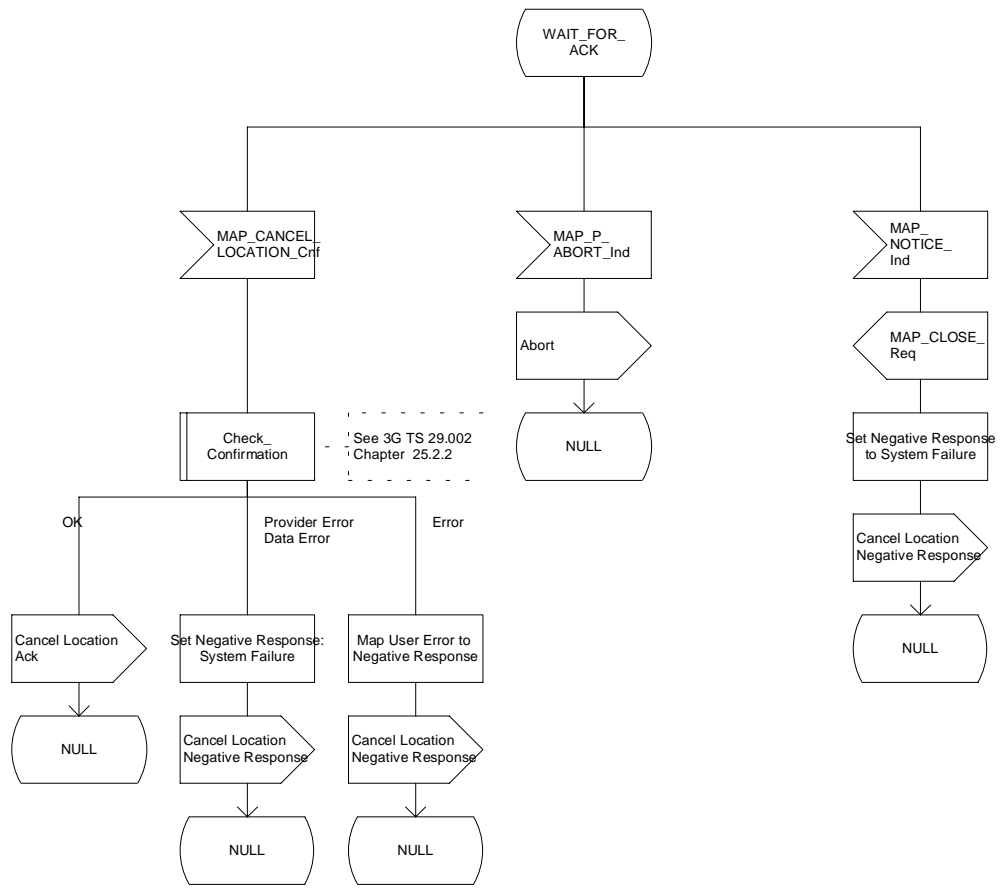


Figure 19.1.2/2 (Sheet 2 of 2): Process Cancel_Location_HLR

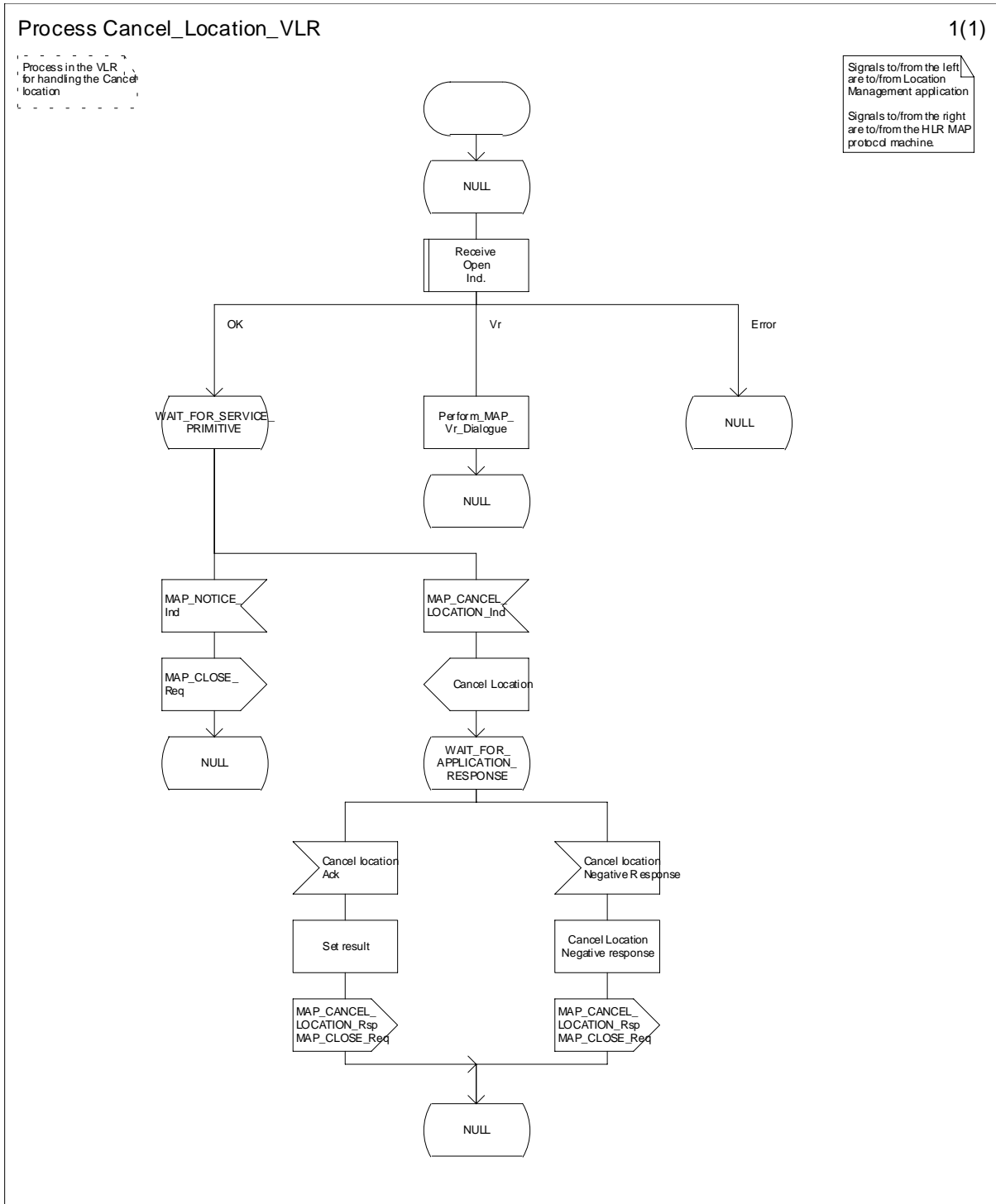


Figure 19.1.2/3: Process Cancel_Location_VLR

19.1.2.4 Detailed procedure in the SGSN

Opening of the dialogue is described in the macro Receive_Open_Ind in subclause 25.1, with outcomes:

- procedure termination; or
- dialogue acceptance, with processing as below.

If the SGSN process receives a MAP_NOTICE indication, it terminates the dialogue by sending a MAP_CLOSE request.

If the SGSN process receives a MAP_CANCEL_LOCATION indication from the HLR (see figure 19.1.2/4), the parameters are checked first (macro Check_Indication, see subclause 25.2). In case of parameter problems the appropriate error is sent in the MAP_CANCEL_LOCATION response.

Thereafter the SGSN checks whether the subscriber identity provided is known in the SGSN:

- if so, the data of the subscriber are deleted from SGSN 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 SGSN process releases any P-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.

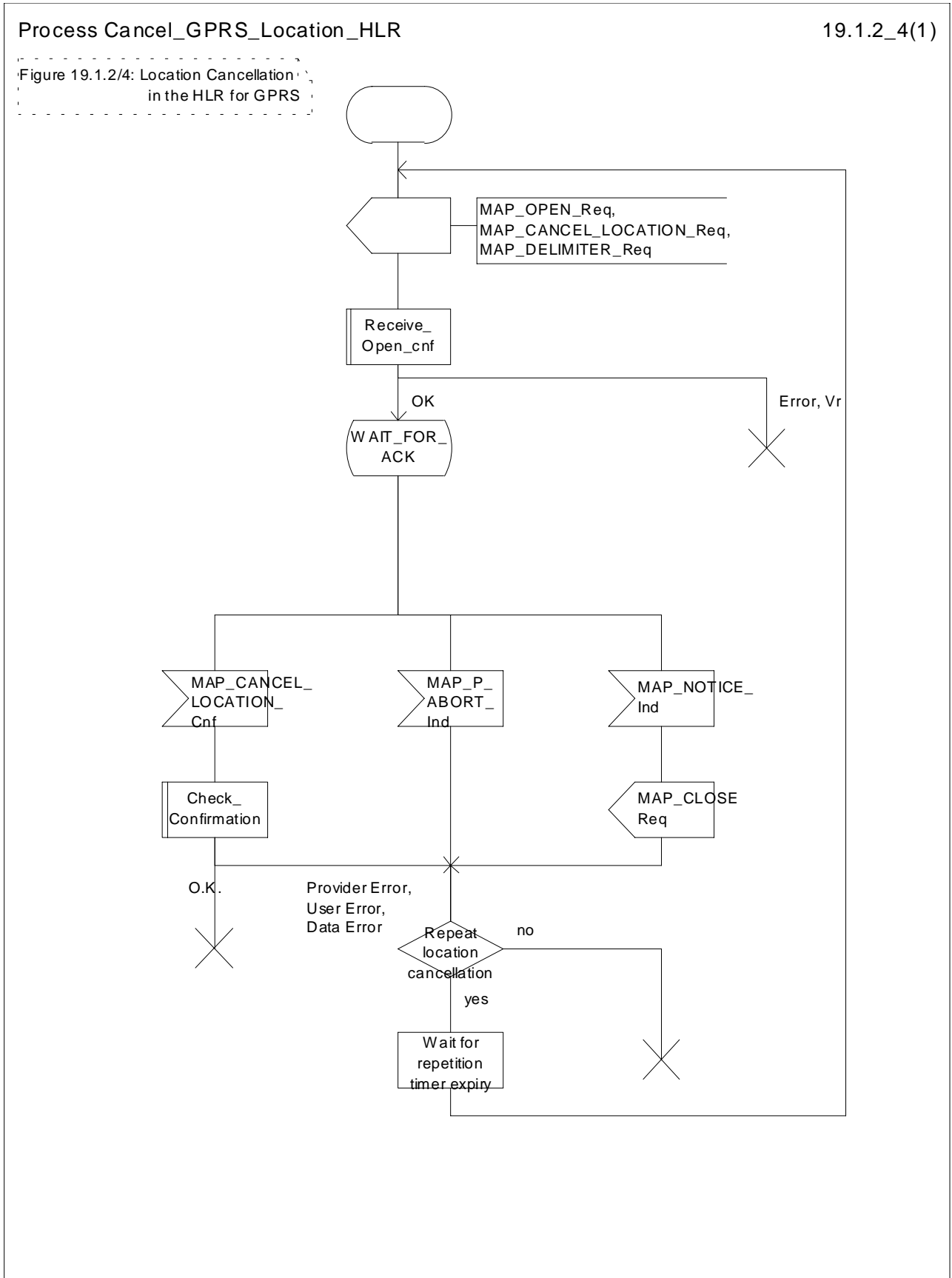


Figure 19.1.2/4: Process Cancel_GPRS_Location_HLR

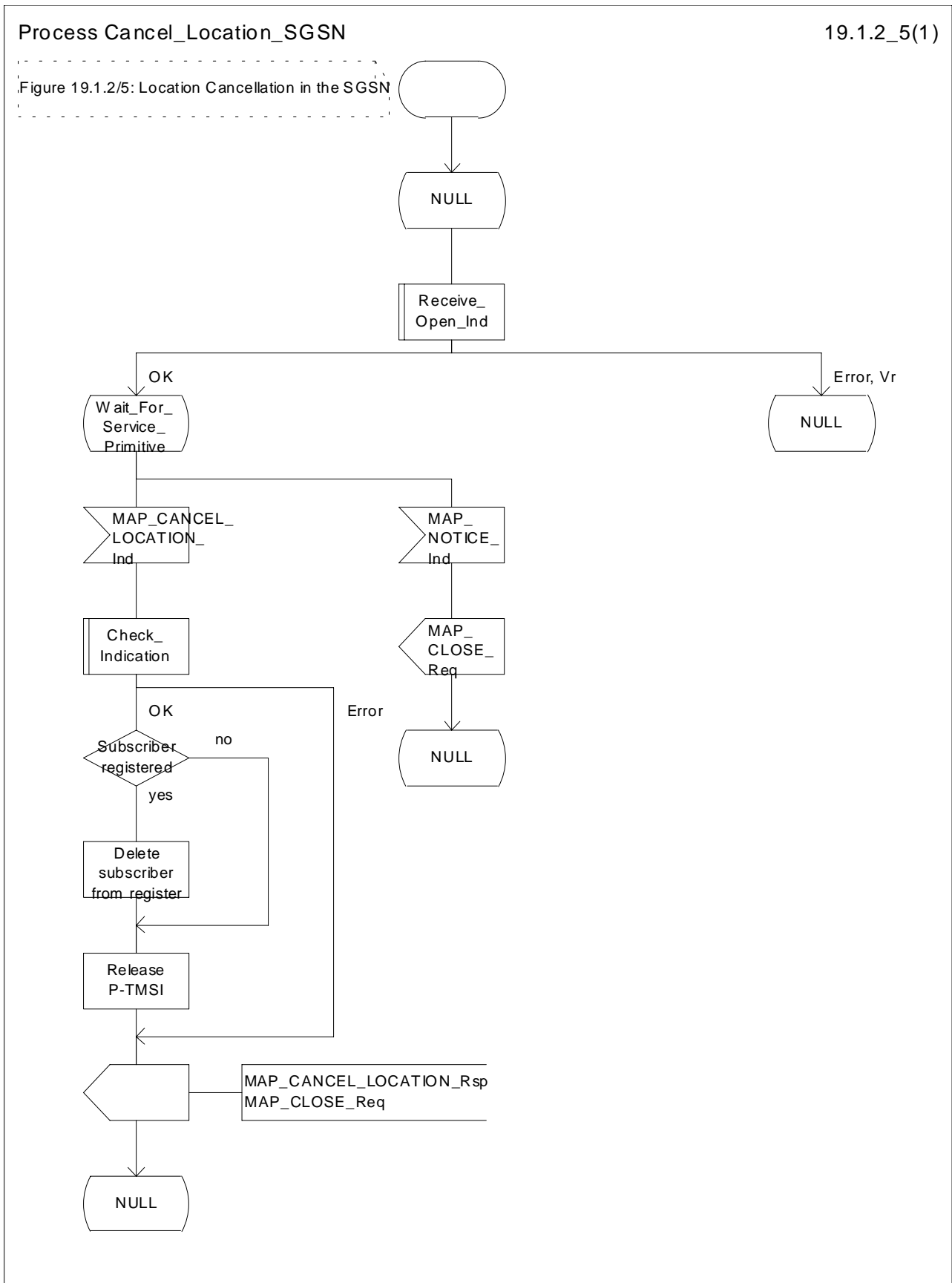


Figure 19.1.2/5: Process Cancel_Location_SGSN

19.1.3 Void

19.1.3.1 Void

19.1.3.2 Void

19.1.3.3 Void

19.1.4 Purge MS

19.1.4.1 General

When the VLR or the SGSN 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 8.1.6 to request the HLR to set the "MS purged for non-GPRS" or the "MS purged for GPRS" 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 flows are shown in figures 19.1.4/1 and 19.1.4/5.

It is optional for the network operator to delete MS records from the VLR or from the SGSN, but if the option is used the VLR or the SGSN shall notify the HLR when a record has been deleted.

The O&M process in the VLR or in the SGSN must ensure that during the MS purging procedure any other attempt to access the MS record is blocked, to maintain consistency of data.

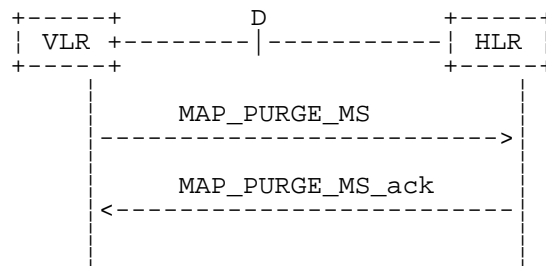


Figure 19.1.4/1: MAP-D Interface and services for MAP_PURGE_MS

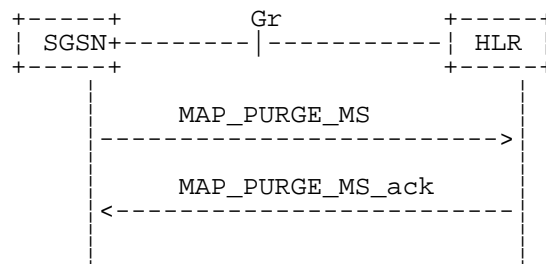


Figure 19.1.4/5: Gr Interface and services for MAP_PURGE_MS

19.1.4.2 Void

19.1.4.3 Void

19.1.4.4 Detailed procedure in the SGSN

Figure 19.1.4/4 shows the MAP process in the SGSN to notify the HLR that an MS record has been purged. The following general macro is used:

Receive_Open_Cnf subclause 25.1;

Sheet 1: The procedure `Purge_MS_In_Serving_Network_Entity` is specific to Super-Charger; it is specified in TS 23.116 [110]. If the SGSN and the originating HLR support the Super-Charger functionality, processing continues from the "Yes" exit of the test "Result=Pass?".

When the SGSN receives an indication from O&M that an MS record is to be purged, it invokes the `MAP_PURGE_MS` service.

The SGSN 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 SGSN number.

The SGSN then waits for the `MAP_OPEN` confirmation indicating one of:

- rejection of the dialogue (process terminates);
- reversion to Vr (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 P-TMSI if so requested by the HLR, and is reported to the O&M interface.

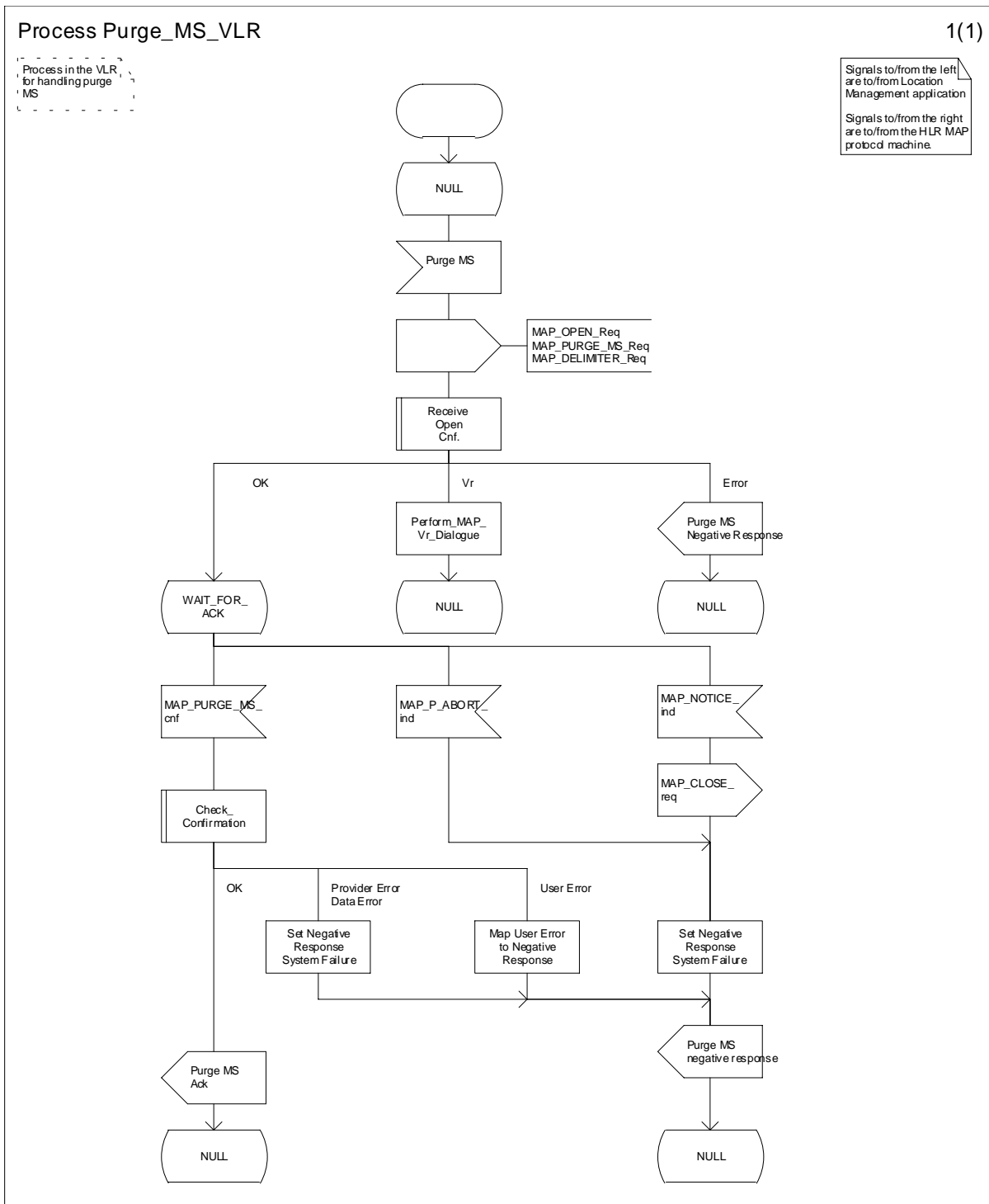


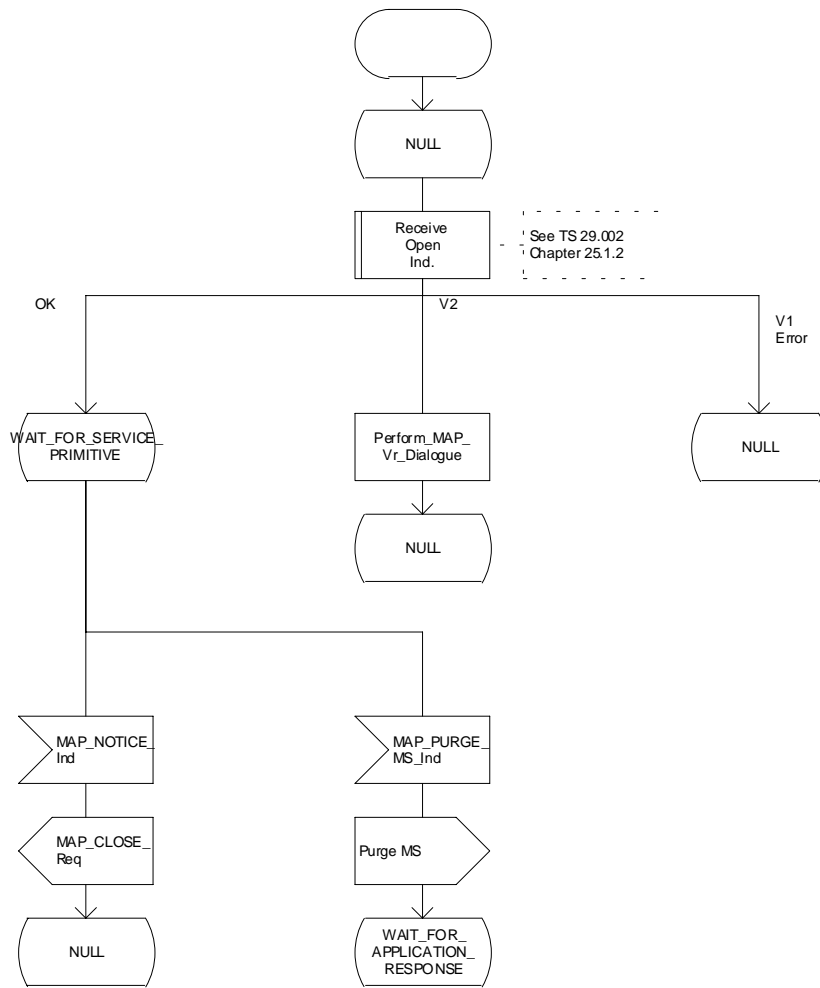
Figure 19.1.4/2: Process Purge_MS_VLR

Process Purge_MS_HLR

1(2)

Process in the HLR MAP Protocol Machine, for handling the Purging of an MS

Signals to/from the left are to/from the VLR
Signals to/from the right are to/from the HLR Application

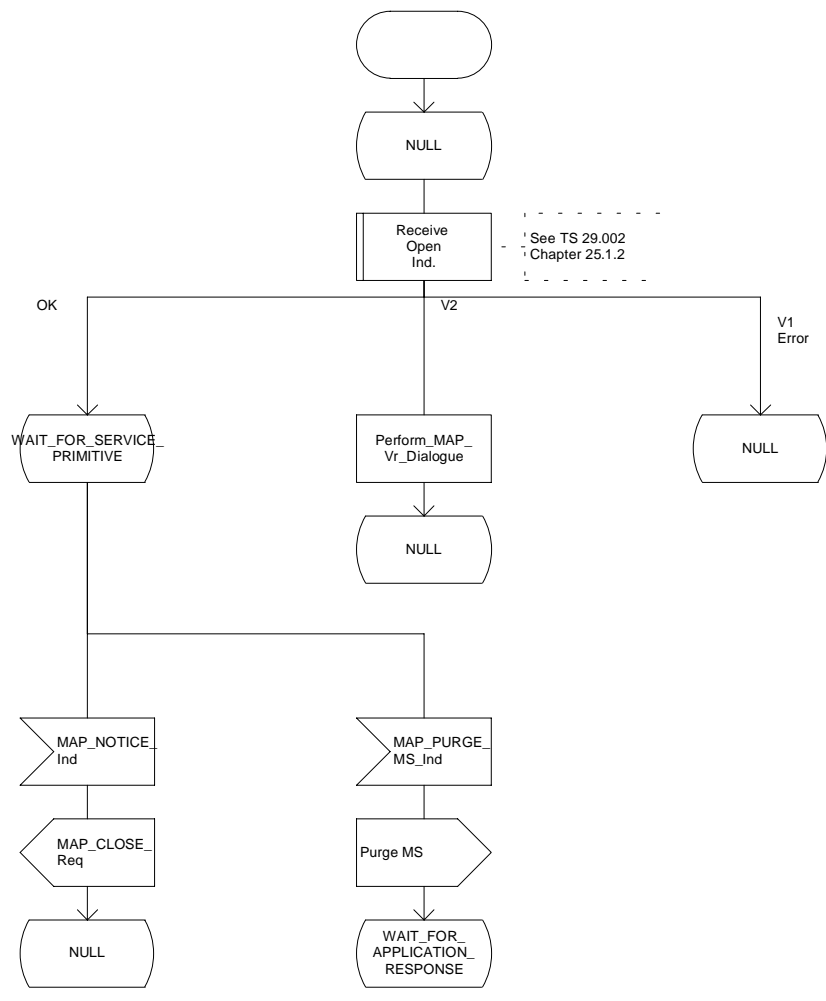


Process Purge_MS_HLR

1(2)

Process in the HLR MAP Protocol Machine, for handling the Purging of an MS

Signals to/from the left are to/from the VLR
Signals to/from the right are to/from the HLR Application



See TS 29.002 Chapter 25.1.2

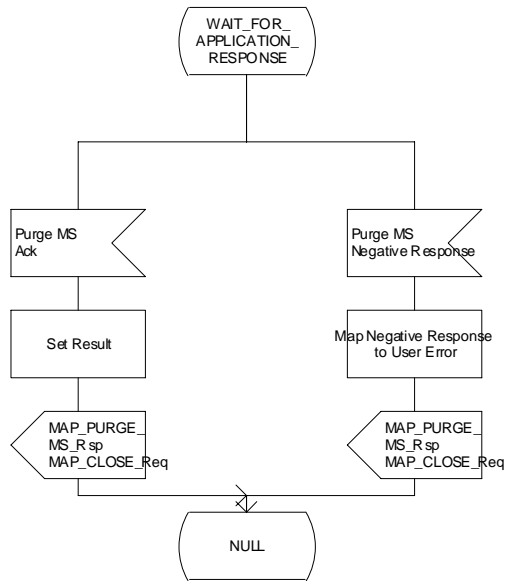
Figure 19.1.4/3 (Sheet 1 of 2): Process Purge_MS_HLR

Process Purge_MS_HLR

2(2)

Process in the HLR MAP Protocol Machine, for handling the Purging of an MS

Signals to/from the left are to/from the VLR
Signals to/from the right are to/from the HLR Application



Process Purge_MS_HLR

2(2)

Process in the HLR MAP Protocol Machine for handling the Purging of an MS

Signals to/from the left are to/from the VLR
Signals to/from the right are to/from the HLR Application

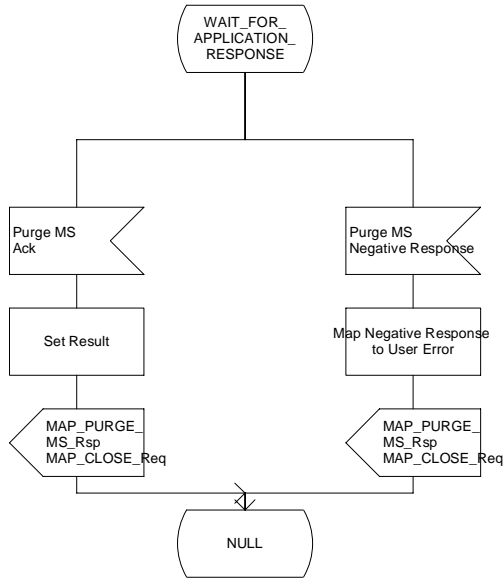


Figure 19.1.4/3 (Sheet 2 of 2): Process Purge_MS_HLR

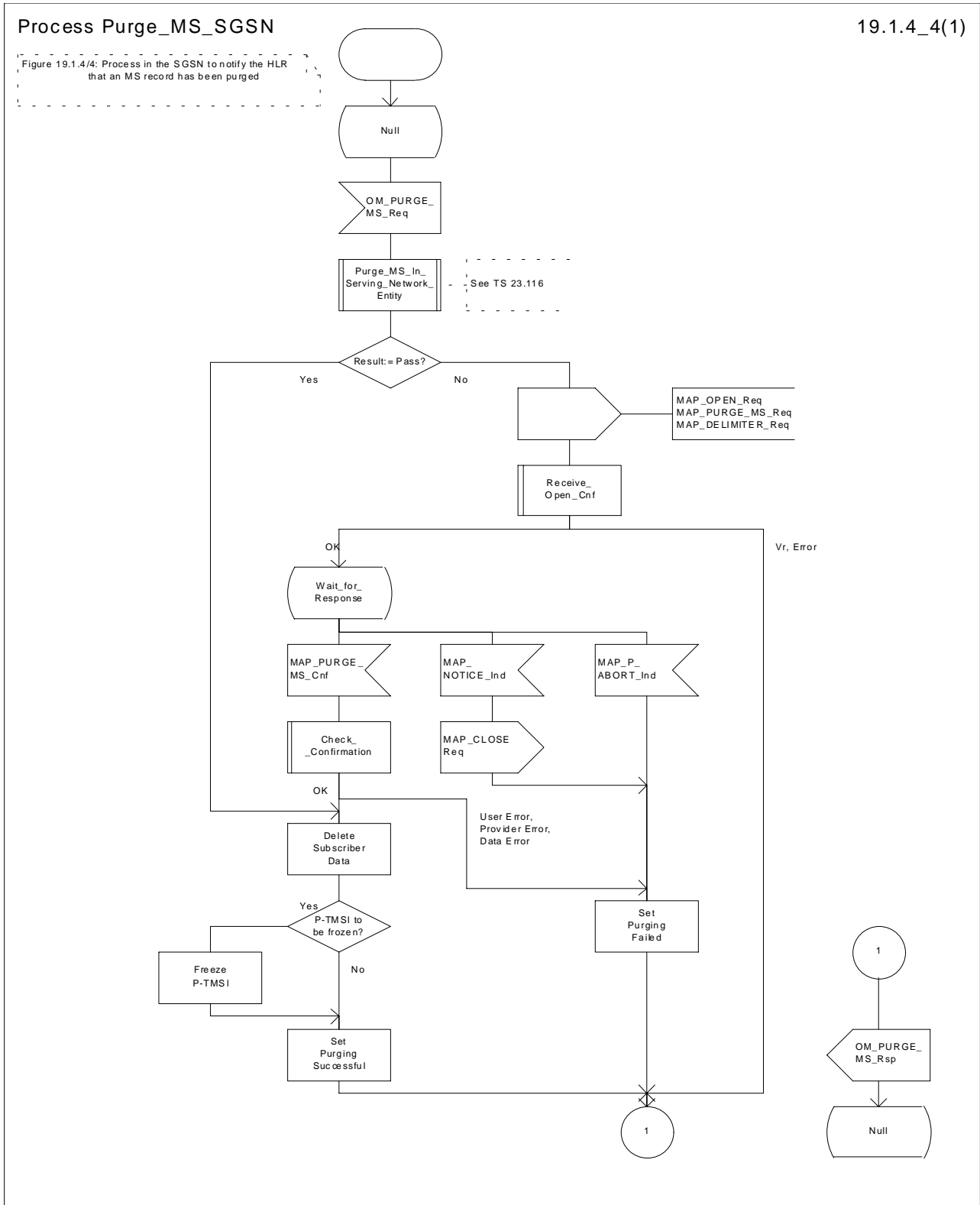


Figure 19.1.4/4: Process Purge_MS_SGSN

19.2 Handover procedure

19.2.1 General

The handover between different MSCs is called Inter-MSC handover. The interfaces involved for Inter-MSC handover are shown in figure 19.2/1. Following two Inter-MSC handover procedures apply:

1) Basic Inter-MSC handover:

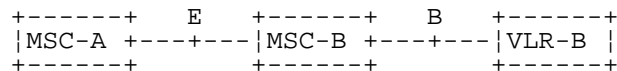
The call is handed over from the controlling MSC, called MSC-A to another MSC, called MSC-B (figure 19.2/1a).

Figure 19.2/2 shows a successful handover between MSC-A and MSC-B including a request for handover number allocation by MSC-B to VLR-B.

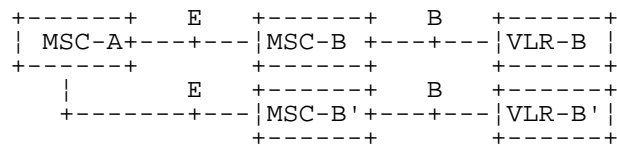
2) Subsequent Inter-MSC handover:

After the call has been handed over from MSC-A to MSC-B, a handover to either MSC-A (figure 19.2/1a) or to a third MSC (MSC-B') (figure 19.2/1b) is necessary in order to continue the connection.

Figure 19.2/3 shows a successful subsequent handover.



a) Basic handover procedure MSC-A to MSC-B and subsequent handover procedure MSC-B to MSC-A.



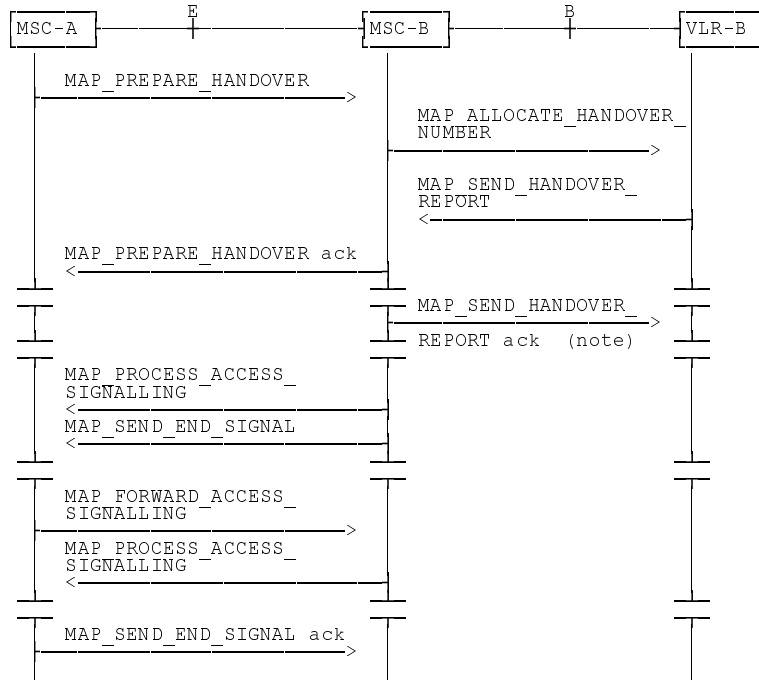
b) Subsequent handover procedure MSC-B to MSC-B'.

Figure 19.2/1: 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.

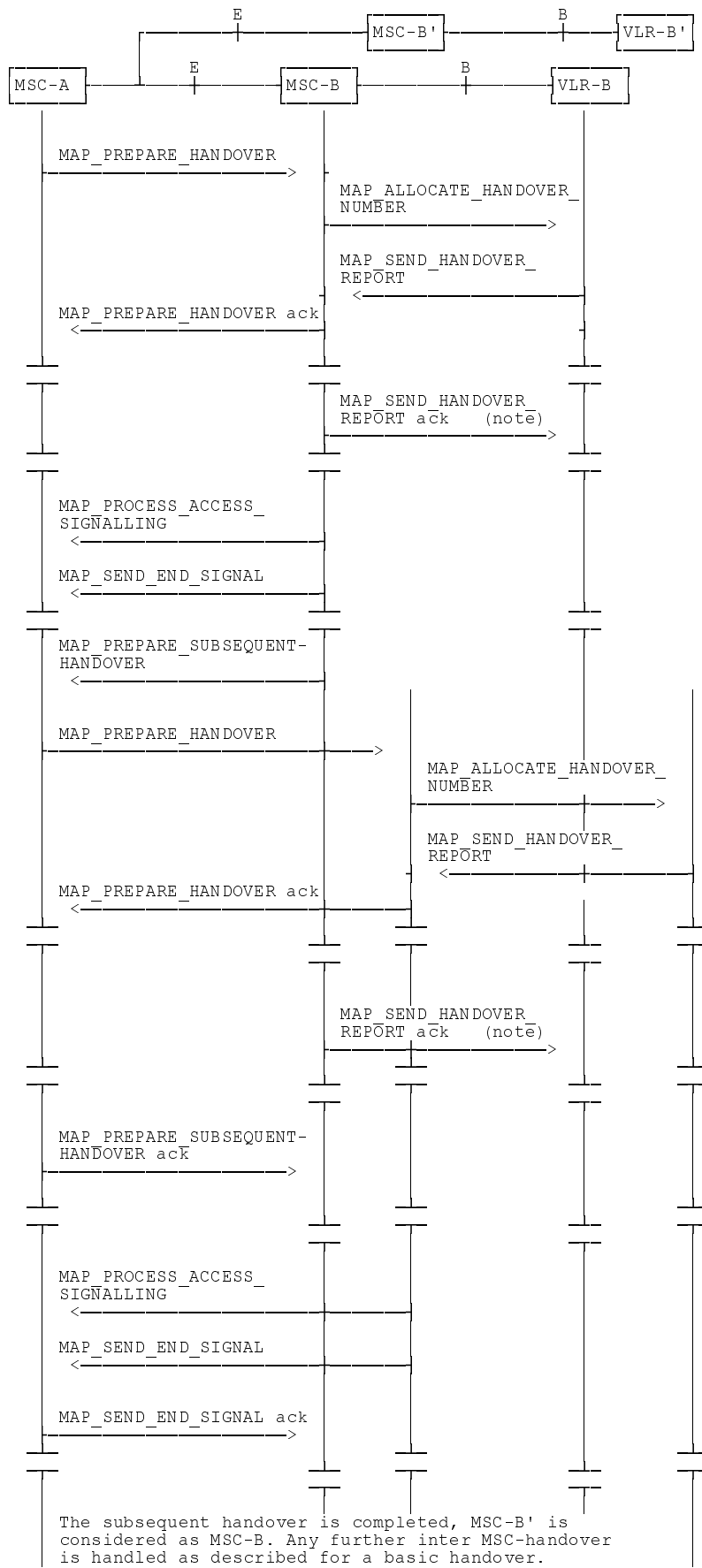
The transported BSSAP messages are controlled and handled by the Handover Control Application in the MSCs. This information will be transparent to the MAP protocol. If the MSC receives via the MAP protocol BSSAP messages, this information will be forwarded to the Handover Control Application (shown in 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.



NOTE: This can be sent at any time after the connection between MSC-A and MSC-B is established.

Figure 19.2/2: Example of a successful basic handover procedure to MSC-B



NOTE: This can be sent at any time after the connection between MSC-A and MSC-B is established.

Figure 19.2/3: Example of a handover towards a third MSC

19.2.2 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).

19.2.2.1 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_HANOVER 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_HANOVER 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_HANOVER 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_HANOVER 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).

19.2.2.2 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.

19.2.2.3 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.

19.2.2.4 Subsequent handover

When MSC-A receives a MAP_PREPARE_SUBSEQUENT_HANOVER 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.

19.2.2.5 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 8.4.

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.

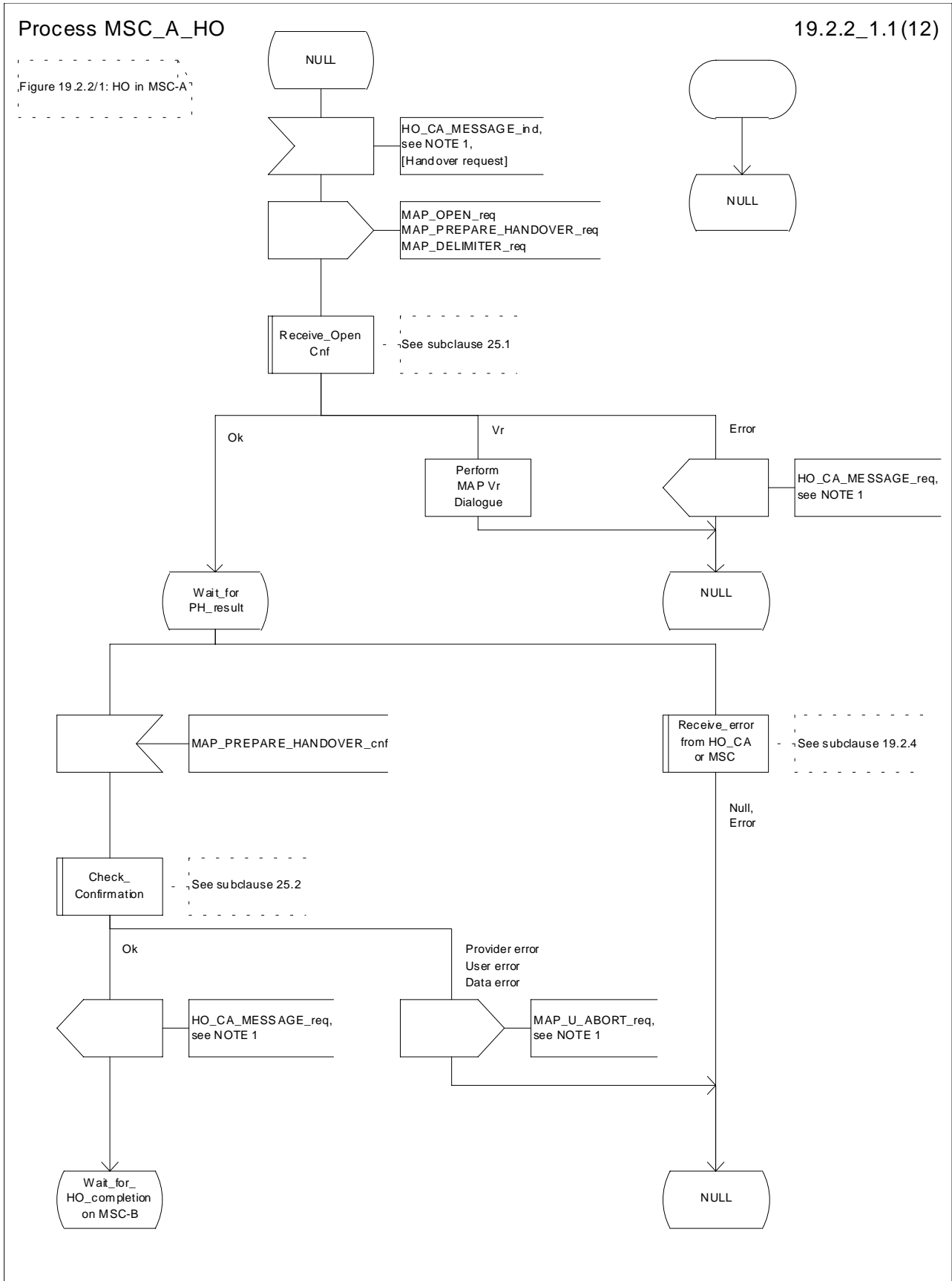


Figure 19.2.2/1 (sheet 1 of 12): Process MSC_A_HO

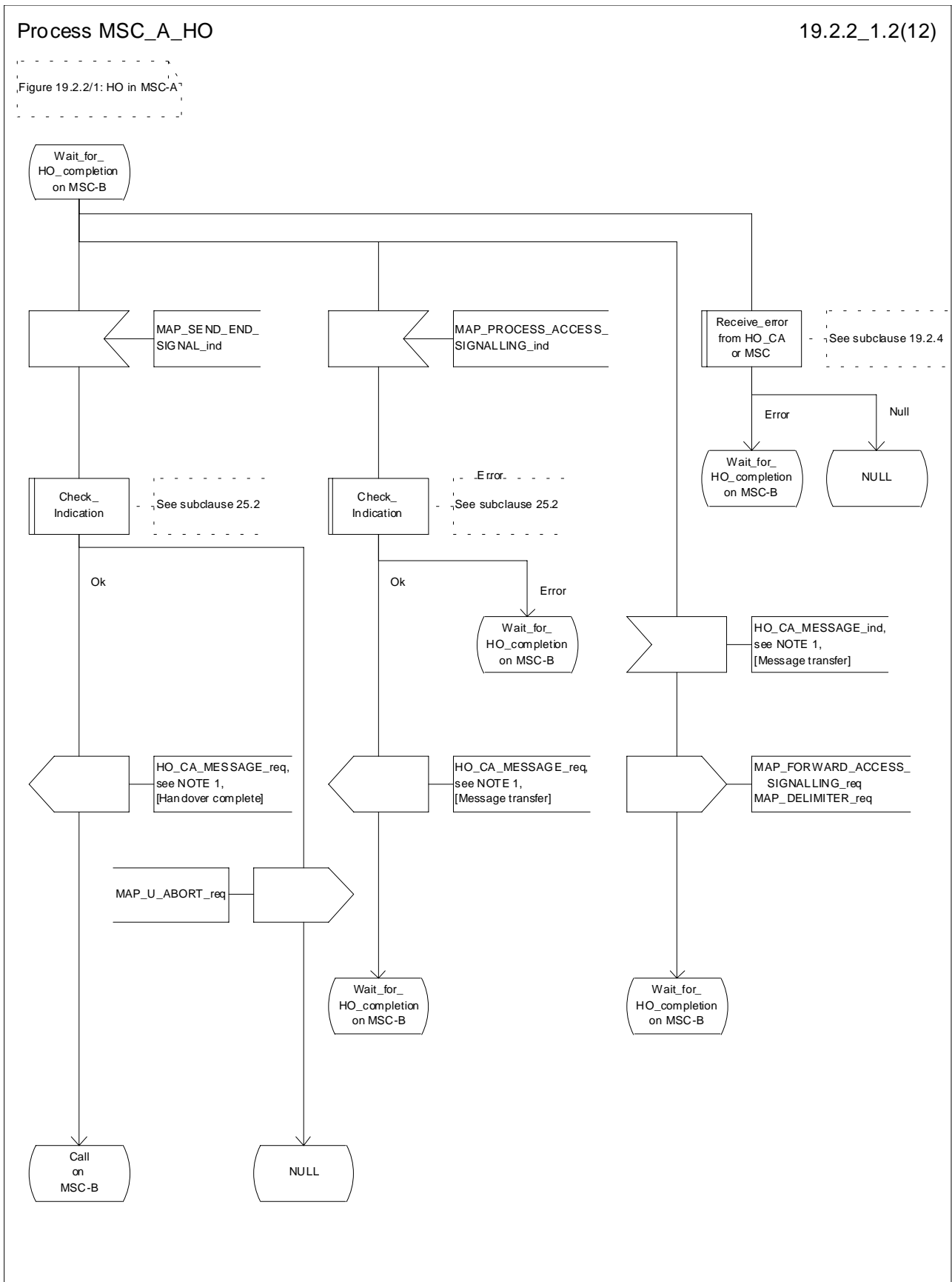


Figure 19.2.2/1 (sheet 2 of 12): Process MSC_A_HO

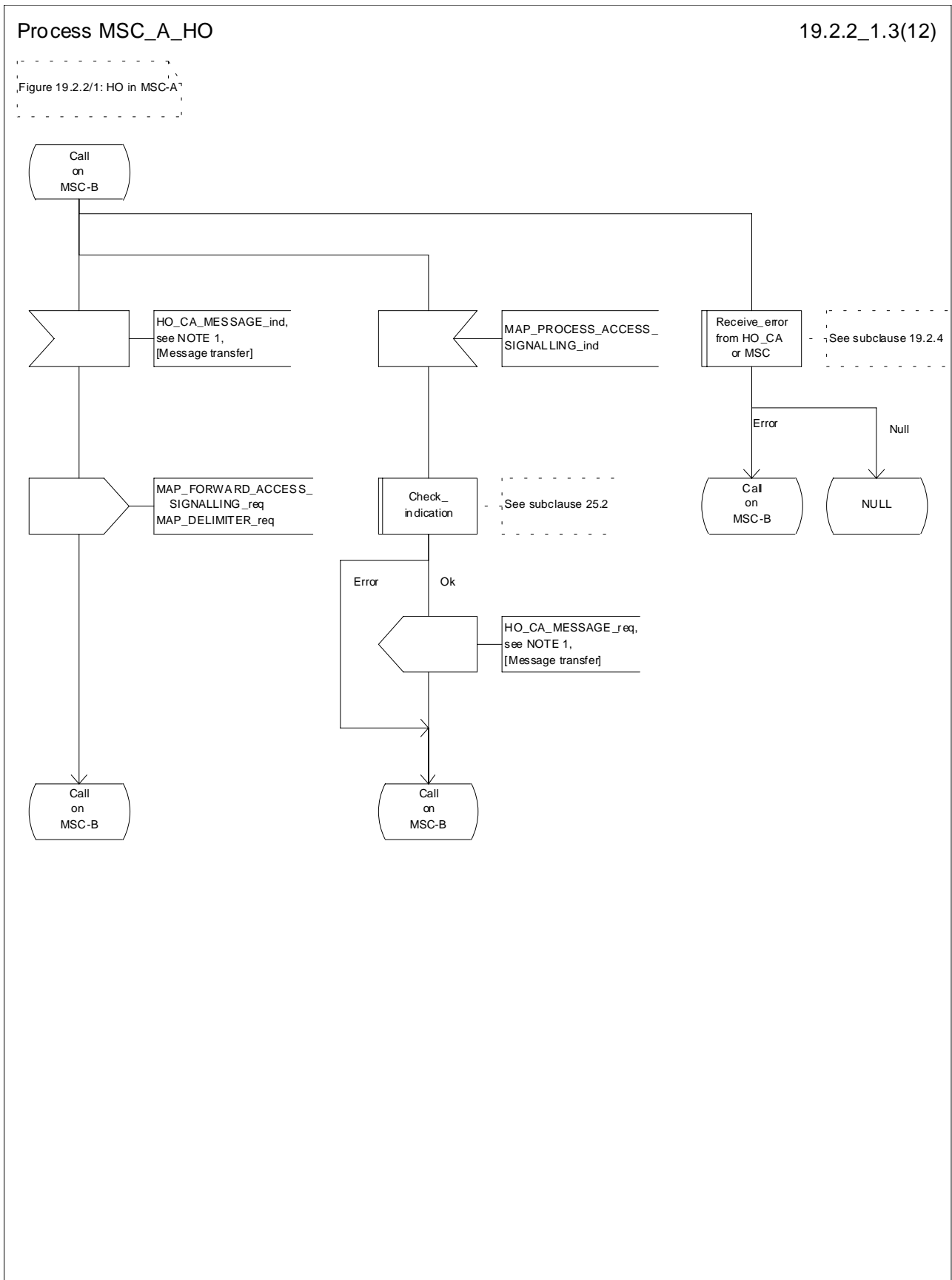


Figure 19.2.2/1 (sheet 3 of 12): Process MSC_A_HO

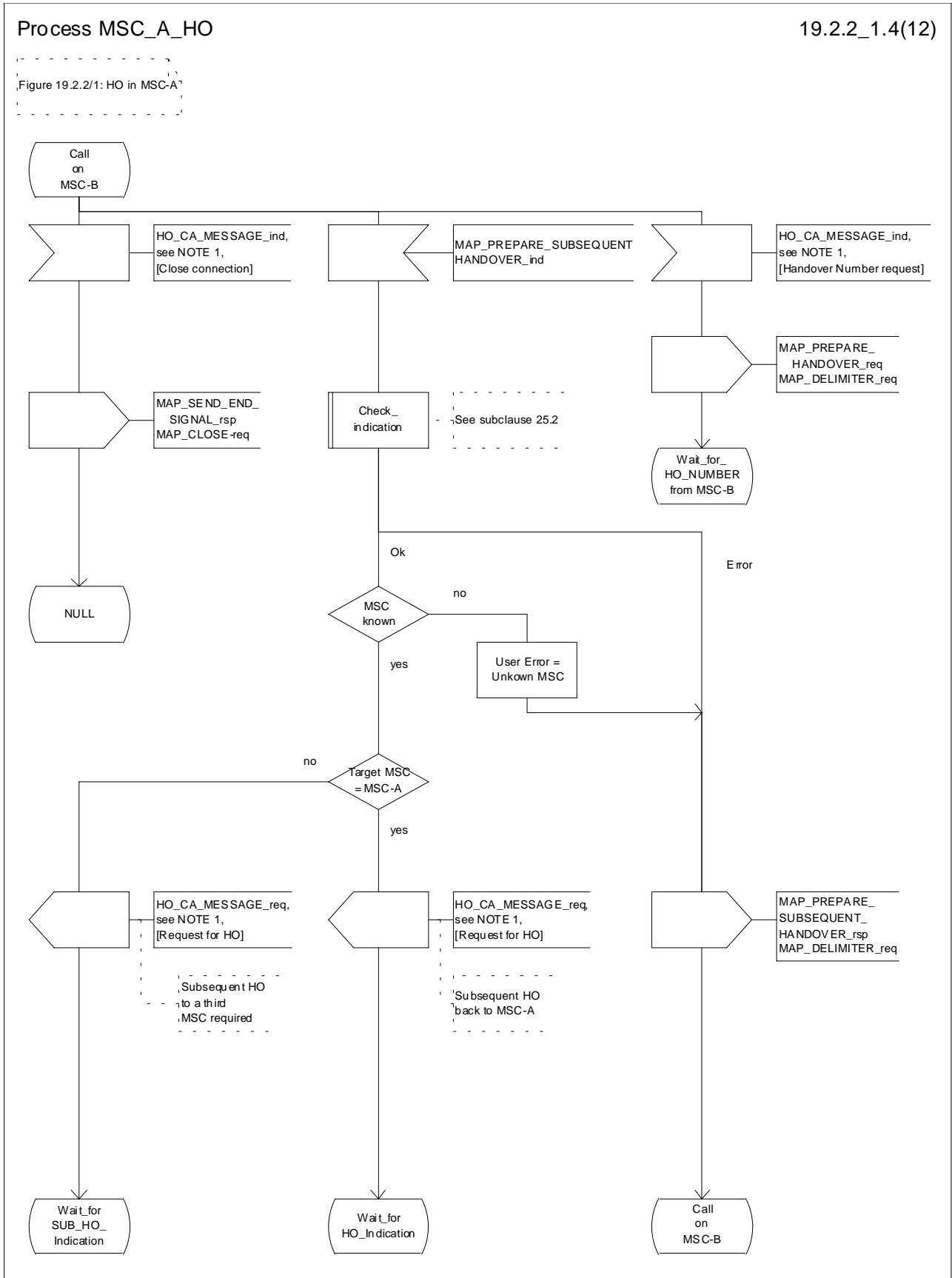


Figure 19.2.2/1 (sheet 4 of 12): Process MSC_A_HO

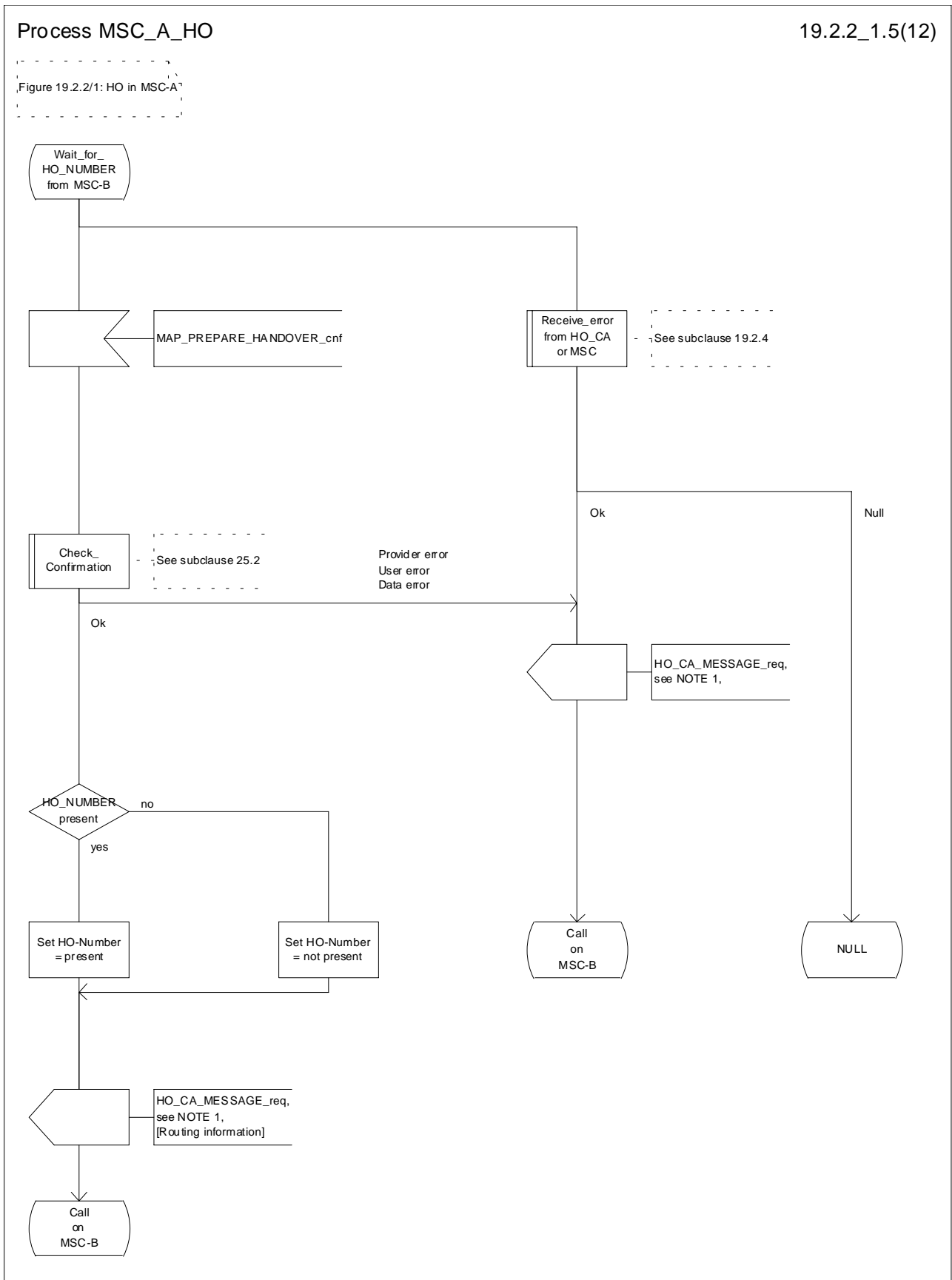


Figure 19.2.2/1 (sheet 5 of 12): Process MSC_A_HO

Process MSC_A_HO

19.2.2_1.6(12)

Figure 19.2.2/1: HO in MSC-A

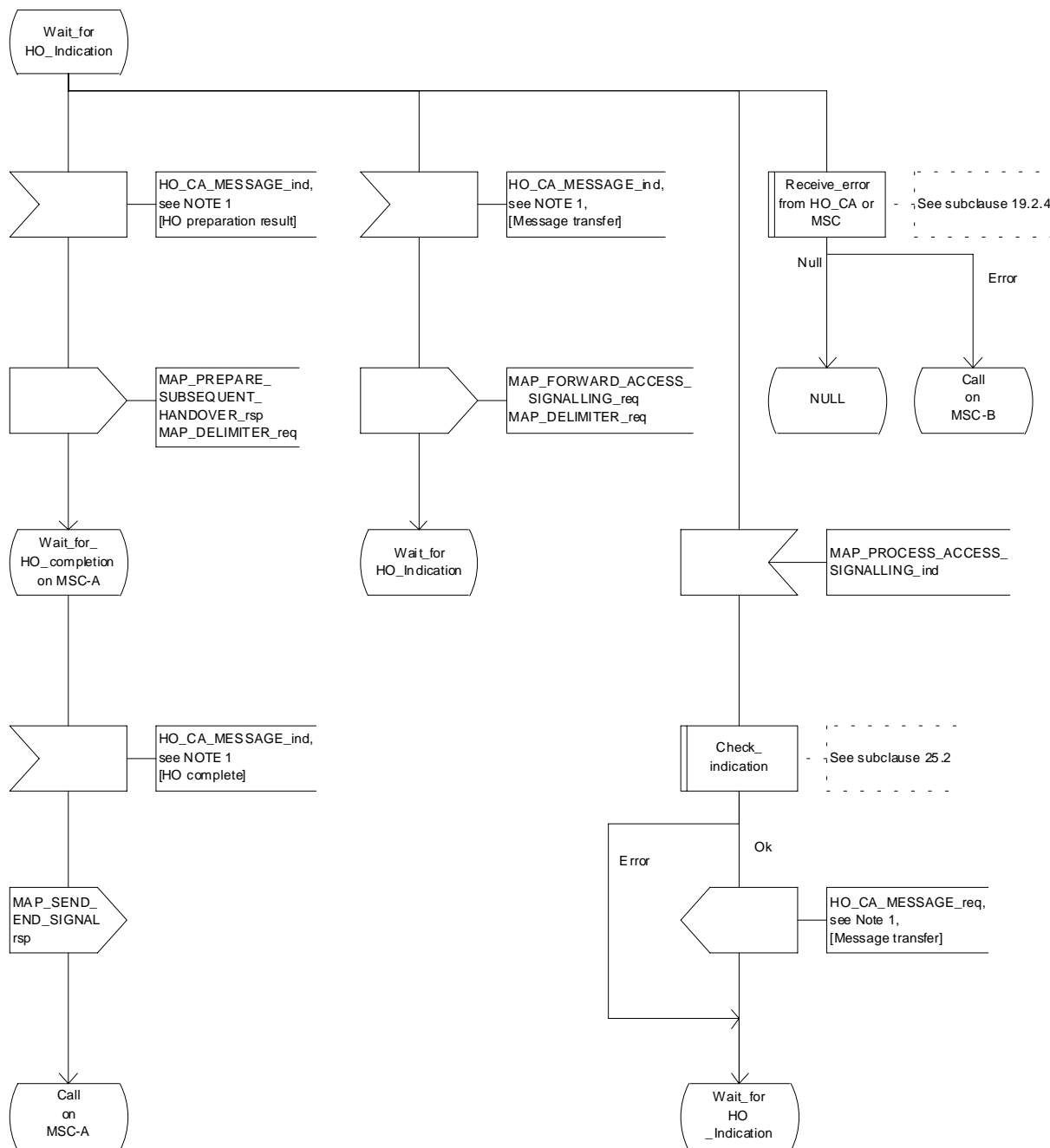


Figure 19.2.2/1 (sheet 6 of 12): Process MSC_A_HO

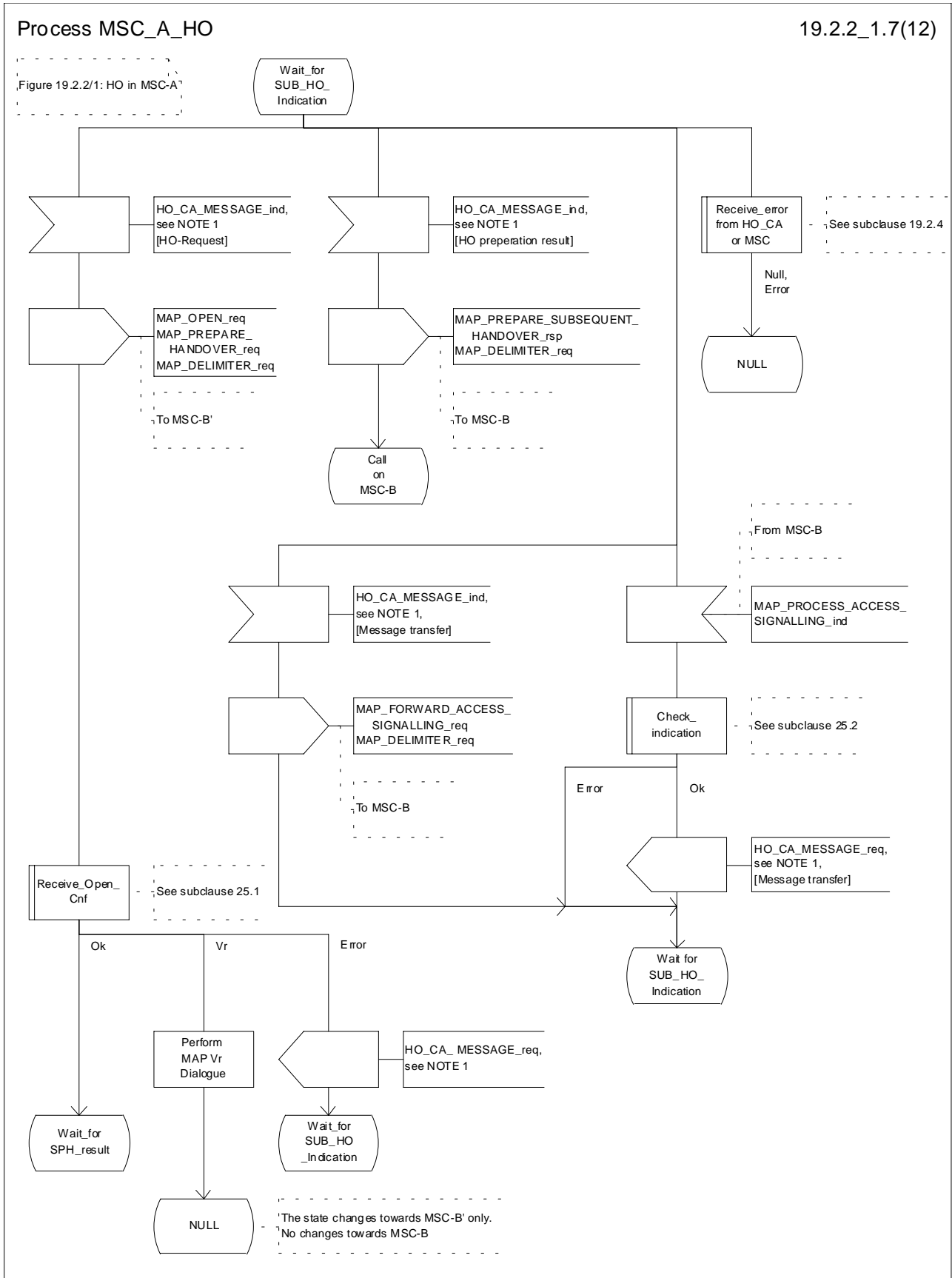


Figure 19.2.2/1 (sheet 7 of 12): Process MSC_A_HO

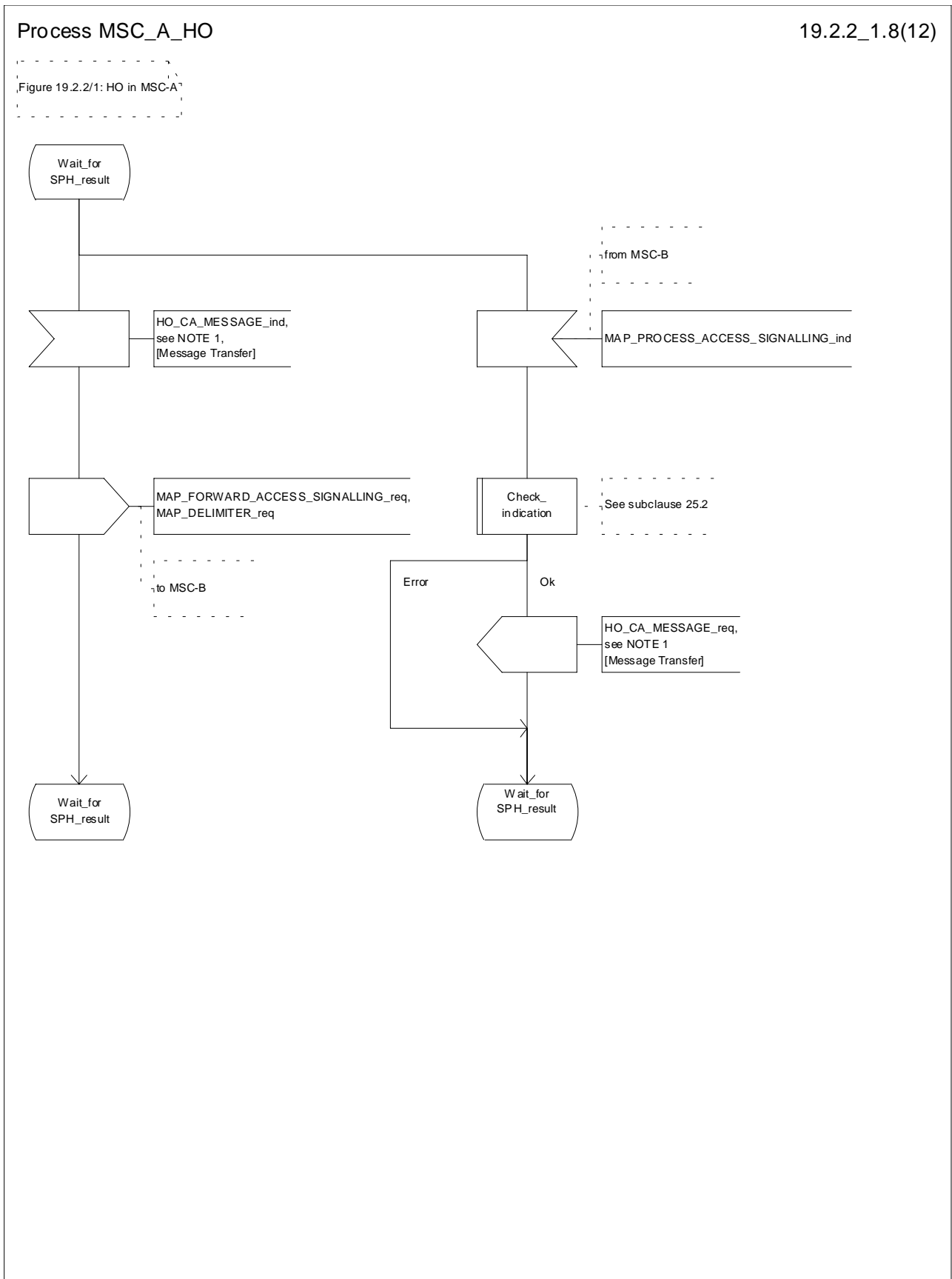


Figure 19.2.2/1 (sheet 8 of 12): Process MSC_A_HO

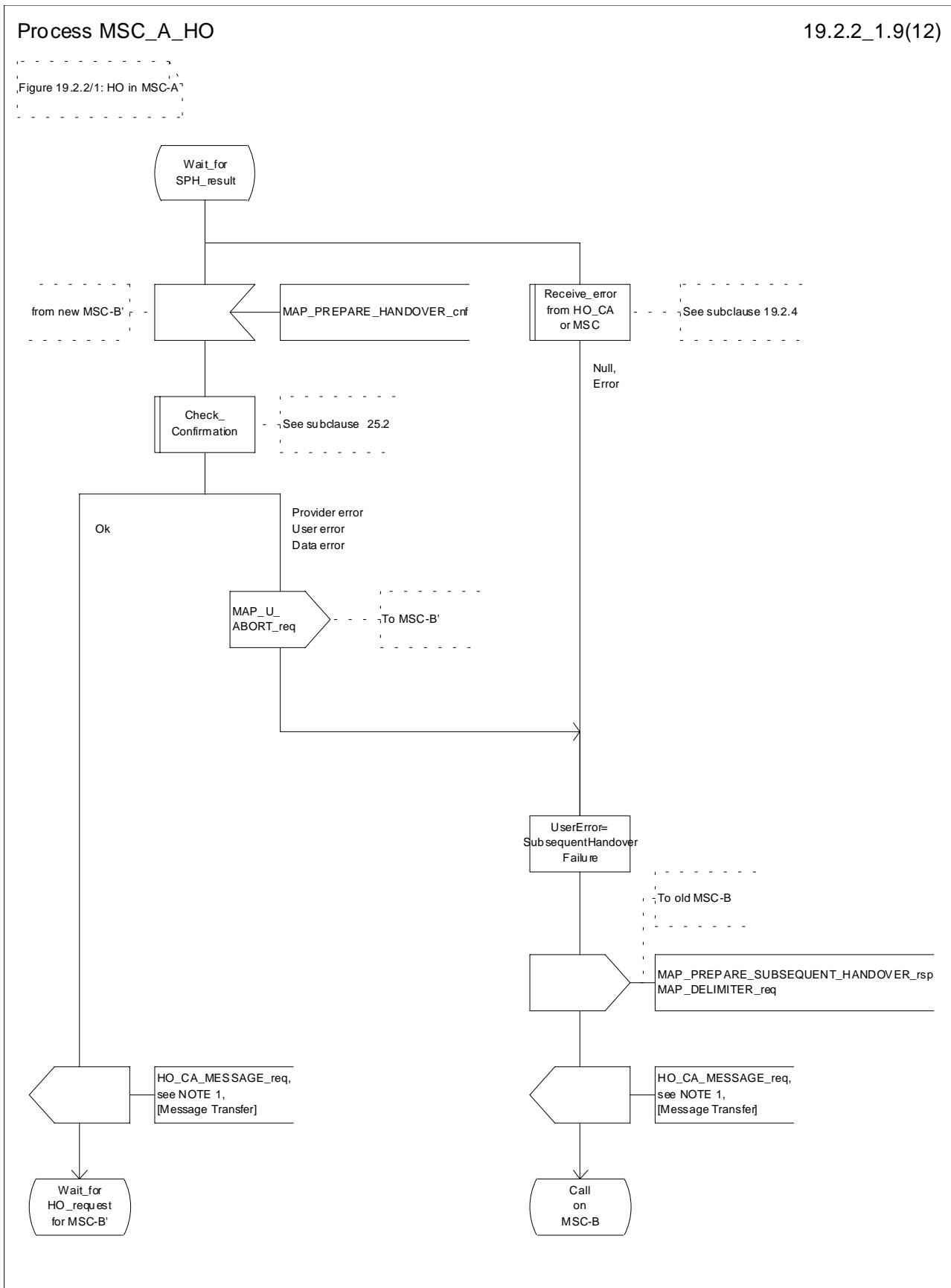


Figure 19.2.2/1 (sheet 9 of 12): Process MSC_A_HO

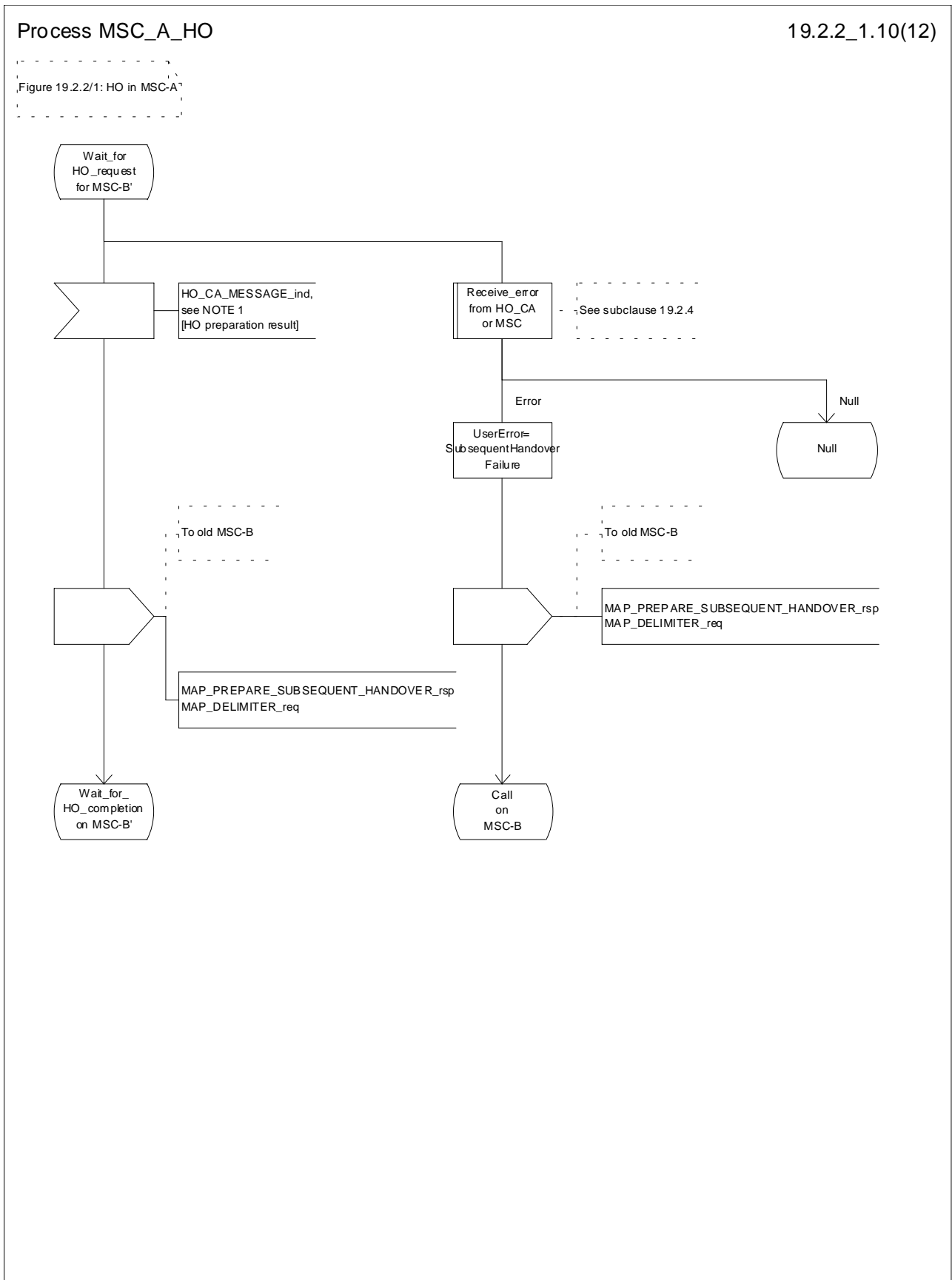


Figure 19.2.2/1 (sheet 10 of 12): Process MSC_A_HO

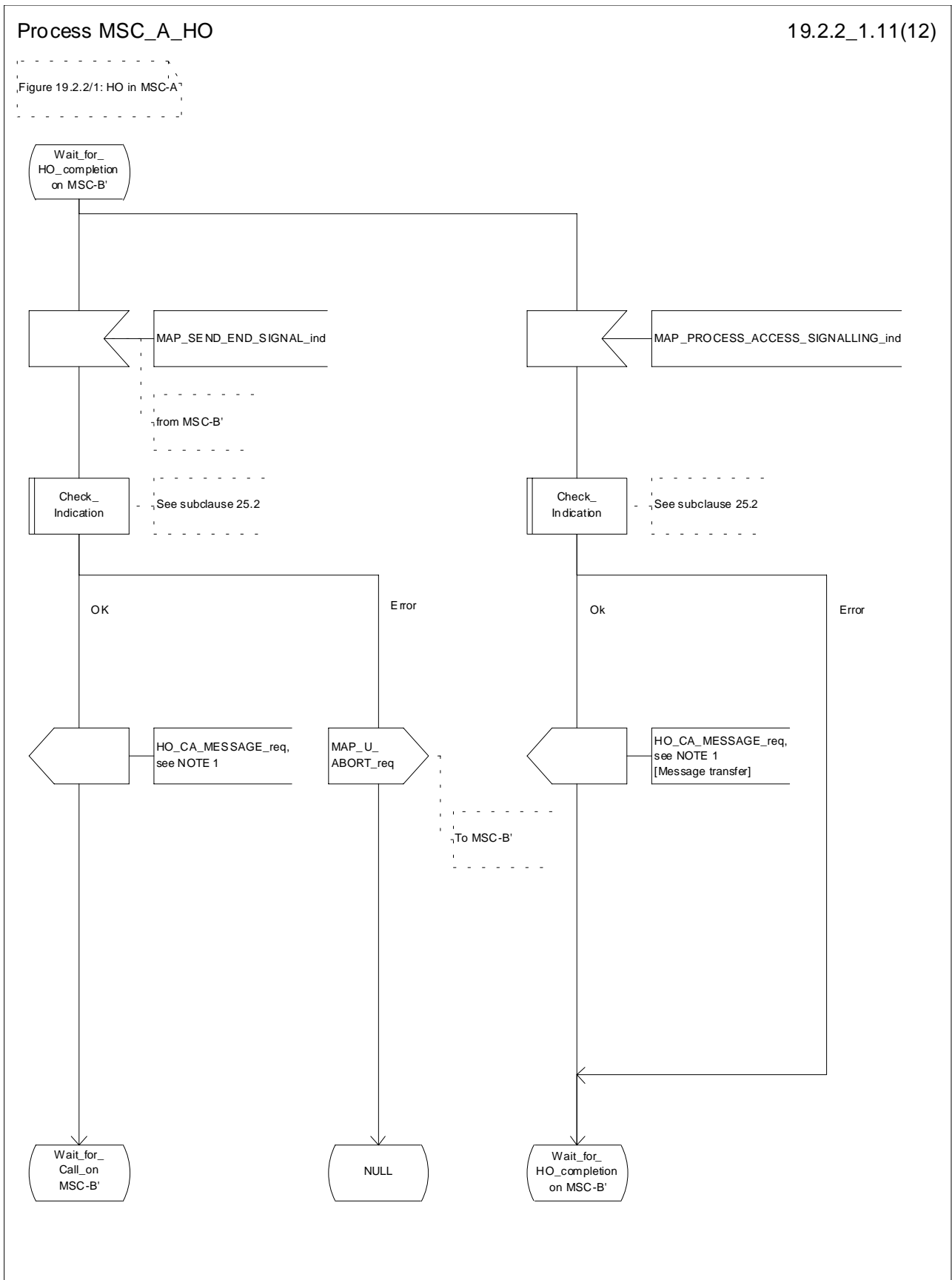


Figure 19.2.2/1 (sheet 11 of 12): Process MSC_A_HO

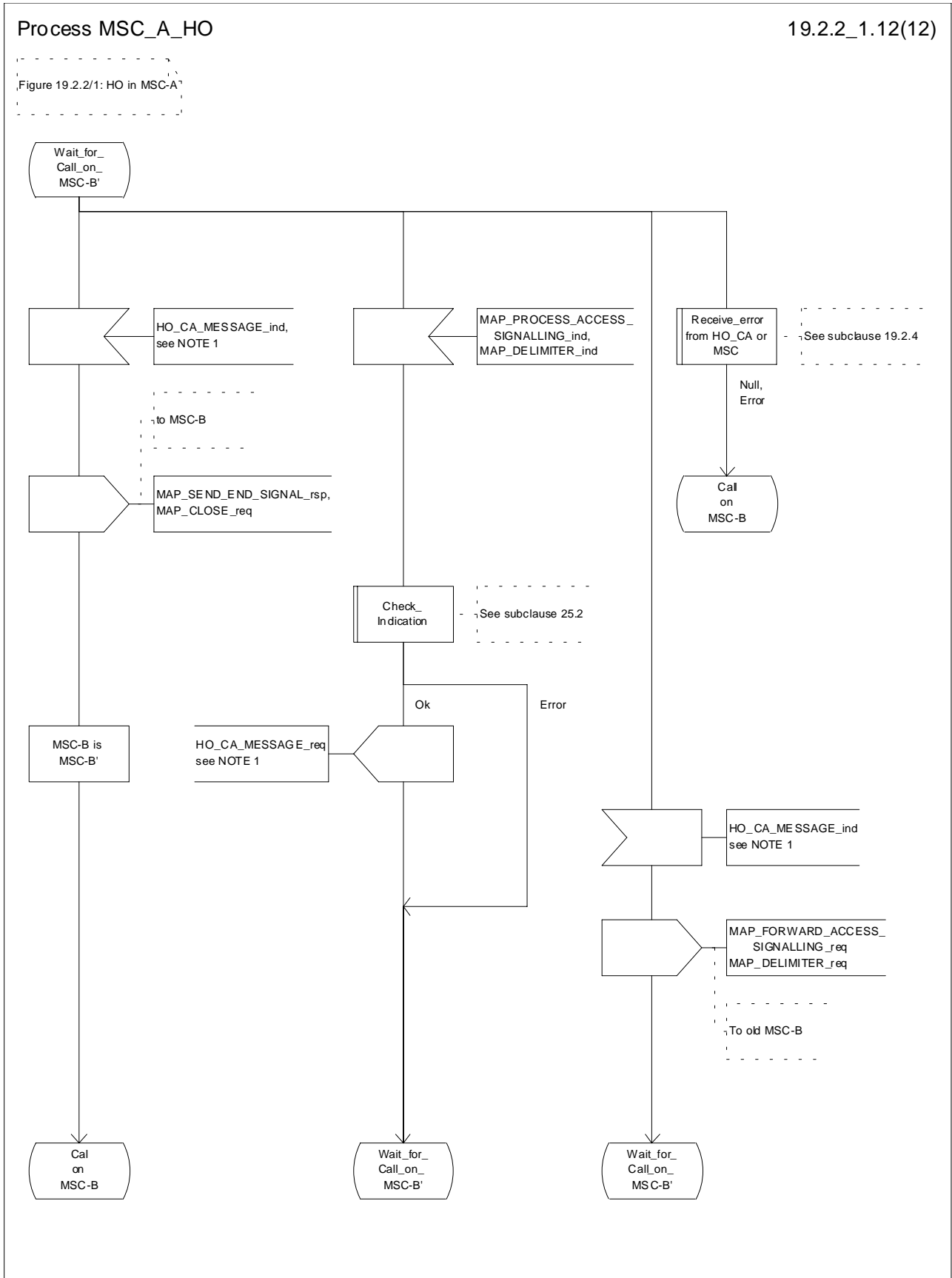


Figure 19.2.2/1 (sheet 12 of 12): Process MSC_A_HO

19.2.3 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).

19.2.3.1 Basic handover

Opening of the dialogue is described in the macro `Receive_Open_Ind` in subclause 25.1.

When MSC-B process receives a `MAP_PREPARE_HANOVER` 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.

19.2.3.2 Allocation of handover number

When a handover number is required, a `MAP_ALLOCATE_HANOVER_NUMBER` request will be sent to the VLR. The handover number is received in the `MAP_SEND_HANOVER_REPORT` request, and will be included in the `MAP_PREPARE_HANOVER` 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_HANOVER_REPORT` response.

19.2.3.3 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.

19.2.3.4 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.

19.2.3.5 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_HANOVER` 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_HANOVER_REPORT` confirmation. VLR-B will then release the allocated handover number.

The subsequent handover procedure is shown in figure 19.2/3.

19.2.3.6 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 8.4.

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.

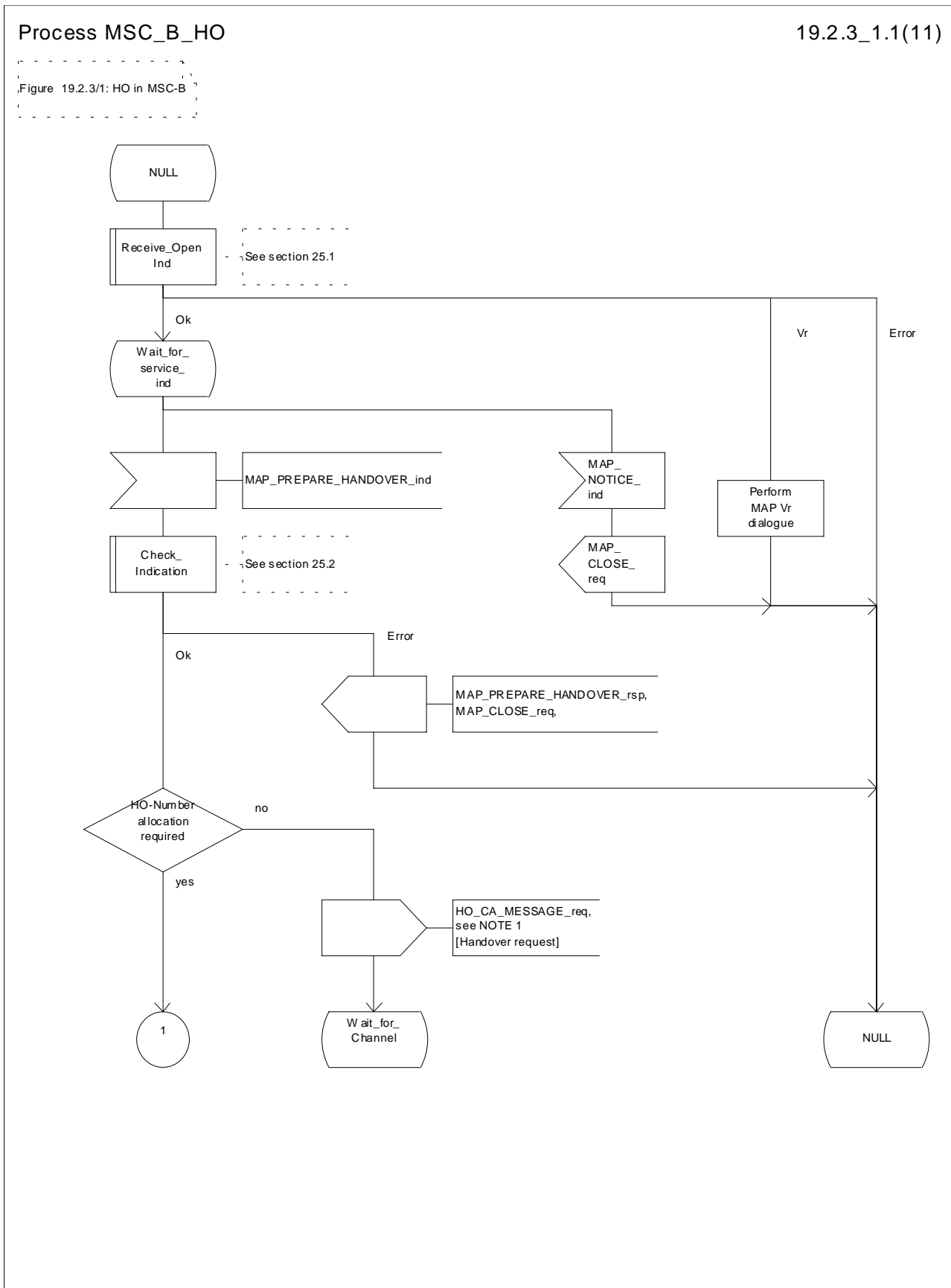


Figure 19.2.3/1 (sheet 1 of 11): Process MSC_B_HO

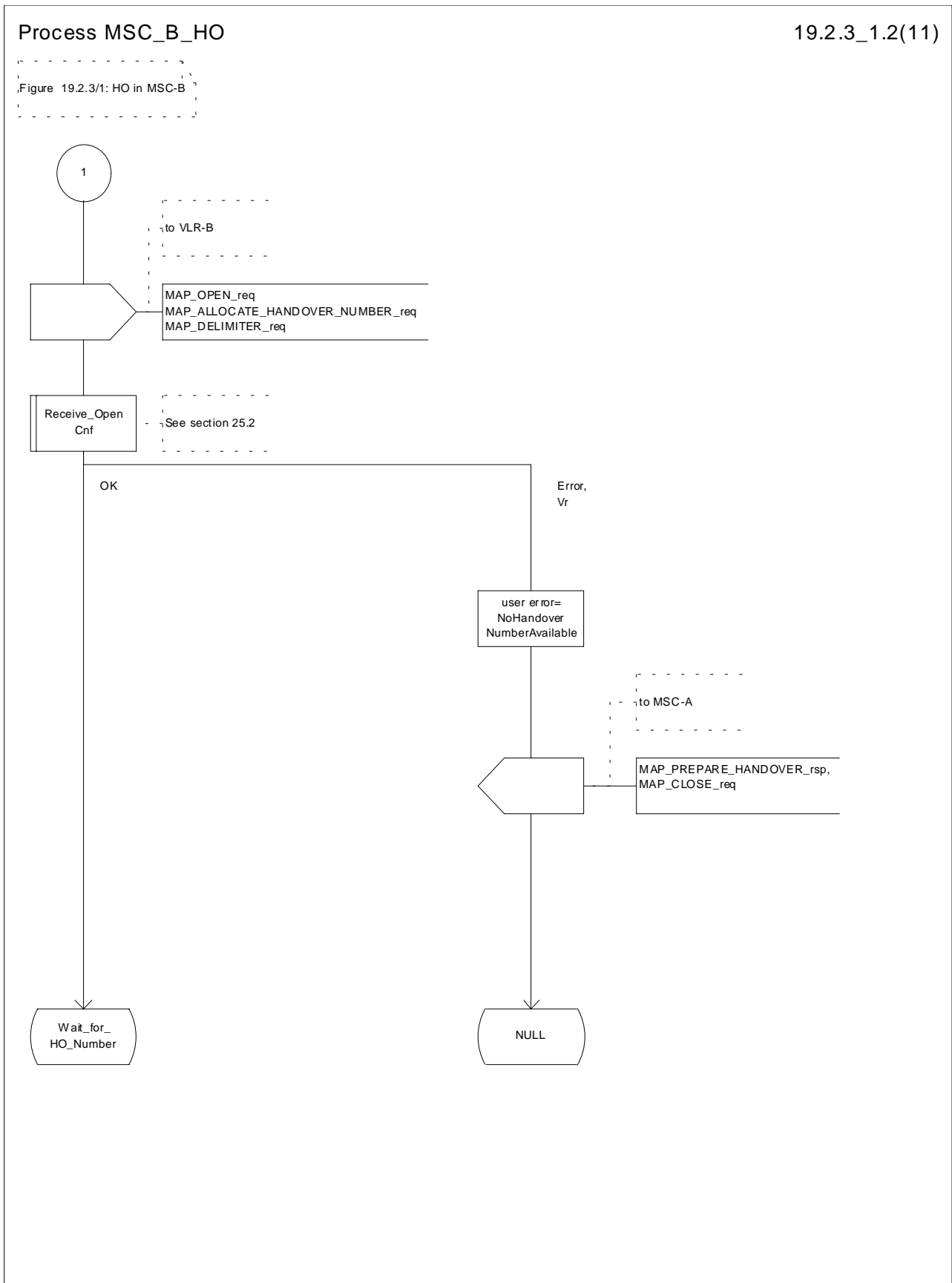


Figure 19.2.3/1 (sheet 2 of 11): Process MSC_B_HO

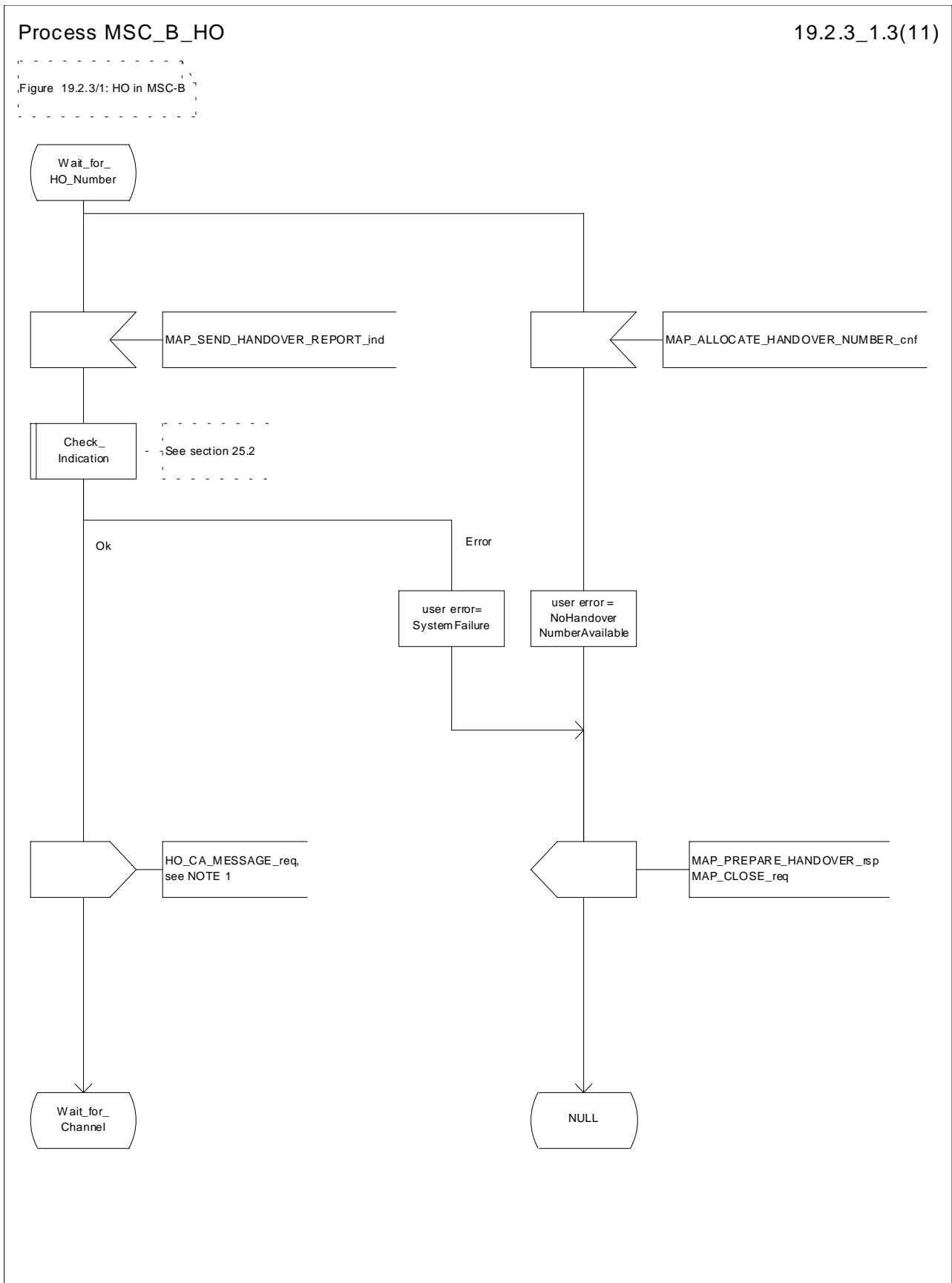


Figure 19.2.3/1 (sheet 3 of 11): Process MSC_B_HO

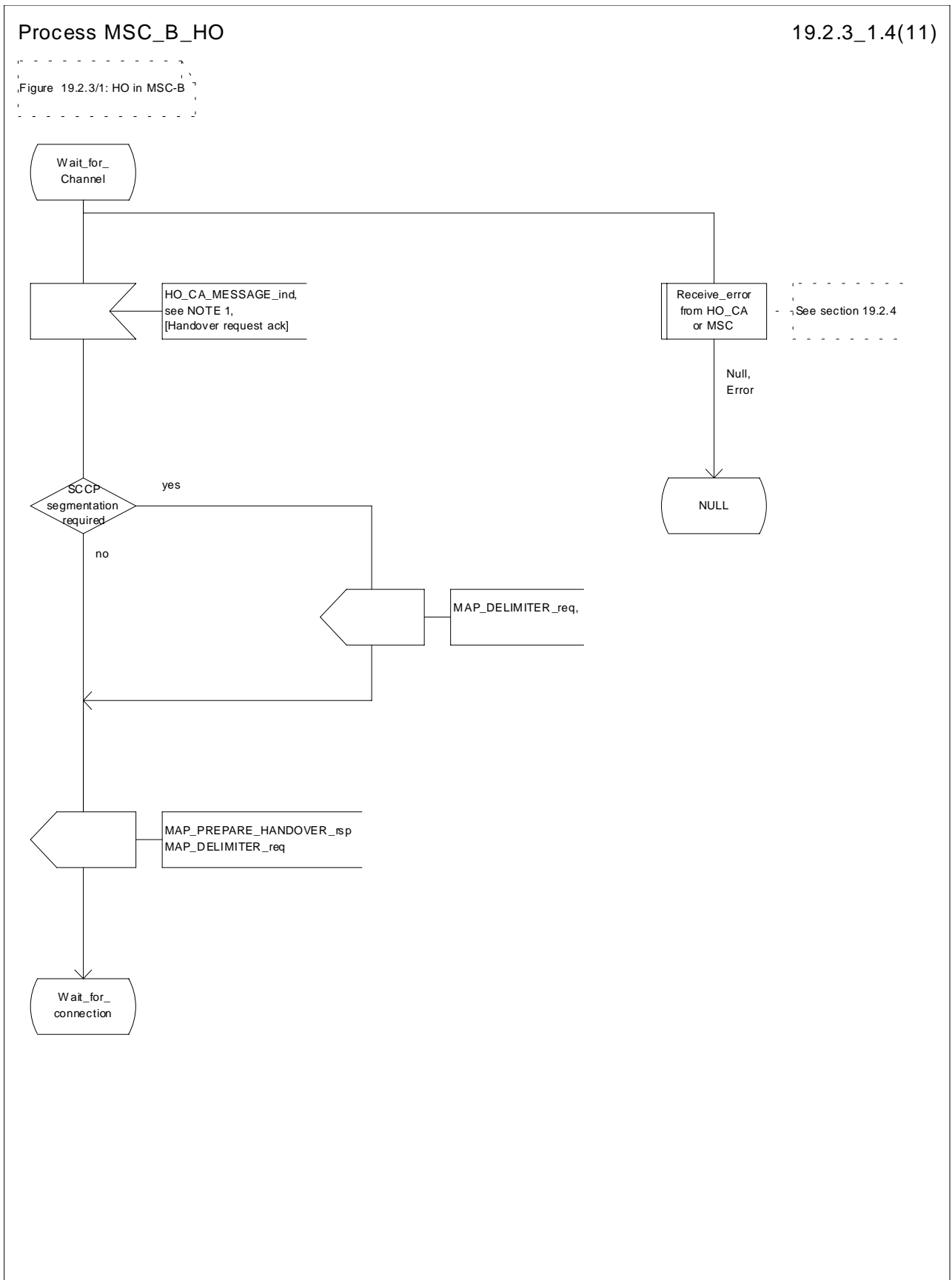


Figure 19.2.3/1 (sheet 4 of 11): Process MSC_B_HO

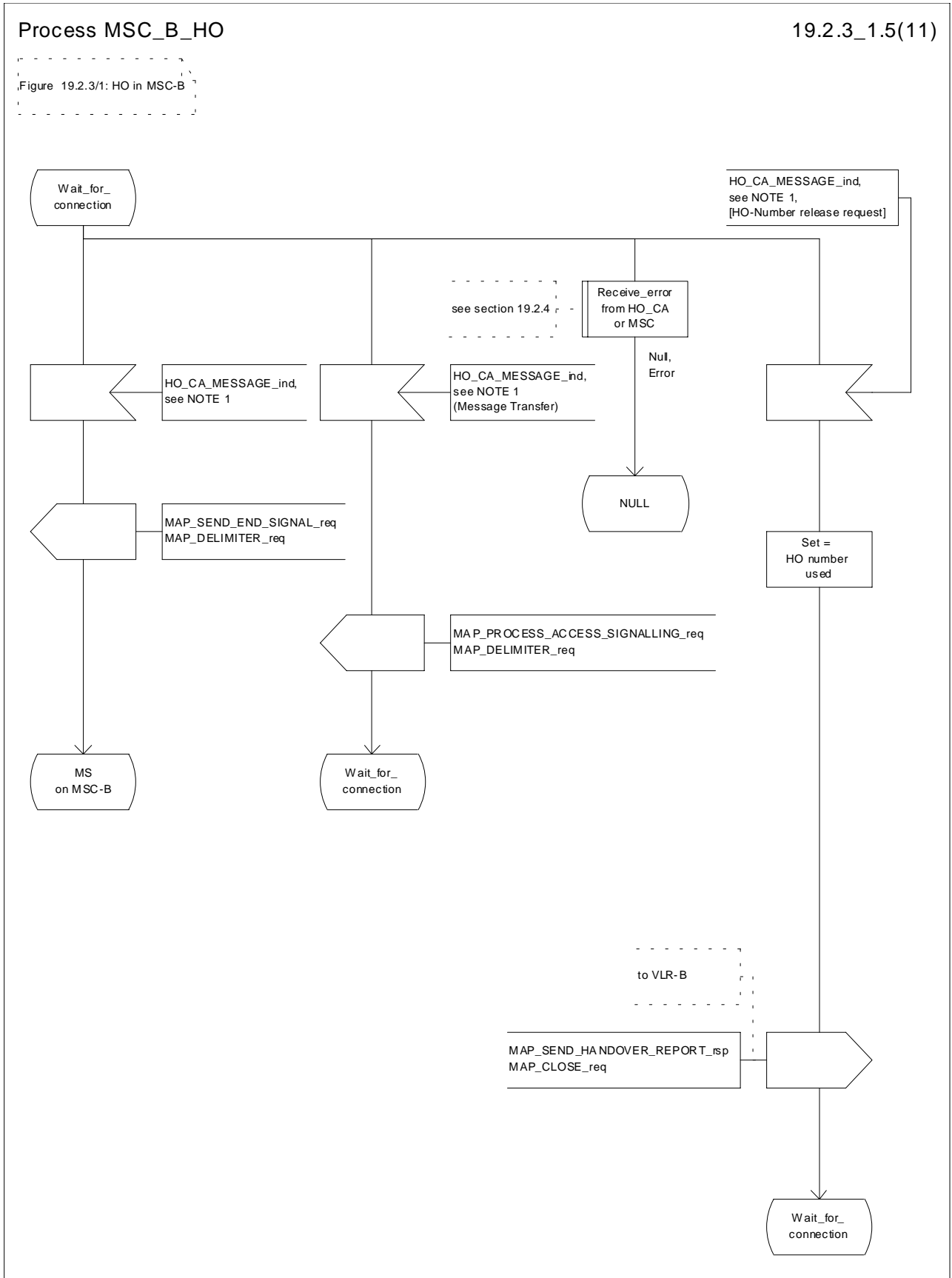


Figure 19.2.3/1 (sheet 5 of 11): Process MSC_B_HO

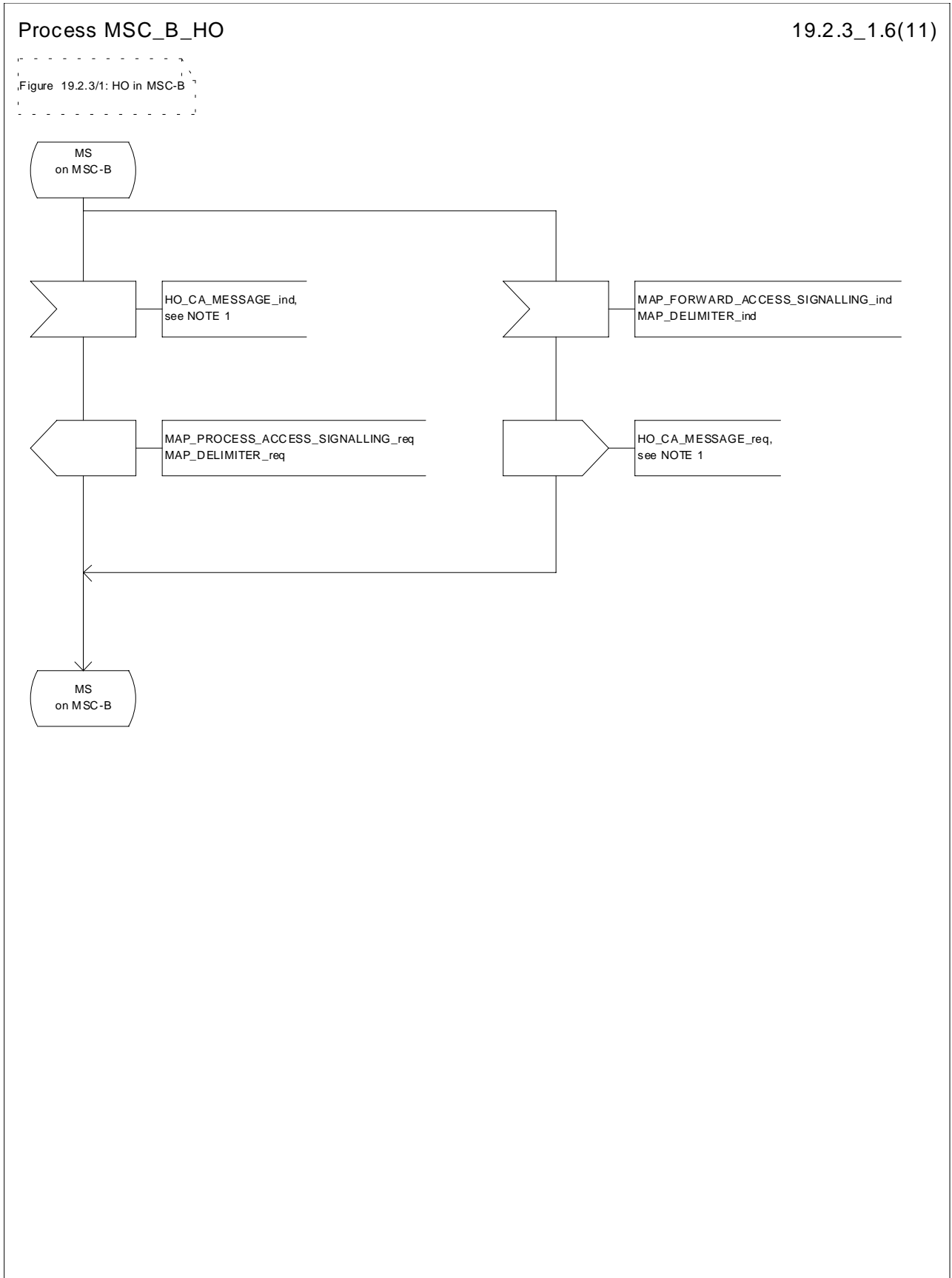


Figure 19.2.3/1 (sheet 6 of 11): Process MSC_B_HO

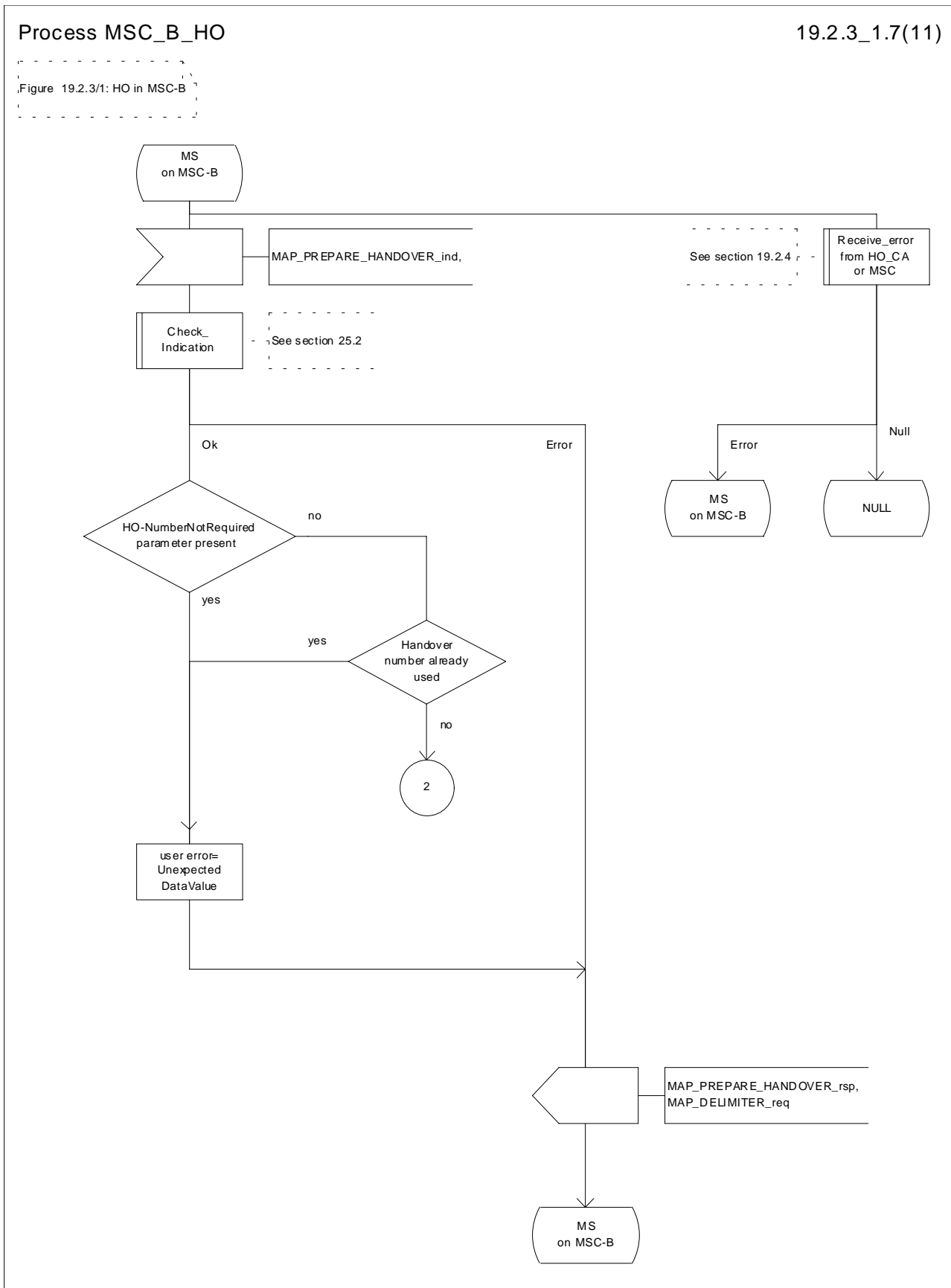


Figure 19.2.3/1 (sheet 7 of 11): Process MSC_B_HO

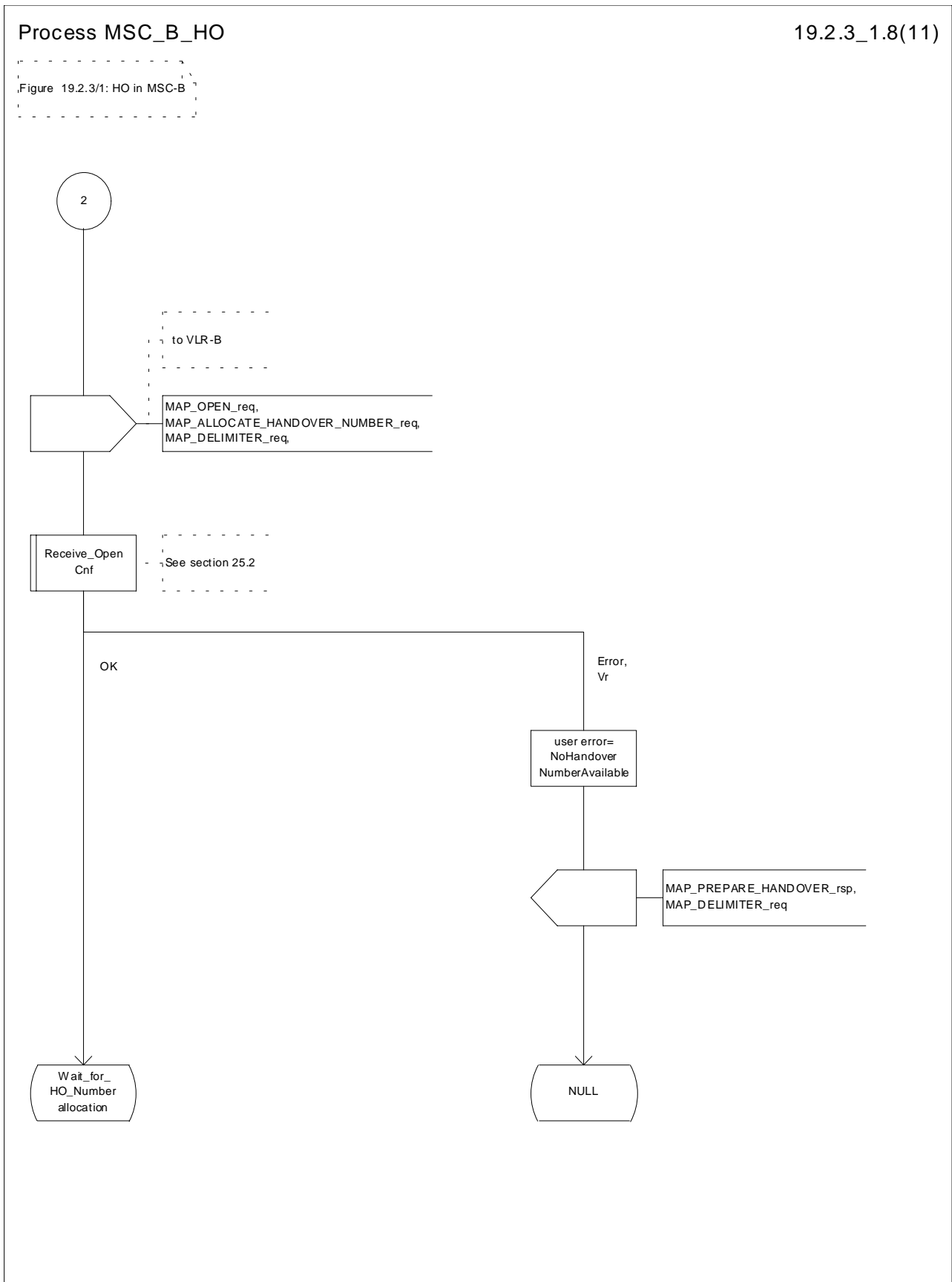


Figure 19.2.3/1 (sheet 8 of 11): Process MSC_B_HO

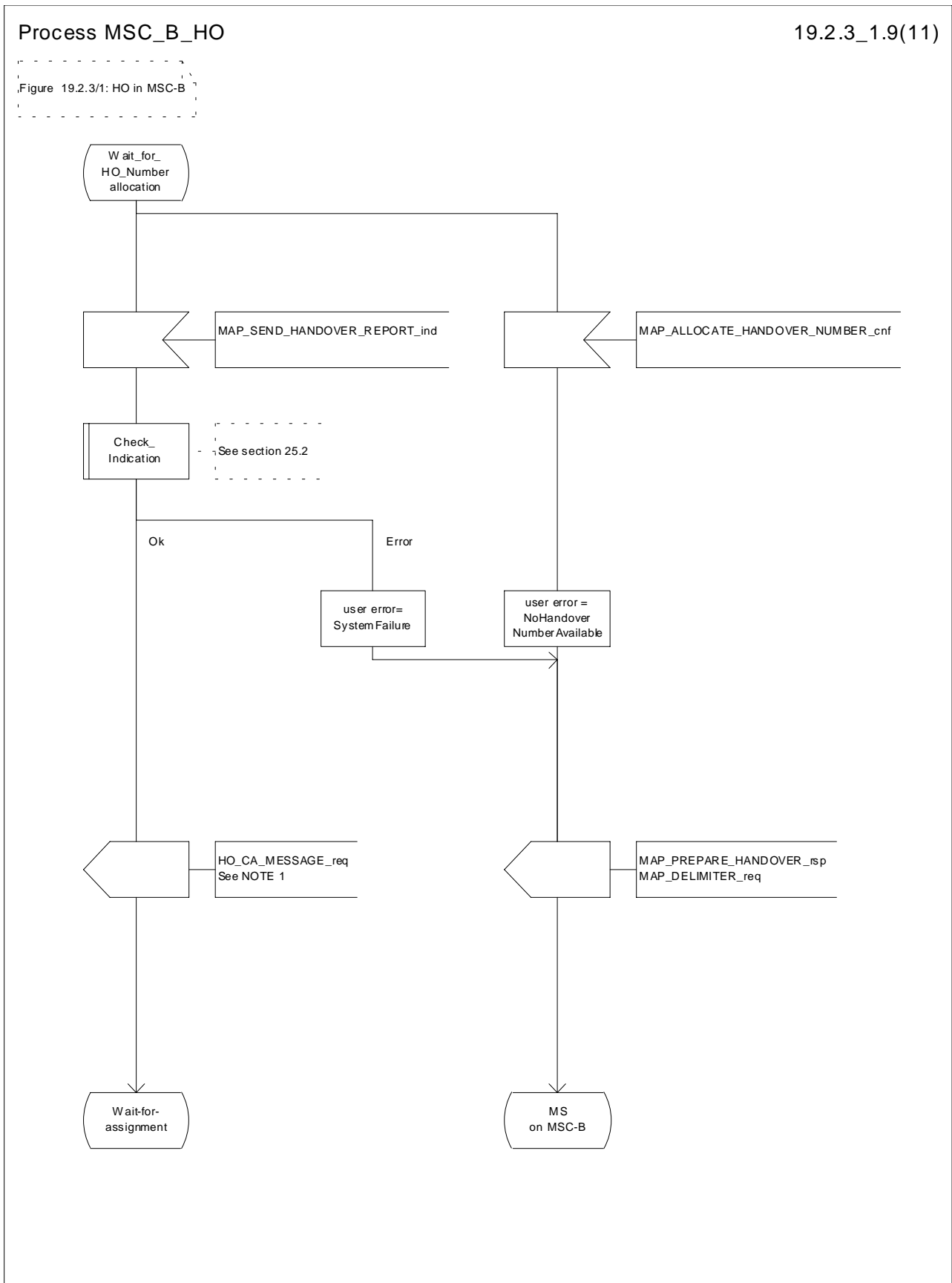


Figure 19.2.3/1 (sheet 9 of 11): Process MSC_B_HO

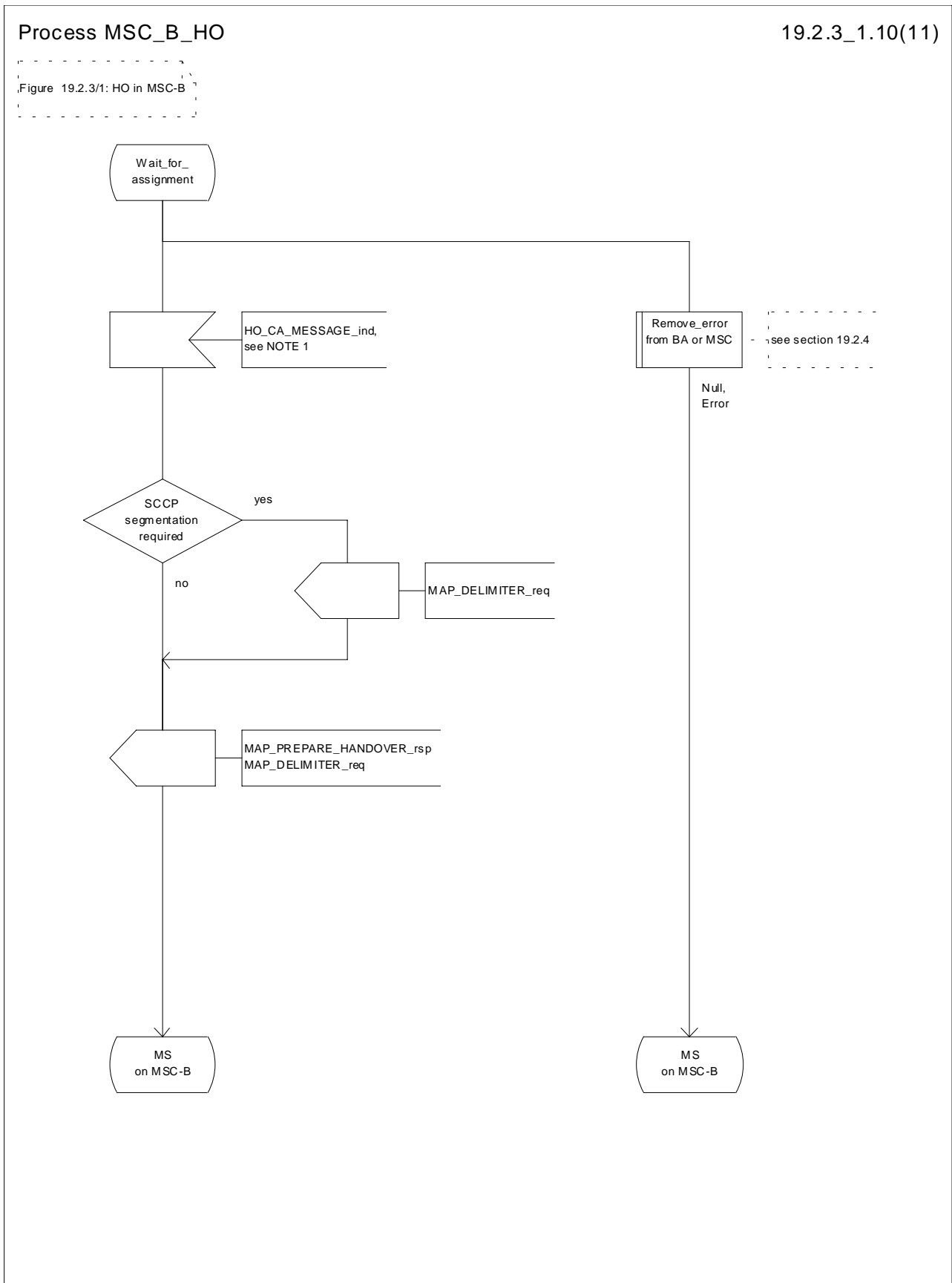


Figure 19.2.3/1 (sheet 10 of 11): Process MSC_B_HO

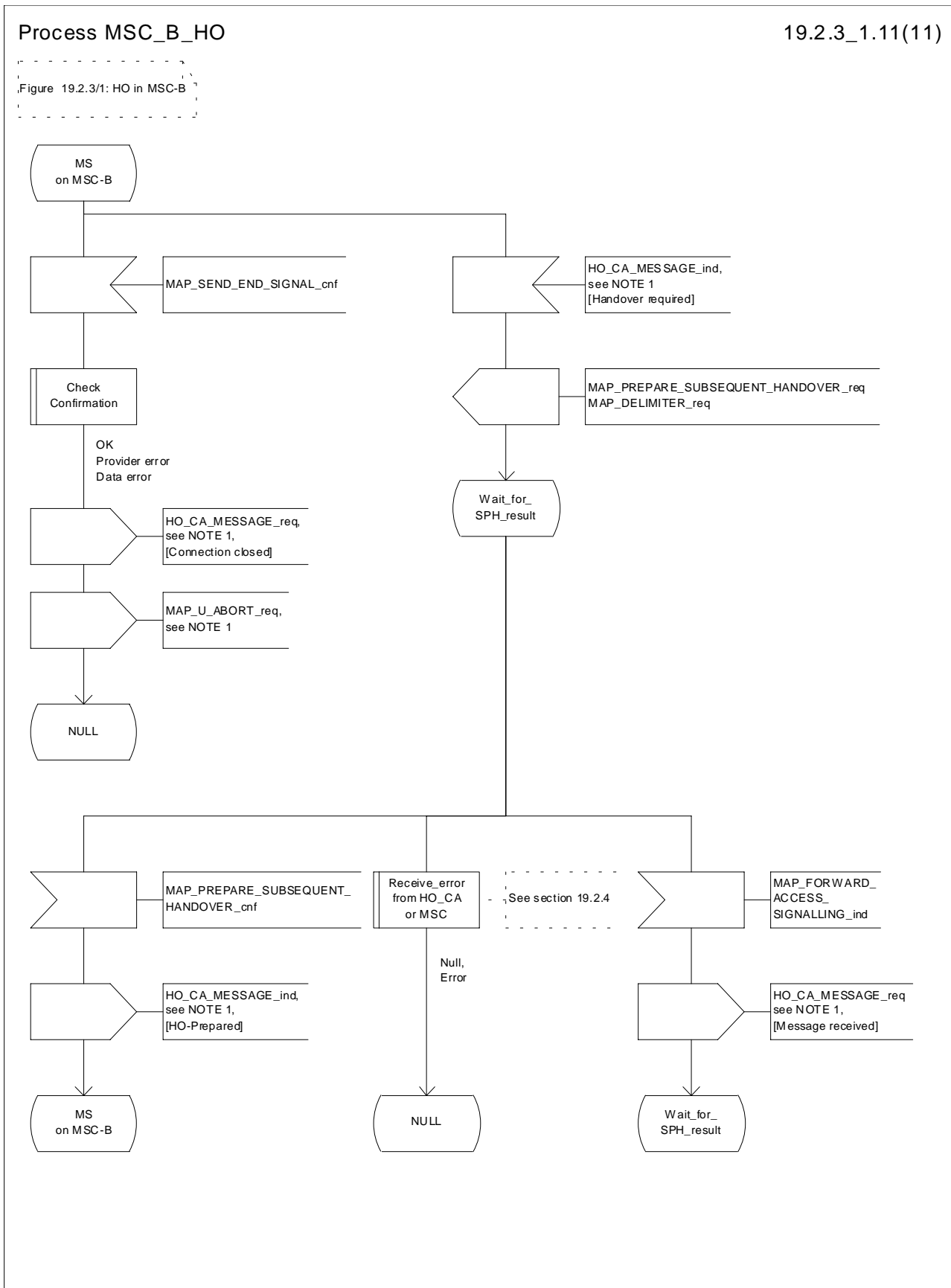


Figure 19.2.3/1 (sheet 11 of 11): Process MSC_B_HO

19.2.4 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.

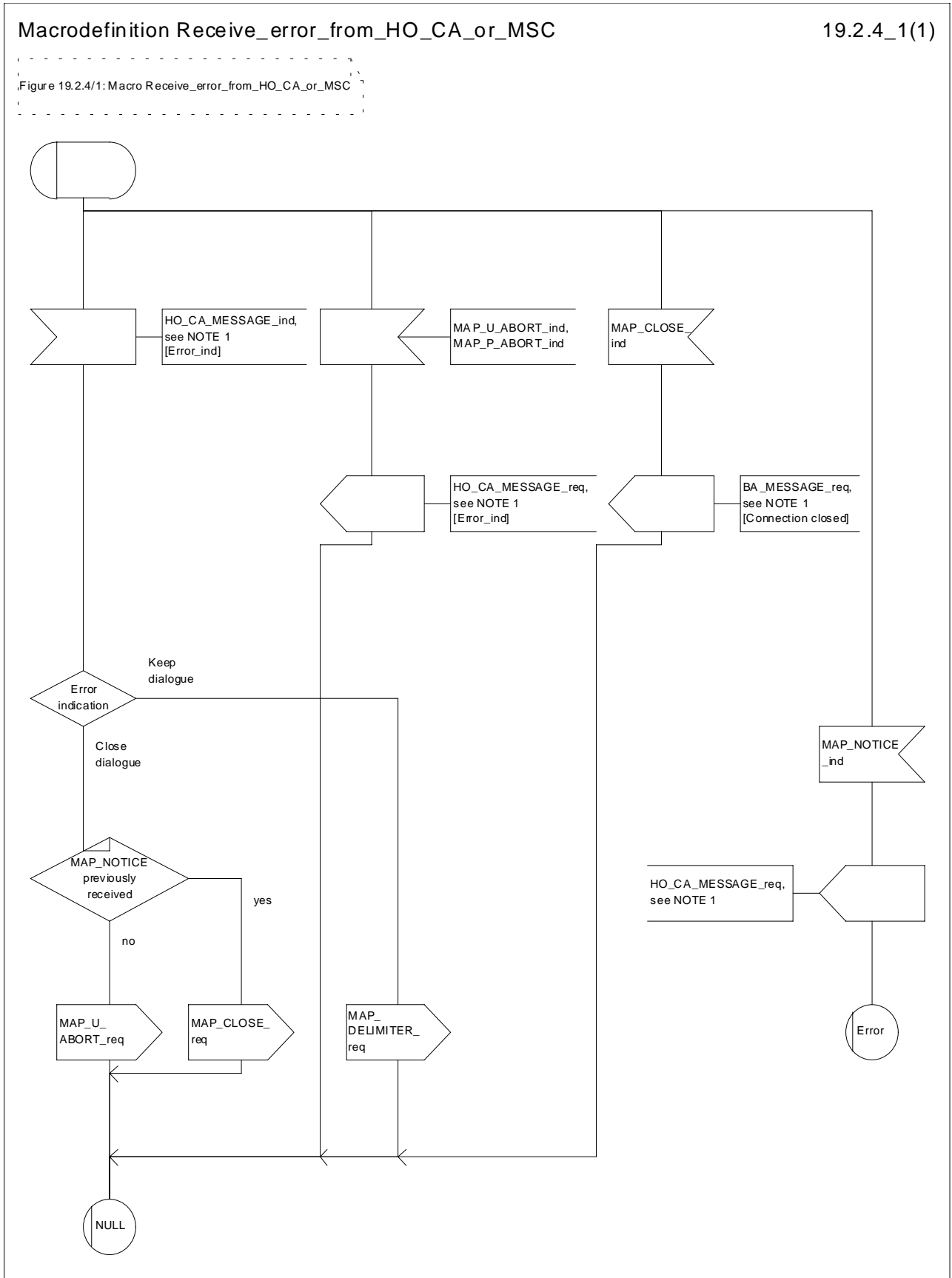


Figure 19.2.4/1: Macro Receive_error_from_HO_CA_or_MSC

19.2.5 Handover procedure in VLR

19.2.5.1 Allocation of handover number

When receiving the MAP_ALLOCATE_HANDBER_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_HANDBER_NUMBER response with the appropriate error.

The handover number allocated will otherwise be returned to MSC-B in the MAP_SEND_HANDBER_REPORT request.

The handover number will be reserved until a MAP_SEND_HANDBER_REPORT confirmation is received from MSC-B.

19.2.5.2 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 8.4.

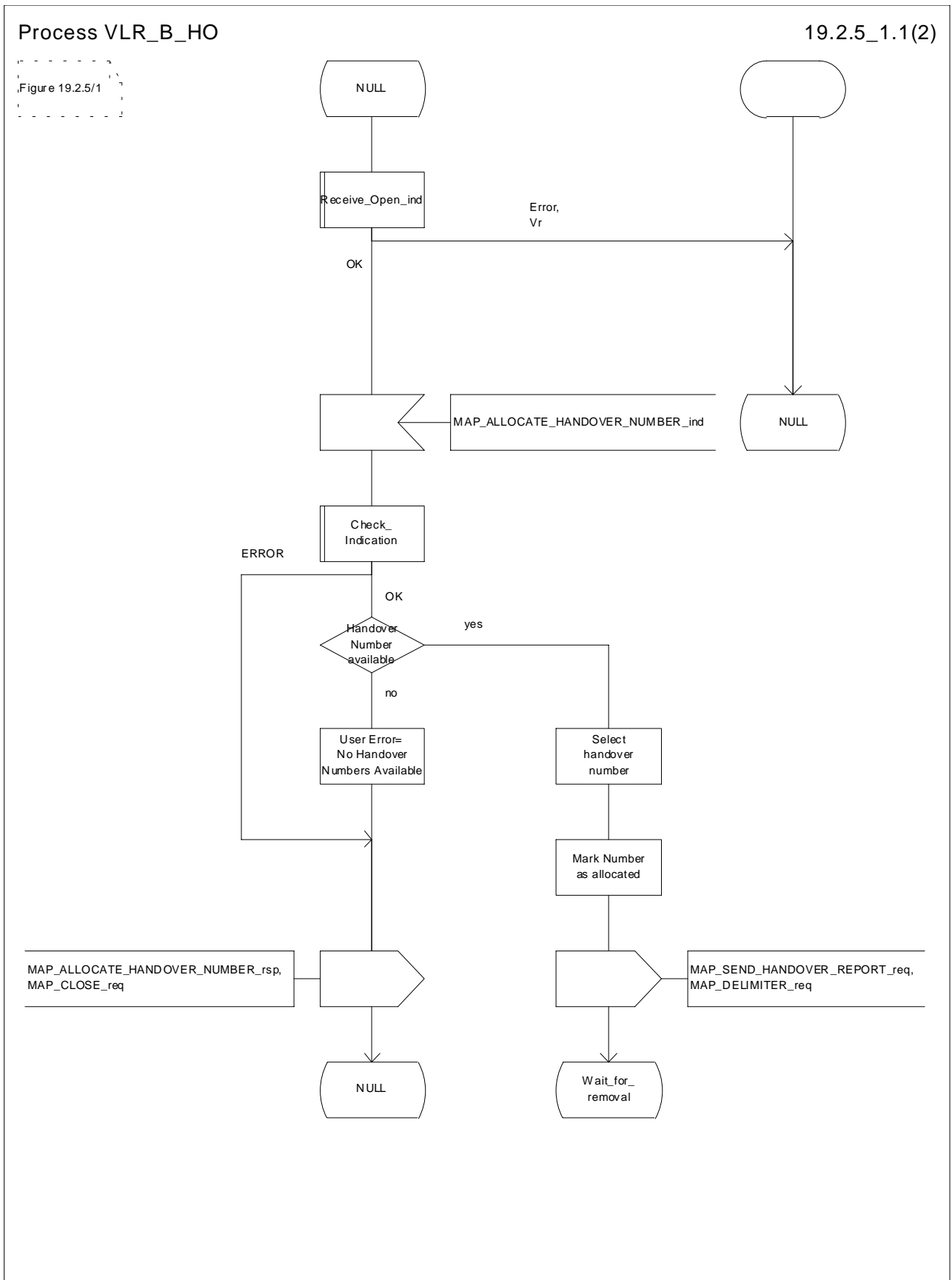


Figure 19.2.5/1 (sheet 1 of 2): Process VLR_B_HO

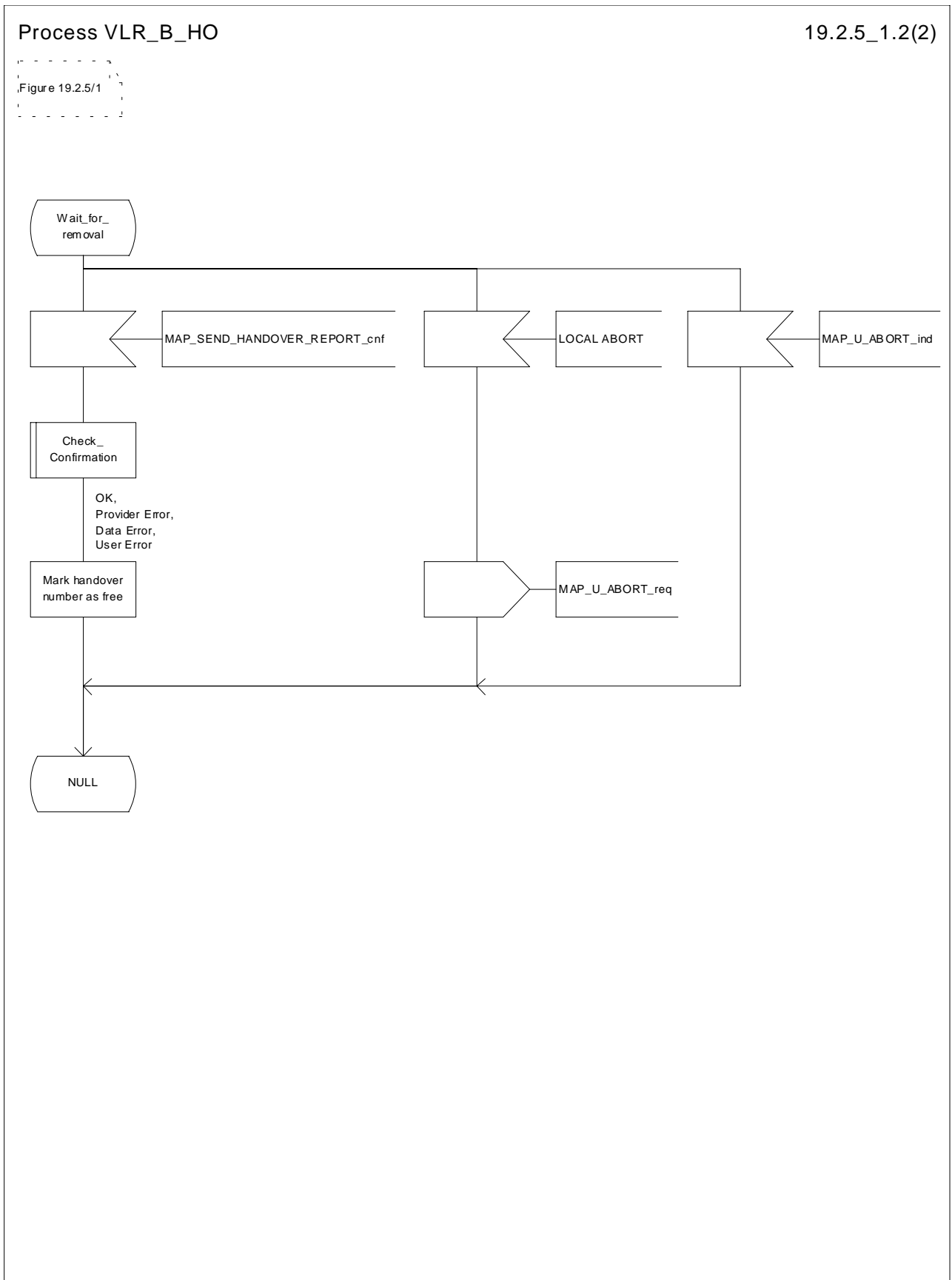


Figure 19.2.5/1 (sheet 2 of 2): Process VLR_B_HO

19.3 Fault recovery procedures

After a fault of a location register, the fault recovery procedures ensure that the subscriber data in the VLR or in the SGSN become consistent with the subscriber data that are stored in the HLR for the MS concerned and that the location information in HLR, VLR and SGSN reflect accurately the current location of the MS.

The detailed specification of fault recovery procedures of location registers is given in GSM 03.07.

19.3.1 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 19.1.1.3;
 - Update_Location_HLR subclause 19.1.1.4.
- In case of a mobile terminated call:
 - PRN_VLR subclause 21.2.4;
 - RESTORE_DATA_VLR subclause 21.2.4;
 - RESTORE_DATA_HLR subclause 19.3.3;
 - ICS_VLR subclause 21.3.3.

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 19.1.1/6).

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 19.1.

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 19.3/1 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 21.2.4.

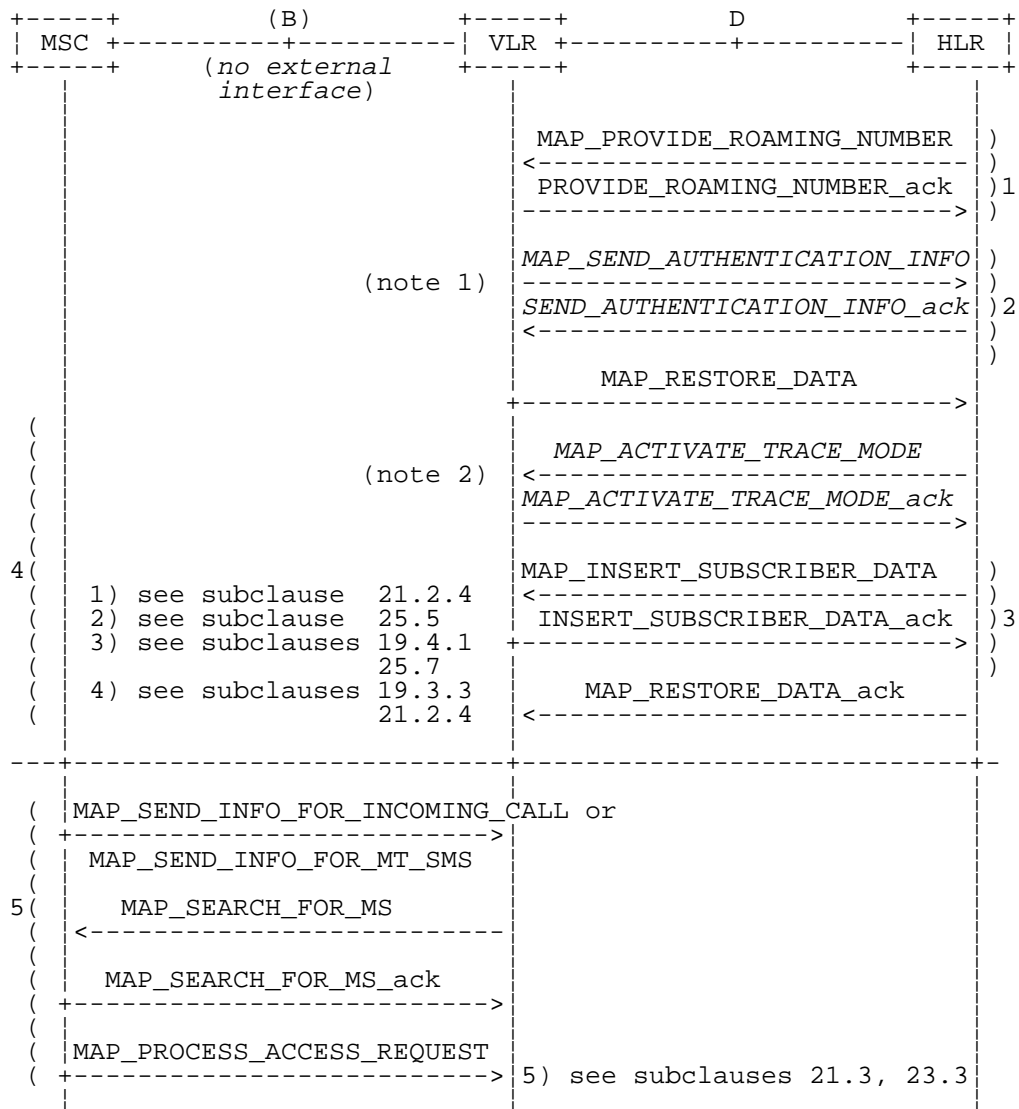
The MAP_RESTORE_DATA process in the HLR is described in subclause 19.3.3.

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 19.3/1) 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.



NOTE 1: If authentication required.

NOTE 2: If subscriber tracing active in HLR.

Figure 19.3/1: Procedures related to restoration of VLR in case of mobile terminated call set-up

19.3.2 HLR fault recovery procedures

The following processes are involved with the restart of the HLR:

- HLR_RESTART subclause 19.3.2;
- REC_RESET_IN_VLR subclause 19.3.2;
- REC_RESET_IN_SGSN subclause 19.3.2.

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 19.1.1.3;
- Update_Location_HLR subclause 19.1.1.4;
- Update_GPRS_Location_HLR subclause 19.1.1.4;

- GPRS_Update_Location_Area_VLR subclause 19.1.1.3;
- SGSN_Update_HLR subclause 19.1.1.8.

In the case of a mobile originated service request, the

- Macro Process_Access_Request_VLR subclause 25.4.2; and the
- Process_Update_Location_HLR subclause 19.1.1.4,

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, SGSN number and, if provided by the VLR, LMSI in the HLR. Consistency of subscriber data that are stored in the VLR or in the SGSN 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. If the Gs-interface is present the VLR shall not forward this notification.

Figures 19.3/2 and 19.3/9 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 19.3/3):

- 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 19.3/4).
- send a MAP_RESET request to the SGSNs where its MSs are located (see figure 19.3/7).

The MAP_RESET request contains the HLR number and optionally the HLR Identity List.

When receiving a MAP_RESET indication, the VLR or the SGSN will derive all involved MSs of that HLR either from the HLR Identity List (if present), or from the HLR number. The VLR or the SGSN 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 figures 19.3/5 and 19.3/8).

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.

Also the status "Not Confirmed" of the indicator "Location Information Confirmed in HLR" forces the SGSN to invoke the MAP_UPDATE_GPRS_LOCATION service after establishment of authenticated radio contact with the MS concerned.

The MAP_UPDATE_LOCATION procedure is performed as described in subclause 19.1.

After receipt of the MAP_UPDATE_LOCATION or the MAP_UPDATE_GPRS_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.

If the MAP_UPDATE_GPRS_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 SGSN 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_GPRS_LOCATION request.

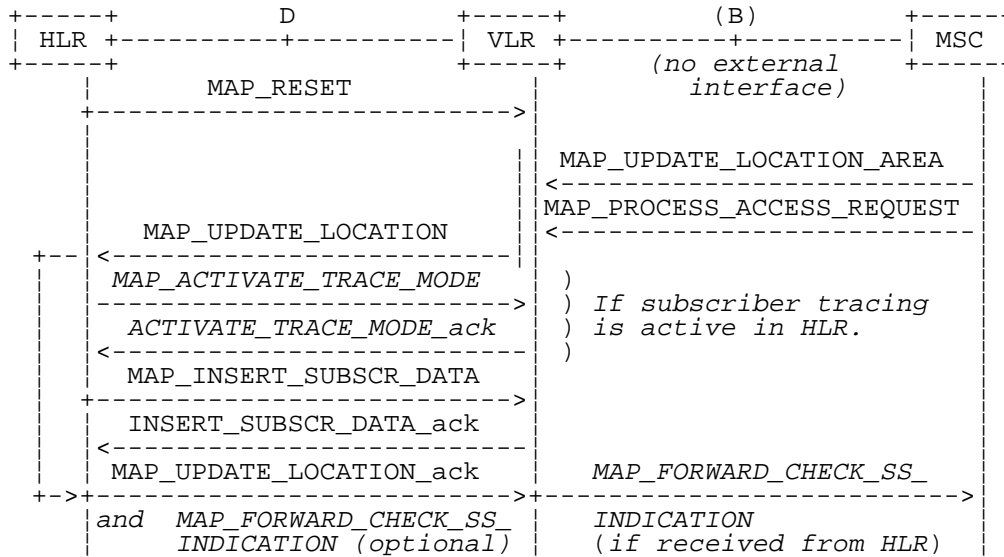


Figure 19.3/2: Procedures related to restoration of HLR

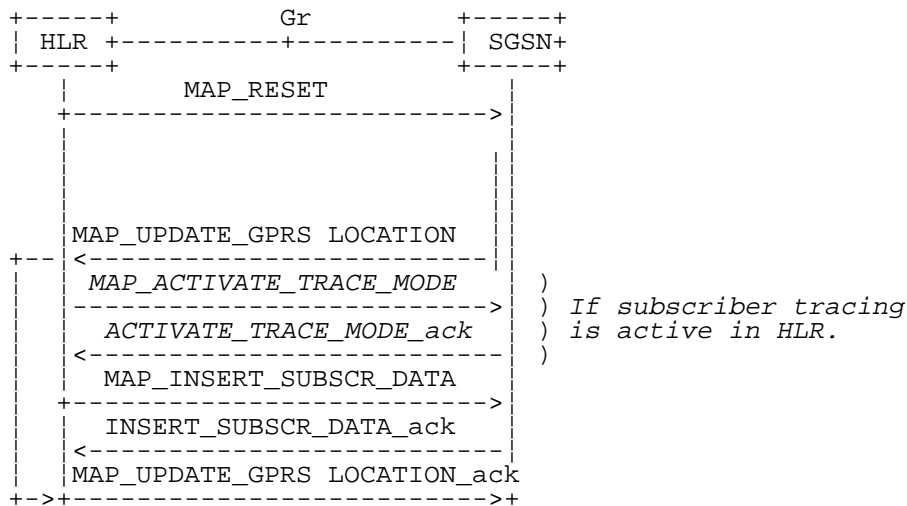


Figure 19.3/9: Procedures related to restoration of HLR for GPRS

Process HLR_RESTART

19.3_3(1)

Figure 19.3/3 Restoration of the HLR
Application process in the HLR for HLR Restart

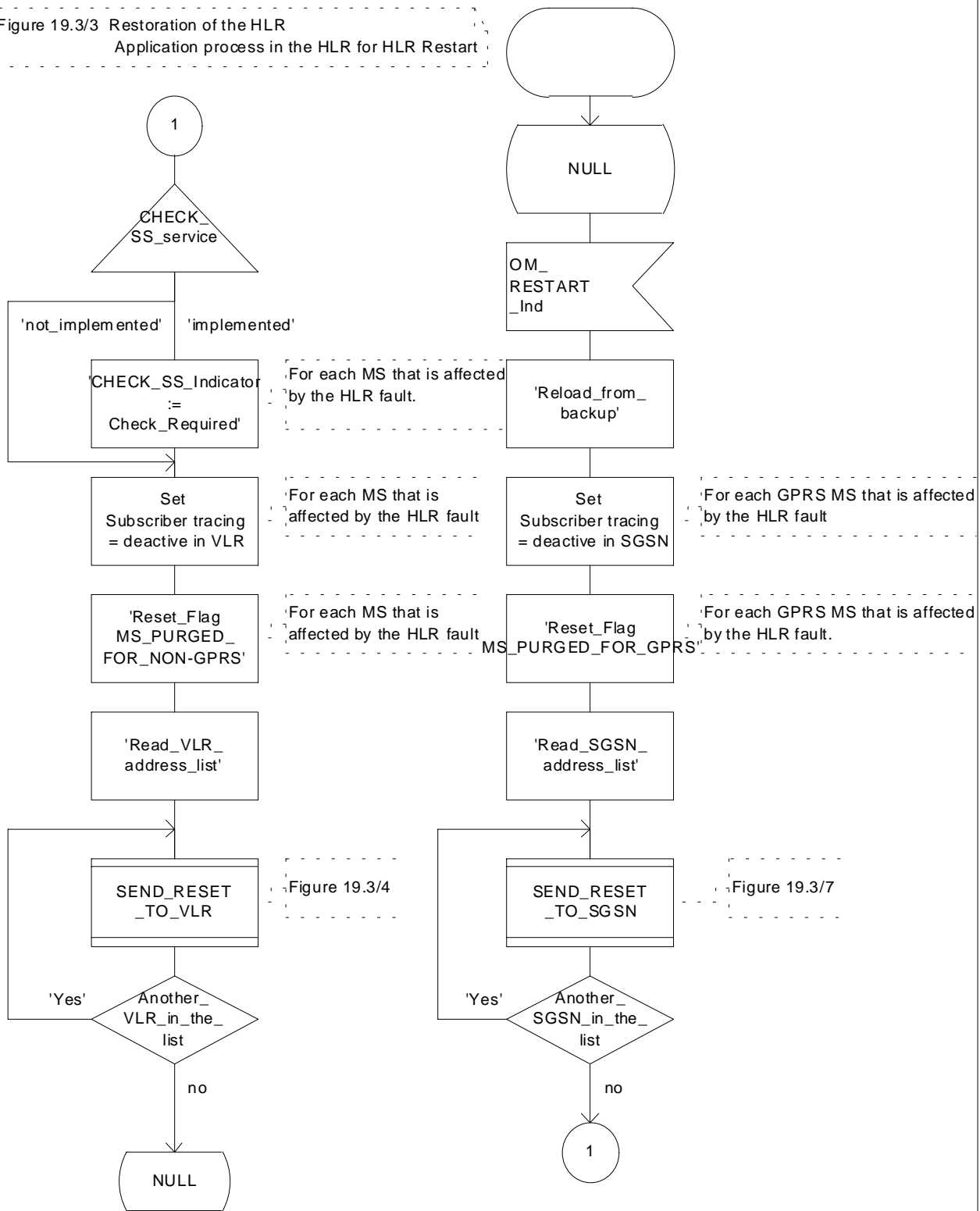


Figure 19.3/3: Process HLR_RESTART

Process SEND_RESET_TO_VLR

19.3_4(1)

Figure 19.3/4 Restoration of the HLR
Process for sending the RESET message
from HLR to VLR

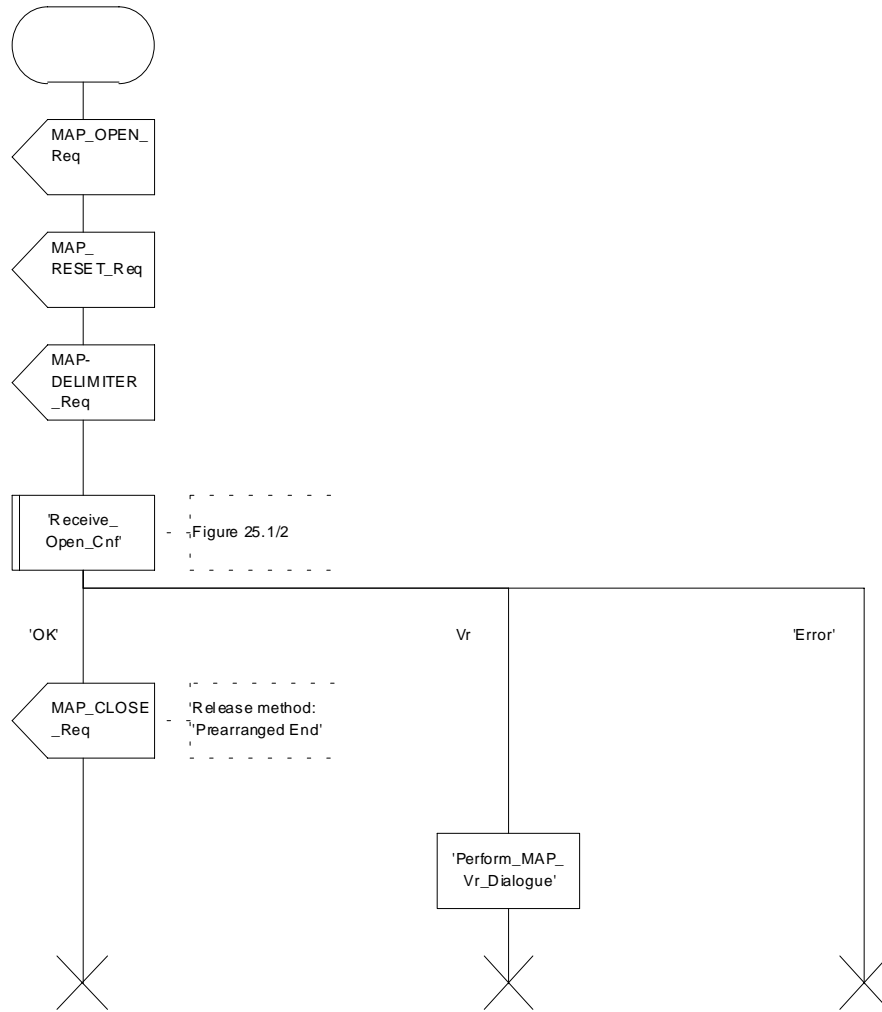


Figure 19.3/4: Process SEND_RESET_TO_VLR

Process REC_RESET_IN_VLR

19.3_5(1)

Figure 19.3/5 Restoration of the HLR - Application process in the VLR for reception of the RESET message from HLR

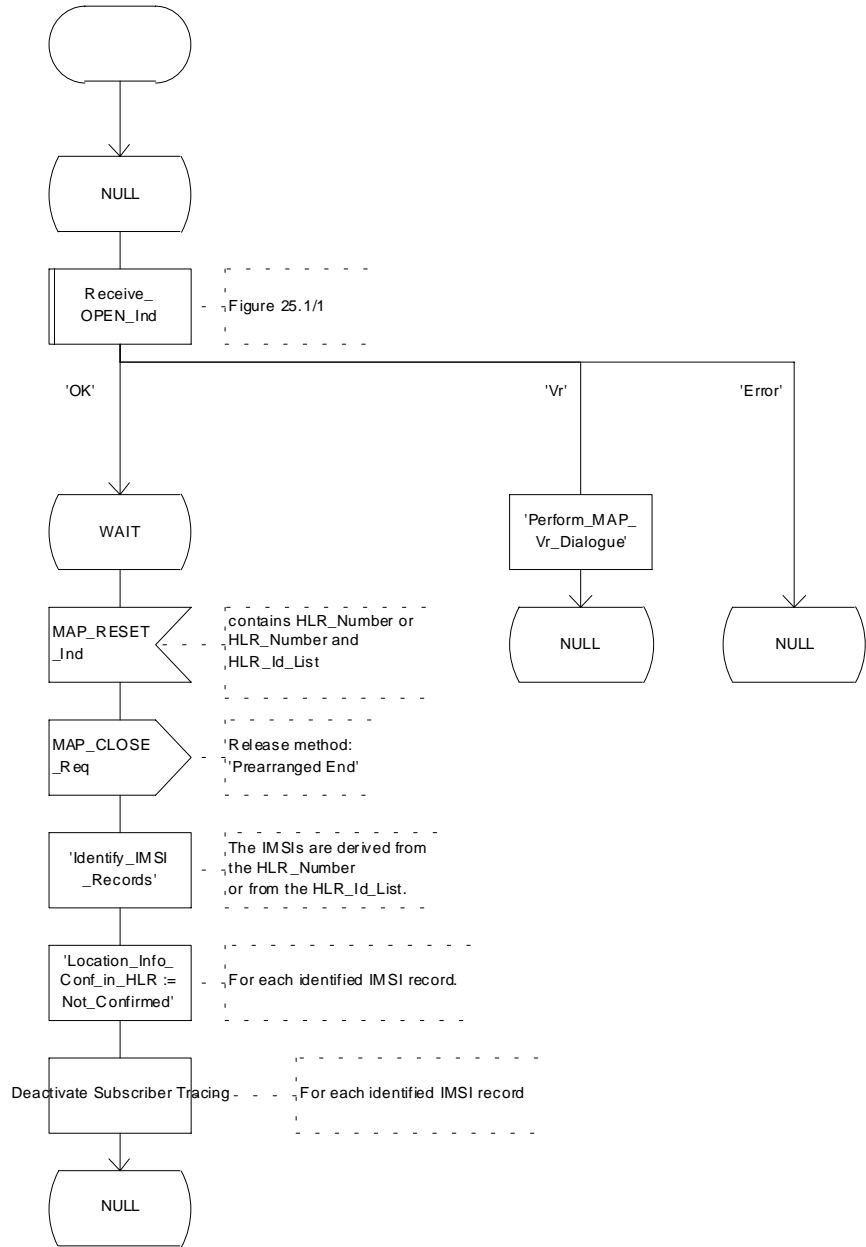


Figure 19.3/5: Process REC_RESET_IN_VLR

Process SEND_RESET_TO_SGSN

19.3_7(1)

Figure 19.3/7: Restoration of the HLR
Process for sending the RESET message
from HLR to SGSN

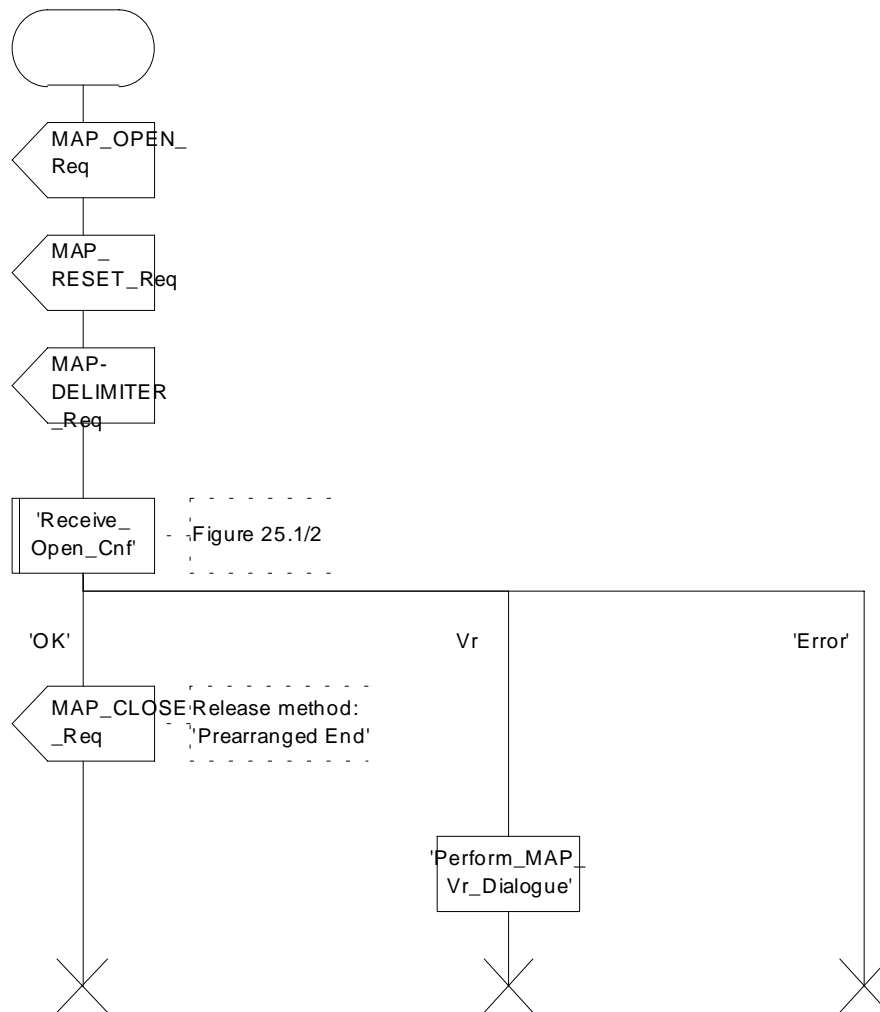


Figure 19.3/7: Process SEND_RESET_TO_SGSN

Process REC_RESET_IN_SGSN

19.3_8(1)

Figure 19.3/8: Restoration of the HLR - Application process in the SGSN for reception of the RESET message from HLR

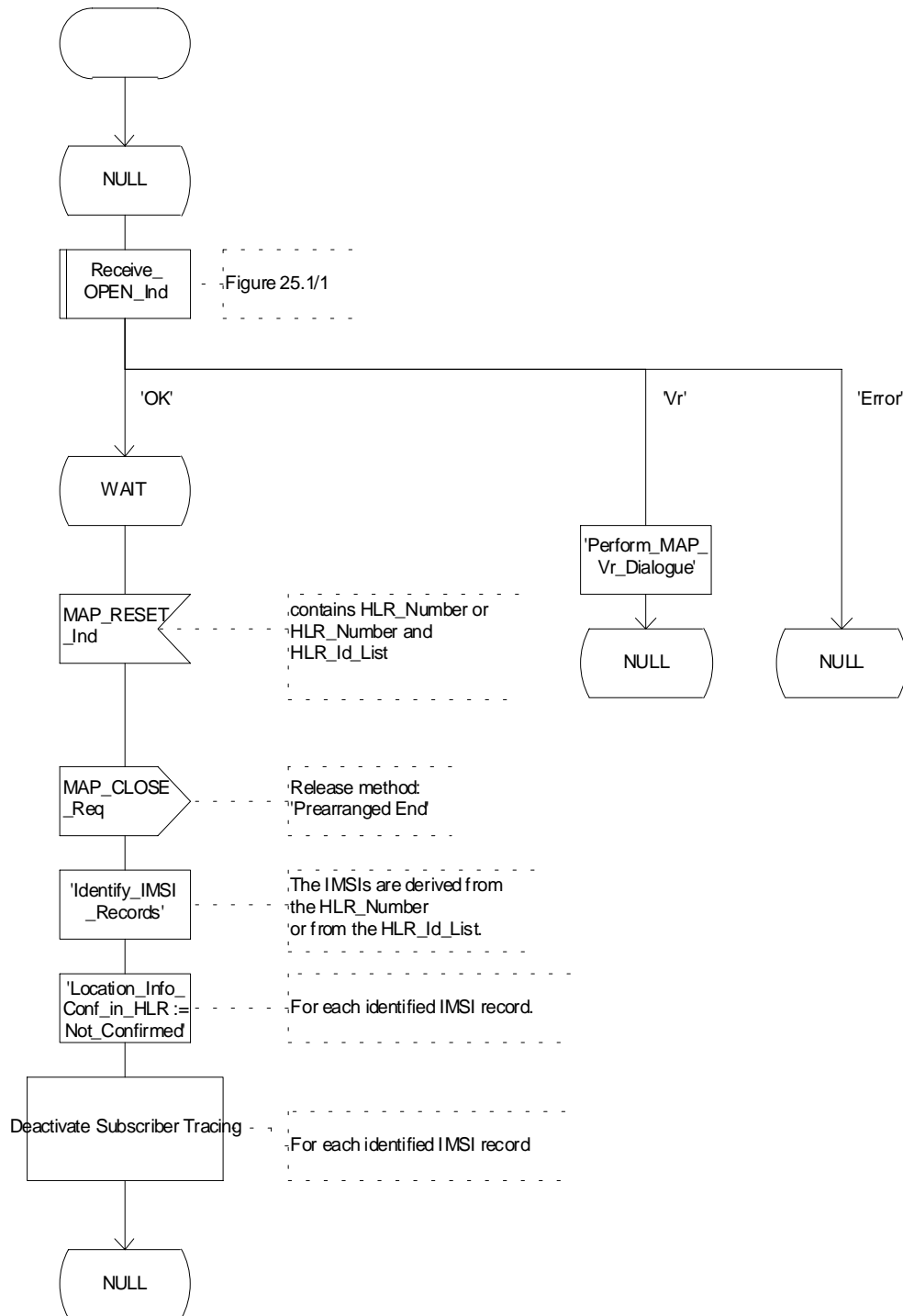


Figure 19.3/8: Process REC_RESET_IN_SGSN

19.3.3 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 21.2.4.

The process RESTORE_DATA_HLR makes use of the following macros:

- Receive_Open_Ind subclause 25.1.1;
- Check_Indication subclause 25.2.1;
- Insert_Subscriber_Data_Framed_HLR subclause 19.4.1.

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 19.3/1). The process RESTORE_DATA_VLR is shown in figure 21.2/6.

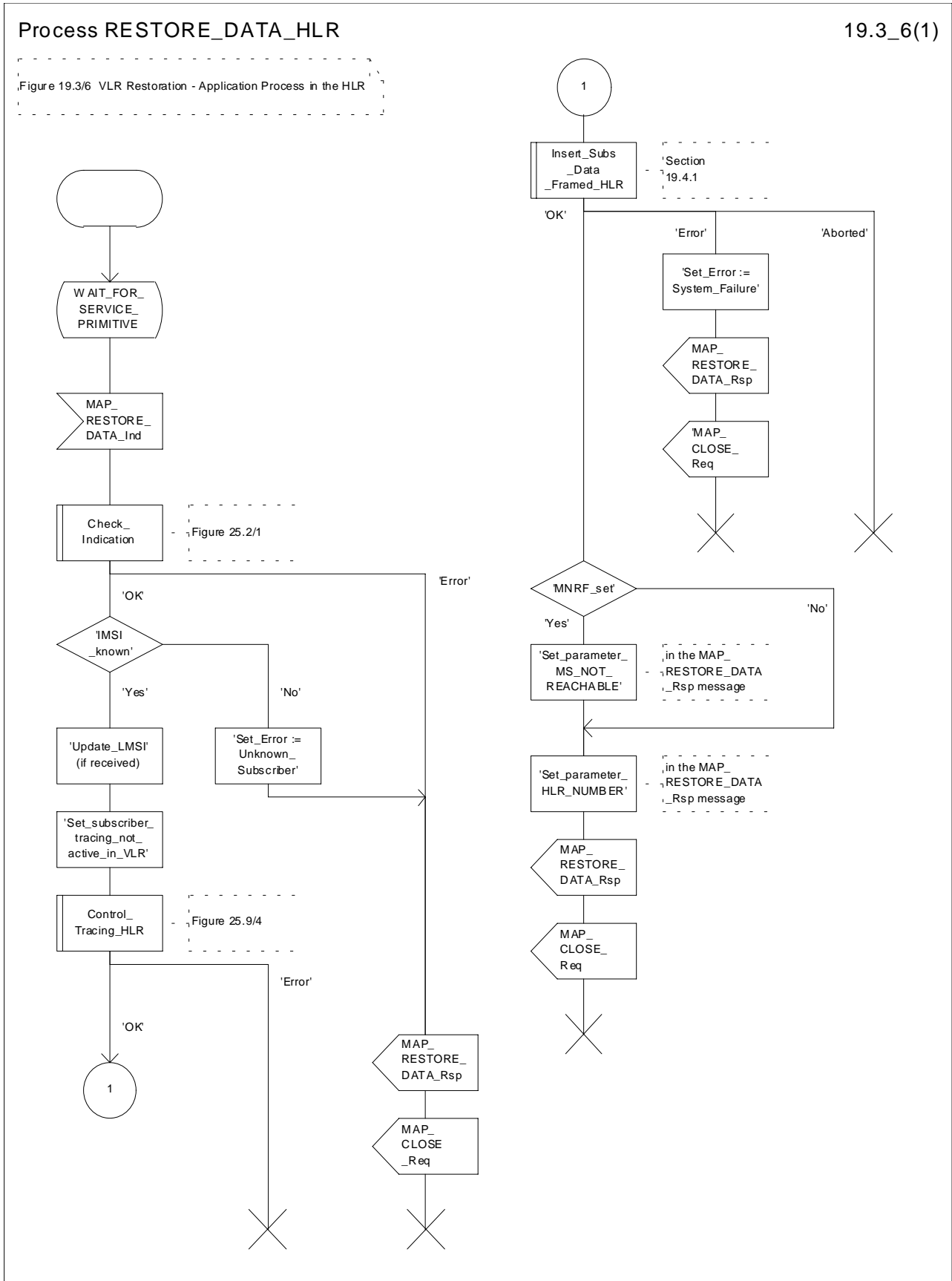
The restore data process in the HLR is activated by receipt of a MAP_RESTORE_DATA indication from the VLR (see figure 19.3/6). If there is a parameter problem in the indication, either of the errors "Unexpected Data Value" or "Data Missing" is returned in the MAP_RESTORE_DATA response; if the subscriber 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 25.9. Thereafter, the macro "Insert_Subscriber_Data_Framed_HLR" that is described in subclause 19.4.1 is invoked. The outcome of the macro Insert_Subscriber_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 21.2.4, figure 21.2/6.



19.4 Macro Insert_Subscriber_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_Subscriber_Data" (see subclause 25.7.4) 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 25.7.3). There the reception and handling of the service confirmations is described.

If certain services required for a subscriber are not supported by the VLR or by the SGSN (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.
- The HLR stores and sends "Roaming Restricted in the SGSN Due To Unsupported Feature" in a subsequent MAP_INSERT_SUBSCRIBER_DATA service. If "Roaming Restricted In SGSN Due To Unsupported Feature" is stored in the HLR, the "SGSN Area Restricted Flag" shall be set to "restricted". This will prevent MT SM from being forwarded to the SGSN and Network Requested PDP-Context Activation;

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".

If subscriber data for CAMEL Phase 2 or 3 services are sent to a VLR which does not support CAMEL Phase 2 or 3, the service behaviour may be unpredictable or incorrect. The HLR therefore needs to ensure that at the conclusion of a location updating dialogue the data in the VLR do not require a capability that the VLR does not have. Possible mechanisms to ensure this are described in 3G TS 23.078.

The HLR should send a Forwarded-to number which is not in E.164 international format to the VLR only when the HLR has ascertained that the VLR supports CAMEL Phase 2 or higher. Thus, the ISD message containing the Forwarded-to number which is not in E.164 international format shall be sent to the VLR only after the HLR receives confirmation in the first ISD message result that CAMEL Phase 2 or higher is supported.

A Forwarded-to number non-international E.164 format shall only be sent from an HLR to a VLR if the VLR supports CAMEL Phase 2, or a subsequent phase of CAMEL.

When the SGSN receives regional subscription data (Zone Code List) it may respond with "SGSN Area Restricted" in the MAP_INSERT_SUBSCRIBER_DATA response. In this case the "SGSN Area Restricted Flag" shall be set to "restricted" in the HLR. This will prevent MT SM from being forwarded to the SGSN and Network Requested PDP-Context Activation.

If the HLR neither stores "Roaming Restricted In SGSN Due To Unsupported Feature" nor receives "SGSN Area Restricted" in the MAP_INSERT_SUBSCRIBER_DATA response, the "SGSN Area Restricted Flag" in the HLR shall be set to "not restricted".

The SDL diagrams are shown in figures 19.4/1 and 19.4/2.

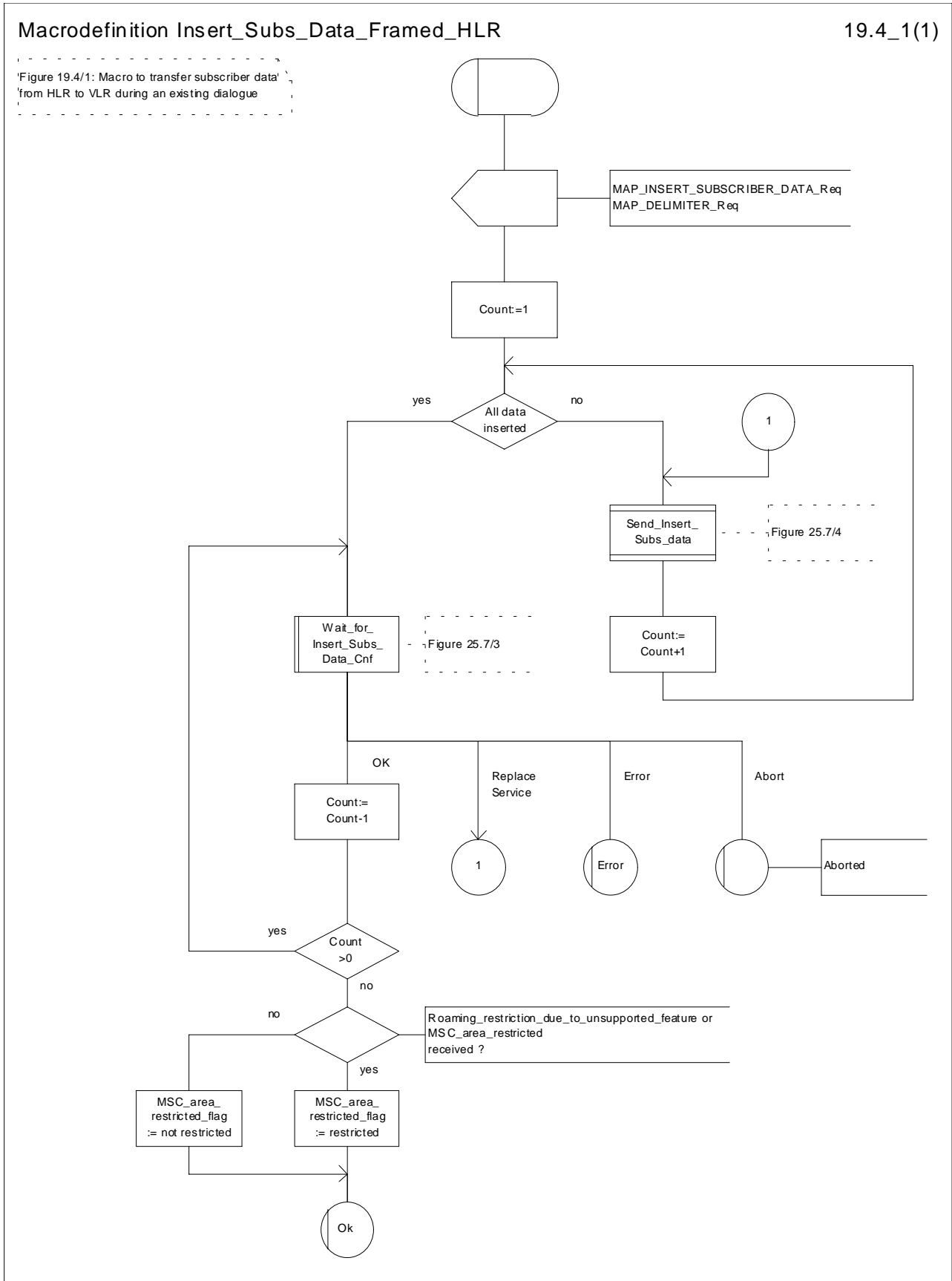


Figure 19.4/1: Macro Insert_Subs_Data_Framed_HLR

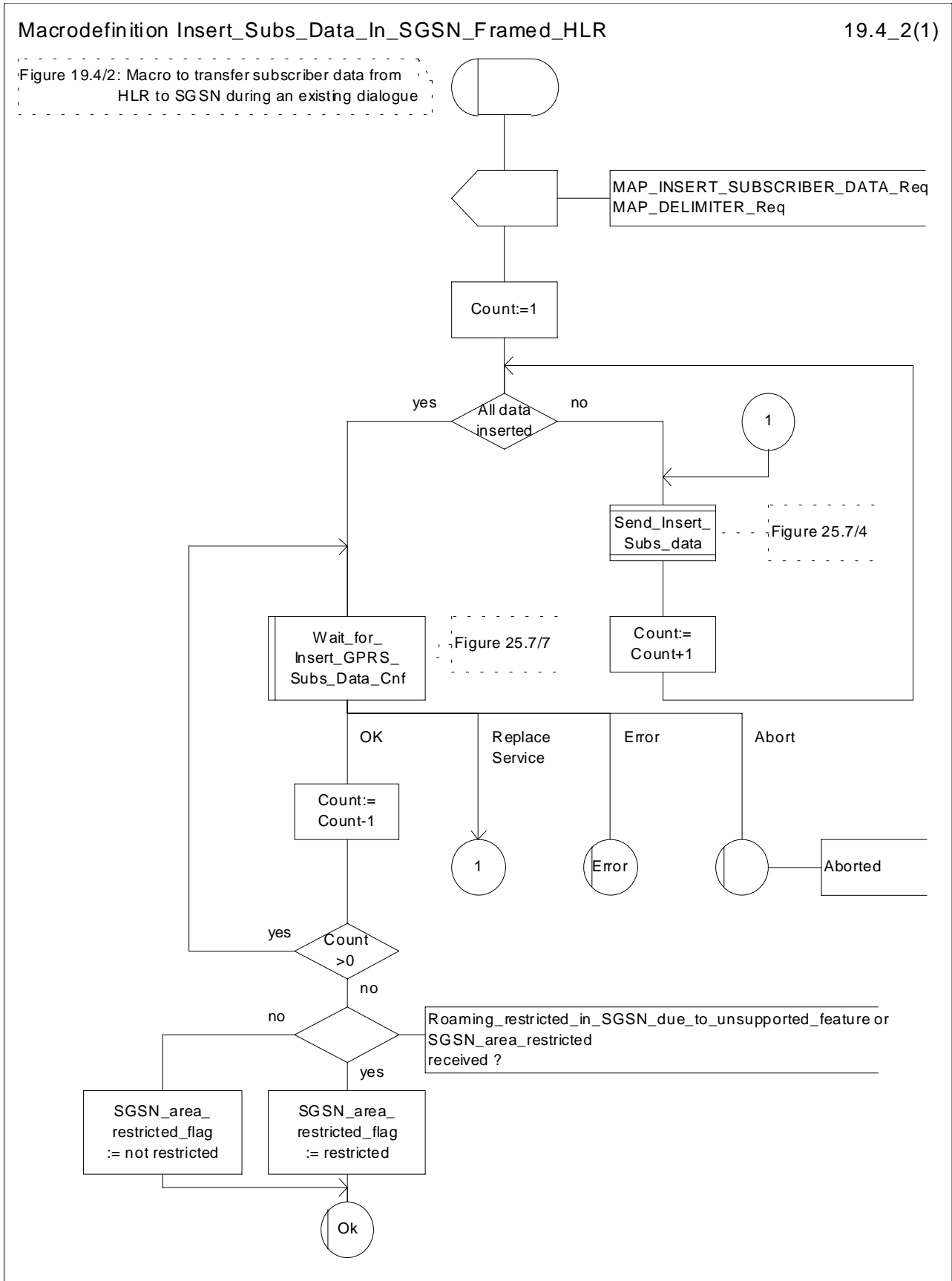


Figure 19.4/2: Macro Insert_Subs_Data_In_SGSN_Framed_HLR

19.5 Mobility Management Event notification procedure

19.5.1 General

The Mobility Management Event Notification VLR process (MMEN_VLR) is used to notify a gsmSCF about the successful completion of a Mobility Management event.

Figure 19.5/1. depicts the MAP signalling used for the event notification.

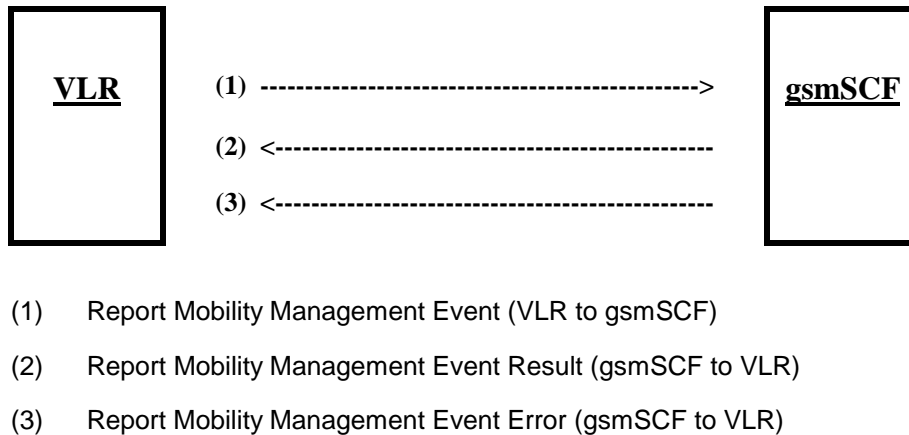


Figure 19.5/1: Interfaces and MAP Messages for Mobility Management Event notifications

19.5.2 Process in the VLR

The Mobility Management event notification procedure in the VLR is triggered when the following conditions are fulfilled:

1. The VLR has successfully completed a Mobility Management event
2. The subscriber has a subscription to Mobility Management event notifications
3. The Mobility Management event is marked for reporting

The VLR notifies the gsmSCF of a mobility management event with the ReportMMEvent MAP message. This message is sent in a TCAP TC-BEGIN primitive. The VLR then awaits a positive result (RESULT) or a negative result (ERROR). This is received in a TCAP TC-END primitive. The Basic End procedure is used.

When the VLR has received the RESULT or ERROR, the relationship between the VLR and the gsmSCF is terminated. The relationship, if existing, is also terminated when the VLR sends a TCAP P-ABORT primitive to the calling procedure or when the VLR receives a TCAP P-ABORT or a TCAP-U-ABORT primitive from the gsmSCF.

The sending process shall indicate to the MMEN_VLR process, which Mobility Management event shall be reported to the gsmSCF.

The MMEN_VLR process is illustrated in Figure 19.5/2.

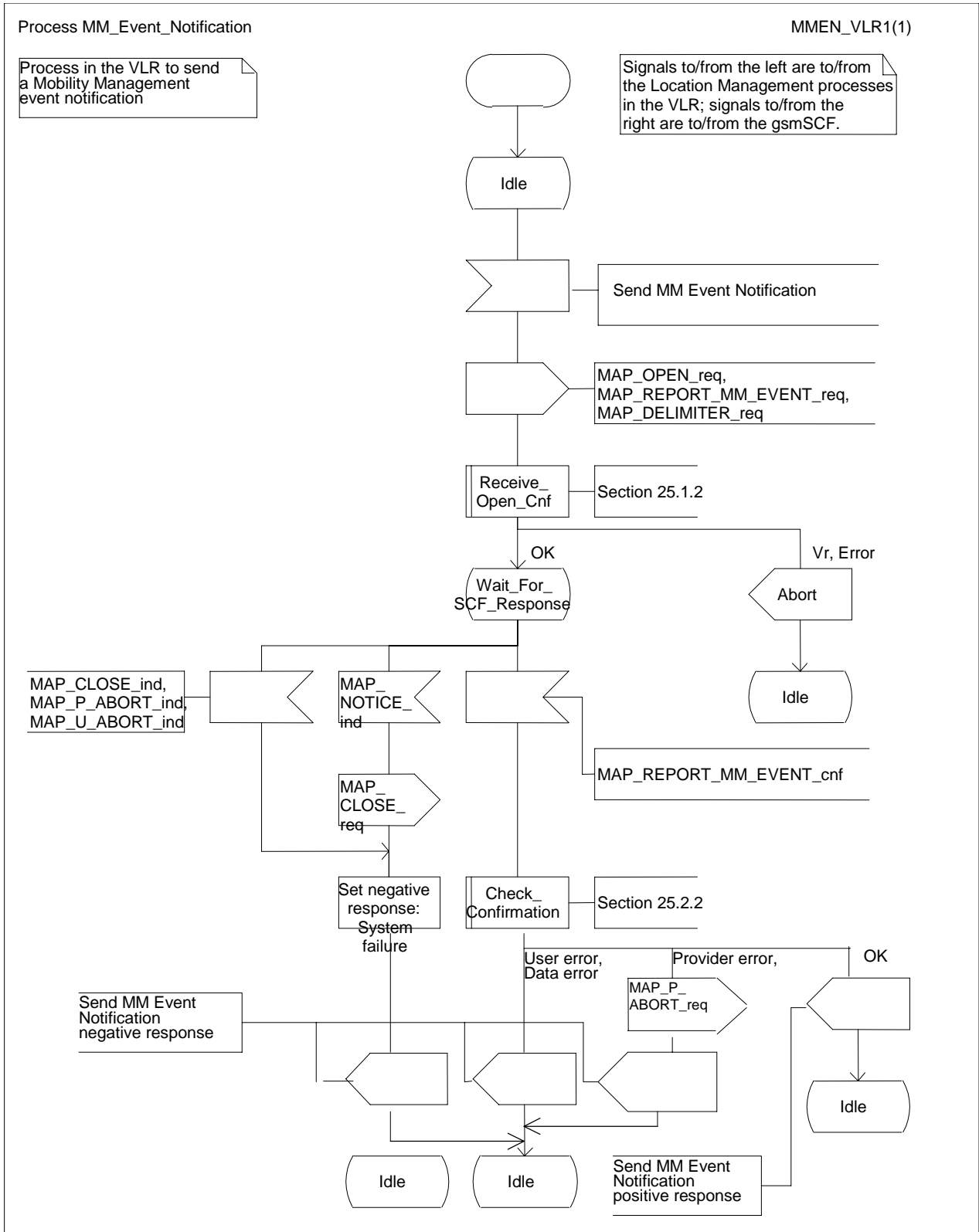


Figure 19.5/2: Process MM_Event_Notification_VLR (sheet 1 of 1)

19.5.3 Process in the gsmSCF

When the gsmSCF receives the ReportMMEEvent MAP Message (in a TCAP TC-BEGIN primitive), the MM_Event_Notification_gsmSCF' (MMEN_SCF) process is started.

If the gsmSCF has validated the information it has received in the ReportMMEEvent MAP Message, then it informs the Service Logic in the SCP and awaits a response.

If a positive response is received from the Service Logic, then a REPORT_MM_EVENT_RESULT is sent to the VLR. If a negative response is received from the Service Logic, then a REPORT_MM_EVENT_ERROR is sent to the VLR. Both RESULT and ERROR are sent in a TCAP TC-END primitive. The Basic End procedure is used.

If the Service Logic returns a User Error, then a MAP U-ABORT primitive is sent to the VLR.

The gsmSCF TCAP service may choose to abort the relationship with the VLR by sending a TCAP P-ABORT primitive to the VLR.

When the gsmSCF receives a TCAP P-ABORT primitive from the VLR, it shall immediately terminate the mobility management process.

This is illustrated in figure 19.5/3.

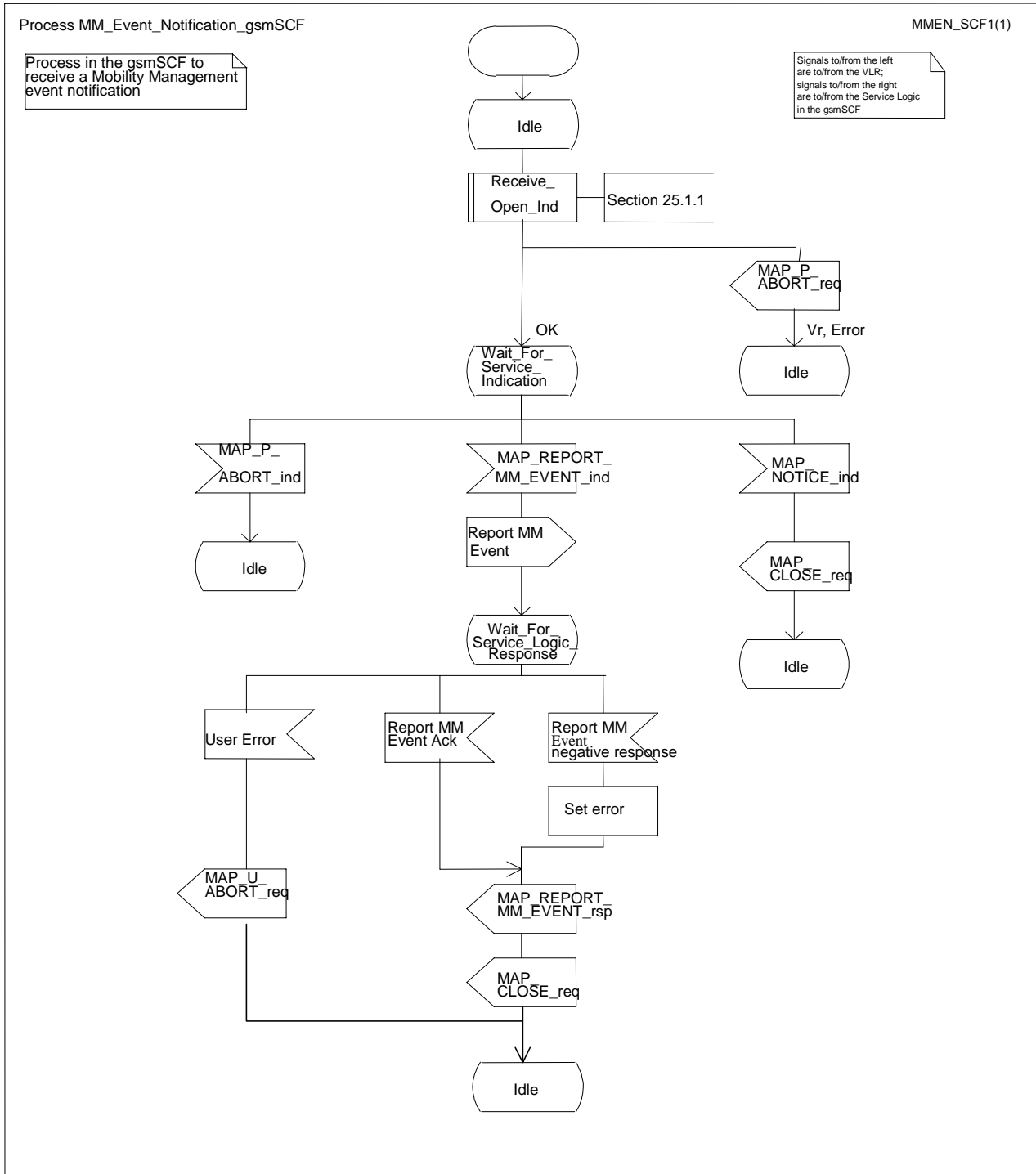


Figure 19.5/3 Process MM_Event_Notification_gsmSCF (sheet 1 of 1)

20 Operation and maintenance procedures

20.1 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 or in the SGSN as described in the following subclauses.

20.1.1 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 first 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 20.1/1.

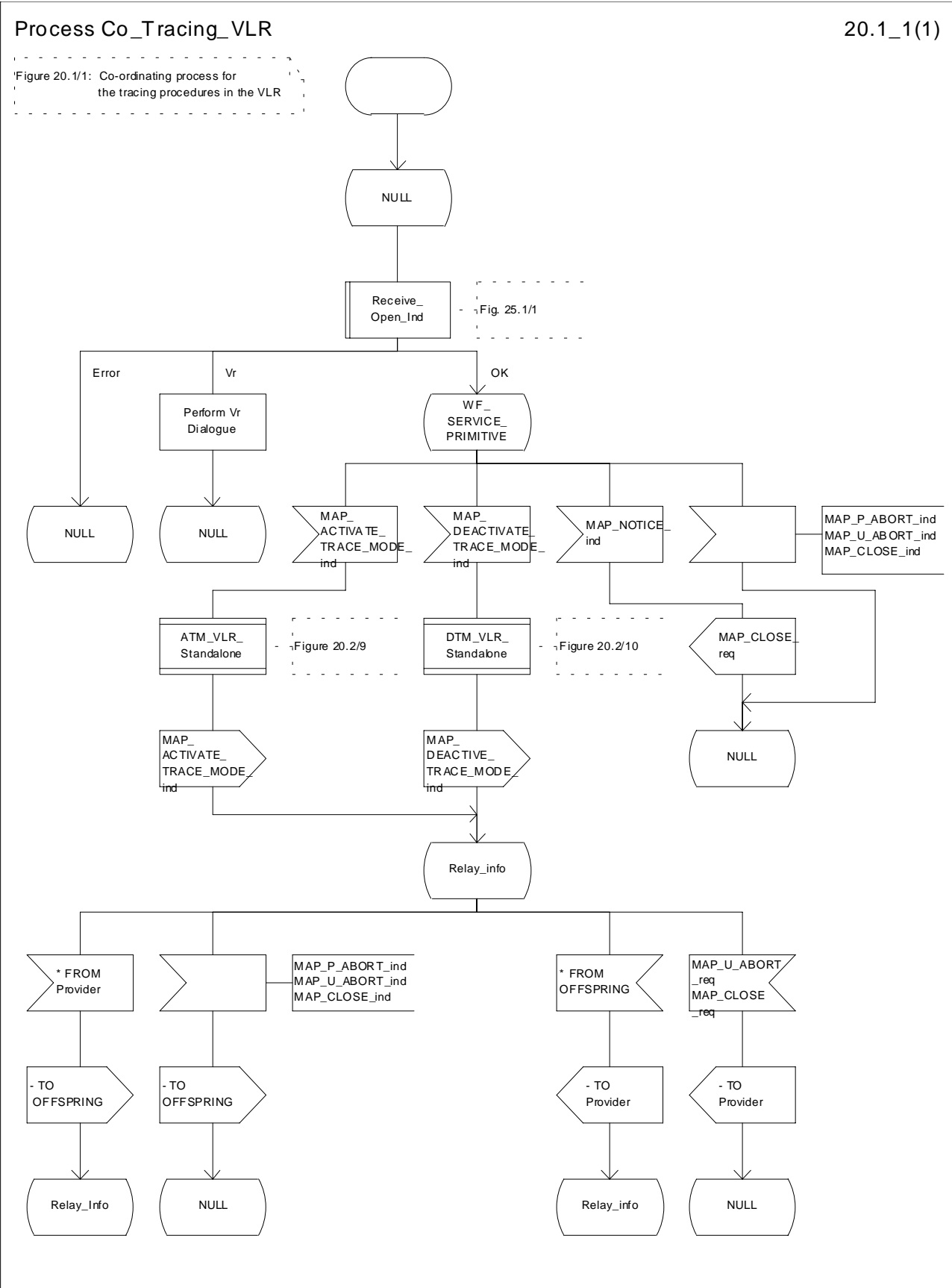


Figure 20.1/1: Process Co_Tracing_VLR

20.1.2 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 first 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 20.1/2.

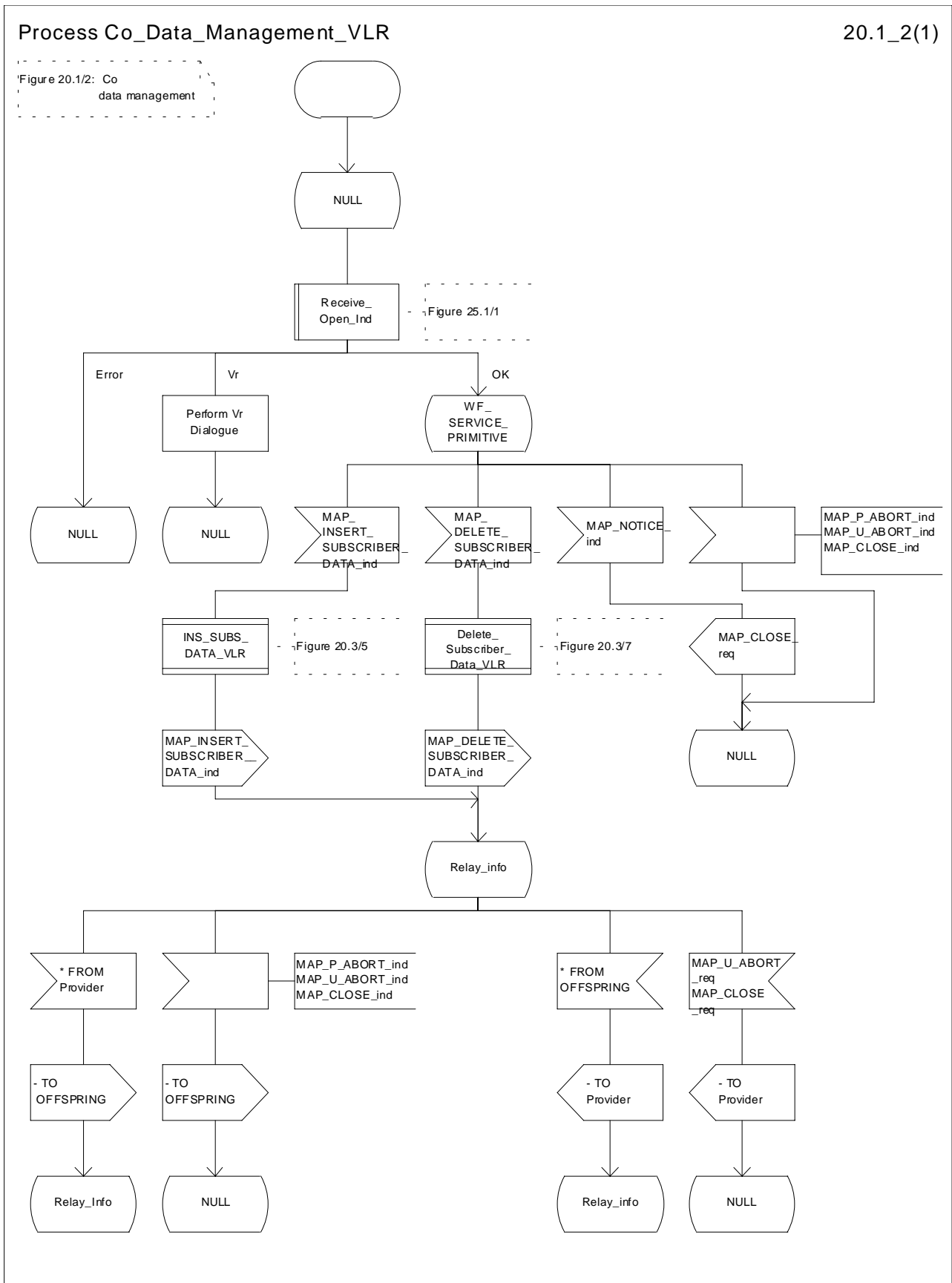


Figure 20.1/2: Process Co_Data_Management_VLR

20.1.3 Tracing Co-ordinator for the SGSN

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 first 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_SGSN_Standalone is created;
- if the MAP_DEACTIVATE_TRACE_MODE indication is received, the process DTM_SGSN_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 for the SGSN is shown in the figure 20.1/3.

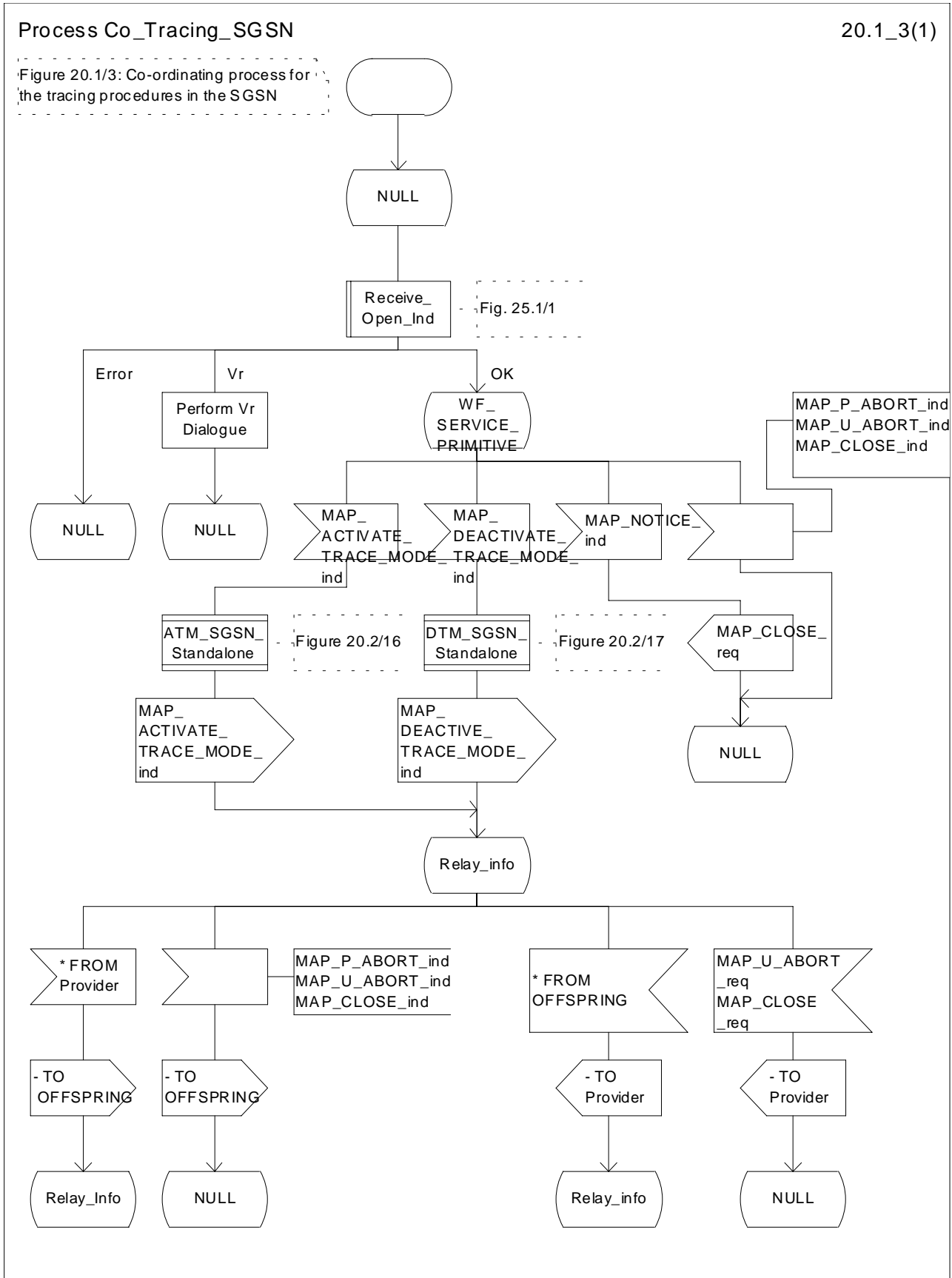


Figure 20.1/3: Process Co_Tracing_SGSN

20.1.4 Subscriber Data Management Co-ordinator for the SGSN

The MAP_OPEN indication opens the dialogue for the stand-alone subscriber data management procedure when the application context subscriberDataManagementContext 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_INSERT_SUBSCRIBER_DATA indication is received, the process INS_SUBS_DATA_SGSN is created;
- if the MAP_DELETE_SUBSCRIBER_DATA indication is received, the process Delete_Subscriber_Data_SGSN 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 20.1/4.

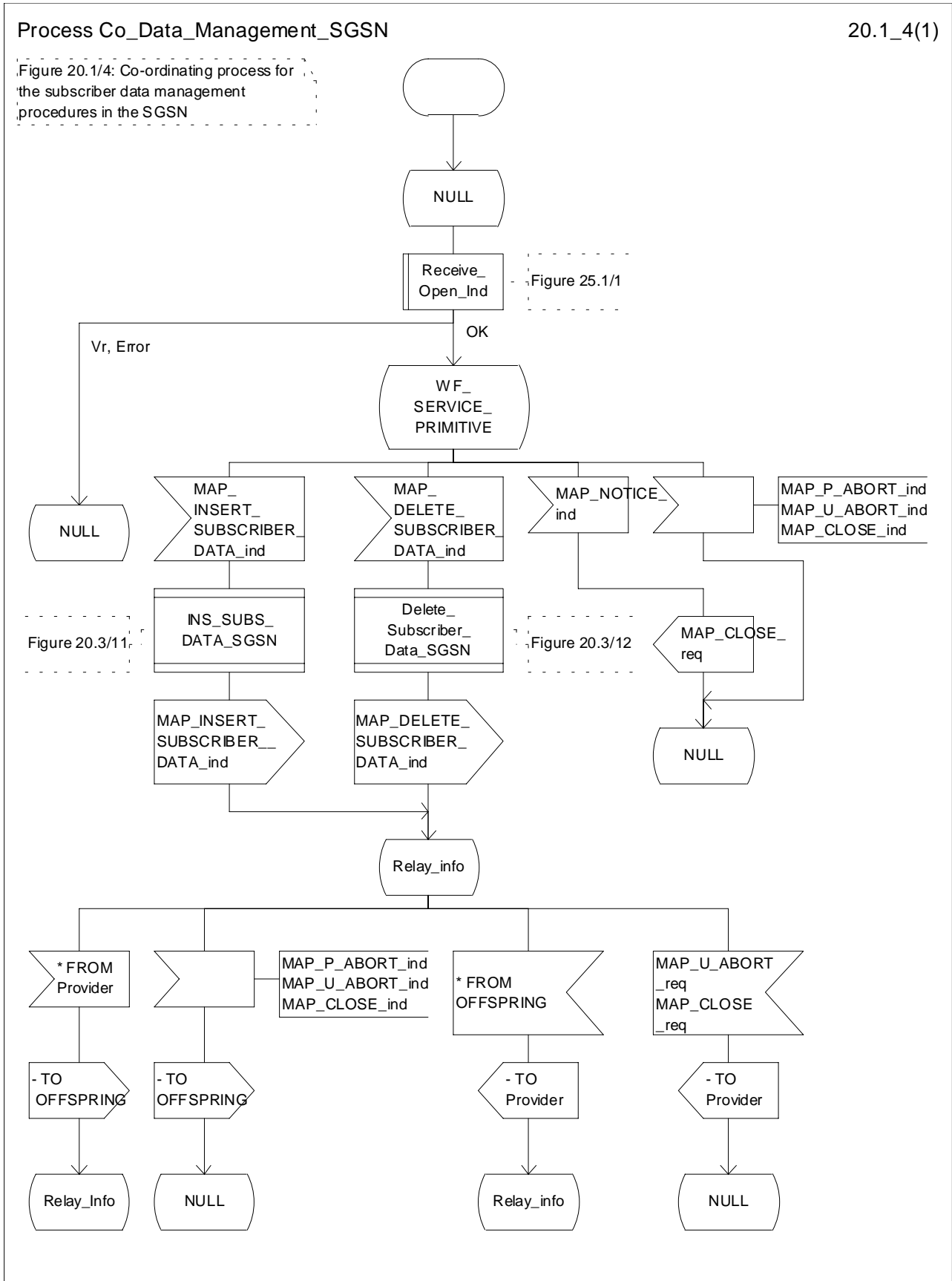


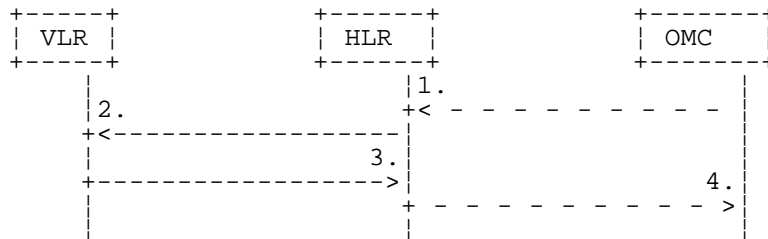
Figure 20.1/4: Process Co_Data_Management_SGSN

20.2 Tracing procedures

Three type of tracing procedures exist:

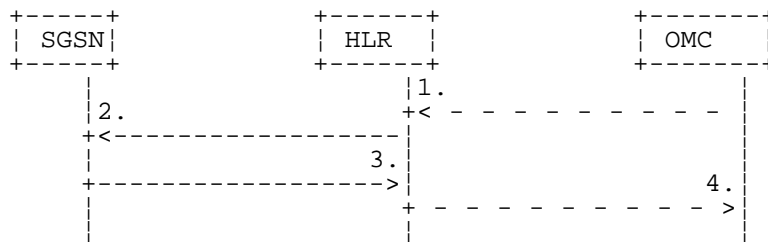
- i) Subscriber tracing management procedures;
- ii) Subscriber tracing procedures;
- iii) Event tracing procedures.

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 20.2/1 and 20.2/2. The procedures for providing a trace request to the SGSN are shown in figures 20.2/11 and 20.2/12.



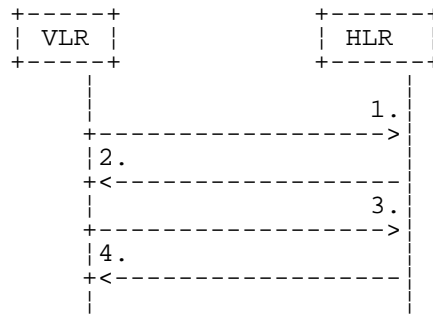
- 1) Subscriber Tracing Activation
- 2) MAP_ACTIVATE_TRACE_MODE
- 3) MAP_ACTIVATE_TRACE_MODE_ACK
- 4) Subscriber Tracing Activation Accepted

Figure 20.2/1: Stand alone subscriber tracing activation procedure



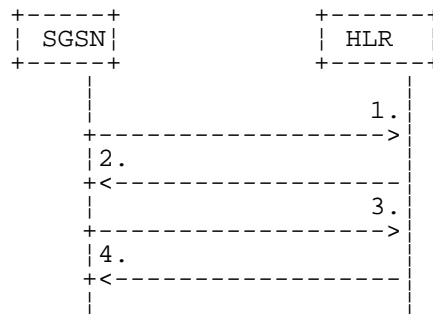
- 1) Subscriber Tracing Activation
- 2) MAP_ACTIVATE_TRACE_MODE
- 3) MAP_ACTIVATE_TRACE_MODE_ACK
- 4) Subscriber Tracing Activation Accepted

Figure 20.2/11: Stand alone subscriber tracing activation procedure for GPRS



- 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 20.2/2: Subscriber tracing activation procedure at location updating or data restoration

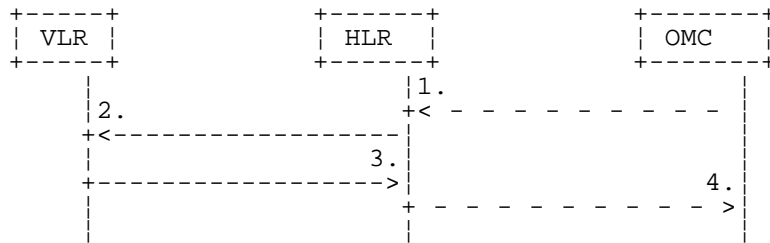


- 1) MAP_UPDATE_GPRS_LOCATION
- 2) MAP_ACTIVATE_TRACE_MODE
- 3) MAP_ACTIVATE_TRACE_MODE_ACK
- 4) MAP_UPDATE_GPRS_LOCATION_ACK

Figure 20.2/12: Subscriber tracing activation procedure at GPRS location updating

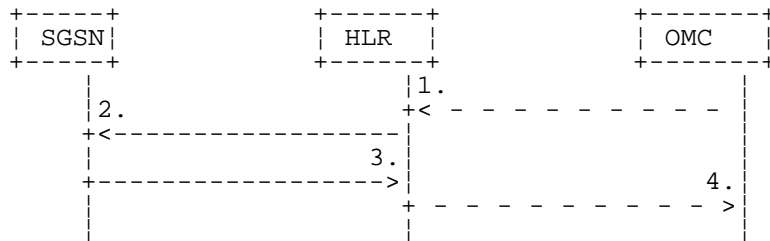
The HLR sends the trace request (IMSI, trace reference, trace type and identity of the OMC) to the VLR or to the SGSN in a MAP_ACTIVATE_TRACE_MODE request. The receipt of this primitive is acknowledged. The acknowledge primitive will indicate that the trace request is accepted by the VLR or by the SGSN. 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 or in the SGSN. The procedures is shown in figures 20.2/3 and 20.2/13.



- 1) Subscriber Tracing Deactivation
- 2) MAP_DEACTIVATE_TRACE_MODE
- 3) MAP_DEACTIVATE_TRACE_MODE_ACK
- 4) Subscriber Tracing Deactivation Accepted

Figure 20.2/3: Subscriber tracing deactivation procedure

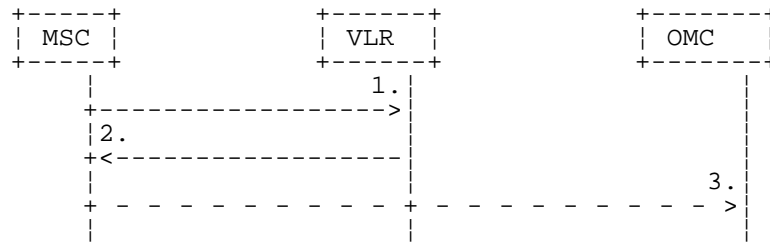


- 1) Subscriber Tracing Deactivation
- 2) MAP_DEACTIVATE_TRACE_MODE
- 3) MAP_DEACTIVATE_TRACE_MODE_ACK
- 4) Subscriber Tracing Deactivation Accepted

Figure 20.2/13: Subscriber tracing deactivation procedure for GPRS

The HLR sends a MAP_DEACTIVATE_TRACE_MODE request to the VLR or to the SGSN. The VLR or the SGSN will acknowledge the deactivation. The acknowledge primitive will indicate that the trace request has been deleted by the VLR or by the SGSN. 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 20.2/4.



- 1) MAP_PROCESS_ACCESS_REQUEST, MAP_UPDATE_LOCATION_AREA,
- 2) MAP_TRACE_SUBSCRIBER_ACTIVITY
- 3) Subscriber tracing information

Figure 20.2/4: 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 20.2/5 and 20.2/6 are spare.]

20.2.1 Procedures in the HLR

20.2.1.1 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 or to the SGSN 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 or the SGSN.

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 or the SGSN 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 25.9/4) shall be initiated by the location updating process in the HLR.

The subscriber tracing activation process in the HLR with VLR is shown in figure 20.2/7.

The subscriber tracing activation process in the HLR with SGSN is shown in figure 20.2/14.

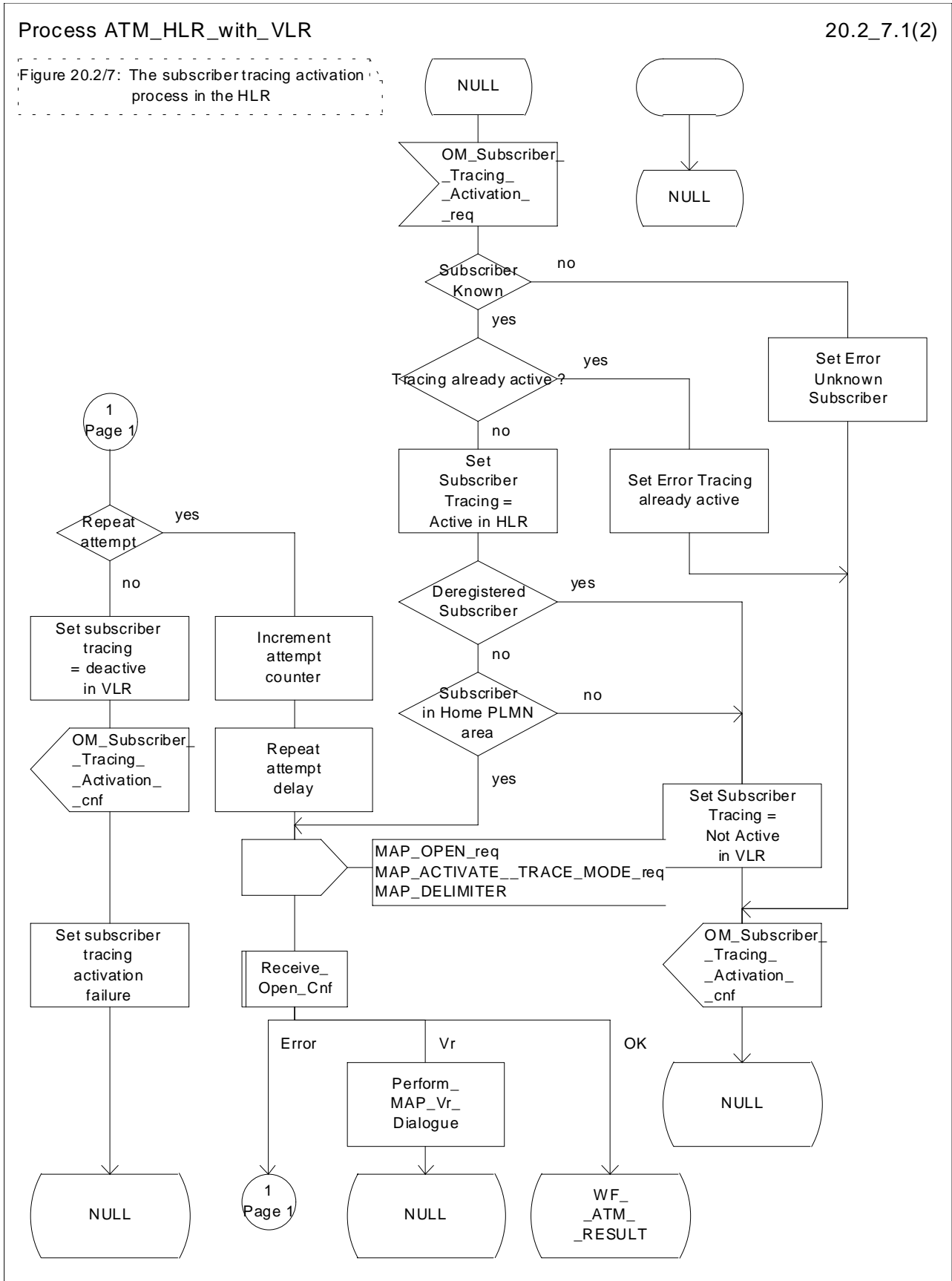


Figure 20.2/7 (sheet 1 of 2): Process ATM_HLR_with_VLR

Process ATM_HLR_with_VLR

20.2_7.2(2)

Figure 20.2/7: The subscriber tracing activation process in the HLR

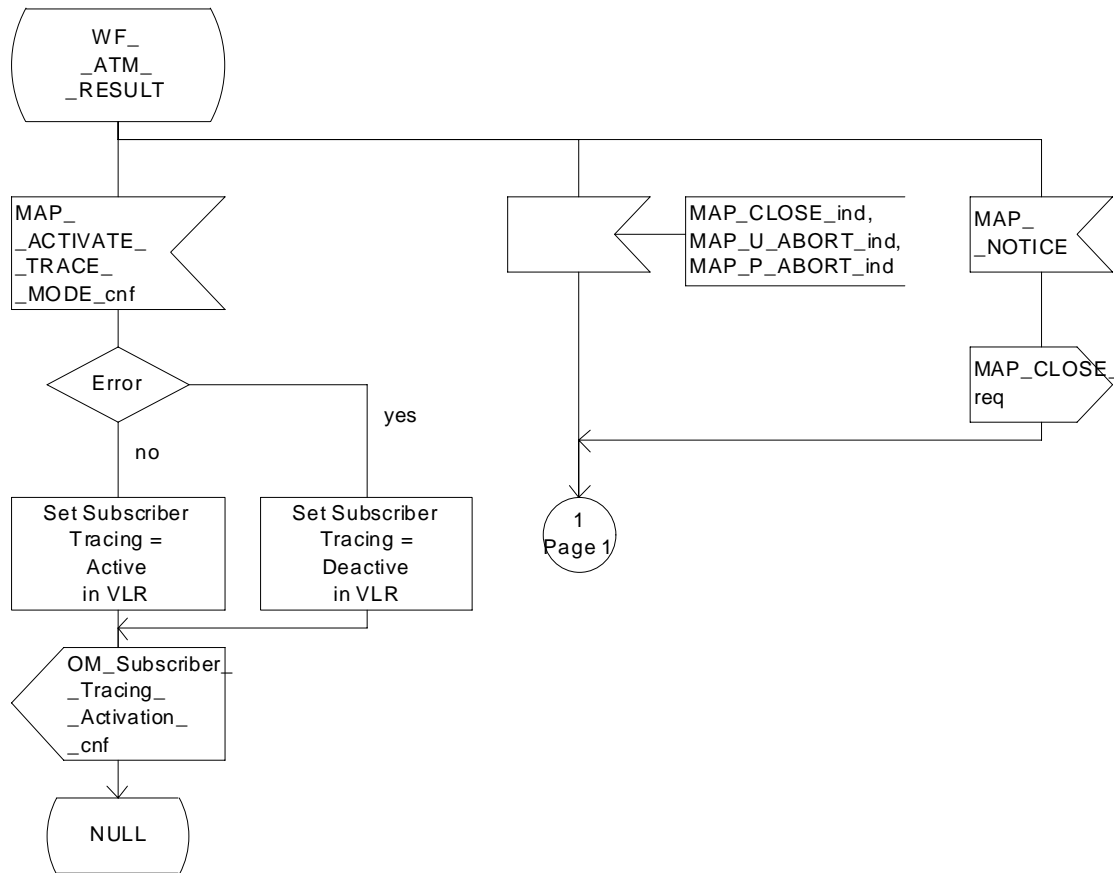


Figure 20.2/7 (sheet 2 of 2): Process ATM_HLR_with_VLR

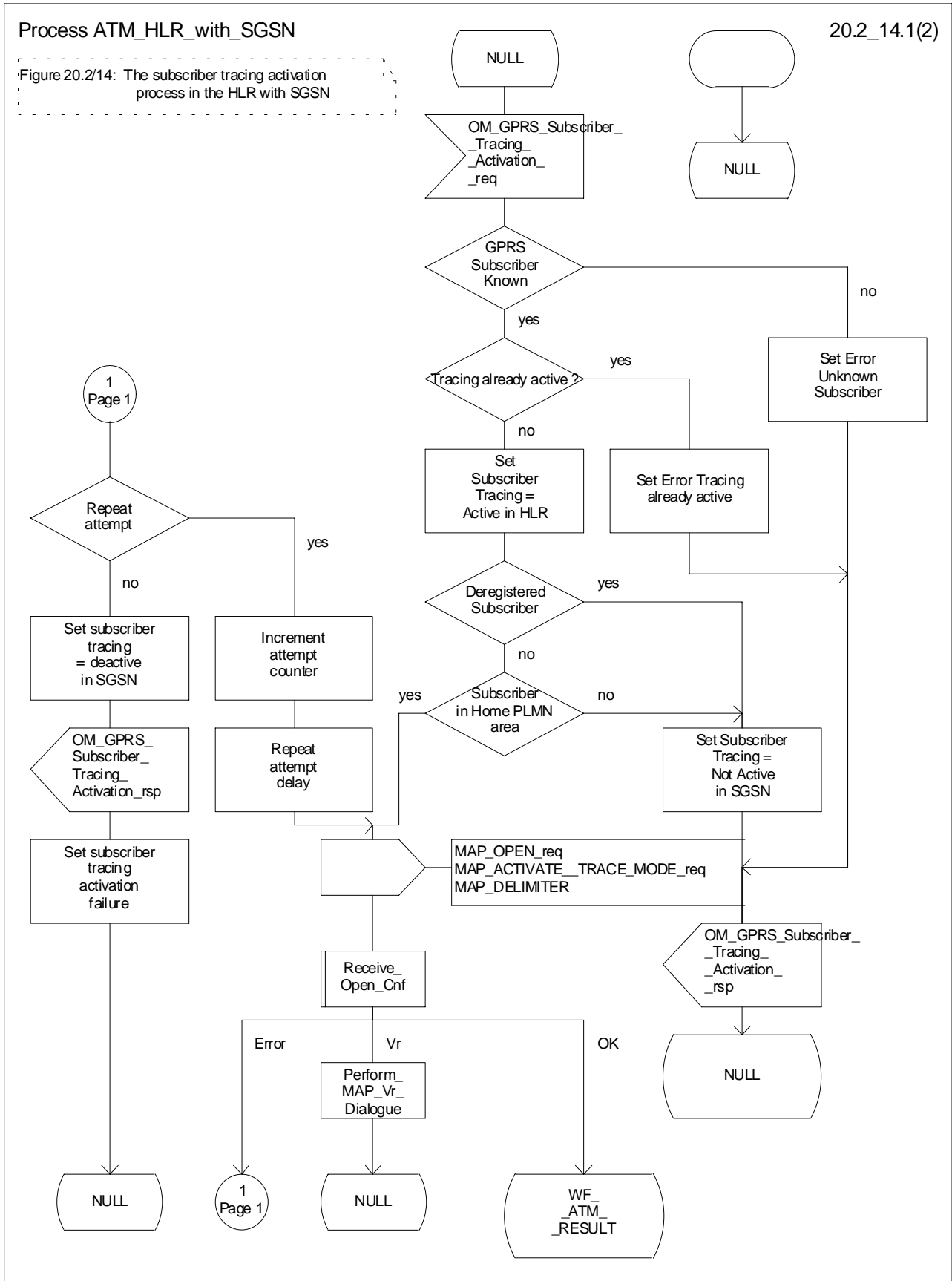


Figure 20.2/14 (sheet 1 of 2): Process ATM_HLR_with_SGSN

Process ATM_HLR_with_SGSN

20.2_14.2(2)

Figure 20.2/14: The subscriber tracing activation process in the HLR with SGSN

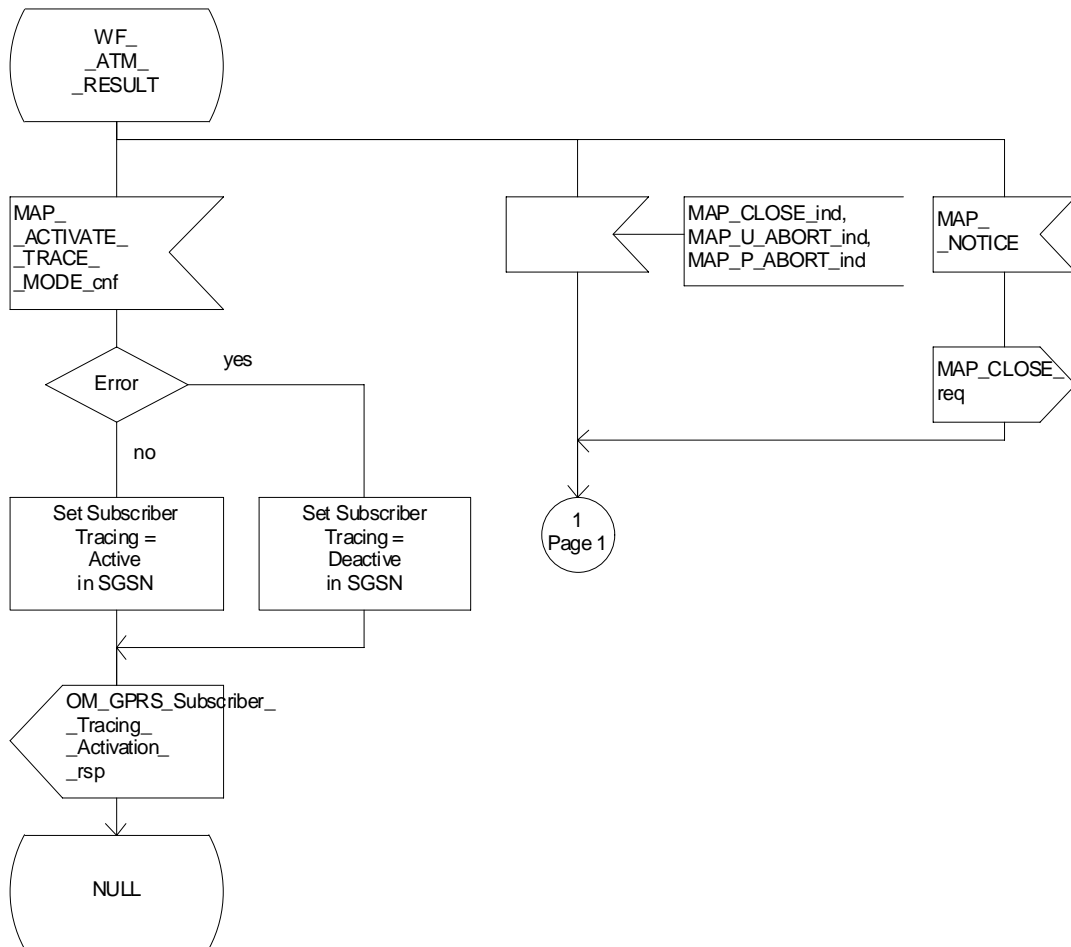


Figure 20.2/14 (sheet 2 of 2): Process ATM_HLR_with_SGSN

20.2.1.2 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 or to the SGSN 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 or in SGSN 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 or by the SGSN.

The subscriber tracing deactivation procedure with VLR is shown in figure 20.2/8.

The subscriber tracing deactivation procedure with SGSN is shown in figure 20.2/15.

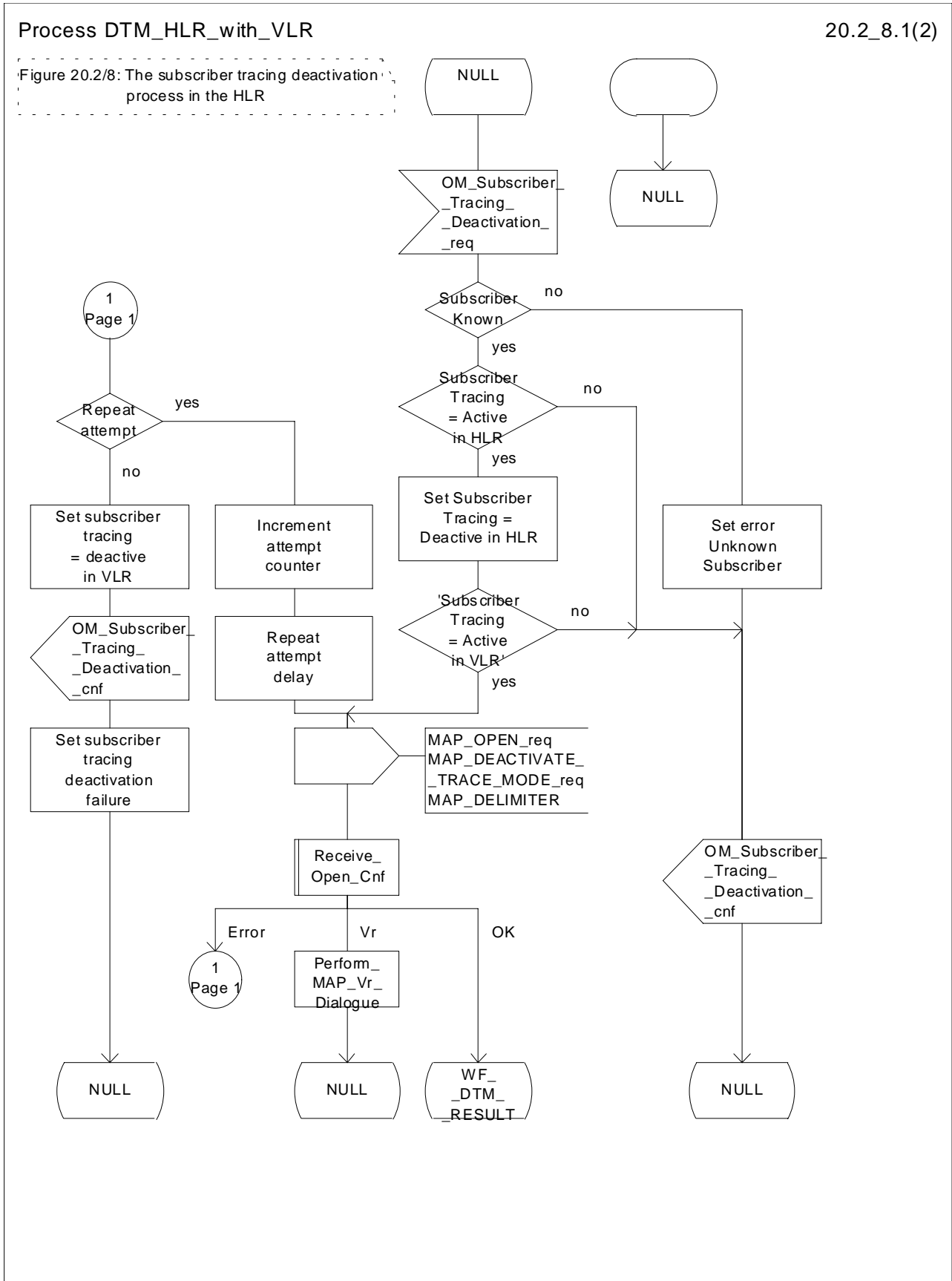


Figure 20.2/8 (sheet 1 of 2): Process DTM_HLR_with_VLR

Process DTM_HLR_with_VLR

20.2_8.2(2)

Figure 20.2/8: The subscriber tracing deactivation process in the HLR

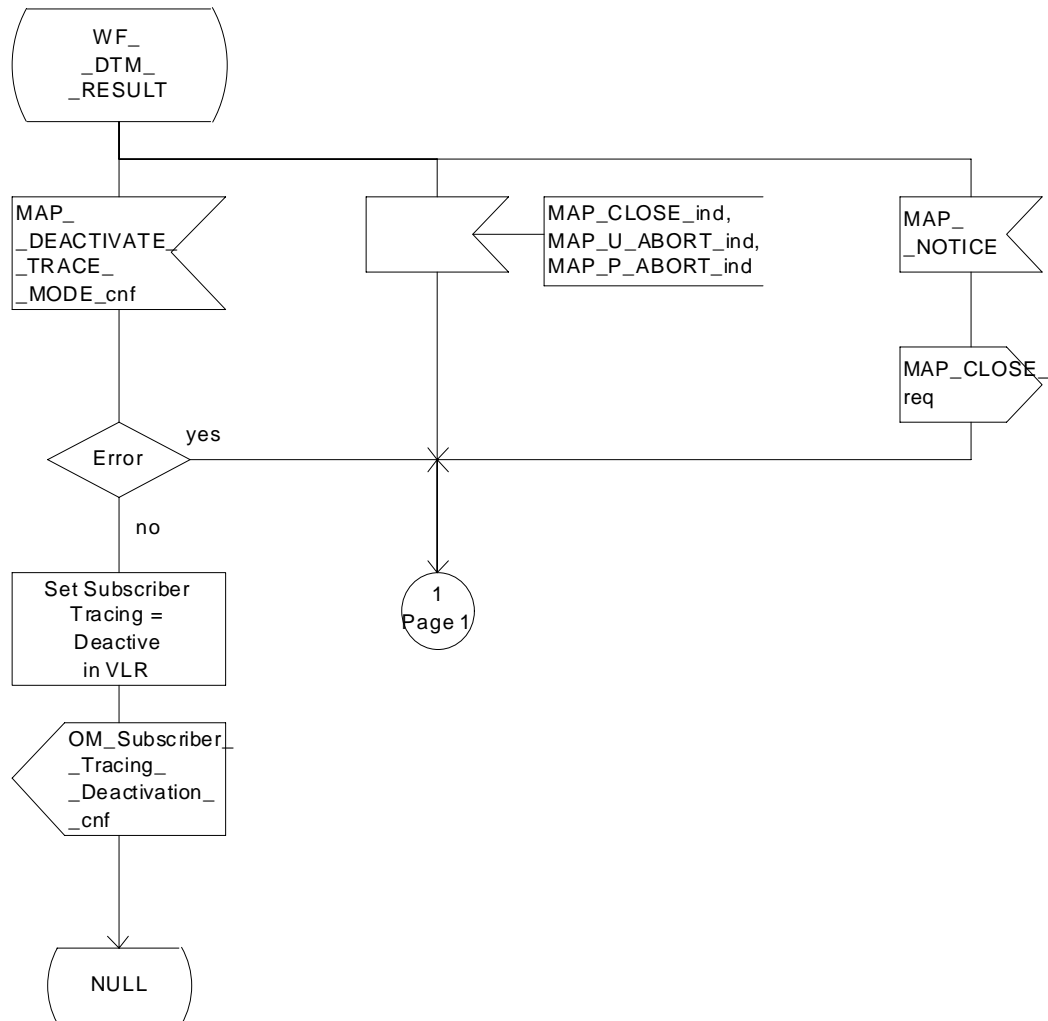


Figure 20.2/8 (sheet 2 of 2): Process DTM_HLR_with_VLR

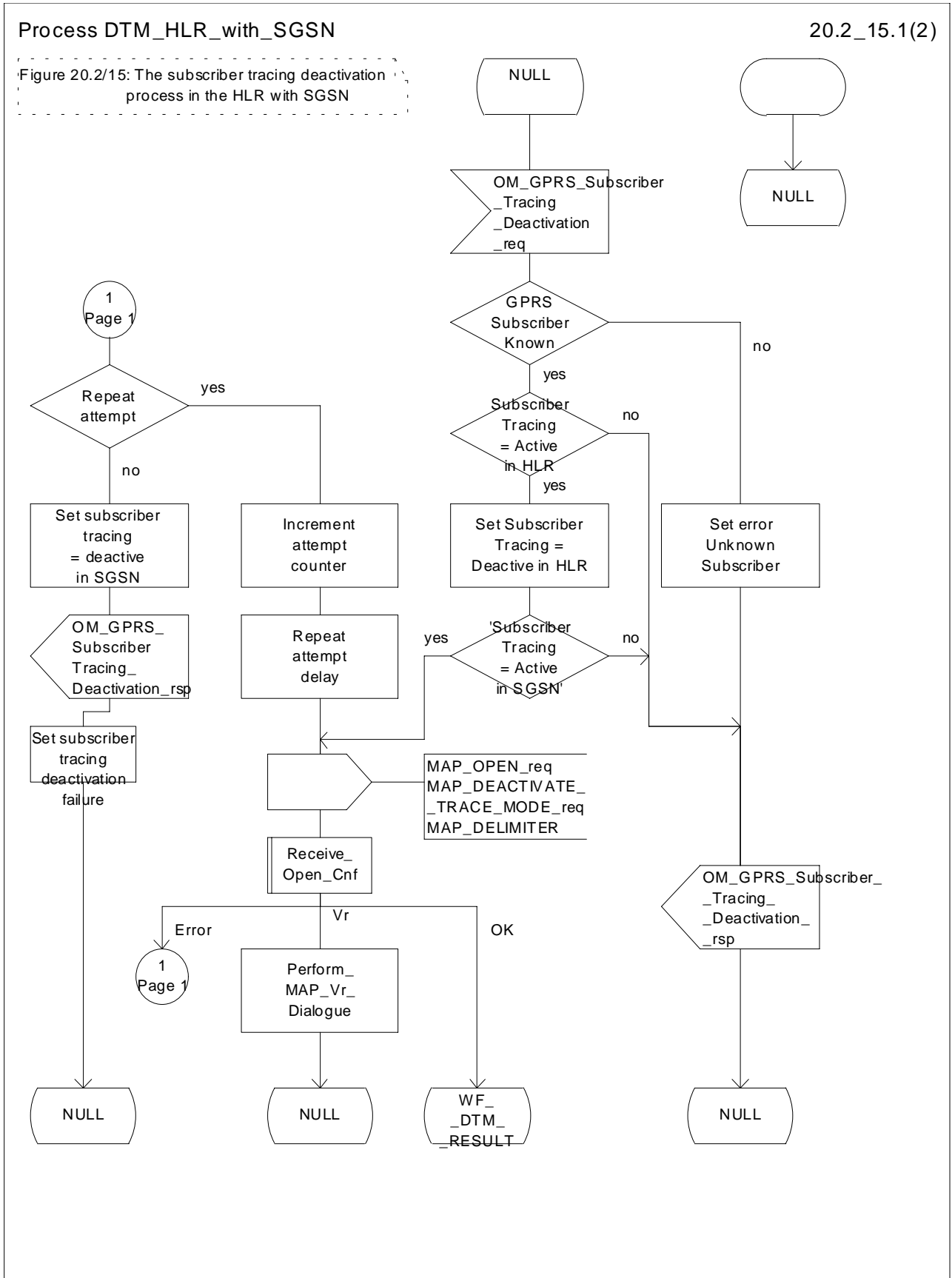


Figure 20.2/15 (sheet 1 of 2): Process DTM_HLR_with_SGSN

Process DTM_HLR_with_SGSN

20.2_15.2(2)

Figure 20.2/15: The subscriber tracing deactivation process in the HLR with SGSN

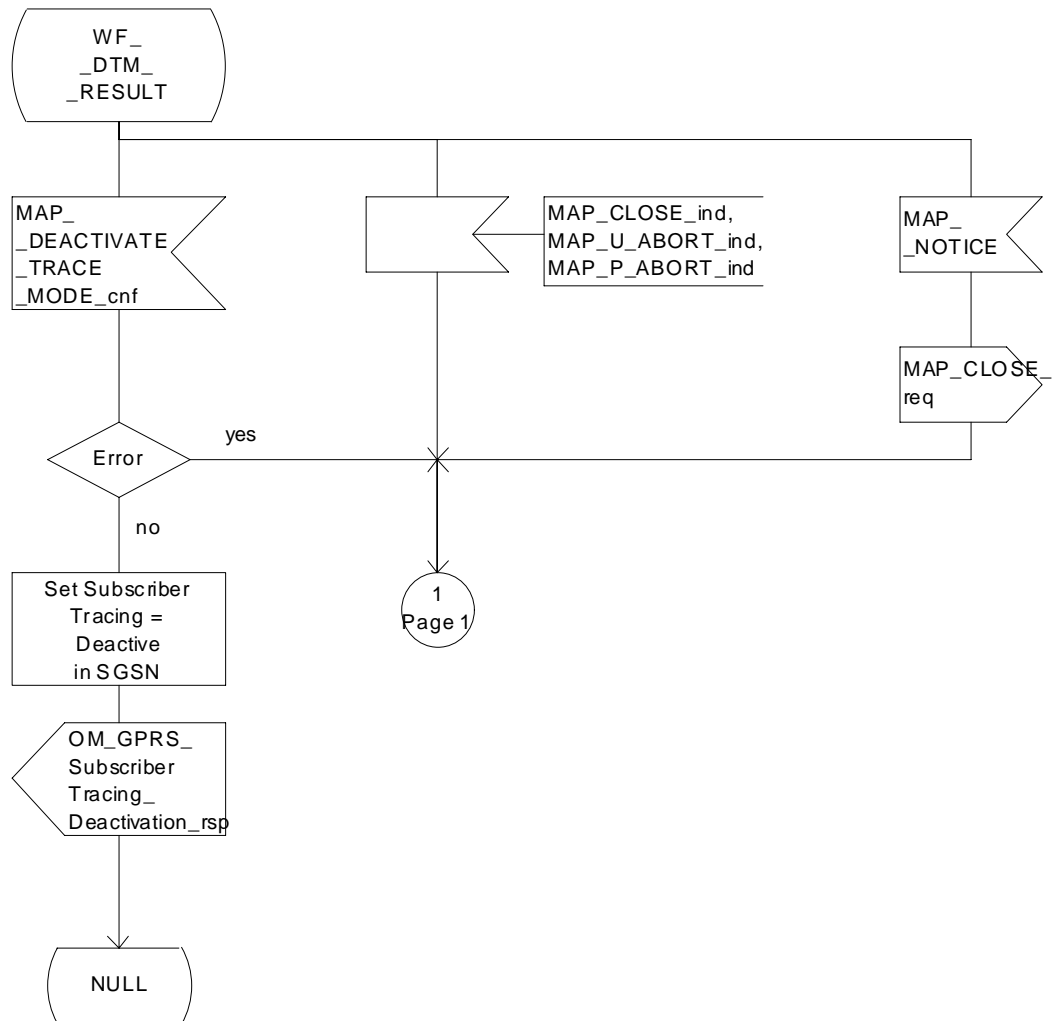


Figure 20.2/15 (sheet 2 of 2): Process DTM_HLR_with_SGSN

20.2.2 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.

20.2.2.1 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 25.9/3).

The subscriber tracing activation process in the VLR is shown in figure 20.2/9.

Process ATM_VLR_Standalone

20.2_9(1)

FIGURE 20.2/9 The subscriber tracing activation process for standalone operation in the VLR

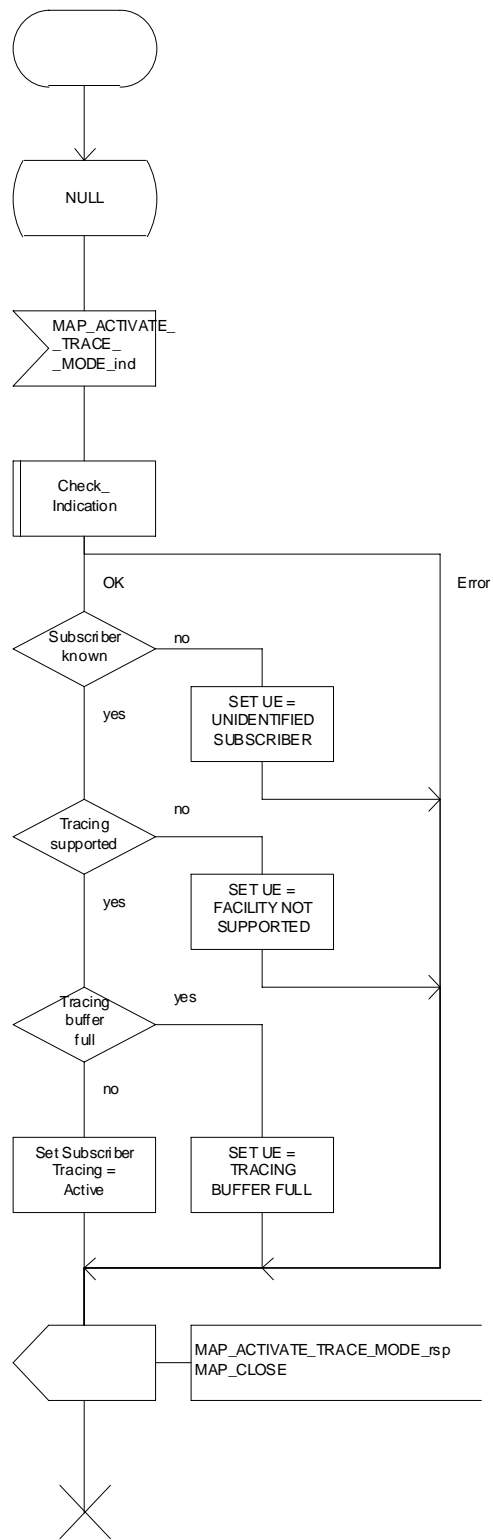


Figure 20.2/9: Process ATM_VLR_Standalone

20.2.2.2 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 20.2/10.

Process DTM_VLR_Standalone

20.2_10(1)

Figure 20.2/10: The subscriber tracing deactivation process in the VLR

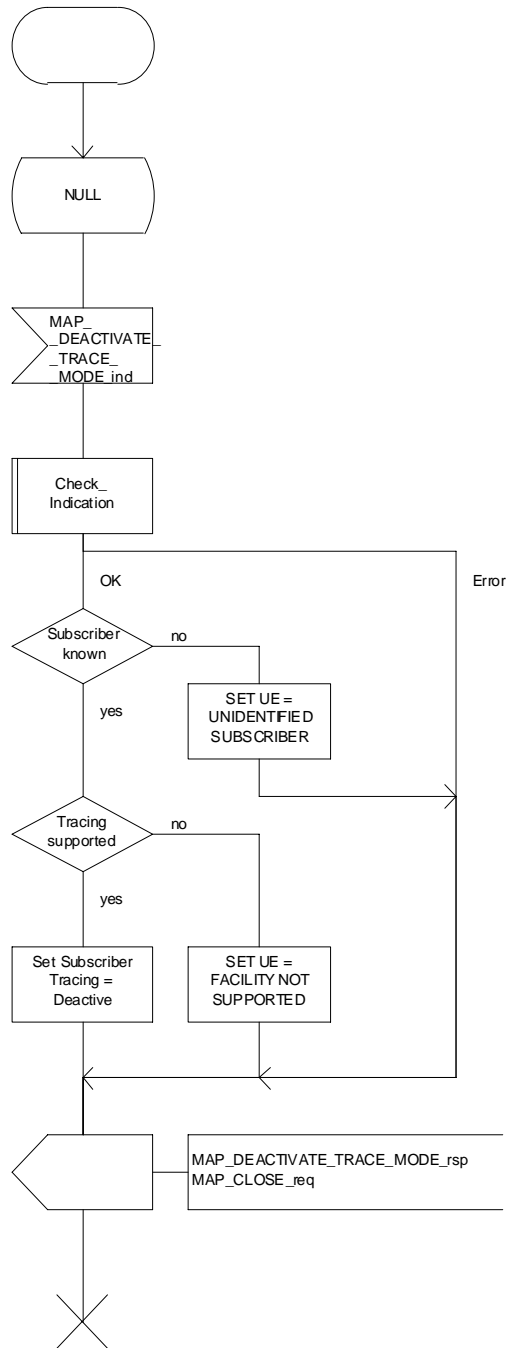


Figure 20.2/10: Process DTM_VLR_Standalone

20.2.2.3 Subscriber tracing procedure

When the VLR receives a MAP_PROCESS_ACCESS_REQUEST or MAP_UPDATE_LOCATION_AREA 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 25.9/2.

20.2.3 Procedures in the MSC

The MSC is involved in the following tracing procedure:

- i) Subscriber tracing procedure.

20.2.3.1 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 25.9/1.

20.2.4 Procedures in the SGSN

The SGSN is involved in the following tracing procedures:

- i) Subscriber tracing activation procedure;
- ii) Subscriber tracing deactivation procedure;

20.2.4.1 Subscriber tracing activation procedure

When receiving a MAP_ACTIVATE_TRACE_MODE indication, the SGSN 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_SGSN (see figure 25.9/7).

The subscriber tracing activation process in the SGSN is shown in figure 20.2/16.

20.2.4.2 Subscriber tracing deactivation procedure in SGSN

When receiving a MAP_DEACTIVATE_TRACE_MODE indication, the SGSN 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 SGSN is shown in figure 20.2/17.

Process ATM_SGSN_Standalone

20.2_16(1)

FIGURE 20.2/16: The subscriber tracing activation process for standalone operation in the SGSN

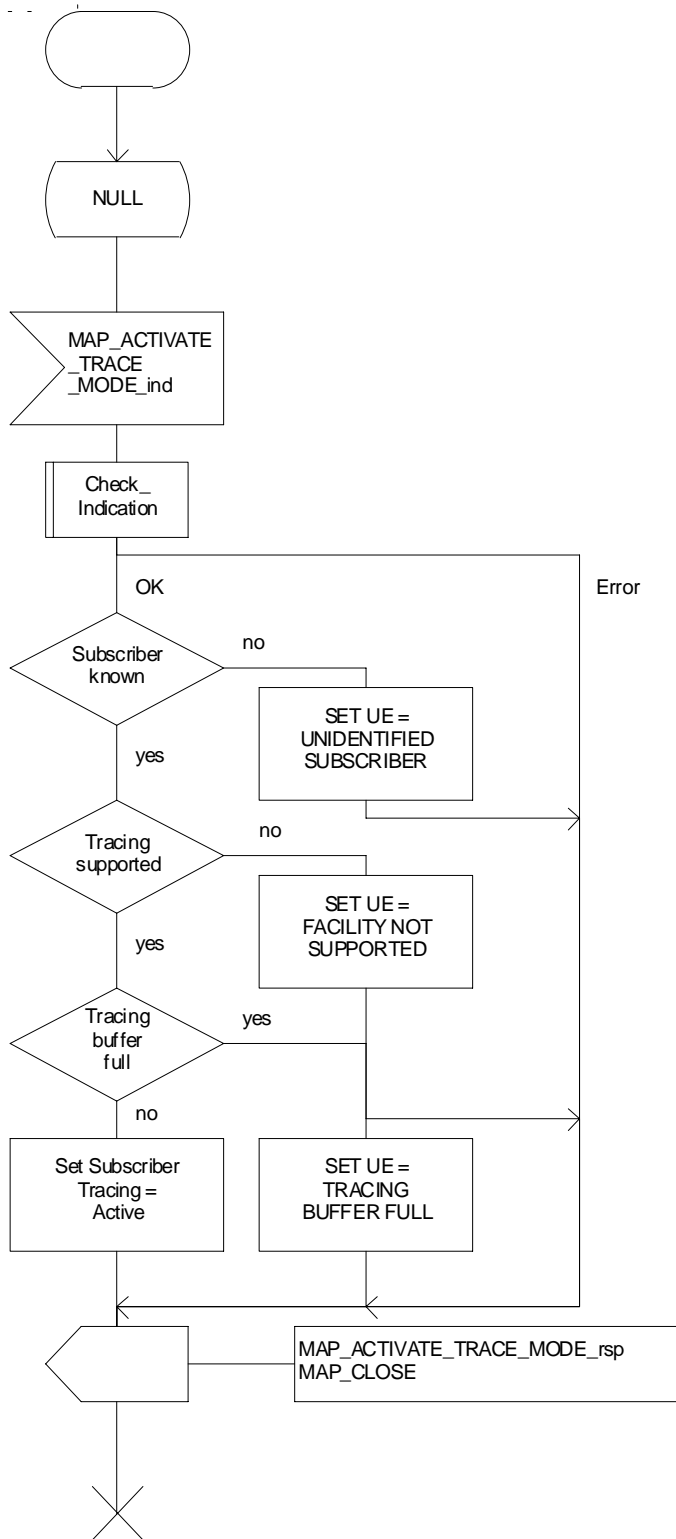


Figure 20.2/16: Process ATM_SGSN_Standalone

Process DTM_SGSN_Standalone

20.2_17(1)

Figure 20.2/17: The subscriber tracing deactivation process in the SGSN

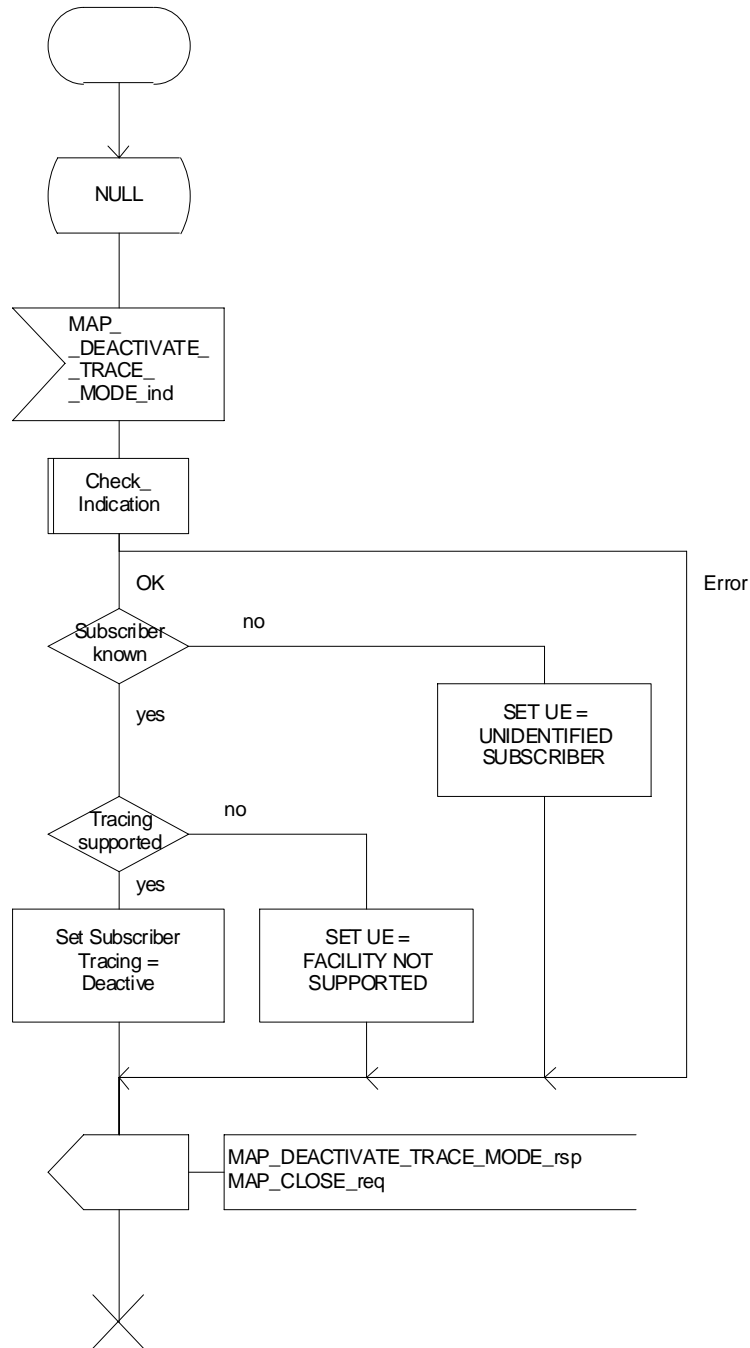


Figure 20.2/17: Process DTM_SGSN_Standalone

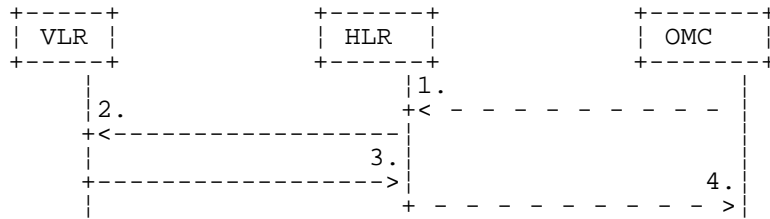
20.3 Subscriber data management procedures

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.

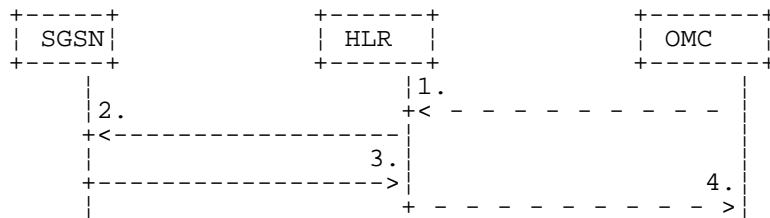
The subscriber deletion and subscriber data modification procedures are initiated by the OMC (see figures 20.3/1 , 20.3/2, 20.3/8 and 20.3/9).



- 1) Delete Subscriber
- 2) MAP_CANCEL_LOCATION
- 3) MAP_CANCEL_LOCATION_ACK
- 4) Subscriber Deleted

Figure 20.3/1: Subscriber deletion procedure

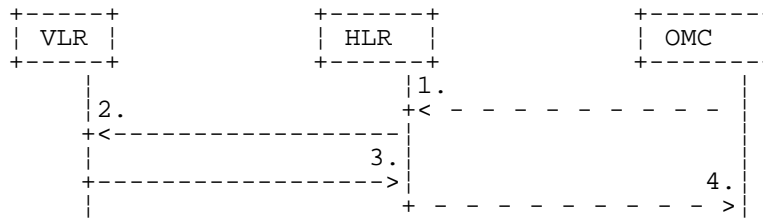
In the subscriber deletion procedure the subscriber data should be removed from the VLR and from the HLR. The HLR uses the MAP_CANCEL_LOCATION service.



- 1) Delete GPRS Subscriber
- 2) MAP_CANCEL_LOCATION
- 3) MAP_CANCEL_LOCATION_ACK
- 4) GPRS Subscriber Deleted

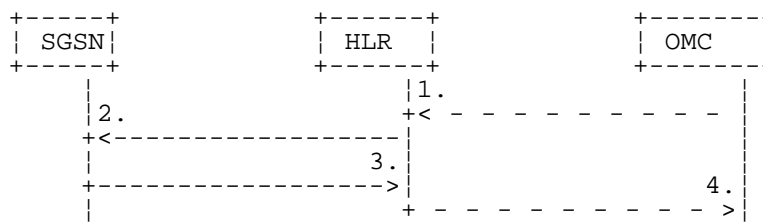
Figure 20.3/8: Subscriber deletion procedure for GPRS

In the subscriber deletion procedure the subscriber data should be removed from the SGSN 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 20.3/2: Subscriber data modification procedure



- 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 20.3/9: Subscriber data modification procedure for GPRS

In the subscriber data modification procedure the subscriber data is modified in the HLR and when necessary also in the VLR or in the SGSN. The HLR initiates either the MAP_INSERT_SUBSCRIBER_DATA, MAP_DELETE_SUBSCRIBER_DATA or MAP_CANCEL_LOCATION service depending on the modified data.

20.3.1 Procedures in the HLR

20.3.1.1 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 or to the SGSN where the subscriber is registered.

The subscriber deletion procedure in the HLR is shown in the figure 20.3/3.

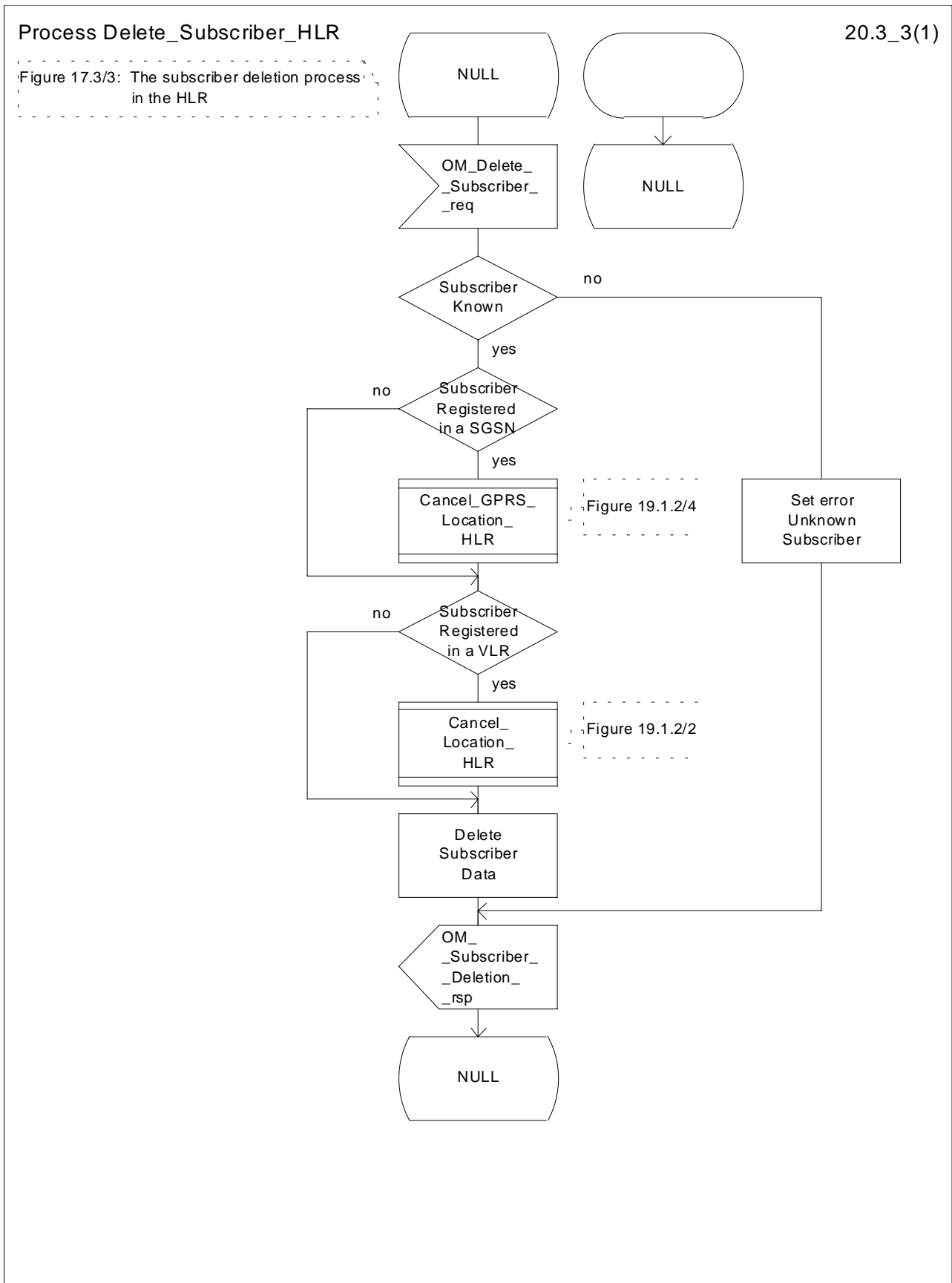


Figure 20.3/3: Process Delete_Subscriber_HLR

20.3.1.2 Subscriber data modification procedure

The OMC can modify the subscriber data in several different ways. The modifications can be categorized in following groups:

- a) no effect in the VLR;
- b) data shall be modified in both the HLR and the VLR;
- c) withdrawal of a basic service or a supplementary service requiring change to VLR data;
- 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;
- f) no effect in the SGSN;
- g) data shall be modified in both the HLR and the SGSN;
- h) withdrawal of a GPRS subscription data or a basic service or both requiring change to SGSN data;
- i) modification affects on the roaming of the subscriber and the subscriber shall be removed from the SGSN data base;
- j) withdrawal of GPRS Subscription related to Network Access Mode;
- k) withdrawal of non-GPRS Subscription related to Network Access Mode;

In case "b" and "g" the MAP_INSERT_SUBSCRIBER_DATA service is initiated in the HLR.

In case "c" and "h" the MAP_DELETE_SUBSCRIBER_DATA service is initiated in the HLR.

In cases "d", "e", "i", "j" and "k" the MAP_CANCEL_LOCATION service is initiated in the HLR.

If the result of a primitive received from the VLR or from the SGSN 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 or by the SGSN.

The subscriber data modification procedure in the HLR is shown in the figures 20.3/4, 20.3/5 and 25.7/2.

Process Modify_Data_HLR

20.3_4.1(2)

Figure 20.3/4: The subscriber data modification process in the HLR

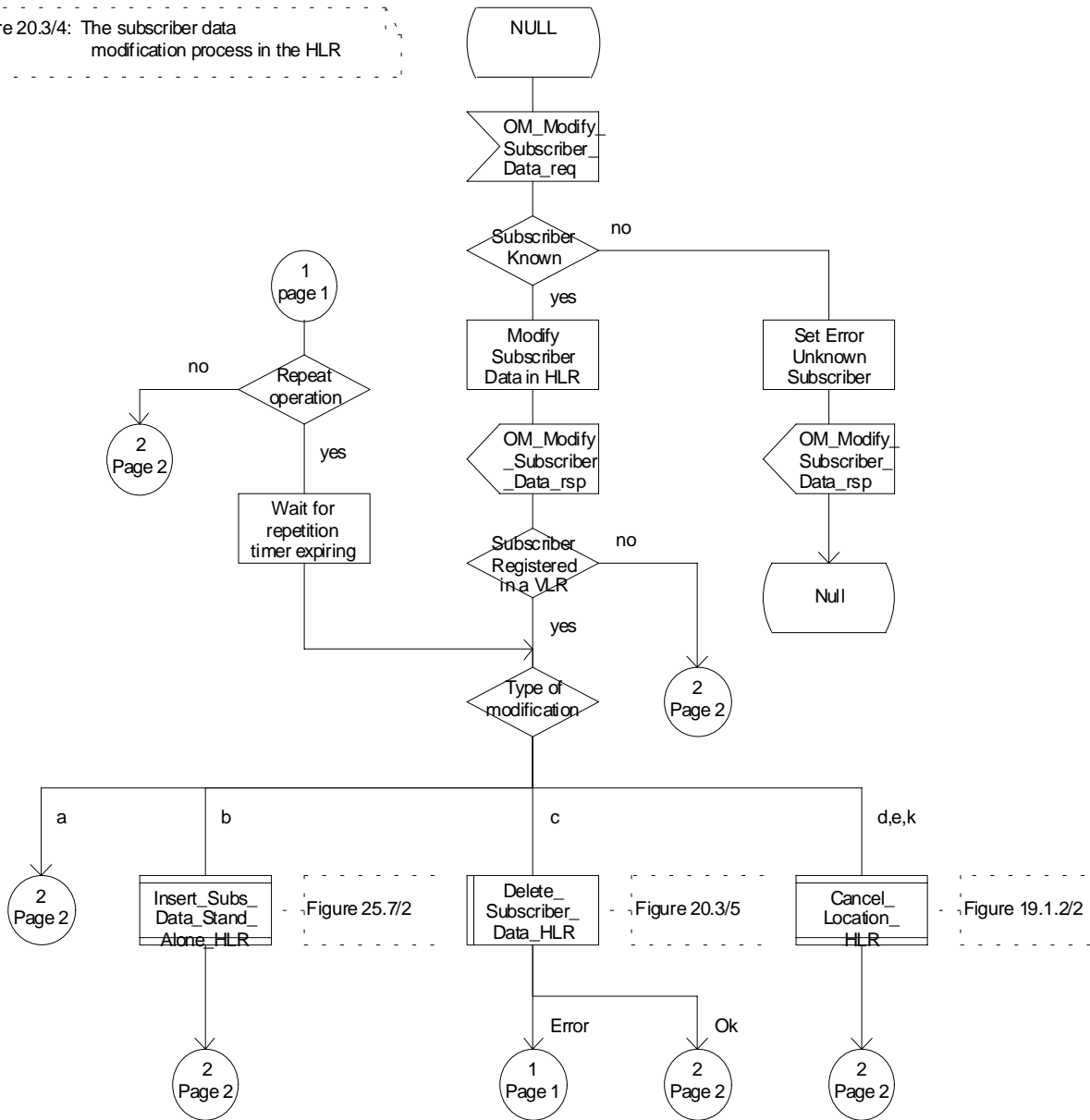


Figure 20.3/4 (sheet 1 of 2): Process Modify_Data_HLR

Process Modify_Data_HLR

20.3_4.2(2)

Figure 20.3/4: The subscriber data modification process in the HLR

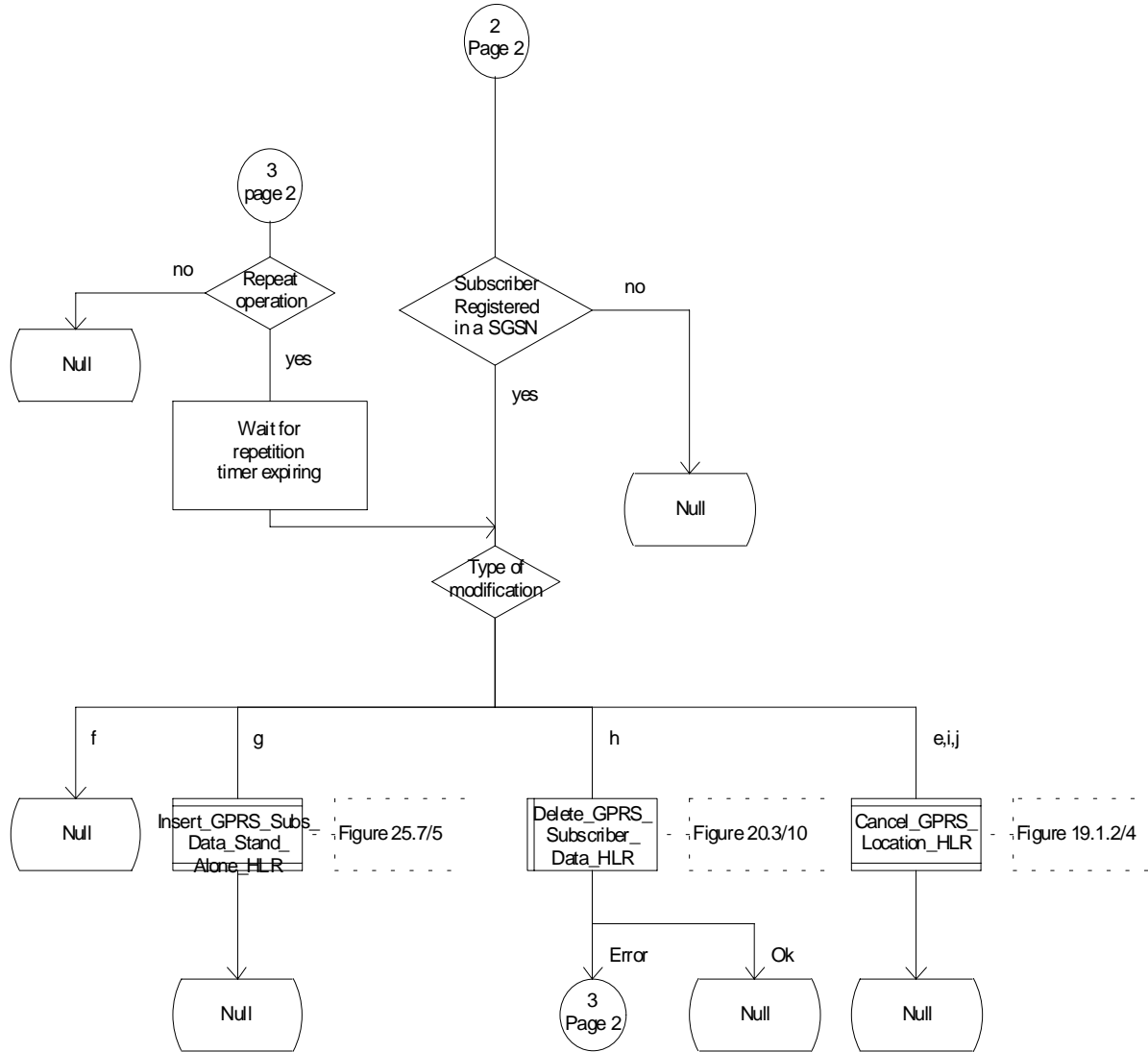


Figure 20.3/4 (sheet 2 of 2): Process Modify_Data_HLR

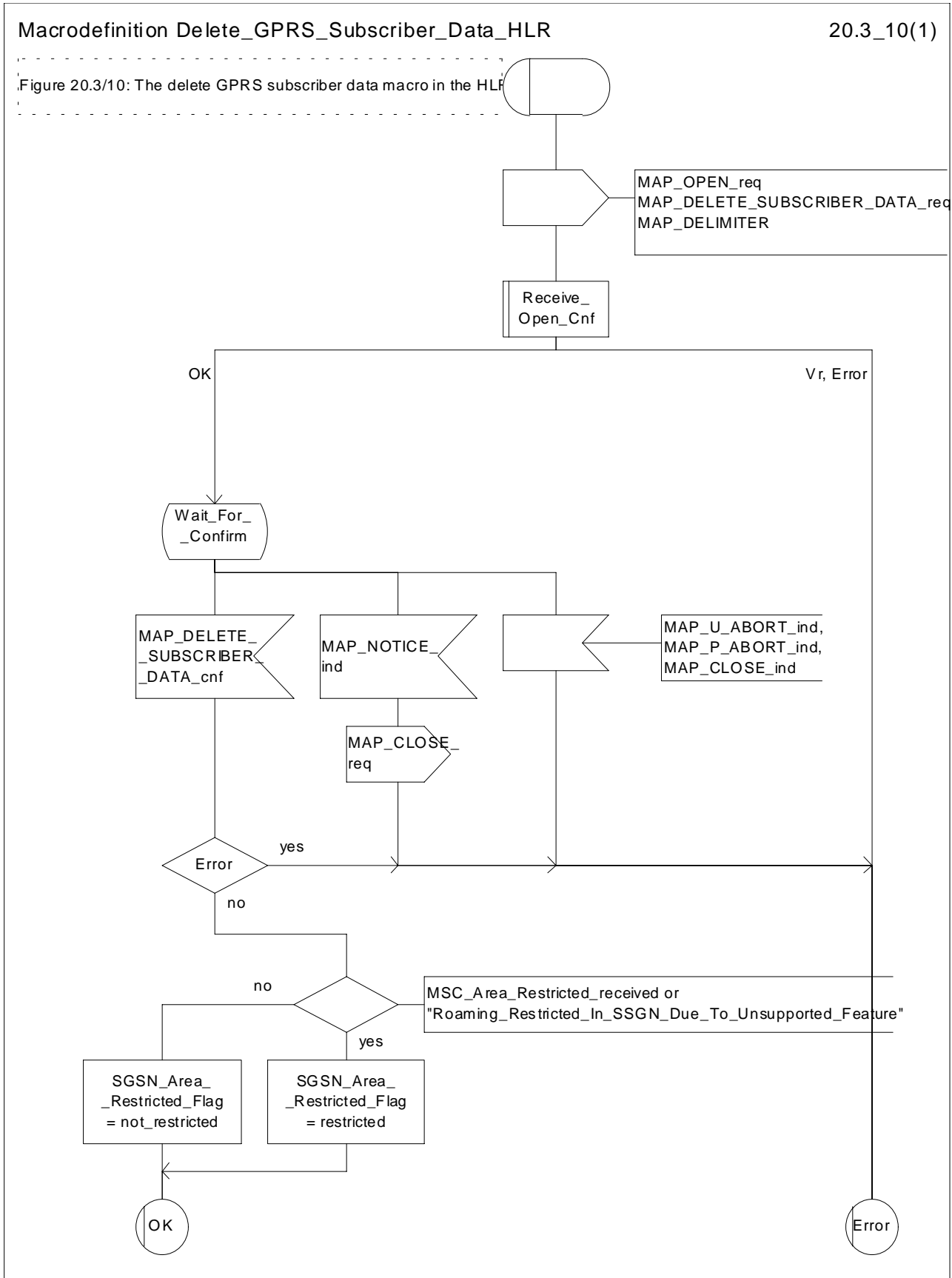


Figure 20.3/10: Macro Delete_GPRS_Subscriber_Data_HLR

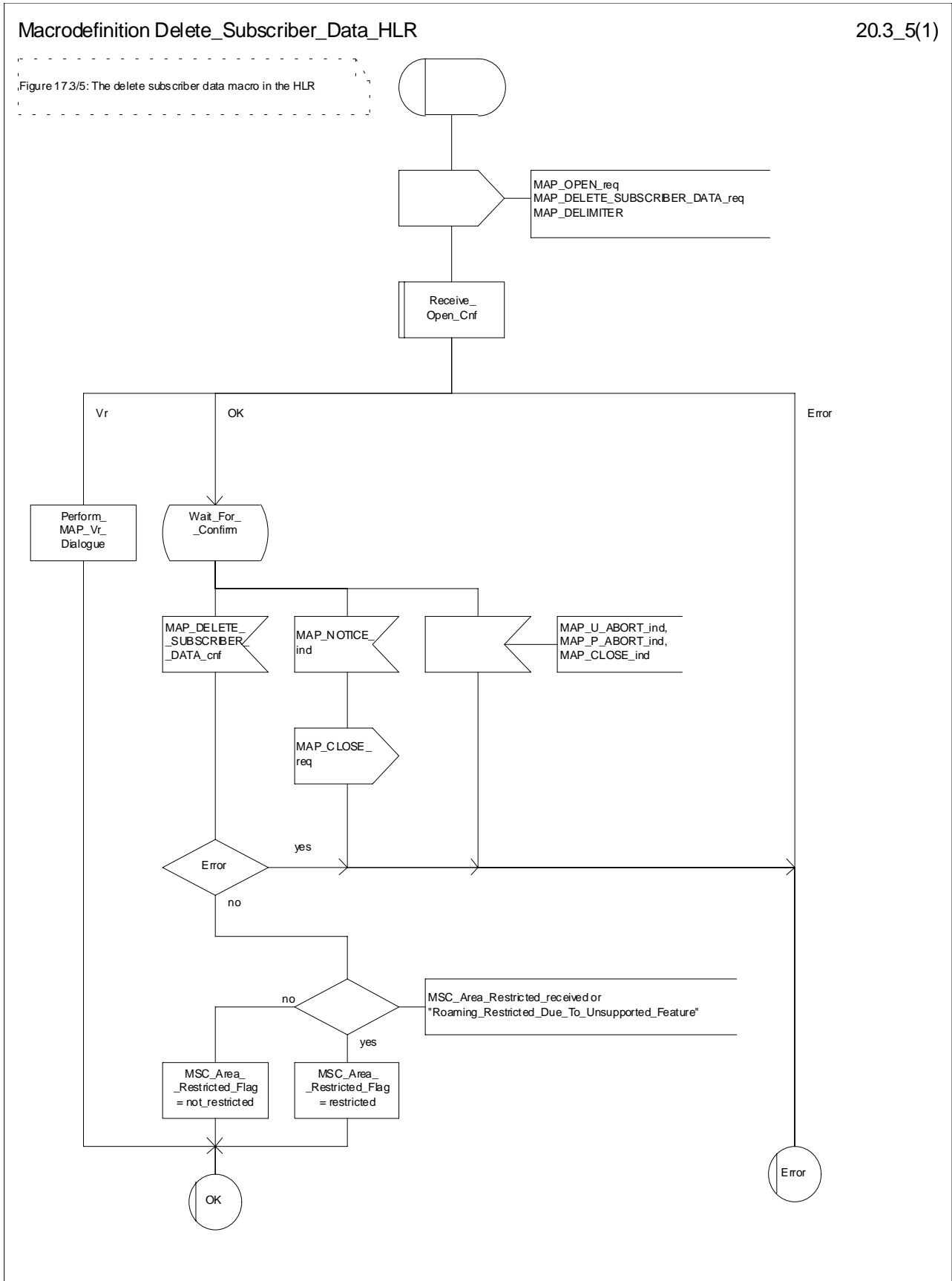


Figure 20.3/5: Macro Delete_Subscriber_Data_HLR

20.3.2 Procedures in the VLR

20.3.2.1 Subscriber deletion procedure

The subscriber deletion procedure in the VLR is described in the subclause 19.1.

20.3.2.2 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 20.3/6, 20.3/7 and 25.7/1.

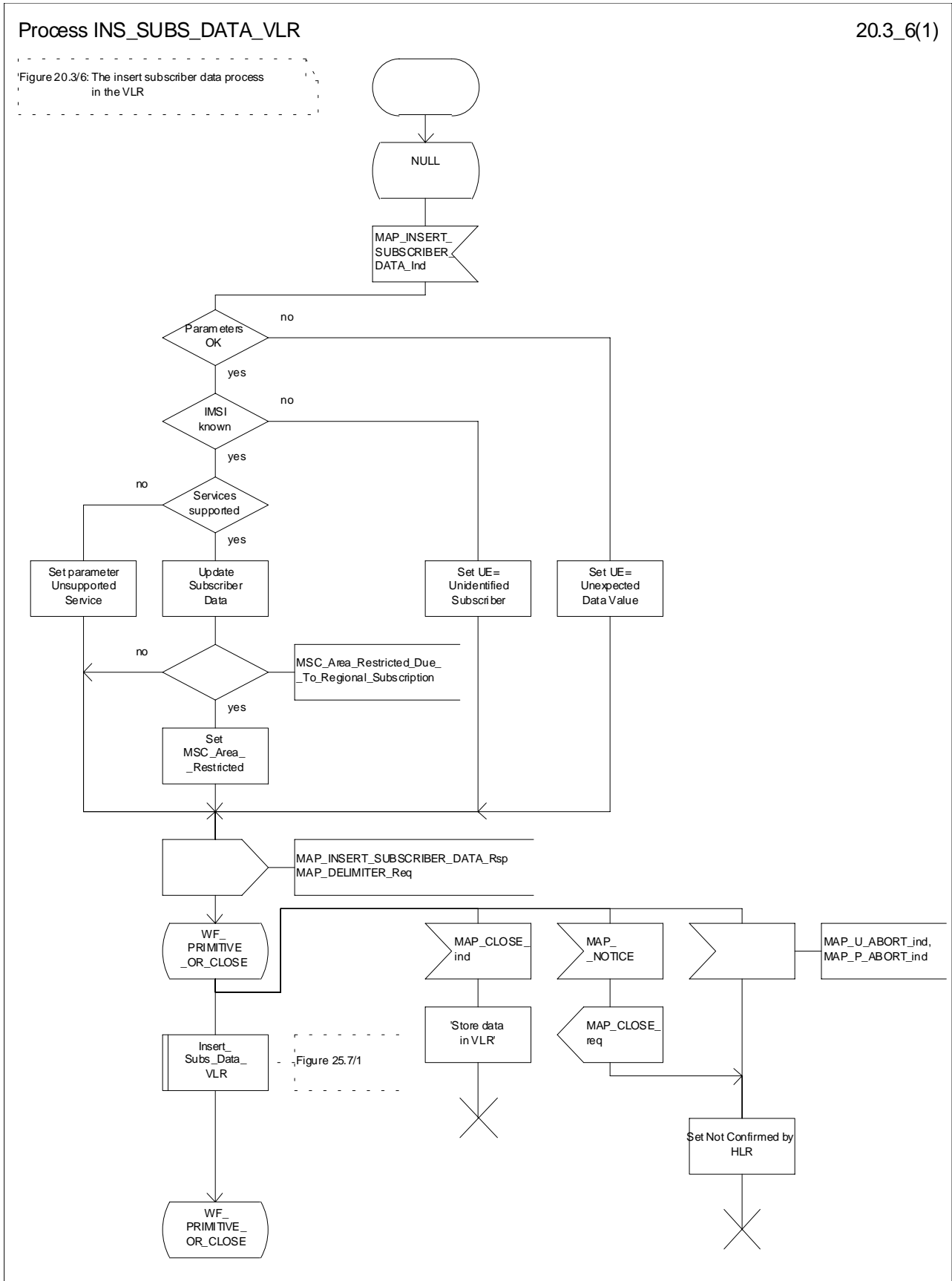


Figure 20.3/6: Process INS_SUBS_DATA_VLR

Process Delete_Subscriber_Data_VLR

20.3_7(1)

Figure 20.3/7: The delete subscriber data process in the VLR

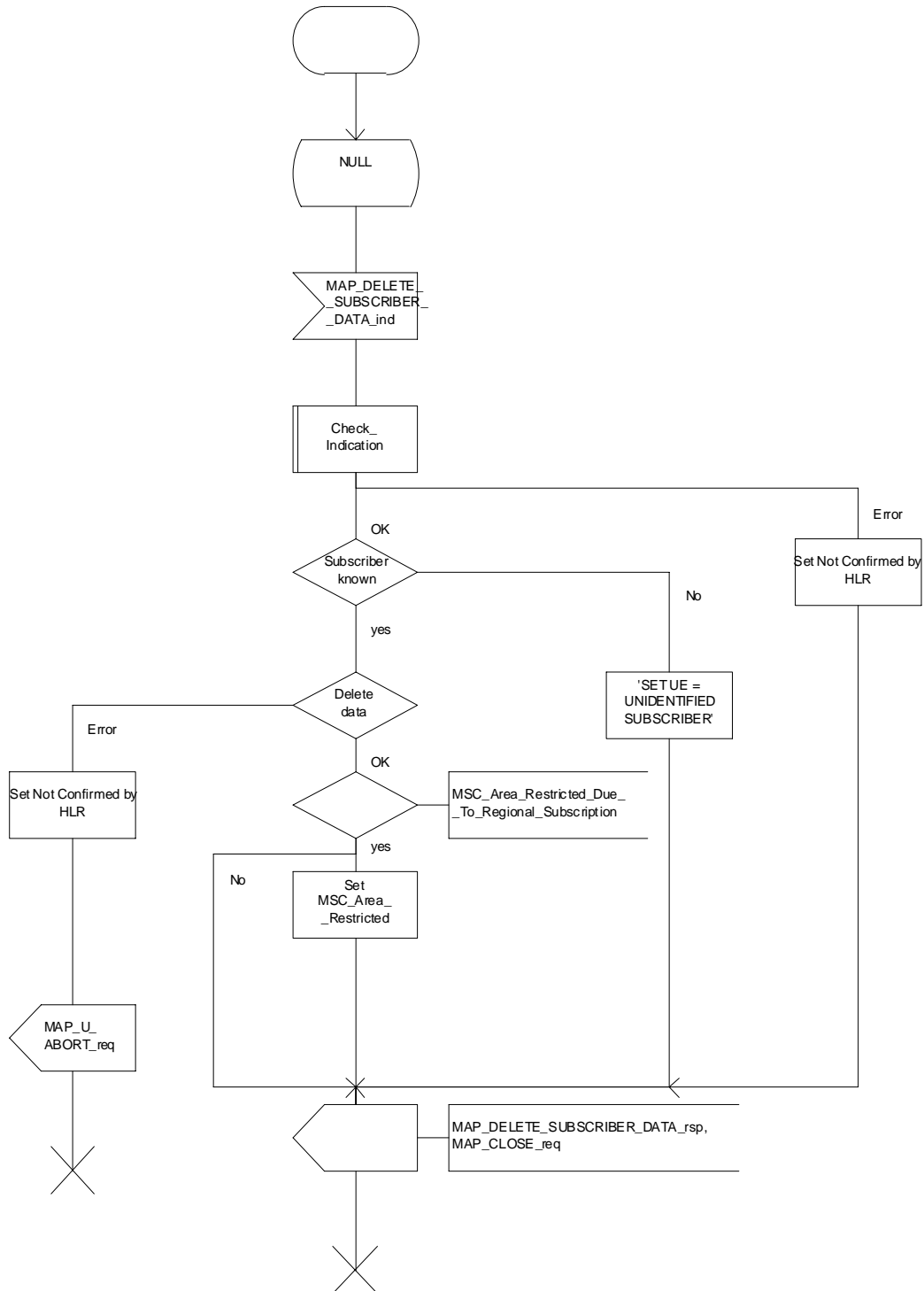


Figure 20.3/7: Process Delete_Subscriber_Data_VLR

20.3.3 Procedures in the SGSN

20.3.3.1 Subscriber deletion procedure

The subscriber deletion procedure in the SGSN is described in the subclause 19.1.

20.3.3.2 Subscriber data modification procedure

When receiving either the MAP_INSERT_SUBSCRIBER_DATA indication or the MAP_DELETE_SUBSCRIBER_DATA indication, the SGSN 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 SGSN will check the IMSI that is included in the primitive. If the IMSI is unknown, the error "Unidentified subscriber" is returned.

If the SGSN does not support received basic 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 SGSN area is restricted due to regional subscription, this is reported to the HLR.

If the updating of the subscriber data is not possible, the SGSN 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 SGSN is shown in the figures 20.3/11, 20.3/12 and 25.7/5.

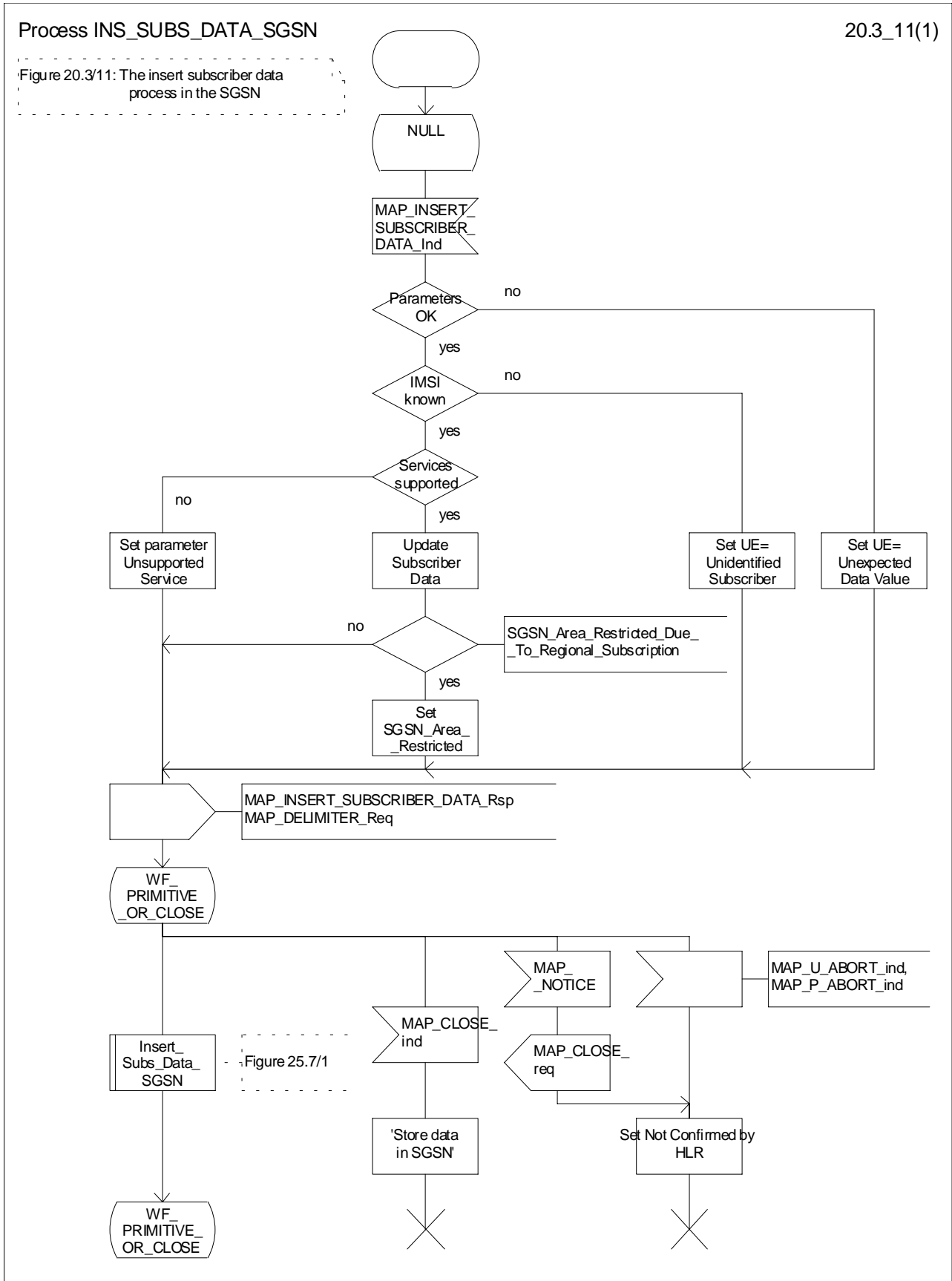


Figure 20.3/11: Process INS_SUBS_DATA_SGSN

Process Delete_Subscriber_Data_SGSN

20.3_12(1)

Figure 20.3/12: The delete subscriber data process in the SGSN

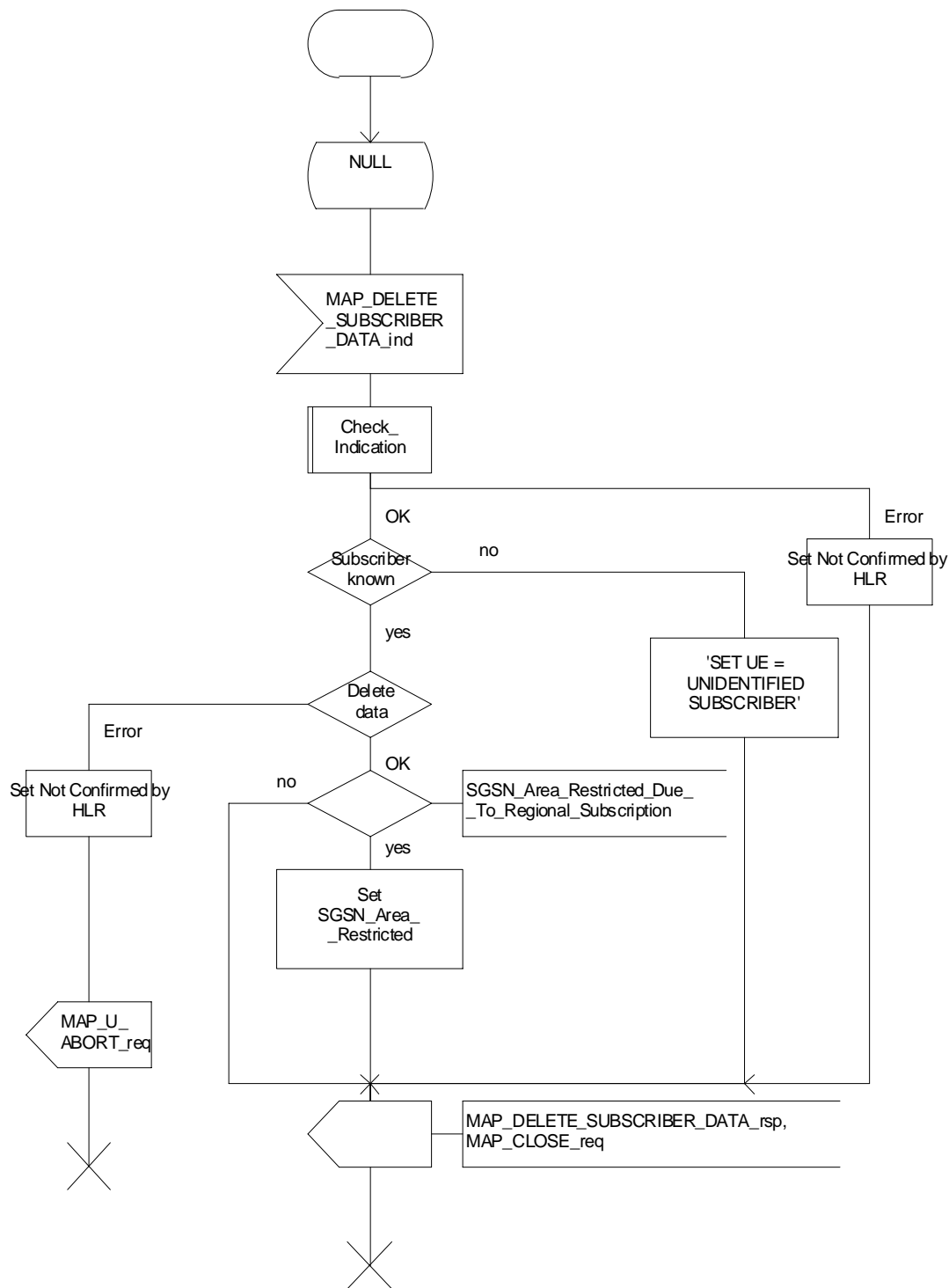
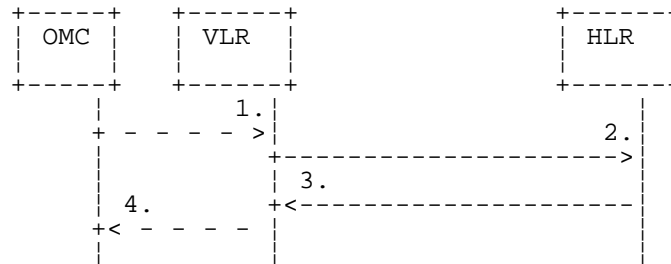


Figure 20.3/12: Process Delete_Subscriber_Data_SGSN

20.4 Subscriber Identity procedure

In the subscriber identity procedure the IMSI of the subscriber is retrieved from the HLR. The procedure is shown in figure 20.4/1.



- 1) Identity request
- 2) MAP_SEND_IMSI
- 3) MAP_SEND_IMSI_ACK
- 4) Identity confirm

Figure 20.4/1: The subscriber identity procedure

20.4.1 Subscriber identity procedure in the HLR

Opening of the dialogue is described in the macro Receive_Open_Ind in subclause 25.1, with outcomes:

- 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 20.4/2.

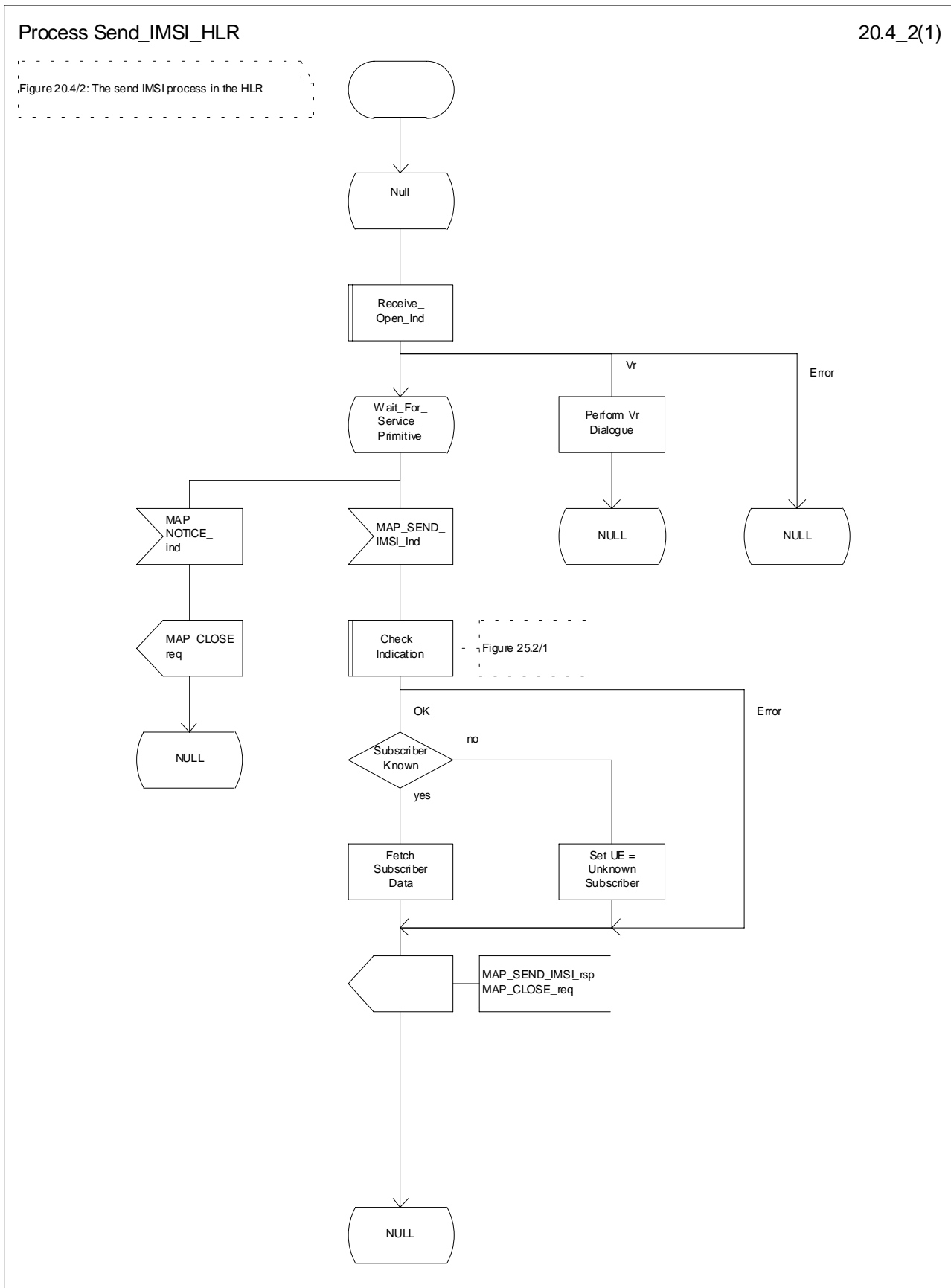


Figure 20.4/2: Process Send_IMSI_HLR

20.4.2 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 20.4/3.

Process Send_IMSI_VLR

20.4_3(1)

Figure 20.4/3: The send IMSI process in the VLR

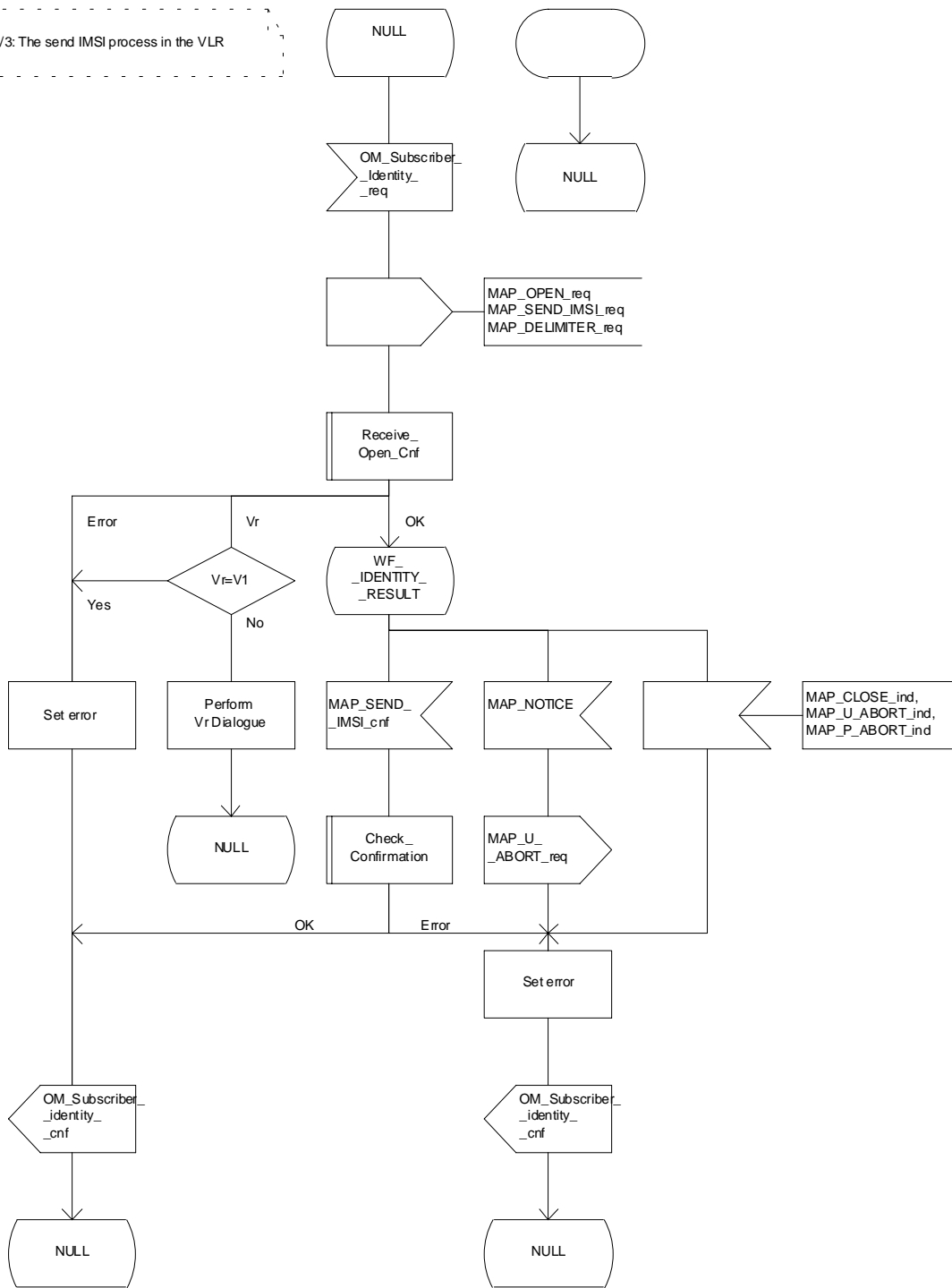


Figure 20.4/3: Process Send_IMSI_VLR

21 Call handling procedures

21.1 General

The MAP call handling procedures are used:

- to retrieve routing information to handle a mobile terminating call;
- to transfer control of a call back to the GMSC if the call is to be forwarded;
- to retrieve and transfer information between anchor MSC and relay MSC for inter MSC group calls / broadcast calls;
- to allocate resources in an SIWFS;
- to handle the reporting of MS status for call completion services;
- to handle the notification of remote user free for CCBS;
- to handle the alerting and termination of ongoing call activities for a specific subscriber.

The procedures to handle a mobile originating call and a mobile terminating call after the call has arrived at the destination MSC do not require any signalling over a MAP interface. These procedures are specified in GSM 03.18 [97].

The stage 2 specification for the retrieval of routing information to handle a mobile terminating call is in GSM 03.18 [97]; modifications to this procedure for CAMEL are specified in 3G TS 23.078 [98], for optimal routing of a basic mobile-to-mobile call in GSM 03.79 [99] and for CCBS in GSM 03.93. The interworking between the MAP signalling procedures and the call handling procedures for each entity (GMSC, HLR and VLR) is shown by the transfer of signals between these procedures.

The stage 2 specification for the transfer of control of a call back to the GMSC if the call is to be forwarded is in GSM 03.79 [99]. The interworking between the MAP signalling procedures and the call handling procedures for each entity (VMSC and GMSC) is shown by the transfer of signals between these procedures.

The stage 2 specifications for inter MSC group calls / broadcast calls are in GSM 03.68 and GSM 03.69. The interworking between the MAP signalling procedures and the group call /broadcast call procedures for each entity (Anchor MSC and Relay MSC) is shown by the transfer of signals between these procedures.

The stage 2 specification for the allocation of resources in an SIWFS is in GSM 03.54. The interworking between the MAP signalling procedures and the call handling procedures for each entity (VMSC and SIWFS) is shown by the transfer of signals between these procedures.

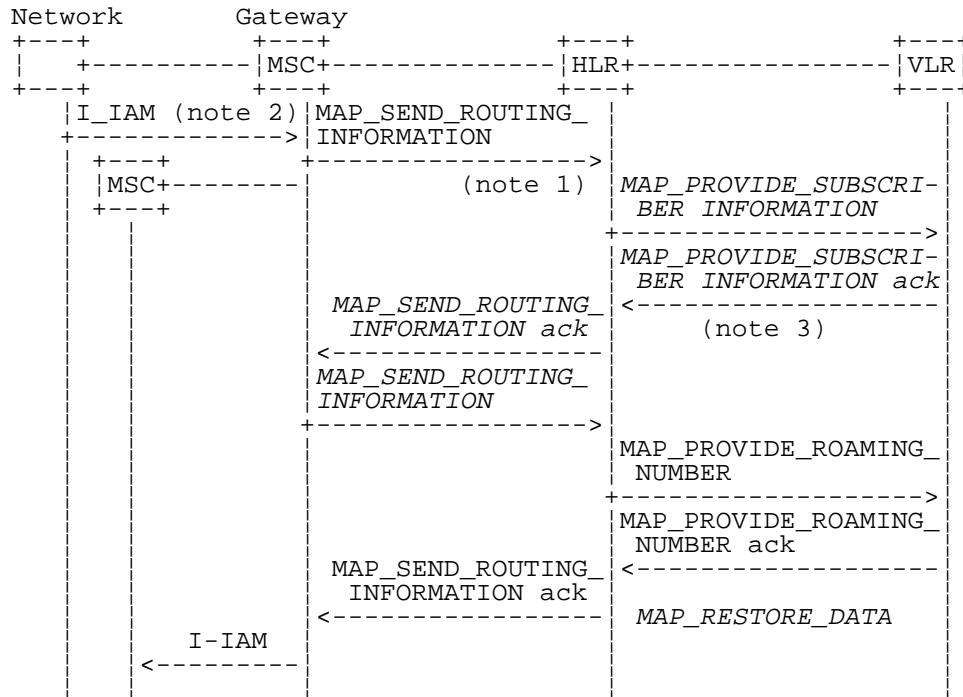
The interworking between the call handling procedures and signalling protocols other than MAP is shown in GSM 03.18, 3G TS 23.078 and GSM 03.79.

The stage 2 specification for the handling of reporting of MS status for call completion services and notification of remote user free for CCBS is in GSM 03.93.

21.2 Retrieval of routing information

21.2.1 General

The message flows for successful retrieval of routing information for a mobile terminating call are shown in figure 21.2/1 (mobile terminating call which has not been optimally routed) and 21.2/2 (mobile-to-mobile call which has been optimally routed).



Notes:

xxx = Optional Procedure

NOTE 1: This service may also be used by an ISDN exchange for obtaining routing information from the HLR.

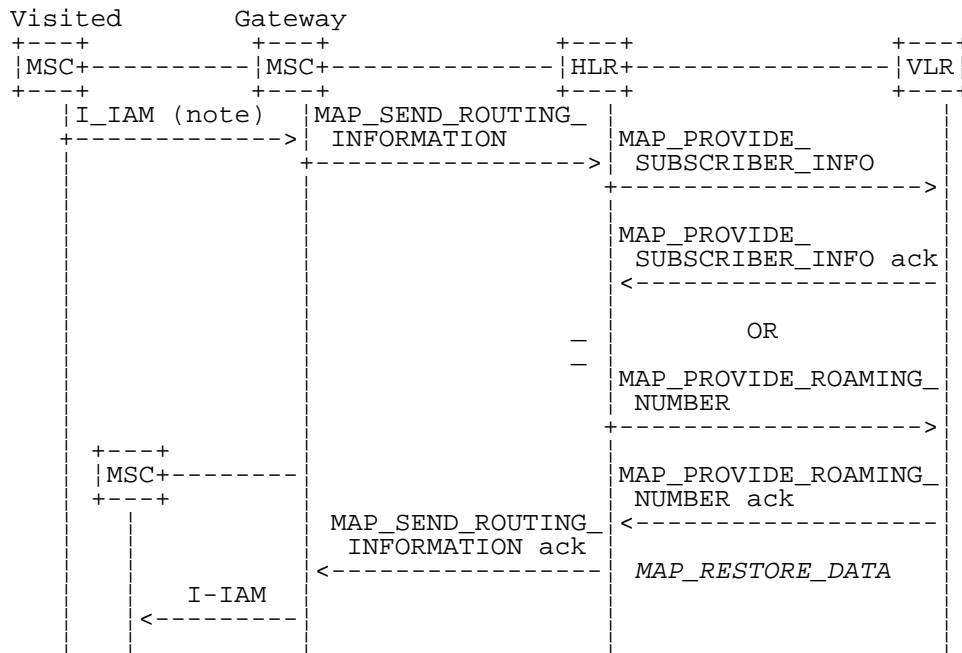
NOTE 2: TUP or ISUP may be used in signalling between MSCs, depending on the network type between the MSCs. For further details on the TUP and ISUP procedures refer to the following ITU-T Recommendations and ETSI specification:

Q.721-725 - Telephone User Part (TUP);

ETS 300 356-1 - Integrated Services Digital Network (ISDN); Signalling System No.7; ISDN User Part (ISUP) version 2 for the international interface; Part 1: Basic services.

NOTE 3: As a network operator option, the HLR sends MAP_PROVIDE_SUBSCRIBER_INFORMATION to the VLR. For further details on the CAMEL procedures refer to GSM TS 03.78;

Figure 21.2/1: Message flow for retrieval of routing information (non-optimally routed call)



Notes:

xxx = Optional Procedure

For Optimal Routing phase 1, only one of the information flows for Provide Subscriber Info and Provide Roaming Number is used. For later phases of Optimal Routing, the HLR may return a MAP_SEND_ROUTEING_INFORMATION ack after the Provide Subscriber Info information flow, and the GMSC may send a second MAP_SEND_ROUTEING_INFORMATION, which will trigger the Provide Roaming Number information flow.

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 & ETSI specification:

Q.721-725 - Telephone User Part (TUP);

ETS 300 356-1 - Integrated Services Digital Network (ISDN); Signalling System No.7; ISDN User Part (ISUP) version 2 for the international interface; Part 1: Basic services.

Figure 21.2/2: Message flow for retrieval of routing information (optimally routed call)

The following MAP services are used to retrieve routing information:

MAP_SEND_ROUTING_INFORMATION	see subclause 10.1;
MAP_PROVIDE_ROAMING_NUMBER	see subclause 10.2;
MAP_PROVIDE_SUBSCRIBER_INFO	see subclause 8.11.2;
MAP_RESTORE_DATA	see subclause 8.10.3.

21.2.2 Process in the GMSC

The MAP process in the GMSC to retrieve routing information for a mobile terminating call is shown in figure 21.2/3. The MAP process invokes macros not defined in this subclause; the definitions of these macros can be found as follows:

Receive_Open_Cnf see subclause 25.1.2;

Check_Confirmation see subclause 25.2.2.

Successful Outcome

When the MAP process receives a Send Routeing Info request from the call handling process in the GMSC, it requests a dialogue with the HLR whose identity is contained in the Send Routeing Info request by sending a MAP_OPEN service request, requests routing information using a MAP_SEND_ROUTING_INFORMATION service request and invokes the macro Receive_Open_Cnf to wait for the response to the dialogue opening request. If the dialogue opening is successful, the MAP process waits for a response from the HLR.

If the MAP process receives a MAP_SEND_ROUTING_INFORMATION service confirm from the HLR, the MAP process invokes the macro Check_Confirmation to check the content of the confirm. If the MAP_SEND_ROUTING_INFORMATION confirm from the HLR cannot be carried in a single TC-Result component, it is carried in one or more TC-Result-NL components (each sent in a TC-CONTINUE), followed by a TC-Result-L component in a TC-END message.

If the macro Check_Confirmation takes the OK exit, the MAP process sends a Send Routeing Info ack containing the routing information received from the HLR to the call handling process in the GMSC and returns to the idle state.

Earlier version MAP dialogue with the HLR

If the macro Receive_Open_Cnf takes the Vr exit, the MAP process checks whether this is an OR interrogation (indicated by the inclusion of the OR interrogation parameter in the MAP_SEND_ROUTING_INFORMATION service request).

If this is not an OR interrogation, the GMSC performs the earlier version MAP dialogue as specified in [51] and the process returns to the idle state.

If this is an OR interrogation, the MAP process sends a Send Routeing Info negative response indicating OR not allowed to the call handling process in the GMSC and returns to the idle state.

Dialogue opening failure

If the macro Receive_Open_Cnf indicates that the dialogue with the HLR could not be opened, the MAP process sends an Abort to the call handling process in the GMSC and returns to the idle state.

Error in MAP_SEND_ROUTING_INFORMATION confirm

If the MAP_SEND_ROUTING_INFORMATION service confirm contains a user error or a provider error, or the macro Check_Confirmation indicates that there is a data error, the MAP process sends a Send Routeing Info negative response to the call handling process in the GMSC and returns to the idle state.

Call release

If the call handling process in the GMSC indicates that the call has been aborted (i.e. prematurely released by the calling subscriber), the MAP process returns to the idle state. Any response from the HLR will be discarded.

Abort of HLR dialogue

After the dialogue with the HLR has been established, the MAP service provider may abort the dialogue by issuing a MAP_P_ABORT indication, or the HLR may send a MAP_U_ABORT indication or a MAP_CLOSE indication. In any of these cases, the MAP process sends a Send Routeing Info negative response to the call handling process in the GMSC and returns to the idle state.

If the MAP provider indicates a protocol problem by sending a MAP_NOTICE indication, the MAP process closes the dialogue with the HLR, sends a Send Routeing Info negative response indicating system failure to the call handling process in the GMSC and returns to the idle state.

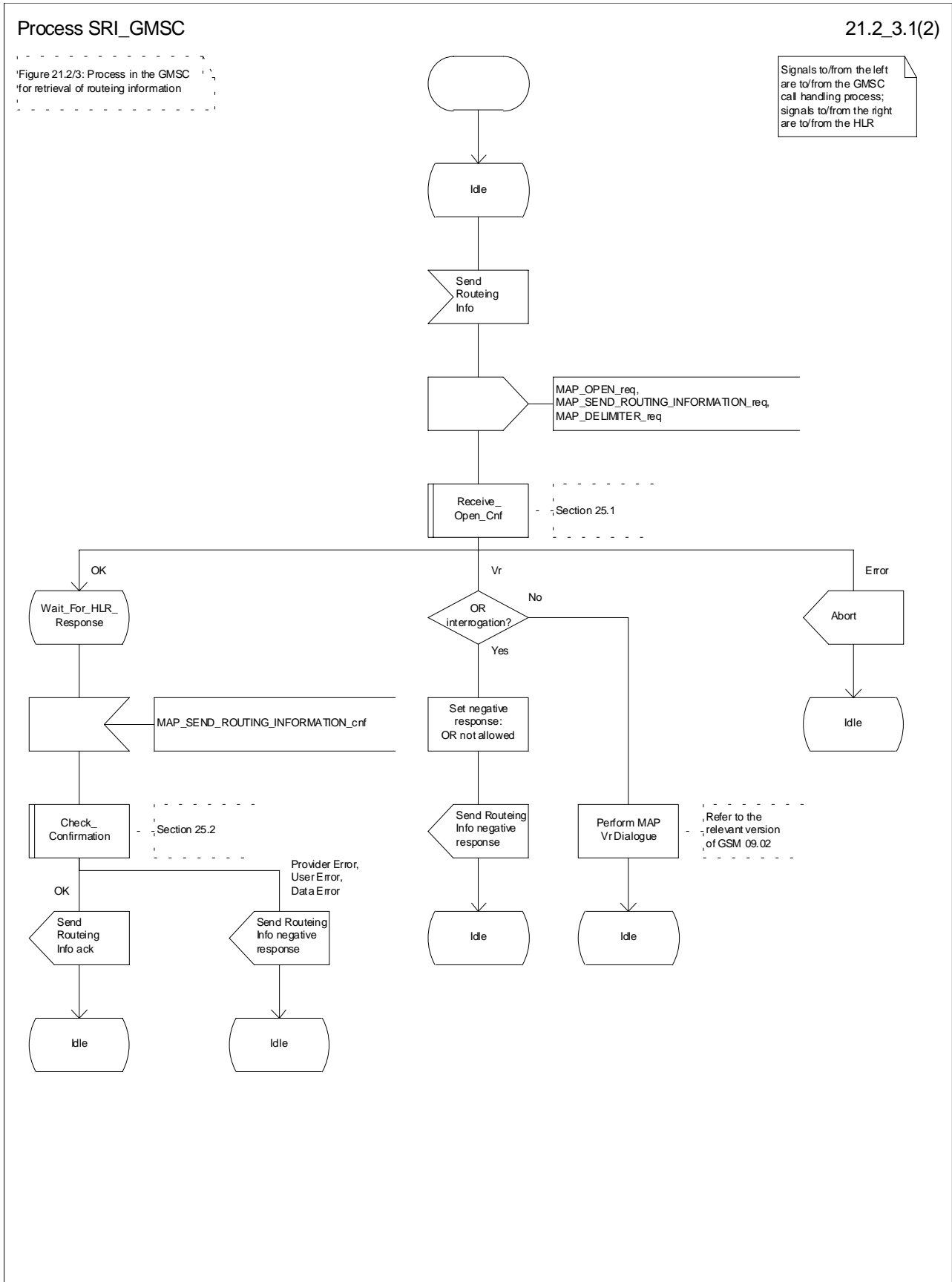


Figure 21.2/3 (sheet 1 of 2): Process SRI_GMSC

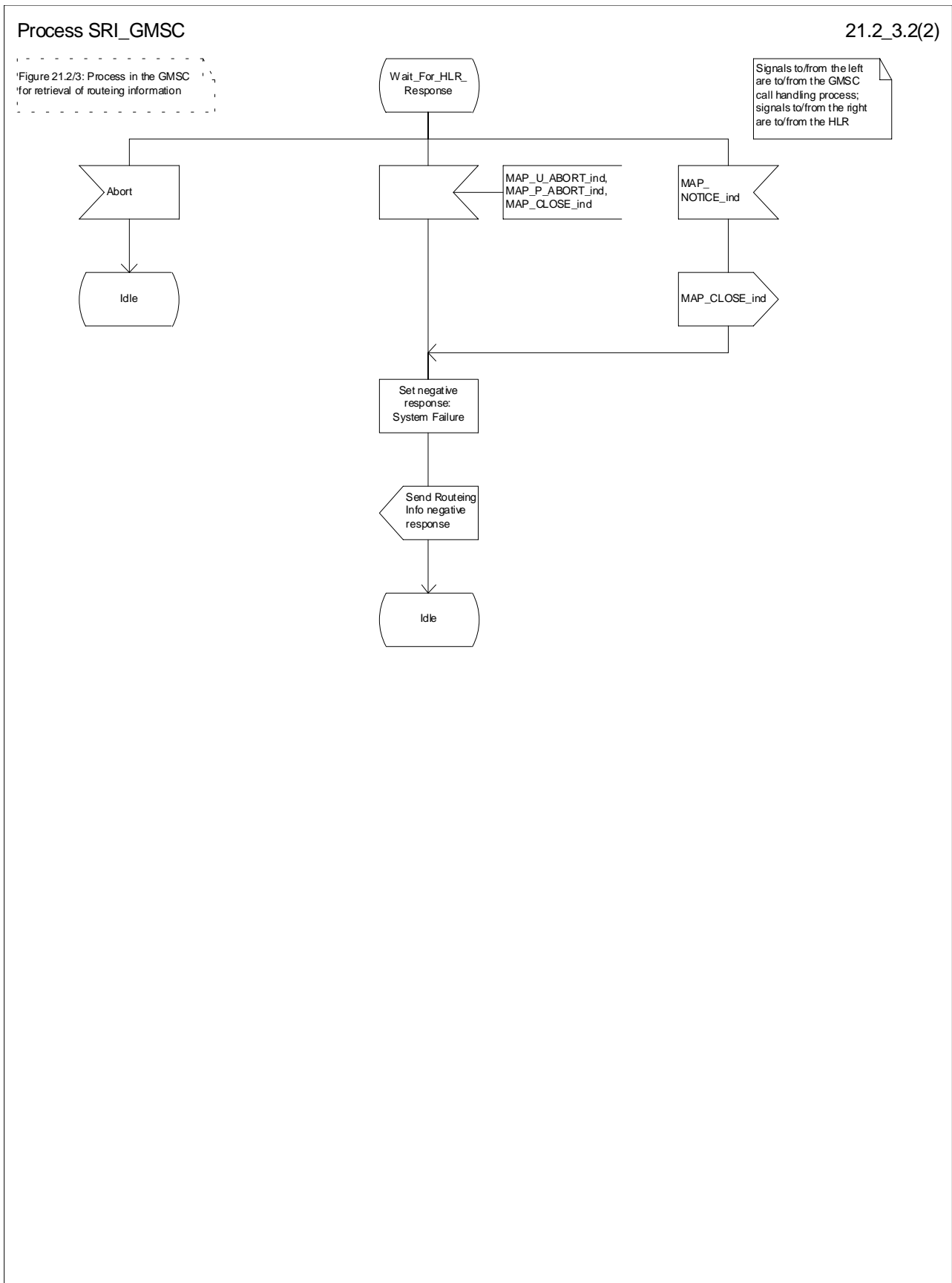


Figure 21.2/3 (sheet 2 of 2): Process SRI_GMSC

21.2.3 Procedures in the HLR

The MAP process in the HLR to retrieve routing information for a mobile terminating call is shown in figure 21.2/4. The MAP process invokes macros not defined in this subclause; the definitions of these macros can be found as follows:

Receive_Open_Ind	see subclause 25.1.1;
Receive_Open_Cnf	see subclause 25.1.2;
Check_Confirmation	see subclause 25.2.2.

Successful outcome

When the MAP process receives a MAP_OPEN indication with the application context locInfoRetrieval, it checks it by invoking the macro Receive_Open_Ind.

If the macro takes the OK exit, the MAP process waits for a service indication.

If a MAP_SEND_ROUTING_INFORMATION service indication is received, the MAP process sends a Send Routing Info request to the call handling process in the HLR, and waits for a response. The Send Routing Info request contains the parameters received in the MAP_SEND_ROUTING_INFORMATION service indication.

If the call handling process in the HLR returns a Send Routing Info ack, the MAP process constructs a MAP_SEND_ROUTING_INFORMATION service response containing the routing information contained in the Send Routing Info ack, constructs a MAP_CLOSE service request, sends them to the GMSC and returns to the idle state. If the MAP_SEND_ROUTING_INFORMATION response cannot be carried in a single TC-Result component, it is carried in one or more TC-Result-NL components (each sent in a TC-CONTINUE), followed by a TC-Result-L component in a TC-END message.

If the call handling process in the HLR returns a Provide Subscriber Info request, the MAP process requests a dialogue with the VLR whose identity is contained in the Provide Subscriber Info request by sending a MAP_OPEN service request, requests the subscriber status using a MAP_PROVIDE_SUBSCRIBER_INFO service request, and invokes the macro Receive_Open_Cnf to wait for the response to the dialogue opening request.

If the macro takes the OK exit, the MAP process waits for the response from the VLR.

If the MAP process receives a MAP_PROVIDE_SUBSCRIBER_INFO service confirm, it invokes the macro Check_Confirmation to check the content of the confirm.

If the Check_Confirmation macro takes the OK exit, the MAP process sends a Provide Subscriber Info ack containing the information received in the MAP_PROVIDE_SUBSCRIBER_INFO service confirm to the call handling process in the HLR, and waits for a response. The handling of the response from the call handling process in the HLR is described above.

If the MAP_PROVIDE_SUBSCRIBER_INFO service confirm contains a provider error or a data error, the MAP process sends a Provide Subscriber Info negative response indicating the type of error to the call handling process in the HLR, and waits for a response. The handling of the response from the call handling process in the HLR is described above.

NOTE: The 'User Error' exit from the macro Check_Confirmation is shown for formal completeness; the MAP_PROVIDE_SUBSCRIBER_INFO_cnf primitive cannot contain a user error.

If the call handling process in the HLR returns a Provide Roaming Number request, the MAP process requests a dialogue with the VLR whose identity is contained in the Provide Roaming Number request by sending a MAP_OPEN service request, requests a roaming number using a MAP_PROVIDE_ROAMING_NUMBER service request, and invokes the macro Receive_Open_Cnf to wait for the response to the dialogue opening request.

If the macro takes the OK exit, the MAP process waits for the response from the VLR.

If the MAP process receives a MAP_PROVIDE_ROAMING_NUMBER service confirm, it invokes the macro Check_Confirmation to check the content of the confirm.

If the Check_Confirmation macro takes the OK exit, the MAP process sends a Provide Roaming Number ack containing the MSRN received in the MAP_PROVIDE_ROAMING_NUMBER service confirm to the call handling process in the HLR, and waits for a response. The handling of the response from the call handling process in the HLR is described above.

If the MAP_PROVIDE_ROAMING_NUMBER service confirm contains a user error or a provider error, or the macro Check_Confirmation indicates that there is a data error, the MAP process sends a Provide Roaming Number negative response indicating the type of error to the call handling process in the HLR, and waits for a response. The handling of the response from the call handling process in the HLR is described above.

Negative response from HLR call handling process

If the call handling process in the HLR returns a negative response, either before or after a dialogue with the VLR to obtain a roaming number, the MAP process constructs a MAP_SEND_ROUTING_INFORMATION service response containing the appropriate error, constructs a MAP_CLOSE service request, sends them to the GMSC and returns to the idle state.

Earlier version MAP Provide Roaming Number dialogue with the VLR

If the macro Receive_Open_Cnf takes the Vr exit after the MAP process has requested opening of a Provide Roaming Number dialogue with the VLR, the MAP process checks whether this is an OR interrogation (indicated by the inclusion of the OR interrogation parameter in the MAP_PROVIDE_ROAMING_NUMBER service request).

If this is not an OR interrogation, the HLR performs the earlier version MAP dialogue as specified in [51], relays the result of the dialogue to the HLR call handling process, and waits for a response. The handling of the response from the call handling process in the HLR is described above.

If this is an OR interrogation, the MAP process sends a Provide Roaming Number negative response indicating OR not allowed to the call handling process in the HLR and waits for a response. The handling of the response from the call handling process in the HLR is described above.

Failure of Provide Subscriber Info dialogue with the VLR

If the Receive_Open_Cnf macro takes the Vr exit or the Error exit after the MAP process has requested opening of a Provide Subscriber Info dialogue with the VLR, the MAP process sends a Provide Subscriber Info negative response indicating system failure to the call handling process in the HLR, and waits for a response. The handling of the response from the call handling process in the HLR is described above.

Failure of Provide Roaming Number dialogue with the VLR

If the Receive_Open_Cnf macro takes the Error exit after the MAP process has requested opening of a Provide Roaming Number dialogue with the VLR, the MAP process sends a Provide Roaming Number negative response indicating system failure to the call handling process in the HLR, and waits for a response. The handling of the response from the call handling process in the HLR is described above.

If the MAP process receives a MAP_U_ABORT, a MAP_P_ABORT or a premature MAP_CLOSE from the MAP provider, it sends a Provide Roaming Number negative response indicating system failure to the call handling process in the HLR, and waits for a response. The handling of the response from the call handling process in the HLR is described above.

If the MAP process receives a MAP_NOTICE from the MAP provider, it returns a MAP_CLOSE request to the MAP provider, sends a Provide Roaming Number negative response indicating system failure to the call handling process in the HLR, and waits for a response. The handling of the response from the call handling process in the HLR is described above.

Earlier version MAP dialogue with the GMSC

If the macro Receive_Open_Ind takes the Vr exit, the the HLR performs the earlier version MAP dialogue as specified in [51] and the process returns to the idle state.

Failure of dialogue opening with the GMSC

If the macro Receive_Open_Ind takes the Error exit, the MAP process returns to the idle state.

If the MAP provider sends a MAP_P_ABORT while the MAP process is waiting for a service indication, the MAP process returns to the idle state.

If the MAP provider sends a MAP_NOTICE while the MAP process is waiting for a service indication, the MAP process sends a MAP_CLOSE request to terminate the dialogue and returns to the idle state.

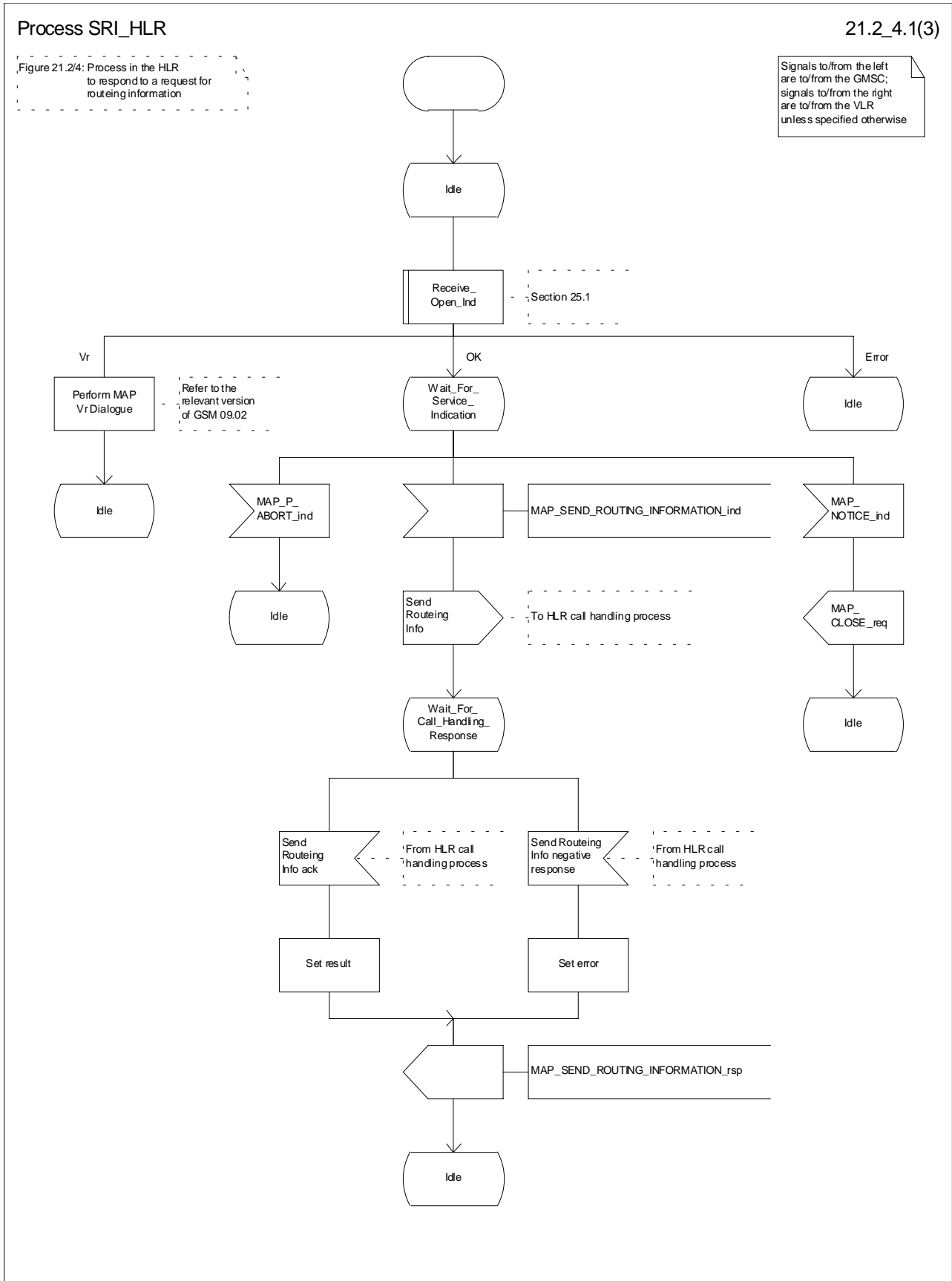


Figure 21.2/4 (sheet 1 of 3): Process SRI_HLR

Process SRI_HLR

21.2_4.2(3)

Figure 21.2/4: Process in the HLR to respond to a request for routing information

Signals to/from the left are to/from the GMSC; signals to/from the right are to/from the VLR unless specified otherwise

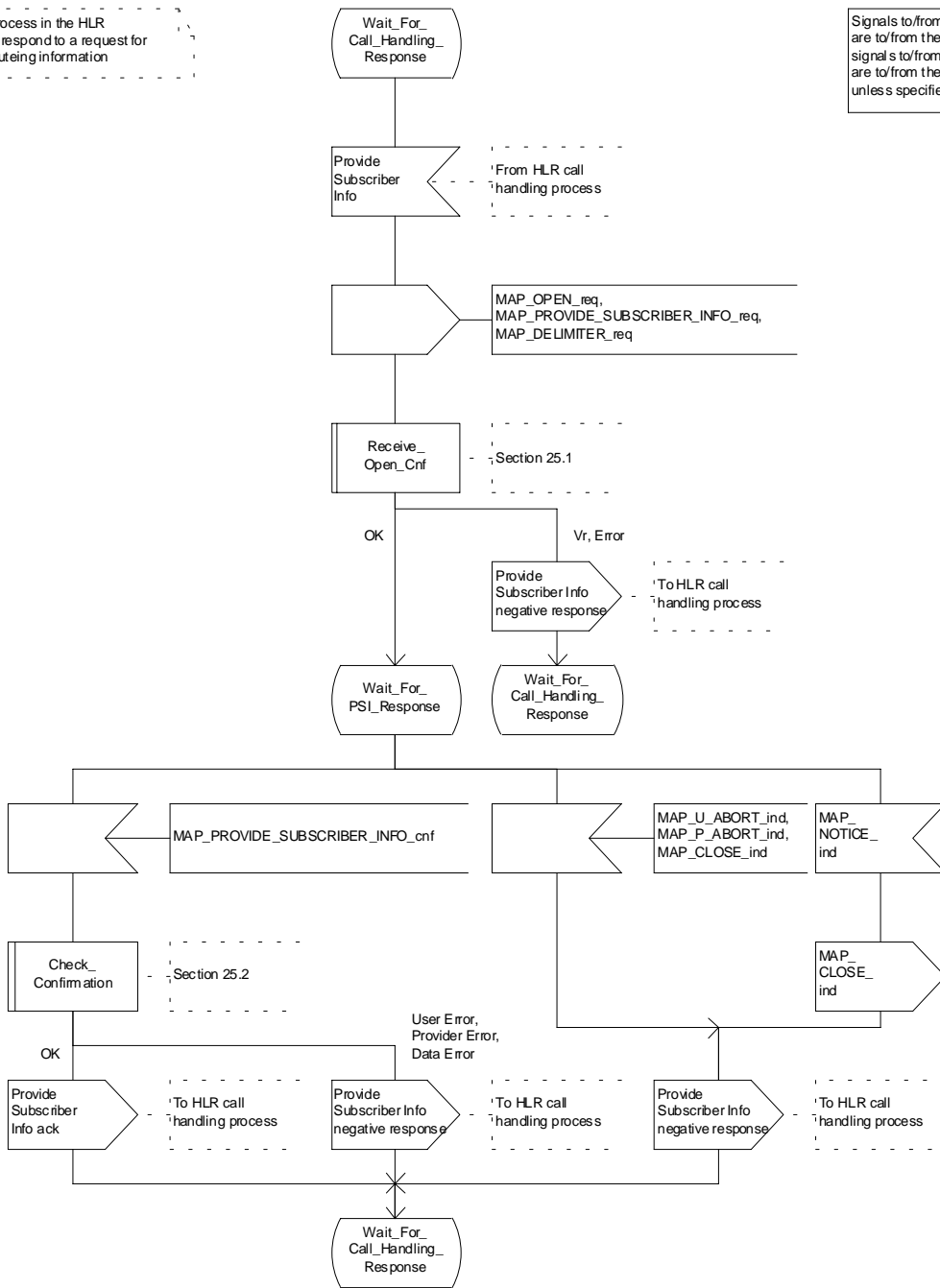


Figure 21.2/4 (sheet 2 of 3): Process SRI_HLR

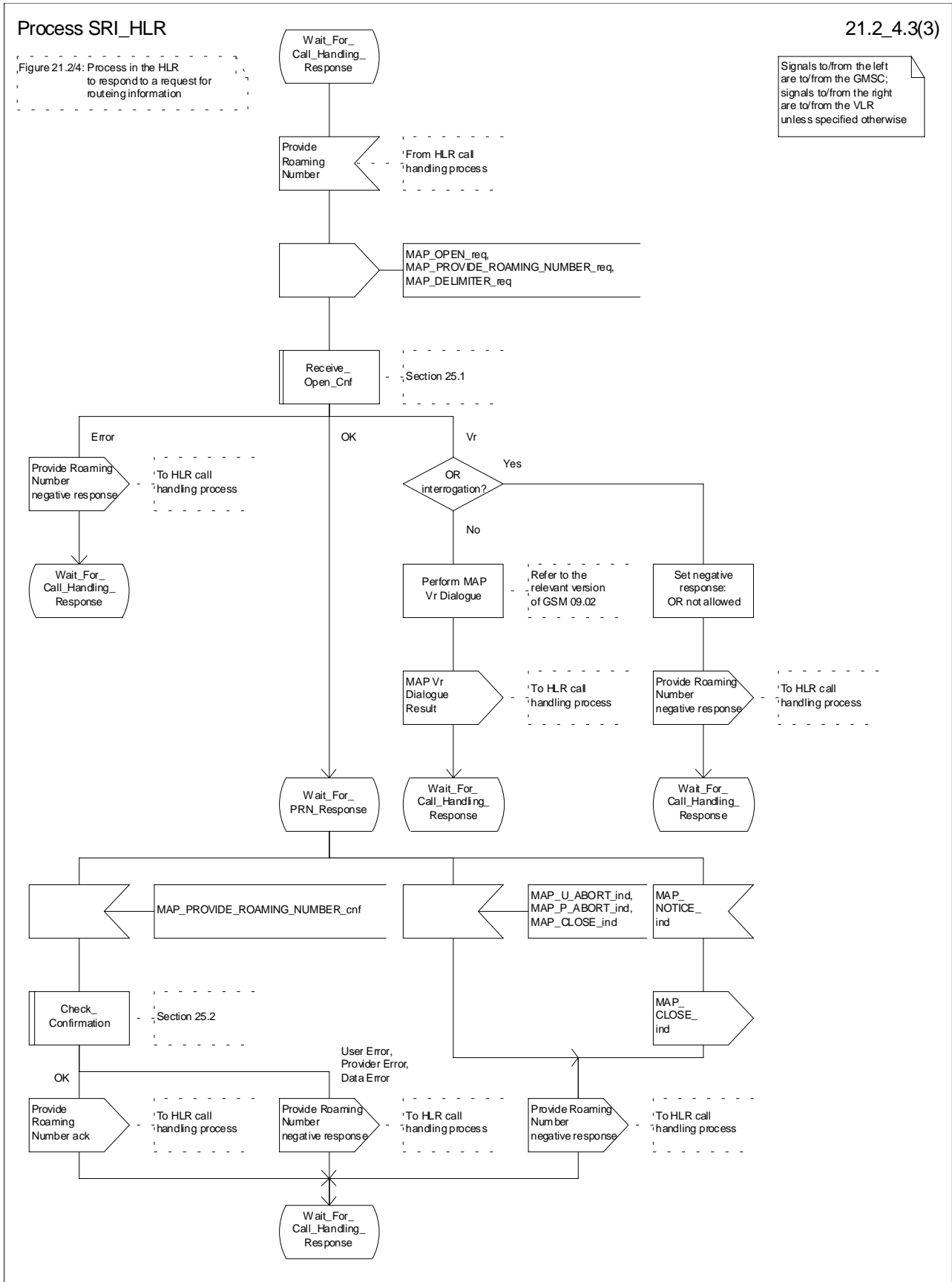


Figure 21.2/4 (sheet 3 of 3): Process SRI_HLR

21.2.4 Process in the VLR to provide a roaming number

The MAP process in the VLR to provide a roaming number for a mobile terminating call is shown in figure 21.2/5. The MAP process invokes a macro not defined in this subclause; the definition of this macro can be found as follows:

Receive_Open_Ind see subclause 25.1.1;

Successful outcome

When the MAP process receives a MAP_OPEN indication with the application context roamingNbEnquiry, it checks it by invoking the macro Receive_Open_Ind.

If the macro takes the OK exit, the MAP process waits for a service indication.

If a MAP_PROVIDE_ROAMING_NUMBER service indication is received, the MAP process sends a Provide Roaming Number request to the call handling process in the VLR, and waits for a response. The Provide Roaming Number request contains the parameters received in the MAP_PROVIDE_ROAMING_NUMBER service indication.

If the call handling process in the VLR returns a Provide Roaming Number ack, the MAP process constructs a MAP_PROVIDE_ROAMING_NUMBER service response containing the roaming number contained in the Send Routeing Info ack, constructs a MAP_CLOSE service request, sends them to the HLR and returns to the idle state.

Earlier version MAP dialogue with the HLR

If the macro Receive_Open_Ind takes the Vr exit, the the VLR performs the earlier version MAP dialogue as specified in [51] and the process returns to the idle state.

Failure of dialogue opening with the HLR

If the macro Receive_Open_Ind takes the Error exit, the MAP process returns to the idle state.

If the MAP provider sends a MAP_P_ABORT while the MAP process is waiting for a service indication, the MAP process returns to the idle state.

If the MAP provider sends a MAP_NOTICE while the MAP process is waiting for a service indication, the MAP process sends a MAP_CLOSE request to terminate the dialogue and returns to the idle state.

Negative response from VLR call handling process

If the call handling process in the HLR returns a negative response, the MAP process constructs a MAP_PROVIDE_ROAMING_NUMBER service response containing the appropriate error, constructs a MAP_CLOSE service request, sends them to the HLR and returns to the idle state.

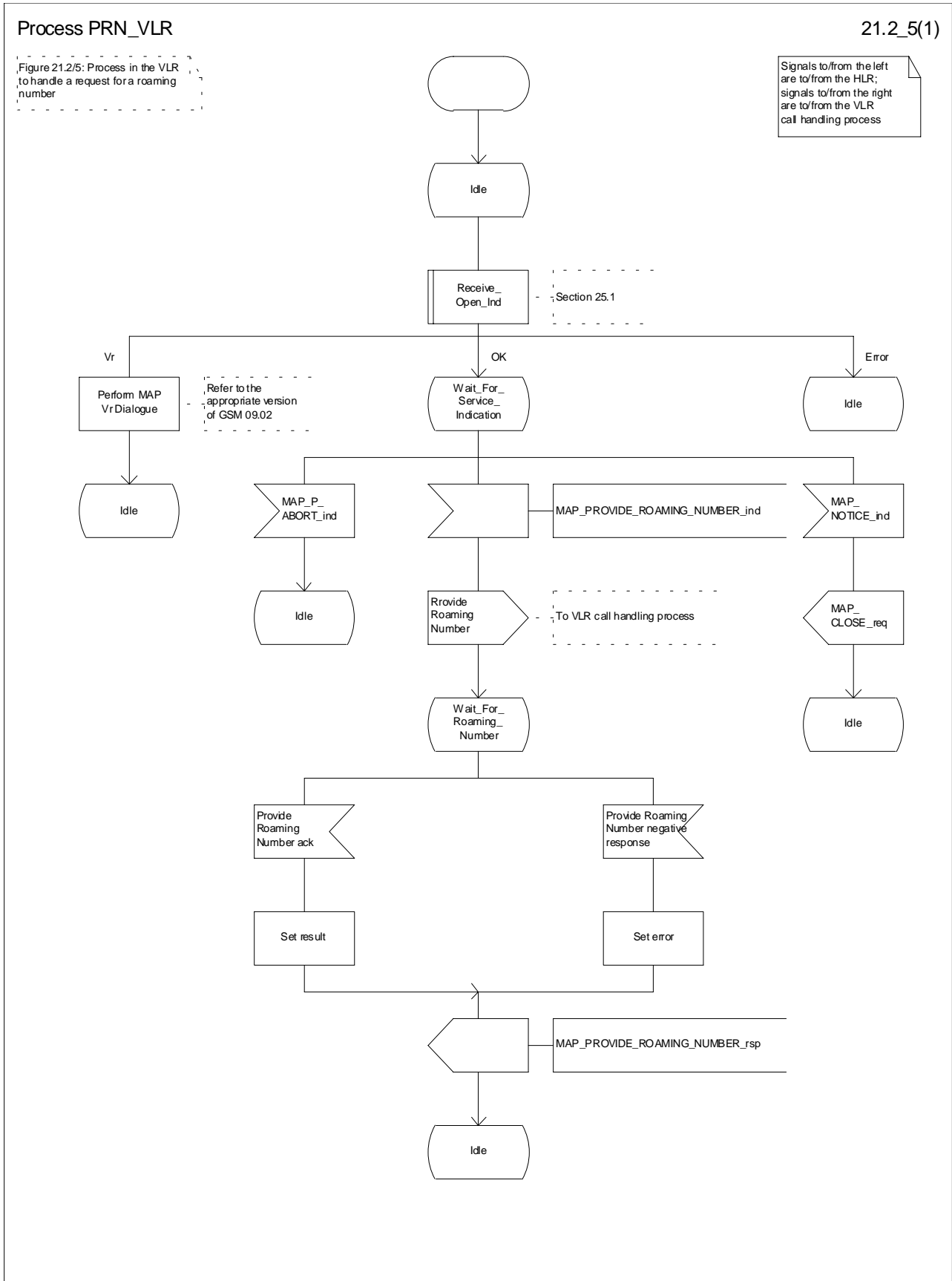


Figure 21.2/5: Process PRN_VLR

21.2.5 Process in the VLR to restore subscriber data

The MAP process in the HLR to restore subscriber data is shown in figure 21.2/6. The MAP process invokes macros not defined in this subclause; the definitions of these macros can be found as follows:

Receive_Open_Cnf	see subclause 25.1.2;
Check_Confirmation	see subclause 25.2.2;
Insert_Subscriber_Data_VLR	see subclause 25.7.1;
Activate_Tracing_VLR	see subclause 25.9.3.

Successful outcome

When the MAP process receives a Restore Data request from the data restoration process in the VLR, it requests a dialogue with the HLR whose identity is contained in the Restore Data request by sending a MAP_OPEN service request, requests data restoration using a MAP_RESTORE_DATA service request and invokes the macro Receive_Open_Cnf to wait for the response to the dialogue opening request. If the dialogue opening is successful, the MAP process waits for a response from the HLR.

The VLR may receive a MAP_INSERT_SUBSCRIBER_DATA service indication from the HLR; this is handled by the macro Insert_Subscriber_Data_VLR as described in subclause 25.7.1, and the MAP process waits for a further response from the HLR.

The VLR may receive a MAP_ACTIVATE_TRACE_MODE service indication from the HLR; this is handled by the macro Activate_Tracing_VLR as described in subclause 25.9.3, and the MAP process waits for a further response from the HLR.

If the MAP process receives a MAP_RESTORE_DATA service confirm, it invokes the macro Check_Confirmation to check the content of the confirm.

If the Check_Confirmation macro takes the OK exit, the MAP process sends a Restore Data ack containing the information received from the HLR to the data restoration process in the VLR and returns to the idle state.

Error in MAP_RESTORE_DATA confirm

If the MAP_RESTORE_DATA service confirm contains a user error or a provider error, or the macro Check_Confirmation indicates that there is a data error, the MAP process sends a Restore Data negative response indicating the type of error to the call handling process in the HLR, and returns to the idle state.

Earlier version MAP dialogue with the HLR

If the macro Receive_Open_Cnf takes the Vr exit, the VLR performs the earlier MAP version dialogue as specified in [51] and the process terminates.

Dialogue opening failure

If the macro Receive_Open_Cnf indicates that the dialogue with the HLR could not be opened, the MAP process sends a negative response indicating system failure to the data restoration process in the GMSC and returns to the idle state.

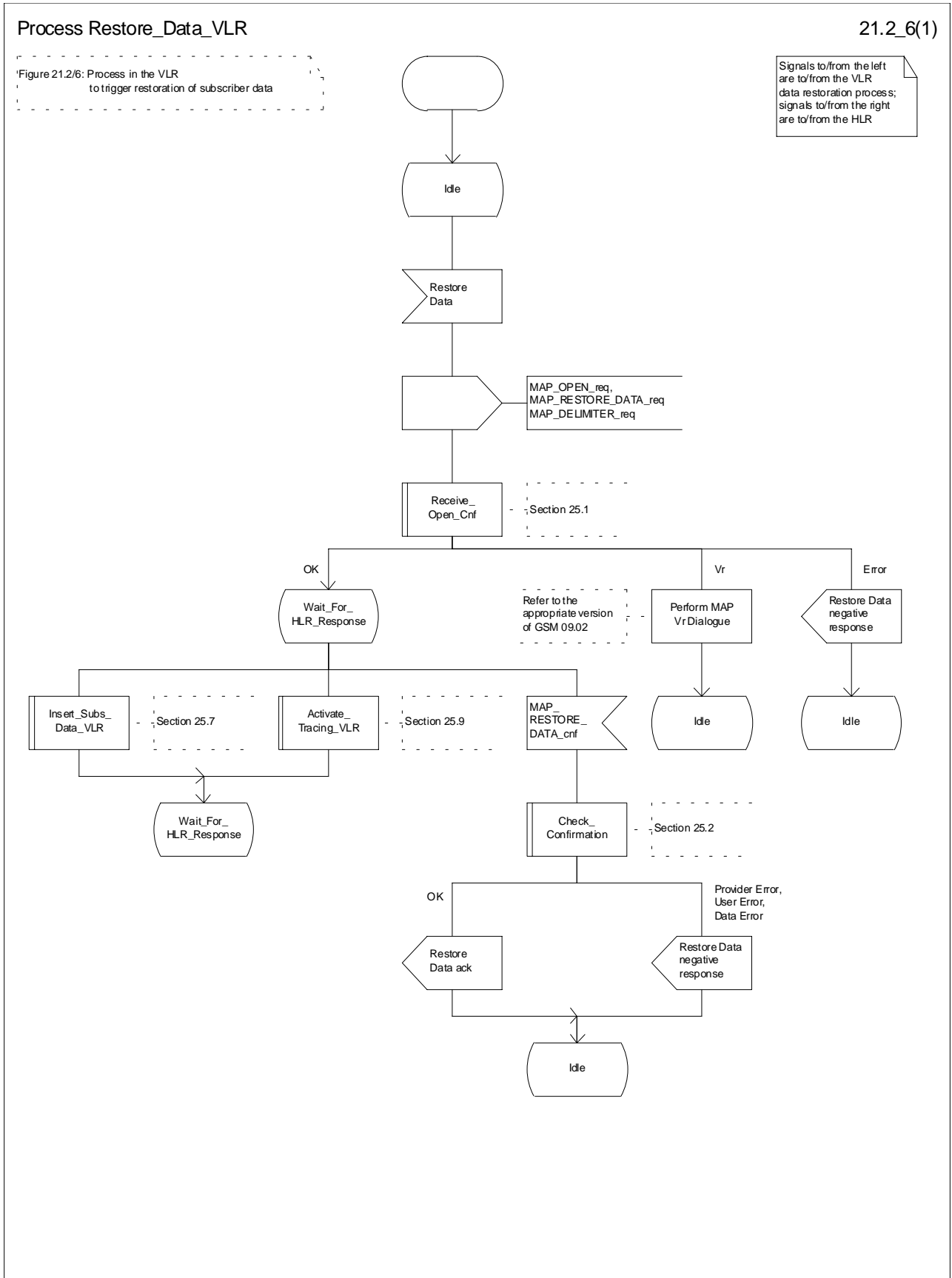


Figure 21.2/6: Process Restore_Data_VLR

21.2.6 Process in the VLR to provide subscriber information

The MAP process in the VLR to provide subscriber information for a mobile terminating call subject to CAMEL invocation is shown in figure 21.2/6. The MAP process invokes a macro not defined in this subclause; the definition of this macro can be found as follows:

Receive_Open_Ind see subclause 25.1.1;

Successful outcome

When the MAP process receives a MAP_OPEN indication with the application context subscriberInfoEnquiry, it checks it by invoking the macro Receive_Open_Ind.

If the macro takes the OK exit, the MAP process waits for a service indication.

If a MAP_PROVIDE_SUBSCRIBER_INFO service indication is received, the MAP process sends a Provide Subscriber Info request to the subscriber information request process in the VLR, and waits for a response. The Provide Subscriber Info request contains the parameters received in the MAP_PROVIDE_SUBSCRIBER_INFO service indication.

If the subscriber information request process in the VLR returns a Provide Subscriber Info ack, the MAP process constructs a MAP_PROVIDE_SUBSCRIBER_INFO service response containing the information contained in the Provide Subscriber Info ack, constructs a MAP_CLOSE service request, sends them to the HLR and returns to the idle state.

Failure of dialogue opening with the HLR

If the macro Receive_Open_Ind takes the Vr exit or the Error exit, the MAP process returns to the idle state.

If the MAP provider sends a MAP_P_ABORT while the MAP process is waiting for a service indication, the MAP process returns to the idle state.

If the MAP provider sends a MAP_NOTICE while the MAP process is waiting for a service indication, the MAP process sends a MAP_CLOSE request to terminate the dialogue and returns to the idle state.

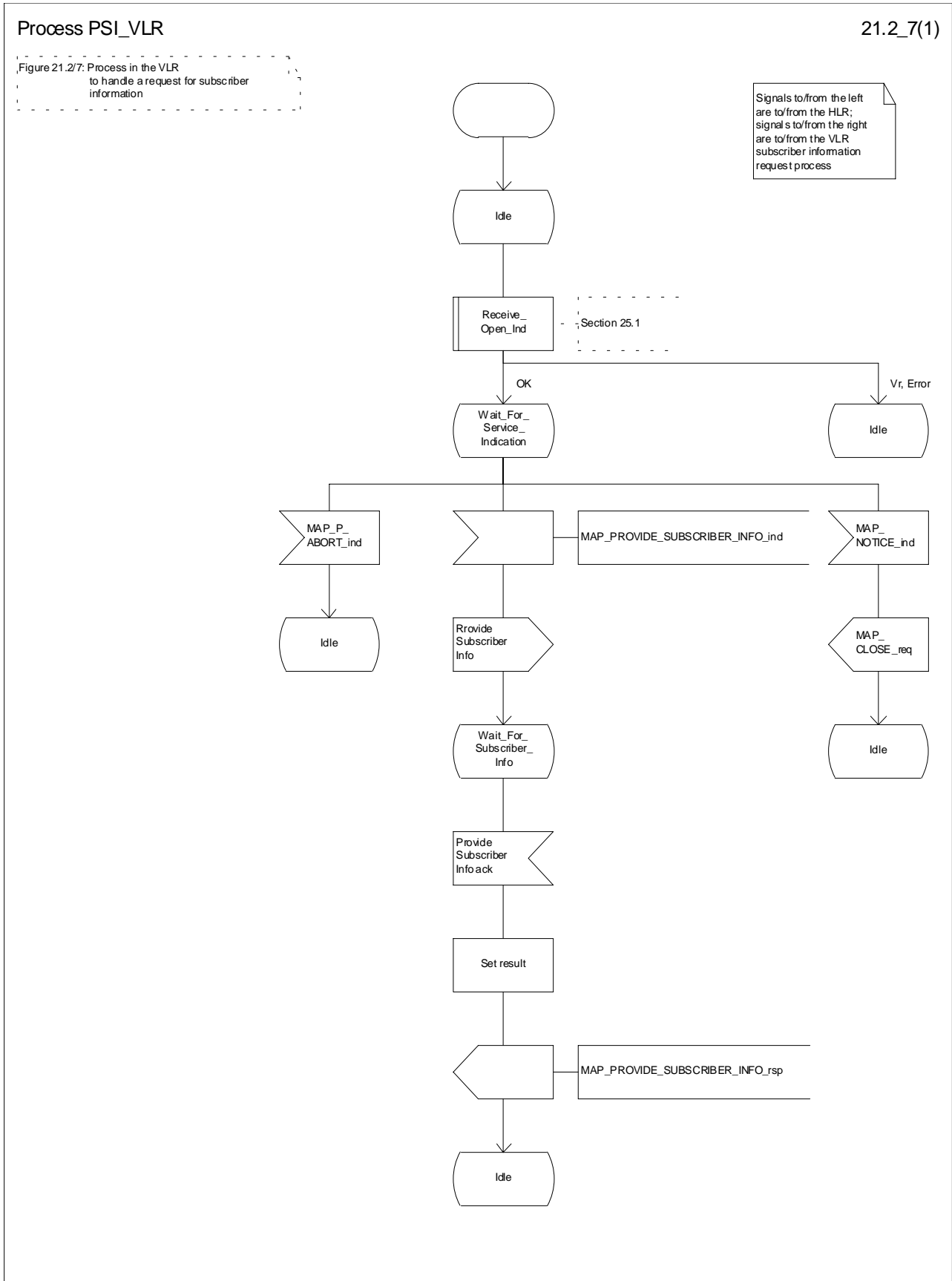


Figure 21.2/7: Process PSI_VLR

21.2.7 Process in the HLR for Any Time Interrogation

The message flows for successful retrieval of subscriber information related to an any time interrogation from the CAMEL server are shown in figure 21.2/8.

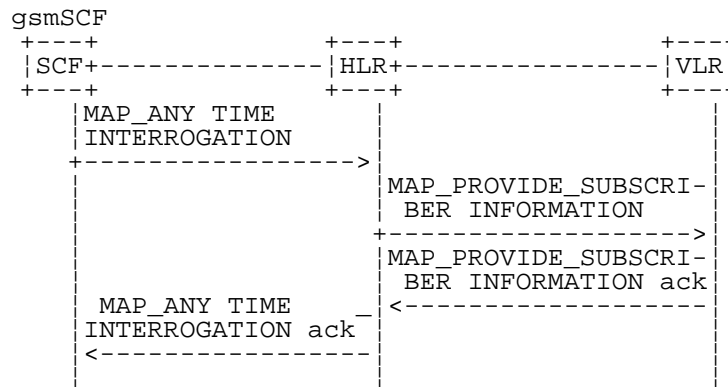


Figure 21.2/8: Message flow for any time interrogation

The following MAP services are used to retrieve routing information:

MAP_ANY_TIME_INTERROGATION see subclause 8.11.1;

MAP_PROVIDE_SUBSCRIBER_INFO see subclause 8.11.2;

21.2.7.1 Process in the gsmSCF

Out of the scope of the MAP specification.

21.2.7.2 Process in the HLR

The MAP process in the HLR to provide subscriber information in response to an interrogation from the CAMEL server is shown in figure 21.2/8. The MAP process invokes macros not defined in this subclause; the definitions of these macros can be found as follows:

Receive_Open_Ind see subclause 25.1.1;

Receive_Open_Cnf see subclause 25.1.2;

Check_Confirmation see subclause 25.2.2.

Successful outcome

When the MAP process receives a MAP_OPEN indication with the application context anyTimeInterrogationEnquiry, it checks it by invoking the macro Receive_Open_Ind.

If the macro takes the OK exit, the MAP process waits for a service indication.

If a MAP_ANY_TIME_INTERROGATION service indication is received, the MAP process sends an Any Time Interrogation request to the call handling process in the HLR (described in 3G TS 23.078), and waits for a response. The Any Time Interrogation request contains the parameters received in the MAP_ANY_TIME_INTERROGATION service indication.

If the call handling process in the HLR returns an Any Time Interrogation response, the MAP process constructs a MAP_ANY_TIME_INTERROGATION service response containing the subscriber information contained in the Any Time Interrogation response, constructs a MAP_CLOSE service request, sends them to the CAMEL server and returns to the idle state.

If the call handling process in the HLR returns a Provide Subscriber Info request, the MAP process requests a dialogue with the VLR whose identity is contained in the Provide Subscriber Info request by sending a MAP_OPEN service

request, requests the subscriber status using a MAP_PROVIDE_SUBSCRIBER_INFO service request, and invokes the macro Receive_Open_Cnf to wait for the response to the dialogue opening request.

If the macro takes the OK exit, the MAP process waits for the response from the VLR.

If the MAP process receives a MAP_PROVIDE_SUBSCRIBER_INFO service confirm, it invokes the macro Check_Confirmation to check the content of the confirm.

If the Check_Confirmation macro takes the OK exit, the MAP process sends a Provide Subscriber Info ack containing the information received in the MAP_PROVIDE_SUBSCRIBER_INFO service confirm to the call handling process in the HLR, and waits for a response. The handling of the response from the call handling process in the HLR is described above.

If the MAP_PROVIDE_SUBSCRIBER_INFO service confirm contains a provider error or a data error, the MAP process sends a Provide Subscriber Info negative response indicating the type of error to the call handling process in the HLR, and waits for a response. The handling of the response from the call handling process in the HLR is described above.

NOTE: The 'User Error' exit from the macro Check_Confirmation is shown for formal completeness; the MAP_PROVIDE_SUBSCRIBER_INFO_cnf primitive cannot contain a user error.

Negative response from HLR call handling process

If the call handling process in the HLR returns a negative response, either before or after a dialogue with the VLR to obtain subscriber information, the MAP process constructs a MAP_ANY_TIME_INTERROGATION service response containing the appropriate error, constructs a MAP_CLOSE service request, sends them to the CAMEL server and returns to the idle state.

Failure of Provide Subscriber Info dialogue with the VLR

If the Receive_Open_Cnf macro takes the Vr exit or the Error exit after the MAP process has requested opening of a Provide Subscriber Info dialogue with the VLR, the MAP process sends a Provide Subscriber Info negative response indicating system failure to the call handling process in the HLR, and waits for a response. The handling of the response from the call handling process in the HLR is described above.

Failure of dialogue opening with the CAMEL server

If the macro Receive_Open_Ind takes the Vr or Error exit, the MAP process returns to the idle state.

If the MAP provider sends a MAP_P_ABORT while the MAP process is waiting for a service indication, the MAP process returns to the idle state.

If the MAP provider sends a MAP_NOTICE while the MAP process is waiting for a service indication, the MAP process sends a MAP_CLOSE request to terminate the dialogue and returns to the idle state.

Process ATI_HLR

21.2_9.1(2)

Figure 21.2/9: Process in the HLR to respond to a request for any time interrogation

Signals to/from the left are to/from the gsmSCF; signals to/from the right are to/from the VLR unless specified otherwise

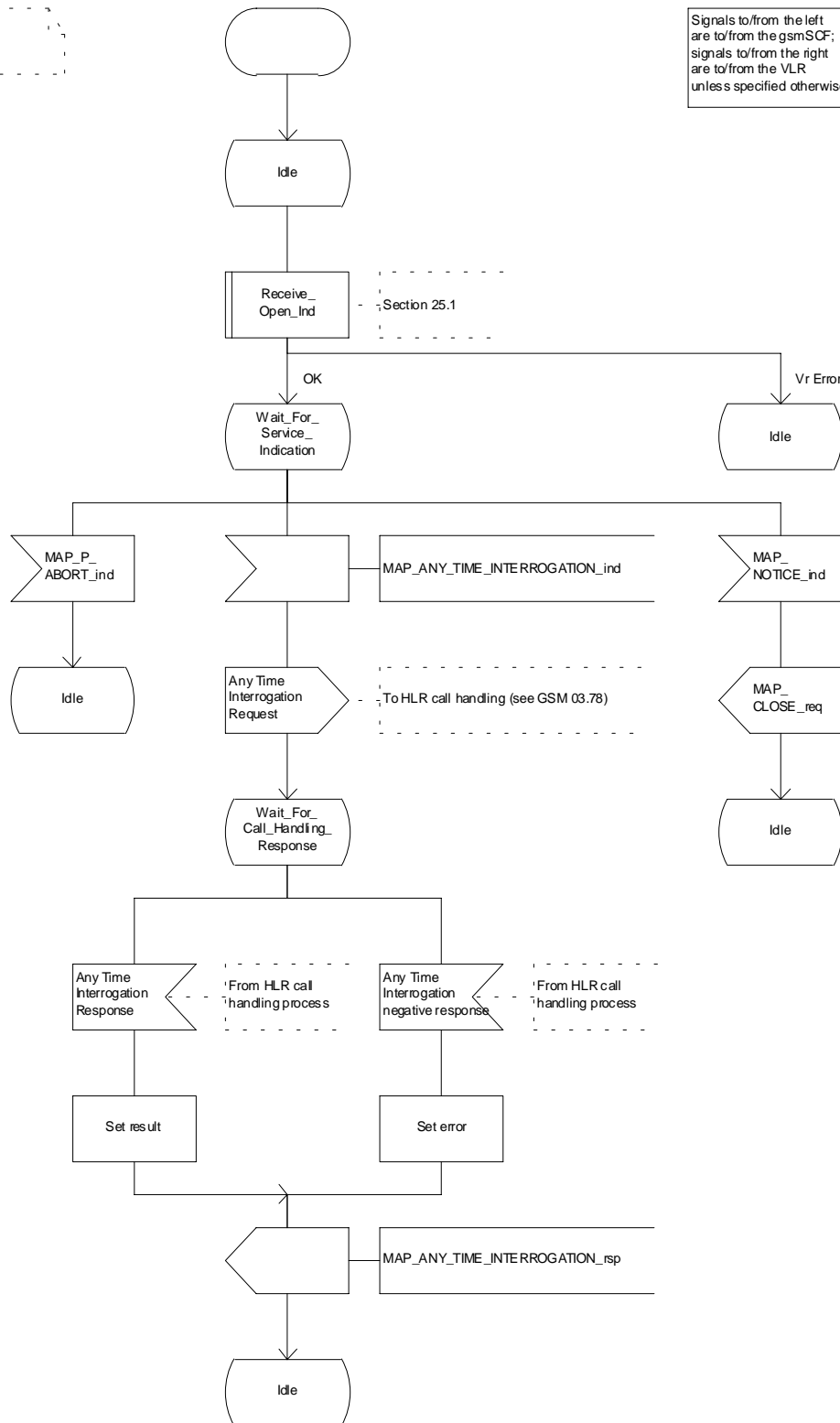


Figure 21.2/9 (sheet 1 of 2): Process ATI_HLR (New)

Process ATI_HLR

21.2_9.2(2)

Figure 21.2/9: Process in the HLR to respond to a request for any time interrogation

Signals to/from the left are to/from the gsmSCF; signals to/from the right are to/from the VLR unless specified otherwise

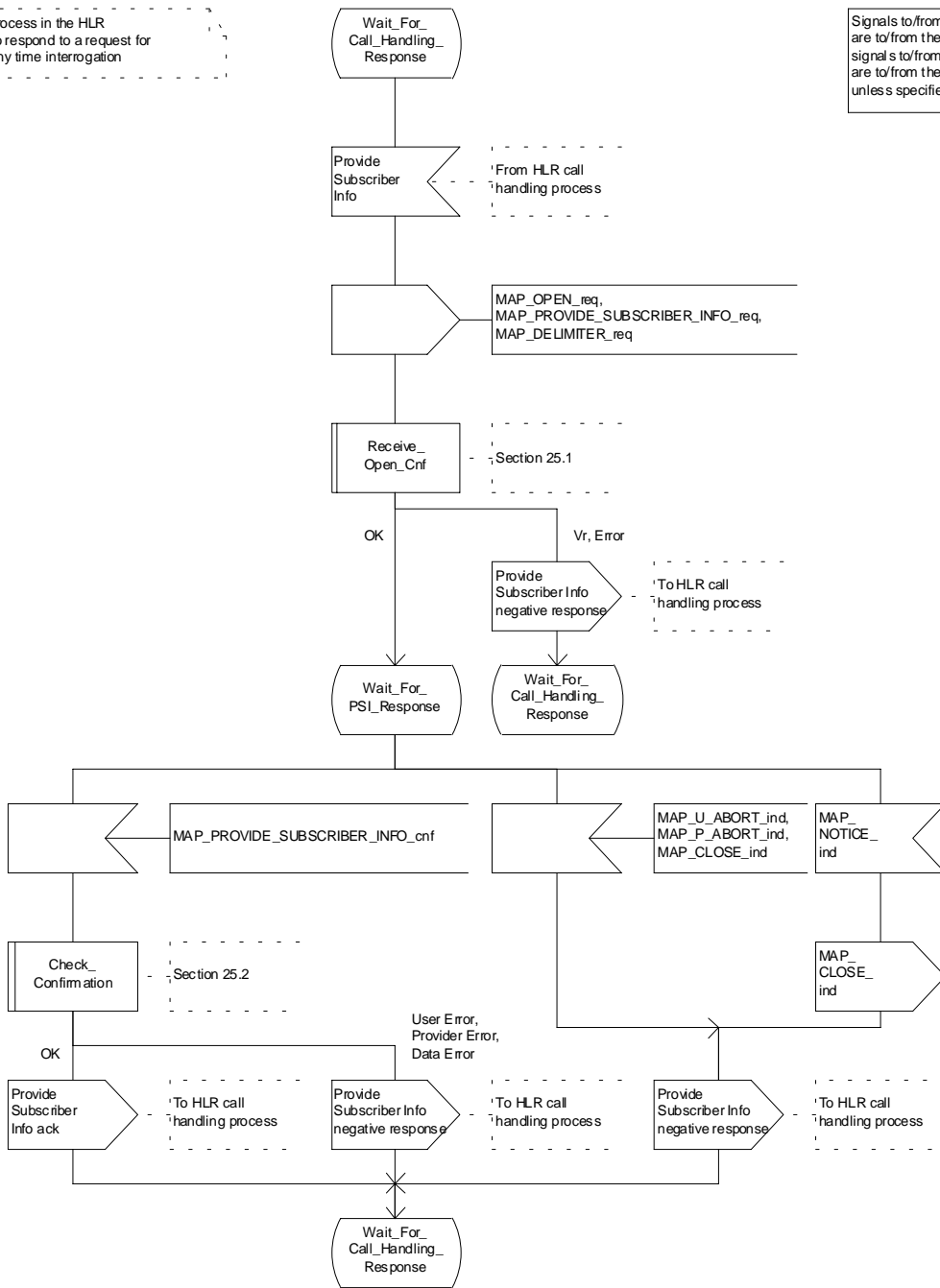
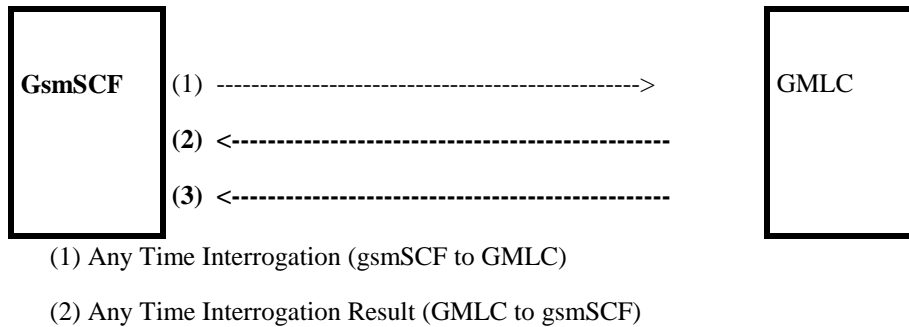


Figure 21.2/9 (sheet 2 of 2): Process ATI_HLR (New)

21.2.8 Process in the GMLC for Any Time Interrogation

The message flows for successful retrieval of subscriber information related to an any time interrogation from the CAMEL server are shown in figure 21.2.8/1.



(3) Any Time Interrogation Error (GMLC to gsmSCF) The following MAP services are used to retrieve routing information:

MAP_ANY_TIME_INTERROGATION see subclause 8.11.1;

In addition, the GMLC may use Location Services specific MAP Services.

21.2.8.1 Process in the gsmSCF

The process in the gsmSCF to request location information from the GMLC is shown in figure 21.2.8/2.

The process is started with internal signal Request_Subscriber_Info_GMLC. This signal is sent by the Service Logic in the gsmSCF.

The process responds with 'Request_Subscriber_Info_GMLC positive response' or 'Request_Subscriber_Info_GMLC negative response'.

21.2.8.2 Process in the GMLC

The MAP process in the GMLC to provide location information in response to a request from the gsmSCF is shown in figure 21.2.8/3.

Successful outcome

When the GMLC has successfully received the MAP Any_Time_Interrogation MAP Message, it will send an internal signal to the Location Service process in the GMLC to obtain the subscriber's Location Information. The result received from that process is sent back to the gsmSCF, in the Any_Time_Interrogation Result MAP Message.

Unsuccessful outcome

In the case of a Provider Error received, the process will terminate.

When a User error is received from the Location Services process in the GMLC, then a User Abort is sent to the gsmSCF.

If a negative response is received from the Location Services process in the GMLC, then this response is forwarded to the gsmSCF.

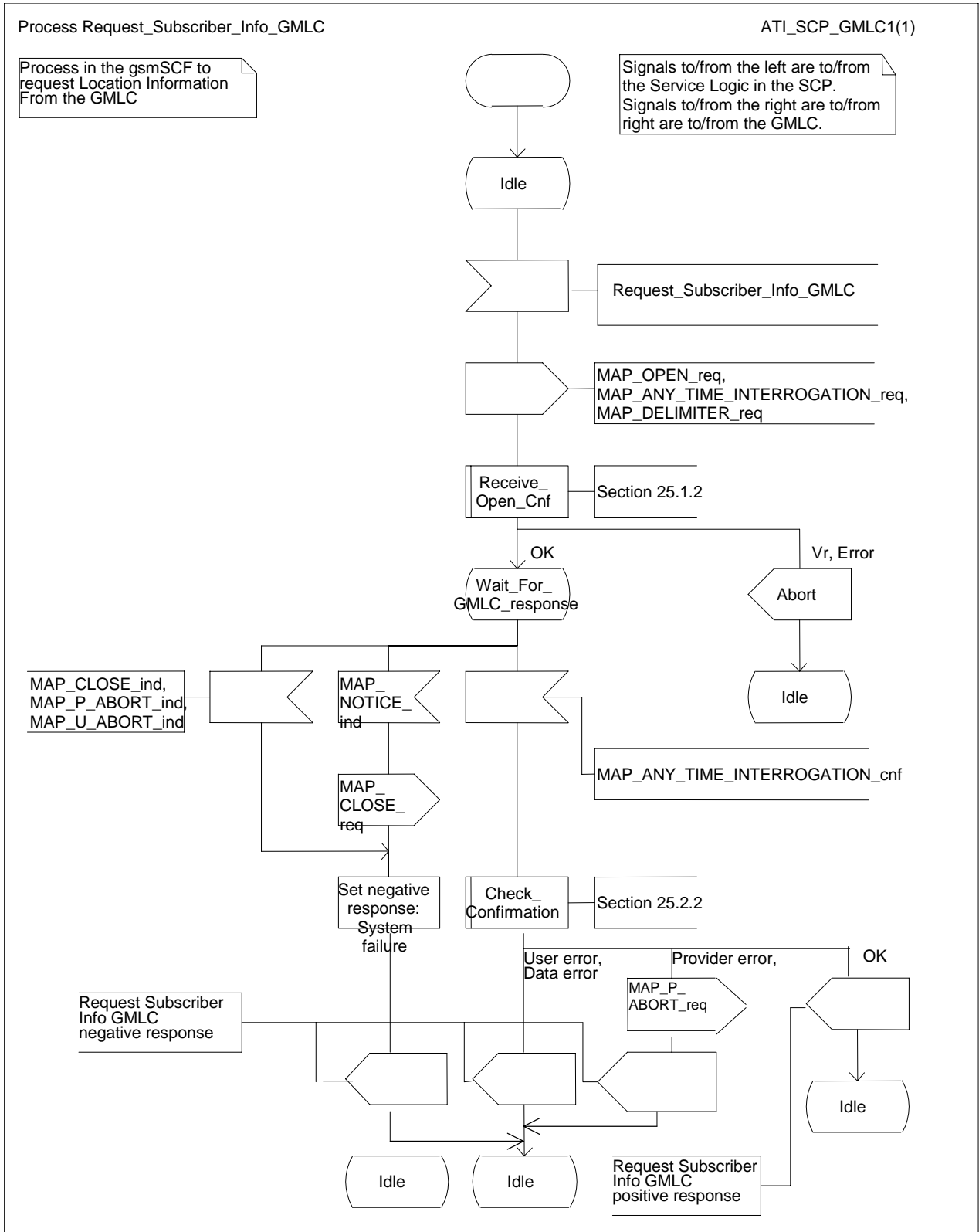


Figure 21.2.8/2: Process Request_Subscriber_Info_GMLC

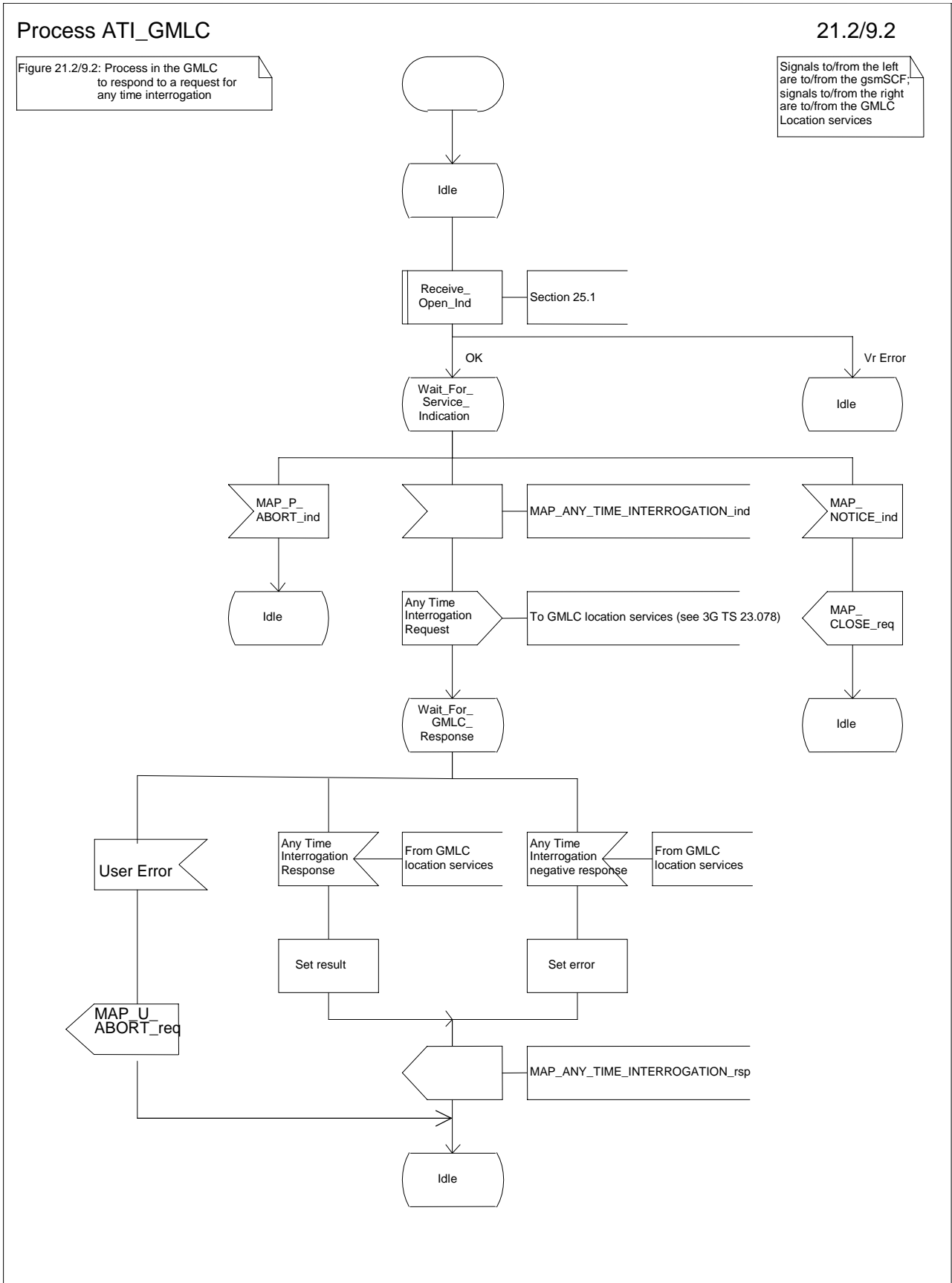
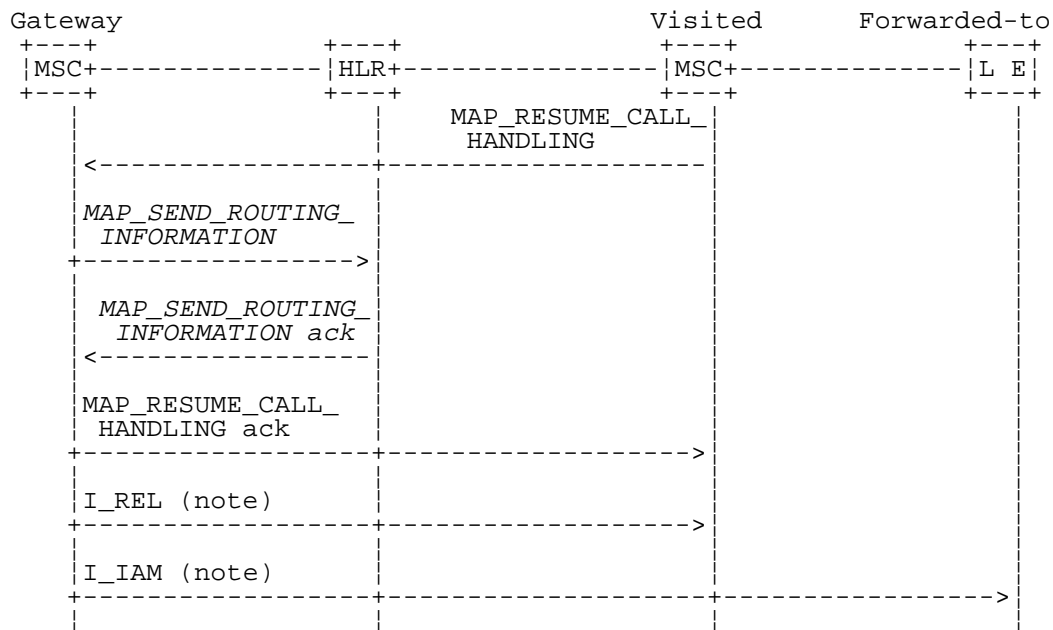


Figure 21.2.8/3: Process ATI_GMLC

21.3 Transfer of call handling

21.3.1 General

The message flow for successful transfer of call handling to forward a call is shown in figure 21.3/1.



NOTES:

xxx = Optional Procedure

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 & ETSI specification:

Q.721-725 - Telephone User Part (TUP);

ETS 300 356-1 - Integrated Services Digital Network (ISDN); Signalling System No.7; ISDN User Part (ISUP) version 2 for the international interface; Part 1: Basic services.

Figure 21.3/1: Message flow for transfer of call handling

If the HLR indicated in the response to the original request for routing information that forwarding interrogation is required, the GMSC executes the Send Routing Information procedure with the HLR to obtain forwarding information; otherwise the GMSC uses the forwarding data which were sent in the `MAP_RESUME_CALL_HANDLING` req/ind.

21.3.2 Process in the VMSC

The MAP process in the VMSC to retrieve routing information for a mobile terminating call is shown in figure 21.3/2. The MAP process invokes macros not defined in this subclause; the definitions of these macros can be found as follows:

Receive_Open_Cnf see subclause 25.1.2;

Check_Confirmation see subclause 25.2.2.

Successful Outcome

When the MAP process receives a Resume Call Handling request from the call handling process in the VMSC, it requests a dialogue with the GMSC whose identity is contained in the Resume Call Handling request by sending a

MAP_OPEN service request, requests routing information using a MAP_RESUME_CALL_HANDLING service request and invokes the macro Receive_Open_Cnf to wait for the response to the dialogue opening request. If the dialogue opening is successful, the MAP process waits for a response from the GMSC. VMSC shall not send any duplicate data to the GMSC.

If the VMSC notices after receiving a Resume Call Handling request that the segmentation is needed the VMSC does not set the "All Information Sent" indicator. Otherwise the indicator is set and the process returns to the Wait For GMSC Response state.

If the MAP process receives a MAP_RESUME_CALL_HANDLING service confirm from the GMSC, the MAP process invokes the macro Check_Confirmation to check the content of the confirm.

If the macro Check_Confirmation takes the OK exit, the MAP process checks if the "All Information Sent" indicator is set. If it is set the MAP process sends a Resume Call Handling ack to the call handling process in the VMSC and returns to the idle state. If the "All Information Sent" indicator is not set the MAP process checks if the further segmentation is needed. If segmentation is needed the VMSC does not set the indicator and sends MAP_RESUME_CALL_HANDLING service request to the GMSC. Otherwise the indicator is set and the MAP_RESUME_CALL_HANDLING service request is sent to the GMSC.

Dialogue opening failure

If the macro Receive_Open_Cnf indicates that the dialogue with the GMSC could not be opened or that the dialogue can be opened only at an earlier version, the MAP process sends an Resume Call Handling negative response indicating system failure to the call handling process in the VMSC and returns to the idle state.

Error in MAP_RESUME_CALL_HANDLING confirm

If the MAP_RESUME_CALL_HANDLING service confirm contains a user error or a provider error, the MAP process sends a Resume Call Handling negative response to the call handling process in the VMSC and returns to the idle state.

NOTE: the 'Data Error' exit from the macro Check_Confirmation is shown for formal completeness; the result is empty, so the MAP_PROVIDE_SUBSCRIBER_INFO_cnf primitive cannot contain a data error.]

Abort of GMSC dialogue

After the dialogue with the GMSC has been established, the MAP service provider may abort the dialogue by issuing a MAP_P_ABORT indication, or the GMSC may send a MAP_CLOSE indication. In either of these cases, the MAP process sends a Resume Call Handling negative response to the call handling process in the GMSC and returns to the idle state.

If the MAP provider indicates a protocol problem by sending a MAP_NOTICE indication, the MAP process closes the dialogue with the GMSC, sends a Resume Call Handling negative response indicating system failure to the call handling process in the VMSC and returns to the idle state.

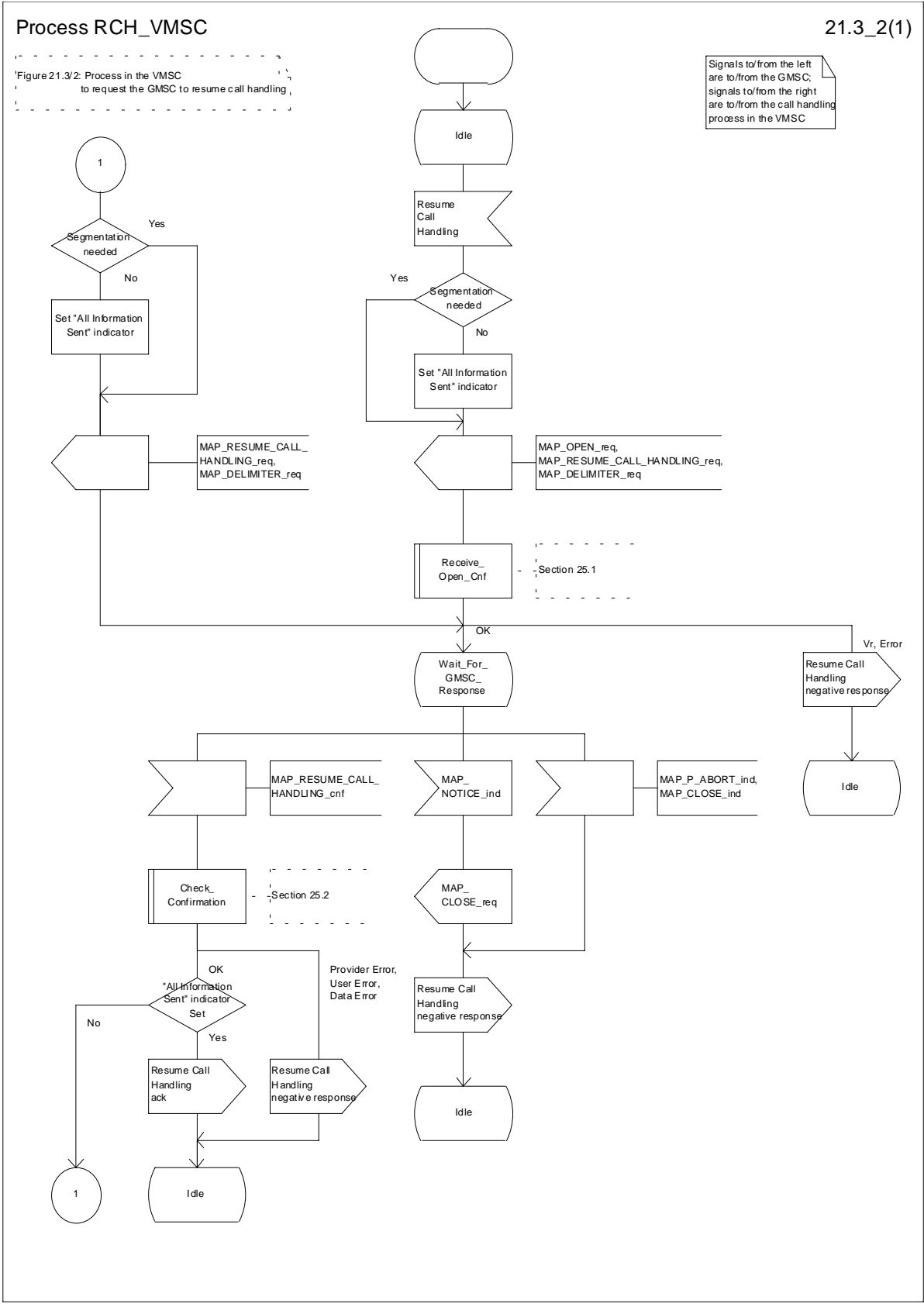


Figure 21.3/2: Process RCH_VMSC

21.3.3 Process in the GMSC

The MAP process in the GMSC to handle a request for the GMSC to resume call handling is shown in figure 21.3/3. The MAP process invokes a macro not defined in this subclause; the definition of this macro can be found as follows:

Receive_Open_Ind see subclause 25.1.1;

Successful outcome

When the MAP process receives a MAP_OPEN indication with the application context callControlTransfer, it checks it by invoking the macro Receive_Open_Ind.

If the macro takes the OK exit, the MAP process waits for a service indication.

If a MAP_RESUME_CALL_HANDLING service indication is received, the MAP process checks if the “All Information Sent” indicator is set and if so it sends a Resume Call Handling request including all the stored data to the call handling process in the GMSC, and waits for a response. The Resume Call Handling request contains the parameters received in the MAP_RESUME_CALL_HANDLING service indication. If the “All Information Sent” indicator is not set, the received data is stored and the MAP process constructs an empty MAP_RESUME_CALL_HANDLING service response, sends it to the VMSC and returns to the Wait For Service Indication state.

If the call handling process in the GMSC returns a Resume Call Handling ack, the MAP process constructs a MAP_RESUME_CALL_HANDLING service response, constructs a MAP_CLOSE service request, sends them to the VMSC and returns to the idle state.

Failure of dialogue opening with the VMSC

If the macro Receive_Open_Ind takes the Vr exit or the Error exit, the MAP process returns to the idle state.

If the MAP provider sends a MAP_P_ABORT while the MAP process is waiting for a service indication, the MAP process returns to the idle state.

If the MAP provider sends a MAP_NOTICE while the MAP process is waiting for a service indication, the MAP process sends a MAP_CLOSE request to terminate the dialogue and returns to the idle state.

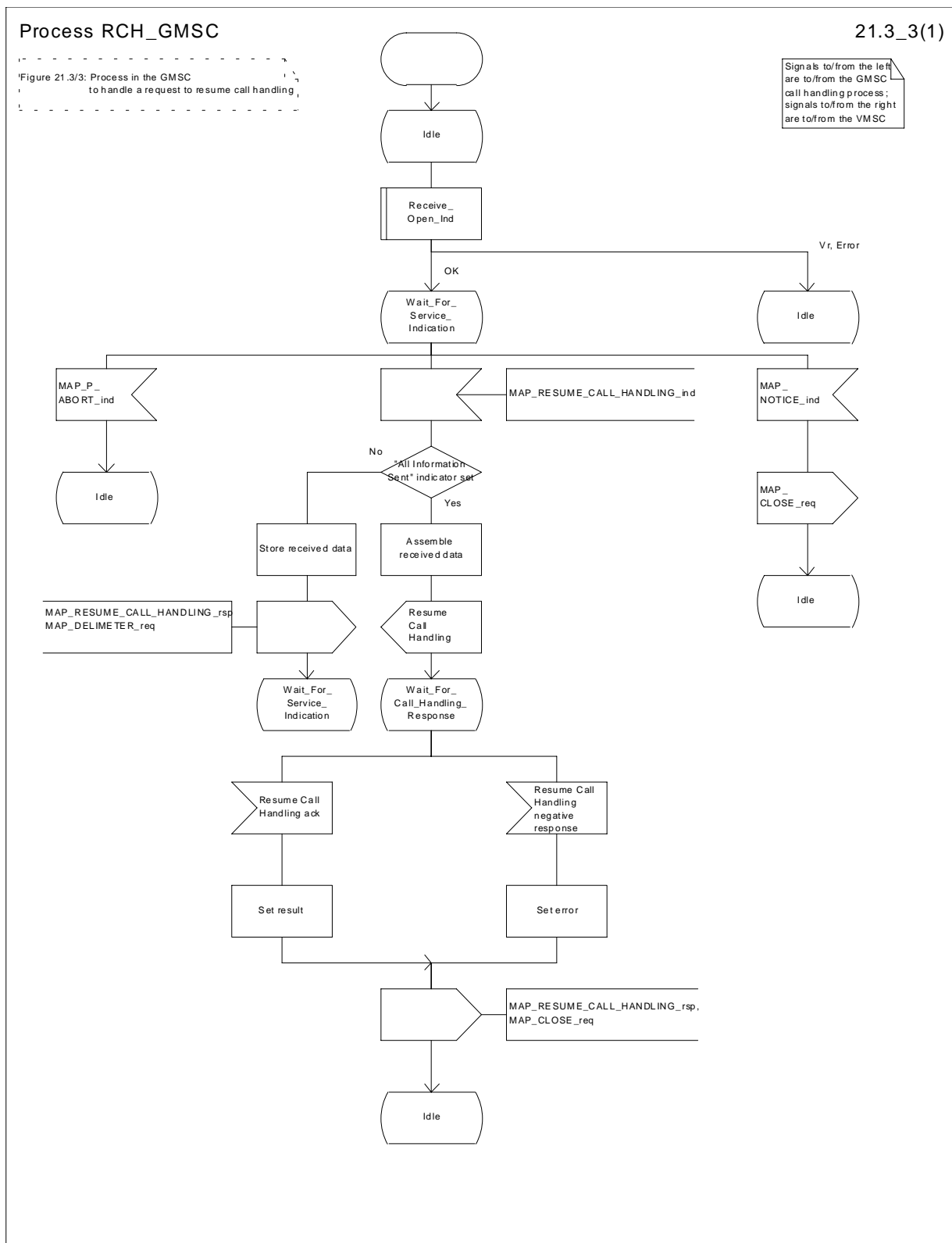
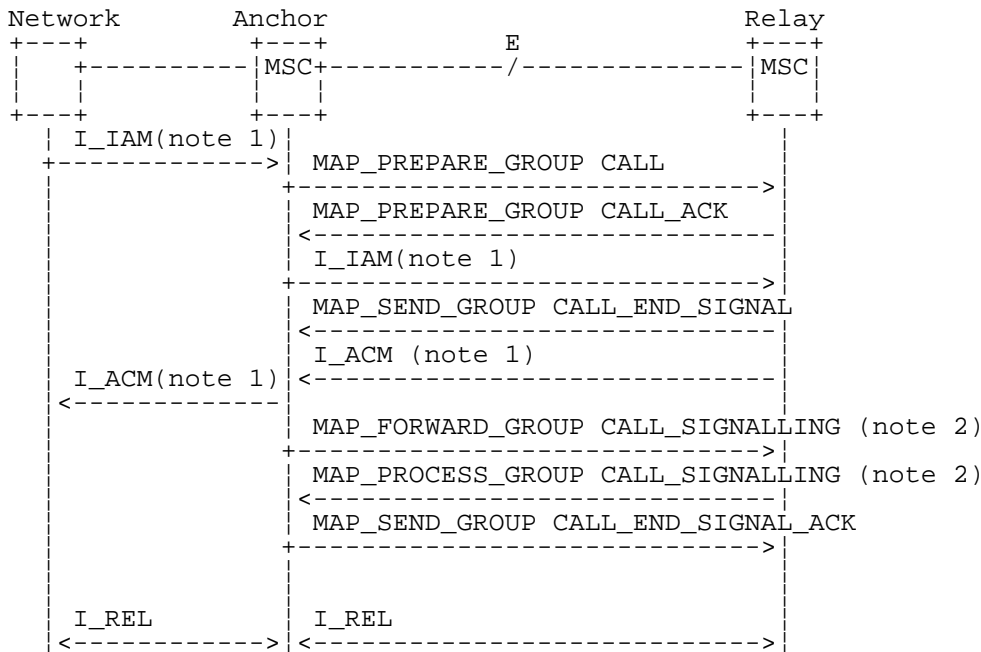


Figure 21.3/3: Process RCH_GMSC

21.4 Inter MSC Group Call Procedures

21.4.1 General

The message flows for successful inter MSC group call / broadcast call setup is shown in figure 21.4/1.



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 ITU-T Recommendations and ETSI specification:

Q.721-725 - Telephone User Part (TUP);

ETS 300 356-1 - Integrated Services Digital Network (ISDN); Signalling System No.7; ISDN User Part (ISUP) version 2 for the international interface; Part 1: Basic services.

NOTE 2: The MAP_FORWARD_GROUP_CALL_SIGNALLING and MAP_PROCESS_GROUP_CALL_SIGNALLING services are not applicable for voice broadcast calls.

Figure 21.4/1: Message flow for inter MSC group call / broadcast call

21.4.2 Process in the Anchor MSC

The MAP process in the Anchor MSC to retrieve and transfer information from / to the Relay MSC for VBS and VGCS calls is shown in figure 21.4/2. The MAP process invokes macros not defined in this subclause; the definitions of these macros can be found as follows:

Receive_Open_Cnf see subclause 25.1.2;

Check_Indication see subclause 25.2.1;

Check_Confirmation see subclause 25.2.2.

Successful Outcome

When the MAP process receives a Prepare Group Call request from the ASCII handling process in the anchor MSC, it requests a dialogue with the relay MSC whose identity is contained in the Prepare Group Call request by sending a MAP_OPEN service request, requests an Group Call number by using a MAP_PREPARE_GROUP_CALL service

request and invokes the macro `Receive_Open_Cnf` to wait for the response to the dialogue opening request. If the dialogue opening is successful, the MAP process waits for a response from the relay MSC.

If the MAP process receives a `MAP_PREPARE_GROUP_CALL` service confirm from the relay MSC, the MAP process invokes the macro `Check_Confirmation` to check the content of the confirm.

If the macro `Check_Confirmation` takes the OK exit, the MAP process sends a Prepare Group Call ack containing the Group Call number received from the relay MSC to the ASCI handling process in the anchor MSC and waits for completion of call setup in the relay MSC.

On receipt of a `MAP_SEND_GROUP_CALL_END_SIGNAL` service indication from the relay MSC the MAP process invokes the macro `Check_Indication` to check the content of the indication.

If the macro `Check_Indication` takes the OK exit, the MAP process sends a Send Group Call End Signal to the ASCI handling process in the anchor MSC and waits for uplink management signals. In this state the following events are processed:

- Reception of a Send Group Call End Signal ack from the ASCI handling process in the anchor MSC;
- Reception of a Forward Group Call Signalling request from the ASCI handling process in the anchor MSC;
- Reception of a `MAP_PROCESS_GROUP_CALL_SIGNALLING` service indication from the relay MSC.

On reception of a Send Group Call End Signal ack from the ASCI handling process in the anchor MSC, the MAP process constructs a `MAP_SEND_GROUP_CALL_END_SIGNAL` service response, constructs a `MAP_CLOSE` service request, sends them to the relay MSC and returns to the idle state.

On reception of a Forward Group Call Signalling request from the ASCI handling process in the anchor MSC, the MAP process constructs a `MAP_FORWARD_GROUP_CALL_SIGNALLING` service request, sends it to the relay MSC and returns to the uplink management state.

On reception of a `MAP_PROCESS_GROUP_CALL_SIGNALLING` service indication from the relay MSC, the MAP process invokes the macro `Check_Indication` to check the content of the indication.

If the macro `Check_Indication` takes the OK exit, the MAP process sends a Process Group Call Signalling to the ASCI handling process in the anchor MSC and returns to the uplink management state.

Dialogue opening failure

If the macro `Receive_Open_Cnf` indicates that the dialogue with the relay MSC could not be opened, the MAP process sends an Abort to the ASCI handling process and returns to the idle state.

Error in MAP_PREPARE_GROUP_CALL confirm

If the `MAP_PREPARE_GROUP_CALL` service confirm contains a user error or a provider error, or the macro `Check_Confirmation` indicates that there is a data error, the MAP process sends a Prepare Group Call negative response to the ASCI handling process in the anchor MSC, sends a `MAP_U_ABORT` request to the relay MSC and returns to the idle state.

Abort of MAP dialogue

After the dialogue with the relay MSC has been established, the MAP service provider may abort the dialogue by issuing a `MAP_P_ABORT` indication, or the relay MSC may send a `MAP_U_ABORT` indication or a `MAP_CLOSE` indication. In any of these cases, the MAP process sends an Abort to the ASCI handling process in the anchor MSC and returns to the idle state.

If the MAP provider indicates a protocol problem by sending a `MAP_NOTICE` indication, the MAP process closes the dialogue with the relay MSC, sends an Abort to the ASCI handling process in the anchor MSC and returns to the idle state.

Process ASCI_Anchor_MSC

21.4_2.1(4)

Figure 21.4/2: Process in the Anchor MSC for ASCI call handling

Signals to/from the left are to/from the A-MSC ASCI process; signals to/from the right are to/from the R-MSC

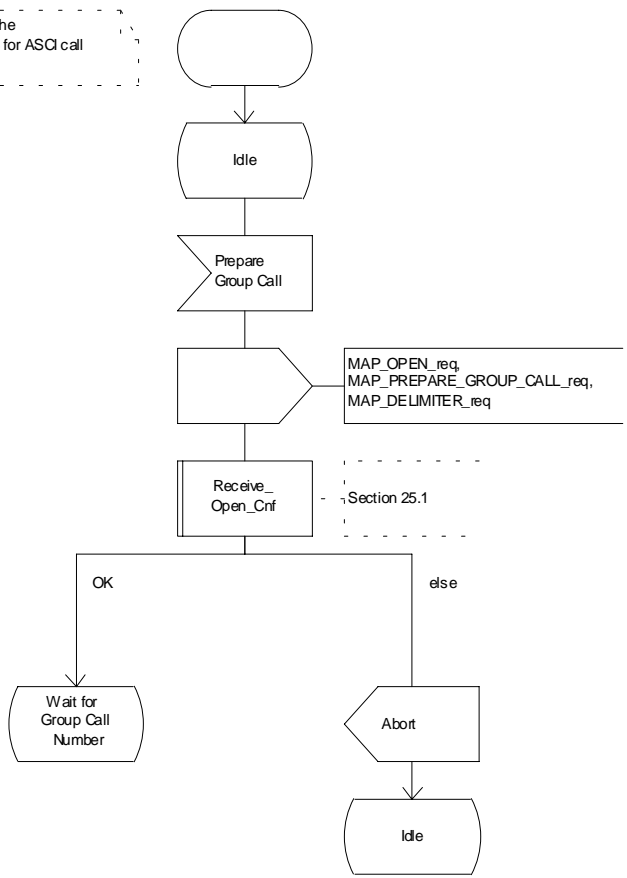


Figure 21.4/2 (sheet 1 of 4): Process ASCI_Anchor_MSC

Process ASCI_Anchor_MSC

21.4_2.2(4)

Figure 21.4/2: Process in the Anchor MSC for ASCI call handling

Signals to/from the left are to/from the A-MSC ASCI process; signals to/from the right are to/from the R-MSC

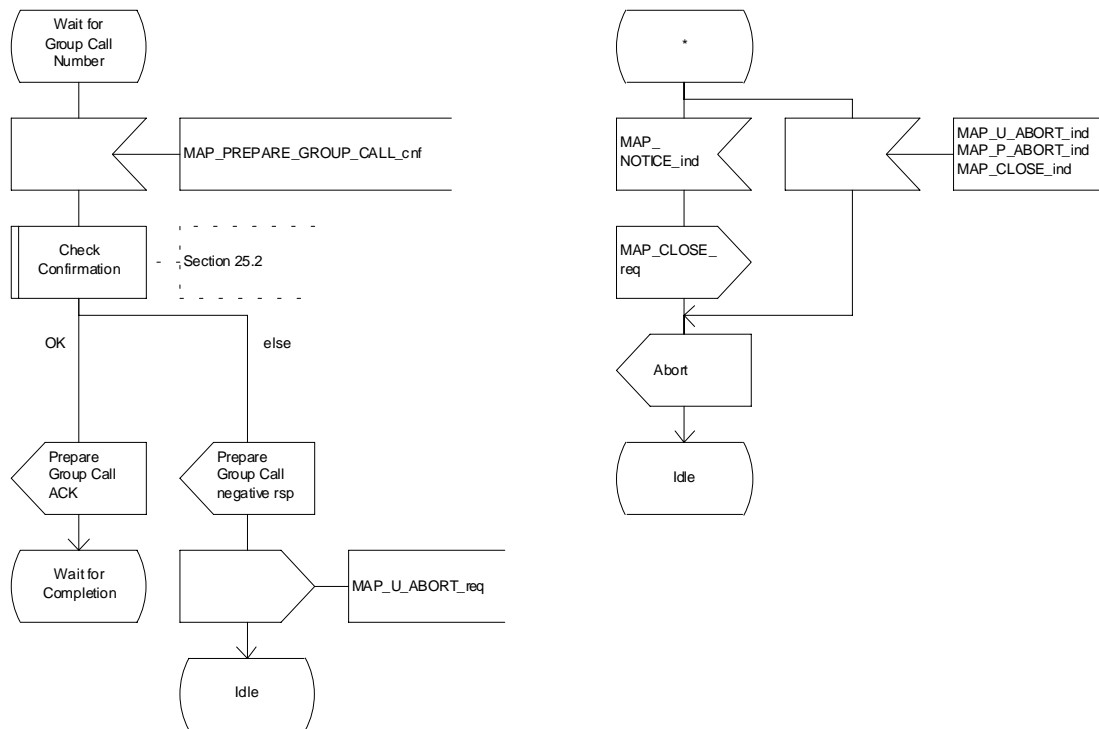


Figure 21.4/2 (sheet 2 of 4): Process ASCI_Anchor_MSC

Process ASCI_Anchor_MSC

21.4_2.3(4)

Figure 21.4/2: Process in the Anchor MSC for ASCI call handling

Signals to/from the left are to/from the A-MSC ASCI process; signals to/from the right are to/from the R-MSC

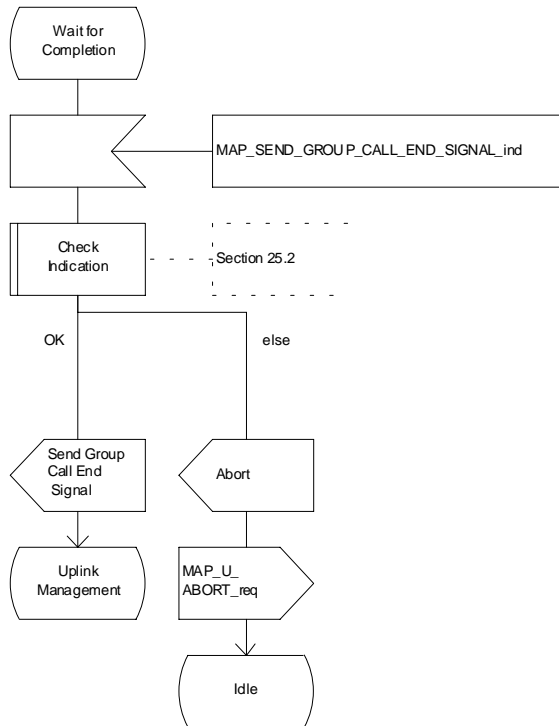


Figure 21.4/2 (sheet 3 of 4): Process ASCI_Anchor_MSC

Process ASCI_Anchor_MSC

21.4_2.4(4)

Figure 21.4/2: Process in the Anchor MSC for ASCI call handling

Signals to/from the left are to/from the A-MSC ASCI process; signals to/from the right are to/from the R-MSC

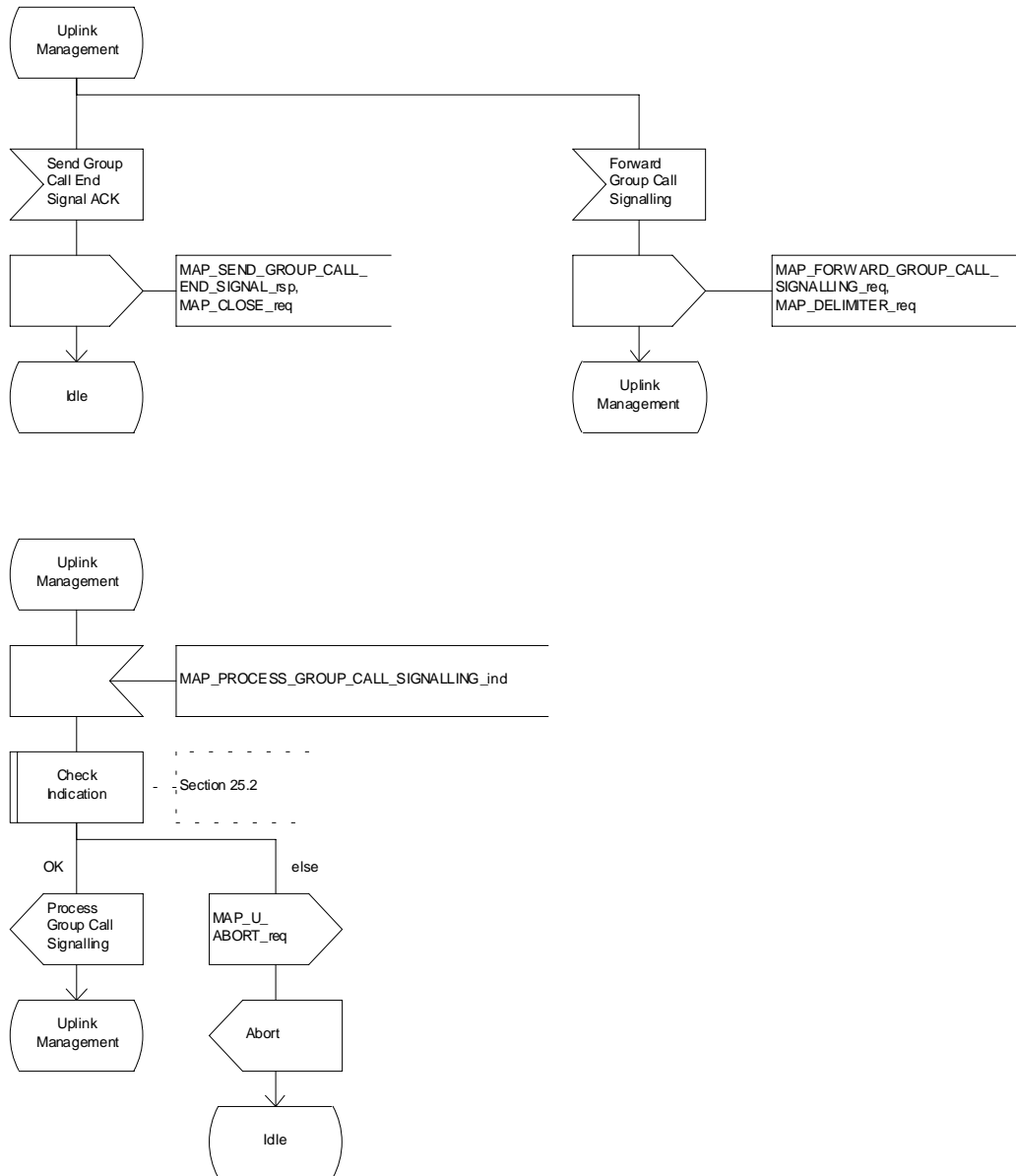


Figure 21.4/2 (sheet 4 of 4): Process ASCI_Anchor_MSC

21.4.3 Process in the Relay MSC

The MAP process in the Relay MSC to receive and transfer information from / to the Anchor MSC for VBS and VGCS calls is shown in figure 21.4/3. The MAP process invokes macros not defined in this subclause; the definitions of these macros can be found as follows:

Receive_Open_Ind	see subclause 25.1.2;
Check_Indication	see subclause 25.2.1.

Successful Outcome

When the MAP process receives a MAP_OPEN indication with the application context groupCallControl, it checks it by invoking the macro Receive_Open_Ind.

If the macro takes the OK exit, the MAP process waits for a service indication.

If a MAP_PREPARE_GROUP_CALL service indication is received, the MAP process invokes the macro Check_Indication.

If the macro takes the OK exit, the MAP process sends a Prepare Group Call request to the ASCI handling process in the relay MSC and waits for a response. The Prepare Group Call request contains the parameters received in the MAP_PREPARE_GROUP_CALL service indication.

If the ASCI handling process in the relay MSC returns a Prepare Group Call ack, the MAP process constructs a MAP_PREPARE_GROUP_CALL service response containing the information contained in the Prepare Group Call ack, constructs a MAP_DELIMITER service request, sends them to the anchor MSC and waits for the GROUP CALL END SIGNAL.

If the ASCI handling process in the relay MSC sends a Send Group Call End Signal request to the MAP process, the MAP process constructs a MAP_SEND_GROUP_CALL_END_SIGNAL service request containing the information contained in the SEND GROUP CALL End Signal request, constructs a MAP_DELIMITER service request, sends them to the anchor MSC and waits for uplink management signals. In this state the following events are processed:

- Reception of a MAP_SEND_GROUP_CALL_END_SIGNAL service confirmation from the anchor MSC;
- Reception of a MAP_FORWARD_GROUP_CALL_SIGNALLING service indication from the anchor MSC;
- Reception of a Process Group Call Signalling request from the ASCI handling process in the relay MSC.

On reception of a MAP_SEND_GROUP_CALL_END_SIGNAL service confirmation from the anchor MSC, the MAP process returns to the idle state.

On reception of a MAP_FORWARD_GROUP_CALL_SIGNALLING service indication from the anchor MSC, the MAP process invokes the macro Check Indication. If the macro takes the OK exit, the MAP process sends a Forward Group Call Signalling request to the ASCI handling process in the relay MSC and waits for further uplink management signals.

On reception of a Process Group Call Signalling request from the ASCI handling process in the relay MSC, the MAP process constructs a MAP_PROCESS_GROUP_CALL_SIGNALLING service request containing the information received in the Process Group Call Signalling request, constructs a MAP_DELIMITER service request, sends them to the anchor MSC and waits for further uplink management signals.

Failure of dialogue opening with the anchor MSC

If the macro Receive_Open_Ind takes the Error exit, the MAP process returns to the idle state.

If the MAP provider sends a MAP_P_ABORT while the MAP process is waiting for a service indication, the MAP process returns to the idle state.

If the MAP provider sends a MAP_NOTICE while the MAP process is waiting for a service indication, the MAP process sends a MAP_CLOSE request to terminate the dialogue and returns to the idle state.

Error in MAP_PREPARE_GROUP_CALL indication

If the macro Check Indication takes the Error exit, the MAP process sends a MAP_U_ABORT request to the anchor MSC and returns to the idle state.

Negative response received from the ASCI handling process

If the ASCI handling process in the relay MSC returns a negative response to the Prepare Group Call request, the MAP process constructs a MAP_PREPARE_GROUP_CALL service response containing the appropriate error, constructs a MAP_CLOSE service request, sends them to the anchor MSC and returns to the idle state.

Error in MAP_FORWARD_GROUP_CALL_SIGNALLING indication

If the macro Check Indication takes the Error exit, the MAP process sends a MAP_U_ABORT request to the anchor MSC, sends an Abort to the ASCI handling process in the relay MSC and returns to the idle state.

Abort of MAP dialogue

After the dialogue with the anchor MSC has been established, the MAP service provider may abort the dialogue by issuing a MAP_P_ABORT indication, or the anchor MSC may send a MAP_U_ABORT indication or a MAP_CLOSE indication. In any of these cases, the MAP process sends an Abort to the ASCI handling process in the relay MSC and returns to the idle state.

If the MAP provider indicates a protocol problem by sending a MAP_NOTICE indication, the MAP process closes the dialogue with the anchor MSC, sends an Abort to the ASCI handling process in the anchor MSC and returns to the idle state.

Process ASCI_Relay_MSC

21.4_3.1(3)

Figure 21.4/3: Process in the Relay MSC for ASCI call handling

Signals to/from the left are to/from the A-MSC;
Signals to/from the right are to/from the R-MSC ASCI process

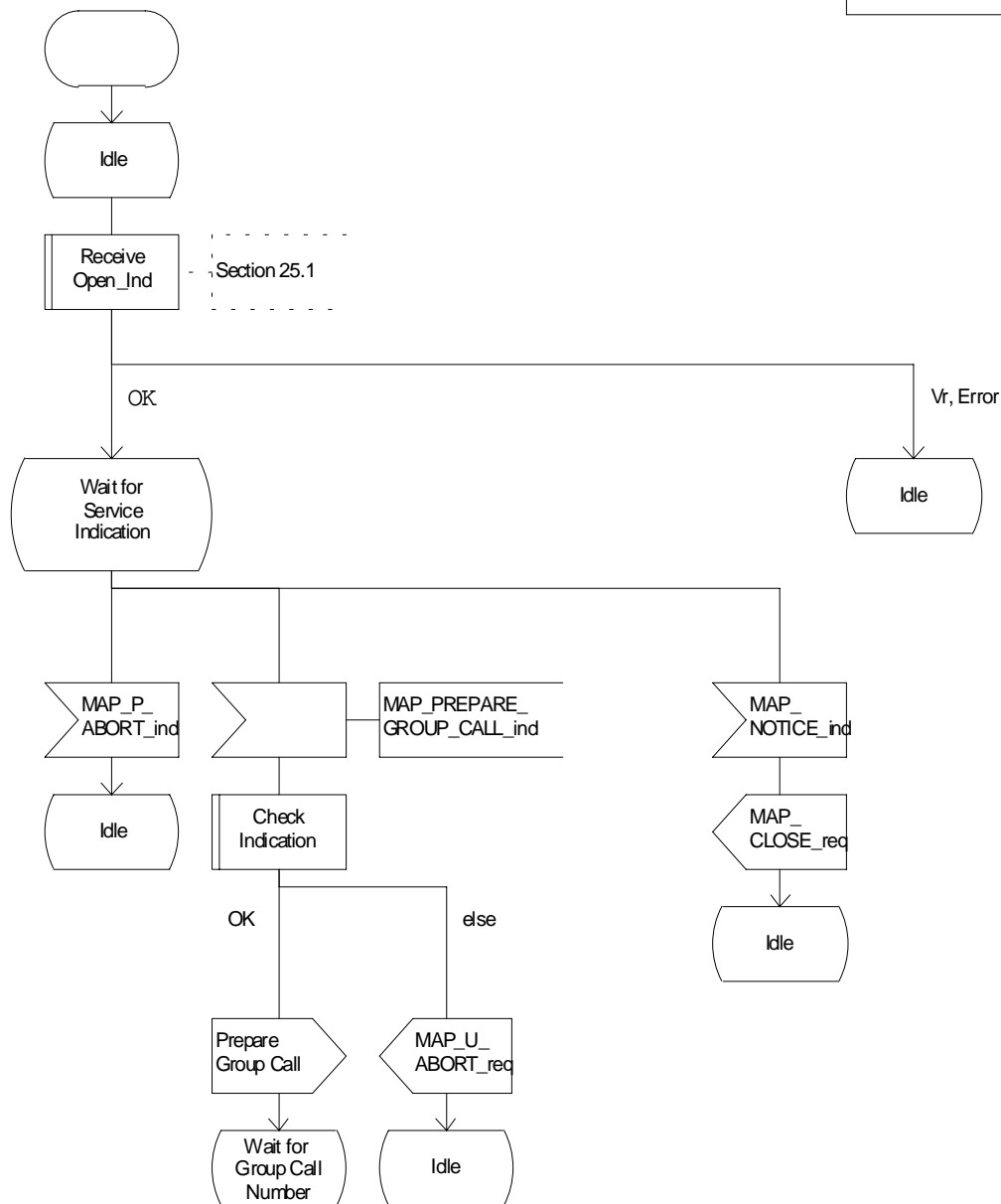


Figure 21.4/3 (sheet 1 of 3): Process ASCI_Relay_MSC

Process ASCII_Relay_MSC

21.4_3.2(3)

Figure 21.4/3: Process in the Relay MSC for ASCII call handling

Signals to/from the left are to/from the A-MSC;
Signals to/from the right are to/from the R-MSC ASCII process

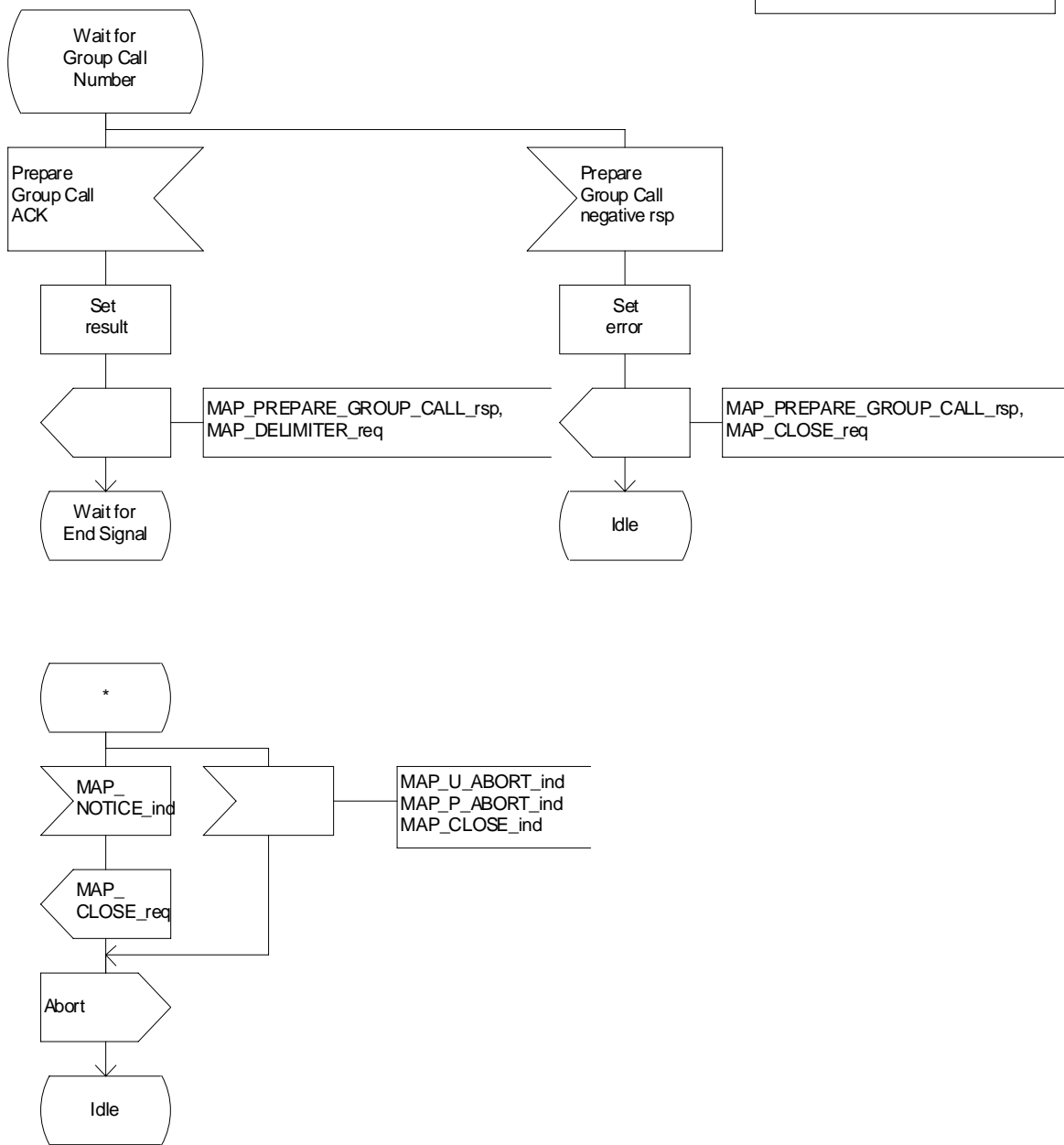


Figure 21.4/3 (sheet 2 of 3): Process ASCII_Relay_MSC

Process ASCII_Relay_MSC

21.4_3.3(3)

Figure 21.4/3: Process in the Relay MSC for ASCII call handling

Signals to/from the left are to/from the A-MSC;
Signals to/from the right are to/from the R-MSC ASCII process

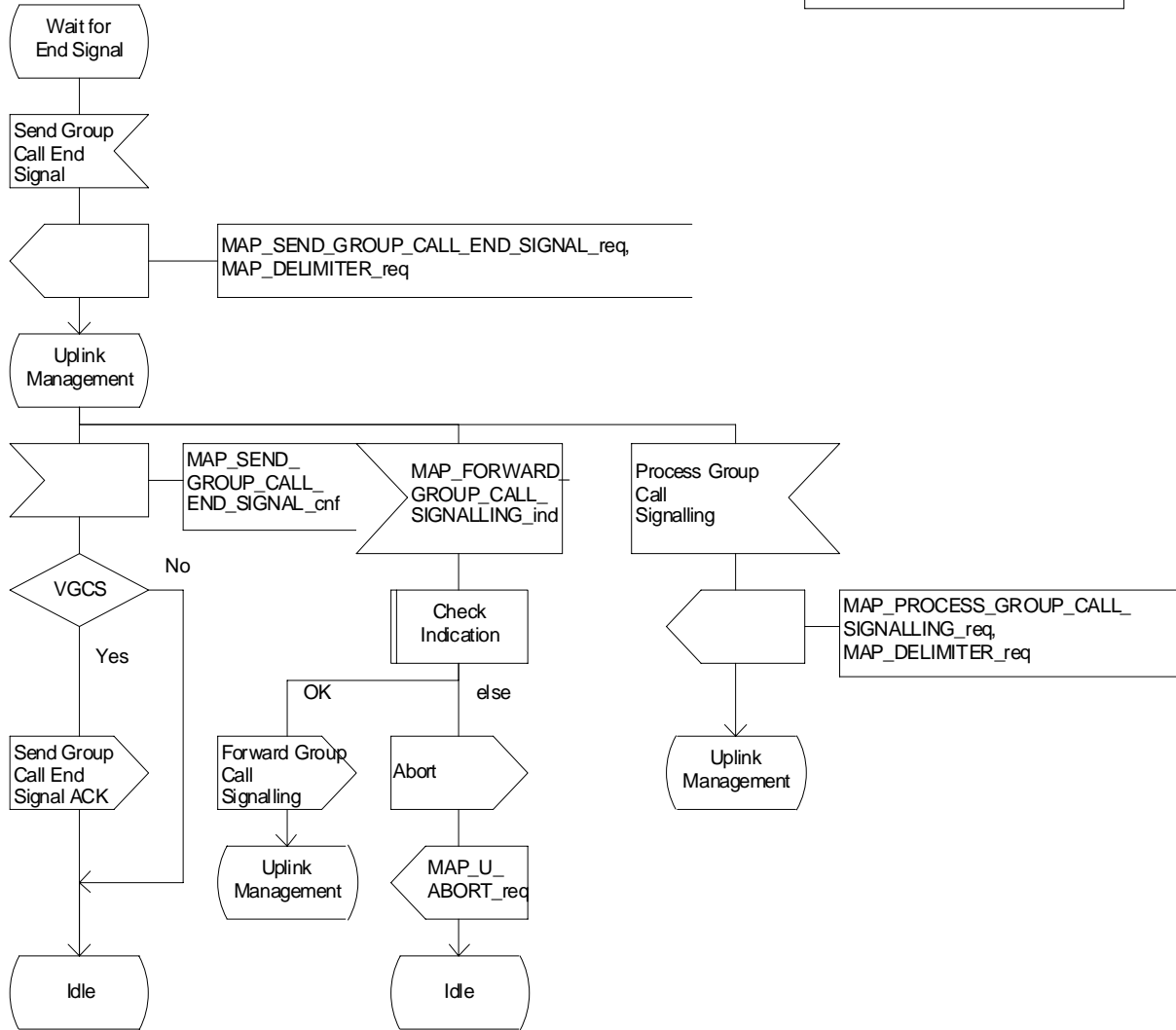
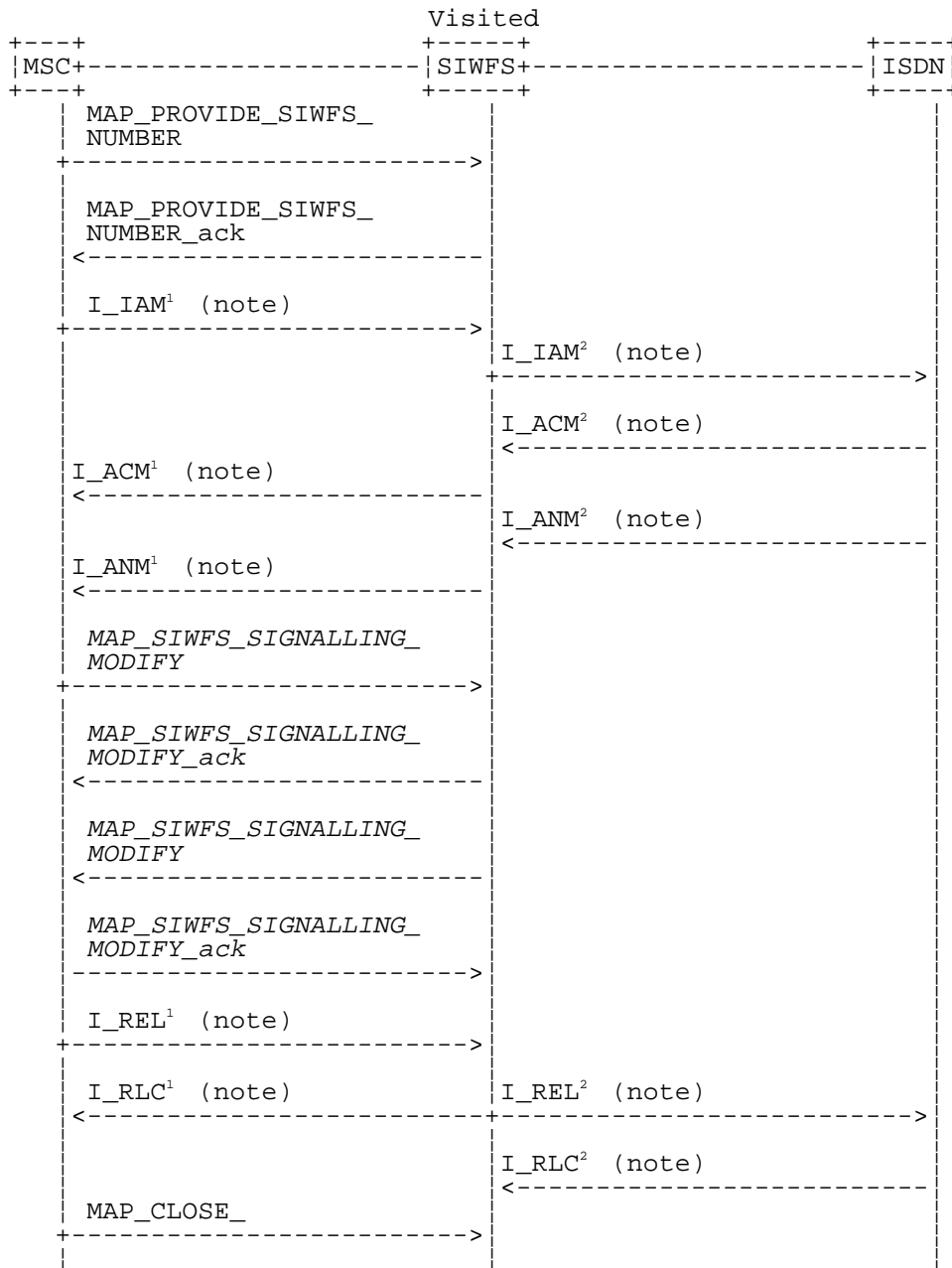


Figure 21.4/3 (sheet 3 of 3): Process ASCII_Relay_MSC

21.5 Allocation and modifications of resources in an SIWFS

21.5.1 General

The message flow for successful allocation and modification of resources in an SIWFS is shown in figure 21.5/1 (mobile originating call non-loop method), 21.5/2 (mobile originating call loop method) and 21.5/3 (mobile terminating call loop method).



Notes: xxx = *Optional Procedure*

TUP or ISUP may be used in signalling between MSCs, depending on the network type between the MSCs. The Release message can be initiated either by the calling or called subscriber. For further details on the TUP and ISUP procedures refer to the following CCITT Recommendations & ETSI specification:

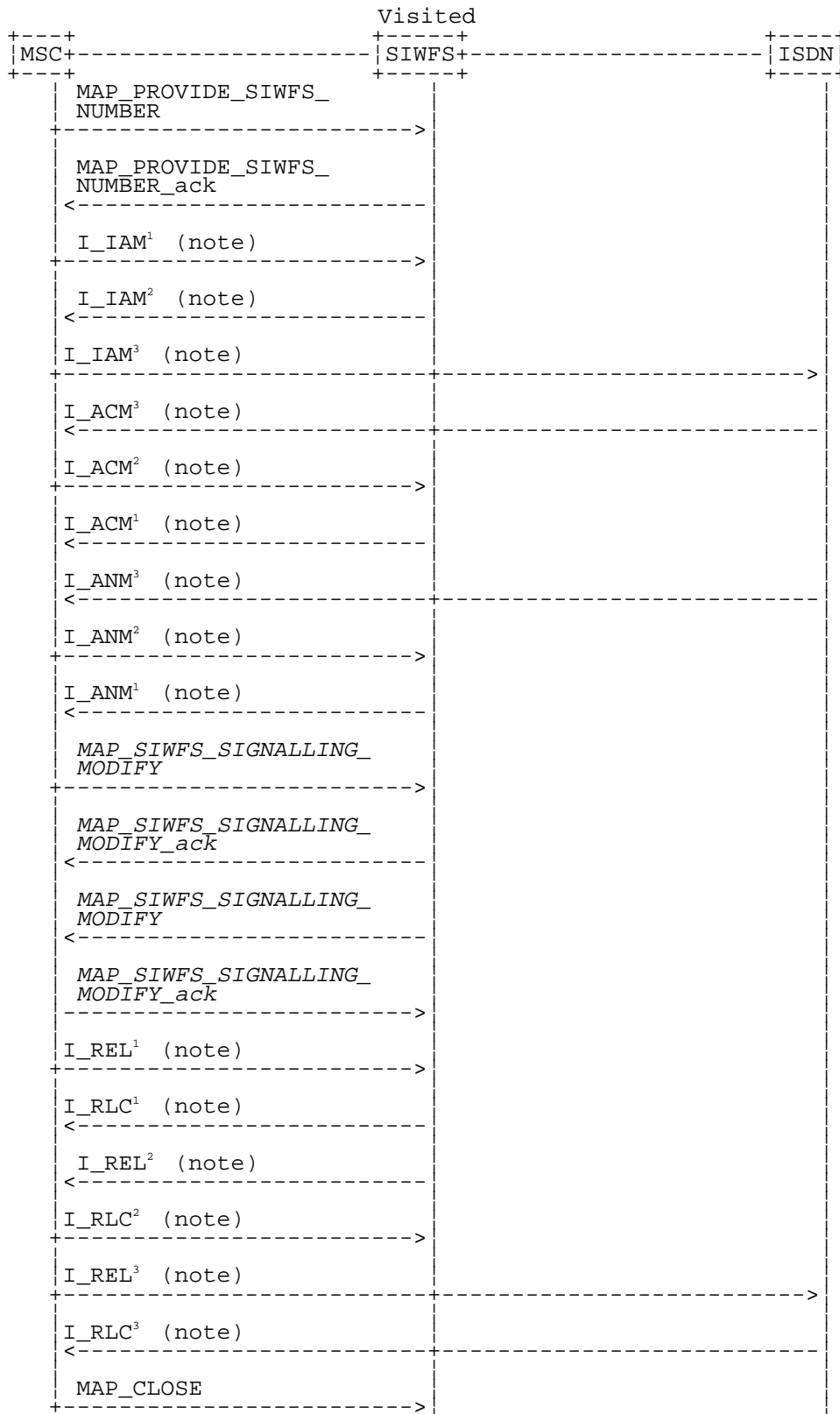
Q.721-725 - Telephone User Part (TUP);

ETS 300 356-1 - Integrated Services Digital Network (ISDN); Signalling System No.7; ISDN User Part (ISUP) version 2 for the international interface; Part 1: Basic services.

The number on the ISUP messages have been added to link the messages to respective signalling sequence.

The modification of SIWF resources could be initiated any time during the call either by the VMSC or the SIWFS.

Figure 21.5/1: Message flow for mobile originating call non-loop method



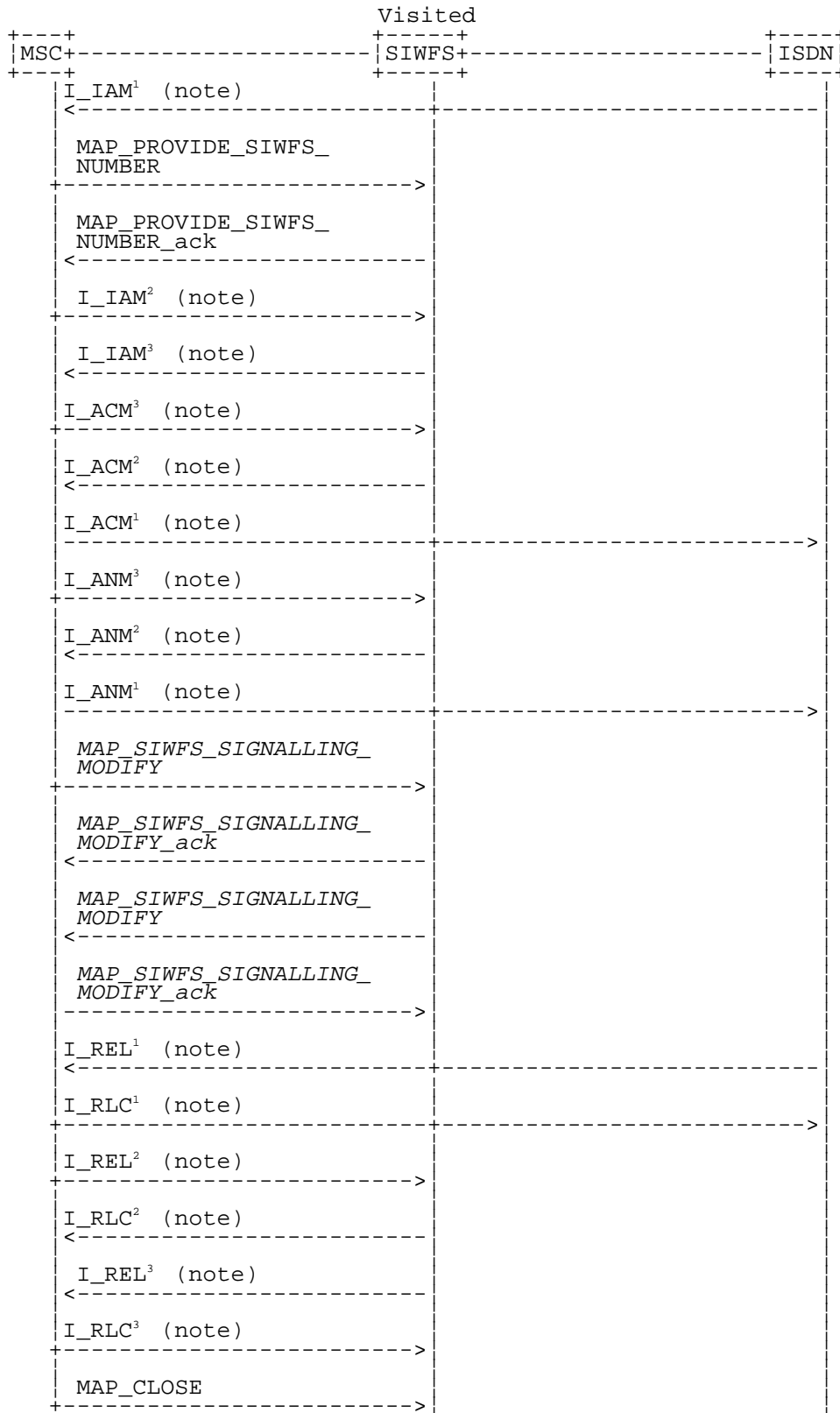
Notes:

xxx = Optional Procedure

TUP or ISUP may be used in signalling between MSCs, depending on the network type between the MSCs. The Release message can be initiated either by calling or called subscriber. For further details on the TUP and ISUP procedures refer to the following CCITT Recommendations & ETSI specification: Q.721-725 - Telephone User Part (TUP); ETS 300 356-1 - Integrated Services Digital Network (ISDN); Signalling System No.7; ISDN User Part (ISUP) version 2 for the international interface; Part 1: Basic services.

The number on the ISUP messages have been added to link the messages to respective signalling sequence.
 The modification of SIWFS resources could be initiated any time during the call either by the VMSC or the SIWFS.

Figure 21.5/2: Message flow for mobile originating call loop method



Notes:

xxx = Optional Procedure

TUP or ISUP may be used in signalling between MSCs, depending on the network type between the MSCs. The Release message can be initiated either by calling or called subscriber. For further details on the TUP and ISUP procedures refer to the following CCITT Recommendations & ETSI specification: Q.721-725 - Telephone User Part (TUP);

ETS 300 356-1 - Integrated Services Digital Network (ISDN); Signalling System No.7; ISDN User Part (ISUP) version 2 for the international interface; Part 1: Basic services.

The number on the ISUP messages have been added to link the messages to respective signalling sequence.

The modification of SIWF resources could be initiated any time during the call either by the VMSC or the SIWFS.

Figure 21.5/3: Message flow for mobile terminating call loop method

The following MAP services are used to allocate resources in an SIWFS:

MAP_PROVIDE_SIWFS_NUMBER see subclause 10.4.

The following MAP services are used to modify resources in an SIWFS:

MAP_SIWFS_SIGNALLING_MODIFY see subclause 10.5.

21.5.2 Process in the VMSC

The MAP process in the VMSC to allocate and modify resources in an SIWFS for a mobile call is shown in figure 21.5/4. The MAP process invokes macros not defined in this subclause; the definitions of these macros can be found as follows:

Receive_Open_Cnf see subclause 25.1.2;

Check_Confirmation see subclause 25.2.2.

21.5.2.1 Allocation of SIWFS resources

Successful Outcome

When the MAP process receives a Provide SIWFS Number request from the call handling process in the VMSC, it requests a dialogue with the SIWF whose identity is contained in the Provide SIWFS Number request by sending a MAP_OPEN service request, requests resources in the SIWFS using a MAP_PROVIDE_SIWFS_NUMBER service request and invokes the macro Receive_Open_Cnf to wait for the response to the dialogue opening request. If the dialogue opening is successful, the MAP process waits for a response from the SIWFS.

If the MAP process receives a MAP_PROVIDE_SIWFS_NUMBER service confirm from the SIWFS, the MAP process invokes the macro Check_Confirmation to check the content of the confirm.

If the macro Check_Confirmation takes the OK exit, the MAP process sends a Provide SIWFS Number ack containing the SIWFS Number received from the SIWFS to the call handling process in the VMSC and go to Wait_For_Modification state.

Earlier version MAP dialogue with the SIWFS

If the macro Receive_Open_Cnf takes the Vr exit, the MAP process sends an Abort to the call handling process in the VMSC and returns to the idle state.

Dialogue opening failure

If the macro Receive_Open_Cnf indicates that the dialogue with the SIWFS could not be opened, the MAP process sends an Abort to the call handling process in the VMSC and returns to the idle state.

Error in MAP_PROVIDE_SIWFS_NUMBER confirm

If the MAP_PROVIDE_SIWFS_NUMBER service confirm contains a user error or a provider error, or the macro Check_Confirmation indicates that there is a data error, the MAP process sends a Provide SIWFS number negative response to the call handling process in the VMSC and returns to the idle state.

Call release

If the call handling process in the VMSC indicates that the call has been aborted, the MAP process returns to the idle state. Any response from the SIWFS will be discarded.

If the call handling process in the VMSC indicates that the traffic channel has been released (i.e. call released by a user) a MAP_CLOSE_req is sent and the process is returned to the idle state.

Abort of SIWFS dialogue

During the time an answer is expected from the SIWFS, the MAP service provider may abort the dialogue by issuing a MAP_P_ABORT indication, or the SIWFS may send a MAP_U_ABORT indication or a MAP_CLOSE indication. In any of these cases, the MAP process sends a Provide SIWFS number negative response to the call handling process in the VMSC and returns to the idle state.

If the MAP provider indicates a protocol problem by sending a MAP_NOTICE indication, the MAP process closes the dialogue with the SIWFS, sends a Provide SIWFS number negative response indicating system failure to the call handling process in the VMSC and returns to the idle state.

After the dialogue with the SIWFS has been established, the MAP service provider may abort the dialogue by issuing a MAP_P_ABORT indication, or the SIWFS may send a MAP_U_ABORT indication or a MAP_CLOSE indication. In any of these cases, the MAP process returns to the idle state.

If the MAP provider indicates a protocol problem by sending a MAP_NOTICE indication, the MAP process closes the dialogue with the SIWFS, and returns to the idle state.

21.5.2.2 Modification of SIWFS resources initiated by the user

Successful Outcome

When the MAP process receives an SIWFS Signalling Modify request from the call handling process in the VMSC, it requests a dialogue with the SIWFS whose identity is contained in the SIWFS Signalling Modify request by sending a MAP_SIWFS_SIGNALLING_MODIFY service request and waits for a response from the SIWFS.

If the MAP process receives a MAP_SIWFS_SIGNALLING_MODIFY service confirm from the SIWFS, the MAP process invokes the macro Check_Confirmation to check the content of the confirm.

If the macro Check_Confirmation takes the OK exit, the MAP process sends an SIWFS Signalling Modify ack containing the response received from the SIWFS to the call handling process in the VMSC and go to Wait_For_Modification state.

Error in MAP_SIWFS_SIGNALLING_MODIFY confirm

If the MAP_SIWFS_SIGNALLING_MODIFY service confirm contains a user error or a provider error, or the macro Check_Confirmation indicates that there is a data error, the MAP process sends an SIWFS Signalling Modify negative response to the call handling process in the VMSC and go to Wait_For_Modification state.

Abort of SIWFS dialogue

During the time an answer is expected from the SIWFS, the MAP service provider may abort the dialogue by issuing a MAP_P_ABORT indication, or the SIWFS may send a MAP_U_ABORT indication or a MAP_CLOSE indication. In any of these cases, the MAP process sends an SIWFS Signalling Modify negative response to the call handling process in the VMSC and returns to the idle state.

If the MAP provider indicates a protocol problem by sending a MAP_NOTICE indication, the MAP process closes the dialogue with the SIWFS, sends an SIWFS Signalling Modify negative response indicating system failure to the call handling process in the VMSC and returns to the idle state.

21.5.2.3 Modification of SIWFS resources initiated by the SIWFS

Successful outcome

If a MAP_SIWFS_SIGNALLING_MODIFY service indication is received, the MAP process sends an SIWFS signalling modify Info request to the call handling process in the VMSC, and waits for a response. The SIWFS signalling modify request contains the parameters received in the MAP_SIWFS_SIGNALLING_MODIFY service indication.

If the call handling process in the VMSC returns an SIWFS signalling modify ack, the MAP process constructs a MAP_SIWFS_SIGNALLING_MODIFY service response contained in the Provide SIWFS Number ack, send it to the SIWFS and go to Wait_For_Modification state.

Negative response from VMSC call handling process

If the call handling process in the VMSC returns a negative response the MAP process constructs a MAP_SIWFS_SIGNALLING_MODIFY service response containing the appropriate error, send it to the SIWFS and go to Wait_For_Modification state.

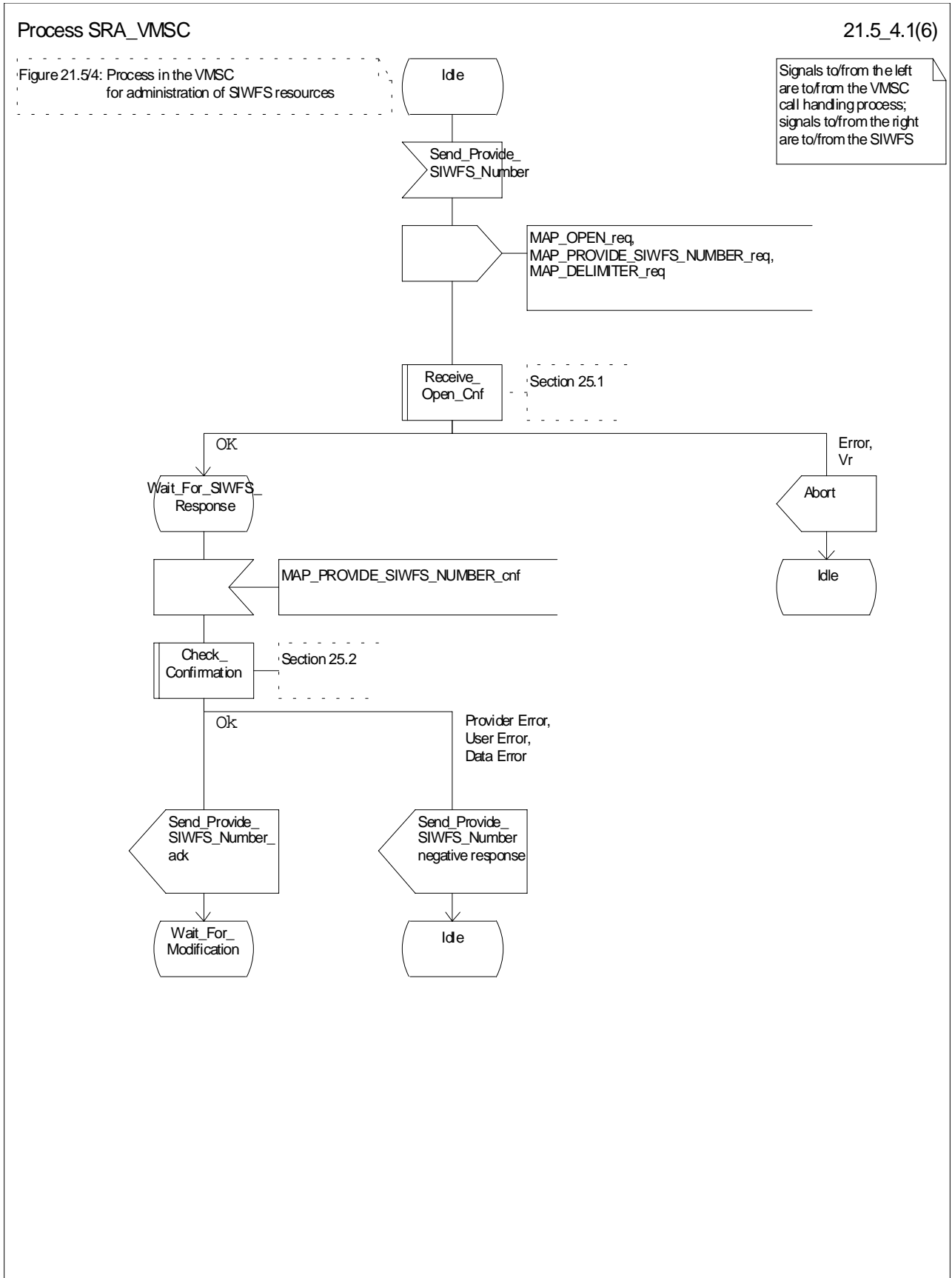


Figure 21.5/4 (sheet 1 of 6): Process SRA (SIWFS_RESOURCE_ADMINISTRATION)_VMSC

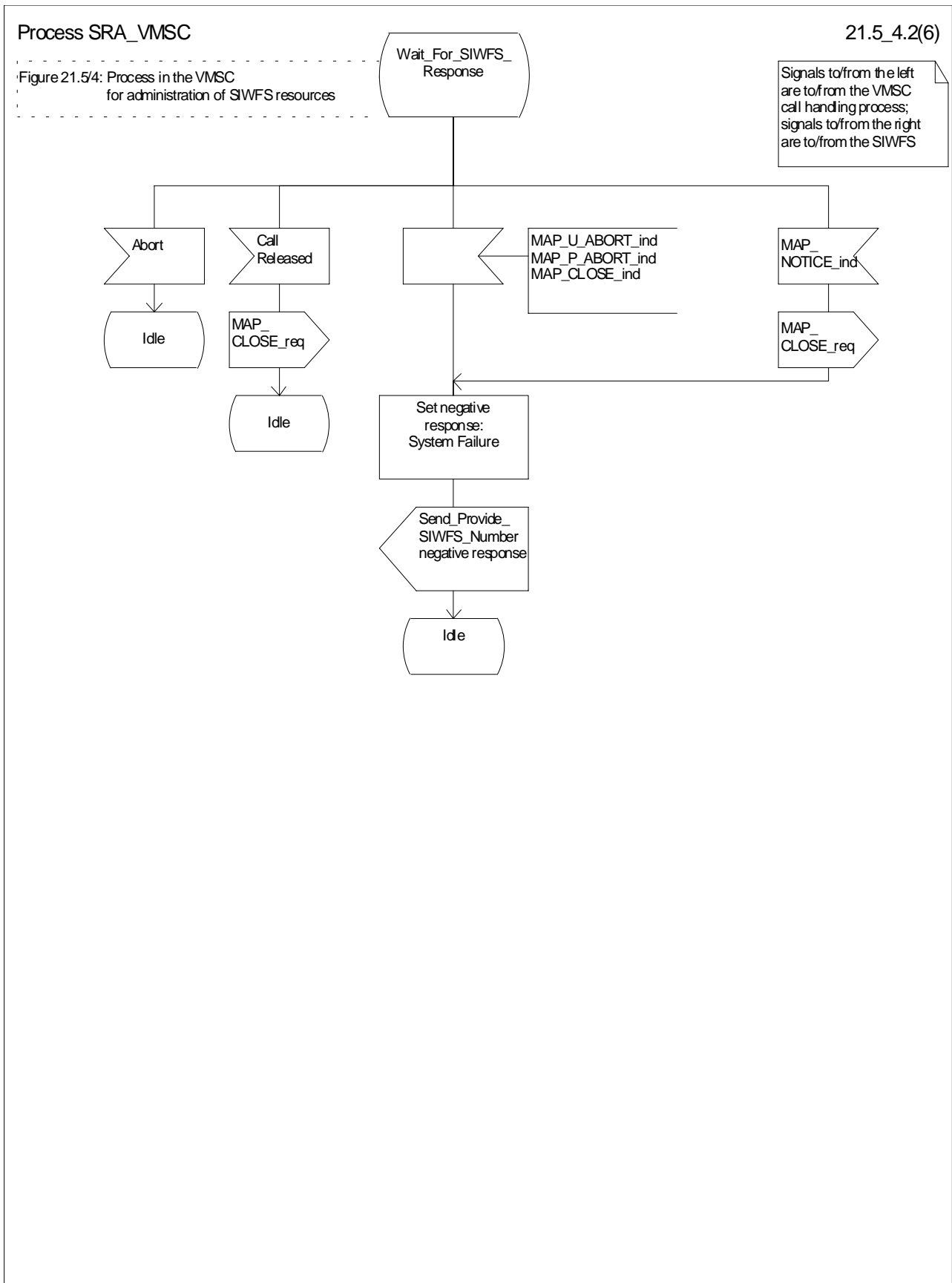


Figure 21.5/4 (sheet 2 of 6): Process SRA_VMSC

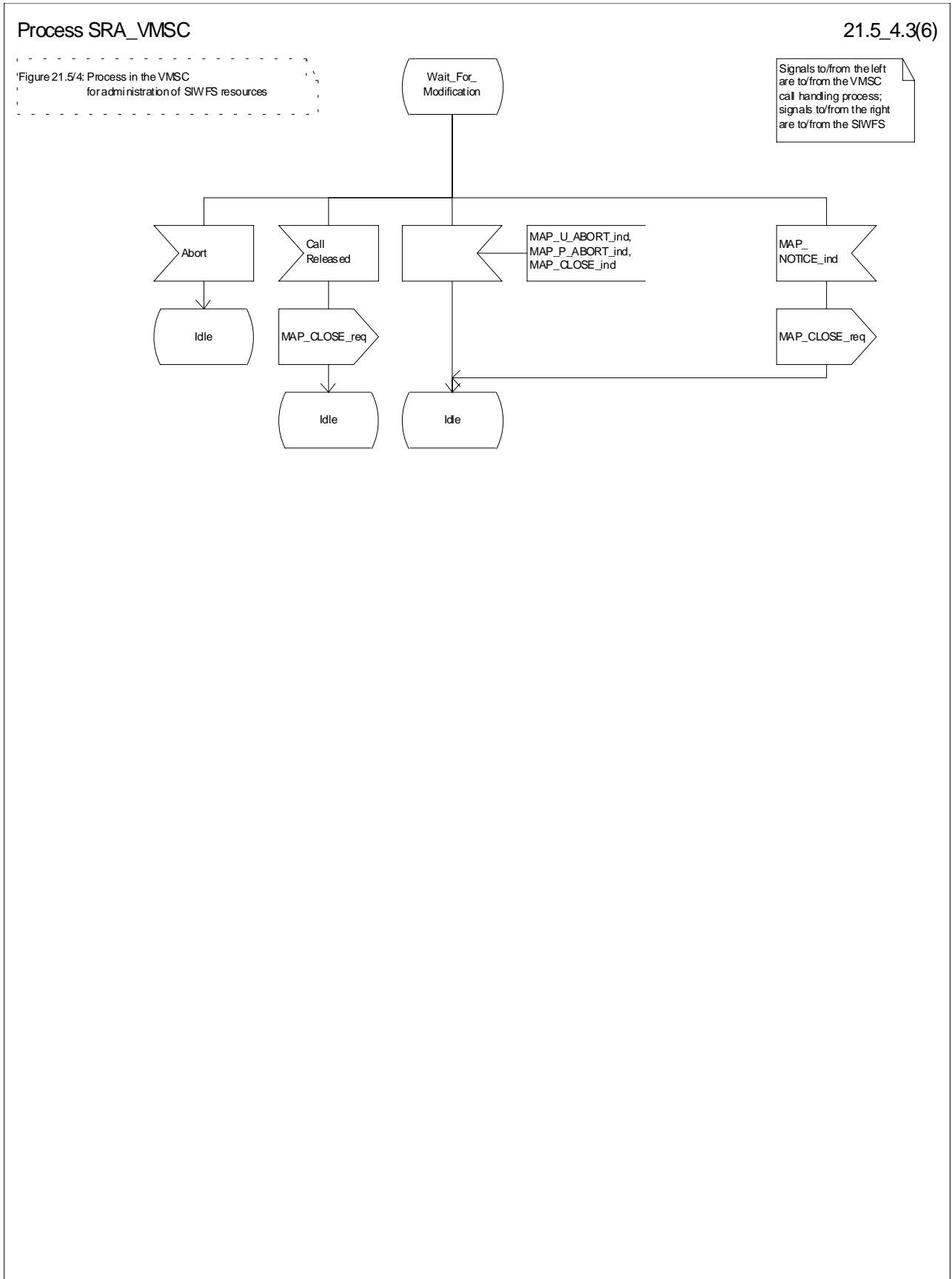


Figure 21.5/4 (sheet 3 of 6): Process SRA_VMSC

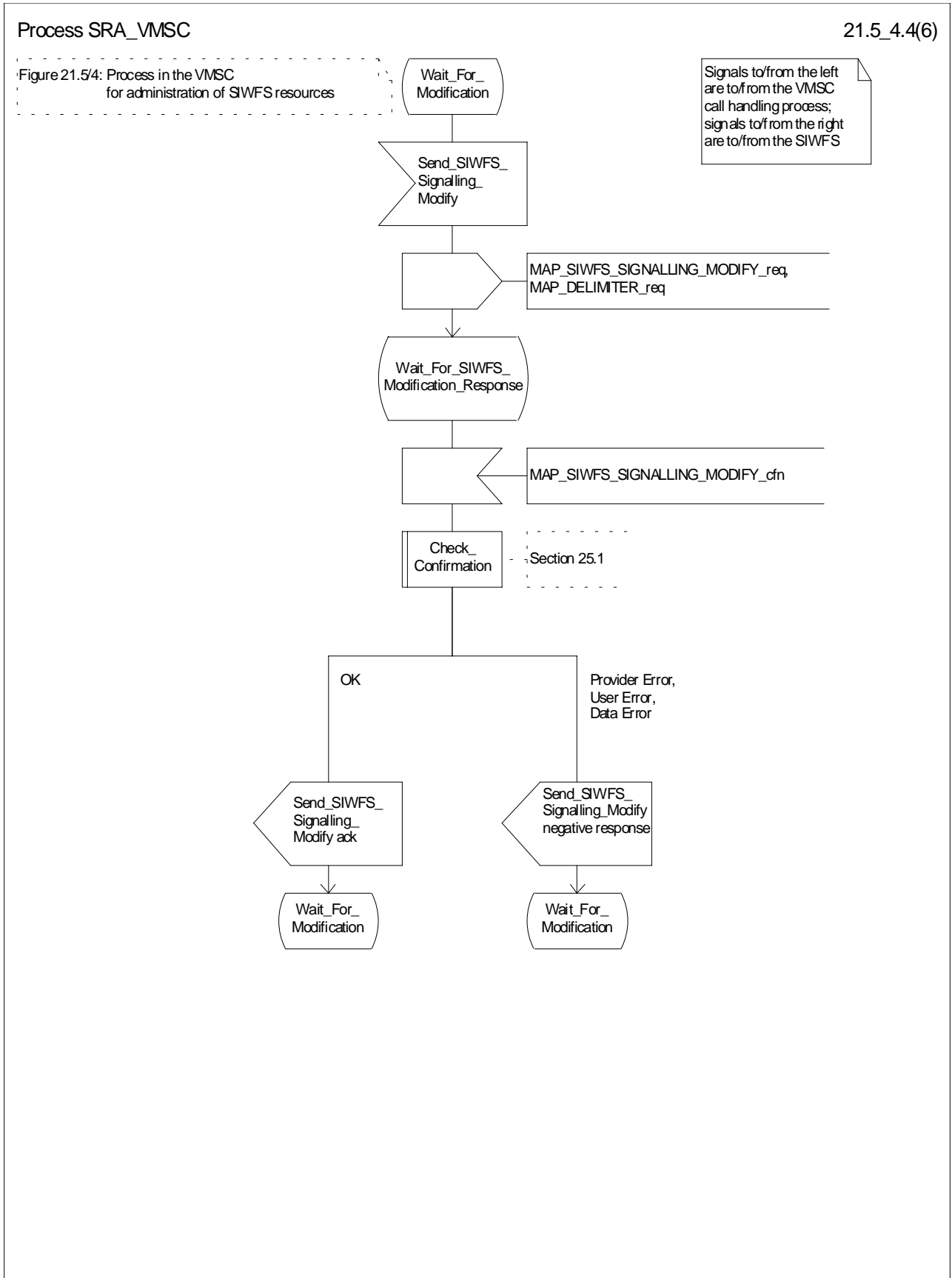


Figure 21.5/4 (sheet 4 of 6): Process SRA_VMSC

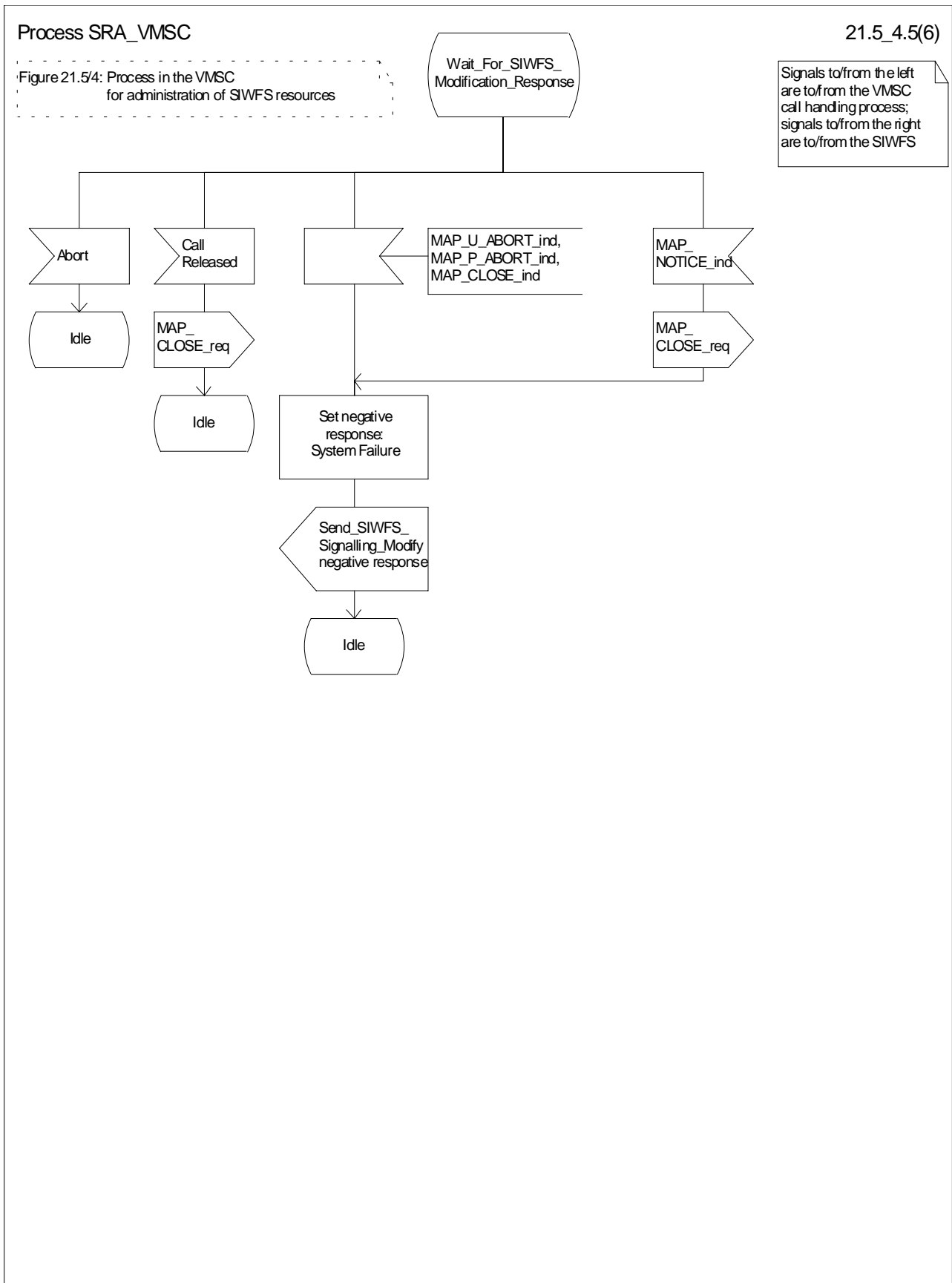


Figure 21.5/4 (sheet 5 of 6): Process SRA_VMSC

Process SRA_VMSC

21.5_4.6(6)

Figure 21.5/4: Process in the VMSC for administration of SIWFS resources

Signals to/from the left are to/from the VMSC call handling process; signals to/from the right are to/from the SIWFS

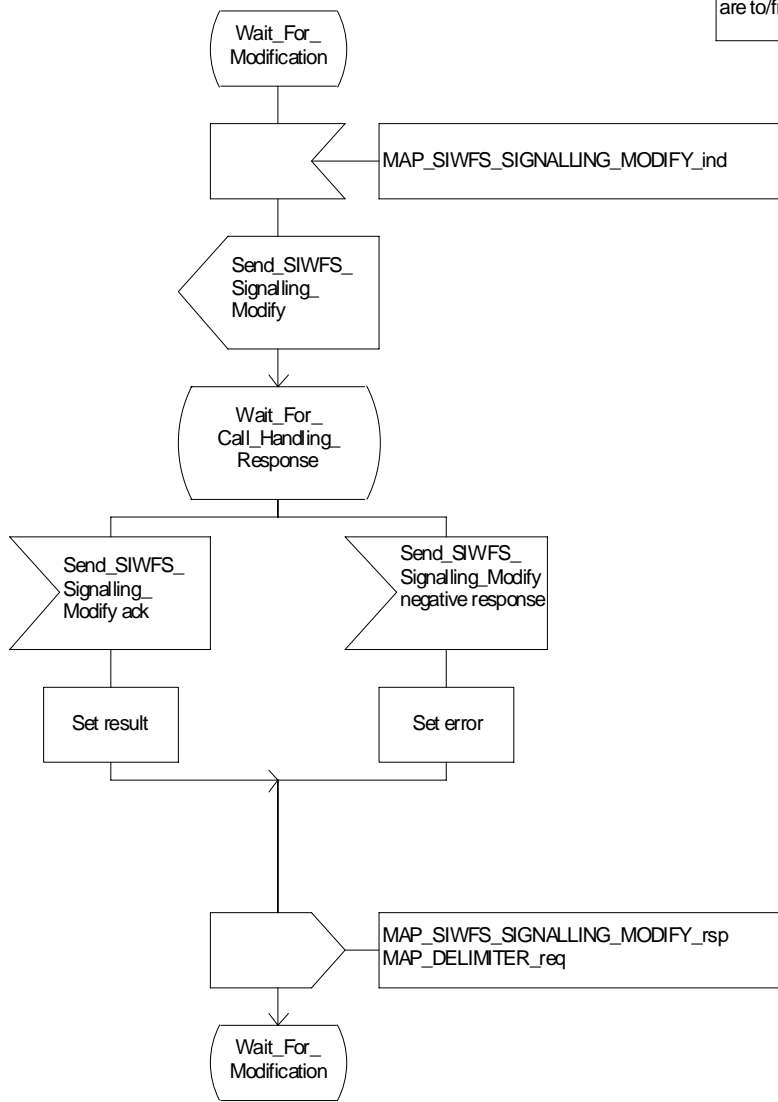


Figure 21.5/4 (sheet 6 of 6): Process SRA_VMSC

21.5.3 Process in the SIWFS

The MAP process in the SIWFS to allocate and modify SIWFS resources for a mobile call is shown in figure 21.5/5. The MAP process invokes macros not defined in this subclause; the definitions of these macros can be found as follows:

Receive_Open_Ind	see subclause 25.1.1.
Check_Confirmation	see subclause 25.2.2.

21.5.3.1 Procedures for allocation of SIWFS resources

Successful outcome

When the MAP process receives a MAP_OPEN indication with the application context locInfoRetrieval, it checks it by invoking the macro Receive_Open_Ind.

If the macro takes the OK exit, the MAP process waits for a service indication.

If a MAP_PROVIDE_SIWFS_NUMBER service indication is received, the MAP process sends a Provide SIWFS number Info request to the call handling process in the SIWFS, and waits for a response. The Provide SIWFS number request contains the parameters received in the MAP_PROVIDE_SIWFS_NUMBER service indication.

If the call handling process in the SIWFS returns a Provide SIWFS number ack, the MAP process constructs a MAP_PROVIDE_SIWFS_NUMBER service response containing the routing information contained in the Provide SIWFS Number ack, constructs a MAP_DELIMITER service request, sends them to the VMSC and go to Wait_For_Modification state.

Earlier version MAP dialogue with the VMSC

If the macro Receive_Open_Ind takes the Vr exit, the MAP process returns to the idle state.

Dialogue opening failure

If the macro Receive_Open_Ind takes the Error exit, the MAP process returns to the idle state.

If the MAP provider sends a MAP_P_ABORT while the MAP process is waiting for a service indication, the MAP process returns to the idle state.

If the MAP provider sends a MAP_NOTICE while the MAP process is waiting for a service indication, the MAP process sends a MAP_CLOSE request to terminate the dialogue and returns to the idle state.

Negative response from SIWFS call handling process

If the call handling process in the SIWFS returns a negative response the MAP process constructs a MAP_PROVIDE_SIWFS_NUMBER service response containing the appropriate error, constructs a MAP_CLOSE service request, sends them to the VMSC and returns to the idle state.

Call release

If the call handling process in the SIWFS indicates that the call has been aborted, the MAP process returns to the idle state. Any response from the VMSC will be discarded.

If the call handling process in the SIWFS indicates that the traffic channel has been released (i.e.call released by a user) a MAP_CLOSE_req is sent and the process is returned to the idle state.

Abort of VMSC dialogue

After the dialogue with the VMSC has been established, the MAP service provider may abort the dialogue by issuing a MAP_P_ABORT indication, or the VMSC may send a MAP_U_ABORT indication or a MAP_CLOSE indication. In any of these cases, the MAP process returns to the idle state.

If the MAP provider indicates a protocol problem by sending a MAP_NOTICE indication, the MAP process closes the dialogue with the VMSC, and returns to the idle state.

21.5.3.2 Process for modification of SIWFS resources initiated by the user

Successful outcome

If a MAP_SIWFS_SIGNALLING_MODIFY service indication is received, the MAP process sends an SIWFS signalling modify Info request to the call handling process in the SIWFS, and waits for a response. The SIWFS signalling modify request contains the parameters received in the MAP_SIWFS_SIGNALLING_MODIFY service indication.

If the call handling process in the SIWFS returns an SIWFS signalling modify ack, the MAP process constructs a MAP_SIWFS_SIGNALLING_MODIFY service response contained in the Provide SIWFS Number ack, send it to the VMSC and go to Wait_For_Modification state.

Negative response from SIWFS call handling process

If the call handling process in the SIWFS returns a negative response the MAP process constructs a MAP_SIWFS_SIGNALLING_MODIFY service response containing the appropriate error, send it to the VMSC and go to Wait_For_Modification state.

21.5.3.3 Process for modification of SIWFS resources initiated by the SIWFS

Successful Outcome

When the MAP process receives an SIWFS Signalling Modify request from the call handling process in the SIWF, it requests a dialogue with the VMSC whose identity is contained in the VMSC Signalling Modify request by sending a MAP_DELIMITER service request, requests resources in the VMSC using a MAP_SIWFS_SIGNALLING_MODIFY service request, the MAP process waits for a response from the VMSC.

If the MAP process receives a MAP_SIWFS_SIGNALLING_MODIFY service confirm from the VMSC, the MAP process invokes the macro Check_Confirmation to check the content of the confirm.

If the macro Check_Confirmation takes the OK exit, the MAP process sends an SIWFS Signalling Modify ack containing the response received from the VMSC to the call handling process in the SIWF and go to Wait_For_Modification state.

Error in MAP_SIWFS_SIGNALLING_MODIFY confirm

If the MAP_SIWFS_SIGNALLING_MODIFY service confirm contains a user error or a provider error, or the macro Check_Confirmation indicates that there is a data error, the MAP process sends an SIWFS Signalling Modify negative response to the call handling process in the SIWFS and go to Wait_For_Modification state.

Abort of SIWFS dialogue

During the time an answer is expected from the VMSC, the MAP service provider may abort the dialogue by issuing a MAP_P_ABORT indication, or the VMSC may send a MAP_U_ABORT indication or a MAP_CLOSE indication. In any of these cases, the MAP process sends an SIWFS Signalling Modify negative response to the call handling process in the SIWFS and returns to the idle state.

If the MAP provider indicates a protocol problem by sending a MAP_NOTICE indication, the MAP process closes the dialogue with the VMSC, sends an SIWFS Signalling Modify negative response indicating system failure to the call handling process in the SIWFS and returns to the idle state.

Process SRA_SIWFS

21.5_5.1(5)

Figure 21.5/5: Process in the SIWFS for administration of SIWFS resources

Signals to/from the left are to/from the SIWFS call handling process; signals to/from the right are to/from the VMSC

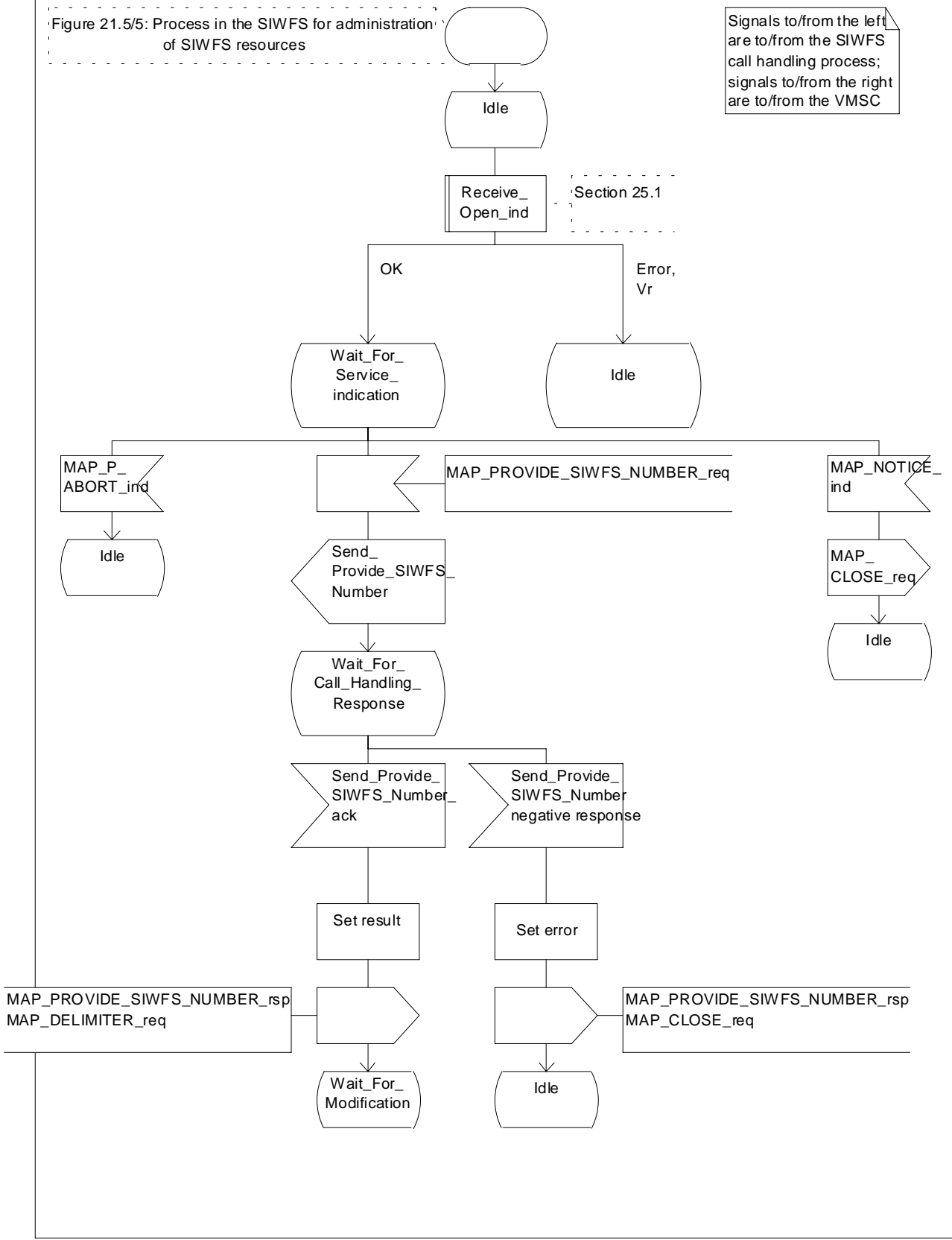


Figure 21.5/5 (sheet 1 of 5): Process SRA_SIWFS

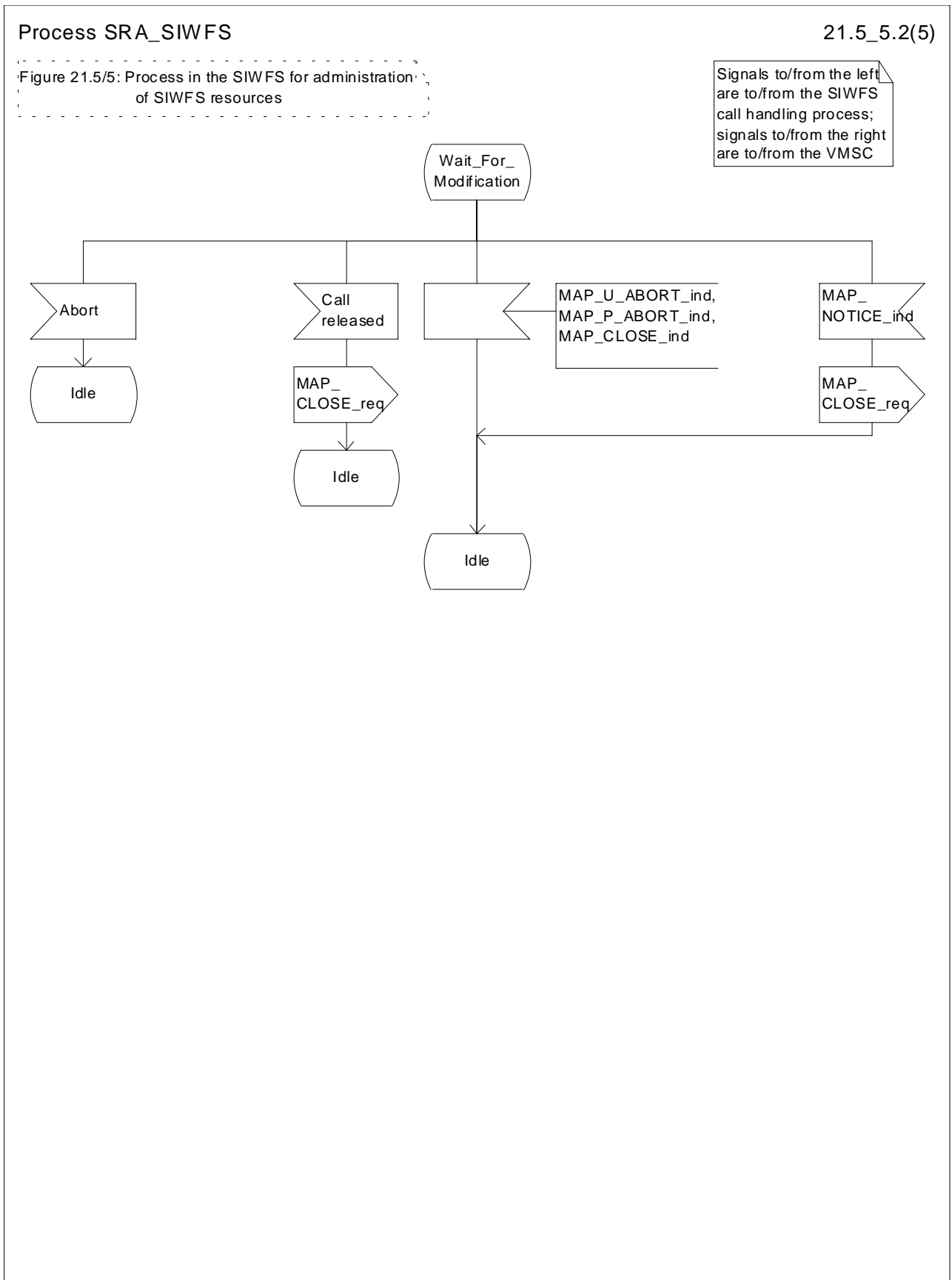


Figure 21.5/5 (sheet 2 of 5): Process SRA_SIWFS

Process SRA_SIWFS

21.5_5.3(5)

Figure 21.5/5: Process in the SIWFS for administration of SIWFS resources

Signals to/from the left are to/from the SIWFS call handling process; signals to/from the right are to/from the VMSC

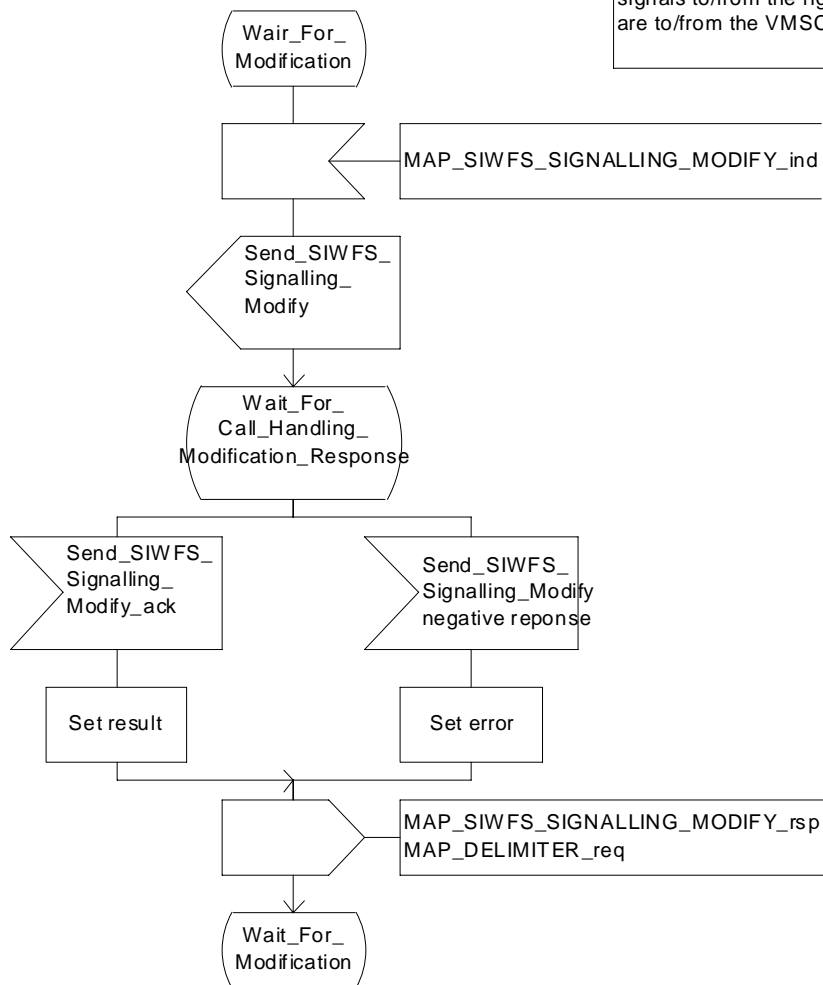


Figure 21.5/5 (sheet 3 of 5): Process SRA_SIWFS

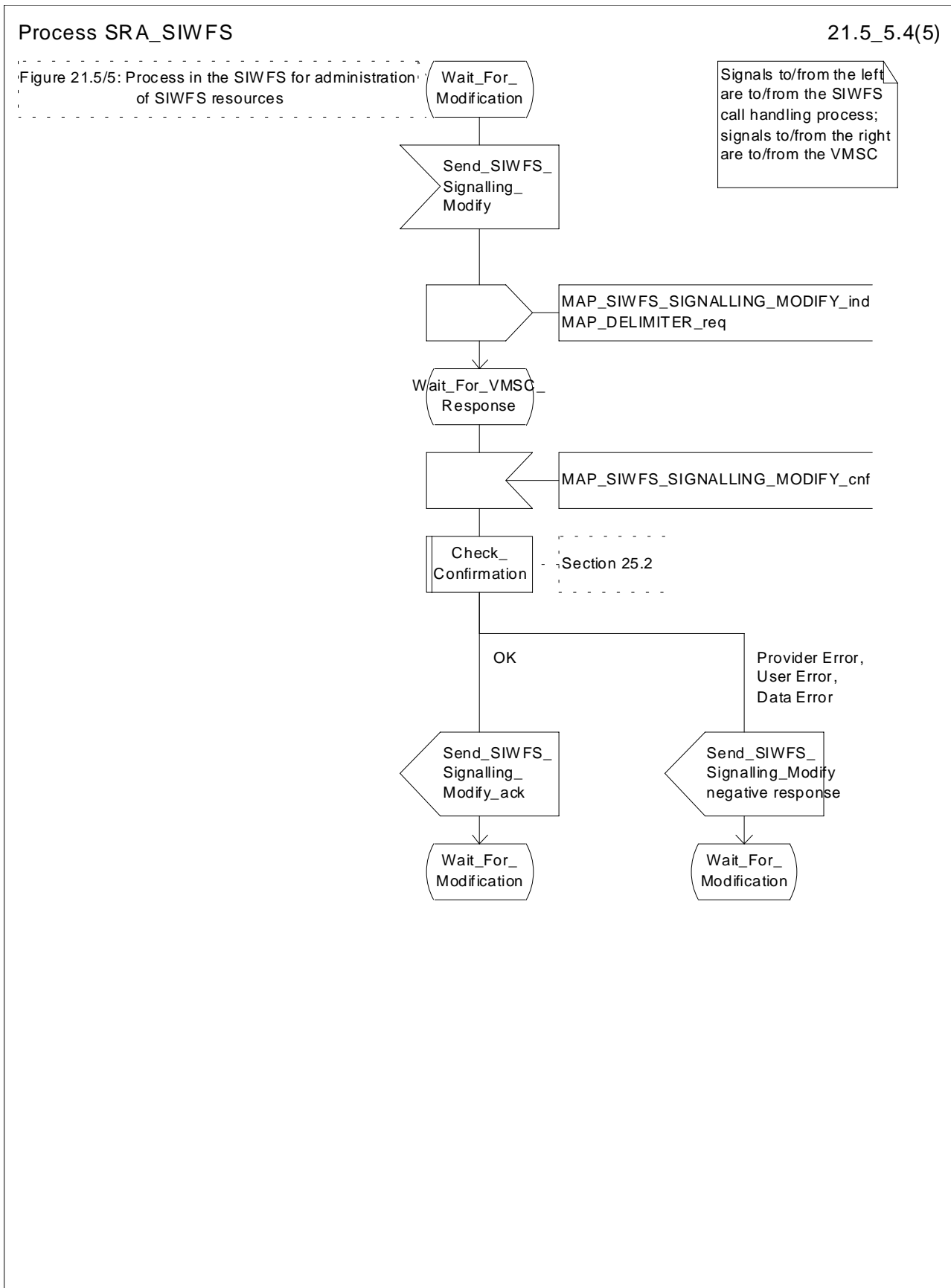


Figure 21.5/5 (sheet 4 of 5): Process SRA_SIWFS

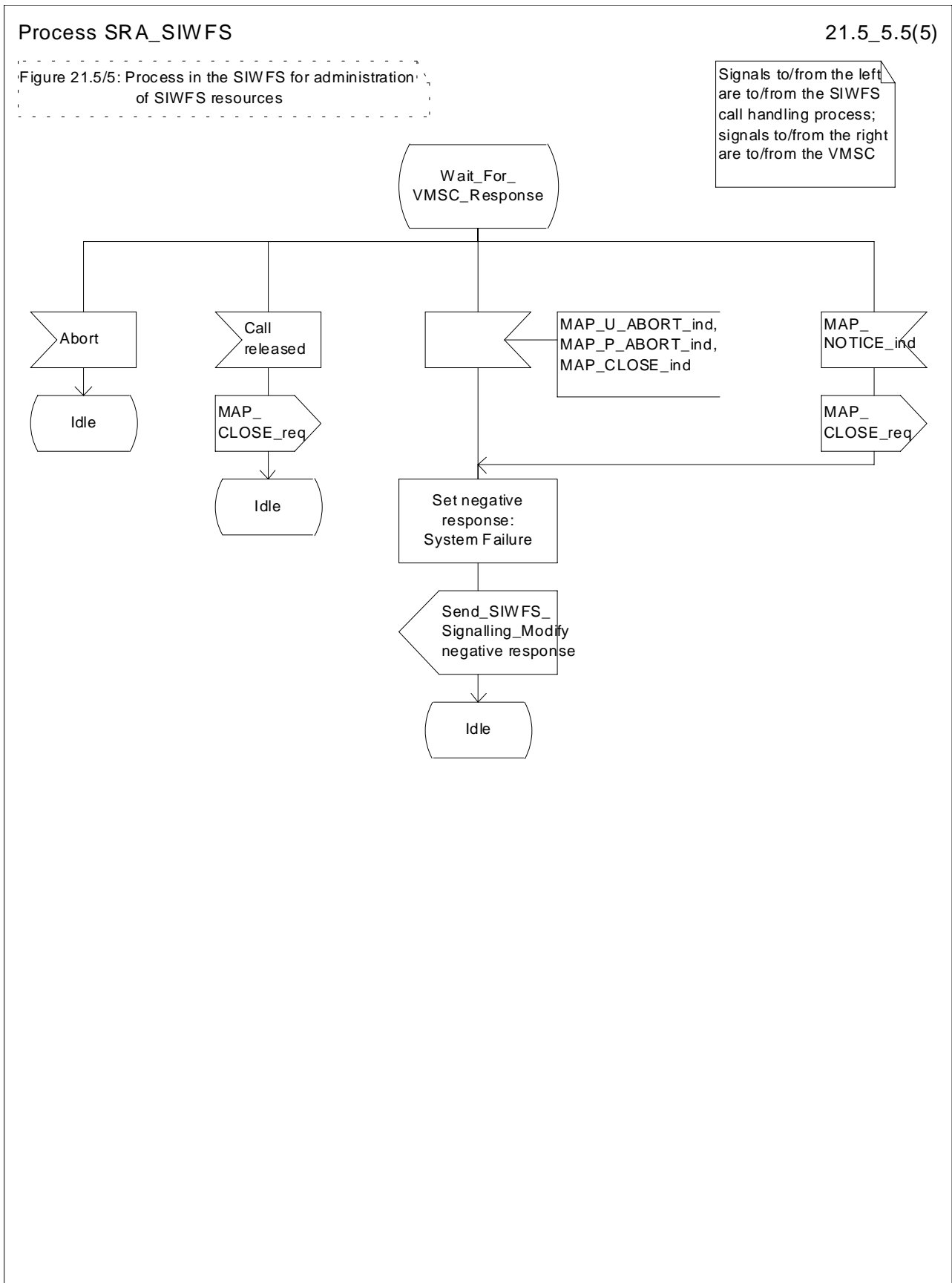


Figure 21.5/5 (sheet 5 of 5): Process SRA_SIWFS

21.6 Setting of Reporting State

21.6.1 General

The message flow for setting the reporting state in a stand-alone dialogue is shown in figure 21.6.1/1.

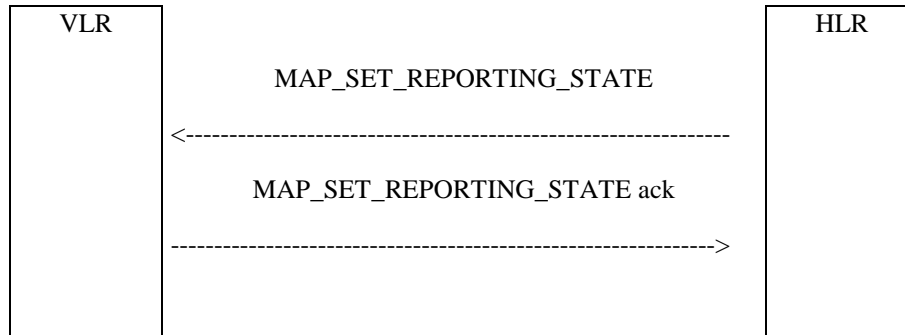


Figure 21.6/1: Message Flow for Setting the Reporting State

In Set Reporting State, the HLR can request a start or a stop of monitoring in the VLR.

21.6.2 Process in the HLR for Set Reporting State stand-alone

The MAP process in the HLR to set the reporting state in the VLR in a separate stand-alone dialogue is shown in figure 21.6/2. The MAP process invokes macros not defined in this subclause; the definitions of these macros can be found as follows:

Receive_Open_Cnf	see subclause 25.1.2;
Check_Confirmation	see subclause 25.2.2.

Successful Outcome

When the MAP process receives a Start Reporting or Stop Reporting request from the CCBS application process in the HLR, it requests a dialogue with the VLR whose identity is contained in the request by sending a MAP_OPEN service request and sending the necessary information using a MAP_SET_REPORTING_STATE service request. The HLR then invokes the macro Receive_Open_Cnf to wait for the response to the dialogue opening request. If the dialogue opening is successful, the MAP process waits for a response from the VLR.

If the MAP process receives a MAP_SET_REPORTING_STATE service confirm from the VLR, the MAP process invokes the macro Check_Confirmation to check the content of the confirm.

If the macro Check_Confirmation takes the OK exit and the request was for Start Reporting, the MAP process sends a positive acknowledgement containing the information received from the VLR to the CCBS application process in the HLR and returns to the idle state. In the case of Stop Reporting the CCBS application process returns to the idle state.

Failure of dialogue opening with the VLR

If the macro Receive_Open_Cnf takes the Vr exit or the Error exit, the MAP process sends (in the case of Start Reporting) a negative response to the CCBS application process in the HLR and returns to the idle state. In the case of Stop Reporting the process returns to the idle state.

Error in MAP_SET_REPORTING_STATE confirm

If the MAP_SET_REPORTING_STATE service confirm contains a user error or a provider error, or the macro Check_Confirmation indicates that there is a data error, the MAP process sends a negative response (in the case of Start Reporting) to the CCBS application process in the HLR and returns to the idle state. In the case of Stop Reporting the CCBS application process returns to the idle state.

Abort of VLR dialogue

After the dialogue with the VLR has been established, the MAP service provider may abort the dialogue by issuing a MAP_P_ABORT or a MAP_U_ABORT indication. If the request was for the Start Reporting, the MAP process sends a Start Reporting negative response to the CCBS application process in the HLR and returns to the idle state.

If the MAP provider indicates a protocol problem by sending a MAP_NOTICE indication, the MAP process closes the dialogue with the VLR, sends a negative response (in the case of the Start Reporting) indicating system failure to the CCBS application process in the HLR and returns to the idle state. In the case of Stop Reporting the CCBS application process returns to the idle state.

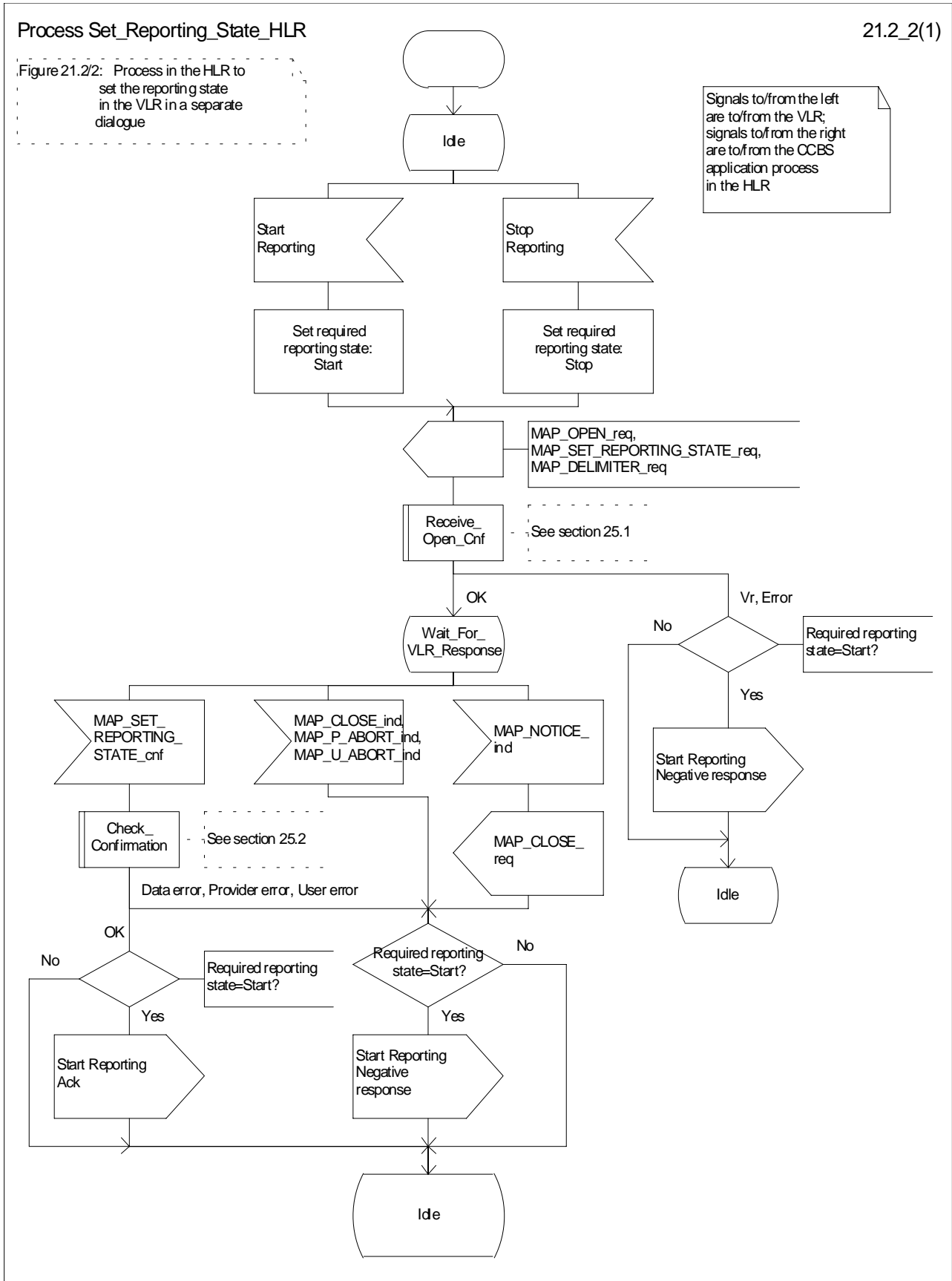


Figure 21.6/2: Process Set_Reporting_State_HLR

21.6.3 Reporting co-ordinator process in the VLR

The MAP co-ordinating process in the VLR to handle a dialogue opened with the reporting application context is shown in figure 21.6/3. The MAP process invokes a macro not defined in this subclause; the definition of this macro can be found as follows:

Receive_Open_Ind see subclause 25.1.1.

Any reporting process in the VLR starts by the VLR receiving a MAP-OPEN service indication. If that service is successful, the VLR can handle reporting indications from the HLR. Table 21.6/1 shows the co-ordinating process' reaction on receipt of specific reporting indications from the HLR. After the relevant process is invoked, the received service indication is sent to that process.

Table 21.6/1: Relationship between received service indication and invoked process in the VLR

Service indication received	Process invoked
MAP_REMOTE_USER_FREE_ind	REMOTE_USER_FREE_VLR
MAP_SET_REPORTING_STATE_ind	SET_REPORTING_STATE_VLR

After creation of the user process the co-ordinator relays the messages between the MAP protocol machine and the invoked process until a request or an indication for dialogue termination is received.

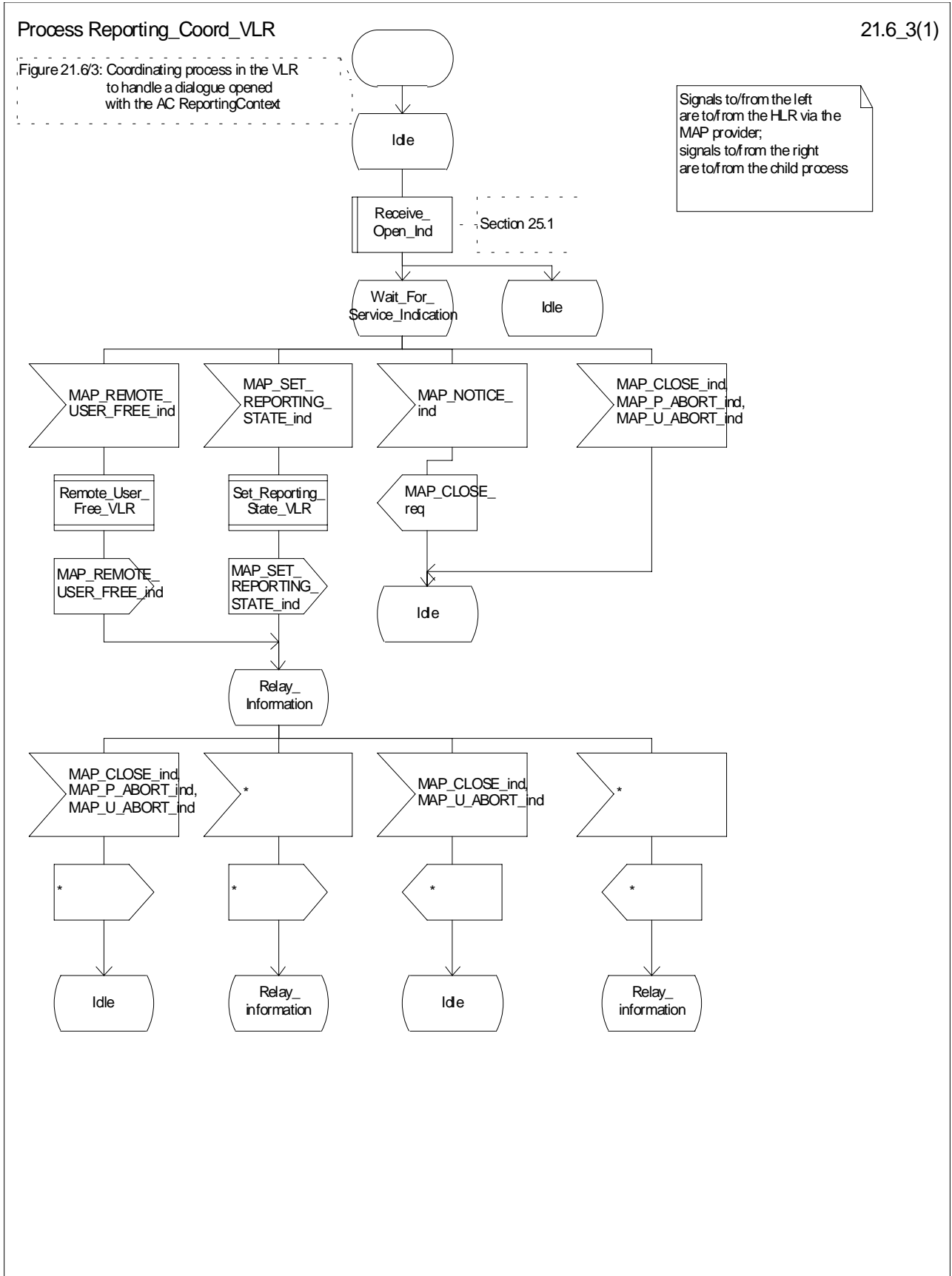


Figure 21.6/3: Process Reporting_Coord_VLR

21.6.4 Process in the VLR to set the reporting state

The MAP process in the VLR to set the reporting state is shown in figure 21.6/4.

The co-ordinator opens the process. The macro `Receive_Set_Reporting_State_VLR` handles the receipt of the request from the HLR, and the possible response from the CCBS application process in the VLR. When the macro exits, a MAP `CLOSE` is sent to the HLR and the process terminates.

The macro `Set_Reporting_State_VLR` is defined in figure 21.6/5.

When the VLR receives a `MAP_SET_REPORTING_STATE` service indication, it checks whether the required monitoring state is stopped.

If the required monitoring state is stopped, the MAP process sends a Stop Reporting message to the CCBS application in the VLR, sends a `MAP_SET_REPORTING_STATE` response to the HLR and exits from the macro.

If the required monitoring state is started, the MAP process sends a Start Reporting message to the CCBS application in the VLR and waits for a response.

If the CCBS application sends a Start Reporting ack, the MAP process sends a `MAP_SET_REPORTING_STATE` response to the HLR and exits from the macro.

If the CCBS application sends a Start Reporting negative response, the MAP process translates the negative response into a MAP user error, sends a `MAP_SET_REPORTING_STATE` response to the HLR and exits from the macro.

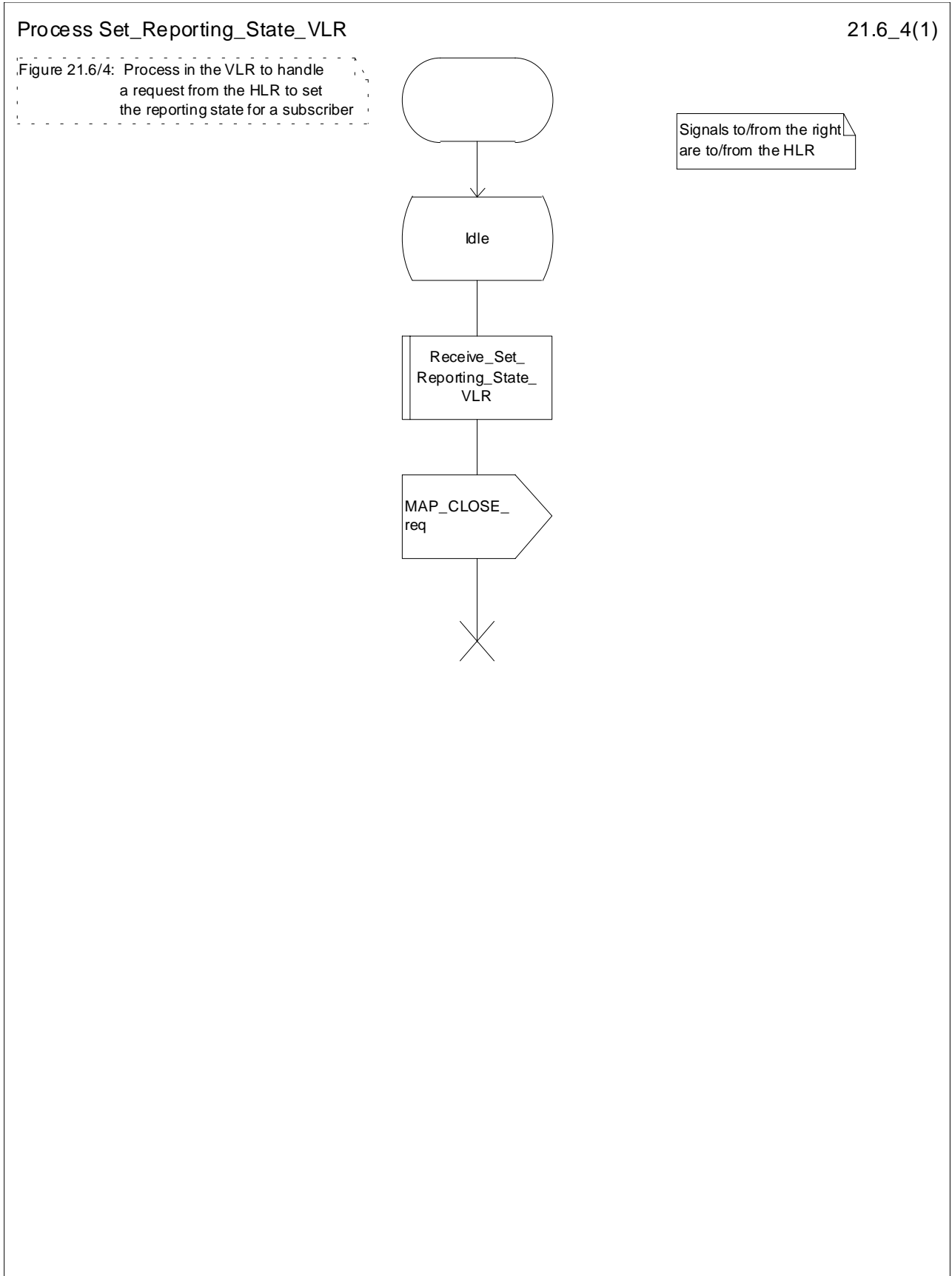


Figure 21.6/4: Process Set_Reporting_State_VLR

Macrodefinition Receive_Set_Reporting_State_VLR

21.6_5(1)

Figure 21.6/5: Macro in the VLR to handle a Set Reporting State instruction from the HLR

Signals to/from the left are to/from the CCBS application process in the VLR; signals to/from the right are to/from the HLR

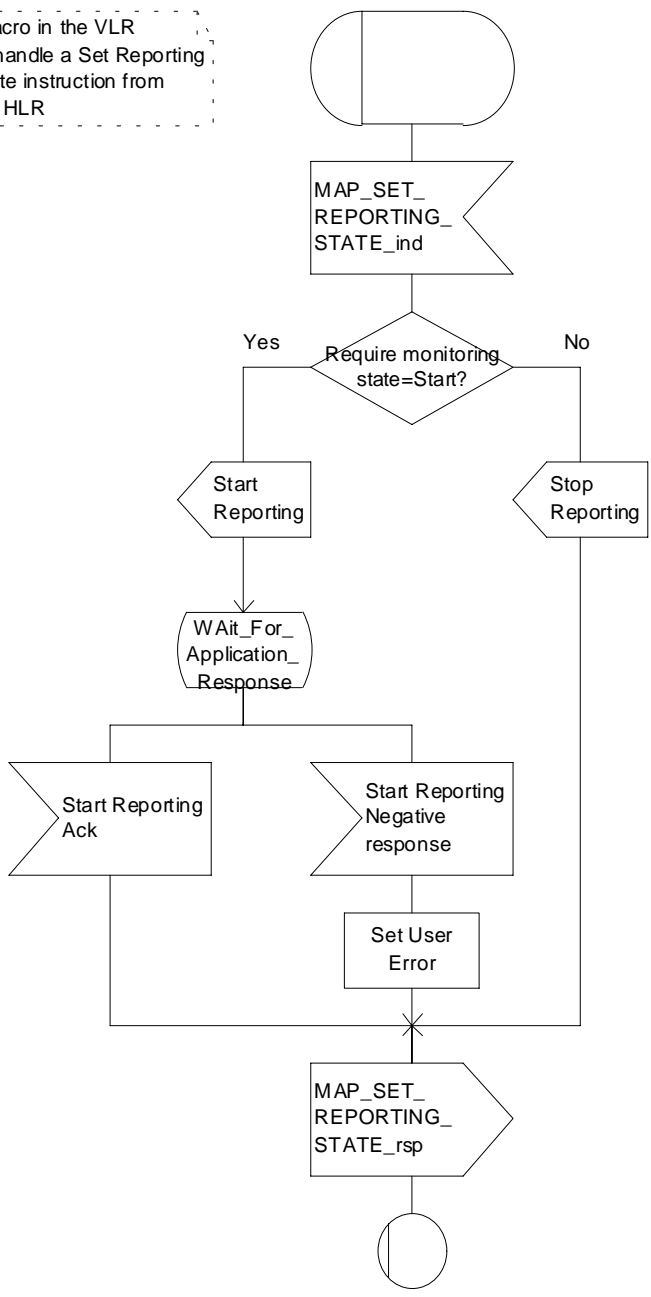


Figure 21.6/5: Macro Receive_Set_Reporting_State_VLR

21.7 Status Reporting

21.7.1 General

The message flows for reporting the status of a subscriber are shown in figures 21.7/1 and 21.7/2.

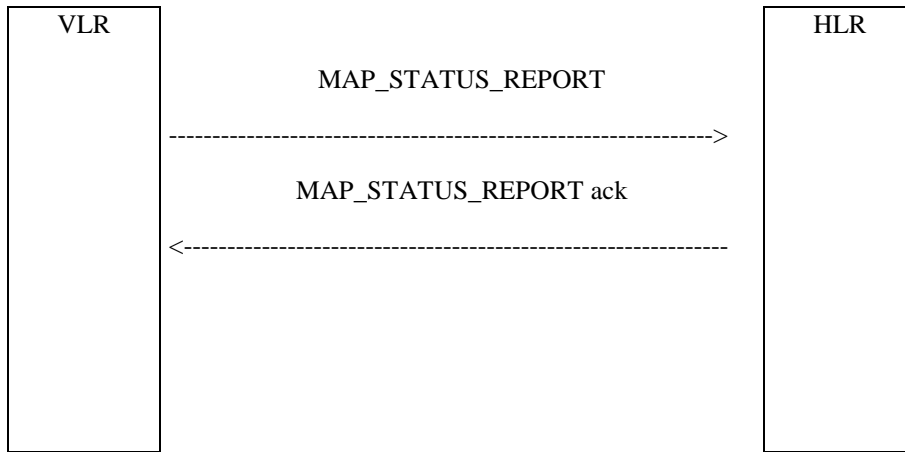


Figure 21.7/1: Status reporting, when monitoring continues in the VLR

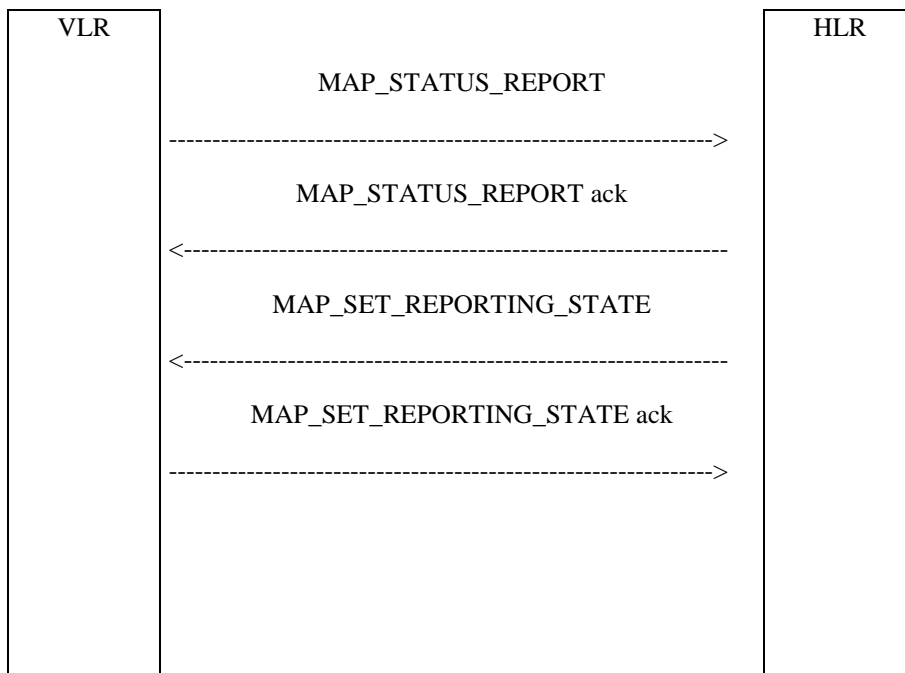


Figure 21.7/2: Status reporting, when monitoring stops

When the HLR sends a MAP_SET_REPORTING_STATE, it requests the stop of monitoring in the VLR.

21.7.2 Process in the VLR for Status Reporting

The MAP process in the VLR to send a status report to the HLR is shown in figure 21.7/3. The MAP process invokes macros not defined in this subclause; the definitions of these macros can be found as follows:

Receive_Open_Cnf	see subclause 25.1.2;
Check_Confirmation	see subclause 25.2.2.

Successful Outcome

When the MAP process receives a Event Report or CCBS Call Report from the CCBS application process in the VLR, it requests a dialogue with the HLR whose identity is contained in the request by sending a MAP_OPEN service request, and requests status report using a MAP_STATUS_REPORT service request. The VLR then invokes the macro Receive_Open_Cnf to wait for the response to the dialogue opening request. If the dialogue opening is successful, the MAP process waits for a response from the HLR.

If the MAP process receives a MAP_STATUS_REPORT service confirm from the HLR, the MAP process invokes the macro Check_Confirmation to check the content of the confirm.

If the macro Check_Confirmation takes the OK exit, the MAP process sends an Event Report ack or a CCBS Call Report ack containing the information received from the HLR to the CCBS application process in the VLR and waits for a possible instruction from the HLR to set the reporting state.

If the HLR requests the VLR to set a reporting state (in the macro Receive_Set_Reporting_State_VLR), the VLR closes the dialogue with the HLR by sending a MAP_CLOSE to the HLR.

If the HLR requires monitoring in the VLR to continue, it closes the dialogue by sending a MAP_CLOSE, and the MAP process in the VLR sends Continue Monitoring message to the CCBS application process in the VLR and returns to the idle state.

Failure of dialogue opening with the HLR

If the macro Receive_Open_Cnf takes the Vr exit or the Error exit, the MAP process sends a Event Report negative response or CCBS Call Report negative response to the CCBS application process in the VLR and returns to the idle state.

Error in MAP_STATUS_REPORT confirm

If the MAP_STATUS_REPORT service confirm contains a user error or a provider error, or the macro Check_Confirmation indicates that there is a data error, the MAP process sends an Event Report negative response or CCBS Call Report negative response to the CCBS application process in the VLR and returns to the idle state.

Abort of HLR dialogue in State Wait_For_HLR_Response

After the dialogue with the HLR has been established, the MAP service provider may abort the dialogue by issuing a MAP_P_ABORT or a MAP_U_ABORT indication. In this case, the MAP process sends a Event Report or CCBS Call Report negative response to the CCBS application process in the VLR and returns to the idle state.

If the MAP provider indicates a protocol problem by sending a MAP_NOTICE indication, the MAP process closes the dialogue with the HLR. The VLR sends an Event Report negative response or CCBS Call Report negative response indicating system failure to the CCBS application process in the VLR and returns to the idle state.

Abort of HLR dialogue in State Wait_For_Set_Reporting

After the dialogue with the HLR has been established, the MAP service provider may abort the dialogue by issuing a MAP_P_ABORT or a MAP_U_ABORT indication. In this case, the VLR returns to the idle state

If the MAP provider indicates a protocol problem by sending a MAP_NOTICE indication, the MAP process closes the dialogue with the HLR and returns to the idle state.

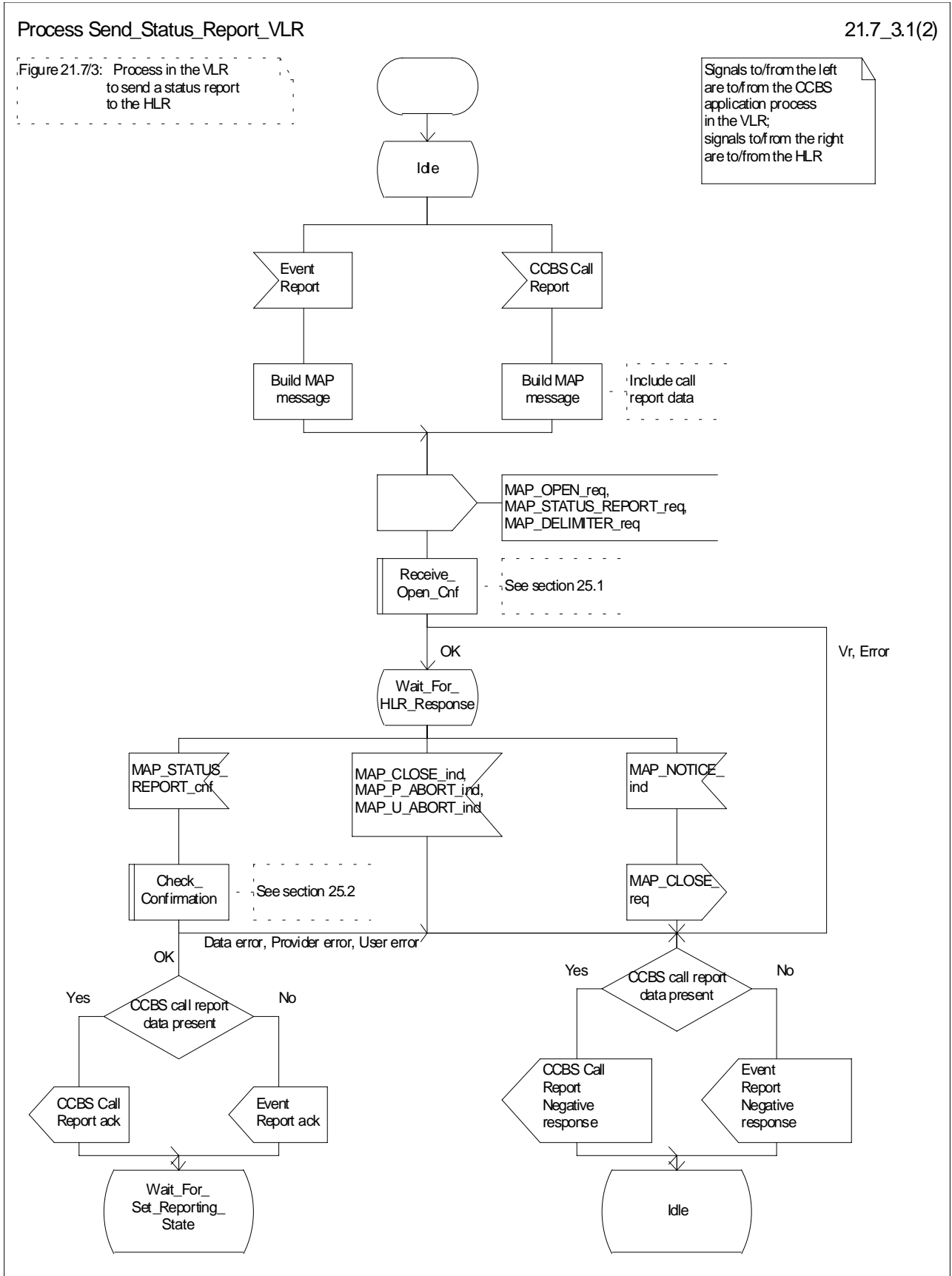


Figure 21.7/3 (sheet 1 of 2): Process Send_Status_Report_VLR

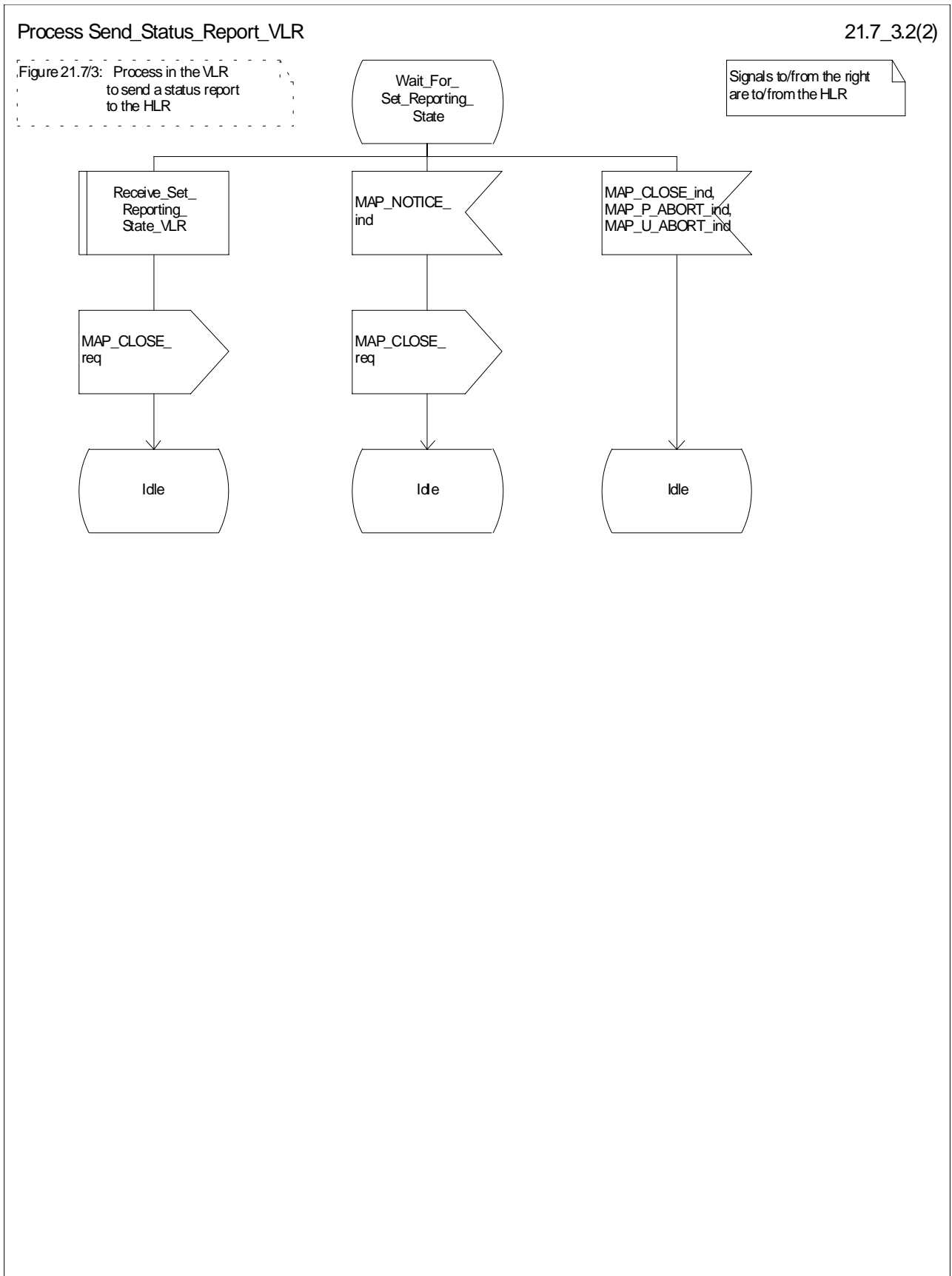


Figure 21.7/3 (sheet 2 of 2): Process Send_Status_Report_VLR

21.7.3 Process in the HLR for Status Reporting

The MAP process in the HLR to handle a status report is shown in figure 21.7/4. The MAP process invokes macros not defined in this subclause; the definitions of these macros can be found as follows:

Receive_Open_Ind	see subclause 25.1.1;
Check_Confirmation	see subclause 25.2.2;

Successful outcome

When the MAP process receives a MAP_OPEN indication with the application context reporting, it checks it by invoking the macro Receive_Open_Ind.

If the macro takes the OK exit, the MAP process waits for a service indication.

The MAP process invokes the macro Receive_Status_Report_HLR to handle a MAP_STATUS_REPORT service indication; this macro is defined in figure 21.7/5. The MAP process then waits for a response from the CCBS application in the HLR.

If the MAP process receives a Stop Reporting message from the CCBS process, it sets the required monitoring state to stop, and may send a MAP_DELIMITER service request to the VLR. The HLR then invokes the macro Set_Reporting_State_HLR. After exiting the macro, the MAP process returns to the idle state.

If the MAP process receives a Continue Reporting from the CCBS process, it sends a MAP_CLOSE Request to VLR and returns to the idle state.

Failure of dialogue opening with the VLR

If the macro Receive_Open_Ind takes the Vr exit or the Error exit, the MAP process returns to the idle state.

Abort of VLR dialogue in State Wait_For_Service_Indication

After the dialogue with the HLR has been established, the MAP service provider may abort the dialogue by issuing a MAP_P_ABORT indication. In this case, the MAP process returns to the idle state.

If the MAP provider indicates a protocol problem by sending a MAP_NOTICE indication, the MAP process closes the dialogue with the VLR and returns to the idle state.

Macro Receive_Status_Report_HLR

The macro Receive_Status_Report_HLR is shown in figure 21.7/5.

When a MAP_STATUS_REPORT service indication is received, the HLR checks whether call report data are present.

If call report data are present, the MAP process sends a CCBS Call Report message to the CCBS application process in the HLR and waits for a response; otherwise it sends an Event Report message to the CCBS application process in the HLR and waits for a response.

If the MAP process receives a CCBS Call Report ack or Event Report ack from the CCBS application process in the HLR, it sends a MAP_STATUS_REPORT service confirm to the VLR and exits from the macro.

If the MAP process receives a CCBS Call Report negative response or Event Report negative response from the CCBS application process in the HLR, it sets the User Error according to the negative response, sends a MAP_STATUS_REPORT service confirm to the VLR and exits from the macro.

Macro Set_Reporting_State_HLR

The macro Set_Reporting_State_HLR is shown in figure 21.7/6.

The MAP process in the HLR sends a MAP_SET_REPORTING_STATE service request to the VLR and waits for a response.

If the MAP process receives a MAP_SET_REPORTING_STATE service confirm from the VLR, it invokes the macro Check_Confirmation to check the content of the confirm.

If the macro Check_Confirmation takes the OK exit, the macro Set_Reporting_State_HLR takes the OK exit.

If the macro Check_Confirmation takes the Data error, Provider error or User error exit, the macro Set_Reporting_State_HLR takes the Error exit.

While the MAP process is waiting for a response from the VLR, the MAP provider may terminate the dialogue by sending a MAP_CLOSE, MAP_P_ABORT or MAP_U_ABORT. In this case the macro takes the Aborted exit.

If the MAP provider indicates a protocol problem by sending a MAP_NOTICE indication, the MAP process closes the dialogue with the VLR and the macro takes the Aborted exit.

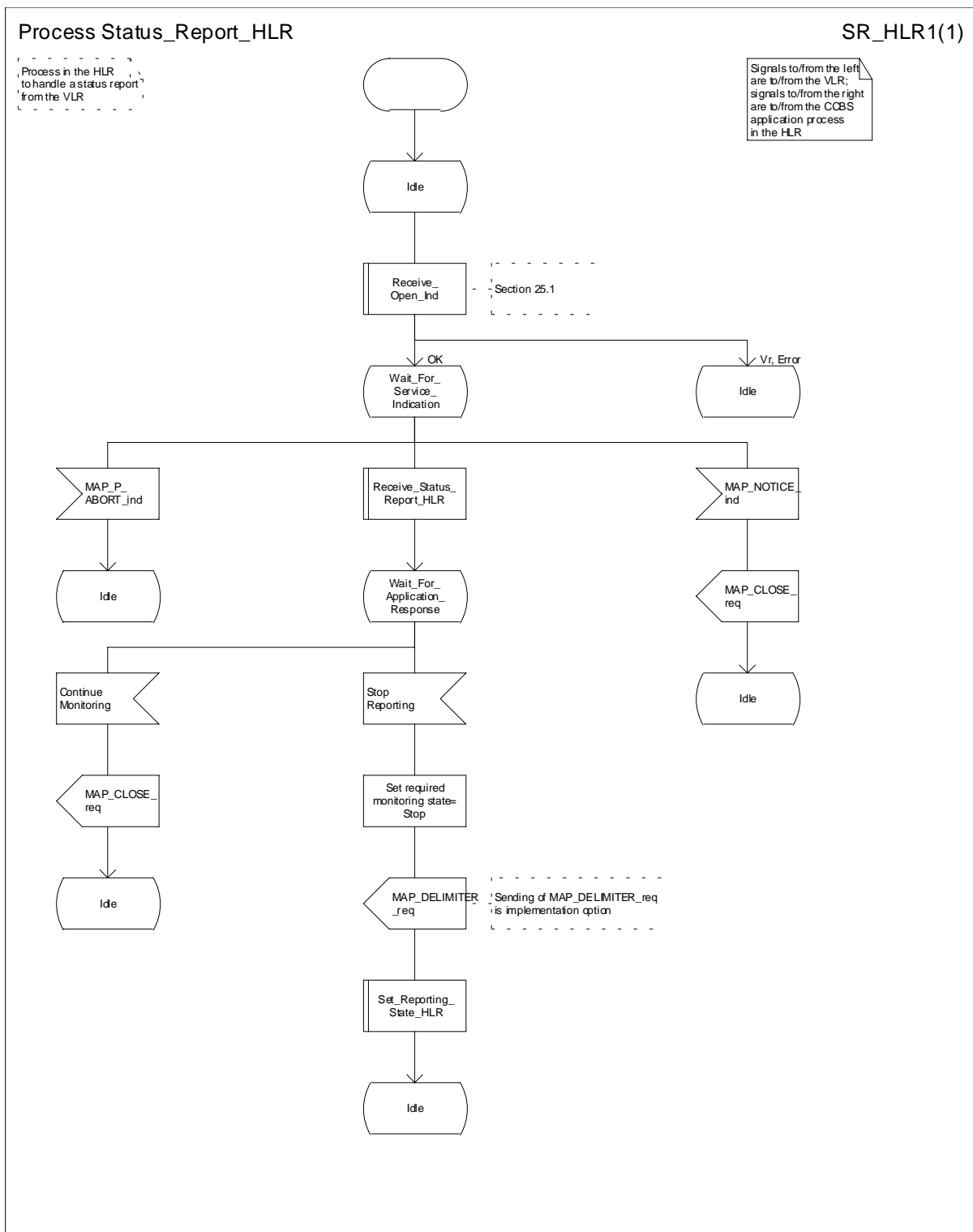


Figure 21.7/4: Process Status Report_HLR

Macrodefinition Receive_Status_Report_HLR

21.7_5(1)

Figure 21.7/5: Macro in the HLR to receive a status report from the VLR

Signals to/from the left are to/from the VLR; signals to/from the right are to/from the CCBS application process in the HLR

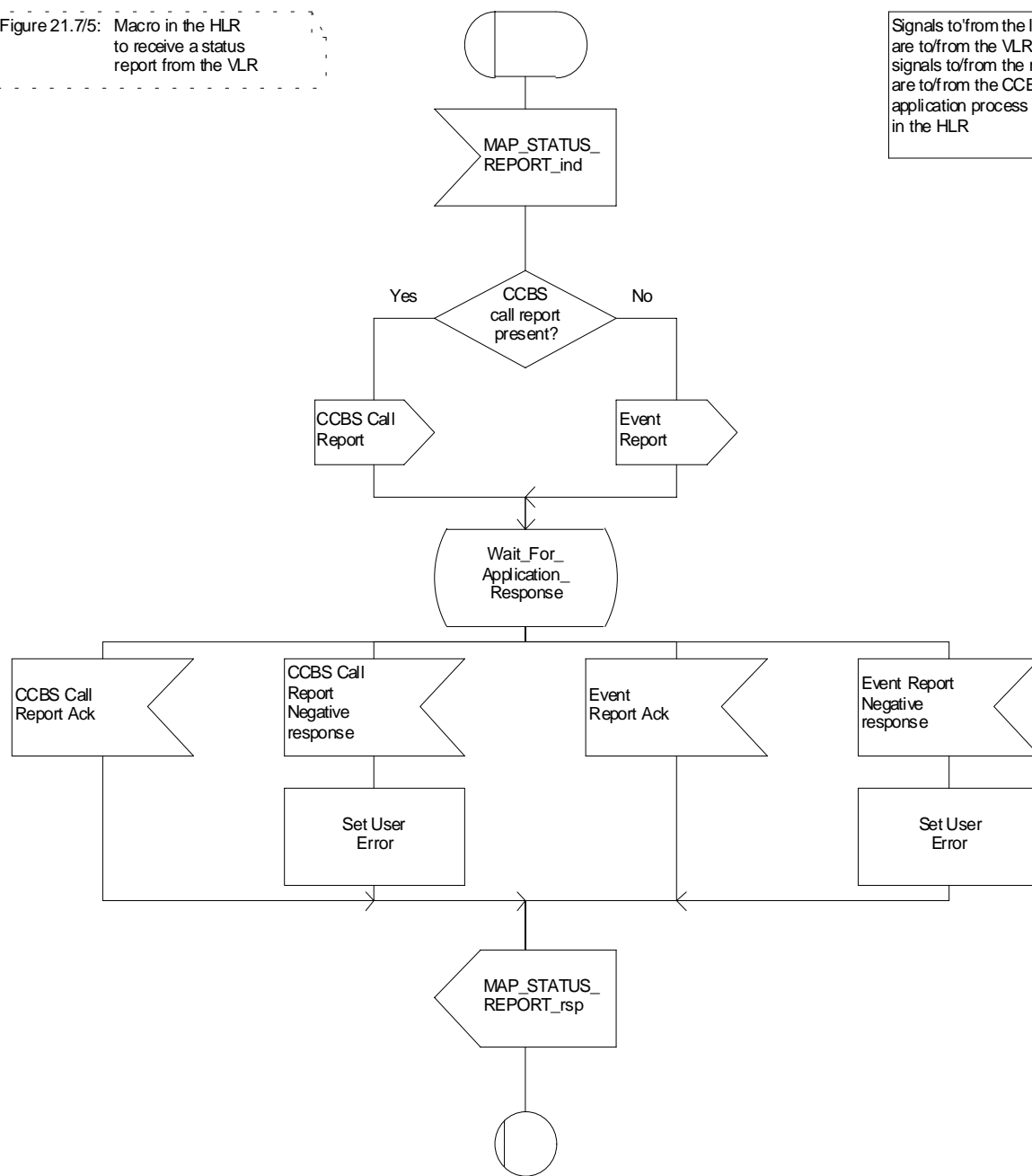


Figure 21.7/5: Macro Receive_Status_Report_HLR

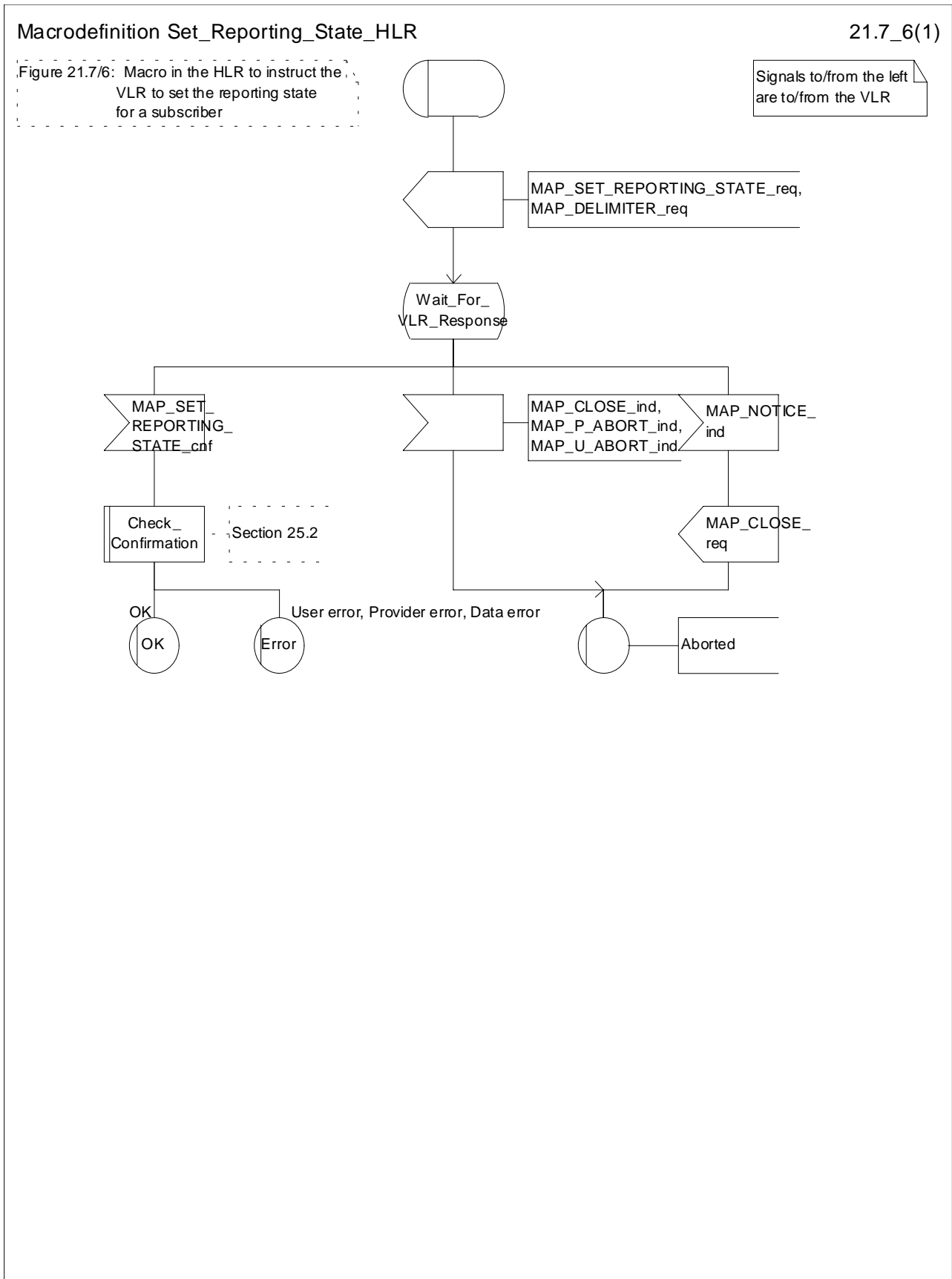


Figure 21.7/6: Macro Set_Reporting_State_HLR

21.8 Remote User Free

21.8.1 General

The message flows for handling remote user free are shown in figures 21.8/1 and 21.8/2.

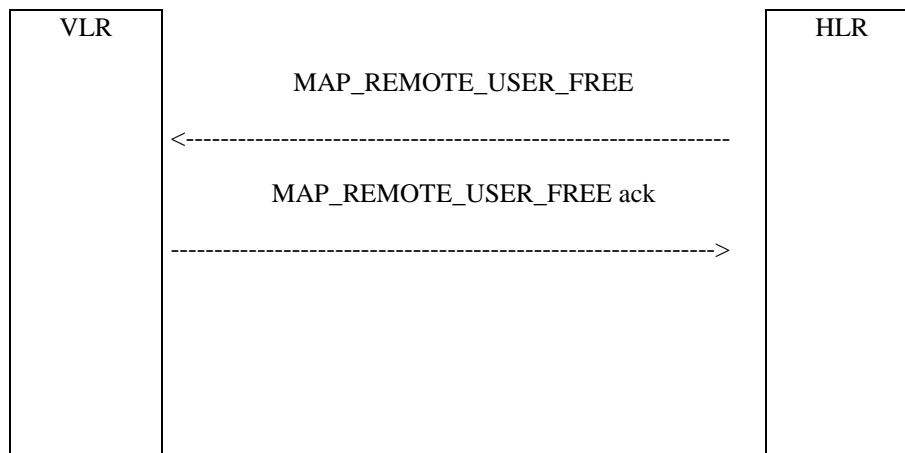


Figure 21.8/1: Remote User Free: recall not accepted

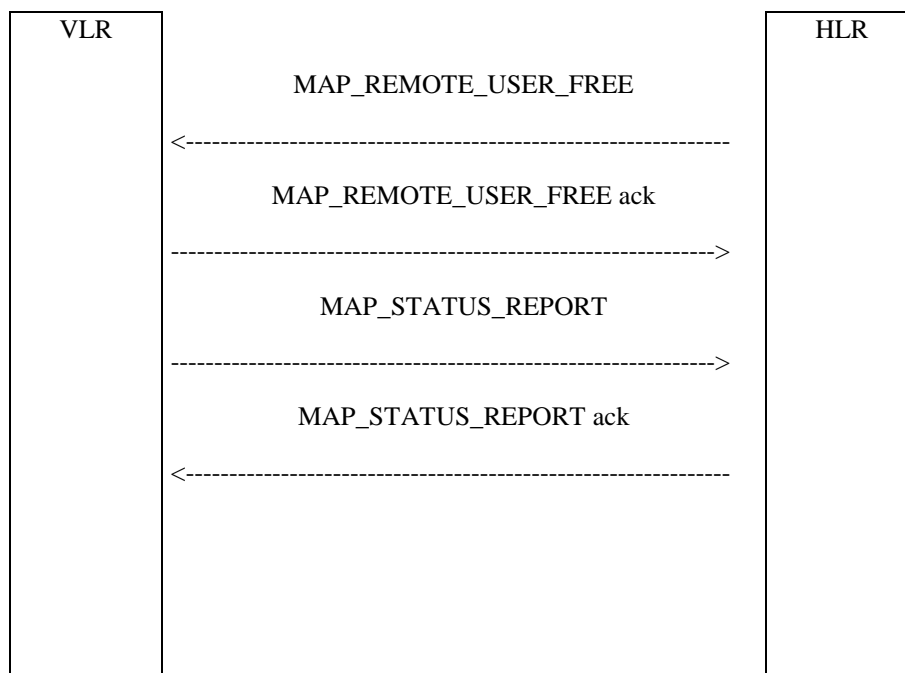


Figure 21.8/2: Remote User Free: recall accepted

21.8.2 Process in the HLR for Remote User Free

The MAP process in the HLR to handle Remote User Free is shown in figure 21.8/3. The MAP process invokes macros not defined in this subclause; the definitions of these macros can be found as follows:

- Receive_Open_Cnf see subclause 25.1.2;
- Check_Confirmation see subclause 25.2.2;

Successful Outcome

When the MAP process receives a CCBS RUF request from the CCBS application process in the HLR, it requests a dialogue with the VLR whose identity is contained in the request by sending a MAP_OPEN service request and sending the necessary information using a MAP_REMOTE_USER_FREE service request. The HLR then invokes the macro Receive_Open_Cnf to wait for the response to the dialogue opening request. If the dialogue opening is successful, the MAP process waits for a response from the VLR.

If the MAP process receives a MAP_REMOTE_USER_FREE service confirm from the VLR, the MAP process invokes the macro Check_Confirmation to check the content of the confirm.

If the macro Check_Confirmation takes the OK exit, the MAP process sends a CCBS RUF ack containing the information received from the VLR to the CCBS application process in the HLR and waits for a MAP_STATUS_REPORT service indication from the VLR. If in this state a MAP_CLOSE service indication is received, the MAP process returns to the idle state. If in this state a MAP_STATUS_REPORT service indication is received, further processing is described by the macro Receive_Status_Report_HLR (described in subclause 21.7.3). When the macro exits, the MAP process constructs a MAP_CLOSE service request, sends it to the VLR and returns to the idle state.

Failure of dialogue opening with the VLR

If the macro Receive_Open_Cnf takes the Vr exit or the Error exit, the MAP process sends a negative response to the CCBS application process in the HLR and returns to the idle state.

Error in MAP_REMOTE_USER_FREE confirm

If the MAP_REMOTE_USER_FREE service confirm contains a user error or a provider error, or the macro Check_Confirmation indicates that there is a data error, the MAP process sends a CCBS RUF negative response to the CCBS application process in the HLR and returns to the idle state.

Abort of VLR dialogue

When the MAP process is waiting for a VLR response to the MAP_REMOTE_USER_FREE, the MAP service provider may abort the dialogue by issuing a MAP_CLOSE, a MAP_P_ABORT or a MAP_U_ABORT indication. In this case, the MAP process sends a CCBS RUF negative response to the CCBS application process in the HLR and returns to the idle state.

If the MAP provider indicates a protocol problem by sending a MAP_NOTICE indication when the MAP process is waiting for a VLR response to the MAP_REMOTE_USER_FREE, the MAP process closes the dialogue with the VLR, sends a CCBS RUF negative response indicating system failure to the CCBS application process in the HLR and returns to the idle state.

When the MAP process is waiting for a possible MAP_STATUS_REPORT from the VLR, the MAP service provider may abort the dialogue by issuing a MAP_P_ABORT or a MAP_U_ABORT indication. In this case, the MAP process returns to the idle state.

If the MAP provider indicates a protocol problem by sending a MAP_NOTICE indication when the MAP process is waiting for a possible MAP_STATUS_REPORT from the VLR, the MAP process closes the dialogue with the VLR and returns to the idle state.

If the CCBS application in the HLR decides to abort the dialogue, it sends an Abort message to the MAP process, which closes the dialogue with the VLR and returns to the idle state.

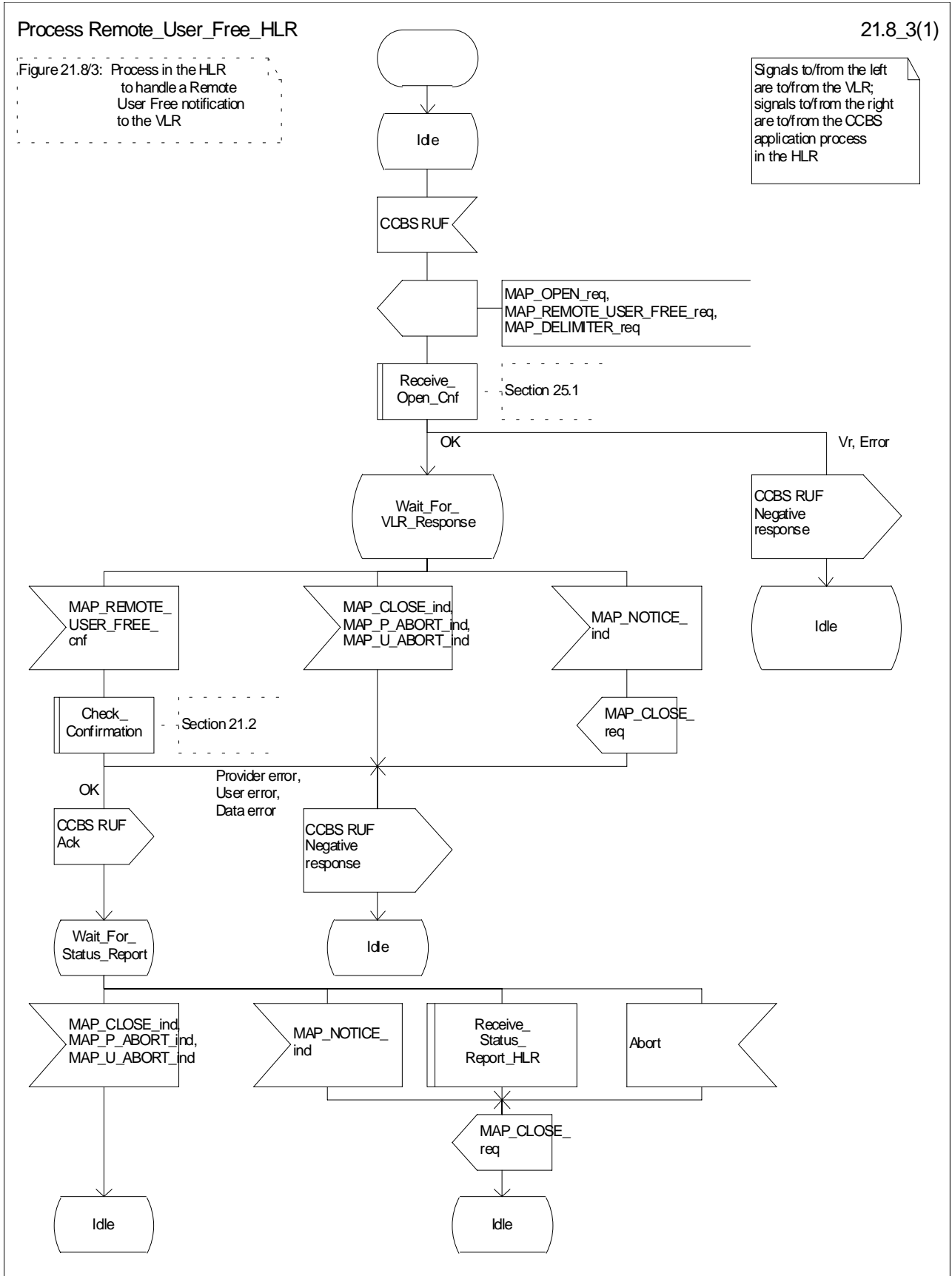


Figure 21.8/3: Process Remote_User_Free_HLR

21.8.3 Process in the VLR for Remote User Free

The MAP process in the VLR to handle Remote User Free is shown in figure 21.8/4. The MAP process invokes a macro not defined in this subclause; the definitions of this macro can be found as follows:

Check_Confirmation see subclause 25.2.2;

Successful outcome (Recall accepted)

When the MAP process receives a MAP_REMOTE_USER_FREE service indication, the VLR sends a CCBS RUF request to the CCBS application process in the VLR, and waits for a response. The request contains the parameters received in the MAP_REMOTE_USER_FREE service indication.

If the CCBS application process in the VLR returns a positive response indicating "recall accepted", the MAP process constructs a MAP_REMOTE_USER_FREE service response and a MAP_DELIMITER service request, sends them to the VLR and waits for a CCBS Call Report message from the CCBS application process in the VLR. When the MAP process receives the CCBS Call Report from the CCBS application process in the VLR, it constructs a MAP_STATUS_REPORT service request and a MAP_DELIMITER service request, sends them to the HLR and waits for a response. If the MAP process receives a MAP_STATUS_REPORT service confirm, the VLR calls the macro Check_Confirmation. If this macro takes the OK exit, the MAP process sends a CCBS Call Report ack to the CCBS application process in the VLR and the MAP process terminates.

Successful outcome (Recall not accepted)

If the CCBS application process in the VLR returns a positive response indicating "recall not accepted", the MAP process constructs a MAP_REMOTE_USER_FREE service response and a MAP_CLOSE service request, sends them to the HLR and terminates.

Negative response from VLR CCBS application process

If the CCBS application process in the VLR returns a negative response, the MAP process constructs a MAP_REMOTE_USER_FREE service response containing the appropriate error and a MAP_CLOSE service request, sends them to the HLR and terminates.

Failure of dialogue with the HLR

When waiting for a response or a call result from the CCBS application process in the VLR, the MAP process may receive a MAP_CLOSE service indication, a MAP_U_ABORT service indication or a MAP_P_ABORT service indication from the co-ordinating process, in which case the MAP process terminates.

When waiting for a call result from the CCBS application process in the VLR, the MAP process may receive a MAP_NOTICE indication from the co-ordinating process, in which case the MAP process constructs a MAP_CLOSE service request, sends it to the co-ordinating process and terminates.

When waiting for a response from the HLR, the MAP process may receive a MAP_CLOSE indication, a MAP_U_ABORT indication or a MAP_P_ABORT indication from the co-ordinating process, in which case the MAP process sends a CCBS Call Report negative response to the CCBS application process in the VLR and terminates.

When waiting for a response from the HLR, the MAP process may receive a MAP_NOTICE indication from the co-ordinating process, in which case the MAP process constructs a MAP_CLOSE service request, sends it to the co-ordinating process, sends a CCBS Call Report negative response to the CCBS application process in the VLR and terminates.

Error in MAP_STATUS_REPORT confirm

If the MAP_STATUS_REPORT service confirm contains a user error or a provider error, the MAP process sends a CCBS Call Report negative response to the CCBS application process in the VLR and terminates.

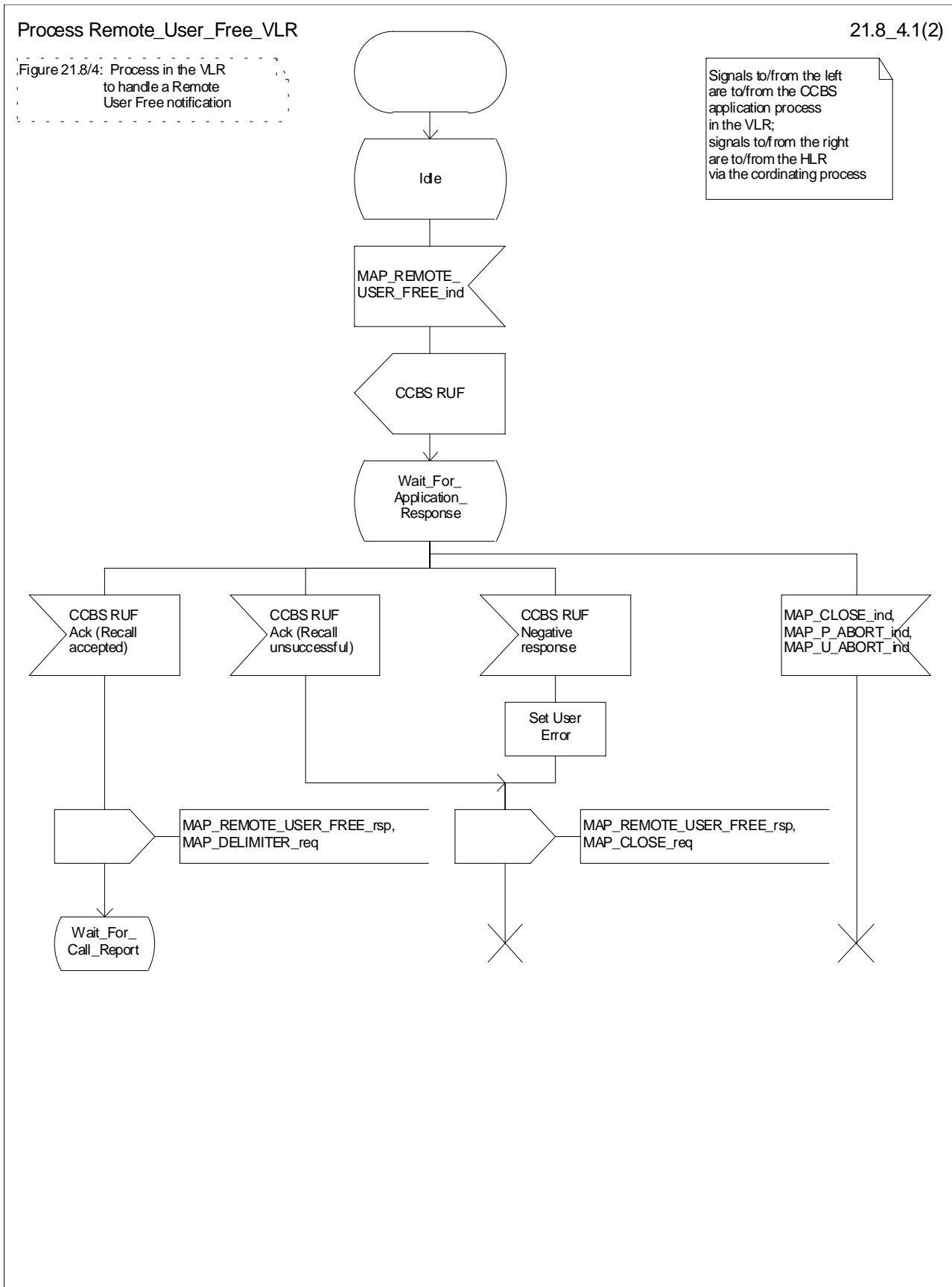


Figure 21.8/4 (sheet 1 of 2): Process Remote_User_Free_VLR

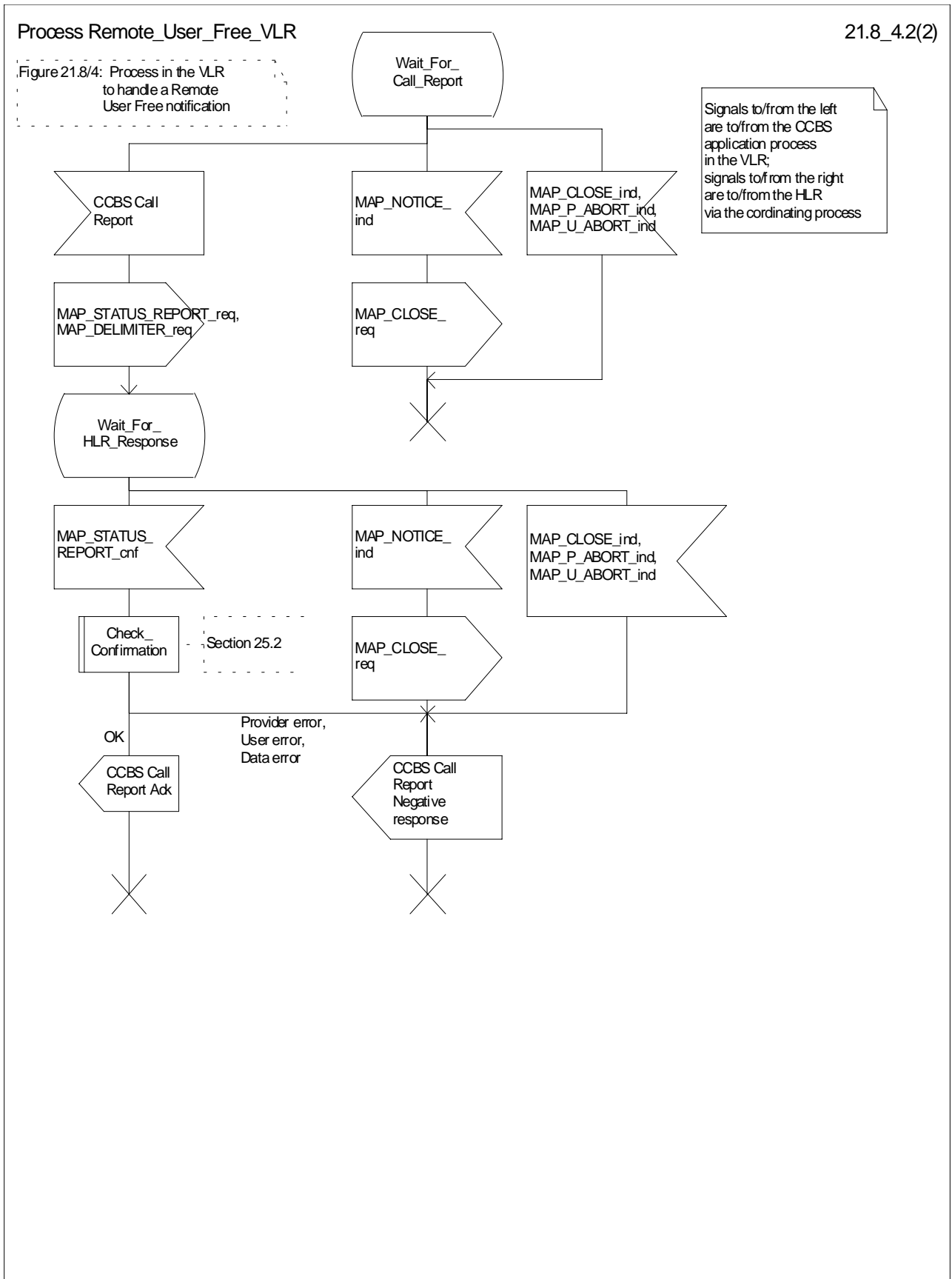


Figure 21.8/4 (sheet 2 of 2): Process Remote_User_Free_VLR

21.9 IST Alert

21.9.1 General

The Immediate Service Termination Alert procedure is used to keep track of the call activities performed by IST subscribers and, eventually, to terminate the alerted call activities, or all the call activities related to the alerted subscriber.

The message flow for alerting and terminating the call(s) is shown in figure 21.9/1, where the MSC may be a Visited MSC or a Gateway MSC.

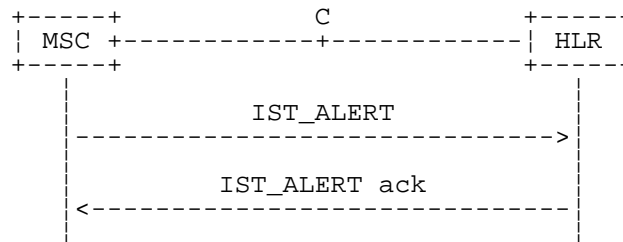


Figure 21.9/1: IST Alert

21.9.2 Procedure in the MSC

The MAP process in the MSC (Visited MSC or Gateway MSC) is shown in figure 21.9/2. The MAP process invokes macros not defined in this subclause; the definitions of these macros can be found as follows:

Receive_Open_Cnf see subclause 25.1.2;

Check_Confirmation see subclause 25.2.2;

Successful Outcome

When the MAP process receives an IST Alert request from a call handling process in the Visited MSC or Gateway MSC, it requests a dialogue with the HLR that the subscriber belongs to, by sending a MAP_OPEN service request, a MAP_IST_ALERT service request, and a MAP_DELIMITER service request. The MSC then invokes the macro Receive_Open_Cnf to wait for the response to the dialogue opening request. If the dialogue opening is successful, the MAP process waits for a response from the HLR.

If the MAP process receives a MAP_IST_ALERT service confirm from the HLR, the MAP process invokes the macro Check_Confirmation to check the content of the confirm.

If the macro Check_Confirmation takes the OK exit, the MAP process sends a IST Alert ack containing the information received from the HLR to the call handling process in the MSC and returns to the idle state.

Failure of dialogue opening with the HLR

If the macro Receive_Open_Cnf takes the Vr exit or the Error exit, the MAP process sends a negative response to the call handling process in the MSC, and returns to the idle state.

Error in MAP_IST_ALERT confirm

If the MAP_IST_ALERT service confirm contains a user error or a provider error, or the macro Check_Confirmation indicates that there is a data error, the MAP process sends a negative response to the call handling process in the MSC, and returns to the idle state.

Abort of HLR dialogue

When the MAP process is waiting for an HLR response to the MAP_IST_ALERT, the MAP service provider may abort the dialogue by issuing a MAP_CLOSE, a MAP_P_ABORT or a MAP_U_ABORT indication. In this case, the MAP process sends a negative response to the call handling process in the MSC, and returns to the idle state.

If the MAP provider indicates a protocol problem by sending a MAP_NOTICE indication when the MAP process is waiting for an HLR response to the MAP_IST_ALERT, the MAP process closes the dialogue with the HLR, sends a negative response to the call handling process in the MSC, and returns to the idle state.

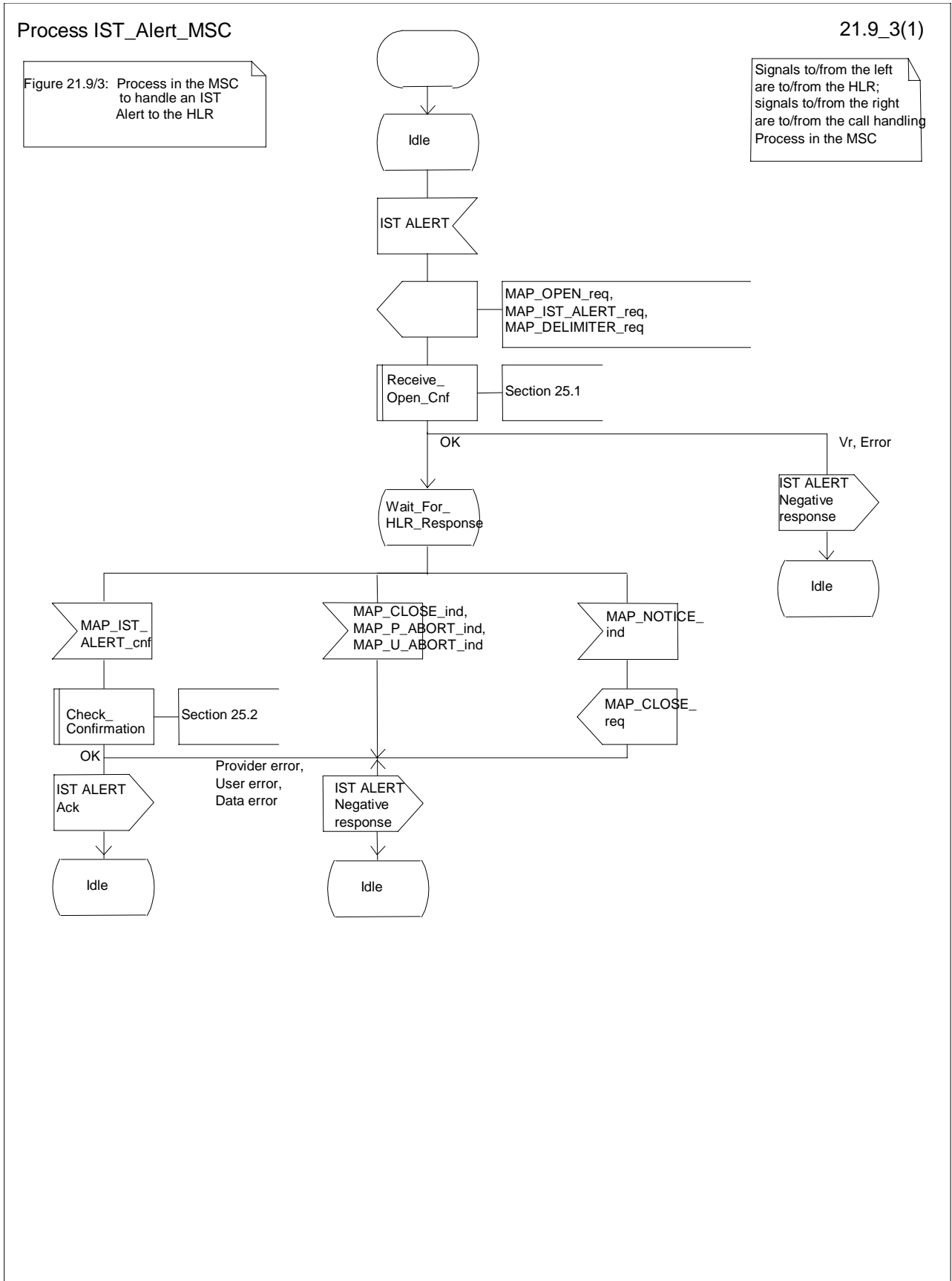


Figure 21.9/2: Process IST_Alert_MSC

21.9.3 Procedure in the HLR

The MAP process in the HLR is shown in figure 21.9/3. The MAP process invokes a macro not defined in this subclause; the definition of this macro can be found as follows:

Receive_Open_Ind see subclause 25.1.1;

Successful outcome

When the MAP process in the HLR receives a request to open a dialogue, it invokes the macro Receive_Open_Ind to check if the dialogue can be opened.

If the dialogue can be opened, and the service indication received is a MAP_IST_ALERT, the HLR then sends the IST alert indication to the call handling process in the HLR, and waits for a response.

If the call handling process in the HLR returns a positive response, the MAP process constructs a MAP_IST_ALERT service response and a MAP_CLOSE service request, sends them to the MSC, and returns to the idle state.

Negative response from HLR call handling process

If the call handling process in the HLR returns a negative response, the MAP process constructs a MAP_IST_ALERT service response containing the appropriate error and a MAP_CLOSE service request, sends them to the MSC and returns to the idle state.

Failure of dialogue opening in the HLR

If the macro Receive_Open_Ind takes the Vr exit or the Error exit, the MAP process returns to the idle state.

Abort of MSC dialogue

If the MAP process receives a MAP_P_ABORT indication before receiving a service indication, the MAP process returns to the idle state.

When the MAP process receives a MAP_NOTICE indication before receiving a service indication, the MAP process closes the dialogue with the MSC, and returns to the idle state.

When the MAP process is waiting for the application response to the IST Alert, the MAP service provider may abort the dialogue by issuing a MAP_CLOSE, a MAP_P_ABORT or a MAP_U_ABORT indication. In this case, the MAP process returns to the idle state.

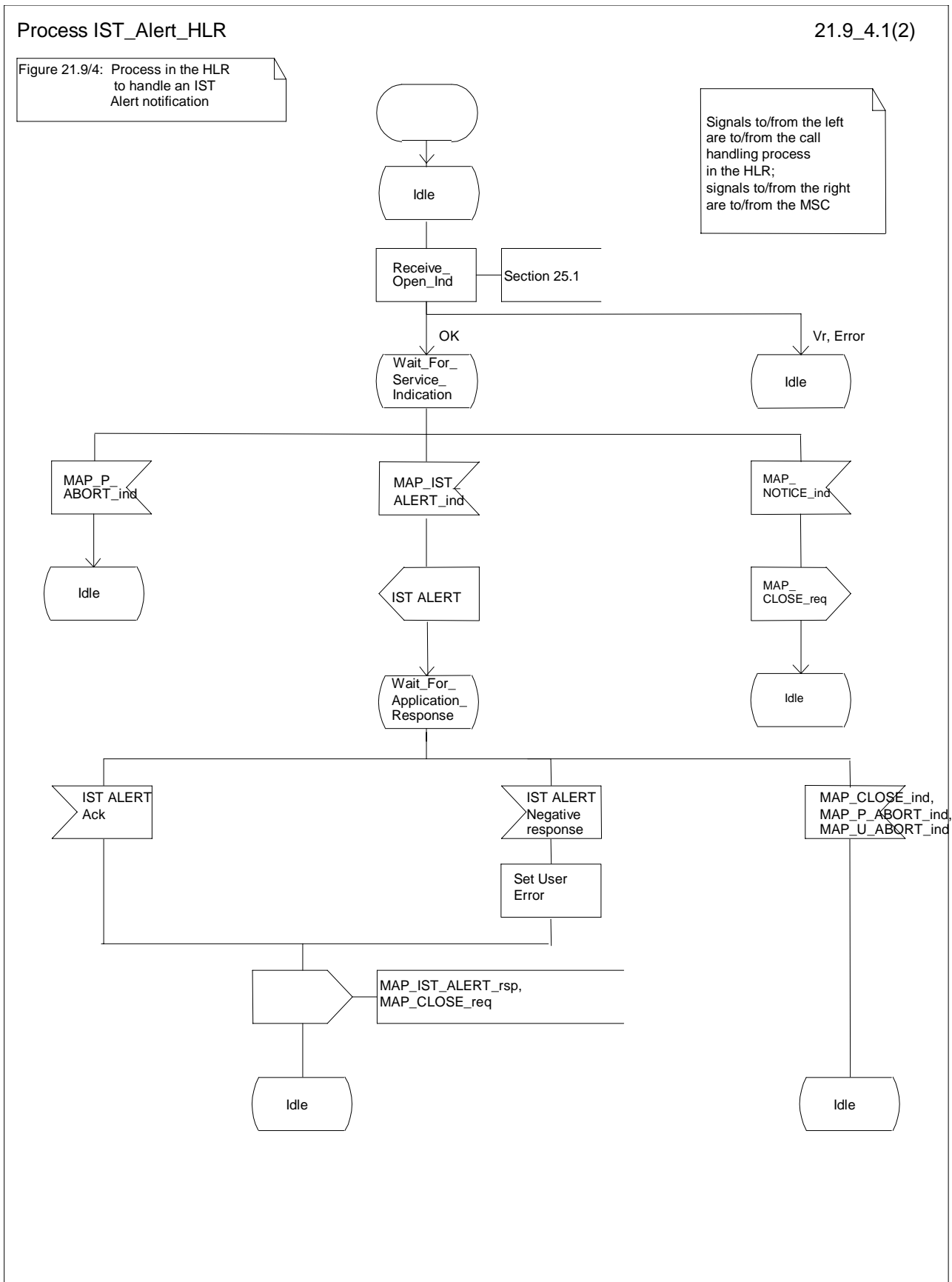


Figure 21.9/3: Process IST_Alert_HLR

21.10 IST Command

21.10.1 General

The Immediate Service Termination Command procedure is used to terminate the call activities related to a subscriber.

The message flow for the IST Command service is shown in figure 21.10/1, where the MSC may be a Visited MSC or a Gateway MSC.

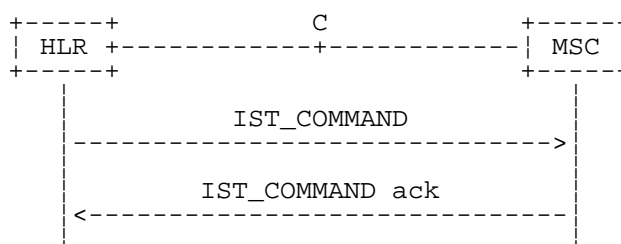


Figure 21.10/1: IST Command

21.10.2 Procedure in the HLR

The MAP process in the HLR is shown in figure 21.10/2. The MAP process invokes macros not defined in this subclause; the definitions of these macros can be found as follows:

Receive_Open_Cnf see subclause 25.1.2;

Check_Confirmation see subclause 25.2.2.

Successful Outcome

When the MAP process receives an IST Command request, it requests a dialogue with the MSC (Gateway MSC or Visited MSC), by sending a MAP_OPEN service request, a MAP_IST_COMMAND service request, and a MAP_DELIMITER service request. The HLR then invokes the macro Receive_Open_Cnf to wait for the response to the dialogue opening request. If the dialogue opening is successful, the MAP process waits for a response from the MSC.

If the MAP process receives a MAP_IST_COMMAND service confirm from the MSC, the MAP process invokes the macro Check_Confirmation to check the content of the confirm.

If the macro Check_Confirmation takes the OK exit, the MAP process sends a IST Command ack containing the information received from the MSC to the call handling process in the HLR and returns to the idle state.

Failure of dialogue opening with the HLR

If the macro Receive_Open_Cnf takes the Vr exit or the Error exit, the MAP process sends a negative response to the call handling process in the HLR, and returns to the idle state.

Error in MAP_IST_COMMAND confirm

If the MAP_IST_COMMAND service confirm contains a user error or a provider error, or the macro Check_Confirmation indicates that there is a data error, the MAP process sends a negative response to the call handling process in the HLR, and returns to the idle state.

Abort of MSC dialogue

When the MAP process is waiting for an MSC response to the MAP_IST_COMMAND, the MAP service provider may abort the dialogue by issuing a MAP_CLOSE, a MAP_P_ABORT or a MAP_U_ABORT indication. In this case, the MAP process sends a negative response to the call handling process in the HLR, and returns to the idle state.

If the MAP provider indicates a protocol problem by sending a MAP_NOTICE indication when the MAP process is waiting for an MSC response to the MAP_IST_COMMAND, the MAP process closes the dialogue with the MSC, sends a negative response to the call handling process in the HLR, and returns to the idle state.

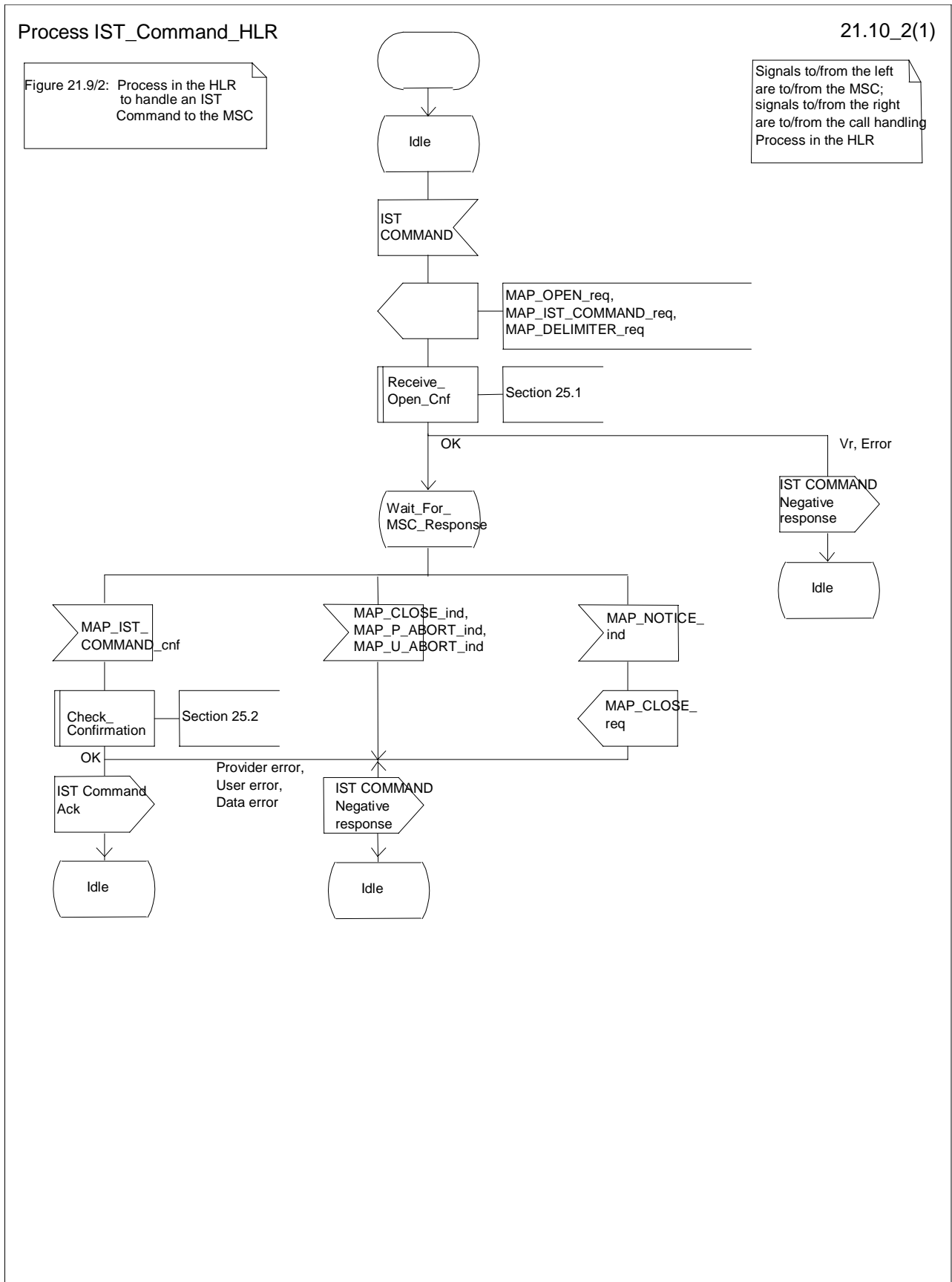


Figure 21.10/2: Process IST_Command_HLR

21.10.3 Procedure in the MSC

The MAP process in the MSC is shown in figure 21.10/3. The MAP process invokes a macro not defined in this subclause; the definition of this macro can be found as follows:

Receive_Open_Ind see subclause 25.1.1;

Successful outcome

When the MAP process in the MSC receives a request to open a dialogue, it invokes the macro Receive_Open_Ind to check if the dialogue can be opened.

If the dialogue can be opened, and the service indication received is a MAP_IST_COMMAND, the MSC then sends the IST command indication to the call handling process in the MSC, and waits for a response.

If the call handling process in the MSC returns a positive response, the MAP process constructs a MAP_IST_COMMAND service response and a MAP_CLOSE service request, sends them to the HLR, and returns to the idle state.

Negative response from MSC call handling process

If the call handling process in the MSC returns a negative response, the MAP process constructs a MAP_IST_COMMAND service response containing the appropriate error and a MAP_CLOSE service request, sends them to the HLR and returns to the idle state.

Failure of dialogue opening in the MSC

If the macro Receive_Open_Ind takes the Vr exit or the Error exit, the MAP process returns to the idle state.

Abort of HLR dialogue

If the MAP process receives a MAP_P_ABORT indication before receiving a service indication, the MAP process returns to the idle state.

When the MAP process receives a MAP_NOTICE indication before receiving a service indication, the MAP process closes the dialogue with the HLR, and returns to the idle state.

When the MAP process is waiting for the application response to the IST Command, the MAP service provider may abort the dialogue by issuing a MAP_CLOSE, a MAP_P_ABORT or a MAP_U_ABORT indication. In this case, the MAP process returns to the idle state.

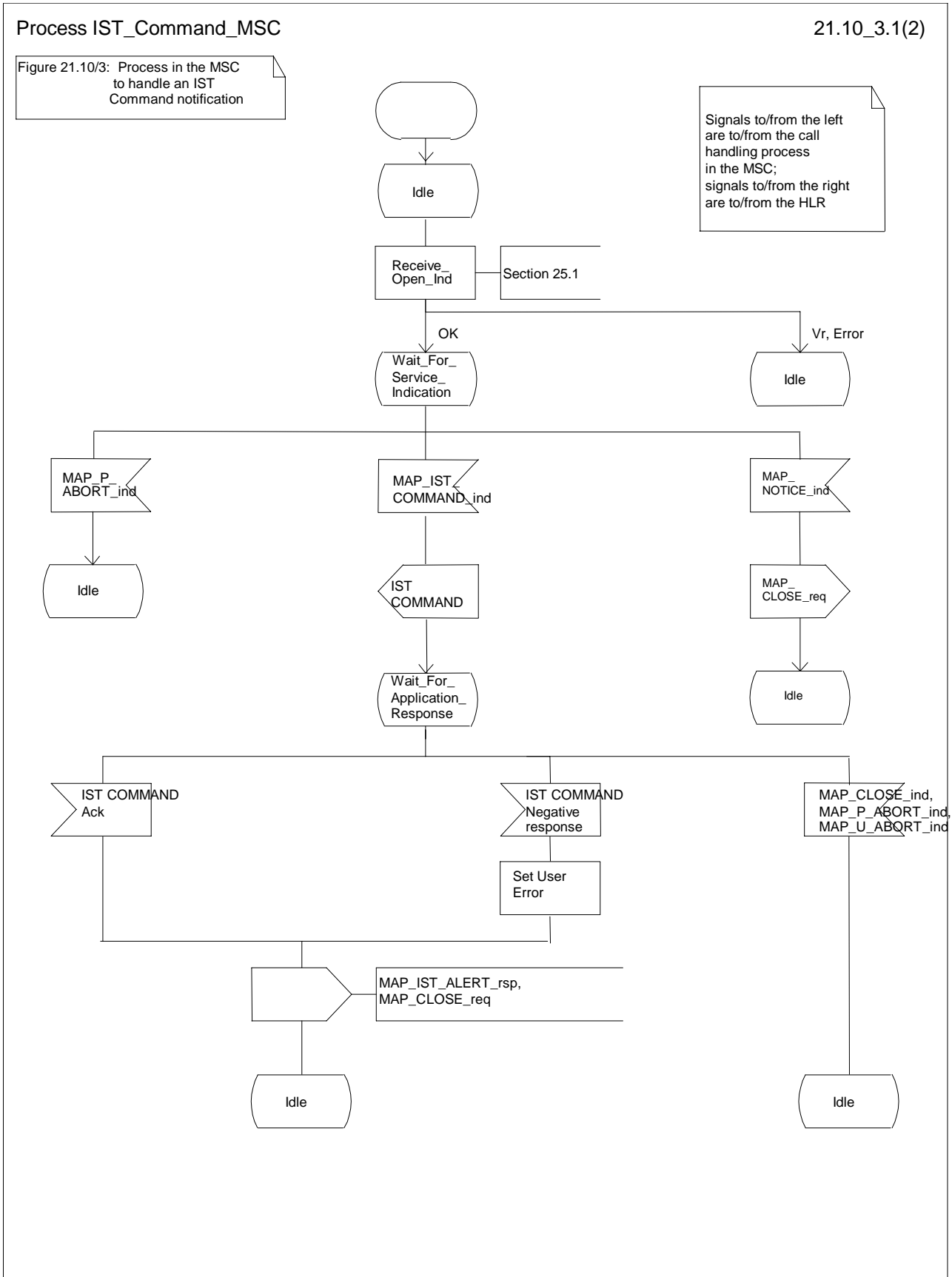


Figure 21.10/3: Process IST_Command_MSC

22 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.

22.1 Functional supplementary service processes

22.1.1 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 25 of the present document.

Once a CM connection is established, the MSC can handle supplementary service indications from the MS. Table 22.1/1 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.

Table 22.1/1: Relationship between received service indication and invoked process in the MSC

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

Figure 22.1/1 shows the co-ordinating process in the MSC.

Process SS_Coordinator_MSC

22.1_1(1)

Figure 22.1/1: Supplementary Service Coordination process in the MSC, to identify which functional supplementary service process shall be invoked.

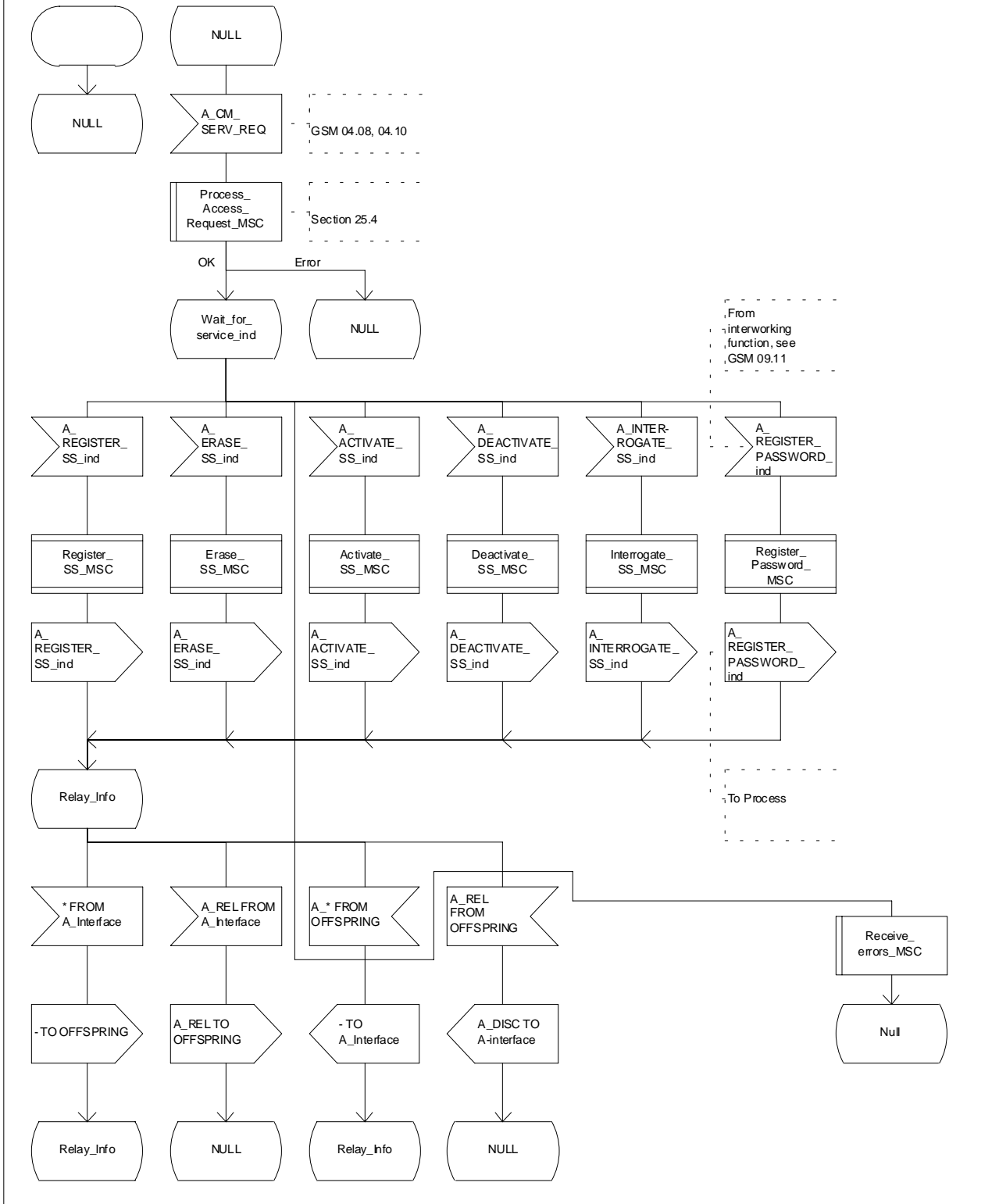


Figure 22.1/1: Process SS_Coordinator_MSC

22.1.2 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 25 of the present document.

If the Process Access Request was successful, the VLR can handle supplementary service indications from the MSC. Table 22.1/2 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.

Table 22.1/2: Relationship between received service indication and invoked process in the VLR

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

Figure 22.1/2 shows the co-ordinating process in the VLR.

Process SS_Coordinator_VLR

22.1_2.1(2)

Figure 22.1/2: 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.

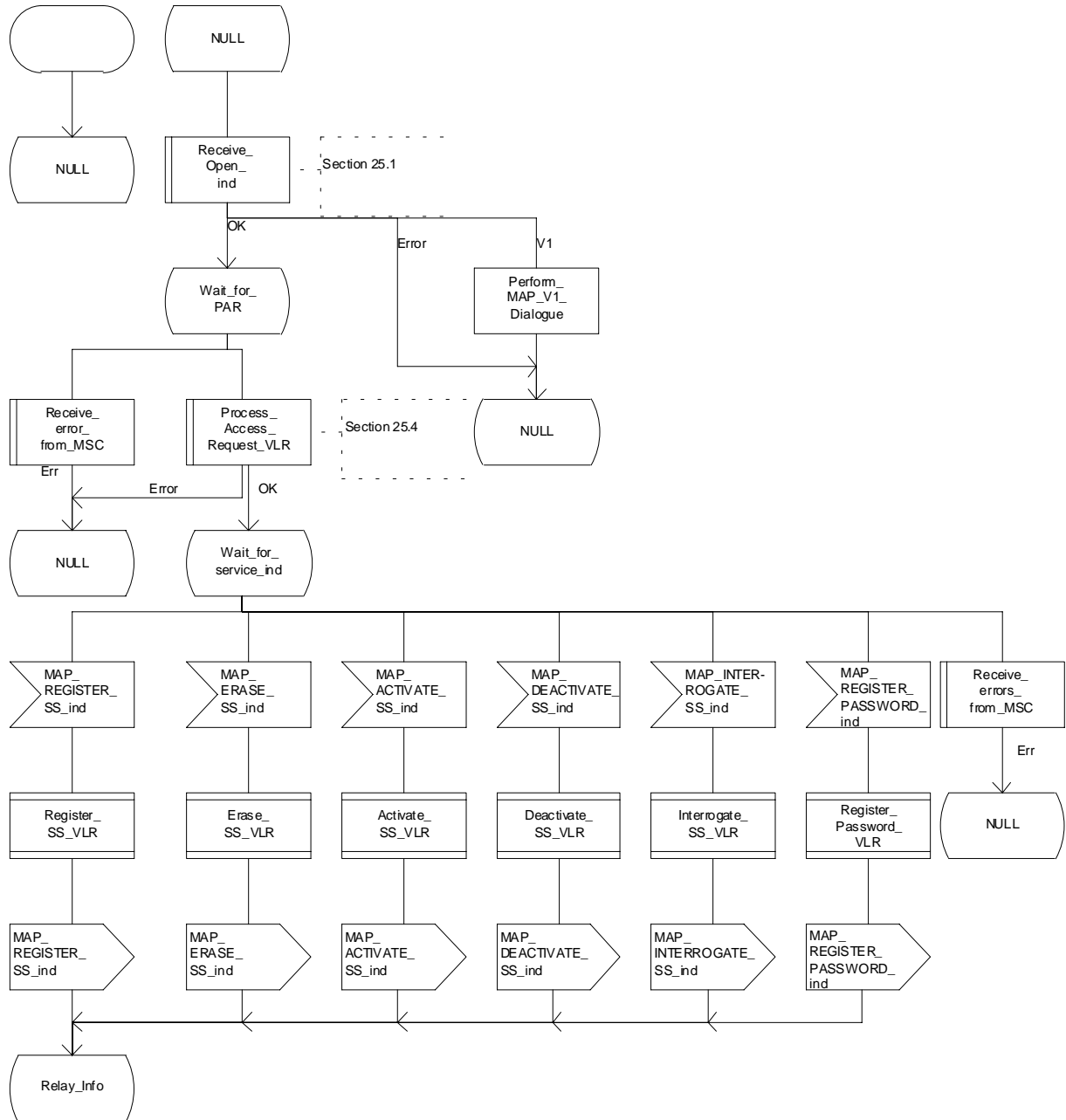


Figure 22.1/2 (sheet 1 of 2): Process SS_Coordinator_VLR

Process SS_Coordinator_VLR

22.1_2.2(2)

Figure 22.1/2: 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.

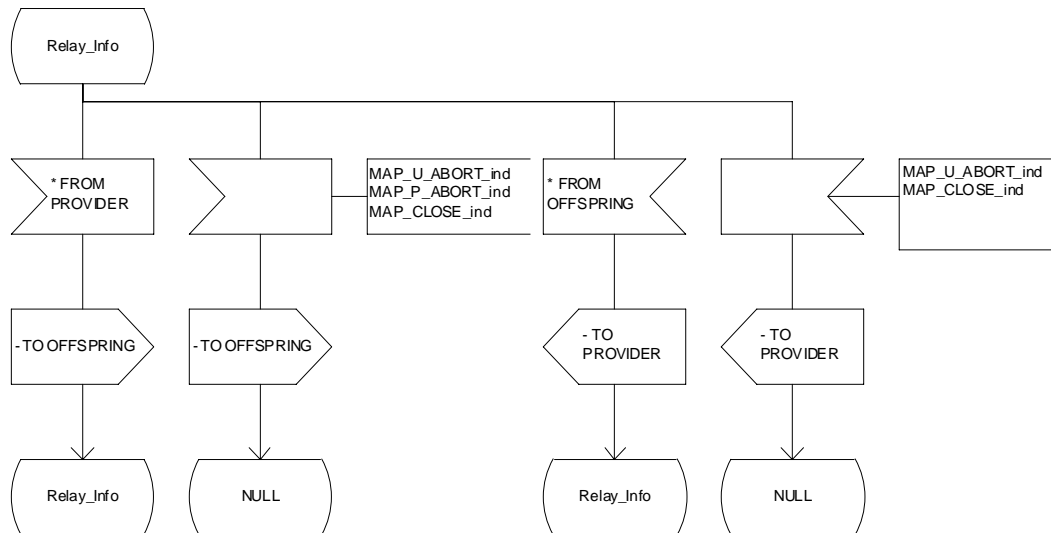


Figure 22.1/2 (sheet 2 of 2): Process SS_Coordinator_VLR

22.1.3 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 22.1/3 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.

Table 22.1/3: Relationship between received service indication and invoked process in the HLR.

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

Figure 22.1/3 shows the co-ordinating process in the HLR.

Process SS_Coordinator_HLR

22.1_3.1(2)

Figure 22.1/3: Supplementary Service Coordination process in the HLR, to identify which functional supplementary service process shall be invoked.

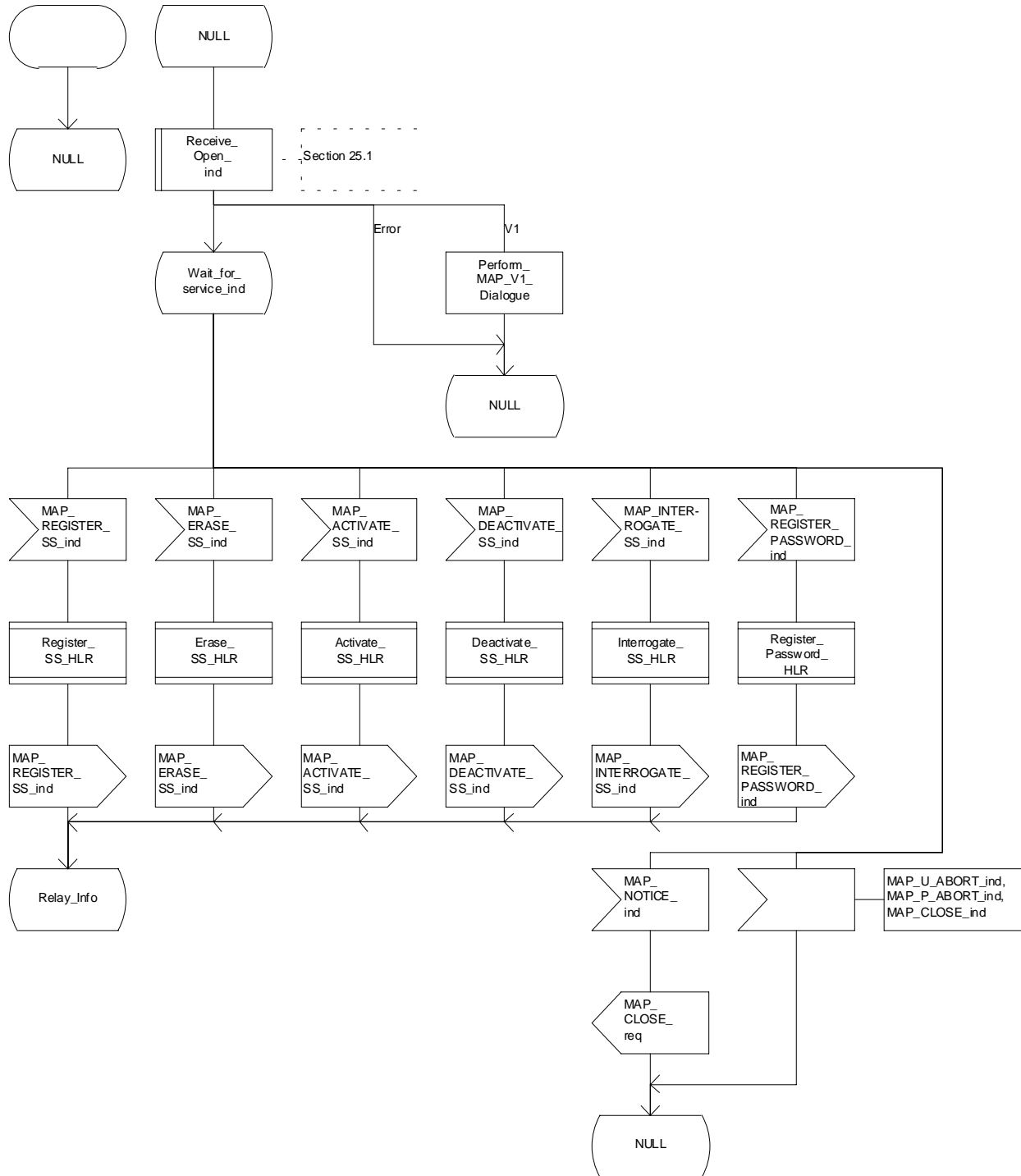


Figure 22.1/3 (sheet 1 of 2): Process SS_Coordinator_HLR

Process SS_Coordinator_HLR

22.1_3.2(2)

Figure 22.1/3: Supplementary Service Coordination process in the HLR, to identify which functional supplementary service process shall be invoked.

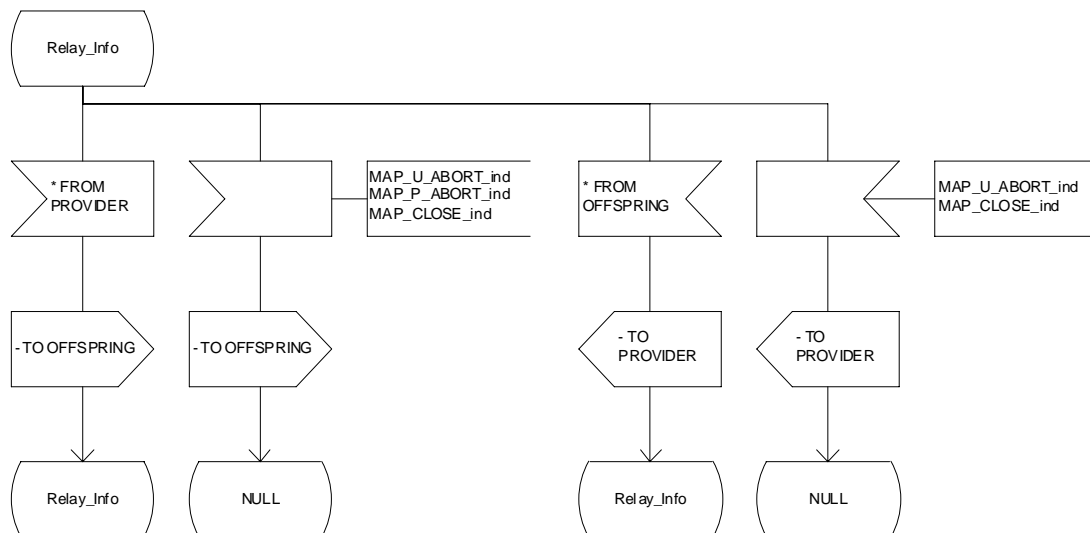


Figure 22.1/3 (sheet 2 of 2): Process SS_Coordinator_HLR

22.1.4 Call completion supplementary service process co-ordinator for HLR

The MAP co-ordinating process in the HLR to handle a dialogue opened with the callCompletion application context is shown in figure 22.1/4. The MAP process invokes a macro not defined in this subclause; the definition of this macro can be found as follows:

Receive_Open_Ind see subclause 25.1.1.

Any call completion SS process in the HLR starts by the HLR receiving a MAP-OPEN service indication. If that service is successful, the HLR can handle call completion supplementary service indications from the VLR. Table 22.1/4 shows the co-ordinating process' reaction on receipt of specific call completion SS service indications from the VLR. After the relevant process is invoked, the received service indication is sent to that process.

Table 22.1/4: Relationship between received service indication and invoked process in the HLR.

Service indication received	Process invoked
MAP_REGISTER_CC_ENTRY_ind	REGISTER_CC_ENTRY_HLR
MAP_ERASE_CC_ENTRY_ind	ERASE_CC_ENTRY_HLR

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 Call_Completion Co-ordinator is shown in figure 22.1/4.

Process CC_Coord_HLR

22.1_4(1)

Figure 22.1/4: Coordinating process in the HLR to handle a dialogue opened with the AC CallCompletionContext

Signals to/from the left are to/from the VLR via the MAP provider; signals to/from the right are to/from the child process

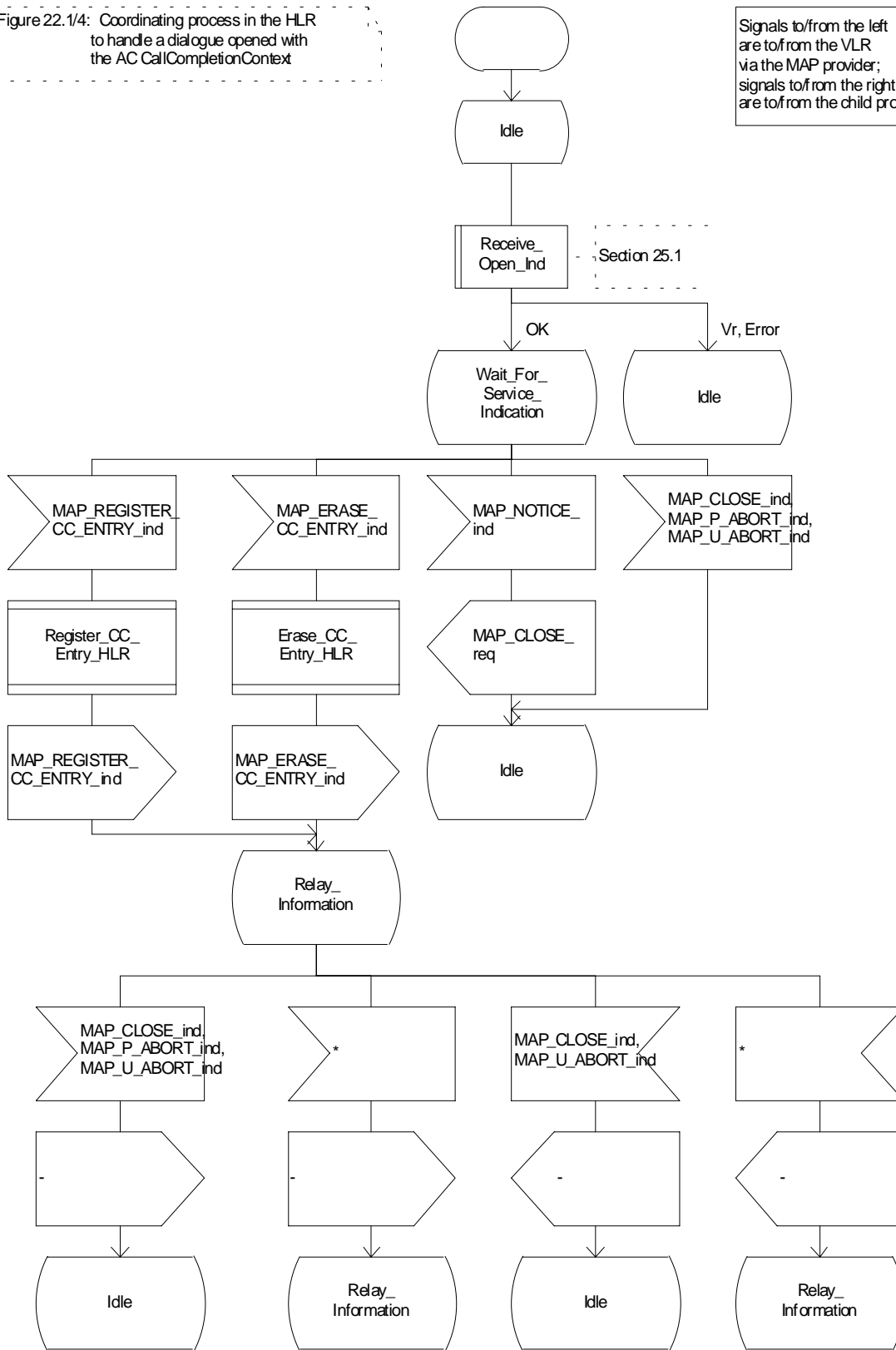


Figure 22.1/4: Process_CC_Coord_HLR

22.2 Registration procedure

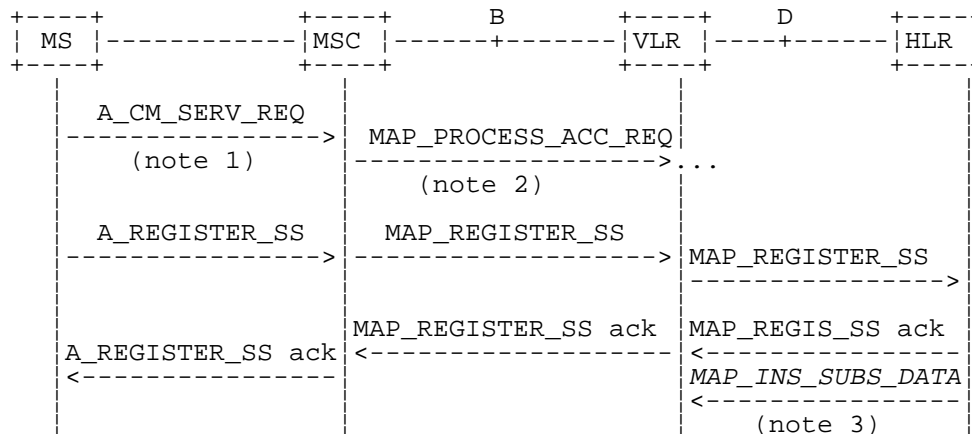
22.2.1 General

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 22.2.1/1.

The following services may be used:

- MAP_PROCESS_ACCESS_REQUEST (defined in clauses 8 and 25);
- MAP_TRACE_SUBSCRIBER_ACTIVITY (defined in clauses 9 and 25);
- MAP_PROVIDE_IMSI (defined in clauses 8 and 25);
- MAP_FORWARD_NEW_TMSI (defined in clauses 8 and 25);
- MAP_AUTHENTICATE (defined in clauses 8 and 25);
- MAP_SET_CIPHERING_MODE (defined in clauses 8 and 25);
- MAP_CHECK_IMEI (defined in clauses 8 and 25);
- MAP_READY_FOR_SM (defined in clauses 12 and 25);
- MAP_INSERT_SUBSCRIBER_DATA (defined in clauses 8 and 25);
- MAP_REGISTER_SS (defined in clause 11).



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 25 in the present document.

NOTE 3: Services printed in *italics* are optional.

Figure 22.2.1/1: Interfaces and services for supplementary service registration

22.2.2 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 22.2.2/1.

Process SS_REGISTER_MSC

22.2.2_1(1)

Figure 22.2.2/1 : Mobile initiated registration of supplementary service in the MSC

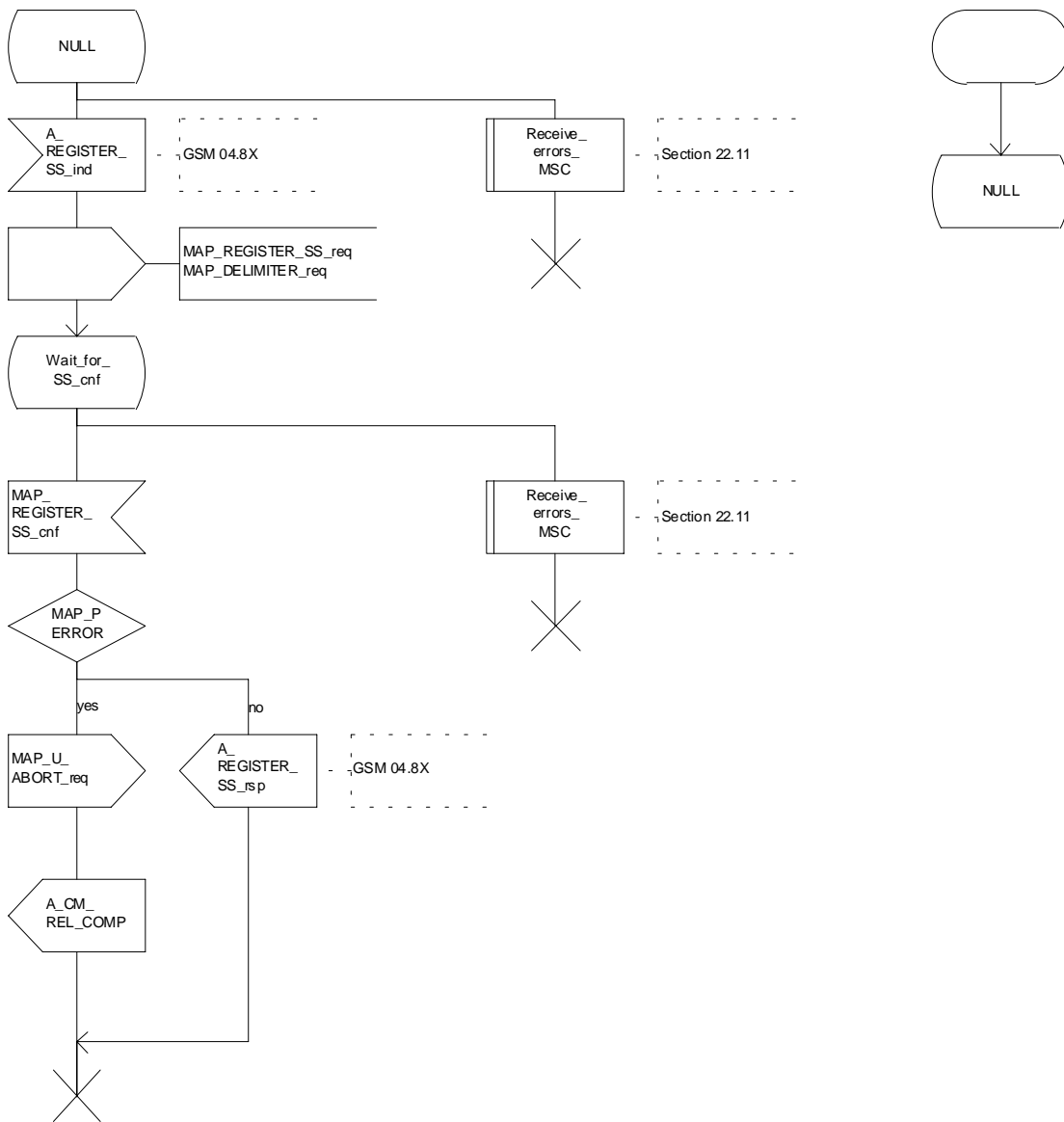


Figure 22.2.2/1: Procedure SS_Register_MSC

22.2.3 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 22.2.3/1.

Process SS_REGISTER_VLR

22.2.3_1.1(2)

Figure 22.2.3/1: Mobile initiated registration of supplementary services in the VLR

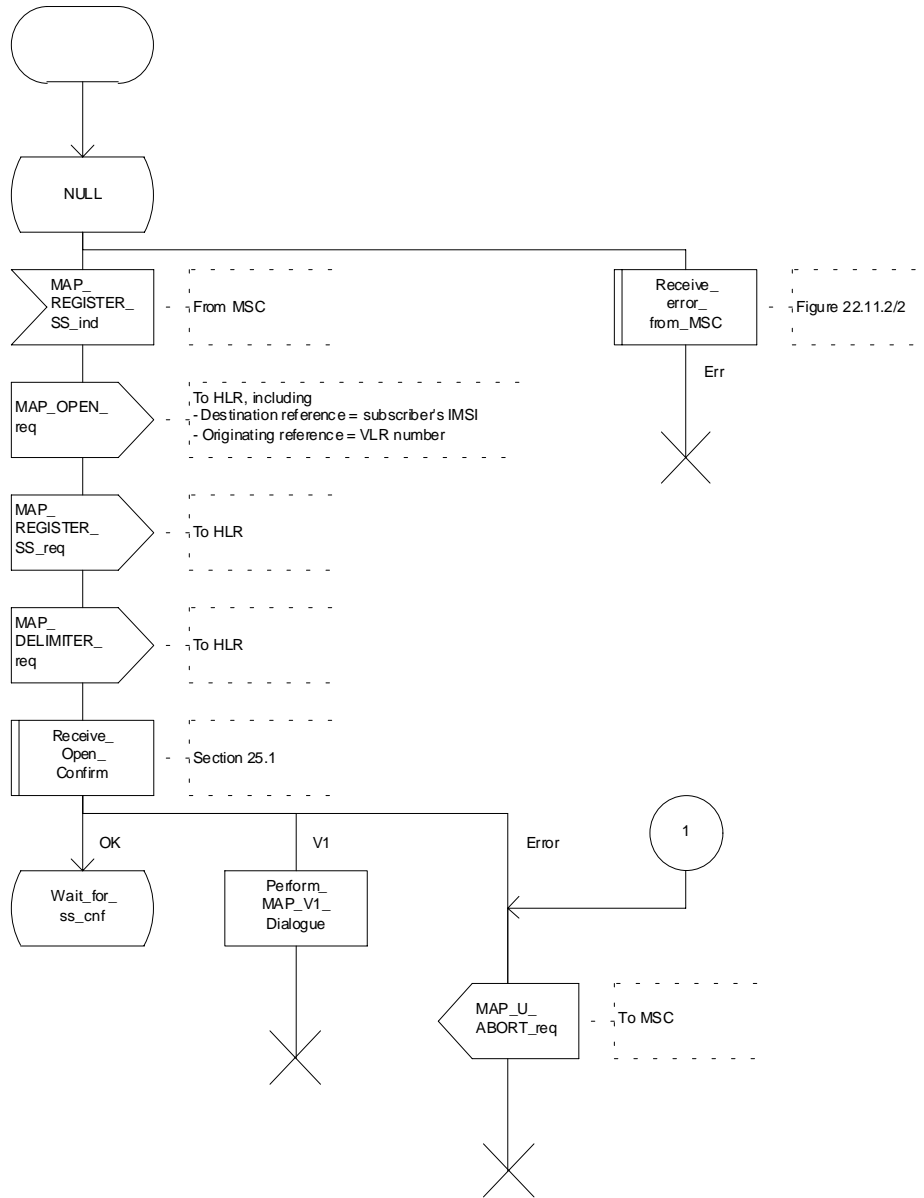


Figure 22.2.3/1 (sheet 1 of 2): Procedure SS_Register_VLR

Process SS_REGISTER_VLR

22.2.3_1.2(2)

Figure 22.2.3/1: Mobile initiated registration of supplementary services in the VLR

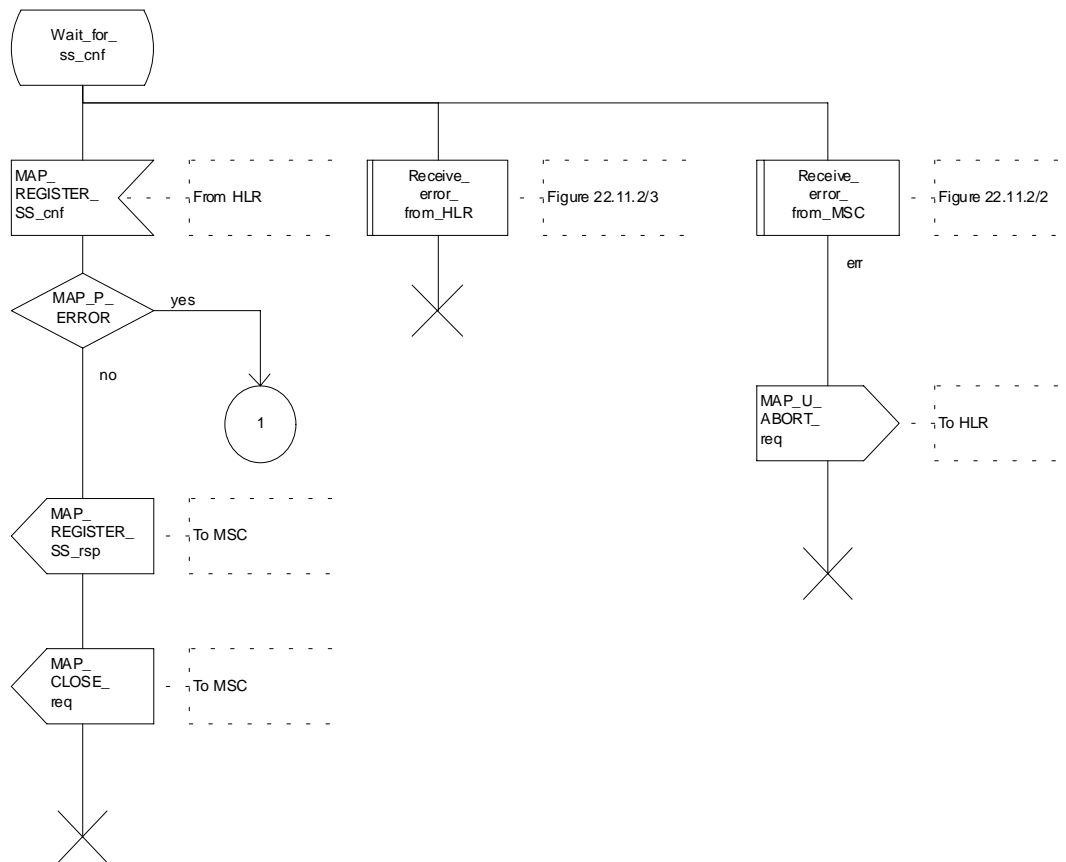


Figure 22.2.3/1 (sheet 2 of 2): Procedure SS_Register_VLR

22.2.4 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 22.2.4/1.

Process SS_REGISTER_HLR

22.2.4_1.1(2)

Figure 22.2.4/1: Registration of supplementary services procedure in HLR

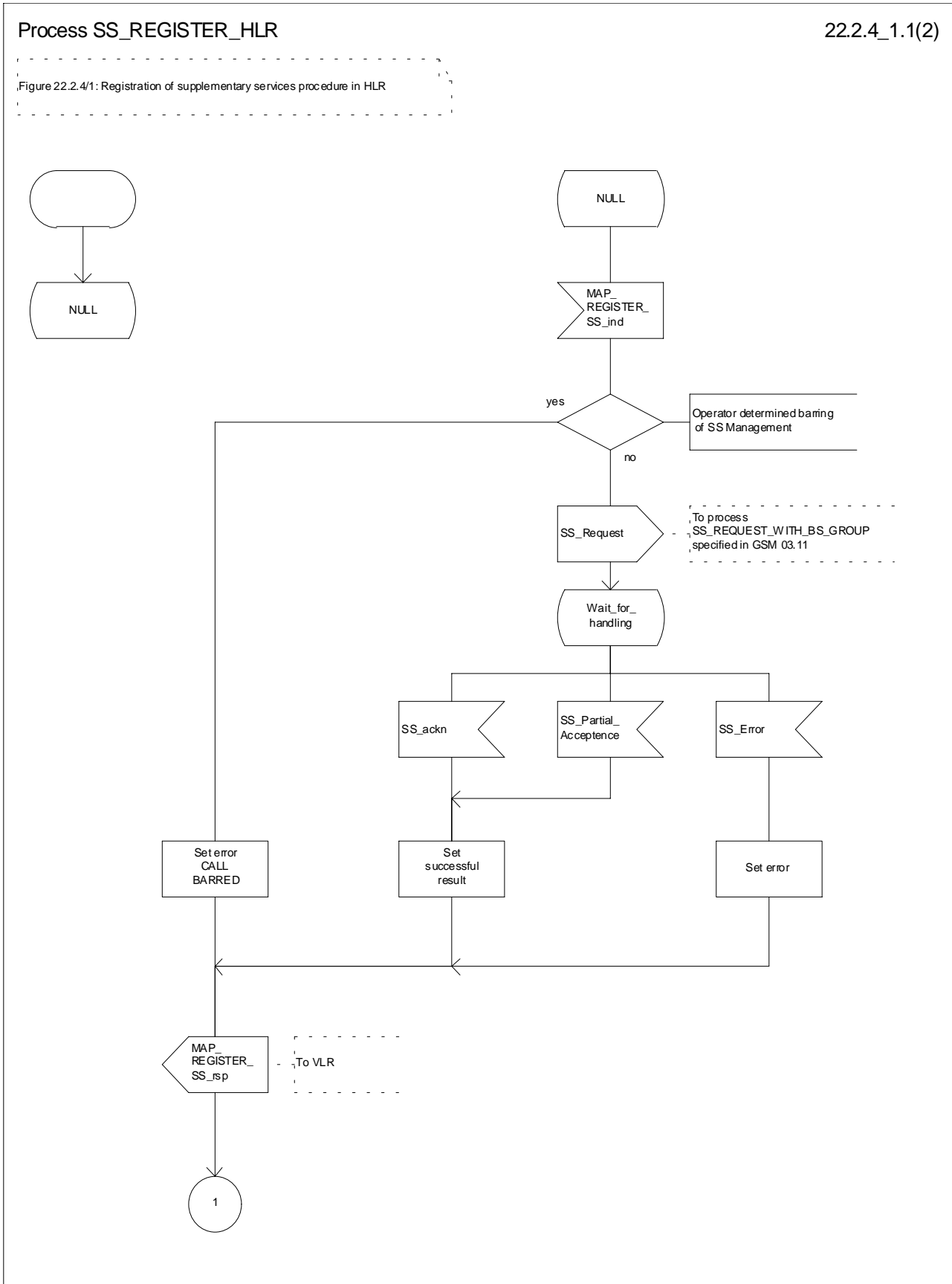


Figure 22.2.4/1 (sheet 1 of 2): Procedure SS_Register_HLR

Process SS_REGISTER_HLR

22.2.4_1.2(2)

Figure 22.2.4/1: Registration of supplementary services procedure in HLR

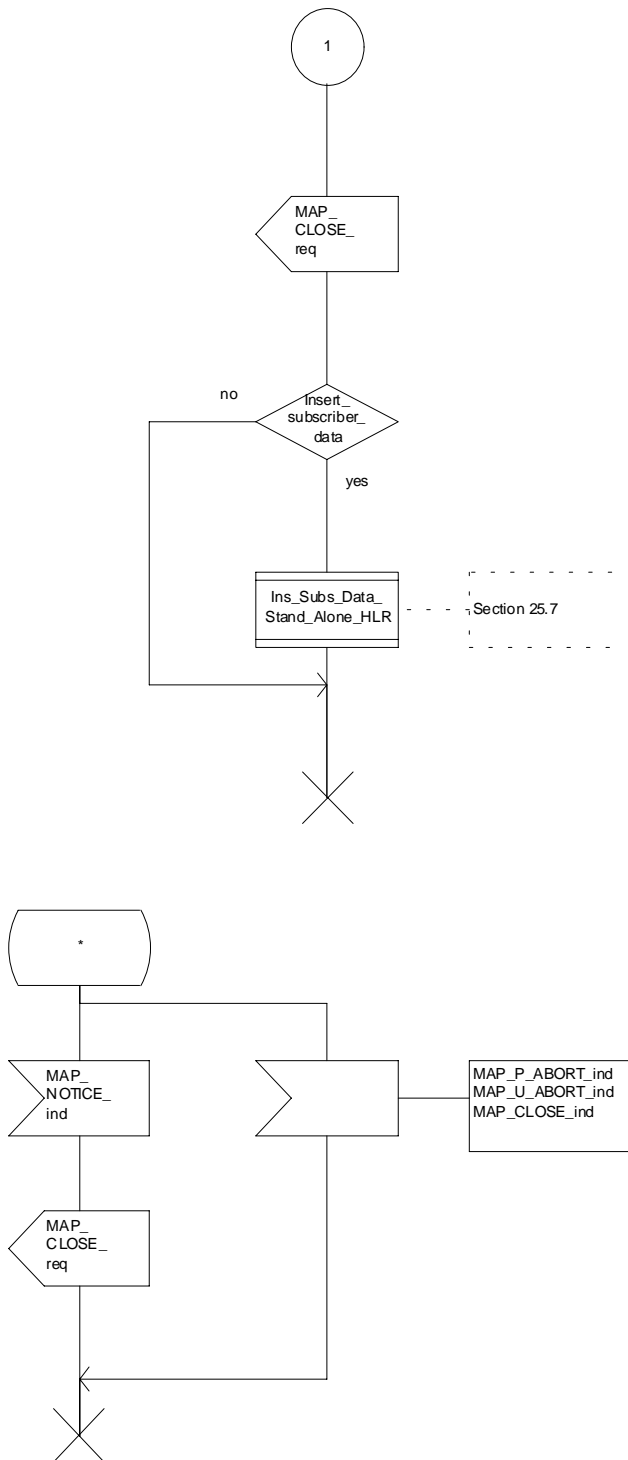


Figure 22.2.4/1 (sheet 2 of 2): Procedure SS_Register_HLR

22.3 Erasure procedure

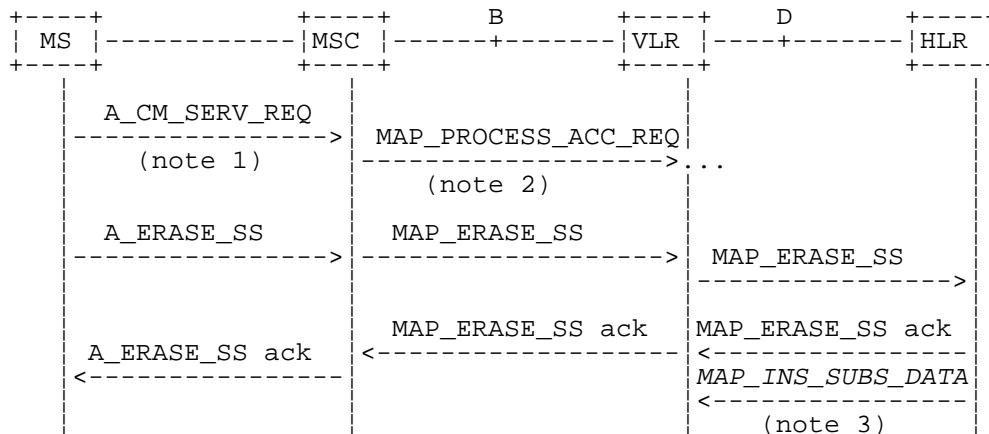
22.3.1 General

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 22.3.1/1.

The following services may be used:

MAP_PROCESS_ACCESS_REQUEST	(defined in subclauses 8 and 25);
MAP_TRACE_SUBSCRIBER_ACTIVITY	(defined in clauses 9 and 25);
MAP_PROVIDE_IMSI	(defined in clauses 8 and 25);
MAP_FORWARD_NEW_TMSI	(defined in clauses 8 and 25);
MAP_AUTHENTICATE	(defined in clauses 8 and 25);
MAP_SET_CIPHERING_MODE	(defined in clauses 8 and 25);
MAP_CHECK_IMEI	(defined in clauses 8 and 25);
MAP_READY_FOR_SM	(defined in clauses 12 and 25);
MAP_INSERT_SUBSCRIBER_DATA	(defined in clauses 8 and 25);
MAP_ERASE_SS	(defined in clause 11).



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 25 in the present document.

NOTE 3: Services printed in *italics* are optional.

Figure 22.3.1/1: Interfaces and services for supplementary service erasure

22.3.2 Procedures in the MSC

The MSC procedures for erasure are identical to those specified for registration in subclause 22.2.2. The text and diagrams in subclause 22.2.2 apply with all references to registration changed to erasure.

22.3.3 Procedures in the VLR

The VLR procedures for erasure are identical to those specified for registration in subclause 22.2.3. The text and diagrams in subclause 22.2.3 apply with all references to registration changed to erasure.

22.3.4 Procedures in the HLR

The HLR procedure for erasure is identical to those specified for registration in subclause 22.2.4. The text and diagrams in subclause 22.2.4 apply with all references to registration changed to erasure.

22.4 Activation procedure

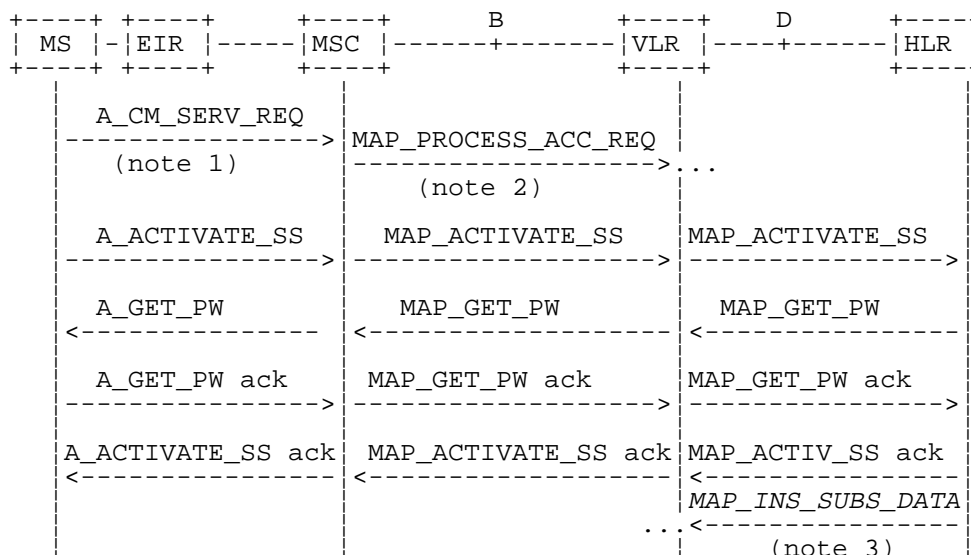
22.4.1 General

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 22.4.1/1.

The following services may be used:

MAP_PROCESS_ACCESS_REQUEST	(defined in clauses 8 and 25);
MAP_TRACE_SUBSCRIBER_ACTIVITY	(defined in clauses 9 and 25);
MAP_PROVIDE_IMSI	(defined in clauses 8 and 25);
MAP_FORWARD_NEW_TMSI	(defined in clauses 8 and 25);
MAP_AUTHENTICATE	(defined in clauses 8 and 25);
MAP_SET_CIPHERING_MODE	(defined in clauses 8 and 25);
MAP_CHECK_IMEI	(defined in clauses 8 and 25);
MAP_READY_FOR_SM	(defined in clauses 12 and 25);
MAP_GET_PASSWORD	(defined in clause 11);
MAP_INSERT_SUBSCRIBER_DATA	(defined in clauses 8 and 25);
MAP_ACTIVATE_SS	(defined in clause 11).



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 25 of this document.

NOTE 3: Services printed in italics are optional.

Figure 22.4.1/1: Interfaces and services for supplementary service activation

22.4.2 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 22.2.2 of the present document.

The activation procedure in the MSC is shown in figure 22.4.2/1.

Process ACTIVATE_SS_MSC

22.4.2_1(1)

Figure 22.4.2/1: Mobile initiated activation of supplementary service in the MSC

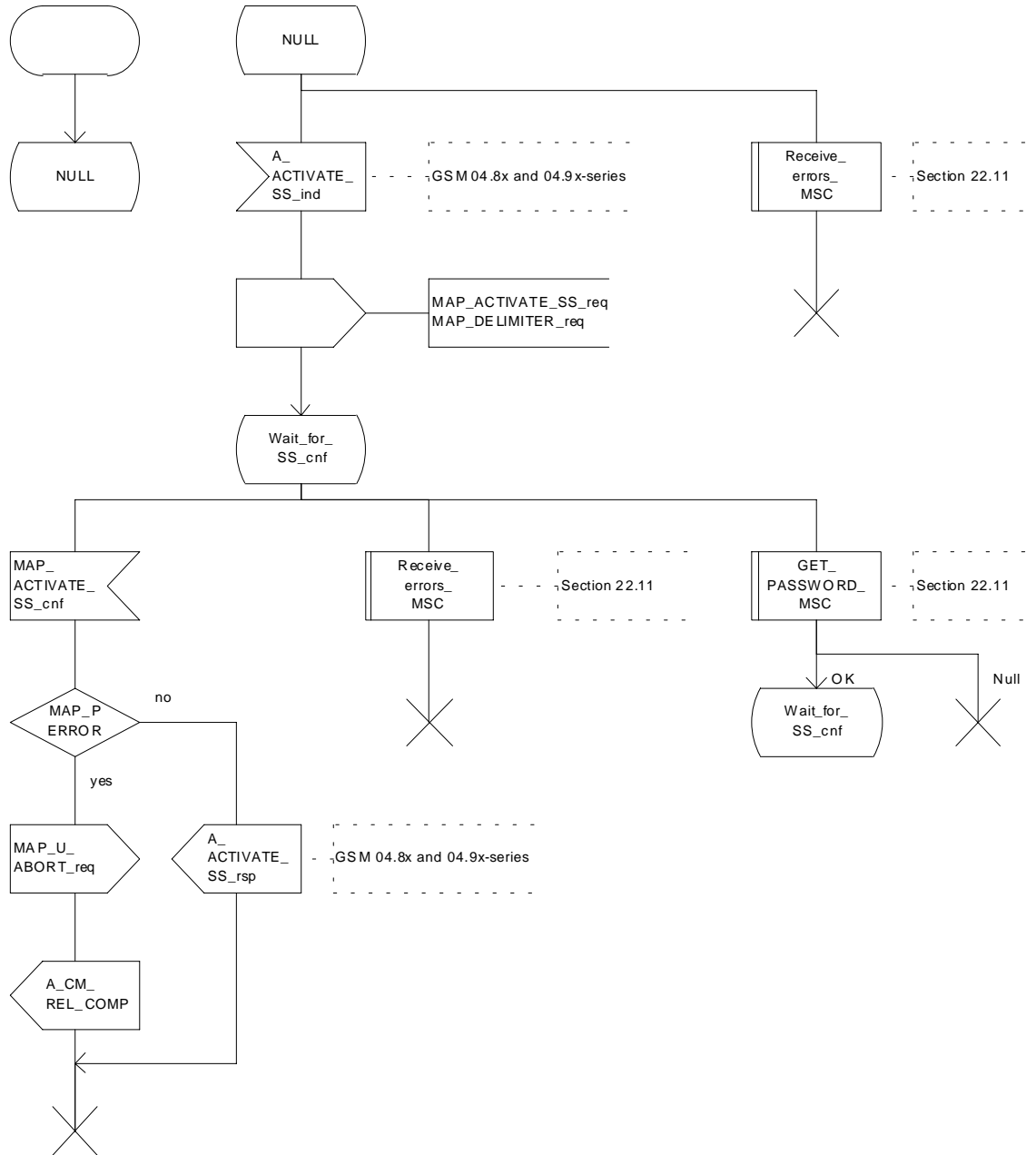


Figure 22.4.2/1: Procedure Activate_SS_MSC

22.4.3 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 22.2.3 of the present document.

The activation procedure in the VLR is shown in figure 22.4.3/1.

Process ACTIVATE_SS_VLR

22.4.3_1.1(2)

Figure 22.4.3/1: Activation of supplementary service procedure in the VLR

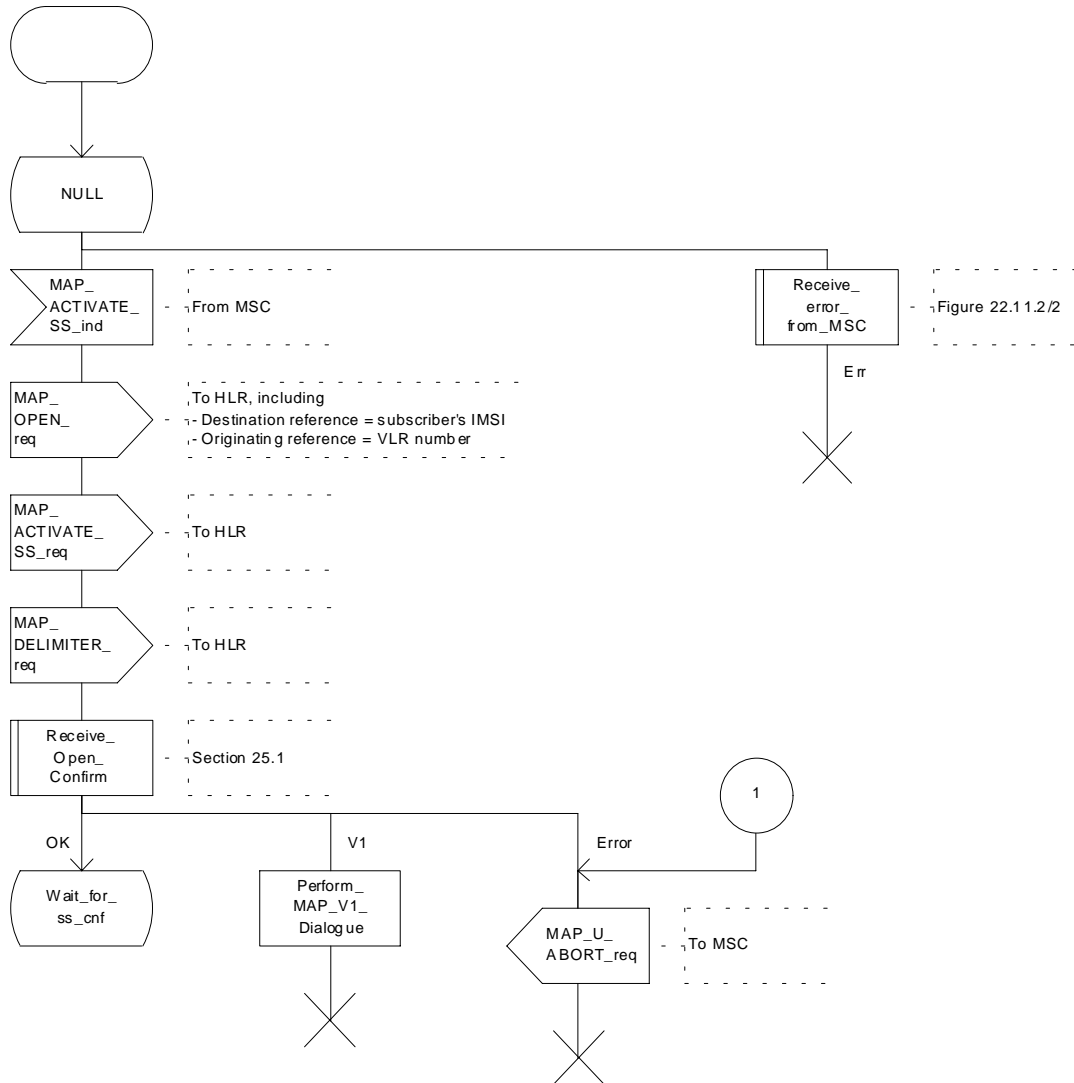


Figure 22.4.3/1 (sheet 1 of 2): Procedure Activate_SS_VLR

Process ACTIVATE_SS_VLR

22.4.3_1.2(2)

Figure 22.4.3/1: Activation of supplementary service procedure in the VLR

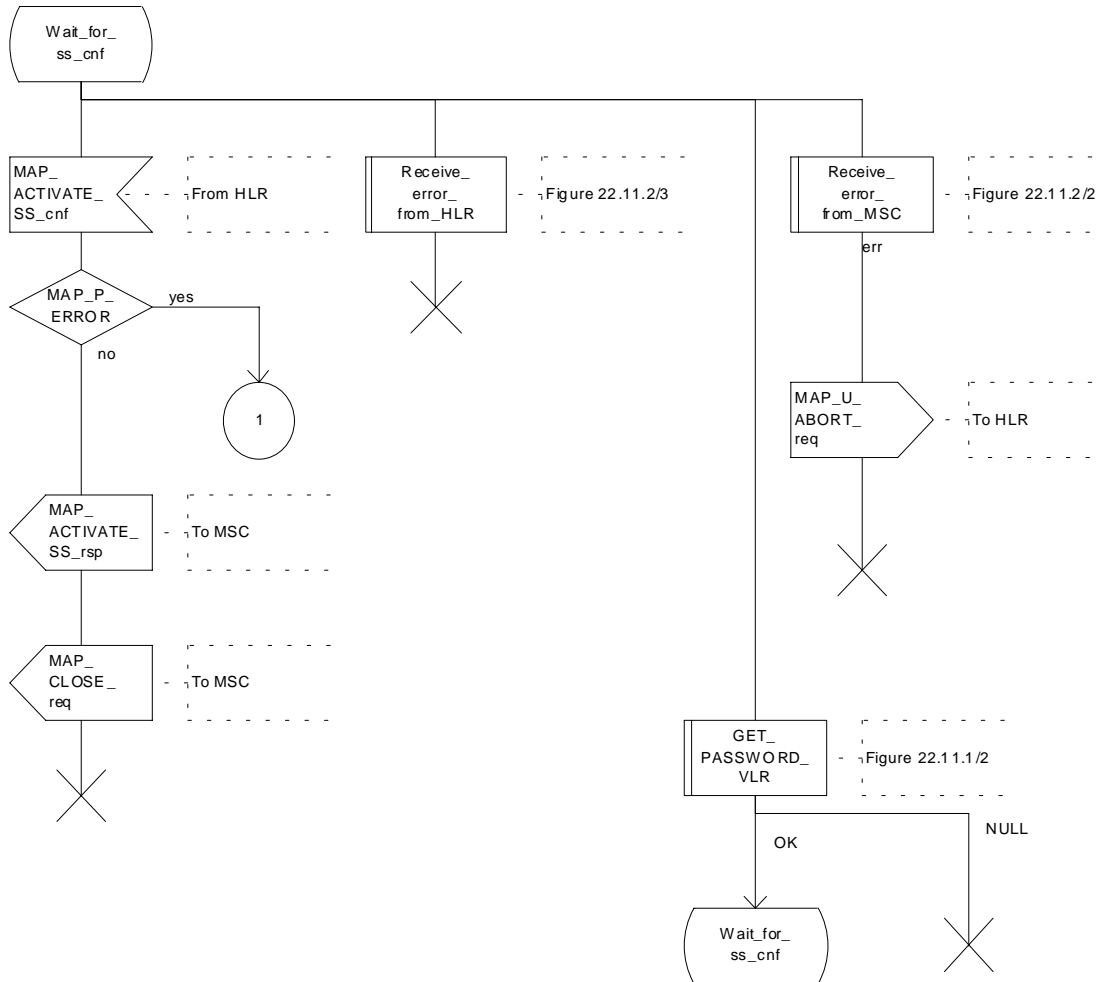


Figure 22.4.3/1 (sheet 2 of 2): Procedure SS_Activate_VLR

22.4.4 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 22.2.4 above.

The activation procedure in the HLR is shown in figure 22.4.4/1.

Process ACTIVATE_SS_HLR

22.4.4_1.1(2)

Figure 22.4.4/1: Activation of supplementary services procedure in HLR.

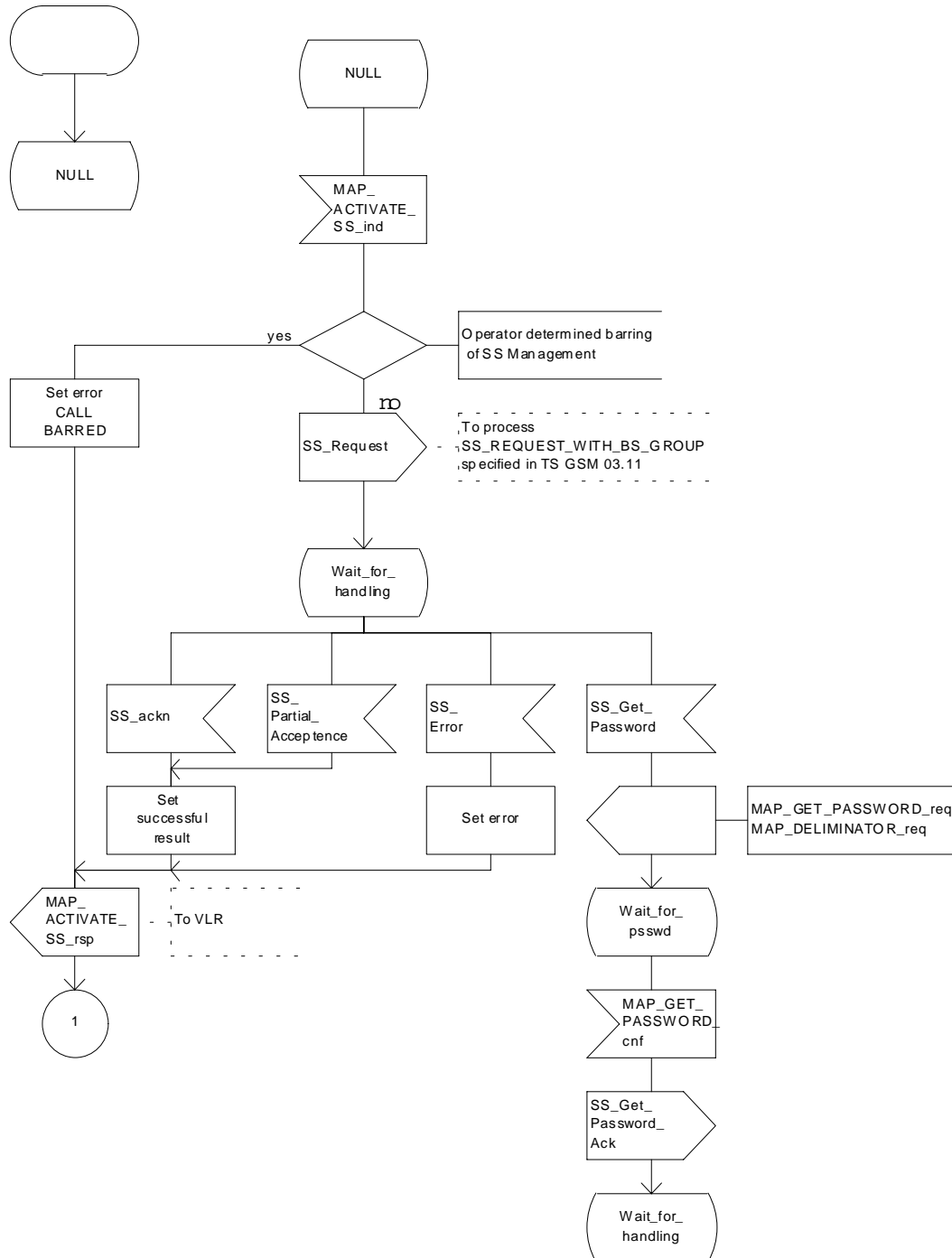


Figure 22.4.4/1 (sheet 1 of 2): Procedure Activate_SS_HLR

Process ACTIVATE_SS_HLR

22.4.4_1.2(2)

Figure 22.4.4/1: Activation of supplementary services procedure in HLR.

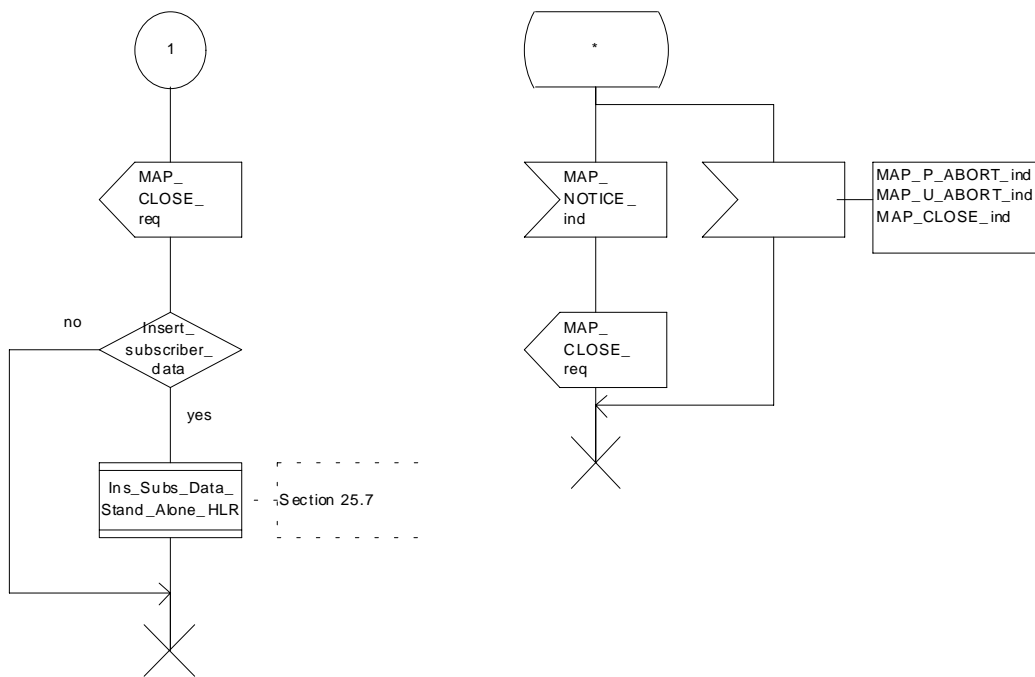


Figure 22.4.4/1 (sheet 2 of 2): Procedure Activate_SS_HLR

22.5 Deactivation procedure

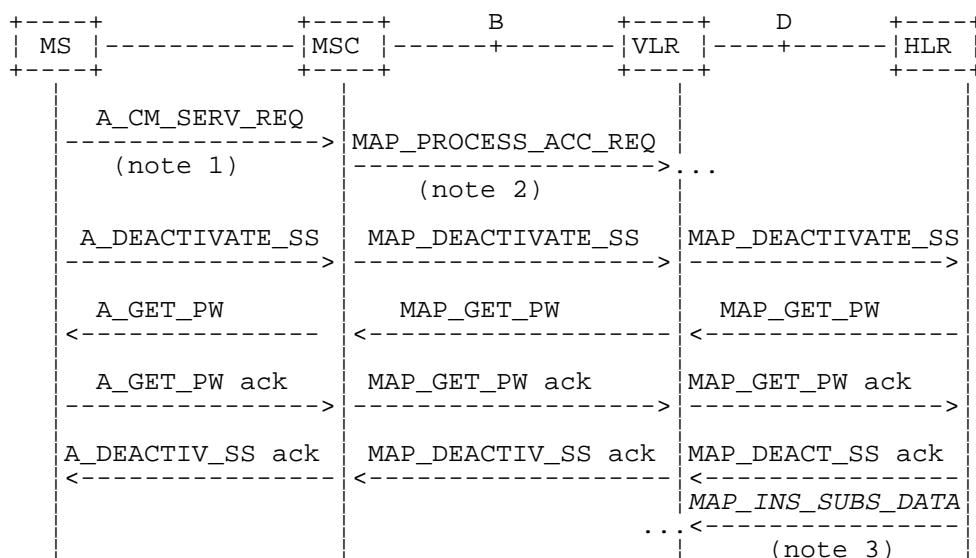
22.5.1 General

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 22.5.1/1.

The following services may be used:

- MAP_PROCESS_ACCESS_REQUEST (defined in clauses 8 and 25);
- MAP_TRACE_SUBSCRIBER_ACTIVITY (defined in clauses 9 and 25);
- MAP_PROVIDE_IMSI (defined in clauses 8 and 25);
- MAP_FORWARD_NEW_TMSI (defined in clauses 8 and 25);
- MAP_AUTHENTICATE (defined in clauses 8 and 25);
- MAP_SET_CIPHERING_MODE (defined in clauses 8 and 25);
- MAP_CHECK_IMEI (defined in clauses 8 and 25);
- MAP_READY_FOR_SM (defined in clauses 12 and 25);
- MAP_GET_PASSWORD (defined in clause 11);
- MAP_INSERT_SUBSCRIBER_DATA (defined in clauses 8 and 25);
- MAP_DEACTIVATE_SS (defined in clause 11).



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 25 in the present document.

NOTE 3: Services printed in *italics* are optional.

Figure 22.5.1/1: Interfaces and services for supplementary service deactivation

22.5.2 Procedures in the MSC

The MSC procedures for deactivation are identical to those specified for activation in subclause 22.4.2. The text and diagrams in subclause 22.4.2 apply with all references to activation changed to deactivation.

22.5.3 Procedures in the VLR

The VLR procedures for deactivation are identical to those specified for activation in subclause 22.4.3. The text and diagrams in subclause 22.4.3 apply with all references to activation changed to deactivation.

22.5.4 Procedures in the HLR

The HLR procedures for deactivation are identical to those specified for activation in subclause 22.4.4. The text and diagrams in subclause 22.4.4 apply with all references to activation changed to deactivation.

22.6 Interrogation procedure

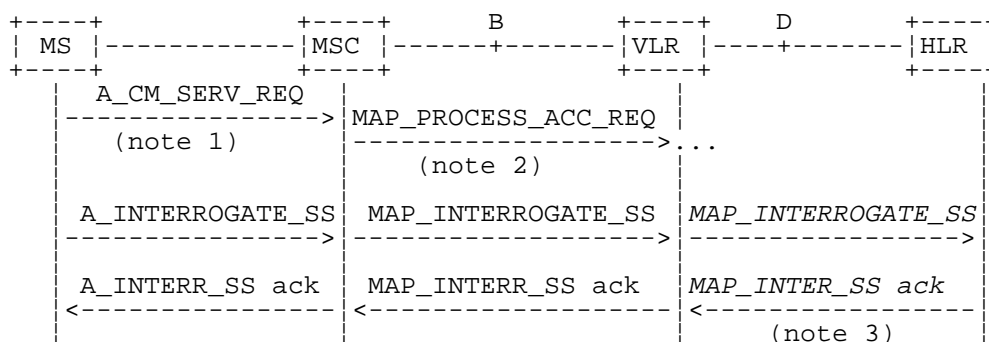
22.6.1 General

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 22.6.1/1.

The following services may be used:

MAP_PROCESS_ACCESS_REQUEST	(defined in clauses 8 and 25);
MAP_TRACE_SUBSCRIBER_ACTIVITY	(defined in clauses 9 and 25);
MAP_PROVIDE_IMSI	(defined in clauses 8 and 25);
MAP_FORWARD_NEW_TMSI	(defined in clauses 8 and 25);
MAP_AUTHENTICATE	(defined in clauses 8 and 25);
MAP_SET_CIPHERING_MODE	(defined in clauses 8 and 25);
MAP_CHECK_IMEI	(defined in clauses 8 and 25);
MAP_READY_FOR_SM	(defined in clauses 12 and 25);
MAP_INTERROGATE_SS	(defined in clause 11).



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 25 in the present document.

NOTE 3: Services printed in italics are optional.

Figure 22.6.1/1: Interfaces and services for supplementary service interrogation

22.6.2 Procedures in the MSC

The MSC procedures for interrogation are identical to those specified for registration in subclause 22.2.2. The text and diagrams in subclause 22.2.2 apply with all references to registration changed to interrogation.

22.6.3 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 22.2.3. The Interrogation procedure is described in figure 22.6.3/1.

Process INTERROGATE_SS_VLR

22.6.3_1.1(3)

Figure 22.6.3/1: Interrogation of supplementary service procedure in VLR

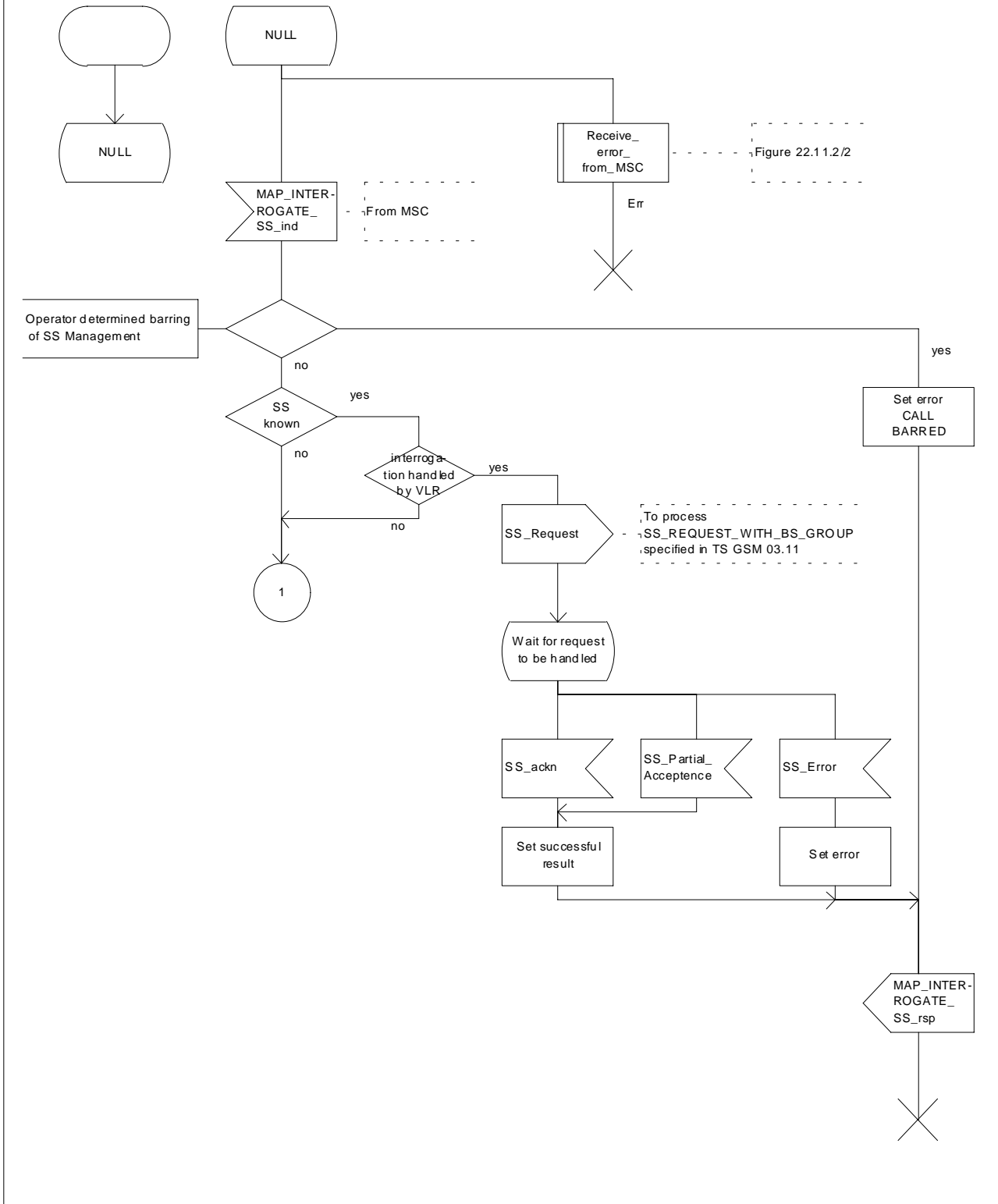


Figure 22.6.3/1 (sheet 1 of 3): Procedure Interrogate_SS_VLR

Process INTERROGATE_SS_VLR

22.6.3_1.2(3)

Figure 22.6.3/1: Interrogation of supplementary service procedure in VLR

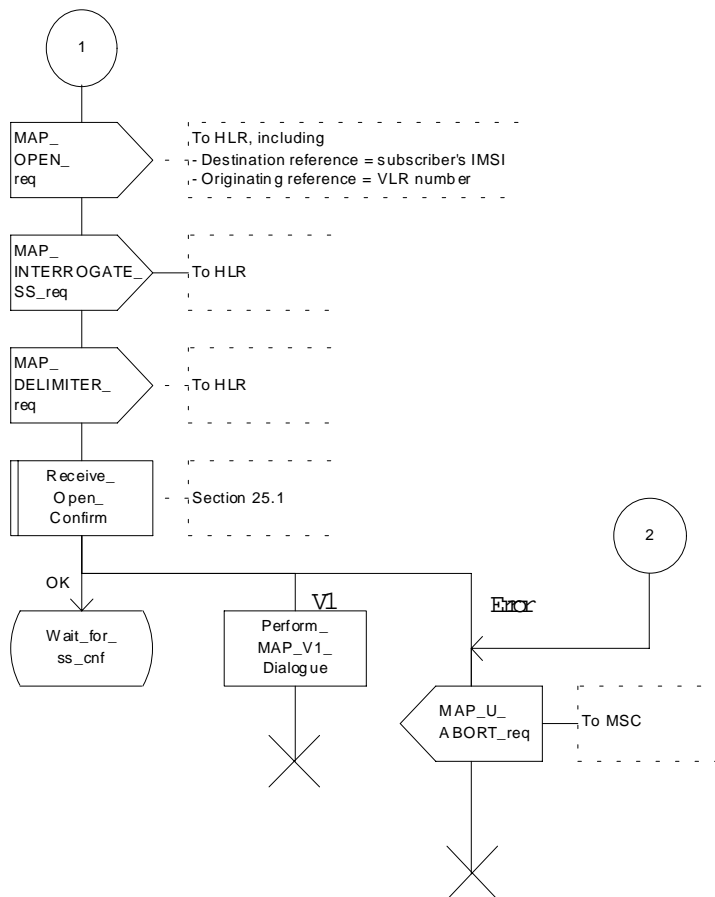


Figure 22.6.3/1 (sheet 2 of 3): Procedure Interrogate_SS_VLR

Process INTERROGATE_SS_VLR

22.6.3_1.3(3)

Figure 22.6.3/1: Interrogation of supplementary service procedure in VLR

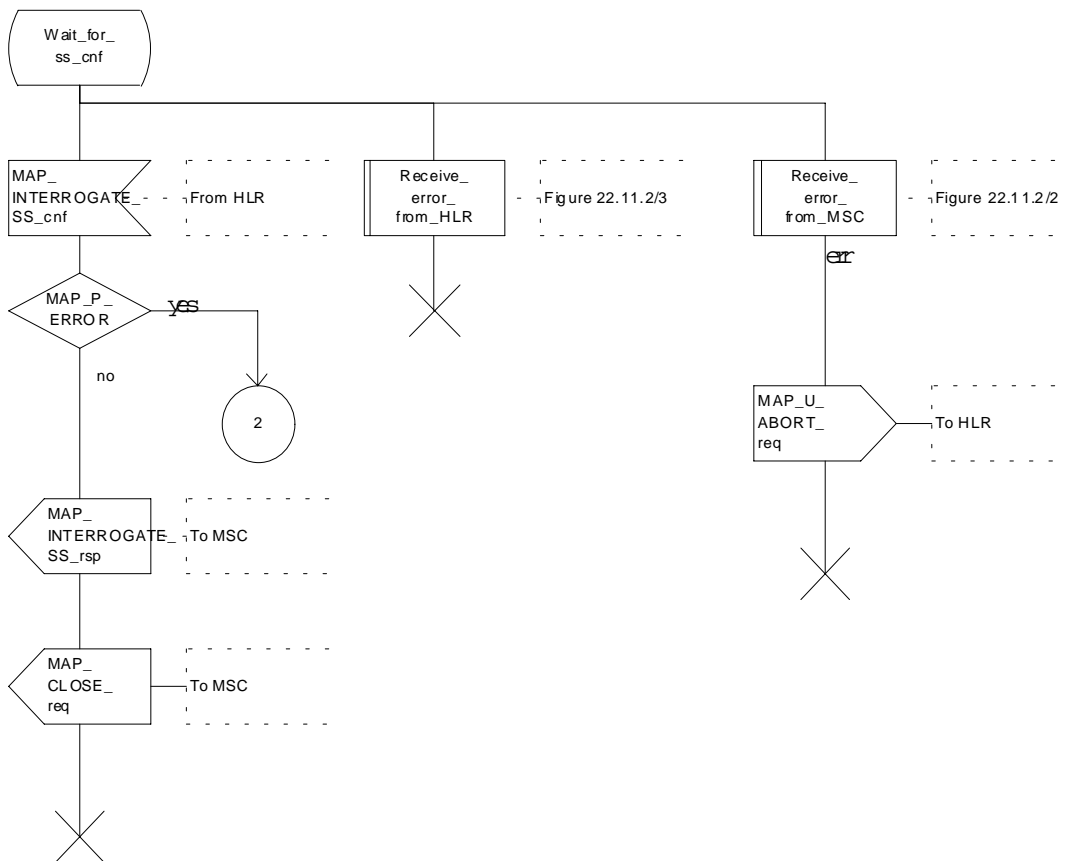


Figure 22.6.3/1 (sheet 3 of 3): Procedure Interrogate_SS_VLR

22.6.4 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 22.2.3. The Interrogation procedure is described in figure 22.6.4/1.

Process INTERROGATE_SS_HLR

22.6.4_1(1)

Figure 22.6.4/1: Interrogation of supplementary services procedure in HLR

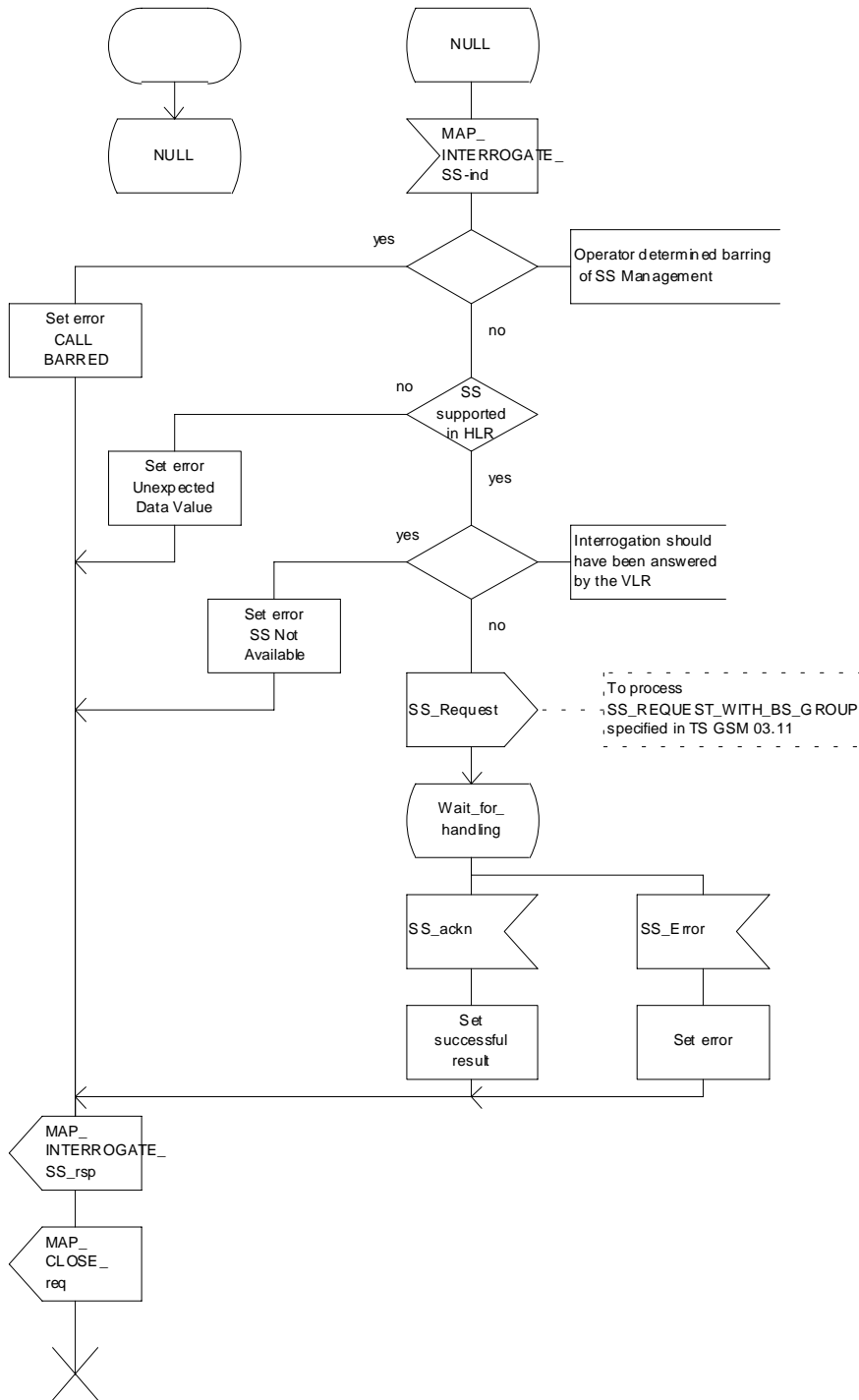


Figure 22.6.4/1: Procedure Interrogate_SS_HLR

22.7 Invocation procedure

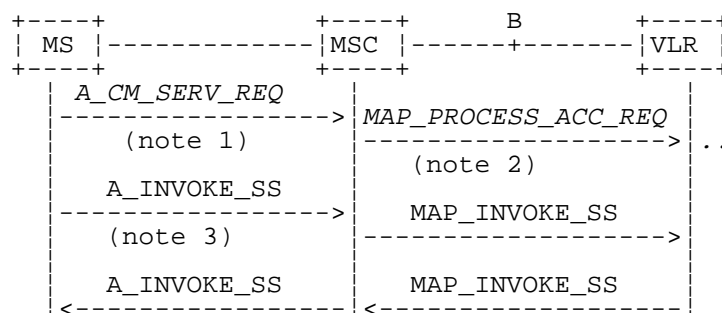
22.7.1 General

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 22.7.1/1. 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 8 and 25);
MAP_TRACE_SUBSCRIBER_ACTIVITY	(defined in clauses 9 and 25);
MAP_PROVIDE_IMSI	(defined in clauses 8 and 25);
MAP_FORWARD_NEW_TMSI	(defined in clauses 8 and 25);
MAP_AUTHENTICATE	(defined in clauses 8 and 25);
MAP_SET_CIPHERING_MODE	(defined in clauses 8 and 25);
MAP_CHECK_IMEI	(defined in clauses 8 and 25);
MAP_READY_FOR_SM	(defined in clauses 12 and 25);
MAP_INVOKE_SS	(defined in clause 11).



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 25 in the present document.

NOTE 3: A_INVOKE_SS is a generic message to illustrate any supplementary service invocation request message on the air interface, e.g. BuildMPTY, see GSM 04.80.

Figure 22.7.1/1: Interfaces and services for supplementary service invocation

22.7.2 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 the present document.

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 25 of the present document.

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.

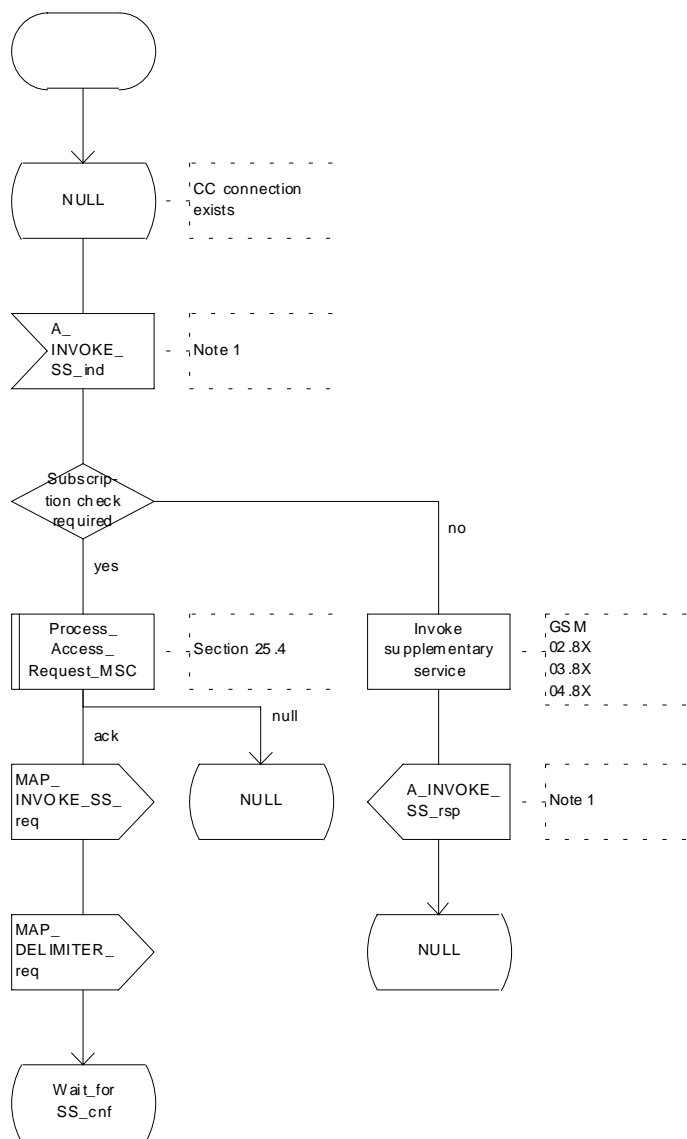
If an A_CM_RELEASE indication is received from the MS, all open transactions are released using the MAP_U_ABORT request indicating application procedure cancellation; the process terminates.

The invocation procedure in the MSC is shown in figure 22.7.2/1.

Process INVOKE_SS_MSC

22.7.2_1.1(2)

Figure 22.7.2/1: Mobile initiated invocation of supplementary service procedure in the MSC



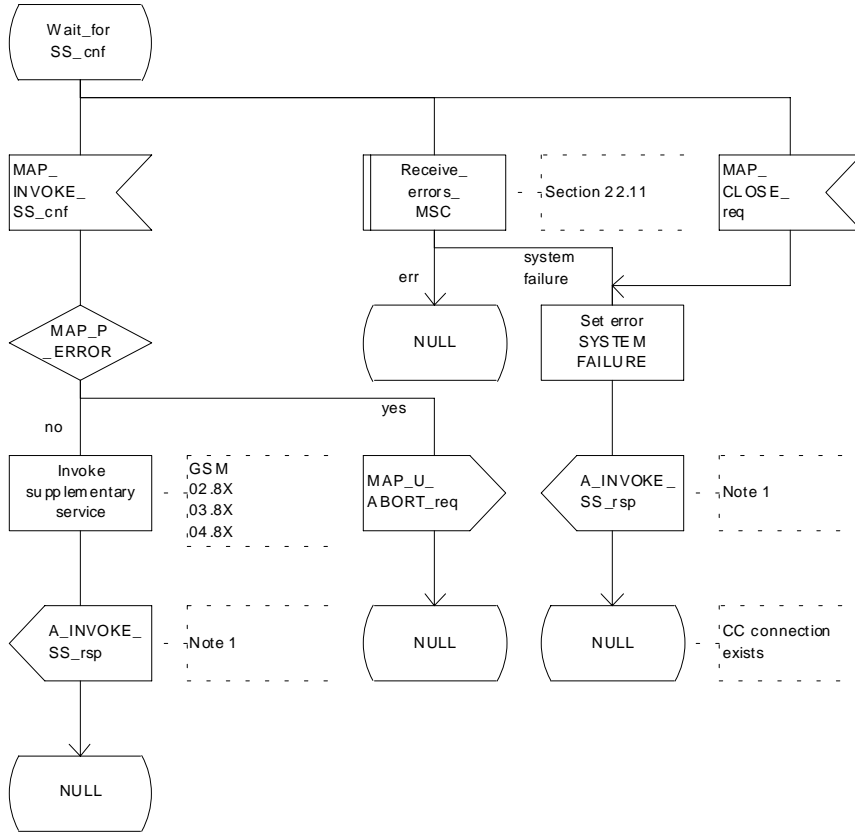
Note 1: Fictitious signal to indicate receipt/sending of SS invocation in voke component on the air interface (eg. BuildMPTY). Described in GSM 04.8X and 09.11.

Figure 22.7.2/1 (sheet 1 of 2): Procedure Invoke_SS_MSC

Process INVOKE_SS_MSC

22.7.2_1.2(2)

Figure 22.7.2/1: Mobile initiated invocation of supplementary service procedure in the MSC



Note 1: Fictitious signal to indicate receipt/sending of SS invocation invoke component on the air interface, (eg. BuildMPTY). Described in GSM 04.08X and 09.11.

Figure 22.7.2/1 (sheet 2 of 2): Procedure Invoke_SS_MSC

22.7.3 Procedures in the VLR

Process Access Request

When receiving the MAP_PROCESS_ACCESS_REQUEST indication, the VLR acts as described in clause 25 of the present document.

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 22.7.3/1.

Process INVOKE_SS_VLR

22.7.3_1(1)

Figure 22.7.3/1: Invocation of supplementary service procedure in VLR

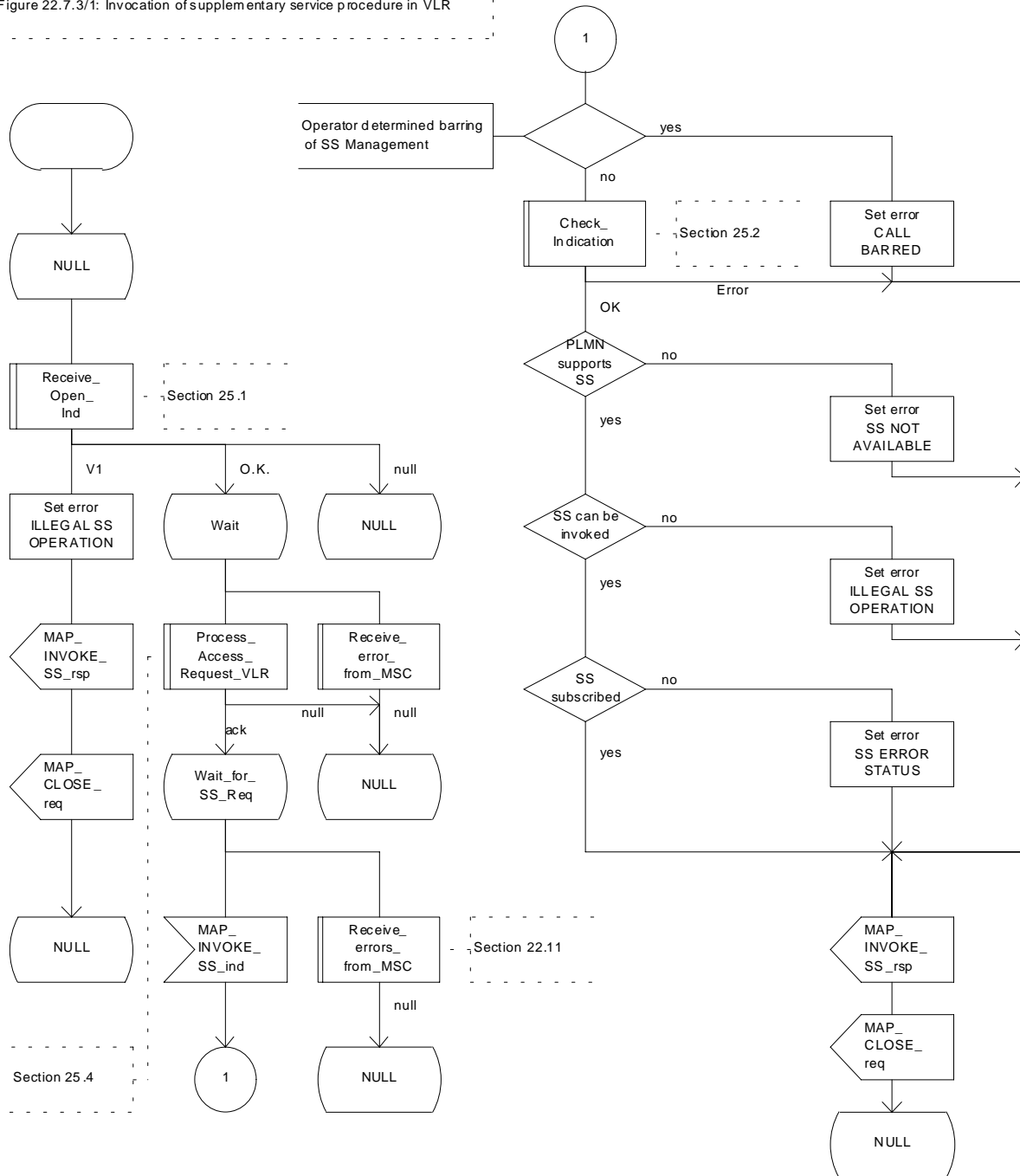


Figure 22.7.3/1: Procedure Invoke_SS_VLR

22.8 Password registration procedure

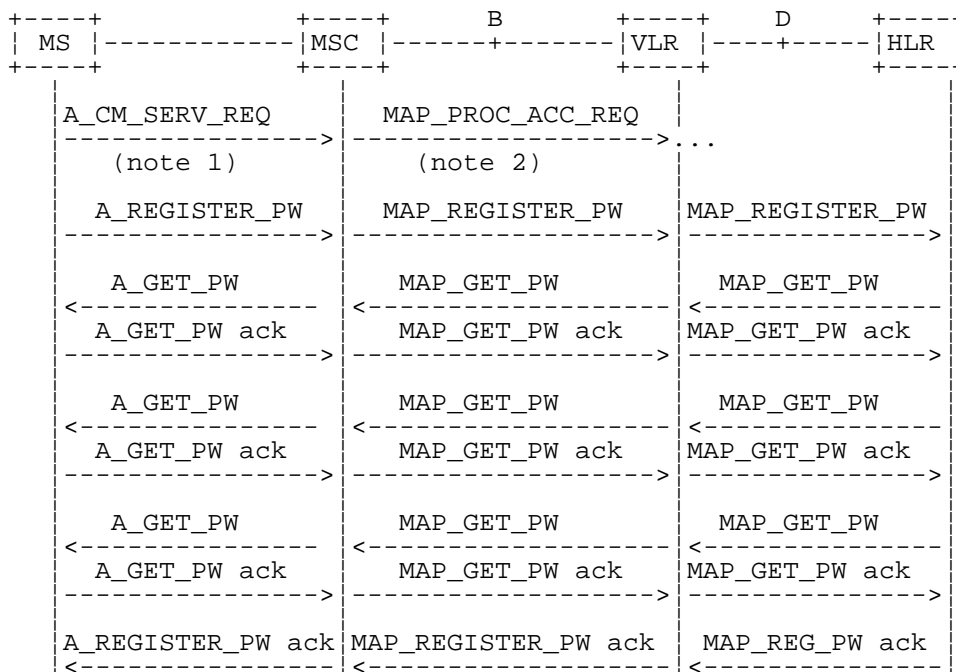
22.8.1 General

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 22.8.1/1.

The following services may be used:

- MAP_PROCESS_ACCESS_REQUEST (defined in clauses 8 and 25);
- MAP_TRACE_SUBSCRIBER_ACTIVITY (defined in clauses 9 and 25);
- MAP_PROVIDE_IMSI (defined in clauses 8 and 25);
- MAP_FORWARD_NEW_TMSI (defined in clauses 8 and 25);
- MAP_AUTHENTICATE (defined in clauses 8 and 25);
- MAP_SET_CIPHERING_MODE (defined in clauses 8 and 25);
- MAP_CHECK_IMEI (defined in clauses 8 and 25);
- MAP_READY_FOR_SM (defined in clauses 12 and 25);
- MAP_GET_PASSWORD (defined in clause 11).



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 25 in the present document.

NOTE 3: Use of each of the three MAP_GET_PASSWORD operations is described in subclause 22.8.4.

Figure 22.8.1/1: Interfaces and services for supplementary service password registration

22.8.2 Procedures in the MSC

The password registration procedure in the MSC is identical to that for activation specified in subclause 22.4.2. All the text and diagrams in subclause 22.4.2 apply with all references to activation changed to password registration.

22.8.3 Procedures in the VLR

The password registration procedure in the VLR is identical to that for activation specified in subclause 22.4.3. All the text and diagrams in subclause 22.4.3 apply with all references to activation changed to password registration.

22.8.4 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 22.2.4 above.

The password registration procedure in the HLR is shown in figure 22.8.4/1.

Process REGISTER_PASSWORD_HLR

22.8.4_1.1(2)

Figure 22.8.4/1: Registration of supplementary service password procedure in HLR

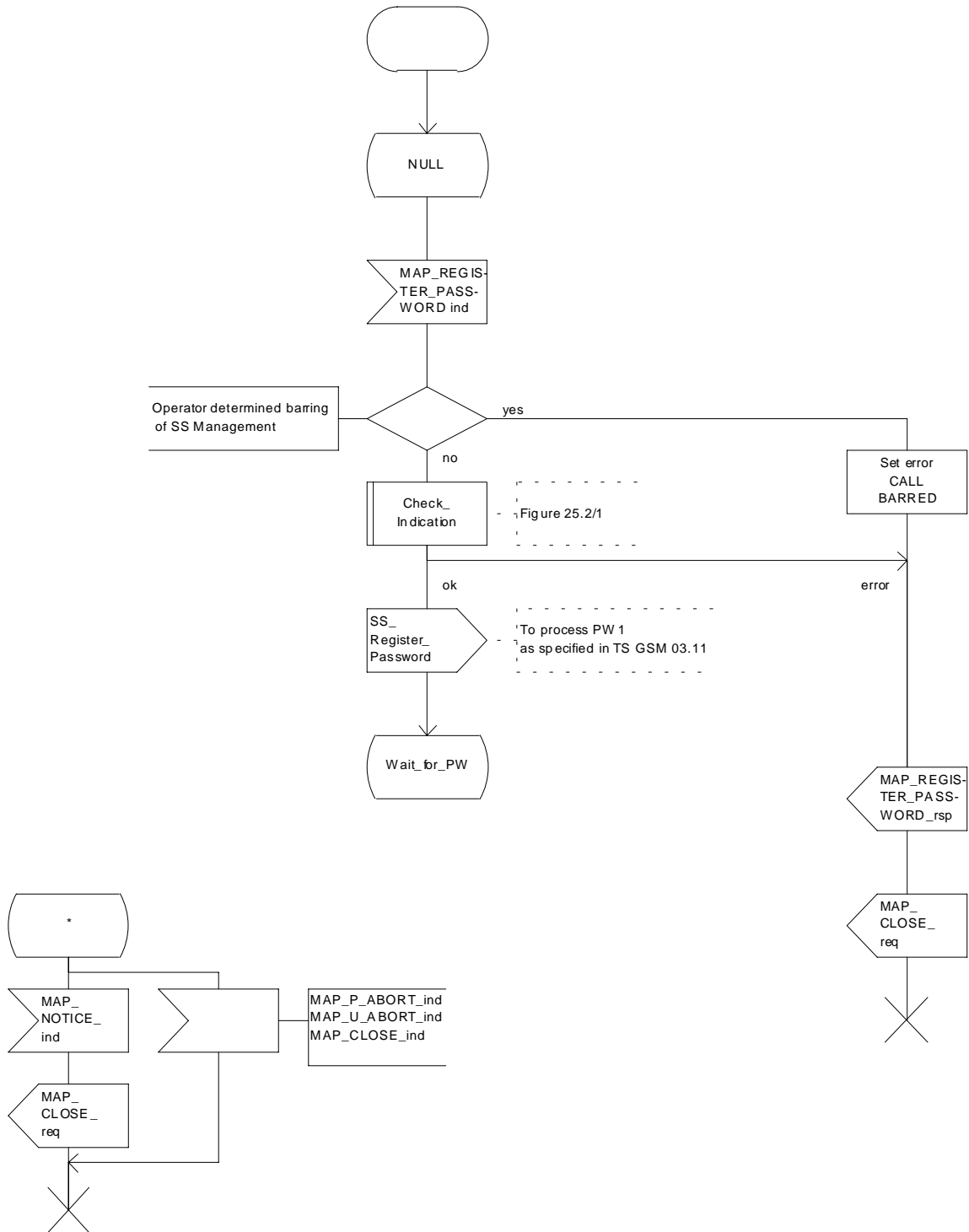


Figure 22.8.4/1 (sheet 1 of 2): Procedure Register_PW_HLR

Process REGISTER_PASSWORD_HLR

22.8.4_1.2(2)

Figure 22.8.4/1: Registration of supplementary service password procedure in HLR

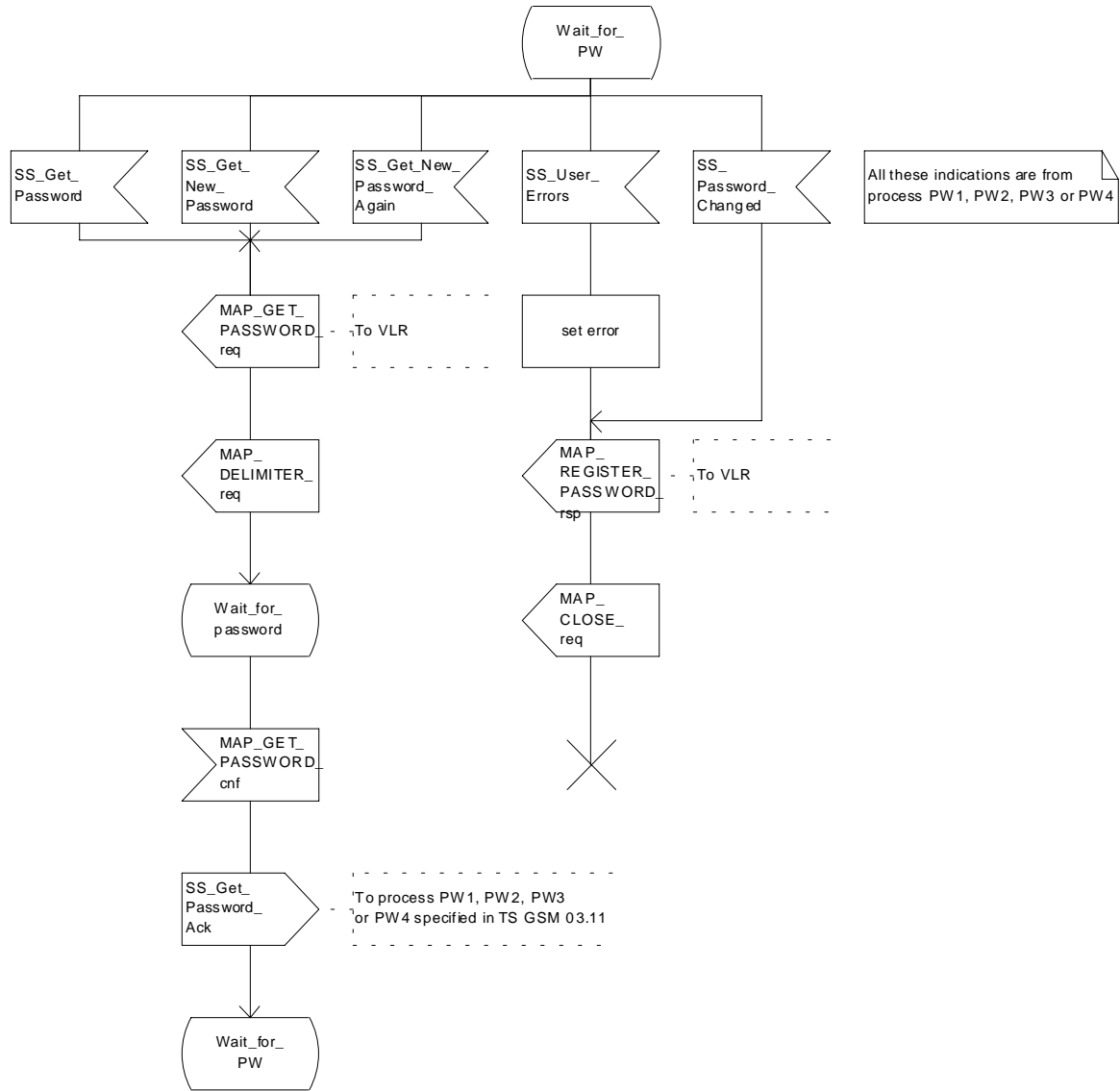


Figure 22.8.4/1 (sheet 2 of 2): Procedure Register_PW_HLR

22.9 Mobile Initiated USSD procedure

22.9.1 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_PROCESS_ACCESS_REQUEST	(defined in clauses 8 and 25);
MAP_TRACE_SUBSCRIBER_ACTIVITY	(defined in clauses 9 and 25);
MAP_PROVIDE_IMSI	(defined in clauses 8 and 25);
MAP_FORWARD_NEW_TMSI	(defined in clauses 8 and 25);
MAP_AUTHENTICATE	(defined in clauses 8 and 25);
MAP_SET_CIPHERING_MODE	(defined in clauses 8 and 25);
MAP_CHECK_IMEI	(defined in clauses 8 and 25);
MAP_READY_FOR_SM	(defined in clauses 12 and 25);
MAP_UNSTRUCTURED_SS_REQUEST	(defined in clause 11);
MAP_UNSTRUCTURED_SS_NOTIFY	(defined in clause 11).

The following service is certainly used:

MAP_PROCESS_UNSTRUCTURED_SS_REQUEST (defined in clause 11).

22.9.2 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 22.9.2/1.

Process MS_INIT_USSD_MSC

22.9.2_1.1(2)

Figure 22.9.2/1: Handling of mobile initiated USSD at MSC

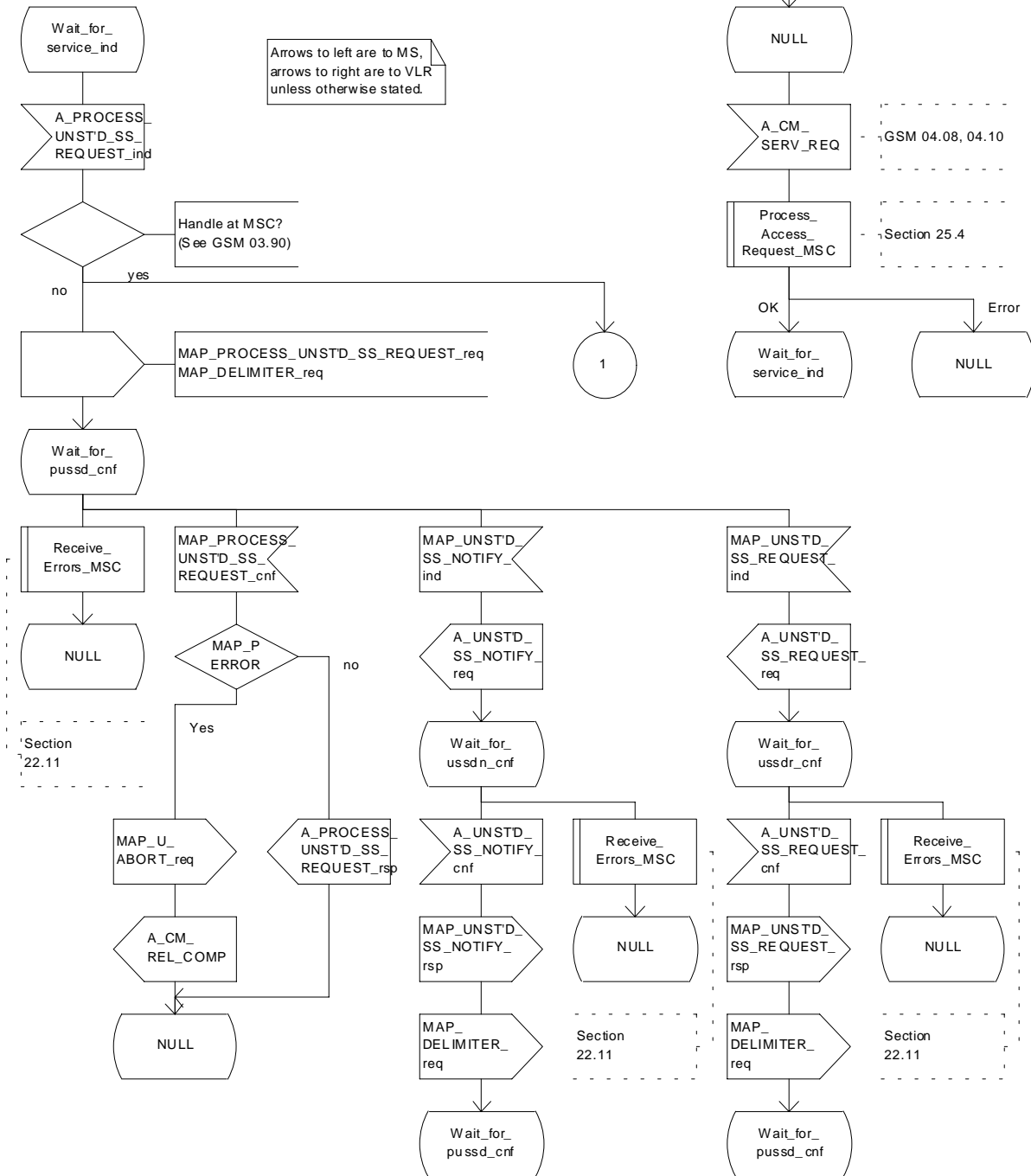


Figure 22.9.2/1 (sheet 1 of 2): Procedure MI_USSD_MSC

Process MS_INIT_USSD_MSC

22.9.2_1.2(2)

Figure 22.9.2/1: Handling of mobile initiated USSD at MSC

Arrows to left are to MS, arrows to right are to USSD application unless otherwise stated.

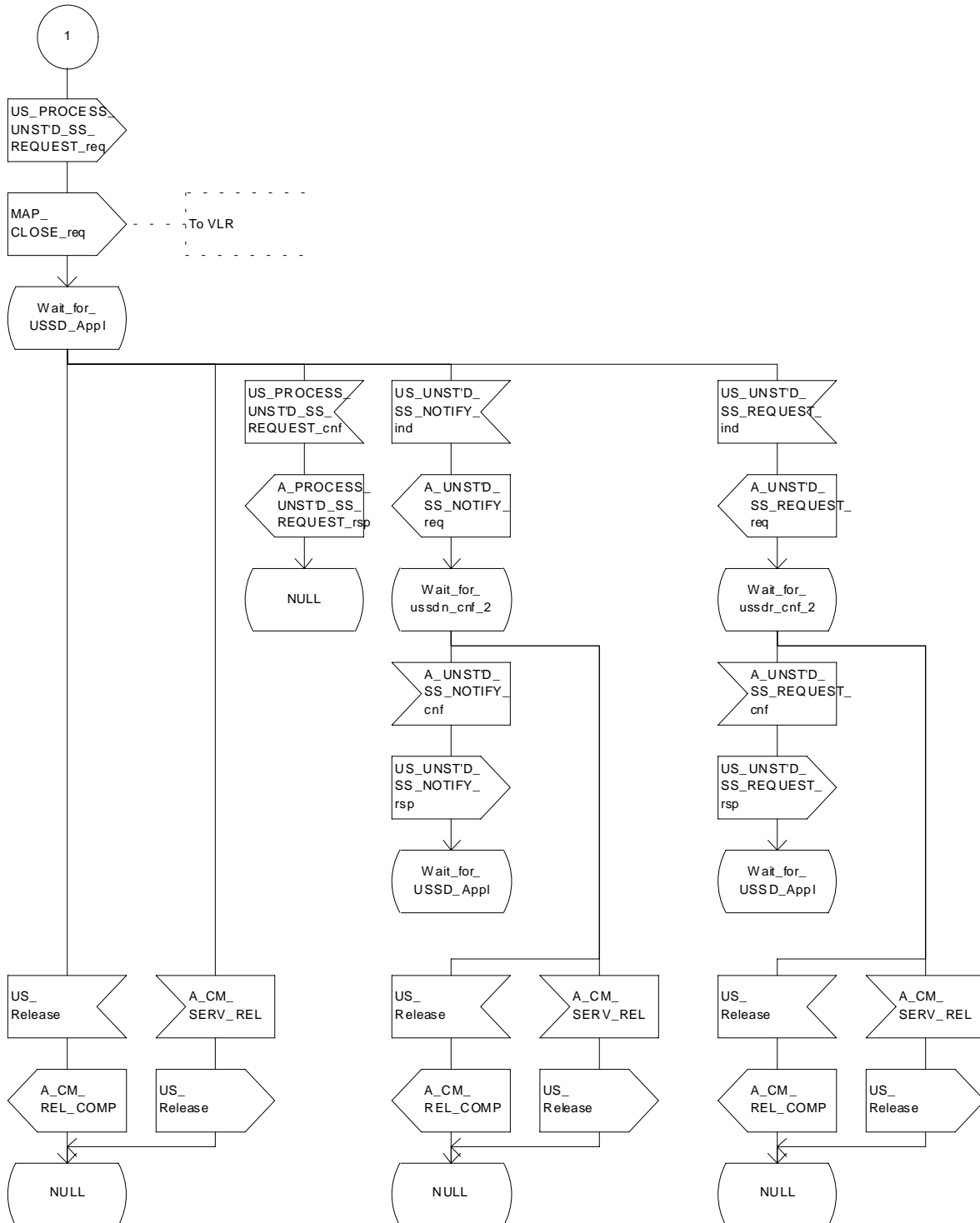


Figure 22.9.2/1 (sheet 2 of 2): Procedure MI_USSD_MSC

22.9.3 Procedures in the VLR

The initiation of the process is shown in subclause 22.1.2.

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 22.9.3/1 and 22.9.3/2.

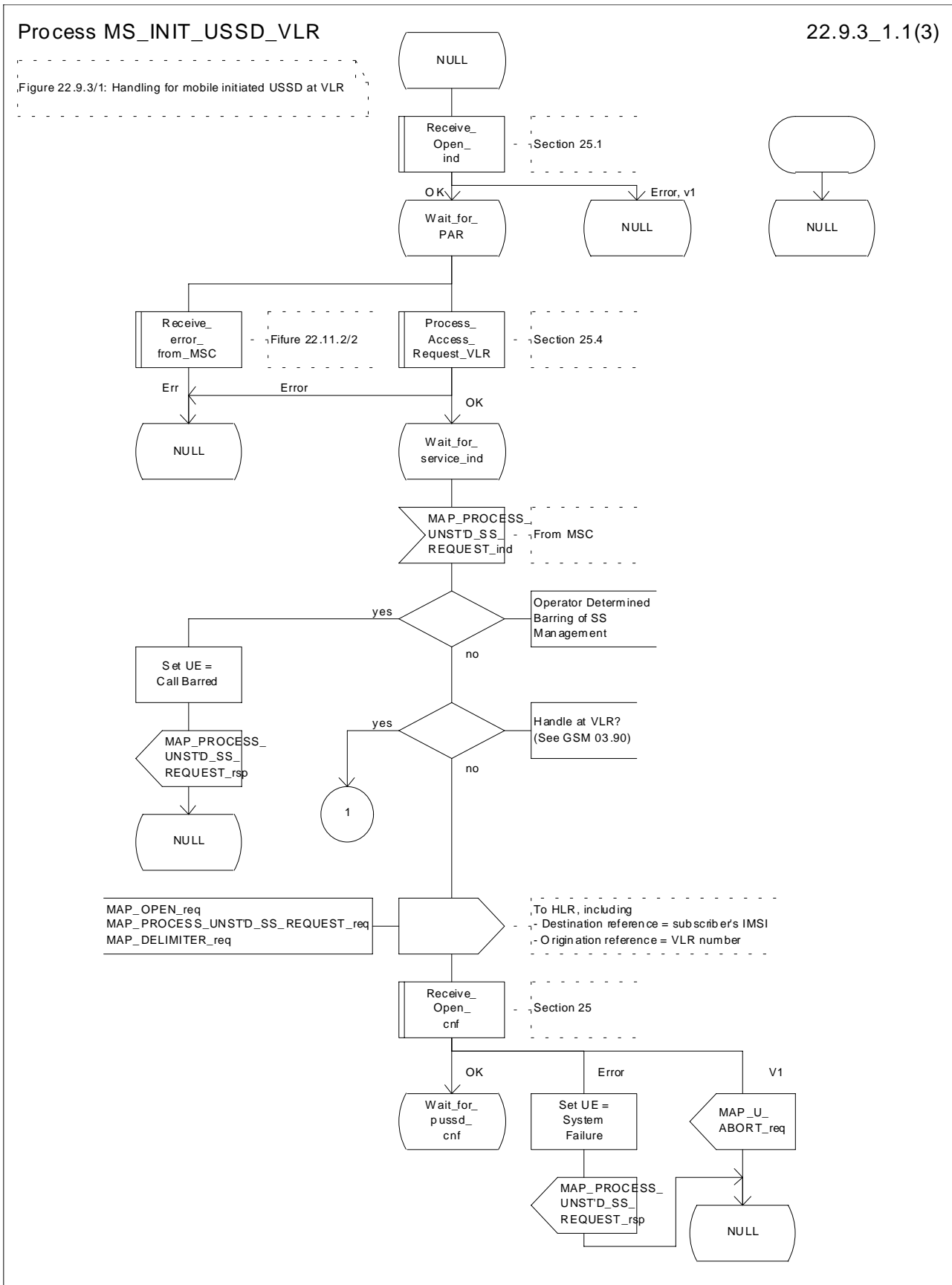


Figure 22.9.3/1 (sheet 1 of 3): Procedure MI_USSD_VLR

Process MS_INIT_USSD_VLR

22.9.3_1.2(3)

Figure 22.9.3/1: Handling for mobile initiated USSD at VLR

Arrows to left are to MSC, arrows to right are to HLR unless otherwise stated.

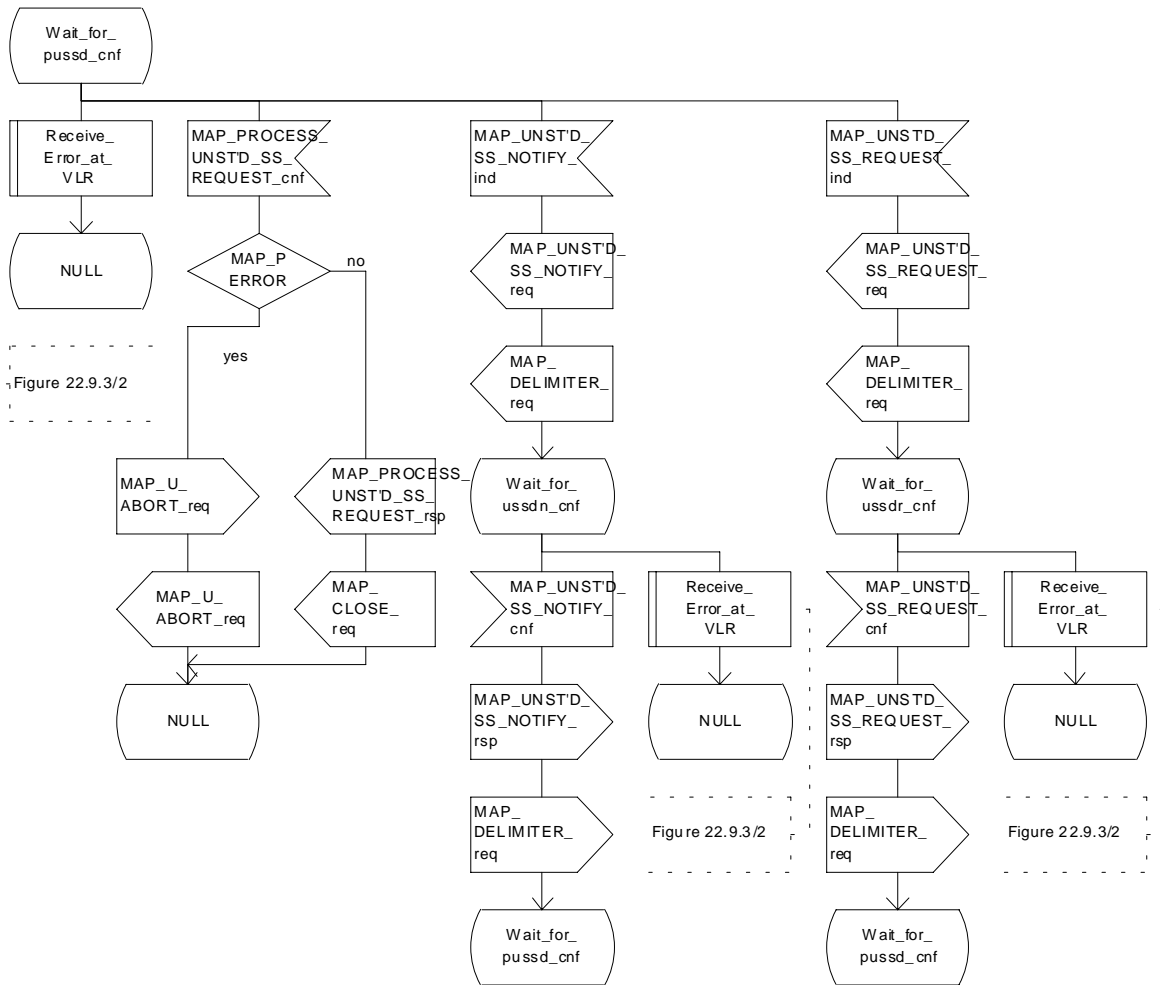


Figure 22.9.3/1 (sheet 2 of 3): Procedure MI_USSD_VLR

Process MS_INIT_USSD_VLR

22.9.3_1.3(3)

Figure 22.9.3/1: Handling for mobile initiated USSD at VLR

Arrows to left are to MSC, arrows to right are to USSD application unless otherwise stated.

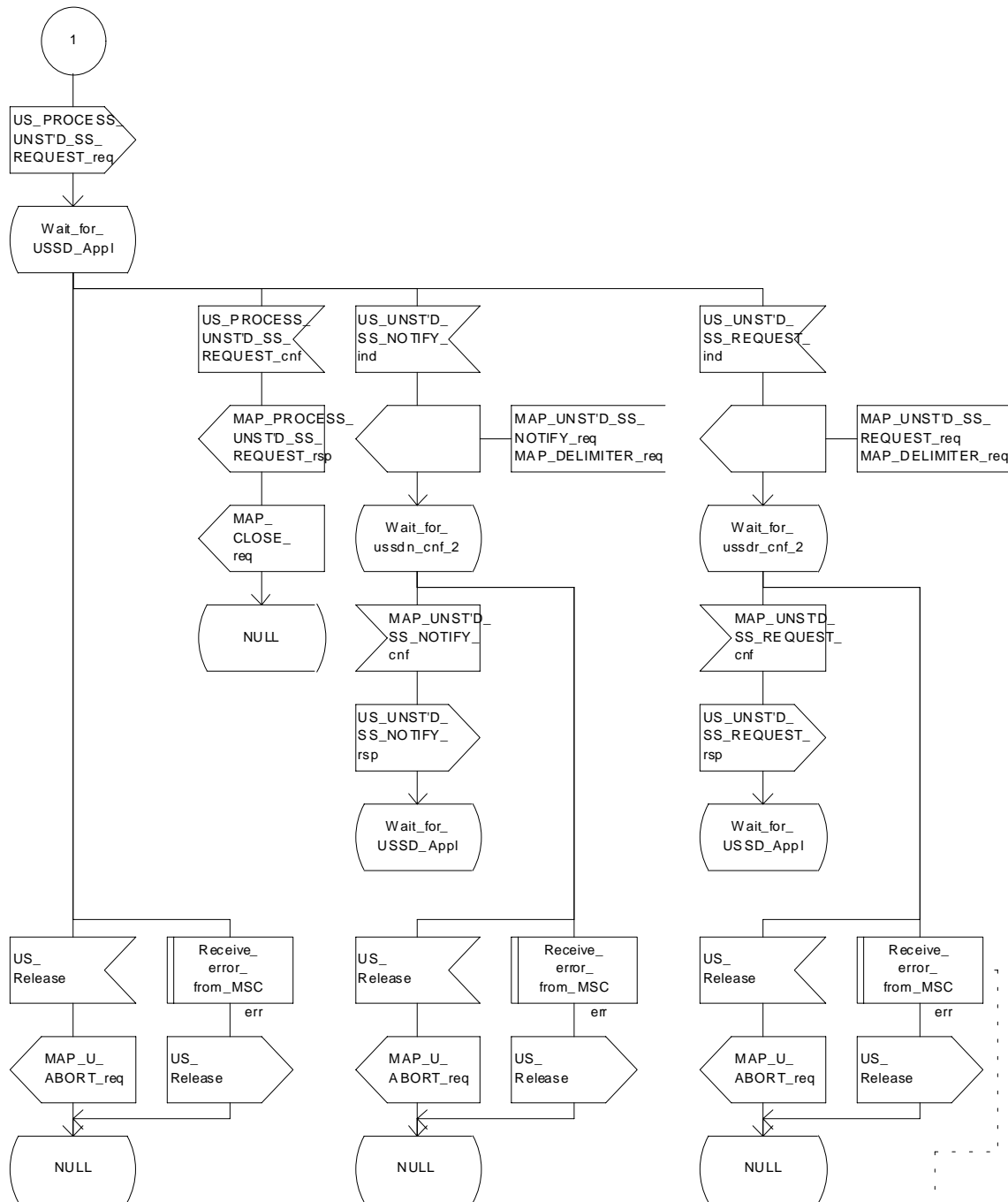


Figure 22.1.1.2/2

Figure 22.9.3/1 (sheet 3 of 3) : Procedure_MI_USSD_VLR

Macrodefinition Receive_Error_at_VLR

22.9.3_2(1)

Figure 22.9.3/2: Handling of errors at VLR for USSD

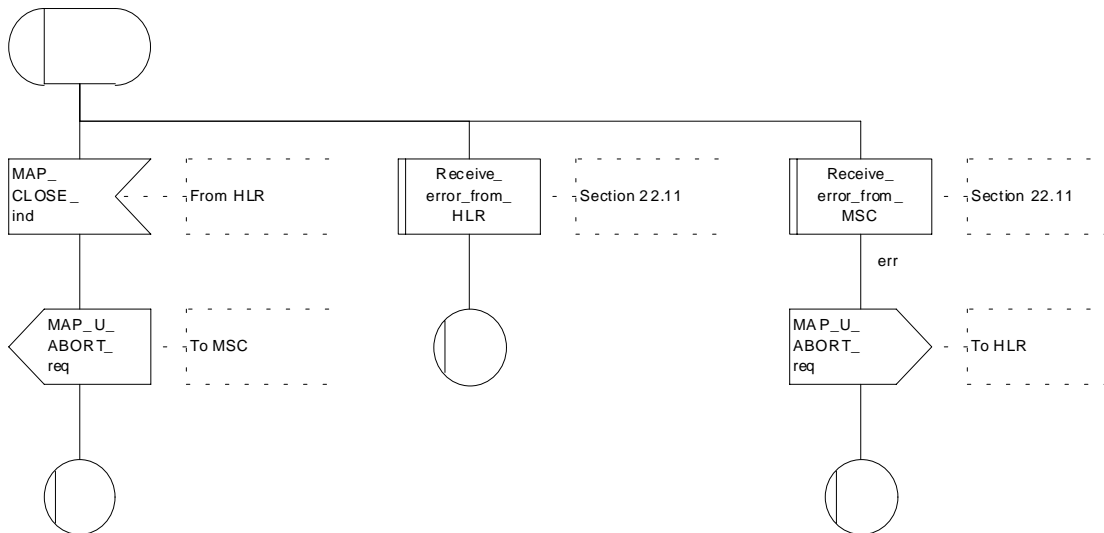


Figure 22.9.3/2: Macro Receive_Error_at_VLR

22.9.4 Procedures in the HLR

The initiation of the process is shown in subclause 22.1.3.

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 either be fed to an application contained locally in the HLR or to the gsmSCF or to a secondary HLR where the USSD application is located. If the alphabet is not understood then the error "UnknownAlphabet" shall be returned.

Message Destined for Local Application

If the message is destined for the local USSD application then the HLR shall transfer the message to the local application.

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.

Message Destined for gsmSCF or secondary HLR

If the message is destined for the gsmSCF or secondary HLR then the primary HLR shall transfer the message transparently to the next node.

The primary HLR may subsequently receive one or more MAP_UNSTRUCTURED_SS_REQUEST or MAP_UNSTRUCTURED_SS_NOTIFY indications from the gsmSCF. These shall be sent transparently to the VLR. When a confirmation is received from the VLR this shall be returned to the gsmSCF.

When the primary HLR receives a MAP_PROCESS_UNSTRUCTURED_SS_REQUEST confirmation from the gsmSCF then it shall pass this to the VLR and closes the MAP provider service.

Error Handling

The VLR, the USSD Application and the gsmSCF or secondary HLR may initiate release of the MAP service at any time. This is handled as shown in the diagrams.

The procedure in the primary and secondary HLR is shown in figure 22.9.4/1.

Process MS_INIT_USSD_HLR

22.9.4_1.1(4)

Figure 22.9.4/1: Handling of mobile initiated USSD at HLR.

Arrows to left are to VLR unless otherwise stated.
Arrow to right are to USSD application unless otherwise stated

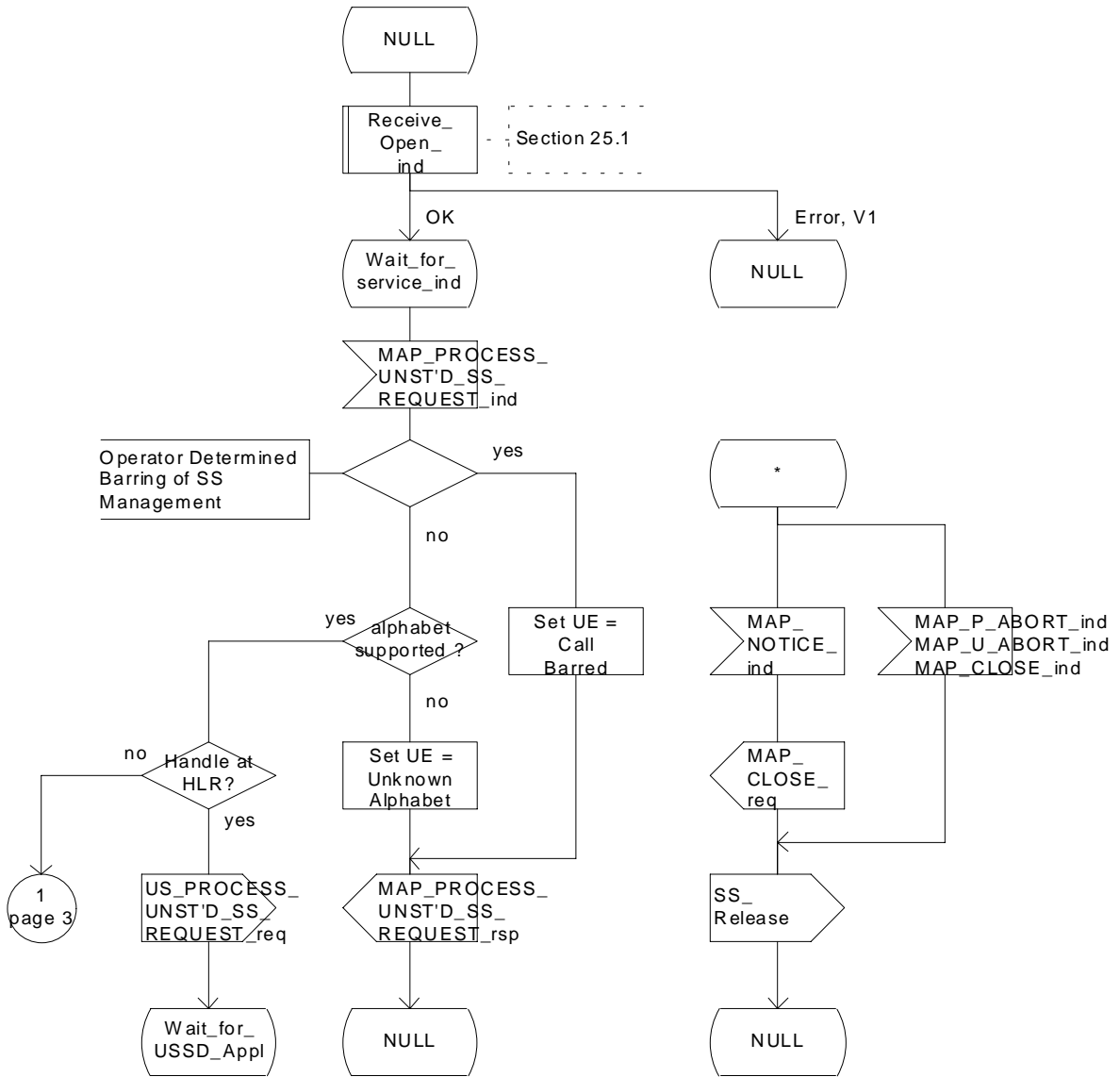


Figure 22.9.4/1 (sheet 1 of 4): Procedure MI_USSD_HLR

Process MS_INIT_USSD_HLR

22.9.4_1.2(4)

Figure 22.9.4/1: Handling of mobile initiated USSD at HLR.

Arrows to left are to VLR, arrows to right are to USSD application unless otherwise stated.

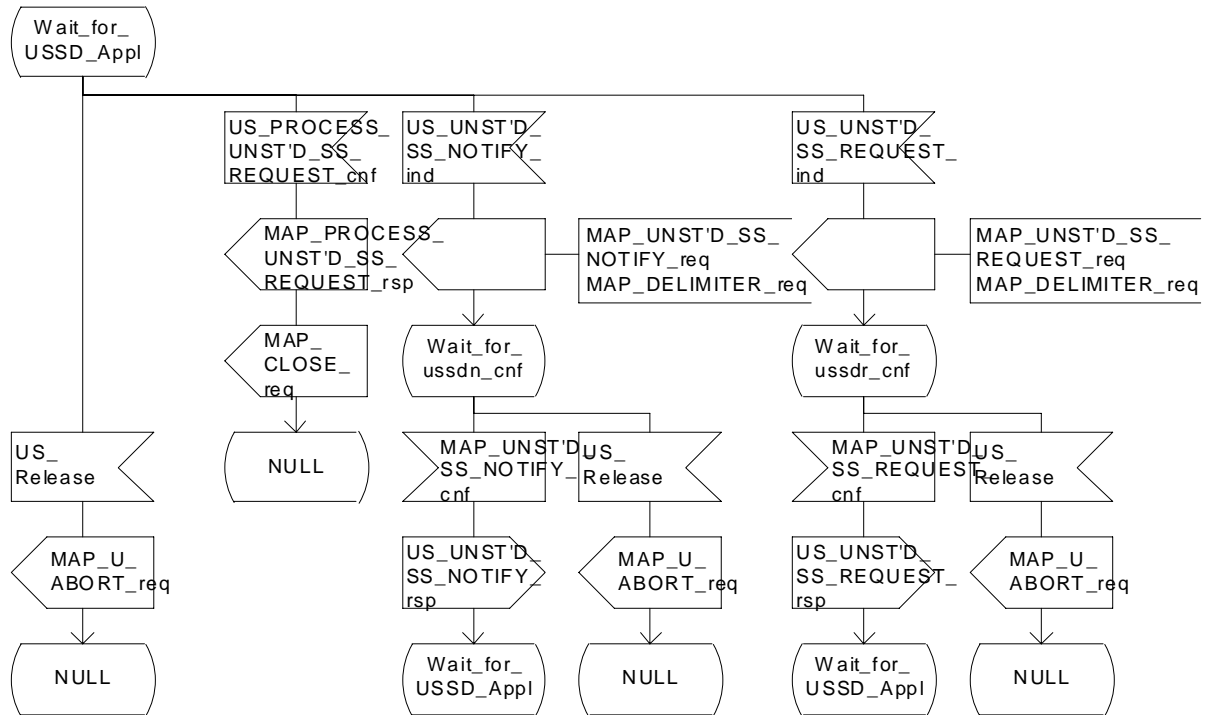


Figure 22.9.4/1 (sheet 2 of 4): Procedure MI_USSD_HLR

Process MS_INIT_USSD_HLR

22.9.4_1.3(4)

Figure 22.9.4/1: Handling of mobile initiated USSD at HLR.

Arrows to left are to VLR, arrows to right are to the next node unless otherwise stated.

Section 22.11

MAP_OPEN_req
MAP_PROCESS_UNSTD_SS_REQUEST_req
MAP_DELIMITER_req

To next node, including
- Destination reference = subscriber's IMSI
- Origination reference = HLR number

Receive_Open_cnf

Section 25

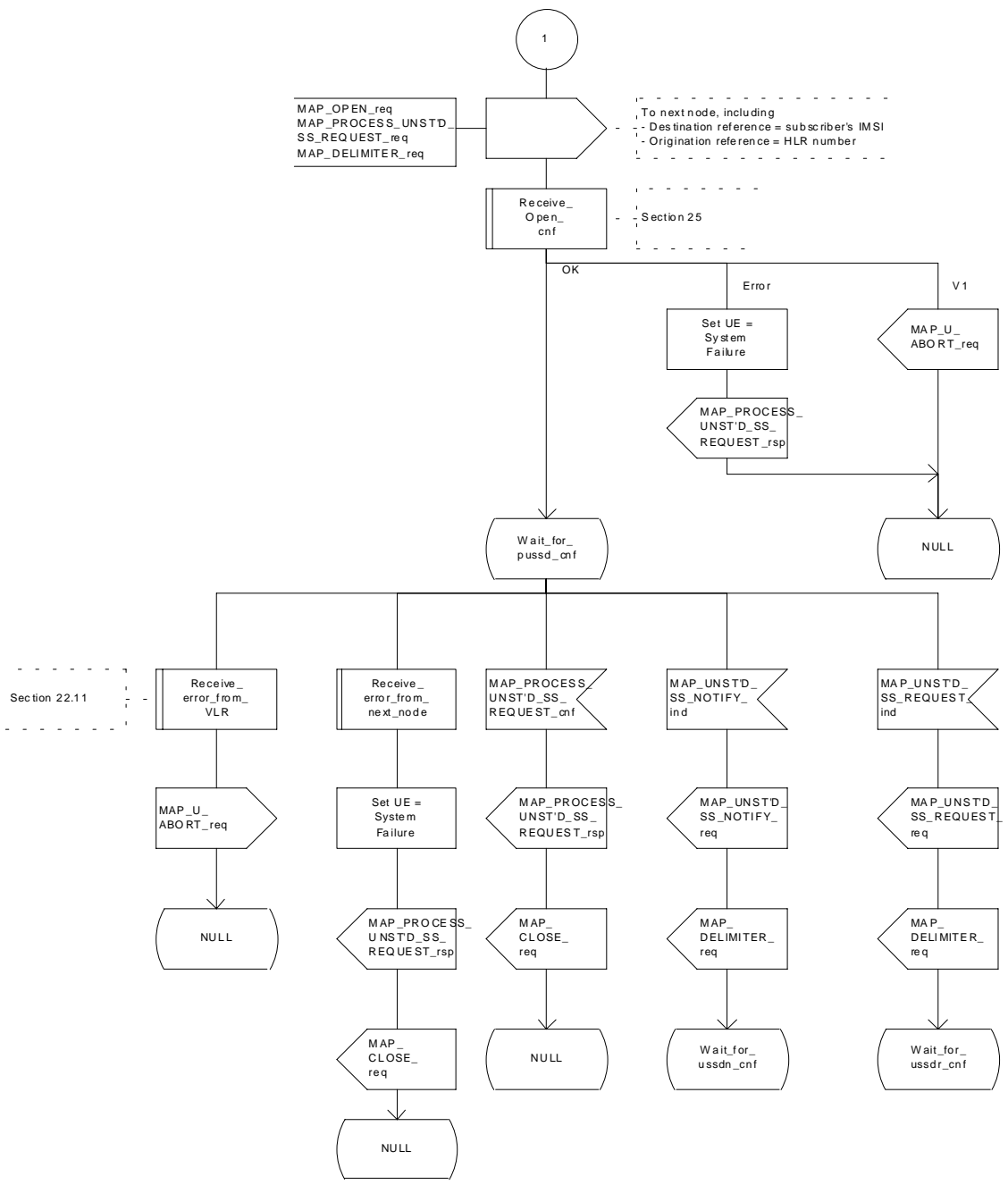


Figure 22.9.4/1 (sheet 3 of 4): Procedure MI_USSD_HLR

Process MS_INIT_USSD_HLR

22.9.4_1.4(4)

Figure 22.9.4/1: Handling of mobile initiated USSD at HLR.

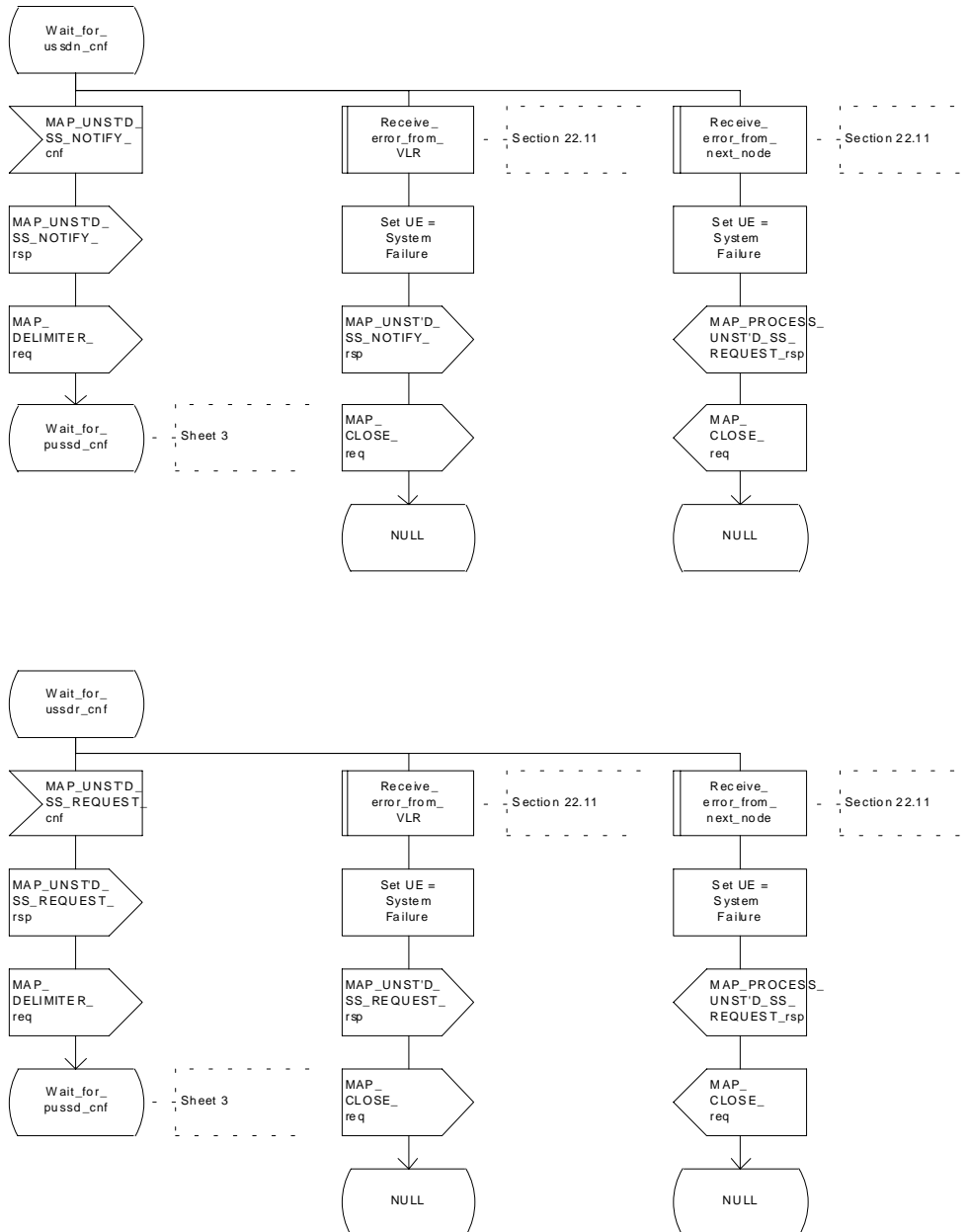


Figure 22.9.4/1 (sheet 4 of 4): Procedure MI_USSD_HLR

22.10 Network initiated USSD procedure

22.10.1 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 8 and 25);
MAP_SEARCH_FOR_MOBILE_SUBSCRIBER	(defined in clauses 8 and 25);
MAP_PROCESS_ACCESS_REQUEST	(defined in clauses 8 and 25);
MAP_AUTHENTICATE	(defined in clauses 8 and 25);
MAP_SET_CIPHERING_MODE	(defined in clauses 8 and 25);
MAP_FORWARD_NEW_TMSI	(defined in clauses 8 and 25);
MAP_READY_FOR_SM	(defined in clauses 12 and 25).

At least one of the following services will certainly be used, and both may be used:

MAP_UNSTRUCTURED_SS_REQUEST	(defined in clause 11);
MAP_UNSTRUCTURED_SS_NOTIFY	(defined in clause 11).

22.10.2 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 25.3). 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 22.10.2/1.

Process NW_INIT_USSD_MSC

22.10.2_1.1(4)

Figure 22.10.2/1: Handling of network initiated USSD in MSC

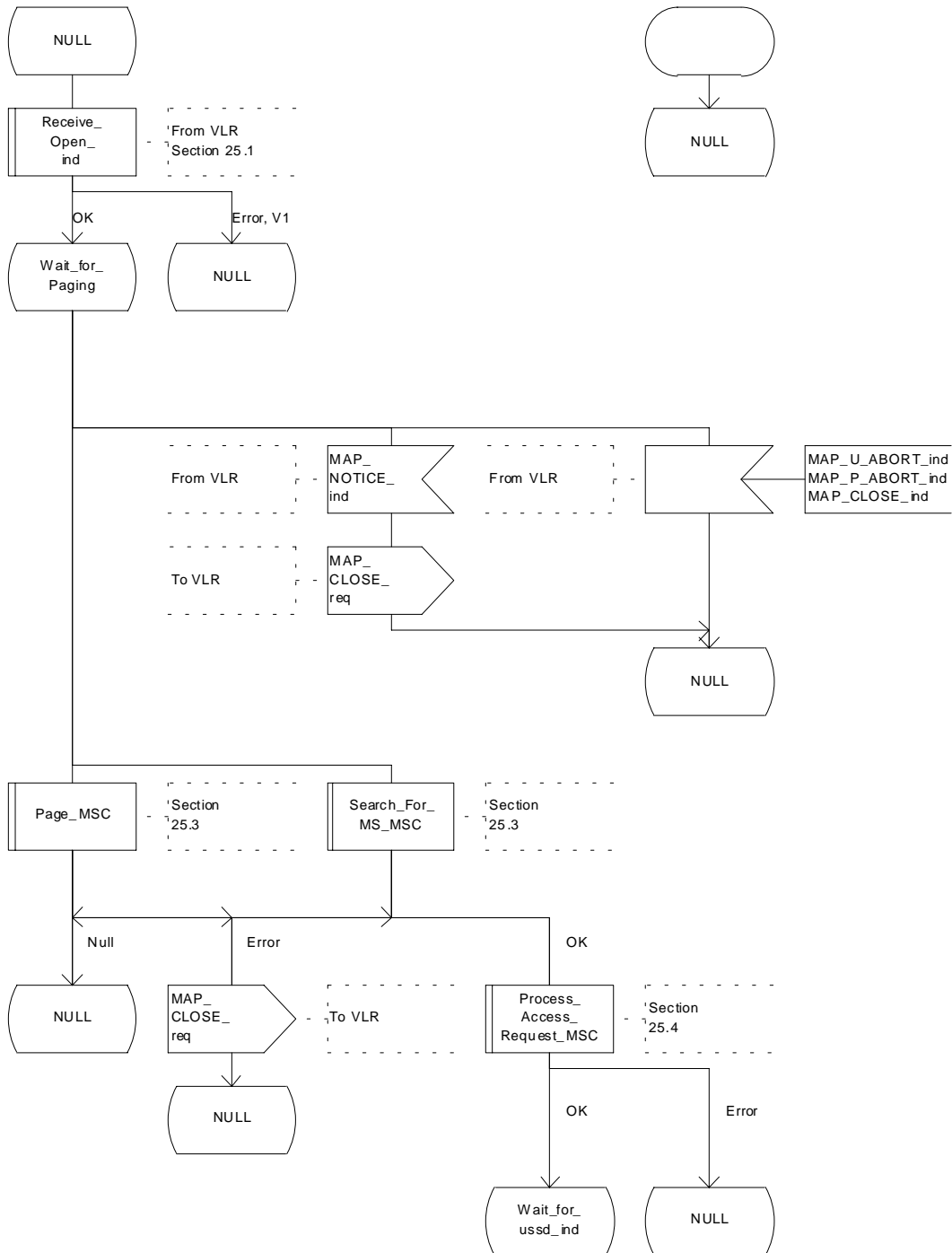


Figure 22.10.2/1 (sheet 1 of 4): Procedure NI_USSD_MSC

Process NW_INIT_USSD_MSC

22.10.2_1.2(4)

Figure 22.10.2/1: Handling of network initiated USSD in MSC

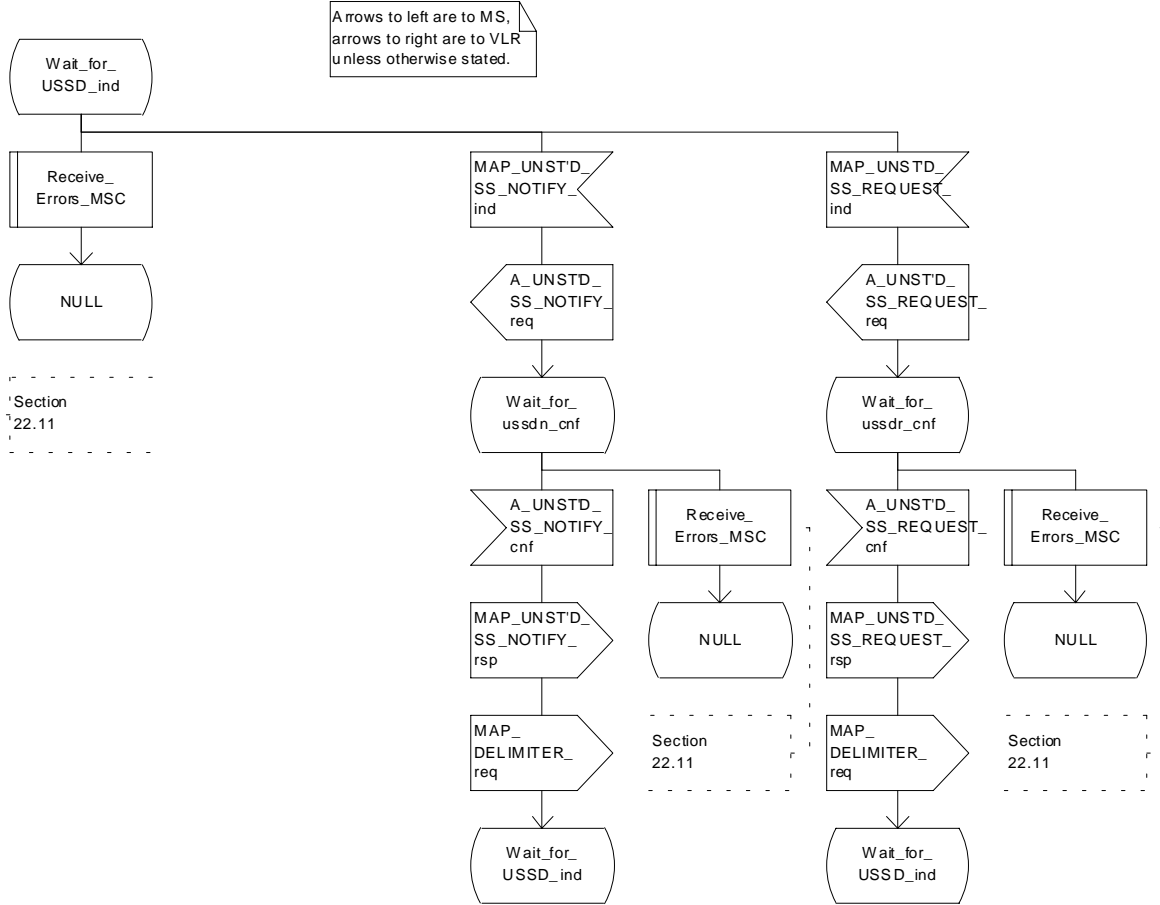


Figure 22.10.2/1 (sheet 2 of 4): Procedure NI_USSD_MSC

Process NW_INIT_USSD_MSC

22.10.2_1.3(4)

Figure 22.10.2/1: Handling of network initiated USSD in MSC

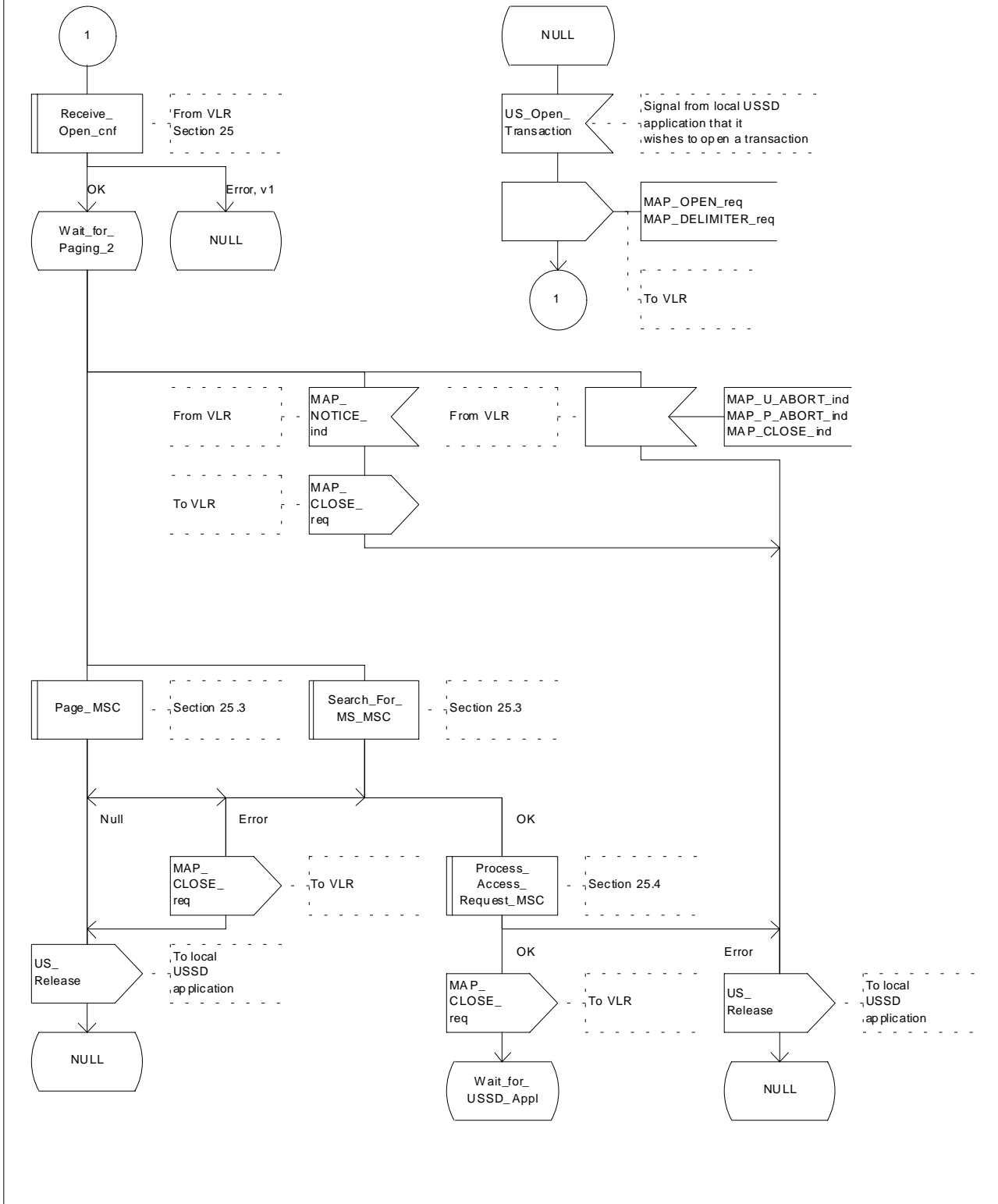


Figure 22.10.2/1 (sheet 3 of 4): Procedure NI_USSD_MSC

Process NW_INIT_USSD_MSC

22.10.2_1.4(4)

Figure 22.10.2/1: Handling of network initiated USSD in MSC

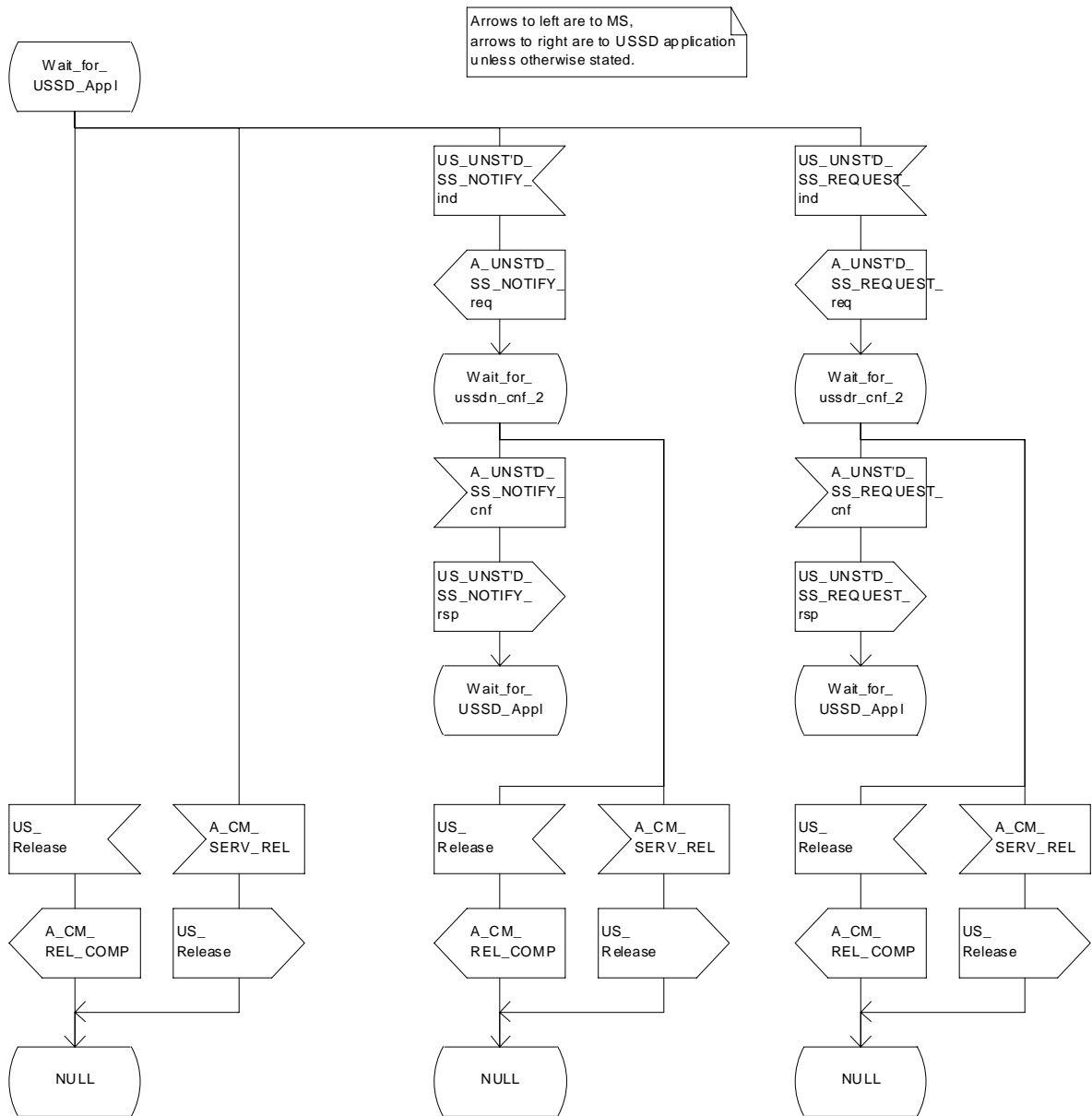


Figure 22.10.2/1 (sheet 4 of 4): Procedure NI_USSD_MSC

22.10.3 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 22.10.3/1.

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 25.4 will be used to handle the CM connection establishment.

The macro is shown in figure 22.10.3/2.

Process NW_INIT_USSD_VLR

22.10.3_1.1(4)

Figure 22.10.3/1: Handling of network initiated USSD at VLR

Arrows to left are to MSC, Arrows to right are to HLR unless otherwise stated.

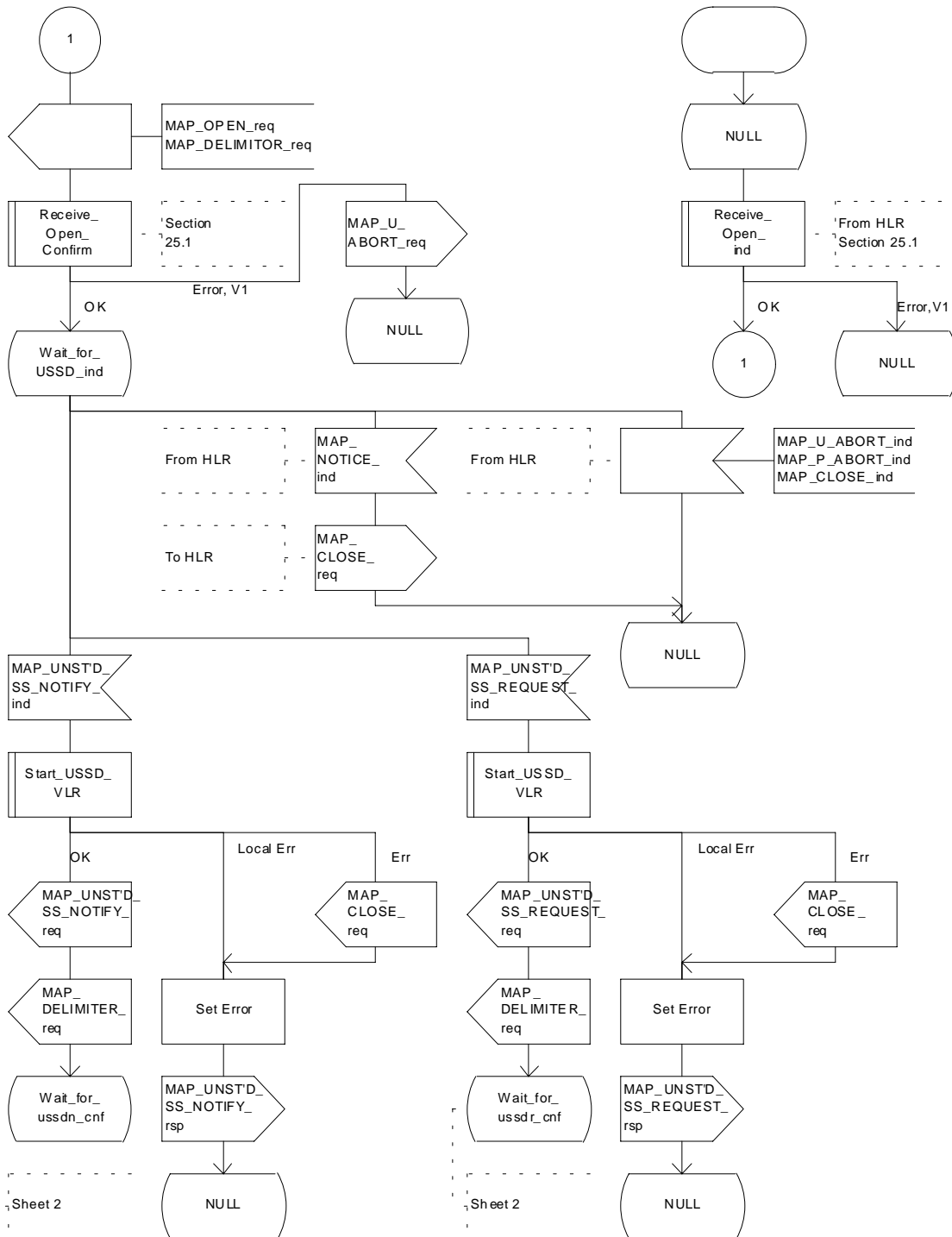


Figure 22.10.3/1 (sheet 1 of 4): Procedure NI_USSD_VLR

Process NW_INIT_USSD_VLR

22.10.3_1.2(4)

Figure 22.10.3/1: Handling of network initiated USSD at VLR

Arrows to left are to MSC, arrows to right are to HLR unless otherwise stated.

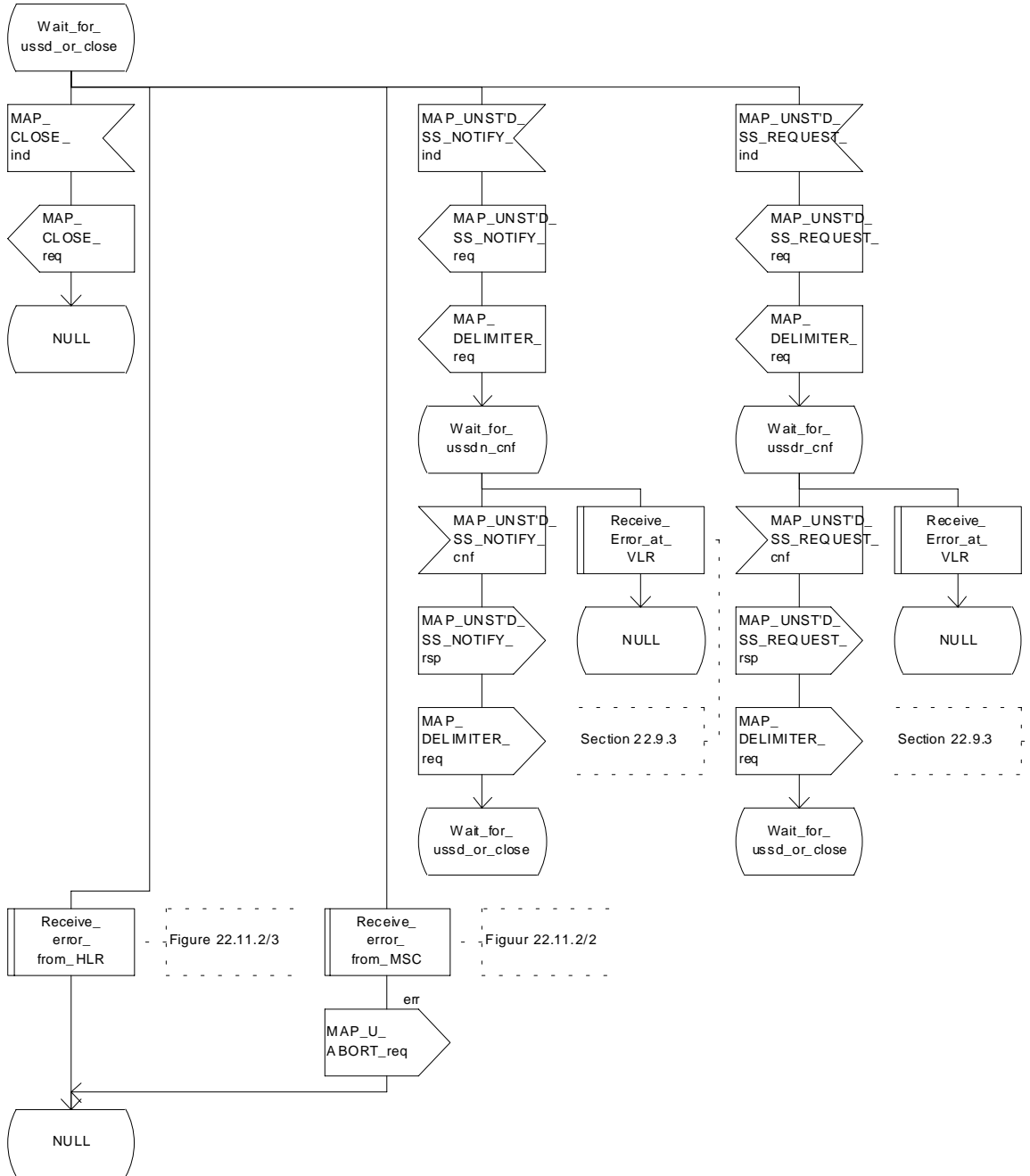


Figure 22.10.3/1 (sheet 2 of 4): Procedure NI_USSD_VLR

Process NW_INIT_USSD_VLR

22.10.3_1.3(4)

Figure 22.10.3/1: Handling of network initiated USSD at VLR

Arrows to left are to MSC, arrows to right are to USSD application unless otherwise stated.

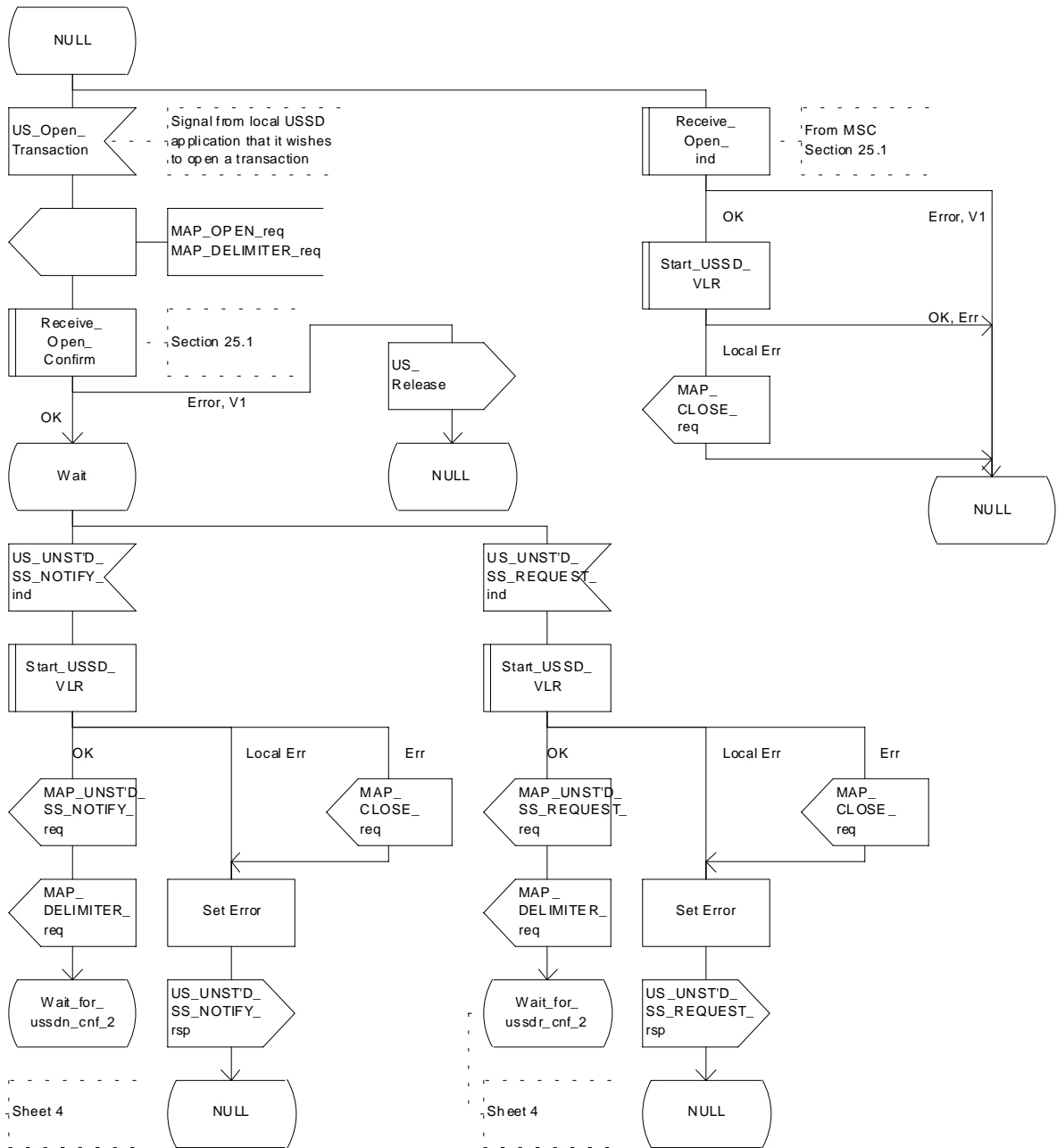


Figure 22.10.3/1 (sheet 3 of 4): Procedure NI_USSD_VLR

Process NW_INIT_USSD_VLR

22.10.3_1.4(4)

Figure 22.10.3/1: Handling of network initiated USSD at VLR

Arrows to left are to MSC, arrows to right are to USSD application unless otherwise stated.

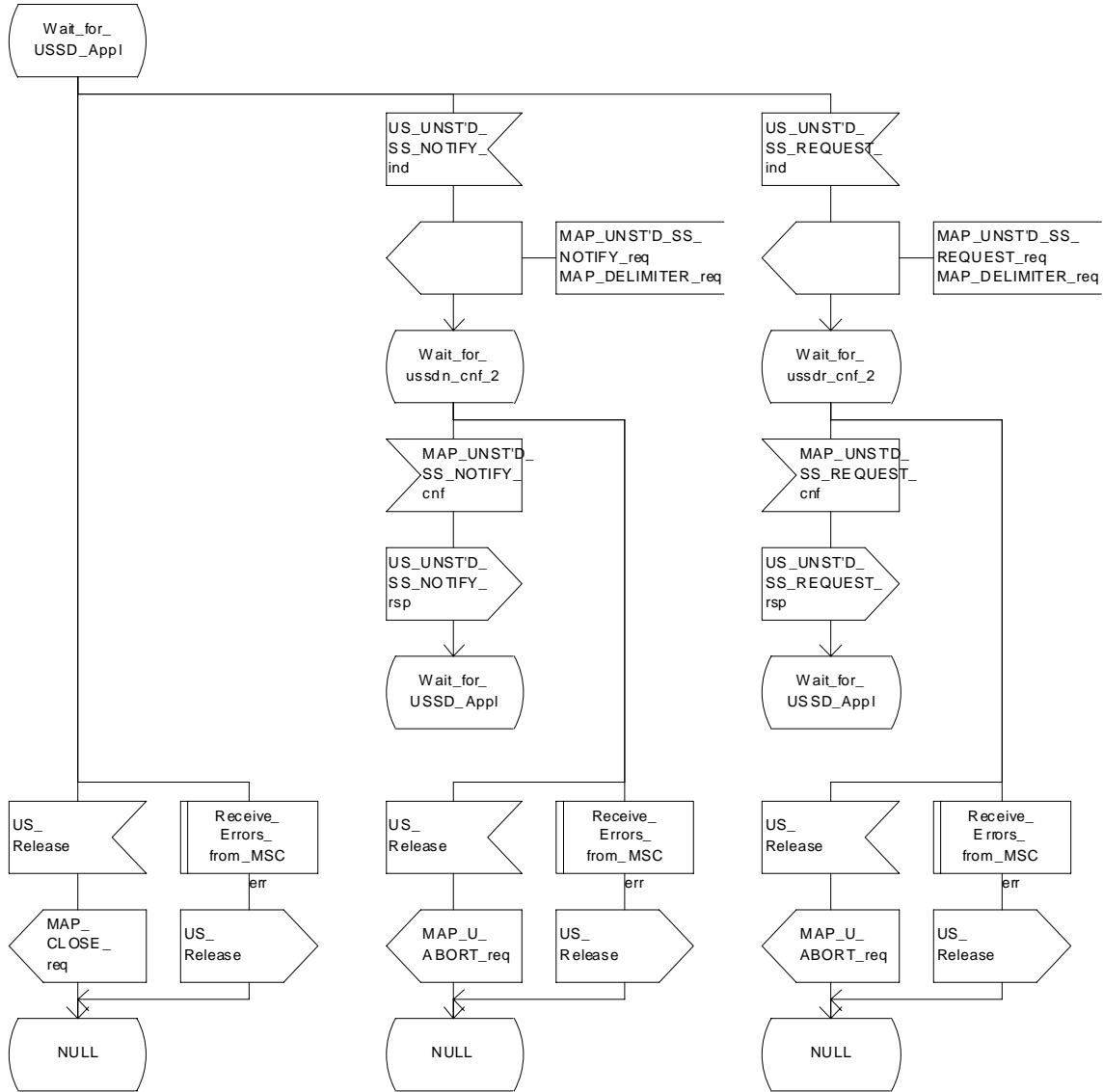


Figure 22.10.3/1 (sheet 4 of 4): Procedure NI_USSD_VLR

Macrodefinition Start_USSD_VLR

22.10.3_2.1(2)

Figure 22.10.3/2: Macro to establish a connection to the MS for a network initiated USSD operation.

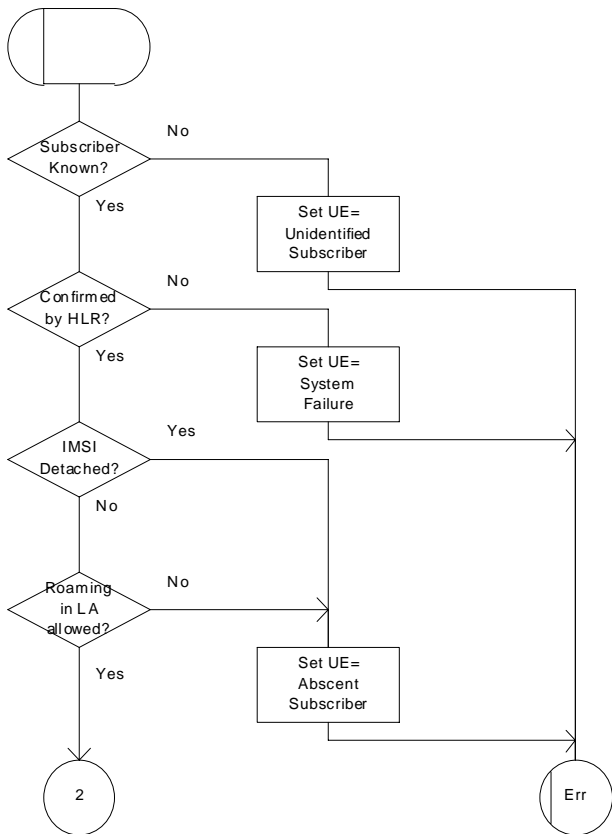


Figure 22.10.3/2 (sheet 1 of 2): Macro Start_USSD_VLR

Macrodefinition Start_USSD_VLR

22.10.3_2.2(2)

Figure 22.10.3/2: Macro to establish a connection to the MS for a network initiated USSD operation.

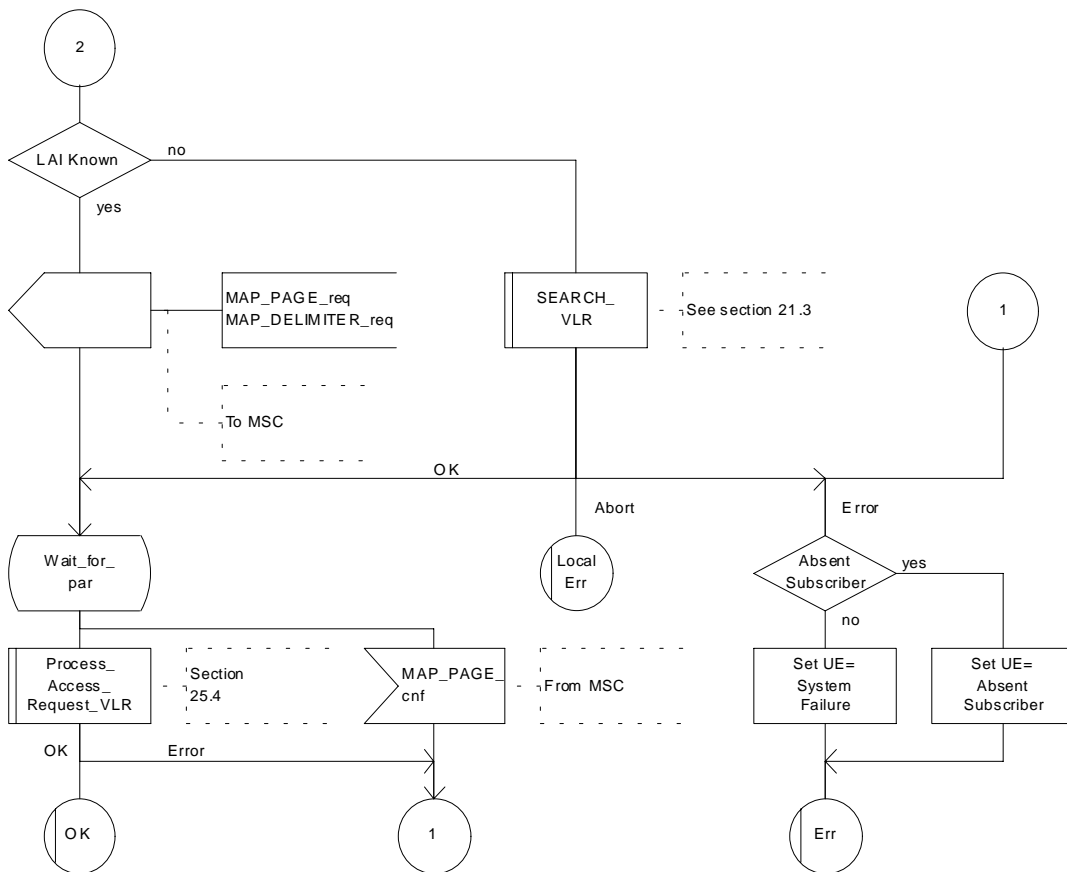


Figure 22.10.3/2 (sheet 2 of 2): Macro Start_USSD_VLR

22.10.4 Procedure in the HLR

The procedure may be invoked either by a gsmSCF, a secondary HLR or by a USSD application local to the primary HLR. It may start by using either the MAP_UNSTRUCTURED_SS_REQUEST or MAP_UNSTRUCTURED_SS_NOTIFY service.

In both cases the primary HLR will first check whether the MS is reachable .

If the MS is reachable, the primary HLR will initiate a MAP dialogue with the VLR. Once the dialogue is successfully established the message received from the gsmSCF or secondary HLR or USSD application will be sent to the VLR.

Following transfer of the message the primary HLR will wait for a confirmation from the VLR. This will be sent to the gsmSCF or secondary HLR or USSD application as appropriate.

Following this, the primary 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 and if necessary the MAP process with the gsmSCF or secondary HLR shall be aborted, as shown in the diagram.

Message Originated by gsmSCF or secondary HLR

If the message is originated by the gsmSCF or a secondary HLR then the primary HLR shall transfer the message transparently to the VLR.

The primary HLR may subsequently receive one or more MAP_UNSTRUCTURED_SS_REQUEST_ind or MAP_UNSTRUCTURED_SS_NOTIFY_ind indications from the gsmSCF or secondary HLR. These shall be sent transparently to the VLR. When a confirmation is received from the VLR this shall be returned to the next node as appropriate.

When the primary HLR receives a MAP_CLOSE_ind from the gsmSCF or secondary HLR then it shall pass this to the VLR and close the MAP dialogue.

The procedure in the primary and secondary HLR is shown in figure 22.10.4/1 and 22.10.4/2.

Process NW_INIT_USSD_HLR

22.10.4_1.1(5)

Figure 22.10.4/1 Handling of network initiated USSD at HLR

Arrows to left are to VLR.
Arrows to right are to the next node
unless otherwise stated.

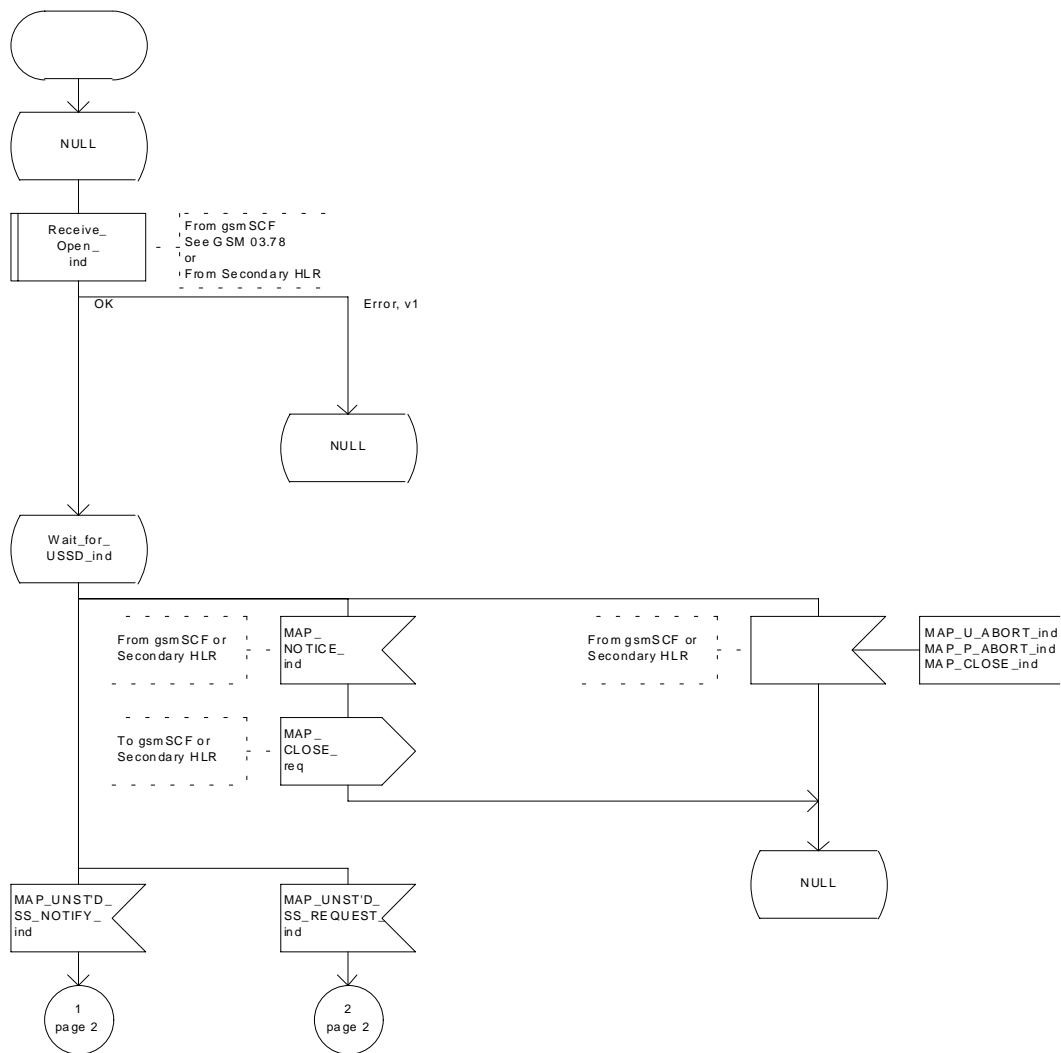


Figure 22.10.4/1 (sheet 1 of 5): Procedure NI_USSD_HLR

Process NW_INIT_USSD_HLR

22.10.4_1.2(5)

Figure 22.10.4/1 Handling of network initiated USSD at HLR

Arrows to left are to VLR, Arrows to right are to the next node unless otherwise stated.

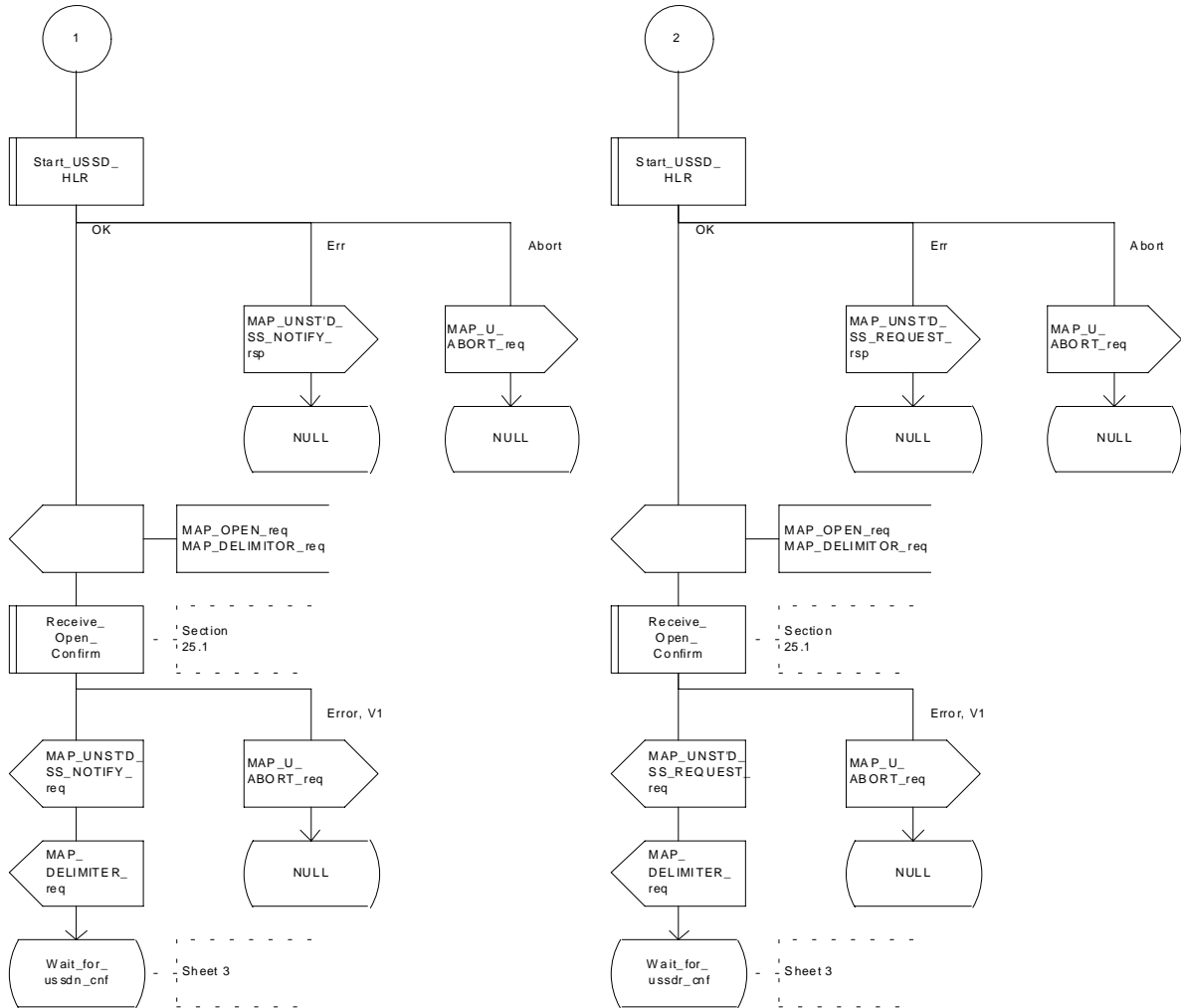


Figure 22.10.4/1 (sheet 2 of 5): Procedure NI_USSD_HLR

Process NW_INIT_USSD_HLR

22.10.4_1.3(5)

Figure 22.10.4/1 Handling of network initiated USSD at HLR

Arrows to left are to VLR,
Arrows to right are to the next node
unless otherwise stated.

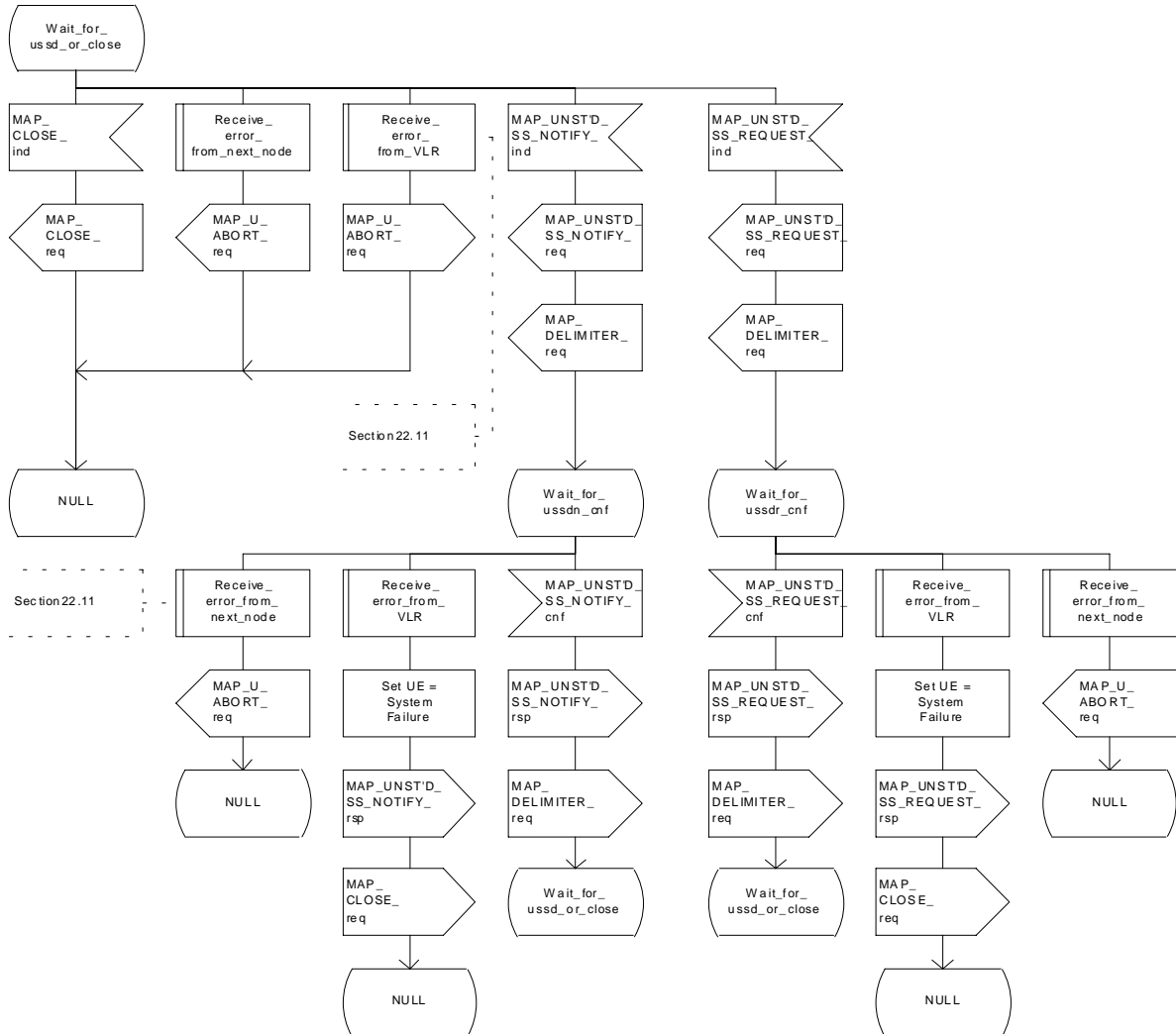


Figure 22.10.4/1 (sheet 3 of 5): Procedure NI_USSD_HLR

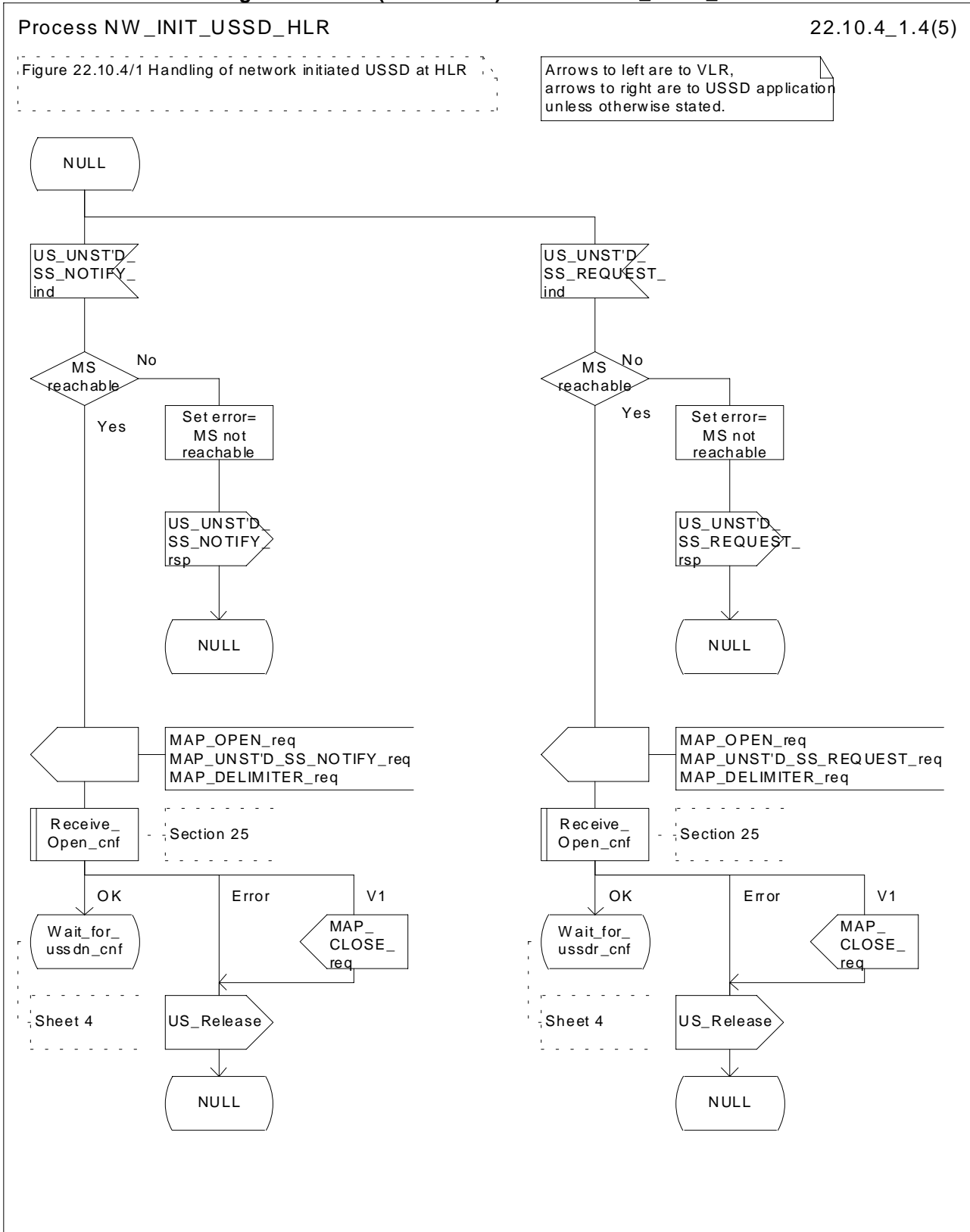


Figure 22.10.4/1 (sheet 4 of 5): Procedure NI_USSD_HLR

Process NW_INIT_USSD_HLR

22.10.4_1.5(5)

Figure 22.10.4/1 Handling of network initiated USSD at HLR

Arrows to left are to VLR, arrows to right are to USSD application unless otherwise stated.

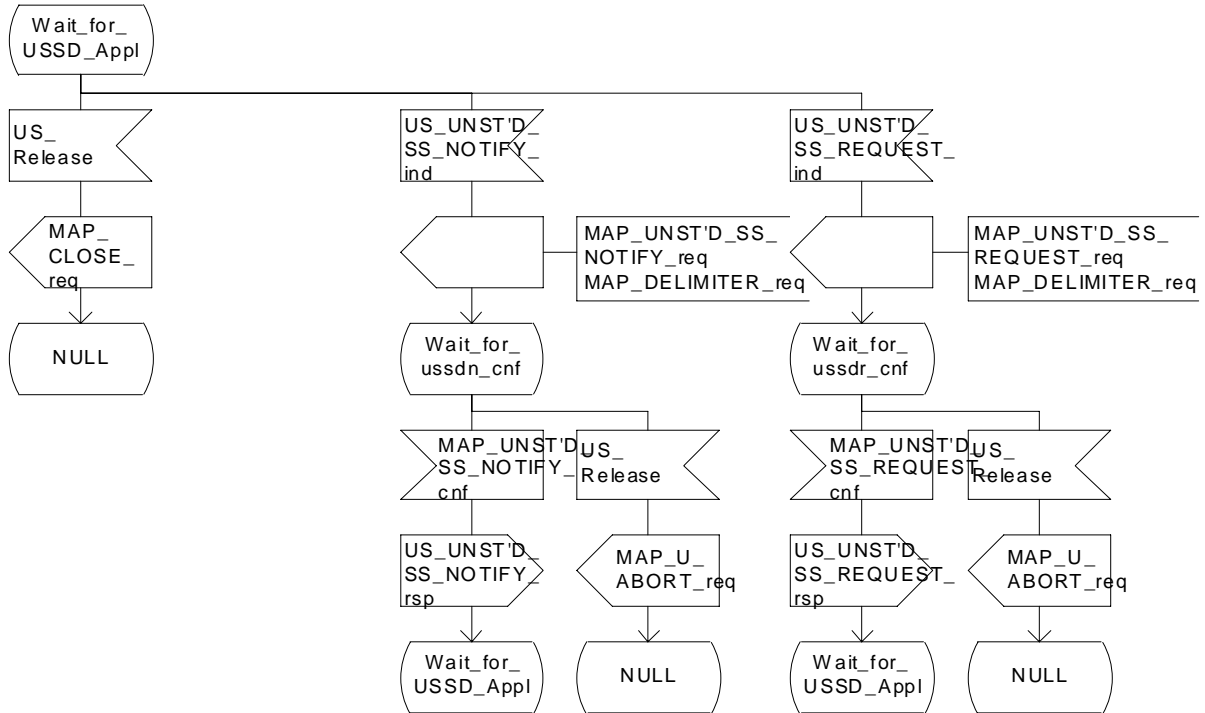


Figure 22.10.4/1 (sheet 5 of 5): Procedure NI_USSD_HLR

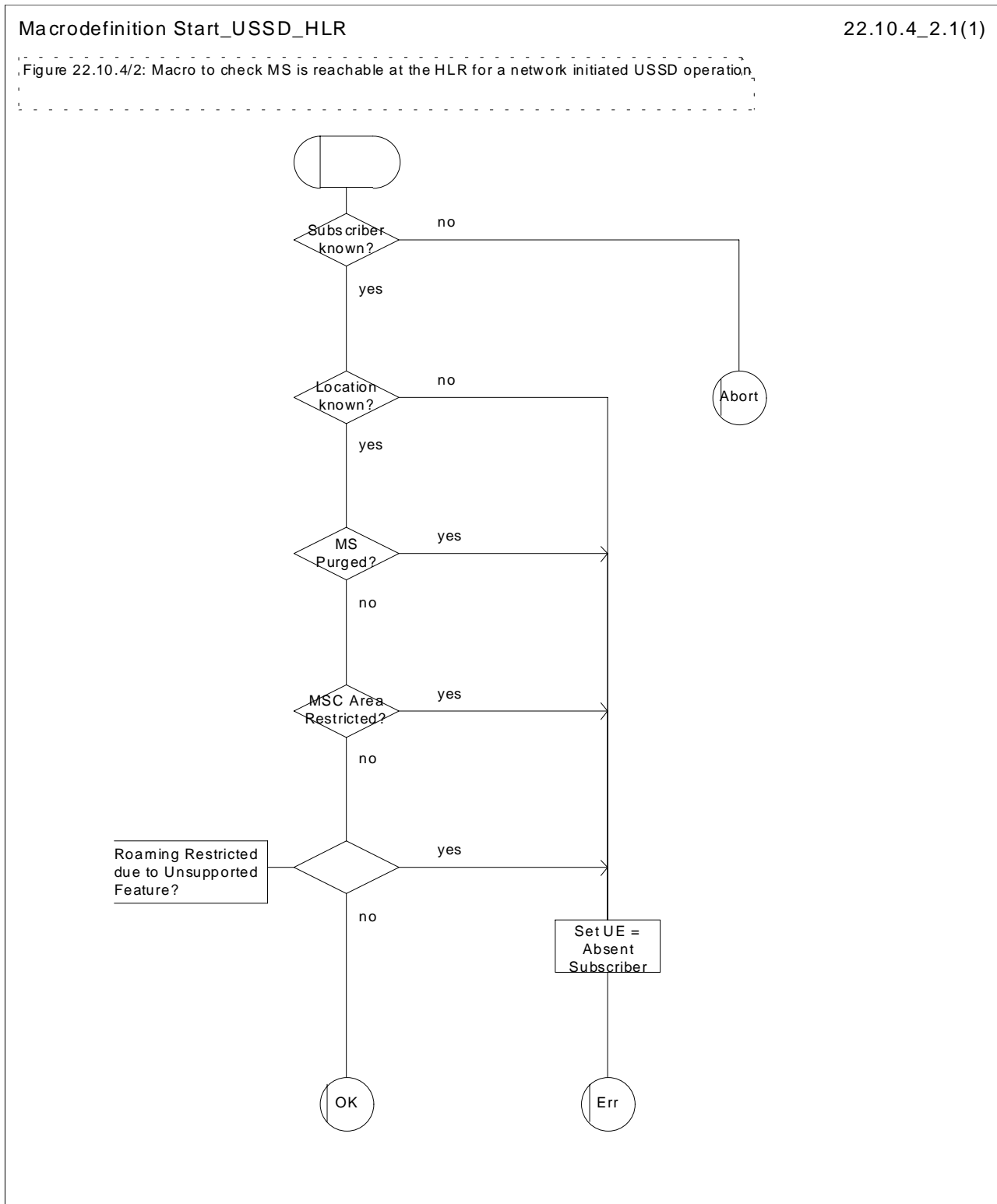


Figure 22.10.4/2: Macro Start_USSD_HLR

22.11 Common macros for clause 22

The following macros are used for the description of more than one of the supplementary service processes described in clause 22:

22.11.1 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 22.11.1/1.

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 22.11.1/2.

Macrodefinition GET_PASSWORD_MSC

22.11.1_1(1)

Figure 22.11.1/1: Macro which relays a GetPassword request from the VLR to the MS and relays the GetPassword response from the MS to the VLR

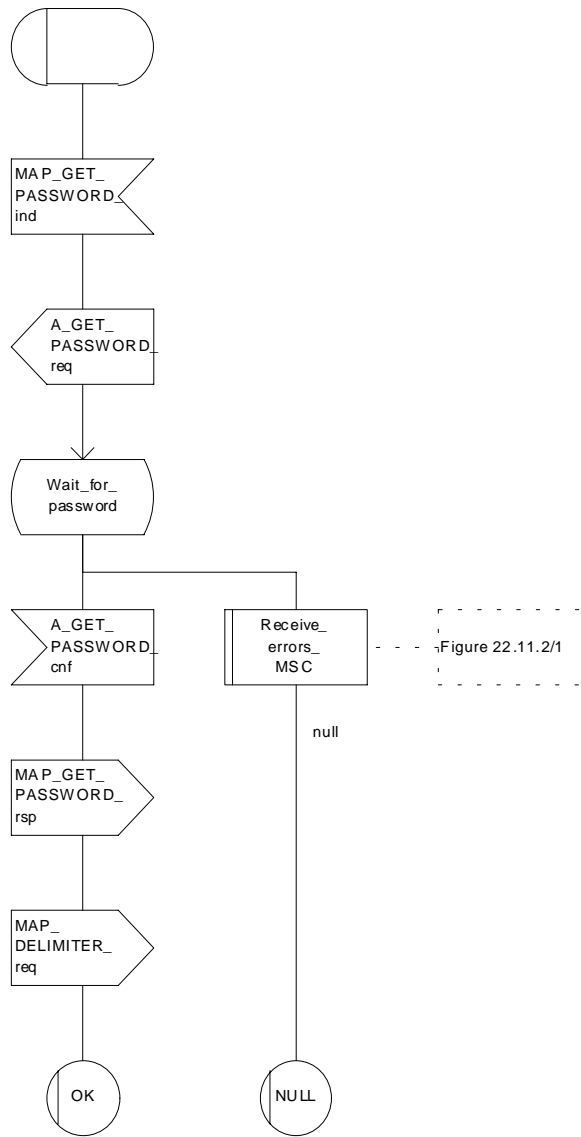


Figure 22.11.1/1: Macro Get_PW_MSC

Macrodefinition GET_PASSWORD_VLR

22.11.1_2(1)

Figure 22.11.1/2: Macro which relay a GetPassword request from the HLR to the VLR and relays the GetPassword response from the VLR to the HLR

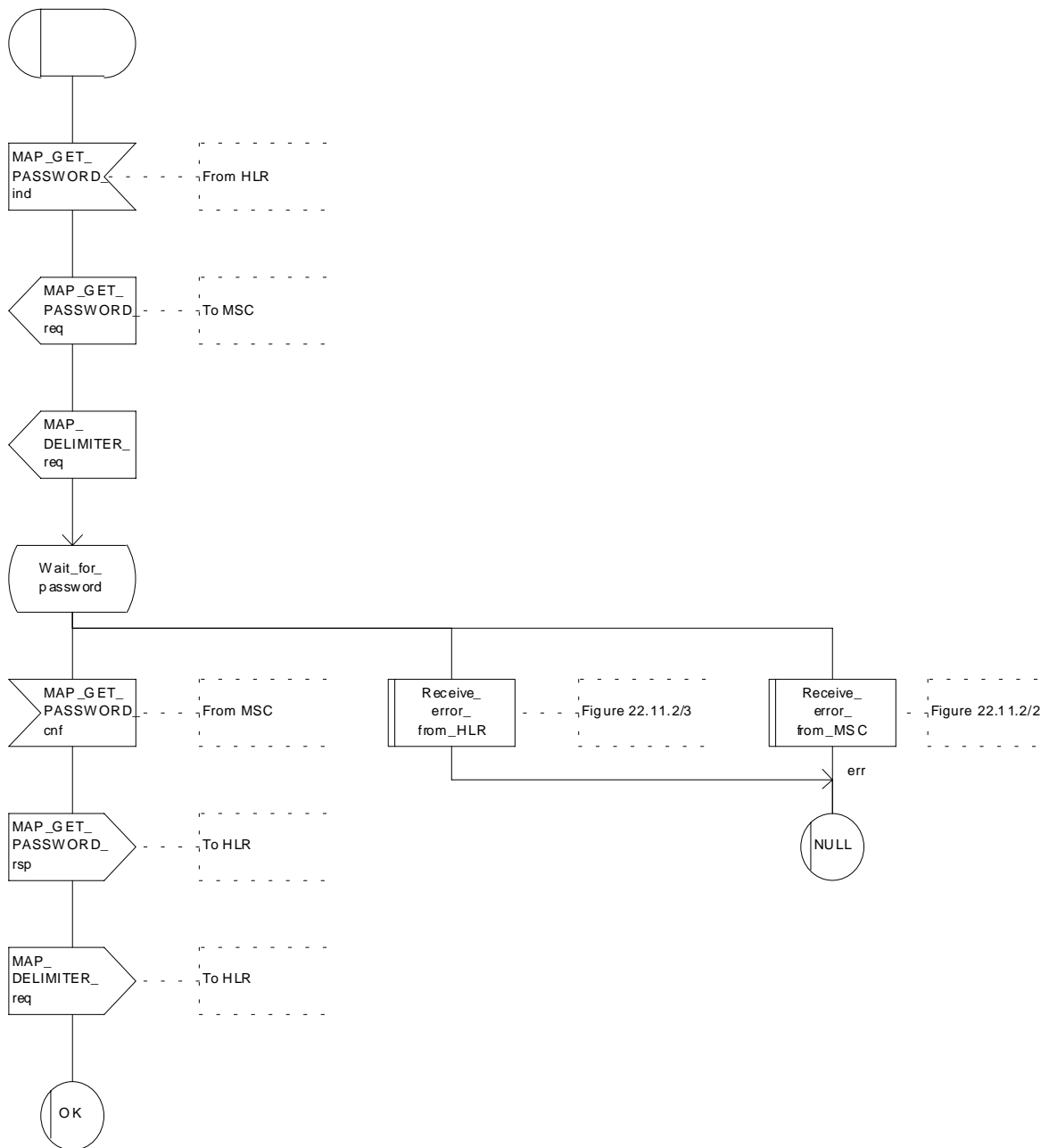


Figure 22.11.1/2: Macro Get_PW_VLR

22.11.2 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 22.11.2/1.

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 22.11.2/2.

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 HLR, that connection is closed. The macro is described in figure 22.11.2/3.

Macro Receive_error_from_VLR

This macro is used by the HLR to receive signals from the VLR that should lead to failure if received in any state of a supplementary service process. If a MAP_NOTICE indication is received from the VLR, 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 22.11.2/4.

Macro Receive_error_from_next_node

This macro is used by the primary HLR to receive signals from the gsmSCF or secondary HLR that should lead to failure if received in any state of a supplementary service process. If a MAP_NOTICE indication is received from the next node, that connection is closed. The macro is described in figure 22.11.2/5.

Macrodefinition Receive_errors_MSC

22.11.2_1(1)

Figure 22.11.2/1: Macro which handles possible error situations while the MSC is waiting for a confirmation of a supplementary service request to the VLR

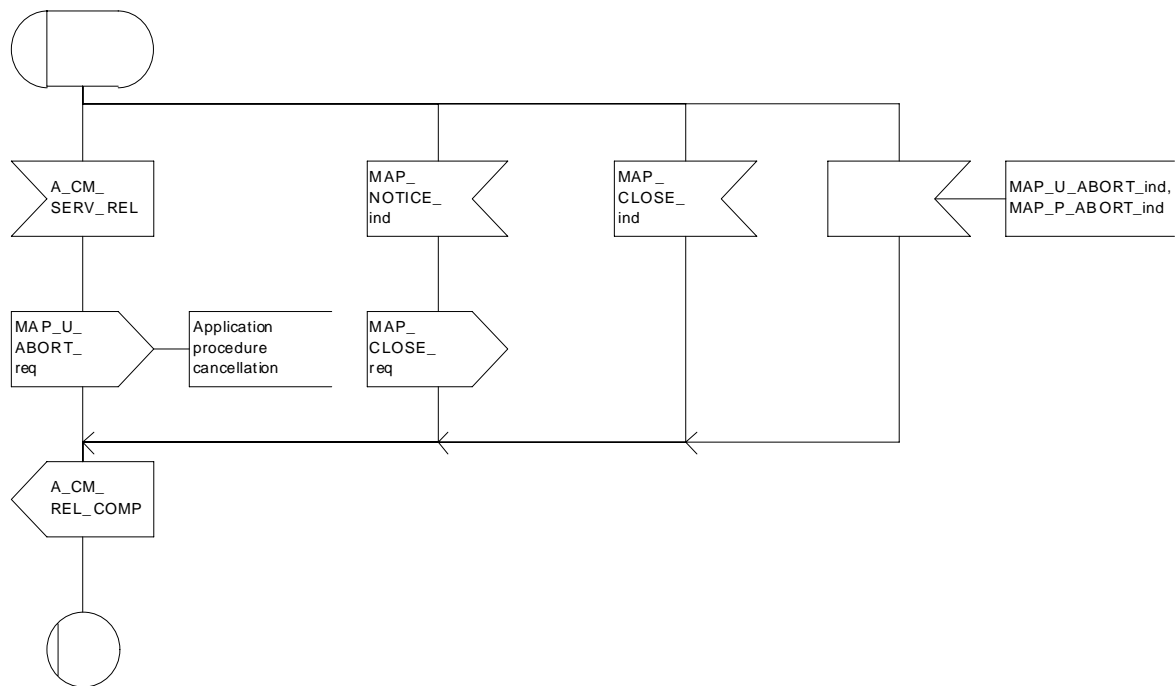


Figure 22.11.2/1: Macro Receive_Errors_MSC

Macrodefinition Receive_error_from_MSC

22.11.2_2(1)

Figure 22.11.2/2: Macro to receive errors from the MSC during supplementary services procedures in the VLR

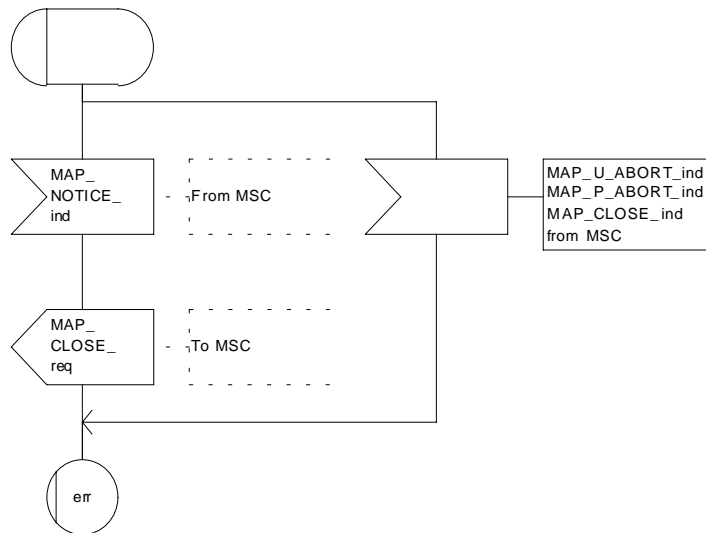


Figure 22.11.2/2: Macro Receive_Error_from_MSC

Macrodefinition Receive_error_from_HLR

22.11.2_3(1)

Figure 22.11.2/3: Macro to receive errors from the HLR while the VLR is waiting for a confirmation of a supplementary service request sent to the HLR

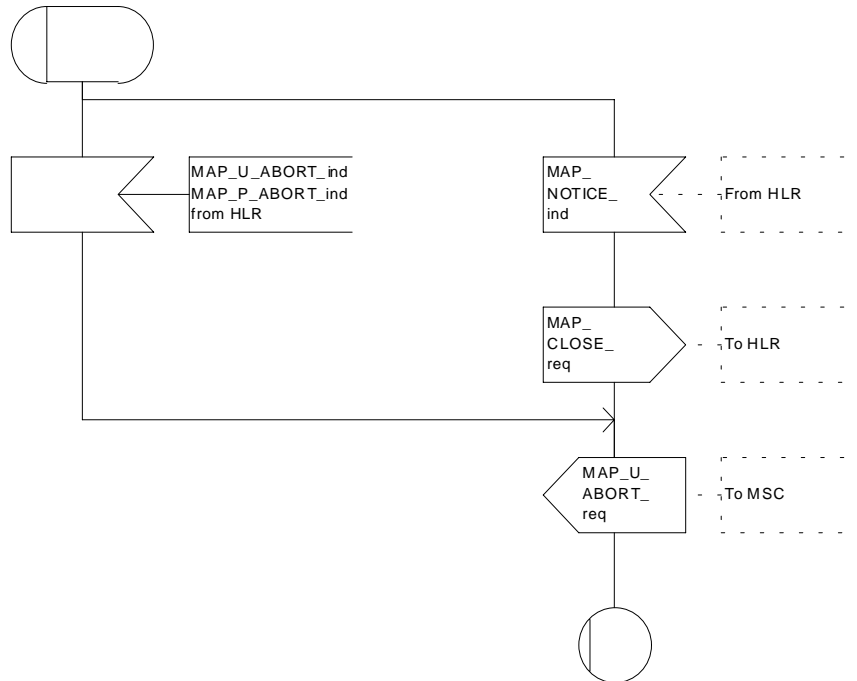


Figure 22.11.2/3: Macro Receive_Errors_HLR

Macrodefinition Receive_error_from_VLR

22.11.2_4(1)

Figure 22.11.2/4: Macro to receive errors from the VLR during supplementary services procedures in the HLR

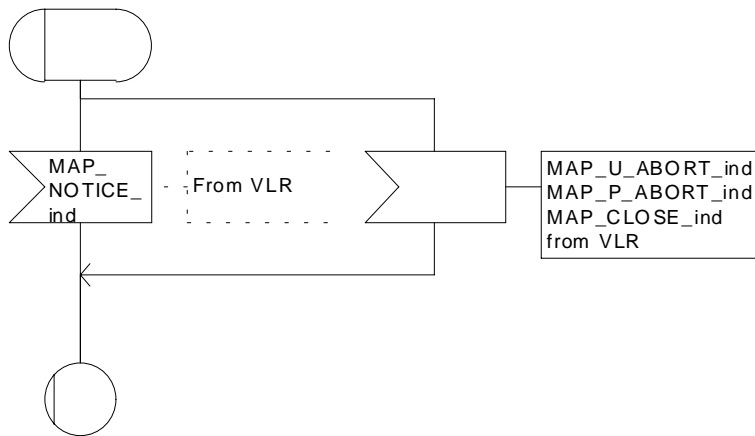


Figure 22.11.2/4: Macro Receive_error_from_VLR

Macrodefinition Receive_error_from_next_node

22.11.2_5(1)

Figure 22.11.2/5: Macro to receive errors from the next node while the HLR is waiting for a confirmation of a supplementary service request sent to the next node

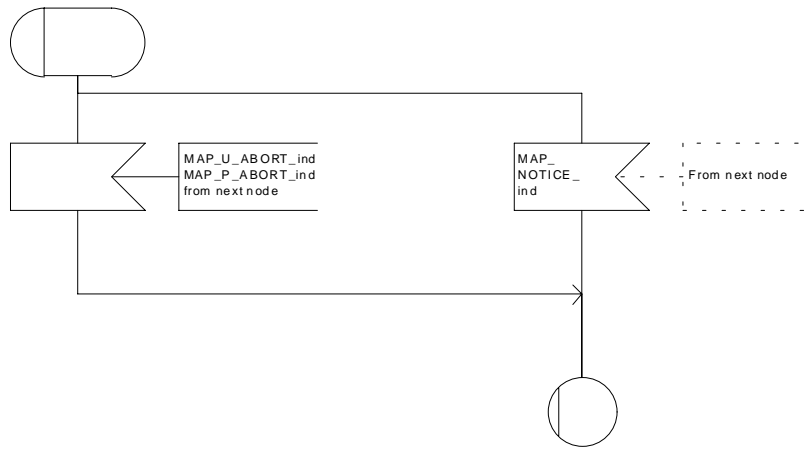


Figure 22.11.2/5: Macro Receive_error_from_next_node

22.12 Supplementary Service Invocation Notification procedure

22.12.1 General

The Supplementary Service Invocation Notification procedure is used to notify a gsmSCF about the invocation of a GSM Supplementary Service.

The password registration procedure is shown in figure 22.12.1/1.

The following services may be used:



(4) MAP-SUPPLEMENTARY-SERVICE-INVOCATION-NOTIFICATION (MSC to gsmSCF)

(5) MAP-SUPPLEMENTARY-SERVICE-INVOCATION-NOTIFICATION-ACK (gsmSCF to MSC)

MAP-SUPPLEMENTARY-SERVICE-INVOCATION-NOTIFICATION (defined in clauses 8 and 25);

Figure 22.12.1/1: Interfaces and services for supplementary service invocation notification

22.12.2 Procedures in the MSC

The supplementary service invocation notification procedure in the MSC is triggered when the requested supplementary service is invoked at the MSC. The MSC notifies the gsmSCF of a supplementary service invocation the MAP-SUPPLEMENTARY-SERVICE-INVOCATION-NOTIFICATION service. This is sent in a TCAP TC-BEGIN primitive. The MSC then awaits a positive or negative acknowledgement from the gsmSCF to the MAP-SUPPLEMENTARY-SERVICE-INVOCATION-NOTIFICATION. This is received in a TCAP TC-END primitive, and upon receipt the relationship between the MSC and the gsmSCF is terminated. Similarly, the relationship is terminated at the MSC by the sending from or receipt of a TCAP P-ABORT primitive. This is illustrated in Figure 22.12.2.

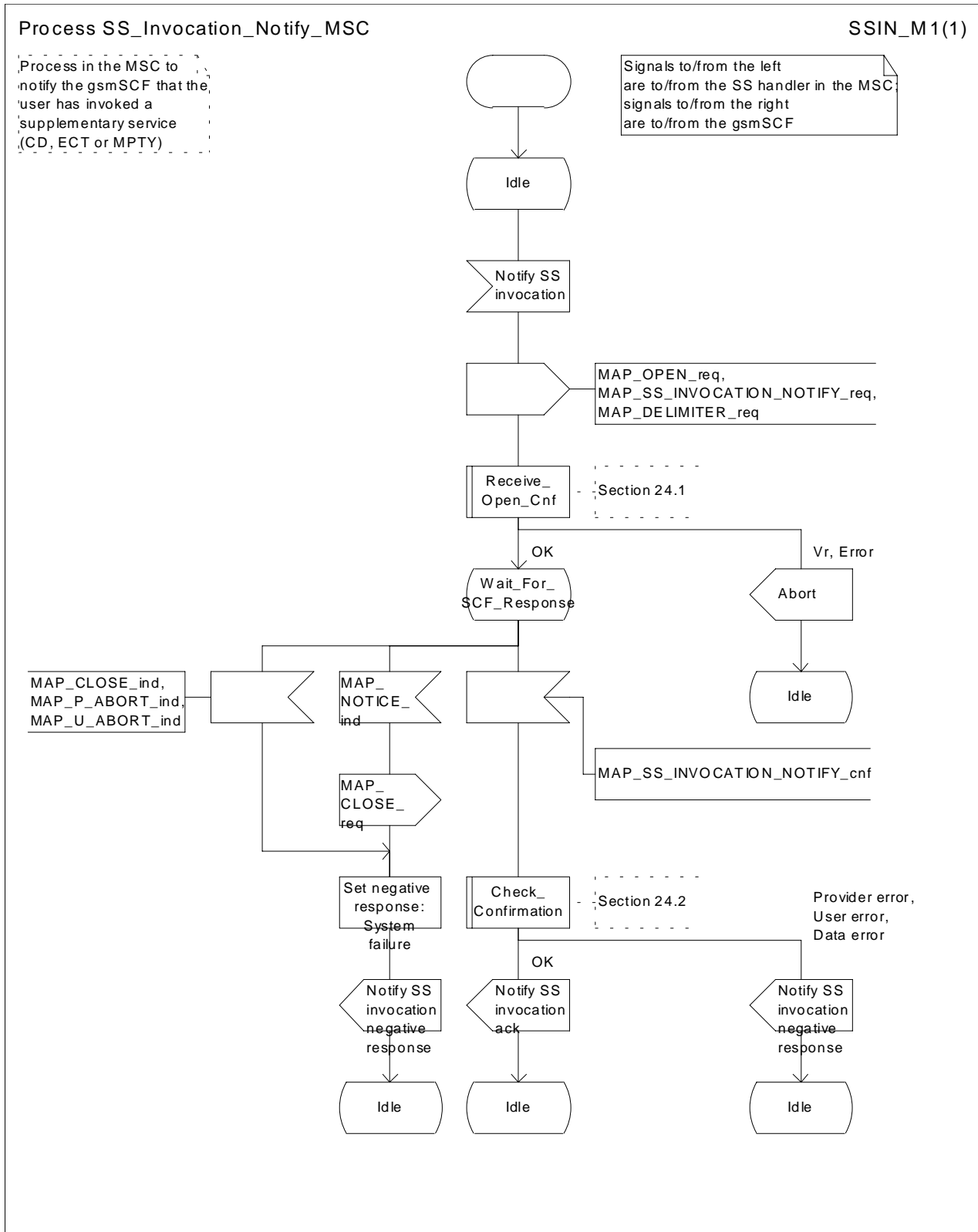


Figure 22.12.2 Process SS_Invocation_Notify_MSC (sheet 1 of 1)

22.12.3 Procedures in the gsmSCF

Upon receiving notification of the supplementary service invocation via the MAP-SUPPLEMENTARY-SERVICE-INVOCATION-NOTIFICATION service, the gsmSCF analyses the received information. If the gsmSCF understands the information sent via the the MAP-SUPPLEMENTARY-SERVICE-INVOCATION-NOTIFICATION service then it returns a positive acknowledgement to the MAP-SUPPLEMENTARY-SERVICE-INVOCATION-NOTIFICATION, indicating the success of the service. This is returned in a TCAP TC-END primitive, using the basic end procedure.

Otherwise, a negative acknowledgement to the MAP-SUPPLEMENTARY-SERVICE-INVOCATION-NOTIFICATION is returned. This is also returned in a TCAP TC-END primitive, again using the basic end procedure. The gsmSCF TCAP service may also choose to abort the relationship to the MSC by sending a TCAP P-ABORT primitive. It will immediately terminate processing of a MAP-SUPPLEMENTARY-SERVICE-INVOCATION-NOTIFICATION should a TCAP P-ABORT primitive be received from the MSC. This is illustrated in Figure 22.12.3.

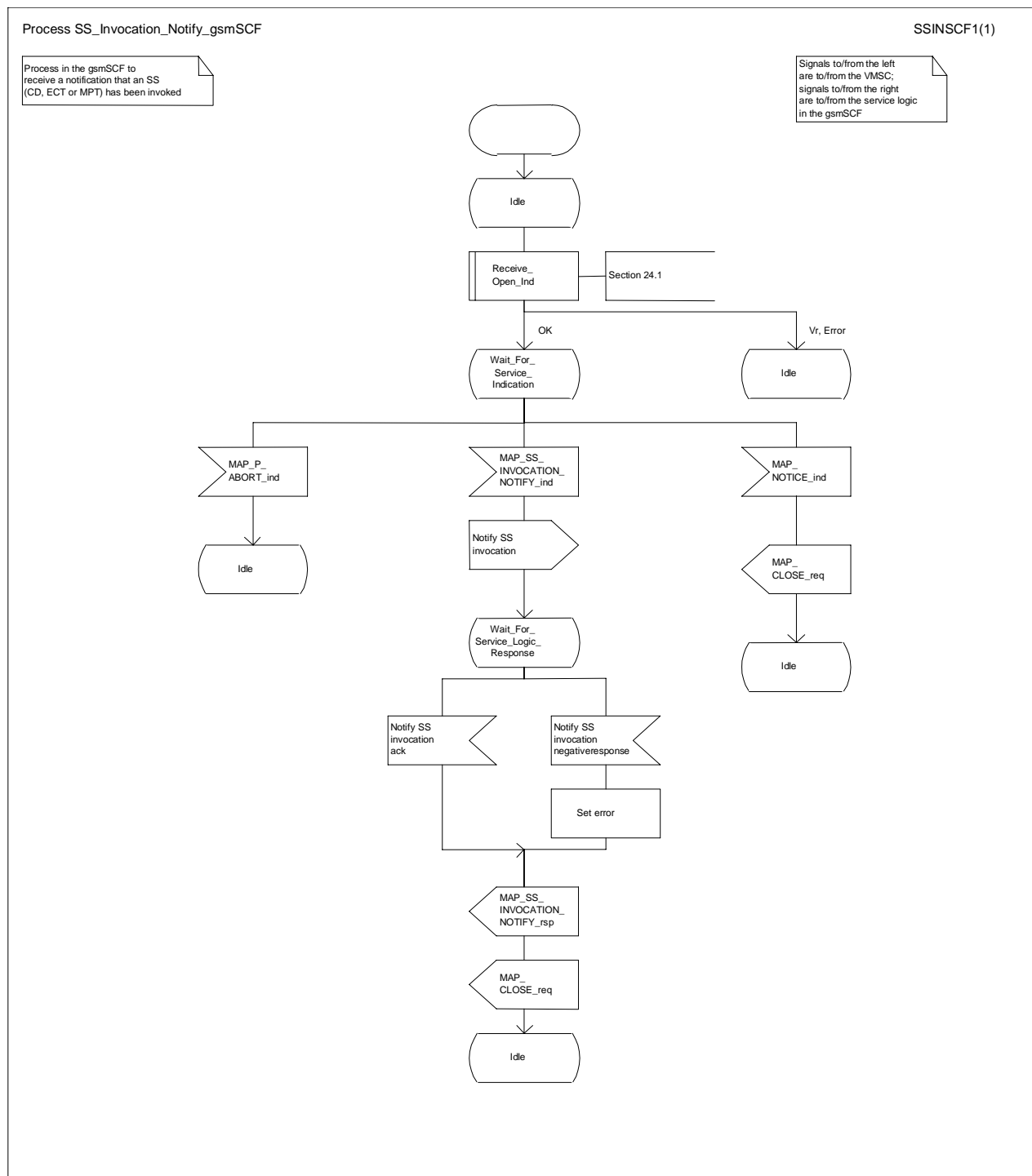


Figure 22.12.3 Process SS_Invocation_Notify_gsmSCF (sheet 1 of 1)

22.13 Activation of a CCBS request

22.13.1 General

The message flow to activate a CCBS request is shown in figure 22.13.1/1.

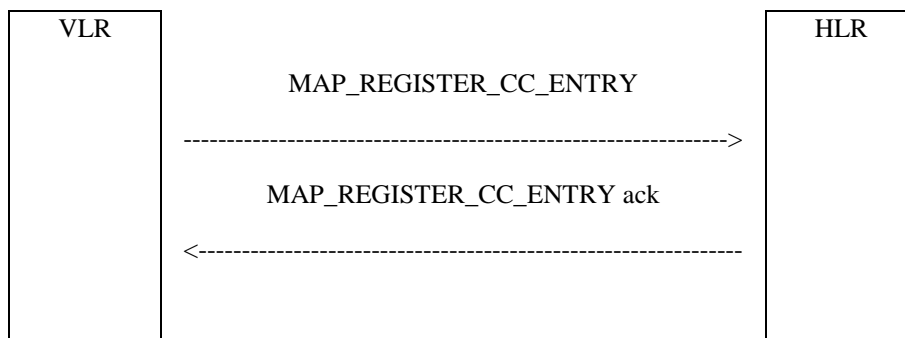


Figure 22.13.1/1: Message flow to activate a CCBS request

22.13.2 Procedure in the VLR

The MAP process in the VLR to activate a CCBS request is shown in figure 22.13.2/1. The MAP process invokes macros not defined in this subclause; the definitions of these macros can be found as follows:

Receive_Open_Cnf	see subclause 25.1.2;
Check_Confirmation	see subclause 25.2.2;

Successful Outcome

When the MAP process receives a CCBS Request message from the CCBS application process in the VLR, it requests a dialogue with the HLR whose identity is contained in the request by sending a MAP_OPEN service request and the necessary information in a MAP_REGISTER_CC_ENTRY service request. The VLR then invokes the macro Receive_Open_Cnf to wait for the response to the dialogue opening request. If the dialogue opening is successful, the MAP process waits for a response from the HLR.

If the MAP process receives a MAP_REGISTER_CC_ENTRY service confirm from the HLR, the MAP process invokes the macro Check_Confirmation to check the content of the confirm.

If the macro Check_Confirmation takes the OK exit, the MAP process sends a CCBS Request Ack message containing the information received from the HLR to the CCBS application process in the VLR and returns to the idle state.

Failure of dialogue opening with the HLR

If the macro Receive_Open_Cnf takes the Vr exit or the Error exit, the MAP process sends a CCBS Request Negative response message to the CCBS application process in the VLR and returns to the idle state.

Error in MAP_REGISTER_CC_ENTRY confirm

If the MAP_REGISTER_CC_ENTRY service confirm contains a user error or a provider error, or the macro Check_Confirmation indicates that there is a data error, the MAP process sends a CCBS Request Negative response message to the CCBS application process in the VLR and returns to the idle state.

Abort of HLR dialogue

After the dialogue with the HLR has been established, the MAP service provider may abort the dialogue by issuing a MAP_P_ABORT indication. In this case, the MAP process sends a CCBS Request negative response to the CCBS application process in the VLR and returns to the idle state.

If the MAP provider indicates a protocol problem by sending a MAP_NOTICE indication, the MAP process closes the dialogue with the HLR, sends a CCBS Request negative response indicating system failure to the CCBS application process in the VLR and returns to the idle state.

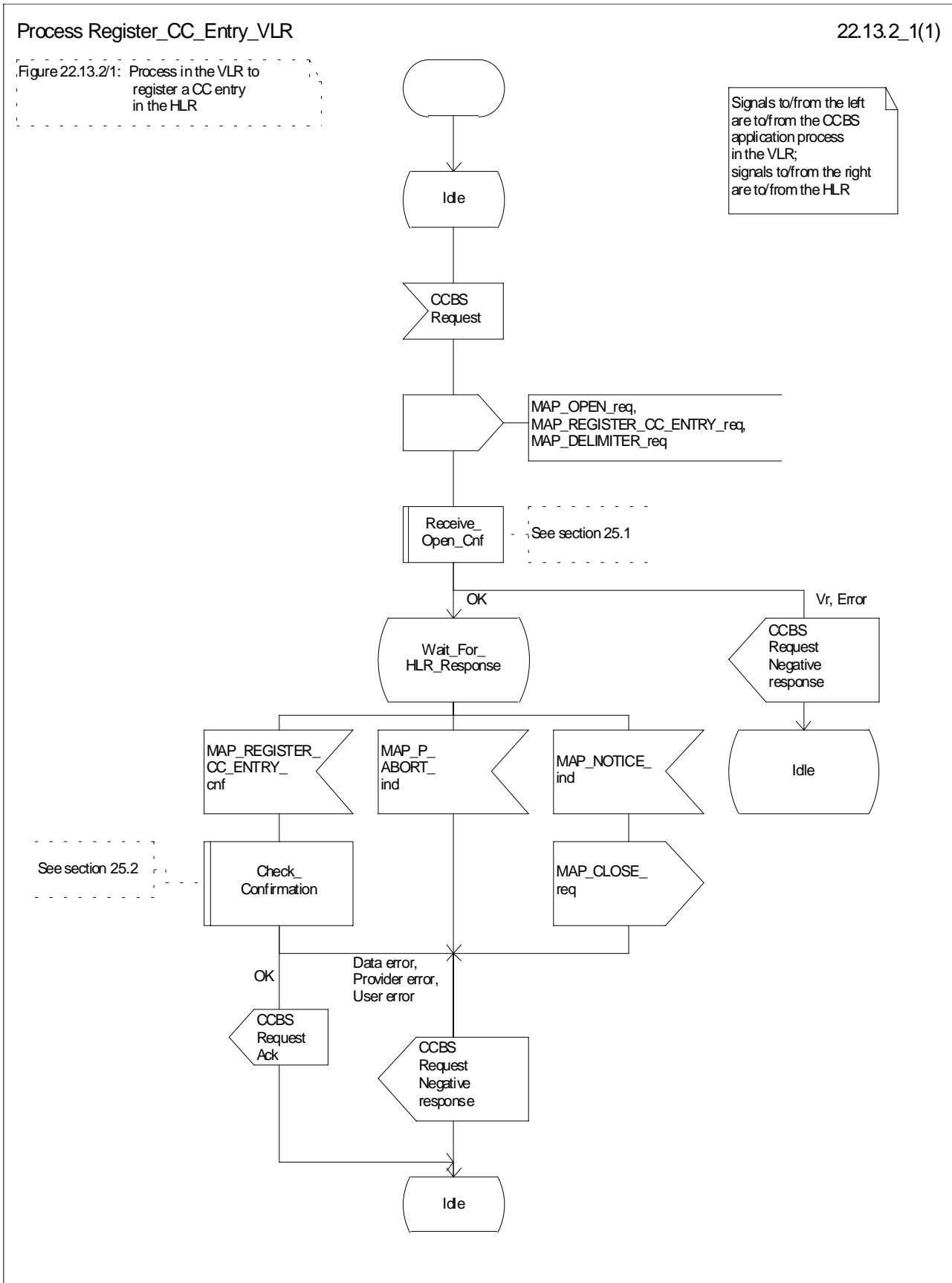


Figure 22.13.2/1: Process Register_CC_Entry_VLR

22.13.3 Procedure in the HLR

Successful outcome

When the MAP process receives a MAP_REGISTER_CC_ENTRY_indication from the co-ordinating process, it sends a CCBS Request message to the CCBS application process in the HLR, and waits for a response. The request contains the parameters received in the MAP_REGISTER_CC_ENTRY service indication.

If the CCBS application process in the HLR returns a positive response, the MAP process constructs a MAP_REGISTER_CC_ENTRY service response, constructs a MAP_CLOSE service request, sends them to the co-ordinating process and terminates.

Negative response from HLR CCBS application process

If the CCBS application process in the HLR returns a negative response, the MAP process constructs a MAP_REGISTER_CC_ENTRY service response containing the appropriate error, constructs a MAP_CLOSE service request, sends them to the co-ordinating process and terminates.

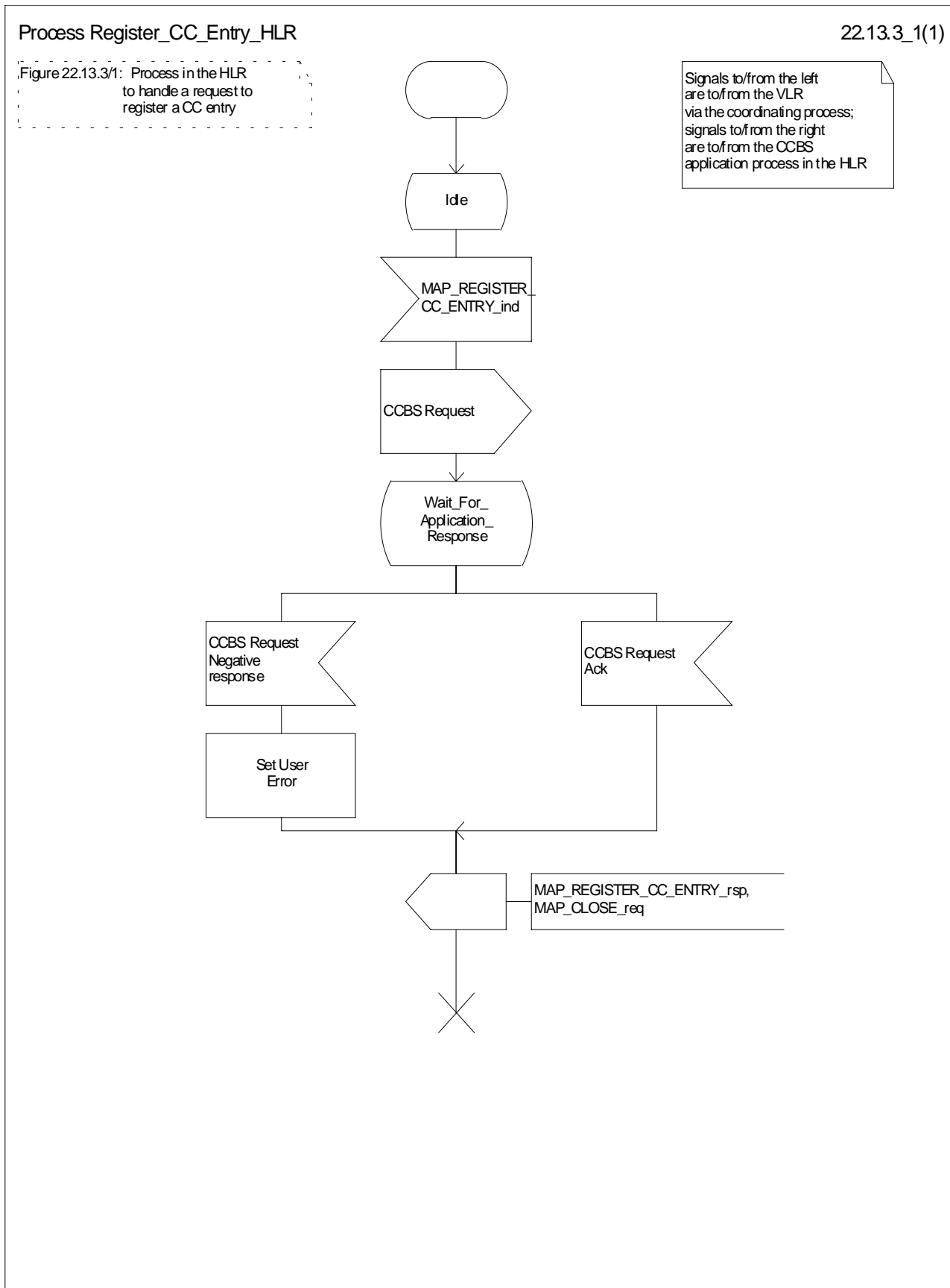


Figure 22.13.3/1: Process Register_CC_Entry_HLR

22.14 Deactivation of a CCBS request

22.14.1 General

The message flow to deactivate a CCBS request is shown in figure 22.14.1/1.

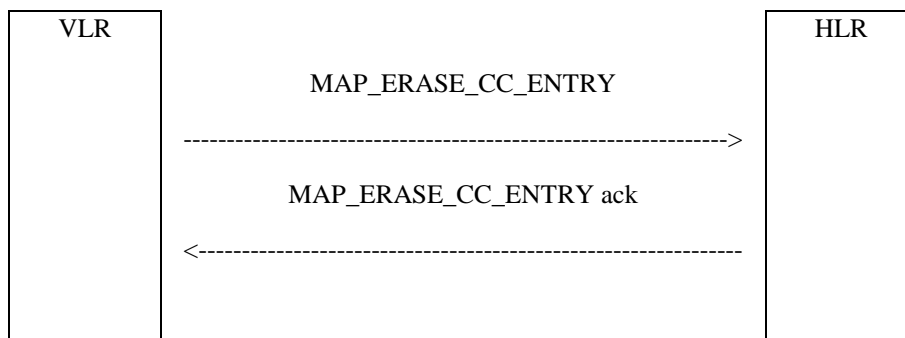


Figure 22.14.1/1: Message flow to deactivate a CCBS request

22.14.2 Procedure in the VLR

The MAP process in the VLR to deactivate a CCBS request is shown in figure 22.14.2/1. The MAP process invokes macros not defined in this subclause; the definitions of these macros can be found as follows:

Receive_Open_Cnf	see subclause 25.1.2;
Check_Confirmation	see subclause 25.2.2;

Successful Outcome

When the MAP process receives a Deactivate CCBS message from the CCBS application process in the VLR, it requests a dialogue with the HLR whose identity is contained in the request by sending a MAP_OPEN service request and the necessary information in a MAP_ERASE_CC_ENTRY service request. The VLR then invokes the macro Receive_Open_Cnf to wait for the response to the dialogue opening request. If the dialogue opening is successful, the MAP process waits for a response from the HLR.

If the MAP process receives a MAP_ERASE_CC_ENTRY service confirm from the HLR, the MAP process invokes the macro Check_Confirmation to check the content of the confirm.

If the macro Check_Confirmation takes the OK exit, the MAP process sends a Deactivate CCBS Ack message containing the information received from the HLR to the CCBS application process in the VLR and returns to the idle state.

Failure of dialogue opening with the HLR

If the macro Receive_Open_Cnf takes the Vr exit or the Error exit, the MAP process sends a Deactivate CCBS Negative response message to the CCBS application process in the VLR and returns to the idle state.

Error in MAP_ERASE_CC_ENTRY confirm

If the MAP_ERASE_CC_ENTRY service confirm contains a user error or a provider error, or the macro Check_Confirmation indicates that there is a data error, the MAP process sends a Deactivate CCBS Negative response message to the CCBS application process in the VLR and returns to the idle state.

Abort of HLR dialogue

After the dialogue with the HLR has been established, the MAP service provider may abort the dialogue by issuing a MAP_P_ABORT indication. In this case, the MAP process sends a Deactivate CCBS negative response to the CCBS application process in the VLR and returns to the idle state.

If the MAP provider indicates a protocol problem by sending a MAP_NOTICE indication, the MAP process closes the dialogue with the HLR, sends a Deactivate CCBS negative response indicating system failure to the CCBS application process in the VLR and returns to the idle state.

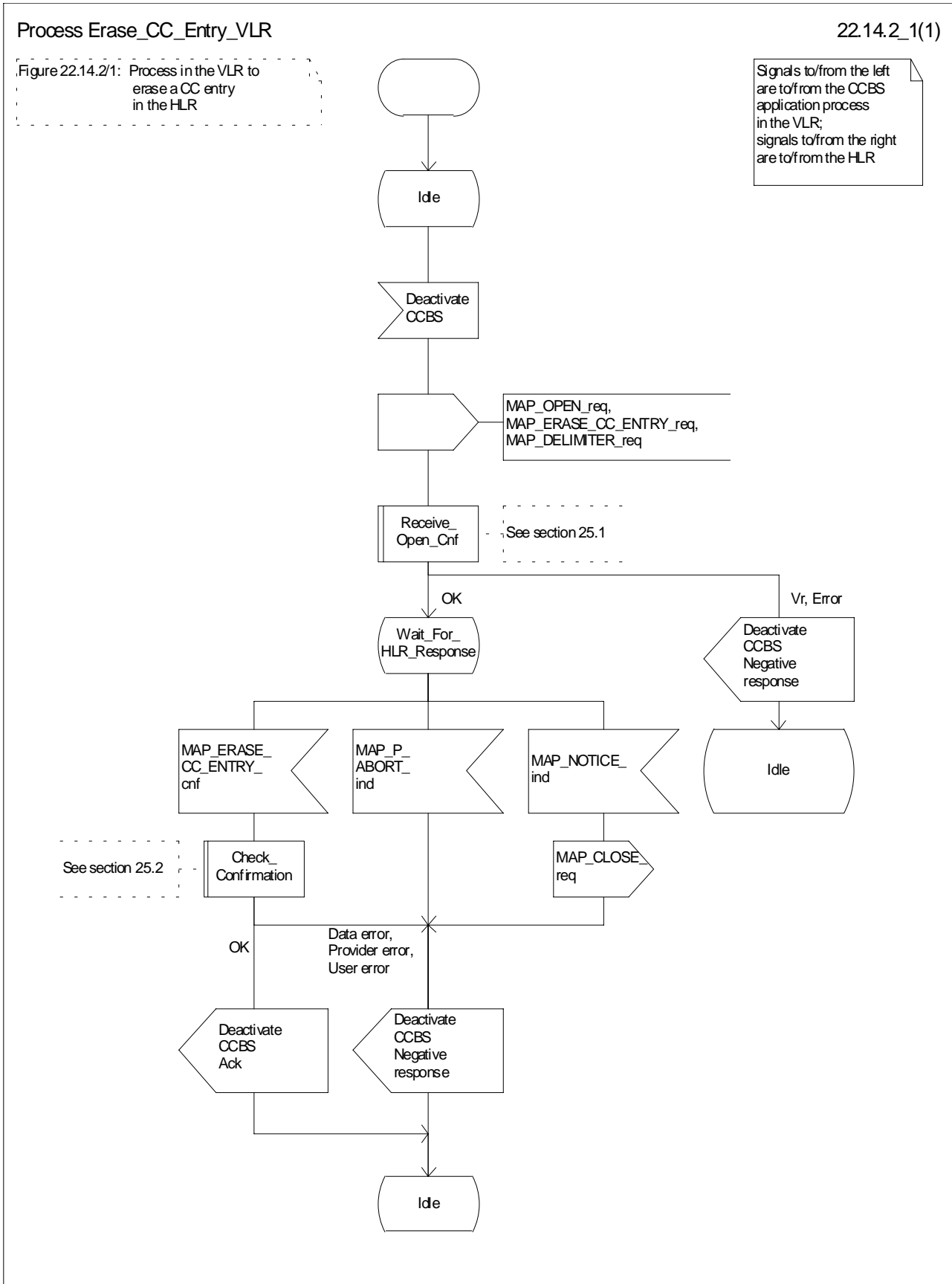


Figure 22.14.2/1: Process Erase_CC_Entry_VLR

22.14.3 Procedure in the HLR

Successful outcome

When the MAP process receives a MAP_ERASE_CC_ENTRY_indication from the co-ordinating process, it sends a Deactivate CCBS message to the CCBS application process in the HLR, and waits for a response. The message contains the parameters received in the MAP_ERASE_CC_ENTRY service indication.

If the CCBS application process in the HLR returns a positive response, the MAP process constructs a MAP_ERASE_CC_ENTRY service response, constructs a MAP_CLOSE service request, sends them to the co-ordinating process and terminates.

Negative response from HLR CCBS application process

If the CCBS application process in the HLR returns a negative response, the MAP process constructs a MAP_ERASE_CC_ENTRY service response containing the appropriate error, constructs a MAP_CLOSE service request, sends them to the co-ordinating process and terminates.

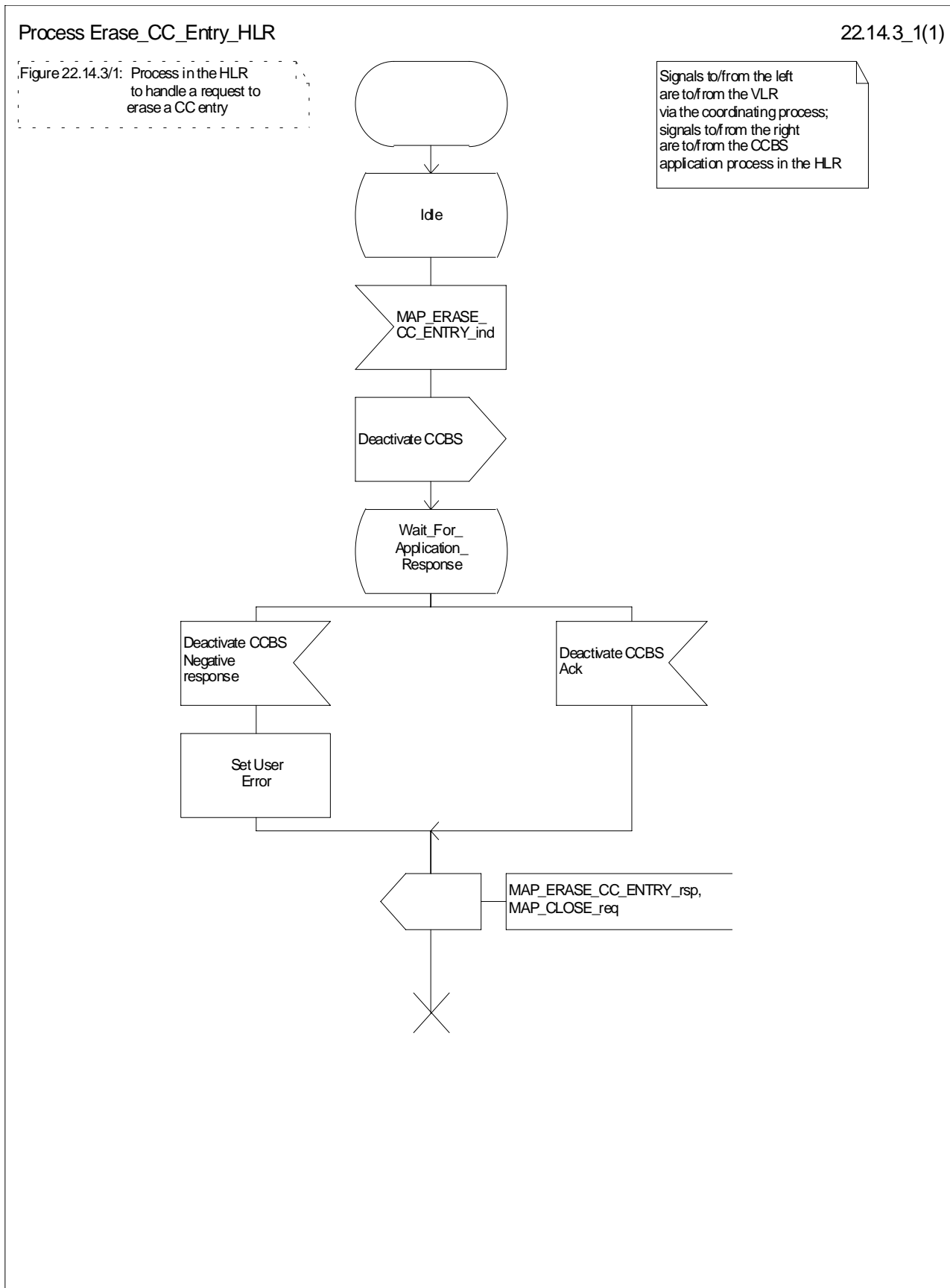


Figure 22.14.3/1: Process Erase_CC_Entry_HLR

23 Short message service procedures

23.1 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.

NOTE: A_CM_SERV_REQ message is not used for SMS over GPRS.

23.1.1 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 23.2.1);
- if the A_RP_SM_MEMORY_AVAILABLE indication is received, the process SC_Alert_MSC is initiated (see subclause 23.4.1).

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 23.1/1.

Process Co_SMS_MSC

23.1_1(1)

Figure 23.1/1: The SMS co-ordinating process in the MSC.

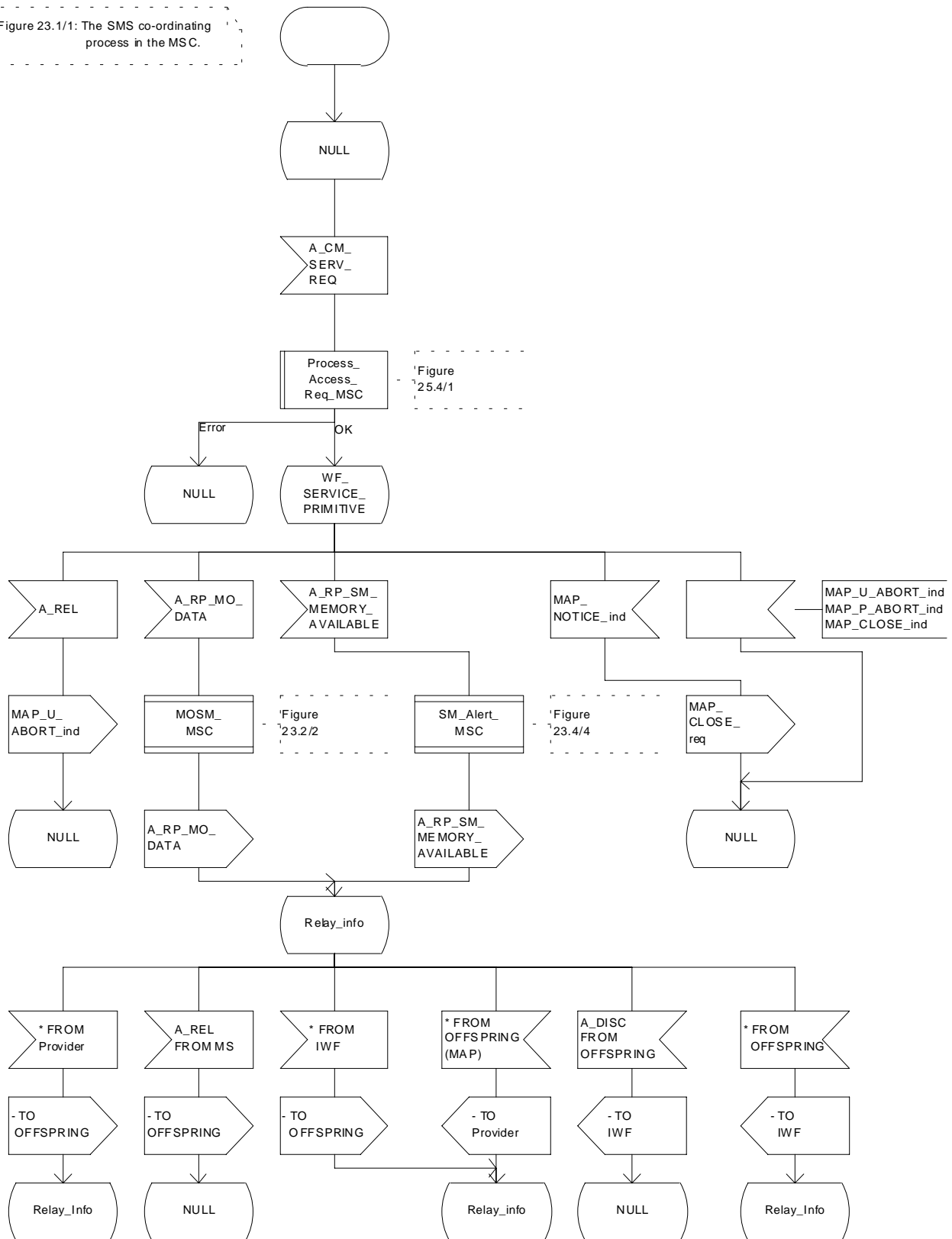


Figure 23.1/1: Process Co_SMS_MSC

23.1.2 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 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 SM Gateway Co-ordinator is shown in the figure 23.1/2.

If the Receive_Open_Ind macro takes the Vr exit then HLR shall perform the MAP Vr dialogue. But based on the subscriber data, handling at the MAP user application level may be performed as described in release 97 :

- If the subscriber is not a GPRS subscriber then the behaviour of the HLR shall be the same as described in the corresponding MAP Vr release.
- If the subscriber is a GPRS subscriber and a non-GPRS subscriber with the option « transfer of SM via the MSC when GPRS is not supported in the GMSC » then the behaviour of the HLR shall be the same as described in the corresponding MAP Vr release.
- If the subscriber is a GPRS subscriber and a non-GPRS subscriber with the option « transfer of SM via the SGSN when GPRS is not supported in the GMSC » or if the subscriber is a GPRS subscriber only then the behaviour of the HLR shall be the same as for the case transfer over GPRS described in MAP release 97, with the following precision : because GMSC does not support MAP release 97, the previous MAP protocol release is used. When the HLR sends the MAP_SEND_ROUTING_INFO_FOR_SM_Resp, the SGSN number is mapped to the MAP parameter « MSC number ». When the HLR sends the MAP_INFORM_SERVICE_CENTRE_resp, the MNRG status shall be mapped to the MAP parameter « mnrf-set ». When the HLR receives the MAP_REPORT_SM_DELIVERY_STATUS_Ind, it shall interpret the delivery outcome as a GPRS delivery outcome.

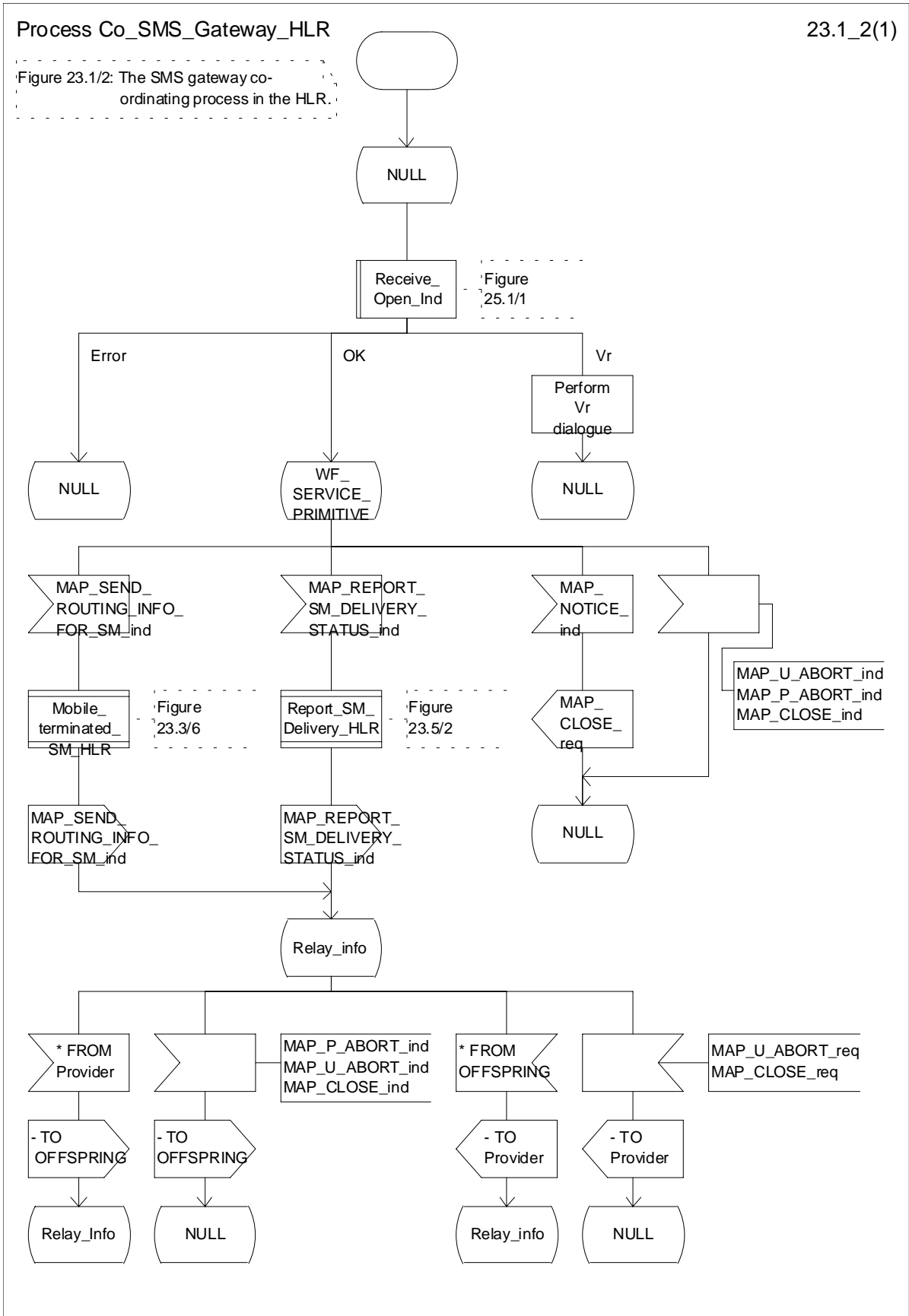


Figure 23.1/2: Process Co_SMS_Gateway_HLR

23.1.3 Mobile originated short message service Co-ordinator for the SGSN

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_SGSN is initiated (see subclause 23.2.4);
- if the A_RP_SM_MEMORY_AVAILABLE indication is received, the process SC_Alert_SGSN is initiated (see subclause 23.4.5).

After creation of the user process the Co-ordinator relays the messages between the SGSN and the MS, 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 23.1/3.

Process Co_SMS_SGSN

23.1_3(1)

Figure 23.1/3: The SMS co-ordinating process in the SGSN

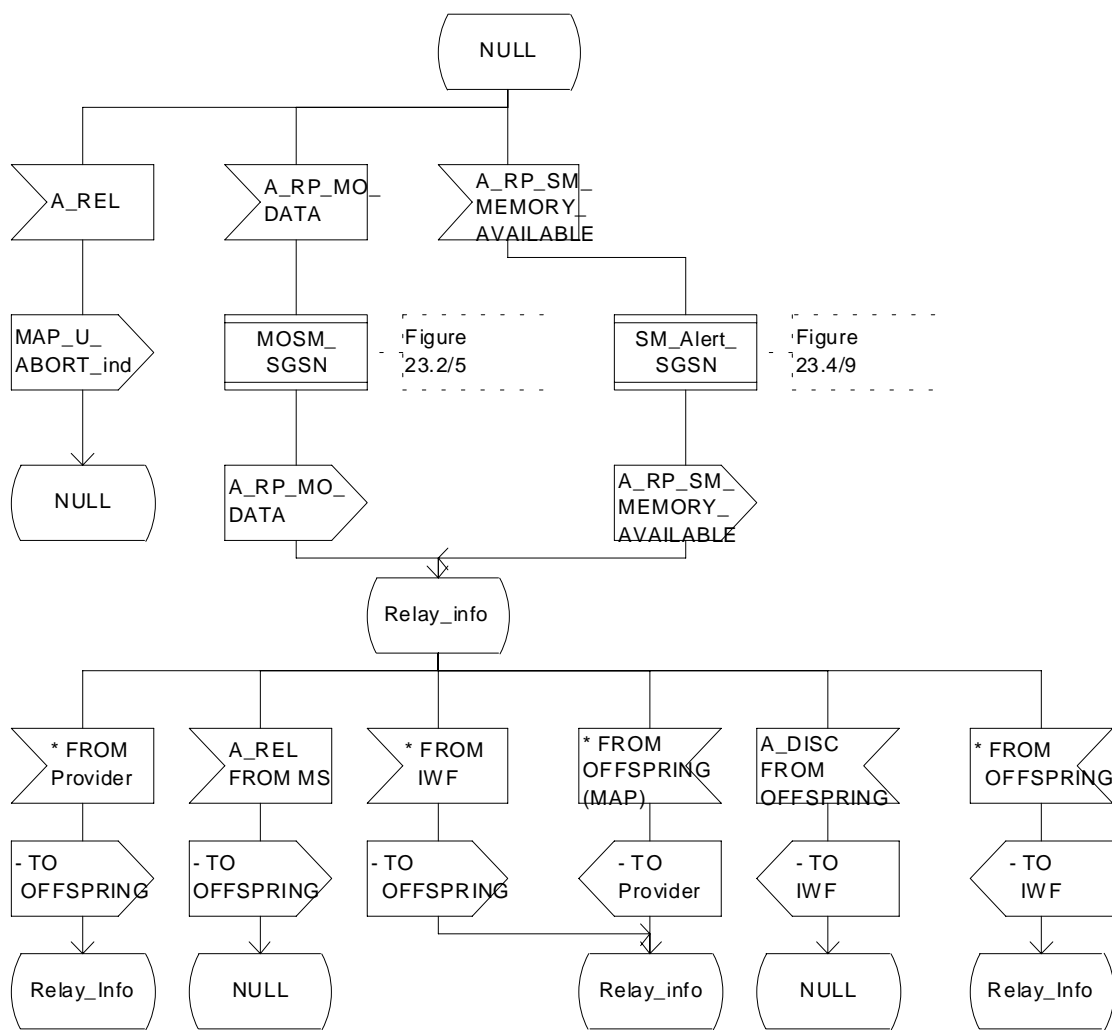
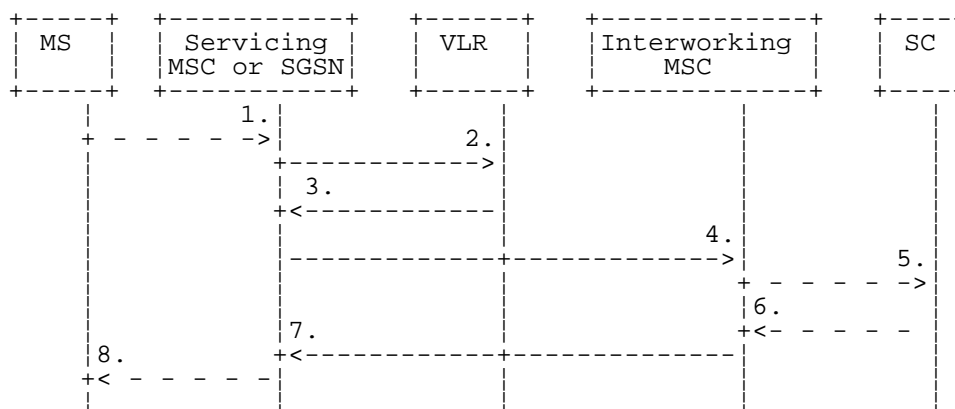


Figure 23.1/3: Process Co_SMS_SGSN

23.2 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 23.2/1.



- 1) Short Message (GSM 04.11)
- 2) MAP_SEND_INFO_FOR_MO_SMS (*)
- 3) MAP_SEND_INFO_FOR_MO_SMS_ACK (*)
- 4) MAP_MO_FORWARD_SHORT_MESSAGE
- 5) Short message (TS GSM 03.40)
- 6) Short message Acknowledgement (TS GSM 03.40)
- 7) MAP_MO_FORWARD_SHORT_MESSAGE_ACK
- 8) Short Message Acknowledgment (GSM 04.11)

(*) Messages 2) and 3) are not used by SGSN

Figure 23.2/1: Mobile originated short message transfer

In addition the following MAP services are used:

- | | |
|-------------------------------|--------------------------|
| MAP_PROCESS_ACCESS_REQUEST | (see subclause 8.3); (*) |
| MAP_AUTHENTICATE | (see subclause 8.5); (*) |
| MAP_SET_CIPHERING_MODE | (see subclause 8.6); (*) |
| MAP_PROVIDE_IMSI | (see subclause 8.9); (*) |
| MAP_CHECK_IMEI | (see subclause 8.7); |
| MAP_FORWARD_NEW_TMSI | (see subclause 8.9); (*) |
| MAP_TRACE_SUBSCRIBER_ACTIVITY | (see subclause 9.1); (*) |
| MAP_READY_FOR_SM | (see subclause 12.4). |

(*) Those messages are not used by SGSN.

23.2.1 Procedure in the servicing MSC

The activation of the MAP_PROCESS_ACCESS_REQUEST service is described in the subclause 25.4.1.

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_MO_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 TS GSM 03.40.

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 mobile station.

The MSC opens a CAMEL dialogue as specified in 3G TS 23.078. If the CAMEL service bars the MO SM then the failure is reported to MS.

The MSC checks the barring as follows;

- if the short message transfer would contravene operator determined barring, the failure is reported to the CAMEL service as specified in 3G TS 23.078 and the call barred error with cause operator barring is returned to MS;
- if the short message transfer would contravene the supplementary service call barring conditions, the failure is reported to the CAMEL service as specified in 3G TS 23.078 and the call barred error with cause barring service active is returned to MS.

If the service MAP_MO_FORWARD_SHORT_MESSAGE is started, the MSC will check whether the grouping of MAP_OPEN request and MAP_MO_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_MO_FORWARD_SHORT_MESSAGE request is sent. As a response to the procedure, the servicing MSC will receive the MAP_MO_FORWARD_SHORT_MESSAGE confirmation from the IWMSC indicating that:

- the short message has been successfully delivered to the Service Centre. The successful submission of SM is reported to the CAMEL service as specified in 3G TS 23.078 and the acknowledgement is sent to the mobile station;
- one of several error cases has occurred. The mapping between MAP error causes and RP_ERROR causes is described in TS GSM 03.40. The failure in the SM submission is reported to the CAMEL service as specified in 3G TS 23.078 and the appropriate indication is provided to the mobile station.

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 mobile station.

If the MSC itself is the interworking MSC, the short message is forwarded to the Service Centre. In that case the service MAP_MO_FORWARD_SHORT_MESSAGE is not initiated. The acknowledge message from the Service Centre is forwarded to the mobile station (TS GSM 03.40, TS GSM 04.11).

The mobile originated short message service procedure is shown in figure 23.2/2.

Process MOSM_MSC

23.2_2.1(3)

Figure 23.2/2: The mobile originated short message service process in the MSC.

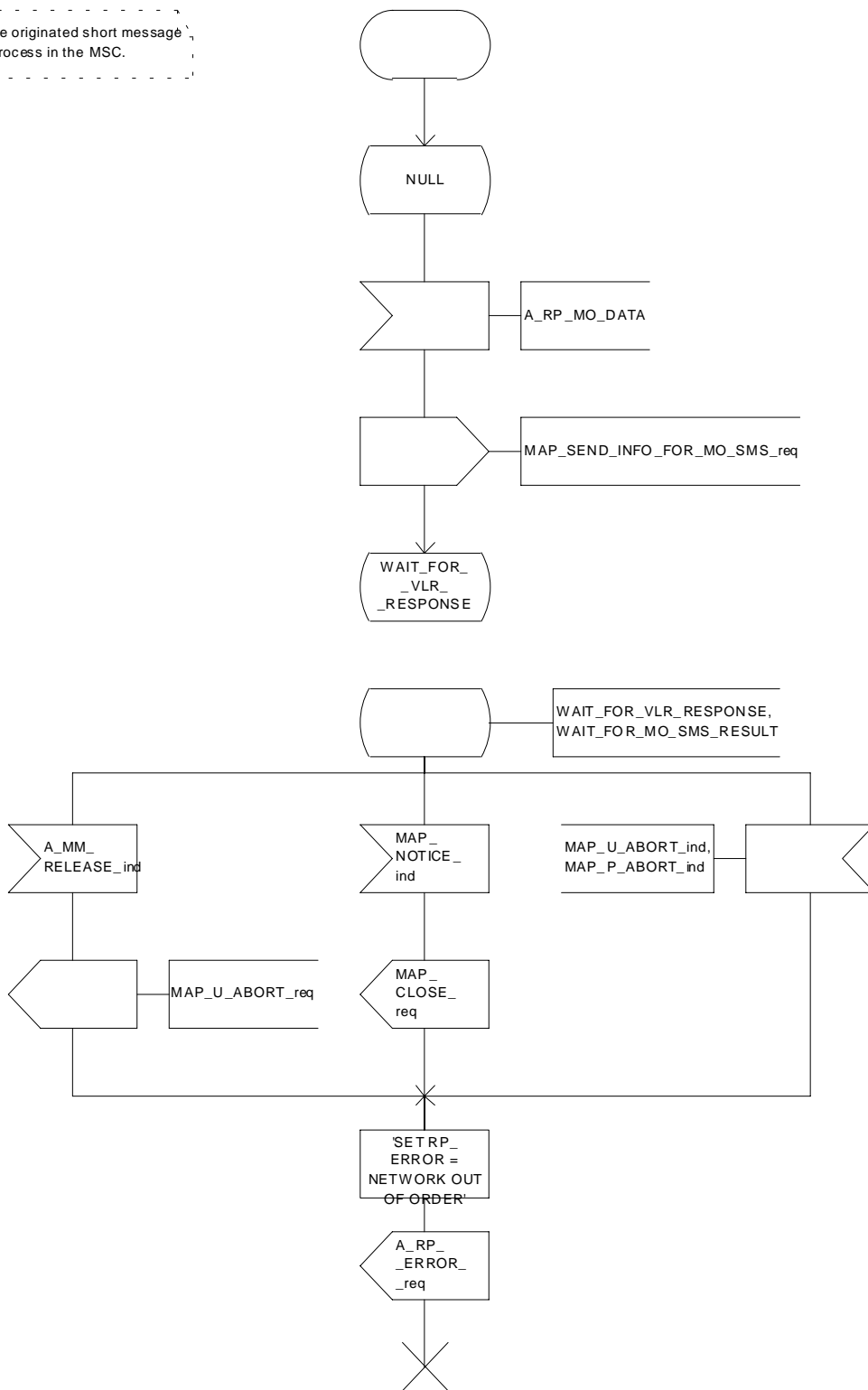


Figure 23.2/2 (sheet 1 of 4): Process MOSM_MSC

Process MOSM_MSC

23.2_2.2(4)

Figure 23.2/2: The mobile originated short message service process in the MSC.

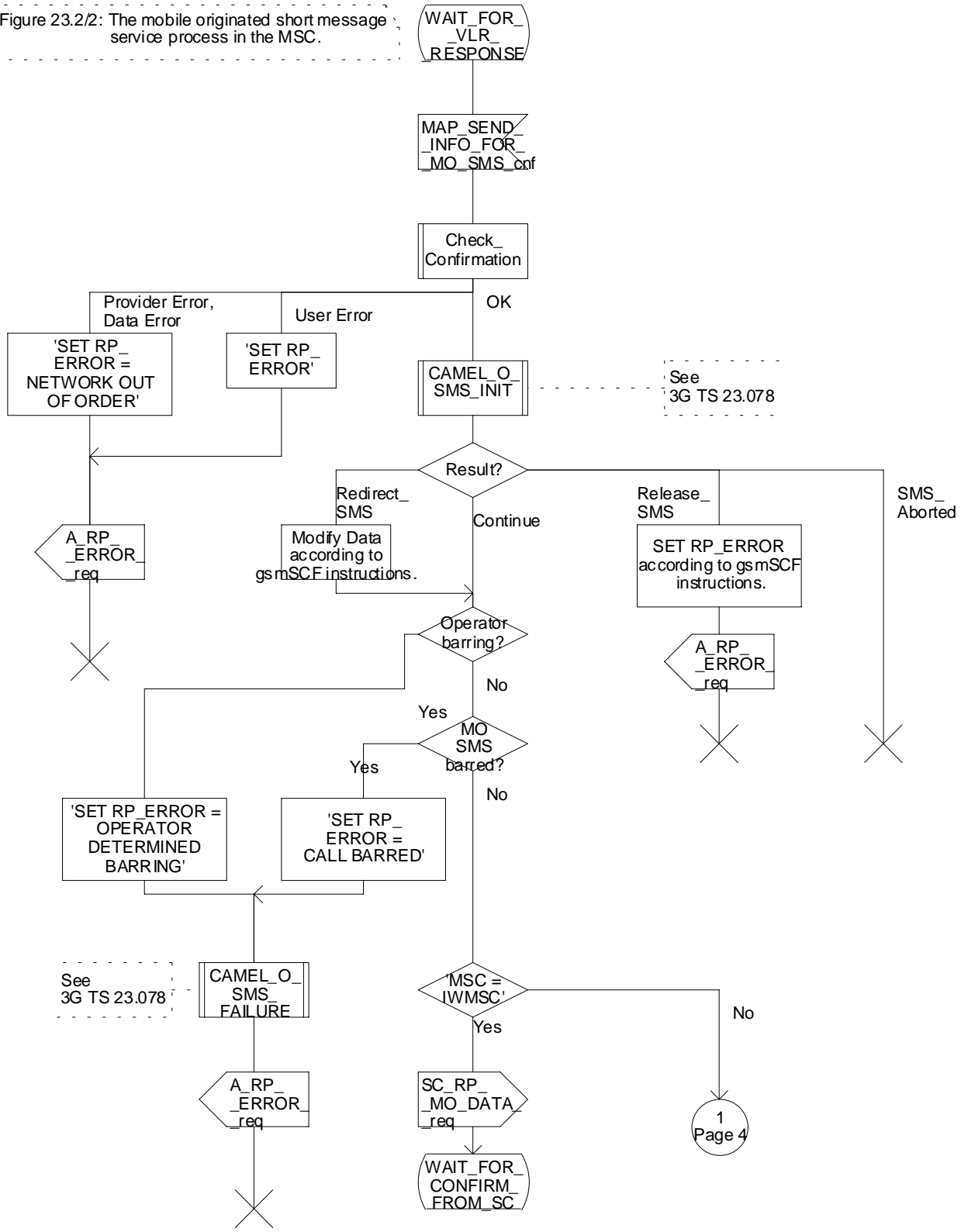


Figure 23.2/2 (sheet 2 of 4): Process MOSM_MSC

Process MOSM_MSC

23.2_2.new3(4)

Figure 23.2/2: The mobile originated short message service process in the MSC.

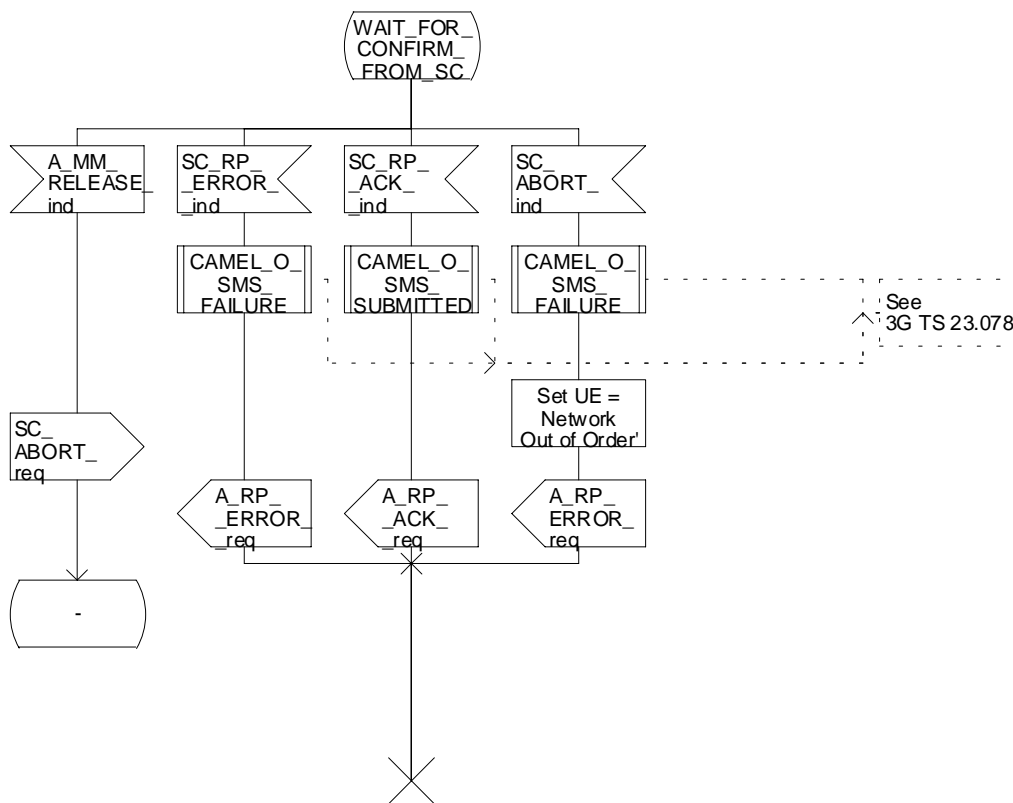


Figure 23.2/2 (sheet 3 of 4): Process MOSM_MSC

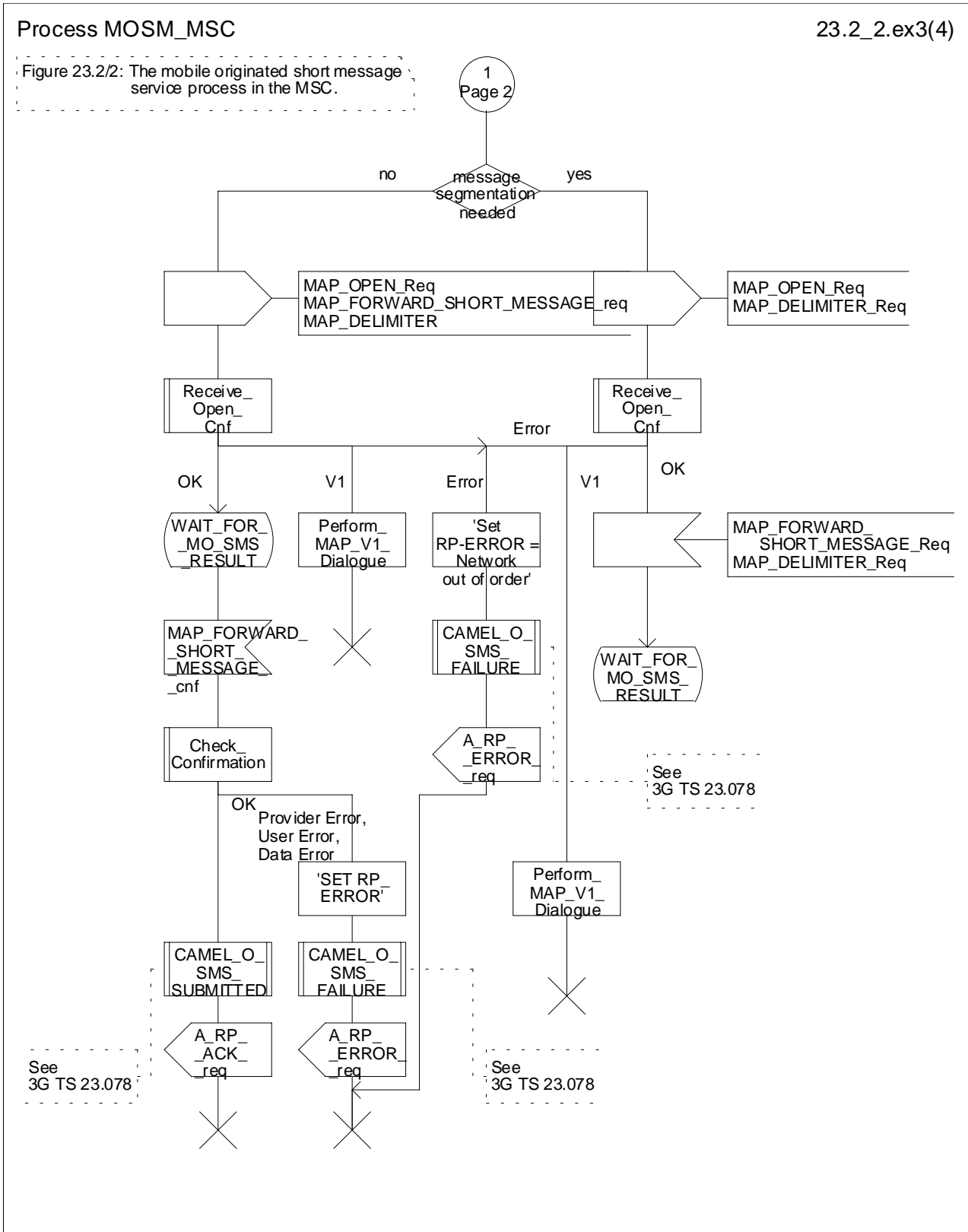


Figure 23.2/2 (sheet 4 of 4): Process MOSM_MSC

23.2.2 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 (BAOC), the call barred error with cause operator barring is returned;
- if the short message transfer would contravene the supplementary service call barring conditions (BAOC) 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 23.2/3.

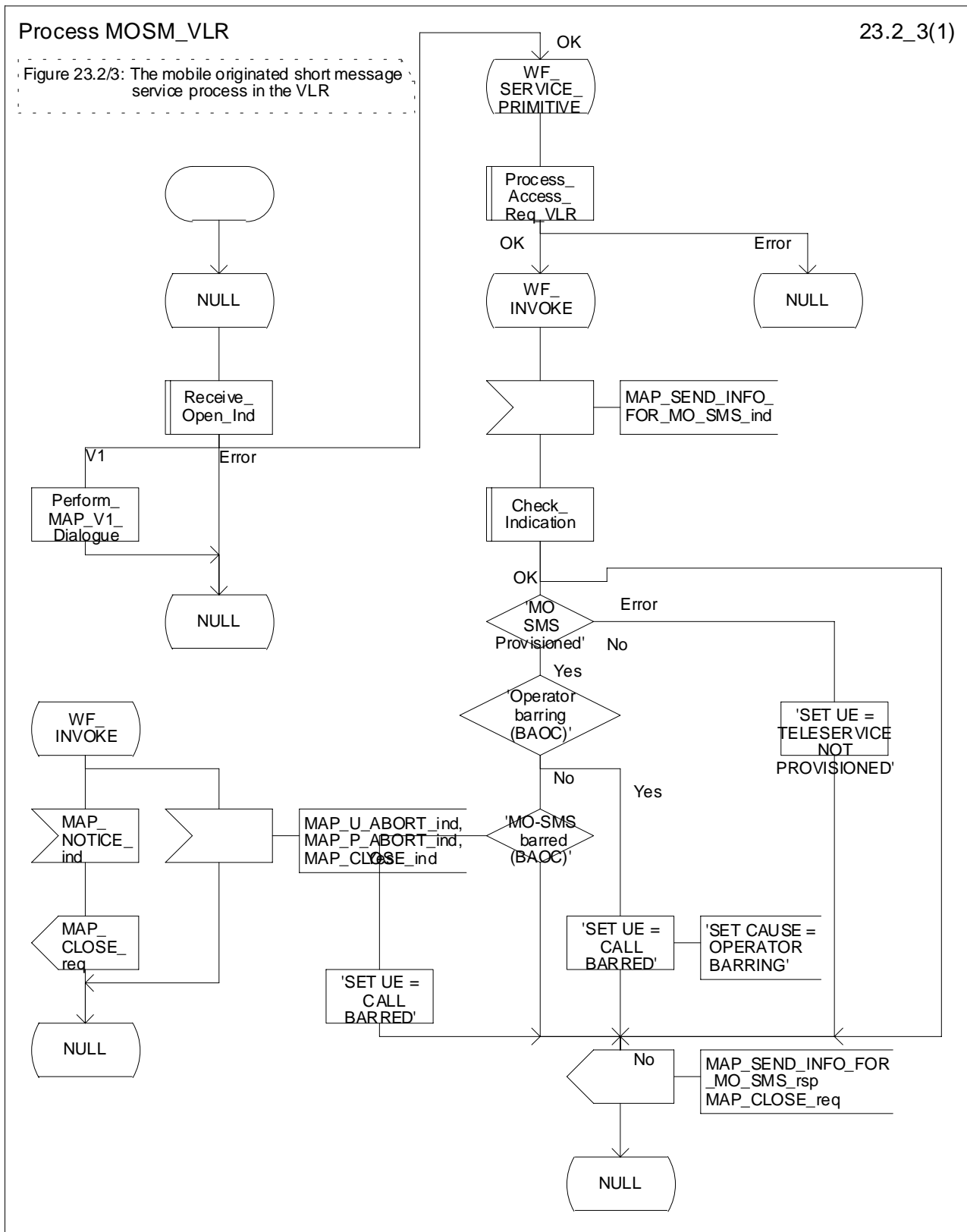


Figure 23.2/3: Process MOSM_VLR

23.2.3 Procedure in the interworking MSC

This procedure applies only when the IWMSC is not the servicing MSC or SGSN.

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_MO_FORWARD_SHORT_MESSAGE indication shall be received.

When a MAP_MO_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 or SGSN.

The outcome of the procedure with the Service Centre is awaited before a MAP_MO_FORWARD_SHORT_MESSAGE response is given back to the servicing MSC or SGSN:

- if a short message is accepted by the Service Centre, an acknowledgement is sent back to the servicing MSC or SGSN;
- if the Service Centre is not identified, the SM Delivery Failure error is returned to the servicing MSC or SGSN;
- 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 or SGSN.

The mobile originated short message service transfer in the IWMSC is shown in figure 23.2/4.

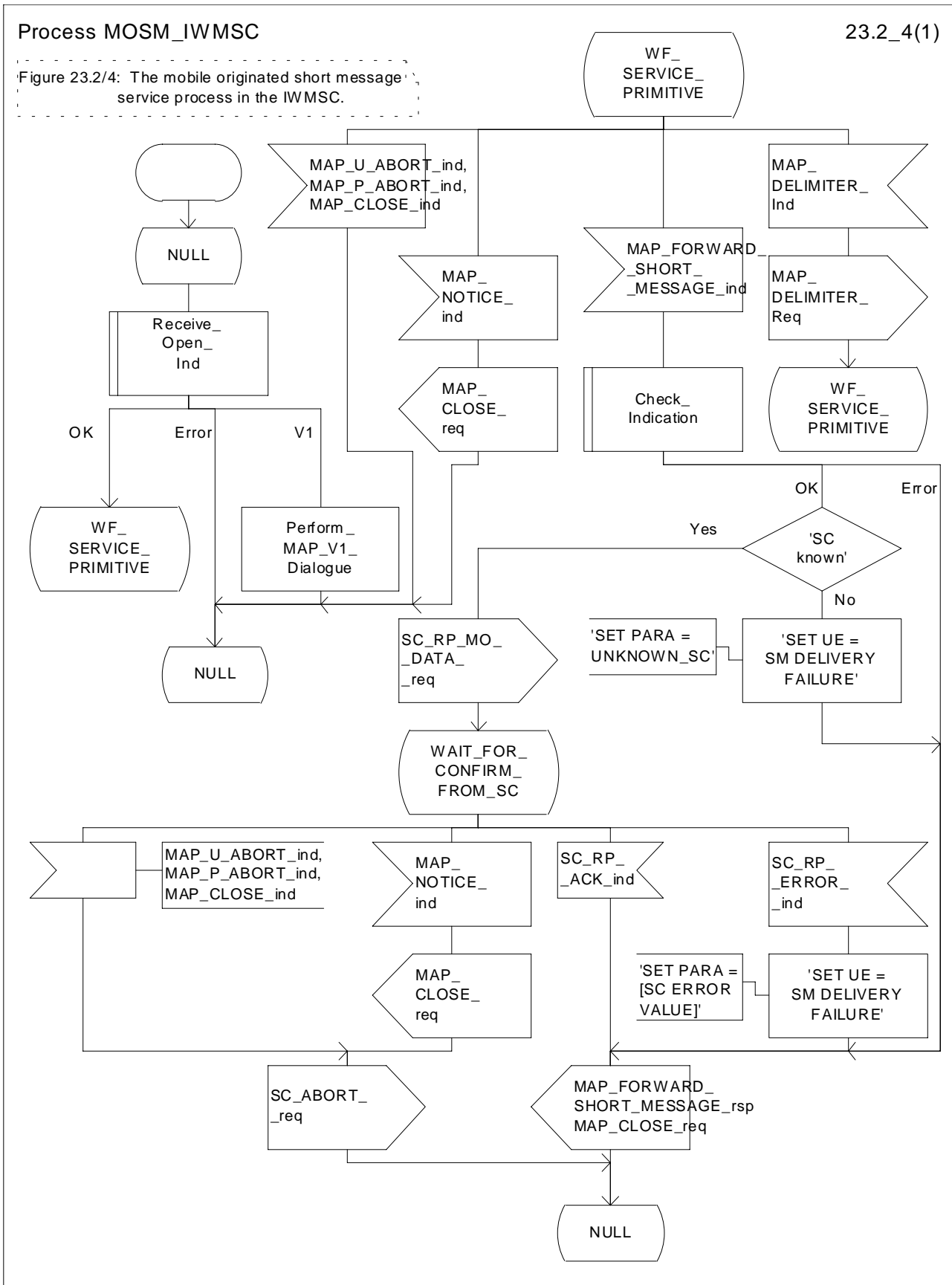


Figure 23.2/4: Process MOSM_IWMSC

23.2.4 Procedure in the servicing SGSN

When receiving the short message from the MS, the SGSN acts as follows:

- if there is incompatibility in the subscription check, the RP_ERROR cause requested facility not subscribed is provided to the mobile station;
- the SGSN opens a CAMEL dialogue as specified in 3G TS 23.078. If the CAMEL service bars the MO SM then the failure is reported to MS;
- if the short message transfer would contravene operator determined barring, , the failure is reported to the CAMEL service as specified in 3G TS 23.078 and the RP_ERROR cause operator determined barring is provided to the mobile station;

NOTE: The RP_ERROR causes are described in TS GSM 04.11

- if no error is detected, the short message transmission towards the IWMSC is initiated using the MAP_MO_FORWARD_SHORT_MESSAGE request.

If the service MAP_MO_FORWARD_SHORT_MESSAGE is started, the SGSN will check whether the grouping of MAP_OPEN request and MAP_MO_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_MO_FORWARD_SHORT_MESSAGE request is sent. As a response to the procedure, the servicing SGSN will receive the MAP_MO_FORWARD_SHORT_MESSAGE confirmation from the IWMSC indicating that:

- the short message has been successfully delivered to the Service Centre. The successful submission of SM is reported to the CAMEL service as specified in 3G TS 23.078 and the acknowledgement is sent to the mobile station;
- one of several error cases has occurred. The mapping between MAP error causes and RP_ERROR causes is described in TS GSM 03.40. The failure in SM submission is reported to the CAMEL service as specified in 3G TS 23.078 and the appropriate indication is provided to the mobile station.

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 mobile station.

The mobile originated short message service procedure is shown in figure 23.2/5

Process MOSM_SGSN

23.2_5.1(3)

Figure 23.2/5: The mobile originated short message service process in the SGSN.

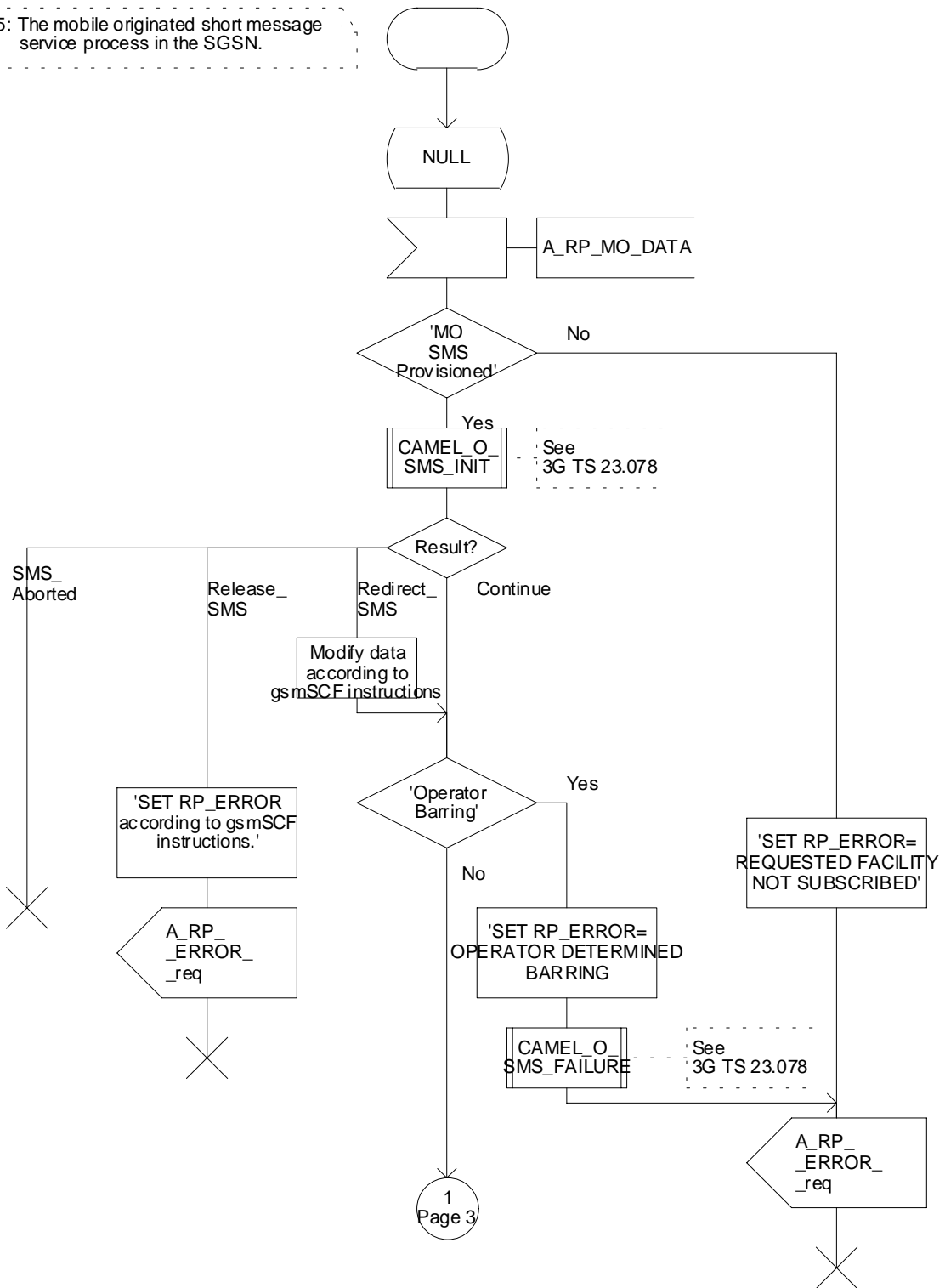


Figure 23.2/5 (sheet 1 of 3): Process MOSM_SGSN

Process MOSM_SGSN

23.2_5.2(3)

Figure 23.2/5: The mobile originated short message service process in the SGSN.

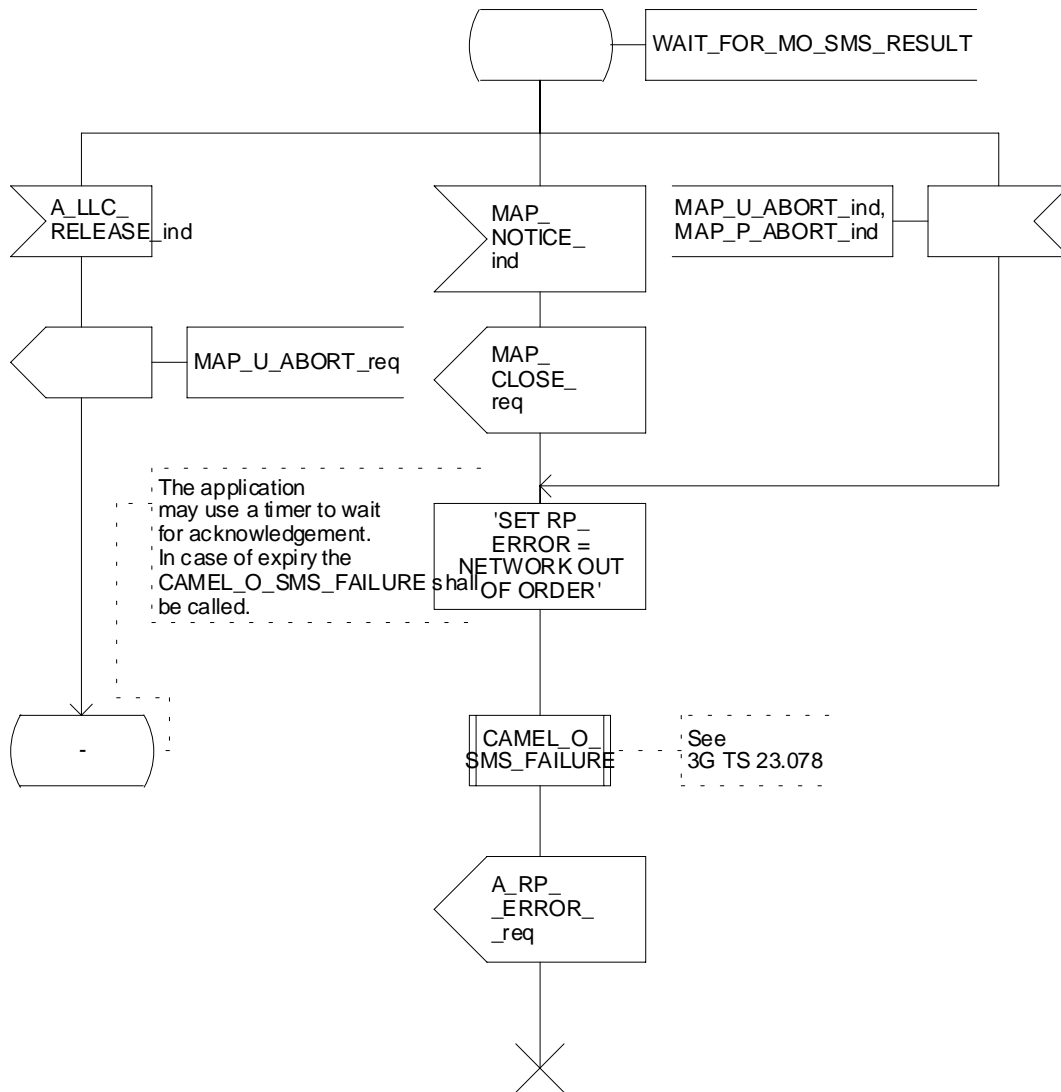


Figure 23.2/5 (sheet 2 of 3): Process MOSM_SGSN

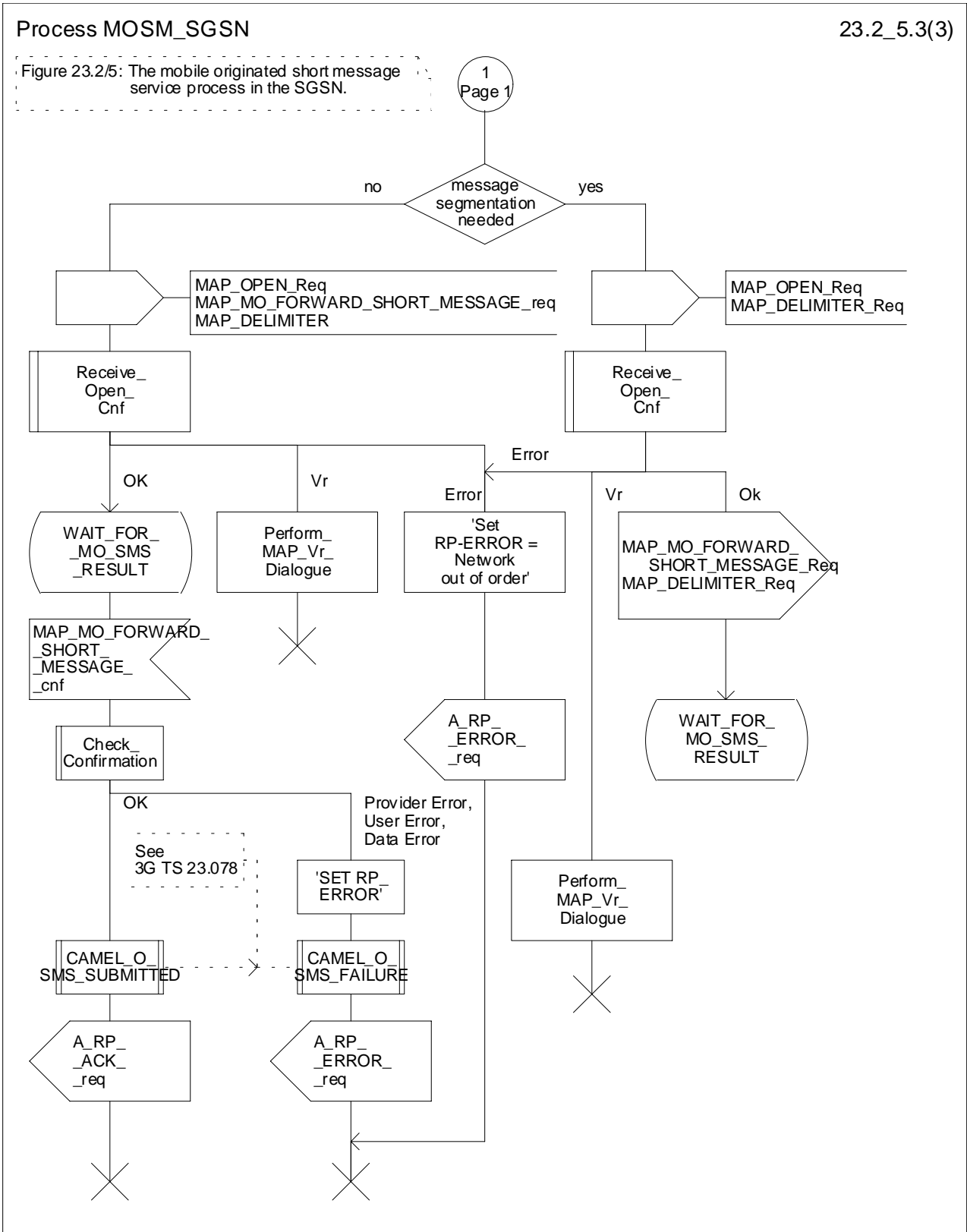
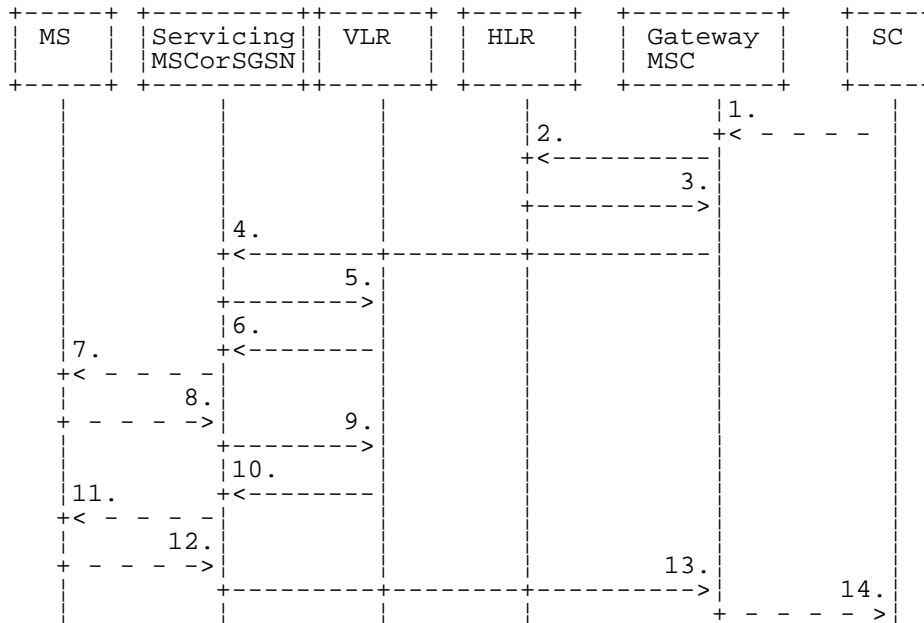


Figure 23.2/5 (sheet 3 of 3): Process MOSM_SGSN

23.3 The mobile terminated short message transfer procedure

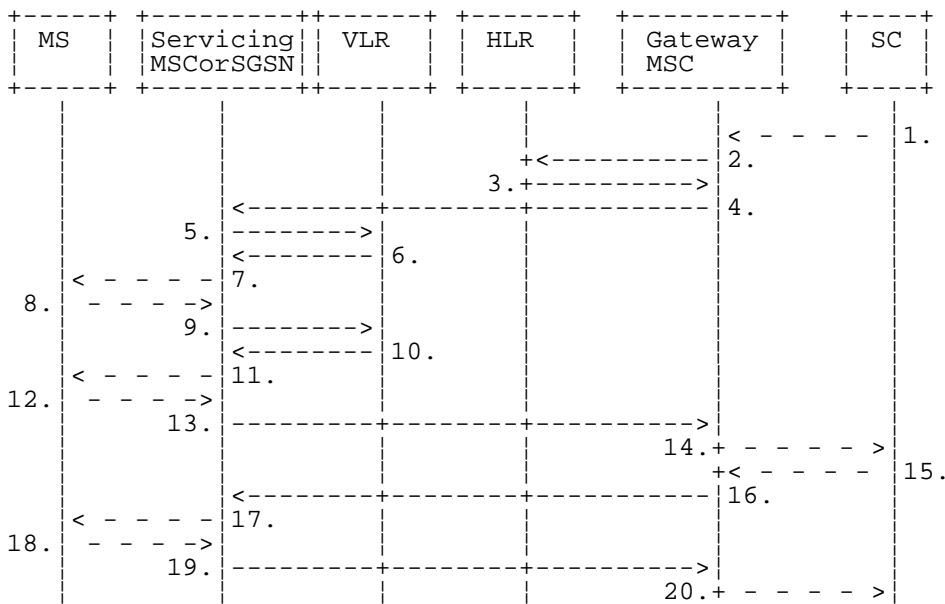
The mobile terminated short message transfer procedure is used for forwarding a short message or several short messages from a Service Centre to a mobile subscriber. The mobile terminated short message procedure for a single short message transfer is shown in figure 23.3/1.



- 1) Short Message (GSM 03.40)
 - 2) MAP_SEND_ROUTING_INFO_FOR_SM
 - 3) MAP_SEND_ROUTING_INFO_FOR_SM_ACK
 - 4) MAP_MT_FORWARD_SHORT_MESSAGE
 - 5) MAP_SEND_INFO_FOR_MT_SMS (*)
 - 6) MAP_PAGE/MAP_SEARCH_FOR_MOBILE_SUBSCRIBER (*)
 - 7) Page (GSM 04.08)
 - 8) Page response (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 (GSM 04.11)
 - 12) Short Message Acknowledgement (GSM 04.11)
 - 13) MAP_MT_FORWARD_SHORT_MESSAGE_ACK
 - 14) Short Message Acknowledgment (GSM 03.40)
- (*) Messages 5), 6), 9), and 10) are not used by SGSN

Figure 23.3/1: Mobile terminated short message service procedures

The mobile terminated short message procedure for multiple short message transfer is shown in figure 23.3/2.



- 1) Short Message (GSM 03.40)
 - 2) MAP_SEND_ROUTING_INFO_FOR_SM
 - 3) MAP_SEND_ROUTING_INFO_FOR_SM_ACK
 - 4) MAP_MT_FORWARD_SHORT_MESSAGE (note 1)
 - 5) MAP_SEND_INFO_FOR_MT_SMS (*)
 - 6) MAP_PAGE/MAP_SEARCH_FOR_MOBILE_SUBSCRIBER (*)
 - 7) Page (GSM 08.08)
 - 8) Page response (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 (GSM 04.11)
 - 12) Short Message Acknowledgement (GSM 04.11)
 - 13) MAP_MT_FORWARD_SHORT_MESSAGE_ACK
 - 14) Short Message Acknowledgment (GSM 03.40)
 - 15) Short Message (GSM 03.40)
 - 16) MAP_MT_FORWARD_SHORT_MESSAGE (note 2)
 - 17) Short Message (GSM 04.11)
 - 18) Short Message Acknowledgement (GSM 04.11)
 - 19) MAP_MT_FORWARD_SHORT_MESSAGE_ACK
 - 20) Short Message Acknowledgment (GSM 03.40)
- (*) Messages 5), 6), 9), and 10) are not used by SGSN

NOTE 1: The More Messages To Send flag is TRUE.

NOTE 2: The More Messages To Send flag is FALSE

Figure 23.3/2: Mobile terminated short message procedure for multiple short message transfer

In the multiple short message transfer the service MAP_MT_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 8.3); (*)
- MAP_PAGE (see subclause 8.2); (*)
- MAP_SEARCH_FOR_MS (see subclause 8.2); (*)
- MAP_AUTHENTICATE (see subclause 8.5); (*)
- MAP_SET_CIPHERING_MODE (see subclause 8.6); (*)
- MAP_CHECK_IMEI (see subclause 8.7);
- MAP_FORWARD_NEW_TMSI (see subclause 8.9); (*)
- MAP_REPORT_SM_DELIVERY_STATUS (see subclause 12.3);
- MAP_INFORM_SERVICE_CENTRE see subclause 12.6);
- MAP_TRACE_SUBSCRIBER_ACTIVITY (see subclause 9.1); (*)
- MAP_READY_FOR_SM (see subclause 12.4).

(*) Those messages are not used by SGSN.

23.3.1 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_MT_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_MT_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_MT_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_MT_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_MT_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_MT_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:

- unidentified subscriber or system failure error. The error code is forwarded to the GMSC;
- absent subscriber error. The absent subscriber_SM error is forwarded to the GMSC with the absent subscriber diagnostic indication set to 'IMSI Detached';
- 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;
- subscriber busy for MT SMS. The error code is forwarded to the GMSC;
- paging procedure invocation (see subclause 25.3) reporting the successful outcome of the procedure;
- search procedure invocation (see subclause 25.3) 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 25.4);
- 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_SM error is forwarded to the GMSC with the absent subscriber diagnostic indication set to 'No Paging Response', 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 7.6.1.4) 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_MT_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_MT_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 20.

The mobile terminated short message transfer procedure in the servicing MSC is shown in figures 23.3/3 and 23.3/4. The page and search procedures are shown in figure 25.3/1 and 25.3/2.

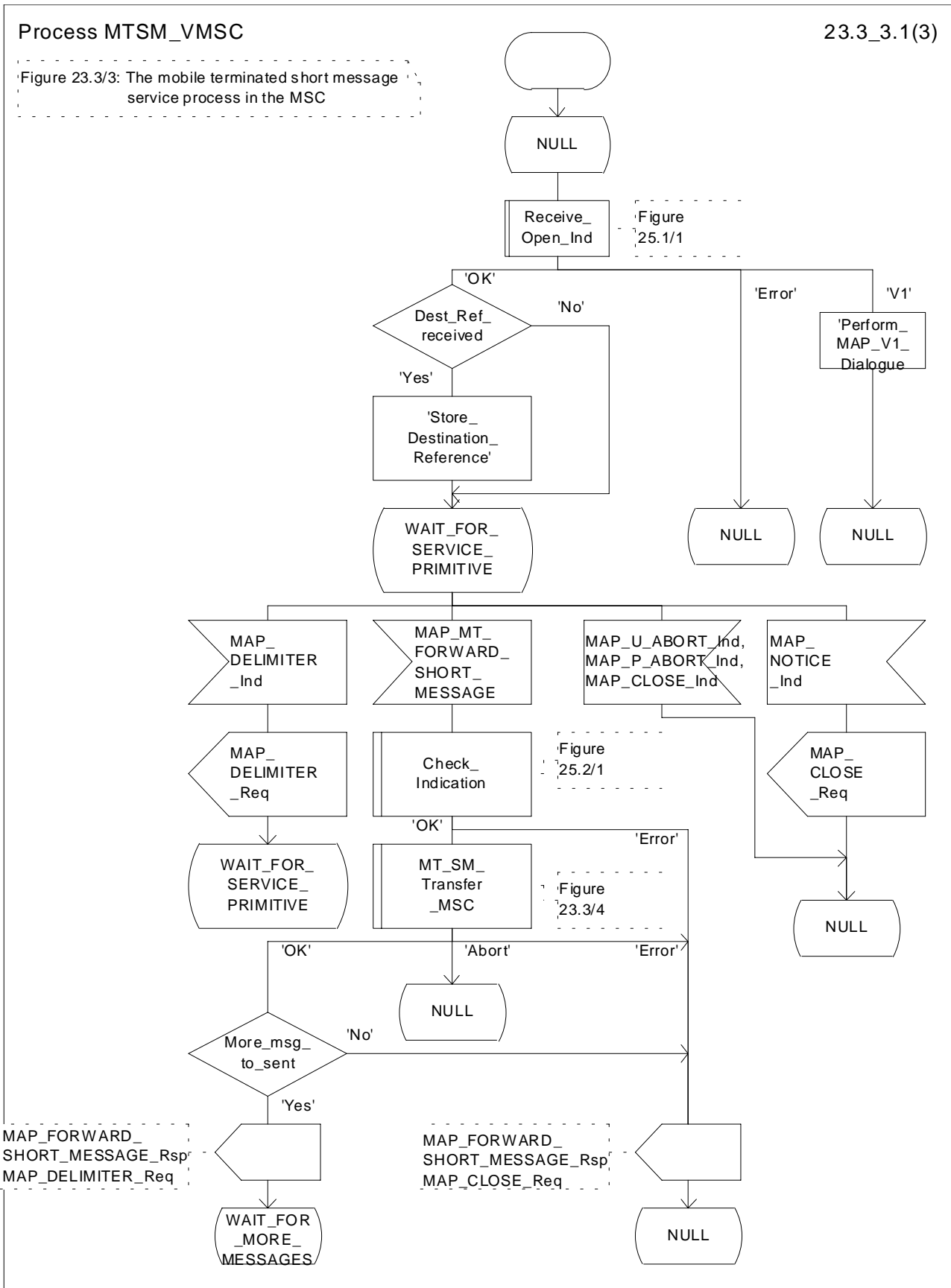


Figure 23.3/3 (sheet 1 of 3): Procedure MTSM_VMSC

Process MTSM_VMSC

23.3_3.2(3)

Figure 23.3/3: The mobile terminated short message service process in the MSC

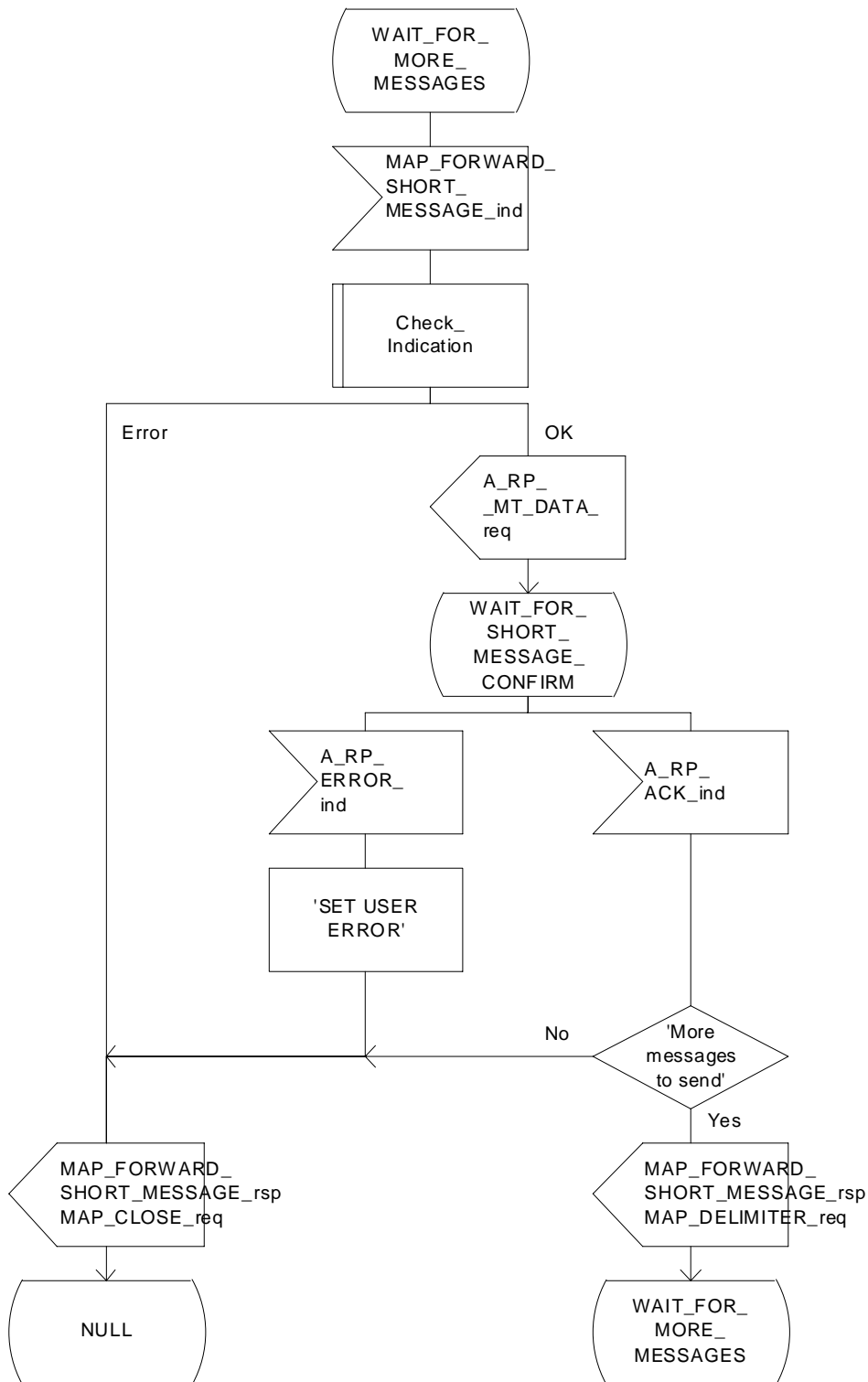


Figure 23.3/3 (sheet 2 of 3): Procedure MTSM_VMSC

Process MTSM_VMSC

23.3_3.3(3)

Figure 23.3/3: The mobile terminated short message service process in the MSC

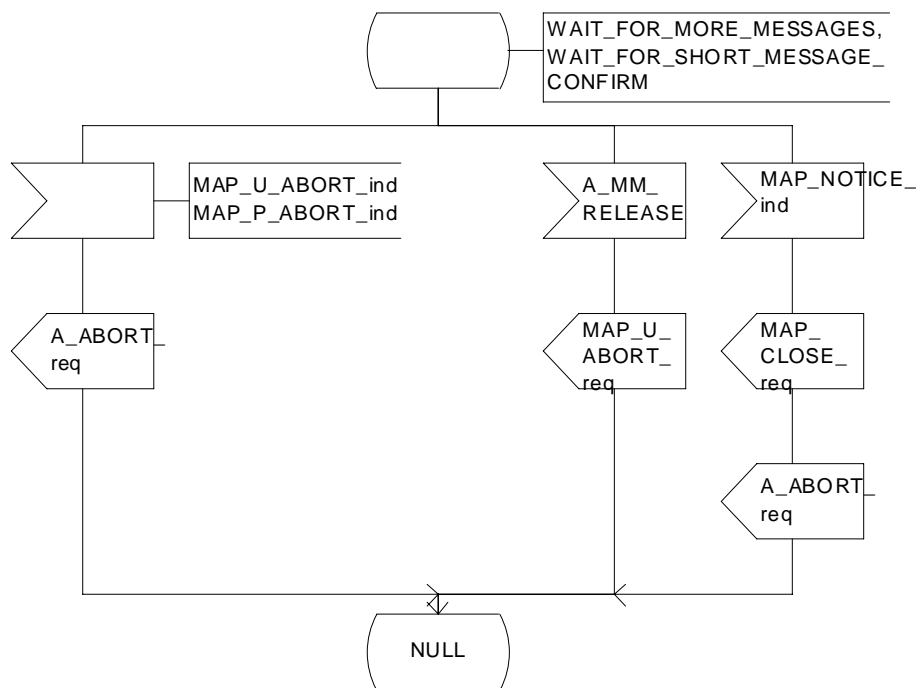


Figure 23.3/3 (sheet 3 of 3): Procedure MTSM_VMSC

Macrodefinition MT_SM_Transfer_MSC

23.3_4.1(3)

Figure 23.3/4: The mobile terminated short message transfer macro in the MSC

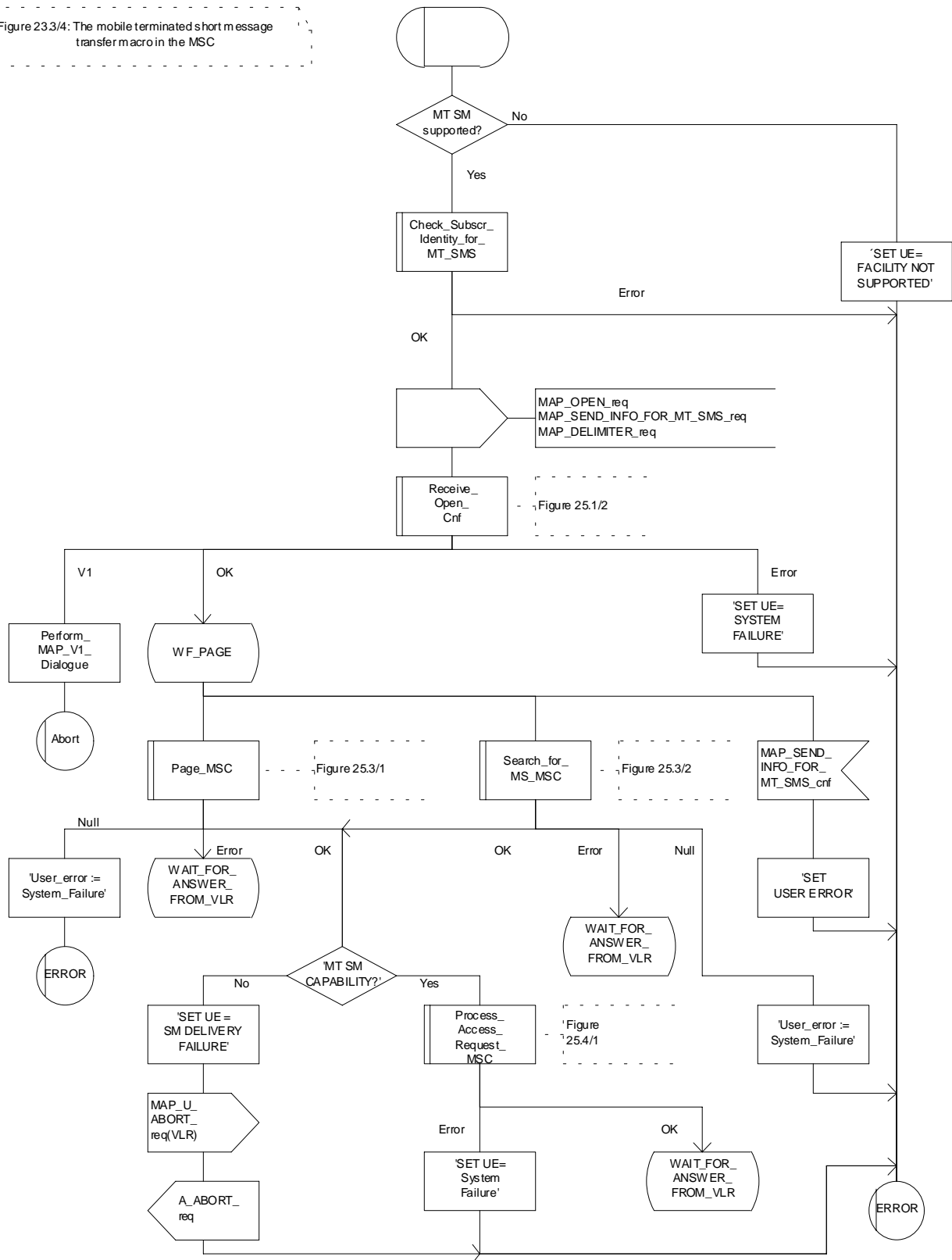
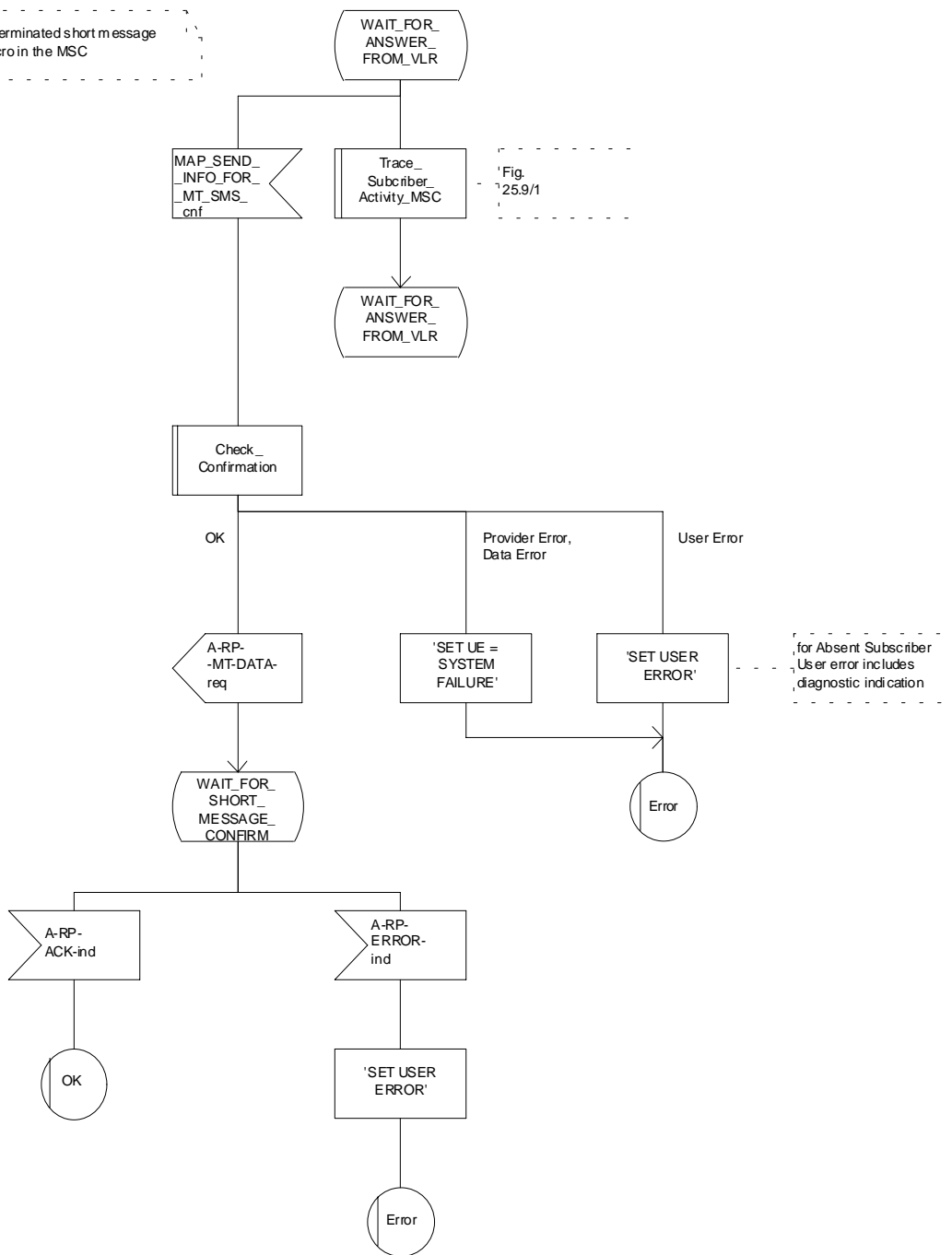


Figure 23.3/4 (sheet 1 of 3): Macro MT_SM_Transfer_MSC

Macrodefinition MT_SM_Transfer_MSC

23.3_4.2(3)

Figure 23.3/4: The mobile terminated short message transfer macro in the MSC



for Absent Subscriber User error includes diagnostic indication

Figure 23.3/4 (sheet 2 of 3): Macro MT_SM_Transfer_MSC

Macrodefinition MT_SM_Transfer_MSC

23.3_4.3(3)

Figure 23.3/4: The mobile terminated short message transfer macro in the MSC

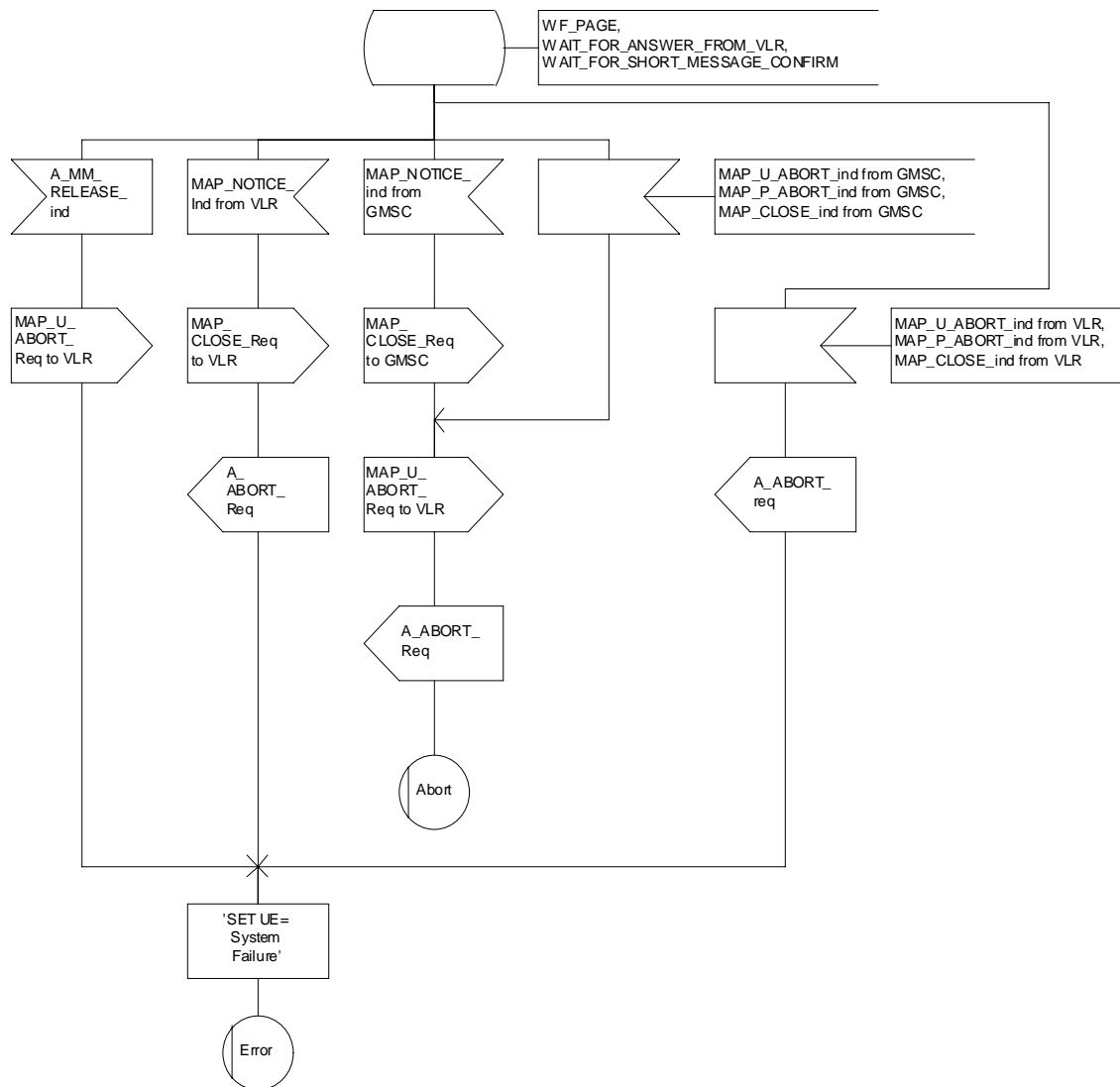


Figure 23.3/4 (sheet 3 of 3): Macro MT_SM_Transfer_MSC

23.3.2 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 with the diagnostic indication set to 'IMSI Detached' 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, the absent subscriber error is forwarded with the diagnostic indication set to 'No Paging Response for non GPRS'. If the unexpected data value, or unknown location area error is received, the system failure indication is given to the MSC; if subscriber busy for MT SMS is received, this cause 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 23.3/5.

Process MT_SM_VLR

23.3_5.1(3)

Figure 23.3/5: The mobile terminated short message service process in the VLR

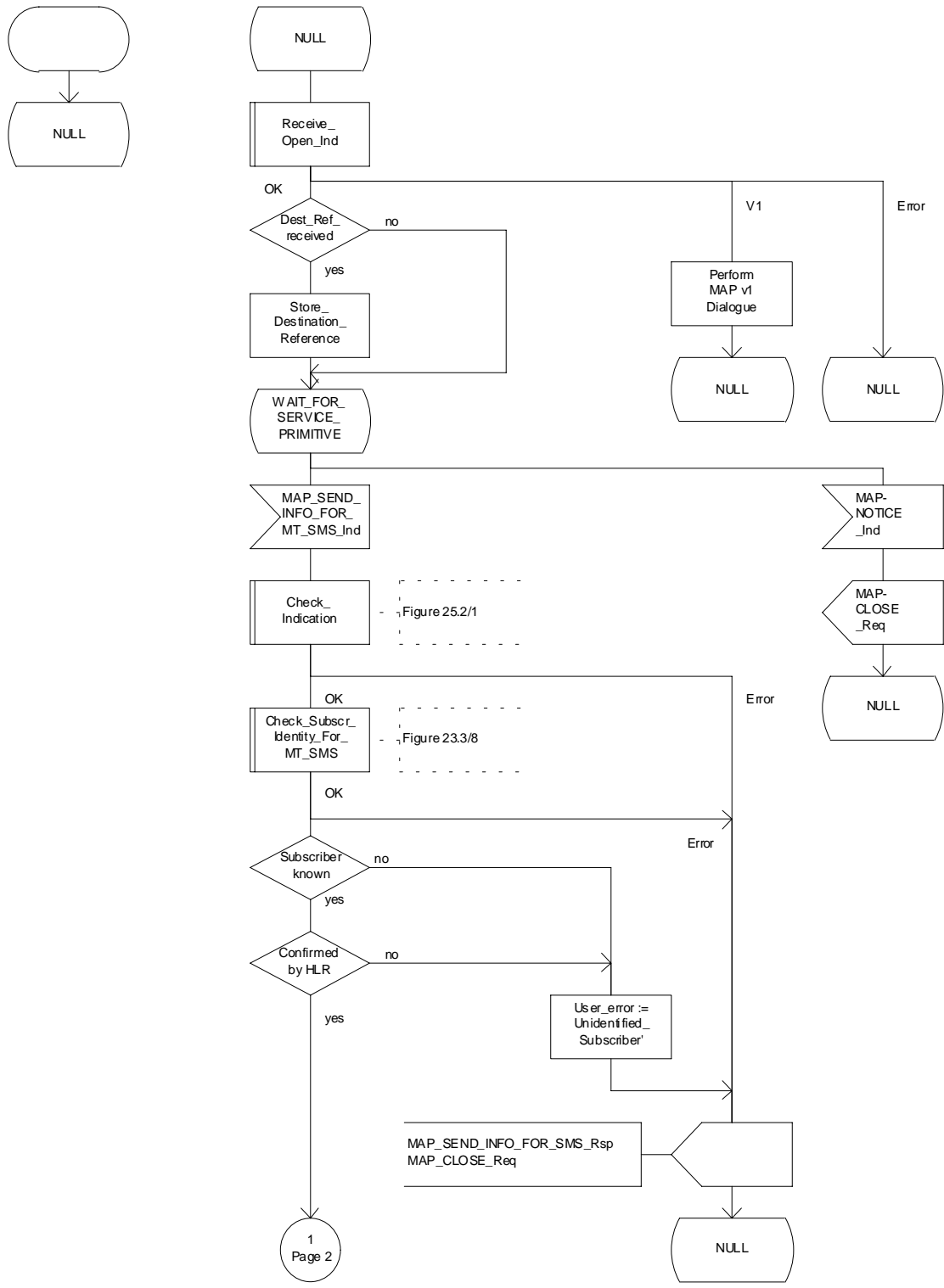


Figure 23.3/5 (sheet 1 of 3): Process MT_SM_VLR

Process MT_SM_VLR

23.3_5.2(3)

Figure 23.3/5: The mobile terminated short message service process in the VLR

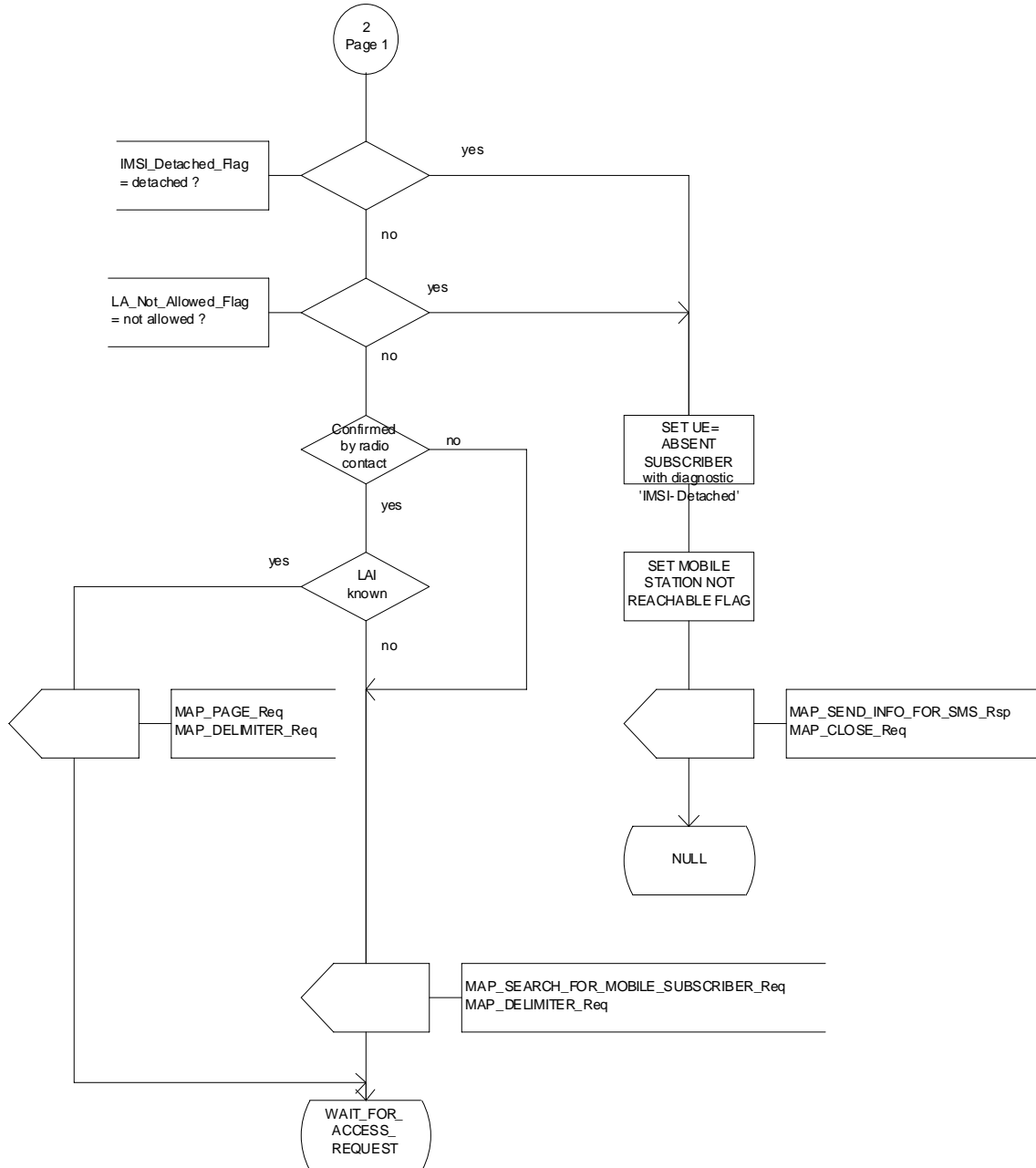


Figure 23.3/5 (sheet 2 of 3): Process MT_SM_VLR

Process MT_SM_VLR

23.3_5.3(3)

Figure 23.3/5: The mobile terminated short message service process in the VLR

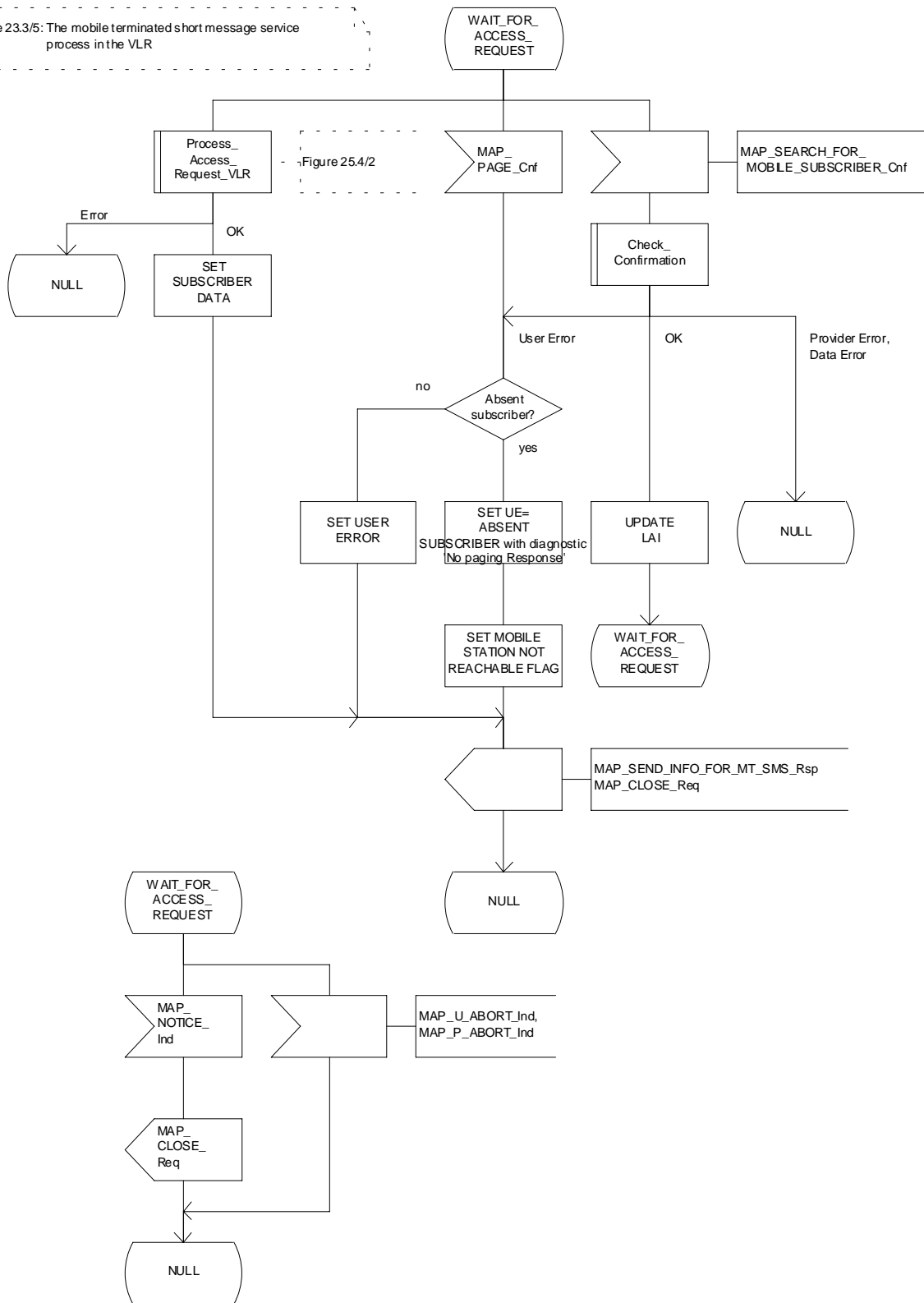


Figure 23.3/5 (sheet 3 to 3): Process MT_SM_VLR

23.3.3 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 « SM filtering by the HPLMN » function criteria, the call barred error with cause unauthorised Message Originator is returned (the definition of the filtering function is out of the scope of GSM specification. Filtering may be based on SM-RP-SMEA information element if received from the GMSC) ;
- 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 GMSC does not support the GPRS functionality, the behaviour of the HLR depends on the following conditions:
 - If the subscriber is not a GPRS subscriber then the behaviour of the HLR shall be the same as for a subscriber only registered as non GPRS and for SMS delivery.
 - If the subscriber is a GPRS subscriber and a non-GPRS subscriber with the option « transfer of SM via the MSC when GPRS is not supported in the GMSC » then the behaviour of the HLR shall be the same as for a subscriber only registered as non GPRS and for SMS delivery.
 - If the subscriber is a GPRS subscriber and a non-GPRS subscriber with the option « transfer of SM via the SGSN when GPRS is not supported in the GMSC » or if the subscriber is a GPRS subscriber only then the behaviour of the HLR shall be the same as for the case transfer over GPRS described in MAP release 97, with the following precision : because GMSC does not support MAP release 97, the previous MAP protocol release is used. When the HLR sends the MAP_SEND_ROUTING_INFO_FOR_SM_Resp, the SGSN number is mapped to the MAP parameter « MSC number ». When the HLR sends the MAP_INFORM_SERVICE_CENTRE_resp, the MNRG status shall be mapped to the MAP parameter « mnrf-set ».

The HLR may send the MSC, SGSN or both numbers as routing information to SMS-GMSC based on the following:

- A) The subscriber may only be registered as non GPRS and for SMS delivery:
- if the short message transfer would contravene the supplementary service barring, the call barred error with cause barring service active is returned;
 - if the location registration of the mobile subscriber shows that the VLR in 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 for non GPRS" flag is set, i.e. the MS is not reachable, the MSISDN-Alert and the SC address are included in the MWD (if possible), the flag MNRF is set and the "Absent Subscriber_SM" error is returned with the appropriate absent subscriber diagnostic indication, i.e. 'Deregistered in HLR for non GPRS ', 'Roaming Restricted' or 'MS-Purged for non GPRS '.

The priority parameter (SM_RP_PRI) is processed as follows:

- if the priority is low (SM_RP_PRI = False) and the mobile station not reachable flag (MNRF) is set, an absent subscriber_SM error is returned. If a reason for the subscriber's absence for non GPRS is stored in the mobile not reachable reason (MNRR) in the subscriber data, then this is returned with the absent subscriber_SM error. 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 with MSC number 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 with MSC number 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.

B) The subscriber may only be registered as GPRS and for SMS delivery:

- if the location registration of the mobile subscriber shows that the SGSN in the visited PLMN does not support the MT short message service, the facility not supported error is returned;
- if no SGSN identity is stored for the mobile subscriber or the "SGSN Area Restricted Flag" is set or the "MS purged for GPRS" flag is set, i.e. the MS is not reachable, the MSISDN-Alert and the SC address are included in the MWD (if possible), the flag MNRG is set and the "Absent Subscriber_SM" error is returned with the appropriate absent subscriber diagnostic indication, i.e. 'Deregistered in HLR for GPRS', 'Roaming Restricted' or 'MS-Purged for GPRS'.

The priority parameter (SM_RP_PRI) is processed as follows:

- if the priority is low (SM_RP_PRI = False) and the mobile station not reachable for GPRS (MNRG) flag is set, an absent subscriber_SM error is returned. If a reason for the subscriber's absence for GPRS is stored in the mobile not reachable reason (MNRR) in the subscriber data, then this is returned with the absent subscriber_SM error. 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 MNRG is clear, the routing information with SGSN number is retrieved as described below;
- if the priority is high (SM_RP_PRI = True) and the MNRG is set, the HLR will send the acknowledge primitive containing the routing information with SGSN number 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.

C) The subscriber may be registered as non GPRS and GPRS and for SMS Delivery:

- if the short message transfer would contravene the supplementary service barring, the behaviour is the same as for a subscriber only registered for GPRS and SMS delivery.
- if the location registration of the mobile subscriber shows that the VLR in the visited PLMN does not support the MT short message service, the behaviour is the same as for a subscriber only registered for GPRS and SMS delivery;
- if the location registration of the mobile subscriber shows that the SGSN in the visited PLMN does not support the MT short message service, the behaviour is the same as for a subscriber only registered for non GPRS and SMS delivery;
- if no MSC and SGSN identities are stored for the mobile subscriber or the "MSC and SGSN Area Restricted Flags" are set or the "MS purged for non GPRS and GPRS" flags are set or a combination of these errors for non GPRS and GPRS are used, i.e. the MS is not reachable, the MSISDN-Alert and the SC address are included in the MWD (if possible), the flags MNRF and MNRG are set and the "Absent Subscriber_SM" error is returned with the appropriate absent subscriber diagnostic indication, i.e. 'Deregistered in HLR for non GPRS or GPRS', 'Roaming Restricted', 'MS-Purged for non GPRS or GPRS' or both.

The priority parameter (SM_RP_PRI) is processed as follows:

- if the priority is low (SM_RP_PRI = False), the MNRF and MNRG are set, an absent subscriber_SM error is returned. If reasons for the subscriber's absence for non GPRS and GPRS are stored in MNRR in the subscriber data, then this is returned with the absent subscriber_SM error. 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 and MNRG is set, the routing information with MSC number is retrieved as described below;
- if the priority is low (SM_RP_PRI = False), and the MNRF is set and MNRG is clear, the routing information with SGSN number is retrieved as described below
- if the priority is low (SM_RP_PRI = False), and the MNRF and MNRG are clear, the routing information with MSC and SGSN numbers is retrieved as described below;
- if the priority is high (SM_RP_PRI = True) and the MNRF, the MNRG or both are set, the HLR will send the acknowledge primitive containing the routing information with both MSC and SGSN numbers 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, MNRF, MNRG or both 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, SGSN or both numbers and may be optionally accompanied by the LMSI.
- an indication specifying which number belongs the MSC and the SGSN will be returned to the GSMC.

LMSI shall not be used in case only the SGSN number is sent by HLR.

The mobile terminated short message transfer procedure in the HLR is shown in figure 23.3/6.

Process Mobile_terminated_SM_HLR

23.3_6.1(5)

Figure 23.3/6: The mobile terminated short message service process in the HLR in case the subscriber is registered as non-GPRS and/or GPRS

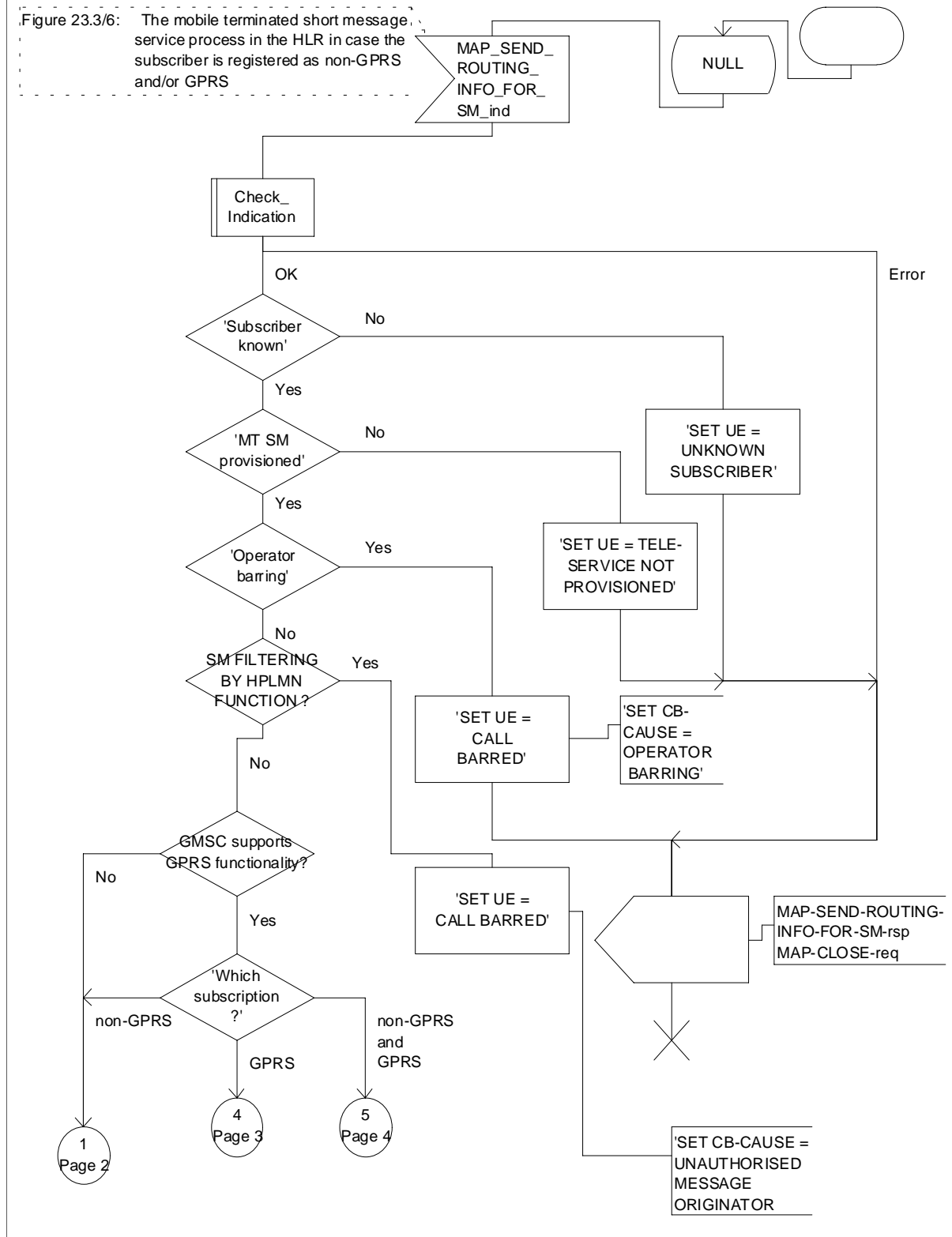


Figure 23.3/6 (sheet 1 of 5): Process Mobile_terminated_SM_HLR

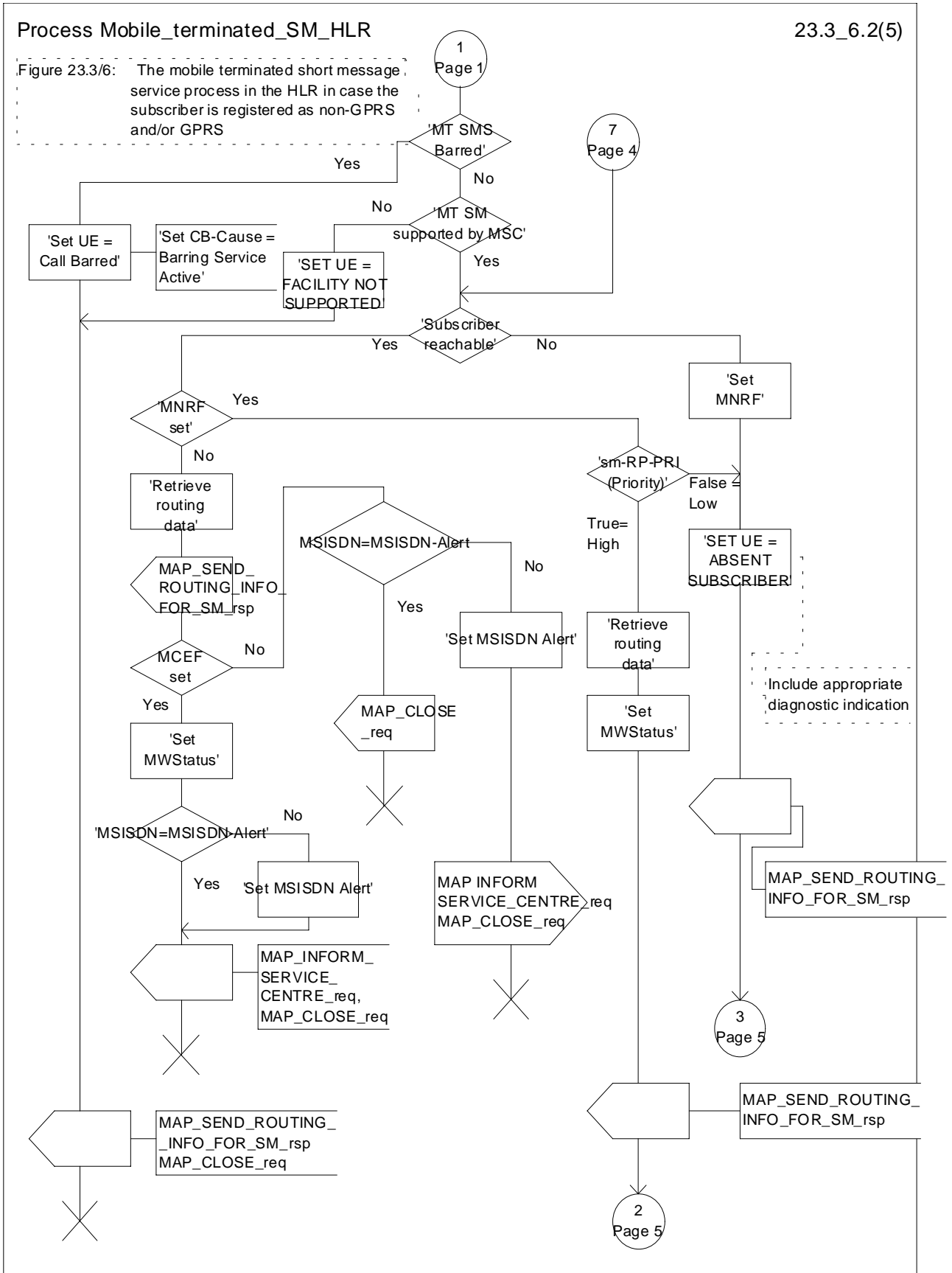


Figure 23.3/6 (sheet 2 of 5): Process Mobile_terminated_SM_HLR

Process Mobile_terminated_SM_HLR

23.3_6.3(5)

Figure 23.3/6: The mobile terminated short message service process in the HLR in case the subscriber is registered as non-GPRS and/or GPRS

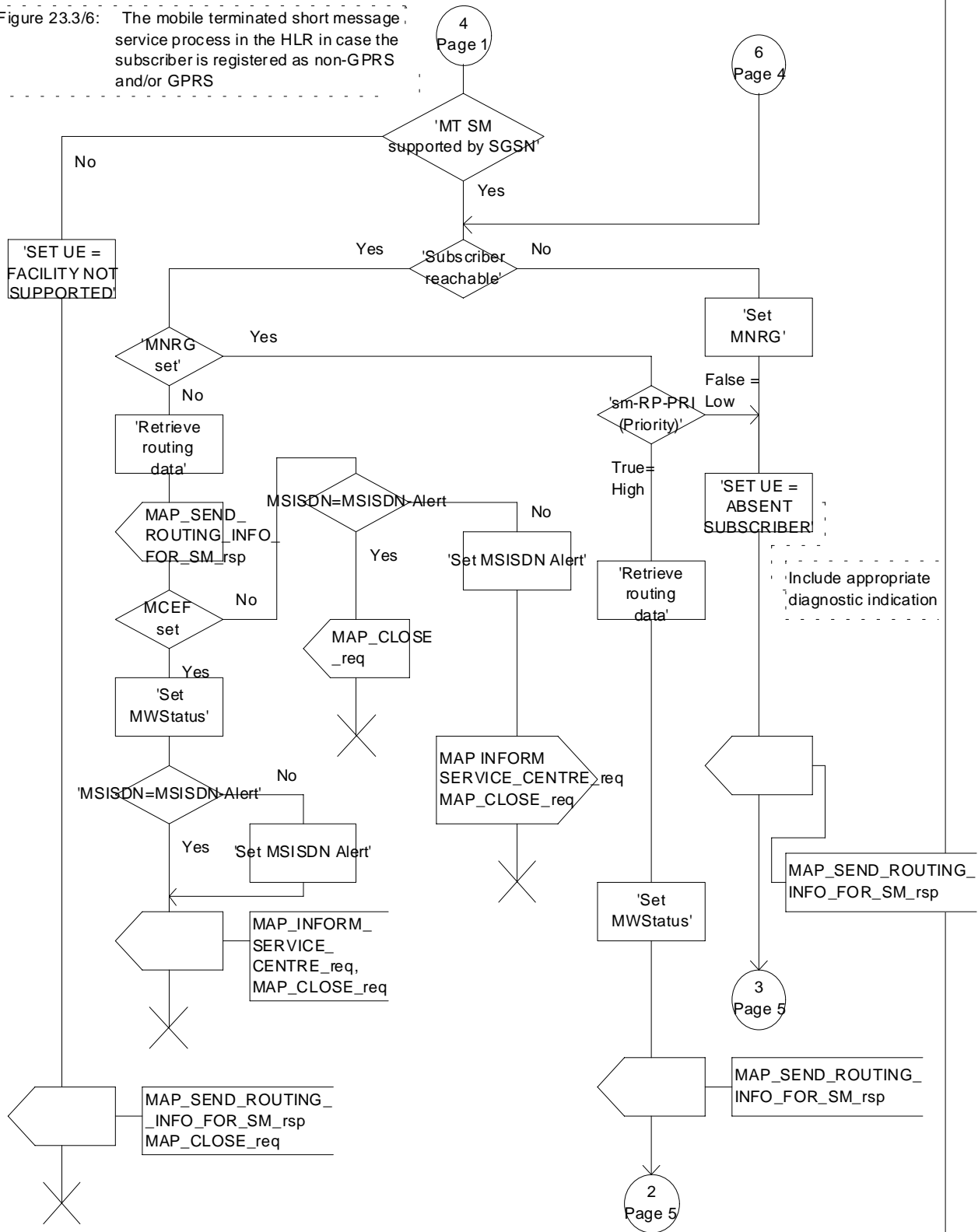


Figure 23.3/6 (sheet 3 of 5): Process Mobile_terminated_SM_HLR

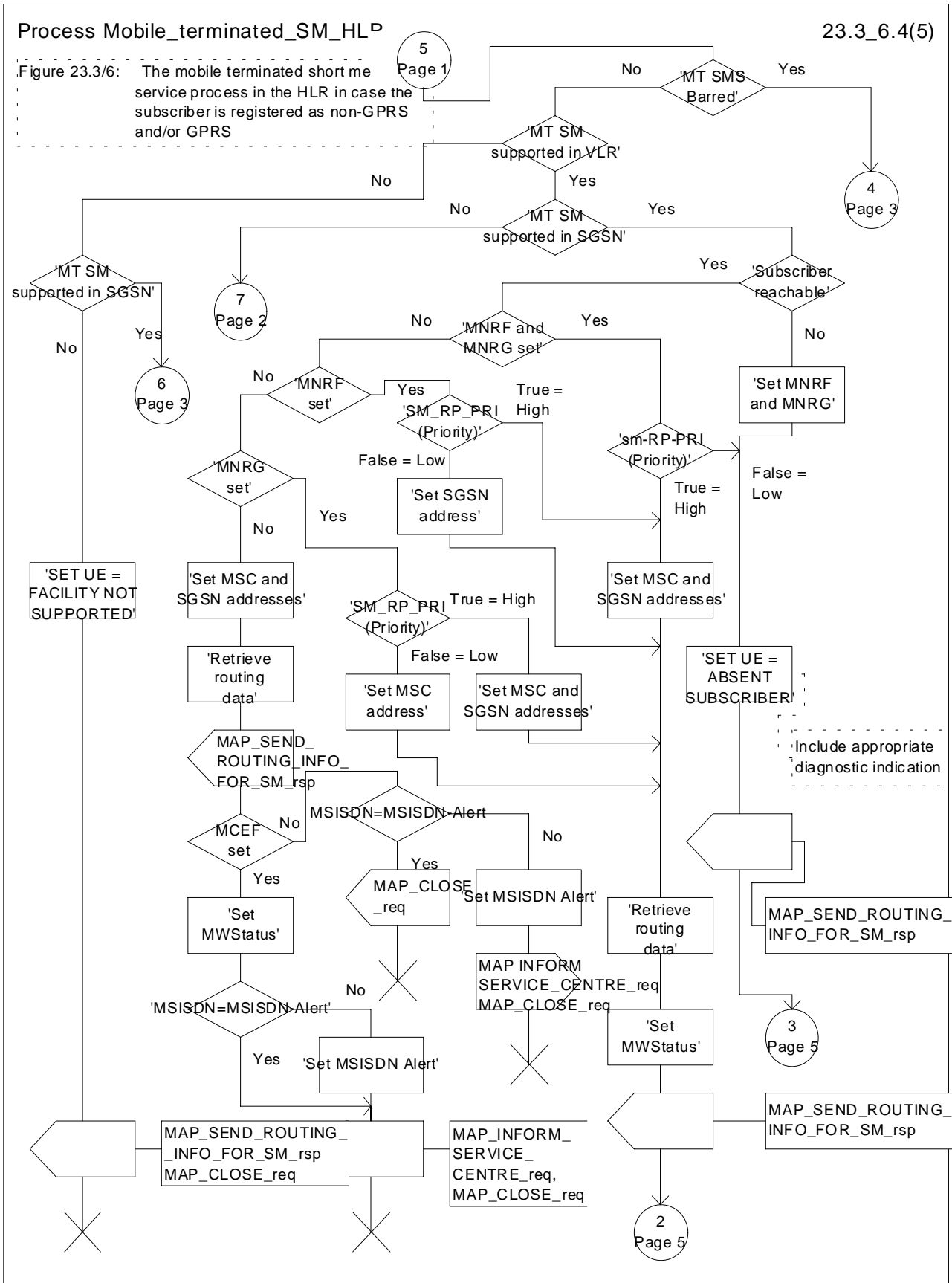


Figure 23.3/6 (sheet 4 of 5): Process Mobile_terminated_SM_HLR

Process Mobile_terminated_SM_HLR

23.3_6.5(5)

Figure 23.3/6: The mobile terminated short message service process in the HLR in case the subscriber is registered as non-GPRS and/or GPRS

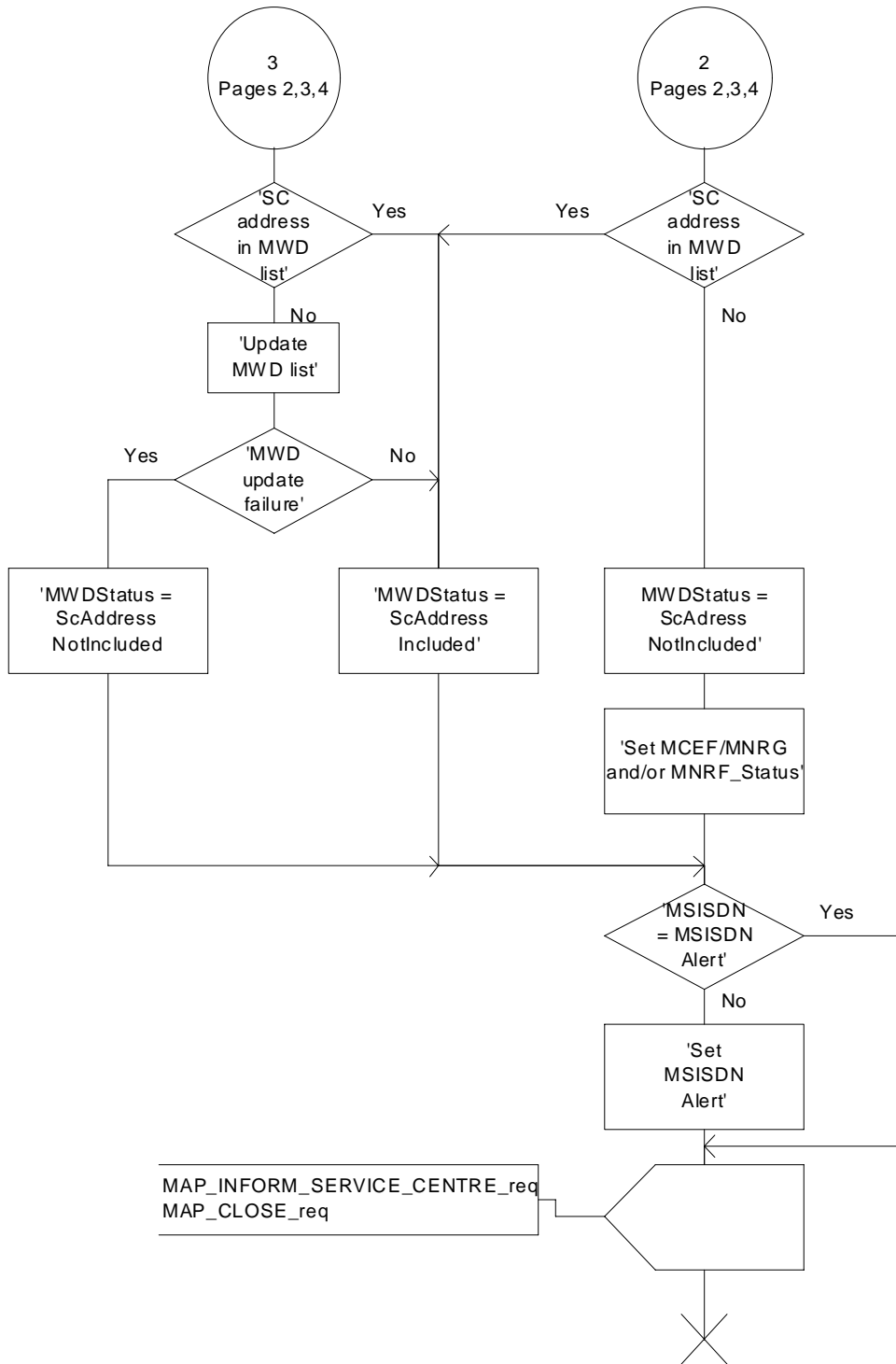


Figure 23.3/6 (sheet 5 of 5): Process Mobile_terminated_SM_HLR

Procedure Select_Transfer_Nodes

23.3_11(1)

Figure 23.3/11: Procedure in the HLR to select the node (MSC or/and SGSN) to which the SMS-GMSC has to send Short Message

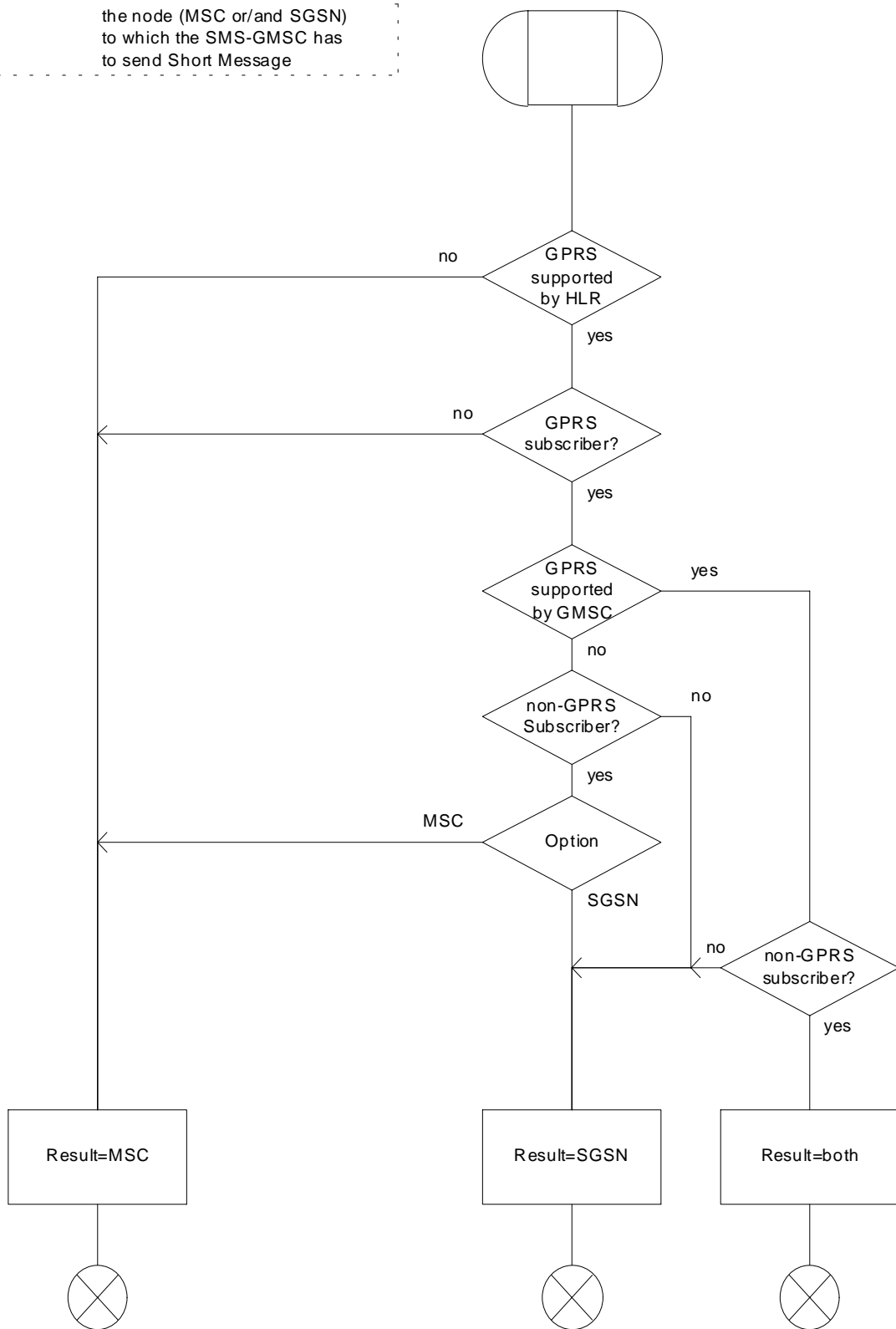


Figure 23.3/11: Procedure Select_Transfer_Nodes

23.3.4 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 and the indication that the SMS-GMSC supports the GPRS functionality.

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 TS GSM 03.40.
- a successful event indication containing following parameters:
 - an IMSI optionally accompanied by an LMSI; and
 - routing addresses (servicing MSC, SGSN or both numbers).

The LMSI shall not be used in case the short message is routed towards the SGSN.

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 , MNRf and MNRg 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 or SGSN.

If both numbers MSC and SGSN are received from HLR as routing information, the SMS-GMSC may choose which path (SGSN or MSC) first the SMS is to be transferred.

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_MT_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_MT_FORWARD_SHORT_MESSAGE request sent to the servicing MSC or SGSN 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 23.3.1 and in the figure 23.3/4.

If the grouping of MAP_OPEN request and MAP_MT_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_MT_FORWARD_SHORT_MESSAGE request is sent.

As a response to the procedure, the GMSC will receive the MAP_MT_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:

In case only one number (MSC or SGSN) was received from HLR as routing information, the mapping of the MAP error causes and the RP_ERROR causes is explained in TS GSM 03.40. The appropriate error indication is sent to the SC.

In case both numbers (MSC and SGSN) were received from HLR as routing information, the transfer of SMS is re-attempted towards the second path only when one of the following errors is received from the unsuccessful transfer over the first path:

Facility Not Supported

Unidentified Subscriber

Absent Subscriber with indication: GPRS or IMSI Detach

Unexpected Data Value

System failure

Data Missing

Subscriber Busy for MT SMS: GPRS Connection Suspended,

otherwise, the mapping of the MAP error causes and the RP_ERROR causes is performed (see TS GSM 03.40) and the appropriate error indication is sent to the SC.

If second forwarding of short message is unsuccessful, the mapping of the MAP error causes and the RP_ERROR causes is explained in TS GSM 03.40. The appropriate error indications are sent to the SC.

If second forwarding of short message is successful, the successful indication is passed 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_SM, an unidentified subscriber or SM delivery failure with error cause MS memory capacity exceeded indication is received from the servicing MSC, SGSN or both, 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.

If absent subscriber diagnostic information (see GSM 03.40) is included with the absent subscriber_SM error indication then this information is relayed to the HLR using the procedure MAP_REPORT_SM_DELIVERY_STATUS.

In case the SMS was attempted to be delivered towards the MSC and the SGSN, and both delivery failed with causes described above, the two unsuccessful SMS delivery outcomes for GPRS and non GPRS are sent to the HLR.

In case the SMS was attempted to be delivered towards the MSC and the SGSN, and the first delivery failed with causes described above and the second delivery succeeded, the unsuccessful and successful SMS delivery outcomes for GPRS and non GPRS are sent to HLR.

The gateway MSC may also invoke the procedure when the first SMS delivery was successful towards MSC, if the MNRF, MCEF flags or both were set in the HLR.

The gateway MSC may also invoke the procedure when the first SMS delivery was successful towards SGSN, if the MNRG, MCEF flags or both were set in the HLR.

This procedure is described in detail in subclause 23.5.

Unexpected data value, system failure 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 TS GSM 03.40 and GSM 04.11.

The unidentified subscriber error is indicated to the SC as absent subscriber with diagnostic information set to 'Unidentified subscriber' as described in TS GSM 03.40.

Note that the indication, on which number belongs the SGSN and MSC, received from the HLR at routing information result (see subclause 23.3.3) will enable the GMSC to map the causes received from the SGSN, MSC or both into the appropriate causes for non GPRS, GPRS or both, and send them to the SC and HLR.

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_MT_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 23.3.1.

The mobile terminated short message transfer procedure in the gateway MSC is shown in figure 23.3/7.

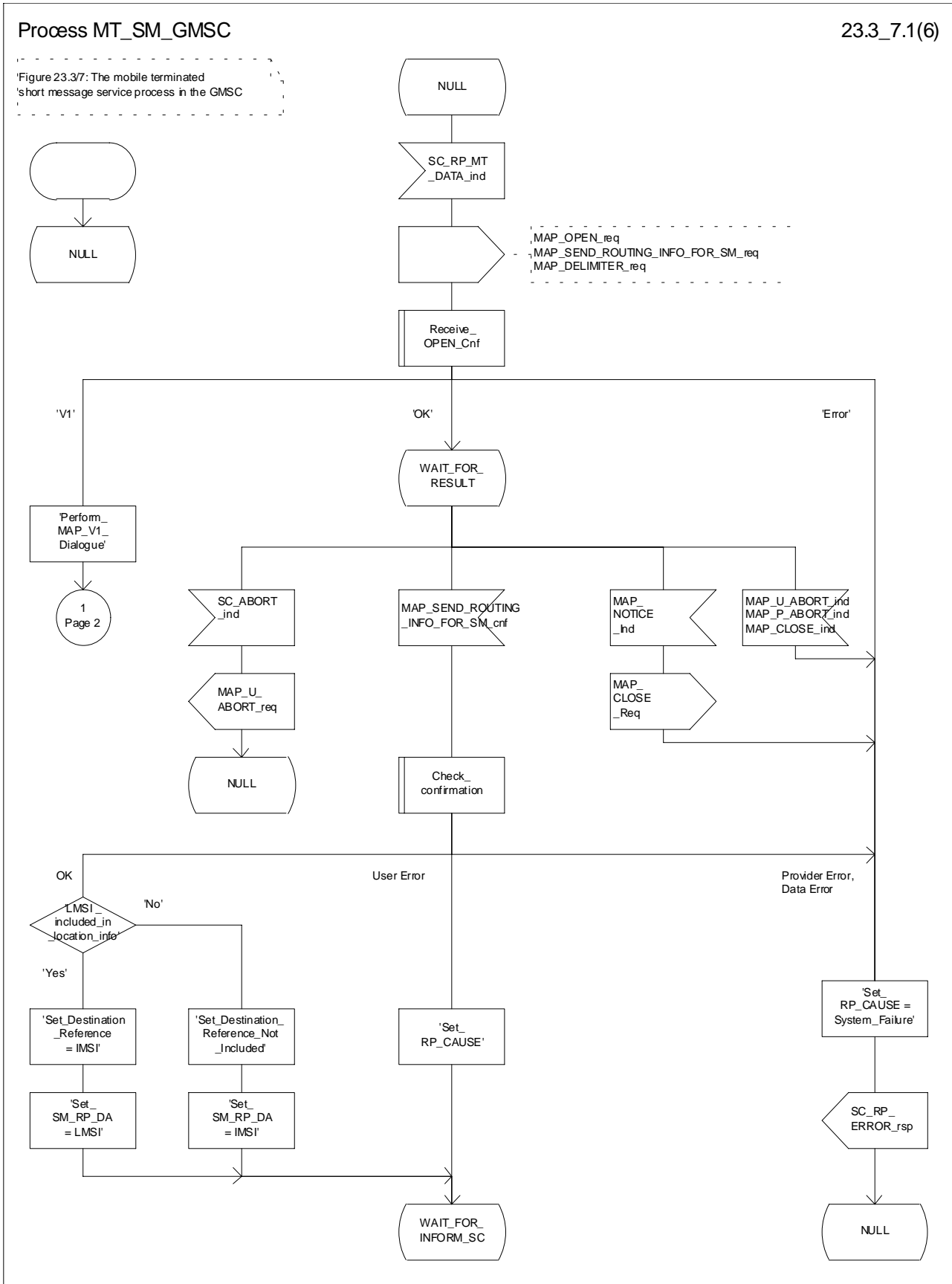


Figure 23.3/7 (sheet 1 of 6): Procedure MT_SM_GMSC

Process MT_SM_GMSC

23.3_7.2(6)

Figure 23.3/7: The mobile terminated short message service process in the GMSC

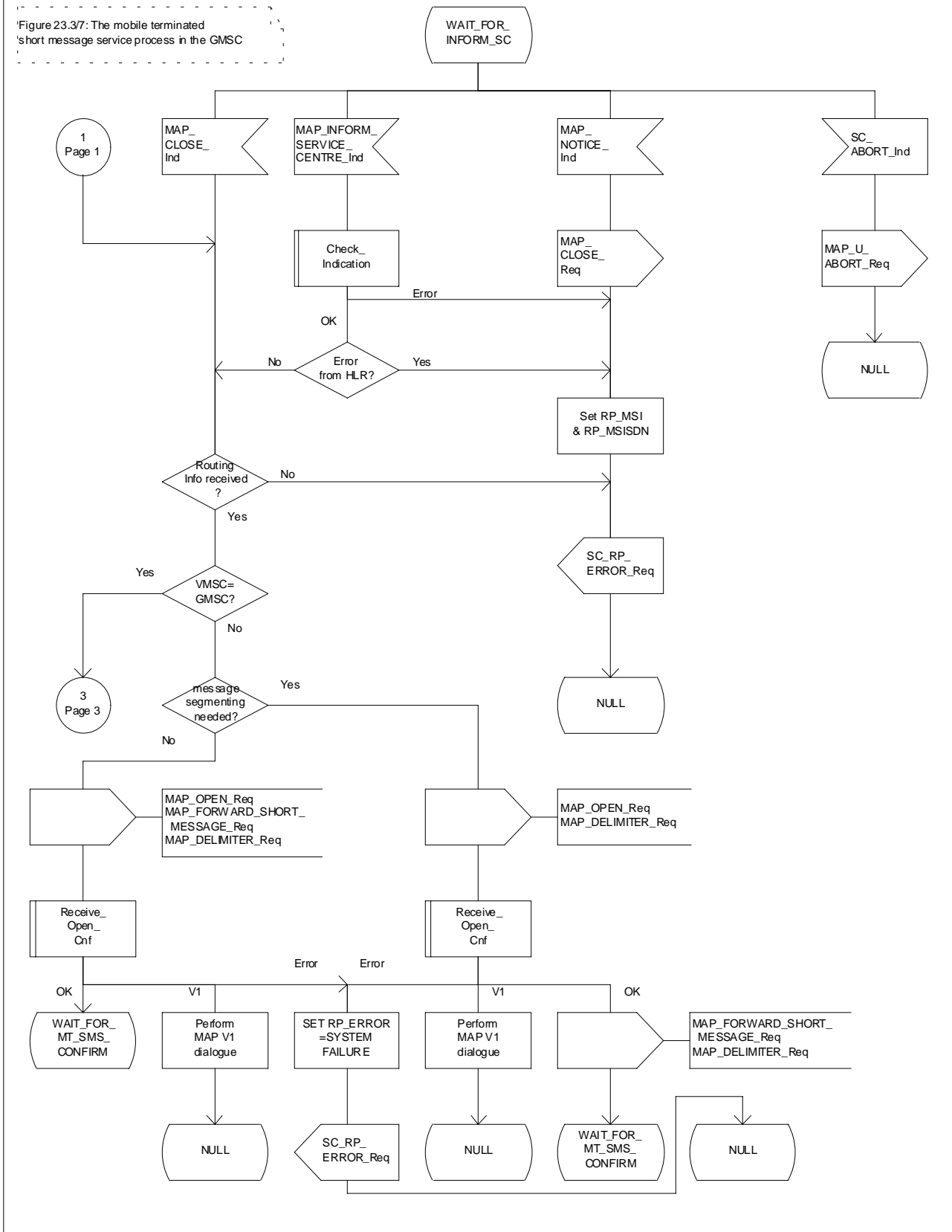


Figure 23.3/7 (sheet 2 to 6): Procedure MT_SM_GMSC

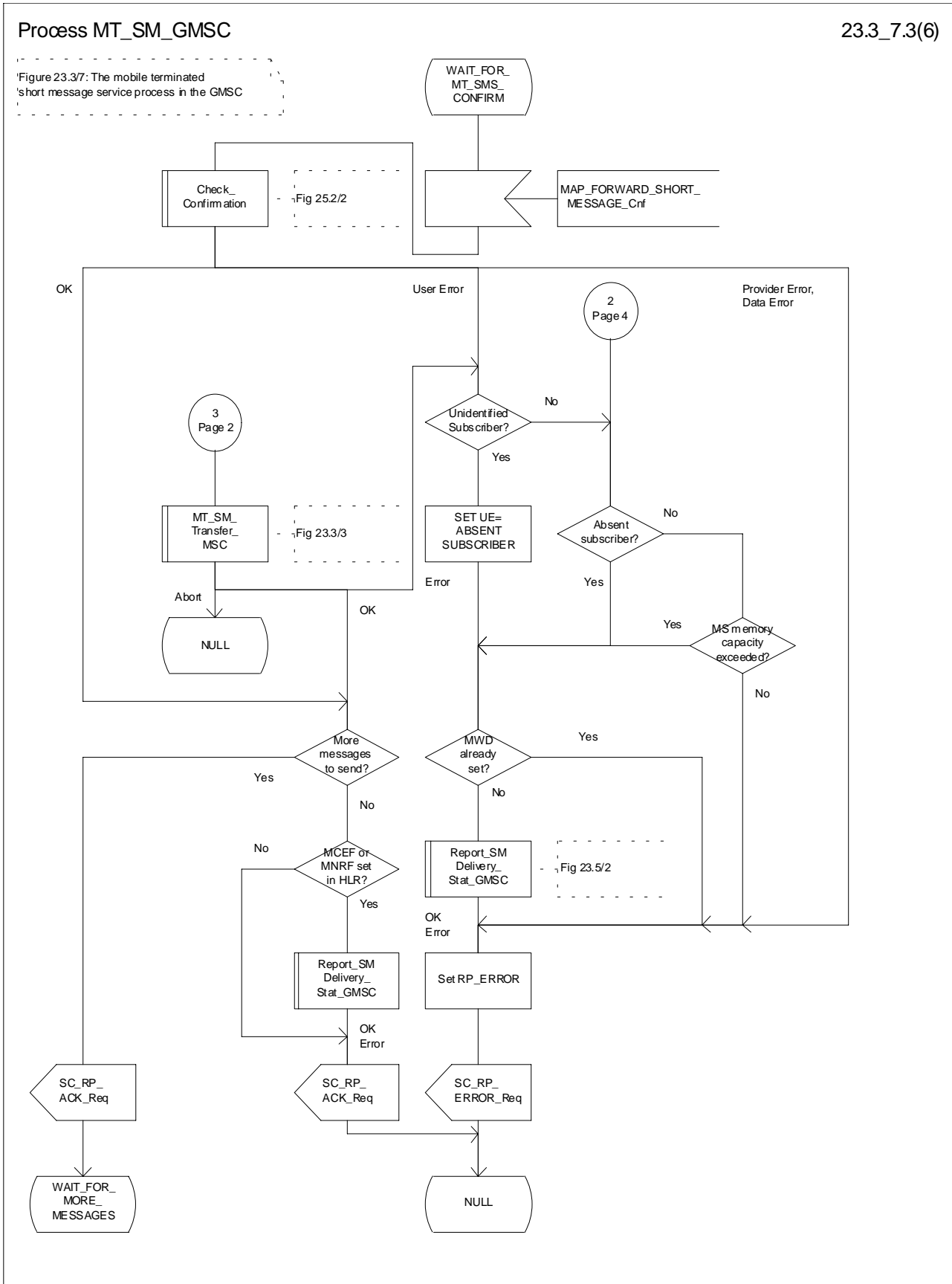


Figure 23.3/7 (sheet 3 of 6): Procedure MT_SM_GMSC

Process MT_SM_GMSC

23.3_7.4(6)

Figure 23.3/7: The mobile terminated short message service process in the GMSC

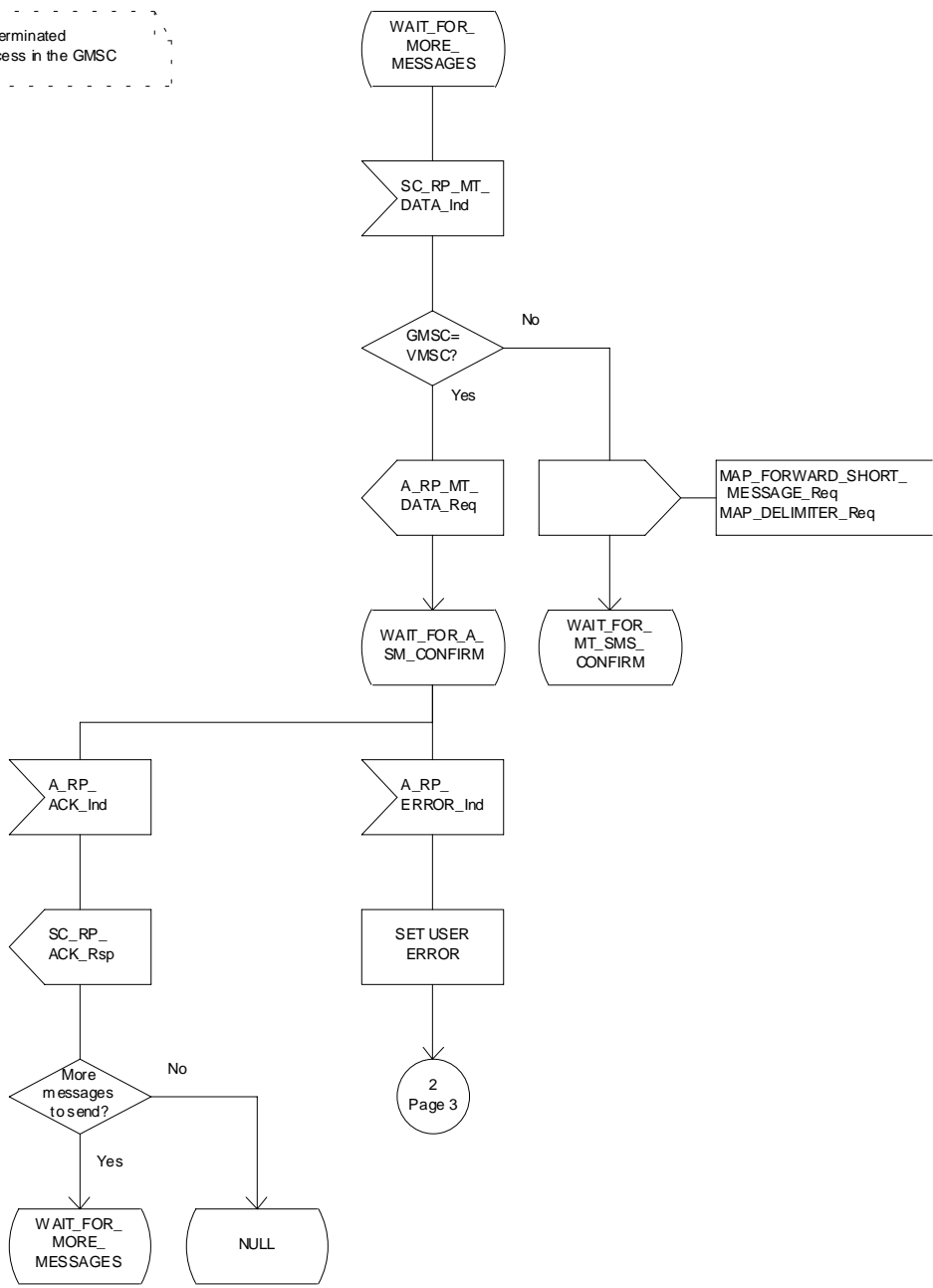


Figure 23.3/7 (sheet 4 of 6): Procedure_MT_SM_GMSC

Process MT_SM_GMSC

23.3_7.5(6)

Figure 23.3/7: The mobile terminated short message service process in the GMSC

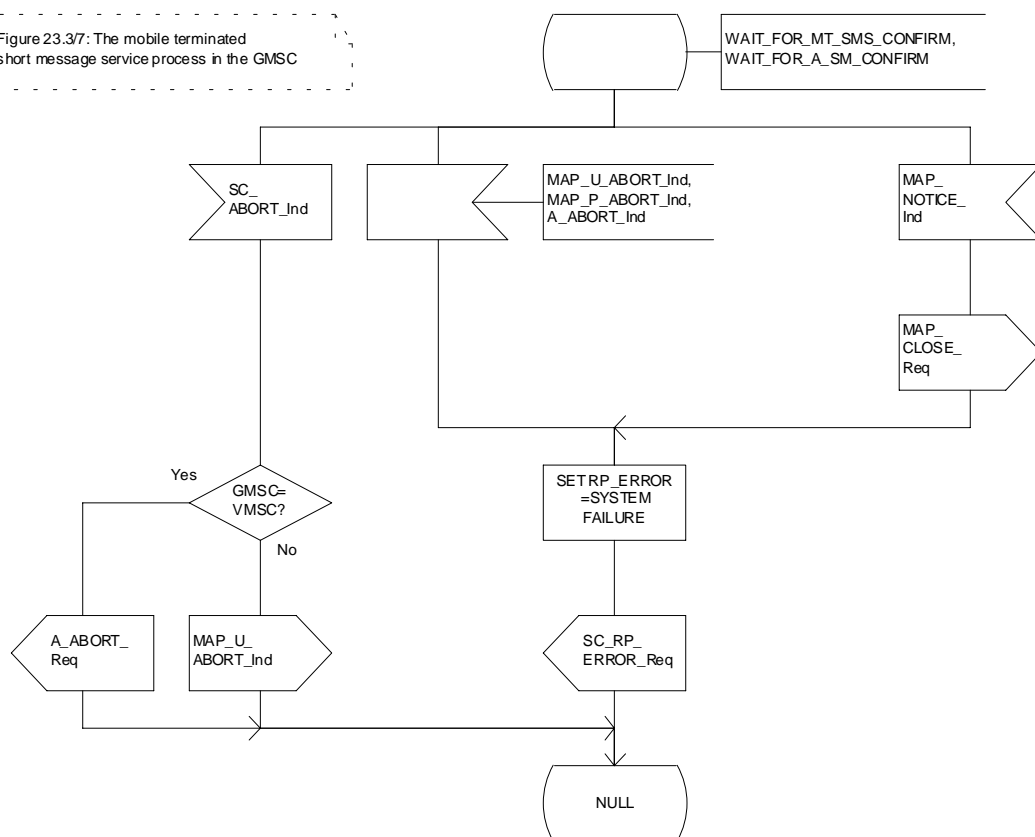


Figure 23.3/7 (sheet 5 to 6): Procedure MT_SM_GMSC

Process MT_SM_GMSC

23.3_7.6(6)

Figure 23.3/7: The mobile terminated short message service process in the GMSC

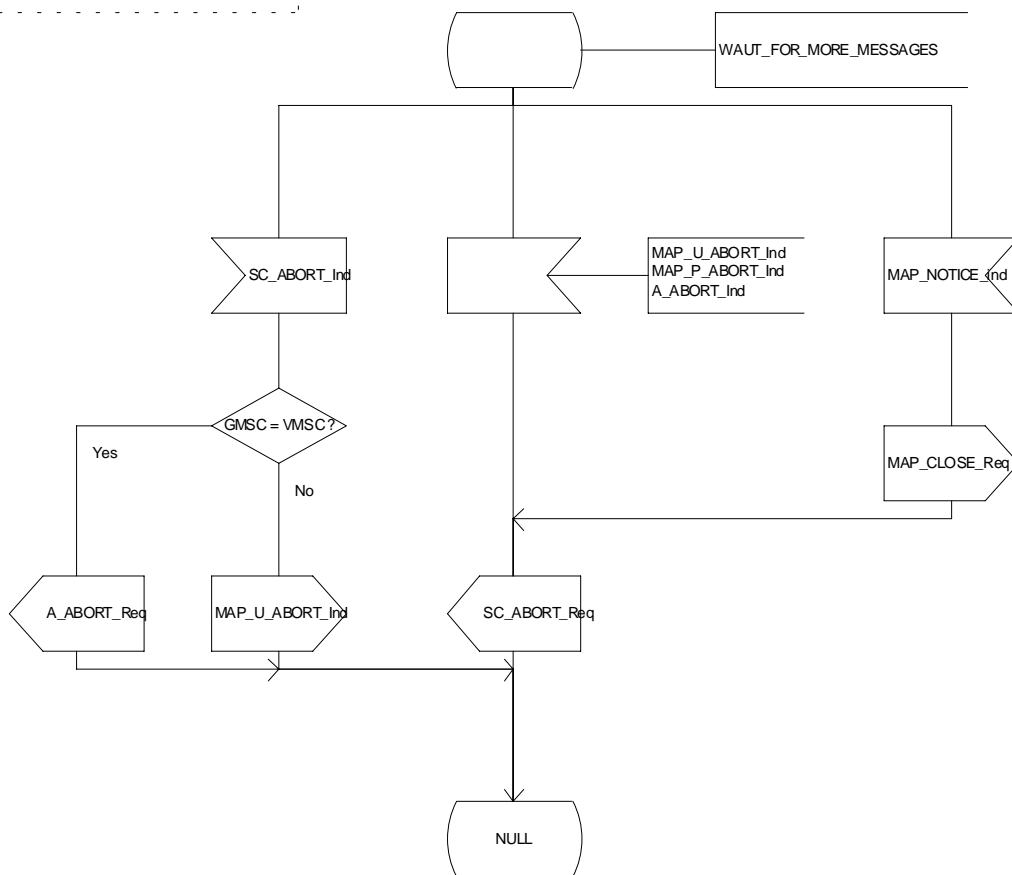


Figure 23.3/7 (sheet 6 of 6): Procedure MT_SM_GMSC

Macrodefinition Check_Subscr_Identity_For_MT_SMS

23.3_8(1)

Figure 23.3/8: Check of the subscriber identity for a mobile terminated short message in the servicing MSC and in the VLR

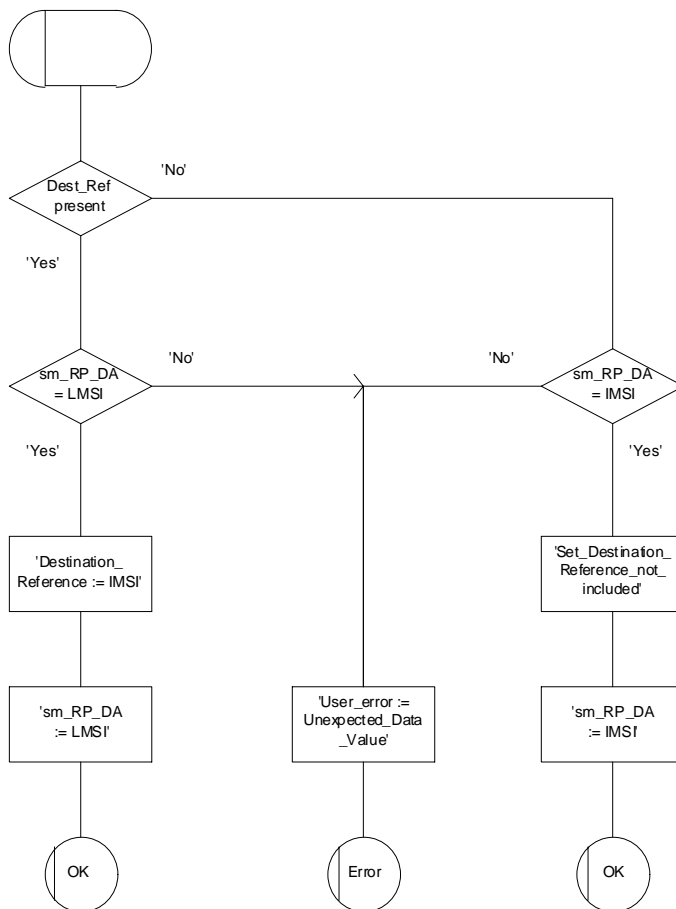


Figure 23.3/8: Macro Check_Subscr_Identity_For_MT_SMS

23.3.5 Procedure in the Servicing SGSN

When initiating the dialogue with the servicing SGSN, the SMS Gateway MSC must provide the IMSI of the subscriber to whom the short message is directed.

The IMSI is included in the sm-RP-DA information field of the MAP_MT_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 SGSN issues a MAP_DELIMITER request primitive in order to trigger the local MAP service-provider to confirm the dialogue.

When receiving the first MAP_MT_FORWARD_SHORT_MESSAGE indication from the gateway MSC, the servicing SGSN performs some subscriber data checks, if the MAP service primitive is accepted and if short message service is supported in the servicing SGSN.

The MAP_MT_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 SGSN does not support the short message service, the service is aborted in the servicing SGSN and the error "Facility Not Supported" is returned to the GMSC.

If the connection is GPRS suspended, the SGSN sends to the GMSC an error specifying that the GPRS connection is suspended.

The subscriber identity information that are included in the MAP service indication primitive is checked by the macro "Check_Subscr_Identity_For_MT_SMS" as follows:

If the IMSI is included in the sm-RP-DA information field of the MAP_MT_FORWARD_SHORT_MESSAGE indication, the MAP_OPEN indication received from the gateway MSC shall not include a Destination Reference.

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 SGSN and the error "Unexpected Data Value" is returned to the GMSC.

The following outcomes from the subscriber data checks can occur in SGSN:

- if the mobile subscriber is unknown, the unidentified subscriber error is forwarded to the GMSC;
- if the "Confirmed by HLR" indicator is set to "Not Confirmed", the unidentified subscriber error is forwarded to the GMSC.
- if the GPRS Detached Flag is set to detached or the LA Not Allowed Flag is set to not allowed in the SGSN, an absent subscriber error with the diagnostic indication set to 'GPRS Detached' is forwarded to the GMSC and the MS not reachable for GPRS (MNRG) flag is set;
- If the location area identification is known and the "Confirmed by Radio Contact" indicator is set to "Confirmed", the paging procedure is invoked (see subclause 25.3). Otherwise the search procedure is invoked (see subclause 25.3).

The result of the paging or the search procedure is processed as follows:

- if the procedure is completed successfully, the SGSN may trigger the Authentication, Ciphering and IMEI check procedures (see subclauses 25.4 and 25.5). Then, if the procedure are completed successfully, the SGSN will send the short message to the MS;
- if the procedure is completed successfully, but the MS has no mobile terminated short message transfer capability, the SM delivery failure indication with cause "equipment not SM equipped" is provided to the GMSC;
- if the procedure is ended unsuccessfully because of subscriber already busy for SMS, another paging, emergency call, location updating, inter SGSN routing area update or a call set-up, the subscriber busy for MT SMS is provided to the GMSC.

- if the procedure is ended unsuccessfully, the absent subscriber_SM error is forwarded to the GMSC with the absent subscriber diagnostic indication set to 'No Paging Response for GPRS', but if the location area is unknown, the system failure indication is provided to the GMSC.

If forwarding of the short message is initiated, the SGSN awaits the result before one of the following responses is sent back to the GMSC:

- an acknowledgement 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 7.6.1.4) 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_MT_FORWARD_SHORT_MESSAGE ends unsuccessfully, the transaction to the gateway MSC is terminated. Otherwise, the servicing SGSN waits for the next short message from the Service Centre.

When receiving the next MAP_MT_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 SGSN 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 SGSN 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 mobile terminated short message transfer procedure in the servicing SGSN is shown in figures 23.3/9 and 23.3/10. The page and search procedures are shown in figure 25.3/1 and 25.3/2.

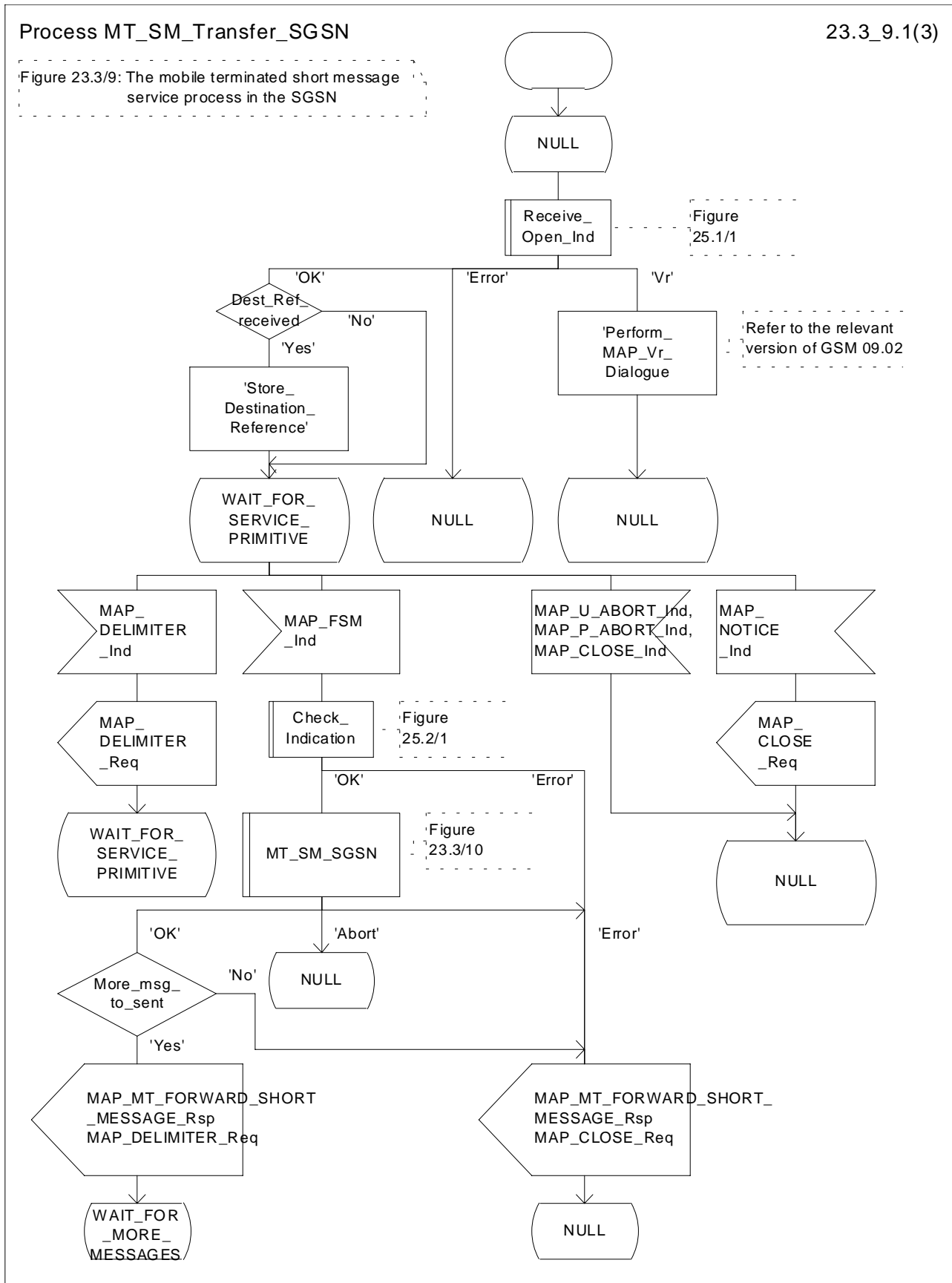


Figure 23.3/9 (sheet 1 of 3): Procedure MT_SM_Transfer_SGSN

Process MT_SM_Transfer_SGSN

23.3_9.2(3)

Figure 23.3/9: The mobile terminated short message service process in the SGSN

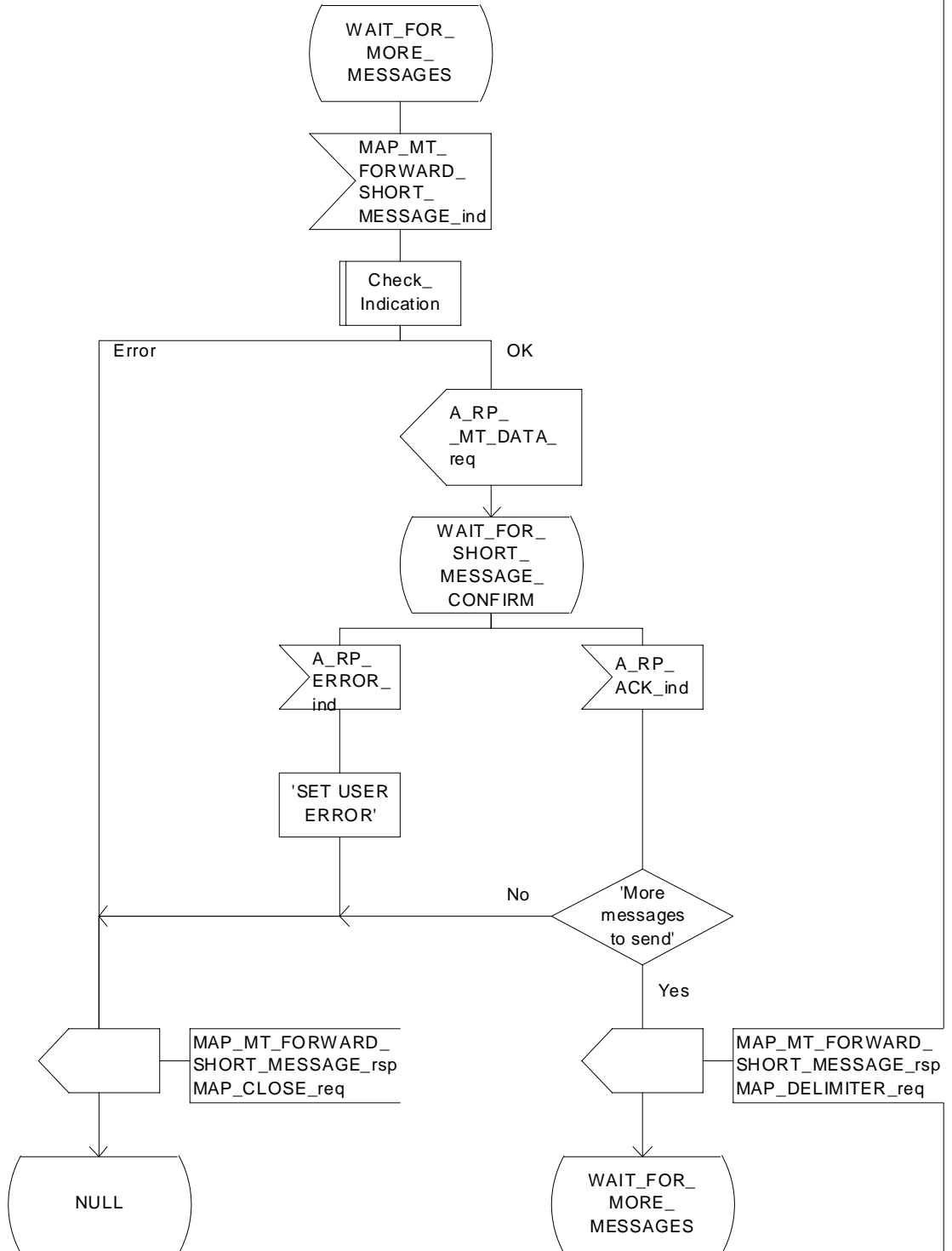


Figure 23.3/9 (sheet 2 of 3): Procedure MT_SM_Transfer_SGSN

Process MT_SM_Transfer_SGSN

23.3_9.3(3)

Figure 23.3/9: The mobile terminated short message service process in the SGSN

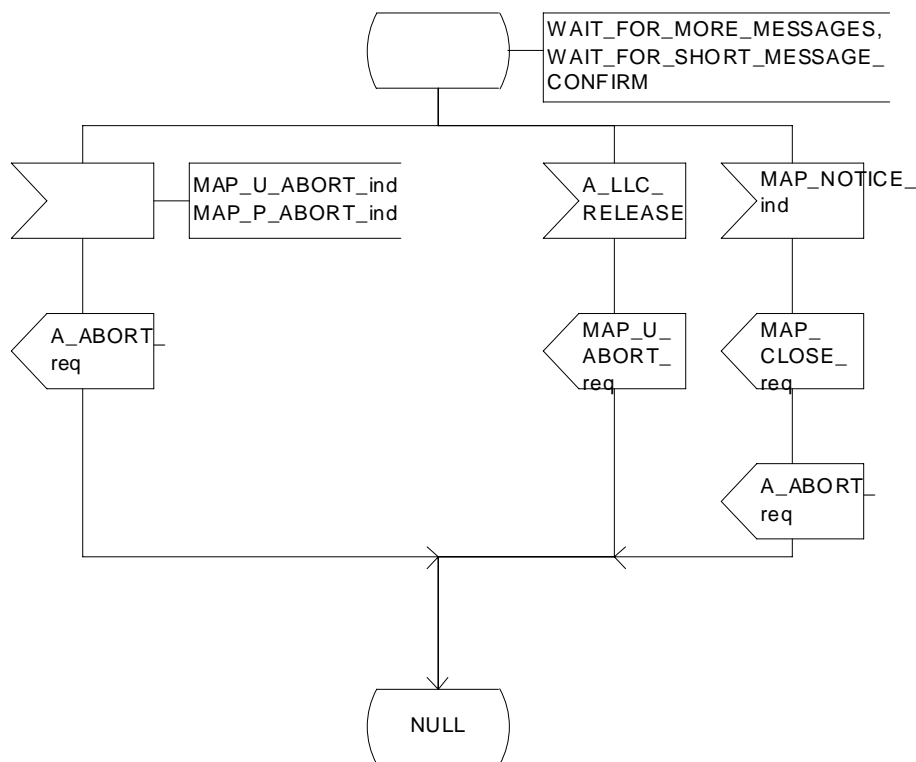


Figure 23.3/9 (sheet 3 of 3): Procedure MT_SM_Transfer_SGSN

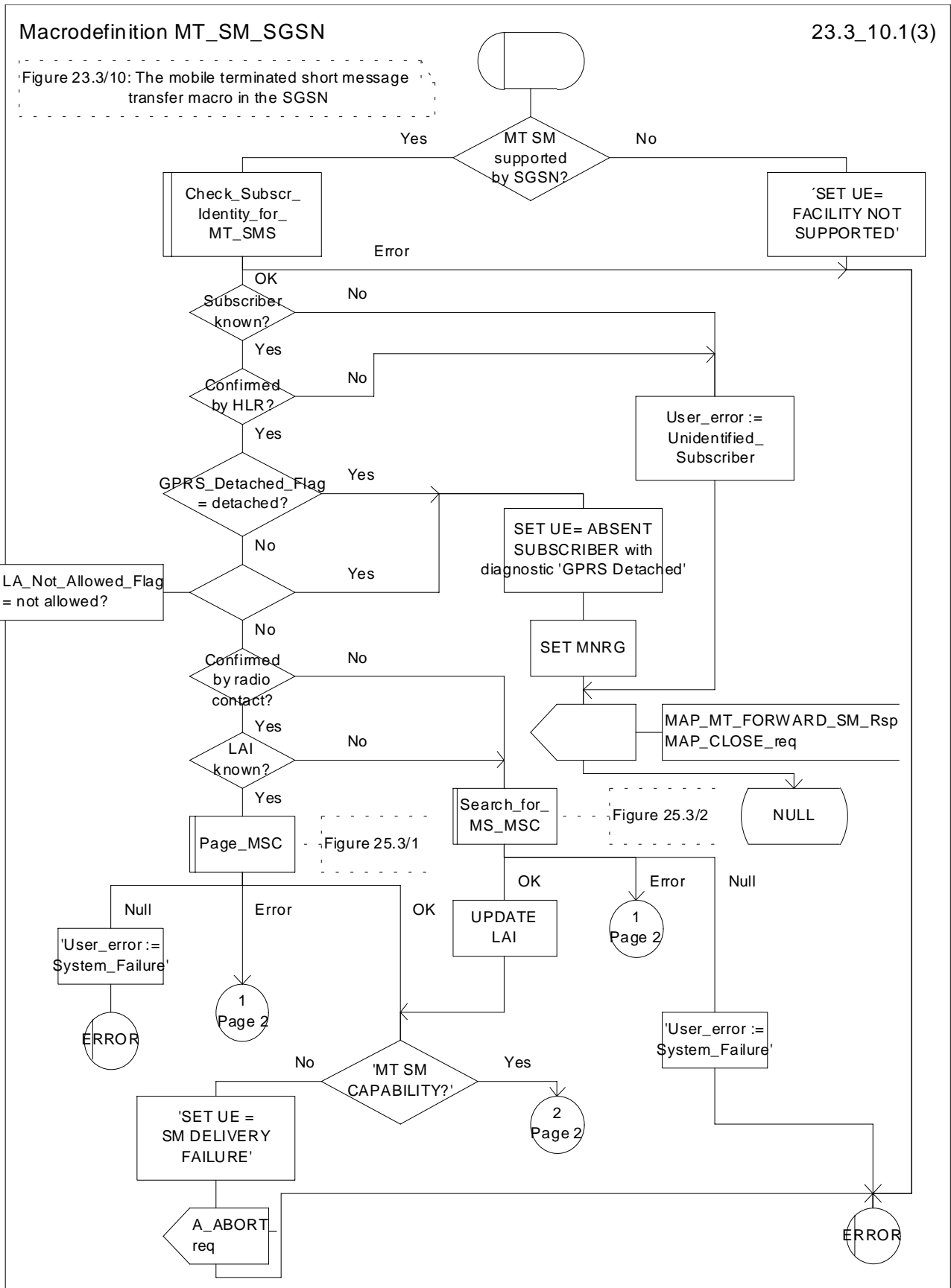


Figure 23.3/10 (sheet 1 of 3): Macro MT_SM_SGSN

Macrodefinition MT_SM_SGSN

23.3_10.2(3)

Figure 23.3/10: The mobile terminated short message transfer macro in the SGSN

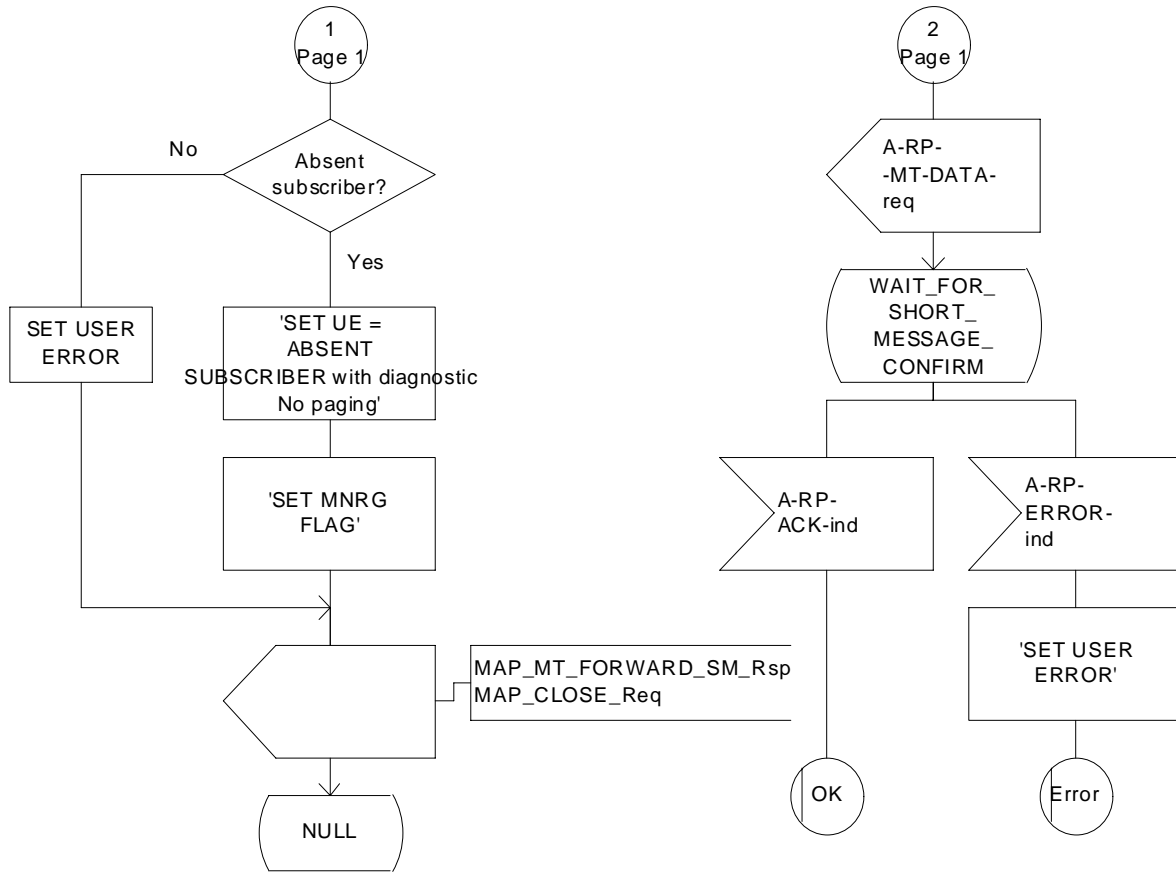


Figure 23.3/10 (sheet 2 of 3): Macro MT_SM_SGSN

Macrodefinition MT_SM_SGSN

23.3_10.3(3)

Figure 23.3/10: The mobile terminated short message transfer macro in the SGSN

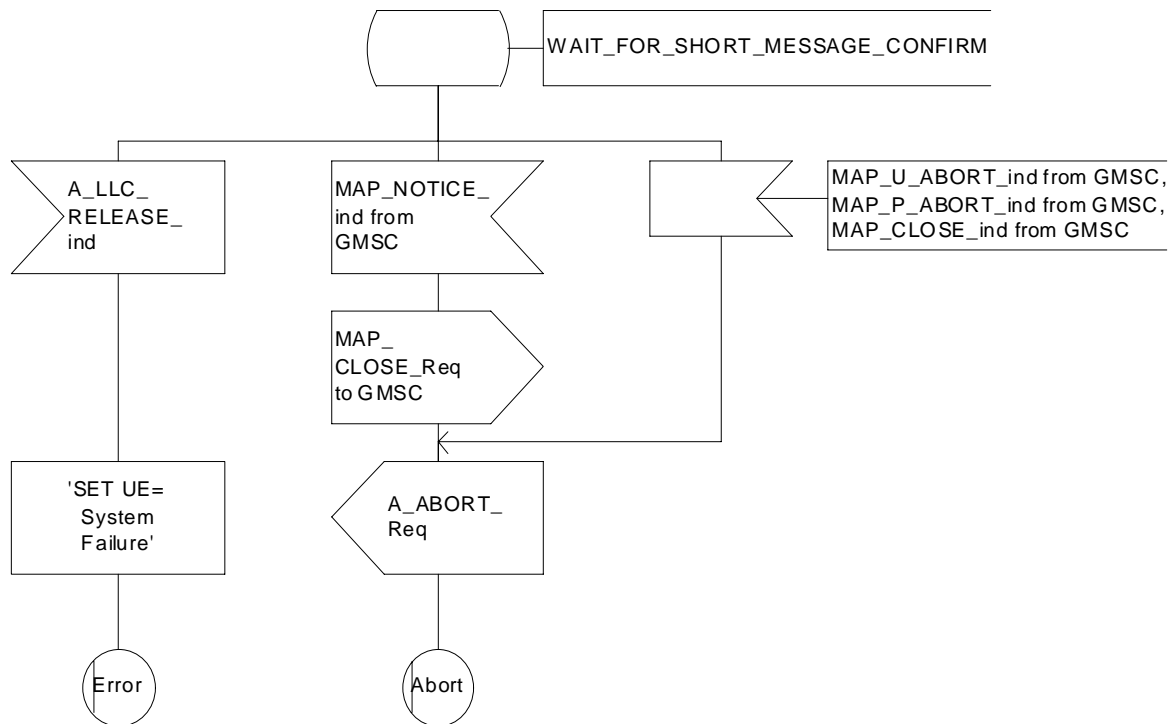
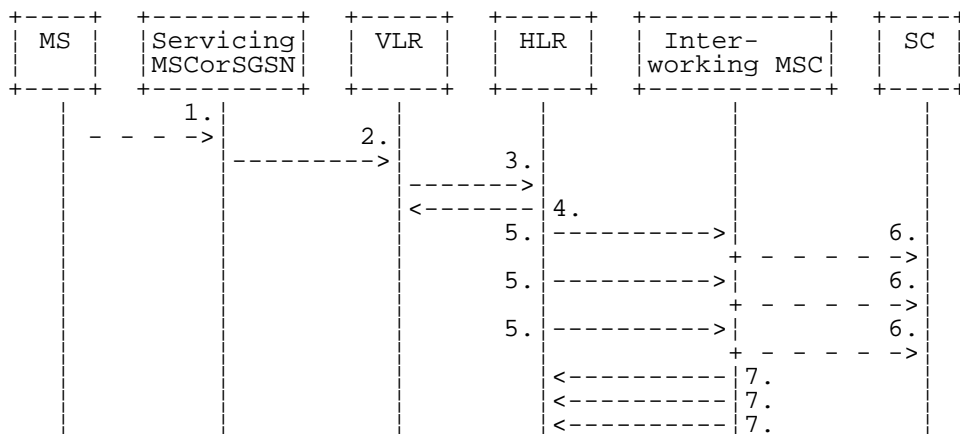


Figure 23.3/10 (sheet 3 of 3): Macro MT_SM_SGSN

23.4 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 23.4/1.



- 1) CM Service Request (**), Page response or Location Updating (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.

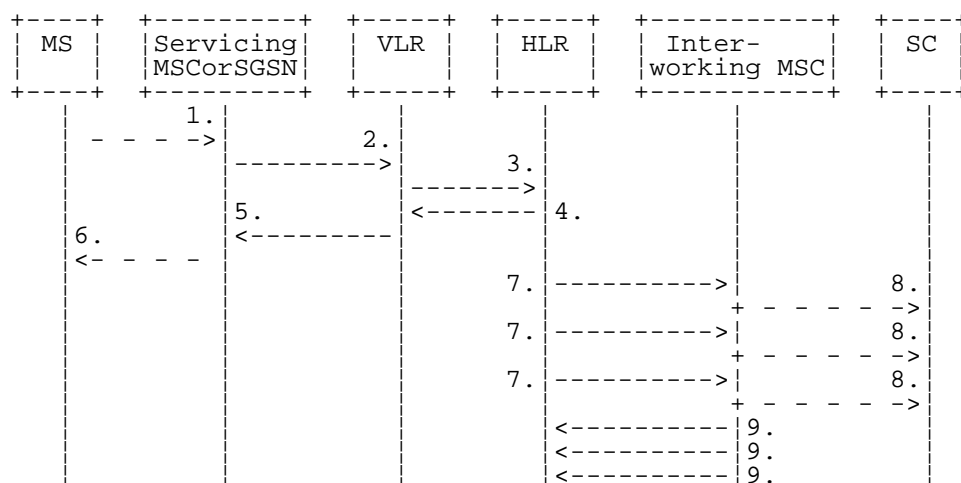
NOTE 2: The HLR initiates the MAP_ALERT_SERVICE_CENTRE service only if the MS Memory Capacity Exceeded flag is clear.

(*) In case of GPRS, messages 3) and 4) are sent/received by SGSN

(**) Those messages are not used by SGSN

Figure 23.4/1: 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 23.4/2.



- 1) SM memory capacity available (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) (GSM 04.11)
- 7) MAP_ALERT_SERVICE_CENTRE (note 1)
- 8) Alert Service Centre (GSM 03.40)
- 9) MAP_ALERT_SERVICE_CENTRE_ACK

NOTE 1: To all Service Centres in the Message Waiting List.

- (*) Message 2) and 5) are not used by SGSN
- (**) In the case of GPRS messages 3) and 4) are sent/received by SGSN

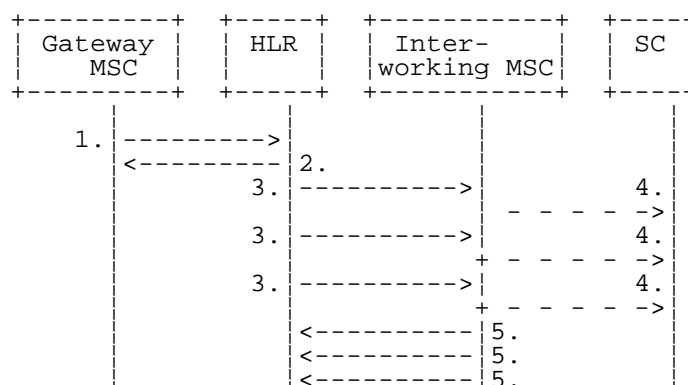
Figure 23.4/2: Short message alert procedure (MS memory capacity available)

In addition the following MAP services are used in the MS memory available case:

- MAP_PROCESS_ACCESS_REQUEST (see subclause 8.3); (*)
- MAP_AUTHENTICATE (see subclause 8.5); (*)
- MAP_SET_CIPHERING_MODE (see subclause 8.6); (*)
- MAP_PROVIDE_IMSI (see subclause 8.9); (*)
- MAP_CHECK_IMEI (see subclause 8.7);
- MAP_FORWARD_NEW_TMSI (see subclause 8.9); (*)
- MAP_TRACE_SUBSCRIBER_ACTIVITY (see subclause 9.1). (*)

(*) Those messages are not used by SGSN.

The Short Message Alert procedure when the MS indicates successful transfer after polling is shown in figure 23.4/3.



- 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 23.4/3: Short message alert procedure (Successful transfer after polling)

23.4.1 Procedures in the Servicing MSC

The activation of the MAP_PROCESS_ACCESS_REQUEST service is described in the subclause 23.6.2.

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 23.4/4.

Process SM_Alert_MSC

23.4_4(1)

Figure 23.4/4: The short message alert process in the servicing MSC for MS memory capacity available.

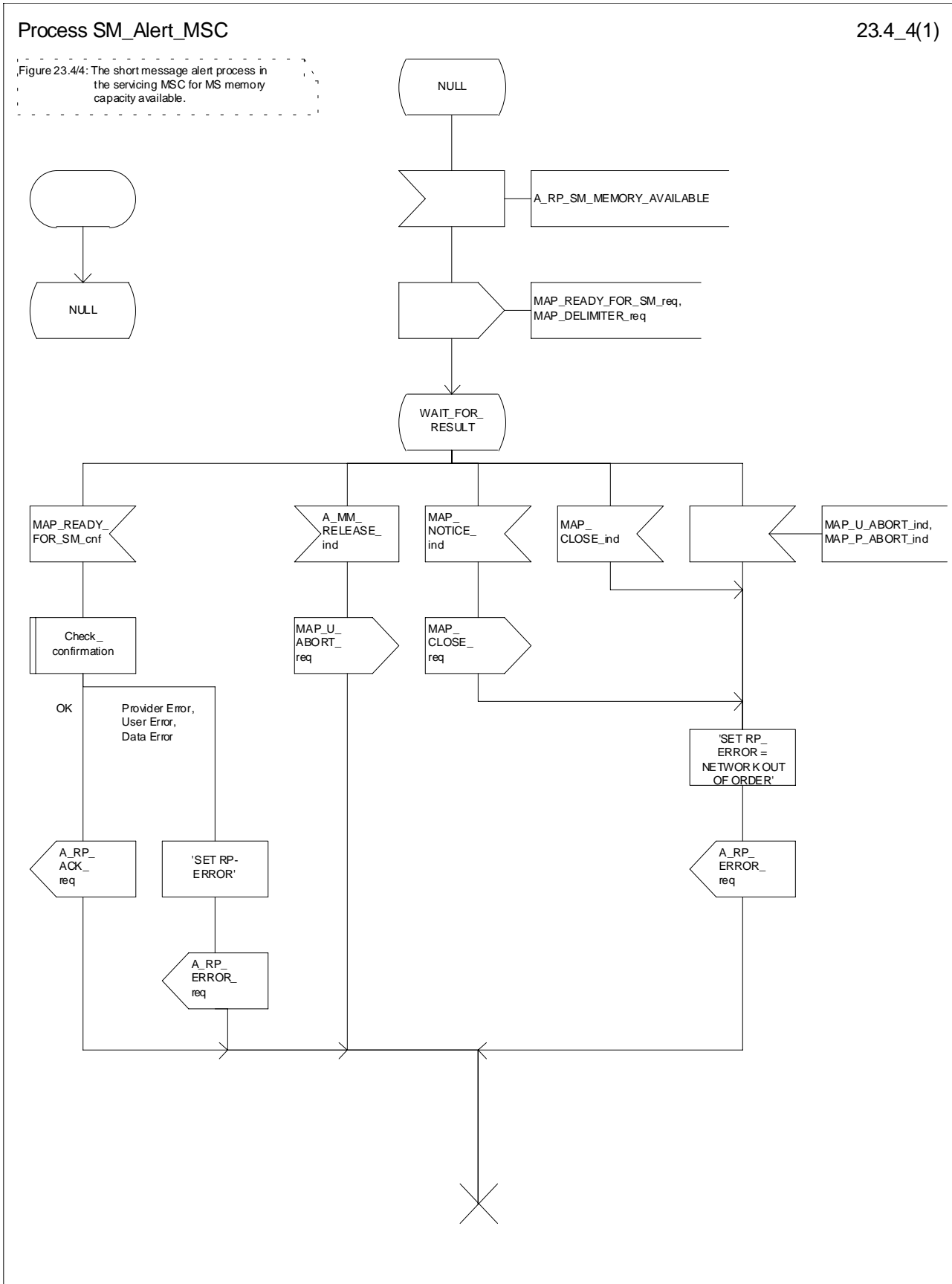


Figure 23.4/4: Procedure SM_Alert_MSC

23.4.2 Procedures in the VLR

23.4.2.1 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 for non GPRS. 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 25.10.

23.4.2.2 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 Vr only;
 - procedure failure is reported as a system failure error.

The short message alert procedure in the VLR is shown in figures 23.4/5.

Process SM_Alert_VLR

23.4_5(1)

Figure 23.4/5: The short messages alert process in the VLR for MS memory capacity available

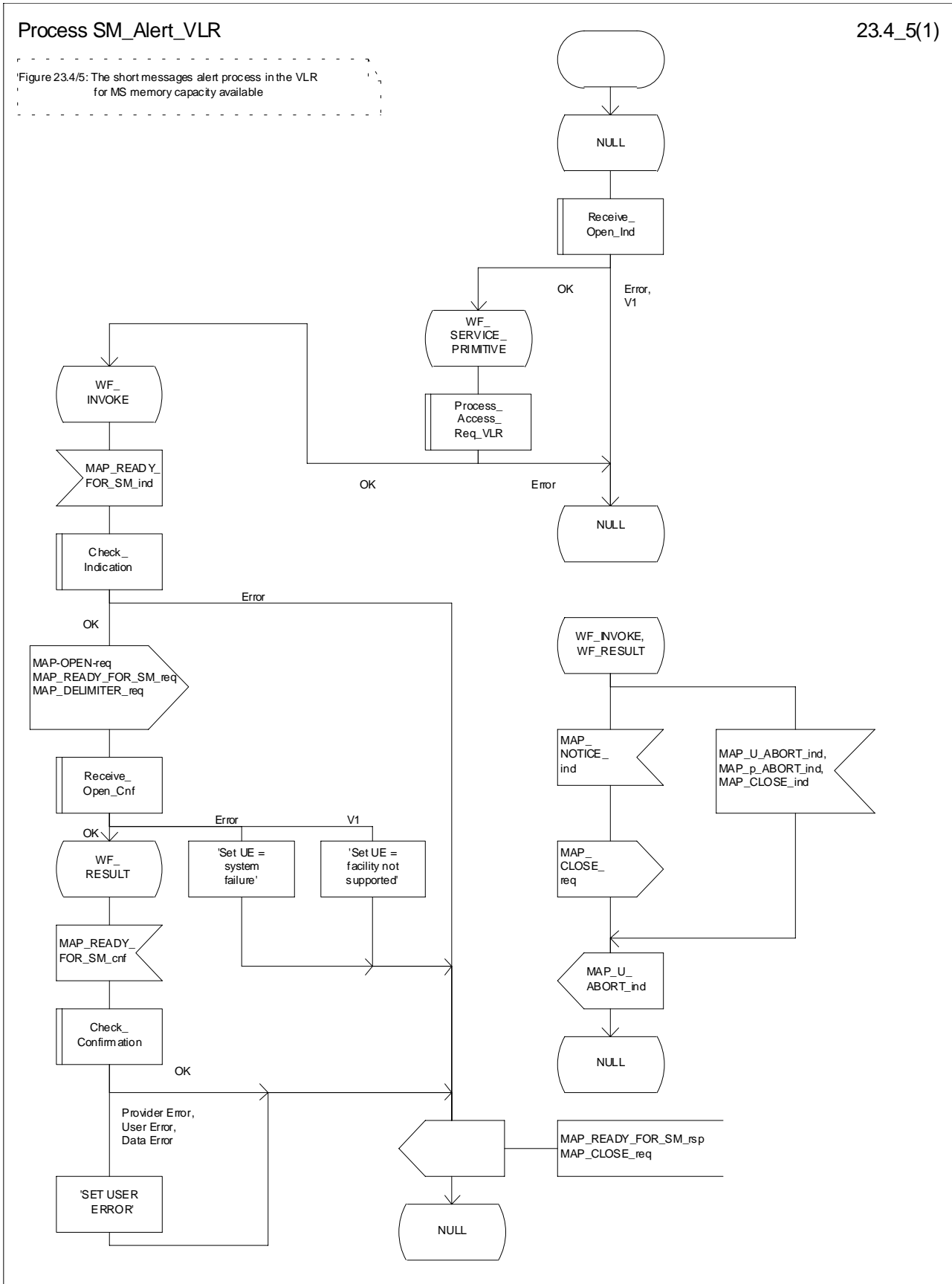


Figure 23.4/5: Procedure SM_Alert_VLR

23.4.3 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 or MNRG, MCEF, and MWD a facility not supported error is reported to the VLR or SGSN. If the IMSI is unknown an unknown subscriber error is reported to the VLR or SGSN. Otherwise an acknowledgement is returned to the VLR or SGSN.

If neither the MS not reachable flag (MNRF) or the MS not reachable for GPRS (MNRG) flag, nor the memory capacity exceeded flag (MCEF) are set, and MAP_READY_FOR_SM is received from the VLR or SGSN, 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 23.6) for the same subscriber can be carried out when delayed in the GMSC.

If the Alert Reason indicates the mobile present for non GPRS situation, or when the update location procedure has been successfully completed or Supplementary Service Control request is received, the MS not reachable flag (MNRF) 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 and stored reason for absence for non GPRS are cleared but the alert procedure is not started.

If the Alert Reason indicates the mobile present for GPRS situation, or when the Update GPRSlocation procedure has been successfully completed, the MS not reachable for GPRS (MNRG) flag is cleared and the service centre alert procedure is initiated. If the memory capacity exceeded flag is set, the MS detach for GPRS flag is cleared and stored reason for absence for GPRS are cleared but the alert procedure is not started.

If the Alert Reason indicates the memory available for non GPRS situation, the HLR initiates the alert procedure. The MS not reachable and memory capacity available flags are cleared.

If the Alert Reason indicates the memory available for GPRS situation, the HLR initiates the alert procedure. The MS detach for GPRS 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 for non GPRS, the HLR initiates the alert procedure described in the subclause 25.10 and clears MCEF and MNRF flags and stored reason for absence for non GPRS are cleared.

If the MAP_REPORT_SM_DELIVERY_STATUS indication is received and it indicates the successful transfer of the mobile terminated short message for GPRS, the HLR initiates the alert procedure described in the subclause 25.10 and clears MCEF and MNRG flags and stored reason for absence for GPRS are cleared.

The short message alert procedure in the HLR is shown in figures 23.4/6 and 25.10/2.

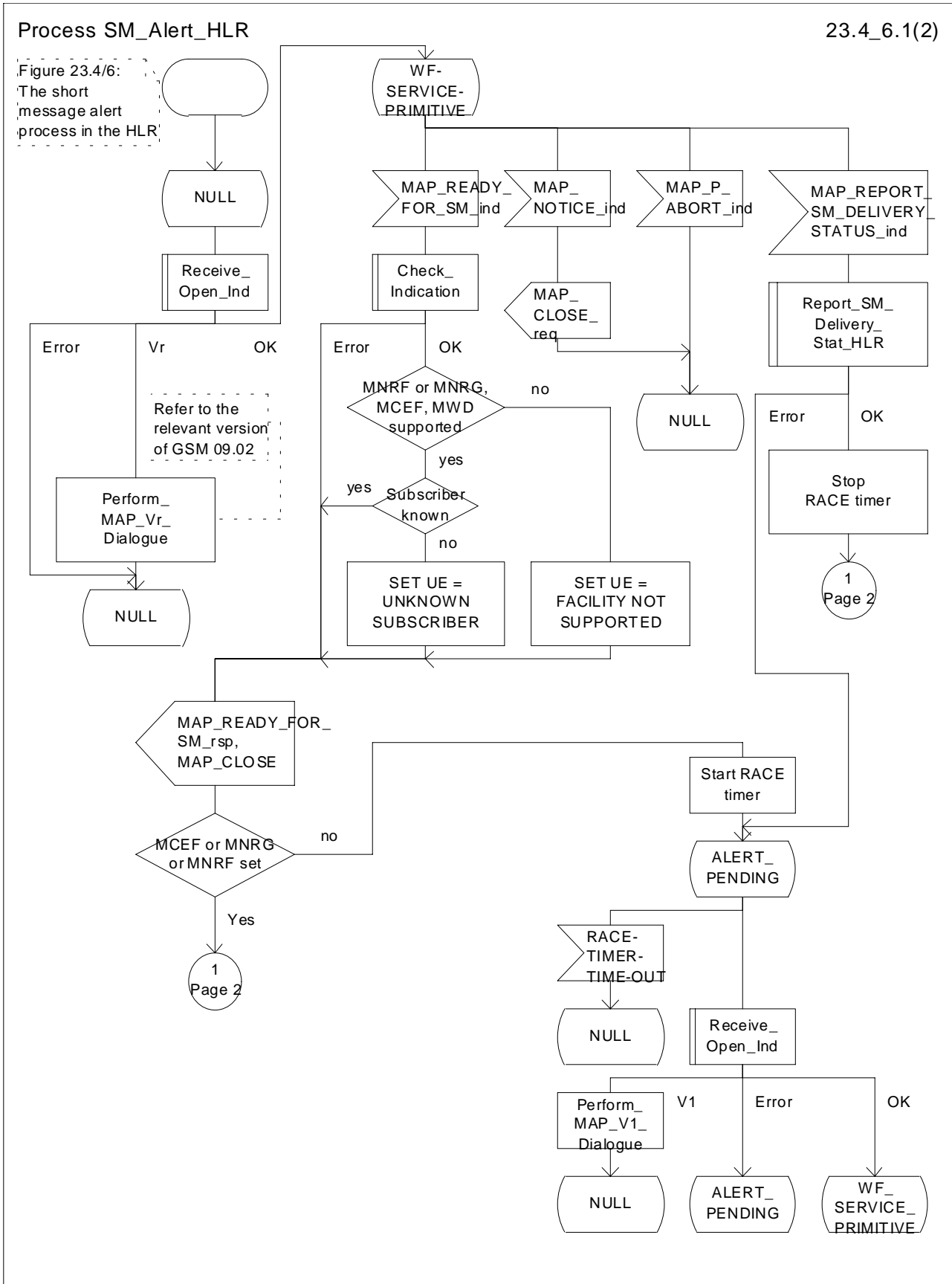


Figure 23.4/6 (sheet 1 of 2): Process SM_Alert_HLR

Process SM_Alert_HLR

23.4_6.2(2)

Figure 23.4/6:
The short
message alert
process in the HLR

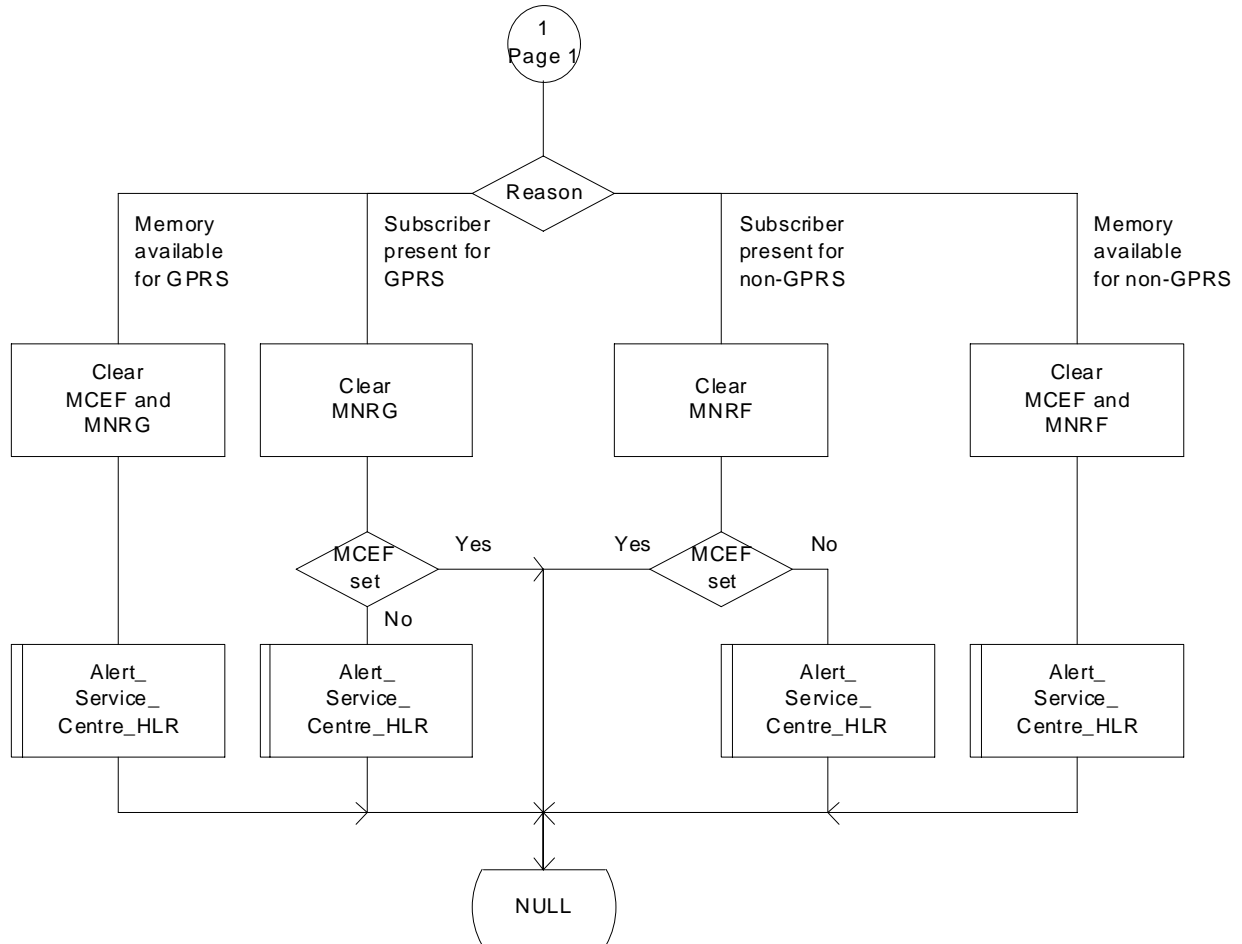


Figure 23.4/6 (sheet 2 of 2): Process SM_Alert_HLR

23.4.4 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 23.4/7.

Process Alert_SC_IWMSC

23.4_7(1)

Figure 23.4/7: The short message alert message in the IWMSC

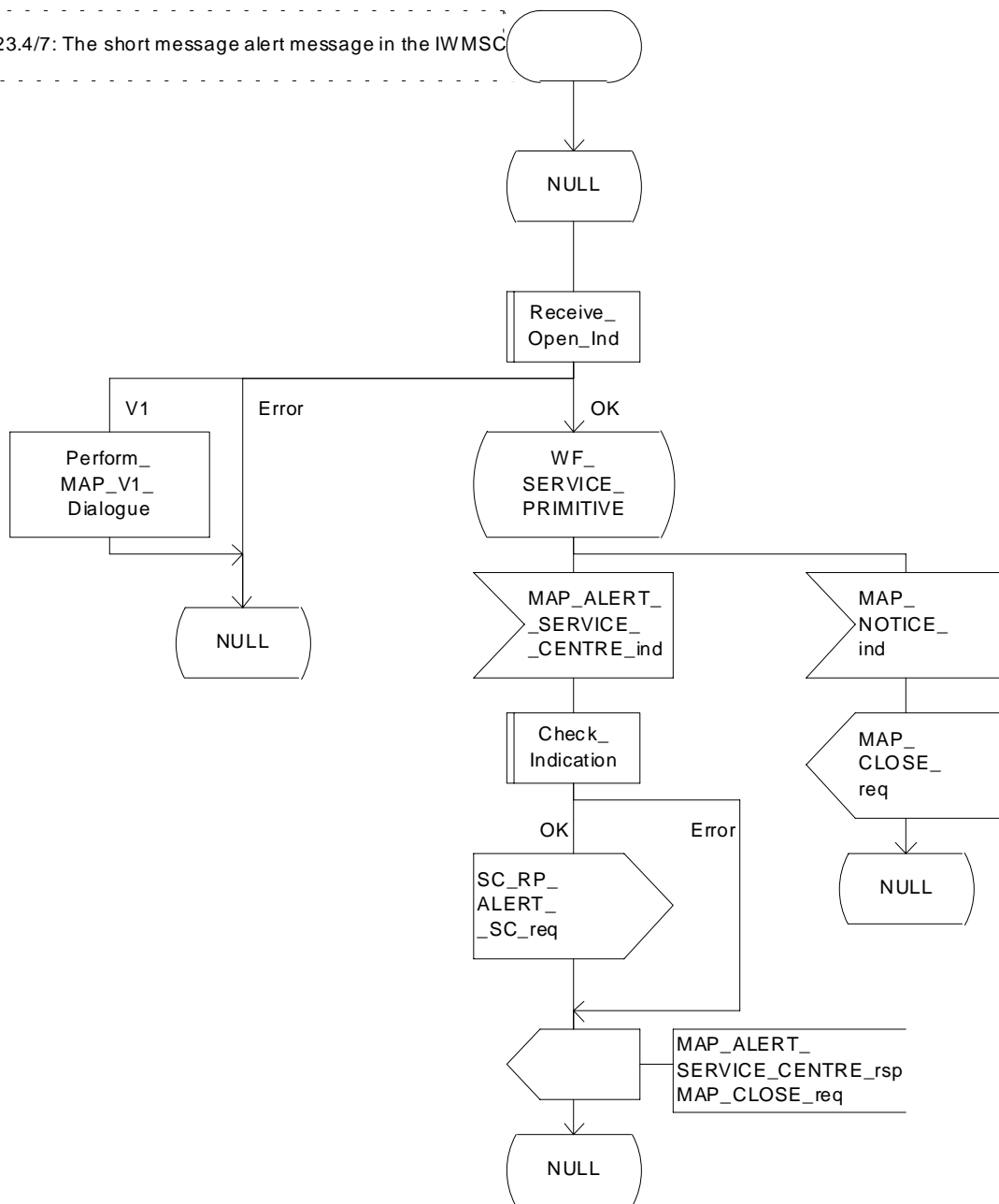


Figure 23.4/7: Process Alert_SC_IWMSC

23.4.5 Procedures in the Servicing SGSN

23.4.5.1 The Mobile Subscriber is present

When receiving Page response, Attach request or Routing area update request messages (TS GSM 04.08), while the MS not reachable for GPRS (MNRG) flag is set, the SGSN will send the MAP_READY_FOR_SM request towards the HLR. The Alert Reason is set to indicate that the mobile subscriber is present for GPRS.

When receiving the answer, the SGSN will act as follows:

- MNRG is cleared if the procedure is successful
- MNRG is not cleared if the procedure is not successful

The process in SGSN is described in detail in the subclause 25.10/3.

23.4.5.2 The Mobile Equipment has memory available

After receiving the SM memory capacity available indication, the servicing SGSN sends the MAP_READY_FOR_SM request to the HLR indicating memory available for GPRS. The outcome of that procedure is one of the following:

- successful acknowledgment. The SGSN 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 SGSN for the MS memory capacity available case is shown in figure 23.4/8.

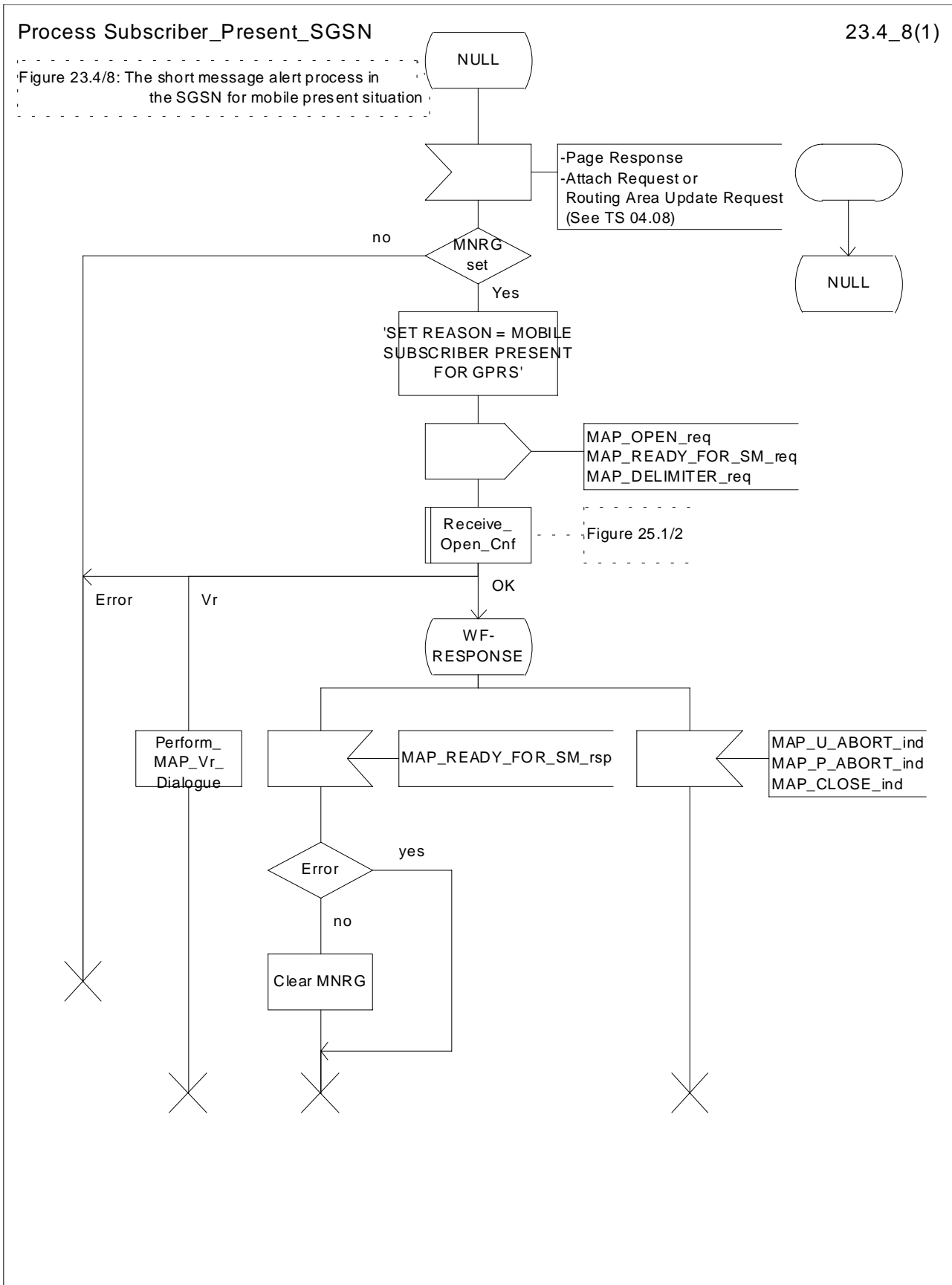


Figure 23.4/8: Process Subscriber_Present_SGSN

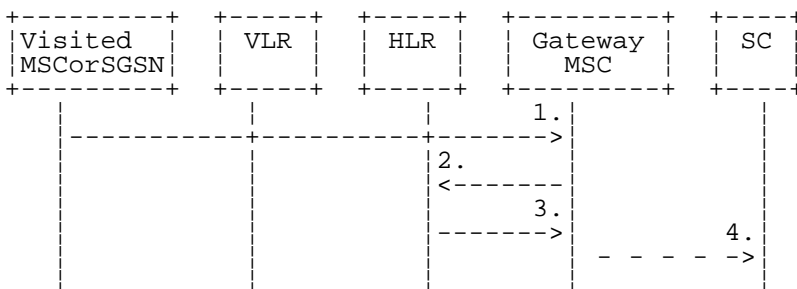
23.5 The SM delivery status report procedure

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
- and/or the MS not reachable flag for non GPRS in the case of unidentified or absent subscriber
- and/or the MS not reachable for GPRS flag in the case of unidentified or absent subscriber for GPRS

Additionally the procedure is used to report the HLR about the successful transfer for GPRS or non GPRS after the Service Centre has polled the subscriber. This procedure is described also in the subclause 23.4.

The SM delivery status report procedure is shown in figure 23.5/1.



- 1) MAP_MT_FORWARD_SHORT_MESSAGE_ACK/_NACK (Absent subscriber_SM, 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 23.5/1: Short message delivery status report procedure

23.5.1 Procedures in the HLR

When the HLR receives a MAP_REPORT_SM_DELIVERY_STATUS indication, it acts as described in the subclause 23.6, macro Report_SM_Delivery_Stat_HLR.

The short message delivery status report process in the HLR is shown in figure 23.5/2.

Process SM_Delivery_Status_Report_HLR

23.5_2(1)

Figure 23.5/2: The report SM delivery process in the HLR

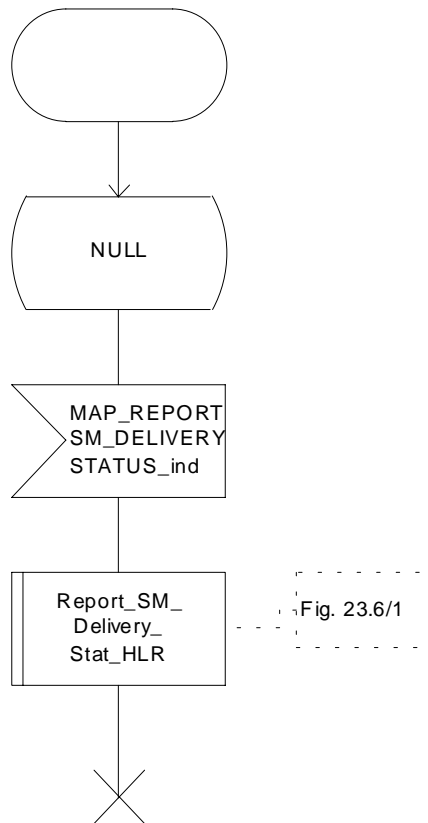


Figure 23.5/2: Process SM_Delivery_Status_Report_HLR

23.5.2 Procedures in the gateway MSC

The GMSC invokes the short message delivery status report procedure if an absent subscriber_SM indication, unidentified subscriber indication, SM delivery failure error indicating MS memory capacity exceeded or both are received from the servicing MSC, SGSN or both 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_SM indication

In case of successful SMS delivery on the second path, the successful SMS Delivery outcome is sent in combination with the unsuccessful SMS Delivery outcome to the HLR.

The service is invoked also when the HLR has indicated that either of the flags MCEF, MNRF or both are set and the first SM delivery was successful from the servicing MSC or, in case of subsequent SM, the last SM delivery was successful from the servicing MSC.

The service is invoked also when the HLR has indicated that either of the flags MCEF, MNRF or both are set and the SM delivery was successful from the servicing SGSN or, in case of subsequent SM, the last SM delivery was successful from the servicing SGSN.

The reason for unsuccessful, successful for GPRS, non GPRS or both deliveries of the short message are included in the SM Delivery Outcome in the MAP_REPORT_SM_DELIVERY_STATUS request. In the case of an unsuccessful delivery due to the subscriber being absent the absent subscriber diagnostic indication (if available) is also included in the MAP_REPORT_SM_DELIVERY_STATUS request.

If the reason for unsuccessful delivery is absent subscriber with diagnostic 'Paging failure' for GPRS or non GPRS, the two SM Delivery Outcomes absent subscriber with both diagnostics 'Paging failure' for GPRS and non GPRS is included 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. In that case, if the SM Delivery Outcome was successful SMS delivery for GPRS or non GPRS (combined or not with another unsuccessful reason), a successful report is forwarded to the SC.

A provider error is indicated as a system failure to the SC.

Note that the indication, on which number belongs the SGSN and MSC, received from the HLR at routing information result (see subclause 23.3.3) will enable the GMSC to map the causes received from the SGSN, MSC or both into the appropriate causes for GPRS, non GPRS or both, and send them to the SC and HLR.

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 23.5/3.

Macrodefinition Report_SM_Delivery_Stat_GMSC

23.5_3(1)

Figure 23.5/3: The report SM delivery status macro in the GMSC

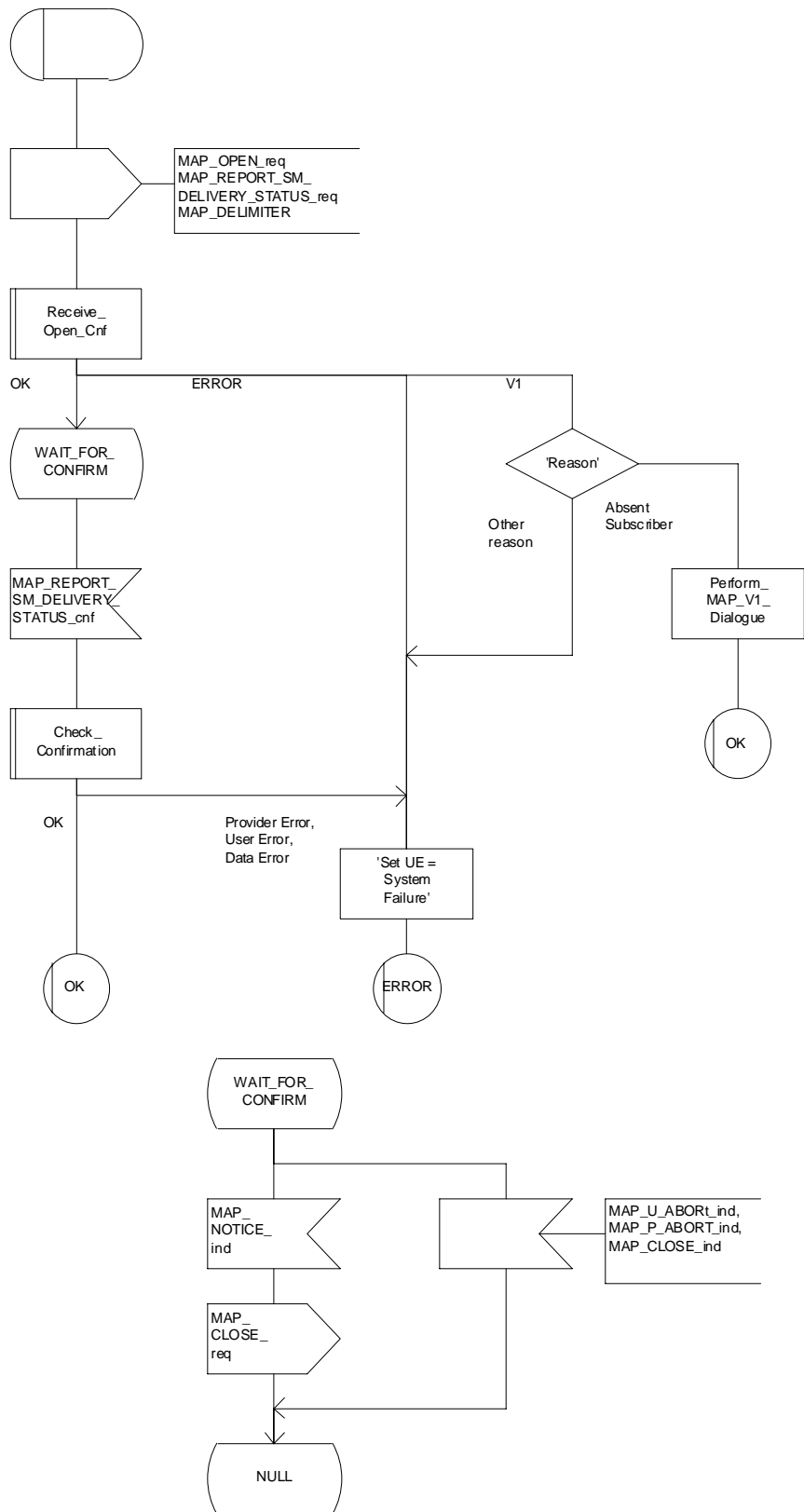


Figure 23.5/3: Macro Report_SM_Delivery_Stat_GMSC

23.6 Common procedures for the short message clause

23.6.1 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 responds to the indication as follows:

- if the flag « GPRS Support Indicator » is absent then if the subscriber is a GPRS subscriber and a non-GPRS subscriber with the option « transfer of SM via the SGSN when GPRS is not supported in the GMSC » or if the subscriber is a GPRS subscriber only, the HLR shall interpret the delivery outcome as a GPRS delivery outcome.
- 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 25.10;
- 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;

The SC address is only stored in the MWD if the unsuccessful SM Delivery Outcome is not received in combination with another successful SM Delivery Outcome

- if the SM Delivery Outcome is MS memory capacity exceeded for non GPRS, the HLR sets the memory capacity exceeded flag in the subscriber data and resets the MNRF;
- if the SM Delivery Outcome is MS memory capacity exceeded for GPRS the HLR sets the memory capacity exceeded flag in the subscriber data and resets the MNRG;
- if the SM Delivery Outcome is absent subscriber for non GPRS, the HLR sets the mobile station not reachable flag in the subscriber data. If a reason for absence is provided by the GMSC then this is stored in the mobile station not reachable reason (MNRR) in the subscriber data.
- if the SM Delivery Outcome is absent subscriber for GPRS, the HLR sets the mobile station not reachable for GPRS flag in the subscriber data. If a reason for absence is provided by the GMSC then this is stored in the mobile station not reachable reason (MNRR) in the subscriber data.

Note that a combination of all the SM Delivery Outcome specified above may be provided to the HLR from the SMS-GMSC.

The short message delivery status report macro in the HLR is shown in figure 23.6/1.

Macrodefinition Report_SM_Delivery_Stat_HLR

23.6_1(1)

Figure 23.6/1: The report SM delivery status macro in the HLR

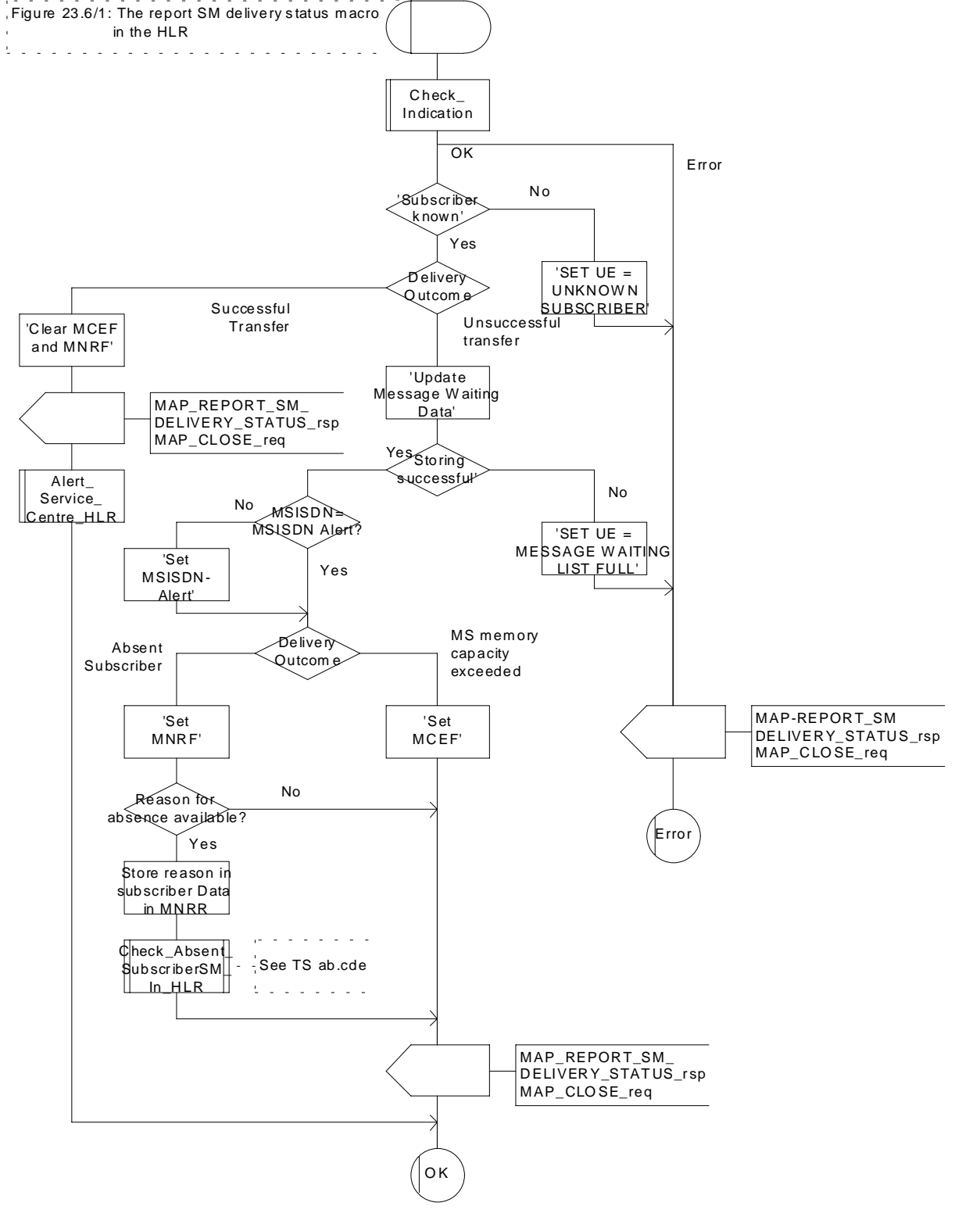


Figure 23.6/1: Macro Report_SM_Delivery_Stat_HLR

24 GPRS process description

24.1 General

The MAP GPRS procedures are used for the Network Requested PDP-Context Activation procedures.

The stage 2 specification for General Packet Radio Service (GPRS) is in GSM 03.60 [100].

24.1.1 Process in the HLR for Send Routing Information for GPRS

The MAP process in the HLR to provide routing information for a network-requested PDP context activation is shown in figure 24.1/1. The MAP process invokes a macro not defined in this subclause; the definition of this macro can be found as follows:

Receive_Open_Ind	see subclause 25.1.1;
Check_Indication	see subclause 25.2.1.

Successful outcome

When the MAP process receives a MAP_OPEN indication with the application context gprsLocationInfoRetrieval, it checks it by invoking the macro Receive_Open_Ind.

If the macro takes the OK exit, the MAP process waits for a service indication.

If a MAP_SEND_ROUTING_INFO_FOR_GPRS service indication is received, the HLR sends a Send Routing Info For Gprs request to the GPRS application process in the HLR, and wait for a response. The Send Routing Info For Gprs request contains the parameter received in the MAP_SEND_ROUTING_INFO_FOR_GPRS service indication

If the GPRS application process in the HLR returns a positive response containing the routing information, the MAP process constructs a MAP_SEND_ROUTING_INFO_FOR_GPRS service response containing the routing info, constructs a MAP_CLOSE service request, sends them to the GGSN and returns to the idle state.

Negative response from HLR GPRS application process

If the GPRS application process in the HLR returns a negative response, the MAP process constructs a MAP_SEND_ROUTING_INFO_FOR_GPRS service response containing the appropriate error, constructs a MAP_CLOSE service request, sends them to the GGSN and returns to the idle state.

Failure of dialogue opening with the GGSN

If the macro Receive_Open_Ind takes the Vr exit or the Error exit, the MAP process returns to the idle state.

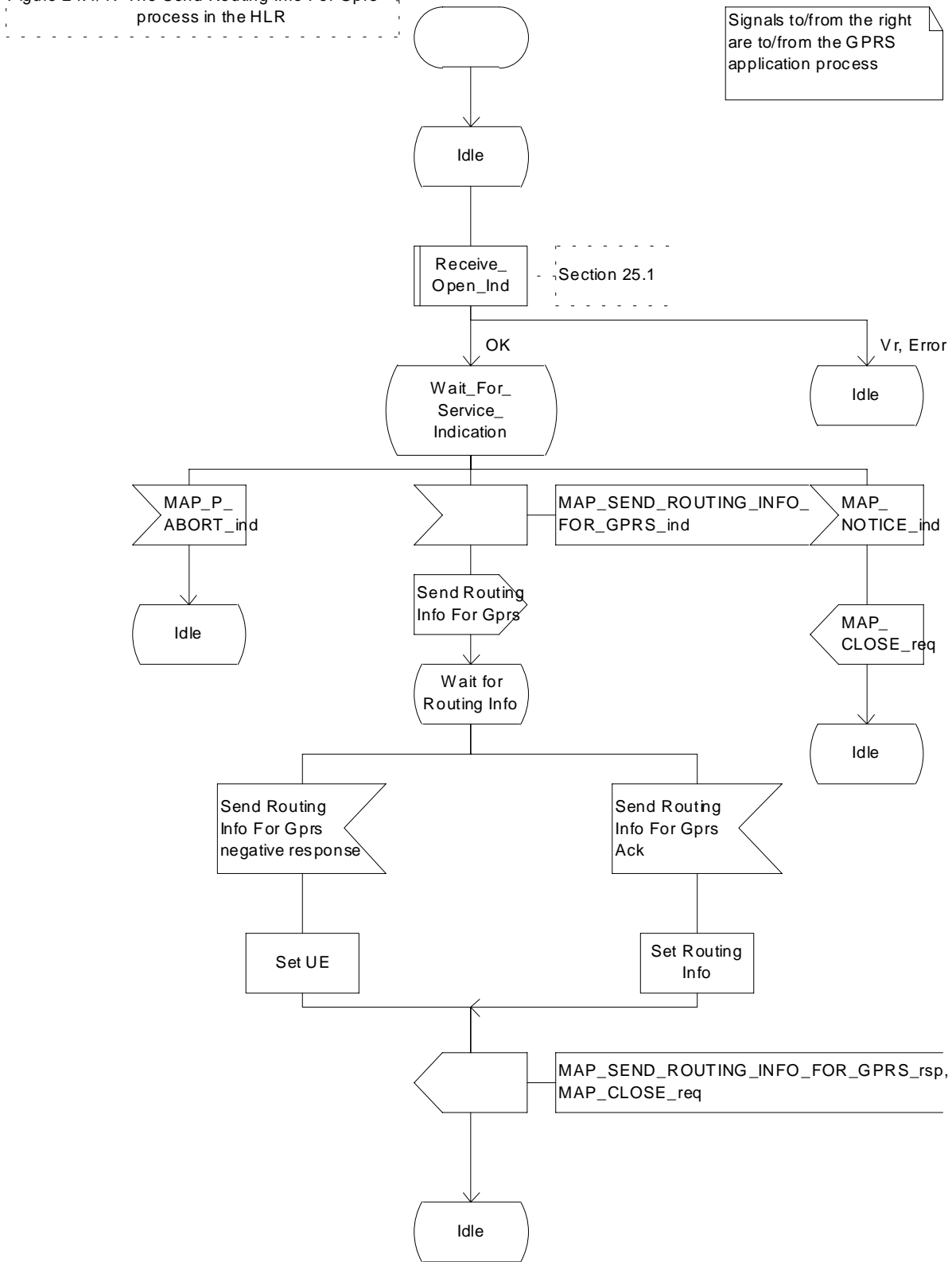
If the MAP provider sends a MAP_P_ABORT while the MAP process is waiting for a service indication, the MAP process returns to the idle state.

If the MAP provider sends a MAP_NOTICE while the MAP process is waiting for a service indication, the MAP process sends a MAP_CLOSE request to terminate the dialogue and returns to the idle state.

Process Send_Routing_Info_For_Gprs_HLR

24.1_1(1)

Figure 24.1/1: The Send Routing Info For Gprs process in the HLR



Process Send_Routing_Info_For_Gprs_HLR

24.1_1(1)

Figure 24.1/1: The Send Routing Info For GPRS process in the HLR

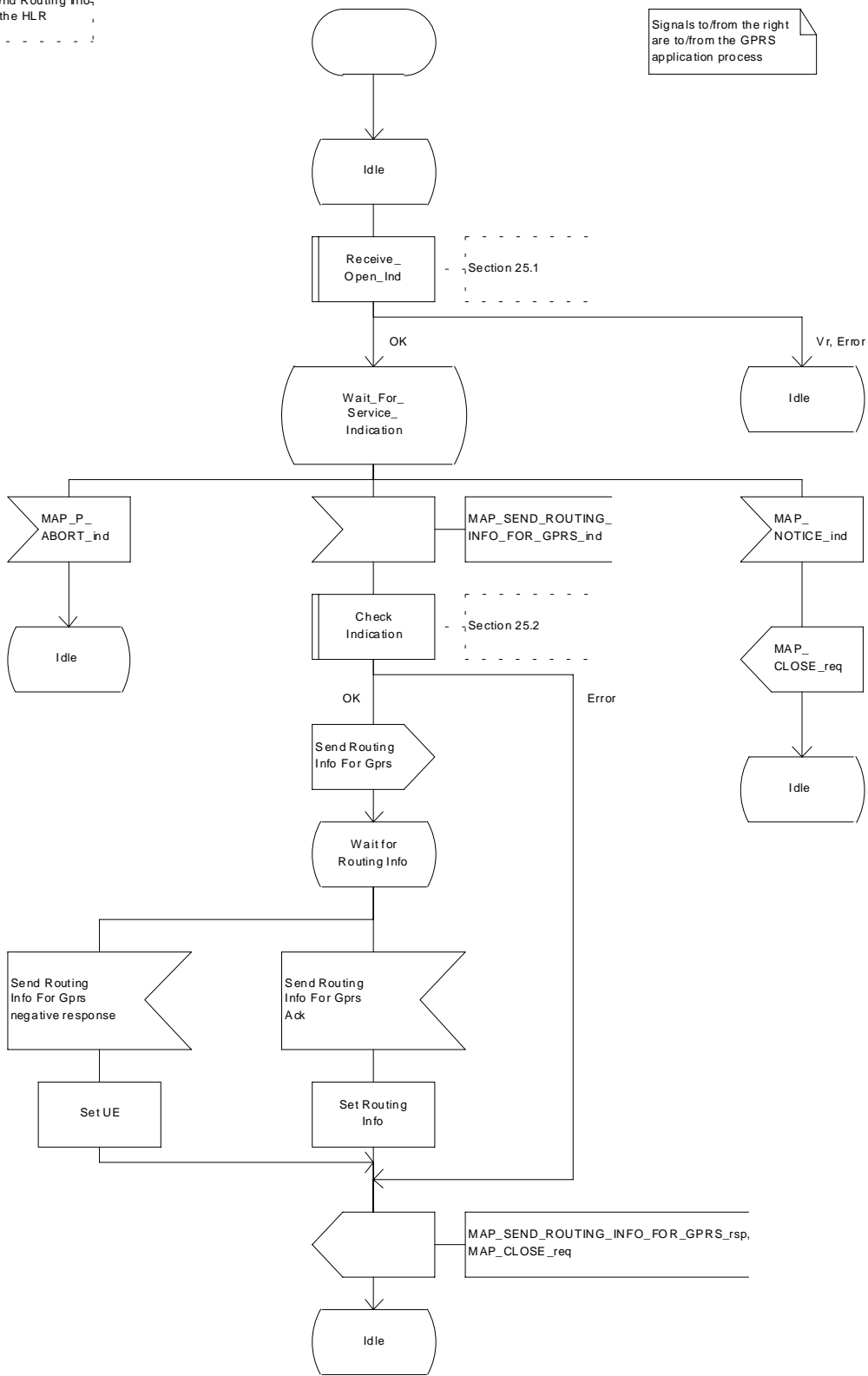


Figure 24.1/1: Process Send Routing Info For Gprs_HLR

24.1.2 Process in the GGSN for Send Routing Information for GPRS

Successful Outcome

When the MAP process receives a Send Routing Info For Gprs request from the GPRS application process in the GGSN, it requests a dialogue with the HLR whose identity is contained in the Send Routing Info For Gprs request by sending a MAP_OPEN service request, requests routing information using a MAP_SEND_ROUTING_INFO_FOR_GPRS service request and invokes the macro Receive_Open_Cnf to wait for the response to the dialogue opening request. If the dialogue opening is successful, the MAP process waits for a response from the HLR.

If the MAP process receives a MAP_SEND_ROUTING_INFO_FOR_GPRS service confirm from the HLR, the MAP process invokes the macro Check_Confirmation to check the content of the confirm.

If the macro Check_Confirmation takes the OK exit, the MAP process sends a Send Routing Info For Gprs ack containing the routing information received from the HLR to the GPRS application process in the GGSN and returns to the idle state.

Failure of dialogue opening with the HLR

If the macro Receive_Open_Cnf takes the Vr exit or the Error exit, the MAP process sends a negative response to the GPRS application process in the GGSN and returns to the idle state.

Error in MAP_SEND_ROUTING_INFO_FOR_GPRS confirm

If the MAP_SEND_ROUTING_INFO_FOR_GPRS service confirm contains a user error or a provider error, or the macro Check_Confirmation indicates that there is a data error, the MAP process sends a Send Routing Info For Gprs negative response to the GPRS application process in the GGSN and returns to the idle state.

Abort of HLR dialogue

After the dialogue with the HLR has been established, the MAP service provider may abort the dialogue by issuing a MAP_P_ABORT or a MAP_U_ABORT indication. In this case, the MAP process sends a Send Routing Info For Gprs negative response to the GPRS application process in the GGSN and returns to the idle state.

If the MAP provider indicates a protocol problem by sending a MAP_NOTICE indication, the MAP process closes the dialogue with the HLR, sends a Send Routing Info For Gprs negative response indicating system failure to the GPRS application process in the GGSN and returns to the idle state.

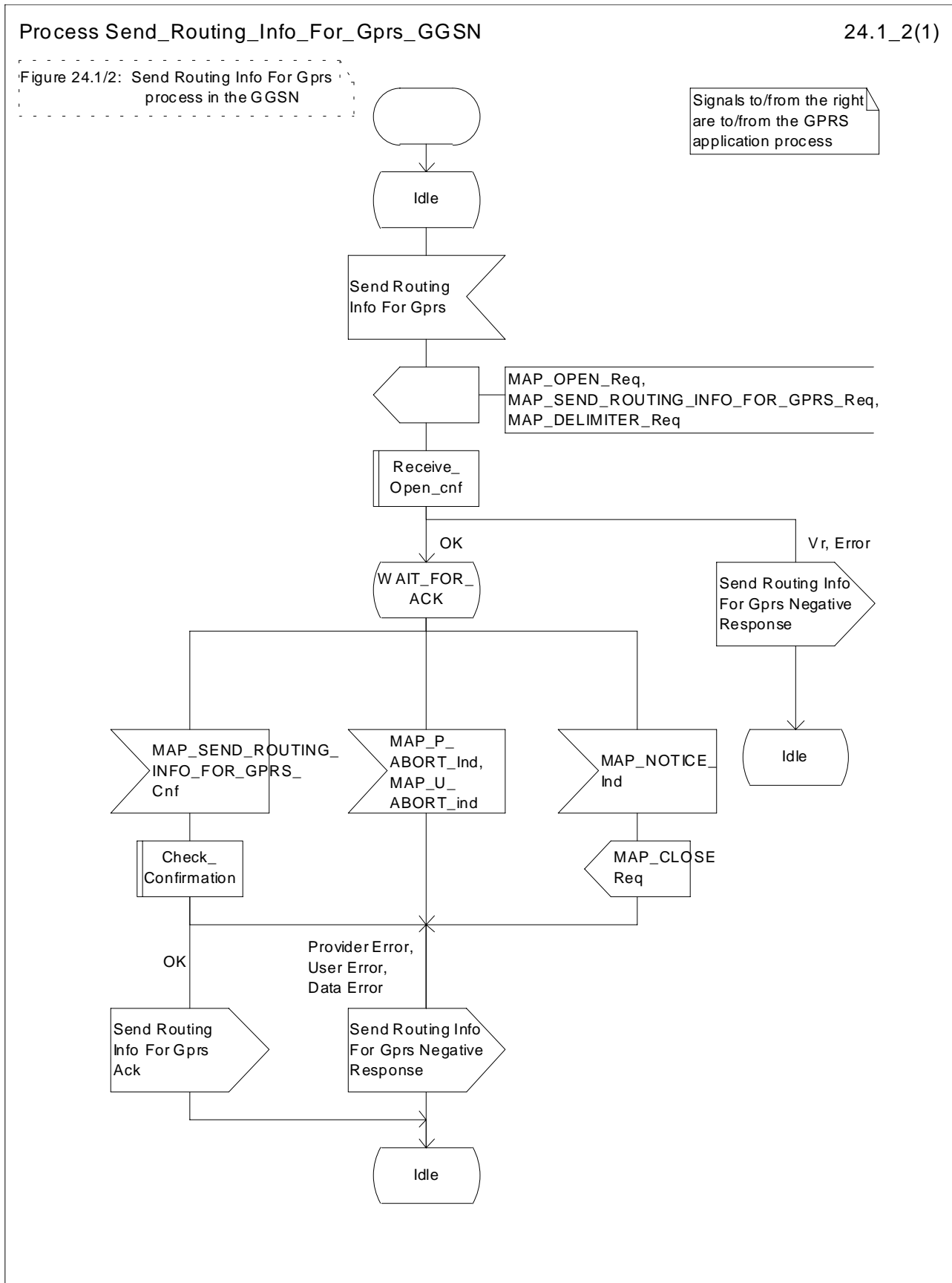


Figure 24.1/2: Process Send_Routing_Info_For_Gprs_GGSN

24.2.1 Process in the HLR for Failure Report

The MAP process in the HLR to set the MNRG (Mobile station Not Reachable for GPRS) flag for the subscriber is shown in figure 24.2/1. The MAP process invokes a macro not defined in this subclause; the definition of this macro can be found as follows:

Receive_Open_Ind	see subclause 25.1.1;
Check Indication	see subclause 25.2.1.

Successful outcome

When the MAP process receives a MAP_OPEN indication with the application context failureReport, it checks it by invoking the macro Receive_Open_Ind.

If the macro takes the OK exit, the MAP process waits for a service indication.

If a MAP_FAILURE_REPORT service indication is received, the HLR sends a Failure Report request to the GPRS application process in the HLR, and wait for a response. The Failure Report request contains the parameter received in the MAP_FAILURE_REPORT service indication.

If a positive response is received, the MAP process constructs a MAP_FAILURE_REPORT service response, constructs a MAP_CLOSE service request, sends them to the GGSN and returns to the idle state.

Negative response from HLR GPRS application process

If the GPRS application process in the HLR returns a negative response, the MAP process constructs a MAP_FAILURE_REPORT service response containing the appropriate error, constructs a MAP_CLOSE service request, sends them to the GGSN and returns to the idle state.

Failure of dialogue opening with the GGSN

If the macro Receive_Open_Ind takes the Vr exit or the Error exit, the MAP process returns to the idle state.

If the MAP provider sends a MAP_P_ABORT while the MAP process is waiting for a service indication, the MAP process returns to the idle state.

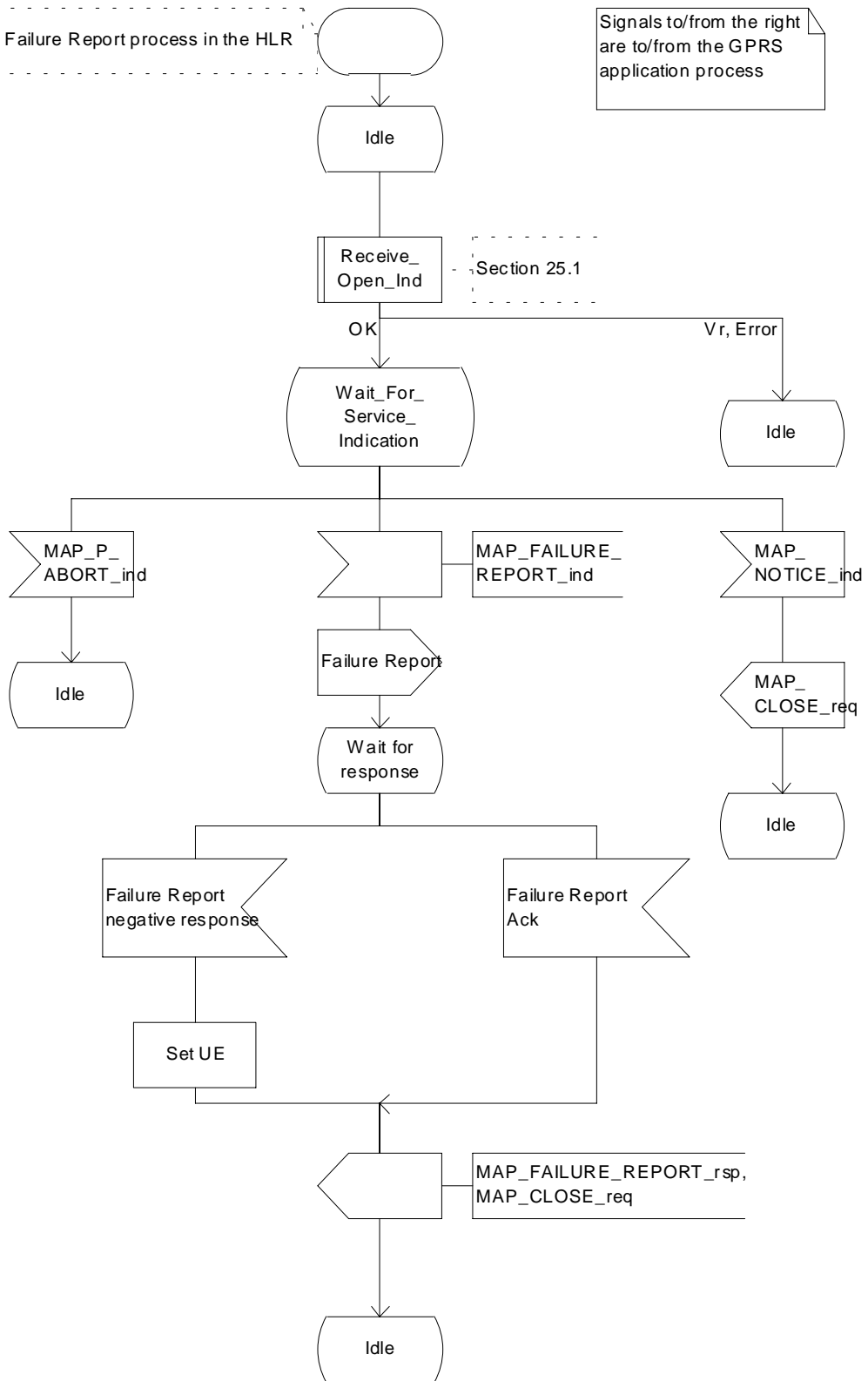
If the MAP provider sends a MAP_NOTICE while the MAP process is waiting for a service indication, the MAP process sends a MAP_CLOSE request to terminate the dialogue and returns to the idle state.

Process Failure_Report_HLR

24.2_1(1)

Figure 24.2/1: The Failure Report process in the HLR

Signals to/from the right are to/from the GPRS application process



Process Failure_Report_HLR

24.2_1(1)

Figure 24.2/1: The Failure Report process in the HLR

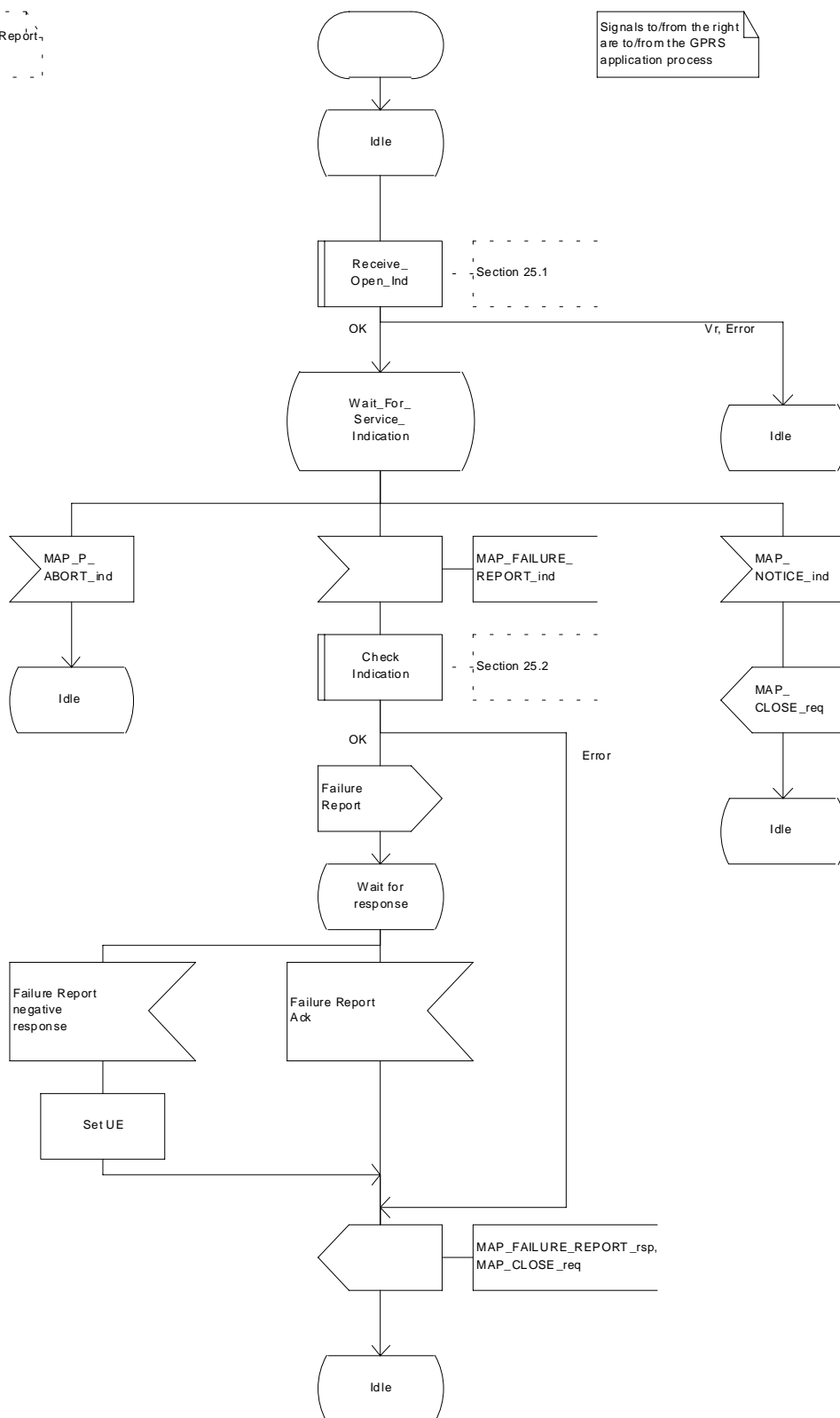


Figure 24.2/1: Process Failure_Report_HLR

24.2.2 Process in the GGSN for Failure Report

Successful Outcome

When the MAP process receives a Failure Report request from the GPRS application process in the GGSN, it requests a dialogue with the HLR whose identity is contained in the Failure Report request by sending a MAP_OPEN service request, sending failure information using a MAP_FAILURE_REPORT service request and invokes the macro Receive_Open_Cnf to wait for the response to the dialogue opening request. If the dialogue opening is successful, the MAP process waits for a response from the HLR.

If the MAP process receives a MAP_FAILURE_REPORT service confirm from the HLR, the MAP process invokes the macro Check_Confirmation to check the content of the confirm.

If the macro Check_Confirmation takes the OK exit, the MAP process sends a Failure Report ack containing the information received from the HLR to the GPRS application process in the GGSN and returns to the idle state.

Failure of dialogue opening with the HLR

If the macro Receive_Open_Cnf takes the Vr exit or the Error exit, the MAP process sends a negative response to the GPRS application process in the GGSN and returns to the idle state.

Error in MAP_FAILURE_REPORT confirm

If the MAP_FAILURE_REPORT service confirm contains a user error or a provider error, or the macro Check_Confirmation indicates that there is a data error, the MAP process sends a Failure Report negative response to the GPRS application process in the GGSN and returns to the idle state.

Abort of HLR dialogue

After the dialogue with the HLR has been established, the MAP service provider may abort the dialogue by issuing a MAP_P_ABORT or a MAP_U_ABORT indication. In this case, the MAP process sends a Failure Report negative response to the GPRS application process in the GGSN and returns to the idle state.

If the MAP provider indicates a protocol problem by sending a MAP_NOTICE indication, the MAP process closes the dialogue with the HLR, sends a Failure Report negative response indicating system failure to the GPRS application process in the GGSN and returns to the idle state.

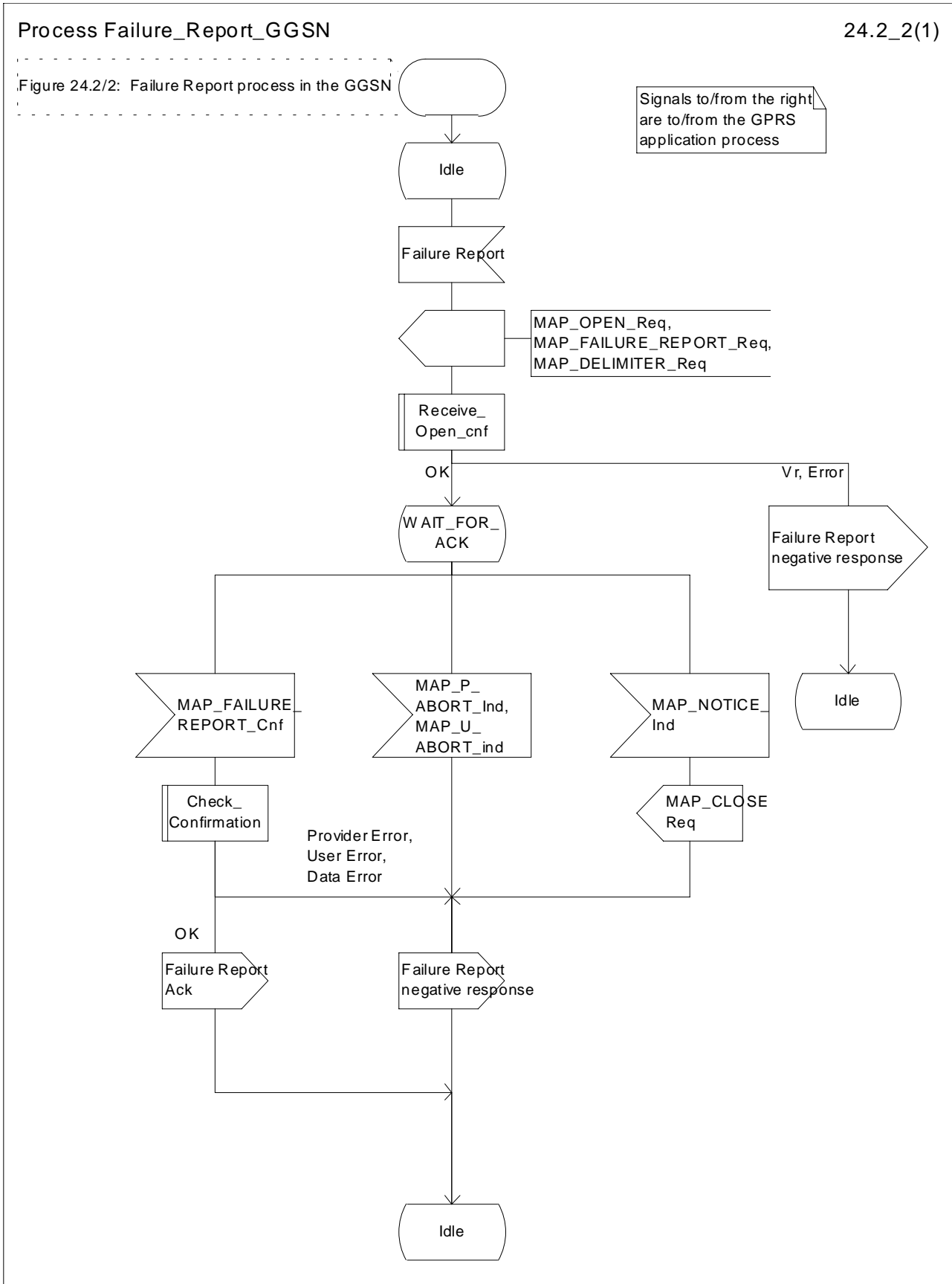


Figure 24.2/2: Process Failure_Report_GGSN

24.3.1 Process in the GGSN for Note Ms Present For Gprs

The MAP process in the GGSN to inform that the subscriber is present for GPRS again is shown in figure 24.3/1. The MAP process invokes a macro not defined in this subclause; the definition of this macro can be found as follows:

Receive_Open_Ind	see subclause 25.1.1;
Check_Indication	see subclause 25.2.1.

Successful outcome

When the MAP process receives a MAP_OPEN indication with the application context gprsNotify, it checks it by invoking the macro Receive_Open_Ind.

If the macro takes the OK exit, the MAP process waits for a service indication.

If a MAP_NOTE_MS_PRESENT_FOR_GPRS service indication is received, the GGSN sends a Note Ms Present For Gprs request to the GPRS application process in the GGSN, and wait for a response. The Note Ms Present For Gprs request contains the parameter received in the MAP_NOTE_MS_PRESENT_FOR_GPRS service indication.

If the GPRS application process in the GGSN returns a positive response, the MAP process constructs a MAP_NOTE_MS_PRESENT_FOR_GPRS service response, constructs a MAP_CLOSE service request, sends them to the HLR and returns to the idle state.

Negative response from GGSN GPRS application process

If the GPRS application process in the GGSN returns a negative response, the MAP process constructs a MAP_NOTE_MS_PRESENT_FOR_GPRS service response containing the appropriate error, constructs a MAP_CLOSE service request, sends them to the HLR and returns to the idle state.

Failure of dialogue opening with the HLR

If the macro Receive_Open_Ind takes the Vr exit or the Error exit, the MAP process returns to the idle state.

If the MAP provider sends a MAP_P_ABORT while the MAP process is waiting for a service indication, the MAP process returns to the idle state.

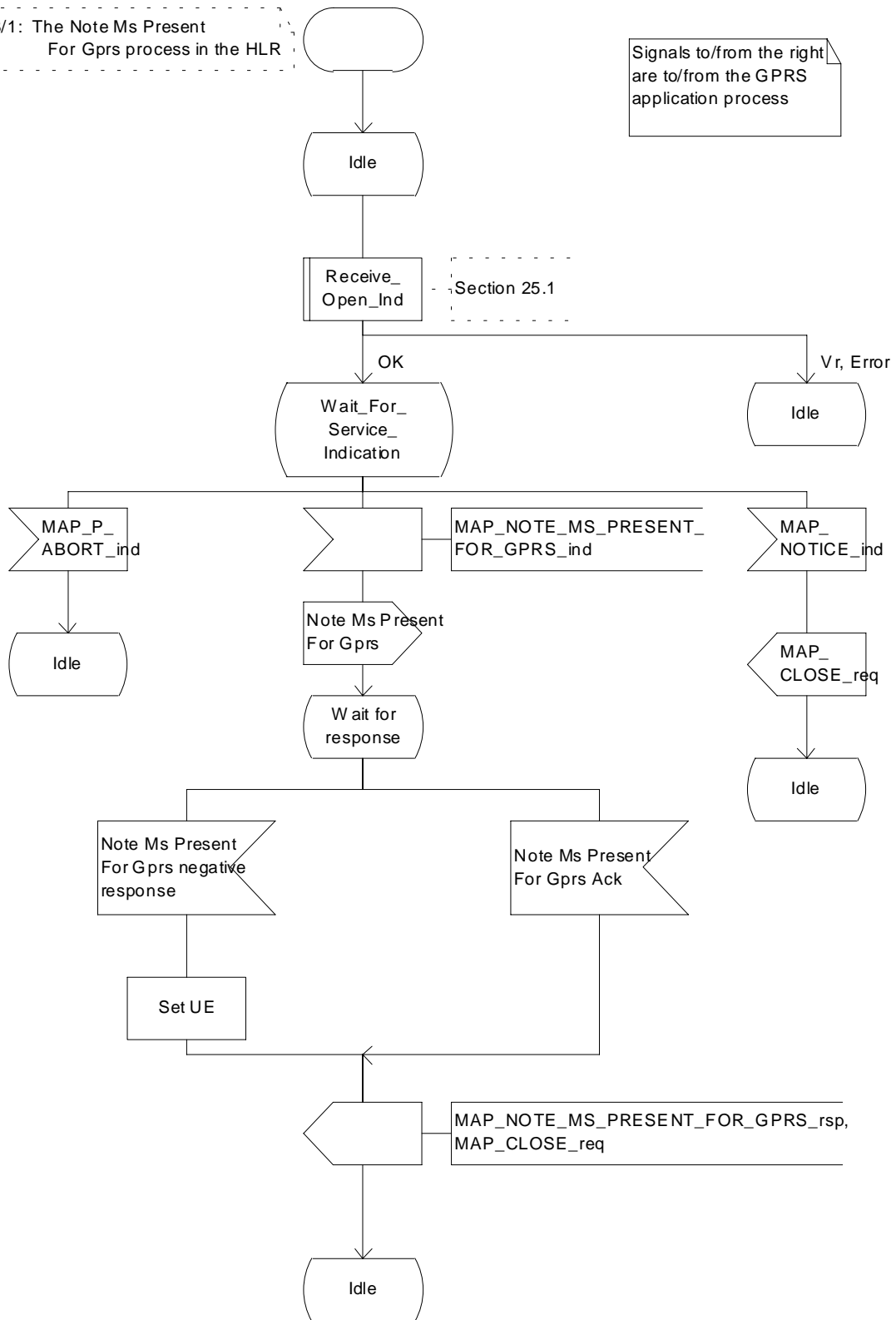
If the MAP provider sends a MAP_NOTICE while the MAP process is waiting for a service indication, the MAP process sends a MAP_CLOSE request to terminate the dialogue and returns to the idle state.

Process Note_Ms_Present_For_Gprs_GGSN

24.3_1(1)

Figure 24.3/1: The Note Ms Present For Gprs process in the HLR

Signals to/from the right are to/from the GPRS application process



Process Note_Ms_Present_For_Gprs_GGSN

24.3_1(1)

Figure 24.3/1: The Note Ms Present For Gprs process in the HLR

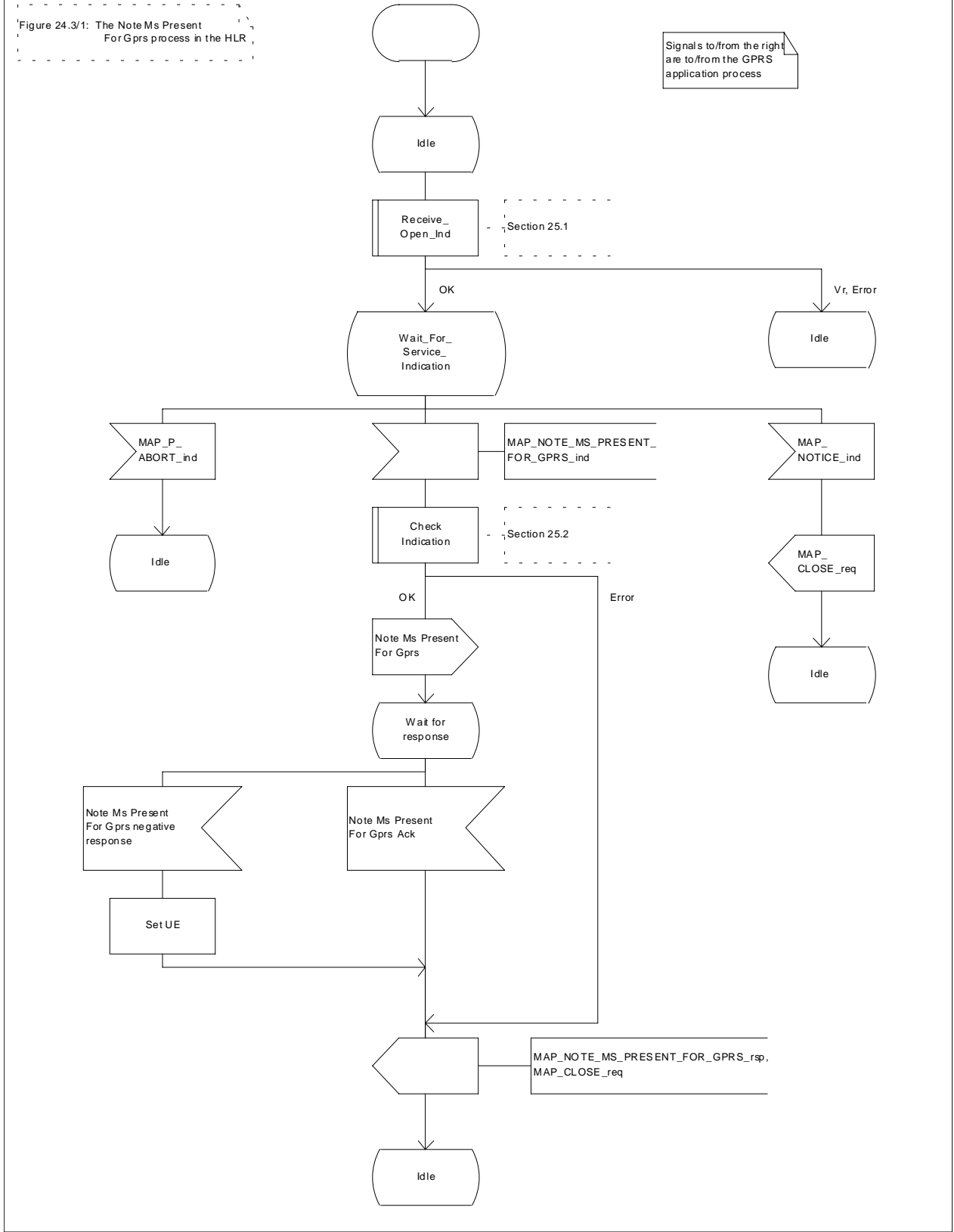


Figure 24.3/1: Process Note_Ms_Present_For_Gprs_GGSN

24.3.2 Process in the HLR for Note Ms Present For Gprs

Successful Outcome

When the MAP process receives a Note Ms Present For Gprs request from the GPRS application process in the HLR, it requests a dialogue with the GGSN whose identity is contained in the Note Ms Present For Gprs request by sending a MAP_OPEN service request, sending necessary information using a MAP_NOTE_MS_PRESENT_FOR_GPRS service request and invokes the macro Receive_Open_Cnf to wait for the response to the dialogue opening request. If the dialogue opening is successful, the MAP process waits for a response from the GGSN.

If the MAP process receives a MAP_NOTE_MS_PRESENT_FOR_GPRS service confirm from the GGSN, the MAP process invokes the macro Check_Confirmation to check the content of the confirm.

If the macro Check_Confirmation takes the OK exit, the MAP process sends a Note Ms Present For Gprs ack containing the information received from the GGSN to the GPRS application process in the HLR and returns to the idle state.

Failure of dialogue opening with the GGSN

If the macro Receive_Open_Cnf takes the Vr exit or the Error exit, the MAP process sends a negative response to the GPRS application process in the HLR and returns to the idle state.

Error in MAP_NOTE_MS_PRESENT_FOR_GPRS confirm

If the MAP_NOTE_MS_PRESENT_FOR_GPRS service confirm contains a user error or a provider error, or the macro Check_Confirmation indicates that there is a data error, the MAP process sends a Note Ms Present For Gprs negative response to the GPRS application process in the HLR and returns to the idle state.

Abort of GGSN dialogue

After the dialogue with the GGSN has been established, the MAP service provider may abort the dialogue by issuing a MAP_P_ABORT or a MAP_U_ABORT indication. In this case, the MAP process sends a Note Ms Present For Gprs negative response to the GPRS application process in the HLR and returns to the idle state.

If the MAP provider indicates a protocol problem by sending a MAP_NOTICE indication, the MAP process closes the dialogue with the GGSN, sends a Failure Report negative response indicating system failure to the GPRS application process in the HLR and returns to the idle state.

Process Note_MS_Present_For_Gprs_HLR

24.3_2(1)

Figure 24.3/2: Note Ms Present For Gprs process in the HLR

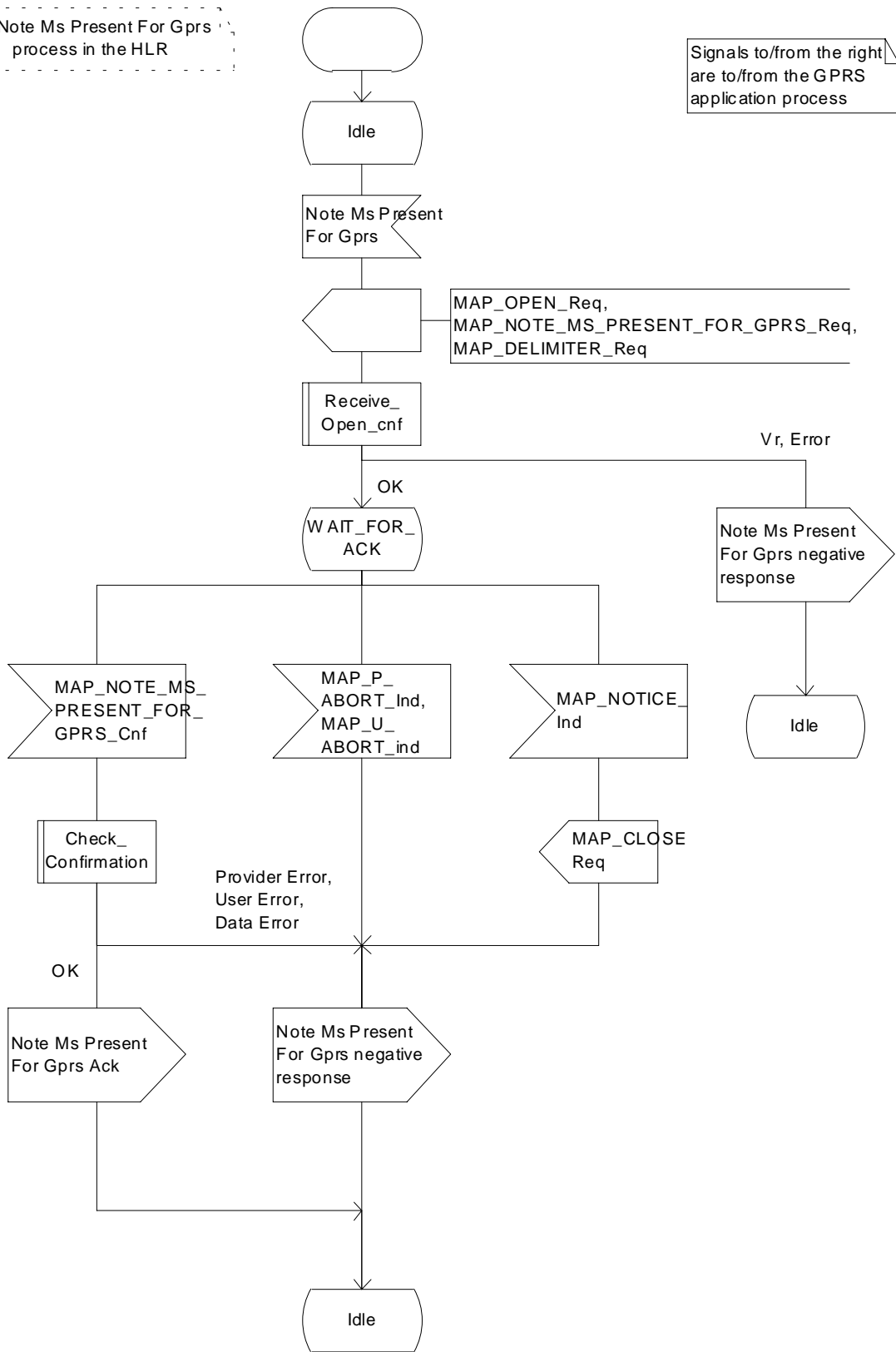


Figure 24.3/2: Process Note_Ms_Present_For_Gprs_HLR

24A CSE control of subscriber data

24A.1 Any Time Subscription Interrogation procedure

24A.1.1 General

The message flows for successful retrieval of subscription information related to an any time interrogation from the CAMEL server are shown in figure 24A.1/1.

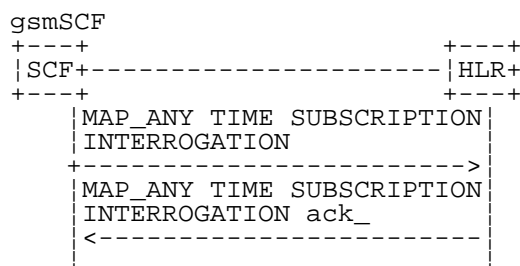


Figure 24A.1/1: Message flow for any time subscription interrogation

The following MAP services are used to retrieve requested information:

MAP_ANY_TIME_SUBSCRIPTION_INTERROGATION see subclause 8.11.x;

24A.1.2 Process in the gsmSCF

Out of the scope of the MAP specification.

24A.1.3 Process in the HLR

The MAP process in the HLR to provide subscription information in response to an interrogation from the CAMEL server is shown in figure 24A.1/2. The MAP process invokes macros not defined in this subclause; the definitions of these macros can be found as follows:

Receive_Open_Ind see subclause 25.1.1;

Successful outcome

When the MAP process receives a MAP_OPEN indication with the application context anyTimeInformationHandling, it checks it by invoking the macro Receive_Open_Ind.

If the macro takes the OK exit, the MAP process waits for a service indication.

If a MAP_ANY_TIME_SUBSCRIPTION_INTERROGATION service indication is received, the MAP process sends an Any Time Subscription Interrogation request to the call handling process in the HLR (described in 3G TS 23.078), and waits for a response. The Any Time Subscription Interrogation request contains the parameters received in the MAP_ANY_TIME_SUBSCRIPTION_INTERROGATION service indication.

If the call handling process in the HLR returns an Any Time Subscription Interrogation response, the MAP process constructs a MAP_ANY_TIME_SUBSCRIPTION_INTERROGATION service response containing the subscription information contained in the Any Time Subscription Interrogation response, constructs a MAP_CLOSE service request, sends them to the CAMEL server and returns to the idle state. If the MAP_ANY_TIME_SUBSCRIPTION_INTERROGATION service response cannot be carried in a single TC-Result component, it is carried in one or more TC-Result-NL components (each sent in a TC-CONTINUE), followed by a TC-Result-L component in a TC-END message.

Negative response from HLR call handling process

If the call handling process in the HLR returns a negative response to obtain subscription information, the MAP process constructs a MAP_ANY_TIME_SUBSCRIPTION_INTERROGATION service response containing the appropriate error, constructs a MAP_CLOSE service request, sends them to the CAMEL server and returns to the idle state.

Failure of dialogue opening with the CAMEL server

If the macro Receive_Open_Ind takes the Vr or Error exit, the MAP process returns to the idle state.

If the MAP provider sends a MAP_P_ABORT while the MAP process is waiting for a service indication, the MAP process returns to the idle state.

If the MAP provider sends a MAP_NOTICE while the MAP process is waiting for a service indication, the MAP process sends a MAP_CLOSE request to terminate the dialogue and returns to the idle state.

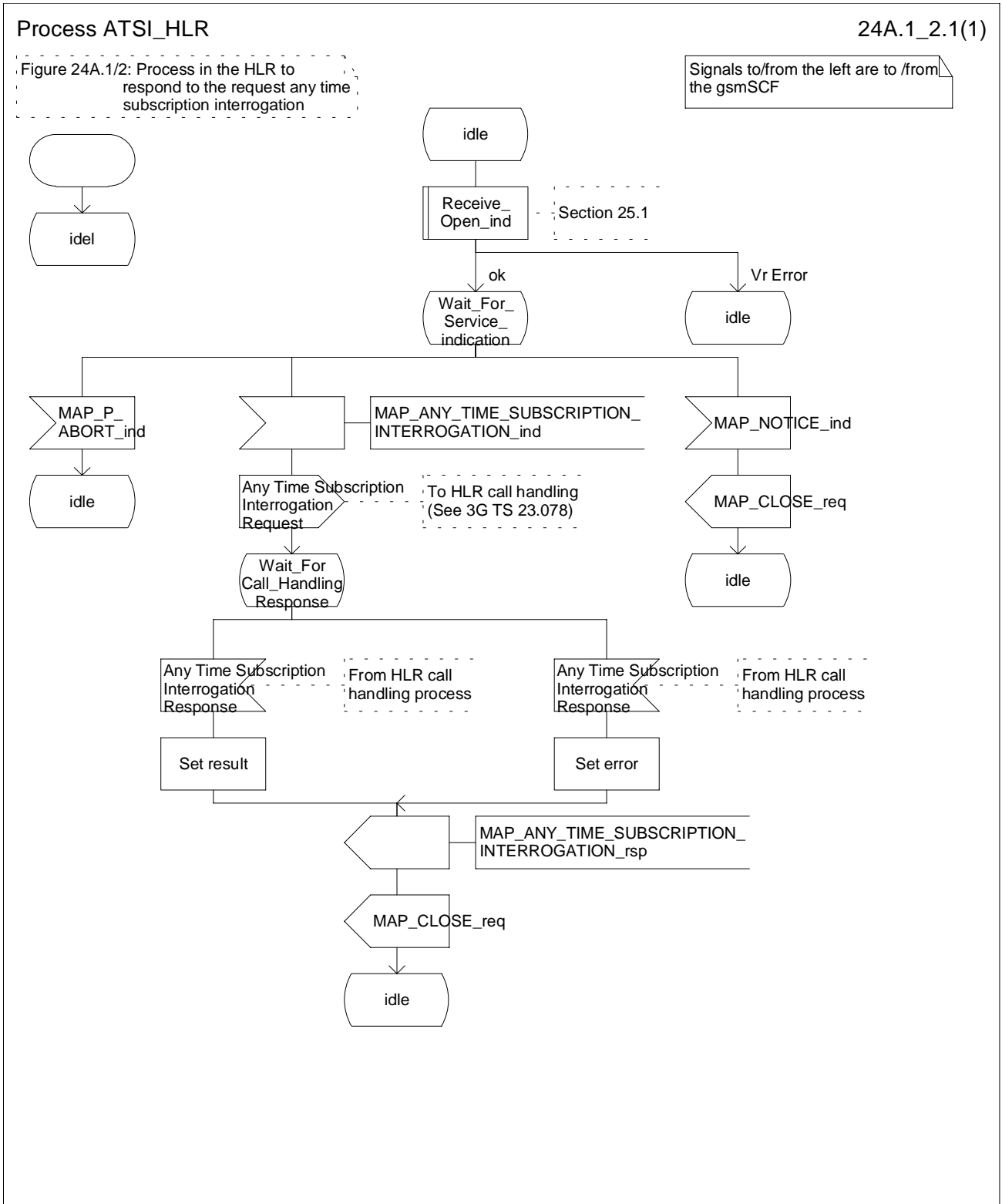


Figure 24A.1/2: Process ATSI_HLR

24A.2 Any Time Modification procedure

24A.2.1 General

The message flows for successful modification of subscriber information related to an any time modification from the CAMEL server are shown in figure 24A.2/1

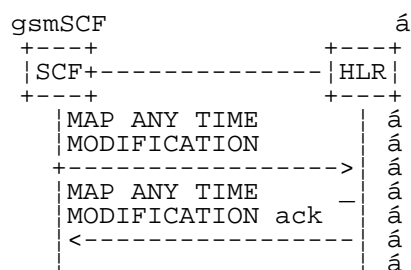


Figure 24A.2/1 Message flow for any time modification

The following MAP services are used to modify subscription information:

MAP_ANY_TIME_MODIFICATION see subclause 8.11.x;

24A.2.2 Process in the gsmSCF

Out of the scope of the MAP specification.

24A.2.3 Process in the HLR

The MAP process in the HLR to modify subscriber information in response to a modification request from the CAMEL server is shown in figure 24A.2/2. The MAP process invokes macros not defined in this subclause; the definitions of these macros can be found as follows:

Receive_Open_Ind see subclause 25.1.1;

Insert_Subst_Data_Stand_Alone_HLR see subclause 25.7.2;

Successful outcome

When the MAP process receives a MAP_OPEN indication with the application context anyTimeInformationHandling, it checks it by invoking the macro Receive_Open_Ind.

If the macro takes the OK exit, the MAP process waits for a service indication.

If a MAP_ANY_TIME_MODIFICATION service indication is received, the MAP process sends an Any Time modification request to the call handling process in the HLR (described in 3G TS 23.078), and waits for a response. The Any Time modification request contains the parameters received in the MAP_ANY_TIME_MODIFICATION service indication.

If the call handling process in the HLR returns an Any Time modification response, the MAP process constructs a MAP_ANY_TIME_MODIFICATION service response containing the modified subscription information contained in the Any Time modification response, constructs a MAP_CLOSE service request, sends them to the CAMEL server. If the MAP_ANY_TIME_MODIFICATION service response cannot be carried in a single TC-Result component, it is carried in one or more TC-Result-NL components (each sent in a TC-CONTINUE), followed by a TC-Result-L component in a TC-END message. If the VLR/SGSN is to be updated after the modification, the MAP_INSERT_SUBS_DATA_HLR process shall be initiated and then returns to the idle state.

Negative response from HLR call handling process

If the call handling process in the HLR returns a negative response to modify subscription information, the MAP process constructs a MAP_ANY_TIME_MODIFICATION service response containing the appropriate error, constructs a MAP_CLOSE service request, sends them to the CAMEL server and returns to the idle state.

Failure of dialogue opening with the CAMEL server

If the macro Receive_Open_Ind takes the Vr or Error exit, the MAP process returns to the idle state.

If the MAP provider sends a MAP_P_ABORT while the MAP process is waiting for a service indication, the MAP process returns to the idle state.

If the MAP provider sends a MAP_NOTICE while the MAP process is waiting for a service indication, the MAP process sends a MAP_CLOSE request to terminate the dialogue and returns to the idle state.

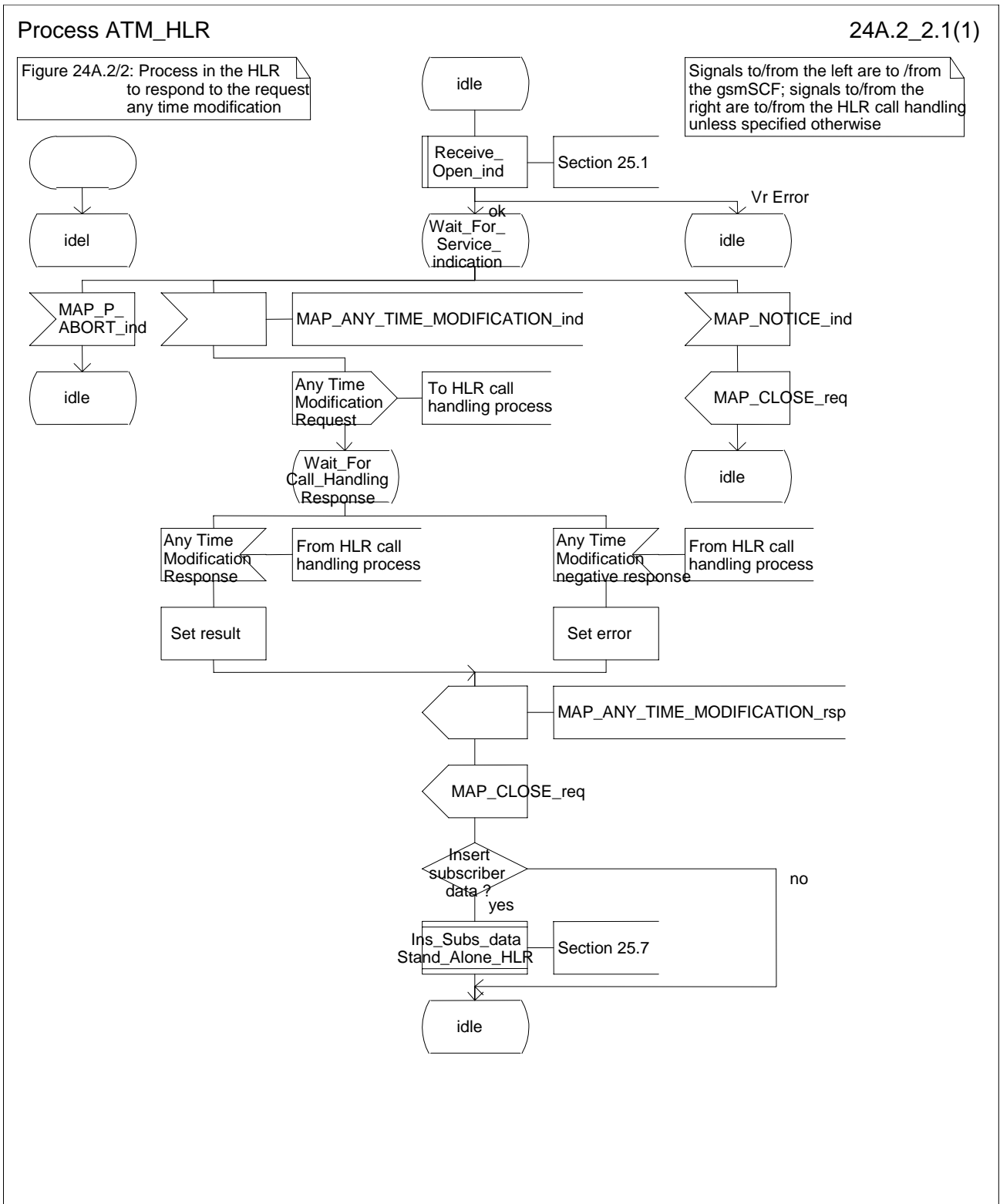


Figure 24A.2/2: Process ATM_HLR

24A.3 Subscriber Data Modification Notification procedure

24A.3.1 General

The Subscriber Data Modification Notification procedure is used to notify a gsmSCF about the modification of subscriber data.

The stage 2 specification for Subscriber Data Modification Notification is in TS 23.078. The interworking between the MAP signalling procedures and the Subscriber Data Modification Notification procedures for each entity (HLR, gsmSCF) is shown by the transfer of signals between these procedures.

The following services are used:



(1) MAP-NOTE_SUBSCRIBER_DATA_MODIFIED (HLR to gsmSCF)

(2) MAP-NOTE_SUBSCRIBER_DATA_MODIFIED-ACK (gsmSCF to HLR)

Figure 24A.3/1: Interfaces and services for subscriber data modification notification

24A.3.2 Processes in the MAP Entities

The text in this clause is a supplement to the definition in the SDL diagrams; it does not duplicate the information in the SDL diagrams.

24A.3.2.1 Process in the HLR

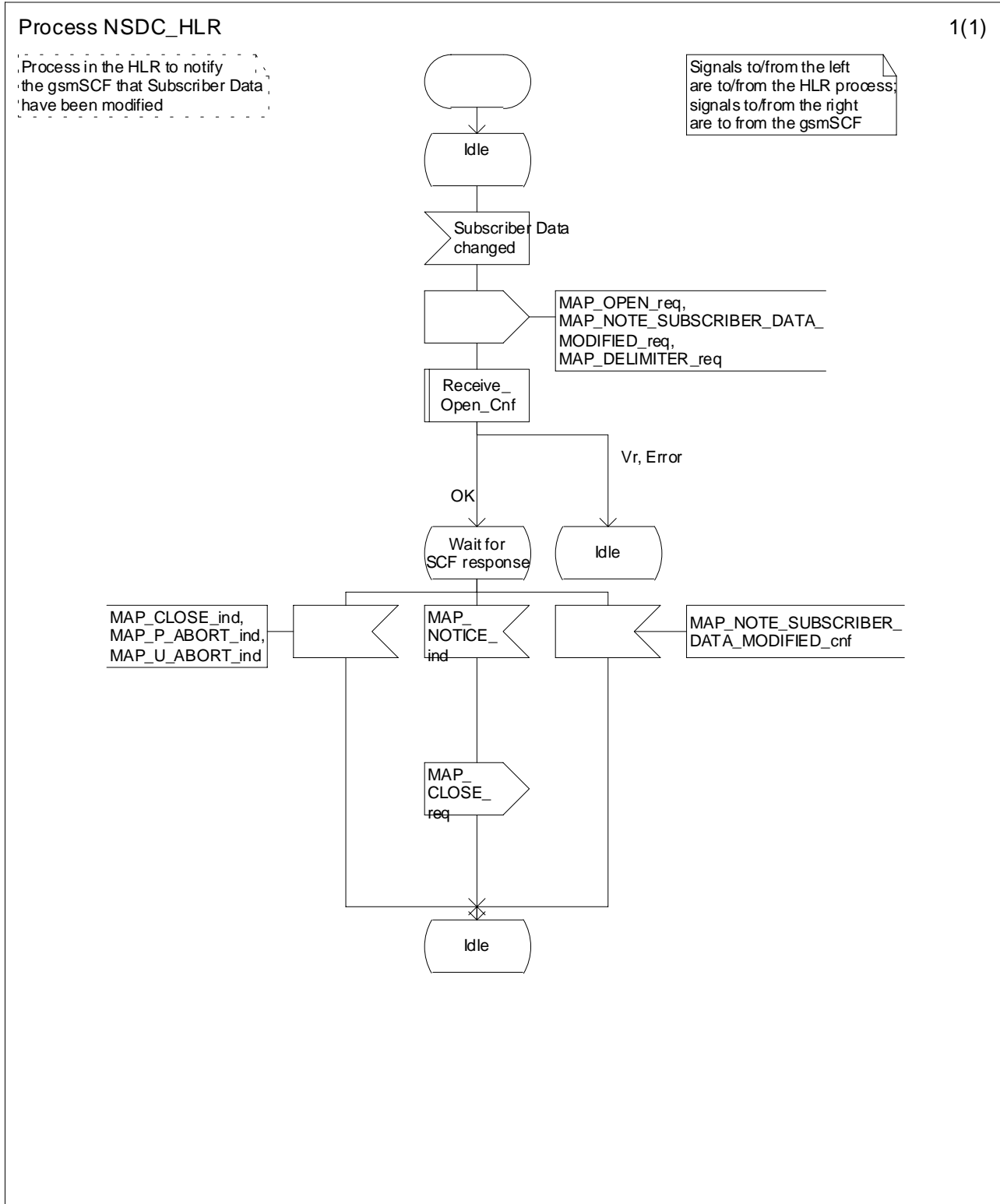


Figure 24A.3/2 Process Subscriber_Data_Modification_Notification_HLR (sheet 1 of 1)

24A.3.2.2 Process in the gsmSCF

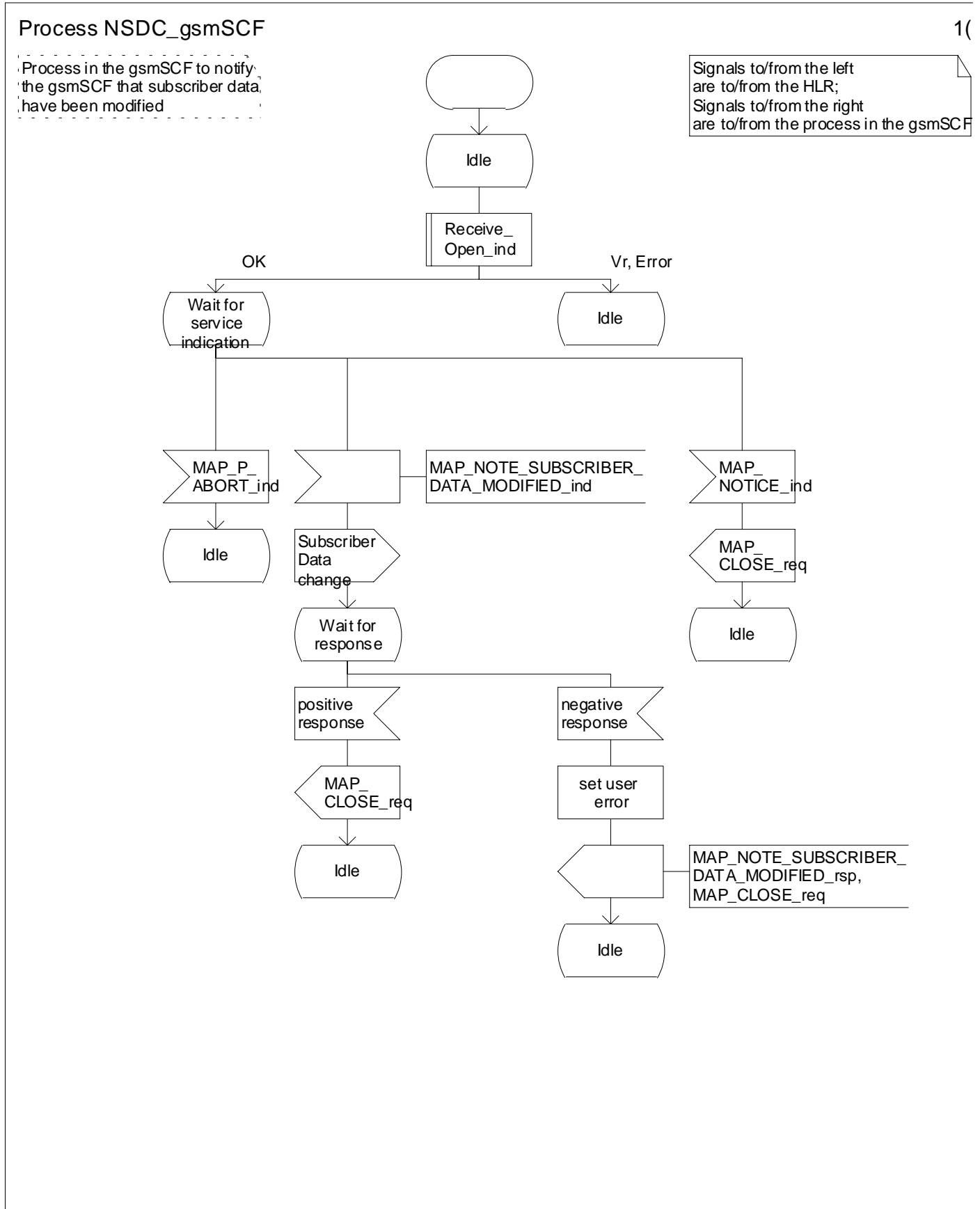


Figure 24A.3/3 Process Subscriber_Data_Modification_Notification_gsmSCF (sheet 1 of 1)

25 General macro description

25.1 MAP open macros

25.1.1 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 takes the Vr exit..

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

If the received version (Vr) is the one described in this version of MAP, the macro takes the OK exit, otherwise it takes the Vr exit..

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 takes the error exit.

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 takes the error exit.

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

If the received version (Vr) is the one described in this version of MAP, the macro takes the OK exit, otherwise it takes the Vr exit.

25.1.2 Macro Receive_Open_Cnf

This macro is used by a user procedure 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 takes the OK exit.

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 Vr" ($V_r < V_n$), the macro terminates in a way that causes restart of the dialogue by using the version Vr protocol. Otherwise, the macro takes the Error exit.

If the "Refuse-reason" parameter indicates neither "Potential Version Incompatibility" nor "Application Context Not Supported", the macro takes the Error exit.

If a MAP_U_ABORT, a MAP_P_ABORT or a MAP_NOTICE Indication is received, the macro takes the Error exit.

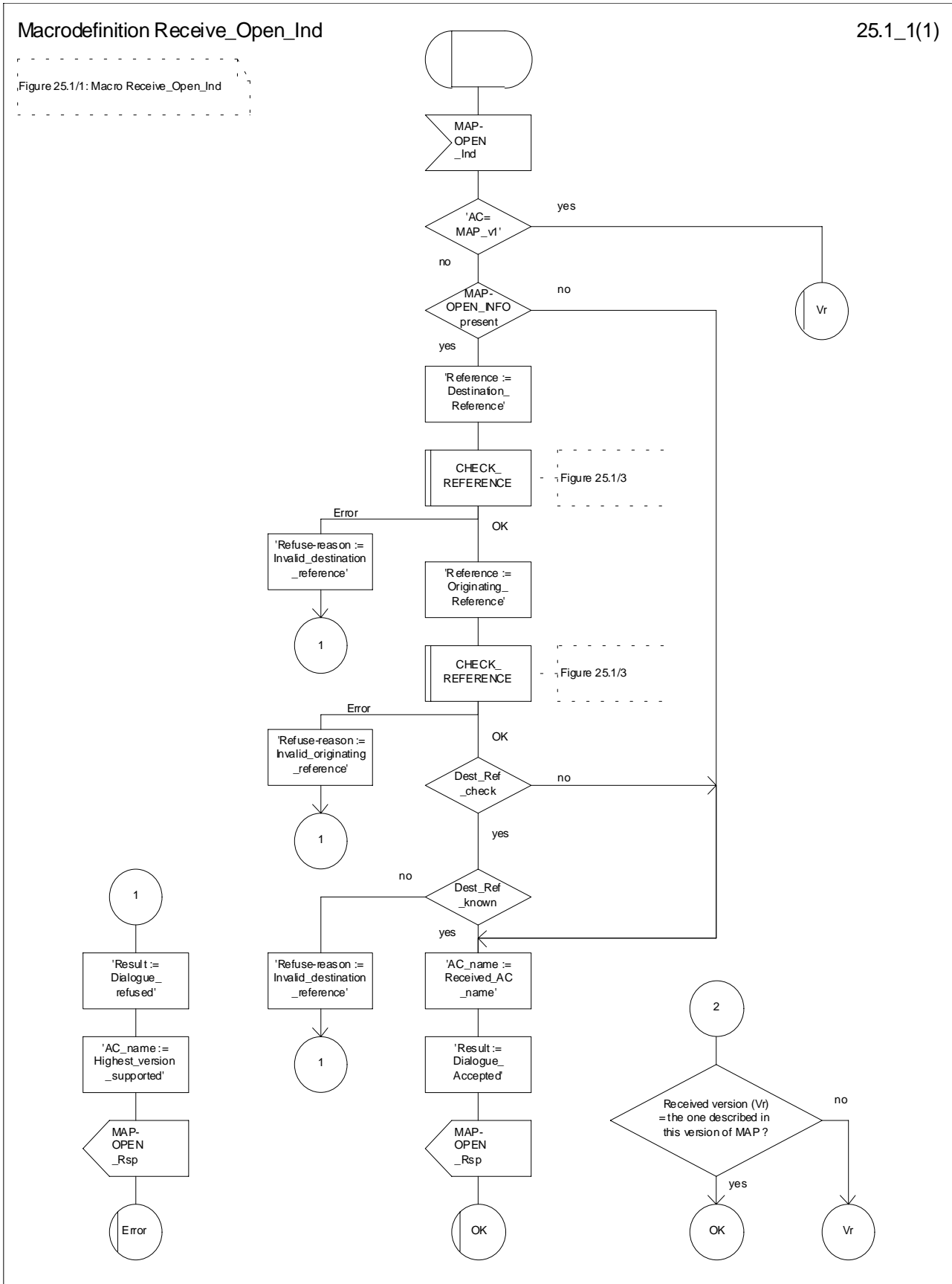


Figure 25.1/1: Macro Receive_Open_Ind

Macrodefinition Receive_Open_Cnf

25.1_2(1)

Figure 25.1/2: Macro to receive a MAP_OPEN_Cnf

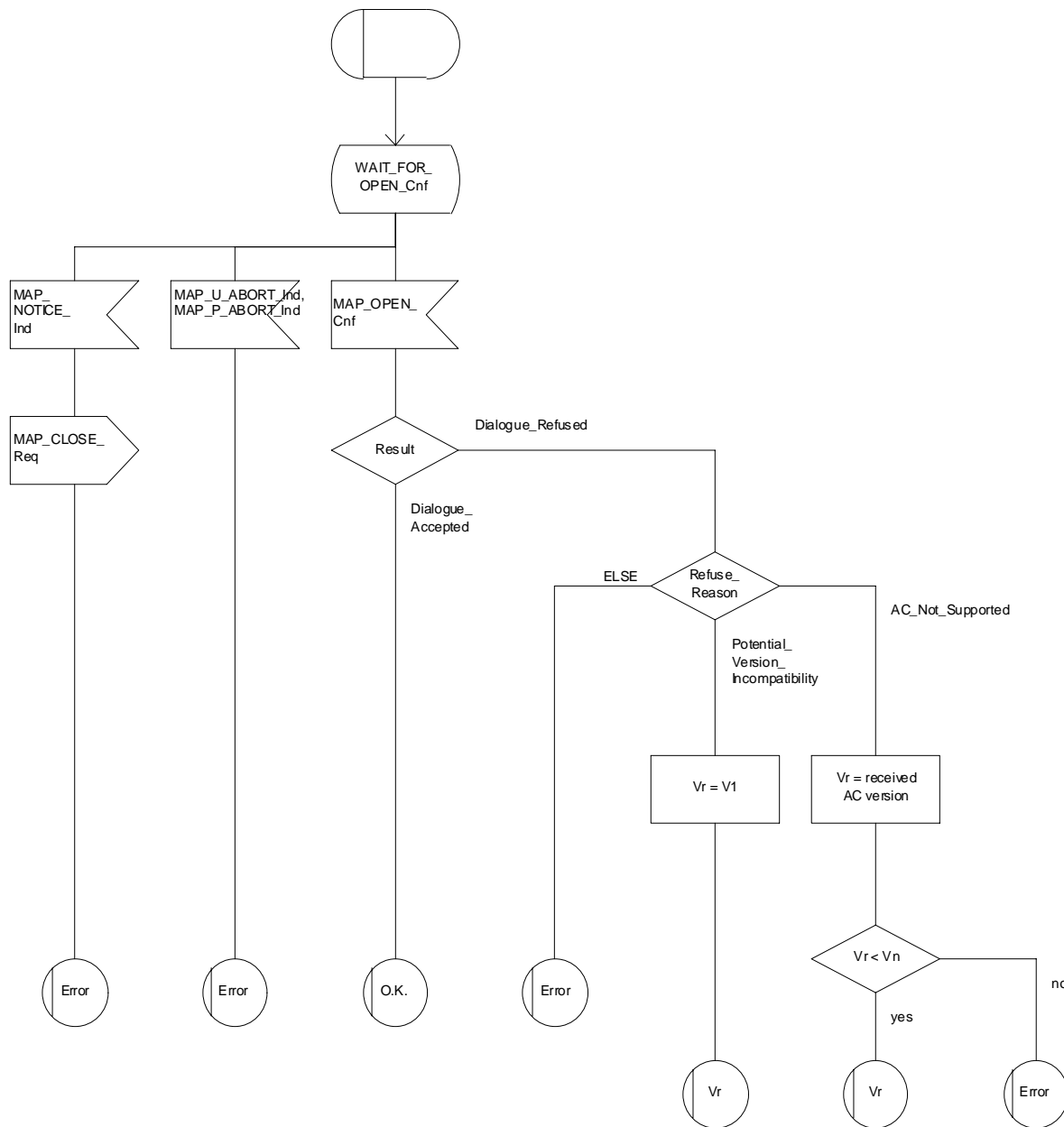


Figure 25.1/2: Macro Receive_Open_Cnf

Macrodefinition CHECK_REFERENCE

25.1_3(1)

Figure 25.1/3: Check of Destination Reference and Originating Reference received in a MAP-OPEN indication primitive

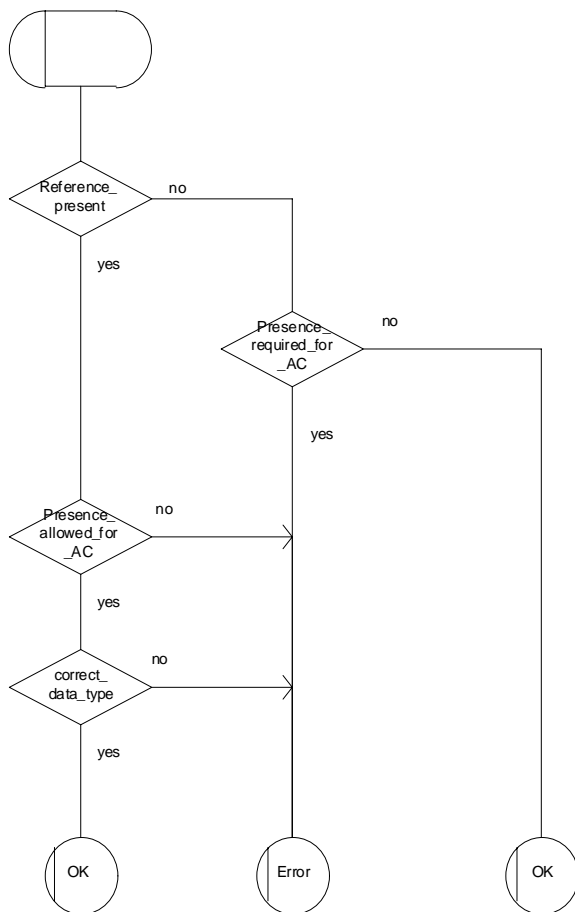


Figure 25.1/3: Macro CHECK_REFERENCE

25.2 Macros to check the content of indication and confirmation primitives

25.2.1 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 25.2/1.

25.2.2 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 25.2/2.

Macrodefinition Check_Indication

25.2_1(1)

Figure 25.2/1: Macro to check the parameters of an indication primitive

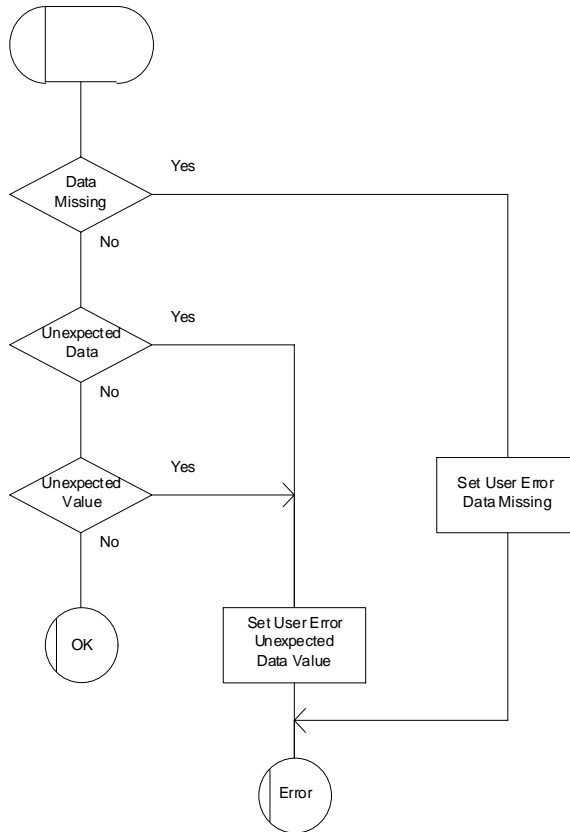


Figure 25.2/1: Macro Check_Indication

Macrodefinition Check_Confirmation

25.2_2(1)

Figure 25.2/2: Macro to check the parameters of a confirmation primitive

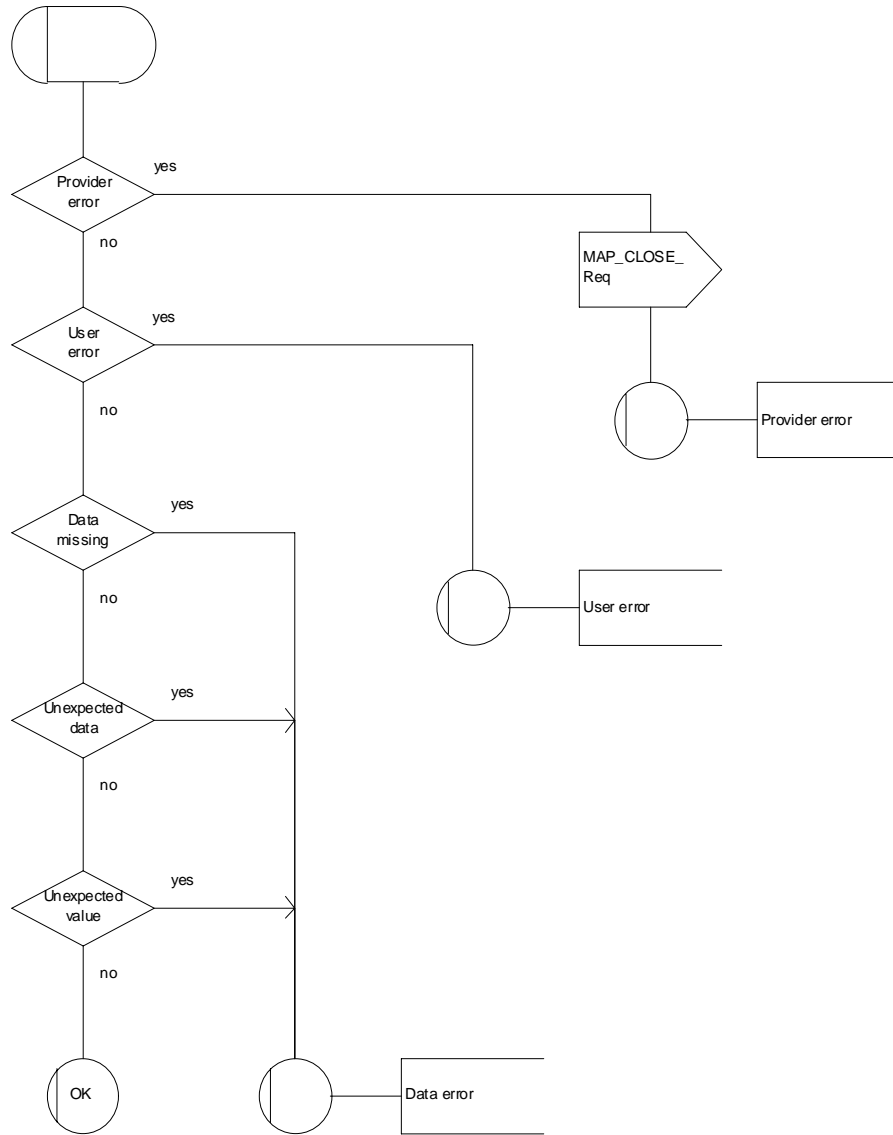


Figure 25.2/2: Macro Check_Confirmation

25.3 The page and search macros

25.3.1 Macro PAGE_MSC

This macro (see figure 25.3/1) 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 25.2). 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.

25.3.2 Macro Search_For_MS_MSC

This macro (see figure 25.3/2) 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 25.2). 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.

Macrodefinition Page_MSC

25.3_1(1)

Figure 25.3/1:
Macro Page_MSC

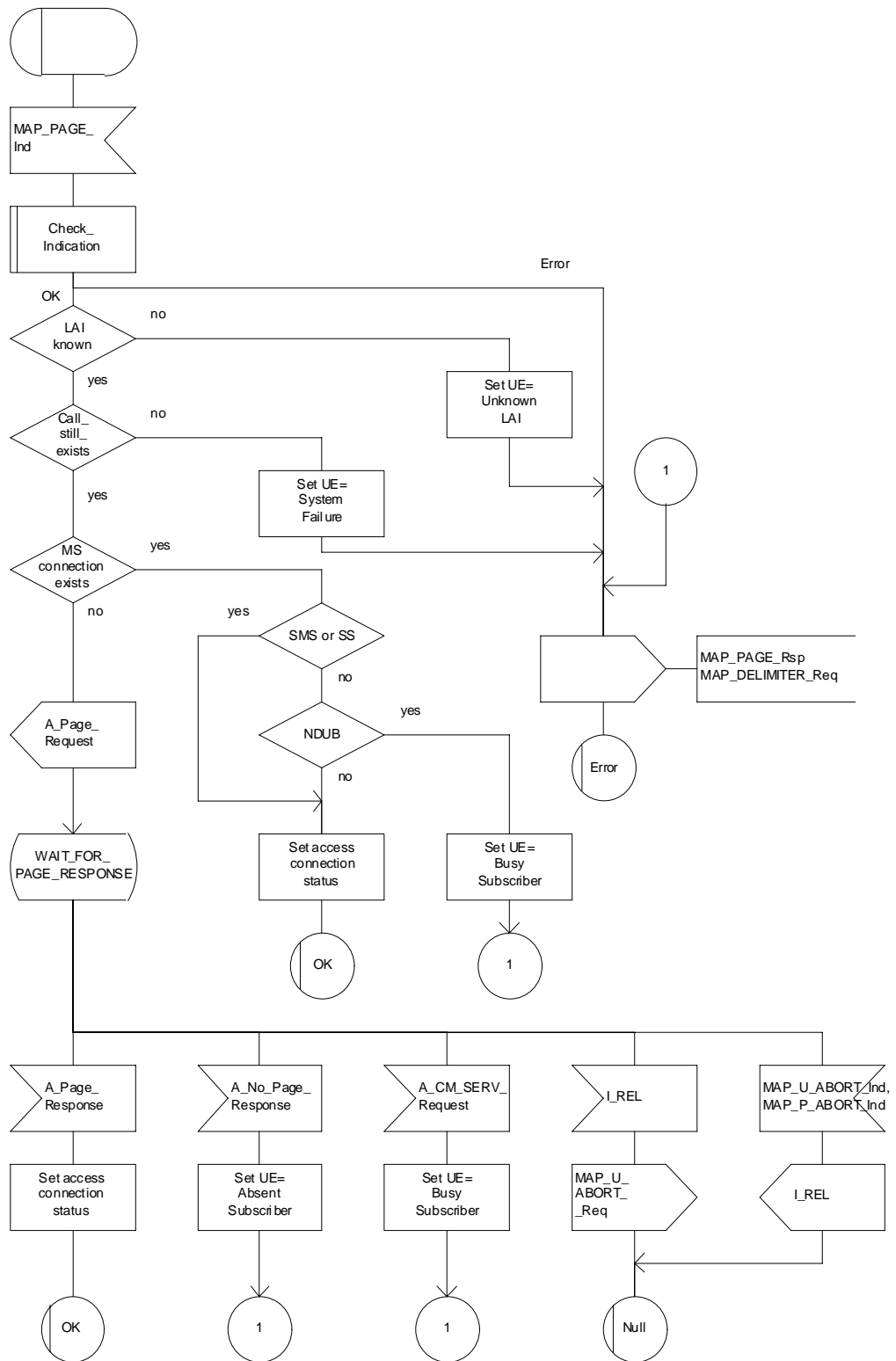


Figure 25.3/1: Macro Page_MSC

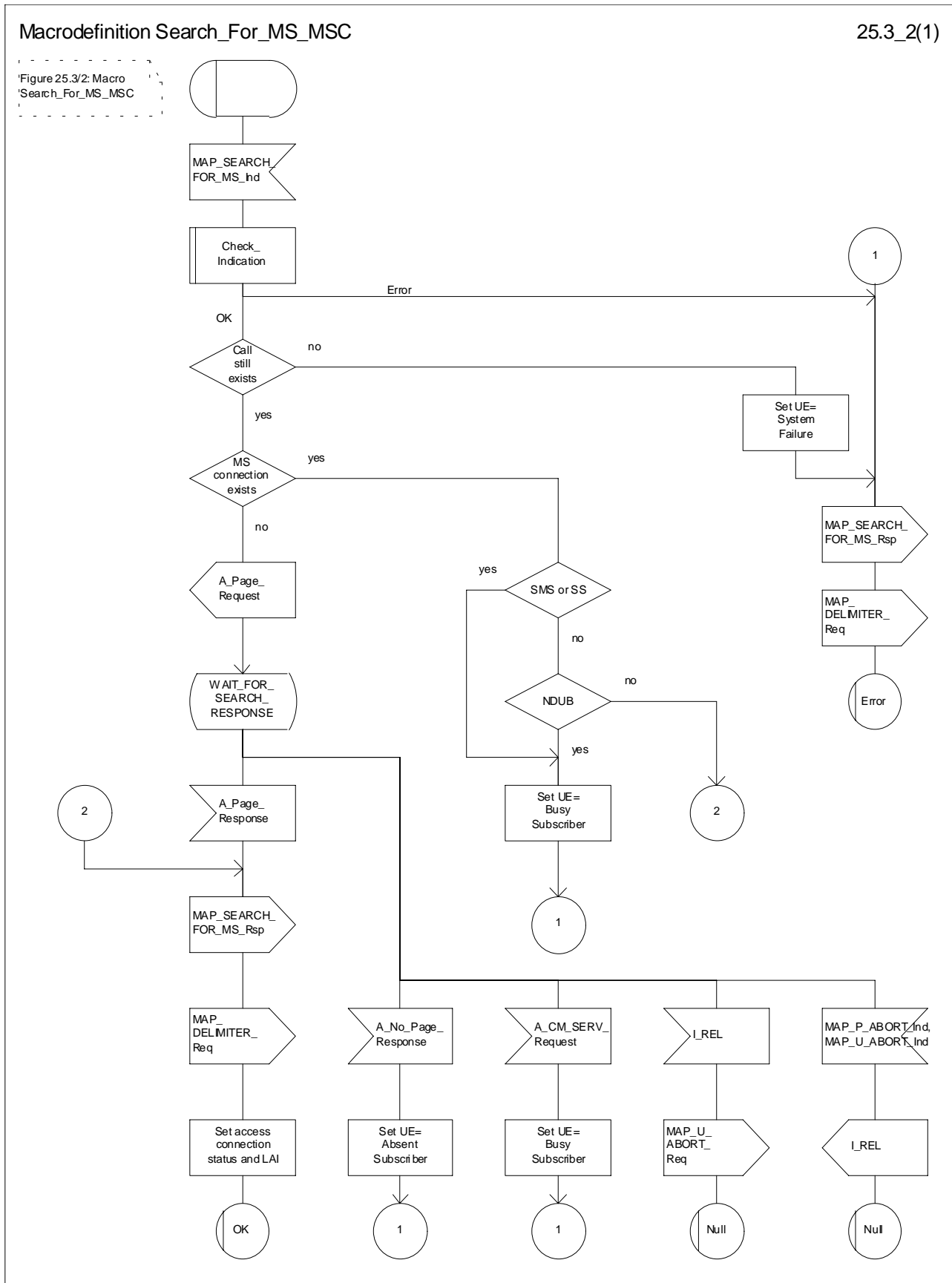


Figure 25.3/2: Macro Search_for_MS_MSC

25.4 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).

25.4.1 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 25.1), 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 25.8. 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 25.5. 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 25.9;
- 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 25.6. 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 25.6. 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).

Macrodefinition Process_Access_Request_MSC

25.4_1.1(3)

Figure 25.4/1: Macro for processing the access request in MSC

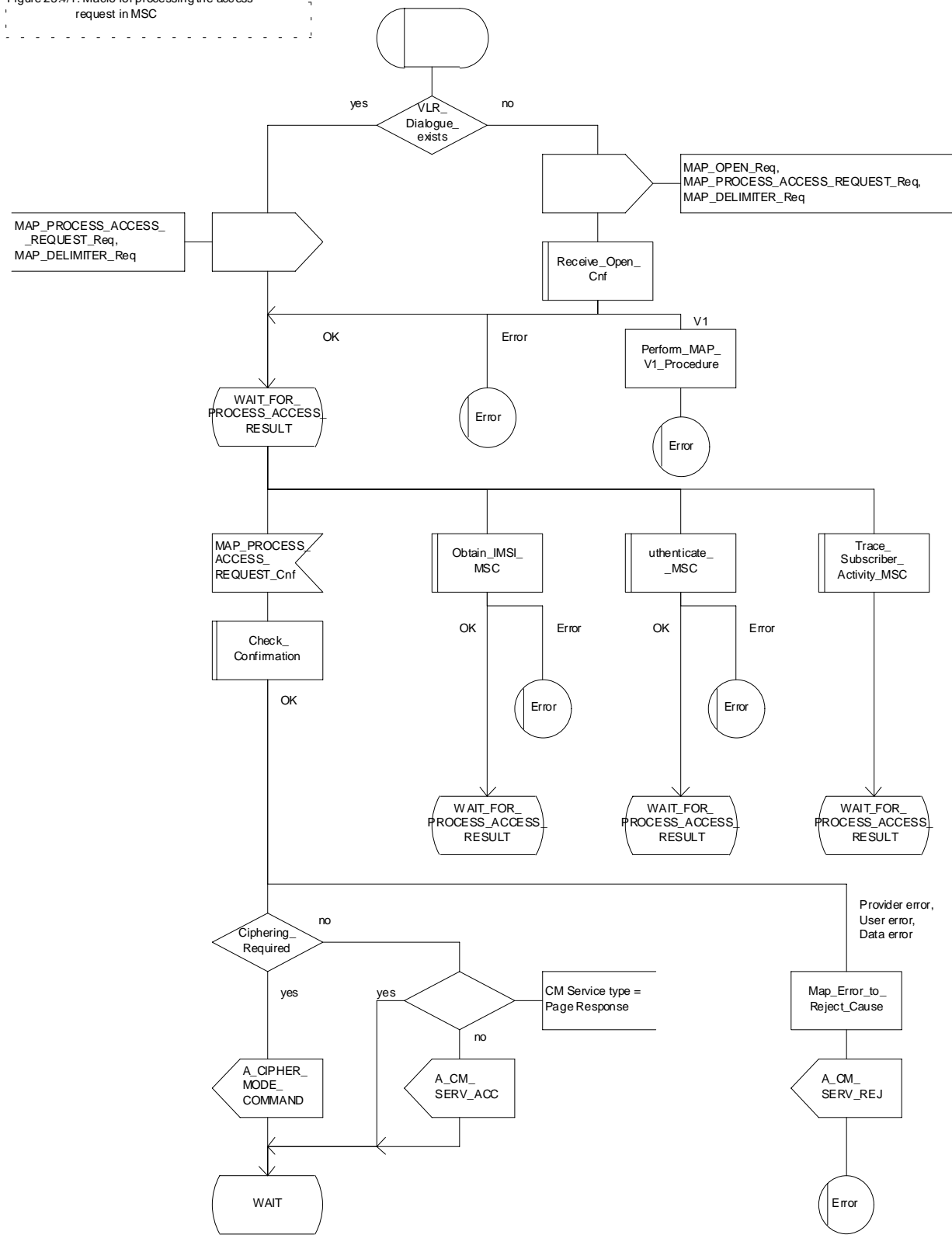


Figure 25.4/1 (sheet 1 of 3): Macro Process_Access_Request_MSC

Macrodefinition Process_Access_Request_MSC

25.4_1.2(3)

Figure 25.4/1: Macro for processing the access request in MSC

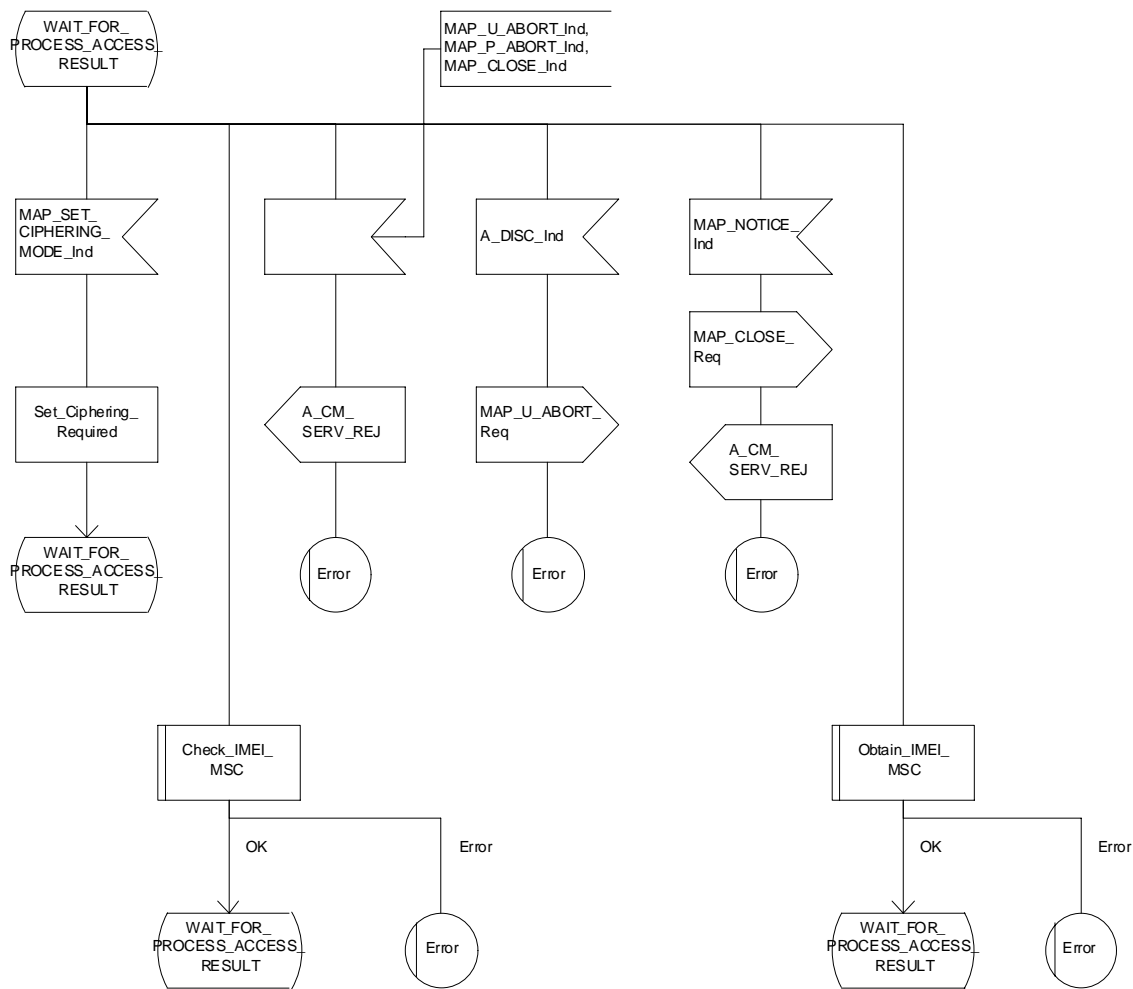


Figure 25.4/1 (sheet 2 of 3): Macro Process_Access_Request_MSC

Macrodefinition Process_Access_Request_MSC

25.4_1.3(3)

Figure 25.4/1: Macro for processing the access request in MSC

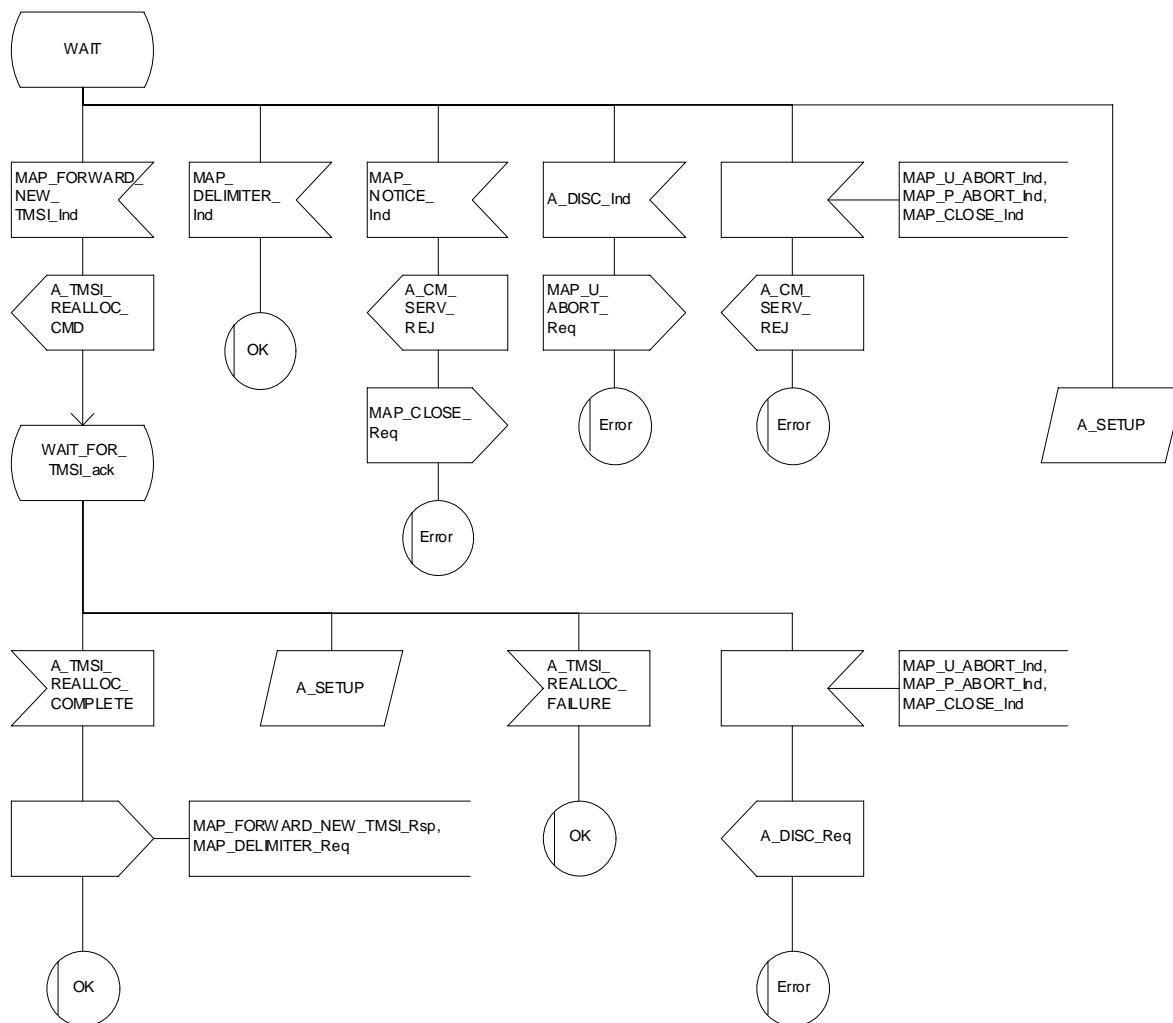


Figure 25.4/1 (sheet 3 of 3): Macro Process_Access_Request_MSC

25.4.2 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 25.2). 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 25.5), 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 25.8):

- 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 19.1.1.6).

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 25.10).

At next, tracing is invoked if required by the operator (macro Trace_Subscriber_Activity_VLR, see subclause 25.9). Thereafter,

if ciphering is not required, IMEI checking is invoked if required by the operator (see macro Check_IMEI_VLR defined in subclause 25.6).

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 25.6).

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.

25.4.3 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 25.8), 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.

Macrodefinition Process_Access_Request_VLR

25.4_2.1(3)

Figure 25.4/2:
Macro for processing
on access request
in VLR

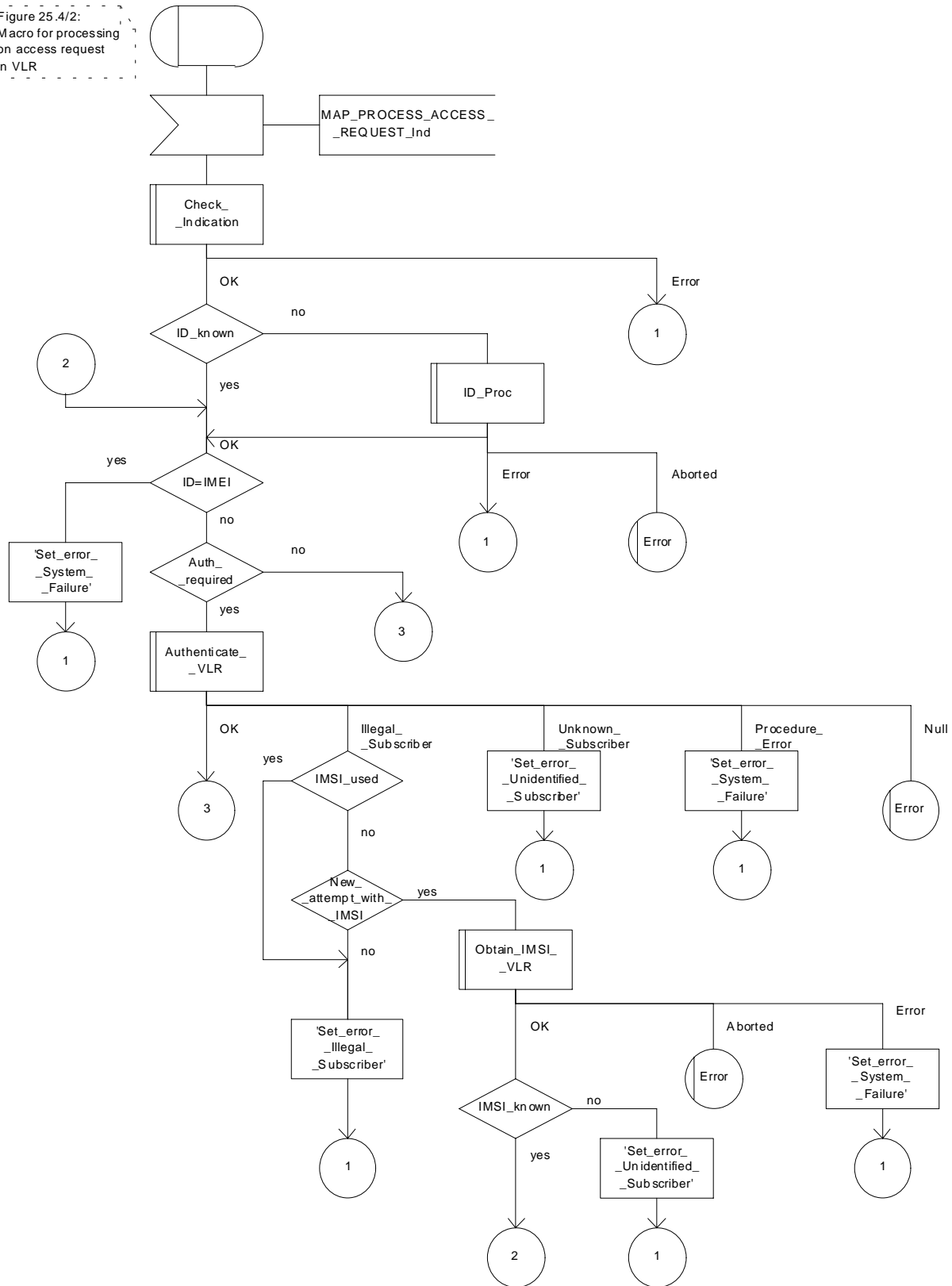


Figure 25.4/2 (sheet 1 of 3): Macro Process_Access_Request_VLR

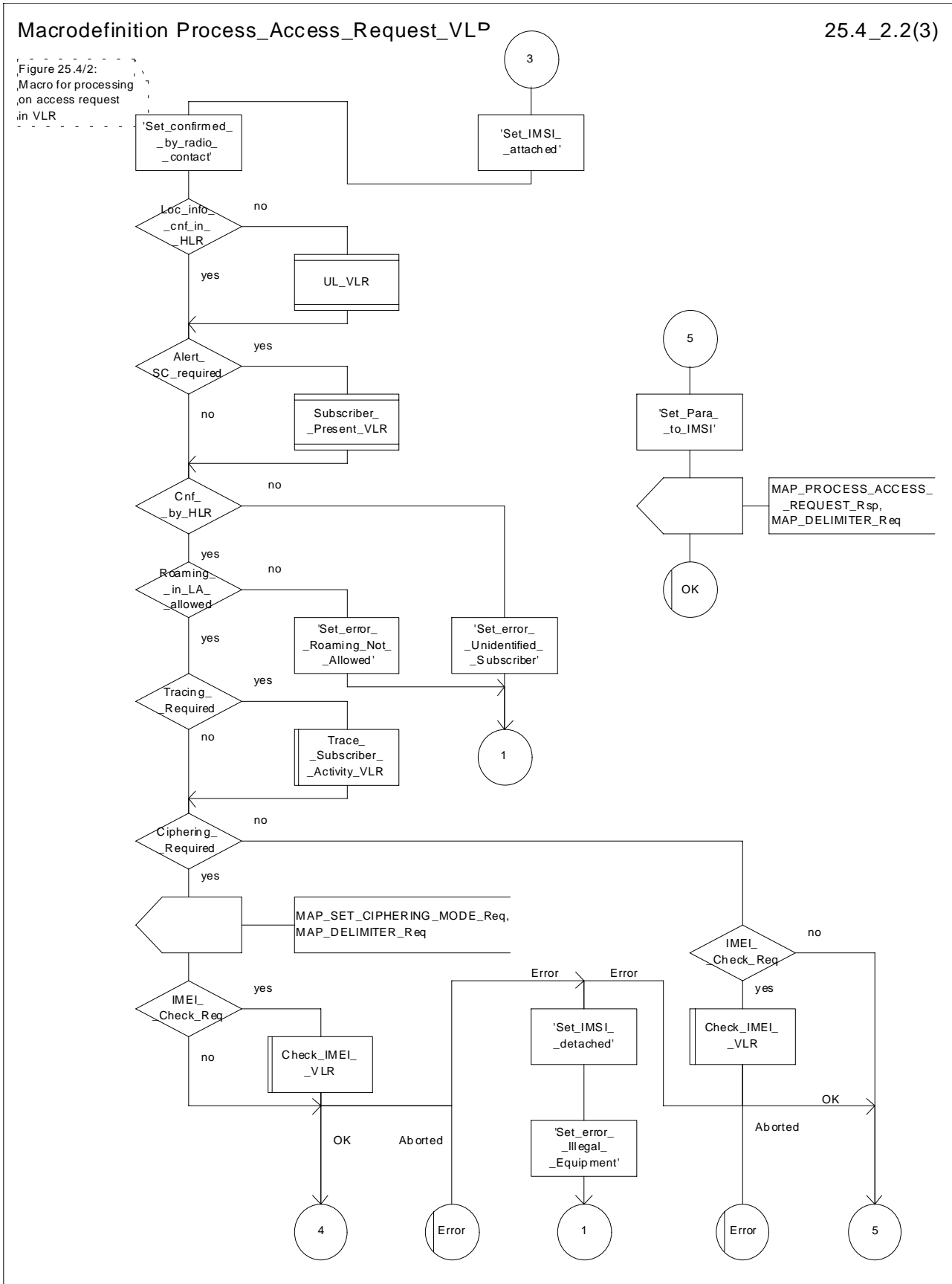


Figure 25.4/2 (sheet 2 of 3): Macro Process_Access_Request_VLR

Macrodefinition Process_Access_Request_VLR

25.4_2.3(3)

Figure 25.4/2:
Macro for processing
on access request
in VLR

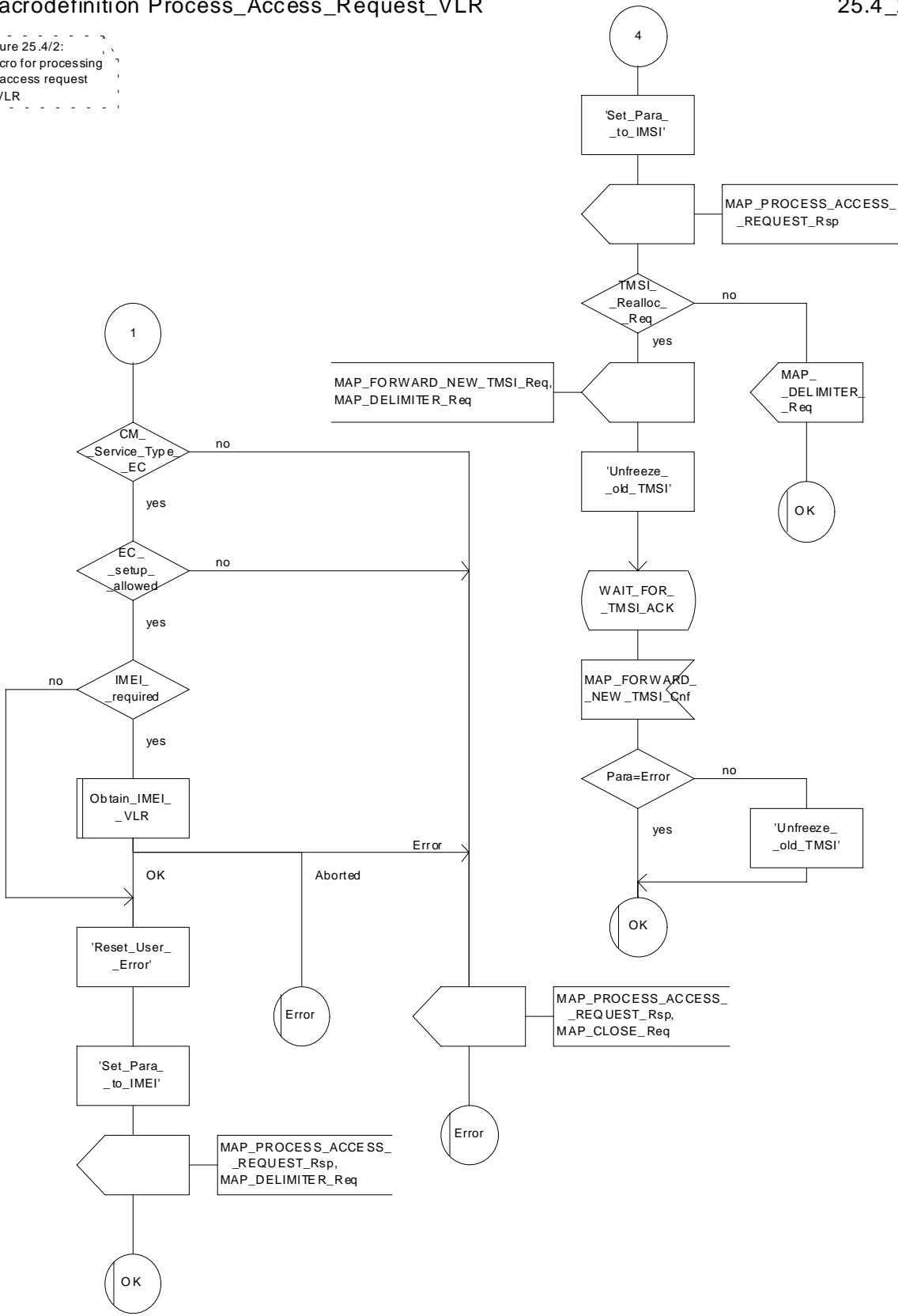


Figure 25.4/2 (sheet 3 of 3): Macro Process_Access_Request_VLR

Macrodefinition ID_Proc_VLR

25.4_3(1)

Figure 25.4/3

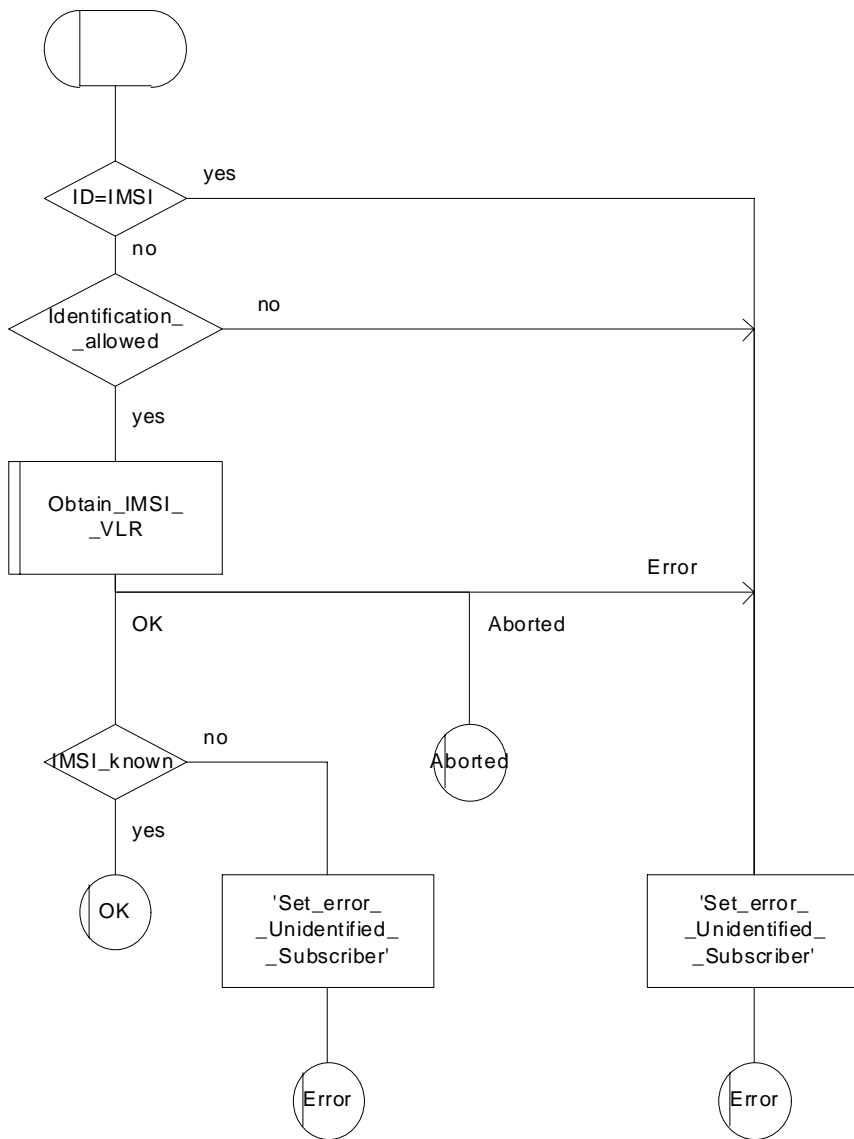


Figure 25.4/3: Macro ID_Proc_VLR

25.5 Authentication macros and processes

The following macros are used in the GSM network in order to enable authentication of a mobile subscriber.

25.5.1 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 25.5/1.

25.5.2 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 25.5/2.

25.5.3 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 25.5/3.

25.5.4 Macro Obtain_Authent_Para_VLR

This macro is used by the VLR to request authentication vectors 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 or 2 dialogue is to be used, the VLR performs the equivalent MAP version 1 or 2 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 returned by the MAP version 1 or 2 dialogue, then the "Unknown Subscriber" exit is used.

In a MAP version 3 dialogue a (possibly empty) set of authentication vectors may be received from the HLR followed by a MAP_CLOSE_Indication or by a MAP_DELIMITER_Indication. If a MAP_DELIMITER_Indication is received, the VLR may request additional authentication vectors from the HLR by sending a new MAP_SEND_AUTHENTICATION_INFO_Request. If a MAP_CLOSE_Indication is received, and authentication vectors have been received during the dialogue, then the "OK" exit is used. If no authentication vectors have been received during the dialogue, the VLR checks whether old GSM Triplets are available and can be re-used. If so, the "OK" exit is used, otherwise the "Procedure Error" exit is used. Note that re-use of old UMTS Quintuplets is not allowed.

If in a MAP version 3 dialogue an "Unknown Subscriber" error is received, then the "Unknown Subscriber" exit is used. If other errors are received, the VLR checks whether old GSM Triplets are available and can be re-used. If so, the "OK" exit is used, otherwise the "Procedure Error" exit is used. Note that re-use of old UMTS Quintuplets is not allowed.

- 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 (GSM triplets) can be re-used. If old parameters cannot be re-used the macro takes the "Procedure Error" exit; otherwise it takes the "OK" exit; note that re-use of old UMTS Quintuplets is not allowed;
- 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 (GSM triplets) can be re-used. If old parameters cannot be re-used the macro takes the "Procedure Error" exit; otherwise it takes the "OK" exit; note that re-use of old UMTS Quintuplets is not allowed.

The macro is described in figure 25.5/4.

25.5.5 Process Obtain_Auth_Sets_HLR

Opening of the dialogue is described in the macro Receive_Open_Ind in subclause 25.1, with outcomes:

- reversion to version one or two procedure;
- procedure termination; or
- dialogue acceptance, with proceeding as below.

This process is used by the HLR to obtain authentication vectors from the AuC, upon request from the VLR or from the SGSN. 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 or to the SGSN in the MAP_SEND_AUTHENTICATION_INFO response. If no errors are detected, authentication vectors are fetched from the AuC. Further details are found in GSM 03.20;
- if errors are detected they are reported to the VLR or to the SGSN in the MAP_SEND_AUTHENTICATION_INFO response. Otherwise the authentication vectors are returned.
- if segmentation of the response message is required and allowed, a MAP_SEND_AUTHENTICATION_INFO_response, containing at least one authentication vector, followed by a MAP_DELIMITER_request is returned to the VLR or SGSN, the remaining authentication vectors are stored and the HLR waits for a new service indication from the VLR or SGSN.

The process is described in figure 25.5/5.

Macrodefinition Authenticate_MSC

25.5_1(1)

Figure 25.5/1: Authentication macro in the MSC, relaying authentication indication from the VLR to the MS, and relaying the confirmation from the MSC to the VLR

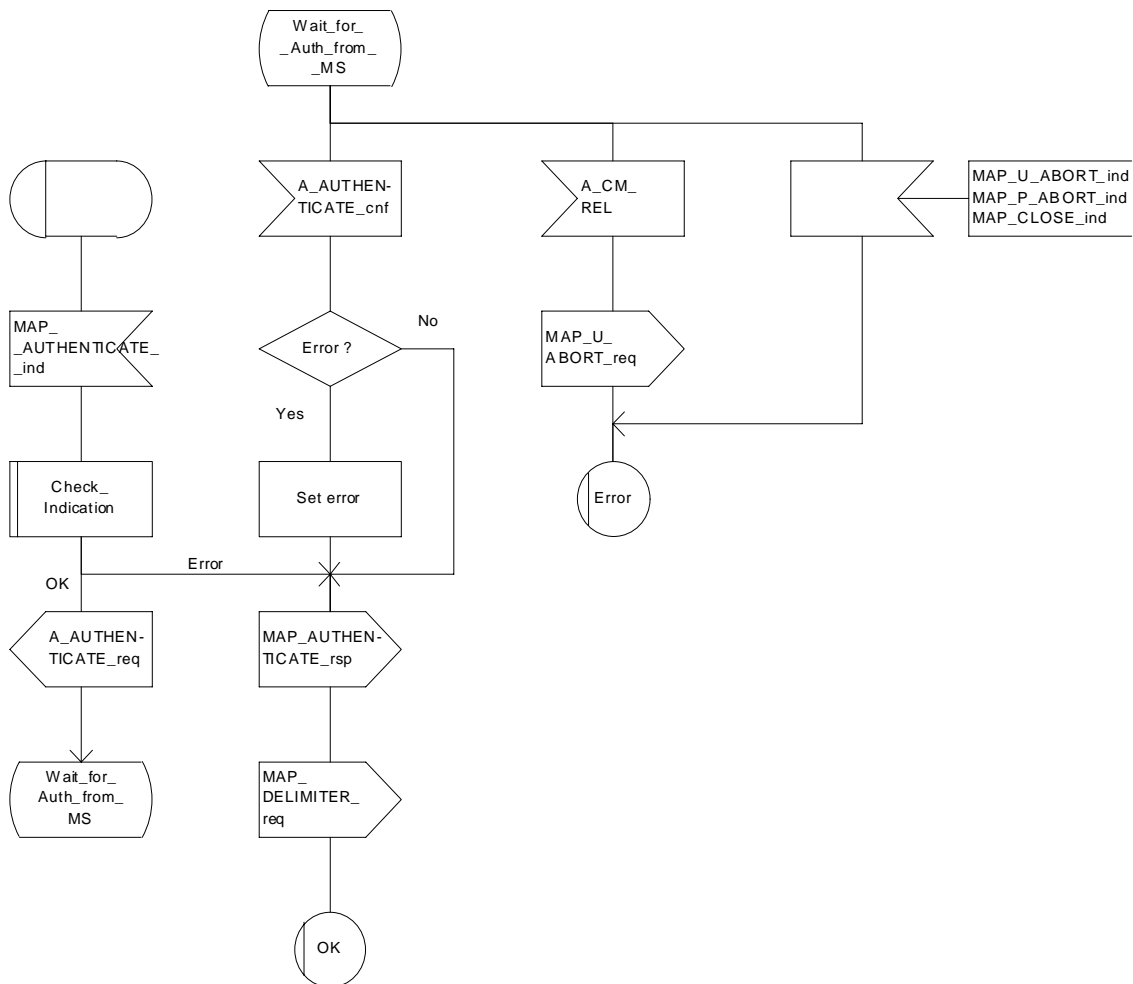


Figure 25.5/1: Macro Authenticate_MSC

Macrodefinition Authenticate_VLR

215_2(1)

Figure 25.5/2: Authentication macro in the VLR, controlling the authentication procedure towards the MSC/MS and obtaining authentication vectors from the HLR as applicable.

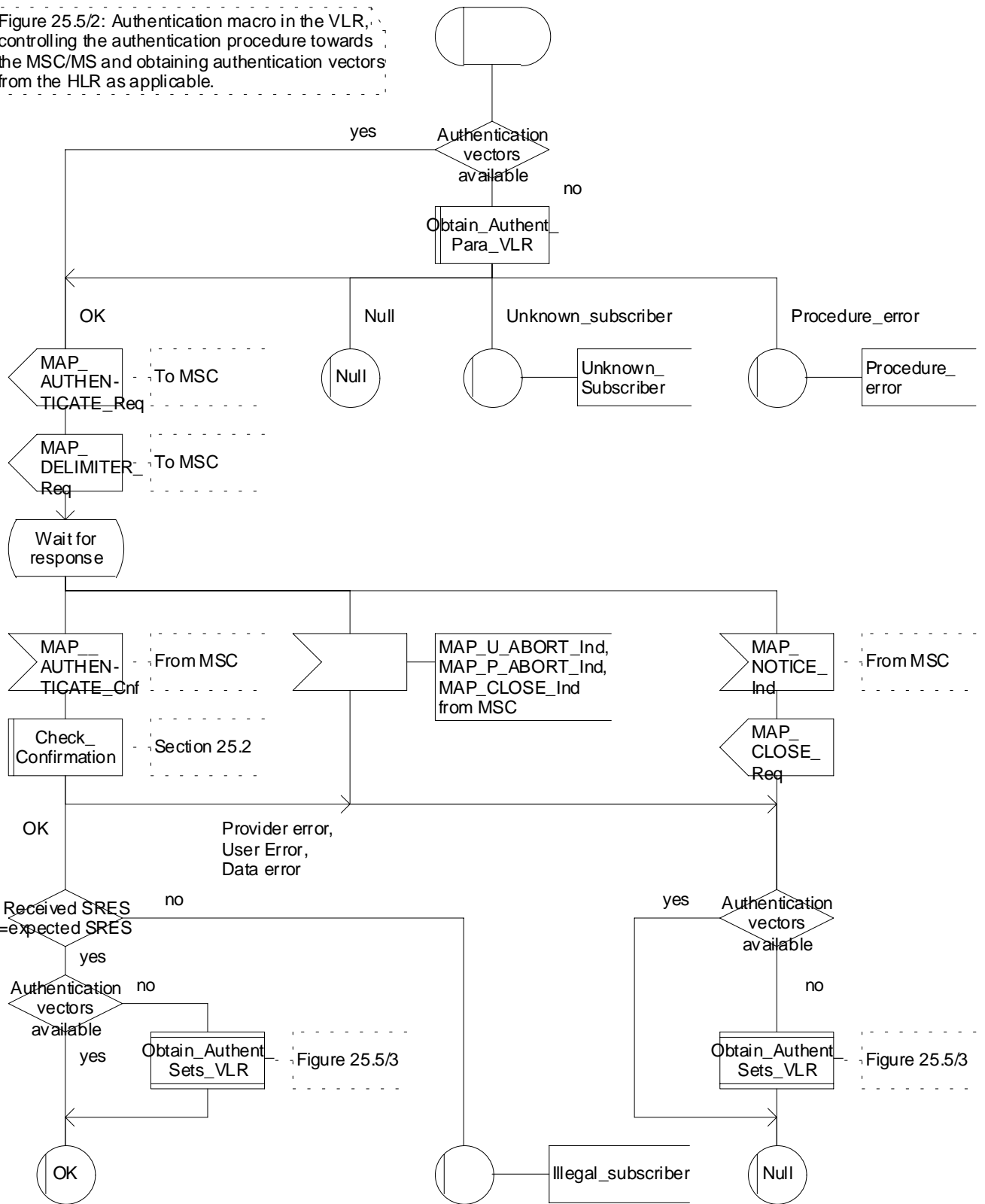


Figure 25.5/2: Macro Authenticate_VLR

Process Obtain_Authent_Sets_VLR

25.5_3(1)

Figure 25.5/3: Process to obtain authentication sets from the HLR to the VLR

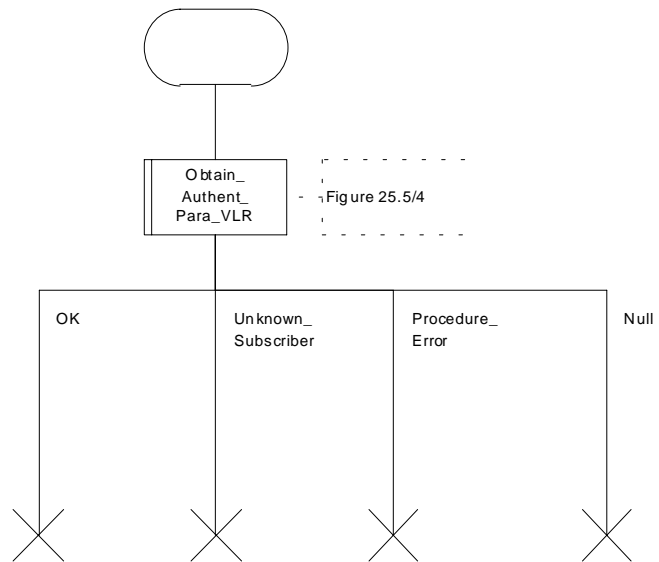


Figure 25.5/3: Process Obtain_Authentication_Sets_VLR

Macrodefinition OBTAIN_AUTHENT_PARA_VLR

1(3)

Figure 25.5/4: Macro to obtain authentication parameters from the HLR to the VLR

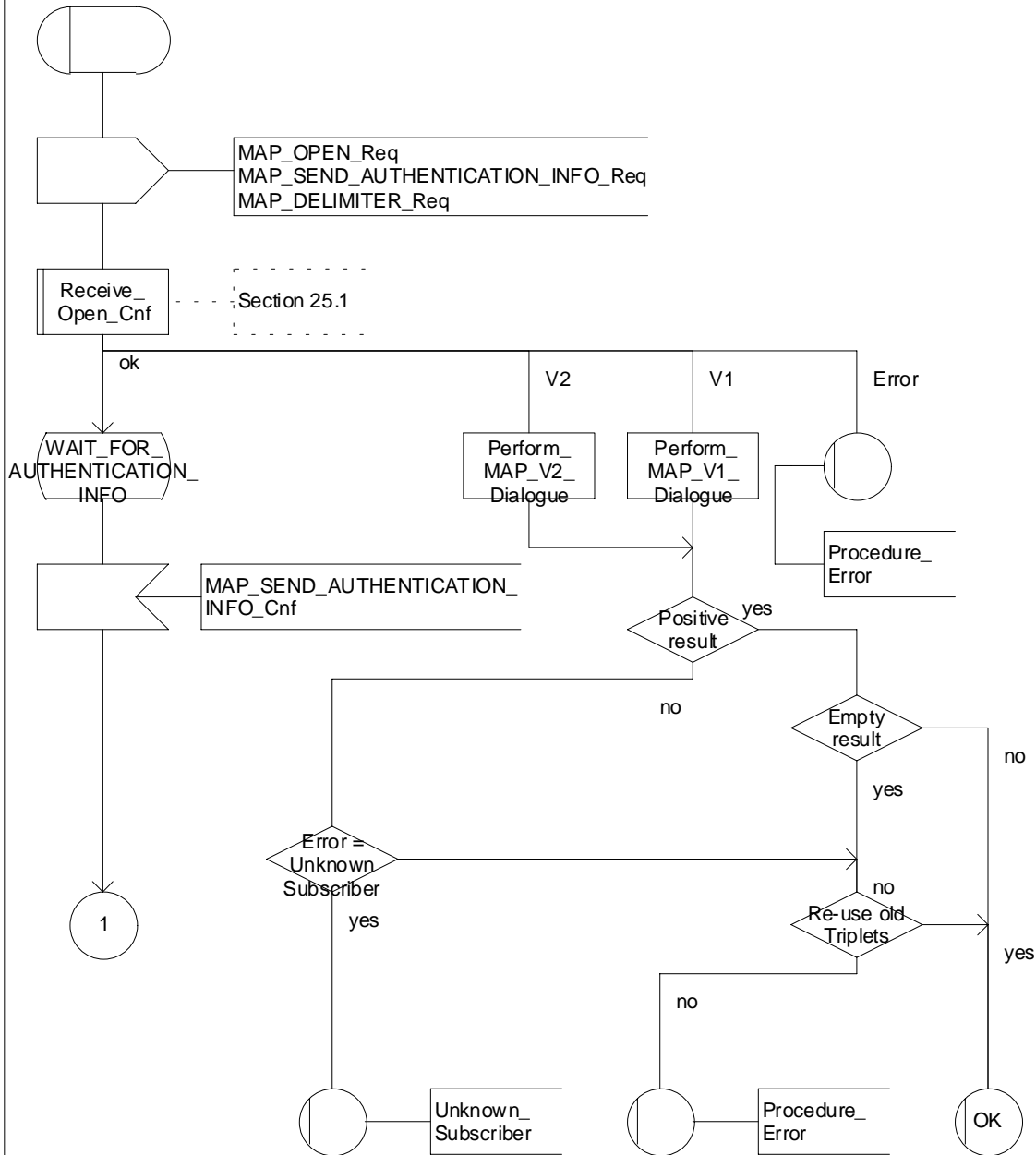


Figure 25.5/4 (sheet 1 of 3): Macro Obtain_Authent_Para_VLR

Macrodefinition OBTAIN_AUTHENT_PARA_VLR

2(3)

Figure 25.5/4: Macro to obtain authentication parameters from the HLR to the VLR

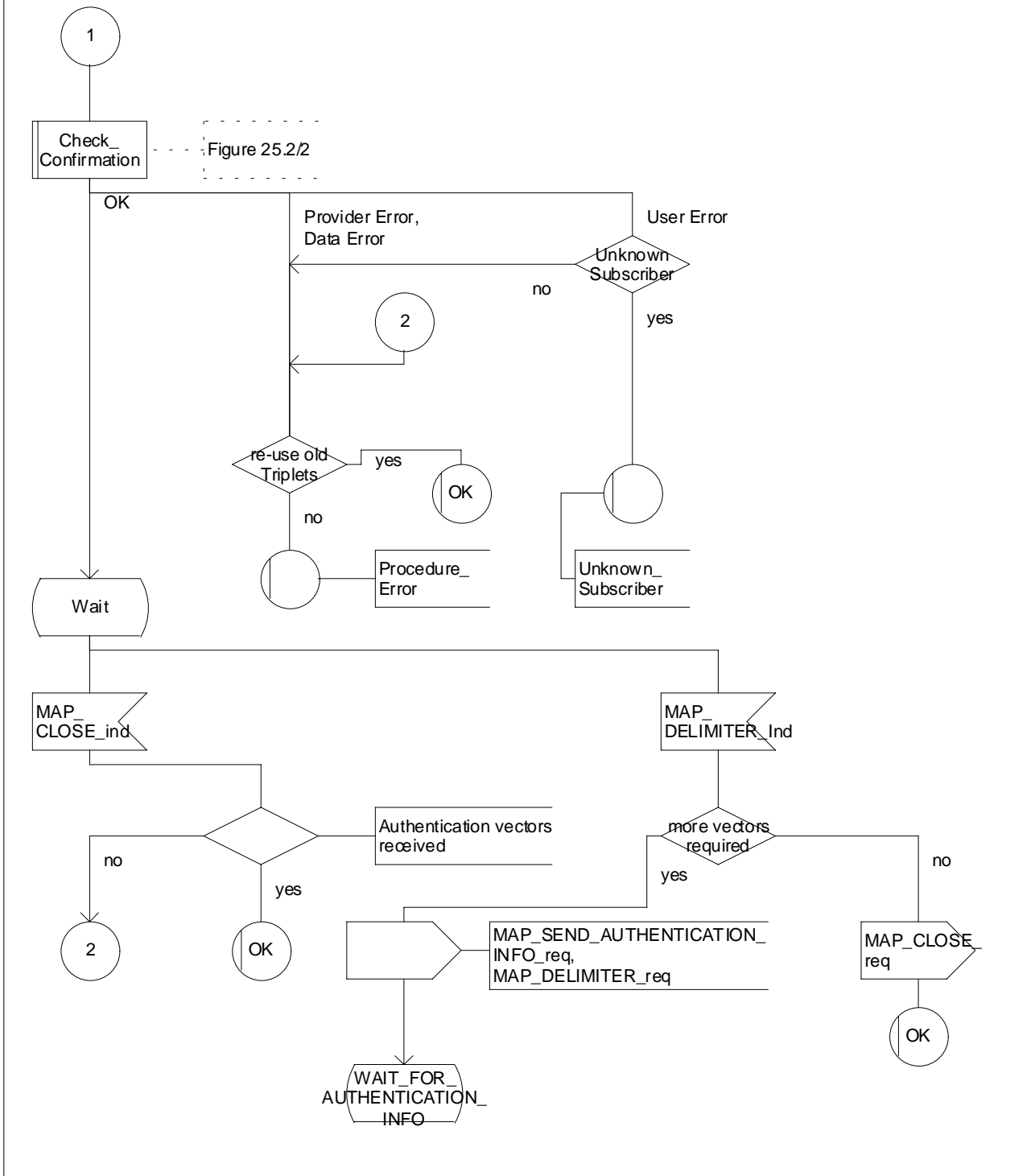


Figure 25.5/4 (sheet 2 of 3): Macro Obtain_Authent_Para_VLR

Macrodefinition OBTAIN_AUTHENT_PARA_VLR

3(3)

Figure 25.5/4: Macro to obtain authentication parameters from the HLR to the VLR

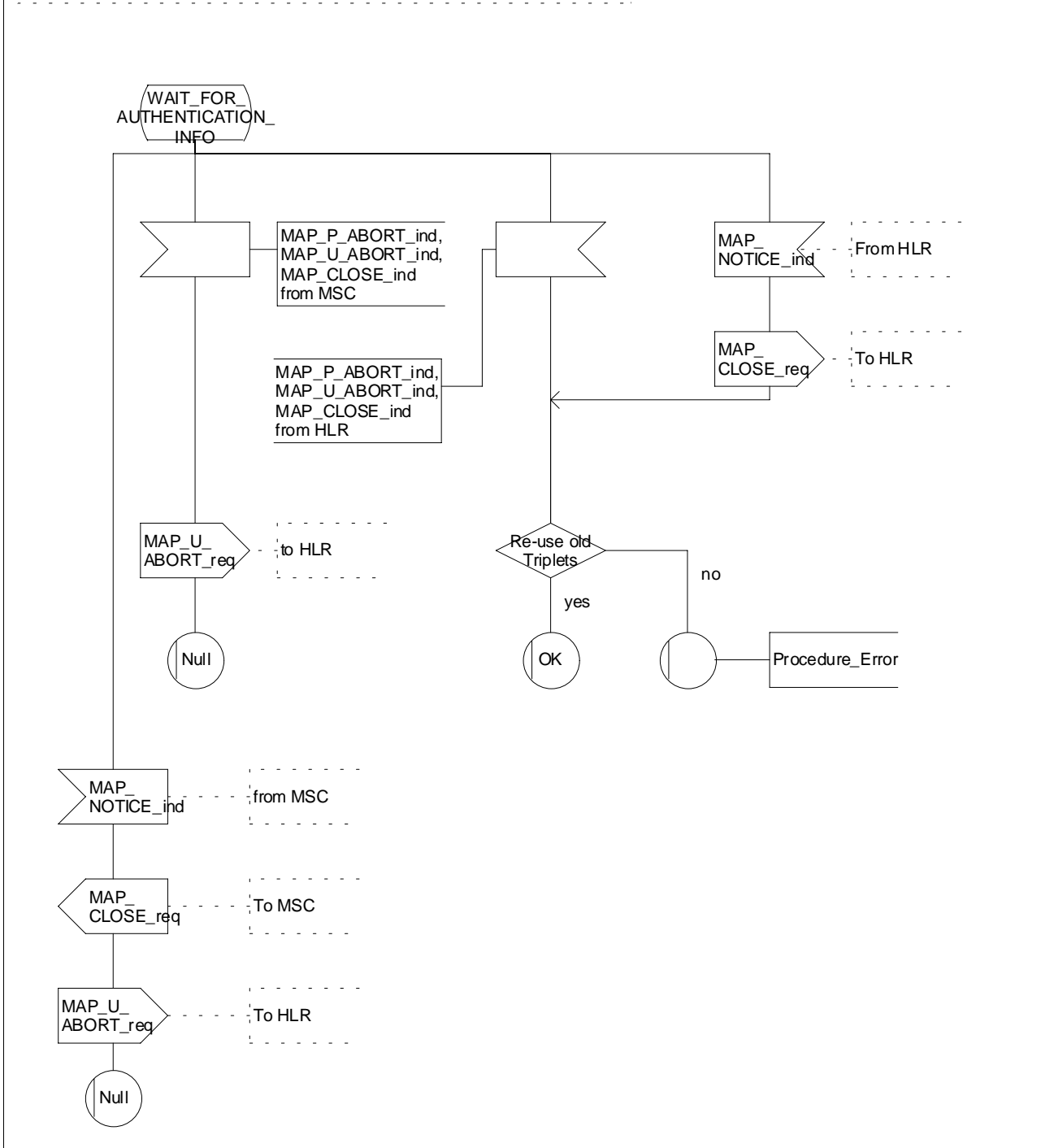


Figure 25.5/4 (sheet 3 of 3): Macro Obtain_Authent_Para_VLR

Process Obtain_Auth_Sets_HLR

1(2)

Figure 25.5/5: Process in the HLR to obtain authentication sets from the AuC and relay them to the VLR

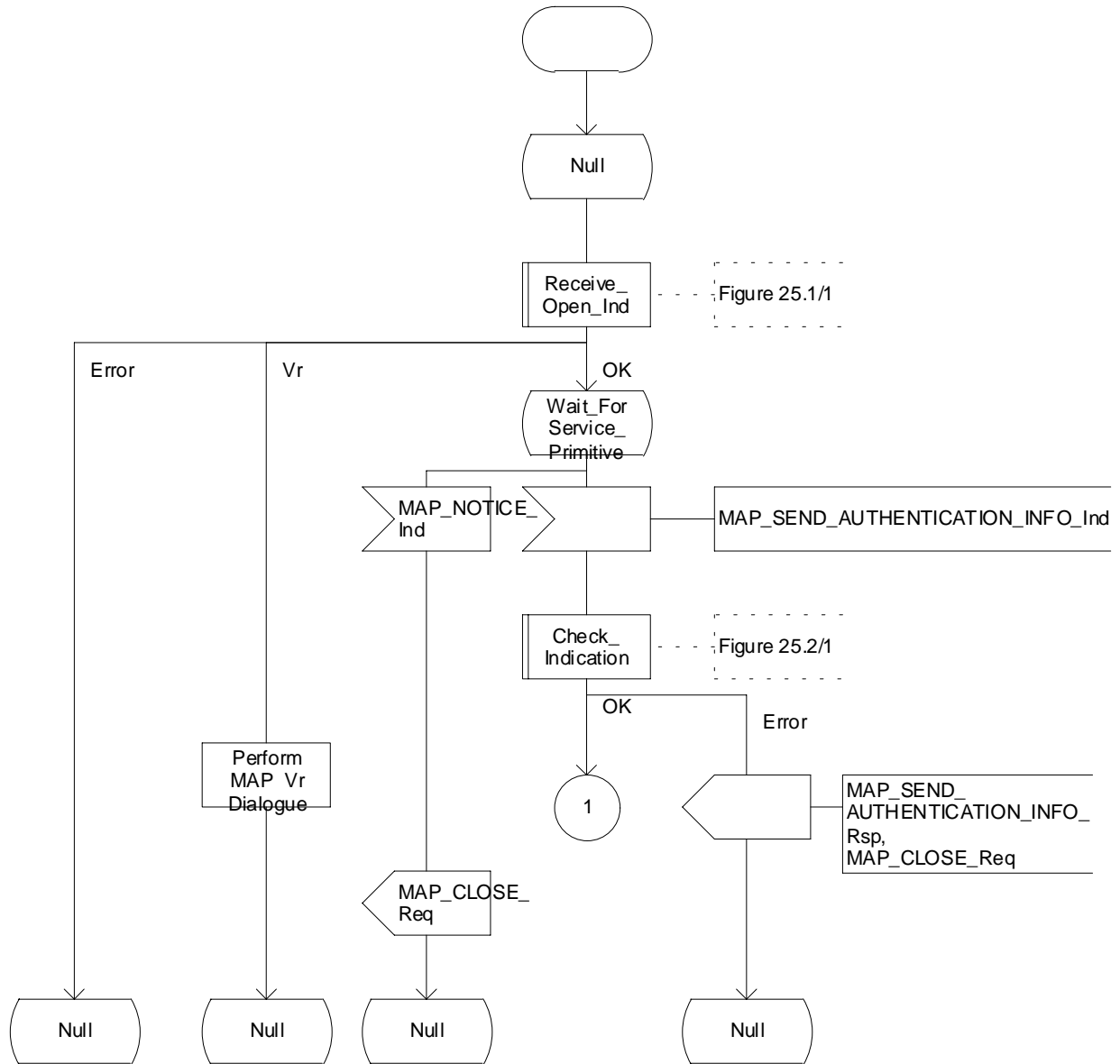


Figure 25.5/5 (sheet 1 of 2): Process Obtain_Auth_Sets_HLR

Process Obtain_Auth_Sets_HLR

2(

Figure 25.5/5: Process in the HLR to obtain authentication sets from the AuC and relay them to the VLR

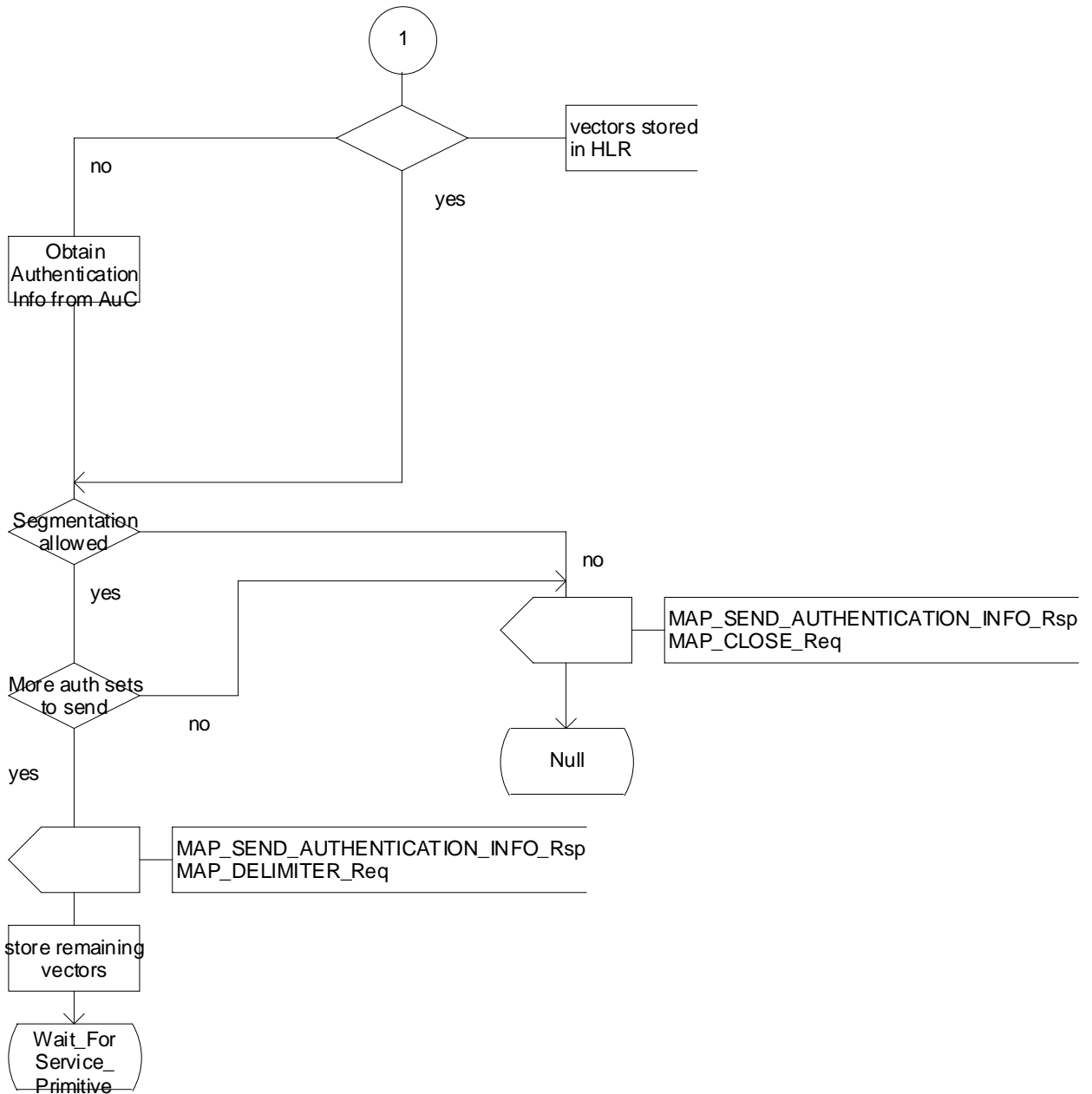


Figure 25.5/5 (sheet 2 of 2): Process Obtain_Auth_Sets_HLR

25.5.6 Process Obtain_Authent_Para_SGSN

For authentication procedure description see GSM 03.60 and GSM 04.08.

This Process is used by the SGSN to request authentication vectors from the HLR. The Process 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 or 2 dialogue is to be used, the SGSN performs the equivalent MAP version 1 or 2 dialogue. which can return a positive result containing authentication sets, an empty positive result, or an error;
- if the dialogue opening fails, the Authentication Parameters negative response with appropriate error is sent to the requesting process. Otherwise, the SGSN waits for the response from the HLR;
- if a MAP_SEND_AUTHENTICATION_INFO confirmation is received from the HLR, the SGSN 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 SGSN may re-use old triplets, if allowed by the PLMN operator.

If the SGSN cannot re-use old triplets (or no such triplets are available) then the the Authentication Parameters negative response with appropriate error is sent to the requesting process.

If the outcome was successful or re-use of old parameters in the SGSN is allowed, then the Authentication Parameters response is sent to the requesting process

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 appropriate error is sent to the requesting process in the Authentication Parameters negative response

In a MAP version 3 dialogue a (possibly empty) set of authentication vectors may be received from the HLR followed by a MAP_CLOSE_Indication or by a MAP_DELIMITER_Indication. If a MAP_DELIMITER_Indication is received, the SGSN may request additional authentication vectors from the HLR by sending a new MAP_SEND_AUTHENTICATION_INFO_Request. If a MAP_CLOSE_Indication is received, and authentication vectors have been received during the dialogue, then the "OK" exit is used. If no authentication vectors have been received during the dialogue, the SGSN checks whether old GSM Triplets are available and can be re-used. If so, the "OK" exit is used, otherwise the "Procedure Error" exit is used. Note that re-use of old UMTS Quintuplets is not allowed.

If in a MAP version 3 dialogue an "Unknown Subscriber" error is received, then the "Unknown Subscriber" exit is used. If other errors are received, the SGSN checks whether old GSM Triplets are available and can be re-used. If so, the "OK" exit is used, otherwise the "Procedure Error" exit is used. Note that re-use of old UMTS Quintuplets is not allowed.

- if a MAP-U-ABORT, MAP_P_ABORT or unexpected MAP_CLOSE service indication is received from the HLR, then the SGSN checks whether old authentication parameters can be re-used. If old parameters cannot be re-used the Authentication Parameters negative response with appropriate error is sent to the requesting process.
- if a MAP_NOTICE service indication is received from the HLR, then the dialogue with the HLR is closed. The SGSN then checks whether old authentication parameters can be re-used. If old parameters cannot be re-used the process terminates and the Authentication Parameters negative response with appropriate error is sent to the requesting process; Otherwise the Authentication Parameters response is sent to requesting process.

The process is described in figure 25.5/6.

Process Obtain_Authent_Para_SGSN

1(2)

Figure 25.5/6: Process to obtain authentication parameters from the HLR to the SGSN

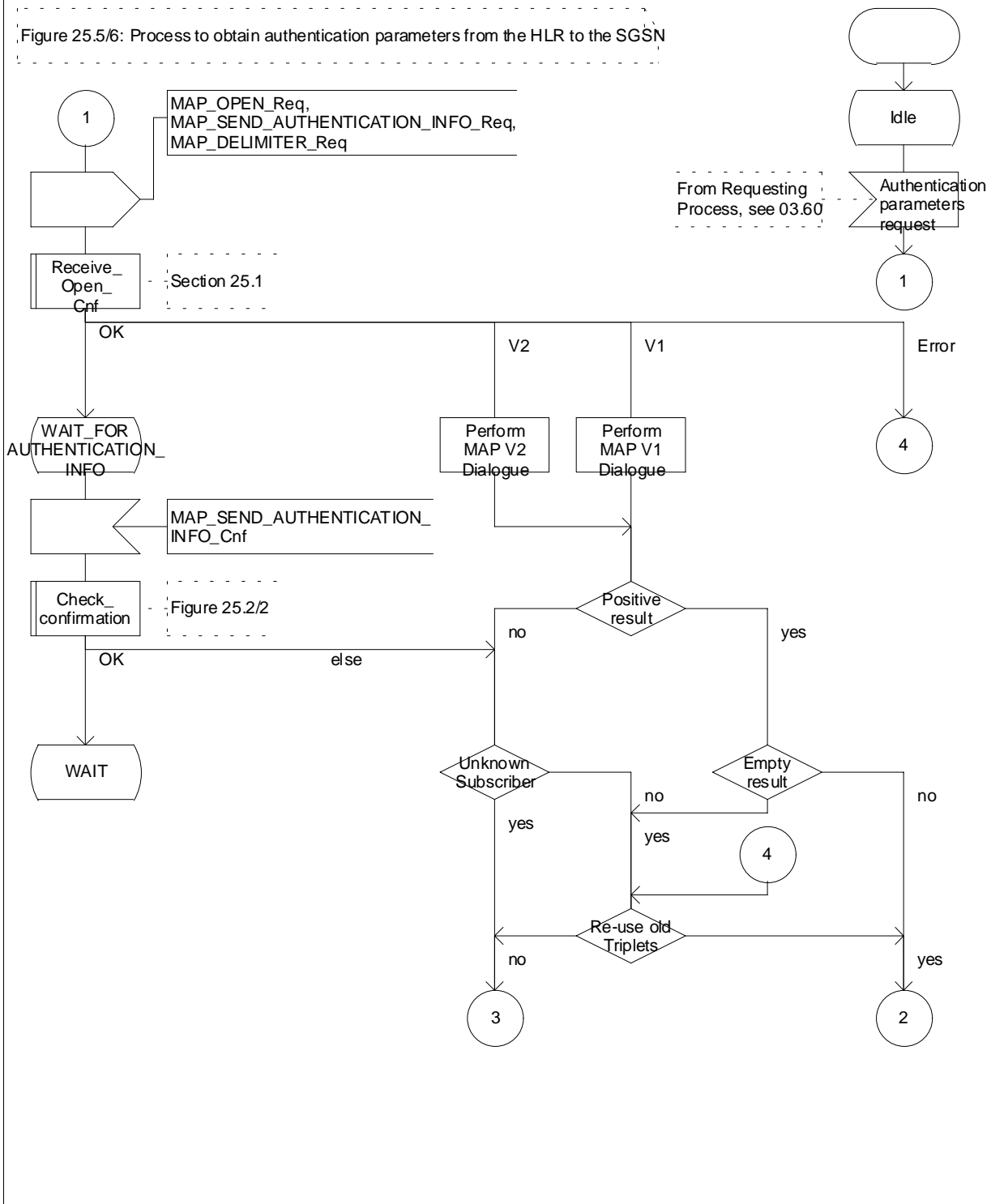


Figure 25.5/6 (sheet 1 of 2): Process Obtain_Authen_Para_SGSN

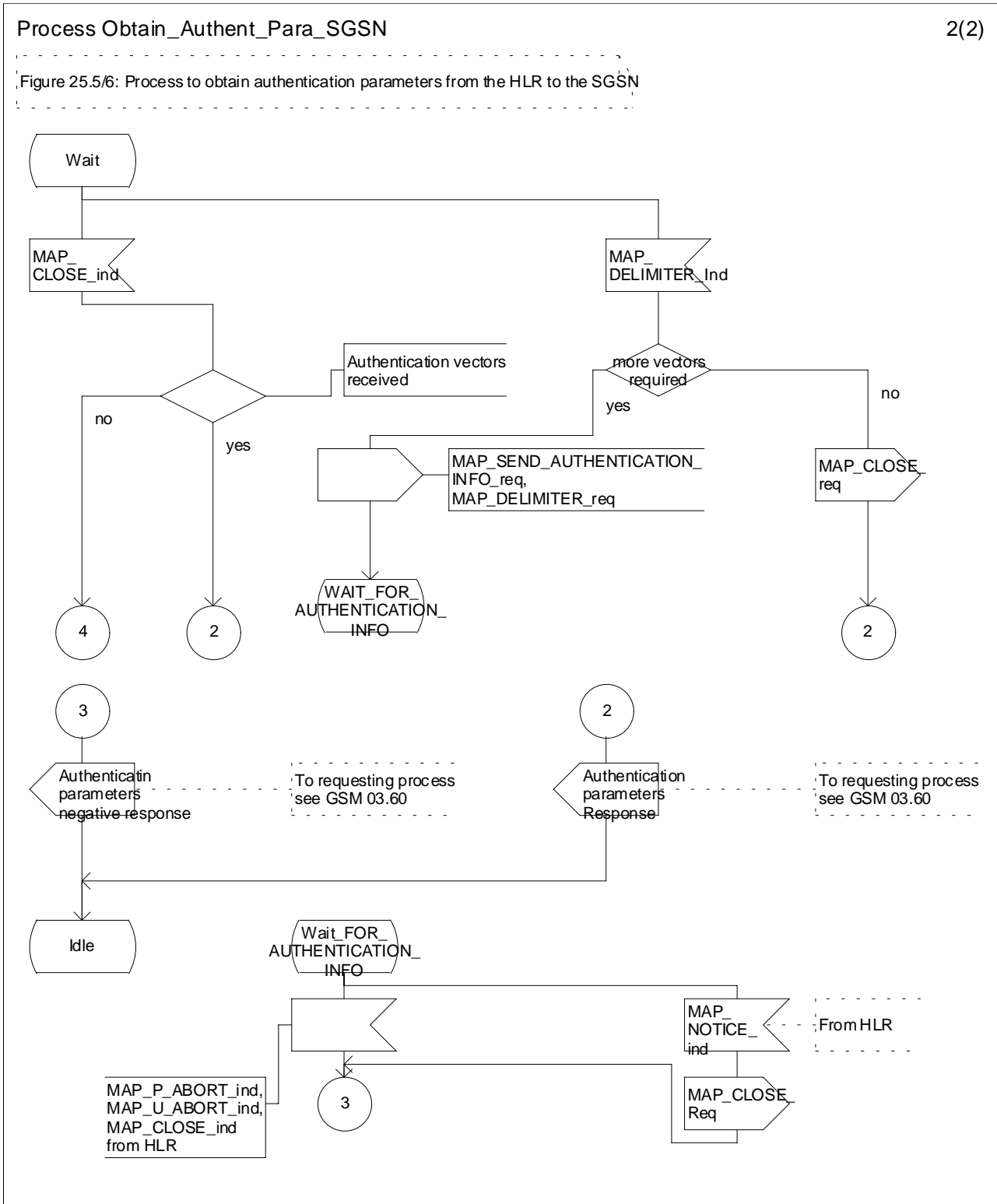


Figure 25.5/6 (sheet 2 of 2): Process Obtain_Authent_Para_SGSN

25.6 IMEI Handling Macros

The following macros are used in the GSM network in order to enable handling and checking of the mobile equipment identity.

25.6.1 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 25.6/1.

25.6.2 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 25.6/2.

25.6.3 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 or from the SGSN. The process acts as follows:

- a MAP_OPEN service indication is received (macro Receive_Open_Ind, subclause 25.1.1). 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 or to the SGSN 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 or to the SGSN 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 or from the SGSN at any time during this process, the process in the EIR terminates.

The process is described in figure 25.6/3.

25.6.4 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 25.6/4.

25.6.5 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 25.6/5.

Macrodefinition Check_IMEI_MSC

25.6_1.1(2)

Figure 25.6/1: Check IMEI macro in the MSC, relaying the IMEI check indication from the VLR to the MSC and relaying the confirmation from the EIR to the VLR

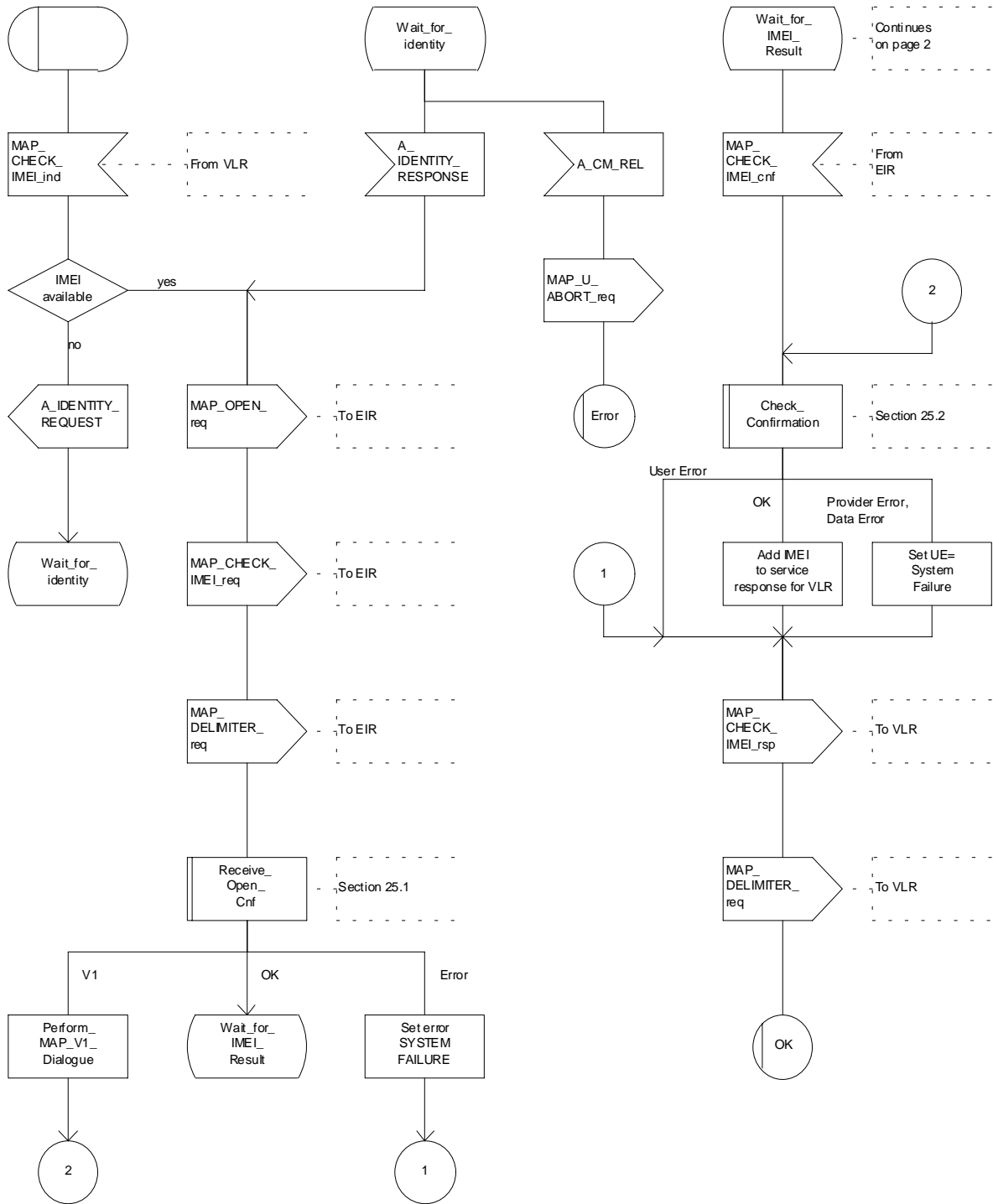


Figure 25.6/1 (sheet 1 of 2): Process Check_IMEI_MSC

Macrodefinition Check_IMEI_MSC

25.6_1.2(2)

Figure 25.6/1: Check IMEI macro in the MSC, relaying the IMEI check indication from the VLR to the MSC and relaying the confirmation from the EIR to the VLR

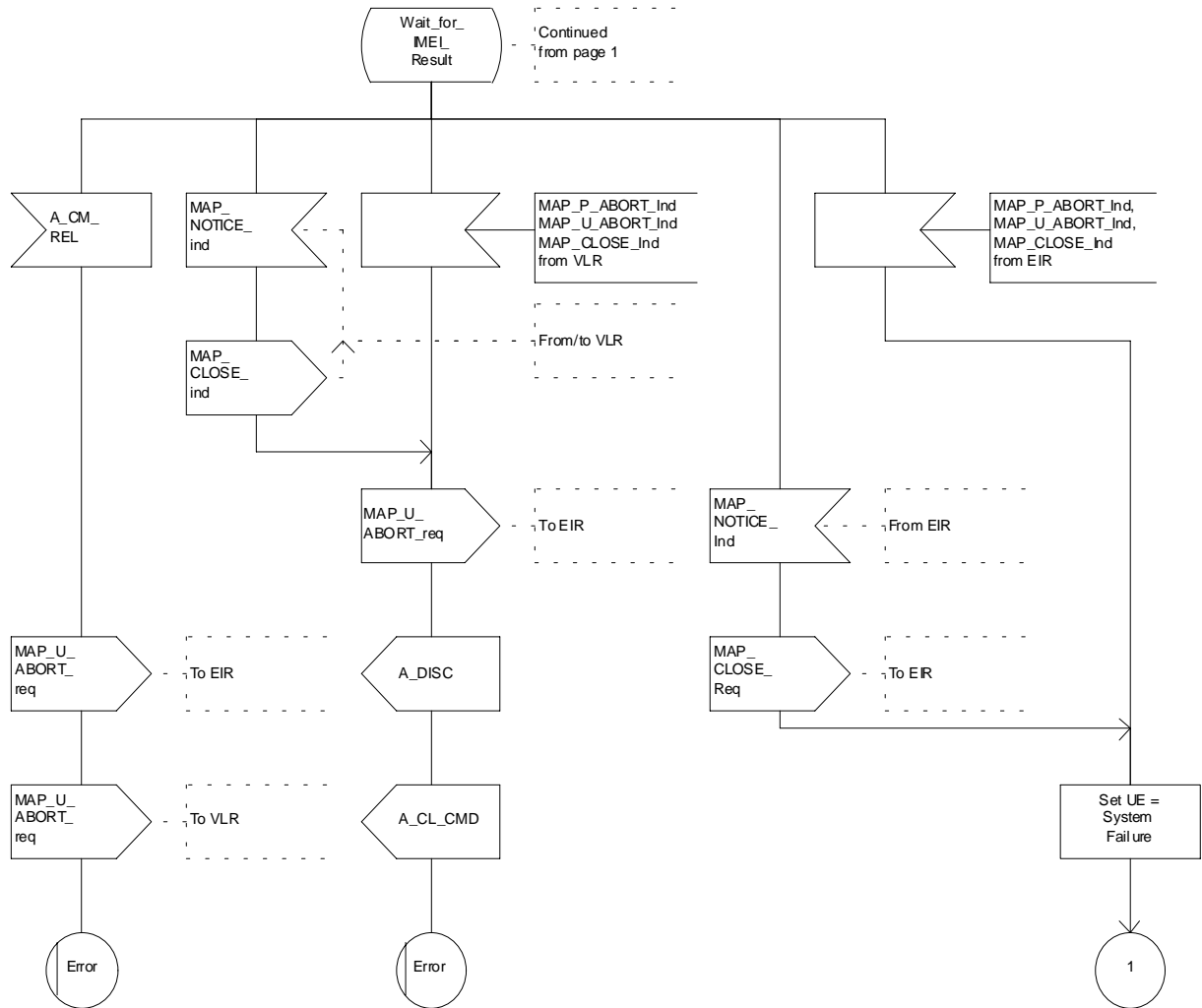


Figure 25.6/1 (sheet 2 of 2): Process Check_IMEI_MSC

Macrodefinition Check_IMEI_VLR

25.6_2(1)

Figure 25.6/2: Check IMEI macro in the VLR, containing the request towards the MSC/EIR

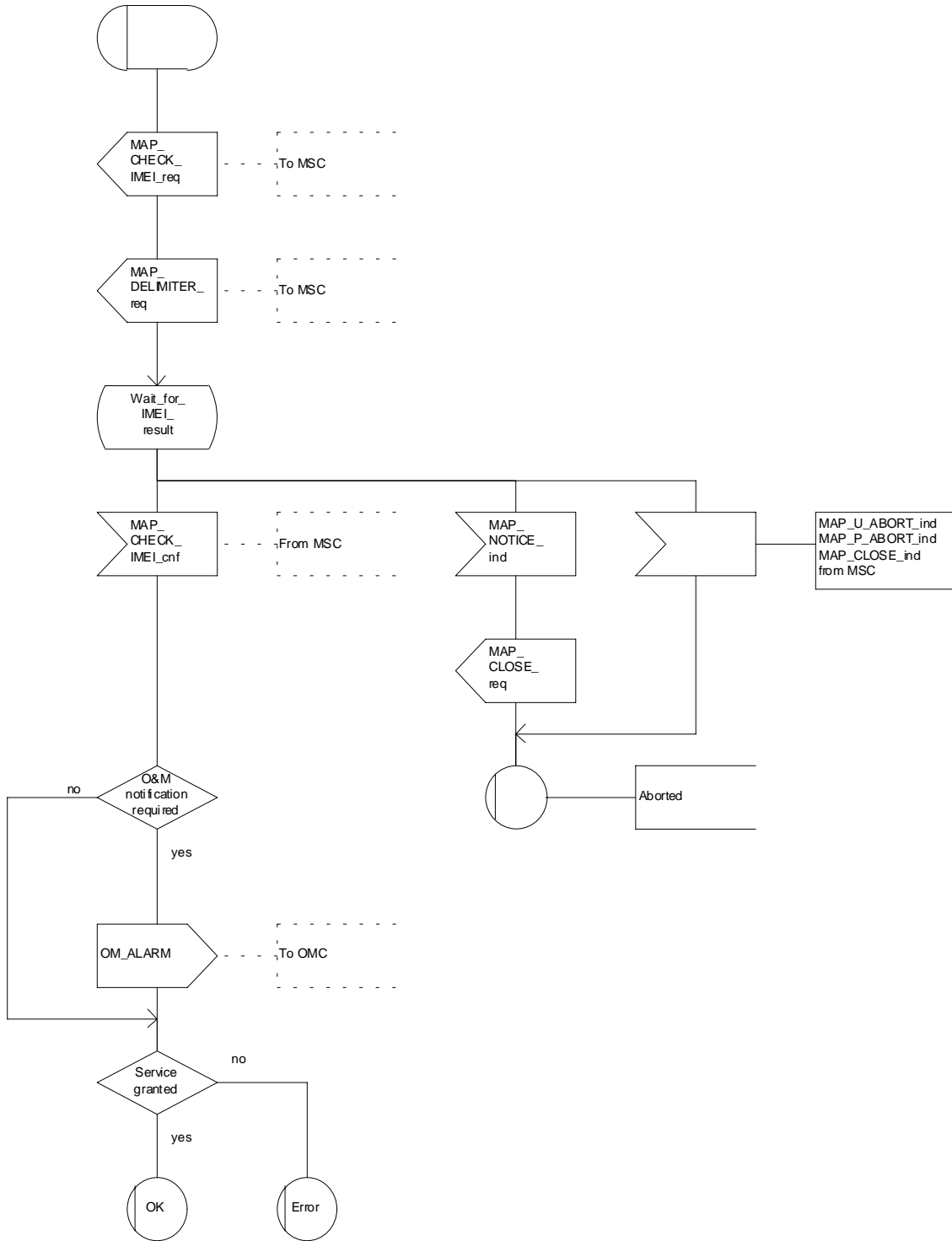


Figure 25.6/2: Process Check_IMEI_VLR

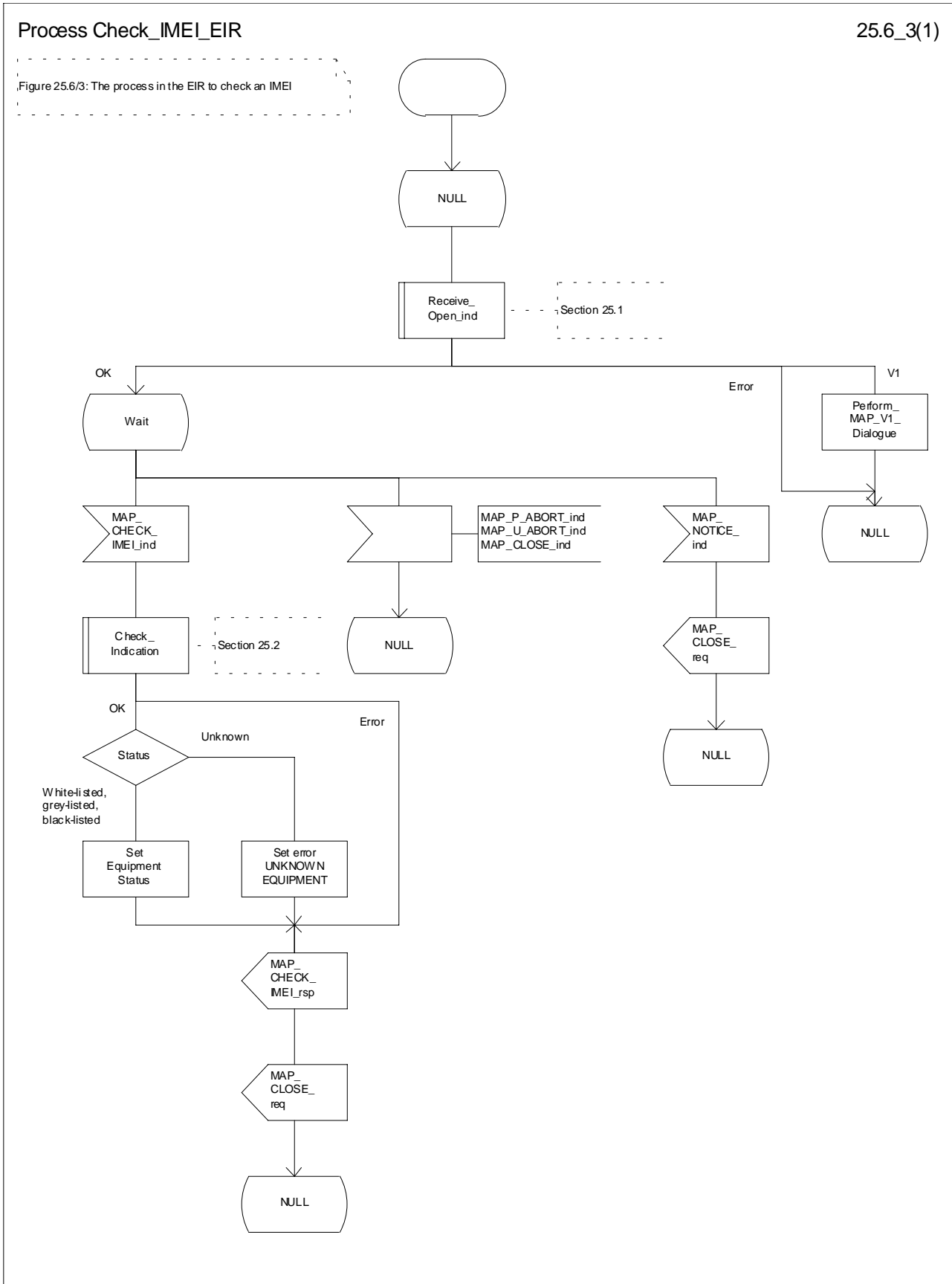


Figure 25.6/3: Process Check_IMEI_EIR

Macrodefinition Obtain_IMEI_MSC

25.6_4(1)

Figure 25.6/4: Obtain IMEI macro in the MSC, receiving the Obtain_IMEI indication from the VLR to the MSC and returning the confirmation to the VLR

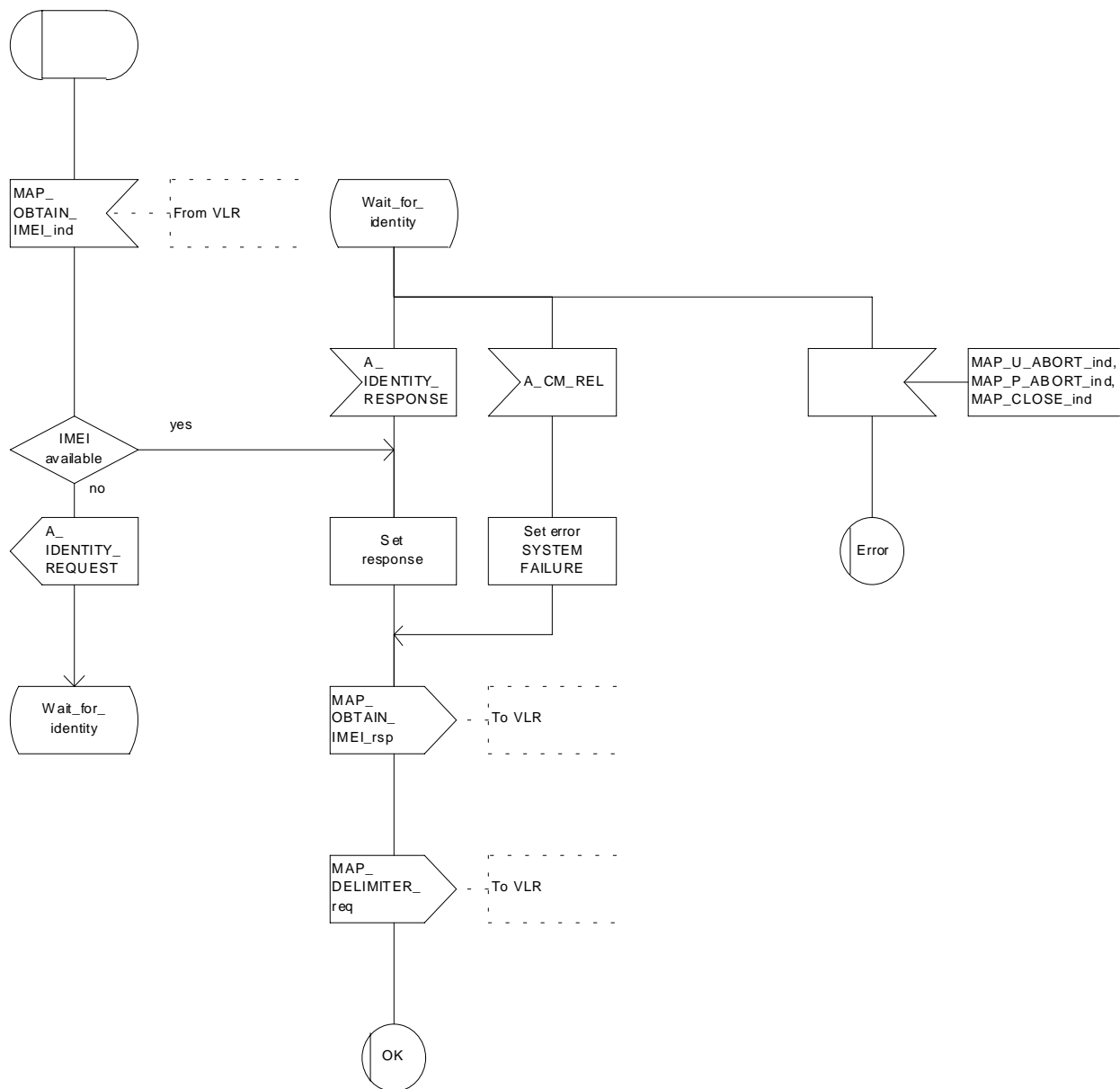


Figure 25.6/4: Process Obtain_IMEI_MSC

Macrodefinition Obtain_IMEI_VLR

25.6_5(1)

Figure 25.6/5: Obtain IMEI macro in the VLR, controlling the request towards the MSC

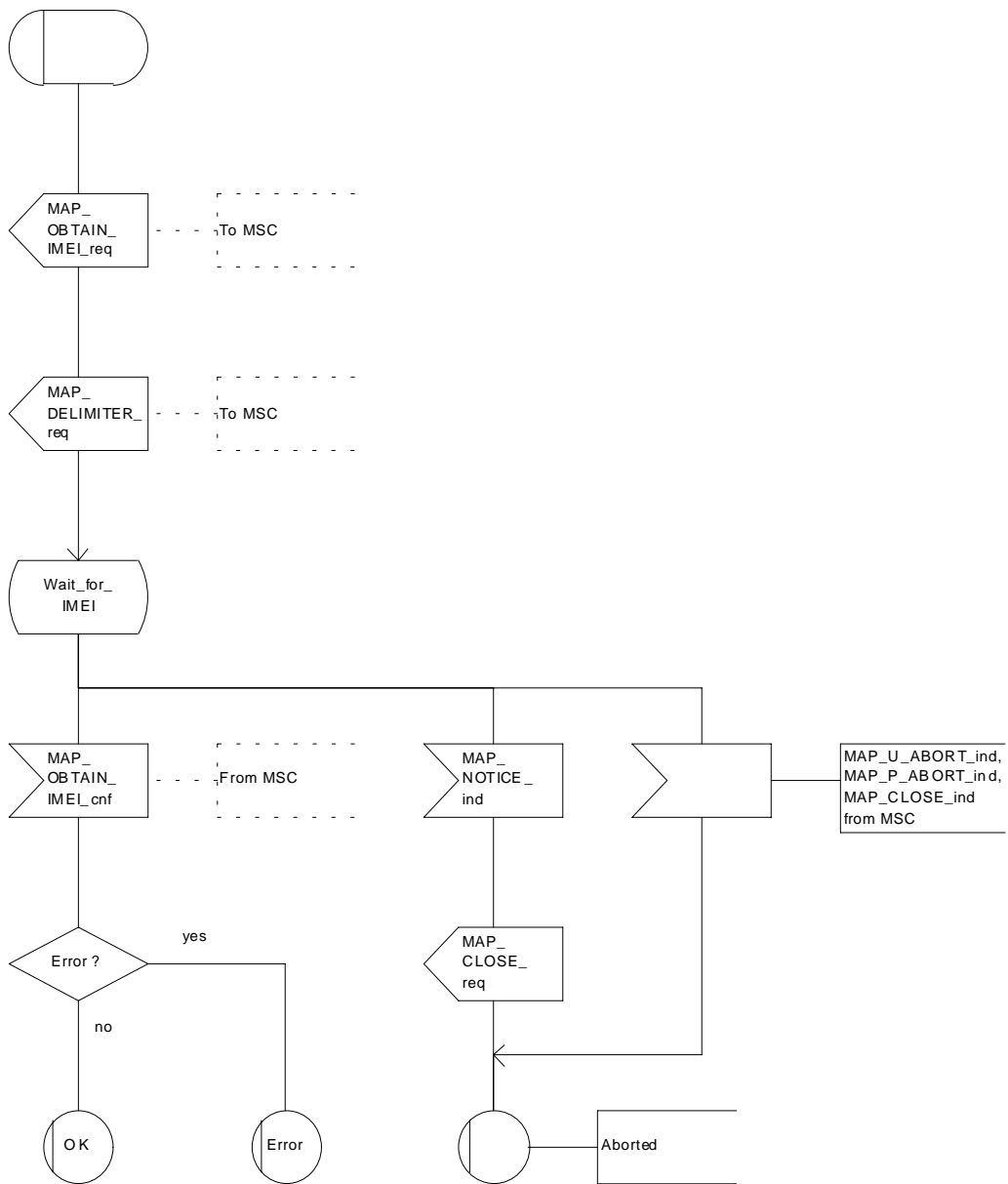


Figure 25.6/5: Process Obtain_IMEI_VLR

25.6.6 Process Check_IMEI_SGSN

This process is used by the SGSN to control the check of a mobile equipment's IMEI. The process proceeds as follows:

- if the MS does not complete successfully the procedure, 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 set. Otherwise, the SGSN waits for a response from the EIR;
- if a MAP_CHECK_IMEI service confirm including either:
 - the IMEI and the Equipment Status; or
 - an error;is received, the SGSN 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 SGSN then checks whether the response from the EIR 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 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 process is described in figure 25.6/6.

Process Check_IMEI_SGSN

25.6_6.1(2)

Figure 25.6/6: Check IMEI process in the SGSN

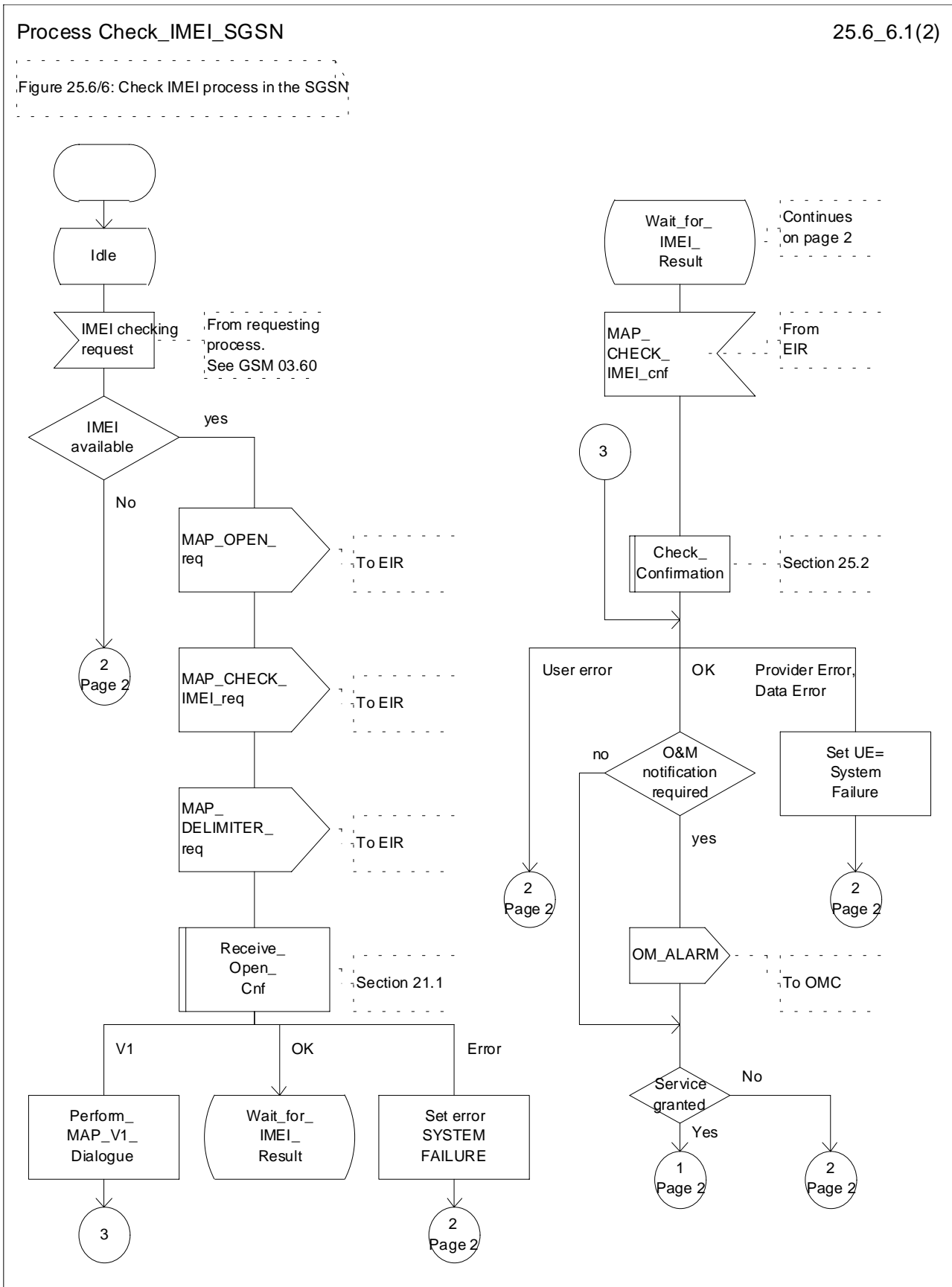


Figure 25.6/6 (sheet 1 of 2): Process Check_IMEI_SGSN

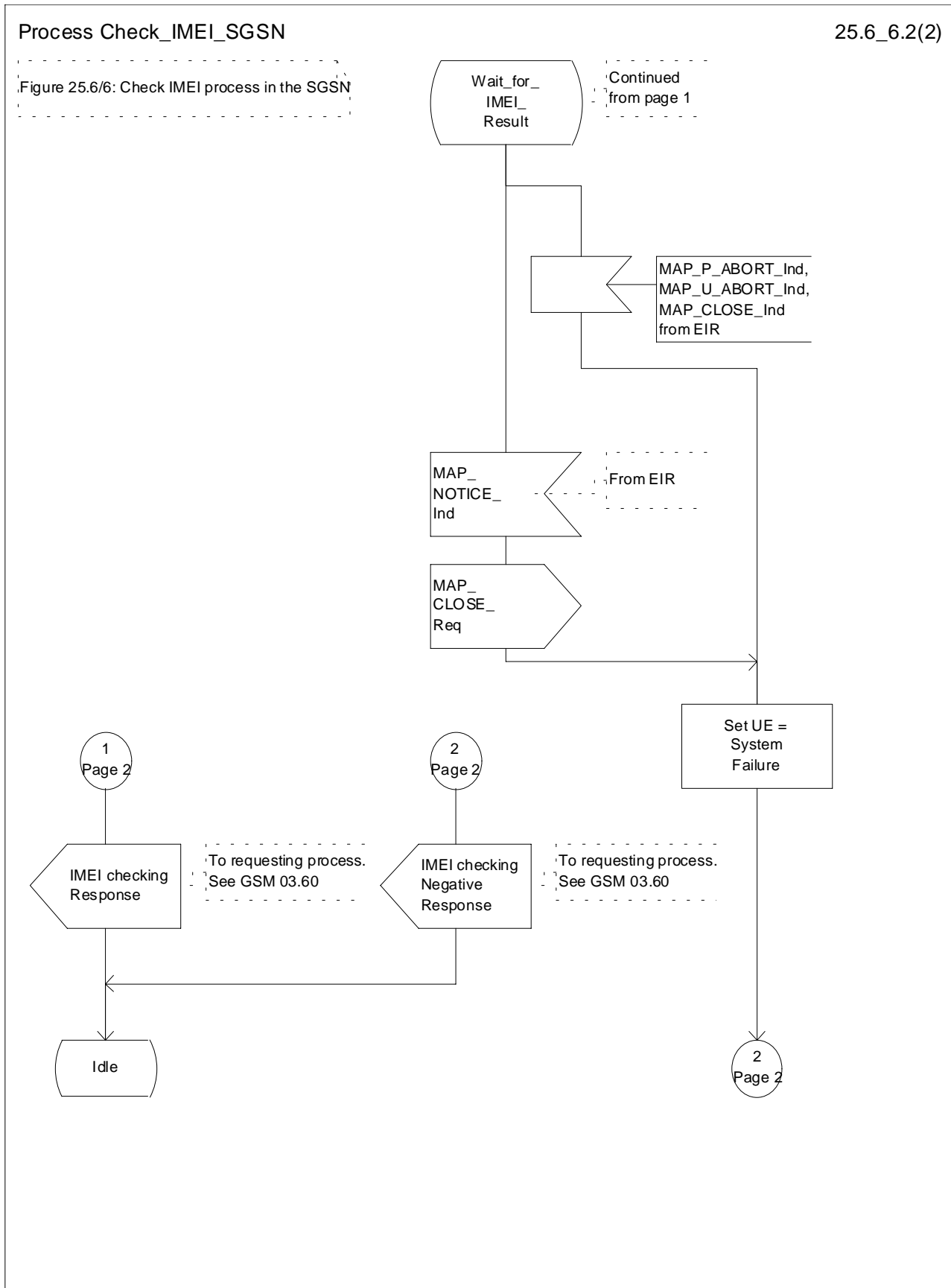


Figure 25.6/6 (sheet 2 of 2): Process Check_IMEI_SGSN

25.7 Insert Subscriber Data Macros

25.7.1 Macro Insert_SubData_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 25.7/1.

Macrodefinition Insert_Subscriber_Data_VLR

25.7_1(1)

Figure 25.7/1: Macro to receive and store subscriber data in the VLR

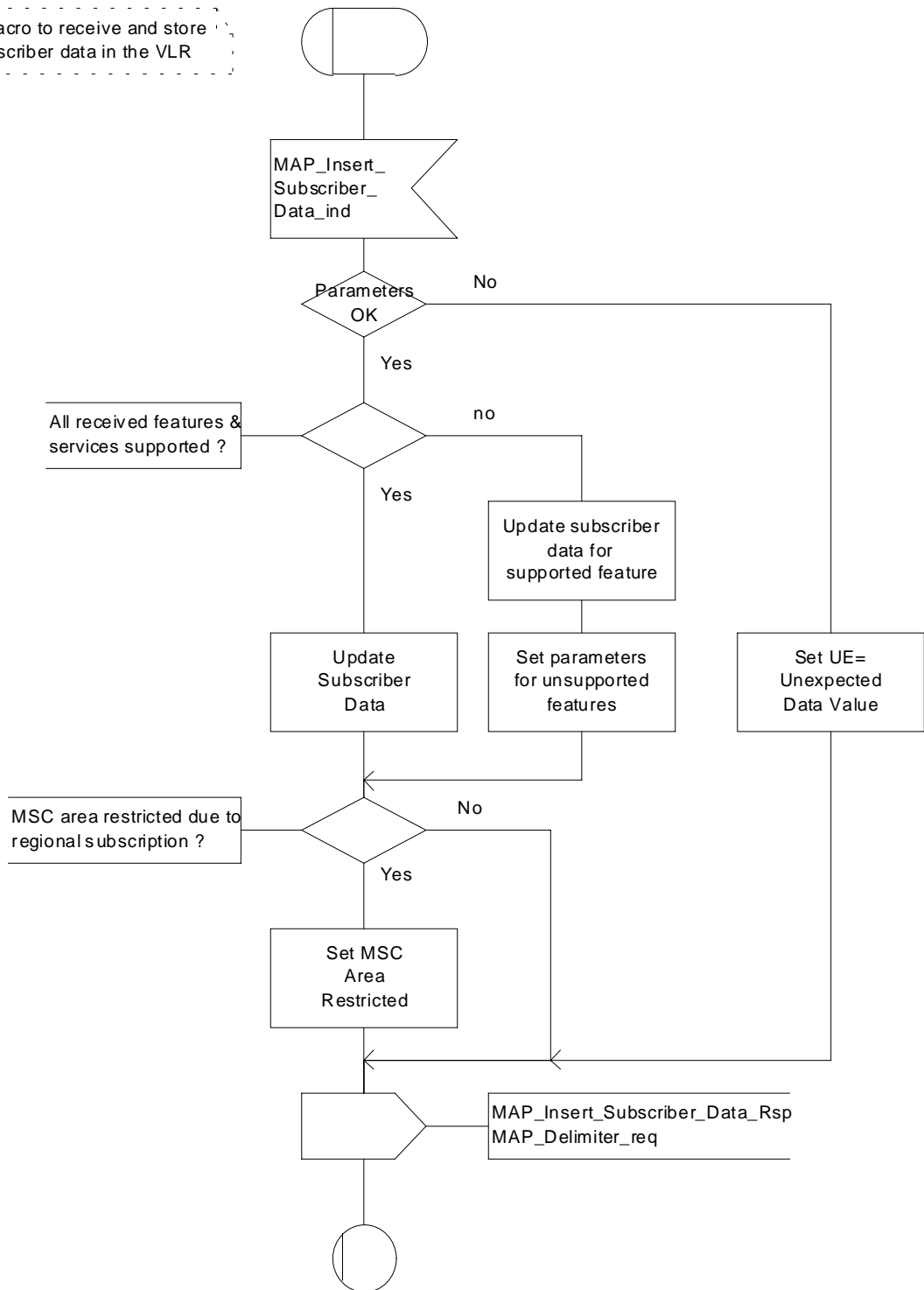


Figure 25.7/1: Macro Insert_Subscriber_Data_VLR

25.7.2 Process Insert_Subscriber_Data_Stand_Alone_HLR

This process is used by HLR to transfer subscriber data to VLR or to SGSN 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 or to SGSN.

The process, after opening the dialogue with VLR or with SGSN, sends as many requests of the InsertSubscriberData service as necessary to transfer the subscriber data. The call to the process "Send_Insert_Subscriber_Data" (see subclause 25.7.4) 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 macros "Wait_for_Insert_Subscriber_Data_Cnf" and "Wait_for_Insert_GPRS_Subscriber_Data_Cnf" (see subclauses 25.7.3 and 25.7.6) are also called in order to handle every single confirmation.

If the result of a primitive received from the VLR or from the SGSN 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 or by the SGSN.

If certain services required for a subscriber are not supported by the VLR or by the SGSN (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.
- the HLR stores and sends "Roaming Restricted In SGSN Due To Unsupported Feature" in a subsequent MAP_INSERT_SUBSCRIBER_DATA service. If "Roaming Restricted In SGSN Due To Unsupported Feature" is stored in the HLR, the "SGSN Area Restricted Flag" shall be set to "restricted". This will prevent MT SM from being forwarded to the SGSN and Network Requested PDP-Context activation.

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.

When the SGSN receives regional subscription data (Zone Code List) it may respond with "SGSN Area Restricted" in the MAP_INSERT_SUBSCRIBER_DATA response. In this case the "SGSN Area Restricted Flag" shall be set to "restricted" in the HLR. This will prevent MT SM from being forwarded to the SGSN and Network Requested PDP-Context activation.

If subscriber data for CAMEL Phase 2 or 3 services are sent to a VLR which does not support CAMEL Phase 2 or 3, the service behaviour may be unpredictable or incorrect. The HLR therefore needs to ensure that at the conclusion of a stand alone Insert Subscriber data procedure that the data in the VLR do not require a capability that the VLR does not have. Possible mechanisms to ensure this are described in 3G TS 23.078.

The HLR should send a Forwarded-to number which is not in E.164 international format to the VLR only when the HLR has ascertained that the VLR supports CAMEL Phase 2 or 3. Thus, the ISD message containing the Forwarded-to number which is not in E.164 international format shall be sent to the VLR only if the HLR previously received confirmation from the VLR at Location Update that CAMEL Phase 2 or 3 is supported.

A Forwarded-to number in non-international E.164 format shall only be sent from an HLR to a VLR if the VLR supports CAMEL Phase 2, or a subsequent version of CAMEL.

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".

If the HLR does not store "Roaming Restricted In SGSN Due To Unsupported Feature" as a consequence of the stand alone Insert Subscriber Data procedure and the HLR does not receive "SGSN Area Restricted" in the MAP_INSERT_SUBSCRIBER_DATA response and "Roaming Restricted In SGSN Due To Unsupported Feature" has not been stored in the HLR in the course of a previous subscriber data retrieval procedure, the "SGSN Area Restricted Flag" in the HLR shall be set to "not restricted".

The SDL diagram of process between HLR and VLR is shown in figure 25.7/2;

The SDL diagram of process between HLR and SGSN is shown in figure 25.7/5.

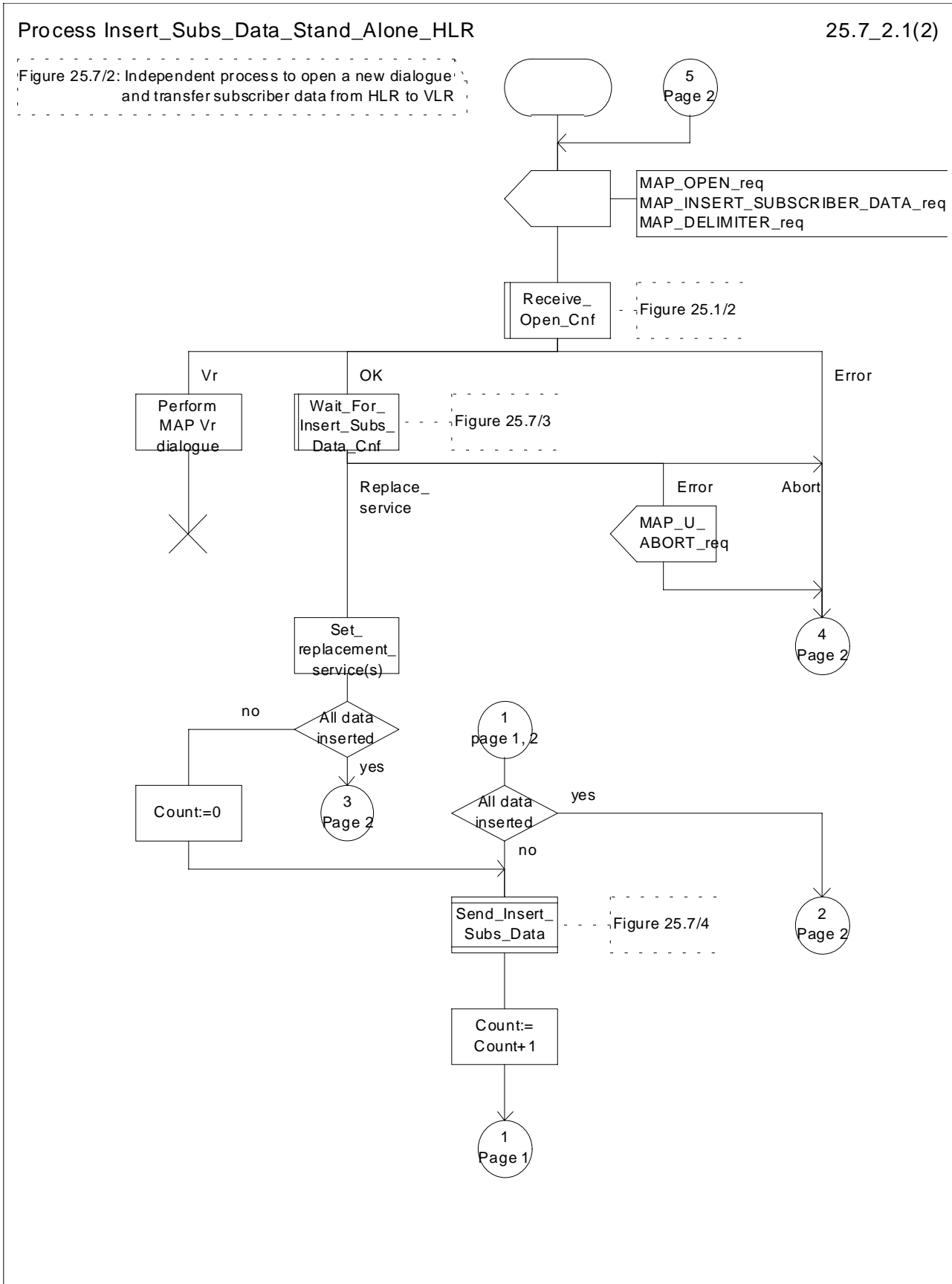


Figure 25.7/2 (sheet 1 of 2): Process Insert_Subs_Data_Stand_Alone_HLR

Process Insert_Subs_Data_Stand_Alone_HLR

25.7_2.2(2)

Figure 25.7/2: Independent process to open a new dialogue and transfer subscriber data from HLR to VLR

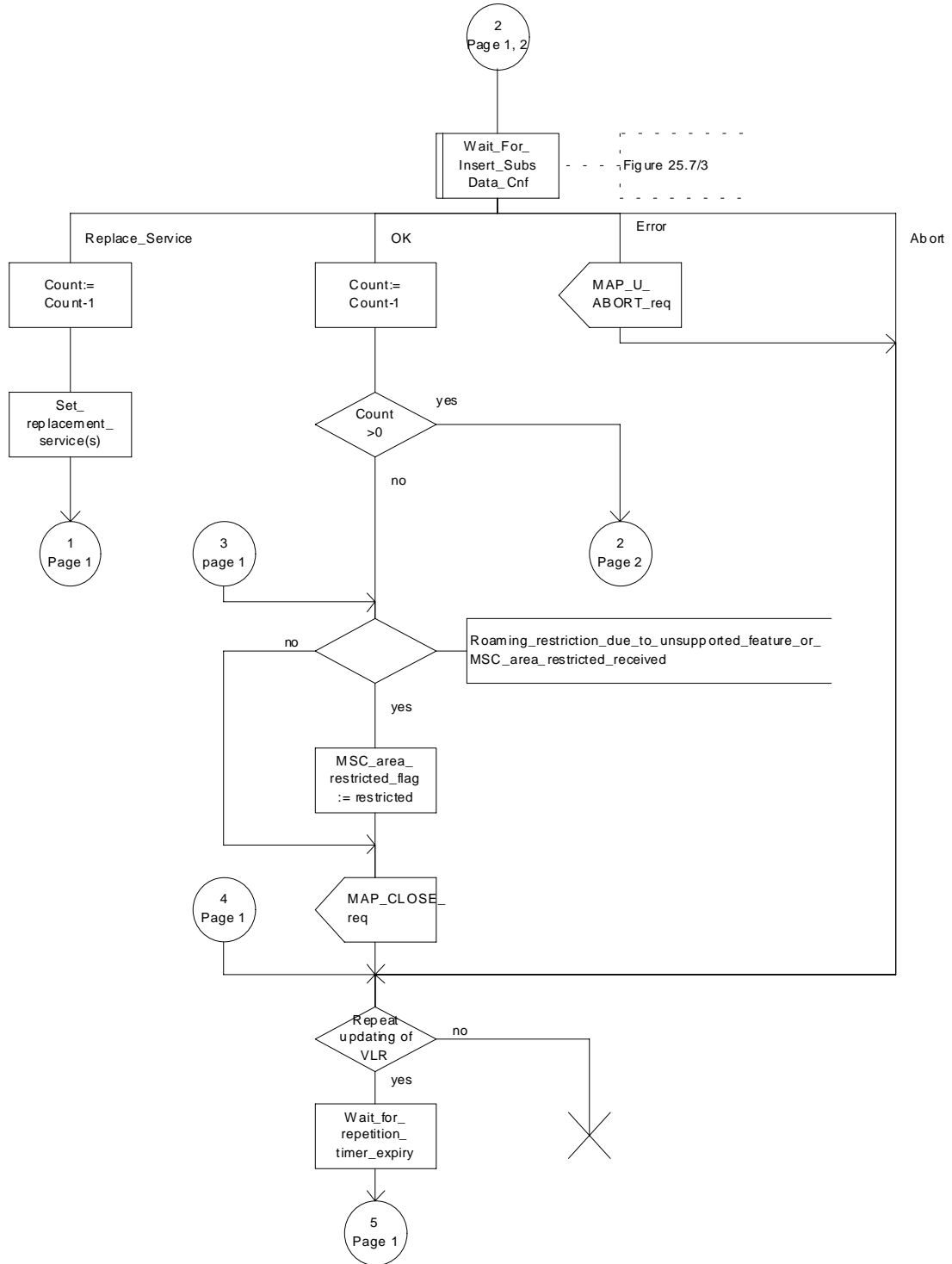


Figure 25.7/2 (sheet 2 of 2): Process Insert_Subs_Data_Stand_Alone_HLR

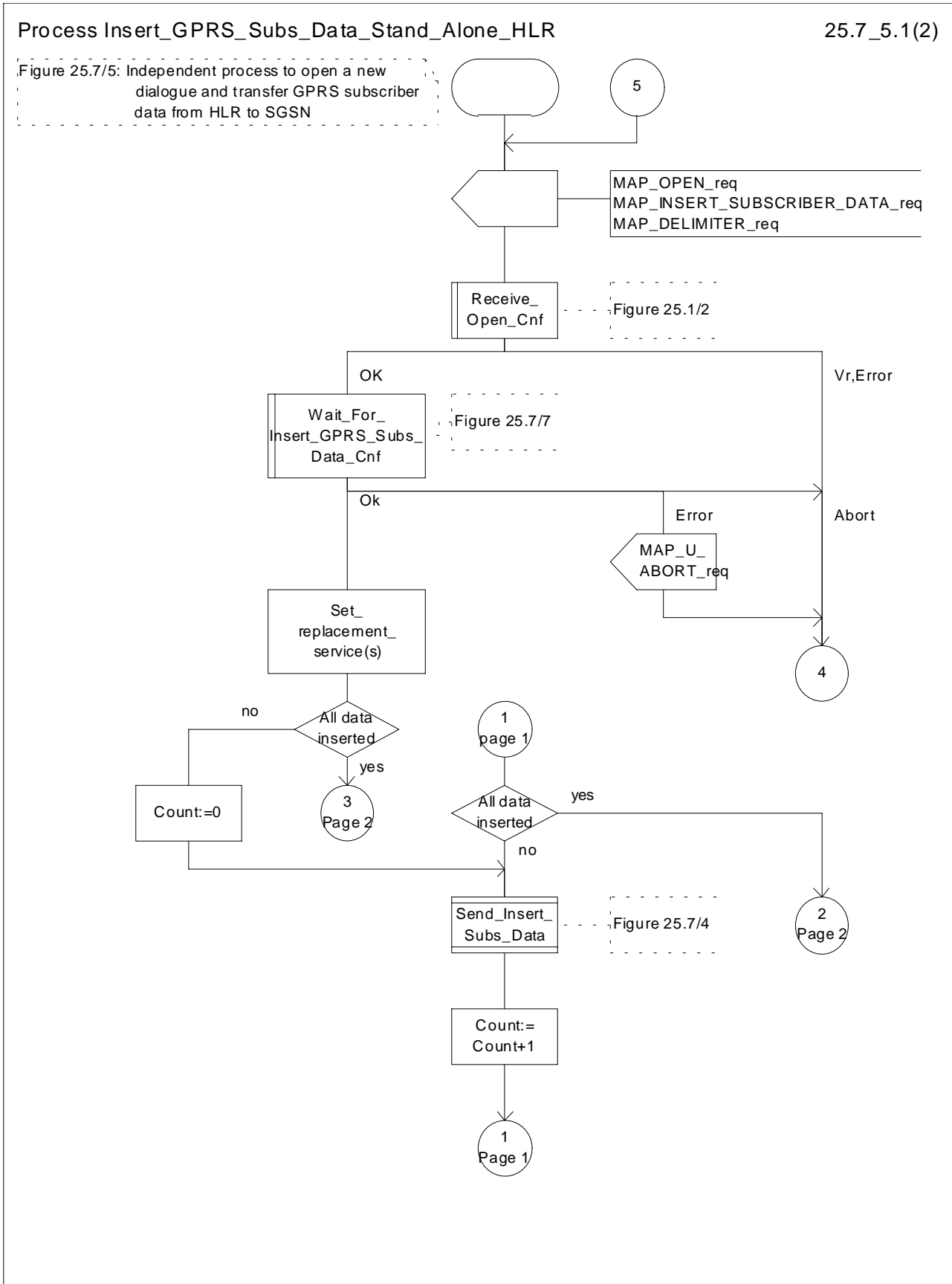


Figure 25.7/5 (sheet 1 of 2): Process Insert_GPRS_Subscriber_Data_Stand_Alone_HLR

Process Insert_GPRS_Subscriber_Data_Stand_Alone_HLR

25.7_5.2(2)

Figure 25.7/5: Independent process to open a new dialogue and transfer GPRS subscriber data from HLR to SGSN

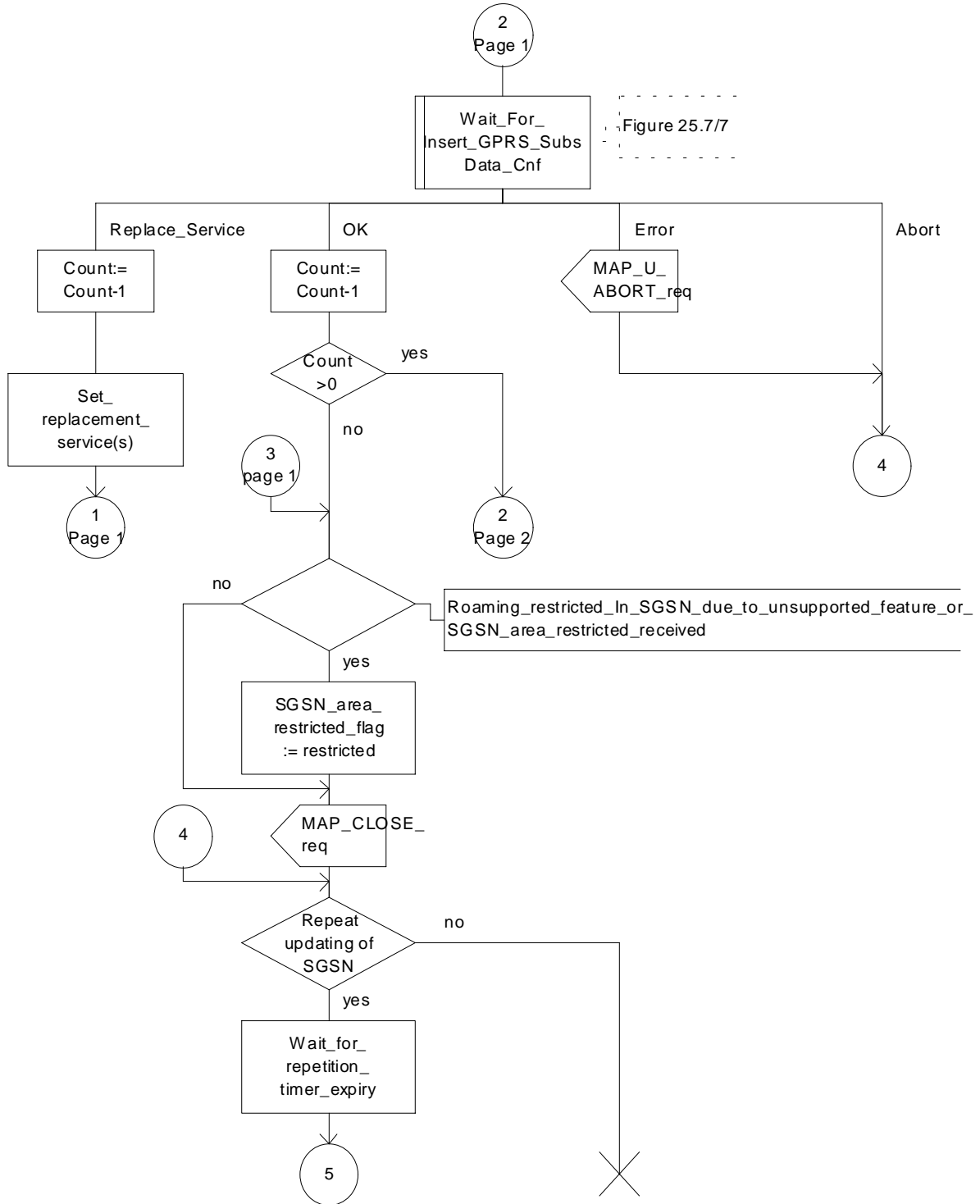


Figure 25.7/5 (sheet 2 of 2): Process Insert_GPRS_Subscriber_Data_Stand_Alone_HLR

25.7.3 Macro Wait_for_Insert_Subscriber_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 that is coming from VLR (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 25.7/3.

Macrodefinition Wait_For_Insert_Subscriber_Data_Cnf

25.7_3(1)

Figure 25.7/3: Macro to receive confirmation or error indication for MAP_INSERT_SUBSCRIBER_DATA

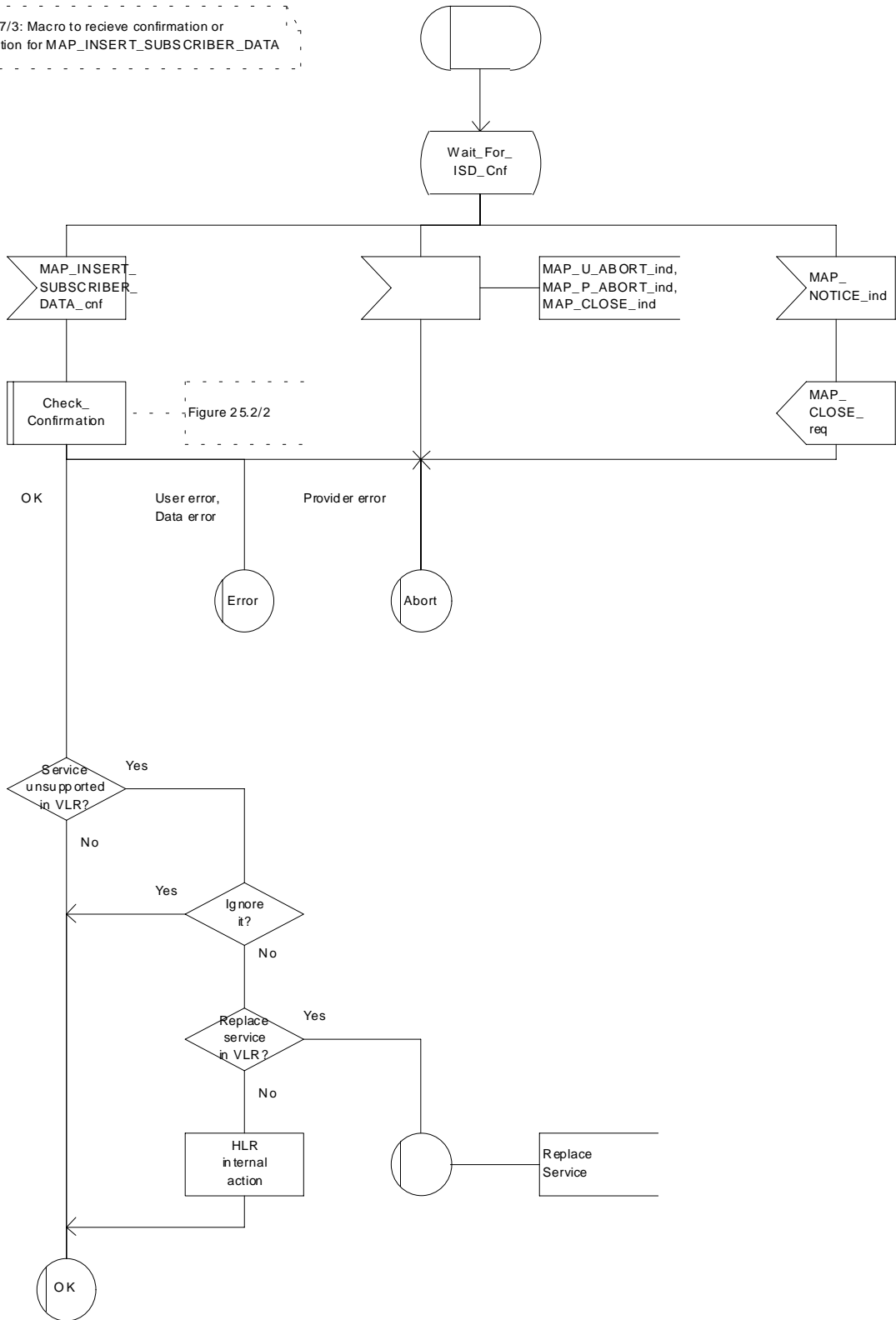


Figure 25.7/3: Macro Wait_for_Insert_Subscriber_Data_Cnf

25.7.4 Process Send_Insert_Subscriber_Data

This process is used by any process or macro where the Insert_Subscriber_Data request is sent to VLR or to SGSN.

The SDL diagram is shown in figure 25.7/4.

Process Send_Insert_SubData

25.7_4(1)

Figure 25.7/4: Independent process to send a component of subscriber data from HLR to VLR

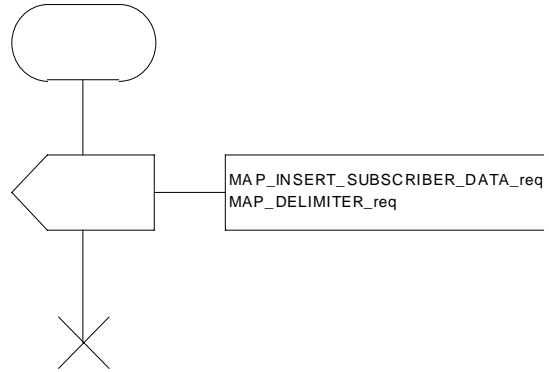


Figure 25.7/4: Process Send_Insert_SubData

25.7.5 Macro Insert_SubscriberData_SGSN

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 GPRS Location).

If the SGSN does not support any basic 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 SGSN area is restricted due to regional subscription this is reported to the HLR.

The SDL diagram is shown in figure 25.7/6.

Macrodefinition Insert_Subs_Data_SGSN

25.7_6(1)

Figure 25.7/6: Macro to receive and store subscriber data in the SGSN

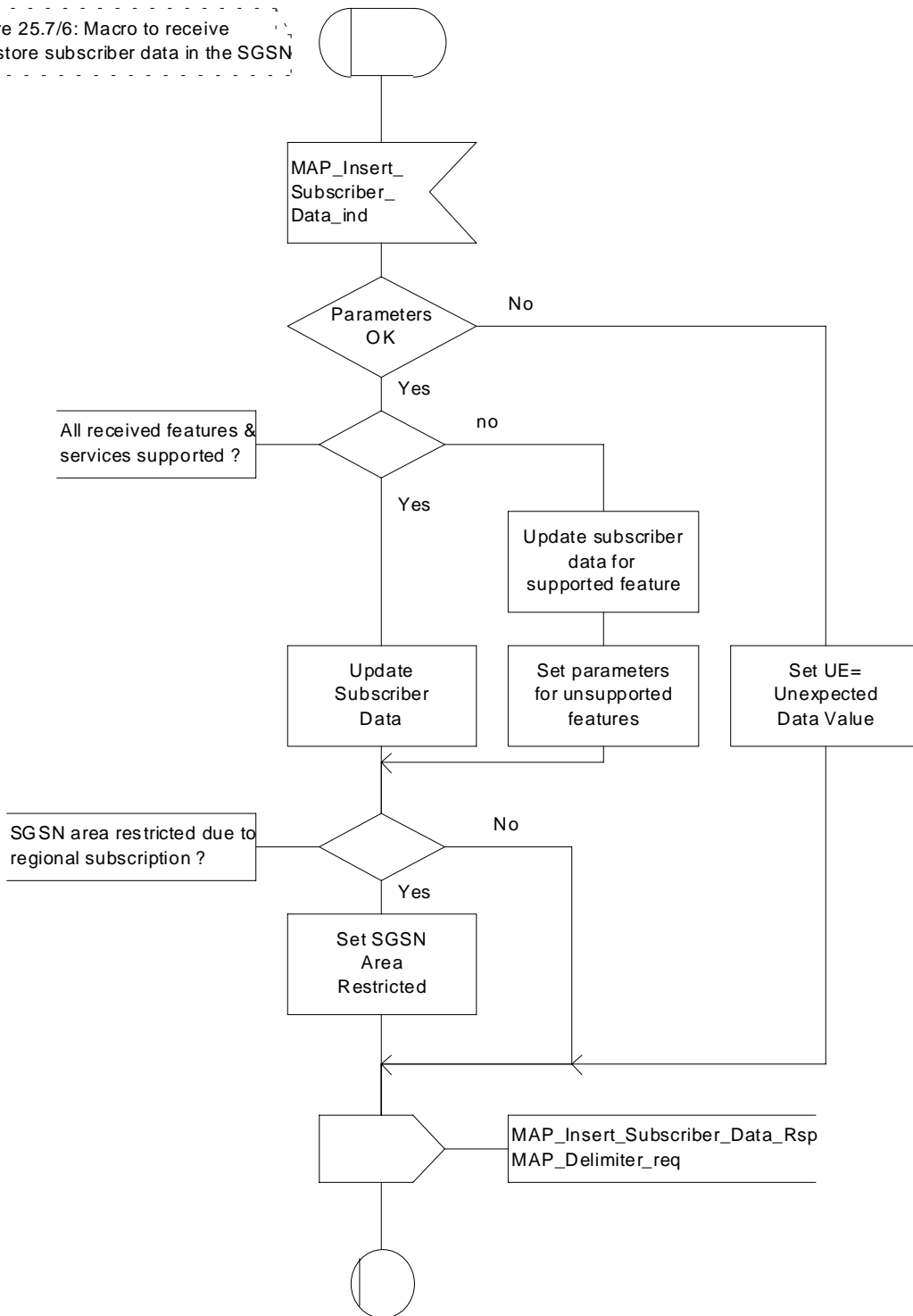


Figure 25.7/6: Macro Insert_Subs_Data_SGSN

25.7.6 Macro Wait_for_Insert_GPRS_Subscriber_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 that is coming from SGSN (e.g. Update GPRS Location).

If the SGSN reports the non-support of some basic 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 25.7/7.

Macrodefinition Wait_For_Insert_GPRS_Subscriber_Data_Cnf

25.7_7(1)

Figure 25.7/7: Macro to receive confirmation or error indication for MAP_INSERT_SUBSCRIBER_DATA from SGSN

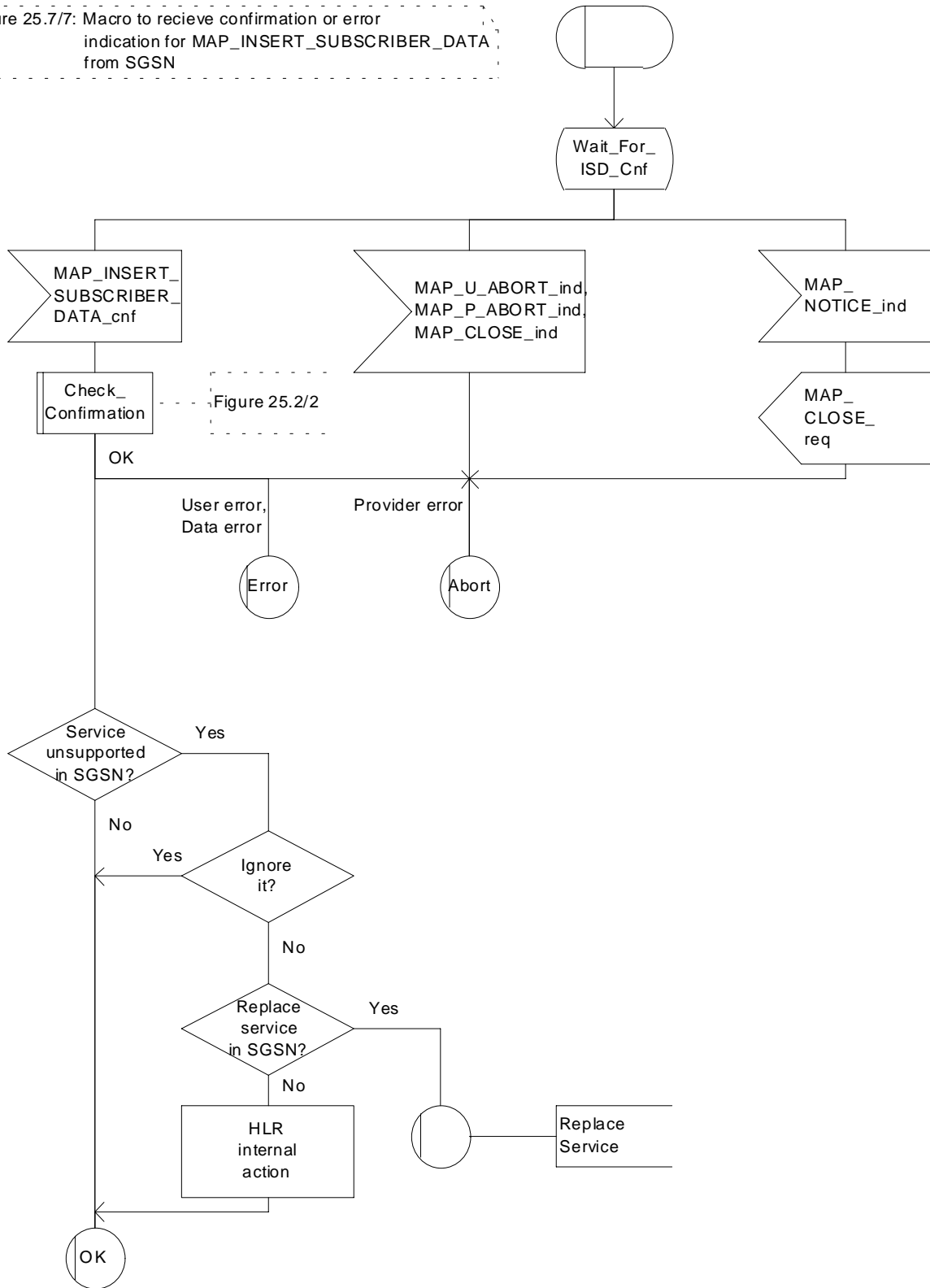


Figure 25.7/7: Macro Wait_for_Insert_GPRS_Subscriber_Data_Cnf

25.8 Request IMSI Macros

25.8.1 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 25.8/1.

Macrodefinition Obtain_IMSI_MSC

25.8_1(1)

Figure 25.8/1: Macro to relay an IMSI request from the VLR to the MS and return the response to the VLR

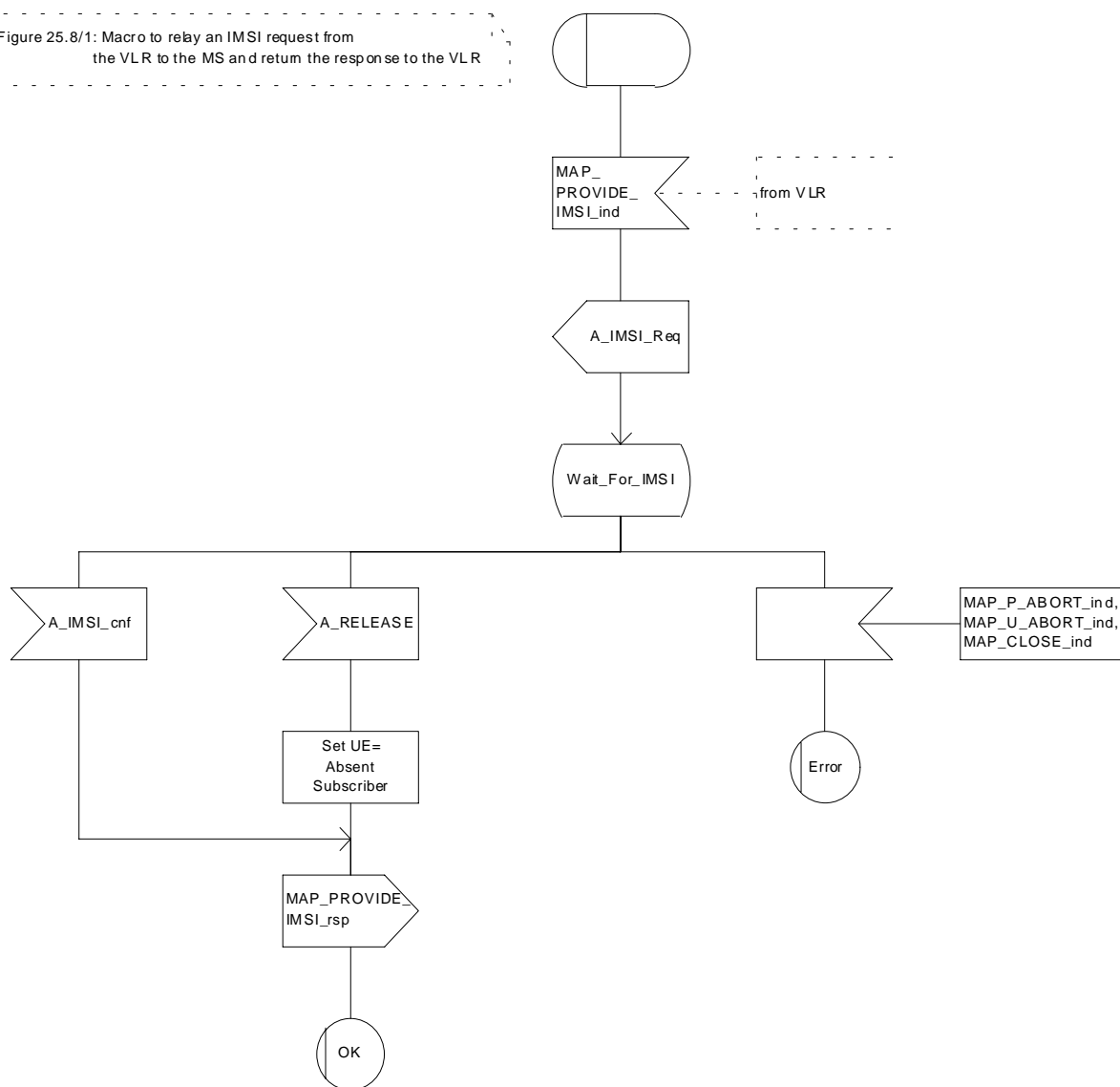


Figure 25.8/1: Macro Obtain_IMSI_MSC

25.8.2 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 25.8/2.

Macrodefinition Obtain_IMSI_VLR

25.8_2(1)

Figure 25.8/2: Macro to obtain the IMSI from the MS via the MSC

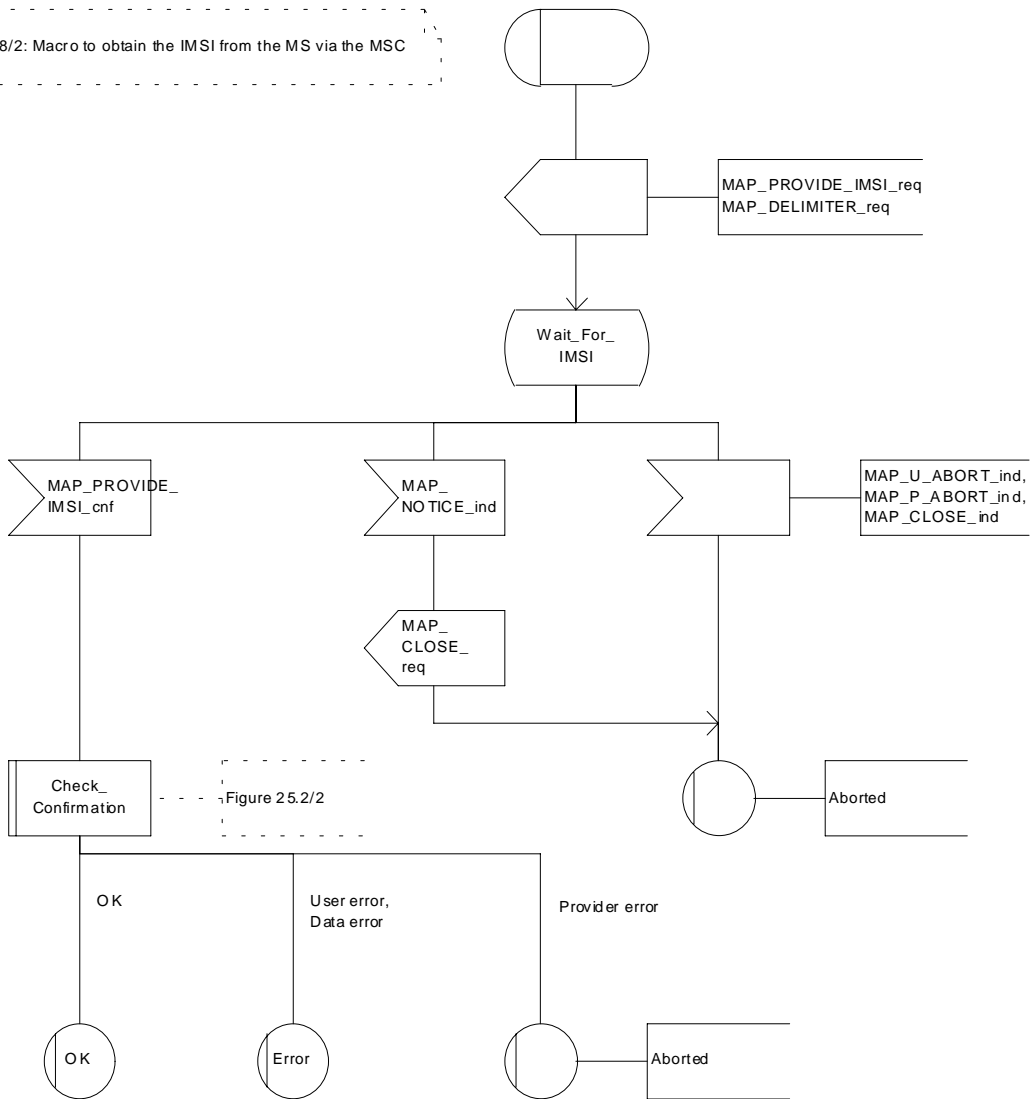


Figure 25.8/2: Macro Obtain_IMSI_VLR

25.9 Tracing macros

25.9.1 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 25.9/1.

Macrodefinition Trace_Subscriber_Activity_MSC

25.9_1(1)

Figure 25.9/1: The Subscriber tracing macro in the MSC

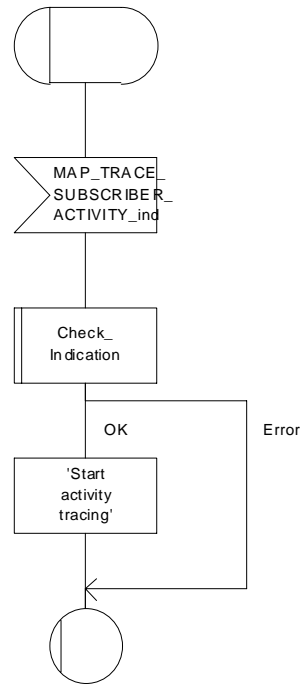


Figure 25.9/1: Macro Trace_Subscriber_Activity_MSC

25.9.2 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 25.9/2.

Macrodefinition Trace_Subscriber_Activity_VLR

25.9_2(1)

Figure 25.9/2: The subscriber tracing macro in the VLR

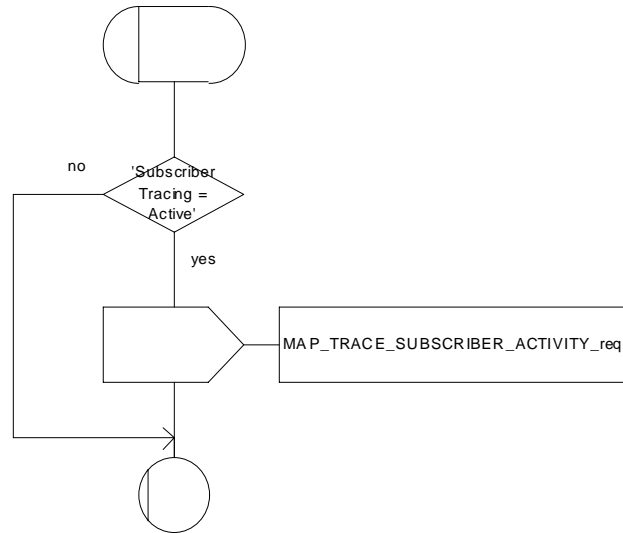


Figure 25.9/2: Macro Trace_Subscriber_Activity_VLR

25.9.3 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 25.9/3.

Macrodefinition Activate_Tracing_VLR

25.9_3(1)

Figure 25.9/3: The activate trace mode macro in the VLR

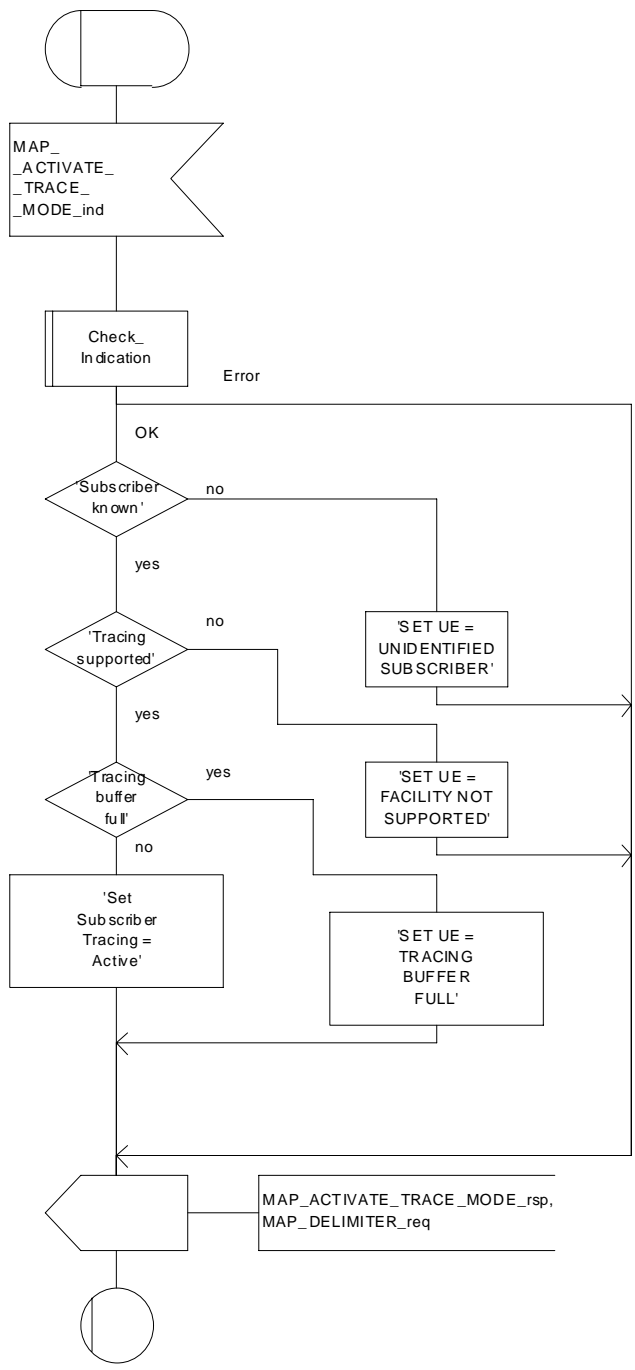


Figure 25.9/3: Macro Activate_Tracing_VLR

25.9.4 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 or in the SGSN, the MAP_ACTIVATE_TRACE_MODE request is sent to the VLR or to the SGSN.

The MAP_ACTIVATE_TRACE_MODE confirmation from the VLR or from the SGSN is processed as follows:

- if the primitive contains a successful acknowledge, the tracing in VLR or in the SGSN is set active;
- if the primitive contains errors, the tracing in VLR or in SGSN is set deactive.

The Control_Tracing_HLR macro between HLR and VLR is shown in the figure 25.9/4

The Control_Tracing_HLR_with_SGSN macro between HLR and SGSN is shown in the figure 25.9/5

Macrodefinition Control_Tracing_HLR

25.9_4(1)

Figure 25.9/4: The subscriber tracing activation macro in the HLR

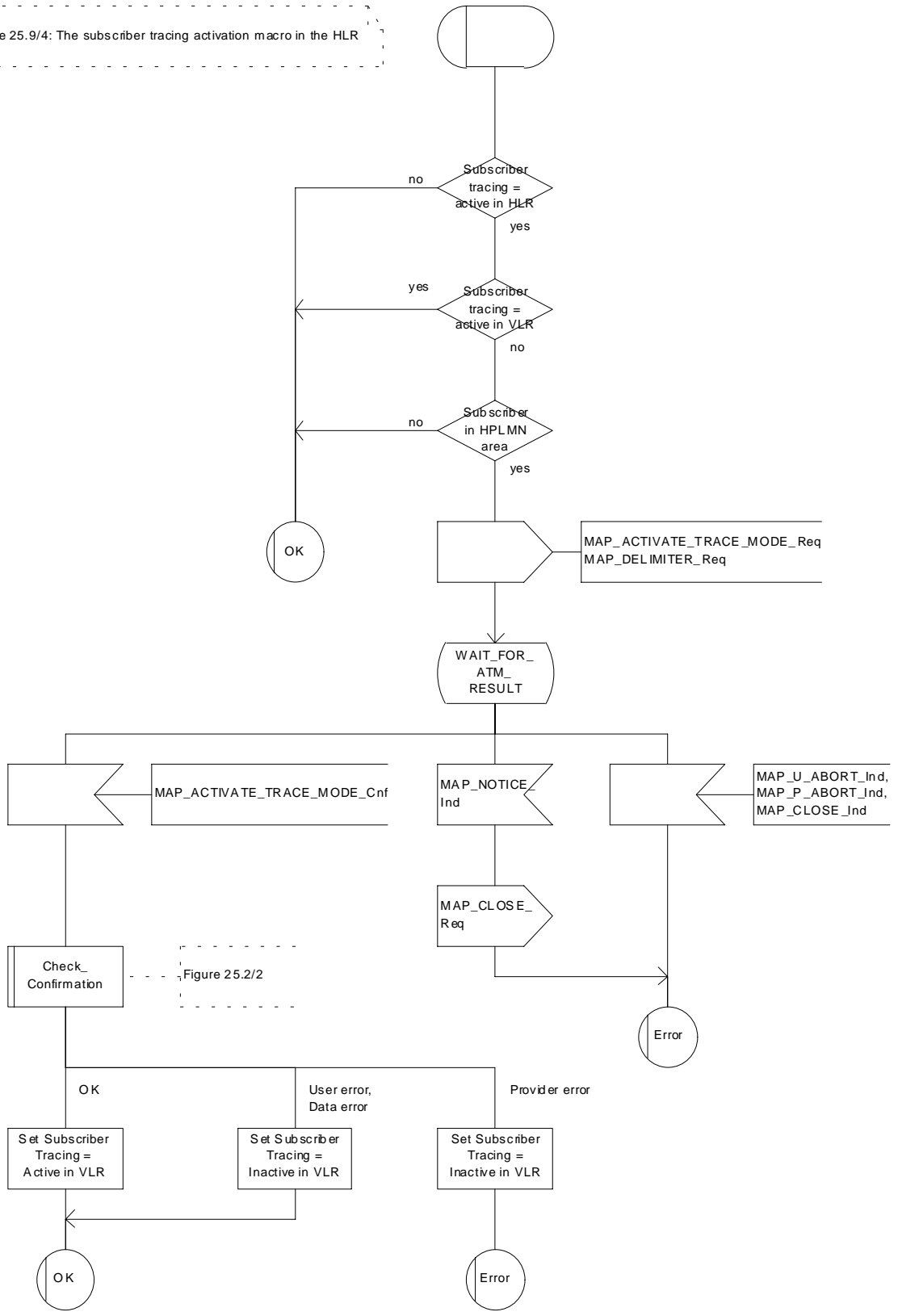


Figure 25.9/4: Macro Control_Tracing_HLR

Macrodefinition Control_Tracing_HLR_with_SGSN

25.9_5(1)

Figure 25.9/5: The subscriber tracing activation macro in the HLR

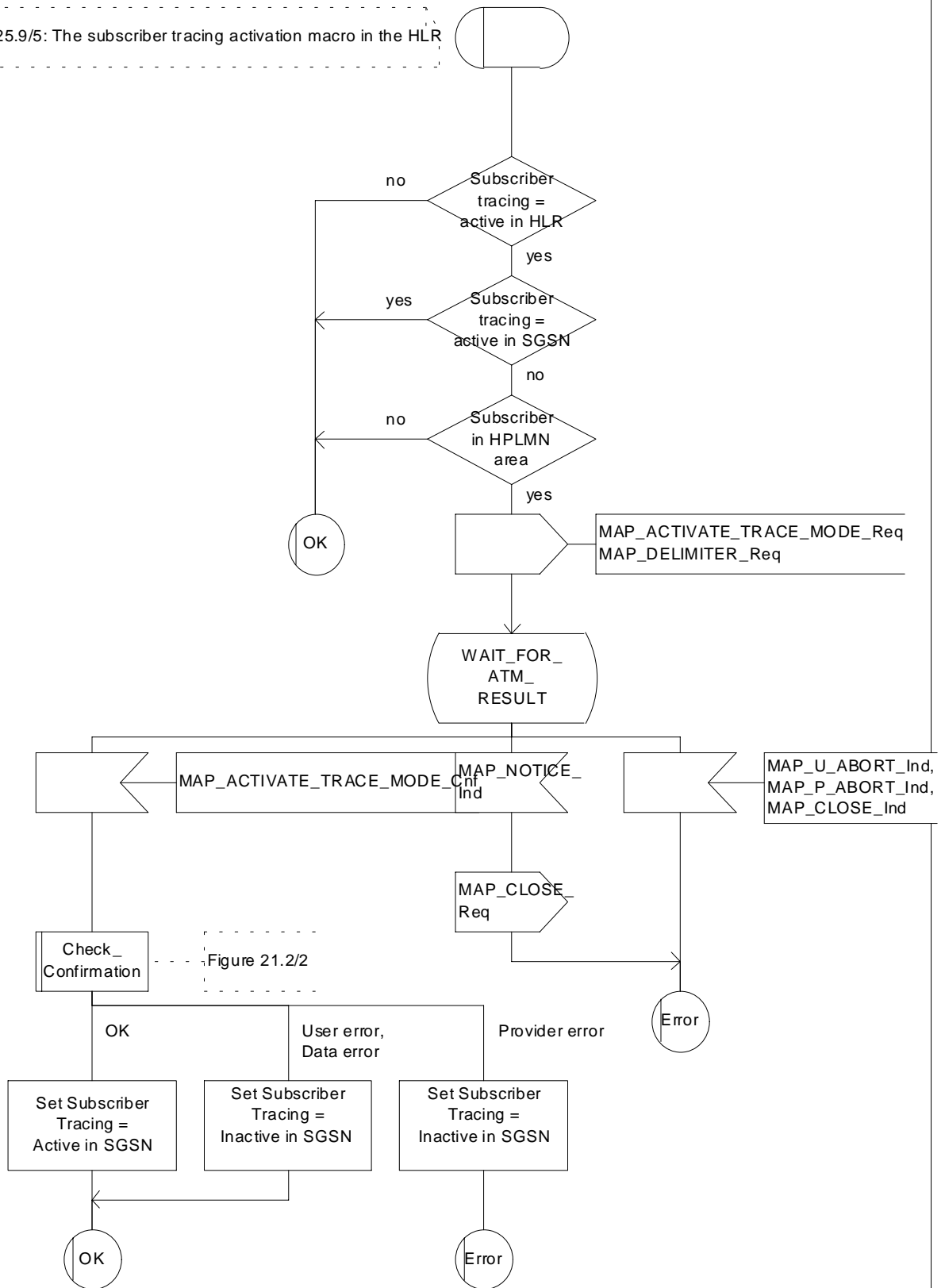


Figure 25.9/5: Macro Control_Tracing_HLR_with_SGSN

25.9.5 Macro Trace_Subscriber_Activity_SGSN

The macro Trace_Subscriber_Activity_SGSN is invoked, if the subscriber activity is detected by the SGSN and the tracing is active.

The Trace_Subscriber_Activity_SGSN macro is shown in the figure 25.9/6.

Macrodefinition Trace_Subscriber_Activity_SGSN

25.9_6(1)

Figure 25.9/6: The subscriber tracing macro in the SGSN

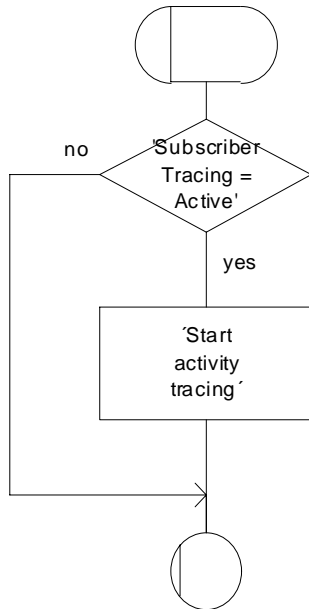


Figure 25.9/6: Macro Trace_Subscriber_Activity_SGSN

25.9.6 Macro Activate_Tracing_SGSN

The Activate_Tracing_SGSN macro is invoked, when the MAP_ACTIVATE_TRACE_MODE indication is received from the HLR. The primitive is processed in the SGSN 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_SGSN macro is described in the figure 25.9/7.

Macrodefinition Activate_Tracing_SGSN

25.9_7(1)

Figure 25.9/7: The activate trace mode macro in the SGSN:

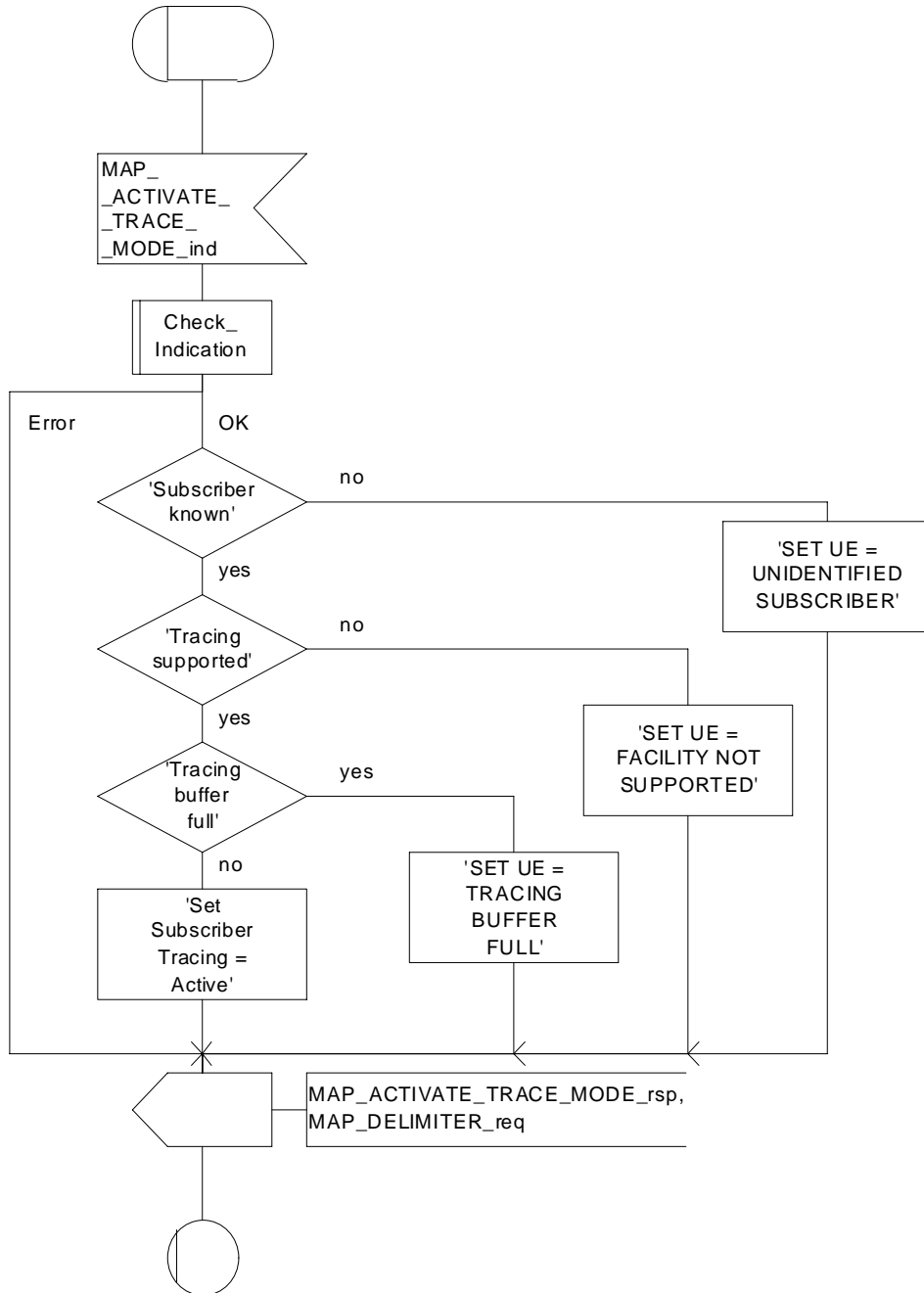


Figure 25.9/7: Macro Activate_Tracing_SGSN

25.10 Short Message Alert procedures

25.10.1 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 23.4.

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 25.10/1.

Process Subscriber_Present_VLR

25.10_1(1)

Figure 25.10/1: The short message alert process in the VLR for mobile present situation

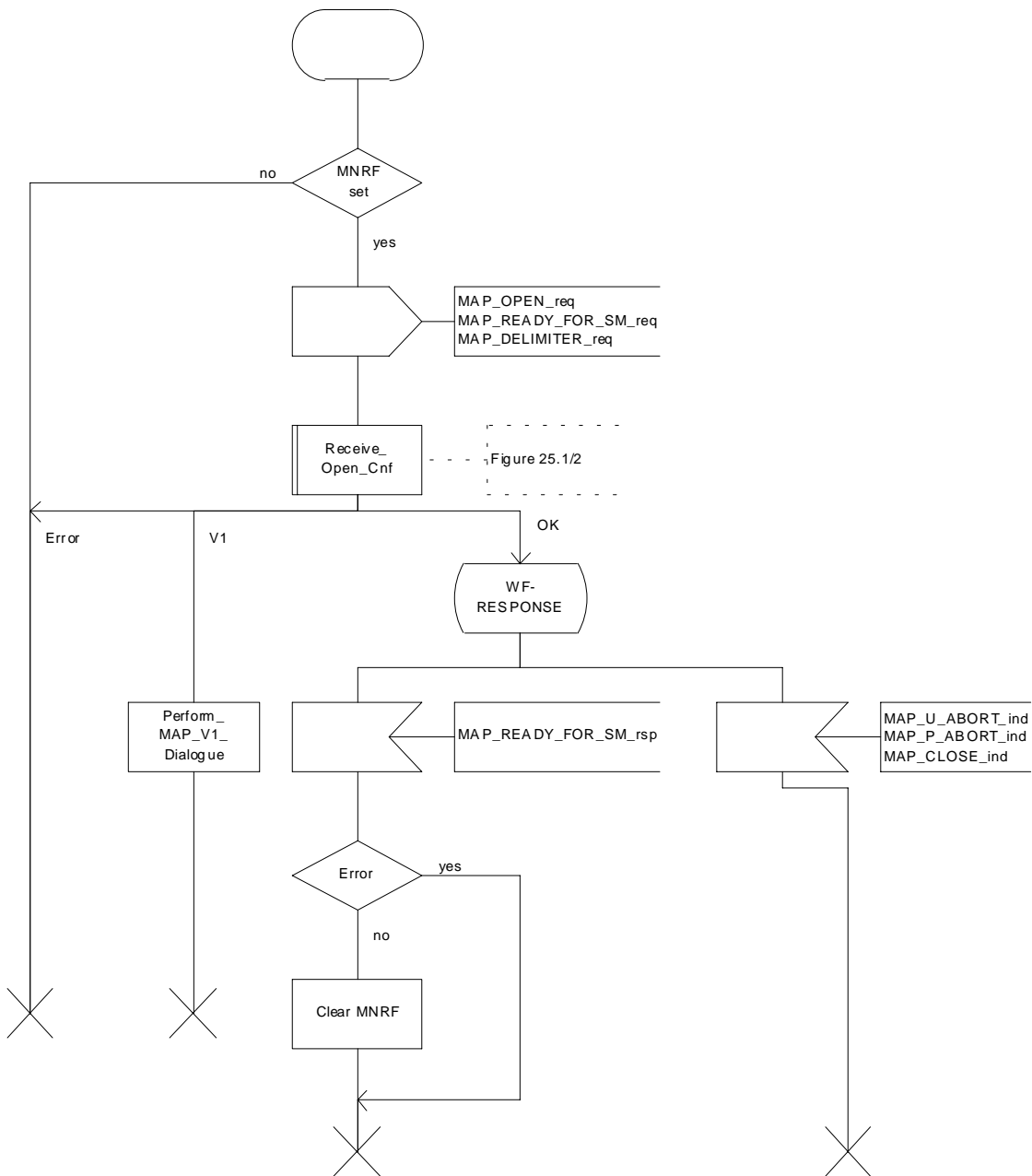


Figure 25.10/1: Process Subscriber_Present_VLR

25.10.2 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 25.10/2 and the Alert_Service_Centre_HLR process is shown in the figure 25.10/3.

Macrodefinition Alert_Service_Centre_HLR

25.10_2(1)

Figure 25.10/2: The short message alert macro in the HLR

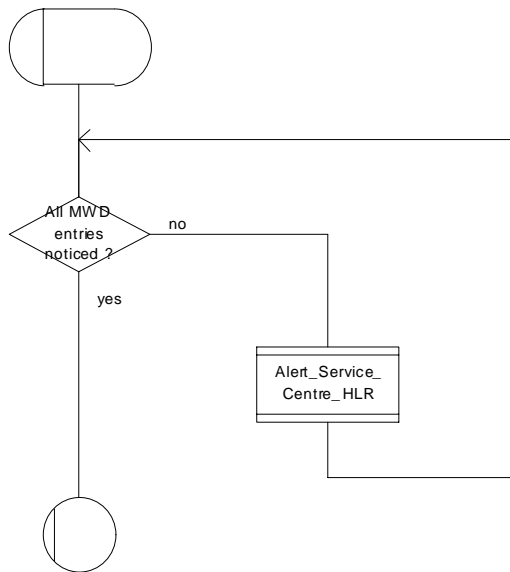


Figure 25.10/2: Macro Alert_Service_Centre_HLR

Process Alert_Service_Centre_HLR

22.10_3(1)

Figure 25.10/3: The short message alert process in the HLR

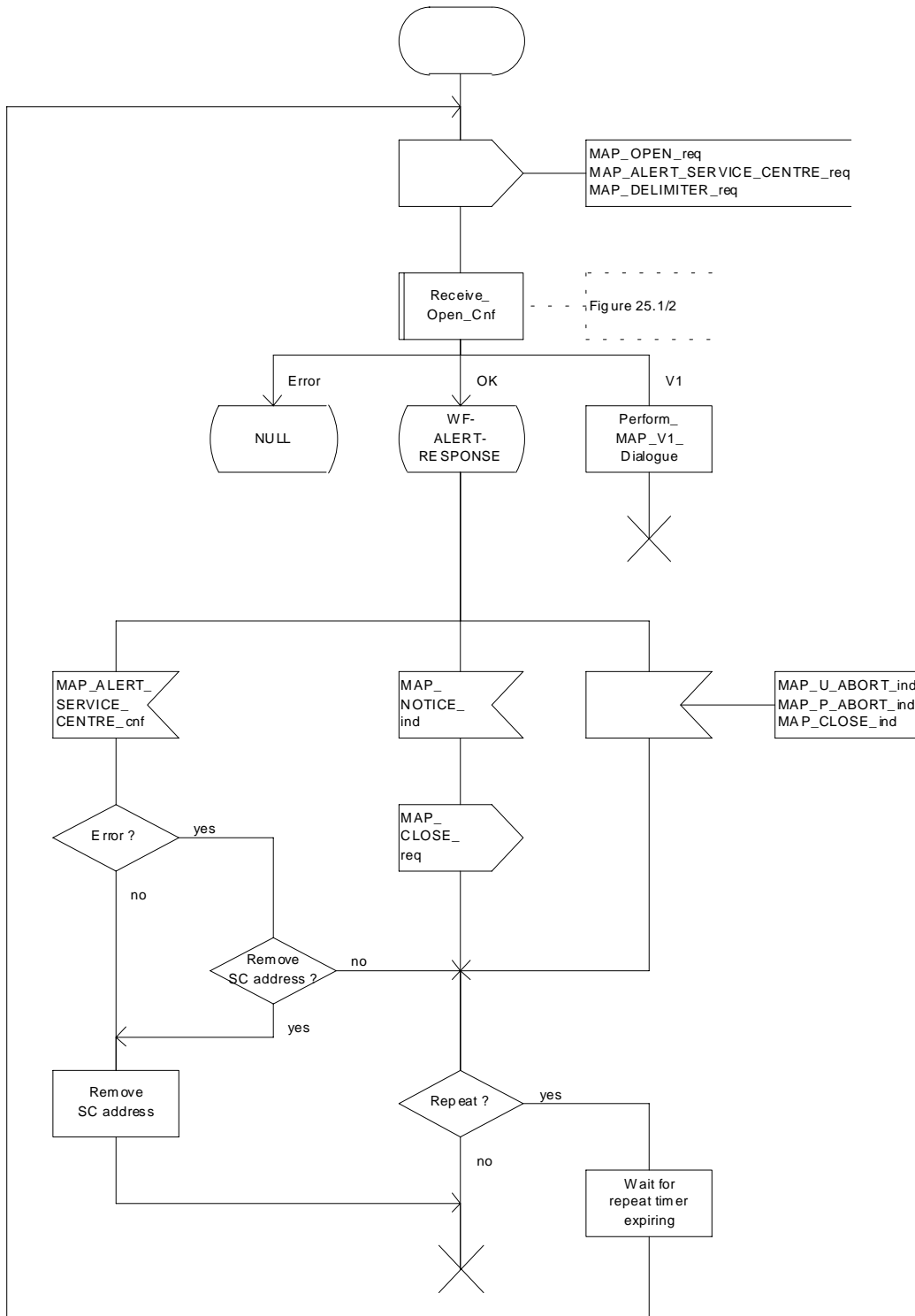


Figure 25.10/3: Process Alert_Service_Centre_HLR

25.10.3 The Mobile Subscriber is present

When receiving Page response, Attach request or Routing area update request messages (TS GSM 04.08), while the MS not reachable for GPRS (MNRG) flag is set, the SGSN will send the MAP_READY_FOR_SM request towards the HLR. The Alert Reason is set to indicate that the mobile subscriber is present for GPRS.

When receiving the answer, the SGSN will act as follows:

- MNRG is cleared if the procedure is successful
- MNRG is not cleared if the procedure is not successful

The Subscriber_Present_SGSN process is shown in the figure 25.10/4.

Process Subscriber_Present_SGSN

25.10_4(1)

Figure 25.10/4: The short message alert process in the SGSN for mobile present situation

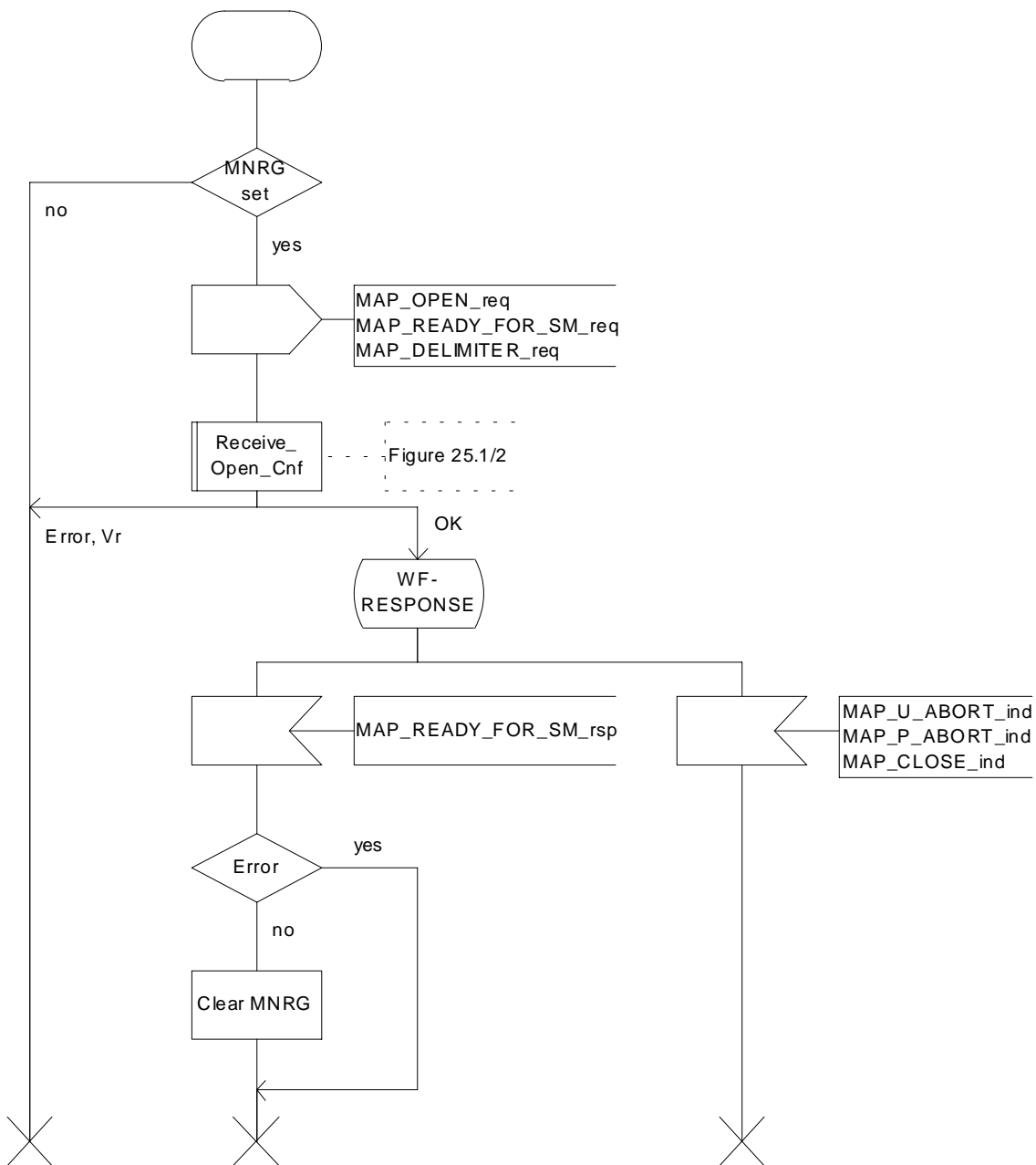


Figure 25.10/4: Process Subscriber_Present_SGSN

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.

TAG R4.21 Cross Reference Listing for MAP-Protocol 00-01-06 07:32:40 PAGE 1

```

&extensionId.....identifier of Fieldspec
  DEFINED in MAP-ExtensionDataTypes : 24
  USED in MAP-ExtensionDataTypes : 41

&ExtensionType.....identifier of Fieldspec
  DEFINED in MAP-ExtensionDataTypes : 23
  USED in MAP-ExtensionDataTypes : 43

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 : 369

AbsentSubscriber.....type reference ERROR
  DEFINED in MAP-Errors : 272
  USED in MAP-Protocol : 134 369
  USED in MAP-MobileServiceOpera : 92 413
  USED in MAP-CallHandlingOperat : 40 96 115 188
  USED in MAP-SupplementaryServi : 50 197 211
  USED in MAP-LocationServiceOpe : 28 63 79
  USED in MAP-Errors : 47

absentSubscriber.....identifier of Named Number, 1
  DEFINED in MAP-SM-DataTypes : 165

absentSubscriberDiagnosticSM.....identifier of [0] AbsentSubscriberDiagnosticSM
  DEFINED in MAP-SM-DataTypes : 147

absentSubscriberDiagnosticSM.....identifier of AbsentSubscriberDiagnosticSM
  DEFINED in MAP-ER-DataTypes : 155

AbsentSubscriberDiagnosticSM.....type reference INTEGER
  DEFINED in MAP-ER-DataTypes : 165
  USED in MAP-MS-DataTypes : 173 1247
  USED in MAP-SM-DataTypes : 40 147 159
  USED in MAP-ER-DataTypes : 43 155 160

absentSubscriberParam.....identifier of AbsentSubscriberParam
  DEFINED in MAP-Errors : 274

AbsentSubscriberParam.....type reference SEQUENCE
  DEFINED in MAP-ER-DataTypes : 241
  USED in MAP-Errors : 124 274
  USED in MAP-ER-DataTypes : 34

absentSubscriberReason.....identifier of [0] AbsentSubscriberReason
  DEFINED in MAP-ER-DataTypes : 244

```

```

AbsentSubscriberReason.....type reference ENUMERATED
  DEFINED in MAP-ER-DataTypes      :    246
  USED in MAP-ER-DataTypes         :    244

absentsubscriberSM.....value reference AbsentSubscriberSM, CHOICE VALUE
  DEFINED in MAP-Protocol           :    414

AbsentSubscriberSM.....type reference ERROR
  DEFINED in MAP-Errors             :    407
  USED in MAP-Protocol              :    155   414
  USED in MAP-ShortMessageService  :    40   79   109
  USED in MAP-Errors                :    79

absentSubscriberSM-Param.....identifier of AbsentSubscriberSM-Param
  DEFINED in MAP-Errors             :    409

AbsentSubscriberSM-Param.....type reference SEQUENCE
  DEFINED in MAP-ER-DataTypes       :    154
  USED in MAP-Errors                :    134   409
  USED in MAP-ER-DataTypes         :    42

accepted.....identifier of Named Number, 0
  DEFINED in MAP-CH-DataTypes       :    398

accessOutsideLSAsAllowed.....identifier of Named Number, 0
  DEFINED in MAP-MS-DataTypes       :    540

```

TAG R4.21 Cross Reference Listing for MAP-Protocol 00-01-06 07:32:40 PAGE 2

accessOutsideLSAsRestricted.....identifier of Named Number, 1
 DEFINED in MAP-MS-DataTypes : 541

activate.....identifier of Named Number, 1
 DEFINED in MAP-MS-DataTypes : 1527

activateSS.....value reference ActivateSS, CHOICE VALUE
 DEFINED in MAP-Protocol : 243

ActivateSS.....type reference OPERATION
 DEFINED in MAP-SupplementaryServi : 121
 USED in MAP-Protocol : 70 243
 USED in MAP-SupplementaryServi : 15

activateTraceMode.....value reference ActivateTraceMode, CHOICE VALUE
 DEFINED in MAP-Protocol : 220

ActivateTraceMode.....type reference OPERATION
 DEFINED in MAP-OperationAndMainte : 50
 USED in MAP-Protocol : 45 220
 USED in MAP-OperationAndMainte : 13

activateTraceModeArg.....identifier of ActivateTraceModeArg
 DEFINED in MAP-OperationAndMainte : 52

ActivateTraceModeArg.....type reference SEQUENCE
 DEFINED in MAP-OM-DataTypes : 36
 USED in MAP-OperationAndMainte : 34 52
 USED in MAP-OM-DataTypes : 14

activateTraceModeRes.....identifier of ActivateTraceModeRes
 DEFINED in MAP-OperationAndMainte : 54

ActivateTraceModeRes.....type reference SEQUENCE
 DEFINED in MAP-OM-DataTypes : 50
 USED in MAP-OperationAndMainte : 35 54
 USED in MAP-OM-DataTypes : 15

additionalAbsentSubscriberDiagnosticSM..identifier of [5] AbsentSubscriberDiagnosticSM
 DEFINED in MAP-SM-DataTypes : 159

additionalAbsentSubscriberDiagnosticSM..identifier of [0] AbsentSubscriberDiagnosticSM
 DEFINED in MAP-ER-DataTypes : 160

additionalSignalInfo.....identifier of [17] Ext-ExternalSignalInfo
 DEFINED in MAP-CH-DataTypes : 112

additionalSignalInfo.....identifier of [14] Ext-ExternalSignalInfo
 DEFINED in MAP-CH-DataTypes : 211

additionalSM-DeliveryOutcome.....identifier of [4] SM-DeliveryOutcome
 DEFINED in MAP-SM-DataTypes : 157

additional-Number.....identifier of [6] Additional-Number
 DEFINED in MAP-SM-DataTypes : 93

Additional-Number.....type reference CHOICE
 DEFINED in MAP-SM-DataTypes : 97
 USED in MAP-SM-DataTypes : 93

AddressString.....type reference OCTET STRING
 DEFINED in MAP-CommonDataTypes : 88
 USED in MAP-MS-DataTypes : 143 1511
 USED in MAP-CommonDataTypes : 16 132 332
 USED in MAP-OM-DataTypes : 21 40
 USED in MAP-SS-DataTypes : 44 72 277
 USED in MAP-SM-DataTypes : 31 55 135 140 145 176
 USED in MAP-LCS-DataTypes : 26 102

AgeIndicator.....type reference OCTET STRING
 DEFINED in MAP-MS-DataTypes : 206
 USED in MAP-MS-DataTypes : 204 411

ageOfLocationEstimate.....identifier of [0] AgeOfLocationInformation
 DEFINED in MAP-LCS-DataTypes : 166

ageOfLocationEstimate.....identifier of [6] AgeOfLocationInformation
 DEFINED in MAP-LCS-DataTypes : 226

ageOfLocationInformation.....identifier of AgeOfLocationInformation
 DEFINED in MAP-MS-DataTypes : 1353

TAG R4.21 Cross Reference Listing for MAP-Protocol 00-01-06 07:32:40 PAGE 3

AgeOfLocationInformation.....type reference INTEGER
 DEFINED in MAP-CommonDataTypes : 410
 USED in MAP-MS-DataTypes : 158 1353
 USED in MAP-CommonDataTypes : 51
 USED in MAP-LCS-DataTypes : 32 166 226

alertingCategory-1.....value reference AlertingPattern, '00000100'B
 DEFINED in MAP-CommonDataTypes : 239

alertingCategory-2.....value reference AlertingPattern, '00000101'B
 DEFINED in MAP-CommonDataTypes : 240

alertingCategory-3.....value reference AlertingPattern, '00000110'B
 DEFINED in MAP-CommonDataTypes : 241

alertingCategory-4.....value reference AlertingPattern, '00000111'B
 DEFINED in MAP-CommonDataTypes : 242

alertingCategory-5.....value reference AlertingPattern, '00001000'B
 DEFINED in MAP-CommonDataTypes : 243

alertingLevel-0.....value reference AlertingPattern, '00000000'B
 DEFINED in MAP-CommonDataTypes : 233

alertingLevel-1.....value reference AlertingPattern, '00000001'B
 DEFINED in MAP-CommonDataTypes : 234

alertingLevel-2.....value reference AlertingPattern, '00000010'B
 DEFINED in MAP-CommonDataTypes : 235

AlertingPattern.....type reference OCTET STRING
 DEFINED in MAP-CommonDataTypes : 220
 USED in MAP-CommonDataTypes : 24 233 234 235 239 240 241 242 243
 USED in MAP-CH-DataTypes : 70 109 208 388
 USED in MAP-SS-DataTypes : 49 214

alertingPattern.....identifier of [14] AlertingPattern
 DEFINED in MAP-CH-DataTypes : 109

alertingPattern.....identifier of [12] AlertingPattern
 DEFINED in MAP-CH-DataTypes : 208

alertingPattern.....identifier of [5] AlertingPattern
 DEFINED in MAP-CH-DataTypes : 388

alertingPattern.....identifier of AlertingPattern
 DEFINED in MAP-SS-DataTypes : 214

alertReason.....identifier of AlertReason
 DEFINED in MAP-SM-DataTypes : 195

AlertReason.....type reference ENUMERATED
 DEFINED in MAP-SM-DataTypes : 207
 USED in MAP-SM-DataTypes : 27 195

alertReasonIndicator.....identifier of NULL
 DEFINED in MAP-SM-DataTypes : 196

alertServiceCentre.....value reference AlertServiceCentre, CHOICE VALUE
 DEFINED in MAP-Protocol : 263

AlertServiceCentre.....type reference OPERATION
 DEFINED in MAP-ShortMessageServic : 123
 USED in MAP-Protocol : 90 263
 USED in MAP-ShortMessageServic : 17

alertServiceCentreArg.....identifier of AlertServiceCentreArg
 DEFINED in MAP-ShortMessageServic : 125

AlertServiceCentreArg.....type reference SEQUENCE
 DEFINED in MAP-SM-DataTypes : 174
 USED in MAP-ShortMessageServic : 53 125
 USED in MAP-SM-DataTypes : 22

allAdditionalInfoTransferSS.....value reference SS-Code, '10000000'B
 DEFINED in MAP-SS-Code : 105

allAlternateSpeech-DataCDA.....value reference BearerServiceCode, '00110000'B
 DEFINED in MAP-BS-Code : 82

allAlternateSpeech-DataCDS.....value reference BearerServiceCode, '00111000'B

DEFINED in MAP-BS-Code : 84

TAG R4.21 Cross Reference Listing for MAP-Protocol 00-01-06 07:32:40 PAGE 4

allAsynchronousServices.....value reference BearerServiceCode, '01100000'B
 DEFINED in MAP-BS-Code : 95

allBarringSS.....value reference SS-Code, '10010000'B
 DEFINED in MAP-SS-Code : 115

allBearerServices.....value reference BearerServiceCode, '00000000'B
 DEFINED in MAP-BS-Code : 49

allCallCompletionSS.....value reference SS-Code, '01000000'B
 DEFINED in MAP-SS-Code : 72

allCallOfferingSS.....value reference SS-Code, '00110000'B
 DEFINED in MAP-SS-Code : 63

allCallPrioritySS.....value reference SS-Code, '10100000'B
 DEFINED in MAP-SS-Code : 151

allChargingSS.....value reference SS-Code, '01110000'B
 DEFINED in MAP-SS-Code : 97

allCommunityOfInterest-SS.....value reference SS-Code, '01100000'B
 DEFINED in MAP-SS-Code : 91

allCondForwardingSS.....value reference SS-Code, '00101000'B
 DEFINED in MAP-SS-Code : 52

allDataCDA-Services.....value reference BearerServiceCode, '00010000'B
 DEFINED in MAP-BS-Code : 51

allDataCDS-Services.....value reference BearerServiceCode, '00011000'B
 DEFINED in MAP-BS-Code : 60

allDataCircuitAsynchronous.....value reference BearerServiceCode, '01010000'B
 DEFINED in MAP-BS-Code : 92

allDataCircuitSynchronous.....value reference BearerServiceCode, '01011000'B
 DEFINED in MAP-BS-Code : 98

allDataPDS-Services.....value reference BearerServiceCode, '00101000'B
 DEFINED in MAP-BS-Code : 76

allDataTeleservices.....value reference TeleserviceCode, '01110000'B
 DEFINED in MAP-TS-Code : 55

allECT-Barred.....identifier of Named Number, 9
 DEFINED in MAP-MS-DataTypes : 625

allFacsimileTransmissionServices.....value reference TeleserviceCode, '01100000'B
 DEFINED in MAP-TS-Code : 48

allForwardingSS.....value reference SS-Code, '00100000'B
 DEFINED in MAP-SS-Code : 48

allGPRSData.....identifier of NULL
 DEFINED in MAP-MS-DataTypes : 910

allInformationSent.....identifier of [11] NULL
 DEFINED in MAP-CH-DataTypes : 231

allLCSPrivacyException.....value reference SS-Code, '10110000'B
 DEFINED in MAP-SS-Code : 157

allLineIdentificationSS.....value reference SS-Code, '00010000'B
 DEFINED in MAP-SS-Code : 25

allLSAData.....identifier of NULL
 DEFINED in MAP-MS-DataTypes : 917

allMOLR-SS.....value reference SS-Code, '11000000'B
 DEFINED in MAP-SS-Code : 169

allMultiPartySS.....value reference SS-Code, '01010000'B
 DEFINED in MAP-SS-Code : 85

allNameIdentificationSS.....value reference SS-Code, '00011000'B
 DEFINED in MAP-SS-Code : 40

allOG-CallsBarred.....identifier of Named Number, 0
 DEFINED in MAP-MS-DataTypes : 616

TAG R4.21 Cross Reference Listing for MAP-Protocol 00-01-06 07:32:40 PAGE 5

allPadAccessCA-Services.....value reference BearerServiceCode, '00100000'B
 DEFINED in MAP-BS-Code : 67

allPLMN-specificBS.....value reference BearerServiceCode, '11010000'B
 DEFINED in MAP-BS-Code : 110

allPLMN-specificSS.....value reference SS-Code, '11110000'B
 DEFINED in MAP-SS-Code : 134

allPLMN-specificTS.....value reference TeleserviceCode, '11010000'B
 DEFINED in MAP-TS-Code : 72

allShortMessageServices.....value reference TeleserviceCode, '00100000'B
 DEFINED in MAP-TS-Code : 44

allSpeechFollowedByDataCDA.....value reference BearerServiceCode, '01000000'B
 DEFINED in MAP-BS-Code : 86

allSpeechFollowedByDataCDS.....value reference BearerServiceCode, '01001000'B
 DEFINED in MAP-BS-Code : 88

allSpeechTransmissionServices.....value reference TeleserviceCode, '00010000'B
 DEFINED in MAP-TS-Code : 40

allSS.....value reference SS-Code, '00000000'B
 DEFINED in MAP-SS-Code : 21

allSynchronousServices.....value reference BearerServiceCode, '01101000'B
 DEFINED in MAP-BS-Code : 101

allTeleservices.....value reference TeleserviceCode, '00000000'B
 DEFINED in MAP-TS-Code : 38

allTeleservices-ExeptSMS.....value reference TeleserviceCode, '10000000'B
 DEFINED in MAP-TS-Code : 58

allVoiceGroupCallServices.....value reference TeleserviceCode, '10010000'B
 DEFINED in MAP-TS-Code : 67

anonymousLocation.....identifier of Named Number, 3
 DEFINED in MAP-CommonDataTypes : 340

anyTimeInterrogation.....value reference AnyTimeInterrogation, CHOICE VALUE
 DEFINED in MAP-Protocol : 272

AnyTimeInterrogation.....type reference OPERATION
 DEFINED in MAP-MobileServiceOpera : 236
 USED in MAP-Protocol : 30 272
 USED in MAP-MobileServiceOpera : 27

anyTimeInterrogationArg.....identifier of AnyTimeInterrogationArg
 DEFINED in MAP-MobileServiceOpera : 238

AnyTimeInterrogationArg.....type reference SEQUENCE
 DEFINED in MAP-MS-DataTypes : 1403
 USED in MAP-MobileServiceOpera : 142 238
 USED in MAP-MS-DataTypes : 84

anyTimeInterrogationRes.....identifier of AnyTimeInterrogationRes
 DEFINED in MAP-MobileServiceOpera : 240

AnyTimeInterrogationRes.....type reference SEQUENCE
 DEFINED in MAP-MS-DataTypes : 1410
 USED in MAP-MobileServiceOpera : 143 240
 USED in MAP-MS-DataTypes : 85

anyTimeModification.....value reference AnyTimeModification, CHOICE VALUE
 DEFINED in MAP-Protocol : 277

AnyTimeModification.....type reference OPERATION
 DEFINED in MAP-MobileServiceOpera : 267
 USED in MAP-Protocol : 32 277
 USED in MAP-MobileServiceOpera : 31

anyTimeModificationArg.....identifier of AnyTimeModificationArg
 DEFINED in MAP-MobileServiceOpera : 269

AnyTimeModificationArg.....type reference SEQUENCE
 DEFINED in MAP-MS-DataTypes : 1493
 USED in MAP-MobileServiceOpera : 138 269
 USED in MAP-MS-DataTypes : 90

TAG R4.21 Cross Reference Listing for MAP-Protocol 00-01-06 07:32:40 PAGE 6

```

anyTimeModificationRes.....identifier of AnyTimeModificationRes
  DEFINED in MAP-MobileServiceOpera : 271

AnyTimeModificationRes.....type reference SEQUENCE
  DEFINED in MAP-MS-DataTypes : 1501
  USED in MAP-MobileServiceOpera : 139 271
  USED in MAP-MS-DataTypes : 91

anyTimeSubscriptionInterrogation.value reference AnyTimeSubscriptionInterrogation, CHOICE
VALUE
  DEFINED in MAP-Protocol : 276

AnyTimeSubscriptionInterrogation.....type reference OPERATION
  DEFINED in MAP-MobileServiceOpera : 250
  USED in MAP-Protocol : 31 276
  USED in MAP-MobileServiceOpera : 30

anyTimeSubscriptionInterrogationArg....identifier of AnyTimeSubscriptionInterrogationArg
  DEFINED in MAP-MobileServiceOpera : 252

AnyTimeSubscriptionInterrogationArg....type reference SEQUENCE
  DEFINED in MAP-MS-DataTypes : 1418
  USED in MAP-MobileServiceOpera : 136 252
  USED in MAP-MS-DataTypes : 88

anyTimeSubscriptionInterrogationRes....identifier of AnyTimeSubscriptionInterrogationRes
  DEFINED in MAP-MobileServiceOpera : 254

AnyTimeSubscriptionInterrogationRes....type reference SEQUENCE
  DEFINED in MAP-MS-DataTypes : 1425
  USED in MAP-MobileServiceOpera : 137 254
  USED in MAP-MS-DataTypes : 89

aocc.....value reference SS-Code, '01110010'B
  DEFINED in MAP-SS-Code : 102

aoci.....value reference SS-Code, '01110001'B
  DEFINED in MAP-SS-Code : 100

apn.....identifier of [20] APN
  DEFINED in MAP-MS-DataTypes : 450

APN.....type reference OCTET STRING
  DEFINED in MAP-MS-DataTypes : 518
  USED in MAP-MS-DataTypes : 450

asciiCallReference.....identifier of ASCII-CallReference
  DEFINED in MAP-GR-DataTypes : 51

ASCII-CallReference.....type reference TBCD-STRING
  DEFINED in MAP-CommonDataTypes : 263
  USED in MAP-CommonDataTypes : 38
  USED in MAP-GR-DataTypes : 26 51

assumedIdle.....identifier of [0] NULL
  DEFINED in MAP-MS-DataTypes : 1390

ati-NotAllowed.....value reference ATI-NotAllowed, CHOICE VALUE
  DEFINED in MAP-Protocol : 380

ATI-NotAllowed.....type reference ERROR
  DEFINED in MAP-Errors : 316
  USED in MAP-Protocol : 141 380
  USED in MAP-MobileServiceOpera : 89 243
  USED in MAP-Errors : 54

ati-NotAllowedParam.....identifier of ATI-NotAllowedParam
  DEFINED in MAP-Errors : 318

ATI-NotAllowedParam.....type reference SEQUENCE
  DEFINED in MAP-ER-DataTypes : 277
  USED in MAP-Errors : 131 318
  USED in MAP-ER-DataTypes : 39

atm-NotAllowed.....value reference ATM-NotAllowed, CHOICE VALUE
  DEFINED in MAP-Protocol : 384

ATM-NotAllowed.....type reference ERROR
  DEFINED in MAP-Errors : 327
  USED in MAP-Protocol : 166 384
  USED in MAP-MobileServiceOpera : 95 273

```

USED in MAP-Errors : 58

TAG R4.21 Cross Reference Listing for MAP-Protocol 00-01-06 07:32:40 PAGE 7

atm-NotAllowedParam.....identifier of ATM-NotAllowedParam
 DEFINED in MAP-Errors : 329

ATM-NotAllowedParam.....type reference SEQUENCE
 DEFINED in MAP-ER-DataTypes : 285
 USED in MAP-Errors : 146 329
 USED in MAP-ER-DataTypes : 55

atsi-NotAllowed.....value reference ATSI-NotAllowed, CHOICE VALUE
 DEFINED in MAP-Protocol : 383

ATSI-NotAllowed.....type reference ERROR
 DEFINED in MAP-Errors : 322
 USED in MAP-Protocol : 165 383
 USED in MAP-MobileServiceOpera : 94 256
 USED in MAP-Errors : 57

atsi-NotAllowedParam.....identifier of ATSI-NotAllowedParam
 DEFINED in MAP-Errors : 324

ATSI-NotAllowedParam.....type reference SEQUENCE
 DEFINED in MAP-ER-DataTypes : 281
 USED in MAP-Errors : 145 324
 USED in MAP-ER-DataTypes : 54

attach.....identifier of Named Number, 1
 DEFINED in MAP-MS-DataTypes : 507

attachChangeOfPosition.....identifier of Named Number, 2
 DEFINED in MAP-MS-DataTypes : 508

AuthenticationQuintuplet.....type reference SEQUENCE
 DEFINED in MAP-MS-DataTypes : 288
 USED in MAP-MS-DataTypes : 280

authenticationSetList.....identifier of AuthenticationSetList
 DEFINED in MAP-MS-DataTypes : 268

AuthenticationSetList.....type reference CHOICE
 DEFINED in MAP-MS-DataTypes : 272
 USED in MAP-MS-DataTypes : 268 381

authenticationSetList.....identifier of AuthenticationSetList
 DEFINED in MAP-MS-DataTypes : 381

AuthenticationTriplet.....type reference SEQUENCE
 DEFINED in MAP-MS-DataTypes : 282
 USED in MAP-MS-DataTypes : 277

autn.....identifier of AUTN
 DEFINED in MAP-MS-DataTypes : 293

AUTN.....type reference OCTET STRING
 DEFINED in MAP-MS-DataTypes : 308
 USED in MAP-MS-DataTypes : 293

automaticFacsimileGroup3.....value reference TeleserviceCode, '01100010'B
 DEFINED in MAP-TS-Code : 50

autonomousSelfLocation.....value reference SS-Code, '11000010'B
 DEFINED in MAP-SS-Code : 173

AUTS.....type reference OCTET STRING
 DEFINED in MAP-MS-DataTypes : 310
 USED in MAP-MS-DataTypes : 377

auts.....identifier of AUTS
 DEFINED in MAP-MS-DataTypes : 377

a-side.....identifier of Named Number, 0
 DEFINED in MAP-CH-DataTypes : 361

badlyFormattedTransactionPortion.....identifier of Named Number, 2
 DEFINED in TCAPMessages : 105

badlyStructuredComponent.....identifier of Named Number, 2
 DEFINED in TCAPMessages : 181

baic.....value reference SS-Code, '10011010'B
 DEFINED in MAP-SS-Code : 128

baoc.....value reference SS-Code, '10010010'B

TAG R4.21 Cross Reference Listing for MAP-Protocol 00-01-06 07:32:40 PAGE 8

DEFINED in MAP-SS-Code : 119

barringOfIncomingCalls.....value reference SS-Code, '10011001'B
DEFINED in MAP-SS-Code : 126

barringOfOutgoingCalls.....value reference SS-Code, '10010001'B
DEFINED in MAP-SS-Code : 117

barringServiceActive.....identifier of Named Number, 0
DEFINED in MAP-ER-DataTypes : 107

basicCall.....identifier of Named Number, 0
DEFINED in MAP-CH-DataTypes : 120

basicISTSupported.....identifier of Named Number, 0
DEFINED in MAP-MS-DataTypes : 211

basicSelfLocation.....value reference SS-Code, '11000001'B
DEFINED in MAP-SS-Code : 171

basicService.....identifier of Ext-BasicServiceCode
DEFINED in MAP-MS-DataTypes : 663

basicService.....identifier of Ext-BasicServiceCode
DEFINED in MAP-MS-DataTypes : 742

basicService.....identifier of Ext-BasicServiceCode
DEFINED in MAP-MS-DataTypes : 785

basicService.....identifier of [1] Ext-BasicServiceCode
DEFINED in MAP-MS-DataTypes : 1509

basicService.....identifier of [5] Ext-BasicServiceCode
DEFINED in MAP-CH-DataTypes : 153

basicService.....identifier of BasicServiceCode
DEFINED in MAP-SS-DataTypes : 71

basicService.....identifier of BasicServiceCode
DEFINED in MAP-SS-DataTypes : 95

basicService.....identifier of BasicServiceCode
DEFINED in MAP-SS-DataTypes : 151

basicService.....identifier of BasicServiceCode
DEFINED in MAP-SS-DataTypes : 179

basicService.....identifier of BasicServiceCode
DEFINED in MAP-ER-DataTypes : 129

BasicServiceCode.....type reference CHOICE
DEFINED in MAP-CommonDataTypes : 379
USED in MAP-CommonDataTypes : 45
USED in MAP-SS-DataTypes : 48 71 95 151 179 199 254
USED in MAP-ER-DataTypes : 73 129

basicServiceCriteria.....identifier of [1] BasicServiceCriteria
DEFINED in MAP-MS-DataTypes : 1045

basicServiceCriteria.....identifier of [0] BasicServiceCriteria
DEFINED in MAP-MS-DataTypes : 1053

BasicServiceCriteria.....type reference SEQUENCE OF
DEFINED in MAP-MS-DataTypes : 1073
USED in MAP-MS-DataTypes : 60 1045 1053

basicServiceGroup.....identifier of [9] Ext-BasicServiceCode
DEFINED in MAP-CH-DataTypes : 103

basicServiceGroup.....identifier of [1] Ext-BasicServiceCode
DEFINED in MAP-CH-DataTypes : 222

basicServiceGroup.....identifier of [3] BasicServiceCode
DEFINED in MAP-SS-DataTypes : 199

basicServiceGroupList.....identifier of Ext-BasicServiceGroupList
DEFINED in MAP-MS-DataTypes : 760

basicServiceGroupList.....identifier of Ext-BasicServiceGroupList
DEFINED in MAP-MS-DataTypes : 807

basicServiceGroupList.....identifier of BasicServiceGroupList

```

TAG   R4.21   Cross Reference Listing for MAP-Protocol   00-01-06  07:32:40  PAGE   9

    DEFINED in MAP-SS-DataTypes           :    159

basicServiceGroupList.....identifier of [2] BasicServiceGroupList
    DEFINED in MAP-SS-DataTypes           :    206

BasicServiceGroupList.....type reference SEQUENCE OF
    DEFINED in MAP-SS-DataTypes           :    253
    USED in MAP-SS-DataTypes              :    159    206

basicServiceList.....identifier of [1] BasicServiceList
    DEFINED in MAP-MS-DataTypes           :    892

BasicServiceList.....type reference SEQUENCE OF
    DEFINED in MAP-MS-DataTypes           :    923
    USED in MAP-MS-DataTypes              :    892

bearerService.....identifier of [2] BearerServiceCode
    DEFINED in MAP-CommonDataTypes        :    380

BearerServiceCode.....type reference OCTET STRING
    DEFINED in MAP-BS-Code                 :     11
    USED in MAP-CommonDataTypes            :     63    380
    USED in MAP-BS-Code                    :     49    51    52    53    54    55    56    57    58
                                           :     60    61    62    63    64    65    67    68    69
                                           :     70    71    72    73    74    76    77    78    79
                                           :     80    82    84    86    88    92    95    98    101
                                           :    110   111   112   113   114   115   116   117   118
                                           :    119   120   121   122   123   124   125

bearerServiceList.....identifier of [4] BearerServiceList
    DEFINED in MAP-MS-DataTypes           :    577

BearerServiceList.....type reference SEQUENCE OF
    DEFINED in MAP-MS-DataTypes           :    599
    USED in MAP-MS-DataTypes              :    577    875

bearerServiceList.....identifier of [2] BearerServiceList
    DEFINED in MAP-MS-DataTypes           :    875

bearerServiceNotProvisioned.....value reference BearerServiceNotProvisioned, CHOICE
VALUE
    DEFINED in MAP-Protocol                 :    347

BearerServiceNotProvisioned.....type reference ERROR
    DEFINED in MAP-Errors                   :    237
    USED in MAP-Protocol                    :    127    347
    USED in MAP-MobileServiceOpera         :     96    260    277
    USED in MAP-CallHandlingOperat        :     37     94
    USED in MAP-SupplementaryServi         :     37     97    114    131    151    169
    USED in MAP-Errors                      :     32

bearerServNotProvParam.....identifier of BearerServNotProvParam
    DEFINED in MAP-Errors                   :    239

BearerServNotProvParam.....type reference SEQUENCE
    DEFINED in MAP-ER-DataTypes             :    225
    USED in MAP-Errors                      :    119    239
    USED in MAP-ER-DataTypes               :     30

begin.....identifier of [APPLICATION 2] IMPLICIT Begin
    DEFINED in TCAPMessages                 :     53

Begin.....type reference SEQUENCE
    DEFINED in TCAPMessages                 :     61
    USED in TCAPMessages                   :     53

bicRoam.....value reference SS-Code, '10011011'B
    DEFINED in MAP-SS-Code                  :    130

blackListed.....identifier of Named Number, 1
    DEFINED in MAP-MS-DataTypes            :    390

boic.....value reference SS-Code, '10010011'B
    DEFINED in MAP-SS-Code                  :    121

boicExHC.....value reference SS-Code, '10010100'B
    DEFINED in MAP-SS-Code                  :    123

bothMSCAndSGSN.....identifier of Named Number, 0
    DEFINED in MAP-MS-DataTypes            :    432
    
```

broadcastInitEntitlement.....identifier of NULL
DEFINED in MAP-MS-DataTypes : 1318

TAG R4.21 Cross Reference Listing for MAP-Protocol 00-01-06 07:32:40 PAGE 10

broadcastService.....identifier of Named Number, 0
 DEFINED in MAP-CommonDataTypes : 337

bss-APDU.....identifier of ExternalSignalInfo
 DEFINED in MAP-MobileServiceOpera : 314

bss-APDU.....identifier of ExternalSignalInfo
 DEFINED in MAP-MobileServiceOpera : 319

bss-APDU.....identifier of ExternalSignalInfo
 DEFINED in MAP-MobileServiceOpera : 323

bss-APDU.....identifier of ExternalSignalInfo
 DEFINED in MAP-MobileServiceOpera : 329

bss-APDU.....identifier of ExternalSignalInfo
 DEFINED in MAP-MS-DataTypes : 343

bss-APDU.....identifier of ExternalSignalInfo
 DEFINED in MAP-MS-DataTypes : 348

bss-APDU.....identifier of ExternalSignalInfo
 DEFINED in MAP-MS-DataTypes : 354

busy.....identifier of Named Number, 1
 DEFINED in MAP-CH-DataTypes : 129

busy.....identifier of Named Number, 2
 DEFINED in MAP-CH-DataTypes : 371

busySubscriber.....value reference BusySubscriber, CHOICE VALUE
 DEFINED in MAP-Protocol : 370

BusySubscriber.....type reference ERROR
 DEFINED in MAP-Errors : 279
 USED in MAP-Protocol : 135 370
 USED in MAP-CallHandlingOperat : 41 97 190
 USED in MAP-Errors : 45

busySubscriberParam.....identifier of BusySubscriberParam
 DEFINED in MAP-Errors : 281

BusySubscriberParam.....type reference SEQUENCE
 DEFINED in MAP-ER-DataTypes : 259
 USED in MAP-Errors : 125 281
 USED in MAP-ER-DataTypes : 35

b-side.....identifier of Named Number, 1
 DEFINED in MAP-CH-DataTypes : 362

b-subscriberNumber.....identifier of [1] ISDN-AddressString
 DEFINED in MAP-SS-DataTypes : 197

b-subscriberSubaddress.....identifier of [2] ISDN-SubaddressString
 DEFINED in MAP-SS-DataTypes : 198

b-Subscriber-Address.....identifier of [3] ISDN-AddressString
 DEFINED in MAP-CH-DataTypes : 281

callBarred.....value reference CallBarred, CHOICE VALUE
 DEFINED in MAP-Protocol : 372

CallBarred.....type reference ERROR
 DEFINED in MAP-Errors : 289
 USED in MAP-Protocol : 137 372
 USED in MAP-MobileServiceOpera : 98 262 279
 USED in MAP-CallHandlingOperat : 43 99
 USED in MAP-SupplementaryServi : 39 99 116 133 153 171 185 226 260
 277
 USED in MAP-ShortMessageServic : 36 78
 USED in MAP-Errors : 48

callBarredParam.....identifier of CallBarredParam
 DEFINED in MAP-Errors : 291

CallBarredParam.....type reference CHOICE
 DEFINED in MAP-ER-DataTypes : 99
 USED in MAP-Errors : 127 291
 USED in MAP-ER-DataTypes : 15

callBarringCause.....identifier of CallBarringCause

TAG	R4.21	Cross Reference Listing for MAP-Protocol	00-01-06	07:32:40	PAGE 11
	DEFINED in MAP-ER-DataTypes	:	100		
CallBarringCause.....	type reference ENUMERATED				
	DEFINED in MAP-ER-DataTypes	:	106		
	USED in MAP-ER-DataTypes	:	100 111		
callBarringCause.....	identifier of CallBarringCause				
	DEFINED in MAP-ER-DataTypes	:	111		
callBarringData.....	identifier of [2] CallBarringData				
	DEFINED in MAP-MS-DataTypes	:	1427		
CallBarringData.....	type reference SEQUENCE				
	DEFINED in MAP-MS-DataTypes	:	1461		
	USED in MAP-MS-DataTypes	:	1427		
CallBarringFeature.....	type reference SEQUENCE				
	DEFINED in MAP-SS-DataTypes	:	150		
	USED in MAP-SS-DataTypes	:	148		
callBarringFeatureList.....	identifier of Ext-CallBarFeatureList				
	DEFINED in MAP-MS-DataTypes	:	734		
callBarringFeatureList.....	identifier of Ext-CallBarFeatureList				
	DEFINED in MAP-MS-DataTypes	:	1462		
callBarringFeatureList.....	identifier of [1] Ext-CallBarFeatureList				
	DEFINED in MAP-MS-DataTypes	:	1583		
callBarringFeatureList.....	identifier of CallBarringFeatureList				
	DEFINED in MAP-SS-DataTypes	:	144		
CallBarringFeatureList.....	type reference SEQUENCE OF				
	DEFINED in MAP-SS-DataTypes	:	147		
	USED in MAP-SS-DataTypes	:	144		
callBarringInfo.....	identifier of [1] Ext-CallBarInfo				
	DEFINED in MAP-MS-DataTypes	:	647		
callBarringInfo.....	identifier of [1] CallBarringInfo				
	DEFINED in MAP-SS-DataTypes	:	82		
CallBarringInfo.....	type reference SEQUENCE				
	DEFINED in MAP-SS-DataTypes	:	142		
	USED in MAP-SS-DataTypes	:	82		
callBarringInfoFor-CSE.....	identifier of [1] Ext-CallBarringInfoFor-CSE				
	DEFINED in MAP-MS-DataTypes	:	1571		
callBarringSS-Data.....	identifier of Named Number, 1				
	DEFINED in MAP-MS-DataTypes	:	1544		
CallDirection.....	type reference OCTET STRING				
	DEFINED in MAP-CH-DataTypes	:	288		
	USED in MAP-CH-DataTypes	:	280		
callDiversionTreatmentIndicator.....	identifier of [20] CallDiversionTreatmentIndicator				
	DEFINED in MAP-CH-DataTypes	:	115		
CallDiversionTreatmentIndicator.....	type reference OCTET STRING				
	DEFINED in MAP-CH-DataTypes	:	138		
	USED in MAP-CH-DataTypes	:	115		
calledPartySS-InteractionViolation.....	identifier of Named Number, 7				
	DEFINED in MAP-ER-DataTypes	:	125		
callForwardingData.....	identifier of [1] CallForwardingData				
	DEFINED in MAP-MS-DataTypes	:	1426		
CallForwardingData.....	type reference SEQUENCE				
	DEFINED in MAP-MS-DataTypes	:	1455		
	USED in MAP-MS-DataTypes	:	1426		
callForwardingSS-Data.....	identifier of Named Number, 0				
	DEFINED in MAP-MS-DataTypes	:	1543		
callInfo.....	identifier of [1] ExternalSignalInfo				
	DEFINED in MAP-CH-DataTypes	:	384		
callInfo.....	identifier of [3] ExternalSignalInfo				
	DEFINED in MAP-SS-DataTypes	:	290		


```

TAG R4.21 Cross Reference Listing for MAP-Protocol 00-01-06 07:32:40 PAGE 12

callOutcome.....identifier of [1] CallOutcome
  DEFINED in MAP-CH-DataTypes : 356

CallOutcome.....type reference ENUMERATED
  DEFINED in MAP-CH-DataTypes : 368
  USED in MAP-CH-DataTypes : 356

callReferenceNumber.....identifier of [7] CallReferenceNumber
  DEFINED in MAP-CH-DataTypes : 101

CallReferenceNumber.....type reference OCTET STRING
  DEFINED in MAP-CH-DataTypes : 125
  USED in MAP-CH-DataTypes : 22 101 204 221

callReferenceNumber.....identifier of [9] CallReferenceNumber
  DEFINED in MAP-CH-DataTypes : 204

callReferenceNumber.....identifier of [0] CallReferenceNumber
  DEFINED in MAP-CH-DataTypes : 221

callrelated.....value reference SS-Code, '10110010'B
  DEFINED in MAP-SS-Code : 161

callReportdata.....identifier of [2] CallReportData
  DEFINED in MAP-CH-DataTypes : 345

CallReportData.....type reference SEQUENCE
  DEFINED in MAP-CH-DataTypes : 354
  USED in MAP-CH-DataTypes : 345

callTerminationIndicator.....identifier of [2] CallTerminationIndicator
  DEFINED in MAP-CH-DataTypes : 421

CallTerminationIndicator.....type reference ENUMERATED
  DEFINED in MAP-CH-DataTypes : 434
  USED in MAP-CH-DataTypes : 421

callToClientNotSetup.....identifier of Named Number, 2
  DEFINED in MAP-ER-DataTypes : 346

callTypeCriteria.....identifier of [2] CallTypeCriteria
  DEFINED in MAP-MS-DataTypes : 1046

CallTypeCriteria.....type reference ENUMERATED
  DEFINED in MAP-MS-DataTypes : 1084
  USED in MAP-MS-DataTypes : 1046

callunrelated.....value reference SS-Code, '10110011'B
  DEFINED in MAP-SS-Code : 164

call-Direction.....identifier of [2] CallDirection
  DEFINED in MAP-CH-DataTypes : 280

camelBusy.....identifier of [1] NULL
  DEFINED in MAP-MS-DataTypes : 1391

camelCapabilityHandling.....identifier of [1] CamelCapabilityHandling
  DEFINED in MAP-MS-DataTypes : 476

camelCapabilityHandling.....identifier of CamelCapabilityHandling
  DEFINED in MAP-MS-DataTypes : 950

camelCapabilityHandling.....identifier of [0] CamelCapabilityHandling
  DEFINED in MAP-MS-DataTypes : 997

CamelCapabilityHandling.....type reference INTEGER
  DEFINED in MAP-MS-DataTypes : 1115
  USED in MAP-MS-DataTypes : 59 476 950 997 1130 1201

camelCapabilityHandling.....identifier of [1] CamelCapabilityHandling
  DEFINED in MAP-MS-DataTypes : 1130

camelCapabilityHandling.....identifier of [0] CamelCapabilityHandling
  DEFINED in MAP-MS-DataTypes : 1201

camelInfo.....identifier of [11] CamelInfo
  DEFINED in MAP-CH-DataTypes : 105

CamelInfo.....type reference SEQUENCE
  DEFINED in MAP-CH-DataTypes : 252
  USED in MAP-CH-DataTypes : 105
    
```


TAG R4.21 Cross Reference Listing for MAP-Protocol 00-01-06 07:32:40 PAGE 13

camelRoutingInfo.....identifier of [8] CamelRoutingInfo
 DEFINED in MAP-CH-DataTypes : 260

CamelRoutingInfo.....type reference SEQUENCE
 DEFINED in MAP-CH-DataTypes : 262
 USED in MAP-CH-DataTypes : 260

camelSubscriptionInformation.....identifier of Named Number, 3
 DEFINED in MAP-MS-DataTypes : 1546

camelSubscriptionInfoWithdraw.....identifier of [9] NULL
 DEFINED in MAP-MS-DataTypes : 900

camel-invoked.....identifier of Named Number, 1
 DEFINED in MAP-SS-DataTypes : 296

camel-SubscriptionInfo.....identifier of [4] CAMEL-SubscriptionInfo
 DEFINED in MAP-MS-DataTypes : 1429

CAMEL-SubscriptionInfo.....type reference SEQUENCE
 DEFINED in MAP-MS-DataTypes : 1477
 USED in MAP-MS-DataTypes : 1429 1503

camel-SubscriptionInfo.....identifier of [1] CAMEL-SubscriptionInfo
 DEFINED in MAP-MS-DataTypes : 1503

cancellationType.....identifier of CancellationType
 DEFINED in MAP-MS-DataTypes : 225

CancellationType.....type reference ENUMERATED
 DEFINED in MAP-MS-DataTypes : 230
 USED in MAP-MS-DataTypes : 225

cancelLocation.....value reference CancelLocation, CHOICE VALUE
 DEFINED in MAP-Protocol : 179

CancelLocation.....type reference OPERATION
 DEFINED in MAP-MobileServiceOpera : 180
 USED in MAP-Protocol : 13 179
 USED in MAP-MobileServiceOpera : 16

cancelLocationArg.....identifier of CancelLocationArg
 DEFINED in MAP-MobileServiceOpera : 182

CancelLocationArg.....type reference [3] SEQUENCE
 DEFINED in MAP-MS-DataTypes : 223
 USED in MAP-MobileServiceOpera : 113 182
 USED in MAP-MS-DataTypes : 18

cancelLocationRes.....identifier of CancelLocationRes
 DEFINED in MAP-MobileServiceOpera : 184

CancelLocationRes.....type reference SEQUENCE
 DEFINED in MAP-MS-DataTypes : 237
 USED in MAP-MobileServiceOpera : 114 184
 USED in MAP-MS-DataTypes : 19

category.....identifier of [2] Category
 DEFINED in MAP-MS-DataTypes : 575

Category.....type reference OCTET STRING
 DEFINED in MAP-MS-DataTypes : 592
 USED in MAP-MS-DataTypes : 575

CauseValue.....type reference OCTET STRING
 DEFINED in MAP-MS-DataTypes : 1102
 USED in MAP-MS-DataTypes : 1093 1096

ccbsIdle.....identifier of Named Number, 1
 DEFINED in MAP-CH-DataTypes : 334

ccbsNotIdle.....identifier of Named Number, 0
 DEFINED in MAP-CH-DataTypes : 333

ccbsNotReachable.....identifier of Named Number, 2
 DEFINED in MAP-CH-DataTypes : 335

ccbs-A.....value reference SS-Code, '01000011'B
 DEFINED in MAP-SS-Code : 79

ccbs-B.....value reference SS-Code, '01000100'B

DEFINED in MAP-SS-Code : 81

TAG R4.21 Cross Reference Listing for MAP-Protocol 00-01-06 07:32:40 PAGE 14

ccbs-Busy.....identifier of [1] NULL
 DEFINED in MAP-ER-DataTypes : 263

ccbs-Call.....identifier of [15] NULL
 DEFINED in MAP-CH-DataTypes : 110

ccbs-Call.....identifier of [13] NULL
 DEFINED in MAP-CH-DataTypes : 209

ccbs-Data.....identifier of [1] CCBS-Data
 DEFINED in MAP-SS-DataTypes : 283

CCBS-Data.....type reference SEQUENCE
 DEFINED in MAP-SS-DataTypes : 286
 USED in MAP-SS-DataTypes : 283

ccbs-Feature.....identifier of [2] CCBS-Feature
 DEFINED in MAP-CH-DataTypes : 385

CCBS-Feature.....type reference SEQUENCE
 DEFINED in MAP-SS-DataTypes : 195
 USED in MAP-CH-DataTypes : 58 385
 USED in MAP-SS-DataTypes : 36 191 287 301

ccbs-Feature.....identifier of [0] CCBS-Feature
 DEFINED in MAP-SS-DataTypes : 287

ccbs-Feature.....identifier of [0] CCBS-Feature
 DEFINED in MAP-SS-DataTypes : 301

ccbs-FeatureList.....identifier of [2] CCBS-FeatureList
 DEFINED in MAP-SS-DataTypes : 188

CCBS-FeatureList.....type reference SEQUENCE OF
 DEFINED in MAP-SS-DataTypes : 190
 USED in MAP-SS-DataTypes : 188

ccbs-Index.....identifier of [0] CCBS-Index
 DEFINED in MAP-SS-DataTypes : 196

CCBS-Index.....type reference INTEGER
 DEFINED in MAP-SS-DataTypes : 202
 USED in MAP-SS-DataTypes : 196 306

ccbs-Index.....identifier of [1] CCBS-Index
 DEFINED in MAP-SS-DataTypes : 306

ccbs-Indicators.....identifier of [11] CCBS-Indicators
 DEFINED in MAP-CH-DataTypes : 160

CCBS-Indicators.....type reference SEQUENCE
 DEFINED in MAP-CH-DataTypes : 175
 USED in MAP-CH-DataTypes : 160

ccbs-Monitoring.....identifier of [2] ReportingState
 DEFINED in MAP-CH-DataTypes : 315

ccbs-Possible.....identifier of [0] NULL
 DEFINED in MAP-CH-DataTypes : 176

ccbs-Possible.....identifier of [8] NULL
 DEFINED in MAP-CH-DataTypes : 228

ccbs-Possible.....identifier of [0] NULL
 DEFINED in MAP-ER-DataTypes : 262

ccbs-SubscriberStatus.....identifier of [0] CCBS-SubscriberStatus
 DEFINED in MAP-CH-DataTypes : 328

CCBS-SubscriberStatus.....type reference ENUMERATED
 DEFINED in MAP-CH-DataTypes : 332
 USED in MAP-CH-DataTypes : 328 350

ccbs-SubscriberStatus.....identifier of [0] CCBS-SubscriberStatus
 DEFINED in MAP-CH-DataTypes : 350

cd.....value reference SS-Code, '00100100'B
 DEFINED in MAP-SS-Code : 60

cellIdFixedLength.....identifier of [0] CellIdFixedLength

DEFINED in MAP-CommonDataTypes : 348

TAG R4.21 Cross Reference Listing for MAP-Protocol 00-01-06 07:32:40 PAGE 15

CellIdFixedLength.....type reference OCTET STRING
 DEFINED in MAP-CommonDataTypes : 351
 USED in MAP-CommonDataTypes : 348

cellIdOrLAI.....identifier of [3] CellIdOrLAI
 DEFINED in MAP-MS-DataTypes : 1357

CellIdOrLAI.....type reference CHOICE
 DEFINED in MAP-CommonDataTypes : 347
 USED in MAP-MS-DataTypes : 153 1357
 USED in MAP-CommonDataTypes : 42

cfb.....value reference SS-Code, '00101001'B
 DEFINED in MAP-SS-Code : 54

cfnrc.....value reference SS-Code, '00101011'B
 DEFINED in MAP-SS-Code : 58

cfnry.....value reference SS-Code, '00101010'B
 DEFINED in MAP-SS-Code : 56

cfu.....value reference SS-Code, '00100001'B
 DEFINED in MAP-SS-Code : 50

channelType.....identifier of [0] ExternalSignalInfo
 DEFINED in MAP-CH-DataTypes : 302

chargeableECT-Barred.....identifier of Named Number, 10
 DEFINED in MAP-MS-DataTypes : 626

checkIMEI.....value reference CheckIMEI, CHOICE VALUE
 DEFINED in MAP-Protocol : 201

CheckIMEI.....type reference OPERATION
 DEFINED in MAP-MobileServiceOpera : 352
 USED in MAP-Protocol : 23 201
 USED in MAP-MobileServiceOpera : 48

chosenChannel.....identifier of [4] ExternalSignalInfo
 DEFINED in MAP-CH-DataTypes : 282

chosenChannel.....identifier of [1] ExternalSignalInfo
 DEFINED in MAP-CH-DataTypes : 303

chosenChannel.....identifier of [0] ExternalSignalInfo
 DEFINED in MAP-CH-DataTypes : 308

cipheringAlgorithm.....identifier of CipheringAlgorithm
 DEFINED in MAP-GR-DataTypes : 53

CipheringAlgorithm.....type reference OCTET STRING
 DEFINED in MAP-GR-DataTypes : 99
 USED in MAP-GR-DataTypes : 53

ck.....identifier of CK
 DEFINED in MAP-MS-DataTypes : 291

CK.....type reference OCTET STRING
 DEFINED in MAP-MS-DataTypes : 304
 USED in MAP-MS-DataTypes : 291

clientIdentity.....identifier of LCSCClientExternalID
 DEFINED in MAP-MS-DataTypes : 840

clientNotInMSPrivacyExceptionList.....identifier of Named Number, 1
 DEFINED in MAP-ER-DataTypes : 345

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 : 165

CliRestrictionOption.....type reference ENUMERATED
 DEFINED in MAP-SS-DataTypes : 168
 USED in MAP-SS-DataTypes : 29 165 184

cliRestrictionOption.....identifier of CliRestrictionOption

TAG R4.21 Cross Reference Listing for MAP-Protocol 00-01-06 07:32:40 PAGE 16

DEFINED in MAP-SS-DataTypes : 184

clir-invoked.....identifier of Named Number, 0
 DEFINED in MAP-SS-DataTypes : 295

cnap.....value reference SS-Code, '00011001'B
 DEFINED in MAP-SS-Code : 42

codec-Info.....identifier of CODEC-Info
 DEFINED in MAP-GR-DataTypes : 52

CODEC-Info.....type reference OCTET STRING
 DEFINED in MAP-GR-DataTypes : 94
 USED in MAP-GR-DataTypes : 52

collectedInfo.....identifier of Named Number, 2
 DEFINED in MAP-MS-DataTypes : 1025

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

completeDataListIncluded.....identifier of NULL
 DEFINED in MAP-MS-DataTypes : 459

completeDataListIncluded.....identifier of NULL
 DEFINED in MAP-MS-DataTypes : 557

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

congestion.....identifier of Named Number, 0
 DEFINED in MAP-ER-DataTypes : 359

ContextId.....type reference INTEGER
 DEFINED in MAP-MS-DataTypes : 456
 USED in MAP-MS-DataTypes : 445 914

contextIdList.....identifier of ContextIdList
 DEFINED in MAP-MS-DataTypes : 911

ContextIdList.....type reference SEQUENCE OF
 DEFINED in MAP-MS-DataTypes : 913
 USED in MAP-MS-DataTypes : 911

Continue.....type reference SEQUENCE
 DEFINED in TCAPMessages : 69
 USED in TCAPMessages : 55

continueCall.....identifier of Named Number, 0
 DEFINED in MAP-MS-DataTypes : 1108

continueTransaction.....identifier of Named Number, 0
 DEFINED in MAP-MS-DataTypes : 499

continueTransaction.....identifier of Named Number, 0
 DEFINED in MAP-MS-DataTypes : 1161

continue-ME.....identifier of [APPLICATION 5] IMPLICIT Continue
 DEFINED in TCAPMessages : 55

controllingMSC.....identifier of Named Number, 4

DEFINED in MAP-CommonDataTypes : 309

TAG R4.21 Cross Reference Listing for MAP-Protocol 00-01-06 07:32:40 PAGE 17

csiActive.....identifier of [4] NULL
 DEFINED in MAP-MS-DataTypes : 479

csiActive.....identifier of [2] NULL
 DEFINED in MAP-MS-DataTypes : 978

csiActive.....identifier of [2] NULL
 DEFINED in MAP-MS-DataTypes : 999

csiActive.....identifier of [4] NULL
 DEFINED in MAP-MS-DataTypes : 1133

csiActive.....identifier of [3] NULL
 DEFINED in MAP-MS-DataTypes : 1174

csi-Active.....identifier of [2] NULL
 DEFINED in MAP-MS-DataTypes : 1203

cug.....value reference SS-Code, '01100001'B
 DEFINED in MAP-SS-Code : 94

cugIC-CallBarred.....identifier of Named Number, 1
 DEFINED in MAP-MS-DataTypes : 771

cugOG-CallBarred.....identifier of Named Number, 2
 DEFINED in MAP-MS-DataTypes : 772

cugSubscriptionFlag.....identifier of [6] NULL
 DEFINED in MAP-CH-DataTypes : 150

CUG-CheckInfo.....type reference SEQUENCE
 DEFINED in MAP-CH-DataTypes : 85
 USED in MAP-CH-DataTypes : 95 149 225

cug-CheckInfo.....identifier of [1] CUG-CheckInfo
 DEFINED in MAP-CH-DataTypes : 95

cug-CheckInfo.....identifier of [3] CUG-CheckInfo
 DEFINED in MAP-CH-DataTypes : 149

cug-CheckInfo.....identifier of [4] CUG-CheckInfo
 DEFINED in MAP-CH-DataTypes : 225

CUG-Feature.....type reference SEQUENCE
 DEFINED in MAP-MS-DataTypes : 784
 USED in MAP-MS-DataTypes : 777

cug-FeatureList.....identifier of CUG-FeatureList
 DEFINED in MAP-MS-DataTypes : 749

CUG-FeatureList.....type reference SEQUENCE OF
 DEFINED in MAP-MS-DataTypes : 776
 USED in MAP-MS-DataTypes : 749

cug-Index.....identifier of CUG-Index
 DEFINED in MAP-MS-DataTypes : 757

CUG-Index.....type reference INTEGER
 DEFINED in MAP-MS-DataTypes : 764
 USED in MAP-MS-DataTypes : 63 757 786

cug-Info.....identifier of [2] CUG-Info
 DEFINED in MAP-MS-DataTypes : 648

CUG-Info.....type reference SEQUENCE
 DEFINED in MAP-MS-DataTypes : 747
 USED in MAP-MS-DataTypes : 648

cug-Interlock.....identifier of CUG-Interlock
 DEFINED in MAP-MS-DataTypes : 758

CUG-Interlock.....type reference OCTET STRING
 DEFINED in MAP-MS-DataTypes : 767
 USED in MAP-MS-DataTypes : 64 758
 USED in MAP-CH-DataTypes : 42 86

cug-Interlock.....identifier of CUG-Interlock
 DEFINED in MAP-CH-DataTypes : 86

cug-OutgoingAccess.....identifier of NULL

DEFINED in MAP-CH-DataTypes : 87

TAG R4.21 Cross Reference Listing for MAP-Protocol 00-01-06 07:32:40 PAGE 18

cug-Reject.....value reference CUG-Reject, CHOICE VALUE
 DEFINED in MAP-Protocol : 376

CUG-Reject.....type reference ERROR
 DEFINED in MAP-Errors : 304
 USED in MAP-Protocol : 140 376
 USED in MAP-CallHandlingOperat : 46 100
 USED in MAP-Errors : 51

cug-RejectCause.....identifier of CUG-RejectCause
 DEFINED in MAP-ER-DataTypes : 117

CUG-RejectCause.....type reference ENUMERATED
 DEFINED in MAP-ER-DataTypes : 121
 USED in MAP-ER-DataTypes : 117

cug-RejectParam.....identifier of CUG-RejectParam
 DEFINED in MAP-Errors : 306

CUG-RejectParam.....type reference SEQUENCE
 DEFINED in MAP-ER-DataTypes : 116
 USED in MAP-Errors : 130 306
 USED in MAP-ER-DataTypes : 16

CUG-Subscription.....type reference SEQUENCE
 DEFINED in MAP-MS-DataTypes : 756
 USED in MAP-MS-DataTypes : 754

cug-SubscriptionList.....identifier of CUG-SubscriptionList
 DEFINED in MAP-MS-DataTypes : 748

CUG-SubscriptionList.....type reference SEQUENCE OF
 DEFINED in MAP-MS-DataTypes : 753
 USED in MAP-MS-DataTypes : 748

currentLocation.....identifier of Named Number, 0
 DEFINED in MAP-LCS-DataTypes : 91

currentOrLastKnownLocation.....identifier of Named Number, 1
 DEFINED in MAP-LCS-DataTypes : 92

currentPassword.....identifier of Password
 DEFINED in MAP-SupplementaryServi : 238

cw.....value reference SS-Code, '01000001'B
 DEFINED in MAP-SS-Code : 75

dataCDA-1200bps.....value reference BearerServiceCode, '00010010'B
 DEFINED in MAP-BS-Code : 53

dataCDA-1200-75bps.....value reference BearerServiceCode, '00010011'B
 DEFINED in MAP-BS-Code : 54

dataCDA-2400bps.....value reference BearerServiceCode, '00010100'B
 DEFINED in MAP-BS-Code : 55

dataCDA-300bps.....value reference BearerServiceCode, '00010001'B
 DEFINED in MAP-BS-Code : 52

dataCDA-4800bps.....value reference BearerServiceCode, '00010101'B
 DEFINED in MAP-BS-Code : 56

dataCDA-9600bps.....value reference BearerServiceCode, '00010110'B
 DEFINED in MAP-BS-Code : 57

dataCDS-1200bps.....value reference BearerServiceCode, '00011010'B
 DEFINED in MAP-BS-Code : 61

dataCDS-2400bps.....value reference BearerServiceCode, '00011100'B
 DEFINED in MAP-BS-Code : 62

dataCDS-4800bps.....value reference BearerServiceCode, '00011101'B
 DEFINED in MAP-BS-Code : 63

dataCDS-9600bps.....value reference BearerServiceCode, '00011110'B
 DEFINED in MAP-BS-Code : 64

dataCodingScheme.....identifier of [0] USSD-DataCodingScheme
 DEFINED in MAP-LCS-DataTypes : 119

dataMissing.....value reference DataMissing, CHOICE VALUE

```

TAG    R4.21    Cross Reference Listing for MAP-Protocol    00-01-06  07:32:40  PAGE  19

    DEFINED in MAP-Protocol    :    326

DataMissing.....type reference ERROR
    DEFINED in MAP-Errors    :    167
    USED in MAP-Protocol    :    116    326
    USED in MAP-MobileServiceOpera :    82    175    187    197    207    231    244    257    274
                                     308    332    346    359    371    382    401    415    429
                                     443    454
    USED in MAP-OperationAndMainte :    24    58    72    83
    USED in MAP-CallHandlingOperat :    31    88    111    128    137    149    163    177    186
    USED in MAP-SupplementaryServi :    34    95    112    129    149    167    182    195    209
                                     224    247    258    275
    USED in MAP-ShortMessageServic :    28    73    101    118    129    143
    USED in MAP-LocationServiceOpe :    24    59    73    91
    USED in MAP-Errors    :    15

dataMissingParam.....identifier of DataMissingParam
    DEFINED in MAP-Errors    :    169

DataMissingParam.....type reference SEQUENCE
    DEFINED in MAP-ER-DataTypes    :    180
    USED in MAP-Errors    :    110    169
    USED in MAP-ER-DataTypes    :    21

dataPDS-2400bps.....value reference BearerServiceCode, '00101100'B
    DEFINED in MAP-BS-Code    :    77

dataPDS-4800bps.....value reference BearerServiceCode, '00101101'B
    DEFINED in MAP-BS-Code    :    78

dataPDS-9600bps.....value reference BearerServiceCode, '00101110'B
    DEFINED in MAP-BS-Code    :    79

deactivate.....identifier of Named Number, 0
    DEFINED in MAP-MS-DataTypes    :    1526

deactivateSS.....value reference DeactivateSS, CHOICE VALUE
    DEFINED in MAP-Protocol    :    244

DeactivateSS.....type reference OPERATION
    DEFINED in MAP-SupplementaryServi :    141
    USED in MAP-Protocol    :    71    244
    USED in MAP-SupplementaryServi :    16

deactivateTraceMode.....value reference DeactivateTraceMode, CHOICE VALUE
    DEFINED in MAP-Protocol    :    221

DeactivateTraceMode.....type reference OPERATION
    DEFINED in MAP-OperationAndMainte :    64
    USED in MAP-Protocol    :    46    221
    USED in MAP-OperationAndMainte :    14

deactivateTraceModeArg.....identifier of DeactivateTraceModeArg
    DEFINED in MAP-OperationAndMainte :    66

DeactivateTraceModeArg.....type reference SEQUENCE
    DEFINED in MAP-OM-DataTypes    :    54
    USED in MAP-OperationAndMainte :    36    66
    USED in MAP-OM-DataTypes    :    16

deactivateTraceModeRes.....identifier of DeactivateTraceModeRes
    DEFINED in MAP-OperationAndMainte :    68

DeactivateTraceModeRes.....type reference SEQUENCE
    DEFINED in MAP-OM-DataTypes    :    60
    USED in MAP-OperationAndMainte :    37    68
    USED in MAP-OM-DataTypes    :    17

defaultCallHandling.....identifier of DefaultCallHandling
    DEFINED in MAP-MS-DataTypes    :    963

defaultCallHandling.....identifier of [1] DefaultCallHandling
    DEFINED in MAP-MS-DataTypes    :    1017

DefaultCallHandling.....type reference ENUMERATED
    DEFINED in MAP-MS-DataTypes    :    1107
    USED in MAP-MS-DataTypes    :    58    963    1017    1220

defaultCallHandling.....identifier of [1] DefaultCallHandling
    DEFINED in MAP-MS-DataTypes    :    1220

```

DefaultGPRS-Handling.....type reference ENUMERATED

TAG R4.21 Cross Reference Listing for MAP-Protocol 00-01-06 07:32:40 PAGE 20

DEFINED in MAP-MS-DataTypes : 498
 USED in MAP-MS-DataTypes : 493

defaultPriority.....identifier of EMLPP-Priority
 DEFINED in MAP-CommonDataTypes : 389

defaultPriority.....identifier of [7] EMLPP-Priority
 DEFINED in MAP-SS-DataTypes : 76

defaultPriority.....identifier of EMLPP-Priority
 DEFINED in MAP-SS-DataTypes : 161

defaultPriority.....identifier of [1] EMLPP-Priority
 DEFINED in MAP-SS-DataTypes : 187

defaultSessionHandling.....identifier of [3] DefaultGPRS-Handling
 DEFINED in MAP-MS-DataTypes : 493

defaultSMS-Handling.....identifier of [3] DefaultSMS-Handling
 DEFINED in MAP-MS-DataTypes : 1147

DefaultSMS-Handling.....type reference ENUMERATED
 DEFINED in MAP-MS-DataTypes : 1160
 USED in MAP-MS-DataTypes : 1147

delaytolerant.....identifier of Named Number, 1
 DEFINED in MAP-LCS-DataTypes : 159

deleteSubscriberData.....value reference DeleteSubscriberData, CHOICE VALUE
 DEFINED in MAP-Protocol : 207

DeleteSubscriberData.....type reference OPERATION
 DEFINED in MAP-MobileServiceOpera : 375
 USED in MAP-Protocol : 25 207
 USED in MAP-MobileServiceOpera : 52

deleteSubscriberDataArg.....identifier of DeleteSubscriberDataArg
 DEFINED in MAP-MobileServiceOpera : 377

DeleteSubscriberDataArg.....type reference SEQUENCE
 DEFINED in MAP-MS-DataTypes : 890
 USED in MAP-MobileServiceOpera : 129 377
 USED in MAP-MS-DataTypes : 45

deleteSubscriberDataRes.....identifier of DeleteSubscriberDataRes
 DEFINED in MAP-MobileServiceOpera : 379

DeleteSubscriberDataRes.....type reference SEQUENCE
 DEFINED in MAP-MS-DataTypes : 928
 USED in MAP-MobileServiceOpera : 130 379
 USED in MAP-MS-DataTypes : 46

deliveryOutcomeIndicator.....identifier of [3] NULL
 DEFINED in MAP-SM-DataTypes : 154

derivable.....identifier of InvokeIdType
 DEFINED in TCAPMessages : 167

destinationNumberCriteria.....identifier of [0] DestinationNumberCriteria
 DEFINED in MAP-MS-DataTypes : 1044

DestinationNumberCriteria.....type reference SEQUENCE
 DEFINED in MAP-MS-DataTypes : 1057
 USED in MAP-MS-DataTypes : 1044

destinationNumberLengthList.....identifier of [2] DestinationNumberLengthList
 DEFINED in MAP-MS-DataTypes : 1060

DestinationNumberLengthList.....type reference SEQUENCE OF
 DEFINED in MAP-MS-DataTypes : 1070
 USED in MAP-MS-DataTypes : 1060

destinationNumberList.....identifier of [1] DestinationNumberList
 DEFINED in MAP-MS-DataTypes : 1059

DestinationNumberList.....type reference SEQUENCE OF
 DEFINED in MAP-MS-DataTypes : 1065
 USED in MAP-MS-DataTypes : 1059

DestTransactionID.....type reference [APPLICATION 9] IMPLICIT
 TransactionID

DEFINED in TCAPMessages	:	98		
USED in TCAPMessages	:	65	70	74

TAG	R4.21	Cross Reference Listing for MAP-Protocol	00-01-06	07:32:40	PAGE	21
diagnosticInfo	identifier of SignalInfo				
	DEFINED in MAP-ER-DataTypes	:	150			
dialledNumber	identifier of ISDN-AddressString				
	DEFINED in MAP-MS-DataTypes	:	960			
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
disallowedByLocalRegulatoryRequirements	identifier of Named Number, 4				
	DEFINED in MAP-ER-DataTypes	:	348			
doublyChargeableECT-Barred	identifier of Named Number, 13				
	DEFINED in MAP-MS-DataTypes	:	629			
dp-AnalysedInfoCriteriaList	identifier of DP-AnalysedInfoCriteriaList				
	DEFINED in MAP-MS-DataTypes	:	949			
DP-AnalysedInfoCriteriaList	type reference SEQUENCE OF				
	DEFINED in MAP-MS-DataTypes	:	954			
	USED in MAP-MS-DataTypes	:	949			
DP-AnalysedInfoCriterium	type reference SEQUENCE				
	DEFINED in MAP-MS-DataTypes	:	959			
	USED in MAP-MS-DataTypes	:	955			
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			
d-CSI	identifier of [9] D-CSI				
	DEFINED in MAP-MS-DataTypes	:	945			
D-CSI	type reference SEQUENCE				
	DEFINED in MAP-MS-DataTypes	:	948			
	USED in MAP-MS-DataTypes	:	53	945		
	USED in MAP-CH-DataTypes	:	44	232	275	
d-csi	identifier of Named Number, 8				
	DEFINED in MAP-MS-DataTypes	:	1453			
d-csi	identifier of [12] D-CSI				
	DEFINED in MAP-CH-DataTypes	:	232			
d-csi	identifier of [5] D-CSI				
	DEFINED in MAP-CH-DataTypes	:	275			
ect	value reference SS-Code, '00110001'B				
	DEFINED in MAP-SS-Code	:	66			
eir	identifier of Named Number, 6				
	DEFINED in MAP-CommonDataTypes	:	311			
emergencyCallOrigination	identifier of Named Number, 0				
	DEFINED in MAP-LCS-DataTypes	:	233			

emergencyCallRelease.....identifier of Named Number, 1

```

TAG   R4.21   Cross Reference Listing for MAP-Protocol           00-01-06  07:32:40  PAGE  22

  DEFINED in MAP-LCS-DataTypes      :    234

emergencyCalls.....value reference TeleserviceCode, '00010010'B
  DEFINED in MAP-TS-Code            :    42

emergencyServices.....identifier of Named Number, 0
  DEFINED in MAP-LCS-DataTypes      :   108

emlpp.....value reference SS-Code, '10100001'B
  DEFINED in MAP-SS-Code            :   154

emlpp-Info.....identifier of [4] EMLPP-Info
  DEFINED in MAP-MS-DataTypes       :   650

EMLPP-Info.....type reference SEQUENCE
  DEFINED in MAP-CommonDataTypes    :   387
  USED in MAP-MS-DataTypes          :   156  650
  USED in MAP-CommonDataTypes       :    47

EMLPP-Priority.....type reference INTEGER
  DEFINED in MAP-CommonDataTypes    :   393
  USED in MAP-CommonDataTypes       :    48  388  389  399  400  401  402  403  404
                                         405
  USED in MAP-SS-DataTypes          :    50  76  161  186  187
  USED in MAP-GR-DataTypes          :    25  56

enabling.....identifier of Named Number, 1
  DEFINED in MAP-MS-DataTypes       :  1090

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       :   239

enterNewPW-Again.....identifier of Named Number, 2
  DEFINED in MAP-SS-DataTypes       :   240

enterPW.....identifier of Named Number, 0
  DEFINED in MAP-SS-DataTypes       :   238

equipmentNotSM-Equipped.....identifier of Named Number, 2
  DEFINED in MAP-ER-DataTypes       :   142

equipmentProtocolError.....identifier of Named Number, 1
  DEFINED in MAP-ER-DataTypes       :   141

equipmentStatus.....identifier of EquipmentStatus
  DEFINED in MAP-MobileServiceOpera :   356

EquipmentStatus.....type reference ENUMERATED
  DEFINED in MAP-MS-DataTypes       :   388
  USED in MAP-MobileServiceOpera   :   126  356
  USED in MAP-MS-DataTypes         :    39

eraseCC-Entry.....value reference EraseCC-Entry, CHOICE VALUE
  DEFINED in MAP-Protocol            :   253

EraseCC-Entry.....type reference OPERATION
  DEFINED in MAP-SupplementaryServi :   268
  USED in MAP-Protocol              :    80  253
  USED in MAP-SupplementaryServi    :    25

eraseCC-EntryArg.....identifier of EraseCC-EntryArg
  DEFINED in MAP-SupplementaryServi :   270

EraseCC-EntryArg.....type reference SEQUENCE
  DEFINED in MAP-SS-DataTypes       :   304
  USED in MAP-SupplementaryServi    :    72  270
  USED in MAP-SS-DataTypes         :    39

eraseCC-EntryRes.....identifier of EraseCC-EntryRes
  DEFINED in MAP-SupplementaryServi :   272

EraseCC-EntryRes.....type reference SEQUENCE
  DEFINED in MAP-SS-DataTypes       :   309
  USED in MAP-SupplementaryServi    :    73  272
    
```

USED in MAP-SS-DataTypes : 40

TAG R4.21 Cross Reference Listing for MAP-Protocol 00-01-06 07:32:40 PAGE 23

eraseSS.....value reference EraseSS, CHOICE VALUE
 DEFINED in MAP-Protocol : 242

EraseSS.....type reference OPERATION
 DEFINED in MAP-SupplementaryServi : 104
 USED in MAP-Protocol : 69 242
 USED in MAP-SupplementaryServi : 14

errorCode.....identifier of ERROR
 DEFINED in TCAPMessages : 158
 USED in TCAPMessages : 159

ets-300102-1.....identifier of Named Number, 4
 DEFINED in MAP-CommonDataTypes : 201

ets-300356.....identifier of Named Number, 1
 DEFINED in MAP-CommonDataTypes : 212

eventMet.....identifier of [0] MM-Code
 DEFINED in MAP-MS-DataTypes : 1556

eventReportData.....identifier of [1] EventReportData
 DEFINED in MAP-CH-DataTypes : 344

EventReportData.....type reference SEQUENCE
 DEFINED in MAP-CH-DataTypes : 349
 USED in MAP-CH-DataTypes : 344

extendedRoutingInfo.....identifier of ExtendedRoutingInfo
 DEFINED in MAP-CH-DataTypes : 148

ExtendedRoutingInfo.....type reference CHOICE
 DEFINED in MAP-CH-DataTypes : 258
 USED in MAP-CH-DataTypes : 148

extensibleCallBarredParam.....identifier of ExtensibleCallBarredParam
 DEFINED in MAP-ER-DataTypes : 102

ExtensibleCallBarredParam.....type reference SEQUENCE
 DEFINED in MAP-ER-DataTypes : 110
 USED in MAP-ER-DataTypes : 102

extensibleSystemFailureParam.....identifier of ExtensibleSystemFailureParam
 DEFINED in MAP-ER-DataTypes : 171

ExtensibleSystemFailureParam.....type reference SEQUENCE
 DEFINED in MAP-ER-DataTypes : 175
 USED in MAP-ER-DataTypes : 171

extensionContainer.....identifier of ExtensionContainer
 DEFINED in MAP-MS-DataTypes : 190

extensionContainer.....identifier of ExtensionContainer
 DEFINED in MAP-MS-DataTypes : 196

extensionContainer.....identifier of ExtensionContainer
 DEFINED in MAP-MS-DataTypes : 220

extensionContainer.....identifier of ExtensionContainer
 DEFINED in MAP-MS-DataTypes : 226

extensionContainer.....identifier of ExtensionContainer
 DEFINED in MAP-MS-DataTypes : 238

extensionContainer.....identifier of ExtensionContainer
 DEFINED in MAP-MS-DataTypes : 245

extensionContainer.....identifier of ExtensionContainer
 DEFINED in MAP-MS-DataTypes : 251

extensionContainer.....identifier of ExtensionContainer
 DEFINED in MAP-MS-DataTypes : 260

extensionContainer.....identifier of [2] ExtensionContainer
 DEFINED in MAP-MS-DataTypes : 269

extensionContainer.....identifier of ExtensionContainer
 DEFINED in MAP-MS-DataTypes : 318

extensionContainer.....identifier of [1] ExtensionContainer

DEFINED in MAP-MS-DataTypes : 324

TAG	R4.21	Cross Reference Listing for MAP-Protocol	00-01-06	07:32:40	PAGE	24
extensionContainer	identifier of ExtensionContainer				
DEFINED in MAP-MS-DataTypes	:	335				
extensionContainer	identifier of [2] ExtensionContainer				
DEFINED in MAP-MS-DataTypes	:	369				
extensionContainer	identifier of ExtensionContainer				
DEFINED in MAP-MS-DataTypes	:	382				
extensionContainer	identifier of [14] ExtensionContainer				
DEFINED in MAP-MS-DataTypes	:	399				
extensionContainer	identifier of [21] ExtensionContainer				
DEFINED in MAP-MS-DataTypes	:	451				
extensionContainer	identifier of [2] ExtensionContainer				
DEFINED in MAP-MS-DataTypes	:	464				
extensionContainer	identifier of [2] ExtensionContainer				
DEFINED in MAP-MS-DataTypes	:	471				
extensionContainer	identifier of [2] ExtensionContainer				
DEFINED in MAP-MS-DataTypes	:	477				
extensionContainer	identifier of [4] ExtensionContainer				
DEFINED in MAP-MS-DataTypes	:	494				
extensionContainer	identifier of [4] ExtensionContainer				
DEFINED in MAP-MS-DataTypes	:	553				
extensionContainer	identifier of [3] ExtensionContainer				
DEFINED in MAP-MS-DataTypes	:	563				
extensionContainer	identifier of ExtensionContainer				
DEFINED in MAP-MS-DataTypes	:	612				
extensionContainer	identifier of [0] ExtensionContainer				
DEFINED in MAP-MS-DataTypes	:	656				
extensionContainer	identifier of [9] ExtensionContainer				
DEFINED in MAP-MS-DataTypes	:	672				
extensionContainer	identifier of ExtensionContainer				
DEFINED in MAP-MS-DataTypes	:	735				
extensionContainer	identifier of ExtensionContainer				
DEFINED in MAP-MS-DataTypes	:	744				
extensionContainer	identifier of [0] ExtensionContainer				
DEFINED in MAP-MS-DataTypes	:	750				
extensionContainer	identifier of [0] ExtensionContainer				
DEFINED in MAP-MS-DataTypes	:	761				
extensionContainer	identifier of ExtensionContainer				
DEFINED in MAP-MS-DataTypes	:	788				
extensionContainer	identifier of [5] ExtensionContainer				
DEFINED in MAP-MS-DataTypes	:	808				
extensionContainer	identifier of [3] ExtensionContainer				
DEFINED in MAP-MS-DataTypes	:	825				
extensionContainer	identifier of [2] ExtensionContainer				
DEFINED in MAP-MS-DataTypes	:	843				
extensionContainer	identifier of [0] ExtensionContainer				
DEFINED in MAP-MS-DataTypes	:	862				
extensionContainer	identifier of [7] ExtensionContainer				
DEFINED in MAP-MS-DataTypes	:	881				
extensionContainer	identifier of [6] ExtensionContainer				
DEFINED in MAP-MS-DataTypes	:	901				
extensionContainer	identifier of ExtensionContainer				
DEFINED in MAP-MS-DataTypes	:	931				
extensionContainer	identifier of [1] ExtensionContainer				
DEFINED in MAP-MS-DataTypes	:	936				

TAG R4.21 Cross Reference Listing for MAP-Protocol 00-01-06 07:32:40 PAGE 25

extensionContainer.....identifier of ExtensionContainer
 DEFINED in MAP-MS-DataTypes : 951

extensionContainer.....identifier of ExtensionContainer
 DEFINED in MAP-MS-DataTypes : 964

extensionContainer.....identifier of ExtensionContainer
 DEFINED in MAP-MS-DataTypes : 969

extensionContainer.....identifier of [0] ExtensionContainer
 DEFINED in MAP-MS-DataTypes : 975

extensionContainer.....identifier of ExtensionContainer
 DEFINED in MAP-MS-DataTypes : 995

extensionContainer.....identifier of [2] ExtensionContainer
 DEFINED in MAP-MS-DataTypes : 1018

extensionContainer.....identifier of [4] ExtensionContainer
 DEFINED in MAP-MS-DataTypes : 1049

extensionContainer.....identifier of [2] ExtensionContainer
 DEFINED in MAP-MS-DataTypes : 1131

extensionContainer.....identifier of [4] ExtensionContainer
 DEFINED in MAP-MS-DataTypes : 1148

extensionContainer.....identifier of [1] ExtensionContainer
 DEFINED in MAP-MS-DataTypes : 1172

extensionContainer.....identifier of ExtensionContainer
 DEFINED in MAP-MS-DataTypes : 1199

extensionContainer.....identifier of [2] ExtensionContainer
 DEFINED in MAP-MS-DataTypes : 1221

extensionContainer.....identifier of [3] ExtensionContainer
 DEFINED in MAP-MS-DataTypes : 1241

extensionContainer.....identifier of [3] ExtensionContainer
 DEFINED in MAP-MS-DataTypes : 1248

extensionContainer.....identifier of [3] ExtensionContainer
 DEFINED in MAP-MS-DataTypes : 1257

extensionContainer.....identifier of [1] ExtensionContainer
 DEFINED in MAP-MS-DataTypes : 1262

extensionContainer.....identifier of [3] ExtensionContainer
 DEFINED in MAP-MS-DataTypes : 1271

extensionContainer.....identifier of [0] ExtensionContainer
 DEFINED in MAP-MS-DataTypes : 1275

extensionContainer.....identifier of ExtensionContainer
 DEFINED in MAP-MS-DataTypes : 1289

extensionContainer.....identifier of ExtensionContainer
 DEFINED in MAP-MS-DataTypes : 1296

extensionContainer.....identifier of ExtensionContainer
 DEFINED in MAP-MS-DataTypes : 1313

extensionContainer.....identifier of ExtensionContainer
 DEFINED in MAP-MS-DataTypes : 1319

extensionContainer.....identifier of [3] ExtensionContainer
 DEFINED in MAP-MS-DataTypes : 1332

extensionContainer.....identifier of ExtensionContainer
 DEFINED in MAP-MS-DataTypes : 1337

extensionContainer.....identifier of [2] ExtensionContainer
 DEFINED in MAP-MS-DataTypes : 1343

extensionContainer.....identifier of [2] ExtensionContainer
 DEFINED in MAP-MS-DataTypes : 1349

extensionContainer.....identifier of [4] ExtensionContainer
 DEFINED in MAP-MS-DataTypes : 1358

extensionContainer.....identifier of [2] ExtensionContainer

TAG R4.21 Cross Reference Listing for MAP-Protocol 00-01-06 07:32:40 PAGE 26

DEFINED in MAP-MS-DataTypes : 1407

extensionContainer.....identifier of ExtensionContainer
DEFINED in MAP-MS-DataTypes : 1412

extensionContainer.....identifier of [3] ExtensionContainer
DEFINED in MAP-MS-DataTypes : 1422

extensionContainer.....identifier of [7] ExtensionContainer
DEFINED in MAP-MS-DataTypes : 1432

extensionContainer.....identifier of [6] ExtensionContainer
DEFINED in MAP-MS-DataTypes : 1441

extensionContainer.....identifier of [0] ExtensionContainer
DEFINED in MAP-MS-DataTypes : 1458

extensionContainer.....identifier of ExtensionContainer
DEFINED in MAP-MS-DataTypes : 1466

extensionContainer.....identifier of ExtensionContainer
DEFINED in MAP-MS-DataTypes : 1474

extensionContainer.....identifier of [12] ExtensionContainer
DEFINED in MAP-MS-DataTypes : 1490

extensionContainer.....identifier of [4] ExtensionContainer
DEFINED in MAP-MS-DataTypes : 1498

extensionContainer.....identifier of [2] ExtensionContainer
DEFINED in MAP-MS-DataTypes : 1504

extensionContainer.....identifier of [7] ExtensionContainer
DEFINED in MAP-MS-DataTypes : 1515

extensionContainer.....identifier of [3] ExtensionContainer
DEFINED in MAP-MS-DataTypes : 1522

extensionContainer.....identifier of ExtensionContainer
DEFINED in MAP-MS-DataTypes : 1535

extensionContainer.....identifier of ExtensionContainer
DEFINED in MAP-MS-DataTypes : 1539

extensionContainer.....identifier of [6] ExtensionContainer
DEFINED in MAP-MS-DataTypes : 1562

extensionContainer.....identifier of ExtensionContainer
DEFINED in MAP-MS-DataTypes : 1566

extensionContainer.....identifier of [3] ExtensionContainer
DEFINED in MAP-MS-DataTypes : 1578

extensionContainer.....identifier of [5] ExtensionContainer
DEFINED in MAP-MS-DataTypes : 1587

extensionContainer.....identifier of ExtensionContainer
DEFINED in MAP-CommonDataTypes : 182

extensionContainer.....identifier of ExtensionContainer
DEFINED in MAP-CommonDataTypes : 208

extensionContainer.....identifier of [1] ExtensionContainer
DEFINED in MAP-CommonDataTypes : 316

extensionContainer.....identifier of [1] ExtensionContainer
DEFINED in MAP-CommonDataTypes : 333

extensionContainer.....identifier of ExtensionContainer
DEFINED in MAP-CommonDataTypes : 390

extensionContainer.....identifier of [4] ExtensionContainer
DEFINED in MAP-OM-DataTypes : 41

extensionContainer.....identifier of [0] ExtensionContainer
DEFINED in MAP-OM-DataTypes : 51

extensionContainer.....identifier of [2] ExtensionContainer
DEFINED in MAP-OM-DataTypes : 57

extensionContainer.....identifier of [0] ExtensionContainer

DEFINED in MAP-OM-DataTypes : 61

TAG	R4.21	Cross Reference Listing for MAP-Protocol	00-01-06	07:32:40	PAGE	27
extensionContainer	identifier of ExtensionContainer				
DEFINED in MAP-CH-DataTypes	:	88				
extensionContainer	identifier of [13] ExtensionContainer				
DEFINED in MAP-CH-DataTypes	:	107				
extensionContainer	identifier of [0] ExtensionContainer				
DEFINED in MAP-CH-DataTypes	:	156				
extensionContainer	identifier of [2] ExtensionContainer				
DEFINED in MAP-CH-DataTypes	:	178				
extensionContainer	identifier of [7] ExtensionContainer				
DEFINED in MAP-CH-DataTypes	:	192				
extensionContainer	identifier of [11] ExtensionContainer				
DEFINED in MAP-CH-DataTypes	:	206				
extensionContainer	identifier of ExtensionContainer				
DEFINED in MAP-CH-DataTypes	:	217				
extensionContainer	identifier of [7] ExtensionContainer				
DEFINED in MAP-CH-DataTypes	:	227				
extensionContainer	identifier of [3] ExtensionContainer				
DEFINED in MAP-CH-DataTypes	:	239				
extensionContainer	identifier of ExtensionContainer				
DEFINED in MAP-CH-DataTypes	:	249				
extensionContainer	identifier of ExtensionContainer				
DEFINED in MAP-CH-DataTypes	:	255				
extensionContainer	identifier of [1] ExtensionContainer				
DEFINED in MAP-CH-DataTypes	:	265				
extensionContainer	identifier of [2] ExtensionContainer				
DEFINED in MAP-CH-DataTypes	:	271				
extensionContainer	identifier of [7] ExtensionContainer				
DEFINED in MAP-CH-DataTypes	:	285				
extensionContainer	identifier of [1] ExtensionContainer				
DEFINED in MAP-CH-DataTypes	:	298				
extensionContainer	identifier of [2] ExtensionContainer				
DEFINED in MAP-CH-DataTypes	:	304				
extensionContainer	identifier of [1] ExtensionContainer				
DEFINED in MAP-CH-DataTypes	:	309				
extensionContainer	identifier of [3] ExtensionContainer				
DEFINED in MAP-CH-DataTypes	:	316				
extensionContainer	identifier of [1] ExtensionContainer				
DEFINED in MAP-CH-DataTypes	:	329				
extensionContainer	identifier of [3] ExtensionContainer				
DEFINED in MAP-CH-DataTypes	:	346				
extensionContainer	identifier of [1] ExtensionContainer				
DEFINED in MAP-CH-DataTypes	:	351				
extensionContainer	identifier of [2] ExtensionContainer				
DEFINED in MAP-CH-DataTypes	:	357				
extensionContainer	identifier of [0] ExtensionContainer				
DEFINED in MAP-CH-DataTypes	:	379				
extensionContainer	identifier of [6] ExtensionContainer				
DEFINED in MAP-CH-DataTypes	:	389				
extensionContainer	identifier of [1] ExtensionContainer				
DEFINED in MAP-CH-DataTypes	:	394				
extensionContainer	identifier of [1] ExtensionContainer				
DEFINED in MAP-CH-DataTypes	:	415				
extensionContainer	identifier of [3] ExtensionContainer				
DEFINED in MAP-CH-DataTypes	:	422				

TAG	R4.21	Cross Reference Listing for MAP-Protocol	00-01-06	07:32:40	PAGE 28
extensionContainer	identifier of [1] ExtensionContainer			
DEFINED in MAP-CH-DataTypes	:	427			
extensionContainer	identifier of ExtensionContainer			
DEFINED in MAP-CH-DataTypes	:	431			
extensionContainer	identifier of [4] ExtensionContainer			
DEFINED in MAP-SS-DataTypes	:	268			
extensionContainer	identifier of ExtensionContainer			
DEFINED in MAP-SS-DataTypes	:	272			
extensionContainer	identifier of [6] ExtensionContainer			
DEFINED in MAP-SM-DataTypes	:	56			
extensionContainer	identifier of [4] ExtensionContainer			
DEFINED in MAP-SM-DataTypes	:	82			
extensionContainer	identifier of ExtensionContainer			
DEFINED in MAP-SM-DataTypes	:	88			
extensionContainer	identifier of ExtensionContainer			
DEFINED in MAP-SM-DataTypes	:	110			
extensionContainer	identifier of ExtensionContainer			
DEFINED in MAP-SM-DataTypes	:	116			
extensionContainer	identifier of ExtensionContainer			
DEFINED in MAP-SM-DataTypes	:	124			
extensionContainer	identifier of ExtensionContainer			
DEFINED in MAP-SM-DataTypes	:	129			
extensionContainer	identifier of [1] ExtensionContainer			
DEFINED in MAP-SM-DataTypes	:	149			
extensionContainer	identifier of ExtensionContainer			
DEFINED in MAP-SM-DataTypes	:	170			
extensionContainer	identifier of ExtensionContainer			
DEFINED in MAP-SM-DataTypes	:	182			
extensionContainer	identifier of ExtensionContainer			
DEFINED in MAP-SM-DataTypes	:	199			
extensionContainer	identifier of ExtensionContainer			
DEFINED in MAP-SM-DataTypes	:	203			
extensionContainer	identifier of [4] ExtensionContainer			
DEFINED in MAP-GR-DataTypes	:	58			
extensionContainer	identifier of ExtensionContainer			
DEFINED in MAP-GR-DataTypes	:	63			
extensionContainer	identifier of ExtensionContainer			
DEFINED in MAP-GR-DataTypes	:	68			
extensionContainer	identifier of ExtensionContainer			
DEFINED in MAP-GR-DataTypes	:	72			
extensionContainer	identifier of ExtensionContainer			
DEFINED in MAP-GR-DataTypes	:	82			
extensionContainer	identifier of ExtensionContainer			
DEFINED in MAP-GR-DataTypes	:	89			
extensionContainer	identifier of [2] ExtensionContainer			
DEFINED in MAP-LCS-DataTypes	:	55			
extensionContainer	identifier of [2] ExtensionContainer			
DEFINED in MAP-LCS-DataTypes	:	61			
extensionContainer	identifier of [1] ExtensionContainer			
DEFINED in MAP-LCS-DataTypes	:	67			
extensionContainer	identifier of [8] ExtensionContainer			
DEFINED in MAP-LCS-DataTypes	:	81			
extensionContainer	identifier of [4] ExtensionContainer			
DEFINED in MAP-LCS-DataTypes	:	141			

extensionContainer.....identifier of [1] ExtensionContainer

TAG	R4.21	Cross Reference Listing for MAP-Protocol	00-01-06	07:32:40	PAGE	29
	DEFINED in MAP-LCS-DataTypes	:	167			
extensionContainer.....	identifier of [7] ExtensionContainer					
	DEFINED in MAP-LCS-DataTypes	:	227			
extensionContainer.....	identifier of ExtensionContainer					
	DEFINED in MAP-LCS-DataTypes	:	242			
extensionContainer.....	identifier of ExtensionContainer					
	DEFINED in MAP-ER-DataTypes	:	92			
extensionContainer.....	identifier of ExtensionContainer					
	DEFINED in MAP-ER-DataTypes	:	112			
extensionContainer.....	identifier of ExtensionContainer					
	DEFINED in MAP-ER-DataTypes	:	118			
extensionContainer.....	identifier of ExtensionContainer					
	DEFINED in MAP-ER-DataTypes	:	151			
extensionContainer.....	identifier of ExtensionContainer					
	DEFINED in MAP-ER-DataTypes	:	158			
extensionContainer.....	identifier of ExtensionContainer					
	DEFINED in MAP-ER-DataTypes	:	177			
extensionContainer.....	identifier of ExtensionContainer					
	DEFINED in MAP-ER-DataTypes	:	181			
extensionContainer.....	identifier of ExtensionContainer					
	DEFINED in MAP-ER-DataTypes	:	185			
extensionContainer.....	identifier of ExtensionContainer					
	DEFINED in MAP-ER-DataTypes	:	189			
extensionContainer.....	identifier of ExtensionContainer					
	DEFINED in MAP-ER-DataTypes	:	193			
extensionContainer.....	identifier of ExtensionContainer					
	DEFINED in MAP-ER-DataTypes	:	197			
extensionContainer.....	identifier of ExtensionContainer					
	DEFINED in MAP-ER-DataTypes	:	210			
extensionContainer.....	identifier of ExtensionContainer					
	DEFINED in MAP-ER-DataTypes	:	214			
extensionContainer.....	identifier of ExtensionContainer					
	DEFINED in MAP-ER-DataTypes	:	218			
extensionContainer.....	identifier of ExtensionContainer					
	DEFINED in MAP-ER-DataTypes	:	222			
extensionContainer.....	identifier of ExtensionContainer					
	DEFINED in MAP-ER-DataTypes	:	226			
extensionContainer.....	identifier of ExtensionContainer					
	DEFINED in MAP-ER-DataTypes	:	230			
extensionContainer.....	identifier of ExtensionContainer					
	DEFINED in MAP-ER-DataTypes	:	234			
extensionContainer.....	identifier of ExtensionContainer					
	DEFINED in MAP-ER-DataTypes	:	238			
extensionContainer.....	identifier of ExtensionContainer					
	DEFINED in MAP-ER-DataTypes	:	242			
extensionContainer.....	identifier of ExtensionContainer					
	DEFINED in MAP-ER-DataTypes	:	260			
extensionContainer.....	identifier of ExtensionContainer					
	DEFINED in MAP-ER-DataTypes	:	266			
extensionContainer.....	identifier of ExtensionContainer					
	DEFINED in MAP-ER-DataTypes	:	270			
extensionContainer.....	identifier of ExtensionContainer					
	DEFINED in MAP-ER-DataTypes	:	274			
extensionContainer.....	identifier of ExtensionContainer					

DEFINED in MAP-ER-DataTypes : 278

TAG R4.21 Cross Reference Listing for MAP-Protocol 00-01-06 07:32:40 PAGE 30

```

extensionContainer.....identifier of ExtensionContainer
  DEFINED in MAP-ER-DataTypes      :    282

extensionContainer.....identifier of ExtensionContainer
  DEFINED in MAP-ER-DataTypes      :    286

extensionContainer.....identifier of ExtensionContainer
  DEFINED in MAP-ER-DataTypes      :    290

extensionContainer.....identifier of ExtensionContainer
  DEFINED in MAP-ER-DataTypes      :    294

extensionContainer.....identifier of ExtensionContainer
  DEFINED in MAP-ER-DataTypes      :    298

extensionContainer.....identifier of ExtensionContainer
  DEFINED in MAP-ER-DataTypes      :    302

extensionContainer.....identifier of ExtensionContainer
  DEFINED in MAP-ER-DataTypes      :    306

extensionContainer.....identifier of ExtensionContainer
  DEFINED in MAP-ER-DataTypes      :    313

extensionContainer.....identifier of ExtensionContainer
  DEFINED in MAP-ER-DataTypes      :    317

extensionContainer.....identifier of ExtensionContainer
  DEFINED in MAP-ER-DataTypes      :    321

extensionContainer.....identifier of ExtensionContainer
  DEFINED in MAP-ER-DataTypes      :    325

extensionContainer.....identifier of ExtensionContainer
  DEFINED in MAP-ER-DataTypes      :    335

extensionContainer.....identifier of [1] ExtensionContainer
  DEFINED in MAP-ER-DataTypes      :    340

extensionContainer.....identifier of [1] ExtensionContainer
  DEFINED in MAP-ER-DataTypes      :    355

extensionContainer.....identifier of ExtensionContainer
  DEFINED in MAP-ER-DataTypes      :    371

extensionContainer.....identifier of ExtensionContainer
  DEFINED in MAP-ER-DataTypes      :    375

ExtensionContainer.....type reference SEQUENCE
  DEFINED in MAP-ExtensionDataTypes :    32
  USED in MAP-MS-DataTypes          :    168    190    196    220    226    238    245    251    260
                                     269    318    324    335    369    382    399    451    464
                                     471    477    494    553    563    612    656    672    735
                                     744    750    761    788    808    825    843    862    881
                                     901    931    936    951    964    969    975    995   1018
                                     1049  1131  1148  1172  1199  1221  1241  1248  1257
                                     1262  1271  1275  1289  1296  1313  1319  1332  1337
                                     1343  1349  1358  1407  1412  1422  1432  1441  1458
                                     1466  1474  1490  1498  1504  1515  1522  1535  1539
                                     1562  1566  1578  1587
  USED in MAP-CommonDataTypes       :    69    182    208    316    333    390
  USED in MAP-OM-DataTypes          :    27    41    51    57    61
  USED in MAP-CH-DataTypes          :    78    88    107   156   178   192   206   217   227
                                     239   249   255   265   271   285   298   304   309
                                     316   329   346   351   357   379   389   394   415
                                     422   427   431
  USED in MAP-SS-DataTypes          :    57   268   272
  USED in MAP-SM-DataTypes          :    45    56    82    88   110   116   124   129   149
                                     170   182   199   203
  USED in MAP-GR-DataTypes          :    42    58    63    68    72    82    89
  USED in MAP-LCS-DataTypes         :    39    55    61    67    81   141   167   227   242
  USED in MAP-ER-DataTypes         :    84    92   112   118   151   158   177   181   185
                                     189   193   197   210   214   218   222   226   230
                                     234   238   242   260   266   270   274   278   282
                                     286   290   294   298   302   306   313   317   321
                                     325   335   340   355   371   375
  USED in MAP-ExtensionDataTypes    :    16

ExtensionSet.....information object set reference MAP-EXTENSION, Information Object

```

Set

DEFINED in MAP-ExtensionDataTypes :	48	
USED in MAP-ExtensionDataTypes :	42	44

TAG R4.21 Cross Reference Listing for MAP-Protocol 00-01-06 07:32:40 PAGE 31

externalAddress.....identifier of [0] AddressString
 DEFINED in MAP-CommonDataTypes : 332

ExternalClient.....type reference SEQUENCE
 DEFINED in MAP-MS-DataTypes : 839
 USED in MAP-MS-DataTypes : 830

externalClientList.....identifier of [1] ExternalClientList
 DEFINED in MAP-MS-DataTypes : 821

ExternalClientList.....type reference SEQUENCE OF
 DEFINED in MAP-MS-DataTypes : 829
 USED in MAP-MS-DataTypes : 821

ExternalSignalInfo.....type reference SEQUENCE
 DEFINED in MAP-CommonDataTypes : 177
 USED in MAP-MobileServiceOpera : 158 314 319 323 329
 USED in MAP-MS-DataTypes : 146 343 348 354
 USED in MAP-CommonDataTypes : 20
 USED in MAP-CH-DataTypes : 65 104 200 201 278 279 282 283 284
 302 303 308 384
 USED in MAP-SS-DataTypes : 51 290 291

extId.....identifier of InformationObjectClassFieldType
 DEFINED in MAP-ExtensionDataTypes : 41

extType.....identifier of InformationObjectClassFieldType
 DEFINED in MAP-ExtensionDataTypes : 43

Ext-BasicServiceCode.....type reference CHOICE
 DEFINED in MAP-CommonDataTypes : 383
 USED in MAP-MS-DataTypes : 154 663 742 780 785 924 1074 1509
 USED in MAP-CommonDataTypes : 46
 USED in MAP-CH-DataTypes : 69 103 153 222

Ext-BasicServiceGroupList.....type reference SEQUENCE OF
 DEFINED in MAP-MS-DataTypes : 779
 USED in MAP-MS-DataTypes : 760 807

ext-BearerService.....identifier of [2] Ext-BearerServiceCode
 DEFINED in MAP-CommonDataTypes : 384

Ext-BearerServiceCode.....type reference OCTET STRING
 DEFINED in MAP-BS-Code : 25
 USED in MAP-MS-DataTypes : 132 600
 USED in MAP-CommonDataTypes : 64 384

Ext-CallBarFeatureList.....type reference SEQUENCE OF
 DEFINED in MAP-MS-DataTypes : 738
 USED in MAP-MS-DataTypes : 734 1462 1583

Ext-CallBarInfo.....type reference SEQUENCE
 DEFINED in MAP-MS-DataTypes : 732
 USED in MAP-MS-DataTypes : 647

Ext-CallBarringFeature.....type reference SEQUENCE
 DEFINED in MAP-MS-DataTypes : 741
 USED in MAP-MS-DataTypes : 739

Ext-CallBarringInfoFor-CSE.....type reference SEQUENCE
 DEFINED in MAP-MS-DataTypes : 1581
 USED in MAP-MS-DataTypes : 1571

Ext-ExternalSignalInfo.....type reference SEQUENCE
 DEFINED in MAP-CommonDataTypes : 203
 USED in MAP-CommonDataTypes : 21
 USED in MAP-CH-DataTypes : 66 112 211

Ext-ForwardingInfoFor-CSE.....type reference SEQUENCE
 DEFINED in MAP-MS-DataTypes : 1574
 USED in MAP-MS-DataTypes : 1570

Ext-ForwFeature.....type reference SEQUENCE
 DEFINED in MAP-MS-DataTypes : 662
 USED in MAP-MS-DataTypes : 660

Ext-ForwFeatureList.....type reference SEQUENCE OF
 DEFINED in MAP-MS-DataTypes : 659
 USED in MAP-MS-DataTypes : 655 1456 1576

Ext-ForwInfo.....type reference SEQUENCE

TAG R4.21 Cross Reference Listing for MAP-Protocol 00-01-06 07:32:40 PAGE 32

DEFINED in MAP-MS-DataTypes : 653
 USED in MAP-MS-DataTypes : 646

Ext-ForwOptions.....type reference OCTET STRING
 DEFINED in MAP-MS-DataTypes : 696
 USED in MAP-MS-DataTypes : 670

Ext-GeographicalInformation.....type reference OCTET STRING
 DEFINED in MAP-LCS-DataTypes : 170
 USED in MAP-LCS-DataTypes : 22 165 225

Ext-NoRepCondTime.....type reference INTEGER
 DEFINED in MAP-MS-DataTypes : 725
 USED in MAP-MS-DataTypes : 671 1513

ext-ProtocolId.....identifier of Ext-ProtocolId
 DEFINED in MAP-CommonDataTypes : 204

Ext-ProtocolId.....type reference ENUMERATED
 DEFINED in MAP-CommonDataTypes : 211
 USED in MAP-CommonDataTypes : 204

ext-QoS-Subscribed.....identifier of [0] Ext-QoS-Subscribed
 DEFINED in MAP-MS-DataTypes : 453

Ext-QoS-Subscribed.....type reference OCTET STRING
 DEFINED in MAP-MS-DataTypes : 536
 USED in MAP-MS-DataTypes : 453

Ext-SS-Data.....type reference SEQUENCE
 DEFINED in MAP-MS-DataTypes : 803
 USED in MAP-MS-DataTypes : 649

Ext-SS-Info.....type reference CHOICE
 DEFINED in MAP-MS-DataTypes : 645
 USED in MAP-MS-DataTypes : 643

Ext-SS-InfoFor-CSE.....type reference CHOICE
 DEFINED in MAP-MS-DataTypes : 1569
 USED in MAP-MS-DataTypes : 1502

Ext-SS-InfoList.....type reference SEQUENCE OF
 DEFINED in MAP-MS-DataTypes : 642
 USED in MAP-MS-DataTypes : 583

Ext-SS-Status.....type reference OCTET STRING
 DEFINED in MAP-MS-DataTypes : 675
 USED in MAP-MS-DataTypes : 664 743 805 818 861 1510

ext-Teleservice.....identifier of [3] Ext-TeleserviceCode
 DEFINED in MAP-CommonDataTypes : 385

Ext-TeleserviceCode.....type reference OCTET STRING
 DEFINED in MAP-TS-Code : 20
 USED in MAP-MS-DataTypes : 137 605
 USED in MAP-CommonDataTypes : 58 385
 USED in MAP-GR-DataTypes : 31 50

facilityNotSupParam.....identifier of FacilityNotSupParam
 DEFINED in MAP-Errors : 181

FacilityNotSupParam.....type reference SEQUENCE
 DEFINED in MAP-ER-DataTypes : 188
 USED in MAP-Errors : 112 181
 USED in MAP-ER-DataTypes : 23

facilityNotSupported.....value reference FacilityNotSupported, CHOICE VALUE
 DEFINED in MAP-Protocol : 328

FacilityNotSupported.....type reference ERROR
 DEFINED in MAP-Errors : 179
 USED in MAP-Protocol : 118 328
 USED in MAP-OperationAndMainte : 26 60 74
 USED in MAP-CallHandlingOperat : 33 90 113 165 203 216
 USED in MAP-SupplementaryServi : 55 266
 USED in MAP-ShortMessageServic : 30 75 90 103 145
 USED in MAP-LocationServiceOpe : 26 61 75
 USED in MAP-Errors : 17

facsimileGroup3AndAlterSpeech.....value reference TeleserviceCode, '01100001'B
 DEFINED in MAP-TS-Code : 49

TAG R4.21 Cross Reference Listing for MAP-Protocol 00-01-06 07:32:40 PAGE 33

facsimileGroup4.....value reference TeleserviceCode, '01100011'B
 DEFINED in MAP-TS-Code : 51

failure.....identifier of Named Number, 1
 DEFINED in MAP-CH-DataTypes : 370

failureReport.....value reference FailureReport, CHOICE VALUE
 DEFINED in MAP-Protocol : 306

FailureReport.....type reference OPERATION
 DEFINED in MAP-MobileServiceOpera : 421
 USED in MAP-Protocol : 34 306
 USED in MAP-MobileServiceOpera : 63

failureReportArg.....identifier of FailureReportArg
 DEFINED in MAP-MobileServiceOpera : 423

FailureReportArg.....type reference SEQUENCE
 DEFINED in MAP-MS-DataTypes : 1253
 USED in MAP-MobileServiceOpera : 146 423
 USED in MAP-MS-DataTypes : 102

failureReportRes.....identifier of FailureReportRes
 DEFINED in MAP-MobileServiceOpera : 425

FailureReportRes.....type reference SEQUENCE
 DEFINED in MAP-MS-DataTypes : 1260
 USED in MAP-MobileServiceOpera : 147 425
 USED in MAP-MS-DataTypes : 103

foreignNumberPortedToForeignNetwork.....identifier of Named Number, 2
 DEFINED in MAP-CH-DataTypes : 169

forwardAccessSignalling.....value reference ForwardAccessSignalling, CHOICE VALUE
 DEFINED in MAP-Protocol : 189

ForwardAccessSignalling.....type reference OPERATION
 DEFINED in MAP-MobileServiceOpera : 321
 USED in MAP-Protocol : 20 189
 USED in MAP-MobileServiceOpera : 41

forwardCheckSS-Indication.....value reference ForwardCheckSS-Indication, CHOICE
 VALUE
 DEFINED in MAP-Protocol : 213

ForwardCheckSS-Indication.....type reference OPERATION
 DEFINED in MAP-MobileServiceOpera : 392
 USED in MAP-Protocol : 27 213
 USED in MAP-MobileServiceOpera : 56

forwarded.....identifier of Named Number, 0
 DEFINED in MAP-MS-DataTypes : 1085

forwardedToNumber.....identifier of [5] ISDN-AddressString
 DEFINED in MAP-MS-DataTypes : 665

forwardedToNumber.....identifier of [3] AddressString
 DEFINED in MAP-MS-DataTypes : 1511

forwardedToNumber.....identifier of [5] ISDN-AddressString
 DEFINED in MAP-CH-DataTypes : 186

forwardedToNumber.....identifier of [4] AddressString
 DEFINED in MAP-SS-DataTypes : 72

forwardedToNumber.....identifier of [5] ISDN-AddressString
 DEFINED in MAP-SS-DataTypes : 97

forwardedToSubaddress.....identifier of [8] ISDN-SubaddressString
 DEFINED in MAP-MS-DataTypes : 669

forwardedToSubaddress.....identifier of [4] ISDN-SubaddressString
 DEFINED in MAP-MS-DataTypes : 1512

forwardedToSubaddress.....identifier of [4] ISDN-SubaddressString
 DEFINED in MAP-CH-DataTypes : 190

forwardedToSubaddress.....identifier of [6] ISDN-SubaddressString
 DEFINED in MAP-SS-DataTypes : 73

forwardedToSubaddress.....identifier of [8] ISDN-SubaddressString

DEFINED in MAP-SS-DataTypes : 98

34 TAG R4.21 Cross Reference Listing for MAP-Protocol 00-01-06 07:32:40 PAGE

VALUE forwardGroupCallSignalling.....value reference ForwardGroupCallSignalling, CHOICE

DEFINED in MAP-Protocol : 293

ForwardGroupCallSignalling.....type reference OPERATION

DEFINED in MAP-Group-Call-Operati : 67

USED in MAP-Protocol : 100 293

USED in MAP-Group-Call-Operati : 15

forwardGroupCallSignallingArg.....identifier of ForwardGroupCallSignallingArg

DEFINED in MAP-Group-Call-Operati : 69

ForwardGroupCallSignallingArg.....type reference SEQUENCE

DEFINED in MAP-GR-DataTypes : 75

USED in MAP-Group-Call-Operati : 35 69

USED in MAP-GR-DataTypes : 18

forwarding.....identifier of Named Number, 1

DEFINED in MAP-CH-DataTypes : 121

forwardingData.....identifier of ForwardingData

DEFINED in MAP-CH-DataTypes : 183

ForwardingData.....type reference SEQUENCE

DEFINED in MAP-CH-DataTypes : 185

USED in MAP-CH-DataTypes : 183 223 263

forwardingData.....identifier of [2] ForwardingData

DEFINED in MAP-CH-DataTypes : 223

forwardingData.....identifier of ForwardingData

DEFINED in MAP-CH-DataTypes : 263

forwardingFailed.....value reference ForwardingFailed, CHOICE VALUE

DEFINED in MAP-Protocol : 373

ForwardingFailed.....type reference ERROR

DEFINED in MAP-Errors : 299

USED in MAP-Protocol : 139 373

USED in MAP-CallHandlingOperat : 45 125

USED in MAP-Errors : 50

forwardingFailedParam.....identifier of ForwardingFailedParam

DEFINED in MAP-Errors : 301

ForwardingFailedParam.....type reference SEQUENCE

DEFINED in MAP-ER-DataTypes : 273

USED in MAP-Errors : 129 301

USED in MAP-ER-DataTypes : 38

ForwardingFeature.....type reference SEQUENCE

DEFINED in MAP-SS-DataTypes : 94

USED in MAP-SS-DataTypes : 92

forwardingFeatureList.....identifier of Ext-ForwFeatureList

DEFINED in MAP-MS-DataTypes : 655

forwardingFeatureList.....identifier of Ext-ForwFeatureList

DEFINED in MAP-MS-DataTypes : 1456

forwardingFeatureList.....identifier of [1] Ext-ForwFeatureList

DEFINED in MAP-MS-DataTypes : 1576

forwardingFeatureList.....identifier of ForwardingFeatureList

DEFINED in MAP-SS-DataTypes : 87

ForwardingFeatureList.....type reference SEQUENCE OF

DEFINED in MAP-SS-DataTypes : 90

USED in MAP-SS-DataTypes : 87 207

forwardingFeatureList.....identifier of [3] ForwardingFeatureList

DEFINED in MAP-SS-DataTypes : 207

forwardingInfo.....identifier of [0] Ext-ForwInfo

DEFINED in MAP-MS-DataTypes : 646

forwardingInfo.....identifier of [0] ForwardingInfo

DEFINED in MAP-SS-DataTypes : 81

ForwardingInfo.....type reference SEQUENCE

DEFINED in MAP-SS-DataTypes	:	85
USED in MAP-SS-DataTypes	:	81

35

TAG R4.21 Cross Reference Listing for MAP-Protocol 00-01-06 07:32:40 PAGE

forwardingInfoFor-CSE.....identifier of [0] Ext-ForwardingInfoFor-CSE
 DEFINED in MAP-MS-DataTypes : 1570

forwardingInterrogationRequired.....identifier of [4] NULL
 DEFINED in MAP-CH-DataTypes : 154

forwardingOptions.....identifier of [6] Ext-ForwOptions
 DEFINED in MAP-MS-DataTypes : 670

forwardingOptions.....identifier of [6] ForwardingOptions
 DEFINED in MAP-CH-DataTypes : 191

forwardingOptions.....identifier of [6] ForwardingOptions
 DEFINED in MAP-SS-DataTypes : 99

ForwardingOptions.....type reference OCTET STRING
 DEFINED in MAP-SS-DataTypes : 118
 USED in MAP-CH-DataTypes : 56 191
 USED in MAP-SS-DataTypes : 31 99

forwardingReason.....identifier of [8] ForwardingReason
 DEFINED in MAP-CH-DataTypes : 102

ForwardingReason.....type reference ENUMERATED
 DEFINED in MAP-CH-DataTypes : 127
 USED in MAP-CH-DataTypes : 102

forwardingViolation.....value reference ForwardingViolation, CHOICE VALUE
 DEFINED in MAP-Protocol : 375

ForwardingViolation.....type reference ERROR
 DEFINED in MAP-Errors : 294
 USED in MAP-Protocol : 138 375
 USED in MAP-CallHandlingOperat : 44 101
 USED in MAP-Errors : 49

forwardingViolationParam.....identifier of ForwardingViolationParam
 DEFINED in MAP-Errors : 296

ForwardingViolationParam.....type reference SEQUENCE
 DEFINED in MAP-ER-DataTypes : 269
 USED in MAP-Errors : 128 296
 USED in MAP-ER-DataTypes : 37

freezeP-TMSI.....identifier of [1] NULL
 DEFINED in MAP-MS-DataTypes : 250

freezeTMSI.....identifier of [0] NULL
 DEFINED in MAP-MS-DataTypes : 249

generalProblem.....identifier of [0] IMPLICIT GeneralProblem
 DEFINED in TCAPMessages : 170

GeneralProblem.....type reference INTEGER
 DEFINED in TCAPMessages : 179
 USED in TCAPMessages : 170

general-dataCDA.....value reference BearerServiceCode, '00010111'B
 DEFINED in MAP-BS-Code : 58

general-dataCDS.....value reference BearerServiceCode, '00011111'B
 DEFINED in MAP-BS-Code : 65

general-dataPDS.....value reference BearerServiceCode, '00101111'B
 DEFINED in MAP-BS-Code : 80

general-padAccessCA.....value reference BearerServiceCode, '00100111'B
 DEFINED in MAP-BS-Code : 74

GenericServiceInfo.....type reference SEQUENCE
 DEFINED in MAP-SS-DataTypes : 182
 USED in MAP-SS-DataTypes : 208

genericServiceInfo.....identifier of [4] GenericServiceInfo
 DEFINED in MAP-SS-DataTypes : 208

geodeticInformation.....identifier of [7] GeodeticInformation
 DEFINED in MAP-MS-DataTypes : 1362

GeodeticInformation.....type reference OCTET STRING

DEFINED in MAP-MS-DataTypes : 1374
USED in MAP-MS-DataTypes : 1362

TAG R4.21 Cross Reference Listing for MAP-Protocol 00-01-06 07:32:40 PAGE

geographicalInformation.....identifier of [0] GeographicalInformation
 DEFINED in MAP-MS-DataTypes : 1354

GeographicalInformation.....type reference OCTET STRING
 DEFINED in MAP-MS-DataTypes : 1364
 USED in MAP-MS-DataTypes : 1354

getPassword.....value reference GetPassword, CHOICE VALUE
 DEFINED in MAP-Protocol : 251

GetPassword.....type reference OPERATION
 DEFINED in MAP-SupplementaryServi : 234
 USED in MAP-Protocol : 77 251
 USED in MAP-SupplementaryServi : 22 232

ggsn-Address.....identifier of [1] GSN-Address
 DEFINED in MAP-MS-DataTypes : 1239

ggsn-Address.....identifier of [1] GSN-Address
 DEFINED in MAP-MS-DataTypes : 1246

ggsn-Address.....identifier of [2] GSN-Address
 DEFINED in MAP-MS-DataTypes : 1256

ggsn-Address.....identifier of [0] GSN-Address
 DEFINED in MAP-MS-DataTypes : 1261

ggsn-Address.....identifier of [2] GSN-Address
 DEFINED in MAP-MS-DataTypes : 1270

ggsn-Number.....identifier of [2] ISDN-AddressString
 DEFINED in MAP-MS-DataTypes : 1240

ggsn-Number.....identifier of [1] ISDN-AddressString
 DEFINED in MAP-MS-DataTypes : 1255

GlobalCellId.....type reference OCTET STRING
 DEFINED in MAP-CommonDataTypes : 291
 USED in MAP-MS-DataTypes : 152 341 352
 USED in MAP-CommonDataTypes : 34

gmlc-List.....identifier of [0] GMLC-List
 DEFINED in MAP-MS-DataTypes : 419

GMLC-List.....type reference SEQUENCE OF
 DEFINED in MAP-MS-DataTypes : 424
 USED in MAP-MS-DataTypes : 419

gmlc-List.....identifier of Named Number, 0
 DEFINED in MAP-MS-DataTypes : 847

gmlc-ListWithdraw.....identifier of [13] NULL
 DEFINED in MAP-MS-DataTypes : 906

gmlc-Restriction.....identifier of [0] GMLC-Restriction
 DEFINED in MAP-MS-DataTypes : 841

GMLC-Restriction.....type reference ENUMERATED
 DEFINED in MAP-MS-DataTypes : 846
 USED in MAP-MS-DataTypes : 841

gmscCamelSubscriptionInfo.....identifier of [0] GmscCamelSubscriptionInfo
 DEFINED in MAP-CH-DataTypes : 264

GmscCamelSubscriptionInfo.....type reference SEQUENCE
 DEFINED in MAP-CH-DataTypes : 268
 USED in MAP-CH-DataTypes : 264

gmsc-Address.....identifier of [6] ISDN-AddressString
 DEFINED in MAP-CH-DataTypes : 100

gmsc-Address.....identifier of [8] ISDN-AddressString
 DEFINED in MAP-CH-DataTypes : 203

gprsConnectionSuspended.....identifier of NULL
 DEFINED in MAP-ER-DataTypes : 308

GPRSDataList.....type reference SEQUENCE OF
 DEFINED in MAP-MS-DataTypes : 439

USED in MAP-MS-DataTypes : 463

37

TAG R4.21 Cross Reference Listing for MAP-Protocol 00-01-06 07:32:40 PAGE

gprsDataList.....identifier of [1] GPRSDataList
 DEFINED in MAP-MS-DataTypes : 463

gprsEnhancementsSupportIndicator.....identifier of [3] NULL
 DEFINED in MAP-MS-DataTypes : 327

gprsNodeIndicator.....identifier of [5] NULL
 DEFINED in MAP-SM-DataTypes : 90

gprsSubscriptionData.....identifier of [16] GPRSSubscriptionData
 DEFINED in MAP-MS-DataTypes : 403

GPRSSubscriptionData.....type reference SEQUENCE
 DEFINED in MAP-MS-DataTypes : 458
 USED in MAP-MS-DataTypes : 403

gprsSubscriptionDataWithdraw.....identifier of [10] GPRSSubscriptionDataWithdraw
 DEFINED in MAP-MS-DataTypes : 903

GPRSSubscriptionDataWithdraw.....type reference CHOICE
 DEFINED in MAP-MS-DataTypes : 909
 USED in MAP-MS-DataTypes : 903

gprsSubscriptionUnknown.....identifier of Named Number, 1
 DEFINED in MAP-ER-DataTypes : 203

gprsSupportIndicator.....identifier of [7] NULL
 DEFINED in MAP-SM-DataTypes : 58

gprsSupportIndicator.....identifier of [2] NULL
 DEFINED in MAP-SM-DataTypes : 151

GPRS-CamelTDPData.....type reference SEQUENCE
 DEFINED in MAP-MS-DataTypes : 489
 USED in MAP-MS-DataTypes : 485

gprs-CamelTDPDataList.....identifier of [0] GPRS-CamelTDPDataList
 DEFINED in MAP-MS-DataTypes : 475

GPRS-CamelTDPDataList.....type reference SEQUENCE OF
 DEFINED in MAP-MS-DataTypes : 484
 USED in MAP-MS-DataTypes : 475

gprs-CSI.....identifier of [0] GPRS-CSI
 DEFINED in MAP-MS-DataTypes : 469

GPRS-CSI.....type reference SEQUENCE
 DEFINED in MAP-MS-DataTypes : 474
 USED in MAP-MS-DataTypes : 469 1486

gprs-CSI.....identifier of Named Number, 4
 DEFINED in MAP-MS-DataTypes : 1449

gprs-CSI.....identifier of [8] GPRS-CSI
 DEFINED in MAP-MS-DataTypes : 1486

gprs-TriggerDetectionPoint.....identifier of [0] GPRS-TriggerDetectionPoint
 DEFINED in MAP-MS-DataTypes : 490

GPRS-TriggerDetectionPoint.....type reference ENUMERATED
 DEFINED in MAP-MS-DataTypes : 506
 USED in MAP-MS-DataTypes : 490

greyListed.....identifier of Named Number, 2
 DEFINED in MAP-MS-DataTypes : 391

groupCallNumber.....identifier of ISDN-AddressString
 DEFINED in MAP-GR-DataTypes : 62

groupId.....identifier of GroupId
 DEFINED in MAP-MS-DataTypes : 1312

groupid.....identifier of GroupId
 DEFINED in MAP-MS-DataTypes : 1317

GroupId.....type reference OCTET STRING
 DEFINED in MAP-MS-DataTypes : 1322
 USED in MAP-MS-DataTypes : 1312 1317

groupKey.....identifier of [1] Kc

DEFINED in MAP-GR-DataTypes : 55

TAG R4.21 Cross Reference Listing for MAP-Protocol 00-01-06 07:32:40 PAGE

groupKeyNumber.....identifier of [0] GroupKeyNumber
 DEFINED in MAP-GR-DataTypes : 54

GroupKeyNumber.....type reference INTEGER
 DEFINED in MAP-GR-DataTypes : 92
 USED in MAP-GR-DataTypes : 54

gsmSCF-Address.....identifier of [2] ISDN-AddressString
 DEFINED in MAP-MS-DataTypes : 492

gsmSCF-Address.....identifier of ISDN-AddressString
 DEFINED in MAP-MS-DataTypes : 962

gsmSCF-Address.....identifier of ISDN-AddressString
 DEFINED in MAP-MS-DataTypes : 974

gsmSCF-Address.....identifier of [0] ISDN-AddressString
 DEFINED in MAP-MS-DataTypes : 1016

gsmSCF-Address.....identifier of [2] ISDN-AddressString
 DEFINED in MAP-MS-DataTypes : 1146

gsmSCF-Address.....identifier of [0] ISDN-AddressString
 DEFINED in MAP-MS-DataTypes : 1171

gsmSCF-Address.....identifier of [0] ISDN-AddressString
 DEFINED in MAP-MS-DataTypes : 1219

gsmSCF-Address.....identifier of [3] ISDN-AddressString
 DEFINED in MAP-MS-DataTypes : 1406

gsmSCF-Address.....identifier of [2] ISDN-AddressString
 DEFINED in MAP-MS-DataTypes : 1421

gsmSCF-Address.....identifier of [1] ISDN-AddressString
 DEFINED in MAP-MS-DataTypes : 1495

gsm-0408.....identifier of Named Number, 1
 DEFINED in MAP-CommonDataTypes : 197

gsm-0806.....identifier of Named Number, 2
 DEFINED in MAP-CommonDataTypes : 198

gsm-BearerCapability.....identifier of [5] ExternalSignalInfo
 DEFINED in MAP-CH-DataTypes : 200

gsm-BearerCapability.....identifier of [0] ExternalSignalInfo
 DEFINED in MAP-CH-DataTypes : 278

gsm-BSSMAP.....identifier of Named Number, 3
 DEFINED in MAP-CommonDataTypes : 199

GSN-Address.....type reference OCTET STRING
 DEFINED in MAP-MS-DataTypes : 330
 USED in MAP-MS-DataTypes : 317 1239 1245 1246 1256 1261 1269 1270

guidanceInfo.....identifier of GuidanceInfo
 DEFINED in MAP-SupplementaryServi : 236

GuidanceInfo.....type reference ENUMERATED
 DEFINED in MAP-SS-DataTypes : 237
 USED in MAP-SupplementaryServi : 67 236
 USED in MAP-SS-DataTypes : 25

handoverNumber.....identifier of ISDN-AddressString
 DEFINED in MAP-MS-DataTypes : 347

highLayerCompatibility.....identifier of [6] ExternalSignalInfo
 DEFINED in MAP-CH-DataTypes : 284

hlr.....identifier of Named Number, 1
 DEFINED in MAP-CommonDataTypes : 306

HLR-Id.....type reference IMSI
 DEFINED in MAP-CommonDataTypes : 280
 USED in MAP-CommonDataTypes : 285

hlr-List.....identifier of HLR-List
 DEFINED in MAP-MS-DataTypes : 1283

HLR-List.....type reference SEQUENCE OF
DEFINED in MAP-CommonDataTypes : 284

39 TAG R4.21 Cross Reference Listing for MAP-Protocol 00-01-06 07:32:40 PAGE

USED in MAP-MS-DataTypes : 149 1283
 USED in MAP-CommonDataTypes : 32

hlr-Number.....identifier of ISDN-AddressString
 DEFINED in MAP-MS-DataTypes : 218

hlr-Number.....identifier of ISDN-AddressString
 DEFINED in MAP-MS-DataTypes : 334

hlr-Number.....identifier of ISDN-AddressString
 DEFINED in MAP-MS-DataTypes : 1282

hlr-Number.....identifier of ISDN-AddressString
 DEFINED in MAP-MS-DataTypes : 1294

hold.....value reference SS-Code, '01000010'B
 DEFINED in MAP-SS-Code : 77

home-Country.....identifier of Named Number, 1
 DEFINED in MAP-MS-DataTypes : 848

horizontal-accuracy.....identifier of [0] Horizontal-Accuracy
 DEFINED in MAP-LCS-DataTypes : 138

Horizontal-Accuracy.....type reference OCTET STRING
 DEFINED in MAP-LCS-DataTypes : 144
 USED in MAP-LCS-DataTypes : 20 138

ho-NumberNotRequired.....identifier of NULL
 DEFINED in MAP-MS-DataTypes : 342

identity.....identifier of Identity
 DEFINED in MAP-MS-DataTypes : 224

Identity.....type reference CHOICE
 DEFINED in MAP-CommonDataTypes : 253
 USED in MAP-MS-DataTypes : 151 224
 USED in MAP-CommonDataTypes : 29

ik.....identifier of IK
 DEFINED in MAP-MS-DataTypes : 292

IK.....type reference OCTET STRING
 DEFINED in MAP-MS-DataTypes : 306
 USED in MAP-MS-DataTypes : 292

illegalEquipment.....value reference IllegalEquipment, CHOICE VALUE
 DEFINED in MAP-Protocol : 346

IllegalEquipment.....type reference ERROR
 DEFINED in MAP-Errors : 231
 USED in MAP-Protocol : 126 346
 USED in MAP-SupplementaryServi : 52 199 213
 USED in MAP-ShortMessageServic : 34 106
 USED in MAP-LocationServiceOpe : 35 78
 USED in MAP-Errors : 31

illegalEquipmentParam.....identifier of IllegalEquipmentParam
 DEFINED in MAP-Errors : 233

IllegalEquipmentParam.....type reference SEQUENCE
 DEFINED in MAP-ER-DataTypes : 221
 USED in MAP-Errors : 118 233
 USED in MAP-ER-DataTypes : 29

illegalSS-Operation.....value reference IllegalSS-Operation, CHOICE VALUE
 DEFINED in MAP-Protocol : 394

IllegalSS-Operation.....type reference ERROR
 DEFINED in MAP-Errors : 340
 USED in MAP-Protocol : 142 394
 USED in MAP-MobileServiceOpera : 99 263 280
 USED in MAP-SupplementaryServi : 40 100 117 134 154 172 261 278
 USED in MAP-Errors : 62

illegalSS-OperationParam.....identifier of IllegalSS-OperationParam
 DEFINED in MAP-Errors : 342

IllegalSS-OperationParam.....type reference SEQUENCE
 DEFINED in MAP-ER-DataTypes : 289

USED in MAP-Errors	:	147	342
USED in MAP-ER-DataTypes	:	56	

40

TAG R4.21 Cross Reference Listing for MAP-Protocol 00-01-06 07:32:40 PAGE

illegalSubscriber.....value reference IllegalSubscriber, CHOICE VALUE
 DEFINED in MAP-Protocol : 345

IllegalSubscriber.....type reference ERROR
 DEFINED in MAP-Errors : 225
 USED in MAP-Protocol : 125 345
 USED in MAP-SupplementaryServi : 51 198 212
 USED in MAP-ShortMessageServic : 33 105
 USED in MAP-LocationServiceOpe : 36 77
 USED in MAP-Errors : 30

illegalSubscriberParam.....identifier of IllegalSubscriberParam
 DEFINED in MAP-Errors : 227

IllegalSubscriberParam.....type reference SEQUENCE
 DEFINED in MAP-ER-DataTypes : 217
 USED in MAP-Errors : 117 227
 USED in MAP-ER-DataTypes : 28

imei.....identifier of IMEI
 DEFINED in MAP-MobileServiceOpera : 354

IMEI.....type reference TBCD-STRING
 DEFINED in MAP-CommonDataTypes : 273
 USED in MAP-MobileServiceOpera : 159 354
 USED in MAP-CommonDataTypes : 31
 USED in MAP-LCS-DataTypes : 28 78 222

imei.....identifier of [5] IMEI
 DEFINED in MAP-LCS-DataTypes : 78

imei.....identifier of [2] IMEI
 DEFINED in MAP-LCS-DataTypes : 222

immediateResponsePreferred.....identifier of [1] NULL
 DEFINED in MAP-MS-DataTypes : 365

imsi.....identifier of IMSI
 DEFINED in MAP-OperationAndMainte : 81

imsi.....identifier of IMSI
 DEFINED in MAP-MS-DataTypes : 185

imsi.....identifier of IMSI
 DEFINED in MAP-MS-DataTypes : 242

imsi.....identifier of IMSI
 DEFINED in MAP-MS-DataTypes : 264

imsi.....identifier of IMSI
 DEFINED in MAP-MS-DataTypes : 315

imsi.....identifier of [0] IMSI
 DEFINED in MAP-MS-DataTypes : 360

imsi.....identifier of [0] IMSI
 DEFINED in MAP-MS-DataTypes : 397

imsi.....identifier of [0] IMSI
 DEFINED in MAP-MS-DataTypes : 891

imsi.....identifier of [0] IMSI
 DEFINED in MAP-MS-DataTypes : 1238

imsi.....identifier of [0] IMSI
 DEFINED in MAP-MS-DataTypes : 1254

imsi.....identifier of [0] IMSI
 DEFINED in MAP-MS-DataTypes : 1268

imsi.....identifier of IMSI
 DEFINED in MAP-MS-DataTypes : 1287

imsi.....identifier of [0] IMSI
 DEFINED in MAP-MS-DataTypes : 1329

imsi.....identifier of IMSI
 DEFINED in MAP-MS-DataTypes : 1532

imsi.....identifier of [1] IMSI
DEFINED in MAP-MS-DataTypes : 1557

41

TAG R4.21 Cross Reference Listing for MAP-Protocol 00-01-06 07:32:40 PAGE

```

IMSI.....type reference TBCD-STRING
  DEFINED in MAP-CommonDataTypes : 250
  USED in MAP-OperationAndMainte : 43 81
  USED in MAP-MS-DataTypes : 147 185 242 264 315 360 397 891 1238
    1254 1268 1287 1329 1532 1557
  USED in MAP-CommonDataTypes : 27 254 258 270 280 327
  USED in MAP-OM-DataTypes : 22 37 55
  USED in MAP-CH-DataTypes : 67 144 196 224 313 343 383 414 426
  USED in MAP-SS-DataTypes : 47 259
  USED in MAP-SM-DataTypes : 34 80 112 133 194
  USED in MAP-GR-DataTypes : 24 67 76
  USED in MAP-LCS-DataTypes : 29 75 221

imsi.....identifier of IMSI
  DEFINED in MAP-CommonDataTypes : 254

imsi.....identifier of IMSI
  DEFINED in MAP-CommonDataTypes : 258

imsi.....identifier of [0] IMSI
  DEFINED in MAP-CommonDataTypes : 270

imsi.....identifier of [0] IMSI
  DEFINED in MAP-CommonDataTypes : 327

imsi.....identifier of [0] IMSI
  DEFINED in MAP-OM-DataTypes : 37

imsi.....identifier of [0] IMSI
  DEFINED in MAP-OM-DataTypes : 55

imsi.....identifier of [9] IMSI
  DEFINED in MAP-CH-DataTypes : 144

imsi.....identifier of [0] IMSI
  DEFINED in MAP-CH-DataTypes : 196

imsi.....identifier of [3] IMSI
  DEFINED in MAP-CH-DataTypes : 224

imsi.....identifier of [0] IMSI
  DEFINED in MAP-CH-DataTypes : 313

imsi.....identifier of [0] IMSI
  DEFINED in MAP-CH-DataTypes : 343

imsi.....identifier of [0] IMSI
  DEFINED in MAP-CH-DataTypes : 383

imsi.....identifier of [0] IMSI
  DEFINED in MAP-CH-DataTypes : 414

imsi.....identifier of [0] IMSI
  DEFINED in MAP-CH-DataTypes : 426

imsi.....identifier of [0] IMSI
  DEFINED in MAP-SS-DataTypes : 259

imsi.....identifier of IMSI
  DEFINED in MAP-SM-DataTypes : 80

imsi.....identifier of IMSI
  DEFINED in MAP-SM-DataTypes : 112

imsi.....identifier of [0] IMSI
  DEFINED in MAP-SM-DataTypes : 133

imsi.....identifier of [0] IMSI
  DEFINED in MAP-SM-DataTypes : 194

imsi.....identifier of IMSI
  DEFINED in MAP-GR-DataTypes : 67

imsi.....identifier of IMSI
  DEFINED in MAP-GR-DataTypes : 76

imsi.....identifier of [2] IMSI
  DEFINED in MAP-LCS-DataTypes : 75
    
```

```
imsi.....identifier of [1] IMSI  
  DEFINED in MAP-LCS-DataTypes      :    221
```

42

TAG R4.21 Cross Reference Listing for MAP-Protocol 00-01-06 07:32:40 PAGE

imsiDetach.....identifier of Named Number, 0
 DEFINED in MAP-ER-DataTypes : 247

imsiDetached.....identifier of Named Number, 1
 DEFINED in MAP-MS-DataTypes : 1397

imsiUnknown.....identifier of Named Number, 0
 DEFINED in MAP-ER-DataTypes : 202

imsi-WithLMSI.....identifier of IMSI-WithLMSI
 DEFINED in MAP-CommonDataTypes : 255

IMSI-WithLMSI.....type reference SEQUENCE
 DEFINED in MAP-CommonDataTypes : 257
 USED in MAP-CommonDataTypes : 255

incomingCallsBarredWithinCUG.....identifier of Named Number, 0
 DEFINED in MAP-ER-DataTypes : 122

incompatibleTerminal.....value reference IncompatibleTerminal, CHOICE VALUE
 DEFINED in MAP-Protocol : 329

IncompatibleTerminal.....type reference ERROR
 DEFINED in MAP-Errors : 185
 USED in MAP-Protocol : 160 329
 USED in MAP-CallHandlingOperat : 48 187
 USED in MAP-Errors : 18

incompatibleTerminalParam.....identifier of IncompatibleTerminalParam
 DEFINED in MAP-Errors : 187

IncompatibleTerminalParam.....type reference SEQUENCE
 DEFINED in MAP-ER-DataTypes : 324
 USED in MAP-Errors : 137 187
 USED in MAP-ER-DataTypes : 46

inconsistentMeasurementData.....identifier of Named Number, 3
 DEFINED in MAP-ER-DataTypes : 362

incorrectTransactionPortion.....identifier of Named Number, 3
 DEFINED in TCAPMessages : 106

informationNotAvailable.....value reference InformationNotAvailable, CHOICE VALUE
 DEFINED in MAP-Protocol : 385

InformationNotAvailable.....type reference ERROR
 DEFINED in MAP-Errors : 332
 USED in MAP-Protocol : 167 385
 USED in MAP-MobileServiceOpera : 104 265 284
 USED in MAP-Errors : 59

informationNotAvailableParam.....identifier of InformationNotAvailableParam
 DEFINED in MAP-Errors : 334

InformationNotAvailableParam.....type reference SEQUENCE
 DEFINED in MAP-ER-DataTypes : 301
 USED in MAP-Errors : 150 334
 USED in MAP-ER-DataTypes : 59

informServiceCentre.....value reference InformServiceCentre, CHOICE VALUE
 DEFINED in MAP-Protocol : 262

InformServiceCentre.....type reference OPERATION
 DEFINED in MAP-ShortMessageServic : 132
 USED in MAP-Protocol : 91 262
 USED in MAP-ShortMessageServic : 18

informServiceCentreArg.....identifier of InformServiceCentreArg
 DEFINED in MAP-ShortMessageServic : 134

InformServiceCentreArg.....type reference SEQUENCE
 DEFINED in MAP-SM-DataTypes : 179
 USED in MAP-ShortMessageServic : 54 134
 USED in MAP-SM-DataTypes : 23

inhibiting.....identifier of Named Number, 0
 DEFINED in MAP-MS-DataTypes : 1089

initialLocation.....identifier of Named Number, 2

DEFINED in MAP-LCS-DataTypes : 93

43

TAG R4.21 Cross Reference Listing for MAP-Protocol 00-01-06 07:32:40 PAGE

initiatingRelease.....identifier of Named Number, 4
 DEFINED in TCAPMessages : 187

insertSubscriberData.....value reference InsertSubscriberData, CHOICE VALUE
 DEFINED in MAP-Protocol : 206

InsertSubscriberData.....type reference OPERATION
 DEFINED in MAP-MobileServiceOpera : 364
 USED in MAP-Protocol : 24 206
 USED in MAP-MobileServiceOpera : 51

insertSubscriberDataArg.....identifier of InsertSubscriberDataArg
 DEFINED in MAP-MobileServiceOpera : 366

InsertSubscriberDataArg.....type reference SEQUENCE
 DEFINED in MAP-MS-DataTypes : 396
 USED in MAP-MobileServiceOpera : 127 366
 USED in MAP-MS-DataTypes : 43

insertSubscriberDataRes.....identifier of InsertSubscriberDataRes
 DEFINED in MAP-MobileServiceOpera : 368

InsertSubscriberDataRes.....type reference SEQUENCE
 DEFINED in MAP-MS-DataTypes : 873
 USED in MAP-MobileServiceOpera : 128 368
 USED in MAP-MS-DataTypes : 44

insufficientMeasurementData.....identifier of Named Number, 2
 DEFINED in MAP-ER-DataTypes : 361

insufficientResources.....identifier of Named Number, 1
 DEFINED in MAP-ER-DataTypes : 360

interCUG-Restrictions.....identifier of InterCUG-Restrictions
 DEFINED in MAP-MS-DataTypes : 787

InterCUG-Restrictions.....type reference OCTET STRING
 DEFINED in MAP-MS-DataTypes : 791
 USED in MAP-MS-DataTypes : 65 787

internationalECT-Barred.....identifier of Named Number, 11
 DEFINED in MAP-MS-DataTypes : 627

internationalOGCallsBarred.....identifier of Named Number, 1
 DEFINED in MAP-MS-DataTypes : 617

internationalOGCallsNotToHPLMN-CountryBaidentifier of Named Number, 2
 DEFINED in MAP-MS-DataTypes : 618

interrogateSS.....value reference InterrogateSS, CHOICE VALUE
 DEFINED in MAP-Protocol : 245

InterrogateSS.....type reference OPERATION
 DEFINED in MAP-SupplementaryServi : 160
 USED in MAP-Protocol : 72 245
 USED in MAP-SupplementaryServi : 17

interrogateSS-Res.....identifier of InterrogateSS-Res
 DEFINED in MAP-SupplementaryServi : 164

InterrogateSS-Res.....type reference CHOICE
 DEFINED in MAP-SS-DataTypes : 204
 USED in MAP-SupplementaryServi : 63 164
 USED in MAP-SS-DataTypes : 19

interrogationType.....identifier of [3] InterrogationType
 DEFINED in MAP-CH-DataTypes : 97

InterrogationType.....type reference ENUMERATED
 DEFINED in MAP-CH-DataTypes : 119
 USED in MAP-CH-DataTypes : 97

interzonalECT-Barred.....identifier of Named Number, 12
 DEFINED in MAP-MS-DataTypes : 628

interzonalOGCallsAndInternationalOGCallsidentifier of Named Number, 8
 DEFINED in MAP-MS-DataTypes : 621

interzonalOGCallsBarred.....identifier of Named Number, 6
 DEFINED in MAP-MS-DataTypes : 619

interzonalOGCallsNotToHPLMN-CountryBarreidentifier of Named Number, 7

44

TAG R4.21 Cross Reference Listing for MAP-Protocol 00-01-06 07:32:40 PAGE

DEFINED in MAP-MS-DataTypes : 620

intraCUG-Options.....identifier of IntraCUG-Options
 DEFINED in MAP-MS-DataTypes : 759

IntraCUG-Options.....type reference ENUMERATED
 DEFINED in MAP-MS-DataTypes : 769
 USED in MAP-MS-DataTypes : 66 759

invalidFormat.....identifier of Named Number, 1
 DEFINED in MAP-ER-DataTypes : 135

invalidSME-Address.....identifier of Named Number, 5
 DEFINED in MAP-ER-DataTypes : 145

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 : 131
 USED in MAP-OperationAndMainte : 42 79
 USED in MAP-MS-DataTypes : 144 187 188 218 243 244 316 334 347
 353 425 492 574 665 960 962 974 1016
 1066 1146 1171 1219 1240 1255 1282 1294 1355
 1361 1406 1421 1495 1533 1558
 USED in MAP-CommonDataTypes : 17 328
 USED in MAP-CH-DataTypes : 63 94 100 155 161 182 186 197 198
 203 216 229 281 297 386
 USED in MAP-SS-DataTypes : 45 97 197 215 260 288
 USED in MAP-SM-DataTypes : 32 53 86 98 99 139 144 169 175
 180
 USED in MAP-GR-DataTypes : 23 62
 USED in MAP-LCS-DataTypes : 27 53 65 72 76 220 223 224

isdn-BearerCapability.....identifier of [1] ExternalSignalInfo
 DEFINED in MAP-CH-DataTypes : 279

ISDN-SubaddressString.....type reference OCTET STRING
 DEFINED in MAP-CommonDataTypes : 137
 USED in MAP-MS-DataTypes : 145 669 1512
 USED in MAP-CommonDataTypes : 19
 USED in MAP-CH-DataTypes : 64 190
 USED in MAP-SS-DataTypes : 46 73 98 198

istAlert.....value reference IST-Alert, CHOICE VALUE
 DEFINED in MAP-Protocol : 235

istAlertArg.....identifier of IST-AlertArg
 DEFINED in MAP-CallHandlingOperat : 194

istAlertRes.....identifier of IST-AlertRes
 DEFINED in MAP-CallHandlingOperat : 196

istAlertTimer.....identifier of [26] IST-AlertTimerValue
DEFINED in MAP-MS-DataTypes : 410

istAlertTimer.....identifier of [14] IST-AlertTimerValue
 DEFINED in MAP-CH-DataTypes : 163

istAlertTimer.....identifier of [0] IST-AlertTimerValue
 DEFINED in MAP-CH-DataTypes : 419

istCommand.....value reference IST-Command, CHOICE VALUE
 DEFINED in MAP-Protocol : 236

istCommandArg.....identifier of IST-CommandArg
 DEFINED in MAP-CallHandlingOperat : 207

istCommandRes.....identifier of IST-CommandRes
 DEFINED in MAP-CallHandlingOperat : 209

istCommandSupported.....identifier of Named Number, 1
 DEFINED in MAP-MS-DataTypes : 212

istInformationWithdraw.....identifier of [14] NULL
 DEFINED in MAP-MS-DataTypes : 907

istInformationWithdraw.....identifier of [1] NULL
 DEFINED in MAP-CH-DataTypes : 420

istSupportIndicator.....identifier of [1] IST-SupportIndicator
 DEFINED in MAP-MS-DataTypes : 199

istSupportIndicator.....identifier of [18] IST-SupportIndicator
 DEFINED in MAP-CH-DataTypes : 113

IST-Alert.....type reference OPERATION
 DEFINED in MAP-CallHandlingOperat : 192
 USED in MAP-Protocol : 61 235
 USED in MAP-CallHandlingOperat : 21

IST-AlertArg.....type reference SEQUENCE
 DEFINED in MAP-CH-DataTypes : 413
 USED in MAP-CallHandlingOperat : 70 194
 USED in MAP-CH-DataTypes : 33

IST-AlertRes.....type reference SEQUENCE
 DEFINED in MAP-CH-DataTypes : 418
 USED in MAP-CallHandlingOperat : 71 196
 USED in MAP-CH-DataTypes : 34

IST-AlertTimerValue.....type reference INTEGER
 DEFINED in MAP-MS-DataTypes : 416
 USED in MAP-MS-DataTypes : 67 410
 USED in MAP-CH-DataTypes : 48 163 419

IST-Command.....type reference OPERATION
 DEFINED in MAP-CallHandlingOperat : 205
 USED in MAP-Protocol : 62 236
 USED in MAP-CallHandlingOperat : 22

IST-CommandArg.....type reference SEQUENCE
 DEFINED in MAP-CH-DataTypes : 425
 USED in MAP-CallHandlingOperat : 72 207
 USED in MAP-CH-DataTypes : 35

IST-CommandRes.....type reference SEQUENCE
 DEFINED in MAP-CH-DataTypes : 430
 USED in MAP-CallHandlingOperat : 73 209
 USED in MAP-CH-DataTypes : 36

IST-SupportIndicator.....type reference ENUMERATED
 DEFINED in MAP-MS-DataTypes : 210
 USED in MAP-MS-DataTypes : 26 199
 USED in MAP-CH-DataTypes : 47 113

kc.....identifier of Kc
 DEFINED in MAP-MS-DataTypes : 285

Kc.....type reference OCTET STRING
 DEFINED in MAP-MS-DataTypes : 300
 USED in MAP-MS-DataTypes : 40 285
 USED in MAP-GR-DataTypes : 36 55

keepCCBS-CallIndicator.....identifier of [1] NULL

DEFINED in MAP-CH-DataTypes : 177

46 TAG R4.21 Cross Reference Listing for MAP-Protocol 00-01-06 07:32:40 PAGE

laiFixedLength.....identifier of [1] LAIFixedLength
 DEFINED in MAP-CommonDataTypes : 349

LAIFixedLength.....type reference OCTET STRING
 DEFINED in MAP-CommonDataTypes : 364
 USED in MAP-CommonDataTypes : 349

lawfulInterceptServices.....identifier of Named Number, 3
 DEFINED in MAP-LCS-DataTypes : 111

lcsClientDialedByMS.....identifier of [2] AddressString
 DEFINED in MAP-LCS-DataTypes : 102

LCSCClientExternalID.....type reference SEQUENCE
 DEFINED in MAP-CommonDataTypes : 331
 USED in MAP-MS-DataTypes : 159 840
 USED in MAP-CommonDataTypes : 52
 USED in MAP-LCS-DataTypes : 33 101

lcsClientExternalID.....identifier of [1] LCSCClientExternalID
 DEFINED in MAP-LCS-DataTypes : 101

LCSCClientInternalID.....type reference ENUMERATED
 DEFINED in MAP-CommonDataTypes : 336
 USED in MAP-MS-DataTypes : 160 835
 USED in MAP-CommonDataTypes : 53
 USED in MAP-LCS-DataTypes : 34 103

lcsClientInternalID.....identifier of [3] LCSCClientInternalID
 DEFINED in MAP-LCS-DataTypes : 103

lcsClientName.....identifier of [4] LCSCClientName
 DEFINED in MAP-LCS-DataTypes : 104

LCSCClientName.....type reference SEQUENCE
 DEFINED in MAP-LCS-DataTypes : 118
 USED in MAP-LCS-DataTypes : 18 104

lcsClientType.....identifier of [0] LCSCClientType
 DEFINED in MAP-LCS-DataTypes : 100

LCSCClientType.....type reference ENUMERATED
 DEFINED in MAP-LCS-DataTypes : 107
 USED in MAP-LCS-DataTypes : 100

lcsInformation.....identifier of [22] LCSInformation
 DEFINED in MAP-MS-DataTypes : 409

LCSInformation.....type reference SEQUENCE
 DEFINED in MAP-MS-DataTypes : 418
 USED in MAP-MS-DataTypes : 409

lcsLocationInfo.....identifier of [1] LCSLocationInfo
 DEFINED in MAP-LCS-DataTypes : 60

LCSLocationInfo.....type reference SEQUENCE
 DEFINED in MAP-LCS-DataTypes : 64
 USED in MAP-LCS-DataTypes : 60 219

lcsLocationInfo.....identifier of LCSLocationInfo
 DEFINED in MAP-LCS-DataTypes : 219

lcs-ClientID.....identifier of [0] LCS-ClientID
 DEFINED in MAP-LCS-DataTypes : 73

LCS-ClientID.....type reference SEQUENCE
 DEFINED in MAP-LCS-DataTypes : 99
 USED in MAP-LCS-DataTypes : 73 218

lcs-ClientID.....identifier of LCS-ClientID
 DEFINED in MAP-LCS-DataTypes : 218

lcs-Event.....identifier of LCS-Event
 DEFINED in MAP-LCS-DataTypes : 217

LCS-Event.....type reference ENUMERATED
 DEFINED in MAP-LCS-DataTypes : 232
 USED in MAP-LCS-DataTypes : 217

lcs-Priority.....identifier of [6] LCS-Priority

DEFINED in MAP-LCS-DataTypes : 79

47

TAG R4.21 Cross Reference Listing for MAP-Protocol 00-01-06 07:32:40 PAGE

LCS-Priority.....type reference OCTET STRING
 DEFINED in MAP-LCS-DataTypes : 132
 USED in MAP-LCS-DataTypes : 79

LCS-PrivacyClass.....type reference SEQUENCE
 DEFINED in MAP-MS-DataTypes : 816
 USED in MAP-MS-DataTypes : 812

lcs-PrivacyExceptionList.....identifier of [1] LCS-PrivacyExceptionList
 DEFINED in MAP-MS-DataTypes : 420

LCS-PrivacyExceptionList.....type reference SEQUENCE OF
 DEFINED in MAP-MS-DataTypes : 811
 USED in MAP-MS-DataTypes : 420

lcs-QoS.....identifier of [7] LCS-QoS
 DEFINED in MAP-LCS-DataTypes : 80

LCS-QoS.....type reference SEQUENCE
 DEFINED in MAP-LCS-DataTypes : 137
 USED in MAP-LCS-DataTypes : 19 80

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 : 189

lmsi.....identifier of LMSI
 DEFINED in MAP-MS-DataTypes : 1288

lmsi.....identifier of [1] LMSI
 DEFINED in MAP-MS-DataTypes : 1330

lmsi.....identifier of LMSI
 DEFINED in MAP-CommonDataTypes : 259

LMSI.....type reference OCTET STRING
 DEFINED in MAP-CommonDataTypes : 289
 USED in MAP-MS-DataTypes : 150 189 1288 1330
 USED in MAP-CommonDataTypes : 33 259
 USED in MAP-CH-DataTypes : 68 199 314
 USED in MAP-SM-DataTypes : 35 87 134
 USED in MAP-LCS-DataTypes : 30 66 77

lmsi.....identifier of [4] LMSI
 DEFINED in MAP-CH-DataTypes : 199

lmsi.....identifier of [1] LMSI
 DEFINED in MAP-CH-DataTypes : 314

lmsi.....identifier of LMSI
 DEFINED in MAP-SM-DataTypes : 87

lmsi.....identifier of [1] LMSI
 DEFINED in MAP-SM-DataTypes : 134

lmsi.....identifier of [0] LMSI
 DEFINED in MAP-LCS-DataTypes : 66

lmsi.....identifier of [4] LMSI
 DEFINED in MAP-LCS-DataTypes : 77

lmu-Indicator.....identifier of [21] NULL
 DEFINED in MAP-MS-DataTypes : 408

locationEstimate.....identifier of Ext-GeographicalInformation
 DEFINED in MAP-LCS-DataTypes : 165

locationEstimate.....identifier of [5] Ext-GeographicalInformation
 DEFINED in MAP-LCS-DataTypes : 225

locationEstimateType.....identifier of [0] LocationEstimateType
 DEFINED in MAP-LCS-DataTypes : 87

LocationEstimateType.....type reference ENUMERATED
 DEFINED in MAP-LCS-DataTypes : 90

USED in MAP-LCS-DataTypes : 87

48 TAG R4.21 Cross Reference Listing for MAP-Protocol 00-01-06 07:32:40 PAGE

locationInformation.....identifier of [0] LocationInformation
 DEFINED in MAP-MS-DataTypes : 1341

locationInformation.....identifier of [0] NULL
 DEFINED in MAP-MS-DataTypes : 1347

LocationInformation.....type reference SEQUENCE
 DEFINED in MAP-MS-DataTypes : 1352
 USED in MAP-MS-DataTypes : 80 1341 1559

locationInformation.....identifier of [3] LocationInformation
 DEFINED in MAP-MS-DataTypes : 1559

locationInfoWithLMSI.....identifier of [0] LocationInfoWithLMSI
 DEFINED in MAP-SM-DataTypes : 81

LocationInfoWithLMSI.....type reference SEQUENCE
 DEFINED in MAP-SM-DataTypes : 85
 USED in MAP-SM-DataTypes : 81

locationNumber.....identifier of [2] LocationNumber
 DEFINED in MAP-MS-DataTypes : 1356

LocationNumber.....type reference OCTET STRING
 DEFINED in MAP-MS-DataTypes : 1386
 USED in MAP-MS-DataTypes : 1356

locationProcedureNotCompleted.....identifier of Named Number, 4
 DEFINED in MAP-ER-DataTypes : 363

locationProcedureNotSupportedByTargetMS.....identifier of Named Number, 5
 DEFINED in MAP-ER-DataTypes : 364

locationType.....identifier of LocationType
 DEFINED in MAP-LCS-DataTypes : 71

LocationType.....type reference SEQUENCE
 DEFINED in MAP-LCS-DataTypes : 86
 USED in MAP-LCS-DataTypes : 17 71

longTermDenial.....value reference LongTermDenial, CHOICE VALUE
 DEFINED in MAP-Protocol : 406

LongTermDenial.....type reference ERROR
 DEFINED in MAP-Errors : 385
 USED in MAP-Protocol : 159 406
 USED in MAP-SupplementaryServi : 54 265
 USED in MAP-Errors : 73

longTermDenialParam.....identifier of LongTermDenialParam
 DEFINED in MAP-Errors : 387

LongTermDenialParam.....type reference SEQUENCE
 DEFINED in MAP-ER-DataTypes : 331
 USED in MAP-Errors : 139 387
 USED in MAP-ER-DataTypes : 48

lowdelay.....identifier of Named Number, 0
 DEFINED in MAP-LCS-DataTypes : 158

lowerLayerCompatibility.....identifier of [5] ExternalSignalInfo
 DEFINED in MAP-CH-DataTypes : 283

lsaActiveModeIndicator.....identifier of [2] NULL
 DEFINED in MAP-MS-DataTypes : 551

lsaActiveModeSupportIndicator.....identifier of [3] NULL
 DEFINED in MAP-MS-DataTypes : 552

LSAData.....type reference SEQUENCE
 DEFINED in MAP-MS-DataTypes : 548
 USED in MAP-MS-DataTypes : 544

LSADataList.....type reference SEQUENCE OF
 DEFINED in MAP-MS-DataTypes : 543
 USED in MAP-MS-DataTypes : 562

lsaDataList.....identifier of [2] LSADataList
 DEFINED in MAP-MS-DataTypes : 562

lsaIdentity.....identifier of [0] LSAIdentity
DEFINED in MAP-MS-DataTypes : 549

49

TAG R4.21 Cross Reference Listing for MAP-Protocol 00-01-06 07:32:40 PAGE

```

LSAIdentity.....type reference OCTET STRING
  DEFINED in MAP-MS-DataTypes : 566
  USED in MAP-MS-DataTypes : 549 921 1360 1560

lsaIdentity.....identifier of [4] LSAIdentity
  DEFINED in MAP-MS-DataTypes : 1560

lsaIdentityList.....identifier of LSAIdentityList
  DEFINED in MAP-MS-DataTypes : 918

LSAIdentityList.....type reference SEQUENCE OF
  DEFINED in MAP-MS-DataTypes : 920
  USED in MAP-MS-DataTypes : 918

lsaInformation.....identifier of [25] LSAInformation
  DEFINED in MAP-MS-DataTypes : 407

LSAInformation.....type reference SEQUENCE
  DEFINED in MAP-MS-DataTypes : 556
  USED in MAP-MS-DataTypes : 407

lsaInformationWithdraw.....identifier of [12] LSAInformationWithdraw
  DEFINED in MAP-MS-DataTypes : 905

LSAInformationWithdraw.....type reference CHOICE
  DEFINED in MAP-MS-DataTypes : 916
  USED in MAP-MS-DataTypes : 905

LSAOnlyAccessIndicator.....type reference ENUMERATED
  DEFINED in MAP-MS-DataTypes : 539
  USED in MAP-MS-DataTypes : 561

lsaOnlyAccessIndicator.....identifier of [1] LSAOnlyAccessIndicator
  DEFINED in MAP-MS-DataTypes : 561

lsaPriority.....identifier of [1] LSAPriority
  DEFINED in MAP-MS-DataTypes : 550

LSAPriority.....type reference OCTET STRING
  DEFINED in MAP-MS-DataTypes : 569
  USED in MAP-MS-DataTypes : 550

mah.....value reference SS-Code, '00110010'B
  DEFINED in MAP-SS-Code : 68

MAP-BS-Code.....module reference
  DEFINED in MAP-BS-Code : 1
  USED in MAP-MS-DataTypes : 133
  USED in MAP-CommonDataTypes : 65

MAP-CallHandlingOperations.....module reference
  DEFINED in MAP-CallHandlingOperat : 1
  USED in MAP-Protocol : 63

MAP-CH-DataTypes.....module reference
  DEFINED in MAP-CH-DataTypes : 1
  USED in MAP-CallHandlingOperat : 74

MAP-CommonDataTypes.....module reference
  DEFINED in MAP-CommonDataTypes : 1
  USED in MAP-MobileServiceOpera : 160
  USED in MAP-OperationAndMainte : 44
  USED in MAP-MS-DataTypes : 164
  USED in MAP-OM-DataTypes : 23
  USED in MAP-CH-DataTypes : 74
  USED in MAP-SS-DataTypes : 53
  USED in MAP-SM-DataTypes : 36
  USED in MAP-GR-DataTypes : 27
  USED in MAP-LCS-DataTypes : 35
  USED in MAP-ER-DataTypes : 75

MAP-Errors.....module reference
  DEFINED in MAP-Errors : 1
  USED in MAP-Protocol : 170
  USED in MAP-MobileServiceOpera : 107
  USED in MAP-OperationAndMainte : 30
  USED in MAP-CallHandlingOperat : 51
  USED in MAP-SupplementaryServi : 56
  USED in MAP-ShortMessageServic : 41
    
```

USED in MAP-Group-Call-Operati : 27
USED in MAP-LocationServiceOpe : 37

MAP-ER-DataTypes.....	module reference			
DEFINED in MAP-ER-DataTypes	:	1		
USED in MAP-Errors	:	154		
USED in MAP-MS-DataTypes	:	174		
USED in MAP-SM-DataTypes	:	41		
MAP-EXTENSION.....	information object class reference CLASS			
DEFINED in MAP-ExtensionDataTypes	:	22		
USED in MAP-ExtensionDataTypes	:	41	43	48
MAP-ExtensionDataTypes.....	module reference			
DEFINED in MAP-ExtensionDataTypes	:	1		
USED in MAP-MS-DataTypes	:	169		
USED in MAP-CommonDataTypes	:	70		
USED in MAP-OM-DataTypes	:	28		
USED in MAP-CH-DataTypes	:	79		
USED in MAP-SS-DataTypes	:	58		
USED in MAP-SM-DataTypes	:	46		
USED in MAP-GR-DataTypes	:	43		
USED in MAP-LCS-DataTypes	:	40		
USED in MAP-ER-DataTypes	:	85		
MAP-Group-Call-Operations.....	module reference			
DEFINED in MAP-Group-Call-Operati	:	1		
USED in MAP-Protocol	:	102		
MAP-GR-DataTypes.....	module reference			
DEFINED in MAP-GR-DataTypes	:	1		
USED in MAP-Group-Call-Operati	:	37		
MAP-LCS-DataTypes.....	module reference			
DEFINED in MAP-LCS-DataTypes	:	1		
USED in MAP-LocationServiceOpe	:	47		
MAP-LocationServiceOperations.....	module reference			
DEFINED in MAP-LocationServiceOpe	:	1		
USED in MAP-Protocol	:	110		
MAP-MobileServiceOperations.....	module reference			
DEFINED in MAP-MobileServiceOpera	:	1		
USED in MAP-Protocol	:	40		
MAP-MS-DataTypes.....	module reference			
DEFINED in MAP-MS-DataTypes	:	1		
USED in MAP-MobileServiceOpera	:	154		
USED in MAP-CH-DataTypes	:	52		
USED in MAP-GR-DataTypes	:	37		
MAP-OM-DataTypes.....	module reference			
DEFINED in MAP-OM-DataTypes	:	1		
USED in MAP-OperationAndMainte	:	38		
MAP-OperationAndMaintenanceOperations...	module reference			
DEFINED in MAP-OperationAndMainte	:	1		
USED in MAP-Protocol	:	48		
MAP-Protocol.....	module reference			
DEFINED in MAP-Protocol	:	1		
MAP-ShortMessageServiceOperations.....	module reference			
DEFINED in MAP-ShortMessageServic	:	1		
USED in MAP-Protocol	:	93		
MAP-SM-DataTypes.....	module reference			
DEFINED in MAP-SM-DataTypes	:	1		
USED in MAP-ShortMessageServic	:	57		
MAP-SS-Code.....	module reference			
DEFINED in MAP-SS-Code	:	1		
USED in MAP-SupplementaryServi	:	79		
USED in MAP-MS-DataTypes	:	128		
USED in MAP-SS-DataTypes	:	63		
USED in MAP-ER-DataTypes	:	80		
MAP-SS-DataTypes.....	module reference			
DEFINED in MAP-SS-DataTypes	:	1		
USED in MAP-SupplementaryServi	:	74		
USED in MAP-Errors	:	102		
USED in MAP-MS-DataTypes	:	123		

USED in MAP-CH-DataTypes	:	59
USED in MAP-LCS-DataTypes	:	46

51

TAG R4.21 Cross Reference Listing for MAP-Protocol 00-01-06 07:32:40 PAGE

USED in MAP-ER-DataTypes : 68

MAP-SupplementaryServiceOperations.....module reference
 DEFINED in MAP-SupplementaryServi : 1
 USED in MAP-Protocol : 81

MAP-TS-Code.....module reference
 DEFINED in MAP-TS-Code : 1
 USED in MAP-MS-DataTypes : 138
 USED in MAP-CommonDataTypes : 59
 USED in MAP-GR-DataTypes : 32

matchType.....identifier of [0] MatchType
 DEFINED in MAP-MS-DataTypes : 1058

MatchType.....type reference ENUMERATED
 DEFINED in MAP-MS-DataTypes : 1088
 USED in MAP-MS-DataTypes : 1058

maxAddressLength.....value reference INTEGER, 20
 DEFINED in MAP-CommonDataTypes : 129
 USED in MAP-CommonDataTypes : 88

maxEventSpecification.....value reference INTEGER, 2
 DEFINED in MAP-SS-DataTypes : 279
 USED in MAP-SS-DataTypes : 276

maxExt-GeographicalInformation.....value reference INTEGER, 20
 DEFINED in MAP-LCS-DataTypes : 212
 USED in MAP-LCS-DataTypes : 170

maximumentitledPriority.....identifier of EMLPP-Priority
 DEFINED in MAP-CommonDataTypes : 388

maximumEntitledPriority.....identifier of [0] EMLPP-Priority
 DEFINED in MAP-SS-DataTypes : 186

maxISDN-AddressLength.....value reference INTEGER, 9
 DEFINED in MAP-CommonDataTypes : 135
 USED in MAP-CommonDataTypes : 18 132

maxISDN-SubaddressLength.....value reference INTEGER, 21
 DEFINED in MAP-CommonDataTypes : 175
 USED in MAP-CommonDataTypes : 138

maxNameStringLength.....value reference INTEGER, 63
 DEFINED in MAP-LCS-DataTypes : 130
 USED in MAP-LCS-DataTypes : 128

maxNumOfBasicServiceGroups.....value reference INTEGER, 13
 DEFINED in MAP-SS-DataTypes : 256
 USED in MAP-SS-DataTypes : 91 147 253

maxNumOfBasicServices.....value reference INTEGER, 70
 DEFINED in MAP-MS-DataTypes : 926
 USED in MAP-MS-DataTypes : 923

maxNumOfBearerServices.....value reference INTEGER, 50
 DEFINED in MAP-MS-DataTypes : 602
 USED in MAP-MS-DataTypes : 599

maxNumOfCamelBasicServiceCriteria.....value reference INTEGER, 5
 DEFINED in MAP-MS-DataTypes : 1082
 USED in MAP-MS-DataTypes : 1073

maxNumOfCamelDestinationNumberLengths...value reference INTEGER, 3
 DEFINED in MAP-MS-DataTypes : 1080
 USED in MAP-MS-DataTypes : 1070

maxNumOfCamelDestinationNumbers.....value reference INTEGER, 10
 DEFINED in MAP-MS-DataTypes : 1078
 USED in MAP-MS-DataTypes : 1065

maxNumOfCamelSSEvents.....value reference INTEGER, 10
 DEFINED in MAP-MS-DataTypes : 991
 USED in MAP-MS-DataTypes : 983

maxNumOfCamelTDPData.....value reference INTEGER, 10
 DEFINED in MAP-MS-DataTypes : 1011
 USED in MAP-MS-DataTypes : 62 484 1004 1036 1039 1138 1208

maxNumOfCAMEL-O-CauseValueCriteria.....value reference INTEGER, 5

52

TAG	R4.21	Cross Reference Listing for MAP-Protocol	00-01-06	07:32:40	PAGE
	DEFINED in MAP-MS-DataTypes	: 1098			
	USED in MAP-MS-DataTypes	: 1092			
	maxNumOfCAMEL-T-CauseValueCriteria.....value	reference INTEGER, 5			
	DEFINED in MAP-MS-DataTypes	: 1100			
	USED in MAP-MS-DataTypes	: 1095			
	maxNumOfCCBS-Requests.....value	reference INTEGER, 5			
	DEFINED in MAP-SS-DataTypes	: 193			
	USED in MAP-SS-DataTypes	: 190 202			
	maxNumOfCUG.....value	reference INTEGER, 10			
	DEFINED in MAP-MS-DataTypes	: 774			
	USED in MAP-MS-DataTypes	: 753			
	maxNumOfDP-AnalysedInfoCriteria.....value	reference INTEGER, 10			
	DEFINED in MAP-MS-DataTypes	: 957			
	USED in MAP-MS-DataTypes	: 954			
	maxNumOfExternalClient.....value	reference INTEGER, 5			
	DEFINED in MAP-MS-DataTypes	: 832			
	USED in MAP-MS-DataTypes	: 829			
	maxNumOfExt-BasicServiceGroups.....value	reference INTEGER, 32			
	DEFINED in MAP-MS-DataTypes	: 782			
	USED in MAP-MS-DataTypes	: 659 738 776 779			
	maxNumOfGMLC.....value	reference INTEGER, 5			
	DEFINED in MAP-MS-DataTypes	: 428			
	USED in MAP-MS-DataTypes	: 424			
	maxNumOfHLR-Id.....value	reference INTEGER, 50			
	DEFINED in MAP-CommonDataTypes	: 287			
	USED in MAP-CommonDataTypes	: 284			
	maxNumOfISDN-AddressDigits.....value	reference INTEGER, 15			
	DEFINED in MAP-MS-DataTypes	: 1076			
	USED in MAP-MS-DataTypes	: 1071			
	maxNumOfLSAs.....value	reference INTEGER, 20			
	DEFINED in MAP-MS-DataTypes	: 546			
	USED in MAP-MS-DataTypes	: 543 920			
	maxNumOfMobilityTriggers.....value	reference INTEGER, 10			
	DEFINED in MAP-MS-DataTypes	: 1182			
	USED in MAP-MS-DataTypes	: 1179			
	maxNumOfMOLR-Class.....value	reference INTEGER, 3			
	DEFINED in MAP-MS-DataTypes	: 857			
	USED in MAP-MS-DataTypes	: 854			
	maxNumOfPDP-Contexts.....value	reference INTEGER, 50			
	DEFINED in MAP-MS-DataTypes	: 442			
	USED in MAP-MS-DataTypes	: 439 456 913			
	maxNumOfPLMNClient.....value	reference INTEGER, 5			
	DEFINED in MAP-MS-DataTypes	: 837			
	USED in MAP-MS-DataTypes	: 834			
	maxNumOfPrivacyClass.....value	reference INTEGER, 4			
	DEFINED in MAP-MS-DataTypes	: 814			
	USED in MAP-MS-DataTypes	: 811			
	maxNumOfPrivateExtensions.....value	reference INTEGER, 10			
	DEFINED in MAP-ExtensionDataTypes	: 46			
	USED in MAP-ExtensionDataTypes	: 37			
	maxNumOfSS.....value	reference INTEGER, 30			
	DEFINED in MAP-SS-DataTypes	: 248			
	USED in MAP-MS-DataTypes	: 118 642			
	USED in MAP-SS-DataTypes	: 32 245 250			
	maxNumOfTeleservices.....value	reference INTEGER, 20			
	DEFINED in MAP-MS-DataTypes	: 607			
	USED in MAP-MS-DataTypes	: 604			
	maxNumOfVBSGroupIds.....value	reference INTEGER, 50			
	DEFINED in MAP-MS-DataTypes	: 1307			
	USED in MAP-MS-DataTypes	: 1301			

maxNumOfVCSCGroupIds.....value reference INTEGER, 50
DEFINED in MAP-MS-DataTypes : 1309

53

TAG R4.21 Cross Reference Listing for MAP-Protocol 00-01-06 07:32:40 PAGE

USED in MAP-MS-DataTypes : 1304

maxNumOfZoneCodes.....value reference INTEGER, 10
 DEFINED in MAP-MS-DataTypes : 871
 USED in MAP-MS-DataTypes : 51 865

maxSignalInfoLength.....value reference INTEGER, 200
 DEFINED in MAP-CommonDataTypes : 188
 USED in MAP-CommonDataTypes : 23 186

maxUSSD-StringLength.....value reference INTEGER, 160
 DEFINED in MAP-SS-DataTypes : 231
 USED in MAP-SS-DataTypes : 227

mcef-Set.....identifier of Named Number, 2
 DEFINED in MAP-SM-DataTypes : 188

mci.....value reference SS-Code, '00010101'B
 DEFINED in MAP-SS-Code : 36

memoryAvailable.....identifier of Named Number, 1
 DEFINED in MAP-SM-DataTypes : 209

memoryCapacityExceeded.....identifier of Named Number, 0
 DEFINED in MAP-SM-DataTypes : 164

memoryCapacityExceeded.....identifier of Named Number, 0
 DEFINED in MAP-ER-DataTypes : 140

MessageType.....type reference CHOICE
 DEFINED in TCAPMessages : 51
 USED in TCAPMessages : 47

messageWaitingListFull.....value reference MessageWaitingListFull, CHOICE VALUE
 DEFINED in MAP-Protocol : 413

MessageWaitingListFull.....type reference ERROR
 DEFINED in MAP-Errors : 402
 USED in MAP-Protocol : 154 413
 USED in MAP-ShortMessageServic : 39 121
 USED in MAP-Errors : 78

messageWaitListFullParam.....identifier of MessageWaitListFullParam
 DEFINED in MAP-Errors : 404

MessageWaitListFullParam.....type reference SEQUENCE
 DEFINED in MAP-ER-DataTypes : 312
 USED in MAP-Errors : 133 404
 USED in MAP-ER-DataTypes : 41

mistypedComponent.....identifier of Named Number, 1
 DEFINED in TCAPMessages : 180

mistypedParameter.....identifier of Named Number, 2
 DEFINED in TCAPMessages : 185

mistypedParameter.....identifier of Named Number, 2
 DEFINED in TCAPMessages : 194

mistypedParameter.....identifier of Named Number, 4
 DEFINED in TCAPMessages : 200

mlcNumber.....identifier of [0] ISDN-AddressString
 DEFINED in MAP-LCS-DataTypes : 53

mlc-Number.....identifier of ISDN-AddressString
 DEFINED in MAP-LCS-DataTypes : 72

MM-Code.....type reference OCTET STRING
 DEFINED in MAP-MS-DataTypes : 1184
 USED in MAP-MS-DataTypes : 1180 1556

mm-EventNotSupported.....value reference MM-EventNotSupported, CHOICE VALUE
 DEFINED in MAP-Protocol : 425

MM-EventNotSupported.....type reference ERROR
 DEFINED in MAP-Errors : 441
 USED in MAP-Protocol : 168 425
 USED in MAP-MobileServiceOpera : 93 457
 USED in MAP-Errors : 91

mm-EventNotSupported-Param.....identifier of MM-EventNotSupported-Param

54

TAG	R4.21	Cross Reference Listing for MAP-Protocol	00-01-06	07:32:40	PAGE
	DEFINED in MAP-Errors	:	443		
MM-EventNotSupported-Param	type reference SEQUENCE			
	DEFINED in MAP-ER-DataTypes	:	374		
	USED in MAP-Errors	:	144	443	
	USED in MAP-ER-DataTypes	:	53		
mnrf-Set	identifier of Named Number, 1			
	DEFINED in MAP-SM-DataTypes	:	187		
mnrg-Set	identifier of Named Number, 3			
	DEFINED in MAP-SM-DataTypes	:	189		
mobileNotReachableReason	identifier of [2] AbsentSubscriberDiagnosticSM			
	DEFINED in MAP-MS-DataTypes	:	1247		
mobilityTriggers	identifier of MobilityTriggers			
	DEFINED in MAP-MS-DataTypes	:	1169		
MobilityTriggers	type reference SEQUENCE OF			
	DEFINED in MAP-MS-DataTypes	:	1179		
	USED in MAP-MS-DataTypes	:	1169		
ModificationInstruction	type reference ENUMERATED			
	DEFINED in MAP-MS-DataTypes	:	1525		
	USED in MAP-MS-DataTypes	:	1514	1520	1521
modificationRequestFor-CSI	identifier of [3] ModificationRequestFor-CSI			
	DEFINED in MAP-MS-DataTypes	:	1497		
ModificationRequestFor-CSI	type reference SEQUENCE			
	DEFINED in MAP-MS-DataTypes	:	1518		
	USED in MAP-MS-DataTypes	:	1497		
modificationRequestFor-SS-Info	identifier of [2] ModificationRequestFor-SS-Info			
	DEFINED in MAP-MS-DataTypes	:	1496		
ModificationRequestFor-SS-Info	type reference SEQUENCE			
	DEFINED in MAP-MS-DataTypes	:	1507		
	USED in MAP-MS-DataTypes	:	1496		
modifyCSI-State	identifier of [2] ModificationInstruction			
	DEFINED in MAP-MS-DataTypes	:	1521		
modifyNotificationToCSE	identifier of [6] ModificationInstruction			
	DEFINED in MAP-MS-DataTypes	:	1514		
modifyNotificationToCSE	identifier of [1] ModificationInstruction			
	DEFINED in MAP-MS-DataTypes	:	1520		
MOLR-Class	type reference SEQUENCE			
	DEFINED in MAP-MS-DataTypes	:	859		
	USED in MAP-MS-DataTypes	:	855		
molr-List	identifier of [2] MOLR-List			
	DEFINED in MAP-MS-DataTypes	:	421		
MOLR-List	type reference SEQUENCE OF			
	DEFINED in MAP-MS-DataTypes	:	854		
	USED in MAP-MS-DataTypes	:	421		
monitoringMode	identifier of [0] MonitoringMode			
	DEFINED in MAP-CH-DataTypes	:	355		
MonitoringMode	type reference ENUMERATED			
	DEFINED in MAP-CH-DataTypes	:	360		
	USED in MAP-CH-DataTypes	:	355		
moreMessagesToSend	identifier of NULL			
	DEFINED in MAP-SM-DataTypes	:	123		
mo-forwardSM	value reference MO-ForwardSM, CHOICE VALUE			
	DEFINED in MAP-Protocol	:	259		
MO-ForwardSM	type reference OPERATION			
	DEFINED in MAP-ShortMessageServic	:	81		
	USED in MAP-Protocol	:	87	259	
	USED in MAP-ShortMessageServic	:	14		
mo-forwardSM-Arg	identifier of MO-ForwardSM-Arg			

DEFINED in MAP-ShortMessageServic : 83

55

TAG R4.21 Cross Reference Listing for MAP-Protocol 00-01-06 07:32:40 PAGE

MO-ForwardSM-Arg.....type reference SEQUENCE
 DEFINED in MAP-SM-DataTypes : 106
 USED in MAP-ShortMessageServic : 47 83
 USED in MAP-SM-DataTypes : 16

mo-forwardSM-Res.....identifier of MO-ForwardSM-Res
 DEFINED in MAP-ShortMessageServic : 85

MO-ForwardSM-Res.....type reference SEQUENCE
 DEFINED in MAP-SM-DataTypes : 114
 USED in MAP-ShortMessageServic : 48 85
 USED in MAP-SM-DataTypes : 17

mo-lr.....identifier of Named Number, 2
 DEFINED in MAP-LCS-DataTypes : 235

msc-Number.....identifier of [1] ISDN-AddressString
 DEFINED in MAP-MS-DataTypes : 187

msc-Number.....identifier of [6] ISDN-AddressString
 DEFINED in MAP-MS-DataTypes : 1361

msc-Number.....identifier of [1] ISDN-AddressString
 DEFINED in MAP-CH-DataTypes : 197

msc-Number.....identifier of [0] ISDN-AddressString
 DEFINED in MAP-SM-DataTypes : 98

msc-Number.....identifier of ISDN-AddressString
 DEFINED in MAP-LCS-DataTypes : 65

msisdn.....identifier of ISDN-AddressString
 DEFINED in MAP-OperationAndMainte : 79

msisdn.....identifier of [1] ISDN-AddressString
 DEFINED in MAP-MS-DataTypes : 574

msisdn.....identifier of ISDN-AddressString
 DEFINED in MAP-MS-DataTypes : 1533

msisdn.....identifier of [2] ISDN-AddressString
 DEFINED in MAP-MS-DataTypes : 1558

msisdn.....identifier of [1] ISDN-AddressString
 DEFINED in MAP-CommonDataTypes : 328

msisdn.....identifier of [0] ISDN-AddressString
 DEFINED in MAP-CH-DataTypes : 94

msisdn.....identifier of [12] ISDN-AddressString
 DEFINED in MAP-CH-DataTypes : 161

msisdn.....identifier of [2] ISDN-AddressString
 DEFINED in MAP-CH-DataTypes : 198

msisdn.....identifier of [9] ISDN-AddressString
 DEFINED in MAP-CH-DataTypes : 229

msisdn.....identifier of [0] ISDN-AddressString
 DEFINED in MAP-SS-DataTypes : 215

msisdn.....identifier of [1] ISDN-AddressString
 DEFINED in MAP-SS-DataTypes : 260

msisdn.....identifier of [0] ISDN-AddressString
 DEFINED in MAP-SM-DataTypes : 53

msisdn.....identifier of [2] ISDN-AddressString
 DEFINED in MAP-SM-DataTypes : 139

msisdn.....identifier of ISDN-AddressString
 DEFINED in MAP-SM-DataTypes : 144

msisdn.....identifier of ISDN-AddressString
 DEFINED in MAP-SM-DataTypes : 175

msisdn.....identifier of [3] ISDN-AddressString
 DEFINED in MAP-LCS-DataTypes : 76

msisdn.....identifier of [0] ISDN-AddressString

DEFINED in MAP-LCS-DataTypes : 220

56

TAG R4.21 Cross Reference Listing for MAP-Protocol 00-01-06 07:32:40 PAGE

msNotReachable.....identifier of NULL
 DEFINED in MAP-MS-DataTypes : 1295

msPurged.....identifier of Named Number, 0
 DEFINED in MAP-MS-DataTypes : 1396

ms-Present.....identifier of Named Number, 0
 DEFINED in MAP-SM-DataTypes : 208

mt-forwardSM.....value reference MT-ForwardSM, CHOICE VALUE
 DEFINED in MAP-Protocol : 260

MT-ForwardSM.....type reference OPERATION
 DEFINED in MAP-ShortMessageServic : 93
 USED in MAP-Protocol : 88 260
 USED in MAP-ShortMessageServic : 15

mt-forwardSM-Arg.....identifier of MT-ForwardSM-Arg
 DEFINED in MAP-ShortMessageServic : 95

MT-ForwardSM-Arg.....type reference SEQUENCE
 DEFINED in MAP-SM-DataTypes : 119
 USED in MAP-ShortMessageServic : 49 95
 USED in MAP-SM-DataTypes : 18

mt-forwardSM-Res.....identifier of MT-ForwardSM-Res
 DEFINED in MAP-ShortMessageServic : 97

MT-ForwardSM-Res.....type reference SEQUENCE
 DEFINED in MAP-SM-DataTypes : 127
 USED in MAP-ShortMessageServic : 50 97
 USED in MAP-SM-DataTypes : 19

multipleECT-Barred.....identifier of Named Number, 14
 DEFINED in MAP-MS-DataTypes : 630

multiPLY.....value reference SS-Code, '01010001'B
 DEFINED in MAP-SS-Code : 88

mw-Status.....identifier of MW-Status
 DEFINED in MAP-SM-DataTypes : 181

MW-Status.....type reference BIT STRING
 DEFINED in MAP-SM-DataTypes : 185
 USED in MAP-SM-DataTypes : 181

m-CSI.....identifier of [5] M-CSI
 DEFINED in MAP-MS-DataTypes : 941

M-CSI.....type reference SEQUENCE
 DEFINED in MAP-MS-DataTypes : 1168
 USED in MAP-MS-DataTypes : 941 1489

m-CSI.....identifier of Named Number, 7
 DEFINED in MAP-MS-DataTypes : 1452

m-CSI.....identifier of [11] M-CSI
 DEFINED in MAP-MS-DataTypes : 1489

NAEA-CIC.....type reference OCTET STRING
 DEFINED in MAP-CommonDataTypes : 319
 USED in MAP-CommonDataTypes : 37 315

naea-PreferredCI.....identifier of [15] NAEA-PreferredCI
 DEFINED in MAP-MS-DataTypes : 401

NAEA-PreferredCI.....type reference SEQUENCE
 DEFINED in MAP-CommonDataTypes : 314
 USED in MAP-MS-DataTypes : 155 401
 USED in MAP-CommonDataTypes : 36
 USED in MAP-CH-DataTypes : 71 158

naea-PreferredCI.....identifier of [10] NAEA-PreferredCI
 DEFINED in MAP-CH-DataTypes : 158

naea-PreferredCIC.....identifier of [0] NAEA-CIC
 DEFINED in MAP-CommonDataTypes : 315

nameString.....identifier of [2] NameString
 DEFINED in MAP-LCS-DataTypes : 120

NameString.....type reference USSD-String

57	TAG	R4.21	Cross Reference Listing for MAP-Protocol	00-01-06	07:32:40	PAGE
			DEFINED in MAP-LCS-DataTypes	:	128	
			USED in MAP-LCS-DataTypes	:	120	
			na-ESRD.....	identifier of [3] ISDN-AddressString		
			DEFINED in MAP-LCS-DataTypes	:	223	
			na-ESRK.....	identifier of [4] ISDN-AddressString		
			DEFINED in MAP-LCS-DataTypes	:	224	
			negativePW-Check.....	value reference NegativePW-Check, CHOICE VALUE		
			DEFINED in MAP-Protocol	:	402	
			NegativePW-Check.....	type reference ERROR		
			DEFINED in MAP-Errors	:	376	
			USED in MAP-Protocol	:	150 402	
			USED in MAP-SupplementaryServi	:	46 138 157 229	
			USED in MAP-Errors	:	70	
			netDetNotReachable.....	identifier of NotReachableReason		
			DEFINED in MAP-MS-DataTypes	:	1392	
			networkAccessMode.....	identifier of [24] NetworkAccessMode		
			DEFINED in MAP-MS-DataTypes	:	406	
			NetworkAccessMode.....	type reference ENUMERATED		
			DEFINED in MAP-MS-DataTypes	:	431	
			USED in MAP-MS-DataTypes	:	406	
			networkNode-AreaRestricted.....	identifier of Named Number, 0		
			DEFINED in MAP-MS-DataTypes	:	885	
			networkNode-Number.....	identifier of [1] ISDN-AddressString		
			DEFINED in MAP-SM-DataTypes	:	86	
			NetworkResource.....	type reference ENUMERATED		
			DEFINED in MAP-CommonDataTypes	:	304	
			USED in MAP-CommonDataTypes	:	35	
			USED in MAP-ER-DataTypes	:	74 169 176	
			networkResource.....	identifier of NetworkResource		
			DEFINED in MAP-ER-DataTypes	:	169	
			networkResource.....	identifier of NetworkResource		
			DEFINED in MAP-ER-DataTypes	:	176	
			networkSignalInfo.....	identifier of [10] ExternalSignalInfo		
			DEFINED in MAP-CH-DataTypes	:	104	
			networkSignalInfo.....	identifier of [6] ExternalSignalInfo		
			DEFINED in MAP-CH-DataTypes	:	201	
			networkSignalInfo.....	identifier of [4] ExternalSignalInfo		
			DEFINED in MAP-SS-DataTypes	:	291	
			newPassword.....	identifier of Password		
			DEFINED in MAP-SupplementaryServi	:	221	
			newPasswordsMismatch.....	identifier of Named Number, 2		
			DEFINED in MAP-ER-DataTypes	:	136	
			noAdditionalInformation.....	identifier of Named Number, 0		
			DEFINED in MAP-ER-DataTypes	:	344	
			noCUG-Restrictions.....	identifier of Named Number, 0		
			DEFINED in MAP-MS-DataTypes	:	770	
			noGroupCallNbParam.....	identifier of NoGroupCallNbParam		
			DEFINED in MAP-Errors	:	416	
			NoGroupCallNbParam.....	type reference SEQUENCE		
			DEFINED in MAP-ER-DataTypes	:	320	
			USED in MAP-Errors	:	136 416	
			USED in MAP-ER-DataTypes	:	45	
VALUE			noGroupCallNumberAvailable.....	value reference NoGroupCallNumberAvailable, CHOICE		
			DEFINED in MAP-Protocol	:	389	
			NoGroupCallNumberAvailable.....	type reference ERROR		
			DEFINED in MAP-Errors	:	414	

USED in MAP-Protocol	:	157	389
USED in MAP-Group-Call-Operati	:	26	53
USED in MAP-Errors	:	82	

58 TAG R4.21 Cross Reference Listing for MAP-Protocol 00-01-06 07:32:40 PAGE

noHandoverNumberAvailable.....value reference NoHandoverNumberAvailable, CHOICE
 VALUE
 DEFINED in MAP-Protocol : 355

NoHandoverNumberAvailable.....type reference ERROR
 DEFINED in MAP-Errors : 252
 USED in MAP-Protocol : 129 355
 USED in MAP-MobileServiceOpera : 90 310
 USED in MAP-Errors : 36

noPageResponse.....identifier of Named Number, 2
 DEFINED in MAP-ER-DataTypes : 249

noReply.....identifier of Named Number, 2
 DEFINED in MAP-CH-DataTypes : 130

noReplyConditionTime.....identifier of [7] Ext-NoRepCondTime
 DEFINED in MAP-MS-DataTypes : 671

noReplyConditionTime.....identifier of [5] Ext-NoRepCondTime
 DEFINED in MAP-MS-DataTypes : 1513

noReplyConditionTime.....identifier of [5] NoReplyConditionTime
 DEFINED in MAP-SS-DataTypes : 74

NoReplyConditionTime.....type reference INTEGER
 DEFINED in MAP-SS-DataTypes : 78
 USED in MAP-SS-DataTypes : 30 74 100

noReplyConditionTime.....identifier of [7] NoReplyConditionTime
 DEFINED in MAP-SS-DataTypes : 100

noResponseFromBusyMS.....identifier of Named Number, 3
 DEFINED in MAP-CH-DataTypes : 401

noResponseFromFreeMS.....identifier of Named Number, 2
 DEFINED in MAP-CH-DataTypes : 400

noRoamingNbParam.....identifier of NoRoamingNbParam
 DEFINED in MAP-Errors : 269

NoRoamingNbParam.....type reference SEQUENCE
 DEFINED in MAP-ER-DataTypes : 237
 USED in MAP-Errors : 122 269
 USED in MAP-ER-DataTypes : 33

noRoamingNumberAvailable.....value reference NoRoamingNumberAvailable, CHOICE
 VALUE
 DEFINED in MAP-Protocol : 368

NoRoamingNumberAvailable.....type reference ERROR
 DEFINED in MAP-Errors : 267
 USED in MAP-Protocol : 133 368
 USED in MAP-CallHandlingOperat : 39 116
 USED in MAP-Errors : 44

noSM-RP-DA.....identifier of [5] NULL
 DEFINED in MAP-SM-DataTypes : 136

noSM-RP-OA.....identifier of [5] NULL
 DEFINED in MAP-SM-DataTypes : 141

noSubscriberReply.....value reference NoSubscriberReply, CHOICE VALUE
 DEFINED in MAP-Protocol : 371

NoSubscriberReply.....type reference ERROR
 DEFINED in MAP-Errors : 284
 USED in MAP-Protocol : 136 371
 USED in MAP-CallHandlingOperat : 42 98
 USED in MAP-Errors : 46

noSubscriberReplyParam.....identifier of NoSubscriberReplyParam
 DEFINED in MAP-Errors : 286

NoSubscriberReplyParam.....type reference SEQUENCE
 DEFINED in MAP-ER-DataTypes : 265
 USED in MAP-Errors : 126 286
 USED in MAP-ER-DataTypes : 36

noteMM-Event.....value reference NoteMM-Event, CHOICE VALUE
DEFINED in MAP-Protocol : 320

NoteMM-Event.....type reference OPERATION

59 TAG R4.21 Cross Reference Listing for MAP-Protocol 00-01-06 07:32:40 PAGE

DEFINED in MAP-MobileServiceOpera : 448
 USED in MAP-Protocol : 36 320
 USED in MAP-MobileServiceOpera : 69

noteMM-EventArg.....identifier of NoteMM-EventArg
 DEFINED in MAP-MobileServiceOpera : 450

NoteMM-EventArg.....type reference SEQUENCE
 DEFINED in MAP-MS-DataTypes : 1554
 USED in MAP-MobileServiceOpera : 150 450
 USED in MAP-MS-DataTypes : 110

noteMM-EventRes.....identifier of NoteMM-EventRes
 DEFINED in MAP-MobileServiceOpera : 452

NoteMM-EventRes.....type reference SEQUENCE
 DEFINED in MAP-MS-DataTypes : 1565
 USED in MAP-MobileServiceOpera : 151 452
 USED in MAP-MS-DataTypes : 111

noteMsPresentForGprs.....value reference NoteMsPresentForGprs, CHOICE VALUE
 DEFINED in MAP-Protocol : 310

NoteMsPresentForGprs.....type reference OPERATION
 DEFINED in MAP-MobileServiceOpera : 435
 USED in MAP-Protocol : 35 310
 USED in MAP-MobileServiceOpera : 66

noteMsPresentForGprsArg.....identifier of NoteMsPresentForGprsArg
 DEFINED in MAP-MobileServiceOpera : 437

NoteMsPresentForGprsArg.....type reference SEQUENCE
 DEFINED in MAP-MS-DataTypes : 1267
 USED in MAP-MobileServiceOpera : 148 437
 USED in MAP-MS-DataTypes : 106

noteMsPresentForGprsRes.....identifier of NoteMsPresentForGprsRes
 DEFINED in MAP-MobileServiceOpera : 439

NoteMsPresentForGprsRes.....type reference SEQUENCE
 DEFINED in MAP-MS-DataTypes : 1274
 USED in MAP-MobileServiceOpera : 149 439
 USED in MAP-MS-DataTypes : 107

noteSubscriberDataModified.....value reference NoteSubscriberDataModified, CHOICE VALUE
 DEFINED in MAP-Protocol : 281

NoteSubscriberDataModified.....type reference OPERATION
 DEFINED in MAP-MobileServiceOpera : 288
 USED in MAP-Protocol : 37 281
 USED in MAP-MobileServiceOpera : 34

noteSubscriberDataModifiedArg.....identifier of NoteSubscriberDataModifiedArg
 DEFINED in MAP-MobileServiceOpera : 290

NoteSubscriberDataModifiedArg.....type reference SEQUENCE
 DEFINED in MAP-MS-DataTypes : 1531
 USED in MAP-MobileServiceOpera : 140 290
 USED in MAP-MS-DataTypes : 94

noteSubscriberDataModifiedRes.....identifier of NoteSubscriberDataModifiedRes
 DEFINED in MAP-MobileServiceOpera : 292

NoteSubscriberDataModifiedRes.....type reference SEQUENCE
 DEFINED in MAP-MS-DataTypes : 1538
 USED in MAP-MobileServiceOpera : 141 292
 USED in MAP-MS-DataTypes : 95

notForwarded.....identifier of Named Number, 1
 DEFINED in MAP-MS-DataTypes : 1086

notification.....identifier of Named Number, 0
 DEFINED in MAP-MS-DataTypes : 851

notificationToCSE.....identifier of [3] NULL
 DEFINED in MAP-MS-DataTypes : 478

notificationToCSE.....identifier of [1] NULL
 DEFINED in MAP-MS-DataTypes : 977

notificationToCSE.....identifier of [1] NULL
DEFINED in MAP-MS-DataTypes : 998

60

TAG R4.21 Cross Reference Listing for MAP-Protocol 00-01-06 07:32:40 PAGE

notificationToCSE.....identifier of [3] NULL
 DEFINED in MAP-MS-DataTypes : 1132

notificationToCSE.....identifier of [2] NULL
 DEFINED in MAP-MS-DataTypes : 1173

notificationToCSE.....identifier of [1] NULL
 DEFINED in MAP-MS-DataTypes : 1202

notificationToCSE.....identifier of NULL
 DEFINED in MAP-MS-DataTypes : 1457

notificationToCSE.....identifier of NULL
 DEFINED in MAP-MS-DataTypes : 1465

notificationToCSE.....identifier of NULL
 DEFINED in MAP-MS-DataTypes : 1473

notificationToCSE.....identifier of [2] NULL
 DEFINED in MAP-MS-DataTypes : 1577

notificationToCSE.....identifier of [4] NULL
 DEFINED in MAP-MS-DataTypes : 1586

notificationToMSUser.....identifier of [1] NotificationToMSUser
 DEFINED in MAP-MS-DataTypes : 842

NotificationToMSUser.....type reference ENUMERATED
 DEFINED in MAP-MS-DataTypes : 850
 USED in MAP-MS-DataTypes : 842

notificationWithPrivacyVerification.....identifier of Named Number, 1
 DEFINED in MAP-MS-DataTypes : 852

notKnownToBePorted.....identifier of Named Number, 0
 DEFINED in MAP-CH-DataTypes : 167

notProvidedFromVLR.....identifier of [2] NULL
 DEFINED in MAP-MS-DataTypes : 1393

notReachable.....identifier of Named Number, 0
 DEFINED in MAP-CH-DataTypes : 128

NotReachableReason.....type reference ENUMERATED
 DEFINED in MAP-MS-DataTypes : 1395
 USED in MAP-MS-DataTypes : 1392

notRegistered.....identifier of Named Number, 3
 DEFINED in MAP-MS-DataTypes : 1399

not-derivable.....identifier of NULL
 DEFINED in TCAPMessages : 168

numberChanged.....value reference NumberChanged, CHOICE VALUE
 DEFINED in MAP-Protocol : 336

NumberChanged.....type reference ERROR
 DEFINED in MAP-Errors : 203
 USED in MAP-Protocol : 120 336
 USED in MAP-CallHandlingOperat : 36 93
 USED in MAP-Errors : 23

numberChangedParam.....identifier of NumberChangedParam
 DEFINED in MAP-Errors : 205

NumberChangedParam.....type reference SEQUENCE
 DEFINED in MAP-ER-DataTypes : 209
 USED in MAP-Errors : 114 205
 USED in MAP-ER-DataTypes : 26

NumberOfForwarding.....type reference INTEGER
 DEFINED in MAP-CH-DataTypes : 91
 USED in MAP-CH-DataTypes : 20 96

numberOfForwarding.....identifier of [2] NumberOfForwarding
 DEFINED in MAP-CH-DataTypes : 96

numberOfPW-AttemptsViolation.....value reference NumberOfPW-AttemptsViolation, CHOICE
 VALUE

DEFINED in MAP-Protocol : 403
NumberOfPW-AttemptsViolation.....type reference ERROR

61	TAG	R4.21	Cross Reference Listing for MAP-Protocol	00-01-06 07:32:40	PAGE
			DEFINED in MAP-Errors	:	378
			USED in MAP-Protocol	:	151 403
			USED in MAP-SupplementaryServi	:	47 139 158 230
			USED in MAP-Errors	:	71
			numberOfRequestedVectors.....	identifier of	NumberOfRequestedVectors
			DEFINED in MAP-MS-DataTypes	:	256
			numberOfRequestedVectors.....	identifier of	NumberOfRequestedVectors
			DEFINED in MAP-MS-DataTypes	:	361
			NumberOfRequestedVectors.....	type reference	INTEGER
			DEFINED in MAP-MS-DataTypes	:	372
			USED in MAP-MS-DataTypes	:	256 361
			numberPortabilityStatus.....	identifier of	[13] NumberPortabilityStatus
			DEFINED in MAP-CH-DataTypes	:	162
			NumberPortabilityStatus.....	type reference	ENUMERATED
			DEFINED in MAP-CH-DataTypes	:	166
			USED in MAP-CH-DataTypes	:	162
			odb.....	identifier of	[2] NULL
			DEFINED in MAP-MS-DataTypes	:	1437
			odb-Data.....	identifier of	[8] ODB-Data
			DEFINED in MAP-MS-DataTypes	:	584
			ODB-Data.....	type reference	SEQUENCE
			DEFINED in MAP-MS-DataTypes	:	609
			USED in MAP-MS-DataTypes	:	48 584 1472
			odb-Data.....	identifier of	ODB-Data
			DEFINED in MAP-MS-DataTypes	:	1472
			odb-GeneralData.....	identifier of	ODB-GeneralData
			DEFINED in MAP-MS-DataTypes	:	610
			ODB-GeneralData.....	type reference	BIT STRING
			DEFINED in MAP-MS-DataTypes	:	615
			USED in MAP-MS-DataTypes	:	610 877
			odb-GeneralData.....	identifier of	[4] ODB-GeneralData
			DEFINED in MAP-MS-DataTypes	:	877
			odb-HPLMN-Data.....	identifier of	ODB-HPLMN-Data
			DEFINED in MAP-MS-DataTypes	:	611
			ODB-HPLMN-Data.....	type reference	BIT STRING
			DEFINED in MAP-MS-DataTypes	:	634
			USED in MAP-MS-DataTypes	:	611
			odb-Info.....	identifier of	[3] ODB-Info
			DEFINED in MAP-MS-DataTypes	:	1428
			ODB-Info.....	type reference	SEQUENCE
			DEFINED in MAP-MS-DataTypes	:	1471
			USED in MAP-MS-DataTypes	:	1428
			omc-Id.....	identifier of	[3] AddressString
			DEFINED in MAP-OM-DataTypes	:	40
			onlyMSC.....	identifier of	Named Number, 1
			DEFINED in MAP-MS-DataTypes	:	433
			onlySGSN.....	identifier of	Named Number, 2
			DEFINED in MAP-MS-DataTypes	:	434
			operationCode.....	identifier of	OPERATION
			DEFINED in TCAPMessages	:	136
			USED in TCAPMessages	:	137
			operationCode.....	identifier of	OPERATION
			DEFINED in TCAPMessages	:	147
			USED in TCAPMessages	:	148
			operatorBarring.....	identifier of	Named Number, 1
			DEFINED in MAP-ER-DataTypes	:	108
			operatorDeterminedBarring.....	identifier of	Named Number, 1

DEFINED in MAP-MS-DataTypes : 597

62

TAG R4.21 Cross Reference Listing for MAP-Protocol 00-01-06 07:32:40 PAGE

operatorDeterminedBarring.....identifier of Named Number, 3
 DEFINED in MAP-ER-DataTypes : 97

operatorDeterminedBarringData.....identifier of Named Number, 2
 DEFINED in MAP-MS-DataTypes : 1545

OrigTransactionID.....type reference [APPLICATION 8] IMPLICIT
 TransactionID
 DEFINED in TCAPMessages : 97
 USED in TCAPMessages : 61 69

orNotSupportedInGMSC.....identifier of [16] NULL
 DEFINED in MAP-CH-DataTypes : 212

or-Capability.....identifier of [5] OR-Phase
 DEFINED in MAP-CH-DataTypes : 99

or-Interrogation.....identifier of [4] NULL
 DEFINED in MAP-CH-DataTypes : 98

or-Interrogation.....identifier of [10] NULL
 DEFINED in MAP-CH-DataTypes : 205

or-NotAllowed.....value reference OR-NotAllowed, CHOICE VALUE
 DEFINED in MAP-Protocol : 374

OR-NotAllowed.....type reference ERROR
 DEFINED in MAP-Errors : 309
 USED in MAP-Protocol : 132 374
 USED in MAP-CallHandlingOperat : 34 91 114 126
 USED in MAP-Errors : 43

or-NotAllowedParam.....identifier of OR-NotAllowedParam
 DEFINED in MAP-Errors : 311

OR-NotAllowedParam.....type reference SEQUENCE
 DEFINED in MAP-ER-DataTypes : 192
 USED in MAP-Errors : 123 311
 USED in MAP-ER-DataTypes : 24

OR-Phase.....type reference INTEGER
 DEFINED in MAP-CH-DataTypes : 123
 USED in MAP-CH-DataTypes : 99

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 : 166

OverrideCategory.....type reference ENUMERATED
 DEFINED in MAP-SS-DataTypes : 173
 USED in MAP-SS-DataTypes : 28 166

overrideDisabled.....identifier of Named Number, 1
 DEFINED in MAP-SS-DataTypes : 175

overrideEnabled.....identifier of Named Number, 0
 DEFINED in MAP-SS-DataTypes : 174

ownNumberPortedOut.....identifier of Named Number, 1
 DEFINED in MAP-CH-DataTypes : 168

o-andM-HPLMN.....identifier of Named Number, 1
 DEFINED in MAP-CommonDataTypes : 338

o-andM-VPLMN.....identifier of Named Number, 2
 DEFINED in MAP-CommonDataTypes : 339

O-BcsmCamelTDPCriteriaList.....type reference SEQUENCE OF
 DEFINED in MAP-MS-DataTypes : 1036
 USED in MAP-MS-DataTypes : 54 939 1479
 USED in MAP-CH-DataTypes : 45 273

O-BcsmCamelTDPData.....type reference SEQUENCE
 DEFINED in MAP-MS-DataTypes : 1013
 USED in MAP-MS-DataTypes : 1005

o-BcsmCamelTDPDataList.....identifier of O-BcsmCamelTDPDataList
DEFINED in MAP-MS-DataTypes : 994

63

TAG R4.21 Cross Reference Listing for MAP-Protocol 00-01-06 07:32:40 PAGE

O-BcsmCamelTDPDataList.....type reference SEQUENCE OF
 DEFINED in MAP-MS-DataTypes : 1004
 USED in MAP-MS-DataTypes : 994

O-BcsmCamelTDP-Criteria.....type reference SEQUENCE
 DEFINED in MAP-MS-DataTypes : 1042
 USED in MAP-MS-DataTypes : 1037

o-BcsmCamelTDP-CriteriaList.....identifier of [4] O-BcsmCamelTDPCriteriaList
 DEFINED in MAP-MS-DataTypes : 939

o-BcsmCamelTDP-CriteriaList.....identifier of [1] O-BcsmCamelTDPCriteriaList
 DEFINED in MAP-MS-DataTypes : 1479

o-BcsmCamelTDP-CriteriaList.....identifier of [3] O-BcsmCamelTDPCriteriaList
 DEFINED in MAP-CH-DataTypes : 273

o-BcsmTriggerDetectionPoint.....identifier of O-BcsmTriggerDetectionPoint
 DEFINED in MAP-MS-DataTypes : 1014

O-BcsmTriggerDetectionPoint.....type reference ENUMERATED
 DEFINED in MAP-MS-DataTypes : 1024
 USED in MAP-MS-DataTypes : 1014 1043

o-BcsmTriggerDetectionPoint.....identifier of O-BcsmTriggerDetectionPoint
 DEFINED in MAP-MS-DataTypes : 1043

o-CauseValueCriteria.....identifier of [3] O-CauseValueCriteria
 DEFINED in MAP-MS-DataTypes : 1048

O-CauseValueCriteria.....type reference SEQUENCE OF
 DEFINED in MAP-MS-DataTypes : 1092
 USED in MAP-MS-DataTypes : 1048

o-CSI.....identifier of [0] O-CSI
 DEFINED in MAP-MS-DataTypes : 935

O-CSI.....type reference SEQUENCE
 DEFINED in MAP-MS-DataTypes : 993
 USED in MAP-MS-DataTypes : 52 935 1478
 USED in MAP-CH-DataTypes : 43 226 270

o-CSI.....identifier of Named Number, 0
 DEFINED in MAP-MS-DataTypes : 1445

o-CSI.....identifier of [0] O-CSI
 DEFINED in MAP-MS-DataTypes : 1478

o-CSI.....identifier of [5] O-CSI
 DEFINED in MAP-CH-DataTypes : 226

o-CSI.....identifier of [1] O-CSI
 DEFINED in MAP-CH-DataTypes : 270

padAccessCA-1200bps.....value reference BearerServiceCode, '00100010'B
 DEFINED in MAP-BS-Code : 69

padAccessCA-1200-75bps.....value reference BearerServiceCode, '00100011'B
 DEFINED in MAP-BS-Code : 70

padAccessCA-2400bps.....value reference BearerServiceCode, '00100100'B
 DEFINED in MAP-BS-Code : 71

padAccessCA-300bps.....value reference BearerServiceCode, '00100001'B
 DEFINED in MAP-BS-Code : 68

padAccessCA-4800bps.....value reference BearerServiceCode, '00100101'B
 DEFINED in MAP-BS-Code : 72

padAccessCA-9600bps.....value reference BearerServiceCode, '00100110'B
 DEFINED in MAP-BS-Code : 73

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

64

TAG R4.21 Cross Reference Listing for MAP-Protocol 00-01-06 07:32:40 PAGE

password.....identifier of Password
 DEFINED in MAP-MS-DataTypes : 1463

password.....identifier of [2] Password
 DEFINED in MAP-MS-DataTypes : 1584

Password.....type reference NumericString
 DEFINED in MAP-SS-DataTypes : 233
 USED in MAP-SupplementaryServi : 66 221 238
 USED in MAP-MS-DataTypes : 122 1463 1584
 USED in MAP-SS-DataTypes : 24

pcs-Extensions.....identifier of [1] PCS-Extensions
 DEFINED in MAP-ExtensionDataTypes : 34

PCS-Extensions.....type reference SEQUENCE
 DEFINED in MAP-ExtensionDataTypes : 56
 USED in MAP-ExtensionDataTypes : 34

pdp-Address.....identifier of [17] PDP-Address
 DEFINED in MAP-MS-DataTypes : 447

PDP-Address.....type reference OCTET STRING
 DEFINED in MAP-MS-DataTypes : 525
 USED in MAP-MS-DataTypes : 447

PDP-Context.....type reference SEQUENCE
 DEFINED in MAP-MS-DataTypes : 444
 USED in MAP-MS-DataTypes : 440

pdp-ContextChangeOfPosition.....identifier of Named Number, 14
 DEFINED in MAP-MS-DataTypes : 511

pdp-ContextEstablishment.....identifier of Named Number, 11
 DEFINED in MAP-MS-DataTypes : 509

pdp-ContextEstablishmentAcknowledgement.....identifier of Named Number, 12
 DEFINED in MAP-MS-DataTypes : 510

pdp-ContextId.....identifier of ContextId
 DEFINED in MAP-MS-DataTypes : 445

pdp-Type.....identifier of [16] PDP-Type
 DEFINED in MAP-MS-DataTypes : 446

PDP-Type.....type reference OCTET STRING
 DEFINED in MAP-MS-DataTypes : 522
 USED in MAP-MS-DataTypes : 446

permanent.....identifier of Named Number, 0
 DEFINED in MAP-SS-DataTypes : 169

phase1.....identifier of Named Number, 0
 DEFINED in MAP-MS-DataTypes : 1122

phase2.....identifier of Named Number, 1
 DEFINED in MAP-MS-DataTypes : 1123

phase3.....identifier of Named Number, 2
 DEFINED in MAP-MS-DataTypes : 1124

plmn.....identifier of Named Number, 0
 DEFINED in MAP-CommonDataTypes : 305

plmnClientList.....identifier of [2] PLMNClientList
 DEFINED in MAP-MS-DataTypes : 823

PLMNClientList.....type reference SEQUENCE OF
 DEFINED in MAP-MS-DataTypes : 834
 USED in MAP-MS-DataTypes : 823

plmnoperator.....value reference SS-Code, '10110100'B
 DEFINED in MAP-SS-Code : 166

plmnOperatorServices.....identifier of Named Number, 2
 DEFINED in MAP-LCS-DataTypes : 110

plmnRoamingNotAllowed.....identifier of Named Number, 0
 DEFINED in MAP-ER-DataTypes : 96

plmn-SpecificBarringType1.....identifier of Named Number, 0
DEFINED in MAP-MS-DataTypes : 635

TAG R4.21 Cross Reference Listing for MAP-Protocol 00-01-06 07:32:40 PAGE

plmn-SpecificBarringType2.....identifier of Named Number, 1
 DEFINED in MAP-MS-DataTypes : 636

plmn-SpecificBarringType3.....identifier of Named Number, 2
 DEFINED in MAP-MS-DataTypes : 637

plmn-SpecificBarringType4.....identifier of Named Number, 3
 DEFINED in MAP-MS-DataTypes : 638

plmn-specificBS-1.....value reference BearerServiceCode, '11010001'B
 DEFINED in MAP-BS-Code : 111

plmn-specificBS-2.....value reference BearerServiceCode, '11010010'B
 DEFINED in MAP-BS-Code : 112

plmn-specificBS-3.....value reference BearerServiceCode, '11010011'B
 DEFINED in MAP-BS-Code : 113

plmn-specificBS-4.....value reference BearerServiceCode, '11010100'B
 DEFINED in MAP-BS-Code : 114

plmn-specificBS-5.....value reference BearerServiceCode, '11010101'B
 DEFINED in MAP-BS-Code : 115

plmn-specificBS-6.....value reference BearerServiceCode, '11010110'B
 DEFINED in MAP-BS-Code : 116

plmn-specificBS-7.....value reference BearerServiceCode, '11010111'B
 DEFINED in MAP-BS-Code : 117

plmn-specificBS-8.....value reference BearerServiceCode, '11011000'B
 DEFINED in MAP-BS-Code : 118

plmn-specificBS-9.....value reference BearerServiceCode, '11011001'B
 DEFINED in MAP-BS-Code : 119

plmn-specificBS-A.....value reference BearerServiceCode, '11011010'B
 DEFINED in MAP-BS-Code : 120

plmn-specificBS-B.....value reference BearerServiceCode, '11011011'B
 DEFINED in MAP-BS-Code : 121

plmn-specificBS-C.....value reference BearerServiceCode, '11011100'B
 DEFINED in MAP-BS-Code : 122

plmn-specificBS-D.....value reference BearerServiceCode, '11011101'B
 DEFINED in MAP-BS-Code : 123

plmn-specificBS-E.....value reference BearerServiceCode, '11011110'B
 DEFINED in MAP-BS-Code : 124

plmn-specificBS-F.....value reference BearerServiceCode, '11011111'B
 DEFINED in MAP-BS-Code : 125

plmn-specificSS-1.....value reference SS-Code, '11110001'B
 DEFINED in MAP-SS-Code : 135

plmn-specificSS-2.....value reference SS-Code, '11110010'B
 DEFINED in MAP-SS-Code : 136

plmn-specificSS-3.....value reference SS-Code, '11110011'B
 DEFINED in MAP-SS-Code : 137

plmn-specificSS-4.....value reference SS-Code, '11110100'B
 DEFINED in MAP-SS-Code : 138

plmn-specificSS-5.....value reference SS-Code, '11110101'B
 DEFINED in MAP-SS-Code : 139

plmn-specificSS-6.....value reference SS-Code, '11110110'B
 DEFINED in MAP-SS-Code : 140

plmn-specificSS-7.....value reference SS-Code, '11110111'B
 DEFINED in MAP-SS-Code : 141

plmn-specificSS-8.....value reference SS-Code, '11111000'B
 DEFINED in MAP-SS-Code : 142

plmn-specificSS-9.....value reference SS-Code, '11111001'B

DEFINED in MAP-SS-Code : 143

66 TAG R4.21 Cross Reference Listing for MAP-Protocol 00-01-06 07:32:40 PAGE

plmn-specificSS-A.....value reference SS-Code, '11111010'B
 DEFINED in MAP-SS-Code : 144

plmn-specificSS-B.....value reference SS-Code, '11111011'B
 DEFINED in MAP-SS-Code : 145

plmn-specificSS-C.....value reference SS-Code, '11111100'B
 DEFINED in MAP-SS-Code : 146

plmn-specificSS-D.....value reference SS-Code, '11111101'B
 DEFINED in MAP-SS-Code : 147

plmn-specificSS-E.....value reference SS-Code, '11111110'B
 DEFINED in MAP-SS-Code : 148

plmn-specificSS-F.....value reference SS-Code, '11111111'B
 DEFINED in MAP-SS-Code : 149

plmn-specificTS-1.....value reference TeleserviceCode, '11010001'B
 DEFINED in MAP-TS-Code : 73

plmn-specificTS-2.....value reference TeleserviceCode, '11010010'B
 DEFINED in MAP-TS-Code : 74

plmn-specificTS-3.....value reference TeleserviceCode, '11010011'B
 DEFINED in MAP-TS-Code : 75

plmn-specificTS-4.....value reference TeleserviceCode, '11010100'B
 DEFINED in MAP-TS-Code : 76

plmn-specificTS-5.....value reference TeleserviceCode, '11010101'B
 DEFINED in MAP-TS-Code : 77

plmn-specificTS-6.....value reference TeleserviceCode, '11010110'B
 DEFINED in MAP-TS-Code : 78

plmn-specificTS-7.....value reference TeleserviceCode, '11010111'B
 DEFINED in MAP-TS-Code : 79

plmn-specificTS-8.....value reference TeleserviceCode, '11011000'B
 DEFINED in MAP-TS-Code : 80

plmn-specificTS-9.....value reference TeleserviceCode, '11011001'B
 DEFINED in MAP-TS-Code : 81

plmn-specificTS-A.....value reference TeleserviceCode, '11011010'B
 DEFINED in MAP-TS-Code : 82

plmn-specificTS-B.....value reference TeleserviceCode, '11011011'B
 DEFINED in MAP-TS-Code : 83

plmn-specificTS-C.....value reference TeleserviceCode, '11011100'B
 DEFINED in MAP-TS-Code : 84

plmn-specificTS-D.....value reference TeleserviceCode, '11011101'B
 DEFINED in MAP-TS-Code : 85

plmn-specificTS-E.....value reference TeleserviceCode, '11011110'B
 DEFINED in MAP-TS-Code : 86

plmn-specificTS-F.....value reference TeleserviceCode, '11011111'B
 DEFINED in MAP-TS-Code : 87

positionMethodFailure.....value reference PositionMethodFailure, CHOICE VALUE
 DEFINED in MAP-Protocol : 420

PositionMethodFailure.....type reference ERROR
 DEFINED in MAP-Errors : 431
 USED in MAP-Protocol : 163 420
 USED in MAP-LocationServiceOpe : 31 82
 USED in MAP-Errors : 87

positionMethodFailure-Diagnostic.....identifier of [0] PositionMethodFailure-Diagnostic
 DEFINED in MAP-ER-DataTypes : 354

PositionMethodFailure-Diagnostic.....type reference ENUMERATED
 DEFINED in MAP-ER-DataTypes : 358
 USED in MAP-ER-DataTypes : 354

positionMethodFailure-Param.....identifier of PositionMethodFailure-Param

DEFINED in MAP-Errors : 433

67	<p>TAG R4.21 Cross Reference Listing for MAP-Protocol</p> <p>00-01-06 07:32:40 PAGE</p>
	<p>PositionMethodFailure-Param.....type reference SEQUENCE DEFINED in MAP-ER-DataTypes : 353 USED in MAP-Errors : 142 433 USED in MAP-ER-DataTypes : 51</p> <p>preferentialCUG-Indicator.....identifier of CUG-Index DEFINED in MAP-MS-DataTypes : 786</p> <p>premiumRateEntertainmentOGCallsBarred..identifier of Named Number, 4 DEFINED in MAP-MS-DataTypes : 623</p> <p>premiumRateInformationOGCallsBarred....identifier of Named Number, 3 DEFINED in MAP-MS-DataTypes : 622</p> <p>prepareGroupCall.....value reference PrepareGroupCall, CHOICE VALUE DEFINED in MAP-Protocol : 290</p> <p>PrepareGroupCall.....type reference OPERATION DEFINED in MAP-Group-Call-Operati : 46 USED in MAP-Protocol : 98 290 USED in MAP-Group-Call-Operati : 13</p> <p>prepareGroupCallArg.....identifier of PrepareGroupCallArg DEFINED in MAP-Group-Call-Operati : 48</p> <p>PrepareGroupCallArg.....type reference SEQUENCE DEFINED in MAP-GR-DataTypes : 49 USED in MAP-Group-Call-Operati : 31 48 USED in MAP-GR-DataTypes : 14</p> <p>prepareGroupCallRes.....identifier of PrepareGroupCallRes DEFINED in MAP-Group-Call-Operati : 50</p> <p>PrepareGroupCallRes.....type reference SEQUENCE DEFINED in MAP-GR-DataTypes : 61 USED in MAP-Group-Call-Operati : 32 50 USED in MAP-GR-DataTypes : 15</p> <p>prepareHandover.....value reference PrepareHandover, CHOICE VALUE DEFINED in MAP-Protocol : 186</p> <p>PrepareHandover.....type reference OPERATION DEFINED in MAP-MobileServiceOpera : 301 USED in MAP-Protocol : 17 186 USED in MAP-MobileServiceOpera : 38</p> <p>prepareHO-Arg.....identifier of PrepareHO-Arg DEFINED in MAP-MobileServiceOpera : 303</p> <p>PrepareHO-Arg.....type reference SEQUENCE DEFINED in MAP-MS-DataTypes : 340 USED in MAP-MobileServiceOpera : 121 303 USED in MAP-MS-DataTypes : 30</p> <p>prepareHO-Res.....identifier of PrepareHO-Res DEFINED in MAP-MobileServiceOpera : 305</p> <p>PrepareHO-Res.....type reference SEQUENCE DEFINED in MAP-MS-DataTypes : 346 USED in MAP-MobileServiceOpera : 122 305 USED in MAP-MS-DataTypes : 31</p> <p>prepareSubsequentHandover.....value reference PrepareSubsequentHandover, CHOICE VALUE DEFINED in MAP-Protocol : 190</p> <p>PrepareSubsequentHandover.....type reference OPERATION DEFINED in MAP-MobileServiceOpera : 325 USED in MAP-Protocol : 21 190 USED in MAP-MobileServiceOpera : 42</p> <p>prepareSubsequentHO-Arg.....identifier of PrepareSubsequentHO-Arg DEFINED in MAP-MobileServiceOpera : 327</p> <p>PrepareSubsequentHO-Arg.....type reference SEQUENCE DEFINED in MAP-MS-DataTypes : 351 USED in MAP-MobileServiceOpera : 123 327 USED in MAP-MS-DataTypes : 32</p> <p>pre-pagingSupported.....identifier of [19] NULL</p>

DEFINED in MAP-CH-DataTypes : 114
pre-pagingSupported.....identifier of [17] NULL

68

TAG R4.21 Cross Reference Listing for MAP-Protocol 00-01-06 07:32:40 PAGE

DEFINED in MAP-CH-DataTypes : 213

priority.....identifier of [2] EMLPP-Priority
 DEFINED in MAP-GR-DataTypes : 56

priorityLevel0.....value reference EMLPP-Priority, 0
 DEFINED in MAP-CommonDataTypes : 401

priorityLevel1.....value reference EMLPP-Priority, 1
 DEFINED in MAP-CommonDataTypes : 402

priorityLevel2.....value reference EMLPP-Priority, 2
 DEFINED in MAP-CommonDataTypes : 403

priorityLevel3.....value reference EMLPP-Priority, 3
 DEFINED in MAP-CommonDataTypes : 404

priorityLevel4.....value reference EMLPP-Priority, 4
 DEFINED in MAP-CommonDataTypes : 405

priorityLevelA.....value reference EMLPP-Priority, 6
 DEFINED in MAP-CommonDataTypes : 399

priorityLevelB.....value reference EMLPP-Priority, 5
 DEFINED in MAP-CommonDataTypes : 400

privacyOverride.....identifier of [1] NULL
 DEFINED in MAP-LCS-DataTypes : 74

privacyOverrideNotApplicable.....identifier of Named Number, 3
 DEFINED in MAP-ER-DataTypes : 347

privacyVerificationByMSuser.....identifier of [0] NULL
 DEFINED in MAP-MS-DataTypes : 819

PrivateExtension.....type reference SEQUENCE
 DEFINED in MAP-ExtensionDataTypes : 40
 USED in MAP-ExtensionDataTypes : 15 38

privateExtensionList.....identifier of [0] PrivateExtensionList
 DEFINED in MAP-ExtensionDataTypes : 33

PrivateExtensionList.....type reference SEQUENCE OF
 DEFINED in MAP-ExtensionDataTypes : 37
 USED in MAP-ExtensionDataTypes : 33

problem.....identifier of CHOICE
 DEFINED in TCAPMessages : 169

processAccessSignalling.....value reference ProcessAccessSignalling, CHOICE VALUE
 DEFINED in MAP-Protocol : 188

ProcessAccessSignalling.....type reference OPERATION
 DEFINED in MAP-MobileServiceOpera : 317
 USED in MAP-Protocol : 19 188
 USED in MAP-MobileServiceOpera : 40

processGroupCallSignalling.....value reference ProcessGroupCallSignalling, CHOICE
 VALUE
 DEFINED in MAP-Protocol : 292

ProcessGroupCallSignalling.....type reference OPERATION
 DEFINED in MAP-Group-Call-Operati : 63
 USED in MAP-Protocol : 99 292
 USED in MAP-Group-Call-Operati : 16

processGroupCallSignallingArg.....identifier of ProcessGroupCallSignallingArg
 DEFINED in MAP-Group-Call-Operati : 65

ProcessGroupCallSignallingArg.....type reference SEQUENCE
 DEFINED in MAP-GR-DataTypes : 85
 USED in MAP-Group-Call-Operati : 36 65
 USED in MAP-GR-DataTypes : 19

processUnstructuredSS-Request.....value reference ProcessUnstructuredSS-Request, CHOICE
 VALUE
 DEFINED in MAP-Protocol : 246

ProcessUnstructuredSS-Request.....type reference OPERATION
 DEFINED in MAP-SupplementaryServi : 175

USED in MAP-Protocol : 73 246
USED in MAP-SupplementaryServi : 18

protocolId.....identifier of ProtocolId

69

TAG R4.21 Cross Reference Listing for MAP-Protocol 00-01-06 07:32:40 PAGE

DEFINED in MAP-CommonDataTypes : 178

ProtocolId.....type reference ENUMERATED
 DEFINED in MAP-CommonDataTypes : 196
 USED in MAP-CommonDataTypes : 178

provideRoamingNumber.....value reference ProvideRoamingNumber, CHOICE VALUE
 DEFINED in MAP-Protocol : 228

ProvideRoamingNumber.....type reference OPERATION
 DEFINED in MAP-CallHandlingOperat : 103
 USED in MAP-Protocol : 54 228
 USED in MAP-CallHandlingOperat : 14

provideRoamingNumberArg.....identifier of ProvideRoamingNumberArg
 DEFINED in MAP-CallHandlingOperat : 106

ProvideRoamingNumberArg.....type reference SEQUENCE
 DEFINED in MAP-CH-DataTypes : 195
 USED in MAP-CallHandlingOperat : 56 106
 USED in MAP-CH-DataTypes : 16

provideRoamingNumberRes.....identifier of ProvideRoamingNumberRes
 DEFINED in MAP-CallHandlingOperat : 108

ProvideRoamingNumberRes.....type reference SEQUENCE
 DEFINED in MAP-CH-DataTypes : 215
 USED in MAP-CallHandlingOperat : 57 108
 USED in MAP-CH-DataTypes : 17

provideSIWFSNumber.....value reference ProvideSIWFSNumber, CHOICE VALUE
 DEFINED in MAP-Protocol : 230

ProvideSIWFSNumber.....type reference OPERATION
 DEFINED in MAP-CallHandlingOperat : 130
 USED in MAP-Protocol : 56 230
 USED in MAP-CallHandlingOperat : 16

provideSIWFSNumberArg.....identifier of ProvideSIWFSNumberArg
 DEFINED in MAP-CallHandlingOperat : 132

ProvideSIWFSNumberArg.....type reference SEQUENCE
 DEFINED in MAP-CH-DataTypes : 277
 USED in MAP-CallHandlingOperat : 60 132
 USED in MAP-CH-DataTypes : 23

provideSIWFSNumberRes.....identifier of ProvideSIWFSNumberRes
 DEFINED in MAP-CallHandlingOperat : 134

ProvideSIWFSNumberRes.....type reference SEQUENCE
 DEFINED in MAP-CH-DataTypes : 296
 USED in MAP-CallHandlingOperat : 61 134
 USED in MAP-CH-DataTypes : 24

provideSubscriberInfo.....value reference ProvideSubscriberInfo, CHOICE VALUE
 DEFINED in MAP-Protocol : 268

ProvideSubscriberInfo.....type reference OPERATION
 DEFINED in MAP-MobileServiceOpera : 225
 USED in MAP-Protocol : 29 268
 USED in MAP-MobileServiceOpera : 24

provideSubscriberInfoArg.....identifier of ProvideSubscriberInfoArg
 DEFINED in MAP-MobileServiceOpera : 227

ProvideSubscriberInfoArg.....type reference SEQUENCE
 DEFINED in MAP-MS-DataTypes : 1328
 USED in MAP-MobileServiceOpera : 134 227
 USED in MAP-MS-DataTypes : 77

provideSubscriberInfoRes.....identifier of ProvideSubscriberInfoRes
 DEFINED in MAP-MobileServiceOpera : 229

ProvideSubscriberInfoRes.....type reference SEQUENCE
 DEFINED in MAP-MS-DataTypes : 1335
 USED in MAP-MobileServiceOpera : 135 229
 USED in MAP-MS-DataTypes : 78

provideSubscriberLocation.....value reference ProvideSubscriberLocation, CHOICE

VALUE

DEFINED in MAP-Protocol : 314
ProvideSubscriberLocation.....type reference OPERATION

70 TAG R4.21 Cross Reference Listing for MAP-Protocol 00-01-06 07:32:40 PAGE

DEFINED in MAP-LocationServiceOpe : 66
 USED in MAP-Protocol : 107 314
 USED in MAP-LocationServiceOpe : 13

provideSubscriberLocation-Arg.....identifier of ProvideSubscriberLocation-Arg
 DEFINED in MAP-LocationServiceOpe : 68

ProvideSubscriberLocation-Arg.....type reference SEQUENCE
 DEFINED in MAP-LCS-DataTypes : 70
 USED in MAP-LocationServiceOpe : 43 68
 USED in MAP-LCS-DataTypes : 13

provideSubscriberLocation-Res.....identifier of ProvideSubscriberLocation-Res
 DEFINED in MAP-LocationServiceOpe : 70

ProvideSubscriberLocation-Res.....type reference SEQUENCE
 DEFINED in MAP-LCS-DataTypes : 164
 USED in MAP-LocationServiceOpe : 44 70
 USED in MAP-LCS-DataTypes : 14

provisionedSS.....identifier of [7] Ext-SS-InfoList
 DEFINED in MAP-MS-DataTypes : 583

purgedMS.....identifier of Named Number, 3
 DEFINED in MAP-ER-DataTypes : 251

purgeMS.....value reference PurgeMS, CHOICE VALUE
 DEFINED in MAP-Protocol : 180

PurgeMS.....type reference OPERATION
 DEFINED in MAP-MobileServiceOpera : 190
 USED in MAP-Protocol : 14 180
 USED in MAP-MobileServiceOpera : 17

purgeMS-Arg.....identifier of PurgeMS-Arg
 DEFINED in MAP-MobileServiceOpera : 192

PurgeMS-Arg.....type reference [3] SEQUENCE
 DEFINED in MAP-MS-DataTypes : 241
 USED in MAP-MobileServiceOpera : 115 192
 USED in MAP-MS-DataTypes : 20

purgeMS-Res.....identifier of PurgeMS-Res
 DEFINED in MAP-MobileServiceOpera : 194

PurgeMS-Res.....type reference SEQUENCE
 DEFINED in MAP-MS-DataTypes : 248
 USED in MAP-MobileServiceOpera : 116 194
 USED in MAP-MS-DataTypes : 21

pvlr.....identifier of Named Number, 3
 DEFINED in MAP-CommonDataTypes : 308

pw-RegistrationFailure.....value reference PW-RegistrationFailure, CHOICE VALUE
 DEFINED in MAP-Protocol : 401

PW-RegistrationFailure.....type reference ERROR
 DEFINED in MAP-Errors : 372
 USED in MAP-Protocol : 149 401
 USED in MAP-SupplementaryServi : 45 228
 USED in MAP-Errors : 69

pw-RegistrationFailureCause.....identifier of PW-RegistrationFailureCause
 DEFINED in MAP-Errors : 374

PW-RegistrationFailureCause.....type reference ENUMERATED
 DEFINED in MAP-ER-DataTypes : 133
 USED in MAP-Errors : 107 374
 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

qoSNotAttainable.....identifier of Named Number, 6
 DEFINED in MAP-ER-DataTypes : 365

qos-Subscribed.....identifier of [18] QoS-Subscribed
DEFINED in MAP-MS-DataTypes : 448

71 TAG R4.21 Cross Reference Listing for MAP-Protocol 00-01-06 07:32:40 PAGE

QoS-Subscribed.....type reference OCTET STRING
 DEFINED in MAP-MS-DataTypes : 533
 USED in MAP-MS-DataTypes : 448

quintupletList.....identifier of [1] QuintupletList
 DEFINED in MAP-MS-DataTypes : 274

QuintupletList.....type reference SEQUENCE OF
 DEFINED in MAP-MS-DataTypes : 279
 USED in MAP-MS-DataTypes : 274

rand.....identifier of RAND
 DEFINED in MAP-MS-DataTypes : 283

rand.....identifier of RAND
 DEFINED in MAP-MS-DataTypes : 289

RAND.....type reference OCTET STRING
 DEFINED in MAP-MS-DataTypes : 296
 USED in MAP-MS-DataTypes : 283 289 375 376

rand.....identifier of RAND
 DEFINED in MAP-MS-DataTypes : 375

rand-ms.....identifier of RAND
 DEFINED in MAP-MS-DataTypes : 376

readyForSM.....value reference ReadyForSM, CHOICE VALUE
 DEFINED in MAP-Protocol : 264

ReadyForSM.....type reference OPERATION
 DEFINED in MAP-ShortMessageServic : 136
 USED in MAP-Protocol : 92 264
 USED in MAP-ShortMessageServic : 19

readyForSM-Arg.....identifier of ReadyForSM-Arg
 DEFINED in MAP-ShortMessageServic : 138

ReadyForSM-Arg.....type reference SEQUENCE
 DEFINED in MAP-SM-DataTypes : 193
 USED in MAP-ShortMessageServic : 55 138
 USED in MAP-SM-DataTypes : 24

readyForSM-Res.....identifier of ReadyForSM-Res
 DEFINED in MAP-ShortMessageServic : 140

ReadyForSM-Res.....type reference SEQUENCE
 DEFINED in MAP-SM-DataTypes : 202
 USED in MAP-ShortMessageServic : 56 140
 USED in MAP-SM-DataTypes : 25

reason.....identifier of CHOICE
 DEFINED in TCAPMessages : 75

regionalSubscNotSupported.....identifier of Named Number, 3
 DEFINED in MAP-MS-DataTypes : 888

regionalSubscriptionData.....identifier of [10] ZoneCodeList
 DEFINED in MAP-MS-DataTypes : 586

regionalSubscriptionIdentifier.....identifier of [5] ZoneCode
 DEFINED in MAP-MS-DataTypes : 897

regionalSubscriptionResponse.....identifier of [5] RegionalSubscriptionResponse
 DEFINED in MAP-MS-DataTypes : 878

RegionalSubscriptionResponse.....type reference ENUMERATED
 DEFINED in MAP-MS-DataTypes : 884
 USED in MAP-MS-DataTypes : 879 930

regionalSubscriptionResponse.....identifier of [0] RegionalSubscriptionResponse
 DEFINED in MAP-MS-DataTypes : 929

registerCC-Entry.....value reference RegisterCC-Entry, CHOICE VALUE
 DEFINED in MAP-Protocol : 252

RegisterCC-Entry.....type reference OPERATION
 DEFINED in MAP-SupplementaryServi : 251
 USED in MAP-Protocol : 79 252

USED in MAP-SupplementaryServi : 24

72

TAG R4.21 Cross Reference Listing for MAP-Protocol 00-01-06 07:32:40 PAGE

registerCC-EntryArg.....identifier of RegisterCC-EntryArg
 DEFINED in MAP-SupplementaryServi : 253

RegisterCC-EntryArg.....type reference SEQUENCE
 DEFINED in MAP-SS-DataTypes : 281
 USED in MAP-SupplementaryServi : 70 253
 USED in MAP-SS-DataTypes : 37

registerCC-EntryRes.....identifier of RegisterCC-EntryRes
 DEFINED in MAP-SupplementaryServi : 255

RegisterCC-EntryRes.....type reference SEQUENCE
 DEFINED in MAP-SS-DataTypes : 300
 USED in MAP-SupplementaryServi : 71 255
 USED in MAP-SS-DataTypes : 38

registerPassword.....value reference RegisterPassword, CHOICE VALUE
 DEFINED in MAP-Protocol : 250

RegisterPassword.....type reference OPERATION
 DEFINED in MAP-SupplementaryServi : 217
 USED in MAP-Protocol : 76 250
 USED in MAP-SupplementaryServi : 21

registerSS.....value reference RegisterSS, CHOICE VALUE
 DEFINED in MAP-Protocol : 241

RegisterSS.....type reference OPERATION
 DEFINED in MAP-SupplementaryServi : 87
 USED in MAP-Protocol : 68 241
 USED in MAP-SupplementaryServi : 13

registerSS-Arg.....identifier of RegisterSS-Arg
 DEFINED in MAP-SupplementaryServi : 89

RegisterSS-Arg.....type reference SEQUENCE
 DEFINED in MAP-SS-DataTypes : 69
 USED in MAP-SupplementaryServi : 60 89
 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

rejected.....identifier of Named Number, 1
 DEFINED in MAP-CH-DataTypes : 399

releaseCall.....identifier of Named Number, 1
 DEFINED in MAP-MS-DataTypes : 1109

releaseGroupCall.....identifier of [2] NULL
 DEFINED in MAP-GR-DataTypes : 88

releaseTransaction.....identifier of Named Number, 1
 DEFINED in MAP-MS-DataTypes : 500

releaseTransaction.....identifier of Named Number, 1
 DEFINED in MAP-MS-DataTypes : 1162

remoteUserFree.....value reference RemoteUserFree, CHOICE VALUE
 DEFINED in MAP-Protocol : 234

RemoteUserFree.....type reference OPERATION
 DEFINED in MAP-CallHandlingOperat : 179
 USED in MAP-Protocol : 60 234
 USED in MAP-CallHandlingOperat : 20

remoteUserFreeArg.....identifier of RemoteUserFreeArg
 DEFINED in MAP-CallHandlingOperat : 181

RemoteUserFreeArg.....type reference SEQUENCE
 DEFINED in MAP-CH-DataTypes : 382
 USED in MAP-CallHandlingOperat : 68 181
 USED in MAP-CH-DataTypes : 31

remoteUserFreeRes.....identifier of RemoteUserFreeRes
 DEFINED in MAP-CallHandlingOperat : 183

RemoteUserFreeRes.....type reference SEQUENCE

73

TAG R4.21 Cross Reference Listing for MAP-Protocol 00-01-06 07:32:40 PAGE

DEFINED in MAP-CH-DataTypes : 392
 USED in MAP-CallHandlingOperat : 69 183
 USED in MAP-CH-DataTypes : 32

replaceB-Number.....identifier of [4] NULL
 DEFINED in MAP-CH-DataTypes : 387

ReportingState.....type reference ENUMERATED
 DEFINED in MAP-CH-DataTypes : 319
 USED in MAP-CH-DataTypes : 315

reportSM-DeliveryStatus.....value reference ReportSM-DeliveryStatus, CHOICE VALUE
 DEFINED in MAP-Protocol : 261

ReportSM-DeliveryStatus.....type reference OPERATION
 DEFINED in MAP-ShortMessageServic : 111
 USED in MAP-Protocol : 89 261
 USED in MAP-ShortMessageServic : 16

reportSM-DeliveryStatusArg.....identifier of ReportSM-DeliveryStatusArg
 DEFINED in MAP-ShortMessageServic : 113

ReportSM-DeliveryStatusArg.....type reference SEQUENCE
 DEFINED in MAP-SM-DataTypes : 143
 USED in MAP-ShortMessageServic : 51 113
 USED in MAP-SM-DataTypes : 20

reportSM-DeliveryStatusRes.....identifier of ReportSM-DeliveryStatusRes
 DEFINED in MAP-ShortMessageServic : 115

ReportSM-DeliveryStatusRes.....type reference SEQUENCE
 DEFINED in MAP-SM-DataTypes : 168
 USED in MAP-ShortMessageServic : 52 115
 USED in MAP-SM-DataTypes : 21

requestedBasicServiceViolatesCUG-Constraidentifier of Named Number, 5
 DEFINED in MAP-ER-DataTypes : 124

requestedCamelSubscriptionInfo.....identifier of [0] RequestedCAMEL-SubscriptionInfo
 DEFINED in MAP-MS-DataTypes : 1519

requestedCAMEL-SubscriptionInfo.....identifier of [3] RequestedCAMEL-SubscriptionInfo
 DEFINED in MAP-MS-DataTypes : 1438

RequestedCAMEL-SubscriptionInfo.....type reference ENUMERATED
 DEFINED in MAP-MS-DataTypes : 1444
 USED in MAP-MS-DataTypes : 1438 1519

requestedInfo.....identifier of [2] RequestedInfo
 DEFINED in MAP-MS-DataTypes : 1331

RequestedInfo.....type reference SEQUENCE
 DEFINED in MAP-MS-DataTypes : 1346
 USED in MAP-MS-DataTypes : 1331 1405

requestedInfo.....identifier of [1] RequestedInfo
 DEFINED in MAP-MS-DataTypes : 1405

requestedSS-Info.....identifier of [1] SS-ForBS-Code
 DEFINED in MAP-MS-DataTypes : 1436

requestedSubscriptionInfo.....identifier of [1] RequestedSubscriptionInfo
 DEFINED in MAP-MS-DataTypes : 1420

RequestedSubscriptionInfo.....type reference SEQUENCE
 DEFINED in MAP-MS-DataTypes : 1435
 USED in MAP-MS-DataTypes : 1420

reset.....value reference Reset, CHOICE VALUE
 DEFINED in MAP-Protocol : 212

Reset.....type reference OPERATION
 DEFINED in MAP-MobileServiceOpera : 388
 USED in MAP-Protocol : 26 212
 USED in MAP-MobileServiceOpera : 55

resetArg.....identifier of ResetArg
 DEFINED in MAP-MobileServiceOpera : 390

ResetArg.....type reference SEQUENCE

DEFINED in MAP-MS-DataTypes : 1281
USED in MAP-MobileServiceOpera : 131 390

74

TAG R4.21 Cross Reference Listing for MAP-Protocol 00-01-06 07:32:40 PAGE

USED in MAP-MS-DataTypes : 72

resourceLimitation.....value reference ResourceLimitation, CHOICE VALUE
 DEFINED in MAP-Protocol : 330

ResourceLimitation.....type reference ERROR
 DEFINED in MAP-Errors : 190
 USED in MAP-Protocol : 156 330
 USED in MAP-CallHandlingOperat : 47 136 148 164 200 213
 USED in MAP-LocationServiceOpe : 32 92
 USED in MAP-Errors : 19

resourceLimitation.....identifier of Named Number, 4
 DEFINED in TCAPMessages : 107

resourceLimitation.....identifier of Named Number, 3
 DEFINED in TCAPMessages : 186

resourceLimitationParam.....identifier of ResourceLimitationParam
 DEFINED in MAP-Errors : 192

ResourceLimitationParam.....type reference SEQUENCE
 DEFINED in MAP-ER-DataTypes : 316
 USED in MAP-Errors : 135 192
 USED in MAP-ER-DataTypes : 44

responseTime.....identifier of [3] ResponseTime
 DEFINED in MAP-LCS-DataTypes : 140

ResponseTime.....type reference SEQUENCE
 DEFINED in MAP-LCS-DataTypes : 152
 USED in MAP-LCS-DataTypes : 21 140

responseTimeCategory.....identifier of ResponseTimeCategory
 DEFINED in MAP-LCS-DataTypes : 153

ResponseTimeCategory.....type reference ENUMERATED
 DEFINED in MAP-LCS-DataTypes : 157
 USED in MAP-LCS-DataTypes : 153

restoreData.....value reference RestoreData, CHOICE VALUE
 DEFINED in MAP-Protocol : 215

RestoreData.....type reference OPERATION
 DEFINED in MAP-MobileServiceOpera : 394
 USED in MAP-Protocol : 28 215
 USED in MAP-MobileServiceOpera : 57

restoreDataArg.....identifier of RestoreDataArg
 DEFINED in MAP-MobileServiceOpera : 396

RestoreDataArg.....type reference SEQUENCE
 DEFINED in MAP-MS-DataTypes : 1286
 USED in MAP-MobileServiceOpera : 132 396
 USED in MAP-MS-DataTypes : 73

restoreDataRes.....identifier of RestoreDataRes
 DEFINED in MAP-MobileServiceOpera : 398

RestoreDataRes.....type reference SEQUENCE
 DEFINED in MAP-MS-DataTypes : 1293
 USED in MAP-MobileServiceOpera : 133 398
 USED in MAP-MS-DataTypes : 74

restrictedArea.....identifier of Named Number, 2
 DEFINED in MAP-MS-DataTypes : 1398

restrictedArea.....identifier of Named Number, 1
 DEFINED in MAP-ER-DataTypes : 248

result-RR.....identifier of SEQUENCE
 DEFINED in TCAPMessages : 146

resumeCallHandling.....value reference ResumeCallHandling, CHOICE VALUE
 DEFINED in MAP-Protocol : 229

ResumeCallHandling.....type reference OPERATION
 DEFINED in MAP-CallHandlingOperat : 118
 USED in MAP-Protocol : 55 229
 USED in MAP-CallHandlingOperat : 15

resumeCallHandlingArg.....identifier of ResumeCallHandlingArg

75

TAG R4.21 Cross Reference Listing for MAP-Protocol 00-01-06 07:32:40 PAGE

DEFINED in MAP-CallHandlingOperat : 120

ResumeCallHandlingArg.....type reference SEQUENCE
 DEFINED in MAP-CH-DataTypes : 220
 USED in MAP-CallHandlingOperat : 58 120
 USED in MAP-CH-DataTypes : 18

resumeCallHandlingRes.....identifier of ResumeCallHandlingRes
 DEFINED in MAP-CallHandlingOperat : 122

ResumeCallHandlingRes.....type reference SEQUENCE
 DEFINED in MAP-CH-DataTypes : 248
 USED in MAP-CallHandlingOperat : 59 122
 USED in MAP-CH-DataTypes : 19

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

re-synchronisationInfo.....identifier of Re-synchronisationInfo
 DEFINED in MAP-MS-DataTypes : 368

Re-synchronisationInfo.....type reference SEQUENCE
 DEFINED in MAP-MS-DataTypes : 374
 USED in MAP-MS-DataTypes : 368

roamingNotAllowed.....value reference RoamingNotAllowed, CHOICE VALUE
 DEFINED in MAP-Protocol : 344

RoamingNotAllowed.....type reference ERROR
 DEFINED in MAP-Errors : 221
 USED in MAP-Protocol : 124 344
 USED in MAP-MobileServiceOpera : 88 178 221
 USED in MAP-Errors : 29

roamingNotAllowedCause.....identifier of RoamingNotAllowedCause
 DEFINED in MAP-ER-DataTypes : 91

RoamingNotAllowedCause.....type reference ENUMERATED
 DEFINED in MAP-ER-DataTypes : 95
 USED in MAP-ER-DataTypes : 91

roamingNotAllowedParam.....identifier of RoamingNotAllowedParam
 DEFINED in MAP-Errors : 223

RoamingNotAllowedParam.....type reference SEQUENCE
 DEFINED in MAP-ER-DataTypes : 90

USED in MAP-Errors	:	116	223
USED in MAP-ER-DataTypes	:	14	

TAG	R4.21	Cross Reference Listing for MAP-Protocol	00-01-06	07:32:40	PAGE
roamingNumber	identifier of ISDN-AddressString			
	DEFINED in MAP-CH-DataTypes	:	182		
roamingNumber	identifier of ISDN-AddressString			
	DEFINED in MAP-CH-DataTypes	:	216		
roamingRestrictedInSgsnDueToUnsupportedFidentifier	of [23] NULL			
	DEFINED in MAP-MS-DataTypes	:	404		
roamingRestrictedInSgsnDueToUnsuppportedidentifier	of [11] NULL			
	DEFINED in MAP-MS-DataTypes	:	904		
roamingRestrictionDueToUnsupportedFeaturidentifier	of [9] NULL			
	DEFINED in MAP-MS-DataTypes	:	585		
roamingRestrictionDueToUnsupportedFeaturidentifier	of [4] NULL			
	DEFINED in MAP-MS-DataTypes	:	896		
routeSelectFailure	identifier of Named Number, 4			
	DEFINED in MAP-MS-DataTypes	:	1027		
RoutingInfo	type reference CHOICE			
	DEFINED in MAP-CH-DataTypes	:	181		
	USED in MAP-CH-DataTypes	:	259		
routingInfo	identifier of RoutingInfo			
	DEFINED in MAP-CH-DataTypes	:	259		
routingInfoForLCS-Arg	identifier of RoutingInfoForLCS-Arg			
	DEFINED in MAP-LocationServiceOpe	:	54		
RoutingInfoForLCS-Arg	type reference SEQUENCE			
	DEFINED in MAP-LCS-DataTypes	:	52		
	USED in MAP-LocationServiceOpe	:	41	54	
	USED in MAP-LCS-DataTypes	:	11		
routingInfoForLCS-Res	identifier of RoutingInfoForLCS-Res			
	DEFINED in MAP-LocationServiceOpe	:	56		
RoutingInfoForLCS-Res	type reference SEQUENCE			
	DEFINED in MAP-LCS-DataTypes	:	58		
	USED in MAP-LocationServiceOpe	:	42	56	
	USED in MAP-LCS-DataTypes	:	12		
routingInfoForSM-Arg	identifier of RoutingInfoForSM-Arg			
	DEFINED in MAP-ShortMessageServic	:	68		
RoutingInfoForSM-Arg	type reference SEQUENCE			
	DEFINED in MAP-SM-DataTypes	:	52		
	USED in MAP-ShortMessageServic	:	45	68	
	USED in MAP-SM-DataTypes	:	14		
routingInfoForSM-Res	identifier of RoutingInfoForSM-Res			
	DEFINED in MAP-ShortMessageServic	:	70		
RoutingInfoForSM-Res	type reference SEQUENCE			
	DEFINED in MAP-SM-DataTypes	:	79		
	USED in MAP-ShortMessageServic	:	46	70	
	USED in MAP-SM-DataTypes	:	15		
rss	identifier of Named Number, 7			
	DEFINED in MAP-CommonDataTypes	:	312		
ruf-Outcome	identifier of [0] RUF-Outcome			
	DEFINED in MAP-CH-DataTypes	:	393		
RUF-Outcome	type reference ENUMERATED			
	DEFINED in MAP-CH-DataTypes	:	397		
	USED in MAP-CH-DataTypes	:	393		
sc-AddressNotIncluded	identifier of Named Number, 0			
	DEFINED in MAP-SM-DataTypes	:	186		
sc-Congestion	identifier of Named Number, 4			
	DEFINED in MAP-ER-DataTypes	:	144		
segmentationProhibited	identifier of NULL			
	DEFINED in MAP-MS-DataTypes	:	257		

segmentationProhibited.....identifier of NULL
DEFINED in MAP-MS-DataTypes : 362

77

TAG R4.21 Cross Reference Listing for MAP-Protocol 00-01-06 07:32:40 PAGE

```

selectedLSA-Id.....identifier of [5] LSAIdentity
  DEFINED in MAP-MS-DataTypes      :   1360

sendAuthenticationInfo.....value reference SendAuthenticationInfo, CHOICE VALUE
  DEFINED in MAP-Protocol           :   196

SendAuthenticationInfo.....type reference OPERATION
  DEFINED in MAP-MobileServiceOpera :   338
  USED in MAP-Protocol              :    22   196
  USED in MAP-MobileServiceOpera    :    45

sendAuthenticationInfoArg.....identifier of SendAuthenticationInfoArg
  DEFINED in MAP-MobileServiceOpera :   340

SendAuthenticationInfoArg.....type reference SEQUENCE
  DEFINED in MAP-MS-DataTypes       :   359
  USED in MAP-MobileServiceOpera    :   124   340
  USED in MAP-MS-DataTypes          :    35

sendAuthenticationInfoRes.....identifier of SendAuthenticationInfoRes
  DEFINED in MAP-MobileServiceOpera :   342

SendAuthenticationInfoRes.....type reference [3] SEQUENCE
  DEFINED in MAP-MS-DataTypes       :   380
  USED in MAP-MobileServiceOpera    :   125   342
  USED in MAP-MS-DataTypes          :    36

sendEndSignal.....value reference SendEndSignal, CHOICE VALUE
  DEFINED in MAP-Protocol           :   187

SendEndSignal.....type reference OPERATION
  DEFINED in MAP-MobileServiceOpera :   312
  USED in MAP-Protocol              :    18   187
  USED in MAP-MobileServiceOpera    :    39

sendGroupCallEndSignal.....value reference SendGroupCallEndSignal, CHOICE VALUE
  DEFINED in MAP-Protocol           :   291

SendGroupCallEndSignal.....type reference OPERATION
  DEFINED in MAP-Group-Call-Operati :    56
  USED in MAP-Protocol              :   101   291
  USED in MAP-Group-Call-Operati    :    14

sendGroupCallEndSignalArg.....identifier of SendGroupCallEndSignalArg
  DEFINED in MAP-Group-Call-Operati :    58

SendGroupCallEndSignalArg.....type reference SEQUENCE
  DEFINED in MAP-GR-DataTypes       :    66
  USED in MAP-Group-Call-Operati    :    33   58
  USED in MAP-GR-DataTypes          :    16

sendGroupCallEndSignalRes.....identifier of SendGroupCallEndSignalRes
  DEFINED in MAP-Group-Call-Operati :    60

SendGroupCallEndSignalRes.....type reference SEQUENCE
  DEFINED in MAP-GR-DataTypes       :    71
  USED in MAP-Group-Call-Operati    :    34   60
  USED in MAP-GR-DataTypes          :    17

sendIdentification.....value reference SendIdentification, CHOICE VALUE
  DEFINED in MAP-Protocol           :   181

SendIdentification.....type reference OPERATION
  DEFINED in MAP-MobileServiceOpera :   201
  USED in MAP-Protocol              :    15   181
  USED in MAP-MobileServiceOpera    :    18

sendIdentificationArg.....identifier of SendIdentificationArg
  DEFINED in MAP-MobileServiceOpera :   203

SendIdentificationArg.....type reference SEQUENCE
  DEFINED in MAP-MS-DataTypes       :   254
  USED in MAP-MobileServiceOpera    :   117   203
  USED in MAP-MS-DataTypes          :    22

sendIdentificationRes.....identifier of SendIdentificationRes
  DEFINED in MAP-MobileServiceOpera :   205

SendIdentificationRes.....type reference [3] SEQUENCE
    
```

DEFINED in MAP-MS-DataTypes : 263
USED in MAP-MobileServiceOpera : 118 205

78

TAG R4.21 Cross Reference Listing for MAP-Protocol 00-01-06 07:32:40 PAGE

USED in MAP-MS-DataTypes : 23

sendIMSI.....value reference SendIMSI, CHOICE VALUE
 DEFINED in MAP-Protocol : 222

SendIMSI.....type reference OPERATION
 DEFINED in MAP-OperationAndMainte : 77
 USED in MAP-Protocol : 47 222
 USED in MAP-OperationAndMainte : 15

sendRoutingInfo.....value reference SendRoutingInfo, CHOICE VALUE
 DEFINED in MAP-Protocol : 227

SendRoutingInfo.....type reference OPERATION
 DEFINED in MAP-CallHandlingOperat : 80
 USED in MAP-Protocol : 53 227
 USED in MAP-CallHandlingOperat : 13

sendRoutingInfoArg.....identifier of SendRoutingInfoArg
 DEFINED in MAP-CallHandlingOperat : 83

SendRoutingInfoArg.....type reference SEQUENCE
 DEFINED in MAP-CH-DataTypes : 93
 USED in MAP-CallHandlingOperat : 54 83
 USED in MAP-CH-DataTypes : 14

sendRoutingInfoForGprs.....value reference SendRoutingInfoForGprs, CHOICE VALUE
 DEFINED in MAP-Protocol : 302

SendRoutingInfoForGprs.....type reference OPERATION
 DEFINED in MAP-MobileServiceOpera : 407
 USED in MAP-Protocol : 33 302
 USED in MAP-MobileServiceOpera : 60

sendRoutingInfoForGprsArg.....identifier of SendRoutingInfoForGprsArg
 DEFINED in MAP-MobileServiceOpera : 409

SendRoutingInfoForGprsArg.....type reference SEQUENCE
 DEFINED in MAP-MS-DataTypes : 1237
 USED in MAP-MobileServiceOpera : 144 409
 USED in MAP-MS-DataTypes : 98

sendRoutingInfoForGprsRes.....identifier of SendRoutingInfoForGprsRes
 DEFINED in MAP-MobileServiceOpera : 411

SendRoutingInfoForGprsRes.....type reference SEQUENCE
 DEFINED in MAP-MS-DataTypes : 1244
 USED in MAP-MobileServiceOpera : 145 411
 USED in MAP-MS-DataTypes : 99

sendRoutingInfoForLCS.....value reference SendRoutingInfoForLCS, CHOICE VALUE
 DEFINED in MAP-Protocol : 315

SendRoutingInfoForLCS.....type reference OPERATION
 DEFINED in MAP-LocationServiceOpe : 52
 USED in MAP-Protocol : 108 315
 USED in MAP-LocationServiceOpe : 14

sendRoutingInfoForSM.....value reference SendRoutingInfoForSM, CHOICE VALUE
 DEFINED in MAP-Protocol : 258

SendRoutingInfoForSM.....type reference OPERATION
 DEFINED in MAP-ShortMessageServic : 66
 USED in MAP-Protocol : 86 258
 USED in MAP-ShortMessageServic : 13

sendRoutingInfoRes.....identifier of SendRoutingInfoRes
 DEFINED in MAP-CallHandlingOperat : 85

SendRoutingInfoRes.....type reference [3] SEQUENCE
 DEFINED in MAP-CH-DataTypes : 143
 USED in MAP-CallHandlingOperat : 55 85
 USED in MAP-CH-DataTypes : 15

sendSubscriberData.....identifier of [0] NULL
 DEFINED in MAP-MS-DataTypes : 203

serviceCentreAddress.....identifier of [2] AddressString
 DEFINED in MAP-SM-DataTypes : 55

serviceCentreAddress.....identifier of AddressString
DEFINED in MAP-SM-DataTypes : 145

TAG	R4.21	Cross Reference Listing for MAP-Protocol	00-01-06	07:32:40	PAGE
serviceCentreAddress	identifier of AddressString			
	DEFINED in MAP-SM-DataTypes	:	176		
serviceCentreAddressDA	identifier of [4] AddressString			
	DEFINED in MAP-SM-DataTypes	:	135		
serviceCentreAddressOA	identifier of [4] AddressString			
	DEFINED in MAP-SM-DataTypes	:	140		
serviceGranted	identifier of Named Number, 0			
	DEFINED in MAP-MS-DataTypes	:	596		
serviceIndicator	identifier of [2] ServiceIndicator			
	DEFINED in MAP-SS-DataTypes	:	289		
ServiceIndicator	type reference BIT STRING			
	DEFINED in MAP-SS-DataTypes	:	294		
	USED in MAP-SS-DataTypes	:	289		
serviceKey	identifier of [1] ServiceKey			
	DEFINED in MAP-MS-DataTypes	:	491		
serviceKey	identifier of ServiceKey			
	DEFINED in MAP-MS-DataTypes	:	961		
serviceKey	identifier of ServiceKey			
	DEFINED in MAP-MS-DataTypes	:	1015		
ServiceKey	type reference INTEGER			
	DEFINED in MAP-MS-DataTypes	:	1022		
	USED in MAP-MS-DataTypes	:	57 491 961 1015 1145 1170 1218 1555		
serviceKey	identifier of [1] ServiceKey			
	DEFINED in MAP-MS-DataTypes	:	1145		
serviceKey	identifier of ServiceKey			
	DEFINED in MAP-MS-DataTypes	:	1170		
serviceKey	identifier of ServiceKey			
	DEFINED in MAP-MS-DataTypes	:	1218		
serviceKey	identifier of ServiceKey			
	DEFINED in MAP-MS-DataTypes	:	1555		
setReportingState	value reference SetReportingState, CHOICE VALUE			
	DEFINED in MAP-Protocol	:	232		
SetReportingState	type reference OPERATION			
	DEFINED in MAP-CallHandlingOperat	:	153		
	USED in MAP-Protocol	:	58 232		
	USED in MAP-CallHandlingOperat	:	18		
setReportingStateArg	identifier of SetReportingStateArg			
	DEFINED in MAP-CallHandlingOperat	:	155		
SetReportingStateArg	type reference SEQUENCE			
	DEFINED in MAP-CH-DataTypes	:	312		
	USED in MAP-CallHandlingOperat	:	64 155		
	USED in MAP-CH-DataTypes	:	27		
setReportingStateRes	identifier of SetReportingStateRes			
	DEFINED in MAP-CallHandlingOperat	:	157		
SetReportingStateRes	type reference SEQUENCE			
	DEFINED in MAP-CH-DataTypes	:	327		
	USED in MAP-CallHandlingOperat	:	65 157		
	USED in MAP-CH-DataTypes	:	28		
sgsn-Address	identifier of GSN-Address			
	DEFINED in MAP-MS-DataTypes	:	317		
sgsn-Address	identifier of [0] GSN-Address			
	DEFINED in MAP-MS-DataTypes	:	1245		
sgsn-Address	identifier of [1] GSN-Address			
	DEFINED in MAP-MS-DataTypes	:	1269		
sgsn-CAMEL-SubscriptionInfo	identifier of [3] SGSN-CAMEL-SubscriptionInfo			
	DEFINED in MAP-MS-DataTypes	:	466		

SGSN-CAMEL-SubscriptionInfo.....type reference SEQUENCE

80

TAG R4.21 Cross Reference Listing for MAP-Protocol 00-01-06 07:32:40 PAGE

DEFINED in MAP-MS-DataTypes : 468
 USED in MAP-MS-DataTypes : 466

sgsn-Capability.....identifier of [0] SGSN-Capability
 DEFINED in MAP-MS-DataTypes : 320

SGSN-Capability.....type reference SEQUENCE
 DEFINED in MAP-MS-DataTypes : 322
 USED in MAP-MS-DataTypes : 320

sgsn-Number.....identifier of [1] ISDN-AddressString
 DEFINED in MAP-MS-DataTypes : 244

sgsn-Number.....identifier of ISDN-AddressString
 DEFINED in MAP-MS-DataTypes : 316

sgsn-Number.....identifier of [1] ISDN-AddressString
 DEFINED in MAP-SM-DataTypes : 99

shortMessageMO-PP.....value reference TeleserviceCode, '00100010'B
 DEFINED in MAP-TS-Code : 46

shortMessageMT-PP.....value reference TeleserviceCode, '00100001'B
 DEFINED in MAP-TS-Code : 45

shortTermDenial.....value reference ShortTermDenial, CHOICE VALUE
 DEFINED in MAP-Protocol : 405

ShortTermDenial.....type reference ERROR
 DEFINED in MAP-Errors : 380
 USED in MAP-Protocol : 158 405
 USED in MAP-SupplementaryServi : 53 264
 USED in MAP-Errors : 72

shortTermDenialParam.....identifier of ShortTermDenialParam
 DEFINED in MAP-Errors : 382

ShortTermDenialParam.....type reference SEQUENCE
 DEFINED in MAP-ER-DataTypes : 328
 USED in MAP-Errors : 138 382
 USED in MAP-ER-DataTypes : 47

signalInfo.....identifier of SignalInfo
 DEFINED in MAP-CommonDataTypes : 179

SignalInfo.....type reference OCTET STRING
 DEFINED in MAP-CommonDataTypes : 186
 USED in MAP-CommonDataTypes : 22 179 205
 USED in MAP-SM-DataTypes : 33 109 115 122 128
 USED in MAP-ER-DataTypes : 72 150

signalInfo.....identifier of SignalInfo
 DEFINED in MAP-CommonDataTypes : 205

siwfsNumber.....identifier of [0] ISDN-AddressString
 DEFINED in MAP-CH-DataTypes : 297

siwfsSignallingModify.....value reference siwfsSignallingModify, CHOICE VALUE
 DEFINED in MAP-Protocol : 231

siwfsSignallingModify.....type reference OPERATION
 DEFINED in MAP-CallHandlingOperat : 141
 USED in MAP-Protocol : 57 231
 USED in MAP-CallHandlingOperat : 17

siwfsSignallingModifyArg.....identifier of siwfsSignallingModifyArg
 DEFINED in MAP-CallHandlingOperat : 143

siwfsSignallingModifyArg.....type reference SEQUENCE
 DEFINED in MAP-CH-DataTypes : 301
 USED in MAP-CallHandlingOperat : 62 143
 USED in MAP-CH-DataTypes : 25

siwfsSignallingModifyRes.....identifier of siwfsSignallingModifyRes
 DEFINED in MAP-CallHandlingOperat : 145

siwfsSignallingModifyRes.....type reference SEQUENCE
 DEFINED in MAP-CH-DataTypes : 307
 USED in MAP-CallHandlingOperat : 63 145
 USED in MAP-CH-DataTypes : 26

SMS-CAMEL-TDP-Data.....type reference SEQUENCE

81

TAG R4.21 Cross Reference Listing for MAP-Protocol 00-01-06 07:32:40 PAGE

DEFINED in MAP-MS-DataTypes : 1143
 USED in MAP-MS-DataTypes : 1139

sms-CAMEL-TDP-DataList.....identifier of [0] SMS-CAMEL-TDP-DataList
 DEFINED in MAP-MS-DataTypes : 1129

SMS-CAMEL-TDP-DataList.....type reference SEQUENCE OF
 DEFINED in MAP-MS-DataTypes : 1138
 USED in MAP-MS-DataTypes : 1129

sms-CollectedInfo.....identifier of Named Number, 1
 DEFINED in MAP-MS-DataTypes : 1153

sms-CSI.....identifier of [1] SMS-CSI
 DEFINED in MAP-MS-DataTypes : 470

sms-CSI.....identifier of [6] SMS-CSI
 DEFINED in MAP-MS-DataTypes : 942

SMS-CSI.....type reference SEQUENCE
 DEFINED in MAP-MS-DataTypes : 1128
 USED in MAP-MS-DataTypes : 470 942 1487

sms-CSI.....identifier of Named Number, 5
 DEFINED in MAP-MS-DataTypes : 1450

sms-CSI.....identifier of [9] SMS-CSI
 DEFINED in MAP-MS-DataTypes : 1487

sms-TriggerDetectionPoint.....identifier of [0] SMS-TriggerDetectionPoint
 DEFINED in MAP-MS-DataTypes : 1144

SMS-TriggerDetectionPoint.....type reference ENUMERATED
 DEFINED in MAP-MS-DataTypes : 1152
 USED in MAP-MS-DataTypes : 1144

sm-DeliveryFailure.....value reference SM-DeliveryFailure, CHOICE VALUE
 DEFINED in MAP-Protocol : 412

SM-DeliveryFailure.....type reference ERROR
 DEFINED in MAP-Errors : 398
 USED in MAP-Protocol : 153 412
 USED in MAP-ShortMessageServic : 38 91 108
 USED in MAP-Errors : 77

sm-DeliveryFailureCause.....identifier of SM-DeliveryFailureCause
 DEFINED in MAP-Errors : 400

SM-DeliveryFailureCause.....type reference SEQUENCE
 DEFINED in MAP-ER-DataTypes : 148
 USED in MAP-Errors : 108 400
 USED in MAP-ER-DataTypes : 19

sm-DeliveryOutcome.....identifier of SM-DeliveryOutcome
 DEFINED in MAP-SM-DataTypes : 146

SM-DeliveryOutcome.....type reference ENUMERATED
 DEFINED in MAP-SM-DataTypes : 163
 USED in MAP-SM-DataTypes : 26 146 157

SM-EnumeratedDeliveryFailureCause.....type reference ENUMERATED
 DEFINED in MAP-ER-DataTypes : 139
 USED in MAP-ER-DataTypes : 149

sm-EnumeratedDeliveryFailureCause.....identifier of SM-EnumeratedDeliveryFailureCause
 DEFINED in MAP-ER-DataTypes : 149

sm-RP-DA.....identifier of SM-RP-DA
 DEFINED in MAP-SM-DataTypes : 107

sm-RP-DA.....identifier of SM-RP-DA
 DEFINED in MAP-SM-DataTypes : 120

SM-RP-DA.....type reference CHOICE
 DEFINED in MAP-SM-DataTypes : 132
 USED in MAP-SM-DataTypes : 107 120

sm-RP-MTI.....identifier of [8] SM-RP-MTI
 DEFINED in MAP-SM-DataTypes : 61

```
SM-RP-MTI.....type reference INTEGER  
  DEFINED in MAP-SM-DataTypes      :      64
```

82

TAG	R4.21	Cross Reference Listing for MAP-Protocol	00-01-06 07:32:40	PAGE
	USED in MAP-SM-DataTypes	:	61	
sm-RP-OA	identifier of SM-RP-OA		
	DEFINED in MAP-SM-DataTypes	:	108	
sm-RP-OA	identifier of SM-RP-OA		
	DEFINED in MAP-SM-DataTypes	:	121	
SM-RP-OA	type reference CHOICE		
	DEFINED in MAP-SM-DataTypes	:	138	
	USED in MAP-SM-DataTypes	:	108 121	
sm-RP-PRI	identifier of [1] BOOLEAN		
	DEFINED in MAP-SM-DataTypes	:	54	
sm-RP-SMEA	identifier of [9] SM-RP-SMEA		
	DEFINED in MAP-SM-DataTypes	:	62	
SM-RP-SMEA	type reference OCTET STRING		
	DEFINED in MAP-SM-DataTypes	:	71	
	USED in MAP-SM-DataTypes	:	62	
sm-RP-UI	identifier of SignalInfo		
	DEFINED in MAP-SM-DataTypes	:	109	
sm-RP-UI	identifier of SignalInfo		
	DEFINED in MAP-SM-DataTypes	:	115	
sm-RP-UI	identifier of SignalInfo		
	DEFINED in MAP-SM-DataTypes	:	122	
sm-RP-UI	identifier of SignalInfo		
	DEFINED in MAP-SM-DataTypes	:	128	
solsaSupportIndicator	identifier of [2] NULL		
	DEFINED in MAP-MS-DataTypes	:	198	
solsaSupportIndicator	identifier of NULL		
	DEFINED in MAP-MS-DataTypes	:	323	
sres	identifier of SRES		
	DEFINED in MAP-MS-DataTypes	:	284	
SRES	type reference OCTET STRING		
	DEFINED in MAP-MS-DataTypes	:	298	
	USED in MAP-MS-DataTypes	:	284	
ss-AccessBarred	identifier of Named Number, 5		
	DEFINED in MAP-MS-DataTypes	:	624	
ss-CamelData	identifier of SS-CamelData		
	DEFINED in MAP-MS-DataTypes	:	968	
SS-CamelData	type reference SEQUENCE		
	DEFINED in MAP-MS-DataTypes	:	972	
	USED in MAP-MS-DataTypes	:	968	
ss-Code	identifier of SS-Code		
	DEFINED in MAP-SupplementaryServi	:	219	
ss-Code	identifier of SS-Code		
	DEFINED in MAP-MS-DataTypes	:	654	
ss-Code	identifier of SS-Code		
	DEFINED in MAP-MS-DataTypes	:	733	
ss-Code	identifier of SS-Code		
	DEFINED in MAP-MS-DataTypes	:	804	
ss-Code	identifier of SS-Code		
	DEFINED in MAP-MS-DataTypes	:	817	
ss-Code	identifier of SS-Code		
	DEFINED in MAP-MS-DataTypes	:	860	
ss-Code	identifier of [0] SS-Code		
	DEFINED in MAP-MS-DataTypes	:	1508	
ss-Code	identifier of [0] SS-Code		
	DEFINED in MAP-MS-DataTypes	:	1575	

ss-Code.....identifier of [0] SS-Code

83

TAG R4.21 Cross Reference Listing for MAP-Protocol 00-01-06 07:32:40 PAGE

```

DEFINED in MAP-MS-DataTypes      : 1582
ss-Code.....identifier of SS-Code
  DEFINED in MAP-SS-DataTypes    : 70
ss-Code.....identifier of SS-Code
  DEFINED in MAP-SS-DataTypes    : 86
ss-Code.....identifier of SS-Code
  DEFINED in MAP-SS-DataTypes    : 143
ss-Code.....identifier of SS-Code
  DEFINED in MAP-SS-DataTypes    : 156
ss-Code.....identifier of SS-Code
  DEFINED in MAP-SS-DataTypes    : 178
ss-Code.....identifier of [0] SS-Code
  DEFINED in MAP-SS-DataTypes    : 282
ss-Code.....identifier of [0] SS-Code
  DEFINED in MAP-SS-DataTypes    : 305
ss-Code.....identifier of [0] SS-Code
  DEFINED in MAP-SS-DataTypes    : 310
SS-Code.....type reference OCTET STRING
  DEFINED in MAP-SS-Code         : 11
  USED in MAP-SupplementaryServi : 78 219
  USED in MAP-MS-DataTypes      : 127 654 733 804 817 860 983 1508 1575
    1582
  USED in MAP-SS-DataTypes      : 62 70 86 143 156 178 246 261 282
    305 310
  USED in MAP-SS-Code          : 21 25 28 30 32 34 36 40 42
    48 50 52 54 56 58 60 63 66
    68 72 75 77 79 81 85 88 91
    94 97 100 102 105 108 110 112 115
    117 119 121 123 126 128 130 134 135
    136 137 138 139 140 141 142 143 144
    145 146 147 148 149 151 154 157 159
    161 164 166 169 171 173 176
  USED in MAP-ER-DataTypes      : 79 128
ss-Code.....identifier of [1] SS-Code
  DEFINED in MAP-ER-DataTypes    : 128
ss-CSI.....identifier of [2] SS-CSI
  DEFINED in MAP-MS-DataTypes    : 938
SS-CSI.....type reference SEQUENCE
  DEFINED in MAP-MS-DataTypes    : 967
  USED in MAP-MS-DataTypes      : 56 938 1488
ss-CSI.....identifier of Named Number, 6
  DEFINED in MAP-MS-DataTypes    : 1451
ss-CSI.....identifier of [10] SS-CSI
  DEFINED in MAP-MS-DataTypes    : 1488
ss-Data.....identifier of [3] Ext-SS-Data
  DEFINED in MAP-MS-DataTypes    : 649
ss-Data.....identifier of [3] SS-Data
  DEFINED in MAP-SS-DataTypes    : 83
SS-Data.....type reference SEQUENCE
  DEFINED in MAP-SS-DataTypes    : 155
  USED in MAP-SS-DataTypes      : 33 83
ss-ErrorStatus.....value reference SS-ErrorStatus, CHOICE VALUE
  DEFINED in MAP-Protocol        : 395
SS-ErrorStatus.....type reference ERROR
  DEFINED in MAP-Errors          : 346
  USED in MAP-Protocol          : 143 395
  USED in MAP-MobileServiceOpera : 100 282
  USED in MAP-SupplementaryServi : 41 101 118 135 155 262 279
  USED in MAP-Errors            : 63
ss-Event.....identifier of [2] SS-Code

```

DEFINED in MAP-SS-DataTypes : 261

84	TAG	R4.21	Cross Reference Listing for MAP-Protocol	00-01-06	07:32:40	PAGE
	ss-EventList	identifier of SS-EventList			
			DEFINED in MAP-MS-DataTypes	:	973	
	SS-EventList	type reference SEQUENCE OF			
			DEFINED in MAP-MS-DataTypes	:	983	
			USED in MAP-MS-DataTypes	:	973	
	ss-EventSpecification	identifier of [3] SS-EventSpecification			
			DEFINED in MAP-SS-DataTypes	:	267	
	SS-EventSpecification	type reference SEQUENCE OF			
			DEFINED in MAP-SS-DataTypes	:	276	
			USED in MAP-SS-DataTypes	:	267	
	ss-ForBS	identifier of SS-ForBS-Code			
			DEFINED in MAP-SupplementaryServi	:	106	
	ss-ForBS	identifier of SS-ForBS-Code			
			DEFINED in MAP-SupplementaryServi	:	123	
	ss-ForBS	identifier of SS-ForBS-Code			
			DEFINED in MAP-SupplementaryServi	:	143	
	ss-ForBS	identifier of SS-ForBS-Code			
			DEFINED in MAP-SupplementaryServi	:	162	
	SS-ForBS-Code	type reference SEQUENCE			
			DEFINED in MAP-SS-DataTypes	:	177	
			USED in MAP-SupplementaryServi	:	62 106 123 143 162	
			USED in MAP-MS-DataTypes	:	121 1436	
			USED in MAP-SS-DataTypes	:	18	
	ss-Incompatibility	value reference SS-Incompatibility, CHOICE VALUE			
			DEFINED in MAP-Protocol	:	398	
	SS-Incompatibility	type reference ERROR			
			DEFINED in MAP-Errors	:	363	
			USED in MAP-Protocol	:	146 398	
			USED in MAP-MobileServiceOpera	:	102 283	
			USED in MAP-SupplementaryServi	:	44 102 137 263	
			USED in MAP-Errors	:	66	
	ss-IncompatibilityCause	identifier of SS-IncompatibilityCause			
			DEFINED in MAP-Errors	:	365	
	SS-IncompatibilityCause	type reference SEQUENCE			
			DEFINED in MAP-ER-DataTypes	:	127	
			USED in MAP-Errors	:	106 365	
			USED in MAP-ER-DataTypes	:	17	
	ss-Info	identifier of SS-Info			
			DEFINED in MAP-SupplementaryServi	:	91	
	ss-Info	identifier of SS-Info			
			DEFINED in MAP-SupplementaryServi	:	108	
	ss-Info	identifier of SS-Info			
			DEFINED in MAP-SupplementaryServi	:	125	
	ss-Info	identifier of SS-Info			
			DEFINED in MAP-SupplementaryServi	:	145	
	SS-Info	type reference CHOICE			
			DEFINED in MAP-SS-DataTypes	:	80	
			USED in MAP-SupplementaryServi	:	61 91 108 125 145	
			USED in MAP-SS-DataTypes	:	15 251	
	ss-InfoFor-CSE	identifier of [0] Ext-SS-InfoFor-CSE			
			DEFINED in MAP-MS-DataTypes	:	1502	
	SS-InfoList	type reference SEQUENCE OF			
			DEFINED in MAP-SS-DataTypes	:	250	
			USED in MAP-SS-DataTypes	:	27	
	ss-InvocationNotification	value reference SS-InvocationNotification, CHOICE			
VALUE			DEFINED in MAP-Protocol	:	285	
	SS-InvocationNotification	type reference OPERATION			
			DEFINED in MAP-SupplementaryServi	:	240	

USED in MAP-Protocol	:	78	285
USED in MAP-SupplementaryServi	:	23	

TAG R4.21 Cross Reference Listing for MAP-Protocol 00-01-06 07:32:40 PAGE

ss-InvocationNotificationArg.....identifier of SS-InvocationNotificationArg
 DEFINED in MAP-SupplementaryServi : 242

SS-InvocationNotificationArg.....type reference SEQUENCE
 DEFINED in MAP-SS-DataTypes : 258
 USED in MAP-SupplementaryServi : 68 242
 USED in MAP-SS-DataTypes : 34

ss-InvocationNotificationRes.....identifier of SS-InvocationNotificationRes
 DEFINED in MAP-SupplementaryServi : 244

SS-InvocationNotificationRes.....type reference SEQUENCE
 DEFINED in MAP-SS-DataTypes : 271
 USED in MAP-SupplementaryServi : 69 244
 USED in MAP-SS-DataTypes : 35

ss-List.....identifier of [3] SS-List
 DEFINED in MAP-MS-DataTypes : 876

ss-List.....identifier of [2] SS-List
 DEFINED in MAP-MS-DataTypes : 895

ss-List.....identifier of [1] SS-List
 DEFINED in MAP-CH-DataTypes : 152

SS-List.....type reference SEQUENCE OF
 DEFINED in MAP-SS-DataTypes : 245
 USED in MAP-MS-DataTypes : 120 876 895
 USED in MAP-CH-DataTypes : 57 152
 USED in MAP-SS-DataTypes : 26

ss-NotAvailable.....value reference SS-NotAvailable, CHOICE VALUE
 DEFINED in MAP-Protocol : 396

SS-NotAvailable.....type reference ERROR
 DEFINED in MAP-Errors : 351
 USED in MAP-Protocol : 144 396
 USED in MAP-MobileServiceOpera : 101 264
 USED in MAP-SupplementaryServi : 42 173
 USED in MAP-Errors : 64

ss-NotAvailableParam.....identifier of SS-NotAvailableParam
 DEFINED in MAP-Errors : 353

SS-NotAvailableParam.....type reference SEQUENCE
 DEFINED in MAP-ER-DataTypes : 293
 USED in MAP-Errors : 148 353
 USED in MAP-ER-DataTypes : 57

ss-Status.....identifier of SS-Status
 DEFINED in MAP-Errors : 348

ss-Status.....identifier of [4] Ext-SS-Status
 DEFINED in MAP-MS-DataTypes : 664

ss-Status.....identifier of [4] Ext-SS-Status
 DEFINED in MAP-MS-DataTypes : 743

ss-Status.....identifier of [4] Ext-SS-Status
 DEFINED in MAP-MS-DataTypes : 805

ss-Status.....identifier of Ext-SS-Status
 DEFINED in MAP-MS-DataTypes : 818

ss-Status.....identifier of Ext-SS-Status
 DEFINED in MAP-MS-DataTypes : 861

ss-Status.....identifier of [2] Ext-SS-Status
 DEFINED in MAP-MS-DataTypes : 1510

ss-Status.....identifier of [4] SS-Status
 DEFINED in MAP-SS-DataTypes : 96

SS-Status.....type reference OCTET STRING
 DEFINED in MAP-SS-DataTypes : 103
 USED in MAP-Errors : 101 348
 USED in MAP-SS-DataTypes : 16 96 152 157 183 205 311
 USED in MAP-ER-DataTypes : 67 130

ss-Status.....identifier of [4] SS-Status

DEFINED in MAP-SS-DataTypes : 152

86

TAG R4.21 Cross Reference Listing for MAP-Protocol 00-01-06 07:32:40 PAGE

ss-Status.....identifier of [4] SS-Status
 DEFINED in MAP-SS-DataTypes : 157

ss-Status.....identifier of SS-Status
 DEFINED in MAP-SS-DataTypes : 183

ss-Status.....identifier of [0] SS-Status
 DEFINED in MAP-SS-DataTypes : 205

ss-Status.....identifier of [1] SS-Status
 DEFINED in MAP-SS-DataTypes : 311

ss-Status.....identifier of [4] SS-Status
 DEFINED in MAP-ER-DataTypes : 130

ss-SubscriptionOption.....identifier of SS-SubscriptionOption
 DEFINED in MAP-MS-DataTypes : 806

ss-SubscriptionOption.....identifier of SS-SubscriptionOption
 DEFINED in MAP-SS-DataTypes : 158

SS-SubscriptionOption.....type reference CHOICE
 DEFINED in MAP-SS-DataTypes : 164
 USED in MAP-MS-DataTypes : 119 806
 USED in MAP-SS-DataTypes : 17 158

ss-SubscriptionViolation.....value reference SS-SubscriptionViolation, CHOICE
 VALUE
 DEFINED in MAP-Protocol : 397

SS-SubscriptionViolation.....type reference ERROR
 DEFINED in MAP-Errors : 357
 USED in MAP-Protocol : 145 397
 USED in MAP-MobileServiceOpera : 103 281
 USED in MAP-SupplementaryServi : 43 136 156 227
 USED in MAP-Errors : 65

ss-SubscriptionViolationParam.....identifier of SS-SubscriptionViolationParam
 DEFINED in MAP-Errors : 359

SS-SubscriptionViolationParam.....type reference SEQUENCE
 DEFINED in MAP-ER-DataTypes : 297
 USED in MAP-Errors : 149 359
 USED in MAP-ER-DataTypes : 58

startMonitoring.....identifier of Named Number, 1
 DEFINED in MAP-CH-DataTypes : 321

statusReport.....value reference StatusReport, CHOICE VALUE
 DEFINED in MAP-Protocol : 233

StatusReport.....type reference OPERATION
 DEFINED in MAP-CallHandlingOperat : 167
 USED in MAP-Protocol : 59 233
 USED in MAP-CallHandlingOperat : 19

statusReportArg.....identifier of StatusReportArg
 DEFINED in MAP-CallHandlingOperat : 169

StatusReportArg.....type reference SEQUENCE
 DEFINED in MAP-CH-DataTypes : 342
 USED in MAP-CallHandlingOperat : 66 169
 USED in MAP-CH-DataTypes : 29

statusReportRes.....identifier of StatusReportRes
 DEFINED in MAP-CallHandlingOperat : 171

StatusReportRes.....type reference SEQUENCE
 DEFINED in MAP-CH-DataTypes : 378
 USED in MAP-CallHandlingOperat : 67 171
 USED in MAP-CH-DataTypes : 30

stopMonitoring.....identifier of Named Number, 0
 DEFINED in MAP-CH-DataTypes : 320

storedMSISDN.....identifier of ISDN-AddressString
 DEFINED in MAP-SM-DataTypes : 169

storedMSISDN.....identifier of ISDN-AddressString
 DEFINED in MAP-SM-DataTypes : 180

subBusyForMT-SMS-Param.....identifier of SubBusyForMT-SMS-Param
DEFINED in MAP-Errors : 395

87

TAG R4.21 Cross Reference Listing for MAP-Protocol 00-01-06 07:32:40 PAGE

```

SubBusyForMT-SMS-Param.....type reference SEQUENCE
  DEFINED in MAP-ER-DataTypes      : 305
  USED in MAP-Errors                : 132 395
  USED in MAP-ER-DataTypes         : 40

subscriberBusyForMT-SMS.....value reference SubscriberBusyForMT-SMS, CHOICE VALUE
  DEFINED in MAP-Protocol           : 411

SubscriberBusyForMT-SMS.....type reference ERROR
  DEFINED in MAP-Errors             : 393
  USED in MAP-Protocol              : 152 411
  USED in MAP-ShortMessageServic   : 37 107
  USED in MAP-Errors                : 76

SubscriberData.....type reference SEQUENCE
  DEFINED in MAP-MS-DataTypes       : 573
  USED in MAP-MS-DataTypes         : 47 398

subscriberDataStored.....identifier of [1] AgeIndicator
  DEFINED in MAP-MS-DataTypes       : 204

SubscriberId.....type reference CHOICE
  DEFINED in MAP-CommonDataTypes    : 269
  USED in MAP-CommonDataTypes      : 30

subscriberIdentity.....identifier of [0] SubscriberIdentity
  DEFINED in MAP-MS-DataTypes       : 1404

subscriberIdentity.....identifier of [0] SubscriberIdentity
  DEFINED in MAP-MS-DataTypes       : 1419

subscriberIdentity.....identifier of [0] SubscriberIdentity
  DEFINED in MAP-MS-DataTypes       : 1494

SubscriberIdentity.....type reference CHOICE
  DEFINED in MAP-CommonDataTypes    : 326
  USED in MAP-MS-DataTypes         : 157 1404 1419 1494
  USED in MAP-CommonDataTypes      : 39
  USED in MAP-LCS-DataTypes        : 31 54 59

subscriberInfo.....identifier of SubscriberInfo
  DEFINED in MAP-MS-DataTypes       : 1336

SubscriberInfo.....type reference SEQUENCE
  DEFINED in MAP-MS-DataTypes       : 1340
  USED in MAP-MS-DataTypes         : 79 1336 1411
  USED in MAP-CH-DataTypes         : 40 151

subscriberInfo.....identifier of SubscriberInfo
  DEFINED in MAP-MS-DataTypes       : 1411

subscriberInfo.....identifier of [7] SubscriberInfo
  DEFINED in MAP-CH-DataTypes       : 151

subscriberLocationReport.....value reference SubscriberLocationReport, CHOICE
VALUE
  DEFINED in MAP-Protocol           : 316

SubscriberLocationReport.....type reference OPERATION
  DEFINED in MAP-LocationServiceOpe : 84
  USED in MAP-Protocol              : 109 316
  USED in MAP-LocationServiceOpe   : 15

subscriberLocationReport-Arg.....identifier of SubscriberLocationReport-Arg
  DEFINED in MAP-LocationServiceOpe : 86

SubscriberLocationReport-Arg.....type reference SEQUENCE
  DEFINED in MAP-LCS-DataTypes       : 216
  USED in MAP-LocationServiceOpe    : 45 86
  USED in MAP-LCS-DataTypes         : 15

subscriberLocationReport-Res.....identifier of SubscriberLocationReport-Res
  DEFINED in MAP-LocationServiceOpe : 88

SubscriberLocationReport-Res.....type reference SEQUENCE
  DEFINED in MAP-LCS-DataTypes       : 241
  USED in MAP-LocationServiceOpe    : 46 88
  USED in MAP-LCS-DataTypes         : 16
    
```

subscriberNotMemberOfCUG.....identifier of Named Number, 1
DEFINED in MAP-ER-DataTypes : 123

88 TAG R4.21 Cross Reference Listing for MAP-Protocol 00-01-06 07:32:40 PAGE

subscriberNotSC-Subscriber.....identifier of Named Number, 6
 DEFINED in MAP-ER-DataTypes : 146

subscriberState.....identifier of [1] SubscriberState
 DEFINED in MAP-MS-DataTypes : 1342

subscriberState.....identifier of [1] NULL
 DEFINED in MAP-MS-DataTypes : 1348

SubscriberState.....type reference CHOICE
 DEFINED in MAP-MS-DataTypes : 1389
 USED in MAP-MS-DataTypes : 81 1342

subscriberStatus.....identifier of [3] SubscriberStatus
 DEFINED in MAP-MS-DataTypes : 576

SubscriberStatus.....type reference ENUMERATED
 DEFINED in MAP-MS-DataTypes : 595
 USED in MAP-MS-DataTypes : 49 576

subscriptionWithdraw.....identifier of Named Number, 1
 DEFINED in MAP-MS-DataTypes : 232

subsequentHandoverFailure.....value reference SubsequentHandoverFailure, CHOICE
 VALUE DEFINED in MAP-Protocol : 357

SubsequentHandoverFailure.....type reference ERROR
 DEFINED in MAP-Errors : 254
 USED in MAP-Protocol : 130 357
 USED in MAP-MobileServiceOpera : 91 334
 USED in MAP-Errors : 37

success.....identifier of Named Number, 0
 DEFINED in MAP-CH-DataTypes : 369

successfulTransfer.....identifier of Named Number, 2
 DEFINED in MAP-SM-DataTypes : 166

SuperChargerInfo.....type reference CHOICE
 DEFINED in MAP-MS-DataTypes : 202
 USED in MAP-MS-DataTypes : 200 326

superChargerSupportedInHLR.....identifier of [27] AgeIndicator
 DEFINED in MAP-MS-DataTypes : 411

superChargerSupportedInServingNetworkEntidentifier of [3] SuperChargerInfo
 DEFINED in MAP-MS-DataTypes : 200

superChargerSupportedInServingNetworkEntidentifier of [2] SuperChargerInfo
 DEFINED in MAP-MS-DataTypes : 326

supportedCamelPhases.....identifier of [0] SupportedCamelPhases
 DEFINED in MAP-MS-DataTypes : 195

supportedCamelPhases.....identifier of [4] SupportedCamelPhases
 DEFINED in MAP-MS-DataTypes : 328

supportedCamelPhases.....identifier of [6] SupportedCamelPhases
 DEFINED in MAP-MS-DataTypes : 880

SupportedCamelPhases.....type reference BIT STRING
 DEFINED in MAP-MS-DataTypes : 1121
 USED in MAP-MS-DataTypes : 61 195 328 880 1430 1431 1561
 USED in MAP-CH-DataTypes : 41 210 253

supportedCAMELPhases.....identifier of [5] SupportedCamelPhases
 DEFINED in MAP-MS-DataTypes : 1561

supportedCamelPhases.....identifier of SupportedCamelPhases
 DEFINED in MAP-CH-DataTypes : 253

supportedCamelPhasesInGMSC.....identifier of [15] SupportedCamelPhases
 DEFINED in MAP-CH-DataTypes : 210

supportedCCBS-Phase.....identifier of [16] SupportedCCBS-Phase
 DEFINED in MAP-CH-DataTypes : 111

SupportedCCBS-Phase.....type reference INTEGER
 DEFINED in MAP-CH-DataTypes : 132

USED in MAP-CH-DataTypes : 111
supportedSGSN-CAMEL-Phases.....identifier of [6] SupportedCamelPhases

89

TAG R4.21 Cross Reference Listing for MAP-Protocol 00-01-06 07:32:40 PAGE

DEFINED in MAP-MS-DataTypes : 1431

supportedSGSN-CAMEL-Phases.....identifier of [5] NULL
 DEFINED in MAP-MS-DataTypes : 1440

supportedVLR-CAMEL-Phases.....identifier of [5] SupportedCamelPhases
 DEFINED in MAP-MS-DataTypes : 1430

supportedVLR-CAMEL-Phases.....identifier of [4] NULL
 DEFINED in MAP-MS-DataTypes : 1439

suppressionOfAnnouncement.....identifier of [12] SuppressionOfAnnouncement
 DEFINED in MAP-CH-DataTypes : 106

SuppressionOfAnnouncement.....type reference NULL
 DEFINED in MAP-CH-DataTypes : 117
 USED in MAP-CH-DataTypes : 21 106 202

suppressionOfAnnouncement.....identifier of [7] SuppressionOfAnnouncement
 DEFINED in MAP-CH-DataTypes : 202

suppress-T-CSI.....identifier of NULL
 DEFINED in MAP-CH-DataTypes : 254

systemFailure.....value reference SystemFailure, CHOICE VALUE
 DEFINED in MAP-Protocol : 325

SystemFailure.....type reference ERROR
 DEFINED in MAP-Errors : 162
 USED in MAP-Protocol : 115 325
 USED in MAP-MobileServiceOpera : 81 174 218 242 307 345 358 400 414
 428 442
 USED in MAP-OperationAndMainte : 23 57 71
 USED in MAP-CallHandlingOperat : 30 87 110 139 151 160 175 189 202
 215
 USED in MAP-SupplementaryServi : 33 94 111 128 148 166 181 194 208
 223 257 274
 USED in MAP-ShortMessageServic : 27 72 88 100 128
 USED in MAP-Group-Call-Operati : 24 52
 USED in MAP-LocationServiceOpe : 23 58 72 90
 USED in MAP-Errors : 14

systemFailureParam.....identifier of SystemFailureParam
 DEFINED in MAP-Errors : 164

SystemFailureParam.....type reference CHOICE
 DEFINED in MAP-ER-DataTypes : 168
 USED in MAP-Errors : 109 164
 USED in MAP-ER-DataTypes : 20

targetCellId.....identifier of GlobalCellId
 DEFINED in MAP-MS-DataTypes : 341

targetCellId.....identifier of GlobalCellId
 DEFINED in MAP-MS-DataTypes : 352

targetMS.....identifier of [1] SubscriberIdentity
 DEFINED in MAP-LCS-DataTypes : 54

targetMS.....identifier of [0] SubscriberIdentity
 DEFINED in MAP-LCS-DataTypes : 59

targetMSC-Number.....identifier of ISDN-AddressString
 DEFINED in MAP-MS-DataTypes : 353

targetMSsubscribedService.....identifier of Named Number, 4
 DEFINED in MAP-CommonDataTypes : 341

TBCD-STRING.....type reference OCTET STRING
 DEFINED in MAP-CommonDataTypes : 78
 USED in MAP-CommonDataTypes : 250 263 273

tBusy.....identifier of Named Number, 13
 DEFINED in MAP-MS-DataTypes : 1227

TCAPMessages.....module reference
 DEFINED in TCAPMessages : 1
 USED in MAP-MobileServiceOpera : 78
 USED in MAP-OperationAndMainte : 20
 USED in MAP-CallHandlingOperat : 27

USED in MAP-SupplementaryServi : 30
USED in MAP-ShortMessageServic : 24

90 TAG R4.21 Cross Reference Listing for MAP-Protocol 00-01-06 07:32:40 PAGE

USED in MAP-Group-Call-Operati : 21
 USED in MAP-LocationServiceOpe : 20
 USED in MAP-Errors : 98

telephony.....value reference TeleserviceCode, '00010001'B
 DEFINED in MAP-TS-Code : 41

teleservice.....identifier of [3] TeleserviceCode
 DEFINED in MAP-CommonDataTypes : 381

teleservice.....identifier of Ext-TeleserviceCode
 DEFINED in MAP-GR-DataTypes : 50

TeleserviceCode.....type reference OCTET STRING
 DEFINED in MAP-TS-Code : 11
 USED in MAP-CommonDataTypes : 57 381
 USED in MAP-TS-Code : 38 40 41 42 44 45 46 48 49
 50 51 55 58 67 69 70 72 73
 74 75 76 77 78 79 80 81 82
 83 84 85 86 87

teleserviceList.....identifier of [6] TeleserviceList
 DEFINED in MAP-MS-DataTypes : 580

TeleserviceList.....type reference SEQUENCE OF
 DEFINED in MAP-MS-DataTypes : 604
 USED in MAP-MS-DataTypes : 580 874

teleserviceList.....identifier of [1] TeleserviceList
 DEFINED in MAP-MS-DataTypes : 874

teleserviceNotProvisioned.....value reference TeleserviceNotProvisioned, CHOICE
 VALUE DEFINED in MAP-Protocol : 349

TeleserviceNotProvisioned.....type reference ERROR
 DEFINED in MAP-Errors : 243
 USED in MAP-Protocol : 128 349
 USED in MAP-MobileServiceOpera : 97 261 278
 USED in MAP-CallHandlingOperat : 38 95
 USED in MAP-SupplementaryServi : 38 98 115 132 152 170
 USED in MAP-ShortMessageServic : 35 77
 USED in MAP-Errors : 33

teleservNotProvParam.....identifier of TeleservNotProvParam
 DEFINED in MAP-Errors : 245

TeleservNotProvParam.....type reference SEQUENCE
 DEFINED in MAP-ER-DataTypes : 229
 USED in MAP-Errors : 120 245
 USED in MAP-ER-DataTypes : 31

temporaryDefaultAllowed.....identifier of Named Number, 2
 DEFINED in MAP-SS-DataTypes : 171

temporaryDefaultRestricted.....identifier of Named Number, 1
 DEFINED in MAP-SS-DataTypes : 170

termAttemptAuthorized.....identifier of Named Number, 12
 DEFINED in MAP-MS-DataTypes : 1225

terminateAllCallActivities.....identifier of Named Number, 1
 DEFINED in MAP-CH-DataTypes : 436

terminateCallActivityReferred.....identifier of Named Number, 0
 DEFINED in MAP-CH-DataTypes : 435

tif-CSI.....identifier of [3] NULL
 DEFINED in MAP-MS-DataTypes : 940

tif-CSI.....identifier of Named Number, 3
 DEFINED in MAP-MS-DataTypes : 1448

tif-CSI.....identifier of [6] NULL
 DEFINED in MAP-MS-DataTypes : 1484

tif-CSI-NotificationToCSE.....identifier of [7] NULL
 DEFINED in MAP-MS-DataTypes : 1485

tmsi.....identifier of TMSI

DEFINED in MAP-MS-DataTypes : 255
TMSI.....type reference OCTET STRING

91

TAG R4.21 Cross Reference Listing for MAP-Protocol 00-01-06 07:32:40 PAGE

DEFINED in MAP-CommonDataTypes : 267
 USED in MAP-MS-DataTypes : 148 255
 USED in MAP-CommonDataTypes : 28 271

tmsi.....identifier of [1] TMSI
 DEFINED in MAP-CommonDataTypes : 271

tNoAnswer.....identifier of Named Number, 14
 DEFINED in MAP-MS-DataTypes : 1228

tooManyZoneCodes.....identifier of Named Number, 1
 DEFINED in MAP-MS-DataTypes : 886

traceReference.....identifier of [1] TraceReference
 DEFINED in MAP-OM-DataTypes : 38

TraceReference.....type reference OCTET STRING
 DEFINED in MAP-OM-DataTypes : 44
 USED in MAP-OM-DataTypes : 38 56

traceReference.....identifier of [1] TraceReference
 DEFINED in MAP-OM-DataTypes : 56

traceType.....identifier of [2] TraceType
 DEFINED in MAP-OM-DataTypes : 39

TraceType.....type reference INTEGER
 DEFINED in MAP-OM-DataTypes : 46
 USED in MAP-OM-DataTypes : 39

tracingBufferFull.....value reference TracingBufferFull, CHOICE VALUE
 DEFINED in MAP-Protocol : 363

TracingBufferFull.....type reference ERROR
 DEFINED in MAP-Errors : 259
 USED in MAP-Protocol : 131 363
 USED in MAP-OperationAndMainte : 29 62
 USED in MAP-Errors : 40

tracingBufferFullParam.....identifier of TracingBufferFullParam
 DEFINED in MAP-Errors : 261

TracingBufferFullParam.....type reference SEQUENCE
 DEFINED in MAP-ER-DataTypes : 233
 USED in MAP-Errors : 121 261
 USED in MAP-ER-DataTypes : 32

TransactionID.....type reference OCTET STRING
 DEFINED in TCAPMessages : 100
 USED in TCAPMessages : 47 97 98

transferToThirdParty.....value reference SS-Code, '1100011'B
 DEFINED in MAP-SS-Code : 176

translatedB-Number.....identifier of [3] ISDN-AddressString
 DEFINED in MAP-CH-DataTypes : 386

translatedB-Number.....identifier of [1] ISDN-AddressString
 DEFINED in MAP-SS-DataTypes : 288

tripletList.....identifier of [0] TripletList
 DEFINED in MAP-MS-DataTypes : 273

TripletList.....type reference SEQUENCE OF
 DEFINED in MAP-MS-DataTypes : 276
 USED in MAP-MS-DataTypes : 273

typeOfModification.....identifier of TypeOfModification
 DEFINED in MAP-MS-DataTypes : 1534

TypeOfModification.....type reference ENUMERATED
 DEFINED in MAP-MS-DataTypes : 1542
 USED in MAP-MS-DataTypes : 1534

T-BcsmCamelTDPData.....type reference SEQUENCE
 DEFINED in MAP-MS-DataTypes : 1216
 USED in MAP-MS-DataTypes : 1209

t-BcsmCamelTDPDataList.....identifier of T-BcsmCamelTDPDataList
 DEFINED in MAP-MS-DataTypes : 1198

T-BcsmCamelTDPDataList.....type reference SEQUENCE OF

92

TAG R4.21 Cross Reference Listing for MAP-Protocol 00-01-06 07:32:40 PAGE

DEFINED in MAP-MS-DataTypes : 1208
 USED in MAP-MS-DataTypes : 1198

t-BcsmTriggerDetectionPoint.....identifier of T-BcsmTriggerDetectionPoint
 DEFINED in MAP-MS-DataTypes : 1217

T-BcsmTriggerDetectionPoint.....type reference ENUMERATED
 DEFINED in MAP-MS-DataTypes : 1224
 USED in MAP-MS-DataTypes : 69 1052 1217

T-BCSM-CAMEL-TDP-Criteria.....type reference SEQUENCE
 DEFINED in MAP-MS-DataTypes : 1051
 USED in MAP-MS-DataTypes : 1040

t-BCSM-CAMEL-TDP-CriteriaList.....identifier of [8] T-BCSM-CAMEL-TDP-CriteriaList
 DEFINED in MAP-MS-DataTypes : 944

T-BCSM-CAMEL-TDP-CriteriaList.....type reference SEQUENCE OF
 DEFINED in MAP-MS-DataTypes : 1039
 USED in MAP-MS-DataTypes : 55 944 1481 1483
 USED in MAP-CH-DataTypes : 46 274

t-BCSM-CAMEL-TDP-CriteriaList.....identifier of [3] T-BCSM-CAMEL-TDP-CriteriaList
 DEFINED in MAP-MS-DataTypes : 1481

t-BCSM-CAMEL-TDP-CriteriaList.....identifier of [4] T-BCSM-CAMEL-TDP-CriteriaList
 DEFINED in MAP-CH-DataTypes : 274

t-BCSM-TriggerDetectionPoint.....identifier of T-BcsmTriggerDetectionPoint
 DEFINED in MAP-MS-DataTypes : 1052

t-CauseValueCriteria.....identifier of [1] T-CauseValueCriteria
 DEFINED in MAP-MS-DataTypes : 1054

T-CauseValueCriteria.....type reference SEQUENCE OF
 DEFINED in MAP-MS-DataTypes : 1095
 USED in MAP-MS-DataTypes : 1054

T-CSI.....type reference SEQUENCE
 DEFINED in MAP-MS-DataTypes : 1197
 USED in MAP-MS-DataTypes : 68 943 1480 1482
 USED in MAP-CH-DataTypes : 49 269

t-CSI.....identifier of Named Number, 1
 DEFINED in MAP-MS-DataTypes : 1446

t-CSI.....identifier of [2] T-CSI
 DEFINED in MAP-MS-DataTypes : 1480

t-CSI.....identifier of [0] T-CSI
 DEFINED in MAP-CH-DataTypes : 269

uubFromBusyMS.....identifier of Named Number, 5
 DEFINED in MAP-CH-DataTypes : 403

uubFromFreeMS.....identifier of Named Number, 4
 DEFINED in MAP-CH-DataTypes : 402

unauthorisedMessageOriginator.....identifier of [1] NULL
 DEFINED in MAP-ER-DataTypes : 114

unauthorizedLCSCClient.....value reference UnauthorizedLCSCClient, CHOICE VALUE
 DEFINED in MAP-Protocol : 419

UnauthorizedLCSCClient.....type reference ERROR
 DEFINED in MAP-Errors : 426
 USED in MAP-Protocol : 162 419
 USED in MAP-LocationServiceOpe : 30 81
 USED in MAP-Errors : 86

unauthorizedLCSCClient-Diagnostic.....identifier of [0] UnauthorizedLCSCClient-Diagnostic
 DEFINED in MAP-ER-DataTypes : 339

UnauthorizedLCSCClient-Diagnostic.....type reference ENUMERATED
 DEFINED in MAP-ER-DataTypes : 343
 USED in MAP-ER-DataTypes : 339

unauthorizedLCSCClient-Param.....identifier of UnauthorizedLCSCClient-Param
 DEFINED in MAP-Errors : 428

UnauthorizedLCSCClient-Param.....type reference SEQUENCE
DEFINED in MAP-ER-DataTypes : 338

93 TAG R4.21 Cross Reference Listing for MAP-Protocol 00-01-06 07:32:40 PAGE

USED in MAP-Errors : 141 428
 USED in MAP-ER-DataTypes : 50

unauthorizedRequestingNetwork.....value reference UnauthorizedRequestingNetwork, CHOICE
 VALUE DEFINED in MAP-Protocol : 418

UnauthorizedRequestingNetwork.....type reference ERROR
 DEFINED in MAP-Errors : 421
 USED in MAP-Protocol : 161 418
 USED in MAP-LocationServiceOpe : 29 64 80 95
 USED in MAP-Errors : 85

unauthorizedRequestingNetwork-Param....identifier of UnauthorizedRequestingNetwork-Param
 DEFINED in MAP-Errors : 423

UnauthorizedRequestingNetwork-Param....type reference SEQUENCE
 DEFINED in MAP-ER-DataTypes : 334
 USED in MAP-Errors : 140 423
 USED in MAP-ER-DataTypes : 49

undetermined.....identifier of Named Number, 0
 DEFINED in MAP-ER-DataTypes : 134

unexpectedDataParam.....identifier of UnexpectedDataParam
 DEFINED in MAP-Errors : 175

UnexpectedDataParam.....type reference SEQUENCE
 DEFINED in MAP-ER-DataTypes : 184
 USED in MAP-Errors : 111 175
 USED in MAP-ER-DataTypes : 22

unexpectedDataValue.....value reference UnexpectedDataValue, CHOICE VALUE
 DEFINED in MAP-Protocol : 327

UnexpectedDataValue.....type reference ERROR
 DEFINED in MAP-Errors : 173
 USED in MAP-Protocol : 117 327
 USED in MAP-MobileServiceOpera : 83 176 188 198 219 232 245 258 275
 295 309 331 347 372 383 402 416 430
 444 455
 USED in MAP-OperationAndMainte : 25 59 73 84
 USED in MAP-CallHandlingOperat : 32 89 112 127 138 150 162 176 185
 199 212
 USED in MAP-SupplementaryServi : 35 96 113 130 150 168 183 196 210
 225 248 259 276
 USED in MAP-ShortMessageServic : 29 74 89 102 119 130 144
 USED in MAP-Group-Call-Operati : 25 54
 USED in MAP-LocationServiceOpe : 25 60 74 93
 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

unidentifiedSubParam.....identifier of UnidentifiedSubParam
 DEFINED in MAP-Errors : 212

UnidentifiedSubParam.....type reference SEQUENCE
 DEFINED in MAP-ER-DataTypes : 213
 USED in MAP-Errors : 115 212
 USED in MAP-ER-DataTypes : 27

unidentifiedSubscriber.....value reference UnidentifiedSubscriber, CHOICE VALUE
 DEFINED in MAP-Protocol : 338

UnidentifiedSubscriber.....type reference ERROR
 DEFINED in MAP-Errors : 210
 USED in MAP-Protocol : 122 338
 USED in MAP-MobileServiceOpera : 86 208 373 384
 USED in MAP-OperationAndMainte : 28 61 75
 USED in MAP-CallHandlingOperat : 49 161
 USED in MAP-ShortMessageServic : 32 104
 USED in MAP-LocationServiceOpe : 34 76
 USED in MAP-Errors : 25

unidirectional.....identifier of [APPLICATION 1] IMPLICIT Unidirectional
 DEFINED in TCAPMessages : 52

Unidirectional.....type reference SEQUENCE
DEFINED in TCAPMessages : 58

94

TAG R4.21 Cross Reference Listing for MAP-Protocol 00-01-06 07:32:40 PAGE

USED in TCAPMessages : 52

universal.....value reference SS-Code, '10110001'B
 DEFINED in MAP-SS-Code : 159

unknownAlphabet.....value reference UnknownAlphabet, CHOICE VALUE
 DEFINED in MAP-Protocol : 399

UnknownAlphabet.....type reference ERROR
 DEFINED in MAP-Errors : 368
 USED in MAP-Protocol : 147 399
 USED in MAP-SupplementaryServi : 48 184 200 214
 USED in MAP-Errors : 67

unknownEquipment.....value reference UnknownEquipment, CHOICE VALUE
 DEFINED in MAP-Protocol : 339

UnknownEquipment.....type reference ERROR
 DEFINED in MAP-Errors : 216
 USED in MAP-Protocol : 123 339
 USED in MAP-MobileServiceOpera : 87 360
 USED in MAP-Errors : 26

unknownMSC.....value reference UnknownMSC, CHOICE VALUE
 DEFINED in MAP-Protocol : 337

UnknownMSC.....type reference ERROR
 DEFINED in MAP-Errors : 208
 USED in MAP-Protocol : 121 337
 USED in MAP-MobileServiceOpera : 85 333
 USED in MAP-Errors : 24

VALUE unknownOrUnreachableLCSClient.....value reference UnknownOrUnreachableLCSClient, CHOICE
 DEFINED in MAP-Protocol : 421

UnknownOrUnreachableLCSClient.....type reference ERROR
 DEFINED in MAP-Errors : 436
 USED in MAP-Protocol : 164 421
 USED in MAP-LocationServiceOpe : 33 96
 USED in MAP-Errors : 88

unknownOrUnreachableLCSClient-Param.....identifier of UnknownOrUnreachableLCSClient-Param
 DEFINED in MAP-Errors : 438

UnknownOrUnreachableLCSClient-Param.....type reference SEQUENCE
 DEFINED in MAP-ER-DataTypes : 370
 USED in MAP-Errors : 143 438
 USED in MAP-ER-DataTypes : 52

unknownServiceCentre.....identifier of Named Number, 3
 DEFINED in MAP-ER-DataTypes : 143

unknownSubscriber.....value reference UnknownSubscriber, CHOICE VALUE
 DEFINED in MAP-Protocol : 335

UnknownSubscriber.....type reference ERROR
 DEFINED in MAP-Errors : 197
 USED in MAP-Protocol : 119 335
 USED in MAP-MobileServiceOpera : 84 177 199 220 246 259 276 296 348
 403 417 431 445 456
 USED in MAP-OperationAndMainte : 27 85
 USED in MAP-CallHandlingOperat : 35 92 174 201 214
 USED in MAP-SupplementaryServi : 36 249
 USED in MAP-ShortMessageServic : 31 76 120 146
 USED in MAP-LocationServiceOpe : 27 62 94
 USED in MAP-Errors : 22

unknownSubscriberDiagnostic.....identifier of UnknownSubscriberDiagnostic
 DEFINED in MAP-ER-DataTypes : 199

UnknownSubscriberDiagnostic.....type reference ENUMERATED
 DEFINED in MAP-ER-DataTypes : 201
 USED in MAP-ER-DataTypes : 199

unknownSubscriberParam.....identifier of UnknownSubscriberParam
 DEFINED in MAP-Errors : 199

UnknownSubscriberParam.....type reference SEQUENCE
 DEFINED in MAP-ER-DataTypes : 196

USED in MAP-Errors	:	113	199
USED in MAP-ER-DataTypes	:	25	

95

TAG R4.21 Cross Reference Listing for MAP-Protocol 00-01-06 07:32:40 PAGE

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 : 249

UnstructuredSS-Notify.....type reference OPERATION
 DEFINED in MAP-SupplementaryServi : 203
 USED in MAP-Protocol : 75 249
 USED in MAP-SupplementaryServi : 20

unstructuredSS-Request.....value reference UnstructuredSS-Request, CHOICE VALUE
 DEFINED in MAP-Protocol : 248

UnstructuredSS-Request.....type reference OPERATION
 DEFINED in MAP-SupplementaryServi : 187
 USED in MAP-Protocol : 74 248
 USED in MAP-SupplementaryServi : 19

updateGprsLocation.....value reference UpdateGprsLocation, CHOICE VALUE
 DEFINED in MAP-Protocol : 298

UpdateGprsLocation.....type reference OPERATION
 DEFINED in MAP-MobileServiceOpera : 212
 USED in MAP-Protocol : 16 298
 USED in MAP-MobileServiceOpera : 21

updateGprsLocationArg.....identifier of UpdateGprsLocationArg
 DEFINED in MAP-MobileServiceOpera : 214

UpdateGprsLocationArg.....type reference SEQUENCE
 DEFINED in MAP-MS-DataTypes : 314
 USED in MAP-MobileServiceOpera : 119 214
 USED in MAP-MS-DataTypes : 24

updateGprsLocationRes.....identifier of UpdateGprsLocationRes
 DEFINED in MAP-MobileServiceOpera : 216

UpdateGprsLocationRes.....type reference SEQUENCE
 DEFINED in MAP-MS-DataTypes : 333
 USED in MAP-MobileServiceOpera : 120 216
 USED in MAP-MS-DataTypes : 25

updateLocation.....value reference UpdateLocation, CHOICE VALUE
 DEFINED in MAP-Protocol : 178

UpdateLocation.....type reference OPERATION
 DEFINED in MAP-MobileServiceOpera : 168
 USED in MAP-Protocol : 12 178
 USED in MAP-MobileServiceOpera : 15

updateLocationArg.....identifier of UpdateLocationArg
 DEFINED in MAP-MobileServiceOpera : 170

UpdateLocationArg.....type reference SEQUENCE
 DEFINED in MAP-MS-DataTypes : 184
 USED in MAP-MobileServiceOpera : 111 170
 USED in MAP-MS-DataTypes : 16

updateLocationRes.....identifier of UpdateLocationRes
DEFINED in MAP-MobileServiceOpera : 172

TAG R4.21 Cross Reference Listing for MAP-Protocol 00-01-06 07:32:40 PAGE

UpdateLocationRes.....type reference SEQUENCE
 DEFINED in MAP-MS-DataTypes : 217
 USED in MAP-MobileServiceOpera : 112 172
 USED in MAP-MS-DataTypes : 17

updateProcedure.....identifier of Named Number, 0
 DEFINED in MAP-MS-DataTypes : 231

uplinkFree.....identifier of [3] NULL
 DEFINED in MAP-GR-DataTypes : 57

uplinkRejectCommand.....identifier of [2] NULL
 DEFINED in MAP-GR-DataTypes : 79

uplinkReleaseCommand.....identifier of [4] NULL
 DEFINED in MAP-GR-DataTypes : 81

uplinkReleaseIndication.....identifier of [1] NULL
 DEFINED in MAP-GR-DataTypes : 78

uplinkReleaseIndication.....identifier of [1] NULL
 DEFINED in MAP-GR-DataTypes : 87

uplinkRequest.....identifier of [0] NULL
 DEFINED in MAP-GR-DataTypes : 86

uplinkRequestAck.....identifier of [0] NULL
 DEFINED in MAP-GR-DataTypes : 77

uplinkSeizedCommand.....identifier of [3] NULL
 DEFINED in MAP-GR-DataTypes : 80

ussd-Arg.....identifier of USSD-Arg
 DEFINED in MAP-SupplementaryServi : 177

ussd-Arg.....identifier of USSD-Arg
 DEFINED in MAP-SupplementaryServi : 189

ussd-Arg.....identifier of USSD-Arg
 DEFINED in MAP-SupplementaryServi : 205

USSD-Arg.....type reference SEQUENCE
 DEFINED in MAP-SS-DataTypes : 210
 USED in MAP-SupplementaryServi : 64 177 189 205
 USED in MAP-SS-DataTypes : 20

ussd-Busy.....value reference USSD-Busy, CHOICE VALUE
 DEFINED in MAP-Protocol : 400

USSD-Busy.....type reference ERROR
 DEFINED in MAP-Errors : 370
 USED in MAP-Protocol : 148 400
 USED in MAP-SupplementaryServi : 49 201 215
 USED in MAP-Errors : 68

ussd-DataCodingScheme.....identifier of USSD-DataCodingScheme
 DEFINED in MAP-SS-DataTypes : 211

ussd-DataCodingScheme.....identifier of USSD-DataCodingScheme
 DEFINED in MAP-SS-DataTypes : 218

USSD-DataCodingScheme.....type reference OCTET STRING
 DEFINED in MAP-SS-DataTypes : 222
 USED in MAP-SS-DataTypes : 22 211 218
 USED in MAP-LCS-DataTypes : 44 119

ussd-Res.....identifier of USSD-Res
 DEFINED in MAP-SupplementaryServi : 179

ussd-Res.....identifier of USSD-Res
 DEFINED in MAP-SupplementaryServi : 191

USSD-Res.....type reference SEQUENCE
 DEFINED in MAP-SS-DataTypes : 217
 USED in MAP-SupplementaryServi : 65 179 191
 USED in MAP-SS-DataTypes : 21

ussd-String.....identifier of USSD-String
 DEFINED in MAP-SS-DataTypes : 212

ussd-String.....identifier of USSD-String

97

TAG R4.21 Cross Reference Listing for MAP-Protocol 00-01-06 07:32:40 PAGE

DEFINED in MAP-SS-DataTypes : 219

USSD-String.....type reference OCTET STRING
 DEFINED in MAP-SS-DataTypes : 227
 USED in MAP-SS-DataTypes : 23 212 219
 USED in MAP-LCS-DataTypes : 45 128

uui.....identifier of [1] UUI
 DEFINED in MAP-CH-DataTypes : 237

UUI.....type reference OCTET STRING
 DEFINED in MAP-CH-DataTypes : 245
 USED in MAP-CH-DataTypes : 237

uuIndicator.....identifier of [0] UUIndicator
 DEFINED in MAP-CH-DataTypes : 236

UUIndicator.....type reference OCTET STRING
 DEFINED in MAP-CH-DataTypes : 242
 USED in MAP-CH-DataTypes : 236

uus1.....value reference SS-Code, '10000001'B
 DEFINED in MAP-SS-Code : 108

uus2.....value reference SS-Code, '10000010'B
 DEFINED in MAP-SS-Code : 110

uus3.....value reference SS-Code, '10000011'B
 DEFINED in MAP-SS-Code : 112

uusCFInteraction.....identifier of [2] NULL
 DEFINED in MAP-CH-DataTypes : 238

uu-Data.....identifier of [10] UU-Data
 DEFINED in MAP-CH-DataTypes : 230

UU-Data.....type reference SEQUENCE
 DEFINED in MAP-CH-DataTypes : 235
 USED in MAP-CH-DataTypes : 230

valueAddedServices.....identifier of Named Number, 1
 DEFINED in MAP-LCS-DataTypes : 109

VBSDataList.....type reference SEQUENCE OF
 DEFINED in MAP-MS-DataTypes : 1301
 USED in MAP-MS-DataTypes : 587

vbsGroupIndication.....identifier of [7] NULL
 DEFINED in MAP-MS-DataTypes : 898

vbsSubscriptionData.....identifier of [11] VBSDataList
 DEFINED in MAP-MS-DataTypes : 587

verticalCoordinateRequest.....identifier of [1] NULL
 DEFINED in MAP-LCS-DataTypes : 139

vertical-accuracy.....identifier of [2] Vertical-Accuracy
 DEFINED in MAP-LCS-DataTypes : 140

Vertical-Accuracy.....type reference OCTET STRING
 DEFINED in MAP-LCS-DataTypes : 148
 USED in MAP-LCS-DataTypes : 140

VGCSDataList.....type reference SEQUENCE OF
 DEFINED in MAP-MS-DataTypes : 1304
 USED in MAP-MS-DataTypes : 588

vgcsGroupIndication.....identifier of [8] NULL
 DEFINED in MAP-MS-DataTypes : 899

vgcsSubscriptionData.....identifier of [12] VGCSDataList
 DEFINED in MAP-MS-DataTypes : 588

vlr.....identifier of Named Number, 2
 DEFINED in MAP-CommonDataTypes : 307

vlrCamelSubscriptionInfo.....identifier of [13] VlrCamelSubscriptionInfo
 DEFINED in MAP-MS-DataTypes : 589

VlrCamelSubscriptionInfo.....type reference SEQUENCE

DEFINED in MAP-MS-DataTypes : 934
USED in MAP-MS-DataTypes : 589

vlr-Capability.....identifier of [6] VLR-Capability
 DEFINED in MAP-MS-DataTypes : 192

VLR-Capability.....type reference SEQUENCE
 DEFINED in MAP-MS-DataTypes : 194
 USED in MAP-MS-DataTypes : 192 1291

vlr-Capability.....identifier of [6] VLR-Capability
 DEFINED in MAP-MS-DataTypes : 1291

vlr-Number.....identifier of ISDN-AddressString
 DEFINED in MAP-MS-DataTypes : 188

vlr-Number.....identifier of [0] ISDN-AddressString
 DEFINED in MAP-MS-DataTypes : 243

vlr-number.....identifier of [1] ISDN-AddressString
 DEFINED in MAP-MS-DataTypes : 1355

vmsc.....identifier of Named Number, 5
 DEFINED in MAP-CommonDataTypes : 310

vmsc-Address.....identifier of [2] ISDN-AddressString
 DEFINED in MAP-CH-DataTypes : 155

voiceBroadcastCall.....value reference TeleserviceCode, '10010010'B
 DEFINED in MAP-TS-Code : 70

VoiceBroadcastData.....type reference SEQUENCE
 DEFINED in MAP-MS-DataTypes : 1316
 USED in MAP-MS-DataTypes : 1302

voiceGroupCall.....value reference TeleserviceCode, '10010001'B
 DEFINED in MAP-TS-Code : 69

VoiceGroupCallData.....type reference SEQUENCE
 DEFINED in MAP-MS-DataTypes : 1311
 USED in MAP-MS-DataTypes : 1305

vplmnAddressAllowed.....identifier of [19] NULL
 DEFINED in MAP-MS-DataTypes : 449

vt-BCSM-CAMEL-TDP-CriteriaList.....identifier of [5] T-BCSM-CAMEL-TDP-CriteriaList
 DEFINED in MAP-MS-DataTypes : 1483

vt-CSI.....identifier of [7] T-CSI
 DEFINED in MAP-MS-DataTypes : 943

vt-CSI.....identifier of Named Number, 2
 DEFINED in MAP-MS-DataTypes : 1447

vt-CSI.....identifier of [4] T-CSI
 DEFINED in MAP-MS-DataTypes : 1482

whiteListed.....identifier of Named Number, 0
 DEFINED in MAP-MS-DataTypes : 389

wrongPasswordAttemptsCounter.....identifier of WrongPasswordAttemptsCounter
 DEFINED in MAP-MS-DataTypes : 1464

WrongPasswordAttemptsCounter.....type reference INTEGER
 DEFINED in MAP-MS-DataTypes : 1469
 USED in MAP-MS-DataTypes : 1464 1585

wrongPasswordAttemptsCounter.....identifier of [3] WrongPasswordAttemptsCounter
 DEFINED in MAP-MS-DataTypes : 1585

xres.....identifier of XRES
 DEFINED in MAP-MS-DataTypes : 290

XRES.....type reference OCTET STRING
 DEFINED in MAP-MS-DataTypes : 302
 USED in MAP-MS-DataTypes : 290

ZoneCode.....type reference OCTET STRING
 DEFINED in MAP-MS-DataTypes : 868
 USED in MAP-MS-DataTypes : 866 897

ZoneCodeList.....type reference SEQUENCE OF

DEFINED in MAP-MS-DataTypes	:	865	
USED in MAP-MS-DataTypes	:	50	586

99

TAG R4.21 Cross Reference Listing for MAP-Protocol

00-01-06 07:32:40 PAGE

zoneCodesConflict.....identifier of Named Number, 2
DEFINED in MAP-MS-DataTypes : 887


```

... ,
solsaSupportIndicator [2] IMPLICIT NULL OPTIONAL,
istSupportIndicator [1] IMPLICIT ENUMERATED {
  basicISTSupported (0 ),
  istCommandSupported (1 ),
  ... } OPTIONAL,
superChargerSupportedInServingNetworkEntity [3] CHOICE {
  sendSubscriberData [0] IMPLICIT NULL,
  subscriberDataStored [1] IMPLICIT OCTET STRING ( SIZE (1..6 ) ) } OPTIONAL}
OPTIONAL}
RESULT
  updateLocationRes SEQUENCE {
    hlr-Number OCTET STRING ( SIZE (1..20 ) ) ( SIZE (1..9 ) ),
    extensionContainer SEQUENCE {
      privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE (1..10 ) ) OF
        SEQUENCE {
          extId MAP-EXTENSION .&extensionId ( {
            '... } ) ,
          extType MAP-EXTENSION .&ExtensionType ( {
            '... } { @extId } ) OPTIONAL} OPTIONAL,
          pcs-Extensions [1] IMPLICIT SEQUENCE {
            ... } OPTIONAL,
            ... } OPTIONAL,
            ... }
    ... }
  ERRORS {
    -- systemFailure -- localValue : 34,
    -- dataMissing -- localValue : 35,
    -- unexpectedDataValue -- localValue : 36,
    -- unknownSubscriber -- localValue : 1,
    -- roamingNotAllowed -- localValue : 8}
  ::= localValue : 2

cancelLocation OPERATION
  ARGUMENT
    cancelLocationArg [3] IMPLICIT SEQUENCE {
      identity CHOICE {
        imsi OCTET STRING ( SIZE (3..8 ) ),
        imsi-WithLMSI SEQUENCE {
          imsi OCTET STRING ( SIZE (3..8 ) ),
          lmsi OCTET STRING ( SIZE (4 ) ),
          ... }},
      cancellationType ENUMERATED {
        updateProcedure (0 ),
        subscriptionWithdraw (1 ),
        ... } OPTIONAL,
      extensionContainer SEQUENCE {
        privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE (1..10 ) ) OF
          SEQUENCE {
            extId MAP-EXTENSION .&extensionId ( {
              '... } ) ,
            extType MAP-EXTENSION .&ExtensionType ( {
              '... } { @extId } ) OPTIONAL} OPTIONAL,
            pcs-Extensions [1] IMPLICIT SEQUENCE {
              ... } OPTIONAL,
              ... } OPTIONAL,
              ... }
    ... }
  RESULT
    cancelLocationRes SEQUENCE {
      extensionContainer SEQUENCE {
        privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE (1..10 ) ) OF
          SEQUENCE {
            extId MAP-EXTENSION .&extensionId ( {
              '... } ) ,
            extType MAP-EXTENSION .&ExtensionType ( {
              '... } { @extId } ) OPTIONAL} OPTIONAL,
            pcs-Extensions [1] IMPLICIT SEQUENCE {
              ... } OPTIONAL,
              ... } OPTIONAL,
              ... }
    ... }
  ERRORS {
    -- dataMissing -- localValue : 35,
    -- unexpectedDataValue -- localValue : 36}
  ::= localValue : 3

purgeMS OPERATION
  ARGUMENT
    purgeMS-Arg [3] IMPLICIT SEQUENCE {

```

```

imsi                OCTET STRING ( SIZE ( 3..8 ) ),
vlr-Number          [0] IMPLICIT OCTET STRING ( SIZE ( 1..20 ) ) ( SIZE ( 1..9 ) ) OPTIONAL,
sgsn-Number         [1] IMPLICIT OCTET STRING ( SIZE ( 1..20 ) ) ( SIZE ( 1..9 ) ) OPTIONAL,
extensionContainer  SEQUENCE {
  privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE ( 1..10 ) ) OF
    SEQUENCE {
      extId          MAP-EXTENSION .&extensionId ( {
        '...' ) ,
      extType       MAP-EXTENSION .&ExtensionType ( {
        '...' { @extId } ) OPTIONAL} OPTIONAL,
      pcs-Extensions [1] IMPLICIT SEQUENCE {
        ... } OPTIONAL,
      ... } OPTIONAL,
      ... }
RESULT
  purgeMS-Res SEQUENCE {
    freezeTMSI      [0] IMPLICIT NULL OPTIONAL,
    freezeP-TMSI    [1] IMPLICIT NULL OPTIONAL,
    extensionContainer SEQUENCE {
      privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE ( 1..10 ) ) OF
        SEQUENCE {
          extId          MAP-EXTENSION .&extensionId ( {
            '...' ) ,
          extType       MAP-EXTENSION .&ExtensionType ( {
            '...' { @extId } ) OPTIONAL} OPTIONAL,
          pcs-Extensions [1] IMPLICIT SEQUENCE {
            ... } OPTIONAL,
          ... } OPTIONAL,
          ... }
ERRORS {
  -- dataMissing -- localValue : 35,
  -- unexpectedDataValue -- localValue : 36,
  -- unknownSubscriber -- localValue : 1}
 ::= localValue : 67

sendIdentification OPERATION
ARGUMENT
  sendIdentificationArg SEQUENCE {
    tmsi                OCTET STRING ( SIZE ( 1..4 ) ),
    numberOfRequestedVectors INTEGER ( 1..5 ),
    segmentationProhibited NULL OPTIONAL,
    extensionContainer  SEQUENCE {
      privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE ( 1..10 ) ) OF
        SEQUENCE {
          extId          MAP-EXTENSION .&extensionId ( {
            '...' ) ,
          extType       MAP-EXTENSION .&ExtensionType ( {
            '...' { @extId } ) OPTIONAL} OPTIONAL,
          pcs-Extensions [1] IMPLICIT SEQUENCE {
            ... } OPTIONAL,
            ... } OPTIONAL,
            ... }
RESULT
  sendIdentificationRes [3] IMPLICIT SEQUENCE {
    imsi                OCTET STRING ( SIZE ( 3..8 ) ) OPTIONAL,
    authenticationSetList CHOICE {
      tripletList       [0] IMPLICIT SEQUENCE ( SIZE ( 1..5 ) ) OF
        SEQUENCE {
          rand          OCTET STRING ( SIZE ( 16 ) ),
          sres          OCTET STRING ( SIZE ( 4 ) ),
          kc            OCTET STRING ( SIZE ( 8 ) ),
          ... },
      quintupletList    [1] IMPLICIT SEQUENCE ( SIZE ( 1..5 ) ) OF
        SEQUENCE {
          rand          OCTET STRING ( SIZE ( 16 ) ),
          xres          OCTET STRING ( SIZE ( 4..16 ) ),
          ck            OCTET STRING ( SIZE ( 16 ) ),
          ik            OCTET STRING ( SIZE ( 16 ) ),
          autn          OCTET STRING ( SIZE ( 14..18 ) ),
          ... }} OPTIONAL,
    extensionContainer  [2] IMPLICIT SEQUENCE {
      privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE ( 1..10 ) ) OF
        SEQUENCE {
          extId          MAP-EXTENSION .&extensionId ( {
            '...' ) ,
          extType       MAP-EXTENSION .&ExtensionType ( {

```

```

        ... } { @extId } ) OPTIONAL} OPTIONAL,
    pcs-Extensions [1] IMPLICIT SEQUENCE {
        ... } OPTIONAL,
        ... } OPTIONAL,
    ... }
ERRORS {
    -- dataMissing -- localValue : 35,
    -- unidentifiedSubscriber -- localValue : 5}
 ::= localValue : 55

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 ) ),
            extensionContainer SEQUENCE {
                privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE ( 1..10 ) ) OF
                SEQUENCE {
                    extId MAP-EXTENSION .&extensionId ( {
                        ... } ),
                    extType MAP-EXTENSION .&ExtensionType ( {
                        ... } { @extId } ) OPTIONAL} OPTIONAL,
                    pcs-Extensions [1] IMPLICIT SEQUENCE {
                        ... } OPTIONAL,
                        ... } OPTIONAL,
                        ... } OPTIONAL,
                    ... } OPTIONAL,
                ... }
RESULT
    prepareHO-Res SEQUENCE {
        handoverNumber OCTET STRING ( SIZE ( 1..20 ) ) ( 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 ) ),
            extensionContainer SEQUENCE {
                privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE ( 1..10 ) ) OF
                SEQUENCE {
                    extId MAP-EXTENSION .&extensionId ( {
                        ... } ),
                    extType MAP-EXTENSION .&ExtensionType ( {
                        ... } { @extId } ) OPTIONAL} OPTIONAL,
                    pcs-Extensions [1] IMPLICIT SEQUENCE {
                        ... } OPTIONAL,
                        ... } OPTIONAL,
                        ... } OPTIONAL,
                    ... } 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 ) ),
        extensionContainer SEQUENCE {
            privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE ( 1..10 ) ) OF
            SEQUENCE {
                extId MAP-EXTENSION .&extensionId ( {
                    ... } ),

```

```

        extType      MAP-EXTENSION .&ExtensionType ( {
            '...'} { @extId } ) OPTIONAL} OPTIONAL,
        pcs-Extensions [1] IMPLICIT SEQUENCE {
            ... } OPTIONAL,
            ... } OPTIONAL,
        ... }
 ::= 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) ),
        extensionContainer SEQUENCE {
            privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE (1..10) ) OF
                SEQUENCE {
                    extId      MAP-EXTENSION .&extensionId ( {
                        '...'} ) ,
                    extType      MAP-EXTENSION .&ExtensionType ( {
                        '...'} { @extId } ) OPTIONAL} OPTIONAL,
                    pcs-Extensions [1] IMPLICIT SEQUENCE {
                        ... } OPTIONAL,
                        ... } OPTIONAL,
                    ... }
                }
        }
 ::= 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) ),
        extensionContainer SEQUENCE {
            privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE (1..10) ) OF
                SEQUENCE {
                    extId      MAP-EXTENSION .&extensionId ( {
                        '...'} ) ,
                    extType      MAP-EXTENSION .&ExtensionType ( {
                        '...'} { @extId } ) OPTIONAL} OPTIONAL,
                    pcs-Extensions [1] IMPLICIT SEQUENCE {
                        ... } OPTIONAL,
                        ... } OPTIONAL,
                    ... }
                }
        }
 ::= localValue : 34

prepareSubsequentHandover OPERATION
ARGUMENT
    prepareSubsequentHO-Arg SEQUENCE {
        targetCellId      OCTET STRING ( SIZE (5..7) ),
        targetMSC-Number  OCTET STRING ( SIZE (1..20) ) ( 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) ),
            extensionContainer SEQUENCE {
                privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE (1..10) ) OF
                    SEQUENCE {
                        extId      MAP-EXTENSION .&extensionId ( {
                            '...'} ) ,
                        extType      MAP-EXTENSION .&ExtensionType ( {
                            '...'} { @extId } ) OPTIONAL} OPTIONAL,
                            pcs-Extensions [1] IMPLICIT SEQUENCE {
                                ... } OPTIONAL,
                                ... } OPTIONAL,
                                ... }
                    },
                ... }
            }
        }
    }

```

```

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 ) ),
    extensionContainer SEQUENCE {
      privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE ( 1..10 ) ) OF
        SEQUENCE {
          extId      MAP-EXTENSION .&extensionId ( {
            '...'} ) ,
          extType    MAP-EXTENSION .&ExtensionType ( {
            '...'} { @extId } ) OPTIONAL} OPTIONAL,
      pcs-Extensions [1] IMPLICIT SEQUENCE {
        ... } OPTIONAL,
        ... } OPTIONAL,
        ... }
  }
  ERRORS {
    -- unexpectedDataValue -- localValue : 36,
    -- dataMissing -- localValue : 35,
    -- unknownMSC -- localValue : 3,
    -- subsequentHandoverFailure -- localValue : 26}
  ::= localValue : 69

```

sendAuthenticationInfo OPERATION

```

ARGUMENT
  sendAuthenticationInfoArg SEQUENCE {
    imsi            [0] IMPLICIT OCTET STRING ( SIZE ( 3..8 ) ),
    numberOfRequestedVectors INTEGER ( 1..5 ),
    segmentationProhibited NULL OPTIONAL,
    immediateResponsePreferred [1] IMPLICIT NULL OPTIONAL,
    re-synchronisationInfo SEQUENCE {
      rand          OCTET STRING ( SIZE ( 16 ) ),
      rand-ms       OCTET STRING ( SIZE ( 16 ) ),
      auts          OCTET STRING ( SIZE ( 12..16 ) ),
      ... } OPTIONAL,
    extensionContainer [2] IMPLICIT SEQUENCE {
      privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE ( 1..10 ) ) OF
        SEQUENCE {
          extId      MAP-EXTENSION .&extensionId ( {
            '...'} ) ,
          extType    MAP-EXTENSION .&ExtensionType ( {
            '...'} { @extId } ) OPTIONAL} OPTIONAL,
      pcs-Extensions [1] IMPLICIT SEQUENCE {
        ... } OPTIONAL,
        ... } OPTIONAL,
        ... }
  }

```

```

RESULT
  sendAuthenticationInfoRes [3] IMPLICIT SEQUENCE {
    authenticationSetList CHOICE {
      tripletList [0] IMPLICIT SEQUENCE ( SIZE ( 1..5 ) ) OF
        SEQUENCE {
          rand      OCTET STRING ( SIZE ( 16 ) ),
          sres      OCTET STRING ( SIZE ( 4 ) ),
          kc        OCTET STRING ( SIZE ( 8 ) ),
          ... },
      quintupletList [1] IMPLICIT SEQUENCE ( SIZE ( 1..5 ) ) OF
        SEQUENCE {
          rand      OCTET STRING ( SIZE ( 16 ) ),
          xres      OCTET STRING ( SIZE ( 4..16 ) ),
          ck        OCTET STRING ( SIZE ( 16 ) ),
          ik        OCTET STRING ( SIZE ( 16 ) ),
          autn      OCTET STRING ( SIZE ( 14..18 ) ),
          ... }} OPTIONAL,
    extensionContainer SEQUENCE {
      privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE ( 1..10 ) ) OF
        SEQUENCE {
          extId      MAP-EXTENSION .&extensionId ( {
            '...'} ) ,
          extType    MAP-EXTENSION .&ExtensionType ( {
            '...'} { @extId } ) OPTIONAL} OPTIONAL,
      pcs-Extensions [1] IMPLICIT SEQUENCE {
        ... } OPTIONAL,
        ... } OPTIONAL,
        ... }
  }

```

```

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,
    -- unknownEquipment -- localValue : 7}
 ::= localValue : 43

insertSubscriberData OPERATION
ARGUMENT
  insertSubscriberDataArg SEQUENCE {
    imsi [0] IMPLICIT OCTET STRING ( SIZE ( 3..8 ) )
OPTIONAL,
    msisdn [1] IMPLICIT OCTET STRING ( SIZE ( 1..20 ) ) (
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..5 ) ) OPTIONAL,
    teleserviceList [6] IMPLICIT SEQUENCE ( SIZE ( 1..20 ) ) OF
      OCTET STRING ( SIZE ( 1..5 ) ) OPTIONAL,
    provisionedSS [7] IMPLICIT SEQUENCE ( SIZE ( 1..30 ) ) OF
      CHOICE {
        forwardingInfo [0] IMPLICIT SEQUENCE {
          ss-Code OCTET STRING ( SIZE ( 1 ) ),
          forwardingFeatureList SEQUENCE ( SIZE ( 1..32 ) ) OF
            SEQUENCE {
              basicService CHOICE {
                ext-BearerService [2] IMPLICIT OCTET STRING ( SIZE ( 1..5 ) ),
                ext-Teleservice [3] IMPLICIT OCTET STRING ( SIZE ( 1..5 ) )}
OPTIONAL,
              ss-Status [4] IMPLICIT OCTET STRING ( SIZE ( 1..5 ) ),
              forwardedToNumber [5] IMPLICIT OCTET STRING ( SIZE ( 1..20 ) ) ( SIZE
( 1..9 ) ) OPTIONAL,
              forwardedToSubaddress [8] IMPLICIT OCTET STRING ( SIZE ( 1..21 ) )
OPTIONAL,
              forwardingOptions [6] IMPLICIT OCTET STRING ( SIZE ( 1..5 ) ) OPTIONAL,
              noReplyConditionTime [7] IMPLICIT INTEGER ( 1..100 ) OPTIONAL,
              extensionContainer [9] IMPLICIT SEQUENCE {
                privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE ( 1..10 ) ) OF
                  SEQUENCE {
                    extId MAP-EXTENSION .&extensionId ( {
                      '...' ) ,
                    extType MAP-EXTENSION .&ExtensionType ( {
                      '...' { @extId } ) OPTIONAL} OPTIONAL,
                pcs-Extensions [1] IMPLICIT SEQUENCE {
                  ... } OPTIONAL,
                  ... } OPTIONAL,
                  ... },
                extensionContainer [0] IMPLICIT SEQUENCE {
                  privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE ( 1..10 ) ) OF
                    SEQUENCE {
                      extId MAP-EXTENSION .&extensionId ( {
                        '...' ) ,
                      extType MAP-EXTENSION .&ExtensionType ( {
                        '...' { @extId } ) OPTIONAL} OPTIONAL,
                  pcs-Extensions [1] IMPLICIT SEQUENCE {
                    ... } OPTIONAL,
                    ... } OPTIONAL,
                    ... },
                callBarringInfo [1] IMPLICIT SEQUENCE {
                  ss-Code OCTET STRING ( SIZE ( 1 ) ),
                  callBarringFeatureList SEQUENCE ( SIZE ( 1..32 ) ) OF

```

```

SEQUENCE {
  basicService          CHOICE {
    ext-BearerService   [2] IMPLICIT OCTET STRING ( SIZE (1..5) ),
    ext-Teleservice     [3] IMPLICIT OCTET STRING ( SIZE (1..5) )}
OPTIONAL,
  ss-Status             [4] IMPLICIT OCTET STRING ( SIZE (1..5) ),
  extensionContainer   SEQUENCE {
    privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE (1..10) ) OF
      SEQUENCE {
        extId           MAP-EXTENSION .&extensionId ( {
          '...' ) ,
        extType        MAP-EXTENSION .&ExtensionType ( {
          '...' { @extId } ) OPTIONAL} OPTIONAL,
        pcs-Extensions [1] IMPLICIT SEQUENCE {
          ... } OPTIONAL,
          ... } OPTIONAL,
          ... },
    extensionContainer SEQUENCE {
      privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE (1..10) ) OF
        SEQUENCE {
          extId           MAP-EXTENSION .&extensionId ( {
            '...' ) ,
          extType        MAP-EXTENSION .&ExtensionType ( {
            '...' { @extId } ) OPTIONAL} OPTIONAL,
          pcs-Extensions [1] IMPLICIT SEQUENCE {
            ... } OPTIONAL,
            ... } OPTIONAL,
            ... },
      cug-Info           [2] IMPLICIT SEQUENCE {
        cug-SubscriptionList SEQUENCE ( SIZE (0..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..32) ) OF
              CHOICE {
                ext-BearerService [2] IMPLICIT OCTET STRING ( SIZE (1..5) ),
                ext-Teleservice [3] IMPLICIT OCTET STRING ( SIZE (1..5) )}
OPTIONAL,
            extensionContainer [0] IMPLICIT SEQUENCE {
              privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE (1..10) ) OF
                SEQUENCE {
                  extId           MAP-EXTENSION .&extensionId ( {
                    '...' ) ,
                  extType        MAP-EXTENSION .&ExtensionType ( {
                    '...' { @extId } ) OPTIONAL} OPTIONAL,
                  pcs-Extensions [1] IMPLICIT SEQUENCE {
                    ... } OPTIONAL,
                    ... } OPTIONAL,
                    ... },
              cug-FeatureList SEQUENCE ( SIZE (1..32) ) OF
                SEQUENCE {
                  basicService          CHOICE {
                    ext-BearerService [2] IMPLICIT OCTET STRING ( SIZE (1..5) ),
                    ext-Teleservice [3] IMPLICIT OCTET STRING ( SIZE (1..5) )}
OPTIONAL,
                  preferentialCUG-Indicator INTEGER ( 0..32767 ) OPTIONAL,
                  interCUG-Restrictions OCTET STRING ( SIZE (1) ),
                  extensionContainer SEQUENCE {
                    privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE (1..10) ) OF
                      SEQUENCE {
                        extId           MAP-EXTENSION .&extensionId ( {
                          '...' ) ,
                        extType        MAP-EXTENSION .&ExtensionType ( {
                          '...' { @extId } ) OPTIONAL} OPTIONAL,
                        pcs-Extensions [1] IMPLICIT SEQUENCE {
                          ... } OPTIONAL,
                          ... } OPTIONAL,
                          ... },
                    extensionContainer [0] IMPLICIT SEQUENCE {
                      privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE (1..10) ) OF
                        SEQUENCE {

```

```

        extId      MAP-EXTENSION .&extensionId ( {
            ... } ) ,
        extType    MAP-EXTENSION .&ExtensionType ( {
            ... } { @extId } ) OPTIONAL} OPTIONAL,
        pcs-Extensions [1] IMPLICIT SEQUENCE {
            ... } OPTIONAL,
        ... } OPTIONAL,
        ... } ,
    ss-Data [3] IMPLICIT SEQUENCE {
        ss-Code      OCTET STRING ( SIZE ( 1 ) ) ,
        ss-Status    [4] IMPLICIT OCTET STRING ( SIZE ( 1..5 ) ) ,
        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..32 ) ) OF
            CHOICE {
                ext-BearerService [2] IMPLICIT OCTET STRING ( SIZE ( 1..5 ) ) ,
                ext-Teleservice [3] IMPLICIT OCTET STRING ( SIZE ( 1..5 ) ) } OPTIONAL,
        extensionContainer [5] IMPLICIT SEQUENCE {
            privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE ( 1..10 ) ) OF
                SEQUENCE {
                    extId      MAP-EXTENSION .&extensionId ( {
                        ... } ) ,
                    extType    MAP-EXTENSION .&ExtensionType ( {
                        ... } { @extId } ) OPTIONAL} OPTIONAL,
                pcs-Extensions [1] IMPLICIT SEQUENCE {
                    ... } OPTIONAL,
                ... } OPTIONAL,
                ... } ,
        emlpp-Info [4] IMPLICIT SEQUENCE {
            maximumTitledPriority INTEGER ( 0..15 ) ,
            defaultPriority      INTEGER ( 0..15 ) ,
            extensionContainer SEQUENCE {
                privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE ( 1..10 ) ) OF
                    SEQUENCE {
                        extId      MAP-EXTENSION .&extensionId ( {
                            ... } ) ,
                        extType    MAP-EXTENSION .&ExtensionType ( {
                            ... } { @extId } ) OPTIONAL} OPTIONAL,
                    pcs-Extensions [1] IMPLICIT SEQUENCE {
                        ... } OPTIONAL,
                    ... } OPTIONAL,
                    ... } } OPTIONAL,
    odb-Data [8] IMPLICIT SEQUENCE {
        odb-GeneralData BIT STRING {
            allog-CallsBarred ( 0 ) ,
            internationalOGCallsBarred ( 1 ) ,
            internationalOGCallsNotToHPLMN-CountryBarred ( 2 ) ,
            interzonalOGCallsBarred ( 6 ) ,
            interzonalOGCallsNotToHPLMN-CountryBarred ( 7 ) ,
            interzonalOGCallsAndInternationalOGCallsNotToHPLMN-CountryBarred ( 8 ) ,
            premiumRateInformationOGCallsBarred ( 3 ) ,
            premiumRateEntertainmentOGCallsBarred ( 4 ) ,
            ss-AccessBarred ( 5 ) ,
            alleCT-Barred ( 9 ) ,
            chargeableECT-Barred ( 10 ) ,
            internationalECT-Barred ( 11 ) ,
            interzonalECT-Barred ( 12 ) ,
            doublyChargeableECT-Barred ( 13 ) ,
            multipleECT-Barred ( 14 ) } ( SIZE ( 15..32 ) ) ,
        odb-HPLMN-Data BIT STRING {
            plmn-SpecificBarringType1 ( 0 ) ,
            plmn-SpecificBarringType2 ( 1 ) ,
            plmn-SpecificBarringType3 ( 2 ) ,
            plmn-SpecificBarringType4 ( 3 ) } ( SIZE ( 4..32 ) ) OPTIONAL,
        extensionContainer SEQUENCE {
            privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE ( 1..10 ) ) OF
                SEQUENCE {
                    extId      MAP-EXTENSION .&extensionId ( {
                        ... } ) ,
                    extType    MAP-EXTENSION .&ExtensionType ( {

```



```

        ... } OPTIONAL,
        pcs-Extensions [1] IMPLICIT SEQUENCE {
        ... } OPTIONAL,
        ... } OPTIONAL,
        ... } OPTIONAL,
        roamingRestrictionDueToUnsupportedFeature [9] IMPLICIT NULL OPTIONAL,
        regionalSubscriptionData [10] IMPLICIT SEQUENCE ( SIZE (1..10 ) ) OF
        OCTET STRING ( SIZE (2 ) ) OPTIONAL,
        vbsSubscriptionData [11] IMPLICIT SEQUENCE ( SIZE (1..50 ) ) OF
        SEQUENCE {
        groupId OCTET STRING ( SIZE (3 ) ),
        broadcastInitEntitlement NULL OPTIONAL,
        extensionContainer SEQUENCE {
        privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE (1..10 ) ) OF
        SEQUENCE {
        extId MAP-EXTENSION .&extensionId ( {
        '... } ) ,
        extType MAP-EXTENSION .&ExtensionType ( {
        '... } { @extId } ) OPTIONAL} OPTIONAL,
        pcs-Extensions [1] IMPLICIT SEQUENCE {
        ... } OPTIONAL,
        ... } OPTIONAL,
        ... } OPTIONAL,
        ... } OPTIONAL,
        vgcSubscriptionData [12] IMPLICIT SEQUENCE ( SIZE (1..50 ) ) OF
        SEQUENCE {
        groupId OCTET STRING ( SIZE (3 ) ),
        extensionContainer SEQUENCE {
        privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE (1..10 ) ) OF
        SEQUENCE {
        extId MAP-EXTENSION .&extensionId ( {
        '... } ) ,
        extType MAP-EXTENSION .&ExtensionType ( {
        '... } { @extId } ) OPTIONAL} OPTIONAL,
        pcs-Extensions [1] IMPLICIT SEQUENCE {
        ... } OPTIONAL,
        ... } OPTIONAL,
        ... } OPTIONAL,
        ... } OPTIONAL,
        vlrCamelSubscriptionInfo [13] IMPLICIT SEQUENCE {
        o-CSI [0] IMPLICIT SEQUENCE {
        o-BcsmCamelTDPDataList SEQUENCE ( SIZE (1..10 ) ) OF
        SEQUENCE {
        o-BcsmTriggerDetectionPoint ENUMERATED {
        collectedInfo (2 ) ,
        ... ,
        routeSelectFailure (4 )},
        serviceKey INTEGER ( 0..2147483647 ) ,
        gsmSCF-Address [0] IMPLICIT OCTET STRING ( SIZE (1..20 ) ) (
        SIZE (1..9 ) ) ,
        defaultCallHandling [1] IMPLICIT ENUMERATED {
        continueCall (0 ) ,
        releaseCall (1 ) ,
        ... },
        extensionContainer [2] IMPLICIT SEQUENCE {
        privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE (1..10 ) ) OF
        SEQUENCE {
        extId MAP-EXTENSION .&extensionId ( {
        '... } ) ,
        extType MAP-EXTENSION .&ExtensionType ( {
        '... } { @extId } ) OPTIONAL} OPTIONAL,
        pcs-Extensions [1] IMPLICIT SEQUENCE {
        ... } OPTIONAL,
        ... } OPTIONAL,
        ... },
        extensionContainer SEQUENCE {
        privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE (1..10 ) ) OF
        SEQUENCE {
        extId MAP-EXTENSION .&extensionId ( {
        '... } ) ,
        extType MAP-EXTENSION .&ExtensionType ( {
        '... } { @extId } ) OPTIONAL} OPTIONAL,
        pcs-Extensions [1] IMPLICIT SEQUENCE {
        ... } OPTIONAL,
        ... } OPTIONAL,
        ... ,

```

```

camelCapabilityHandling [0] IMPLICIT INTEGER ( 1..16 ) OPTIONAL,
notificationToCSE      [1] IMPLICIT NULL OPTIONAL,
csiActive               [2] IMPLICIT NULL OPTIONAL} OPTIONAL,
extensionContainer      [1] IMPLICIT SEQUENCE {
privateExtensionList   [0] IMPLICIT SEQUENCE ( SIZE (1..10 ) ) OF
SEQUENCE {
    extId      MAP-EXTENSION .&extensionId ( {
        '...' ) ,
    extType    MAP-EXTENSION .&ExtensionType ( {
        '...' { @extId } ) OPTIONAL} OPTIONAL,
pcs-Extensions [1] IMPLICIT SEQUENCE {
    ... } OPTIONAL,
... } OPTIONAL,
... ,
ss-CSI [2] IMPLICIT SEQUENCE {
ss-CamelData SEQUENCE {
    ss-EventList SEQUENCE ( SIZE (1..10 ) ) OF
    OCTET STRING ( SIZE (1 ) ),
    gsmSCF-Address OCTET STRING ( SIZE (1..20 ) ) ( SIZE (1..9 ) ),
    extensionContainer [0] IMPLICIT SEQUENCE {
        privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE (1..10 ) ) OF
        SEQUENCE {
            extId      MAP-EXTENSION .&extensionId ( {
                '...' ) ,
            extType    MAP-EXTENSION .&ExtensionType ( {
                '...' { @extId } ) OPTIONAL} OPTIONAL,
        pcs-Extensions [1] IMPLICIT SEQUENCE {
            ... } OPTIONAL,
        ... } OPTIONAL,
        ... ,
        notificationToCSE [1] IMPLICIT NULL OPTIONAL,
        csiActive         [2] IMPLICIT NULL OPTIONAL},
    extensionContainer SEQUENCE {
        privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE (1..10 ) ) OF
        SEQUENCE {
            extId      MAP-EXTENSION .&extensionId ( {
                '...' ) ,
            extType    MAP-EXTENSION .&ExtensionType ( {
                '...' { @extId } ) OPTIONAL} OPTIONAL,
        pcs-Extensions [1] IMPLICIT SEQUENCE {
            ... } OPTIONAL,
        ... } OPTIONAL,
        ... } OPTIONAL,
o-BcsmCamelTDP-CriteriaList [4] IMPLICIT SEQUENCE ( SIZE (1..10 ) ) OF
SEQUENCE {
    o-BcsmTriggerDetectionPoint ENUMERATED {
        collectedInfo (2 ) ,
        ... ,
        routeSelectFailure (4 ) } ,
    destinationNumberCriteria [0] IMPLICIT SEQUENCE {
        matchType [0] IMPLICIT ENUMERATED {
            inhibiting (0 ) ,
            enabling (1 ) } ,
        destinationNumberList [1] IMPLICIT SEQUENCE ( SIZE (1..10 ) ) OF
        OCTET STRING ( SIZE (1..20 ) ) ( SIZE (1..9 ) ) OPTIONAL,
        destinationNumberLengthList [2] IMPLICIT SEQUENCE ( SIZE (1..3 ) ) OF
        INTEGER ( 1..15 ) OPTIONAL,
        ... } OPTIONAL,
    basicServiceCriteria [1] IMPLICIT SEQUENCE ( SIZE (1..5 ) ) OF
    CHOICE {
        ext-BearerService [2] IMPLICIT OCTET STRING ( SIZE (1..5 ) ) ,
        ext-Teleservice [3] IMPLICIT OCTET STRING ( SIZE (1..5 ) ) } OPTIONAL,
    callTypeCriteria [2] IMPLICIT ENUMERATED {
        forwarded (0 ) ,
        notForwarded (1 ) } OPTIONAL,
    ... ,
    o-CauseValueCriteria [3] IMPLICIT SEQUENCE ( SIZE (1..5 ) ) OF
    OCTET STRING ( SIZE (1 ) ) OPTIONAL,
    extensionContainer [4] IMPLICIT SEQUENCE {
        privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE (1..10 ) ) OF
        SEQUENCE {
            extId      MAP-EXTENSION .&extensionId ( {
                '...' ) ,
            extType    MAP-EXTENSION .&ExtensionType ( {
                '...' { @extId } ) OPTIONAL} OPTIONAL,

```

```

        pcs-Extensions          [1] IMPLICIT SEQUENCE {
            ... } OPTIONAL,
        ... } OPTIONAL} OPTIONAL,
tif-CSI                        [3] IMPLICIT NULL OPTIONAL,
m-CSI                          [5] IMPLICIT SEQUENCE {
    mobilityTriggers           SEQUENCE ( SIZE (1..10 ) ) OF
        OCTET STRING ( SIZE (1 ) ),
    serviceKey                 INTEGER ( 0..2147483647 ),
    gsmSCF-Address             [0] IMPLICIT OCTET STRING ( SIZE (1..20 ) ) ( SIZE (1..9 ) ),
    extensionContainer         [1] IMPLICIT SEQUENCE {
        privateExtensionList   [0] IMPLICIT SEQUENCE ( SIZE (1..10 ) ) OF
            SEQUENCE {
                extId           MAP-EXTENSION .&extensionId ( {
                    '...' ) ,
                extType        MAP-EXTENSION .&ExtensionType ( {
                    '...' { @extId } ) OPTIONAL} OPTIONAL,
                pcs-Extensions [1] IMPLICIT SEQUENCE {
                    ... } OPTIONAL,
                    ... } OPTIONAL,
                notificationToCSE [2] IMPLICIT NULL OPTIONAL,
                csiActive         [3] IMPLICIT NULL OPTIONAL,
                ... } OPTIONAL,
            sms-CSI              [6] IMPLICIT SEQUENCE {
                sms-CAMEL-TDP-DataList [0] IMPLICIT SEQUENCE ( SIZE (1..10 ) ) OF
                    SEQUENCE {
                        sms-TriggerDetectionPoint [0] IMPLICIT ENUMERATED {
                            sms-CollectedInfo (1 ) ,
                            ... },
                        serviceKey              [1] IMPLICIT INTEGER ( 0..2147483647 ),
                        gsmSCF-Address         [2] IMPLICIT OCTET STRING ( SIZE (1..20 ) ) ( SIZE
(1..9 ) ),
                        defaultsSMS-Handling [3] IMPLICIT ENUMERATED {
                            continueTransaction (0 ) ,
                            releaseTransaction (1 ) ,
                            ... },
                        extensionContainer     [4] IMPLICIT SEQUENCE {
                            privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE (1..10 ) ) OF
                                SEQUENCE {
                                    extId           MAP-EXTENSION .&extensionId ( {
                                        '...' ) ,
                                    extType        MAP-EXTENSION .&ExtensionType ( {
                                        '...' { @extId } ) OPTIONAL} OPTIONAL,
                                    pcs-Extensions [1] IMPLICIT SEQUENCE {
                                        ... } OPTIONAL,
                                        ... } OPTIONAL,
                                        ... },
                                camelCapabilityHandling [1] IMPLICIT INTEGER ( 1..16 ) ,
                                extensionContainer [2] IMPLICIT SEQUENCE {
                                    privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE (1..10 ) ) OF
                                        SEQUENCE {
                                            extId           MAP-EXTENSION .&extensionId ( {
                                                '...' ) ,
                                            extType        MAP-EXTENSION .&ExtensionType ( {
                                                '...' { @extId } ) OPTIONAL} OPTIONAL,
                                            pcs-Extensions [1] IMPLICIT SEQUENCE {
                                                ... } OPTIONAL,
                                                ... } OPTIONAL,
                                                notificationToCSE [3] IMPLICIT NULL OPTIONAL,
                                                csiActive         [4] IMPLICIT NULL OPTIONAL,
                                                ... } OPTIONAL,
                                        vt-CSI              [7] IMPLICIT SEQUENCE {
                                            t-BcsmCamelTDPDataList SEQUENCE ( SIZE (1..10 ) ) OF
                                                SEQUENCE {
                                                    t-BcsmTriggerDetectionPoint ENUMERATED {
                                                        termAttemptAuthorized (12 ) ,
                                                        ... ,
                                                        tBusy (13 ) ,
                                                        tNoAnswer (14 ) },
                                                    serviceKey INTEGER ( 0..2147483647 ) ,
                                                    gsmSCF-Address [0] IMPLICIT OCTET STRING ( SIZE (1..20 ) ) (
SIZE (1..9 ) ),
                                                    defaultCallHandling [1] IMPLICIT ENUMERATED {
                                                        continueCall (0 ) ,
                                                        releaseCall (1 ) ,
                                                        ... },
                                                    extensionContainer [2] IMPLICIT SEQUENCE {
                                                        privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE (1..10 ) ) OF

```

```

SEQUENCE {
    extId      MAP-EXTENSION .&extensionId ( {
        '...' ) ,
    extType    MAP-EXTENSION .&ExtensionType ( {
        '...' { @extId } ) OPTIONAL} OPTIONAL,
    pcs-Extensions [1] IMPLICIT SEQUENCE {
        ... } OPTIONAL,
    ... } OPTIONAL,
    ... },
extensionContainer SEQUENCE {
    privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE (1..10 ) ) OF
    SEQUENCE {
        extId      MAP-EXTENSION .&extensionId ( {
            '...' ) ,
        extType    MAP-EXTENSION .&ExtensionType ( {
            '...' { @extId } ) OPTIONAL} OPTIONAL,
        pcs-Extensions [1] IMPLICIT SEQUENCE {
            ... } OPTIONAL,
        ... } OPTIONAL,
        ... },
    camelCapabilityHandling [0] IMPLICIT INTEGER ( 1..16 ) OPTIONAL,
    notificationToCSE [1] IMPLICIT NULL OPTIONAL,
    csi-Active [2] IMPLICIT NULL OPTIONAL} OPTIONAL,
t-BCSM-CAMEL-TDP-CriteriaList [8] IMPLICIT SEQUENCE ( SIZE (1..10 ) ) OF
SEQUENCE {
    t-BCSM-TriggerDetectionPoint ENUMERATED {
        termAttemptAuthorized (12 ) ,
        ... ,
        tBusy (13 ) ,
        tNoAnswer (14 ) } ,
    basicServiceCriteria [0] IMPLICIT SEQUENCE ( SIZE (1..5 ) ) OF
    CHOICE {
        ext-BearerService [2] IMPLICIT OCTET STRING ( SIZE (1..5 ) ) ,
        ext-Teleservice [3] IMPLICIT OCTET STRING ( SIZE (1..5 ) ) } OPTIONAL,
    t-CauseValueCriteria [1] IMPLICIT SEQUENCE ( SIZE (1..5 ) ) OF
    OCTET STRING ( SIZE (1 ) ) OPTIONAL,
    ... } OPTIONAL,
d-CSI [9] IMPLICIT SEQUENCE {
    dp-AnalysedInfoCriteriaList SEQUENCE ( SIZE (1..10 ) ) OF
    SEQUENCE {
        dialledNumber OCTET STRING ( SIZE (1..20 ) ) ( SIZE (1..9 ) ) ,
        serviceKey INTEGER ( 0..2147483647 ) ,
        gsmSCF-Address OCTET STRING ( SIZE (1..20 ) ) ( SIZE (1..9 ) ) ,
        defaultCallHandling ENUMERATED {
            continueCall (0 ) ,
            releaseCall (1 ) ,
            ... } ,
        extensionContainer SEQUENCE {
            privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE (1..10 ) ) OF
            SEQUENCE {
                extId      MAP-EXTENSION .&extensionId ( {
                    '...' ) ,
                extType    MAP-EXTENSION .&ExtensionType ( {
                    '...' { @extId } ) OPTIONAL} OPTIONAL,
                pcs-Extensions [1] IMPLICIT SEQUENCE {
                    ... } OPTIONAL,
                ... } OPTIONAL,
                ... },
            camelCapabilityHandling INTEGER ( 1..16 ) ,
            extensionContainer SEQUENCE {
                privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE (1..10 ) ) OF
                SEQUENCE {
                    extId      MAP-EXTENSION .&extensionId ( {
                        '...' ) ,
                    extType    MAP-EXTENSION .&ExtensionType ( {
                        '...' { @extId } ) OPTIONAL} OPTIONAL,
                    pcs-Extensions [1] IMPLICIT SEQUENCE {
                        ... } OPTIONAL,
                        ... } OPTIONAL,
                        ... } OPTIONAL} OPTIONAL,
                    extensionContainer [14] IMPLICIT SEQUENCE {
                        privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE (1..10 ) ) OF
                        SEQUENCE {
                            extId      MAP-EXTENSION .&extensionId ( {

```

```

        ...} ) ,
        extType      MAP-EXTENSION .&ExtensionType ( {
            '...'} { @extId } ) OPTIONAL} OPTIONAL,
    pcs-Extensions      [1] IMPLICIT SEQUENCE {
        ... } OPTIONAL,
        ... } OPTIONAL,
    ... ,
    naea-PreferredCI      [15] IMPLICIT SEQUENCE {
    naea-PreferredCIC      [0] IMPLICIT OCTET STRING ( SIZE ( 3 ) ),
    extensionContainer      [1] IMPLICIT SEQUENCE {
        privateExtensionList      [0] IMPLICIT SEQUENCE ( SIZE ( 1..10 ) ) OF
            SEQUENCE {
                extId      MAP-EXTENSION .&extensionId ( {
                    '...'} ) ,
                extType      MAP-EXTENSION .&ExtensionType ( {
                    '...'} { @extId } ) OPTIONAL} OPTIONAL,
            pcs-Extensions      [1] IMPLICIT SEQUENCE {
                ... } OPTIONAL,
                ... } OPTIONAL,
                ... } OPTIONAL,
            ... } OPTIONAL,
    gprsSubscriptionData      [16] IMPLICIT SEQUENCE {
    completedDataListIncluded      NULL OPTIONAL,
    gprsDataList      [1] IMPLICIT SEQUENCE ( SIZE ( 1..50 ) ) OF
        SEQUENCE {
            pdp-ContextId      INTEGER ( 1..50 ),
            pdp-Type      [16] IMPLICIT OCTET STRING ( SIZE ( 2 ) ),
            pdp-Address      [17] IMPLICIT OCTET STRING ( SIZE ( 1..16 ) ) OPTIONAL,
            qos-Subscribed      [18] IMPLICIT OCTET STRING ( SIZE ( 3 ) ),
            vplmnAddressAllowed      [19] IMPLICIT NULL OPTIONAL,
            apn      [20] IMPLICIT OCTET STRING ( SIZE ( 2..63 ) ),
            extensionContainer      [21] IMPLICIT SEQUENCE {
                privateExtensionList      [0] IMPLICIT SEQUENCE ( SIZE ( 1..10 ) ) OF
                    SEQUENCE {
                        extId      MAP-EXTENSION .&extensionId ( {
                            '...'} ) ,
                        extType      MAP-EXTENSION .&ExtensionType ( {
                            '...'} { @extId } ) OPTIONAL} OPTIONAL,
                    pcs-Extensions      [1] IMPLICIT SEQUENCE {
                        ... } OPTIONAL,
                        ... } OPTIONAL,
                        ... } OPTIONAL,
                    ... ,
                    ext-QoS-Subscribed      [0] IMPLICIT OCTET STRING ( SIZE ( 3..15 ) ) OPTIONAL},
                extensionContainer      [2] IMPLICIT SEQUENCE {
                    privateExtensionList      [0] IMPLICIT SEQUENCE ( SIZE ( 1..10 ) ) OF
                        SEQUENCE {
                            extId      MAP-EXTENSION .&extensionId ( {
                                '...'} ) ,
                            extType      MAP-EXTENSION .&ExtensionType ( {
                                '...'} { @extId } ) OPTIONAL} OPTIONAL,
                                pcs-Extensions      [1] IMPLICIT SEQUENCE {
                                    ... } OPTIONAL,
                                    ... } OPTIONAL,
                                    ... } OPTIONAL,
                                ... ,
                                sgsn-CAMEL-SubscriptionInfo      [3] IMPLICIT SEQUENCE {
                                    gprs-CSI      [0] IMPLICIT SEQUENCE {
                                        gprs-CamelTDPDataList      [0] IMPLICIT SEQUENCE ( SIZE ( 1..10 ) ) OF
                                            SEQUENCE {
                                                gprs-TriggerDetectionPoint      [0] IMPLICIT ENUMERATED {
                                                    attach      ( 1 ) ,
                                                    attachChangeOfPosition      ( 2 ) ,
                                                    pdp-ContextEstablishment      ( 11 ) ,
                                                    pdp-ContextEstablishmentAcknowledgement      ( 12 ) ,
                                                    pdp-ContextChangeOfPosition      ( 14 ) ,
                                                    ... } ,
                                                serviceKey      [1] IMPLICIT INTEGER ( 0..2147483647 ) ,
                                                gsmSCF-Address      [2] IMPLICIT OCTET STRING ( SIZE ( 1..20 ) ) (
SIZE ( 1..9 ) ) ,
                                                defaultSessionHandling      [3] IMPLICIT ENUMERATED {
                                                    continueTransaction      ( 0 ) ,
                                                    releaseTransaction      ( 1 ) ,
                                                    ... } ,
                                                extensionContainer      [4] IMPLICIT SEQUENCE {
                                                    privateExtensionList      [0] IMPLICIT SEQUENCE ( SIZE ( 1..10 ) ) OF
                                                        SEQUENCE {
                                                            extId      MAP-EXTENSION .&extensionId ( {

```

```

        ... } ) ,
        extType    MAP-EXTENSION .&ExtensionType ( {
            '... } { @extId } ) OPTIONAL} OPTIONAL,
        pcs-Extensions [1] IMPLICIT SEQUENCE {
            ... } OPTIONAL,
            ... } OPTIONAL,
            ... } ,
        camelCapabilityHandling [1] IMPLICIT INTEGER ( 1..16 ) ,
        extensionContainer [2] IMPLICIT SEQUENCE {
            privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE (1..10 ) ) OF
                SEQUENCE {
                    extId    MAP-EXTENSION .&extensionId ( {
                        '... } ) ,
                    extType    MAP-EXTENSION .&ExtensionType ( {
                        '... } { @extId } ) OPTIONAL} OPTIONAL,
                    pcs-Extensions [1] IMPLICIT SEQUENCE {
                        ... } OPTIONAL,
                        ... } OPTIONAL,
                    notificationToCSE [3] IMPLICIT NULL OPTIONAL,
                    csiActive [4] IMPLICIT NULL OPTIONAL,
                    ... } OPTIONAL,
                sms-CSI [1] IMPLICIT SEQUENCE {
                    sms-CAMEL-TDP-DataList [0] IMPLICIT SEQUENCE ( SIZE (1..10 ) ) OF
                        SEQUENCE {
                            sms-TriggerDetectionPoint [0] IMPLICIT ENUMERATED {
                                sms-CollectedInfo ( 1 ) ,
                                ... } ,
                            serviceKey [1] IMPLICIT INTEGER ( 0..2147483647 ) ,
                            gsmSCF-Address [2] IMPLICIT OCTET STRING ( SIZE (1..20 ) ) (
SIZE (1..9 ) ) ,
                            defaultSMS-Handling [3] IMPLICIT ENUMERATED {
                                continueTransaction ( 0 ) ,
                                releaseTransaction ( 1 ) ,
                                ... } ,
                            extensionContainer [4] IMPLICIT SEQUENCE {
                                privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE (1..10 ) ) OF
                                    SEQUENCE {
                                        extId    MAP-EXTENSION .&extensionId ( {
                                            '... } ) ,
                                        extType    MAP-EXTENSION .&ExtensionType ( {
                                            '... } { @extId } ) OPTIONAL} OPTIONAL,
                                        pcs-Extensions [1] IMPLICIT SEQUENCE {
                                            ... } OPTIONAL,
                                            ... } OPTIONAL,
                                        ... } ,
                                    camelCapabilityHandling [1] IMPLICIT INTEGER ( 1..16 ) ,
                                    extensionContainer [2] IMPLICIT SEQUENCE {
                                        privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE (1..10 ) ) OF
                                            SEQUENCE {
                                                extId    MAP-EXTENSION .&extensionId ( {
                                                    '... } ) ,
                                                extType    MAP-EXTENSION .&ExtensionType ( {
                                                    '... } { @extId } ) OPTIONAL} OPTIONAL,
                                                pcs-Extensions [1] IMPLICIT SEQUENCE {
                                                    ... } OPTIONAL,
                                                    ... } OPTIONAL,
                                                notificationToCSE [3] IMPLICIT NULL OPTIONAL,
                                                csiActive [4] IMPLICIT NULL OPTIONAL,
                                                ... } OPTIONAL,
                                            extensionContainer [2] IMPLICIT SEQUENCE {
                                                privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE (1..10 ) ) OF
                                                    SEQUENCE {
                                                        extId    MAP-EXTENSION .&extensionId ( {
                                                            '... } ) ,
                                                        extType    MAP-EXTENSION .&ExtensionType ( {
                                                            '... } { @extId } ) OPTIONAL} OPTIONAL,
                                                        pcs-Extensions [1] IMPLICIT SEQUENCE {
                                                            ... } OPTIONAL,
                                                            ... } OPTIONAL,
                                                            ... } OPTIONAL} OPTIONAL,
                                                roamingRestrictedInSgsnDueToUnsupportedFeature [23] IMPLICIT NULL OPTIONAL,
                                                networkAccessMode [24] IMPLICIT ENUMERATED {
                                                    bothMSCAndSGSN ( 0 ) ,
                                                    onlyMSC ( 1 ) ,

```

```

onlySGSN          ( 2 ),
... } OPTIONAL,
lsaInformation          [25] IMPLICIT SEQUENCE {
completeDataListIncluded  NULL OPTIONAL,
lsaOnlyAccessIndicator    [1] IMPLICIT ENUMERATED {
    accessOutsideLSAsAllowed    ( 0 ),
    accessOutsideLSAsRestricted  ( 1 ) } OPTIONAL,
lsaDataList              [2] IMPLICIT SEQUENCE ( SIZE ( 1..20 ) ) OF
SEQUENCE {
    lsaIdentity              [0] IMPLICIT OCTET STRING ( SIZE ( 3 ) ),
    lsaPriority              [1] IMPLICIT OCTET STRING ( SIZE ( 1 ) ),
    lsaActiveModeIndicator   [2] IMPLICIT NULL OPTIONAL,
    lsaActiveModeSupportIndicator [3] IMPLICIT NULL OPTIONAL,
    extensionContainer       [4] IMPLICIT SEQUENCE {
        privateExtensionList  [0] IMPLICIT SEQUENCE ( SIZE ( 1..10 ) ) OF
SEQUENCE {
            extId          MAP-EXTENSION .&extensionId ( {
                '...' ) ,
            extType       MAP-EXTENSION .&ExtensionType ( {
                '...' { @extId } ) OPTIONAL } OPTIONAL,
            pcs-Extensions [1] IMPLICIT SEQUENCE {
                ... } OPTIONAL,
                ... } OPTIONAL,
                ... } OPTIONAL,
                ... } OPTIONAL,
            extensionContainer [3] IMPLICIT SEQUENCE {
                privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE ( 1..10 ) ) OF
SEQUENCE {
                    extId          MAP-EXTENSION .&extensionId ( {
                        '...' ) ,
                    extType       MAP-EXTENSION .&ExtensionType ( {
                        '...' { @extId } ) OPTIONAL } OPTIONAL,
                    pcs-Extensions [1] IMPLICIT SEQUENCE {
                        ... } OPTIONAL,
                        ... } OPTIONAL,
                        ... } OPTIONAL,
                        ... } OPTIONAL,
                    lmu-Indicator [21] IMPLICIT NULL OPTIONAL,
                    lcsInformation [22] IMPLICIT SEQUENCE {
                        gmlc-List [0] IMPLICIT SEQUENCE ( SIZE ( 1..5 ) ) OF
OCTET STRING ( SIZE ( 1..20 ) ) ( SIZE ( 1..9 ) ) OPTIONAL,
                        lcs-PrivacyExceptionList [1] IMPLICIT SEQUENCE ( SIZE ( 1..4 ) ) OF
SEQUENCE {
                            ss-Code          OCTET STRING ( SIZE ( 1 ) ),
                            ss-Status       OCTET STRING ( SIZE ( 1..5 ) ),
                            privacyVerificationByMSUser [0] IMPLICIT NULL OPTIONAL,
                            externalClientList [1] IMPLICIT SEQUENCE ( SIZE ( 0..5 ) ) OF
SEQUENCE {
                                clientIdentity SEQUENCE {
                                    externalAddress [0] IMPLICIT OCTET STRING ( SIZE ( 1..20 ) )
                                }
                            }
                        }
                    extensionContainer [1] IMPLICIT SEQUENCE {
                        privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE ( 1..10 ) ) OF
SEQUENCE {
                            extId          MAP-EXTENSION .&extensionId ( {
                                '...' ) ,
                            extType       MAP-EXTENSION .&ExtensionType ( {
                                '...' { @extId } ) OPTIONAL } OPTIONAL,
                            pcs-Extensions [1] IMPLICIT SEQUENCE {
                                ... } OPTIONAL,
                                ... } OPTIONAL,
                                ... } OPTIONAL,
                                ... } OPTIONAL,
                            gmlc-Restriction [0] IMPLICIT ENUMERATED {
                                gmlc-List ( 0 ),
                                home-Country ( 1 ) } OPTIONAL,
                            notificationToMSUser [1] IMPLICIT ENUMERATED {
                                notification ( 0 ),
                                notificationWithPrivacyVerification ( 1 ) } OPTIONAL,
                            extensionContainer [2] IMPLICIT SEQUENCE {
                                privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE ( 1..10 ) ) OF
SEQUENCE {
                                    extId          MAP-EXTENSION .&extensionId ( {
                                        '...' ) ,
                                    extType       MAP-EXTENSION .&ExtensionType ( {
                                        '...' { @extId } ) OPTIONAL } OPTIONAL,
                                    pcs-Extensions [1] IMPLICIT SEQUENCE {
                                        ... } OPTIONAL,
                                        ... } OPTIONAL,
                                        ... } OPTIONAL,
                                        ... } OPTIONAL,
                                }
                            }
                        }
                    }
                }
            }
        }
    }
}
OPTIONAL,

```

```

        ... } OPTIONAL,
        ... } OPTIONAL,
    plmnClientList [2] IMPLICIT SEQUENCE ( SIZE (1..5 ) ) OF
        ENUMERATED {
            broadcastService (0 ),
            o-andM-HPLMN (1 ),
            o-andM-VPLMN (2 ),
            anonymousLocation (3 ),
            targetMSsubscribedService (4 ),
            ... } OPTIONAL,
    extensionContainer [3] IMPLICIT SEQUENCE {
        privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE (1..10 ) ) OF
            SEQUENCE {
                extId MAP-EXTENSION .&extensionId ( {
                    '...'} ) ,
                extType MAP-EXTENSION .&ExtensionType ( {
                    '...'} { @extId } ) OPTIONAL} OPTIONAL,
        pcs-Extensions [1] IMPLICIT SEQUENCE {
            ... } OPTIONAL,
            ... } OPTIONAL,
        ... } OPTIONAL,
    molr-List [2] IMPLICIT SEQUENCE ( SIZE (1..3 ) ) OF
        SEQUENCE {
            ss-Code OCTET STRING ( SIZE (1 ) ),
            ss-Status OCTET STRING ( SIZE (1..5 ) ),
            extensionContainer [0] IMPLICIT SEQUENCE {
                privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE (1..10 ) ) OF
                    SEQUENCE {
                        extId MAP-EXTENSION .&extensionId ( {
                            '...'} ) ,
                        extType MAP-EXTENSION .&ExtensionType ( {
                            '...'} { @extId } ) OPTIONAL} OPTIONAL,
                pcs-Extensions [1] IMPLICIT SEQUENCE {
                    ... } OPTIONAL,
                    ... } OPTIONAL,
                    ... } OPTIONAL,
            ... } OPTIONAL,
            istAlertTimer [26] IMPLICIT INTEGER ( 15..255 )
        OPTIONAL,
        superChargerSupportedInHLR [27] IMPLICIT OCTET STRING ( SIZE (1..6 )
    ) OPTIONAL}
    RESULT
        insertSubscriberDataRes SEQUENCE {
            teleserviceList [1] IMPLICIT SEQUENCE ( SIZE (1..20 ) ) OF
                OCTET STRING ( SIZE (1..5 ) ) OPTIONAL,
            bearerServiceList [2] IMPLICIT SEQUENCE ( SIZE (1..50 ) ) OF
                OCTET STRING ( SIZE (1..5 ) ) 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 ),
                interzonalOGCallsBarred (6 ),
                interzonalOGCallsNotToHPLMN-CountryBarred (7 ),
                interzonalOGCallsAndInternationalOGCallsNotToHPLMN-CountryBarred (8 ),
                premiumRateInformationOGCallsBarred (3 ),
                premiumRateEntertainmentOGCallsBarred (4 ),
                ss-AccessBarred (5 ),
                allECT-Barred (9 ),
                chargeableECT-Barred (10 ),
                internationalECT-Barred (11 ),
                interzonalECT-Barred (12 ),
                doublyChargeableECT-Barred (13 ),
                multipleECT-Barred (14 )} ( SIZE (15..32 ) ) OPTIONAL,
            regionalSubscriptionResponse [5] IMPLICIT ENUMERATED {
                networkNode-AreaRestricted (0 ),
                tooManyZoneCodes (1 ),
                zoneCodesConflict (2 ),
                regionalSubscNotSupported (3 )} OPTIONAL,
            supportedCamelPhases [6] IMPLICIT BIT STRING {
                phase1 (0 ),
                phase2 (1 ),
                phase3 (2 )} ( SIZE (1..16 ) ) OPTIONAL,
            extensionContainer [7] IMPLICIT SEQUENCE {
                privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE (1..10 ) ) OF
                    SEQUENCE {
                        extId MAP-EXTENSION .&extensionId ( {

```



```

        ... } ) ,
        extType      MAP-EXTENSION .&ExtensionType ( {
        '... } { @extId } ) OPTIONAL} OPTIONAL,
    pcs-Extensions  [1] IMPLICIT SEQUENCE {
        ... } OPTIONAL,
        ... } 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 {
            ext-BearerService                   [2] IMPLICIT OCTET STRING ( SIZE (1..5 ) ),
            ext-Teleservice                     [3] IMPLICIT OCTET STRING ( SIZE (1..5 ) )} 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,
        vbsGroupIndication                    [7] IMPLICIT NULL OPTIONAL,
        vgcsGroupIndication                   [8] IMPLICIT NULL OPTIONAL,
        camelSubscriptionInfoWithdraw          [9] IMPLICIT NULL OPTIONAL,
        extensionContainer                    [6] IMPLICIT SEQUENCE {
            privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE (1..10 ) ) OF
                SEQUENCE {
                    extId      MAP-EXTENSION .&extensionId ( {
                        '... } ) ,
                    extType    MAP-EXTENSION .&ExtensionType ( {
                        '... } { @extId } ) OPTIONAL} OPTIONAL,
                pcs-Extensions [1] IMPLICIT SEQUENCE {
                    ... } OPTIONAL,
                    ... } OPTIONAL,
                ... ,
            gprsSubscriptionDataWithdraw [10] CHOICE {
                allGPRSData      NULL,
                contextIdList   SEQUENCE ( SIZE (1..50 ) ) OF
                    INTEGER ( 1..50 )} OPTIONAL,
                roamingRestrictedInSgsnDueToUnsupportedFeature [11] IMPLICIT NULL OPTIONAL,
                lsaInformationWithdraw [12] CHOICE {
                    allLSAData      NULL,
                    lsaIdentityList SEQUENCE ( SIZE (1..20 ) ) OF
                        OCTET STRING ( SIZE (3 ) )} OPTIONAL,
                gmlc-ListWithdraw [13] IMPLICIT NULL OPTIONAL,
                istInformationWithdraw [14] IMPLICIT NULL OPTIONAL}
RESULT
    deleteSubscriberDataRes SEQUENCE {
        regionalSubscriptionResponse [0] IMPLICIT ENUMERATED {
            networkNode-AreaRestricted (0 ),
            tooManyZoneCodes (1 ),
            zoneCodesConflict (2 ),
            regionalSubscNotSupported (3 )} OPTIONAL,
        extensionContainer SEQUENCE {
            privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE (1..10 ) ) OF
                SEQUENCE {
                    extId      MAP-EXTENSION .&extensionId ( {
                        '... } ) ,
                    extType    MAP-EXTENSION .&ExtensionType ( {
                        '... } { @extId } ) OPTIONAL} OPTIONAL,
                pcs-Extensions [1] IMPLICIT SEQUENCE {
                    ... } OPTIONAL,
                    ... } OPTIONAL,
                ... }
    ERRORS {
        -- dataMissing -- localValue : 35,
        -- unexpectedDataValue -- localValue : 36,
        -- unidentifiedSubscriber -- localValue : 5}
 ::= localValue : 8

```

```

reset      OPERATION
  ARGUMENT
    resetArg SEQUENCE {
      hlr-Number  OCTET STRING ( SIZE (1..20 ) ) ( 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,
      extensionContainer SEQUENCE {
        privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE (1..10 ) ) OF
          SEQUENCE {
            extId      MAP-EXTENSION .&extensionId ( {
              '...' ) ,
            extType    MAP-EXTENSION .&ExtensionType ( {
              '...' { @extId } ) OPTIONAL} OPTIONAL,
            pcs-Extensions [1] IMPLICIT SEQUENCE {
              ... } OPTIONAL,
            ... } OPTIONAL,
          ... ,
        vlr-Capability [6] IMPLICIT SEQUENCE {
          supportedCamelPhases [0] IMPLICIT BIT STRING {
            phase1 (0 ),
            phase2 (1 ),
            phase3 (2 )} ( SIZE (1..16 ) ) OPTIONAL,
          extensionContainer SEQUENCE {
            privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE (1..10 ) ) OF
              SEQUENCE {
                extId      MAP-EXTENSION .&extensionId ( {
                  '...' ) ,
                extType    MAP-EXTENSION .&ExtensionType ( {
                  '...' { @extId } ) OPTIONAL} OPTIONAL,
                pcs-Extensions [1] IMPLICIT SEQUENCE {
                  ... } OPTIONAL,
                ... } OPTIONAL,
              ... ,
            solsaSupportIndicator [2] IMPLICIT NULL OPTIONAL,
            istSupportIndicator [1] IMPLICIT ENUMERATED {
              basicISTSupported (0 ),
              istCommandSupported (1 ),
              ... } OPTIONAL,
            superChargerSupportedInServingNetworkEntity [3] CHOICE {
              sendSubscriberData [0] IMPLICIT NULL,
              subscriberDataStored [1] IMPLICIT OCTET STRING ( SIZE (1..6 ) )} OPTIONAL}
        OPTIONAL}
      RESULT
        restoreDataRes SEQUENCE {
          hlr-Number  OCTET STRING ( SIZE (1..20 ) ) ( SIZE (1..9 ) ),
          msNotReachable NULL OPTIONAL,
          extensionContainer SEQUENCE {
            privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE (1..10 ) ) OF
              SEQUENCE {
                extId      MAP-EXTENSION .&extensionId ( {
                  '...' ) ,
                extType    MAP-EXTENSION .&ExtensionType ( {
                  '...' { @extId } ) OPTIONAL} OPTIONAL,
                pcs-Extensions [1] IMPLICIT SEQUENCE {
                  ... } OPTIONAL,
                ... } 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,
    extensionContainer  [4] IMPLICIT SEQUENCE {
        privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE (1..10 ) ) OF
            SEQUENCE {
                extId        MAP-EXTENSION .&extensionId ( {
                    '...' ) ,
                extType      MAP-EXTENSION .&ExtensionType ( {
                    '...' { @extId } ) OPTIONAL} OPTIONAL,
                pcs-Extensions [1] IMPLICIT SEQUENCE {
                    ... } OPTIONAL,
                ... } OPTIONAL,
            ... }
RESULT
    activateTraceModeRes SEQUENCE {
        extensionContainer [0] IMPLICIT SEQUENCE {
            privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE (1..10 ) ) OF
                SEQUENCE {
                    extId        MAP-EXTENSION .&extensionId ( {
                        '...' ) ,
                    extType      MAP-EXTENSION .&ExtensionType ( {
                        '...' { @extId } ) OPTIONAL} OPTIONAL,
                    pcs-Extensions [1] IMPLICIT SEQUENCE {
                        ... } OPTIONAL,
                    ... } 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 ) ),
        extensionContainer  [2] IMPLICIT SEQUENCE {
            privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE (1..10 ) ) OF
                SEQUENCE {
                    extId        MAP-EXTENSION .&extensionId ( {
                        '...' ) ,
                    extType      MAP-EXTENSION .&ExtensionType ( {
                        '...' { @extId } ) OPTIONAL} OPTIONAL,
                    pcs-Extensions [1] IMPLICIT SEQUENCE {
                        ... } OPTIONAL,
                    ... } OPTIONAL,
                ... }
RESULT
    deactivateTraceModeRes SEQUENCE {
        extensionContainer [0] IMPLICIT SEQUENCE {
            privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE (1..10 ) ) OF
                SEQUENCE {
                    extId        MAP-EXTENSION .&extensionId ( {
                        '...' ) ,
                    extType      MAP-EXTENSION .&ExtensionType ( {
                        '...' { @extId } ) OPTIONAL} OPTIONAL,
                    pcs-Extensions [1] IMPLICIT SEQUENCE {
                        ... } OPTIONAL,
                    ... } OPTIONAL,
                ... }
ERRORS {
    -- systemFailure -- localValue : 34,
    -- dataMissing -- localValue : 35,
    -- unexpectedDataValue -- localValue : 36,
    -- facilityNotSupported -- localValue : 21,
    -- unidentifiedSubscriber -- localValue : 5}
 ::= localValue : 51

```

```

sendIMSI OPERATION
ARGUMENT
  msisdn      OCTET STRING ( SIZE ( 1..20 ) ) ( 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..20 ) ) ( SIZE ( 1..9
  ) ),
    cug-CheckInfo [1] IMPLICIT SEQUENCE {
      cug-Interlock OCTET STRING ( SIZE ( 4 ) ),
      cug-OutgoingAccess NULL OPTIONAL,
      extensionContainer SEQUENCE {
        privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE ( 1..10 ) ) OF
          SEQUENCE {
            extId MAP-EXTENSION .&extensionId ( {
              '... } ) ,
            extType MAP-EXTENSION .&ExtensionType ( {
              '... } { @extId } ) OPTIONAL} OPTIONAL,
        pcs-Extensions [1] IMPLICIT SEQUENCE {
          ... } OPTIONAL,
          ... } OPTIONAL,
          ... } OPTIONAL,
          ... } OPTIONAL,
        numberOfForwarding [2] IMPLICIT INTEGER ( 1..5 ) OPTIONAL,
        interrogationType [3] IMPLICIT ENUMERATED {
          basicCall ( 0 ) ,
          forwarding ( 1 ) } ,
        or-Interrogation [4] IMPLICIT NULL OPTIONAL,
        or-Capability [5] IMPLICIT INTEGER ( 1..127 ) OPTIONAL,
        gsmc-Address [6] IMPLICIT OCTET STRING ( SIZE ( 1..20 ) ) ( SIZE ( 1..9
      ) ),
        callReferenceNumber [7] IMPLICIT OCTET STRING ( SIZE ( 1..8 ) ) OPTIONAL,
        forwardingReason [8] IMPLICIT ENUMERATED {
          notReachable ( 0 ) ,
          busy ( 1 ) ,
          noReply ( 2 ) } OPTIONAL,
        basicServiceGroup [9] CHOICE {
          ext-BearerService [2] IMPLICIT OCTET STRING ( SIZE ( 1..5 ) ) ,
          ext-Teleservice [3] IMPLICIT OCTET STRING ( SIZE ( 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 ) ) ,
          extensionContainer SEQUENCE {
            privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE ( 1..10 ) ) OF
              SEQUENCE {
                extId MAP-EXTENSION .&extensionId ( {
                  '... } ) ,
                extType MAP-EXTENSION .&ExtensionType ( {
                  '... } { @extId } ) OPTIONAL} OPTIONAL,
            pcs-Extensions [1] IMPLICIT SEQUENCE {
              ... } OPTIONAL,
              ... } OPTIONAL,
              ... } OPTIONAL,
              ... } OPTIONAL,
            camelInfo [11] IMPLICIT SEQUENCE {
              supportedCamelPhases BIT STRING {
                phase1 ( 0 ) ,
                phase2 ( 1 ) ,
                phase3 ( 2 ) } ( SIZE ( 1..16 ) ) ,
              suppress-T-CSI NULL OPTIONAL,
              extensionContainer SEQUENCE {
                privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE ( 1..10 ) ) OF
                  SEQUENCE {
                    extId MAP-EXTENSION .&extensionId ( {
                      '... } ) ,
                    extType MAP-EXTENSION .&ExtensionType ( {
                      '... } { @extId } ) OPTIONAL} OPTIONAL,

```

```

    pcs-Extensions          [1] IMPLICIT SEQUENCE {
        ... } OPTIONAL,
        ... } OPTIONAL,
        ... } OPTIONAL,
    suppressionOfAnnouncement [12] IMPLICIT NULL OPTIONAL,
    extensionContainer        [13] IMPLICIT SEQUENCE {
        privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE (1..10) ) OF
            SEQUENCE {
                extId          MAP-EXTENSION .&extensionId ( {
                    '...'} ) ,
                extType        MAP-EXTENSION .&ExtensionType ( {
                    '...'} { @extId } ) OPTIONAL} OPTIONAL,
        pcs-Extensions        [1] IMPLICIT SEQUENCE {
            ... } OPTIONAL,
            ... } OPTIONAL,
        ... }
    alertingPattern          [14] IMPLICIT OCTET STRING ( SIZE (1) ) OPTIONAL,
    ccbs-Call                [15] IMPLICIT NULL OPTIONAL,
    supportedCCBS-Phase      [16] IMPLICIT INTEGER ( 1..127 ) OPTIONAL,
    additionalSignalInfo     [17] IMPLICIT SEQUENCE {
        ext-ProtocolId        ENUMERATED {
            ets-300356 (1) ,
            ... } ,
        signalInfo            OCTET STRING ( SIZE (1..200) ) ,
        extensionContainer    SEQUENCE {
            privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE (1..10) ) OF
                SEQUENCE {
                    extId          MAP-EXTENSION .&extensionId ( {
                        '...'} ) ,
                    extType        MAP-EXTENSION .&ExtensionType ( {
                        '...'} { @extId } ) OPTIONAL} OPTIONAL,
                pcs-Extensions    [1] IMPLICIT SEQUENCE {
                    ... } OPTIONAL,
                    ... } OPTIONAL,
                    ... } OPTIONAL,
            istSupportIndicator   [18] IMPLICIT ENUMERATED {
                basicISTSupported (0) ,
                istCommandSupported (1) ,
                ... } OPTIONAL,
            pre-pagingSupported [19] IMPLICIT NULL OPTIONAL,
            callDiversionTreatmentIndicator [20] IMPLICIT OCTET STRING ( SIZE (1) ) OPTIONAL}
    RESULT
    sendRoutingInfoRes [3] IMPLICIT SEQUENCE {
        imsi                [9] IMPLICIT OCTET STRING ( SIZE (3..8) ) OPTIONAL,
        extendedRoutingInfo CHOICE {
            routingInfo      CHOICE {
                roamingNumber OCTET STRING ( SIZE (1..20) ) ( SIZE (1..9) ) ,
                forwardingData SEQUENCE {
                    forwardedToNumber [5] IMPLICIT OCTET STRING ( SIZE (1..20) ) ( SIZE (1..9) )
                }
            }
        }
    ) OPTIONAL,
        forwardedToSubaddress [4] IMPLICIT OCTET STRING ( SIZE (1..21) ) OPTIONAL,
        forwardingOptions     [6] IMPLICIT OCTET STRING ( SIZE (1) ) OPTIONAL,
        extensionContainer    [7] IMPLICIT SEQUENCE {
            privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE (1..10) ) OF
                SEQUENCE {
                    extId          MAP-EXTENSION .&extensionId ( {
                        '...'} ) ,
                    extType        MAP-EXTENSION .&ExtensionType ( {
                        '...'} { @extId } ) OPTIONAL} OPTIONAL,
            pcs-Extensions      [1] IMPLICIT SEQUENCE {
                ... } OPTIONAL,
                ... } OPTIONAL,
                ... }
        }
    camelRoutingInfo [8] IMPLICIT SEQUENCE {
        forwardingData        SEQUENCE {
            forwardedToNumber [5] IMPLICIT OCTET STRING ( SIZE (1..20) ) ( SIZE (1..9) )
        }
    ) OPTIONAL,
        forwardedToSubaddress [4] IMPLICIT OCTET STRING ( SIZE (1..21) ) OPTIONAL,
        forwardingOptions     [6] IMPLICIT OCTET STRING ( SIZE (1) ) OPTIONAL,
        extensionContainer    [7] IMPLICIT SEQUENCE {
            privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE (1..10) ) OF
                SEQUENCE {
                    extId          MAP-EXTENSION .&extensionId ( {
                        '...'} ) ,
                    extType        MAP-EXTENSION .&ExtensionType ( {

```

```

        ... } { @extId } ) OPTIONAL} OPTIONAL,
pcs-Extensions [1] IMPLICIT SEQUENCE {
    ... } OPTIONAL,
    ... } OPTIONAL,
    ... } OPTIONAL,
gmscCamelSubscriptionInfo [0] IMPLICIT SEQUENCE {
t-CSI [0] IMPLICIT SEQUENCE {
t-BcsmCamelTDPDataList SEQUENCE ( SIZE (1..10 ) ) OF
SEQUENCE {
t-BcsmTriggerDetectionPoint ENUMERATED {
termAttemptAuthorized (12 ),
... ,
tBusy (13 ),
tNoAnswer (14 )},
serviceKey INTEGER ( 0..2147483647 ),
gsmSCF-Address [0] IMPLICIT OCTET STRING ( SIZE (1..20 ) )
( SIZE (1..9 ) ),
defaultCallHandling [1] IMPLICIT ENUMERATED {
continueCall (0 ),
releaseCall (1 ),
... },
extensionContainer [2] IMPLICIT SEQUENCE {
privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE (1..10 ) ) OF
SEQUENCE {
extId MAP-EXTENSION .&extensionId ( {
'...' } ),
extType MAP-EXTENSION .&ExtensionType ( {
'...' } { @extId } ) OPTIONAL} OPTIONAL,
pcs-Extensions [1] IMPLICIT SEQUENCE {
... } OPTIONAL,
... } OPTIONAL,
... },
extensionContainer SEQUENCE {
privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE (1..10 ) ) OF
SEQUENCE {
extId MAP-EXTENSION .&extensionId ( {
'...' } ),
extType MAP-EXTENSION .&ExtensionType ( {
'...' } { @extId } ) OPTIONAL} OPTIONAL,
pcs-Extensions [1] IMPLICIT SEQUENCE {
... } OPTIONAL,
... } OPTIONAL,
... ,
camelCapabilityHandling [0] IMPLICIT INTEGER ( 1..16 ) OPTIONAL,
notificationToCSE [1] IMPLICIT NULL OPTIONAL,
csi-Active [2] IMPLICIT NULL OPTIONAL,
o-CSI [1] IMPLICIT SEQUENCE {
o-BcsmCamelTDPDataList SEQUENCE ( SIZE (1..10 ) ) OF
SEQUENCE {
o-BcsmTriggerDetectionPoint ENUMERATED {
collectedInfo (2 ),
... ,
routeSelectFailure (4 )},
serviceKey INTEGER ( 0..2147483647 ),
gsmSCF-Address [0] IMPLICIT OCTET STRING ( SIZE (1..20 ) )
( SIZE (1..9 ) ),
defaultCallHandling [1] IMPLICIT ENUMERATED {
continueCall (0 ),
releaseCall (1 ),
... },
extensionContainer [2] IMPLICIT SEQUENCE {
privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE (1..10 ) ) OF
SEQUENCE {
extId MAP-EXTENSION .&extensionId ( {
'...' } ),
extType MAP-EXTENSION .&ExtensionType ( {
'...' } { @extId } ) OPTIONAL} OPTIONAL,
pcs-Extensions [1] IMPLICIT SEQUENCE {
... } OPTIONAL,
... } OPTIONAL,
... },
extensionContainer SEQUENCE {
privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE (1..10 ) ) OF
SEQUENCE {
extId MAP-EXTENSION .&extensionId ( {
'...' } ),

```

```

        extType      MAP-EXTENSION .&ExtensionType ( {
            ... } { @extId } ) OPTIONAL} OPTIONAL,
    pcs-Extensions  [1] IMPLICIT SEQUENCE {
        ... } OPTIONAL,
        ... } OPTIONAL,
    ... ,
    camelCapabilityHandling [0] IMPLICIT INTEGER ( 1..16 ) OPTIONAL,
    notificationToCSE      [1] IMPLICIT NULL OPTIONAL,
    csiActive               [2] IMPLICIT NULL OPTIONAL} OPTIONAL,
    extensionContainer      [2] IMPLICIT SEQUENCE {
    privateExtensionList    [0] IMPLICIT SEQUENCE ( SIZE (1..10 ) ) OF
    SEQUENCE {
        extId      MAP-EXTENSION .&extensionId ( {
            ... } ) ,
        extType    MAP-EXTENSION .&ExtensionType ( {
            ... } { @extId } ) OPTIONAL} OPTIONAL,
    pcs-Extensions [1] IMPLICIT SEQUENCE {
        ... } OPTIONAL,
        ... } OPTIONAL,
    ... ,
    o-BcsmCamelTDP-CriteriaList [3] IMPLICIT SEQUENCE ( SIZE (1..10 ) ) OF
    SEQUENCE {
        o-BcsmTriggerDetectionPoint  ENUMERATED {
            collectedInfo ( 2 ) ,
            ... ,
            routeSelectFailure ( 4 ) } ,
        destinationNumberCriteria [0] IMPLICIT SEQUENCE {
            matchType [0] IMPLICIT ENUMERATED {
                inhibiting ( 0 ) ,
                enabling ( 1 ) } ,
            destinationNumberList [1] IMPLICIT SEQUENCE ( SIZE (1..10 ) ) OF
            OCTET STRING ( SIZE (1..20 ) ) ( SIZE (1..9 ) ) OPTIONAL,
            destinationNumberLengthList [2] IMPLICIT SEQUENCE ( SIZE (1..3 ) ) OF
            INTEGER ( 1..15 ) OPTIONAL,
            ... } OPTIONAL,
        basicServiceCriteria [1] IMPLICIT SEQUENCE ( SIZE (1..5 ) ) OF
        CHOICE {
            ext-BearerService [2] IMPLICIT OCTET STRING ( SIZE (1..5 ) ) ,
            ext-Teleservice [3] IMPLICIT OCTET STRING ( SIZE (1..5 ) ) }
OPTIONAL,

        callTypeCriteria [2] IMPLICIT ENUMERATED {
            forwarded ( 0 ) ,
            notForwarded ( 1 ) } OPTIONAL,
        ... ,
        o-CauseValueCriteria [3] IMPLICIT SEQUENCE ( SIZE (1..5 ) ) OF
        OCTET STRING ( SIZE (1 ) ) OPTIONAL,
        extensionContainer [4] IMPLICIT SEQUENCE {
        privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE (1..10 ) ) OF
        SEQUENCE {
            extId      MAP-EXTENSION .&extensionId ( {
                ... } ) ,
            extType    MAP-EXTENSION .&ExtensionType ( {
                ... } { @extId } ) OPTIONAL} OPTIONAL,
        pcs-Extensions [1] IMPLICIT SEQUENCE {
            ... } OPTIONAL,
            ... } OPTIONAL} OPTIONAL,
    t-BCSM-CAMEL-TDP-CriteriaList [4] IMPLICIT SEQUENCE ( SIZE (1..10 ) ) OF
    SEQUENCE {
        t-BCSM-TriggerDetectionPoint  ENUMERATED {
            termAttemptAuthorized ( 12 ) ,
            ... ,
            tBusy ( 13 ) ,
            tNoAnswer ( 14 ) } ,
        basicServiceCriteria [0] IMPLICIT SEQUENCE ( SIZE (1..5 ) ) OF
        CHOICE {
            ext-BearerService [2] IMPLICIT OCTET STRING ( SIZE (1..5 ) ) ,
            ext-Teleservice [3] IMPLICIT OCTET STRING ( SIZE (1..5 ) ) }
OPTIONAL,

        t-CauseValueCriteria [1] IMPLICIT SEQUENCE ( SIZE (1..5 ) ) OF
        OCTET STRING ( SIZE (1 ) ) OPTIONAL,
        ... } OPTIONAL,
    d-csi [5] IMPLICIT SEQUENCE {
    dp-AnalysedInfoCriteriaList SEQUENCE ( SIZE (1..10 ) ) OF
    SEQUENCE {
        dialledNumber OCTET STRING ( SIZE (1..20 ) ) ( SIZE (1..9 ) ) ,
        serviceKey INTEGER ( 0..2147483647 ) ,
        gsmSCF-Address OCTET STRING ( SIZE (1..20 ) ) ( SIZE (1..9 ) ) ,
        defaultCallHandling ENUMERATED {

```

```

        continueCall      ( 0 ),
        releaseCall      ( 1 ),
        ... },
    extensionContainer    SEQUENCE {
        privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE ( 1..10 ) ) OF
            SEQUENCE {
                extId      MAP-EXTENSION .&extensionId ( {
                    '...'} ),
                extType    MAP-EXTENSION .&ExtensionType ( {
                    '...'} { @extId } ) OPTIONAL} OPTIONAL,
        pcs-Extensions    [1] IMPLICIT SEQUENCE {
            ... } OPTIONAL,
        ... } OPTIONAL,
        ... },
    camelCapabilityHandling    INTEGER ( 1..16 ),
    extensionContainer        SEQUENCE {
        privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE ( 1..10 ) ) OF
            SEQUENCE {
                extId      MAP-EXTENSION .&extensionId ( {
                    '...'} ),
                extType    MAP-EXTENSION .&ExtensionType ( {
                    '...'} { @extId } ) OPTIONAL} OPTIONAL,
        pcs-Extensions    [1] IMPLICIT SEQUENCE {
            ... } OPTIONAL,
        ... } OPTIONAL,
        ... } OPTIONAL},
    extensionContainer        [1] IMPLICIT SEQUENCE {
        privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE ( 1..10 ) ) OF
            SEQUENCE {
                extId      MAP-EXTENSION .&extensionId ( {
                    '...'} ),
                extType    MAP-EXTENSION .&ExtensionType ( {
                    '...'} { @extId } ) OPTIONAL} OPTIONAL,
        pcs-Extensions    [1] IMPLICIT SEQUENCE {
            ... } OPTIONAL,
        ... } OPTIONAL,
        ... } OPTIONAL},
    cug-CheckInfo            [3] IMPLICIT SEQUENCE {
    cug-Interlock            OCTET STRING ( SIZE ( 4 ) ),
    cug-OutgoingAccess      NULL OPTIONAL,
    extensionContainer       SEQUENCE {
        privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE ( 1..10 ) ) OF
            SEQUENCE {
                extId      MAP-EXTENSION .&extensionId ( {
                    '...'} ),
                extType    MAP-EXTENSION .&ExtensionType ( {
                    '...'} { @extId } ) OPTIONAL} OPTIONAL,
        pcs-Extensions    [1] IMPLICIT SEQUENCE {
            ... } OPTIONAL,
        ... } OPTIONAL,
        ... } OPTIONAL,
        ... } OPTIONAL},
    cugSubscriptionFlag      [6] IMPLICIT NULL OPTIONAL,
    subscriberInfo           [7] IMPLICIT SEQUENCE {
    locationInformation [0] IMPLICIT SEQUENCE {
        ageOfLocationInformation    INTEGER ( 0..32767 ) OPTIONAL,
        geographicalInformation      [0] IMPLICIT OCTET STRING ( SIZE ( 8 ) ) OPTIONAL,
        vlr-number                  [1] IMPLICIT OCTET STRING ( SIZE ( 1..20 ) ) ( SIZE ( 1..9 )
) OPTIONAL,
        locationNumber              [2] IMPLICIT OCTET STRING ( SIZE ( 2..10 ) ) OPTIONAL,
        cellIdOrLAI                 [3] CHOICE {
            cellIdFixedLength [0] IMPLICIT OCTET STRING ( SIZE ( 7 ) ),
            laiFixedLength    [1] IMPLICIT OCTET STRING ( SIZE ( 5 ) )} OPTIONAL,
        extensionContainer          [4] IMPLICIT SEQUENCE {
            privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE ( 1..10 ) ) OF
                SEQUENCE {
                    extId      MAP-EXTENSION .&extensionId ( {
                        '...'} ),
                    extType    MAP-EXTENSION .&ExtensionType ( {
                        '...'} { @extId } ) OPTIONAL} OPTIONAL,
                pcs-Extensions [1] IMPLICIT SEQUENCE {
                    ... } OPTIONAL,
                ... } OPTIONAL,
                ... },
        ... },
    ... },

```



```

selectedLSA-Id          [5] IMPLICIT OCTET STRING ( SIZE ( 3 ) ) OPTIONAL,
msc-Number              [6] IMPLICIT OCTET STRING ( SIZE ( 1..20 ) ) ( SIZE ( 1..9 )
) OPTIONAL,
geodeticInformation     [7] IMPLICIT OCTET STRING ( SIZE ( 10 ) ) OPTIONAL}
OPTIONAL,
subscriberState        [1] CHOICE {
  assumedIdle          [0] IMPLICIT NULL,
  camelBusy            [1] IMPLICIT NULL,
  netDetNotReachable  ENUMERATED {
    msPurged          ( 0 ),
    imsiDetached      ( 1 ),
    restrictedArea     ( 2 ),
    notRegistered      ( 3 )},
  notProvidedFromVLR  [2] IMPLICIT NULL} OPTIONAL,
extensionContainer     [2] IMPLICIT SEQUENCE {
  privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE ( 1..10 ) ) OF
  SEQUENCE {
    extId              MAP-EXTENSION .&extensionId ( {
      '...' ) ,
    extType            MAP-EXTENSION .&ExtensionType ( {
      '...' { @extId } ) OPTIONAL} OPTIONAL,
  pcs-Extensions      [1] IMPLICIT SEQUENCE {
    ... } OPTIONAL,
    ... } OPTIONAL,
  ... } OPTIONAL,
ss-List                [1] IMPLICIT SEQUENCE ( SIZE ( 1..30 ) ) OF
OCTET STRING ( SIZE ( 1 ) ) OPTIONAL,
basicService           [5] CHOICE {
  ext-BearerService    [2] IMPLICIT OCTET STRING ( SIZE ( 1..5 ) ),
  ext-Teleservice      [3] IMPLICIT OCTET STRING ( SIZE ( 1..5 ) )} OPTIONAL,
forwardingInterrogationRequired [4] IMPLICIT NULL OPTIONAL,
vmsc-Address           [2] IMPLICIT OCTET STRING ( SIZE ( 1..20 ) ) ( SIZE ( 1..9
) ) OPTIONAL,
extensionContainer     [0] IMPLICIT SEQUENCE {
  privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE ( 1..10 ) ) OF
  SEQUENCE {
    extId              MAP-EXTENSION .&extensionId ( {
      '...' ) ,
    extType            MAP-EXTENSION .&ExtensionType ( {
      '...' { @extId } ) OPTIONAL} OPTIONAL,
  pcs-Extensions      [1] IMPLICIT SEQUENCE {
    ... } OPTIONAL,
    ... } OPTIONAL,
  ... } OPTIONAL,
naea-PreferredCI      [10] IMPLICIT SEQUENCE {
  naea-PreferredCIC    [0] IMPLICIT OCTET STRING ( SIZE ( 3 ) ),
  extensionContainer   [1] IMPLICIT SEQUENCE {
    privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE ( 1..10 ) ) OF
    SEQUENCE {
      extId              MAP-EXTENSION .&extensionId ( {
        '...' ) ,
      extType            MAP-EXTENSION .&ExtensionType ( {
        '...' { @extId } ) OPTIONAL} OPTIONAL,
    pcs-Extensions      [1] IMPLICIT SEQUENCE {
      ... } OPTIONAL,
      ... } OPTIONAL,
    ... } OPTIONAL,
ccbs-Indicators       [11] IMPLICIT SEQUENCE {
  ccbs-Possible        [0] IMPLICIT NULL OPTIONAL,
  keepCCBS-CallIndicator [1] IMPLICIT NULL OPTIONAL,
  extensionContainer   [2] IMPLICIT SEQUENCE {
    privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE ( 1..10 ) ) OF
    SEQUENCE {
      extId              MAP-EXTENSION .&extensionId ( {
        '...' ) ,
      extType            MAP-EXTENSION .&ExtensionType ( {
        '...' { @extId } ) OPTIONAL} OPTIONAL,
    pcs-Extensions      [1] IMPLICIT SEQUENCE {
      ... } OPTIONAL,
      ... } OPTIONAL,
      ... } OPTIONAL,
msisdn                [12] IMPLICIT OCTET STRING ( SIZE ( 1..20 ) ) ( SIZE ( 1..9
) ) OPTIONAL,
numberPortabilityStatus [13] IMPLICIT ENUMERATED {
  notKnownToBePorted  ( 0 ),

```

```

        ownNumberPortedOut          ( 1 ),
        foreignNumberPortedToForeignNetwork ( 2 ),
        ... } OPTIONAL,
    istAlertTimer                    [14] IMPLICIT INTEGER ( 15..255 ) OPTIONAL}
ERRORS {
-- systemFailure -- localValue : 34,
-- dataMissing -- localValue : 35,
-- unexpectedDataValue -- localValue : 36,
-- facilityNotSupported -- localValue : 21,
-- or-NotAllowed -- localValue : 48,
-- unknownSubscriber -- localValue : 1,
-- numberChanged -- localValue : 44,
-- bearerServiceNotProvisioned -- localValue : 10,
-- teleserviceNotProvisioned -- localValue : 11,
-- absentSubscriber -- localValue : 27,
-- busySubscriber -- localValue : 45,
-- noSubscriberReply -- localValue : 46,
-- 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..20) ) ( SIZE (1..9) ),
        msisdn [2] IMPLICIT OCTET STRING ( SIZE (1..20) ) ( 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) ),
            extensionContainer SEQUENCE {
                privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE (1..10) ) OF
                    SEQUENCE {
                        extId MAP-EXTENSION .&extensionId ( {
                            '...' ) ,
                        extType MAP-EXTENSION .&ExtensionType ( {
                            '...' { @extId } ) OPTIONAL} OPTIONAL,
                pcs-Extensions [1] IMPLICIT SEQUENCE {
                    ... } OPTIONAL,
                    ... } OPTIONAL,
                    ... } 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) ),
                    extensionContainer SEQUENCE {
                        privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE (1..10) ) OF
                            SEQUENCE {
                                extId MAP-EXTENSION .&extensionId ( {
                                    '...' ) ,
                                extType MAP-EXTENSION .&ExtensionType ( {
                                    '...' { @extId } ) OPTIONAL} OPTIONAL,
                            pcs-Extensions [1] IMPLICIT SEQUENCE {
                                ... } OPTIONAL,
                                ... } OPTIONAL,
                                ... } OPTIONAL,
                            suppressionOfAnnouncement [7] IMPLICIT NULL OPTIONAL,
                            gmsc-Address [8] IMPLICIT OCTET STRING ( SIZE (1..20) ) ( SIZE (1..9) )
OPTIONAL,
                            callReferenceNumber [9] IMPLICIT OCTET STRING ( SIZE (1..8) ) OPTIONAL,
                            or-Interrogation [10] IMPLICIT NULL OPTIONAL,
                            extensionContainer [11] IMPLICIT SEQUENCE {
                                privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE (1..10) ) OF
                                    SEQUENCE {
                                        extId MAP-EXTENSION .&extensionId ( {
                                            '...' ) ,
                                        extType MAP-EXTENSION .&ExtensionType ( {

```

```

        ... } { @extId } ) OPTIONAL} OPTIONAL,
pcs-Extensions [1] IMPLICIT SEQUENCE {
    ... } OPTIONAL,
    ... } OPTIONAL,
... ,
alertingPattern [12] IMPLICIT OCTET STRING ( SIZE ( 1 ) ) OPTIONAL,
ccbs-Call [13] IMPLICIT NULL OPTIONAL,
supportedCamelPhasesInGMSC [15] IMPLICIT BIT STRING {
    phase1 ( 0 ),
    phase2 ( 1 ),
    phase3 ( 2 ) } ( SIZE ( 1..16 ) ) OPTIONAL,
additionalSignalInfo [14] IMPLICIT SEQUENCE {
    ext-ProtocolId ENUMERATED {
        ets-300356 ( 1 ),
        ... },
    signalInfo OCTET STRING ( SIZE ( 1..200 ) ),
    extensionContainer SEQUENCE {
        privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE ( 1..10 ) ) OF
        SEQUENCE {
            extId MAP-EXTENSION .&extensionId ( {
                '... } ) ,
            extType MAP-EXTENSION .&ExtensionType ( {
                '... } { @extId } ) OPTIONAL} OPTIONAL,
            pcs-Extensions [1] IMPLICIT SEQUENCE {
                ... } OPTIONAL,
                ... } OPTIONAL,
                ... } OPTIONAL,
            orNotSupportedInGMSC [16] IMPLICIT NULL OPTIONAL,
            pre-pagingSupported [17] IMPLICIT NULL OPTIONAL}
RESULT
    provideRoamingNumberRes SEQUENCE {
        roamingNumber OCTET STRING ( SIZE ( 1..20 ) ) ( SIZE ( 1..9 ) ),
        extensionContainer SEQUENCE {
            privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE ( 1..10 ) ) OF
            SEQUENCE {
                extId MAP-EXTENSION .&extensionId ( {
                    '... } ) ,
                extType MAP-EXTENSION .&ExtensionType ( {
                    '... } { @extId } ) OPTIONAL} OPTIONAL,
                pcs-Extensions [1] IMPLICIT SEQUENCE {
                    ... } OPTIONAL,
                    ... } OPTIONAL,
                    ... } OPTIONAL,
                ... }
ERRORS {
    -- systemFailure -- localValue : 34,
    -- dataMissing -- localValue : 35,
    -- unexpectedDataValue -- localValue : 36,
    -- facilityNotSupported -- localValue : 21,
    -- or-NotAllowed -- localValue : 48,
    -- absentSubscriber -- localValue : 27,
    -- noRoamingNumberAvailable -- localValue : 39}
 ::= localValue : 4

resumeCallHandling OPERATION
ARGUMENT
    resumeCallHandlingArg SEQUENCE {
        callReferenceNumber [0] IMPLICIT OCTET STRING ( SIZE ( 1..8 ) ) OPTIONAL,
        basicServiceGroup [1] CHOICE {
            ext-BearerService [2] IMPLICIT OCTET STRING ( SIZE ( 1..5 ) ),
            ext-Teleservice [3] IMPLICIT OCTET STRING ( SIZE ( 1..5 ) )} OPTIONAL,
        forwardingData [2] IMPLICIT SEQUENCE {
            forwardedToNumber [5] IMPLICIT OCTET STRING ( SIZE ( 1..20 ) ) ( SIZE ( 1..9 ) )
OPTIONAL,
            forwardedToSubaddress [4] IMPLICIT OCTET STRING ( SIZE ( 1..21 ) ) OPTIONAL,
            forwardingOptions [6] IMPLICIT OCTET STRING ( SIZE ( 1 ) ) OPTIONAL,
            extensionContainer [7] IMPLICIT SEQUENCE {
                privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE ( 1..10 ) ) OF
                SEQUENCE {
                    extId MAP-EXTENSION .&extensionId ( {
                        '... } ) ,
                    extType MAP-EXTENSION .&ExtensionType ( {
                        '... } { @extId } ) OPTIONAL} OPTIONAL,
                    pcs-Extensions [1] IMPLICIT SEQUENCE {
                        ... } OPTIONAL,
                        ... } OPTIONAL,
                        ... } OPTIONAL,
                    ... } OPTIONAL,
            imsi [3] IMPLICIT OCTET STRING ( SIZE ( 3..8 ) ) OPTIONAL,

```

```

cug-CheckInfo          [4] IMPLICIT SEQUENCE {
  cug-Interlock         OCTET STRING ( SIZE ( 4 ) ),
  cug-OutgoingAccess    NULL OPTIONAL,
  extensionContainer    SEQUENCE {
    privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE ( 1..10 ) ) OF
      SEQUENCE {
        extId           MAP-EXTENSION .&extensionId ( {
          '...' } ),
        extType         MAP-EXTENSION .&ExtensionType ( {
          '...' { @extId } ) OPTIONAL} OPTIONAL,
        pcs-Extensions [1] IMPLICIT SEQUENCE {
          ... } OPTIONAL,
        ... } OPTIONAL,
        ... } OPTIONAL,
  o-CSI                 [5] IMPLICIT SEQUENCE {
    o-BcsmCamelTDPDataList SEQUENCE ( SIZE ( 1..10 ) ) OF
      SEQUENCE {
        o-BcsmTriggerDetectionPoint ENUMERATED {
          collectedInfo ( 2 ),
          ... ,
          routeSelectFailure ( 4 ) },
        serviceKey          INTEGER ( 0..2147483647 ),
        gsmSCF-Address      [0] IMPLICIT OCTET STRING ( SIZE ( 1..20 ) ) ( SIZE
(1..9) ),
        defaultCallHandling [1] IMPLICIT ENUMERATED {
          continueCall ( 0 ),
          releaseCall ( 1 ),
          ... },
        extensionContainer [2] IMPLICIT SEQUENCE {
          privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE ( 1..10 ) ) OF
            SEQUENCE {
              extId           MAP-EXTENSION .&extensionId ( {
                '...' } ),
              extType         MAP-EXTENSION .&ExtensionType ( {
                '...' { @extId } ) OPTIONAL} OPTIONAL,
              pcs-Extensions [1] IMPLICIT SEQUENCE {
                ... } OPTIONAL,
                ... } OPTIONAL,
                ... },
          extensionContainer SEQUENCE {
            privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE ( 1..10 ) ) OF
              SEQUENCE {
                extId           MAP-EXTENSION .&extensionId ( {
                  '...' } ),
                extType         MAP-EXTENSION .&ExtensionType ( {
                  '...' { @extId } ) OPTIONAL} OPTIONAL,
                pcs-Extensions [1] IMPLICIT SEQUENCE {
                  ... } OPTIONAL,
                  ... } OPTIONAL,
                  ... ,
                camelCapabilityHandling [0] IMPLICIT INTEGER ( 1..16 ) OPTIONAL,
                notificationToCSE      [1] IMPLICIT NULL OPTIONAL,
                csiActive               [2] IMPLICIT NULL OPTIONAL} OPTIONAL,
            extensionContainer [7] IMPLICIT SEQUENCE {
              privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE ( 1..10 ) ) OF
                SEQUENCE {
                  extId           MAP-EXTENSION .&extensionId ( {
                    '...' } ),
                  extType         MAP-EXTENSION .&ExtensionType ( {
                    '...' { @extId } ) OPTIONAL} OPTIONAL,
                  pcs-Extensions [1] IMPLICIT SEQUENCE {
                    ... } OPTIONAL,
                    ... } OPTIONAL,
                    ... ,
                  ccbs-Possible [8] IMPLICIT NULL OPTIONAL,
                  msisdn        [9] IMPLICIT OCTET STRING ( SIZE ( 1..20 ) ) ( SIZE ( 1..9 ) )
OPTIONAL,
                  uu-Data      [10] IMPLICIT SEQUENCE {
                    uuIndicator [0] IMPLICIT OCTET STRING ( SIZE ( 1 ) ) OPTIONAL,
                    uui         [1] IMPLICIT OCTET STRING ( SIZE ( 1..131 ) ) OPTIONAL,
                    uusCFInteraction [2] IMPLICIT NULL OPTIONAL,
                    extensionContainer [3] IMPLICIT SEQUENCE {
                      privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE ( 1..10 ) ) OF
                        SEQUENCE {
                          extId           MAP-EXTENSION .&extensionId ( {

```

```

        ... } ) ,
        extType     MAP-EXTENSION .&ExtensionType ( {
            '...' { @extId } ) OPTIONAL} OPTIONAL,
        pcs-Extensions [1] IMPLICIT SEQUENCE {
            ... } OPTIONAL,
            ... } OPTIONAL,
            ... } OPTIONAL,
        allInformationSent [11] IMPLICIT NULL OPTIONAL,
        d-csi [12] IMPLICIT SEQUENCE {
            dp-AnalysedInfoCriteriaList SEQUENCE ( SIZE (1..10 ) ) OF
            SEQUENCE {
                dialledNumber     OCTET STRING ( SIZE (1..20 ) ) ( SIZE (1..9 ) ) ,
                serviceKey         INTEGER ( 0..2147483647 ) ,
                gsmSCF-Address     OCTET STRING ( SIZE (1..20 ) ) ( SIZE (1..9 ) ) ,
                defaultCallHandling ENUMERATED {
                    continueCall (0 ) ,
                    releaseCall (1 ) ,
                    ... } ,
                extensionContainer SEQUENCE {
                    privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE (1..10 ) ) OF
                    SEQUENCE {
                        extId     MAP-EXTENSION .&extensionId ( {
                            '...' ) ,
                            extType     MAP-EXTENSION .&ExtensionType ( {
                                '...' { @extId } ) OPTIONAL} OPTIONAL,
                                pcs-Extensions [1] IMPLICIT SEQUENCE {
                                    ... } OPTIONAL,
                                    ... } OPTIONAL,
                                    ... } OPTIONAL,
                                ... } OPTIONAL,
                                ... }
                    }
                camelCapabilityHandling     INTEGER ( 1..16 ) ,
                extensionContainer SEQUENCE {
                    privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE (1..10 ) ) OF
                    SEQUENCE {
                        extId     MAP-EXTENSION .&extensionId ( {
                            '...' ) ,
                            extType     MAP-EXTENSION .&ExtensionType ( {
                                '...' { @extId } ) OPTIONAL} OPTIONAL,
                                pcs-Extensions [1] IMPLICIT SEQUENCE {
                                    ... } OPTIONAL,
                                    ... } OPTIONAL,
                                    ... } OPTIONAL,
                                ... }
                    }
                ... }
            }
        RESULT
        resumeCallHandlingRes SEQUENCE {
            extensionContainer SEQUENCE {
                privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE (1..10 ) ) OF
                SEQUENCE {
                    extId     MAP-EXTENSION .&extensionId ( {
                        '...' ) ,
                        extType     MAP-EXTENSION .&ExtensionType ( {
                            '...' { @extId } ) OPTIONAL} OPTIONAL,
                            pcs-Extensions [1] IMPLICIT SEQUENCE {
                                ... } OPTIONAL,
                                ... } OPTIONAL,
                                ... } OPTIONAL,
                            ... }
                }
            }
        ERRORS {
            -- forwardingFailed -- localValue : 47,
            -- or-NotAllowed -- localValue : 48,
            -- unexpectedDataValue -- localValue : 36,
            -- dataMissing -- localValue : 35}
        ::= localValue : 6

provideSIWFSNumber OPERATION
ARGUMENT
provideSIWFSNumberArg SEQUENCE {
    gsm-BearerCapability [0] IMPLICIT SEQUENCE {
        protocolId     ENUMERATED {
            gsm-0408 (1 ) ,
            gsm-0806 (2 ) ,
            gsm-BSSMAP (3 ) ,
            ets-300102-1 (4 ) } ,
        signalInfo     OCTET STRING ( SIZE (1..200 ) ) ,
        extensionContainer SEQUENCE {
            privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE (1..10 ) ) OF
            SEQUENCE {
                extId     MAP-EXTENSION .&extensionId ( {

```

```

        ... } ) ,
        extType    MAP-EXTENSION .&ExtensionType ( {
            ... } { @extId } ) OPTIONAL} OPTIONAL,
        pcs-Extensions [1] IMPLICIT SEQUENCE {
            ... } OPTIONAL,
            ... } OPTIONAL,
            ... } ,
        isdn-BearerCapability [1] IMPLICIT SEQUENCE {
        protocolId    ENUMERATED {
            gsm-0408    (1 ) ,
            gsm-0806    (2 ) ,
            gsm-BSSMAP  (3 ) ,
            ets-300102-1 (4 ) } ,
        signalInfo    OCTET STRING ( SIZE (1..200 ) ) ,
        extensionContainer SEQUENCE {
            privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE (1..10 ) ) OF
                SEQUENCE {
                    extId    MAP-EXTENSION .&extensionId ( {
                        ... } ) ,
                    extType    MAP-EXTENSION .&ExtensionType ( {
                        ... } { @extId } ) OPTIONAL} OPTIONAL,
                    pcs-Extensions [1] IMPLICIT SEQUENCE {
                        ... } OPTIONAL,
                        ... } OPTIONAL,
                        ... } ,
            call-Direction [2] IMPLICIT OCTET STRING ( SIZE (1 ) ) ,
            b-Subscriber-Address [3] IMPLICIT OCTET STRING ( SIZE (1..20 ) ) ( SIZE (1..9 ) ) ,
            chosenChannel [4] IMPLICIT SEQUENCE {
            protocolId    ENUMERATED {
                gsm-0408    (1 ) ,
                gsm-0806    (2 ) ,
                gsm-BSSMAP  (3 ) ,
                ets-300102-1 (4 ) } ,
            signalInfo    OCTET STRING ( SIZE (1..200 ) ) ,
            extensionContainer SEQUENCE {
                privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE (1..10 ) ) OF
                    SEQUENCE {
                        extId    MAP-EXTENSION .&extensionId ( {
                            ... } ) ,
                        extType    MAP-EXTENSION .&ExtensionType ( {
                            ... } { @extId } ) OPTIONAL} OPTIONAL,
                            pcs-Extensions [1] IMPLICIT SEQUENCE {
                                ... } OPTIONAL,
                                ... } OPTIONAL,
                                ... } ,
                lowerLayerCompatibility [5] IMPLICIT SEQUENCE {
                protocolId    ENUMERATED {
                    gsm-0408    (1 ) ,
                    gsm-0806    (2 ) ,
                    gsm-BSSMAP  (3 ) ,
                    ets-300102-1 (4 ) } ,
                signalInfo    OCTET STRING ( SIZE (1..200 ) ) ,
                extensionContainer SEQUENCE {
                    privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE (1..10 ) ) OF
                        SEQUENCE {
                            extId    MAP-EXTENSION .&extensionId ( {
                                ... } ) ,
                            extType    MAP-EXTENSION .&ExtensionType ( {
                                    ... } { @extId } ) OPTIONAL} OPTIONAL,
                                    pcs-Extensions [1] IMPLICIT SEQUENCE {
                                        ... } OPTIONAL,
                                        ... } OPTIONAL,
                                        ... } ,
                    highLayerCompatibility [6] IMPLICIT SEQUENCE {
                    protocolId    ENUMERATED {
                        gsm-0408    (1 ) ,
                        gsm-0806    (2 ) ,
                        gsm-BSSMAP  (3 ) ,
                        ets-300102-1 (4 ) } ,
                    signalInfo    OCTET STRING ( SIZE (1..200 ) ) ,
                    extensionContainer SEQUENCE {
                        privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE (1..10 ) ) OF
                            SEQUENCE {
                                extId    MAP-EXTENSION .&extensionId ( {

```

```

        ... } ) ,
        extType      MAP-EXTENSION .&ExtensionType ( {
            '... } { @extId } ) OPTIONAL} OPTIONAL,
        pcs-Extensions [1] IMPLICIT SEQUENCE {
            ... } OPTIONAL,
            ... } OPTIONAL,
            ... } OPTIONAL,
        extensionContainer [7] IMPLICIT SEQUENCE {
        privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE (1..10) ) OF
        SEQUENCE {
            extId      MAP-EXTENSION .&extensionId ( {
                '... } ) ,
            extType      MAP-EXTENSION .&ExtensionType ( {
                '... } { @extId } ) OPTIONAL} OPTIONAL,
            pcs-Extensions [1] IMPLICIT SEQUENCE {
                ... } OPTIONAL,
                ... } OPTIONAL,
                ... } OPTIONAL,
            ... }
        RESULT
        provideSIWFSSNumberRes SEQUENCE {
            siWFSSNumber [0] IMPLICIT OCTET STRING ( SIZE (1..20) ) ( SIZE (1..9) ),
            extensionContainer [1] IMPLICIT SEQUENCE {
                privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE (1..10) ) OF
                SEQUENCE {
                    extId      MAP-EXTENSION .&extensionId ( {
                        '... } ) ,
                    extType      MAP-EXTENSION .&ExtensionType ( {
                        '... } { @extId } ) OPTIONAL} OPTIONAL,
                    pcs-Extensions [1] IMPLICIT SEQUENCE {
                        ... } OPTIONAL,
                        ... } OPTIONAL,
                        ... } OPTIONAL,
                    ... }
            ERRORS {
                -- resourceLimitation -- localValue : 51,
                -- dataMissing -- localValue : 35,
                -- unexpectedDataValue -- localValue : 36,
                -- systemFailure -- localValue : 34
            ::= localValue : 31

        siWFSSignallingModify OPERATION
        ARGUMENT
        siWFSSignallingModifyArg SEQUENCE {
            channelType [0] IMPLICIT SEQUENCE {
                protocolId      ENUMERATED {
                    gsm-0408      (1 ),
                    gsm-0806      (2 ),
                    gsm-BSSMAP     (3 ),
                    ets-300102-1  (4 )},
                signalInfo      OCTET STRING ( SIZE (1..200) ),
                extensionContainer SEQUENCE {
                    privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE (1..10) ) OF
                    SEQUENCE {
                        extId      MAP-EXTENSION .&extensionId ( {
                            '... } ) ,
                        extType      MAP-EXTENSION .&ExtensionType ( {
                            '... } { @extId } ) OPTIONAL} OPTIONAL,
                        pcs-Extensions [1] IMPLICIT SEQUENCE {
                            ... } OPTIONAL,
                            ... } OPTIONAL,
                            ... } OPTIONAL,
                        ... } OPTIONAL,
                    chosenChannel [1] IMPLICIT SEQUENCE {
                        protocolId      ENUMERATED {
                            gsm-0408      (1 ),
                            gsm-0806      (2 ),
                            gsm-BSSMAP     (3 ),
                            ets-300102-1  (4 )},
                        signalInfo      OCTET STRING ( SIZE (1..200) ),
                        extensionContainer SEQUENCE {
                            privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE (1..10) ) OF
                            SEQUENCE {
                                extId      MAP-EXTENSION .&extensionId ( {
                                    '... } ) ,
                                extType      MAP-EXTENSION .&ExtensionType ( {
                                    '... } { @extId } ) OPTIONAL} OPTIONAL,

```

```

        pcs-Extensions          [1] IMPLICIT SEQUENCE {
            ... } OPTIONAL,
            ... } OPTIONAL,
            ... } OPTIONAL,
    extensionContainer          [2] IMPLICIT SEQUENCE {
        privateExtensionList    [0] IMPLICIT SEQUENCE ( SIZE (1..10) ) OF
            SEQUENCE {
                extId            MAP-EXTENSION .&extensionId ( {
                    '...') },
                extType          MAP-EXTENSION .&ExtensionType ( {
                    '...'} { @extId } ) OPTIONAL} OPTIONAL,
        pcs-Extensions          [1] IMPLICIT SEQUENCE {
            ... } OPTIONAL,
            ... } OPTIONAL,
            ... }
    ... }
RESULT
    siWFSSignallingModifyRes SEQUENCE {
        chosenChannel           [0] IMPLICIT SEQUENCE {
            protocolId          ENUMERATED {
                gsm-0408        (1 ),
                gsm-0806        (2 ),
                gsm-BSSMAP      (3 ),
                ets-300102-1    (4 )},
            signalInfo          OCTET STRING ( SIZE (1..200) ),
            extensionContainer SEQUENCE {
                privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE (1..10) ) OF
                    SEQUENCE {
                        extId            MAP-EXTENSION .&extensionId ( {
                            '...') },
                        extType          MAP-EXTENSION .&ExtensionType ( {
                            '...'} { @extId } ) OPTIONAL} OPTIONAL,
                pcs-Extensions          [1] IMPLICIT SEQUENCE {
                    ... } OPTIONAL,
                    ... } OPTIONAL,
                    ... } OPTIONAL,
            extensionContainer [1] IMPLICIT SEQUENCE {
                privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE (1..10) ) OF
                    SEQUENCE {
                        extId            MAP-EXTENSION .&extensionId ( {
                            '...') },
                        extType          MAP-EXTENSION .&ExtensionType ( {
                            '...'} { @extId } ) OPTIONAL} OPTIONAL,
                pcs-Extensions          [1] IMPLICIT SEQUENCE {
                    ... } OPTIONAL,
                    ... } OPTIONAL,
                    ... }
            ... }
    ERRORS {
        -- resourceLimitation -- localValue : 51,
        -- dataMissing -- localValue : 35,
        -- unexpectedDataValue -- localValue : 36,
        -- systemFailure -- localValue : 34}
    ::= localValue : 32

setReportingState OPERATION
    ARGUMENT
        setReportingStateArg SEQUENCE {
            imsi                [0] IMPLICIT OCTET STRING ( SIZE (3..8) ) OPTIONAL,
            lmsi                [1] IMPLICIT OCTET STRING ( SIZE (4) ) OPTIONAL,
            ccbs-Monitoring     [2] IMPLICIT ENUMERATED {
                stopMonitoring   (0 ),
                startMonitoring  (1 ),
                ... } OPTIONAL,
            extensionContainer [3] IMPLICIT SEQUENCE {
                privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE (1..10) ) OF
                    SEQUENCE {
                        extId            MAP-EXTENSION .&extensionId ( {
                            '...') },
                        extType          MAP-EXTENSION .&ExtensionType ( {
                            '...'} { @extId } ) OPTIONAL} OPTIONAL,
                pcs-Extensions          [1] IMPLICIT SEQUENCE {
                    ... } OPTIONAL,
                    ... } OPTIONAL,
                    ... }
            ... }
    RESULT
        setReportingStateRes SEQUENCE {

```



```

ccbs-SubscriberStatus [0] IMPLICIT ENUMERATED {
  ccbsNotIdle (0 ),
  ccbsIdle (1 ),
  ccbsNotReachable (2 ),
  ... } OPTIONAL,
extensionContainer [1] IMPLICIT SEQUENCE {
  privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE (1..10 ) ) OF
  SEQUENCE {
    extId MAP-EXTENSION .&extensionId ( {
      '... } ) ,
    extType MAP-EXTENSION .&ExtensionType ( {
      '... } { @extId } ) OPTIONAL} OPTIONAL,
  pcs-Extensions [1] IMPLICIT SEQUENCE {
    ... } OPTIONAL,
    ... } OPTIONAL,
  ... }
ERRORS {
  -- systemFailure -- localValue : 34,
  -- unidentifiedSubscriber -- localValue : 5,
  -- unexpectedDataValue -- localValue : 36,
  -- dataMissing -- localValue : 35,
  -- resourceLimitation -- localValue : 51,
  -- facilityNotSupported -- localValue : 21}
 ::= localValue : 73

statusReport OPERATION
ARGUMENT
statusReportArg SEQUENCE {
  imsi [0] IMPLICIT OCTET STRING ( SIZE (3..8 ) ),
  eventReportData [1] IMPLICIT SEQUENCE {
    ccbs-SubscriberStatus [0] IMPLICIT ENUMERATED {
      ccbsNotIdle (0 ),
      ccbsIdle (1 ),
      ccbsNotReachable (2 ),
      ... } OPTIONAL,
    extensionContainer [1] IMPLICIT SEQUENCE {
      privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE (1..10 ) ) OF
      SEQUENCE {
        extId MAP-EXTENSION .&extensionId ( {
          '... } ) ,
        extType MAP-EXTENSION .&ExtensionType ( {
          '... } { @extId } ) OPTIONAL} OPTIONAL,
      pcs-Extensions [1] IMPLICIT SEQUENCE {
        ... } OPTIONAL,
        ... } OPTIONAL,
        ... } OPTIONAL,
      ... } OPTIONAL,
    callReportdata [2] IMPLICIT SEQUENCE {
      monitoringMode [0] IMPLICIT ENUMERATED {
        a-side (0 ),
        b-side (1 ),
        ... } OPTIONAL,
      callOutcome [1] IMPLICIT ENUMERATED {
        success (0 ),
        failure (1 ),
        busy (2 ),
        ... } OPTIONAL,
      extensionContainer [2] IMPLICIT SEQUENCE {
        privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE (1..10 ) ) OF
        SEQUENCE {
          extId MAP-EXTENSION .&extensionId ( {
            '... } ) ,
          extType MAP-EXTENSION .&ExtensionType ( {
            '... } { @extId } ) OPTIONAL} OPTIONAL,
          pcs-Extensions [1] IMPLICIT SEQUENCE {
            ... } OPTIONAL,
            ... } OPTIONAL,
            ... } OPTIONAL,
          extensionContainer [3] IMPLICIT SEQUENCE {
            privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE (1..10 ) ) OF
            SEQUENCE {
              extId MAP-EXTENSION .&extensionId ( {
                '... } ) ,
              extType MAP-EXTENSION .&ExtensionType ( {
                '... } { @extId } ) OPTIONAL} OPTIONAL,
              pcs-Extensions [1] IMPLICIT SEQUENCE {

```

```

        ... } OPTIONAL,
        ... } OPTIONAL,
    ... }
RESULT
statusReportRes SEQUENCE {
    extensionContainer [0] IMPLICIT SEQUENCE {
        privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE (1..10) ) OF
            SEQUENCE {
                extId MAP-EXTENSION .&extensionId ( {
                    '...'} ) ,
                extType MAP-EXTENSION .&ExtensionType ( {
                    '...'} { @extId } ) OPTIONAL} OPTIONAL,
        pcs-Extensions [1] IMPLICIT SEQUENCE {
            ... } OPTIONAL,
            ... } OPTIONAL,
        ... }
ERRORS {
    -- unknownSubscriber -- localValue : 1,
    -- systemFailure -- localValue : 34,
    -- unexpectedDataValue -- localValue : 36,
    -- dataMissing -- localValue : 35}
 ::= localValue : 74

remoteUserFree OPERATION
ARGUMENT
remoteUserFreeArg SEQUENCE {
    imsi [0] IMPLICIT OCTET STRING ( SIZE (3..8) ),
    callInfo [1] IMPLICIT SEQUENCE {
        protocolId ENUMERATED {
            gsm-0408 (1 ),
            gsm-0806 (2 ),
            gsm-BSSMAP (3 ),
            ets-300102-1 (4 )},
        signalInfo OCTET STRING ( SIZE (1..200) ),
        extensionContainer SEQUENCE {
            privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE (1..10) ) OF
                SEQUENCE {
                    extId MAP-EXTENSION .&extensionId ( {
                        '...'} ) ,
                    extType MAP-EXTENSION .&ExtensionType ( {
                        '...'} { @extId } ) OPTIONAL} OPTIONAL,
            pcs-Extensions [1] IMPLICIT SEQUENCE {
                ... } OPTIONAL,
                ... } OPTIONAL,
            ... },
        ccbs-Feature [2] IMPLICIT SEQUENCE {
            ccbs-Index [0] IMPLICIT INTEGER ( 1..5 ) OPTIONAL,
            b-subscriberNumber [1] IMPLICIT OCTET STRING ( SIZE (1..20) ) ( SIZE (1..9) )
OPTIONAL,
            b-subscriberSubaddress [2] IMPLICIT OCTET STRING ( SIZE (1..21) ) OPTIONAL,
            basicServiceGroup [3] CHOICE {
                bearerService [2] IMPLICIT OCTET STRING ( SIZE (1) ),
                teleservice [3] IMPLICIT OCTET STRING ( SIZE (1) )} OPTIONAL,
            ... },
            translatedB-Number [3] IMPLICIT OCTET STRING ( SIZE (1..20) ) ( SIZE (1..9) ),
            replaceB-Number [4] IMPLICIT NULL OPTIONAL,
            alertingPattern [5] IMPLICIT OCTET STRING ( SIZE (1) ) OPTIONAL,
            extensionContainer [6] IMPLICIT SEQUENCE {
                privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE (1..10) ) OF
                    SEQUENCE {
                        extId MAP-EXTENSION .&extensionId ( {
                            '...'} ) ,
                        extType MAP-EXTENSION .&ExtensionType ( {
                            '...'} { @extId } ) OPTIONAL} OPTIONAL,
                pcs-Extensions [1] IMPLICIT SEQUENCE {
                    ... } OPTIONAL,
                    ... } OPTIONAL,
                ... }
    ... }
RESULT
remoteUserFreeRes SEQUENCE {
    ruf-Outcome [0] IMPLICIT ENUMERATED {
        accepted (0 ),
        rejected (1 ),
        noResponseFromFreeMS (2 ),
        noResponseFromBusyMS (3 ),
        udubFromFreeMS (4 ),
        udubFromBusyMS (5 ),

```

```

... },
extensionContainer [1] IMPLICIT SEQUENCE {
  privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE (1..10) ) OF
    SEQUENCE {
      extId MAP-EXTENSION .&extensionId ( {
        '... } ) ,
      extType MAP-EXTENSION .&ExtensionType ( {
        '... } { @extId } ) OPTIONAL} OPTIONAL,
  pcs-Extensions [1] IMPLICIT SEQUENCE {
    ... } OPTIONAL,
  ... } OPTIONAL,
... }
ERRORS {
  -- unexpectedDataValue -- localValue : 36,
  -- dataMissing -- localValue : 35,
  -- incompatibleTerminal -- localValue : 28,
  -- absentSubscriber -- localValue : 27,
  -- systemFailure -- localValue : 34,
  -- busySubscriber -- localValue : 45}
 ::= localValue : 75

istAlert OPERATION
ARGUMENT
  istAlertArg SEQUENCE {
    imsi [0] IMPLICIT OCTET STRING ( SIZE (3..8) ),
    extensionContainer [1] IMPLICIT SEQUENCE {
      privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE (1..10) ) OF
        SEQUENCE {
          extId MAP-EXTENSION .&extensionId ( {
            '... } ) ,
          extType MAP-EXTENSION .&ExtensionType ( {
            '... } { @extId } ) OPTIONAL} OPTIONAL,
        pcs-Extensions [1] IMPLICIT SEQUENCE {
          ... } OPTIONAL,
        ... } OPTIONAL,
      ... }
    RESULT
      istAlertRes SEQUENCE {
        istAlertTimer [0] IMPLICIT INTEGER ( 15..255 ) OPTIONAL,
        istInformationWithdraw [1] IMPLICIT NULL OPTIONAL,
        callTerminationIndicator [2] IMPLICIT ENUMERATED {
          terminateCallActivityReferred ( 0 ),
          terminateAllCallActivities ( 1 ),
          ... } OPTIONAL,
        extensionContainer [3] IMPLICIT SEQUENCE {
          privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE (1..10) ) OF
            SEQUENCE {
              extId MAP-EXTENSION .&extensionId ( {
                '... } ) ,
              extType MAP-EXTENSION .&ExtensionType ( {
                '... } { @extId } ) OPTIONAL} OPTIONAL,
            pcs-Extensions [1] IMPLICIT SEQUENCE {
              ... } OPTIONAL,
            ... } OPTIONAL,
          ... }
        ERRORS {
          -- unexpectedDataValue -- localValue : 36,
          -- resourceLimitation -- localValue : 51,
          -- unknownSubscriber -- localValue : 1,
          -- systemFailure -- localValue : 34,
          -- facilityNotSupported -- localValue : 21}
        ::= localValue : 87

istCommand OPERATION
ARGUMENT
  istCommandArg SEQUENCE {
    imsi [0] IMPLICIT OCTET STRING ( SIZE (3..8) ),
    extensionContainer [1] IMPLICIT SEQUENCE {
      privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE (1..10) ) OF
        SEQUENCE {
          extId MAP-EXTENSION .&extensionId ( {
            '... } ) ,
          extType MAP-EXTENSION .&ExtensionType ( {
            '... } { @extId } ) OPTIONAL} OPTIONAL,
        pcs-Extensions [1] IMPLICIT SEQUENCE {

```

```

        ... } OPTIONAL,
        ... } OPTIONAL,
    ... }
RESULT
istCommandRes SEQUENCE {
    extensionContainer SEQUENCE {
        privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE (1..10) ) OF
            SEQUENCE {
                extId MAP-EXTENSION .&extensionId ( {
                    '...'} ),
                extType MAP-EXTENSION .&ExtensionType ( {
                    '...'} { @extId } ) OPTIONAL} OPTIONAL,
        pcs-Extensions [1] IMPLICIT SEQUENCE {
            ... } OPTIONAL,
            ... } OPTIONAL,
        ... }
ERRORS {
    -- unexpectedDataValue -- localValue : 36,
    -- resourceLimitation -- localValue : 51,
    -- unknownSubscriber -- localValue : 1,
    -- systemFailure -- localValue : 34,
    -- facilityNotSupported -- localValue : 21}
 ::= localValue : 88

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,
        ... ,
        defaultPriority [7] IMPLICIT INTEGER ( 0..15 ) 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..20) ) ( 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,
                    ... },
                ... },
        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,
                ... ,
            defaultPriority INTEGER ( 0..15 ) OPTIONAL}}
ERRORS {

```

```

-- systemFailure -- localValue : 34,
-- dataMissing -- localValue : 35,
-- unexpectedDataValue -- localValue : 36,
-- bearerServiceNotProvisioned -- localValue : 10,
-- teleserviceNotProvisioned -- localValue : 11,
-- callBarred -- localValue : 13,
-- illegalSS-Operation -- localValue : 16,
-- ss-ErrorStatus -- localValue : 17,
-- 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..20 ) ) ( 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,
            ... },
          ... },
    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,
          ... ,
      defaultPriority INTEGER ( 0..15 ) OPTIONAL}}
  ERRORS {
    -- systemFailure -- localValue : 34,
    -- dataMissing -- localValue : 35,
    -- unexpectedDataValue -- localValue : 36,
    -- bearerServiceNotProvisioned -- localValue : 10,
    -- teleserviceNotProvisioned -- localValue : 11,
    -- callBarred -- localValue : 13,
    -- illegalSS-Operation -- localValue : 16,
    -- ss-ErrorStatus -- localValue : 17}
 ::= 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..20 ) ) ( 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,
        ... },
    ... },
  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,
    ... ,
    defaultPriority INTEGER ( 0..15 ) OPTIONAL}}
ERRORS {
-- systemFailure -- localValue : 34,
-- dataMissing -- localValue : 35,
-- unexpectedDataValue -- localValue : 36,
-- 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..20 ) ) ( 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,
        ... },
      ... },
  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,
        ... ,
        defaultPriority INTEGER ( 0..15 ) OPTIONAL}}
ERRORS {
  -- systemFailure -- localValue : 34,
  -- dataMissing -- localValue : 35,
  -- unexpectedDataValue -- localValue : 36,
  -- 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 ) ),
      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..20 ) ) ( 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,
            ... },
      genericServiceInfo [4] IMPLICIT SEQUENCE {
        ss-Status OCTET STRING ( SIZE ( 1 ) ),
        cliRestrictionOption ENUMERATED {
          permanent ( 0 ),
          temporaryDefaultRestricted ( 1 ),
          temporaryDefaultAllowed ( 2 )} OPTIONAL,
        ... ,
        maximumEntitledPriority [0] IMPLICIT INTEGER ( 0..15 ) OPTIONAL,
        defaultPriority [1] IMPLICIT INTEGER ( 0..15 ) OPTIONAL,
        ccbs-FeatureList [2] IMPLICIT SEQUENCE ( SIZE ( 1..5 ) ) OF
          SEQUENCE {
            ccbs-Index [0] IMPLICIT INTEGER ( 1..5 ) OPTIONAL,
            b-subscriberNumber [1] IMPLICIT OCTET STRING ( SIZE ( 1..20 ) ) ( SIZE ( 1..9 ) )
            OPTIONAL,
            b-subscriberSubaddress [2] IMPLICIT OCTET STRING ( SIZE ( 1..21 ) ) OPTIONAL,

```

```

        basicServiceGroup      [3] CHOICE {
            bearerService      [2] IMPLICIT OCTET STRING ( SIZE ( 1 ) ),
            teleservice        [3] IMPLICIT OCTET STRING ( SIZE ( 1 ) ) } OPTIONAL,
        ... } OPTIONAL}}
ERRORS {
    -- systemFailure -- localValue : 34,
    -- dataMissing -- localValue : 35,
    -- unexpectedDataValue -- localValue : 36,
    -- bearerServiceNotProvisioned -- localValue : 10,
    -- teleserviceNotProvisioned -- localValue : 11,
    -- callBarred -- localValue : 13,
    -- illegalSS-Operation -- localValue : 16,
    -- ss-NotAvailable -- localValue : 18}
 ::= localValue : 14

processUnstructuredSS-Request OPERATION
ARGUMENT
    ussd-Arg SEQUENCE {
        ussd-DataCodingScheme OCTET STRING ( SIZE ( 1 ) ),
        ussd-String           OCTET STRING ( SIZE ( 1..160 ) ),
        ... ,
        alertingPattern       OCTET STRING ( SIZE ( 1 ) ) OPTIONAL,
        msisdn                [0] IMPLICIT OCTET STRING ( SIZE ( 1..20 ) ) ( SIZE ( 1..9 ) )
OPTIONAL}
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 ) ),
        ... ,
        alertingPattern       OCTET STRING ( SIZE ( 1 ) ) OPTIONAL,
        msisdn                [0] IMPLICIT OCTET STRING ( SIZE ( 1..20 ) ) ( SIZE ( 1..9 ) )
OPTIONAL}
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 ) ),
        ... ,
        alertingPattern       OCTET STRING ( SIZE ( 1 ) ) OPTIONAL,
        msisdn                [0] IMPLICIT OCTET STRING ( SIZE ( 1..20 ) ) ( SIZE ( 1..9 ) )
OPTIONAL}
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 )}
  RESULT
    currentPassword NumericString ( FROM ("0"|"1"|"2"|"3"|"4"|"5"|"6"|"7"|"8"|"9" )|SIZE ( 4 ) )
 ::= localValue : 18

registerCC-Entry OPERATION
  ARGUMENT
    registerCC-EntryArg SEQUENCE {
      ss-Code    [0] IMPLICIT OCTET STRING ( SIZE ( 1 ) ),
      ccbs-Data  [1] IMPLICIT SEQUENCE {
        ccbs-Feature [0] IMPLICIT SEQUENCE {
          ccbs-Index [0] IMPLICIT INTEGER ( 1..5 ) OPTIONAL,
          b-subscriberNumber [1] IMPLICIT OCTET STRING ( SIZE ( 1..20 ) ) ( SIZE ( 1..9 ) )
        OPTIONAL,
          b-subscriberSubaddress [2] IMPLICIT OCTET STRING ( SIZE ( 1..21 ) ) OPTIONAL,
          basicServiceGroup [3] CHOICE {
            bearerService [2] IMPLICIT OCTET STRING ( SIZE ( 1 ) ),
            teleservice [3] IMPLICIT OCTET STRING ( SIZE ( 1 ) )} OPTIONAL,
            ... },
          translatedB-Number [1] IMPLICIT OCTET STRING ( SIZE ( 1..20 ) ) ( SIZE ( 1..9 ) ),
          serviceIndicator [2] IMPLICIT BIT STRING {
            clir-invoked ( 0 ),
            camel-invoked ( 1 )} ( SIZE ( 2..32 ) ) OPTIONAL,
          callInfo [3] IMPLICIT SEQUENCE {
            protocolId ENUMERATED {
              gsm-0408 ( 1 ),
              gsm-0806 ( 2 ),
              gsm-BSSMAP ( 3 ),
              ets-300102-1 ( 4 )},
            signalInfo OCTET STRING ( SIZE ( 1..200 ) ),
            extensionContainer SEQUENCE {
              privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE ( 1..10 ) ) OF
                SEQUENCE {
                  extId MAP-EXTENSION .&extensionId ( {
                    ... } ) ,
                  extType MAP-EXTENSION .&ExtensionType ( {
                    ... } { @extId } ) OPTIONAL} OPTIONAL,
              pcs-Extensions [1] IMPLICIT SEQUENCE {
                ... } OPTIONAL,
                ... } OPTIONAL,
                ... },
            networkSignalInfo [4] IMPLICIT SEQUENCE {
              protocolId ENUMERATED {
                gsm-0408 ( 1 ),
                gsm-0806 ( 2 ),
                gsm-BSSMAP ( 3 ),
                ets-300102-1 ( 4 )},
              signalInfo OCTET STRING ( SIZE ( 1..200 ) ),
              extensionContainer SEQUENCE {
                privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE ( 1..10 ) ) OF
                  SEQUENCE {
                    extId MAP-EXTENSION .&extensionId ( {
                      ... } ) ,
                    extType MAP-EXTENSION .&ExtensionType ( {
                      ... } { @extId } ) OPTIONAL} OPTIONAL,
                  pcs-Extensions [1] IMPLICIT SEQUENCE {
                    ... } OPTIONAL,
                    ... } OPTIONAL,
                    ... }
            }

```

```

        ... } OPTIONAL,
        ... } OPTIONAL,
        ... }
RESULT
  registerCC-EntryRes SEQUENCE {
    ccbs-Feature [0] IMPLICIT SEQUENCE {
      ccbs-Index [0] IMPLICIT INTEGER ( 1..5 ) OPTIONAL,
      b-subscriberNumber [1] IMPLICIT OCTET STRING ( SIZE (1..20 ) ) ( SIZE (1..9 ) )
OPTIONAL,
      b-subscriberSubaddress [2] IMPLICIT OCTET STRING ( SIZE (1..21 ) ) OPTIONAL,
      basicServiceGroup [3] CHOICE {
        bearerService [2] IMPLICIT OCTET STRING ( SIZE (1 ) ),
        teleservice [3] IMPLICIT OCTET STRING ( SIZE (1 ) ) } OPTIONAL,
        ... } OPTIONAL,
        ... }
ERRORS {
  -- systemFailure -- localValue : 34,
  -- dataMissing -- localValue : 35,
  -- unexpectedDataValue -- localValue : 36,
  -- callBarred -- localValue : 13,
  -- illegalSS-Operation -- localValue : 16,
  -- ss-ErrorStatus -- localValue : 17,
  -- ss-Incompatibility -- localValue : 20,
  -- shortTermDenial -- localValue : 29,
  -- longTermDenial -- localValue : 30,
  -- facilityNotSupported -- localValue : 21}
 ::= localValue : 76

eraseCC-Entry OPERATION
ARGUMENT
  eraseCC-EntryArg SEQUENCE {
    ss-Code [0] IMPLICIT OCTET STRING ( SIZE (1 ) ),
    ccbs-Index [1] IMPLICIT INTEGER ( 1..5 ) OPTIONAL,
    ... }
RESULT
  eraseCC-EntryRes SEQUENCE {
    ss-Code [0] IMPLICIT OCTET STRING ( SIZE (1 ) ),
    ss-Status [1] IMPLICIT OCTET STRING ( SIZE (1 ) ) OPTIONAL,
    ... }
ERRORS {
  -- systemFailure -- localValue : 34,
  -- dataMissing -- localValue : 35,
  -- unexpectedDataValue -- localValue : 36,
  -- callBarred -- localValue : 13,
  -- illegalSS-Operation -- localValue : 16,
  -- ss-ErrorStatus -- localValue : 17}
 ::= localValue : 77

sendRoutingInfoForSM OPERATION
ARGUMENT
  routingInfoForSM-Arg SEQUENCE {
    msisdn [0] IMPLICIT OCTET STRING ( SIZE (1..20 ) ) ( SIZE (1..9 ) ),
    sm-RP-PRI [1] IMPLICIT BOOLEAN,
    serviceCentreAddress [2] IMPLICIT OCTET STRING ( SIZE (1..20 ) ),
    extensionContainer [6] IMPLICIT SEQUENCE {
      privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE (1..10 ) ) OF
        SEQUENCE {
          extId MAP-EXTENSION .&extensionId ( {
            '
            ... } ) ,
          extType MAP-EXTENSION .&ExtensionType ( {
            '
            ... } { @extId } ) OPTIONAL} OPTIONAL,
          pcs-Extensions [1] IMPLICIT SEQUENCE {
            ... } OPTIONAL,
            ... } OPTIONAL,
            ... ,
            gprsSupportIndicator [7] IMPLICIT NULL OPTIONAL,
            sm-RP-MTI [8] IMPLICIT INTEGER ( 0..10 ) OPTIONAL,
            sm-RP-SMEA [9] IMPLICIT OCTET STRING ( SIZE (1..12 ) ) OPTIONAL}
RESULT
  routingInfoForSM-Res SEQUENCE {
    imsi OCTET STRING ( SIZE (3..8 ) ),
    locationInfoWithLMSI [0] IMPLICIT SEQUENCE {
      networkNode-Number [1] IMPLICIT OCTET STRING ( SIZE (1..20 ) ) ( SIZE (1..9 ) ),
      lmsi OCTET STRING ( SIZE (4 ) ) OPTIONAL,
      extensionContainer SEQUENCE {
        privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE (1..10 ) ) OF
          SEQUENCE {
            extId MAP-EXTENSION .&extensionId ( {
              '
              ... } ) ,

```

```

        extType      MAP-EXTENSION .&ExtensionType ( {
            '...' { @extId } ) OPTIONAL} OPTIONAL,
        pcs-Extensions [1] IMPLICIT SEQUENCE {
            ... } OPTIONAL,
            ... } OPTIONAL,
            ... ,
        gprsNodeIndicator [5] IMPLICIT NULL OPTIONAL,
        additional-Number [6] CHOICE {
            msc-Number [0] IMPLICIT OCTET STRING ( SIZE (1..20 ) ) ( SIZE (1..9 ) ),
            sgsn-Number [1] IMPLICIT OCTET STRING ( SIZE (1..20 ) ) ( SIZE (1..9 ) )}
OPTIONAL},
        extensionContainer [4] IMPLICIT SEQUENCE {
            privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE (1..10 ) ) OF
                SEQUENCE {
                    extId      MAP-EXTENSION .&extensionId ( {
                        '...' ) ,
                    extType    MAP-EXTENSION .&ExtensionType ( {
                        '...' { @extId } ) OPTIONAL} OPTIONAL,
                    pcs-Extensions [1] IMPLICIT SEQUENCE {
                        ... } OPTIONAL,
                        ... } OPTIONAL,
                        ... }
                }
        ERRORS {
            -- systemFailure -- localValue : 34,
            -- dataMissing -- localValue : 35,
            -- unexpectedDataValue -- localValue : 36,
            -- facilityNotSupported -- localValue : 21,
            -- unknownSubscriber -- localValue : 1,
            -- teleserviceNotProvisioned -- localValue : 11,
            -- callBarred -- localValue : 13,
            -- absentSubscriberSM -- localValue : 6}
        ::= localValue : 45

mo-forwardSM OPERATION
    ARGUMENT
        mo-forwardSM-Arg SEQUENCE {
            sm-RP-DA CHOICE {
                imsi [0] IMPLICIT OCTET STRING ( SIZE (3..8 ) ),
                lmsi [1] IMPLICIT OCTET STRING ( SIZE (4 ) ),
                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..20 ) ) ( 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 ) ),
            extensionContainer SEQUENCE {
                privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE (1..10 ) ) OF
                    SEQUENCE {
                        extId      MAP-EXTENSION .&extensionId ( {
                            '...' ) ,
                        extType    MAP-EXTENSION .&ExtensionType ( {
                            '...' { @extId } ) OPTIONAL} OPTIONAL,
                        pcs-Extensions [1] IMPLICIT SEQUENCE {
                            ... } OPTIONAL,
                            ... } OPTIONAL,
                            ... }
                    }
            }
            imsi OCTET STRING ( SIZE (3..8 ) ) OPTIONAL}
        RESULT
            mo-forwardSM-Res SEQUENCE {
                sm-RP-UI OCTET STRING ( SIZE (1..200 ) ) OPTIONAL,
                extensionContainer SEQUENCE {
                    privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE (1..10 ) ) OF
                        SEQUENCE {
                            extId      MAP-EXTENSION .&extensionId ( {
                                '...' ) ,
                            extType    MAP-EXTENSION .&ExtensionType ( {
                                '...' { @extId } ) OPTIONAL} OPTIONAL,
                                pcs-Extensions [1] IMPLICIT SEQUENCE {
                                    ... } OPTIONAL,
                                    ... } OPTIONAL,
                                    ... }
                            }
                    }
            }
            ERRORS {
                -- systemFailure -- localValue : 34,
                -- unexpectedDataValue -- localValue : 36,
                -- facilityNotSupported -- localValue : 21,

```

```

-- sm-DeliveryFailure -- localValue : 32}
 ::= localValue : 46

mt-forwardSM OPERATION
ARGUMENT
  mt-forwardSM-Arg SEQUENCE {
    sm-RP-DA CHOICE {
      imsi [0] IMPLICIT OCTET STRING ( SIZE ( 3..8 ) ),
      lmsi [1] IMPLICIT OCTET STRING ( SIZE ( 4 ) ),
      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..20 ) ) ( 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,
    extensionContainer SEQUENCE {
      privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE ( 1..10 ) ) OF
        SEQUENCE {
          extId MAP-EXTENSION .&extensionId ( {
            '...'} ) ,
          extType MAP-EXTENSION .&ExtensionType ( {
            '...'} { @extId } ) OPTIONAL} OPTIONAL,
      pcs-Extensions [1] IMPLICIT SEQUENCE {
        ... } OPTIONAL,
        ... } OPTIONAL,
        ... }
    ... }
RESULT
  mt-forwardSM-Res SEQUENCE {
    sm-RP-UI OCTET STRING ( SIZE ( 1..200 ) ) OPTIONAL,
    extensionContainer SEQUENCE {
      privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE ( 1..10 ) ) OF
        SEQUENCE {
          extId MAP-EXTENSION .&extensionId ( {
            '...'} ) ,
          extType MAP-EXTENSION .&ExtensionType ( {
            '...'} { @extId } ) OPTIONAL} OPTIONAL,
      pcs-Extensions [1] IMPLICIT SEQUENCE {
        ... } OPTIONAL,
        ... } OPTIONAL,
        ... }
    ... }
ERRORS {
-- systemFailure -- localValue : 34,
-- dataMissing -- localValue : 35,
-- unexpectedDataValue -- localValue : 36,
-- facilityNotSupported -- localValue : 21,
-- unidentifiedSubscriber -- localValue : 5,
-- illegalSubscriber -- localValue : 9,
-- illegalEquipment -- localValue : 12,
-- subscriberBusyForMT-SMS -- localValue : 31,
-- sm-DeliveryFailure -- localValue : 32,
-- absentsubscriberSM -- localValue : 6}
 ::= localValue : 44

reportSM-DeliveryStatus OPERATION
ARGUMENT
  reportSM-DeliveryStatusArg SEQUENCE {
    msisdn OCTET STRING ( SIZE ( 1..20 ) ) ( SIZE ( 1..9 ) ),
    serviceCentreAddress OCTET STRING ( SIZE ( 1..20 ) ),
    sm-DeliveryOutcome ENUMERATED {
      memoryCapacityExceeded ( 0 ),
      absentSubscriber ( 1 ),
      successfulTransfer ( 2 )},
    absentSubscriberDiagnosticSM [0] IMPLICIT INTEGER ( 0..255 ) OPTIONAL,
    extensionContainer [1] IMPLICIT SEQUENCE {
      privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE ( 1..10 ) ) OF
        SEQUENCE {
          extId MAP-EXTENSION .&extensionId ( {
            '...'} ) ,
          extType MAP-EXTENSION .&ExtensionType ( {
            '...'} { @extId } ) OPTIONAL} OPTIONAL,
      pcs-Extensions [1] IMPLICIT SEQUENCE {
        ... } OPTIONAL,
        ... } OPTIONAL,
        ... ,
    gprsSupportIndicator [2] IMPLICIT NULL OPTIONAL,

```

```

deliveryOutcomeIndicator [3] IMPLICIT NULL OPTIONAL,
additionalSM-DeliveryOutcome [4] IMPLICIT ENUMERATED {
    memoryCapacityExceeded (0 ),
    absentSubscriber (1 ),
    successfulTransfer (2 )} OPTIONAL,
additionalAbsentSubscriberDiagnosticSM [5] IMPLICIT INTEGER ( 0..255 ) OPTIONAL}
RESULT
reportSM-DeliveryStatusRes SEQUENCE {
    storedMSISDN OCTET STRING ( SIZE (1..20 ) ) ( SIZE (1..9 ) ) OPTIONAL,
    extensionContainer SEQUENCE {
        privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE (1..10 ) ) OF
            SEQUENCE {
                extId MAP-EXTENSION .&extensionId ( {
                    '... } ) ,
                extType MAP-EXTENSION .&ExtensionType ( {
                    '... } { @extId } ) OPTIONAL} OPTIONAL,
        pcs-Extensions [1] IMPLICIT SEQUENCE {
            ... } OPTIONAL,
            ... } OPTIONAL,
            ... }
ERRORS {
    -- dataMissing -- localValue : 35,
    -- unexpectedDataValue -- localValue : 36,
    -- unknownSubscriber -- localValue : 1,
    -- messageWaitingListFull -- localValue : 33}
 ::= localValue : 47

informServiceCentre OPERATION
ARGUMENT
informServiceCentreArg SEQUENCE {
    storedMSISDN OCTET STRING ( SIZE (1..20 ) ) ( SIZE (1..9 ) ) OPTIONAL,
    mw-Status BIT STRING {
        sc-AddressNotIncluded (0 ),
        mnrf-Set (1 ),
        mcef-Set (2 ),
        mnrg-Set (3 )} ( SIZE (6..16 ) ) OPTIONAL,
    extensionContainer SEQUENCE {
        privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE (1..10 ) ) OF
            SEQUENCE {
                extId MAP-EXTENSION .&extensionId ( {
                    '... } ) ,
                extType MAP-EXTENSION .&ExtensionType ( {
                    '... } { @extId } ) OPTIONAL} OPTIONAL,
        pcs-Extensions [1] IMPLICIT SEQUENCE {
            ... } OPTIONAL,
            ... } OPTIONAL,
            ... }
 ::= localValue : 63

alertServiceCentre OPERATION
ARGUMENT
alertServiceCentreArg SEQUENCE {
    msisdn OCTET STRING ( SIZE (1..20 ) ) ( 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 )} ,
    alertReasonIndicator NULL OPTIONAL,
    extensionContainer SEQUENCE {
        privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE (1..10 ) ) OF
            SEQUENCE {
                extId MAP-EXTENSION .&extensionId ( {
                    '... } ) ,
                extType MAP-EXTENSION .&ExtensionType ( {
                    '... } { @extId } ) OPTIONAL} OPTIONAL,
        pcs-Extensions [1] IMPLICIT SEQUENCE {

```

```

        ... } OPTIONAL,
        ... } OPTIONAL,
    ... }
RESULT
  readyForSM-Res SEQUENCE {
    extensionContainer SEQUENCE {
      privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE (1..10) ) OF
        SEQUENCE {
          extId MAP-EXTENSION .&extensionId ( {
            '...'} ) ,
          extType MAP-EXTENSION .&ExtensionType ( {
            '...'} { @extId } ) OPTIONAL} OPTIONAL,
      pcs-Extensions [1] IMPLICIT SEQUENCE {
        ... } OPTIONAL,
        ... } OPTIONAL,
    ... }
  ERRORS {
    -- dataMissing -- localValue : 35,
    -- unexpectedDataValue -- localValue : 36,
    -- facilityNotSupported -- localValue : 21,
    -- unknownSubscriber -- localValue : 1}
  ::= localValue : 66

provideSubscriberInfo OPERATION
  ARGUMENT
    provideSubscriberInfoArg SEQUENCE {
      imsi [0] IMPLICIT OCTET STRING ( SIZE (3..8) ),
      lmsi [1] IMPLICIT OCTET STRING ( SIZE (4) ) OPTIONAL,
      requestedInfo [2] IMPLICIT SEQUENCE {
        locationInformation [0] IMPLICIT NULL OPTIONAL,
        subscriberState [1] IMPLICIT NULL OPTIONAL,
        extensionContainer [2] IMPLICIT SEQUENCE {
          privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE (1..10) ) OF
            SEQUENCE {
              extId MAP-EXTENSION .&extensionId ( {
                '...'} ) ,
              extType MAP-EXTENSION .&ExtensionType ( {
                '...'} { @extId } ) OPTIONAL} OPTIONAL,
            pcs-Extensions [1] IMPLICIT SEQUENCE {
              ... } OPTIONAL,
              ... } OPTIONAL,
            ... },
          extensionContainer [3] IMPLICIT SEQUENCE {
            privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE (1..10) ) OF
              SEQUENCE {
                extId MAP-EXTENSION .&extensionId ( {
                  '...'} ) ,
                extType MAP-EXTENSION .&ExtensionType ( {
                  '...'} { @extId } ) OPTIONAL} OPTIONAL,
              pcs-Extensions [1] IMPLICIT SEQUENCE {
                ... } OPTIONAL,
                ... } OPTIONAL,
              ... }
          }
    }
  RESULT
    provideSubscriberInfoRes SEQUENCE {
      subscriberInfo SEQUENCE {
        locationInformation [0] IMPLICIT SEQUENCE {
          ageOfLocationInformation INTEGER ( 0..32767 ) OPTIONAL,
          geographicalInformation [0] IMPLICIT OCTET STRING ( SIZE (8) ) OPTIONAL,
          vlr-number [1] IMPLICIT OCTET STRING ( SIZE (1..20) ) ( SIZE (1..9)
) OPTIONAL,
          locationNumber [2] IMPLICIT OCTET STRING ( SIZE (2..10) ) OPTIONAL,
          cellIdOrLAI [3] CHOICE {
            cellIdFixedLength [0] IMPLICIT OCTET STRING ( SIZE (7) ),
            laiFixedLength [1] IMPLICIT OCTET STRING ( SIZE (5) )} OPTIONAL,
          extensionContainer [4] IMPLICIT SEQUENCE {
            privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE (1..10) ) OF
              SEQUENCE {
                extId MAP-EXTENSION .&extensionId ( {
                  '...'} ) ,
                extType MAP-EXTENSION .&ExtensionType ( {
                  '...'} { @extId } ) OPTIONAL} OPTIONAL,
            pcs-Extensions [1] IMPLICIT SEQUENCE {
              ... } OPTIONAL,
              ... } OPTIONAL,
            ... }
          }
    }
  }

```

```

... ,
selectedLSA-Id          [5] IMPLICIT OCTET STRING ( SIZE ( 3 ) ) OPTIONAL,
msc-Number              [6] IMPLICIT OCTET STRING ( SIZE ( 1..20 ) ) ( SIZE ( 1..9 )
) OPTIONAL,
geodeticInformation     [7] IMPLICIT OCTET STRING ( SIZE ( 10 ) ) OPTIONAL}
OPTIONAL,
subscriberState         [1] CHOICE {
  assumedIdle           [0] IMPLICIT NULL,
  camelBusy             [1] IMPLICIT NULL,
  netDetNotReachable    ENUMERATED {
    msPurged            ( 0 ),
    imsiDetached        ( 1 ),
    restrictedArea       ( 2 ),
    notRegistered       ( 3 )},
  notProvidedFromVLR    [2] IMPLICIT NULL} OPTIONAL,
extensionContainer      [2] IMPLICIT SEQUENCE {
  privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE ( 1..10 ) ) OF
  SEQUENCE {
    extId                MAP-EXTENSION .&extensionId ( {
      '...' ) ,
    extType              MAP-EXTENSION .&ExtensionType ( {
      '...' { @extId } ) OPTIONAL} OPTIONAL,
  pcs-Extensions         [1] IMPLICIT SEQUENCE {
    ... } OPTIONAL,
  ... } OPTIONAL,
  ... },
extensionContainer      SEQUENCE {
  privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE ( 1..10 ) ) OF
  SEQUENCE {
    extId                MAP-EXTENSION .&extensionId ( {
      '...' ) ,
    extType              MAP-EXTENSION .&ExtensionType ( {
      '...' { @extId } ) OPTIONAL} OPTIONAL,
  pcs-Extensions         [1] IMPLICIT SEQUENCE {
    ... } OPTIONAL,
  ... } OPTIONAL,
  ... }
ERRORS {
  -- dataMissing -- localValue : 35,
  -- unexpectedDataValue -- localValue : 36}
 ::= localValue : 70

anyTimeInterrogation OPERATION
ARGUMENT
  anyTimeInterrogationArg SEQUENCE {
    subscriberIdentity [0] CHOICE {
      imsi             [0] IMPLICIT OCTET STRING ( SIZE ( 3..8 ) ),
      msisdn           [1] IMPLICIT OCTET STRING ( SIZE ( 1..20 ) ) ( SIZE ( 1..9 ) )},
    requestedInfo      [1] IMPLICIT SEQUENCE {
      locationInformation [0] IMPLICIT NULL OPTIONAL,
      subscriberState     [1] IMPLICIT NULL OPTIONAL,
      extensionContainer  [2] IMPLICIT SEQUENCE {
        privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE ( 1..10 ) ) OF
        SEQUENCE {
          extId                MAP-EXTENSION .&extensionId ( {
            '...' ) ,
          extType              MAP-EXTENSION .&ExtensionType ( {
            '...' { @extId } ) OPTIONAL} OPTIONAL,
        pcs-Extensions         [1] IMPLICIT SEQUENCE {
          ... } OPTIONAL,
        ... } OPTIONAL,
        ... },
      gsmSCF-Address        [3] IMPLICIT OCTET STRING ( SIZE ( 1..20 ) ) ( SIZE ( 1..9 ) ),
      extensionContainer    [2] IMPLICIT SEQUENCE {
        privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE ( 1..10 ) ) OF
        SEQUENCE {
          extId                MAP-EXTENSION .&extensionId ( {
            '...' ) ,
          extType              MAP-EXTENSION .&ExtensionType ( {
            '...' { @extId } ) OPTIONAL} OPTIONAL,
        pcs-Extensions         [1] IMPLICIT SEQUENCE {
          ... } OPTIONAL,
        ... } OPTIONAL,
        ... }
  ... }
RESULT

```

```

anyTimeInterrogationRes SEQUENCE {
  subscriberInfo SEQUENCE {
    locationInformation [0] IMPLICIT SEQUENCE {
      ageOfLocationInformation INTEGER ( 0..32767 ) OPTIONAL,
      geographicalInformation [0] IMPLICIT OCTET STRING ( SIZE ( 8 ) ) OPTIONAL,
      vlr-number [1] IMPLICIT OCTET STRING ( SIZE ( 1..20 ) ) ( SIZE ( 1..9 )
) OPTIONAL,
      locationNumber [2] IMPLICIT OCTET STRING ( SIZE ( 2..10 ) ) OPTIONAL,
      cellIdOrLAI [3] CHOICE {
        cellIdFixedLength [0] IMPLICIT OCTET STRING ( SIZE ( 7 ) ),
        laiFixedLength [1] IMPLICIT OCTET STRING ( SIZE ( 5 ) ) } OPTIONAL,
      extensionContainer [4] IMPLICIT SEQUENCE {
        privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE ( 1..10 ) ) OF
          SEQUENCE {
            extId MAP-EXTENSION .&extensionId ( {
              '...' ) ,
            extType MAP-EXTENSION .&ExtensionType ( {
              '...' { @extId } ) OPTIONAL} OPTIONAL,
            pcs-Extensions [1] IMPLICIT SEQUENCE {
              ... } OPTIONAL,
            ... } OPTIONAL,
            ... ,
            selectedLSA-Id [5] IMPLICIT OCTET STRING ( SIZE ( 3 ) ) OPTIONAL,
            msc-Number [6] IMPLICIT OCTET STRING ( SIZE ( 1..20 ) ) ( SIZE ( 1..9 )
) OPTIONAL,
            geodeticInformation [7] IMPLICIT OCTET STRING ( SIZE ( 10 ) ) OPTIONAL}
OPTIONAL,
      subscriberState [1] CHOICE {
        assumedIdle [0] IMPLICIT NULL,
        camelBusy [1] IMPLICIT NULL,
        netDetNotReachable ENUMERATED {
          msPurged ( 0 ),
          imsiDetached ( 1 ),
          restrictedArea ( 2 ),
          notRegistered ( 3 ) },
        notProvidedFromVLR [2] IMPLICIT NULL} OPTIONAL,
      extensionContainer [2] IMPLICIT SEQUENCE {
        privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE ( 1..10 ) ) OF
          SEQUENCE {
            extId MAP-EXTENSION .&extensionId ( {
              '...' ) ,
            extType MAP-EXTENSION .&ExtensionType ( {
              '...' { @extId } ) OPTIONAL} OPTIONAL,
            pcs-Extensions [1] IMPLICIT SEQUENCE {
              ... } OPTIONAL,
            ... } OPTIONAL,
            ... },
        extensionContainer SEQUENCE {
          privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE ( 1..10 ) ) OF
            SEQUENCE {
              extId MAP-EXTENSION .&extensionId ( {
                '...' ) ,
              extType MAP-EXTENSION .&ExtensionType ( {
                '...' { @extId } ) OPTIONAL} OPTIONAL,
              pcs-Extensions [1] IMPLICIT SEQUENCE {
                ... } OPTIONAL,
              ... } OPTIONAL,
              ... }
ERRORS {
  -- systemFailure -- localValue : 34,
  -- ati-NotAllowed -- localValue : 49,
  -- dataMissing -- localValue : 35,
  -- unexpectedDataValue -- localValue : 36,
  -- unknownSubscriber -- localValue : 1}
 ::= localValue : 71

anyTimeSubscriptionInterrogation OPERATION
ARGUMENT
anyTimeSubscriptionInterrogationArg SEQUENCE {
  subscriberIdentity [0] CHOICE {
    imsi [0] IMPLICIT OCTET STRING ( SIZE ( 3..8 ) ),
    msisdn [1] IMPLICIT OCTET STRING ( SIZE ( 1..20 ) ) ( SIZE ( 1..9 ) ) },
  requestedSubscriptionInfo [1] IMPLICIT SEQUENCE {
    requestedSS-Info [1] IMPLICIT SEQUENCE {
      ss-Code OCTET STRING ( SIZE ( 1 ) ),
      basicService CHOICE {
        bearerService [2] IMPLICIT OCTET STRING ( SIZE ( 1 ) ),

```



```

        teleservice          [3] IMPLICIT OCTET STRING ( SIZE ( 1 ) ) } OPTIONAL,
        ... } OPTIONAL,
    odb                      [2] IMPLICIT NULL OPTIONAL,
    requestedCAMEL-SubscriptionInfo [3] IMPLICIT ENUMERATED {
        o-CSI      ( 0 ),
        t-CSI      ( 1 ),
        vt-CSI     ( 2 ),
        tif-CSI    ( 3 ),
        gprs-CSI   ( 4 ),
        sms-CSI    ( 5 ),
        ss-CSI     ( 6 ),
        m-CSI      ( 7 ),
        d-csi      ( 8 ) } OPTIONAL,
    supportedVLR-CAMEL-Phases [4] IMPLICIT NULL OPTIONAL,
    supportedSGSN-CAMEL-Phases [5] IMPLICIT NULL OPTIONAL,
    extensionContainer        [6] IMPLICIT SEQUENCE {
        privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE ( 1..10 ) ) OF
            SEQUENCE {
                extId      MAP-EXTENSION .&extensionId ( {
                    '
                    ... } ) ,
                extType    MAP-EXTENSION .&ExtensionType ( {
                    '
                    ... } { @extId } ) OPTIONAL } OPTIONAL,
        pcs-Extensions      [1] IMPLICIT SEQUENCE {
            ... } OPTIONAL,
        ... } OPTIONAL,
        ... },
    gsmSCF-Address          [2] IMPLICIT OCTET STRING ( SIZE ( 1..20 ) ) ( SIZE ( 1..9 ) ),
    extensionContainer      [3] IMPLICIT SEQUENCE {
        privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE ( 1..10 ) ) OF
            SEQUENCE {
                extId      MAP-EXTENSION .&extensionId ( {
                    '
                    ... } ) ,
                extType    MAP-EXTENSION .&ExtensionType ( {
                    '
                    ... } { @extId } ) OPTIONAL } OPTIONAL,
        pcs-Extensions      [1] IMPLICIT SEQUENCE {
            ... } OPTIONAL,
        ... } OPTIONAL,
        ... }
    RESULT
    anyTimeSubscriptionInterrogationRes SEQUENCE {
        callForwardingData [1] IMPLICIT SEQUENCE {
            forwardingFeatureList SEQUENCE ( SIZE ( 1..32 ) ) OF
                SEQUENCE {
                    basicService CHOICE {
                        ext-BearerService [2] IMPLICIT OCTET STRING ( SIZE ( 1..5 ) ),
                        ext-Teleservice [3] IMPLICIT OCTET STRING ( SIZE ( 1..5 ) ) } OPTIONAL,
                        ss-Status [4] IMPLICIT OCTET STRING ( SIZE ( 1..5 ) ),
                        forwardedToNumber [5] IMPLICIT OCTET STRING ( SIZE ( 1..20 ) ) ( SIZE ( 1..9 ) )
                    } OPTIONAL,
                    forwardedToSubaddress [8] IMPLICIT OCTET STRING ( SIZE ( 1..21 ) ) OPTIONAL,
                    forwardingOptions [6] IMPLICIT OCTET STRING ( SIZE ( 1..5 ) ) OPTIONAL,
                    noReplyConditionTime [7] IMPLICIT INTEGER ( 1..100 ) OPTIONAL,
                    extensionContainer [9] IMPLICIT SEQUENCE {
                        privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE ( 1..10 ) ) OF
                            SEQUENCE {
                                extId      MAP-EXTENSION .&extensionId ( {
                                    '
                                    ... } ) ,
                                extType    MAP-EXTENSION .&ExtensionType ( {
                                    '
                                    ... } { @extId } ) OPTIONAL } OPTIONAL,
                            pcs-Extensions [1] IMPLICIT SEQUENCE {
                                ... } OPTIONAL,
                            ... } OPTIONAL,
                            ... },
                    notificationToCSE NULL OPTIONAL,
                    extensionContainer [0] IMPLICIT SEQUENCE {
                        privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE ( 1..10 ) ) OF
                            SEQUENCE {
                                extId      MAP-EXTENSION .&extensionId ( {
                                    '
                                    ... } ) ,
                                extType    MAP-EXTENSION .&ExtensionType ( {
                                    '
                                    ... } { @extId } ) OPTIONAL } OPTIONAL,
                            pcs-Extensions [1] IMPLICIT SEQUENCE {
                                ... } OPTIONAL,
                            ... } OPTIONAL,
                            ... } OPTIONAL,
                    ... } OPTIONAL,
                }
            }
        }
    }

```

```

callBarringData          [2] IMPLICIT SEQUENCE {
  callBarringFeatureList SEQUENCE ( SIZE (1..32 ) ) OF
    SEQUENCE {
      basicService        CHOICE {
        ext-BearerService [2] IMPLICIT OCTET STRING ( SIZE (1..5 ) ),
        ext-Teleservice   [3] IMPLICIT OCTET STRING ( SIZE (1..5 ) ) } OPTIONAL,
        ss-Status         [4] IMPLICIT OCTET STRING ( SIZE (1..5 ) ),
        extensionContainer SEQUENCE {
          privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE (1..10 ) ) OF
            SEQUENCE {
              extId        MAP-EXTENSION .&extensionId ( {
                '...' ) ,
              extType      MAP-EXTENSION .&ExtensionType ( {
                '...' { @extId } ) OPTIONAL} OPTIONAL,
            pcs-Extensions [1] IMPLICIT SEQUENCE {
              ... } OPTIONAL,
            ... } OPTIONAL,
          ... },
      password            NumericString ( FROM
        ("0"|"1"|"2"|"3"|"4"|"5"|"6"|"7"|"8"|"9" ) |SIZE (4 ) ),
      wrongPasswordAttemptsCounter INTEGER ( 0..4 ),
      notificationToCSE   NULL OPTIONAL,
      extensionContainer  SEQUENCE {
        privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE (1..10 ) ) OF
          SEQUENCE {
            extId        MAP-EXTENSION .&extensionId ( {
              '...' ) ,
            extType      MAP-EXTENSION .&ExtensionType ( {
              '...' { @extId } ) OPTIONAL} OPTIONAL,
            pcs-Extensions [1] IMPLICIT SEQUENCE {
              ... } OPTIONAL,
            ... } OPTIONAL,
            ... } OPTIONAL,
          ... },
      odb-Info            [3] IMPLICIT SEQUENCE {
        odb-Data          SEQUENCE {
          odb-GeneralData BIT STRING {
            allOG-CallsBarred (0 ),
            internationalOGCallsBarred (1 ),
            internationalOGCallsNotToHPLMN-CountryBarred (2 ),
            interzonalOGCallsBarred (6 ),
            interzonalOGCallsNotToHPLMN-CountryBarred (7 ),
            interzonalOGCallsAndInternationalOGCallsNotToHPLMN-CountryBarred (8 ),
            premiumRateInformationOGCallsBarred (3 ),
            premiumRateEntertainmentOGCallsBarred (4 ),
            ss-AccessBarred (5 ),
            allECT-Barred (9 ),
            chargeableECT-Barred (10 ),
            internationalECT-Barred (11 ),
            interzonalECT-Barred (12 ),
            doublyChargeableECT-Barred (13 ),
            multipleECT-Barred (14 )} ( SIZE (15..32 ) ),
          odb-HPLMN-Data BIT STRING {
            plmn-SpecificBarringType1 (0 ),
            plmn-SpecificBarringType2 (1 ),
            plmn-SpecificBarringType3 (2 ),
            plmn-SpecificBarringType4 (3 )} ( SIZE (4..32 ) ) OPTIONAL,
          extensionContainer SEQUENCE {
            privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE (1..10 ) ) OF
              SEQUENCE {
                extId        MAP-EXTENSION .&extensionId ( {
                  '...' ) ,
                extType      MAP-EXTENSION .&ExtensionType ( {
                  '...' { @extId } ) OPTIONAL} OPTIONAL,
                pcs-Extensions [1] IMPLICIT SEQUENCE {
                  ... } OPTIONAL,
                ... } OPTIONAL,
                ... },
            notificationToCSE NULL OPTIONAL,
            extensionContainer SEQUENCE {
              privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE (1..10 ) ) OF
                SEQUENCE {
                  extId        MAP-EXTENSION .&extensionId ( {
                    '...' ) ,
                  extType      MAP-EXTENSION .&ExtensionType ( {
                    '...' { @extId } ) OPTIONAL} OPTIONAL,

```

```

pcs-Extensions          [1] IMPLICIT SEQUENCE {
  ... } OPTIONAL,
  ... } OPTIONAL,
  ... } OPTIONAL,
camel-SubscriptionInfo  [4] IMPLICIT SEQUENCE {
  o-CSI                  [0] IMPLICIT SEQUENCE {
    o-BcsmCamelTDPDataList SEQUENCE ( SIZE (1..10) ) OF
      SEQUENCE {
        o-BcsmTriggerDetectionPoint ENUMERATED {
          collectedInfo (2),
          ... ,
          routeSelectFailure (4)},
        serviceKey          INTEGER ( 0..2147483647 ),
        gsmSCF-Address      [0] IMPLICIT OCTET STRING ( SIZE (1..20) ) (
SIZE (1..9) ),
        defaultCallHandling [1] IMPLICIT ENUMERATED {
          continueCall (0),
          releaseCall (1),
          ... },
        extensionContainer [2] IMPLICIT SEQUENCE {
          privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE (1..10) ) OF
            SEQUENCE {
              extId          MAP-EXTENSION .&extensionId ( {
                '...'} ),
              extType        MAP-EXTENSION .&ExtensionType ( {
                '...'} { @extId } ) OPTIONAL} OPTIONAL,
              pcs-Extensions [1] IMPLICIT SEQUENCE {
                ... } OPTIONAL,
                ... } OPTIONAL,
                ... },
          extensionContainer SEQUENCE {
            privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE (1..10) ) OF
              SEQUENCE {
                extId          MAP-EXTENSION .&extensionId ( {
                  '...'} ),
                extType        MAP-EXTENSION .&ExtensionType ( {
                  '...'} { @extId } ) OPTIONAL} OPTIONAL,
                pcs-Extensions [1] IMPLICIT SEQUENCE {
                  ... } OPTIONAL,
                  ... } OPTIONAL,
                  ... },
            camelCapabilityHandling [0] IMPLICIT INTEGER ( 1..16 ) OPTIONAL,
            notificationToCSE       [1] IMPLICIT NULL OPTIONAL,
            csiActive                [2] IMPLICIT NULL OPTIONAL} OPTIONAL,
          o-BcsmCamelTDP-CriteriaList [1] IMPLICIT SEQUENCE ( SIZE (1..10) ) OF
            SEQUENCE {
              o-BcsmTriggerDetectionPoint ENUMERATED {
                collectedInfo (2),
                ... ,
                routeSelectFailure (4)},
              destinationNumberCriteria [0] IMPLICIT SEQUENCE {
                matchType [0] IMPLICIT ENUMERATED {
                  inhibiting (0),
                  enabling (1)},
                destinationNumberList [1] IMPLICIT SEQUENCE ( SIZE (1..10) ) OF
                  OCTET STRING ( SIZE (1..20) ) ( SIZE (1..9) ) OPTIONAL,
                destinationNumberLengthList [2] IMPLICIT SEQUENCE ( SIZE (1..3) ) OF
                  INTEGER ( 1..15 ) OPTIONAL,
                ... } OPTIONAL,
              basicServiceCriteria [1] IMPLICIT SEQUENCE ( SIZE (1..5) ) OF
                CHOICE {
                  ext-BearerService [2] IMPLICIT OCTET STRING ( SIZE (1..5) ),
                  ext-Teleservice [3] IMPLICIT OCTET STRING ( SIZE (1..5) )} OPTIONAL,
              callTypeCriteria [2] IMPLICIT ENUMERATED {
                forwarded (0),
                notForwarded (1)} OPTIONAL,
                ... ,
              o-CauseValueCriteria [3] IMPLICIT SEQUENCE ( SIZE (1..5) ) OF
                OCTET STRING ( SIZE (1) ) OPTIONAL,
              extensionContainer [4] IMPLICIT SEQUENCE {
                privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE (1..10) ) OF
                  SEQUENCE {
                    extId          MAP-EXTENSION .&extensionId ( {
                      '...'} ),
                    extType        MAP-EXTENSION .&ExtensionType ( {
                      '...'} { @extId } ) OPTIONAL} OPTIONAL,
                    pcs-Extensions [1] IMPLICIT SEQUENCE {

```

```

        ... } OPTIONAL,
        ... } OPTIONAL} OPTIONAL,
t-CSI
  t-BcsmCamelTDPDataList SEQUENCE ( SIZE (1..10) ) OF
    SEQUENCE {
      t-BcsmTriggerDetectionPoint ENUMERATED {
        termAttemptAuthorized (12 ),
        ... ,
        tBusy (13 ),
        tNoAnswer (14 )},
      serviceKey INTEGER ( 0..2147483647 ),
      gsmSCF-Address [0] IMPLICIT OCTET STRING ( SIZE (1..20) ) (
SIZE (1..9) ),
      defaultCallHandling [1] IMPLICIT ENUMERATED {
        continueCall (0 ),
        releaseCall (1 ),
        ... },
      extensionContainer [2] IMPLICIT SEQUENCE {
        privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE (1..10) ) OF
          SEQUENCE {
            extId MAP-EXTENSION .&extensionId ( {
              ... } ),
            extType MAP-EXTENSION .&ExtensionType ( {
              ... } { @extId } ) OPTIONAL} OPTIONAL,
            pcs-Extensions [1] IMPLICIT SEQUENCE {
              ... } OPTIONAL,
              ... } OPTIONAL,
              ... },
            extensionContainer SEQUENCE {
              privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE (1..10) ) OF
                SEQUENCE {
                  extId MAP-EXTENSION .&extensionId ( {
                    ... } ),
                  extType MAP-EXTENSION .&ExtensionType ( {
                    ... } { @extId } ) OPTIONAL} OPTIONAL,
                  pcs-Extensions [1] IMPLICIT SEQUENCE {
                    ... } OPTIONAL,
                    ... } OPTIONAL,
                    ... ,
                    camelCapabilityHandling [0] IMPLICIT INTEGER ( 1..16 ) OPTIONAL,
                    notificationToCSE [1] IMPLICIT NULL OPTIONAL,
                    csi-Active [2] IMPLICIT NULL OPTIONAL} OPTIONAL,
                    t-BCSM-CAMEL-TDP-CriteriaList [3] IMPLICIT SEQUENCE ( SIZE (1..10) ) OF
                      SEQUENCE {
                        t-BCSM-TriggerDetectionPoint ENUMERATED {
                          termAttemptAuthorized (12 ),
                          ... ,
                          tBusy (13 ),
                          tNoAnswer (14 )},
                        basicServiceCriteria [0] IMPLICIT SEQUENCE ( SIZE (1..5) ) OF
                          CHOICE {
                            ext-BearerService [2] IMPLICIT OCTET STRING ( SIZE (1..5) ),
                            ext-Teleservice [3] IMPLICIT OCTET STRING ( SIZE (1..5) )} OPTIONAL,
                            t-CauseValueCriteria [1] IMPLICIT SEQUENCE ( SIZE (1..5) ) OF
                              OCTET STRING ( SIZE (1) ) OPTIONAL,
                              ... } OPTIONAL,
                              vt-CSI [4] IMPLICIT SEQUENCE {
                                t-BcsmCamelTDPDataList SEQUENCE ( SIZE (1..10) ) OF
                                  SEQUENCE {
                                    t-BcsmTriggerDetectionPoint ENUMERATED {
                                      termAttemptAuthorized (12 ),
                                      ... ,
                                      tBusy (13 ),
                                      tNoAnswer (14 )},
                                    serviceKey INTEGER ( 0..2147483647 ),
                                    gsmSCF-Address [0] IMPLICIT OCTET STRING ( SIZE (1..20) ) (
SIZE (1..9) ),
                                    defaultCallHandling [1] IMPLICIT ENUMERATED {
                                      continueCall (0 ),
                                      releaseCall (1 ),
                                      ... },
                                    extensionContainer [2] IMPLICIT SEQUENCE {
                                      privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE (1..10) ) OF
                                        SEQUENCE {
                                          extId MAP-EXTENSION .&extensionId ( {
                                            ... } ),
                                          extType MAP-EXTENSION .&ExtensionType ( {

```

```

        ... } { @extId } ) OPTIONAL} OPTIONAL,
        pcs-Extensions [1] IMPLICIT SEQUENCE {
            ... } OPTIONAL,
            ... } OPTIONAL,
            ... },
extensionContainer SEQUENCE {
    privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE (1..10 ) ) OF
        SEQUENCE {
            extId MAP-EXTENSION .&extensionId ( {
                '... } ) ,
            extType MAP-EXTENSION .&ExtensionType ( {
                '... } { @extId } ) OPTIONAL} OPTIONAL,
        pcs-Extensions [1] IMPLICIT SEQUENCE {
            ... } OPTIONAL,
            ... } OPTIONAL,
            ... },
camelCapabilityHandling [0] IMPLICIT INTEGER ( 1..16 ) OPTIONAL,
notificationToCSE [1] IMPLICIT NULL OPTIONAL,
csi-Active [2] IMPLICIT NULL OPTIONAL} OPTIONAL,
vt-BCSM-CAMEL-TDP-CriteriaList [5] IMPLICIT SEQUENCE ( SIZE (1..10 ) ) OF
SEQUENCE {
    t-BCSM-TriggerDetectionPoint ENUMERATED {
        termAttemptAuthorized (12 ) ,
        ... ,
        tBusy (13 ) ,
        tNoAnswer (14 ) } ,
    basicServiceCriteria [0] IMPLICIT SEQUENCE ( SIZE (1..5 ) ) OF
        CHOICE {
            ext-BearerService [2] IMPLICIT OCTET STRING ( SIZE (1..5 ) ) ,
            ext-Teleservice [3] IMPLICIT OCTET STRING ( SIZE (1..5 ) ) } OPTIONAL,
    t-CauseValueCriteria [1] IMPLICIT SEQUENCE ( SIZE (1..5 ) ) OF
        OCTET STRING ( SIZE (1 ) ) OPTIONAL,
        ... } OPTIONAL,
tif-CSI [6] IMPLICIT NULL OPTIONAL,
tif-CSI-NotificationToCSE [7] IMPLICIT NULL OPTIONAL,
gprs-CSI [8] IMPLICIT SEQUENCE {
    gprs-CamelTDPDataList [0] IMPLICIT SEQUENCE ( SIZE (1..10 ) ) OF
        SEQUENCE {
            gprs-TriggerDetectionPoint [0] IMPLICIT ENUMERATED {
                attach (1 ) ,
                attachChangeOfPosition (2 ) ,
                pdp-ContextEstablishment (11 ) ,
                pdp-ContextEstablishmentAcknowledgement (12 ) ,
                pdp-ContextChangeOfPosition (14 ) ,
                ... } ,
            serviceKey [1] IMPLICIT INTEGER ( 0..2147483647 ) ,
            gsmSCF-Address [2] IMPLICIT OCTET STRING ( SIZE (1..20 ) ) ( SIZE
(1..9 ) ) ,
            defaultSessionHandling [3] IMPLICIT ENUMERATED {
                continueTransaction (0 ) ,
                releaseTransaction (1 ) ,
                ... } ,
            extensionContainer [4] IMPLICIT SEQUENCE {
                privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE (1..10 ) ) OF
                    SEQUENCE {
                        extId MAP-EXTENSION .&extensionId ( {
                            '... } ) ,
                        extType MAP-EXTENSION .&ExtensionType ( {
                            '... } { @extId } ) OPTIONAL} OPTIONAL,
                    pcs-Extensions [1] IMPLICIT SEQUENCE {
                        ... } OPTIONAL,
                        ... } OPTIONAL,
                        ... },
                camelCapabilityHandling [1] IMPLICIT INTEGER ( 1..16 ) ,
                extensionContainer [2] IMPLICIT SEQUENCE {
                    privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE (1..10 ) ) OF
                        SEQUENCE {
                            extId MAP-EXTENSION .&extensionId ( {
                                '... } ) ,
                            extType MAP-EXTENSION .&ExtensionType ( {
                                '... } { @extId } ) OPTIONAL} OPTIONAL,
                            pcs-Extensions [1] IMPLICIT SEQUENCE {
                                ... } OPTIONAL,
                                ... } OPTIONAL,
                                ... },
                    notificationToCSE [3] IMPLICIT NULL OPTIONAL,
                    csiActive [4] IMPLICIT NULL OPTIONAL,
                    ... } OPTIONAL,

```

```

sms-CSI [9] IMPLICIT SEQUENCE {
  sms-CAMEL-TDP-DataList [0] IMPLICIT SEQUENCE ( SIZE (1..10 ) ) OF
  SEQUENCE {
    sms-TriggerDetectionPoint [0] IMPLICIT ENUMERATED {
      sms-CollectedInfo (1 ) ,
      ... } ,
    serviceKey [1] IMPLICIT INTEGER ( 0..2147483647 ) ,
    gsmSCF-Address [2] IMPLICIT OCTET STRING ( SIZE (1..20 ) ) ( SIZE
(1..9 ) ) ,
    defaultSMS-Handling [3] IMPLICIT ENUMERATED {
      continueTransaction (0 ) ,
      releaseTransaction (1 ) ,
      ... } ,
    extensionContainer [4] IMPLICIT SEQUENCE {
      privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE (1..10 ) ) OF
      SEQUENCE {
        extId MAP-EXTENSION .&extensionId ( {
          '...' } ) ,
        extType MAP-EXTENSION .&ExtensionType ( {
          '...' { @extId } ) OPTIONAL} OPTIONAL ,
        pcs-Extensions [1] IMPLICIT SEQUENCE {
          ... } OPTIONAL ,
          ... } OPTIONAL ,
          ... } ,
    camelCapabilityHandling [1] IMPLICIT INTEGER ( 1..16 ) ,
    extensionContainer [2] IMPLICIT SEQUENCE {
      privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE (1..10 ) ) OF
      SEQUENCE {
        extId MAP-EXTENSION .&extensionId ( {
          '...' } ) ,
        extType MAP-EXTENSION .&ExtensionType ( {
          '...' { @extId } ) OPTIONAL} OPTIONAL ,
        pcs-Extensions [1] IMPLICIT SEQUENCE {
          ... } OPTIONAL ,
          ... } OPTIONAL ,
          notificationToCSE [3] IMPLICIT NULL OPTIONAL ,
          csiActive [4] IMPLICIT NULL OPTIONAL ,
          ... } OPTIONAL ,
    ss-CSI [10] IMPLICIT SEQUENCE {
      ss-CamelData SEQUENCE {
        ss-EventList SEQUENCE ( SIZE (1..10 ) ) OF
        OCTET STRING ( SIZE (1 ) ) ,
        gsmSCF-Address OCTET STRING ( SIZE (1..20 ) ) ( SIZE (1..9 ) ) ,
        extensionContainer [0] IMPLICIT SEQUENCE {
          privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE (1..10 ) ) OF
          SEQUENCE {
            extId MAP-EXTENSION .&extensionId ( {
              '...' } ) ,
            extType MAP-EXTENSION .&ExtensionType ( {
              '...' { @extId } ) OPTIONAL} OPTIONAL ,
            pcs-Extensions [1] IMPLICIT SEQUENCE {
              ... } OPTIONAL ,
              ... } OPTIONAL ,
              ... } ,
            notificationToCSE [1] IMPLICIT NULL OPTIONAL ,
            csiActive [2] IMPLICIT NULL OPTIONAL} ,
            extensionContainer SEQUENCE {
              privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE (1..10 ) ) OF
              SEQUENCE {
                extId MAP-EXTENSION .&extensionId ( {
                  '...' } ) ,
                extType MAP-EXTENSION .&ExtensionType ( {
                  '...' { @extId } ) OPTIONAL} OPTIONAL ,
                pcs-Extensions [1] IMPLICIT SEQUENCE {
                  ... } OPTIONAL ,
                  ... } OPTIONAL ,
                  ... } OPTIONAL ,
            m-CSI [11] IMPLICIT SEQUENCE {
              mobilityTriggers SEQUENCE ( SIZE (1..10 ) ) OF
              OCTET STRING ( SIZE (1 ) ) ,
              serviceKey INTEGER ( 0..2147483647 ) ,
              gsmSCF-Address [0] IMPLICIT OCTET STRING ( SIZE (1..20 ) ) ( SIZE (1..9 ) ) ,
              extensionContainer [1] IMPLICIT SEQUENCE {
                privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE (1..10 ) ) OF
                SEQUENCE {

```

```

        extId      MAP-EXTENSION .&extensionId ( {
            '...' ) ,
        extType    MAP-EXTENSION .&ExtensionType ( {
            '...' { @extId } ) OPTIONAL} OPTIONAL,
        pcs-Extensions [1] IMPLICIT SEQUENCE {
            ... } OPTIONAL,
            ... } OPTIONAL,
        notificationToCSE [2] IMPLICIT NULL OPTIONAL,
        csiActive [3] IMPLICIT NULL OPTIONAL,
            ... } OPTIONAL,
        extensionContainer [12] IMPLICIT SEQUENCE {
        privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE (1..10 ) ) OF
        SEQUENCE {
            extId      MAP-EXTENSION .&extensionId ( {
                '...' ) ,
            extType    MAP-EXTENSION .&ExtensionType ( {
                '...' { @extId } ) OPTIONAL} OPTIONAL,
            pcs-Extensions [1] IMPLICIT SEQUENCE {
                ... } OPTIONAL,
                ... } OPTIONAL,
                ... } OPTIONAL,
        supportedVLR-CAMEL-Phases [5] IMPLICIT BIT STRING {
            phase1 ( 0 ),
            phase2 ( 1 ),
            phase3 ( 2 ) } ( SIZE (1..16 ) ) OPTIONAL,
        supportedSGSN-CAMEL-Phases [6] IMPLICIT BIT STRING {
            phase1 ( 0 ),
            phase2 ( 1 ),
            phase3 ( 2 ) } ( SIZE (1..16 ) ) OPTIONAL,
        extensionContainer [7] IMPLICIT SEQUENCE {
        privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE (1..10 ) ) OF
        SEQUENCE {
            extId      MAP-EXTENSION .&extensionId ( {
                '...' ) ,
            extType    MAP-EXTENSION .&ExtensionType ( {
                '...' { @extId } ) OPTIONAL} OPTIONAL,
            pcs-Extensions [1] IMPLICIT SEQUENCE {
                ... } OPTIONAL,
                ... } OPTIONAL,
                ... }
        ... }
ERRORS {
-- atsi-NotAllowed -- localValue : 60,
-- 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,
-- informationNotAvailable -- localValue : 62}
 ::= localValue : 62

anyTimeModification OPERATION
ARGUMENT
    anyTimeModificationArg SEQUENCE {
        subscriberIdentity [0] CHOICE {
            imsi [0] IMPLICIT OCTET STRING ( SIZE (3..8 ) ),
            msisdn [1] IMPLICIT OCTET STRING ( SIZE (1..20 ) ) ( SIZE (1..9 ) )},
            gsmSCF-Address [1] IMPLICIT OCTET STRING ( SIZE (1..20 ) ) ( SIZE (1..9 ) )
        },
        modificationRequestFor-SS-Info [2] IMPLICIT SEQUENCE {
            ss-Code [0] IMPLICIT OCTET STRING ( SIZE (1 ) ),
            basicService [1] CHOICE {
                ext-BearerService [2] IMPLICIT OCTET STRING ( SIZE (1..5 ) ),
                ext-Teleservice [3] IMPLICIT OCTET STRING ( SIZE (1..5 ) )} OPTIONAL,
            ss-Status [2] IMPLICIT OCTET STRING ( SIZE (1..5 ) ) OPTIONAL,
            forwardedToNumber [3] IMPLICIT OCTET STRING ( SIZE (1..20 ) ) OPTIONAL,
            forwardedToSubaddress [4] IMPLICIT OCTET STRING ( SIZE (1..21 ) ) OPTIONAL,
            noReplyConditionTime [5] IMPLICIT INTEGER ( 1..100 ) OPTIONAL,
            modifyNotificationToCSE [6] IMPLICIT ENUMERATED {
                deactivate ( 0 ),
                activate ( 1 )} OPTIONAL,
            extensionContainer [7] IMPLICIT SEQUENCE {
                privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE (1..10 ) ) OF
                SEQUENCE {
                    extId      MAP-EXTENSION .&extensionId ( {

```

```

        ... } ) ,
        extType    MAP-EXTENSION .&ExtensionType ( {
            ... } { @extId } ) OPTIONAL} OPTIONAL,
        pcs-Extensions [1] IMPLICIT SEQUENCE {
            ... } OPTIONAL,
            ... } OPTIONAL,
            ... } OPTIONAL,
modificationRequestFor-CSI [3] IMPLICIT SEQUENCE {
    requestedCamelSubscriptionInfo [0] IMPLICIT ENUMERATED {
        o-CSI (0 ),
        t-CSI (1 ),
        vt-CSI (2 ),
        tif-CSI (3 ),
        gprs-CSI (4 ),
        sms-CSI (5 ),
        ss-CSI (6 ),
        m-CSI (7 ),
        d-csi (8 )} OPTIONAL,
    modifyNotificationToCSE [1] IMPLICIT ENUMERATED {
        deactivate (0 ),
        activate (1 )} OPTIONAL,
    modifyCSI-State [2] IMPLICIT ENUMERATED {
        deactivate (0 ),
        activate (1 )} OPTIONAL,
    extensionContainer [3] IMPLICIT SEQUENCE {
        privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE (1..10 ) ) OF
        SEQUENCE {
            extId    MAP-EXTENSION .&extensionId ( {
                ... } ) ,
            extType    MAP-EXTENSION .&ExtensionType ( {
                ... } { @extId } ) OPTIONAL} OPTIONAL,
            pcs-Extensions [1] IMPLICIT SEQUENCE {
                ... } OPTIONAL,
                ... } OPTIONAL,
                ... } OPTIONAL,
                ... } OPTIONAL,
extensionContainer [4] IMPLICIT SEQUENCE {
    privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE (1..10 ) ) OF
    SEQUENCE {
        extId    MAP-EXTENSION .&extensionId ( {
            ... } ) ,
            extType    MAP-EXTENSION .&ExtensionType ( {
                ... } { @extId } ) OPTIONAL} OPTIONAL,
            pcs-Extensions [1] IMPLICIT SEQUENCE {
                ... } OPTIONAL,
                ... } OPTIONAL,
                ... } OPTIONAL,
                ... }
RESULT
anyTimeModificationRes SEQUENCE {
    ss-InfoFor-CSE [0] CHOICE {
        forwardingInfoFor-CSE [0] IMPLICIT SEQUENCE {
            ss-Code [0] IMPLICIT OCTET STRING ( SIZE (1 ) ),
            forwardingFeatureList [1] IMPLICIT SEQUENCE ( SIZE (1..32 ) ) OF
            SEQUENCE {
                basicService CHOICE {
                    ext-BearerService [2] IMPLICIT OCTET STRING ( SIZE (1..5 ) ),
                    ext-Teleservice [3] IMPLICIT OCTET STRING ( SIZE (1..5 ) )} OPTIONAL,
                    ss-Status [4] IMPLICIT OCTET STRING ( SIZE (1..5 ) ),
                    forwardedToNumber [5] IMPLICIT OCTET STRING ( SIZE (1..20 ) ) ( SIZE
(1..9 ) ) OPTIONAL,
                    forwardedToSubaddress [8] IMPLICIT OCTET STRING ( SIZE (1..21 ) ) OPTIONAL,
                    forwardingOptions [6] IMPLICIT OCTET STRING ( SIZE (1..5 ) ) OPTIONAL,
                    noReplyConditionTime [7] IMPLICIT INTEGER ( 1..100 ) OPTIONAL,
                    extensionContainer [9] IMPLICIT SEQUENCE {
                        privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE (1..10 ) ) OF
                        SEQUENCE {
                            extId    MAP-EXTENSION .&extensionId ( {
                                ... } ) ,
                                extType    MAP-EXTENSION .&ExtensionType ( {
                                    ... } { @extId } ) OPTIONAL} OPTIONAL,
                                    pcs-Extensions [1] IMPLICIT SEQUENCE {
                                        ... } OPTIONAL,
                                        ... } OPTIONAL,
                                        ... } OPTIONAL,
                                        ... } OPTIONAL,
                                notificationToCSE [2] IMPLICIT NULL,
                                extensionContainer [3] IMPLICIT SEQUENCE {

```



```

privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE (1..10 ) ) OF
  SEQUENCE {
    extId      MAP-EXTENSION .&extensionId ( {
      ... } ) ,
    extType    MAP-EXTENSION .&ExtensionType ( {
      ... } { @extId } ) OPTIONAL} OPTIONAL,
  pcs-Extensions [1] IMPLICIT SEQUENCE {
    ... } OPTIONAL,
  ... } OPTIONAL,
  ... },
callBarringInfoFor-CSE [1] IMPLICIT SEQUENCE {
  ss-Code [0] IMPLICIT OCTET STRING ( SIZE ( 1 ) ),
  callBarringFeatureList [1] IMPLICIT SEQUENCE ( SIZE (1..32 ) ) OF
  SEQUENCE {
    basicService CHOICE {
      ext-BearerService [2] IMPLICIT OCTET STRING ( SIZE (1..5 ) ),
      ext-Teleservice [3] IMPLICIT OCTET STRING ( SIZE (1..5 ) )} OPTIONAL,
    ss-Status [4] IMPLICIT OCTET STRING ( SIZE (1..5 ) ),
    extensionContainer SEQUENCE {
      privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE (1..10 ) ) OF
      SEQUENCE {
        extId      MAP-EXTENSION .&extensionId ( {
          ... } ) ,
        extType    MAP-EXTENSION .&ExtensionType ( {
          ... } { @extId } ) OPTIONAL} OPTIONAL,
        pcs-Extensions [1] IMPLICIT SEQUENCE {
          ... } OPTIONAL,
        ... } OPTIONAL,
        ... },
      password [2] IMPLICIT NumericString ( FROM
        ("0"|"1"|"2"|"3"|"4"|"5"|"6"|"7"|"8"|"9" )|SIZE ( 4 ) ),
      wrongPasswordAttemptsCounter [3] IMPLICIT INTEGER ( 0..4 ),
      notificationToCSE [4] IMPLICIT NULL,
      extensionContainer [5] IMPLICIT SEQUENCE {
        privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE (1..10 ) ) OF
        SEQUENCE {
          extId      MAP-EXTENSION .&extensionId ( {
            ... } ) ,
          extType    MAP-EXTENSION .&ExtensionType ( {
            ... } { @extId } ) OPTIONAL} OPTIONAL,
          pcs-Extensions [1] IMPLICIT SEQUENCE {
            ... } OPTIONAL,
            ... } OPTIONAL,
            ... } } OPTIONAL,
          camel-SubcriptionInfo [1] IMPLICIT SEQUENCE {
            o-CSI [0] IMPLICIT SEQUENCE {
              o-BcsmCamelTDPDataList SEQUENCE ( SIZE (1..10 ) ) OF
              SEQUENCE {
                o-BcsmTriggerDetectionPoint ENUMERATED {
                  collectedInfo ( 2 ) ,
                  ... ,
                  routeSelectFailure ( 4 )},
                serviceKey INTEGER ( 0..2147483647 ) ,
                gsmSCF-Address [0] IMPLICIT OCTET STRING ( SIZE (1..20 ) ) (
SIZE (1..9 ) ) ,
                defaultCallHandling [1] IMPLICIT ENUMERATED {
                  continueCall ( 0 ) ,
                  releaseCall ( 1 ) ,
                  ... },
                extensionContainer [2] IMPLICIT SEQUENCE {
                  privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE (1..10 ) ) OF
                  SEQUENCE {
                    extId      MAP-EXTENSION .&extensionId ( {
                      ... } ) ,
                    extType    MAP-EXTENSION .&ExtensionType ( {
                      ... } { @extId } ) OPTIONAL} OPTIONAL,
                    pcs-Extensions [1] IMPLICIT SEQUENCE {
                      ... } OPTIONAL,
                      ... } OPTIONAL,
                      ... },
                    extensionContainer SEQUENCE {
                      privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE (1..10 ) ) OF
                      SEQUENCE {
                        extId      MAP-EXTENSION .&extensionId ( {

```

```

        ... } ) ,
        extType      MAP-EXTENSION .&ExtensionType ( {
            '... } { @extId } ) OPTIONAL } OPTIONAL,
    pcs-Extensions  [1] IMPLICIT SEQUENCE {
        ... } OPTIONAL,
        ... } OPTIONAL,
    ... ,
    camelCapabilityHandling [0] IMPLICIT INTEGER ( 1..16 ) OPTIONAL,
    notificationToCSE      [1] IMPLICIT NULL OPTIONAL,
    csiActive               [2] IMPLICIT NULL OPTIONAL } OPTIONAL,
o-BcsmCamelTDP-CriteriaList [1] IMPLICIT SEQUENCE ( SIZE (1..10) ) OF
SEQUENCE {
    o-BcsmTriggerDetectionPoint  ENUMERATED {
        collectedInfo          ( 2 ) ,
        ... ,
        routeSelectFailure     ( 4 ) } ,
    destinationNumberCriteria    [0] IMPLICIT SEQUENCE {
        matchType               [0] IMPLICIT ENUMERATED {
            inhibiting          ( 0 ) ,
            enabling            ( 1 ) } ,
        destinationNumberList   [1] IMPLICIT SEQUENCE ( SIZE (1..10) ) OF
            OCTET STRING ( SIZE (1..20) ) ( SIZE (1..9) ) OPTIONAL,
        destinationNumberLengthList [2] IMPLICIT SEQUENCE ( SIZE (1..3) ) OF
            INTEGER ( 1..15 ) OPTIONAL,
        ... } OPTIONAL,
    basicServiceCriteria        [1] IMPLICIT SEQUENCE ( SIZE (1..5) ) OF
        CHOICE {
            ext-BearerService    [2] IMPLICIT OCTET STRING ( SIZE (1..5) ) ,
            ext-Teleservice      [3] IMPLICIT OCTET STRING ( SIZE (1..5) ) } OPTIONAL,
    callTypeCriteria           [2] IMPLICIT ENUMERATED {
        forwarded                ( 0 ) ,
        notForwarded            ( 1 ) } OPTIONAL,
    ... ,
    o-CauseValueCriteria       [3] IMPLICIT SEQUENCE ( SIZE (1..5) ) OF
        OCTET STRING ( SIZE (1) ) OPTIONAL,
    extensionContainer         [4] IMPLICIT SEQUENCE {
        privateExtensionList    [0] IMPLICIT SEQUENCE ( SIZE (1..10) ) OF
            SEQUENCE {
                extId            MAP-EXTENSION .&extensionId ( {
                    '... } ) ,
                extType          MAP-EXTENSION .&ExtensionType ( {
                    '... } { @extId } ) OPTIONAL } OPTIONAL,
            pcs-Extensions      [1] IMPLICIT SEQUENCE {
                ... } OPTIONAL,
                ... } OPTIONAL,
            ... } OPTIONAL,
        ... } ,
    t-CSI                       [2] IMPLICIT SEQUENCE {
        t-BcsmCamelTDPDataList  SEQUENCE ( SIZE (1..10) ) OF
            SEQUENCE {
                t-BcsmTriggerDetectionPoint  ENUMERATED {
                    termAttemptAuthorized    (12) ,
                    ... ,
                    tBusy                    (13) ,
                    tNoAnswer                (14) } ,
                serviceKey                   INTEGER ( 0..2147483647 ) ,
                gsmSCF-Address               [0] IMPLICIT OCTET STRING ( SIZE (1..20) ) (
SIZE (1..9) ) ,
                defaultCallHandling         [1] IMPLICIT ENUMERATED {
                    continueCall            ( 0 ) ,
                    releaseCall             ( 1 ) ,
                    ... } ,
                extensionContainer          [2] IMPLICIT SEQUENCE {
                    privateExtensionList    [0] IMPLICIT SEQUENCE ( SIZE (1..10) ) OF
                        SEQUENCE {
                            extId            MAP-EXTENSION .&extensionId ( {
                                '... } ) ,
                            extType          MAP-EXTENSION .&ExtensionType ( {
                                    '... } { @extId } ) OPTIONAL } OPTIONAL,
                                pcs-Extensions      [1] IMPLICIT SEQUENCE {
                                    ... } OPTIONAL,
                                    ... } OPTIONAL,
                                ... } ,
                    extensionContainer      SEQUENCE {
                        privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE (1..10) ) OF
                            SEQUENCE {
                                extId            MAP-EXTENSION .&extensionId ( {
                                    '... } ) ,
                                    extType          MAP-EXTENSION .&ExtensionType ( {

```

```

        ... } { @extId } ) OPTIONAL} OPTIONAL,
    pcs-Extensions [1] IMPLICIT SEQUENCE {
        ... } OPTIONAL,
        ... } OPTIONAL,
    ...
    camelCapabilityHandling [0] IMPLICIT INTEGER ( 1..16 ) OPTIONAL,
    notificationToCSE [1] IMPLICIT NULL OPTIONAL,
    csi-Active [2] IMPLICIT NULL OPTIONAL} OPTIONAL,
    t-BCSM-CAMEL-TDP-CriteriaList [3] IMPLICIT SEQUENCE ( SIZE (1..10) ) OF
    SEQUENCE {
        t-BCSM-TriggerDetectionPoint ENUMERATED {
            termAttemptAuthorized (12 ),
            ...
            tBusy (13 ),
            tNoAnswer (14 )},
        basicServiceCriteria [0] IMPLICIT SEQUENCE ( SIZE (1..5) ) OF
        CHOICE {
            ext-BearerService [2] IMPLICIT OCTET STRING ( SIZE (1..5) ),
            ext-Teleservice [3] IMPLICIT OCTET STRING ( SIZE (1..5) )} OPTIONAL,
        t-CauseValueCriteria [1] IMPLICIT SEQUENCE ( SIZE (1..5) ) OF
        OCTET STRING ( SIZE (1) ) OPTIONAL,
        ... } OPTIONAL,
    vt-CSI [4] IMPLICIT SEQUENCE {
        t-BcsmCamelTDPDataList SEQUENCE ( SIZE (1..10) ) OF
        SEQUENCE {
            t-BcsmTriggerDetectionPoint ENUMERATED {
                termAttemptAuthorized (12 ),
                ...
                tBusy (13 ),
                tNoAnswer (14 )},
            serviceKey INTEGER ( 0..2147483647 ),
            gsmSCF-Address [0] IMPLICIT OCTET STRING ( SIZE (1..20) ) (
SIZE (1..9) ),
            defaultCallHandling [1] IMPLICIT ENUMERATED {
                continueCall (0 ),
                releaseCall (1 ),
                ... },
            extensionContainer [2] IMPLICIT SEQUENCE {
                privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE (1..10) ) OF
                SEQUENCE {
                    extId MAP-EXTENSION .&extensionId ( {
                        ... } ),
                    extType MAP-EXTENSION .&ExtensionType ( {
                        ... } { @extId } ) OPTIONAL} OPTIONAL,
                pcs-Extensions [1] IMPLICIT SEQUENCE {
                    ... } OPTIONAL,
                    ... } OPTIONAL,
                ... },
            extensionContainer SEQUENCE {
                privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE (1..10) ) OF
                SEQUENCE {
                    extId MAP-EXTENSION .&extensionId ( {
                        ... } ),
                    extType MAP-EXTENSION .&ExtensionType ( {
                        ... } { @extId } ) OPTIONAL} OPTIONAL,
                pcs-Extensions [1] IMPLICIT SEQUENCE {
                    ... } OPTIONAL,
                    ... } OPTIONAL,
                ...
            camelCapabilityHandling [0] IMPLICIT INTEGER ( 1..16 ) OPTIONAL,
            notificationToCSE [1] IMPLICIT NULL OPTIONAL,
            csi-Active [2] IMPLICIT NULL OPTIONAL} OPTIONAL,
            vt-BCSM-CAMEL-TDP-CriteriaList [5] IMPLICIT SEQUENCE ( SIZE (1..10) ) OF
            SEQUENCE {
                t-BCSM-TriggerDetectionPoint ENUMERATED {
                    termAttemptAuthorized (12 ),
                    ...
                    tBusy (13 ),
                    tNoAnswer (14 )},
                basicServiceCriteria [0] IMPLICIT SEQUENCE ( SIZE (1..5) ) OF
                CHOICE {
                    ext-BearerService [2] IMPLICIT OCTET STRING ( SIZE (1..5) ),
                    ext-Teleservice [3] IMPLICIT OCTET STRING ( SIZE (1..5) )} OPTIONAL,
                t-CauseValueCriteria [1] IMPLICIT SEQUENCE ( SIZE (1..5) ) OF
                OCTET STRING ( SIZE (1) ) OPTIONAL,
                ... } OPTIONAL,
            tif-CSI [6] IMPLICIT NULL OPTIONAL,
            tif-CSI-NotificationToCSE [7] IMPLICIT NULL OPTIONAL,

```

```

gprs-C SI [8] IMPLICIT SEQUENCE {
  gprs-CamelTDPDataList [0] IMPLICIT SEQUENCE ( SIZE (1..10 ) ) OF
  SEQUENCE {
    gprs-TriggerDetectionPoint [0] IMPLICIT ENUMERATED {
      attach (1 ),
      attachChangeOfPosition (2 ),
      pdp-ContextEstablishment (11 ),
      pdp-ContextEstablishmentAcknowledgement (12 ),
      pdp-ContextChangeOfPosition (14 ),
      ... },
    serviceKey [1] IMPLICIT INTEGER ( 0..2147483647 ),
    gsmSCF-Address [2] IMPLICIT OCTET STRING ( SIZE (1..20 ) ) ( SIZE
(1..9 ) ),
    defaultSessionHandling [3] IMPLICIT ENUMERATED {
      continueTransaction (0 ),
      releaseTransaction (1 ),
      ... },
    extensionContainer [4] IMPLICIT SEQUENCE {
      privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE (1..10 ) ) OF
      SEQUENCE {
        extId MAP-EXTENSION .&extensionId ( {
          ... } ) ,
        extType MAP-EXTENSION .&ExtensionType ( {
          ... } { @extId } ) OPTIONAL} OPTIONAL,
      pcs-Extensions [1] IMPLICIT SEQUENCE {
        ... } OPTIONAL,
        ... } OPTIONAL,
        ... },
    camelCapabilityHandling [1] IMPLICIT INTEGER ( 1..16 ),
    extensionContainer [2] IMPLICIT SEQUENCE {
      privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE (1..10 ) ) OF
      SEQUENCE {
        extId MAP-EXTENSION .&extensionId ( {
          ... } ) ,
        extType MAP-EXTENSION .&ExtensionType ( {
          ... } { @extId } ) OPTIONAL} OPTIONAL,
      pcs-Extensions [1] IMPLICIT SEQUENCE {
        ... } OPTIONAL,
        ... } OPTIONAL,
      notificationToCSE [3] IMPLICIT NULL OPTIONAL,
      csiActive [4] IMPLICIT NULL OPTIONAL,
      ... } OPTIONAL,
    sms-C SI [9] IMPLICIT SEQUENCE {
      sms-CAMEL-TDP-DataList [0] IMPLICIT SEQUENCE ( SIZE (1..10 ) ) OF
      SEQUENCE {
        sms-TriggerDetectionPoint [0] IMPLICIT ENUMERATED {
          sms-CollectedInfo (1 ),
          ... },
        serviceKey [1] IMPLICIT INTEGER ( 0..2147483647 ),
        gsmSCF-Address [2] IMPLICIT OCTET STRING ( SIZE (1..20 ) ) ( SIZE
(1..9 ) ),
        defaultSMS-Handling [3] IMPLICIT ENUMERATED {
          continueTransaction (0 ),
          releaseTransaction (1 ),
          ... },
        extensionContainer [4] IMPLICIT SEQUENCE {
          privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE (1..10 ) ) OF
          SEQUENCE {
            extId MAP-EXTENSION .&extensionId ( {
              ... } ) ,
            extType MAP-EXTENSION .&ExtensionType ( {
              ... } { @extId } ) OPTIONAL} OPTIONAL,
          pcs-Extensions [1] IMPLICIT SEQUENCE {
            ... } OPTIONAL,
            ... } OPTIONAL,
            ... },
          camelCapabilityHandling [1] IMPLICIT INTEGER ( 1..16 ),
          extensionContainer [2] IMPLICIT SEQUENCE {
            privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE (1..10 ) ) OF
            SEQUENCE {
              extId MAP-EXTENSION .&extensionId ( {
                ... } ) ,
              extType MAP-EXTENSION .&ExtensionType ( {
                ... } { @extId } ) OPTIONAL} OPTIONAL,
              pcs-Extensions [1] IMPLICIT SEQUENCE {

```



```

... }
ERRORS {
-- atm-NotAllowed -- localValue : 61,
-- dataMissing -- localValue : 35,
-- unexpectedDataValue -- localValue : 36,
-- unknownSubscriber -- localValue : 1,
-- bearerServiceNotProvisioned -- localValue : 10,
-- teleserviceNotProvisioned -- localValue : 11,
-- callBarred -- localValue : 13,
-- illegalSS-Operation -- localValue : 16,
-- ss-SubscriptionViolation -- localValue : 19,
-- ss-ErrorStatus -- localValue : 17,
-- ss-Incompatibility -- localValue : 20,
-- informationNotAvailable -- localValue : 62}
 ::= localValue : 65

noteSubscriberDataModified OPERATION
ARGUMENT
noteSubscriberDataModifiedArg SEQUENCE {
imsi OCTET STRING ( SIZE (3..8) ),
msisdn OCTET STRING ( SIZE (1..20) ) ( SIZE (1..9) ),
typeOfModification ENUMERATED {
callForwardingSS-Data (0),
callBarringSS-Data (1),
operatorDeterminedBarringData (2),
camelSubscriptionInformation (3),
... },
extensionContainer SEQUENCE {
privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE (1..10) ) OF
SEQUENCE {
extId MAP-EXTENSION .&extensionId ( {
'...' } ),
extType MAP-EXTENSION .&ExtensionType ( {
'...' { @extId } ) OPTIONAL} OPTIONAL,
pcs-Extensions [1] IMPLICIT SEQUENCE {
... } OPTIONAL,
... } OPTIONAL,
... }
}
RESULT
noteSubscriberDataModifiedRes SEQUENCE {
extensionContainer SEQUENCE {
privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE (1..10) ) OF
SEQUENCE {
extId MAP-EXTENSION .&extensionId ( {
'...' } ),
extType MAP-EXTENSION .&ExtensionType ( {
'...' { @extId } ) OPTIONAL} OPTIONAL,
pcs-Extensions [1] IMPLICIT SEQUENCE {
... } OPTIONAL,
... } OPTIONAL,
... }
}
ERRORS {
-- unexpectedDataValue -- localValue : 36,
-- unknownSubscriber -- localValue : 1}
 ::= localValue : 5

ss-InvocationNotification OPERATION
ARGUMENT
ss-InvocationNotificationArg SEQUENCE {
imsi [0] IMPLICIT OCTET STRING ( SIZE (3..8) ),
msisdn [1] IMPLICIT OCTET STRING ( SIZE (1..20) ) ( SIZE (1..9) ),
ss-Event [2] IMPLICIT OCTET STRING ( SIZE (1) ),
ss-EventSpecification [3] IMPLICIT SEQUENCE ( SIZE (1..2) ) OF
OCTET STRING ( SIZE (1..20) ) OPTIONAL,
extensionContainer [4] IMPLICIT SEQUENCE {
privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE (1..10) ) OF
SEQUENCE {
extId MAP-EXTENSION .&extensionId ( {
'...' } ),
extType MAP-EXTENSION .&ExtensionType ( {
'...' { @extId } ) OPTIONAL} OPTIONAL,
pcs-Extensions [1] IMPLICIT SEQUENCE {
... } OPTIONAL,
... } OPTIONAL,
... }
}
RESULT
ss-InvocationNotificationRes SEQUENCE {

```

```

extensionContainer SEQUENCE {
  privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE (1..10) ) OF
    SEQUENCE {
      extId MAP-EXTENSION .&extensionId ( {
        '...'} ) ,
      extType MAP-EXTENSION .&ExtensionType ( {
        '...'} { @extId } ) OPTIONAL} OPTIONAL,
  pcs-Extensions [1] IMPLICIT SEQUENCE {
    ... } OPTIONAL,
  ... } OPTIONAL,
  ... }
ERRORS {
  -- dataMissing -- localValue : 35,
  -- unexpectedDataValue -- localValue : 36,
  -- unknownSubscriber -- localValue : 1}
 ::= localValue : 72

prepareGroupCall OPERATION
ARGUMENT
  prepareGroupCallArg SEQUENCE {
    teleservice OCTET STRING ( SIZE (1..5) ),
    asciiCallReference OCTET STRING ( SIZE (1..8) ),
    codec-info OCTET STRING ( SIZE (5..10) ),
    cipheringAlgorithm OCTET STRING ( SIZE (1) ),
    groupKeyNumber [0] IMPLICIT INTEGER ( 0..15 ) OPTIONAL,
    groupKey [1] IMPLICIT OCTET STRING ( SIZE (8) ) OPTIONAL,
    priority [2] IMPLICIT INTEGER ( 0..15 ) OPTIONAL,
    uplinkFree [3] IMPLICIT NULL OPTIONAL,
    extensionContainer [4] IMPLICIT SEQUENCE {
      privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE (1..10) ) OF
        SEQUENCE {
          extId MAP-EXTENSION .&extensionId ( {
            '...'} ) ,
          extType MAP-EXTENSION .&ExtensionType ( {
            '...'} { @extId } ) OPTIONAL} OPTIONAL,
      pcs-Extensions [1] IMPLICIT SEQUENCE {
        ... } OPTIONAL,
      ... } OPTIONAL,
      ... }
  RESULT
    prepareGroupCallRes SEQUENCE {
      groupCallNumber OCTET STRING ( SIZE (1..20) ) ( SIZE (1..9) ),
      extensionContainer SEQUENCE {
        privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE (1..10) ) OF
          SEQUENCE {
            extId MAP-EXTENSION .&extensionId ( {
              '...'} ) ,
            extType MAP-EXTENSION .&ExtensionType ( {
              '...'} { @extId } ) OPTIONAL} OPTIONAL,
          pcs-Extensions [1] IMPLICIT SEQUENCE {
            ... } OPTIONAL,
          ... } OPTIONAL,
          ... }
      ERRORS {
        -- systemFailure -- localValue : 34,
        -- noGroupCallNumberAvailable -- localValue : 50,
        -- unexpectedDataValue -- localValue : 36}
      ::= localValue : 39

sendGroupCallEndSignal OPERATION
ARGUMENT
  sendGroupCallEndSignalArg SEQUENCE {
    imsi OCTET STRING ( SIZE (3..8) ) OPTIONAL,
    extensionContainer SEQUENCE {
      privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE (1..10) ) OF
        SEQUENCE {
          extId MAP-EXTENSION .&extensionId ( {
            '...'} ) ,
          extType MAP-EXTENSION .&ExtensionType ( {
            '...'} { @extId } ) OPTIONAL} OPTIONAL,
      pcs-Extensions [1] IMPLICIT SEQUENCE {
        ... } OPTIONAL,
        ... } OPTIONAL,
        ... }
  RESULT

```

```

sendGroupCallEndSignalRes SEQUENCE {
  extensionContainer SEQUENCE {
    privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE (1..10) ) OF
      SEQUENCE {
        extId MAP-EXTENSION .&extensionId ( {
          '...'} ) ,
        extType MAP-EXTENSION .&ExtensionType ( {
          '...'} { @extId } ) OPTIONAL} OPTIONAL,
    pcs-Extensions [1] IMPLICIT SEQUENCE {
      ... } OPTIONAL,
    ... } OPTIONAL,
    ... }
  ::= localValue : 40
}

processGroupCallSignalling OPERATION
ARGUMENT
  processGroupCallSignallingArg SEQUENCE {
    uplinkRequest [0] IMPLICIT NULL OPTIONAL,
    uplinkReleaseIndication [1] IMPLICIT NULL OPTIONAL,
    releaseGroupCall [2] IMPLICIT NULL OPTIONAL,
    extensionContainer SEQUENCE {
      privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE (1..10) ) OF
        SEQUENCE {
          extId MAP-EXTENSION .&extensionId ( {
            '...'} ) ,
          extType MAP-EXTENSION .&ExtensionType ( {
            '...'} { @extId } ) OPTIONAL} OPTIONAL,
        pcs-Extensions [1] IMPLICIT SEQUENCE {
          ... } OPTIONAL,
        ... } OPTIONAL,
        ... }
      ::= localValue : 41
    }

forwardGroupCallSignalling OPERATION
ARGUMENT
  forwardGroupCallSignallingArg SEQUENCE {
    imsi OCTET STRING ( SIZE (3..8) ) OPTIONAL,
    uplinkRequestAck [0] IMPLICIT NULL OPTIONAL,
    uplinkReleaseIndication [1] IMPLICIT NULL OPTIONAL,
    uplinkRejectCommand [2] IMPLICIT NULL OPTIONAL,
    uplinkSeizedCommand [3] IMPLICIT NULL OPTIONAL,
    uplinkReleaseCommand [4] IMPLICIT NULL OPTIONAL,
    extensionContainer SEQUENCE {
      privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE (1..10) ) OF
        SEQUENCE {
          extId MAP-EXTENSION .&extensionId ( {
            '...'} ) ,
          extType MAP-EXTENSION .&ExtensionType ( {
            '...'} { @extId } ) OPTIONAL} OPTIONAL,
        pcs-Extensions [1] IMPLICIT SEQUENCE {
          ... } OPTIONAL,
        ... } OPTIONAL,
        ... }
      ::= localValue : 42
    }

updateGprsLocation OPERATION
ARGUMENT
  updateGprsLocationArg SEQUENCE {
    imsi OCTET STRING ( SIZE (3..8) ),
    sgsn-Number OCTET STRING ( SIZE (1..20) ) ( SIZE (1..9) ),
    sgsn-Address OCTET STRING ( SIZE (5..17) ),
    extensionContainer SEQUENCE {
      privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE (1..10) ) OF
        SEQUENCE {
          extId MAP-EXTENSION .&extensionId ( {
            '...'} ) ,
          extType MAP-EXTENSION .&ExtensionType ( {
            '...'} { @extId } ) OPTIONAL} OPTIONAL,
        pcs-Extensions [1] IMPLICIT SEQUENCE {
          ... } OPTIONAL,
        ... } OPTIONAL,
        ... ,
    sgsn-Capability [0] IMPLICIT SEQUENCE {
      solsaSupportIndicator NULL OPTIONAL,
      extensionContainer [1] IMPLICIT SEQUENCE {

```



```

privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE (1..10 ) ) OF
  SEQUENCE {
    extId      MAP-EXTENSION .&extensionId ( {
      '...' ) ,
    extType    MAP-EXTENSION .&ExtensionType ( {
      '...' { @extId } ) OPTIONAL} OPTIONAL,
    pcs-Extensions [1] IMPLICIT SEQUENCE {
      ... } OPTIONAL,
    ... } OPTIONAL,
    ... ,
superChargerSupportedInServingNetworkEntity [2] CHOICE {
  sendSubscriberData [0] IMPLICIT NULL,
  subscriberDataStored [1] IMPLICIT OCTET STRING ( SIZE (1..6 ) )} OPTIONAL,
gprsEnhancementsSupportIndicator [3] IMPLICIT NULL OPTIONAL,
supportedCamelPhases [4] IMPLICIT BIT STRING {
  phase1 ( 0 ),
  phase2 ( 1 ),
  phase3 ( 2 )} ( SIZE (1..16 ) ) OPTIONAL} OPTIONAL}

RESULT
updateGprsLocationRes SEQUENCE {
  hlr-Number OCTET STRING ( SIZE (1..20 ) ) ( SIZE (1..9 ) ),
  extensionContainer SEQUENCE {
    privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE (1..10 ) ) OF
      SEQUENCE {
        extId      MAP-EXTENSION .&extensionId ( {
          '...' ) ,
        extType    MAP-EXTENSION .&ExtensionType ( {
          '...' { @extId } ) OPTIONAL} OPTIONAL,
        pcs-Extensions [1] IMPLICIT SEQUENCE {
          ... } OPTIONAL,
        ... } OPTIONAL,
        ... }
    ... }
ERRORS {
  -- systemFailure -- localValue : 34,
  -- unexpectedDataValue -- localValue : 36,
  -- unknownSubscriber -- localValue : 1,
  -- roamingNotAllowed -- localValue : 8}
 ::= localValue : 23

sendRoutingInfoForGprs OPERATION
ARGUMENT
  sendRoutingInfoForGprsArg SEQUENCE {
    imsi [0] IMPLICIT OCTET STRING ( SIZE (3..8 ) ),
    ggsn-Address [1] IMPLICIT OCTET STRING ( SIZE (5..17 ) ) OPTIONAL,
    ggsn-Number [2] IMPLICIT OCTET STRING ( SIZE (1..20 ) ) ( SIZE (1..9 ) ),
    extensionContainer [3] IMPLICIT SEQUENCE {
      privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE (1..10 ) ) OF
        SEQUENCE {
          extId      MAP-EXTENSION .&extensionId ( {
            '...' ) ,
          extType    MAP-EXTENSION .&ExtensionType ( {
            '...' { @extId } ) OPTIONAL} OPTIONAL,
          pcs-Extensions [1] IMPLICIT SEQUENCE {
            ... } OPTIONAL,
            ... } OPTIONAL,
            ... }
    ... }
RESULT
  sendRoutingInfoForGprsRes SEQUENCE {
    sgsn-Address [0] IMPLICIT OCTET STRING ( SIZE (5..17 ) ),
    ggsn-Address [1] IMPLICIT OCTET STRING ( SIZE (5..17 ) ) OPTIONAL,
    mobileNotReachableReason [2] IMPLICIT INTEGER ( 0..255 ) OPTIONAL,
    extensionContainer [3] IMPLICIT SEQUENCE {
      privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE (1..10 ) ) OF
        SEQUENCE {
          extId      MAP-EXTENSION .&extensionId ( {
            '...' ) ,
          extType    MAP-EXTENSION .&ExtensionType ( {
            '...' { @extId } ) OPTIONAL} OPTIONAL,
          pcs-Extensions [1] IMPLICIT SEQUENCE {
            ... } OPTIONAL,
            ... } OPTIONAL,
            ... }
    ... }
ERRORS {
  -- absentSubscriber -- localValue : 27,
  -- systemFailure -- localValue : 34,

```

```

-- dataMissing -- localValue : 35,
-- unexpectedDataValue -- localValue : 36,
-- unknownSubscriber -- localValue : 1}
 ::= localValue : 24

failureReport OPERATION
ARGUMENT
  failureReportArg SEQUENCE {
    imsi [0] IMPLICIT OCTET STRING ( SIZE (3..8) ),
    ggsn-Number [1] IMPLICIT OCTET STRING ( SIZE (1..20) ) ( SIZE (1..9) ),
    ggsn-Address [2] IMPLICIT OCTET STRING ( SIZE (5..17) ) OPTIONAL,
    extensionContainer [3] IMPLICIT SEQUENCE {
      privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE (1..10) ) OF
        SEQUENCE {
          extId MAP-EXTENSION .&extensionId ( {
            '...'} ) ,
          extType MAP-EXTENSION .&ExtensionType ( {
            '...'} { @extId } ) OPTIONAL} OPTIONAL,
      pcs-Extensions [1] IMPLICIT SEQUENCE {
        ... } OPTIONAL,
      ... } OPTIONAL,
      ... }
RESULT
  failureReportRes SEQUENCE {
    ggsn-Address [0] IMPLICIT OCTET STRING ( SIZE (5..17) ) OPTIONAL,
    extensionContainer [1] IMPLICIT SEQUENCE {
      privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE (1..10) ) OF
        SEQUENCE {
          extId MAP-EXTENSION .&extensionId ( {
            '...'} ) ,
          extType MAP-EXTENSION .&ExtensionType ( {
            '...'} { @extId } ) OPTIONAL} OPTIONAL,
      pcs-Extensions [1] IMPLICIT SEQUENCE {
        ... } OPTIONAL,
      ... } OPTIONAL,
      ... }
ERRORS {
  -- systemFailure -- localValue : 34,
  -- dataMissing -- localValue : 35,
  -- unexpectedDataValue -- localValue : 36,
  -- unknownSubscriber -- localValue : 1}
 ::= localValue : 25

noteMsPresentForGprs OPERATION
ARGUMENT
  noteMsPresentForGprsArg SEQUENCE {
    imsi [0] IMPLICIT OCTET STRING ( SIZE (3..8) ),
    sgsn-Address [1] IMPLICIT OCTET STRING ( SIZE (5..17) ),
    ggsn-Address [2] IMPLICIT OCTET STRING ( SIZE (5..17) ) OPTIONAL,
    extensionContainer [3] IMPLICIT SEQUENCE {
      privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE (1..10) ) OF
        SEQUENCE {
          extId MAP-EXTENSION .&extensionId ( {
            '...'} ) ,
          extType MAP-EXTENSION .&ExtensionType ( {
            '...'} { @extId } ) OPTIONAL} OPTIONAL,
      pcs-Extensions [1] IMPLICIT SEQUENCE {
        ... } OPTIONAL,
      ... } OPTIONAL,
      ... }
RESULT
  noteMsPresentForGprsRes SEQUENCE {
    extensionContainer [0] IMPLICIT SEQUENCE {
      privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE (1..10) ) OF
        SEQUENCE {
          extId MAP-EXTENSION .&extensionId ( {
            '...'} ) ,
          extType MAP-EXTENSION .&ExtensionType ( {
            '...'} { @extId } ) OPTIONAL} OPTIONAL,
      pcs-Extensions [1] IMPLICIT SEQUENCE {
        ... } OPTIONAL,
      ... } OPTIONAL,
      ... }
ERRORS {
  -- systemFailure -- localValue : 34,

```

```

-- dataMissing -- localValue : 35,
-- unexpectedDataValue -- localValue : 36,
-- unknownSubscriber -- localValue : 1}
 ::= localValue : 26

provideSubscriberLocation OPERATION
ARGUMENT
  provideSubscriberLocation-Arg SEQUENCE {
    locationType SEQUENCE {
      locationEstimateType [0] IMPLICIT ENUMERATED {
        currentLocation (0 ),
        currentOrLastKnownLocation (1 ),
        initialLocation (2 ),
        ... },
      ... },
    mlc-Number OCTET STRING ( SIZE (1..20 ) ) ( SIZE (1..9 ) ),
    lcs-ClientID [0] IMPLICIT SEQUENCE {
      lcsClientType [0] IMPLICIT ENUMERATED {
        emergencyServices (0 ),
        valueAddedServices (1 ),
        plmnOperatorServices (2 ),
        lawfulInterceptServices (3 ),
        ... },
      lcsClientExternalID [1] IMPLICIT SEQUENCE {
        externalAddress [0] IMPLICIT OCTET STRING ( SIZE (1..20 ) ) OPTIONAL,
        extensionContainer [1] IMPLICIT SEQUENCE {
          privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE (1..10 ) ) OF
            SEQUENCE {
              extId MAP-EXTENSION .&extensionId ( {
                '... } ) ,
              extType MAP-EXTENSION .&ExtensionType ( {
                '... } { @extId } ) OPTIONAL} OPTIONAL,
            pcs-Extensions [1] IMPLICIT SEQUENCE {
              ... } OPTIONAL,
              ... } OPTIONAL,
              ... } OPTIONAL,
            lcsClientDialedByMS [2] IMPLICIT OCTET STRING ( SIZE (1..20 ) ) OPTIONAL,
            lcsClientInternalID [3] IMPLICIT ENUMERATED {
              broadcastService (0 ),
              o-andM-HPLMN (1 ),
              o-andM-VPLMN (2 ),
              anonymousLocation (3 ),
              targetMSsubscribedService (4 ),
              ... } OPTIONAL,
            lcsClientName [4] IMPLICIT SEQUENCE {
              dataCodingScheme [0] IMPLICIT OCTET STRING ( SIZE (1 ) ),
              nameString [2] IMPLICIT OCTET STRING ( SIZE (1..160 ) ) ( SIZE (1..63 ) ),
              ... } OPTIONAL,
              ... } OPTIONAL,
            privacyOverride [1] IMPLICIT NULL OPTIONAL,
            imsi [2] IMPLICIT OCTET STRING ( SIZE (3..8 ) ) OPTIONAL,
            msisdn [3] IMPLICIT OCTET STRING ( SIZE (1..20 ) ) ( SIZE (1..9 ) ) OPTIONAL,
            lmsi [4] IMPLICIT OCTET STRING ( SIZE (4 ) ) OPTIONAL,
            imei [5] IMPLICIT OCTET STRING ( SIZE (8 ) ) OPTIONAL,
            lcs-Priority [6] IMPLICIT OCTET STRING ( SIZE (1 ) ) OPTIONAL,
            lcs-QoS [7] IMPLICIT SEQUENCE {
              horizontal-accuracy [0] IMPLICIT OCTET STRING ( SIZE (1 ) ) OPTIONAL,
              verticalCoordinateRequest [1] IMPLICIT NULL OPTIONAL,
              vertical-accuracy [2] IMPLICIT OCTET STRING ( SIZE (1 ) ) OPTIONAL,
              responseTime [3] IMPLICIT SEQUENCE {
                responseTimeCategory ENUMERATED {
                  lowdelay (0 ),
                  delaytolerant (1 ),
                  ... },
                ... } OPTIONAL,
              extensionContainer [4] IMPLICIT SEQUENCE {
                privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE (1..10 ) ) OF
                  SEQUENCE {
                    extId MAP-EXTENSION .&extensionId ( {
                      '... } ) ,
                    extType MAP-EXTENSION .&ExtensionType ( {
                      '... } { @extId } ) OPTIONAL} OPTIONAL,
                  pcs-Extensions [1] IMPLICIT SEQUENCE {
                    ... } OPTIONAL,
                    ... } OPTIONAL,
                    ... } OPTIONAL,
                  extensionContainer [8] IMPLICIT SEQUENCE {
                    privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE (1..10 ) ) OF
                      SEQUENCE {

```

```

        extId      MAP-EXTENSION .&extensionId ( {
            '...'} ) ,
        extType    MAP-EXTENSION .&ExtensionType ( {
            '...'} { @extId } ) OPTIONAL} OPTIONAL,
        pcs-Extensions [1] IMPLICIT SEQUENCE {
            ... } OPTIONAL,
        ... } OPTIONAL,
    ... }
RESULT
provideSubscriberLocation-Res SEQUENCE {
    locationEstimate    OCTET STRING ( SIZE (1..20) ),
    ageOfLocationEstimate [0] IMPLICIT INTEGER ( 0..32767 ) OPTIONAL,
    extensionContainer [1] IMPLICIT SEQUENCE {
        privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE (1..10) ) OF
            SEQUENCE {
                extId      MAP-EXTENSION .&extensionId ( {
                    '...'} ) ,
                extType    MAP-EXTENSION .&ExtensionType ( {
                    '...'} { @extId } ) OPTIONAL} OPTIONAL,
                pcs-Extensions [1] IMPLICIT SEQUENCE {
                    ... } OPTIONAL,
                ... } 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,
-- unauthorizedRequestingNetwork -- localValue : 52,
-- unauthorizedLCSClient -- localValue : 53,
-- positionMethodFailure -- localValue : 54}
 ::= localValue : 83

sendRoutingInfoForLCS OPERATION
ARGUMENT
routingInfoForLCS-Arg SEQUENCE {
    mlcNumber [0] IMPLICIT OCTET STRING ( SIZE (1..20) ) ( SIZE (1..9) ),
    targetMS [1] CHOICE {
        imsi [0] IMPLICIT OCTET STRING ( SIZE (3..8) ),
        msisdn [1] IMPLICIT OCTET STRING ( SIZE (1..20) ) ( SIZE (1..9) )},
    extensionContainer [2] IMPLICIT SEQUENCE {
        privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE (1..10) ) OF
            SEQUENCE {
                extId      MAP-EXTENSION .&extensionId ( {
                    '...'} ) ,
                extType    MAP-EXTENSION .&ExtensionType ( {
                    '...'} { @extId } ) OPTIONAL} OPTIONAL,
                pcs-Extensions [1] IMPLICIT SEQUENCE {
                    ... } OPTIONAL,
                ... } OPTIONAL,
            ... }
RESULT
routingInfoForLCS-Res SEQUENCE {
    targetMS [0] CHOICE {
        imsi [0] IMPLICIT OCTET STRING ( SIZE (3..8) ),
        msisdn [1] IMPLICIT OCTET STRING ( SIZE (1..20) ) ( SIZE (1..9) )},
    lcsLocationInfo [1] IMPLICIT SEQUENCE {
        msc-Number OCTET STRING ( SIZE (1..20) ) ( SIZE (1..9) ),
        lmsi [0] IMPLICIT OCTET STRING ( SIZE (4) ) OPTIONAL,
        extensionContainer [1] IMPLICIT SEQUENCE {
            privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE (1..10) ) OF
                SEQUENCE {
                    extId      MAP-EXTENSION .&extensionId ( {
                        '...'} ) ,
                    extType    MAP-EXTENSION .&ExtensionType ( {
                        '...'} { @extId } ) OPTIONAL} OPTIONAL,
                    pcs-Extensions [1] IMPLICIT SEQUENCE {
                        ... } OPTIONAL,
                        ... } OPTIONAL,
                    ... },
            extensionContainer [2] IMPLICIT SEQUENCE {

```

```

privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE (1..10 ) ) OF
  SEQUENCE {
    extId      MAP-EXTENSION .&extensionId ( {
      '...' ) ,
    extType    MAP-EXTENSION .&ExtensionType ( {
      '...' { @extId } ) OPTIONAL} OPTIONAL,
    pcs-Extensions [1] IMPLICIT SEQUENCE {
      ... } OPTIONAL,
    ... } OPTIONAL,
  ... }
ERRORS {
  -- systemFailure -- localValue : 34,
  -- dataMissing -- localValue : 35,
  -- unexpectedDataValue -- localValue : 36,
  -- facilityNotSupported -- localValue : 21,
  -- unknownSubscriber -- localValue : 1,
  -- absentSubscriber -- localValue : 27,
  -- unauthorizedRequestingNetwork -- localValue : 52}
 ::= localValue : 85

subscriberLocationReport OPERATION
ARGUMENT
  subscriberLocationReport-Arg SEQUENCE {
    lcs-Event      ENUMERATED {
      emergencyCallOrigination (0 ),
      emergencyCallRelease (1 ),
      mo-lr (2 ),
      ... },
    lcs-ClientID   SEQUENCE {
      lcsClientType [0] IMPLICIT ENUMERATED {
        emergencyServices (0 ),
        valueAddedServices (1 ),
        plmnOperatorServices (2 ),
        lawfulInterceptServices (3 ),
        ... },
      lcsClientExternalID [1] IMPLICIT SEQUENCE {
        externalAddress [0] IMPLICIT OCTET STRING ( SIZE (1..20 ) ) OPTIONAL,
        extensionContainer [1] IMPLICIT SEQUENCE {
          privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE (1..10 ) ) OF
            SEQUENCE {
              extId      MAP-EXTENSION .&extensionId ( {
                '...' ) ,
              extType    MAP-EXTENSION .&ExtensionType ( {
                '...' { @extId } ) OPTIONAL} OPTIONAL,
            pcs-Extensions [1] IMPLICIT SEQUENCE {
              ... } OPTIONAL,
            ... } OPTIONAL,
            ... } OPTIONAL,
          lcsClientDialedByMS [2] IMPLICIT OCTET STRING ( SIZE (1..20 ) ) OPTIONAL,
          lcsClientInternalID [3] IMPLICIT ENUMERATED {
            broadcastService (0 ),
            o-andM-HPLMN (1 ),
            o-andM-VPLMN (2 ),
            anonymousLocation (3 ),
            targetMSsubscribedService (4 ),
            ... } OPTIONAL,
          lcsClientName [4] IMPLICIT SEQUENCE {
            dataCodingScheme [0] IMPLICIT OCTET STRING ( SIZE (1 ) ),
            nameString [2] IMPLICIT OCTET STRING ( SIZE (1..160 ) ) ( SIZE (1..63 ) ),
            ... } OPTIONAL,
          ... },
        lcsLocationInfo SEQUENCE {
          msc-Number OCTET STRING ( SIZE (1..20 ) ) ( SIZE (1..9 ) ),
          lmsi [0] IMPLICIT OCTET STRING ( SIZE (4 ) ) OPTIONAL,
          extensionContainer [1] IMPLICIT SEQUENCE {
            privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE (1..10 ) ) OF
              SEQUENCE {
                extId      MAP-EXTENSION .&extensionId ( {
                  '...' ) ,
                extType    MAP-EXTENSION .&ExtensionType ( {
                  '...' { @extId } ) OPTIONAL} OPTIONAL,
                pcs-Extensions [1] IMPLICIT SEQUENCE {
                  ... } OPTIONAL,
                ... } OPTIONAL,
                ... },
            msisdn [0] IMPLICIT OCTET STRING ( SIZE (1..20 ) ) ( SIZE (1..9 ) )
          } OPTIONAL,

```

```

imsi [1] IMPLICIT OCTET STRING ( SIZE ( 3..8 ) ) OPTIONAL,
imei [2] IMPLICIT OCTET STRING ( SIZE ( 8 ) ) OPTIONAL,
na-ESRD [3] IMPLICIT OCTET STRING ( SIZE ( 1..20 ) ) ( SIZE ( 1..9 ) )
OPTIONAL,
na-ESRK [4] IMPLICIT OCTET STRING ( SIZE ( 1..20 ) ) ( SIZE ( 1..9 ) )
OPTIONAL,
locationEstimate [5] IMPLICIT OCTET STRING ( SIZE ( 1..20 ) ) OPTIONAL,
ageOfLocationEstimate [6] IMPLICIT INTEGER ( 0..32767 ) OPTIONAL,
extensionContainer [7] IMPLICIT SEQUENCE {
  privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE ( 1..10 ) ) OF
    SEQUENCE {
      extId MAP-EXTENSION .&extensionId ( {
        '...' ) ,
      extType MAP-EXTENSION .&ExtensionType ( {
        '...' { @extId } ) OPTIONAL} OPTIONAL,
      pcs-Extensions [1] IMPLICIT SEQUENCE {
        ... } OPTIONAL,
      ... } OPTIONAL,
      ... }
RESULT
subscriberLocationReport-Res SEQUENCE {
  extensionContainer SEQUENCE {
    privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE ( 1..10 ) ) OF
      SEQUENCE {
        extId MAP-EXTENSION .&extensionId ( {
          '...' ) ,
        extType MAP-EXTENSION .&ExtensionType ( {
          '...' { @extId } ) OPTIONAL} OPTIONAL,
        pcs-Extensions [1] IMPLICIT SEQUENCE {
          ... } OPTIONAL,
        ... } OPTIONAL,
        ... }
ERRORS {
  -- systemFailure -- localValue : 34,
  -- dataMissing -- localValue : 35,
  -- resourceLimitation -- localValue : 51,
  -- unexpectedDataValue -- localValue : 36,
  -- unknownSubscriber -- localValue : 1,
  -- unauthorizedRequestingNetwork -- localValue : 52,
  -- unknownOrUnreachableLCSClient -- localValue : 58}
 ::= localValue : 86

noteMM-Event OPERATION
ARGUMENT
noteMM-EventArg SEQUENCE {
  serviceKey INTEGER ( 0..2147483647 ),
  eventMet [0] IMPLICIT OCTET STRING ( SIZE ( 1 ) ),
  imsi [1] IMPLICIT OCTET STRING ( SIZE ( 3..8 ) ),
  msisdn [2] IMPLICIT OCTET STRING ( SIZE ( 1..20 ) ) ( SIZE ( 1..9 ) ),
  locationInformation [3] IMPLICIT SEQUENCE {
    ageOfLocationInformation INTEGER ( 0..32767 ) OPTIONAL,
    geographicalInformation [0] IMPLICIT OCTET STRING ( SIZE ( 8 ) ) OPTIONAL,
    vlr-number [1] IMPLICIT OCTET STRING ( SIZE ( 1..20 ) ) ( SIZE ( 1..9 ) )
OPTIONAL,
  locationNumber [2] IMPLICIT OCTET STRING ( SIZE ( 2..10 ) ) OPTIONAL,
  cellIdOrLAI [3] CHOICE {
    cellIdFixedLength [0] IMPLICIT OCTET STRING ( SIZE ( 7 ) ),
    laiFixedLength [1] IMPLICIT OCTET STRING ( SIZE ( 5 ) ) } OPTIONAL,
  extensionContainer [4] IMPLICIT SEQUENCE {
    privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE ( 1..10 ) ) OF
      SEQUENCE {
        extId MAP-EXTENSION .&extensionId ( {
          '...' ) ,
        extType MAP-EXTENSION .&ExtensionType ( {
          '...' { @extId } ) OPTIONAL} OPTIONAL,
        pcs-Extensions [1] IMPLICIT SEQUENCE {
          ... } OPTIONAL,
        ... } OPTIONAL,
        ... ,
        selectedLSA-Id [5] IMPLICIT OCTET STRING ( SIZE ( 3 ) ) OPTIONAL,
        msc-Number [6] IMPLICIT OCTET STRING ( SIZE ( 1..20 ) ) ( SIZE ( 1..9 ) )
OPTIONAL,
        geodeticInformation [7] IMPLICIT OCTET STRING ( SIZE ( 10 ) ) OPTIONAL} OPTIONAL,
  lsaIdentity [4] IMPLICIT OCTET STRING ( SIZE ( 3 ) ) OPTIONAL,
  supportedCAMELPhases [5] IMPLICIT BIT STRING {
    phase1 ( 0 ),
    phase2 ( 1 ),

```

```

    phase3 ( 2 ) } ( SIZE ( 1..16 ) ) OPTIONAL,
    extensionContainer [6] IMPLICIT SEQUENCE {
      privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE ( 1..10 ) ) OF
        SEQUENCE {
          extId MAP-EXTENSION .&extensionId ( {
            '... } ) ,
          extType MAP-EXTENSION .&ExtensionType ( {
            '... } { @extId } ) OPTIONAL} OPTIONAL,
      pcs-Extensions [1] IMPLICIT SEQUENCE {
        ... } OPTIONAL,
        ... } OPTIONAL,
        ... }
  RESULT
    noteMM-EventRes SEQUENCE {
      extensionContainer SEQUENCE {
        privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE ( 1..10 ) ) OF
          SEQUENCE {
            extId MAP-EXTENSION .&extensionId ( {
              '... } ) ,
            extType MAP-EXTENSION .&ExtensionType ( {
              '... } { @extId } ) OPTIONAL} OPTIONAL,
          pcs-Extensions [1] IMPLICIT SEQUENCE {
            ... } OPTIONAL,
            ... } OPTIONAL,
            ... }
  ERRORS {
    -- dataMissing -- localValue : 35,
    -- unexpectedDataValue -- localValue : 36,
    -- unknownSubscriber -- localValue : 1,
    -- mm-EventNotSupported -- localValue : 59}
 ::= localValue : 89

systemFailure ERROR
PARAMETER
  systemFailureParam CHOICE {
    networkResource ENUMERATED {
      plmn ( 0 ),
      hlr ( 1 ),
      vlr ( 2 ),
      pvlr ( 3 ),
      controllingMSC ( 4 ),
      vmsc ( 5 ),
      eir ( 6 ),
      rss ( 7 )},
    extensibleSystemFailureParam SEQUENCE {
      networkResource ENUMERATED {
        plmn ( 0 ),
        hlr ( 1 ),
        vlr ( 2 ),
        pvlr ( 3 ),
        controllingMSC ( 4 ),
        vmsc ( 5 ),
        eir ( 6 ),
        rss ( 7 )} OPTIONAL,
      extensionContainer SEQUENCE {
        privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE ( 1..10 ) ) OF
          SEQUENCE {
            extId MAP-EXTENSION .&extensionId ( {
              '... } ) ,
            extType MAP-EXTENSION .&ExtensionType ( {
              '... } { @extId } ) OPTIONAL} OPTIONAL,
          pcs-Extensions [1] IMPLICIT SEQUENCE {
            ... } OPTIONAL,
            ... } OPTIONAL,
            ... }
  ::= localValue : 34

dataMissing ERROR
PARAMETER
  dataMissingParam SEQUENCE {
    extensionContainer SEQUENCE {
      privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE ( 1..10 ) ) OF
        SEQUENCE {
          extId MAP-EXTENSION .&extensionId ( {
            '... } ) ,
          extType MAP-EXTENSION .&ExtensionType ( {

```

```

        '... } { @extId } ) OPTIONAL} OPTIONAL,
    pcs-Extensions [1] IMPLICIT SEQUENCE {
        ... } OPTIONAL,
        ... } OPTIONAL,
    ... }
 ::= localValue : 35

unexpectedDataValue ERROR
PARAMETER
    unexpectedDataParam SEQUENCE {
        extensionContainer SEQUENCE {
            privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE (1..10) ) OF
                SEQUENCE {
                    extId MAP-EXTENSION .&extensionId ( {
                        '... } ) ,
                    extType MAP-EXTENSION .&ExtensionType ( {
                        '... } { @extId } ) OPTIONAL} OPTIONAL,
            pcs-Extensions [1] IMPLICIT SEQUENCE {
                ... } OPTIONAL,
                ... } OPTIONAL,
            ... }
 ::= localValue : 36

facilityNotSupported ERROR
PARAMETER
    facilityNotSupParam SEQUENCE {
        extensionContainer SEQUENCE {
            privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE (1..10) ) OF
                SEQUENCE {
                    extId MAP-EXTENSION .&extensionId ( {
                        '... } ) ,
                    extType MAP-EXTENSION .&ExtensionType ( {
                        '... } { @extId } ) OPTIONAL} OPTIONAL,
            pcs-Extensions [1] IMPLICIT SEQUENCE {
                ... } OPTIONAL,
                ... } OPTIONAL,
            ... }
 ::= localValue : 21

incompatibleTerminal ERROR
PARAMETER
    incompatibleTerminalParam SEQUENCE {
        extensionContainer SEQUENCE {
            privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE (1..10) ) OF
                SEQUENCE {
                    extId MAP-EXTENSION .&extensionId ( {
                        '... } ) ,
                    extType MAP-EXTENSION .&ExtensionType ( {
                        '... } { @extId } ) OPTIONAL} OPTIONAL,
            pcs-Extensions [1] IMPLICIT SEQUENCE {
                ... } OPTIONAL,
                ... } OPTIONAL,
            ... }
 ::= localValue : 28

resourceLimitation ERROR
PARAMETER
    resourceLimitationParam SEQUENCE {
        extensionContainer SEQUENCE {
            privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE (1..10) ) OF
                SEQUENCE {
                    extId MAP-EXTENSION .&extensionId ( {
                        '... } ) ,
                    extType MAP-EXTENSION .&ExtensionType ( {
                        '... } { @extId } ) OPTIONAL} OPTIONAL,
            pcs-Extensions [1] IMPLICIT SEQUENCE {
                ... } OPTIONAL,
                ... } OPTIONAL,
            ... }
 ::= localValue : 51

unknownSubscriber ERROR
PARAMETER
    unknownSubscriberParam SEQUENCE {

```



```

extensionContainer          SEQUENCE {
  privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE (1..10) ) OF
    SEQUENCE {
      extId          MAP-EXTENSION .&extensionId ( {
        '...'} ) ,
      extType        MAP-EXTENSION .&ExtensionType ( {
        '...'} { @extId } ) OPTIONAL} OPTIONAL,
  pcs-Extensions [1] IMPLICIT SEQUENCE {
    ... } OPTIONAL,
    ... } OPTIONAL,
  ... ,
  unknownSubscriberDiagnostic  ENUMERATED {
    imsiUnknown          ( 0 ),
    gprsSubscriptionUnknown ( 1 ),
    ... } OPTIONAL}
 ::= localValue : 1

numberChanged ERROR
PARAMETER
  numberChangedParam SEQUENCE {
    extensionContainer SEQUENCE {
      privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE (1..10) ) OF
        SEQUENCE {
          extId          MAP-EXTENSION .&extensionId ( {
            '...'} ) ,
          extType        MAP-EXTENSION .&ExtensionType ( {
            '...'} { @extId } ) OPTIONAL} OPTIONAL,
        pcs-Extensions [1] IMPLICIT SEQUENCE {
          ... } OPTIONAL,
          ... } OPTIONAL,
        ... }
    ... }
 ::= localValue : 44

unknownMSC ERROR
 ::= localValue : 3

unidentifiedSubscriber ERROR
PARAMETER
  unidentifiedSubParam SEQUENCE {
    extensionContainer SEQUENCE {
      privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE (1..10) ) OF
        SEQUENCE {
          extId          MAP-EXTENSION .&extensionId ( {
            '...'} ) ,
          extType        MAP-EXTENSION .&ExtensionType ( {
            '...'} { @extId } ) OPTIONAL} OPTIONAL,
        pcs-Extensions [1] IMPLICIT SEQUENCE {
          ... } OPTIONAL,
          ... } OPTIONAL,
        ... }
    ... }
 ::= localValue : 5

unknownEquipment ERROR
 ::= localValue : 7

roamingNotAllowed ERROR
PARAMETER
  roamingNotAllowedParam SEQUENCE {
    roamingNotAllowedCause  ENUMERATED {
      plmnRoamingNotAllowed ( 0 ),
      operatorDeterminedBarring ( 3 )},
    extensionContainer SEQUENCE {
      privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE (1..10) ) OF
        SEQUENCE {
          extId          MAP-EXTENSION .&extensionId ( {
            '...'} ) ,
          extType        MAP-EXTENSION .&ExtensionType ( {
            '...'} { @extId } ) OPTIONAL} OPTIONAL,
        pcs-Extensions [1] IMPLICIT SEQUENCE {
          ... } OPTIONAL,
          ... } OPTIONAL,
        ... }
    ... }
 ::= localValue : 8

illegalSubscriber ERROR

```

```

PARAMETER
  illegalSubscriberParam SEQUENCE {
    extensionContainer SEQUENCE {
      privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE (1..10) ) OF
        SEQUENCE {
          extId MAP-EXTENSION .&extensionId ( {
            '...' ) ,
          extType MAP-EXTENSION .&ExtensionType ( {
            '...' { @extId } ) OPTIONAL} OPTIONAL,
          pcs-Extensions [1] IMPLICIT SEQUENCE {
            ... } OPTIONAL,
            ... } OPTIONAL,
            ... }
    ::= localValue : 9

illegalEquipment ERROR
PARAMETER
  illegalEquipmentParam SEQUENCE {
    extensionContainer SEQUENCE {
      privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE (1..10) ) OF
        SEQUENCE {
          extId MAP-EXTENSION .&extensionId ( {
            '...' ) ,
          extType MAP-EXTENSION .&ExtensionType ( {
            '...' { @extId } ) OPTIONAL} OPTIONAL,
          pcs-Extensions [1] IMPLICIT SEQUENCE {
            ... } OPTIONAL,
            ... } OPTIONAL,
            ... }
    ::= localValue : 12

bearerServiceNotProvisioned ERROR
PARAMETER
  bearerServNotProvParam SEQUENCE {
    extensionContainer SEQUENCE {
      privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE (1..10) ) OF
        SEQUENCE {
          extId MAP-EXTENSION .&extensionId ( {
            '...' ) ,
          extType MAP-EXTENSION .&ExtensionType ( {
            '...' { @extId } ) OPTIONAL} OPTIONAL,
          pcs-Extensions [1] IMPLICIT SEQUENCE {
            ... } OPTIONAL,
            ... } OPTIONAL,
            ... }
    ::= localValue : 10

teleserviceNotProvisioned ERROR
PARAMETER
  teleservNotProvParam SEQUENCE {
    extensionContainer SEQUENCE {
      privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE (1..10) ) OF
        SEQUENCE {
          extId MAP-EXTENSION .&extensionId ( {
            '...' ) ,
          extType MAP-EXTENSION .&ExtensionType ( {
            '...' { @extId } ) OPTIONAL} OPTIONAL,
          pcs-Extensions [1] IMPLICIT SEQUENCE {
            ... } OPTIONAL,
            ... } OPTIONAL,
            ... }
    ::= localValue : 11

noHandoverNumberAvailable ERROR
::= localValue : 25

subsequentHandoverFailure ERROR
::= localValue : 26

tracingBufferFull ERROR
PARAMETER
  tracingBufferFullParam SEQUENCE {
    extensionContainer SEQUENCE {
      privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE (1..10) ) OF
        SEQUENCE {

```

```

        extId      MAP-EXTENSION .&extensionId ( {
            '... } ) ,
        extType    MAP-EXTENSION .&ExtensionType ( {
            '... } { @extId } ) OPTIONAL} OPTIONAL,
        pcs-Extensions [1] IMPLICIT SEQUENCE {
            ... } OPTIONAL,
            ... } OPTIONAL,
            ... }
 ::= localValue : 40

noRoamingNumberAvailable ERROR
PARAMETER
    noRoamingNbParam SEQUENCE {
        extensionContainer SEQUENCE {
            privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE (1..10) ) OF
                SEQUENCE {
                    extId      MAP-EXTENSION .&extensionId ( {
                        '... } ) ,
                    extType    MAP-EXTENSION .&ExtensionType ( {
                        '... } { @extId } ) OPTIONAL} OPTIONAL,
                    pcs-Extensions [1] IMPLICIT SEQUENCE {
                        ... } OPTIONAL,
                        ... } OPTIONAL,
                        ... }
                ::= localValue : 39

absentSubscriber ERROR
PARAMETER
    absentSubscriberParam SEQUENCE {
        extensionContainer SEQUENCE {
            privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE (1..10) ) OF
                SEQUENCE {
                    extId      MAP-EXTENSION .&extensionId ( {
                        '... } ) ,
                    extType    MAP-EXTENSION .&ExtensionType ( {
                        '... } { @extId } ) OPTIONAL} OPTIONAL,
                    pcs-Extensions [1] IMPLICIT SEQUENCE {
                        ... } OPTIONAL,
                        ... } OPTIONAL,
                        ... }
                ,
            absentSubscriberReason [0] IMPLICIT ENUMERATED {
                imsiDetach (0 ),
                restrictedArea (1 ),
                noPageResponse (2 ),
                ... ,
                purgedMS (3 )} OPTIONAL}
        ::= localValue : 27

busySubscriber ERROR
PARAMETER
    busySubscriberParam SEQUENCE {
        extensionContainer SEQUENCE {
            privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE (1..10) ) OF
                SEQUENCE {
                    extId      MAP-EXTENSION .&extensionId ( {
                        '... } ) ,
                    extType    MAP-EXTENSION .&ExtensionType ( {
                        '... } { @extId } ) OPTIONAL} OPTIONAL,
                    pcs-Extensions [1] IMPLICIT SEQUENCE {
                        ... } OPTIONAL,
                        ... } OPTIONAL,
                        ... }
                ,
            ccbs-Possible [0] IMPLICIT NULL OPTIONAL,
            ccbs-Busy [1] IMPLICIT NULL OPTIONAL}
        ::= localValue : 45

noSubscriberReply ERROR
PARAMETER
    noSubscriberReplyParam SEQUENCE {
        extensionContainer SEQUENCE {
            privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE (1..10) ) OF
                SEQUENCE {
                    extId      MAP-EXTENSION .&extensionId ( {
                        '... } ) ,

```

```

        extType      MAP-EXTENSION .&ExtensionType ( {
            '... } { @extId } ) OPTIONAL} OPTIONAL,
        pcs-Extensions [1] IMPLICIT SEQUENCE {
            ... } OPTIONAL,
            ... } OPTIONAL,
        ... }
 ::= localValue : 46

callBarred ERROR
PARAMETER
    callBarredParam CHOICE {
        callBarringCause      ENUMERATED {
            barringServiceActive (0 ),
            operatorBarring      (1 )},
        extensibleCallBarredParam SEQUENCE {
            callBarringCause      ENUMERATED {
                barringServiceActive (0 ),
                operatorBarring      (1 )} OPTIONAL,
            extensionContainer      SEQUENCE {
                privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE (1..10 ) ) OF
                SEQUENCE {
                    extId      MAP-EXTENSION .&extensionId ( {
                        '... } ) ,
                    extType      MAP-EXTENSION .&ExtensionType ( {
                        '... } { @extId } ) OPTIONAL} OPTIONAL,
                pcs-Extensions [1] IMPLICIT SEQUENCE {
                    ... } OPTIONAL,
                    ... } OPTIONAL,
                ... }
            unauthorisedMessageOriginator [1] IMPLICIT NULL OPTIONAL}}
 ::= localValue : 13

forwardingFailed ERROR
PARAMETER
    forwardingFailedParam SEQUENCE {
        extensionContainer SEQUENCE {
            privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE (1..10 ) ) OF
            SEQUENCE {
                extId      MAP-EXTENSION .&extensionId ( {
                    '... } ) ,
                extType      MAP-EXTENSION .&ExtensionType ( {
                    '... } { @extId } ) OPTIONAL} OPTIONAL,
            pcs-Extensions [1] IMPLICIT SEQUENCE {
                ... } OPTIONAL,
                ... } OPTIONAL,
            ... }
        ... }
 ::= localValue : 47

or-NotAllowed ERROR
PARAMETER
    or-NotAllowedParam SEQUENCE {
        extensionContainer SEQUENCE {
            privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE (1..10 ) ) OF
            SEQUENCE {
                extId      MAP-EXTENSION .&extensionId ( {
                    '... } ) ,
                extType      MAP-EXTENSION .&ExtensionType ( {
                    '... } { @extId } ) OPTIONAL} OPTIONAL,
            pcs-Extensions [1] IMPLICIT SEQUENCE {
                ... } OPTIONAL,
                ... } OPTIONAL,
            ... }
        ... }
 ::= localValue : 48

forwardingViolation ERROR
PARAMETER
    forwardingViolationParam SEQUENCE {
        extensionContainer SEQUENCE {
            privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE (1..10 ) ) OF
            SEQUENCE {
                extId      MAP-EXTENSION .&extensionId ( {
                    '... } ) ,
                extType      MAP-EXTENSION .&ExtensionType ( {
                    '... } { @extId } ) OPTIONAL} OPTIONAL,

```

```

        pcs-Extensions          [1] IMPLICIT SEQUENCE {
            ... } OPTIONAL,
            ... } OPTIONAL,
        ... }
 ::= localValue : 14

cug-Reject ERROR
PARAMETER
    cug-RejectParam SEQUENCE {
        cug-RejectCause          ENUMERATED {
            incomingCallsBarredWithinCUG          ( 0 ),
            subscriberNotMemberOfCUG              ( 1 ),
            requestedBasicServiceViolatesCUG-Constraints ( 5 ),
            calledPartySS-InteractionViolation    ( 7 ) } OPTIONAL,
        extensionContainer      SEQUENCE {
            privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE ( 1..10 ) ) OF
                SEQUENCE {
                    extId          MAP-EXTENSION .&extensionId ( {
                        '... } ) ,
                    extType       MAP-EXTENSION .&ExtensionType ( {
                        '... } { @extId } ) OPTIONAL } OPTIONAL,
                    pcs-Extensions [1] IMPLICIT SEQUENCE {
                        ... } OPTIONAL,
                        ... } OPTIONAL,
                    ... }
                ::= localValue : 15

ati-NotAllowed ERROR
PARAMETER
    ati-NotAllowedParam SEQUENCE {
        extensionContainer SEQUENCE {
            privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE ( 1..10 ) ) OF
                SEQUENCE {
                    extId          MAP-EXTENSION .&extensionId ( {
                        '... } ) ,
                    extType       MAP-EXTENSION .&ExtensionType ( {
                        '... } { @extId } ) OPTIONAL } OPTIONAL,
                    pcs-Extensions [1] IMPLICIT SEQUENCE {
                        ... } OPTIONAL,
                        ... } OPTIONAL,
                    ... }
                ::= localValue : 49

atsi-NotAllowed ERROR
PARAMETER
    atsi-NotAllowedParam SEQUENCE {
        extensionContainer SEQUENCE {
            privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE ( 1..10 ) ) OF
                SEQUENCE {
                    extId          MAP-EXTENSION .&extensionId ( {
                        '... } ) ,
                    extType       MAP-EXTENSION .&ExtensionType ( {
                        '... } { @extId } ) OPTIONAL } OPTIONAL,
                    pcs-Extensions [1] IMPLICIT SEQUENCE {
                        ... } OPTIONAL,
                        ... } OPTIONAL,
                    ... }
                ::= localValue : 60

atm-NotAllowed ERROR
PARAMETER
    atm-NotAllowedParam SEQUENCE {
        extensionContainer SEQUENCE {
            privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE ( 1..10 ) ) OF
                SEQUENCE {
                    extId          MAP-EXTENSION .&extensionId ( {
                        '... } ) ,
                    extType       MAP-EXTENSION .&ExtensionType ( {
                        '... } { @extId } ) OPTIONAL } OPTIONAL,
                    pcs-Extensions [1] IMPLICIT SEQUENCE {
                        ... } OPTIONAL,
                        ... } OPTIONAL,
                    ... }
                ::= localValue : 61

```

```

informationNotAvailable ERROR
PARAMETER
  informationNotAvailableParam SEQUENCE {
    extensionContainer SEQUENCE {
      privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE (1..10) ) OF
        SEQUENCE {
          extId MAP-EXTENSION .&extensionId ( {
            '...' ) ,
          extType MAP-EXTENSION .&ExtensionType ( {
            '...' { @extId } ) OPTIONAL} OPTIONAL,
        pcs-Extensions [1] IMPLICIT SEQUENCE {
          ... } OPTIONAL,
          ... } OPTIONAL,
          ... }
    }
  ::= localValue : 62

noGroupCallNumberAvailable ERROR
PARAMETER
  noGroupCallNbParam SEQUENCE {
    extensionContainer SEQUENCE {
      privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE (1..10) ) OF
        SEQUENCE {
          extId MAP-EXTENSION .&extensionId ( {
            '...' ) ,
          extType MAP-EXTENSION .&ExtensionType ( {
            '...' { @extId } ) OPTIONAL} OPTIONAL,
        pcs-Extensions [1] IMPLICIT SEQUENCE {
          ... } OPTIONAL,
          ... } OPTIONAL,
          ... }
    }
  ::= localValue : 50

illegalSS-Operation ERROR
PARAMETER
  illegalSS-OperationParam SEQUENCE {
    extensionContainer SEQUENCE {
      privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE (1..10) ) OF
        SEQUENCE {
          extId MAP-EXTENSION .&extensionId ( {
            '...' ) ,
          extType MAP-EXTENSION .&ExtensionType ( {
            '...' { @extId } ) OPTIONAL} OPTIONAL,
        pcs-Extensions [1] IMPLICIT SEQUENCE {
          ... } OPTIONAL,
          ... } OPTIONAL,
          ... }
    }
  ::= localValue : 16

ss-ErrorStatus ERROR
PARAMETER
  ss-Status OCTET STRING ( SIZE (1) )
  ::= localValue : 17

ss-NotAvailable ERROR
PARAMETER
  ss-NotAvailableParam SEQUENCE {
    extensionContainer SEQUENCE {
      privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE (1..10) ) OF
        SEQUENCE {
          extId MAP-EXTENSION .&extensionId ( {
            '...' ) ,
          extType MAP-EXTENSION .&ExtensionType ( {
            '...' { @extId } ) OPTIONAL} OPTIONAL,
        pcs-Extensions [1] IMPLICIT SEQUENCE {
          ... } OPTIONAL,
          ... } OPTIONAL,
          ... }
    }
  ::= localValue : 18

ss-SubscriptionViolation ERROR
PARAMETER
  ss-SubscriptionViolationParam SEQUENCE {
    extensionContainer SEQUENCE {
      privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE (1..10) ) OF
        SEQUENCE {

```

```

        extId      MAP-EXTENSION .&extensionId ( {
            '...'} ) ,
        extType    MAP-EXTENSION .&ExtensionType ( {
            '...'} { @extId } ) OPTIONAL} OPTIONAL,
        pcs-Extensions [1] IMPLICIT SEQUENCE {
            ... } OPTIONAL,
            ... } OPTIONAL,
        ... }
 ::= 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

shortTermDenial ERROR
PARAMETER
    shortTermDenialParam SEQUENCE {
        ... }
 ::= localValue : 29

longTermDenial ERROR
PARAMETER
    longTermDenialParam SEQUENCE {
        ... }
 ::= localValue : 30

subscriberBusyForMT-SMS ERROR
PARAMETER
    subBusyForMT-SMS-Param SEQUENCE {
        extensionContainer SEQUENCE {
            privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE ( 1..10 ) ) OF
                SEQUENCE {
                    extId      MAP-EXTENSION .&extensionId ( {
                        '...'} ) ,
                    extType    MAP-EXTENSION .&ExtensionType ( {
                        '...'} { @extId } ) OPTIONAL} OPTIONAL,
                pcs-Extensions [1] IMPLICIT SEQUENCE {
                    ... } OPTIONAL,
                    ... } OPTIONAL,
                ... ,
                gprsConnectionSuspended NULL OPTIONAL}
        ... }
 ::= localValue : 31

sm-DeliveryFailure ERROR
PARAMETER
    sm-DeliveryFailureCause 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,
    extensionContainer      SEQUENCE {
        privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE (1..10 ) ) OF
            SEQUENCE {
                extId          MAP-EXTENSION .&extensionId ( {
                    '...'} ) ,
                extType      MAP-EXTENSION .&ExtensionType ( {
                    '...'} { @extId } ) OPTIONAL} OPTIONAL,
        pcs-Extensions      [1] IMPLICIT SEQUENCE {
            ... } OPTIONAL,
        ... } OPTIONAL,
        ... }
 ::= localValue : 32

messageWaitingListFull ERROR
PARAMETER
    messageWaitListFullParam SEQUENCE {
        extensionContainer SEQUENCE {
            privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE (1..10 ) ) OF
                SEQUENCE {
                    extId          MAP-EXTENSION .&extensionId ( {
                        '...'} ) ,
                    extType      MAP-EXTENSION .&ExtensionType ( {
                        '...'} { @extId } ) OPTIONAL} OPTIONAL,
            pcs-Extensions      [1] IMPLICIT SEQUENCE {
                ... } OPTIONAL,
                ... } OPTIONAL,
                ... }
 ::= localValue : 33

absentsubscriberSM ERROR
PARAMETER
    absentSubscriberSM-Param SEQUENCE {
        absentSubscriberDiagnosticSM          INTEGER ( 0..255 ) OPTIONAL,
        extensionContainer SEQUENCE {
            privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE (1..10 ) ) OF
                SEQUENCE {
                    extId          MAP-EXTENSION .&extensionId ( {
                        '...'} ) ,
                    extType      MAP-EXTENSION .&ExtensionType ( {
                        '...'} { @extId } ) OPTIONAL} OPTIONAL,
            pcs-Extensions      [1] IMPLICIT SEQUENCE {
                ... } OPTIONAL,
                ... } OPTIONAL,
                ... ,
            additionalAbsentSubscriberDiagnosticSM [0] IMPLICIT INTEGER ( 0..255 ) OPTIONAL}
 ::= localValue : 6

unauthorizedRequestingNetwork ERROR
PARAMETER
    unauthorizedRequestingNetwork-Param SEQUENCE {
        extensionContainer SEQUENCE {
            privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE (1..10 ) ) OF
                SEQUENCE {
                    extId          MAP-EXTENSION .&extensionId ( {
                        '...'} ) ,
                    extType      MAP-EXTENSION .&ExtensionType ( {
                        '...'} { @extId } ) OPTIONAL} OPTIONAL,
            pcs-Extensions      [1] IMPLICIT SEQUENCE {
                ... } OPTIONAL,
                ... } OPTIONAL,
                ... }
 ::= localValue : 52

unauthorizedLCSClient ERROR
PARAMETER
    unauthorizedLCSClient-Param SEQUENCE {
        unauthorizedLCSClient-Diagnostic [0] IMPLICIT ENUMERATED {
            noAdditionalInformation          ( 0 ),
            clientNotInMSPrivacyExceptionList ( 1 ),
            callToClientNotSetup           ( 2 ),
            privacyOverrideNotApplicable    ( 3 ),
            disallowedByLocalRegulatoryRequirements ( 4 ),

```



```

... } OPTIONAL,
extensionContainer [1] IMPLICIT SEQUENCE {
  privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE (1..10 ) ) OF
    SEQUENCE {
      extId MAP-EXTENSION .&extensionId ( {
        '... } ) ,
      extType MAP-EXTENSION .&ExtensionType ( {
        '... } { @extId } ) OPTIONAL} OPTIONAL,
  pcs-Extensions [1] IMPLICIT SEQUENCE {
    ... } OPTIONAL,
    ... } OPTIONAL,
    ... }
 ::= localValue : 53

positionMethodFailure ERROR
PARAMETER
  positionMethodFailure-Param SEQUENCE {
    positionMethodFailure-Diagnostic [0] IMPLICIT ENUMERATED {
      congestion (0 ),
      insufficientResources (1 ),
      insufficientMeasurementData (2 ),
      inconsistentMeasurementData (3 ),
      locationProcedureNotCompleted (4 ),
      locationProcedureNotSupportedByTargetMS (5 ),
      qosNotAttainable (6 ),
      ... } OPTIONAL,
    extensionContainer [1] IMPLICIT SEQUENCE {
      privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE (1..10 ) ) OF
        SEQUENCE {
          extId MAP-EXTENSION .&extensionId ( {
            '... } ) ,
          extType MAP-EXTENSION .&ExtensionType ( {
            '... } { @extId } ) OPTIONAL} OPTIONAL,
        pcs-Extensions [1] IMPLICIT SEQUENCE {
          ... } OPTIONAL,
          ... } OPTIONAL,
          ... }
 ::= localValue : 54

unknownOrUnreachableLCSCClient ERROR
PARAMETER
  unknownOrUnreachableLCSCClient-Param SEQUENCE {
    extensionContainer SEQUENCE {
      privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE (1..10 ) ) OF
        SEQUENCE {
          extId MAP-EXTENSION .&extensionId ( {
            '... } ) ,
          extType MAP-EXTENSION .&ExtensionType ( {
            '... } { @extId } ) OPTIONAL} OPTIONAL,
        pcs-Extensions [1] IMPLICIT SEQUENCE {
          ... } OPTIONAL,
          ... } OPTIONAL,
          ... }
 ::= localValue : 58

mm-EventNotSupported ERROR
PARAMETER
  mm-EventNotSupported-Param SEQUENCE {
    extensionContainer SEQUENCE {
      privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE (1..10 ) ) OF
        SEQUENCE {
          extId MAP-EXTENSION .&extensionId ( {
            '... } ) ,
          extType MAP-EXTENSION .&ExtensionType ( {
            '... } { @extId } ) OPTIONAL} OPTIONAL,
        pcs-Extensions [1] IMPLICIT SEQUENCE {
          ... } OPTIONAL,
          ... } OPTIONAL,
          ... }
 ::= localValue : 59

```

END

B.2 Fully Expanded ASN.1 Source of MAP-DialogueInformation

```

-- Expanded ASN1 Module 'MAP-DialogueInformation'
--SIEMENS ASN.1 Compiler      R4.21 (42-00-04)
--                               Date: 00-01-06 Time: 07:32:40

MAP-DialogueInformation{ 0 identified-organization (4) etsi (0) mobileDomain (0) gsm-Network (1)
modules (3) map-DialogueInformation (3) version6 (6) }

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,
        ... ,
        extensionContainer   SEQUENCE {
            privateExtensionList  [0] IMPLICIT SEQUENCE ( SIZE (1..10 ) ) OF
                SEQUENCE {
                    extId          MAP-EXTENSION .&extensionId ( {
                        '...' } ) ,
                    extType       MAP-EXTENSION .&ExtensionType ( {
                        '...' } { @extId } ) OPTIONAL} OPTIONAL,
            pcs-Extensions      [1] IMPLICIT SEQUENCE {
                ... } OPTIONAL,
            ... } OPTIONAL},
    map-accept        [1] IMPLICIT SEQUENCE {
        ... ,
        extensionContainer   SEQUENCE {
            privateExtensionList  [0] IMPLICIT SEQUENCE ( SIZE (1..10 ) ) OF
                SEQUENCE {
                    extId          MAP-EXTENSION .&extensionId ( {
                        '...' } ) ,
                    extType       MAP-EXTENSION .&ExtensionType ( {
                        '...' } { @extId } ) OPTIONAL} OPTIONAL,
            pcs-Extensions      [1] IMPLICIT SEQUENCE {
                ... } OPTIONAL,
            ... } OPTIONAL},
    map-close         [2] IMPLICIT SEQUENCE {
        ... ,
        extensionContainer   SEQUENCE {
            privateExtensionList  [0] IMPLICIT SEQUENCE ( SIZE (1..10 ) ) OF
                SEQUENCE {
                    extId          MAP-EXTENSION .&extensionId ( {
                        '...' } ) ,
                    extType       MAP-EXTENSION .&ExtensionType ( {
                        '...' } { @extId } ) OPTIONAL} OPTIONAL,
            pcs-Extensions      [1] IMPLICIT SEQUENCE {
                ... } OPTIONAL,
            ... } OPTIONAL},
    map-refuse        [3] IMPLICIT SEQUENCE {
        reason             ENUMERATED {
            noReasonGiven      ( 0 ),
            invalidDestinationReference  ( 1 ),
            invalidOriginatingReference  ( 2 )},
        ... ,
        extensionContainer   SEQUENCE {
            privateExtensionList  [0] IMPLICIT SEQUENCE ( SIZE (1..10 ) ) OF
                SEQUENCE {
                    extId          MAP-EXTENSION .&extensionId ( {

```

```

        '... } ) ,
        extType    MAP-EXTENSION .&ExtensionType ( {
            '... } { @extId } ) OPTIONAL} OPTIONAL,
        pcs-Extensions    [1] IMPLICIT SEQUENCE {
            ... } OPTIONAL,
            ... } OPTIONAL},
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 ) }},
        ... ,
    extensionContainer    SEQUENCE {
        privateExtensionList    [0] IMPLICIT SEQUENCE ( SIZE (1..10 ) ) OF
            SEQUENCE {
                extId    MAP-EXTENSION .&extensionId ( {
                    '... } ) ,
                extType    MAP-EXTENSION .&ExtensionType ( {
                    '... } { @extId } ) OPTIONAL} OPTIONAL,
                pcs-Extensions    [1] IMPLICIT SEQUENCE {
                    ... } OPTIONAL,
                    ... } OPTIONAL},
map-providerAbort    [5] IMPLICIT SEQUENCE {
    map-ProviderAbortReason    ENUMERATED {
        abnormalDialogue    ( 0 ),
        invalidPDU    ( 1 ) },
        ... ,
    extensionContainer    SEQUENCE {
        privateExtensionList    [0] IMPLICIT SEQUENCE ( SIZE (1..10 ) ) OF
            SEQUENCE {
                extId    MAP-EXTENSION .&extensionId ( {
                    '... } ) ,
                extType    MAP-EXTENSION .&ExtensionType ( {
                    '... } { @extId } ) OPTIONAL} OPTIONAL,
                pcs-Extensions    [1] IMPLICIT SEQUENCE {
                    ... } OPTIONAL,
                    ... } OPTIONAL}}
MAP-OpenInfo ::= SEQUENCE {
    destinationReference    [0] IMPLICIT OCTET STRING ( SIZE (1..20 ) ) OPTIONAL,
    originationReference    [1] IMPLICIT OCTET STRING ( SIZE (1..20 ) ) OPTIONAL,
    ... ,
    extensionContainer    SEQUENCE {
        privateExtensionList    [0] IMPLICIT SEQUENCE ( SIZE (1..10 ) ) OF
            SEQUENCE {
                extId    MAP-EXTENSION .&extensionId ( {
                    '... } ) ,
                extType    MAP-EXTENSION .&ExtensionType ( {
                    '... } { @extId } ) OPTIONAL} OPTIONAL,
                pcs-Extensions    [1] IMPLICIT SEQUENCE {
                    ... } OPTIONAL,
                    ... } OPTIONAL}
MAP-AcceptInfo ::= SEQUENCE {
    ... ,
    extensionContainer    SEQUENCE {
        privateExtensionList    [0] IMPLICIT SEQUENCE ( SIZE (1..10 ) ) OF
            SEQUENCE {
                extId    MAP-EXTENSION .&extensionId ( {
                    '... } ) ,
                extType    MAP-EXTENSION .&ExtensionType ( {
                    '... } { @extId } ) OPTIONAL} OPTIONAL,
                pcs-Extensions    [1] IMPLICIT SEQUENCE {

```

```

... } OPTIONAL,
... } OPTIONAL}

```

```

MAP-CloseInfo ::= SEQUENCE {
... ,
extensionContainer SEQUENCE {
privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE (1..10 ) ) OF
SEQUENCE {
extId MAP-EXTENSION .&extensionId ( {
'
... } ) ,
extType MAP-EXTENSION .&ExtensionType ( {
'
... } { @extId } ) OPTIONAL} OPTIONAL,
pcs-Extensions [1] IMPLICIT SEQUENCE {
... } OPTIONAL,
... } OPTIONAL}

```

```

MAP-RefuseInfo ::= SEQUENCE {
reason ENUMERATED {
noReasonGiven (0 ),
invalidDestinationReference (1 ),
invalidOriginatingReference (2 )},
... ,
extensionContainer SEQUENCE {
privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE (1..10 ) ) OF
SEQUENCE {
extId MAP-EXTENSION .&extensionId ( {
'
... } ) ,
extType MAP-EXTENSION .&ExtensionType ( {
'
... } { @extId } ) OPTIONAL} OPTIONAL,
pcs-Extensions [1] IMPLICIT SEQUENCE {
... } OPTIONAL,
... } OPTIONAL}

```

```

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 )}},
... ,
extensionContainer SEQUENCE {
privateExtensionList [0] IMPLICIT SEQUENCE ( SIZE (1..10 ) ) OF
SEQUENCE {
extId MAP-EXTENSION .&extensionId ( {
'
... } ) ,
extType MAP-EXTENSION .&ExtensionType ( {
'
... } { @extId } ) OPTIONAL} OPTIONAL,
pcs-Extensions [1] IMPLICIT SEQUENCE {
... } OPTIONAL,
... } OPTIONAL}

```

```

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 )},
  ... ,
  extensionContainer       SEQUENCE {
    privateExtensionList  [0] IMPLICIT SEQUENCE ( SIZE (1..10 ) ) OF
      SEQUENCE {
        extId      MAP-EXTENSION .&extensionId ( {
          ... } ) ,
        extType    MAP-EXTENSION .&ExtensionType ( {
          ... } { @extId } ) OPTIONAL} OPTIONAL,
    pcs-Extensions [1] IMPLICIT SEQUENCE {
      ... } OPTIONAL,
    ... } OPTIONAL}
```

```
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.

Annex D (informative): Clause mapping table

D.1 Mapping of Clause numbers

The clause numbers have been modified according to table D.1.

Table D.1: Clause mapping from Version 5.9.0 to Version 6.0.0

Old Clause No (V5.9.0)	New Clause No (V6.0.0)	Old Clause No (V5.9.0)	New Clause No (V6.0.0)
1.1	2	17.*	20.*
1.2	3	18.*	21.*
2.*	4.*	19.*	22.*
3.*	5.*	19.0.*	22.1.*
4.*	6.*	19.1.*	22.2.*
5.*	7.*	19.2.*	22.3.*
6.*	8.*	19.3.*	22.4.*
7.*	9.*	19.4.*	22.5.*
8.*	10.*	19.5.*	22.6.*
9.*	11.*	19.6.*	22.7.*
10.*	12.*	19.7.*	22.8.*
new11.*	13.*	19.8.*	22.9.*
old11.*	14.*	19.9.*	22.10.*
12.*	15.*	19.10.*	22.11.*
13.*	16.*	19.11.*	22.12.*
14.*	17.*	20.*	23.*
15.*	18.*	new22.*	24.*
16.*	19.*	old21.*	25.*

Annex E (informative): Change History

SMG#	TDoc	SPEC	VERS	CR	R E V	PHASE	C A T	SUBJECT	NEW_VERS	WORKITEM
04	N2-99227	29.002	3.0.0	A002	3	R98	A	Use of E interface	3.1.0	
04	N2-99578	29.002	3.0.0	A003		R98	B	Introduction of TIF-CSI for Call Deflection	3.1.0	
04	N2-99233	29.002	3.0.0	A004		R98	A	Clarification in ASN.1 encoding of O-CSI and T-CSI	3.1.0	
04	N2-99269	29.002	3.0.0	A005		R98	C	Introduction of MSISDN in USSD operation	3.1.0	
04	N2-99650	29.002	3.0.0	A006		R98	A	Modification of the O-CSI ASN.1 structure	3.1.0	
04	N2-99250	29.002	3.0.0	A007		R98	A	Adding of MAP_DELIMITER_req to the Status report operation	3.1.0	
04	N2-99628	29.002	3.0.0	A008		R98	A	Correction to the Purge MS "Detailed procedure in the HLR"	3.1.0	
04	N2-99677	29.002	3.0.0	A009		R98	A	Adding of MNP-indicator to the SRI ack	3.1.0	
04	N2-99228	29.002	3.0.0	A010		R98	F	New subscription options for call forwarding	3.1.0	
04	N2-99585	29.002	3.0.0	A011		R98	C	Adding the support of ANSI SCCP which is required in North America (World Zone 1)	3.1.0	
04	N2-99515	29.002	3.0.0	A012		R98	A	Introduction of 3-digit MNCs correction	3.1.0	
04	N2-99520	29.002	3.0.0	A013		R98	F	Export of NAEA-CIC	3.1.0	
04	N2-99548	29.002	3.0.0	A014		R98	D	Clarification to text to identify how the LSA data relevant in the current VPLMN can be determined	3.1.0	
04	3C99-468	29.002	3.0.0	A015		R97	F	Alignment with 04.80	3.1.0	
04	N2-99519	29.002	3.0.0	A016		R98	A	VBS data	3.1.0	
04	N2-99461	29.002	3.0.0	A017		R98	F	Introduction of Data Missing error to the Resume Call Handling	3.1.0	
04	N2-99583	29.002	3.0.0	A018		R97	F	Removal of 3-digit MNCs	3.1.0	
04	N2-99676	29.002	3.0.0	A019		R98	A	Corrections of mapping from MAP service to TC service	3.1.0	
04	3C99-206	29.002	3.0.0	A020		R98	B	Introduction of UUS service to Resume Call Handling	3.1.0	
05	N2-99906	29.002	3.1.0	021		R99	A	Clarification on VLR CAMEL Subscription Info	3.2.0	CAMEL Phase 2
05	N2-99908	29.002	3.1.0	022		R99	A	Clarification on DestinationNumberCriteria	3.2.0	CAMEL Phase 2

SMG#	TDoc	SPEC	VERS	CR	R E V	PHASE	C A T	SUBJECT	NEW_VERS	WORKITEM
05	N2-99910	29.002	3.1.0	023		R99	A	Removal of TDP-Criteria from RCH	3.2.0	CAMEL Phase 2
05	N2-99934	29.002	3.1.0	025		R99	A	Various corrections related to GGSN-HLR Interface.	3.2.0	GPRS
05	N2-99936	29.002	3.1.0	034		R99	A	Update Location handling for GPRS-only subscription	3.2.0	GPRS
05	N2-99938	29.002	3.1.0	035		R99	A	Correction of OP & AC definitions for NoteMS-PresentForGPRS	3.2.0	GPRS
05	N2-99952	29.002	3.1.0	036		R99	A	Removal of redundant information from RCH	3.2.0	UUS
05	N2-99956	29.002	3.1.0	026		R99	A	OR capability IE in PRN	3.2.0	TEI
05	N2-99964	29.002	3.1.0	024	1	R99	A	GMSC-CAMEL phase 2 support IE in PRN	3.2.0	CAMEL Phase 2
05	N2-99A19	29.002	3.1.0	028		R99	A	Alignment of 29.002 with 02.67	3.2.0	eMLPP
05	N2-99A45	29.002	3.1.0	029	1	R99	B	Non-CAMEL IST implementation	3.2.0	IST
05	N2-99B57	29.002	3.1.0	027	2	R99	B	Addition of the information elements and the ASN.1 definitions for Pre-paging	3.2.0	Pre-Paging
05	N2-99C27	29.002	3.1.0	042		R99	A	Clarification on 'Supported CAMEL Phases' in ISD ack	3.2.0	CAMEL Phase 2
05	N2-99C78	29.002	3.1.0	044		R99	A	Editing error correction on VLR capabilities	3.2.0	SoLSA
05	N2-99D06	29.002	3.1.0	043	1	R99	A	Addition of exception handling to the CancellationType	3.2.0	GPRS
05	N2-99D33	29.002	3.1.0	046		R99	A	Clarification of LR-REJECT cause corresponding to RoamingRestrictionDueToUnsupportedFeature	3.2.0	TEI
05	N2-99D35	29.002	3.1.0	047		R99		Clarification of returning the MSISDN in SRIack	3.2.0	MNP
06	N2-99G06	29.002	3.2.0	033	3	R99	C	Introduction of the Super-Charger Concept in TS 29.002	3.3.0	Super Charger
06	N2-99G18	29.002	3.2.0	032	2	R99	C	Introduction of White Book SCCP in MAP	3.3.0	TEI
06	N2-99G50	29.002	3.2.0	070		R99	A	Addition of GGSN number for the SRIforGPRS	3.3.0	GPRS
06	N2-99J88	29.002	3.2.0	075	1	R99	B	Introduction of Follow Me	3.3.0	Follow Me
06	N2-99K12	29.002	3.2.0	077		R99	A	Use of SSN for GPRS	3.3.0	GPRS

SMG#	TDoc	SPEC	VERS	CR	R E V	PHASE	C A T	SUBJECT	NEW_VERS	WORKITEM
06	N2-99K24	29.002	3.2.0	069		R99	A	Correction of the USSD procedure in the HLR.	3.3.0	USSD & Follow Me
06	N2-99K52	29.002	3.2.0	060	1	R99	C	MAP Impacts for Location Services (LCS)	3.3.0	Location Services
06	N2-99K58	29.002	3.2.0	045	4	R99	B	Authentication Enhancements	3.3.0	Security
06	N2-99K60	29.002	3.2.0	050	5	R99	C	QoS-Subscribed field modification	3.3.0	QoS enhancements
06	N2-99L20	29.002	3.2.0	073	1	R99	C	Introduction of CAMEL Phase 3 in 3G TS 29.002	3.3.0	CAMEL Phase 3
06	N2-99J52	29.002	3.2.0	074		R99	D	Restructuring of MAP Location Management Procedures for the Circuit Switched Domain	3.3.0	TEI
06	N2-99J92	29.002	3.2.0	068		R99	B	Update of SDLs to support Super-Charger	3.3.0	Super-Charger

History

Document history		
V3.3.0	January 2000	Publication