3GPP TS 29.002 V3.20.0 (2004-06)

Technical Specification

3rd Generation Partnership Project; Technical Specification Group Core Network; Mobile Application Part (MAP) specification (Release 1999)

The present document has been developed within the 3rd Generation Partnership Project (3GPP TM) and may be further elaborated for the purposes of 3GPP.

The present document has not been subject to any approval process by the 3GPP Organisational Partners and shall not be implemented.
This Specification is provided for future development work within 3GPP only. The Organisational Partners accept no liability for any use of this Specification.
Specifications and reports for implementation of the 3GPP TM system should be obtained via the 3GPP Organisational Partners' Publications Offices.

Keywords
GSM. UMTS, MAP, SS7, network

3GPP

Postal address

3GPP support office address
650 Route des Lucioles - Sophia Antipolis
Valbonne - FRANCE
Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Internet
http://www.3gpp.org

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.

© 2004, 3GPP Organizational Partners (ARIB, ATIS, CCSA, ETSI, TTA, TTC).
All rights reserved.

Contents

Foreword 25
1 Scope 26
2 References 26
3 Abbreviations 31
4 Void 31
5 Overload and compatibility overview 32
5.1 Overload control 32
5.1.1 Overload control for MSC (outside MAP) 32
5.1.2 Overload control for MAP entities 32
5.1.3 Congestion control for Signalling System No. 7 36
5.2 Compatibility 36
5.2.1 General 36
5.2.2 Strategy for selecting the Application Context (AC) version 36
5.2.2.1 Proposed method 36
5.2.2.2 Managing the version look-up table 37
5.2.2.3 Optimising the method 38
6 Requirements concerning the use of SCCP and TC 38
6.1 Use of SCCP 38
6.1.1 SCCP Class 38
6.1.2 Sub-System Number (SSN) 39
6.1.3 SCCP addressing 39
6.1.3.1 Introduction 39
6.1.3.2 The Mobile-services Switching Centre (MSC) 41
6.1.3.2.1 MSC interaction during handover or relocation 41
6.1.3.2.2 MSC for short message routing 41
6.1.3.2.3 MSC for location request routing 41
6.1.3.2.4 MSC for LMU Control 41
6.1.3.3 The Home Location Register (HLR) 41
6.1.3.3.1 During call set-up 41
6.1.3.3.2 Before location updating completion 42
6.1.3.3.3 After location updating completion 42
6.1.3.3.4 VLR restoration 43
6.1.3.3.5 During Network-Requested PDP Context Activation 43
6.1.3.3.6 Before GPRS location updating completion 43
6.1.3.3.7 After GPRS location updating completion 44
6.1.3.3.8 Query for a Location Request 44
6.1.3.4 The Visitor Location Register (VLR) 44
6.1.3.4.1 Inter-VLR information retrieval 44
6.1.3.4.2 HLR request 44
6.1.3.5 The Interworking MSC (IWMSC) for Short Message Service 44
6.1.3.6 The Equipment Identity Register (EIR) 44
6.1.3.7 The Shared Inter Working Function (SIWF) 45
6.1.3.8 The Serving GPRS Support Node (SGSN) 45
6.1.3.9 The Gateway GPRS Support Node (GGSN) 45
6.1.3.10 The Gateway MSC (GMSC) for Short Message Service 45
6.1.3.10A Void 45
6.1.3.10A.1 Void 45
6.1.3.10A.2 Void 45
6.1.3.10B The Gateway Mobile Location Centre (GMLC) 45
6.1.3.11 Summary table 45
6.2 Use of TC 48
7 General on MAP services 49
7.1 Terminology and definitions 49

7.2 Modelling principles 49
7.3 Common MAP services 50
7.3.1 MAP-OPEN service 51
7.3.2 MAP-CLOSE service 54
7.3.3 MAP-DELIMITER service 54
7.3.4 MAP-U-ABORT service 54
7.3.5 MAP-P-ABORT service 55
7.3.6 MAP-NOTICE service 56
7.4 Sequencing of services 56
7.5 General rules for mapping of services onto TC 58
7.5.1 Mapping of common services 58
7.5.2 Mapping of user specific services 59
7.6 Definition of parameters 60
7.6.1 Common parameters 63
7.6.1.1 Invoke Id 63
7.6.1.2 Linked Id 63
7.6.1.3 Provider error 63
7.6.1.4 User error 63
7.6.1.5 All Information Sent 66
7.6.2 Numbering and identification parameters 66
7.6.2.1 IMSI 66
7.6.2.2 TMSI 66
7.6.2.3 IMEI 66
7.6.2.4 Previous location area Id 66
7.6.2.5 Stored location area Id 66
7.6.2.6 Current location area Id 67
7.6.2.7 Target location area Id 67
7.6.2.8 Target cell Id 67
7.6.2.8A Target RNC Id 67
7.6.2.9 Void 67
7.6.2.10 Originating entity number 67
7.6.2.11 MSC number 67
7.6.2.12 Target MSC number 67
7.6.2.13 HLR number 67
7.6.2.14 VLR number 67
7.6.2.15 HLR Id 67
7.6.2.16 LMSI 67
7.6.2.17 MS ISDN 67
7.6.2.18 OMC Id 67
7.6.2.19 Roaming number 68
7.6.2.19A Relocation Number List 68
7.6.2.20 Void 68
7.6.2.21 Handover number 68
7.6.2.22 Forwarded-to number 68
7.6.2.22A Long forwarded-to number 68
7.6.2.22B Long FTN Supported 68
7.6.2.23 Forwarded-to subaddress 68
7.6.2.24 Called number 68
7.6.2.25 Calling number 68
7.6.2.26 Originally dialled number 68
7.6.2.27 Service centre address 68
7.6.2.28 Zone Code 68
7.6.2.29 MSIsdn-Alert 68
7.6.2.30 Location Information 69
7.6.2.31 GMSC Address 69
7.6.2.32 VMSC Address 69
7.6.2.33 Group Id 69
7.6.2.34 North American Equal Access preferred Carrier Id 69
7.6.2.35 SIWFS Number 69
7.6.2.36 B-subscriber address 69
7.6.2.37 Serving cell Id 69
7.6.2.38 SGSN number 69

7.6.2.39 SGSN address ... 69
7.6.2.40 GGSN address ... 69
7.6.2.41 GGSN number ... 69
7.6.2.42 APN ... 70
7.6.2.43 Network Node number ... 70
7.6.2.44 PDP-Type ... 70
7.6.2.45 PDP-Address ... 70
7.6.2.46 Additional number ... 70
7.6.2.47 P-TMSI ... 70
7.6.2.48 B-subscriber number ... 70
7.6.2.49 B-subscriber subaddress ... 70
7.6.2.50 LMU Number ... 70
7.6.2.51 MLC Number ... 70
7.6.2.52 Multicall Bearer Information ... 70
7.6.2.53 Multiple Bearer Requested ... 70
7.6.2.54 Multiple Bearer Not Supported ... 70
7.6.2.55 PDP-Charging Characteristics ... 70
7.6.2.56 Selected RAB ID ... 71
7.6.2.57 RAB ID ... 71
7.6.3 Subscriber management parameters ... 71
7.6.3.1 Category ... 71
7.6.3.2 Equipment status ... 71
7.6.3.3 Extensible Bearer service ... 71
7.6.3.4 Extensible Teleservice ... 71
7.6.3.5 Extensible Basic Service Group ... 71
7.6.3.6 GSM bearer capability ... 71
7.6.3.7 Subscriber Status ... 71
7.6.3.8 CUG Outgoing Access indicator ... 71
7.6.3.9 Operator Determined Barring General Data ... 71
7.6.3.10 ODB HPLMN Specific Data ... 72
7.6.3.11 Regional Subscription Data ... 73
7.6.3.12 Regional Subscription Response ... 73
7.6.3.13 Roaming Restriction Due To Unsupported Feature ... 73
7.6.3.14 Extensible SS-Info ... 73
7.6.3.15 Extensible forwarding information ... 73
7.6.3.16 Extensible forwarding feature ... 73
7.6.3.17 Extensible SS-Status ... 74
7.6.3.18 Extensible Forwarding Options ... 74
7.6.3.19 Extensible No reply condition timer ... 74
7.6.3.20 Extensible Call barring information ... 74
7.6.3.21 Extensible Call barring feature ... 74
7.6.3.22 CUG info ... 74
7.6.3.23 CUG subscription ... 75
7.6.3.24 CUG interlock ... 75
7.6.3.25 CUG index ... 75
7.6.3.26 CUG feature ... 75
7.6.3.27 Inter CUG options ... 75
7.6.3.28 Intra CUG restrictions ... 75
7.6.3.29 Extensible SS-Data ... 76
7.6.3.30 Subscriber State ... 76
7.6.3.31 Requested Info ... 76
7.6.3.32 Suppression of Announcement ... 76
7.6.3.33 Suppress T-CSI ... 76
7.6.3.34 GMSC CAMEL Subscription Info ... 76
7.6.3.35 VLR CAMEL Subscription Info ... 76
7.6.3.36 Supported CAMEL Phases in the VLR ... 76
7.6.3.36A Supported CAMEL Phases in the SGSN ... 76
7.6.3.37 CUG Subscription Flag ... 76
7.6.3.38 CAMEL Subscription Info Withdraw ... 76
7.6.3.39 Voice Group Call Service (VGCS) Data ... 76
7.6.3.40 Voice Broadcast Service (VBS) Data ... 77
7.6.3.41 ISDN bearer capability ... 77

7.6.3.42 Lower layer Compatibility ... 77
7.6.3.43 High Layer Compatibility ... 77
7.6.3.44 Alerting Pattern ... 77
7.6.3.45 GPRS Subscription Data Withdraw ... 77
7.6.3.46 GPRS Subscription Data ... 77
7.6.3.47 QoS-Subscribed ... 77
7.6.3.48 VPLMN address allowed ... 77
7.6.3.49 Roaming Restricted In SGSN Due To Unsupported Feature ... 77
7.6.3.50 Network Access Mode ... 77
7.6.3.51 Mobile Not Reachable Reason ... 77
7.6.3.52 Cancellation Type ... 77
7.6.3.53 All GPRS Data ... 78
7.6.3.54 Complete Data List Included ... 78
7.6.3.55 PDP Context Identifier ... 78
7.6.3.56 LSA Information ... 78
7.6.3.57 SoLSA support indicator ... 78
7.6.3.58 LSA Information Withdraw ... 78
7.6.3.59 LMU Indicator ... 78
7.6.3.60 LCS Information ... 78
7.6.3.61 GMLC List ... 78
7.6.3.62 LCS Privacy Exception List ... 78
7.6.3.63 LCS Privacy Exception Parameters ... 78
7.6.3.64 External Client List ... 79
7.6.3.65 Internal Client List ... 79
7.6.3.65A MO-LR List ... 79
7.6.3.65B Privacy Notification to MS User ... 79
7.6.3.65C GMLC List Withdraw ... 79
7.6.3.66 IST Alert Timer ... 79
7.6.3.67 Call Termination Indicator ... 79
7.6.3.68 IST Information Withdraw ... 79
7.6.3.69 IST Support Indicator ... 79
7.6.3.70 Super-Charger Supported In HLR ... 80
7.6.3.71 Super-Charger Supported In Serving Network Entity ... 80
7.6.3.72 Age Indicator ... 80
7.6.3.73 GPRS enhancements support indicator ... 80
7.6.3.74 Extensible QoS-Subscribed ... 80
7.6.3.75 SGSN Camel Subscription Info ... 80
7.6.3.76 SMS-CSI ... 80
7.6.3.77 GPRS-CSI ... 80
7.6.3.78 CAMEL subscription info ... 80
7.6.3.79 Extensible Call barring information for CSE ... 80
7.6.3.80 Extensible Forwarding information for CSE ... 81
7.6.3.81 Modification Request for CSI ... 81
7.6.3.82 Modification Request for SS Information ... 81
7.6.3.83 Call Barring Data ... 81
7.6.3.84 Call Forwarding Data ... 81
7.6.3.85 ODB Data ... 81
7.6.3.86 Requested Subscription Info ... 81
7.6.3.87 CS Allocation/Retention priority ... 81
7.6.3.88 ODB Info ... 81
7.6.4 Supplementary services parameters ... 81
7.6.4.1 SS-Code ... 81
7.6.4.2 SS-Status ... 82
7.6.4.3 SS-Data ... 82
7.6.4.4 Override Category ... 82
7.6.4.5 CLI Restriction Option ... 82
7.6.4.6 Forwarding Options ... 83
7.6.4.7 No reply condition timer ... 83
7.6.4.8 - 7.6.4.14 Void ... 83
7.6.4.15 Forwarding information ... 83
7.6.4.16 Forwarding feature ... 83
7.6.4.17 Void ... 83

7.6.4.18 Call barring information 83
7.6.4.19 Call barring feature 84
7.6.4.20 New password 84
7.6.4.21 Current password 84
7.6.4.22 Guidance information 84
7.6.4.23 Void 84
7.6.4.24 SS-Info 84
7.6.4.25 - 7.6.4.35 Void 84
7.6.4.36 USSD Data Coding Scheme 85
7.6.4.37 USSD String 85
7.6.4.38 Bearer service 85
7.6.4.39 Teleservice 85
7.6.4.40 Basic Service Group 85
7.6.4.41 eMLPP information 85
7.6.4.42 SS-event 85
7.6.4.43 SS-event data 85
7.6.4.44 LCS Privacy Exceptions 86
7.6.4.45 Mobile Originating Location Request (MO-LR) 86
7.6.4.46 NbrUser 86
7.6.4.47 MC Subscription Data 86
7.6.4.48 MC Information 86
7.6.4.49 CCBS Request State 86
7.6.5 Call parameters 87
7.6.5.1 Call reference number 87
7.6.5.2 Interrogation type 87
7.6.5.3 OR interrogation 87
7.6.5.4 OR capability 87
7.6.5.5 Forwarding reason 87
7.6.5.6 Forwarding interrogation required 87
7.6.5.7 O-CSI 87
7.6.5.7A D-CSI 87
7.6.5.7B T-CSI 87
7.6.5.7C VT-CSI 88
7.6.5.8 Call Direction 88
7.6.5.9 Channel Type 88
7.6.5.10 Chosen Channel 88
7.6.5.11 CCBS Feature 88
7.6.5.12 UU Data 88
7.6.5.13 UUS CF Interaction 88
7.6.5.14 Number Portability Status 88
7.6.5.15 Pre-paging supported 88
7.6.6 Radio parameters 88
7.6.6.1 - 7.6.6.4 Void 88
7.6.6.5 BSSMAP Service Handover 88
7.6.6.5A BSSMAP Service Handover List 89
7.6.6.6 RANAP Service Handover 89
7.6.6.7 HO-Number Not Required 89
7.6.6.8 Integrity Protection Information 89
7.6.6.9 Encryption Information 89
7.6.6.10 Radio Resource Information 89
This parameter refers to the Channel Type information element defined in GSM 08.08. 89
7.6.6.10A Radio Resource List 89
7.6.6.10B Chosen Radio Resource Information 89
7.6.6.11 Key Status 89
7.6.6.12 Selected UMTS Algorithms 89
7.6.6.13 Allowed GSM Algorithms 89
7.6.6.14 Allowed UMTS Algorithms 89
7.6.6.15 Selected GSM Algorithm 89
7.6.7 Authentication parameters 90
7.6.7.1 Authentication set list 90
7.6.7.2 Rand 90
7.6.7.3 Sres 90

7.6.7.4 Kc 90
7.6.7.5 Xres 90
7.6.7.5A Ck 90
7.6.7.5B Ik 90
7.6.7.5C Autn 90
7.6.7.6 Cksn 90
7.6.7.6A Ksi 90
7.6.7.6B Auts 90
7.6.7.7 Ciphering mode 90
Current Security Context 91
7.6.7.9 Failure cause 91
7.6.8 Short message parameters 91
7.6.8.1 SM-RP-DA 91
7.6.8.2 SM-RP-OA 91
7.6.8.3 MWD status 91
7.6.8.4 SM-RP-UI 91
7.6.8.5 SM-RP-PRI 91
7.6.8.6 SM Delivery Outcome 91
7.6.8.7 More Messages To Send 92
7.6.8.8 Alert Reason 92
7.6.8.9 Absent Subscriber Diagnostic SM 92
7.6.8.10 Alert Reason Indicator 92
7.6.8.11 Additional SM Delivery Outcome 92
7.6.8.12 Additional Absent Subscriber Diagnostic SM 92
7.6.8.13 Delivery Outcome Indicator 92
7.6.8.14 GPRS Node Indicator 92
7.6.8.15 GPRS Support Indicator 92
7.6.8.16 SM-RP-MTI 92
7.6.8.17 SM-RP-SMEA 93
7.6.9 Access and signalling system related parameters 93
7.6.9.1 AN-apdu 93
7.6.9.2 CM service type 93
7.6.9.3 Access connection status 93
7.6.9.4 External Signal Information 93
7.6.9.5 Access signalling information 93
7.6.9.6 Location update type 93
7.6.9.7 Protocol ID 94
7.6.9.8 Network signal information 94
7.6.9.9 Call Info 94
7.6.9.10 Additional signal info 95
7.6.10 System operations parameters 95
7.6.10.1 Network resources 95
7.6.10.2 Trace reference 95
7.6.10.3 Trace type 95
7.6.11 Location Service Parameters 95
7.6.11.1 Age of Location Estimate 95
7.6.11.2 Void 96
7.6.11.3 Void 96
7.6.11.4 LCS Client ID 96
7.6.11.5 LCS Event 96
7.6.11.6 LCS MLC Data 96
7.6.11.7 LCS Priority 96
7.6.11.8 LCS QoS 96
7.6.11.9 Void 96
7.6.11.10 Void 96
7.6.11.11 Location Estimate 96
7.6.11.12 Location Type 97
7.6.11.13 NA-ESRD 97
7.6.11.14 NA-ESRK 97
7.6.11.15 Void 97
7.6.11.16 Privacy Override 97
7.6.11.17 Void 97

7.6.11.18 Void 97
7.6.11.19 NA-ESRK Request 97
7.6.11.20 Supported GAD Shapes 97
7.6.11.21 Additional Location Estimate 98
7.7 Representation of a list of a basic parameter in service-primitives 98

8 Mobility services 98
8.1 Location management services 98
8.1.1 Void 98
8.1.1.1 Void 98
8.1.1.2 Void 98
8.1.1.3 Void 98
8.1.2 MAP_UPDATE_LOCATION service 98
8.1.2.1 Definition 98
8.1.2.2 Service primitives 98
8.1.2.3 Parameter definitions and use 99
8.1.3 MAP_CANCEL_LOCATION service 100
8.1.3.1 Definition 100
8.1.3.2 Service primitives 101
8.1.3.3 Parameter definitions and use 101
8.1.4 MAP_SEND_IDENTIFICATION service 101
8.1.4.1 Definition 101
8.1.4.2 Service primitives 101
8.1.4.3 Parameter definitions and use 102
8.1.5 Void 103
8.1.5.1 Void 103
8.1.5.2 Void 103
8.1.5.3 Void 103
8.1.6 MAP_PURGE_MS service 103
8.1.6.1 Definition 103
8.1.6.2 Service primitives 103
8.1.6.3 Parameter definitions and use 103
8.1.7 MAP_UPDATE_GPRS_LOCATION service 104
8.1.7.1 Definition 104
8.1.7.2 Service primitives 104
8.1.7.3 Parameter definitions and use 104
8.1.8 MAP-NOTE-MM-EVENT 106
8.1.8.1 Definition 106
8.1.8.2 Service primitives 106
8.1.8.3 Parameter use 106
8.2 Paging and search 107
8.2.1 MAP_PAGE service 107
8.2.1.1 Definition 107
8.2.1.2 Service primitives 107
8.2.1.3 Parameter definitions and use 107
8.2.2 MAP_SEARCH_FOR_MS service 108
8.2.2.1 Definition 108
8.2.2.2 Service primitives 108
8.2.2.3 Parameter definitions and use 108
8.3 Access management services 109
8.3.1 MAP_PROCESS_ACCESS_REQUEST service 109
8.3.1.1 Definition 109
8.3.1.2 Service primitives 109
8.3.1.3 Parameter definitions and use 109
8.4 Handover services 111
8.4.1 MAP_PREPARE_HANDOVER service 111
8.4.1.1 Definition 111
8.4.1.2 Service primitives 111
8.4.1.3 Parameter use 111
8.4.2 MAP_SEND_END_SIGNAL service 114
8.4.2.1 Definition 114
8.4.2.2 Service primitives 114

8.4.2.3 Parameter use ... 114
8.4.3 MAP_PROCESS_ACCESS_SIGNALLING service ... 115
8.4.3.1 Definition ... 115
8.4.3.2 Service primitives ... 115
8.4.3.3 Parameter use ... 115
8.4.4 MAP_FORWARD_ACCESS_SIGNALLING service ... 115
8.4.4.1 Definition ... 115
8.4.4.2 Service primitives ... 116
8.4.4.3 Parameter use ... 116
8.4.5 MAP_PREPARE_SUBSEQUENT_HANDOVER service ... 117
8.4.5.1 Definition ... 117
8.4.5.2 Service primitives ... 117
8.4.5.3 Parameter use ... 117
8.4.6 MAP_ALLOCATE_HANDOVER_NUMBER service ... 118
8.4.6.1 Definition ... 118
8.4.6.2 Service primitives ... 118
8.4.6.3 Parameter use ... 118
8.4.7 MAP_SEND_HANDOVER_REPORT service ... 119
8.4.7.1 Definition ... 119
8.4.7.2 Service primitives ... 119
8.4.7.3 Parameter use ... 119
8.5 Authentication management services ... 119
8.5.1 MAP_AUTHENTICATE service ... 119
8.5.1.1 Definition ... 119
8.5.1.2 Service primitives ... 120
8.5.1.3 Parameter use ... 120
8.5.2 MAP_SEND_AUTHENTICATION_INFO service ... 120
8.5.2.1 Definition ... 120
8.5.2.2 Service primitives ... 121
8.5.2.3 Parameter use ... 121
8.5.3 MAP_AUTHENTICATION_FAILURE_REPORT service ... 122
8.5.3.1 Definition ... 122
8.5.3.2 Service primitives ... 122
8.5.3.3 Parameter use ... 122
8.6 Security management services ... 123
8.6.1 MAP_SET_CIPHERING_MODE service ... 123
8.6.1.1 Definitions ... 123
8.6.1.2 Service primitives ... 123
8.6.1.3 Parameter use ... 123
8.7 International mobile equipment identities management services ... 123
8.7.1 MAP_CHECK_IMEI service ... 123
8.7.1.1 Definition ... 123
8.7.1.2 Service primitives ... 123
8.7.1.3 Parameter use ... 124
8.7.2 MAP_OBTAIN_IMEI service ... 124
8.7.2.1 Definition ... 124
8.7.2.2 Service primitives ... 124
8.7.2.3 Parameter use ... 125
8.8 Subscriber management services ... 125
8.8.1 MAP-INSERT-SUBSCRIBER-DATA service ... 125
8.8.1.1 Definition ... 125
8.8.1.2 Service primitives ... 126
8.8.1.3 Parameter use ... 126
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 ... 135
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 ... 139

8.9.1.3 Parameter use 139
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 140
8.10.1.3 Parameter definition and use 140
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 143
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 144
8.11.3.1 Definition 144
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 147
9.1 Subscriber tracing services 147
9.1.1 MAP-ACTIVATE-TRACE-MODE service 147
9.1.1.1 Definition 147
9.1.1.2 Service primitives 147
9.1.1.3 Parameter use 148
9.1.2 MAP-DEACTIVATE-TRACE-MODE service 148
9.1.2.1 Definition 148
9.1.2.2 Service primitives 148
9.1.2.3 Parameter use 149
9.1.3 MAP-TRACE-SUBSCRIBER-ACTIVITY service 149
9.1.3.1 Definition 149
9.1.3.2 Service primitives 149
9.1.3.3 Parameter use 149
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 150
9.2.1.3 Parameter use 150

10 Call handling services 150
10.1 MAP_SEND_ROUTING_INFORMATION service 150

10.1.1 Definition 150
10.1.2 Service primitives 151
10.1.3 Parameter use 151
10.2 MAP_PROVIDE_ROAMING_NUMBER service 155
10.2.1 Definition 155
10.2.2 Service primitives 156
10.2.3 Parameter use 156
10.3 MAP_RESUME_CALL_HANDLING service 158
10.3.1 Definition 158
10.3.2 Service primitives 158
10.3.3 Parameter use 158
10.4 MAP_PREPARE_GROUP_CALL service 160
10.4.1 Definition 160
10.4.2 Service primitives 160
10.4.3 Parameter definitions and use 160
10.5 MAP_PROCESS_GROUP CALL_SIGNALLING service 161
10.5.1 Definitions 161
10.5.2 Service primitives 161
10.5.3 Parameter definitions and use 161
10.6 MAP_FORWARD_GROUP_CALL_SIGNALLING service 162
10.6.1 Definitions 162
10.6.2 Service primitives 162
10.6.3 Parameter definitions and use 162
10.7 MAP_SEND_GROUP_CALL_END_SIGNAL service 163
10.7.1 Definitions 163
10.7.2 Service primitives 163
10.7.3 Parameter definitions and use 163
10.8 MAP_Provide_SIWFS_Number 163
10.8.1 Definition 163
10.8.2 Service primitive 164
10.8.3 Parameter use 164
10.9 MAP_SIWFS_Signalling_Modify 165
10.9.1 Definition 165
10.9.2 Service primitive 165
10.9.3 Parameter use 165
10.10 MAP_SET_REPORTING_STATE service 166
10.10.1 Definition 166
10.10.2 Service primitives 166
10.10.3 Parameter use 166
10.11 MAP_STATUS_REPORT service 167
10.11.1 Definition 167
10.11.2 Service primitives 167
10.11.3 Parameter use 167
10.12 MAP_REMOTE_USER_FREE service 168
10.12.1 Definition 168
10.12.2 Service primitives 168
10.12.3 Parameter use 168
10.13 MAP_IST_ALERT service 169
10.13.1 Definition 169
10.13.2 Service primitives 169
10.13.3 Parameter use 169
10.14 MAP_IST_COMMAND service 170
10.14.1 Definition 170
10.14.2 Service primitives 170
10.14.3 Parameter use 170

11 Supplementary services related services 171
11.1 MAP_REGISTER_SS service 171
11.1.1 Definition 171
11.1.2 Service primitives 171
11.1.3 Parameter use 171
11.2 MAP_ERASE_SS service 172

11.2.1 Definition 172
11.2.2 Service primitives 172
11.2.3 Parameter use 173
11.3 MAP_ACTIVATE_SS service 173
11.3.1 Definition 173
11.3.2 Service primitives 174
11.3.3 Parameter use 174
11.4 MAP_DEACTIVATE_SS service 175
11.4.1 Definitions 175
11.4.2 Service primitives 175
11.4.3 Parameter use 175
11.5 MAP_INTERROGATE_SS service 176
11.5.1 Definitions 176
11.5.2 Service primitives 176
11.5.3 Parameter use 177
11.6 MAP_INVOKE_SS service 178
11.6.1 Definitions 178
11.6.2 Service primitives 178
11.6.3 Parameter use 179
11.7 MAP_REGISTER_PASSWORD service 179
11.7.1 Definitions 179
11.7.2 Service primitives 180
11.7.3 Parameter use 180
11.8 MAP_GET_PASSWORD service 180
11.8.1 Definitions 180
11.8.2 Service primitives 181
11.8.3 Parameter use 181
11.9 MAP_PROCESS_UNSTRUCTURED_SS_REQUEST service 181
11.9.1 Definitions 181
11.9.2 Service primitives 181
11.9.3 Parameter use 182
11.10 MAP_UNSTRUCTURED_SS_REQUEST service 182
11.10.1 Definitions 182
11.10.2 Service primitives 183
11.10.3 Parameter use 183
11.11 MAP_UNSTRUCTURED_SS_NOTIFY service 184
11.11.1 Definitions 184
11.11.2 Service primitives 184
11.11.3 Parameter use 184
11.12 MAP_SS_INVOCATION_NOTIFY 185
11.12.1 Definition 185
11.12.2 Service primitives 185
11.12.3 Parameter use 185
11.13 MAP_REGISTER_CC_ENTRY service 185
11.13.1 Definition 185
11.13.2 Service primitives 186
11.13.3 Parameter use 186
11.14 MAP_ERASE_CC_ENTRY service 187
11.14.1 Definition 187
11.14.2 Service primitives 187
11.14.3 Parameter use 187

12 Short message service management services 188
12.1 MAP-SEND-ROUTING-INFO-FOR-SM service 188
12.1.1 Definition 188
12.1.2 Service primitives 188
12.1.3 Parameter use 188
12.2 MAP-MO-FORWARD-SHORT-MESSAGE service 190
12.2.1 Definition 190
12.2.2 Service primitives 190
12.2.3 Parameter use 190
12.3 MAP-REPORT-SM-DELIVERY-STATUS service 191

12.3.1 Definition 191
12.3.2 Service primitives 191
12.3.3 Parameter use 191
12.4 MAP-READY-FOR-SM service 192
12.4.1 Definition 192
12.4.2 Service primitives 193
12.4.3 Parameter use 193
12.5 MAP-ALERT-SERVICE-CENTRE service 193
12.5.1 Definition 193
12.5.2 Service primitives 194
12.5.3 Parameter use 194
12.6 MAP-INFORM-SERVICE-CENTRE service 194
12.6.1 Definition 194
12.6.2 Service primitives 195
12.6.3 Parameter use 195
12.7 MAP-SEND-INFO-FOR-MT-SMS service 195
12.7.1 Definition 195
12.7.2 Service primitives 195
12.7.3 Parameter use 195
12.8 MAP-SEND-INFO-FOR-MO-SMS service 196
12.8.1 Definition 196
12.8.2 Service primitives 196
12.8.3 Parameter use 196
12.9 MAP-MT-FORWARD-SHORT-MESSAGE service 197
12.9.1 Definition 197
12.9.2 Service primitives 197
12.9.3 Parameter use 197

13 Network-Requested PDP Context Activation services 199
13.1 MAP_SEND_ROUTING_INFO_FOR_GPRS service 199
13.1.1 Definition 199
13.1.2 Service primitives 199
13.1.3 Parameter definition and use 199
13.2 MAP_FAILURE_REPORT service 200
13.2.1 Definition 200
13.2.2 Service primitives 200
13.2.3 Parameter definition and use 200
13.3 MAP_NOTE_MS_PRESENT_FOR_GPRS service 201
13.3.1 Definition 201
13.3.2 Service primitives 201
13.3.3 Parameter definition and use 201

13A Location Service Management Services 202
13A.1 MAP-SEND-ROUTING-INFO-FOR-LCS Service 202
13A.1.1 Definition 202
13A.1.2 Service Primitives 202
13A.1.3 Parameter Use 202
13A.2 MAP-PROVIDE-SUBSCRIBER-LOCATION Service 203
13A.2.1 Definition 203
13A.2.2 Service Primitives 203
13A.2.3 Parameter Definition and Use 203
13A.3 MAP-SUBSCRIBER-LOCATION-REPORT Service 205
13A.3.1 Definition 205
13A.3.2 Service Primitives 205
13A.3.3 Parameter Definition and Use 205

14 General 207
14.1 Overview 207
14.2 Underlying services 207
14.3 Model 207
14.4 Conventions 207

15 Elements of procedure 208

15.1 Dialogue establishment 208
15.1.1 Handling of unknown operations 208
15.1.2 Receipt of a MAP-OPEN request primitive 209
15.1.3 Receipt of a TC-BEGIN indication 209
15.1.4 Receipt of a MAP-OPEN response 211
15.1.5 Receipt of the first TC-CONTINUE ind 212
15.1.6 Receipt of a TC-END ind 212
15.1.7 Receipt of a TC-U-ABORT ind 212
15.1.8 Receipt of a TC-P-ABORT ind 212
15.2 Dialogue continuation 213
15.2.1 Sending entity 213
15.2.2 Receiving entity 213
15.3 Dialogue termination 213
15.3.1 Receipt of a MAP-CLOSE request 213
15.3.2 Receipt of a TC-END indication 213
15.4 User Abort 213
15.4.1 MAP-U-ABORT request 213
15.4.2 TC-U-ABORT ind 214
15.5 Provider Abort 214
15.5.1 MAP PM error situation 214
15.5.2 TC-P-ABORT ind 214
15.5.3 TC-U-ABORT ind 214
15.6 Procedures for MAP specific services 214
15.6.1 Service invocation 215
15.6.2 Service invocation receipt 215
15.6.3 Service response 216
15.6.4 Receipt of a response 216
15.6.4.1 Receipt of a TC-RESULT-NL indication 216
15.6.4.2 Receipt of a TC-RESULT-L indication 217
15.6.4.3 Receipt of a TC-U-ERROR indication 217
15.6.4.4 Receipt of a TC-INVOKE indication 217
15.6.4.5 Receipt of a TC-U-REJECT indication 218
15.6.4.6 Receipt of a TC-L-REJECT indication 218
15.6.4.7 Receipt of a TC-L-CANCEL indication 218
15.6.4.8 Receipt of a TC-NOTICE indication 219
15.6.5 Other events 219
15.6.5.1 Receipt of a TC-U-REJECT 219
15.6.5.2 Receipt of a TC-R-REJECT indication 219
15.6.5.3 Receipt of a TC-L-REJECT indication 219
15.6.6 Parameter checks 219
15.6.7 Returning state machines to idle 220
15.6.8 Load control 220

16 Mapping on to TC services 220
16.1 Dialogue control 220
16.1.1 Directly mapped parameters 220
16.1.2 Use of other parameters of dialogue handling primitives 220
16.1.2.1 Dialogue Id 220
16.1.2.2 Application-context-name 221
16.1.2.3 User information 221
16.1.2.4 Component present 221
16.1.2.5 Termination 221
16.1.2.6 P-Abort-Cause 221
16.1.2.7 Quality of service 221
16.2 Service specific procedures 222
16.2.1 Directly mapped parameters 222
16.2.2 Use of other parameters of component handling primitives 222
16.2.2.1 Dialogue Id 222
16.2.2.2 Class 222
16.2.2.3 Linked Id 222
16.2.2.4 Operation 223
16.2.2.5 Error 224

16.2.2.6 Parameters 224
16.2.2.7 Time out 224
16.2.2.8 Last component 224
16.2.2.9 Problem code 224
16.2.2.9.1 Mapping to MAP User Error 224
16.2.2.9.2 Mapping to MAP Provider Error parameter 225
16.2.2.9.3 Mapping to diagnostic parameter 225
16.3 SDL descriptions 226

17 Abstract syntax of the MAP protocol 251
17.1 General 251
17.1.1 Encoding rules 251
17.1.2 Use of TC 251
17.1.2.1 Use of Global Operation and Error codes defined outside MAP 252
17.1.3 Use of information elements defined outside MAP 252
17.1.4 Compatibility considerations 253
17.1.5 Structure of the Abstract Syntax of MAP 253
17.1.6 Application Contexts 255
17.2 Operation packages 256
17.2.1 General aspects 256
17.2.2 Packages specifications 257
17.2.2.1 Location updating 257
17.2.2.2 Location cancellation 258
17.2.2.3 Roaming number enquiry 258
17.2.2.4 Information retrieval 258
17.2.2.5 Inter-VLR information retrieval 259
17.2.2.6 IMSI retrieval 259
17.2.2.7 Call control transfer 259
17.2.2.8 - 17.2.2.9 Void 259
17.2.2.10 Interrogation 259
17.2.2.11 Void 260
17.2.2.12 Handover Control 260
17.2.2.13 Subscriber Data management stand alone 260
17.2.2.14 Equipment management 260
17.2.2.15 Subscriber data management 260
17.2.2.16 Location register restart 261
17.2.2.17 Tracing stand-alone 261
17.2.2.18 Functional SS handling 261
17.2.2.19 Tracing 261
17.2.2.20 Binding 261
17.2.2.21 Unstructured SS handling 262
17.2.2.22 MO Short message relay services 262
17.2.2.23 Short message gateway services 262
17.2.2.24 MT Short message relay services 263
17.2.2.25 Void 263
17.2.2.26 Message waiting data management 263
17.2.2.27 Alerting 263
17.2.2.28 Data restoration 263
17.2.2.29 Purging 264
17.2.2.30 Subscriber information enquiry 264
17.2.2.31 Any time information enquiry 264
17.2.2.32 Group Call Control 264
17.2.2.33 Provide SIWFS number 264
17.2.2.34 SIWFS Signalling Modify 265
17.2.2.35 Gprs location updating 265
17.2.2.36 Gprs Interrogation 265
17.2.2.37 Failure reporting 265
17.2.2.38 GPRS notifying 265
17.2.2.39 Supplementary Service invocation notification 266
17.2.2.40 Set Reporting State 266
17.2.2.41 Status Report 266
17.2.2.42 Remote User Free 266

17.2.2.43 Call Completion 266
17.2.2.44 Location service gateway services 266
17.2.2.45 Location service enquiry 267
17.2.2.45A Location service reporting 267
17.2.2.46 Void 267
17.2.2.47 Void 267
17.2.2.48 Void 267
17.2.2.49 IST Alerting 267
17.2.2.50 Service Termination 267
17.2.2.51 Mobility Management event notification 268
17.2.2.52 Any time information handling 268
17.2.2.53 Subscriber Data modification notification 268
17.2.2.54 Authentication Failure Report 268
17.3 Application contexts 268
17.3.1 General aspects 268
17.3.2 Application context definitions 269
17.3.2.1 Void 269
17.3.2.2 Location Updating 269
17.3.2.3 Location Cancellation 270
17.3.2.4 Roaming number enquiry 270
17.3.2.5 Void 270
17.3.2.6 Location Information Retrieval 270
17.3.2.7 Call control transfer 271
17.3.2.8 - 17.3.2.10 Void 271
17.3.2.11 Location registers restart 271
17.3.2.12 Handover control 271
17.3.2.13 IMSI Retrieval 271
17.3.2.14 Equipment Management 272
17.3.2.15 Information retrieval 272
17.3.2.16 Inter-VLR information retrieval 272
17.3.2.17 Stand Alone Subscriber Data Management 273
17.3.2.18 Tracing 273
17.3.2.19 Network functional SS handling 273
17.3.2.20 Network unstructured SS handling 274
17.3.2.21 Short Message Gateway 274
17.3.2.22 Mobile originating Short Message Relay 274
17.3.2.23 Void 275
17.3.2.24 Short message alert 275
17.3.2.25 Short message waiting data management 275
17.3.2.26 Mobile terminating Short Message Relay 275
17.3.2.27 MS purging 276
17.3.2.28 Subscriber information enquiry 276
17.3.2.29 Any time information enquiry 276
17.3.2.30 Group Call Control 276
17.3.2.31 Provide SIWFS Number 276
17.3.2.32 Gprs Location Updating 277
17.3.2.33 Gprs Location Information Retreival 277
17.3.2.34 Failure Reporting 277
17.3.2.35 GPRS Notifying 277
17.3.2.36 Supplementary Service invocation notification 277
17.3.2.37 Reporting 278
17.3.2.38 Call Completion 278
17.3.2.39 Location Service Gateway 278
17.3.2.40 Location Service Enquiry 278
17.3.2.41 Void 279
17.3.2.42 Void 279
17.3.2.43 Void 279
17.3.2.44 IST Alerting 279
17.3.2.45 Service Termination 279
17.3.2.46 Mobility Management event notification 279
17.3.2.47 Any time information handling 279
17.3.2.48 Subscriber Data modification notification 280

17.3.2.49 Authentication Failure Report 280
17.3.3 ASN.1 Module for application-context-names 280
17.4 MAP Dialogue Information 282
17.5 MAP operation and error codes 284
17.6 MAP operation and error types 290
17.6.1 Mobile Service Operations 290
17.6.2 Operation and Maintenance Operations 297
17.6.3 Call Handling Operations 298
17.6.4 Supplementary service operations 301
17.6.5 Short message service operations 305
17.6.6 Errors 307
17.6.7 Group Call operations 313
17.6.8 Location service operations 314
17.7 MAP constants and data types 315
17.7.1 Mobile Service data types 315
17.7.2 Operation and maintenance data types 343
17.7.3 Call handling data types 344
17.7.4 Supplementary service data types 350
17.7.5 Supplementary service codes 354
17.7.6 Short message data types 357
17.7.7 Error data types 360
17.7.8 Common data types 365
17.7.9 Teleservice Codes 372
17.7.10 Bearer Service Codes 374
17.7.11 Extension data types 375
17.7.12 Group Call data types 376
17.7.13 Location service data types 378

18 General on MAP user procedures 384
18.1 Introduction 384
18.2 Common aspects of user procedure descriptions 384
18.2.1 General conventions 384
18.2.2 Naming conventions 384
18.2.3 Convention on primitives parameters 385
18.2.3.1 Open service 385
18.2.3.2 Close service 386
18.2.4 Version handling at dialogue establishment 386
18.2.4.1 Behaviour at the initiating side 386
18.2.4.2 Behaviour at the responding side 386
18.2.5 Abort Handling 386
18.2.6 SDL conventions 386
18.3 Interaction between MAP Provider and MAP Users 387

19 Mobility procedures 387
19.1 Location management Procedures 387
19.1.1 Location updating 390
19.1.1.1 General 390
19.1.1.3 Detailed procedure in the VLR 395
19.1.1.4 Detailed procedure in the HLR 404
19.1.1.5 Send Identification 411
19.1.1.5.1 General 411
19.1.1.5.2 Detailed procedure in the VLR 411
19.1.1.5.3 Detailed procedure in the PVLR 411
19.1.1.6 Process Update Location VLR 416
19.1.1.8 Detailed procedure in the SGSN 418
19.1.2 Location Cancellation 421
19.1.2.1 General 421
19.1.2.2 Detailed procedure in the HLR 421
19.1.2.3 Detailed procedure in the VLR 422
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 ... 433
19.2.1 General ... 434
19.2.2 Handover procedure in MSC-A ... 437
19.2.2.1 Basic handover ... 437
19.2.2.2 Handling of access signalling ... 437
19.2.2.3 Other procedures in stable handover situation ... 438
19.2.2.4 Subsequent handover ... 438
19.2.2.5 SDL Diagrams ... 438
19.2.3 Handover procedure in MSC-B ... 451
19.2.3.1 Basic handover ... 452
19.2.3.2 Allocation of handover number ... 452
19.2.3.3 Handling of access signalling ... 452
19.2.3.4 Other procedures in stable handover situation ... 452
19.2.3.5 Subsequent handover ... 452
19.2.3.6 SDL Diagrams ... 452
19.2.4 Handover error handling macro ... 465
19.2.5 Handover procedure in VLR ... 467
19.2.5.1 Allocation of handover number ... 467
19.2.5.2 SDL Diagrams ... 467
19.3 Fault recovery procedures ... 470
19.3.1 VLR fault recovery procedures ... 470
19.3.2 HLR fault recovery procedures ... 472
19.3.3 VLR restoration: the restore data procedure in the HLR ... 480
19.4 Macro Insert_Subs_Data_Framed_HLR ... 482
19.5 Mobility Management Event notification procedure ... 485
19.5.1 General ... 485
19.5.2 Process in the VLR ... 485
19.5.3 Process in the gsmSCF ... 487

20 Operation and maintenance procedures ... 489
20.1 General ... 489
20.1.1 Tracing Co-ordinator for the VLR ... 489
20.1.2 Subscriber Data Management Co-ordinator for the VLR ... 491
20.1.3 Tracing Co-ordinator for the SGSN ... 493
20.1.4 Subscriber Data Management Co-ordinator for the SGSN ... 495
20.2 Tracing procedures ... 497
20.2.1 Procedures in the HLR ... 500
20.2.1.1 Subscriber tracing activation procedure ... 500
20.2.1.2 Subscriber tracing deactivation procedure ... 505
20.2.2 Procedures in the VLR ... 510
20.2.2.1 Subscriber tracing activation procedure ... 510
20.2.2.2 Subscriber tracing deactivation procedure ... 512
20.2.2.3 Subscriber tracing procedure ... 514
20.2.3 Procedures in the MSC ... 514
20.2.3.1 Subscriber tracing procedure ... 514
20.2.4 Procedures in the SGSN ... 514
20.2.4.1 Subscriber tracing activation procedure ... 514
20.2.4.2 Subscriber tracing deactivation procedure in SGSN ... 514
20.3 Subscriber data management procedures ... 517
20.3.1 Procedures in the HLR ... 518
20.3.1.1 Subscriber deletion procedure ... 518
20.3.1.2 Subscriber data modification procedure ... 520
20.3.2 Procedures in the VLR ... 525
20.3.2.1 Subscriber deletion procedure ... 525
20.3.2.2 Subscriber data modification procedure ... 525

20.3.3 Procedures in the SGSN 528
20.3.3.1 Subscriber deletion procedure 528
20.3.3.2 Subscriber data modification procedure 528
20.4 Subscriber Identity procedure 531
20.4.1 Subscriber identity procedure in the HLR 531
20.4.2 Subscriber identity procedure in the VLR 533

21 Call handling procedures 535
21.1 General 535
21.2 Retrieval of routing information 535
21.2.1 General 535
21.2.2 Process in the GMSC 537
21.2.3 Procedures in the HLR 541
21.2.4 Process in the VLR to provide a roaming number 546
21.2.5 Process in the VLR to restore subscriber data 548
21.2.6 Process in the VLR to provide subscriber information 550
21.2.7 Process in the HLR for Any Time Interrogation 552
21.2.7.1 Process in the gsmSCF 552
21.2.7.2 Process in the HLR 552
21.2.8 Process in the GMLC for Any Time Interrogation 556
21.2.8.1 Process in the gsmSCF 556
21.2.8.2 Process in the GMLC 556
21.3 Transfer of call handling 559
21.3.1 General 559
21.3.2 Process in the VMSC 559
21.3.3 Process in the GMSC 562
21.4 Inter MSC Group Call Procedures 564
21.4.1 General 564
21.4.2 Process in the Anchor MSC 564
21.4.3 Process in the Relay MSC 570
21.5 Allocation and modifications of resources in an SIWFS 575
21.5.1 General 575
21.5.2 Process in the VMSC 578
21.5.2.1 Allocation of SIWFS resources 578
21.5.2.2 Modification of SIWFS resources initiated by the user 579
21.5.2.3 Modification of SIWFS resources initiated by the SIWFS 580
21.5.3 Process in the SIWFS 587
21.5.3.1 Procedures for allocation of SIWFS resources 587
21.5.3.2 Process for modification of SIWFS resources initiated by the user 588
21.5.3.3 Process for modification of SIWFS resources initiated by the SIWFS 588
21.6 Setting of Reporting State 594
21.6.1 General 594
21.6.2 Process in the HLR for Set Reporting State stand-alone 594
21.6.3 Reporting co-ordinator process in the VLR 597
21.6.4 Process in the VLR to set the reporting state 599
21.7 Status Reporting 602
21.7.1 General 602
21.7.2 Process in the VLR for Status Reporting 602
21.7.3 Process in the HLR for Status Reporting 606
21.8 Remote User Free 611
21.8.1 General 611
21.8.2 Process in the HLR for Remote User Free 611
21.8.3 Process in the VLR for Remote User Free 614
21.9 IST Alert 617
21.9.1 General 617
21.9.2 Procedure in the MSC 617
21.9.3 Procedure in the HLR 619
21.10 IST Command 621
21.10.1 General 621
21.10.2 Procedure in the HLR 621
21.10.3 Procedure in the MSC 623

22 Supplementary services procedures 625
22.1 Functional supplementary service processes 625
22.1.1 Functional supplementary service process co-ordinator for MSC 625
22.1.2 Functional supplementary service process co-ordinator for VLR 627
22.1.3 Functional supplementary service process co-ordinator for HLR 629
22.1.4 Call completion supplementary service process co-ordinator for HLR 632
22.2 Registration procedure 635
22.2.1 General 635
22.2.2 Procedures in the MSC 635
22.2.3 Procedures in the VLR 638
22.2.4 Procedures in the HLR 640
22.3 Erasure procedure 644
22.3.1 General 644
22.3.2 Procedures in the MSC 644
22.3.3 Procedures in the VLR 645
22.3.4 Procedures in the HLR 645
22.4 Activation procedure 645
22.4.1 General 645
22.4.2 Procedures in the MSC 646
22.4.3 Procedures in the VLR 648
22.4.4 Procedures in the HLR 651
22.5 Deactivation procedure 653
22.5.1 General 653
22.5.2 Procedures in the MSC 654
22.5.3 Procedures in the VLR 655
22.5.4 Procedures in the HLR 655
22.6 Interrogation procedure 655
22.6.1 General 655
22.6.2 Procedures in the MSC 656
22.6.3 Procedures in the VLR 656
22.6.4 Procedures in the HLR 660
22.7 Invocation procedure 662
22.7.1 General 662
22.7.2 Procedures in the MSC 662
22.7.3 Procedures in the VLR 666
22.8 Password registration procedure 668
22.8.1 General 668
22.8.2 Procedures in the MSC 669
22.8.3 Procedures in the VLR 669
22.8.4 Procedures in the HLR 669
22.9 Mobile Initiated USSD procedure 672
22.9.1 General 672
22.9.2 Procedures in the MSC 672
22.9.3 Procedures in the VLR 676
22.9.4 Procedures in the HLR 681
22.9.5 Procedures in the gsmSCF/secondary HLR 685
22.10 Network initiated USSD procedure 688
22.10.1 General 688
22.10.2 Procedure in the MSC 688
22.10.3 Procedure in the VLR 693
22.10.4 Procedure in the HLR 700
22.10.5 Procedure in the gsmSCF and secondary HLR 706
22.11 Common macros for clause 22 710
22.11.1 SS Password handling macros 710
22.11.2 SS Error handling macros 713
22.12 Supplementary Service Invocation Notification procedure 719
22.12.1 General 719
22.12.2 Procedures in the MSC 719
22.12.3 Procedures in the gsmSCF 721
22.13 Activation of a CCBS request 723
22.13.1 General 723
22.13.2 Procedure in the VLR 723

22.13.3 Procedure in the HLR 725
22.14 Deactivation of a CCBS request 727
22.14.1 General 727
22.14.2 Procedure in the VLR 727
22.14.3 Procedure in the HLR 729

23 Short message service procedures 731
23.1 General 731
23.1.1 Mobile originated short message service Co-ordinator for the MSC 731
23.1.2 Short message Gateway Co-ordinator for the HLR 733
23.1.3 Mobile originated short message service Co-ordinator for the SGSN 735
23.2 The mobile originated short message transfer procedure 737
23.2.1 Procedure in the serving MSC 738
23.2.2 Procedure in the VLR 744
23.2.3 Procedure in the interworking MSC 747
23.2.4 Procedure in the SGSN 749
23.3 The mobile terminated short message transfer procedure 754
23.3.1 Procedure in the Servicing MSC 756
23.3.2 Procedures in the VLR 765
23.3.3 Procedures in the HLR 769
23.3.4 Procedures in the gateway MSC 778
23.3.5 Procedure in the Servicing SGSN 787
23.4 The Short Message Alert procedure 795
23.4.1 Procedures in the Servicing MSC 797
23.4.2 Procedures in the VLR 799
23.4.2.1 The Mobile Subscriber is present 799
23.4.2.2 The Mobile Equipment has memory available 799
23.4.3 Procedures in the HLR 801
23.4.4 Procedures in the Interworking MSC 804
23.4.5 Procedures in the Servicing SGSN 806
23.4.5.1 The Mobile Subscriber is present 806
23.4.5.2 The Mobile Equipment has memory available 806
23.5 The SM delivery status report procedure 808
23.5.1 Procedures in the HLR 808
23.5.2 Procedures in the gateway MSC 810
23.6 Common procedures for the short message clause 812
23.6.1 The macro Report_SM_Delivery_Stat_HLR 812

24 GPRS process description 814
24.1 General 814
24.1.1 Process in the HLR for Send Routing Information for GPRS 814
24.1.2 Process in the GGSN for Send Routing Information for GPRS 816
24.2.1 Process in the HLR for Failure Report 818
24.2.2 Process in the GGSN for Failure Report 820
24.3.1 Process in the GGSN for Note Ms Present For Gprs 822
24.3.2 Process in the HLR for Note Ms Present For Gprs 824

24A CSE control of subscriber data 826
24A.1 Any Time Subscription Interrogation procedure 826
24A.1.1 General 826
24A.1.2 Process in the gsmSCF 826
24A.1.3 Process in the HLR 826
24A.2 Any Time Modification procedure 829
24A.2.1 General 829
24A.2.2 Process in the gsmSCF 829
24A.2.3 Process in the HLR 829
24A.3 Subscriber Data Modification Notification procedure 832
24A.3.1 General 832
24A.3.2 Processes in the MAP Entities 832
24A.3.2.1 Process in the HLR 832
24A.3.2.2 Process in the gsmSCF 835

25 General macro description 837

25.1 MAP open macros 837
25.1.1 Macro Receive_Open_Ind 837
25.1.2 Macro Receive_Open_Cnf 837
25.2 Macros to check the content of indication and confirmation primitives 842
25.2.1 Macro Check_Indication 842
25.2.2 Macro Check_Confirmation 842
25.3 The page and search macros 845
25.3.1 Macro PAGE_MSC 845
25.3.2 Macro Search_For_MS_MSC 846
25.4 Macros for handling an Access Request 849
25.4.1 Macro Process_Access_Request_MSC 849
25.4.2 Macro Process_Access_Request_VLR 854
25.4.3 Macro Identification Procedure 856
25.5 Authentication macros and processes 861
25.5.1 Macro Authenticate_MSC 861
25.5.2 Macro Authenticate_VLR 861
25.5.3 Process Obtain_Authentication_Sets_VLR 861
25.5.4 Macro Obtain_Authent_Para_VLR 861
25.5.5 Process Obtain_Auth_Sets_HLR 862
25.5.6 Process Obtain_Authent_Para_SGSN 872
25.5.7 Process Authentication_Failure_Report 877
25.5.7.1 General 877
25.5.7.2 Process in the VLR 878
25.5.7.3 Process in the SGSN 879
25.5.7.4 Process in the HLR 880
25.6 IMEI Handling Macros 881
25.6.1 Macro Check_IMEI_MSC 881
25.6.2 Macro Check_IMEI_VLR 881
25.6.3 Process Check_IMEI_EIR 882
25.6.4 Macro Obtain_IMEI_MSC 882
25.6.5 Macro Obtain_IMEI_VLR 882
25.6.6 Process Check_IMEI_SGSN 889
25.7 Insert Subscriber Data Macros 892
25.7.1 Macro Insert_Subs_Data_VLR 892
25.7.2 Process Insert_Subs_Data_Stand_Alone_HLR 894
25.7.3 Macro Wait_for_Insert_Subs_Data_Cnf 900
25.7.4 Process Send_Insert_Subs_Data 902
25.7.5 Macro Insert_Subs_Data_SGSN 902
25.7.6 Macro Wait_for_Insert_GPRS_Subs_Data_Cnf 905
25.8 Request IMSI Macros 907
25.8.1 Macro Obtain_IMSI_MSC 907
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 1148

B.1 Fully Expanded ASN.1 Source of MAP-Protocol/TCAPMessages 1148

B.2 Fully Expanded ASN.1 Source of MAP-DialogueInformation 1249

Annex C: Void 1253

Annex D (informative): Clause mapping table 1254
D.1 Mapping of Clause numbers 1254
Annex E (informative): Change History 1255

Foreword

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

The present document specifies the Mobile Application Part (MAP), the requirements for the signalling system and procedures within the 3GPP system at application level.

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

Version x.y.z

where:

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

1 Scope

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

The present document describes the requirements for the signalling system and the procedures needed at the application level in order to fulfil these signalling needs.

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

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

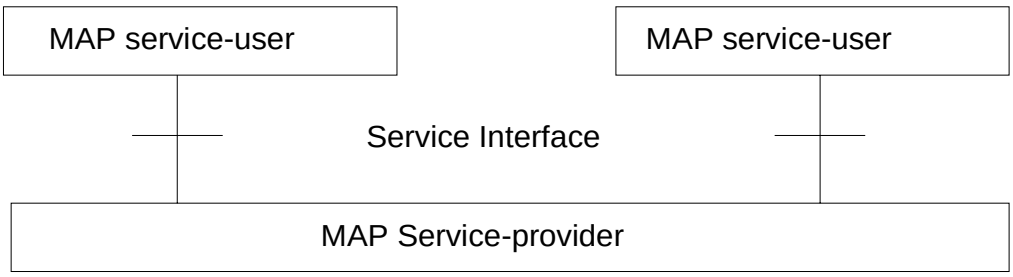


Figure 1.1/1: Modelling principles

Clauses 7 to 13A of the present document describe the MAP services.

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

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

2 References

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

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

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

[2] 3GPP TS 22.001: "Principles of telecommunication services supported by a GSM Public Land Mobile Network (PLMN)".

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

[4] 3GPP TS 22.003: "Teleservices Supported by a GSM Public Land Mobile Network (PLMN)".

[5] 3GPP TS 22.004: "General on Supplementary Services".

[6] GSM 02.09: "Digital cellular telecommunications system (Phase 2+); Security aspects".

[7] 3GPP TS 22.016: "International Mobile station Equipment Identities (IMEI)".

[8] 3GPP TS 22.041: "Operator Determined Barring".

[9] 3GPP TS 22.081: "Line identification supplementary services - Stage 1".

[10] 3GPP TS 22.082: "Call Forwarding (CF) supplementary services - Stage 1".

[11] 3GPP TS 22.083: "Call Waiting (CW) and Call Hold (HOLD) Supplementary Services - Stage 1".

[12] 3GPP TS 22.084: "Multi Party (MPTY) Supplementary Services - Stage 1".

[13] 3GPP TS 22.085: "Closed User Group (CUG) supplementary services - Stage 1".

[14] 3GPP TS 22.086: "Advice of charge (AoC) Supplementary Services - Stage 1".

[15] 3GPP TS 22.088: "Call Barring (CB) supplementary services - Stage 1".

[16] 3GPP TS 22.090: "Unstructured Supplementary Service Data (USSD); - Stage 1".

[17] 3GPP TS 23.003: "Numbering, addressing and identification".

[18] GSM 03.04: "Digital cellular telecommunications system (Phase 2+); Signalling requirements relating to routeing of calls to mobile subscribers".

[19] 3GPP TS 23.007: "Restoration procedures".

[20] 3GPP TS 23.008: "Organisation of subscriber data".

[21] 3GPP TS 23.009: "Handover procedures".

[22] 3GPP TS 23.011: "Technical realization of Supplementary Services - General Aspects".

[23] 3GPP TS 23.012: "Location registration procedures".

[24] GSM 03.20: "Digital cellular telecommunications system (Phase 2+); Security related network functions".

[25] 3GPP TS 23.038: "Alphabets and language".

[26] 3GPP TS 23.040: "Technical realization of the Short Message Service (SMS) Point to Point (PP)".

[26a] 3GPP TS 23.171: "Location Services (LCS); Functional Description; Stage 2".

[27] 3GPP TS 23.081: "Line Identification Supplementary Services - Stage 2".

[28] 3GPP TS 23.082: "Call Forwarding (CF) Supplementary Services - Stage 2".

[29] 3GPP TS 23.083: "Call Waiting (CW) and Call Hold (HOLD) Supplementary Services - Stage 2".

[30] 3GPP TS 23.084: "Multi Party (MPTY) Supplementary Services - Stage 2".

[31] 3GPP TS 23.085: "Closed User Group (CUG) Supplementary Services - Stage 2".

[32] 3GPP TS 23.086: "Advice of Charge (AoC) Supplementary Services - Stage 2".

[33] 3GPP TS 23.088: "Call Barring (CB) Supplementary Services - Stage 2".

[34] 3GPP TS 23.090: "Unstructured Supplementary Services Data (USSD) - Stage 2".

[35] 3GPP TS 24.008: "Mobile Radio Interface Layer 3 specification; Core Network Protocols - Stage 3".

[36] 3GPP TS 24.010: "Mobile radio interface layer 3 Supplementary Services specification - General aspects".

[37] 3GPP TS 24.011: "Point-to-Point (PP) Short Message Service (SMS) support on mobile radio interface".

[37a] GSM 04.71: "Digital cellular telecommunications system (Phase 2+); Mobile radio interface layer 3 location services specification".

[38] 3GPP TS 24.080: "Mobile radio interface layer 3 supplementary services specification - Formats and coding".

[39] 3GPP TS 24.081: "Line identification supplementary services - Stage 3".

[40] 3GPP TS 24.082: "Call Forwarding (CF) Supplementary Services - Stage 3".

[41] 3GPP TS 24.083: "Call Waiting (CW) and Call Hold (HOLD) supplementary services - Stage 3".

[42] 3GPP TS 24.084: "Multi Party (MPTY) Supplementary Services - Stage 3".

[43] 3GPP TS 24.085: "Closed User Group (CUG) Supplementary Services - Stage 3".

[44] 3GPP TS 24.086: "Advice of Charge (AoC) Supplementary Services - Stage 3".

[45] 3GPP TS 24.088: "Call Barring (CB) Supplementary Services - Stage 3".

[46] 3GPP TS 24.090: "Unstructured Supplementary Services Data - Stage 3".

[47] GSM 08.02: "Digital cellular telecommunications system (Phase 2+); Base Station System - Mobile-services Switching Centre (BSS - MSC) interface principles".

[48] GSM 08.06: "Digital cellular telecommunications system (Phase 2+); Signalling transport mechanism specification for the Base Station System - Mobile-services Switching Centre (BSS - MSC) interface".

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

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

[49a1] GSM 08.31: "Digital cellular telecommunications system (Phase 2+); Location Services (LCS); Serving Mobile Location Centre (SMLC) – Serving Mobile Location Centre (SMLC); SMLC Peer Protocol (SMLCPP)".

[49b] GSM 08.71: "Digital cellular telecommunications system (Phase 2+); Location Services (LCS); Serving Mobile Location Centre - Base Station System (SMLC - BSS) interface Layer 3 specification".

[50] GSM 09.01: "Digital cellular telecommunications system (Phase 2+); General network interworking scenarios".

[51] 3GPP TS 29.002: "Mobile Application Part (MAP) specification".

[52] GSM 09.03: "Digital cellular telecommunications system (Phase 2+); Signalling requirements on interworking between the Integrated Services Digital Network (ISDN) or Public Switched Telephone Network (PSTN) and the Public Land Mobile Network (PLMN)".

[53] GSM 09.04: "Digital cellular telecommunications system (Phase 2+); Interworking between the Public Land Mobile Network (PLMN) and the Circuit Switched Public Data Network (CSPDN)".

[54] GSM 09.05: "Digital cellular telecommunications system (Phase 2+); Interworking between the Public Land Mobile Network (PLMN) and the Packet Switched Public Data Network (PSPDN) for Packet Assembly/Disassembly facility (PAD) access".

[55] 3GPP TS 29.006: "Interworking between a Public Land Mobile Network (PLMN) and a Packet Switched Public Data Network/Integrated Services Digital Network (PSPDN/ISDN) for the support of Packet Switched data transmission services".

[56] 3GPP TS 29.007: "General requirements on interworking between the Public Land Mobile Network (PLMN) and the Integrated Services Digital Network (ISDN) or Public Switched Telephone Network (PSTN)".

[57] GSM 09.08: "Digital cellular telecommunications system (Phase 2+); Application of the Base Station System Application Part (BSSAP) on the E-interface".

[58] 3GPP TS 29.010: "Information element mapping between Mobile Station - Base Station System and BSS - Mobile-services Switching Centre (MS - BSS - MSC) Signalling procedures and the Mobile Application Part (MAP)".

[59] 3GPP TS 29.011: "Signalling interworking for Supplementary Services".

[59a] GSM 09.31: "Digital cellular telecommunications system (Phase 2+); Location Services (LCS); Base Station System Application Part LCS Extension (BSSAP-LE)".

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

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

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

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

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

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

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

[67] ITU-T Recommendation E.164: "Numbering plan for the ISDN era".

[68] ITU-T Recommendation E.212: "Identification plan for land mobile stations".

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

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

[71] CCITT Recommendation Q.699: "Interworking between the Digital Subscriber Signalling System Layer 3 protocol and the Signalling System No.7 ISDN User part".

[72] ITU-T Recommendation Q.711: "Specifications of Signalling System No.7; Functional description of the Signalling Connection Control Part".

[73] ITU-T Recommendation Q.712: "Definition and function of SCCP messages".

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

[75] ITU-T Recommendation Q.714: "Specifications of Signalling System No.7; Signalling Connection Control Part procedures".

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

[97] 3GPP TS 23.018: "Basic Call Handling".

[98] 3GPP TS 23.078: "Customised Applications for Mobile network Enhanced Logic (CAMEL) Phase 3 - Stage 2".

[99] 3GPP TS 23.079: "Support of Optimal Routeing (SOR) - Stage 2".

[100] GSM 03.68: "Digital cellular telecommunications system (Phase 2+); - Stage 2".

[101] GSM 03.69: "Digital cellular telecommunications system (Phase 2+); - Stage 2".

[102] ANSI T1.113: "Signaling System No. 7 (SS7) - ISDN User Part".

[103] 3GPP TS 23.054 "description for the use of a Shared Inter Working Function (SIWF) - Stage 2".

[104] 3GPP TS 23.060: "General Packet Radio Service (GPRS) Description; Stage 2".

[105] 3GPP TS 29.060: "General Packet Radio Service (GPRS); GPRS Tunnelling Protocol (GTP) across the Gn and Gp Interface".

[106] 3GPP TS 29.018: "General Packet Radio Service (GPRS); Serving GPRS Support Node (SGSN) - Visitors Location Register (VLR); Gs interface layer 3 specification".

[107] 3GPP TS 23.093: "Technical Realization of Completion of Calls to Busy Subscriber (CCBS); Stage 2".

[108] 3GPP TS 23.066: "Support of Mobile Number Portability (MNP); Technical Realisation Stage 2".

[109] ANSI T1.112 (1996): "Telecommunication – Signalling No. 7 – Signaling Connection Control Part (SCCP)".

[110] 3GPP TS 23.116: "Super-Charger Technical Realisation; Stage 2."

[111] ITU-T Recommendation Q.711: "Specifications of Signalling System No.7; Signalling System No. 7 – Functional Description of the Signalling Connection Control Part".

[112] ITU-T Recommendation Q.712: "Specifications of Signalling System No.7; Signalling System No. 7 – Definition and Function of SCCP Messages".

[113] ITU-T Recommendation Q.713: "Specifications of Signalling System No.7; Signalling System No. 7 – SCCP formats and codes".

[114] ITU-T Recommendation Q.714: "Specifications of Signalling System No.7; Signalling System No. 7 – Signalling Connection Control Part Procedures".

[115] ITU-T Recommendation Q.716: "Specifications of Signalling System No.7; Signalling System No. 7 – Signalling Connection Control Part (SCCP) Performance".

[116] ITU-T Q.850, May 1998: "Usage of cause and location in the Digital Subscriber Signalling System No. 1 and the Signalling System No. 7 ISDN User Part".

[117] 3GPP TS 22.135: "Multicall; Service description; Stage 1".

[118] 3GPP TS 23.135: "Multicall supplementary service; Stage 2".

[119] 3GPP TS 24.135: "Multicall supplementary service; Stage 3".

[120] 3GPP TS 25.413: "UTRAN Iu Interface RANAP Signalling".

[121] 3GPP TS 23.032: "Universal Geographical Area Description (GAD)"

3 Abbreviations

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

4 Void

5 Overload and compatibility overview

5.1 Overload control

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

5.1.1 Overload control for MSC (outside MAP)

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

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

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

5.1.2 Overload control for MAP entities

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

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

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

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

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 a proposal that might be changed due to network operator/implementation matters.

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

	Responder = HLR	**Initiating Entity**
Priority high		
	Mobility Management	
	networkLocUp (updateLocation), (restoreData/v2), (sendParameters/v1)	VLR
	gprsLocationUpdate (updateGPRSLocation/v3),	SGSN
	infoRetrieval (sendAuthenticationInfo/v2/v3), (sendParameters/v1)	VLR/SGSN
	istAlerting (istAlert/v3) (purgeMS/v2/v3)	MSC msPurging VLR
	msPurging (purgeMS/v3)	SGSN
	Short Message Service	
	shortMsgGateway (sendRoutingInfoforSM), (reportSM-DeliveryStatus)	GMSC
	mwdMngt VLR/SGSN (readyForSM/v2/v3), (noteSubscriberPresent/v1)	
	Mobile Terminating Traffic	
	locInfoRetrieval (sendRoutingInfo)	GMSC
	anyTimeEnquiry (anyTimeInterrogation)	gsmSCF
	reporting (statusReport)	VLR
	Location Services	
	locationSvcGateway (sendRoutingInfoforLCS/v3)	GMLC
	Subscriber Controlled Inputs (Supplementary Services)	
	networkFunctionalSs (registerSS), (eraseSS), (activateSS), (deactivateSS), (interrogateSS), (registerPassword), (processUnstructuredSS-Data/v1), (beginSubscriberActivity/v1)	VLR
	callCompletion (registerCCEntry), (eraseCCEntry)	VLR
	networkUnstructuredSs (processUnstructuredSS-Request/v2)	VLR
	imsiRetrieval (sendIMSI/v2)	VLR
	gprsLocationInfoRetrieval (sendRoutingInfoForGprs/v3)	GGSN/SGSN
	failureReport (failureReport/v3)	GGSN/SGSN
	authenticationFailureReport (authenticationFailureReport/v3)	VLR/SGSN
Priority low		

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

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

Responder = SGSN	Initiating Entity
Priority high	
Mobility and Location Register Management	
locationCancel	HLR
(cancelLocation v3)	
reset	HLR
(reset)	
subscriberDataMngt	HLR
(insertSubscriberData v3),	
(deleteSubscriberData v3)	
tracing	HLR
(activateTraceMode),	
(deactivateTraceMode)	
Short Message Service	
shortMsgMT-Relay	MSC
(MT-ForwardSM v3)	
(forwardSM v1/v2)	
Network-Requested PDP context activation	
gprsNotify	HLR
(noteMsPresentForGprs v3),	
Priority low	

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

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

Responder = MSC/VLR	Initiating Entity
Priority high	
Handover	
handoverControl (prepareHandover/v2/v3), (performHandover/v1)	MSC
Group call and Broadcast call	
groupCallControl (prepareGroupCall/v3)	MSC
Mobility and Location Register Management	
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
Short Message Service	
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
Mobile Terminating Traffic	
roamingNbEnquiry (provideRoamingNumber)	HLR
callControlTransfer (resumeCallHandling)	MSC
subscriberInfoEnquiry (provideSubscriberInformation)	HLR
reporting (remoteUserFree) (SetReportingState)	HLR
Location Services	
locationSvcEnquiry (provideSubscriberLocation v3)	GMLC
Network-Initiated USSD	
networkUnstructuredSs (unstructuredSS-Request/v2), (unstructuredSS-Notify/v2)	HLR
Priority low	

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

5.1.3 Congestion control for Signalling System No. 7

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

Means that could be applied to achieve the required traffic reductions are described in clauseclauses 5.1.1 and 5.1.2.

5.2 Compatibility

5.2.1 General

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

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

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

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

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

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

5.2.2 Strategy for selecting the Application Context (AC) version

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

5.2.2.1 Proposed method

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

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

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

a) the version shown in table 1 is "version n-1", where 'n' is the highest version existing in this specification; table 2 is not used;

b) the version shown in table 1 is "unknown"; table 2 is used, and maintained as described in clauseclause 5.2.2.2;

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

5.2.2.2 Managing the version look-up table

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

IF the entity number is known:

THEN

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

ELSE

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

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

IF the destination address is known and not timed-out.

THEN

It retrieves the appropriate AC-name and uses it

IF the dialogue is accepted by the peer

THEN

It does not modify the version control table

ELSE (this should never occur)

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

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

ELSE

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

IF the dialogue is accepted by the peer.

THEN

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

ELSE

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

IF the destination node was not known

THEN

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

ELSE

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

5.2.2.3 Optimising the method

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 each subscriber the version of the AC-name used at location updating. Then:

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

  for HLR:

  - Subscriber data modification (stand alone);

  for VLR:

  - Data Restoration.

6 Requirements concerning the use of SCCP and TC

6.1 Use of SCCP

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

MAP supports the following SCCP versions:

- Signalling Connection Control Part , Signalling System no. 7 CCITT (‘Blue Book SCCP’).
- Signalling Connection Control Part, Signalling System no. 7 ITU-T Recommendation (07/96) Q.711 to Q.716 (‘White Book SCCP’). Support of White Book SCCP at the receiving side shall be mandated from 00:01hrs, 1st July 2002(UTC). However, for signalling over the MAP E-interface to support inter-MSC handover/relocation, the support of White Book SCCP shall be mandated with immediate effect.

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

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

6.1.1 SCCP Class

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

6.1.2 Sub-System Number (SSN)

The Application Entities (AEs) defined for MAP consist of several Application Service Elements (ASEs) and are addressed by sub-system numbers (SSNs). The SSNs for MAP are specified in 3GPP TS 23.003 [17].

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

6.1.3 SCCP addressing

6.1.3.1 Introduction

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

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

1) Intra-PLMN addressing

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

2) Inter-PLMN addressing

   a) Called Party Address

      - SSN indicator = 1 (MAP SSN always included);
      - Global title indicator = 0100 (Global title includes translation type, numbering plan, encoding scheme and nature of address indicator);
      - the translation type field will be coded "00000000" (Not used). For call related messages for non-optimal routed calls (as described in 3GPP TS 23.066) directed to another PLMN the translation type field may be coded "10000000" (CRMNP);
      - Routing indicator = 0 (Routing on global title);

   b) Calling Party Address

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

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

1) Intra-PLMN addressing

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

2) Inter-PLMN addressing

   a) Called Party Address

   - SSN indicator = 1 (MAP SSN always included);
   - Global title indicator = 0010 (Global title includes translation type);
   - the Translation Type (TT) field will be coded as follows:

     TT = 9, if IMSI is included;

     TT = 14, if MSISDN is included;

     Or TT = 10, if Network Element is included. (If TT=10, then Number Portability GTT is not invoked, if TT=14, then Number Portability GTT may be invoked).

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

   b) Calling Party Address

   - SSN indicator = 1 (MAP SSNs always included);
   - Point code indicator = 0;
   - Global Title indicator = 0010 (Global title includes translation type);

     TT = 9, if IMSI is included;

     TT = 14, if MSISDN is included;

     Or TT = 10, if Network Element is included. (If TT=10, then Number Portability GTT is not invoked, if TT=14, then Number Portability GTT may be invoked).

   Routing indicator = 0 (Routing on Global Title).

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

- E.212 numbering plan.

  When CCITT or ITU-T SCCP is used, an E.212 number must not be included as Global Title in an SCCP UNITDATA message. The translation of an E.212 number into a Mobile Global Title is applicable in a dialogue initiating VLR, SGSN or GGSN if the routeing information towards the HLR is derived from the subscriber's IMSI. In World Zone 1 when ANSI SCCP is used, the IMSI (E.212 number) is used as a Global Title to address the HLR. When an MS moves from one VLR service area to another, the new VLR may derive the address of the previous VLR from the Location Area Identification provided by the MS in the location registration request. The PLMN where the previous VLR is located is identified by the E.212 numbering plan elements of the Location Area Identification, i.e. the Mobile Country Code (MCC) and the Mobile Network Code (MNC).

- E.214 and E.164 numbering plans.

  When CCITT or ITU-T SCCP is used, only address information belonging to either E.214 or E.164 numbering plan is allowed to be included as Global Title in the Called and Calling Party Address. In World Zone 1 when ANSI SCCP is used, the IMSI (E.212 number) is used as a Global Title to address the HLR.

  If the Calling Party Address associated with the dialogue initiating message contains a Global Title, the sending network entity shall include its E.164 entity number.

When receiving an SCCP UNITDATA message, SCCP shall accept either of the valid numbering plans in the Called Party Address and in the Calling Party Address.

When CCITT or ITU-T SCCP is used and an N-UNITDATA-REQUEST primitive from TC is received, SCCP shall accept an E.164 number or an E.214 number in the Called Address and in the Calling Address. In World Zone 1 when ANSI SCCP is used, the IMSI (E.212 number) is used instead of E.214 number.

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

- the Mobile-services Switching Centre (MSC);
- the Home location Register (HLR);
- the Visitor Location Register (VLR);
- the Gateway Mobile-services Switching Centre (GMSC);
- the GSM Service Control Function (gsmSCF);
- the Interworking Mobile-services Switching Centre (IWMSC);
- the Shared Inter Working Function (SIWF);
- the Serving GPRS Support Node (SGSN);
- the Gateway GPRS Support Node (GGSN);
- the Gateway Mobile Location Centre (GMLC).

6.1.3.2 The Mobile-services Switching Centre (MSC)

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

6.1.3.2.1 MSC interaction during handover or relocation

The address is derived from the target Cell id or from the target RNC id.

6.1.3.2.2 MSC for short message routing

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

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

6.1.3.2.3 MSC for location request routing

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

6.1.3.2.4 MSC for LMU Control

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

6.1.3.3 The Home Location Register (HLR)

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

6.1.3.3.1 During call set-up

When a call is initiated the HLR of the called mobile subscriber will be interrogated to discover the whereabouts of the MS. The addressing required by the SCCP will be derived from the MSISDN dialled by the calling subscriber. The

dialled number will be translated into either an SPC, in the case of communications within a PLMN, or a Global Title if other networks are involved (i.e. if the communication is across a PLMN boundary).

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

6.1.3.3.2 Before location updating completion

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

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

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

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

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

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

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

6.1.3.3.3 After location updating completion

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

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

6.1.3.3.4 VLR restoration

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

6.1.3.3.5 During Network-Requested PDP Context Activation

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

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

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

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

6.1.3.3.6 Before GPRS location updating completion

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

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

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

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

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

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

6.1.3.3.7 After GPRS location updating completion

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

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

6.1.3.3.8 Query for a Location Request

For a location request from an external client, the GMLC needs to address the home HLR of the target MS to obtain the address of the target MS's serving MSC. The GMLC uses either the international E.164 MSISDN, the international 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 as means to route a query to the HLR.

6.1.3.4 The Visitor Location Register (VLR)

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

6.1.3.4.1 Inter-VLR information retrieval

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

6.1.3.4.2 HLR request

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

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

6.1.3.5 The Interworking MSC (IWMSC) for Short Message Service

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

6.1.3.6 The Equipment Identity Register (EIR)

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

6.1.3.7 The Shared Inter Working Function (SIWF)

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

6.1.3.8 The Serving GPRS Support Node (SGSN)

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

When the GMSC initiates dialogues towards the SGSN the SGSN (MAP) SSN (See 3GPP TS 23.003) shall be included in the called party address. The routeing information used by the GMSC is derived from the E.164 SGSN number received as a parameter of the MAP message initiating the forward short message procedure. If the GMSC does not support the GPRS functionality the MSC (MAP) SSN value shall be included in the called party address.

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

6.1.3.9 The Gateway GPRS Support Node (GGSN)

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

6.1.3.10 The Gateway MSC (GMSC) for Short Message Service

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

6.1.3.10A Void

6.1.3.10A.1 Void

6.1.3.10A.2 Void

6.1.3.10B The Gateway Mobile Location Centre (GMLC)

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

6.1.3.11 Summary table

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

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

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/HLR 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 Centre	---	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	---	---	---	---	---

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 routeing information towards the HLR from the Calling Party Address received with the first responding CONTINUE message until the dialogue terminating message is received.
For transactions invoked by the VLR after update location completion, the VLR may derive the information for addressing the HLR from addresses received in the course of the update location procedure (MSISDN or HLR number) or from the IMSI.
When invoking the Restore Data procedure and an authentication information retrieval from the HLR preceding it, the VLR must derive the information for addressing the HLR from the address information received in association with the roaming number request. This may be either the IMSI received as a parameter of the MAP message requesting the Roaming Number or the Calling Party Address associated with the MAP message requesting the Roaming Number.
The gsmSCF shall be addressed using more than one Global Title number. The first Global Title number is used to address a gsmSCF for MAP. The second Global Title number is used to address a gsmSCF for CAP.
For querying the HLR to obtain the VMSC address to support location services, the GMLC has to derive the HLR address from either the MSISDN or IMSI of the target MS. When using the IMSI, 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).

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 Centre		

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

6.2 Use of TC

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

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

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

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

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

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

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

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

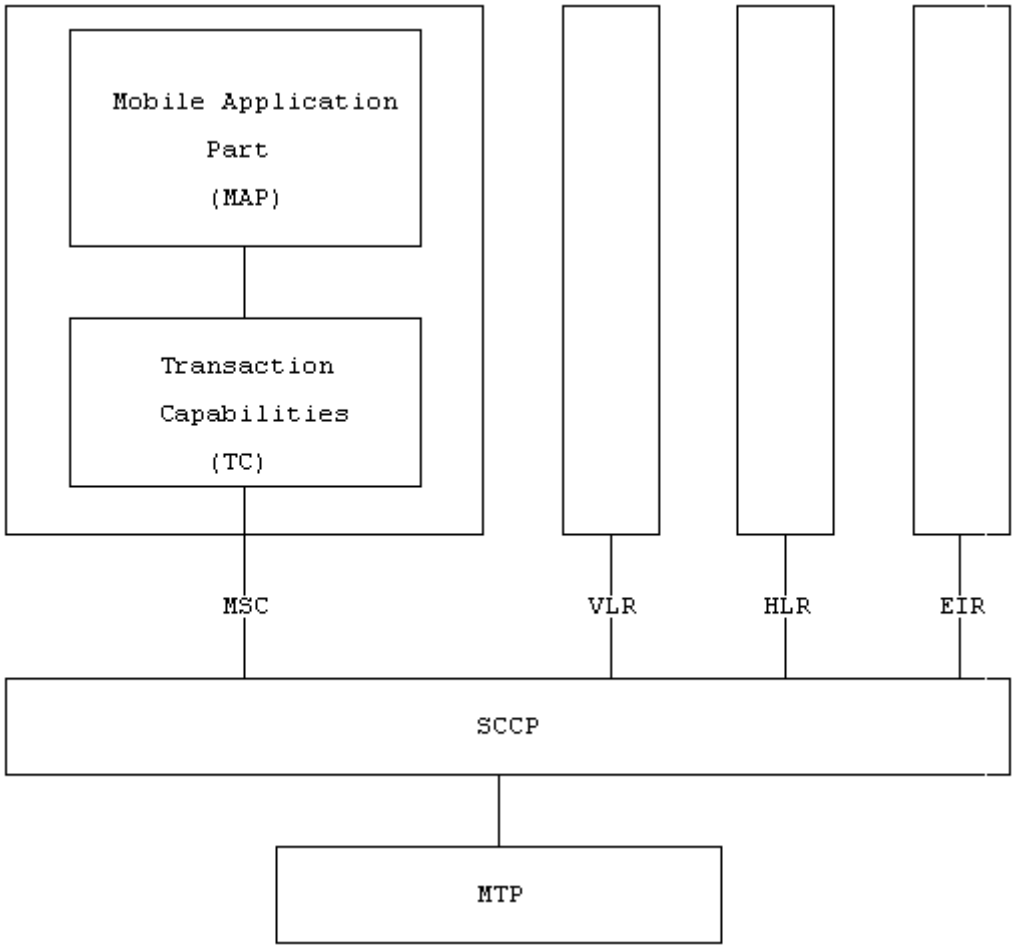


Figure 6.2/1: Facilities for supporting the Mobile Application Part in Signalling System No.7

7 General on MAP services

7.1 Terminology and definitions

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

7.2 Modelling principles

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

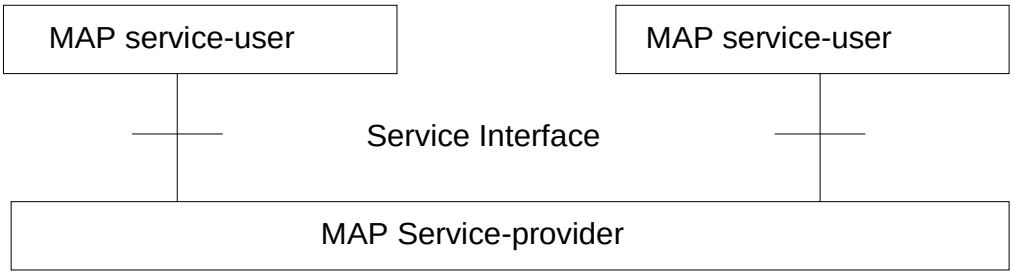


Figure 7.2/1: Modelling principles

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

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

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

MAP-ServicePrimitiveName **type**

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

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

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

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

7.3 Common MAP services

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

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

For these purposes the following common services are defined:

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

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

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

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

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

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

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

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

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

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

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

blank the parameter is not present.

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

7.3.1 MAP-OPEN service

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

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

Application context name:

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

Destination address:

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

Destination-reference:

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

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 (note 1)	Subscriber identity
MAP-UNSTRUCTURED-SS-REQUEST	IMSI (note 2)	Subscriber identity
MAP-UNSTRUCTURED-SS-NOTIFY	IMSI (note 2)	Subscriber identity
MAP-FORWARD-SHORT-MESSAGE	IMSI (note 3)	Subscriber identity
MAP-REGISTER-CC-ENTRY	IMSI	Subscriber identity
MAP-ERASE-CC-ENTRY	IMSI	Subscriber identity

NOTE 1: On the HLR - HLR interface and on the HLR - gsmSCF interface the Destination reference shall be either IMSI or MSISDN.

NOTE 2: On the gsmSCF - HLR interface and on the HLR - HLR interface the Destination reference shall be either IMSI or MSISDN.

NOTE 3: Only when the IMSI and the LMSI are received together from the HLR in the mobile terminated short message transfer.

Originating address:

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

Originating-reference:

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

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-UNSTRUCTURED-SS-REQUEST	ISDN-Address-String (note)	Originated entity address
MAP-UNSTRUCTURED-SS-NOTIFY	ISDN-Address-String (note)	Originated entity address
MAP-REGISTER-CC-ENTRY	ISDN-Address-String	Originated entity address
MAP-ERASE-CC-ENTRY	ISDN-Address-String	Originated entity address

NOTE: The Originating reference may be omitted.

Specific information:

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

Responding address:

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

Result:

This parameter indicates whether the peer accepts the dialogue.

Refuse reason:

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

- Application-context-not-supported;
- Invalid-destination-reference;
- Invalid-originating-reference;
- No-reason-given;
- Remote node not reachable;
- Potential version incompatibility.

7.3.2 MAP-CLOSE service

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

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

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

Release method:

This parameter can take the following two values:

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

Specific information:

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

7.3.3 MAP-DELIMITER service

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

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

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

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

Parameters	Request	Indication

7.3.4 MAP-U-ABORT service

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

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(=)

User reason:

This parameter can take the following values:

- resource limitation (congestion);

  the requested user resource is unavailable due to congestion;

- resource unavailable;

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

- application procedure cancellation;

  the procedure is cancelled for reasons detailed in the diagnostic information parameter;

- procedure error;

  processing of the procedure is terminated for procedural reasons.

Diagnostic information:

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

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	-

Specific information:

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

7.3.5 MAP-P-ABORT service

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

Table 7.3/8: Service-primitives for the MAP-P-ABORT service

Parameters		Indication
Provider reason		M
Source		M

Provider reason:

This parameter indicates the reason for aborting the MAP dialogue:

- provider malfunction;
- supporting dialogue/transaction released;
- resource limitation;
- maintenance activity;
- version incompatibility;
- abnormal MAP dialogue.

Source:

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

- MAP problem;
- TC problem;
- network service problem.

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

Provider reason	Source	Corresponding event
Provider	MAP	Malfunction at MAP level at peer entity
malfunction	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	MAP	Congestion towards MAP peer service-user
limitation	TC	"Resource limitation" received in TC-P-ABORT
Maintenance	MAP	Maintenance at MAP peer service-user
activity	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

7.3.6 MAP-NOTICE service

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

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

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

Parameters	Indication
Problem diagnostic	M

Problem diagnostic:

This parameter can take one of the following values:

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

7.4 Sequencing of services

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

Opening:

The MAP-OPEN service is invoked before any user specific service-primitive is accepted. The sequence may contain none, one or several user specific service-primitives. If no user specific service-primitive is contained

between the MAP-OPEN and the MAP-DELIMITER primitives, then this will correspond to sending an empty Begin message in TC. If more than one user specific service-primitive is included, all are to be sent in the same Begin message. The sequence ends with a MAP-DELIMITER primitive.

Continuing:

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

Closing:

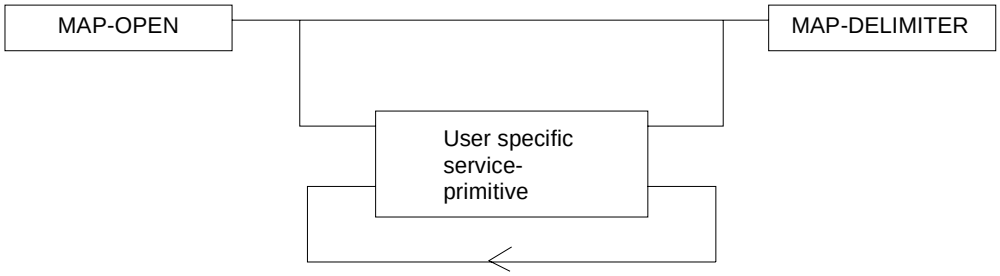
The sequence can only appear after an opening sequence or a continuing sequence. The sequence may contain none, one or several user specific service-primitives if the MAP-CLOSE primitive specifies normal release. If no user specific service-primitive is included, then this will correspond to sending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

Aborting:

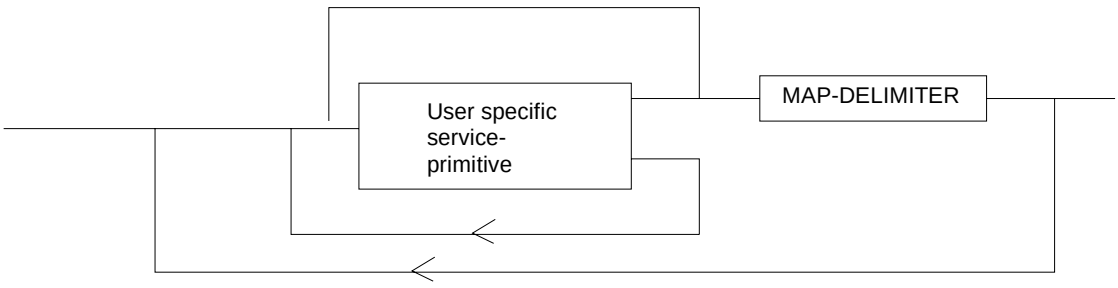
A MAP service-user can issue a MAP-U-ABORT primitive at any time after the MAP dialogue has been opened or as a response to an attempt to open a MAP dialogue.

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

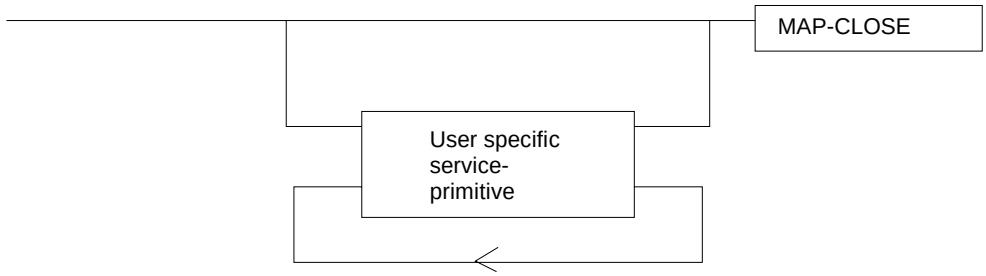
MAP-U-ABORT primitives and MAP-P-ABORT primitives terminate the MAP dialogue.



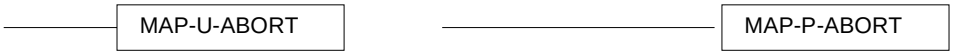
a) Opening



b) Continuing



c) Closing



d) Aborting

Figure 7.4/1: Sequencing of services

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

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

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

7.5 General rules for mapping of services onto TC

7.5.1 Mapping of common services

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

Protocol machine description is given in clauses 14 to 17.

Table 7.5/1: Mapping of common services onto 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
NOTE: Or TC-END if the MAP-CLOSE request has been received before the MAP-DELIMITER request.	

Table 7.5/2: Mapping of TC services onto 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)
NOTE 1: It may not be necessary to present this primitive to the user for MAP version 2 applications. NOTE 2: The mapping depends on whether the TC-U-ABORT indication primitive contains a MAP-abort-PDU from the remote MAP service-provider or a MAP-user-abort-PDU from the remote MAP service-user. NOTE 3: Only if the opening sequence is pending and if the "Abort Reason" in the TC-U-ABORT indication is set to "Application Context Not Supported". NOTE 4: If the "Abort Reason" in the TC-P-ABORT indication is set to a value different from "Incorrect Transaction Portion". NOTE 5: Only if the opening sequence is pending and if the "Abort Reason" in the TC-P-ABORT indication is set to "Incorrect Transaction Portion". NOTE 6: Only if opening sequence is pending.	

7.5.2 Mapping of user specific services

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

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)

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)

Notes to tables 7.5/3 and 7.5/4:

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

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

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

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

7.6 Definition of parameters

Following is an alphabetic list of parameters used in the common MAP-services in clauseclause 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		

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

Absent Subscriber Diagnostic SM	7.6.8.9
Access connection status	7.6.9.3
Access signalling information	7.6.9.5
Additional Absent Subscriber Diagnostic SM	7.6.8.12
Additional Location Estimate	7.6.11.21
Additional number	7.6.2.46
Additional signal info	7.6.9.10
Additional SM Delivery Outcome	7.6.8.11
Age Indicator	7.6.3.72
Alert Reason	7.6.8.8
Alert Reason Indicator	7.6.8.10
Alerting Pattern	7.6.3.44
All GPRS Data	7.6.3.53
All Information Sent	7.6.1.5
AN-apdu	7.6.9.1
APN	7.6.2.42
Authentication set list	7.6.7.1
B-subscriber Address	7.6.2.36
B subscriber Number	7.6.2.48
B subscriber subaddress	7.6.2.49
Basic Service Group	7.6.4.40
Bearer service	7.6.4.38
BSSMAP Service Handover	7.6.6.5
BSSMAP Service Handover List	7.6.6.5A
Call Barring Data	7.6.3.83
Call barring feature	7.6.4.19
Call barring information	7.6.4.18
Call Direction	7.6.5.8
Call Forwarding Data	7.6.3.84
Call Info	7.6.9.9
Call reference	7.6.5.1
Call Termination Indicator	7.6.3.67
Called number	7.6.2.24
Calling number	7.6.2.25
CAMEL Subscription Info	7.6.3.78
CAMEL Subscription Info Withdraw	7.6.3.38
Cancellation Type	7.6.3.52
Category	7.6.3.1
CCBS Feature	7.6.5.8
CCBS Request State	7.6.4.49
Channel Type	7.6.5.9
Chosen Channel	7.6.5.10
Chosen Radio Resource Information	7.6.6.10B
Ciphering mode	7.6.7.7
Cksn	7.6.7.5
CLI Restriction	7.6.4.5
CM service type	7.6.9.2
Complete Data List Included	7.6.3.54
CS Allocation Retention priority	7.6.3.87
CUG feature	7.6.3.26
CUG index	7.6.3.25
CUG info	7.6.3.22
CUG interlock	7.6.3.24
CUG Outgoing Access indicator	7.6.3.8
CUG subscription	7.6.3.23
Invoke Id	7.6.1.1
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
Kc	7.6.7.4
Linked Id	7.6.1.2
LMSI	7.6.2.16
Location Information	7.6.2.30
Location update type	7.6.9.6
Long Forwarded-to Number	7.6.2.22A
Long FTN Supported	7.6.2.22B
Lower Layer Compatibility	7.6.3.42
LSA Information	7.6.3.56
LSA Information Withdraw	7.6.3.58
MC Information	7.6.4.48
MC Subscription Data	7.6.4.47
Mobile Not Reachable Reason	7.6.3.51
Modification request for CSI	7.6.3.81
Modification request for SS Information	7.6.3.82
More Messages To Send	7.6.8.7
MS ISDN	7.6.2.17
MSC number	7.6.2.11
MSIsdn-Alert	7.6.2.29
Multicall Bearer Information	7.6.2.52
Multiple Bearer Requested	7.6.2.53
Multiple Bearer Not Supported	7.6.2.54
MWD status	7.6.8.3
NbrUser	7.6.4.45
Network Access Mode	7.6.3.50
Network node number	7.6.2.43
Network resources	7.6.10.1
Network signal information	7.6.9.8
New password	7.6.4.20
No reply condition timer	7.6.4.7
North American Equal Access preferred Carrier Id	7.6.2.34
Number Portability Status	7.6.5.14
ODB Data	7.6.3.85
ODB General Data	7.6.3.9
ODB HPLMN Specific Data	7.6.3.10
OMC Id	7.6.2.18
Originally dialled number	7.6.2.26
Originating entity number	7.6.2.10
Override Category	7.6.4.4
P-TMSI	7.6.2.47
PDP-Address	7.6.2.45
PDP-Context identifier	7.6.3.55
PDP-Type	7.6.2.44
Pre-paging supported	7.6.5.15
Previous location area Id	7.6.2.4
Protocol Id	7.6.9.7
Provider error	7.6.1.3
QoS-Subscribed	7.6.3.47
Radio Resource Information	7.6.6.10
Radio Resource List	7.6.6.10A
RANAP Service Handover	7.6.6.6
Rand	7.6.7.2
Regional Subscription Data	7.6.3.11
Regional Subscription Response	7.6.3.12

CUG Subscription Flag	7.6.3.37
Current location area Id	7.6.2.6
Current password	7.6.4.21
eMLPP Information	7.6.4.41
Encryption Information	7.6.6.9
Equipment status	7.6.3.2
Extensible Basic Service Group	7.6.3.5
Extensible Bearer service	7.6.3.3
Extensible Call barring feature	7.6.3.21
Extensible Call barring information	7.6.3.20
Extensible Call barring information for CSE	7.6.3.79
Extensible Forwarding feature	7.6.3.16
Extensible Forwarding info	7.6.3.15
Extensible Forwarding information for CSE	7.6.3.80
Extensible Forwarding Options	7.6.3.18
Extensible No reply condition timer	7.6.3.19
Extensible QoS-Subscribed	7.6.3.74
Extensible SS-Data	7.6.3.29
Extensible SS-Info	7.6.3.14
Extensible SS-Status	7.6.3.17
Extensible Teleservice	7.6.3.4
External Signal Information	7.6.9.4
Failure Cause	7.6.7.9
Forwarded-to number	7.6.2.22
Forwarded-to subaddress	7.6.2.23
Forwarding feature	7.6.4.16
Forwarding information	7.6.4.15
Forwarding Options	7.6.4.6
GGSN address	7.6.2.40
GGSN number	7.6.2.41
GMSC CAMEL Subscription Info	7.6.3.34
GPRS enhancements support indicator	7.6.3.73
GPRS Node Indicator	7.6.8.14
GPRS Subscription Data	7.6.3.46
GPRS Subscription Data Withdraw	7.6.3.45
GPRS Support Indicator	7.6.8.15
Group Id	7.6.2.33
GSM bearer capability	7.6.3.6
Guidance information	7.6.4.22
Handover number	7.6.2.21
High Layer Compatibility	7.6.3.43
HLR Id	7.6.2.15
HLR number	7.6.2.13
HO-Number Not Required	7.6.6.7
IMEI	7.6.2.3
IMSI	7.6.2.1
Integrity Protection Information	7.6.6.8
Inter CUG options	7.6.3.27
Intra CUG restrictions	7.6.3.28
Relocation Number List	7.6.2.19A
Requested Info	7.6.3.31
Requested Subscription Info	7.6.3.86
Roaming number	7.6.2.19
Roaming Restricted In SGSN Due To Unsupported Feature	7.6.3.49
Roaming Restriction Due To Unsupported Feature	7.6.3.13
Current Security Context	7.6.7.8
Selected RAB ID	7.6.2.56
Service centre address	7.6.2.27
Serving Cell Id	7.6.2.37
SGSN address	7.6.2.39
SGSN CAMEL Subscription Info	7.6.3.75
SGSN number	7.6.2.38
SIWF Number	7.6.2.35
SoLSA Support Indicator	7.6.3.57
SM Delivery Outcome	7.6.8.6
SM-RP-DA	7.6.8.1
SM-RP-MTI	7.6.8.16
SM-RP-OA	7.6.8.2
SM-RP-PRI	7.6.8.5
SM-RP-SMEA	7.6.8.17
SM-RP-UI	7.6.8.4
Sres	7.6.7.3
SS-Code	7.6.4.1
SS-Data	7.6.4.3
SS-Event	7.6.4.42
SS-Event-Data	7.6.4.43
SS-Info	7.6.4.24
SS-Status	7.6.4.2
Stored location area Id	7.6.2.5
Subscriber State	7.6.3.30
Subscriber Status	7.6.3.7
Super-Charger Supported in HLR	7.6.3.70
Super-Charger Supported in Serving Network Entity	7.6.3.71
Supported CAMEL Phases in VLR	7.6.3.36
Supported CAMEL Phases in SGSN	7.6.3.36A
Supported GAD Shapes	7.6.11.20
Suppress T-CSI	7.6.3.33
Suppression of Announcement	7.6.3.32
Target cell Id	7.6.2.8
Target location area Id	7.6.2.7
Target RNC Id	7.6.2.8A
Target MSC number	7.6.2.12
Teleservice	7.6.4.39
TMSI	7.6.2.2
Trace reference	7.6.10.2
Trace type	7.6.10.3
User error	7.6.1.4
USSD Data Coding Scheme	7.6.4.36
USSD String	7.6.4.37
UU Data	7.6.5.12
UUS CF Interaction	7.6.5.13
VBS Data	7.6.3.40
VGCS Data	7.6.3.39
VLR CAMEL Subscription Info	7.6.3.35
VLR number	7.6.2.14
VPLMN address allowed	7.6.3.48
Zone Code	7.6.2.28

7.6.1 Common parameters

The following set of parameters is used in several MAP service-primitives.

7.6.1.1 Invoke Id

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

7.6.1.2 Linked Id

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

7.6.1.3 Provider error

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

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

7.6.1.4 User error

This parameter can take values as follows:

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

a) Generic error:

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

b) Identification or numbering problem:

- unknown subscriber, i.e. no such subscription exists;
- number changed, i.e. the subscription does not exist for that number any more;

- unknown MSC;
- unidentified subscriber, i.e. if the subscriber is not contained in the database and it has not or cannot be established whether or not a subscription exists;
- unallocated roaming number;
- unknown equipment;
- unknown location area.

c) Subscription problem:

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

d) Handover problem:

- no handover number available, i.e. the VLR cannot allocate a number for handover or cannot allocate the required amount of numbers for relocation;
- subsequent handover failure, i.e. handover to a third MSC failed for some reason;
- target cell outside group call area.

e) Operation and maintenance problem:

- tracing buffer full, i.e. tracing cannot be performed because the tracing capacity is exceeded.

f) Call set-up problem:

- no roaming number available, i.e. a roaming number cannot be allocated because all available numbers are in use;
- absent subscriber, i.e. the subscriber has activated the detach service or the system detects the absence condition. This error may be qualified to indicate whether the subscriber was IMSI detached, in a restricted area or did not respond to paging;
- busy subscriber. This error may be qualified to indicate that the subscriber was busy due to CCBS and that CCBS is possible;
- no subscriber reply;
- forwarding violation, i.e. the call has already been forwarded the maximum number of times that is allowed;
- CUG reject, i.e. the call does not pass a CUG check; additional information may also be given in order to indicate rejection due to e.g. incoming call barred or non-CUG membership;
- call barred. Optionally, additional information may be included for indicating either that the call meets a barring condition set by the subscriber or that the call is barred for operator reasons. In case of barring of Mobile Terminating Short Message, the additional information may indicate a barring condition due to «unauthorised Message Originator»;
- optimal routeing not allowed, i.e. the entity which sends the error does not support optimal routeing, or the HLR will not accept an optimal routeing interrogation from the GMSC, or the call cannot be optimally routed because it would contravene optimal routeing constraints;
- forwarding failed, i.e. the GMSC interrogated the HLR for forwarding information but the HLR returned an error.

g) Supplementary services problem:

- call barred;
- illegal SS operation;
- SS error status;
- SS not available;
- SS subscription violation;
- SS incompatibility;
- negative password check;
- password registration failure;
- Number of Password Attempts;
- USSD Busy;
- Unknown Alphabet;
- short term denial;
- long term denial.

For definition of these errors see 3GPP TS 24.080 [38].

h) Short message problem:

- SM delivery failure with detailed reason as follows:
  - memory capacity exceeded;
  - MS protocol error;
  - MS not equipped;
  - unknown service centre (SC);
  - SC congestion;
  - invalid SME address;
  - subscriber is not an SC subscriber;
  - and possibly detailed diagnostic information, coded as specified in 3GPP TS 23.040 [26], under SMS-SUBMIT-REPORT and SMS-DELIVERY-REPORT. If the SM entity that returns the SM Delivery Failure error includes detailed diagnostic information, it shall be forwarded in the MAP_MO_FORWARD_SHORT_MESSAGE and in the MAP_MT_FORWARD_SHORT_MESSAGE response.
- message waiting list full, i.e. no further SC address can be added to the message waiting list.
- Subscriber busy for MT SMS, i.e. the mobile terminated short message transfer cannot be completed because:
  - another mobile terminated short message transfer is going on and the delivery node does not support message buffering; or
  - another mobile terminated short message transfer is going on and it is not possible to buffer the message for later delivery; or
  - the message was buffered but it is not possible to deliver the message before the expiry of the buffering time defined in 3GPP TS 23.040 [26];

- 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.

i) Location services problem:

- Unauthorized Requesting Network
- Unauthorized LCS Client with detailed reasons as follows:
  - Unauthorized Privacy Class
  - Unauthorized Call Unrelated External Client
  - Unauthorized Call Related External Client
- Privacy override not applicable
- Position method failure with detailed reasons as follows:
  - Congestion
  - Insufficient resources
  - Insufficient Measurement Data
  - Inconsistent Measurement Data
  - Location procedure not completed
  - QoS not attainable
  - Position Method Not Available in Network
  - Position Method Not Available in Location Area
  - Unknown or unreachable LCS Client

7.6.1.5 All Information Sent

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

7.6.2 Numbering and identification parameters

7.6.2.1 IMSI

This parameter is the International Mobile Subscriber Identity defined in 3GPP TS 23.003 [17].

7.6.2.2 TMSI

This parameter is the Temporary Mobile Subscriber Identity defined in 3GPP TS 23.003 [17].

7.6.2.3 IMEI

This parameter is the International Mobile Equipment Identity defined in 3GPP TS 23.003 [17].

7.6.2.4 Previous location area Id

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

7.6.2.5 Stored location area Id

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

7.6.2.6 Current location area Id

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

7.6.2.7 Target location area Id

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

7.6.2.8 Target cell Id

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

7.6.2.8A Target RNC Id

This parameter refers to the identity of the RNC to which a call has to be relocated.

7.6.2.9 Void

7.6.2.10 Originating entity number

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

7.6.2.11 MSC number

This parameter refers to the ISDN number of an MSC.

7.6.2.12 Target MSC number

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

7.6.2.13 HLR number

This parameter refers to the ISDN number of an HLR.

7.6.2.14 VLR number

This parameter refers to the ISDN number of a VLR.

7.6.2.15 HLR Id

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

7.6.2.16 LMSI

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

7.6.2.17 MS ISDN

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

7.6.2.18 OMC Id

This parameter refers to the identity of an Operation and Maintenance Centre.

7.6.2.19 Roaming number

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

7.6.2.19A Relocation Number List

This parameter refers to the number(s) used for routing one call or several calls between MSCs during relocation.

7.6.2.20 Void

7.6.2.21 Handover number

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

7.6.2.22 Forwarded-to number

This parameter refers to the address to which a call is to be forwarded. A subaddress may be appended. For subscribers having an originating CAMEL Phase 2 or higher subscription, this address need not be in E.164 international format.

7.6.2.22A Long forwarded-to number

This parameter refers to the address to which a call is to be forwarded. A subaddress may be appended. For subscribers having an originating CAMEL Phase 2 or higher subscription this address need not be in international format.

7.6.2.22B Long FTN Supported

This parameter indicates that the sending entity supports Long Forwarded-to Numbers.

7.6.2.23 Forwarded-to subaddress

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

7.6.2.24 Called number

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

7.6.2.25 Calling number

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

7.6.2.26 Originally dialled number

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

7.6.2.27 Service centre address

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

7.6.2.28 Zone Code

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

7.6.2.29 MSIsdn-Alert

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

7.6.2.30 Location Information

This parameter indicates the location of the served subscriber as defined in 3GPP TS 23.018.

7.6.2.31 GMSC Address

This parameter refers to the E.164 address of a GMSC.

7.6.2.32 VMSC Address

This parameter refers to the E.164 address of a VMSC.

7.6.2.33 Group Id

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

7.6.2.34 North American Equal Access preferred Carrier Id

This parameter refers to the carrier identity preferred by the subscriber for calls requiring routing via an inter-exchange carrier. This identity is used at:

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

7.6.2.35 SIWFS Number

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

7.6.2.36 B-subscriber address

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

7.6.2.37 Serving cell Id

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

7.6.2.38 SGSN number

This parameter refers to the ISDN number of a SGSN.

7.6.2.39 SGSN address

This parameter refers to the IP-address of a SGSN. This parameter is defined in 3GPP TS 23.003 [17].

7.6.2.40 GGSN address

This parameter refers to the IP-address of a GGSN. This parameter is defined in 3GPP TS 23.003 [17].

7.6.2.41 GGSN number

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

7.6.2.42 APN

This parameter refers to the DNS name of a GGSN. This parameter is defined in 3GPP TS 23.060.

7.6.2.43 Network Node number

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

7.6.2.44 PDP-Type

This parameter indicates which type of protocol is used by the MS as defined in 3GPP TS 23.060.

7.6.2.45 PDP-Address

This parameter indicates the address of the data protocol as defined in 3GPP TS 23.060.

7.6.2.46 Additional number

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

7.6.2.47 P-TMSI

This parameter is the Packet Temporary Mobile Subscriber Identity defined in 3GPP TS 23.003 [17].

7.6.2.48 B-subscriber number

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

7.6.2.49 B-subscriber subaddress

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

7.6.2.50 LMU Number

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

7.6.2.51 MLC Number

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

7.6.2.52 Multicall Bearer Information

This parameter refers to the number of simultaneous bearers supported per user by the serving network.

7.6.2.53 Multiple Bearer Requested

This parameter indicates whether multiple bearers are requested for a relocation.

7.6.2.54 Multiple Bearer Not Supported

This parameter indicates whether multiple bearers are supported.

7.6.2.55 PDP-Charging Characteristics

This parameter indicates the charging characteristics associated with a specific PDP context as defined in 3GPP TS 32.015.

7.6.2.56 Selected RAB ID

The selected radio access bearer to be kept at subsequent inter-MSC handover from UMTS to GSM.

7.6.2.57 RAB ID

This parameter indicates the radio access bearer identifier as defined in 3GPP TS 25.413. This parameter is used to relate the radio resources with the radio access bearers.

7.6.3 Subscriber management parameters

7.6.3.1 Category

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

7.6.3.2 Equipment status

This parameter refers to the status of the mobile equipment as defined in 3GPP TS 22.016 [7].

7.6.3.3 Extensible Bearer service

This parameter may refer to a single bearer service, a set of bearer services or to all bearer services as defined in 3GPP TS 22.002 [3]. This parameter is used only for subscriber profile management. Extensible Bearer service values include all values defined for a Bearer service parameter (7.6.4.38).

7.6.3.4 Extensible Teleservice

This parameter may refer to a single teleservice, a set of teleservices or to all teleservices as defined in 3GPP TS 22.002 [3]. This parameter is used only for subscriber profile management. Extensible Teleservice values include all values defined for a Teleservice parameter (7.6.4.39).

7.6.3.5 Extensible Basic Service Group

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

7.6.3.6 GSM bearer capability

This parameter refers to the GSM bearer capability information element defined in 3GPP TS 24.008.

7.6.3.7 Subscriber Status

This parameter refers to the barring status of the subscriber:

- service granted;
- Operator Determined Barring.

7.6.3.8 CUG Outgoing Access indicator

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

7.6.3.9 Operator Determined Barring General Data

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

- All outgoing calls barred; (*)
- International outgoing calls barred; (*)
- International outgoing calls except those to the home PLMN country barred; (*)
- Interzonal outgoing calls barred; (*)
- Interzonal outgoing calls except those to the home PLMN country barred; (*)
- Interzonal outgoing calls AND international outgoing calls except those directed to the home PLMN country barred; (*)
- Premium rate (information) outgoing calls barred;
- Premium rate (entertainment) outgoing calls barred;
- Supplementary service access barred;
- Invocation of call transfer barred;
- Invocation of chargeable call transfer barred;
- Invocation of internationally chargeable call transfer barred;
- Invocation of interzonally chargeable call transfer barred;
- Invocation of call transfer where both legs are chargeable barred;
- Invocation of call transfer if there is already an ongoing transferred call for the served subscriber in the serving MSC/VLR barred;
- Outgoing calls when roaming outside the home PLMN country; (**)
- All incoming calls; (**)
- Incoming calls when roaming outside the home PLMN country; (**)
- Incoming calls when roaming outside the zone of the home PLMN country; (**)
- Roaming outside the home PLMN; (**)
- Roaming outside the home PLMN country; (**)
- Registration of any call forwarded-to number; (**)
- Registration of any international call forwarded-to number; (**)
- Registration of any international call forwarded-to number except to a number within the HPLMN country; (**)
- Registration of any inter-zone call forwarded-to number; (**)
- Registration of any inter-zone call forwarded-to number except to a number within the HPLMN country. (**)

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

(**) These ODBs are not used for InsertSubscriberData.

7.6.3.10 ODB HPLMN Specific Data

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

- Operator Determined Barring Type 1;

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

7.6.3.11 Regional Subscription Data

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

7.6.3.12 Regional Subscription Response

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

7.6.3.13 Roaming Restriction Due To Unsupported Feature

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

7.6.3.14 Extensible SS-Info

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

- extensible forwarding information (see clauseclause 7.6.3.15);
- extensible call barring information (see clauseclause 7.6.3.20);
- CUG info (see clauseclause 7.6.3.22);
- extensible SS-Data (see clauseclause 7.6.3.29).

7.6.3.15 Extensible forwarding information

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

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

  The list may contain one item per Basic Service Group.

7.6.3.16 Extensible forwarding feature

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

- extensible Basic Service Group (see clauseclause 7.6.3.5);
- extensible SS-Status (see clauseclause 7.6.3.17);
- forwarded-to number (see clauseclause 7.6.2.22);
- forwarded-to subaddress (see clauseclause 7.6.2.23);
- extensible forwarding options (see clauseclause 7.6.3.18);
- extensible no reply condition timer (see clauseclause 7.6.4.19);
- long forwarded-to number (see clauseclause 7.6.2.22A).

If a number is required to define the forwarded-to destination then:

- If the VLR supports Long Forwarded-to Numbers then the long forwarded-to number shall be present and the forwarded-to number shall be absent;
- If the VLR does not support Long Forwarded-to Numbers then the forwarded-to number shall be present and the long forwarded-to number shall be absent.

7.6.3.17 Extensible SS-Status

This parameter refers to the state information of individual supplementary services as defined in 3GPP TS 23.011[22].

7.6.3.18 Extensible Forwarding Options

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

- notification to forwarding party (see 3GPP TS 22.082[10] for the meaning of this parameter);
- redirection notification to the forwarded-to party (see 3GPP TS 22.082[10] for the meaning of this parameter);
- notification to calling party (see 3GPP TS 22.082[10] for the meaning of this parameter);
- redirecting presentation (see 3GPP TS 22.082[10] for the meaning of this parameter);
- forwarding reason (see 3GPP TS 22.082[10] for the meaning of this parameter).

7.6.3.19 Extensible No reply condition timer

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

7.6.3.20 Extensible Call barring information

This parameter contains for each call barring service:

- SS-Code (see clauseclause 7.6.4.1);
- a list of extensible call barring feature parameters (see clauseclause 7.6.3.21).

  The list may contain one item per Basic Service Group.

7.6.3.21 Extensible Call barring feature

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

- Extensible Basic Service Group (see clauseclause 7.6.3.5);
- provisioned SS-Status (see clauseclause 7.6.3.17).

7.6.3.22 CUG info

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

- CUG subscriptionList;
- CUG featureList.

7.6.3.23 CUG subscription

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

- CUG index;
- CUG interlock;
- Intra CUG restrictions;
- Basic Service Group List.

7.6.3.24 CUG interlock

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

7.6.3.25 CUG index

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

7.6.3.26 CUG feature

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

- Preferential CUG indicator:
  - indicates which CUG index is to be used at outgoing call set-up using the associated Basic Service Group;
- Inter CUG Option:
  - describes whether it for the associated Basic Service Group is allowed to make calls outside the CUG and whether incoming calls are allowed;
- Basic Service Group.

See 3GPP TS 22.085[13] for meaning of this parameter.

7.6.3.27 Inter CUG options

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

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

7.6.3.28 Intra CUG restrictions

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

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

7.6.3.29 Extensible SS-Data

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

- SS-Code (see clauseclause 7.6.4.1);
- Extensible SS-Status (if applicable) (see clauseclause 7.6.3.17);
- Extensible Override subscription option (if applicable) (see clauseclause 7.6.3.30);
- Extensible CLI Restriction (if applicable) (see clauseclause 7.6.3.31);
- Extensible Basic Service Group Code (see clauseclause 7.6.3.5).

7.6.3.30 Subscriber State

This parameter indicates the state of the MS as defined in 3GPP TS 23.018.

7.6.3.31 Requested Info

This parameter indicates the subscriber information being requested as defined in 3GPP TS 23.018.

7.6.3.32 Suppression of Announcement

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

7.6.3.33 Suppress T-CSI

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

7.6.3.34 GMSC CAMEL Subscription Info

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

7.6.3.35 VLR CAMEL Subscription Info

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

7.6.3.36 Supported CAMEL Phases in the VLR

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

7.6.3.36A Supported CAMEL Phases in the SGSN

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

7.6.3.37 CUG Subscription Flag

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

7.6.3.38 CAMEL Subscription Info Withdraw

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

7.6.3.39 Voice Group Call Service (VGCS) Data

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

7.6.3.40 Voice Broadcast Service (VBS) Data

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

7.6.3.41 ISDN bearer capability

This parameter refers to the ISDN bearer capability information element defined in 3GPP TS 29.007[56].

7.6.3.42 Lower layer Compatibility

This parameter refers to the lower layer compatibility information element defined in 3GPP TS 24.008.

7.6.3.43 High Layer Compatibility

This parameter refers to the high layer compatibility information element defined in 3GPP TS 24.008.

7.6.3.44 Alerting Pattern

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

7.6.3.45 GPRS Subscription Data Withdraw

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

7.6.3.46 GPRS Subscription Data

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

7.6.3.47 QoS-Subscribed

This parameter indicates the quality of service subscribed for a certain service. It is defined in 3GPP TS 23.060[104].

7.6.3.48 VPLMN address allowed

This parameter specifies whether the MS is allowed to use a dynamic address allocated in the VPLMN. It is defined in 3GPP TS 23.060[104].

7.6.3.49 Roaming Restricted In SGSN Due To Unsupported Feature

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 feature or service is indicated as unsupported by the SGSN.

7.6.3.50 Network Access Mode

This parameter is defined in 3GPP TS 23.008[20].

7.6.3.51 Mobile Not Reachable Reason

This parameter stores the reason for the MS being absent when an attempt to deliver a short message to an MS fails at the MSC, SGSN or both. It is defined in 3GPP TS 23.040[26].

7.6.3.52 Cancellation Type

This parameter indicates the reason of location cancellation. It is defined in 3GPP TS 23.060[104].

7.6.3.53 All GPRS Data

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

7.6.3.54 Complete Data List Included

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

7.6.3.55 PDP Context Identifier

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

7.6.3.56 LSA Information

This parameter refers to one or more localised service areas a subscriber may be a member of, together with the priority, the preferential access indicator, the active mode support indicator and active mode indication of each localised service area. The access right outside these localised service areas is also indicated.

7.6.3.57 SoLSA support indicator

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

7.6.3.58 LSA Information Withdraw

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

7.6.3.59 LMU Indicator

This parameter indicates the presence of an LMU.

7.6.3.60 LCS Information

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

- GMLC List (see clauseclause 7.6.3.61).
- LCS Privacy Exception List (see clauseclause 7.6.3.62).
- MO-LR List (see clauseclause 7.6.3.65A).

7.6.3.61 GMLC List

This parameter contains the addresses of all GMLCs that are permitted to issue a non-call related MT-LR location request for this MS. Usage of this parameter is defined in 3GPP TS 23.171.

7.6.3.62 LCS Privacy Exception List

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

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

7.6.3.63 LCS Privacy Exception Parameters

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

- provisioned SS-Status (see clauseclause 7.6.3.17);
- privacy notification to MS user (see clauseclause 7.6.3.65B);
- external client List (see clauseclause 7.6.3.64);
- internal client List (see clauseclause 7.6.3.65).

7.6.3.64 External Client List

This parameter is only applicable to the non-call related privacy class and gives the identities of the external clients that 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 each identified external client, GMLC restrictions may be defined. It may also be indicated if the MS shall be notified of a non-restricted MT-LR from each identified LCS client and, if so, whether notification only or notification with privacy verification shall apply. Usage of this parameter is defined in 3GPP TS 23.171.

7.6.3.65 Internal Client List

This parameter is only applicable to the PLMN operator privacy class and gives the identities of the internal PLMN operator clients that are allowed to locate a target MS for an NI-LR or MT-LR. Usage of this parameter is defined in 3GPP TS 23.171.

7.6.3.65A MO-LR List

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

- SS-Code (see clauseclause 7.6.4.1).

7.6.3.65B Privacy Notification to MS User

This parameter is applicable to the non-call related privacy class and call related privacy class. For non-call related privacy class it indicates whether the MS user shall be notified for a non-call related MT-LR from any value added LCS client when the MT-LR is restricted and be enabled to accept or override the restriction. For call related privacy class it indicates whether the MS shall be notified of a call related MT-LR and, if so, whether notification only or notification with privacy verification shall apply. Usage of this parameter is defined in 3GPP TS 23.171.

7.6.3.65C GMLC List Withdraw

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

7.6.3.66 IST Alert Timer

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

7.6.3.67 Call Termination Indicator

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

7.6.3.68 IST Information Withdraw

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

7.6.3.69 IST Support Indicator

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

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

7.6.3.70 Super-Charger Supported In HLR

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

7.6.3.71 Super-Charger Supported In Serving Network Entity

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

7.6.3.72 Age Indicator

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

7.6.3.73 GPRS enhancements support indicator

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

7.6.3.74 Extensible QoS-Subscribed

This parameter indicates the enhanced QoS subscribed for a certain service. It is defined in 3GPP TS 23.060 [104]. This parameter is an extension to QoS-Subscribed.

7.6.3.75 SGSN Camel Subscription Info

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

7.6.3.76 SMS-CSI

This parameter identifies the subscriber as having SMS CAMEL services as defined in 3GPP TS 23.078 [98].

7.6.3.77 GPRS-CSI

This parameter identifies the subscriber as having GPRS CAMEL services as defined in 3GPP TS 23.078 [98].

7.6.3.78 CAMEL subscription info

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

7.6.3.79 Extensible Call barring information for CSE

This parameter contains for each call barring service for CSE:

- SS-Code;
- a list of extensible call barring feature parameters.

  The list may contain one item per Basic Service Group.
- password;
- wrong password attempt counter;
- notification-to-CSE flag.

7.6.3.80 Extensible Forwarding information for CSE

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

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

7.6.3.81 Modification Request for CSI

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

7.6.3.82 Modification Request for SS Information

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

7.6.3.83 Call Barring Data

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

7.6.3.84 Call Forwarding Data

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

7.6.3.85 ODB Data

This parameter contains the ODB general data, ODB HPLMN specific data .

7.6.3.86 Requested Subscription Info

This parameter indicates the subscription information being requested.

7.6.3.87 CS Allocation/Retention priority

This parameter indicates the allocation/retention priority for Circuit Switched (CS). It corresponds to the allocation/retention priority that is defined in 3GPP TS 23.107.

7.6.3.88 ODB Info

This parameter contains the ODB data and Notification to CSE flag.

7.6.4 Supplementary services parameters

7.6.4.1 SS-Code

This parameter may refer to one supplementary service or a set of supplementary services as defined in 3GPP TS 22.004. For MAP Release '99 this includes:

- Calling Line Identification Presentation service (CLIP);
- Calling Line Identification Restriction service (CLIR);
- Connected Line Identification Presentation service (COLP);
- Connected Line Identification Restriction service (COLR);

- Calling Name Presentation (CNAP);
- All Call Forwarding services;
- Call Waiting (CW);
- Call Hold (HOLD);
- Multi-Party service (MPTY);
- Closed User Group (CUG);
- All Charging services;
- All Call Restriction services;
- Explicit Call Transfer service (ECT);
- enhanced Multi-Level Precedence and Pre-emption service (eMLPP);
- Completion of Calls to Busy Subscriber, originating side (CCBS-A);
- Completion of Calls to Busy Subscriber, destination side (CCBS-B);
- All LCS privacy exceptions (see clause 7.6.4.44);
- Mobile Originating Location Request (MO-LR) (see clause 7.6.4.44A);
- Multicall (MC).

7.6.4.2 SS-Status

This parameter refers to the state information of individual supplementary services as defined in 3GPP TS 23.011.

7.6.4.3 SS-Data

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

- SS-Code (see clause 7.6.4.1);
- SS-Status (if applicable) (see clause 7.6.4.2);
- Override subscription option (see clause 7.6.4.4);
- CLI Restriction (see clause 7.6.4.5);
- Basic Service Group Code (see clause 7.6.4.40).

7.6.4.4 Override Category

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

- Enabled;
- Disabled.

7.6.4.5 CLI Restriction Option

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

- Permanent;
- Temporary (Default Restricted);

- Temporary (Default Allowed).

7.6.4.6 Forwarding Options

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

- notification to forwarding party (see 3GPP TS 22.082[10] for the meaning of this parameter);
- notification to calling party (see 3GPP TS 22.082[10] for the meaning of this parameter);
- redirecting presentation (see 3GPP TS 22.082[10] for the meaning of this parameter);
- Forwarding reason (see 3GPP TS 22.082[10] for the meaning of this parameter).

7.6.4.7 No reply condition timer

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

7.6.4.8 - 7.6.4.14 Void

7.6.4.15 Forwarding information

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

- the SS-Code of the relevant call forwarding service (see clause 7.6.4.1);
- if required, a list of forwarding feature parameters (see clause 7.6.4.16).

  the list may contain one item per Basic Service Group.

7.6.4.16 Forwarding feature

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

- Basic Service Group (see clause 7.6.4.40);
- SS-Status (see clause 7.6.4.2);
- forwarded-to number (see clause 7.6.2.22);
- forwarded-to subaddress (see clause 7.6.2.23);
- forwarding options (see clause 7.6.4.6);
- no reply condition timer (see clause 7.6.4.7);
- long forwarded-to number (see clause 7.6.2.22A).

If a number is required to define the forwarded-to destination then:

- If the VLR supports Long Forwarded-to Numbers then the long forwarded-to number shall be present and the forwarded-to number shall be absent.
- If the VLR does not support Long Forwarded-to Numbers then the forwarded-to number shall be present and the long forwarded-to number shall be absent.

7.6.4.17 Void

7.6.4.18 Call barring information

This parameter contains for each call barring service:

- SS-Code (see clause 7.6.4.1);
- a list of call barring feature parameters (see clause 7.6.4.19).

  The list may contain one item per Basic Service Group.

7.6.4.19 Call barring feature

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

- Basic Service Group (see clause 7.6.4.40);
- SS-Status (see clause 7.6.4.2).

7.6.4.20 New password

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

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

7.6.4.21 Current password

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

7.6.4.22 Guidance information

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

- "enter password";

  this information is used for checking of the old password;
- "enter new password";

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

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

7.6.4.23 Void

7.6.4.24 SS-Info

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

- forwarding information (see clause 7.6.4.15);
- call barring information (see clause 7.6.4.18);
- CUG info (see clause 7.6.4.8);
- SS-Data (see clause 7.6.4.3).
- eMLPP information (see clause 7.6.4.41).

7.6.4.25 - 7.6.4.35 Void

7.6.4.36 USSD Data Coding Scheme

This parameter contains the information of the alphabet and the language used for the unstructured information in an Unstructured Supplementary Service Data operation. The coding of this parameter is according to the Cell Broadcast Data Coding Scheme as specified in 3GPP TS 23.038[25].

7.6.4.37 USSD String

This parameter contains a string of unstructured information in an Unstructured Supplementary Service Data operation. The string is sent either by the mobile user or the network. The contents of a string sent by the MS are interpreted by the network as specified in 3GPP TS 22.090[16].

7.6.4.38 Bearer service

This parameter may refer to a single bearer service, a set of bearer services or to all bearer services as defined in 3GPP TS 22.002[3]. This parameter is used only for supplementary service management.

7.6.4.39 Teleservice

This parameter may refer to a single teleservice, a set of teleservices or to all teleservices as defined in 3GPP TS 22.003[4]. This parameter is used only for supplementary service management.

7.6.4.40 Basic Service Group

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

7.6.4.41 eMLPP information

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

- maximum entitled priority:
  - indicates the highest priority level the subscriber is allowed to apply for an outgoing call set-up;
- default priority:
  - defines the priority level which shall be assigned to a call if no explicit priority is indicated during call set-up.

7.6.4.42 SS-event

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

- Explicit Call Transfer (ECT)
- Call Deflection (CD)
- Multi-Party call (MPTY)
- Completion of Calls to Busy Subscriber (CCBS)

7.6.4.43 SS-event data

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

ECT A list with all Called Party Numbers involved.

CD The called Party number involved.

7.6.4.44 LCS Privacy Exceptions

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

- Universal Class;
- Call related value added class;
- Non-Call related value added class;
- PLMN operator class.

7.6.4.45 Mobile Originating Location Request (MO-LR)

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

- Basic Self Location;
- Autonomous Self Location;
- Transfer to Third Party.

7.6.4.46 NbrUser

This parameter indicates the maximum number of parallel bearers that may be used as defined by the user at registration of the MC SS.

7.6.4.47 MC Subscription Data

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

- NbrUser:

  indicates the maximum number of parallel bearers that may be used as defined by the user at registration of the MC SS

- NbrSB:

  indicates the maximum number of parallel bearers that may be used as defined by the user's subscription.

7.6.4.48 MC Information

This parameter contains three parameters which are associated with the MC service. The following parameters are included:

- NbrSB;
- NbrUser;
- NbrSN.

Definitions of these parameters are provided in 3GPP TS 23.135.

7.6.4.49 CCBS Request State

This parameter indicates the current state of the CCBS request. It can take one of seven values:

- request;
- recall;
- active;

- completed;
- suspended;
- frozen;
- deleted.

7.6.5 Call parameters

7.6.5.1 Call reference number

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

7.6.5.2 Interrogation type

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

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

7.6.5.3 OR interrogation

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

7.6.5.4 OR capability

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

7.6.5.5 Forwarding reason

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

- busy subscriber;
- mobile subscriber not reachable;
- no subscriber reply.

7.6.5.6 Forwarding interrogation required

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

7.6.5.7 O-CSI

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

7.6.5.7A D-CSI

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

7.6.5.7B T-CSI

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

7.6.5.7C VT-CSI

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

7.6.5.8 Call Direction

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

7.6.5.9 Channel Type

This parameter is the result of a Channel Mode Modification for TS 61/62. It contains the changed Air Interface User Rate. The information is sent from the SIWFS to the MSC to assign the correct radio resource. This parameter is defined in GSM 08.08.

7.6.5.10 Chosen Channel

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

7.6.5.11 CCBS Feature

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

- CCBS Index (see 3GPP TS 23.093 for the use of this parameter);
- B-subscriber number (see clause 7.6.2.48);
- B-subscriber subaddress (see clause 7.6.2.49);
- Basic Service Group Code (see clause 7.6.4.40).

7.6.5.12 UU Data

This parameter includes User-To-User Data. It is defined in 3GPP TS 23.087.

7.6.5.13 UUS CF Interaction

This parameter indicates if the call forwarding or call deflection has been activated after UUS1 request has been accepted . It is defined in 3GPP TS 23.087.

7.6.5.14 Number Portability Status

This parameter indicates the number portability status of subscriber. See 3GPP TS 23.066 [108].

7.6.5.15 Pre-paging supported

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

7.6.6 Radio parameters

7.6.6.1 - 7.6.6.4 Void

7.6.6.5 BSSMAP Service Handover

This parameter refers to the Service Handover information element defined in GSM 08.08.

7.6.6.5A BSSMAP Service Handover List

This parameter refers to the list of Service Handover information elements defined in GSM 08.08. This parameter shall be used when there are multiple bearers and at least one of the bearers has an associated BSSMAP Service Handover parameter.

7.6.6.6 RANAP Service Handover

This parameter refers to the Service Handover information element defined in 3GPP TS 25.413.

7.6.6.7 HO-Number Not Required

This parameter indicates that no handover or relocation number allocation is necessary.

7.6.6.8 Integrity Protection Information

This parameter refers to the Integrity Protection Information element defined in 3GPP TS 25.413.

7.6.6.9 Encryption Information

This parameter refers to the Encryption Information element defined in 3GPP TS 25.413.

7.6.6.10 Radio Resource Information

This parameter refers to the Channel Type information element defined in GSM 08.08.

7.6.6.10A Radio Resource List

This parameter refers to list of RAB-id's and their associated Channel Type information elements defined in GSM 08.08. This parameter shall be used when there are multiple bearers and at least one of the bearers has an associated Radio Resource Information parameter.

7.6.6.10B Chosen Radio Resource Information

This parameter refers to the Chosen Channel and Speech Version information elements defined in GSM 08.08.

7.6.6.11 Key Status

This parameter refers to the Key Status element defined in 3GPP TS 25.413.

7.6.6.12 Selected UMTS Algorithms

This parameters identifies the UMTS integrity and optionally encryption algorithms selected by MSC-B. Coding of this parameter is defined in 3GPP TS 25.413.

7.6.6.13 Allowed GSM Algorithms

This parameters identifies the allowed GSM algorithms in MSC-B. The coding of this parameter is defined in GSM 08.08.

7.6.6.14 Allowed UMTS Algorithms

This parameters identifies the allowed UMTS algorithms in MSC-B. Coding of this parameter is defined in 3GPP TS 25.413.

7.6.6.15 Selected GSM Algorithm

This parameter identifies the GSM algorithm selected by GSM BSC controlled by MSC-B. Coding of this parameter is defined in GSM 08.08.

7.6.7 Authentication parameters

7.6.7.1 Authentication set list

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

The list either contains Authentication Triplets (Rand, Sres, Kc) or Authentication Quintuplets (Rand, Xres, Ck, Ik, Autn). If the list contains Authentication Quintuplets, the order of sequence in this list is chronological, the first quintuplet in the list is the oldest one.

7.6.7.2 Rand

This parameter represents a random number used for authentication.

7.6.7.3 Sres

This parameter represents the response to an authentication request.

7.6.7.4 Kc

This parameter refers to a key used for ciphering purposes.

7.6.7.5 Xres

This parameter represents the response to an UMTS authentication request.

7.6.7.5A Ck

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

7.6.7.5B Ik

This parameter refers to the Integrity Key.

7.6.7.5C Autn

This parameter refers to the Authentication Token.

7.6.7.6 Cksn

This parameter refers to a ciphering key sequence number.

7.6.7.6A Ksi

This parameter refers to a key set identifier.

7.6.7.6B Auts

This parameter refers to the resynchronisation token.

7.6.7.7 Ciphering mode

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

- no encryption;

- identification of specific ciphering algorithm.

Current Security Context

This parameter represents a list of security context parameters for a given subscriber.

The list either contains GSM Security Context data (Kc, Cksn) or UMTS Security Context Data (Ck, Ik, Ksi).

7.6.7.9 Failure cause

This parameter refers to an authentication failure which has occurred. It may take values as follows:

- wrong user response;
- wrong network signature.

7.6.8 Short message parameters

7.6.8.1 SM-RP-DA

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

- IMSI (see clause 7.6.2.1);
- LMSI (see clause 7.6.2.16);
- MS-ISDN (see clause 7.6.2.17);
- roaming number (see clause 7.6.2.19);
- service centre address (see clause 7.6.2.27).

7.6.8.2 SM-RP-OA

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

- MS-ISDN (see clause 7.6.2.17);
- service centre address (see clause 7.6.2.27).

7.6.8.3 MWD status

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

7.6.8.4 SM-RP-UI

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

7.6.8.5 SM-RP-PRI

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

7.6.8.6 SM Delivery Outcome

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

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

7.6.8.7 More Messages To Send

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

7.6.8.8 Alert Reason

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

- MS present;
- Memory Available.

7.6.8.9 Absent Subscriber Diagnostic SM

This parameter is used to indicate the reason why the subscriber is absent. For the values for this parameter see 3GPP TS 23.040[26].

7.6.8.10 Alert Reason Indicator

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

7.6.8.11 Additional SM Delivery Outcome

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

7.6.8.12 Additional Absent Subscriber Diagnostic SM

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

7.6.8.13 Delivery Outcome Indicator

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

7.6.8.14 GPRS Node Indicator

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

7.6.8.15 GPRS Support Indicator

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

7.6.8.16 SM-RP-MTI

This parameter represents the RP-Message Type Indicator of the Short Message. It is used to distinguish a SM sent to the mobile station in order to acknowledge an MO-SM initiated by the mobile from a normal MT-SM. This parameter is formatted according to the formatting rules of address fields as described in 3GPP TS 23.040[26].

7.6.8.17 SM-RP-SMEA

This parameter represents the RP-Originating SME-address of the Short Message Entity that has originated the SM. This parameter is used by the short message service relay sub-layer protocol and is formatted according to the formatting rules of address fields as described in 3GPP TS 23.040[26].

7.6.9 Access and signalling system related parameters

7.6.9.1 AN-apdu

This parameter includes one or two concatenated complete 3GPP TS 25.413 or GSM 08.06 messages, as described in 3GPP TS 23.009 and 3GPP TS 29.010. The access network protocol ID indicates that the message or messages are according to either GSM 08.06 or 3GPP TS 25.413. For the coding of the messages see 3GPP TS 25.413, GSM 08.06 and GSM 08.08.

7.6.9.2 CM service type

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

- mobile originating call;
- emergency call establishment;
- short message service;
- mobile originating call re-establishment;
- mobile terminating call;
- SS request;
- Voice group call set-up;
- Voice broadcast set-up.

7.6.9.3 Access connection status

This parameter represents the following access connection status information:

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

7.6.9.4 External Signal Information

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

7.6.9.5 Access signalling information

This parameter refers to any set of information elements imported from 3GPP TS 24.008.

7.6.9.6 Location update type

This parameter refers to the location update type (normal, periodic or IMSI attach) contained in the 3GPP TS 24.008 LOCATION REGISTRATION REQUEST message.

7.6.9.7 Protocol ID

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

The following values are defined:

- 04.08;
- 08.06;
- ETS 300 102-1.

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

7.6.9.8 Network signal information

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

The network signal information may include the following information elements as defined in 3GPP TS 29.007 [56]:

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

  For the content, see 3GPP TS 29.007 [56].

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

  For the content, see 3GPP TS 29.007 [56].

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

  For the content, see 3GPP TS 29.007 [56].

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

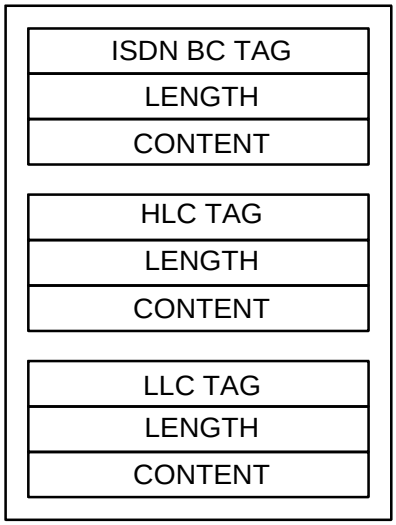


Figure 7.6/1: Network signal information parameter

7.6.9.9 Call Info

This parameter is transported as external signal information. The protocol ID shall be set to "3GPP TS 24.008".

The Call Info includes the set of information elements from the original SETUP message and is imported from 3GPP TS 24.008.

7.6.9.10 Additional signal info

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

The additional signal information may include the following information elements:

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

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

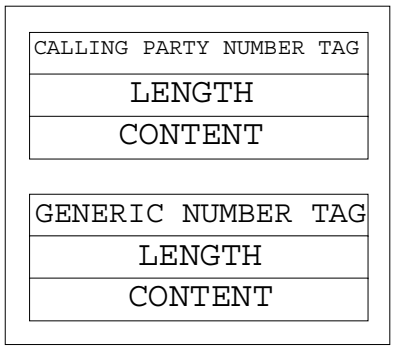


Figure 7.6/2: Additional signal information parameter

7.6.10 System operations parameters

7.6.10.1 Network resources

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

- PLMN;
- HLR;
- VLR (current or previous);
- MSC (controlling or current);
- EIR;
- radio sub-system.

7.6.10.2 Trace reference

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

7.6.10.3 Trace type

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

7.6.11 Location Service Parameters

7.6.11.1 Age of Location Estimate

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

7.6.11.2 Void

7.6.11.3 Void

7.6.11.4 LCS Client ID

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

7.6.11.5 LCS Event

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

7.6.11.6 LCS MLC Data

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

7.6.11.7 LCS Priority

This parameter gives the priority of the location request.

7.6.11.8 LCS QoS

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

1) Response Time

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

2) Horizontal Accuracy

   Indicates the required horizontal accuracy of the location estimate.

3) Vertical Coordinate

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

4) Vertical Accuracy

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

7.6.11.9 Void

7.6.11.10 Void

7.6.11.11 Location Estimate

This parameter gives an estimate of the location of an MS in universal coordinates and the accuracy of the estimate. The estimate is expressed in terms of the geographical shapes defined by 3GPP TS 23.032. and is composed of the type of shape plus the encoding of the shape itself. Any type of shape defined in 3GPP TS 23.032 can be filled in in the Location Estimate parameter, but only the encoding of the following shapes shall be carried by Location Estimate:

- Ellipsoid point with uncertainty circle
- Ellipsoid point with uncertainty ellipse
- Ellipsoid point with altitude and uncertainty ellipsoid
- Ellipsoid arc
- Ellipsoid point

The encoding for the remaining types of shape, defined in the 3GPP TS 23.032, shall be filled in in the Additional Location Estimate parameter.

7.6.11.12 Location Type

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

- current location;
- current or last known location;
- initial location for an emergency services call.

7.6.11.13 NA-ESRD

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

7.6.11.14 NA-ESRK

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

7.6.11.15 Void

7.6.11.16 Privacy Override

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

7.6.11.17 Void

7.6.11.18 Void

7.6.11.19 NA-ESRK Request

This parameter allows the MSC to indicate that it requires the GMLC to allocate a NA-ESRK based on the target MS location estimate. This parameter only applies to emergency services calls in North America.

7.6.11.20 Supported GAD Shapes

This parameter indicates which of the shapes defined in 3GPP TS 23.032 are supported. If the parameter is not provided then the receiving node shall assume that the sending entity supports the following shapes:

- Ellipsoid point with uncertainty circle
- Ellipsoid point with uncertainty ellipse
- Ellipsoid point with altitude and uncertainty ellipsoid
- Ellipsoid arc
- Ellipsoid point

7.6.11.21 Additional Location Estimate

This parameter gives an estimate of the location of an MS/UE in universal coordinates and the accuracy of the estimate. This parameter allows the location estimate to be expressed in any of the geographical shapes defined in 3GPP TS 23.032.

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

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

ParameterNameLIST

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

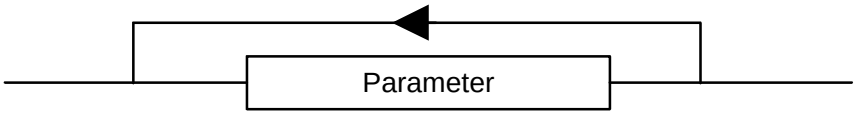


Figure 7.7/1: Construction of Lists

8 Mobility services

8.1 Location management services

8.1.1 Void

8.1.1.1 Void

8.1.1.2 Void

8.1.1.3 Void

8.1.2 MAP_UPDATE_LOCATION service

8.1.2.1 Definition

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

The MAP_UPDATE_LOCATION service is a confirmed service using the service primitives given in table 8.1/2.

8.1.2.2 Service primitives

Table 8.1/2: MAP_UPDATE_LOCATION

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 Network Entity	C	C(=)		
Long FTN Supported	C	C(=)		
Inform Previous Network Entity	C	C(=)		
HLR number			C	C(=)
User error			C	C(=)
Provider error				O

8.1.2.3 Parameter definitions and use

Invoke Id

See definition in clause 7.6.1.

IMSI

See definition in clause 7.6.2.

MSC Address

See definition for MSC number in clause 7.6.2. The MSC address is used for short message delivery only and for each incoming call set-up attempt the MSRN will be requested from the VLR.

VLR number

See definition in clause 7.6.2.

LMSI

See definition in clause 7.6.2. It is an operator option to provide the LMSI from the VLR; it is mandatory for the HLR to support the LMSI handling procedures.

Supported CAMEL Phases

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

HLR number

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

SoLSA Support Indicator

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

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

IST Support Indicator

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

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

Long FTN Supported

This parameter indicates that the VLR supports Long Forwarded-to Numbers.

Super-Charger Supported in Serving Network Entity

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

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

Inform Previous Network Entity

This parameter is used by the VLR to ask the HLR to inform the previous network entity about the update by sending the previous network entity a Cancel Location message. It is used in case Super-Charger is supported in the network and the serving network entity has not been able to inform the previous network entity that MS has moved, that is if it has not sent Send Identification to the previous serving entity.

User error

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

- unknown subscriber;
- roaming not allowed;

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

Provider error

For definition of provider errors see clause 7.6.1.

8.1.3 MAP_CANCEL_LOCATION service

8.1.3.1 Definition

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

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

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

8.1.3.2 Service primitives

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

8.1.3.3 Parameter definitions and use

Invoke Id

See definition in clause 7.6.1.

IMSI

See definition in clause 7.6.2.

LMSI

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

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

Cancellation Type

See definition in clause 7.6.3. The presence of this parameter is mandatory when the Cancel Location is sent to the SGSN. If the VLR receives this parameter and do not understand it the VLR shall ignore it.

User error

If the cancellation fails, an error cause is to be returned by the VLR or by the SGSN. One of the following error causes defined in clause 7.6.1 shall be used:

- unexpected data value;
- data missing.

Provider error

For definition of provider errors see clause 7.6.1.

8.1.4 MAP_SEND_IDENTIFICATION service

8.1.4.1 Definition

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

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

8.1.4.2 Service primitives

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(=)
Current Security Context			U	C(=)
User error			C	C(=)
Provider error				O

8.1.4.3 Parameter definitions and use

Invoke Id

See definition in clause 7.6.1.

TMSI

See definition in clause 7.6.2.

If multiple service requests are present in a dialogue then this parameter shall be present in every service request.

Number of requested vectors

A number indicating how many authentication vectors the new VLR is prepared to receive. The previous VLR shall not return more vectors than indicated by this parameter.

This parameter shall be present in the first (or only) request of the dialogue. If multiple service requests are present in a dialogue then this parameter shall not be present in any service request other than the first one

Segmentation prohibited indicator

This parameter indicates if the new VLR or SGSN allows segmentation of the response at MAP user level.

This parameter may be present only in the first request of the dialogue.

IMSI

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

If multiple service requests are present in a dialogue and the service succeeds then this parameter shall not be present in any service response other than the first one.

Authentication set

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

Current Security Context

See definition in clause 7.6.7. If the service succeeds, a list of either GSM or UMTS Security Context parameters can be returned.

User error

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

- unidentified subscriber.

Provider error

For definition of provider errors see clause 7.6.1.

8.1.5 Void

8.1.5.1 Void

8.1.5.2 Void

8.1.5.3 Void

8.1.6 MAP_PURGE_MS service

8.1.6.1 Definition

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

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 request for routing information for a mobile terminated short message or a network requested PDP-context activation 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 SGSN, either by MMI interaction or automatically, e.g. because the MS has been inactive for several days. This service shall not be used if both the SGSN and HLR support the Super-Charger functionality.

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

8.1.6.2 Service primitives

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

8.1.6.3 Parameter definitions and use

Invoke ID

See definition in clause 7.6.1.

IMSI

See definition in clause 7.6.2.

VLR number

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

SGSN number

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

Freeze TMSI

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

Freeze P-TMSI

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 number matches the stored SGSN number.

User error

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

- Data Missing;
- Unexpected Data Value;
- Unknown Subscriber.

Provider error

See definition of provider errors in clause 7.6.1.

8.1.7 MAP_UPDATE_GPRS_LOCATION service

8.1.7.1 Definition

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

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

8.1.7.2 Service primitives

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(=)		
Inform Previous Network Entity	C	C(=)		
HLR number			C	C(=)
User error			C	C(=)
Provider error				O

8.1.7.3 Parameter definitions and use

Invoke Id

See definition in clause 7.6.1.

IMSI

See definition in clause 7.6.2.

SGSN number

See definition in clause 7.6.2.

SGSN address

See definition in clause 7.6.2.

Supported CAMEL Phases

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

SoLSA Support Indicator

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

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

Super-Charger Supported in Serving Network Entity

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

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

GPRS enhancements support indicator

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

Inform Previous Network Entity

This parameter is used by the SGSN to ask the HLR to inform the previous network entity about the update by sending the previous network entity a Cancel Location message. It is used in case Super-Charger is supported in the network and the serving network entity has not been able to inform the previous network entity that MS has moved, that is if it has not sent SGSN Context Request to the previous serving entity.

HLR number

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

User error

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

- unknown subscriber;
- roaming not allowed.

  This cause will be sent if the MS is not allowed to roam into the PLMN indicated by the SGSN number. The cause is qualified by the roaming restriction reason "PLMN Not Allowed" or "Operator Determined Barring".
- system failure;
- unexpected data value.

The diagnostic in the Unknown Subscriber may indicate “Imsi Unknown” or “Gprs Subscription Unknown”.

Provider error

For definition of provider errors see clause 7.6.1.

8.1.8 MAP-NOTE-MM-EVENT

8.1.8.1 Definition

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

8.1.8.2 Service primitives

The service primitives are shown in table 8.1/8.

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

8.1.8.3 Parameter use

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

- Location update in the same VLR service area;
- Location update to another VLR service area;
- IMSI attach;
- MS initiated IMSI detach (explicit detach);
- Network initiated IMSI detach (implicit detach).

Service Key

See clause 7.6.x.

IMSI

See clause 7.6.x.

Basic MSISDN

See clause 7.6.x.

Location Information

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

LSA Identity

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

Supported CAMEL Phases

See clause 7.6.x. This information shall always be sent.

User error

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

- Data Missing;
- Unexpected Data Value;
- Unknown Subscriber;
- MM-EventNotSupported.

Provider error

This is defined in clause 7.6.1.

8.2 Paging and search

8.2.1 MAP_PAGE service

8.2.1.1 Definition

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

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

8.2.1.2 Service primitives

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

8.2.1.3 Parameter definitions and use

Invoke Id

See definition in clause 7.6.1.

IMSI

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

Stored location area Id

See definition in clause 7.6.2.

TMSI

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

User error

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

- absent subscriber;
- unknown location area;
- busy subscriber;
- system failure;
- this corresponds to the case where there is no call associated with the MAP_PAGE service, i.e. if the call has been released but the dialogue to the VLR has not been aborted;
- unexpected data value.

Provider error

See definition in clause 7.6.1.

8.2.2 MAP_SEARCH_FOR_MS service

8.2.2.1 Definition

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

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

8.2.2.2 Service primitives

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

8.2.2.3 Parameter definitions and use

Invoke Id

See definition in clause 7.6.1.

IMSI

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

Current location area Id

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

User error

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

- absent subscriber;

  this error cause is returned by the MSC if the MS does not respond to the paging request;

- system failure;

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

- busy subscriber;

- unexpected data value.

Provider error

See definition in clause 7.6.1.

8.3 Access management services

8.3.1 MAP_PROCESS_ACCESS_REQUEST service

8.3.1.1 Definition

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

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

8.3.1.2 Service primitives

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

8.3.1.3 Parameter definitions and use

Invoke Id

See definition in clause 7.6.1.

CM service type

See definition in clause 7.6.9.

Access connection status

See definition in clause 7.6.9.

Current Location Area Id

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

Serving cell Id

See definition in clause 7.6.2.

TMSI

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

Cksn

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

IMSI

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

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

IMEI

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

MSISDN

See definition in clause 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.

User error

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

- unidentified subscriber;
- illegal subscriber;

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

  this error is sent if an IMEI check failed, i.e. the IMEI is blacklisted or not white-listed;
- roaming not allowed;
- this cause is used after VLR restart if the subscriber has no subscription for the current location area, e.g. due to regional subscription. The cause will be qualified by "location area not allowed" or "national roaming not allowed", respectively;
- unknown location area;
- system failure;
- unexpected data value.

Provider error

For definition of provider errors see clause 7.6.1.

8.4 Handover services

It should be noted that the handover services used on the B-interface have not been updated for Release 99. The B-interface is not fully operational specified. It is strongly recommended not to implement the B-interface as an external interface.

8.4.1 MAP_PREPARE_HANDOVER service

8.4.1.1 Definition

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

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

8.4.1.2 Service primitives

Table 8.4/1: MAP_PREPARE_HANDOVER

Parameter name	Request	Indication	Response	Confirm
Invoke Id	M	M(=)	M(=)	M(=)
Target Cell Id	C	C(=)		
Target RNC Id	C	C(=)		
HO-NumberNotRequired	C	C(=)		
IMSI	C	C(=)		
Integrity Protection Information	C	C(=)		
Encryption Information	C	C(=)		
Radio Resource Information	C	C(=)		
AN-APDU	C	C(=)	C	C(=)
Allowed GSM Algorithms	C	C(=)		
Allowed UMTS Algorithms	C	C(=)		
Radio Resource List	C	C(=)		
RAB ID	C	C(=)		
BSSMAP Service Handover	C	C(=)		
BSSMAP Service Handover List	C	C(=)		
RANAP Service Handover	C	C(=)		
ASCI Call Reference	C	C(=)		
Handover Number			C	C(=)
Relocation Number List			C	C(=)
Multicall Bearer Information			C	C(=)
Multiple Bearer Requested	C	C(=)		
Multiple Bearer Not Supported			C	C(=)
Selected UMTS Algorithms			C	C(=)
Chosen Radio Resource Information			C	C(=)
User error			C	C(=)
Provider error				O

8.4.1.3 Parameter use

Invoke Id

For definition of this parameter see clause 7.6.1.

Target Cell Id

For definition of this parameter see clause 7.6.2. This parameter is only included if the service is not in an ongoing transaction. This parameter shall also be excluded if the service is a part of the Inter-MSC SRNS Relocation procedure or the inter-system handover GSM to UMTS procedure described in 3GPP TS 23.009.

Target RNC Id

For definition of this parameter see clause 7.6.2. This parameter shall be included if the service is a part of the Inter-MSC SRNS Relocation procedure or the inter-system handover GSM to UMTS procedure described in 3GPP TS 23.009.

HO-Number Not Required

For definition of this parameter see clause 7.6.6.

IMSI

For definition of this parameter see clause 7.6.2. This UMTS parameter shall be included if:

- it is available and
- if the access network protocol is BSSAP and
- there is an indication that the MS also supports UMTS.

Integrity Protection Information

For definition of this parameter see clause 7.6.6. This UMTS parameter shall be included if available and if the access network protocol is BSSAP.

Encryption Information

For definition of this parameter see clause 7.6.6. This UMTS parameter shall be included if available and if the access network protocol is BSSAP.

Radio Resource Information

For definition of this parameter see clause 7.6.6. This GSM parameter shall be included if the access network protocol is RANAP and there is an indication that the UE also supports GSM. If the parameter Radio Resource List is sent , the parameter Radio Resource Information shall not be sent.

AN-APDU

For definition of this parameter see clause 7.6.9.

Allowed GSM Algorithms

For definition of this parameter see clause 7.6.6. This parameters includes allowed GSM algorithms. This GSM parameter shall be included if:

- the service is a part of the Inter-MSC SRNS Relocation procedure and
- Ciphering or Security Mode Setting procedure has been performed.and
- there is an indication that the UE also supports GSM.

Allowed UMTS Algorithms

For definition of this parameter see clause 7.6.6. This UMTS parameter shall be included if all of the following conditions apply:

- access network protocol is BSSAP and
- Integrity Protection Information and Encryption Information are not available and
- Ciphering or Security Mode Setting procedure has been performed.

Radio Resource List

For definition of this parameter see clause 7.6.6. This parameter shall be included if the access network protocol is RANAP and there is an indication that the UE also supports GSM. This parameter shall be sent when MSC-A requests multiple bearers to MSC-B. If the parameter Radio Resource Information is sent , the parameter Radio Resource List shall not be sent.

RAB ID

For definition of this parameter see subclause 7.6.2. This parameter shall be included when MSC-A supports multiple bearers and access network protocol is BSSAP and the RAB ID has a value other than 1.

BSSMAP Service Handover

For definition of this parameter see clause 7.6.6. It shall be present if it is available and the access network protocol is RANAP. If the parameter BSSMAP Service Handover List is sent, the parameter BSSMAP Service Handover shall not be sent.

BSSMAP Service Handover List

For definition of this parameter see clause 7.6.6. It shall be present if it is available and the access network protocol is RANAP. This parameter shall be sent when MSC-A requests multiple bearers to MSC-B. If the parameter BSSMAP Service Handover is sent, the parameter BSSMAP Service Handover List shall not be sent.

RANAP Service Handover

For definition of this parameter see clause 7.6.6. It shall be present if it is available and the access network protocol is BSSAP.

ASCI Call Reference

This parameter contains either the broadcast call reference or group call reference. It shall be included if a subscriber is undergoing handover during a VGCS or VBS call where MSC-B already has a Bearer established, so that MSC-B can determine the Group or Broadcast Call to which it shall attach the subscriber, see 3GPP TS 48.008 [49].

Handover Number

For definition of this parameter see clause 7.6.2. This parameter shall be returned at handover, unless the parameter HO-NumberNotRequired is sent. If the parameter Handover Number is returned, the parameter Relocation Number List shall not be returned.

Relocation Number List

For definition of this parameter see clause 7.6.2. This parameter shall be returned at relocation, unless the parameter HO-NumberNotRequired is sent. If the parameter Relocation Number List is returned, the parameter Handover Number shall not be returned.

Multicall Bearer Information

For a definition of this parameter see clause 7.6.2. This parameter shall be returned at relocation in the case that MSC-B supports multiple bearers.

Multiple Bearer Requested

For a definition of this parameter see clause 7.6.2. This parameter shall be sent when MSC-A requests multiple bearers to MSC-B.

Multiple Bearer Not Supported

For a definition of this parameter see clause 7.6.2. This parameter shall be returned at relocation when MSC-B receives Multiple Bearer Requested parameter and MSC-B does not support multiple bearers.

Selected UMTS Algorithms

For definition of this parameter see clause 7.6.6. This parameters includes the UMTS integrity and optionally encryption algorithms selected by RNC under the control of MSC-B. This UMTS parameter shall be included if the service is a part of the inter MSC inter system handover from GSM to UMTS.

Chosen Radio Resource Information

For definition of this parameter see clause 7.6.6. This parameter shall be returned at relocation if the encapsulated PDU is RANAP RAB Assignment Response and MS is in GSM access.

User error

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

- No handover number available.
- Target cell outside group call area;
- System failure.
- Unexpected data value.
- Data Missing.

Provider error

See definition of provider errors in clause 7.6.1.

8.4.2 MAP_SEND_END_SIGNAL service

8.4.2.1 Definition

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

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

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

8.4.2.2 Service primitives

Table 8.4/2: MAP_SEND_END_SIGNAL

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

8.4.2.3 Parameter use

Invoke Id

For definition of this parameter see clause 7.6.1.

AN-APDU

For definition of this parameter see clause 7.6.9.

Provider error

For definition of this parameter see clause 7.6.1.

8.4.3 MAP_PROCESS_ACCESS_SIGNALLING service

8.4.3.1 Definition

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

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

8.4.3.2 Service primitives

Table 8.4/3: MAP_PROCESS_ACCESS_SIGNALLING

Parameter name	Request	Indication
Invoke Id	M	M(=)
AN-APDU	M	M(=)
Selected GSM Algorithm	C	C(=)
Selected UMTS Algorithms	C	C(=)
Chosen Radio Resource Information	C	C(=)
Selected RAB id	C	C(=)

8.4.3.3 Parameter use

Invoke Id

For definition of this parameter see clause 7.6.1.

AN-APDU

For definition of this parameter see clause 7.6.9.

Selected GSM algorithm

For definition of this parameter see clause 7.6.6. This parameter shall be present if the encapsulated PDU is Security Mode Complete and MS is in GSM access.

Selected UMTS Algorithms

For definition of this parameter see clause 7.6.6. This parameters includes the UMTS integrity and optionally encryption algorithms selected by RNC under the control of MSC-B. This UMTS parameter shall be included if the encapsulated PDU is BSSMAP Cipher Mode Complete and the MS is in UMTS, or an interystem handover to UMTS is performed in MSC-B, or in the case of intra MSC-B intra UMTS relocation.

Chosen Radio Resource Information

For definition of this parameter see clause 7.6.6. This parameter shall be sent if the encapsulated PDU is RANAP RAB Assignment Response and MS is in GSM access.

Selected RAB ID

The selected radio access bearer that was kept at subsequent intra-MSC handover from UMTS to GSM after multiple bearers were used.

8.4.4 MAP_FORWARD_ACCESS_SIGNALLING service

8.4.4.1 Definition

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

The MAP_FORWARD_ACCESS_SIGNALLING service is a non-confirmed service using the primitives from table 8.4/4.

8.4.4.2 Service primitives

Table 8.4/4: MAP_FORWARD_ACCESS_SIGNALLING

Parameter name	Request	Indication
Invoke Id	M	M(=)
Integrity Protection Information	C	C(=)
Encryption Information	C	C(=)
Key Status	C	C(=)
AN-APDU	M	M(=)
Allowed GSM Algorithms	C	C(=)
Allowed UMTS Algorithms	C	C(=)
Radio Resource Information	C	C(=)
Radio Resource List	C	C(=)
BSSMAP Service Handover	C	C(=)
BSSMAP Service Handover List	C	C(=)
RANAP Service Handover	C	C(=)

8.4.4.3 Parameter use

For the definition and use of all parameters and errors, see clause 7.6.1.

Invoke Id

For definition of this parameter see clause 7.6.1.

Integrity Protection Information

For definition of this parameter see clause 7.6.6. This UMTS parameter shall be included if available and if the encapsulated PDU is BSSMAP Cipher Mode Command.

Encryption Information

For definition of this parameter see clause 7.6.6. This UMTS parameter shall be included if available and if the encapsulated PDU is BSSMAP Cipher Mode Command.

Key Status

For definition of this parameter see clause 7.6.6. This UMTS parameter shall be included if available and if the encapsulated PDU is BSSMAP Cipher Mode Command.

AN-APDU

For definition of this parameter see clause 7.6.9.

Allowed GSM Algorithms

This parameters includes allowed GSM algorithms. This GSM parameter shall be included if the encapsulated PDU is RANAP Security Mode Command and there is an indication that the UE also supports GSM.

Allowed UMTS Algorithms

For definition of this parameter see clause 7.6.6. This UMTS parameter shall be included if Integrity Protection Information and Encryption Information are not available and the encapsulated PDU is BSSMAP Cipher Mode Command.

Radio Resource Information

For definition of this parameter see clause 7.6.6. This parameter shall be sent if the encapsulated PDU is RANAP RAB Assignment Request. If the parameter Radio Resource List is sent, the parameter Radio Resource Information shall not be sent.

Radio Resource List

For definition of this parameter see clause 7.6.6. This parameter shall be sent if the encapsulated PDU is RANAP RAB Assignment Request and MSC-A requests modification of multiple bearers. If the parameter Radio Resource Information is sent, the parameter Radio Resource List shall not be sent.

BSSMAP Service Handover

For definition of this parameter see clause 7.6.6. It shall be present if it is available and the encapsulated PDU is RANAP RAB Assignment Request . If the parameter BSSMAP Service Handover List is sent, the parameter BSSMAP Service Handover shall not be sent.

BSSMAP Service Handover List

For definition of this parameter see clause 7.6.6. It shall be present if it is available and the encapsulated PDU is RANAP RAB Assignment Request and MSC-A requests modification of multiple bearers. If the parameter BSSMAP Service Handover is sent, the parameter BSSMAP Service Handover List shall not be sent.

RANAP Service Handover

For definition of this parameter see clause 7.6.6. It shall be present if it is available and the encapsulated PDU is BSSMAP Assignment Request.

8.4.5 MAP_PREPARE_SUBSEQUENT_HANDOVER service

8.4.5.1 Definition

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

The MAP_PREPARE_SUBSEQUENT_HANDOVER service is a confirmed service using the primitives from table 8.4/5.

8.4.5.2 Service primitives

Table 8.4/5: MAP_PREPARE_SUBSEQUENT_HANDOVER

Parameter name	Request	Indication	Response	Confirm
Invoke Id	M	M(=)	M(=)	M(=)
Target Cell Id	C	C(=)		
Target RNC Id	C	C(=)		
Target MSC Number	M	M(=)		
Selected RAB ID	C	C(=)		
AN-APDU	M	M(=)	C	C(=)
User error			C	C(=)
Provider error				O

8.4.5.3 Parameter use

Invoke Id

For definition of this parameter see clause 7.6.1.

Target Cell Id

For definition of this parameter see clause 7.6.2. This parameter shall be excluded if the service is a part of the Inter-MSC SRNS Relocation procedure or the inter-system handover GSM to UMTS procedure described in 3GPP TS 23.009.

Target RNC Id

For definition of this parameter see clause 7.6.2. This parameter shall be included if the service is a part of the Inter-MSC SRNS Relocation procedure or the inter-system handover GSM to UMTS procedure described in 3GPP TS 23.009.

Target MSC Number

For definition of this parameter see clause 7.6.2.

Selected RAB ID

For definition of this parameter see clause 7.6.2.

AN-APDU

For definition of this parameter see clause 7.6.9.

User error

For definition of this parameter see clause 7.6.1. The following error causes defined in clause 7.6.1 may be used, depending on the nature of the fault:

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

Provider error

For definition of this parameter see clause 7.6.1.

8.4.6 MAP_ALLOCATE_HANDOVER_NUMBER service

8.4.6.1 Definition

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

The MAP_ALLOCATE_HANDOVER_NUMBER service is a confirmed service using the primitives from table 8.4/6.

8.4.6.2 Service primitives

Table 8.4/6: MAP_ALLOCATE_HANDOVER_NUMBER

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

8.4.6.3 Parameter use

Invoke Id

For definition of this parameter see clause 7.6.1.

User error

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

- No handover number available.

Provider error

For definition of this parameter see clause 7.6.1.

8.4.7 MAP_SEND_HANDOVER_REPORT service

8.4.7.1 Definition

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

The MAP_SEND_HANDOVER_REPORT service is a confirmed service using the primitives from table 8.4/7.

8.4.7.2 Service primitives

Table 8.4/7: MAP_SEND_HANDOVER_REPORT

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

8.4.7.3 Parameter use

Invoke Id

For definition of this parameter see clause 7.6.1.

Handover Number

For definition of this parameter see clause 7.6.2.

Linked Id

For definition of this parameter see clause 7.6.1. This service is linked with MAP_ALLOCATE_HANDOVER_NUMBER.

Provider error

For definition of this parameter see clause 7.6.1.

8.5 Authentication management services

8.5.1 MAP_AUTHENTICATE service

The MAP_AUTHENTICATE service is used on the MAP B interface. This interface is not fully operational specified. It is strongly recommended not to implement the B-interface as an external interface.

8.5.1.1 Definition

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

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

8.5.1.2 Service primitives

The service primitives are shown in table 8.5/1.

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

8.5.1.3 Parameter use

Invoke id

See clause 7.6.1 for the use of this parameter.

RAND

See clause 7.6.7 for the use of this parameter.

CKSN

See clause 7.6.7 for the use of this parameter.

SRES

See clause 7.6.7 for the use of this parameter.

Provider error

See clause 7.6.1 for the use of this parameter.

8.5.2 MAP_SEND_AUTHENTICATION_INFO service

8.5.2.1 Definition

This service is used between the VLR and the HLR for the VLR to retrieve authentication information from the HLR. The VLR requests up to five authentication vectors.

Also this service is used between the SGSN and the HLR for the SGSN to retrieve authentication information from the HLR. The SGSN requests up to five authentication vectors.

If the user is a UMTS subscriber, the HLR shall return authentication quintuplets. If the user is a GSM subscriber, the HLR shall return authentication triplets.

If the HLR cannot provide the VLR or the SGSN with triplets, an empty response is returned. The VLR or the SGSN may then re-use old authentication triplets, except where this is forbidden under the conditions specified in GSM 03.20 [24].

If the HLR cannot provide the VLR or the SGSN with quintuplets, an empty response is returned. The VLR or the SGSN shall not re-use old authentication quintuplets.

If the VLR or SGSN receives a MAP_SEND_AUTHENTICATION_INFO response containing a User Error parameter as part of the handling of an authentication procedure, the authentication procedure in the VLR or SGSN shall fail.

Security related network functions are further described in GSM 03.20 and 3GPP TS 33.102.

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

8.5.2.2 Service primitives

The service primitives are shown in table 8.5/2.

Table 8.5/2: MAP_SEND_AUTHENTICATION_INFO parameters

Parameter name	Request	Indication	Response	Confirm
Invoke id	M	M(=)	M(=)	M(=)
IMSI	C	C(=)		
Number of requested vectors	C	C(=)		
Re-synchronisation Info	C	C(=)		
Segmentation prohibited indicator	C	C (=)		
Immediate response preferred indicator	U	C (=)		
AuthenticationSetList			C	C(=)
User error			C	C(=)
Provider error				O

8.5.2.3 Parameter use

Invoke id

See clause 7.6.1 for the use of this parameter.

IMSI

See clause 7.6.2 for the use of this parameter.

This parameter shall be present in the first (or only) request of the dialogue. If multiple service requests are present in a dialogue then this parameter shall not be present in any service request other than the first one.

Number of requested vectors

A number indicating how many authentication vectors the VLR or SGSN is prepared to receive. The HLR shall not return more vectors than indicated by this parameter.

This parameter shall be present in the first (or only) request of the dialogue. If multiple service requests are present in a dialogue then this parameter shall not be present in any service request other than the first one.

Re-synchronisation Info

For definition and use of this parameter see 3GPP TS 33.102.

If multiple service requests are present in a dialogue then this parameter shall not be present in any service request other than the first one.

Segmentation prohibited indicator

This parameter indicates if the VLR or SGSN allows segmentation of the response at MAP user level.

This parameter may be present only in the first request of the dialogue.

Immediate response preferred indicator

This parameter indicates that one of the requested authentication vectors is requested for immediate use in the VLR or SGSN. It may be used by the HLR together with the number of requested vectors and the number of vectors stored in the HLR to determine the number of vectors to be obtained from the AuC. It shall be ignored if the number of available vectors is greater than the number of requested vectors.

If multiple service requests are present in a dialogue then this parameter shall not be present in any service request other than the first one.

AuthenticationSetList

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 outcome of the service was successful.

User error

One of the following error causes defined in clause 7.6.1 shall be sent by the user in case of unsuccessful outcome of the service, depending on the respective failure reason:

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

Provider error

See clause 7.6.1 for the use of this parameter.

8.5.3 MAP_AUTHENTICATION_FAILURE_REPORT service

8.5.3.1 Definition

This service is used between the VLR and the HLR or between the SGSN or HLR for reporting of authentication failures.

8.5.3.2 Service primitives

The service primitives are shown in table 8.5/3.

Table 8.5/3: MAP_AUTHENTICATION_FAILURE_REPORT parameters

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

8.5.3.3 Parameter use

Invoke id

See clause 7.6.1 for the use of this parameter.

IMSI

See clause 7.6.2 for the use of this parameter.

Failure Cause

See clause 7.6.7 for use of this parameter.

User error

This parameter is sent by the responder upon unsuccessful outcome of the service, and then takes one of the following values defined in clause 7.6.1:

- Unknown Subscriber;
- System Failure;
- Unexpected Data Value.

Provider error

These are defined in clause 7.6.

8.6 Security management services

8.6.1 MAP_SET_CIPHERING_MODE service

8.6.1.1 Definitions

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

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

8.6.1.2 Service primitives

The service primitives are shown in table 8.6/1.

Table 8.6/1: MAP_SET_CIPHERING_MODE parameters

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

8.6.1.3 Parameter use

Invoke id

See clause 7.6.1 for the use of this parameter.

Ciphering mode

See clause 7.6.7 for the use of this parameter.

Kc

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

8.7 International mobile equipment identities management services

8.7.1 MAP_CHECK_IMEI service

8.7.1.1 Definition

This service is used between the VLR and the MSC and between the MSC and the EIR and between the SGSN and EIR 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 transferred to the EIR in the service request.

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

8.7.1.2 Service primitives

The service primitives are shown in table 8.7/1.

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

8.7.1.3 Parameter use

Invoke id

See clause 7.6.1 for the use of this parameter.

IMEI

See clause 7.6.2 for the use of this parameter. The parameter shall not be included in the service request between the VLR and the MSC, but is mandatory in the service request from the MSC to the EIR and from the SGSN to the EIR. It is not included in the service response from the EIR to the MSC or to the SGSN, but is mandatory in the service response from the MSC to the VLR on successful outcome.

Equipment status

See clause 7.6.4 for the use of this parameter. This parameter is sent by the responder in case of successful outcome of the service.

User error

One of the following error causes defined in clause 7.6.1 shall be sent by the user in case of unsuccessful outcome of the service, depending on the respective failure reason:

- unknown equipment;

  this error is returned by the responder when the IMEI is not known in the EIR;

- system failure;

- unexpected data value.

Provider error

See clause 7.6.1 for the use of this parameter.

8.7.2 MAP_OBTAIN_IMEI service

8.7.2.1 Definition

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

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

8.7.2.2 Service primitives

The service primitives are shown in table 8.7/2.

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

8.7.2.3 Parameter use

Invoke id

See clause 7.6.1 for the use of this parameter.

IMEI

See clause 7.6.2 for the use of this parameter. The parameter is included in the service response from the MSC to the VLR on successful outcome of the service.

User error

If the service fails, the VLR sends the user error System Failure (see clause 7.6.1) to the MSC.

Provider error

See clause 7.6.1 for the use of this parameter.

8.8 Subscriber management services

8.8.1 MAP-INSERT-SUBSCRIBER-DATA service

8.8.1.1 Definition

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

- the operator has changed the subscription of one or more supplementary services, basic services or data of a subscriber. Note that in case of withdrawal of a Basic or Supplementary service this primitive shall not be used;
- the operator has applied, changed or removed Operator Determined Barring;
- the subscriber has changed data concerning one or more supplementary services by using a subscriber procedure;
- the HLR provides the VLR with subscriber parameters at location updating of a subscriber or at restoration. In this case, this service is used to indicate explicitly that a supplementary service is not provisioned, if the supplementary service specification requires it. The only supplementary services which have this requirement are the CLIR and COLR services. Network access mode is provided only in restoration. If the Super-Charger functionality is supported the HLR may not need to provide the VLR with subscriber parameters at location updating of a subscriber. See TS 23.116.

Also this service is used by an HLR to update an SGSN with certain subscriber data in the following occasions:

- if the GPRS subscription has changed;
- if the network access mode is changed;
- the operator has applied, changed or removed Operator Determined Barring;
- the HLR provides the SGSN with subscriber parameters at GPRS location updating of a subscriber. If the Super-Charger functionality is supported the HLR may not need to provide the SGSN with subscriber parameters. See 3GPP TS 23.116.

It is a confirmed service and consists of the primitives shown in table 8.8/1.

8.8.1.2 Service primitives

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(=)		
MC-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(=)		
CS Allocation/Retention priority	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

8.8.1.3 Parameter use

All parameters are described in clause 7.6. The following clarifications are applicable:

Network access mode

This parameter defines if the subscriber has access to MSC/VLR and/or to SGSN. This parameter is used by SGSN and MSC/VLR. In VLR, the parameter is used only as part of Restore Data Procedure and the parameter is not stored in the VLR. This parameter shall always be sent to the SGSN as part of the GPRS subscriber data at GPRS location updating. It shall be sent to the SGSN if it is changed as a result of administrative action.

IMSI

It is only included if the service is not used in an ongoing transaction (e.g. location updating). This parameter is used by the VLR and the SGSN.

MSISDN

It is included either at location updating or when it is changed. The MSISDN sent shall be the basic MSISDN. This parameter is used by the VLR and the SGSN.

Category

It is included either at location updating or when it is changed. This parameter is used only by the VLR and if the SGSN receives this parameter it shall ignore it.

Subscriber Status

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

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

To remove all Operator Determined Barring Categories the Subscriber Status shall be set to "Service Granted". This parameter is used by the VLR and the SGSN.

Bearer service List

A list of Extensible Bearer service parameters (Extensible Bearer service is defined in clause 7.6). An Extensible Bearer service parameter must be the code for an individual Bearer service, except in the cases described below.

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

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

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

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

Teleservice List

A list of Extensible Teleservice parameters (Extensible Teleservice is defined in clause 7.6). An Extensible Teleservice parameter must be the code for an individual Teleservice.

If it is included in the Request/Indication, it contains either all Extensible Teleservices subscribed (at location updating or at restoration) or the ones added (at subscriber data modification). Only the Extensible Teleservices that are relevant to the node at which the message is received should be included in the Teleservice List.

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

Forwarding information List

A list of Extensible Forwarding information parameters (Extensible Forwarding information is defined in clause 7.6). It includes Call Forwarding services either at location updating or at restoration or when they are changed. Each Extensible Forwarding information parameter shall be treated independently of all other parameters in the primitive.

The Extensible Forwarding information shall include the SS-Code for an individual call forwarding supplementary service. The Extensible Forwarding information shall contain one or more Extensible Forwarding Features (Extensible Forwarding Feature is defined in clause 7.6).

The Extensible Forwarding Feature may include an Extensible Basic Service Group. This shall be interpreted according to the rules in clause 8.8.1.4.

The Extensible Forwarding Feature shall contain an Extensible SS-Status parameter.

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

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

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

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

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

Call barring information List

A list of Extensible Call barring information parameters (Extensible Call barring information is defined in clause 7.6). It includes Call Barring services either at location updating or at restoration or when they are changed. Each Extensible Call barring information parameter shall be treated independently of all other parameters in the primitive.

The Extensible Call barring information shall include the SS-Code for an individual call barring supplementary service. The Extensible Call barring information shall contain one or more Extensible Call Barring Features (Extensible Call Barring Feature is defined in clause 7.6).

The Extensible Call Barring Feature may include an Extensible Basic Service Group. This shall be interpreted according to the rules in clause 8.8.1.4.

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

If the VLR receives an Indication containing any Extensible Call Barring service codes which it does not support/allocate it returns them to the HLR in the parameter SS-Code List and discards the unsupported Extensible Call Barring service codes (no error is sent back). This parameter is used only by the VLR and if the SGSN receives this parameter it shall ignore it.

CUG information List

A list of CUG information list parameters (CUG information is defined in clause 7.6). It includes CUG information either at location updating or at restoration or when it is changed.

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

In any dialogue, the first insertSubscriberData message which contains CUG information shall include a non-empty CUG-SubscriptionList.

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

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

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

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

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

If the VLR does not support the CUG service it returns its code to the HLR in the parameter SS-Code List and discards the received information (no error is sent back). This parameter is used only by the VLR and if the SGSN receives this parameter it shall ignore it.

SS-Data List

A list of Extensible SS-Data parameters (Extensible SS-Data is defined in clause 7.6). It is sent for any other supplementary service than Call Forwarding, Call Barring, CUG and eMLPP either at location updating or at restoration or when they are changed. Each SS-Data parameter shall be treated independently of all other parameters in the primitive.

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

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

The SS-Data may include a Basic Service Group List. This shall be interpreted according to the rules in clause 8.8.1.4.

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

Operator Determined Barring General data

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

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

Operator Determined Barring HPLMN data

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

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

eMLPP Subscription Data

If included in the Insert Subscriber Data request this parameter defines the priorities the subscriber might apply for a call (as defined in clause 7.6). It contains both subparameters of eMLPP.

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

eMLPP subscription data that have been stored previously in a subscriber data record in the VLR are completely replaced by the new eMLPP subscription data received in a MAP_INSERT_SUBSCRIBER_DATA during either an Update Location or Restore Data procedure or a stand alone Insert Subscriber data procedure. This parameter is used only by the VLR and if the SGSN receives this parameter it shall ignore it.

MC Subscription Data

If included in the Insert Subscriber Data request, this parameter provides the MC Subscription Data as defined in clause 7.6.

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

MC subscription data that have been stored previously in a subscriber data record in the VLR are completely replaced by the new MC subscription data received in a MAP_INSERT_SUBSCRIBER_DATA during either an Update Location or Restore Data procedure or a stand alone Insert Subscriber data procedure. This parameter is used only by the VLR and if the SGSN receives this parameter it shall ignore it.

Roaming Restriction Due To Unsupported Feature

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

If this parameter is sent to the VLR the MSC area is restricted by the HLR and the VLR. This parameter is used only by the VLR and if the SGSN receives this parameter it shall ignore it.

Regional Subscription Data

If included in the Insert Subscriber Data request this parameter defines the subscriber's subscription area for the addressed VLR or for the addressed SGSN (as defined in clause 7.6). It contains the complete list of up to 10 Zone Codes that apply to a subscriber in the currently visited PLMN. The HLR shall send only those Zone Codes which are stored against the CC and NDC of the VLR or the CC and NDC of the SGSN to be updated.

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

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

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

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

- Too Many Zone Codes: more than 10 Zone Codes are to be stored in the VLR or in the SGSN.
- Regional Subscription Not Supported by the VLR or the SGSN.
- Zone Codes Conflict: the VLR or the SGSN detects that the zone codes indicate conflicting service permission for a location area.

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

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

If Regional Subscription Data are not included in any MAP_INSERT_SUBSCRIBER_DATA service, there is no restriction of roaming due to Regional Subscription. This parameter is used by the VLR and the SGSN.

Voice Broadcast Data

This parameter contains a list of group id's a user might have subscribed to; (VBS-Data is defined in clause 7.6). It includes VBS information either at location updating or at restoration or when it is changed.

At location updating, restoration or when there is a change in VBS data, the HLR shall include the complete VBS-Data.

When the VLR receives VBS-Data within a dialogue it shall replace the stored VBS-data with the received data set. All subsequent VBS-dta received within this dialogue shall be interpreted as add-on data.

If VBS-data is omitted in the Insert Subscriber Data operation the VLR shall keep the previously stored VBS data.

If the VLR detects that there is overlapping in the information received within a dialogue, it shall send the error Unexpected Data Value. This parameter is used only by the VLR and if the SGSN receives this parameter it shall ignore it.

Voice Group Call Data

This parameter contains a list of group id's a user might have subscribed to; see clause 7.6.

At location updating, restoration or when there is a change in VGCS data, the HLR shall include the complete VGCS-Data.

When the VLR receives VGCS-Data within a dialogue it shall replace the stored VGCS-Data with the received data set. All VGCS-Data received within this dialogue shall be interpreted as add-on data.

If VBCS-Data is omitted in the Insert Subscriber Data operation the VLR shall keep the previously stored VGCS-Data.

If the VLR detects that there is overlapping in the information received within a dialogue, it shall send the error Unexpected Data Value. This parameter is used only by the VLR and if the SGSN receives this parameter it shall ignore it.

North American Equal Access preferred Carrier Id List

A list of the preferred carrier identity codes that are subscribed to.

When the VLR receives this parameter from the HLR, it shall replace the previously stored preferred carrier identity codes with the received ones. It is not possible to delete all the preferred carrier identity codes from the VLR using this service. To delete all the preferred carrier identity codes from the VLR, the HLR shall use the MAP_CANCEL_LOCATION service.

LSA Information

If included in the ISD request, this parameter contains a list of localised service area identities a user might have subscribed to together with the priority, the preferential access indicator, the active mode support indicator and active mode indication of each localised service area; see clause 7.6. The access right outside these localised service areas is also indicated. In all cases mentioned below, the LSA information shall only include LSA Data applicable to the VPLMN where the Subscriber is located. The VLR number, received in the MAP-UPDATE_LOCATION primitive, or the SGSN number, received in the MAP_UPDATE_GPRS_LOCATION primitive, can be used, alongside data stored in the HLR, to determine the LSA Data applicable to the VPLMN.

At restoration, location updating or GPRS location updating the HLR shall include the complete set of applicable LSA Information.

When there is a change in LSA data the HLR shall include at least the new and/or modified LSA data.

When there is a change in the access right outside the localised service areas the HLR shall include the LSA only access indicator.

When the SGSN or the VLR receives LSA information within a dialogue it shall check if the received data has to be considered as the entire LSA information. If so, it shall replace the stored LSA information with the received data set, otherwise it shall replace the data only for the modified LSA data (if any) and/or access right, and add the new LSA data (if any) to the stored LSA Information.

If the entire LSA information is received, it shall always include the LSA only access indicator value together with the LSA data applicable for the PLMN (if any).

If LSA Information is omitted in the Insert Subscriber Data operation the SGSN or the VLR shall keep the previously stored LSA Information.

If the SGSN or the VLR detects that there is overlapping in the information received within a dialogue, it shall send the error Unexpected Data Value. This parameter is used by the VLR and the SGSN.

IST Alert Timer

This parameter contains the IST Alert timer value that must be used to inform the HLR about the call activities that the subscriber performs.

At Location Updating, restoration, or when there is a change in the IST data defined for the Subscriber, the HLR shall include the IST Alert timer.

LMU Identifier

This parameter indicates the presence of an LMU. This parameter is used only by the VLR and shall be ignored if received by an SGSN.

LCS Information

This parameter provides the following LCS related information for an MS subscriber:

- list of GMLCs in the HPLMN;
- privacy exception list;
- MO-LR list.

At restoration and location updating, the HLR shall include the complete LCS data of the subscriber.

When there is a change in LCS subscriber data the HLR shall include at least the new and/or modified LCS data. LCS data that is not modified need not be included.

The VLR shall keep any previously stored LCS Information that is not included in an Insert Subscriber Data operation.

If the VLR detects that there is overlapping in the LCS information received within a dialogue, it shall send the error Unexpected Data Value. However, if the VLR receives the LCS code in both the LCS Information and the SS-Data List, then the VLR shall not interpret this as overlapping data.

This parameter is used only by the VLR and shall be ignored if received by an SGSN.

Super-Charger Supported In HLR

This parameter is used by the HLR to indicate support for the Super-Charger functionality. If this parameter is present it shall include an indication of the age of the subscription data stored in the HLR.

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

SS-Code List

The list of SS-Code parameters for the services that are provided to a subscriber but are not supported/allocated by the VLR (SS-Code is defined in clause 7.6). The list can only include individual SS-Codes that were sent in the service request and SS-Codes for the eMLPP and/or CUG services if the above mentioned conditions, as described in eMLPP Subscription Data and/or CUG information List, are met (that is, eMLPP Subscription Data and/or CUG information List are received). This parameter is used only by the VLR.

Regional Subscription Response

If included in the response this parameter indicates one of:

- MSC Area Restricted entirely because of regional subscription;
- SGSN Area Restricted entirely because of regional subscription;
- Too Many Zone Codes to be inserted;
- Zone Codes Conflict;
- Regional Subscription not Supported by the VLR or by the SGSN.

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

If the SGSN determines after insertion of Regional Subscription Data that the entire SGSN area is restricted, the SGSN shall respond with a Regional Subscription Response indicating SGSN Area Restricted. Otherwise SGSN Area

Restricted is not sent. The HLR shall check whether the current SGSN area is no longer restricted. This parameter is used by the VLR and by the SGSN.

VLR CAMEL Subscription Info

This parameter is sent for subscribers who have CAMEL services which are invoked in the MSC. In CAMEL phase 1, this parameter contains only the O-CSI. In CAMEL Phase 2, this parameter may contain OCSI, SS-CSI and TIF-CSI. In CAMEL Phase 3, this parameter may contain O-CSI, D-CSI, SS-CSI, VT-CSI, SMS-CSI, M-CSI and TIF-CSI. In CAMEL Phase 2 TDP-Criteria for O-CSI may be associated with O-CSI. In CAMEL Phase 3, additionally, TDP-Criteria for VT-CSI may be associated with VT-CSI.. The VLR CAMEL Subscription Info is sent at location updating or when any information in the applicable CAMEL Subscription Info in the HLR has been changed.

At location updating, the complete set of VLR CAMEL Subscription Info is sent in one dialogue.

When CAMEL Subscription Information is changed in the HLR and changed data have to be sent to the VLR, then:

- for CAMEL Phase 1 and CAMEL Phase 2, the complete set of VLR CAMEL Subscription Info is sent in one dialogue;
- for CAMEL Phase 3, one or more specific elements of VLR CAMEL Subscription Info are sent in one dialogue.

When the VLR receives a specific element of VLR CAMEL Subscription Info, it shall overwrite the corresponding specific element of VLR CAMEL Subscription Info (if any) which it has stored for that subscriber.

For CAMEL Phase 1 and CAMEL Phase 2 , the VLR CAMEL Subscription Info consists of any one or more of:

- O-CSI (irrespective of the value of the “CAMEL Capability Handling” inside O-CSI),TDP-Criteria for O-CSI,SS-CSI and TIF-CSI.

  (The complete set of above shall be sent even if only one CSI has changed in case of stand alone ISD. The omitted elements of above list will be withdrawn in the VLR.)

From CAMEL phase 3 onwards, the specific elements of VLR CAMEL Subscription Info which may be sent are:

- O-CSI (irrespective of the value of the “CAMEL Capability Handling” inside O-CSI), TDP criteria for O-CSI, SS-CSI and TIF-CSI;

  (The complete set of above shall be sent even if only one CSI has changed in case of stand alone ISD. The omitted elements of above list will be withdrawn in the VLR.)

- D-CSI;
- VT-CSI;
- TDP-Criteria for VT-CSI;
- SMS-CSI;
- M-CSI.

If the VLR CAMEL Subscription Info is omitted in the Insert Subscriber Data operation the VLR shall keep the previously stored VLR CAMEL Subscription Info. Within one dialogue subsequent received data are interpreted as add-on data. If the VLR detects that there is overlapping in the information received within a dialogue, it shall send the error Unexpected Data Value. This parameter is used only by the VLR and if the SGSN receives this parameter it shall ignore it.

The VLR CAMEL Subscription Info may contain the TIF-CSI (Translation Information Flag) for CAMEL Phase 2 and 3. See 3GPP TS 23.072 for the use of this parameter and the conditions for its presence.

Supported CAMEL Phases

The use of this parameter and the requirements for its presence are specified in 3GPP TS 23.078. This parameter is used by the VLR and SGSN.

A VLR or SGSN not supporting any CAMEL Phase may omit this parameter.

GPRS Subscription Data

This parameter contains a list of PDP-contexts a user has subscribed to; see clause 7.6.

At GPRS location updating the HLR shall include the complete GPRS Subscription Data.

When there is a change in GPRS subscriber data the HLR shall include only the new and/or modified PDP contexts.

When the SGSN receives GPRS Subscription Data within a dialogue it shall check if the received data has to be considered as the entire GPRS subscription data. If so, it shall replace the stored GPRS Subscription Data with the received data set, otherwise it shall replace the data only for the modified PDP contexts (if any) and add the new PDP contexts (if any) to the stored GPRS Subscription Data.

If GPRS Subscription Data is omitted in the Insert Subscriber Data operation the SGSN shall keep the previously stored GPRS Subscription Data.

If the SGSN detects that there is overlapping in the information received within a dialogue, it shall send the error Unexpected Data Value. This parameter is used only by the SGSN and if the VLR receives this parameter it shall ignore it.

SGSN CAMEL Subscription Info

The SGSN CAMEL Subscription Info is sent at GPRS location updating or when any information in the applicable SGSN CAMEL Subscription Info in the HLR has been changed. In CAMEL Phase 3, this parameter may contain one or both of GPRS-CSI and SMS-CSI.

At GPRS location updating the complete set of SGSN CAMEL Subscription Info is sent.

When CAMEL Subscription Information is changed in the HLR and changed data have to be sent to the SGSN, then one or more specific elements of SGSN CAMEL Subscription Info are sent in one dialogue.

When the SGSN receives a specific element of SGSN CAMEL Subscription Info, it shall overwrite the corresponding specific element of SGSN CAMEL Subscription Info (if any) which it has stored for that subscriber.

The specific elements of SGSN CAMEL Subscription Info which may be sent are:

- SMS-CSI;
- GPRS-CSI.

This parameter is used only by the SGSN and if the VLR receives this parameter it shall ignore it.

Roaming Restricted In SGSN Due To Unsupported Feature

The HLR may decide to include this parameter in the request if certain services or features are indicated as not supported by the SGSN. This parameter is used only by the SGSN and if the VLR receives this parameter it shall ignore it.

CS Allocation/Retention priority

The CS Allocation/Retention priority is used only for Circuit Switched (CS). This parameter specifies relative importance to compare with other bearers about allocation and retention of bearer. This parameter is used only by the VLR and if the SGSN receives this parameter it shall ignore it.

User error

Only one of the following values is applicable:

- Unidentified subscriber;
- Data missing;
- Unexpected data value.

8.8.1.4 Basic service information related to supplementary services

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

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

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

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

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

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

- the supplementary service is applicable to at least one basic service in the Basic Service Group; and
- the subscriber has a subscription to at least one basic service in the Basic Service Group.

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

8.8.2 MAP-DELETE-SUBSCRIBER-DATA service

8.8.2.1 Definition

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

Also this service is used by an HLR to remove GPRS subscription data from a SGSN.

It is a confirmed service and consists of the primitives shown in table 8.8/2.

8.8.2.2 Service primitives

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(=)		
Specific CSI 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

8.8.2.3 Parameter use

All parameters are described in clause 7.6. The following clarifications are applicable:

Basic service List

A list of Extensible Basic service parameters (Extensible Basic service is defined in clause 7.6). It is used when one, several or all basic services are to be withdrawn from the subscriber. If the VLR or the SGSN receives a value for an Extensible Basic Service which it does not support, it shall ignore that value. This parameter is used by the VLR and by the SGSN.

SS-Code List

A list of SS-Code parameters (SS-Code is defined in clause 7.6). It is used when several or all supplementary services are to be withdrawn from the subscriber.

There are three possible options:

- deletion of basic service(s);

  The parameter Basic service List is only included.

- deletion of supplementary service(s);

  The parameter SS-Code List is only included.

- deletion of basic and supplementary services;

  Both Basic service List and SS-Code List are included.

This parameter is used only by the VLR and if the SGSN receives this parameter it shall ignore it.

Roaming Restriction Due To Unsupported Feature

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

If this parameter is sent the VLR shall check if the current Location Area is possibly allowed now. This parameter is used only by the VLR and if the SGSN receives this parameter it shall ignore it.

CAMEL Subscription Info Withdraw

This parameter is used to indicate that CAMEL Subscription Info shall be deleted from the VLR or from the SGSN. All CAMEL Subscription Info for the subscriber shall be deleted. This parameter is used by the VLR and by the SGSN. This parameter should not be sent in the same message as the Specific CSI Withdraw parameter.

Specific CSI Withdraw

This parameter is used to indicate that one or more specific elements of CAMEL Subscription Info shall be deleted from the VLR or from the SGSN.

The specific elements of CAMEL Subscription Info which may be withdrawn are:

- O-CSI with TDP criteria for O-CSI;

- SS-CSI;
- TIF-CSI;
- D-CSI;
- VT-CSI with TDP criteria for VT-CSI;
- SMS-CSI;
- M-CSI;
- GPRS-CSI.

This parameter is used by the VLR and by the SGSN. It shall not be sent to VLRs that do not support CAMEL phase 3. This parameter should not be sent in the same message as the CAMEL Subscription Info Withdraw parameter.

Regional Subscription Identifier

Contains one single Zone Code (as defined in clause 7.6) and is used if all Zone Codes shall be deleted from the subscriber data. When all the Zone Codes are deleted, the VLR or the SGSN shall check for its location areas whether they are allowed or not. If the whole MSC area is restricted, VLR will report it to HLR by returning the Regional Subscription Response "MSC Area Restricted". If the whole SGSN area is restricted, SGSN will report it to HLR by returning the Regional Subscription Response "SGSN Area Restricted".

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

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

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

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

VBS Group Indication

Contains an indication (flag) which is used if all Group Ids shall be deleted from the subscriber data for the Voice Broadcast teleservice.

If VBS is not supported in the VLR or no Group Ids are stored for VBS in the respective subscriber record, the request for deletion of all Group Ids shall be ignored. This parameter is used only by the VLR and if the SGSN receives this parameter it shall ignore it.

VGCS Group Indication

Contains an indication (flag) which is used if all Group Id's shall be deleted from the subscriber data for the Voice Group Call teleservice. This parameter is used only by the VLR and if the SGSN receives this parameter it shall ignore it.

If VGCS is not supported in the VLR or no Group Ids are stored for VGCS in the respective subscriber record, the request for deletion of all Group Ids shall be ignored.

GPRS Subscription Data Withdraw

This parameter is used to indicate whether all GPRS Subscription Data for the subscriber shall be deleted or if only a subset of the stored GPRS Subscription Data for the subscriber shall be deleted. In the latter case only those PDP contexts whose identifiers are included in the subsequent identifier list will be deleted. This parameter is used only by the SGSN and if the VLR receives this parameter it shall ignore it.

Roaming Restricted In SGSN Due To Unsupported Feature

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

If this parameter is sent the SGSN shall check if the current Location Area is possibly allowed now. This parameter is used only by the SGSN and if the VLR receives this parameter it shall ignore it.

LSA Information Withdraw

This parameter is used to indicate whether all LSA Information for the subscriber shall be deleted or if only a subset of the stored LSA Information for the subscriber shall be deleted. In the latter case only the LSA data whose LSA identities are included in the subsequent LSA data list will be deleted. This parameter is used by the VLR and the SGSN.

IST Information Withdraw

This parameter is used to indicate that the IST condition has been removed for the subscriber. See 3GPP TS 23.035 for the use of this parameter.

Regional Subscription Response

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

- MSC Area Restricted;
- SGSN Area Restricted;
- Regional Subscription Not Supported.

This parameter is used by the VLR and by the SGSN.

GMLC List Withdraw

This parameter indicates that the subscriber's LCS GMLC List shall be deleted from the VLR.

This parameter is used only by the VLR and shall be ignored if received by an SGSN.

User error

Only one of the following values is applicable:

- Unidentified subscriber;
- Data missing;
- Unexpected data value.

8.9 Identity management services

8.9.1 MAP-PROVIDE-IMSI service

8.9.1.1 Definition

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

It is a confirmed service and consists of the primitives shown in table 8.9/1.

8.9.1.2 Service primitives

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

8.9.1.3 Parameter use

All parameters are described in clause 7.6. The following clarifications are applicable:

IMSI

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

User error

Only one of the following values is applicable:

- Absent subscriber.

8.9.2 MAP-FORWARD-NEW-TMSI service

8.9.2.1 Definition

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

It is a confirmed service and consists of the primitives shown in table 8.9/2.

8.9.2.2 Service primitives

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

8.9.2.3 Parameter use

The parameter TMSI is described in clause 7.6.

8.10 Fault recovery services

8.10.1 MAP_RESET service

8.10.1.1 Definition

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

The MAP_RESET service is a non-confirmed service using the service primitives defined in table 8.10/1.

8.10.1.2 Service primitives

Table 8.10/1: MAP_RESET

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

HLR Id LIST	U	C(=)

8.10.1.3 Parameter definition and use

Invoke Id

See definition in clause 7.6.1.

HLR number

See definition in clause 7.6.2.

HLR Id LIST

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

8.10.2 MAP_FORWARD_CHECK_SS_INDICATION service

8.10.2.1 Definition

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

The MAP_FORWARD_CHECK_SS_INDICATION service is a non-confirmed service using the service primitives defined in table 8.10/2.

8.10.2.2 Service primitives

Table 8.10/2: MAP_FORWARD_CHECK_SS_INDICATION

Parameter name	Request	Indication
Invoke Id	M	M(=)

8.10.2.3 Parameter definition and use

Invoke Id

See definition in clause 7.6.1.

8.10.3 MAP_RESTORE_DATA service

8.10.3.1 Definition

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

The MAP_RESTORE_DATA service is a confirmed service using the service primitives defined in table 8.10/3.

8.10.3.2 Service primitives

Table 8.10/3: MAP_RESTORE_DATA

Parameter name	Request	Indication	Response	Confirm
Invoke Id	M	M(=)	M(=)	M(=)
IMSI	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 Network Entity	C	C(=)		
Long FTN Supported	C	C(=)		
HLR number			C	C(=)
MS Not Reachable Flag			C	C(=)
User error			C	C(=)
Provider error				O

8.10.3.3 Parameter definitions and use

Invoke Id

See definition in clause 7.6.1.

IMSI

See definition in clause 7.6.2.

LMSI

See definition in clause 7.6.2. It is an operator option to provide the LMSI from the VLR; it is mandatory for the HLR to support the LMSI handling procedures.

Supported CAMEL Phases

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

SoLSA Support Indicator

This parameter is used by the VLR to indicate to the HLR in the Restore Data indication that SoLSA is supported. If this parameter is not included in the Restore Data indication then the HLR shall not perform any specific error handling.

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

IST Support Indicator

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

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

Long FTN Supported

This parameter indicates that the VLR supports Long Forwarded-to Numbers.

Super-Charger Supported in Serving Network Entity

This parameter is used by the VLR to indicate to the HLR that the VLR supports the Super-Charger functionality and that subscriber data is required.

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

HLR number

See definition in clause 7.6.2. The presence of this parameter is mandatory in case of successful outcome of the service.

MS Not Reachable Flag

See definition in clause 7.6.8. This parameter shall be present in case of successful outcome of the service, if the "MS Not Reachable flag" was set in the HLR.

User error

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

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

Provider error

For definition of provider errors see clause 7.6.1.

8.11 Subscriber Information services

8.11.1 MAP-ANY-TIME-INTERROGATION service

8.11.1.1 Definition

This service is used by the gsmSCF, to request information (e.g. subscriber state and location) from the HLR or the GMLC at any time.

When this service is used to the HLR, the subscriber state or location may be requested.

When this service is used to the GMLC, only the location may be requested.

The MAP-ANY-TIME-INTERROGATION service is a confirmed service using the service primitives defined in table 8.11/1.

8.11.1.2 Service primitives

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

8.11.1.3 Parameter definition and use

All parameters are described in clause 7.6.

The HLR or GMLC may be able to use the value of the parameter gsmSCF-address to screen a MAP_Any_Time_Interrogation indication.

The use of the parameters and the requirements for their presence are specified in 3GPP TS 23.078.

User error

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

- System Failure;
- Any Time Interrogation Not Allowed;
- Data Missing;
- Unexpected Data Value;
- Unknown Subscriber.

Provider error

These are defined in clause 7.6.1.

8.11.2 MAP-PROVIDE-SUBSCRIBER-Info service

8.11.2.1 Definition

This service is used to request information (e.g. subscriber state and location) from the VLR at any time.

The MAP-PROVIDE-SUBSCRIBER-Info service is a confirmed service using the primitives defined in table 8.11/2.

8.11.2.2 Service primitives

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

8.11.2.3 Parameter definition and use

All parameters are defined in clause 7.6. The use of these parameters and the requirements for their presence are specified in 3GPP TS 23.018 [97].

User error

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

- Data Missing;
- Unexpected Data Value.

Provider error

These are defined in clause 7.6.1.

8.11.3 MAP-ANY-TIME-SUBSCRIPTION-INTERROGATION service

8.11.3.1 Definition

This service is used by the gsmSCF, to request subscription information (e.g. call forwarding supplementary service data or CSI) from the HLR at any time.

8.11.3.2 Service primitives

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(=)		
Long FTN Supported	C	C(=)		
Call Forwarding Data			C	C(=)
Call Barring Data			C	C(=)
ODB Info			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

8.11.3.3 Parameter definition and use

All parameters are described in clause 7.6.

The HLR may be able to use the value of the parameter gsmSCF-address to screen a MAP_Any_Time_Subscription_Interrogation indication.

The use of the parameters and the requirements for their presence are specified in 3GPP TS 23.078.

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;
- BearerServiceNotProvisioned;
- TeleserviceNotProvisioned;
- CallBarred;
- IllegalSS-Operation;

- SS-NotAvailable;
- InformationNotAvailable;
- Any Time Subscription Interrogation Not Allowed;
- Data Missing.

Provider error

These are defined in clause 7.6.1.

8.11.4 MAP-ANY-TIME-MODIFICATION service

8.11.4.1 Definition

This service is used by the gsmSCF, to modify information of the HLR at any time.

8.11.4.2 Service primitives

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(=)		
Long FTN Supported	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

8.11.4.3 Parameter definition and use

All parameters are described in clause 7.6.

The HLR may be able to use the value of the parameter gsmSCF-address to screen a MAP_Any_Time_Modification indication.

The use of these parameters and the requirements for their presence are specified in 3GPP TS 23.078.

User error

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

- Any Time Modification Not Allowed;
- Data Missing;
- Unexpected Data Value;
- Unknown Subscriber;
- Bearer service not provisioned;

  This error is returned only if not even a subset of the requested bearer service group has been subscribed to;

- Teleservice not provisioned;

This error is returned only if not even a subset of the requested teleservice group has been subscribed to;

- Call Barred;
- Illegal SS operation;
- SS error status;
- SS incompatibility;
- SS subscription violation;
- Information Not Available.

Provider error

These are defined in clause 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(=)		
Ext Forwarding information-for-CSE	C	C(=)		
Ext Call barring information-for-CSE	C	C(=)		
ODB info	C	C(=)		
CAMEL subscription info	C	C(=)		
All Information Sent	C	C(=)		
User error			C	C(=)
Provider error				O

8.11.5.3 Parameter definition and use

Invoke id

See clause 7.6.1 for the use of this parameter.

IMSI

See clause 7.6.2 for the use of this parameter.

MSISDN

See clause 7.6.2 for the use of this parameter.

Ext Forwarding information-for-CSE

See clause 7.6.3 for the use of this parameter. The use of this parameter and the requirements for their presence are specified in 3GPP TS 23.078.

Ext Call barring information-for-CSE

See clause 7.6.3 for the use of this parameter. The use of this parameter and the requirements for their presence are specified in 3GPP TS 23.078.

ODB Info

See clause 7.6.3 for the use of this parameter. The use of this parameter and the requirements for their presence are specified in 3GPP TS 23.078.

CAMEL subscription info

See clause 7.6.3 for the use of this parameter. The use of this parameter and the requirements for their presence are specified in 3GPP TS 23.078.

All Information Sent

This parameter is set when the HLR has sent all information to gsmSCF.

User error

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

- Data Missing;
- Unexpected Data Value;
- Unknown Subscriber.

Provider error

These are defined in clause 7.6.1.

The use of the parameters and the requirements for their presence are specified in 3GPP TS 23.078.

9 Operation and maintenance services

9.1 Subscriber tracing services

9.1.1 MAP-ACTIVATE-TRACE-MODE service

9.1.1.1 Definition

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

Also this service is used between the HLR and the SGSN to activate subscriber tracing in the SGSN.

The MAP-ACTIVATE-TRACE-MODE service is a confirmed service using the primitives from table 9.1/1.

9.1.1.2 Service primitives

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

9.1.1.3 Parameter use

Invoke id

See definition in clause 7.6.1.

IMSI

See definition in clause 7.6.2. The IMSI is a mandatory parameter in a stand-alone operation.

Trace reference

See definition in clause 7.6.10.

Trace type

See definition in clause 7.6.10.

OMC Id

See definition in clause 7.6.2. The use of this parameter is an operator option.

User error

The following errors defined in clause 7.6.1 may be used, depending on the nature of the fault:

- Unidentified Subscriber;
- Facility Not Supported;
- Tracing Buffer Full;
- System Failure;
- Unexpected Data Value;
- Data missing.

Provider error

For definition of provider errors see clause 7.6.1.

9.1.2 MAP-DEACTIVATE-TRACE-MODE service

9.1.2.1 Definition

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

Also this service is used between the SGSN and the HLR for deactivating subscriber tracing in the SGSN.

The MAP-DEACTIVATE-TRACE-MODE service is a confirmed service using the primitives from table 9.1/2.

9.1.2.2 Service primitives

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

9.1.2.3 Parameter use

Invoke id

See definition in clause 7.6.1.

IMSI

See definition in clause 7.6.2. The IMSI is a mandatory parameter in a stand-alone operation.

Trace reference

See definition in clause 7.6.10.

User error

The following errors defined in clause 7.6.1 may be used, depending on the nature of the fault:

- Unidentified Subscriber;
- Facility Not Supported;
- System Failure;
- Unexpected Data Value;
- Data missing.

Provider error

For definition of provider errors see clause 7.6.1.

9.1.3 MAP-TRACE-SUBSCRIBER-ACTIVITY service

9.1.3.1 Definition

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

The MAP-TRACE-SUBSCRIBER-ACTIVITY service is a non-confirmed service using the primitives from table 9.1/3.

9.1.3.2 Service primitives

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(=)

9.1.3.3 Parameter use

Invoke id

See definition in clause 7.6.1.

IMSI

See definition in clause 7.6.2. The controlling MSC shall provide either the IMSI or the IMEI to the servicing MSC.

Trace reference

See definition in clause 7.6.10.

Trace type

See definition in clause 7.6.10.

OMC Id

See definition in clause 7.6.2. The use of this parameter is an operator option.

9.2 Other operation and maintenance services

9.2.1 MAP-SEND-IMSI service

9.2.1.1 Definition

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

It is a confirmed service and consists of the primitives shown in table 9.2/1.

9.2.1.2 Service primitives

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

9.2.1.3 Parameter use

All parameters are described in clause 7.6. The following clarifications are applicable.

User error

Only one of the following values is applicable:

- Unknown subscriber;
- Unexpected data value;
- Data missing.

10 Call handling services

10.1 MAP_SEND_ROUTING_INFORMATION service

10.1.1 Definition

This service is used between the Gateway MSC and the HLR. The service is invoked by the Gateway MSC to perform the interrogation of the HLR in order to route a call towards the called MS.

This is a confirmed service using the primitives listed in table 10.1/1.

This service is also used between the GMSC and the NPLR.

10.1.2 Service primitives

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(=)		
Long FTN Supported	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 Carrier Id			U	C(=)
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

10.1.3 Parameter use

See clause 7.6 for a definition of the parameters used in addition to the following. Note that:

- a conditional parameter whose use is defined only in 3GPP TS 23.078 shall be absent if the sending entity does not support CAMEL;

- a conditional parameter whose use is defined only in 3GPP TS 23.079 [99] shall be absent if the sending entity does not support optimal routeing;

- a conditional parameter whose use is defined only in 3GPP TS 23.078 & 3GPP TS 23.079 [99] shall be absent if the sending entity supports neither CAMEL nor optimal routeing.

Interrogation Type

See 3GPP TS 23.079 [99] for the use of this parameter.

GMSC address

The E.164 address of the GMSC.

MSISDN

This is the Mobile Subscriber ISDN number assigned to the called subscriber. In the Request & Indication it is the number received by the GMSC in the IAM. If the call is to be forwarded and the HLR supports determination of the redirecting number, the HLR inserts the basic MSISDN in the Response.

See 3GPP TS 23.066 [108] for the use of this parameter and the conditions for its presence in the response.

OR Interrogation

See 3GPP TS 23.079 [99] for the use of this parameter and the conditions for its presence.

OR Capability

See 3GPP TS 23.079 [99] for the use of this parameter and the conditions for its presence.

CUG Interlock

See 3GPP TS 23.018 [97] for the use of this parameter and the conditions for its presence.

CUG Outgoing Access

See 3GPP TS 23.018 [97] for the use of this parameter and the conditions for its presence.

Number of Forwarding

See 3GPP TS 23.018 [97] for the use of this parameter and the conditions for its presence.

Network Signal Info

See 3GPP TS 23.018 [97] for the conditions for the presence of the components of this parameter.

Supported CAMEL Phases

The use of this parameter and the requirements for its presence are specified in 3GPP TS 23.078.

T-CSI Suppression

The use of this parameter and the requirements for its presence are specified in 3GPP TS 23.078.

Suppression Of Announcement

The use of this parameter and the requirements for its presence are specified in 3GPP TS 23.078.

Call Reference Number

The use of this parameter and the conditions for its presence are specified in 3GPP TS 23.078 [98] and 3GPP TS 23.079 [99].

Forwarding Reason

See 3GPP TS 23.079 [99] for the use of this parameter and the conditions for its presence.

Basic Service Group

See 3GPP TS 23.079 [99] for the use of this parameter and the conditions for its presence.

Alerting Pattern

See 3GPP TS 23.018 [97] for the use of this parameter and the conditions for its presence.

CCBS Call

See 3GPP TS 23.093 [107] for the use of this parameter and the conditions for its presence.

Supported CCBS Phase

This parameter indicates by its presence that CCBS is supported and the phase of CCBS which is supported.

Additional Signal Info

See 3GPP TS 23.081 [27] for the conditions for the presence of the components of this parameter.

IST Support Indicator

This parameter is used to indicate to the HLR that the GMSC supports basic IST functionality, that is, the GMSC is able to terminate the subscriber call activity that originated the IST Alert when it receives the IST Alert response indicating that the call(s) shall be terminated. If this parameter is not included in the Send Routing Information indication and the subscriber is marked as an IST subscriber, then the HLR may limit the service for the call (by barring the incoming call if it is not subject to forwarding, or suppressing Call Forwarding from the GMSC), or allow the call assuming the associated risk of not having the basic IST mechanism available.

This parameter can also indicate that the GMSC supports the IST Command, including the ability to terminate all calls being carried for the identified subscriber by using the IMSI as a key. If this additional capability is not included in the Send Routing Information indication and the subscriber is marked as an IST subscriber, then the HLR may limit the service for the subscriber (by barring the incoming calls if they are not subject to forwarding, or suppressing Call Forwarding from the GMSC), or allow the incoming calls assuming the associated risk of not having the IST Command mechanism available.

Pre-paging supported

See 3GPP TS 23.018 for the use of this parameter and the conditions for its presence.

Call Diversion Treatment Indicator

This parameter indicates whether or not call diversion is allowed.

IMSI

See 3GPP TS 23.018 [97] and 3GPP TS 23.066 [108] for the use of this parameter and the conditions for its presence.

MSRN

See 3GPP TS 23.018 [97], 3GPP TS 23.066 [108] and 3GPP TS 23.079 [99] for the use of this parameter and the conditions for its presence. If the NPLR returns only the MSISDN-number without Routeing Number to the GMSC, the MSISDN-number shall be returned as MSRN.

Forwarding Data

This parameter includes a number to define the forwarded-to destination, the forwarding reason and the forwarding options Notification to calling party and Redirecting presentation, and can include the forwarded-to subaddress. See 3GPP TS 23.018 [97] and 3GPP TS 23.079 [99] for the conditions for the presence of its components.

Forwarding Interrogation Required

See 3GPP TS 23.079 [99] for the use of this parameter and the conditions for its presence.

Long FTN Supported

This parameter indicates that the GMSC supports Long Forwarded-to Numbers.

VMSC address

See 3GPP TS 23.079 [99] for the use of this parameter and the conditions for its presence.

GMSC CAMEL Subscription Info

The use of this parameter and the requirements for its presence are specified in 3GPP TS 23.078.

Location Information

The use of this parameter and the requirements for its presence are specified in 3GPP TS 23.078.

Subscriber State

The use of this parameter and the requirements for its presence are specified in 3GPP TS 23.078.

CUG Subscription Flag

The use of this parameter and the requirements for its presence are specified in 3GPP TS 23.078.

North American Equal Access preferred Carrier Id

This parameter is returned to indicate the preferred carrier identity to be used to set-up the call (i.e. forwarding the call or establishing the roaming leg).

SS-List

This parameter includes SS-codes and will be returned as an operator option. The HLR shall not send PLMN-specific SS-codes across PLMN boundaries. However if the GMSC receives PLMN-specific SS-codes from a foreign PLMN's HLR the GMSC may ignore it. If the GMSC attempts to process the PLMN- specific SS- codes, this may lead to unpredictable behaviour but the GMSC shall continue call processing.

Basic Service Code

The use of this parameter and the requirements for its presence are specified in 3GPP TS 23.078.

If the CAMEL service is not involved, this parameter includes the basic service code and will be returned as an operator option. The HLR shall not send a PLMN-specific Basic Service Code across PLMN boundaries. However if the GMSC receives a PLMN-specific Basic Service Code from a foreign PLMN's HLR the GMSC may ignore it. If the GMSC attempts to process the PLMN specific Basic Service codes, this may lead to unpredictable behaviour but the GMSC shall continue call processing.

CCBS Target

See 3GPP TS 23.093 [107] for the use of this parameter and the conditions for its presence.

Keep CCBS Call Indicator

See 3GPP TS 23.093 [107] for the use of this parameter and the conditions for its presence.

IST Alert Timer

It includes the IST Alert timer value that must be used to inform the HLR about the call activities that the subscriber performs. This parameter is only sent to the GMSC in response to a Send Routing Information request which indicates the the GMSC supports IST.

Number Portability Status

This parameter indicates the number portability status of the subscriber. This parameter may be present if the sender of SRIack is NPLR.

User error

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

- Unknown Subscriber;

  The diagnostic for the Unknown Subscriber error may indicate “NPDB Mismatch”.

- Number changed;

- Call Barred;

  This error will indicate that either incoming calls are barred for this MS or that calls are barred due to Operator Determined Barring (see 3GPP TS 22.041 for a definition of this network feature);

- CUG Reject;

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

- Bearer Service Not Provisioned;
- Teleservice Not Provisioned;

  A subscription check has been performed and the call has not passed the check due to incompatibility with regard to the requested service. Depending on the nature of the incompatibility, either of these messages will be returned;

- Facility Not Supported;
- Absent Subscriber;

  This indicates that the location of the MS is not known (either the station is not registered and there is no location information available or the Provide Roaming Number procedure fails due to IMSI detached flag being set), or the GMSC requested forwarding information with a forwarding reason of not reachable, and the call forwarding on MS not reachable service is not active;

- Busy Subscriber;

  This indicates that Call Forwarding on Busy was not active for the specified basic service group when the GMSC requested forwarding information with a forwarding reason of busy;

  The error may also indicate that the subscriber is busy due to an outstanding CCBS recall. In the error data it may then be specified that CCBS is possible for the busy encountered call;

- No Subscriber Reply;

  This indicates that Call Forwarding on No Reply was not active for the specified basic service group when the GMSC requested forwarding information with a forwarding reason of no reply;

- OR Not Allowed;

  This indicates that the HLR is not prepared to accept an OR interrogation from the GMSC, or that calls to the specified subscriber are not allowed to be optimally routed;

- Forwarding Violation;
- System Failure;
- Data Missing;
- Unexpected Data Value.

See clause 7.6 for a definition of these errors.

Provider error

These are defined in clause 7.6.

10.2 MAP_PROVIDE_ROAMING_NUMBER service

10.2.1 Definition

This service is used between the HLR and VLR. The service is invoked by the HLR to request a VLR to send back a roaming number to enable the HLR to instruct the GMSC to route an incoming call to the called MS.

This is a confirmed service which uses the primitives described in table 10.2/1.

10.2.2 Service primitives

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(=)		
Long FTN Supported	C	C(=)		
Roaming Number			C	C(=)
User error			C	C(=)
Provider error				O

10.2.3 Parameter use

See clause 7.6 for a definition of the parameters used, in addition to the following. Note that:

- a conditional parameter whose use is defined only in 3GPP TS 23.078 shall be absent if the sending entity does not support CAMEL;
- a conditional parameter whose use is defined only in 3GPP TS 23.079 shall be absent if the sending entity does not support optimal routeing;
- a conditional parameter whose use is defined only in 3GPP TS 23.078 & 3GPP TS 23.079 shall be absent if the sending entity supports neither CAMEL nor optimal routeing.

IMSI

This is the IMSI of the called Subscriber.

MSC Number

This is the ISDN number assigned to the MSC currently serving the MS. The MSC number will have been stored in the HLR as provided at location updating.

MSISDN

See 3GPP TS 23.018 [97] for the use of this parameter and the conditions for its presence.

LMSI

See 3GPP TS 23.018 [97] for the use of this parameter and the conditions for its presence.

GSM Bearer Capability

See 3GPP TS 23.018 [97] for the use of this parameter and the conditions for its presence.

This information is passed according to the rules specified in TS 3GPP TS 29.007 [56].

There may be two GSM Bearer Capabilities supplied.

Network Signal Info

See 3GPP TS 23.018 [97] for the conditions for the presence of the components of this parameter.

Suppression Of Announcement

The use of this parameter and the requirements for its presence are specified in 3GPP TS 23.078.

Call Reference Number

The use of this parameter and the conditions for its presence are specified in 3GPP TS 23.078 [98] and 3GPP TS 23.079 [99].

GMSC Address

The use of this parameter and the conditions for its presence are specified in 3GPP TS 23.078 [98] and 3GPP TS 23.079 [99].

OR Interrogation

See 3GPP TS 23.079 [99] for the use of this parameter and the conditions for its presence.

OR Not Supported in GMSC

See 3GPP TS 23.079 [99] for the use of this parameter and the conditions for its presence.

Supported CAMEL Phases in GMSC

See 3GPP TS 23.078 [98] for the use of this parameter and the conditions for its presence.

Alerting Pattern

See 3GPP TS 23.078 [98] for the use of this parameter and the conditions for its presence.

CCBS Call

See 3GPP TS 23.093 [107] for the use of this parameter and the conditions for its presence.

Additional Signal Info

See 3GPP TS 23.081 [27] for the conditions for the presence of the components of this parameter.

Pre-paging supported

See 3GPP TS 23.018 for the use of this parameter and the conditions for its presence.

Long FTN supported

See 3GPP TS 23.082 for the use of this parameter and the conditions for its presence.

Roaming Number

See 3GPP TS 23.018 [97] for the use of this parameter and the conditions for its presence.

User error

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

- Absent Subscriber;

  This error will be returned if the IMSI detach flag is set.

- No Roaming Number Available;

- OR Not Allowed;

This indicates that the MAP_PROVIDE_ROAMING_NUMBER indication included the OR interrogation indicator, but the VLR does not support optimal routeing.

- Facility Not Supported;
- System Failure;
- Data Missing;
- Unexpected Data Value.

See clause 7.6 for a definition of these reasons.

Provider error

These are defined in clause 7.6.

10.3 MAP_RESUME_CALL_HANDLING service

10.3.1 Definition

This service is used between the terminating VMSC and the GMSC. The service is invoked by the terminating VMSC to request the GMSC to resume handling the call and forward it to the specified destination.

This is a confirmed service which uses the Primitives listed in table 10.3/1.

10.3.2 Service primitives

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

10.3.3 Parameter use

Information received in subsequent segment of a segmented dialogue shall not overwrite information received in an earlier segment.

See clause 7.6 for a definition of the parameters used, in addition to the following.

Call Reference Number

See 3GPP TS 23.079 [99] for the use of this parameter. This parameter shall be present in a first segment of the dialogue.

Basic Service Group

See 3GPP TS 23.079 [99] for the use of this parameter. This parameter shall be present in a first segment of the dialogue.

IMSI

This is the IMSI of the forwarding Subscriber. This parameter shall be present in a first segment of the dialogue.

Forwarding Data

This parameter includes a number to define the forwarded-to destination, the forwarding reason and the forwarding options Notification to calling party and Redirecting presentation, and can include the forwarded-to subaddress. See 3GPP TS 23.079 [99] for the conditions for the presence of its components. This parameter shall be present in a first segment of the dialogue.

CUG Interlock

See 3GPP TS 23.079 [99] for the use of this parameter and the conditions for its presence.

CUG Outgoing Access

See 3GPP TS 23.079 [99] for the use of this parameter and the conditions for its presence.

O-CSI

See 3GPP TS 23.078 for the use of this parameter and the conditions for its presence.

For CAMEL phases 1 & 2, the O-CSI shall contain only one set of O-BCSM TDP data.

D-CSI

The Dialled Services-CSI.

See 3GPP TS 23.078 for the use of this parameter and the conditions for its presence.

CCBS Target

See 3GPP TS 23.093 [107] for the use of this parameter and the conditions for its presence.

UU Data

See 3GPP TS 23.087 for the use of this parameter and the conditions for its presence.

UUS CF Interaction

See 3GPP TS 23.087 for the use of this parameter and the conditions for its presence.

All Information Sent

This parameter is set when the VMSC has sent all information to GMSC.

MSISDN

This parameter is the basic MSISDN of the forwarding subscriber. It shall be present if the VMSC supports determination of the redirecting number.

User error

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

- Optimal Routeing not allowed;
- Forwarding failed;
- Unexpected Data Value;
- Data Missing.

Provider error

These are defined in clause 7.6.

10.4 MAP_PREPARE_GROUP_CALL service

10.4.1 Definition

This service is used by the Anchor_MSC to inform the Relay_MSC about a group call set-up.

The MAP_PREPARE_GROUP_CALL service is a confirmed service using the service primitives given in table 10.4/1.

10.4.2 Service primitives

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

10.4.3 Parameter definitions and use

Invoke Id

See definition in clause 7.6.1.

Teleservice

Voice Broadcast Service or Voice Group Call Service.

ASCI Call Reference

Broadcast call reference or group call reference. This item is used to access the VBS-GCR or VGCS-GCR within the Relay_MSC.

Ciphering Algorithm

The ciphering algorithm to be used for the group call.

Group Key Number

This number has to be broadcasted and is used by the mobile station to select the chosen group key.

Shall be present if the ciphering applies.

Group Key

This key is used for ciphering on the radio interface.

Shall be present if the ciphering applies.

Priority

Default priority level related to the call if eMLPP applies.

CODEC-Information

Information on the codecs allowed for this call.

Uplink Free Indicator

A flag indicating whether the call is initiated from a dispatcher.

Group Call Number

This temporary allocated E.164 number is used for routing the call from the Anchor MSC to the Relay MSC.

User Error

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

- No Group Call Number available;
- System Failure;
- Unexpected Data Value.

Provider Error

See definition of provider error in clause 7.6.1.

10.5 MAP_PROCESS_GROUP CALL_SIGNALLING service

10.5.1 Definitions

This service is used between Relay MSC and Anchor MSC for transmission of Group Call notifications.

The MAP_PROCESS_GROUP_CALL_SIGNALLING service is a non-confirmed service using the service primitives given in table 10.5/1.

10.5.2 Service primitives

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(=)

10.5.3 Parameter definitions and use

Invoke Id

See definition in clause 7.6.1

Uplink Request

This information element indicates to the anchor MSC that a service subscriber roaming in the relay MSC area requests access to the uplink.

Uplink Release Indication

This information element if included by the Relay MSC indicates to the Anchor MSC that the uplink has become free.

Release Group Call

This information element if included by the Relay MSC indicates to the Anchor MSC that the service subscriber who has initiated the call and who currently has access to the uplink terminates the call.

10.6 MAP_FORWARD_GROUP_CALL_SIGNALLING service

10.6.1 Definitions

This service is used between Anchor MSC and Relay MSC for transmission of Group Call notifications.

The MAP_FORWARD_GROUP_CALL_SIGNALLING service is a non-confirmed service using the service primitives given in table 10.6/1.

10.6.2 Service primitives

Table 10.6/1: 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(=)
State Attributes	C	C(=)

10.6.3 Parameter definitions and use

IMSI

Identity of the service subscriber who has established the call and who is allowed to terminate the call.

Invoke Id

See definition in clause 7.6.1.

Uplink Request Acknowledgement

This information element is used for positive acknowledgement of an uplink request.

Uplink Release Indication

This information element if included by the Anchor MSC indicates to the Relay MSC that the uplink has become free.

Uplink Reject Command

This information element is used for negative acknowledgement of an uplink request.

Uplink Seized Command

This information element if included by the Anchor MSC indicates to the Relay MSC that the uplink is no longer free.

Uplink Release Command

This information element if included by the Anchor MSC indicates to the Relay MSC that the uplink which is granted to a MS in the relay MSC area shall be released.

State Attributes

This information element is used to allow service logic running in an Anchor MSC to mute a VGCS talker even when the talker is served on a Relay MSC. The IE is used to build a GCC message that provides a mechanism to induce the VGCS talker terminal to mute/unmute the downlink at the Anchor MSC, as defined in 3GPP TS 24.068.

10.7 MAP_SEND_GROUP_CALL_END_SIGNAL service

10.7.1 Definitions

This service is used between the Relay MSC and the Anchor MSC indicating that VGCS / VBS channels have been established in the Relay MSC area. The response is used by the Anchor MSC to inform the Relay MSC that all resources for the call can be released in the Relay MSC because the call has been released in the Anchor MSC.

The MAP_SEND_GROUP_CALL_END_SIGNAL service is a confirmed service using the service primitives given in table 10.7/1.

10.7.2 Service primitives

Table 10.7/1: 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

10.7.3 Parameter definitions and use

IMSI

Identity of the service subscriber who has established the call and who is allowed to terminate the call.

Shall be present if the call was established by a service subscriber roaming in the relay MSC area.

Invoke Id

See definition in clause 7.6.1

Provider Error

See definition of provider error in clause 7.6.1.

10.8 MAP_Provide_SIWFS_Number

10.8.1 Definition

This service is used between an MSC and SIWFS. It is invoked by an MSC receiving an incoming call (call to or from MS) to request the SIWFS to allocate IWU resources. The service is defined in 3GPP TS 23.054 [103].

This is a confirmed service using the primitives described in table 10.8/1.

10.8.2 Service primitive

Table 10.8/1: 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

10.8.3 Parameter use

See clause 7.6 for a definition of the parameters used, in addition to the following.

GSM Bearer Capability

This information is the result from the negotiation with the mobile station. The information is sent from the MSC to the SIWFS to allocate the correct IWU.

ISDN Bearer Capability

This parameter refers to the ISDN Bearer Capability information element. For the MTC this parameter is received in the ISUP User Service Information parameter. For the MOC call this parameter is mapped from the GSM BC parameter according to 3GPP TS 29.007 [56]. The parameter is used by the SIWFS to route the call and to allocate the outgoing circuit.

Call Direction

This parameter indicates the direction of the call (mobile originated or mobile terminated) at call set-up.

B-subscriber address

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 IAM. If the loop method is used this parameter will indicate the address to the VMSC. This address is allocated by the VMSC in the same way as a MSRN and is used to correlate the incoming IAM to the corresponding MAP dialogue. If the non-loop method is used this parameter will indicate the address to the B-subscriber.

Chosen Channel

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

Lower Layer Compatibility

This parameter is sent from the MSC to the SIWF to allow the interworking unit to perform a compatibility check. This parameter is handled as specified in 3GPP TS 29.007 [56]. This parameter is defined in 3GPP TS 24.008.

High Layer Compatibility

This parameter is sent from the MSC to the SIWF to allow the interworking unit to perform a compatibility check. This parameter is handled as specified in 3GPP TS 29.007 [56]. This parameter is defined in 3GPP TS 24.008.

SIWFS number

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 IAM to the SIWFS (similar to MSRN) and will be used by the SIWFS to correlate the incoming IAM to the

corresponding MAP message. This parameter must always be sent from the SIWFS when a successful allocation of SIWFS resources has been made.

User error

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

- Resource limitation;
- Facility Not Supported;
- Unexpected Data Value;
- System Failure.

See clause 7.6 for a definition of these reasons.

Provider error

These are defined in clause 7.6.

10.9 MAP_SIWFS_Signalling_Modify

10.9.1 Definition

This service is used to transport signalling information between an MSC and an SIWFS in the case of a request to modify the configuration (e.g. HSCSD). It is invoked either by an MSC or by the SIWFS. The service is defined in 3GPP TS 23.054 [103].

This is a confirmed service using the primitives described in table 10.9/1.

10.9.2 Service primitive

Table 10.9/1: 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

10.9.3 Parameter use

See clause 7.6 for a definition of the parameter used, in addition to the following.

Channel Type

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

Chosen Channel

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

User error

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

- Resource limitation;

- Facility Not Supported;
- Data Missing;
- Unexpected Data Value;
- System Failure.

See clause 7.6 for a definition of these reasons.

Provider error

These are defined in clause 7.6.

10.10 MAP_SET_REPORTING_STATE service

10.10.1 Definition

This service is used between the HLR and the VLR to set the reporting state for a requested service. It is a confirmed service using the service primitives shown in table 10.10/1.

10.10.2 Service primitives

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

10.10.3 Parameter use

See clause 7.6 for a definition of the parameters used, in addition to the following.

IMSI

The IMSI is a mandatory parameter if the service is used as the only one in a dialogue.

CCBS Monitoring

This parameter indicates whether monitoring for CCBS shall be started or stopped. If it indicates that monitoring shall be started this service corresponds to the message 'Start Reporting' in 3GPP TS 23.093; if it indicates that monitoring shall be stopped this service corresponds to the message 'Stop Reporting' in 3GPP TS 23.093.

CCBS Subscriber Status

See 3GPP TS 23.093 for the use of this parameter and the conditions for its presence.

User error

This parameter is sent by the responder upon unsuccessful outcome of the service, and then takes one of the following values defined in clause 7.6.1:

- System Failure;
- Unidentified Subscriber;
- Unexpected Data Value;

- Data Missing;
- Resource Limitation;
- Facility Not Supported.

NOTE: This error is reserved for future use.

Provider error

These are defined in clause 7.6.

10.11 MAP_STATUS_REPORT service

10.11.1 Definition

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 primitives shown in table 10.11/1.

10.11.2 Service primitives

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

10.11.3 Parameter use

See clause 7.6 for a definition of the parameters used, in addition to the following.

CCBS Subscriber Status

If this parameter is present without Monitoring Mode and Call Outcome this service corresponds to the message 'Event Report' in 3GPP TS 23.093 [107]. See 3GPP TS 23.093 [107] for the use of this parameter and the conditions for its presence.

Monitoring Mode

If this parameter is present with CCBS Call Outcome this service corresponds to the message 'CCBS Call Report' in 3GPP TS 23.093 [107]. See 3GPP TS 23.093 [107] for the use of this parameter and the conditions for its presence.

Call Outcome

See 3GPP TS 23.093 [107] for the use of this parameter and the conditions for its presence.

User error

This parameter is sent by the responder upon unsuccessful outcome of the service, and then takes one of the following values defined in clause 7.6.1:

- Unknown Subscriber;
- System Failure;
- Unexpected Data Value;

- Data Missing.

Provider error

These are defined in clause 7.6.

10.12 MAP_REMOTE_USER_FREE service

10.12.1 Definition

This service is used between the HLR and the VLR to report that the B subscriber is now idle and that the A subscriber can be notified. It is a confirmed service using the service primitives shown in table 10.12/1.

10.12.2 Service primitives

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

10.12.3 Parameter use

See clause 7.6 for a definition of the parameters used, in addition to the following.

Call Info

See 3GPP TS 23.093 [107] for the use of this parameter.

CCBS Feature

See 3GPP TS 23.093 [107] for the conditions for the presence of the parameters included in the CCBS feature.

Translated B Number

See 3GPP TS 23.093 [107] for the use of this parameter.

Replace B Number

See 3GPP TS 23.093 [107] for the use of this parameter and the conditions for its presence.

Alerting Pattern

See 3GPP TS 23.093 [107] for the use of this parameter and the conditions for its presence.

RUF Outcome

See 3GPP TS 23.093 [107] for the use of this parameter and the conditions for its presence.

User error

This parameter is sent by the responder upon unsuccessful outcome of the service, and then takes one of the following values defined in clause 7.6.1:

- Unexpected Data Value;
- Data Missing;
- Incompatible Terminal;
- This error is returned by the responder when the terminal used for CCBS activation is not compatible with the terminal used for the CCBS recall. For details refer to 3GPP TS 24.008;
- Absent Subscriber (IMSI Detach; Restricted Area; No Page Response);
- System Failure;
- Busy Subscriber (CCBS Busy).

Provider error

These are defined in clause 7.6.

10.13 MAP_IST_ALERT service

10.13.1 Definition

This service is used between the MSC (Visited MSC or Gateway MSC) and the HLR, to report that the IST timer running for a call for the Subscriber has expired. It is a confirmed service using the service primitives shown in table 10.13/1.

10.13.2 Service primitives

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

10.13.3 Parameter use

All parameters are described in clause 7.6. The following clarifications are applicable:

IST Alert Timer

If included in the IST Alert response, it includes the new IST Alert timer value that must be used to inform the HLR about the call activities that the subscriber performs.

IST Information Withdraw

If included in the IST Alert response, this parameter is used to indicate that the IST condition has been removed for the subscriber. When the MSC receives this parameter, IST control for that call shall be terminated.

Call termination Indicator

If included in the IST Alert response, this parameter is used to indicate whether the MSC shall terminate the call activity that had previously triggered the IST Alert procedure, or it shall also release all other call activities for the specified subscriber (outgoing call activities if the IST Alert is initiated by the VMSC, or incoming call activities if the IST Alert is initiated by the GMSC). Release of all other call activities is possible only if the MSC has the capability to link the call activities for the Subscriber by using the IMSI as key.

User error

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

- System Failure;
- Unexpected Data Value;
- Resource Limitation;
- Facility Not Supported;
- Unknown Subscriber.

10.14 MAP_IST_COMMAND service

10.14.1 Definition

This service is used by the HLR to instruct the MSC (Visited MSC or Gateway MSC) to terminate ongoing call activities for a specific subscriber. It is a confirmed service using the service primitives shown in table 10.14/1.

10.14.2 Service primitives

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

10.14.3 Parameter use

All parameters are described in clause 7.6. The following clarifications are applicable:

User error

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

- System Failure;
- Unexpected Data Value;
- Resource Limitation;
- Facility Not Supported;
- Unknown Subscriber.

11 Supplementary services related services

11.1 MAP_REGISTER_SS service

11.1.1 Definition

This service is used between the MSC and the VLR and between the VLR and the HLR to register data related to a supplementary service. The VLR will relay the message to the HLR.

The service is a confirmed service and uses the service primitives shown in table 11.1./1.

11.1.2 Service primitives

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(=)
Long FTN Supported	C	C(=)		
NbrUser	C	C(=)	C	C(=)
Forwarding information			C	C(=)
User error			C	C(=)
Provider error				O

11.1.3 Parameter use

Invoke id

See clause 7.6.1 for the use of this parameter.

SS-Code

This parameter indicates the supplementary service which the mobile subscriber wants to register.

Basic service

This parameter indicates for which basic service group the supplementary service is to be registered. If it is not included, the registration request applies to all basic services.

Forwarded-to number with subaddress

This parameter is obligatory if the registration applies to one or more call forwarding supplementary services. It can optionally include a sub-address.

No reply condition time

This parameter is included if the registration applies to the Call Forwarding on No Reply supplementary service (or a superset of this service) and the mobile subscriber supplies a value for this time.

EMLPP default priority

This parameter is sent by the initiator to register the eMLPP default priority level and is returned by the responder at successful outcome of the service.

Long FTN Supported

This parameter indicates that the mobile station supports Long Forwarded-to Numbers.

NbrUser

This parameter is sent by the initiator to register the MC maximum number of user defined circuit switched bearers to be used.

Forwarding information

This parameter is returned by the responder at successful outcome of the service, if the registration request concerned one or a group of Call Forwarding supplementary services.

User error

This parameter is sent by the responder upon unsuccessful outcome of the service, and then takes one of the following values defined in clause 7.6.1:

- System failure;
- Data missing;
- Unexpected data value;
- Call Barred;
- Bearer service not provisioned;
- This error is returned only if not even a subset of the requested bearer service group has been subscribed to;
- Teleservice not provisioned;

  This error is returned only if not even a subset of the requested teleservice group has been subscribed to;
- Illegal SS operation;
- SS error status;
- SS incompatibility.

Provider error

See clause 7.6.1 for the use of this parameter.

11.2 MAP_ERASE_SS service

11.2.1 Definition

This service is used between the MSC and the VLR and between the VLR and the HLR to erase data related to a supplementary service. The VLR will relay the message to the HLR.

The service is a confirmed service and uses the service primitives shown in table 11.2/1.

11.2.2 Service primitives

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

11.2.3 Parameter use

Invoke id

See clause 7.6.1 for the use of this parameter.

SS-Code

This parameter indicates the supplementary service which the mobile subscriber wants to erase.

Basic service

This parameter indicates for which basic service group the supplementary service should be erased. If it is not included, the erasure request applies to all basic services.

Forwarding information

This parameter is returned by the responder at successful outcome of the service, if the erasure request concerned one or a group of Call Forwarding supplementary services.

User error

This parameter is sent by the responder upon unsuccessful outcome of the service, and then takes one of the following values, defined in clause 7.6.1:

- System failure;
- Data Missing;
- Unexpected data value;
- Bearer service not provisioned;

  This error is returned only if not even a subset of the requested bearer service group has been subscribed to;
- Teleservice not provisioned;

  This error is returned only if not even a subset of the requested teleservice group has been subscribed to;
- Call Barred;
- Illegal SS operation;
- SS error status.

Provider error

See clause 7.6.1 for the use of this parameter.

11.3 MAP_ACTIVATE_SS service

11.3.1 Definition

This service is used between the MSC and the VLR and between the VLR and the HLR to activate a supplementary service. The VLR will relay the message to the HLR.

The service is a confirmed service and uses the service primitives shown in table 11.3/1.

11.3.2 Service primitives

Table 11.3/1: MAP_ACTIVATE_SS parameters

Parameter name	Request	Indication	Response	Confirm
Invoke id	M	M(=)	M(=)	M(=)
SS-Code	M	M(=)		
Long FTN Supported	C	C(=)		
Basic service	C	C(=)		
Forwarding information			C	C(=)
Call barring information			C	C(=)
SS-Data			C	C(=)
User error			C	C(=)
Provider error				O

11.3.3 Parameter use

Invoke id

See clause 7.6.1 for the use of this parameter.

SS-Code

This parameter indicates the supplementary service which the mobile subscriber wants to activate.

Basic service

This parameter indicates for which basic service groups the requested supplementary service(s) should be activated. If it is not included, the activation request applies to all basic services.

Forwarding information

This parameter is returned by the responder at successful outcome of the service, if the activation request concerned Call Forwarding.

Long FTN Supported

This parameter indicates that the mobile station supports Long Forwarded-to Numbers.

Call barring information

This parameter is returned by the responder at successful outcome of the service, if the activation request concerned Call Barring.

SS-Data

This parameter is returned by the responder at successful outcome of the service, if the activation request concerned for example Call Waiting.

User error

This parameter is sent by the responder upon unsuccessful outcome of the service, and then takes one of the following values, defined in clause 7.6.1:

- System failure;
- Data Missing;
- Unexpected data value;
- Bearer service not provisioned;
- This error is returned only if not even a subset of the requested bearer service group has been subscribed to.
- Teleservice not provisioned;

- This error is returned only if not even a subset of the requested teleservice group has been subscribed to.
- Call Barred;
- Illegal SS operation;
- SS error status;
- SS subscription violation;
- SS incompatibility;
- Negative PW check;
- Number Of PW Attempts Violation.

Provider error

See clause 7.6.1 for the use of this parameter.

11.4 MAP_DEACTIVATE_SS service

11.4.1 Definitions

This service is used between the MSC and the VLR and between the VLR and the HLR to deactivate a supplementary service. The VLR will relay the message to the HLR.

The service is a confirmed service and uses the service primitives shown in table 11.4/1.

11.4.2 Service primitives

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

11.4.3 Parameter use

Invoke id

See clause 7.6.1 for the use of this parameter.

SS-Code

This parameter indicates the supplementary service which the mobile subscriber wants to deactivate.

Basic service

This parameter indicates for which basic service group the requested supplementary service(s) should be deactivated. If it is not included the deactivation request applies to all basic services.

Forwarding information

This parameter is returned by the responder at successful outcome of the service, if the deactivation request concerned one or a group of Call Forwarding supplementary services.

Call barring information

This parameter is returned by the responder at successful outcome of the service, if the activation request concerned one or a group of Call Barring supplementary services.

SS-Data

This parameter is returned by the responder at successful outcome of the service, for example if the deactivation request concerned the Call Waiting supplementary service.

User error

This parameter is sent by the responder upon unsuccessful outcome of the service, and then takes one of the following values, defined in clause 7.6.1:

- System failure;
- Data Missing;
- Unexpected data value;
- Bearer service not provisioned;

  This error is returned only if not even a subset of the requested bearer service group has been subscribed to;
- Teleservice not provisioned;

  This error is returned only if not even a subset of the requested teleservice group has been subscribed to;
- Call Barred;
- Illegal SS operation;
- SS error status;
- SS subscription violation;
- Negative PW check;
- Number Of PW Attempts Violation.

Provider error

See clause 7.6.1 for the use of this parameter.

11.5 MAP_INTERROGATE_SS service

11.5.1 Definitions

This service is used between the MSC and the VLR and between the VLR and the HLR to retrieve information related to a supplementary service. The VLR will relay the message to the HLR if necessary.

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

11.5.2 Service primitives

The service primitives are shown in table 11.5/1.

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(=)		
Long FTN Supported	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(=)
MC Information			C	C(=)
CCBS Feature LIST			C	C(=)
User error			C	C(=)
Provider error				O

11.5.3 Parameter use

For additional information on parameter use refer to the 3GPP TS 24.08x and 3GPP TS 24.09x-series of technical specifications.

Invoke id

See clause 7.6.1 for the use of this parameter.

SS-Code

The mobile subscriber can only interrogate a single supplementary service per service request.

Basic service

This parameter indicates for which basic service group the given supplementary service is interrogated. If it is not included, the interrogation request applies to all basic services.

SS-Status

This parameter is included by the responder if:

- the interrogated supplementary service can only be subscribed for all applicable basic services simultaneously; or
- the interrogated supplementary service is not active for any of the interrogated basic services, or
- the interrogation was for the CCBS supplementary service and no CCBS request is active or the service is not provisioned.

Basic service group LIST

This parameter LIST is used to include one or a series of basic service groups for which the interrogated supplementary service is active. If the interrogated supplementary service is not active for any of the interrogated (and provisioned) basic service groups, the SS-Status parameter is returned.

Long FTN Supported

This parameter indicates that the mobile station supports Long Forwarded-to Numbers.

Forwarding feature LIST

The forwarding feature parameter is described in clause 7.6.4. A list of one or more forwarding features is returned by the responder when the interrogation request applied to Call Forwarding supplementary service.

If no basic service code parameter is provided within this sequence, the forwarding feature parameter applies to all provisioned basic services.

CLI restriction Info

The CLI-RestrictionInfo parameter is returned by the responder when the interrogation request applies to the CLIR supplementary service.

EMLPP Info

The eMLPP info (maximum entitled priority and default priority) is returned by the responder if the interrogation request applies to the eMLPP supplementary service.

MC Information

The MC information (NbrSB, NbrUser and NbrSN) is returned by the responder if the interrogation request applies to the MC supplementary service. For a definition of these 3 components, refer to 3GPP TS 23.135 and 3GPP TS 24.135.

CCBS Feature LIST

The CCBS feature parameter is described in clause 7.6. A list of one or more CCBS features is returned by the responder when the interrogation request applied to the CCBS supplementary service. See 3GPP TS 23.093 [107] for the conditions for the presence of the parameters included in the CCBS feature.

User error

This error is sent by the responder upon unsuccessful outcome of the interrogation service, and then takes one of the following values, defined in clause 7.6.1:

- System failure;
- Data Missing;
- Unexpected data value;
- Bearer Service not provisioned;

  This error is returned only if not even a subset of the interrogated bearer services are provided;
- Teleservice not provisioned;

  This error is returned only if not even a subset of the interrogated teleservices are provided;
- Call Barred;
- Illegal SS operation;
- SS not available.

Provider error

See clause 7.6.1 for the use of this parameter.

11.6 MAP_INVOKE_SS service

11.6.1 Definitions

This service is used between the MSC and the VLR to check the subscriber's subscription to a given supplementary service in the VLR, in connection with in-call invocation of that supplementary service, i.e. after the call set-up phase is finished. For supplementary service invocation during call set-up phase, please refer to the call handling descriptions.

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

11.6.2 Service primitives

The service primitives are shown in table 11.6/1.

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

11.6.3 Parameter use

Invoke id

See clause 7.6.1 for the use of this parameter.

SS-Code

This SS-Code can only refer to a single supplementary service, e.g. the Call Hold or Multi Party supplementary services.

Basic service

This parameter indicates for which basic service the supplementary service invocation is required.

User error

This parameter is sent by the responder upon unsuccessful outcome of the service, and then takes one of the following values:

- System Failure;
- Data Missing;
- Unexpected data value;
- Call Barred;
- Illegal SS operation;
- SS error status;
- SS not available.

Provider error

See clause 7.6.1 for the use of this parameter.

11.7 MAP_REGISTER_PASSWORD service

11.7.1 Definitions

This service is used between the MSC and the VLR and between the VLR and the HLR if the mobile subscriber requests to register a new password. The VLR will relay the message to the HLR.

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

11.7.2 Service primitives

The service primitives are shown in table 11.7/1.

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

11.7.3 Parameter use

Invoke id

See clause 7.6.1 for the use of this parameter.

SS-Code

This parameter indicates for which supplementary service(s) the password should be registered.

New Password

See clause 7.6.4 for the use of this parameter.

User error

This parameter is sent by the responder upon unsuccessful outcome of the service, and then takes one of the following values, defined in clause 7.6.1:

- System failure;
- Data Missing;
- Unexpected data value;
- Call Barred;
- SS subscription violation;
- Password registration failure;
- Negative PW check;
- Number Of PW Attempts Violation.

Provider error

See clause 7.6.1 for the use of this parameter.

11.8 MAP_GET_PASSWORD service

11.8.1 Definitions

This service is used between the HLR and the VLR and between the VLR and the MSC when the HLR receives a request from the mobile subscriber for an operation on a supplementary service which requires a password from the subscriber. The VLR will relay the message to the MSC.

The service is a confirmed service and uses the service primitives shown in table 11.8/1.

11.8.2 Service primitives

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

11.8.3 Parameter use

Invoke id

See clause 7.6.1 for the use of this parameter.

Linked Id

See clause 7.6.1 for the use of this parameter. If the MAP_GET_PASSWORD service is used in conjunction with the MAP_REGISTER_PASSWORD service, this parameter must be present; otherwise it must be absent.

Guidance info

See clause 7.6.4 for the use of this parameter.

Current password

See clause 7.6.4 for the use of this parameter.

Provider error

See clause 7.6.1 for the use of this parameter.

11.9 MAP_PROCESS_UNSTRUCTURED_SS_REQUEST service

11.9.1 Definitions

This service is used between the MSC and the VLR, between the VLR and the HLR, between the HLR and gsmSCF and between the HLR and HLR to relay information in order to allow unstructured supplementary service operation.

The MAP_PROCESS_UNSTRUCTURED_SS_REQUEST service is a confirmed service using the primitives from table 11.9/1.

11.9.2 Service primitives

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

11.9.3 Parameter use

Invoke id

See clause 7.6.1 for the use of this parameter.

USSD Data Coding Scheme

See clause 7.6.4 for the use of this parameter. The presence of the parameter in the response is dependent on the unstructured supplementary service application. If this parameter is present, then the USSD String parameter has to be present.

USSD String

See clause 7.6.1 for the use of this parameter. The presence of the parameter in the response is dependent on the unstructured supplementary service application. If this parameter is present, then the USSD Data Coding Scheme parameter has to be present.

MSISDN

The subscriber's basic MSISDN.

See definition in clause 7.6.2. The MSISDN is included as an operator option, e.g. to allow addressing the subscriber's data in the gsmSCF with the MSISDN.

User error

This parameter is sent by the responder upon unsuccessful outcome of the service, and then takes one of the following values defined in clause 7.6.1:

- System failure;
- Data missing;
- Unexpected data value;

  This error is returned by the responder if it is not able to deal with the contents of the USSD string.
- Call Barred;
- Unknown Alphabet.

Provider error

See clause 7.6.1 for the use of this parameter.

11.10 MAP_UNSTRUCTURED_SS_REQUEST service

11.10.1 Definitions

This service is used between the gsmSCF and the HLR, the HLR and the VLR and between the VLR and the MSC when the invoking entity requires information from the mobile user, in connection with unstructured supplementary service handling.

The MAP_UNSTRUCTURED_SS_REQUEST service is a confirmed service using the primitives from table 11.10/1.

11.10.2 Service primitives

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

11.10.3 Parameter use

Invoke id

See clause 7.6.1 for the use of this parameter.

USSD Data Coding Scheme

See clause 7.6.4 for the use of this parameter. The presence of the parameter in the response is dependent on the mobile user's MMI input. If this parameter is present, then the USSD String parameter has to be present.

USSD String

See clause 7.6.1 for the use of this parameter. The presence of the parameter in the response is dependent on the mobile user's MMI input. If this parameter is present, then the USSD Data Coding Scheme parameter has to be present.

Alerting Pattern

See clause 7.6.3 for the use of this parameter.

User error

This parameter is sent by the responder upon unsuccessful outcome of the service, and then takes one of the following values defined in clause 7.6.1:

- System failure;
- Data missing;
- Unexpected data value;

  This error is returned by the responder if it is not able to deal with the contents of the USSD string;

- Absent Subscriber;
- Illegal Subscriber;

  This error indicates that delivery of the unstructured supplementary service data failed because the MS failed authentication;

- Illegal Equipment;
- USSD Busy;
- Unknown Alphabet.

Provider error

See clause 7.6.1 for the use of this parameter.

11.11 MAP_UNSTRUCTURED_SS_NOTIFY service

11.11.1 Definitions

This service is used between the gsmSCF and the HLR, the HLR and the VLR and between the VLR and the MSC when the invoking entity requires a notification to be sent to the mobile user, in connection with unstructured supplementary services handling.

The MAP_UNSTRUCTURED_SS_NOTIFY service is a confirmed service using the primitives from table 11.11/1.

11.11.2 Service primitives

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

11.11.3 Parameter use

Invoke id

See clause 7.6.1 for the use of this parameter.

USSD Data Coding Scheme:

See clause 7.6.4 for the use of this parameter.

USSD String:

See clause 7.6.1 for the use of this parameter.

Alerting Pattern

See clause 7.6.3 for the use of this parameter.

User error

This parameter is sent by the responder upon unsuccessful outcome of the service, and then takes one of the following values defined in clause 7.6.1:

- System failure;
- Data missing;
- Unexpected data value;

  This error is returned by the responder if it is not able to deal with the contents of the USSD string.

- Absent Subscriber;
- Illegal Subscriber;

  This error indicates that delivery of the unstructured supplementary service data failed because the MS failed authentication.

- Illegal Equipment;
- USSD Busy;

- Unknown Alphabet.

Provider error

See clause 7.6.1 for the use of this parameter.

11.12 MAP_SS_INVOCATION_NOTIFY

11.12.1 Definition

This service is used between the MSC and the gsmSCF when the subscriber invokes one of the following supplementary services; Call Deflection (CD), Explicit Call Transfer (ECT) or Multi Party (MPTY).

This service is used between the HLR and the gsmSCF when the subscriber invokes the CCBS supplementary service.

11.12.2 Service primitives

The service primitives are shown in table 11.12/1.

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(=)		
B-subscriber Number	C	C(=)		
CCBS Request State	C	C(=)		
User error			C	C(=)
Provider error				O

11.12.3 Parameter use

All parameters are described in clause 7.6. The use of these parameters and the requirements for their presence are specified in 3GPP TS 23.078.

User error

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

- Data Missing;
- Unexpected Data Value;
- Unknown Subscriber.

Provider error

This is defined in clause 7.6.1.

11.13 MAP_REGISTER_CC_ENTRY service

11.13.1 Definition

This service is used between the MSC and the VLR and between the VLR and the HLR to register data for a requested call completion supplementary service. The VLR will relay the message to the HLR.

The service is a confirmed service and uses the service primitives shown in table 11.13/1.

11.13.2 Service primitives

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

11.13.3 Parameter use

See clause 7.6 for a definition of the parameters used, in addition to the following.

SS-Code

This parameter indicates the call completion supplementary service for which the mobile subscriber wants to register an entry.

CCBS Feature

See 3GPP TS 23.093 [107] for the conditions for the presence of the parameters included in the CCBS feature.

Translated B Number

See 3GPP TS 23.093 [107] for the use of this parameter and the conditions for its presence.

Service Indicator

This parameter corresponds to the parameters 'Presentation Indicator' and 'CAMEL Invoked' in 3GPP TS 23.093 [107]. It indicates which services have been invoked for the original call (e.g. CLIR, CAMEL). See 3GPP TS 23.093 [107] for the use of this parameter and the conditions for its presence.

Call Info

See 3GPP TS 23.093 [107] for the use of this parameter and the conditions for its presence.

Network Signal Info

See 3GPP TS 23.093 [107] for the use of this parameter and the conditions for its presence.

User error

This parameter is sent by the responder upon unsuccessful outcome of the service, and then takes one of the following values, defined in clause 7.6.1:

- System failure;
- Data missing;
- Unexpected data value;
- Call Barred;
- Illegal SS operation;
- SS error status;
- SS incompatibility.

- Short Term Denial;
- Long Term Denial;
- Facility Not Supported;

NOTE: This error is reserved for future use.

Private Extensions shall not be sent with these user errors for this operation.

Provider error

See clause 7.6.1 for the use of this parameter.

11.14 MAP_ERASE_CC_ENTRY service

11.14.1 Definition

This service is used between the MSC and the VLR and between the VLR and the HLR to erase data related to a call completion supplementary service. The VLR will relay the message to the HLR.

The service is a confirmed service and uses the service primitives shown in table 11.14/1.

11.14.2 Service primitives

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

11.14.3 Parameter use

See clause 7.6 for a definition of the parameters used, in addition to the following.

SS-Code

This parameter indicates the call completion supplementary service for which the mobile subscriber wants to erase an entry/entries.

CCBS Index

See 3GPP TS 23.093 [107] for the use of this parameter and the condition for its presence.

SS-Status

Depending on the outcome of the service request this parameter may indicate either provisioned and active or not provisioned.

User error

This parameter is sent by the responder upon unsuccessful outcome of the service, and then takes one of the following values, defined in clause 7.6.1:

- System failure;
- Data Missing;

- Unexpected data value;
- Call Barred;
- Illegal SS operation;
- SS error status.

Private Extensions shall not be sent with these user errors for this operation.

Provider error

See clause 7.6.1 for the use of this parameter.

12 Short message service management services

12.1 MAP-SEND-ROUTING-INFO-FOR-SM service

12.1.1 Definition

This service is used between the gateway MSC and the HLR to retrieve the routing information needed for routing the short message to the servicing MSC.

The MAP-SEND-ROUTING-INFO-FOR-SM is a confirmed service using the primitives from table 12.1/1.

12.1.2 Service primitives

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

12.1.3 Parameter use

Invoke id

See definition in clause 7.6.1.

MSISDN

See definition in clause 7.6.2.

SM-RP-PRI

See definition in clause 7.6.8.

Service Centre Address

See definition in clause 7.6.2.

SM-RP-MTI

See definition in clause 7.6.8. This parameter shall be present when the feature « SM filtering by the HPLMN » is supported by the SMS-GMSC and when the equivalent parameter is received from the short message service relay sub-layer protocol.

SM-RP-SMEA

See definition in clause 7.6.8. This parameter shall be present when the feature « SM filtering by the HPLMN » is supported by the SMS-GMSC and when the equivalent parameter is received from the short message service relay sub-layer protocol.

GPRS Support Indicator

See definition in clause 7.6.8. The presence of this parameter is mandatory if the SMS-GMSC supports receiving of the two numbers from the HLR.

IMSI

See definition in clause 7.6.2. The presence of this parameter is mandatory in a successful case.

Network Node Number

See definition in clause 7.6.2. This parameter is provided in a successful response.

LMSI

See definition in clause 7.6.2. It is an operator option to provide this parameter from the VLR; it is mandatory for the HLR to include the LMSI in a successful response, if the VLR has used the LMSI.

GPRS Node Indicator

See definition in clause 7.6.8. The presence of this parameter is mandatory if only the SGSN number is sent in the Network Node Number.

Additional Number

See definition in clause 7.6.2. This parameter is provided in a successful response.

User error

The following errors defined in clause 7.6.1 may be used, depending on the nature of the fault:

- Unknown subscriber;
- Call Barred;
- Teleservice Not Provisioned;
- Absent Subscriber_SM;
- Facility Not Supported;
- System failure;
- Unexpected Data Value;
- Data missing.

Provider error

For definition of provider errors see clause 7.6.1.

12.2 MAP-MO-FORWARD-SHORT-MESSAGE service

12.2.1 Definition

This service is used between the serving MSC or the SGSN and the SMS InterworkingMSC to forward mobile originated short messages.

The MAP-MO-FORWARD-SHORT-MESSAGE service is a confirmed service using the service primitives given in table 12.2/1.

12.2.2 Service primitives

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 User error	C	C(=)	 C	 C(=)
Provider error				O

12.2.3 Parameter use

Invoke id

See definition in clause 7.6.1.

SM RP DA

See definition in clause 7.6.8.

In the mobile originated SM transfer this parameter contains the Service Centre address received from the mobile station.

SM RP OA

See definition in clause 7.6.8.

The MSISDN received from the VLR or from the SGSN is inserted in this parameter in the mobile originated SM transfer.

SM RP UI

See definition in clause 7.6.8. The short message transfer protocol data unit received from the Service Centre is inserted in this parameter.

IMSI

See definition in clause 7.6.2.1. The IMSI of the originating subscriber is inserted in this parameter in the mobile originated SM transfer.

This parameter shall be included if the sending entity, whether MSC or SGSN, supports mobile number portability.

User error

The following errors defined in clause 7.6.1 may be used, depending on the nature of the fault:

- Facility Not Supported;
- System Failure;

- SM Delivery Failure;
  - The reason of the SM Delivery Failure can be one of the following in the mobile originated SM:
    - unknown Service Centre address;
    - Service Centre congestion;
    - invalid Short Message Entity address;
    - subscriber not Service Centre subscriber;
    - protocol error.
- Unexpected Data Value

Provider error

For definition of provider errors see clause 7.6.1.

12.3 MAP-REPORT-SM-DELIVERY-STATUS service

12.3.1 Definition

This service is used between the gateway MSC and the HLR. The MAP-REPORT-SM-DELIVERY-STATUS service is used to set the Message Waiting Data into the HLR or to inform the HLR of successful SM transfer after polling. This service is invoked by the gateway MSC.

The MAP-REPORT-SM-DELIVERY-STATUS service is a confirmed service using the service primitives given in table 12.3/1.

12.3.2 Service primitives

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 Diagnostic SM	C	C(=)		
GPRS Support Indicator	C	C(=)		
Delivery Outcome Indicator	C	C(=)		
Additional SM Delivery Outcome	C	C(=)		
Additional Absent Subscriber Diagnostic SM	C	C(=)		
MSIsdn-Alert			C	C(=)
User error			C	C(=)
Provider error				O

12.3.3 Parameter use

Invoke id

See definition in clause 7.6.1.

MSISDN

See definition in clause 7.6.2.

Service Centre Address

See definition in clause 7.6.2.

SM Delivery Outcome

See definition in clause 7.6.8. This parameter indicates the status of the mobile terminated SM delivery.

Absent Subscriber Diagnostic SM

See definition in clause 7.6.8.

GPRS Support Indicator

See definition in clause 7.6.8. The presence of this parameter is mandatory if the SMS-GMSC supports handling of two delivery outcomes.

Delivery Outcome Indicator

See definition in clause 7.6.8.

Additional SM Delivery Outcome

See definition in clause 7.6.8.

Additional Absent Subscriber Diagnostic SM

See definition in clause 7.6.8.

MSIsdn-Alert

See definition in clause 7.6.2. This parameter shall be present in case of unsuccessful delivery, when the MSISDN received in the operation is different from the stored MSIsdn-Alert; the stored MSIsdn-Alert is the value that is returned to the gateway MSC.

User error

The following errors defined in clause 7.6.1 may be used, depending on the nature of the fault:

- Unknown Subscriber;
- Message Waiting List Full;
- Unexpected Data Value;
- Data missing.

Provider error

For definition of provider errors see clause 7.6.1.

12.4 MAP-READY-FOR-SM service

12.4.1 Definition

This service is used between the MSC and VLR as well as between the VLR and the HLR. The MSC initiates this service if a subscriber indicates memory available situation. The VLR uses the service to indicate this to the HLR.

The VLR initiates this service if a subscriber, whose message waiting flag is active in the VLR, has radio contact in the MSC.

Also this service is used between the SGSN and the HLR. The SGSN initiates this service if a subscriber indicates memory available situation. The SGSN uses the service to indicate this to the HLR.

The SGSN initiates this service if a subscriber, whose message waiting flag is active in the SGSN, has radio contact in the GPRS.

The MAP-READY-FOR-SM service is a confirmed service using the primitives from table 12.4/1.

12.4.2 Service primitives

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

12.4.3 Parameter use

Invoke id

See definition in clause 7.6.1.

IMSI

See definition in clause 7.6.2. The IMSI is used always between the VLR and the HLR and between the SGSN and the HLR. Between the MSC and the VLR the identification can be either IMSI or TMSI.

TMSI

See definition in clause 7.6.2. The identification can be either IMSI or TMSI between MSC and VLR.

Alert Reason

See definition in clause 7.6.8. This parameter indicates if the mobile subscriber is present or the MS has memory available.

Alert Reason Indicator

See definition in clause 7.6.8.

User error

The following errors defined in clause 7.6.1 may be used, depending on the nature of the fault:

- Unknown Subscriber;
- Facility Not Supported;
- System Failure;
- Unexpected Data Value;
- Data missing.

Provider error

For definition of provider errors see clause 7.6.1.

12.5 MAP-ALERT-SERVICE-CENTRE service

12.5.1 Definition

This service is used between the HLR and the interworking MSC. The HLR initiates this service, if the HLR detects that a subscriber, whose MSISDN is in the Message Waiting Data file, is active or the MS has memory available.

The MAP-ALERT-SERVICE-CENTRE service is a confirmed service using the primitives from table 12.5/1.

12.5.2 Service primitives

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

12.5.3 Parameter use

Invoke id

See definition in clause 7.6.1.

MSIsdn-Alert

See definition in clause 7.6.2. The provided MSISDN shall be the one which is stored in the Message Waiting Data file.

Service Centre Address

See definition in clause 7.6.2.

User error

The following errors defined in clause 7.6.1 may be used, depending on the nature of the fault:

- System Failure;
- Unexpected Data Value;
- Data missing.

Provider error

For definition of provider errors see clause 7.6.1.

12.6 MAP-INFORM-SERVICE-CENTRE service

12.6.1 Definition

This service is used between the HLR and the gateway MSC to inform the Service Centre which MSISDN number is stored in the Message Waiting Data file. If the stored MSISDN number is not the same as the one received from the gateway MSC in the MAP-SEND-ROUTING-INFO-FOR-SM service primitive the stored MSISDN number is included in the message.

Additionally the status of MCEF, MNRF and MNRG flags and the inclusion of the particular Service Centre address in the Message Waiting Data list is informed to the gateway MSC when appropriate.

The MAP-INFORM-SERVICE-CENTRE service is a non-confirmed service using the primitives from table 12.6/1.

12.6.2 Service primitives

Table 12.6/1: MAP-INFORM-SERVICE-CENTRE

Parameter name	Request	Indication
Invoke Id	M	M(=)
MSIsdn-Alert	C	C(=)
MWD Status	C	C(=)

12.6.3 Parameter use

Invoke id

See definition in clause 7.6.1.

MSIsdn-Alert

See definition in clause 7.6.2 This parameter refers to the MSISDN stored in a Message Waiting Data file in the HLR.

MWD Status

See definition in clause 7.6.8. This parameter indicates the status of the MCEF, MNRF and MNRG flags and the status of the particular SC address presence in the Message Waiting Data list.

12.7 MAP-SEND-INFO-FOR-MT-SMS service

12.7.1 Definition

This service is used between the MSC and the VLR. The service is invoked by the MSC receiving an mobile terminated short message to request subscriber related information from the VLR.

The MAP-SEND-INFO-FOR-MT-SMS service is a confirmed service using the primitives from table 12.7/1.

12.7.2 Service primitives

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

12.7.3 Parameter use

Invoke id

See definition in clause 7.6.1.

SM RP DA

See definition in clause 7.6.8. This parameter shall contain either an IMSI or a LMSI.

MSISDN

See definition in clause 7.6.2.

User error

The following errors defined in clause 7.6.1 may be used, depending on the nature of the fault:

- Unknown subscriber;
- Unidentified Subscriber;
- Absent subscriber;
- Unexpected Data Value;
- Data Missing;
- Illegal subscriber;
- Illegal equipment;
- Subscriber busy for MT SMS;
- System Failure.

Provider error

For definition of provider errors see clause 7.6.1.

12.8 MAP-SEND-INFO-FOR-MO-SMS service

12.8.1 Definition

This service is used between the MSC and the VLR. The service is invoked by the MSC which has to handle a mobile originated short message request to request the subscriber related information from the VLR.

The MAP-SEND-INFO-FOR-MO-SMS service is a confirmed service using the primitives from table 12.8/1.

12.8.2 Service primitives

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

12.8.3 Parameter use

Invoke id

See definition in clause 7.6.1.

Service Centre Address

See definition in clause 7.6.2.

MSISDN

See definition in clause 7.6.2.

User error

The following errors defined in clause 7.6.1 may be used, depending on the nature of the fault:

- Teleservice Not Provisioned;
- Call Barred;
- Unexpected Data Value;
- Data Missing.

Provider error

For definition of provider errors see clause 7.6.1.

12.9 MAP-MT-FORWARD-SHORT-MESSAGE service

12.9.1 Definition

This service is used between the gateway MSC and the servicing MSC or the SGSN to forward mobile terminated short messages.

The MAP-MT-FORWARD-SHORT-MESSAGE service is a confirmed service using the service primitives given in table 12.9/1.

12.9.2 Service primitives

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

12.9.3 Parameter use

Invoke id

See definition in clause 7.6.1.

SM RP DA

See definition in clause 7.6.8. This parameter can contain either an IMSI or a LMSI. The use of the LMSI is an operator option. The LMSI can be provided if it is received from the HLR. The IMSI is used if the use of the LMSI is not available.

This parameter is omitted in the mobile terminated subsequent SM transfers.

SM RP OA

See definition in clause 7.6.8. The Service Centre address received from the originating Service Centre is inserted in this parameter.

This parameter is omitted in the mobile terminated subsequent SM transfers.

SM RP UI

See definition in clause 7.6.8. The short message transfer protocol data unit received from the Service Centre is inserted in this parameter. A short message transfer protocol data unit may also be inserted in this parameter in the message delivery acknowledgement from the MSC or from the SGSN to the Service Centre.

More Messages To Send

See definition in clause 7.6.8. The information from the MMS indication received from the Service Centre is inserted in this parameter.

User error

The following errors defined in clause 7.6.1 may be used, depending on the nature of the fault:

- Unidentified subscriber;
- Absent Subscriber_SM;
- Subscriber busy for MT SMS;
- Facility Not Supported;
- Illegal Subscriber indicates that delivery of the mobile terminated short message failed because the mobile station failed authentication;
- Illegal equipment indicates that delivery of the mobile terminated short message failed because an IMEI check failed, i.e. the IMEI was blacklisted or not white-listed;
- System Failure;
- SM Delivery Failure:
    - The reason of the SM Delivery Failure can be one of the following in the mobile terminated SM:
        - memory capacity exceeded in the mobile equipment;
        - protocol error;
        - mobile equipment does not support the mobile terminated short message service.
- Unexpected Data Value;
- Data Missing.

Provider error

For definition of provider errors see clause 7.6.1.

13 Network-Requested PDP Context Activation services

13.1 MAP_SEND_ROUTING_INFO_FOR_GPRS service

13.1.1 Definition

This service is used by the GGSN to request GPRS routing information from the HLR.

13.1.2 Service primitives

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

13.1.3 Parameter definition and use

Invoke Id

See definition in clause 7.6.1.

IMSI

See definition in clause 7.6.2.

GGSN address

This parameter shall be present if the protocol-converting GSN is used between the GGSN and the HLR.

GGSN number

See definition in clause 7.6.2.

SGSN address

This parameter shall be present if the outcome of the Send Routing Info For GPRS request to the GPRS application process in the HLR is positive.

Mobile Not Reachable Reason

This parameter shall be present if the outcome of the Send Routing Info For GPRS request to the GPRS application process in the HLR is positive and the MNRG flag in the HLR is set. See definition in clause 7.6.3.51.

User error

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

- Absent Subscriber;
- System Failure;
- Data Missing;
- Unexpected Data Value;

- UnknownSubscriber.

The diagnostic in the Unknown Subscriber may indicate “Imsi Unknown” or “Gprs Subscription Unknown”.

Provider error

These are defined in clause 7.6.1.

13.2 MAP_FAILURE_REPORT service

13.2.1 Definition

This service is used by the GGSN to inform the HLR that network requested PDP-context activation has failed.

13.2.2 Service primitives

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

13.2.3 Parameter definition and use

Invoke Id

See definition in clause 7.6.1.

IMSI

See definition in clause 7.6.2.

GGSN address

This parameter shall be present if the protocol-converting GSN is used between the GGSN and the HLR.

GGSN number

See definition in clause 7.6.2.

User error

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

- System Failure;
- Data Missing;
- Unexpected Data Value;
- UnknownSubscriber.

Provider error

These are defined in clause 7.6.1.

13.3 MAP_NOTE_MS_PRESENT_FOR_GPRS service

13.3.1 Definition

This service is used by the HLR to inform the GGSN that the MS is present for GPRS again.

13.3.2 Service primitives

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

13.3.3 Parameter definition and use

Invoke Id

See definition in clause 7.6.1.

IMSI

See definition in clause 7.6.2.

GGSN address

This parameter shall be present if the protocol-converting GSN is used between the GGSN and the HLR.

SGSN address

See definition in clause 7.6.2.

User error

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

- System Failure;
- Data Missing;
- Unexpected Data Value;
- UnknownSubscriber.

Provider error

These are defined in clause 7.6.1.

13A Location Service Management Services

13A.1 MAP-SEND-ROUTING-INFO-FOR-LCS Service

13A.1.1 Definition

This service is used between the GMLC and the HLR to retrieve the routing information needed for routing a location service request to the servicing VMSC. The MAP-SEND-ROUTING-INFO-FOR-LCS is a confirmed service using the primitives from table 13A.1/1.

13A.1.2 Service Primitives

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

13A.1.3 Parameter Use

Invoke id

See definition in clause 7.6.1.

MLC Number

See definition in clause 7.6.2.

MSISDN

See definition in clause 7.6.2. The request shall carry either the IMSI or MSISDN. The response shall carry whichever of these was not included in the request (see 3GPP TS 23.171 for details).

IMSI

See definition in clause 7.6.2.

LMSI

See definition in clause 7.6.2. It is an operator option to provide this parameter from the VLR; it is mandatory for the HLR to include the LMSI in a successful response, if the VLR has used the LMSI.

MSC Number

See definition in clause 7.6.2. This parameter is provided in a successful response.

User error

The following errors defined in clause 7.6.1 may be used, depending on the nature of the fault:

- Unknown subscriber;
- Absent Subscriber;
- Facility Not Supported;

- System failure;
- Unexpected Data Value;
- Data missing;
- Unauthorised requesting network.

Provider error

For definition of provider errors see clause 7.6.1.

13A.2 MAP-PROVIDE-SUBSCRIBER-LOCATION Service

13A.2.1 Definition

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 confirmed service using the primitives from table 13A.2/1.

13A.2.2 Service Primitives

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(=)		
Supported GAD Shapes	C	C(=)		
Location Estimate			M	M(=)
Age of Location Estimate			C	C(=)
Additional Location Estimate			C	C(=)
User error			C	C(=)
Provider error				O

13A.2.3 Parameter Definition and Use

All parameters are defined in clause 7.6. The use of these parameters and the requirements for their presence are specified in 3GPP TS 23.171.

Location Type

This parameter identifies the type of location information requested.

MLC Number

This is the E.164 number of the requesting GMLC.

LCS Client ID

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

Privacy Override

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

IMSI

The IMSI is provided to identify the target MS. At least one of the IMSI or MSISDN is mandatory.

MSISDN

The MSISDN is provided to identify the target MS. At least one of the IMSI or MSISDN is mandatory.

LMSI

The LMSI shall be provided if previously supplied by the HLR.

LCS Priority

This parameter indicates the priority of the location request.

LCS QoS

This parameter indicates the required quality of service in terms of response time and accuracy.

IMEI

Inclusion of the IMEI is optional.

Supported GAD Shapes

This parameter indicates which of the shapes defined in 3GPP TS 23.032 are supported.

Location Estimate

This parameter provides the location estimate if this is encoded in one of the supported geographical shapes. Otherwise this parameter shall consist of one octet, which shall be discarded by the receiving node.

Age of Location Estimate

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

Additional Location Estimate

This parameter provides the location estimate when not provided by the Location Estimate parameter. It may be sent only if the parameter Supported GAD Shapes has been received in the Provide Subscriber Location indication and the shape to be included is supported by the GMLC.

User error

This parameter is sent by the responder when the location request has failed or cannot proceed and if present, takes one of the following values defined in clause 7.6.1.

- System Failure;
- Data Missing;
- Unexpected Data Value;
- Facility Not Supported;
- Unidentified Subscriber;
- Illegal Subscriber;
- Illegal Equipment;
- Absent Subscriber (diagnostic information may also be provided);
- Unauthorised requesting network;

- Unauthorised LCS Client with detailed reason;
- Position method failure with detailed reason.

Provider error

These are defined in clause 7.6.1.

13A.3 MAP-SUBSCRIBER-LOCATION-REPORT Service

13A.3.1 Definition

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 implicitly administered or made at some earlier time. This is a confirmed service using the primitives from table 13A.3/1.

13A.3.2 Service Primitives

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(=)	C	C(=)
IMEI	U	C(=)		
Location Estimate	C	C(=)		
Age of Location Estimate	C	C(=)		
LMSI	U	C(=)		
Additional Location Estimate	C	C(=)		
NA-ESRK Request	C	C(=)		
User error			C	C(=)
Provider error				O

13A.3.3 Parameter Definition and Use

All parameters are defined in clause 7.6. The use of these parameters and the requirements for their presence are specified in 3GPP TS 23.171.

LCS Event

This parameter indicates the event that triggered the Subscriber Location Report.

LCS Client ID

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

MSC Number

See definition in clause 7.6.2. This parameter provides the address of the visited MSC for target MS.

IMSI

The IMSI shall be provided if available to the VMSC.

MSISDN

The MSISDN shall be provided if available to the VMSC.

NA-ESRD

If the target MS has originated an emergency service call in North America, the NA-ESRD shall be provided by the VMSC if available.

NA-ESRK

If the target MS has originated an emergency service call in North America, the NA-ESRK shall be provided by the VMSC if assigned.

If the target MS has originated an emergency service call in North America and NA-ESRK Request is included in Subscriber_Location_Report-Arg, NA-ESRK may also be included in the response to the MSC, see 3GPP TS 23.271 [26a].

IMEI

Inclusion of the IMEI is optional.

Location Estimate

This parameter provides the location estimate. The absence of this parameter implies that a location estimate was not available or could not be successfully obtained. If the obtained location estimate is not encoded in one of the supported geographical shapes then this parameter shall consist of one octet, which shall be discarded by the receiving node.

Age of Location Estimate

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

LMSI

The LMSI may be provided if assigned by the VLR.

Additional Location Estimate

This parameter provides the location estimate when not provided by the Location Estimate parameter.

NA-ESRK Request

If the target MS has originated an emergency service call in North America, NA-ESRK Request may be included to indicate that the MSC is able to accept NA-ESRK in the Response message, see section 7.6.11.19.

User error

This parameter is sent by the responder when the received message contains an error, cannot be forwarded or stored for an LCS client or cannot be accepted for some other reason and if present, takes one of the following values defined in clause 7.6.1.

- System Failure;
- Data Missing;
- Unexpected Data Value;
- Resource Limitation;
- Unknown Subscriber;
- Unauthorised requesting network;
- Unknown or unreachable LCS Client.

Provider error

These are defined in clause 7.6.1.

14 General

14.1 Overview

Clauses 14 to 17 specify the protocol elements to be used to provide the MAP services described in clause 7.

Clause 15 specifies the elements of procedures for the MAP protocol. Clause 16 specifies the mapping onto TC service primitives. Clause 17 specifies the application contexts, operation packages and abstract syntaxes for the MAP protocol as well as the encoding rules to be applied.

14.2 Underlying services

The MAP protocol relies on the services provided by the Transaction Capabilities (TC) of Signalling System Number No. 7, as referenced in clause 6.

14.3 Model

The MAP Protocol Machine (MAP PM) can be modelled as a collection of service state machines (SSMs) - one per MAP specific service invoked - coordinated by a MAP dialogue control function with its one state machine: MAP dialogue state machine (DSM). There are two types of Service State Machines: Requesting Service State Machines (RSM) and Performing Service State Machines (PSM).

A new invocation of a MAP PM is employed on the receipt of a MAP-OPEN request primitive or a TC-BEGIN indication primitive. Each invocation controls exactly one MAP dialogue. For each MAP specific service invoked during a dialogue, a MAP RSM is created at the requestor's side and a MAP PSM is created at the performer's side.

This modelling is used only to facilitate understanding and the MAP behaviour descriptions and is not intended to suggest any implementation. SDL descriptions are organised according to this model.

How the MAP-service-user and the MAP refer to a MAP dialogue (i.e. a MAP PM invocation) is a local implementation matter.

How TC dialogue identifiers are assigned to a MAP PM invocation is also a local implementation matter.

14.4 Conventions

The behaviour of the MAP PM depends on the application-context-name associated with the dialogue. One major difference is that the MAP requests the transfer of the application-context-name by TC only for those contexts which do not belong to the so-called "version one context set".

The "version one context set" is a set of application-contexts which model the behaviour of a MAP V1 implementation according to the latest phase 1 version of GSM 09.02. This set is defined in clause 15.

The procedures described in clause 15 are used when the application-context-name does not refer to a dialogue between an MSC and its VLR. When the application-context-name refers to a dialogue between an MSC and its VLR the MAP PM procedures are a local implementation matter.

15 Elements of procedure

15.1 Dialogue establishment

The establishment of a MAP dialogue involves two MAP-service-users, one that is the dialogue-initiator and one that is the dialogue-responder.

This procedure is driven by the following signals:

- a MAP-OPEN request primitive from the dialogue-initiator;
- a TC-BEGIN indication primitive occurring at the responding side;
- a MAP-OPEN response primitive from the dialogue-responder;
- the first TC-CONTINUE indication primitive occurring at the initiating side;

and under specific conditions:

- a TC-END indication primitive occurring at the initiating side;
- a TC-U-ABORT indication primitive occurring at the initiating side;
- a TC-P-ABORT indication primitive occurring at the initiating side.

15.1.1 Handling of unknown operations

Unknown operations (i.e. a standard operation introduced in a later version of 09.02 or a private operation) can be introduced in MAP in a backwards compatible way. This means, that the receiver of an unknown operation shall, if the dialogue state allows it, send a TC-REJECT component to the sender of the operation indicating 'unrecognised operation' and continue with the processing of further components or messages exchanged within the dialogue as if the unknown operation had not been received.

The standardised structure of a MAP dialogue shall not be affected by the invocation of unknown operations, i.e. if a dialogue uses only a TC-BEGIN message which is acknowledged by a TC-END message, a TC-CONTINUE message shall not be used to invoke an unknown operation. However the standardised structure of a MAP dialogue may be affected by the rejection of unknown operations, i.e. if a dialogue uses only a TC-BEGIN message which is acknowledged by a TC-END message, a TC-CONTINUE message followed by a TC-END message may be used to carry the rejection of an unknown operation and the response to the standardised operation. The entity which initiated a dialogue whose standardised structure is a TC-BEGIN message which is acknowledged by a TC-END message shall not send any messages in that dialogue after the TC-BEGIN.

Note that if the dialogue structure is affected as described in this paragraph the TC-CONTINUE shall include the dialogue portion required to confirm the acceptance of the dialogue.

Unknown operations can be invoked in the following types of messages (there is no restriction as to how many unknown operations can be invoked in a message):

- TC-BEGIN the component to invoke the unknown operation shall follow the component of the standard operation that is included in this message.
- TC-CONTINUE: the component to invoke the unknown operation may be transported as the only component in a stand-alone message or can be grouped with existing operations. In the latter case a specific sequencing of components is not required.
- TC-END: if the component to invoke the unknown operation is grouped with an existing operation a specific sequencing of components is not required.

The TC-REJECT component may be sent in the following messages:

- TC-CONTINUE or TC-END: either as the only component of the message or grouped with an existing component. The choice is up to the MAP-Service User.

  If the received message contains only unknown operations the MAP-Service User shall send the TC-REJECT components in a TC-CONTINUE message to the peer entity, if the dialogue state allows it.

  If the received message contains unknown operations and standard operations and the standardised structure of the dialogue requires the response to the standard operation to be sent within a TC-END message, then the MAP-Service User may send the response to the standard operations and the TC-REJECT components for the unknown operations in a TC-CONTINUE message followed by a TC-END message. A specific distribution of the components to the TC messages or a specific sequencing of components is not required.

Note that SDLs of chapters 19 - 25 do not show the report to the MAP-Service User about the reception of the unknown operation. This has been done for the sake of simplicity of description; the MAP PM may inform the MAP-Service User.

The sender of the unknown operation shall ensure that there is enough room in the used message for the unknown operation.

15.1.2 Receipt of a MAP-OPEN request primitive

On receipt of a MAP-OPEN request primitive the behaviour of the MAP PM shall be as follows:

The MAP PM shall accept zero, one or several user request primitives until a MAP-DELIMITER request primitive is received.

For each user request primitive, the MAP PM shall request the invocation of the associated operation using the TC-INVOKE service. See clause 15.6 for a description of the associated SSMs.

On receipt of the MAP-DELIMITER request primitive the MAP PM shall issue a TC-BEGIN request primitive. The application-context-name as well as the user information parameter (if any) shall be mapped to the corresponding TC-BEGIN parameters.

The requesting MAP PM waits for a TC indication primitive and does not accept any other primitive from its user, except a MAP-U-ABORT request or a MAP-CLOSE request.

15.1.3 Receipt of a TC-BEGIN indication

On receipt of a TC-BEGIN indication primitive, the MAP PM shall:

- if no application-context-name is included in the primitive and if the "Components present" indicator indicates "no components", issue a TC-U-ABORT request primitive (note 2). The local MAP-User is not informed;

- if no application-context-name is included in the primitive and if presence of components is indicated, wait for the first TC-INVOKE primitive, and derive a version 1 application-context-name from the operation code according to table 15.1/1 (note 1).

NOTE 1: In some cases, it may be necessary to analyse the operation argument.

Then:

a) if no application-context-name can be derived (i.e. the operation code does not exist in MAP V1 specifications), the MAP PM shall issue a TC-U-ABORT request primitive (note 2). The local MAP-User is not informed.

b) if an application-context-name can be derived and if it is acceptable from a load control point of view, the MAP PM shall:

i) if this primitive requests the beginSubscriberActivity operation, the MAP PM shall check whether more components have been received associated with this operation. If more components are present, the MAP PM shall issue a MAP-OPEN indication primitive with the version 1 application-context-name "networkFunctionalSsContext-v1". The Destination-reference shall include the IMSI taken from the argument of the beginSubscriberActivity operation; the Originating-reference shall cover the originatingEntityNumber.

A beginSubscriberActivity operation that is not associated with any other Component shall be rejected by the MAP PM by issuing a TC-U-ABORT request primitive (note 2). The local MAP-User shall not be informed.

ii) otherwise, the MAP PM shall issue a MAP-OPEN indication primitive with the version 1 application-context-name set according to table 15.1/1. DestinationReference and OriginatingReference must not be included in the MAP-OPEN indication primitive.

Then the MAP PM shall function in a way that the dialogue responding MAP behaves as specified in the GSM phase 1 protocol (latest version of GSM 09.02 phase 1).

NOTE 2: If no AARQ apdu was included in the BEGIN message, TC (Component Sub-layer) will not include an AARE apdu or an ABRT apdu in a TR-U-ABORT request primitive that is to be issued on receipt of a TC-U-ABORT request primitive from the local MAP service provider.

c) if an application-context-name can be derived but if it is not acceptable from a load control point of view, the MAP PM shall ignore this dialogue request and not inform the MAP-user;

- if a version 1 application-context-name is included, the MAP PM shall issue a TC-U-ABORT request primitive with abort-reason "User-specific" and user-information "MAP-ProviderAbortInfo" indicating "abnormalDialogue". The local MAP-user shall not be informed.

- if an application-context-name different from version 1 is included in the primitive and if User-information is present, the User-information must constitute a syntactically correct MAP-OPEN dialogue PDU. Otherwise a TC-U-ABORT request primitive with abort-reason "User-specific" and user-information "MAP-ProviderAbortInfo" indicating "abnormalDialogue" shall be issued and the local MAP-user shall not be informed.

  - if no User-information is present it is checked whether presence of User Information in the TC-BEGIN indication primitive is required for the received application-context-name. If User Information is required but not present, a TC-U-ABORT request primitive with abort-reason "User-specific" and user-information "MAP-ProviderAbortInfo" indicating "abnormalDialogue" shall be issued. The local MAP-user shall not be informed.

- if an application-context-name different from version 1 is received in a syntactically correct TC-BEGIN indication primitive but is not acceptable from a load control point of view, the MAP PM shall ignore this dialogue request. The MAP-user is not informed.

- if an application-context-name different from version 1 is received in a syntactically correct TC-BEGIN indication primitive and if it is acceptable from a load control point of view, the MAP PM shall check whether the application-context-name is supported.

NOTE 3: Unknown application-context-names are treated like unsupported ones.

If it is, the MAP PM shall issue a MAP-OPEN indication primitive with all parameters (application-context-name included) set according to the value of the corresponding parameter of the TC-BEGIN indication primitive.

The MAP PM shall then process any other indication primitives received from TC as described in clause 15.6. Once all the received components have been processed, the MAP PM shall inform the local MAP service user by a MAP-DELIMITER indication primitive.

If the TC-BEGIN indication primitive is not associated with any component, the MAP PM shall inform the MAP User by a MAP-DELIMITER indication primitive.

Once all the received primitives have been processed, the MAP PM does not accept any primitive from the provider and waits for a MAP-OPEN response primitive from its user.

- if an application-context-name different from version 1 is received in a syntactically correct TC-BEGIN indication primitive and if it is acceptable from a load control point of view but the application-context-name is not supported, the MAP PM shall issue a TC-U-ABORT request primitive with abort-reason indicating "application-context-not-supported". If an alternative application-context-name cannot be offered, the received application-context-name shall be returned in the TC-U-ABORT Req primitive.

In the following cases an alternative application-context can be offered and its name included in the TC-U-ABORT Req primitive:

a) if an application-context of version 2 or higher is requested, but only version 1 application-context supported, then the v1 application context shall be returned;

b) if an application-context of version 3 or higher is requested, but only version 2 application-context supported, then the v2 application context shall be returned.

c) if an application-context of version 4 or higher is requested, but only version 3 application-context supported, then the v3 application context shall be returned.

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
NOTE 1: These symbolic names refer to the object identifier value defined in clause 17 and allocated to each application-context used for the MAP. NOTE 2: The choice between the application contexts is based on the parameters received in the operation.	

15.1.4 Receipt of a MAP-OPEN response

On receipt of a MAP-OPEN response primitive indicating that the dialogue is accepted, the MAP PM shall build a MAP-Accept PDU if the user-information parameter is included in the response primitive and accept any MAP specific service request or service response until a MAP-DELIMITER request or a MAP-CLOSE request is received from the MAP user. The MAP PM shall process the MAP specific primitives as described in clause 15.6. The MAP PM shall then issue a TC-CONTINUE request primitive after it receives the MAP-DELIMITER request primitive if no MAP-CLOSE request primitive has been received, otherwise it shall issue a TC-END request primitive. In both cases the MAP-Accept PDU (if any) is included in the user-information parameter of the TC primitive.

If the dialogue is not associated with a version 1 application context, the MAP PM shall include the application-context-name in the TC primitive.

If no MAP-CLOSE request has been received, the MAP PM waits for a request primitive from its user or an indication primitive from TC.

On receipt of a MAP-OPEN response primitive indicating that the dialogue is not accepted, the MAP PM shall build a MAP-Refuse PDU and request its transfer using the TC-U-ABORT req primitive (abort reason = user specific).

15.1.5 Receipt of the first TC-CONTINUE ind

On receipt of the first TC-CONTINUE indication primitive for a dialogue, the MAP PM shall check the value of the application-context-name parameter. If this value matches the one used in the MAP-OPEN request primitive, the MAP PM shall issue a MAP-OPEN confirm primitive with the result parameter indicating "accepted", then process the following TC component handling indication primitives as described in clause 15.6, and then waits for a request primitive from its user or an indication primitive from TC, otherwise it shall issue a TC-U-ABORT request primitive with a MAP-providerAbort PDU indicating "abnormal dialogue" and a MAP-P-ABORT indication primitive with the "provider-reason" parameter indicating "abnormal dialogue".

15.1.6 Receipt of a TC-END ind

On receipt of a TC-END indication primitive in the dialogue initiated state, the MAP PM shall check the value of the application-context-name parameter. If this value does not match the one used in the MAP-OPEN request primitive, the MAP PM shall discard any following component handling primitive and shall issue a MAP-P-ABORT indication primitive with the "provider-reason" parameter indicating "abnormal dialogue".

Otherwise it shall issue a MAP-OPEN confirm primitive with the result parameter set to "accepted" and process the following TC component handling indication primitives as described in clause 15.6; then it shall issue a MAP-CLOSE indication primitive and return to idle all state machines associated with the dialogue.

15.1.7 Receipt of a TC-U-ABORT ind

On receipt of a TC-U-ABORT indication primitive in the "Dialogue Initiated" state with an abort-reason parameter indicating "ApplicationContextNotSupported", the MAP PM shall issue a MAP-OPEN confirm primitive with the result parameter indicating "Dialogue Refused" and the refuse-reason parameter indicating "ApplicationContextNotSupported".

On receipt of a TC-U-ABORT indication primitive in the "Dialogue Initiated" state with an abort-reason parameter indicating "User Specific" and without user information, the MAP PM shall issue a MAP-OPEN confirm primitive with the result parameter indicating "Dialogue Refused" and the refuse-reason parameter indicating "Potential Version Incompatibility".

On receipt of a TC-U-ABORT indication primitive in the "Dialogue Initiated" state with an abort-reason parameter indicating "User Specific" and a MAP-Refuse PDU included as user information, the MAP PM shall issue a MAP-OPEN confirm primitive with the result set to refused and the refuse reason set as received in the MAP Refuse PDU.

Receipt of a TC-U-ABORT indication primitive with abort-reason "User Specific" and with user information is described as part of abnormal termination (see clause 15.4.2).

15.1.8 Receipt of a TC-P-ABORT ind

On receipt of a TC-P-ABORT indication primitive in the "Dialogue Initiated" state with a P-abort parameter indicating "Incorrect Transaction Portion", the MAP PM shall issue a MAP-OPEN confirm primitive with the result parameter indicating "Dialogue Refused" and the refuse reason parameter indicating "Potential Version Incompatibility".

On receipt of a TC-P-ABORT indication primitive in the "Dialogue Initiated" state with a P-abort parameter indicating "No Common Dialogue Portion", the MAP PM shall issue a MAP-P-ABORT indication primitive with the provider reason parameter indicating "Version Incompatibility".

Receipt of a TC-P-ABORT indication primitive with another P-abort parameter value is described as part of abnormal termination (see clause 15.5.2).

15.2 Dialogue continuation

Once established the dialogue is said to be in a continuation phase.

Both MAP users can request the transfer of MAP APDUs until one of them requests the termination of the dialogue.

15.2.1 Sending entity

The MAP PM shall accept any MAP specific service request or response primitives and process them as described in clause 15.6.

On receipt of a MAP-DELIMITER request primitive, the MAP PM shall issue a TC-CONTINUE request primitive.

15.2.2 Receiving entity

On receipt of a TC-CONTINUE indication primitive the MAP PM shall accept zero, one or several TC component handling indication primitives and process them as described in clause 15.6.

15.3 Dialogue termination

Both the dialogue-initiator and the dialogue-responder have the ability to request the termination of a dialogue after it has been established.

The dialogue termination procedure is driven by the following events:

- a MAP-CLOSE request primitive;
- a TC-END indication primitive.

15.3.1 Receipt of a MAP-CLOSE request

On receipt of a MAP-CLOSE request primitive, the MAP PM shall issue a TC-END request primitive and, if applicable, return to idle the associated active SSMs. Note that if the release method parameter of the MAP-CLOSE request indicates "normal" the TC-END request primitive will trigger the transmission of components associated with any user specific request or response primitives which may have been issued after the last MAP-DELIMITER request.

15.3.2 Receipt of a TC-END indication

On receipt of a TC-END indication primitive, the MAP shall accept any component handling indication primitives and process them as described in clause 15.6.

Once all the received primitives have been processed, the MAP PM shall return to idle the associated SSMs and issue a MAP-CLOSE indication primitive.

15.4 User Abort

Both the dialogue-initiator and the dialogue-responder have the ability to abort a dialogue at any time.

The user abort procedure is driven by one of the following events:

- a MAP-U-ABORT request primitive;
- a TC-U-ABORT indication primitive carrying a MAP-user-abort PDU.

15.4.1 MAP-U-ABORT request

On receipt of a MAP-U-ABORT request the MAP PM shall construct a MAP-user-abort PDU from the user-reason and diagnostic parameters and issue a TC-U-ABORT request primitive. All state machines associated with the dialogue are returned to idle.

15.4.2 TC-U-ABORT ind

On receipt of a TC-U-ABORT indication carrying a MAP-user-abort PDU, the MAP PM shall issue a MAP-U-ABORT indication primitive. The user-reason and diagnostic information elements are mapped to the corresponding parameters of the MAP-U-ABORT indication primitive.

All state machines associated with the dialogue are returned to idle.

15.5 Provider Abort

The MAP has the ability to abort a dialogue at both the dialogue-initiator side and the dialogue-responder side.

The provider abort procedure is driven by one of the following events:

- a MAP PM error situation;
- a TC-P-ABORT indication primitive;

- a TC-U-ABORT indication primitive carrying a MAP-abort PDU.

15.5.1 MAP PM error situation

In the case of an abnormal situation detected at the MAP level during an established dialogue, the MAP PM shall:

- issue a MAP-P-ABORT indication primitive with the appropriate value of the provider-reason parameter;
- construct a MAP-abort PDU from the value of these parameters and request its transfer using a TC-U-ABORT request primitive.

15.5.2 TC-P-ABORT ind

On receipt of a TC-P-ABORT indication, the MAP PM shall issue a MAP-P-ABORT indication primitive.

All state machines associated with the dialogue are returned to idle.

15.5.3 TC-U-ABORT ind

On receipt of a TC-U-ABORT indication carrying a MAP-abort PDU, the MAP PM shall issue a MAP-P-ABORT indication primitive, with the appropriate value of the provider-reason parameter. The source parameter shall indicate "MAP-provider".

All state machines associated with the dialogue are returned to idle.

15.6 Procedures for MAP specific services

This clause describes the MAP procedures for MAP specific services.

These procedures are driven by the following types of events:

- a MAP specific request or a MAP specific MAP response primitive;
- a component handling primitive from TC.

A Service State Machine is activated on receipt of one of the following signals:

- a MAP request primitive, which activates a requesting SSM;
- a TC-INVOKE indication primitive without linked identifier, which activates a responding SSM.

For component handling primitives there are two types of events:

- events which activate a Service State Machine or which can be related to an existing one;

  The procedure elements driven by these events are described in clauses 15.6.1 to 15.6.4.

- events which cannot be related to a Service State Machine.

  The procedure elements driven by these events are described in clause 15.6.5.

15.6.1 Service invocation

The MAP specific procedures are initiated by the MAP request primitives.

On receipt of a MAP request primitive, the MAP PM shall build an operation argument from the parameters received in the request primitive and request the invocation of the associated operation using the TC-INVOKE procedure. If a linked ID parameter is inserted in the primitive this indicates a child service and implies that the operation on which the service is mapped is linked to the operation on which the parent service is mapped.

The mapping of MAP specific services on to remote operations is given in table 16.2/1.

15.6.2 Service invocation receipt

On receipt of a TC-INVOKE indication primitive, the MAP PM shall:

- if the invoke ID is already in use by an active service, request the transfer of a reject component using the TC-U-REJECT request primitive with the appropriate problem code (duplicated invokeID) and issue a MAP-NOTICE indication primitive with a diagnostic parameter set to "abnormal event received from the peer";

- if the operation code does not correspond to an operation supported by the application-context, request the transfer of a reject component using the TC-U-REJECT request primitive, with the appropriate problem code (unrecognised operation), and, if the dialogue version is lower than 3, issue a MAP-NOTICE indication primitive with a diagnostic parameter set to "abnormal event received from the peer";

- if a linked ID is included, perform the following checks: If the operation referred to by the linked ID does not allow linked operations or if the operation code does not correspond to a permitted linked operation, issue a TC-U-REJECT request primitive with the appropriate problem code (linked response unexpected or unexpected linked operation);

- if the type of the argument is not the one defined for the operation, request the transfer of a reject component using the TC-U-REJECT request primitive, with the appropriate problem code (mistyped parameter), and issue a MAP-NOTICE indication primitive with a diagnostic parameter set to "abnormal event from the peer";

- if the type of the argument is correct but the values of the information elements it contains do not permit the type of MAP service being invoked to be determined, request the transfer of an error component using the TC-U-ERROR request primitive with an error code set to "unexpected data value" and issue a MAP-NOTICE indication primitive with a diagnostic parameter set to "abnormal event from the peer";

NOTE 1: These checks are only relevant when there is not a one-to-one mapping between a service and an operation.

- if the type of the argument is correct but information elements required for the service being invoked are missing, request the transfer of an error component using the TC-U-ERROR request primitive with an error code set to "data missing" and issue a MAP-NOTICE indication primitive with a diagnostic parameter set to "abnormal event from the peer";

NOTE 2: These checks are only relevant when there is not a one-to-one mapping between a service and an operation.

- if the type of the argument is correct but contains information elements which are not relevant for the type of MAP service being invoked, request the transfer of an error component using the TC-U-ERROR request primitive with an error code set to "unexpected data value" and issue a MAP-NOTICE indication primitive with a diagnostic parameter set to "abnormal event from the peer";

NOTE 3: These checks are only relevant when there is not a one-to-one mapping between a service and an operation.

- Otherwise, issue the relevant MAP indication primitive to the MAP-service-user. If the service is to be user confirmed, the MAP PM waits for the corresponding response primitive.

15.6.3 Service response

For user confirmed services, the MAP PM shall accept a MAP response primitive and shall:

- if no error indication is included in the primitive and the service maps on to a class 1 or 3 operation, construct a result information element from the parameters received and request its transfer using the TC-RESULT-L service and optionally the TC-RESULT-NL service.

The TC-RESULT-NL services shall be used when the user specific parameters of the response primitives cannot be transferred in a single signalling frame and no segmenting mechanism is available from the underlying layers. The MAP PM shall issue one or several TC-RESULT-NL request primitives followed by a TC-RESULT-L primitive. The user parameters shall be split so that each portion contains sufficient information to construct a value compatible with the type defined for the result of the associated operation.

- if no error indication is included in the primitive and the service response maps on to a class 4 linked operation, construct an operation argument from the parameters received and request its transfer using the TC-INVOKE service for this class 4 linked operation. The operation to be invoked is deduced from the value of the result parameter of the service primitive;

- if an error indication is included in the primitive and the service maps on to a class 1 or 2 operation, either issue a TC-U-REJECT request primitive if the user error parameter indicates "resource limitation" or "initiating release", or construct an error parameter from the parameters received and request its transfer using the TC-U-ERROR request primitive. The error code should be the one associated with the value of the user error parameter of the response primitive.

NOTE: The only user errors that a MAP user can generate in addition to the list of errors attached to the operation which is associated with the service are: resource limitation and initiating release. Any other abnormal situation is detected either by the TC entity or by the MAP entity.

- if an error indication is received and the operation maps on to a class 3 operation, or if no error indication is received but the service maps on to a class 2 operation which has no class 4 linked operation, return the local service state machine to idle without requesting any service from TC.

15.6.4 Receipt of a response

A component handling indication primitive is considered as driving a response for a confirmed service if the invoke ID parameter value matches the one stored for the service, or if the linked ID parameter value matches the one stored for the service and the operation invoked is a class 4 operation. On receipt of a response (except a TC-L-CANCEL indication) for an unconfirmed service the MAP PM shall issue a MAP-NOTICE indication primitive with the appropriate provider error (return result unexpected or return error unexpected).

15.6.4.1 Receipt of a TC-RESULT-NL indication

If the type of the partial result parameter is not compatible with the one defined for the complete result of this operation, request the transfer of a reject component using the TC-U-REJECT request primitive, with the appropriate problem code (mistyped parameter) and issue a confirm primitive with the provider error parameter set to "invalid response received". The MAP PM shall also issue a TC-U-CANCEL request primitive so that all subsequent result components for this operation are discarded by TC.

Otherwise, store the value of the partial result parameter and wait for subsequent TC-RESULT-NL indication primitives until a TC-RESULT-L indication primitive is received.

15.6.4.2 Receipt of a TC-RESULT-L indication

If the type of the result parameter is not the one defined for the result of this operation, request the transfer of a reject component using the TC-U-REJECT request primitive, with the appropriate problem code (mistyped parameter), and issue a confirm primitive with the provider error parameter set to "invalid response received".

If the type of the result parameter is correct but does not contain all the information elements required by the service associated with the invocation, issue a confirm primitive with the provider error parameter set to "invalid response received".

NOTE 1: These checks are only relevant when there is not a one-to-one mapping between a service and an operation.

If the type of the result parameter is correct but contains information elements which are not relevant for the service associated with the invocation are missing, issue a confirm primitive with the provider error parameter set to "invalid response received".

NOTE 2: These checks are only relevant when there is not a one-to-one mapping between a service and an operation.

Otherwise, issue a MAP confirm primitive to the MAP-service-user mapping the result parameter of the TC-RESULT-L primitive on to the MAP specific parameters.

If partial results have been previously received, the value of the partial result parameters shall also be taken into account before performing the three previous checks.

15.6.4.3 Receipt of a TC-U-ERROR indication

If the error code is not defined for the MAP or is not one associated with the operation referred to by the invoke identifier, request the transfer of a reject component using the TC-U-REJECT request primitive, with the appropriate problem code (unrecognised error or unexpected error), and issue a confirm primitive with the provider error parameter set to "invalid response received".

If the type of the error parameter is not the one defined for this error, request the transfer of a reject component using the TC-U-REJECT request primitive, with the appropriate problem code (mistyped parameter), and issue a confirm primitive with the provider error parameter set to "invalid response received".

If the type of the error parameter is correct but does not contain all the information elements required by the service associated with the invocation, issue a confirm primitive with the provider error parameter set to "invalid response received".

NOTE 1: In some cases, it may be necessary to analyse the operation argument.

If the type of the error parameter is correct but its value includes information elements which are not relevant for the service associated with the invocation, issue a confirm primitive with the provider error parameter set to "invalid response received".

NOTE 2: In some cases, it may be necessary to analyse the operation argument.

Otherwise, issue a MAP confirm primitive to the MAP-service-user with the user error parameter set according to the received error code. If applicable the error parameter is mapped to the diagnostic parameter.

15.6.4.4 Receipt of a TC-INVOKE indication

A TC-INVOKE indication primitive is considered as carrying a possible response to a specific service if the linked ID refers to an active specific service and the associated operation is a class 4 operation. Note that the presence of a linked ID parameter in a TC-INVOKE primitive requesting a non class 4 operation indicates a child service whose procedures are the same as the procedures for the parent service.

On receipt of a TC-INVOKE indication confirming an active service, the MAP PM shall:

- if the operation code is not defined for MAP and the dialogue version is at least 3, issue a TC-U-REJECT request primitive with the appropriate problem code (unrecognised operation).

- if the operation code is not defined for MAP and the dialogue version is lower than 3, or if the operation referred to by the linked ID does not allow linked operations or if the operation code does not correspond to an allowed linked operation, issue a TC-U-REJECT request primitive with the appropriate problem code (unrecognised operation, linked response unexpected or unexpected linked operation). If the service is confirmed, the MAP shall also issue a Confirm primitive with provider error indication "unexpected response from the peer", otherwise it may issue a MAP-NOTICE indication primitive with an appropriate diagnostic "abnormal event received from the peer".

- otherwise issue a confirm primitive mapping the operation argument parameter to the user specific parameters and setting the result parameter according to the operation code of the linked operation.

15.6.4.5 Receipt of a TC-U-REJECT indication

On receipt of a TC-U-REJECT indication primitive which affects a pending service, the MAP PM shall issue a MAP confirm primitive to the MAP-service-user with the appropriate value of the provider error or user error parameter.

The mapping of TC invoke problem codes on to MAP Provider Error and MAP User Error parameter values is described in clause 16.

15.6.4.6 Receipt of a TC-L-REJECT indication

This event occurs when the local TC detects a protocol error in an incoming component which affects an active specific service.

On receipt of a TC-L-REJECT indicating "return result problem, unexpected return result", the MAP shall issue a confirm primitive with the parameter provider error indicating "unexpected response from the peer".

On receipt of a TC-L-REJECT indicating "return error problem, unexpected error result", the MAP shall issue a confirm primitive with the parameter provider error indicating "unexpected response from the peer".

Note that when the problem code indicates a general problem, it is considered that the event cannot be related to an existing SSM even if the invoke Id is provided by TC. This is because whether the invoke Id refers to a local or remote invocation is ambiguous. The behaviour of the MAP PM in such a case is described in clause 15.6.5.3.

15.6.4.7 Receipt of a TC-L-CANCEL indication

On receipt of a TC-L-CANCEL indication, the MAP PM shall:

- if the associated operation is a class 1 operation, issue a confirm primitive with the provider error cause indicating "no response from the peer";
- if the associated operation is a class 2 operation and no linked operations are defined for this operation, issue a confirm primitive without parameter (i.e. indicating implicitly the successful completion of the service);
- if the associated operation is a class 2 operation and has linked operations but none of them has been invoked, issue a confirm primitive with the provider error parameter indicating "service completion failure";
- if the associated operation is a class 2 operation and a linked operation invocation has already been received in response to this operation, ignore the primitive;
- if the associated operation is a class 3 operation, issue a confirm primitive with the provider error cause indicating "service completion failure";
- if the associated operation is a class 4 operation, ignore the primitive.

NOTE: When a TC-L-CANCEL ind primitive is received before the dialogue has been confirmed (i.e. no backward message is received by the dialogue initiator node), the MAP PM shall first issue a MAP-OPEN Cnf primitive with the result parameter indicating "accepted" (which means that the dialogue is considered as being implicitly accepted). Then, as indicated above, the TC-L-CANCEL Indication is interpreted according to the class of the operation to which it refers.

15.6.4.8 Receipt of a TC-NOTICE indication

If a TC-NOTICE indication primitive is received before the dialogue has been confirmed (i.e. no backward message is received by the dialogue initiator node), the MAP PM shall issue a MAP-OPEN Cnf primitive with the result parameter indicating Refused and a refuse reason Remote node not reachable”.

If a TC-NOTICE indication primitive is received after the dialogue has been confirmed, the MAP PM shall issue a MAP-NOTICE indication to the user, with a problem diagnostic indicating "message cannot be delivered to the peer".

15.6.5 Other events

This clause describes the behaviour of the MAP PM on receipt of a component handling indication primitive which cannot be related to any service or which does not affect a pending one. The MAP user is only informed that an abnormal event occurred during the associated dialogue. It is up to the MAP user to abort, continue or terminate the dialogue.

15.6.5.1 Receipt of a TC-U-REJECT

On receipt of a TC-U-REJECT indication primitive which does not affect an active SSM (i.e. indicating a return result or return error problem), the MAP PM shall issue a MAP-NOTICE indication primitive with the diagnostic parameter set to "response rejected by the peer".

This is also applicable for invoke problems related to a class 4 linked operation.

15.6.5.2 Receipt of a TC-R-REJECT indication

On receipt of a TC-R-REJECT indication (i.e. when a protocol error has been detected by the peer TC entity) which does not affect an active SSM, the MAP PM shall either discard this indication or issue a MAP-NOTICE indication primitive with the provider error indicating "abnormal event detected by the peer".

In case of notification, it is up to the MAP user to continue, abort or terminate the dialogue. Note also that for MAP V1 the reject component is received in an END message and therefore the dialogue is terminated anyway.

15.6.5.3 Receipt of a TC-L-REJECT indication

On receipt of a TC-L-REJECT indication primitive (i.e. when a protocol error has been detected by the local TC entity) which cannot be related to an active SSM, the MAP PM shall either discard this indication or issue a MAP-NOTICE indication primitive with the provider error indicating "abnormal event received from the peer".

In case of notification, it is up to the MAP user to continue, or to terminate the dialogue and implicitly trigger the transmission of the reject component or to abort the dialogue.

15.6.6 Parameter checks

As described in the previous clauses, the MAP PM performs a set of checks to ensure the correctness of the information elements received; these are:

- check if the syntax and encoding (note) of the operation argument, result or error parameter are correct.

NOTE: Depending on the implementation, encoding problems on the TC user portion may be detected at TC level or by the MAP user. In the second case the problem is reported in a similar manner to a syntactical problem.

The syntax shall be considered incorrect if a mandatory information element is missing in any constructed element or if the value of an information element is out of the range defined for the type it is supposed to belong to;

- if there is not a one-to-one mapping between a service and an operation:

  i) check if the value of the information elements (generally a single one) permits the MAP PM to determine the service associated with the operation invocation;

  ii) check that there are no information elements which are irrelevant for the indication or a confirm primitive to be issued;

- check if all the information elements required to build an indication or a confirm primitive are available.

However some additional checks may have to be performed by the MAP user (see clause 18).

15.6.7 Returning state machines to idle

Unlike TC invocation state machines, service state machines exist at both requestor and performer side.

A service state machine at the requestor side is returned to idle when the MAP-specific confirm primitive is issued or when the dialogue terminates.

A service state machine at the performer side is returned to idle on receipt of a MAP-specific response primitive from the MAP user, when the dialogue terminates or at expiry of an implementation dependent watch-dog timer which is started when the state machine is created.

15.6.8 Load control

As stated in the previous clauses, before issuing a MAP-OPEN indication primitive the MAP PM performs a check to verify if there are sufficient resources to open the dialogue taking into account possible overload conditions.

The decision is based on the priority allocated to the application-context whose name is explicitly included in the TC-BEGIN indication primitive or implied by the first operation invocation when V1 contexts are in use. How a V1 application-context-name is derived from an operation code is described in table 15.1/1.

The priority level allocated to each application-context is described in clause 5, tables 5.1/1 and 5.1/2.

16 Mapping on to TC services

16.1 Dialogue control

Dialogue control services are mapped to TC dialogue handling services. The TC-UNI service is not used by the MAP PM.

16.1.1 Directly mapped parameters

The following parameters of the MAP-OPEN request and indication primitives are directly mapped on to the corresponding parameters of the TC-BEGIN primitives:

- destination address;
- originating address.

16.1.2 Use of other parameters of dialogue handling primitives

16.1.2.1 Dialogue Id

The value of this parameter is associated with the MAP PM invocation in an implementation dependent manner.

16.1.2.2 Application-context-name

The application-context-name parameter of a MAP primitive is mapped to the application-context-name parameter of TC dialogue handling primitives according to the rules described in clause 15.1.

16.1.2.3 User information

The user information parameter of TC dialogue primitives is used to carry the MAP dialogue APDUs.

16.1.2.4 Component present

This parameter is used by the MAP PM as described in CCITT Recommendation Q.771. It is not visible to the MAP user.

16.1.2.5 Termination

The value of this parameter of the TC-END request primitive is set by the MAP PM on the basis of the release method parameter of the MAP-CLOSE request primitive, except when the dialogue state machine is in the state DIALOGUE INITIATED, in which case the Termination parameter shall always indicate "pre-arranged end".

16.1.2.6 P-Abort-Cause

Values of the P-abort-cause parameter are mapped to the values of the provider-reason parameter of the MAP-P-ABORT indication primitive according to table 16.1/1, except in the dialogue initiated phase for the "incorrectTransactionPortion" and "noCommonDialoguePortion" values which are mapped to the "potential incompatibility problem" value of the refuse-reason parameter of the MAP-OPEN cnf primitive. The source parameter in the MAP-P-ABORT ind takes the value "TC problem".

16.1.2.7 Quality of service

The quality of service of TC request primitives is set by the MAP as shown below.

- Return option: "Return message on error" or "Discard message on error" as required by the network operator;
- Sequence control: "Sequence guaranteed" or "Sequence result not guaranteed" as required by the network operator;
- "Sequence guaranteed" shall be used when a segmented result is to be transferred (e.g. subscriber data in response to SendParameters). It may also be appropriate to use Sequence guaranteed when a series of InsertSubscriberData, ProcessAccessSignalling or ForwardAccessSignalling operations is used.

It is essential that the TC message which indicates acceptance of a dialogue opening request is received by the dialogue initiator before any subsequent message in that dialogue; otherwise the dialogue opening will fail. The dialogue responder shall ensure that this requirement is met by:

- Sending the dialogue acceptance message in a TC-END, if the dialogue structure requires it; or
- Using "Sequence guaranteed", if the dialogue acceptance message is sent in a TC-CONTINUE; or
- Waiting until the dialogue acceptance message has been acknowledged by the dialogue initiator before sending a subsequent message, if the dialogue acceptance message is sent in a TC-CONTINUE.

Table 16.1/1: Mapping of P-Abort cause in TC-P-ABORT indication on to provider-reason in MAP-P-ABORT indication

TC P-Abort cause	MAP provider-reason
unrecognised message type	provider malfunction
unrecognised transaction Id	supporting dialogue released
badlyFormattedTransactionPortion	provider malfunction
incorrectTransactionPortion	provider malfunction (note)
resourceLimitation	resource limitation
abnormalDialogue	provider malfunction
noCommonDialoguePortion	version incompatibility
NOTE: Or version incompatibility in the dialogue initiated phase.	

16.2 Service specific procedures

Specific services are mapped to TC component handling services.

16.2.1 Directly mapped parameters

The Invoke Id parameter of the MAP request and indication primitive is directly mapped on to the Invoke Id parameter of the component handling primitives.

16.2.2 Use of other parameters of component handling primitives

16.2.2.1 Dialogue Id

The value of this parameter is associated with the MAP PM invocation in an implementation dependent manner.

16.2.2.2 Class

The value of this parameter is set by the MAP PM according to the type of the operation to be invoked.

16.2.2.3 Linked Id

When a service response is mapped to a class 4 operation, the value of this parameter is set by the MAP PM and corresponds to the value assigned by the user to the initial service request (i.e. the value of the invoke ID parameter of

the request primitive). Otherwise if such a parameter is included in MAP request/indication primitives it is directly mapped to the linked ID parameter of the associated TC-INVOKE request/indication primitives.

16.2.2.4 Operation

When mapping a request primitive on to a Remote Operations PDU (invoke), the MAP PM shall set the operation code according to the mapping described in table 16.2/1.

When mapping a response primitive on to a Remote Operations service, the MAP PM shall set the operation code of the TC-RESULT-L/NL primitive (if required) to the same value as the one received at invocation time.

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_AUTHENTICATION_FAILURE_REPORT	authenticationFailureReport
MAP-ANY-TIME-MODIFICATION	anyTimeModification
MAP-ANY-TIME-SUBSCRIPTION-INTERROGATION	anyTimeSubscriptionInterrogation
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

16.2.2.5 Error

The error parameter in a TC-U-ERROR indication primitive is mapped to the user error parameter in the MAP confirm primitive of the service associated with the operation to which the error is attached.

The user error parameter in MAP response primitives is mapped to the error parameter of the TC-U-ERROR request primitive, except for "initiating-release" and "resource-limitation" which are mapped to the problem code parameter of the TC-U-REJECT request primitive.

16.2.2.6 Parameters

The parameters of MAP specific request and indication primitives are mapped to the argument parameter of TC-INVOKE primitives.

The parameters of MAP specific response and confirm primitives are mapped to the result parameter of TC-RESULT-L primitives, the parameter of TC-U-ERROR primitives or the argument of TC-INVOKE primitives when mapping on linked class 4 operations is used.

16.2.2.7 Time out

The value of this parameter is set by the MAP PM according to the type of operation invoked.

16.2.2.8 Last component

This parameter is used by the MAP PM as described in CCITT Recommendation Q.711. It is not visible from the MAP user.

16.2.2.9 Problem code

16.2.2.9.1 Mapping to MAP User Error

The following values of the user error parameter are mapped as follows to values of the TC problem code parameter. These values are generated by the MAP user. This mapping is valid from the TC-U-REJECT indication primitive to the MAP confirm service primitive and from the MAP response service primitive to the TC-U-REJECT request primitive.

Table 16.2/2: Mapping of MAP User Error parameter on to TC problem code in TC-U-REJECT primitives

MAP User Error	TC problem code
resource limitation	resource limitation
initiating release	initiating release

16.2.2.9.2 Mapping to MAP Provider Error parameter

The following values of the TC problem code parameter of the TC-U-REJECT indication primitive are mapped as follows to values of the MAP Provider Error parameter of the MAP confirm primitive.

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
unrecognised operation	service not supported
mistyped parameter	mistyped parameter

The following values of the problem code parameters of the TC-L-REJECT primitive are mapped to values of the provider error parameter of the MAP confirm primitive as follows.

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

16.2.2.9.3 Mapping to diagnostic parameter

The following values of the problem code parameter of the TC-R-REJECT and TC-U-REJECT primitive are mapped to values of the diagnostic parameter of the MAP-NOTICE indication primitive as follows:

Table 16.2/5: Mapping of TC problem code of TC-R-REJECT and TC-U-REJECT on to diagnostic parameter

TC problem code	MAP diagnostic
General problem	- abnormal event detected by the peer
Invoke problem	
- unrecognised 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	
- unrecognised 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	
- unrecognised invoke ID	- response rejected by the peer
- return error unexpected	- response rejected by the peer
- unrecognised error	- response rejected by the peer
- unexpected error	- response rejected by the peer
- mistyped parameter	- response rejected by the peer

The following values of the problem code parameter of the TC-L-REJECT primitive are mapped to values of the diagnostic parameter of the MAP-NOTICE indication primitive as follows.

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	
- unrecognised linked ID	- abnormal event received from the peer
Return result problem	
- unrecognised invoke ID	- abnormal event received from the peer
Return error problem	
- unrecognised invoke ID	- abnormal event received from the peer

16.3 SDL descriptions

The following SDL specification describes a system which includes three blocks: MAP-user, MAP-provider and TC.

Such a system resides in each network component supporting MAP and communicates with its peers via the lower layers of the signalling network which are part of the environment.

Only the MAP-provider is fully described in this clause. The various types of processes which form the MAP-User block and the TC block are described respectively in clauses 18 to 25 of the present document and in CCITT Recommendation Q.774.

The MAP-Provider block communicates with the MAP_USER via two channels U1 and U2. Via U1 the MAP-provider receives the MAP request and response primitives. Via U2 it sends the MAP indication and confirm primitives.

The MAP-Provider block communicates with TC via two channels P1 and P2. Via P1 the MAP-Provider sends all the TC request primitives. Via P2 it receives all the TC indication primitives.

The MAP-Provider block is composed of the four following types of processes:

a) MAP_DSM: This type of process handles a dialogue. There exists one process instance per MAP dialogue.

b) LOAD_CTRL: This type of process is in charge of load control. There is only one instance of this process in each system.

c) PERFORMING_MAP_SSM: This type of process handles a MAP service performed during a dialogue. An instance of this process is created by the instance of the MAP_DSM process for each MAP-service to be performed.

d) REQUESTING_MAP_SSM: This type of process handles a MAP service requested during a dialogue. An instance of this process is created by the instance of the MAP_DSM process for each requested MAP-service.

A process MAP_DSM exchanges external signals with other blocks as well as internal signals with the other processes of the MAP-Provider block. The external signals are either MAP service primitives or TC service primitives.

The signal routes used by the various processes are organised as follows:

a) A process MAP_DSM receives and sends events from/to the MAP_user via signal route User1/User2. These routes use respectively channel U1 and U2.

b) A process MAP_DSM receives and sends events from/to the TC via signal route Tc1/Tc2. These routes use respectively channel P1 and P2.

c) A process MAP_DSM receives and sends events from/to the LOAD_CTRL process via signal route Load1/Load2. These routes are internal.

d) A process MAP_DSM sends events to the PERFORMING_MAP_SSM processes via signal route Intern1. This route is internal.

e) A process MAP_DSM sends events to the REQUESTING_MAP_SSM processes via signal route Intern2. This route is internal.

f) A process MAP_PERFORMING_SSM sends events to the MAP_USER via signal route User4. This route uses channel U2.

g) A process MAP_PERFORMING_SSM sends events to TC via signal route Tc3. This route uses channel P1.

h) A process MAP_REQUESTING_SSM sends events to the MAP_USER via signal route User5. This route uses channel U2.

j) A process MAP_REQUESTING_SSM sends events to TC via signal route Tc4. This route uses channel P1.

09.02 version 6.6.0

System MAP_STACK 16.2_1(1)

Figure 16.2/1:

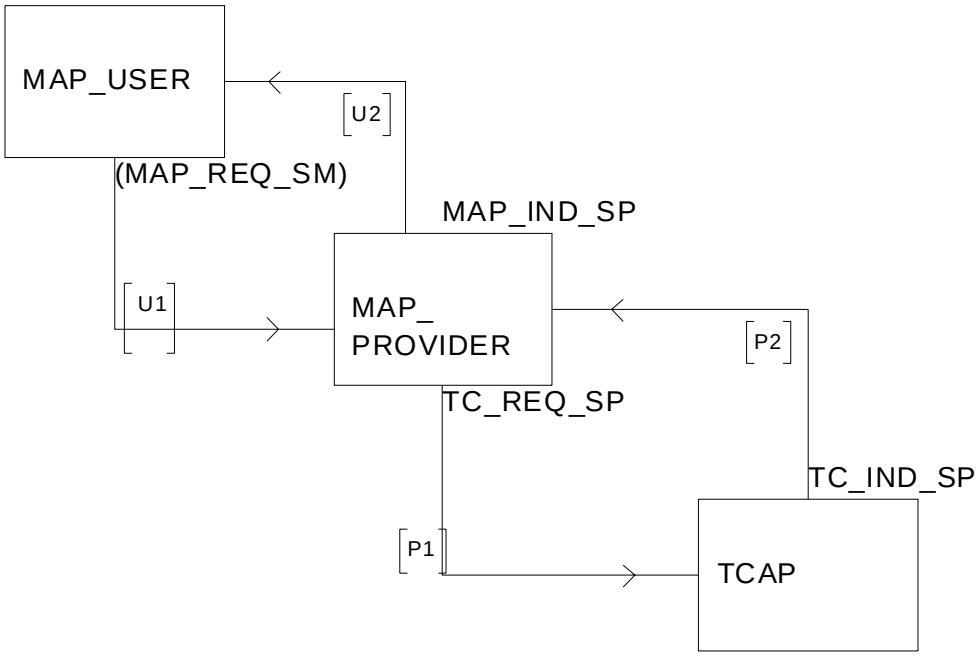


Figure 16.2/1: System MAP_STACK

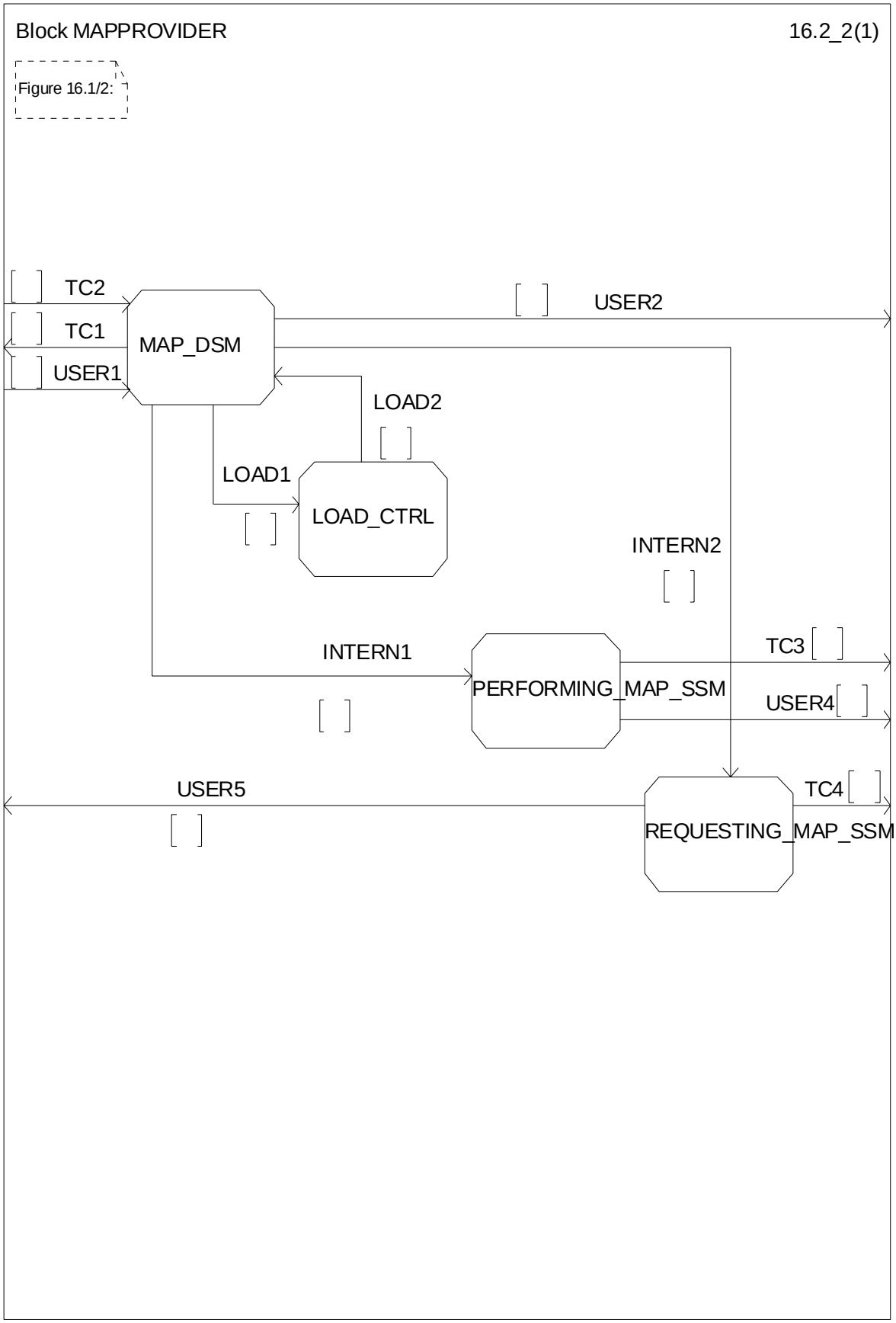


Figure 16.2/2: Block MAPPROVIDER

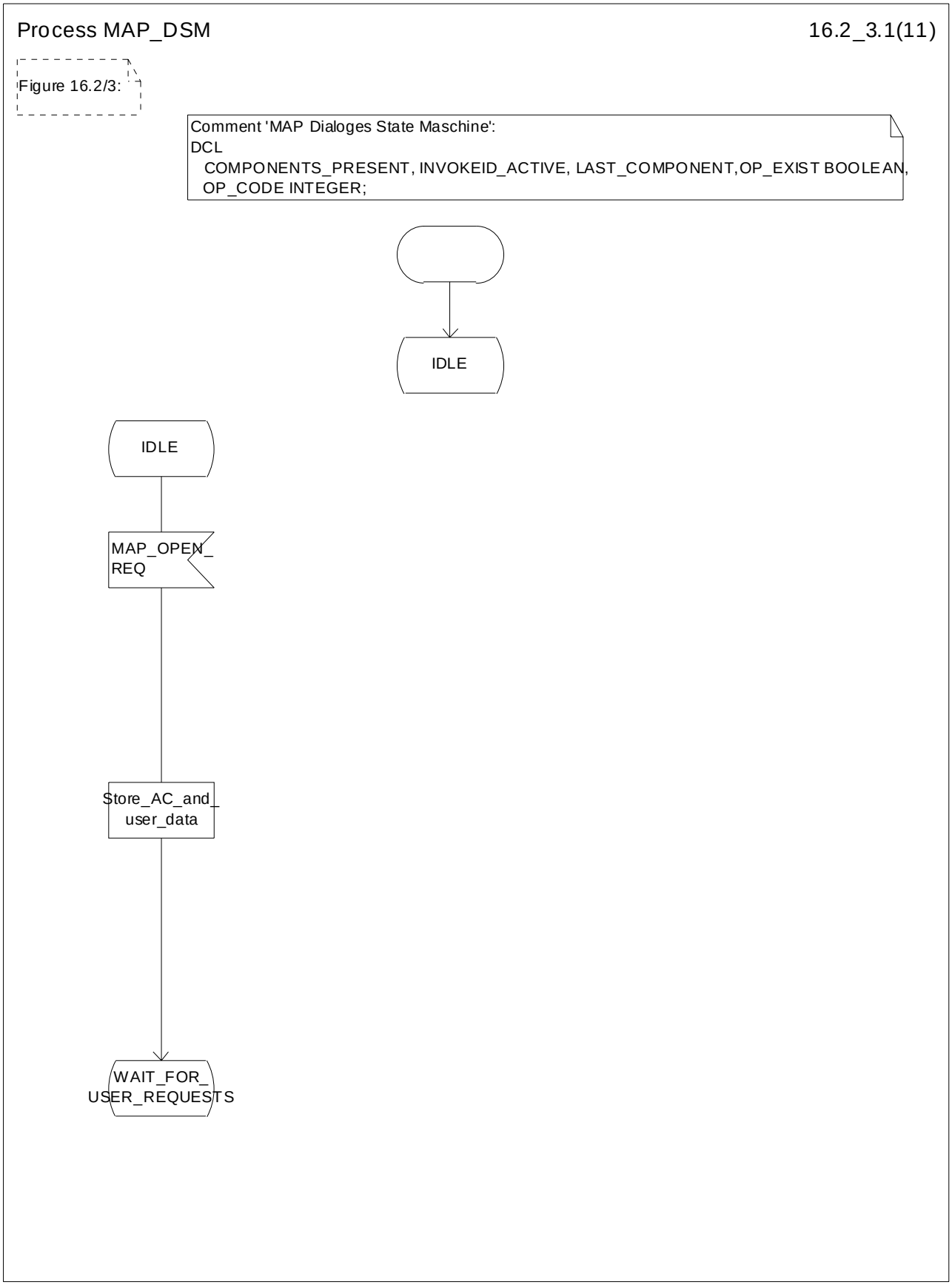


Figure 16.2/3 (sheet 1 of 11): Process MAP_DSM

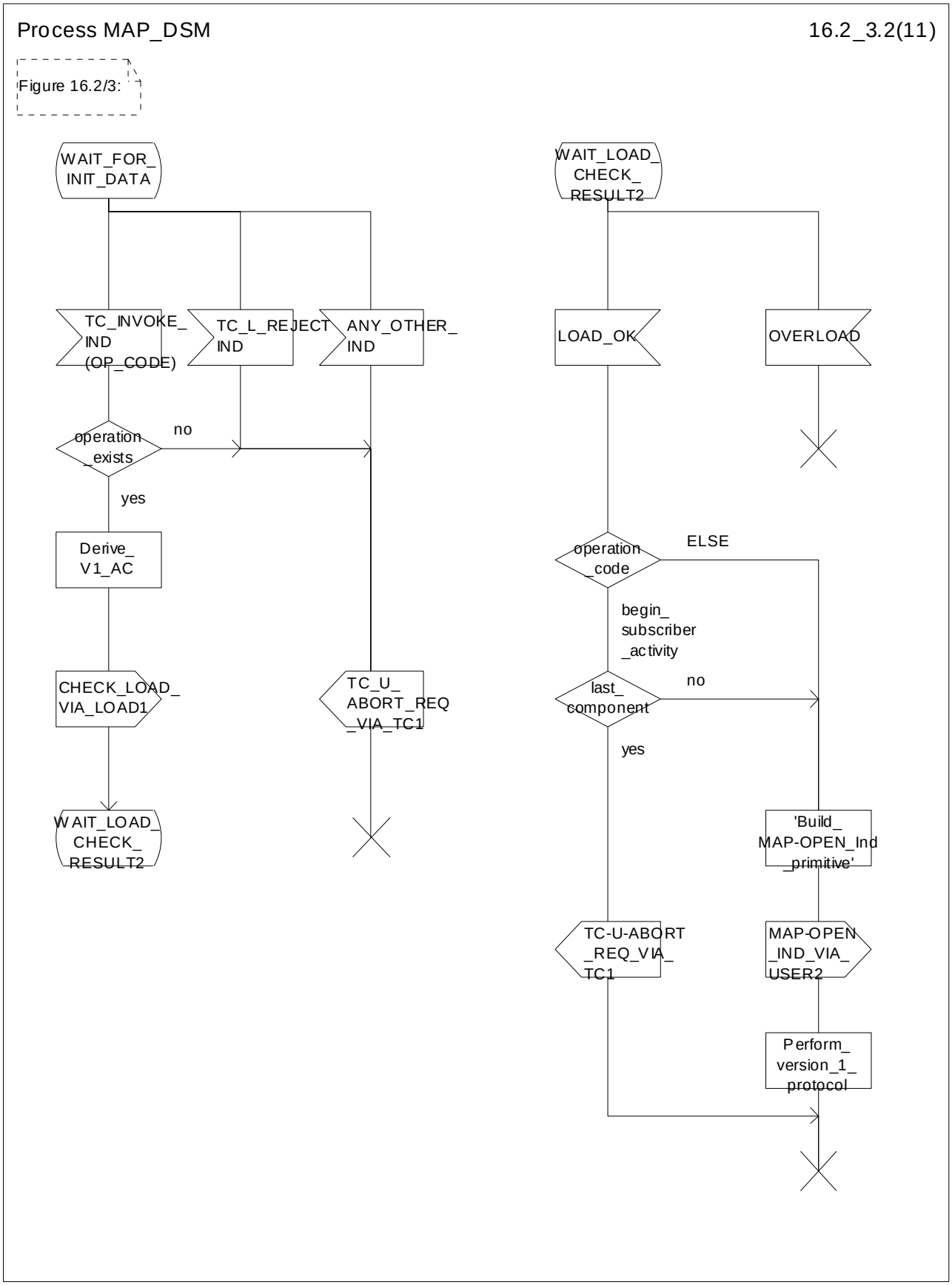


Figure 16.2/3 (sheet 2 of 11): Process MAP_DSM

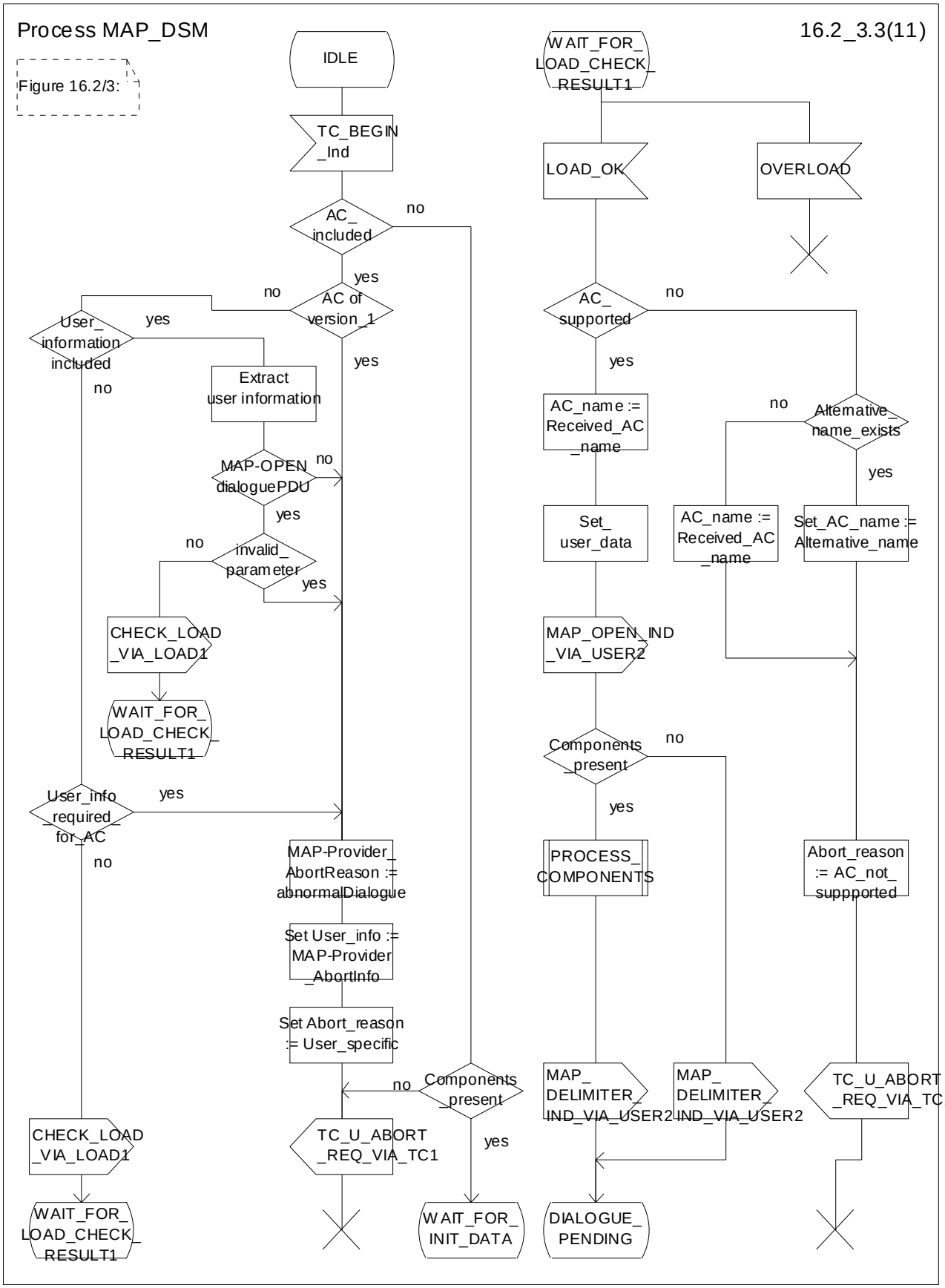


Figure 16.2/3 (sheet 3 of 11): Process MAP_DSM

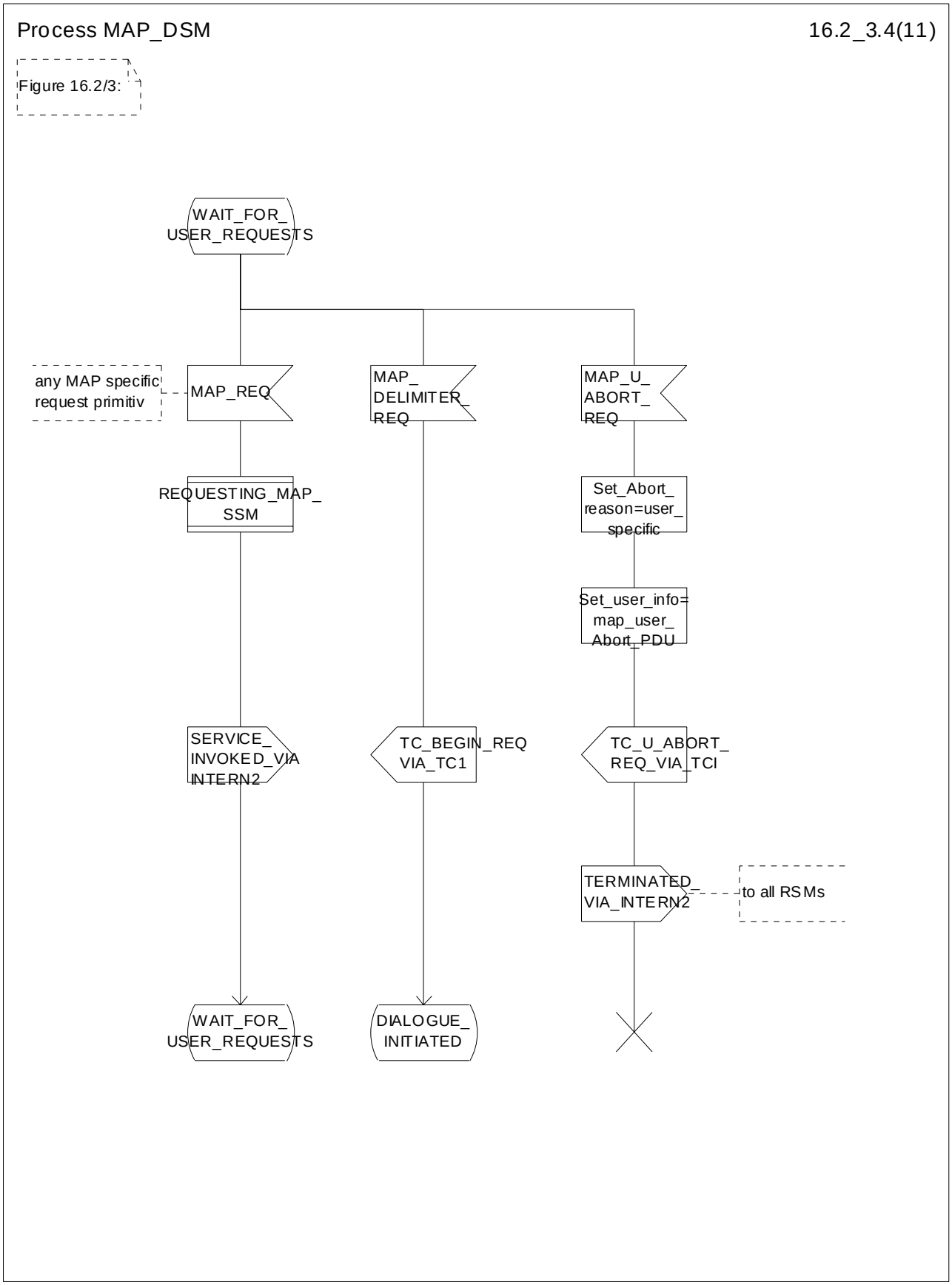


Figure 16.2/3 (sheet 4 of 11): Process MAP_DSM

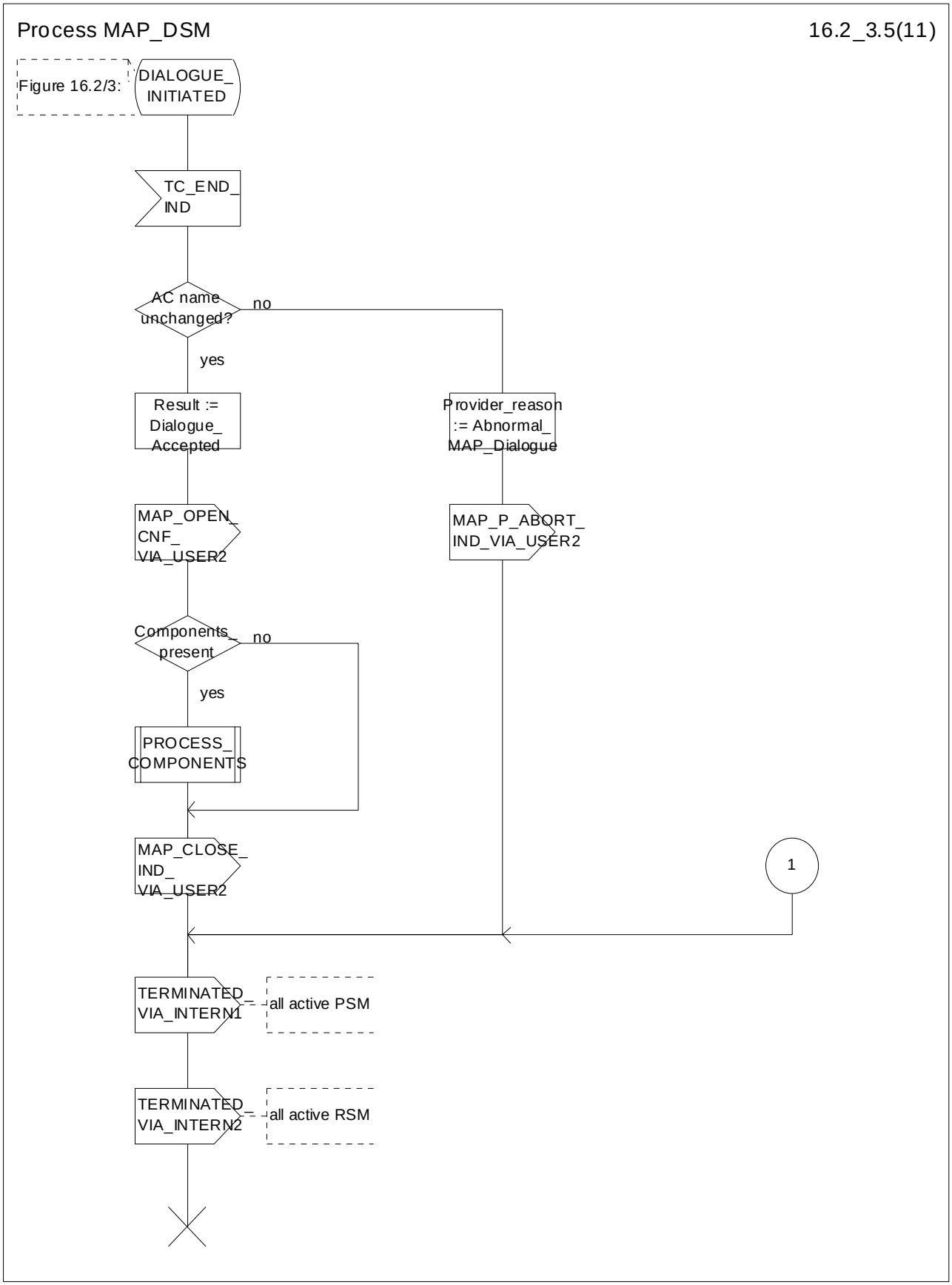


Figure 16.2/3 (sheet 5 of 11): Process MAP_DSM

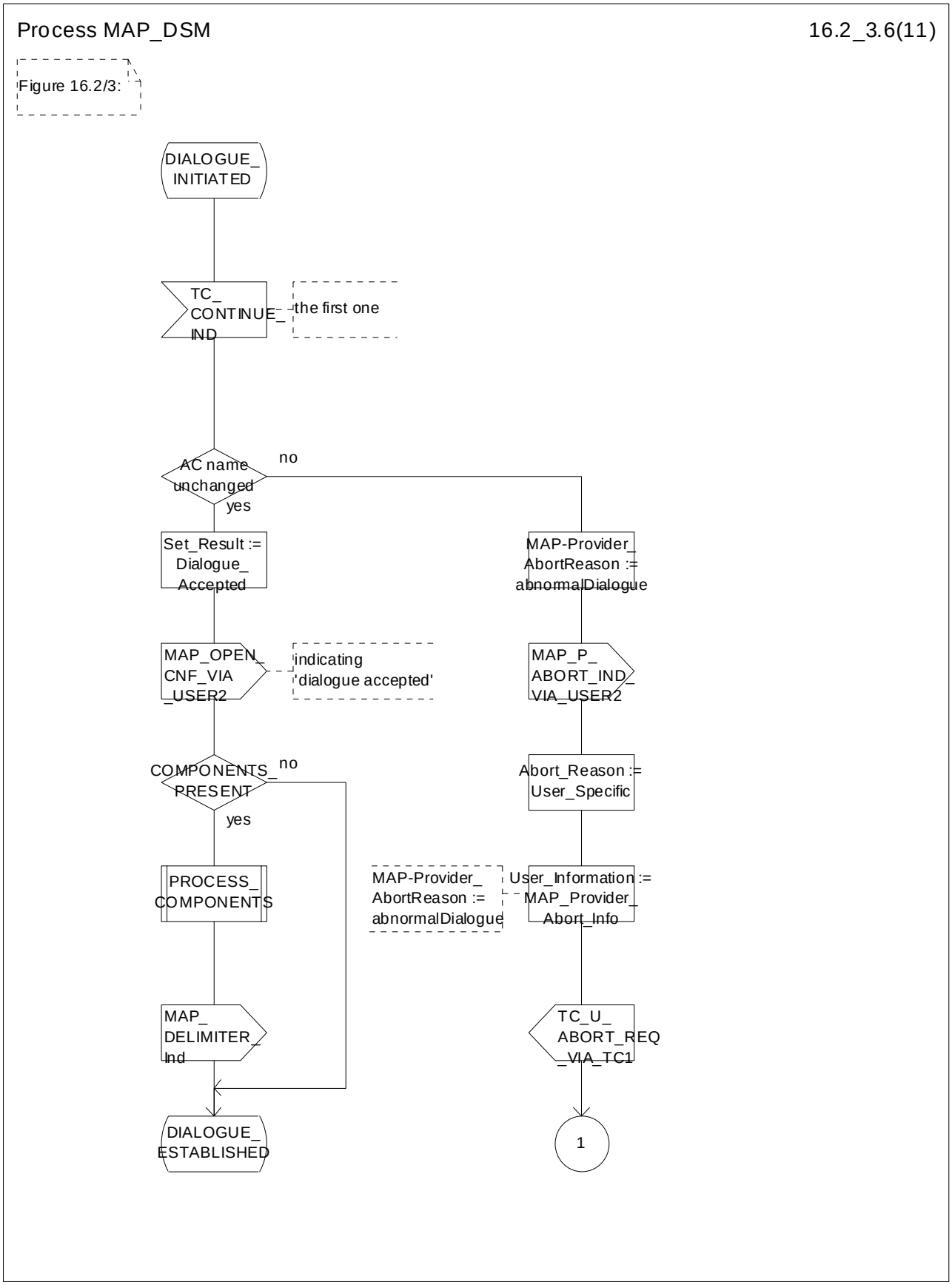


Figure 16.2/3 (sheet 6 of 11): Process MAP_DSM

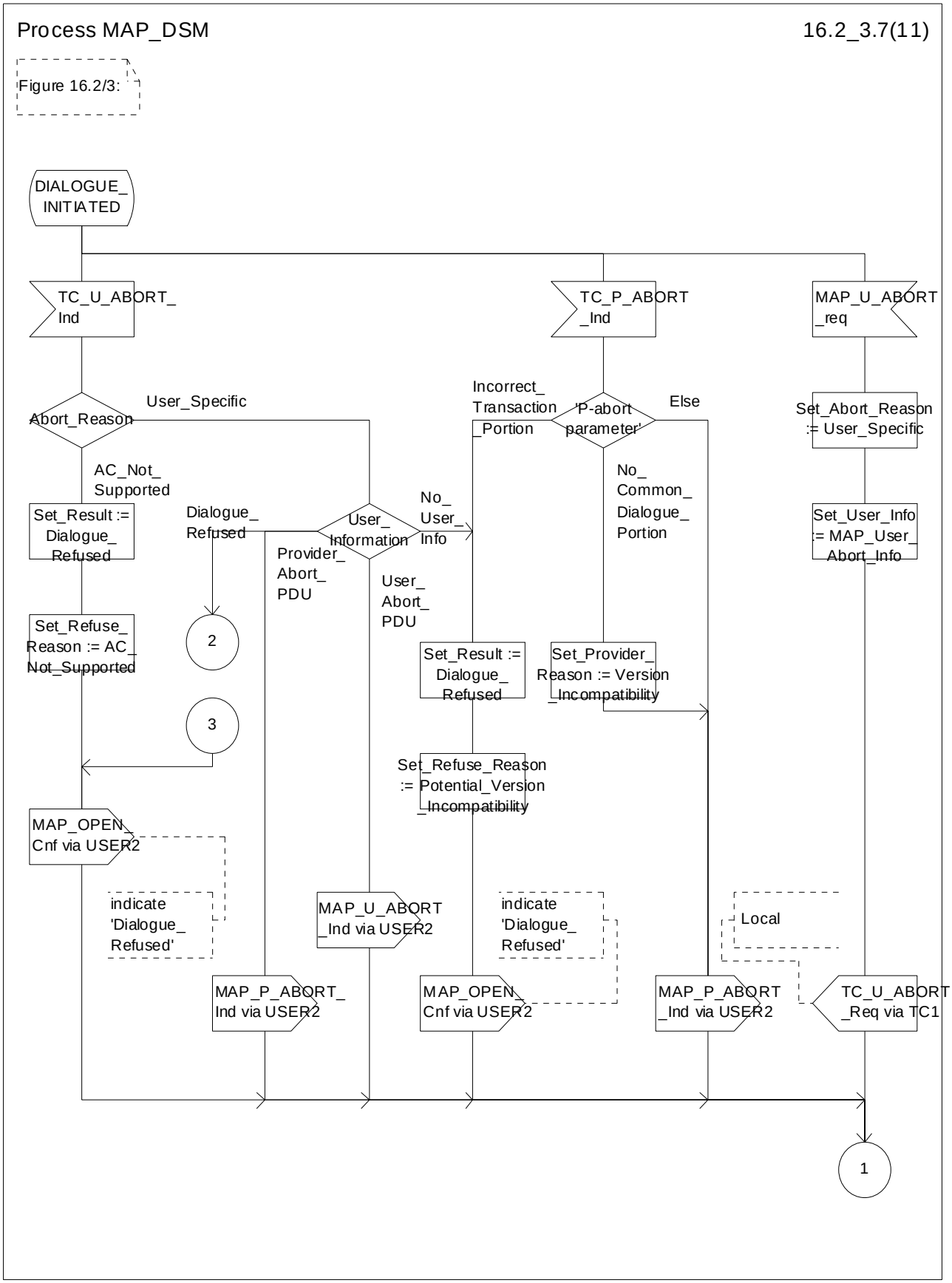


Figure 16.2/3 (sheet 7 of 11): Process MAP_DSM

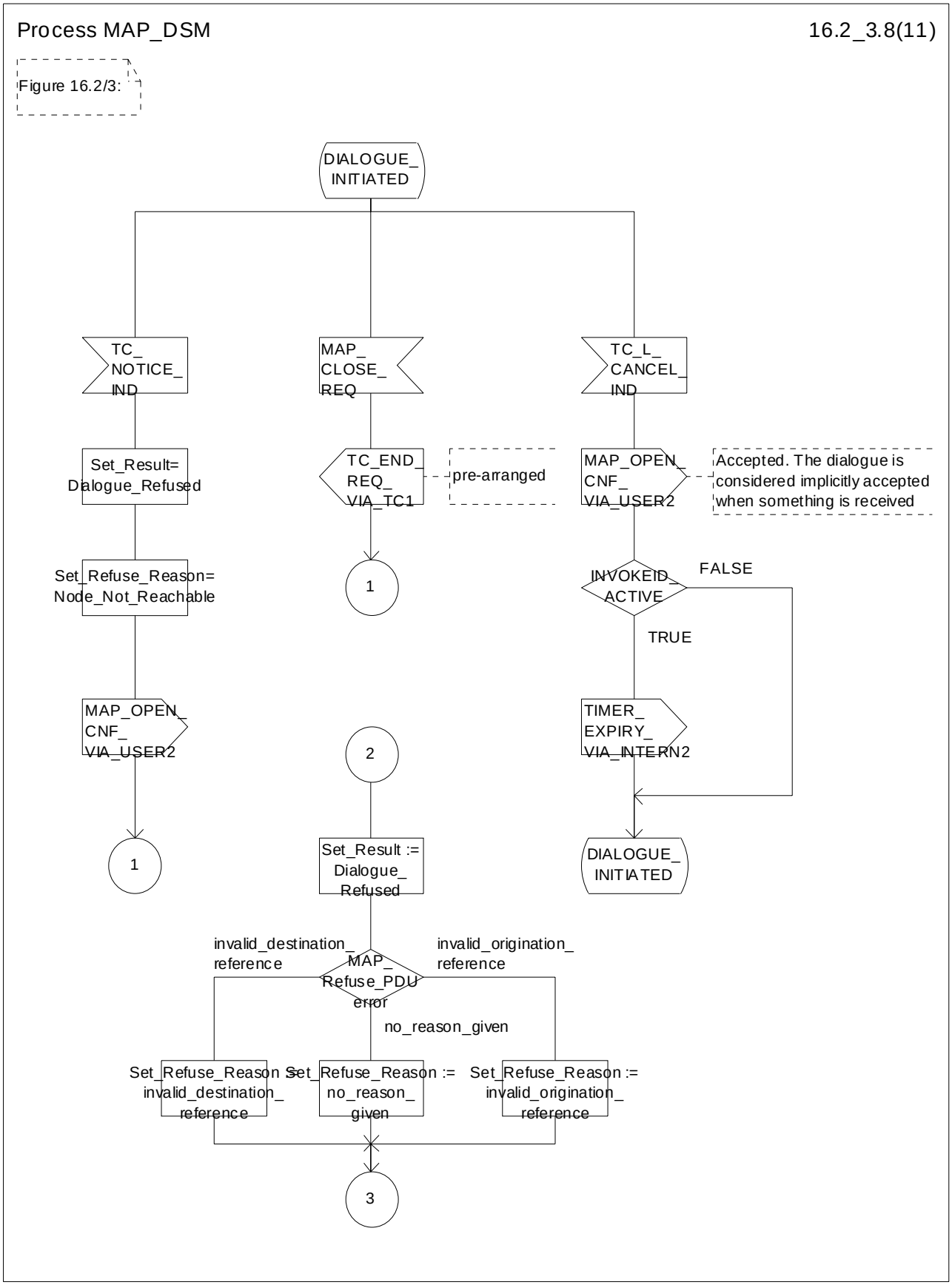


Figure 16.2/3 (sheet 8 of 11): Process MAP_DSM

Process MAP_DSM 16.2_3.9(11)

Figure 16.2/3:

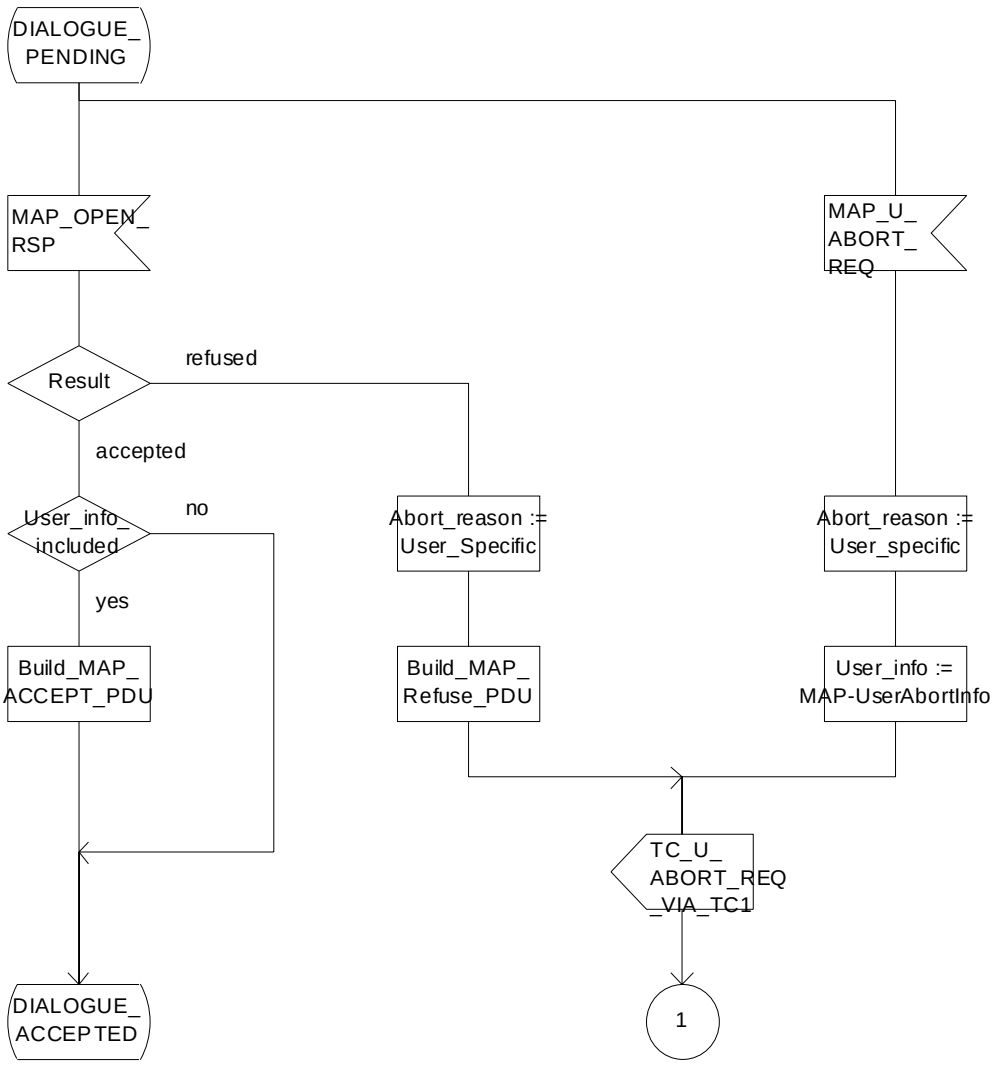


Figure 16.2/3 (sheet 9 of 11): Process MAP_DSM

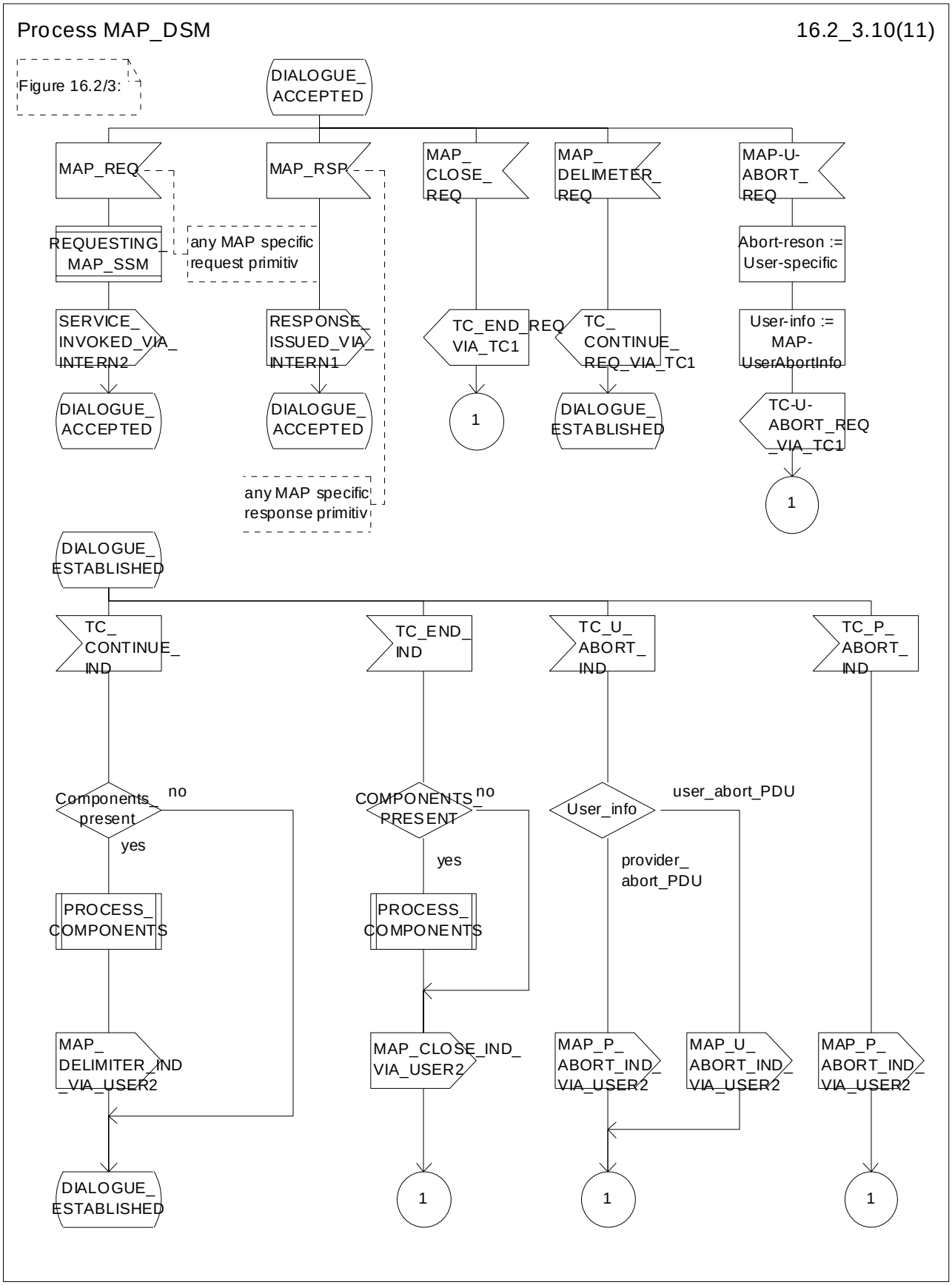


Figure 16.2/3 (sheet 10 of 11): Process MAP_DSM

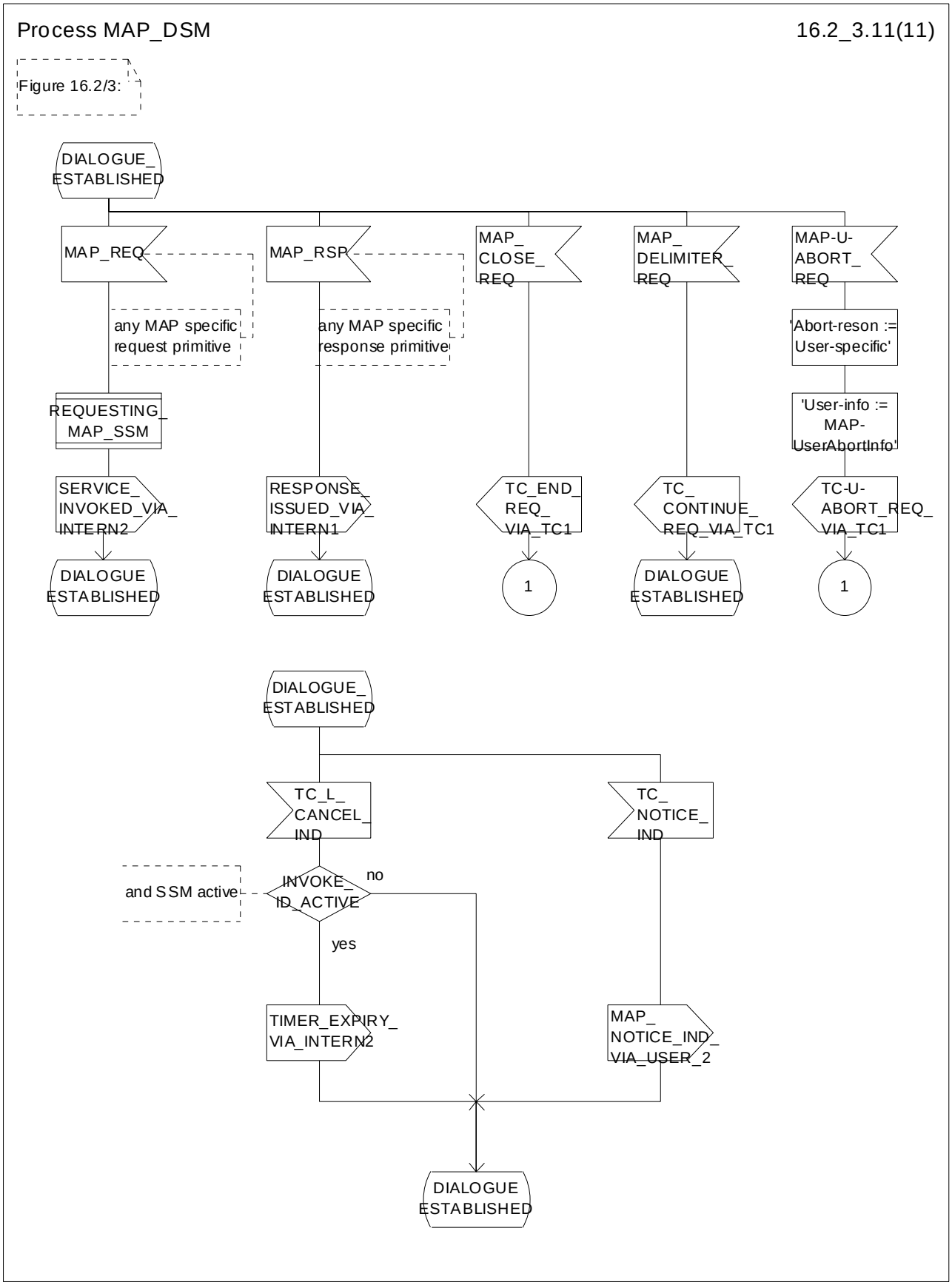


Figure 16.2/3 (sheet 11 of 11): Process MAP_DSM

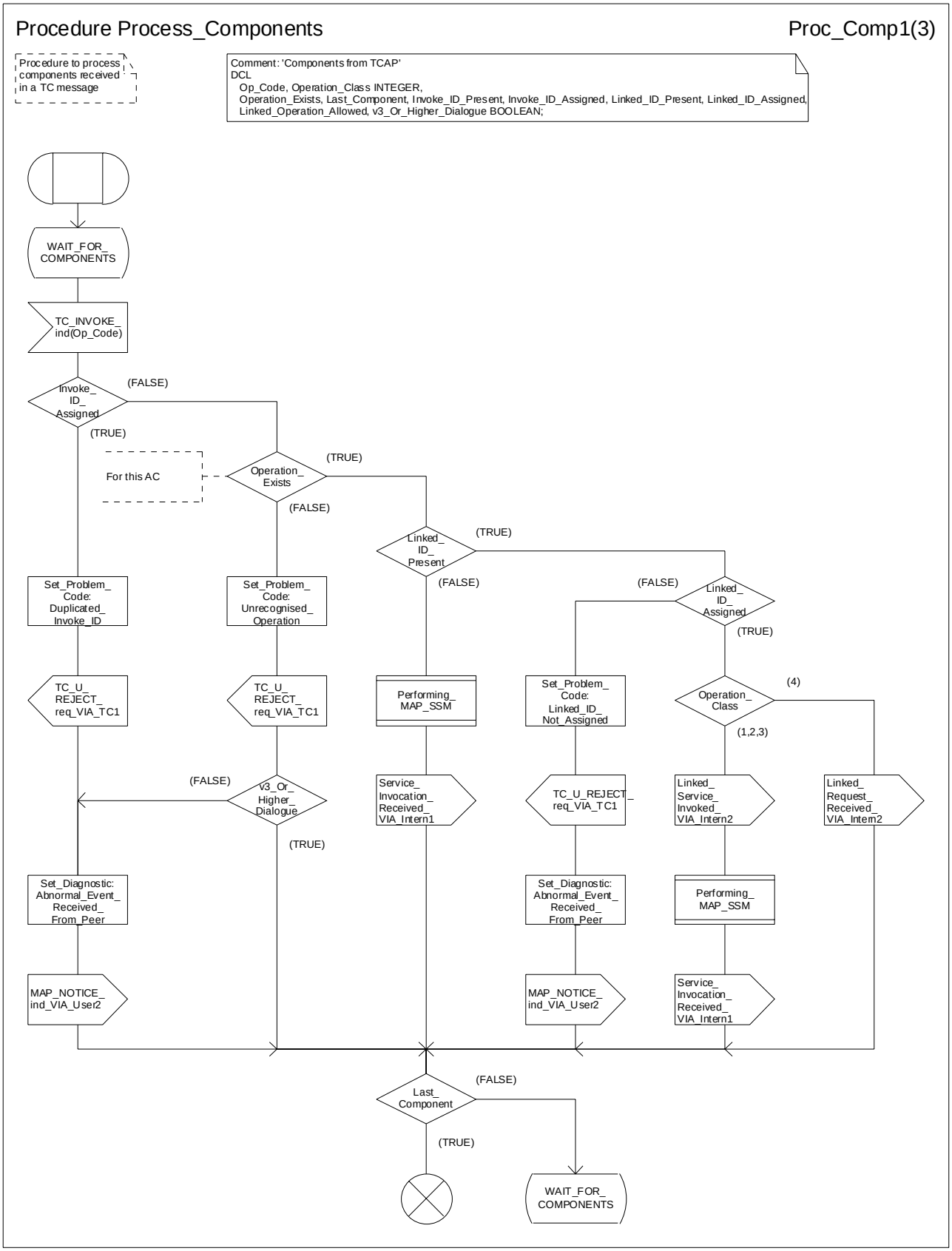


Figure 16.2/4 (sheet 1 of 3): Procedure PROCESS_COMPONENTS

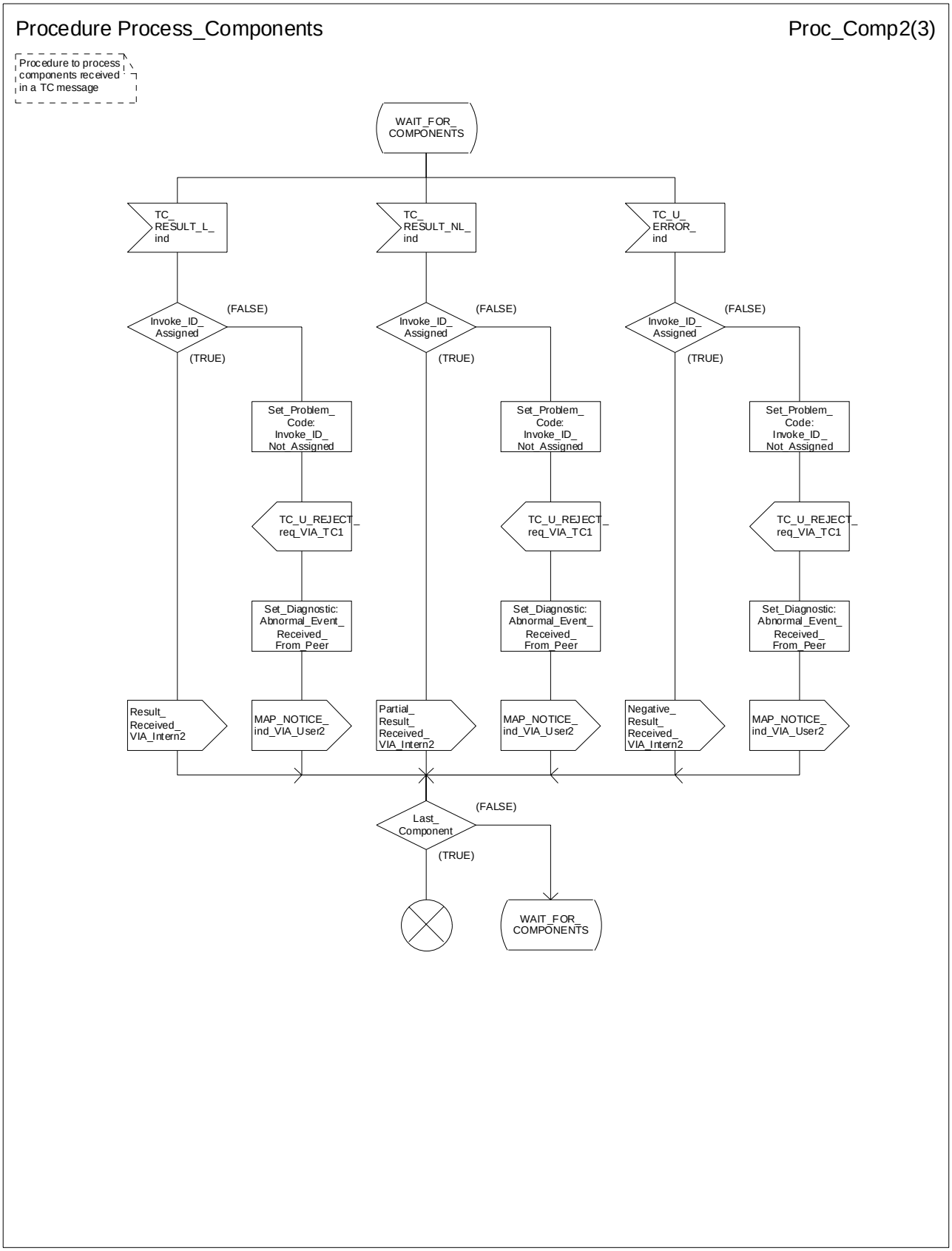


Figure 16.2/4 (sheet 2 of 3): Procedure PROCESS_COMPONENTS

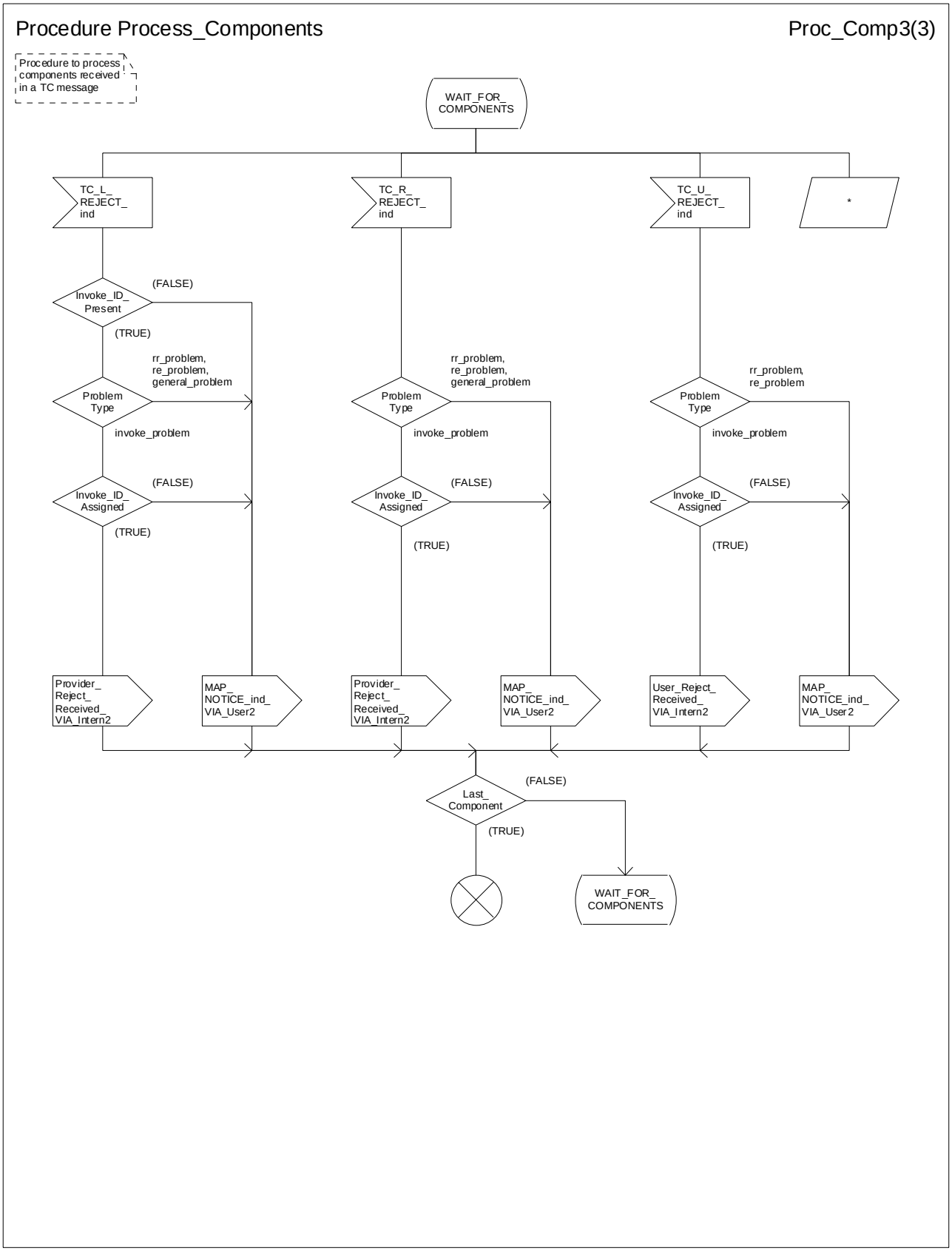


Figure 16.2/4 (sheet 3 of 3): Procedure PROCESS_COMPONENTS

Process LOAD_CTRL 16.2_5(1)

Figure 16.2/5:

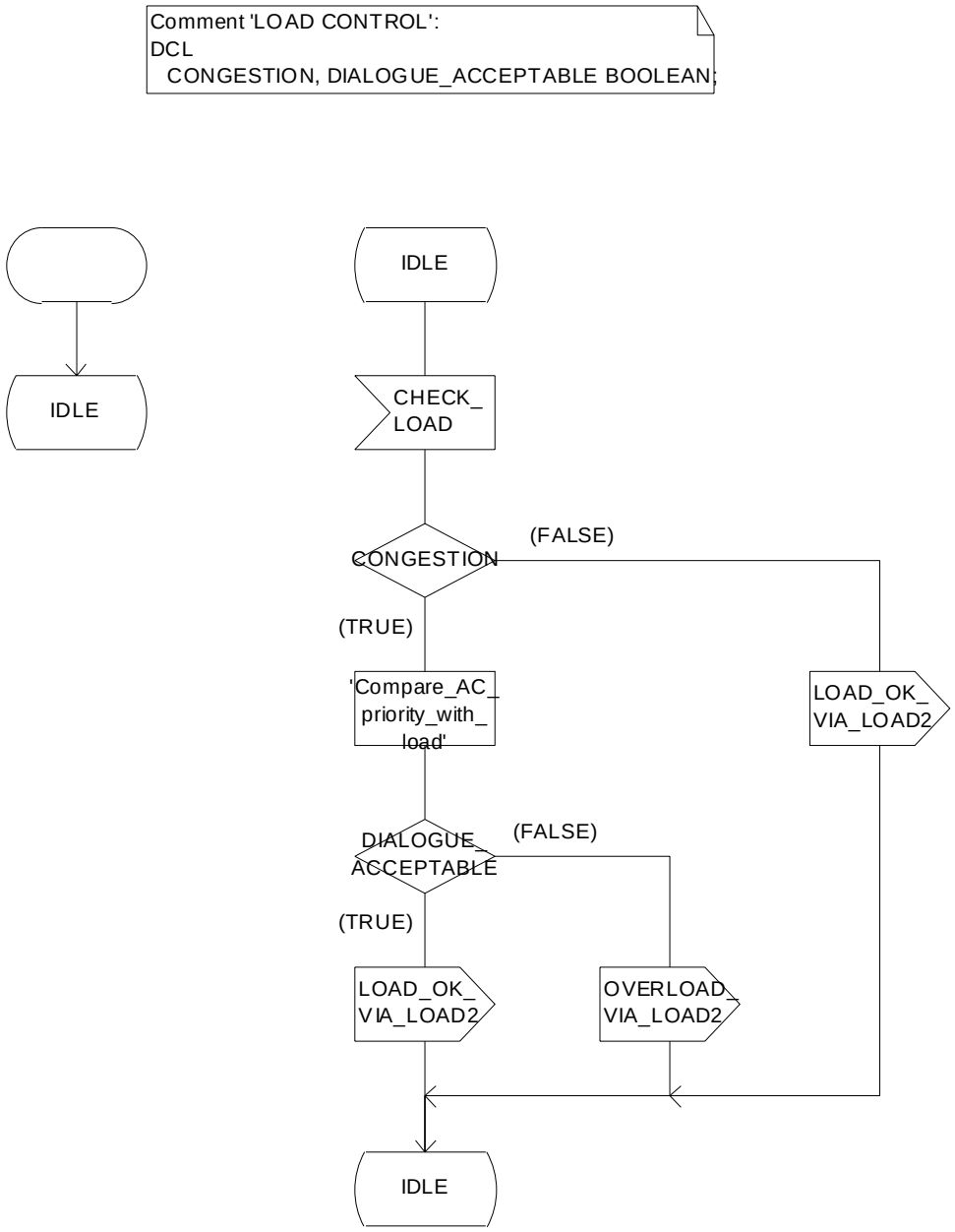


Figure 16.2/5: Process LOAD_CTRL

Process PERFORMING_MAP_SSM 16.2_6.1(3)

Figure 16.2/6:

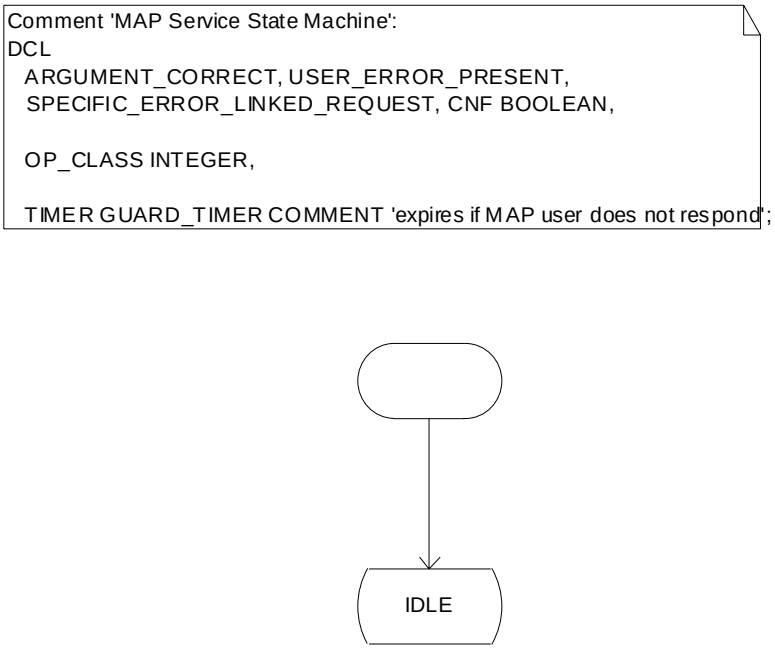


Figure 16.2/6 (sheet 1 of 3): Process PERFORMING_MAP_SSM

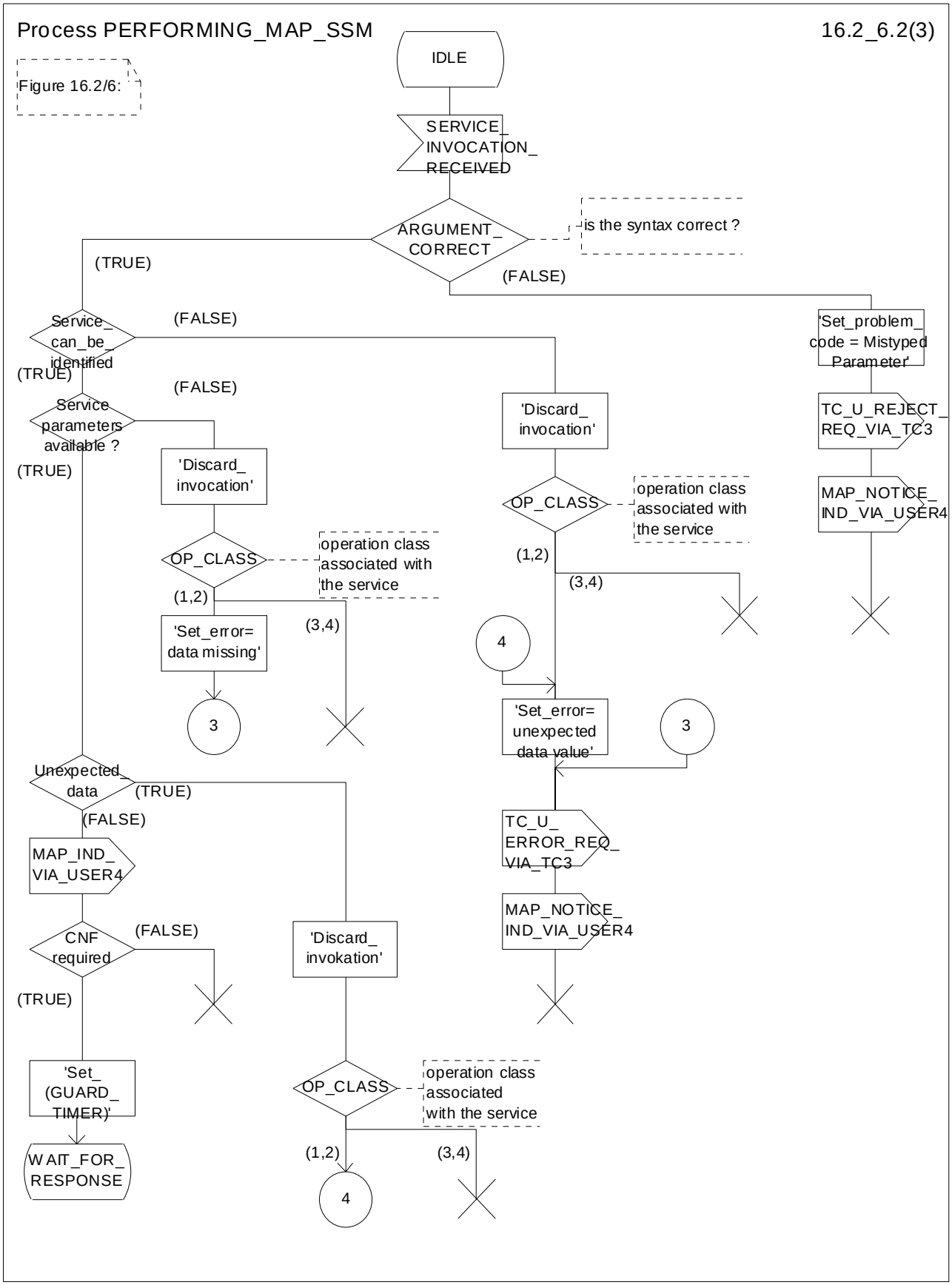


Figure 16.2/6 (sheet 2 of 3): Process PERFORMING_MAP_SSM

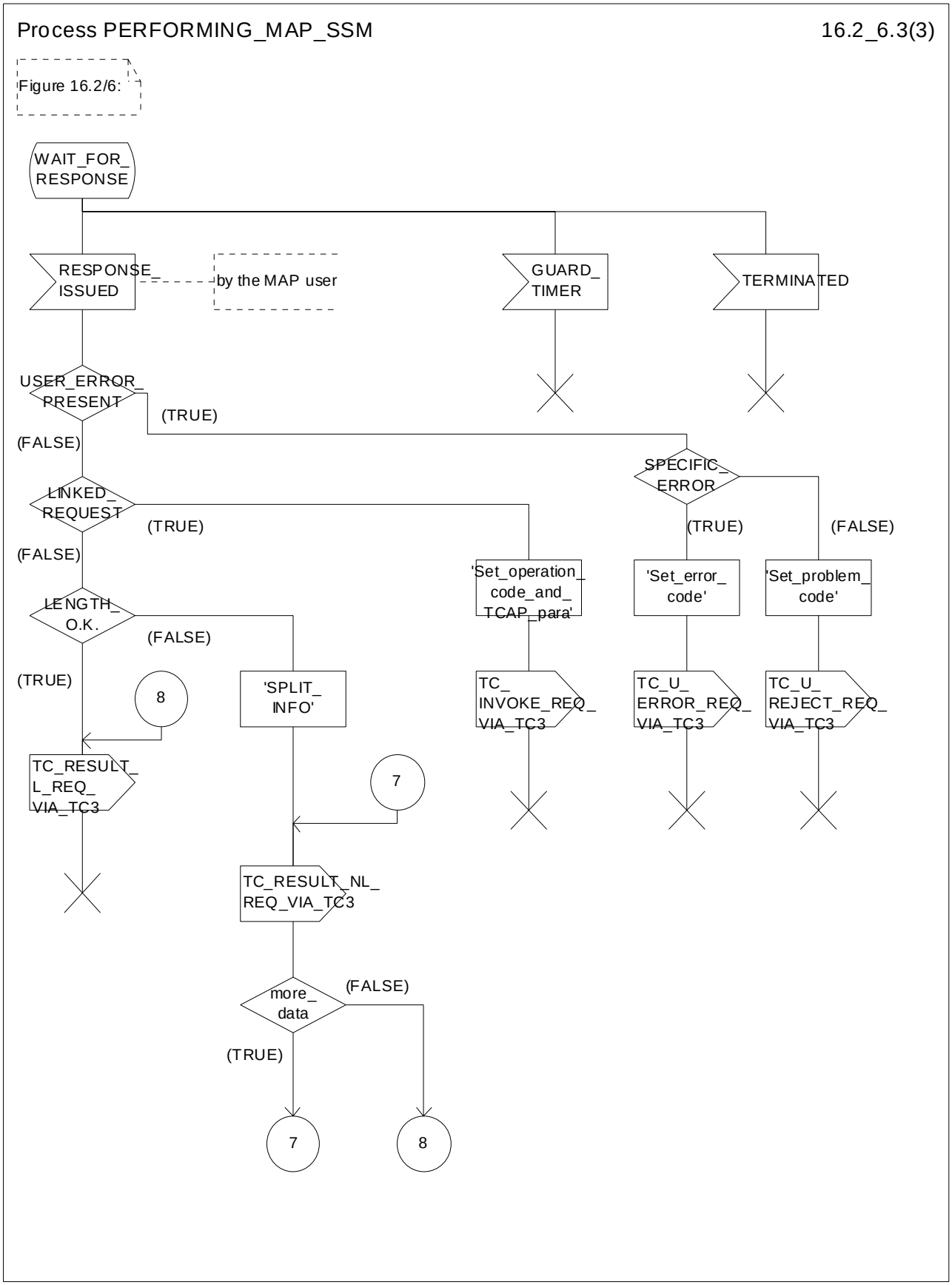


Figure 16.2/6 (sheet 3 of 3): Process PERFORMING_MAP_SSM

Process REQUESTING_MAP_SSM 16.2_7.1(4)

Figure 16.2/7:

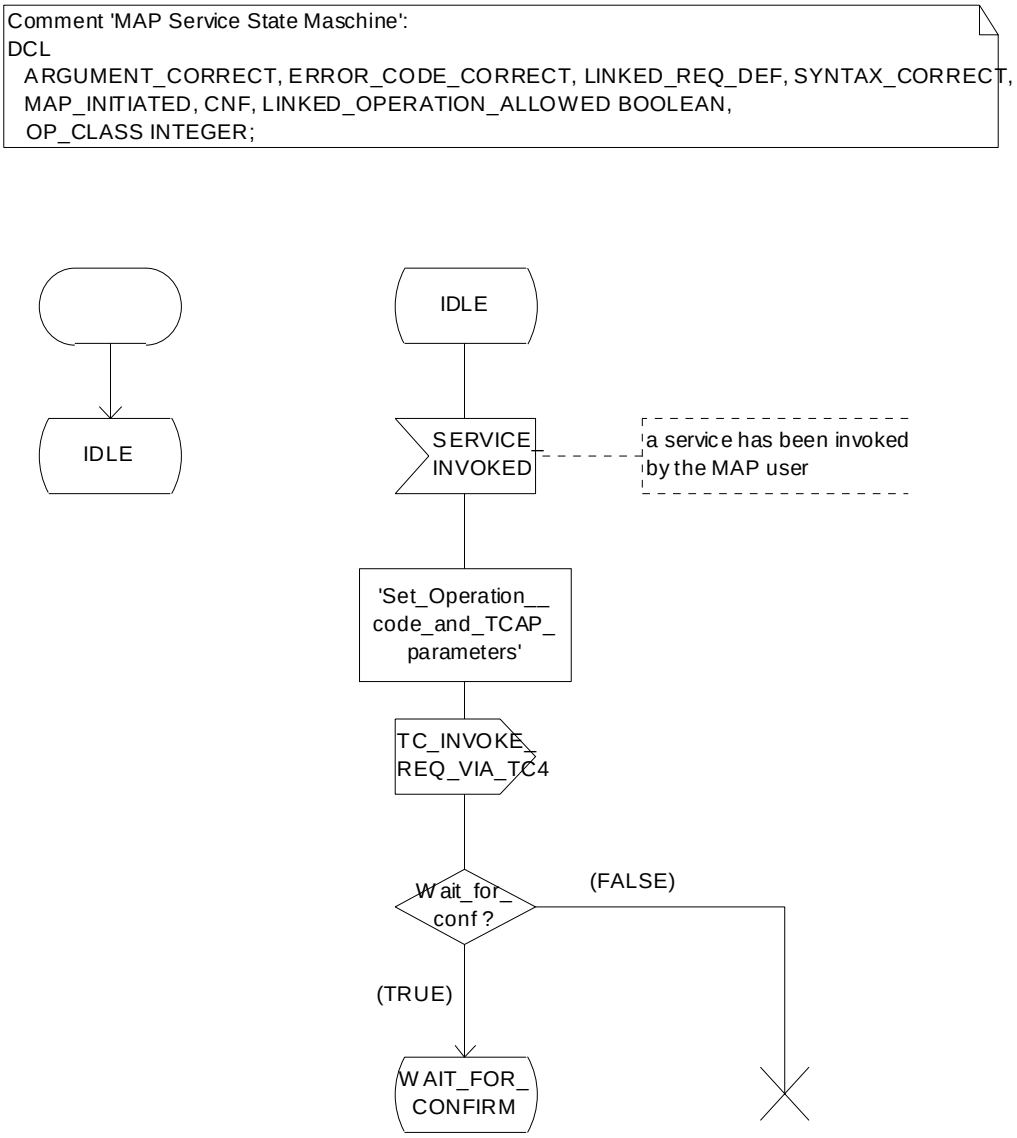


Figure 16.2/7 (sheet 1 of 4): Process REQUESTING_MAP_SSM

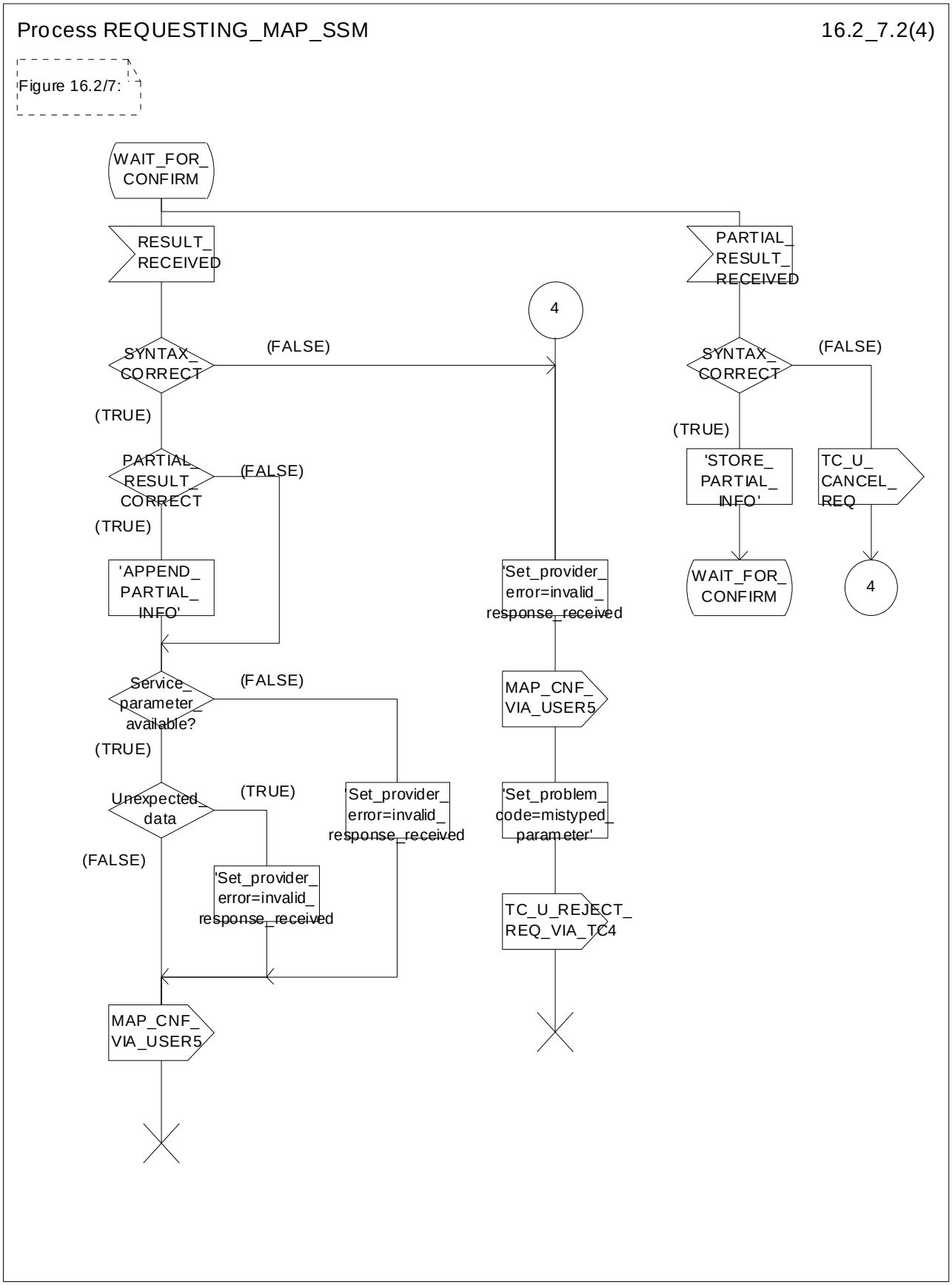


Figure 16.2/7 (sheet 2 of 4): Process REQUESTING_MAP_SSM

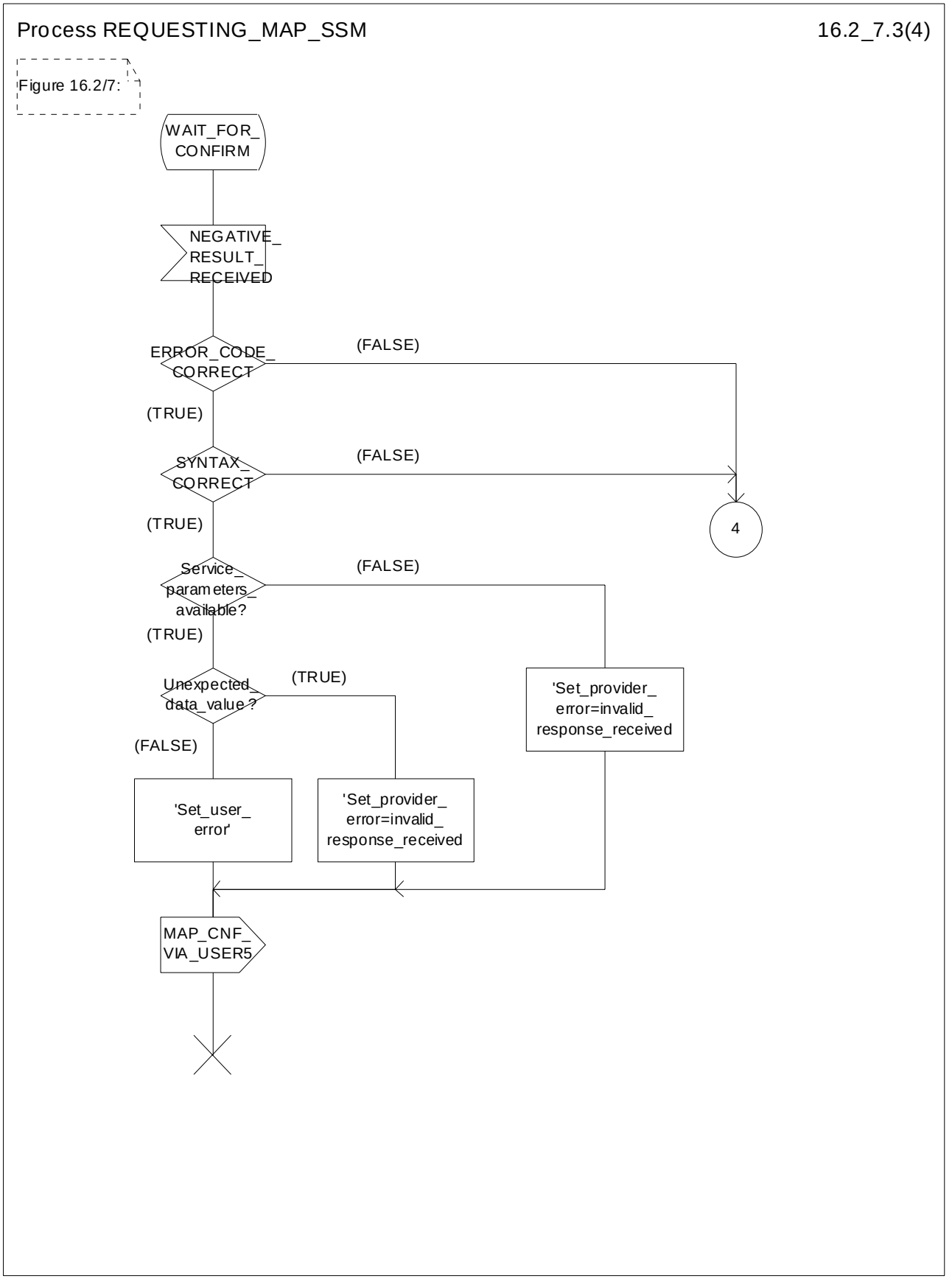


Figure 16.2/7 (sheet 3 of 4): Process REQUESTING_MAP_SSM

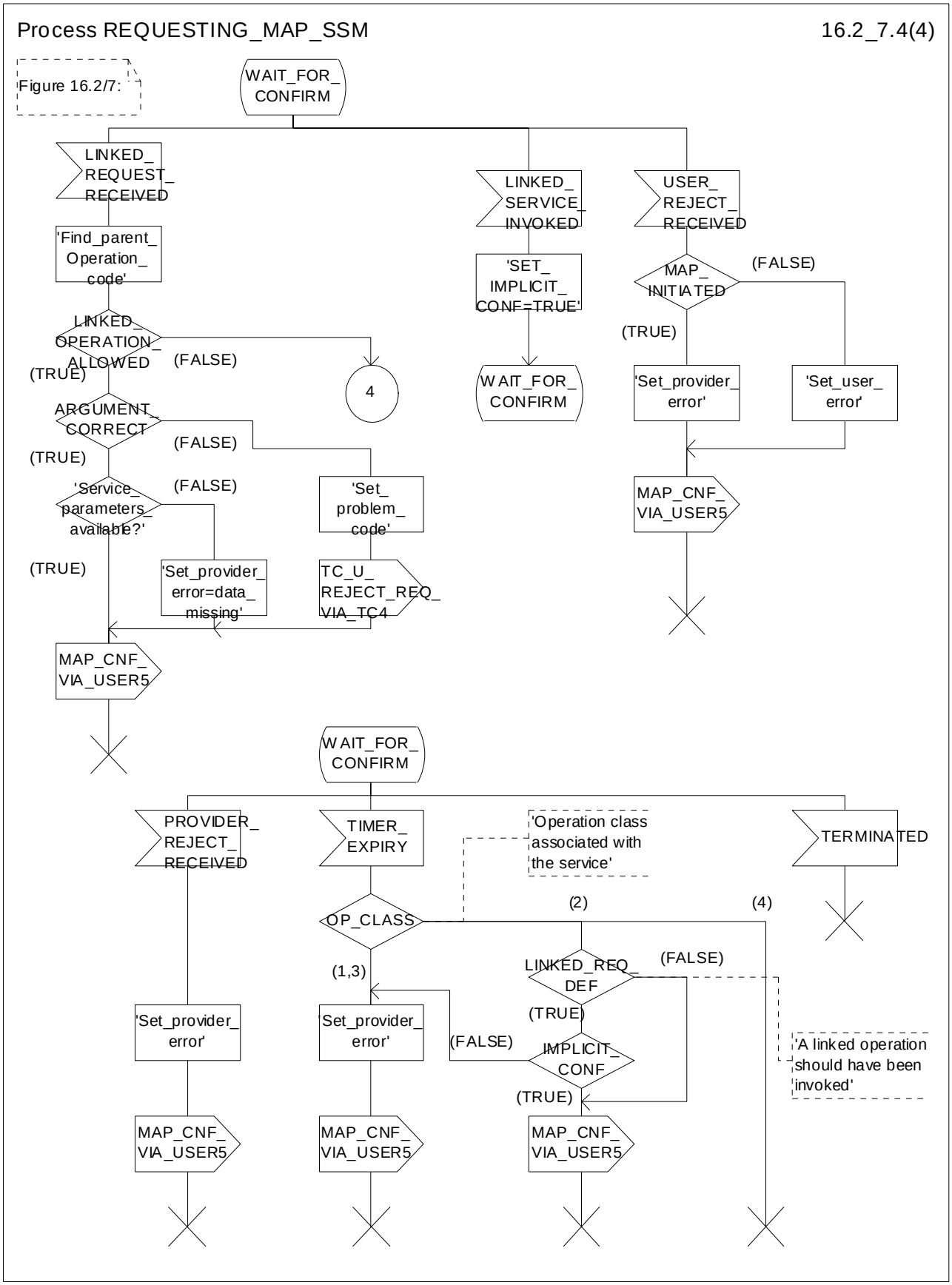


Figure 16.2/7 (sheet 4 of 4): Process REQUESTING_MAP_SSM

17 Abstract syntax of the MAP protocol

17.1 General

This clause specifies the Abstract Syntaxes for the Mobile Application Part as well as the associated set of Operations and Errors, using the Abstract Syntax Notation One (ASN.1), defined in CCITT Recommendation X.208 (1988) or X.680 (1994) with additions as defined in clause 17.1.4 on Compatibility Considerations and the OPERATION and ERROR external MACROs, defined in CCITT Recommendation Q.773.

The Abstract Syntax is defined for all interfaces specified in clause 4.4 except for the A- and B-interfaces.

The Mobile Application Part protocol is defined by two Abstract Syntaxes:

- one Abstract Syntax which encompass all Operations; and
- Errors identified by the various MAP subsystem numbers.

This Abstract Syntax represents the set of values each of which is a value of the ASN.1 type TCAPMessages. MessageType as defined in CCITT Recommendation Q.773 with the ANY DEFINED BY sections resolved by the operation and error codes included in the ASN.1 module MAP-Protocol. However, only the subset of this abstract syntax which is required by the procedures defined for an entity needs to be supported:

- one Abstract Syntax identified by the OBJECT IDENTIFIER value MAP-DialogueInformation.map-DialogueAS.

This Abstract Syntax represents the set of values each of which is a value of the ASN.1 type MAP-DialogueInformation.MAP-DialoguePDU. Such a value of the ASN.1 single-ASN.1-type element is contained within the user-information element of the TCAPMessages.DialoguePortion ASN.1 type. This Abstract Syntax name is to be used as a direct reference.

17.1.1 Encoding rules

The encoding rules which are applicable to the defined Abstract Syntaxes are the Basic Encoding Rules for Abstract Syntax Notation One, defined in CCITT Recommendation X.690 with the same exceptions as in CCITT Recommendation Q.773 clause 4 Message Representation.

When the definite form is used for length encoding, a data value of length less than 128 octets must have the length encoded in the short form.

When the long form is employed to code a length, the minimum number of octets shall be used to code the length field.

OCTET STRING values and BIT STRING values must be encoded in a primitive form.

There is no restriction to the use of empty constructors (e.g. an empty SEQUENCE type). That is, the encoding of the content of any data value shall consist of zero, one or more octets.

17.1.2 Use of TC

The mapping of OPERATION and ERROR to TC components is defined in ETS 300 287 (version 2) which is based on CCITT Recommendation Q.773 (1992).

NOTE 1: The class of an operation is not stated explicitly but is specified as well in the ASN.1 operation type definition.

Class 1: RESULT and ERROR appear in ASN.1 operation type definition.

Class 2: only ERROR appears in ASN.1 operation type definition.

Class 3: only RESULT appears in ASN.1 operation type definition.

Class 4: both RESULT and ERROR do not appear in ASN.1 operation type definition.

The ASN.1 data type which follows the keywords "ARGUMENT", "PARAMETER" or "RESULT" (for OPERATION and ERROR) is always optional from a syntactic point of view. However, except when specifically mentioned with the ASN.1 comment «-- optional» , the «parameter» part of a component has to be considered as mandatory from a semantic point of view.

When an optional element is missing in an invoke component or in an inner data structure while it is required by the context, an error component is returned if specified in the operation type; the associated type of error is DataMissing. This holds also when the entire parameter of an invoke component is missing while it is required by the context.

NOTE 2: When a mandatory element is missing in the parameter or inner data structure of any component, a reject component is returned (if the dialogue still exists). The problem code to be used is "Mistyped parameter".

The Timer Values used in the operation type definitions are indicated as ASN.1 comments. The Timer Value Ranges are:

s = from 3 seconds to 10 seconds;

m = from 15 seconds to 30 seconds;

ml = from 1 minute to 10 minutes;

l = from 28 hours to 38 hours.

17.1.2.1 Use of Global Operation and Error codes defined outside MAP

An entity supporting an application context greater than 2 shall be capable of receiving an operation or error code, within an application context defined in 3GPP TS 29.002, encoded as either an Object Identifier (as defined in CCITT Recommendation X.690 (1994)) or an integer value (as defined in clause 17.5). Related restrictions regarding the use of Object Identifiers are as follows:

- The length of the Object Identifier shall not exceed 16 octets and the number of components of the Object Identifier shall not exceed 16.
- Object Identifiers shall be used only for operations or errors defined outside of 3GPP TS 29.002.
- Global error codes may be sent only in response to a global operation. If a standard operation is received then a global error code shall not be sent in response.

Handling of an unknown operation codes by the receiving entity is defined in clause 15.1.1.

17.1.3 Use of information elements defined outside MAP

An information element or a set of information elements (messages) transparently carried in the Mobile Application Part but defined in other recommendations/technical specifications are handled in one of the following ways:

i) The contents of each information element (without the octets encoding the identifier and the length in the recommendation/technical specification where it is defined unless explicitly stated otherwise) is carried as the value of an ASN.1 NamedType derived from the OCTET STRING data type. Additionally, the internal structure may be explained by means of comments. In case of misalignment the referred to recommendation/technical specification takes precedence.

ii) The complete information element (including the octets encoding the identifier and the length in the recommendation/technical specification where it is defined) or set of information elements and the identity of the associated protocol are carried as the value of the ExternalSignalInfo data type defined in the present document. Where more than one information element is carried, the information elements are sent contiguously with no filler octets between them.

17.1.4 Compatibility considerations

The following ASN.1 modules conform to CCITT Recommendation X.208 (1988) or X.680 (1994) (the only module which makes use of X.680 is MAP-ExtensionDataTypes), but in addition Ellipsis Notation ("..." - notation) is used as described in ITU-T Recommendation X.680 Amendment 1 (1995) wherever future protocol extensions are foreseen.

The "..." construct applies only to SEQUENCE and ENUMERATED data types. An entity supporting a version greater than 1 shall not reject an unsupported extension following "..." of that SEQUENCE or ENUMERATED data type. The Encoding Rules from clause 17.1.1 apply to every element of the whole Transfer Syntax especially to the ASN.1 type EXTERNAL.

The extension container "privateExtensionList" is defined in this specification in order to carry extensions which are defined outside this specification. Private extensions can be defined by, for example, network operators, manufacturers, and regional standardisation bodies.

Private extensions shall:

1) if included in operations of an AC of V2, follow the extension marker and be tagged using PRIVATE tags up to and including 29.

NOTE: This type of extension is in most cases used only within a PLMN.

2) if included in operations of an AC of V3 or higher: be included only in the Private Extension Container that is defined in the specification.

NOTE: This type of extension can be used between PLMNs.

Private extensions shall not be included in v2 supplementary service operations.

Private extensions shall not be included within user error for RegisterCCEntry and EraseCCEntry operations.

PCS extensions shall be included in the PCS Extension Container that is defined in this specification.

In order to improve extensibility, a few error parameters have been defined as a CHOICE between the version 2 description and a SEQUENCE including the version 2 description and an extension container. Operations used in a v2-application-context must consider only the first alternative while operations used in a vn-application-context (n>2) must consider only the second alternative.

17.1.5 Structure of the Abstract Syntax of MAP

For each MAP parameter which has to be transferred by a MAP Protocol Data Unit (MAP message), there is a PDU field (an ASN.1 NamedType) whose ASN.1 identifier has the same name as the corresponding parameter, except for the differences required by the ASN.1 notation (blanks between words are removed or replaced by hyphen, the first letter of the first word is lower-case and the first letter of the following words are capitalised, e.g. "no reply condition time" is mapped to "noReplyConditionTime"). Additionally some words may be abbreviated as follows:

bs basic service

ch call handling

cug closed user group

ho handover

ic incoming call

id identity

info information

mm mobility management

lcs location services

ms mobile service

oc outgoing call

om operation & maintenance

pw Password

sm short message service

ss supplementary service

The MAP protocol is composed of several ASN.1 modules dealing with either operations, errors, data types, and, if applicable, split into those dealing with mobile services, call handling services, supplementary services and short message services. For operations and errors no values are assigned, but only the operation and error types in order to allow use of the defined types also by other protocols (e.g. 3GPP TS 24.080 [38]). The values (operation codes and error codes) are defined in a separate module. The ASN.1 source lines are preceded by line-numbers at the left margin in order to enable the usage of the cross-reference in annex A.

The module containing the definition of the operation packages for MAP is:

1. MAP-OperationPackages.

The module containing the definition of the application contexts for MAP is:

2. MAP-ApplicationContexts.

The module containing the data types for the Abstract Syntax to be used for TCAPMessages.DialoguePortion for MAP is:

3. MAP-DialogueInformation.

The module containing the operation codes and error codes for MAP is:

4. MAP-Protocol.

The modules containing all operation type definitions for MAP are:

5. MAP-MobileServiceOperations;
6. MAP-OperationAndMaintenanceOperations;
7. MAP-CallHandlingOperations;
8. MAP-SupplementaryServiceOperations;
9. MAP-ShortMessageServiceOperations;
10. MAP-Group-Call-Operations.
11. MAP-LocationServiceOperations

The module containing all error type definitions for MAP is:

12. MAP-Errors.

Modules containing all data type definitions for MAP are:

13. MAP-MS-DataTypes;
14. MAP-OM-DataTypes;
15. MAP-CH-DataTypes;
16. MAP-SS-DataTypes;
17. MAP-SS-Code;
18. MAP-SM-DataTypes;
19. MAP-ER-DataTypes;
20. MAP-CommonDataTypes;

21. MAP-TS-Code;

22. MAP-BS-Code;

23. MAP-ExtensionDataTypes;

24. MAP-GR-DataTypes;

25. MAP-LCS-DataTypes.

References are made also to modules defined outside of the present document. They are defined in the technical specification Mobile Services Domain and technical specification Transaction Capability respectively:

MobileDomainDefinitions;

TCAPMessages;

DialoguePDUs.

17.1.6 Application Contexts

The following informative table lists the latest versions of the Application Contexts used in this specification, with the operations used by them and, where applicable, whether or not the operation description is exactly the same as for previous versions. Information in 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	v3	prepareHandover forwardAccessSignalling sendEndSignal processAccessSignalling prepareSubsequentHandover	the syntax of this operation has been extended in comparison with release 98 version
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 activateTraceMode	the syntax is the same in v1 & v2
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	
sIWFSAllocationContext	v3	provideSIWFSNumber sIWFSSignallingModify	
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	
authenticationFailureReportContext	v3	authenticationFailureReport	

NOTE (*): The syntax of the operations is not the same as in previous versions unless explicitly stated

17.2 Operation packages

17.2.1 General aspects

This clause describes the operation-packages which are used to build the application-contexts defined in clause 17.3.

Each operation-package is a specification of the roles of a pair of communicating objects (i.e. a pair of MAP-Providers), in terms of operations which they can invoke of each other.

The grouping of operations into one or several packages does not necessarily imply any grouping in terms of Application Service Elements.

The following ASN.1 MACRO is used to describe operation-packages in this clause:

```
OPERATION-PACKAGE MACRO ::=

BEGIN

TYPE NOTATION ::= Symmetric | ConsumerInvokes SupplierInvokes |
empty

VALUE NOTATION ::= value(VALUE OBJECT IDENTIFIER)
Symmetric ::= "OPERATIONS" "{" OperationList "}"
ConsumerInvokes ::= "CONSUMER INVOKES" "{" OperationList "}"
SupplierInvokes ::= "SUPPLIER INVOKES" "{" OperationList "}" | empty
OperationList ::= Operation | OperationList "," Operation
Operation ::= value(OPERATION)

END
```

Since the application-context definitions provided in clause 17.3 use only an informal description technique, only the type notation is used in the following clauses to define operation-packages.

The following definitions are used throughout this clause (n>=2):

- v1-only operation: An operation which shall be used only in v1 application-contexts;
- vn-only operation: An operation which shall be used only in vn application-contexts;
- v(n-1)-operation: An operation whose specification has not been modified since the MAP v(n-1) specifications or if the modifications are considered as not affecting v(n-1) implementations;
- v(n-1)-equivalent operation: The version of an operation which excludes all the information elements and errors which have been added since the MAP v(n-1) specification;
- vn-only package: An operation package which contains only vn-only operations;
- v(n-1)-package: An operation package which contains only v(n-1)- operations.

The names of vn-packages are suffixed by "-vn" where n>=2.

For each operation package which is not vn-only (n>=2) and which does not include only v(n-1)-operations, there is a v(n-1)-equivalent package. Except when a definition is explicitly provided in the following clauses, the v(n-1)-equivalent package includes the v(n-1)-equivalent operations of the operations which belong to this package.

17.2.2 Packages specifications

17.2.2.1 Location updating

This operation package includes the operations required for location management procedures between HLR and VLR.

```
LocationUpdatingPackage-v3 ::= OPERATION-PACKAGE
     -- Supplier is HLR if Consumer is VLR
     CONSUMER INVOKES {
          updateLocation}
     SUPPLIER INVOKES {
          forwardCheckSs-Indication}
```

The v1-equivalent and v2-equivalent packages can be determined according to the rules described in clause 17.2.1.

17.2.2.2 Location cancellation

This operation package includes the operations required for location cancellation and MS purging procedures between HLR and VLR and between HLR and SGSN.

```
LocationCancellationPackage-v3 ::= OPERATION-PACKAGE
     -- Supplier is VLR or SGSN if Consumer is HLR
     CONSUMER INVOKES {
          cancelLocation}
```

The v1-equivalent and v2-equivalent packages can be determined according to the rules described in clause 17.2.1.

17.2.2.3 Roaming number enquiry

This operation package includes the operations required for roaming number enquiry procedures between HLR and VLR.

```
RoamingNumberEnquiryPackage-v3 ::= OPERATION-PACKAGE
     -- Supplier is VLR if Consumer is HLR
     CONSUMER INVOKES {
          provideRoamingNumber}
```

The v1-equivalent and v2-equivalent packages can be determined according to the rules described in clause 17.2.1.

17.2.2.4 Information retrieval

This operation package includes the operation required for the authentication information retrieval procedure between HLR and VLR and between HLR and SGSN.

```
InfoRetrievalPackage-v3 ::= OPERATION-PACKAGE
     -- Supplier is HLR if Consumer is VLR
     -- Supplier is HLR if Consumer is SGSN
     CONSUMER INVOKES {
          sendAuthenticationInfo}
```

The v2-equivalent package is defined as follows:

```
InfoRetrievalPackage-v2 ::= OPERATION-PACKAGE
     -- Supplier is HLR if Consumer is VLR
     -- Supplier is HLR if Consumer is SGSN
     CONSUMER INVOKES {
          sendAuthenticationInfo}
```

The v1-equivalent package is defined as follows:

```
InfoRetrievalPackage-v1 ::= OPERATION-PACKAGE
     -- Supplier is HLR or VLR if Consumer is VLR
     -- Supplier is HLR if Consumer is SGSN
     CONSUMER INVOKES {
          sendParameters}
```

17.2.2.5 Inter-VLR information retrieval

This operation package includes the operations required for inter VLR information retrieval procedures.

```
InterVlrInfoRetrievalPackage-v3 ::= OPERATION-PACKAGE
     -- Supplier is VLR if Consumer is VLR
     CONSUMER INVOKES {
          sendIdentification}
```

The v2-equivalent package is defined as follows:

```
InterVlrInfoRetrievalPackage-v2 ::= OPERATION-PACKAGE
     -- Supplier is VLR if Consumer is VLR
     CONSUMER INVOKES {
          sendIdentification}
```

The v1-equivalent package is : InfoRetrievalPackage-v1.

17.2.2.6 IMSI retrieval

This operation package includes the operation required for the IMSI retrieval procedure between HLR and VLR.

```
IMSIRetrievalPackage-v2 ::= OPERATION-PACKAGE
     -- Supplier is HLR if Consumer is VLR
     CONSUMER INVOKES {
          sendIMSI}
```

This package is v2 only.

17.2.2.7 Call control transfer

This operation package includes the operation required for the call control transfer procedure between VMSC and GMSC.

```
CallControlTransferPackage-v4 ::= OPERATION-PACKAGE
     -- Supplier is GMSC if Consumer is VMSC
     CONSUMER INVOKES {
          resumeCallHandling}
```

The v3-equivalent package can be determined according to the rules described in clause 17.2.1.

17.2.2.8 - 17.2.2.9 Void

17.2.2.10 Interrogation

This operation package includes the operations required for interrogation procedures between MSC and HLR or NPLR.

```
InterrogationPackage-v3 ::= OPERATION-PACKAGE
     -- Supplier is HLR or NPLR if Consumer is MSC
     CONSUMER INVOKES {
          sendRoutingInfo}
```

The v1-equivalent and v2-equivalent packages can be determined according to the rules described in clause 17.2.1.

17.2.2.11 Void

17.2.2.12 Handover Control

This operation package includes the operations required for handover procedures between MSCs.

```
HandoverControlPackage-v3 ::= OPERATION-PACKAGE
     -- Supplier is MSCB if Consumer is MSCA
     CONSUMER INVOKES {
          prepareHandover,
          forwardAccessSignalling}
     SUPPLIER INVOKES {
          sendEndSignal,
          processAccessSignalling,
          prepareSubsequentHandover}
```

The v2-equivalent package can be determined according to the rules described in clause 17.2.1.

The v1-equivalent package is defined as follows.

```
HandoverControlPackage-v1 ::= OPERATION-PACKAGE
     -- Supplier is MSCB if Consumer is MSCA
     CONSUMER INVOKES {
          performHandover,
          forwardAccessSignalling,
          traceSubscriberActivity}
     SUPPLIER INVOKES {
          sendEndSignal,
          noteInternalHandover,
          processAccessSignalling,
          performSubsequentHandover}
```

17.2.2.13 Subscriber Data management stand alone

This operation package includes the operations required for stand alone subscriber data management procedures between HLR and VLR or between HLR and SGSN.

```
SubscriberDataMngtStandAlonePackage-v3 ::= OPERATION-PACKAGE
     -- Supplier is VLR or SGSN if Consumer is HLR
     CONSUMER INVOKES {
          insertSubscriberData,
          deleteSubscriberData}
```

The v1-equivalent and v2-equivalent packages can be determined according to the rules described in clause 17.2.1.

17.2.2.14 Equipment management

This operation package includes the operations required for equipment management procedures between EIR and MSC or between EIR and SGSN.

```
EquipmentMngtPackage-v2 ::= OPERATION-PACKAGE
     -- Supplier is EIR if Consumer is MSC
     -- Supplier is EIR if Consumer is SGSN
     CONSUMER INVOKES {
          checkIMEI}
```

The v1-equivalent package can be determined according to the rules described in clause 17.2.1.

17.2.2.15 Subscriber data management

This operation package includes the operations required for subscriber data management procedures between HLR and VLR or between HLR and SGSN.

```
SubscriberDataMngtPackage-v3 ::= OPERATION-PACKAGE
     -- Supplier is VLR or SGSN if Consumer is HLR
     CONSUMER INVOKES {
          insertSubscriberData}
```

The v1-equivalent and v2-equivalent packages can be determined according to the rules described in clause 17.2.1.

17.2.2.16 Location register restart

This operation package includes the operations required for location register restart procedures between HLR and VLR or between HLR and SGSN.

```
ResetPackage-v2 ::= OPERATION-PACKAGE
     -- Supplier is VLR or SGSN if Consumer is HLR
     CONSUMER INVOKES {
          reset}
```

The v1-equivalent package can be determined according to the rules described in clause 17.2.1.

17.2.2.17 Tracing stand-alone

This operation package includes the operations required for stand alone tracing procedures between HLR and VLR or between HLR and SGSN.

```
TracingStandAlonePackage-v3 ::= OPERATION-PACKAGE
     -- Supplier is VLR or SGSN if Consumer is HLR
     CONSUMER INVOKES {
          activateTraceMode,
          deactivateTraceMode}
```

The v1-equivalent and v2-equivalent packages can be determined according to the rules described in clause 17.2.1.

17.2.2.18 Functional SS handling

This operation package includes the operations required for functional supplementary services procedures between VLR and HLR.

```
FunctionalSsPackage-v2 ::= OPERATION-PACKAGE
     -- Supplier is HLR if Consumer is VLR
     CONSUMER INVOKES {
          registerSS,
          eraseSS,
          activateSS,
          deactivateSS,
          registerPassword,
          interrogateSS}
     SUPPLIER INVOKES {
          getPassword}
```

The v1-equivalent package can be determined according to the rules described in clause 17.2.1.

17.2.2.19 Tracing

This operation package includes the operations required for tracing procedures between HLR and VLR or between HLR and SGSN.

```
TracingPackage-v3 ::= OPERATION-PACKAGE
     -- Supplier is VLR or SGSN if Consumer is HLR
     CONSUMER INVOKES {
          activateTraceMode}
```

The v1-equivalent and v2-equivalent packages can be determined according to the rules described in clause 17.2.1.

17.2.2.20 Binding

This operation package includes the operation required to initialise a supplementary service procedure between VLR and HLR or between gsmSCF and HLR.

```
BindingPackage-v1 ::= OPERATION-PACKAGE
     -- Supplier is HLR if Consumer is VLR
     -- Supplier is gsmSCF if Consumer is HLR
     CONSUMER INVOKES {
          beginSubscriberActivity}
```

This package is v1 only.

17.2.2.21 Unstructured SS handling

This operation package includes the operations required for unstructured supplementary services procedures between VLR and HLR, between the HLR and the gsmSCF, and between HLR and HLR.

```
UnstructuredSsPackage-v2 ::= OPERATION-PACKAGE
     -- Supplier is HLR if Consumer is VLR
     -- Supplier is gsmSCF or HLR if Consumer is HLR
     CONSUMER INVOKES {
          processUnstructuredSS-Request}
     SUPPLIER INVOKES {
          unstructuredSS-Request,
          unstructuredSS-Notify}
```

The v1-equivalent package is defined as follows:

```
UnstructuredSsPackage-v1 ::= OPERATION-PACKAGE
     -- Supplier is HLR if Consumer is VLR
     -- Supplier is gsmSCF if Consumer is HLR
     CONSUMER INVOKES {
          processUnstructuredSS-Data}
```

17.2.2.22 MO Short message relay services

This operation package includes the operations required for short message relay service procedures between IWMSC and VMSC or between GMSC and MSC or between SGSN and IWMSC.

```
MOShortMsgRelayPackage-v3 ::= OPERATION-PACKAGE
     -- Supplier is IWMSC if Consumer is MSC
     -- Supplier is IWMSC if Consumer is SGSN
     CONSUMER INVOKES {
          MO-forwardSM}
```

The v2-equivalent package is defined as follows:

```
ShortMsgRelayPackage-v2 ::= OPERATION-PACKAGE
     -- Supplier is IWMSC if Consumer is MSC
     -- Supplier is MSC or SGSN if Consumer is GMSC
     -- Supplier is IWMSC if Consumer is SGSN
     CONSUMER INVOKES {
          forwardSM}
```

The v1-equivalent package can be determined according to the rules described in clause 17.2.1.

17.2.2.23 Short message gateway services

This operation package includes the operations required for short message service gateway procedures between MSC and HLR.

```
ShortMsgGatewayPackage-v3 ::= OPERATION-PACKAGE
     -- Supplier is HLR if Consumer is GMSC
     CONSUMER INVOKES {
          sendRoutingInfoForSM,
          reportSM-DeliveryStatus}
     SUPPLIER INVOKES {
          informServiceCentre}
```

The v2-equivalent package can be determined according to the rules described in clause 17.2.1.

The v1-equivalent package is defined as follows:

```
ShortMsgGatewayPackage-v1 ::= OPERATION-PACKAGE
     -- Supplier is HLR if Consumer is GMSC
     CONSUMER INVOKES {
          sendRoutingInfoForSM,
          reportSMDeliveryStatus}
```

17.2.2.24 MT Short message relay services

This operation package includes the operations required for short message relay service procedures between GMSC and MSC or between GMSC and SGSN.

```
MTShortMsgRelayPackage-v3 ::= OPERATION-PACKAGE
     -- Supplier is MSC or SGSN if Consumer is GMSC
     CONSUMER INVOKES {
          MT-forwardSM}
```

The v2-equivalent package is: **ShortMsgRelayPackage-v2**

17.2.2.25 Void

17.2.2.26 Message waiting data management

This operation package includes the operations required for short message waiting data procedures between HLR and VLR, between HLR and SGSN.

```
MwdMngtPackage-v3 ::= OPERATION-PACKAGE
     -- Supplier is HLR if Consumer is SGSN
     -- Supplier is HLR if Consumer is VLR
     CONSUMER INVOKES {
          readyForSM}
```

The v2-equivalent package can be determined according to the rules described in clause 17.2.1.

The v1-equivalent package is defined as follows:

```
MwdMngtPackage-v1 ::= OPERATION-PACKAGE
     -- Supplier is HLR if Consumer is VLR
     CONSUMER INVOKES {
          noteSubscriberPresent}
```

17.2.2.27 Alerting

This operation package includes the operations required for alerting between HLR and IWMSC.

```
AlertingPackage-v2 ::= OPERATION-PACKAGE
     -- Supplier is IWMSC if Consumer is HLR
     CONSUMER INVOKES {
          alertServiceCentre}
```

The v1-equivalent package is defined as follows.

```
AlertingPackage-v1 ::= OPERATION-PACKAGE
     -- Supplier is IWMSC if Consumer is HLR
     CONSUMER INVOKES {
          alertServiceCentreWithoutResult}
```

17.2.2.28 Data restoration

This operation package includes the operations required for VLR data restoration between HLR and VLR.

```
DataRestorationPackage-v3 ::= OPERATION-PACKAGE
     -- Supplier is HLR if Consumer is VLR
     CONSUMER INVOKES {
          restoreData}
```

The v2-equivalent package can be determined according to the rules described in clause 17.2.1.

The v1-equivalent package is: InfoRetrievalPackage-v1

17.2.2.29 Purging

This operation package includes the operations required for purging between HLR and VLR or between HLR and SGSN.

```
PurgingPackage-v3 ::= OPERATION-PACKAGE
     -- Supplier is HLR if Consumer is VLR
     -- Supplier is HLR if Consumer is SGSN
     CONSUMER INVOKES {
          purgeMS}
```

The v2-equivalent package can be determined according to the rules described in clause 17.2.1.

17.2.2.30 Subscriber information enquiry

This operation package includes the operations required for subscriber information enquiry procedures between HLR and VLR.

```
SubscriberInformationEnquiryPackage-v3 ::= OPERATION-PACKAGE
     -- Supplier is VLR if Consumer is HLR
     CONSUMER INVOKES {
          provideSubscriberInfo}
```

This package is v3 only.

17.2.2.31 Any time information enquiry

This operation package includes the operations required for any time information enquiry procedures between gsmSCF and HLR or GMLC.

```
AnyTimeInformationEnquiryPackage-v3 ::= OPERATION-PACKAGE
     -- Supplier is HLR or GMLC if Consumer is gsmSCF
     CONSUMER INVOKES {
          anyTimeInterrogation}
```

This package is v3 only.

17.2.2.32 Group Call Control

This operation package includes the operations required for group call and broadcast call procedures between MSCs.

```
GroupCallControlPackage-v3 ::= OPERATION-PACKAGE
     -- Supplier is relay MSC if Consumer is anchor MSC
     CONSUMER INVOKES {
          prepareGroupCall,
          forwardGroupCallSignalling}
     SUPPLIER INVOKES {
          sendGroupCallEndSignal,
          processGroupCallSignalling}
```

This package is v3 only.

17.2.2.33 Provide SIWFS number

This operation package includes the operations required between VMSC and SIWF for requesting resources from an SIWF.

```
ProvideSIWFSNumberPackage-v3 ::= OPERATION-PACKAGE
     -- Supplier is SIWF if Consumer is VMSC
     CONSUMER INVOKES {
          provideSIWFSNumber}
```

This package is v3 only.

17.2.2.34 SIWFS Signalling Modify

This operation package includes the operations required for the modification of the resources in an SIWF between the VMSC and SIWF.

```
SIWFSSignallingModifyPackage-v3 ::= OPERATION-PACKAGE
     -- Supplier is SIWF if Consumer is VMSC
     CONSUMER INVOKES {
          sIWFSSignallingModify}
```

This package is v3 only.

17.2.2.35 Gprs location updating

This operation package includes the operations required for the gprs location management procedures between HLR and SGSN.

```
GprsLocationUpdatingPackage-v3 ::= OPERATION-PACKAGE
     -- Supplier is HLR if Consumer is SGSN
     CONSUMER INVOKES {
          updateGprsLocation}
```

This package is v3 only.

17.2.2.36 Gprs Interrogation

This operation package includes the operations required for interrogation procedures between HLR and GGSN.

```
GprsInterrogationPackage-v3 ::= OPERATION-PACKAGE
     -- Supplier is HLR if Consumer is GGSN
     CONSUMER INVOKES {
          sendRoutingInfoForGprs}
```

This package is v3 only.

17.2.2.37 Failure reporting

This operation package includes the operations required for failure reporting between HLR and GGSN.

```
FailureReportingPackage-v3 ::= OPERATION-PACKAGE
     -- Supplier is HLR if Consumer is GGSN
     CONSUMER INVOKES {
          failureReport}
```

This package is v3 only.

17.2.2.38 GPRS notifying

This operation package includes the operations required for notifying that GPRS subscriber is present between HLR and GGSN.

```
GprsNotifyingPackage-v3 ::= OPERATION-PACKAGE
     -- Supplier is GGSN if Consumer is HLR
     CONSUMER INVOKES {
          noteMsPresentForGprs}
```

This package is v3 only.

17.2.2.39 Supplementary Service invocation notification

This operation package includes the operations required for Supplementary Service invocation notification procedures between the MSC and the gsmSCF and between the HLR and the gsmSCF.

```
SS-InvocationNotificationPackage-v3 ::= OPERATION-PACKAGE
     -- Supplier is gsmSCF if Consumer is MSC
     -- Supplier is gsmSCF if Consumer is HLR
     CONSUMER INVOKES {
          ss-InvocationNotification}
```

This package is v3 only.

17.2.2.40 Set Reporting State

This operation package includes the operation required for procedures between HLR and VLR to set the reporting state.

```
SetReportingStatePackage-v3 ::= OPERATION-PACKAGE
     -- Supplier is VLR if Consumer is HLR
     CONSUMER INVOKES {
          setReportingState}
```

This package is v3 only.

17.2.2.41 Status Report

This operation package includes the operation required for procedures between VLR and HLR to report call results and events.

```
StatusReportPackage-v3 ::= OPERATION-PACKAGE
     -- Supplier is HLR if Consumer is VLR
     CONSUMER INVOKES {
          statusReport}
```

This package is v3 only.

17.2.2.42 Remote User Free

This operation package includes the operation required by the HLR to indicate to the VLR that the remote user is free.

```
RemoteUserFreePackage-v3 ::= OPERATION-PACKAGE
     -- Supplier is VLR if Consumer is HLR
     CONSUMER INVOKES {
          remoteUserFree}
```

This package is v3 only.

17.2.2.43 Call Completion

This operation package includes the operations required for procedures between VLR and HLR for subscriber control of call completion services.

```
CallCompletionPackage-v3 ::= OPERATION-PACKAGE
     -- Supplier is HLR if Consumer is VLR
     CONSUMER INVOKES {
          registerCC-Entry,
          eraseCC-Entry}
```

This package is v3 only.

17.2.2.44 Location service gateway services

This operation package includes the operations required for location service gateway procedures between GMLC and HLR.

```
LocationSvcGatewayPackage-v3 ::= OPERATION-PACKAGE
     -- Supplier is HLR if Consumer is GMLC
     CONSUMER INVOKES {
          sendRoutingInfoForLCS}
```

This package is v3 only.

17.2.2.45 Location service enquiry

This operation package includes the operations required for the location service enquiry procedures between GMLC and MSC.

```
LocationSvcEnquiryPackage-v3 ::= OPERATION-PACKAGE
     -- Supplier is MSC if Consumer is GMLC
     CONSUMER INVOKES {
          provideSubscriberLocation}
```

This package is v3 only.

17.2.2.45A Location service reporting

This operation package includes the operations required for the location service enquiry procedures between MSC and GMLC.

```
LocationSvcReportingPackage-v3 ::= OPERATION-PACKAGE
     -- Supplier is GMLC if Consumer is MSC
     CONSUMER INVOKES {
          subscriberLocationReport}
```

17.2.2.46 Void

17.2.2.47 Void

17.2.2.48 Void

17.2.2.49 IST Alerting

This operation package includes the operation required for alerting procedures between the MSC (Visited MSC or Gateway MSC) and HLR.

```
IST-AlertingPackage-v3 ::= OPERATION-PACKAGE
     -- Supplier is HLR if Consumer is VMSC
     -- Supplier is HLR if Consumer is GMSC
     CONSUMER INVOKES {
          istAlert}
```

This package is v3 only.

17.2.2.50 Service Termination

This operation package includes the operation required for immediate service termination procedures between the HLR and the Visited MSC or between the HLR and the Gateway MSC.

```
ServiceTerminationPackage-v3 ::= OPERATION-PACKAGE
     -- Supplier is VMSC or GMSC if Consumer is HLR
     CONSUMER INVOKES {
          istCommand}
```

This package is v3 only.

17.2.2.51 Mobility Management event notification

This operation package includes the operations required for Mobility Management event notification procedures between VLR and gsmSCF.

```
MM-EventReportingPackage-v3 ::= OPERATION-PACKAGE
     -- Supplier is gsmSCF if Consumer is VLR
     CONSUMER INVOKES {
          noteMM-Event}
```

This package is v3 only.

17.2.2.52 Any time information handling

This operation package includes the operations required for any time information handling procedures between gsmSCF and HLR.

```
AnyTimeInformationHandlingPackage-v3 ::= OPERATION-PACKAGE
     -- Supplier is HLR if Consumer is gsmSCF
     CONSUMER INVOKES {
          anyTimeSubscriptionInterrogation,
          anyTimeModification}
```

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.

17.2.2.54 Authentication Failure Report

This operation package includes the operation required for procedures between VLR and HLR or the SGSN and the HLR for reporting of authentication failures.

```
AuthenticationFailureReportPackage-v3 ::= OPERATION-PACKAGE
     -- Supplier is HLR if Consumer is VLR
     -- Supplier is HLR if Consumer is SGSN
     CONSUMER INVOKES {
          authenticationFailureReport}
```

This package is v3 only.

17.3 Application contexts

17.3.1 General aspects

An application-context is assigned for each dialogue established by a MAP-user. In the present document each application-context is assigned a name which is supplied in the MAP-OPEN Req primitive by the MAP-User and transmitted to the peer under certain circumstances.

The following ASN.1 MACRO is used to describe the main aspects of application-contexts in the following clauses:

```
APPLICATION-CONTEXT MACRO ::=

BEGIN

TYPE NOTATION ::= Symmetric | InitiatorConsumerOf
ResponderConsumerOf | empty

VALUE NOTATION ::= value(VALUE OBJECT IDENTIFIER)

Symmetric ::= "OPERATIONS OF" "{" PackageList "}"

InitiatorConsumerOf ::= "INITIATOR CONSUMER OF" "{" PackageList "}"

ResponderConsumerOf ::= "RESPONDER CONSUMER OF" "{" PackageList "}"
          | empty

PackageList ::= Package | PackageList "," Package

Package ::= value(OPERATION-PACKAGE)
          | type -- shall reference a package type

END
```

The following definitions are used throughout this clause:

- v1-application-context: An application-context which contains only v1-packages and uses only TC v1 facilities;
- v1 context set: the set of v1-application-contexts defined in the present document.
- vn-application-context (n>=2): An application-context which contains only vn-packages;

The names of v1-application-contexts are suffixed by "-v1" while other names are suffixed by "-vn" where n>=2.

Application-contexts which do not belong to the v1 context set use v2 TC facilities.

The last component of each application-context-name (i.e. the last component of the object identifier value) assigned to an application-context which belongs to the v1 context set indicates explicitly "version1".

For each application-context which does not belong to the "v1 context set" there is a v1-equivalent application context. This is a v1-application-context which includes the v1-equivalents of the packages included in the original context.

Each application-context uses the abstract-syntax associated with the operation-packages it includes and uses the transfer-syntax derived from it by applying the encoding rules defined in clause 17.1.1.

ACs which do not belong to the v1 context set require the support of the abstract-syntax identified by the object identifier value: MAP-DialogueInformation.map-Dialogue-AS defined in clause 17.4.

17.3.2 Application context definitions

17.3.2.1 Void

17.3.2.2 Location Updating

This application context is used between HLR and VLR for location updating procedures.

```
networkLocUpContext-v3 APPLICATION-CONTEXT
     -- Responder is HLR if Initiator is VLR
     INITIATOR CONSUMER OF {
          LocationUpdatingPackage-v3,
          DataRestorationPackage-v3}
     RESPONDER CONSUMER OF {
          SubscriberDataMngtPackage-v3
          TracingPackage-v3}
::= {map-ac networkLocUp(1) version3(3)}
```

The following application-context-name is assigned to the v2-equivalent application-context:

```
{map-ac networkLocUp(1) version2(2)}
```

The following application-context-name is assigned to the v1-equivalent application-context:

```
{map-ac networkLocUp(1) version1(1)}
```

17.3.2.3 Location Cancellation

This application context is used between HLR and VLR or between HLR and SGSN for location cancellation procedures. For the HLR - SGSN interface only version 3 of this application context is applicable.

```
locationCancellationContext-v3 APPLICATION-CONTEXT
     -- Responder is VLR or SGSN if Initiator is HLR
     INITIATOR CONSUMER OF {
          LocationCancellationPackage-v3}
::= {map-ac locationCancel(2) version3(3)}
```

The following application-context-name is assigned to the v2-equivalent application-context:

```
map-ac locationCancel(2) version2(2)
```

The following application-context-name is assigned to the v1-equivalent application-context:

```
map-ac locationCancel(2) version1(1)
```

17.3.2.4 Roaming number enquiry

This application context is used between HLR and VLR for roaming number enquiry procedures.

```
roamingNumberEnquiryContext-v3 APPLICATION-CONTEXT
     -- Responder is VLR if Initiator is HLR
     INITIATOR CONSUMER OF {
          RoamingNumberEnquiryPackage-v3}
::= {map-ac roamingNbEnquiry(3) version3(3)}
```

The following application-context-name is assigned to the v2-equivalent application-context:

```
{map-ac roamingNbEnquiry(3) version2(2)}
```

The following application-context-name is assigned to the v1-equivalent application-context:

```
{map-ac roamingNbEnquiry(3) version1(1)}
```

17.3.2.5 Void

17.3.2.6 Location Information Retrieval

This application-context is used between GMSC and HLR or between GMSC and NPLR when retrieving location information. For the GMSC - NPLR interface version 1, version 2 and version 3 of this application context are applicable.

```
locationInfoRetrievalContext-v3 APPLICATION-CONTEXT
     -- Responder is HLR or NPLR if Initiator is GMSC
     INITIATOR CONSUMER OF {
          InterrogationPackage-v3}
::= {map-ac locInfoRetrieval(5) version3(3)}
```

The following application-context-name is assigned to the v2-equivalent application-context:

```
{map-ac locInfoRetrieval(5) version2(2)}
```

The following application-context-name is assigned to the v1-equivalent application-context:

```
{map-ac locInfoRetrieval(5) version1(1)}
```

17.3.2.7 Call control transfer

This application context is used for the call control transfer procedure between the VMSC and the GMSC.

```
callControlTransferContext-v4 APPLICATION-CONTEXT
     -- Responder is GMSC if Initiator is VMSC
     INITIATOR CONSUMER OF {
          CallControlTransferPackage-v4}
::= {map-ac callControlTransfer(6) version4(4)}
```

The following application-context-name is assigned to the v3-equivalent application-context:

```
{map-ac callControlTransfer(6) version3(3)}
```

17.3.2.8 - 17.3.2.10 Void

17.3.2.11 Location registers restart

This application context is used between HLR and VLR or between HLR and SGSN for location register restart procedures. For the HLR - SGSN interface version 1 and version 2 of this application context are applicable.

```
resetContext-v2 APPLICATION-CONTEXT
     -- Responder is VLR or SGSN if Initiator is HLR
     INITIATOR CONSUMER OF {
          ResetPackage-v2}
::= {map-ac reset(10) version2(2)}
```

The following application-context-name is assigned to the v1-equivalent application-context:

```
{map-ac reset(10) version1(1)}
```

17.3.2.12 Handover control

This application context is used for handover procedures between MSCs.

```
handoverControlContext-v3 APPLICATION-CONTEXT
     -- Responder is MSCB if Initiator is MSCA
     INITIATOR CONSUMER OF {
          HandoverControlPackage-v3}
::= {map-ac handoverControl(11) version3(3)}
```

The following application-context-name is assigned to the v2-equivalent application-context:

```
{map-ac handoverControl(11) version2(2)}
```

The following application-context-name is assigned to the v1-equivalent application-context:

```
{map-ac handoverControl(11) version1(1)}
```

17.3.2.13 IMSI Retrieval

This application context is used for IMSI retrieval between HLR and VLR.

```
imsiRetrievalContext-v2 APPLICATION-CONTEXT
     -- Responder is HLR if Initiator is VLR
     INITIATOR CONSUMER OF {
          IMSIRetrievalPackage-v2}
::= {map-ac imsiRetrieval(26) version2(2)}
```

This application-context is v2 only.

17.3.2.14 Equipment Management

This application context is used for equipment checking between MSC and EIR or between SGSN and EIR. For the SGSN - EIR interface version 1 and version 2 of this application context are applicable:

```
equipmentMngtContext-v2 APPLICATION-CONTEXT
     -- Responder is EIR if Initiator is MSC
     -- Responder is EIR if Initiator is SGSN
     INITIATOR CONSUMER OF {
          EquipmentMngtPackage-v2}
::= {map-ac equipmentMngt(13) version2(2)}
```

The following application-context-name is assigned to the v1-equivalent application-context:

```
{map-ac equipmentMngt(13) version1(1)}
```

17.3.2.15 Information retrieval

This application context is used for authentication information retrieval between HLR and VLR or between HLR and SGSN. For the HLR - SGSN interface version 1 and version 2 and version 3 of this application context are applicable.

```
infoRetrievalContext-v3 APPLICATION-CONTEXT
     -- Responder is HLR if Initiator is VLR
     -- Responder is HLR if Initiator is SGSN
     INITIATOR CONSUMER OF {
          InfoRetrievalPackage-v3}
::= {map-ac infoRetrieval(14) version3(3)}
```

The following application-context-name is assigned to the v2-equivalent application-context:

```
infoRetrievalContext-v2 APPLICATION-CONTEXT
     -- Responder is HLR if Initiator is VLR
     -- Responder is HLR if Initiator is SGSN
     INITIATOR CONSUMER OF {
          InfoRetrievalPackage-v2}
::= {map-ac infoRetrieval(14) version2(2)}
```

The following application-context-name is assigned to the v1-equivalent application-context:

```
{map-ac infoRetrieval(14) version1(1)}
```

17.3.2.16 Inter-VLR information retrieval

This application context is used for information retrieval between VLRs.

```
interVlrInfoRetrievalContext-v3 APPLICATION-CONTEXT
     -- Responder is VLR if Initiator is VLR
     INITIATOR CONSUMER OF {
          InterVlrInfoRetrievalPackage-v3}
::= {map-ac interVlrInfoRetrieval(15) version3(3)}
```

The v2-equivalent application-context is:

```
interVlrInfoRetrievalContext-v2 APPLICATION-CONTEXT
     -- Responder is VLR if Initiator is VLR
     INITIATOR CONSUMER OF {
          InterVlrInfoRetrievalPackage-v2}
::= {map-ac interVlrInfoRetrieval(15) version2(2)}
```

The v1-equivalent application-context is:

```
{map-ac infoRetrieval(14) version1(1)}
```

17.3.2.17 Stand Alone Subscriber Data Management

This application context is used for stand alone subscriber data management between HLR and VLR or between HLR and SGSN. For the HLR - SGSN interface only version 3 of this application context is applicable:

```
subscriberDataMngtContext-v3 APPLICATION-CONTEXT
     -- Responder is VLR or SGSN if Initiator is HLR
     INITIATOR CONSUMER OF {
          SubscriberDataMngtStandAlonePackage-v3}
::= {map-ac subscriberDataMngt(16) version3(3)}
```

The following application-context-name is assigned to the v2-equivalent application-context:

```
{map-ac subscriberDataMngt(16) version2(2)}
```

The following application-context-name is assigned to the v1-equivalent application-context:

```
{map-ac subscriberDataMngt(16) version1(1)}
```

17.3.2.18 Tracing

This application context is used between HLR and VLR or between HLR and SGSN for stand alone tracing control procedures. For the HLR - SGSN interface version 1, version 2 and version 3 of this application context are applicable.

```
tracingContext-v3 APPLICATION-CONTEXT
     -- Responder is VLR or SGSN if Initiator is HLR
     INITIATOR CONSUMER OF {
          TracingStandAlonePackage-v3}
::= {map-ac tracing(17) version3(3)}
```

The following application-context-name is assigned to the v2-equivalent application-context:

```
{map-ac tracing(17) version2(2)}
```

The following application-context-name is assigned to the v1-equivalent application-context:

```
{map-ac tracing(17) version1(1)}
```

17.3.2.19 Network functional SS handling

This application context is used for functional-like SS handling procedures between VLR and HLR.

```
networkFunctionalSsContext-v2 APPLICATION-CONTEXT
     -- Responder is HLR, Initiator is VLR
     INITIATOR CONSUMER OF {
          FunctionalSsPackage-v2}
::= {map-ac networkFunctionalSs(18) version2(2)}
```

The v1-equivalent application-context is defined as follows:

```
networkFunctionalSsContext-v1 APPLICATION-CONTEXT
     -- Responder is HLR, Initiator is VLR
     INITIATOR CONSUMER OF {
          FunctionalSsPackage-v1,
          UnstructuredSsPackage-v1,
          BindingPackage-v1}
::= {map-ac networkFunctionalSs(18) version1(1)}
```

17.3.2.20 Network unstructured SS handling

This application context is used for handling stimuli-like procedures between HLR and VLR, between the HLR and gsmSCF, and between HLR and HLR.

```
networkUnstructuredSsContext-v2 APPLICATION-CONTEXT
     -- Responder is HLR, Initiator is VLR
     -- Responder is VLR, Initiator is HLR
     -- Responder is gsmSCF, Initiator is HLR
     -- Responder is HLR, Initiator is gsmSCF
     -- Responder is HLR, Initiator is HLR
     OPERATIONS OF {
          UnstructuredSsPackage-v2}
::= {map-ac networkUnstructuredSs(19) version2(2)}
```

The following application-context-name is assigned to the v1-equivalent application-context:

```
{map-ac networkFunctionalSs(18) version1(1)}
```

17.3.2.21 Short Message Gateway

This application context is used for short message gateway procedures.

```
shortMsgGatewayContext-v3 APPLICATION-CONTEXT
     -- Responder is HLR if Initiator is GMSC
     INITIATOR CONSUMER OF {
          ShortMsgGatewayPackage-v3}
::= {map-ac shortMsgGateway(20) version3(3)}
```

The following application-context-name is assigned to the v2-equivalent application-context:

```
{map-ac shortMsgGateway(20) version2(2)}
```

The following application-context-name is assigned to the v1-equivalent application-context:

```
{map-ac shortMsgGateway(20) version1(1)}
```

17.3.2.22 Mobile originating Short Message Relay

This application context is used between MSC and IWMSC or between SGSN and IWMSC for mobile originating short message relay procedures. For the SGSN - IWMSC interface version 1, version 2 and version 3 of this application context are applicable.

```
shortMsgMO-RelayContext-v3 APPLICATION-CONTEXT
     -- Responder is IWMSC if Initiator is MSC
     -- Responder is IWMSC if Initiator is SGSN
     INITIATOR CONSUMER OF {
          MOShortMsgRelayPackage-v3}
::= {map-ac shortMsgMO-Relay(21) version3(3)}
```

The following application-context-name is assigned to the v2-equivalent application-context:

```
{map-ac shortMsgMO-Relay(21) version2(2)}
```

The following application-context-name is assigned to the v1-equivalent application-context:

```
{map-ac shortMsg-Relay(21) version1(1)}
```

17.3.2.23 Void

17.3.2.24 Short message alert

This application context is used for short message alerting procedures.

```
shortMsgAlertContext-v2 APPLICATION-CONTEXT
     -- Responder is IWMSC if Initiator is HLR
     INITIATOR CONSUMER OF {
          AlertingPackage-v2}
::= {map-ac shortMsgAlert(23) version2(2)}
```

The following application-context-name is symbolically assigned to the v1-equivalent application-context:

```
{map-ac shortMsgAlert(23) version1(1)}
```

17.3.2.25 Short message waiting data management

This application context is used between VLR and HLR or between SGSN and HLR for short message waiting data management procedures. For the SGSN - HLR interface only version 3 of this application context is applicable.

```
mwdMngtContext-v3 APPLICATION-CONTEXT
     -- Responder is HLR if Initiator is SGSN
     -- Responder is HLR if Initiator is VLR
     INITIATOR CONSUMER OF {
          MwdMngtPackage-v3}
::= {map-ac mwdMngt(24) version3(3)}
```

The following application-context-name is assigned to the v2-equivalent application-context:

```
{map-ac mwdMngt(24) version2(2)}
```

The following application-context-name is assigned to the v1-equivalent application-context:

```
{map-ac mwdMngt(24) version1(1)}
```

17.3.2.26 Mobile terminating Short Message Relay

This application context is used between GMSC and MSC or between GMSC and SGSN for mobile terminating short message relay procedures. For the GMSC - SGSN interface version 2 and version 3 of this application context and the equivalent version 1 application context are applicable.

```
shortMsgMT-RelayContext-v3 APPLICATION-CONTEXT
     -- Responder is MSC or SGSN if Initiator is GMSC
     INITIATOR CONSUMER OF {
          MTShortMsgRelayPackage-v3}
::= {map-ac shortMsgMT-Relay(25) version3(3)}
```

The following application-context-name is assigned to the v2-equivalent application-context:

```
{map-ac shortMsgMT-Relay(25) version2(2)}
```

The following application-context-name is assigned to the v1-equivalent application-context:

```
{map-ac shortMsgMO-Relay(21) version1(1)}
```

17.3.2.27 MS purging

This application context is used between HLR and VLR or between HLR and SGSN for MS purging procedures. For the SGSN - HLR interface only version 3 of this application context is applicable.

```
msPurgingContext-v3 APPLICATION-CONTEXT
     -- Responder is HLR if Initiator is VLR
     -- Responder is HLR if Initiator is SGSN
     INITIATOR CONSUMER OF {
          purgingPackage-v3}
::= {map-ac msPurging(27) version3(3)}
```

The following application-context-name is assigned to the v2-equivalent application-context:

```
{map-ac msPurging(27) version2(2)}
```

17.3.2.28 Subscriber information enquiry

This application context is used between HLR and VLR for subscriber information enquiry procedures.

```
subscriberInfoEnquiryContext-v3 APPLICATION-CONTEXT
     -- Responder is VLR if Initiator is HLR
     INITIATOR CONSUMER OF {
          SubscriberInformationEnquiryPackage-v3}
::= {map-ac subscriberInfoEnquiry(28) version3(3)}
```

This application-context is v3 only.

17.3.2.29 Any time information enquiry

This application context is used between gsmSCF and HLR or GMLC for any time information enquiry procedures.

```
anyTimeInfoEnquiryContext-v3 APPLICATION-CONTEXT
     -- Responder is HLR or GMLC if Initiator is gsmSCF
     INITIATOR CONSUMER OF {
          AnyTimeInformationEnquiryPackage-v3}
::= {map-ac anyTimeInfoEnquiry(29) version3(3)}
```

This application-context is v3 only.

17.3.2.30 Group Call Control

This application context is used between anchor MSC and relay MSC for group call and broadcast call procedures.

```
groupCallControlContext-v3 APPLICATION-CONTEXT
     -- Responder is relay MSC if Initiator is anchor MSC
     INITIATOR CONSUMER OF {
          GroupCallControlPackage-v3}
::= {map-ac groupCallControl(31) version3(3)}
```

This application-context is v3 only.

17.3.2.31 Provide SIWFS Number

This application context is used for activation or modification of SIWF resources.

```
sIWFSAllocationContext-v3 APPLICATION-CONTEXT
     -- Responder is SIWF if Initiater is VMSC
     INITIATOR CONSUMER OF {
          ProvideSIWFSNumberPackage-v3,
          SIWFSSignallingModifyPackage-v3}
::= {map-ac sIWFSAllocation (12) version3(3)}
```

This application-context is v3 only.

17.3.2.32 Gprs Location Updating

This application context is used between HLR and SGSN for gprs location updating procedures.

```
gprsLocationUpdateContext-v3 APPLICATION-CONTEXT
     -- Responder is HLR if Initiator is SGSN
     INITIATOR CONSUMER OF {
          GprsLocationUpdatingPackage-v3}
     RESPONDER CONSUMER OF {
          SubscriberDataMngtPackage-v3
          TracingPackage-v3}
::= {map-ac gprsLocationUpdate(32) version3(3)}
```

This application-context is v3 only.

17.3.2.33 Gprs Location Information Retreival

This application context is used between HLR and GGSN when retrieving gprs location information.

```
gprsLocationInfoRetrievalContext-v3 APPLICATION-CONTEXT
     -- Responder is HLR if Initiator is GGSN
     INITIATOR CONSUMER OF {
          GprsInterrogationPackage-v3}
::= {map-ac gprsLocationInfoRetrieval(33) version3(3)}
```

This application-context is v3 only.

17.3.2.34 Failure Reporting

This application context is used between HLR and GGSN to inform that network requested PDP-context activation has failed.

```
failureReportContext-v3 APPLICATION-CONTEXT
     -- Responder is HLR if Initiator is GGSN
     INITIATOR CONSUMER OF {
          FailureReportingPackage-v3}
::= {map-ac failureReport(34) version3(3)}
```

This application-context is v3 only.

17.3.2.35 GPRS Notifying

This application context is used between HLR and GGSN for notifying that GPRS subscriber is present again.

```
gprsNotifyContext-v3 APPLICATION-CONTEXT
     -- Responder is GGSN if Initiator is HLR
     INITIATOR CONSUMER OF {
          GprsNotifyingPackage-v3}
::= {map-ac gprsNotify(35) version3(3)}
```

This application-context is v3 only.

17.3.2.36 Supplementary Service invocation notification

This application context is used between the MSC and the gsmSCF and between the HLR and the gsmSCF for Supplementary Service invocation notification procedures.

```
ss-InvocationNotificationContext-v3 APPLICATION-CONTEXT
     -- Responder is gsmSCF, Initiator is MSC
     -- Responder is gsmSCF, Initiator is HLR
     INITIATOR CONSUMER OF {
          SS-InvocationNotificationPackage-v3}
::= {map-ac ss-InvocationNotification(36) version3(3)}
```

This application-context is v3 only.

17.3.2.37 Reporting

This application context is used between HLR and VLR for reporting procedures.

```
reportingContext-v3 APPLICATION-CONTEXT
     -- Responder is VLR if Initiator is HLR
     -- Responder is HLR if Initiator is VLR
     INITIATOR CONSUMER OF {
          SetReportingStatePackage-v3,
          StatusReportPackage-v3,
          RemoteUserFreePackage-v3}
     RESPONDER CONSUMER OF {
          SetReportingStatePackage-v3,
          StatusReportPackage-v3}
::= {map-ac reporting(7) version3(3)}
```

This application-context is v3 only.

17.3.2.38 Call Completion

This application context is used between VLR and the HLR for subscriber control of call completion services.

```
callCompletionContext-v3 APPLICATION-CONTEXT
     -- Responder is HLR if Initiator is VLR
     INITIATOR CONSUMER OF {
          CallCompletionPackage-v3}
::= {map-ac callCompletion(8) version3(3)}
```

This application-context is v3 only.

17.3.2.39 Location Service Gateway

This application context is used for location service gateway procedures.

```
locationSvcGatewayContext-v3 APPLICATION-CONTEXT
     -- Responder is HLR if Initiator is GMLC
     INITIATOR CONSUMER OF {
          locationSvcGatewayPackage-v3}
::= {map-ac locationSvcGateway(37) version3(3)}
```

17.3.2.40 Location Service Enquiry

This application context is used for location service enquiry procedures.

```
locationSvcEnquiryContext-v3 APPLICATION-CONTEXT
     -- Responder is MSC if Initiator is GMLC
     -- Responder is GMLC if Initiator is MSC
     INITIATOR CONSUMER OF {
          locationSvcEnquiryPackage-v3,
          locationSvcReportingPackage-v3}
::= {map-ac locationSvcEnquiry(38) version3 (3)}
```

17.3.2.41 Void

17.3.2.42 Void

17.3.2.43 Void

17.3.2.44 IST Alerting

This application context is used between MSC (Visited MSC or Gateway MSC) and HLR for alerting services within IST procedures.

```
istAlertingContext-v3 APPLICATION-CONTEXT
     -- Responder is HLR if Initiator is VMSC
     -- Responder is HLR if Initiator is GMSC
     INITIATOR CONSUMER OF {
          IST-AlertingPackage-v3}
::= {map-ac alerting(4) version3(3)}
```

This application-context is v3 only.

17.3.2.45 Service Termination

This application context is used between HLR and MSC (Visited MSC or Gateway MSC) for service termination services within IST procedures.

```
serviceTerminationContext-v3 APPLICATION-CONTEXT
     -- Responder is VMSC or GMSC if Initiator is HLR
     INITIATOR CONSUMER OF {
          ServiceTerminationPackage-v3}
::= {map-ac serviceTermination(9) version3(3)}
```

This application-context is v3 only.

17.3.2.46 Mobility Management event notification

This application context is used between VLR and gsmSCF for Mobility Management event notification procedures.

```
mm-EventReportingContext-v3 APPLICATION-CONTEXT
     -- Responder is gsmSCF, Initiator is VLR
     INITIATOR CONSUMER OF {
          MM-EventReportingPackage-v3}
::= {map-ac mm-EventReporting(42) version3(3)}
```

This application-context is v3 only.

17.3.2.47 Any time information handling

This application context is used between gsmSCF and HLR for any time information handling procedures.

```
anyTimeInfohandlingContext-v3 APPLICATION-CONTEXT
     -- Responder is HLR if Initiator is gsmSCF
     INITIATOR CONSUMER OF {
          AnyTimeInformationHandlingPackage-v3}
::= {map-ac anyTimeInfoHandling(43) version3(3)}
```

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.

17.3.2.49 Authentication Failure Report

This application context is used between VLR and HLR or SGSN and HLR for reporting of authentication failures.

```
authenticationFailureReportContext-v3 APPLICATION-CONTEXT
     -- Responder is HLR if Initiator is VLR
     -- Responder is HLR if Initiator is SGSN
     INITIATOR CONSUMER OF {
          AuthenticationFailureReportPackage-v3 }
::= {map-ac authenticationFailureReport(39) version3(3)}
```

This application-context is v3 only.

17.3.3 ASN.1 Module for application-context-names

The following ASN.1 module summarises the application-context-name assigned to MAP application-contexts.

```
MAP-ApplicationContexts {
   ccitt identified-organization (4) etsi (0) mobileDomain (0)
   gsm-Network (1) modules (3) map-ApplicationContexts (2) version6 (6)}

DEFINITIONS

::=

BEGIN


-- EXPORTS everything


IMPORTS
   gsm-NetworkId,
   ac-Id
FROM MobileDomainDefinitions {
   ccitt (0) identified-organization (4) etsi (0) mobileDomain (0)
   mobileDomainDefinitions (0) version1 (1)}
;

-- application-context-names

map-ac  OBJECT IDENTIFIER ::= {gsm-NetworkId ac-Id}

networkLocUpContext-v3  OBJECT IDENTIFIER ::=
     {map-ac networkLocUp(1) version3(3)}

locationCancellationContext-v3  OBJECT IDENTIFIER ::=
     {map-ac locationCancel(2) version3(3)}

roamingNumberEnquiryContext-v3  OBJECT IDENTIFIER ::=
     {map-ac roamingNbEnquiry(3) version3(3)}

authenticationFailureReportContext-v3  OBJECT IDENTIFIER ::=
     {map-ac authenticationFailureReport(39) version3(3)}

locationInfoRetrievalContext-v3  OBJECT IDENTIFIER ::=
     {map-ac locInfoRetrieval(5) version3(3)}

resetContext-v2  OBJECT IDENTIFIER ::=
     {map-ac reset(10) version2(2)}
```

```
handoverControlContext-v3  OBJECT IDENTIFIER ::=
     {map-ac handoverControl(11) version3(3)}

equipmentMngtContext-v2  OBJECT IDENTIFIER ::=
     {map-ac equipmentMngt(13) version2(2)}

infoRetrievalContext-v3  OBJECT IDENTIFIER ::=
     {map-ac infoRetrieval(14) version3(3)}

interVlrInfoRetrievalContext-v3  OBJECT IDENTIFIER ::=
     {map-ac interVlrInfoRetrieval(15) version3(3)}

subscriberDataMngtContext-v3  OBJECT IDENTIFIER ::=
     {map-ac subscriberDataMngt(16) version3(3)}

tracingContext-v3  OBJECT IDENTIFIER ::=
     {map-ac tracing(17) version3(3)}

networkFunctionalSsContext-v2  OBJECT IDENTIFIER ::=
     {map-ac networkFunctionalSs(18) version2(2)}

networkUnstructuredSsContext-v2  OBJECT IDENTIFIER ::=
     {map-ac networkUnstructuredSs(19) version2(2)}

shortMsgGatewayContext-v3  OBJECT IDENTIFIER ::=
     {map-ac shortMsgGateway(20) version3(3)}

shortMsgMO-RelayContext-v3  OBJECT IDENTIFIER ::=
     {map-ac shortMsgMO-Relay(21) version3(3)}

shortMsgAlertContext-v2  OBJECT IDENTIFIER ::=
     {map-ac shortMsgAlert(23) version2(2)}

mwdMngtContext-v3  OBJECT IDENTIFIER ::=
     {map-ac mwdMngt(24) version3(3)}

shortMsgMT-RelayContext-v3  OBJECT IDENTIFIER ::=
     {map-ac shortMsgMT-Relay(25) version3(3)}

imsiRetrievalContext-v2  OBJECT IDENTIFIER ::=
     {map-ac imsiRetrieval(26) version2(2)}

msPurgingContext-v3  OBJECT IDENTIFIER ::=
     {map-ac msPurging(27) version3(3)}

subscriberInfoEnquiryContext-v3  OBJECT IDENTIFIER ::=
     {map-ac subscriberInfoEnquiry(28) version3(3)}

anyTimeInfoEnquiryContext-v3  OBJECT IDENTIFIER ::=
     {map-ac anyTimeInfoEnquiry(29) version3(3)}

callControlTransferContext-v4  OBJECT IDENTIFIER ::=
     {map-ac callControlTransfer(6) version4(4)}

ss-InvocationNotificationContext-v3  OBJECT IDENTIFIER ::=
     {map-ac ss-InvocationNotification(36) version3(3)}

sIWFSAllocationContext-v3  OBJECT IDENTIFIER ::=
     {map-ac sIWFSAllocation(12) version3(3)}

groupCallControlContext-v3  OBJECT IDENTIFIER ::=
     {map-ac groupCallControl(31) version3(3)}

gprsLocationUpdateContext-v3  OBJECT IDENTIFIER ::=
     {map-ac gprsLocationUpdate(32) version3(3)}

gprsLocationInfoRetrievalContext-v3  OBJECT IDENTIFIER ::=
     {map-ac gprsLocationInfoRetrieval(33) version3(3)}

failureReportContext-v3  OBJECT IDENTIFIER ::=
     {map-ac failureReport(34) version3(3)}

gprsNotifyContext-v3  OBJECT IDENTIFIER ::=
     {map-ac gprsNotify(35) version3(3)}
```

```
reportingContext-v3  OBJECT IDENTIFIER ::=
     {map-ac reporting(7) version3(3)}

callCompletionContext-v3  OBJECT IDENTIFIER ::=
     {map-ac callCompletion(8) version3(3)}

istAlertingContext-v3  OBJECT IDENTIFIER ::=
     {map-ac istAlerting(4) version3(3)}

serviceTerminationContext-v3  OBJECT IDENTIFIER ::=
     {map-ac immediateTermination(9) version3(3)}

locationSvcGatewayContext-v3  OBJECT IDENTIFIER ::=
     {map-ac locationSvcGateway(37) version3(3)}

locationSvcEnquiryContext-v3  OBJECT IDENTIFIER ::=
     {map-ac locationSvcEnquiry(38) version3(3)}

mm-EventReportingContext-v3  OBJECT IDENTIFIER ::=
     {map-ac mm-EventReporting(42) version3(3)}

anyTimeInfoHandlingContext-v3  OBJECT IDENTIFIER ::=
     {map-ac anyTimeInfoHandling(43) version3(3)}

subscriberDataModificationNotificationContext-v3  OBJECT IDENTIFIER ::=
     {map-ac subscriberDataModificationNotification(22) version3(3)}

-- The following Object Identifiers are reserved for application-
-- contexts existing in previous versions of the protocol
```

-- AC Name & Version	Object Identifier	
--		
-- networkLocUpContext-v1	map-ac networkLocUp (1)	version1 (1)
-- networkLocUpContext-v2	map-ac networkLocUp (1)	version2 (2)
-- locationCancellationContext-v1	map-ac locationCancellation (2)	version1 (1)
-- locationCancellationContext-v2	map-ac locationCancellation (2)	version2 (2)
-- roamingNumberEnquiryContext-v1	map-ac roamingNumberEnquiry (3)	version1 (1)
-- roamingNumberEnquiryContext-v2	map-ac roamingNumberEnquiry (3)	version2 (2)
-- locationInfoRetrievalContext-v1	map-ac locationInfoRetrieval (5)	version1 (1)
-- locationInfoRetrievalContext-v2	map-ac locationInfoRetrieval (5)	version2 (2)
-- resetContext-v1	map-ac reset (10)	version1 (1)
-- handoverControlContext-v1	map-ac handoverControl (11)	version1 (1)
-- handoverControlContext-v2	map-ac handoverControl (11)	version2 (2)
-- equipmentMngtContext-v1	map-ac equipmentMngt (13)	version1 (1)
-- infoRetrievalContext-v1	map-ac infoRetrieval (14)	version1 (1)
-- infoRetrievalContext-v2	map-ac infoRetrieval (14)	version2 (2)
-- interVlrInfoRetrievalContext-v2	map-ac interVlrInfoRetrieval (15)	version2 (2)
-- subscriberDataMngtContext-v1	map-ac subscriberDataMngt (16)	version1 (1)
-- subscriberDataMngtContext-v2	map-ac subscriberDataMngt (16)	version2 (2)
-- tracingContext-v1	map-ac tracing (17)	version1 (1)
-- tracingContext-v2	map-ac tracing (17)	version2 (2)
-- *networkFunctionalSsContext-v1*	*map-ac networkFunctionalSs (18)*	*version1 (1)*
-- shortMsgGatewayContext-v1	map-ac shortMsgGateway (20)	version1 (1)
-- shortMsgGatewayContext-v2	map-ac shortMsgGateway (20)	version2 (2)
-- shortMsgRelayContext-v1	map-ac shortMsgRelay (21)	version1 (1)
-- shortMsgAlertContext-v1	map-ac shortMsgAlert (23)	version1 (1)
-- *mwdMngtContext-v1*	*map-ac mwdMngt (24)*	*version1 (1)*
-- mwdMngtContext-v2	map-ac mwdMngt (24)	version2 (2)
-- shortMsgMT-RelayContext-v2	map-ac shortMsgMT-Relay (25)	version2 (2)
-- msPurgingContext-v2	map-ac msPurging (27)	version2 (2)
-- callControlTransferContext-v3	map-ac callControlTransferContext (6)	version3 (3)

```
END
```

17.4 MAP Dialogue Information

```
MAP-DialogueInformation {
   ccitt identified-organization (4) etsi (0) mobileDomain (0)
   gsm-Network (1) modules (3) map-DialogueInformation (3) version6 (6)}

DEFINITIONS

IMPLICIT TAGS
```

```
::=

BEGIN

EXPORTS
   map-DialogueAS,
   MAP-DialoguePDU
;

IMPORTS
   gsm-NetworkId,
   as-Id
FROM MobileDomainDefinitions {
   ccitt (0) identified-organization (4) etsi (0) mobileDomain (0)
   mobileDomainDefinitions (0) version1 (1)}

   AddressString
FROM MAP-CommonDataTypes {
   ccitt identified-organization (4) etsi (0) mobileDomain (0)
   gsm-Network(1) modules (3) map-CommonDataTypes (18) version6 (6)}

   ExtensionContainer
FROM MAP-ExtensionDataTypes {
   ccitt identified-organization (4) etsi (0) mobileDomain (0)
   gsm-Network (1) modules (3) map-ExtensionDataTypes (21) version6 (6)}

;


-- abstract syntax name for MAP-DialoguePDU
```

```
map-DialogueAS  OBJECT IDENTIFIER ::=
     {gsm-NetworkId as-Id map-DialoguePDU (1) version1 (1)}
```

```
MAP-DialoguePDU ::= CHOICE {
     map-open                              [0] MAP-OpenInfo,
     map-accept                            [1] MAP-AcceptInfo,
     map-close                             [2] MAP-CloseInfo,
     map-refuse                            [3] MAP-RefuseInfo,
     map-userAbort                         [4] MAP-UserAbortInfo,
     map-providerAbort                     [5] MAP-ProviderAbortInfo}
```

```
MAP-OpenInfo ::= SEQUENCE {
     destinationReference                  [0] AddressString                 OPTIONAL,
     originationReference                  [1] AddressString                 OPTIONAL,
     ...,
     extensionContainer                    ExtensionContainer                OPTIONAL
     -- extensionContainer must not be used in version 2
     }
```

```
MAP-AcceptInfo ::= SEQUENCE {
     ...,
     extensionContainer                    ExtensionContainer                OPTIONAL
     -- extensionContainer must not be used in version 2
     }
```

```
MAP-CloseInfo ::= SEQUENCE {
     ...,
     extensionContainer                    ExtensionContainer                OPTIONAL
     -- extensionContainer must not be used in version 2
     }
```

```
MAP-RefuseInfo ::= SEQUENCE {
     reason    Reason,
     ...,
     extensionContainer                    ExtensionContainer                OPTIONAL
     -- extensionContainer must not be used in version 2
     }
```

```
Reason ::= ENUMERATED {
     noReasonGiven  (0),
     invalidDestinationReference  (1),
     invalidOriginatingReference  (2)}
```

```
MAP-UserAbortInfo ::= SEQUENCE {
     map-UserAbortChoice                MAP-UserAbortChoice,
     ...,
     extensionContainer                 ExtensionContainer              OPTIONAL
     -- extensionContainer must not be used in version 2
     }
```

```
MAP-UserAbortChoice ::= CHOICE {
     userSpecificReason                 [0] NULL,
     userResourceLimitation             [1] NULL,
     resourceUnavailable                [2] ResourceUnavailableReason,
     applicationProcedureCancellation   [3] ProcedureCancellationReason}
```

```
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            MAP-ProviderAbortReason,
     ...,
     extensionContainer                 ExtensionContainer              OPTIONAL
     -- extensionContainer must not be used in version 2
     }
```

```
MAP-ProviderAbortReason ::= ENUMERATED {
     abnormalDialogue  (0),
     invalidPDU  (1)}
```

```
END
```

17.5 MAP operation and error codes

```
MAP-Protocol {
   ccitt identified-organization (4) etsi (0) mobileDomain (0)
   gsm-Network (1) modules (3) map-Protocol (4) version6 (6)}

DEFINITIONS

::=

BEGIN

IMPORTS
   UpdateLocation,
   CancelLocation,
   PurgeMS,
   SendIdentification,
   UpdateGprsLocation,
   PrepareHandover,
   SendEndSignal,
   ProcessAccessSignalling,
   ForwardAccessSignalling,
   PrepareSubsequentHandover,
   SendAuthenticationInfo,
   AuthenticationFailureReport,
   CheckIMEI,
   InsertSubscriberData,
   DeleteSubscriberData,
   Reset,
   ForwardCheckSS-Indication,
   RestoreData,
   ProvideSubscriberInfo,
   AnyTimeInterrogation,
   AnyTimeSubscriptionInterrogation,
   AnyTimeModification,
   SendRoutingInfoForGprs,
   FailureReport,
   NoteMsPresentForGprs,
```

```
   NoteMM-Event,
   NoteSubscriberDataModified


FROM MAP-MobileServiceOperations {
   ccitt identified-organization (4) etsi (0) mobileDomain (0)
   gsm-Network (1) modules (3) map-MobileServiceOperations (5)
   version6 (6)}

   ActivateTraceMode,
   DeactivateTraceMode,
   SendIMSI
FROM MAP-OperationAndMaintenanceOperations {
   ccitt identified-organization (4) etsi (0) mobileDomain (0)
   gsm-Network (1) modules (3) map-OperationAndMaintenanceOperations (6)
   version6 (6)}

   SendRoutingInfo,
   ProvideRoamingNumber,
   ResumeCallHandling,
   ProvideSIWFSNumber,
   SIWFSSignallingModify,
   SetReportingState,
   StatusReport,
   RemoteUserFree,
   IST-Alert,
   IST-Command
FROM MAP-CallHandlingOperations {
   ccitt identified-organization (4) etsi (0) mobileDomain (0)
   gsm-Network (1) modules (3) map-CallHandlingOperations (7)
   version6 (6)}

   RegisterSS,
   EraseSS,
   ActivateSS,
   DeactivateSS,
   InterrogateSS,
   ProcessUnstructuredSS-Request,
   UnstructuredSS-Request,
   UnstructuredSS-Notify,
   RegisterPassword,
   GetPassword,
   SS-InvocationNotification,
   RegisterCC-Entry,
   EraseCC-Entry
FROM MAP-SupplementaryServiceOperations {
   ccitt identified-organization (4) etsi (0) mobileDomain (0)
   gsm-Network (1) modules (3) map-SupplementaryServiceOperations (8)
   version6 (6)}

   SendRoutingInfoForSM,
   MO-ForwardSM,
   MT-ForwardSM,
   ReportSM-DeliveryStatus,
   AlertServiceCentre,
   InformServiceCentre,
   ReadyForSM
FROM MAP-ShortMessageServiceOperations {
   ccitt identified-organization (4) etsi (0) mobileDomain (0)
   gsm-Network (1) modules (3) map-ShortMessageServiceOperations (9)
   version6 (6)}

   PrepareGroupCall,
   ProcessGroupCallSignalling,
   ForwardGroupCallSignalling,
   SendGroupCallEndSignal
FROM MAP-Group-Call-Operations {
   ccitt identified-organization (4) etsi (0) mobileDomain (0)
   gsm-Network (1) modules (3) map-Group-Call-Operations (22)
   version6 (6)}

   ProvideSubscriberLocation,
   SendRoutingInfoForLCS,
   SubscriberLocationReport
FROM MAP-LocationServiceOperations {
   ccitt identified-organization (4) etsi (0) mobileDomain (0)
   gsm-Network (1) modules (3) map-LocationServiceOperations (24)
   version6 (6)}
```

```
   SystemFailure,
   DataMissing,
   UnexpectedDataValue,
   FacilityNotSupported,
   UnknownSubscriber,
   NumberChanged,
   UnknownMSC,
   UnidentifiedSubscriber,
   UnknownEquipment,
   RoamingNotAllowed,
   IllegalSubscriber,
   IllegalEquipment,
   BearerServiceNotProvisioned,
   TeleserviceNotProvisioned,
   NoHandoverNumberAvailable,
   SubsequentHandoverFailure,
   TracingBufferFull,
   OR-NotAllowed,
   NoRoamingNumberAvailable,
   AbsentSubscriber,
   BusySubscriber,
   NoSubscriberReply,
   CallBarred,
   ForwardingViolation,
   ForwardingFailed,
   CUG-Reject,
   ATI-NotAllowed,
   IllegalSS-Operation,
   SS-ErrorStatus,
   SS-NotAvailable,
   SS-SubscriptionViolation,
   SS-Incompatibility,
   UnknownAlphabet,
   USSD-Busy,
   PW-RegistrationFailure,
   NegativePW-Check,
   NumberOfPW-AttemptsViolation,
   SubscriberBusyForMT-SMS,
   SM-DeliveryFailure,
   MessageWaitingListFull,
   AbsentSubscriberSM,
   ResourceLimitation,
   NoGroupCallNumberAvailable,
   ShortTermDenial,
   LongTermDenial,
   IncompatibleTerminal,
   UnauthorizedRequestingNetwork,
   UnauthorizedLCSClient,
   PositionMethodFailure,
   UnknownOrUnreachableLCSClient,
   ATSI-NotAllowed,
   ATM-NotAllowed,
   InformationNotAvailable,
   MM-EventNotSupported,
   TargetCellOutsideGroupCallArea


FROM MAP-Errors {
   ccitt identified-organization (4) etsi (0) mobileDomain (0)
   gsm-Network (1) modules (3) map-Errors (10) version6 (6)}
;


-- location registration operation codes

updateLocation  UpdateLocation ::= localValue 2
cancelLocation  CancelLocation ::= localValue 3
purgeMS  PurgeMS ::= localValue 67
sendIdentification  SendIdentification ::= localValue 55


-- handover operation codes
```

```
prepareHandover  PrepareHandover ::= localValue 68
sendEndSignal  SendEndSignal ::= localValue 29
processAccessSignalling  ProcessAccessSignalling ::= localValue 33
forwardAccessSignalling  ForwardAccessSignalling ::= localValue 34
prepareSubsequentHandover  PrepareSubsequentHandover ::=
     localValue 69
```

-- authentication operation codes

```
sendAuthenticationInfo  SendAuthenticationInfo ::= localValue 56
authenticationFailureReport  AuthenticationFailureReport ::= localValue 15
```

-- IMEI MANAGEMENT operation codes

```
checkIMEI  CheckIMEI ::= localValue 43
```

-- subscriber management operation codes

```
insertSubscriberData  InsertSubscriberData ::= localValue 7
deleteSubscriberData  DeleteSubscriberData ::= localValue 8
```

-- fault recovery operation codes

```
reset  Reset ::= localValue 37
forwardCheckSS-Indication  ForwardCheckSS-Indication ::=
     localValue 38
restoreData  RestoreData ::= localValue 57
```

-- operation and maintenance operation codes

```
activateTraceMode  ActivateTraceMode ::= localValue 50
deactivateTraceMode  DeactivateTraceMode ::= localValue 51
sendIMSI  SendIMSI ::= localValue 58
```

-- call handling operation codes

```
sendRoutingInfo  SendRoutingInfo ::= localValue 22
provideRoamingNumber  ProvideRoamingNumber ::= localValue 4
resumeCallHandling  ResumeCallHandling ::= localValue 6
provideSIWFSNumber  ProvideSIWFSNumber ::= localValue 31
sIWFSSignallingModify  SIWFSSignallingModify ::= localValue 32
setReportingState  SetReportingState ::= localValue 73
statusReport StatusReport ::= localValue 74
remoteUserFree RemoteUserFree ::= localValue 75
istAlert IST-Alert ::= localValue 87
istCommand IST-Command ::= localValue 88
```

-- supplementary service handling operation codes

```
registerSS  RegisterSS ::= localValue 10
eraseSS  EraseSS ::= localValue 11
activateSS  ActivateSS ::= localValue 12
deactivateSS  DeactivateSS ::= localValue 13
interrogateSS  InterrogateSS ::= localValue 14
processUnstructuredSS-Request  ProcessUnstructuredSS-Request ::=
     localValue 59
unstructuredSS-Request  UnstructuredSS-Request ::= localValue 60
unstructuredSS-Notify  UnstructuredSS-Notify ::= localValue 61
registerPassword  RegisterPassword ::= localValue 17
getPassword  GetPassword ::= localValue 18
registerCC-Entry  RegisterCC-Entry ::= localValue 76
eraseCC-Entry  EraseCC-Entry ::= localValue 77
```

-- short message service operation codes

```
sendRoutingInfoForSM  SendRoutingInfoForSM ::= localValue 45
mo-forwardSM  MO-ForwardSM ::= localValue 46
mt-forwardSM  MT-ForwardSM ::= localValue 44
reportSM-DeliveryStatus  ReportSM-DeliveryStatus ::= localValue 47
informServiceCentre  InformServiceCentre ::= localValue 63
alertServiceCentre  AlertServiceCentre ::= localValue 64
readyForSM  ReadyForSM ::= localValue 66
```

```
-- provide subscriber info operation codes
```

```
provideSubscriberInfo  ProvideSubscriberInfo ::= localValue 70
```

```
-- any time interrogation operation codes
```

```
anyTimeInterrogation  AnyTimeInterrogation ::= localValue 71
```

```
-- any time information handling operation codes
```

```
anyTimeSubscriptionInterrogation  AnyTimeSubscriptionInterrogation ::= localValue 62
anyTimeModification  AnyTimeModification ::= localValue 65
```

```
-- subscriber data modification notification operation codes
```

```
noteSubscriberDataModified  NoteSubscriberDataModified ::= localValue 5
```

```
-- supplementary service invocation notification operation codes
```

```
ss-InvocationNotification  SS-InvocationNotification ::= localValue 72
```

```
--Group Call operation codes
```

```
prepareGroupCall  PrepareGroupCall ::= localValue 39
sendGroupCallEndSignal  SendGroupCallEndSignal ::= localValue 40
processGroupCallSignalling  ProcessGroupCallSignalling ::= localValue 41
forwardGroupCallSignalling  ForwardGroupCallSignalling ::= localValue 42
```

```
-- gprs location updating operation codes
```

```
updateGprsLocation  UpdateGprsLocation ::= localValue 23
```

```
-- gprs location information retrieval operation codes
```

```
sendRoutingInfoForGprs  SendRoutingInfoForGprs ::= localValue 24
```

```
-- failure reporting operation codes
```

```
failureReport  FailureReport ::= localValue 25
```

```
-- GPRS notification operation codes
```

```
noteMsPresentForGprs  NoteMsPresentForGprs ::= localValue 26
```

```
-- Location service operation codes
```

```
provideSubscriberLocation  ProvideSubscriberLocation ::= localValue 83
sendRoutingInfoForLCS  SendRoutingInfoForLCS ::= localValue 85
subscriberLocationReport  SubscriberLocationReport ::= localValue 86
```

```
-- Mobility Management operation codes
```

```
noteMM-Event  NoteMM-Event ::= localValue 89
```

```
-- generic error codes
```

```
systemFailure  SystemFailure ::= localValue 34
dataMissing  DataMissing ::= localValue 35
unexpectedDataValue  UnexpectedDataValue ::= localValue 36
facilityNotSupported  FacilityNotSupported ::= localValue 21
incompatibleTerminal  IncompatibleTerminal ::= localValue 28
resourceLimitation  ResourceLimitation ::= localValue 51
```

```
-- identification and numbering error codes

unknownSubscriber  UnknownSubscriber ::= localValue 1
numberChanged  NumberChanged ::= localValue 44
unknownMSC  UnknownMSC ::= localValue 3
unidentifiedSubscriber  UnidentifiedSubscriber ::= localValue 5
unknownEquipment  UnknownEquipment ::= localValue 7

-- subscription error codes

roamingNotAllowed  RoamingNotAllowed ::= localValue 8
illegalSubscriber  IllegalSubscriber ::= localValue 9
illegalEquipment  IllegalEquipment ::= localValue 12
bearerServiceNotProvisioned  BearerServiceNotProvisioned ::=
     localValue 10
teleserviceNotProvisioned  TeleserviceNotProvisioned ::=
     localValue 11

-- handover error codes

noHandoverNumberAvailable  NoHandoverNumberAvailable ::=
     localValue 25
subsequentHandoverFailure  SubsequentHandoverFailure ::=
     localValue 26
targetCellOutsideGroupCallArea  TargetCellOutsideGroupCallArea ::=
     localValue 42

-- operation and maintenance error codes

tracingBufferFull  TracingBufferFull ::= localValue 40

-- call handling error codes

noRoamingNumberAvailable  NoRoamingNumberAvailable ::= localValue 39
absentSubscriber  AbsentSubscriber ::= localValue 27
busySubscriber  BusySubscriber ::= localValue 45
noSubscriberReply  NoSubscriberReply ::= localValue 46
callBarred  CallBarred ::= localValue 13
forwardingFailed  ForwardingFailed ::= localValue 47
or-NotAllowed  OR-NotAllowed ::= localValue 48
forwardingViolation  ForwardingViolation ::= localValue 14
cug-Reject  CUG-Reject ::= localValue 15

-- any time interrogation error codes

ati-NotAllowed  ATI-NotAllowed ::= localValue 49

-- any time information handling error codes

atsi-NotAllowed  ATSI-NotAllowed ::= localValue 60
atm-NotAllowed  ATM-NotAllowed ::= localValue 61
informationNotAvailable  InformationNotAvailable ::= localValue 62

-- Group Call error codes

noGroupCallNumberAvailable  NoGroupCallNumberAvailable ::= localValue 50

-- supplementary service error codes
```

```
illegalSS-Operation  IllegalSS-Operation ::= localValue 16
ss-ErrorStatus  SS-ErrorStatus ::= localValue 17
ss-NotAvailable  SS-NotAvailable ::= localValue 18
ss-SubscriptionViolation  SS-SubscriptionViolation ::= localValue 19
ss-Incompatibility  SS-Incompatibility ::= localValue 20
unknownAlphabet  UnknownAlphabet ::= localValue 71
ussd-Busy  USSD-Busy ::= localValue 72
pw-RegistrationFailure  PW-RegistrationFailure ::= localValue 37
negativePW-Check  NegativePW-Check ::= localValue 38
numberOfPW-AttemptsViolation  NumberOfPW-AttemptsViolation ::=
     localValue 43
shortTermDenial  ShortTermDenial ::= localValue 29
longTermDenial  LongTermDenial ::= localValue 30
```

```
-- short message service error codes
```

```
subscriberBusyForMT-SMS  SubscriberBusyForMT-SMS ::= localValue 31
sm-DeliveryFailure  SM-DeliveryFailure ::= localValue 32
messageWaitingListFull  MessageWaitingListFull ::= localValue 33
absentsubscriberSM  AbsentSubscriberSM ::= localValue 6
```

```
-- location service error codes
```

```
unauthorizedRequestingNetwork  UnauthorizedRequestingNetwork::= localValue 52
unauthorizedLCSClient  UnauthorizedLCSClient ::= localValue 53
positionMethodFailure  PositionMethodFailure ::= localValue 54
unknownOrUnreachableLCSClient  UnknownOrUnreachableLCSClient ::= localValue 58
```

```
-- Mobility Management error codes
mm-EventNotSupported  MM-EventNotSupported ::= localValue 59
```

```
-- The following operation codes are reserved for operations
-- existing in previous versions of the protocol
```

-- Operation Name	AC used	Oper. Code
--		
-- sendParameters	map-ac infoRetrieval (14) version1 (1)	localValue 9
-- processUnstructuredSS-Data	map-ac networkFunctionalSs (18) version1 (1)	localValue 19
-- performHandover	map-ac handoverControl (11) version1 (1)	localValue 28
-- performSubsequentHandover	map-ac handoverControl (11) version1 (1)	localValue 30
-- noteInternalHandover	map-ac handoverControl (11) version1 (1)	localValue 35
-- noteSubscriberPresent	map-ac mwdMngt (24) version1 (1)	localValue 48
-- alertServiceCentreWithoutResult	map-ac shortMsgAlert (23) version1 (1)	localValue 49
-- traceSubscriberActivity	map-ac handoverControl (11) version1 (1)	localValue 52
-- beginSubscriberActivity	map-ac networkFunctionalSs (18) version1 (1)	localValue 54

```
-- The following error codes are reserved for errors
-- existing in previous versions of the protocol
```

-- Error Name	AC used	Error Code
--		
-- unknownBaseStation	map-ac handoverControl (11) version1 (1)	localValue 2
-- invalidTargetBaseStation	map-ac handoverControl (11) version1 (1)	localValue 23
-- noRadioResourceAvailable	map-ac handoverControl (11) version1 (1)	localValue 24

```
END
```

17.6 MAP operation and error types

17.6.1 Mobile Service Operations

```
MAP-MobileServiceOperations {
   ccitt identified-organization (4) etsi (0) mobileDomain (0)
   gsm-Network (1) modules (3) map-MobileServiceOperations (5)
   version6 (6)}

DEFINITIONS

::=

BEGIN
```

```
EXPORTS

   -- location registration operations
   UpdateLocation,
   CancelLocation,
   PurgeMS,
   SendIdentification,

   -- gprs location registration operations
   UpdateGprsLocation,

   -- subscriber information enquiry operations
   ProvideSubscriberInfo,

   -- any time information enquiry operations
   AnyTimeInterrogation,

   -- any time information handling operations
   AnyTimeSubscriptionInterrogation,
   AnyTimeModification,

   -- subscriber data modification notification operations
   NoteSubscriberDataModified,


   -- handover operations
   PrepareHandover,
   SendEndSignal,
   ProcessAccessSignalling,
   ForwardAccessSignalling,
   PrepareSubsequentHandover,

   -- authentication management operations
   SendAuthenticationInfo,
   AuthenticationFailureReport,

   -- IMEI management operations
   CheckIMEI,

   -- subscriber management operations
   InsertSubscriberData,
   DeleteSubscriberData,

   -- fault recovery operations
   Reset,
   ForwardCheckSS-Indication,
   RestoreData,

-- gprs location information retrieval operations
   SendRoutingInfoForGprs,

   -- failure reporting operations
   FailureReport,

   -- gprs notification operations
   NoteMsPresentForGprs,

   -- Mobility Management operations
   NoteMM-Event




;

IMPORTS
   OPERATION
FROM TCAPMessages {
   ccitt recommendation q 773 modules (2) messages (1) version2 (2)}

   SystemFailure,
   DataMissing,
   UnexpectedDataValue,
   UnknownSubscriber,
   UnknownMSC,
   UnidentifiedSubscriber,
   UnknownEquipment,
   RoamingNotAllowed,
```

```
   ATI-NotAllowed,
   NoHandoverNumberAvailable,
   SubsequentHandoverFailure,
   AbsentSubscriber,
   MM-EventNotSupported,
   ATSI-NotAllowed,
   ATM-NotAllowed,
   BearerServiceNotProvisioned,
   TeleserviceNotProvisioned,
   CallBarred,
   IllegalSS-Operation,
   SS-ErrorStatus,
   SS-NotAvailable,
   SS-Incompatibility,
   SS-SubscriptionViolation,
   InformationNotAvailable,
   TargetCellOutsideGroupCallArea


FROM MAP-Errors {
   ccitt identified-organization (4) etsi (0) mobileDomain (0)
   gsm-Network (1) modules (3) map-Errors (10) version6 (6)}

   UpdateLocationArg,
   UpdateLocationRes,
   CancelLocationArg,
   CancelLocationRes,
   PurgeMS-Arg,
   PurgeMS-Res,
   SendIdentificationArg,
   SendIdentificationRes,
   UpdateGprsLocationArg,
   UpdateGprsLocationRes,
   PrepareHO-Arg,
   PrepareHO-Res,
   ForwardAccessSignalling-Arg,
   ProcessAccessSignalling-Arg,
   SendEndSignal-Arg,
   SendEndSignal-Res,
   PrepareSubsequentHO-Res,
   PrepareSubsequentHO-Arg,
   SendAuthenticationInfoArg,
   SendAuthenticationInfoRes,
   AuthenticationFailureReportArg,
   AuthenticationFailureReportRes,
   EquipmentStatus,
   InsertSubscriberDataArg,
   InsertSubscriberDataRes,
   DeleteSubscriberDataArg,
   DeleteSubscriberDataRes,
   ResetArg,
   RestoreDataArg,
   RestoreDataRes,
   ProvideSubscriberInfoArg,
   ProvideSubscriberInfoRes,
   AnyTimeSubscriptionInterrogationArg,
   AnyTimeSubscriptionInterrogationRes,
   AnyTimeModificationArg,
   AnyTimeModificationRes,
   NoteSubscriberDataModifiedArg,
   NoteSubscriberDataModifiedRes,
   AnyTimeInterrogationArg,
   AnyTimeInterrogationRes,
   SendRoutingInfoForGprsArg,
   SendRoutingInfoForGprsRes,
   FailureReportArg,
   FailureReportRes,
   NoteMsPresentForGprsArg,
   NoteMsPresentForGprsRes,
   NoteMM-EventArg,
   NoteMM-EventRes


FROM MAP-MS-DataTypes {
   ccitt identified-organization (4) etsi (0) mobileDomain (0)
   gsm-Network (1) modules (3) map-MS-DataTypes (11) version6 (6)}

   IMEI
FROM MAP-CommonDataTypes {
```

```
   ccitt identified-organization (4) etsi (0) mobileDomain (0)
   gsm-Network (1) modules (3) map-CommonDataTypes (18) version6 (6)}
;


-- location registration operations

UpdateLocation ::= OPERATION                                          --Timer m
     ARGUMENT
          updateLocationArg               UpdateLocationArg
     RESULT
          updateLocationRes               UpdateLocationRes
     ERRORS {
          SystemFailure,
          DataMissing,
          UnexpectedDataValue,
          UnknownSubscriber,
          RoamingNotAllowed}

CancelLocation ::= OPERATION                                          --Timer m
     ARGUMENT
          cancelLocationArg               CancelLocationArg
     RESULT
          cancelLocationRes               CancelLocationRes
               -- optional
     ERRORS {
          DataMissing,
          UnexpectedDataValue}

PurgeMS ::= OPERATION                                                 --Timer m
     ARGUMENT
          purgeMS-Arg                     PurgeMS-Arg
     RESULT
          purgeMS-Res                     PurgeMS-Res
               -- optional
     ERRORS{
          DataMissing,
          UnexpectedDataValue,
          UnknownSubscriber}

SendIdentification ::= OPERATION                                      --Timer s
     ARGUMENT
          sendIdentificationArg                SendIdentificationArg
     RESULT
          sendIdentificationRes           SendIdentificationRes
     ERRORS {
          DataMissing,
          UnidentifiedSubscriber}

-- gprs location registration operations

UpdateGprsLocation ::= OPERATION                                      --Timer m
     ARGUMENT
          updateGprsLocationArg           UpdateGprsLocationArg
     RESULT
          updateGprsLocationRes           UpdateGprsLocationRes
     ERRORS {
          SystemFailure,
          UnexpectedDataValue,
          UnknownSubscriber,
          RoamingNotAllowed}

-- subscriber information enquiry operations

ProvideSubscriberInfo ::= OPERATION                                   --Timer m
     ARGUMENT
          provideSubscriberInfoArg        ProvideSubscriberInfoArg
     RESULT
          provideSubscriberInfoRes        ProvideSubscriberInfoRes
     ERRORS {
          DataMissing,
          UnexpectedDataValue}

-- any time information enquiry operations
```

```
AnyTimeInterrogation ::= OPERATION                                    --Timer m
     ARGUMENT
          anyTimeInterrogationArg         AnyTimeInterrogationArg
     RESULT
          anyTimeInterrogationRes         AnyTimeInterrogationRes
     ERRORS {
          SystemFailure,
          ATI-NotAllowed,
          DataMissing,
          UnexpectedDataValue,
          UnknownSubscriber}
```

-- any time information handling operations

```
AnyTimeSubscriptionInterrogation ::= OPERATION                         --Timer m
     ARGUMENT
          anyTimeSubscriptionInterrogationArg  AnyTimeSubscriptionInterrogationArg
     RESULT
          anyTimeSubscriptionInterrogationRes  AnyTimeSubscriptionInterrogationRes
     ERRORS {
          ATSI-NotAllowed,
          DataMissing,
          UnexpectedDataValue,
          UnknownSubscriber,
          BearerServiceNotProvisioned,
          TeleserviceNotProvisioned,
          CallBarred,
          IllegalSS-Operation,
          SS-NotAvailable,
          InformationNotAvailable}
```

```
AnyTimeModification ::= OPERATION                                     --Timer m
     ARGUMENT
          anyTimeModificationArg          AnyTimeModificationArg
     RESULT
          anyTimeModificationRes          AnyTimeModificationRes
     ERRORS {
          ATM-NotAllowed,
          DataMissing,
          UnexpectedDataValue,
          UnknownSubscriber,
          BearerServiceNotProvisioned,
          TeleserviceNotProvisioned,
          CallBarred,
          IllegalSS-Operation,
          SS-SubscriptionViolation,
          SS-ErrorStatus,
          SS-Incompatibility,
          InformationNotAvailable}
```

-- subscriber data modification notification operations

```
NoteSubscriberDataModified ::= OPERATION                              --Timer m
     ARGUMENT
          noteSubscriberDataModifiedArg   NoteSubscriberDataModifiedArg
     RESULT
          noteSubscriberDataModifiedRes   NoteSubscriberDataModifiedRes
               -- optional
     ERRORS {
          DataMissing,
          UnexpectedDataValue,
          UnknownSubscriber}
```

-- handover operations

```
PrepareHandover ::= OPERATION                                         --Timer m
     ARGUMENT
          prepareHO-Arg                   PrepareHO-Arg
     RESULT
          prepareHO-Res                   PrepareHO-Res
     ERRORS {
          SystemFailure,
          DataMissing,
          UnexpectedDataValue,
          NoHandoverNumberAvailable,
          TargetCellOutsideGroupCallArea }
```

```
SendEndSignal ::= OPERATION                                          --Timer l
     ARGUMENT
          sendEndSignal-Arg              SendEndSignal-Arg
     RESULT
          sendEndSignal-Res              SendEndSignal-Res
```

```
ProcessAccessSignalling ::= OPERATION                                --Timer s
     ARGUMENT
          processAccessSignalling-Arg    ProcessAccessSignalling-Arg
```

```
ForwardAccessSignalling ::= OPERATION                                --Timer s
     ARGUMENT
          forwardAccessSignalling-Arg    ForwardAccessSignalling-Arg
```

```
PrepareSubsequentHandover ::= OPERATION                              --Timer m
     ARGUMENT
          prepareSubsequentHO-Arg        PrepareSubsequentHO-Arg
     RESULT
          prepareSubsequentHO-Res        PrepareSubsequentHO-Res
     ERRORS {
          UnexpectedDataValue,
          DataMissing,
          UnknownMSC,
          SubsequentHandoverFailure}
```

-- authentication management operations

```
SendAuthenticationInfo ::= OPERATION                                 --Timer m
     ARGUMENT
          sendAuthenticationInfoArg      SendAuthenticationInfoArg
          -- optional
          -- within a dialogue sendAuthenticationInfoArg shall not be present in
          -- subsequent invoke components. If received in a subsequent invoke component
          -- it shall be discarded.

     RESULT
          sendAuthenticationInfoRes      SendAuthenticationInfoRes
          -- optional
     ERRORS {
          SystemFailure,
          DataMissing,
          UnexpectedDataValue,
          UnknownSubscriber}
```

```
AuthenticationFailureReport ::= OPERATION          --Timer m
     ARGUMENT
          authenticationFailureReportArg  AuthenticationFailureReportArg
     RESULT
          authenticationFailureReportRes  AuthenticationFailureReportRes
          -- optional
     ERRORS {
          SystemFailure,
          UnexpectedDataValue,
          UnknownSubscriber}
```

-- IMEI management operations

```
CheckIMEI ::= OPERATION                                              --Timer m
     ARGUMENT
          imei                           IMEI
     RESULT
          equipmentStatus                EquipmentStatus
     ERRORS {
          SystemFailure,
          DataMissing,
          UnknownEquipment}
```

-- subscriber management operations

```
InsertSubscriberData ::= OPERATION                                    --Timer m
     ARGUMENT
          insertSubscriberDataArg         InsertSubscriberDataArg
     RESULT
          insertSubscriberDataRes         InsertSubscriberDataRes
          -- optional
     ERRORS {
          DataMissing,
          UnexpectedDataValue,
          UnidentifiedSubscriber}
```

```
DeleteSubscriberData ::= OPERATION                                    --Timer m
     ARGUMENT
          deleteSubscriberDataArg         DeleteSubscriberDataArg
     RESULT
          deleteSubscriberDataRes         DeleteSubscriberDataRes
          -- optional
     ERRORS {
          DataMissing,
          UnexpectedDataValue,
          UnidentifiedSubscriber}
```

-- fault recovery operations

```
Reset ::= OPERATION                                                   --Timer m
     ARGUMENT
          resetArg                        ResetArg
```

```
ForwardCheckSS-Indication ::= OPERATION            --Timer s
```

```
RestoreData ::= OPERATION                                             --Timer m
     ARGUMENT
          restoreDataArg                  RestoreDataArg
     RESULT
          restoreDataRes                  RestoreDataRes
     ERRORS {
          SystemFailure,
          DataMissing,
          UnexpectedDataValue,
          UnknownSubscriber}
```

-- gprs location information retrieval operations

```
SendRoutingInfoForGprs ::= OPERATION                                  --Timer m
     ARGUMENT
          sendRoutingInfoForGprsArg       SendRoutingInfoForGprsArg
     RESULT
          sendRoutingInfoForGprsRes       SendRoutingInfoForGprsRes
     ERRORS {
          AbsentSubscriber,
          SystemFailure,
          DataMissing,
          UnexpectedDataValue,
          UnknownSubscriber}
```

-- failure reporting operations

```
FailureReport ::= OPERATION                                           --Timer m
     ARGUMENT
          failureReportArg                FailureReportArg
     RESULT
          failureReportRes                FailureReportRes
               -- optional
     ERRORS {
          SystemFailure,
          DataMissing,
          UnexpectedDataValue,
          UnknownSubscriber}
```

-- gprs notification operations

```
NoteMsPresentForGprs ::= OPERATION                                          --Timer m
     ARGUMENT
          noteMsPresentForGprsArg         NoteMsPresentForGprsArg
     RESULT
          noteMsPresentForGprsRes         NoteMsPresentForGprsRes
               -- optional
     ERRORS {
          SystemFailure,
          DataMissing,
          UnexpectedDataValue,
          UnknownSubscriber}
```

```
NoteMM-Event ::= OPERATION                                                  --Timer m
     ARGUMENT
          noteMM-EventArg                 NoteMM-EventArg
     RESULT
          noteMM-EventRes                 NoteMM-EventRes
     ERRORS {
          DataMissing,
          UnexpectedDataValue,
          UnknownSubscriber,
          MM-EventNotSupported}
```

```
END
```

17.6.2 Operation and Maintenance Operations

```
MAP-OperationAndMaintenanceOperations {
   ccitt identified-organization (4) etsi (0) mobileDomain (0)
   gsm-Network (1) modules (3) map-OperationAndMaintenanceOperations (6)
   version6 (6)}

DEFINITIONS

::=

BEGIN

EXPORTS
   ActivateTraceMode,
   DeactivateTraceMode,
   SendIMSI
;

IMPORTS
   OPERATION
FROM TCAPMessages {
   ccitt recommendation q 773 modules (2) messages (1) version2 (2)}

   SystemFailure,
   DataMissing,
   UnexpectedDataValue,
   FacilityNotSupported,
   UnknownSubscriber,
   UnidentifiedSubscriber,
   TracingBufferFull
FROM MAP-Errors {
   ccitt identified-organization (4) etsi (0) mobileDomain (0)
   gsm-Network (1) modules (3) map-Errors (10) version6 (6)}

   ActivateTraceModeArg,
   ActivateTraceModeRes,
   DeactivateTraceModeArg,
   DeactivateTraceModeRes
FROM MAP-OM-DataTypes {
   ccitt identified-organization (4) etsi (0) mobileDomain (0)
   gsm-Network (1) modules (3) map-OM-DataTypes (12) version6 (6)}

   ISDN-AddressString,
   IMSI
FROM MAP-CommonDataTypes {
   ccitt identified-organization (4) etsi (0) mobileDomain (0)
   gsm-Network (1) modules (3) map-CommonDataTypes (18) version6 (6)}
;
```

```
ActivateTraceMode ::= OPERATION                                          --Timer m
     ARGUMENT
          activateTraceModeArg            ActivateTraceModeArg
     RESULT
          activateTraceModeRes            ActivateTraceModeRes
          -- optional
     ERRORS {
          SystemFailure,
          DataMissing,
          UnexpectedDataValue,
          FacilityNotSupported,
          UnidentifiedSubscriber,
          TracingBufferFull}
```

```
DeactivateTraceMode ::= OPERATION                                        --Timer m
     ARGUMENT
          deactivateTraceModeArg          DeactivateTraceModeArg
     RESULT
          deactivateTraceModeRes          DeactivateTraceModeRes
          -- optional
     ERRORS {
          SystemFailure,
          DataMissing,
          UnexpectedDataValue,
          FacilityNotSupported,
          UnidentifiedSubscriber}
```

```
SendIMSI ::= OPERATION                                                   --Timer m
     ARGUMENT
          msisdn                          ISDN-AddressString
     RESULT
          imsi                            IMSI
     ERRORS {
          DataMissing,
          UnexpectedDataValue,
          UnknownSubscriber}
```

```
END
```

17.6.3 Call Handling Operations

```
MAP-CallHandlingOperations {
   ccitt identified-organization (4) etsi (0) mobileDomain (0)
   gsm-Network (1) modules (3) map-CallHandlingOperations (7)
   version6 (6)}

DEFINITIONS

::=

BEGIN

EXPORTS
   SendRoutingInfo,
   ProvideRoamingNumber,
   ResumeCallHandling,
   ProvideSIWFSNumber,
   SIWFSSignallingModify,
   SetReportingState,
   StatusReport,
   RemoteUserFree,
   IST-Alert,
   IST-Command
;

IMPORTS
   OPERATION
FROM TCAPMessages {
   ccitt recommendation q 773 modules (2) messages (1) version2 (2)}

   SystemFailure,
   DataMissing,
   UnexpectedDataValue,
   FacilityNotSupported,
   OR-NotAllowed,
   UnknownSubscriber,
   NumberChanged,
   BearerServiceNotProvisioned,
```

```
   TeleserviceNotProvisioned,
   NoRoamingNumberAvailable,
   AbsentSubscriber,
   BusySubscriber,
   NoSubscriberReply,
   CallBarred,
   ForwardingViolation,
   ForwardingFailed,
   CUG-Reject,
   ResourceLimitation,
   IncompatibleTerminal,
   UnidentifiedSubscriber

FROM MAP-Errors {
   ccitt identified-organization (4) etsi (0) mobileDomain (0)
   gsm-Network (1) modules (3) map-Errors (10) version6 (6)}
   SendRoutingInfoArg,
   SendRoutingInfoRes,
   ProvideRoamingNumberArg,
   ProvideRoamingNumberRes,
   ResumeCallHandlingArg,
   ResumeCallHandlingRes,
   ProvideSIWFSNumberArg,
   ProvideSIWFSNumberRes,
   SIWFSSignallingModifyArg,
   SIWFSSignallingModifyRes,
   SetReportingStateArg,
   SetReportingStateRes,
   StatusReportArg,
   StatusReportRes,
   RemoteUserFreeArg,
   RemoteUserFreeRes,
   IST-AlertArg,
   IST-AlertRes,
   IST-CommandArg,
IST-CommandRes
FROM MAP-CH-DataTypes {
   ccitt identified-organization (4) etsi (0) mobileDomain (0)
   gsm-Network (1) modules (3) map-CH-DataTypes (13) version6 (6)}

;
```

```
SendRoutingInfo ::= OPERATION                                          --Timer m
-- The timer is set to the upper limit of the range if the GMSC supports pre-paging.
     ARGUMENT
          sendRoutingInfoArg              SendRoutingInfoArg
     RESULT
          sendRoutingInfoRes              SendRoutingInfoRes
     ERRORS {
          SystemFailure,
          DataMissing,
          UnexpectedDataValue,
          FacilityNotSupported,
          OR-NotAllowed,
          UnknownSubscriber,
          NumberChanged,
          BearerServiceNotProvisioned,
          TeleserviceNotProvisioned,
          AbsentSubscriber,
          BusySubscriber,
          NoSubscriberReply,
          CallBarred,
          CUG-Reject,
          ForwardingViolation}
```

```
ProvideRoamingNumber ::= OPERATION                                    --Timer m
-- The timer is set to the upper limit of the range if the HLR supports pre-paging.
     ARGUMENT
          provideRoamingNumberArg         ProvideRoamingNumberArg
     RESULT
          provideRoamingNumberRes         ProvideRoamingNumberRes
     ERRORS {
          SystemFailure,
          DataMissing,
          UnexpectedDataValue,
          FacilityNotSupported,
          OR-NotAllowed,
          AbsentSubscriber,
          NoRoamingNumberAvailable}
```

```
ResumeCallHandling ::= OPERATION                                      --Timer m
     ARGUMENT
          resumeCallHandlingArg           ResumeCallHandlingArg
     RESULT
          resumeCallHandlingRes           ResumeCallHandlingRes
          -- optional
     ERRORS {
          ForwardingFailed,
          OR-NotAllowed,
          UnexpectedDataValue,
          DataMissing }
```

```
ProvideSIWFSNumber ::= OPERATION                                      --Timer m
     ARGUMENT
          provideSIWFSNumberArg           ProvideSIWFSNumberArg
     RESULT
          provideSIWFSNumberRes           ProvideSIWFSNumberRes
     ERRORS {
          ResourceLimitation,
          DataMissing,
          UnexpectedDataValue,
          SystemFailure}
```

```
SIWFSSignallingModify ::= OPERATION                                   --Timer m
     ARGUMENT
          sIWFSSignallingModifyArg        SIWFSSignallingModifyArg
     RESULT
          sIWFSSignallingModifyRes        SIWFSSignallingModifyRes
          -- optional
     ERRORS {
          ResourceLimitation,
          DataMissing,
          UnexpectedDataValue,
          SystemFailure}
```

```
SetReportingState ::= OPERATION                                       --Timer m
     ARGUMENT
          setReportingStateArg            SetReportingStateArg
     RESULT
          setReportingStateRes            SetReportingStateRes
          -- optional
     ERRORS {
          SystemFailure,
          UnidentifiedSubscriber,
          UnexpectedDataValue,
          DataMissing,
          ResourceLimitation,
          FacilityNotSupported}
```

```
StatusReport ::= OPERATION                                            --Timer m
     ARGUMENT
          statusReportArg                 StatusReportArg
     RESULT
          statusReportRes                 StatusReportRes
          -- optional
     ERRORS {
          UnknownSubscriber,
          SystemFailure,
          UnexpectedDataValue,
          DataMissing}
```

```
RemoteUserFree ::= OPERATION                                          --Timer ml
     ARGUMENT
          remoteUserFreeArg               RemoteUserFreeArg
     RESULT
          remoteUserFreeRes               RemoteUserFreeRes
     ERRORS {
          UnexpectedDataValue,
          DataMissing,
          IncompatibleTerminal,
          AbsentSubscriber,
          SystemFailure,
          BusySubscriber}
```

```
IST-Alert ::= OPERATION                                               --Timer m
     ARGUMENT
          istAlertArg                     IST-AlertArg
     RESULT
          istAlertRes                     IST-AlertRes
          -- optional
     ERRORS {
          UnexpectedDataValue,
          ResourceLimitation,
          UnknownSubscriber,
          SystemFailure,
          FacilityNotSupported}
```

```
IST-Command ::= OPERATION                                             --Timer m
     ARGUMENT
          istCommandArg                   IST-CommandArg
     RESULT
          istCommandRes                   IST-CommandRes
          -- optional
     ERRORS {
          UnexpectedDataValue,
          ResourceLimitation,
          UnknownSubscriber,
          SystemFailure,
          FacilityNotSupported}
```

```
END
```

17.6.4 Supplementary service operations

```
MAP-SupplementaryServiceOperations {
   ccitt identified-organization (4) etsi (0) mobileDomain (0)
   gsm-Network (1) modules (3) map-SupplementaryServiceOperations (8)
   version6 (6)}

DEFINITIONS

::=

BEGIN

EXPORTS
   RegisterSS,
   EraseSS,
   ActivateSS,
   DeactivateSS,
   InterrogateSS,
   ProcessUnstructuredSS-Request,
   UnstructuredSS-Request,
   UnstructuredSS-Notify,
   RegisterPassword,
   GetPassword,
   SS-InvocationNotification,
   RegisterCC-Entry,
   EraseCC-Entry
;

IMPORTS
   OPERATION
FROM TCAPMessages {
   ccitt recommendation q 773 modules (2) messages (1) version2 (2)}

   SystemFailure,
   DataMissing,
   UnexpectedDataValue,
```

```
   UnknownSubscriber,
   BearerServiceNotProvisioned,
   TeleserviceNotProvisioned,
   CallBarred,
   IllegalSS-Operation,
   SS-ErrorStatus,
   SS-NotAvailable,
   SS-SubscriptionViolation,
   SS-Incompatibility,
   PW-RegistrationFailure,
   NegativePW-Check,
   NumberOfPW-AttemptsViolation,
   UnknownAlphabet,
   USSD-Busy,
   AbsentSubscriber,
   IllegalSubscriber,
   IllegalEquipment,
   ShortTermDenial,
   LongTermDenial,
   FacilityNotSupported
FROM MAP-Errors {
   ccitt identified-organization (4) etsi (0) mobileDomain (0)
   gsm-Network (1) modules (3) map-Errors (10) version6 (6)}

   RegisterSS-Arg,
   SS-Info,
   SS-ForBS-Code,
   InterrogateSS-Res,
   USSD-Arg,
   USSD-Res,
   Password,
   GuidanceInfo,
   SS-InvocationNotificationArg,
   SS-InvocationNotificationRes,
   RegisterCC-EntryArg,
   RegisterCC-EntryRes,
   EraseCC-EntryArg,
   EraseCC-EntryRes
FROM MAP-SS-DataTypes {
   ccitt identified-organization (4) etsi (0) mobileDomain (0)
   gsm-Network (1) modules (3) map-SS-DataTypes (14) version6 (6)}

   SS-Code
FROM MAP-SS-Code {
   ccitt identified-organization (4) etsi (0) mobileDomain (0)
   gsm-Network (1) modules (3) map-SS-Code (15) version6 (6)}
;


-- supplementary service handling operations

RegisterSS ::= OPERATION                                          --Timer m
     ARGUMENT
          registerSS-Arg                  RegisterSS-Arg
     RESULT
          ss-Info                         SS-Info
          -- optional
     ERRORS {
          SystemFailure,
          DataMissing,
          UnexpectedDataValue,
          BearerServiceNotProvisioned,
          TeleserviceNotProvisioned,
          CallBarred,
          IllegalSS-Operation,
          SS-ErrorStatus,
          SS-Incompatibility}
```

```
EraseSS ::= OPERATION                                                    --Timer m
     ARGUMENT
          ss-ForBS                        SS-ForBS-Code
     RESULT
          ss-Info                         SS-Info
          -- optional
     ERRORS {
          SystemFailure,
          DataMissing,
          UnexpectedDataValue,
          BearerServiceNotProvisioned,
          TeleserviceNotProvisioned,
          CallBarred,
          IllegalSS-Operation,
          SS-ErrorStatus
          }
```

```
ActivateSS ::= OPERATION                                                 --Timer m
     ARGUMENT
          ss-ForBS                        SS-ForBS-Code
     RESULT
          ss-Info                         SS-Info
          -- optional
     ERRORS {
          SystemFailure,
          DataMissing,
          UnexpectedDataValue,
          BearerServiceNotProvisioned,
          TeleserviceNotProvisioned,
          CallBarred,
          IllegalSS-Operation,
          SS-ErrorStatus,
          SS-SubscriptionViolation,
          SS-Incompatibility,
          NegativePW-Check,
          NumberOfPW-AttemptsViolation}
```

```
DeactivateSS ::= OPERATION                                               --Timer m
     ARGUMENT
          ss-ForBS                        SS-ForBS-Code
     RESULT
          ss-Info                         SS-Info
          -- optional
     ERRORS {
          SystemFailure,
          DataMissing,
          UnexpectedDataValue,
          BearerServiceNotProvisioned,
          TeleserviceNotProvisioned,
          CallBarred,
          IllegalSS-Operation,
          SS-ErrorStatus,
          SS-SubscriptionViolation,
          NegativePW-Check,
          NumberOfPW-AttemptsViolation}
```

```
InterrogateSS ::= OPERATION                                              --Timer m
     ARGUMENT
          ss-ForBS                        SS-ForBS-Code
     RESULT
          interrogateSS-Res               InterrogateSS-Res
     ERRORS {
          SystemFailure,
          DataMissing,
          UnexpectedDataValue,
          BearerServiceNotProvisioned,
          TeleserviceNotProvisioned,
          CallBarred,
          IllegalSS-Operation,
          SS-NotAvailable}
```

```
ProcessUnstructuredSS-Request ::= OPERATION              --Timer 10 minutes
     ARGUMENT
          ussd-Arg                          USSD-Arg
     RESULT
          ussd-Res                          USSD-Res
     ERRORS {
          SystemFailure,
          DataMissing,
          UnexpectedDataValue,
          UnknownAlphabet,
          CallBarred}
```

```
UnstructuredSS-Request ::= OPERATION                                  --Timer ml
     ARGUMENT
          ussd-Arg                          USSD-Arg
     RESULT
          ussd-Res                          USSD-Res
          -- optional
     ERRORS {
          SystemFailure,
          DataMissing,
          UnexpectedDataValue,
          AbsentSubscriber,
          IllegalSubscriber,
          IllegalEquipment,
          UnknownAlphabet,
          USSD-Busy}
```

```
UnstructuredSS-Notify ::= OPERATION                                   --Timer ml
     ARGUMENT
          ussd-Arg                          USSD-Arg
     RESULT
     ERRORS {
          SystemFailure,
          DataMissing,
          UnexpectedDataValue,
          AbsentSubscriber,
          IllegalSubscriber,
          IllegalEquipment,
          UnknownAlphabet,
          USSD-Busy}
```

```
RegisterPassword ::= OPERATION                                        --Timer ml
     ARGUMENT
          ss-Code                           SS-Code
     RESULT
          newPassword                       Password
     ERRORS {
          SystemFailure,
          DataMissing,
          UnexpectedDataValue,
          CallBarred,
          SS-SubscriptionViolation,
          PW-RegistrationFailure,
          NegativePW-Check,
          NumberOfPW-AttemptsViolation}
     LINKED {
          GetPassword}
```

```
GetPassword ::= OPERATION                                             --Timer m
     ARGUMENT
          guidanceInfo                      GuidanceInfo
     RESULT
          currentPassword                   Password
```

```
SS-InvocationNotification ::= OPERATION                               --Timer m
     ARGUMENT
          ss-InvocationNotificationArg         SS-InvocationNotificationArg
     RESULT
          ss-InvocationNotificationRes         SS-InvocationNotificationRes
          -- optional
     ERRORS {
          DataMissing,
          UnexpectedDataValue,
          UnknownSubscriber}
```

```
RegisterCC-Entry ::= OPERATION                                          --Timer m
     ARGUMENT
          registerCC-EntryArg             RegisterCC-EntryArg
     RESULT
          registerCC-EntryRes             RegisterCC-EntryRes
     ERRORS {
          SystemFailure,
          DataMissing,
          UnexpectedDataValue,
          CallBarred,
          IllegalSS-Operation,
          SS-ErrorStatus,
          SS-Incompatibility,
          ShortTermDenial,
          LongTermDenial,
          FacilityNotSupported}
```

```
EraseCC-Entry ::= OPERATION                                             --Timer m
     ARGUMENT
          eraseCC-EntryArg                EraseCC-EntryArg
     RESULT
          eraseCC-EntryRes                EraseCC-EntryRes
     ERRORS {
          SystemFailure,
          DataMissing,
          UnexpectedDataValue,
          CallBarred,
          IllegalSS-Operation,
          SS-ErrorStatus}
```

```
END
```

17.6.5 Short message service operations

```
MAP-ShortMessageServiceOperations {
   ccitt identified-organization (4) etsi (0) mobileDomain (0)
   gsm-Network (1) modules (3) map-ShortMessageServiceOperations (9)
   version6 (6)}

DEFINITIONS

::=

BEGIN

EXPORTS
   SendRoutingInfoForSM,
   MO-ForwardSM,
   MT-ForwardSM,
   ReportSM-DeliveryStatus,
   AlertServiceCentre,
   InformServiceCentre,
   ReadyForSM
;

IMPORTS
   OPERATION
FROM TCAPMessages {
   ccitt recommendation q 773 modules (2) messages (1) version2 (2)}

   SystemFailure,
   DataMissing,
   UnexpectedDataValue,
   FacilityNotSupported,
   UnknownSubscriber,
   UnidentifiedSubscriber,
   IllegalSubscriber,
   IllegalEquipment,
   TeleserviceNotProvisioned,
   CallBarred,
   SubscriberBusyForMT-SMS,
   SM-DeliveryFailure,
   MessageWaitingListFull,
   AbsentSubscriberSM
FROM MAP-Errors {
   ccitt identified-organization (4) etsi (0) mobileDomain (0)
   gsm-Network (1) modules (3) map-Errors (10) version6 (6)}
```

```
   RoutingInfoForSM-Arg,
   RoutingInfoForSM-Res,
   MO-ForwardSM-Arg,
   MO-ForwardSM-Res,
   MT-ForwardSM-Arg,
   MT-ForwardSM-Res,
   ReportSM-DeliveryStatusArg,
   ReportSM-DeliveryStatusRes,
   AlertServiceCentreArg,
   InformServiceCentreArg,
   ReadyForSM-Arg,
   ReadyForSM-Res
FROM MAP-SM-DataTypes {
   ccitt identified-organization (4) etsi (0) mobileDomain (0)
   gsm-Network (1) modules (3) map-SM-DataTypes (16) version6 (6)}



;
```

```
SendRoutingInfoForSM ::= OPERATION                                    --Timer m
     ARGUMENT
          routingInfoForSM-Arg            RoutingInfoForSM-Arg
     RESULT
          routingInfoForSM-Res            RoutingInfoForSM-Res
     ERRORS {
          SystemFailure,
          DataMissing,
          UnexpectedDataValue,
          FacilityNotSupported,
          UnknownSubscriber,
          TeleserviceNotProvisioned,
          CallBarred,
          AbsentSubscriberSM}
```

```
MO-ForwardSM ::= OPERATION                                            --Timer ml
     ARGUMENT
          mo-forwardSM-Arg                MO-ForwardSM-Arg
     RESULT
          mo-forwardSM-Res                MO-ForwardSM-Res
          -- optional
     ERRORS {
          SystemFailure,
          UnexpectedDataValue,
          FacilityNotSupported,
          SM-DeliveryFailure}
```

```
MT-ForwardSM ::= OPERATION                                            --Timer ml
     ARGUMENT
          mt-forwardSM-Arg                MT-ForwardSM-Arg
     RESULT
          mt-forwardSM-Res                MT-ForwardSM-Res
          -- optional
     ERRORS {
          SystemFailure,
          DataMissing,
          UnexpectedDataValue,
          FacilityNotSupported,
          UnidentifiedSubscriber,
          IllegalSubscriber,
          IllegalEquipment,
          SubscriberBusyForMT-SMS,
          SM-DeliveryFailure,
          AbsentSubscriberSM}
```

```
ReportSM-DeliveryStatus ::= OPERATION              --Timer s
     ARGUMENT
          reportSM-DeliveryStatusArg      ReportSM-DeliveryStatusArg
     RESULT
          reportSM-DeliveryStatusRes      ReportSM-DeliveryStatusRes
          -- optional
     ERRORS {
          DataMissing,
          UnexpectedDataValue,
          UnknownSubscriber,
          MessageWaitingListFull}
```

```
AlertServiceCentre ::= OPERATION                                        --Timer s
     ARGUMENT
          alertServiceCentreArg          AlertServiceCentreArg
     RESULT
     ERRORS {
          SystemFailure,
          DataMissing,
          UnexpectedDataValue}
```

```
InformServiceCentre ::= OPERATION                                       --Timer s
     ARGUMENT
          informServiceCentreArg         InformServiceCentreArg
```

```
ReadyForSM ::= OPERATION                                                --Timer m
     ARGUMENT
          readyForSM-Arg                 ReadyForSM-Arg
     RESULT
          readyForSM-Res                 ReadyForSM-Res
          -- optional
     ERRORS {
          DataMissing,
          UnexpectedDataValue,
          FacilityNotSupported,
          UnknownSubscriber}
```

```
END
```

17.6.6 Errors

```
MAP-Errors {
   ccitt identified-organization (4) etsi (0) mobileDomain (0)
   gsm-Network (1) modules (3) map-Errors (10) version6 (6)}

DEFINITIONS

::=

BEGIN

EXPORTS

   -- generic errors
   SystemFailure,
   DataMissing,
   UnexpectedDataValue,
   FacilityNotSupported,
   IncompatibleTerminal,
   ResourceLimitation,

   -- identification and numbering errors
   UnknownSubscriber,
   NumberChanged,
   UnknownMSC,
   UnidentifiedSubscriber,
   UnknownEquipment,

   -- subscription errors
   RoamingNotAllowed,
   IllegalSubscriber,
   IllegalEquipment,
   BearerServiceNotProvisioned,
   TeleserviceNotProvisioned,

   -- handover errors
   NoHandoverNumberAvailable,
   SubsequentHandoverFailure,
   TargetCellOutsideGroupCallArea,

   -- operation and maintenance errors
   TracingBufferFull,

   -- call handling errors
   OR-NotAllowed,
   NoRoamingNumberAvailable,
   BusySubscriber,
   NoSubscriberReply,
   AbsentSubscriber,
   CallBarred,
```

```
   ForwardingViolation,
   ForwardingFailed,
   CUG-Reject,

   -- any time interrogation errors
   ATI-NotAllowed,

   -- any time information handling errors
   ATSI-NotAllowed,
   ATM-NotAllowed,
   InformationNotAvailable,

   -- supplementary service errors
   IllegalSS-Operation,
   SS-ErrorStatus,
   SS-NotAvailable,
   SS-SubscriptionViolation,
   SS-Incompatibility,
   UnknownAlphabet,
   USSD-Busy,
   PW-RegistrationFailure,
   NegativePW-Check,
   NumberOfPW-AttemptsViolation,
   ShortTermDenial,
   LongTermDenial,

   -- short message service errors
   SubscriberBusyForMT-SMS,
   SM-DeliveryFailure,
   MessageWaitingListFull,
   AbsentSubscriberSM,

   -- Group Call errors
   NoGroupCallNumberAvailable,

   -- location service errors
   UnauthorizedRequestingNetwork,
   UnauthorizedLCSClient,
   PositionMethodFailure,
   UnknownOrUnreachableLCSClient,

   -- Mobility Management errors
   MM-EventNotSupported


;

IMPORTS
   ERROR
FROM TCAPMessages {
   ccitt recommendation q 773 modules (2) messages (1) version2 (2)}

   SS-Status
FROM MAP-SS-DataTypes {
   ccitt identified-organization (4) etsi (0) mobileDomain (0)
   gsm-Network (1) modules (3) map-SS-DataTypes (14) version6 (6)}

   SS-IncompatibilityCause,
   PW-RegistrationFailureCause,
   SM-DeliveryFailureCause,
   SystemFailureParam,
   DataMissingParam,
   UnexpectedDataParam,
   FacilityNotSupParam,
   UnknownSubscriberParam,
   NumberChangedParam,
   UnidentifiedSubParam,
   RoamingNotAllowedParam,
   IllegalSubscriberParam,
   IllegalEquipmentParam,
   BearerServNotProvParam,
   TeleservNotProvParam,
   TracingBufferFullParam,
   NoRoamingNbParam,
   OR-NotAllowedParam,
   AbsentSubscriberParam,
   BusySubscriberParam,
   NoSubscriberReplyParam,
   CallBarredParam,
```

```
   ForwardingViolationParam,
   ForwardingFailedParam,
   CUG-RejectParam,
   ATI-NotAllowedParam,
   SubBusyForMT-SMS-Param,
   MessageWaitListFullParam,
   AbsentSubscriberSM-Param,
   ResourceLimitationParam,
   NoGroupCallNbParam,
   IncompatibleTerminalParam,
   ShortTermDenialParam,
   LongTermDenialParam,
   UnauthorizedRequestingNetwork-Param,
   UnauthorizedLCSClient-Param,
   PositionMethodFailure-Param,
   UnknownOrUnreachableLCSClient-Param,
   MM-EventNotSupported-Param,
   ATSI-NotAllowedParam,
   ATM-NotAllowedParam,
   IllegalSS-OperationParam,
   SS-NotAvailableParam,
   SS-SubscriptionViolationParam,
   InformationNotAvailableParam,
   TargetCellOutsideGCA-Param


FROM MAP-ER-DataTypes {
   ccitt identified-organization (4) etsi (0) mobileDomain (0)
   gsm-Network (1) modules (3) map-ER-DataTypes (17) version6 (6)}
;


-- generic errors

SystemFailure ::= ERROR
     PARAMETER
          systemFailureParam              SystemFailureParam
          -- optional

DataMissing ::= ERROR
     PARAMETER
          dataMissingParam                DataMissingParam
          -- optional
          -- dataMissingParam must not be used in version <3

UnexpectedDataValue ::= ERROR
     PARAMETER
          unexpectedDataParam             UnexpectedDataParam
          -- optional
          -- unexpectedDataParam must not be used in version <3

FacilityNotSupported ::= ERROR
     PARAMETER
          facilityNotSupParam             FacilityNotSupParam
          -- optional
          -- facilityNotSupParam must not be used in version <3

IncompatibleTerminal ::= ERROR
     PARAMETER
          incompatibleTerminalParam       IncompatibleTerminalParam
          -- optional

ResourceLimitation ::= ERROR
     PARAMETER
          resourceLimitationParam         ResourceLimitationParam
          -- optional

-- identification and numbering errors

UnknownSubscriber ::= ERROR
     PARAMETER
          unknownSubscriberParam          UnknownSubscriberParam
          -- optional
          -- unknownSubscriberParam must not be used in version <3
```

```
NumberChanged ::= ERROR
     PARAMETER
          numberChangedParam              NumberChangedParam
          -- optional
```

```
UnknownMSC ::= ERROR
```

```
UnidentifiedSubscriber ::= ERROR
     PARAMETER
          unidentifiedSubParam            UnidentifiedSubParam
          -- optional
          -- unidentifiedSubParam must not be used in version <3
```

```
UnknownEquipment ::= ERROR
```

-- subscription errors

```
RoamingNotAllowed ::= ERROR
     PARAMETER
          roamingNotAllowedParam          RoamingNotAllowedParam
```

```
IllegalSubscriber ::= ERROR
     PARAMETER
          illegalSubscriberParam          IllegalSubscriberParam
          -- optional
          -- illegalSubscriberParam must not be used in version <3
```

```
IllegalEquipment ::= ERROR
     PARAMETER
          illegalEquipmentParam           IllegalEquipmentParam
          -- optional
          -- illegalEquipmentParam must not be used in version <3
```

```
BearerServiceNotProvisioned ::= ERROR
     PARAMETER
          bearerServNotProvParam          BearerServNotProvParam
          -- optional
          -- bearerServNotProvParam must not be used in version <3
```

```
TeleserviceNotProvisioned ::= ERROR
     PARAMETER
          teleservNotProvParam            TeleservNotProvParam
          -- optional
          -- teleservNotProvParam must not be used in version <3
```

-- handover errors

```
NoHandoverNumberAvailable ::= ERROR
```

```
SubsequentHandoverFailure ::= ERROR
```

```
TargetCellOutsideGroupCallArea ::= ERROR
     PARAMETER
          targetCellOutsideGCA-Param      TargetCellOutsideGCA-Param
          -- optional
```

-- operation and maintenance errors

```
TracingBufferFull ::= ERROR
     PARAMETER
          tracingBufferFullParam          TracingBufferFullParam
          -- optional
```

-- call handling errors

```
NoRoamingNumberAvailable ::= ERROR
     PARAMETER
          noRoamingNbParam                NoRoamingNbParam
          -- optional
```

```
AbsentSubscriber ::= ERROR
     PARAMETER
          absentSubscriberParam          AbsentSubscriberParam
          -- optional
          -- absentSubscriberParam must not be used in version <3
```

```
BusySubscriber ::= ERROR
     PARAMETER
          busySubscriberParam            BusySubscriberParam
          -- optional
```

```
NoSubscriberReply ::= ERROR
     PARAMETER
          noSubscriberReplyParam         NoSubscriberReplyParam
          -- optional
```

```
CallBarred ::= ERROR
     PARAMETER
          callBarredParam                CallBarredParam
          -- optional
```

```
ForwardingViolation ::= ERROR
     PARAMETER
          forwardingViolationParam       ForwardingViolationParam
          -- optional
```

```
ForwardingFailed ::= ERROR
     PARAMETER
          forwardingFailedParam          ForwardingFailedParam
          -- optional
```

```
CUG-Reject ::= ERROR
     PARAMETER
          cug-RejectParam                CUG-RejectParam
          -- optional
```

```
OR-NotAllowed ::= ERROR
     PARAMETER
          or-NotAllowedParam             OR-NotAllowedParam
          -- optional
```

```
-- any time interrogation errors
ATI-NotAllowed ::= ERROR
     PARAMETER
          ati-NotAllowedParam            ATI-NotAllowedParam
          -- optional
```

```
-- any time information handling errors
ATSI-NotAllowed ::= ERROR
     PARAMETER
          atsi-NotAllowedParam           ATSI-NotAllowedParam
          -- optional
```

```
ATM-NotAllowed ::= ERROR
     PARAMETER
          atm-NotAllowedParam            ATM-NotAllowedParam
          -- optional
```

```
InformationNotAvailable ::= ERROR
     PARAMETER
          informationNotAvailableParam   InformationNotAvailableParam
          -- optional
```

```
-- supplementary service errors
```

```
IllegalSS-Operation ::= ERROR
     PARAMETER
          illegalSS-OperationParam       IllegalSS-OperationParam
          -- optional
          -- illegalSS-OperationParam must not be used in version <3
```

```
SS-ErrorStatus ::= ERROR
     PARAMETER
          ss-Status                      SS-Status
          -- optional
```

```
SS-NotAvailable ::= ERROR
     PARAMETER
          ss-NotAvailableParam           SS-NotAvailableParam
          -- optional
          -- ss-NotAvailableParam must not be used in version <3
```

```
SS-SubscriptionViolation ::= ERROR
     PARAMETER
          ss-SubscriptionViolationParam  SS-SubscriptionViolationParam
          -- optional
          -- ss-NotAvailableParam must not be used in version <3
```

```
SS-Incompatibility ::= ERROR
     PARAMETER
          ss-IncompatibilityCause        SS-IncompatibilityCause
          -- optional
```

```
UnknownAlphabet ::= ERROR
```

```
USSD-Busy ::= ERROR
```

```
PW-RegistrationFailure ::= ERROR
     PARAMETER
          pw-RegistrationFailureCause    PW-RegistrationFailureCause
```

```
NegativePW-Check ::= ERROR
```

```
NumberOfPW-AttemptsViolation ::= ERROR
```

```
ShortTermDenial ::= ERROR
     PARAMETER
          shortTermDenialParam                ShortTermDenialParam
          -- optional
```

```
LongTermDenial ::= ERROR
     PARAMETER
          longTermDenialParam                 LongTermDenialParam
          -- optional
```

```
-- short message service errors
```

```
SubscriberBusyForMT-SMS ::= ERROR
     PARAMETER
          subBusyForMT-SMS-Param         SubBusyForMT-SMS-Param
          -- optional
```

```
SM-DeliveryFailure ::= ERROR
     PARAMETER
          sm-DeliveryFailureCause        SM-DeliveryFailureCause
```

```
MessageWaitingListFull ::= ERROR
     PARAMETER
          messageWaitListFullParam       MessageWaitListFullParam
          -- optional
```

```
AbsentSubscriberSM ::= ERROR
     PARAMETER
          absentSubscriberSM-Param       AbsentSubscriberSM-Param
          -- optional
```

```
-- Group Call errors
```

```
NoGroupCallNumberAvailable ::= ERROR
     PARAMETER
          noGroupCallNbParam             NoGroupCallNbParam
          -- optional
```

```
-- location service errors
```

```
UnauthorizedRequestingNetwork ::= ERROR
     PARAMETER
          unauthorizedRequestingNetwork-Param  UnauthorizedRequestingNetwork-Param
          -- optional
```

```
UnauthorizedLCSClient ::= ERROR
     PARAMETER
          unauthorizedLCSClient-Param     UnauthorizedLCSClient-Param
          -- optional
```

```
PositionMethodFailure ::= ERROR
     PARAMETER
          positionMethodFailure-Param     PositionMethodFailure-Param
          -- optional
```

```
UnknownOrUnreachableLCSClient ::= ERROR
     PARAMETER
          unknownOrUnreachableLCSClient-Param   UnknownOrUnreachableLCSClient-Param
          -- optional
```

```
MM-EventNotSupported ::= ERROR
     PARAMETER
          mm-EventNotSupported-Param      MM-EventNotSupported-Param
          -- optional
```

```
END
```

17.6.7 Group Call operations

```
MAP-Group-Call-Operations {
   ccitt identified-organization (4) etsi (0) mobileDomain (0)
   gsm-Network (1) modules (3) map-Group-Call-Operations (22)
   version6 (6)}

DEFINITIONS

::=

BEGIN

EXPORTS
   PrepareGroupCall,
   SendGroupCallEndSignal,
   ForwardGroupCallSignalling,
   ProcessGroupCallSignalling
;

IMPORTS
   OPERATION
FROM TCAPMessages {
   ccitt recommendation q 773 modules (2) messages (1) version2 (2)}

   SystemFailure,
   UnexpectedDataValue,
   NoGroupCallNumberAvailable
FROM MAP-Errors {
   ccitt identified-organization (4) etsi (0) mobileDomain (0)
   gsm-Network (1) modules (3) map-Errors (10) version6 (6)}

   PrepareGroupCallArg,
   PrepareGroupCallRes,
   SendGroupCallEndSignalArg,
   SendGroupCallEndSignalRes,
   ForwardGroupCallSignallingArg,
   ProcessGroupCallSignallingArg
FROM MAP-GR-DataTypes {
   ccitt identified-organization (4) etsi (0) mobileDomain (0)
   gsm-Network (1) modules (3) map-GR-DataTypes (23) version6 (6)}



;
```

```
PrepareGroupCall ::= OPERATION                                          --Timer m
     ARGUMENT
          prepareGroupCallArg             PrepareGroupCallArg
     RESULT
          prepareGroupCallRes             PrepareGroupCallRes
     ERRORS {
          SystemFailure,
          NoGroupCallNumberAvailable,
          UnexpectedDataValue}
```

```
SendGroupCallEndSignal ::= OPERATION                                    --Timer l
     ARGUMENT
          sendGroupCallEndSignalArg       SendGroupCallEndSignalArg
     RESULT
          sendGroupCallEndSignalRes       SendGroupCallEndSignalRes
```

```
ProcessGroupCallSignalling ::= OPERATION                                --Timer s
     ARGUMENT
          processGroupCallSignallingArg   ProcessGroupCallSignallingArg
```

```
ForwardGroupCallSignalling ::= OPERATION                                --Timer s
     ARGUMENT
          forwardGroupCallSignallingArg   ForwardGroupCallSignallingArg
```

```
END
```

17.6.8 Location service operations

```
MAP-LocationServiceOperations {
   ccitt identified-organization (4) etsi (0) mobileDomain (0)
   gsm-Network (1) modules (3) map-LocationServiceOperations (24)
   version6 (6)}

DEFINITIONS

::=

BEGIN

EXPORTS
   ProvideSubscriberLocation,
   SendRoutingInfoForLCS,
   SubscriberLocationReport
;

IMPORTS
   OPERATION
FROM TCAPMessages {
   ccitt recommendation q 773 modules (2) messages (1) version2 (2)}

   SystemFailure,
   DataMissing,
   UnexpectedDataValue,
   FacilityNotSupported,
   UnknownSubscriber,
   AbsentSubscriber,
   UnauthorizedRequestingNetwork,
   UnauthorizedLCSClient,
   PositionMethodFailure,
   ResourceLimitation,
   UnknownOrUnreachableLCSClient,
   UnidentifiedSubscriber,
   IllegalEquipment,
   IllegalSubscriber
FROM MAP-Errors {
   ccitt identified-organization (4) etsi (0) mobileDomain (0)
   gsm-Network (1) modules (3) map-Errors (10) version6 (6)}

   RoutingInfoForLCS-Arg,
   RoutingInfoForLCS-Res,
   ProvideSubscriberLocation-Arg,
   ProvideSubscriberLocation-Res,
   SubscriberLocationReport-Arg,
   SubscriberLocationReport-Res
FROM MAP-LCS-DataTypes {
   ccitt identified-organization (4) etsi (0) mobileDomain (0)
   gsm-Network (1) modules (3) map-LCS-DataTypes (25) version6 (6)}
```

```
;

SendRoutingInfoForLCS ::= OPERATION                                    --Timer m
     ARGUMENT
          routingInfoForLCS-Arg          RoutingInfoForLCS-Arg
     RESULT
          routingInfoForLCS-Res          RoutingInfoForLCS-Res
     ERRORS {
          SystemFailure,
          DataMissing,
          UnexpectedDataValue,
          FacilityNotSupported,
          UnknownSubscriber,
          AbsentSubscriber,
          UnauthorizedRequestingNetwork }

ProvideSubscriberLocation ::= OPERATION                                --Timer ml
     ARGUMENT
          provideSubscriberLocation-Arg   ProvideSubscriberLocation-Arg
     RESULT
          provideSubscriberLocation-Res   ProvideSubscriberLocation-Res
     ERRORS {
          SystemFailure,
          DataMissing,
          UnexpectedDataValue,
          FacilityNotSupported,
          UnidentifiedSubscriber,
          IllegalSubscriber,
          IllegalEquipment,
          AbsentSubscriber,
          UnauthorizedRequestingNetwork,
          UnauthorizedLCSClient,
          PositionMethodFailure }

SubscriberLocationReport ::= OPERATION                                 --Timer m
     ARGUMENT
          subscriberLocationReport-Arg    SubscriberLocationReport-Arg
     RESULT
          subscriberLocationReport-Res    SubscriberLocationReport-Res
     ERRORS {
          SystemFailure,
          DataMissing,
          ResourceLimitation,
          UnexpectedDataValue,
          UnknownSubscriber,
          UnauthorizedRequestingNetwork,
          UnknownOrUnreachableLCSClient}

END
```

17.7 MAP constants and data types

17.7.1 Mobile Service data types

```
MAP-MS-DataTypes {
   ccitt identified-organization (4) etsi (0) mobileDomain (0)
   gsm-Network (1) modules (3) map-MS-DataTypes (11) version6 (6)}

DEFINITIONS

IMPLICIT TAGS

::=

BEGIN

EXPORTS

   -- location registration types
   UpdateLocationArg,
   UpdateLocationRes,
   CancelLocationArg,
   CancelLocationRes,
   PurgeMS-Arg,
```

```
    PurgeMS-Res,
    SendIdentificationArg,
    SendIdentificationRes,
    UpdateGprsLocationArg,
    UpdateGprsLocationRes,
    IST-SupportIndicator,

    -- gprs location registration types
    GSN-Address,

    -- handover types
    ForwardAccessSignalling-Arg,
    PrepareHO-Arg,
    PrepareHO-Res,
    PrepareSubsequentHO-Arg,
    PrepareSubsequentHO-Res,
    ProcessAccessSignalling-Arg,
    SendEndSignal-Arg,
    SendEndSignal-Res,


    -- authentication management types
    SendAuthenticationInfoArg,
    SendAuthenticationInfoRes,
    AuthenticationFailureReportArg,
    AuthenticationFailureReportRes,

    -- security management types
    EquipmentStatus,
    Kc,

    -- subscriber management types
    InsertSubscriberDataArg,
    InsertSubscriberDataRes,
    LSAIdentity,
    DeleteSubscriberDataArg,
    DeleteSubscriberDataRes,
    Ext-QoS-Subscribed,
    SubscriberData,
    ODB-Data,
    SubscriberStatus,
    ZoneCodeList,
    maxNumOfZoneCodes,
    O-CSI,
    D-CSI,
    O-BcsmCamelTDPCriteriaList,
    T-BCSM-CAMEL-TDP-CriteriaList,
    SS-CSI,
    ServiceKey,
    DefaultCallHandling,
    CamelCapabilityHandling,
    BasicServiceCriteria,
    SupportedCamelPhases,
    maxNumOfCamelTDPData,
    CUG-Index,
    CUG-Info,
    CUG-Interlock,
    InterCUG-Restrictions,
    IntraCUG-Options,
    NotificationToMSUser,
    QoS-Subscribed,
    IST-AlertTimerValue,
    T-CSI,
    T-BcsmTriggerDetectionPoint,

    -- fault recovery types
    ResetArg,
    RestoreDataArg,
    RestoreDataRes,

    -- provide subscriber info types
    GeographicalInformation,

    -- subscriber information enquiry types
    ProvideSubscriberInfoArg,
    ProvideSubscriberInfoRes,
    SubscriberInfo,
    LocationInformation,
    SubscriberState,
```

```
   -- any time information enquiry types
   AnyTimeInterrogationArg,
   AnyTimeInterrogationRes,

   -- any time information handling types
   AnyTimeSubscriptionInterrogationArg,
   AnyTimeSubscriptionInterrogationRes,
   AnyTimeModificationArg,
   AnyTimeModificationRes,

   -- subscriber data modification notification types
   NoteSubscriberDataModifiedArg,
   NoteSubscriberDataModifiedRes,

   -- gprs location information retrieval types
   SendRoutingInfoForGprsArg,
   SendRoutingInfoForGprsRes,

   -- failure reporting types
   FailureReportArg,
   FailureReportRes,

   -- gprs notification types
   NoteMsPresentForGprsArg,
   NoteMsPresentForGprsRes,

   -- Mobility Management types
   NoteMM-EventArg,
   NoteMM-EventRes



;

IMPORTS
   maxNumOfSS,
   SS-SubscriptionOption,
   SS-List,
   SS-ForBS-Code,
   Password
FROM MAP-SS-DataTypes {
   ccitt identified-organization (4) etsi (0) mobileDomain (0)
   gsm-Network (1) modules (3) map-SS-DataTypes (14) version6 (6)}

   SS-Code
FROM MAP-SS-Code {
   ccitt identified-organization (4) etsi (0) mobileDomain (0)
   gsm-Network (1) modules (3) map-SS-Code (15) version6 (6)}

   Ext-BearerServiceCode
FROM MAP-BS-Code {
   ccitt identified-organization (4) etsi (0) mobileDomain (0)
   gsm-Network (1) modules (3) map-BS-Code (20) version6 (6)}

   Ext-TeleserviceCode
FROM MAP-TS-Code {
   ccitt identified-organization (4) etsi (0) mobileDomain (0)
   gsm-Network (1) modules (3) map-TS-Code (19) version6 (6)}


   AddressString,
   ISDN-AddressString,
   ISDN-SubaddressString,
   FTN-AddressString,
   AccessNetworkSignalInfo,
   IMSI,
   TMSI,
   HLR-List,
   LMSI,
   Identity,
   GlobalCellId,
   CellGlobalIdOrServiceAreaIdOrLAI,
   Ext-BasicServiceCode,
   NAEA-PreferredCI,
   EMLPP-Info,
   MC-SS-Info,
   SubscriberIdentity,
   AgeOfLocationInformation,
```

```
   LCSClientExternalID,
   LCSClientInternalID,
   Ext-SS-Status,
   ASCI-CallReference,
   TBCD-STRING



FROM MAP-CommonDataTypes {
   ccitt identified-organization (4) etsi (0) mobileDomain (0)
   gsm-Network (1) modules (3) map-CommonDataTypes (18) version6 (6)}

   ExtensionContainer
FROM MAP-ExtensionDataTypes {
   ccitt identified-organization (4) etsi (0) mobileDomain (0)
   gsm-Network (1) modules (3) map-ExtensionDataTypes (21) version6 (6)}

   AbsentSubscriberDiagnosticSM
FROM MAP-ER-DataTypes {
   ccitt identified-organization (4) etsi (0) mobileDomain (0)
   gsm-Network (1) modules (3) map-ER-DataTypes (17) version6 (6)}


;


-- location registration types

UpdateLocationArg ::= SEQUENCE {
     imsi                                IMSI,

     msc-Number                          [1] ISDN-AddressString,
     vlr-Number                          ISDN-AddressString,
     lmsi                                [10] LMSI OPTIONAL,
     extensionContainer                  ExtensionContainer              OPTIONAL,
     ... ,
     vlr-Capability                      [6] VLR-Capability              OPTIONAL,
     informPreviousNetworkEntity         [11] NULL                       OPTIONAL }

VLR-Capability ::= SEQUENCE{
     supportedCamelPhases                [0] SupportedCamelPhases        OPTIONAL,
     extensionContainer                  ExtensionContainer              OPTIONAL,
     ... ,
     solsaSupportIndicator               [2] NULL                        OPTIONAL,
     istSupportIndicator                 [1] IST-SupportIndicator        OPTIONAL,
     superChargerSupportedInServingNetworkEntity   [3] SuperChargerInfo  OPTIONAL,
     longFTN-Supported                   [4]  NULL                       OPTIONAL }

SuperChargerInfo ::= CHOICE {
     sendSubscriberData                  [0] NULL,
     subscriberDataStored                [1] AgeIndicator }

AgeIndicator ::= OCTET STRING (SIZE (1..6))
     -- The internal structure of this parameter is implementation specific.

IST-SupportIndicator ::=  ENUMERATED {
     basicISTSupported                   (0),
     istCommandSupported                 (1),
     ...}
-- exception handling:
-- reception of values > 1 shall be mapped to ' istCommandSupported '

UpdateLocationRes ::= SEQUENCE {
     hlr-Number                          ISDN-AddressString,

     extensionContainer                  ExtensionContainer              OPTIONAL,
     ... }

CancelLocationArg ::= [3] SEQUENCE {
     identity                            Identity,
     cancellationType                    CancellationType                OPTIONAL,
     extensionContainer                  ExtensionContainer              OPTIONAL,
     ...}
```

```
CancellationType ::= ENUMERATED {
     updateProcedure                    (0),
     subscriptionWithdraw               (1),
     ...}
     -- The HLR shall not send values other than listed above
```

```
CancelLocationRes ::= SEQUENCE {
     extensionContainer                 ExtensionContainer              OPTIONAL,
     ...}
```

```
PurgeMS-Arg ::= [3] SEQUENCE {
     imsi                               IMSI,
     vlr-Number                         [0] ISDN-AddressString          OPTIONAL,
     sgsn-Number                        [1]  ISDN-AddressString         OPTIONAL,
     extensionContainer                 ExtensionContainer              OPTIONAL,
     ...}
```

```
PurgeMS-Res ::= SEQUENCE {
     freezeTMSI                         [0]  NULL                       OPTIONAL,
     freezeP-TMSI                       [1]  NULL                       OPTIONAL,
     extensionContainer                 ExtensionContainer              OPTIONAL,
     ...}
```

```
SendIdentificationArg ::= SEQUENCE {
     tmsi                               TMSI,
     numberOfRequestedVectors           NumberOfRequestedVectors        OPTIONAL,
     --  within a dialogue numberOfRequestedVectors shall be present in
     -- the first service request and shall not be present in subsequent
     -- service requests. If received in a subsequent  service request it
     -- shall be discarded.
     segmentationProhibited             NULL                            OPTIONAL,
     extensionContainer                 ExtensionContainer              OPTIONAL,
     ...}
```

```
SendIdentificationRes ::= [3] SEQUENCE {
     imsi                               IMSI                            OPTIONAL,
     -- IMSI shall be present in the first (or only) service response of a dialogue.
     -- If multiple service requests are present in a dialogue then IMSI
     -- shall not be present in any service response other than the first one.
     authenticationSetList              AuthenticationSetList           OPTIONAL,
     currentSecurityContext             [2]CurrentSecurityContext       OPTIONAL,
     extensionContainer                 [3] ExtensionContainer          OPTIONAL,
     ...}
```

```
-- authentication management types
```

```
AuthenticationSetList ::= CHOICE {
     tripletList                        [0] TripletList,
     quintupletList                     [1] QuintupletList }
```

```
TripletList ::= SEQUENCE SIZE (1..5) OF
                                        AuthenticationTriplet
```

```
QuintupletList ::= SEQUENCE SIZE (1..5) OF
                                        AuthenticationQuintuplet
```

```
AuthenticationTriplet ::= SEQUENCE {
     rand                               RAND,
     sres                               SRES,
     kc                                 Kc,
     ...}
```

```
AuthenticationQuintuplet ::= SEQUENCE {
     rand                               RAND,
     xres                               XRES,
     ck                                 CK,
     ik                                 IK,
     autn                               AUTN,
     ...}
```

```
CurrentSecurityContext ::= CHOICE {
     gsm-SecurityContextData            [0] GSM-SecurityContextData,
     umts-SecurityContextData           [1] UMTS-SecurityContextData }
```

```
GSM-SecurityContextData ::= SEQUENCE {
    kc                                  Kc,
    cksn                                Cksn,
    ... }
```

```
UMTS-SecurityContextData ::= SEQUENCE {
    ck                                  CK,
    ik                                  IK,
    ksi                                 KSI,
    ... }
```

```
RAND ::= OCTET STRING (SIZE (16))
```

```
SRES ::= OCTET STRING (SIZE (4))
```

```
Kc ::= OCTET STRING (SIZE (8))
```

```
XRES ::= OCTET STRING (SIZE (4..16))
```

```
CK ::= OCTET STRING (SIZE (16))
```

```
IK ::= OCTET STRING (SIZE (16))
```

```
AUTN ::= OCTET STRING (SIZE (16))
```

```
AUTS ::= OCTET STRING (SIZE (14))
```

```
Cksn ::= OCTET STRING (SIZE (1))
    -- The internal structure is defined in 3GPP TS 24.008
```

```
KSI ::= OCTET STRING (SIZE (1))
    -- The internal structure is defined in 3GPP TS 24.008
```

```
AuthenticationFailureReportArg ::= SEQUENCE {
    imsi                                IMSI,
    failureCause                        FailureCause,
    extensionContainer                  ExtensionContainer              OPTIONAL,
    ...}
```

```
AuthenticationFailureReportRes ::= SEQUENCE {
    extensionContainer                  ExtensionContainer              OPTIONAL,
    ...}
```

```
FailureCause ::= ENUMERATED {
    wrongUserResponse  (0),
    wrongNetworkSignature  (1)}
```

-- gprs location registration types

```
UpdateGprsLocationArg ::= SEQUENCE {
    imsi                                IMSI,
    sgsn-Number                         ISDN-AddressString,
    sgsn-Address                        GSN-Address,
    extensionContainer                  ExtensionContainer              OPTIONAL,
    ... ,
    sgsn-Capability                     [0] SGSN-Capability             OPTIONAL,
    informPreviousNetworkEntity         [1]  NULL                       OPTIONAL }
```

```
SGSN-Capability ::= SEQUENCE{
    solsaSupportIndicator               NULL                            OPTIONAL,
    extensionContainer                  [1] ExtensionContainer          OPTIONAL,
    ... ,
    superChargerSupportedInServingNetworkEntity   [2] SuperChargerInfo  OPTIONAL ,
    gprsEnhancementsSupportIndicator    [3] NULL                        OPTIONAL,
    supportedCamelPhases                [4] SupportedCamelPhases        OPTIONAL }
```

```
GSN-Address ::= OCTET STRING (SIZE (5..17))
    -- Octets are coded according to 3GPP TS 23.003
```

```
UpdateGprsLocationRes ::= SEQUENCE {
    hlr-Number                          ISDN-AddressString,
    extensionContainer                  ExtensionContainer              OPTIONAL,
    ...}
```

```
-- handover types

ForwardAccessSignalling-Arg ::= [3] SEQUENCE {
     an-APDU                          AccessNetworkSignalInfo,
     integrityProtectionInfo          [0] IntegrityProtectionInformation  OPTIONAL,
     encryptionInfo                   [1] EncryptionInformation           OPTIONAL,
     keyStatus                        [2]  KeyStatus                      OPTIONAL,
     allowedGSM-Algorithms            [4]  AllowedGSM-Algorithms          OPTIONAL,
     allowedUMTS-Algorithms           [5]  AllowedUMTS-Algorithms         OPTIONAL,
     radioResourceInformation         [6] RadioResourceInformation        OPTIONAL,
     extensionContainer               [3]  ExtensionContainer             OPTIONAL,
     ...,
     radioResourceList                [7]  RadioResourceList              OPTIONAL,
     bssmap-ServiceHandover           [9]  BSSMAP-ServiceHandover         OPTIONAL,
     ranap-ServiceHandover            [8]  RANAP-ServiceHandover          OPTIONAL,
     bssmap-ServiceHandoverList       [10] BSSMAP-ServiceHandoverList     OPTIONAL }

AllowedGSM-Algorithms ::= OCTET STRING (SIZE (1))
     -- internal structure is coded as Algorithm identifier octet from
     -- Permitted Algorithms defined in GSM 08.08
     -- A node shall mark all GSM algorithms that are allowed in MSC-B

AllowedUMTS-Algorithms ::= SEQUENCE {
     integrityProtectionAlgorithms        [0]  PermittedIntegrityProtectionAlgorithms
     OPTIONAL,
     encryptionAlgorithms                 [1]  PermittedEncryptionAlgorithms  OPTIONAL,
     extensionContainer                   [2]  ExtensionContainer             OPTIONAL,
     ...}

PermittedIntegrityProtectionAlgorithms ::=
          OCTET STRING (SIZE (1..maxPermittedIntegrityProtectionAlgorithmsLength))
     -- Octets contain a complete PermittedIntegrityProtectionAlgorithms data type
     -- as defined in 3GPP TS 25.413, encoded according to the encoding scheme
     -- mandated by 3GPP TS 25.413
     -- Padding bits are included, if needed, in the least significant bits of the
     -- last octet of the octet string.

maxPermittedIntegrityProtectionAlgorithmsLength INTEGER ::= 9

PermittedEncryptionAlgorithms ::=
          OCTET STRING (SIZE (1..maxPermittedEncryptionAlgorithmsLength))
     -- Octets contain a complete PermittedEncryptionAlgorithms data type
     -- as defined in 3GPP TS 25.413, encoded according to the encoding scheme
     -- mandated by 3GPP TS 25.413
     -- Padding bits are included, if needed, in the least significant bits of the
     -- last octet of the octet string.

maxPermittedEncryptionAlgorithmsLength INTEGER ::= 9

KeyStatus ::= ENUMERATED {
     old  (0),
     new  (1),
     ...}
     -- exception handling:
     -- received values in range 2-31 shall be treated as "old"
     -- received values greater than 31 shall be treated as "new"
```

```
PrepareHO-Arg ::= [3] SEQUENCE {
     targetCellId                        [0] GlobalCellId                    OPTIONAL,
     ho-NumberNotRequired                NULL                                OPTIONAL,
     targetRNCId                         [1] RNCId                           OPTIONAL,
     an-APDU                             [2] AccessNetworkSignalInfo         OPTIONAL,
     multipleBearerRequested             [3] NULL                            OPTIONAL,
     imsi                                [4] IMSI                            OPTIONAL,
     integrityProtectionInfo             [5] IntegrityProtectionInformation  OPTIONAL,
     encryptionInfo                      [6] EncryptionInformation           OPTIONAL,
     radioResourceInformation            [7] RadioResourceInformation        OPTIONAL,
     allowedGSM-Algorithms               [9]  AllowedGSM-Algorithms          OPTIONAL,
     allowedUMTS-Algorithms              [10] AllowedUMTS-Algorithms         OPTIONAL,
     radioResourceList                   [11] RadioResourceList              OPTIONAL,
     extensionContainer                  [8] ExtensionContainer              OPTIONAL,
     ... ,
     rab-Id                              [12] RAB-Id                         OPTIONAL,
     bssmap-ServiceHandover              [13] BSSMAP-ServiceHandover         OPTIONAL,
     ranap-ServiceHandover               [14] RANAP-ServiceHandover          OPTIONAL,
     bssmap-ServiceHandoverList          [15] BSSMAP-ServiceHandoverList     OPTIONAL,
     asciCallReference                   [20] ASCI-CallReference             OPTIONAL
      }
```

```
BSSMAP-ServiceHandoverList ::= SEQUENCE SIZE (1.. maxNumOfServiceHandovers) OF
                                    BSSMAP-ServiceHandoverInfo
```

```
BSSMAP-ServiceHandoverInfo ::= SEQUENCE {
     bssmap-ServiceHandover              BSSMAP-ServiceHandover,
     rab-Id                              RAB-Id,
     -- RAB Identity is needed to relate the service handovers with the radio access bearers.
     ...}
```

```
maxNumOfServiceHandovers  INTEGER ::= 7
```

```
BSSMAP-ServiceHandover ::= OCTET STRING (SIZE (1))
     -- Octets are coded according the Service Handover information element in
     -- GSM 08.08.
```

```
RANAP-ServiceHandover ::= OCTET STRING (SIZE (1))
     -- Octet contains a complete Service-Handover data type
     -- as defined in 3GPP TS 25.413, encoded according to the encoding scheme
     -- mandated by 3GPP TS 25.413
     -- Padding bits are included in the least significant bits.
```

```
RadioResourceList ::= SEQUENCE SIZE (1.. maxNumOfRadioResources) OF
                                    RadioResource
```

```
RadioResource ::= SEQUENCE {
     radioResourceInformation            RadioResourceInformation,
     rab-Id                              RAB-Id,
     -- RAB Identity is needed to relate the radio resources with the radio access bearers.
     ...}
```

```
maxNumOfRadioResources  INTEGER ::= 7
```

```
PrepareHO-Res ::= [3] SEQUENCE {
     handoverNumber                      [0] ISDN-AddressString              OPTIONAL,
     relocationNumberList                [1]  RelocationNumberList           OPTIONAL,
     an-APDU                             [2]  AccessNetworkSignalInfo        OPTIONAL,
     multicallBearerInfo                 [3]  MulticallBearerInfo            OPTIONAL,
     multipleBearerNotSupported          NULL                                OPTIONAL,
     selectedUMTS-Algorithms             [5]  SelectedUMTS-Algorithms        OPTIONAL,
     chosenRadioResourceInformation      [6] ChosenRadioResourceInformation OPTIONAL,
     extensionContainer                  [4]  ExtensionContainer             OPTIONAL,
     ...}
```

```
SelectedUMTS-Algorithms ::= SEQUENCE {
     integrityProtectionAlgorithm        [0]  ChosenIntegrityProtectionAlgorithm   OPTIONAL,
     encryptionAlgorithm                 [1]  ChosenEncryptionAlgorithm      OPTIONAL,
     extensionContainer                  [2]  ExtensionContainer             OPTIONAL,
     ...}
```

```
ChosenIntegrityProtectionAlgorithm ::= OCTET STRING (SIZE (1))
     -- Octet contains a complete IntegrityProtectionAlgorithm data type
     -- as defined in 3GPP TS 25.413, encoded according to the encoding scheme
     -- mandated by 3GPP TS 25.413
     -- Padding bits are included in the least significant bits.
```

```
ChosenEncryptionAlgorithm ::= OCTET STRING (SIZE (1))
     -- Octet contains a complete EncryptionAlgorithm data type
     -- as defined in 3GPP TS 25.413, encoded according to the encoding scheme
     -- mandated by 3GPP TS 25.413
     -- Padding bits are included in the least significant bits.
```

```
ChosenRadioResourceInformation ::= SEQUENCE {
     chosenChannelInfo                  [0] ChosenChannelInfo             OPTIONAL,
     chosenSpeechVersion                [1] ChosenSpeechVersion           OPTIONAL,
     ...}
```

```
ChosenChannelInfo ::= OCTET STRING (SIZE (1))
     -- Octets are coded according the Chosen Channel information element in GSM 08.08
```

```
ChosenSpeechVersion ::= OCTET STRING (SIZE (1))
     -- Octets are coded according the Speech Version (chosen) information element in GSM
     -- 08.08
```

```
PrepareSubsequentHO-Arg ::= [3] SEQUENCE {
     targetCellId                       [0] GlobalCellId                  OPTIONAL,
     targetMSC-Number                   [1] ISDN-AddressString,
     targetRNCId                        [2] RNCId                         OPTIONAL,
     an-APDU                            [3]  AccessNetworkSignalInfo      OPTIONAL,
     selectedRab-Id                     [4]  RAB-Id                       OPTIONAL,
     extensionContainer                 [5]  ExtensionContainer           OPTIONAL,
     ...}
```

```
PrepareSubsequentHO-Res ::= [3] SEQUENCE {
     an-APDU                           AccessNetworkSignalInfo,
     extensionContainer                 [0]  ExtensionContainer           OPTIONAL,
     ...}
```

```
ProcessAccessSignalling-Arg ::= [3] SEQUENCE {
     an-APDU                           AccessNetworkSignalInfo,
     selectedUMTS-Algorithms            [1]  SelectedUMTS-Algorithms      OPTIONAL,
     selectedGSM-Algorithm              [2]  SelectedGSM-Algorithm        OPTIONAL,
     chosenRadioResourceInformation     [3] ChosenRadioResourceInformation OPTIONAL,
     selectedRab-Id                     [4] RAB-Id                        OPTIONAL,
     extensionContainer                 [0]  ExtensionContainer           OPTIONAL,
     ...}
```

```
SelectedGSM-Algorithm ::= OCTET STRING (SIZE (1))
     -- internal structure is coded as Algorithm identifier octet from Chosen Encryption
     -- Algorithm defined in GSM 08.08
     -- A node shall mark only the selected GSM algorithm
```

```
SendEndSignal-Arg ::= [3] SEQUENCE {
     an-APDU                           AccessNetworkSignalInfo,
     extensionContainer                 [0]  ExtensionContainer           OPTIONAL,
     ...}
```

```
SendEndSignal-Res ::= SEQUENCE {
     extensionContainer                 [0]  ExtensionContainer           OPTIONAL,
     ...}
```

```
RNCId ::= OCTET STRING (SIZE (7))
     -- The internal structure is defined as follows:
     -- octet 1 bits 4321                Mobile Country Code 1st digit
     --         bits 8765                Mobile Country Code 2nd digit
     -- octet 2 bits 4321                Mobile Country Code 3rd digit
     --         bits 8765                Mobile Network Code 3rd digit
     --                                  or filler (1111) for 2 digit MNCs
     -- octet 3 bits 4321                Mobile Network Code 1st digit
     --         bits 8765                Mobile Network Code 2nd digit
     -- octets 4 and 5                   Location Area Code according to 3GPP TS 24.008
     -- octets 6 and 7                   RNC Id value according to 3GPP TS 25.413
```

```
RelocationNumberList ::= SEQUENCE SIZE (1..maxNumOfRelocationNumber) OF
                                  RelocationNumber
```

```
MulticallBearerInfo ::= INTEGER (1..maxNumOfRelocationNumber)
```

```
RelocationNumber ::= SEQUENCE {
     handoverNumber                   ISDN-AddressString,
     rab-Id                           RAB-Id,
     -- RAB Identity is needed to relate the calls with the radio access bearers.
     ...}
```

```
RAB-Id ::= INTEGER (1..maxNrOfRABs)
```

```
maxNrOfRABs INTEGER ::= 255
```

```
maxNumOfRelocationNumber  INTEGER ::= 7
```

```
RadioResourceInformation ::= OCTET STRING (SIZE (3..13))
     -- Octets are coded according the Channel Type information element in GSM 08.08
```

```
IntegrityProtectionInformation ::= OCTET STRING (SIZE (18..maxNumOfIntegrityInfo))
     -- Octets contain a complete IntegrityProtectionInformation data type
     -- as defined in 3GPP TS 25.413, encoded according to the encoding scheme
     -- mandated by 3GPP TS 25.413
     -- Padding bits are included, if needed, in the least significant bits of the
     -- last octet of the octet string.
```

```
maxNumOfIntegrityInfo INTEGER ::= 100
```

```
EncryptionInformation ::= OCTET STRING (SIZE (18..maxNumOfEncryptionInfo))
     -- Octets contain a complete EncryptionInformation data type
     -- as defined in 3GPP TS 25.413, encoded according to the encoding scheme
     -- mandated by 3GPP TS 25.413
     -- Padding bits are included, if needed, in the least significant bits of the
     -- last octet of the octet string.
```

```
maxNumOfEncryptionInfo INTEGER ::= 100
```

```
-- authentication management types
```

```
SendAuthenticationInfoArg ::= SEQUENCE {
     imsi                             [0] IMSI,
     numberOfRequestedVectors         NumberOfRequestedVectors,
     segmentationProhibited           NULL                                OPTIONAL,
     immediateResponsePreferred       [1] NULL                             OPTIONAL,
     re-synchronisationInfo           Re-synchronisationInfo              OPTIONAL,
     extensionContainer               [2] ExtensionContainer              OPTIONAL,
     ...}
```

```
NumberOfRequestedVectors ::= INTEGER (1..5)
```

```
Re-synchronisationInfo ::= SEQUENCE {
     rand                             RAND,
     auts                             AUTS,
     ...}
```

```
SendAuthenticationInfoRes ::= [3] SEQUENCE {
     authenticationSetList            AuthenticationSetList               OPTIONAL,
     extensionContainer               ExtensionContainer                  OPTIONAL,
     ...}
```

```
-- security management types

EquipmentStatus ::= ENUMERATED {
     whiteListed  (0),
     blackListed  (1),
     greyListed  (2)}

-- subscriber management types

InsertSubscriberDataArg ::= SEQUENCE {
     imsi                                [0] IMSI                              OPTIONAL,
     COMPONENTS OF                       SubscriberData,
     extensionContainer                  [14] ExtensionContainer               OPTIONAL,
     ... ,
     naea-PreferredCI                    [15] NAEA-PreferredCI                 OPTIONAL,
     -- naea-PreferredCI is included at the discretion of the HLR operator.
     gprsSubscriptionData                [16] GPRSSubscriptionData             OPTIONAL,
     roamingRestrictedInSgsnDueToUnsupportedFeature [23]                       NULL
                                                                               OPTIONAL,
     networkAccessMode                   [24] NetworkAccessMode                OPTIONAL,
     lsaInformation                      [25] LSAInformation                   OPTIONAL,
     lmu-Indicator                       [21] NULL                             OPTIONAL,
     lcsInformation                      [22] LCSInformation                   OPTIONAL,
          istAlertTimer                  [26] IST-AlertTimerValue              OPTIONAL,
          superChargerSupportedInHLR     [27] AgeIndicator                     OPTIONAL,
          mc-SS-Info                     [28] MC-SS-Info                       OPTIONAL,
     cs-AllocationRetentionPriority      [29] CS-AllocationRetentionPriority   OPTIONAL,
     sgsn-CAMEL-SubscriptionInfo         [17] SGSN-CAMEL-SubscriptionInfo      OPTIONAL,
     chargingCharacteristics             [18] ChargingCharacteristics          OPTIONAL
 }
     -- If the Network Access Mode parameter is sent, it shall be present only in
     -- the first sequence if segmentation is used

CS-AllocationRetentionPriority ::= OCTET STRING (SIZE (1))
     -- This data type encodes each priority level defined in TS 23.107 as the binary value
     -- of the priority level.

IST-AlertTimerValue ::= INTEGER (15..255)

LCSInformation ::= SEQUENCE {
     gmlc-List [0]                       GMLC-List OPTIONAL,
     lcs-PrivacyExceptionList            [1]  LCS-PrivacyExceptionList     OPTIONAL,
     molr-List                           [2]  MOLR-List                    OPTIONAL,
     ...}

GMLC-List ::= SEQUENCE SIZE (1..maxNumOfGMLC) OF
                                         ISDN-AddressString
     -- if segmentation is used, the complete GMLC-List shall be sent in one segment

maxNumOfGMLC  INTEGER ::= 5

NetworkAccessMode ::= ENUMERATED {
     bothMSCAndSGSN                      (0),
     onlyMSC                             (1),
     onlySGSN                            (2),
     ...}
     -- if unknown values are received in NetworkAccessMode
     -- they shall be discarded.

GPRSDataList ::= SEQUENCE SIZE (1..maxNumOfPDP-Contexts) OF
                                         PDP-Context

maxNumOfPDP-Contexts  INTEGER ::= 50
```

```
PDP-Context ::= SEQUENCE {
     pdp-ContextId                        ContextId,
     pdp-Type                             [16] PDP-Type,
     pdp-Address                          [17] PDP-Address                    OPTIONAL,
     qos-Subscribed                       [18] QoS-Subscribed,
     vplmnAddressAllowed                  [19] NULL OPTIONAL,
     apn                                  [20] APN,
     extensionContainer                   [21] ExtensionContainer             OPTIONAL,
     ... ,
     ext-QoS-Subscribed                   [0] Ext-QoS-Subscribed              OPTIONAL,
     pdp-ChargingCharacteristics          [1] ChargingCharacteristics         OPTIONAL }
     -- qos-Subscribed shall be discarded if ext-QoS-Subscribed is received and supported
```

```
ContextId ::= INTEGER (1..maxNumOfPDP-Contexts)
```

```
GPRSSubscriptionData ::= SEQUENCE {
     completeDataListIncluded             NULL                                OPTIONAL,

          -- If segmentation is used, completeDataListIncluded may only be present in the
          -- first segment.
     gprsDataList                         [1]  GPRSDataList,
     extensionContainer                   [2] ExtensionContainer              OPTIONAL,
     ... }
```

```
SGSN-CAMEL-SubscriptionInfo ::= SEQUENCE {
     gprs-CSI                             [0]  GPRS-CSI                       OPTIONAL,
     sms-CSI                              [1]  SMS-CSI                        OPTIONAL,
     extensionContainer                   [2]  ExtensionContainer             OPTIONAL,
     ...}
```

```
GPRS-CSI ::= SEQUENCE {
     gprs-CamelTDPDataList                [0] GPRS-CamelTDPDataList           OPTIONAL,
     camelCapabilityHandling              [1] CamelCapabilityHandling         OPTIONAL,
     extensionContainer                   [2] ExtensionContainer              OPTIONAL,
     notificationToCSE                    [3]  NULL                           OPTIONAL,
     csi-Active                           [4]  NULL                           OPTIONAL,
     ...}
--   notificationToCSE and csi-Active shall not be present when GPRS-CSI is sent to SGSN.
--   They may only be included in ATSI/ATM ack/NSDC message.
--   GPRS-CamelTDPData and  camelCapabilityHandling shall be present in
--   the GPRS-CSI sequence.
--    If GPRS-CSI is segmented, gprs-CamelTDPDataList and camelCapabilityHandling shall be
--    present in the first segment
```

```
GPRS-CamelTDPDataList ::= SEQUENCE SIZE (1..maxNumOfCamelTDPData) OF
     GPRS-CamelTDPData
--   GPRS-CamelTDPDataList shall not contain more than one instance of
--   GPRS-CamelTDPData containing the same value for gprs-TriggerDetectionPoint.
```

```
GPRS-CamelTDPData ::= SEQUENCE {
     gprs-TriggerDetectionPoint           [0] GPRS-TriggerDetectionPoint,
     serviceKey                           [1] ServiceKey,
     gsmSCF-Address                       [2] ISDN-AddressString,
     defaultSessionHandling               [3] DefaultGPRS-Handling,
     extensionContainer                   [4] ExtensionContainer              OPTIONAL,
     ...
     }
```

```
DefaultGPRS-Handling ::= ENUMERATED {
     continueTransaction (0) ,
     releaseTransaction (1) ,
     ...}
-- exception handling:
-- reception of values in range 2-31 shall be treated as "continueTransaction"
-- reception of values greater than 31 shall be treated as "releaseTransaction"
```

```
GPRS-TriggerDetectionPoint ::= ENUMERATED {
     attach                                  (1),
     attachChangeOfPosition                  (2),
     pdp-ContextEstablishment                (11),
     pdp-ContextEstablishmentAcknowledgement (12),
     pdp-ContextChangeOfPosition             (14),
     ... }
-- exception handling:
-- For GPRS-CamelTDPData sequences containing this parameter with any
-- other value than the ones listed the receiver shall ignore the whole
-- GPRS-CamelTDPDatasequence.
```

```
APN ::=  OCTET STRING (SIZE (2..63))
     -- Octets are coded according to 3GPP TS 23.003
```

```
PDP-Type ::= OCTET STRING (SIZE (2))
     -- Octets are coded according to 3GPP TS 29.060
```

```
PDP-Address ::= OCTET STRING (SIZE (1..16))
     -- Octets are coded according to 3GPP TS 29.060

     -- The possible size values are:
     -- 1-7 octets  X.25 address type
     --  4  octets  IPv4 address type
     -- 16  octets  Ipv6 address type
```

```
QoS-Subscribed ::= OCTET STRING (SIZE (3))
     -- Octets are coded according to 3GPP TS 24.008.
```

```
Ext-QoS-Subscribed ::= OCTET STRING (SIZE (1..9))
     -- OCTET 1:
     --  Allocation/Retention Priority (This octet encodes each priority level defined in
     --     23.107 as the binary value of the priority level, declaration in 29.060)
     -- Octets 2-9 are coded according to 3GPP TS 24.008 Quality of Service Octets
     -- 6-13.
```

```
ChargingCharacteristics ::= OCTET STRING (SIZE (2))
     -- Octets are coded according to 3GPP TS 32.015.
```

```
LSAOnlyAccessIndicator ::= ENUMERATED {
     accessOutsideLSAsAllowed  (0),
     accessOutsideLSAsRestricted (1)}
```

```
LSADataList ::= SEQUENCE SIZE (1..maxNumOfLSAs) OF
                                     LSAData
```

```
maxNumOfLSAs  INTEGER ::= 20
```

```
LSAData ::= SEQUENCE {
     lsaIdentity                     [0] LSAIdentity,
     lsaAttributes                   [1] LSAAttributes,
     lsaActiveModeIndicator          [2] NULL                        OPTIONAL,
     extensionContainer              [3] ExtensionContainer          OPTIONAL,
     ...}
```

```
LSAInformation ::= SEQUENCE {
     completeDataListIncluded            NULL                            OPTIONAL,

          -- If segmentation is used, completeDataListIncluded may only be present in the
          -- first segment.
     lsaOnlyAccessIndicator          [1]  LSAOnlyAccessIndicator     OPTIONAL,
     lsaDataList                     [2]  LSADataList                OPTIONAL,
     extensionContainer              [3] ExtensionContainer          OPTIONAL,
     ...}
```

```
LSAIdentity ::= OCTET STRING (SIZE (3))
     -- Octets are coded according to 3GPP TS 23.003
```

```
LSAAttributes ::= OCTET STRING (SIZE (1))
     -- Octets are coded according to TS GSM 08.08
```

```
SubscriberData ::= SEQUENCE {
     msisdn                                [1] ISDN-AddressString            OPTIONAL,
     category                              [2] Category                      OPTIONAL,
     subscriberStatus                      [3] SubscriberStatus              OPTIONAL,
     bearerServiceList                     [4] BearerServiceList             OPTIONAL,
     -- The exception handling for reception of unsupported / not allocated
     -- bearerServiceCodes is defined in section 8.8.1
     teleserviceList                       [6] TeleserviceList               OPTIONAL,
     -- The exception handling for reception of unsupported / not allocated
     -- teleserviceCodes is defined in section 8.8.1
     provisionedSS                         [7] Ext-SS-InfoList               OPTIONAL,
     odb-Data                              [8] ODB-Data                      OPTIONAL,
     roamingRestrictionDueToUnsupportedFeature  [9] NULL                     OPTIONAL,
     regionalSubscriptionData              [10] ZoneCodeList                 OPTIONAL,
     vbsSubscriptionData                   [11] VBSDataList                  OPTIONAL,
     vgcsSubscriptionData                  [12] VGCSDataList                 OPTIONAL,
     vlrCamelSubscriptionInfo              [13] VlrCamelSubscriptionInfo     OPTIONAL
     }
```

```
Category ::= OCTET STRING (SIZE (1))
     -- The internal structure is defined in CCITT Rec Q.763.
```

```
SubscriberStatus ::= ENUMERATED {
     serviceGranted  (0),
     operatorDeterminedBarring  (1)}
```

```
BearerServiceList ::= SEQUENCE SIZE (1..maxNumOfBearerServices) OF
                                       Ext-BearerServiceCode
```

```
maxNumOfBearerServices  INTEGER ::= 50
```

```
TeleserviceList ::= SEQUENCE SIZE (1..maxNumOfTeleservices) OF
                                       Ext-TeleserviceCode
```

```
maxNumOfTeleservices  INTEGER ::= 20
```

```
ODB-Data ::= SEQUENCE {
     odb-GeneralData                       ODB-GeneralData,
     odb-HPLMN-Data                        ODB-HPLMN-Data                    OPTIONAL,
     extensionContainer                    ExtensionContainer                OPTIONAL,
     ...}
```

```
ODB-GeneralData ::= BIT STRING {
     allOG-CallsBarred  (0),
     internationalOGCallsBarred  (1),
     internationalOGCallsNotToHPLMN-CountryBarred  (2),
     interzonalOGCallsBarred (6),
     interzonalOGCallsNotToHPLMN-CountryBarred (7),
     interzonalOGCallsAndInternationalOGCallsNotToHPLMN-CountryBarred (8),
     premiumRateInformationOGCallsBarred  (3),
     premiumRateEntertainementOGCallsBarred  (4),
     ss-AccessBarred  (5),
     allECT-Barred (9),
     chargeableECT-Barred (10),
     internationalECT-Barred (11),
     interzonalECT-Barred (12),
     doublyChargeableECT-Barred (13),
     multipleECT-Barred (14) ,
     roamingOutsidePLMNOG-CallsBarred  (18),
     allIC-CallsBarred  (19),
     roamingOutsidePLMNIC-CallsBarred  (20),
     roamingOutsidePLMNICountryIC-CallsBarred  (21),
     roamingOutsidePLMN-Barred  (22),
     roamingOutsidePLMN-CountryBarred  (23),
     registrationAllCF-Barred  (24),
     registrationCFNotToHPLMN-Barred  (25),
     registrationInterzonalCF-Barred  (26),
     registrationInterzonalCFNotToHPLMN-Barred  (27),
     registrationInternationalCF-Barred  (28)} (SIZE (15..32))
     -- exception handling: reception of unknown bit assignments in the
     -- ODB-GeneralData type shall be treated like unsupported ODB-GeneralData
     -- When the ODB-GeneralData type is removed from the HLR for a given subscriber,
     -- in NoteSubscriberDataModified operation sent toward the gsmSCF
     -- all bits shall be set to "0".
```

```
ODB-HPLMN-Data ::= BIT STRING {
     plmn-SpecificBarringType1  (0),
     plmn-SpecificBarringType2  (1),
     plmn-SpecificBarringType3  (2),
     plmn-SpecificBarringType4  (3)} (SIZE (4..32))
     -- exception handling: reception of unknown bit assignments in the
     -- ODB-HPLMN-Data type shall be treated like unsupported ODB-HPLMN-Data
     -- When the ODB-HPLMN-Data type is removed from the HLR for a given subscriber,
     -- in NoteSubscriberDataModified operation sent toward the gsmSCF
     -- all bits shall be set to "0".
```

```
Ext-SS-InfoList ::= SEQUENCE SIZE (1..maxNumOfSS) OF
                                    Ext-SS-Info
```

```
Ext-SS-Info ::= CHOICE {
     forwardingInfo                 [0] Ext-ForwInfo,
     callBarringInfo                [1] Ext-CallBarInfo,
     cug-Info                       [2] CUG-Info,
     ss-Data                        [3] Ext-SS-Data,
     emlpp-Info                     [4] EMLPP-Info}
```

```
Ext-ForwInfo ::= SEQUENCE {
     ss-Code                        SS-Code,
     forwardingFeatureList          Ext-ForwFeatureList,
     extensionContainer             [0] ExtensionContainer          OPTIONAL,
     ...}
```

```
Ext-ForwFeatureList ::= SEQUENCE SIZE (1..maxNumOfExt-BasicServiceGroups) OF
                                    Ext-ForwFeature
```

```
Ext-ForwFeature ::= SEQUENCE {
     basicService                   Ext-BasicServiceCode            OPTIONAL,
     ss-Status                      [4] Ext-SS-Status,
     forwardedToNumber              [5] ISDN-AddressString          OPTIONAL,
     -- When this data type is sent from an HLR which supports CAMEL Phase 2
     -- to a VLR that supports CAMEL Phase 2 the VLR shall not check the
     -- format of the number
     forwardedToSubaddress          [8] ISDN-SubaddressString       OPTIONAL,
     forwardingOptions              [6] Ext-ForwOptions             OPTIONAL,
     noReplyConditionTime           [7] Ext-NoRepCondTime           OPTIONAL,
     extensionContainer             [9] ExtensionContainer          OPTIONAL,
     ...,
     longForwardedToNumber          [10] FTN-AddressString          OPTIONAL }
```

```
Ext-ForwOptions ::= OCTET STRING (SIZE (1..5))

     -- OCTET 1:

     --  bit 8: notification to forwarding party
     --   0  no notification
     --   1  notification

     --  bit 7: redirecting presentation
     --   0 no presentation
     --   1  presentation

     --  bit 6: notification to calling party
     --   0  no notification
     --   1  notification

     --  bit 5: 0 (unused)

     --  bits 43: forwarding reason
     --   00  ms not reachable
     --   01  ms busy
     --   10  no reply
     --   11  unconditional

     -- bits 21: 00 (unused)

     -- OCTETS 2-5: reserved for future use. They shall be discarded if
     -- received and not understood.
```

```
Ext-NoRepCondTime ::= INTEGER (1..100)
    -- Only values 5-30 are used.
    -- Values in the ranges 1-4 and 31-100 are reserved for future use
    -- If received:
    --       values 1-4 shall be mapped on to value 5
    --       values 31-100 shall be mapped on to value 30
```

```
Ext-CallBarInfo ::= SEQUENCE {
    ss-Code                         SS-Code,
    callBarringFeatureList          Ext-CallBarFeatureList,
    extensionContainer              ExtensionContainer              OPTIONAL,
    ...}
```

```
Ext-CallBarFeatureList ::= SEQUENCE SIZE (1..maxNumOfExt-BasicServiceGroups) OF
                                    Ext-CallBarringFeature
```

```
Ext-CallBarringFeature ::= SEQUENCE {
    basicService                    Ext-BasicServiceCode            OPTIONAL,
    ss-Status                       [4] Ext-SS-Status,
    extensionContainer              ExtensionContainer              OPTIONAL,
    ...}
```

```
CUG-Info ::= SEQUENCE {
    cug-SubscriptionList            CUG-SubscriptionList,
    cug-FeatureList                 CUG-FeatureList                 OPTIONAL,
    extensionContainer              [0] ExtensionContainer          OPTIONAL,
    ...}
```

```
CUG-SubscriptionList ::= SEQUENCE SIZE (0..maxNumOfCUG) OF
                                    CUG-Subscription
```

```
CUG-Subscription ::= SEQUENCE {
    cug-Index CUG-Index,
    cug-Interlock                   CUG-Interlock,
    intraCUG-Options                IntraCUG-Options,
    basicServiceGroupList           Ext-BasicServiceGroupList       OPTIONAL,
    extensionContainer              [0] ExtensionContainer          OPTIONAL,
    ...}
```

```
CUG-Index ::= INTEGER (0..32767)
    -- The internal structure is defined in ETS 300 138.
```

```
CUG-Interlock ::= OCTET STRING (SIZE (4))
```

```
IntraCUG-Options ::= ENUMERATED {
    noCUG-Restrictions  (0),
    cugIC-CallBarred  (1),
    cugOG-CallBarred  (2)}
```

```
maxNumOfCUG  INTEGER ::= 10
```

```
CUG-FeatureList ::= SEQUENCE SIZE (1..maxNumOfExt-BasicServiceGroups) OF
                                    CUG-Feature
```

```
Ext-BasicServiceGroupList ::= SEQUENCE SIZE (1..maxNumOfExt-BasicServiceGroups) OF
                                    Ext-BasicServiceCode
```

```
maxNumOfExt-BasicServiceGroups  INTEGER ::= 32
```

```
CUG-Feature ::= SEQUENCE {
    basicService                    Ext-BasicServiceCode            OPTIONAL,
    preferentialCUG-Indicator       CUG-Index OPTIONAL,
    interCUG-Restrictions           InterCUG-Restrictions,
    extensionContainer              ExtensionContainer              OPTIONAL,
    ...}
```

```
InterCUG-Restrictions ::= OCTET STRING (SIZE (1))

     -- bits 876543: 000000 (unused)
     -- Exception handling:
     -- bits 876543 shall be ignored if received and not understood

     -- bits 21
     --   00  CUG only facilities
     --   01  CUG with outgoing access
     --   10  CUG with incoming access
     --   11  CUG with both outgoing and incoming access
```

```
Ext-SS-Data ::= SEQUENCE {
     ss-Code                              SS-Code,
     ss-Status [4] Ext-SS-Status,
     ss-SubscriptionOption                SS-SubscriptionOption             OPTIONAL,
     basicServiceGroupList                Ext-BasicServiceGroupList         OPTIONAL,
     extensionContainer                   [5] ExtensionContainer            OPTIONAL,
     ...}
```

```
LCS-PrivacyExceptionList ::= SEQUENCE SIZE (1..maxNumOfPrivacyClass) OF
                                     LCS-PrivacyClass
```

```
maxNumOfPrivacyClass  INTEGER ::= 4
```

```
LCS-PrivacyClass ::= SEQUENCE {
     ss-Code                              SS-Code,
     ss-Status                            Ext-SS-Status,
     notificationToMSUser                 [0] NotificationToMSUser          OPTIONAL,
     -- notificationToMSUser may be sent only for SS-codes callrelated
     -- and callunrelated. If not received for SS-codes callrelated and callunrelated,
     -- the default values according to 3GPP TS 23.171 and 3GPP TS 23.171 shall be assumed.
     externalClientList                   [1] ExternalClientList            OPTIONAL,
     -- externalClientList may be sent only for SS-code callunrelated.
     plmnClientList                       [2] PLMNClientList                OPTIONAL,
     -- plmnClientList may be sent only for SS-code plmnoperator.
     extensionContainer                   [3] ExtensionContainer            OPTIONAL,
     -- if segmentation is used, the complete LCS-PrivacyClass shall be sent in one segment
     ...}
```

```
ExternalClientList ::= SEQUENCE SIZE (0..maxNumOfExternalClient) OF
                                     ExternalClient
```

```
maxNumOfExternalClient  INTEGER ::= 5
```

```
PLMNClientList ::= SEQUENCE SIZE (1..maxNumOfPLMNClient) OF
                                     LCSClientInternalID
```

```
maxNumOfPLMNClient  INTEGER ::= 5
```

```
ExternalClient ::= SEQUENCE {
     clientIdentity                       LCSClientExternalID,
     gmlc-Restriction                     [0] GMLC-Restriction              OPTIONAL,
     notificationToMSUser                 [1] NotificationToMSUser          OPTIONAL,
     -- If notificationToMSUser is not received, the default value according to
     -- 3GPP TS 23.171 and 3GPP TS 23.171 shall be assumed.
     extensionContainer                   [2] ExtensionContainer            OPTIONAL,
     ... }
```

```
GMLC-Restriction ::= ENUMERATED {
     gmlc-List                            (0),
     home-Country                         (1) ,
     ... }
-- exception handling:
-- At reception of any other value than the ones listed the receiver shall ignore
-- GMLC-Restriction.
```

```
NotificationToMSUser ::= ENUMERATED {
     notifyLocationAllowed                (0),
     notifyAndVerify-LocationAllowedIfNoResponse   (1),
     notifyAndVerify-LocationNotAllowedIfNoResponse(2),
     ... }
-- exception handling:
-- At reception of any other value than the ones listed the receiver shall ignore
-- NotificationToMSUser.
```

```
MOLR-List ::= SEQUENCE SIZE (1..maxNumOfMOLR-Class) OF
                                       MOLR-Class
```

```
maxNumOfMOLR-Class  INTEGER ::= 3
```

```
MOLR-Class ::= SEQUENCE {
     ss-Code                          SS-Code,
     ss-Status                        Ext-SS-Status,
     extensionContainer               [0] ExtensionContainer            OPTIONAL,
     ...}
```

```
ZoneCodeList ::= SEQUENCE SIZE (1..maxNumOfZoneCodes)
                                       OF ZoneCode
```

```
ZoneCode ::= OCTET STRING (SIZE (2))
     -- internal structure is defined in 3GPP TS 23.003
```

```
maxNumOfZoneCodes  INTEGER ::= 10
```

```
InsertSubscriberDataRes ::= SEQUENCE {
     teleserviceList                  [1] TeleserviceList               OPTIONAL,
     bearerServiceList                [2] BearerServiceList             OPTIONAL,
     ss-List                          [3] SS-List                       OPTIONAL,
     odb-GeneralData                  [4] ODB-GeneralData               OPTIONAL,
     regionalSubscriptionResponse     [5] RegionalSubscriptionResponse   OPTIONAL,
     supportedCamelPhases             [6] SupportedCamelPhases          OPTIONAL,
     extensionContainer               [7] ExtensionContainer            OPTIONAL,
     ...}
```

```
RegionalSubscriptionResponse ::= ENUMERATED {
     networkNode-AreaRestricted       (0),
     tooManyZoneCodes                 (1),
     zoneCodesConflict                (2),
     regionalSubscNotSupported        (3)}
```

```
DeleteSubscriberDataArg ::= SEQUENCE {
     imsi                             [0] IMSI,
     basicServiceList                 [1] BasicServiceList              OPTIONAL,
     -- The exception handling for reception of unsupported/not allocated
     -- basicServiceCodes is defined in section 6.8.2
     ss-List                          [2] SS-List                       OPTIONAL,
     roamingRestrictionDueToUnsupportedFeature [4] NULL                 OPTIONAL,
     regionalSubscriptionIdentifier   [5] ZoneCode                      OPTIONAL,
     vbsGroupIndication               [7] NULL                          OPTIONAL,
     vgcsGroupIndication              [8] NULL   OPTIONAL,
     camelSubscriptionInfoWithdraw    [9] NULL   OPTIONAL,
     extensionContainer               [6] ExtensionContainer OPTIONAL,
     ...,
     gprsSubscriptionDataWithdraw     [10] GPRSSubscriptionDataWithdraw OPTIONAL,
     roamingRestrictedInSgsnDueToUnsuppportedFeature [11] NULL          OPTIONAL,
     lsaInformationWithdraw           [12] LSAInformationWithdraw       OPTIONAL,
     gmlc-ListWithdraw                [13] NULL                         OPTIONAL,
     istInformationWithdraw           [14] NULL                         OPTIONAL,
     specificCSI-Withdraw             [15] SpecificCSI-Withdraw         OPTIONAL }
```

```
SpecificCSI-Withdraw ::= BIT STRING {
     o-csi (0),
     ss-csi (1),
     tif-csi (2),
     d-csi (3),
     vt-csi (4),
     sms-csi (5),
     m-csi (6),
     gprs-csi(7),
     t-csi (8)} (SIZE(8..32))
-- exception handling:
-- bits 9 to 31 shall be ignored if received.
-- Bit 8 is only applicable for the NoteSubscriberDataModified operation.
```

```
GPRSSubscriptionDataWithdraw ::= CHOICE {
     allGPRSData                         NULL,
     contextIdList                       ContextIdList}
```

```
ContextIdList ::= SEQUENCE SIZE (1..maxNumOfPDP-Contexts) OF
                                         ContextId
```

```
LSAInformationWithdraw ::= CHOICE {
     allLSAData                          NULL,
     lsaIdentityList                     LSAIdentityList }
```

```
LSAIdentityList ::= SEQUENCE SIZE (1..maxNumOfLSAs) OF
                                         LSAIdentity
```

```
BasicServiceList ::= SEQUENCE SIZE (1..maxNumOfBasicServices) OF
                                         Ext-BasicServiceCode
```

```
maxNumOfBasicServices  INTEGER ::= 70
```

```
DeleteSubscriberDataRes ::= SEQUENCE {
     regionalSubscriptionResponse        [0] RegionalSubscriptionResponse  OPTIONAL,
     extensionContainer                  ExtensionContainer                OPTIONAL,
     ...}
```

```
VlrCamelSubscriptionInfo ::= SEQUENCE {
     o-CSI                               [0] O-CSI                         OPTIONAL,
     extensionContainer                  [1] ExtensionContainer            OPTIONAL,
     ...,
     ss-CSI                              [2] SS-CSI                        OPTIONAL,
     o-BcsmCamelTDP-CriteriaList         [4] O-BcsmCamelTDPCriteriaList    OPTIONAL,
     tif-CSI                             [3] NULL                          OPTIONAL,
     m-CSI                               [5] M-CSI                         OPTIONAL,
     sms-CSI                             [6] SMS-CSI                       OPTIONAL,
     vt-CSI                              [7] T-CSI                         OPTIONAL,
     t-BCSM-CAMEL-TDP-CriteriaList       [8] T-BCSM-CAMEL-TDP-CriteriaList OPTIONAL,
     d-CSI                               [9] D-CSI                         OPTIONAL}
```

```
D-CSI ::= SEQUENCE {
     dp-AnalysedInfoCriteriaList         [0] DP-AnalysedInfoCriteriaList   OPTIONAL,
     camelCapabilityHandling             [1] CamelCapabilityHandling       OPTIONAL,
     extensionContainer                  [2] ExtensionContainer            OPTIONAL,
     notificationToCSE                   [3]  NULL                         OPTIONAL,
     csi-Active                          [4]  NULL                         OPTIONAL,
     ...}
--   notificationToCSE and csi-Active shall not be present when D-CSI is sent to VLR/GMSC.
--   They may only be included in ATSI/ATM ack/NSDC message.
--   DP-AnalysedInfoCriteria and  camelCapabilityHandling shall be present in
--   the D-CSI sequence.
--   If D-CSI is segmented, dp-AnalysedInfoCriteriaList and camelCapabilityHandling shall be
--   present in the first segment
```

```
DP-AnalysedInfoCriteriaList  ::= SEQUENCE SIZE (1..maxNumOfDP-AnalysedInfoCriteria) OF
                                         DP-AnalysedInfoCriterium
```

```
maxNumOfDP-AnalysedInfoCriteria INTEGER ::= 10
```

```
DP-AnalysedInfoCriterium ::= SEQUENCE {
     dialledNumber                    ISDN-AddressString,
     serviceKey                       ServiceKey,
     gsmSCF-Address                   ISDN-AddressString,
     defaultCallHandling              DefaultCallHandling,
     extensionContainer               ExtensionContainer              OPTIONAL,
     ...}
```

```
SS-CSI ::= SEQUENCE {
     ss-CamelData                     SS-CamelData,
     extensionContainer               ExtensionContainer              OPTIONAL,
     ...,
     notificationToCSE                [0]  NULL                       OPTIONAL,
     csi-Active                       [1]  NULL                       OPTIONAL
--   notificationToCSE and csi-Active shall not be present when SS-CSI is sent to VLR.
--   They may only be included in ATSI/ATM ack/NSDC message.
}
```

```
SS-CamelData  ::= SEQUENCE {
     ss-EventList                     SS-EventList,
     gsmSCF-Address                   ISDN-AddressString,
     extensionContainer               [0] ExtensionContainer          OPTIONAL,
     ... }
```

```
SS-EventList  ::= SEQUENCE SIZE (1..maxNumOfCamelSSEvents) OF SS-Code
     -- Actions for the following SS-Code values are defined in CAMEL Phase 3:
     -- ect                           SS-Code ::= '00110001'B
     -- multiPTY                      SS-Code ::= '01010001'B
     -- cd                            SS-Code ::= '00100100'B
     -- ccbs                          SS-Code ::= '01000100'B
     -- all other SS codes shall be ignored
     -- When SS-CSI is sent to the VLR, it shall not contain a marking for ccbs.
     -- If the VLR receives SS-CSI containing a marking for ccbs, the VLR shall discard the
     -- ccbs marking in SS-CSI.
```

```
maxNumOfCamelSSEvents INTEGER ::= 10
```

```
O-CSI ::= SEQUENCE {
     o-BcsmCamelTDPDataList           O-BcsmCamelTDPDataList,
     extensionContainer               ExtensionContainer              OPTIONAL,
     ...,
     camelCapabilityHandling          [0] CamelCapabilityHandling     OPTIONAL,
     notificationToCSE                [1]  NULL                       OPTIONAL,
     csiActive                        [2]  NULL                       OPTIONAL}
--   notificationtoCSE and csiActive shall not be present when O-CSI is sent to VLR/GMSC.
--   They may only be included in ATSI/ATM ack/NSDC message.
--   O-CSI shall not be segmented.
```

```
O-BcsmCamelTDPDataList ::= SEQUENCE SIZE (1..maxNumOfCamelTDPData) OF
     O-BcsmCamelTDPData
     -- O-BcsmCamelTDPDataList shall not contain more than one instance of
     -- O-BcsmCamelTDPData containing the same value for o-BcsmTriggerDetectionPoint.
     -- For CAMEL Phase 2, this means that only one instance of O-BcsmCamelTDPData is allowed
     -- with o-BcsmTriggerDetectionPoint being equal to DP2.
```

```
maxNumOfCamelTDPData  INTEGER ::= 10
```

```
O-BcsmCamelTDPData ::= SEQUENCE {
     o-BcsmTriggerDetectionPoint      O-BcsmTriggerDetectionPoint,
     serviceKey                       ServiceKey,
     gsmSCF-Address                   [0] ISDN-AddressString,
     defaultCallHandling              [1] DefaultCallHandling,
     extensionContainer               [2] ExtensionContainer          OPTIONAL,
     ...
     }
```

```
ServiceKey ::= INTEGER (0..2147483647)
```

```
O-BcsmTriggerDetectionPoint ::= ENUMERATED {
     collectedInfo (2),
     ...,
     routeSelectFailure (4) }
     -- exception handling:
     -- For O-BcsmCamelTDPData sequences containing this parameter with any
     -- other value than the ones listed the receiver shall ignore the whole
     -- O-BcsmCamelTDPDatasequence.
     -- For O-BcsmCamelTDP-Criteria sequences containing this parameter with any
     -- other value than the ones listed the receiver shall ignore the whole
     -- O-BcsmCamelTDP-Criteria sequence.
```

```
O-BcsmCamelTDPCriteriaList ::= SEQUENCE SIZE (1..maxNumOfCamelTDPData) OF
     O-BcsmCamelTDP-Criteria
```

```
T-BCSM-CAMEL-TDP-CriteriaList ::= SEQUENCE SIZE (1..maxNumOfCamelTDPData) OF
     T-BCSM-CAMEL-TDP-Criteria
```

```
O-BcsmCamelTDP-Criteria ::= SEQUENCE {
     o-BcsmTriggerDetectionPoint          O-BcsmTriggerDetectionPoint,
     destinationNumberCriteria            [0] DestinationNumberCriteria     OPTIONAL,
     basicServiceCriteria                 [1] BasicServiceCriteria          OPTIONAL,
     callTypeCriteria                     [2] CallTypeCriteria              OPTIONAL,
     ...,
     o-CauseValueCriteria                 [3] O-CauseValueCriteria          OPTIONAL,
     extensionContainer                   [4] ExtensionContainer            OPTIONAL }
```

```
T-BCSM-CAMEL-TDP-Criteria ::= SEQUENCE {
     t-BCSM-TriggerDetectionPoint         T-BcsmTriggerDetectionPoint,
     basicServiceCriteria                 [0] BasicServiceCriteria          OPTIONAL,
     t-CauseValueCriteria                 [1] T-CauseValueCriteria          OPTIONAL,
     ... }
```

```
DestinationNumberCriteria  ::= SEQUENCE {
     matchType                            [0] MatchType,
     destinationNumberList                [1] DestinationNumberList         OPTIONAL,
     destinationNumberLengthList          [2] DestinationNumberLengthList   OPTIONAL,
     -- one or both of destinationNumberList and destinationNumberLengthList
     -- shall be present
     ...}
```

```
DestinationNumberList  ::= SEQUENCE SIZE   (1..maxNumOfCamelDestinationNumbers) OF
                                    ISDN-AddressString
     -- The receiving entity shall not check the format of a number in
     -- the dialled number list
```

```
DestinationNumberLengthList  ::= SEQUENCE SIZE (1..maxNumOfCamelDestinationNumberLengths)
OF
                                        INTEGER(1..maxNumOfISDN-AddressDigits)
```

```
BasicServiceCriteria   ::= SEQUENCE SIZE(1..maxNumOfCamelBasicServiceCriteria) OF
     Ext-BasicServiceCode
```

```
maxNumOfISDN-AddressDigits  INTEGER ::= 15
```

```
maxNumOfCamelDestinationNumbers  INTEGER ::= 10
```

```
maxNumOfCamelDestinationNumberLengths  INTEGER ::= 3
```

```
maxNumOfCamelBasicServiceCriteria  INTEGER ::= 5
```

```
CallTypeCriteria       ::= ENUMERATED {
     forwarded                            (0),
     notForwarded                         (1)}
```

```
MatchType       ::= ENUMERATED {
     inhibiting                           (0),
     enabling                             (1)}
```

```
O-CauseValueCriteria   ::= SEQUENCE SIZE(1..maxNumOfCAMEL-O-CauseValueCriteria) OF
     CauseValue
```

```
T-CauseValueCriteria   ::= SEQUENCE SIZE(1..maxNumOfCAMEL-T-CauseValueCriteria) OF
     CauseValue
```

```
maxNumOfCAMEL-O-CauseValueCriteria  INTEGER ::= 5
```

```
maxNumOfCAMEL-T-CauseValueCriteria  INTEGER ::= 5
```

```
CauseValue ::= OCTET STRING (SIZE(1))
-- Type extracted from Cause parameter in ITU-T Recommendation Q.763.
-- For the use of cause value refer to ITU-T Recommendation Q.850.
```

```
DefaultCallHandling ::= ENUMERATED {
     continueCall (0) ,
     releaseCall (1) ,
     ...}
     -- exception handling:
     -- reception of values in range 2-31 shall be treated as "continueCall"
     -- reception of values greater than 31 shall be treated as "releaseCall"
```

```
CamelCapabilityHandling ::= INTEGER(1..16)
     -- value 1 = CAMEL phase 1,
     -- value 2 = CAMEL phase 2,
     -- value 3 = CAMEL Phase 3:
     -- reception of values greater than 3 shall be treated as CAMEL phase 3.
```

```
SupportedCamelPhases ::= BIT STRING {
     phase1 (0),
     phase2 (1) ,
     phase3 (2) } (SIZE (1..16))
-- A node shall mark in the BIT STRING all CAMEL Phases it supports.
-- Other values than listed above shall be discarded.
```

```
SMS-CSI ::= SEQUENCE {
     sms-CAMEL-TDP-DataList              [0] SMS-CAMEL-TDP-DataList        OPTIONAL,
     camelCapabilityHandling             [1] CamelCapabilityHandling       OPTIONAL,
     extensionContainer                  [2] ExtensionContainer            OPTIONAL,
     notificationToCSE                   [3] NULL                          OPTIONAL,
     csi-Active                          [4] NULL                          OPTIONAL,
     ...}
--   notificationToCSE and csi-Active shall not be present when SMS-CSI is sent to VLR/SGSN.
--   They may only be included in ATSI/ATM ack/NSDC message.
--   SMS-CAMEL-TDP-Data and  camelCapabilityHandling shall be present in
--   the SMS-CSI sequence.
--   If SMS-CSI is segmented, sms-CAMEL-TDP-DataList and camelCapabilityHandling shall be
--   present in the first segment
```

```
SMS-CAMEL-TDP-DataList ::= SEQUENCE SIZE (1..maxNumOfCamelTDPData) OF
     SMS-CAMEL-TDP-Data
--   SMS-CAMEL-TDP-DataList shall not contain more than one instance of
--   SMS-CAMEL-TDP-Data containing the same value for sms-TriggerDetectionPoint.
```

```
SMS-CAMEL-TDP-Data ::= SEQUENCE {
     sms-TriggerDetectionPoint           [0] SMS-TriggerDetectionPoint,
     serviceKey                          [1] ServiceKey,
     gsmSCF-Address                      [2] ISDN-AddressString,
     defaultSMS-Handling                 [3] DefaultSMS-Handling,
     extensionContainer                  [4] ExtensionContainer            OPTIONAL,
     ...
     }
```

```
SMS-TriggerDetectionPoint ::= ENUMERATED {
     sms-CollectedInfo (1),
     ... }
--   exception handling:
--   For SMS-CAMEL-TDP-Data sequences containing this parameter with any
--   other value than the ones listed the receiver shall ignore the whole
--   SMS-CAMEL-TDP-Data sequence.
```

```
DefaultSMS-Handling ::= ENUMERATED {
     continueTransaction (0) ,
     releaseTransaction (1) ,
     ...}
--   exception handling:
--   reception of values in range 2-31 shall be treated as "continueTransaction"
--   reception of values greater than 31 shall be treated as "releaseTransaction"
```

```
M-CSI ::= SEQUENCE {
     mobilityTriggers                    MobilityTriggers,
     serviceKey                          ServiceKey,
     gsmSCF-Address                      [0]  ISDN-AddressString,
     extensionContainer                  [1]  ExtensionContainer          OPTIONAL,
     notificationToCSE                   [2] NULL                         OPTIONAL,
     csi-Active                          [3] NULL                         OPTIONAL,
     ...}
--   notificationToCSE and csi-Active shall not be present when M-CSI is sent to VLR.
--   They may only be included in ATSI/ATM ack/NSDC message.
```

```
MobilityTriggers  ::= SEQUENCE SIZE (1..maxNumOfMobilityTriggers) OF
     MM-Code
```

```
maxNumOfMobilityTriggers INTEGER ::= 10
```

```
MM-Code ::= OCTET STRING (SIZE (1))
--   This type is used to indicate a Mobility Management event.
--   Actions for the following M-Code values are defined in CAMEL Phase 3:
--
--   Location-update-in-same-VLR         MM-Code ::= '00000000'B
--   Location-update-to-other-VLR        MM-Code ::= '00000001'B
--   IMSI-Attach                         MM-Code ::= '00000010'B
--   MS-initiated-IMSI-Detach            MM-Code ::= '00000011'B
--   Network-initiated-IMSI-Detach       MM-Code ::= '00000100'B
--
--   If any other MM-code is received in M-CSI, then that MM-code shall be
--   ignored.
```

```
T-CSI ::= SEQUENCE {
     t-BcsmCamelTDPDataList              T-BcsmCamelTDPDataList,
     extensionContainer                  ExtensionContainer               OPTIONAL,
     ...,
     camelCapabilityHandling             [0] CamelCapabilityHandling      OPTIONAL,
     notificationToCSE                   [1] NULL                         OPTIONAL,
     csi-Active                          [2] NULL                         OPTIONAL}
--   notificationToCSE and csi-Active shall not be present when VT-CSI/T-CSI is sent
--   to VLR/GMSC.
--   They may only be included in ATSI/ATM ack/NSDC message.
--   T-CSI shall not be segmented.
```

```
T-BcsmCamelTDPDataList ::= SEQUENCE SIZE (1..maxNumOfCamelTDPData) OF
     T-BcsmCamelTDPData
     --- T-BcsmCamelTDPDataList shall not contain more than one instance of
     --- T-BcsmCamelTDPData containing the same value for t-BcsmTriggerDetectionPoint.
     --- For CAMEL Phase 2, this means that only one instance of T-BcsmCamelTDPData is
allowed
     --- with t-BcsmTriggerDetectionPoint being equal to DP12.
     --- For CAMEL Phase 3, more TDP's are allowed.
```

```
T-BcsmCamelTDPData ::= SEQUENCE {
     t-BcsmTriggerDetectionPoint         T-BcsmTriggerDetectionPoint,
     serviceKey                          ServiceKey,
     gsmSCF-Address                      [0] ISDN-AddressString,
     defaultCallHandling                 [1] DefaultCallHandling,
     extensionContainer                  [2] ExtensionContainer           OPTIONAL,
     ...}
```

```
T-BcsmTriggerDetectionPoint ::= ENUMERATED {
     termAttemptAuthorized (12),
     ... ,
     tBusy (13),
     tNoAnswer (14)}
     -- exception handling:
     -- For T-BcsmCamelTDPData sequences containing this parameter with any other
     -- value than the ones listed above, the receiver shall ignore the whole
     -- T-BcsmCamelTDPData sequence.
```

```
-- gprs location information retrieval types
```

```
SendRoutingInfoForGprsArg ::= SEQUENCE {
    imsi                                [0] IMSI,
    ggsn-Address                        [1] GSN-Address              OPTIONAL,
    ggsn-Number                         [2]  ISDN-AddressString,
    extensionContainer                  [3] ExtensionContainer       OPTIONAL,
    ...}
```

```
SendRoutingInfoForGprsRes ::= SEQUENCE {
    sgsn-Address                        [0] GSN-Address,
    ggsn-Address                        [1]  GSN-Address             OPTIONAL,
    mobileNotReachableReason            [2]  AbsentSubscriberDiagnosticSM    OPTIONAL,
    extensionContainer                  [3] ExtensionContainer       OPTIONAL,
    ...}
```

-- failure report types

```
FailureReportArg ::= SEQUENCE {
    imsi                                [0] IMSI,
    ggsn-Number                         [1] ISDN-AddressString       ,
    ggsn-Address                        [2] GSN-Address              OPTIONAL,
    extensionContainer                  [3] ExtensionContainer       OPTIONAL,
    ...}
```

```
FailureReportRes ::= SEQUENCE {
    ggsn-Address                        [0] GSN-Address              OPTIONAL,
    extensionContainer                  [1] ExtensionContainer       OPTIONAL,
    ...}
```

-- gprs notification types

```
NoteMsPresentForGprsArg ::= SEQUENCE {
    imsi                                [0] IMSI,
    sgsn-Address                        [1] GSN-Address,
    ggsn-Address                        [2] GSN-Address              OPTIONAL,
    extensionContainer                  [3] ExtensionContainer       OPTIONAL,
    ...}
```

```
NoteMsPresentForGprsRes ::= SEQUENCE {
    extensionContainer                  [0] ExtensionContainer       OPTIONAL,
    ...}
```

-- fault recovery types

```
ResetArg ::= SEQUENCE {
    hlr-Number                          ISDN-AddressString,
    hlr-List                            HLR-List                     OPTIONAL,
    ...}
```

```
RestoreDataArg ::= SEQUENCE {
    imsi                                IMSI,
    lmsi                                LMSI                         OPTIONAL,
    extensionContainer                  ExtensionContainer           OPTIONAL,
    ... ,
    vlr-Capability                      [6] VLR-Capability           OPTIONAL }
```

```
RestoreDataRes ::= SEQUENCE {
    hlr-Number                          ISDN-AddressString,
    msNotReachable                      NULL                         OPTIONAL,
    extensionContainer                  ExtensionContainer           OPTIONAL,
    ...}
```

-- VBS/VGCS types

```
VBSDataList ::= SEQUENCE SIZE (1..maxNumOfVBSGroupIds) OF
                                        VoiceBroadcastData
```

```
VGCSDataList ::= SEQUENCE SIZE (1..maxNumOfVGCSGroupIds) OF
                                        VoiceGroupCallData
```

```
maxNumOfVBSGroupIds  INTEGER ::= 50
```

```
maxNumOfVGCSGroupIds  INTEGER ::= 50
```

```
VoiceGroupCallData  ::= SEQUENCE {
     groupId                          GroupId,
     extensionContainer               ExtensionContainer              OPTIONAL,
     ...}
```

```
VoiceBroadcastData ::= SEQUENCE {
     groupid                          GroupId,
     broadcastInitEntitlement         NULL                            OPTIONAL,
     extensionContainer               ExtensionContainer              OPTIONAL,
     ...}
```

```
GroupId  ::= TBCD-STRING (SIZE (3))
     -- When Group-Id is less than six characters in length, the TBCD filler (1111)
     -- is used to fill unused half octets.
     -- Refers to the Group Identification as specified in 3GPP TS 23.003
     -- and GSM TS 03.68/ 03.69
```

-- provide subscriber info types

```
ProvideSubscriberInfoArg ::= SEQUENCE {
     imsi      [0] IMSI,
     lmsi      [1] LMSI               OPTIONAL,
     requestedInfo                    [2] RequestedInfo,
     extensionContainer               [3] ExtensionContainer          OPTIONAL,
     ...}
```

```
ProvideSubscriberInfoRes ::= SEQUENCE {
     subscriberInfo                   SubscriberInfo,
     extensionContainer               ExtensionContainer              OPTIONAL,
     ...}
```

```
SubscriberInfo ::= SEQUENCE {
     locationInformation              [0] LocationInformation         OPTIONAL,
     subscriberState                  [1] SubscriberState             OPTIONAL,
     extensionContainer               [2] ExtensionContainer          OPTIONAL,
     ...}
```

```
RequestedInfo ::= SEQUENCE {
     locationInformation              [0] NULL                        OPTIONAL,
     subscriberState                  [1] NULL                        OPTIONAL,
     extensionContainer               [2] ExtensionContainer          OPTIONAL,
     ...,
     currentLocation                  [3] NULL                        OPTIONAL }
-- currentLocation shall not be present if locationInformation
-- is not present in the RequestedInfo parameter
```

```
LocationInformation ::= SEQUENCE {
     ageOfLocationInformation         AgeOfLocationInformation        OPTIONAL,
     geographicalInformation          [0] GeographicalInformation     OPTIONAL,
     vlr-number                       [1] ISDN-AddressString          OPTIONAL,
     locationNumber                   [2] LocationNumber              OPTIONAL,
     cellGlobalIdOrServiceAreaIdOrLAI [3] CellGlobalIdOrServiceAreaIdOrLAI  OPTIONAL,
     extensionContainer               [4] ExtensionContainer          OPTIONAL,
     ... ,
     selectedLSA-Id                   [5] LSAIdentity                 OPTIONAL,
     msc-Number                       [6] ISDN-AddressString          OPTIONAL,
     geodeticInformation              [7] GeodeticInformation         OPTIONAL,
     currentLocationRetrieved         [8] NULL                        OPTIONAL,
     sai-Present                      [9] NULL                        OPTIONAL }
-- sai-Present indicates that the cellGlobalIdOrServiceAreaIdOrLAI parameter contains
-- a Service Area Identity.
-- currentLocationRetrieved shall be present
-- if the location information were retrieved after a successfull paging.
```

```
GeographicalInformation ::= OCTET STRING (SIZE (8))
--   Refers to geographical Information defined in 3GPP TS 23.032.
--   Only the description of an ellipsoid point with uncertainty circle
--   as specified in 3GPP TS 23.032 is allowed to be used
--   The internal structure according to 3GPP TS 23.032 is as follows:
--        Type of shape (ellipsoid point with uncertainty circle)      1 octet
--        Degrees of Latitude                                          3 octets
--        Degrees of Longitude                                         3 octets
--        Uncertainty code                                             1 octet
```

```
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
```

```
LocationNumber ::= OCTET STRING (SIZE (2..10))
     -- the internal structure is defined in CCITT Rec Q.763
```

```
SubscriberState ::= CHOICE {
     assumedIdle                    [0] NULL,
     camelBusy [1] NULL,
     netDetNotReachable             NotReachableReason,
     notProvidedFromVLR             [2] NULL}
```

```
NotReachableReason ::= ENUMERATED {
     msPurged (0),
     imsiDetached (1),
     restrictedArea (2),
     notRegistered (3)}
```

-- any time interrogation info types

```
AnyTimeInterrogationArg ::= SEQUENCE {
     subscriberIdentity             [0] SubscriberIdentity,
     requestedInfo                  [1] RequestedInfo,
     gsmSCF-Address                 [3] ISDN-AddressString,
     extensionContainer             [2] ExtensionContainer          OPTIONAL,
     ...}
```

```
AnyTimeInterrogationRes ::= SEQUENCE {
     subscriberInfo                 SubscriberInfo,
     extensionContainer             ExtensionContainer              OPTIONAL,
     ...}
```

-- any time information handling types

```
AnyTimeSubscriptionInterrogationArg ::= SEQUENCE {
     subscriberIdentity             [0] SubscriberIdentity,
     requestedSubscriptionInfo      [1] RequestedSubscriptionInfo,
     gsmSCF-Address                 [2] ISDN-AddressString,
     extensionContainer             [3] ExtensionContainer          OPTIONAL,
     longFTN-Supported              [4]  NULL                       OPTIONAL,
     ...}
```

```
AnyTimeSubscriptionInterrogationRes ::= SEQUENCE {
     callForwardingData             [1] CallForwardingData          OPTIONAL,
     callBarringData                [2] CallBarringData             OPTIONAL,
     odb-Info                       [3] ODB-Info                    OPTIONAL,
     camel-SubscriptionInfo         [4] CAMEL-SubscriptionInfo      OPTIONAL,
     supportedVLR-CAMEL-Phases      [5] SupportedCamelPhases        OPTIONAL,
     supportedSGSN-CAMEL-Phases     [6] SupportedCamelPhases        OPTIONAL,
     extensionContainer             [7] ExtensionContainer          OPTIONAL,
     ...}
```

```
RequestedSubscriptionInfo ::= SEQUENCE {
     requestedSS-Info               [1] SS-ForBS-Code               OPTIONAL,
     odb                            [2] NULL                        OPTIONAL,
     requestedCAMEL-SubscriptionInfo [3] RequestedCAMEL-SubscriptionInfo   OPTIONAL,
     supportedVLR-CAMEL-Phases      [4] NULL                        OPTIONAL,
     supportedSGSN-CAMEL-Phases     [5] NULL                        OPTIONAL,
     extensionContainer             [6] ExtensionContainer          OPTIONAL,
     ...}
```

```
RequestedCAMEL-SubscriptionInfo ::= 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)}
```

```
CallForwardingData ::= SEQUENCE {
    forwardingFeatureList           Ext-ForwFeatureList,
    notificationToCSE               NULL                            OPTIONAL,
    extensionContainer              [0] ExtensionContainer          OPTIONAL,
    ...}
```

```
CallBarringData ::= SEQUENCE {
    callBarringFeatureList          Ext-CallBarFeatureList,
    password                        Password                        OPTIONAL,
    wrongPasswordAttemptsCounter    WrongPasswordAttemptsCounter    OPTIONAL,
    notificationToCSE               NULL                            OPTIONAL,
    extensionContainer              ExtensionContainer              OPTIONAL,
    ...}
```

```
WrongPasswordAttemptsCounter ::= INTEGER (0..4)
```

```
ODB-Info ::= SEQUENCE {
    odb-Data                        ODB-Data,
    notificationToCSE               NULL                            OPTIONAL,
    extensionContainer              ExtensionContainer              OPTIONAL,
    ...}
```

```
CAMEL-SubscriptionInfo ::= SEQUENCE {
    o-CSI                           [0]  O-CSI                          OPTIONAL,
    o-BcsmCamelTDP-CriteriaList     [1]  O-BcsmCamelTDPCriteriaList     OPTIONAL,
    d-CSI                           [2]  D-CSI                          OPTIONAL,
    t-CSI                           [3]  T-CSI                          OPTIONAL,
    t-BCSM-CAMEL-TDP-CriteriaList   [4]  T-BCSM-CAMEL-TDP-CriteriaList OPTIONAL,
    vt-CSI                          [5]  T-CSI                          OPTIONAL,
    vt-BCSM-CAMEL-TDP-CriteriaList  [6]  T-BCSM-CAMEL-TDP-CriteriaList OPTIONAL,
    tif-CSI                         [7]  NULL                           OPTIONAL,
    tif-CSI-NotificationToCSE       [8]  NULL                           OPTIONAL,
    gprs-CSI                        [9]  GPRS-CSI                       OPTIONAL,
    sms-CSI                         [10] SMS-CSI                        OPTIONAL,
    ss-CSI                          [11] SS-CSI                         OPTIONAL,
    m-CSI                           [12] M-CSI                          OPTIONAL,
    extensionContainer              [13] ExtensionContainer             OPTIONAL,
    ...,
    specificCSIDeletedList          [14] SpecificCSI-Withdraw           OPTIONAL }
```

```
AnyTimeModificationArg ::= SEQUENCE {
    subscriberIdentity              [0]  SubscriberIdentity,
    gsmSCF-Address                  [1]  ISDN-AddressString,
    modificationRequestFor-CF-Info  [2]  ModificationRequestFor-CF-Info OPTIONAL,
    modificationRequestFor-CB-Info  [3]  ModificationRequestFor-CB-Info OPTIONAL,
    modificationRequestFor-CSI      [4]  ModificationRequestFor-CSI     OPTIONAL,
    extensionContainer              [5]  ExtensionContainer             OPTIONAL,
    longFTN-Supported               [6]  NULL                           OPTIONAL,
    ...}
```

```
AnyTimeModificationRes ::= SEQUENCE {
    ss-InfoFor-CSE                  [0]  Ext-SS-InfoFor-CSE             OPTIONAL,
    camel-SubscriptionInfo          [1]  CAMEL-SubscriptionInfo         OPTIONAL,
    extensionContainer              [2]  ExtensionContainer             OPTIONAL,
    ...}
```

```
ModificationRequestFor-CF-Info ::= SEQUENCE {
     ss-Code                              [0]  SS-Code,
     basicService                         [1]  Ext-BasicServiceCode          OPTIONAL,
     ss-Status                            [2]  Ext-SS-Status                 OPTIONAL,
     forwardedToNumber                    [3]  AddressString                 OPTIONAL,
     forwardedToSubaddress                [4]  ISDN-SubaddressString         OPTIONAL,
     noReplyConditionTime                 [5]  Ext-NoRepCondTime             OPTIONAL,
     modifyNotificationToCSE              [6]  ModificationInstruction       OPTIONAL,
     extensionContainer                   [7]  ExtensionContainer            OPTIONAL,
     ...}
```

```
ModificationRequestFor-CB-Info ::= SEQUENCE {
     ss-Code                              [0]  SS-Code,
     basicService                         [1]  Ext-BasicServiceCode          OPTIONAL,
     ss-Status                            [2]  Ext-SS-Status                 OPTIONAL,
     password                             [3]  Password                      OPTIONAL,
     wrongPasswordAttemptsCounter         [4]  WrongPasswordAttemptsCounter  OPTIONAL,
     modifyNotificationToCSE              [5]  ModificationInstruction       OPTIONAL,
     extensionContainer                   [6]  ExtensionContainer            OPTIONAL,
     ...}
```

```
ModificationRequestFor-CSI ::= SEQUENCE {
     requestedCamel-SubscriptionInfo      [0]  RequestedCAMEL-SubscriptionInfo,
     modifyNotificationToCSE              [1]  ModificationInstruction       OPTIONAL,
     modifyCSI-State                      [2]  ModificationInstruction       OPTIONAL,
     extensionContainer                   [3]  ExtensionContainer            OPTIONAL,
     ...}
```

```
ModificationInstruction ::= ENUMERATED {
     deactivate                           (0),
     activate                             (1)}
```

-- subscriber data modification notification types

```
NoteSubscriberDataModifiedArg ::= SEQUENCE {
     imsi                                 IMSI,
     msisdn                               ISDN-AddressString,
     forwardingInfoFor-CSE                [0] Ext-ForwardingInfoFor-CSE      OPTIONAL,
     callBarringInfoFor-CSE               [1] Ext-CallBarringInfoFor-CSE     OPTIONAL,
     odb-Info                             [2] ODB-Info                       OPTIONAL,
     camel-SubscriptionInfo               [3] CAMEL-SubscriptionInfo         OPTIONAL,
     allInformationSent                   [4] NULL                           OPTIONAL,
     extensionContainer                   ExtensionContainer                 OPTIONAL,
     ...}
```

```
NoteSubscriberDataModifiedRes ::= SEQUENCE {
     extensionContainer                   ExtensionContainer                 OPTIONAL,
     ...}
```

-- mobility management event notificatioon info types

```
NoteMM-EventArg::= SEQUENCE {
     serviceKey                           ServiceKey,
     eventMet                             [0]  MM-Code,
     imsi                                 [1]  IMSI,
     msisdn                               [2]  ISDN-AddressString,
     locationInformation                  [3]  LocationInformation           OPTIONAL,
     supportedCAMELPhases                 [5]  SupportedCamelPhases          OPTIONAL,
     extensionContainer                   [6]  ExtensionContainer            OPTIONAL,
     ...}
```

```
NoteMM-EventRes ::= SEQUENCE {
     extensionContainer                   ExtensionContainer                 OPTIONAL,
     ...}
```

```
Ext-SS-InfoFor-CSE ::= CHOICE {
     forwardingInfoFor-CSE                [0] Ext-ForwardingInfoFor-CSE,
     callBarringInfoFor-CSE               [1] Ext-CallBarringInfoFor-CSE
     }
```

```
Ext-ForwardingInfoFor-CSE ::= SEQUENCE {
     ss-Code                              [0]  SS-Code,
     forwardingFeatureList                [1]  Ext-ForwFeatureList,
     notificationToCSE                    [2]  NULL                           OPTIONAL,
     extensionContainer                   [3]  ExtensionContainer             OPTIONAL,
     ...}
```

```
Ext-CallBarringInfoFor-CSE ::= SEQUENCE {
     ss-Code                              [0]  SS-Code,
     callBarringFeatureList               [1]  Ext-CallBarFeatureList,
     password                             [2]  Password                       OPTIONAL,
     wrongPasswordAttemptsCounter         [3]  WrongPasswordAttemptsCounter   OPTIONAL,
     notificationToCSE                    [4]  NULL                           OPTIONAL,
     extensionContainer                   [5]  ExtensionContainer             OPTIONAL,
     ...}
```

```
END
```

17.7.2 Operation and maintenance data types

```
MAP-OM-DataTypes {
   ccitt identified-organization (4) etsi (0) mobileDomain (0)
   gsm-Network (1) modules (3) map-OM-DataTypes (12) version6 (6)}

DEFINITIONS

IMPLICIT TAGS

::=

BEGIN

EXPORTS
   ActivateTraceModeArg,
   ActivateTraceModeRes,
   DeactivateTraceModeArg,
   DeactivateTraceModeRes
;

IMPORTS
   AddressString,
   IMSI
FROM MAP-CommonDataTypes {
   ccitt identified-organization (4) etsi (0) mobileDomain (0)
   gsm-Network (1) modules (3) map-CommonDataTypes (18) version6 (6)}

   ExtensionContainer
FROM MAP-ExtensionDataTypes {
   ccitt identified-organization (4) etsi (0) mobileDomain (0)
   gsm-Network (1) modules (3) map-ExtensionDataTypes (21) version6 (6)}


;
```

```
ActivateTraceModeArg ::= SEQUENCE {
     imsi                                 [0] IMSI                            OPTIONAL,
     traceReference                       [1] TraceReference,
     traceType                            [2] TraceType,
     omc-Id                               [3] AddressString                   OPTIONAL,
     extensionContainer                   [4] ExtensionContainer              OPTIONAL,
     ...}
```

```
TraceReference ::= OCTET STRING (SIZE (1..2))
```

```
TraceType ::= INTEGER
     (0..255)
     -- Trace types are fully defined in TS GSM 12.08.
```

```
ActivateTraceModeRes ::= SEQUENCE {
     extensionContainer                   [0] ExtensionContainer              OPTIONAL,
     ...}
```

```
DeactivateTraceModeArg ::= SEQUENCE {
     imsi                                [0] IMSI                            OPTIONAL,
     traceReference                      [1] TraceReference,
     extensionContainer                  [2] ExtensionContainer              OPTIONAL,
     ...}
```

```
DeactivateTraceModeRes ::= SEQUENCE {
     extensionContainer                  [0] ExtensionContainer              OPTIONAL,
     ...}
```

```
END
```

17.7.3 Call handling data types

```
MAP-CH-DataTypes {
   ccitt identified-organization (4) etsi (0) mobileDomain (0)
   gsm-Network (1) modules (3) map-CH-DataTypes (13) version6 (6)}

DEFINITIONS

IMPLICIT TAGS

::=

BEGIN

EXPORTS
   SendRoutingInfoArg,
   SendRoutingInfoRes,
   ProvideRoamingNumberArg,
   ProvideRoamingNumberRes,
   ResumeCallHandlingArg,
   ResumeCallHandlingRes,
   NumberOfForwarding,
   SuppressionOfAnnouncement,
   CallReferenceNumber,
   ProvideSIWFSNumberArg,
   ProvideSIWFSNumberRes,
   SIWFSSignallingModifyArg,
   SIWFSSignallingModifyRes,
   SetReportingStateArg,
   SetReportingStateRes,
   StatusReportArg,
   StatusReportRes,
   RemoteUserFreeArg,
   RemoteUserFreeRes,
   IST-AlertArg,
   IST-AlertRes,
   IST-CommandArg,
   IST-CommandRes
;

IMPORTS
   SubscriberInfo,
   SupportedCamelPhases,
   CUG-Interlock,
   O-CSI,
   D-CSI,
   O-BcsmCamelTDPCriteriaList,
   T-BCSM-CAMEL-TDP-CriteriaList,
   IST-SupportIndicator,
   IST-AlertTimerValue,
   T-CSI


FROM MAP-MS-DataTypes {
   ccitt identified-organization (4) etsi (0) mobileDomain (0)
   gsm-Network (1) modules (3) map-MS-DataTypes (11) version6 (6)}

   ForwardingOptions,
   SS-List,
   CCBS-Feature
FROM MAP-SS-DataTypes {
   ccitt identified-organization (4) etsi (0) mobileDomain (0)
   gsm-Network (1) modules (3) map-SS-DataTypes (14) version6 (6)}

   ISDN-AddressString,
   ISDN-SubaddressString,
```

```
   FTN-AddressString,
   ExternalSignalInfo,
   Ext-ExternalSignalInfo,
   IMSI,
   LMSI,
   Ext-BasicServiceCode,
   AlertingPattern,
   NAEA-PreferredCI


FROM MAP-CommonDataTypes {
   ccitt identified-organization (4) etsi (0) mobileDomain (0)
   gsm-Network (1) modules (3) map-CommonDataTypes (18) version6 (6)}

   ExtensionContainer
FROM MAP-ExtensionDataTypes {
   ccitt identified-organization (4) etsi (0) mobileDomain (0)
   gsm-Network (1) modules (3) map-ExtensionDataTypes (21) version6 (6)}
;
```

```
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,
     longFTN-Supported                [21] NULL                     OPTIONAL }
```

```
SuppressionOfAnnouncement ::= NULL
```

```
InterrogationType ::= ENUMERATED {
     basicCall  (0),
     forwarding  (1)}
```

```
OR-Phase ::= INTEGER (1..127)
```

```
CallReferenceNumber ::= OCTET STRING (SIZE (1..8))
```

```
ForwardingReason ::= ENUMERATED {
     notReachable  (0),
     busy  (1),
     noReply  (2)}
```

```
SupportedCCBS-Phase ::= INTEGER (1..127)
-- exception handling:
-- Only value 1 is used.
-- Values in the ranges 2-127 are reserved for future use.
-- If received values 2-127 shall be mapped on to value 1.
```

```
CallDiversionTreatmentIndicator ::= OCTET STRING (SIZE(1))
--   callDiversionAllowed (xxxx xx01)
--   callDiversionNotAllowed (xxxx xx10)
--   network default is call diversion allowed
```

```
SendRoutingInfoRes ::= [3] SEQUENCE {
     imsi                                [9] IMSI                              OPTIONAL,
     -- IMSI must be present if SendRoutingInfoRes is not segmented.
     -- If the TC-Result-NL segmentation option is taken the IMSI must be
     -- present in one segmented transmission of SendRoutingInfoRes.
     extendedRoutingInfo                ExtendedRoutingInfo                   OPTIONAL,
     cug-CheckInfo                       [3] CUG-CheckInfo                    OPTIONAL,
     cugSubscriptionFlag                 [6] NULL                             OPTIONAL,
     subscriberInfo                      [7] SubscriberInfo                   OPTIONAL,
     ss-List                             [1] SS-List                          OPTIONAL,
     basicService                        [5] Ext-BasicServiceCode             OPTIONAL,
     forwardingInterrogationRequired     [4] NULL                             OPTIONAL,
     vmsc-Address                        [2] ISDN-AddressString               OPTIONAL,
     extensionContainer                  [0] ExtensionContainer               OPTIONAL,
     ... ,
     naea-PreferredCI                    [10] NAEA-PreferredCI                OPTIONAL,
     -- naea-PreferredCI is included at the discretion of the HLR operator.
     ccbs-Indicators                     [11] CCBS-Indicators                 OPTIONAL,
     msisdn                              [12] ISDN-AddressString              OPTIONAL,
     numberPortabilityStatus             [13] NumberPortabilityStatus         OPTIONAL,
     istAlertTimer                       [14] IST-AlertTimerValue             OPTIONAL
     }
```

```
NumberPortabilityStatus ::= ENUMERATED {
     notKnownToBePorted                  (0),
     ownNumberPortedOut                  (1),
     foreignNumberPortedToForeignNetwork (2),
     ...}
     --   exception handling:
     --  reception of other values than the ones listed the receiver shall ignore the
     --  whole NumberPortabilityStatus
```

```
CCBS-Indicators ::= SEQUENCE {
     ccbs-Possible                       [0]  NULL                            OPTIONAL,
     keepCCBS-CallIndicator              [1]  NULL                            OPTIONAL,
     extensionContainer                  [2]  ExtensionContainer              OPTIONAL,
     ...}
```

```
RoutingInfo ::= CHOICE {
     roamingNumber                      ISDN-AddressString,
     forwardingData                     ForwardingData}
```

```
ForwardingData ::= SEQUENCE {
     forwardedToNumber                   [5] ISDN-AddressString               OPTIONAL,
     -- When this datatype is sent from an HLR which supports CAMEL Phase 2
     -- to a GMSC which supports CAMEL Phase 2 the GMSC shall not check the
     -- format of the number
     forwardedToSubaddress               [4] ISDN-SubaddressString            OPTIONAL,
     forwardingOptions                   [6] ForwardingOptions                OPTIONAL,
     extensionContainer                  [7] ExtensionContainer               OPTIONAL,
     ...,
     longForwardedToNumber               [8] FTN-AddressString                OPTIONAL}
```

```
ProvideRoamingNumberArg ::= SEQUENCE {
    imsi                              [0] IMSI,
    msc-Number                        [1] ISDN-AddressString,
    msisdn                            [2] ISDN-AddressString            OPTIONAL,
    lmsi                              [4] LMSI                          OPTIONAL,
    gsm-BearerCapability              [5] ExternalSignalInfo            OPTIONAL,
    networkSignalInfo                 [6] ExternalSignalInfo            OPTIONAL,
    suppressionOfAnnouncement         [7] SuppressionOfAnnouncement     OPTIONAL,
    gmsc-Address                      [8] ISDN-AddressString            OPTIONAL,
    callReferenceNumber               [9] CallReferenceNumber           OPTIONAL,
    or-Interrogation                  [10] NULL                         OPTIONAL,
    extensionContainer                [11] ExtensionContainer           OPTIONAL,
    ... ,
    alertingPattern                   [12] AlertingPattern              OPTIONAL,
    ccbs-Call                         [13] NULL                         OPTIONAL,
    supportedCamelPhasesInGMSC        [15] SupportedCamelPhases         OPTIONAL,
    additionalSignalInfo              [14] Ext-ExternalSignalInfo       OPTIONAL,
    orNotSupportedInGMSC              [16] NULL                         OPTIONAL,
    pre-pagingSupported               [17] NULL                         OPTIONAL,
    longFTN-Supported                 [18] NULL                         OPTIONAL }
```

```
ProvideRoamingNumberRes ::= SEQUENCE {
    roamingNumber                     ISDN-AddressString,
    extensionContainer                ExtensionContainer                OPTIONAL,
    ...}
```

```
ResumeCallHandlingArg ::= SEQUENCE {
    callReferenceNumber               [0] CallReferenceNumber           OPTIONAL,
    basicServiceGroup                 [1] Ext-BasicServiceCode          OPTIONAL,
    forwardingData                    [2] ForwardingData                OPTIONAL,
    imsi                              [3] IMSI                          OPTIONAL,
    cug-CheckInfo                     [4] CUG-CheckInfo                 OPTIONAL,
    o-CSI                             [5] O-CSI                         OPTIONAL,
    extensionContainer                [7] ExtensionContainer            OPTIONAL,
    ccbs-Possible                     [8]  NULL                         OPTIONAL,
    msisdn                            [9]  ISDN-AddressString           OPTIONAL,
    uu-Data                           [10] UU-Data                      OPTIONAL,
    allInformationSent                [11] NULL                         OPTIONAL,
    ...,
    d-csi                             [12] D-CSI                        OPTIONAL}
```

```
UU-Data ::= SEQUENCE {
    uuIndicator                       [0] UUIndicator                   OPTIONAL,
    uui                               [1] UUI                           OPTIONAL,
    uusCFInteraction                  [2] NULL                          OPTIONAL,
    extensionContainer                [3] ExtensionContainer            OPTIONAL,
    ...}
```

```
UUIndicator ::= OCTET STRING (SIZE (1))
    -- Octets are coded according to ETS 300 356
```

```
UUI  ::= OCTET STRING (SIZE (1..131))
    -- Octets are coded according to ETS 300 356
```

```
ResumeCallHandlingRes ::= SEQUENCE {
    extensionContainer                ExtensionContainer                OPTIONAL,
    ...}
```

```
CamelInfo ::= SEQUENCE {
    supportedCamelPhases              SupportedCamelPhases,
    suppress-T-CSI                    NULL                              OPTIONAL,
    extensionContainer                ExtensionContainer                OPTIONAL,
    ...}
```

```
ExtendedRoutingInfo ::= CHOICE {
    routingInfo                       RoutingInfo,
    camelRoutingInfo                  [8] CamelRoutingInfo}
```

```
CamelRoutingInfo ::= SEQUENCE {
    forwardingData                    ForwardingData                    OPTIONAL,
    gmscCamelSubscriptionInfo         [0] GmscCamelSubscriptionInfo,
    extensionContainer                [1] ExtensionContainer            OPTIONAL,
    ...}
```

```
GmscCamelSubscriptionInfo ::= SEQUENCE {
     t-CSI                              [0] T-CSI OPTIONAL,
     o-CSI                              [1] O-CSI OPTIONAL,
     extensionContainer                 [2] ExtensionContainer            OPTIONAL,
     ...,
     o-BcsmCamelTDP-CriteriaList        [3] O-BcsmCamelTDPCriteriaList    OPTIONAL,
     t-BCSM-CAMEL-TDP-CriteriaList      [4]  T-BCSM-CAMEL-TDP-CriteriaList OPTIONAL,
     d-csi                              [5]  D-CSI                        OPTIONAL}
```

```
ProvideSIWFSNumberArg ::= SEQUENCE {
     gsm-BearerCapability               [0] ExternalSignalInfo,
     isdn-BearerCapability              [1] ExternalSignalInfo,
     call-Direction                     [2] CallDirection,
     b-Subscriber-Address               [3] ISDN-AddressString,
     chosenChannel                      [4] ExternalSignalInfo,
     lowerLayerCompatibility            [5] ExternalSignalInfo            OPTIONAL,
     highLayerCompatibility             [6] ExternalSignalInfo            OPTIONAL,
     extensionContainer                 [7] ExtensionContainer            OPTIONAL,
     ...}
```

```
CallDirection ::= OCTET STRING (SIZE (1))
     -- OCTET 1

     -- bit 1 (direction of call)
     -- 0  Mobile Originated Call (MOC)
     -- 1  Mobile Terminated Call (MTC)
```

```
ProvideSIWFSNumberRes ::= SEQUENCE {
     sIWFSNumber                        [0] ISDN-AddressString,
     extensionContainer                 [1] ExtensionContainer            OPTIONAL,
     ...}
```

```
SIWFSSignallingModifyArg ::= SEQUENCE {
     channelType                        [0] ExternalSignalInfo            OPTIONAL,
     chosenChannel                      [1] ExternalSignalInfo            OPTIONAL,
     extensionContainer                 [2] ExtensionContainer            OPTIONAL,
     ...}
```

```
SIWFSSignallingModifyRes ::= SEQUENCE {
     chosenChannel                      [0] ExternalSignalInfo            OPTIONAL,
     extensionContainer                 [1] ExtensionContainer            OPTIONAL,
     ...}
```

```
SetReportingStateArg ::= SEQUENCE {
     imsi                               [0]  IMSI                         OPTIONAL,
     lmsi                               [1]  LMSI                         OPTIONAL,
     ccbs-Monitoring                    [2]  ReportingState               OPTIONAL,
     extensionContainer                 [3]  ExtensionContainer           OPTIONAL,
     ...}
```

```
ReportingState ::= ENUMERATED {
     stopMonitoring                     (0),
     startMonitoring                    (1),
     ...}
     -- exception handling:
     -- reception of values 2-10 shall be mapped to 'stopMonitoring'
     -- reception of values > 10 shall be mapped to 'startMonitoring'
```

```
SetReportingStateRes ::= SEQUENCE{
     ccbs-SubscriberStatus              [0]  CCBS-SubscriberStatus        OPTIONAL,
     extensionContainer                 [1]  ExtensionContainer           OPTIONAL,
     ...}
```

```
CCBS-SubscriberStatus ::= ENUMERATED {
     ccbsNotIdle                        (0),
     ccbsIdle                           (1),
     ccbsNotReachable                   (2),
     ...}
     --  exception handling:
     --  reception of values 3-10 shall be mapped to 'ccbsNotIdle'
     --  reception of values 11-20 shall be mapped to 'ccbsIdle'
     --  reception of values > 20 shall be mapped to 'ccbsNotReachable'
```

```
StatusReportArg ::= SEQUENCE{
     imsi                              [0]  IMSI,
     eventReportData                   [1]  EventReportData              OPTIONAL,
     callReportdata                    [2]  CallReportData               OPTIONAL,
     extensionContainer                [3]  ExtensionContainer           OPTIONAL,
     ...}
```

```
EventReportData ::= SEQUENCE{
     ccbs-SubscriberStatus             [0]  CCBS-SubscriberStatus        OPTIONAL,
     extensionContainer                [1]  ExtensionContainer           OPTIONAL,
     ...}
```

```
CallReportData ::= SEQUENCE{
     monitoringMode                    [0]  MonitoringMode               OPTIONAL,
     callOutcome                       [1]  CallOutcome                  OPTIONAL,
     extensionContainer                [2]  ExtensionContainer           OPTIONAL,
     ...}
```

```
MonitoringMode ::= ENUMERATED {
     a-side                            (0),
     b-side                            (1),
     ...}
     --   exception handling:
     --  reception of values 2-10 shall be mapped 'a-side'
     --  reception of values > 10 shall be mapped to 'b-side'
```

```
CallOutcome ::= ENUMERATED {
     success                           (0),
     failure                           (1),
     busy                              (2),
     ...}
     --   exception handling:
     --  reception of values 3-10 shall be mapped to 'success'
     --  reception of values 11-20 shall be mapped to 'failure'
     --  reception of values > 20 shall be mapped to 'busy'
```

```
StatusReportRes ::= SEQUENCE {
     extensionContainer                [0]  ExtensionContainer           OPTIONAL,
     ...}
```

```
RemoteUserFreeArg ::= SEQUENCE{
     imsi                              [0]  IMSI,
     callInfo                          [1]  ExternalSignalInfo,
     ccbs-Feature                      [2]  CCBS-Feature,
     translatedB-Number                [3]  ISDN-AddressString,
     replaceB-Number                   [4]  NULL                         OPTIONAL,
     alertingPattern                   [5]  AlertingPattern              OPTIONAL,
     extensionContainer                [6]  ExtensionContainer           OPTIONAL,
     ...}
```

```
RemoteUserFreeRes ::= SEQUENCE{
     ruf-Outcome                       [0]  RUF-Outcome,
     extensionContainer                [1]  ExtensionContainer           OPTIONAL,
     ...}
```

```
RUF-Outcome ::= ENUMERATED{
     accepted (0),
     rejected (1),
     noResponseFromFreeMS (2), -- T4 Expiry
     noResponseFromBusyMS (3), -- T10 Expiry
     udubFromFreeMS (4),
     udubFromBusyMS (5),
     ...}
     -- exception handling:
     -- reception of values 6-20 shall be mapped to 'accepted'
     -- reception of values 21-30 shall be mapped to 'rejected'
     -- reception of values 31-40 shall be mapped to 'noResponseFromFreeMS'
     -- reception of values 41-50 shall be mapped to 'noResponseFromBusyMS'
     -- reception of values 51-60 shall be mapped to 'udubFromFreeMS'
     -- reception of values > 60 shall be mapped to 'udubFromBusyMS'
```

```
IST-AlertArg ::= SEQUENCE{
     imsi                              [0]  IMSI,
     extensionContainer                [1]  ExtensionContainer           OPTIONAL,
     ...}
```

```
IST-AlertRes ::= SEQUENCE{
     istAlertTimer                    [0]  IST-AlertTimerValue          OPTIONAL,
     istInformationWithdraw           [1]  NULL                         OPTIONAL,
     callTerminationIndicator         [2]  CallTerminationIndicator     OPTIONAL,
     extensionContainer               [3]  ExtensionContainer           OPTIONAL,
     ...}
```

```
IST-CommandArg ::= SEQUENCE{
     imsi                             [0]  IMSI,
     extensionContainer               [1]  ExtensionContainer           OPTIONAL,
     ...}
```

```
IST-CommandRes ::= SEQUENCE{
     extensionContainer               ExtensionContainer                OPTIONAL,
     ...}
```

```
CallTerminationIndicator ::= ENUMERATED {
     terminateCallActivityReferred       (0),
     terminateAllCallActivities          (1),
     ...}
     -- exception handling:
     -- reception of values 2-10 shall be mapped to ' terminateCallActivityReferred '
     -- reception of values > 10 shall be mapped to ' terminateAllCallActivities '

     -- In MSCs not supporting linkage of all call activities, any value received shall
     -- be interpreted as ' terminateCallActivityReferred '
```

```
END
```

17.7.4 Supplementary service data types

```
MAP-SS-DataTypes {
   ccitt identified-organization (4) etsi (0) mobileDomain (0)
   gsm-Network (1) modules (3) map-SS-DataTypes (14) version6 (6)}

DEFINITIONS

IMPLICIT TAGS

::=

BEGIN

EXPORTS
   RegisterSS-Arg,
   SS-Info,
   SS-Status,
   SS-SubscriptionOption,
   SS-ForBS-Code,
   InterrogateSS-Res,
   USSD-Arg,
   USSD-Res,
   USSD-DataCodingScheme,
   USSD-String,
   Password,
   GuidanceInfo,
   SS-List,
   SS-InfoList,
   OverrideCategory,
   CliRestrictionOption,
   NoReplyConditionTime,
   ForwardingOptions,
   maxNumOfSS,
   SS-Data,
   SS-InvocationNotificationArg,
   SS-InvocationNotificationRes,
   CCBS-Feature,
   RegisterCC-EntryArg,
   RegisterCC-EntryRes,
   EraseCC-EntryArg,
   EraseCC-EntryRes
;

IMPORTS
   AddressString,
   ISDN-AddressString,
   ISDN-SubaddressString,
   FTN-AddressString,
```

```
   IMSI,
   BasicServiceCode,
   AlertingPattern,
   EMLPP-Priority,
   MaxMC-Bearers,
   MC-Bearers,
   ExternalSignalInfo

FROM MAP-CommonDataTypes {
   ccitt identified-organization (4) etsi (0) mobileDomain (0)
   gsm-Network (1) modules (3) map-CommonDataTypes (18) version6 (6)}

   ExtensionContainer
FROM MAP-ExtensionDataTypes {
   ccitt identified-organization (4) etsi (0) mobileDomain (0)
   gsm-Network (1) modules (3) map-ExtensionDataTypes (21) version6 (6)}

   SS-Code
FROM MAP-SS-Code {
   ccitt identified-organization (4) etsi (0) mobileDomain (0)
   gsm-Network (1) modules (3) map-SS-Code (15) version6 (6)}
;
```

```
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,
     nbrUser                          [8] MC-Bearers                   OPTIONAL,
     longFTN-Supported                [9]  NULL                        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,
     ...,
     longForwardedToNumber            [9] FTN-AddressString            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 3GPP TS 23.011

     -- bit 4: "Q bit"

     -- bit 3: "P bit"

     -- bit 2: "R bit"

     -- bit 1: "A bit"
```

```
ForwardingOptions ::= OCTET STRING (SIZE (1))

     -- bit 8: notification to forwarding party
     --   0  no notification
     --   1  notification

     -- bit 7: redirecting presentation
     --   0 no presentation
     --   1  presentation

     -- bit 6: notification to calling party
     --   0  no notification
     --   1  notification

     -- bit 5: 0 (unused)

     -- bits 43: forwarding reason
     --   00  ms not reachable
     --   01  ms busy
     --   10  no reply
     --   11  unconditional when used in a SRI Result,
     --       or call deflection when used in a RCH Argument
     -- bits 21: 00 (unused)
```

```
CallBarringInfo ::= SEQUENCE {
     ss-Code                         SS-Code                         OPTIONAL,
     callBarringFeatureList          CallBarringFeatureList,
     ...}
```

```
CallBarringFeatureList ::= SEQUENCE SIZE (1..maxNumOfBasicServiceGroups) OF
                                     CallBarringFeature
```

```
CallBarringFeature ::= SEQUENCE {
     basicService                    BasicServiceCode                OPTIONAL,
     ss-Status [4] SS-Status         OPTIONAL,
     ...}
```

```
SS-Data ::= SEQUENCE {
     ss-Code                         SS-Code                         OPTIONAL,
     ss-Status                       [4] SS-Status                   OPTIONAL,
     ss-SubscriptionOption           SS-SubscriptionOption           OPTIONAL,
     basicServiceGroupList           BasicServiceGroupList           OPTIONAL,
     ...,
     defaultPriority                 EMLPP-Priority                  OPTIONAL,
     nbrUser                         [5] MC-Bearers                  OPTIONAL
     }
```

```
SS-SubscriptionOption ::= CHOICE {
     cliRestrictionOption            [2] CliRestrictionOption,
     overrideCategory                [1] OverrideCategory}
```

```
CliRestrictionOption ::= ENUMERATED {
     permanent  (0),
     temporaryDefaultRestricted  (1),
     temporaryDefaultAllowed  (2)}
```

```
OverrideCategory ::= ENUMERATED {
     overrideEnabled  (0),
     overrideDisabled  (1)}
```

```
SS-ForBS-Code ::= SEQUENCE {
     ss-Code                         SS-Code,
     basicService                    BasicServiceCode                OPTIONAL,
     ...,
     longFTN-Supported               [4]  NULL                       OPTIONAL }
```

```
GenericServiceInfo ::= SEQUENCE {
     ss-Status SS-Status,
     cliRestrictionOption            CliRestrictionOption            OPTIONAL,
     ...,
     maximumEntitledPriority         [0] EMLPP-Priority              OPTIONAL,
     defaultPriority                 [1] EMLPP-Priority              OPTIONAL,
     ccbs-FeatureList                [2] CCBS-FeatureList            OPTIONAL,
     nbrSB                           [3] MaxMC-Bearers               OPTIONAL,
     nbrUser                         [4] MC-Bearers                  OPTIONAL,
     nbrSN                           [5] MC-Bearers                  OPTIONAL }
```

```
CCBS-FeatureList ::= SEQUENCE SIZE (1..maxNumOfCCBS-Requests) OF
                                        CCBS-Feature
```

```
maxNumOfCCBS-Requests  INTEGER ::= 5
```

```
CCBS-Feature ::= SEQUENCE {
     ccbs-Index                     [0] CCBS-Index                  OPTIONAL,
     b-subscriberNumber             [1] ISDN-AddressString          OPTIONAL,
     b-subscriberSubaddress         [2] ISDN-SubaddressString       OPTIONAL,
     basicServiceGroup              [3] BasicServiceCode            OPTIONAL,
     ...}
```

```
CCBS-Index  ::= INTEGER (1..maxNumOfCCBS-Requests)
```

```
InterrogateSS-Res ::= CHOICE {
     ss-Status                      [0] SS-Status,
     basicServiceGroupList          [2] BasicServiceGroupList,
     forwardingFeatureList          [3] ForwardingFeatureList,
     genericServiceInfo             [4]  GenericServiceInfo }
```

```
USSD-Arg ::= SEQUENCE {
     ussd-DataCodingScheme          USSD-DataCodingScheme,
     ussd-String                    USSD-String,
     ... ,
     alertingPattern                AlertingPattern                 OPTIONAL,
     msisdn                         [0] ISDN-AddressString          OPTIONAL }
```

```
USSD-Res ::= SEQUENCE {
     ussd-DataCodingScheme          USSD-DataCodingScheme,
     ussd-String                    USSD-String,
     ...}
```

```
USSD-DataCodingScheme ::= OCTET STRING (SIZE (1))
     -- The structure of the USSD-DataCodingScheme is defined by
     -- the Cell Broadcast Data Coding Scheme as described in
     -- TS 3GPP TS 23.038[25]
```

```
USSD-String ::= OCTET STRING (SIZE (1..maxUSSD-StringLength))
     -- The structure of the contents of the USSD-String is dependent
     -- on the USSD-DataCodingScheme as described in TS 3GPP TS 23.038[25].
```

```
maxUSSD-StringLength  INTEGER ::= 160
```

```
Password ::= NumericString
     (FROM ("0"|"1"|"2"|"3"|"4"|"5"|"6"|"7"|"8"|"9"))
     (SIZE (4))
```

```
GuidanceInfo ::= ENUMERATED {
     enterPW  (0),
     enterNewPW  (1),
     enterNewPW-Again  (2)}
     -- How this information is really delivered to the subscriber
     -- (display, announcement, ...) is not part of this
     -- specification.
```

```
SS-List ::= SEQUENCE SIZE (1..maxNumOfSS) OF
                                        SS-Code
```

```
maxNumOfSS  INTEGER ::= 30
```

```
SS-InfoList ::= SEQUENCE SIZE (1..maxNumOfSS) OF
                                        SS-Info
```

```
BasicServiceGroupList ::= SEQUENCE SIZE (1..maxNumOfBasicServiceGroups) OF
                                        BasicServiceCode
```

```
maxNumOfBasicServiceGroups  INTEGER ::= 13
```

```
SS-InvocationNotificationArg ::= SEQUENCE {
     imsi                                [0] IMSI,
     msisdn                              [1] ISDN-AddressString,
     ss-Event                            [2] SS-Code,
     -- The following SS-Code values are allowed :
     -- ect                              SS-Code ::= '00110001'B
     -- multiPTY                         SS-Code ::= '01010001'B
     -- cd                               SS-Code ::= '00100100'B
     -- ccbs                             SS-Code ::= '01000100'B
     ss-EventSpecification               [3] SS-EventSpecification         OPTIONAL,
     extensionContainer                  [4] ExtensionContainer            OPTIONAL,
     ...,
     b-subscriberNumber                  [5]  ISDN-AddressString           OPTIONAL,
     ccbs-RequestState                   [6]  CCBS-RequestState            OPTIONAL
     }
```

```
CCBS-RequestState ::= ENUMERATED {
     request   (0),
     recall    (1),
     active    (2),
     completed (3),
     suspended (4),
     frozen    (5),
     deleted   (6)
     }
```

```
SS-InvocationNotificationRes ::= SEQUENCE {
     extensionContainer                  ExtensionContainer                OPTIONAL,
     ...
     }
```

```
SS-EventSpecification ::= SEQUENCE SIZE (1..maxEventSpecification) OF
                                         AddressString
```

```
maxEventSpecification  INTEGER ::= 2
```

```
RegisterCC-EntryArg ::= SEQUENCE {
     ss-Code                             [0]  SS-Code,
     ccbs-Data                           [1]  CCBS-Data                    OPTIONAL,
     ...}
```

```
CCBS-Data ::= SEQUENCE {
     ccbs-Feature                        [0]  CCBS-Feature,
     translatedB-Number                  [1]  ISDN-AddressString,
     serviceIndicator                    [2]  ServiceIndicator             OPTIONAL,
     callInfo                            [3]  ExternalSignalInfo,
     networkSignalInfo                   [4]  ExternalSignalInfo,
     ...}
```

```
ServiceIndicator ::= BIT STRING {
     clir-invoked (0),
     camel-invoked (1)} (SIZE(2..32))
     -- exception handling:
     -- bits 2 to 31 shall be ignored if received and not understood
```

```
RegisterCC-EntryRes ::= SEQUENCE {
     ccbs-Feature                        [0] CCBS-Feature                  OPTIONAL,
     ...}
```

```
EraseCC-EntryArg ::= SEQUENCE {
     ss-Code                             [0]  SS-Code,
     ccbs-Index                          [1]  CCBS-Index                   OPTIONAL,
     ...}
```

```
EraseCC-EntryRes ::= SEQUENCE {
     ss-Code                             [0]  SS-Code,
     ss-Status                           [1] SS-Status                     OPTIONAL,
     ...}
```

```
END
```

17.7.5 Supplementary service codes

```
MAP-SS-Code {
   ccitt identified-organization (4) etsi (0) mobileDomain (0)
   gsm-Network (1) modules (3) map-SS-Code (15) version6 (6)}
```

```
DEFINITIONS

::=

BEGIN

SS-Code ::= OCTET STRING (SIZE (1))
     -- This type is used to represent the code identifying a single
     -- supplementary service, a group of supplementary services, or
     -- all supplementary services. The services and abbreviations
     -- used are defined in 3GPP TS 22.004. The internal structure is
     -- defined as follows:
     --
     -- bits 87654321: group (bits 8765), and specific service
     -- (bits 4321)

allSS                                   SS-Code ::= '00000000'B
     -- reserved for possible future use
     -- all SS

allLineIdentificationSS                 SS-Code ::= '00010000'B
     -- reserved for possible future use
     -- all line identification SS
clip                                    SS-Code ::= '00010001'B
     -- calling line identification presentation
clir                                    SS-Code ::= '00010010'B
     -- calling line identification restriction
colp                                    SS-Code ::= '00010011'B
     -- connected line identification presentation
colr                                    SS-Code ::= '00010100'B
     -- connected line identification restriction
mci                                     SS-Code ::= '00010101'B
     -- reserved for possible future use
     -- malicious call identification

allNameIdentificationSS                 SS-Code ::= '00011000'B
     -- all name identification SS
cnap                                    SS-Code ::= '00011001'B
     -- calling name presentation

     -- SS-Codes '00011010'B to '00011111'B are reserved for future
     -- NameIdentification Supplementary Service use.

allForwardingSS                         SS-Code ::= '00100000'B
     -- all forwarding SS
cfu                                     SS-Code ::= '00100001'B
     -- call forwarding unconditional
allCondForwardingSS                     SS-Code ::= '00101000'B
     -- all conditional forwarding SS
cfb                                     SS-Code ::= '00101001'B
     -- call forwarding on mobile subscriber busy
cfnry                                   SS-Code ::= '00101010'B
     -- call forwarding on no reply
cfnrc                                   SS-Code ::= '00101011'B
     -- call forwarding on mobile subscriber not reachable
cd                                      SS-Code ::= '00100100'B
     -- call deflection

allCallOfferingSS                       SS-Code ::= '00110000'B
     -- reserved for possible future use
     -- all call offering SS includes also all forwarding SS
ect                                     SS-Code ::= '00110001'B
          -- explicit call transfer
mah                                     SS-Code ::= '00110010'B
     -- reserved for possible future use
     -- mobile access hunting
```

```
allCallCompletionSS                     SS-Code ::= '01000000'B
     -- reserved for possible future use
     -- all Call completion SS
cw                                      SS-Code ::= '01000001'B
     -- call waiting
hold                                    SS-Code ::= '01000010'B
     -- call hold
ccbs-A                                  SS-Code ::= '01000011'B
     -- completion of call to busy subscribers, originating side
ccbs-B                                  SS-Code ::= '01000100'B
     -- completion of call to busy subscribers, destination side
     -- this SS-Code is used only in InsertSubscriberData and DeleteSubscriberData
mc                                      SS-Code ::= '01000101'B
     -- multicall
```

```
allMultiPartySS                         SS-Code ::= '01010000'B
     -- reserved for possible future use
     -- all multiparty SS
multiPTY                                SS-Code ::= '01010001'B
     -- multiparty
```

```
allCommunityOfInterest-SS               SS-Code ::= '01100000'B
     -- reserved for possible future use
     -- all community of interest SS
cug                                     SS-Code ::= '01100001'B
     -- closed user group
```

```
allChargingSS                           SS-Code ::= '01110000'B
     -- reserved for possible future use
     -- all charging SS
aoci                                    SS-Code ::= '01110001'B
     -- advice of charge information
aocc                                    SS-Code ::= '01110010'B
     -- advice of charge charging
```

```
allAdditionalInfoTransferSS             SS-Code ::= '10000000'B
     -- reserved for possible future use
     -- all additional information transfer SS
uus1                                    SS-Code ::= '10000001'B
     -- UUS1 user-to-user signalling
uus2                                    SS-Code ::= '10000010'B
     -- UUS2 user-to-user signalling
uus3                                    SS-Code ::= '10000011'B
     -- UUS3 user-to-user signalling
```

```
allBarringSS                            SS-Code ::= '10010000'B
     -- all barring SS
barringOfOutgoingCalls                  SS-Code ::= '10010001'B
     -- barring of outgoing calls
baoc                                    SS-Code ::= '10010010'B
     -- barring of all outgoing calls
boic                                    SS-Code ::= '10010011'B
     -- barring of outgoing international calls
boicExHC                                SS-Code ::= '10010100'B
     -- barring of outgoing international calls except those directed
     -- to the home PLMN
barringOfIncomingCalls                  SS-Code ::= '10011001'B
     -- barring of incoming calls
baic                                    SS-Code ::= '10011010'B
     -- barring of all incoming calls
bicRoam                                 SS-Code ::= '10011011'B
     -- barring of incoming calls when roaming outside home PLMN
     -- Country
```

```
allPLMN-specificSS                      SS-Code ::= '11110000'B
plmn-specificSS-1                       SS-Code ::= '11110001'B
plmn-specificSS-2                       SS-Code ::= '11110010'B
plmn-specificSS-3                       SS-Code ::= '11110011'B
plmn-specificSS-4                       SS-Code ::= '11110100'B
plmn-specificSS-5                       SS-Code ::= '11110101'B
plmn-specificSS-6                       SS-Code ::= '11110110'B
plmn-specificSS-7                       SS-Code ::= '11110111'B
plmn-specificSS-8                       SS-Code ::= '11111000'B
plmn-specificSS-9                       SS-Code ::= '11111001'B
plmn-specificSS-A                       SS-Code ::= '11111010'B
plmn-specificSS-B                       SS-Code ::= '11111011'B
plmn-specificSS-C                       SS-Code ::= '11111100'B
plmn-specificSS-D                       SS-Code ::= '11111101'B
plmn-specificSS-E                       SS-Code ::= '11111110'B
plmn-specificSS-F                       SS-Code ::= '11111111'B
```

```
allCallPrioritySS                       SS-Code ::= '10100000'B
     -- reserved for possible future use
     -- all call priority SS
emlpp                                   SS-Code ::= '10100001'B
     -- enhanced Multilevel Precedence Pre-emption (EMLPP) service
```

```
allLCSPrivacyException                  SS-Code ::= '10110000'B
     -- all LCS Privacy Exception Classes
universal                               SS-Code ::= '10110001'B
     -- allow location by any LCS client
callrelated                             SS-Code ::= '10110010'B
     -- allow location by any value added LCS client to which a call
     -- is established from the target MS
callunrelated                           SS-Code ::= '10110011'B
     -- allow location by designated external value added LCS clients
plmnoperator                            SS-Code ::= '10110100'B
     -- allow location by designated PLMN operator LCS clients
```

```
allMOLR-SS                              SS-Code ::= '11000000'B
     -- all Mobile Originating Location Request Classes
basicSelfLocation                       SS-Code ::= '11000001'B
     -- allow an MS to request its own location
autonomousSelfLocation                  SS-Code ::= '11000010'B
     -- allow an MS to perform self location without interaction
     -- with the PLMN for a predetermined period of time
transferToThirdParty                    SS-Code ::= '11000011'B
     -- allow an MS to request transfer of its location to another LCS client
```

```
END
```

17.7.6 Short message data types

```
MAP-SM-DataTypes {
   ccitt identified-organization (4) etsi (0) mobileDomain (0)
   gsm-Network (1) modules (3) map-SM-DataTypes (16) version6 (6)}

DEFINITIONS

IMPLICIT TAGS

::=

BEGIN

EXPORTS
   RoutingInfoForSM-Arg,
   RoutingInfoForSM-Res,
   MO-ForwardSM-Arg,
   MO-ForwardSM-Res,
   MT-ForwardSM-Arg,
   MT-ForwardSM-Res,
   ReportSM-DeliveryStatusArg,
   ReportSM-DeliveryStatusRes,
   AlertServiceCentreArg,
   InformServiceCentreArg,
   ReadyForSM-Arg,
   ReadyForSM-Res,
   SM-DeliveryOutcome,
   AlertReason
;
```

```
IMPORTS
   AddressString,
   ISDN-AddressString,
   SignalInfo,
   IMSI,
   LMSI
FROM MAP-CommonDataTypes {
   ccitt identified-organization (4) etsi (0) mobileDomain (0)
   gsm-Network (1) modules (3) map-CommonDataTypes (18) version6 (6)}

   AbsentSubscriberDiagnosticSM
FROM MAP-ER-DataTypes {
   ccitt identified-organization (4) etsi (0) mobileDomain (0)
   gsm-Network (1) modules (3) map-ER-DataTypes (17) version6 (6)}

   ExtensionContainer
FROM MAP-ExtensionDataTypes {
   ccitt identified-organization (4) etsi (0) mobileDomain (0)
   gsm-Network (1) modules (3) map-ExtensionDataTypes (21) version6 (6)}
;
```

```
RoutingInfoForSM-Arg ::= SEQUENCE {
     msisdn                              [0] ISDN-AddressString,
     sm-RP-PRI                           [1] BOOLEAN,
     serviceCentreAddress                [2] AddressString,
     extensionContainer                  [6] ExtensionContainer          OPTIONAL,
     ... ,
     gprsSupportIndicator                [7]  NULL                       OPTIONAL,
     -- gprsSupportIndicator is set only if the SMS-GMSC supports
     -- receiving of two numbers from the HLR
     sm-RP-MTI                           [8] SM-RP-MTI                   OPTIONAL,
     sm-RP-SMEA                          [9] SM-RP-SMEA                  OPTIONAL }
```

```
SM-RP-MTI ::= INTEGER (0..10)
     -- 0 SMS Deliver
     -- 1 SMS Status Report
     -- other values are reserved for future use and shall be discarded if
     -- received
```

```
SM-RP-SMEA ::= OCTET STRING (SIZE (1..12))
     -- this parameter contains an address field which is encoded
     -- as defined in 3GPP TS 23.040[26]. An address field contains 3 elements :
     --       address-length
     --       type-of-address
     --       address-value
```

```
RoutingInfoForSM-Res ::= SEQUENCE {
     imsi                                IMSI,
     locationInfoWithLMSI                [0] LocationInfoWithLMSI,
     extensionContainer                  [4] ExtensionContainer          OPTIONAL,
     ...}
```

```
LocationInfoWithLMSI ::= SEQUENCE {
     networkNode-Number                  [1] ISDN-AddressString,
     lmsi                                LMSI                            OPTIONAL,
     extensionContainer                  ExtensionContainer              OPTIONAL,
     ...,
     gprsNodeIndicator                   [5]  NULL                       OPTIONAL,
     -- gprsNodeIndicator is set only if the SGSN number is sent as the
     -- Network Node Number
     additional-Number                   [6] Additional-Number           OPTIONAL
     -- NetworkNode-number can be either msc-number or sgsn-number
     }
```

```
Additional-Number ::= CHOICE {
     msc-Number                          [0] ISDN-AddressString,
     sgsn-Number                         [1] ISDN-AddressString}
     -- additional-number can be either msc-number or sgsn-number
     -- if received networkNode-number is msc-number then the
     -- additional number is sgsn-number
     -- if received networkNode-number is sgsn-number then the
     -- additional number is msc-number
```

```
MO-ForwardSM-Arg ::= SEQUENCE {
     sm-RP-DA                          SM-RP-DA,
     sm-RP-OA                          SM-RP-OA,
     sm-RP-UI                          SignalInfo,
     extensionContainer                ExtensionContainer              OPTIONAL,
     ... ,
     imsi                              IMSI                            OPTIONAL }
```

```
MO-ForwardSM-Res ::= SEQUENCE {
     sm-RP-UI                          SignalInfo                      OPTIONAL,
     extensionContainer                ExtensionContainer              OPTIONAL,
     ...}
```

```
MT-ForwardSM-Arg ::= SEQUENCE {
     sm-RP-DA                          SM-RP-DA,
     sm-RP-OA                          SM-RP-OA,
     sm-RP-UI                          SignalInfo,
     moreMessagesToSend                NULL                            OPTIONAL,
     extensionContainer                ExtensionContainer              OPTIONAL,
     ...}
```

```
MT-ForwardSM-Res ::= SEQUENCE {
     sm-RP-UI                          SignalInfo                      OPTIONAL,
     extensionContainer                ExtensionContainer              OPTIONAL,
     ...}
```

```
SM-RP-DA ::= CHOICE {
     imsi                              [0] IMSI,
     lmsi                              [1] LMSI,
     serviceCentreAddressDA            [4] AddressString,
     noSM-RP-DA                        [5] NULL}
```

```
SM-RP-OA ::= CHOICE {
     msisdn                            [2] ISDN-AddressString,
     serviceCentreAddressOA            [4] AddressString,
     noSM-RP-OA                        [5] NULL}
```

```
ReportSM-DeliveryStatusArg ::= SEQUENCE {
     msisdn                            ISDN-AddressString,
     serviceCentreAddress              AddressString,
     sm-DeliveryOutcome                SM-DeliveryOutcome,
     absentSubscriberDiagnosticSM      [0] AbsentSubscriberDiagnosticSM
                                                                       OPTIONAL,
     extensionContainer                [1] ExtensionContainer          OPTIONAL,
     ...,
     gprsSupportIndicator              [2]  NULL                       OPTIONAL,
     -- gprsSupportIndicator is set only if the SMS-GMSC supports
     -- handling of two delivery outcomes
     deliveryOutcomeIndicator          [3]  NULL                       OPTIONAL,
     -- DeliveryOutcomeIndicator is set when the SM-DeliveryOutcome
     -- is for GPRS
     additionalSM-DeliveryOutcome      [4]  SM-DeliveryOutcome         OPTIONAL,
     -- If received, additionalSM-DeliveryOutcome is for GPRS
     -- If DeliveryOutcomeIndicator is set, then AdditionalSM-DeliveryOutcome shall be absent
     additionalAbsentSubscriberDiagnosticSM   [5]  AbsentSubscriberDiagnosticSM OPTIONAL
     -- If received additionalAbsentSubscriberDiagnosticSM is for GPRS
     -- If DeliveryOutcomeIndicator is set, then AdditionalAbsentSubscriberDiagnosticSM
     -- shall be absent
     }
```

```
SM-DeliveryOutcome ::= ENUMERATED {
     memoryCapacityExceeded  (0),
     absentSubscriber  (1),
     successfulTransfer  (2)}
```

```
ReportSM-DeliveryStatusRes ::= SEQUENCE {
     storedMSISDN                      ISDN-AddressString              OPTIONAL,
     extensionContainer                ExtensionContainer              OPTIONAL,
     ...}
```

```
AlertServiceCentreArg ::= SEQUENCE {
     msisdn                            ISDN-AddressString,
     serviceCentreAddress              AddressString,
     ...}
```

```
InformServiceCentreArg ::= SEQUENCE {
     storedMSISDN                        ISDN-AddressString              OPTIONAL,
     mw-Status MW-Status                 OPTIONAL,
     extensionContainer                  ExtensionContainer              OPTIONAL,
     ...}
```

```
MW-Status ::= BIT STRING {
     sc-AddressNotIncluded  (0),
     mnrf-Set  (1),
     mcef-Set  (2) ,
     mnrg-Set    (3)} (SIZE (6..16))
     -- exception handling:
     -- bits 4 to 15 shall be ignored if received and not understood
```

```
ReadyForSM-Arg ::= SEQUENCE {
     imsi                                [0] IMSI,
     alertReason                         AlertReason,
     alertReasonIndicator                NULL                            OPTIONAL,
     -- alertReasonIndicator is set only when the alertReason
     -- sent to HLR is for GPRS
     extensionContainer                  ExtensionContainer              OPTIONAL,
     ...}
```

```
ReadyForSM-Res ::= SEQUENCE {
     extensionContainer                  ExtensionContainer              OPTIONAL,
     ...}
```

```
AlertReason ::= ENUMERATED {
     ms-Present  (0),
     memoryAvailable  (1)}
```

```
END
```

17.7.7 Error data types

```
MAP-ER-DataTypes {
   ccitt identified-organization (4) etsi (0) mobileDomain (0)
   gsm-Network (1) modules (3) map-ER-DataTypes (17) version6 (6)}

DEFINITIONS

IMPLICIT TAGS

::=

BEGIN

EXPORTS
   RoamingNotAllowedParam,
   CallBarredParam,
   CUG-RejectParam,
   SS-IncompatibilityCause,
   PW-RegistrationFailureCause,
   SM-DeliveryFailureCause,
   SystemFailureParam,
   DataMissingParam,
   UnexpectedDataParam,
   FacilityNotSupParam,
   OR-NotAllowedParam,
   UnknownSubscriberParam,
   NumberChangedParam,
   UnidentifiedSubParam,
   IllegalSubscriberParam,
   IllegalEquipmentParam,
   BearerServNotProvParam,
   TeleservNotProvParam,
   TracingBufferFullParam,
   NoRoamingNbParam,
   AbsentSubscriberParam,
   BusySubscriberParam,
   NoSubscriberReplyParam,
   ForwardingViolationParam,
   ForwardingFailedParam,
   ATI-NotAllowedParam,
   SubBusyForMT-SMS-Param,
   MessageWaitListFullParam,
   AbsentSubscriberSM-Param,
```

```
   AbsentSubscriberDiagnosticSM,
   ResourceLimitationParam,
   NoGroupCallNbParam,
   IncompatibleTerminalParam,
   ShortTermDenialParam,
   LongTermDenialParam,
   UnauthorizedRequestingNetwork-Param,
   UnauthorizedLCSClient-Param,
   PositionMethodFailure-Param,
   UnknownOrUnreachableLCSClient-Param,
   MM-EventNotSupported-Param,
   ATSI-NotAllowedParam,
   ATM-NotAllowedParam,
   IllegalSS-OperationParam,
   SS-NotAvailableParam,
   SS-SubscriptionViolationParam,
   InformationNotAvailableParam,
   TargetCellOutsideGCA-Param




;

IMPORTS
   SS-Status
FROM MAP-SS-DataTypes {
   ccitt identified-organization (4) etsi (0) mobileDomain (0)
   gsm-Network (1) modules (3) map-SS-DataTypes (14) version6 (6)}

   SignalInfo,
   BasicServiceCode,
   NetworkResource
FROM MAP-CommonDataTypes {
   ccitt identified-organization (4) etsi (0) mobileDomain (0)
   gsm-Network (1) modules (3) map-CommonDataTypes (18) version6 (6)}

   SS-Code
FROM MAP-SS-Code {
   ccitt identified-organization (4) etsi (0) mobileDomain (0)
   gsm-Network (1) modules (3) map-SS-Code (15) version6 (6)}

   ExtensionContainer
FROM MAP-ExtensionDataTypes {
   ccitt identified-organization (4) etsi (0) mobileDomain (0)
   gsm-Network (1) modules (3) map-ExtensionDataTypes (21) version6 (6)}
;
```

```
RoamingNotAllowedParam ::= SEQUENCE {
     roamingNotAllowedCause              RoamingNotAllowedCause,
     extensionContainer                  ExtensionContainer                OPTIONAL,
     ...}
```

```
RoamingNotAllowedCause ::= ENUMERATED {
     plmnRoamingNotAllowed  (0),
     operatorDeterminedBarring  (3)}
```

```
CallBarredParam ::= CHOICE {
     callBarringCause                    CallBarringCause,
     -- call BarringCause must not be used in version 3
     extensibleCallBarredParam           ExtensibleCallBarredParam
     -- extensibleCallBarredParam must not be used in version <3
     }
```

```
CallBarringCause ::= ENUMERATED {
     barringServiceActive  (0),
     operatorBarring  (1)}
```

```
ExtensibleCallBarredParam ::= SEQUENCE {
     callBarringCause                    CallBarringCause                  OPTIONAL,
     extensionContainer                  ExtensionContainer                OPTIONAL,
     ... ,
     unauthorisedMessageOriginator       [1] NULL                          OPTIONAL }
```

```
CUG-RejectParam ::= SEQUENCE {
     cug-RejectCause                    CUG-RejectCause                 OPTIONAL,
     extensionContainer                 ExtensionContainer              OPTIONAL,
     ...}
```

```
CUG-RejectCause ::= ENUMERATED {
     incomingCallsBarredWithinCUG  (0),
     subscriberNotMemberOfCUG  (1),
     requestedBasicServiceViolatesCUG-Constraints  (5),
     calledPartySS-InteractionViolation  (7)}
```

```
SS-IncompatibilityCause ::= SEQUENCE {
     ss-Code                            [1] SS-Code                     OPTIONAL,
     basicService                       BasicServiceCode                OPTIONAL,
     ss-Status                          [4] SS-Status                   OPTIONAL,
     ...}
```

```
PW-RegistrationFailureCause ::= ENUMERATED {
     undetermined  (0),
     invalidFormat  (1),
     newPasswordsMismatch  (2)}
```

```
SM-EnumeratedDeliveryFailureCause ::= ENUMERATED {
     memoryCapacityExceeded  (0),
     equipmentProtocolError  (1),
     equipmentNotSM-Equipped  (2),
     unknownServiceCentre  (3),
     sc-Congestion  (4),
     invalidSME-Address  (5),
     subscriberNotSC-Subscriber  (6)}
```

```
SM-DeliveryFailureCause ::= SEQUENCE {
     sm-EnumeratedDeliveryFailureCause  SM-EnumeratedDeliveryFailureCause,
     diagnosticInfo                     SignalInfo                      OPTIONAL,
     extensionContainer                 ExtensionContainer              OPTIONAL,
     ...}
```

```
AbsentSubscriberSM-Param ::= SEQUENCE {
     absentSubscriberDiagnosticSM       AbsentSubscriberDiagnosticSM    OPTIONAL,
     -- AbsentSubscriberDiagnosticSM can be either for non-GPRS
     -- or for GPRS
     extensionContainer                 ExtensionContainer              OPTIONAL,
     ...,
     additionalAbsentSubscriberDiagnosticSM    [0]  AbsentSubscriberDiagnosticSM OPTIONAL }
     -- if received, additionalAbsentSubscriberDiagnosticSM
     -- is for GPRS and absentSubscriberDiagnosticSM is
     -- for non-GPRS
```

```
AbsentSubscriberDiagnosticSM ::= INTEGER (0..255)
     -- AbsentSubscriberDiagnosticSM values are defined in (3GPP TS 23.040[26])
```

```
SystemFailureParam ::= CHOICE {
     networkResource                    NetworkResource,
     -- networkResource must not be used in version 3
     extensibleSystemFailureParam       ExtensibleSystemFailureParam
     -- extensibleSystemFailureParam must not be used in version <3
     }
```

```
ExtensibleSystemFailureParam ::= SEQUENCE {
     networkResource                    NetworkResource                 OPTIONAL,
     extensionContainer                 ExtensionContainer              OPTIONAL,
     ...}
```

```
DataMissingParam ::= SEQUENCE {
     extensionContainer                 ExtensionContainer              OPTIONAL,
     ...}
```

```
UnexpectedDataParam ::= SEQUENCE {
     extensionContainer                 ExtensionContainer              OPTIONAL,
     ...}
```

```
FacilityNotSupParam ::= SEQUENCE {
     extensionContainer                 ExtensionContainer              OPTIONAL,
     ...}
```

```
OR-NotAllowedParam ::= SEQUENCE {
     extensionContainer                  ExtensionContainer                OPTIONAL,
     ...}
```

```
UnknownSubscriberParam ::= SEQUENCE {
     extensionContainer                  ExtensionContainer                OPTIONAL,
     ...,
     unknownSubscriberDiagnostic         UnknownSubscriberDiagnostic       OPTIONAL}
```

```
UnknownSubscriberDiagnostic ::= ENUMERATED {
     imsiUnknown  (0),
     gprsSubscriptionUnknown  (1),
     ...,
     npdbMismatch  (2)}
     -- if unknown values are received in
     -- unknownSubscriberDiagnostic they shall be discarded
```

```
NumberChangedParam ::= SEQUENCE {
     extensionContainer                  ExtensionContainer                OPTIONAL,
     ...}
```

```
UnidentifiedSubParam ::= SEQUENCE {
     extensionContainer                  ExtensionContainer                OPTIONAL,
     ...}
```

```
IllegalSubscriberParam ::= SEQUENCE {
     extensionContainer                  ExtensionContainer                OPTIONAL,
     ...}
```

```
IllegalEquipmentParam ::= SEQUENCE {
     extensionContainer                  ExtensionContainer                OPTIONAL,
     ...}
```

```
BearerServNotProvParam ::= SEQUENCE {
     extensionContainer                  ExtensionContainer                OPTIONAL,
     ...}
```

```
TeleservNotProvParam ::= SEQUENCE {
     extensionContainer                  ExtensionContainer                OPTIONAL,
     ...}
```

```
TracingBufferFullParam ::= SEQUENCE {
     extensionContainer                  ExtensionContainer                OPTIONAL,
     ...}
```

```
NoRoamingNbParam ::= SEQUENCE {
     extensionContainer                  ExtensionContainer                OPTIONAL,
     ...}
```

```
AbsentSubscriberParam ::= SEQUENCE {
     extensionContainer                  ExtensionContainer                OPTIONAL,
     ...,
     absentSubscriberReason              [0] AbsentSubscriberReason        OPTIONAL}
```

```
AbsentSubscriberReason ::= ENUMERATED {
     imsiDetach (0),
     restrictedArea (1),
     noPageResponse (2),
     ... ,
     purgedMS (3)}
-- exception handling: at reception of other values than the ones listed the
-- AbsentSubscriberReason shall be ignored.
-- The AbsentSubscriberReason: purgedMS is defined for the Super-Charger feature
-- (see TS 23.116). If this value is received in a Provide Roaming Number response
-- it shall be mapped to the AbsentSubscriberReason: imsiDetach in the Send Routeing
-- Information response
```

```
BusySubscriberParam ::= SEQUENCE {
     extensionContainer                  ExtensionContainer                OPTIONAL,
     ...,
     ccbs-Possible                       [0] NULL                          OPTIONAL,
     ccbs-Busy                           [1] NULL                          OPTIONAL}
```

```
NoSubscriberReplyParam ::= SEQUENCE {
     extensionContainer                  ExtensionContainer                  OPTIONAL,
     ...}
```

```
ForwardingViolationParam ::= SEQUENCE {
     extensionContainer                  ExtensionContainer                  OPTIONAL,
     ...}
```

```
ForwardingFailedParam ::= SEQUENCE {
     extensionContainer                  ExtensionContainer                  OPTIONAL,
     ...}
```

```
ATI-NotAllowedParam ::= SEQUENCE {
     extensionContainer                  ExtensionContainer                  OPTIONAL,
     ...}
```

```
ATSI-NotAllowedParam ::= SEQUENCE {
     extensionContainer                  ExtensionContainer                  OPTIONAL,
     ...}
```

```
ATM-NotAllowedParam ::= SEQUENCE {
     extensionContainer                  ExtensionContainer                  OPTIONAL,
     ...}
```

```
IllegalSS-OperationParam ::= SEQUENCE {
     extensionContainer                  ExtensionContainer                  OPTIONAL,
     ...}
```

```
SS-NotAvailableParam ::= SEQUENCE {
     extensionContainer                  ExtensionContainer                  OPTIONAL,
     ...}
```

```
SS-SubscriptionViolationParam ::= SEQUENCE {
     extensionContainer                  ExtensionContainer                  OPTIONAL,
     ...}
```

```
InformationNotAvailableParam ::= SEQUENCE {
     extensionContainer                  ExtensionContainer                  OPTIONAL,
     ...}
```

```
SubBusyForMT-SMS-Param ::= SEQUENCE {
     extensionContainer                  ExtensionContainer                  OPTIONAL,
     ... ,
     gprsConnectionSuspended             NULL                                OPTIONAL }
     -- If GprsConnectionSuspended is not understood it shall
     -- be discarded
```

```
MessageWaitListFullParam ::= SEQUENCE {
     extensionContainer                  ExtensionContainer                  OPTIONAL,
     ...}
```

```
ResourceLimitationParam ::= SEQUENCE {
     extensionContainer                  ExtensionContainer                  OPTIONAL,
     ...}
```

```
NoGroupCallNbParam ::= SEQUENCE {
     extensionContainer                  ExtensionContainer                  OPTIONAL,
     ...}
```

```
IncompatibleTerminalParam ::= SEQUENCE {
     extensionContainer                  ExtensionContainer                  OPTIONAL,
     ...}
```

```
ShortTermDenialParam ::= SEQUENCE {
     ...}
```

```
LongTermDenialParam ::= SEQUENCE {
     ...}
```

```
UnauthorizedRequestingNetwork-Param ::= SEQUENCE {
     extensionContainer                  ExtensionContainer                  OPTIONAL,
     ...}
```

```
UnauthorizedLCSClient-Param ::= SEQUENCE {
     unauthorizedLCSClient-Diagnostic     [0] UnauthorizedLCSClient-Diagnostic   OPTIONAL,
     extensionContainer                   [1] ExtensionContainer                 OPTIONAL,
     ... }
```

```
UnauthorizedLCSClient-Diagnostic ::= ENUMERATED {
     noAdditionalInformation (0),
     clientNotInMSPrivacyExceptionList (1),
     callToClientNotSetup (2),
     privacyOverrideNotApplicable (3),
     disallowedByLocalRegulatoryRequirements (4),
     ... }
--   exception handling:
--   any unrecognized value shall be ignored
```

```
PositionMethodFailure-Param ::= SEQUENCE {
     positionMethodFailure-Diagnostic     [0] PositionMethodFailure-Diagnostic   OPTIONAL,
     extensionContainer                   [1] ExtensionContainer                 OPTIONAL,
     ... }
```

```
PositionMethodFailure-Diagnostic ::= ENUMERATED {
     congestion  (0),
     insufficientResources  (1),
     insufficientMeasurementData  (2),
     inconsistentMeasurementData  (3),
     locationProcedureNotCompleted  (4),
     locationProcedureNotSupportedByTargetMS  (5),
     qoSNotAttainable  (6),
     positionMethodNotAvailableInNetwork  (7),
     positionMethodNotAvailableInLocationArea  (8),
     ... }
--   exception handling:
--   any unrecognized value shall be ignored
```

```
UnknownOrUnreachableLCSClient-Param ::= SEQUENCE {
     extensionContainer                   ExtensionContainer                 OPTIONAL,
     ...}
```

```
MM-EventNotSupported-Param ::= SEQUENCE {
     extensionContainer                   ExtensionContainer                 OPTIONAL,
     ...}
```

```
TargetCellOutsideGCA-Param ::= SEQUENCE {
     extensionContainer                   ExtensionContainer                 OPTIONAL,
     ...}
```

```
END
```

17.7.8 Common data types

```
MAP-CommonDataTypes {
   ccitt identified-organization (4) etsi (0) mobileDomain (0)
   gsm-Network (1) modules (3) map-CommonDataTypes (18) version6 (6)}

DEFINITIONS

IMPLICIT TAGS

::=

BEGIN

EXPORTS

   -- general data types and values
   AddressString,
   ISDN-AddressString,
   maxISDN-AddressLength,
   FTN-AddressString,
   ISDN-SubaddressString,
   ExternalSignalInfo,
   Ext-ExternalSignalInfo,
   AccessNetworkSignalInfo,
   SignalInfo,
   maxSignalInfoLength,
```

```
   AlertingPattern,
   TBCD-STRING,

   -- data types for numbering and identification
   IMSI,
   TMSI,
   Identity,
   SubscriberId,
   IMEI,
   HLR-List,
   LMSI,
   GlobalCellId,
   NetworkResource,
   NAEA-PreferredCI,
   NAEA-CIC,
   ASCI-CallReference,
   SubscriberIdentity,

   -- data types for CAMEL
   CellGlobalIdOrServiceAreaIdOrLAI,

   -- data types for subscriber management
   BasicServiceCode,
   Ext-BasicServiceCode,
   EMLPP-Info,
   EMLPP-Priority,
   MC-SS-Info,
   MaxMC-Bearers,
   MC-Bearers,
   Ext-SS-Status,

   -- data types for geographic location
   AgeOfLocationInformation,
   LCSClientExternalID,
   LCSClientInternalID
;

IMPORTS
   TeleserviceCode,
   Ext-TeleserviceCode
FROM MAP-TS-Code {
   ccitt identified-organization (4) etsi (0) mobileDomain (0)
   gsm-Network (1) modules (3) map-TS-Code (19) version6 (6)}

   BearerServiceCode,
   Ext-BearerServiceCode
FROM MAP-BS-Code {
   ccitt identified-organization (4) etsi (0) mobileDomain (0)
   gsm-Network (1) modules (3) map-BS-Code (20) version6 (6)}

   SS-Code
FROM MAP-SS-Code {
   ccitt identified-organization (4) etsi (0) mobileDomain (0)
   gsm-Network (1) modules (3) map-SS-Code (15) version6 (6)}

   ExtensionContainer
FROM MAP-ExtensionDataTypes {
   ccitt identified-organization (4) etsi (0) mobileDomain (0)
   gsm-Network (1) modules (3) map-ExtensionDataTypes (21) version6 (6)}
;


-- general data types

TBCD-STRING ::= OCTET STRING
     -- This type (Telephony Binary Coded Decimal String) is used to
     -- represent several digits from 0 through 9, *, #, a, b, c, two
     -- digits per octet, each digit encoded 0000 to 1001 (0 to 9),
     -- 1010 (*), 1011 (#), 1100 (a), 1101 (b) or 1110 (c); 1111 used
     -- as filler when there is an odd number of digits.

     -- bits 8765 of octet n encoding digit 2n
     -- bits 4321 of octet n encoding digit 2(n-1) +1
```

```
AddressString ::= OCTET STRING (SIZE (1..maxAddressLength))
     -- This type is used to represent a number for addressing
     -- purposes. It is composed of
     --   a)   one octet for nature of address, and numbering plan
     --        indicator.
     --   b)   digits of an address encoded as TBCD-String.

     -- a)     The first octet includes a one bit extension indicator, a
     --        3 bits nature of address indicator and a 4 bits numbering
     --        plan indicator, encoded as follows:

     -- bit 8: 1  (no extension)

     -- bits 765: nature of address indicator
     --   000  unknown
     --   001  international number
     --   010  national significant number
     --   011  network specific number
     --   100  subscriber number
     --   101  reserved
     --   110  abbreviated number
     --   111  reserved for extension

     -- bits 4321: numbering plan indicator
     --   0000  unknown
     --   0001  ISDN/Telephony Numbering Plan (Rec CCITT E.164)
     --   0010  spare
     --   0011  data numbering plan (CCITT Rec X.121)
     --   0100  telex numbering plan (CCITT Rec F.69)
     --   0101  spare
     --   0110  land mobile numbering plan (CCITT Rec E.212)
     --   0111  spare
     --   1000  national numbering plan
     --   1001  private numbering plan
     --   1111  reserved for extension

     --   all other values are reserved.

     -- b)     The following octets representing digits of an address
     --        encoded as a TBCD-STRING.
```

```
maxAddressLength  INTEGER ::= 20
```

```
ISDN-AddressString ::=
               AddressString (SIZE (1..maxISDN-AddressLength))
     -- This type is used to represent ISDN numbers.
```

```
maxISDN-AddressLength  INTEGER ::= 9
```

```
FTN-AddressString ::=
               AddressString (SIZE (1..maxFTN-AddressLength))
     -- This type is used to represent forwarded-to numbers.
```

```
maxFTN-AddressLength  INTEGER ::= 15
```

```
ISDN-SubaddressString ::=
            OCTET STRING (SIZE (1..maxISDN-SubaddressLength))
    -- This type is used to represent ISDN subaddresses.
    -- It is composed of
    --   a)   one octet for type of subaddress and odd/even indicator.
    --   b)   20 octets for subaddress information.

    --   a)   The first octet includes a one bit extension indicator, a
    --        3 bits type of subaddress and a one bit odd/even indicator,
    --        encoded as follows:

    --   bit 8: 1  (no extension)

    --   bits 765: type of subaddress
    --        000  NSAP (X.213/ISO 8348 AD2)
    --        010  User Specified
    --        All other values are reserved

    --   bit 4: odd/even indicator
    --        0  even number of address signals
    --        1  odd number of address signals
    --        The odd/even indicator is used when the type of subaddress
    --        is "user specified" and the coding is BCD.

    --   bits 321: 000 (unused)

    --   b) Subaddress information.
    --   The NSAP X.213/ISO8348AD2 address shall be formatted as specified
    --   by octet 4 which contains the Authority and Format Identifier
    --   (AFI). The encoding is made according to the "preferred binary
    --   encoding" as defined in X.213/ISO834AD2. For the definition
    --   of this type of subaddress, see CCITT Rec I.334.

    --   For User-specific subaddress, this field is encoded according
    --   to the user specification, subject to a maximum length of 20
    --   octets. When interworking with X.25 networks BCD coding should
    --   be applied.
```

```
maxISDN-SubaddressLength  INTEGER ::= 21
```

```
ExternalSignalInfo ::= SEQUENCE {
    protocolId                      ProtocolId,
    signalInfo                      SignalInfo,
    -- Information about the internal structure is given in
    -- clause 7.6.9.
    extensionContainer              ExtensionContainer              OPTIONAL,
    -- extensionContainer must not be used in version 2
    ...}
```

```
SignalInfo ::= OCTET STRING (SIZE (1..maxSignalInfoLength))
```

```
maxSignalInfoLength  INTEGER ::= 200
    -- This NamedValue represents the theoretical maximum number of octets which is
    -- available to carry a single instance of the SignalInfo data type,
    -- without requiring segmentation to cope with the network layer service.
    -- However, the actual maximum size available for an instance of the data
    -- type may be lower, especially when other information elements
    -- have to be included in the same component.
```

```
ProtocolId ::= ENUMERATED {
    gsm-0408  (1),
    gsm-0806  (2),
    gsm-BSSMAP  (3),
    -- Value 3 is reserved and must not be used
    ets-300102-1  (4)}
```

```
Ext-ExternalSignalInfo ::= SEQUENCE {
    ext-ProtocolId                  Ext-ProtocolId,
    signalInfo                      SignalInfo,
    -- Information about the internal structure is given in
    -- clause 7.6.9.10
    extensionContainer              ExtensionContainer              OPTIONAL,
    ...}
```

```
Ext-ProtocolId ::= ENUMERATED {
     ets-300356  (1),
     ...
      }
-- exception handling:
-- For Ext-ExternalSignalInfo sequences containing this parameter with any
-- other value than the ones listed the receiver shall ignore the whole
-- Ext-ExternalSignalInfo sequence.
```

```
AccessNetworkSignalInfo ::= SEQUENCE {
     accessNetworkProtocolId             AccessNetworkProtocolId,
     signalInfo                          LongSignalInfo,
     -- Information about the internal structure is given in
     extensionContainer                  ExtensionContainer              OPTIONAL,
     ...}
```

```
LongSignalInfo ::= OCTET STRING (SIZE (1..maxLongSignalInfoLength))
```

```
maxLongSignalInfoLength  INTEGER ::= 2560
     -- This Named Value represents the maximum number of octets which is available
     -- to carry a single instance of the LongSignalInfo data type using
     -- White Book SCCP with the maximum number of segments.
     -- It takes account of the octets used by the lower layers of the protocol, and
     -- other information elements which may be included in the same component.
```

```
AccessNetworkProtocolId ::= ENUMERATED {
     gsm-0806   (1),
     ts3G-25413 (2),
     ...}
     -- exception handling:
     -- For AccessNetworkSignalInfo sequences containing this parameter with any
     -- other value than the ones listed the receiver shall ignore the whole
     -- AccessNetworkSignalInfo sequence.
```

```
AlertingPattern ::= OCTET STRING (SIZE (1) )
     -- This type is used to represent Alerting Pattern

     --   bits 8765 : 0000 (unused)

     --   bits 43 : type of Pattern
     --        00 level
     --        01 category
     --        10 category
     --        all other values are reserved.

     --   bits 21 : type of alerting

alertingLevel-0   AlertingPattern ::= '00000000'B
alertingLevel-1   AlertingPattern ::= '00000001'B
alertingLevel-2   AlertingPattern ::= '00000010'B
     -- all other values of Alerting level are reserved
     -- Alerting Levels are defined in GSM 02.07

alertingCategory-1   AlertingPattern ::= '00000100'B
alertingCategory-2   AlertingPattern ::= '00000101'B
alertingCategory-3   AlertingPattern ::= '00000110'B
alertingCategory-4   AlertingPattern ::= '00000111'B
alertingCategory-5   AlertingPattern ::= '00001000'B
     -- all other values of Alerting Category are reserved
     -- Alerting categories are defined in GSM 02.07
```

-- data types for numbering and identification

```
IMSI ::= TBCD-STRING (SIZE (3..8))
     -- digits of MCC, MNC, MSIN are concatenated in this order.
```

```
Identity ::= CHOICE {
     imsi                                IMSI,
     imsi-WithLMSI                       IMSI-WithLMSI}
```

```
IMSI-WithLMSI ::= SEQUENCE {
    imsi                                IMSI,
    lmsi                                LMSI,
    -- a special value 00000000 indicates that the LMSI is not in use
    ...}
```

```
ASCI-CallReference ::= TBCD-STRING (SIZE (1..8))
    -- digits of VGCS/VBC-area,Group-ID are concatenated in this order.
```

```
TMSI ::= OCTET STRING (SIZE (1..4))
```

```
SubscriberId ::= CHOICE {
    imsi                                [0] IMSI,
    tmsi                                [1] TMSI}
```

```
IMEI ::= TBCD-STRING (SIZE (8))
    --   Refers to International Mobile Station Equipment Identity
    --   and Software Version Number (SVN) defined in 3GPP TS 23.003.
    --   If the SVN is not present the last octet shall contain the
    --   digit 0 and a filler.
    --   If present the SVN shall be included in the last octet.
```

```
HLR-Id ::= IMSI
    -- leading digits of IMSI, i.e. (MCC, MNC, leading digits of
    -- MSIN) forming HLR Id defined in 3GPP TS 23.003.
```

```
HLR-List ::= SEQUENCE SIZE (1..maxNumOfHLR-Id) OF
                                        HLR-Id
```

```
maxNumOfHLR-Id  INTEGER ::= 50
```

```
LMSI ::= OCTET STRING (SIZE (4))
```

```
GlobalCellId ::= OCTET STRING (SIZE (5..7))
    -- Refers to Cell Global Identification defined in 3GPP TS 23.003.
    -- The internal structure is defined as follows:
    -- octet 1 bits 4321                Mobile Country Code 1st digit
    --         bits 8765                Mobile Country Code 2nd digit
    -- octet 2 bits 4321                Mobile Country Code 3rd digit
    --         bits 8765                Mobile Network Code 3rd digit
    --                                  or filler (1111) for 2 digit MNCs
    -- octet 3 bits 4321                Mobile Network Code 1st digit
    --         bits 8765                Mobile Network Code 2nd digit
    -- octets 4 and 5                   Location Area Code according to 3GPP TS 24.008
    -- octets 6 and 7                   Cell Identity (CI) according to 3GPP TS 24.008
```

```
NetworkResource ::= ENUMERATED {
    plmn  (0),
    hlr  (1),
    vlr  (2),
    pvlr  (3),
    controllingMSC  (4),
    vmsc  (5),
    eir  (6),
    rss  (7)}
```

```
NAEA-PreferredCI ::= SEQUENCE {
    naea-PreferredCIC                   [0] NAEA-CIC,
    extensionContainer                  [1] ExtensionContainer          OPTIONAL,
    ...}
```

```
NAEA-CIC ::= OCTET STRING (SIZE (3))
    -- The internal structure is defined by the Carrier Identification
    -- parameter in ANSI T1.113.3. Carrier codes between "000" and "999" may
    -- be encoded as 3 digits using "000" to "999" or as 4 digits using
    -- "0000" to "0999". Carrier codes between "1000" and "9999" are encoded
    -- using 4 digits.
```

```
SubscriberIdentity ::= CHOICE {
    imsi                                [0] IMSI,
    msisdn                              [1] ISDN-AddressString
    }
```

```
LCSClientExternalID ::= SEQUENCE {
     externalAddress                    [0] AddressString                OPTIONAL,
     extensionContainer                 [1] ExtensionContainer           OPTIONAL,
     ... }
```

```
LCSClientInternalID ::= ENUMERATED {
     broadcastService                   (0),
     o-andM-HPLMN                       (1),
     o-andM-VPLMN                       (2),
     anonymousLocation                  (3),
     targetMSsubscribedService          (4),
     ... }
-- for a CAMEL phase 3 PLMN operator client, the value targetMSsubscribedService shall be used
```

-- data types for CAMEL

```
CellGlobalIdOrServiceAreaIdOrLAI ::= CHOICE {
     cellGlobalIdOrServiceAreaIdFixedLength   [0] CellGlobalIdOrServiceAreaIdFixedLength,
     laiFixedLength                     [1] LAIFixedLength}
```

```
CellGlobalIdOrServiceAreaIdFixedLength ::= OCTET STRING (SIZE (7))
     -- Refers to Cell Global Identification or Service Are Identification
     -- defined in 3GPP TS 23.003.
     -- The internal structure is defined as follows:
     -- octet 1 bits 4321               Mobile Country Code 1st digit
     --         bits 8765               Mobile Country Code 2nd digit
     -- octet 2 bits 4321               Mobile Country Code 3rd digit
     --         bits 8765               Mobile Network Code 3rd digit
     --                                 or filler (1111) for 2 digit MNCs
     -- octet 3 bits 4321               Mobile Network Code 1st digit
     --         bits 8765               Mobile Network Code 2nd digit
     -- octets 4 and 5                  Location Area Code according to 3GPP TS 24.008
     -- octets 6 and 7                  Cell Identity (CI) value or
     --                                 Service Area Code (SAC) value
     --                                 according to 3GPP TS 23.003
```

```
LAIFixedLength ::= OCTET STRING (SIZE (5))
     -- Refers to Location Area Identification defined in 3GPP TS 23.003.
     -- The internal structure is defined as follows:
     -- octet 1 bits 4321               Mobile Country Code 1st digit
     --         bits 8765               Mobile Country Code 2nd digit
     -- octet 2 bits 4321               Mobile Country Code 3rd digit
     --         bits 8765               Mobile Network Code 3rd digit
     --                                 or filler (1111) for 2 digit MNCs
     -- octet 3 bits 4321               Mobile Network Code 1st digit
     --         bits 8765               Mobile Network Code 2nd digit
     -- octets 4 and 5                  Location Area Code according to 3GPP TS 24.008
```

-- data types for subscriber management

```
BasicServiceCode ::= CHOICE {
     bearerService                      [2] BearerServiceCode,
     teleservice                        [3] TeleserviceCode}
```

```
Ext-BasicServiceCode ::= CHOICE {
     ext-BearerService                  [2] Ext-BearerServiceCode,
     ext-Teleservice                    [3] Ext-TeleserviceCode}
```

```
EMLPP-Info ::= SEQUENCE {
     maximumentitledPriority            EMLPP-Priority,
     defaultPriority                    EMLPP-Priority,
     extensionContainer                 ExtensionContainer               OPTIONAL,
     ...}
```

```
EMLPP-Priority ::= INTEGER (0..15)
     -- The mapping from the values A,B,0,1,2,3,4 to the integer-value is
     -- specified as follows where A is the highest and 4 is the lowest
     -- priority level
     -- the integer values 7-15 are spare and shall be mapped to value 4
```

```
priorityLevelA                          EMLPP-Priority ::= 6
priorityLevelB                          EMLPP-Priority ::= 5
priorityLevel0                          EMLPP-Priority ::= 0
priorityLevel1                          EMLPP-Priority ::= 1
priorityLevel2                          EMLPP-Priority ::= 2
priorityLevel3                          EMLPP-Priority ::= 3
priorityLevel4                          EMLPP-Priority ::= 4
```

```
MC-SS-Info ::= SEQUENCE {
     ss-Code                            [0] SS-Code,
     ss-Status                          [1] Ext-SS-Status,
     nbrSB                              [2] MaxMC-Bearers,
     nbrUser                            [3] MC-Bearers,
     extensionContainer                 [4] ExtensionContainer          OPTIONAL,
     ...}
```

```
MaxMC-Bearers ::= INTEGER (2..maxNumOfMC-Bearers)
```

```
MC-Bearers ::= INTEGER (1..maxNumOfMC-Bearers)
```

```
maxNumOfMC-Bearers  INTEGER ::= 7
```

```
Ext-SS-Status ::= OCTET STRING (SIZE (1..5))

     -- OCTET 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 3GPP TS 23.011

     -- bit 4: "Q bit"

     -- bit 3: "P bit"

     -- bit 2: "R bit"

     -- bit 1: "A bit"

     -- OCTETS 2-5: reserved for future use. They shall be discarded if
     -- received and not understood.
```

```
   -- data types for geographic location

AgeOfLocationInformation ::= INTEGER (0..32767)
-- the value represents the elapsed time in minutes since the last
-- network contact of the mobile station (i.e. the actuality of the
-- location information).
-- value "0" indicates that the MS is currently in contact with the
--           network
-- value "32767" indicates that the location information is at least
--               32767 minutes old
```

```
END
```

17.7.9 Teleservice Codes

```
MAP-TS-Code {
   ccitt identified-organization (4) etsi (0) mobileDomain (0)
   gsm-Network (1) modules (3) map-TS-Code (19) version6 (6)}

DEFINITIONS

::=

BEGIN
```

```
TeleserviceCode ::= OCTET STRING (SIZE (1))
     -- This type is used to represent the code identifying a single
     -- teleservice, a group of teleservices, or all teleservices. The
     -- services are defined in TS 3GPP TS 22.003[4].
     -- The internal structure is defined as follows:

     -- bits 87654321: group (bits 8765) and specific service
     -- (bits 4321)
```

```
Ext-TeleserviceCode ::= OCTET STRING (SIZE (1..5))
     -- This type is used to represent the code identifying a single
     -- teleservice, a group of teleservices, or all teleservices. The
     -- services are defined in TS 3GPP TS 22.003[4].
     -- The internal structure is defined as follows:

     -- OCTET 1:
     -- bits 87654321: group (bits 8765) and specific service
     -- (bits 4321)

     -- OCTETS 2-5: reserved for future use. If received the
    -- Ext-TeleserviceCode shall be
     -- treated according to the exception handling defined for the
     -- operation that uses this type.

     -- Ext-TeleserviceCode includes all values defined for TeleserviceCode.
```

```
allTeleservices                         TeleserviceCode ::= '00000000'B
```

```
allSpeechTransmissionServices           TeleserviceCode ::= '00010000'B
telephony                               TeleserviceCode ::= '00010001'B
emergencyCalls                          TeleserviceCode ::= '00010010'B
```

```
allShortMessageServices                 TeleserviceCode ::= '00100000'B
shortMessageMT-PP                       TeleserviceCode ::= '00100001'B
shortMessageMO-PP                       TeleserviceCode ::= '00100010'B
```

```
allFacsimileTransmissionServices        TeleserviceCode ::= '01100000'B
facsimileGroup3AndAlterSpeech           TeleserviceCode ::= '01100001'B
automaticFacsimileGroup3                TeleserviceCode ::= '01100010'B
facsimileGroup4                         TeleserviceCode ::= '01100011'B
```

```
-- The following non-hierarchical Compound Teleservice Groups
-- are defined in 3GPP TS 22.030:
allDataTeleservices                     TeleserviceCode ::= '01110000'B
     -- covers Teleservice Groups 'allFacsimileTransmissionServices'
     -- and 'allShortMessageServices'
allTeleservices-ExeptSMS                TeleserviceCode ::= '10000000'B
     -- covers Teleservice Groups 'allSpeechTransmissionServices' and
     -- 'allFacsimileTransmissionServices'
--
-- Compound Teleservice Group Codes are only used in call
-- independent supplementary service operations, i.e. they
-- are not used in InsertSubscriberData or in
-- DeleteSubscriberData messages.
```

```
allVoiceGroupCallServices               TeleserviceCode ::= '10010000'B
voiceGroupCall                          TeleserviceCode ::= '10010001'B
voiceBroadcastCall                      TeleserviceCode ::= '10010010'B
```

```
allPLMN-specificTS                      TeleserviceCode ::= '11010000'B
plmn-specificTS-1                       TeleserviceCode ::= '11010001'B
plmn-specificTS-2                       TeleserviceCode ::= '11010010'B
plmn-specificTS-3                       TeleserviceCode ::= '11010011'B
plmn-specificTS-4                       TeleserviceCode ::= '11010100'B
plmn-specificTS-5                       TeleserviceCode ::= '11010101'B
plmn-specificTS-6                       TeleserviceCode ::= '11010110'B
plmn-specificTS-7                       TeleserviceCode ::= '11010111'B
plmn-specificTS-8                       TeleserviceCode ::= '11011000'B
plmn-specificTS-9                       TeleserviceCode ::= '11011001'B
plmn-specificTS-A                       TeleserviceCode ::= '11011010'B
plmn-specificTS-B                       TeleserviceCode ::= '11011011'B
plmn-specificTS-C                       TeleserviceCode ::= '11011100'B
plmn-specificTS-D                       TeleserviceCode ::= '11011101'B
plmn-specificTS-E                       TeleserviceCode ::= '11011110'B
plmn-specificTS-F                       TeleserviceCode ::= '11011111'B
```

```
END
```

17.7.10 Bearer Service Codes

```
MAP-BS-Code {
   ccitt identified-organization (4) etsi (0) mobileDomain (0)
   gsm-Network (1) modules (3) map-BS-Code (20) version6 (6)}

DEFINITIONS

::=

BEGIN
```

```
BearerServiceCode ::= OCTET STRING (SIZE (1))
     -- This type is used to represent the code identifying a single
     -- bearer service, a group of bearer services, or all bearer
     -- services. The services are defined in TS 3GPP TS 22.002[3].
     -- The internal structure is defined as follows:
     --
     -- plmn-specific bearer services:
     -- bits 87654321: defined by the HPLMN operator

     -- rest of bearer services:
     -- bit 8: 0 (unused)
     -- bits 7654321: group (bits 7654), and rate, if applicable
     -- (bits 321)
```

```
Ext-BearerServiceCode ::= OCTET STRING (SIZE (1..5))
     -- This type is used to represent the code identifying a single
     -- bearer service, a group of bearer services, or all bearer
     -- services. The services are defined in TS 3GPP TS 22.002[3].
     -- The internal structure is defined as follows:
     --
     -- OCTET 1:
     -- plmn-specific bearer services:
     -- bits 87654321: defined by the HPLMN operator
     --
     -- rest of bearer services:
     -- bit 8: 0 (unused)
     -- bits 7654321: group (bits 7654), and rate, if applicable
     -- (bits 321)

     -- OCTETS 2-5: reserved for future use. If received the
    -- Ext-TeleserviceCode shall be
     -- treated according to the exception handling defined for the
     -- operation that uses this type.


     -- Ext-BearerServiceCode includes all values defined for BearerServiceCode.
```

```
allBearerServices                    BearerServiceCode ::= '00000000'B
```

```
allDataCDA-Services                  BearerServiceCode ::= '00010000'B
dataCDA-300bps                       BearerServiceCode ::= '00010001'B
dataCDA-1200bps                      BearerServiceCode ::= '00010010'B
dataCDA-1200-75bps                   BearerServiceCode ::= '00010011'B
dataCDA-2400bps                      BearerServiceCode ::= '00010100'B
dataCDA-4800bps                      BearerServiceCode ::= '00010101'B
dataCDA-9600bps                      BearerServiceCode ::= '00010110'B
general-dataCDA                      BearerServiceCode ::= '00010111'B
```

```
allDataCDS-Services                  BearerServiceCode ::= '00011000'B
dataCDS-1200bps                      BearerServiceCode ::= '00011010'B
dataCDS-2400bps                      BearerServiceCode ::= '00011100'B
dataCDS-4800bps                      BearerServiceCode ::= '00011101'B
dataCDS-9600bps                      BearerServiceCode ::= '00011110'B
general-dataCDS                      BearerServiceCode ::= '00011111'B
```

```
allPadAccessCA-Services                 BearerServiceCode ::= '00100000'B
padAccessCA-300bps                      BearerServiceCode ::= '00100001'B
padAccessCA-1200bps                     BearerServiceCode ::= '00100010'B
padAccessCA-1200-75bps                  BearerServiceCode ::= '00100011'B
padAccessCA-2400bps                     BearerServiceCode ::= '00100100'B
padAccessCA-4800bps                     BearerServiceCode ::= '00100101'B
padAccessCA-9600bps                     BearerServiceCode ::= '00100110'B
general-padAccessCA                     BearerServiceCode ::= '00100111'B
```

```
allDataPDS-Services                     BearerServiceCode ::= '00101000'B
dataPDS-2400bps                         BearerServiceCode ::= '00101100'B
dataPDS-4800bps                         BearerServiceCode ::= '00101101'B
dataPDS-9600bps                         BearerServiceCode ::= '00101110'B
general-dataPDS                         BearerServiceCode ::= '00101111'B
```

```
allAlternateSpeech-DataCDA              BearerServiceCode ::= '00110000'B
```

```
allAlternateSpeech-DataCDS              BearerServiceCode ::= '00111000'B
```

```
allSpeechFollowedByDataCDA              BearerServiceCode ::= '01000000'B
```

```
allSpeechFollowedByDataCDS              BearerServiceCode ::= '01001000'B
```

```
-- The following non-hierarchical Compound Bearer Service
-- Groups are defined in 3GPP TS 22.030:
allDataCircuitAsynchronous              BearerServiceCode ::= '01010000'B
     -- covers "allDataCDA-Services", "allAlternateSpeech-DataCDA" and
     -- "allSpeechFollowedByDataCDA"
allAsynchronousServices                 BearerServiceCode ::= '01100000'B
     -- covers "allDataCDA-Services", "allAlternateSpeech-DataCDA",
     -- "allSpeechFollowedByDataCDA" and "allPadAccessCDA-Services"
allDataCircuitSynchronous               BearerServiceCode ::= '01011000'B
     -- covers "allDataCDS-Services", "allAlternateSpeech-DataCDS" and
     -- "allSpeechFollowedByDataCDS"
allSynchronousServices                  BearerServiceCode ::= '01101000'B
     -- covers "allDataCDS-Services", "allAlternateSpeech-DataCDS",
     -- "allSpeechFollowedByDataCDS" and "allDataPDS-Services"
--
-- Compound Bearer Service Group Codes are only used in call
-- independent supplementary service operations, i.e. they
-- are not used in InsertSubscriberData or in
-- DeleteSubscriberData messages.
```

```
allPLMN-specificBS                      BearerServiceCode ::= '11010000'B
plmn-specificBS-1                       BearerServiceCode ::= '11010001'B
plmn-specificBS-2                       BearerServiceCode ::= '11010010'B
plmn-specificBS-3                       BearerServiceCode ::= '11010011'B
plmn-specificBS-4                       BearerServiceCode ::= '11010100'B
plmn-specificBS-5                       BearerServiceCode ::= '11010101'B
plmn-specificBS-6                       BearerServiceCode ::= '11010110'B
plmn-specificBS-7                       BearerServiceCode ::= '11010111'B
plmn-specificBS-8                       BearerServiceCode ::= '11011000'B
plmn-specificBS-9                       BearerServiceCode ::= '11011001'B
plmn-specificBS-A                       BearerServiceCode ::= '11011010'B
plmn-specificBS-B                       BearerServiceCode ::= '11011011'B
plmn-specificBS-C                       BearerServiceCode ::= '11011100'B
plmn-specificBS-D                       BearerServiceCode ::= '11011101'B
plmn-specificBS-E                       BearerServiceCode ::= '11011110'B
plmn-specificBS-F                       BearerServiceCode ::= '11011111'B
```

```
END
```

17.7.11 Extension data types

```
MAP-ExtensionDataTypes {
   ccitt identified-organization (4) etsi (0) mobileDomain (0)
   gsm-Network (1) modules (3) map-ExtensionDataTypes (21) version6 (6)}

DEFINITIONS

IMPLICIT TAGS

::=

BEGIN
```

```
EXPORTS

   PrivateExtension,
   ExtensionContainer,
   SLR-ArgExtensionContainer;


-- IOC for private MAP extensions

MAP-EXTENSION  ::= CLASS {
     &ExtensionType                                                  OPTIONAL,
     &extensionId                        OBJECT IDENTIFIER }
     -- The length of the Object Identifier shall not exceed 16 octets and the
     -- number of components of the Object Identifier shall not exceed 16

SLR-ArgExtensionContainer ::= SEQUENCE {
     privateExtensionList                [0]PrivateExtensionList         OPTIONAL,
     slr-Arg-PCS-Extensions              [1]SLR-Arg-PCS-Extensions       OPTIONAL,
     ...}


-- data types

ExtensionContainer ::= SEQUENCE {
     privateExtensionList                [0]PrivateExtensionList         OPTIONAL,
     pcs-Extensions                      [1]PCS-Extensions               OPTIONAL,
     ...}

PrivateExtensionList ::= SEQUENCE SIZE (1..maxNumOfPrivateExtensions) OF
                                         PrivateExtension

PrivateExtension ::= SEQUENCE {
     extId                               MAP-EXTENSION.&extensionId
                                         ({ExtensionSet}),
     extType                             MAP-EXTENSION.&ExtensionType
                                         ({ExtensionSet}{@extId})        OPTIONAL}

maxNumOfPrivateExtensions  INTEGER ::= 10

ExtensionSet                             MAP-EXTENSION ::=
          {...
           -- ExtensionSet is the set of all defined private extensions
     }

     -- Unsupported private extensions shall be discarded if received.

PCS-Extensions ::= SEQUENCE {
     ...}

SLR-Arg-PCS-Extensions ::= SEQUENCE {
     ...,
     na-ESRK-Request                     [0]  NULL                       OPTIONAL }

END
```

17.7.12 Group Call data types

```
MAP-GR-DataTypes {
   ccitt identified-organization (4) etsi (0) mobileDomain (0)
   gsm-Network (1) modules (3) map-GR-DataTypes (23) version6 (6)}

DEFINITIONS

IMPLICIT TAGS

::=

BEGIN

EXPORTS
   PrepareGroupCallArg,
   PrepareGroupCallRes,
   SendGroupCallEndSignalArg,
```

```
   SendGroupCallEndSignalRes,
   ForwardGroupCallSignallingArg,
   ProcessGroupCallSignallingArg
;

IMPORTS
   ISDN-AddressString,
   IMSI,
   EMLPP-Priority,
   ASCI-CallReference
FROM MAP-CommonDataTypes {
   ccitt identified-organization (4) etsi (0) mobileDomain (0)
   gsm-Network (1) modules (3) map-CommonDataTypes (18) version6 (6)}

   Ext-TeleserviceCode
FROM MAP-TS-Code {
   ccitt identified-organization (4) etsi (0) mobileDomain (0)
   gsm-Network (1) modules (3) map-TS-Code (19) version6 (6)}

   Kc
FROM MAP-MS-DataTypes {
   ccitt identified-organization (4) etsi (0) mobileDomain (0)
   gsm-Network (1) modules (3) map-MS-DataTypes (11) version6 (6)}


   ExtensionContainer
FROM MAP-ExtensionDataTypes {
   ccitt identified-organization (4) etsi (0) mobileDomain (0)
   gsm-Network (1) modules (3) map-ExtensionDataTypes (21) version6 (6)}
;
```

```
PrepareGroupCallArg ::= SEQUENCE {
     teleservice                    Ext-TeleserviceCode,
     asciCallReference              ASCI-CallReference,
     codec-Info                     CODEC-Info,
     cipheringAlgorithm             CipheringAlgorithm,
     groupKeyNumber                 [0]GroupKeyNumber           OPTIONAL,
     groupKey                       [1]Kc                       OPTIONAL,
     priority                       [2]EMLPP-Priority           OPTIONAL,
     uplinkFree                     [3] NULL                    OPTIONAL,
     extensionContainer             [4] ExtensionContainer      OPTIONAL,
     ...}
```

```
PrepareGroupCallRes ::= SEQUENCE {
     groupCallNumber                ISDN-AddressString,
     extensionContainer             ExtensionContainer          OPTIONAL,
     ...}
```

```
SendGroupCallEndSignalArg ::= SEQUENCE {
     imsi                           IMSI                        OPTIONAL,
     extensionContainer             ExtensionContainer          OPTIONAL,
     ...}
```

```
SendGroupCallEndSignalRes ::= SEQUENCE {
     extensionContainer             ExtensionContainer          OPTIONAL,
     ...}
```

```
ForwardGroupCallSignallingArg ::= SEQUENCE {
     imsi                           IMSI                        OPTIONAL,
     uplinkRequestAck               [0] NULL                    OPTIONAL,
     uplinkReleaseIndication        [1] NULL                    OPTIONAL,
     uplinkRejectCommand            [2] NULL                    OPTIONAL,
     uplinkSeizedCommand            [3] NULL                    OPTIONAL,
     uplinkReleaseCommand           [4] NULL                    OPTIONAL,
     extensionContainer             ExtensionContainer          OPTIONAL,
     ...,
     stateAttributes                [5] StateAttributes         OPTIONAL }
```

```
ProcessGroupCallSignallingArg ::= SEQUENCE {
     uplinkRequest                  [0] NULL                    OPTIONAL,
     uplinkReleaseIndication        [1] NULL                    OPTIONAL,
     releaseGroupCall               [2] NULL                    OPTIONAL,
     extensionContainer             ExtensionContainer          OPTIONAL,
     ...}
```

```
GroupKeyNumber ::= INTEGER (0..15)
```

```
CODEC-Info ::= OCTET STRING (SIZE (5..10))
     -- Refers to channel type
     -- coded according to GSM 08.08 and including Element identifier and Length
```

```
CipheringAlgorithm ::= OCTET STRING (SIZE (1))
     -- Refers to 'permitted algorithms' in 'encryption information'
     -- coded according to GSM 08.08:

     -- Bits 8-1
     -- 8765 4321
     -- 0000 0001                        No encryption
     -- 0000 0010                        GSM A5/1
     -- 0000 0100                        GSM A5/2
     -- 0000 1000                        GSM A5/3
     -- 0001 0000                        GSM A5/4
     -- 0010 0000                        GSM A5/5
     -- 0100 0000                        GSM A5/6
     -- 1000 0000                        GSM A5/7
```

```
StateAttributes ::= SEQUENCE {
     downlinkAttached                    [5] NULL                OPTIONAL,
     uplinkAttached                      [6] NULL                OPTIONAL,
     dualCommunication                   [7] NULL                OPTIONAL,
     callOriginator                      [8] NULL                OPTIONAL }

     -- Refers to 3GPP TS 24.068 for definitions of StateAttributes fields.
```

```
END
```

17.7.13 Location service data types

```
MAP-LCS-DataTypes {
   ccitt identified-organization (4) etsi (0) mobileDomain (0)
   gsm-Network (1) modules (3) map-LCS-DataTypes (25) version6 (6)}

DEFINITIONS
IMPLICIT TAGS
::=
BEGIN

EXPORTS
   RoutingInfoForLCS-Arg,
   RoutingInfoForLCS-Res,
   ProvideSubscriberLocation-Arg,
   ProvideSubscriberLocation-Res,
   SubscriberLocationReport-Arg,
   SubscriberLocationReport-Res,
   LocationType,
   LCSClientName,
   LCS-QoS,
   Horizontal-Accuracy,
   ResponseTime,
   Ext-GeographicalInformation,
   SupportedGADShapes,
   Add-GeographicalInformation
;
```

```
IMPORTS
   AddressString,
   ISDN-AddressString,
   IMEI,
   IMSI,
   LMSI,
   SubscriberIdentity,
   AgeOfLocationInformation,
   LCSClientExternalID,
   LCSClientInternalID
FROM MAP-CommonDataTypes {
   ccitt identified-organization (4) etsi (0) mobileDomain (0)
   gsm-Network (1) modules (3) map-CommonDataTypes (18) version6 (6)}

   ExtensionContainer,
   SLR-ArgExtensionContainer
FROM MAP-ExtensionDataTypes {
   ccitt identified-organization (4) etsi (0) mobileDomain (0)
   gsm-Network (1) modules (3) map-ExtensionDataTypes (21) version6 (6)}

   USSD-DataCodingScheme,
   USSD-String
FROM MAP-SS-DataTypes {
   ccitt identified-organization (4) etsi (0) mobileDomain (0) gsm-Network (1) modules (3)
   map-SS-DataTypes (14) version6 (6)}
;
```

```
RoutingInfoForLCS-Arg ::= SEQUENCE {
     mlcNumber                          [0] ISDN-AddressString,
     targetMS                           [1] SubscriberIdentity,
     extensionContainer                 [2] ExtensionContainer          OPTIONAL,
     ...}
```

```
RoutingInfoForLCS-Res ::= SEQUENCE {
     targetMS                           [0] SubscriberIdentity,
     lcsLocationInfo                    [1] LCSLocationInfo,
     extensionContainer                 [2] ExtensionContainer          OPTIONAL,
     ...}
```

```
LCSLocationInfo ::= SEQUENCE {
     msc-Number                         ISDN-AddressString,
     lmsi                               [0] LMSI                        OPTIONAL,
     extensionContainer                 [1] ExtensionContainer          OPTIONAL,
     ...}
```

```
ProvideSubscriberLocation-Arg ::= SEQUENCE {
     locationType                      LocationType,
     mlc-Number                        ISDN-AddressString,
     lcs-ClientID                      [0] LCS-ClientID                OPTIONAL,
     privacyOverride                   [1] NULL                        OPTIONAL,
     imsi                              [2] IMSI                        OPTIONAL,
     msisdn                            [3] ISDN-AddressString          OPTIONAL,
     lmsi                              [4] LMSI                        OPTIONAL,
     imei                              [5] IMEI                        OPTIONAL,
     lcs-Priority                      [6] LCS-Priority                OPTIONAL,
     lcs-QoS                           [7] LCS-QoS                     OPTIONAL,
     extensionContainer                [8] ExtensionContainer          OPTIONAL,
     ...,
     supportedGADShapes                [9]  SupportedGADShapes         OPTIONAL }

     -- one of imsi or msisdn is mandatory
```

```
LocationType ::= SEQUENCE {
     locationEstimateType               [0] LocationEstimateType,
     ... }
```

```
LocationEstimateType ::= ENUMERATED {
     currentLocation                    (0),
     currentOrLastKnownLocation         (1),
     initialLocation                    (2),
     ... }
--   exception handling:
--   a ProvideSubscriberLocation-Arg containing an unrecognized LocationEstimateType
--   shall be rejected by the receiver with a return error cause of unexpected data value
```

```
LCS-ClientID ::= SEQUENCE {
     lcsClientType                      [0] LCSClientType,
     lcsClientExternalID                [1] LCSClientExternalID          OPTIONAL,
     lcsClientDialedByMS                [2] AddressString                OPTIONAL,
     lcsClientInternalID                [3] LCSClientInternalID          OPTIONAL,
     lcsClientName                      [4] LCSClientName                OPTIONAL,
     ... }
```

```
LCSClientType ::= ENUMERATED {
     emergencyServices                  (0),
     valueAddedServices                 (1),
     plmnOperatorServices               (2),
     lawfulInterceptServices            (3),
     ... }
     --   exception handling:
     --   unrecognized values may be ignored if the LCS client uses the privacy override
     --   otherwise, an unrecognized value shall be treated as unexpected data by a receiver
     --   a return error shall then be returned if received in a MAP invoke
```

```
LCSClientName ::= SEQUENCE {
     dataCodingScheme                   [0] USSD-DataCodingScheme,
     nameString                         [2] NameString,
     ...}

-- The USSD-DataCodingScheme shall indicate use of the default alphabet through the
-- following encoding
--   bit  7 6 5 4 3 2 1 0
--        0 0 0 0 1 1 1 1
```

```
NameString ::= USSD-String (SIZE (1..maxNameStringLength))
```

```
maxNameStringLength  INTEGER ::= 63
```

```
LCS-Priority ::= OCTET STRING (SIZE (1))
     -- 0 = highest priority
     -- 1 = normal priority
     -- all other values treated as 1
```

```
LCS-QoS ::= SEQUENCE {
     horizontal-accuracy                [0] Horizontal-Accuracy          OPTIONAL,
     verticalCoordinateRequest          [1] NULL                         OPTIONAL,
     vertical-accuracy                  [2] Vertical-Accuracy            OPTIONAL,
     responseTime                       [3] ResponseTime                 OPTIONAL,
     extensionContainer                 [4] ExtensionContainer           OPTIONAL,
     ...}
```

```
Horizontal-Accuracy ::= OCTET STRING (SIZE (1))
     -- bit 8 = 0
     -- bits 7-1 = 7 bit Uncertainty Code defined in 3GPP TS 23.032. The horizontal location
     -- error should be less than the error indicated by the uncertainty code with 67 %
     -- confidence.
```

```
Vertical-Accuracy ::= OCTET STRING (SIZE (1))
     -- bit 8 = 0
     -- bits 7-1 = 7 bit Vertical Uncertainty Code defined in 3GPP TS 23.032.
     -- The vertical location error should be less than the error indicated
     -- by the uncertainty code with 67 % confidence.
```

```
ResponseTime ::= SEQUENCE {
     responseTimeCategory               ResponseTimeCategory,
     ...}
--   note: an expandable SEQUENCE simplifies later addition of a numeric response time.
```

```
ResponseTimeCategory ::= ENUMERATED {
     lowdelay  (0),
     delaytolerant  (1),
     ... }
--   exception handling:
--   an unrecognized value shall be treated the same as value 1 (delaytolerant)
```

```
SupportedGADShapes ::= BIT STRING {
     ellipsoidPoint  (0),
     ellipsoidPointWithUncertaintyCircle (1),
     ellipsoidPointWithUncertaintyEllipse (2),
     polygon (3),
     ellipsoidPointWithAltitude (4),
     ellipsoidPointWithAltitudeAndUncertaintyElipsoid (5),
     ellipsoidArc  (6) } (SIZE (7..16))
-- A node shall mark in the BIT STRING all Shapes defined in 3GPP TS 23.032 it supports.
-- exception handling: bits 7 to 15 shall be ignored if received.
```

```
ProvideSubscriberLocation-Res ::= SEQUENCE {
     locationEstimate                    Ext-GeographicalInformation,
     ageOfLocationEstimate               [0] AgeOfLocationInformation      OPTIONAL,
     extensionContainer                  [1] ExtensionContainer            OPTIONAL,
     ...,
     add-LocationEstimate                [2] Add-GeographicalInformation   OPTIONAL }

-- the add-LocationEstimate parameter shall not be sent to a node that did not indicate the
-- geographic shapes supported in the ProvideSubscriberLocation-Arg.
-- The locationEstimate and the add-locationEstimate parameters shall not be sent if
-- the supportedGADShapes parameter has been received in ProvideSubscriberLocation-Arg
-- and the shape encoded in locationEstimate or add-LocationEstimate is not marked
-- as supported in supportedGADShapes. In such a case ProvideSubscriberLocation
-- shall be rejected with error FacilityNotSupported.
```

```
Ext-GeographicalInformation ::= OCTET STRING (SIZE (1..maxExt-GeographicalInformation))
    -- Refers to geographical Information defined in 3GPP TS 23.032.
    -- This is composed of 1 or more octets with an internal structure according
    -- to 3GPP TS 23.032
    -- Octet 1: Type of shape, only the following shapes in 3GPP TS 23.032 are allowed:
    --       (a) Ellipsoid point with uncertainty circle
    --       (b) Ellipsoid point with uncertainty ellipse
    --       (c) Ellipsoid point with altitude and uncertainty ellipsoid
    --       (d) Ellipsoid Arc
    --       (e) Ellipsoid Point
    -- Any other value in octet 1 shall be treated as invalid
    -- Octets 2 to 8 for case (a) - Ellipsoid point with uncertainty circle
    --       Degrees of Latitude                                        3 octets
    --       Degrees of Longitude                                       3 octets
    --       Uncertainty code                                           1 octet
    -- Octets 2 to 11 for case (b) - Ellipsoid point with uncertainty ellipse:
    --       Degrees of Latitude                                        3 octets
    --       Degrees of Longitude                                       3 octets
    --       Uncertainty semi-major axis                                1 octet
    --       Uncertainty semi-minor axis                                1 octet
    --       Angle of major axis                                        1 octet
    --       Confidence                                                 1 octet
    -- Octets 2 to 14 for case (c) - Ellipsoid point with altitude and uncertainty ellipsoid
    --       Degrees of Latitude                                        3 octets
    --       Degrees of Longitude                                       3 octets
    --       Altitude                                                   2 octets
    --       Uncertainty semi-major axis                                1 octet
    --       Uncertainty semi-minor axis                                1 octet
    --       Angle of major axis                                        1 octet
    --       Uncertainty altitude                                       1 octet
    --       Confidence                                                 1 octet
    -- Octets 2 to 13 for case (d) - Ellipsoid Arc
    --       Degrees of Latitude                                        3 octets
    --       Degrees of Longitude                                       3 octets
    --       Inner radius                                               2 octets
    --       Uncertainty radius                                         1 octet
    --       Offset angle                                               1 octet
    --       Included angle                                             1 octet
    --       Confidence                                                 1 octet
    -- Octets 2 to 7 for case (e) - Ellipsoid Point
    --       Degrees of Latitude                                        3 octets
    --       Degrees of Longitude                                       3 octets

    --
    -- An Ext-GeographicalInformation parameter comprising more than one octet and
    -- containing any other shape or an incorrect number of octets or coding according
    -- to 3GPP TS 23.032 shall be treated as invalid data by a receiver.
    --
    -- An Ext-GeographicalInformation parameter comprising one octet shall be discarded
    -- by the receiver if an Add-GeographicalInformation parameter is received
    -- in the same message.
    --
    -- An Ext-GeographicalInformation parameter comprising one octet shall be treated as
    -- invalid data by the receiver if an Add-GeographicalInformation parameter is not
    -- received in the same message.
```

```
maxExt-GeographicalInformation  INTEGER ::= 20
    -- the maximum length allows for further shapes in 3GPP TS 23.032 to be included in later
    -- versions of 3GPP TS 29.002
```

```
Add-GeographicalInformation ::= OCTET STRING (SIZE (1..maxAdd-GeographicalInformation))
     -- Refers to geographical Information defined in 3GPP TS 23.032.
     -- This is composed of 1 or more octets with an internal structure according to
     -- 3GPP TS 23.032
     -- Octet 1: Type of shape, all the shapes defined in 3GPP TS 23.032 are allowed:
     -- Octets 2 to n (where n is the total number of octets necessary to encode the shape
     -- according to 3GPP TS 23.032) are used to encode the shape itself in accordance with
the
     -- encoding defined in 3GPP TS 23.032
     --
     -- An Add-GeographicalInformation parameter, whether valid or invalid, received
     -- together with a valid Ext-GeographicalInformation parameter in the same message
     -- shall be discarded.
     --
     -- An Add-GeographicalInformation parameter containing any shape not defined in
     -- 3GPP TS 23.032 or an incorrect number of octets or coding according to
     -- 3GPP TS 23.032 shall be treated as invalid data by a receiver if not received
     -- together with a valid Ext-GeographicalInformation parameter in the same message.
```

```
maxAdd-GeographicalInformation  INTEGER ::= 91
     -- the maximum length allows support for all the shapes currently defined in 3GPP TS
23.032
```

```
SubscriberLocationReport-Arg ::= SEQUENCE {
     lcs-Event                          LCS-Event,
     lcs-ClientID                       LCS-ClientID,
     lcsLocationInfo                    LCSLocationInfo,
     msisdn                             [0] ISDN-AddressString            OPTIONAL,
     imsi                               [1] IMSI                          OPTIONAL,
     imei                               [2] IMEI                          OPTIONAL,
     na-ESRD                            [3] ISDN-AddressString            OPTIONAL,
     na-ESRK                            [4] ISDN-AddressString            OPTIONAL,
     locationEstimate                   [5] Ext-GeographicalInformation   OPTIONAL,
     ageOfLocationEstimate              [6] AgeOfLocationInformation      OPTIONAL,
     slr-ArgExtensionContainer          [7] SLR-ArgExtensionContainer     OPTIONAL,
     ... ,
     add-LocationEstimate               [8] Add-GeographicalInformation   OPTIONAL }

     -- one of msisdn or imsi is mandatory
     -- a location estimate that is valid for the locationEstimate parameter should
     -- be transferred in this parameter in preference to the add-LocationEstimate
```

```
LCS-Event ::= ENUMERATED {
     emergencyCallOrigination  (0),
     emergencyCallRelease  (1),
     mo-lr  (2),
     ... }
     --   exception handling:
     --   a SubscriberLocationReport-Arg containing an unrecognized LCS-Event
     --   shall be rejected by a receiver with a return error cause of unexpected data value
```

```
SubscriberLocationReport-Res ::= SEQUENCE {
     extensionContainer                 ExtensionContainer                OPTIONAL,
     ...,
     na-ESRK                            [0] ISDN-AddressString            OPTIONAL }
```

```
END
```

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 capitalised).

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 capitalised).

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 summarised 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 released is requested, a comment is attached to the signal which represents the MAP-CLOSE request primitive. In the absence of comment, a normal release is assumed.

18.2.4 Version handling at dialogue establishment

Unless explicitly indicated in subsequent clauses, 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-OPEN 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-OPEN 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".

If version 4 is selected and a MAP-OPEN 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, version two or version three 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; if it refers to a version three context, the associated V3 procedure is executed, otherwise the associated V4 procedure is executed.

18.2.5 Abort Handling

Unless explicitly indicated in subsequent clauses, 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 clause 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 initialisation 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 clause 14.3). There is a fixed 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 implicitly 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 clause 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 clause:

Process Update Location Area:

Update_Location_Area_VLR, clause 19.1.1.3;

Process Update Location:

Initiator: Update_Location_Area_VLR, clause 19.1.1.3;

Responder: Update_Location_HLR, clause 19.1.1.4;

Process Send Identification:

Initiator: Send_Identification_VLR, clause 19.1.1.x;

Responder: Send_Identification_PVLR, clause 19.1.1.5;

Process Cancel Location:

Initiator: Cancel_Location_HLR, clause 19.1.2.2;

Responder: Cancel_Location_VLR, clause 19.1.2.3;

Process Purge MS:

Initiator: Purge_MS_VLR, clause 19.1.4.2;

Responder: Purge_MS_HLR, clause 19.1.4.3.

For GPRS subscribers, this clause 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 clause:

Process GPRS Update Location:

Initiator: GPRS_Update_Location_Area_VLR, clause 19.1.1.3, or

SGSN_Update_HLR, clause 19.1.1.8,

Responder: Update_GPRS_Location_HLR, clause 19.1.1.4;

Process Cancel Location:

Initiator: Cancel_GPRS_Location_HLR, clause 19.1.2.2;

Responder: Cancel_Location_SGSN, clause 19.1.2.4;

Process Purge MS:

Initiator: Purge_MS_SGSN, clause 19.1.4.4;

Responder: Purge_MS_HLR, clause 19.1.4.3.

The following existing process is also used for GPRS subscribers :

Process Subscriber Present HLR:

Initiator: Subscriber_Present_HLR, clause 19.1.1.7;

Responder: Short_Message_Alert_IWMSC, clause 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.

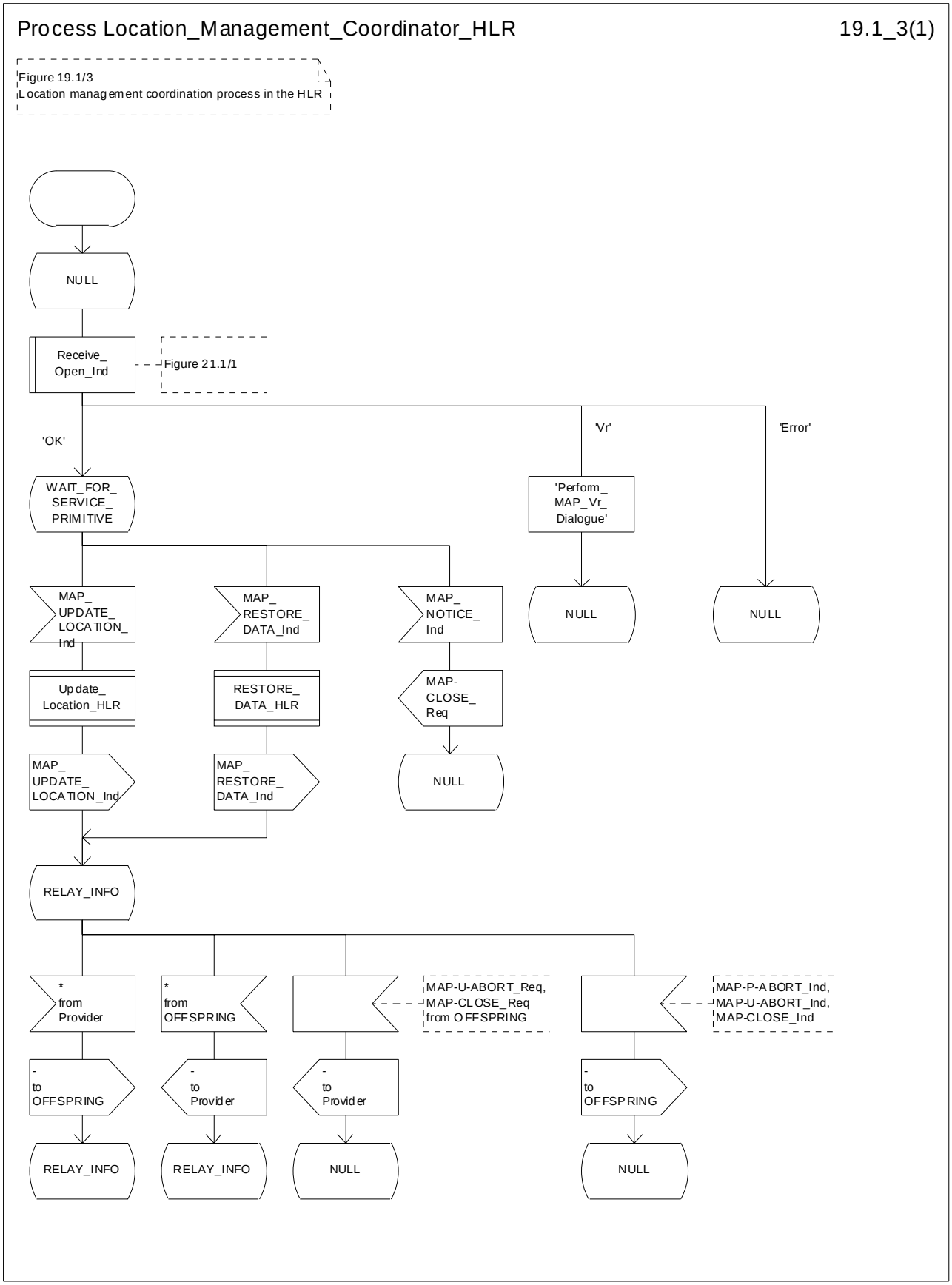


Figure 19.1/3: Process Location_Management_Coordinator_HLR

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 3GPP TS 23.012 [23] and 3GPP TS 23.060[104]). The use of the IMSI Detach / Attach feature is optional for the network operator.

To minimise 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 3GPP TS 23.060[104]. 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 clause 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 clause 8.1);(**)
MAP_UPDATE_LOCATION	(see clause 8.1);(**)
MAP_UPDATE_GPRS_LOCATION	(see clause 8.1) (*);
MAP_CANCEL_LOCATION	(see clause 8.1);
MAP_INSERT_SUBSCRIBER_DATA	(see clause 8.8);
MAP_SEND_IDENTIFICATION	(see clause 8.1) (**);
MAP_PROVIDE_IMSI	(see clause 8.9) (**);
MAP_AUTHENTICATE	(see clause 8.5) (**);
MAP_SET_CIPHERING_MODE	(see clause 8.6) (**);
MAP_FORWARD_NEW_TMSI	(see clause 8.9) (**);

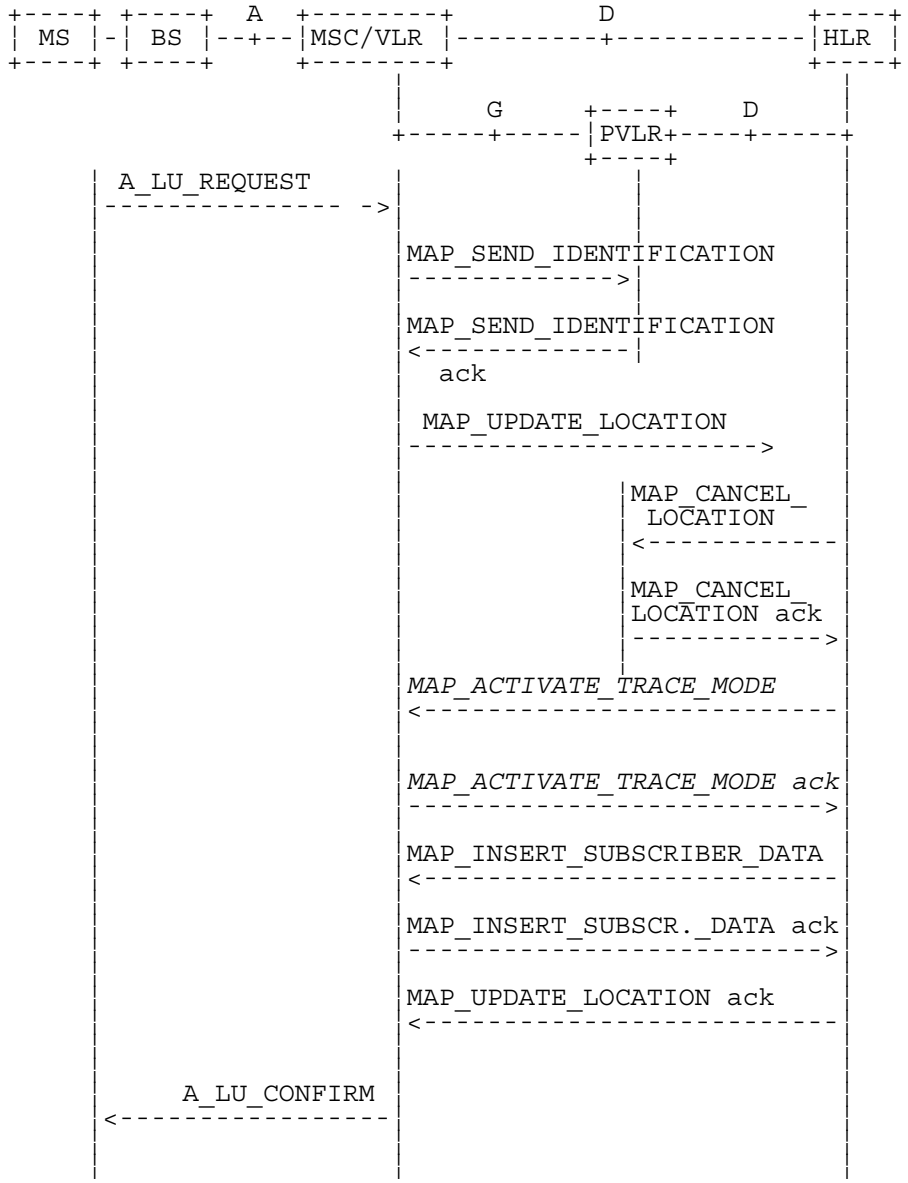
MAP_CHECK_IMEI (see clause 8.7) (**);

MAP_ACTIVATE_TRACE_MODE (see clause 9.2);

MAP_TRACE_SUBSCRIBER_ACTIVITY (see clause 9.2) (**).

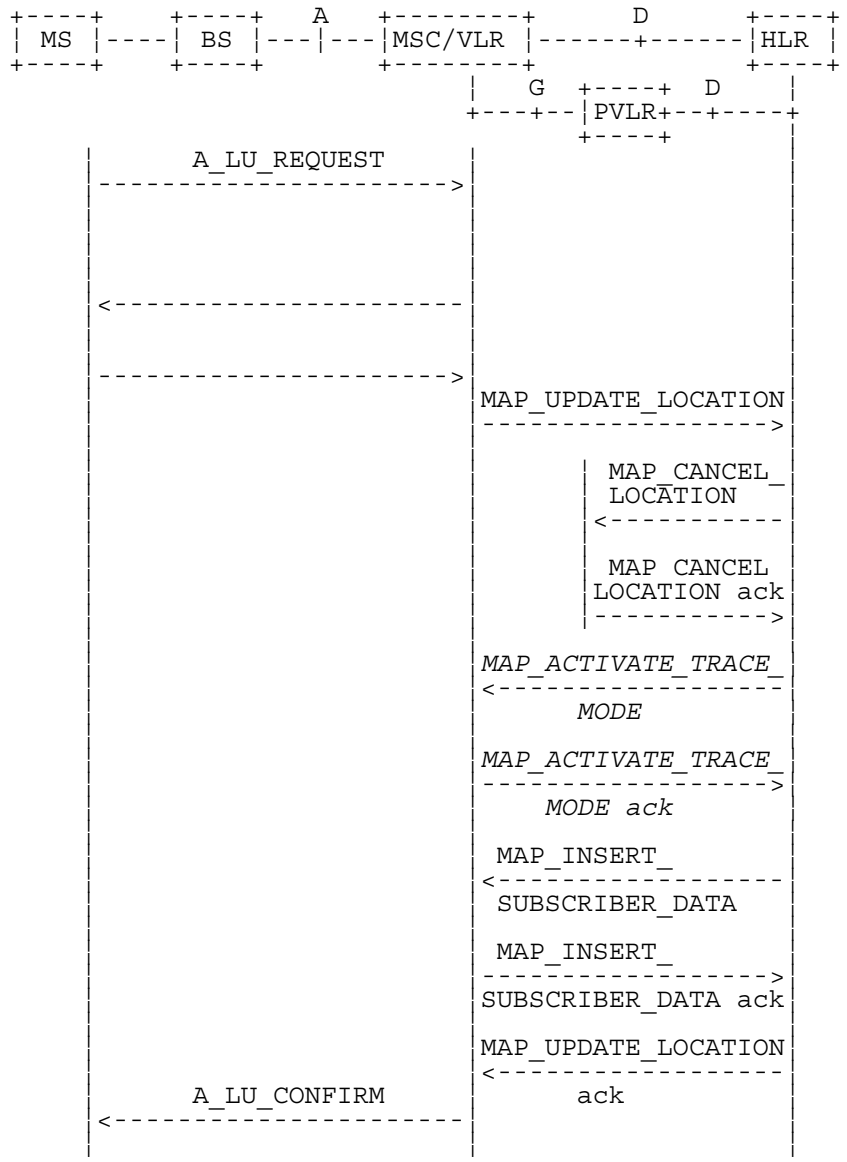
(*): only used in SGSN and HLR for GPRS

(**): not used in SGSN



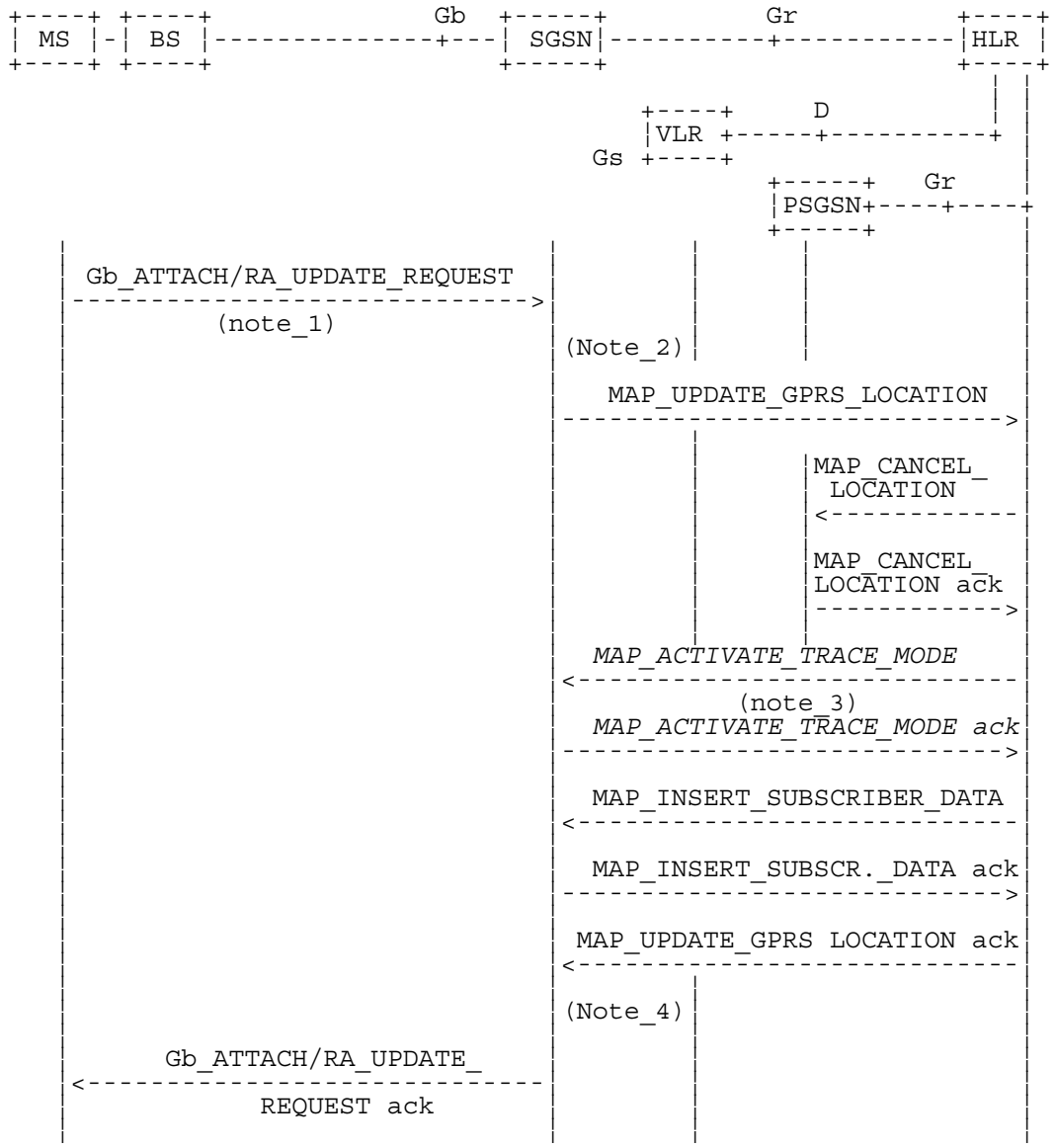
NOTE: Procedures shown in italics are optional.

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

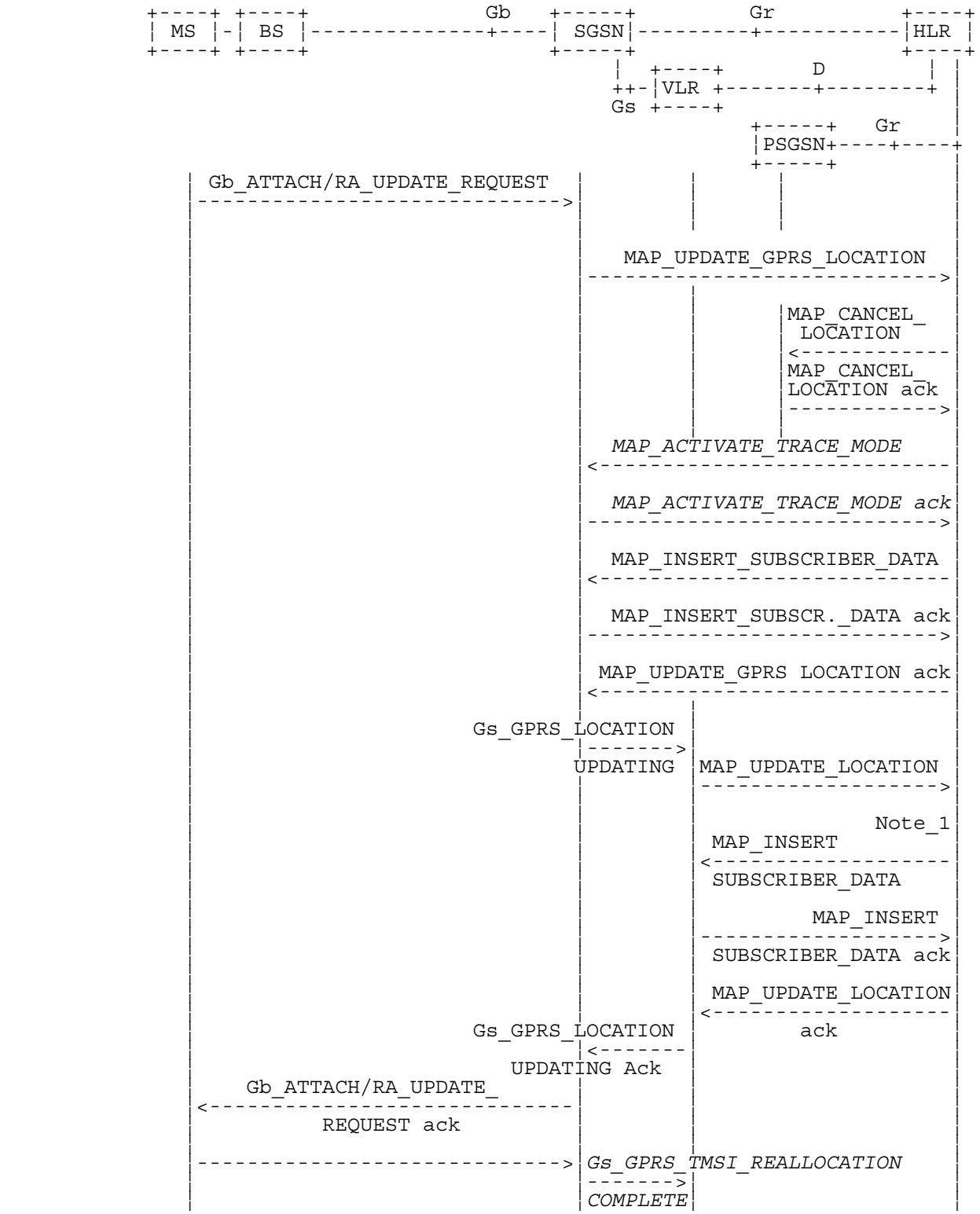
NOTE 1: For details of the procedure on the radio path, see GSM 08.18. The services shown in dotted lines indicate the trigger provided by the signalling on the radio path, and the signalling triggered on the radio path.

NOTE 2: For security functions (authentication, ciphering, IMEI check) triggering refer to 3GPP TS 23.060[104]. MAP processes invoked for those procedures are described in section 25.

NOTE 3: Optional services are printed in *italics*.

NOTE 4: Refer to 3GPP TS 23.060[104] 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 1: The optional procedures in figure 19.1.1/14 apply here respectively. For details of the procedure on the Gs-interface, see 3GPP TS 29.018 [106].

NOTE 2: 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 3GPP TS 23.060[104] 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 exits 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 3GPP TS 29.018 [106]) 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 3GPP TS 29.018 [106]) 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 3GPP TS 29.018 [106]) 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 3GPP TS 29.018 [106]) 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 3GPP TS 29.018 [106]) 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 3GPP TS 29.018 [106]) 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 clause 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 3GPP TS 23.060[104]). 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 clause 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_Subs_Data_VLR defined in clause 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 clause 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.

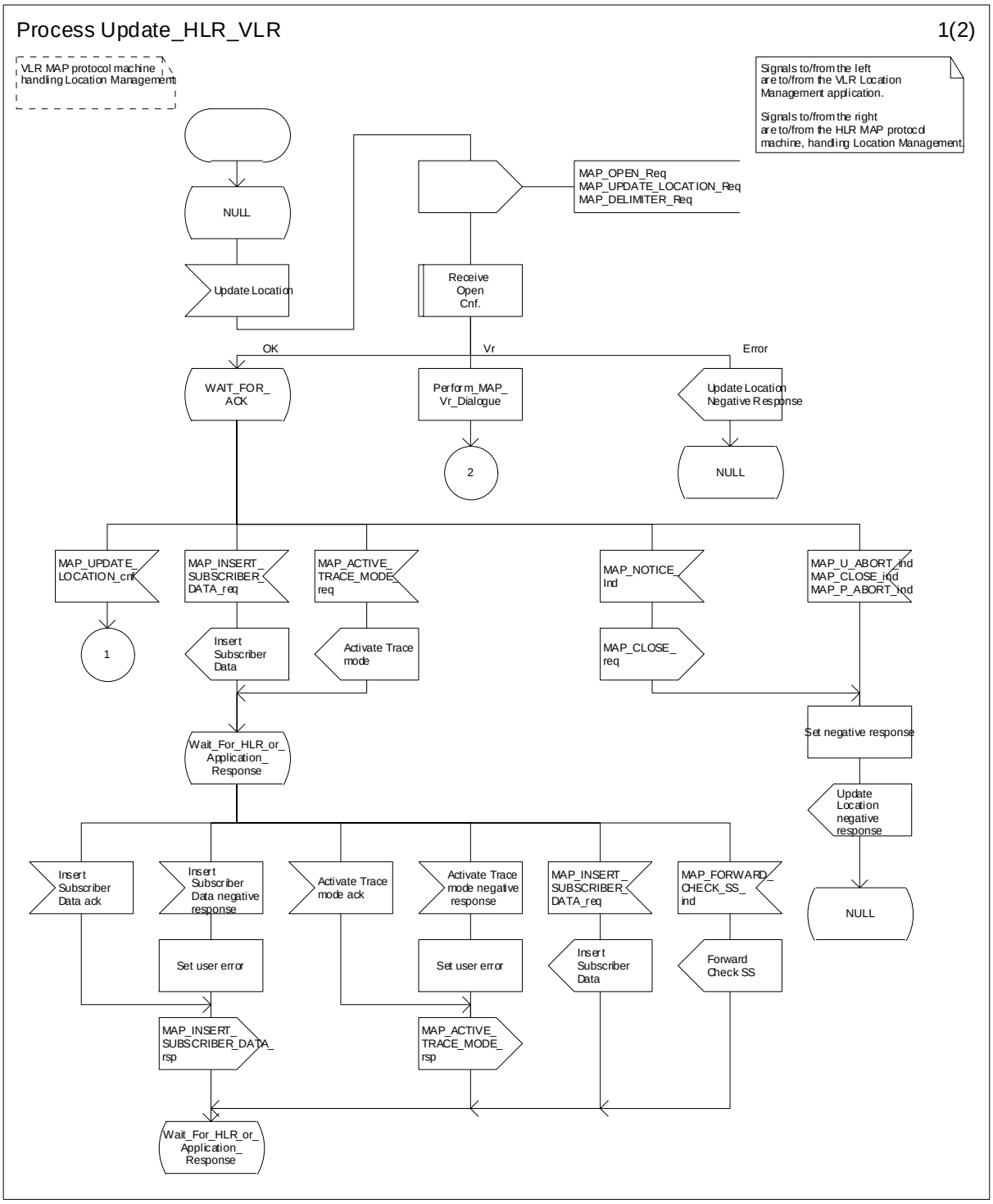


Figure 19.1.1/6 (sheet 1 of 2): Process Update_HLR_VLR

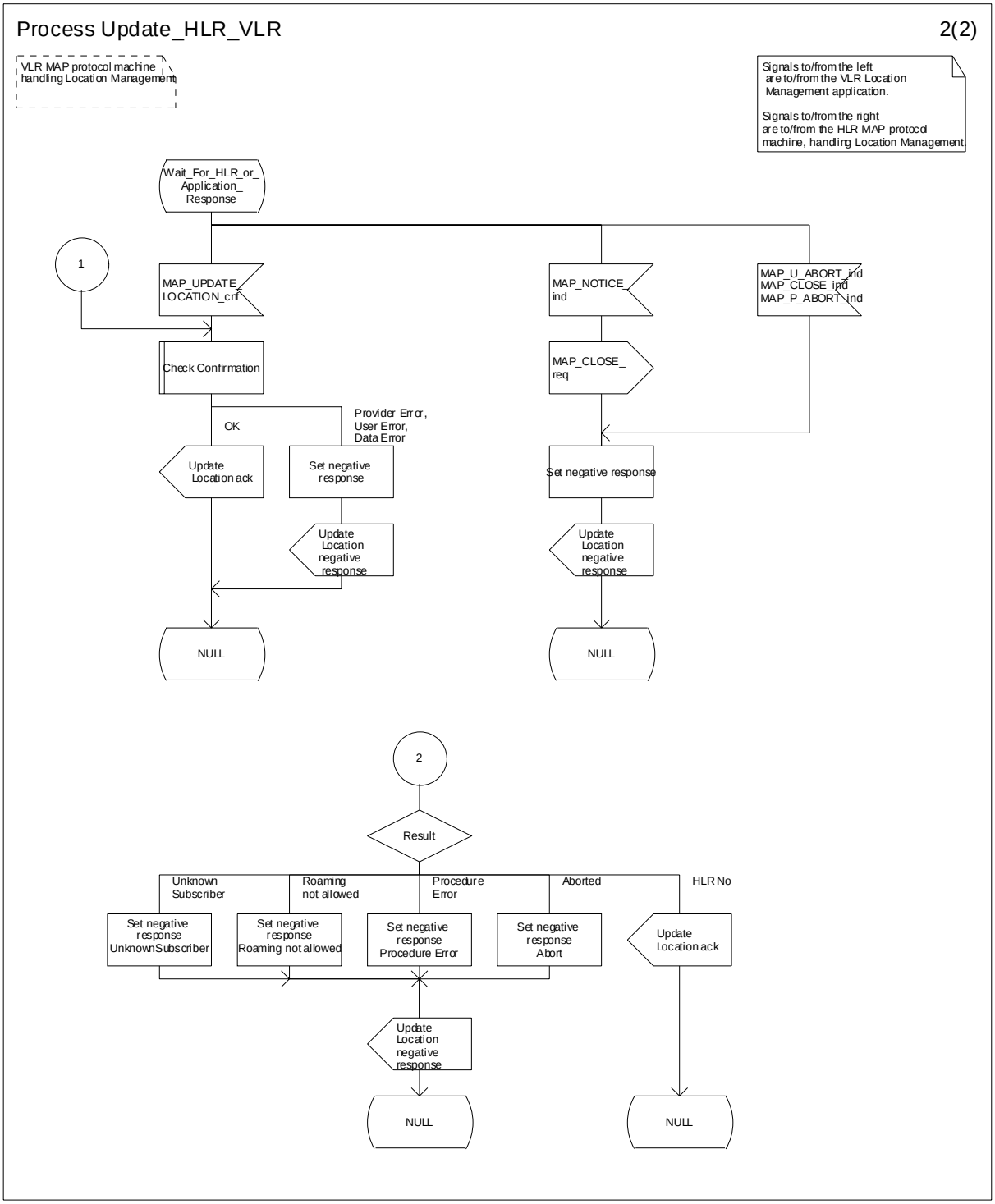


Figure 19.1.1/6 (sheet 2 of 2): Process Update_HLR_VLR

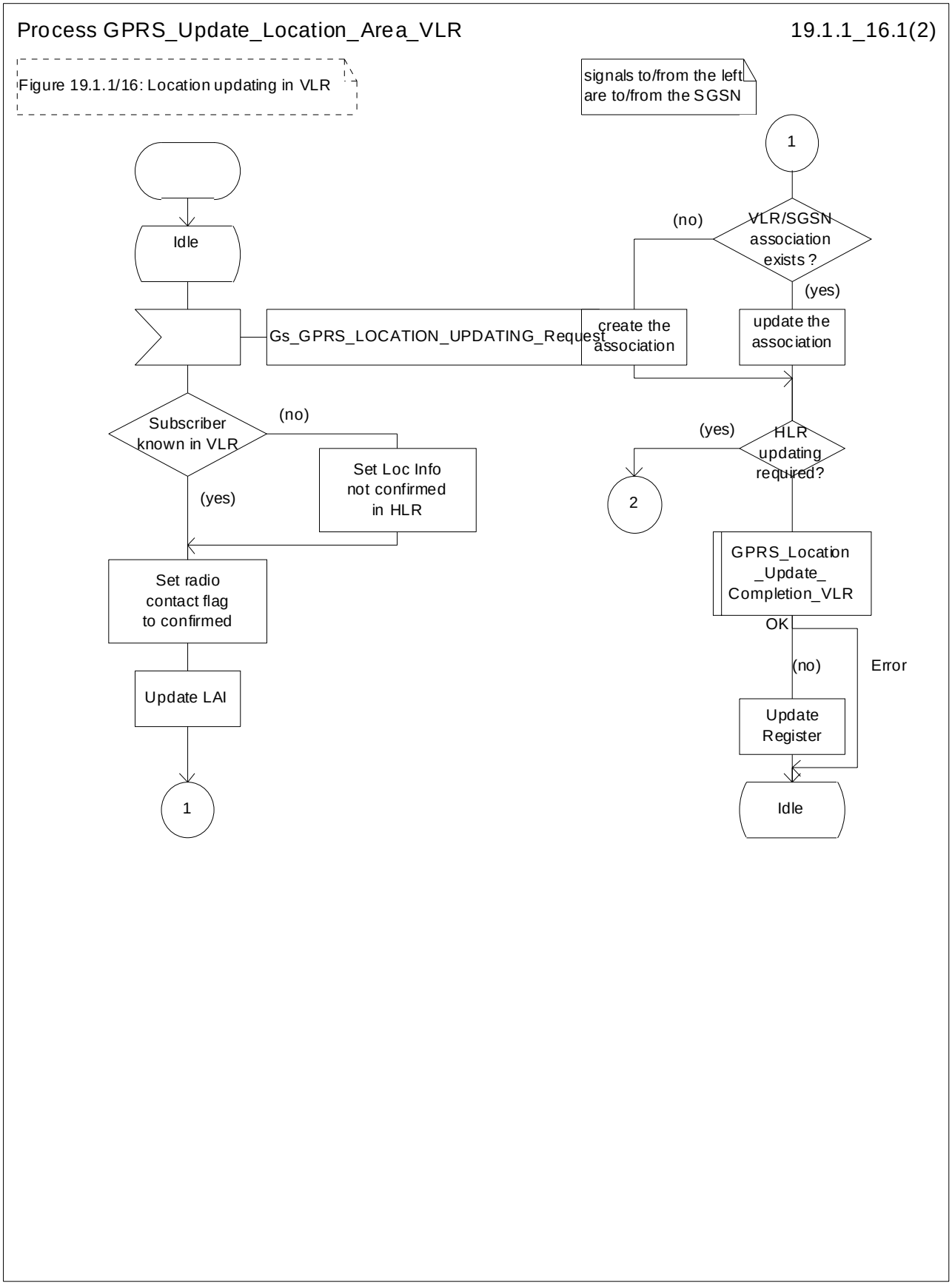


Figure 19.1.1/16 (sheet 1 of 2): Process GPRS_Update_Location_Area_VLR

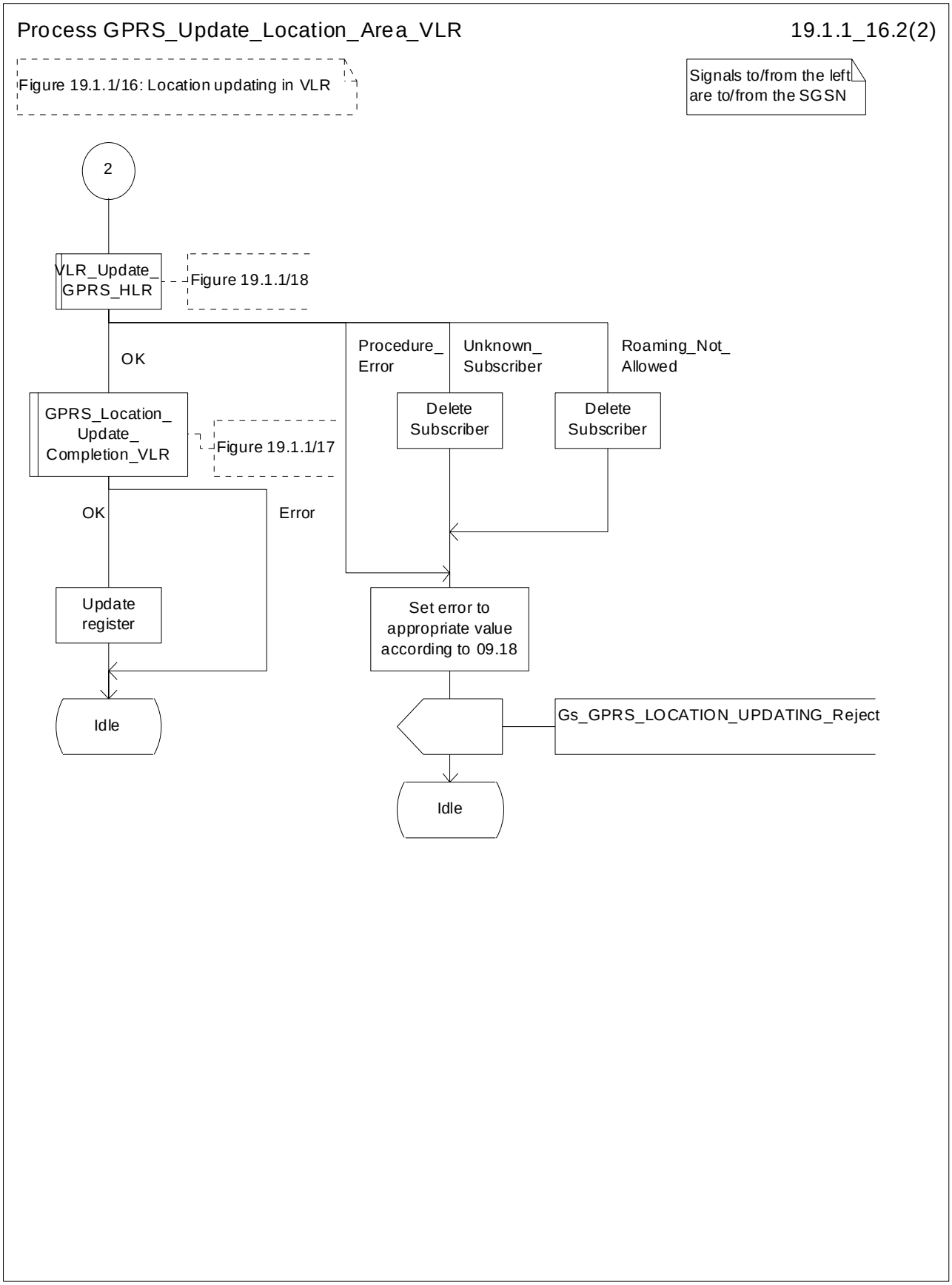


Figure 19.1.1/16 (sheet 2 of 2): Process GPRS_Update_Location_Area_VLR

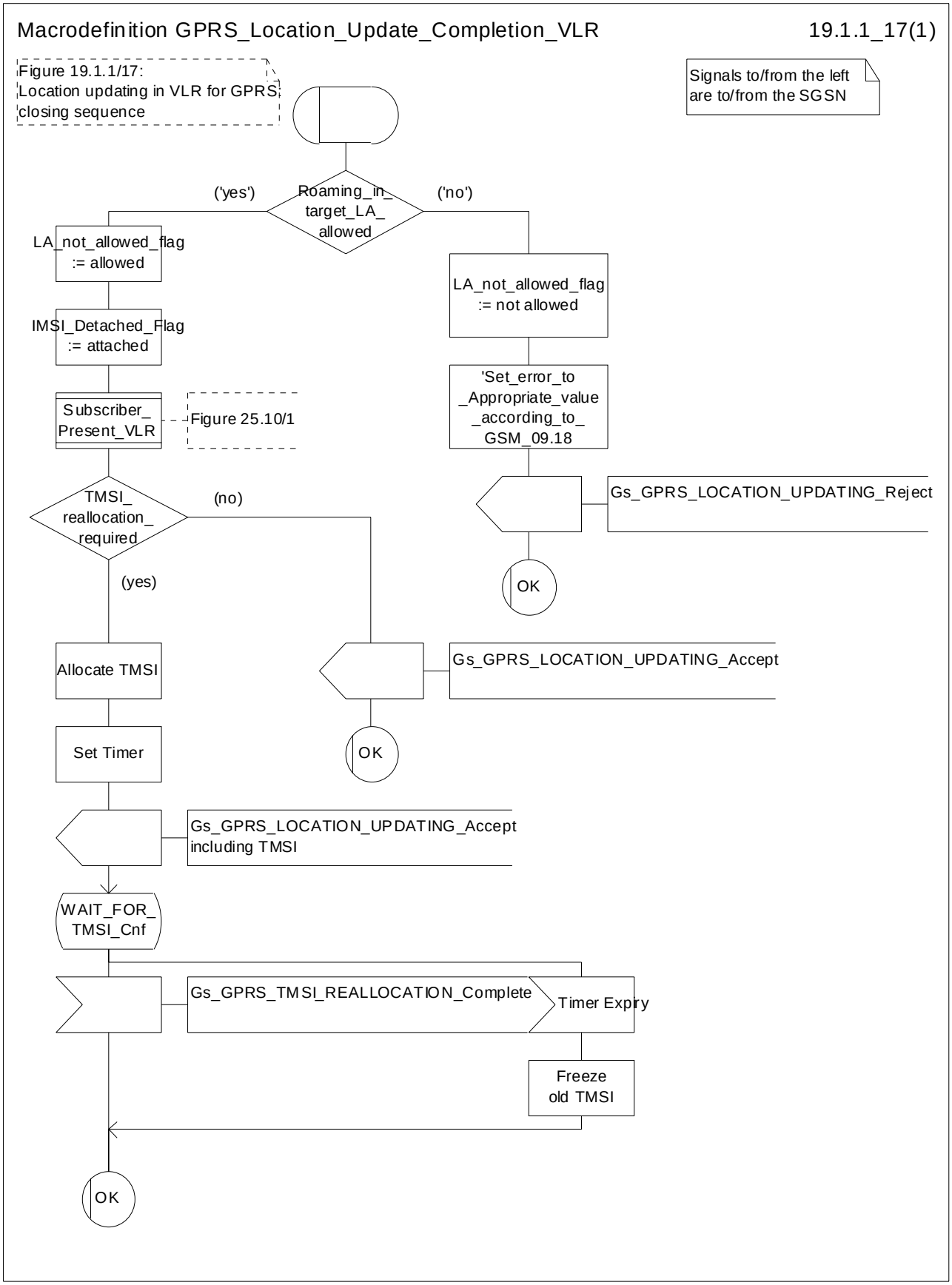


Figure 19.1.1/17: Macro GPRS_Location_Update_Completion_VLR

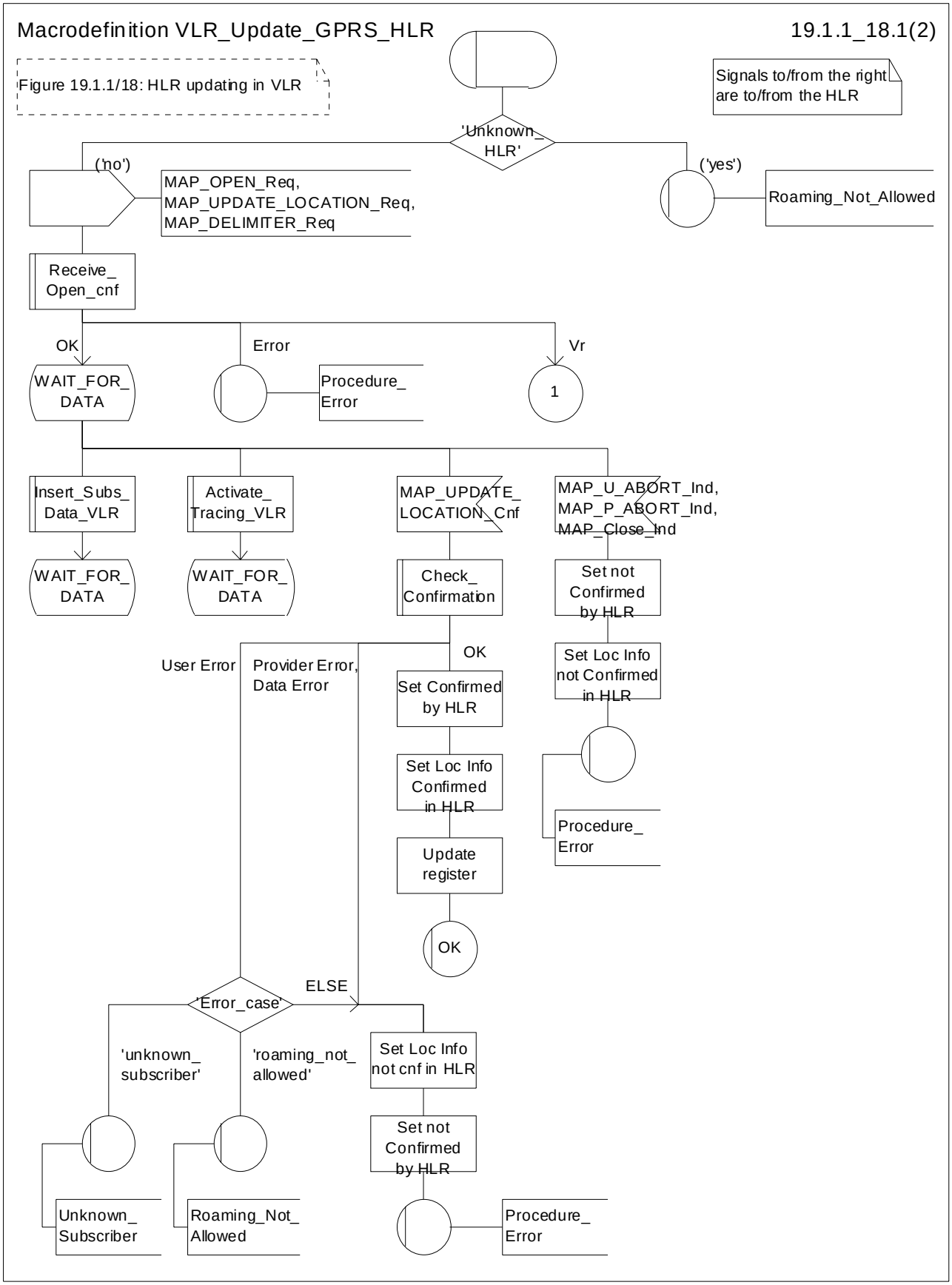


Figure 19.1.1/18 (sheet 1 of 2): Macro VLR_Update_GPRS_HLR

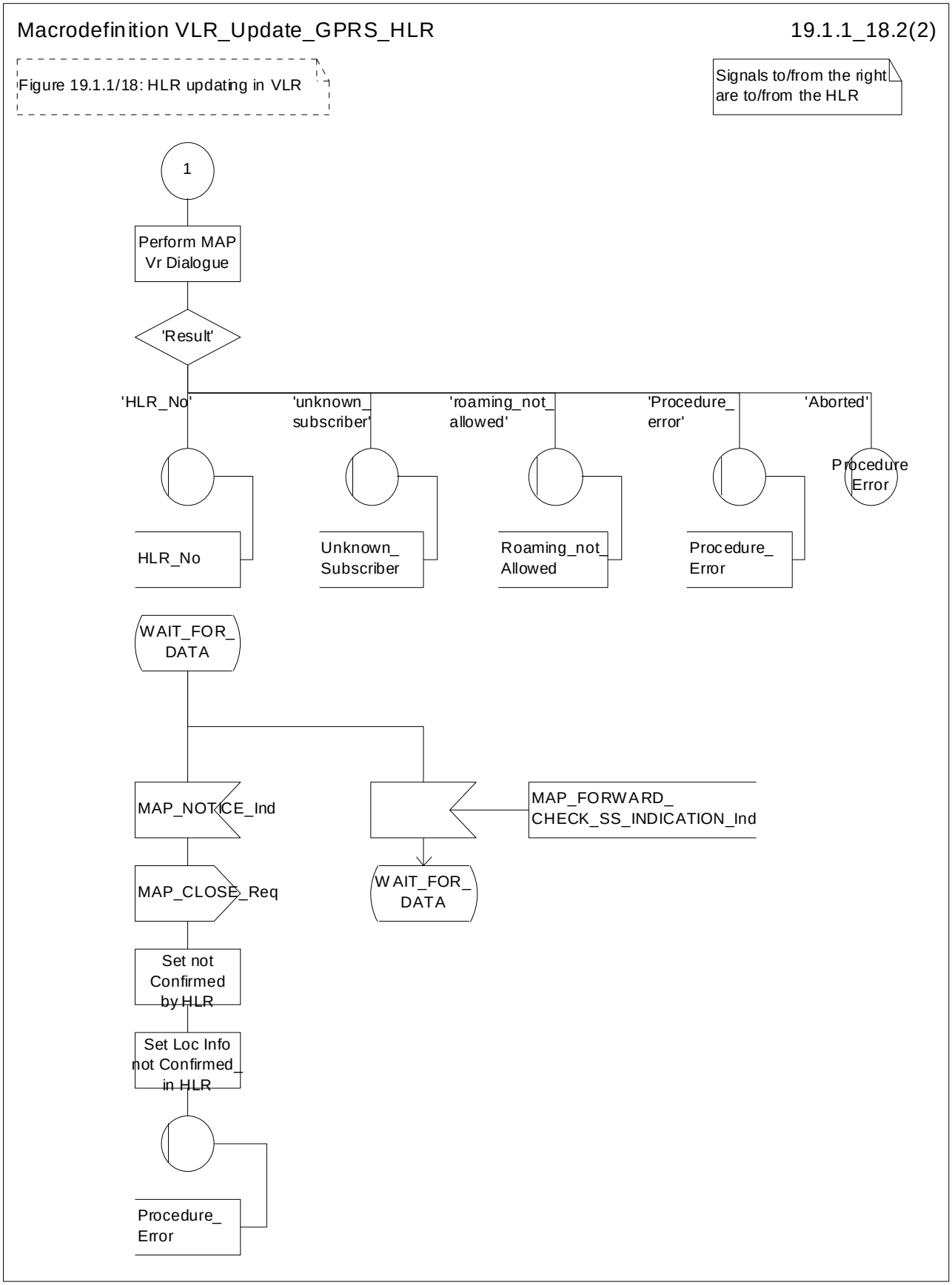


Figure 19.1.1/18 (sheet 2 of 2): Macro VLR_Update_GPRS_HLR

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 clause 25.1;
- Check_indication, defined in clause 25.2;
- Insert_Subs_Data_In_SGSN_Framed_HLR, described in clause 19.4;
- Control_Tracing_HLR_with_SGSN, described in clause 25.9;

and the processes Cancel_Location_HLR (see clause 19.1.2) and Subscriber_Present_HLR (see clause 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_LOCATION response (see Check_indication macro defined in clause 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 deactive 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 clause 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 clause 25.9.

Thereafter, the macro Insert_Subs_Data_In_SGSN_Framed_HLR described in clause 19.4 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 clause 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.

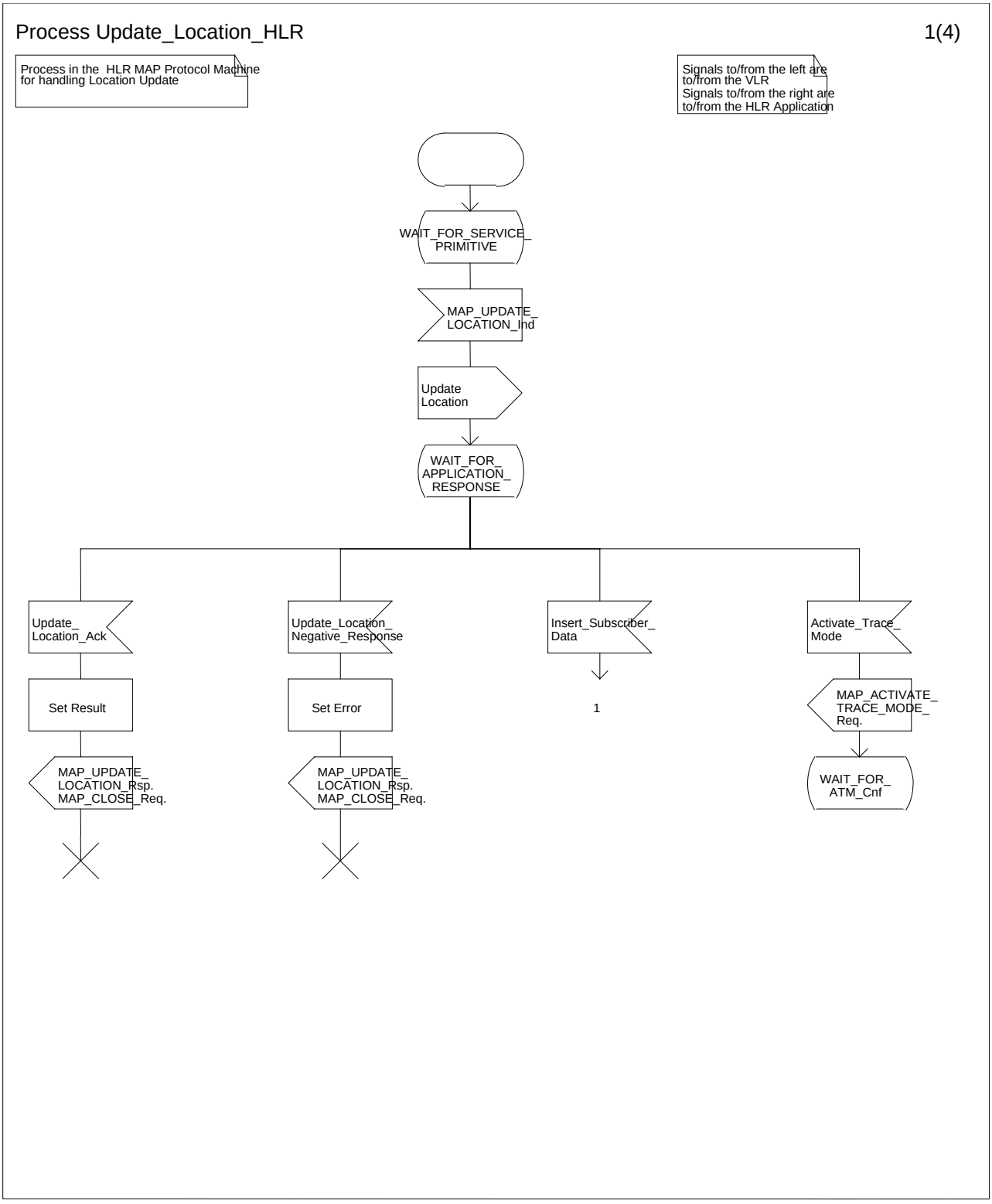


Figure 19.1.1/9 (sheet 1 of 4): Process Update_Location_HLR

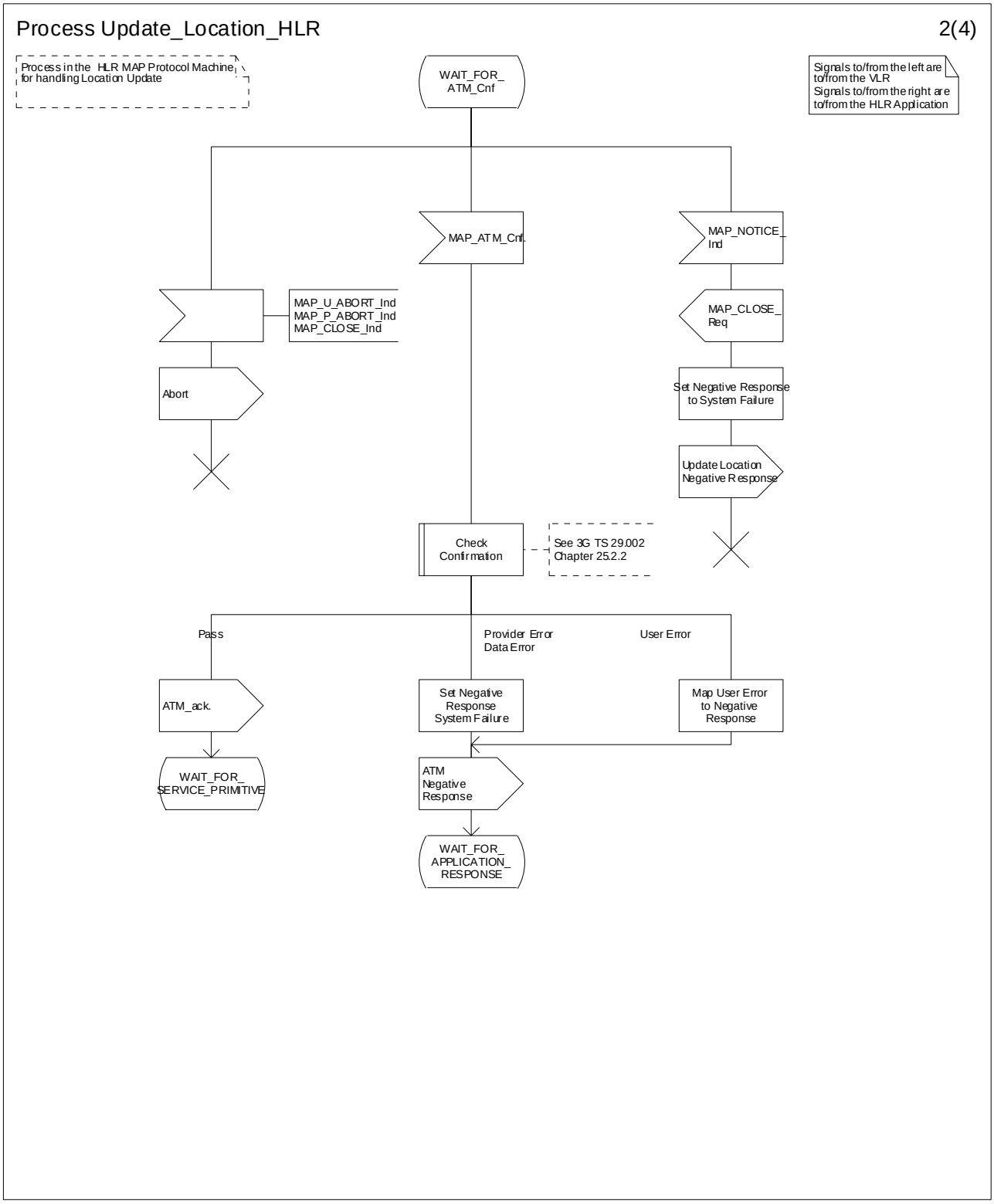


Figure 19.1.1/9 (sheet 2 of 4): Process Update_Location_HLR

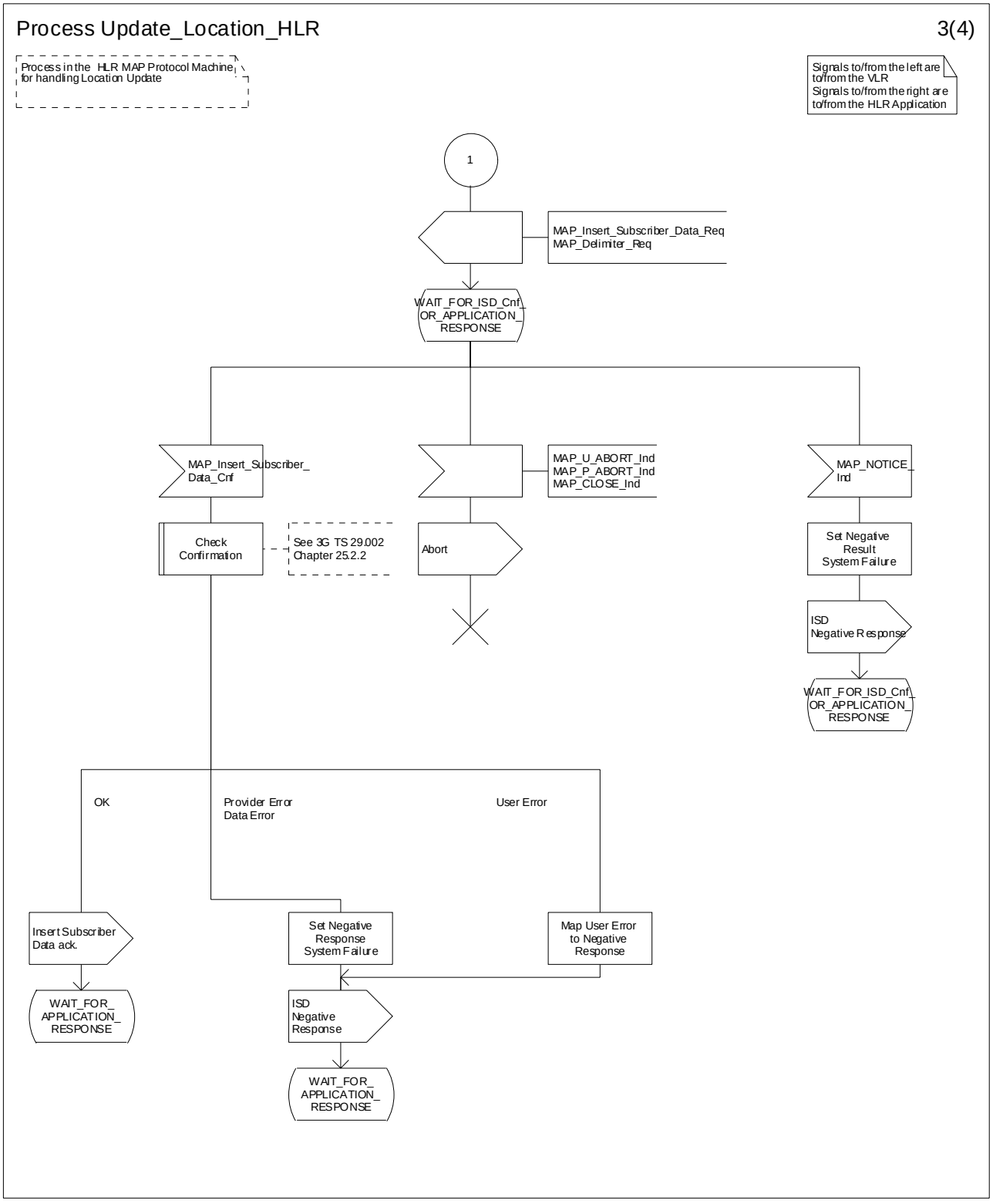


Figure 19.1.1/9 (sheet 3 of 4): Process Update_Location_HLR

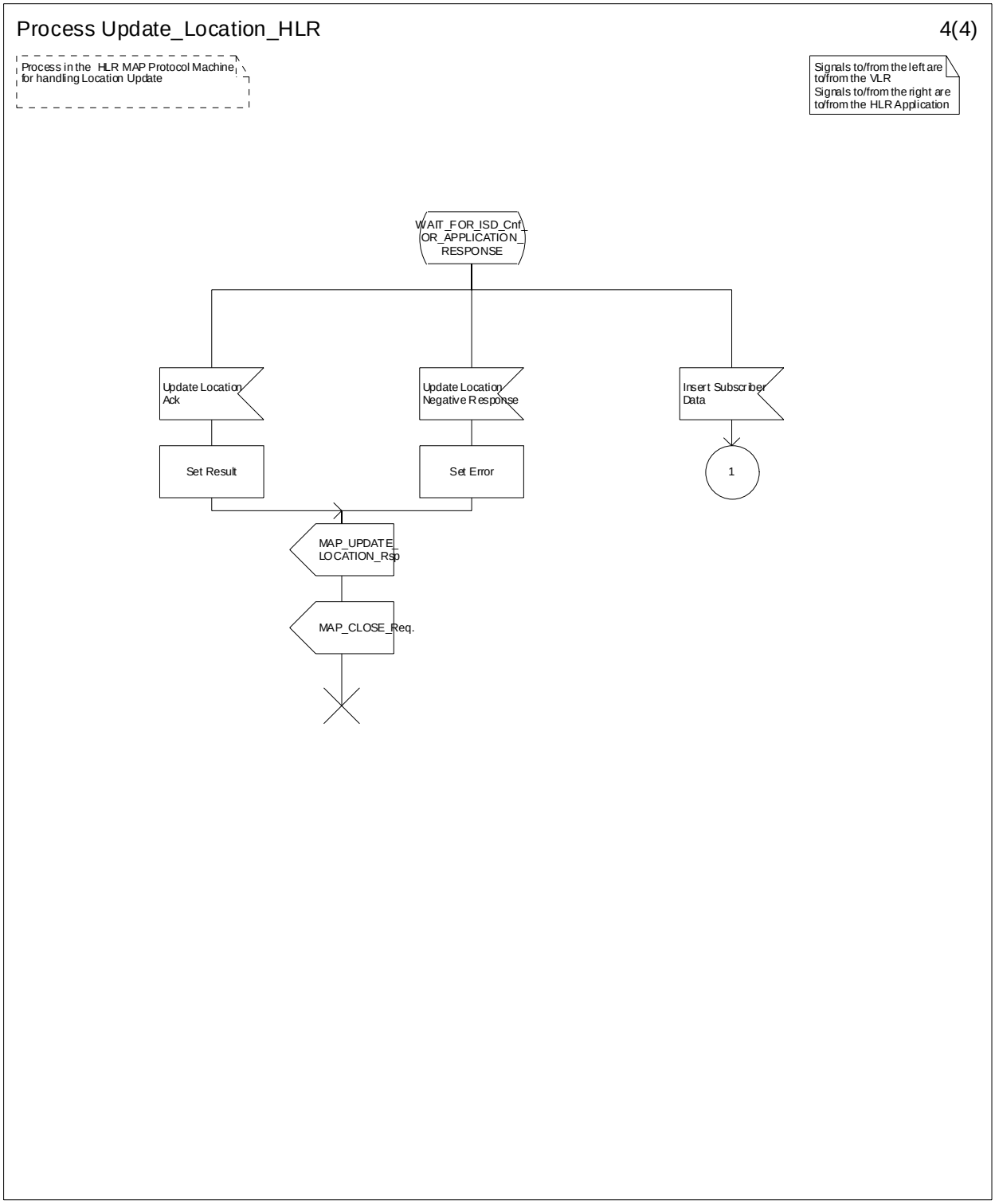


Figure 19.1.1/9 (sheet 4 of 4): Process Update_Location_HLR

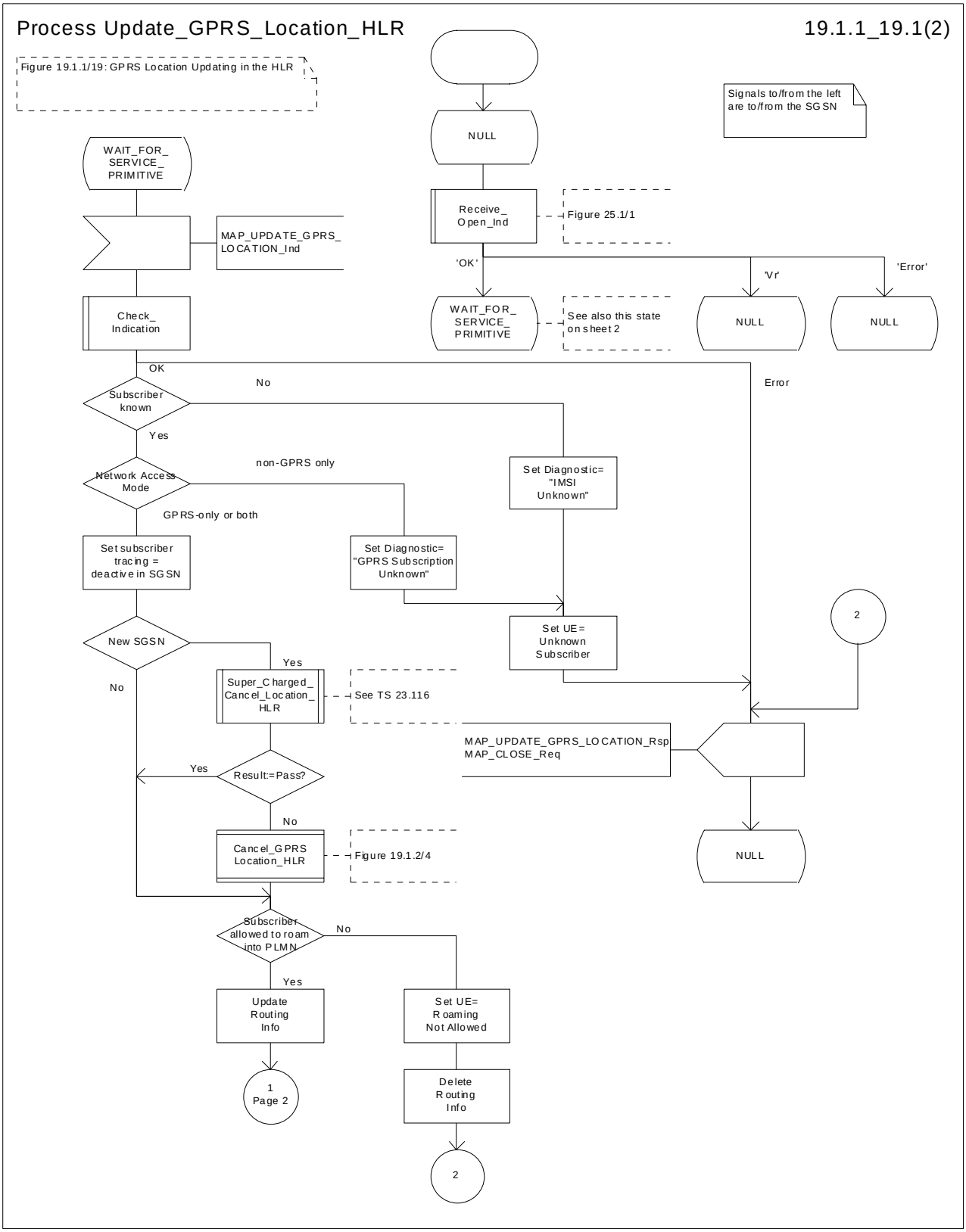


Figure 19.1.1/19 (sheet 1 of 2): Process Update_GPRS_Location_HLR

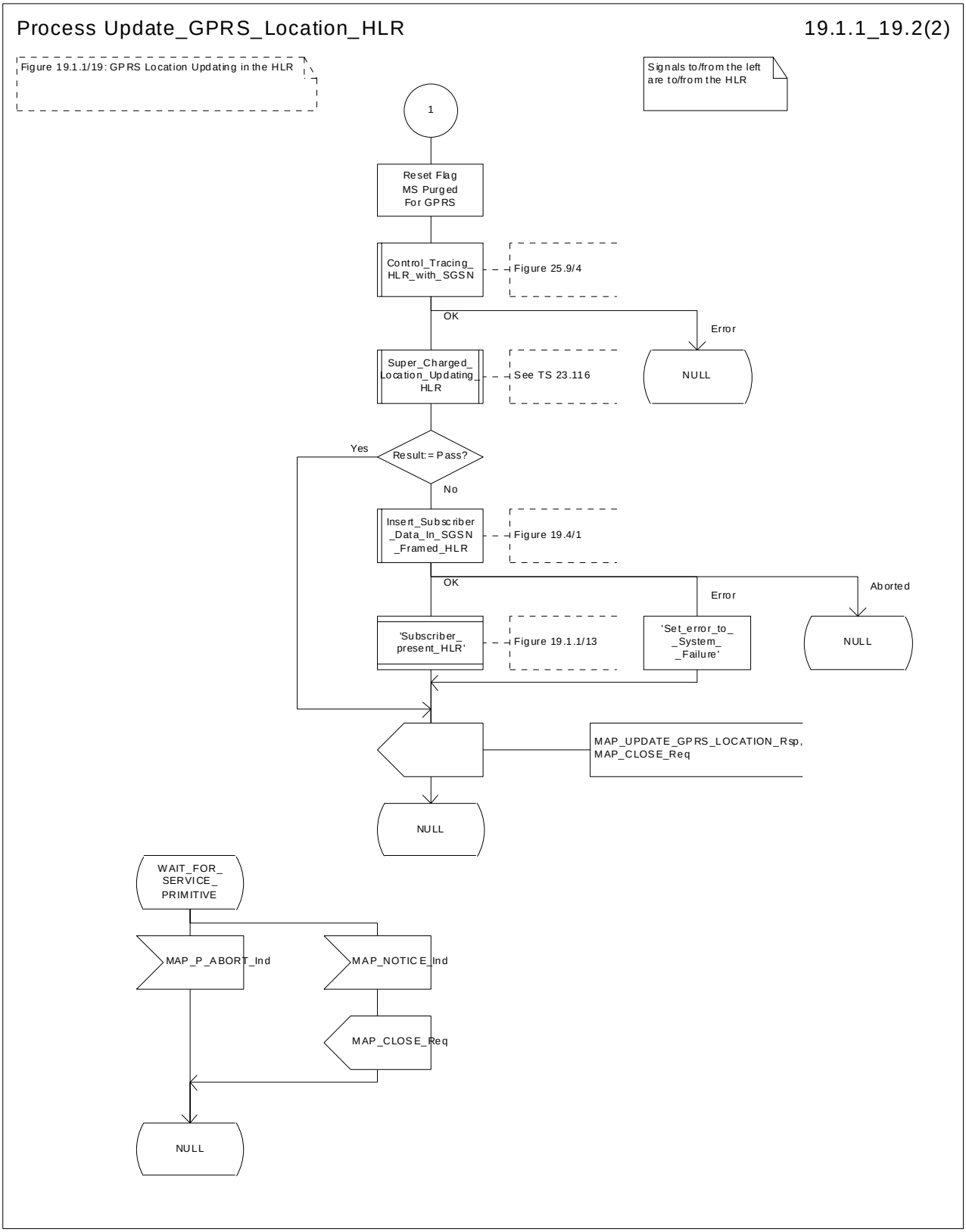


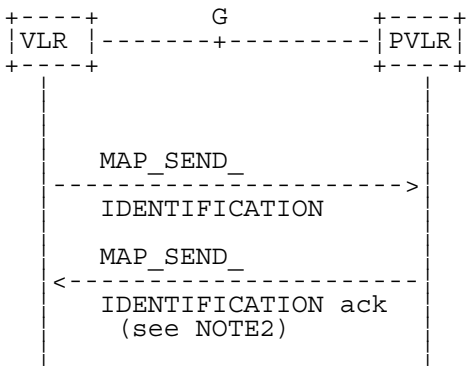
Figure 19.1.1/19 (sheet 2 of 2): Process Update_GPRS_Location_HLR

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 clause 19.1.1.3), using the service described in clause 8.1.4.

If the version negotiation between R99 VLR and pre-R99 PVLR leads to the MAP version 1 or 2, the VLR shall request authentication sets from the HLR.



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 clause 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 clause 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.

If the VLR has indicated that segmentation is prohibited then the PVLR sends a MAP_SEND_IDENTIFICATION response to the VLR by means of the TC-RESULT-L service and terminates the dialogue towards the VLR by a MAP_CLOSE request with parameter Release Method indicating Normal Release.

If the VLR has not indicated that segmentation is prohibited then the PVLR sends a MAP_SEND_IDENTIFICATION response to the VLR by means of the TC-RESULT-L service, followed either by a MAP_DELIMITER if more authentication sets are to be returned, or by a MAP_CLOSE request with parameter Release Method indicating Normal Release.

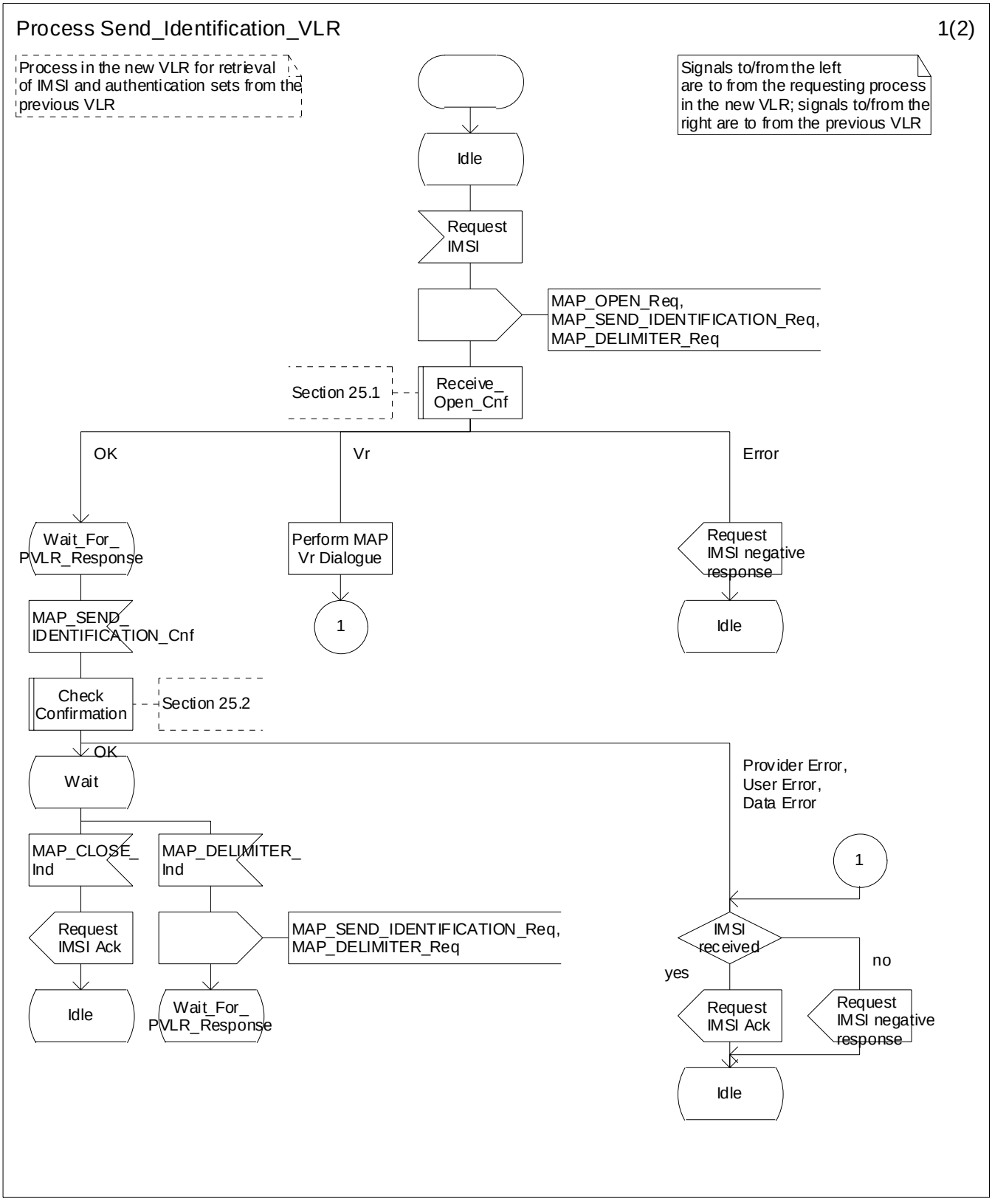


Figure 19.1.1/XX (sheet 1 of 2): Process Send_Identification_VLR

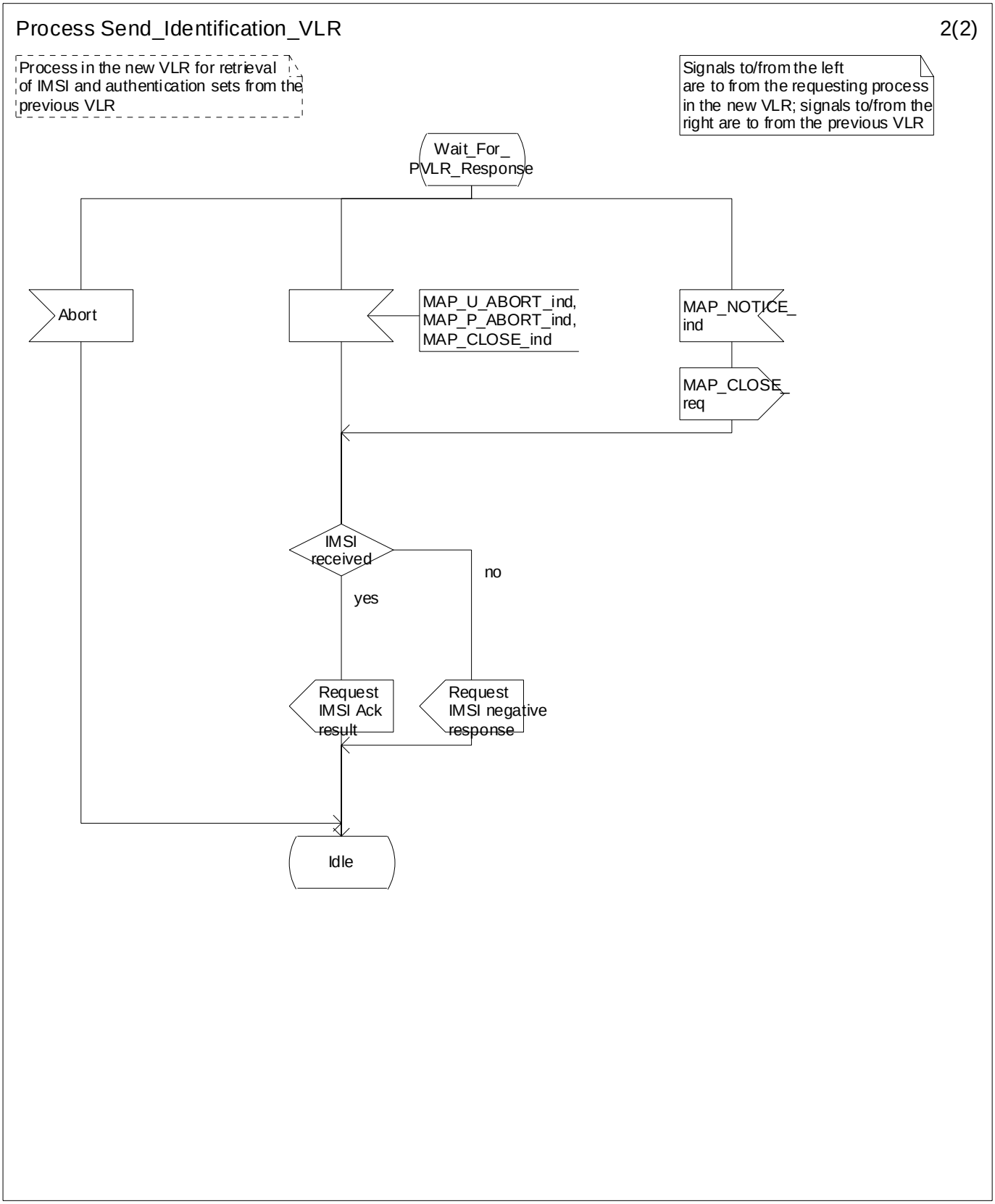


Figure 19.1.1/XX (sheet 2 of 2): Process Send_Identification_VLR

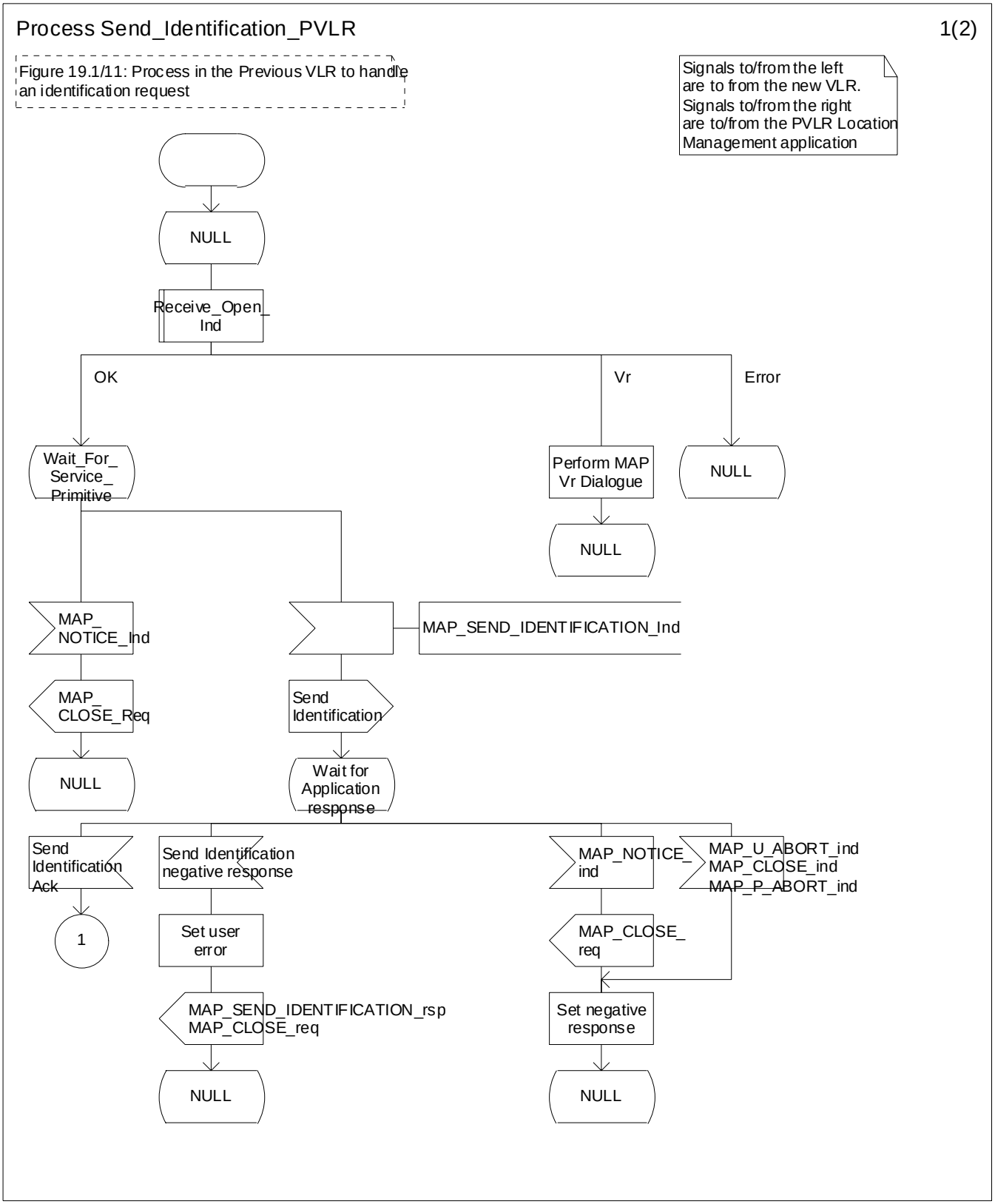


Figure 19.1.1/XX (sheet 1 of 2): Process Send_Identification_PVLR

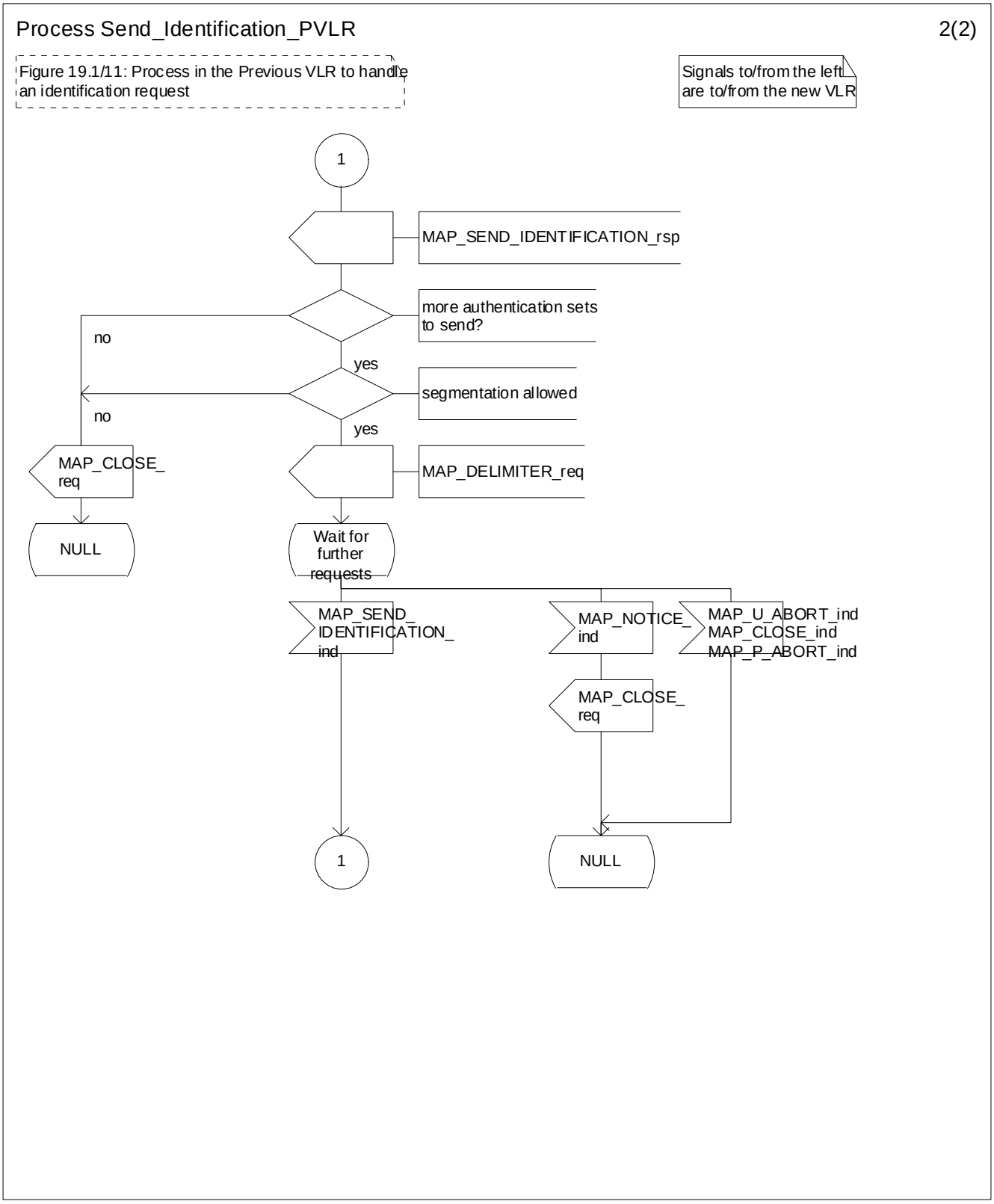


Figure 19.1.1/XX (sheet 2 of 2): Process Send_Identification_PVLR

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 clause 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 1: This process will be performed independent from the calling process, no coordination is required.

NOTE 2: The procedure Notify_gsmSCF is specified in 3GPP TS 23.078.

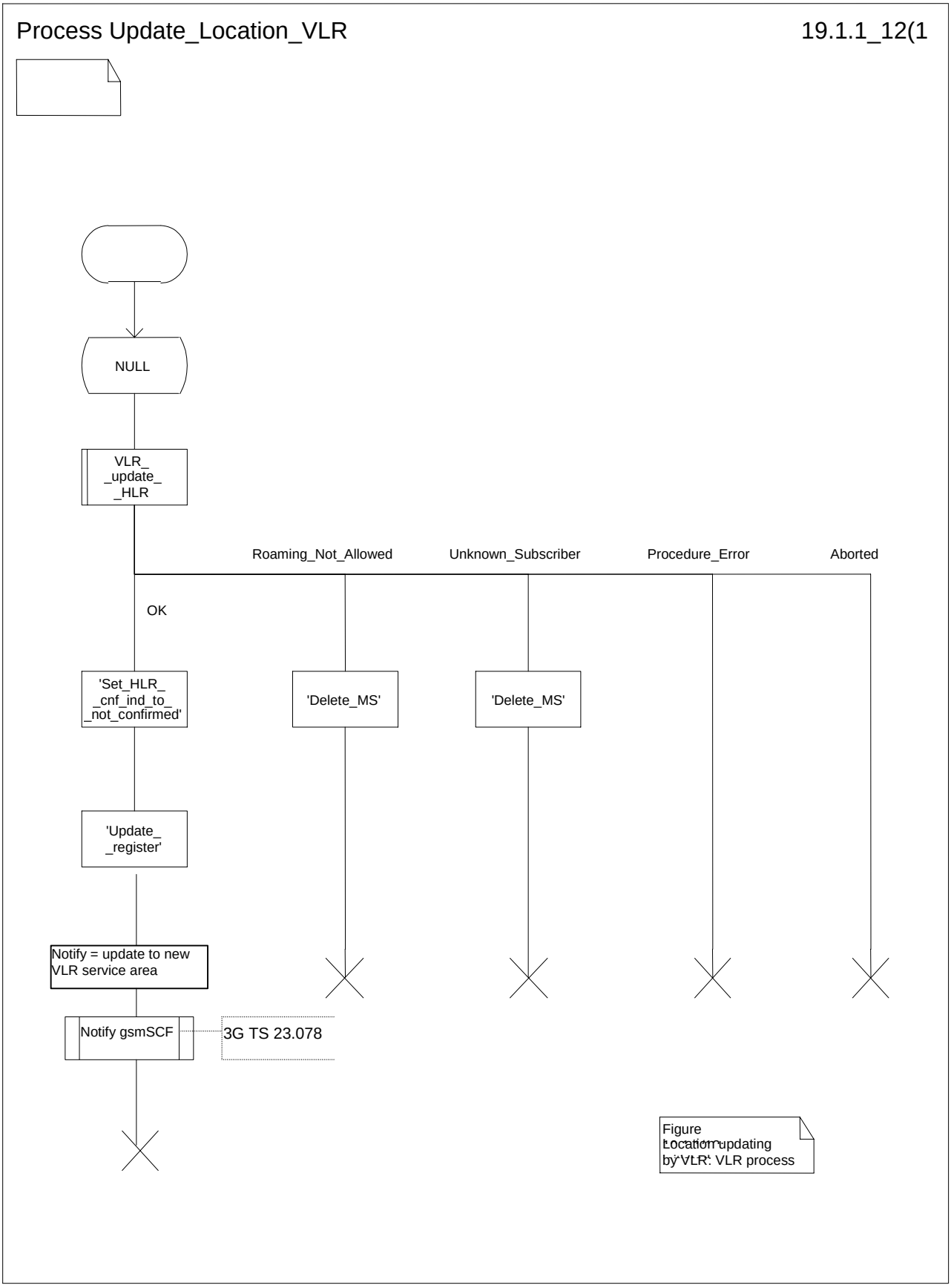


Figure 19.1.1/12: Process UL_VLR

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	clause 25.1;
Insert_Subscriber_Data_SGSN	clause 25.7;
Activate_Tracing_SGSN	clause 25.9;

Sheet 2: The procedure Check_User_Error_In_Serving_Network_Entity is specific to Super-Charger; it is specified in 3GPP TS 23.116 [110].

The location updating process

The MAP process receives an « Update HLR request » from the relevant process in the SGSN (see 3GPP TS 23.060[104]) 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 Unknown HLR 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 clause 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 clause 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 clause 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 3GPP TS 23.060[104]).
  - 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.

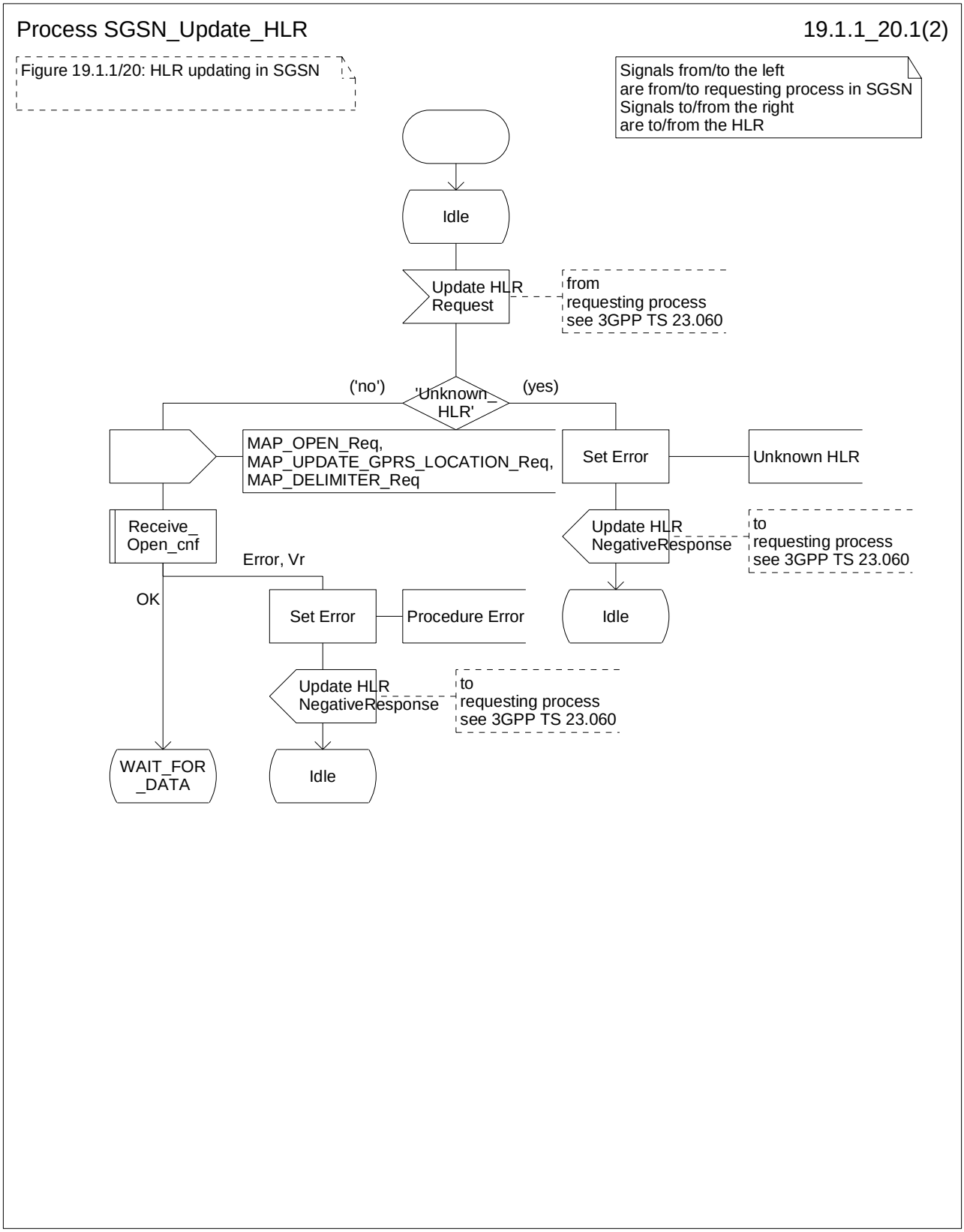


Figure 19.1.1/20 (sheet 1 of 2): Process SGSN_Update_HLR

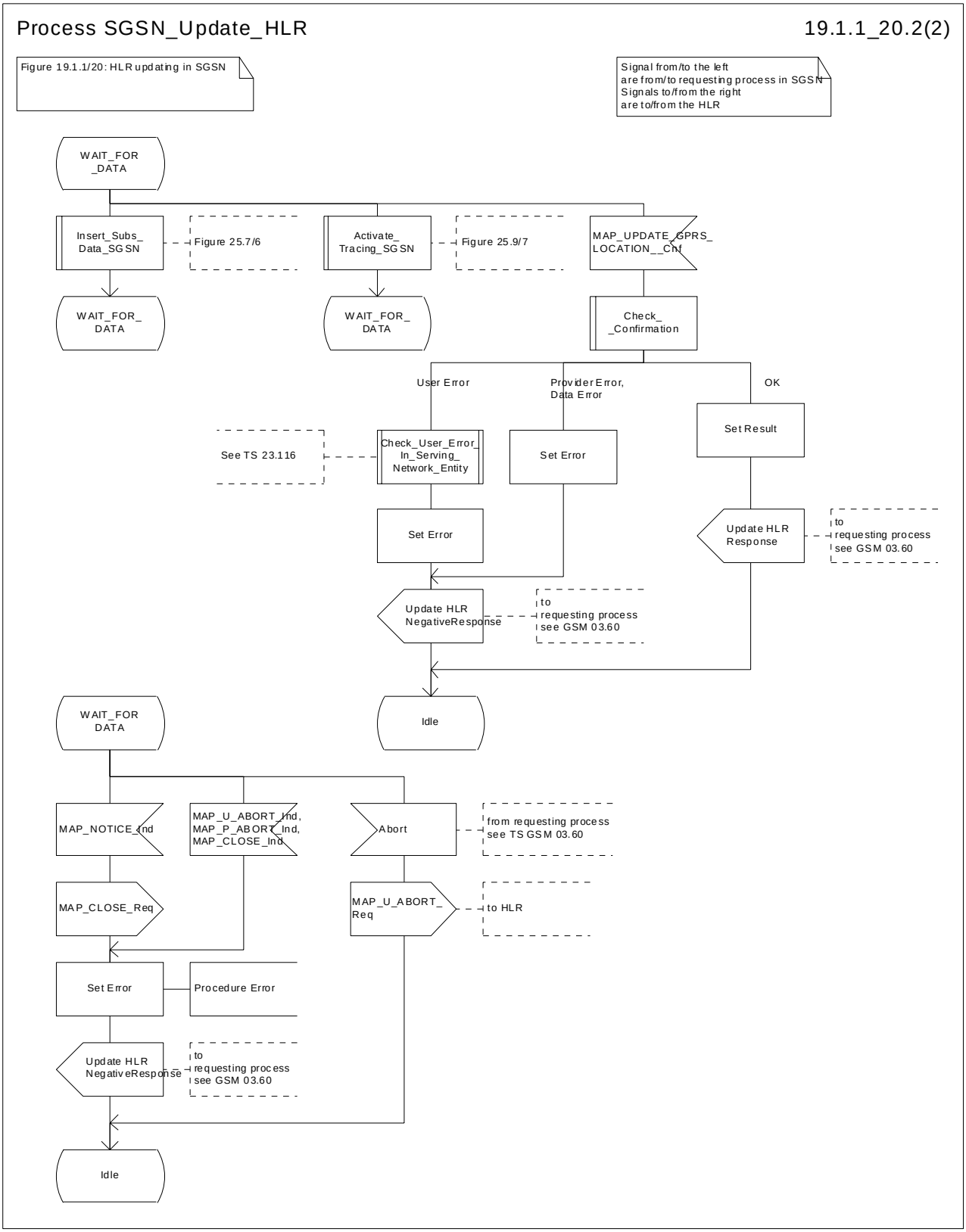


Figure 19.1.1/20 (sheet 2 of 2): Process SGSN_Update_HLR

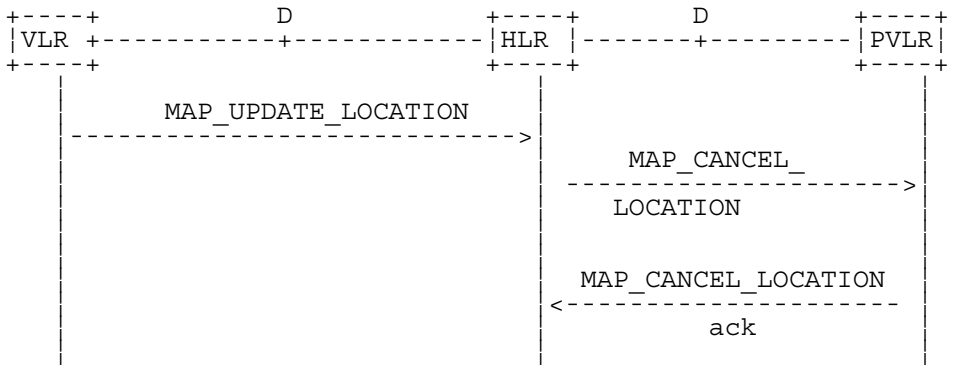
19.1.2 Location Cancellation

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 clause 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

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 19.1.2/2 and 19.1.2/4), 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 does not understand it, this parameter shall be ignored.

19.1.2.3 Detailed procedure in the VLR

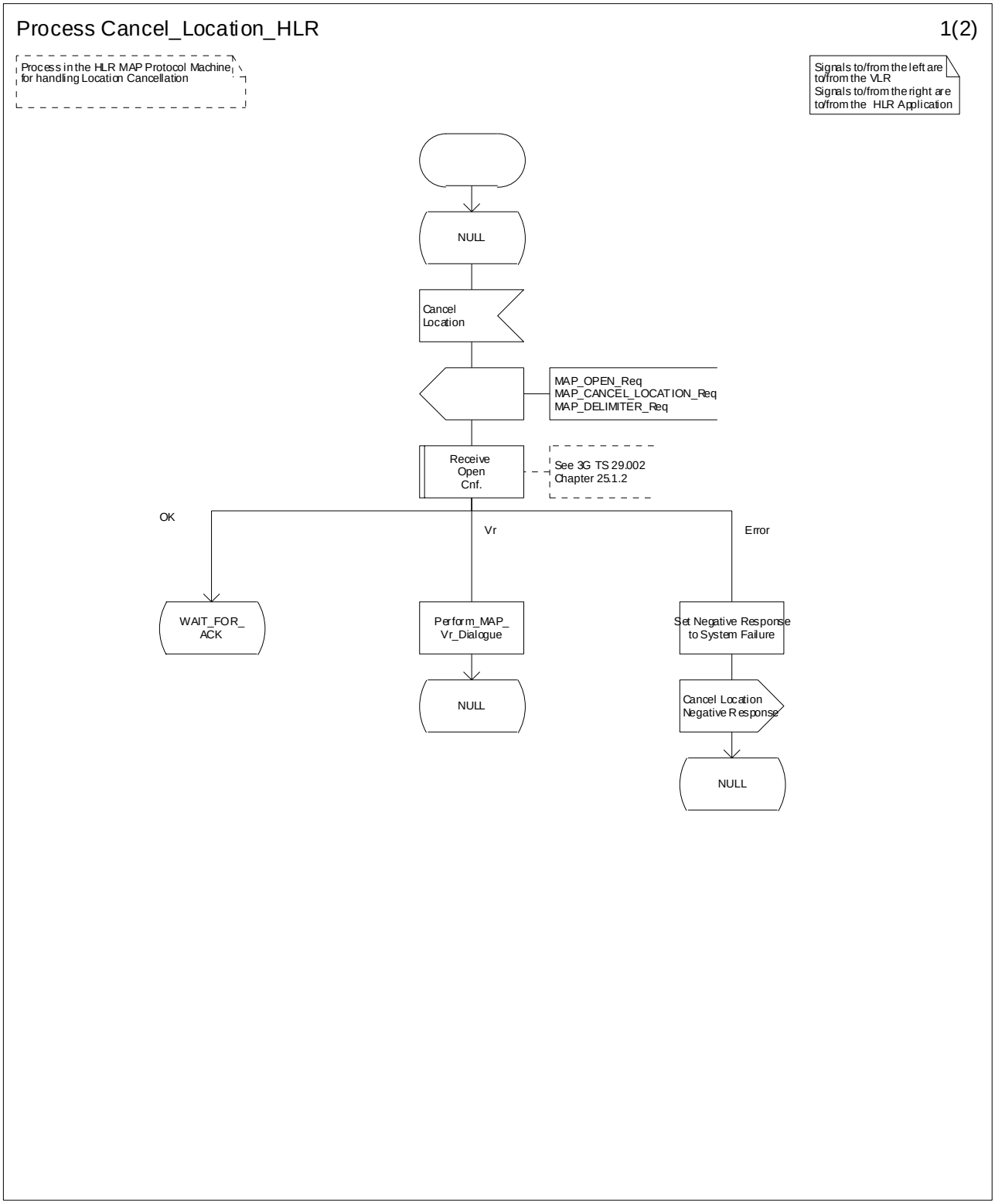


Figure 19.1.2/2 (Sheet 1 of 2): Process Cancel_Location_HLR

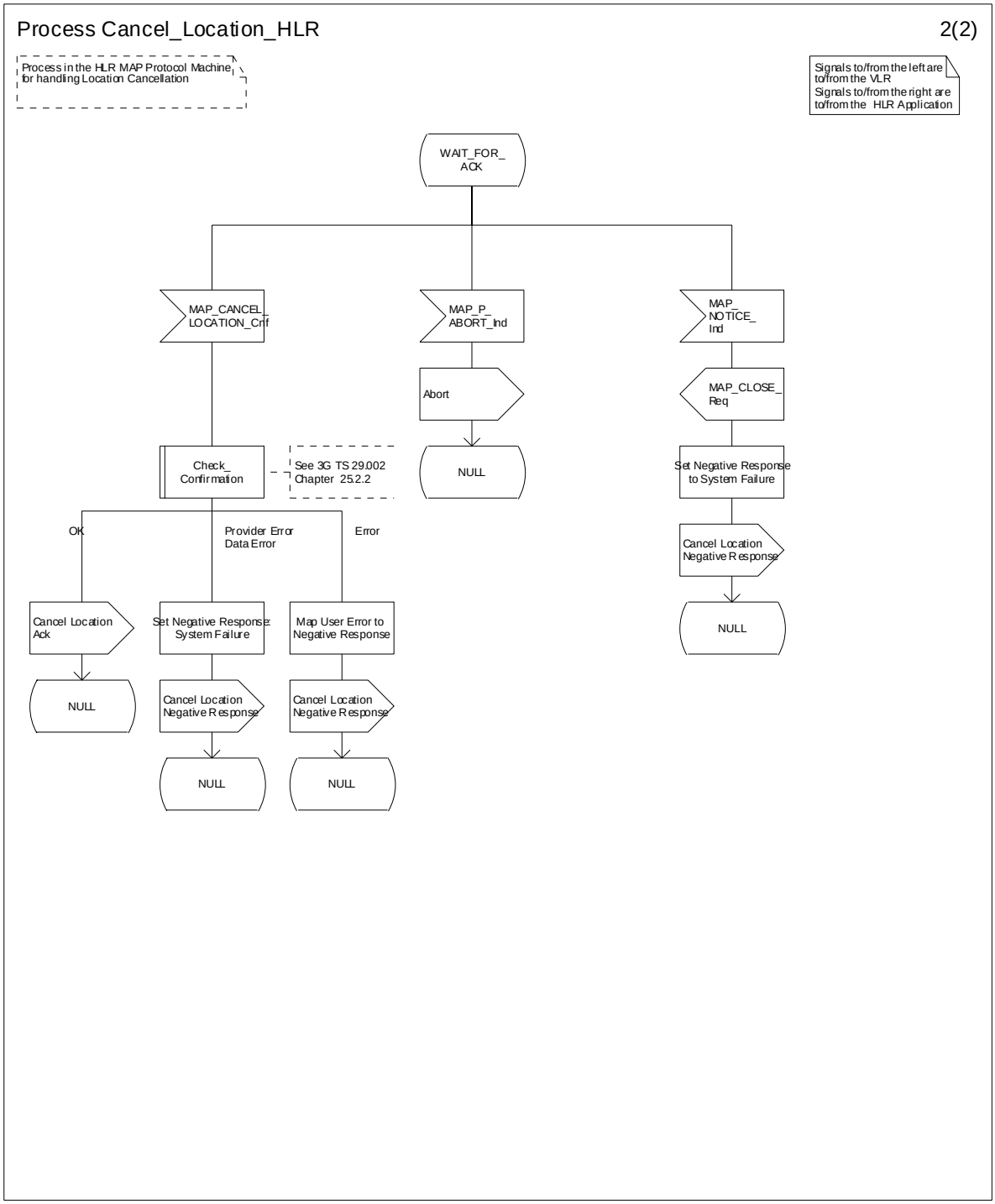


Figure 19.1.2/2 (Sheet 2 of 2): Process Cancel_Location_HLR

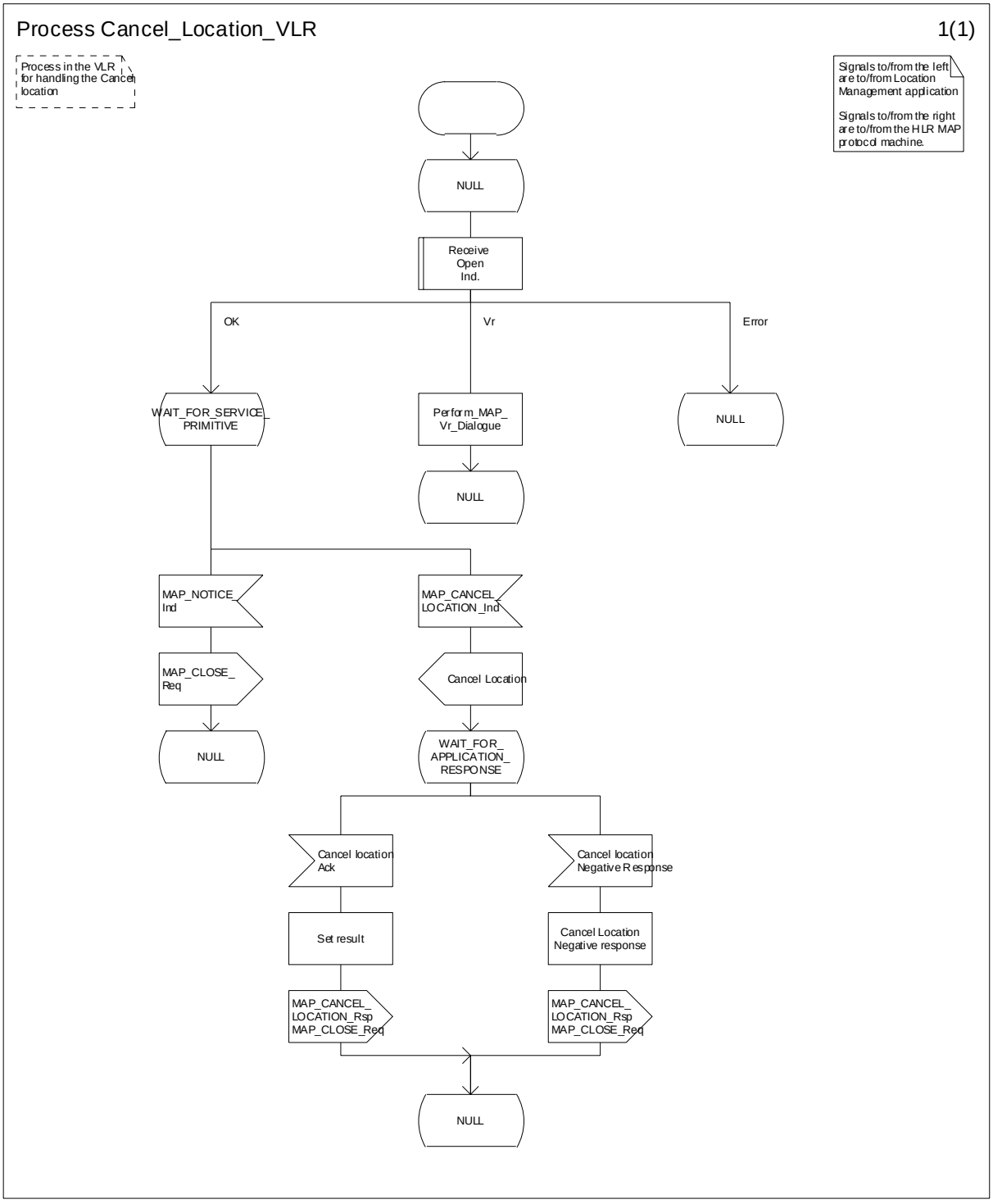


Figure 19.1.2/3: Process Cancel_Location_VLR

19.1.2.4 Detailed procedure in the SGSN

Opening of the dialogue is described in the macro Receive_Open_Ind in clause 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 clause 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.

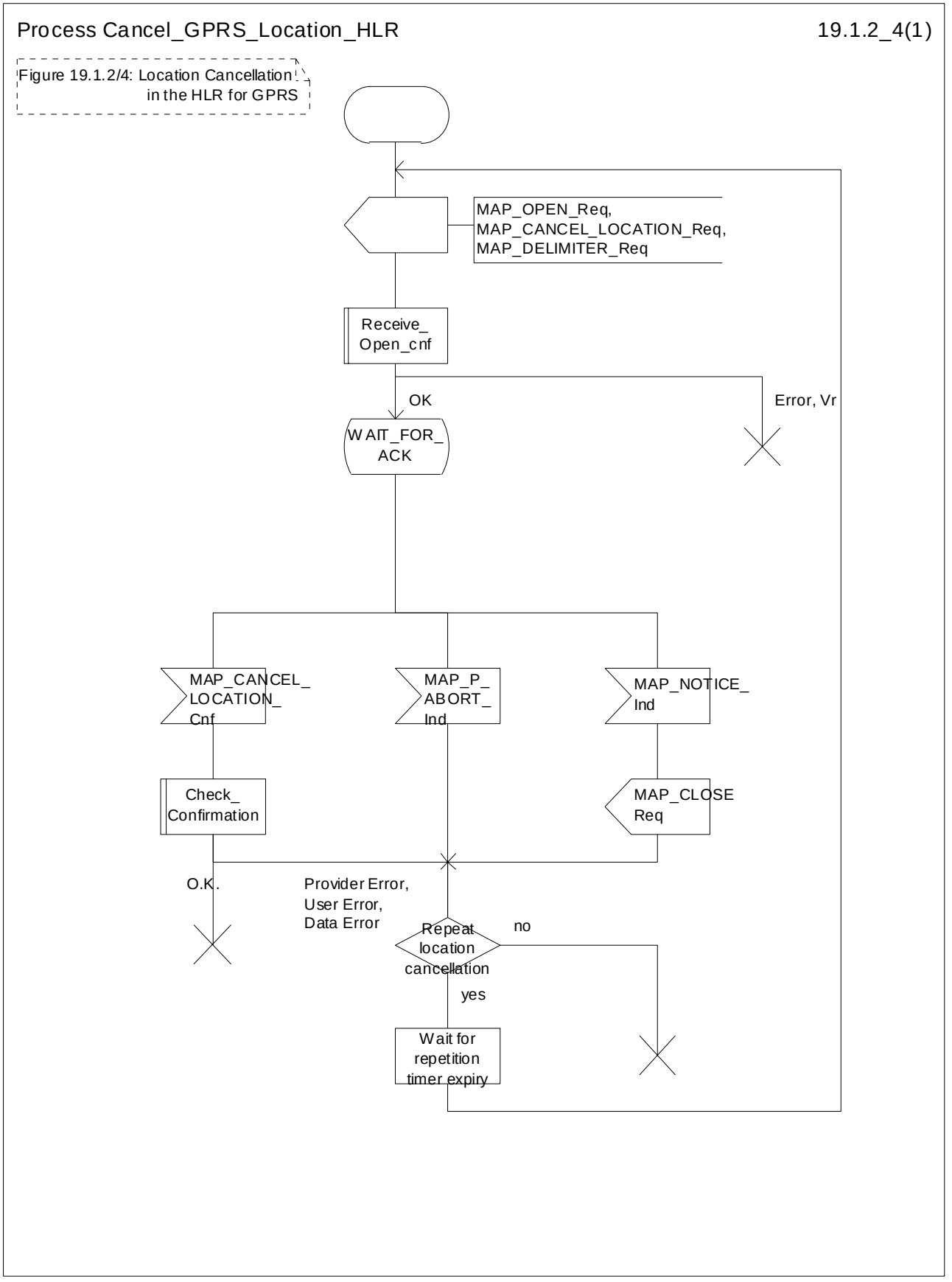


Figure 19.1.2/4: Process Cancel_GPRS_Location_HLR

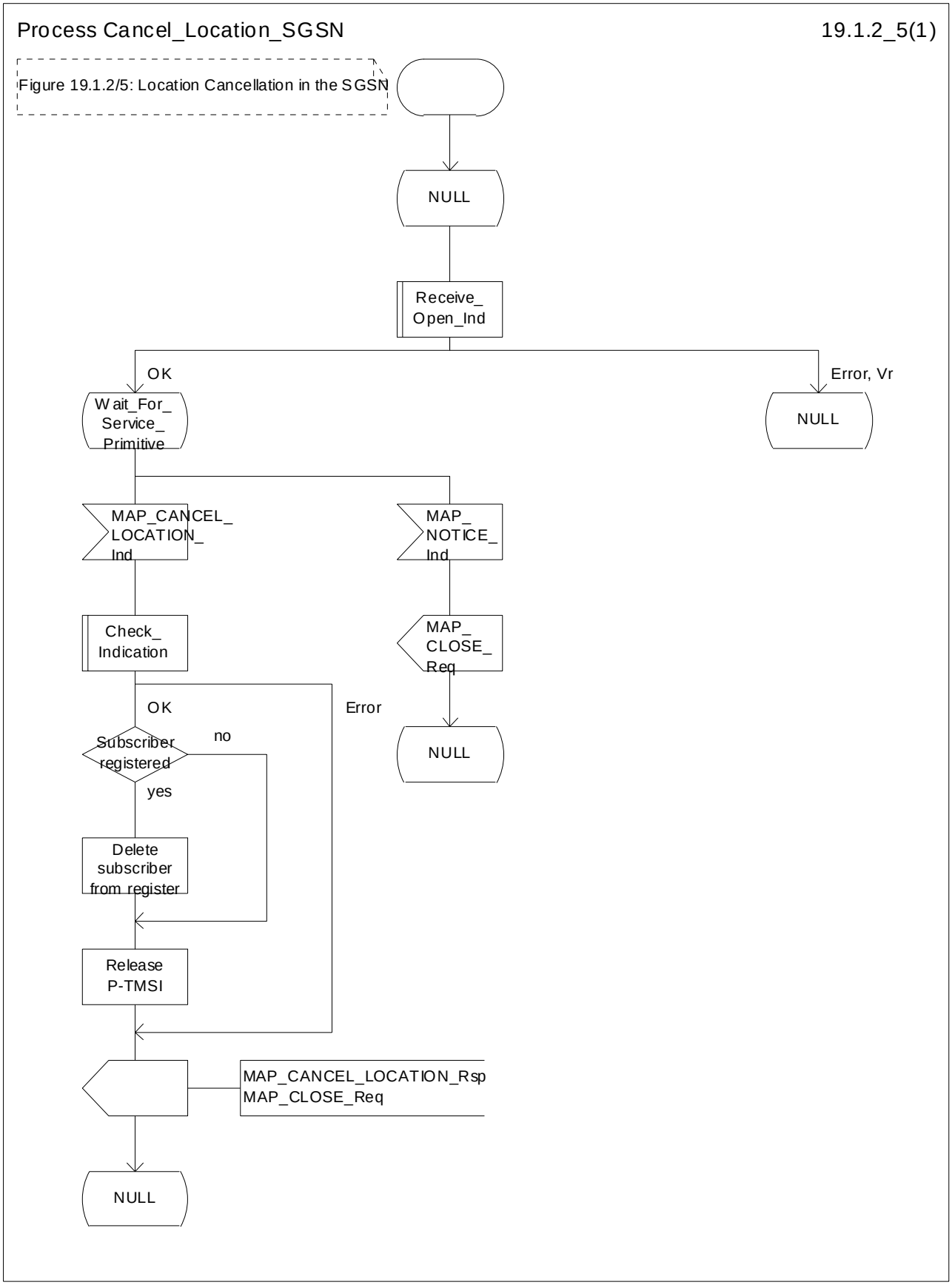


Figure 19.1.2/5: Process Cancel_Location_SGSN

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 clause 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 19.1.4/1and 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.

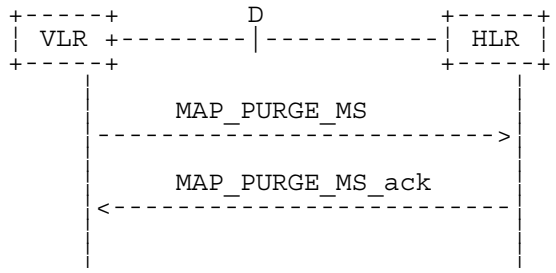


Figure 19.1.4/1: MAP-D Interface and services for MAP_PURGE_MS

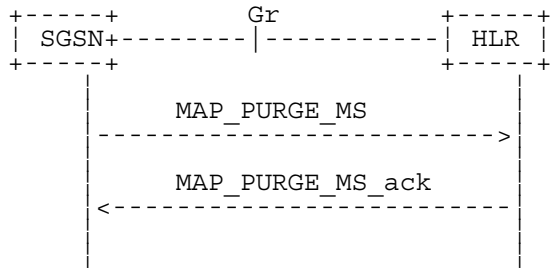


Figure 19.1.4/5: Gr Interface and services for MAP_PURGE_MS

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 clause 25.1;

Sheet 1: The procedure Purge_MS_In_Serving_Network_Entity is specific to Super-Charger; it is specified in 3GPP 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.

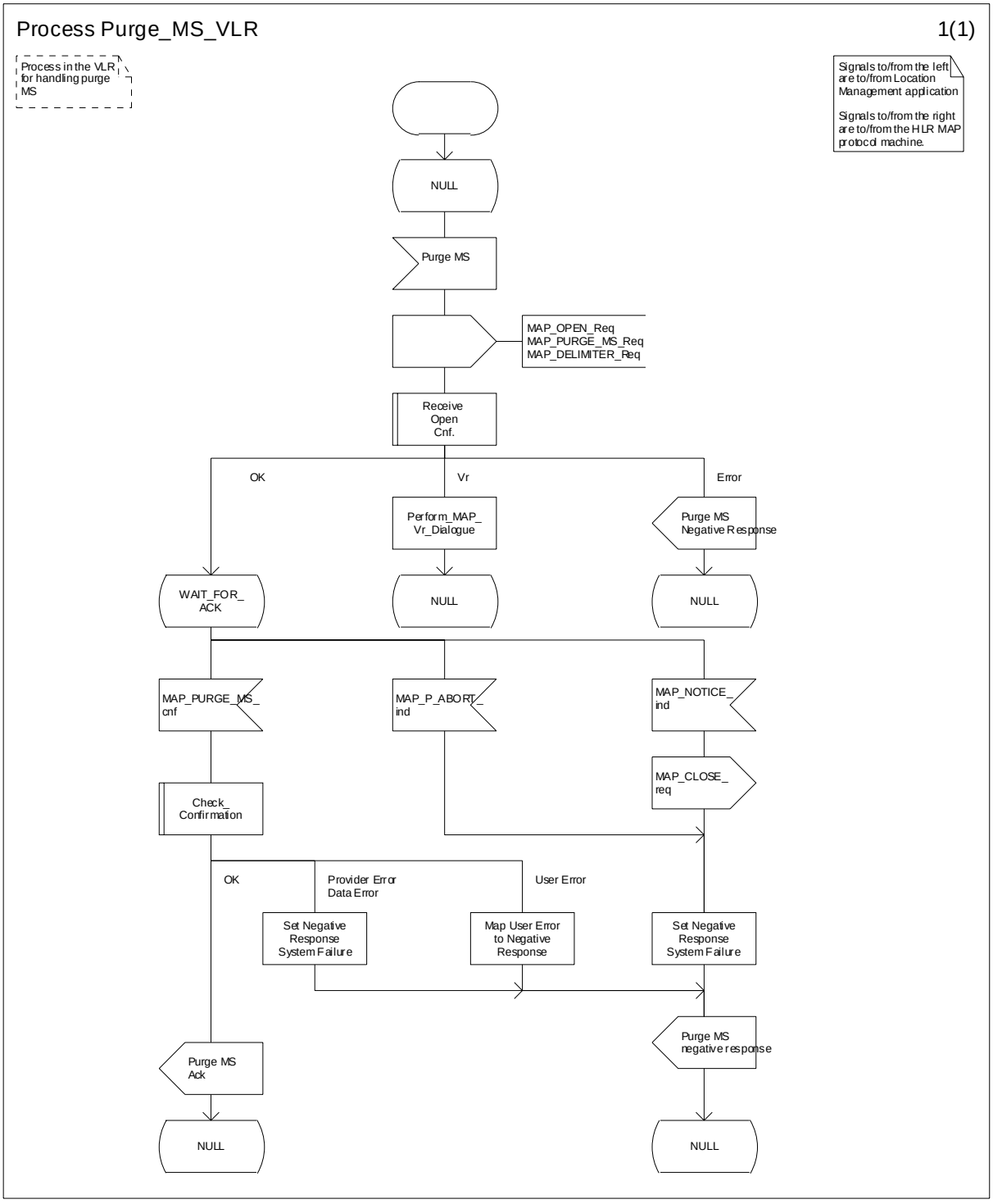


Figure 19.1.4/2: Process Purge_MS_VLR

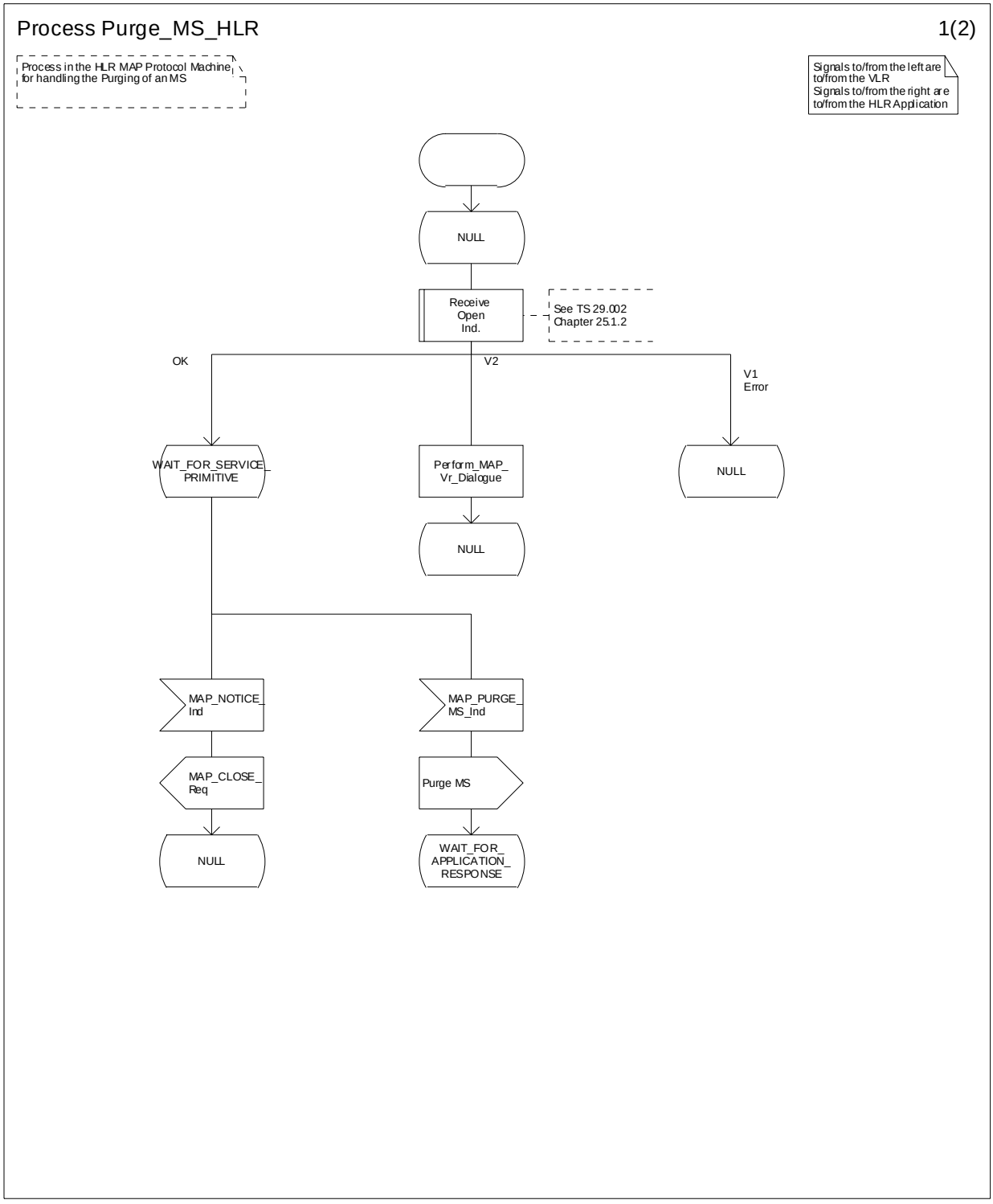


Figure 19.1.4/3 (Sheet 1 of 2): Process Purge_MS_HLR

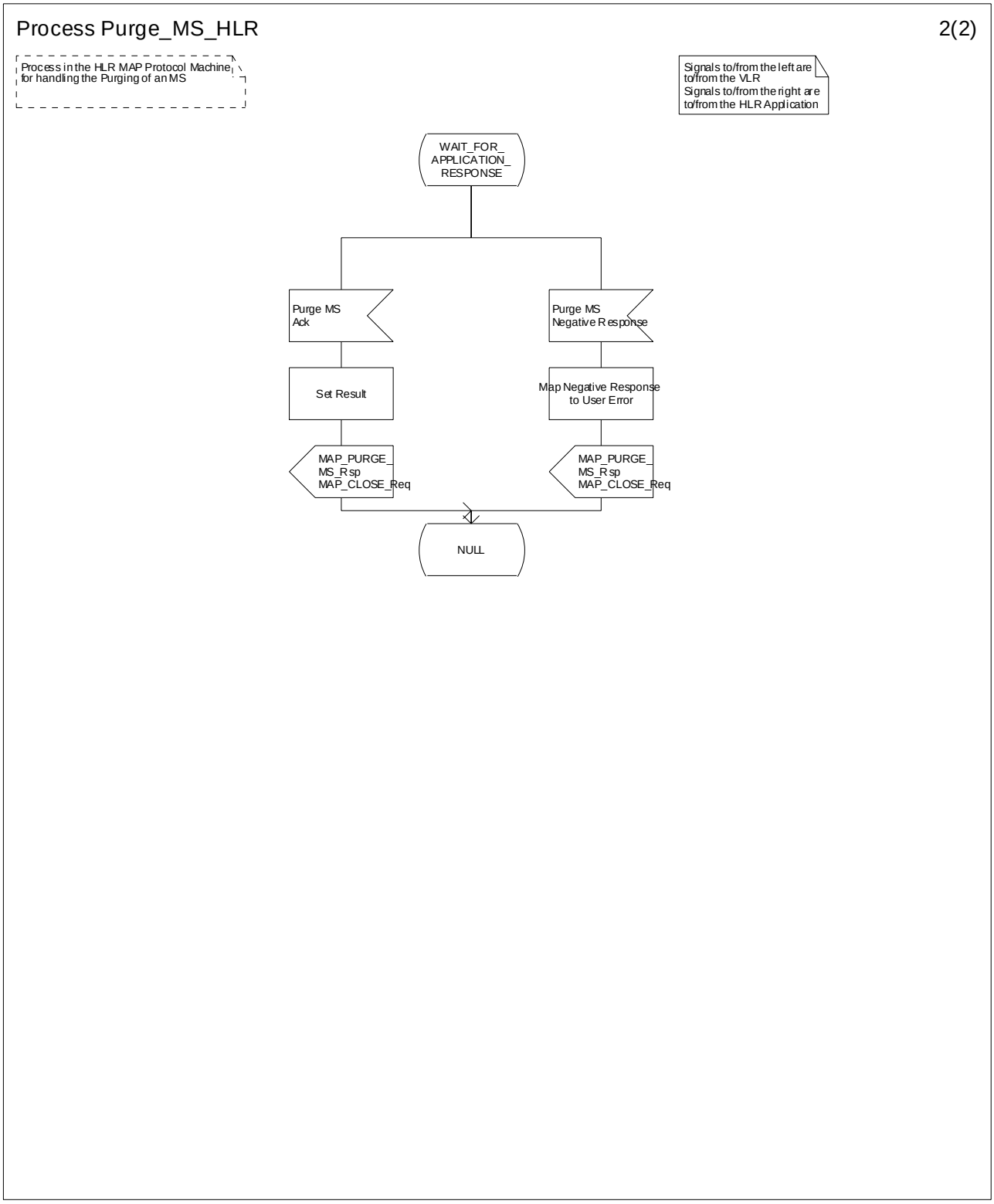


Figure 19.1.4/3 (Sheet 2 of 2): Process Purge_MS_HLR

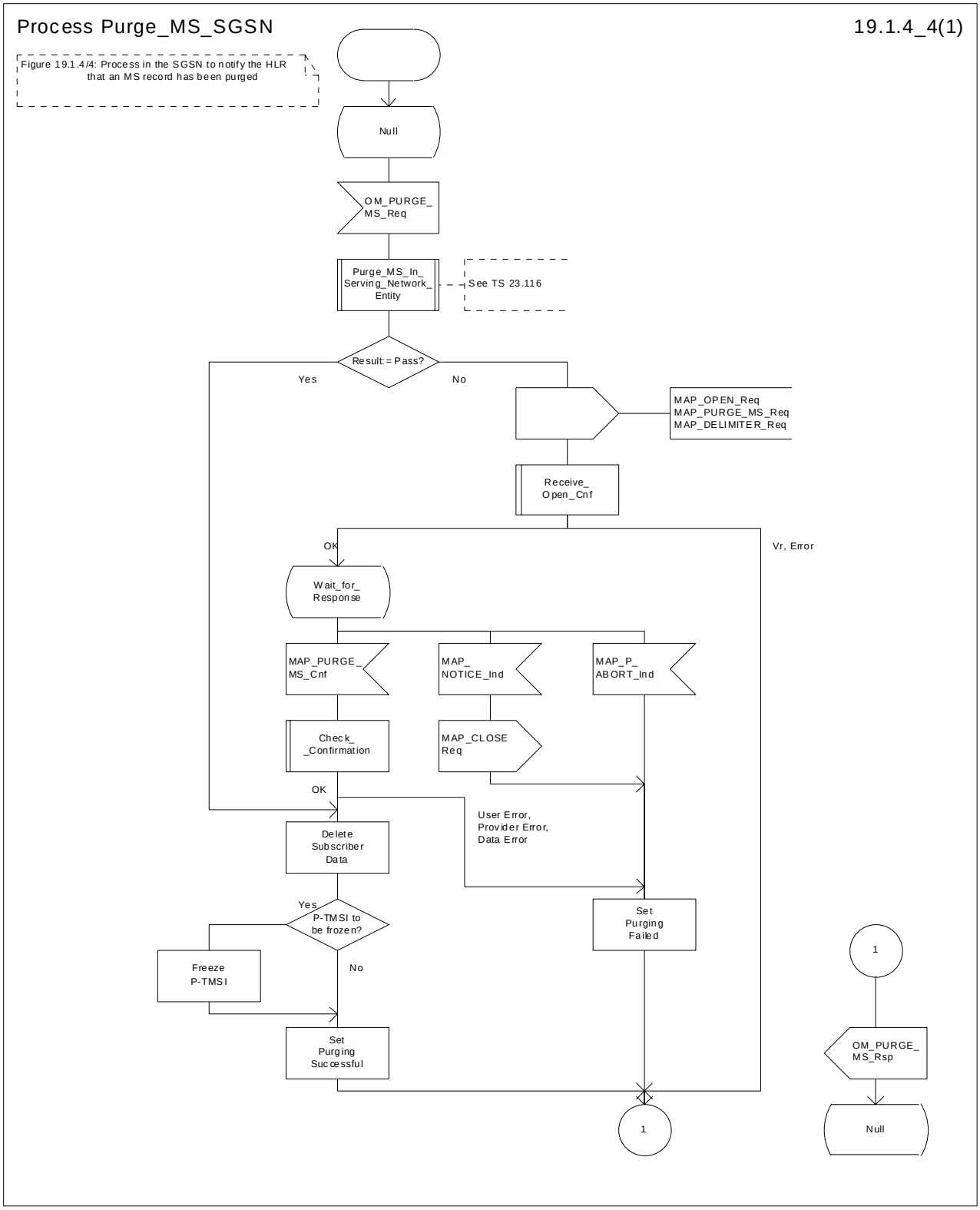


Figure 19.1.4/4: Process Purge_MS_SGSN

19.2 Handover procedure

It should be noted that procedures related to the B-interface have not been updated for Release 99. The B-interface is not fully operational specified. It is strongly recommended not to implement the B-interface as an external interface.

19.2.1 General

The handover or relocation 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:

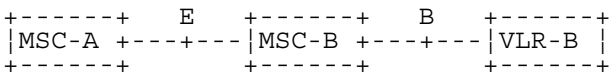
   The call is handed over from the controlling MSC, called MSC-A to another MSC, called MSC-B (figure 19.2/1a).

   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.

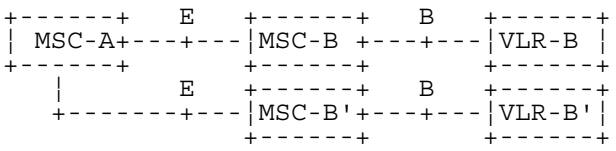
2) Subsequent Inter-MSC handover:

   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 or one or several relocation numbers and to transport BSSAP or RANAP messages.

Minimum applicable MAP AC for intersystem inter-MSC GSM BSS to UTRAN handover shall be MAP handover AC version 3.

Minimum applicable MAP AC for intersystem inter-MSC UTRAN to GSM BSS handover shall be MAP handover AC version 2.

NOTE: If MAP AC version 2 is used, subsequent handover to UTRAN is not possible.

Minimum applicable MAP AC for inter-MSC GSM BSS to GSM BSS handover should be MAP handover AC version 2.

NOTE: If MAP AC version 2 or lower is used, subsequent handover to UTRAN is not possible.

The transported BSSAP or RANAP 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 or RANAP 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 or RANAP messages via the MAP protocol.

For detailed interworking between the A-interface and MAP procedures or the Iu-interface and MAP procedures, see 3GPP TS 23.009 and 3GPP TS 29.010.

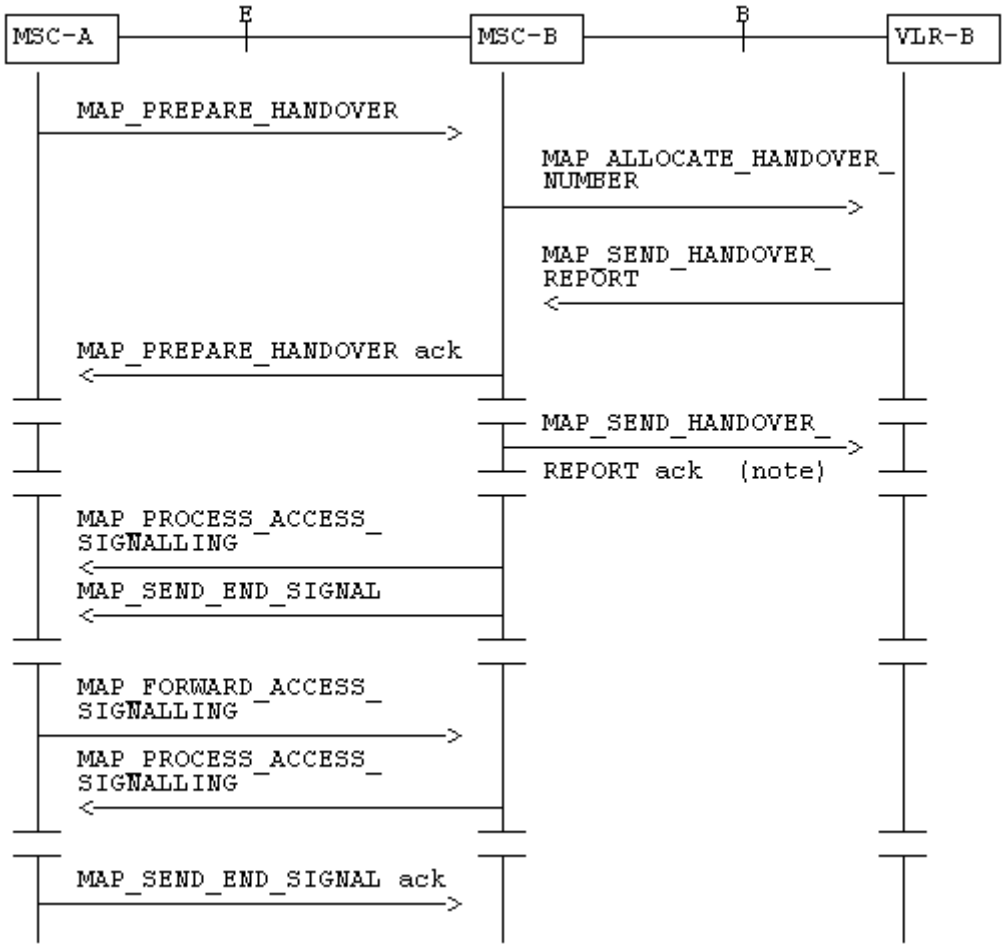


Figure 19.2/2: Example of a successful basic handover procedure to MSC-B

NOTE: This can be sent at any time after the connection between MSC-A and MSC-B is established.

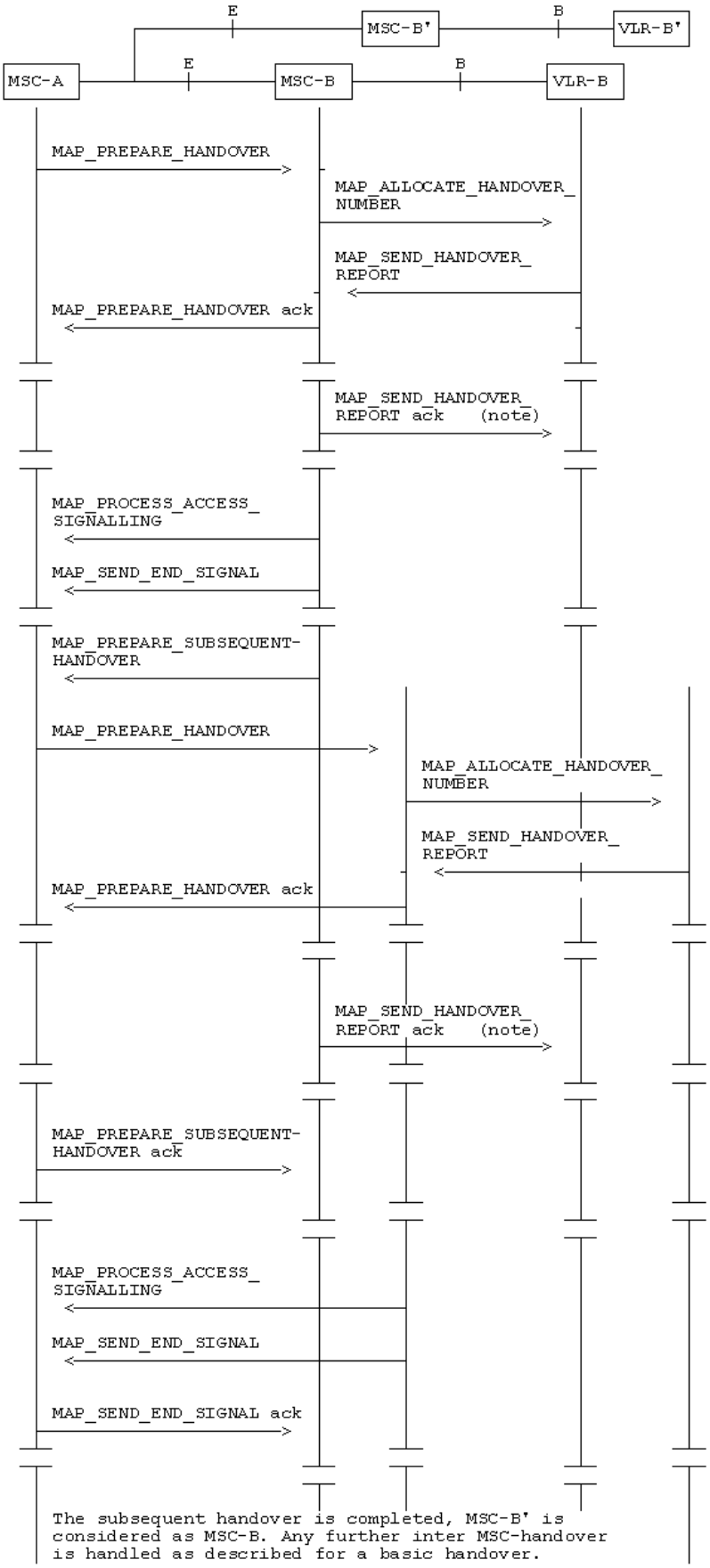


Figure 19.2/3: Example of a handover towards a third MSC

NOTE: This can be sent at any time after the connection between MSC-A and MSC-B is established.

19.2.2 Handover procedure in MSC-A

This clause describes the handover or relocation procedure in MSC-A, including the request for a basic handover or relocation to another MSC (MSC-B), subsequent handover or relocation 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 or relocated to MSC-B, the Handover Control Application in MSC-A requests the MAP application to initiate the MAP_PREPARE_HANDOVER request to MSC-B.

MSC-A opens the dialogue to MSC-B with a MAP_OPEN request containing no user specific parameters and sends a MAP_PREPARE_HANDOVER request. This request shall contain all the information required by MSC-B to allocate the necessary radio resources. In addition, it may optionally contain:

- an indication that a handover number allocation is not required;
- the targetCellId, for compatibility reasons in case of handover or inter-system handover to GSM ;
- the targetRNCId, in case of SRNS relocation or inter-system handover from GSM to UMTS;
- the IMSI;
- UMTS encryption information and UMTS integrity protection information, which are necessary parameters for inter-system handover from GSM to UMTS;
- GSM radio resource information (channel type) shall be included at inter-MSC relocation to prepare for a possible subsequent intra-MSC handover from UMTS to GSM in MSC-B.

The conditions when these parameters shall be included and the processing of them in MSC-B (3G_MSC-B) are described in detail in 3GPP TS 29.010 and 3GPP TS 23.009.

If MSC-B accepts the dialogue, it returns a MAP_PREPARE_HANDOVER confirmation containing a handover number or one or several relocation numbers, unless the request has included the HO-NumberNotRequired parameter, and BSSAP or RANAP information which is forwarded to and handled by the Handover Control Application in MSC-A.

Optionally MSC-A can receive, after a MAP_PREPARE_HANDOVER confirmation, a MAP_PROCESS_ACCESS_SIGNALLING indication containing BSSAP or RANAP 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 or relocation 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 or the Iu-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 or the Iu-interface for call control and mobility management information. Any response received in MSC-B from the A-interface or the Iu-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 or relocation, a number of procedures between MSC-A and BSS-B or RNS-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_HANDOVER request, it will start the procedure of handing or relocating the call over to a third MSC (MSC-B'), or back to the controlling MSC (MSC-A). If the new handover or relocation 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 clause.

The services used are defined in clause 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 or relocation procedure, see 3GPP TS 23.009.

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.

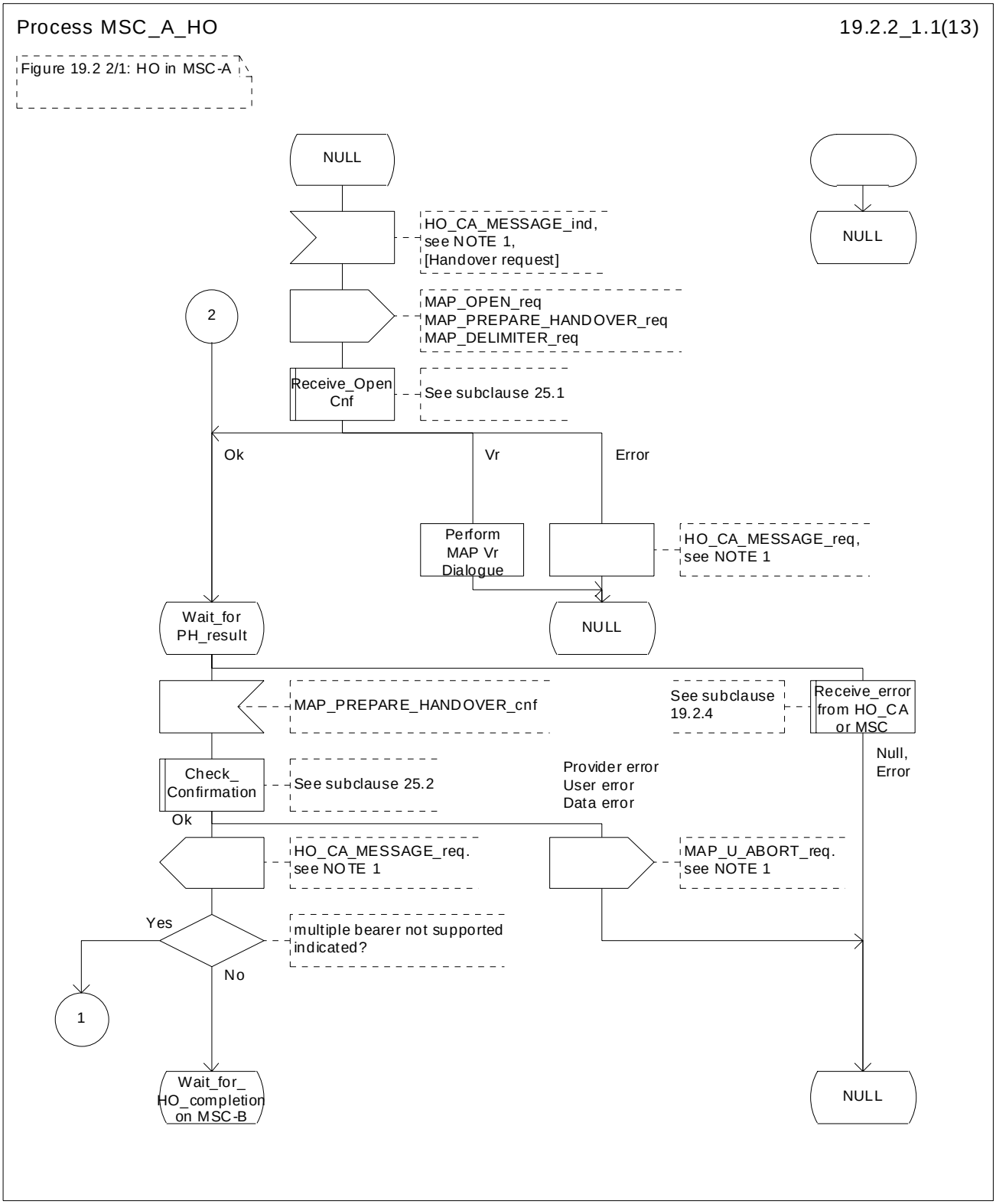


Figure 19.2.2/1 (sheet 1 of 13): Process MSC_A_HO

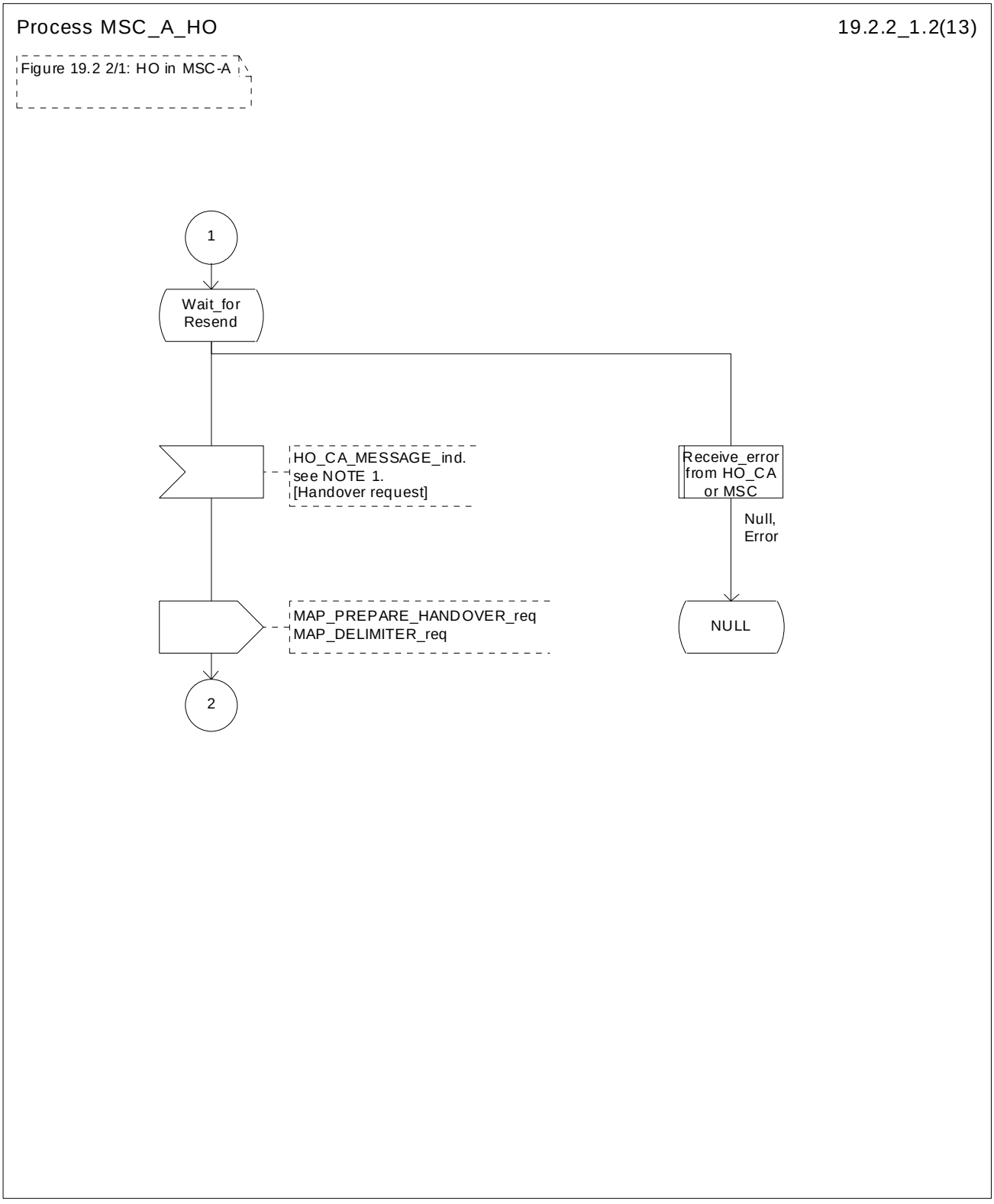


Figure 19.2.2/1 (sheet 2 of 13): Process MSC_A_HO

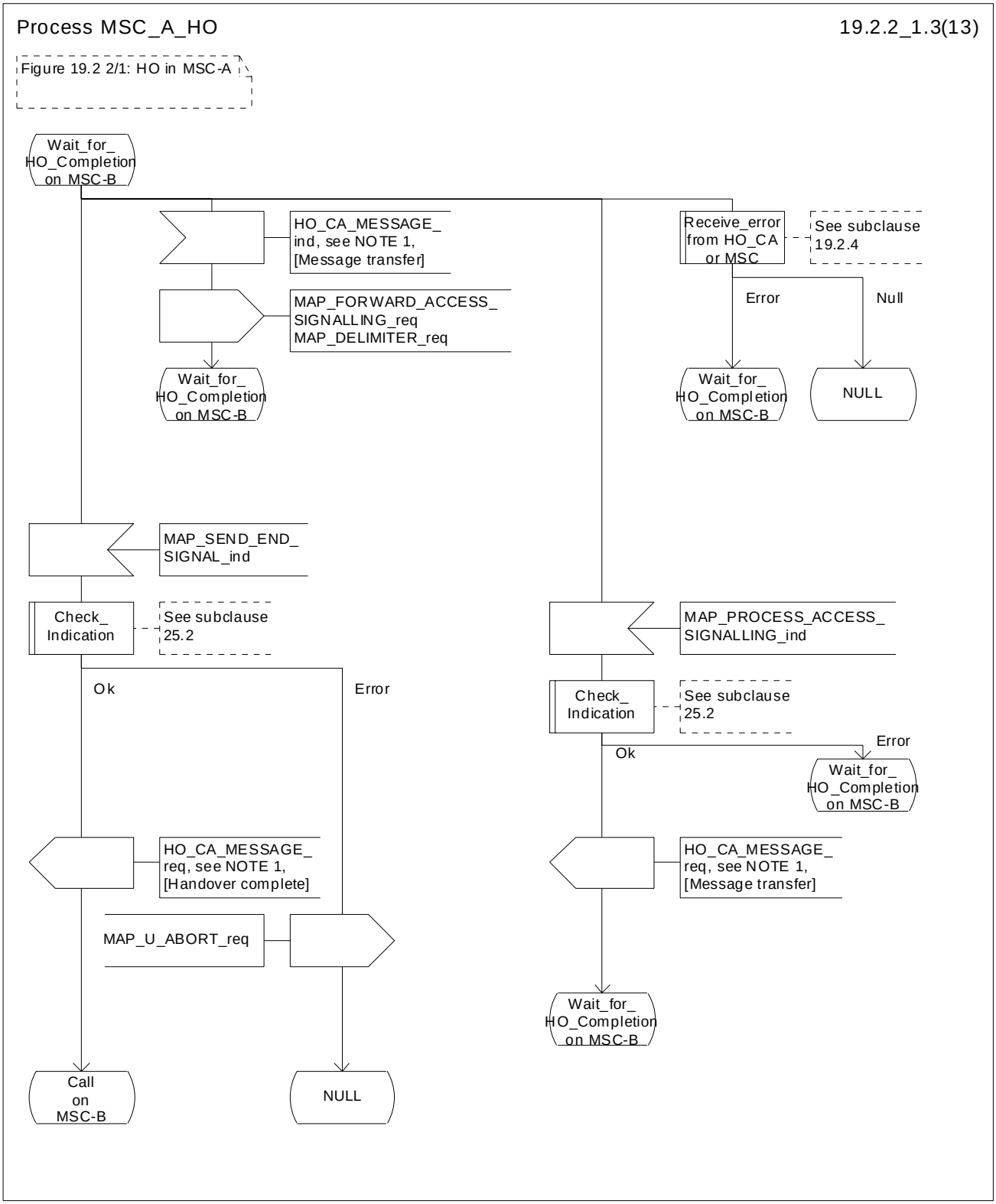


Figure 19.2.2/1 (sheet 3 of 13): Process MSC_A_HO

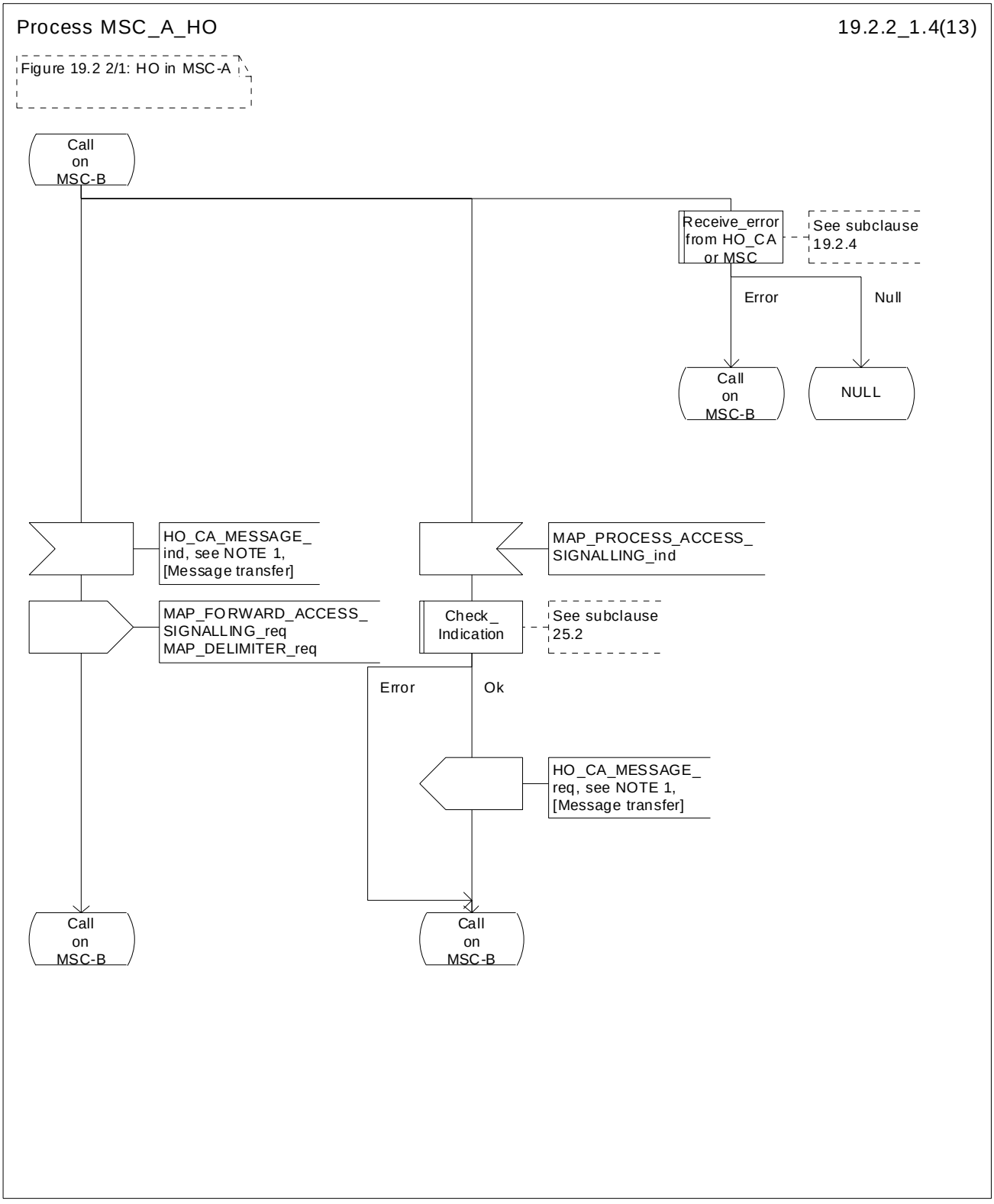


Figure 19.2.2/1 (sheet 4 of 13): Process MSC_A_HO

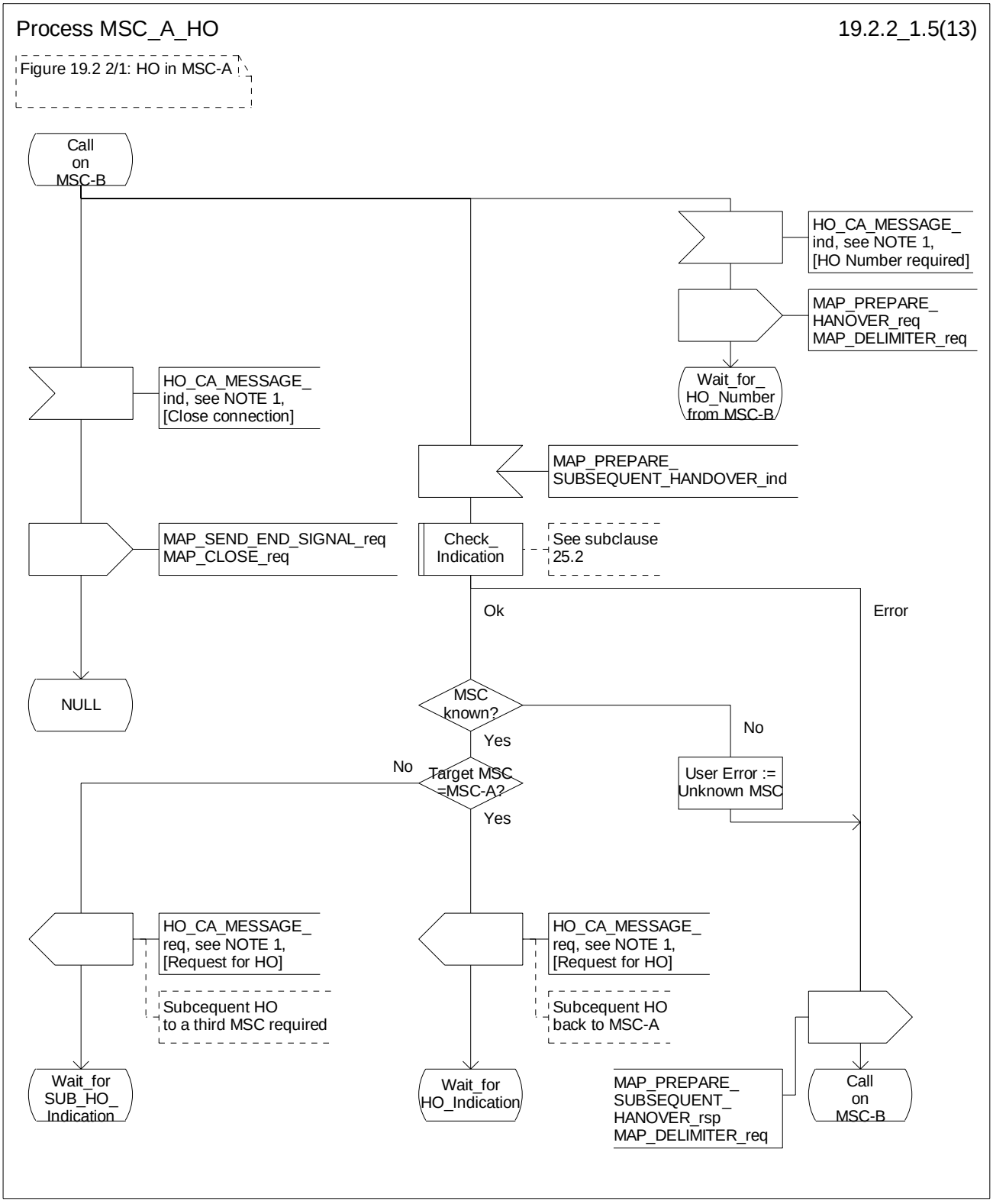


Figure 19.2.2/1 (sheet 5 of 13): Process MSC_A_HO

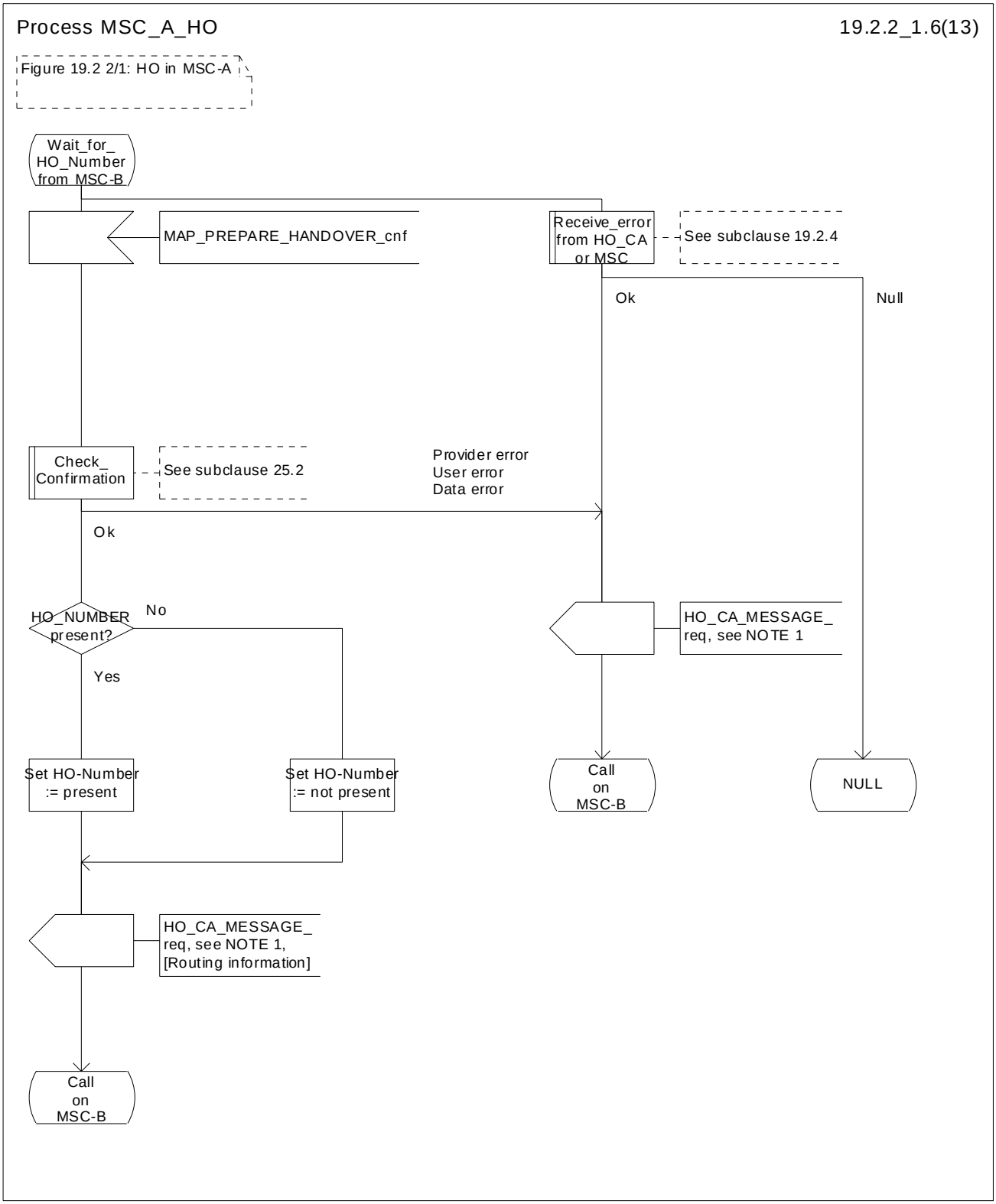


Figure 19.2.2/1 (sheet 6 of 13): Process MSC_A_HO

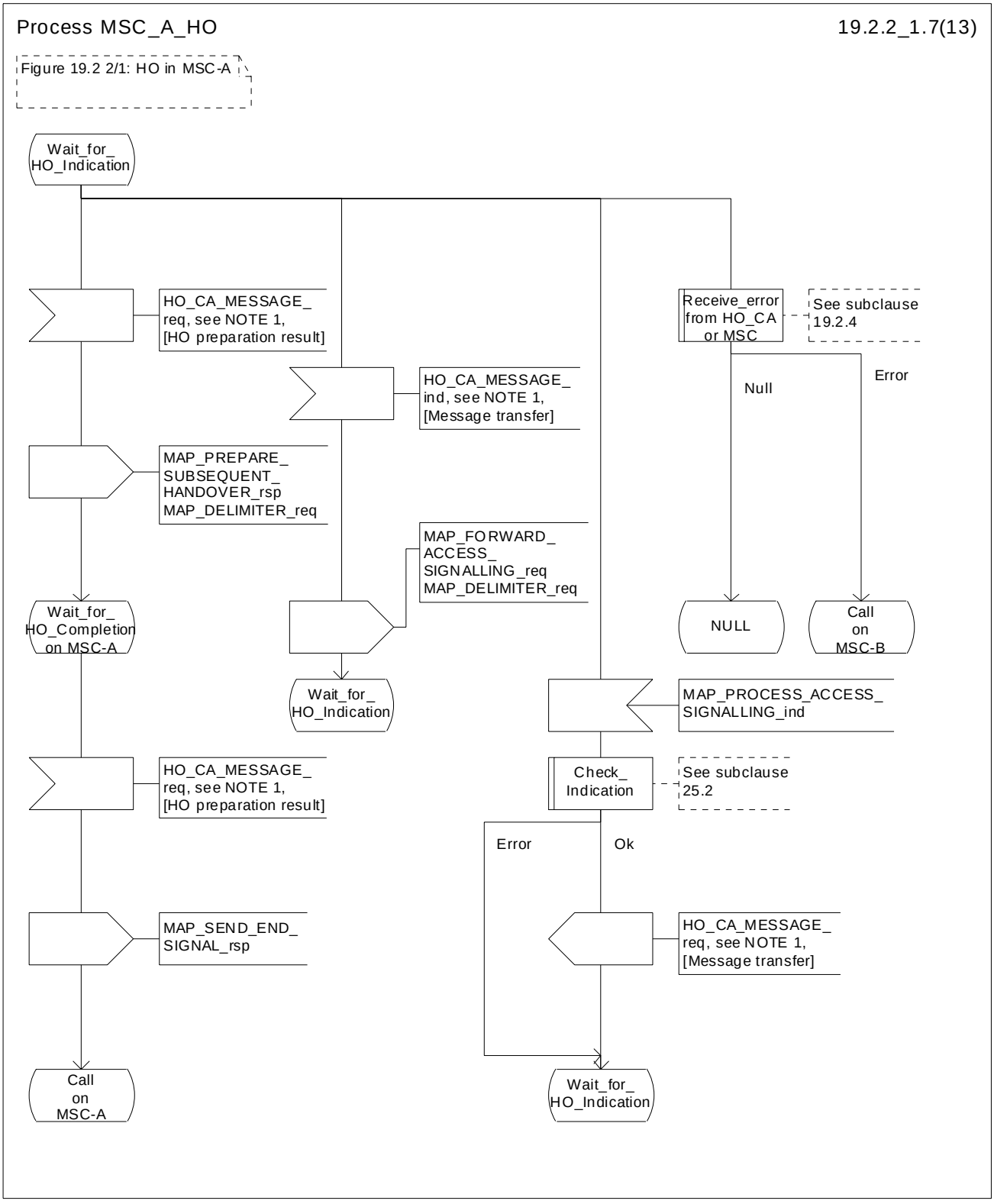


Figure 19.2.2/1 (sheet 7 of 13): Process MSC_A_HO

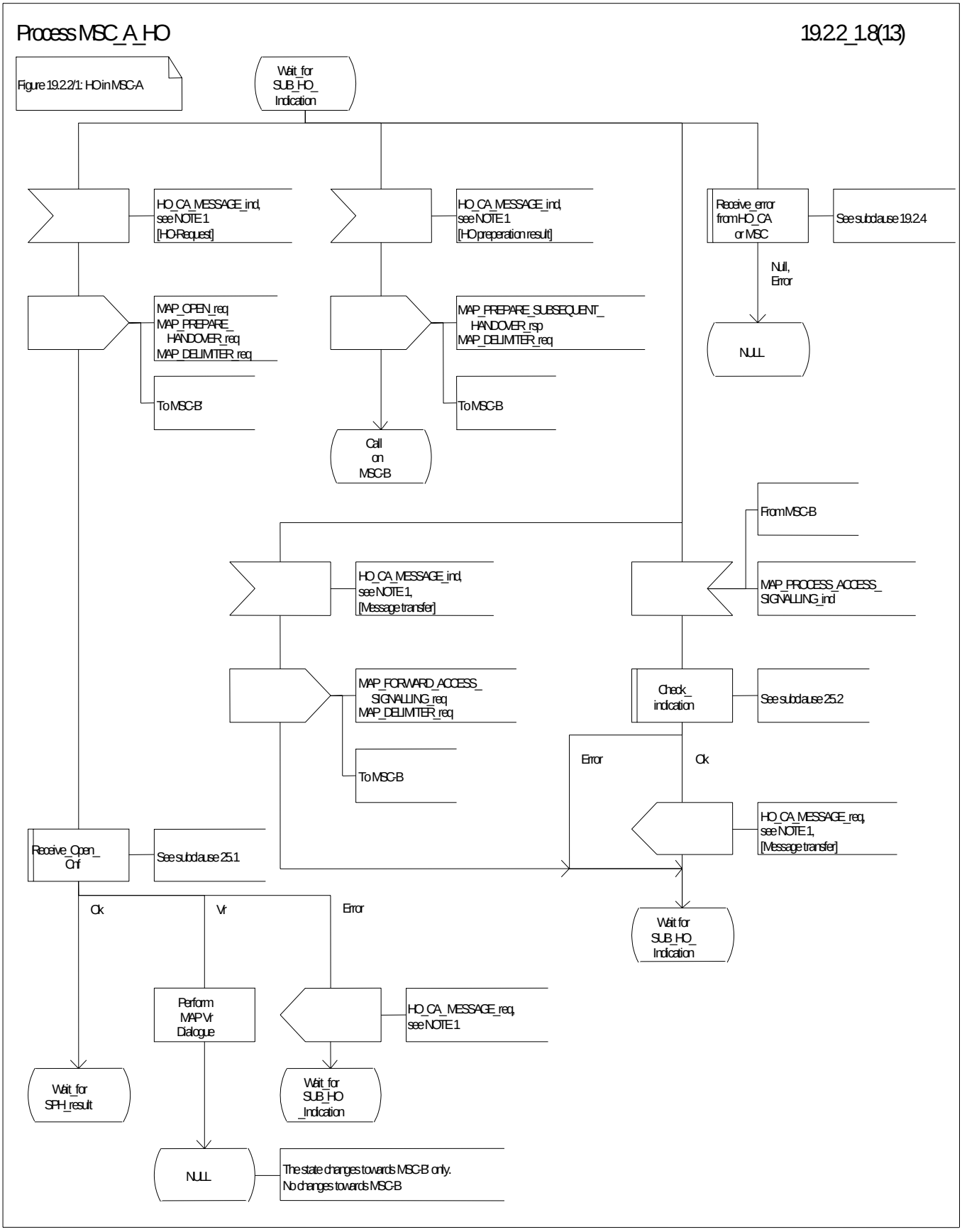


Figure 19.2.2/1 (sheet 8 of 13): Process MSC_A_HO

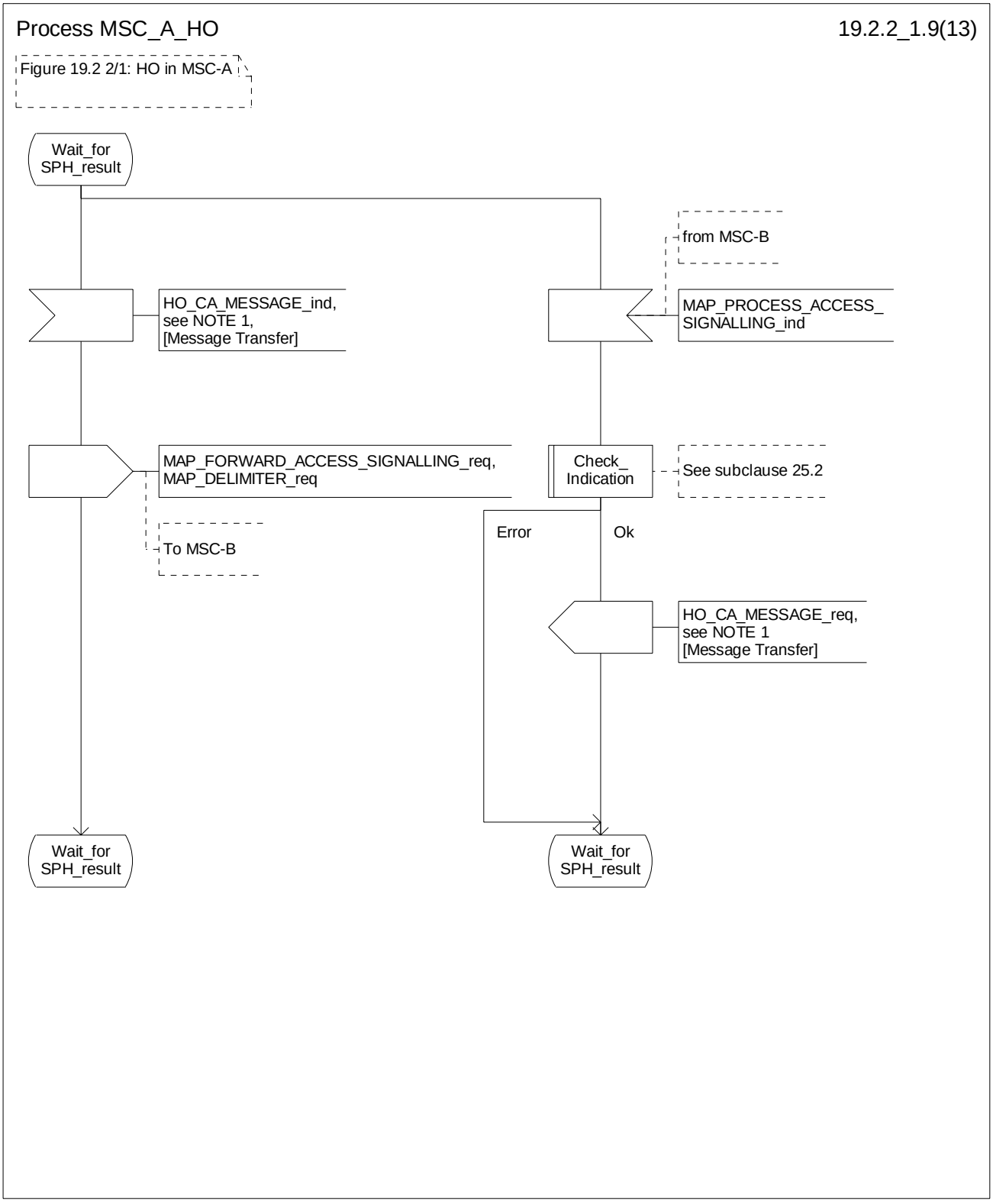


Figure 19.2.2/1 (sheet 9 of 13): Process MSC_A_HO

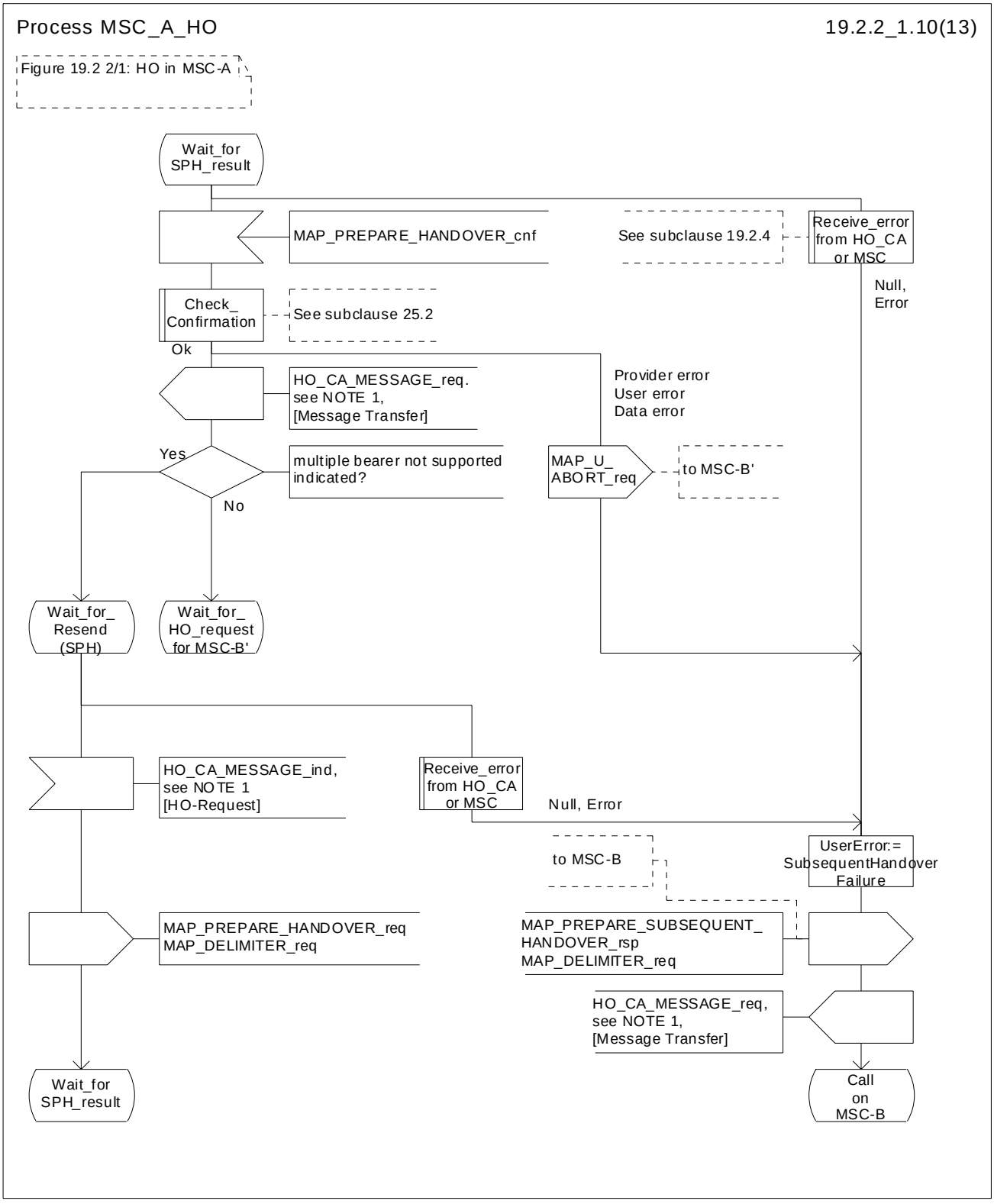


Figure 19.2.2/1 (sheet 10 of 13): Process MSC_A_HO

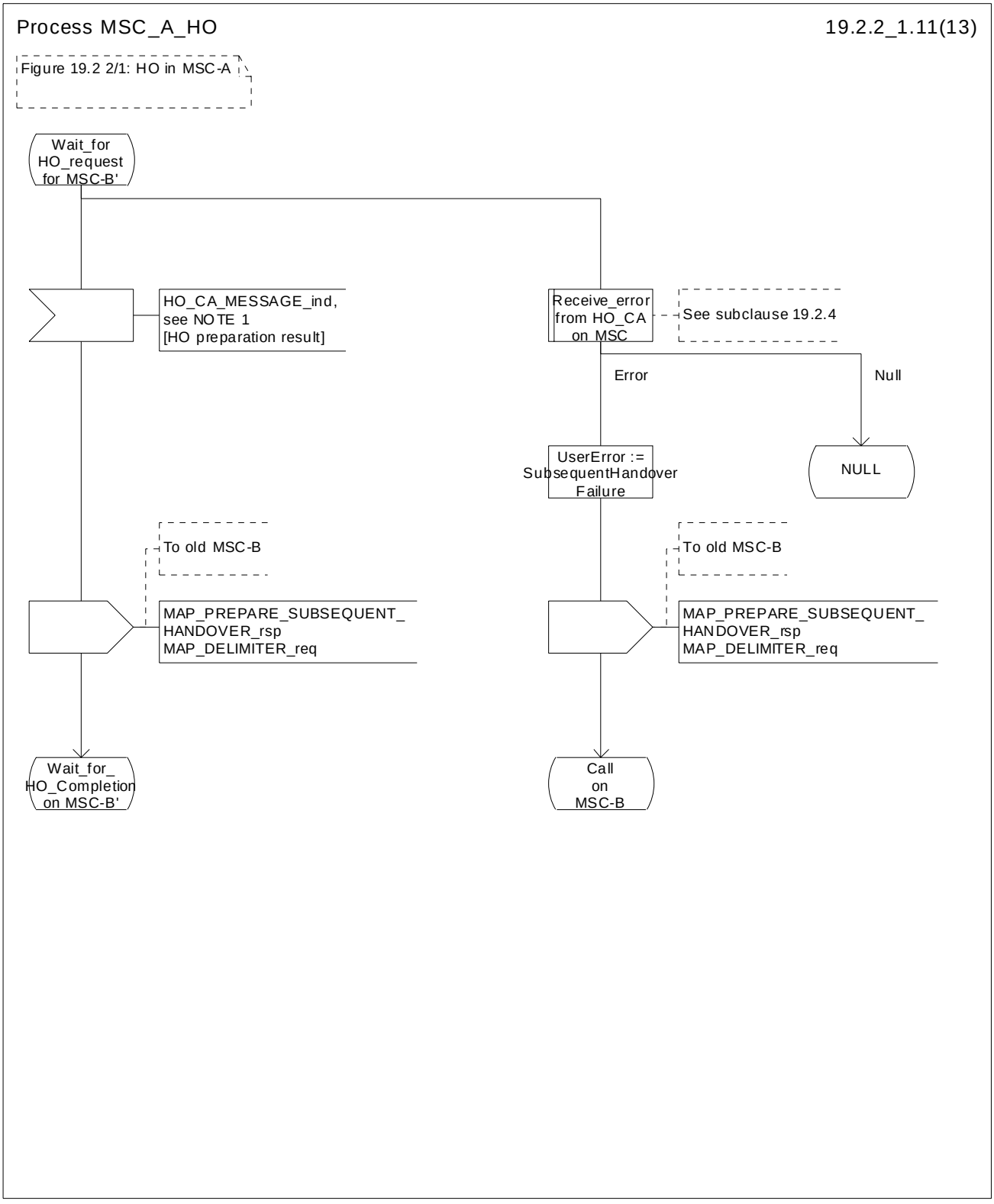


Figure 19.2.2/1 (sheet 11 of 13): Process MSC_A_HO

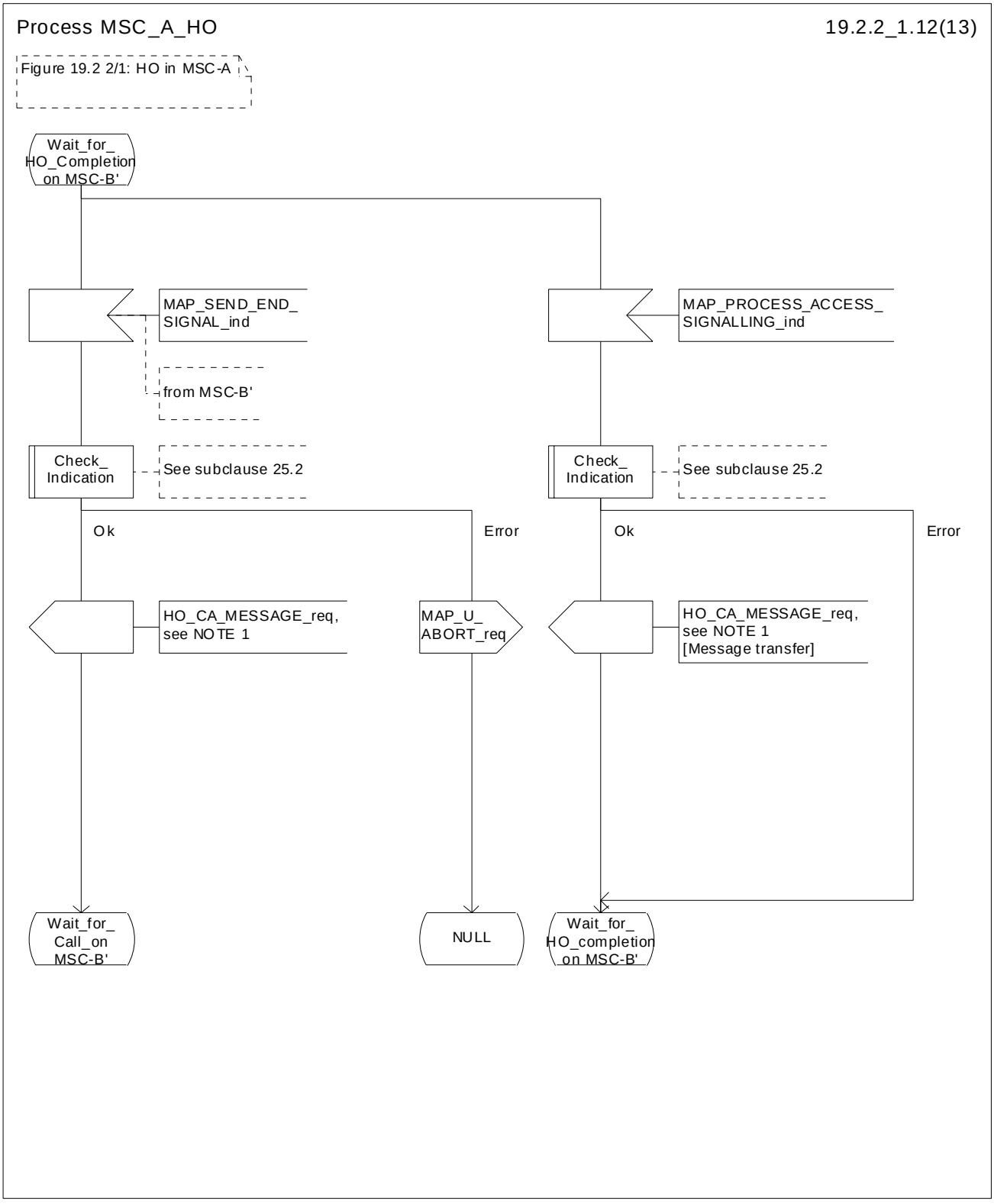


Figure 19.2.2/1 (sheet 12 of 13): Process MSC_A_HO

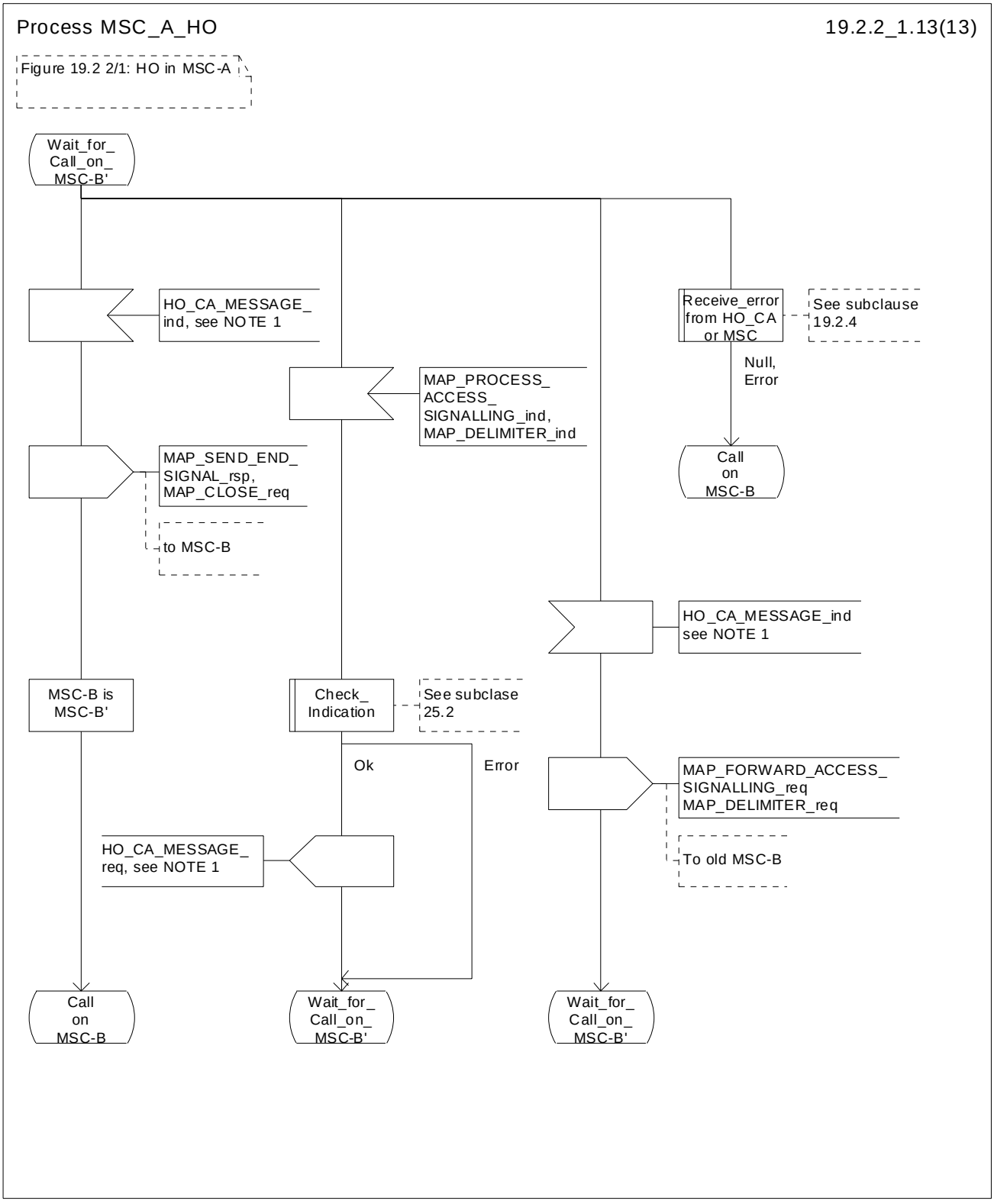


Figure 19.2.2/1 (sheet 13 of 13): Process MSC_A_HO

19.2.3 Handover procedure in MSC-B

This clause describes the handover or relocation procedure in MSC-B, including the request for a handover or relocation from another MSC (MSC-A), subsequent handover or relocation 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 clause 25.1.

When MSC-B process receives a MAP_PREPARE_HANDOVER indication from MSC-A, MSC-B requests its associated VLR to provide a handover number, unless the parameter HO-NumberNotRequired is received in the indication.

When the connection between the MS and MSC-B is established on MSC-B, the Handover Control Application will request the MAP application to indicate this event to MSC-A by invoking the MAP_SEND_END_SIGNAL request. When a call is released, MSC-A will inform MSC-B by MAP_SEND_END_SIGNAL response and the MAP dialogue between MSC-A and MSC-B is closed.

19.2.3.2 Allocation of handover number

When a handover number is required, a MAP_ALLOCATE_HANDOVER_NUMBER request will be sent to the VLR. The handover number is received in the MAP_SEND_HANDOVER_REPORT request, and will be included in the MAP_PREPARE_HANDOVER response to MSC-A.

When relocation numbers are required, one or several MAP_ALLOCATE_HANDOVER_NUMBER requests will be sent to the VLR. Each relocation number is received in a MAP_SEND_HANDOVER_REPORT request, and the collected relocation numbers will be included in the MAP_PREPARE_HANDOVER response to MSC-A.

As soon as the call from MSC-A using the handover number arrives in MSC-B, MSC-B shall release the handover number in the VLR using the MAP_SEND_HANDOVER_REPORT response.

As soon as a call from MSC-A using a relocation number arrives in MSC-B, MSC-B shall release the relocation number in the VLR using the MAP_SEND_HANDOVER_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 or the Iu-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 or relocation, a number of procedures between MSC-A and BSS-B or RNS-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 or relocated to another MSC (either back to the controlling MSC (MSC-A) or to a third MSC (MSC-B')).

After the MAP_PREPARE_SUBSEQUENT_HANDOVER response is received from MSC-A, MSC-B will await the disconnection of the call. Once the disconnect is complete, MSC-B will inform its VLR by invoking the MAP_SEND_HANDOVER_REPORT confirmation. VLR-B will then release the allocated handover number.

The subsequent handover procedure is shown in figure 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 clause.

The services used are defined in clause 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 3GPP TS 23.009.

NOTE 2: The order in the SDL diagrams to allocate first the handover number and then the radio resources is not binding.

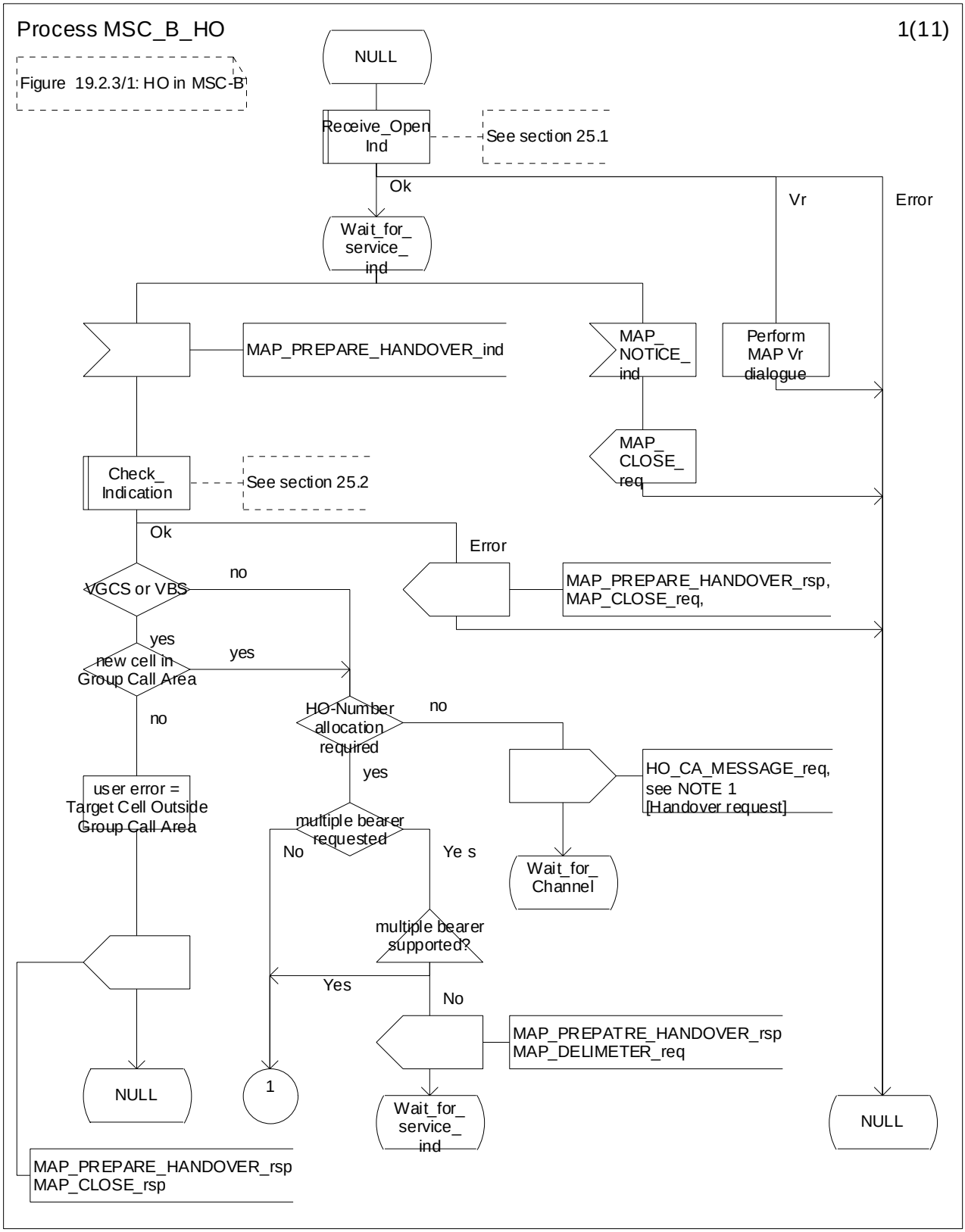


Figure 19.2.3/1 (sheet 1 of 11): Process MSC_B_HO

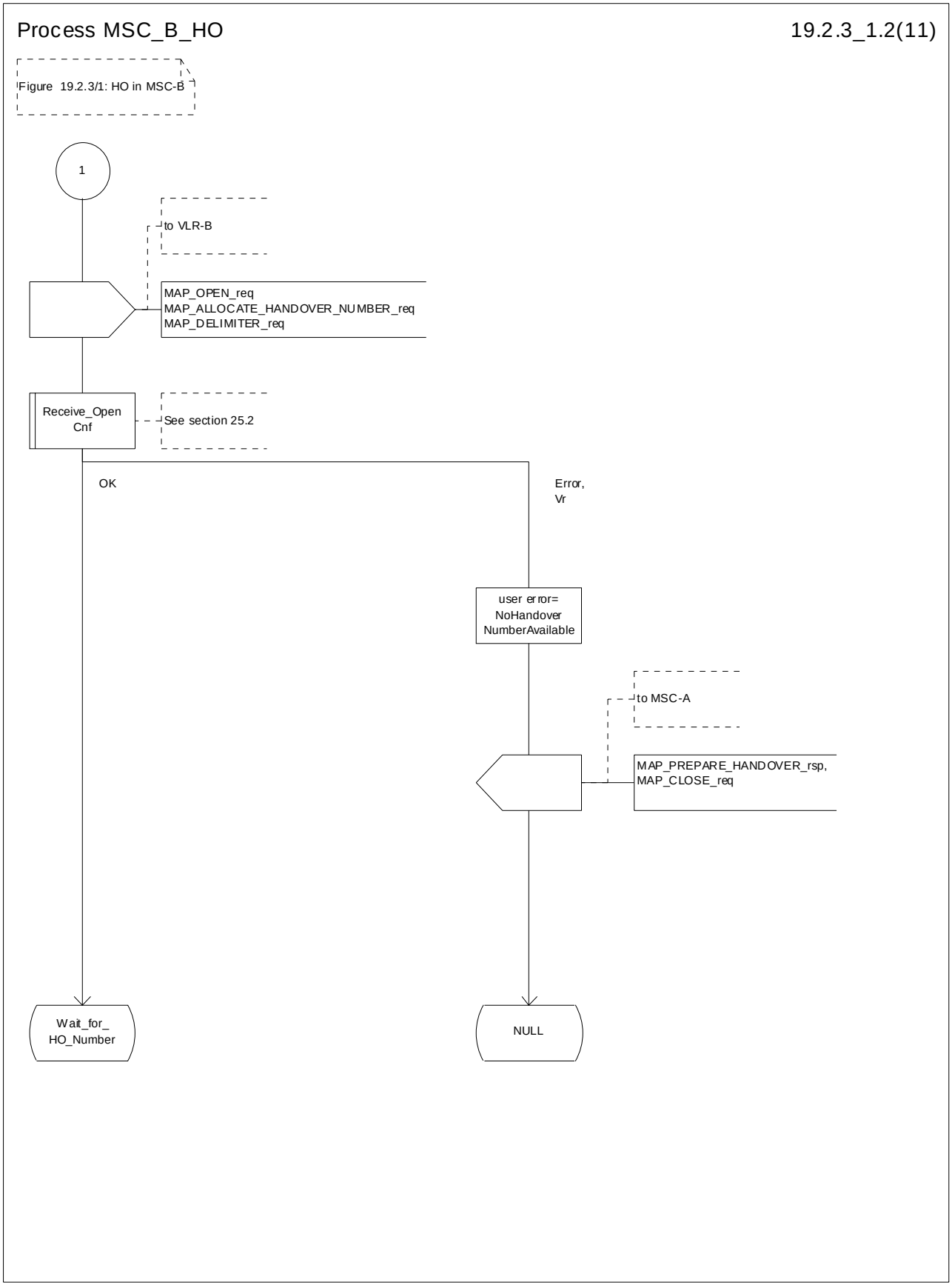


Figure 19.2.3/1 (sheet 2 of 11): Process MSC_B_HO

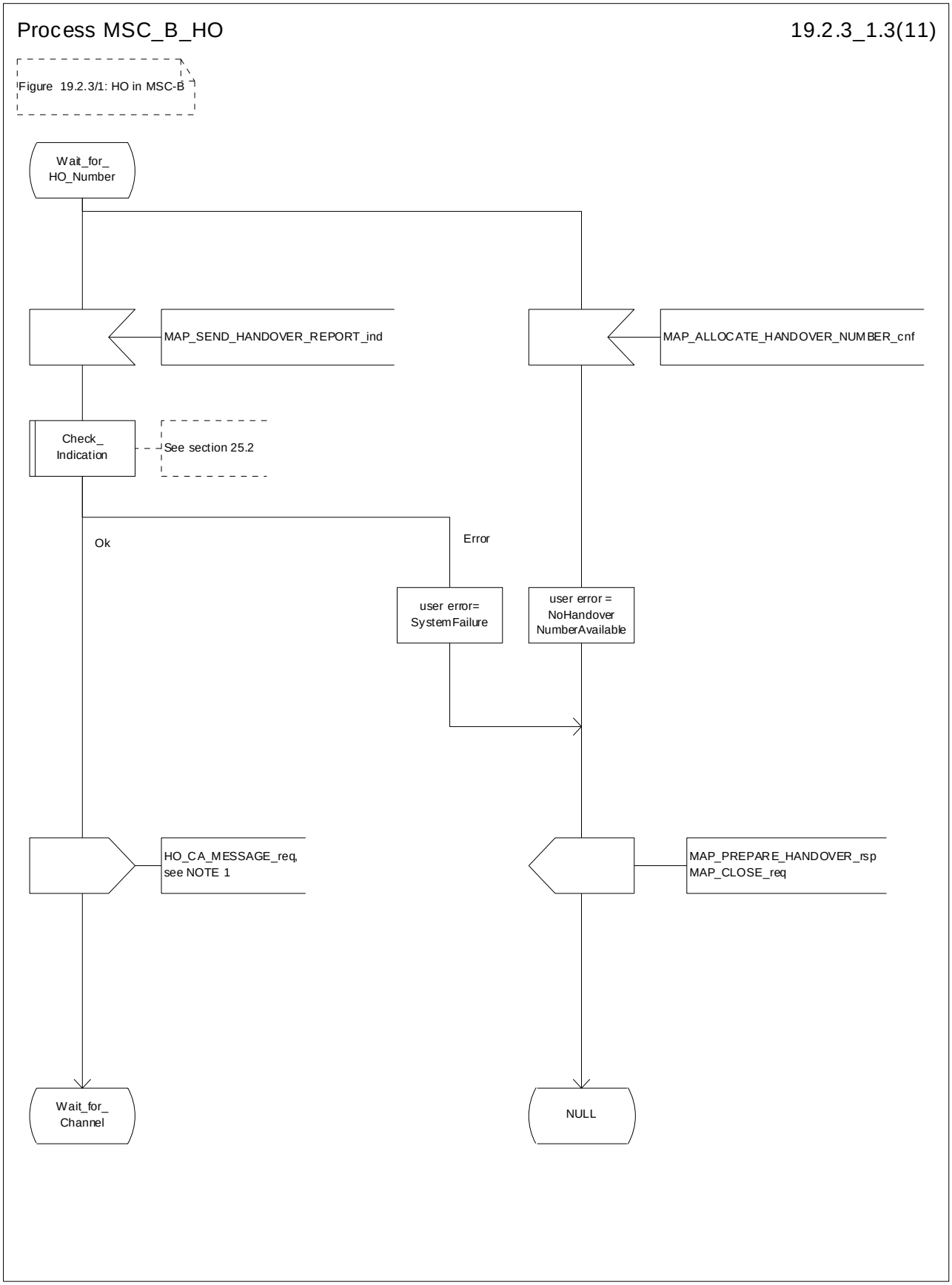


Figure 19.2.3/1 (sheet 3 of 11): Process MSC_B_HO

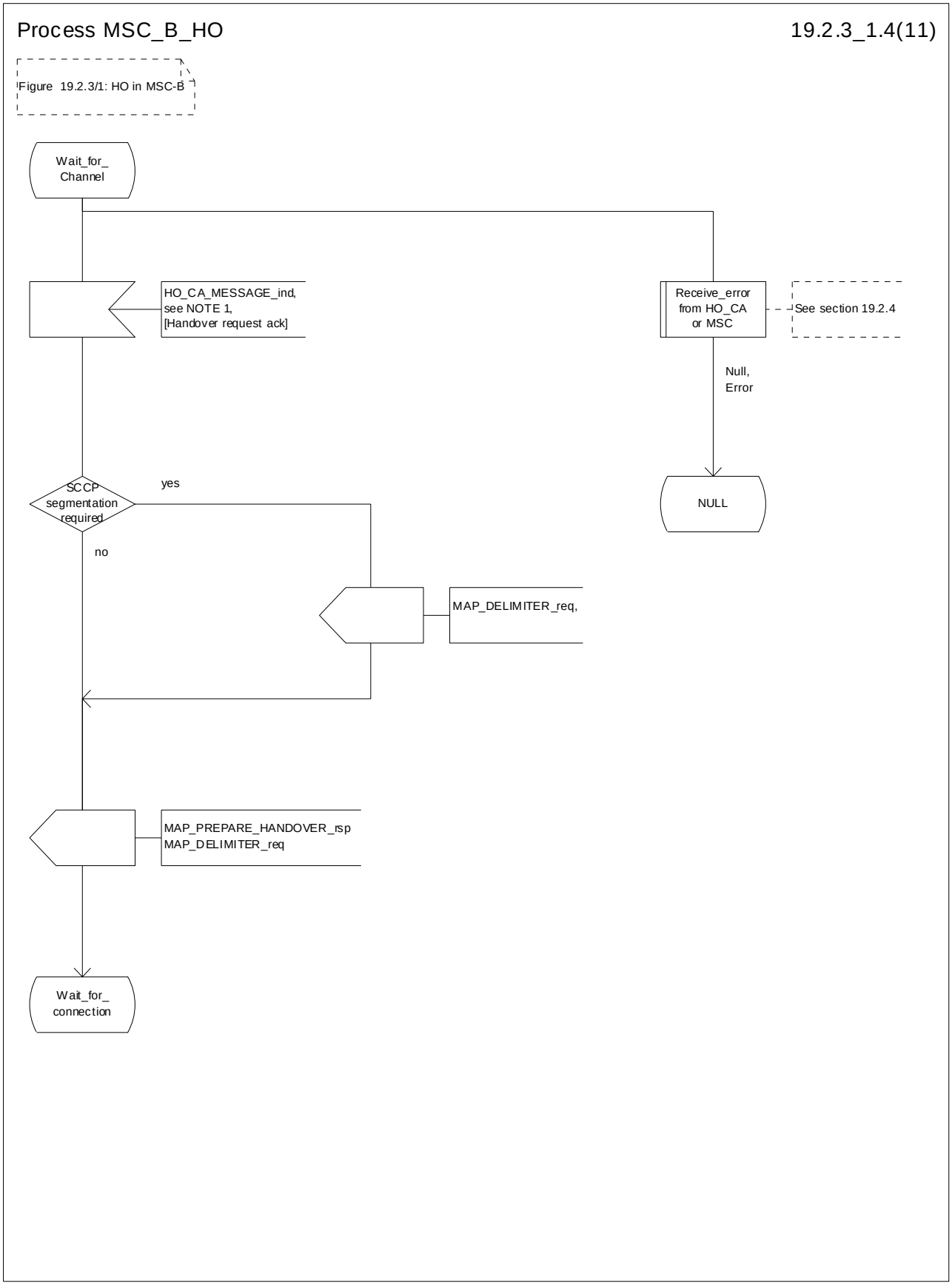


Figure 19.2.3/1 (sheet 4 of 11): Process MSC_B_HO

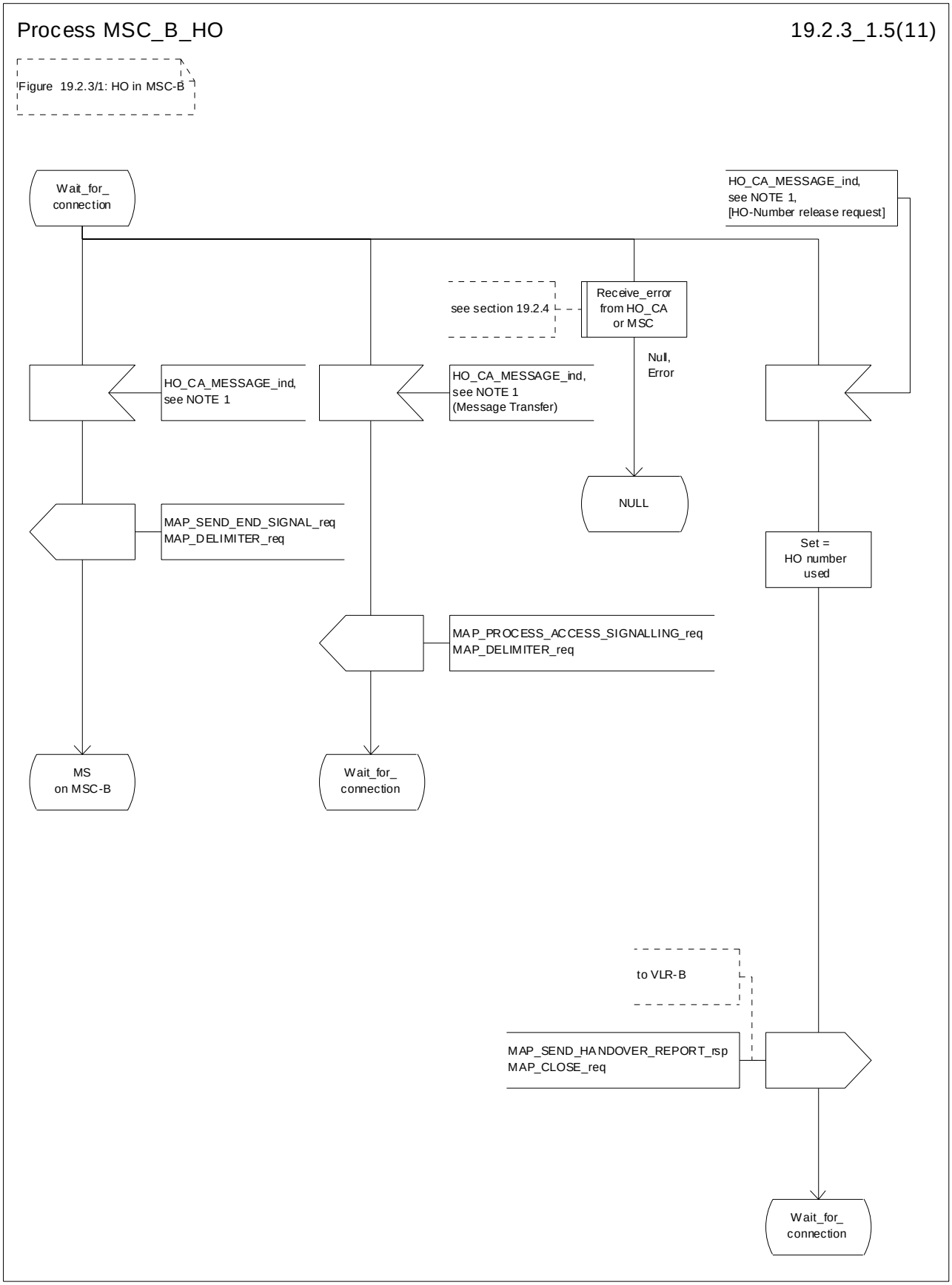


Figure 19.2.3/1 (sheet 5 of 11): Process MSC_B_HO

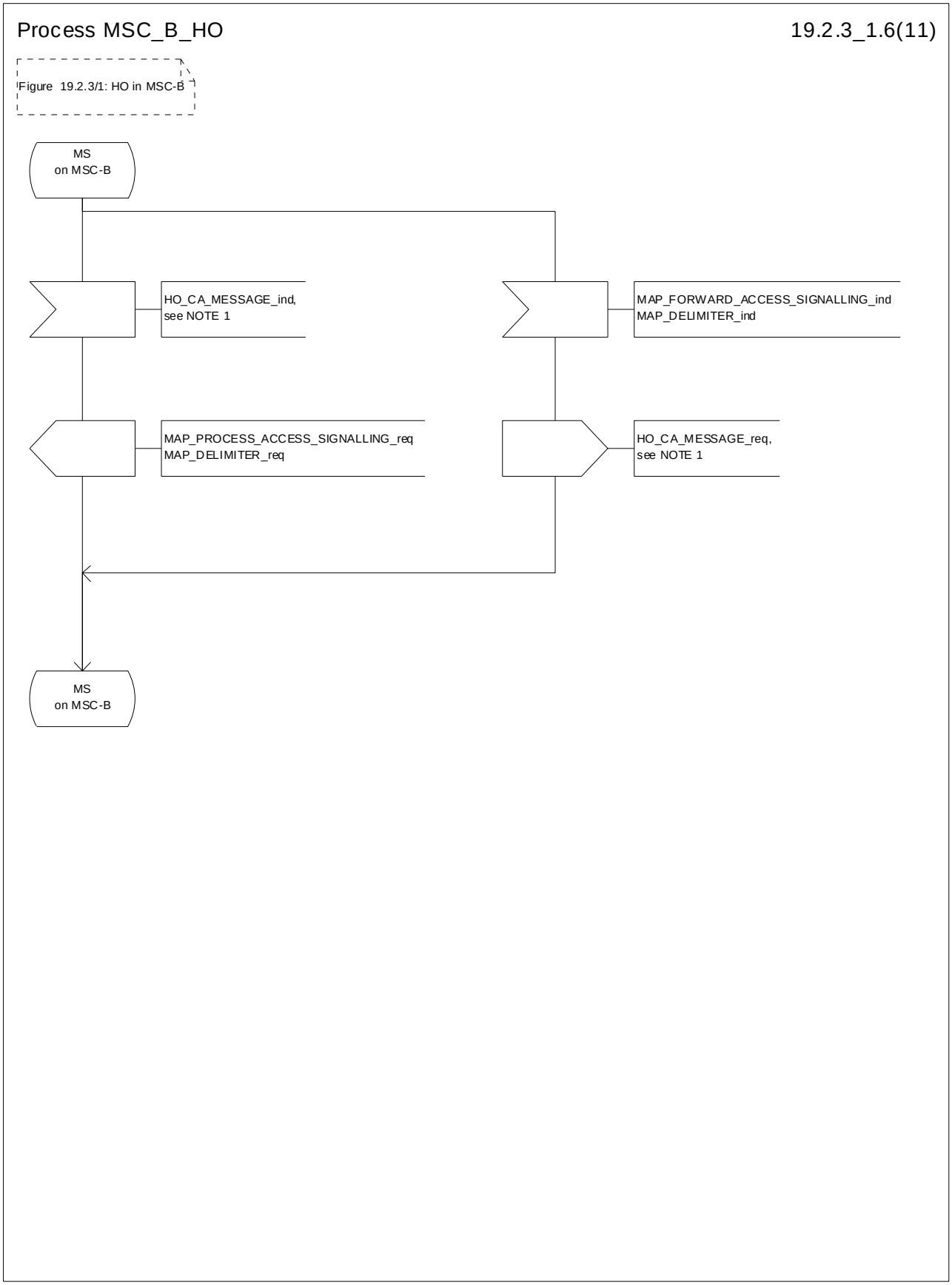


Figure 19.2.3/1 (sheet 6 of 11): Process MSC_B_HO

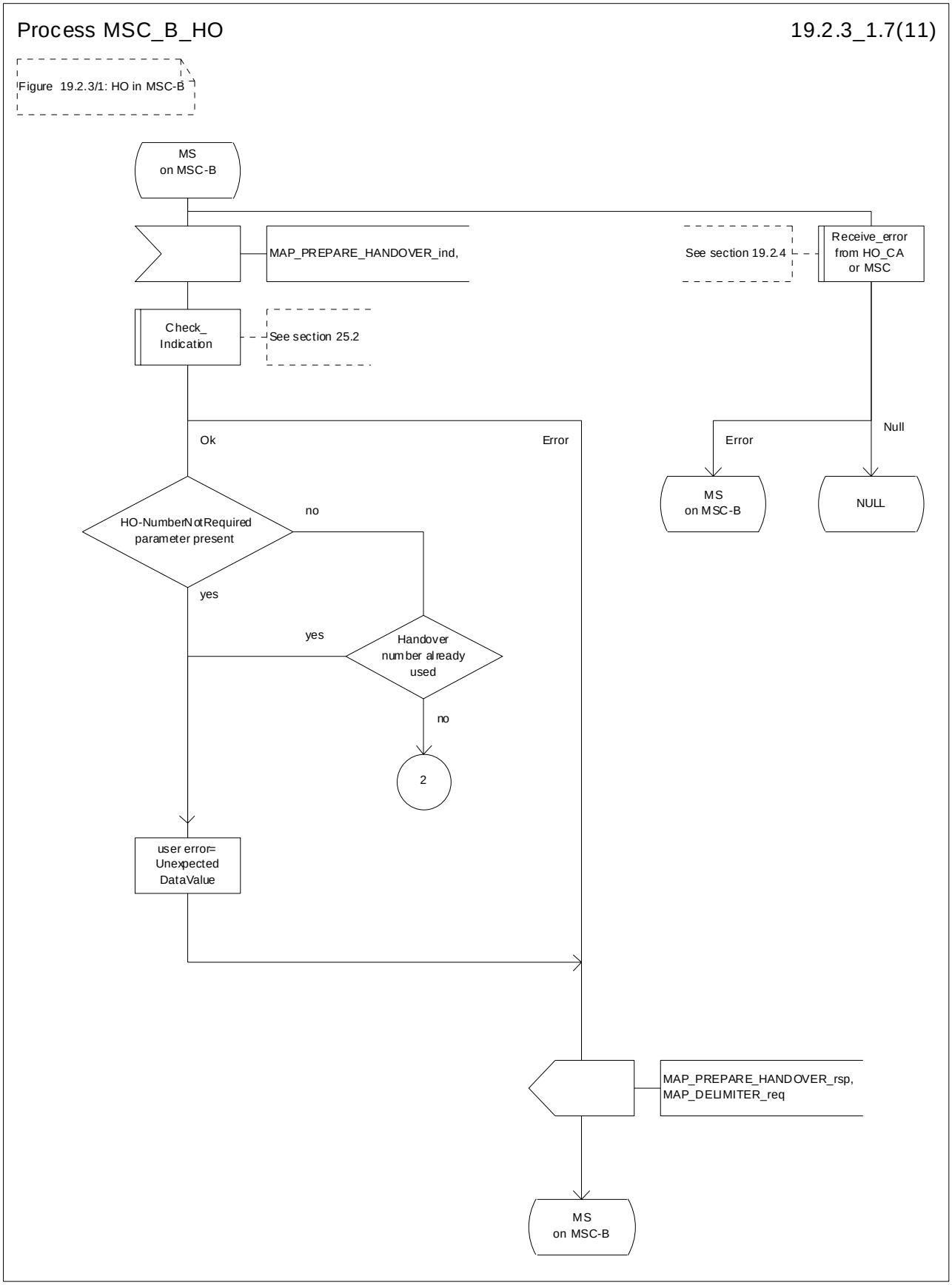


Figure 19.2.3/1 (sheet 7 of 11): Process MSC_B_HO

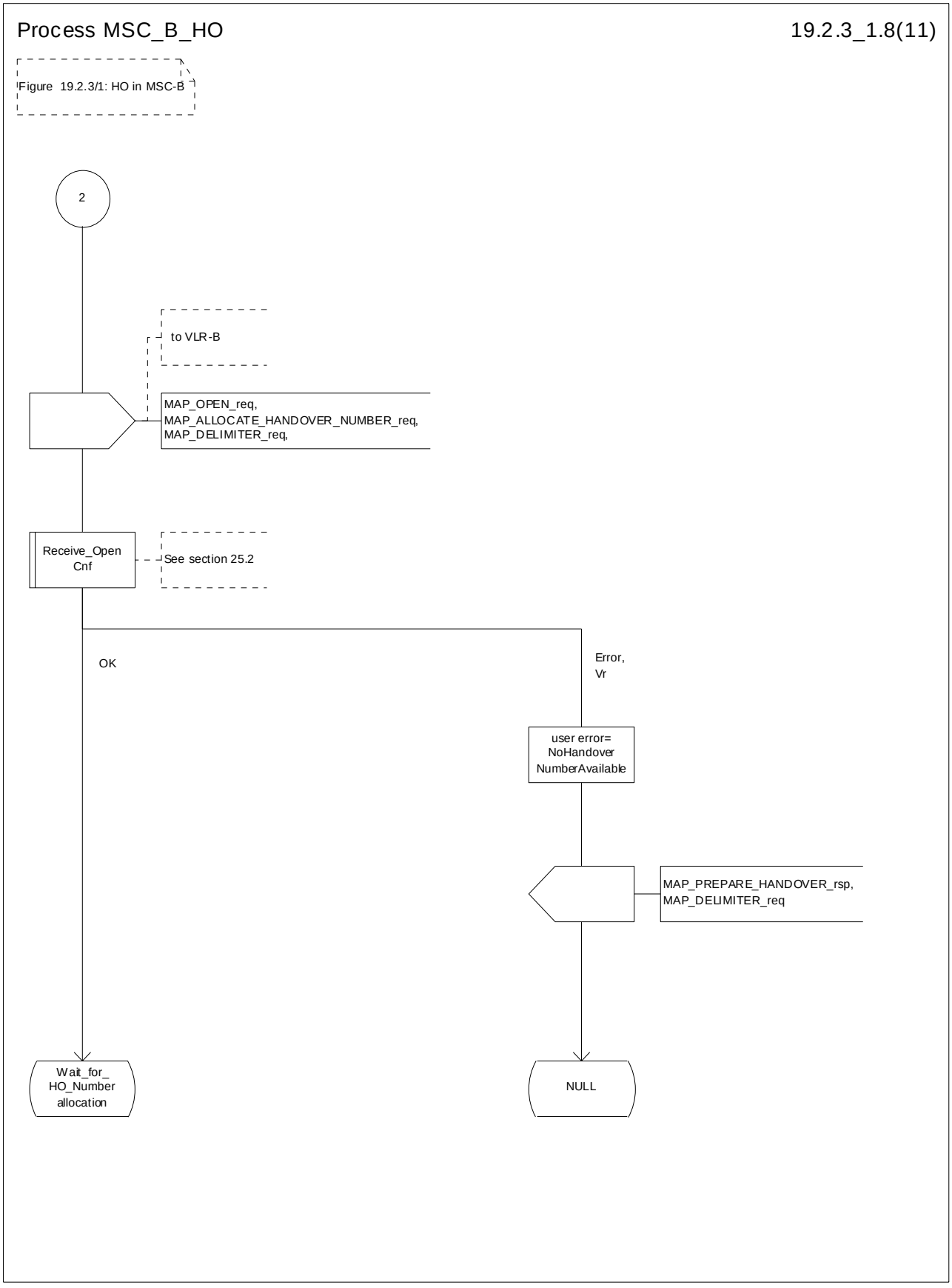


Figure 19.2.3/1 (sheet 8 of 11): Process MSC_B_HO

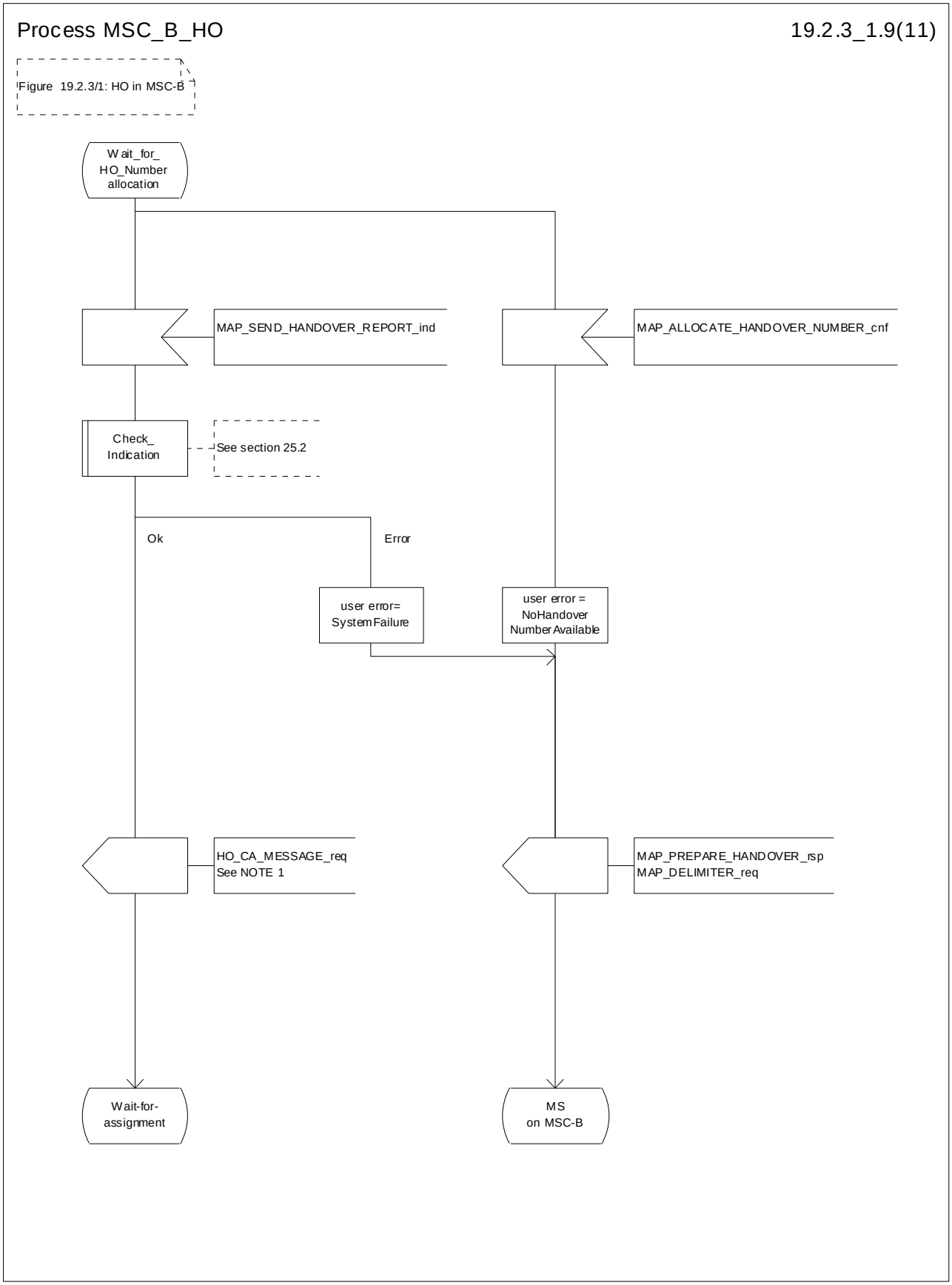


Figure 19.2.3/1 (sheet 9 of 11): Process MSC_B_HO

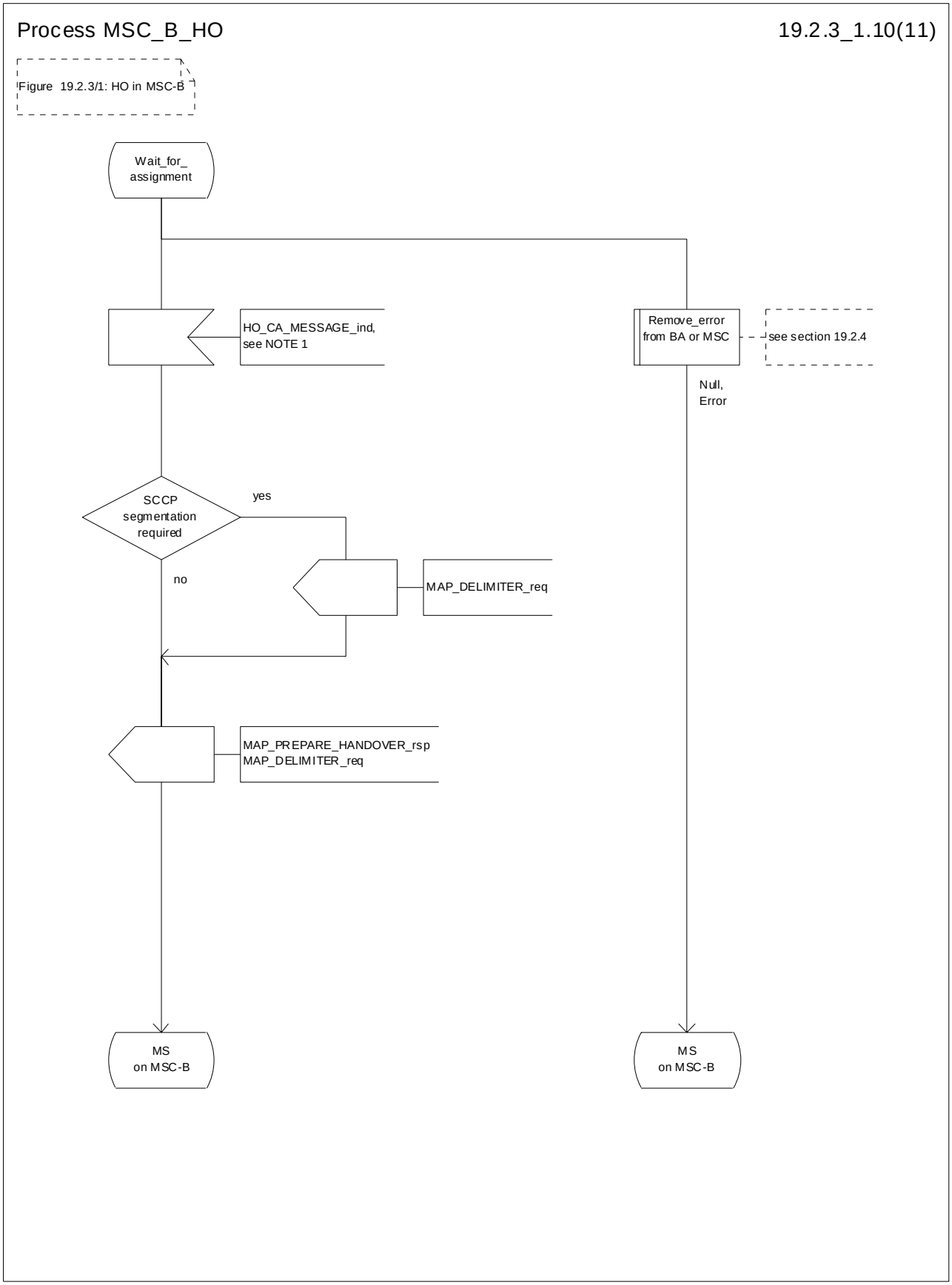


Figure 19.2.3/1 (sheet 10 of 11): Process MSC_B_HO

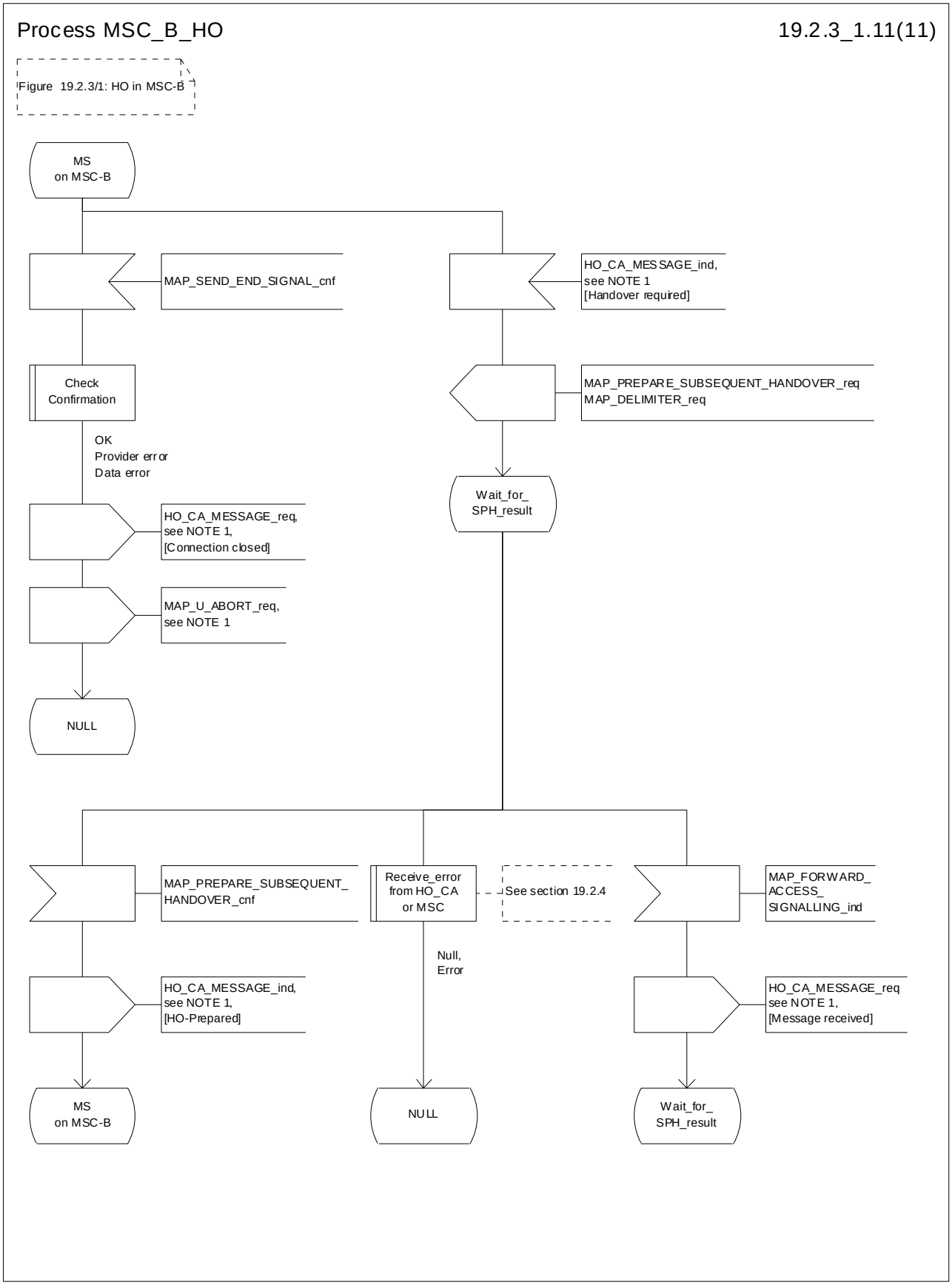


Figure 19.2.3/1 (sheet 11 of 11): Process MSC_B_HO

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 or relocation process should continue. In all other cases the MSC is returned to a "NULL" state.

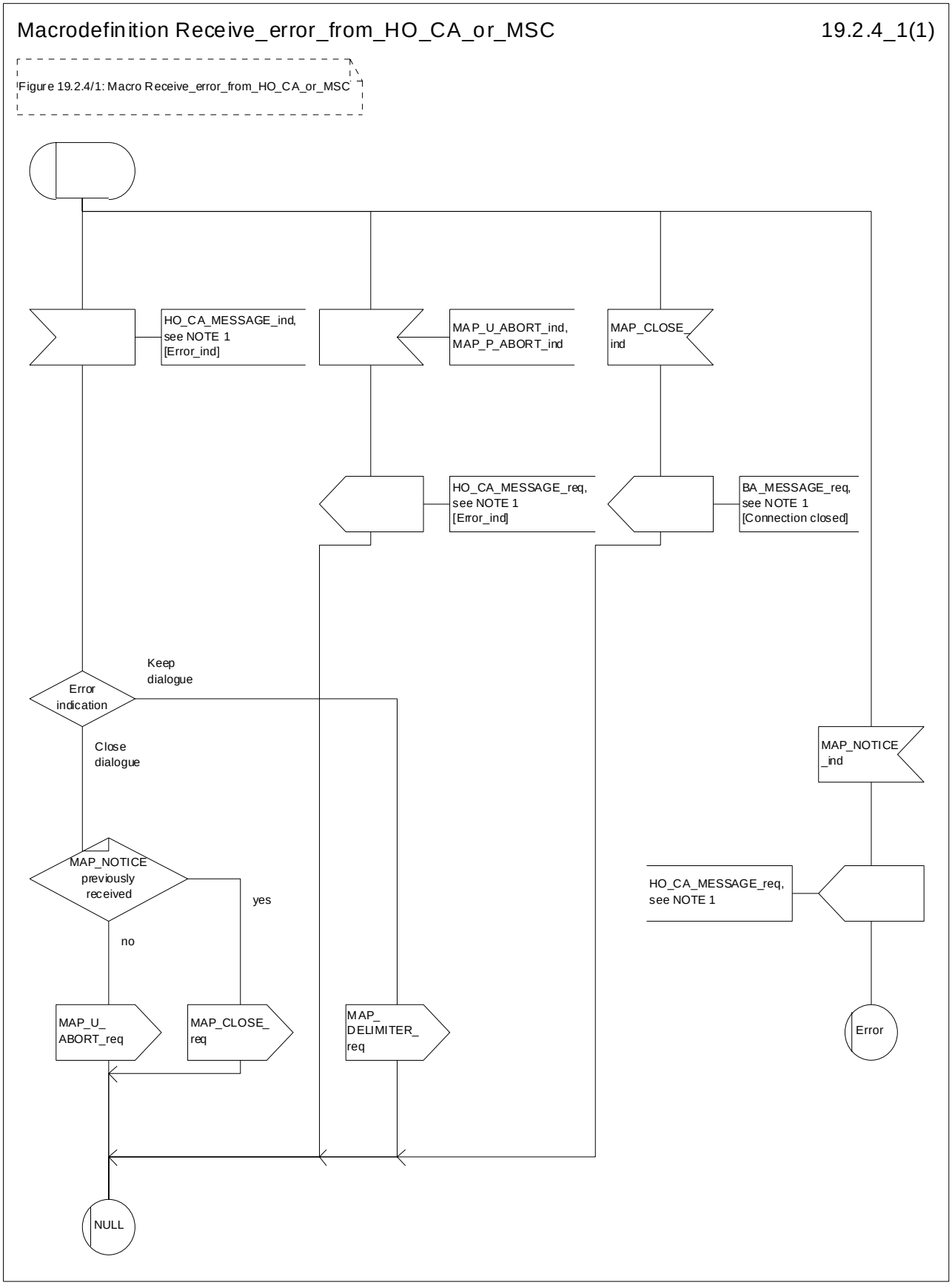


Figure 19.2.4/1: Macro Receive_error_from_HO_CA_or_MSC

19.2.5 Handover procedure in VLR

19.2.5.1 Allocation of handover number

When receiving the MAP_ALLOCATE_HANDOVER_NUMBER indication, the VLR will determine whether a handover number is available. If no handover number is available, this will be indicated by a MAP_ALLOCATE_HANDOVER_NUMBER response with the appropriate error.

The handover number allocated will otherwise be returned to MSC-B in the MAP_SEND_HANDOVER_REPORT request.

The handover number will be reserved until a MAP_SEND_HANDOVER_REPORT confirmation is received from MSC-B.

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 clause.

The services used are defined in clause 8.4.

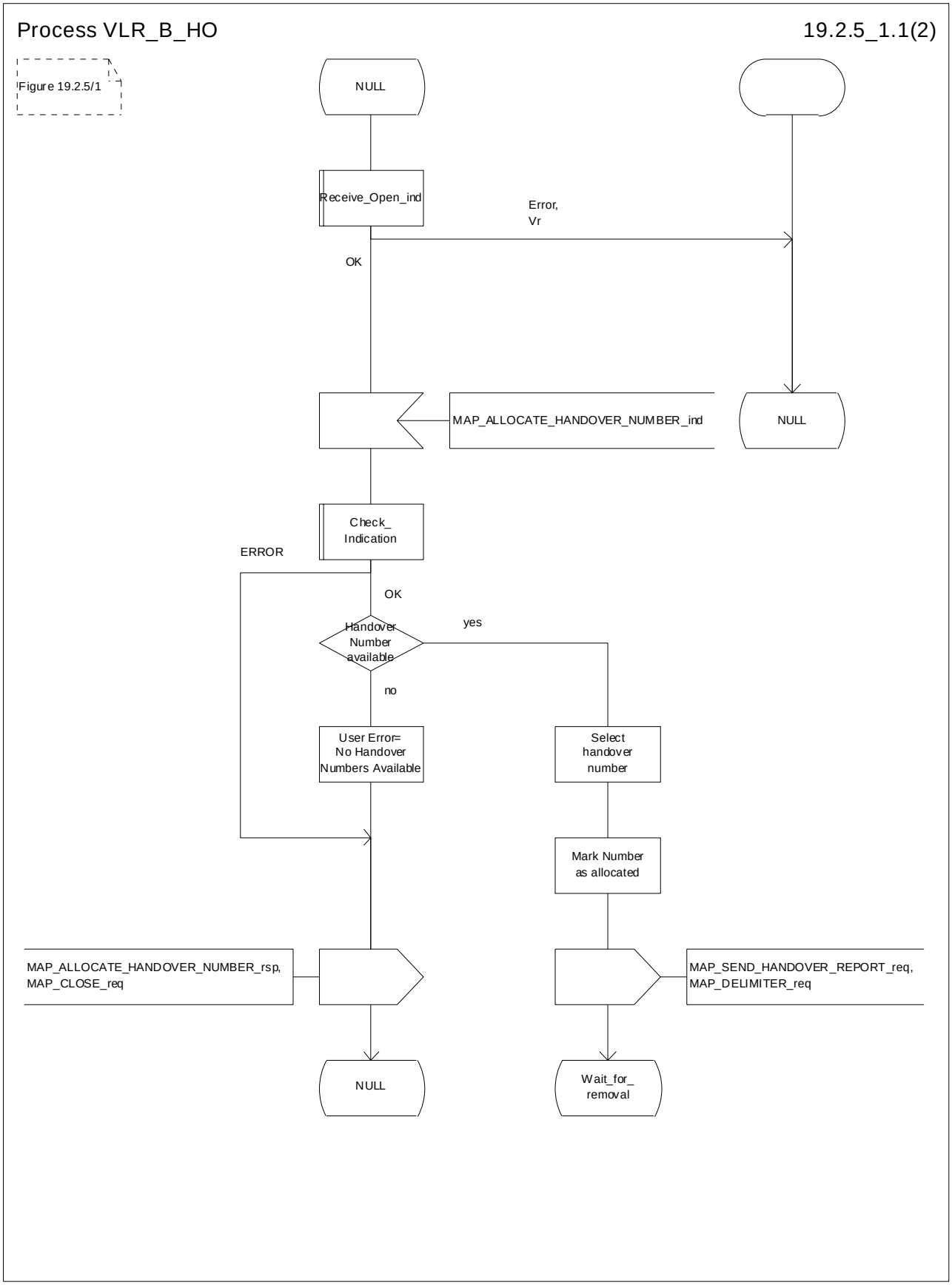


Figure 19.2.5/1 (sheet 1 of 2): Process VLR_B_HO

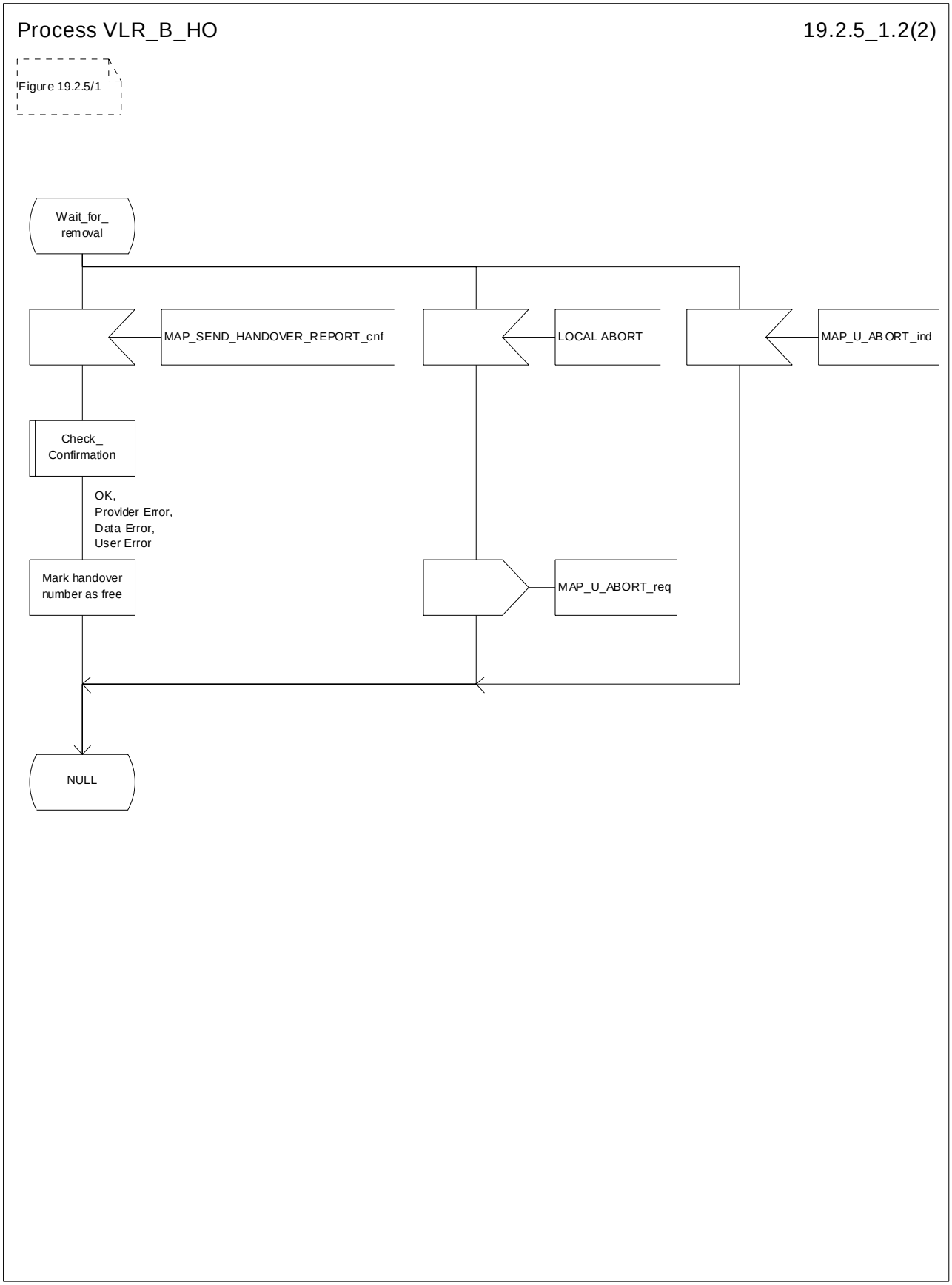


Figure 19.2.5/1 (sheet 2 of 2): Process VLR_B_HO

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 3GPP TS 23.007 [19].

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 clause 19.1.1.3;

  Update_Location_HLR clause 19.1.1.4.

- In case of a mobile terminated call:

  PRN_VLR clause 21.2.4;

  RESTORE_DATA_VLR clause 21.2.4;

  RESTORE_DATA_HLR clause 19.3.3;

  ICS_VLR clause 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 3GPP TS 23.007 [19]); and

- not confirmed by HLR by the indicator "Confirmed by HLR" (The function of this indicator is described in 3GPP TS 23.007 [19]).

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 3GPP TS 23.007 [19]).

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 clause 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 clause 21.2.4.

The MAP_RESTORE_DATA process in the HLR is described in clause 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.

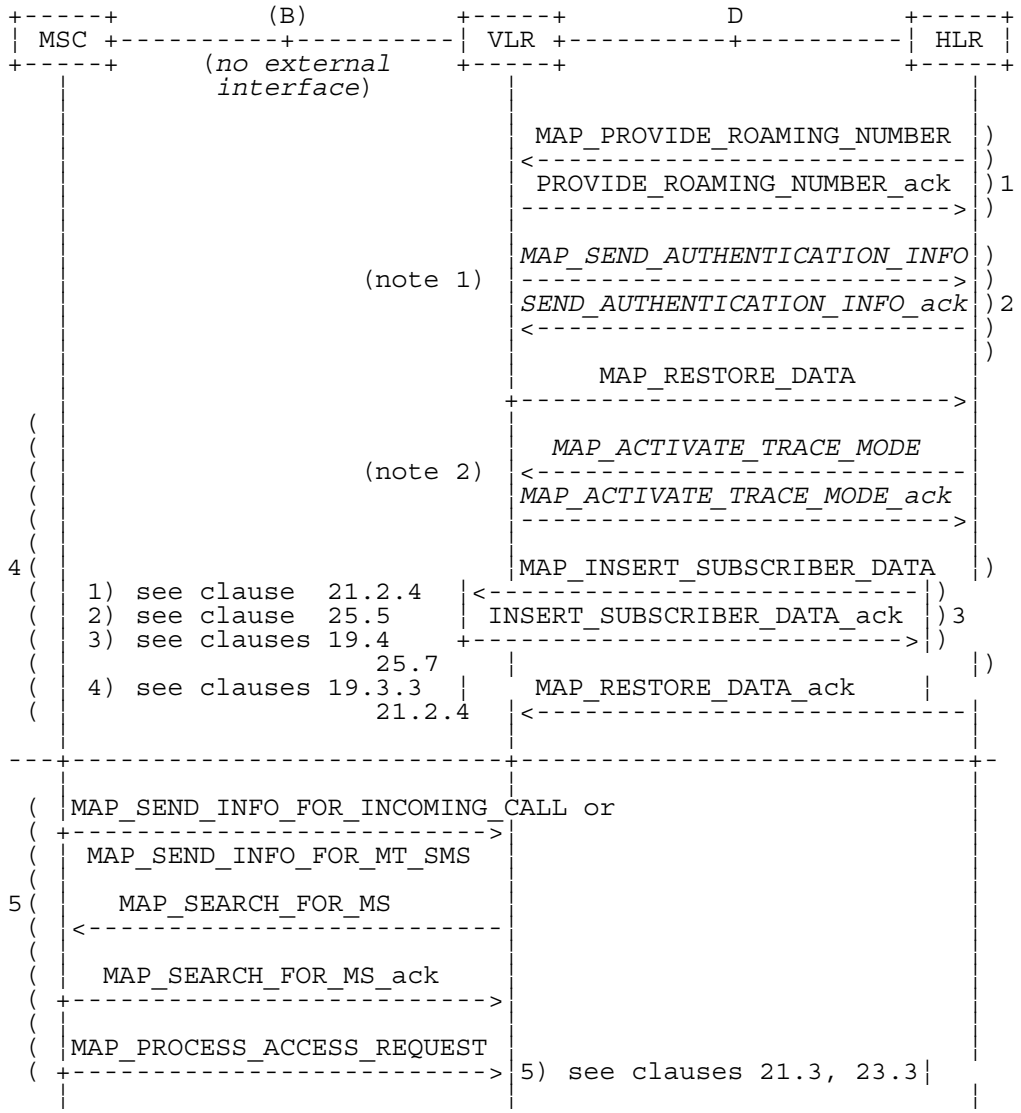


Figure 19.3/1: Procedures related to restoration of VLR in case of mobile terminated call set-up

NOTE 1: If authentication required.

NOTE 2: If subscriber tracing active in HLR.

19.3.2 HLR fault recovery procedures

The following processes are involved with the restart of the HLR:

- HLR_RESTART clause 19.3.2;
- REC_RESET_IN_VLR clause 19.3.2;
- REC_RESET_IN_SGSN clause 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 clause 19.1.1.3;
- Update_Location_HLR clause 19.1.1.4;
- Update_GPRS_Location_HLR clause 19.1.1.4;

- GPRS_Update_Location_Area_VLR clause 19.1.1.3;
- SGSN_Update_HLR clause 19.1.1.8.

In the case of a mobile originated service request, the

- Macro Process_Access_Request_VLR clause 25.4.2; and the
- Process Update_Location_HLR clause 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 clause 19.1.

After receipt of the MAP_UPDATE_LOCATION or the MAP_UPDATE_GPRS_LOCATION acknowledgement 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.

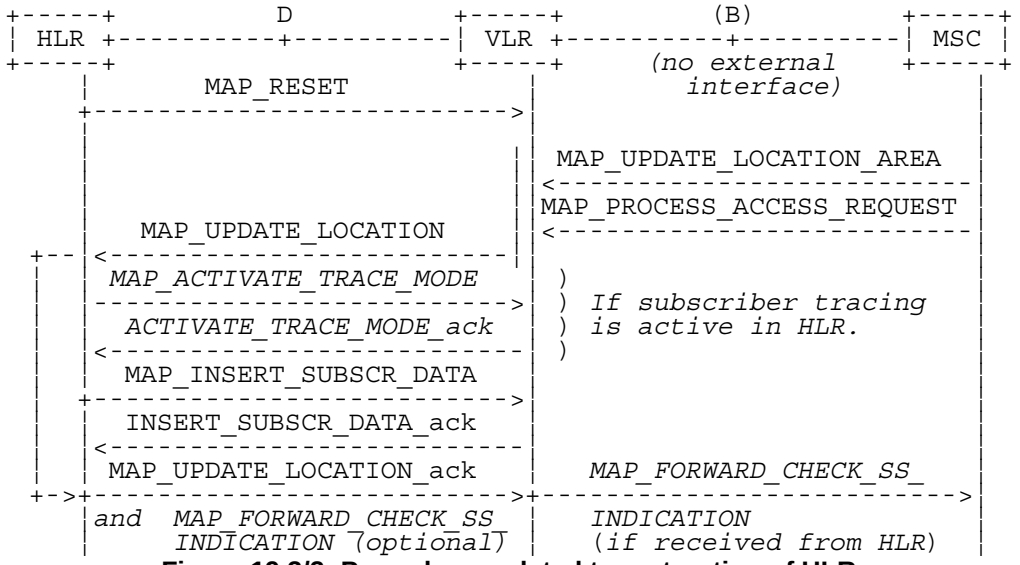


Figure 19.3/2: Procedures related to restoration of HLR

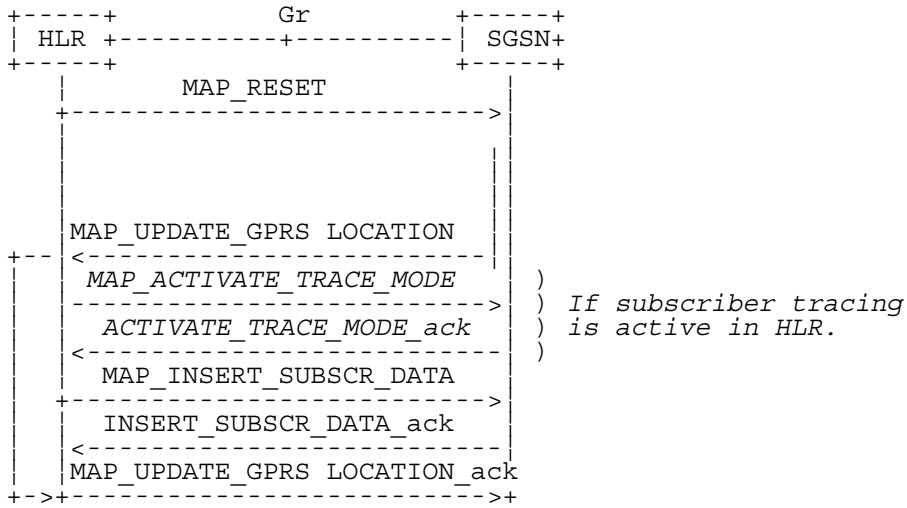


Figure 19.3/9: Procedures related to restoration of HLR for GPRS

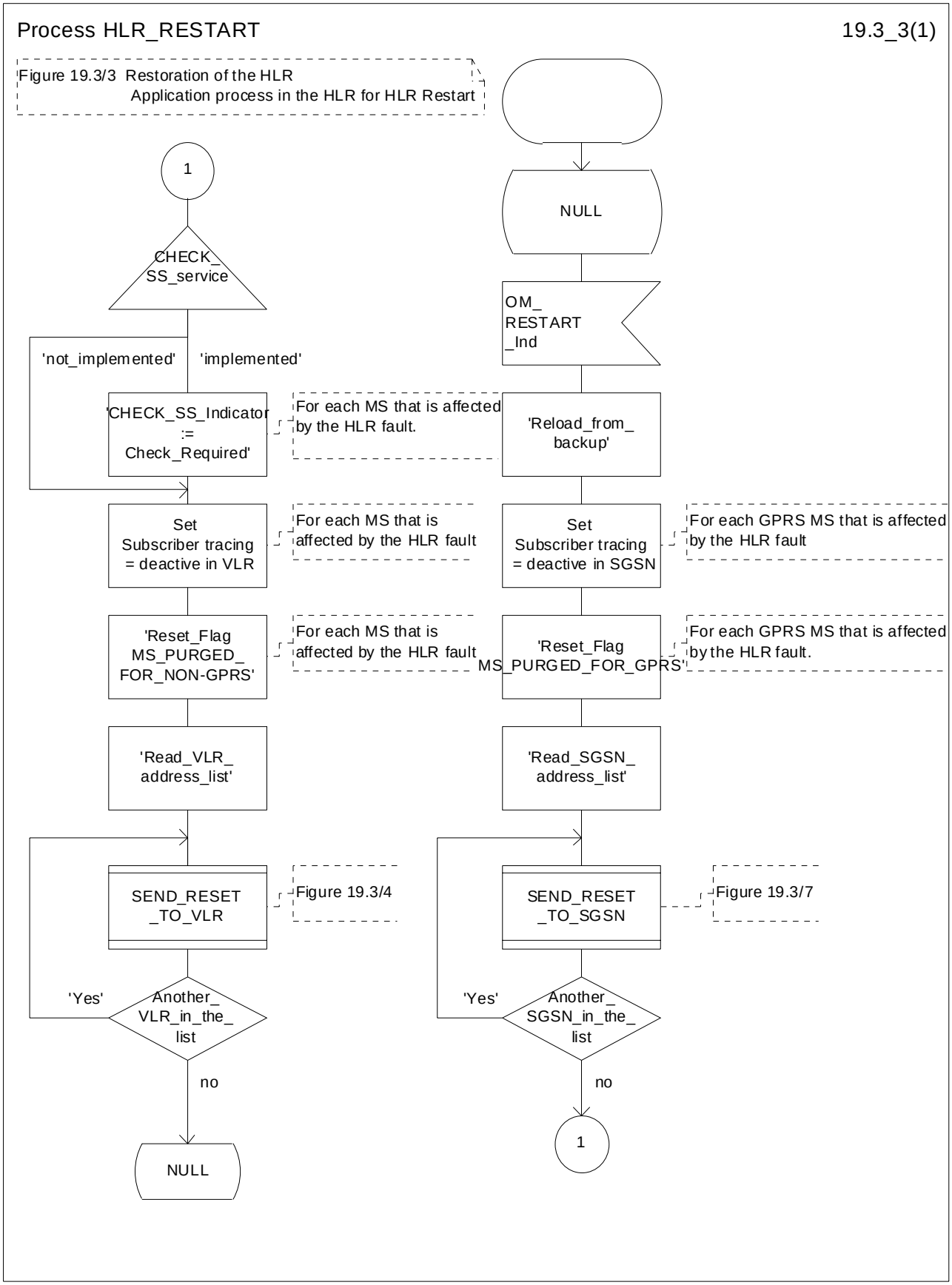


Figure 19.3/3: Process HLR_RESTART

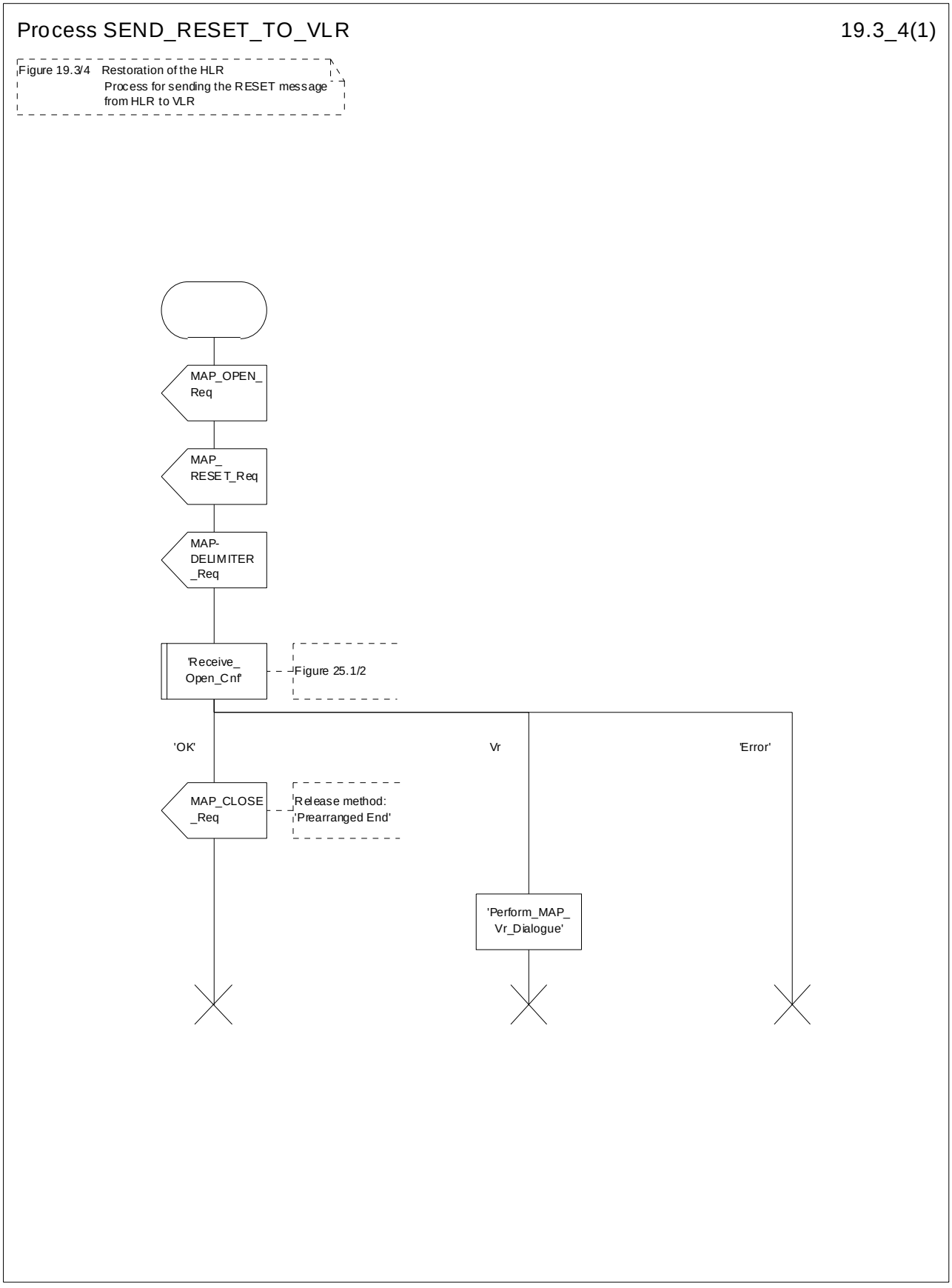


Figure 19.3/4: Process SEND_RESET_TO_VLR

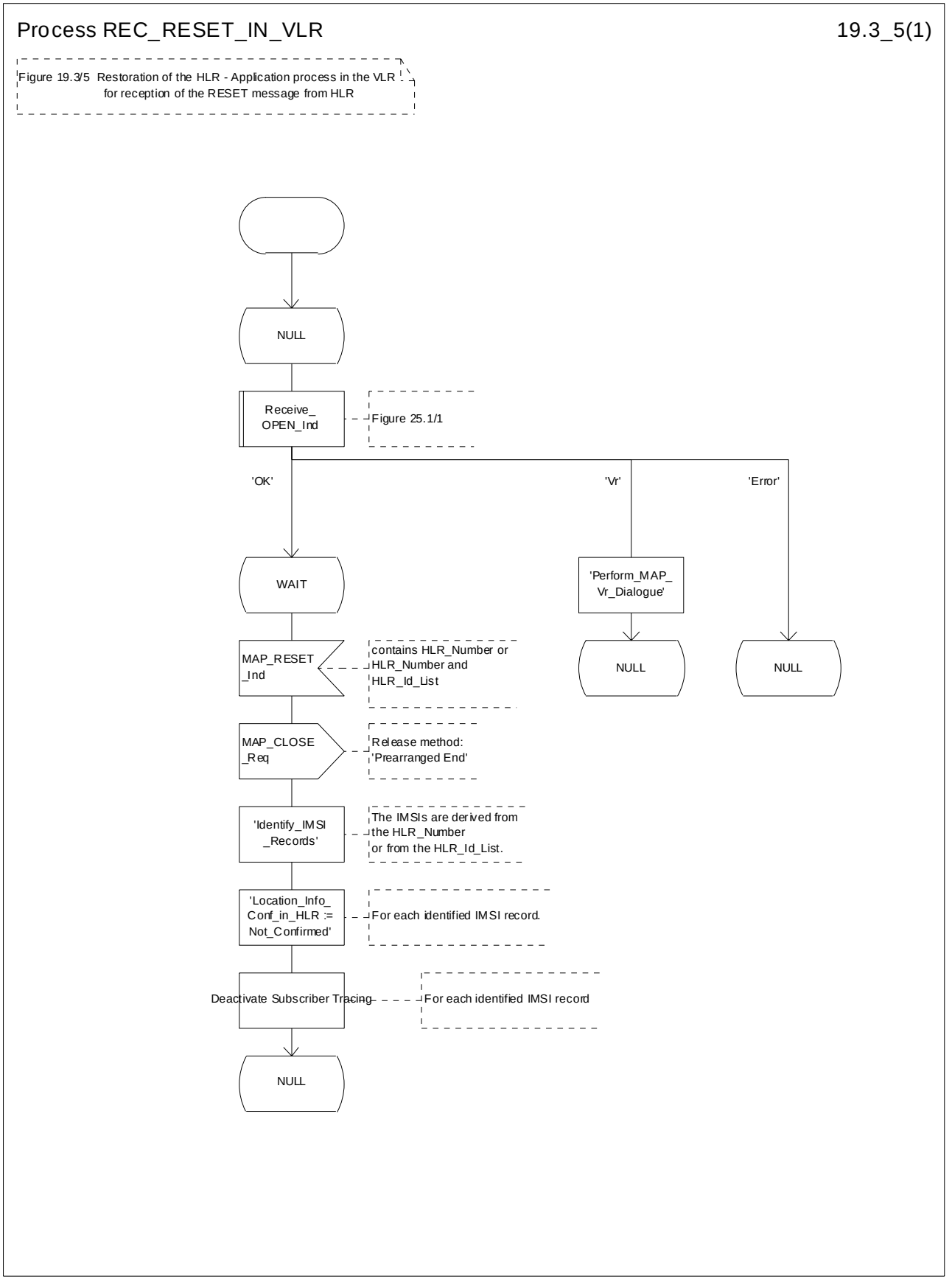


Figure 19.3/5: Process REC_RESET_IN_VLR

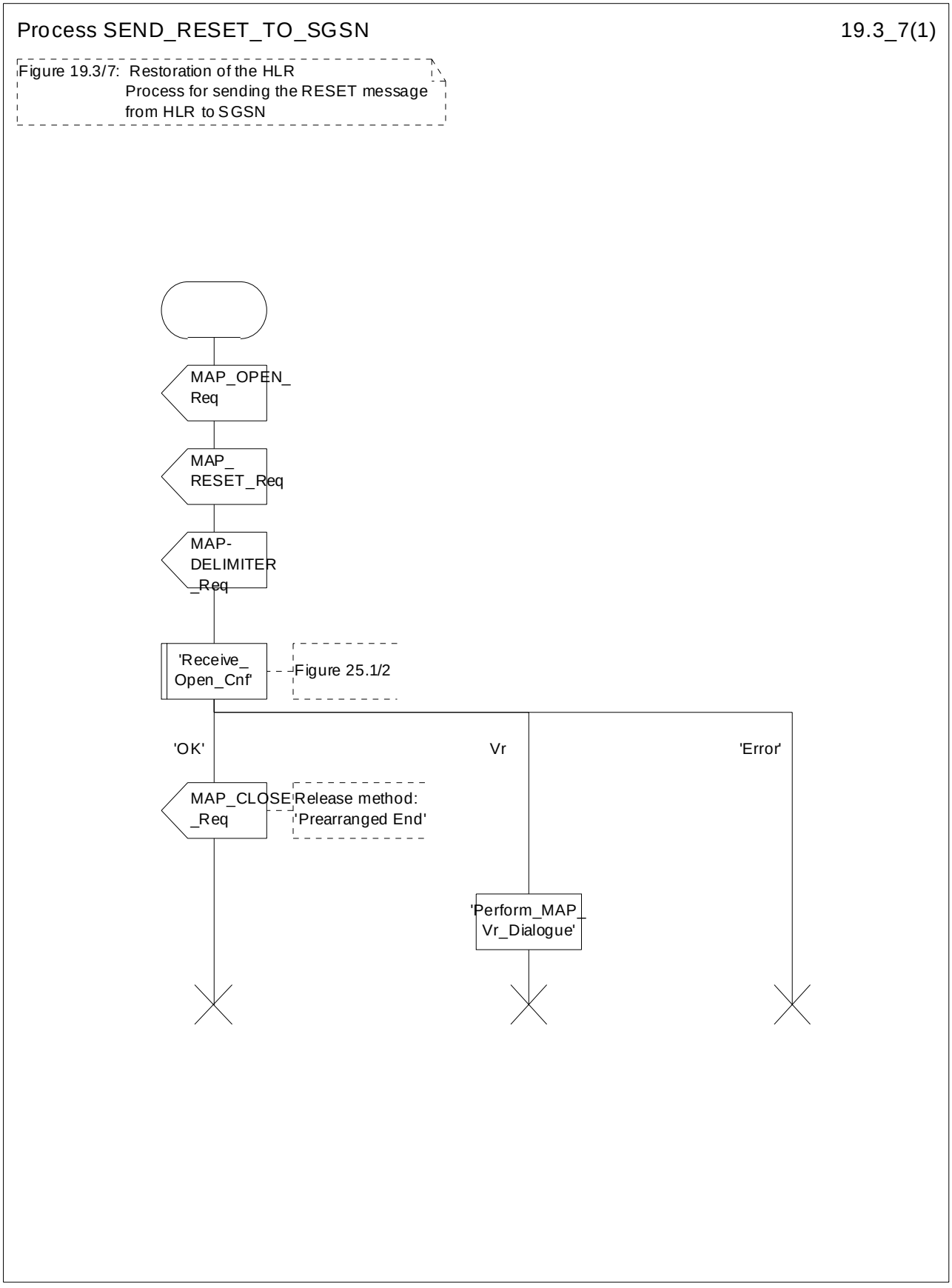


Figure 19.3/7: Process SEND_RESET_TO_SGSN

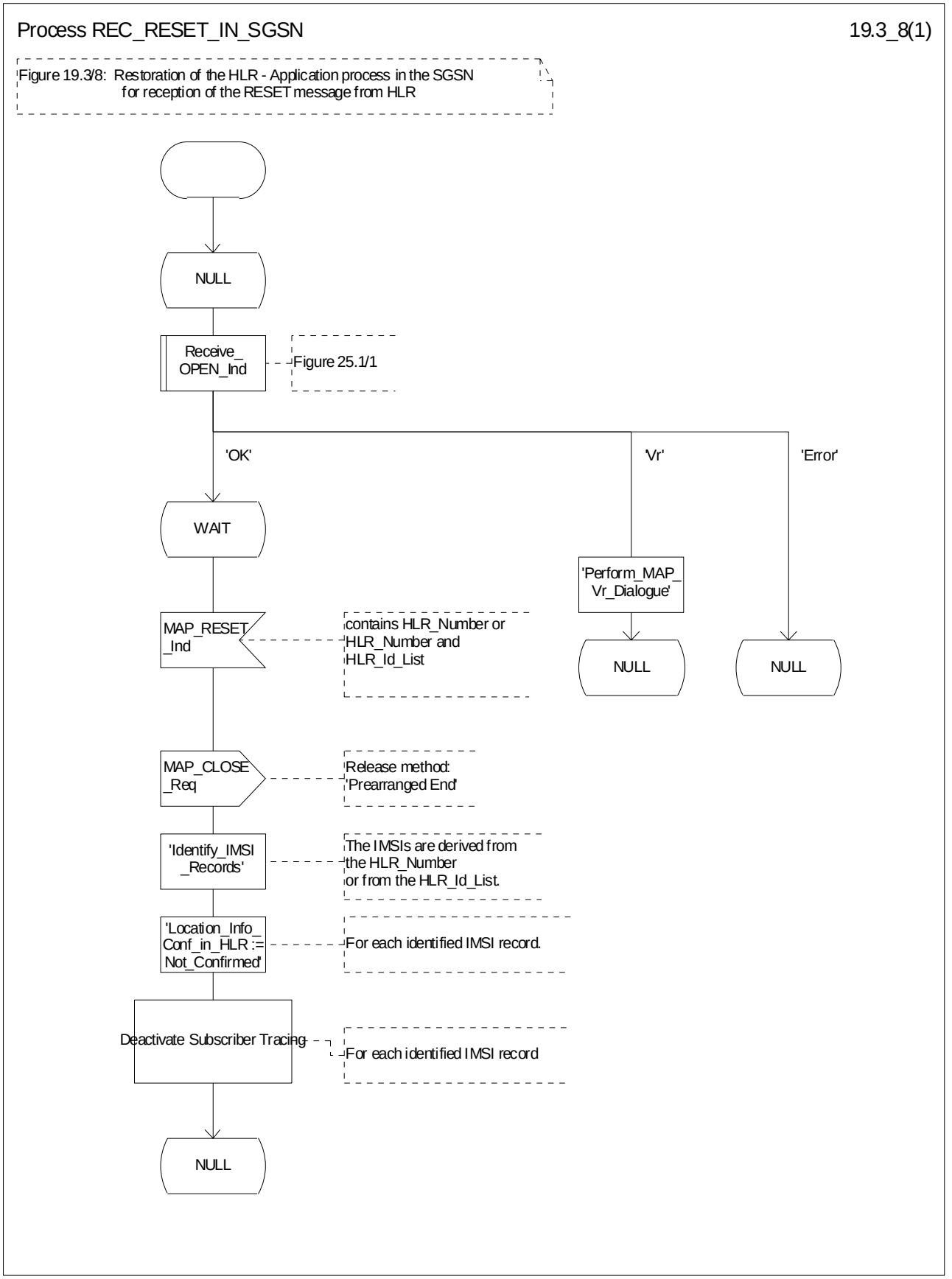


Figure 19.3/8: Process REC_RESET_IN_SGSN

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 clause; the corresponding procedure in the VLR (RESTORE_DATA_VLR) is described in clause 21.2.4.

The process RESTORE_DATA_HLR makes use of the following macros:

- Receive_Open_Ind clause 25.1.1;
- Check_Indication clause 25.2.1;
- Insert_Subs_Data_Framed_HLR clause 19.4.

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 clause 25.9. Thereafter, the macro "Insert_Subs_Data_Framed_HLR" that is described in clause 19.4 is invoked. The outcome of the macro Insert_Subs_Data_Framed_HLR is one of:

- abort, in which case the process terminates;
- error, in which case the HLR returns the error "System Failure" in the MAP_RESTORE_DATA response, and the process terminates;
- OK, indicating successful outcome of downloading the subscriber data to the VLR.

After successful completion of the framed MAP_INSERT_SUBSCRIBER_DATA procedure, the HLR Number and, if applicable, the "MS Not Reachable Flag" which is used for SMS, are provided in the MAP_RESTORE_DATA response.

Upon receipt of the MAP_RESTORE_DATA confirmation, the VLR behaves as described in clause 21.2.4, figure 21.2/6.

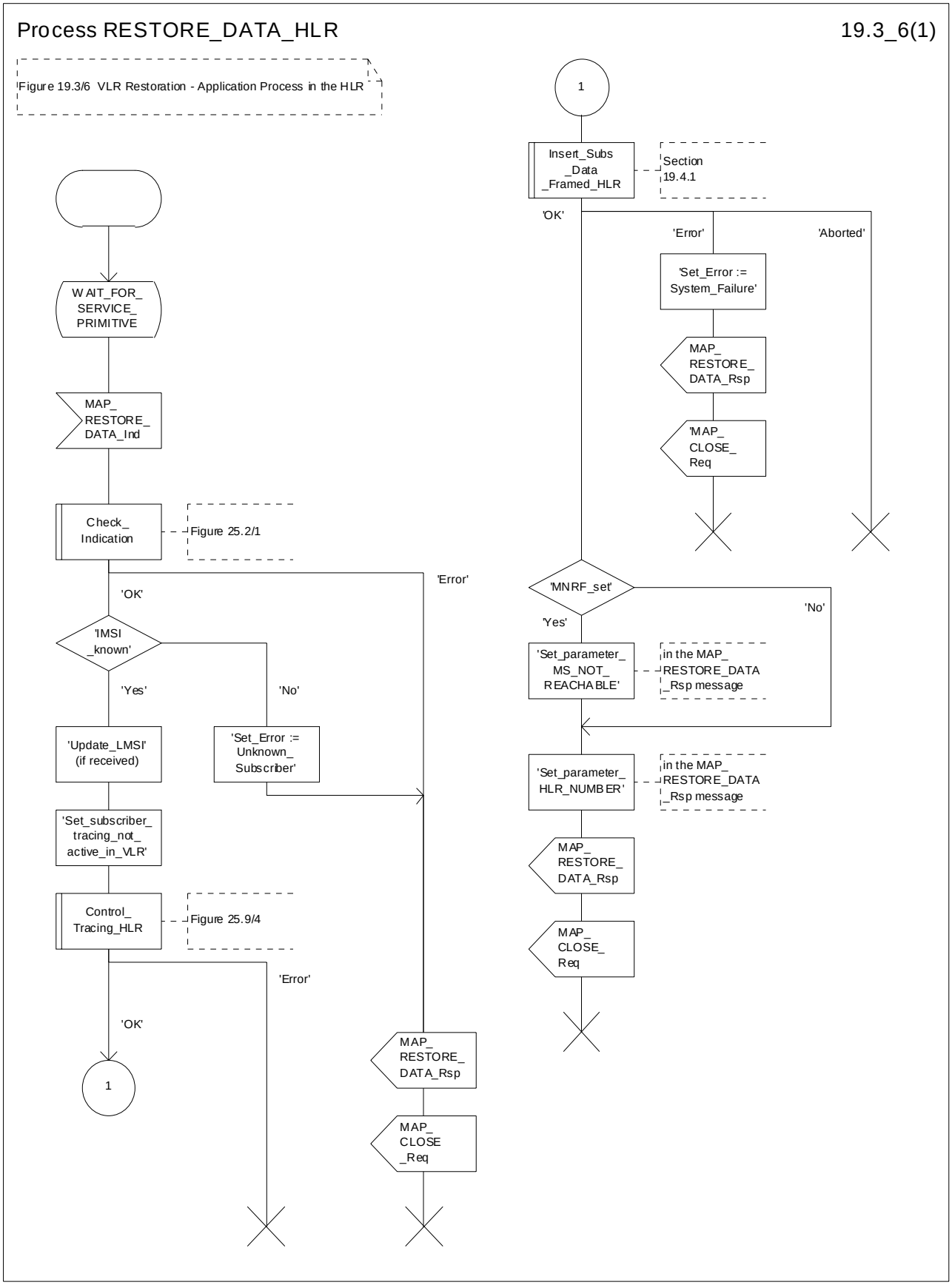


Figure 19.3/6: Process RESTORE_DATA_HLR

19.4 Macro Insert_Subs_Data_Framed_HLR

This macro is used by any procedure invoked in HLR which requires the transfer of subscriber data by means of the InsertSubscriberData operation (e.g. Update Location or Restore Data).

The invocation of the operation is done in a dialogue already opened by the framing procedure. Therefore the latter is the one that handles the reception of the open indication and sends the dialogue close request.

The macro calls the process "Send_Insert_Subs_Data" (see clause 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 clause 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 3GPP 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.

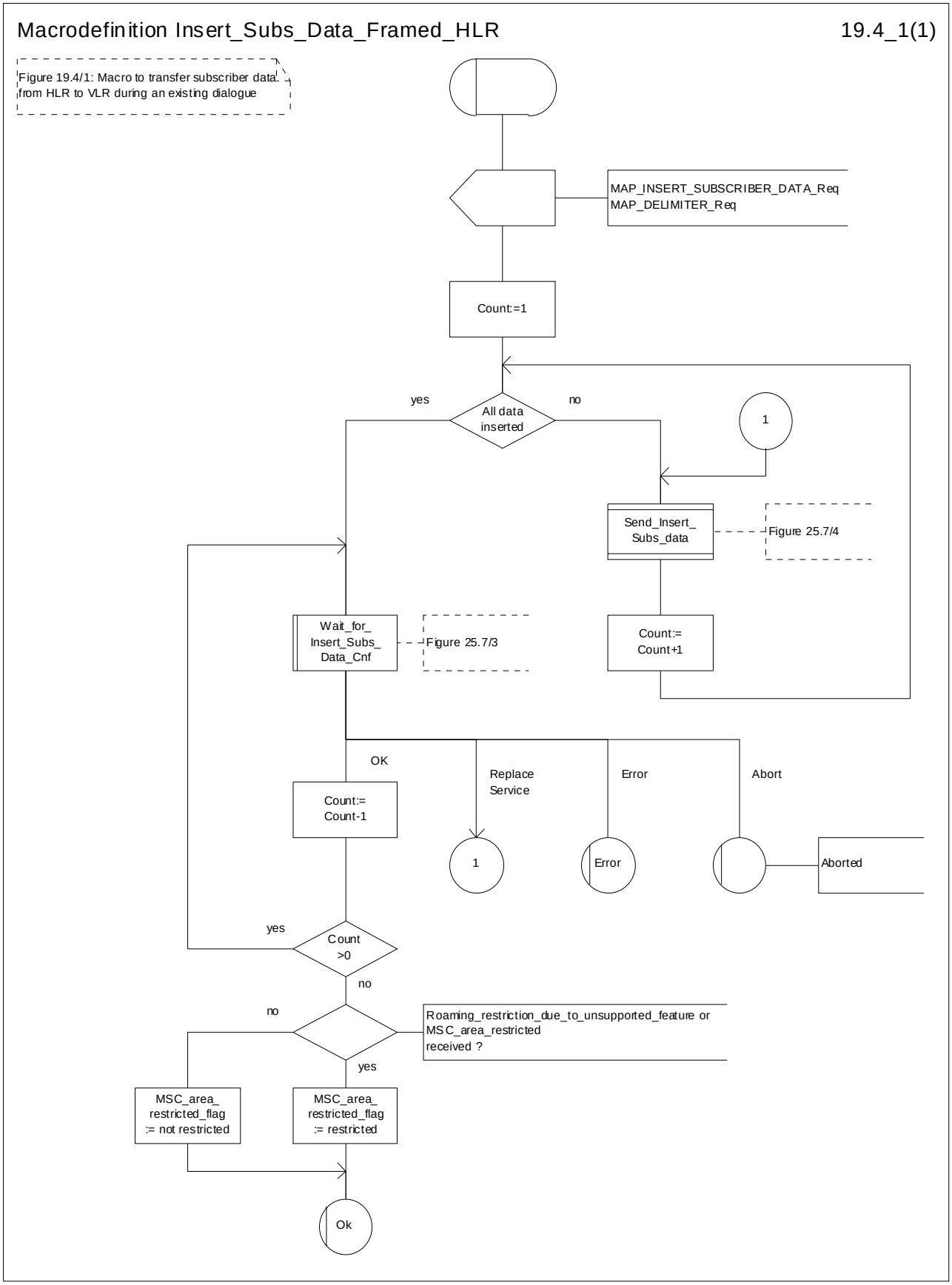


Figure 19.4/1: Macro Insert_Subs_Data_Framed_HLR

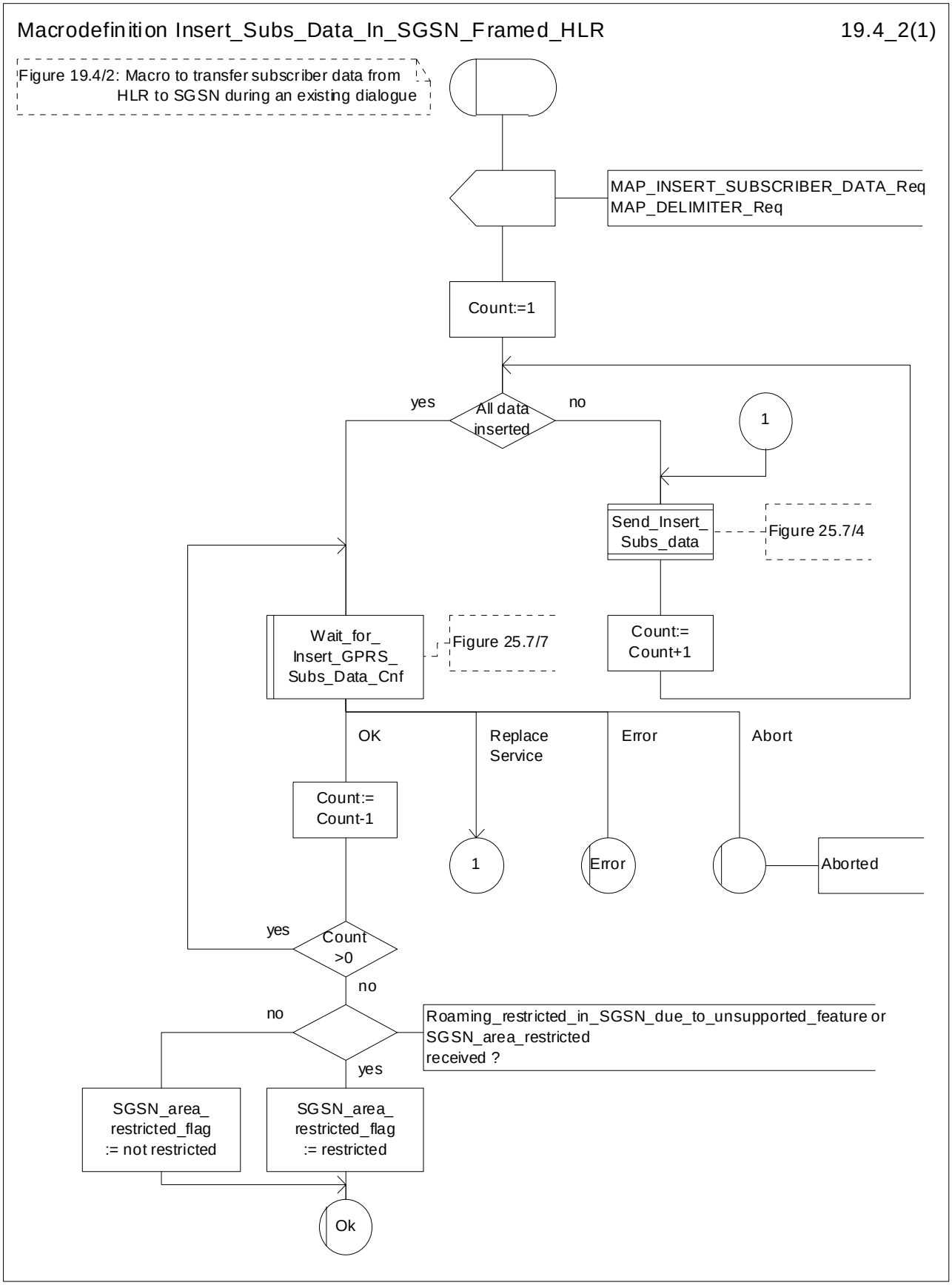


Figure 19.4/2: Macro Insert_Subs_Data_In_SGSN_Framed_HLR

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.#

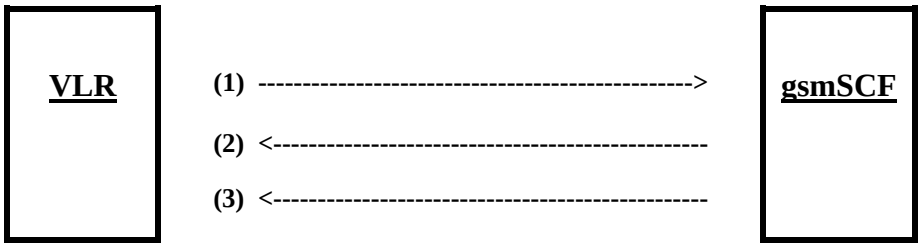


Figure 19.5/1: Interfaces and MAP Messages for Mobility Management Event notifications

(1) Report Mobility Management Event (VLR to gsmSCF).

(2) Report Mobility Management Event Result (gsmSCF to VLR).

(3) Report Mobility Management Event Error (gsmSCF to VLR).

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.

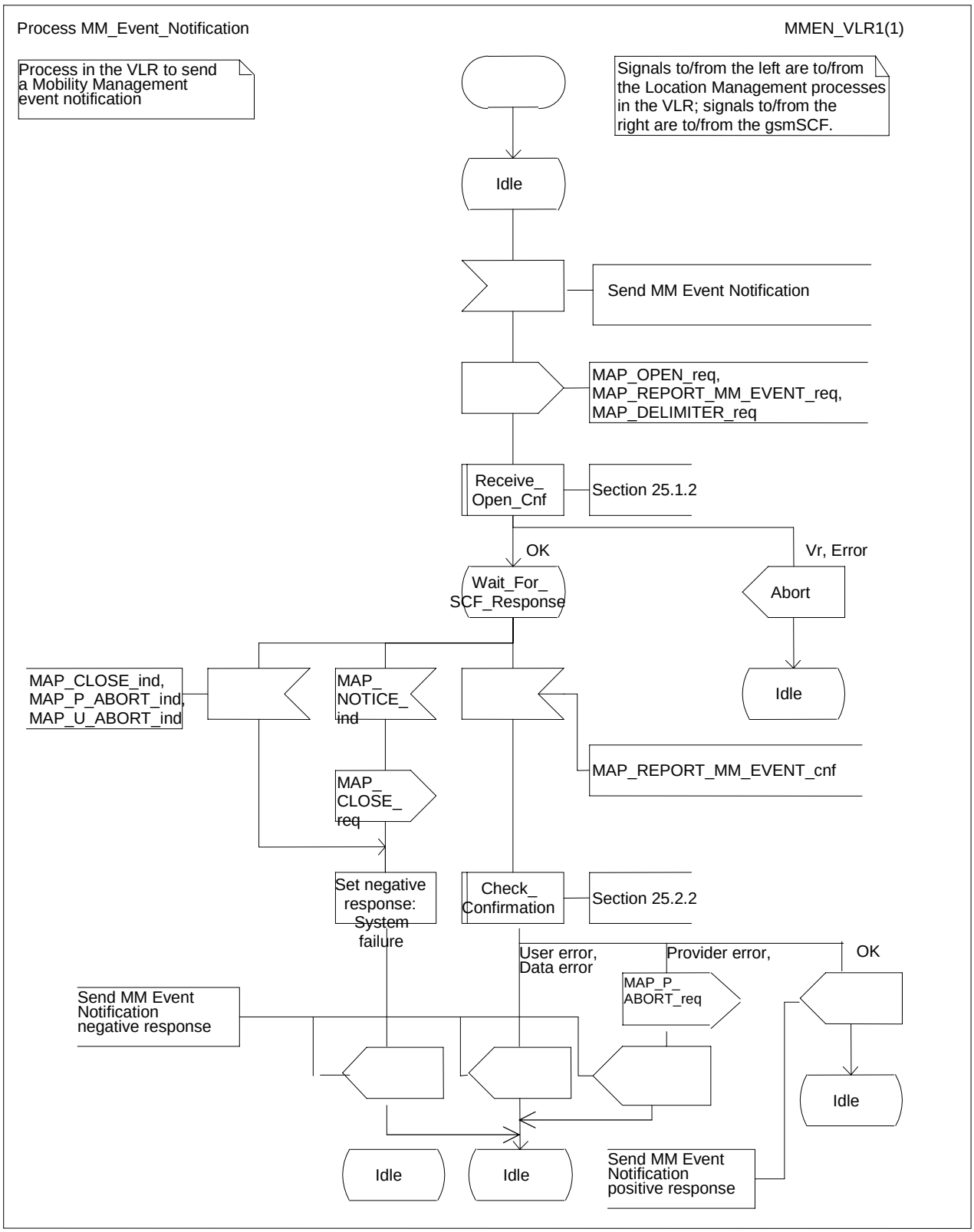


Figure 19.5/2: Process MM_Event_Notification_VLR (sheet 1 of 1)

19.5.3 Process in the gsmSCF

When the gsmSCF receives the ReportMMEvent 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 ReportMMEvent 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.

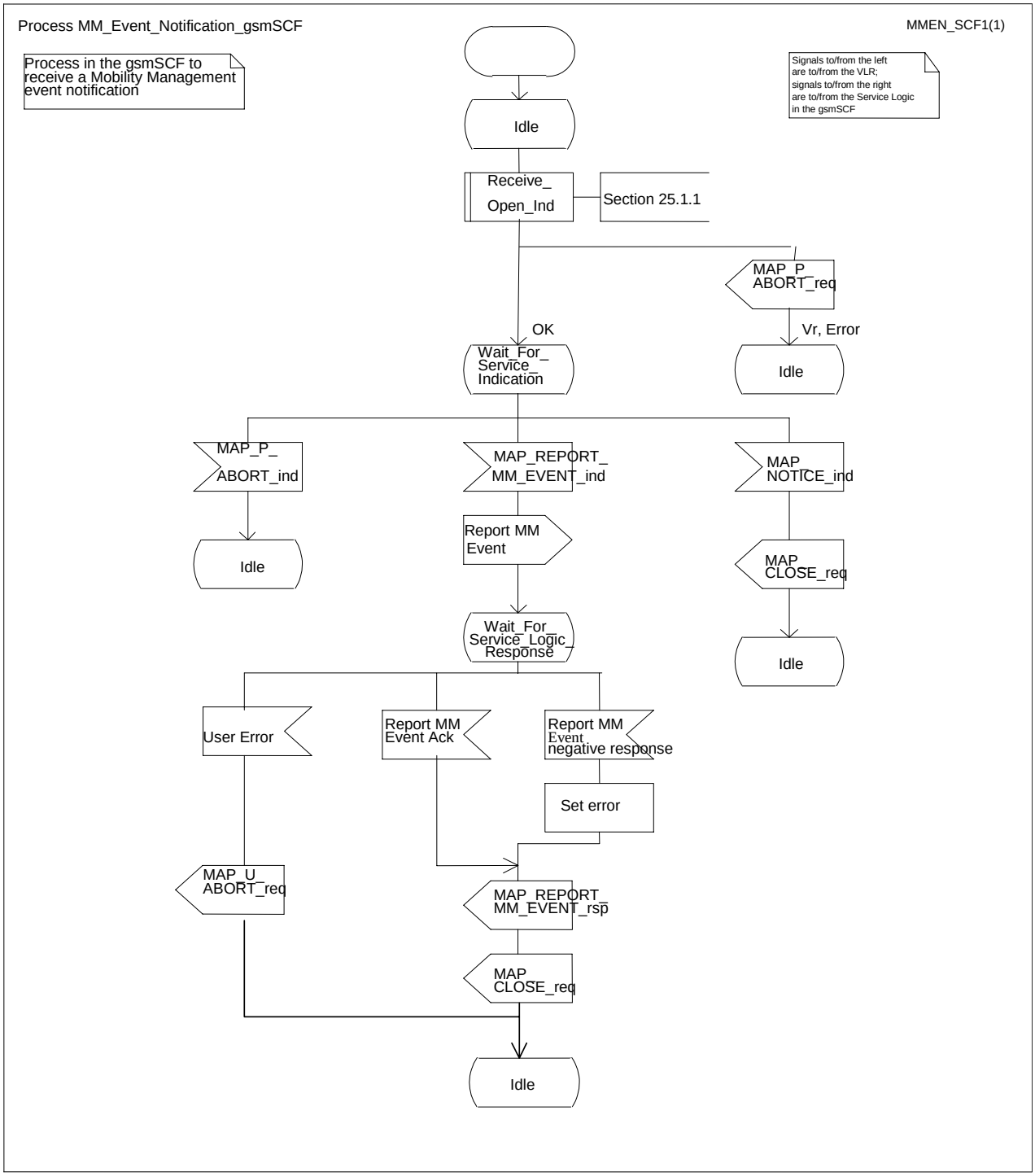


Figure 19.5/3: Process MM_Event_Notification_gsmSCF (sheet 1 of 1)

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 clauses.

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.

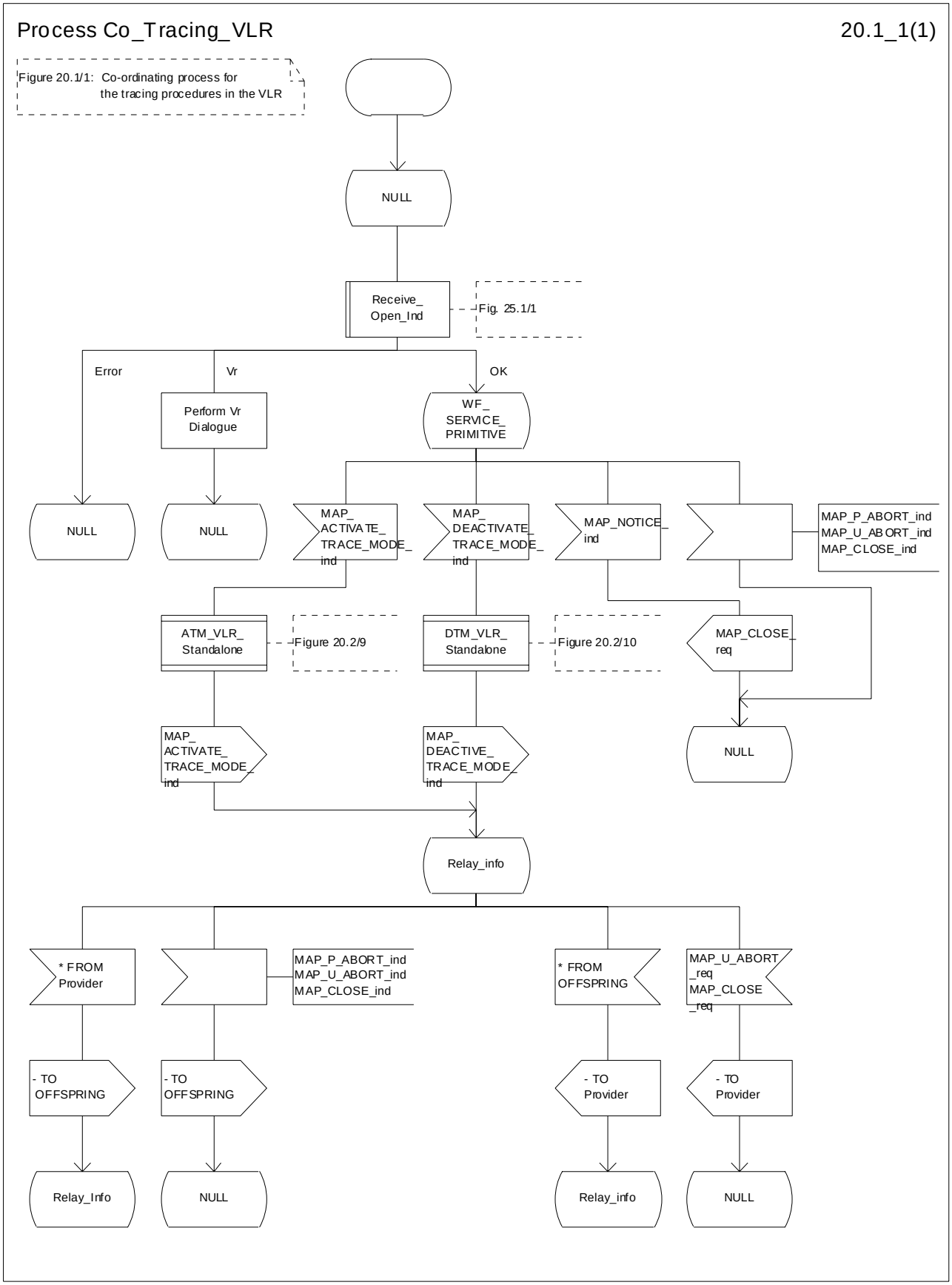


Figure 20.1/1: Process Co_Tracing_VLR

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.

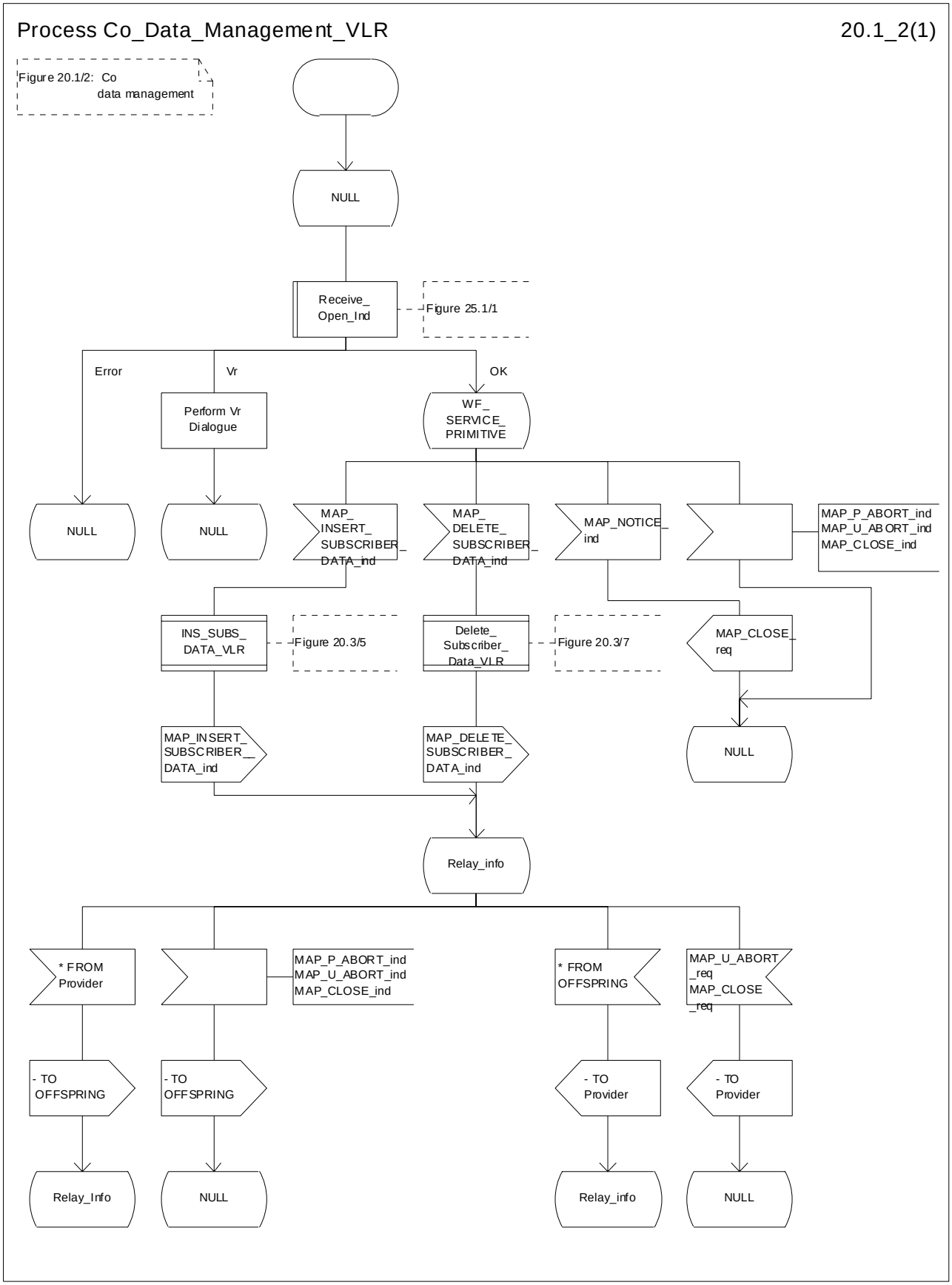


Figure 20.1/2: Process Co_Data_Management_VLR

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.

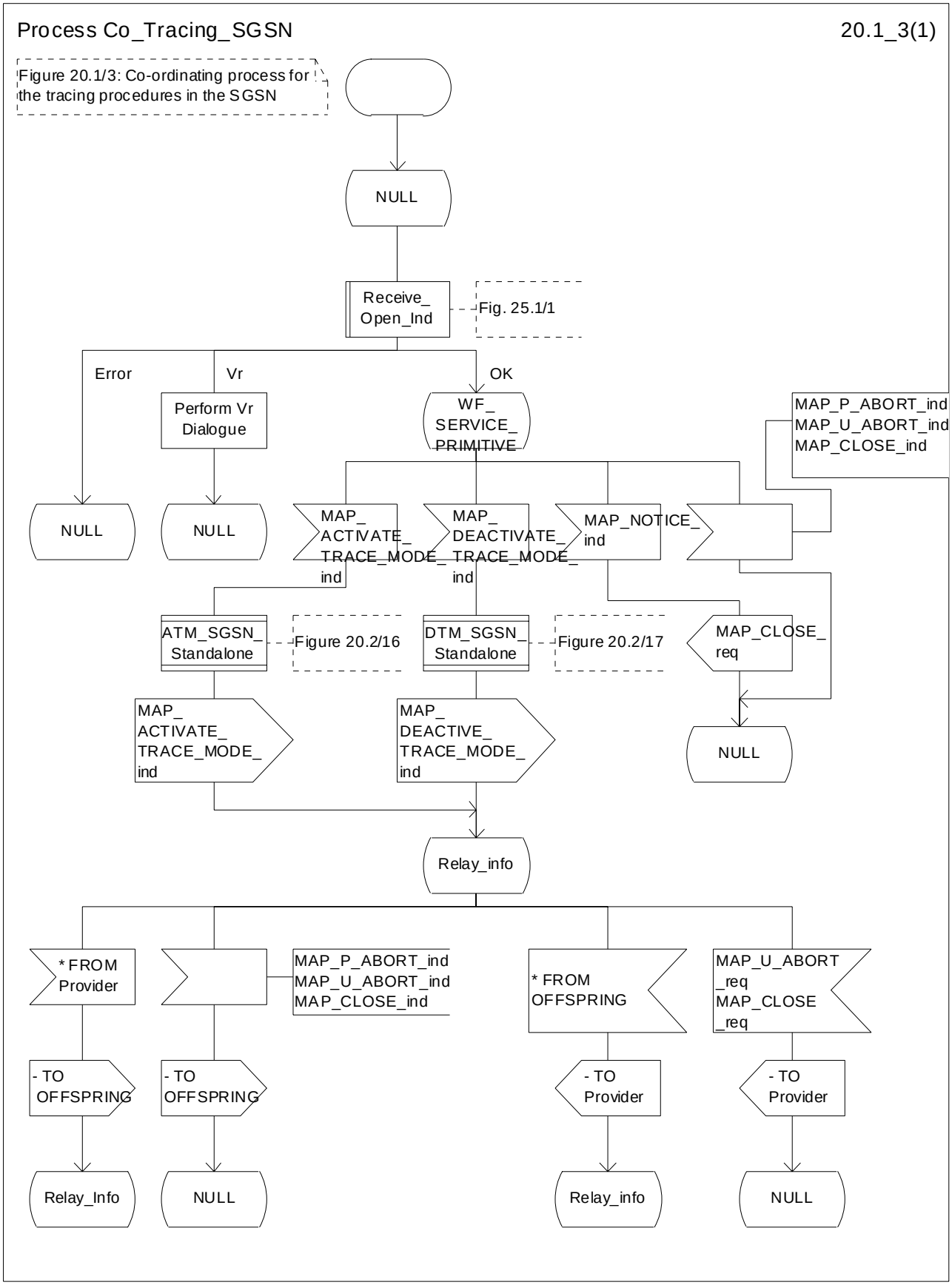


Figure 20.1/3: Process Co_Tracing_SGSN

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.

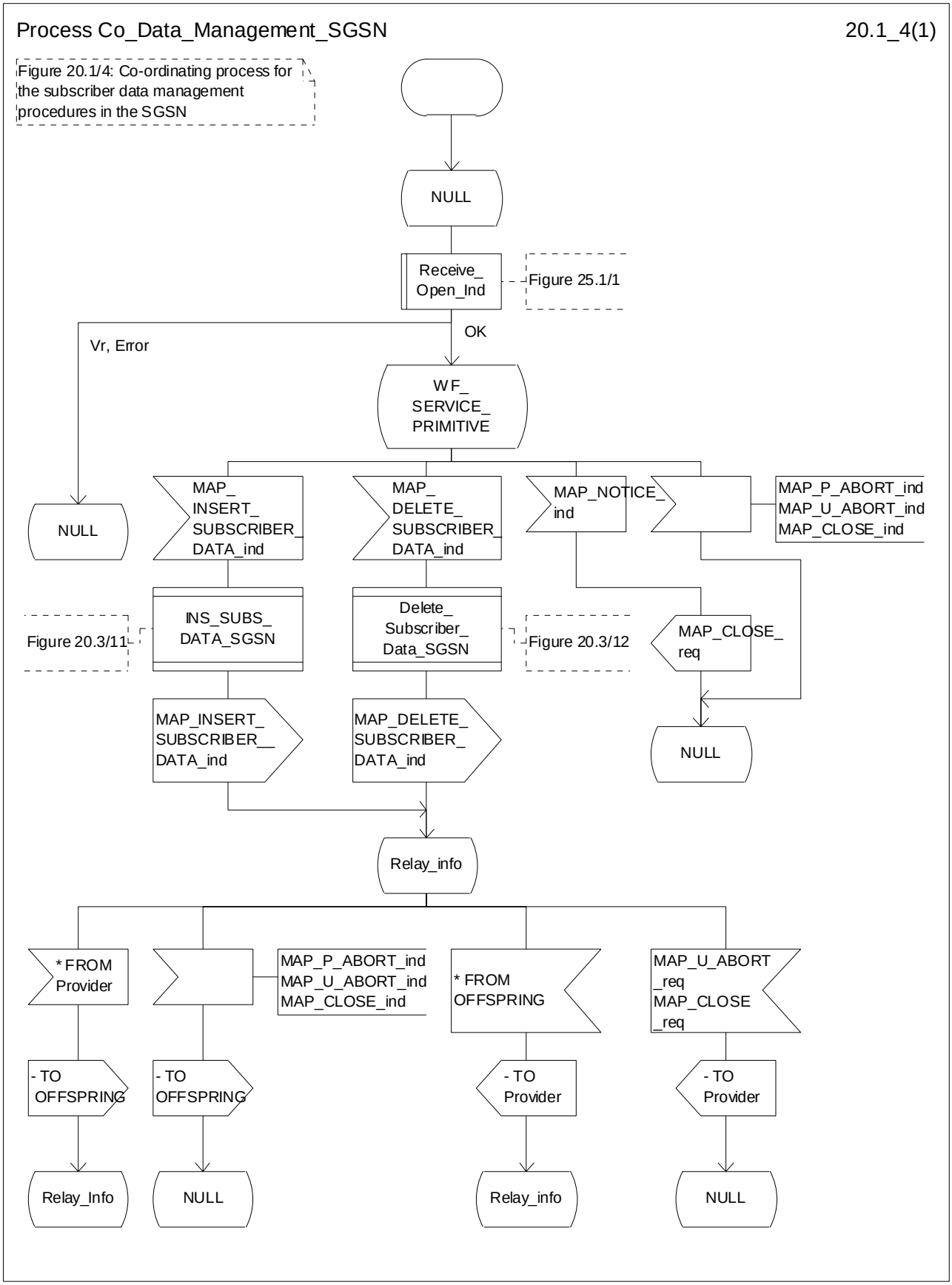


Figure 20.1/4: Process Co_Data_Management_SGSN

20.2 Tracing procedures

Three types 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.

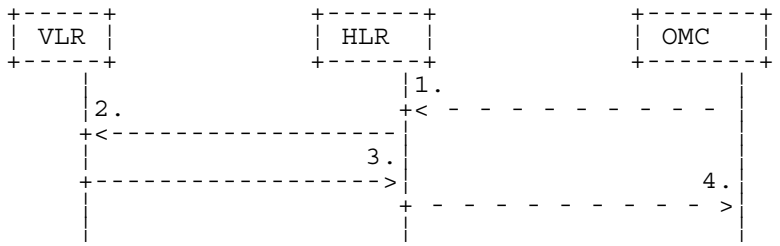


Figure 20.2/1: Stand alone subscriber tracing activation procedure

1) Subscriber Tracing Activation.

2) MAP_ACTIVATE_TRACE_MODE.

3) MAP_ACTIVATE_TRACE_MODE_ACK.

4) Subscriber Tracing Activation Accepted.

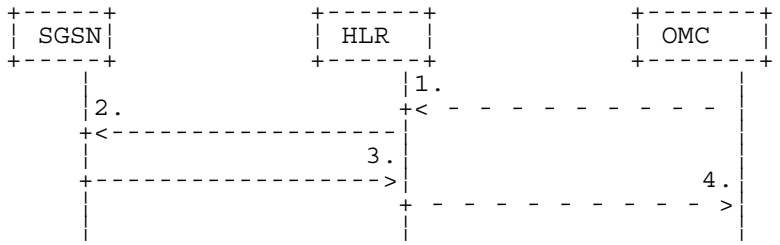


Figure 20.2/11: Stand alone subscriber tracing activation procedure for GPRS

1) Subscriber Tracing Activation.

2) MAP_ACTIVATE_TRACE_MODE.

3) MAP_ACTIVATE_TRACE_MODE_ACK.

4) Subscriber Tracing Activation Accepted.

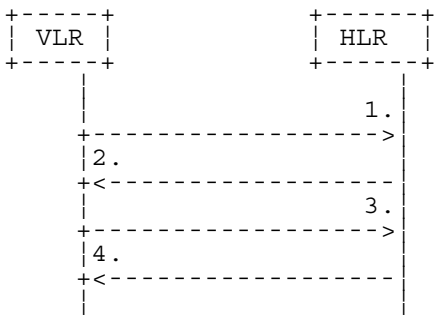


Figure 20.2/2: Subscriber tracing activation procedure at location updating or data restoration

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.

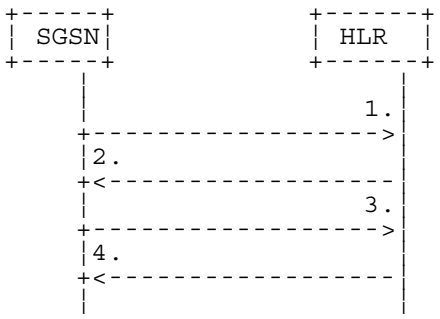


Figure 20.2/12: Subscriber tracing activation procedure at GPRS location updating

1) MAP_UPDATE_GPRS_LOCATION.

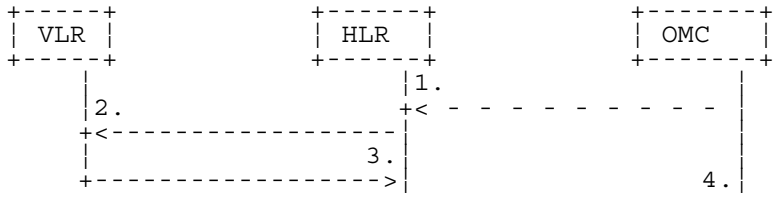
2) MAP_ACTIVATE_TRACE_MODE.

3) MAP_ACTIVATE_TRACE_MODE_ACK.

4) MAP_UPDATE_GPRS_LOCATION_ACK.

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.



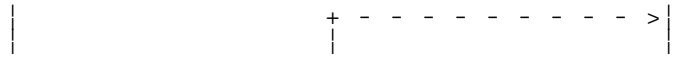


Figure 20.2/3: Subscriber tracing deactivation procedure

1) Subscriber Tracing Deactivation.

2) MAP_DEACTIVATE_TRACE_MODE.

3) MAP_DEACTIVATE_TRACE_MODE_ACK.

4) Subscriber Tracing Deactivation Accepted.

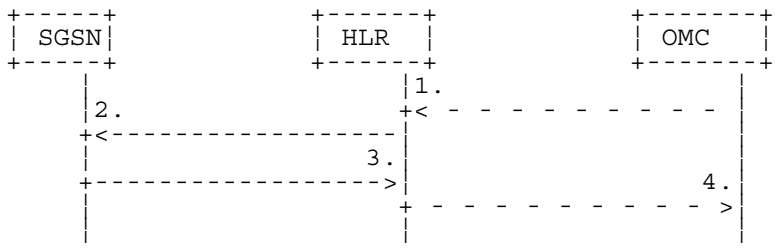


Figure 20.2/13: Subscriber tracing deactivation procedure for GPRS

1) Subscriber Tracing Deactivation.

2) MAP_DEACTIVATE_TRACE_MODE.

3) MAP_DEACTIVATE_TRACE_MODE_ACK.

4) Subscriber Tracing Deactivation Accepted.

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.

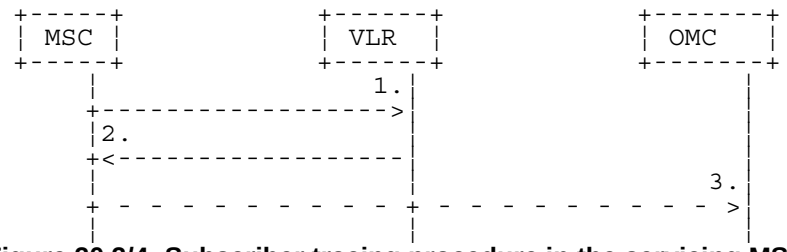


Figure 20.2/4: Subscriber tracing procedure in the servicing MSC

1) MAP_PROCESS_ACCESS_REQUEST, MAP_UPDATE_LOCATION_AREA.

2) MAP_TRACE_SUBSCRIBER_ACTIVITY.

3) Subscriber tracing information.

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.

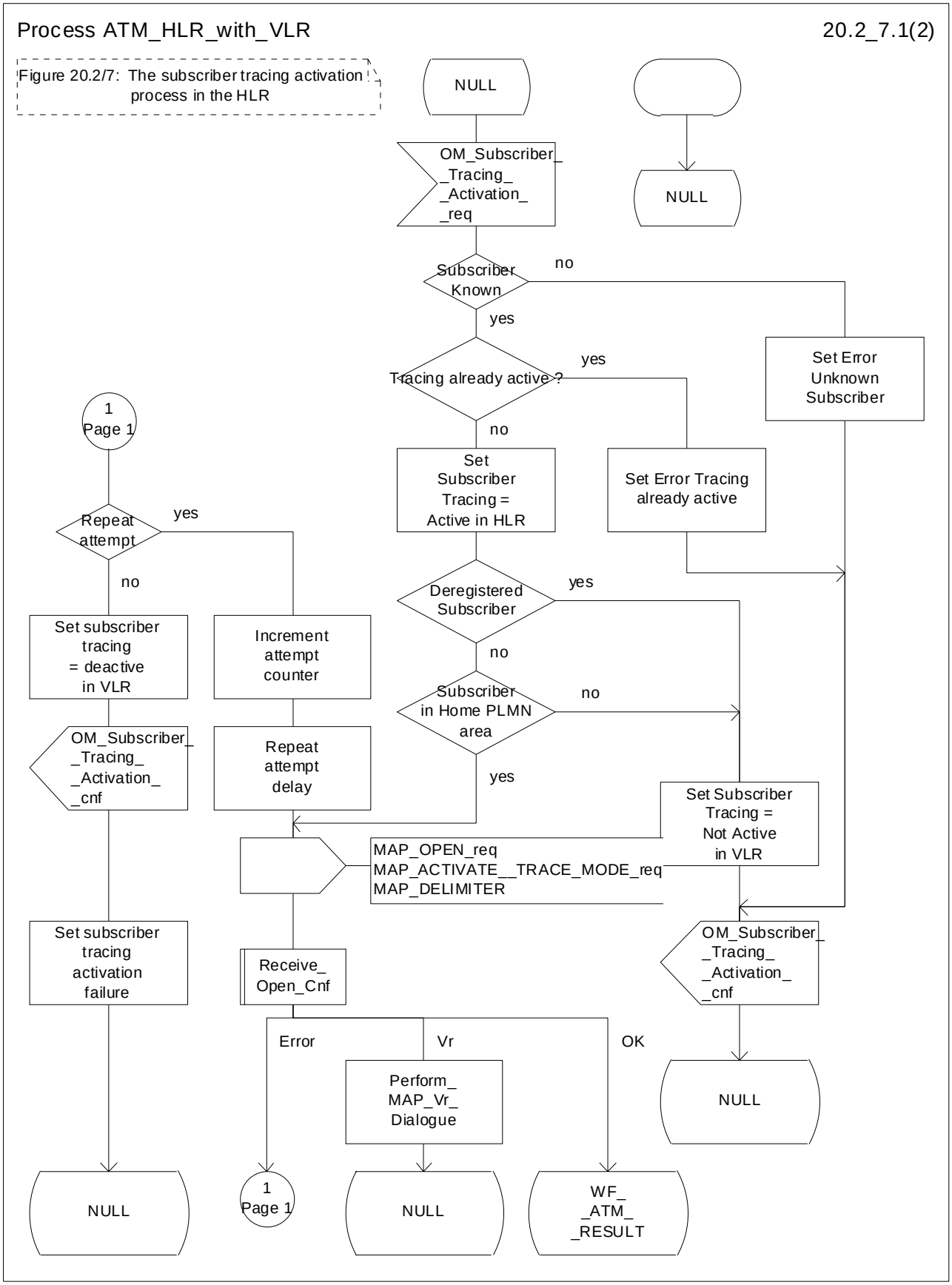


Figure 20.2/7 (sheet 1 of 2): Process ATM_HLR_with_VLR

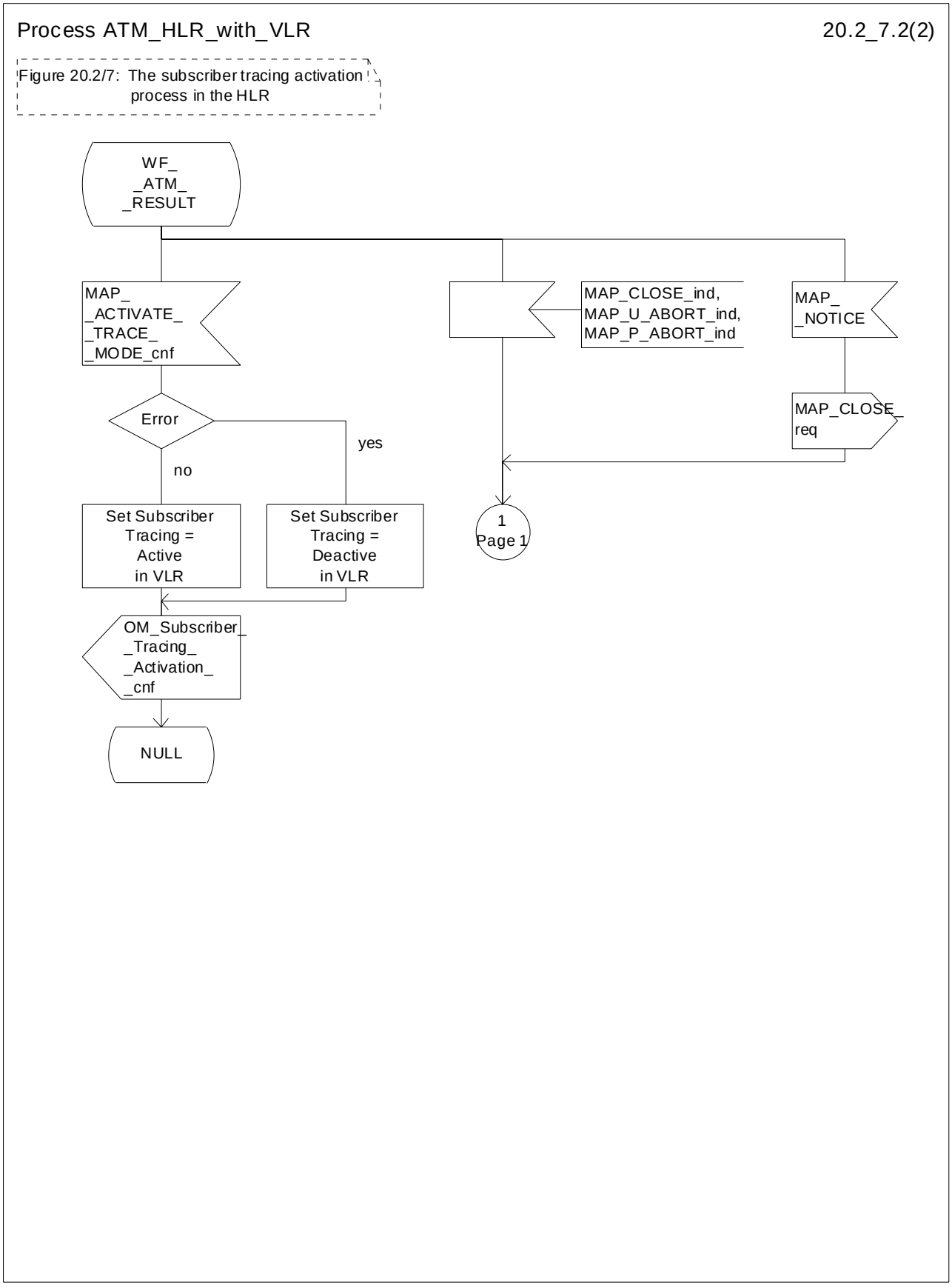


Figure 20.2/7 (sheet 2 of 2): Process ATM_HLR_with_VLR

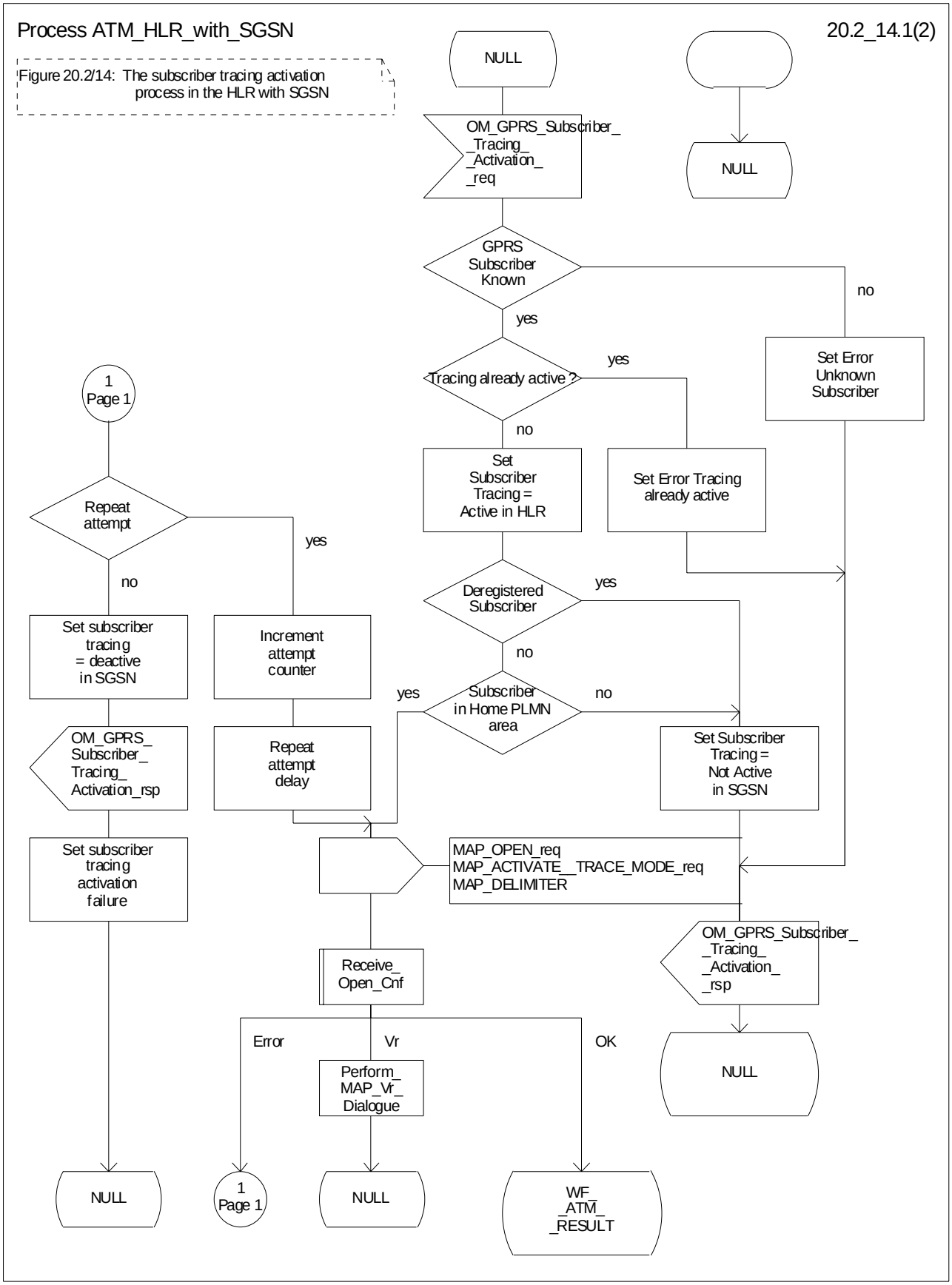


Figure 20.2/14 (sheet 1 of 2): Process ATM_HLR_with_SGSN

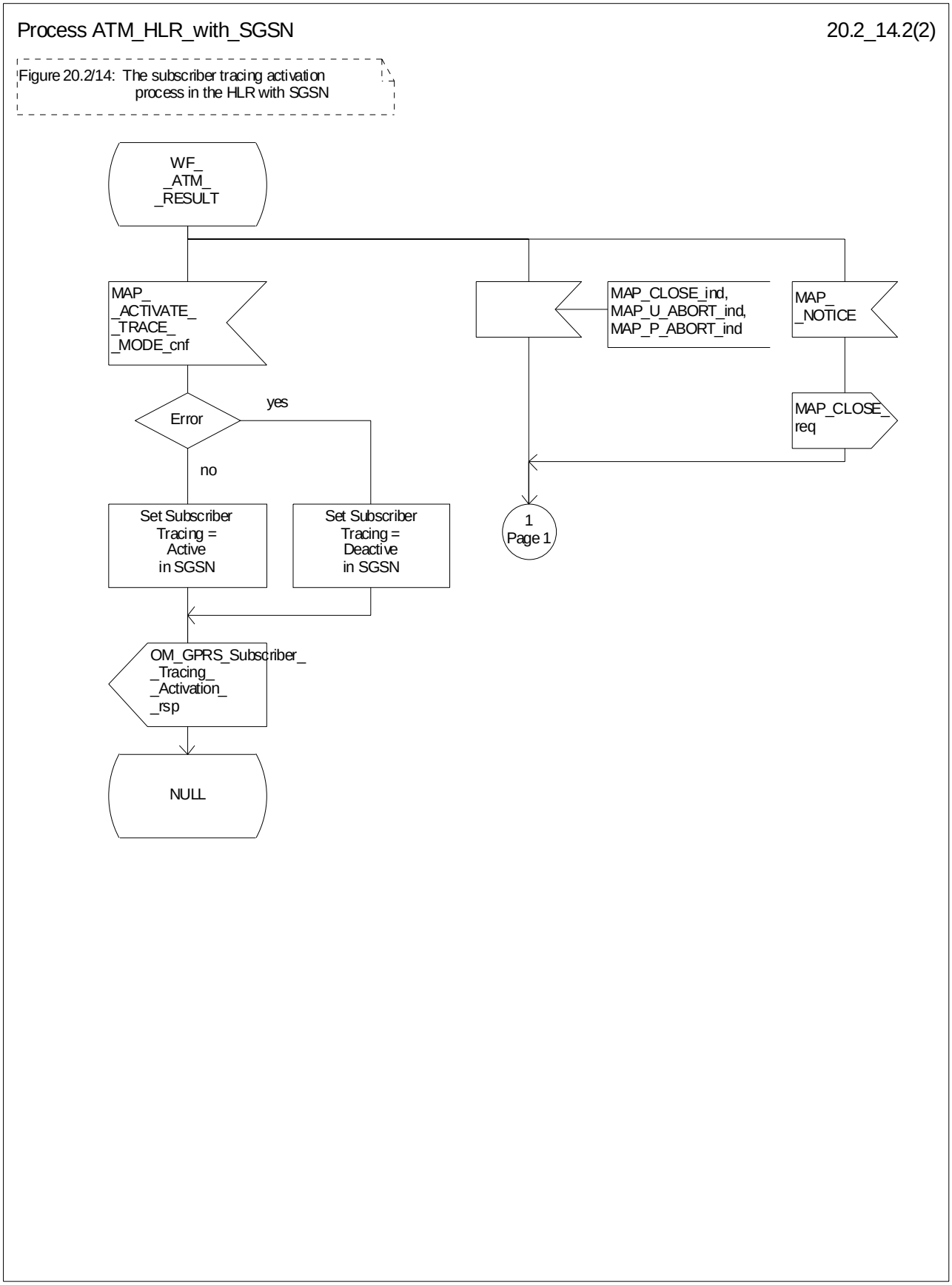


Figure 20.2/14 (sheet 2 of 2): Process ATM_HLR_with_SGSN

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.

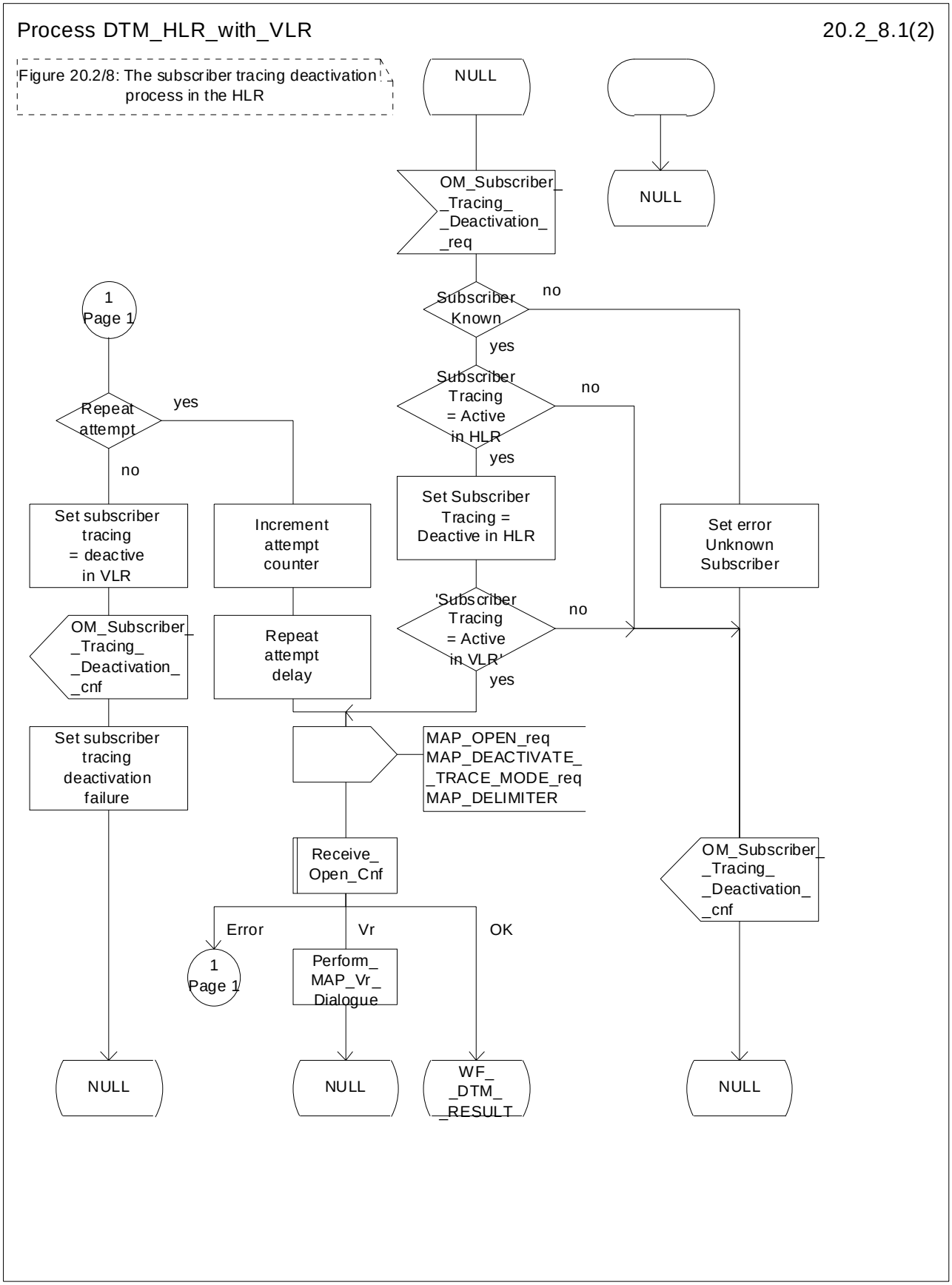


Figure 20.2/8 (sheet 1 of 2): Process DTM_HLR_with_VLR

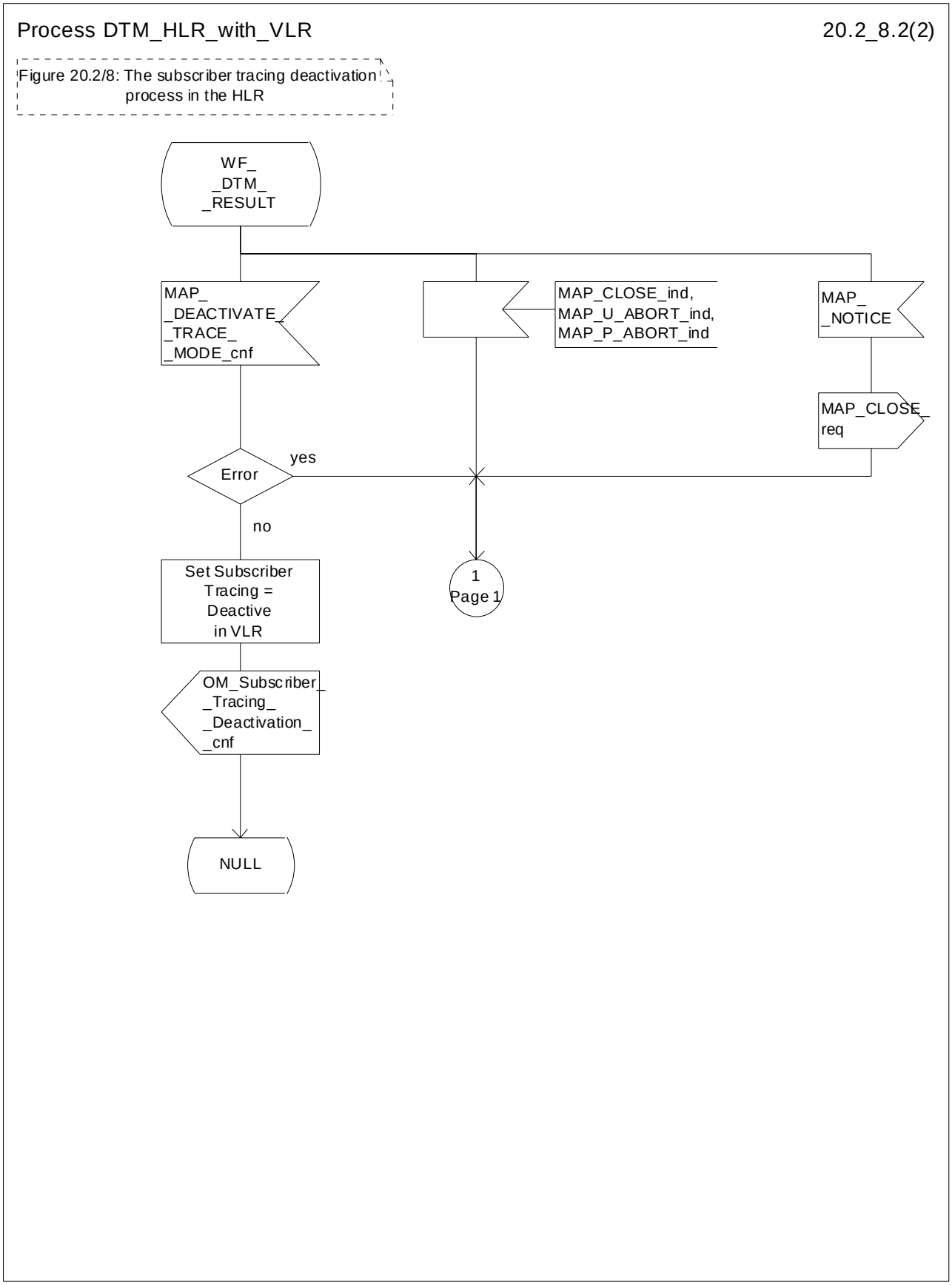


Figure 20.2/8 (sheet 2 of 2): Process DTM_HLR_with_VLR

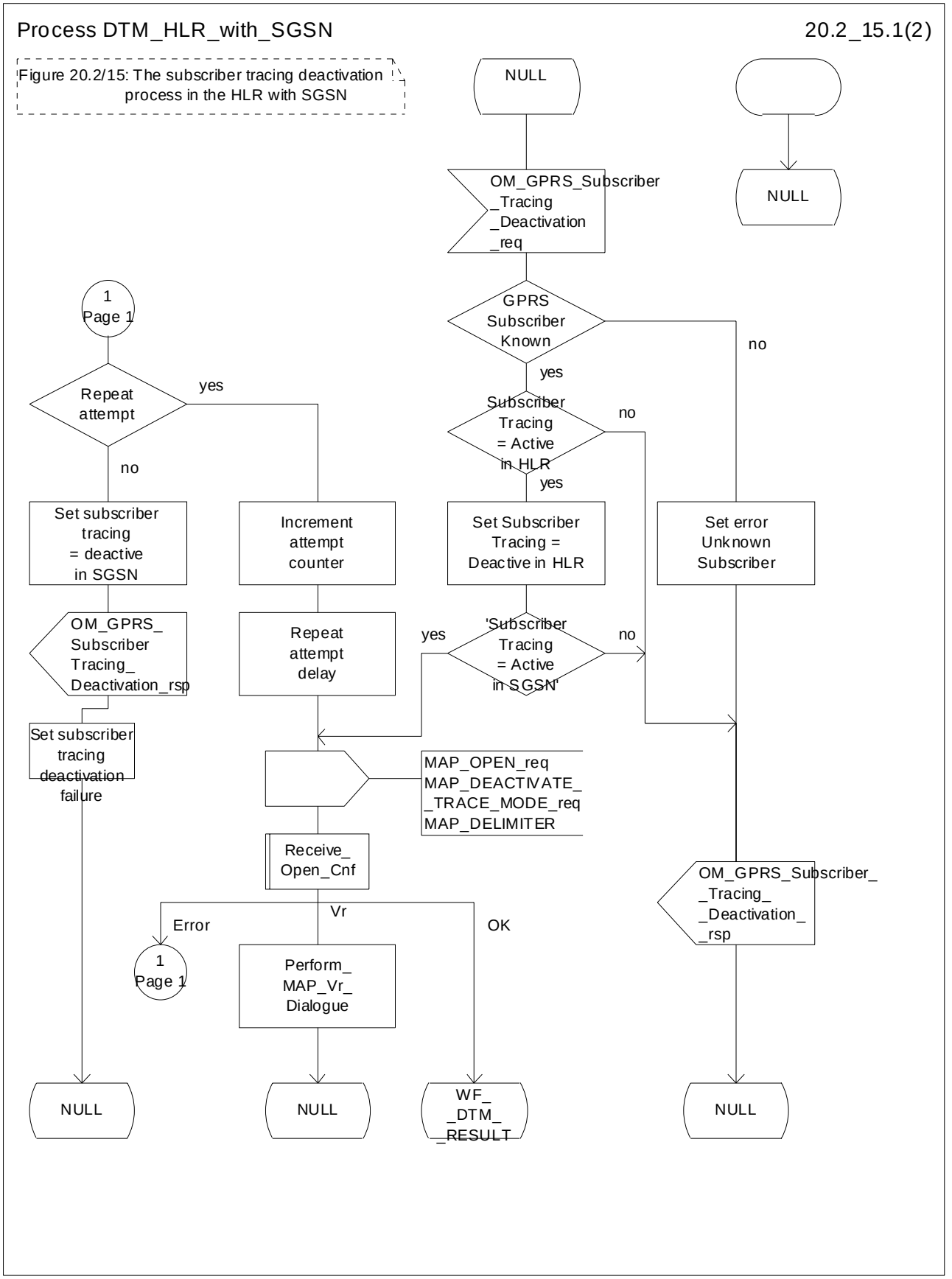


Figure 20.2/15 (sheet 1 of 2): Process DTM_HLR_with_SGSN

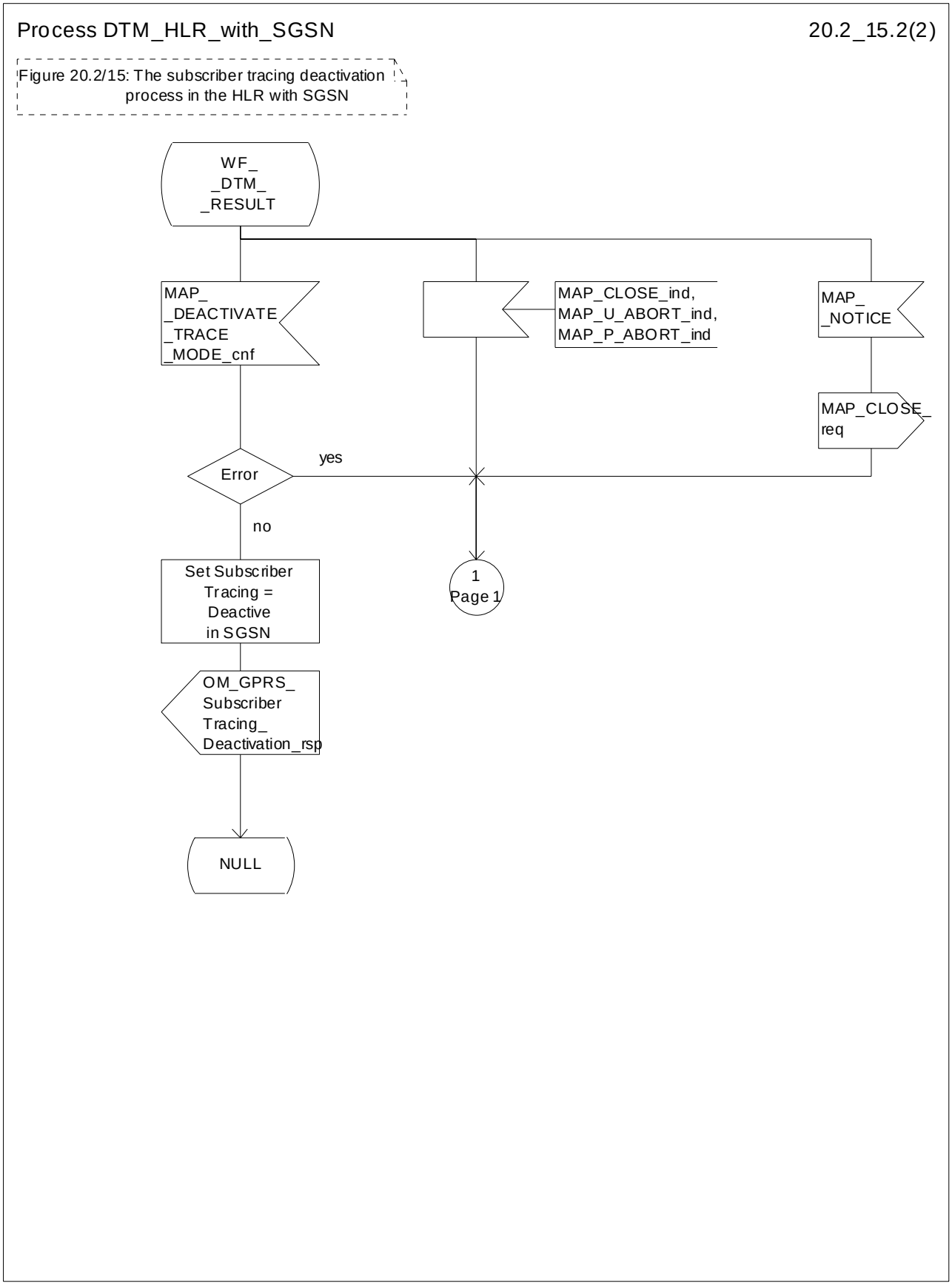


Figure 20.2/15 (sheet 2 of 2): Process DTM_HLR_with_SGSN

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.

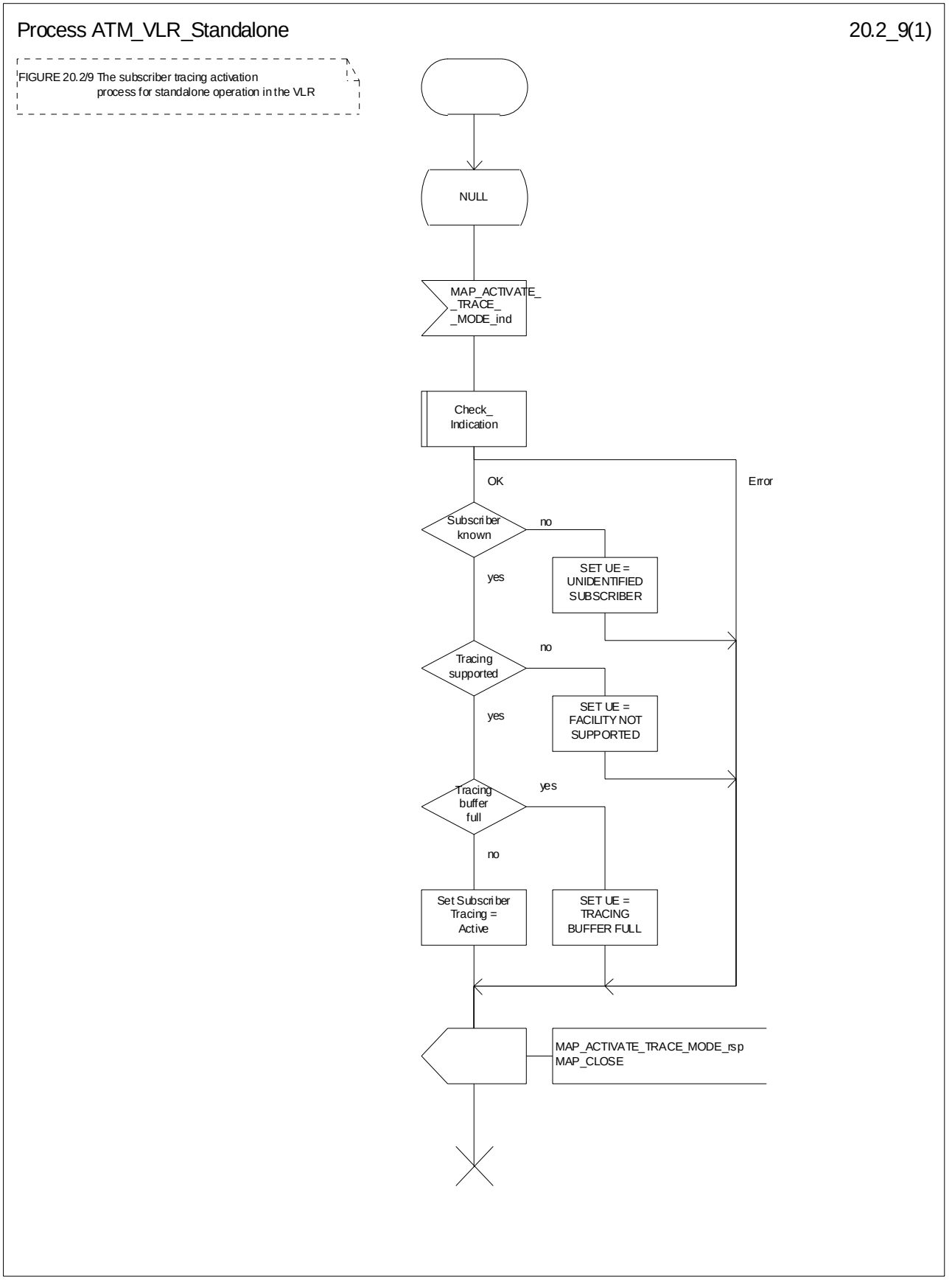


Figure 20.2/9: Process ATM_VLR_Standalone

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

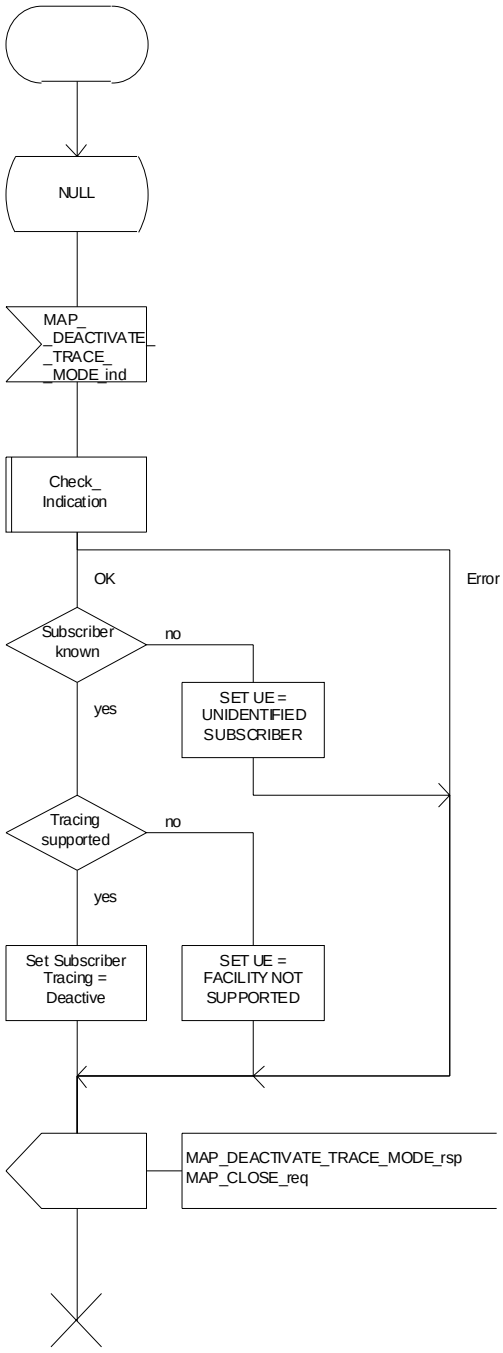


Figure 20.2/10: Process DTM_VLR_Standalone

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.

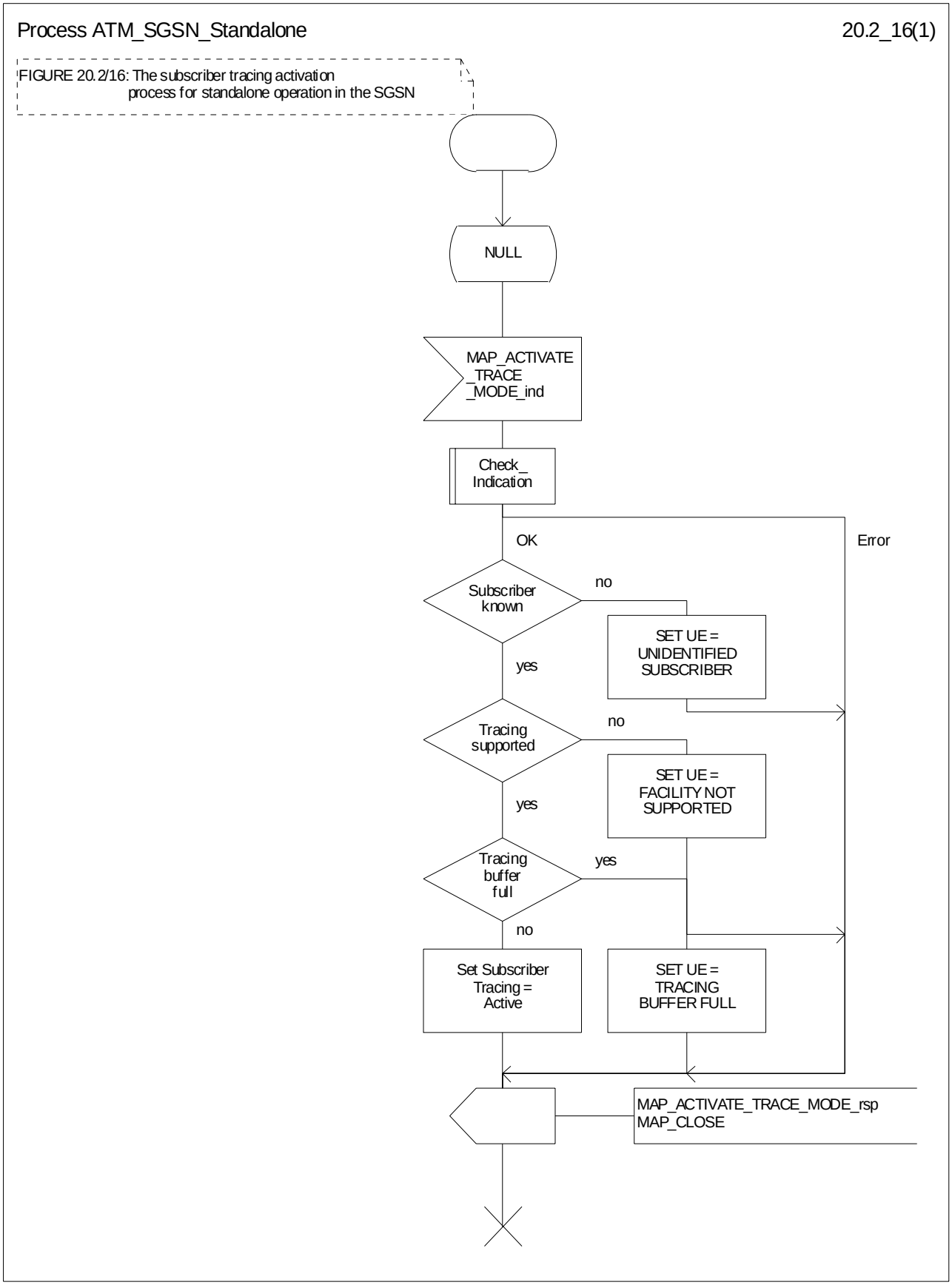


Figure 20.2/16: Process ATM_SGSN_Standalone

Process DTM_SGSN_Standalone 20.2_17(1)

Figure 20.2/17: The subscriber tracing deactivation process in the SGSN

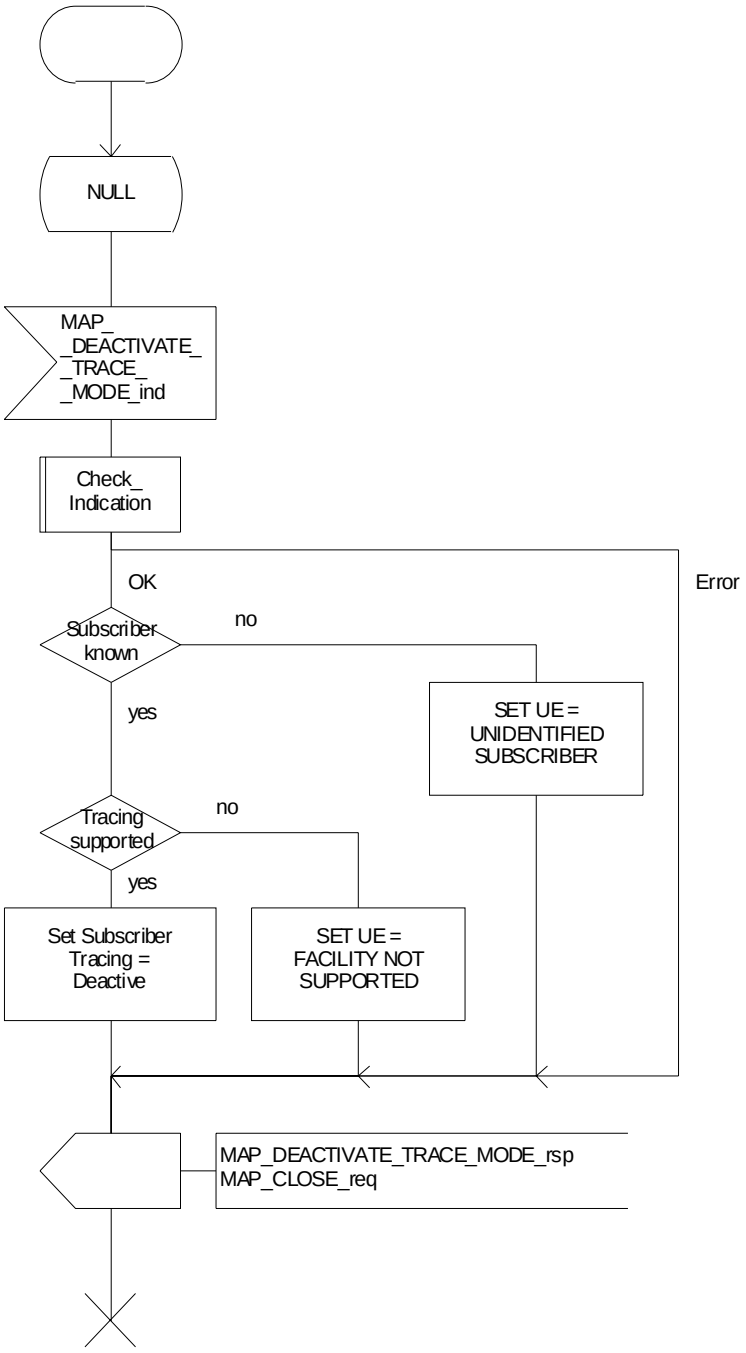


Figure 20.2/17: Process DTM_SGSN_Standalone

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).

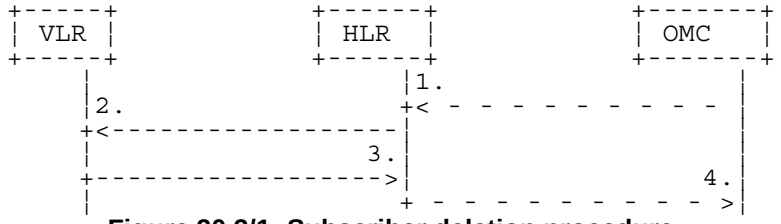


Figure 20.3/1: Subscriber deletion procedure

1) Delete Subscriber.

2) MAP_CANCEL_LOCATION.

3) MAP_CANCEL_LOCATION_ACK.

4) Subscriber Deleted.

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.

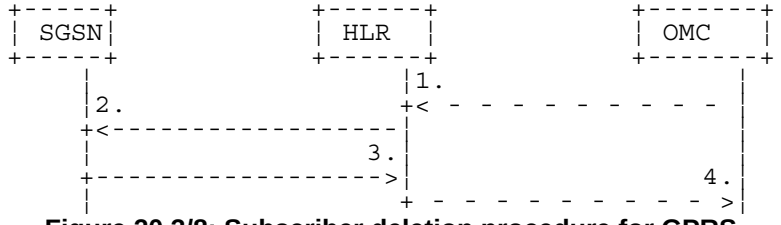


Figure 20.3/8: Subscriber deletion procedure for GPRS

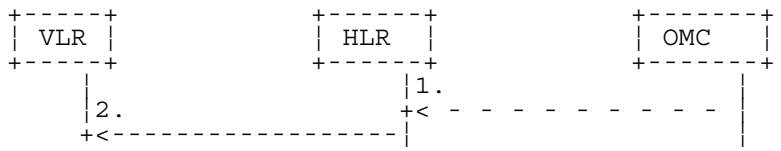
1) Delete GPRS Subscriber.

2) MAP_CANCEL_LOCATION.

3) MAP_CANCEL_LOCATION_ACK.

4) GPRS Subscriber Deleted.

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.



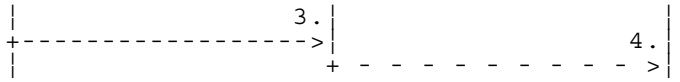


Figure 20.3/2: Subscriber data modification procedure

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.

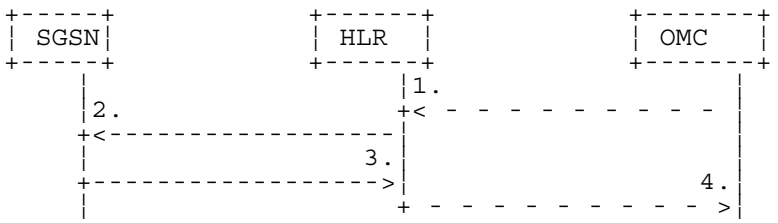


Figure 20.3/9: Subscriber data modification procedure for GPRS

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.

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.

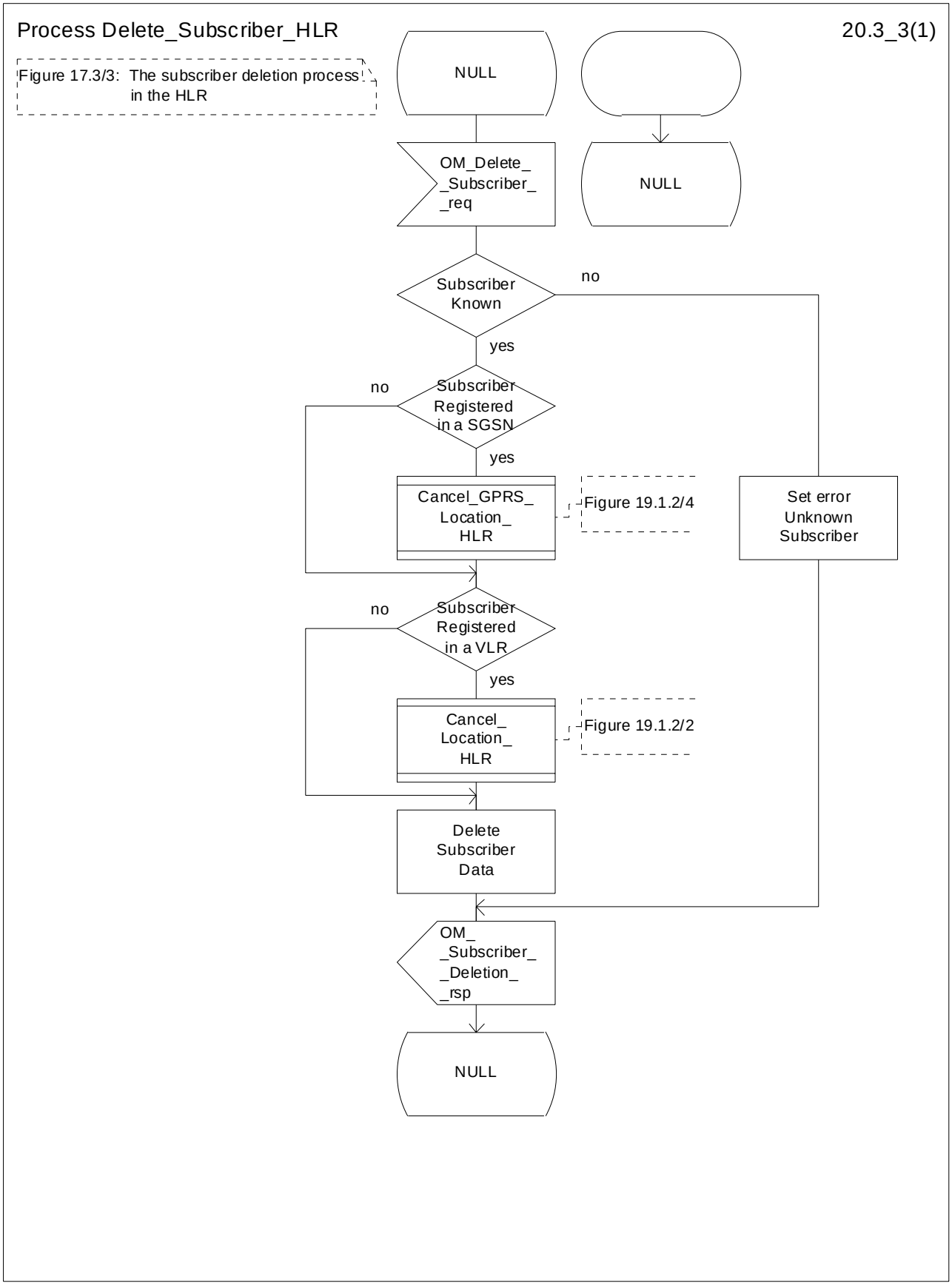


Figure 20.3/3: Process Delete_Subscriber_HLR

20.3.1.2 Subscriber data modification procedure

The OMC can modify the subscriber data in several different ways. The modifications can be categorised 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.

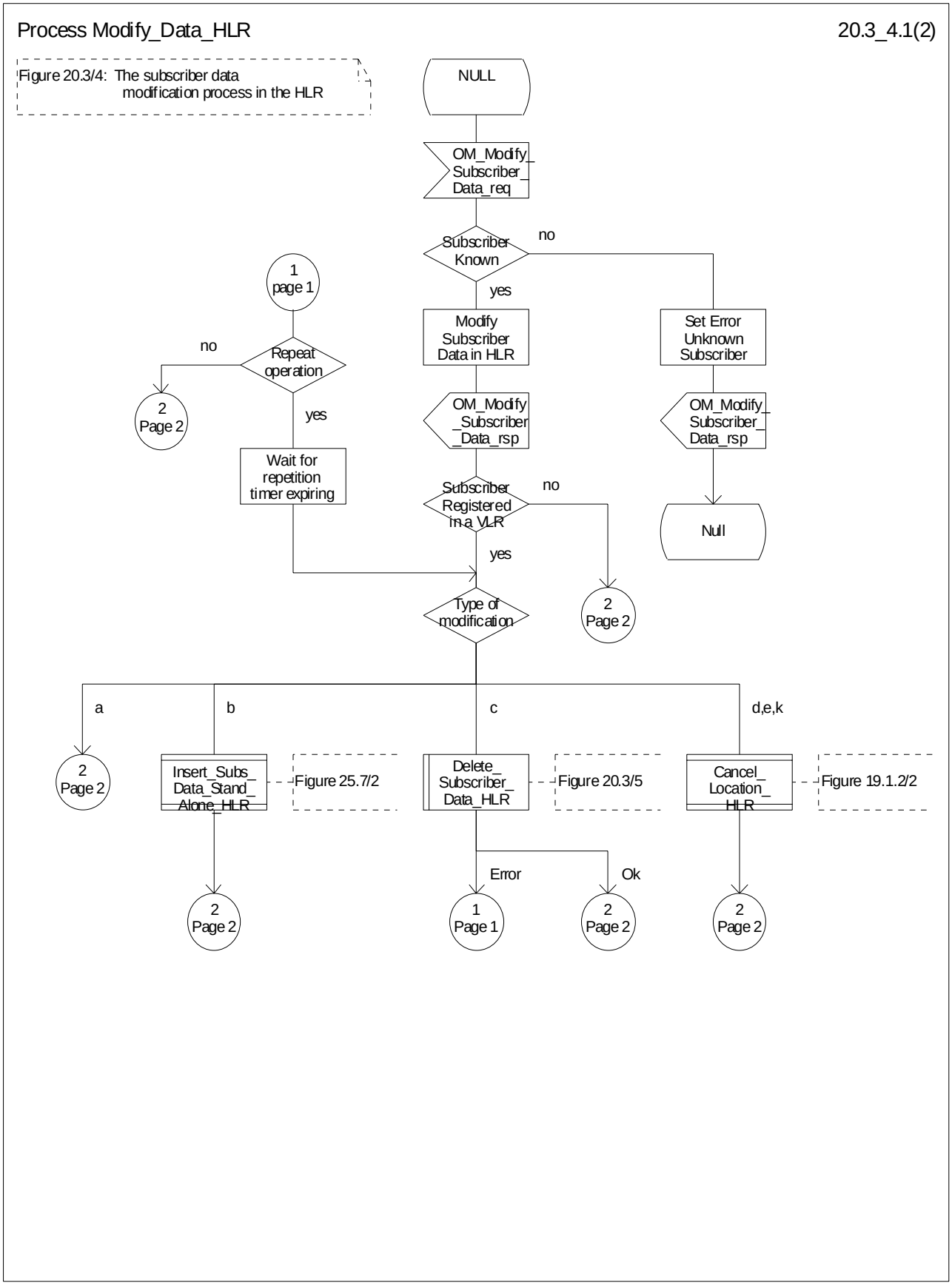


Figure 20.3/4 (sheet 1 of 2): Process Modify_Data_HLR

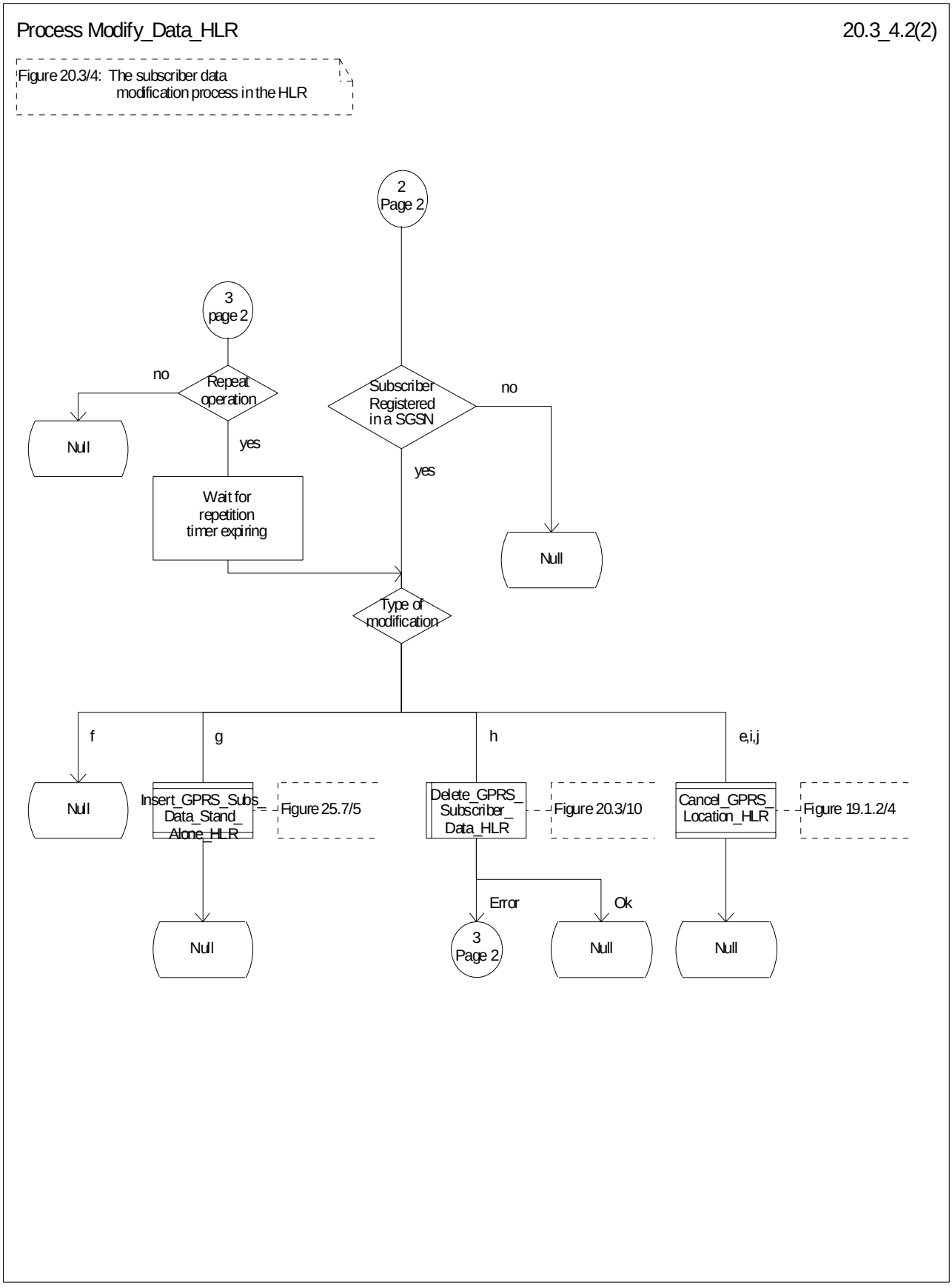


Figure 20.3/4 (sheet 2 of 2): Process Modify_Data_HLR

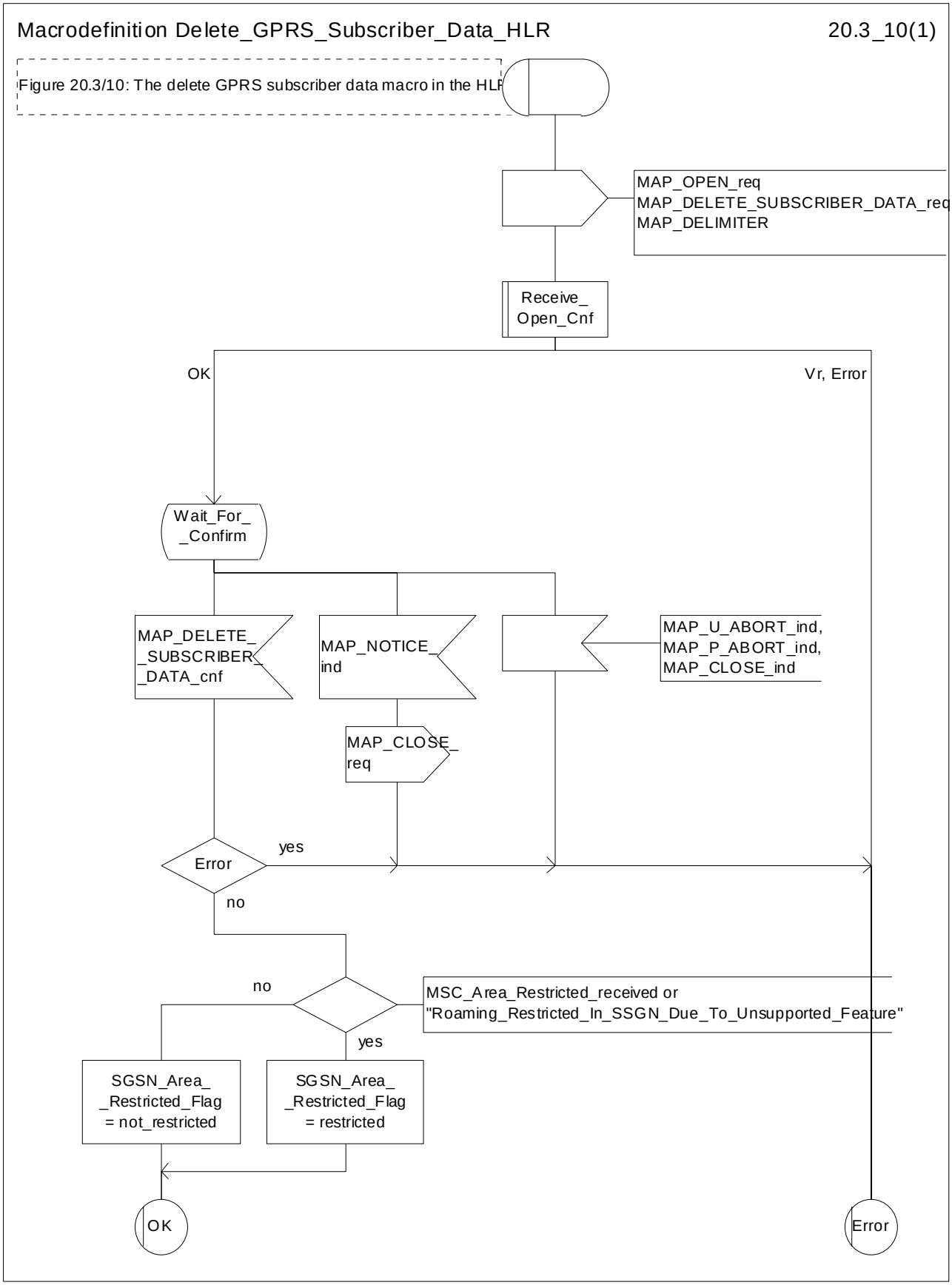


Figure 20.3/10: Macro Delete_GPRS_Subscriber_Data_HLR

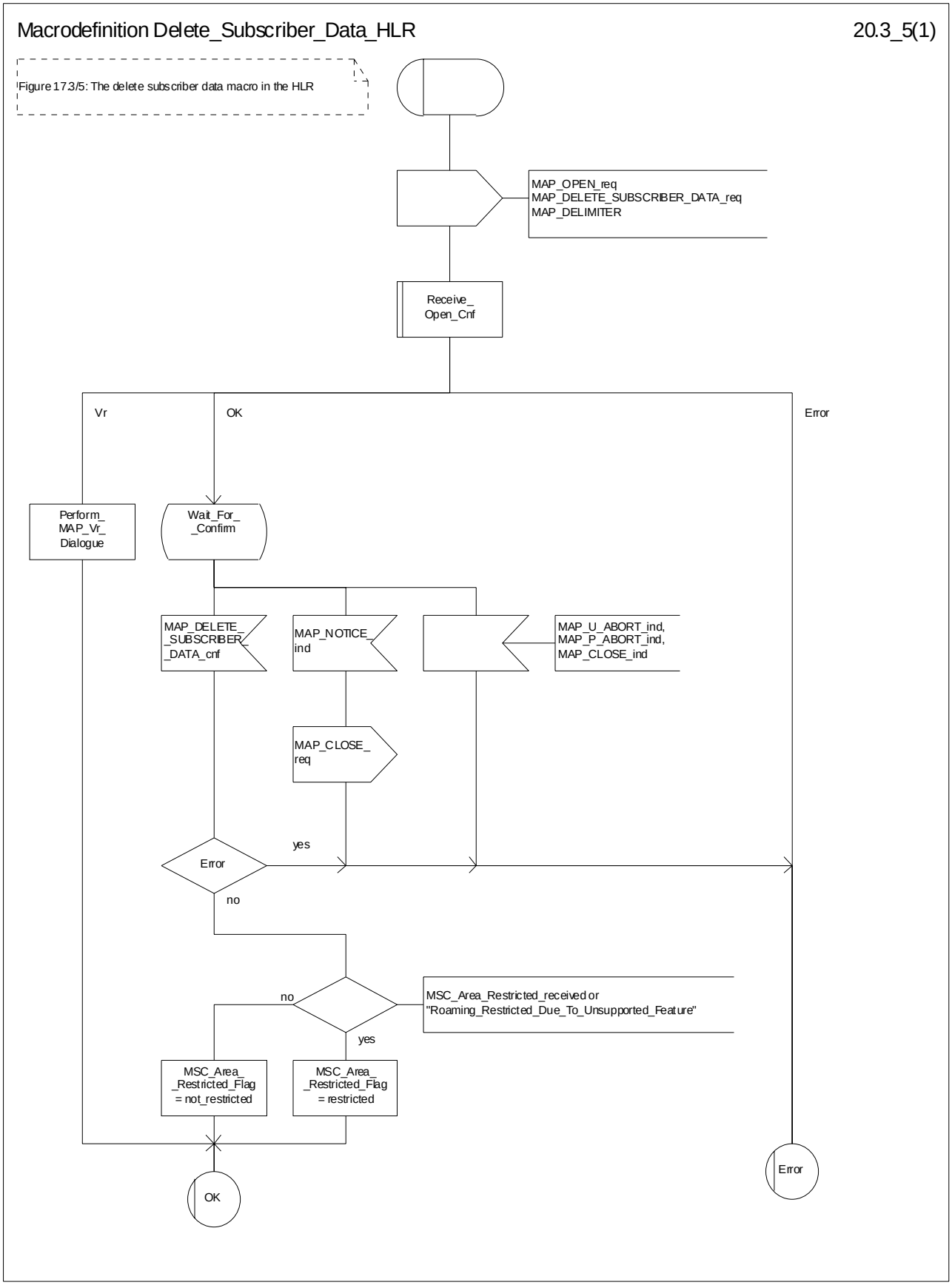


Figure 20.3/5: Macro Delete_Subscriber_Data_HLR

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 clause 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 checks 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.

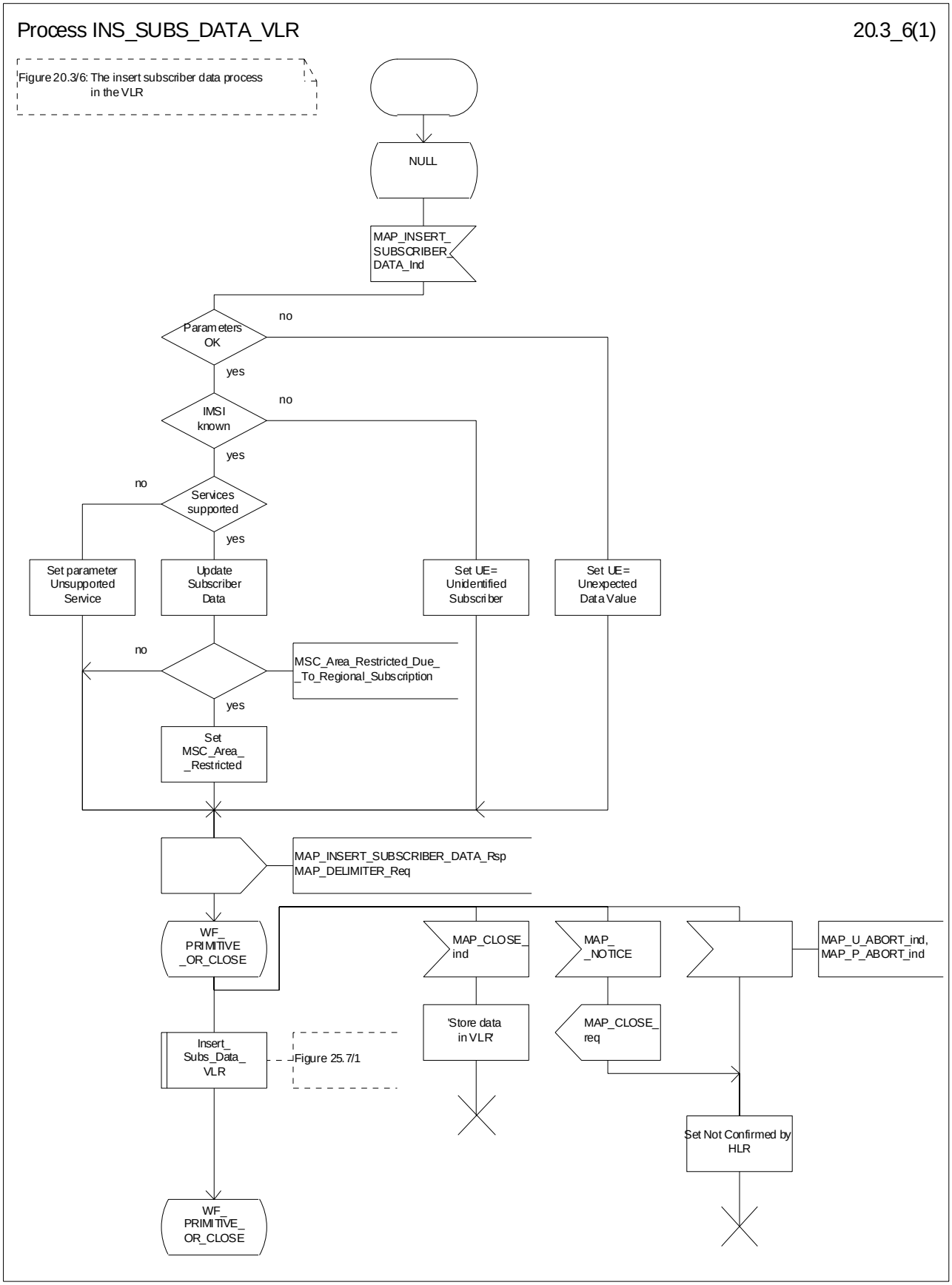


Figure 20.3/6: Process INS_SUBS_DATA_VLR

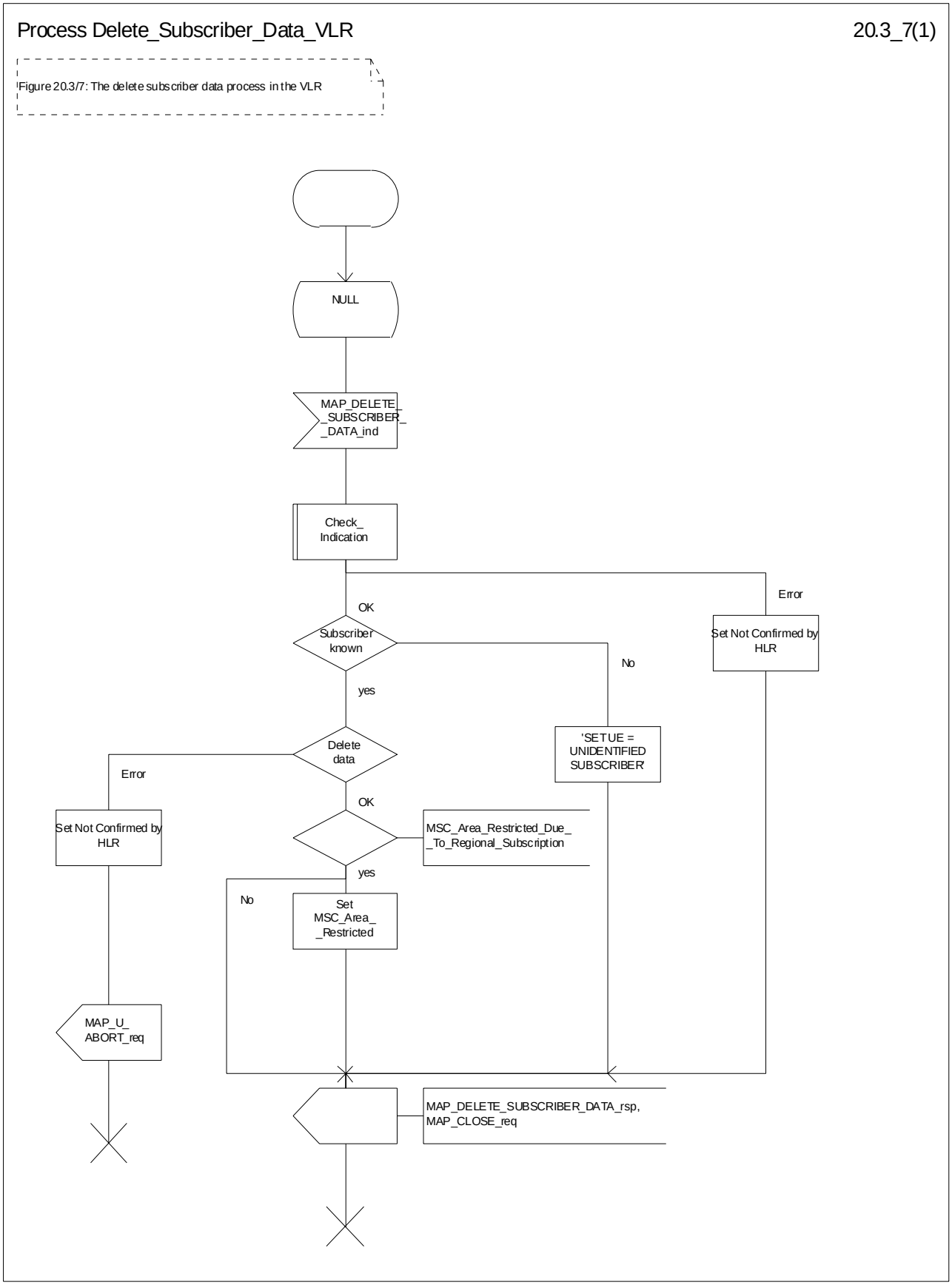


Figure 20.3/7: Process Delete_Subscriber_Data_VLR

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 clause 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 checks 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.

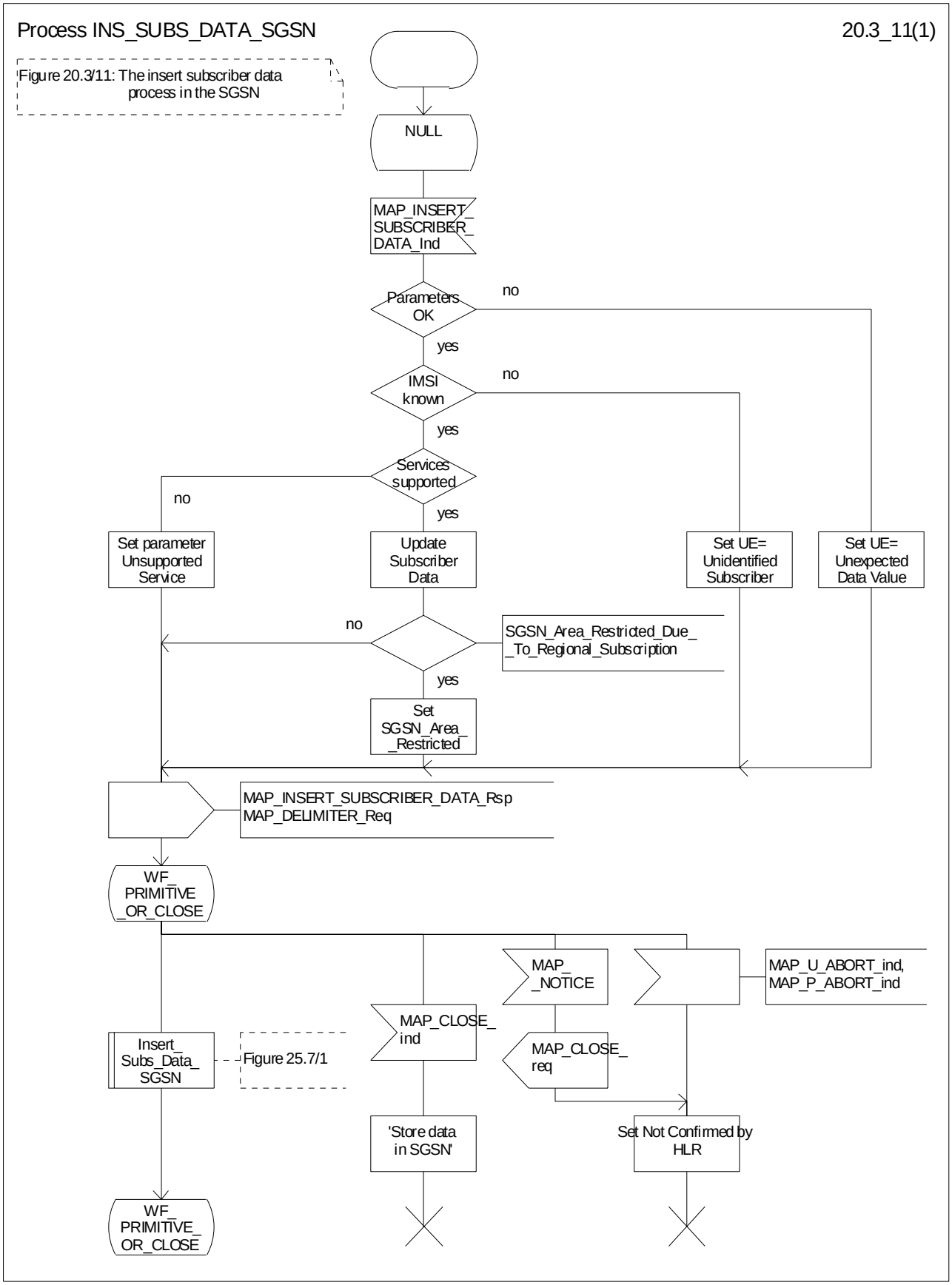


Figure 20.3/11: Process INS_SUBS_DATA_SGSN

Process Delete_Subscriber_Data_SGSN 20.3_12(1)

Figure 20.3/12: The delete subscriber data process in the SGSN

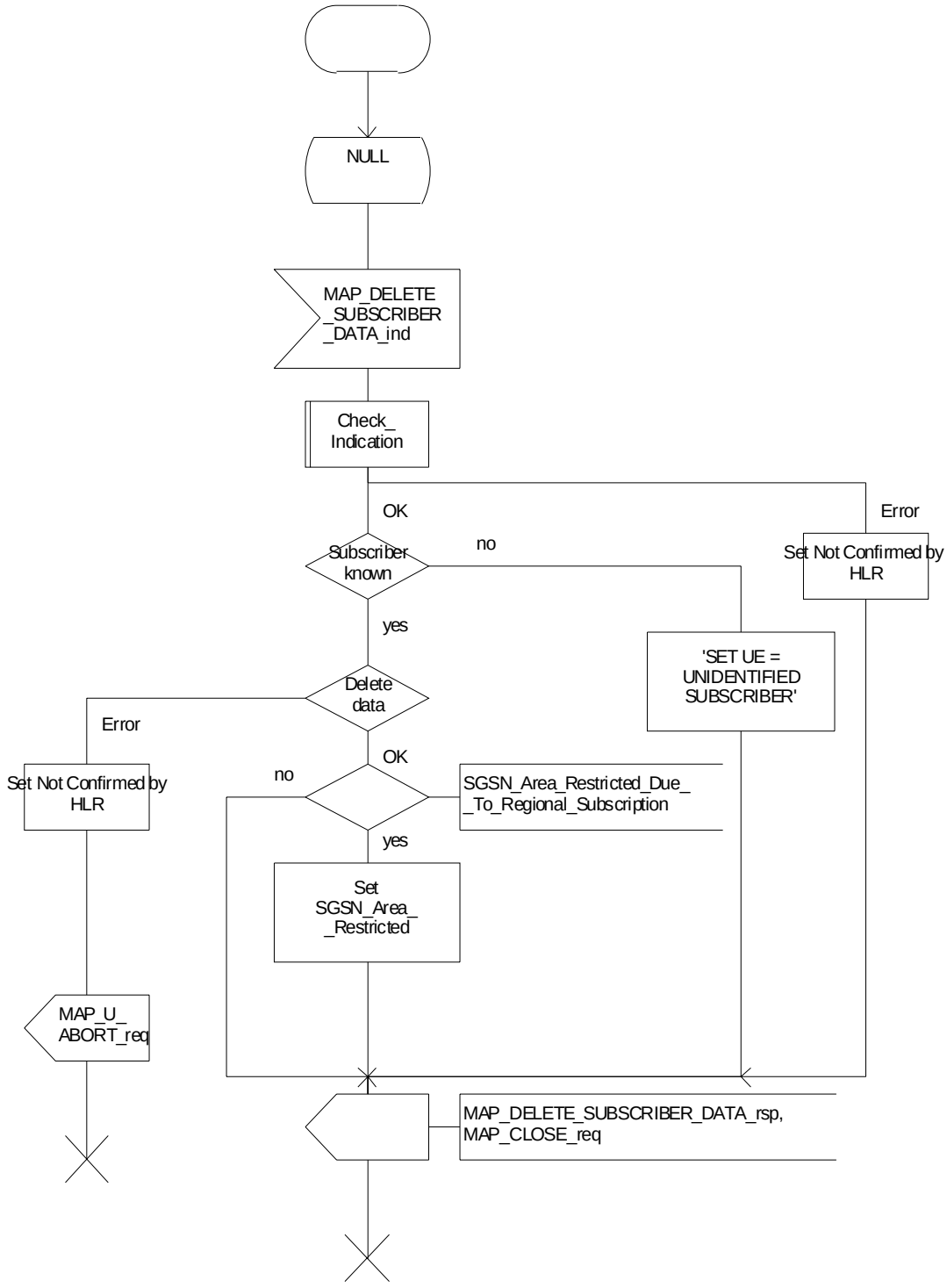


Figure 20.3/12: Process Delete_Subscriber_Data_SGSN

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.

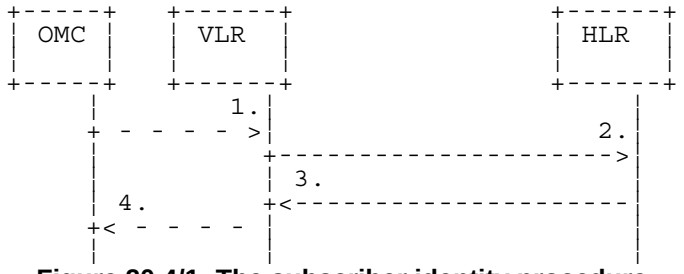


Figure 20.4/1: The subscriber identity procedure

1) Identity request.

2) MAP_SEND_IMSI.

3) MAP_SEND_IMSI_ACK.

4) Identity confirm.

20.4.1 Subscriber identity procedure in the HLR

Opening of the dialogue is described in the macro Receive_Open_Ind in clause 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.

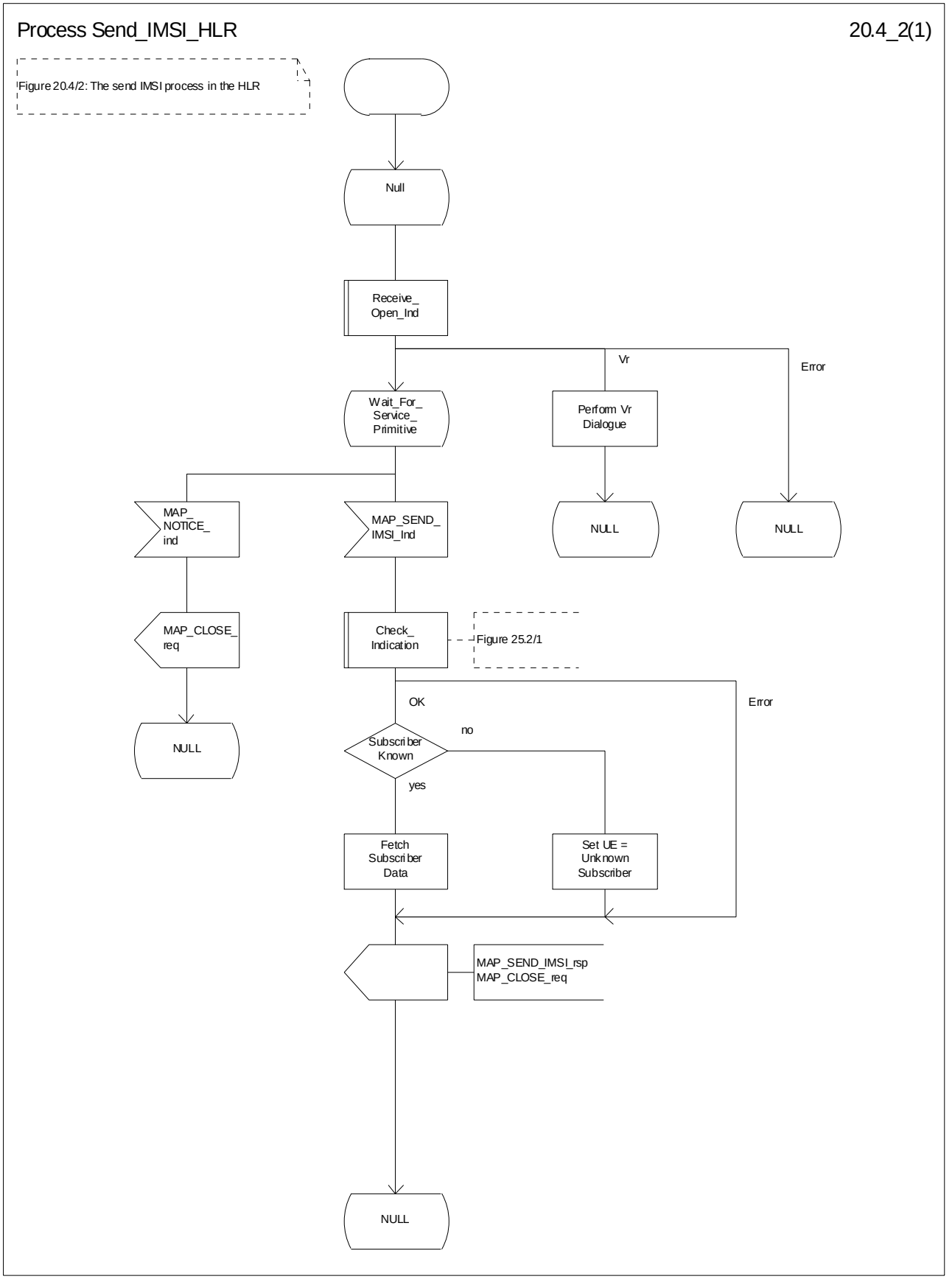


Figure 20.4/2: Process Send_IMSI_HLR

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 are sent to the OMC.

The subscriber identity procedure in the VLR is shown in figure 20.4/3.

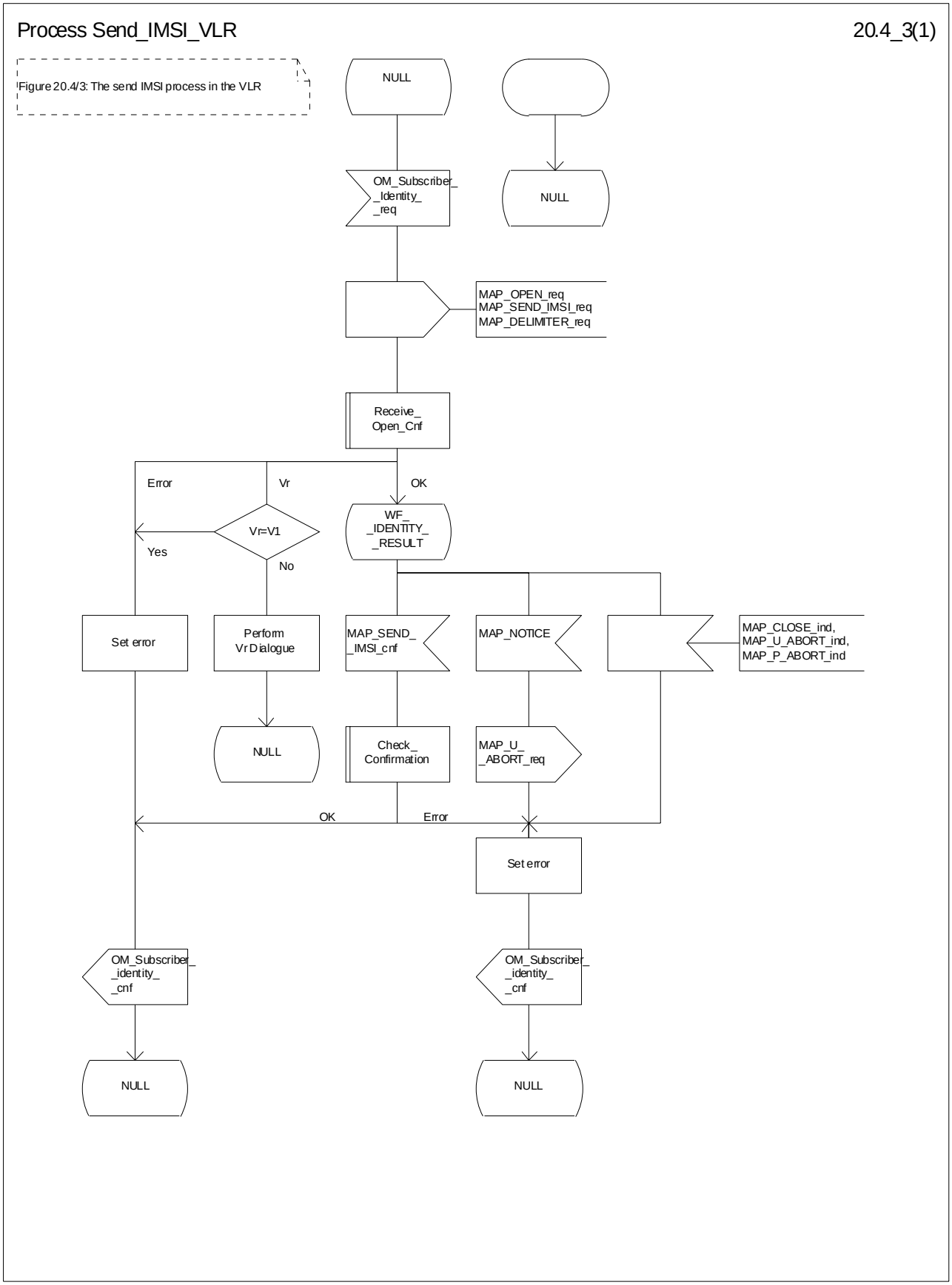


Figure 20.4/3: Process Send_IMSI_VLR

21 Call handling procedures

21.1 General

The MAP call handling procedures are used:

- to retrieve routeing 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 3GPP TS 23.018 [97].

The stage 2 specification for the retrieval of routeing information to handle a mobile terminating call is in 3GPP TS 23.018 [97]; modifications to this procedure for CAMEL are specified in 3GPP TS 23.078 [98], for optimal routeing of a basic mobile-to-mobile call in 3GPP TS 23.079 [99] and for CCBS in 3GPP TS 23.093 [107]. 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 3GPP TS 23.079 [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 3GPP TS 23.054 [103]. 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 are shown in 3GPP TS 23.018, 3GPP TS 23.078 and 3GPP TS 23.079 [99].

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 3GPP TS 23.093 [107].

21.2 Retrieval of routing information

21.2.1 General

The message flows for successful retrieval of routeing 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).

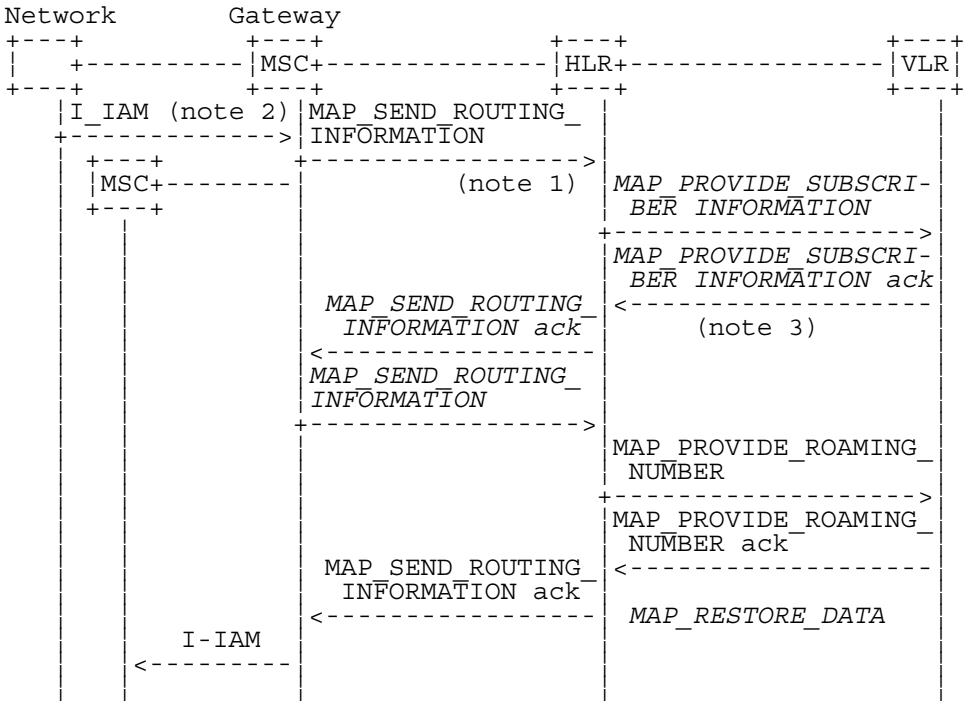


Figure 21.2/1: Message flow for retrieval of routeing information (non-optimally routed call)

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 3GPP TS 23.078 [98].

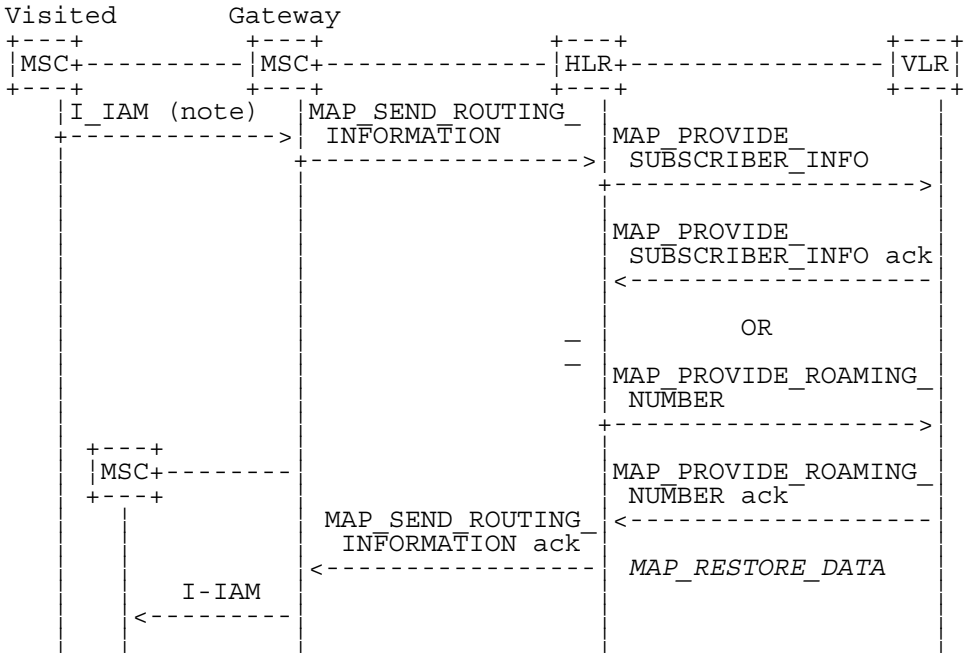


Figure 21.2/2: Message flow for retrieval of routeing information (optimally routed call)

xxx = Optional Procedure

NOTE 1: For Optimal Routeing phase 1, only one of the information flows for Provide Subscriber Info and Provide Roaming Number is used. For later phases of Optimal Routeing, 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.

NOTE 2: TUP or ISUP may be used in signalling between MSCs, depending on the network type between the MSCs. For further details on the TUP and ISUP procedures refer to the following CCITT Recommendations & 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 following MAP services are used to retrieve routing information:

MAP_SEND_ROUTING_INFORMATION	see clause 10.1;
MAP_PROVIDE_ROAMING_NUMBER	see clause 10.2;
MAP_PROVIDE_SUBSCRIBER_INFO	see clause 8.11.2;
MAP_RESTORE_DATA	see clause 8.10.3.

21.2.2 Process in the GMSC

The MAP process in the GMSC to retrieve routeing information for a mobile terminating call is shown in figure 21.2/3. The MAP process invokes macros not defined in this clause; the definitions of these macros can be found as follows:

Receive_Open_Cnf	see clause 25.1.2;
Check_Confirmation	see clause 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 routeing 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 routeing 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.

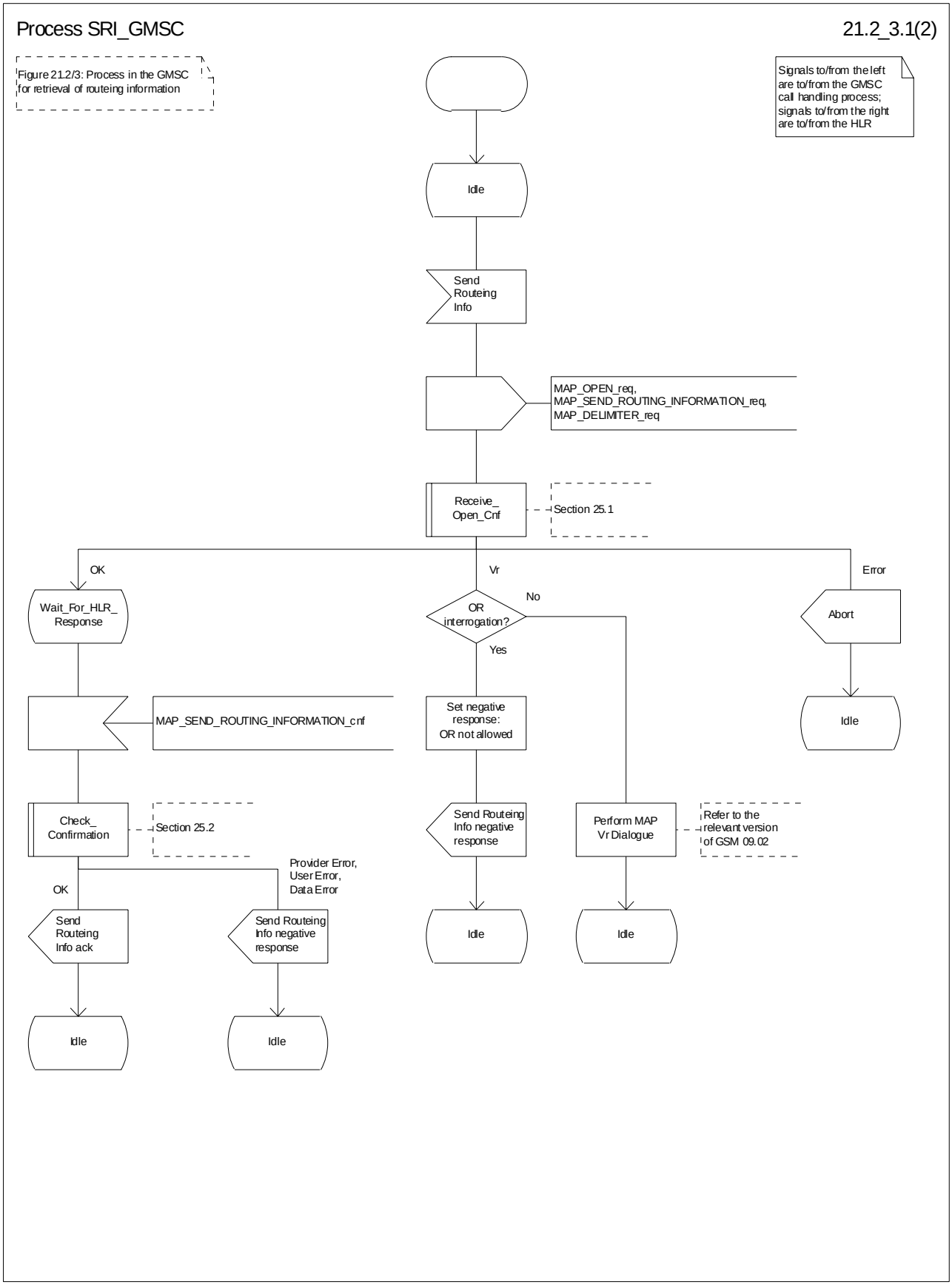


Figure 21.2/3 (sheet 1 of 2): Process SRI_GMSC

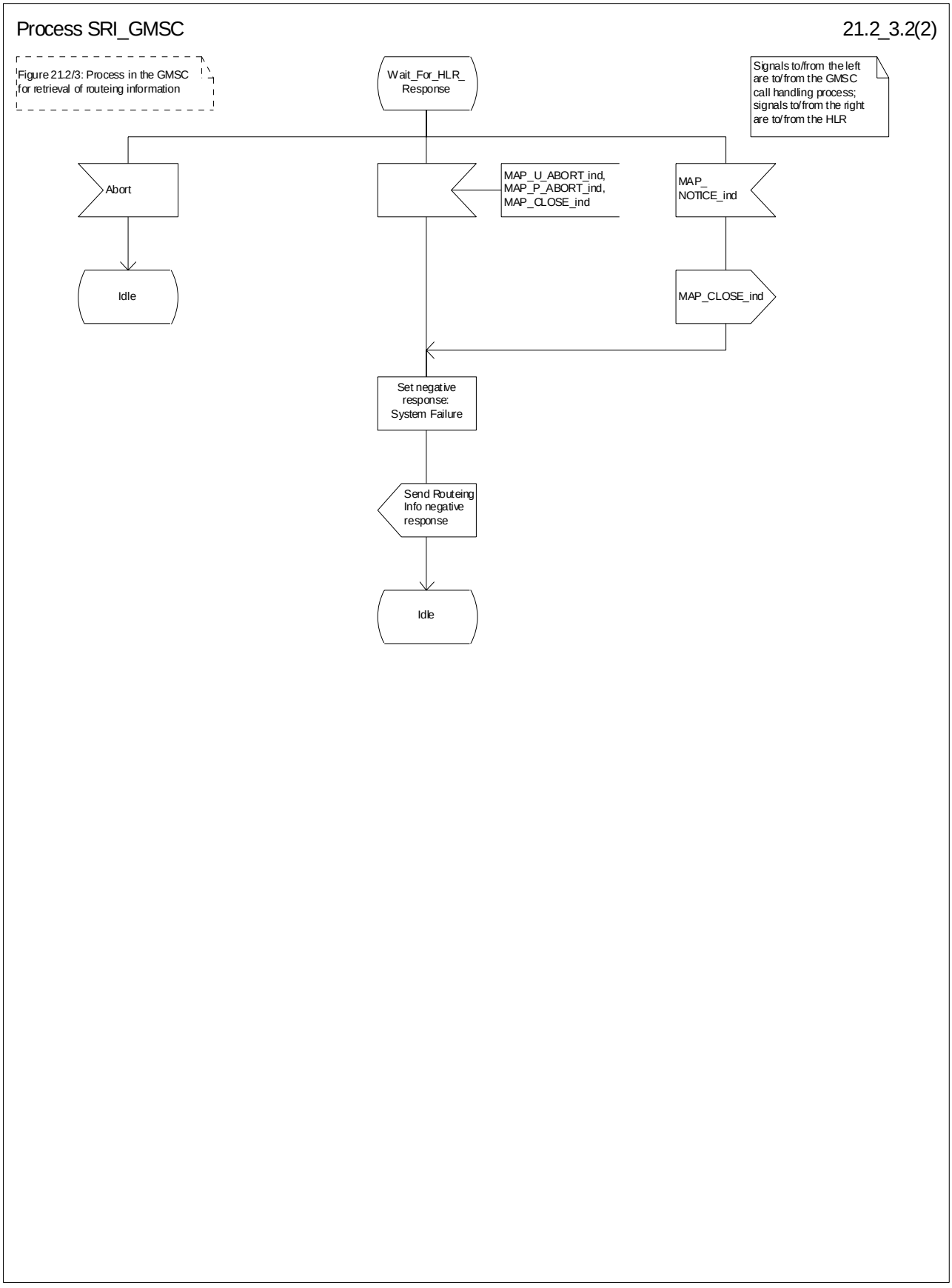


Figure 21.2/3 (sheet 2 of 2): Process SRI_GMSC

21.2.3 Procedures in the HLR

The MAP process in the HLR to retrieve routeing information for a mobile terminating call is shown in figure 21.2/4. The MAP process invokes macros not defined in this clause; the definitions of these macros can be found as follows:

Receive_Open_Ind	see clause 25.1.1;
Receive_Open_Cnf	see clause 25.1.2;
Check_Confirmation	see clause 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 Routeing Info request to the call handling process in the HLR, and waits for a response. The Send Routeing 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 Routeing Info ack, the MAP process constructs a MAP_SEND_ROUTING_INFORMATION service response containing the routeing information contained in the Send Routeing 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 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.

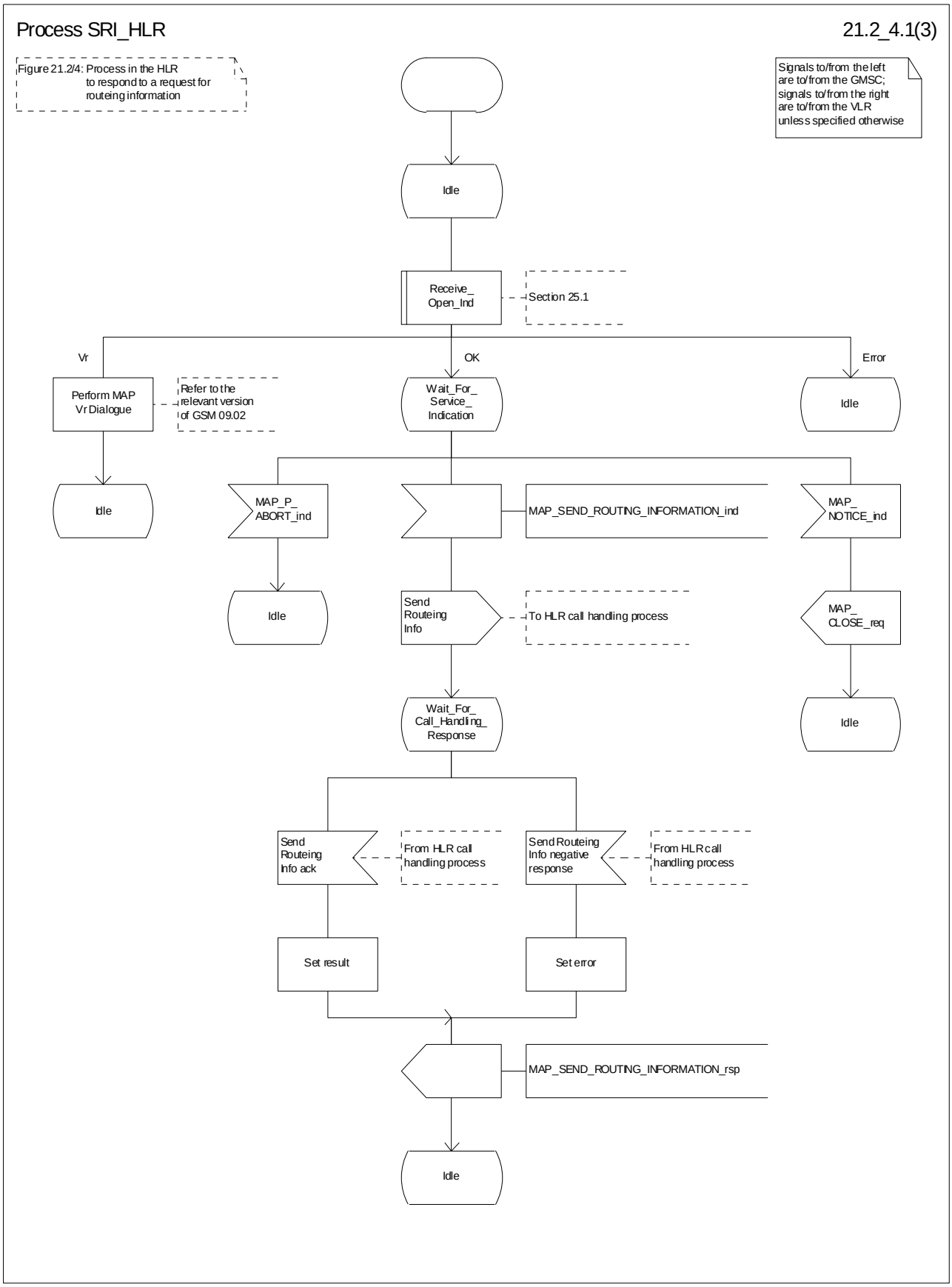


Figure 21.2/4 (sheet 1 of 3): Process SRI_HLR

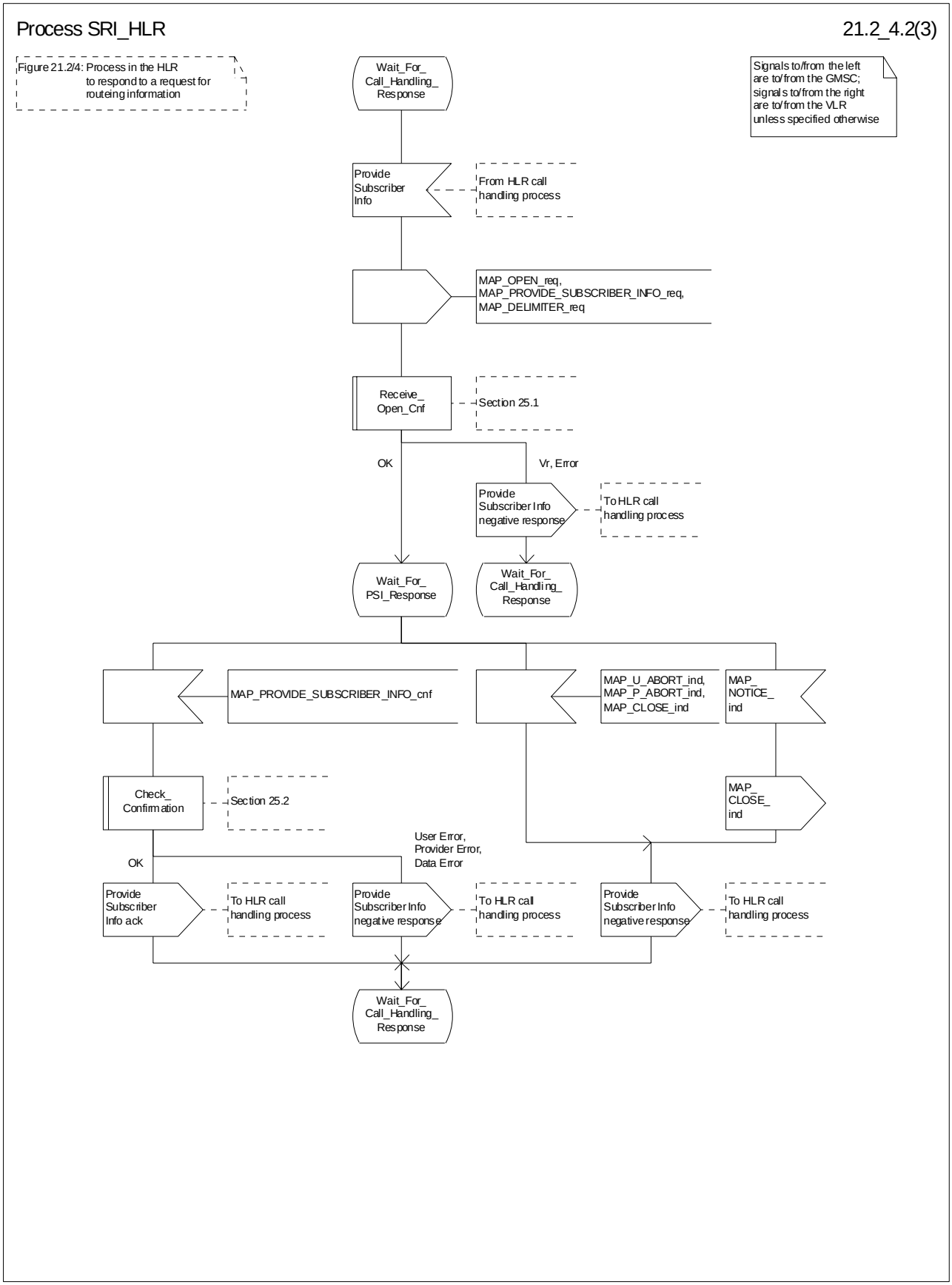


Figure 21.2/4 (sheet 2 of 3): Process SRI_HLR

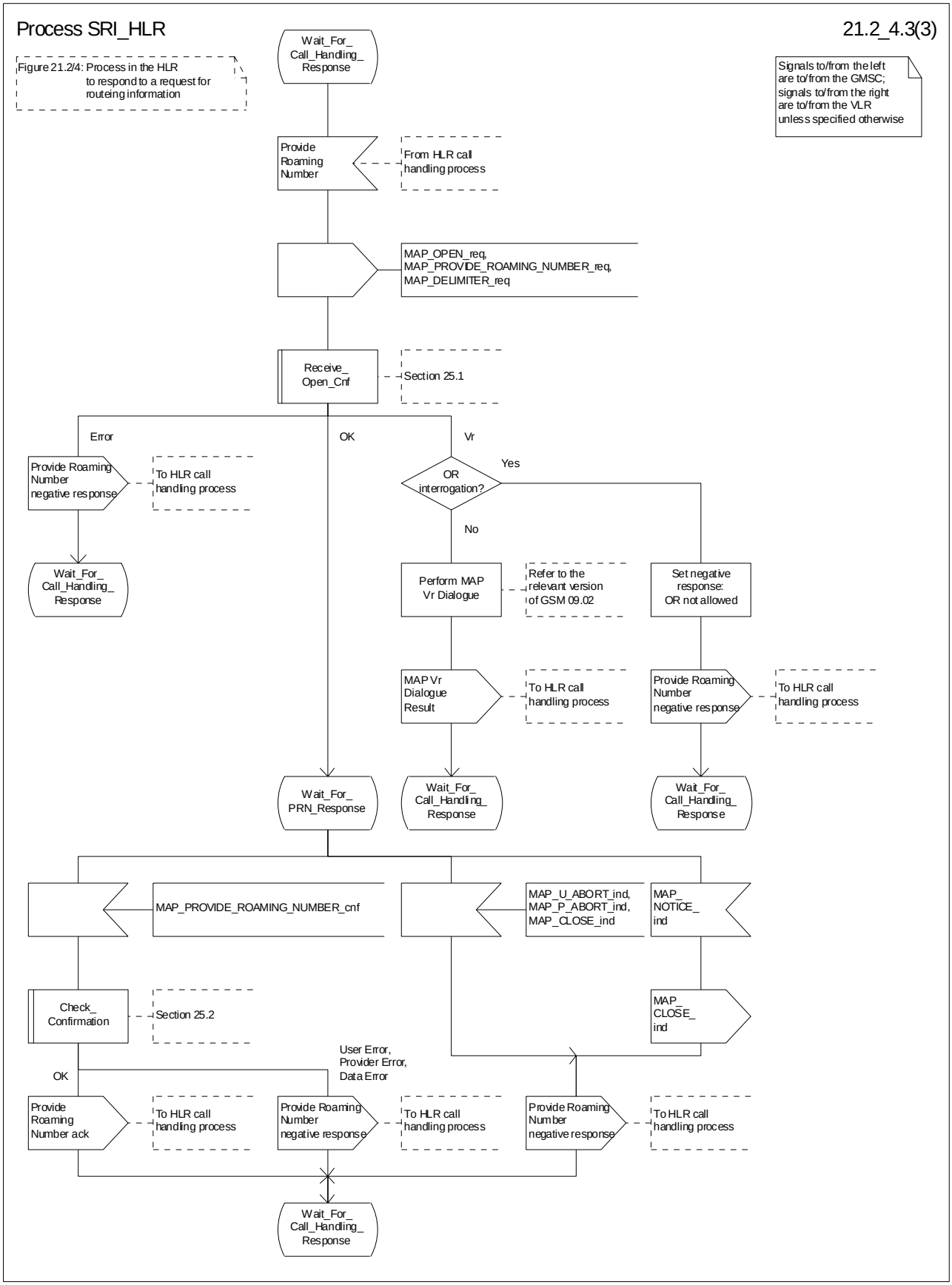


Figure 21.2/4 (sheet 3 of 3): Process SRI_HLR

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 clause; the definition of this macro can be found as follows:

Receive_Open_Ind see clause 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 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.

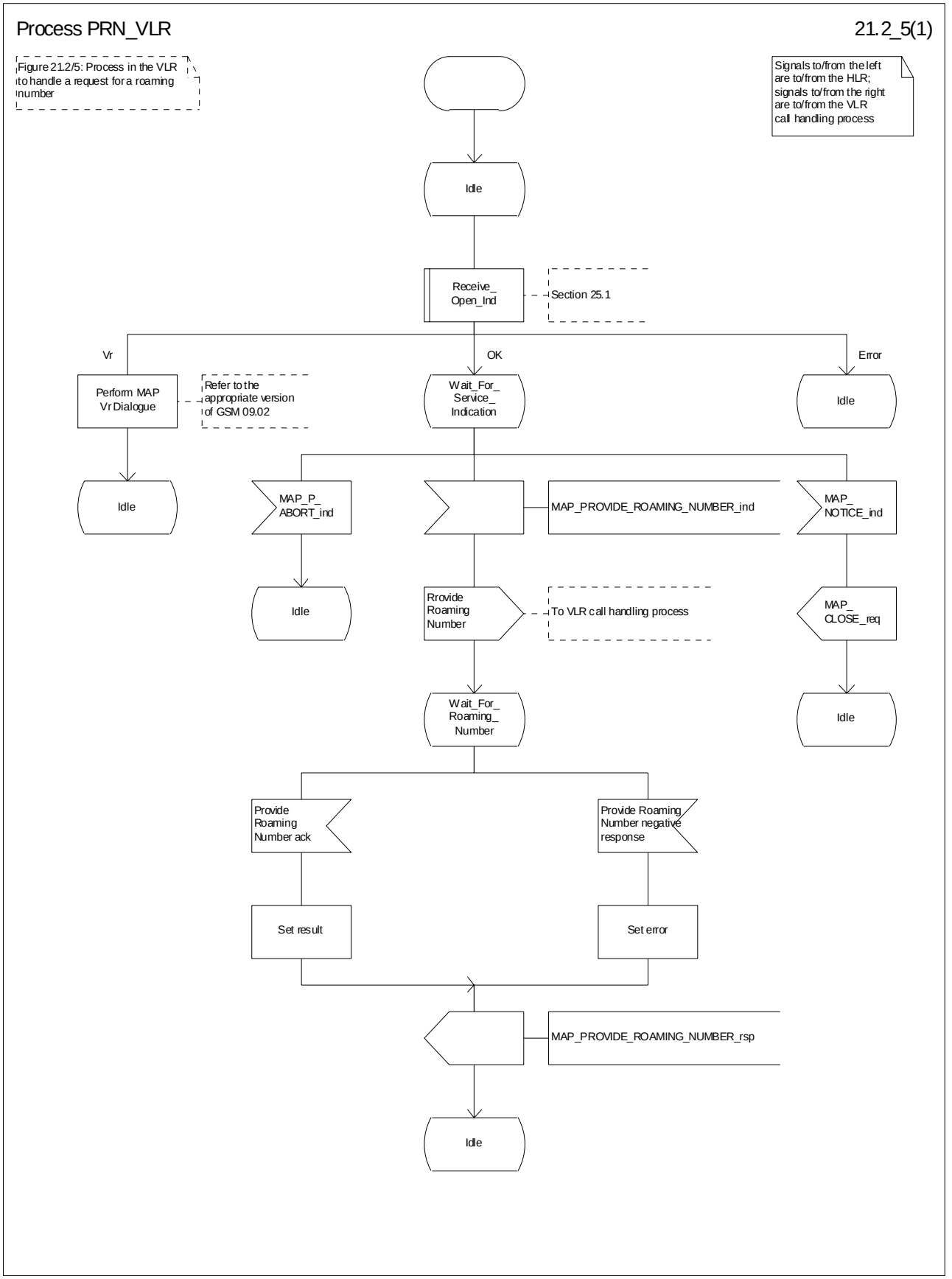


Figure 21.2/5: Process PRN_VLR

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 clause; the definitions of these macros can be found as follows:

Receive_Open_Cnf	see clause 25.1.2;
Check_Confirmation	see clause 25.2.2;
Insert_Subs_Data_VLR	see clause 25.7.1;
Activate_Tracing_VLR	see clause 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_Subs_Data_VLR as described in clause 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 clause 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.

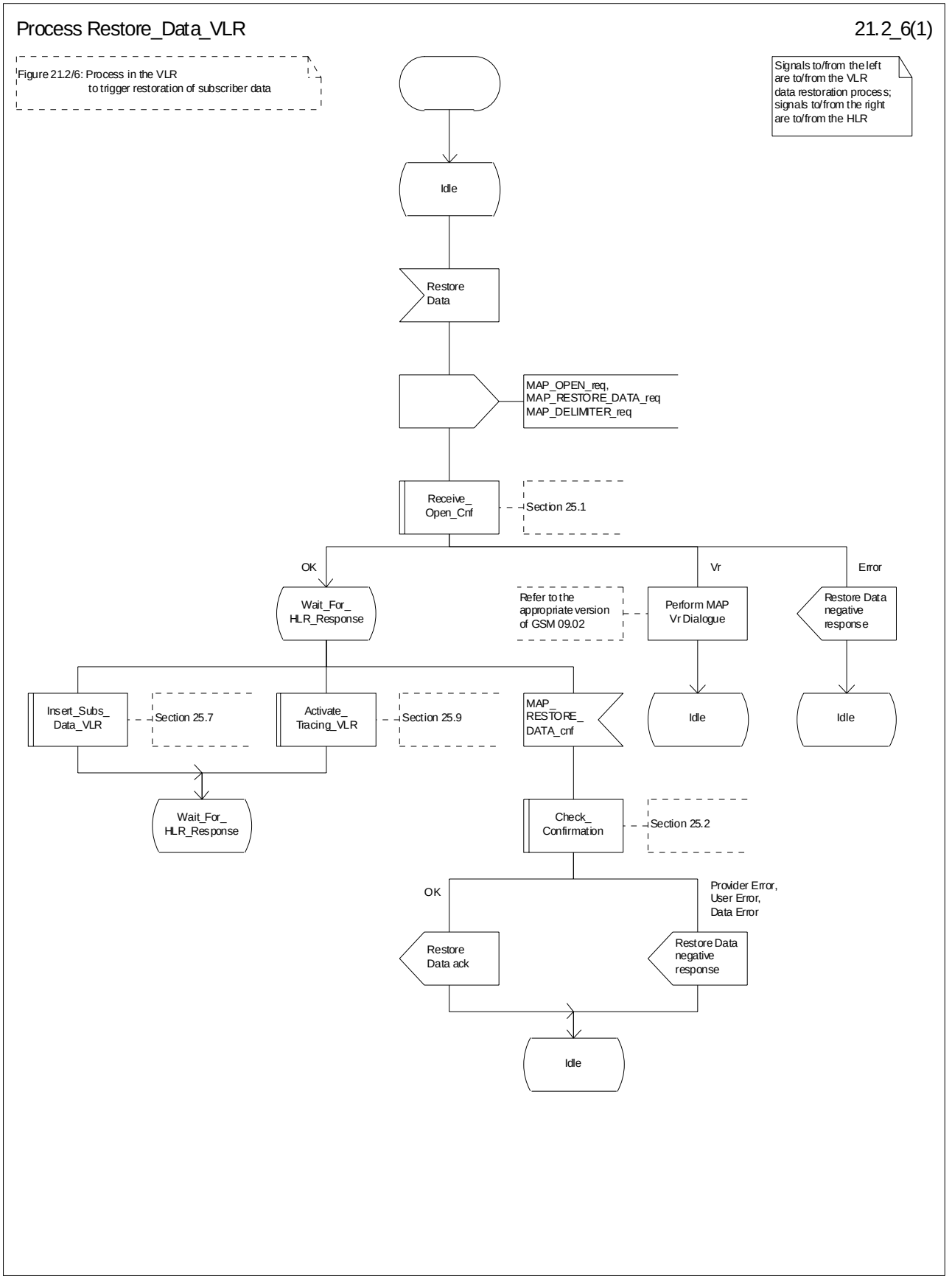


Figure 21.2/6: Process Restore_Data_VLR

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 clause; the definition of this macro can be found as follows:

Receive_Open_Ind see clause 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.

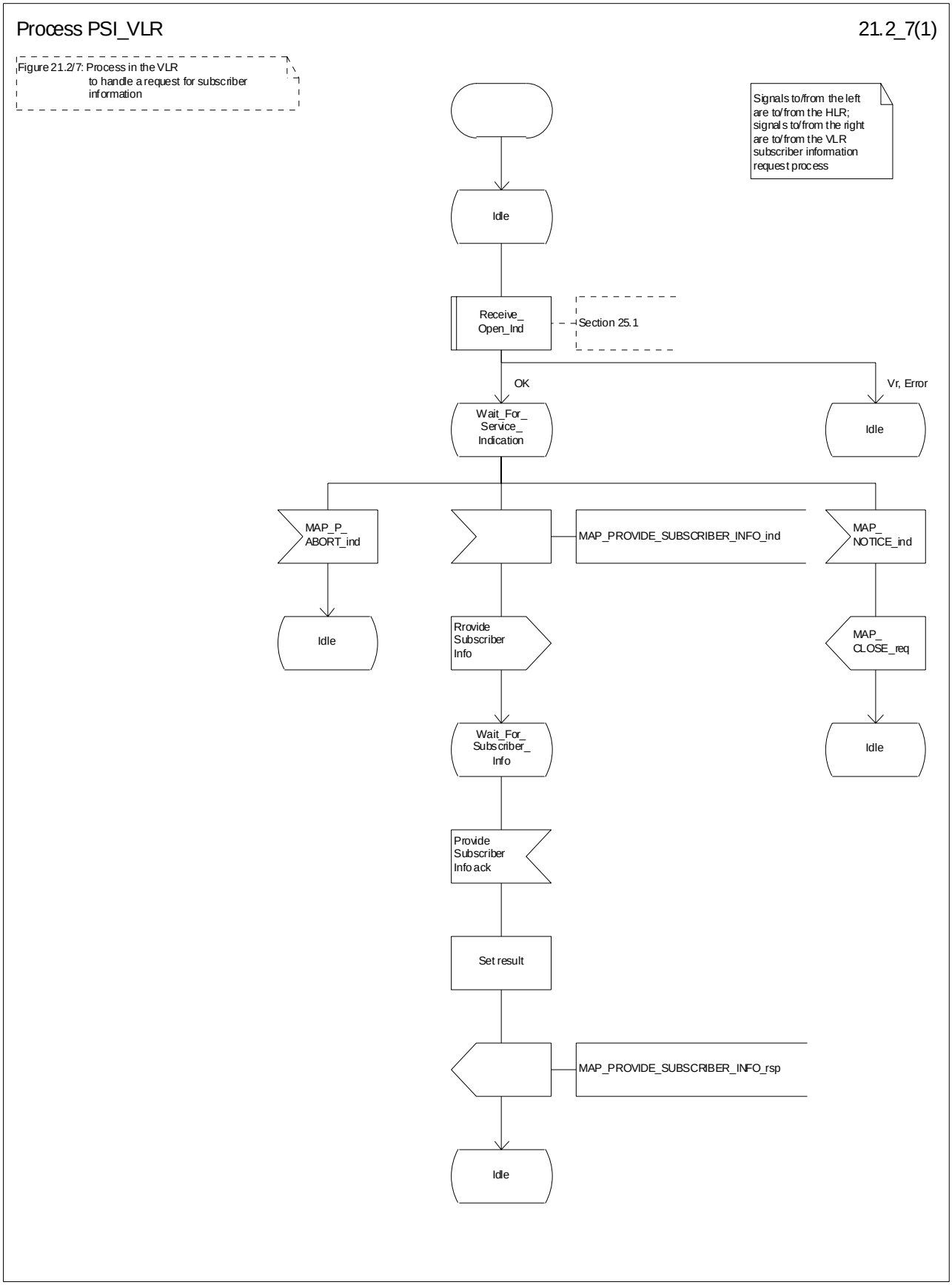


Figure 21.2/7: Process PSI_VLR

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.

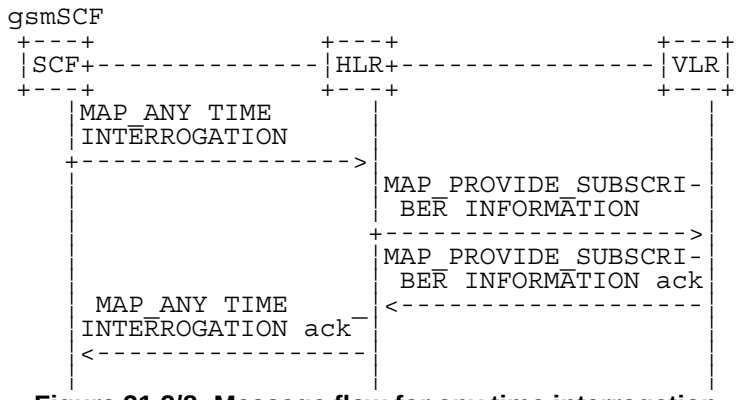


Figure 21.2/8: Message flow for any time interrogation

The following MAP services are used to retrieve routing information:

MAP_ANY_TIME_INTERROGATION see clause 8.11.1;

MAP_PROVIDE_SUBSCRIBER_INFO see clause 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 clause; the definitions of these macros can be found as follows:

Receive_Open_Ind see clause 25.1.1;

Receive_Open_Cnf see clause 25.1.2;

Check_Confirmation see clause 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 3GPP 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.

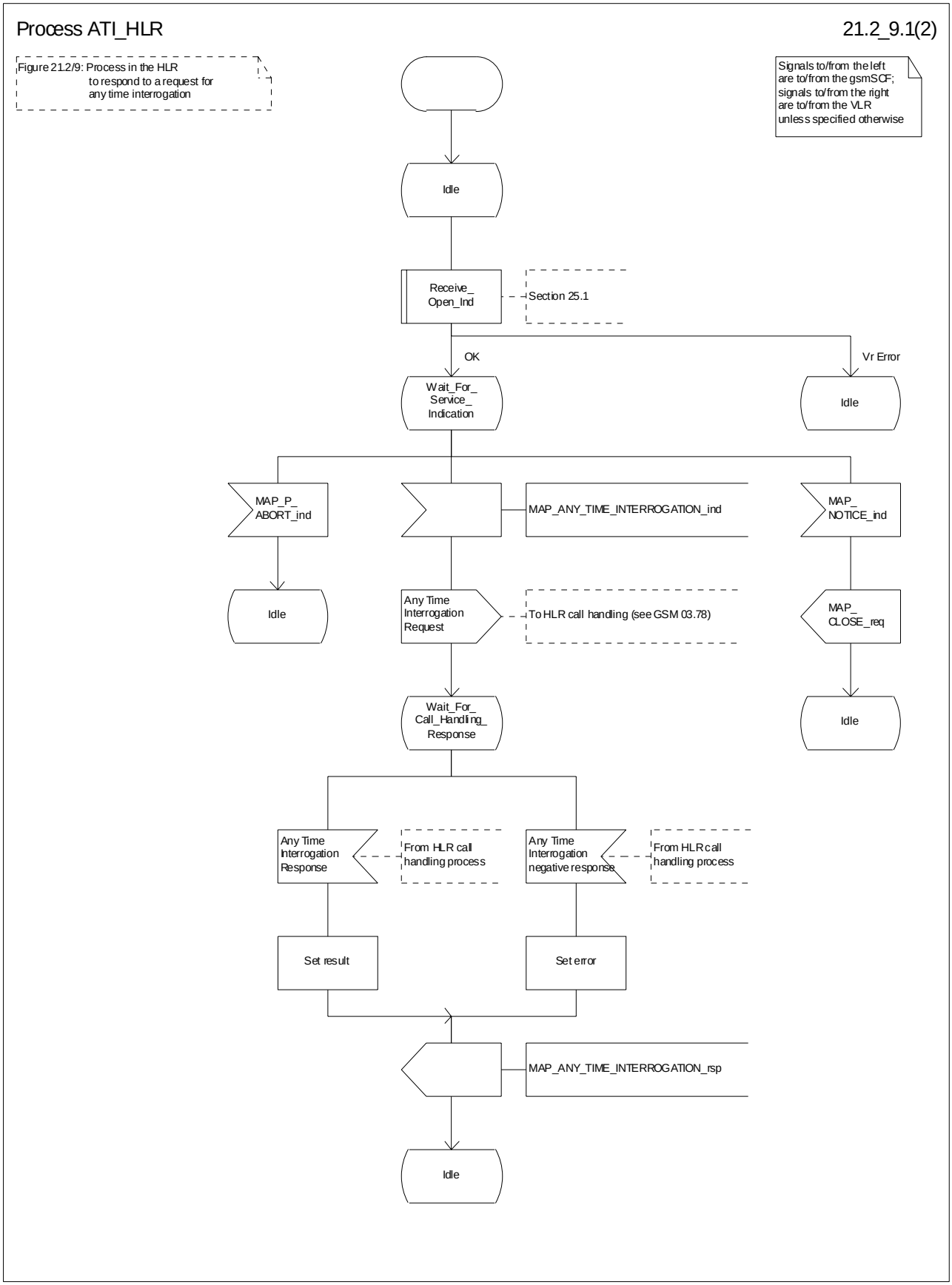


Figure 21.2/9 (sheet 1 of 2): Process ATI_HLR (New)

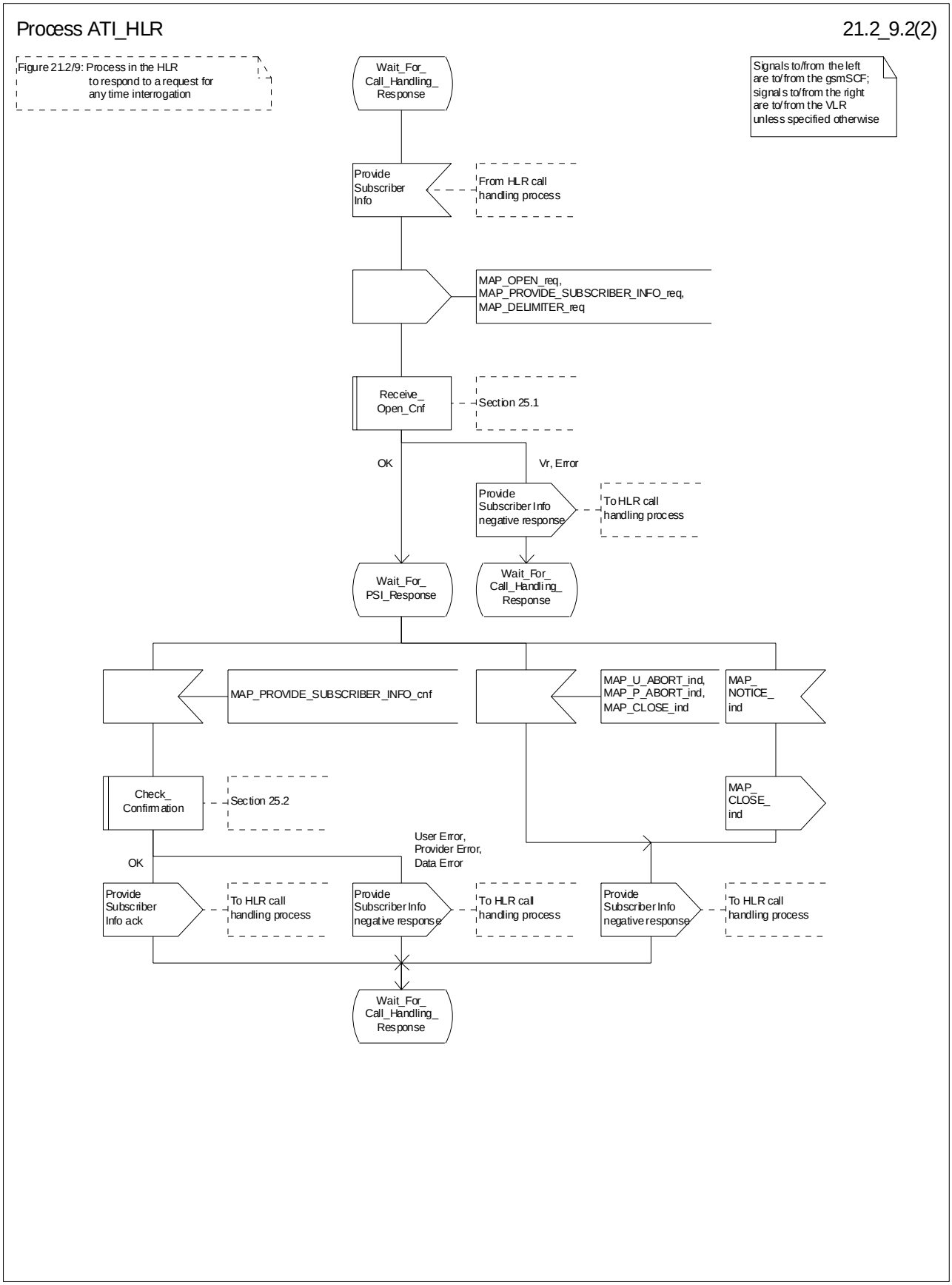


Figure 21.2/9 (sheet 2 of 2): Process ATI_HLR (New)

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.

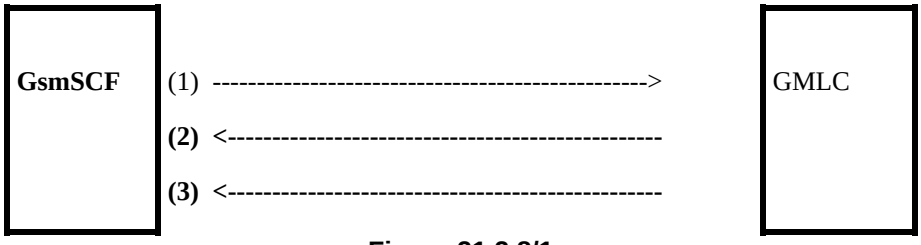


Figure 21.2.8/1

(1) Any Time Interrogation (gsmSCF to GMLC).

(2) Any Time Interrogation Result (GMLC to gsmSCF).

(3) Any Time Interrogation Error (GMLC to gsmSCF)The following MAP services are used to retrieve routing information:

- MAP_ANY_TIME_INTERROGATION see clause 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.

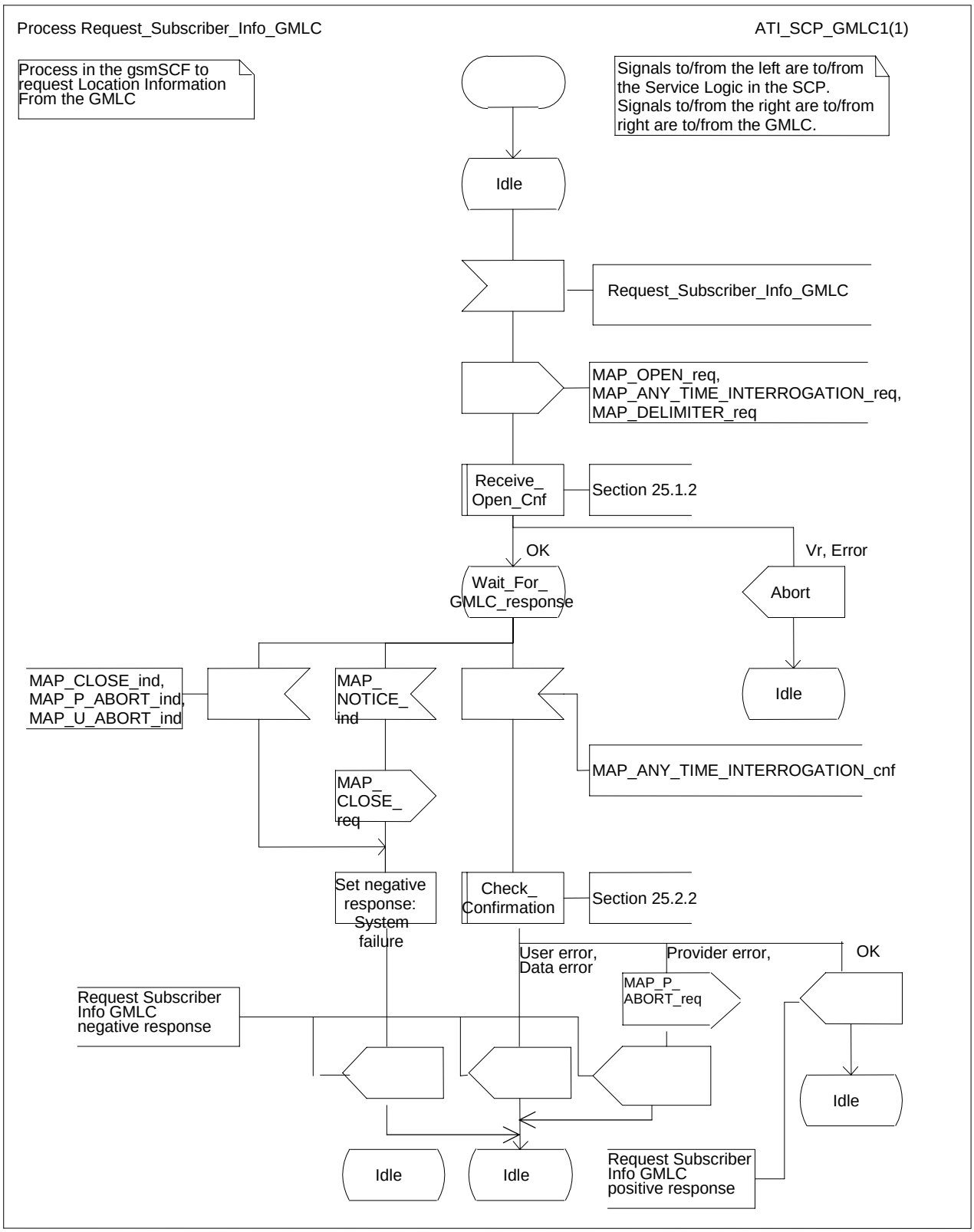


Figure 21.2.8/2: Process Request_Subscriber_Info_GMLC

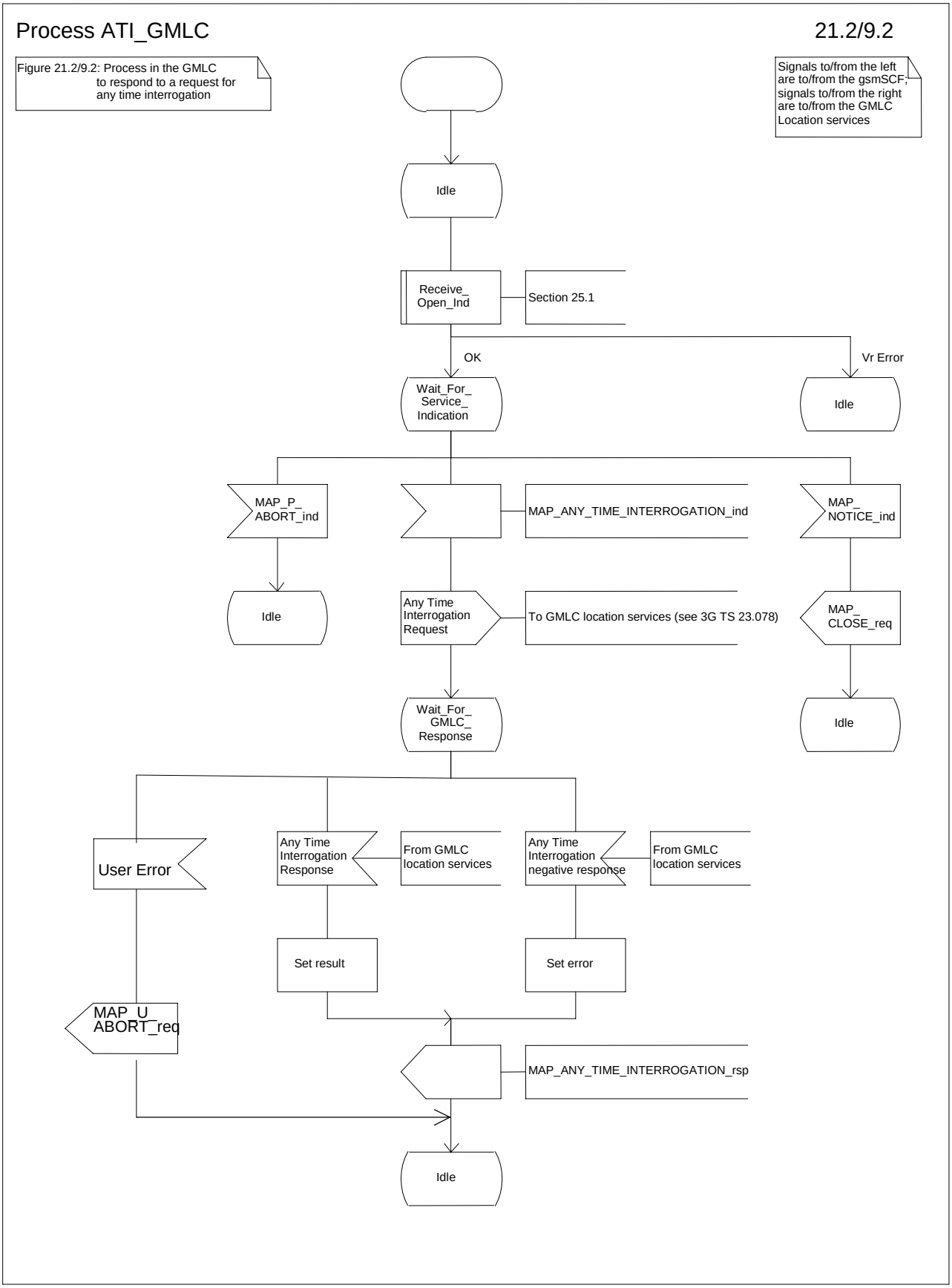


Figure 21.2.8/3: Process ATI_GMLC

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.

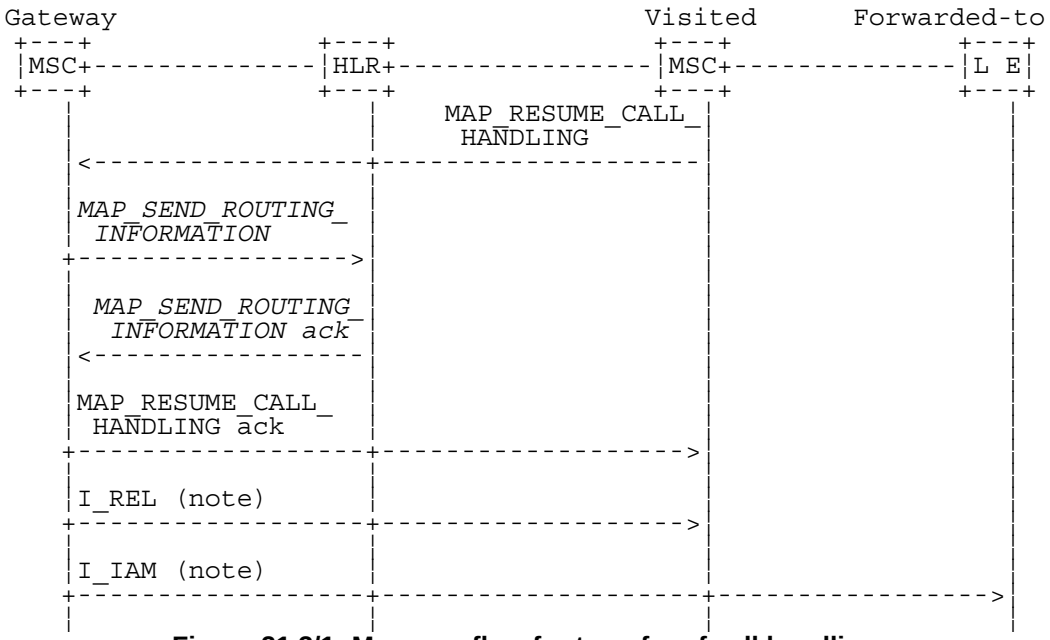


Figure 21.3/1: Message flow for transfer of call handling

xxx = Optional Procedure

NOTE: 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.

If the HLR indicated in the response to the original request for routeing information that forwarding interrogation is required, the GMSC executes the Send Routeing 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 routeing information for a mobile terminating call is shown in figure 21.3/2. The MAP process invokes macros not defined in this clause; the definitions of these macros can be found as follows:

Receive_Open_Cnf see clause 25.1.2;

Check_Confirmation see clause 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 routeing 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 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.

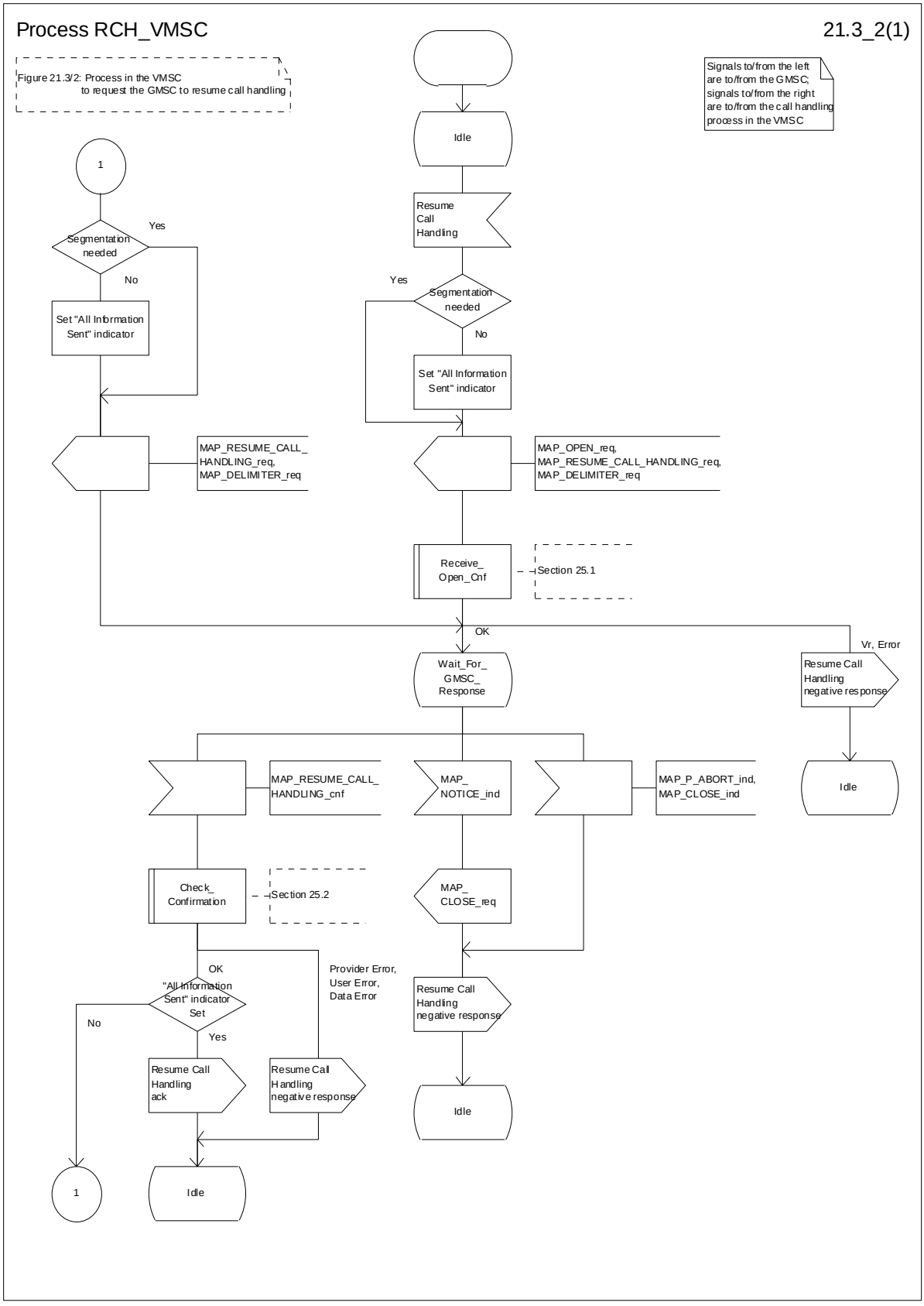


Figure 21.3/2: Process RCH_VMSC

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 clause; the definition of this macro can be found as follows:

Receive_Open_Ind see clause 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.

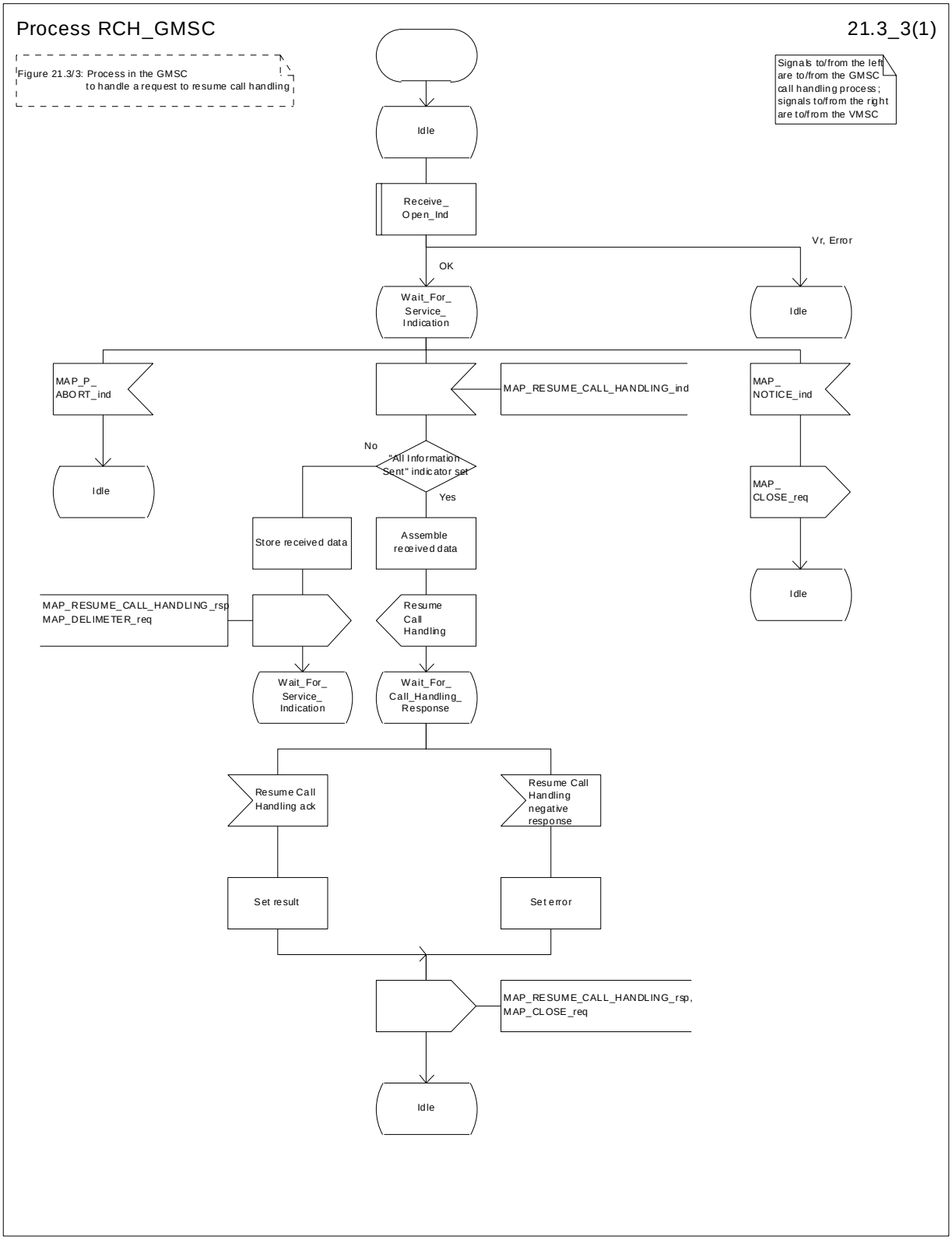


Figure 21.3/3: Process RCH_GMSC

21.4 Inter MSC Group Call Procedures

21.4.1 General

The message flows for successful inter MSC group call / broadcast call set-up is shown in figure 21.4/1.

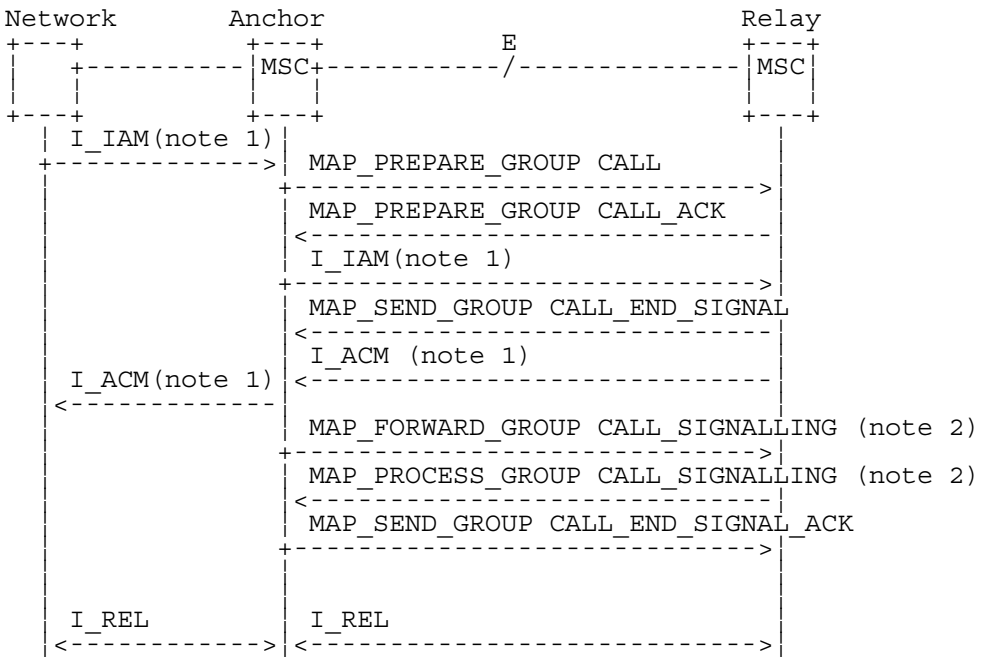


Figure 21.4/1: Message flow for inter MSC group call / broadcast call

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.

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 clause; the definitions of these macros can be found as follows:

Receive_Open_Cnf	see clause 25.1.2;
Check_Indication	see clause 25.2.1;
Check_Confirmation	see clause 25.2.2.

Successful Outcome

When the MAP process receives a Prepare Group Call request from the ASCI 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 set-up 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.

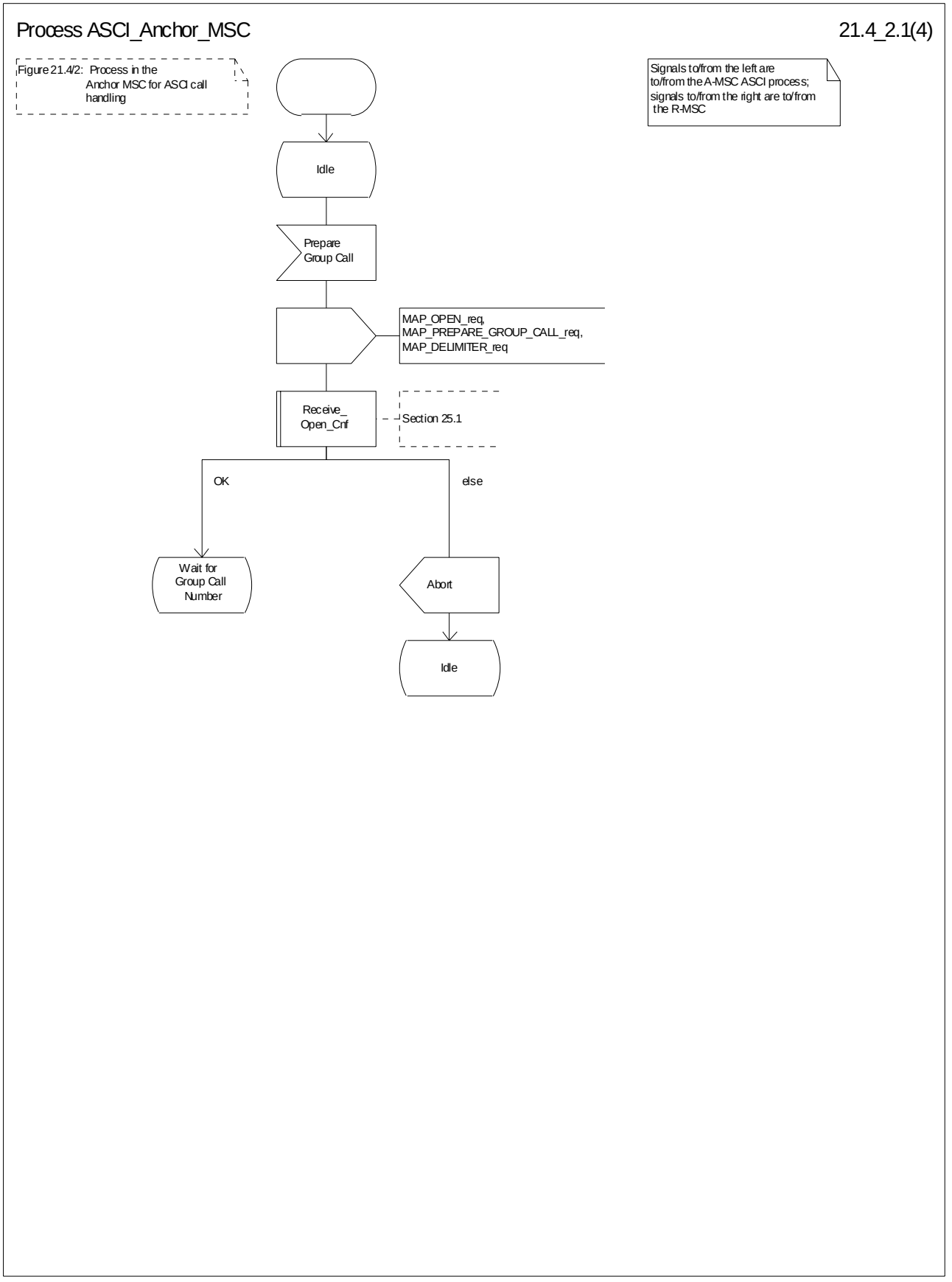


Figure 21.4/2 (sheet 1 of 4): Process ASCI_Anchor_MSC

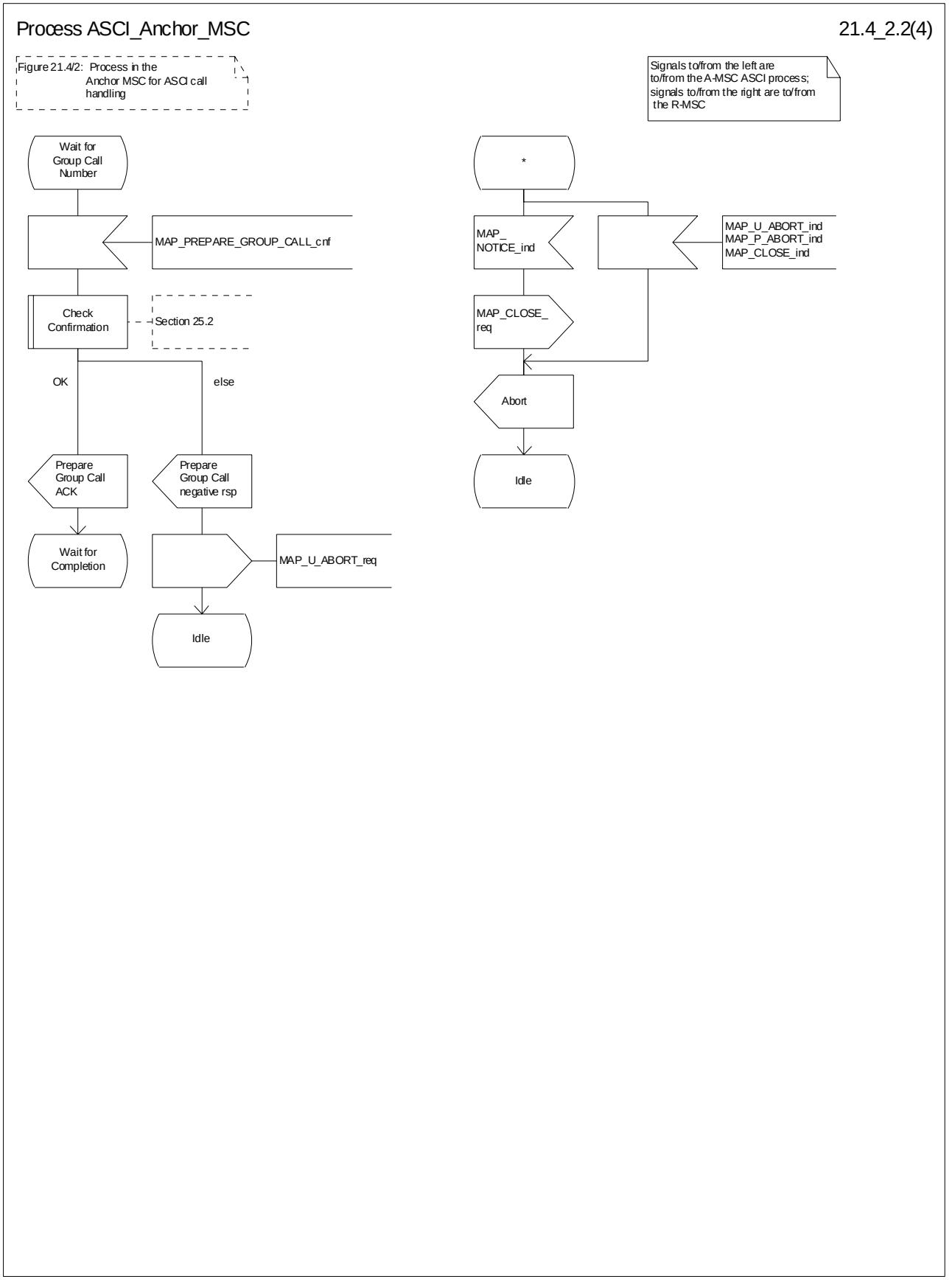


Figure 21.4/2 (sheet 2 of 4): Process ASCI_Anchor_MSC

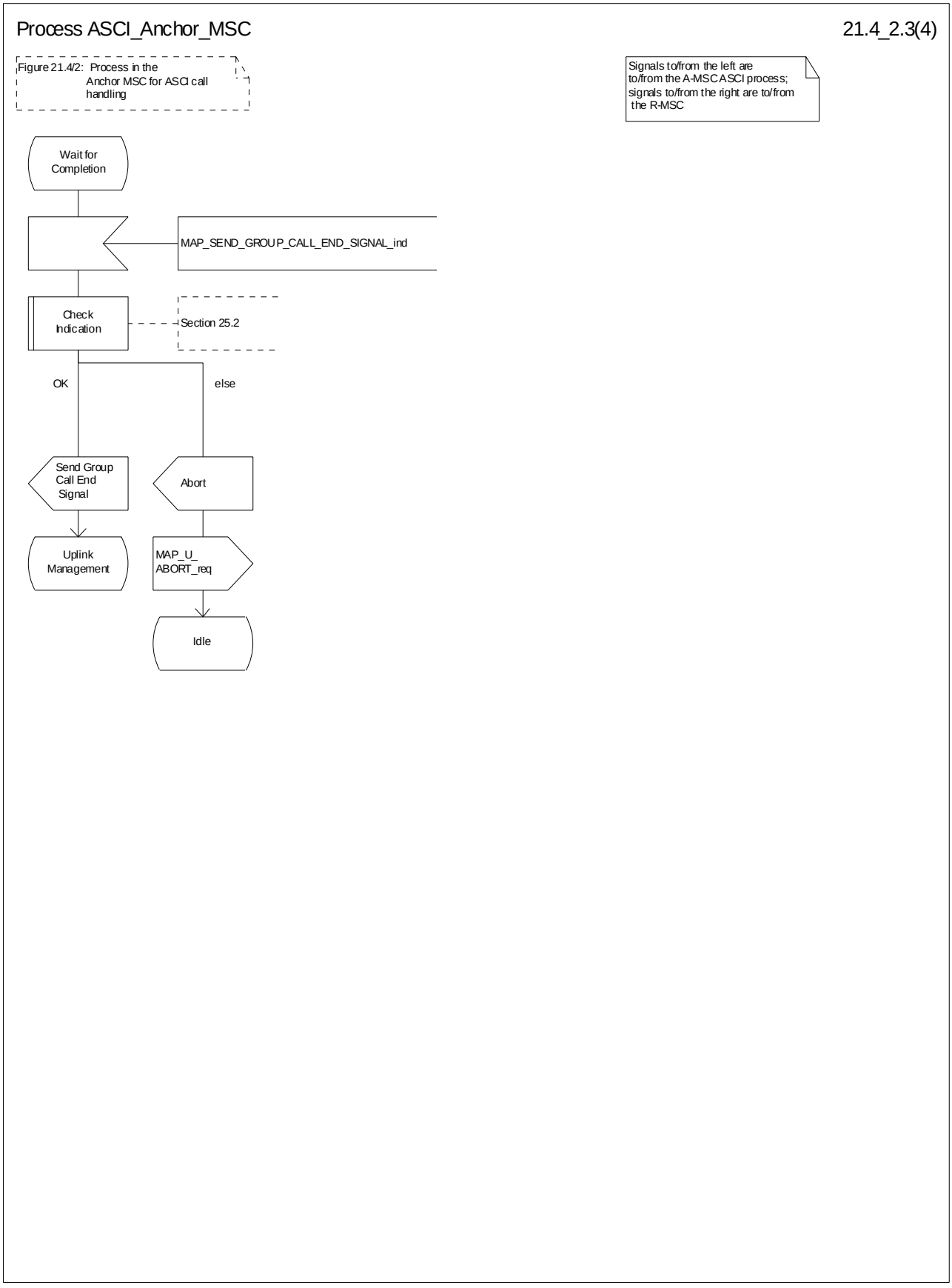


Figure 21.4/2 (sheet 3 of 4): Process ASCI_Anchor_MSC

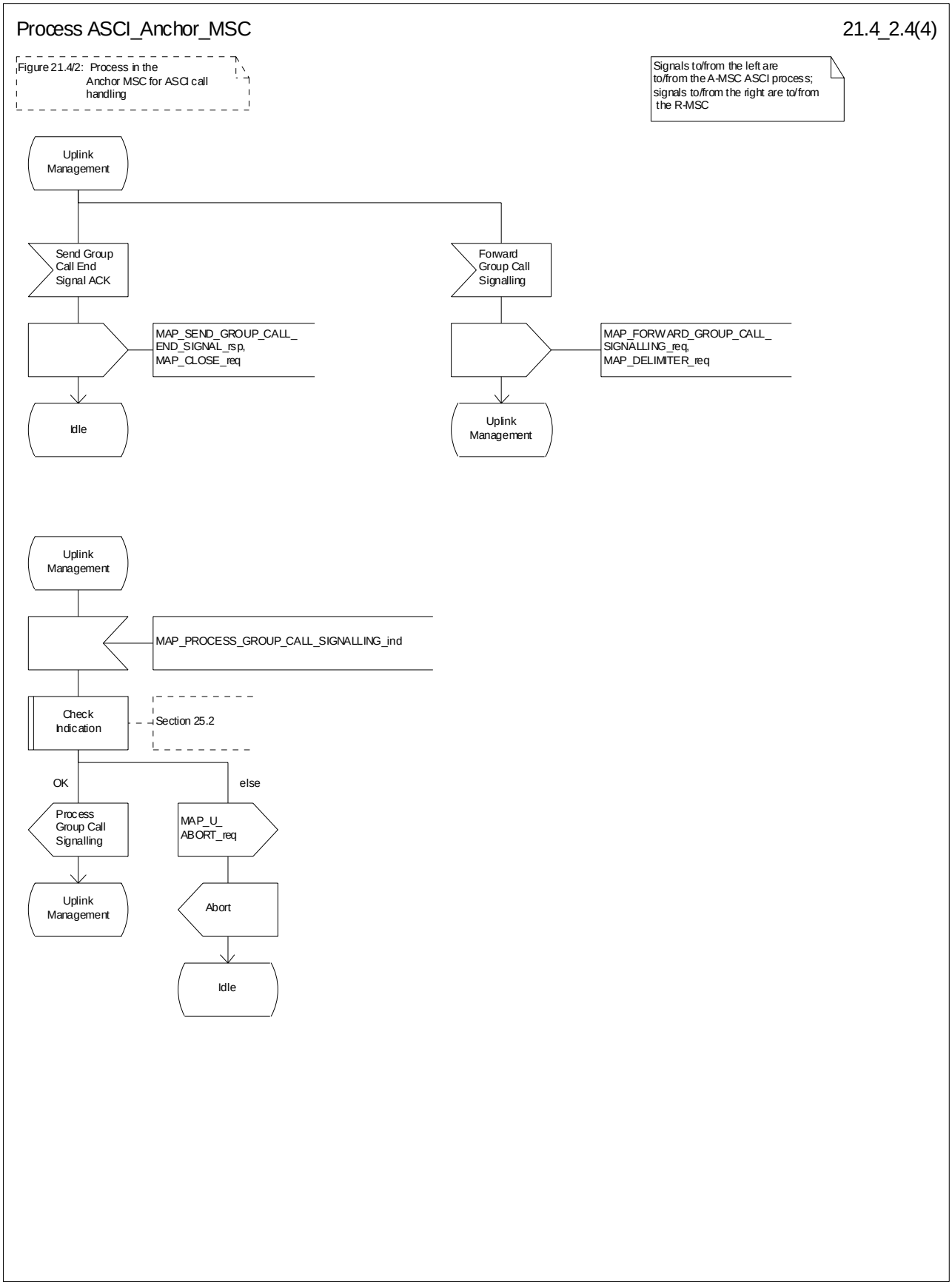


Figure 21.4/2 (sheet 4 of 4): Process ASCI_Anchor_MSC

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 clause; the definitions of these macros can be found as follows:

Receive_Open_Ind see clause 25.1.2;

Check_Indication see clause 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 ind 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.

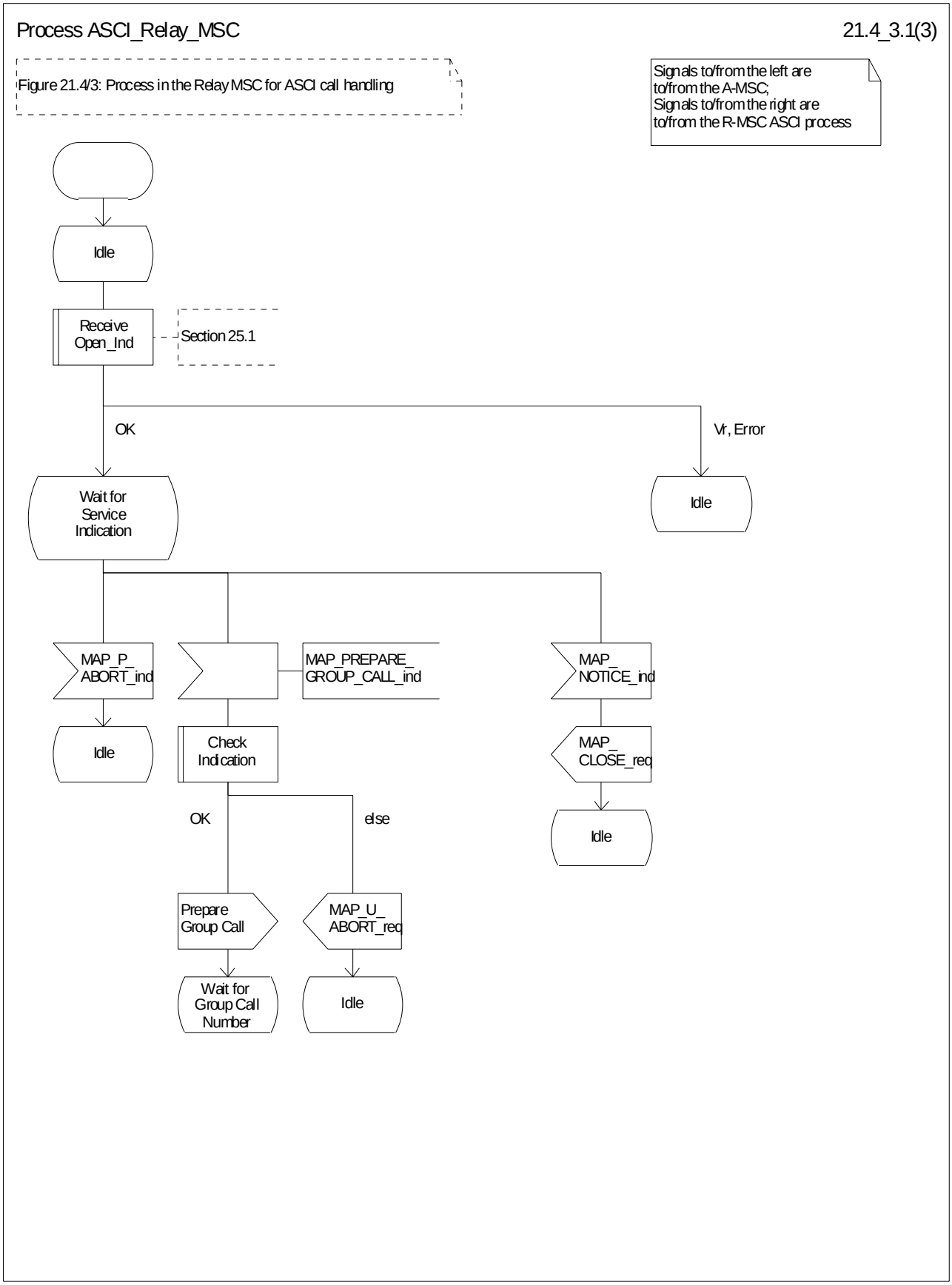


Figure 21.4/3 (sheet 1 of 3): Process ASCI_Relay_MSC

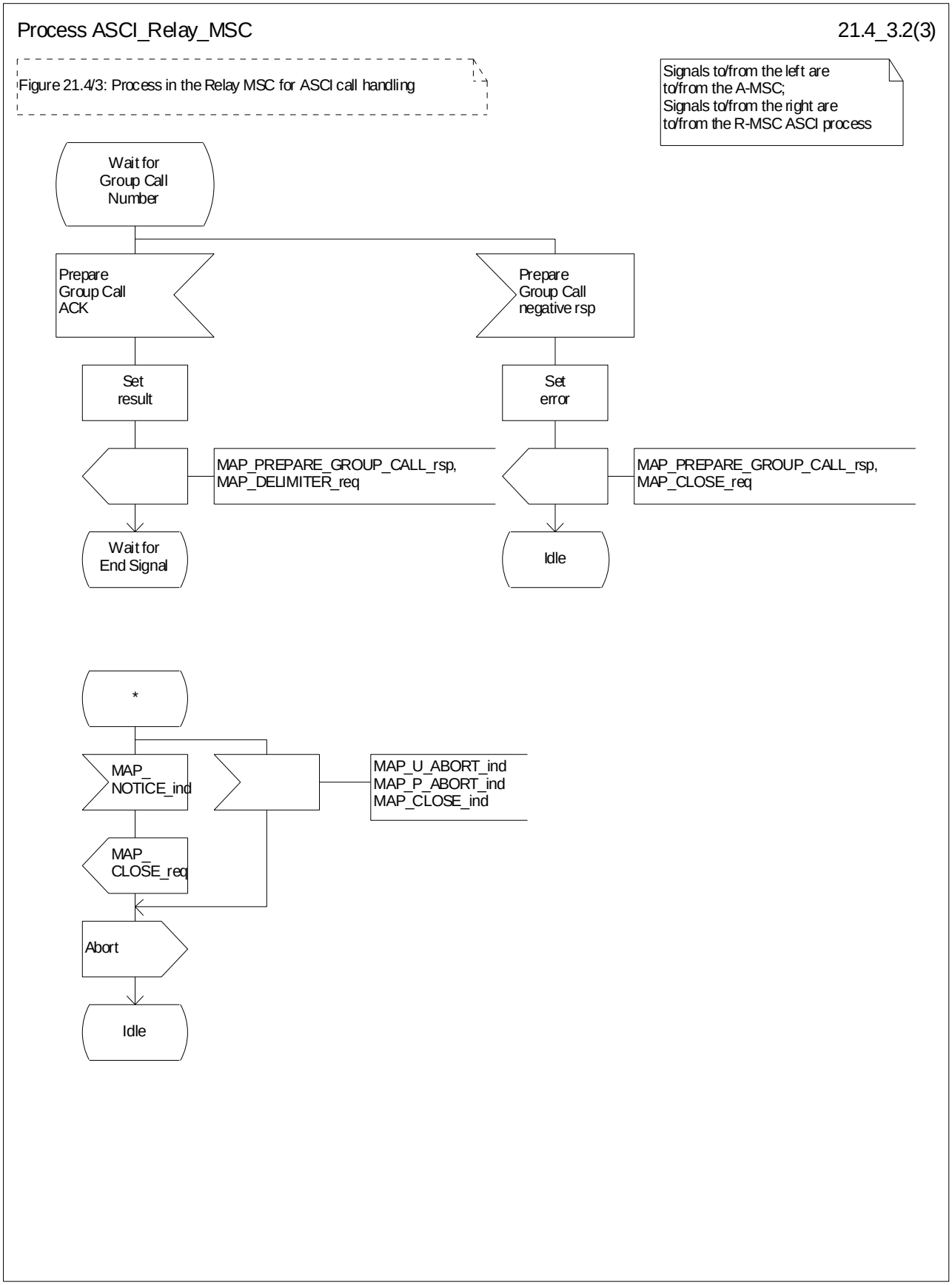


Figure 21.4/3 (sheet 2 of 3): Process ASCI_Relay_MSC

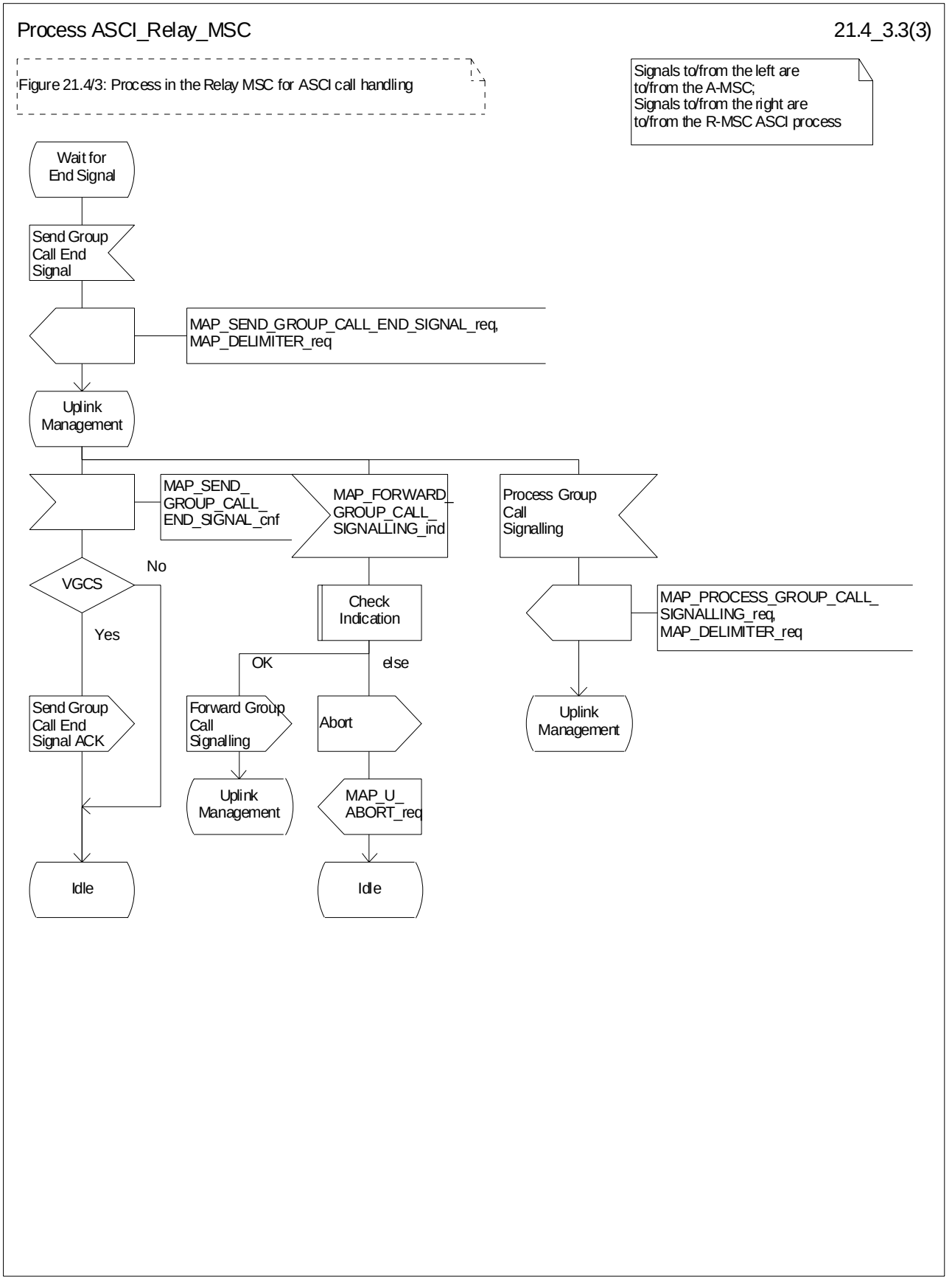


Figure 21.4/3 (sheet 3 of 3): Process ASCI_Relay_MSC

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).

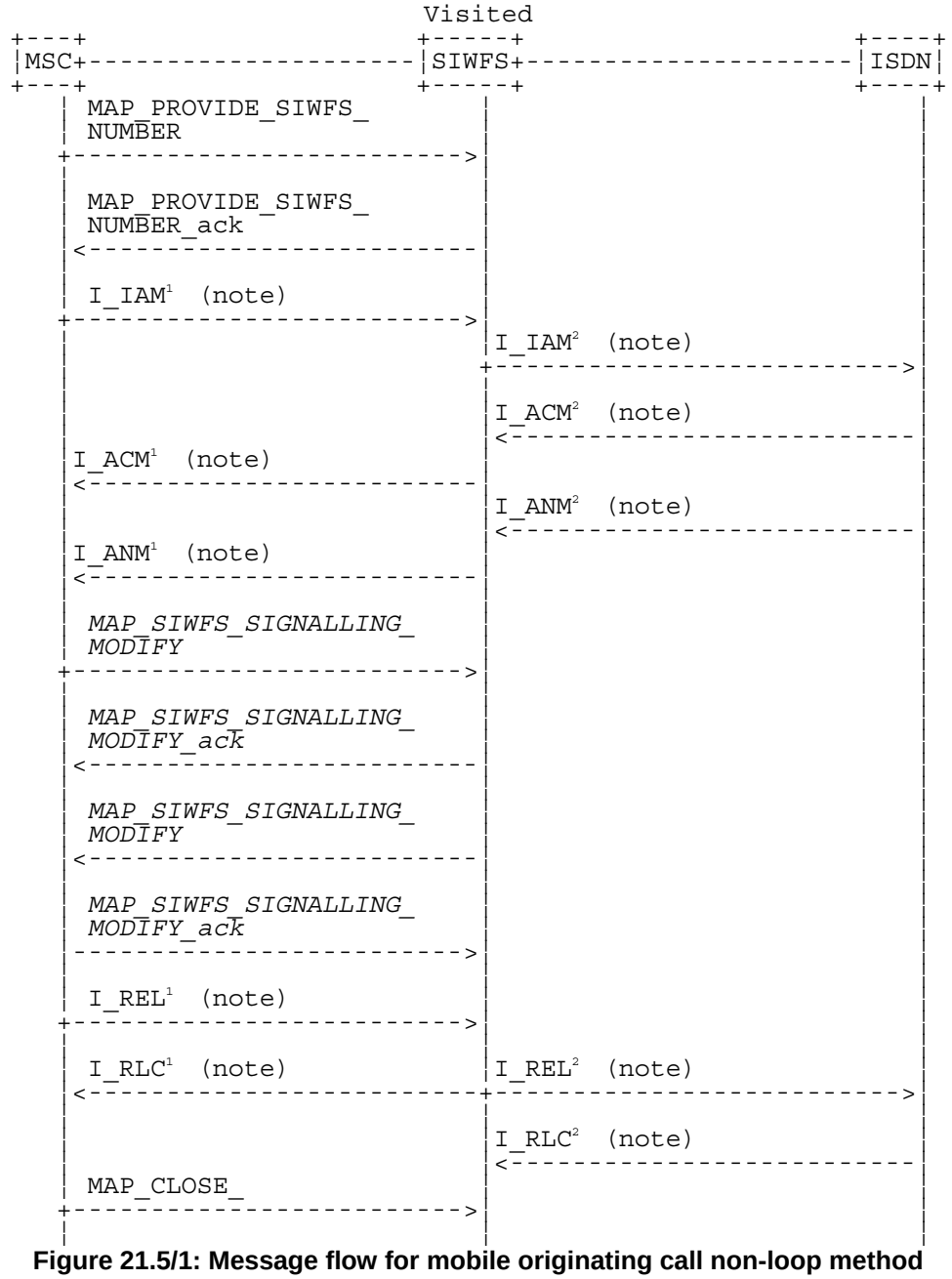


Figure 21.5/1: Message flow for mobile originating call non-loop method

xxx = Optional Procedure

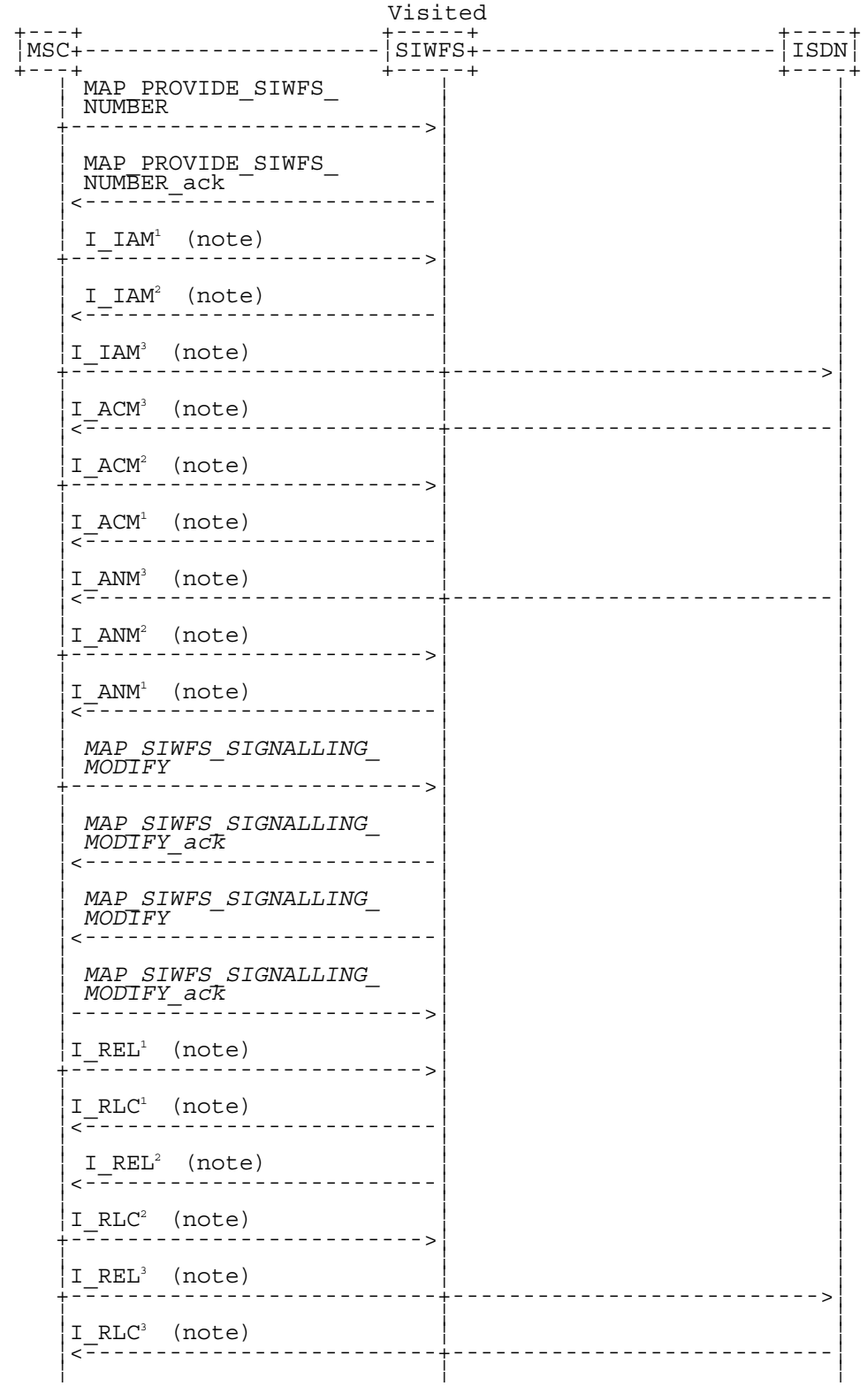
NOTE 1: TUP or ISUP may be used in signalling between MSCs, depending on the network type between the MSCs. The Release message can be initiated either by the calling or called subscriber. For further details on the TUP and ISUP procedures refer to the following CCITT Recommendations & ETSI specification:

- 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.

NOTE 2: The number on the ISUP messages have been added to link the messages to respective signalling sequence.

NOTE 3: The modification of SIWF resources could be initiated any time during the call either by the VMSC or the SIWFS.



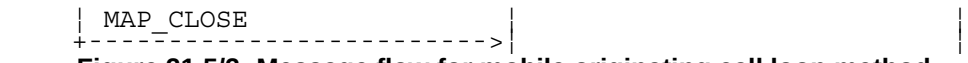


Figure 21.5/2: Message flow for mobile originating call loop method

xxx = Optional Procedure

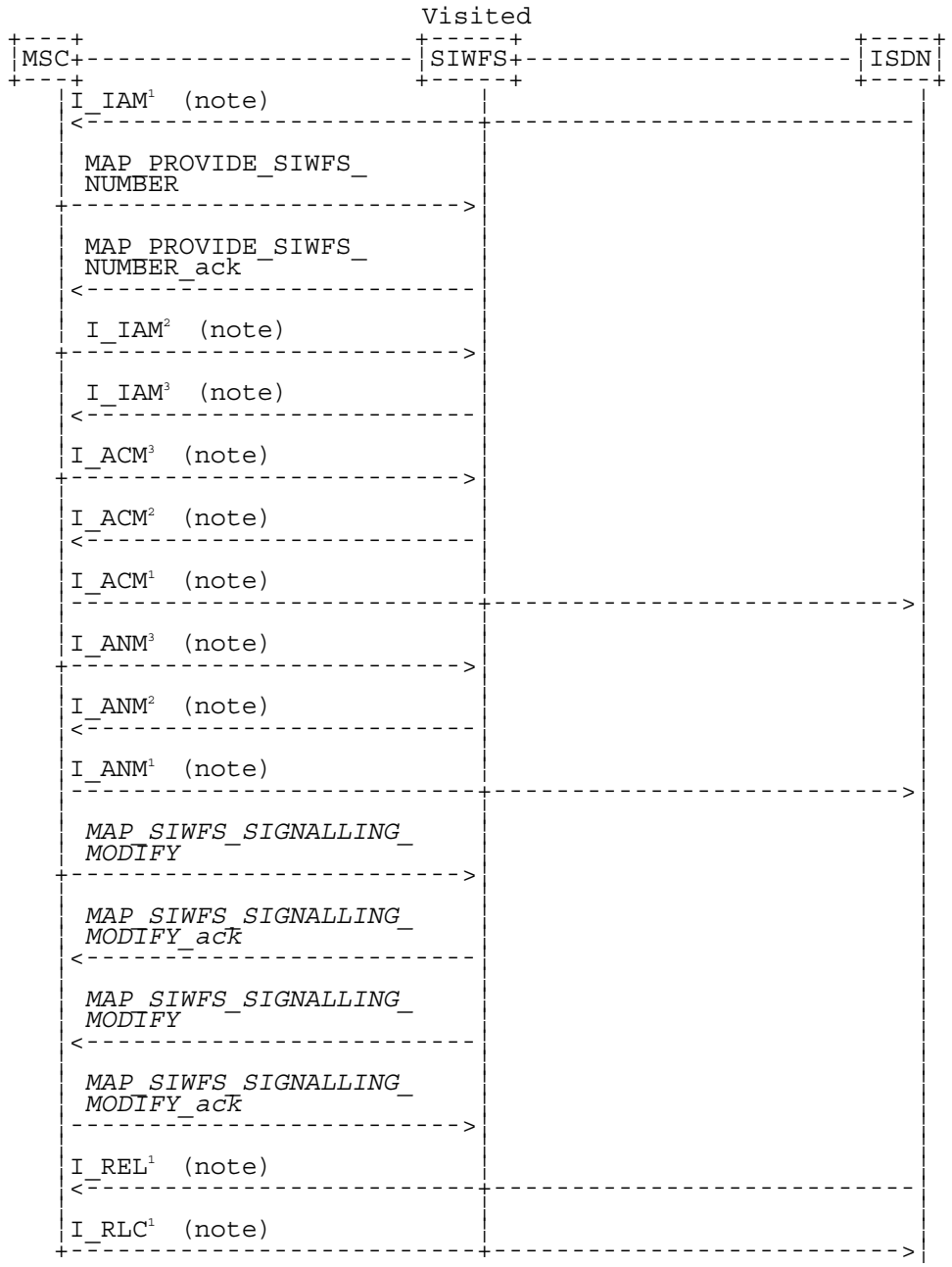
NOTE 1: TUP or ISUP may be used in signalling between MSCs, depending on the network type between the MSCs. The Release message can be initiated either by calling or called subscriber. For further details on the TUP and ISUP procedures refer to the following CCITT Recommendations & ETSI specification:

- 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.

NOTE 2: The number on the ISUP messages have been added to link the messages to respective signalling sequence.

NOTE 3: The modification of SIWF resources could be initiated any time during the call either by the VMSC or the SIWFS.



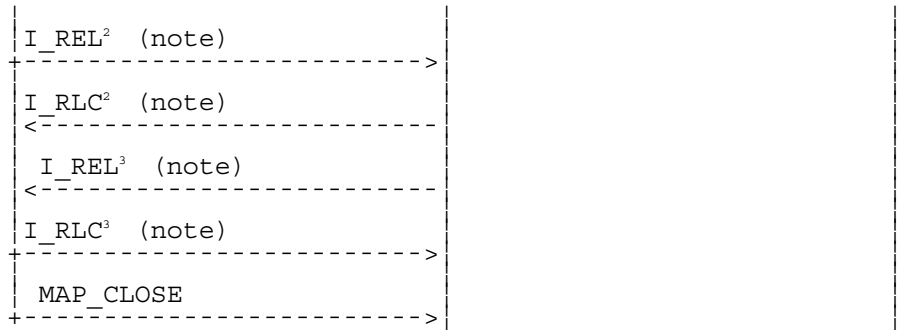


Figure 21.5/3: Message flow for mobile terminating call loop method

xxx = Optional Procedure

NOTE 1: 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.

NOTE 2: The number on the ISUP messages have been added to link the messages to respective signalling sequence.

NOTE 3: The modification of SIWF resources could be initiated any time during the call either by the VMSC or the SIWFS.

The following MAP services are used to allocate resources in an SIWFS:

MAP_PROVIDE_SIWFS_NUMBER see clause 10.8.

The following MAP services are used to modify resources in an SIWFS:

MAP_SIWFS_SIGNALLING_MODIFY see clause 10.9.

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 clause; the definitions of these macros can be found as follows:

Receive_Open_Cnf see clause 25.1.2;

Check_Confirmation see clause 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.

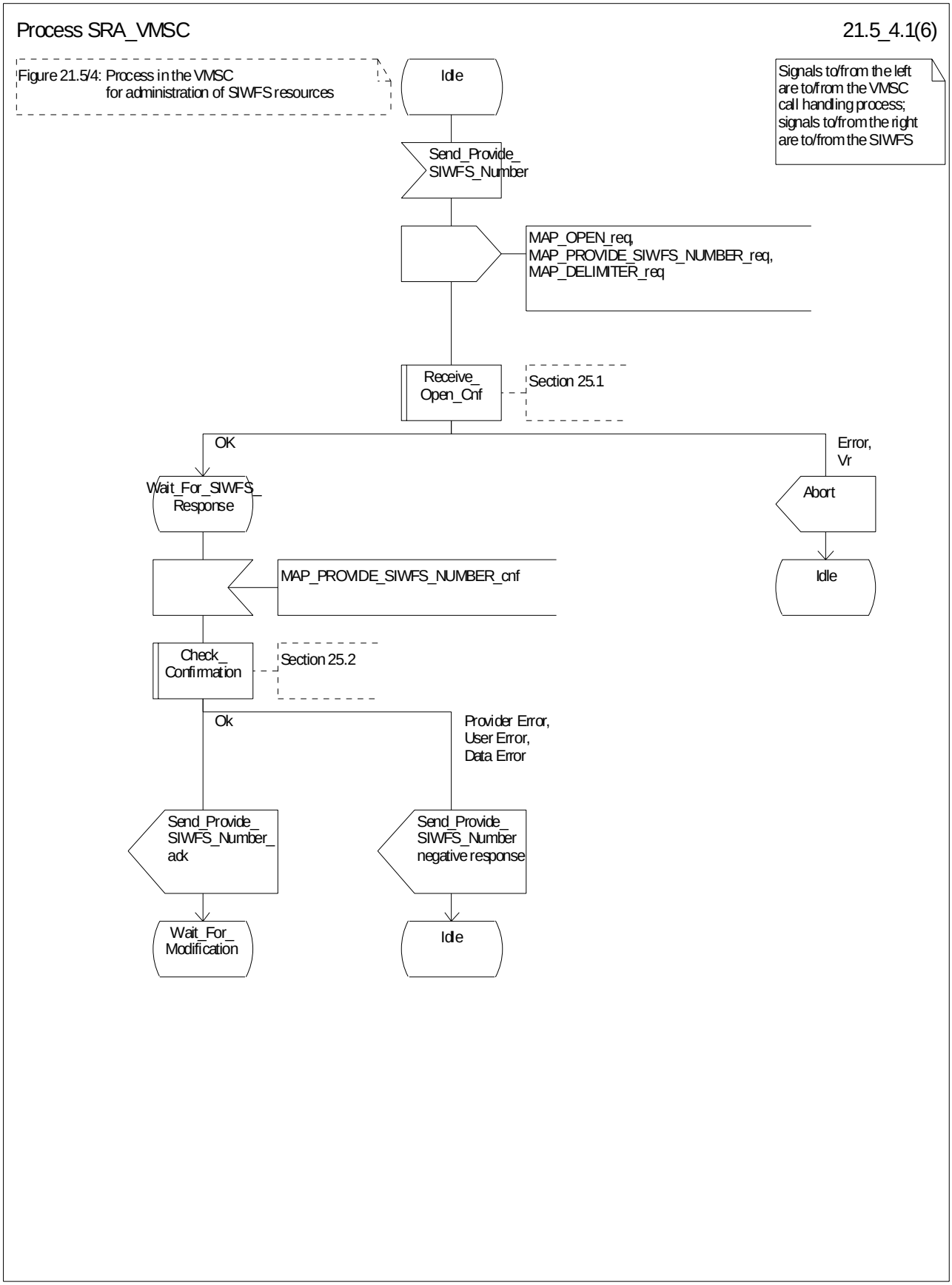


Figure 21.5/4 (sheet 1 of 6): Process SRA (SIWFS_RESOURCE_ADMINISTRATION)_VMSC

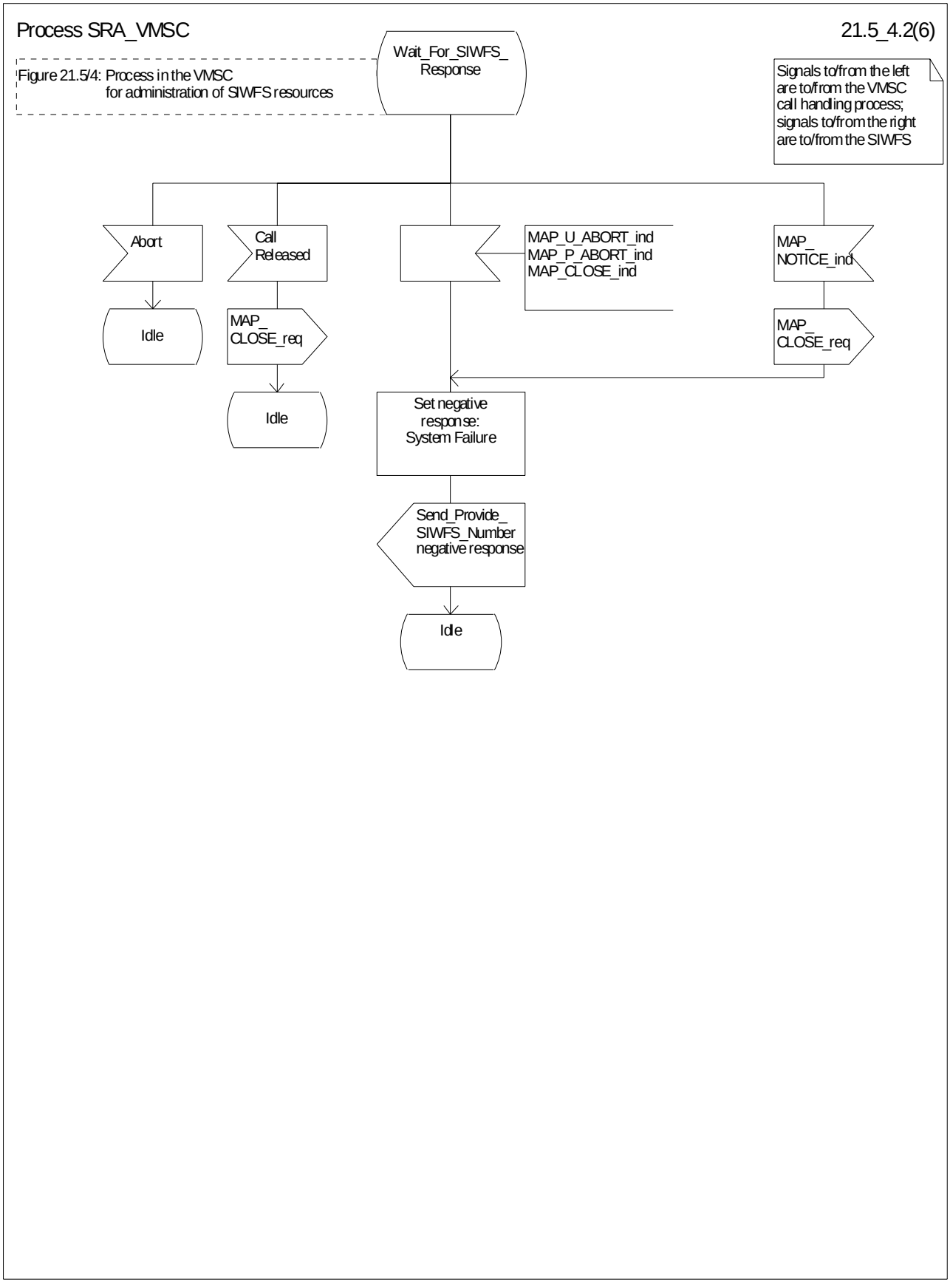


Figure 21.5/4 (sheet 2 of 6): Process SRA_VMSC

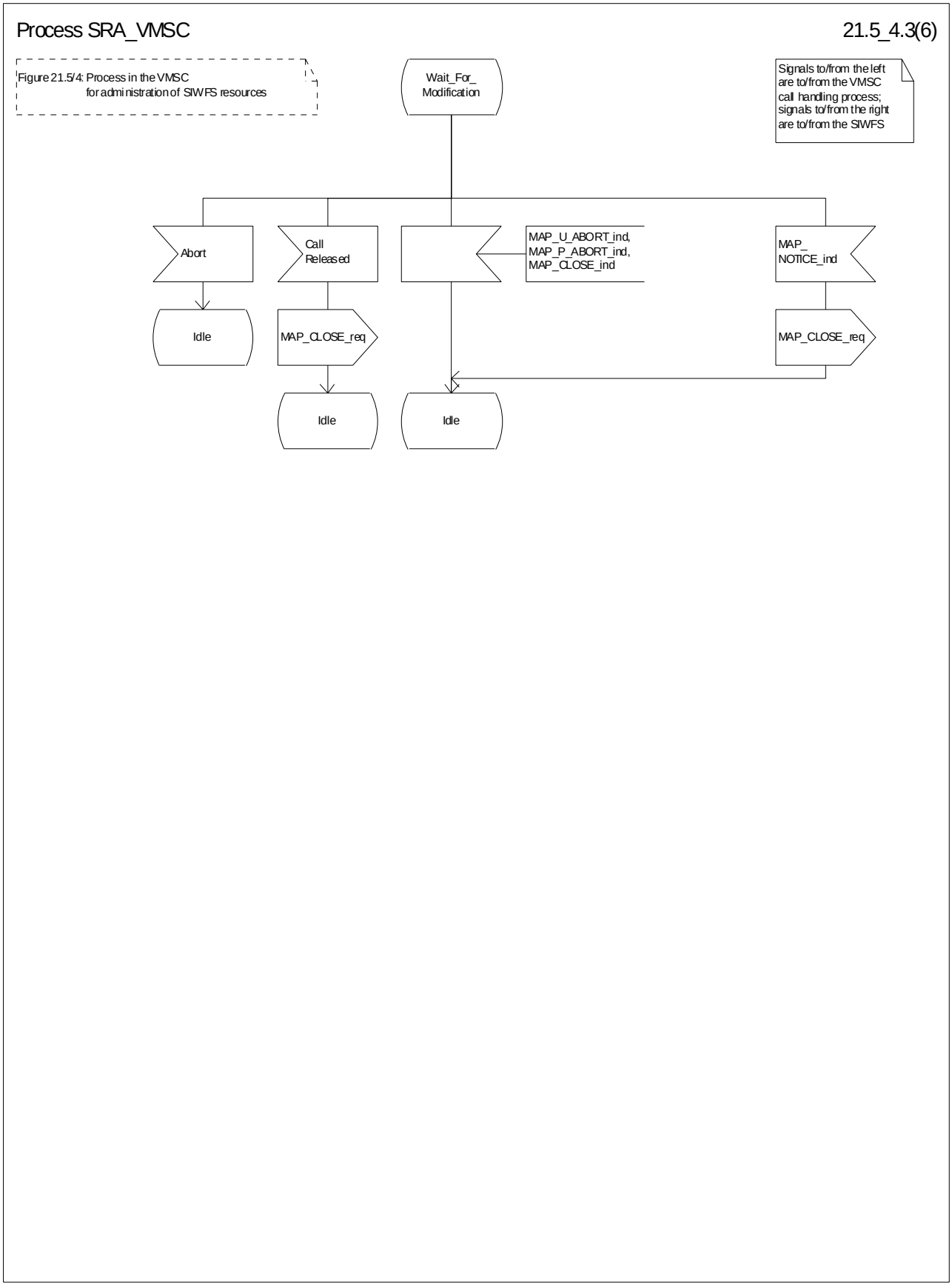


Figure 21.5/4 (sheet 3 of 6): Process SRA_VMSC

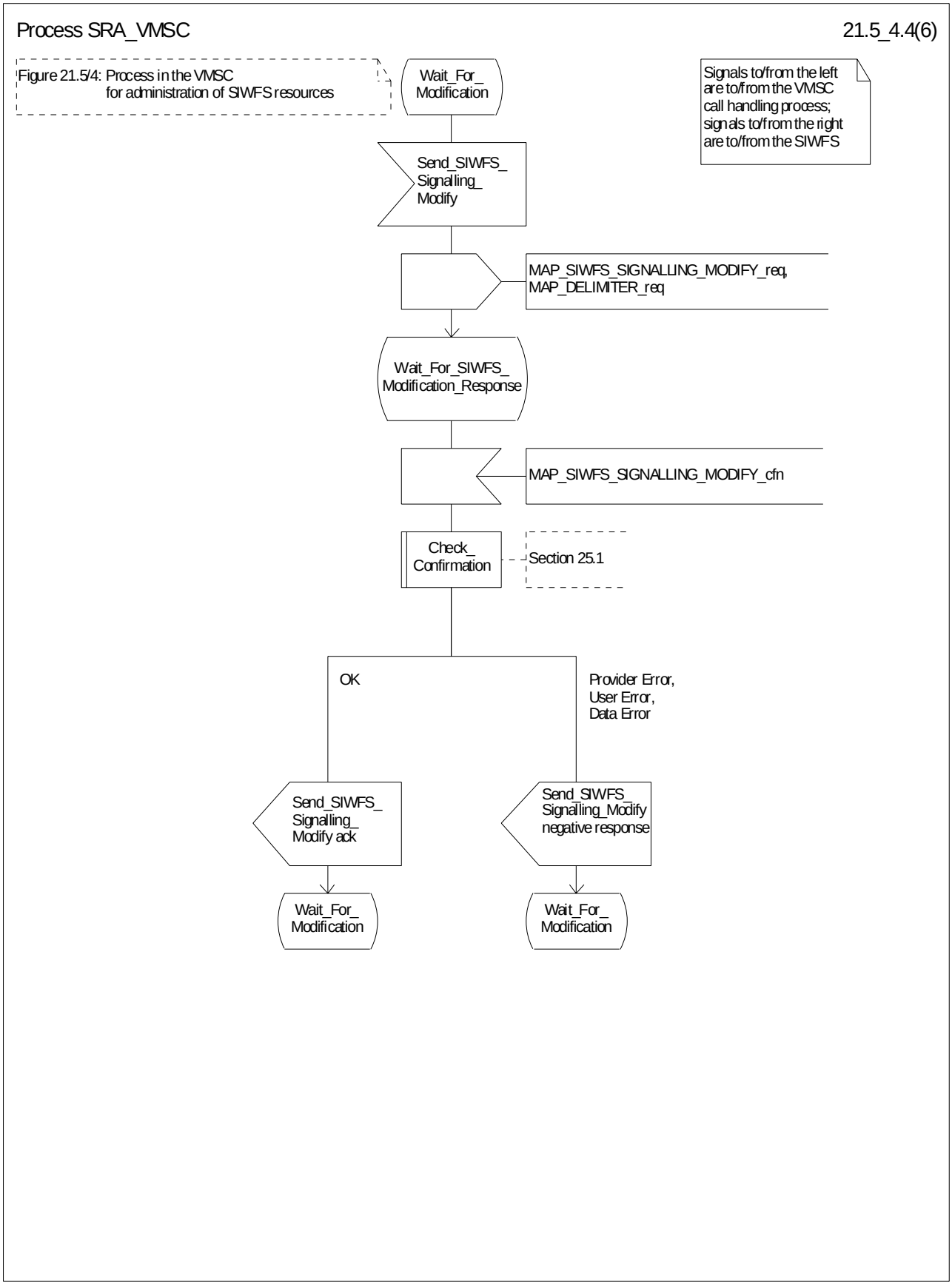


Figure 21.5/4 (sheet 4 of 6): Process SRA_VMSC

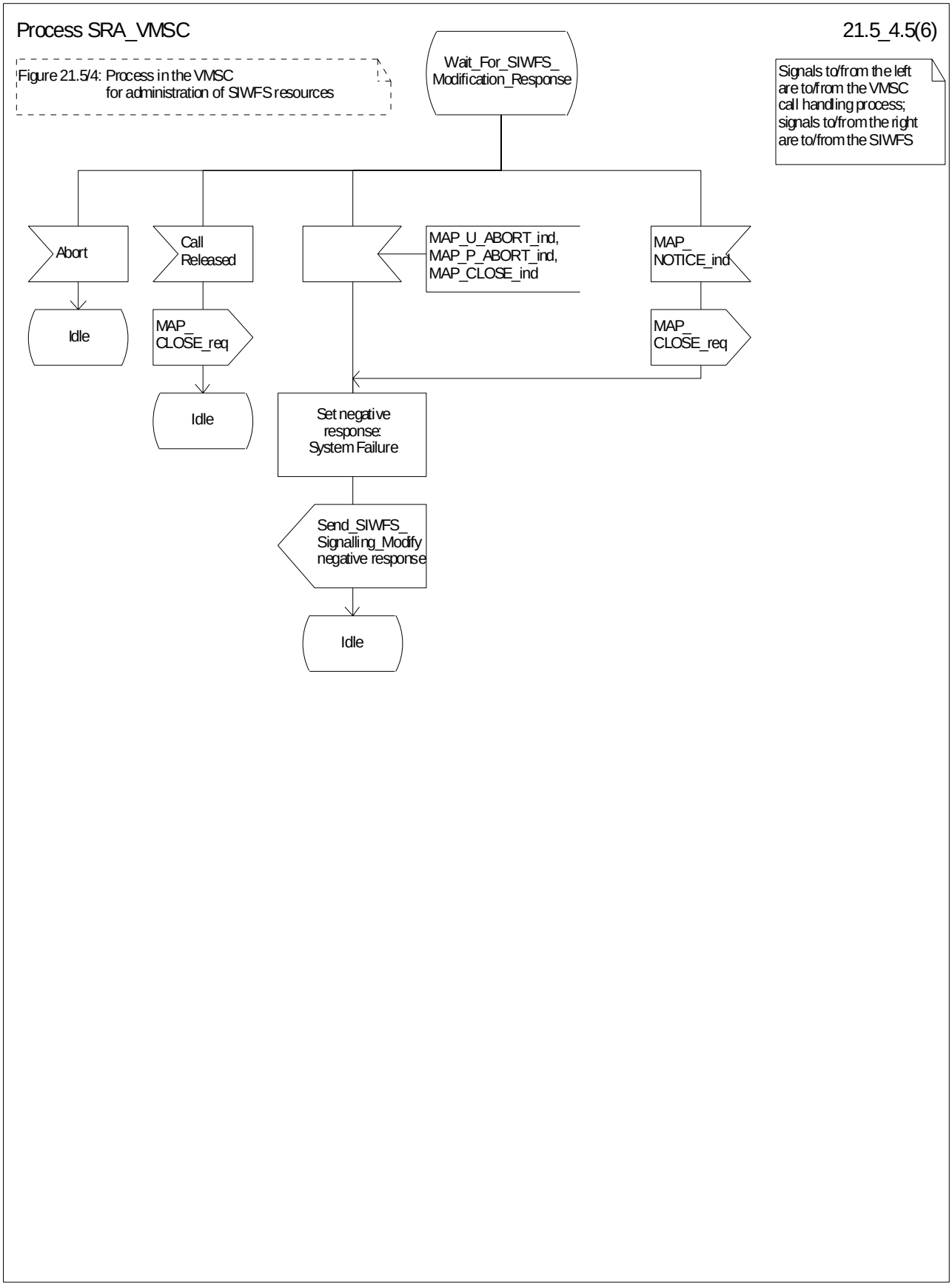


Figure 21.5/4 (sheet 5 of 6): Process SRA_VMSC

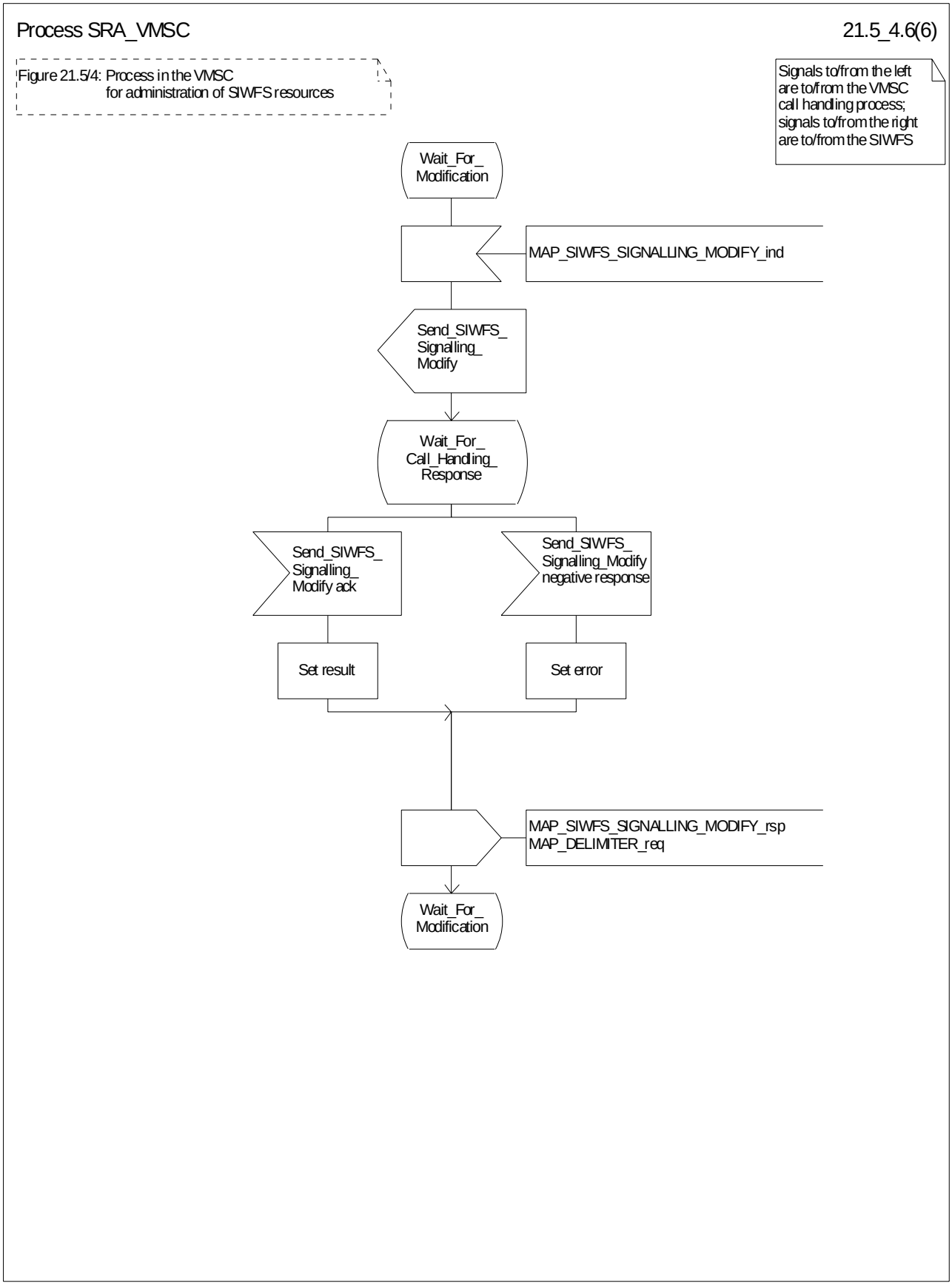


Figure 21.5/4 (sheet 6 of 6): Process SRA_VMSC

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 clause; the definitions of these macros can be found as follows:

Receive_Open_Ind	see clause 25.1.1.
Check_Confirmation	see clause 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.

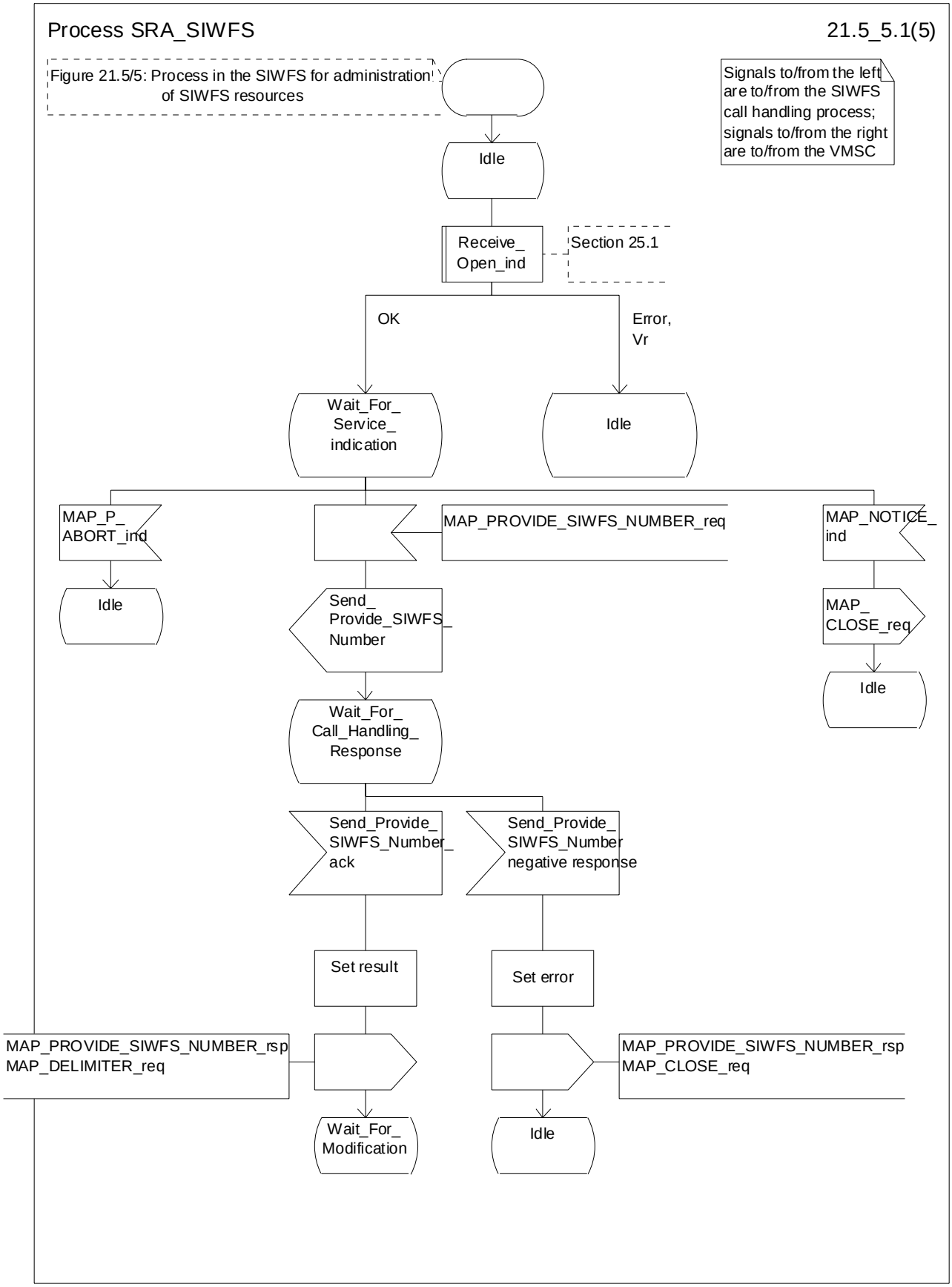


Figure 21.5/5 (sheet 1 of 5): Process SRA_SIWFS

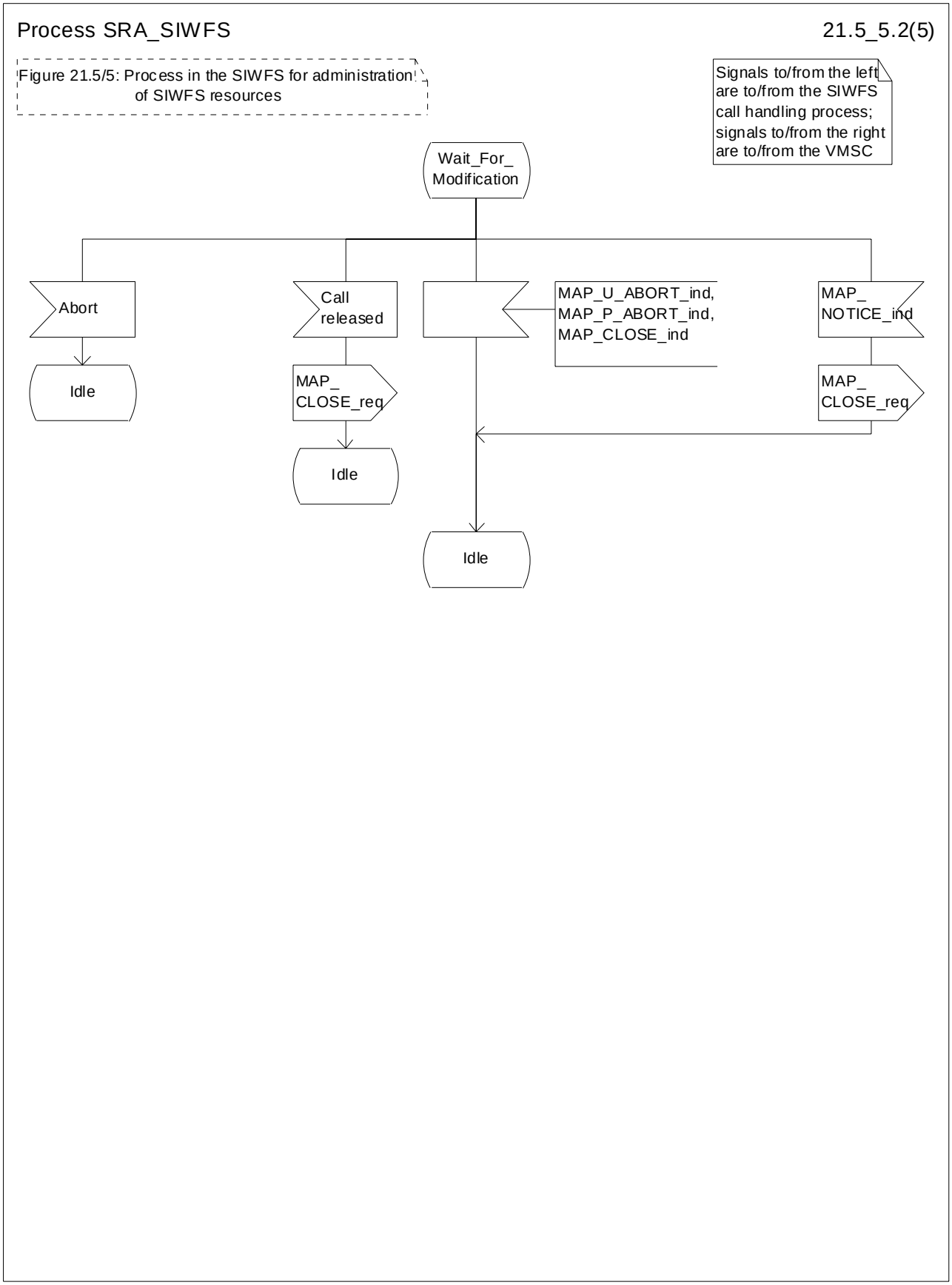


Figure 21.5/5 (sheet 2 of 5): Process SRA_SIWFS

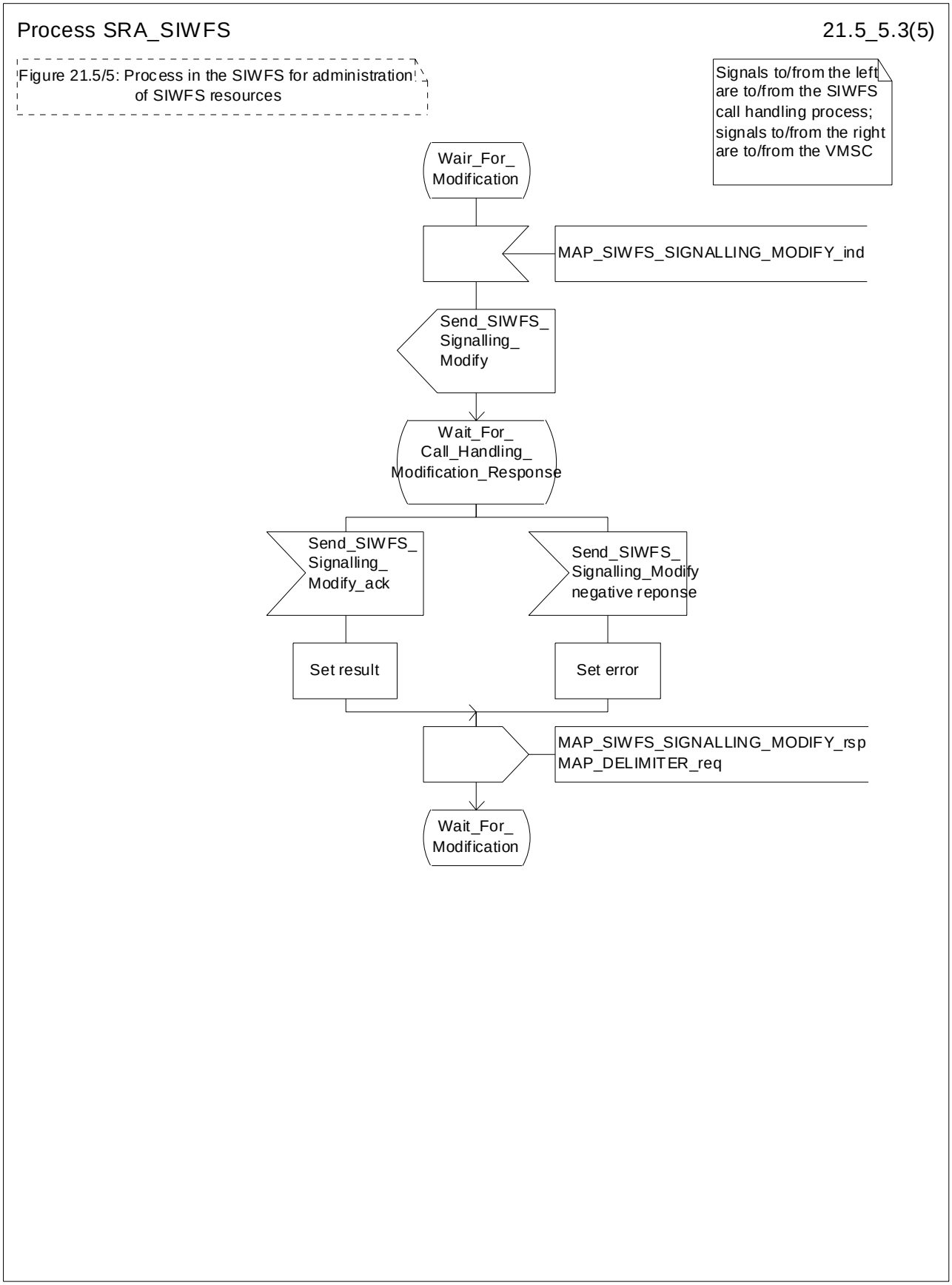


Figure 21.5/5 (sheet 3 of 5): Process SRA_SIWFS

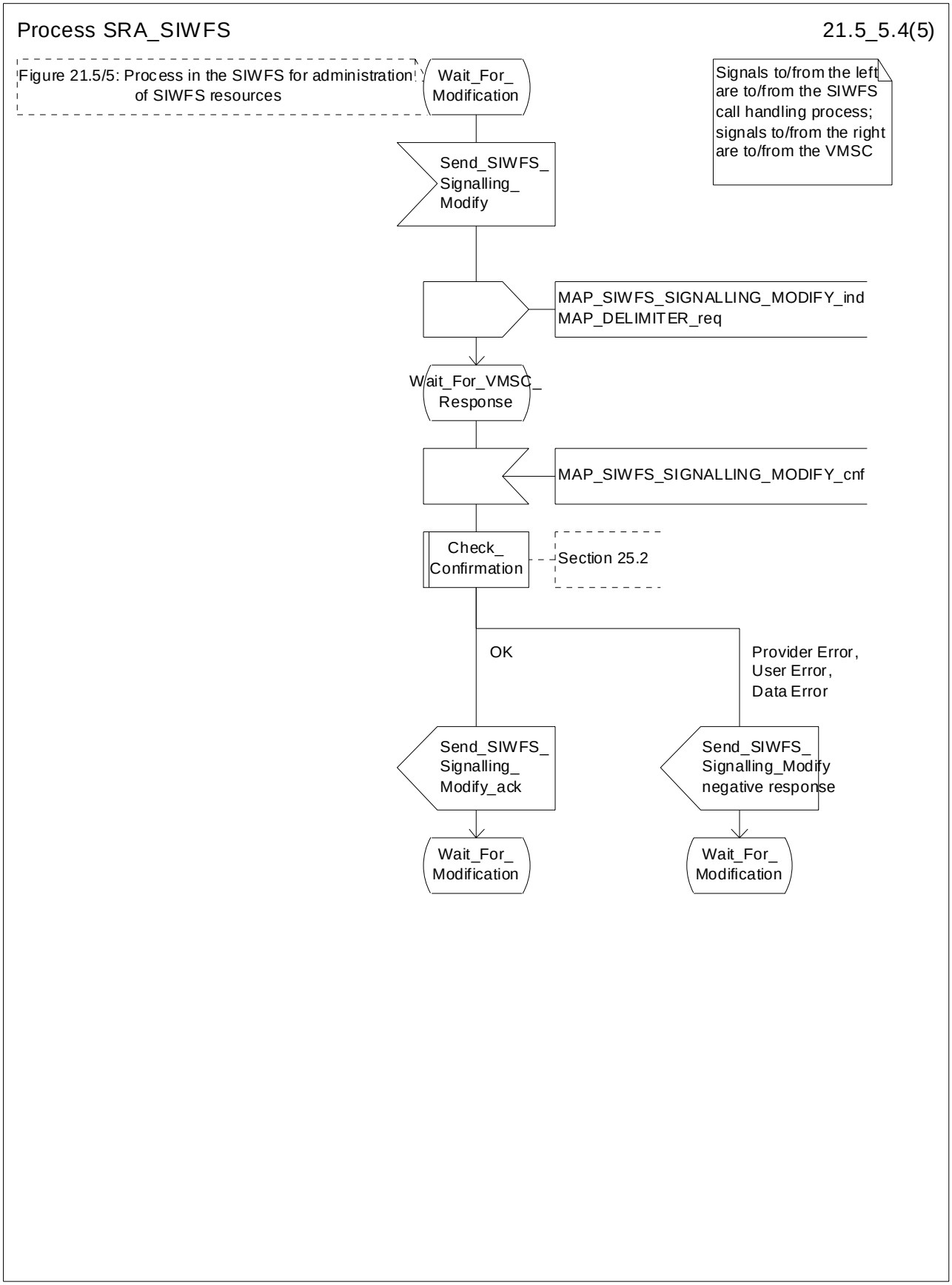


Figure 21.5/5 (sheet 4 of 5): Process SRA_SIWFS

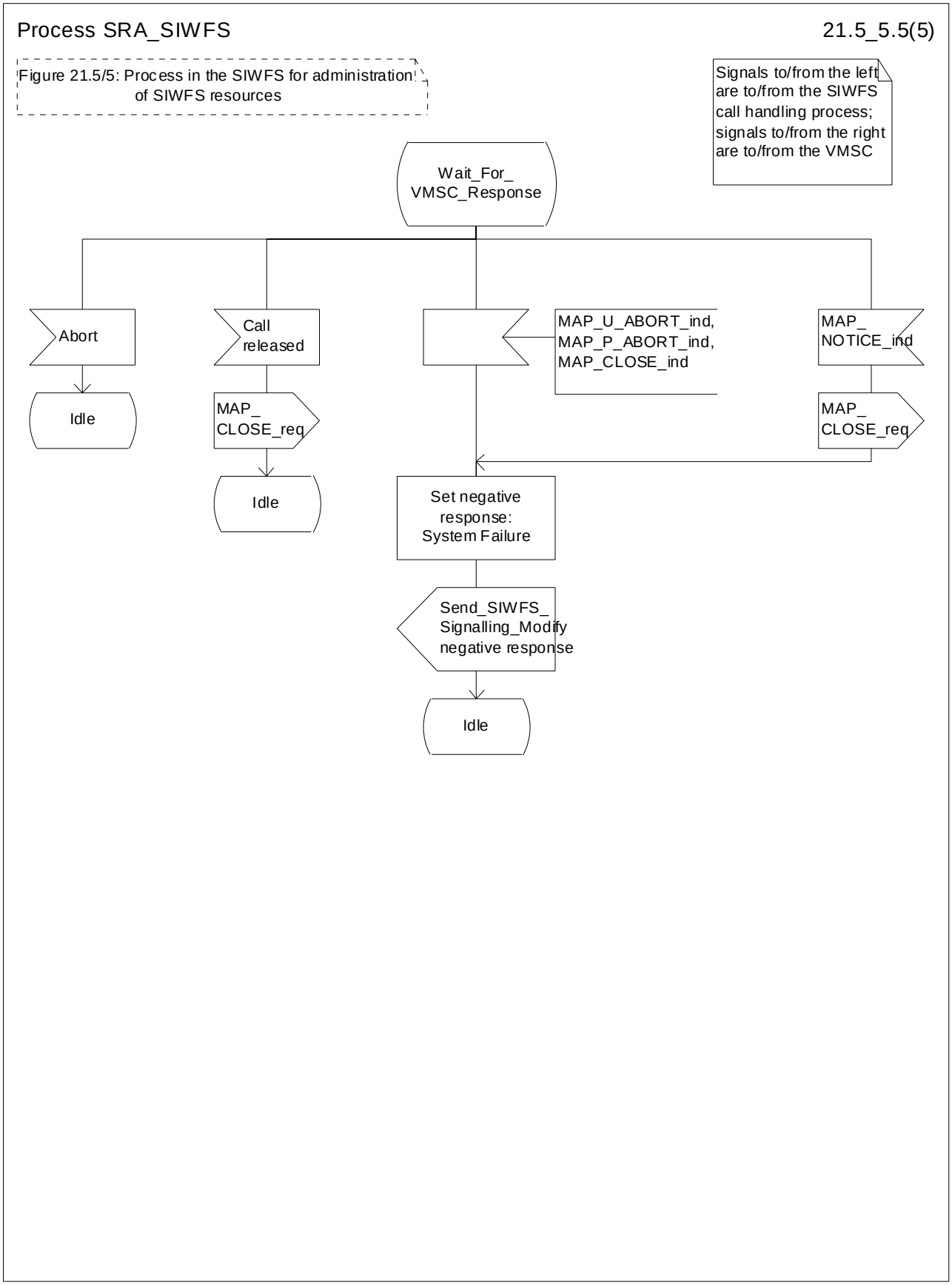
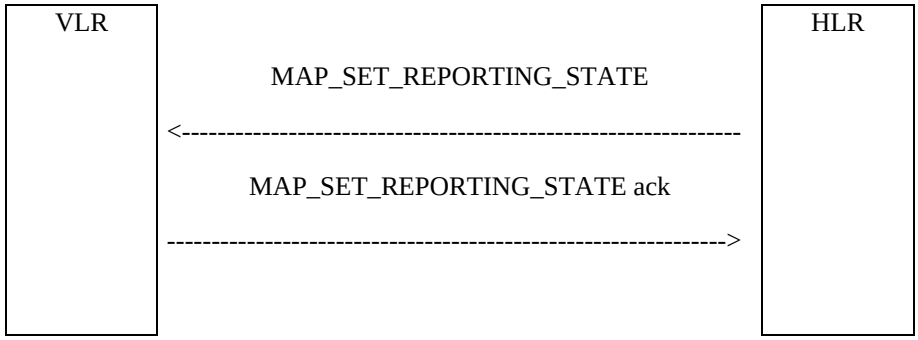


Figure 21.5/5 (sheet 5 of 5): Process SRA_SIWFS

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.



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 clause; the definitions of these macros can be found as follows:

Receive_Open_Cnf see clause 25.1.2;

Check_Confirmation see clause 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.

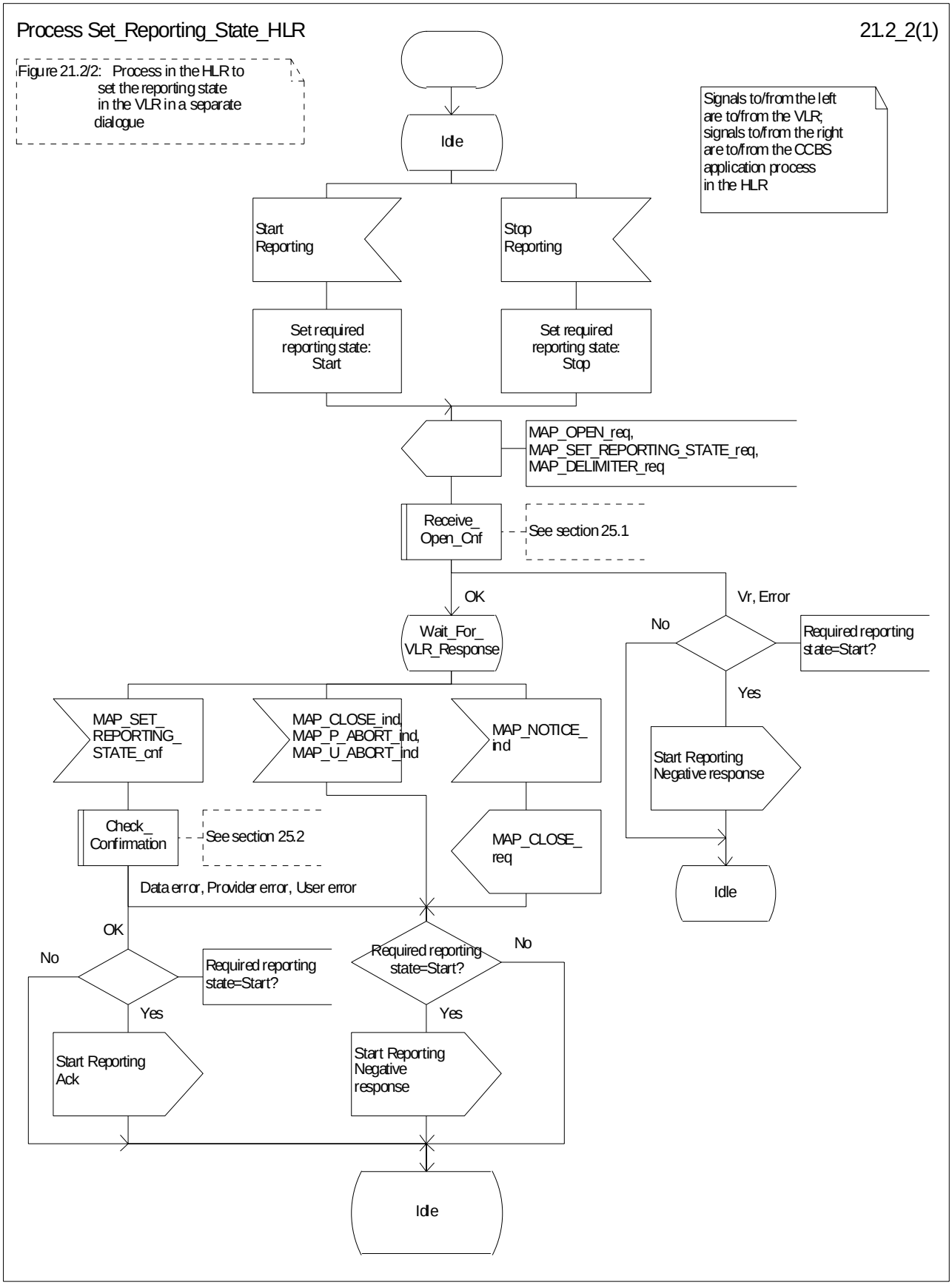


Figure 21.6/2: Process Set_Reporting_State_HLR

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 clause; the definition of this macro can be found as follows:

Receive_Open_Ind see clause 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.

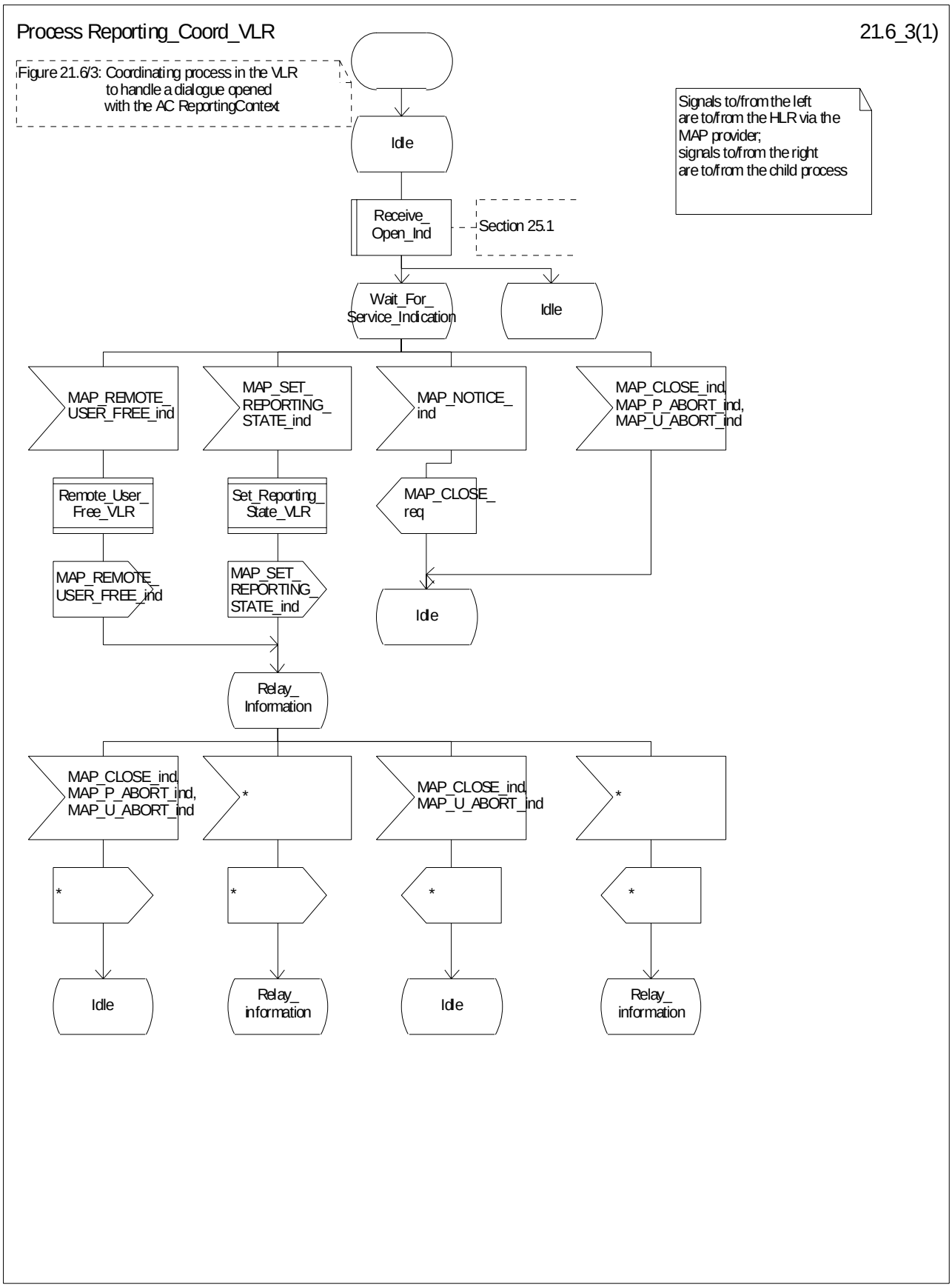


Figure 21.6/3: Process Reporting_Coord_VLR

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.

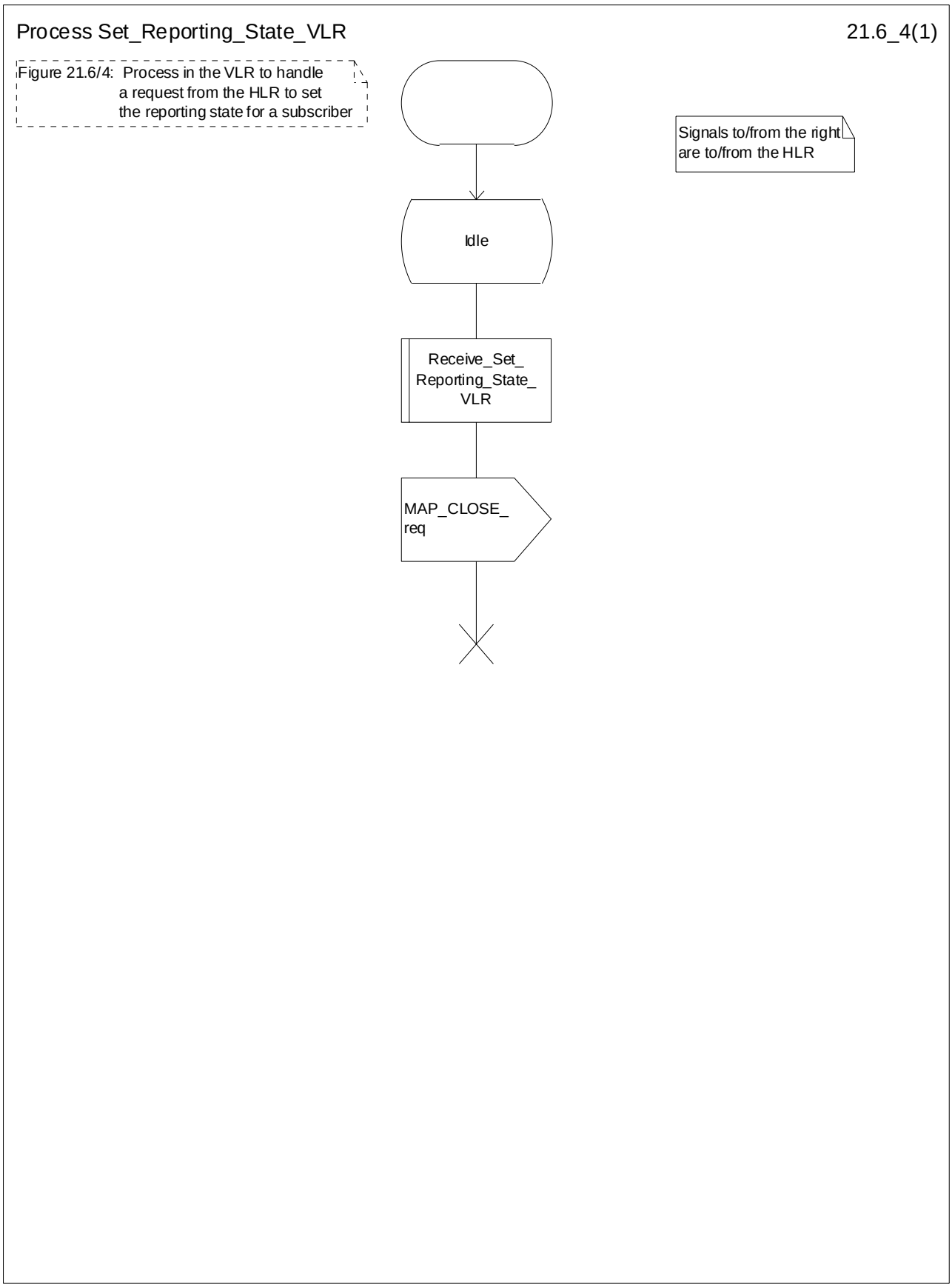


Figure 21.6/4: Process Set_Reporting_State_VLR

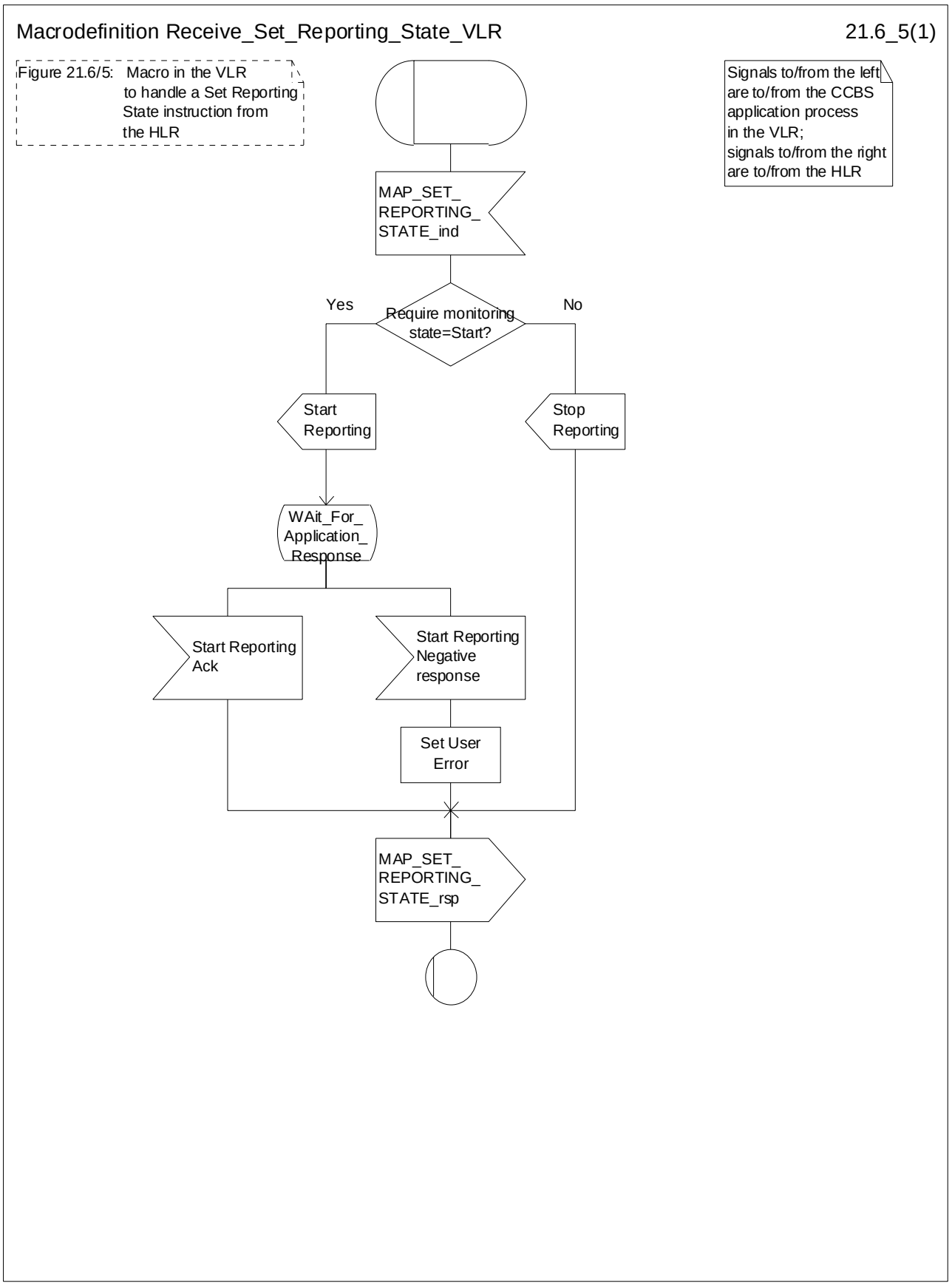


Figure 21.6/5: Macro Receive_Set_Reporting_State_VLR

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.

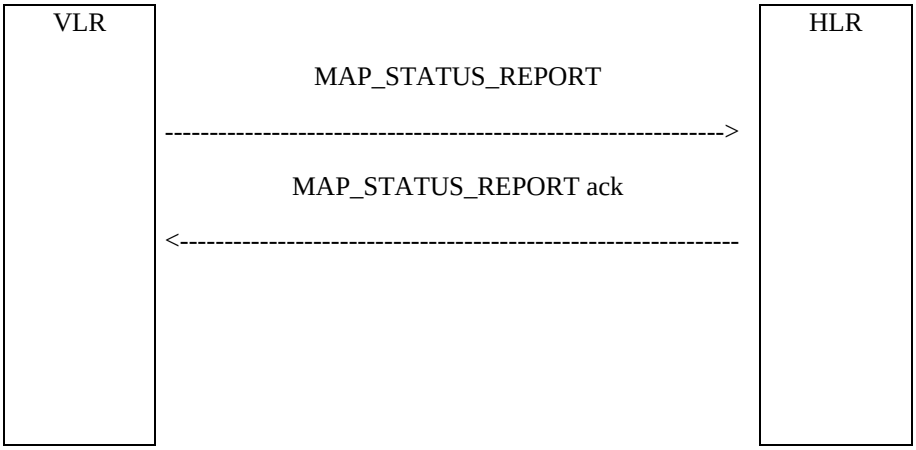


Figure 21.7/1: Status reporting, when monitoring continues in the VLR

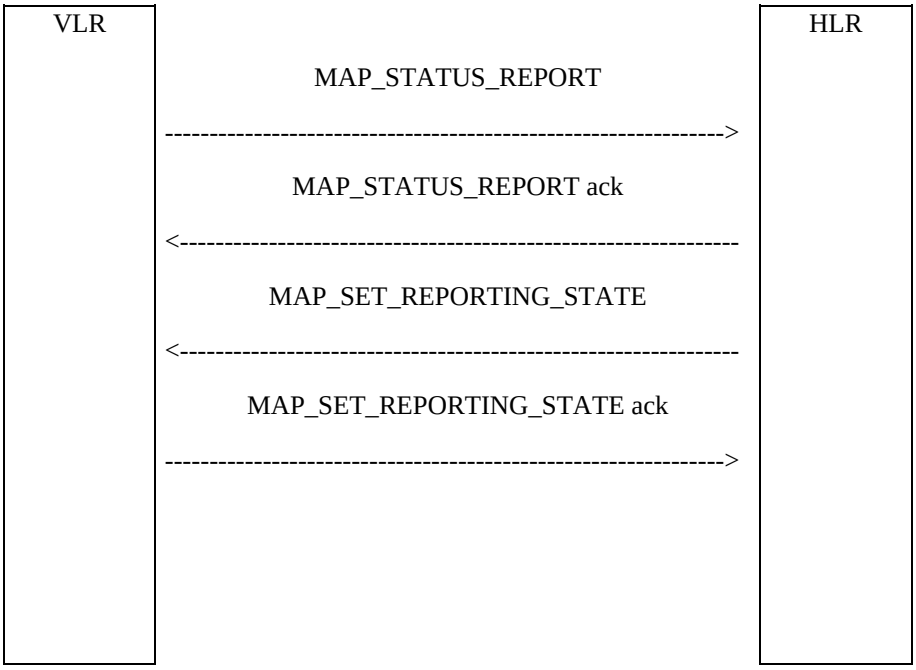


Figure 21.7/2: Status reporting, when monitoring stops

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 clause; the definitions of these macros can be found as follows:

Receive_Open_Cnf see clause 25.1.2;

Check_Confirmation see clause 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.

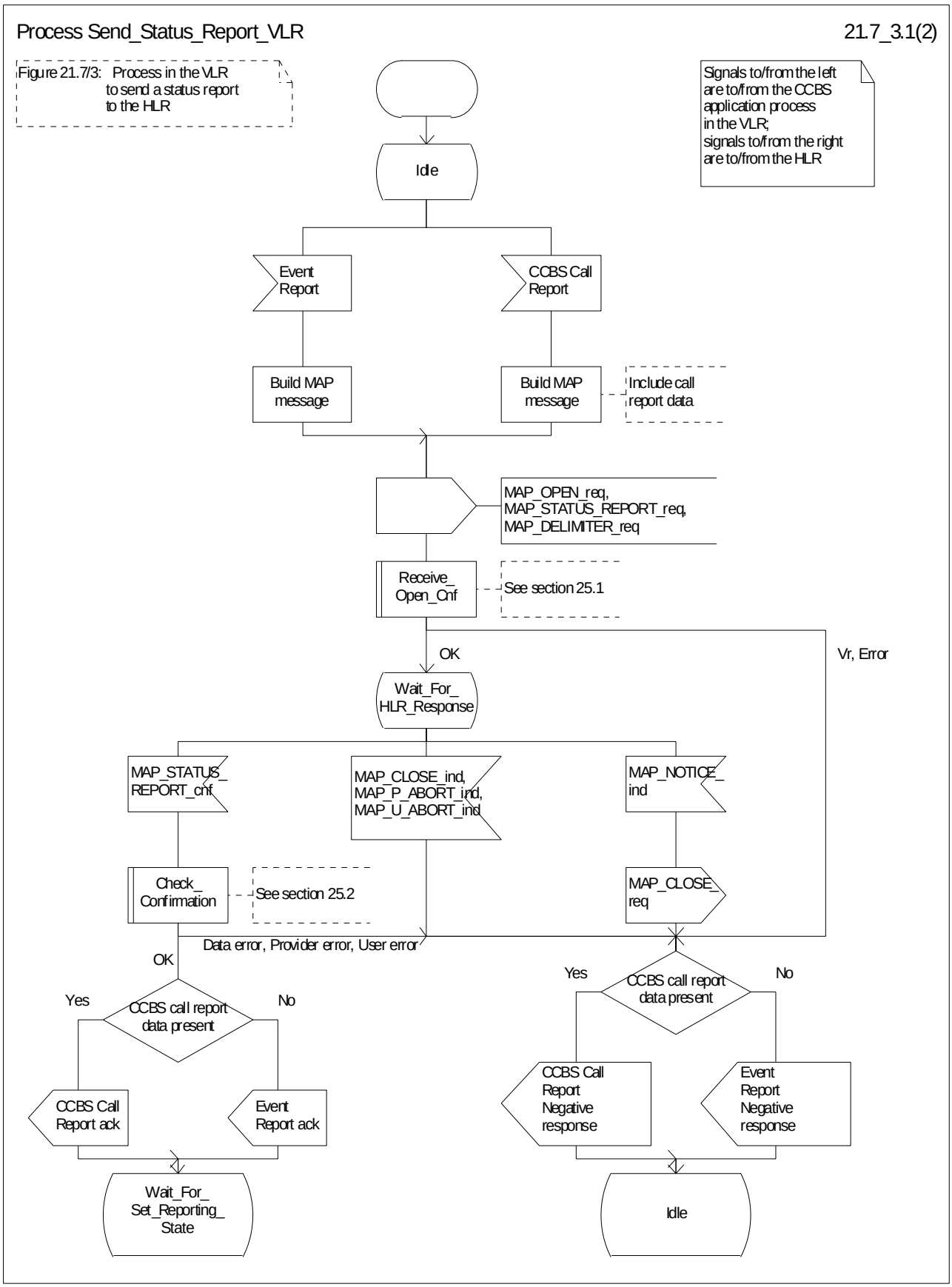


Figure 21.7/3 (sheet 1 of 2): Process Send_Status_Report_VLR

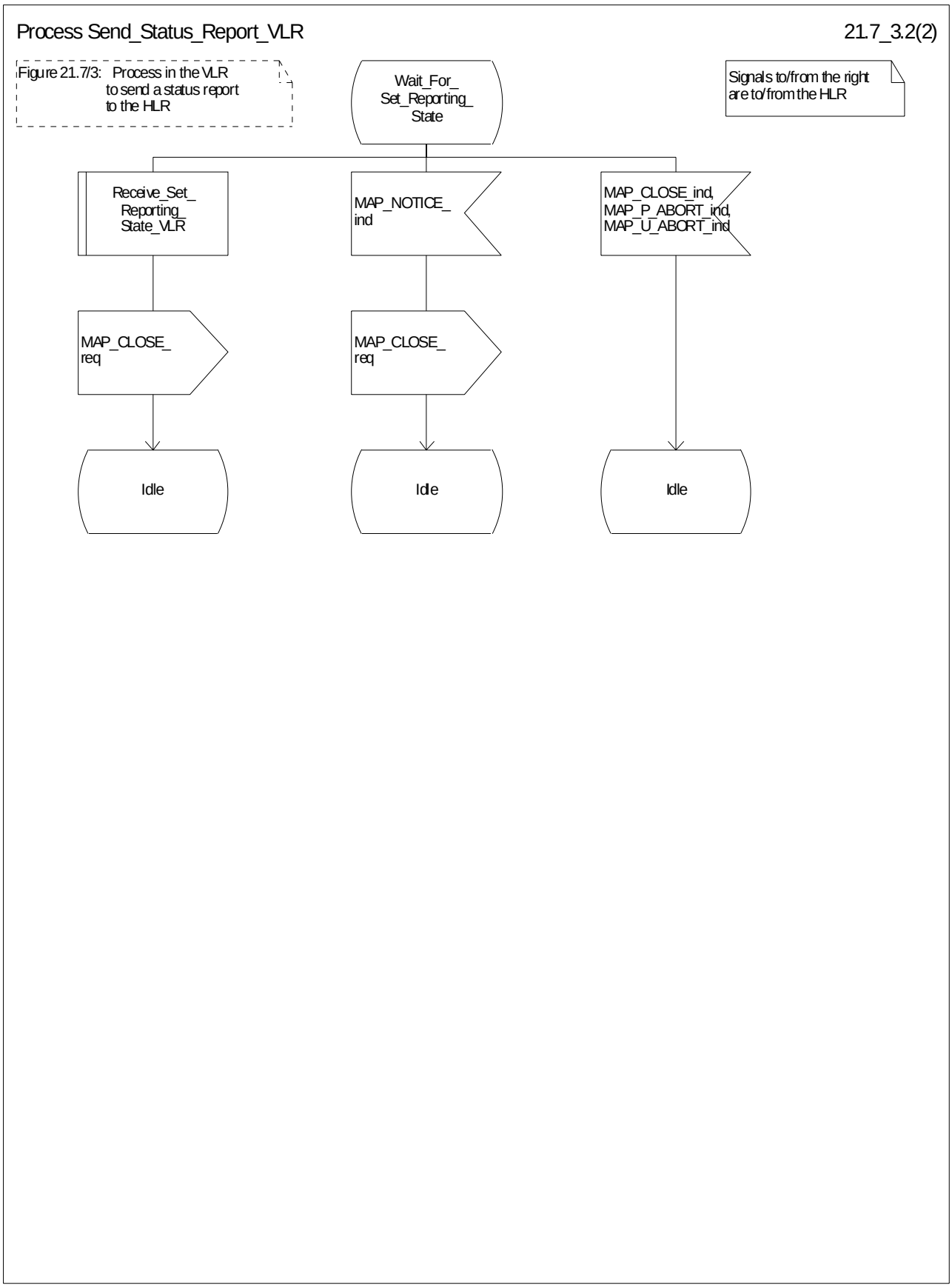


Figure 21.7/3 (sheet 2 of 2): Process Send_Status_Report_VLR

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 clause; the definitions of these macros can be found as follows:

Receive_Open_Ind see clause 25.1.1;

Check_Confirmation see clause 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.

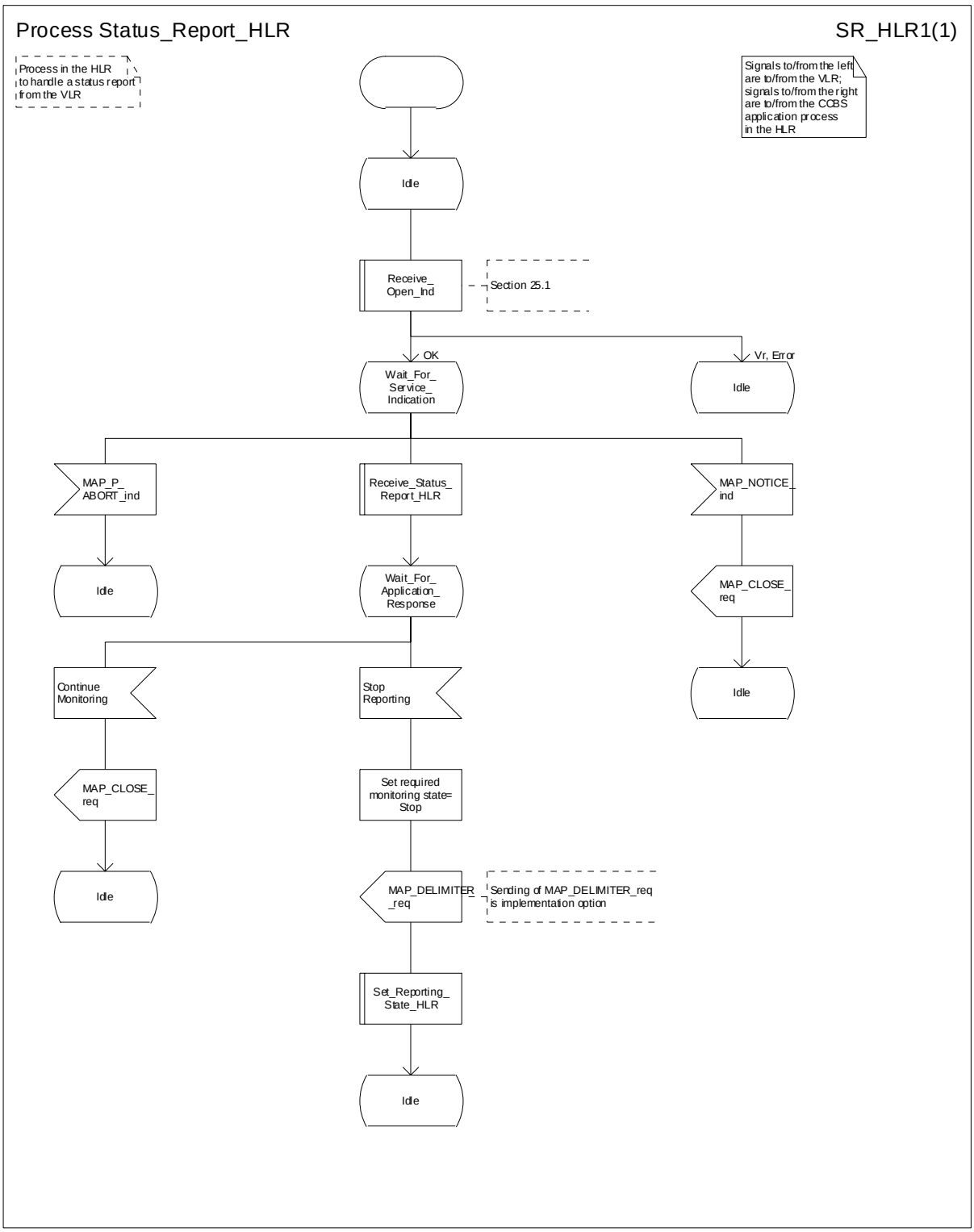


Figure 21.7/4: Process Status Report_HLR

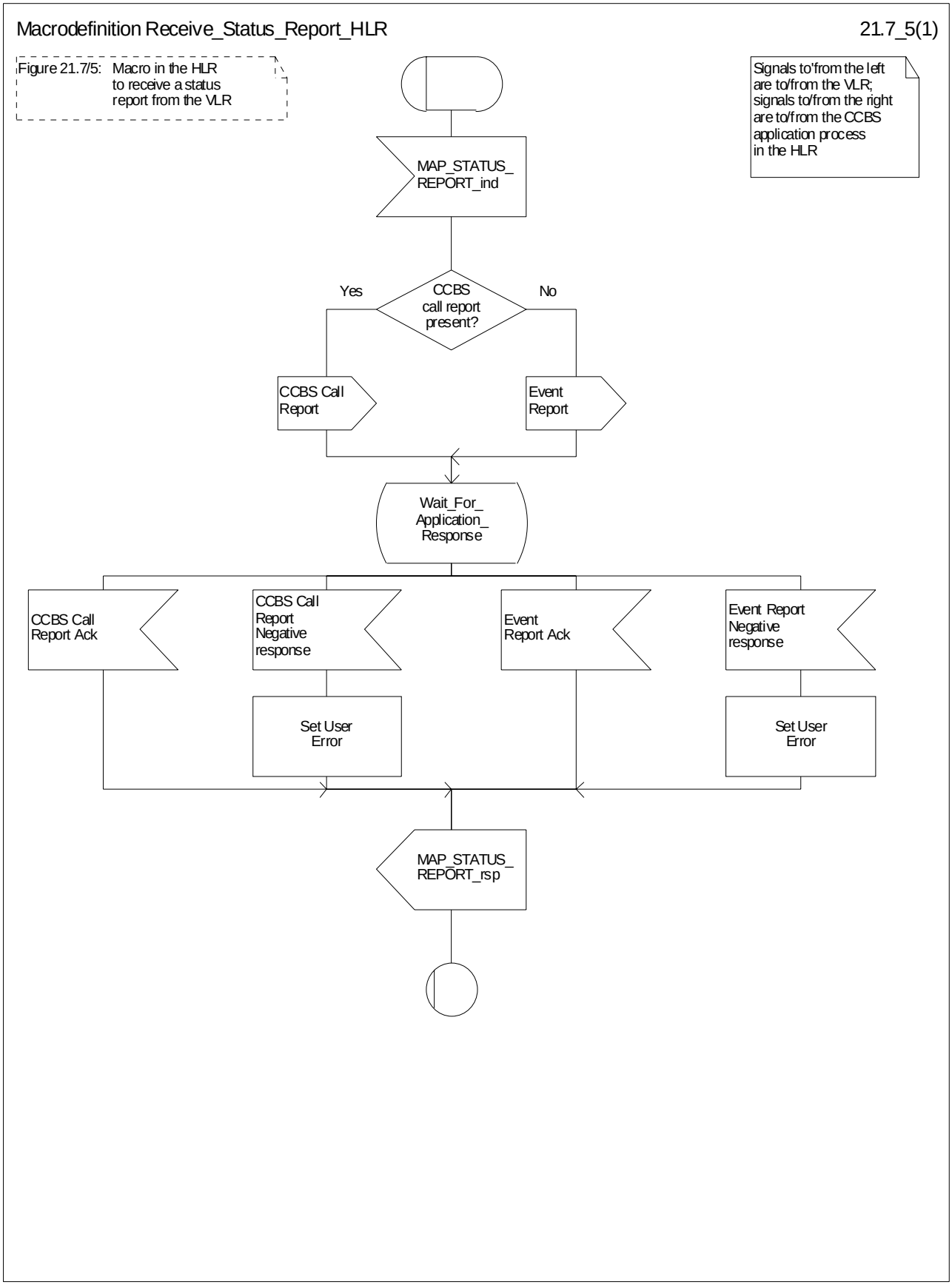


Figure 21.7/5: Macro Receive_Status_Report_HLR

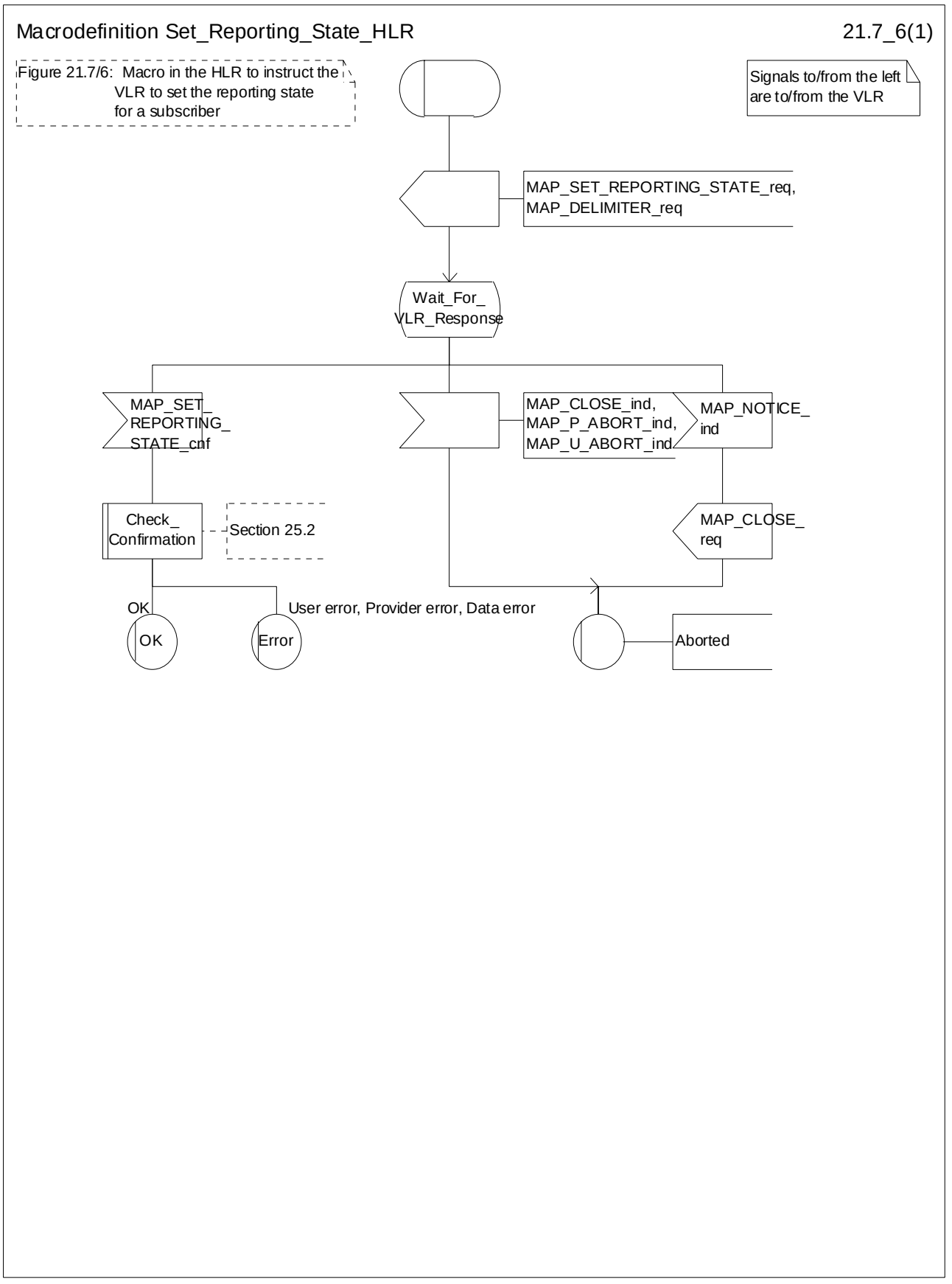


Figure 21.7/6: Macro Set_Reporting_State_HLR

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.

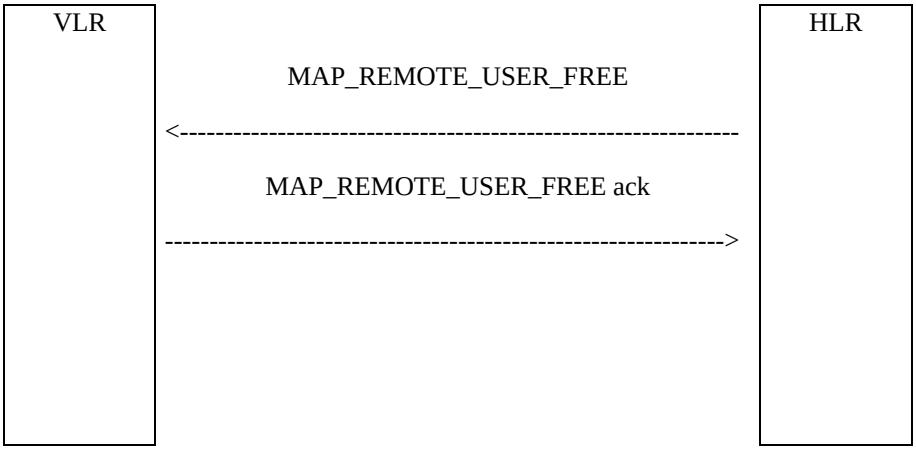


Figure 21.8/1: Remote User Free: recall not accepted

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 clause; the definitions of these macros can be found as follows:

Receive_Open_Cnf see clause 25.1.2;

Check_Confirmation see clause 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 clause 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.

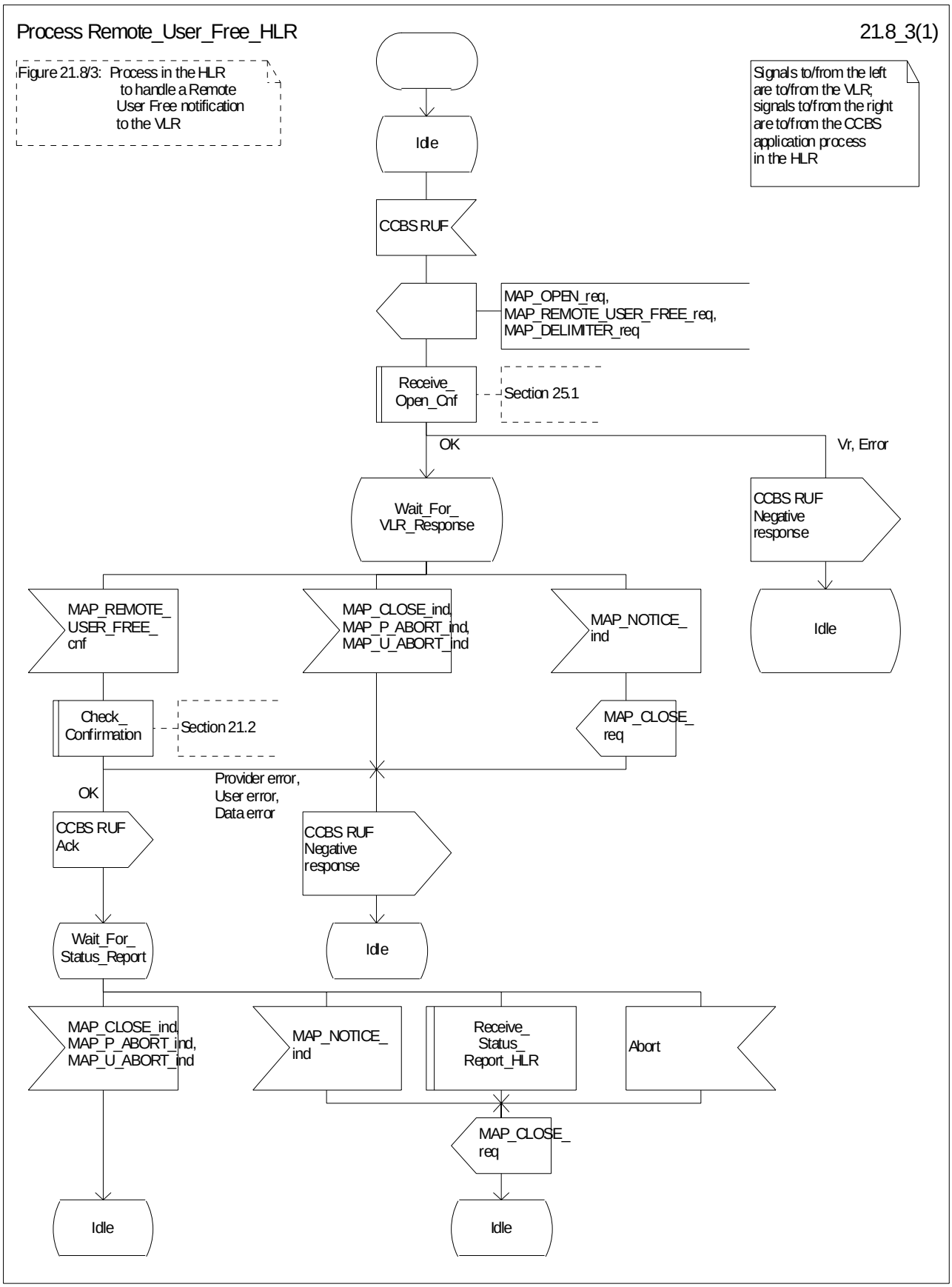


Figure 21.8/3: Process Remote_User_Free_HLR

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 clause; the definitions of this macro can be found as follows:

Check_Confirmation see clause 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.

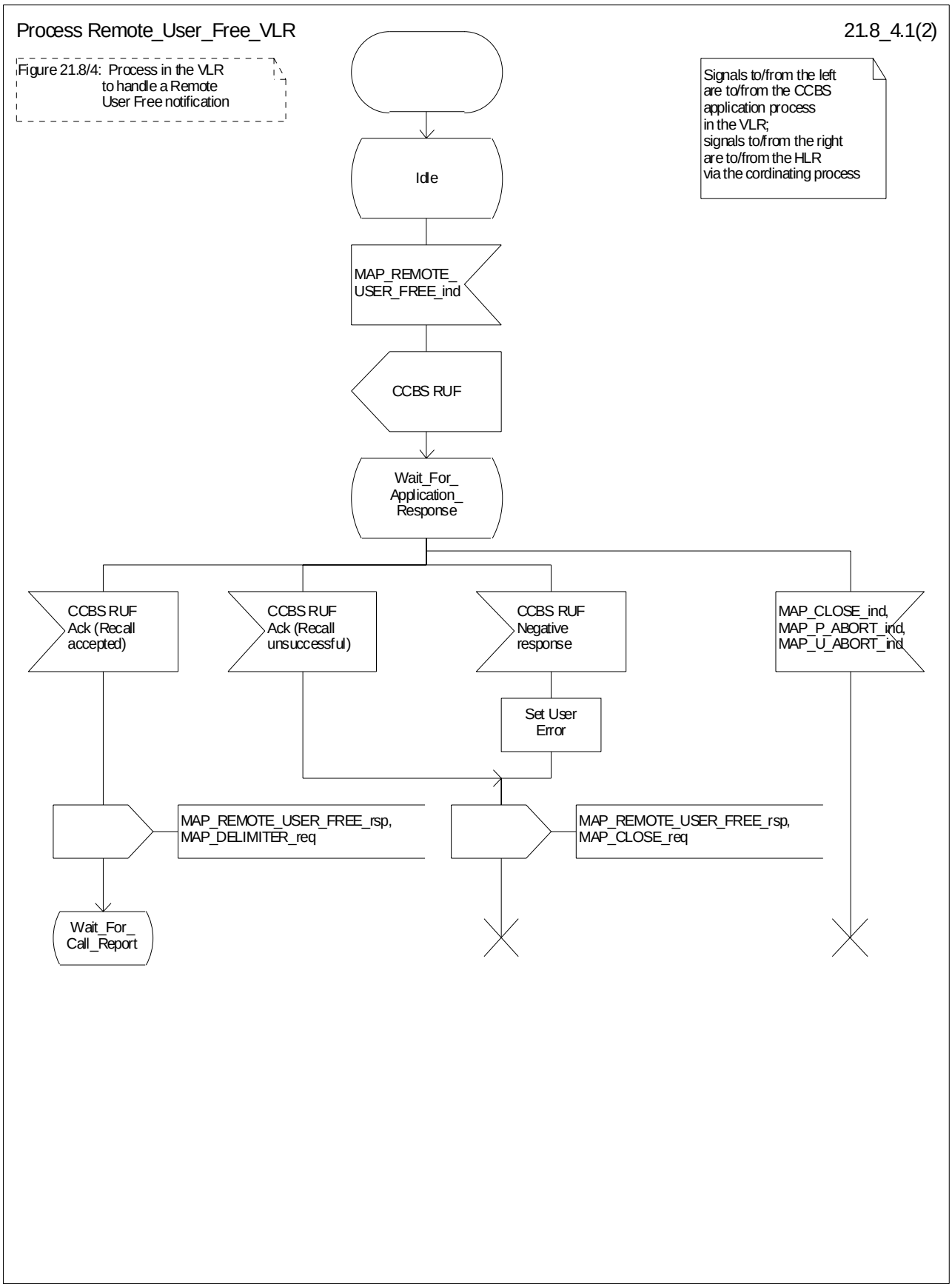


Figure 21.8/4 (sheet 1 of 2): Process Remote_User_Free_VLR

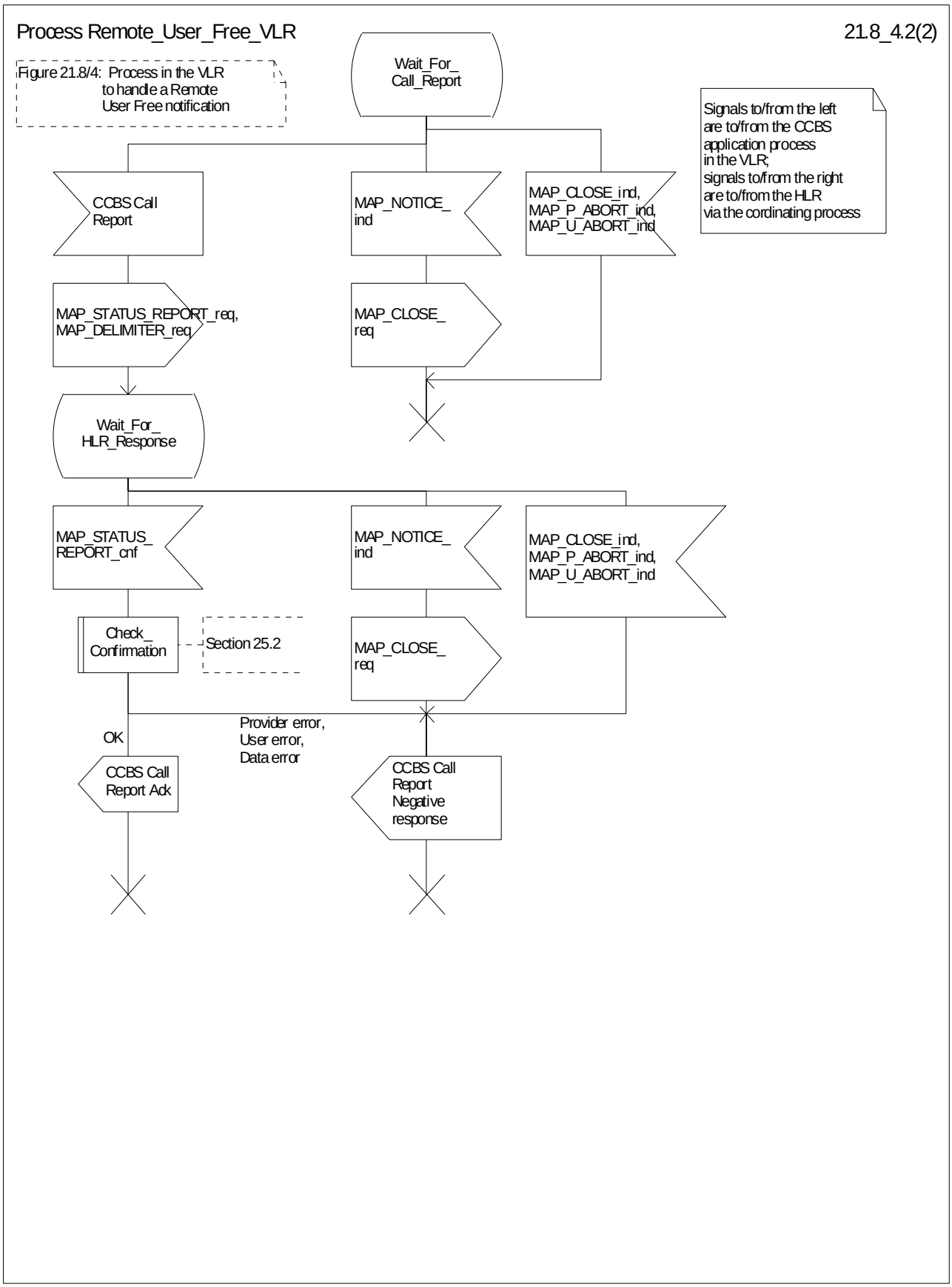


Figure 21.8/4 (sheet 2 of 2): Process Remote_User_Free_VLR

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.

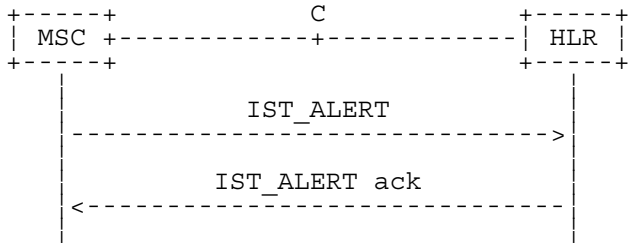


Figure 21.9/1: IST Alert

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 clause; the definitions of these macros can be found as follows:

Receive_Open_Cnf see clause 25.1.2;

Check_Confirmation see clause 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.

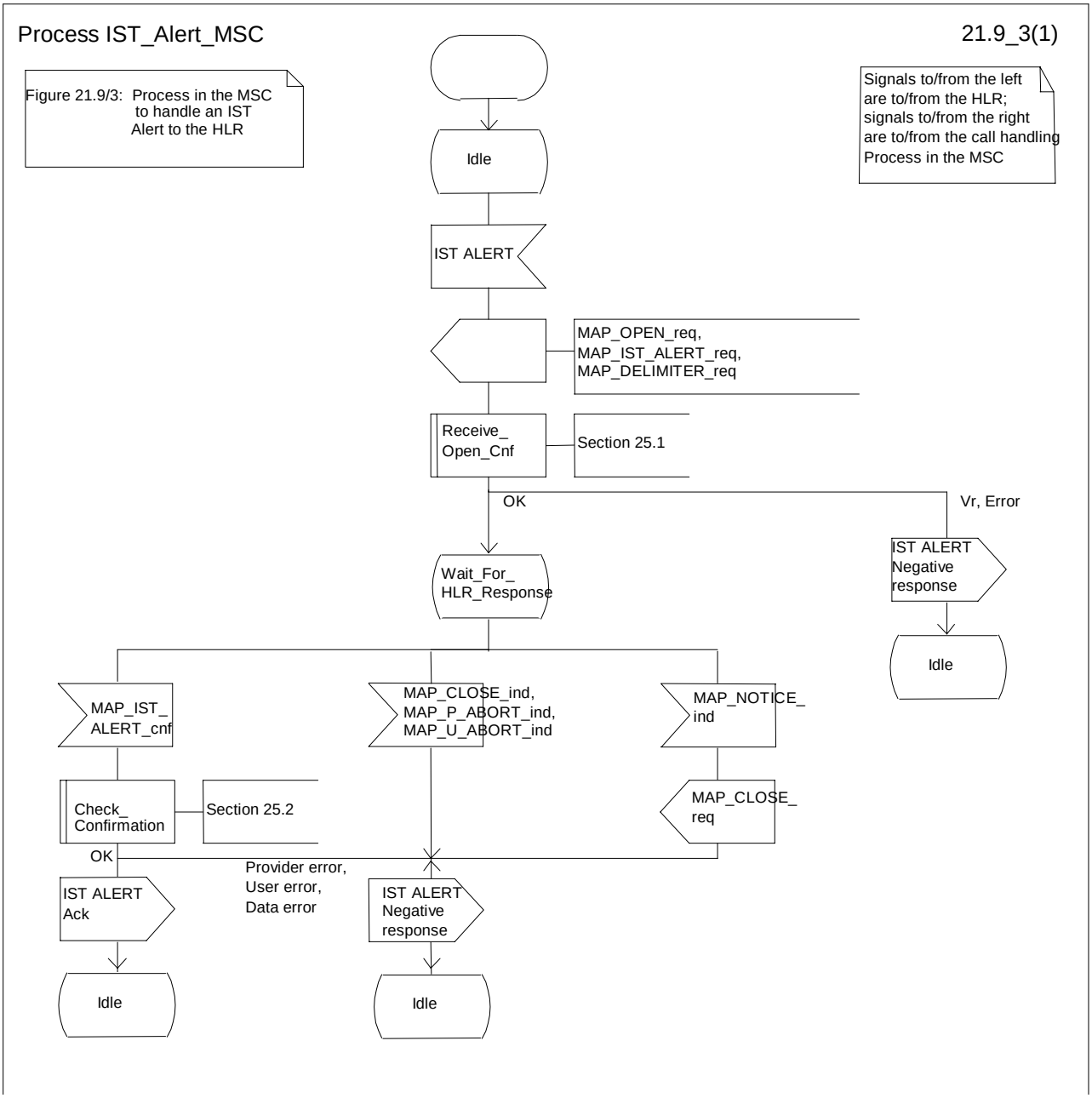


Figure 21.9/2: Process IST_Alert_MSC

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 clause; the definition of this macro can be found as follows:

Receive_Open_Ind see clause 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 Reeive_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.

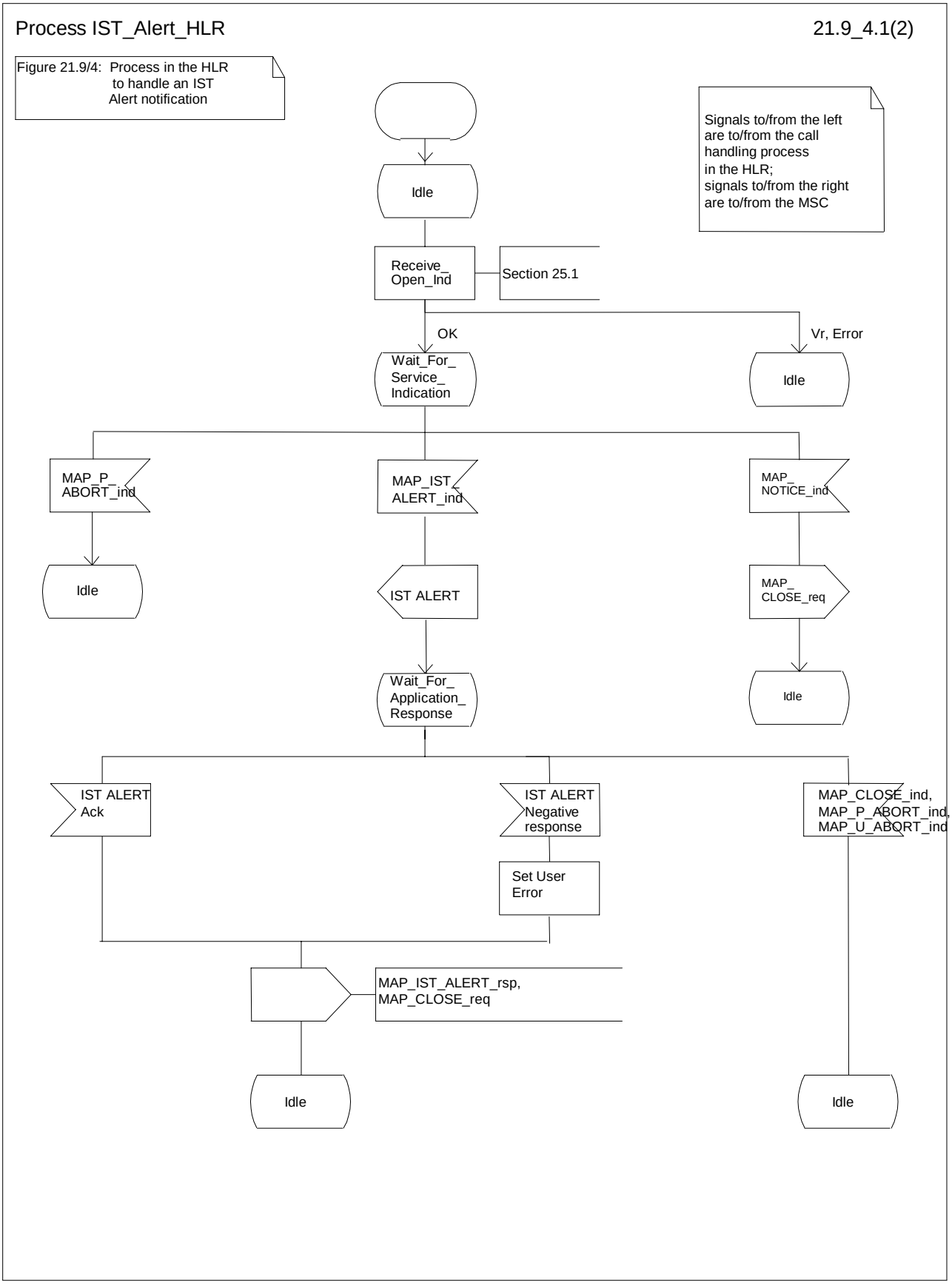


Figure 21.9/3: Process IST_Alert_HLR

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.

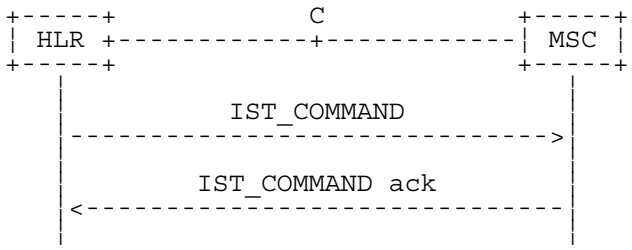


Figure 21.10/1: IST Command

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 clause; the definitions of these macros can be found as follows:

Receive_Open_Cnf see clause 25.1.2;

Check_Confirmation see clause 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.

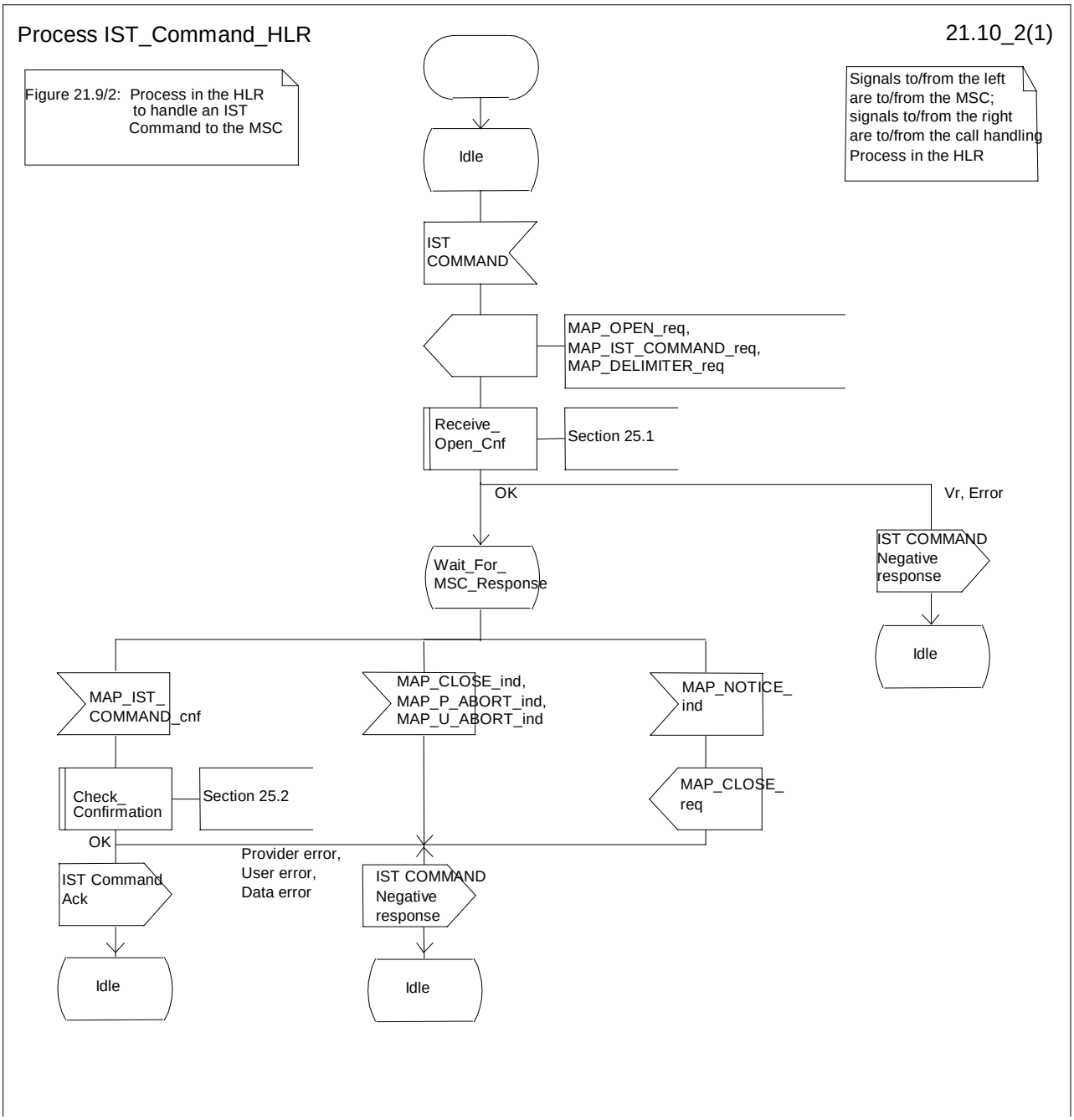


Figure 21.10/2: Process IST_Command_HLR

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 clause; the definition of this macro can be found as follows:

Receive_Open_Ind see clause 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.

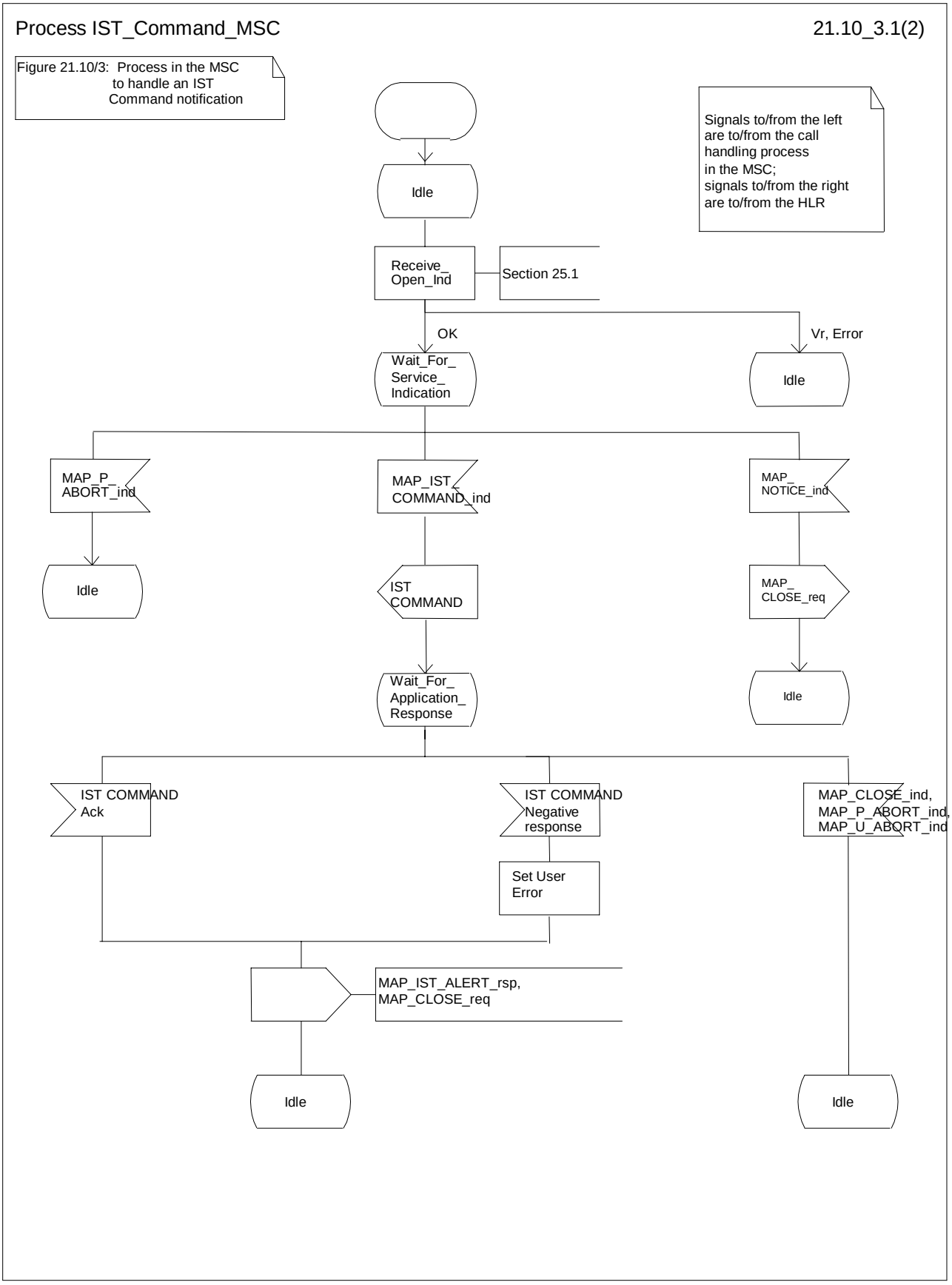


Figure 21.10/3: Process IST_Command_MSC

22 Supplementary services procedures

The following application contexts exist for handling of supplementary services:

- accessUnstructuredSsContext;
- accessFunctionalSsContext.

The accessUnstructuredSsContext refers to a simple MAP user, 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 3GPP TS 29.011 [59].

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.

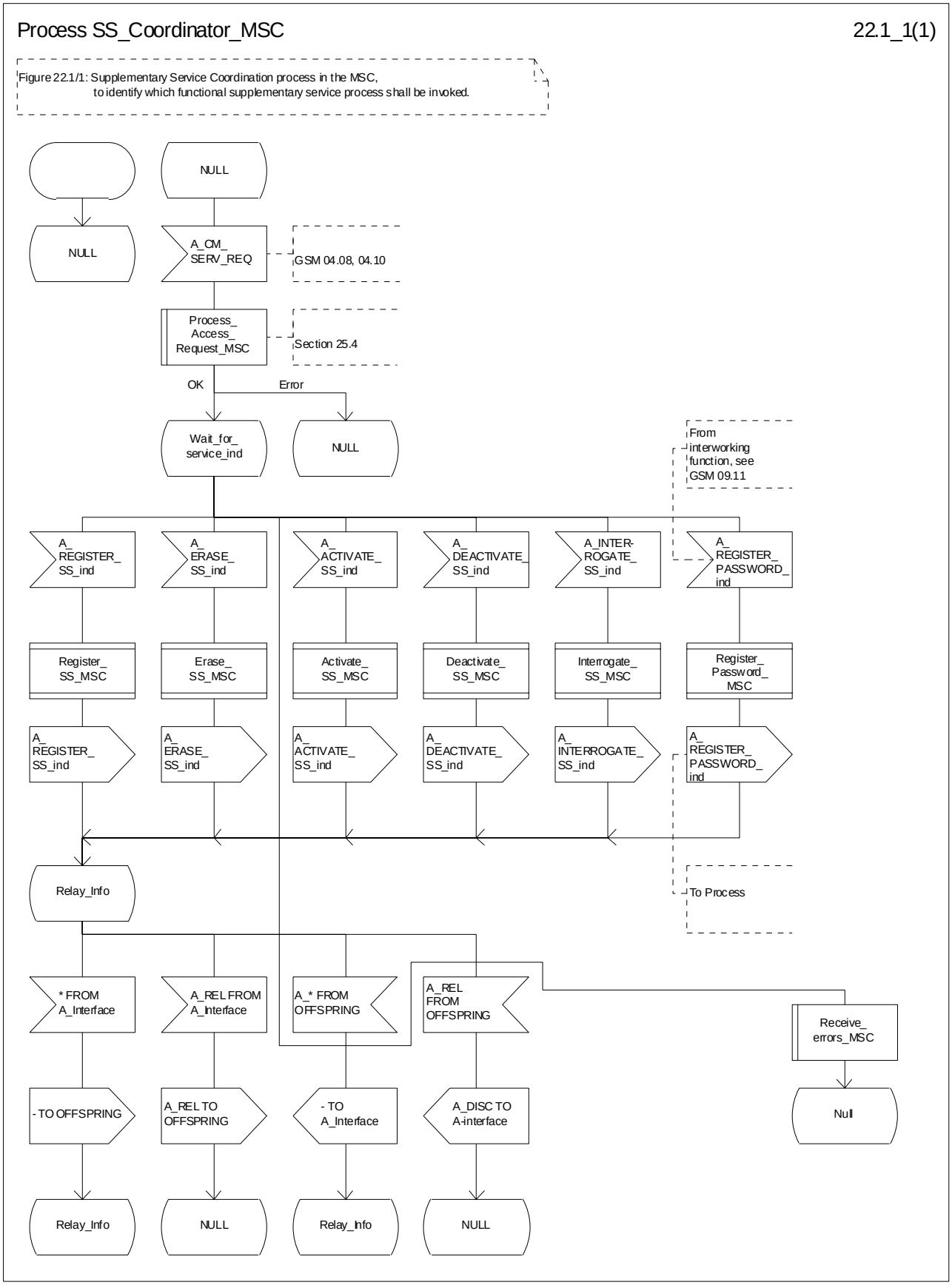


Figure 22.1/1: Process SS_Coordinator_MSC

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.

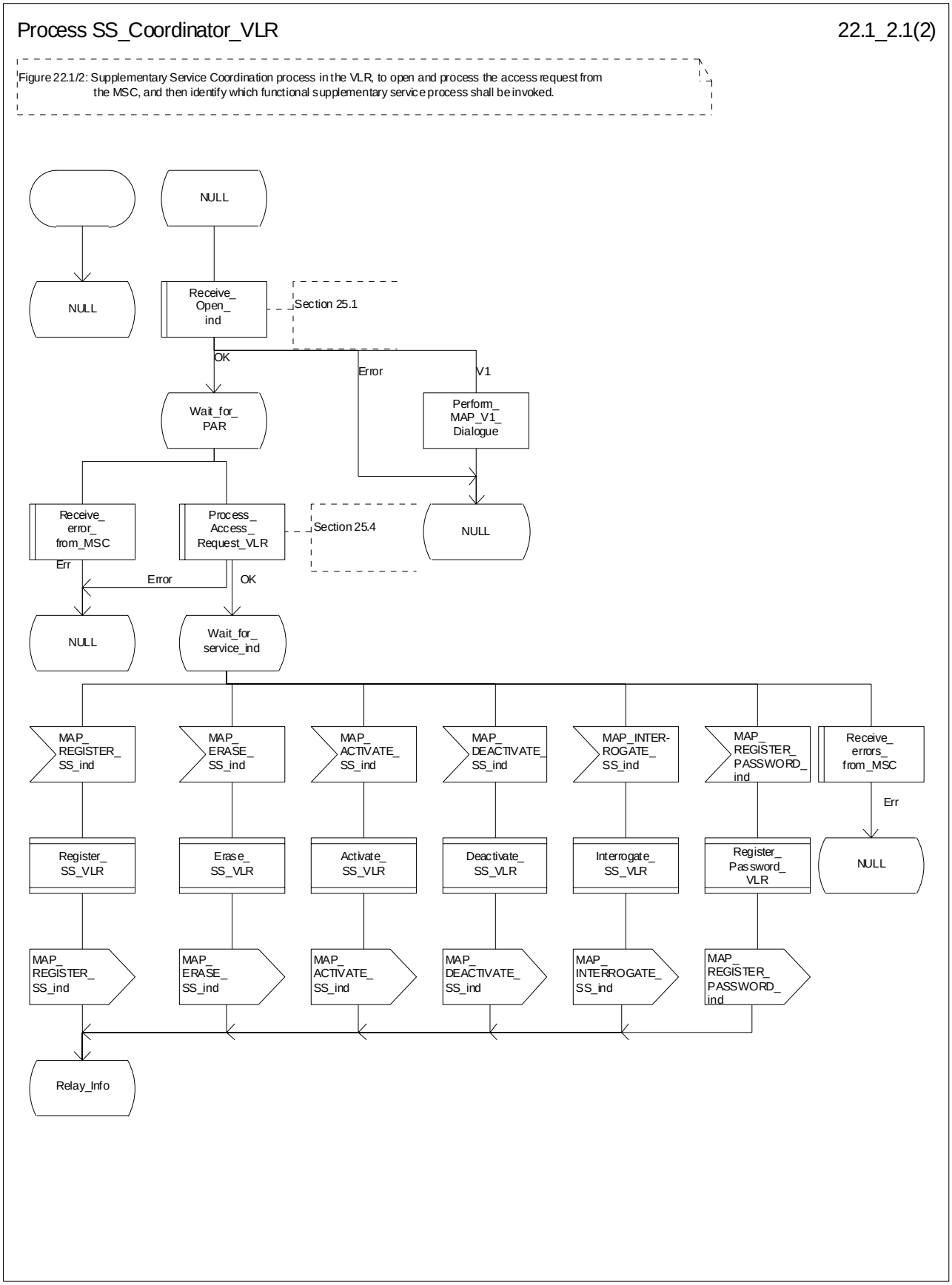


Figure 22.1/2 (sheet 1 of 2): Process SS_Coordinator_VLR

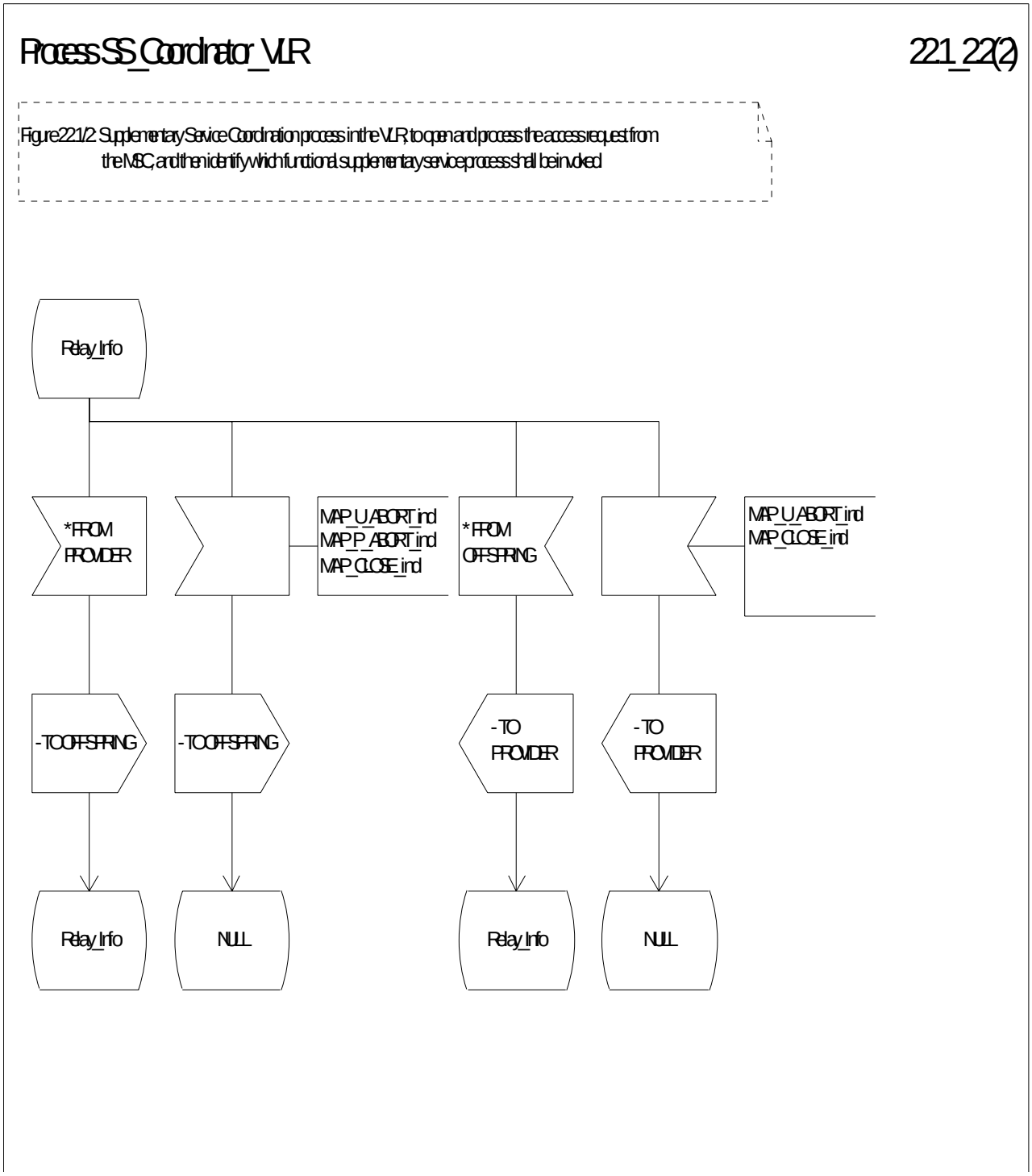


Figure 22.1/2 (sheet 2 of 2): Process SS_Coordinator_VLR

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.

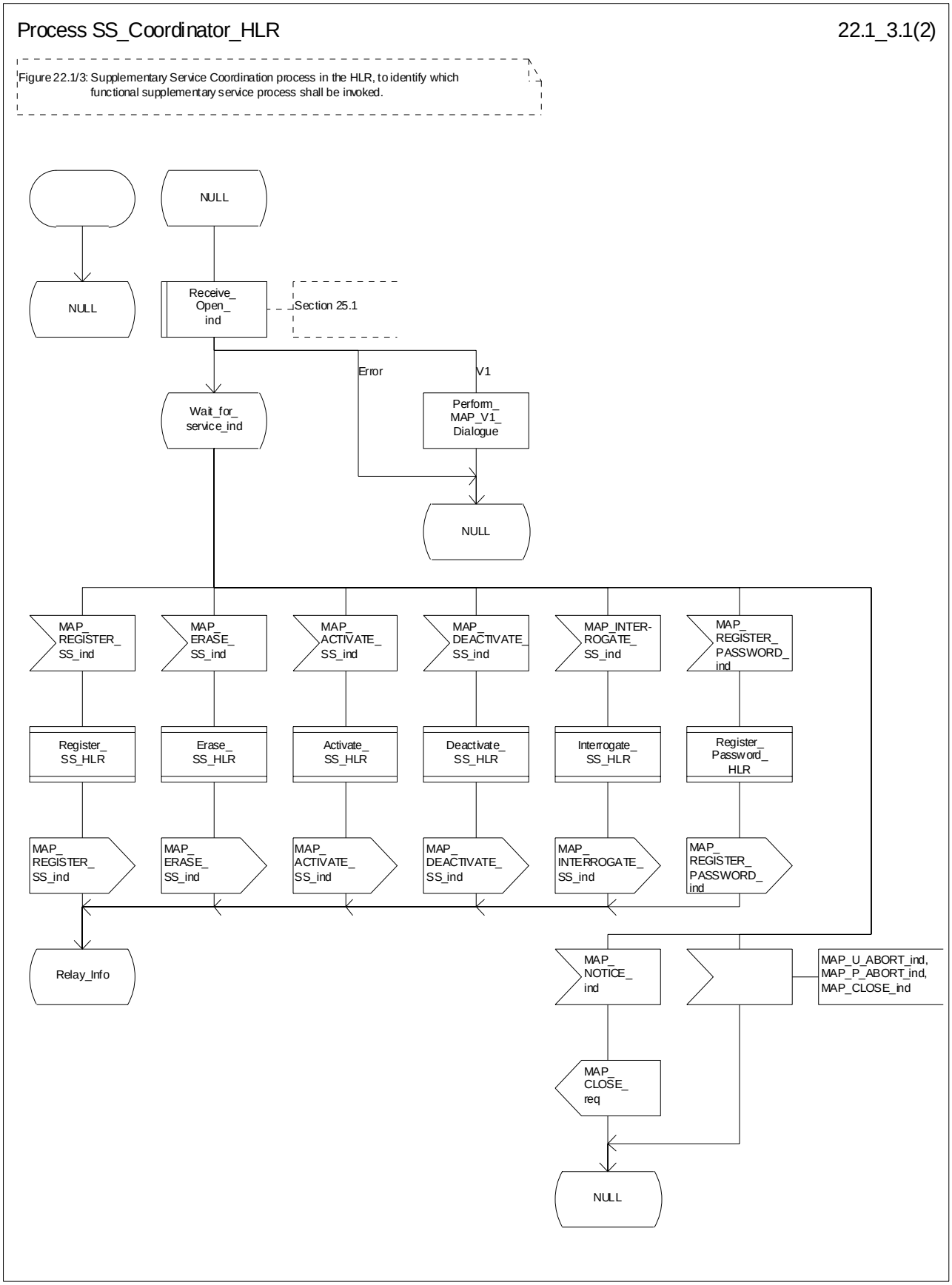


Figure 22.1/3 (sheet 1 of 2): Process SS_Coordinator_HLR

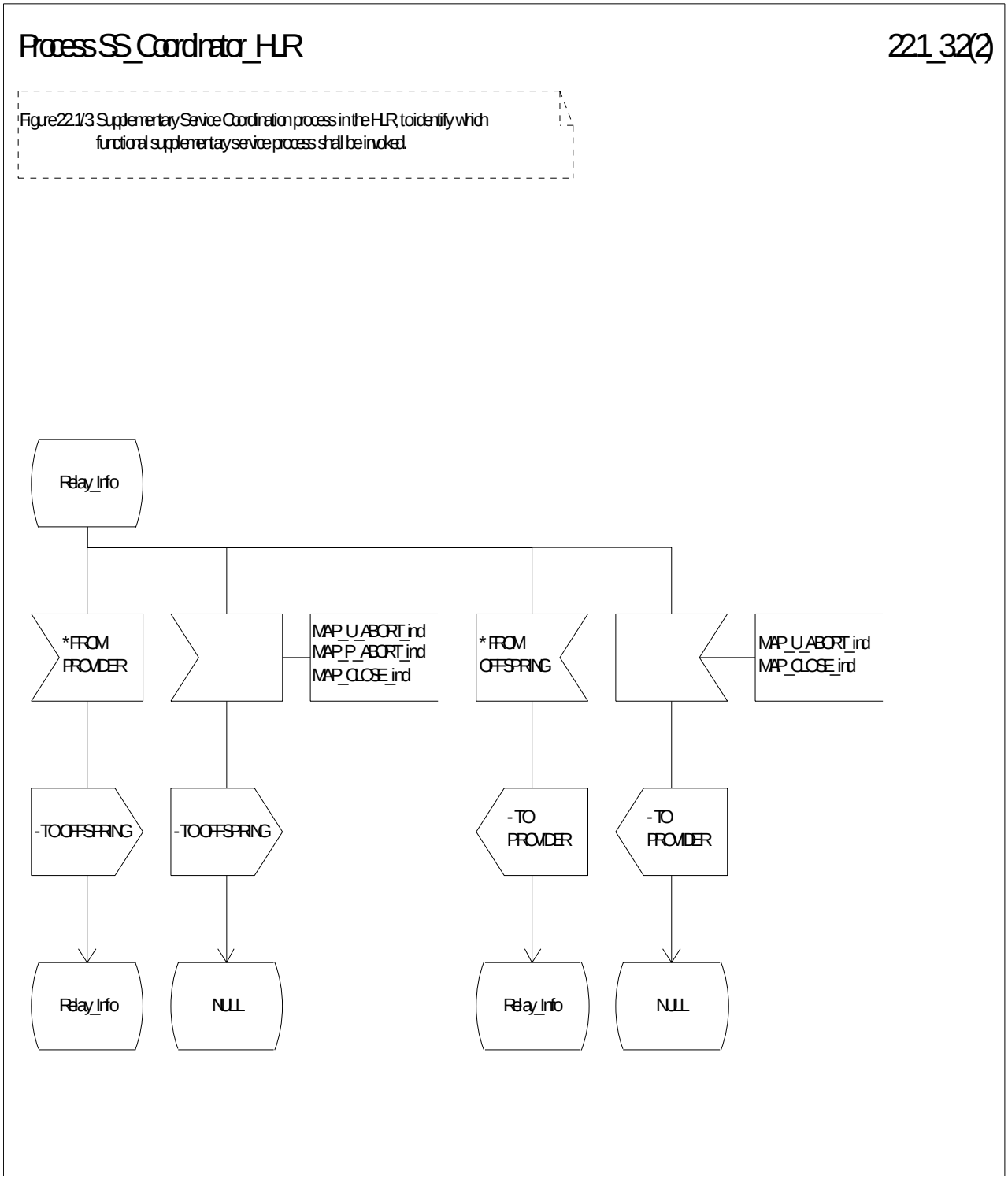


Figure 22.1/3 (sheet 2 of 2): Process SS_Coordinator_HLR

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 clause; the definition of this macro can be found as follows:

Receive_Open_Ind see clause 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.

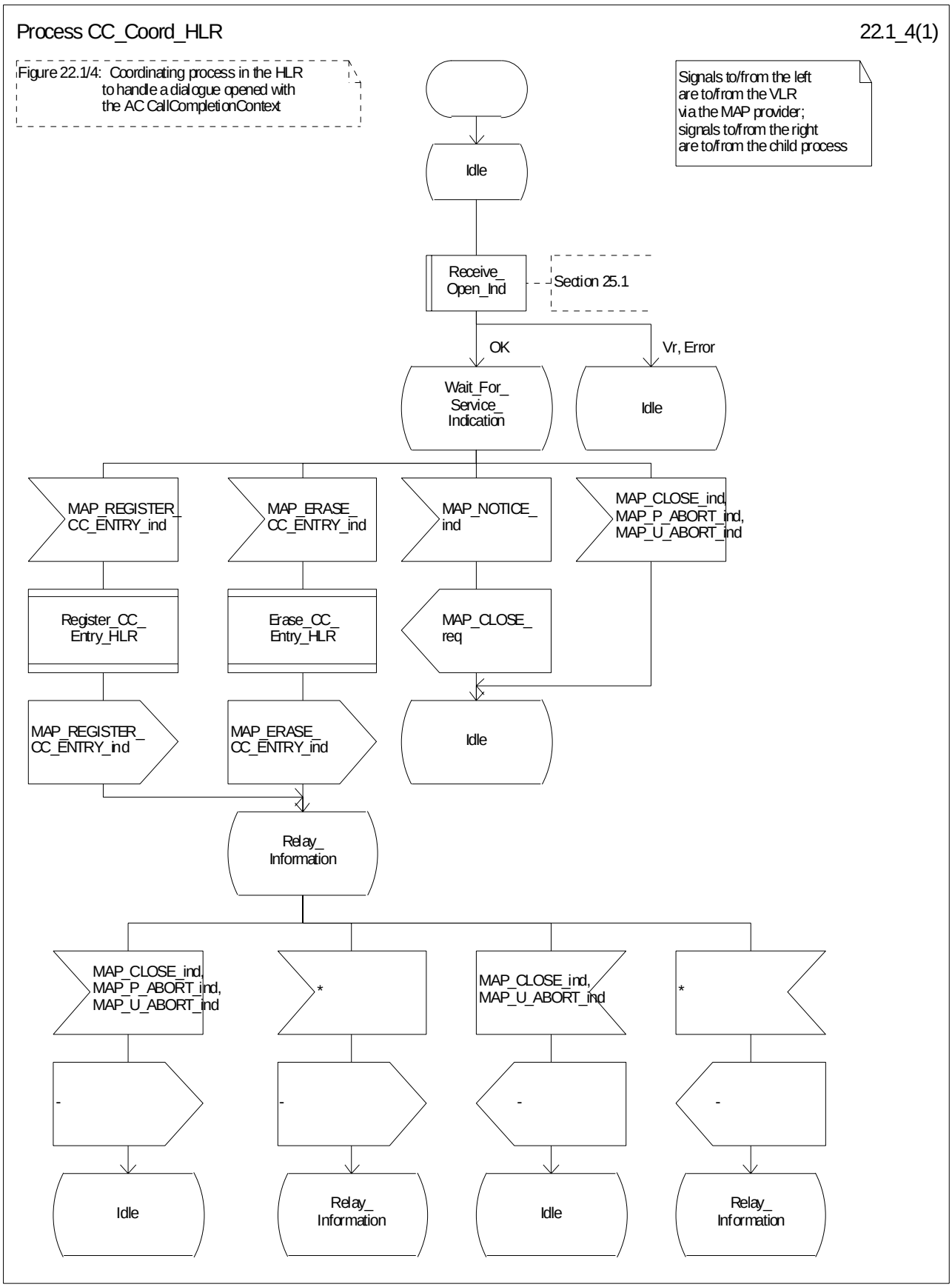


Figure 22.1/4: Process_CC_Coord_HLR

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 clauses 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).

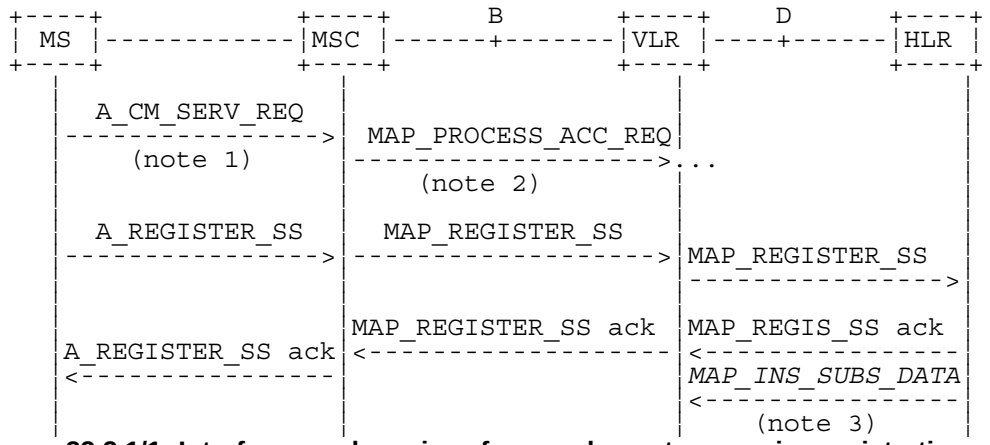


Figure 22.2.1/1: Interfaces and services for supplementary service registration

NOTE 1: For details of the procedure on the radio path, see 3GPP TS 24.008, 04.10, 3GPP TS 24.08x and 3GPP TS 24.09x. 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.

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 3GPP TS 29.011 [59].

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 3GPP TS 24.08x, 3GPP TS 24.09x and 3GPP TS 29.011. 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 3GPP TS 29.011 [59]).

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 3GPP TS 29.011 [59]). 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.

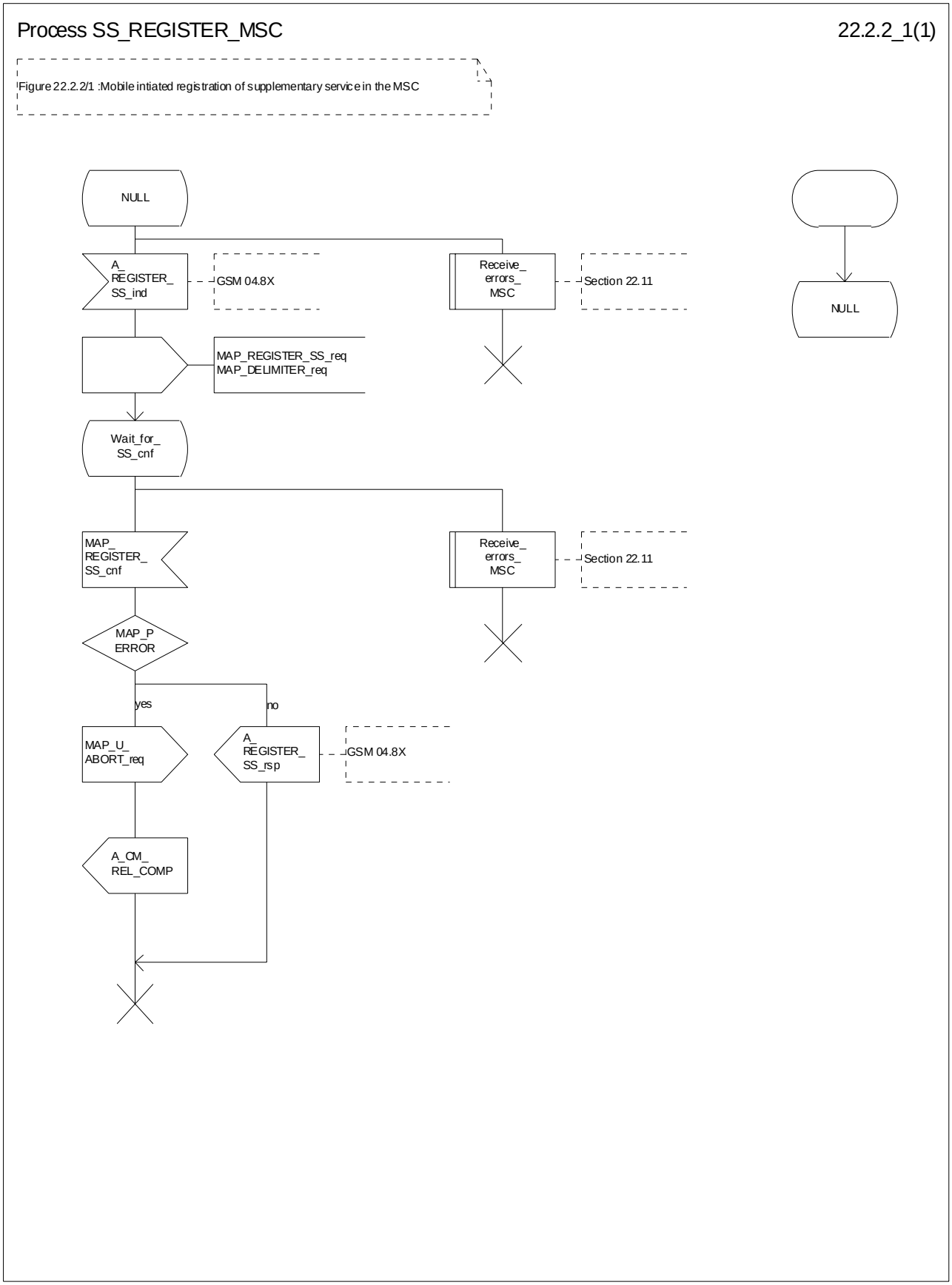


Figure 22.2.2/1: Procedure SS_Register_MSC

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 3GPP TS 29.011 [59]).

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.

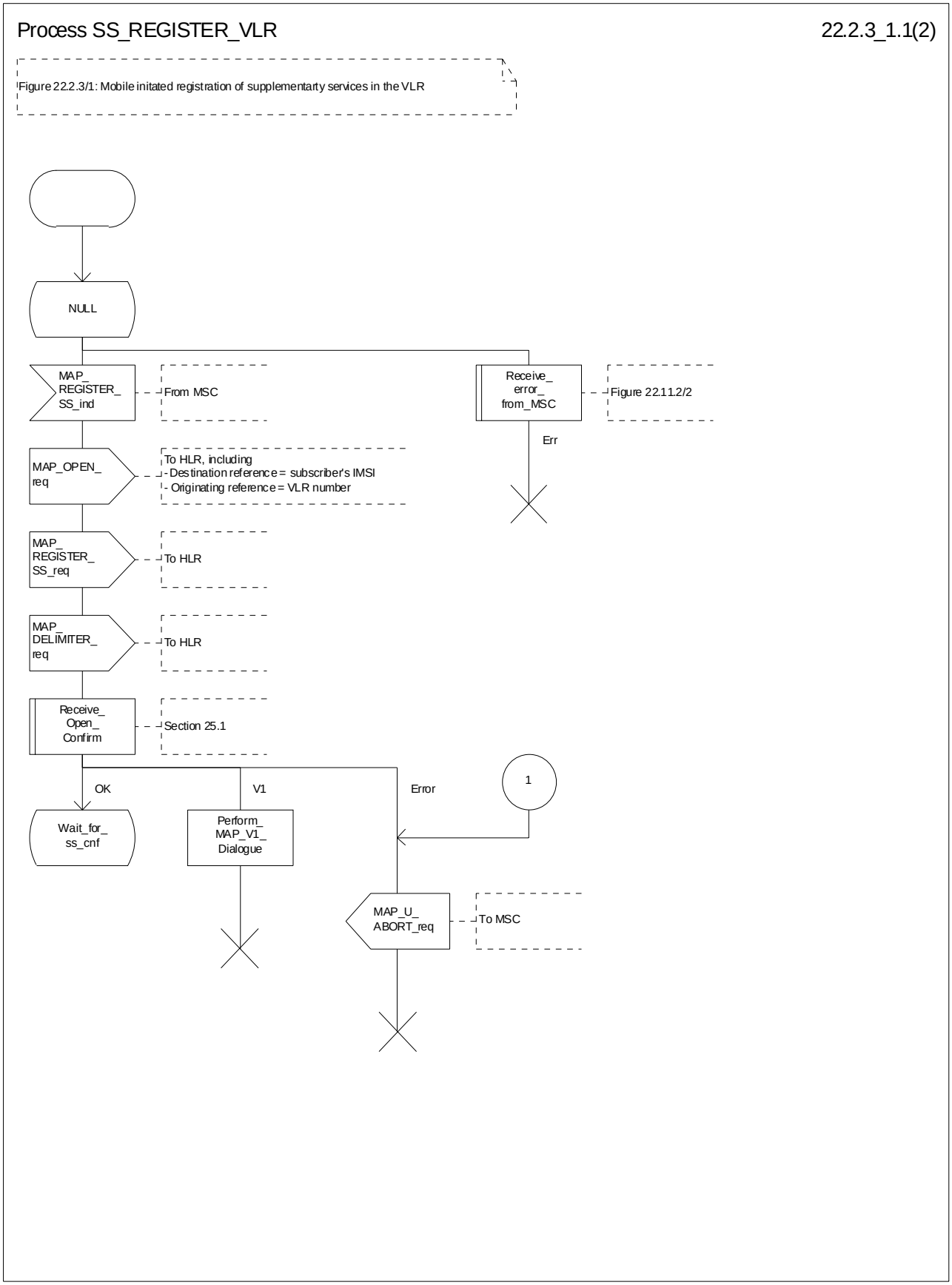


Figure 22.2.3/1 (sheet 1 of 2): Procedure SS_Register_VLR

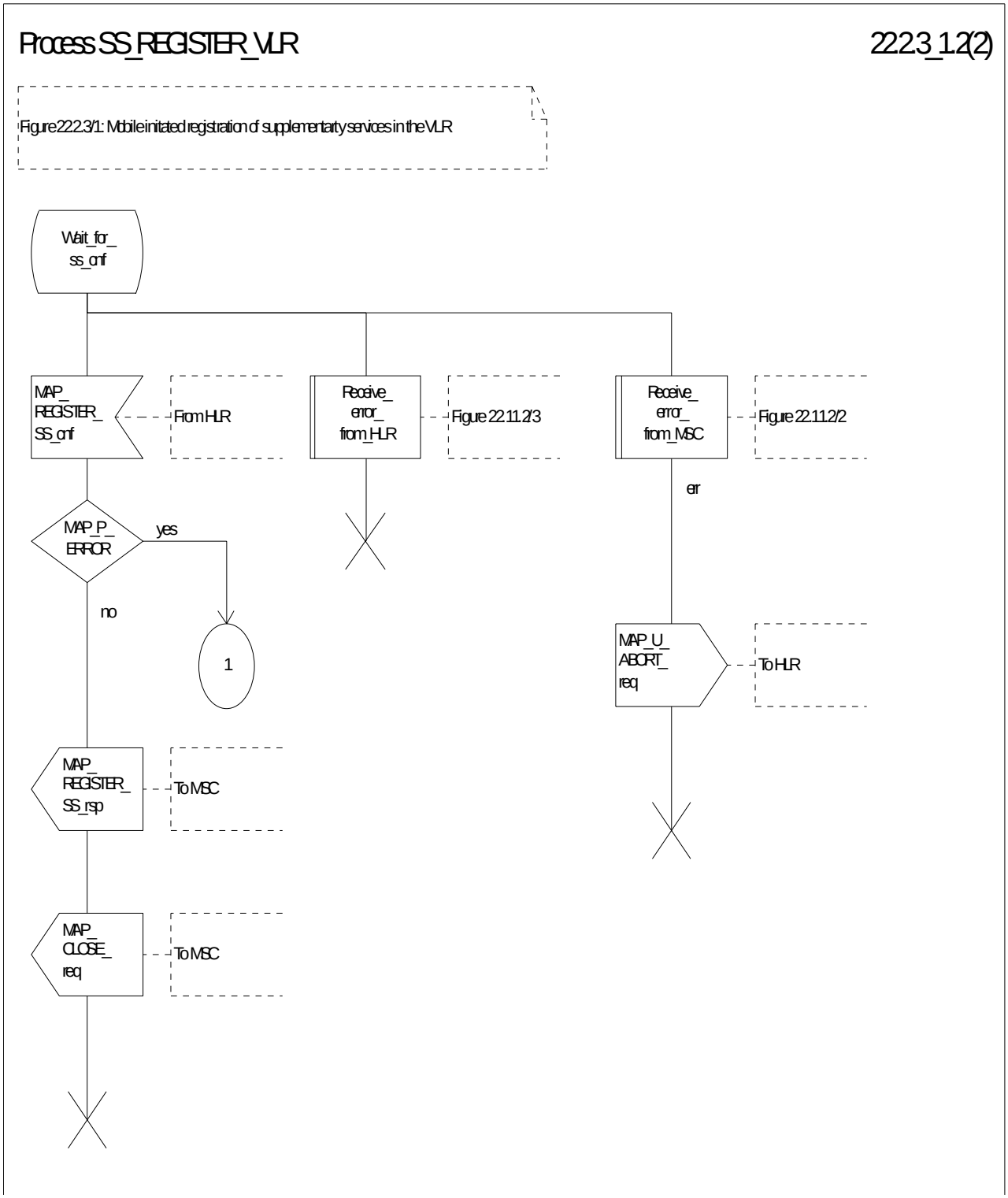


Figure 22.2.3/1 (sheet 2 of 2): Procedure SS_Register_VLR

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 3GPP TS 23.011 [22] and the 3GPP TS 23.08x and 3GPP TS 23.09x-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 3GPP TS 29.011 [59]):

- 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 is sent towards the VLR.

The registration procedure in the HLR is shown in figure 22.2.4/1.

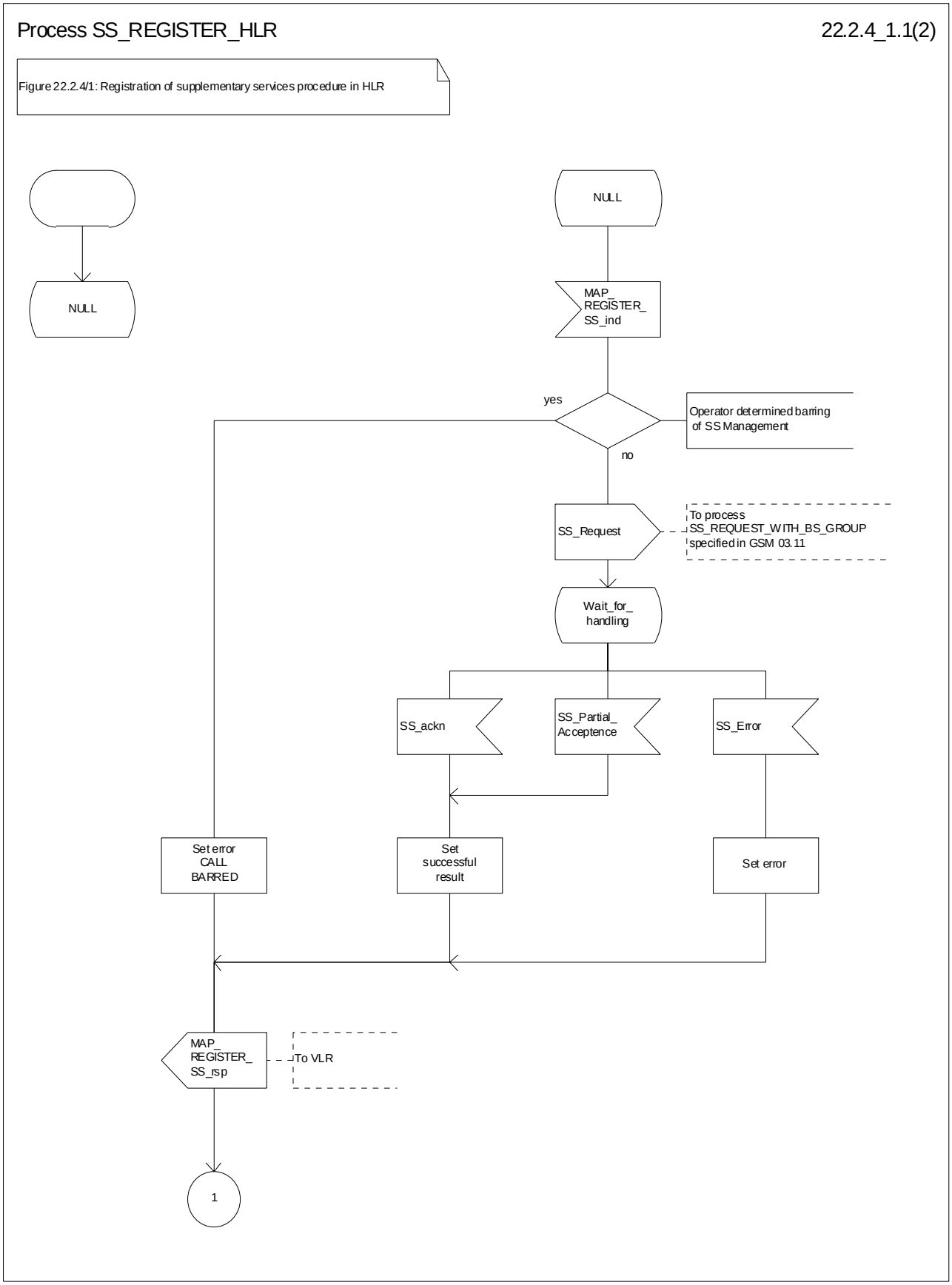


Figure 22.2.4/1 (sheet 1 of 2): Procedure SS_Register_HLR

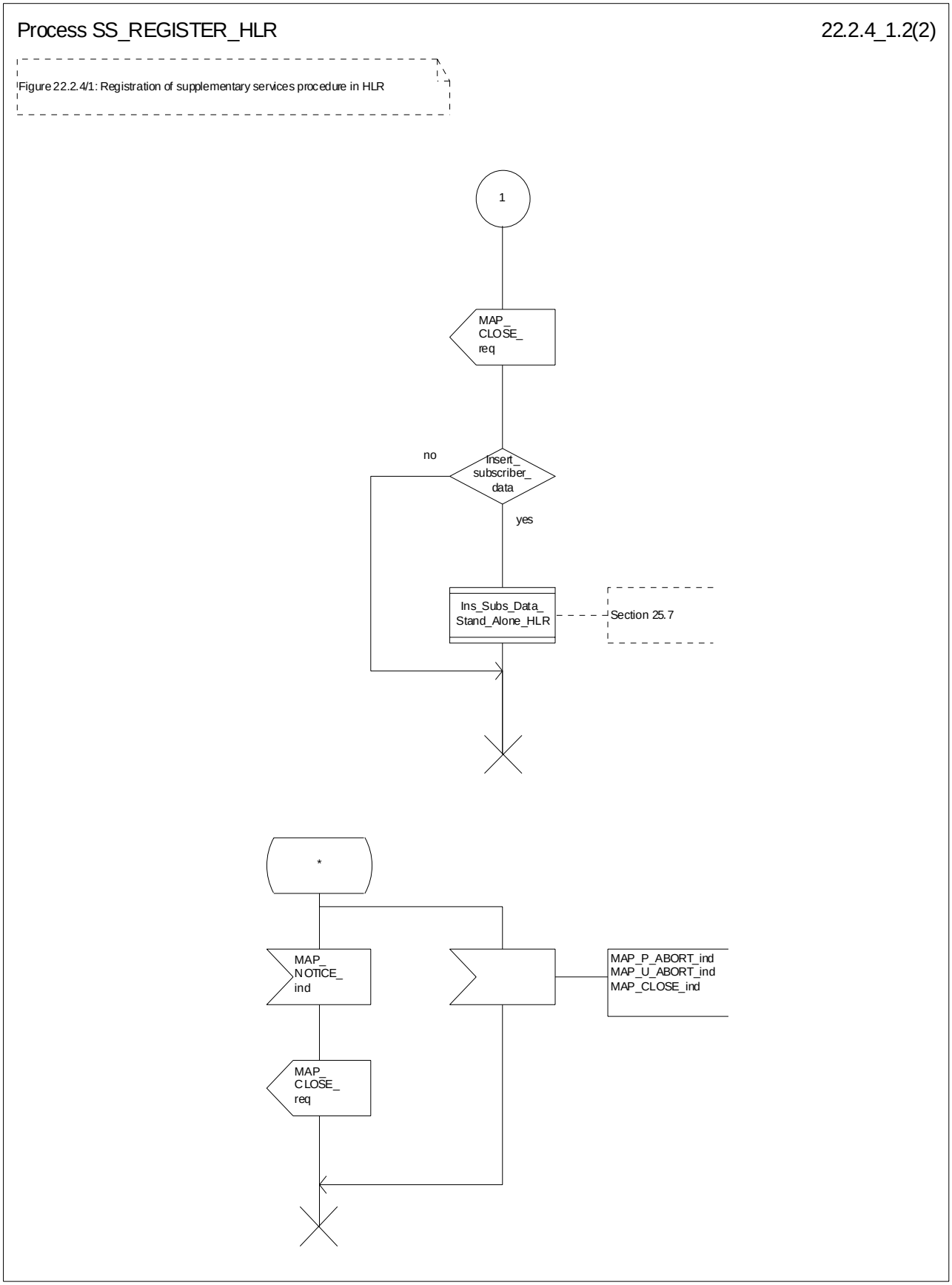


Figure 22.2.4/1 (sheet 2 of 2): Procedure SS_Register_HLR

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 clauses below.

The erasure procedure is shown in figure 22.3.1/1.

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_ERASE_SS (defined in clause 11).

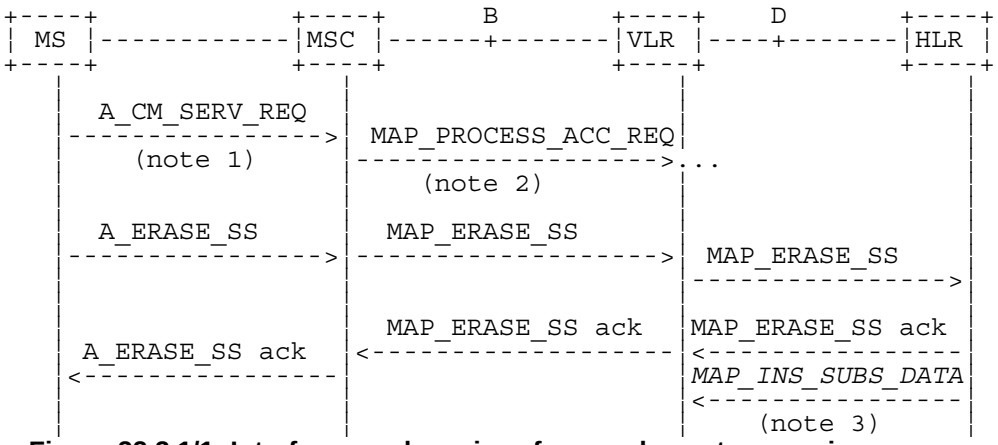


Figure 22.3.1/1: Interfaces and services for supplementary service erasure

NOTE 1: For details of the procedure on the radio path, see 3GPP TS 24.008, 04.10, 3GPP TS 24.08x and 3GPP TS 24.09x. 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.

22.3.2 Procedures in the MSC

The MSC procedures for erasure are identical to those specified for registration in clause 22.2.2. The text and diagrams in clause 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 clause 22.2.3. The text and diagrams in clause 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 clause 22.2.4. The text and diagrams in clause 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 clauses 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).

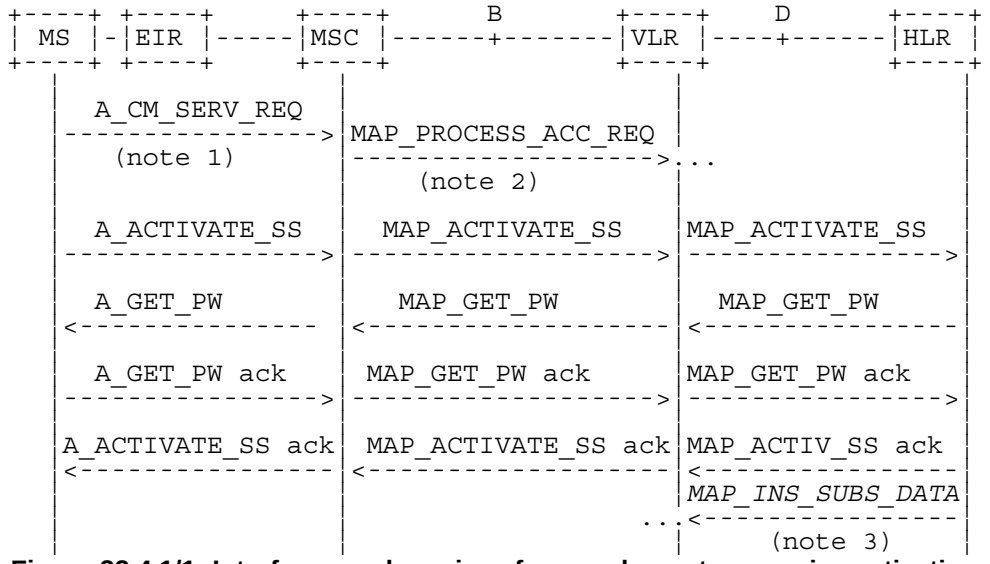


Figure 22.4.1/1: Interfaces and services for supplementary service activation

NOTE 1: For details of the procedure on the radio path, see 3GPP TS 24.008, 04.10, 3GPP TS 24.08x and 3GPP TS 24.09x. 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.

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 3GPP TS 29.011 [59].

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 3GPP TS 29.011 [59].

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 3GPP TS 24.08x, 3GPP TS 24.09x and 3GPP TS 29.011. 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 3GPP TS 29.011 [59]).

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 clause 22.2.2 of the present document.

The activation procedure in the MSC is shown in figure 22.4.2/1.

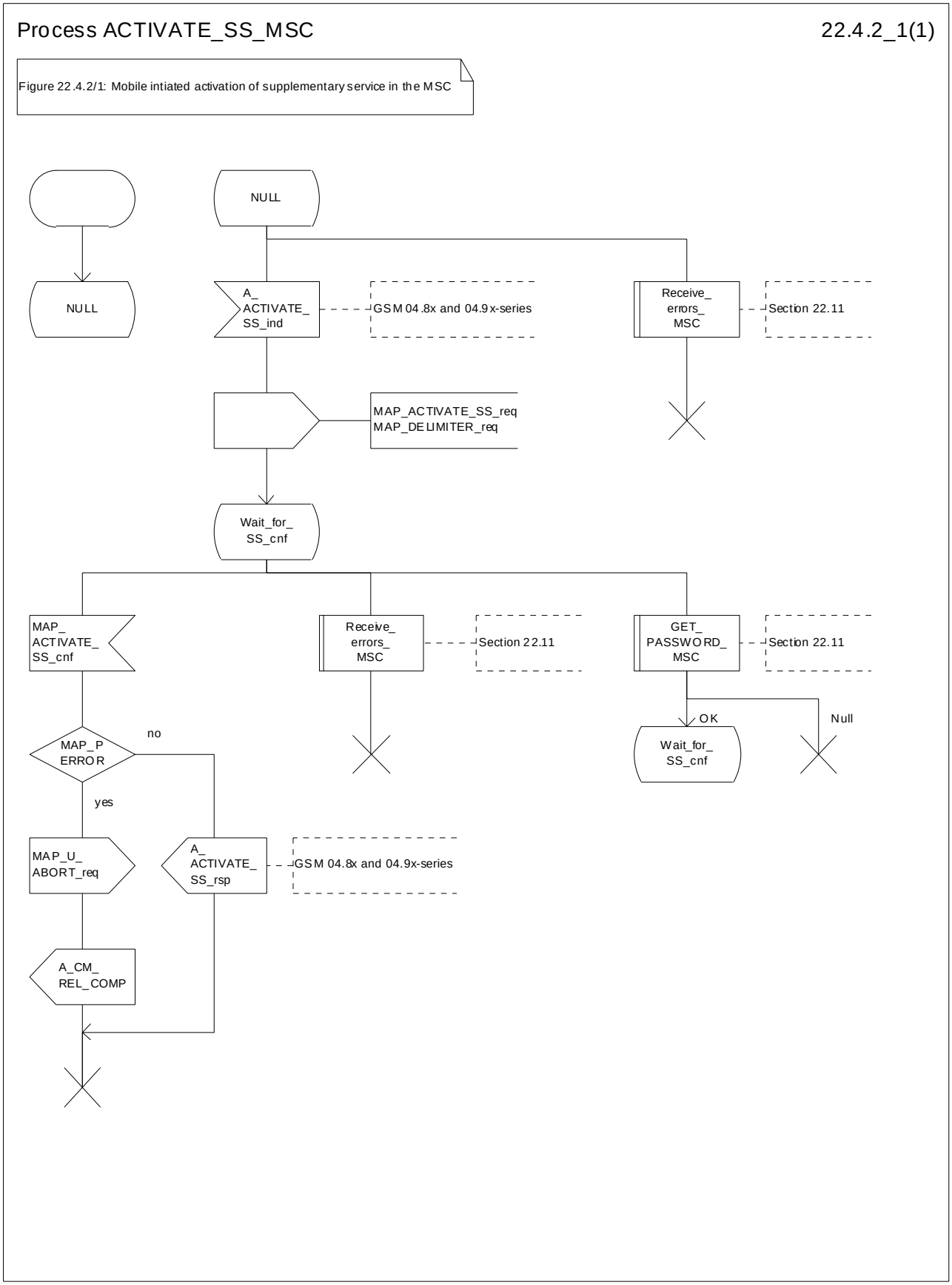


Figure 22.4.2/1: Procedure Activate_SS_MSC

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 3GPP TS 29.011 [59]).

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 clause 22.2.3 of the present document.

The activation procedure in the VLR is shown in figure 22.4.3/1.

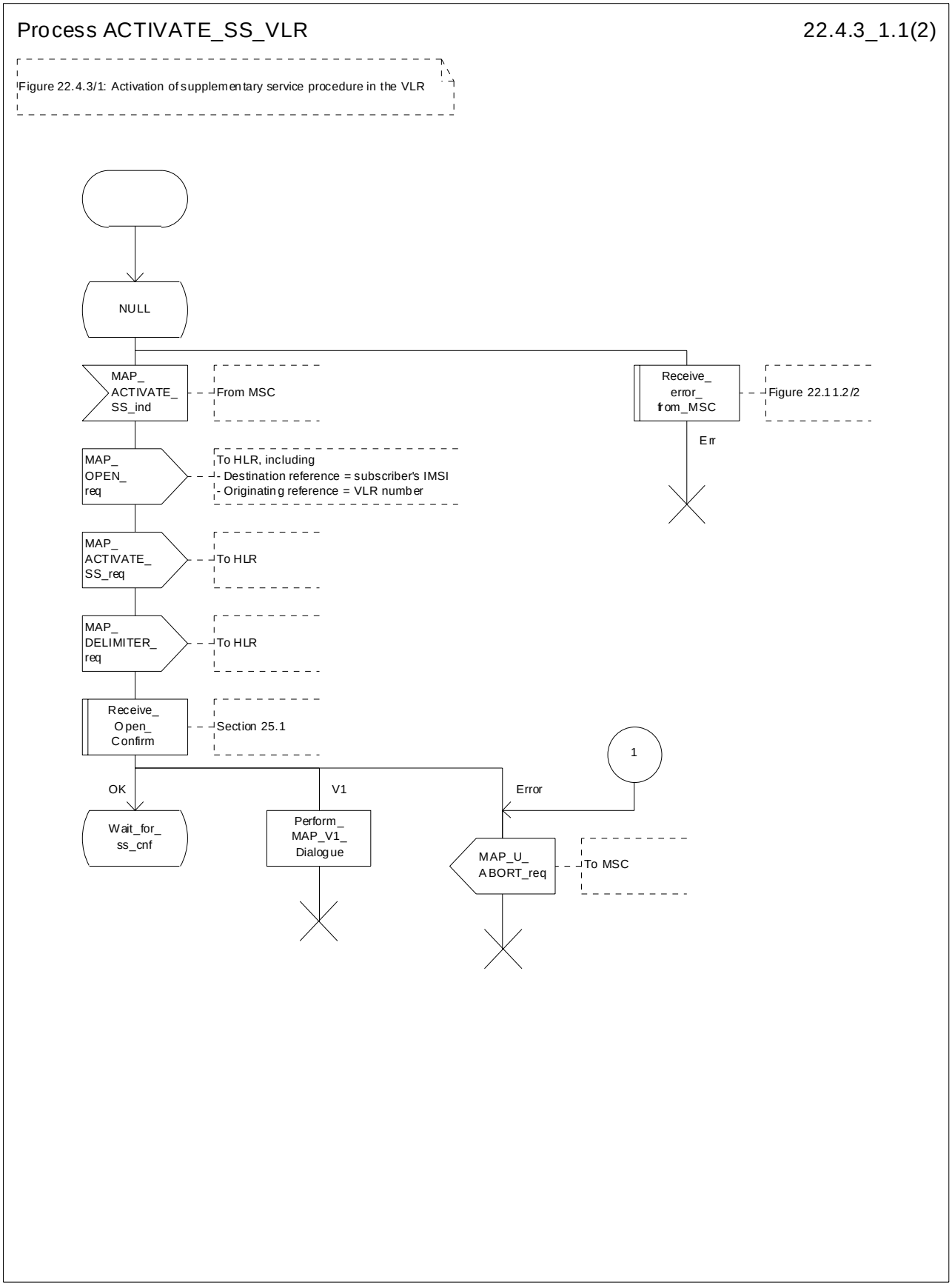


Figure 22.4.3/1 (sheet 1 of 2): Procedure Activate_SS_VLR

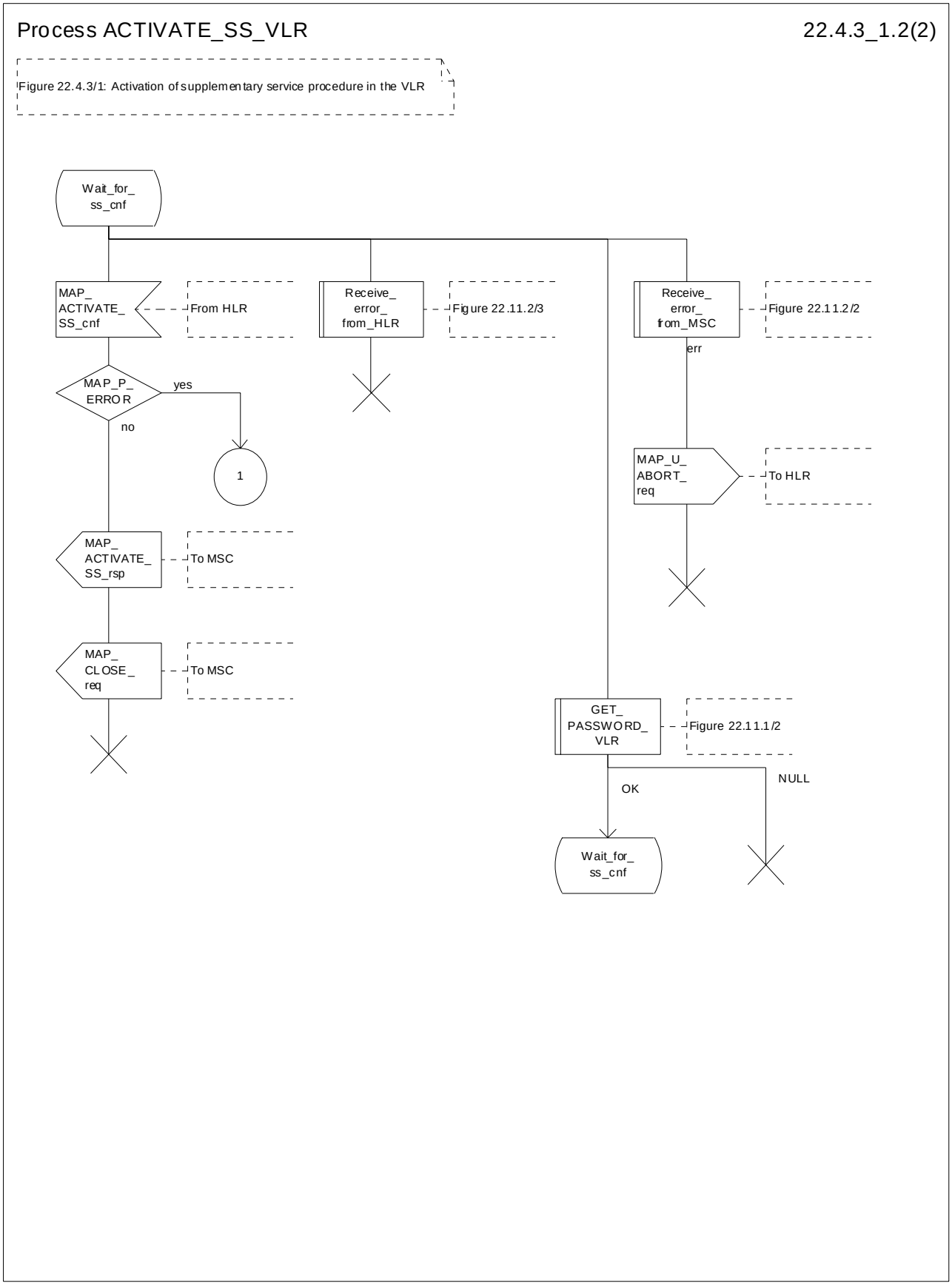


Figure 22.4.3/1 (sheet 2 of 2): Procedure SS_Activate_VLR

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 3GPP TS 23.011 [22] and the 3GPP TS 23.08x and 3GPP TS 23.09x-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 3GPP TS 23.011 [22]). 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 3GPP TS 29.011 [59]):

- 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 clause 22.2.4 above.

The activation procedure in the HLR is shown in figure 22.4.4/1.

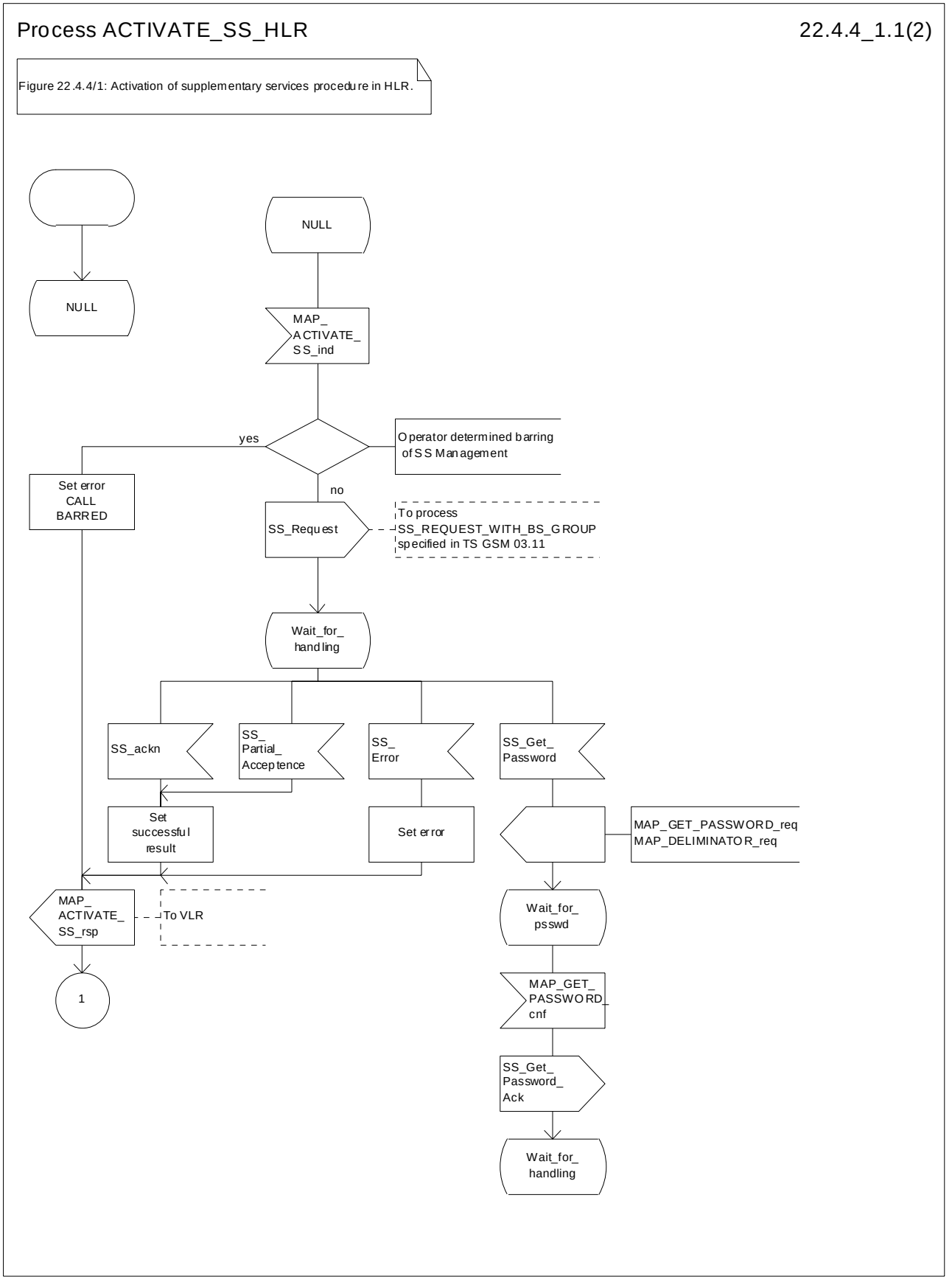


Figure 22.4.4/1 (sheet 1 of 2): Procedure Activate_SS_HLR

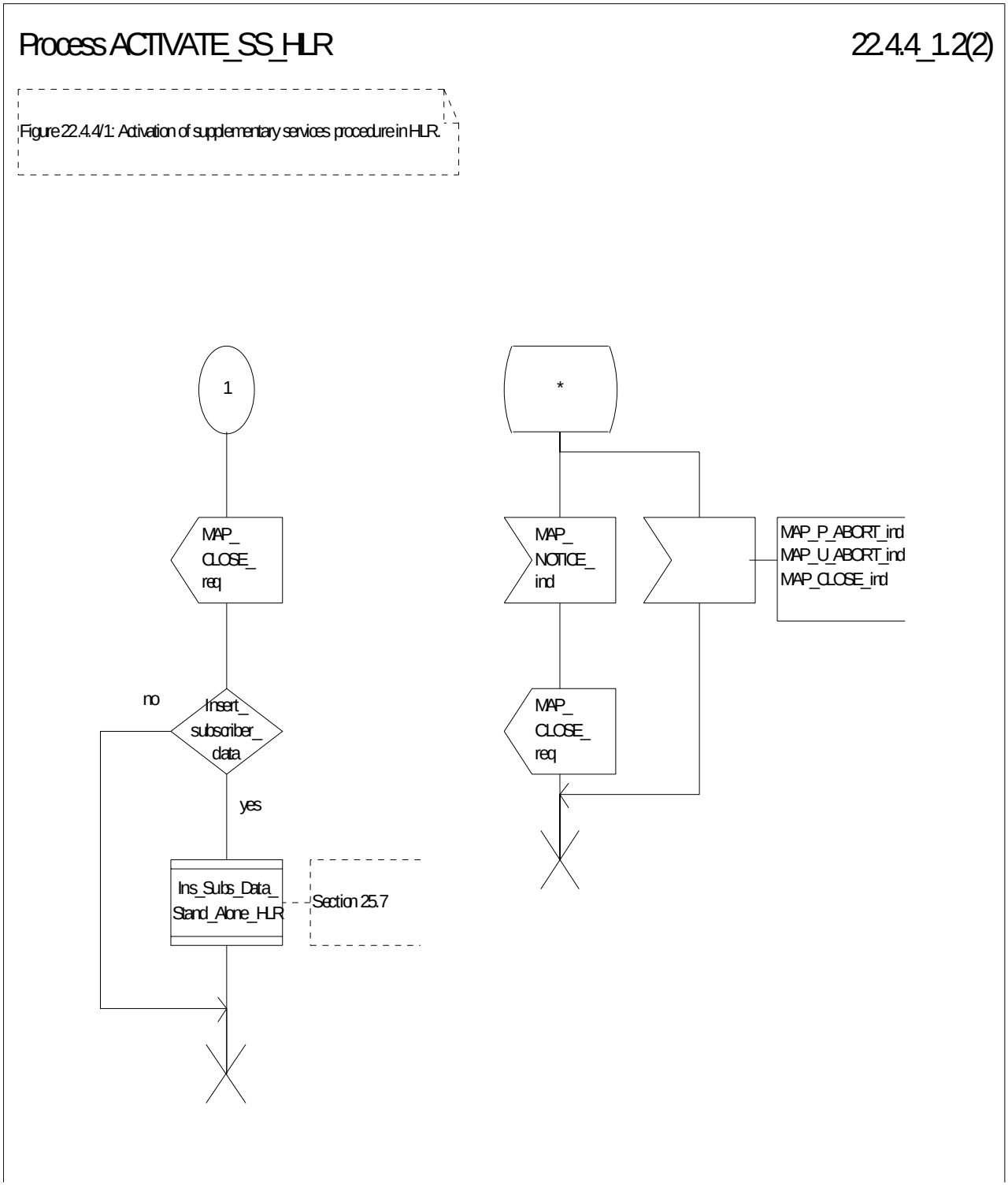


Figure 22.4.4/1 (sheet 2 of 2): Procedure Activate_SS_HLR

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 clauses 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).

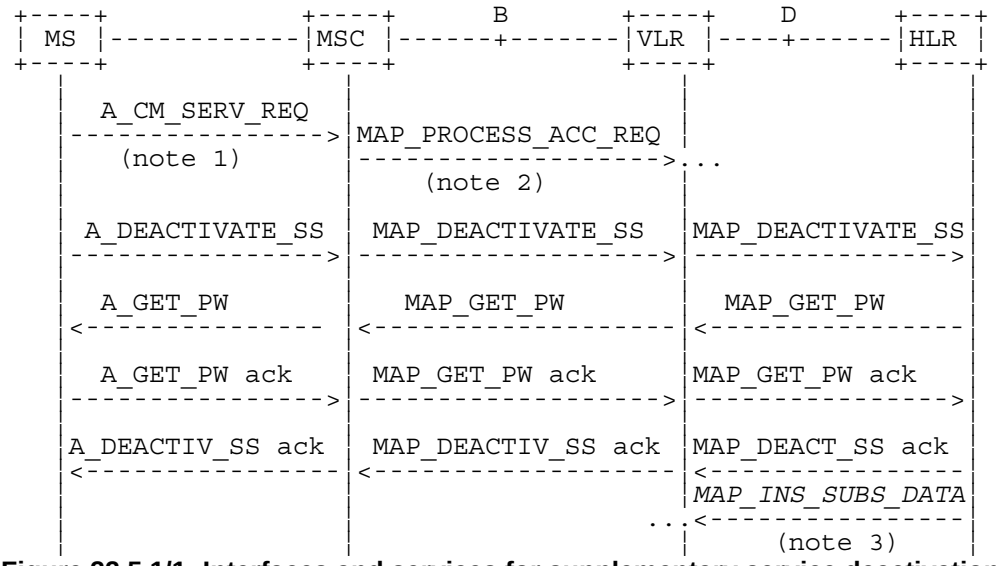


Figure 22.5.1/1: Interfaces and services for supplementary service deactivation

NOTE 1: For details of the procedure on the radio path, see 3GPP TS 24.008, 04.10, 3GPP TS 24.08x and 3GPP TS 24.09x. 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.

22.5.2 Procedures in the MSC

The MSC procedures for deactivation are identical to those specified for activation in clause 22.4.2. The text and diagrams in clause 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 clause 22.4.3. The text and diagrams in clause 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 clause 22.4.4. The text and diagrams in clause 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 clauses 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).

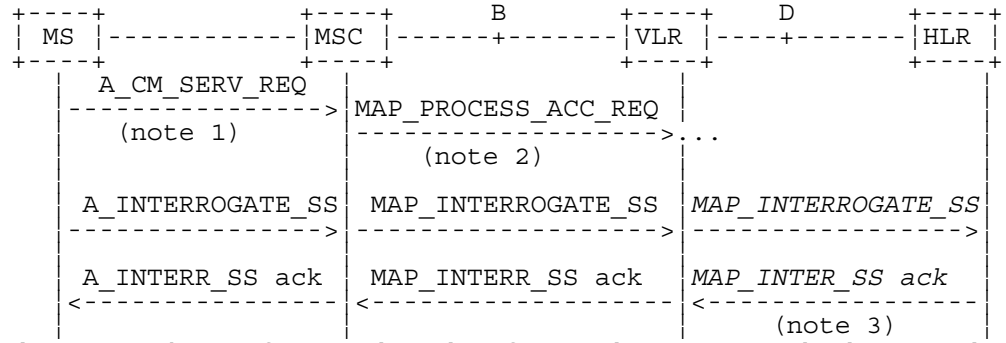


Figure 22.6.1/1: Interfaces and services for supplementary service interrogation

NOTE 1: For details of the procedure on the radio path, see 3GPP TS 24.008, 04.10, 3GPP TS 24.08x and 3GPP TS 24.09x. 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.

22.6.2 Procedures in the MSC

The MSC procedures for interrogation are identical to those specified for registration in clause 22.2.2. The text and diagrams in clause 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 3GPP TS 23.011 [22] and the 3GPP TS 23.08x and 3GPP TS 23.09x-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 3GPP TS 29.011 [59]).

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 3GPP TS 29.011 [59]).

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, clause 22.2.3. The Interrogation procedure is described in figure 22.6.3/1.

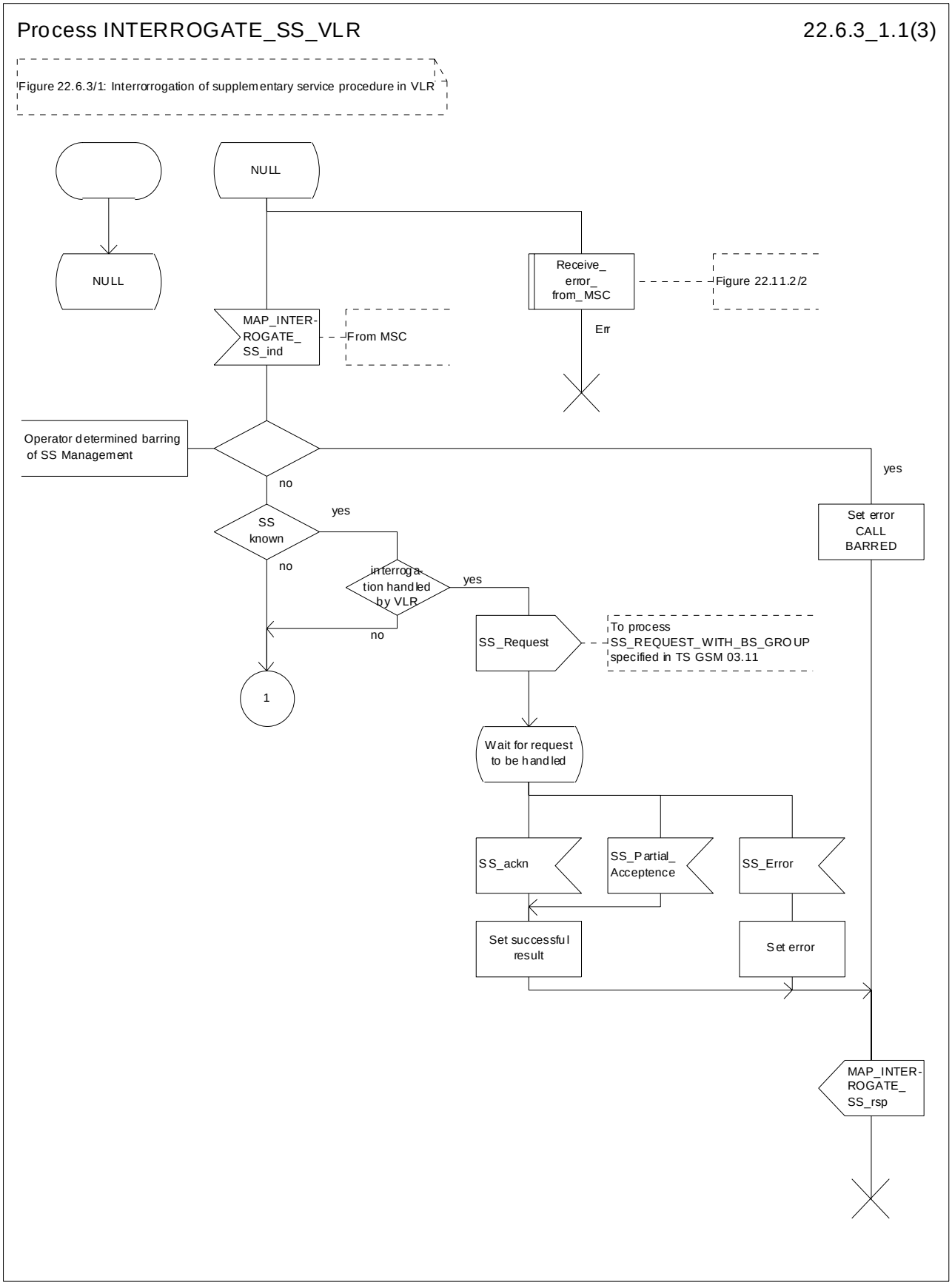


Figure 22.6.3/1 (sheet 1 of 3): Procedure Interrogate_SS_VLR

Process INTERROGATE_SS_VLR 22.6.3_1.2(3)

Figure 22.6.3/1: Interrorrogation of supplementary service procedure in VLR

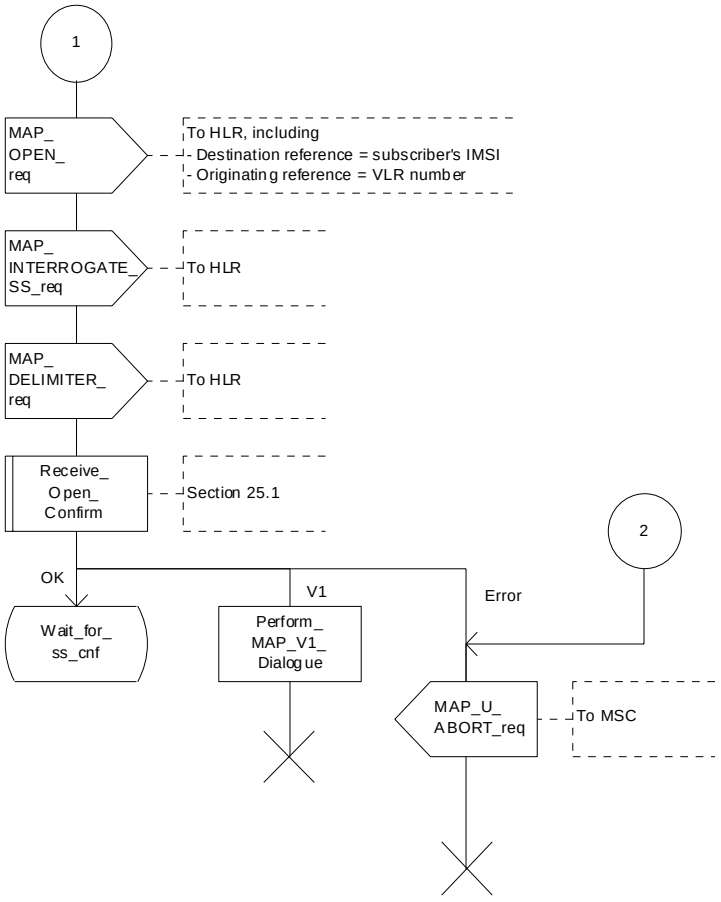


Figure 22.6.3/1 (sheet 2 of 3): Procedure Interrogate_SS_VLR

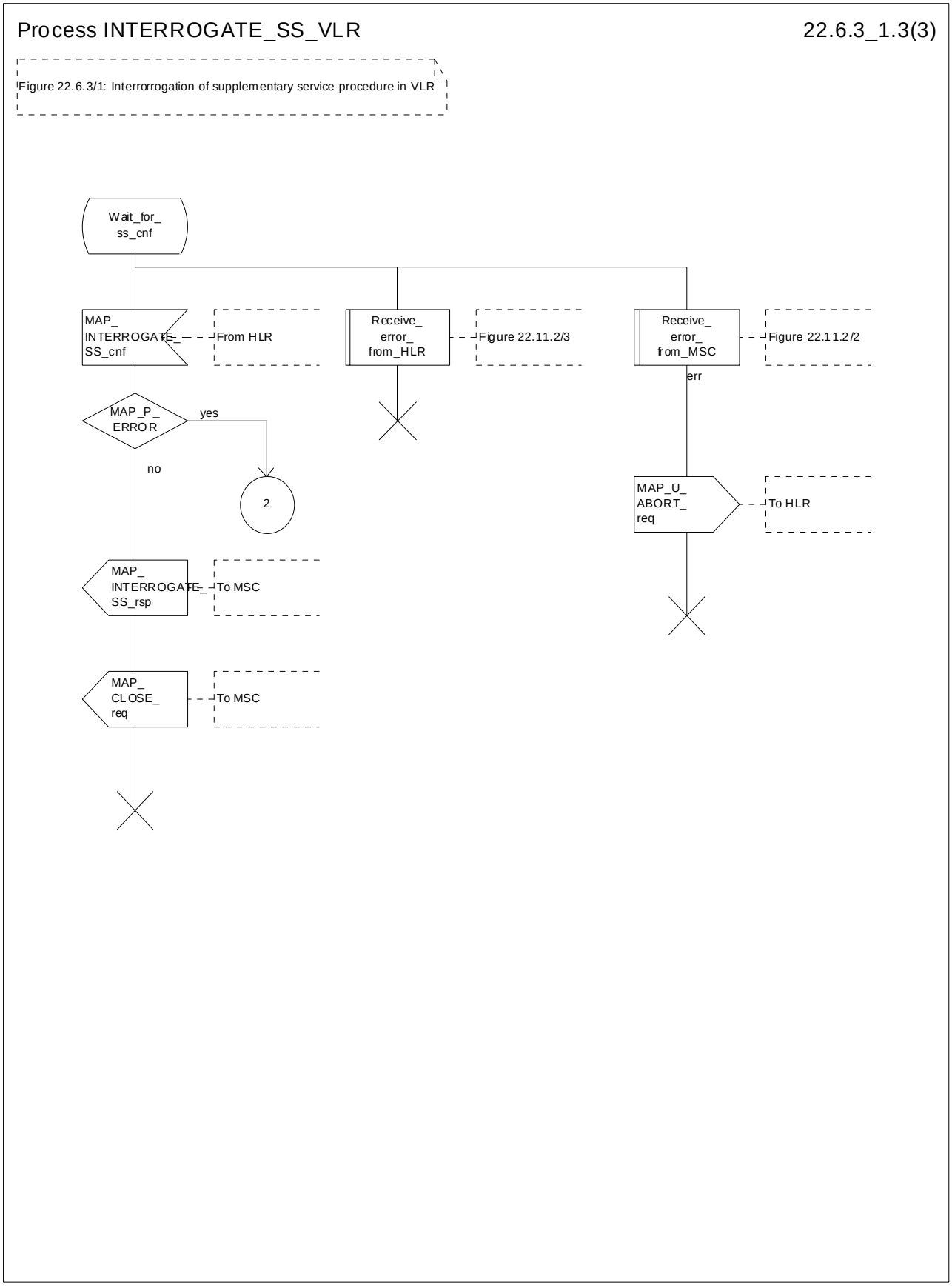


Figure 22.6.3/1 (sheet 3 of 3): Procedure Interrogate_SS_VLR

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 the 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 3GPP TS 23.011 [22] and the 3GPP TS 23.08x and 3GPP TS 23.09x-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, clause 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

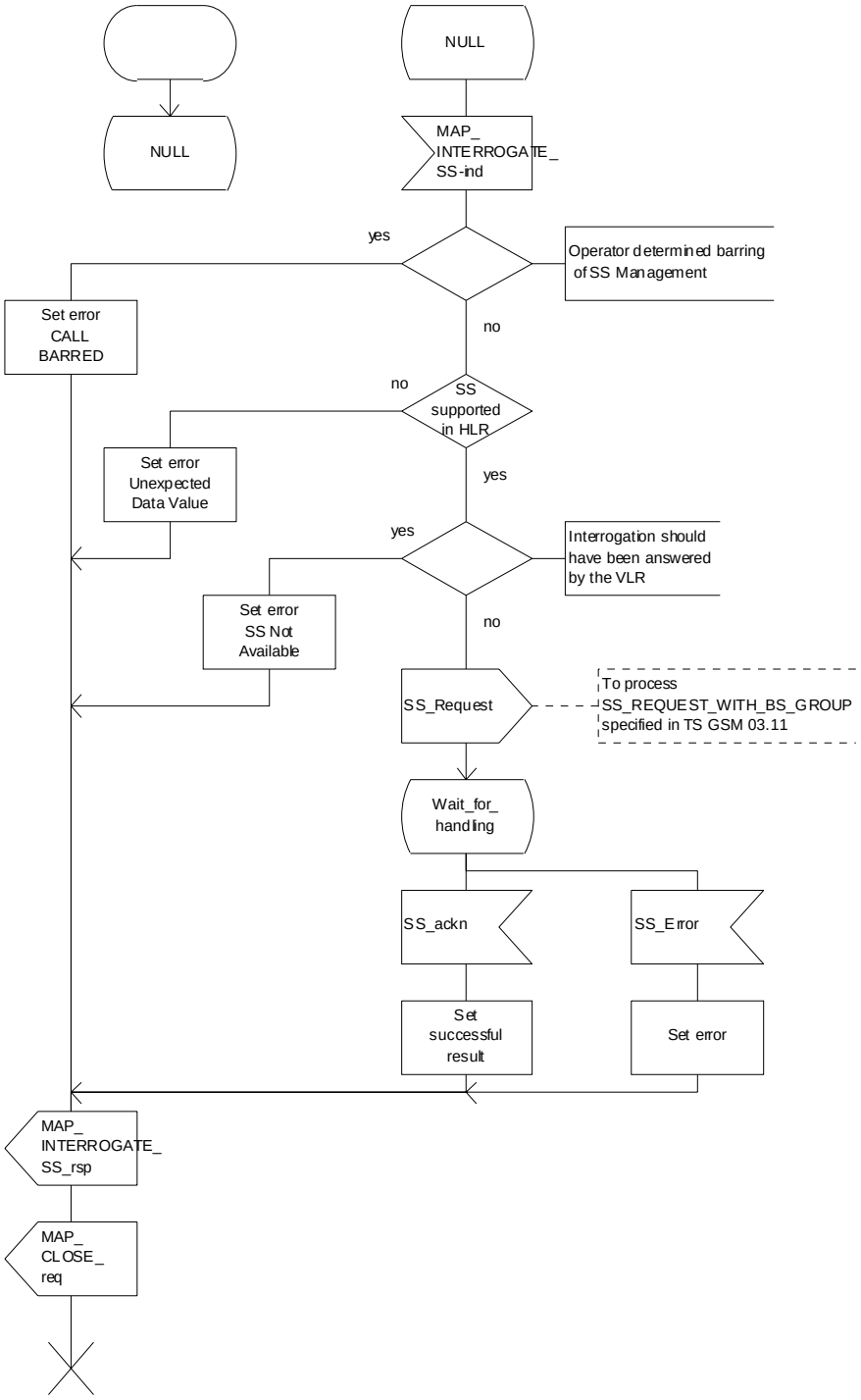


Figure 22.6.4/1: Procedure Interrogate_SS_HLR

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 clause 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).

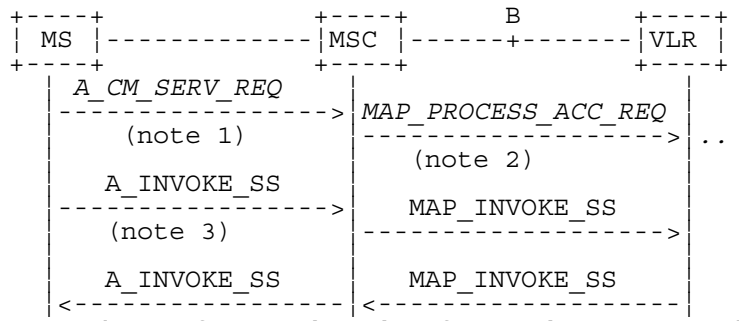


Figure 22.7.1/1: Interfaces and services for supplementary service invocation

NOTE 1: For details of the procedure on the radio path, see 3GPP TS 24.008, 04.10, 3GPP TS 24.08x and 3GPP TS 24.09x. 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_INVOKESS is a generic message to illustrate any supplementary service invocation request message on the air interface, e.g. BuildMPTY, see 3GPP TS 24.080 [38].

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 3GPP TS 24.008 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 3GPP TS 24.010 [36], 3GPP TS 24.08x and 3GPP TS 24.09x-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 3GPP TS 29.011 [59].

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 3GPP TS 22.08x-series, 3GPP TS 23.08x and 3GPP TS 23.09x-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 3GPP TS 29.011 [59] and 3GPP TS 24.08x and 3GPP TS 24.09x-series of technical specifications.

Error handling

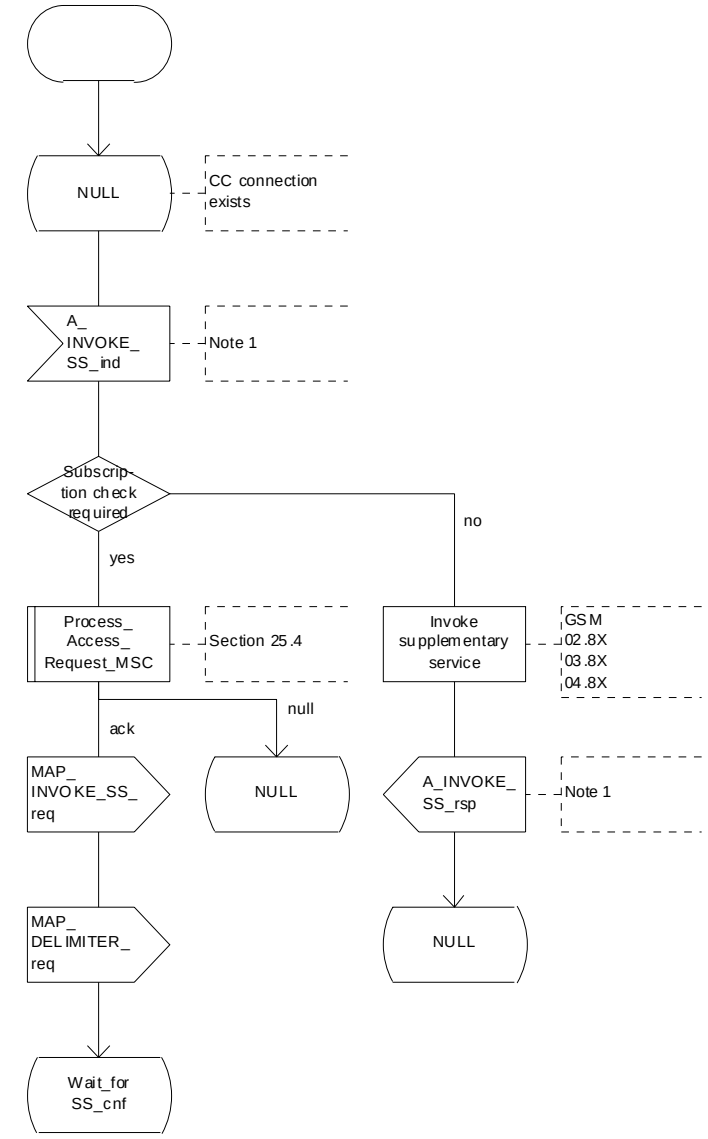
If at any time during this procedure a MAP_P_ABORT, MAP_U_ABORT, MAP_NOTICE or MAP_CLOSE indication concerning the process is received from the VLR, the process is terminated. If a MAP_NOTICE indication was received from the VLR, the VLR dialogue must also be aborted by sending a MAP_U_ABORT request indicating Procedure error towards the VLR. Possible signalling to the MS is described in 3GPP TS 24.010 [36].

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

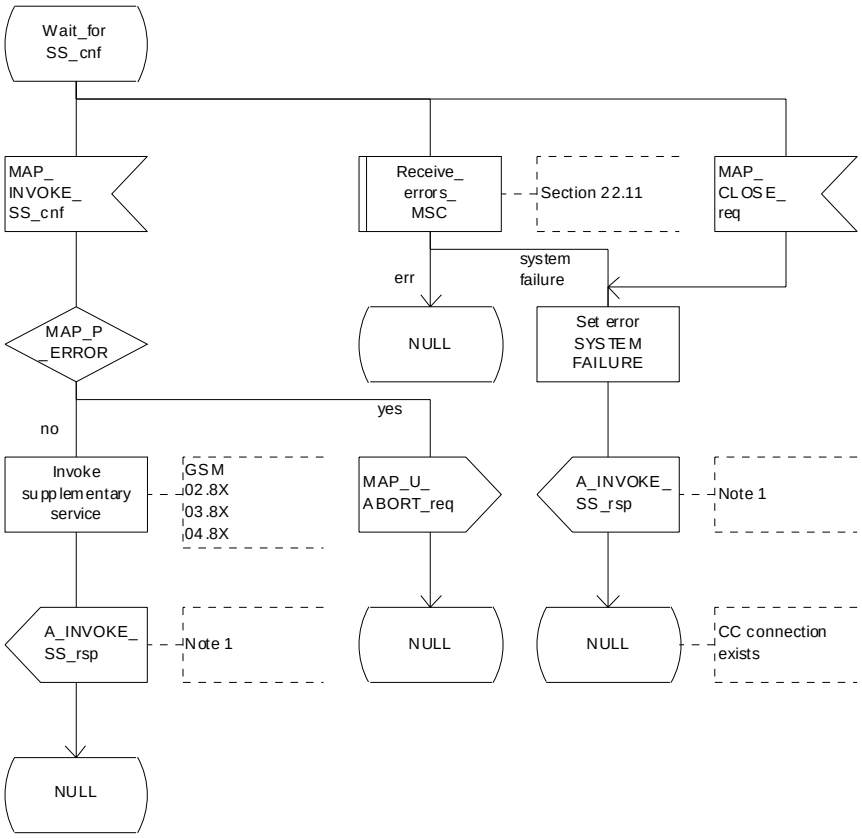


Note 1: Fictitious signal to indicate receipt/sending of SS invocation invoke component on the air interface (eg. BuildMPTY). Described in GSM 04.8X and 09.11.

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 04.8X 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.

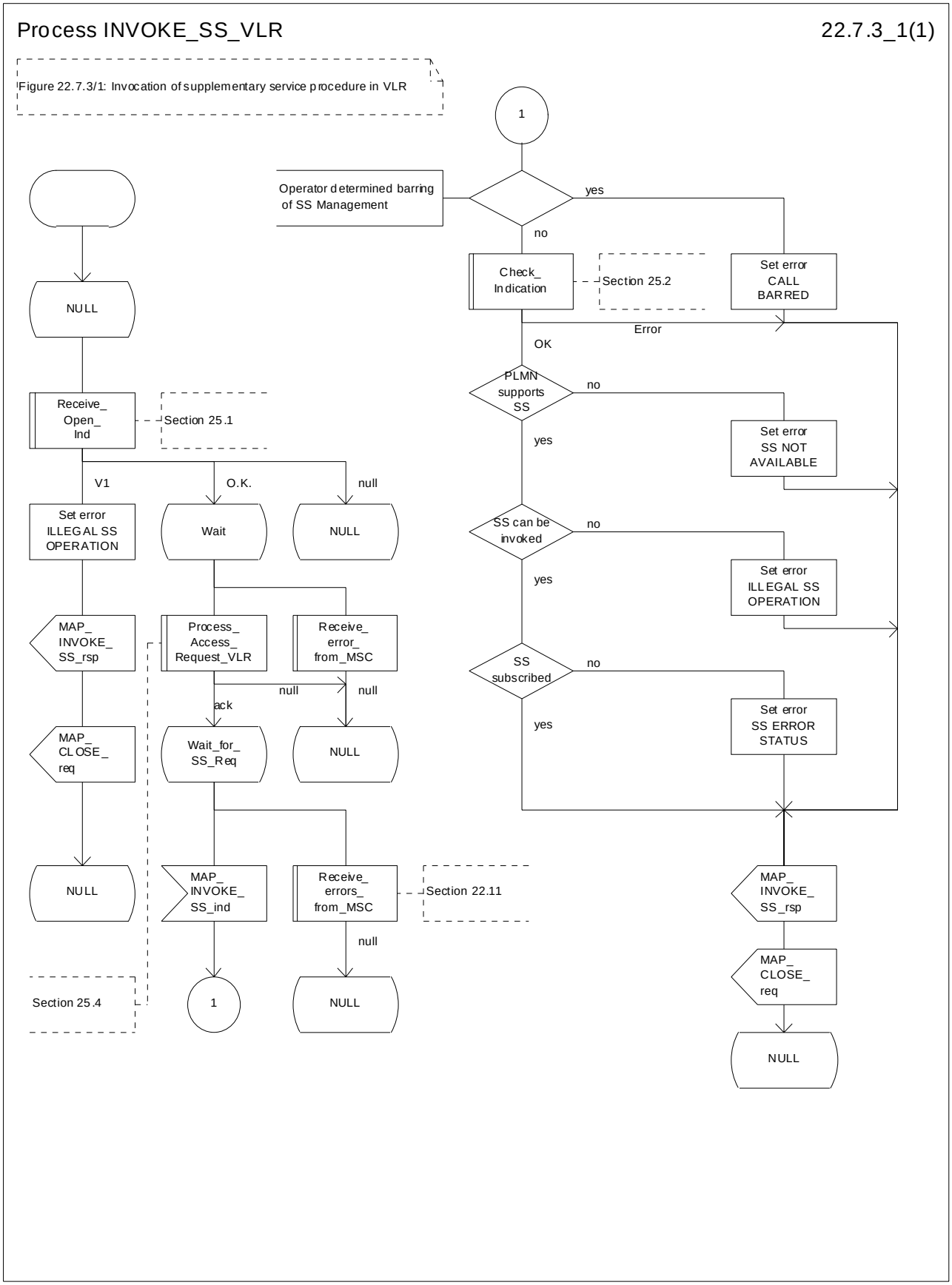


Figure 22.7.3/1: Procedure Invoke_SS_VLR

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).

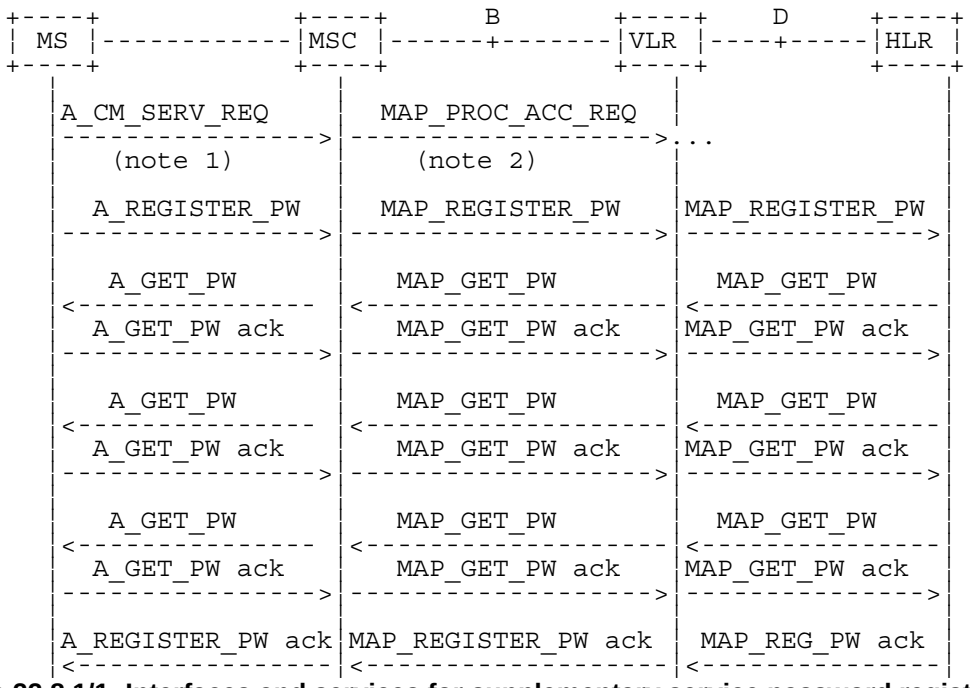


Figure 22.8.1/1: Interfaces and services for supplementary service password registration

NOTE 1: For details of the procedure on the radio path, see 3GPP TS 24.008, 04.10, 3GPP TS 24.08x and 3GPP TS 24.09x. 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 clause 22.8.4.

22.8.2 Procedures in the MSC

The password registration procedure in the MSC is identical to that for activation specified in clause 22.4.2. All the text and diagrams in clause 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 clause 22.4.3. All the text and diagrams in clause 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 3GPP TS 23.011 [22]. During the handling of password registration, the password procedure will be initiated (as specified in 3GPP TS 23.011 [22]) 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 clause 22.2.4 above.

The password registration procedure in the HLR is shown in figure 22.8.4/1.

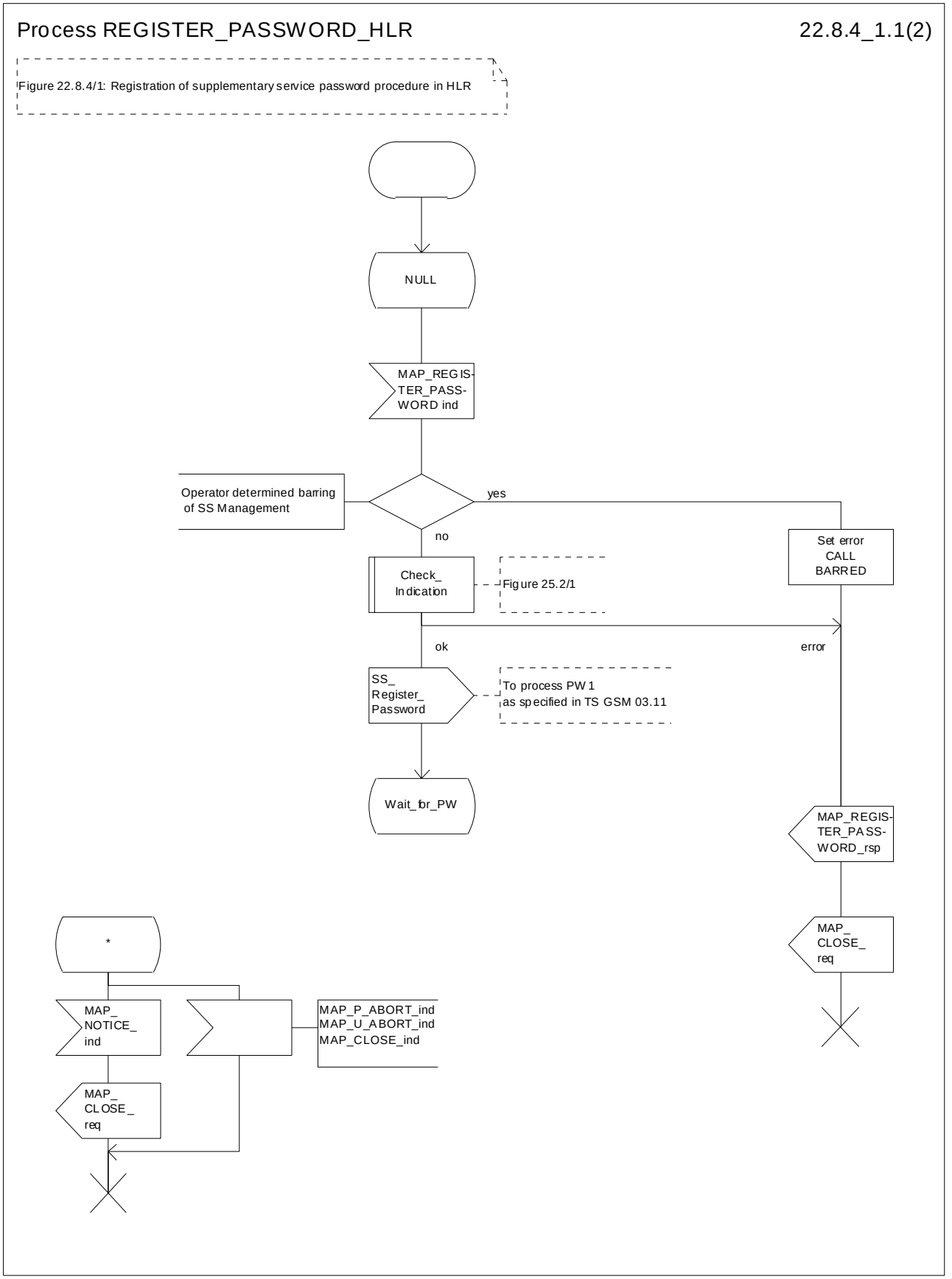


Figure 22.8.4/1 (sheet 1 of 2): Procedure Register_PW_HLR

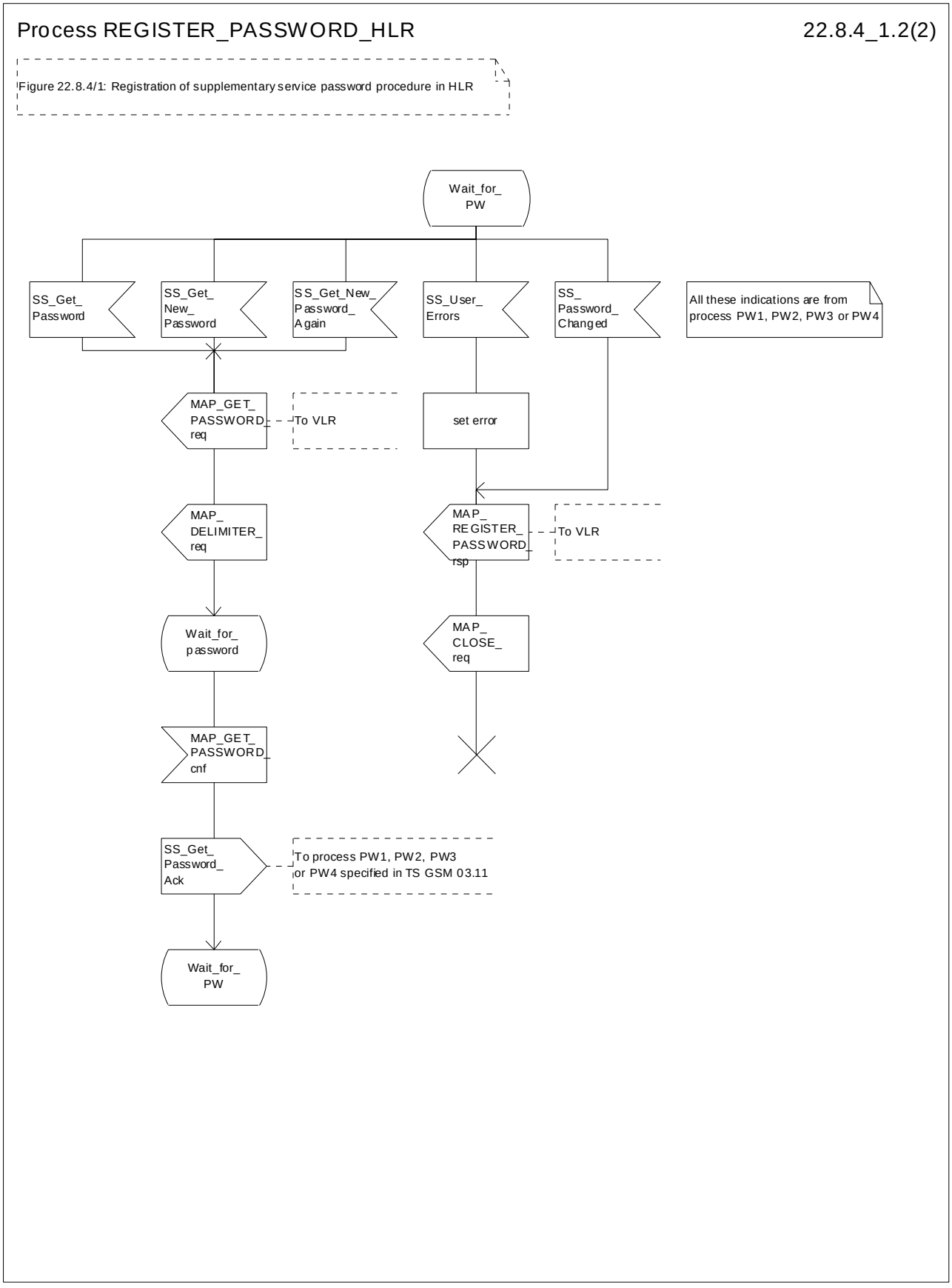


Figure 22.8.4/1 (sheet 2 of 2): Procedure Register_PW_HLR

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 3GPP TS 23.090 [34].

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 3GPP TS 23.090 [34].

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 3GPP TS 29.011 [59].

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.

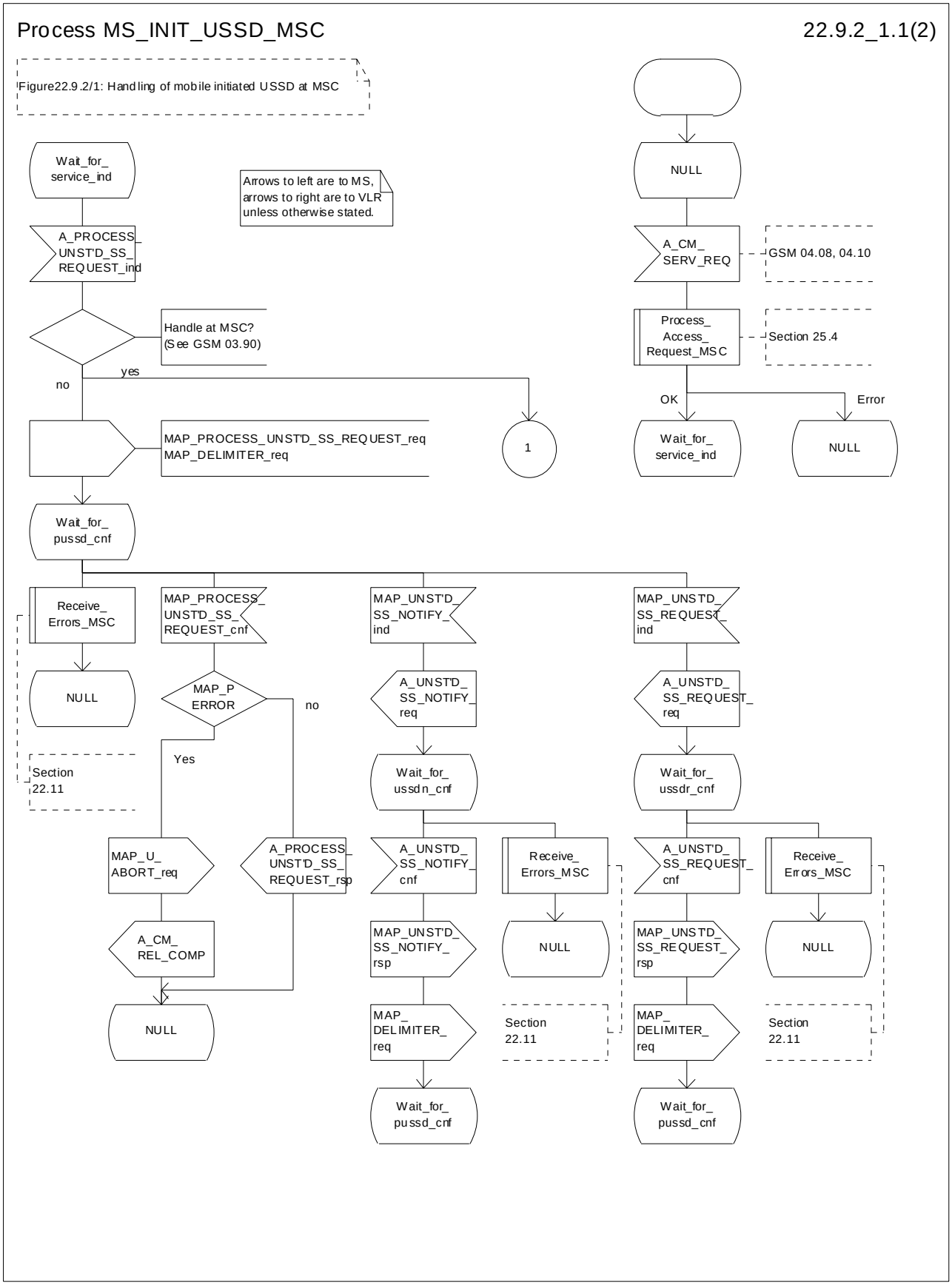


Figure 22.9.2/1 (sheet 1 of 2): Procedure MI_USSD_MSC

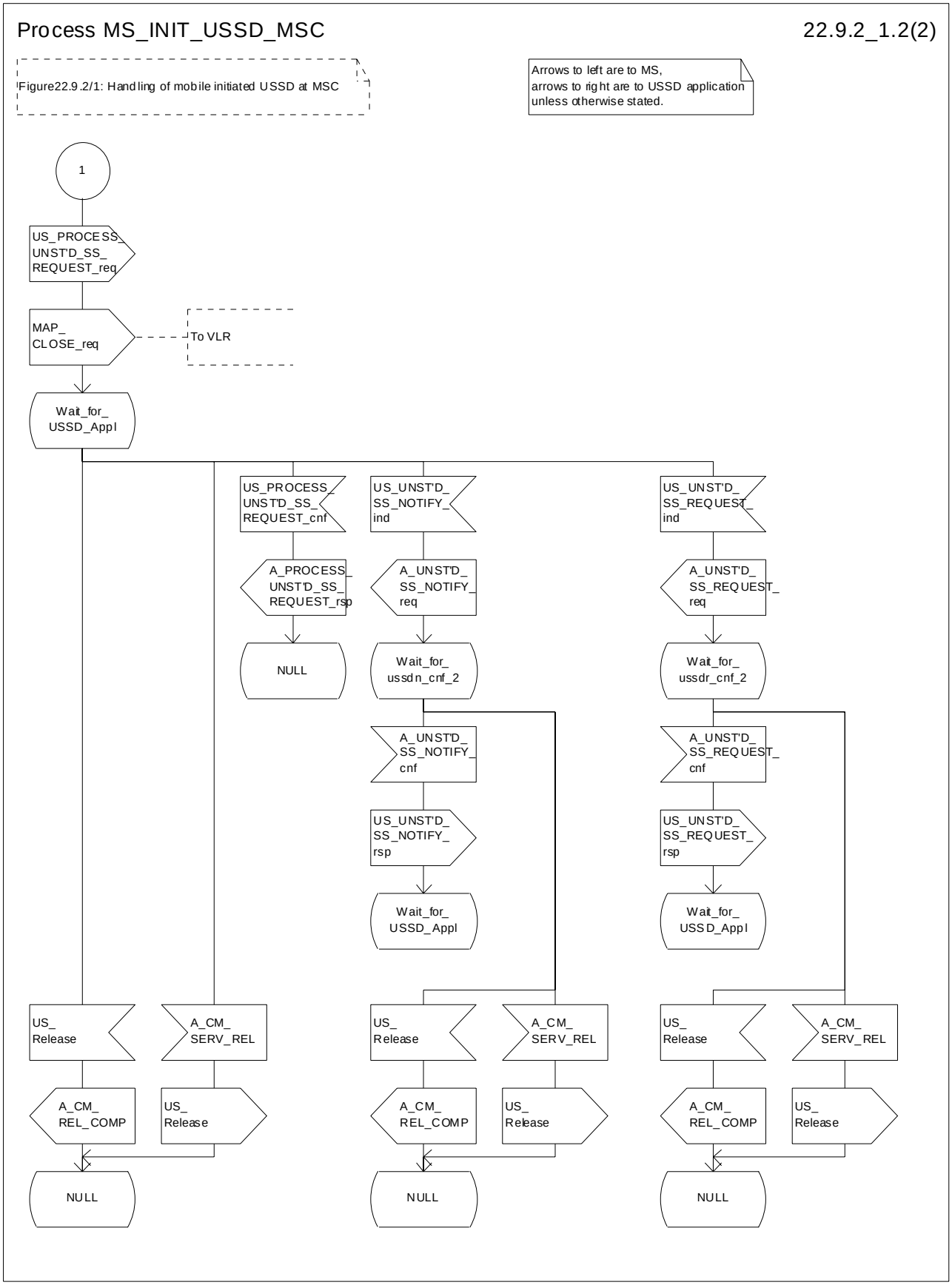


Figure 22.9.2/1 (sheet 2 of 2): Procedure MI_USSD_MSC

22.9.3 Procedures in the VLR

The initiation of the process is shown in clause 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 3GPP TS 23.090 [34].

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.

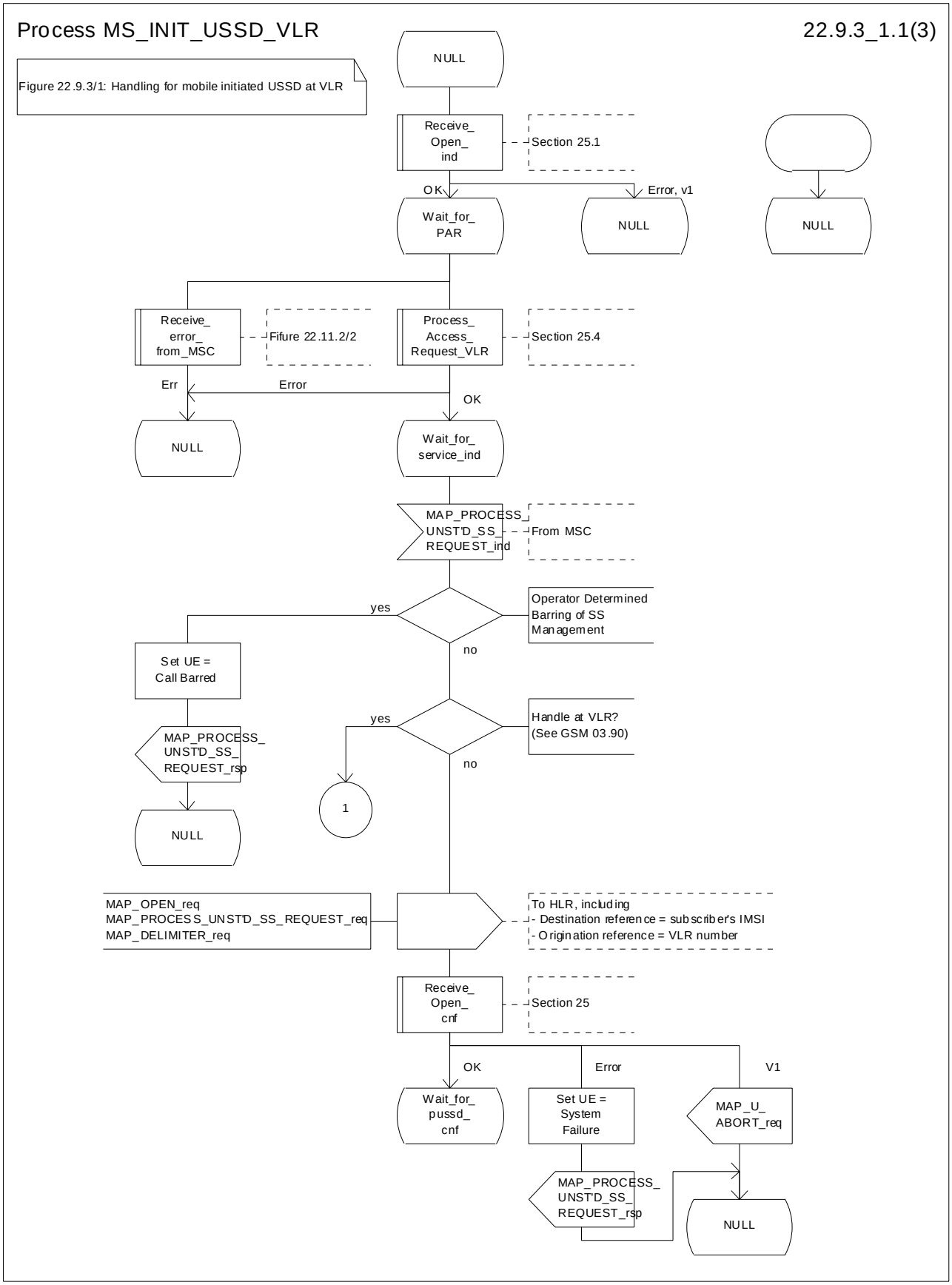


Figure 22.9.3/1 (sheet 1 of 3): Procedure MI_USSD_VLR

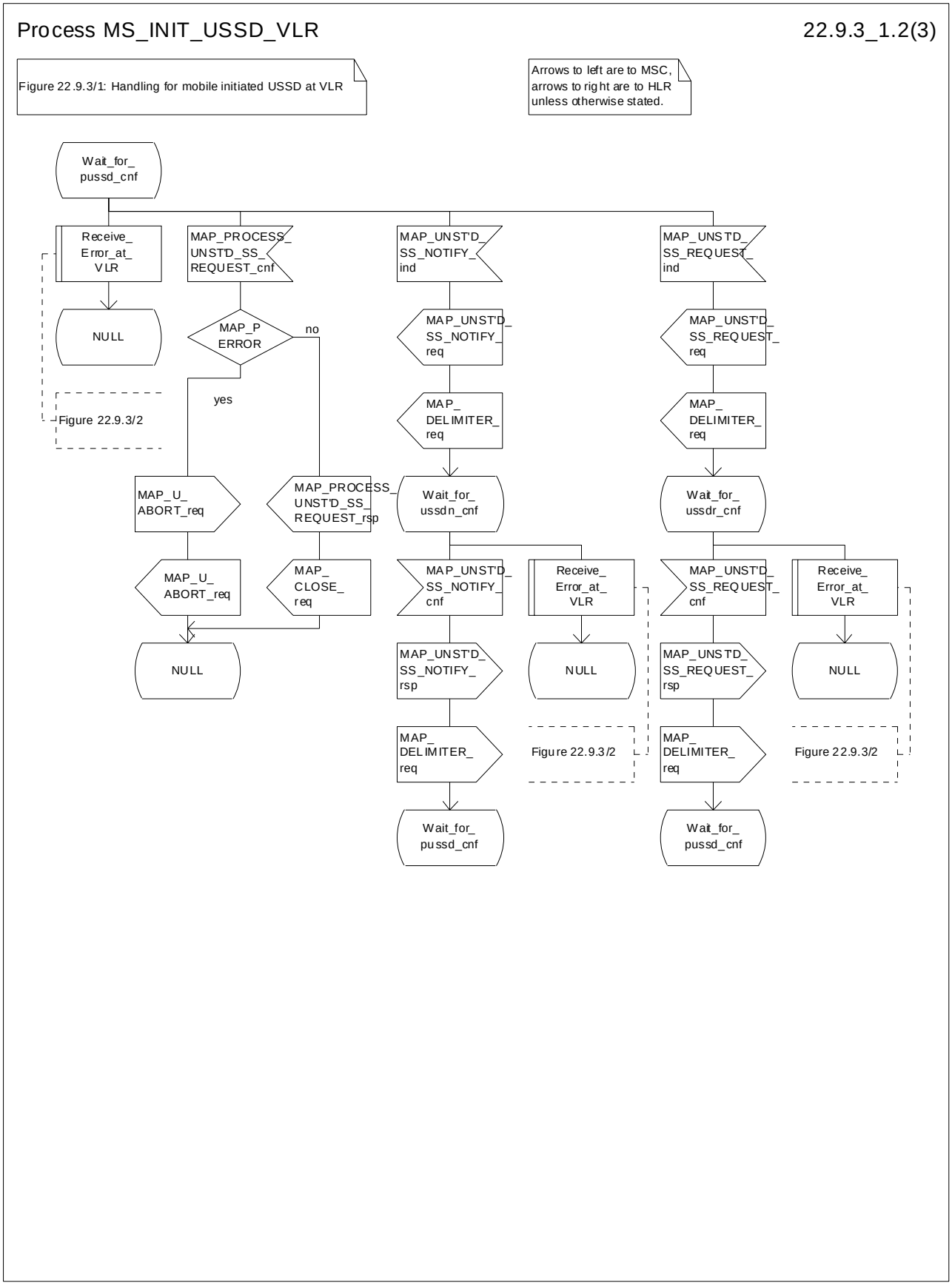


Figure 22.9.3/1 (sheet 2 of 3): Procedure MI_USSD_VLR

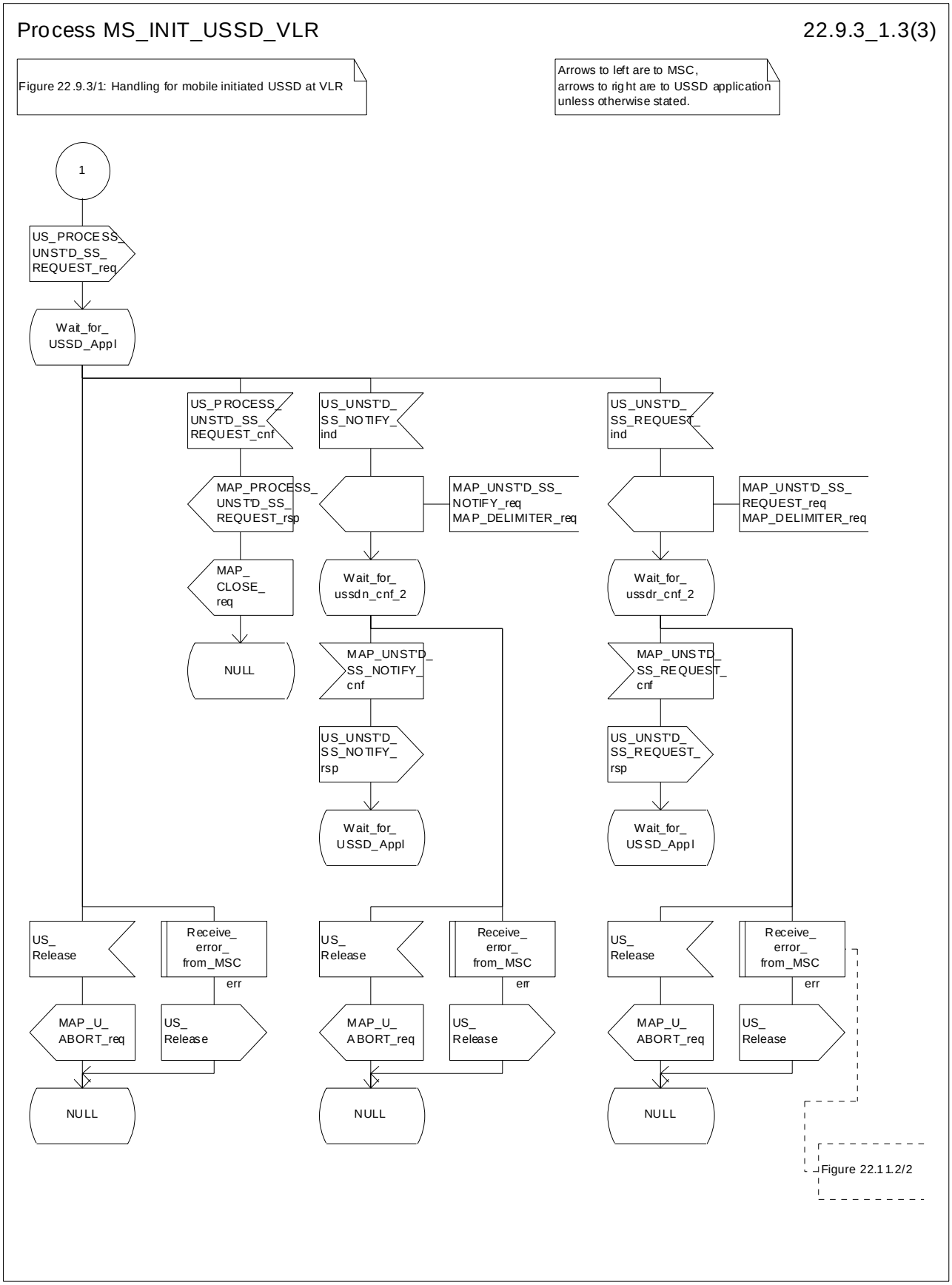


Figure 22.9.3/1 (sheet 3 of 3): Procedure_MI_USSD_VLR

Macrodefinition Receive_Error_at_VLR 22.9.3_2(1)

Figure 22.9.3/2: Handling of errors at VLR for USSD

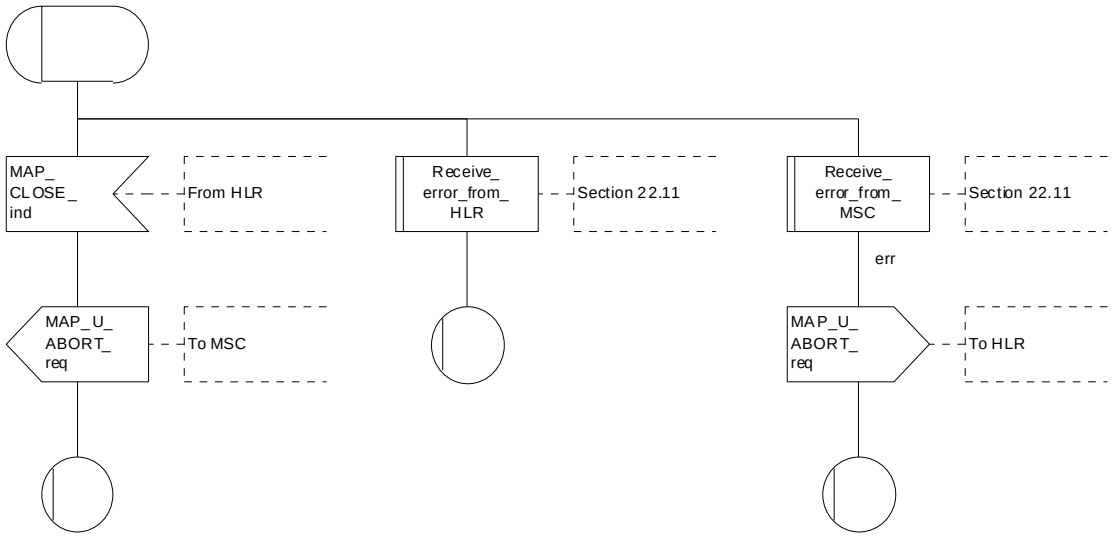


Figure 22.9.3/2: Macro Receive_Error_at_VLR

22.9.4 Procedures in the HLR

The Mobile initiated USSD Procedure in the HLR starts by the HLR receiving a MAP-OPEN service indication from the VLR.

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 HLR is shown in figure 22.9.4/1.

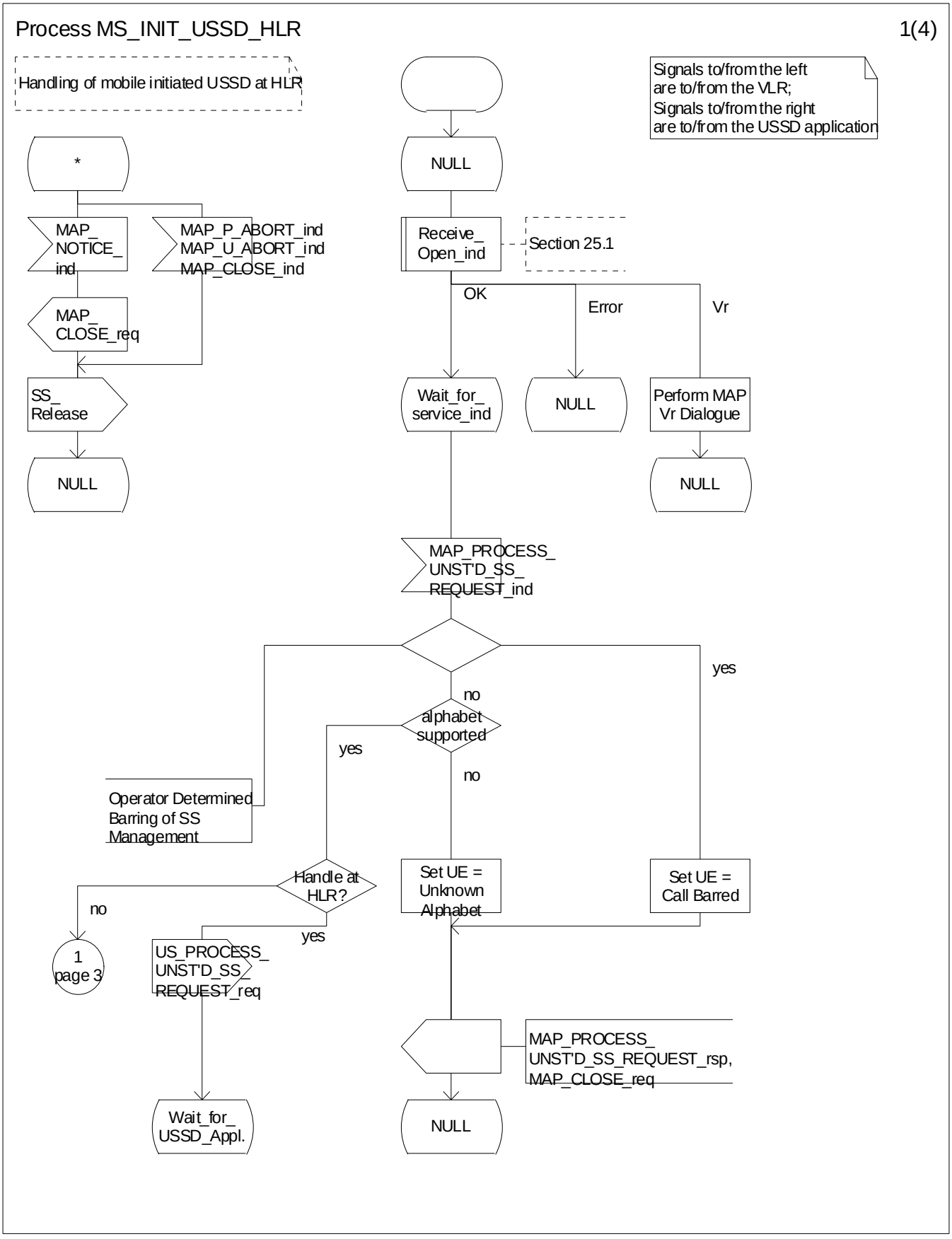


Figure 22.9.4/1 (sheet 1 of 4): Procedure MI_USSD_HLR

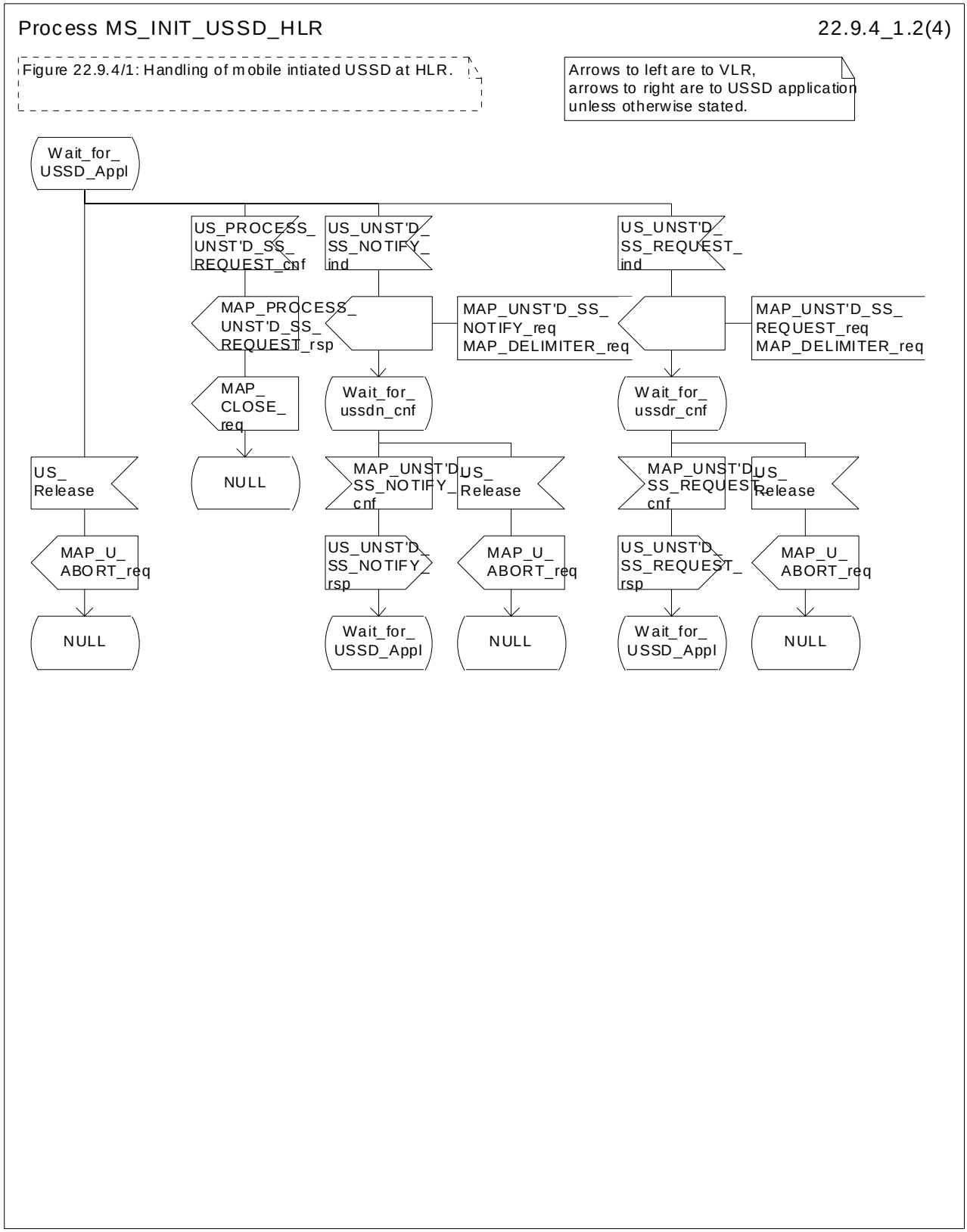


Figure 22.9.4/1 (sheet 2 of 4): Procedure MI_USSD_HLR

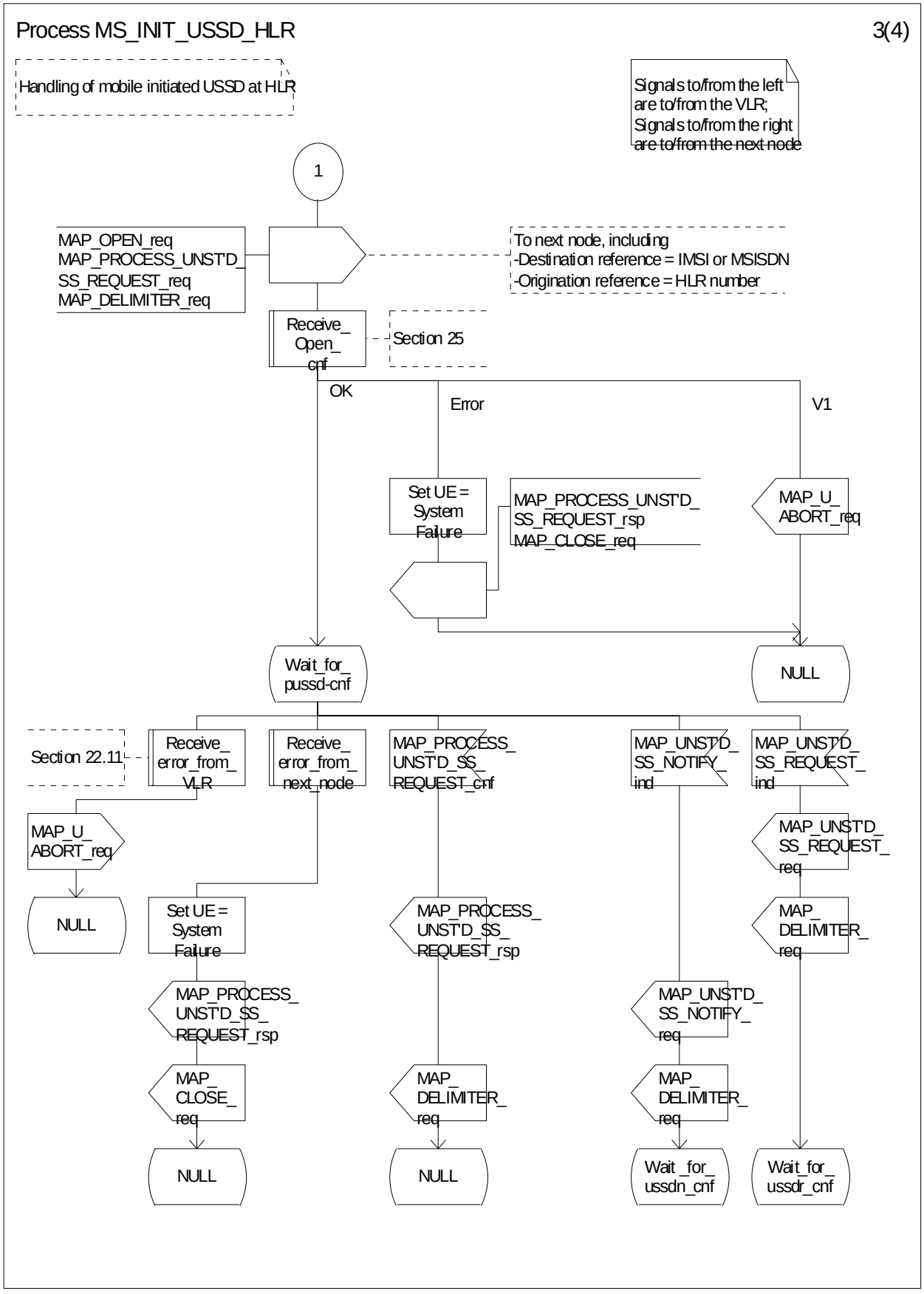


Figure 22.9.4/1 (sheet 3 of 4): Procedure MI_USSD_HLR

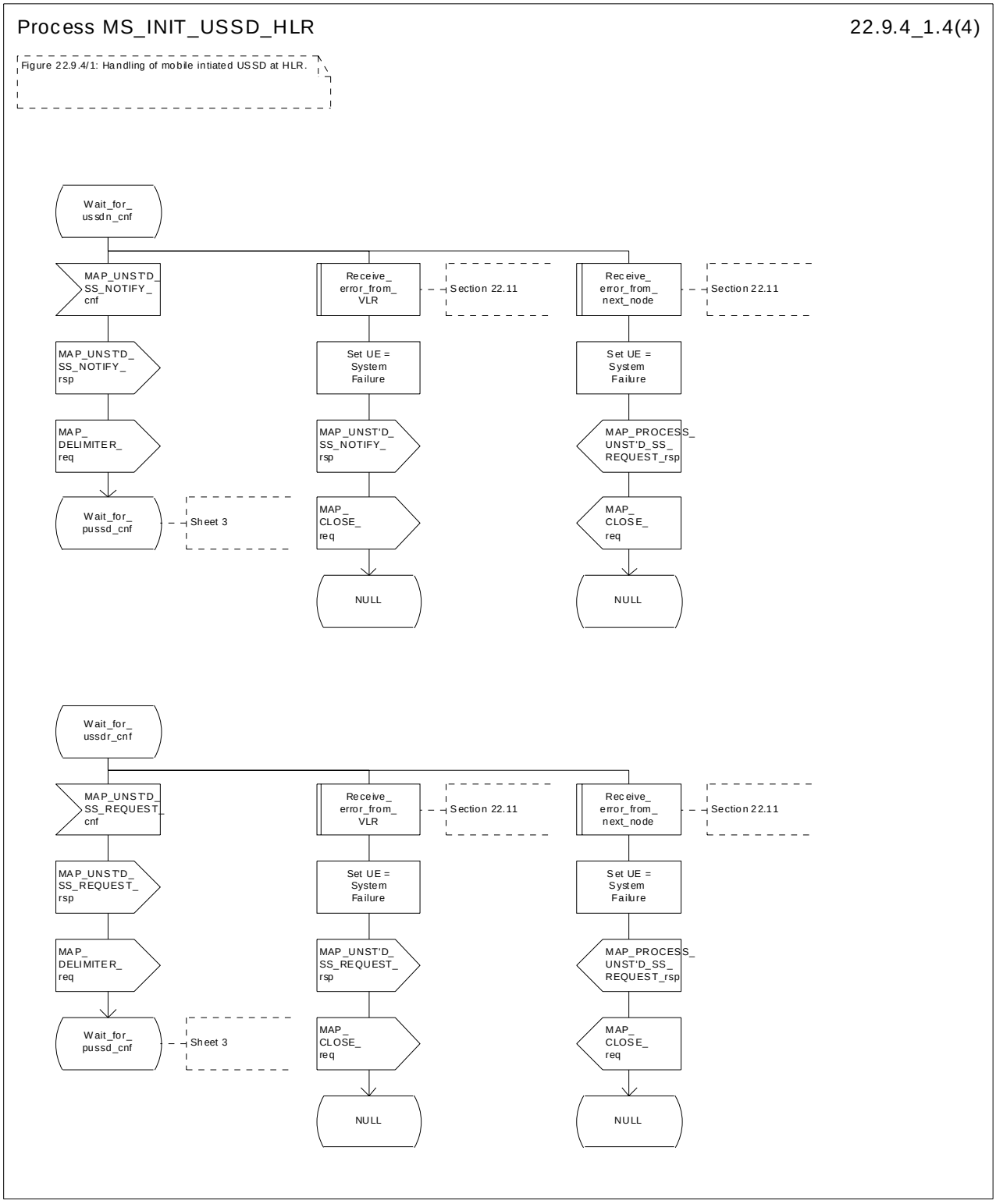


Figure 22.9.4/1 (sheet 4 of 4): Procedure MI_USSD_HLR

22.9.5 Procedures in the gsmSCF/secondary HLR

The Mobile initiated USSD Procedure in the gsmSCF/secondary HLR starts by the gsmSCF/secondary HLR receiving a MAP-OPEN service indication from the HLR.

Once a MAP dialogue is established, the gsmSCF/secondary HLR may handle the MAP_PROCESS_UNSTRUCTURED_SS_REQUEST from the HLR.

The gsmSCF/secondary HLR shall transfer the message to the local application.

The gsmSCF/secondary 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 HLR. When a confirmation is received from the HLR this shall be returned to the application.

When the gsmSCF/secondary HLR receives the result of the original operation from the application then it shall pass this to the HLR and initiate release of the CM connection.

Error Handling

Both the HLR and the USSD Application may initiate release of the MAP service at any time. This is handled as shown in the diagrams.

The procedure in the gsmSCF and secondary HLR is shown in figure 22.9.5/1.

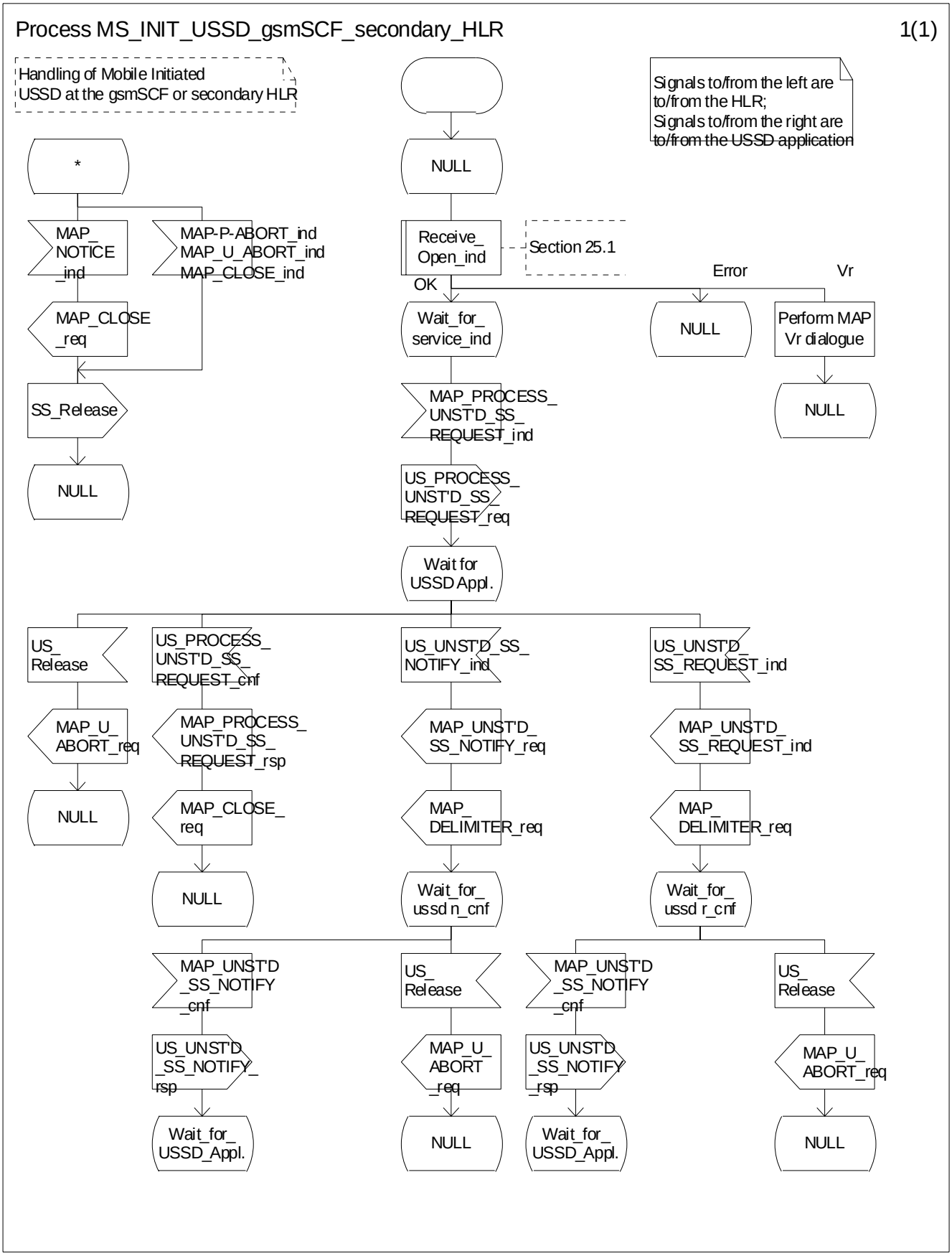


Figure 22.9.5/1 Process MS_INIT_USSD_gsmSCF_secondary_HLR

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 3GPP TS 23.090 [34].

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 VLR.

In both cases the MSC will initiate a CM connection to the MS (using the page or search macros defined in clause 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 3GPP TS 29.011 [59].

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.

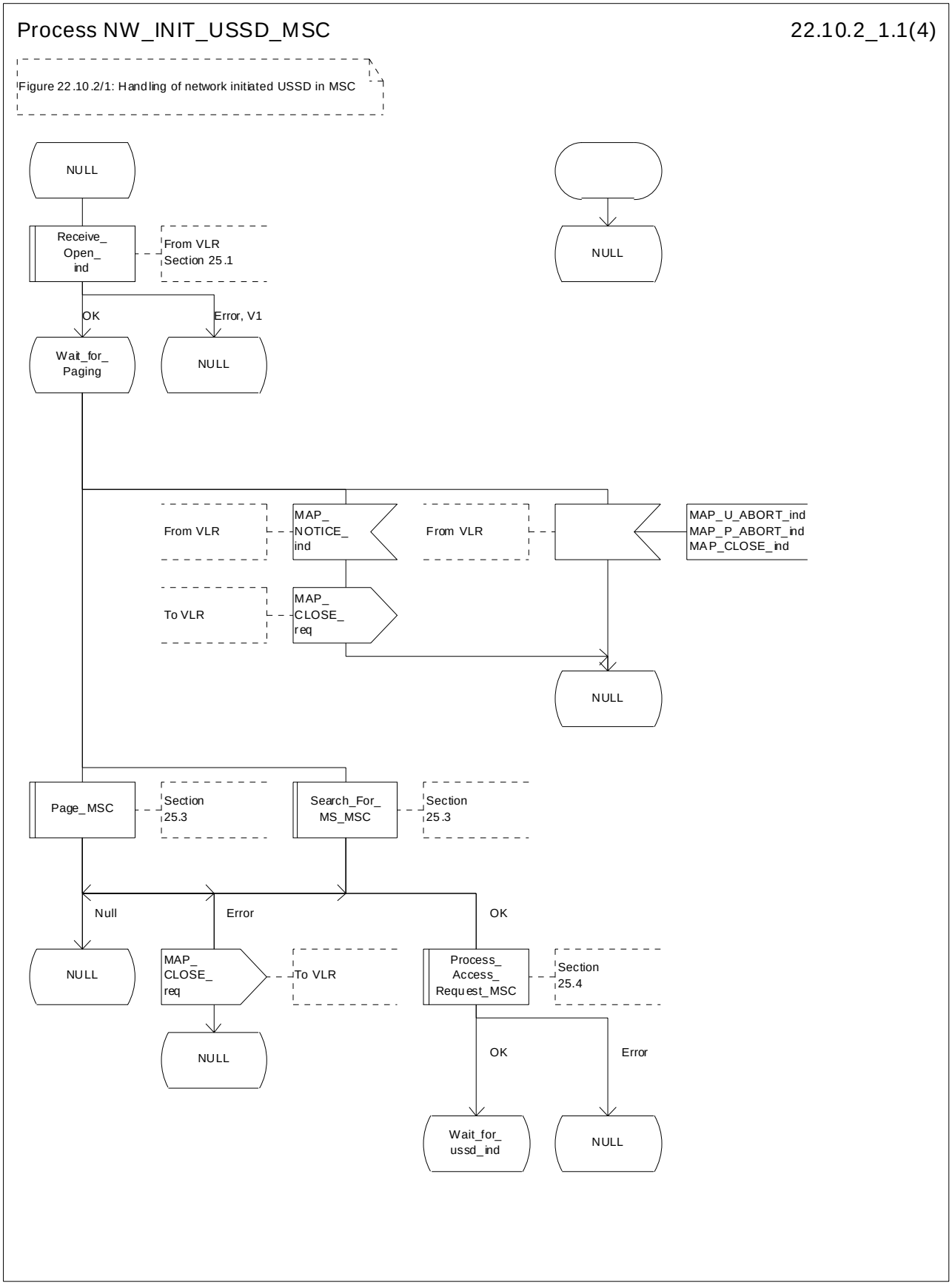


Figure 22.10.2/1 (sheet 1 of 4): Procedure NI_USSD_MSC

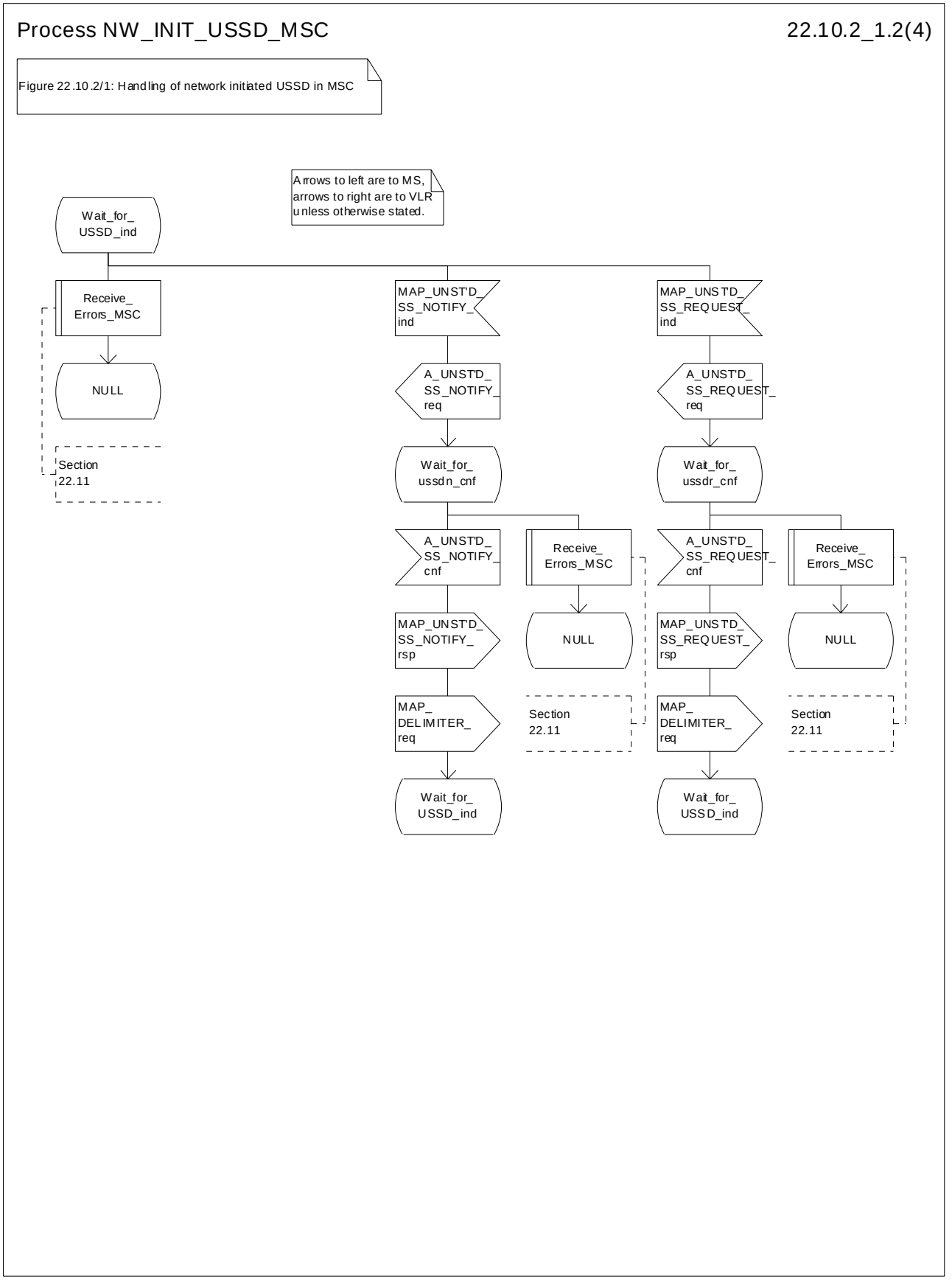


Figure 22.10.2/1 (sheet 2 of 4): Procedure NI_USSD_MSC

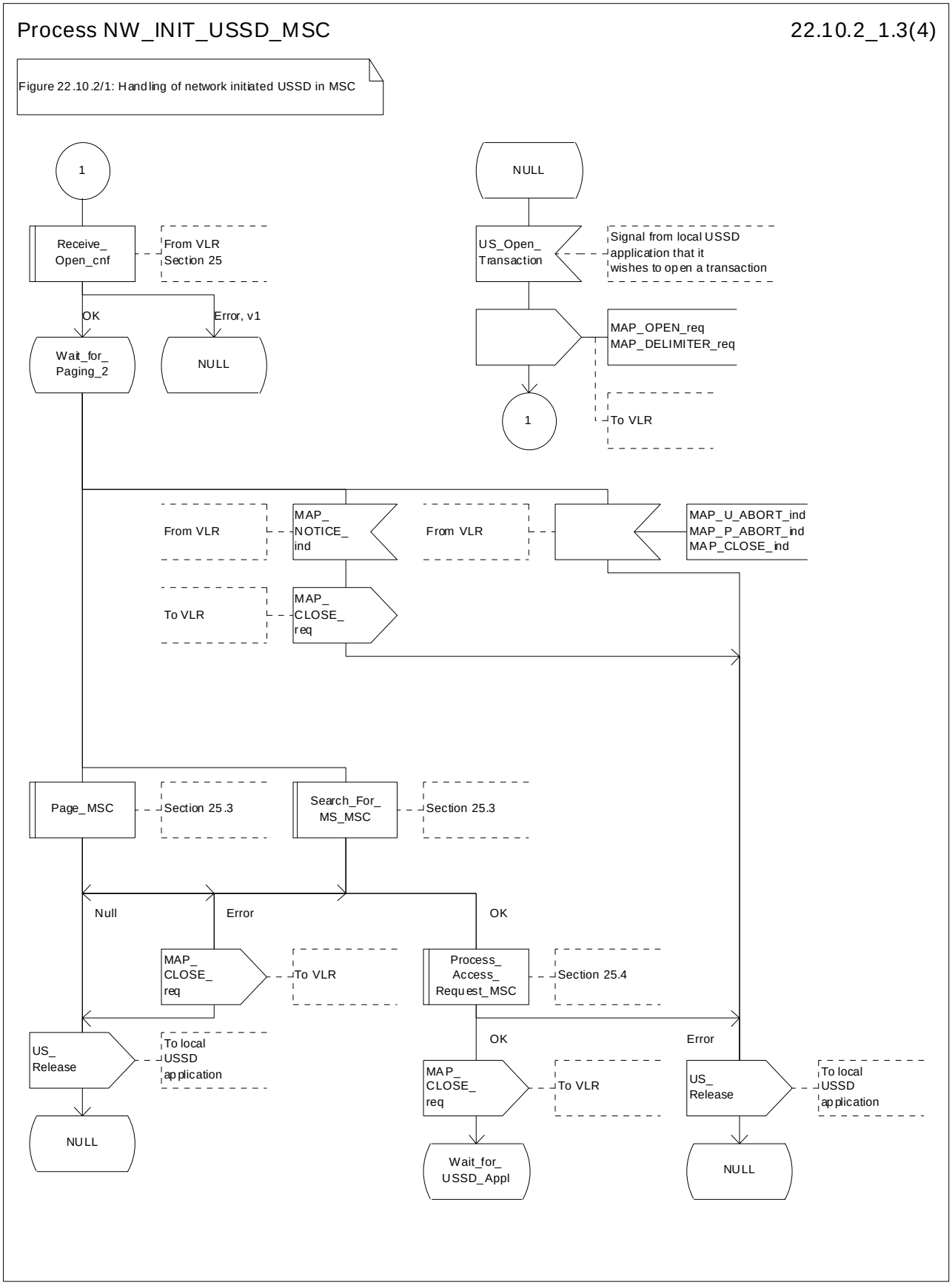


Figure 22.10.2/1 (sheet 3 of 4): Procedure NI_USSD_MSC

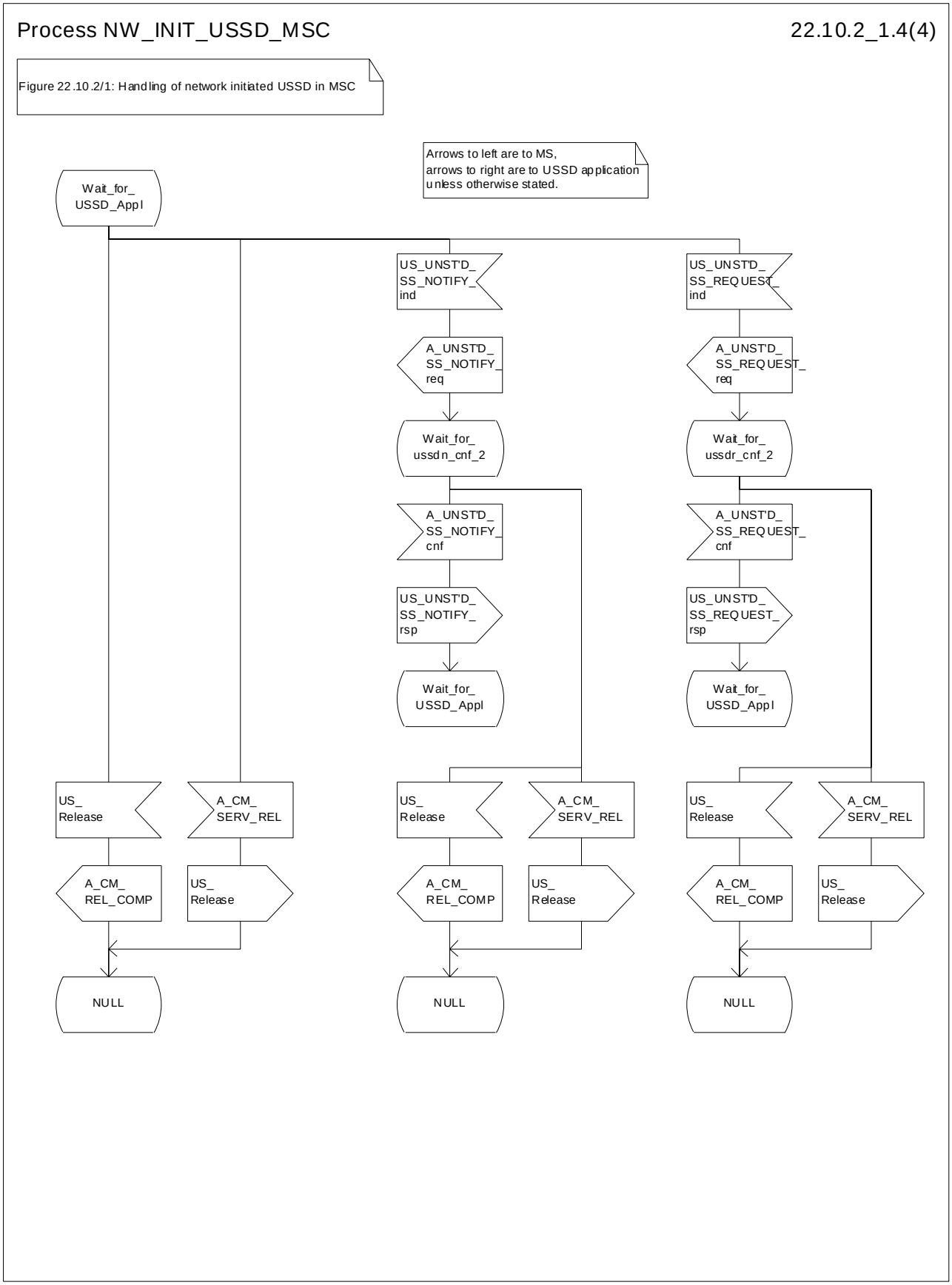


Figure 22.10.2/1 (sheet 4 of 4): Procedure NI_USSD_MSC

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 clause 25.4 will be used to handle the CM connection establishment.

The macro is shown in figure 22.10.3/2.

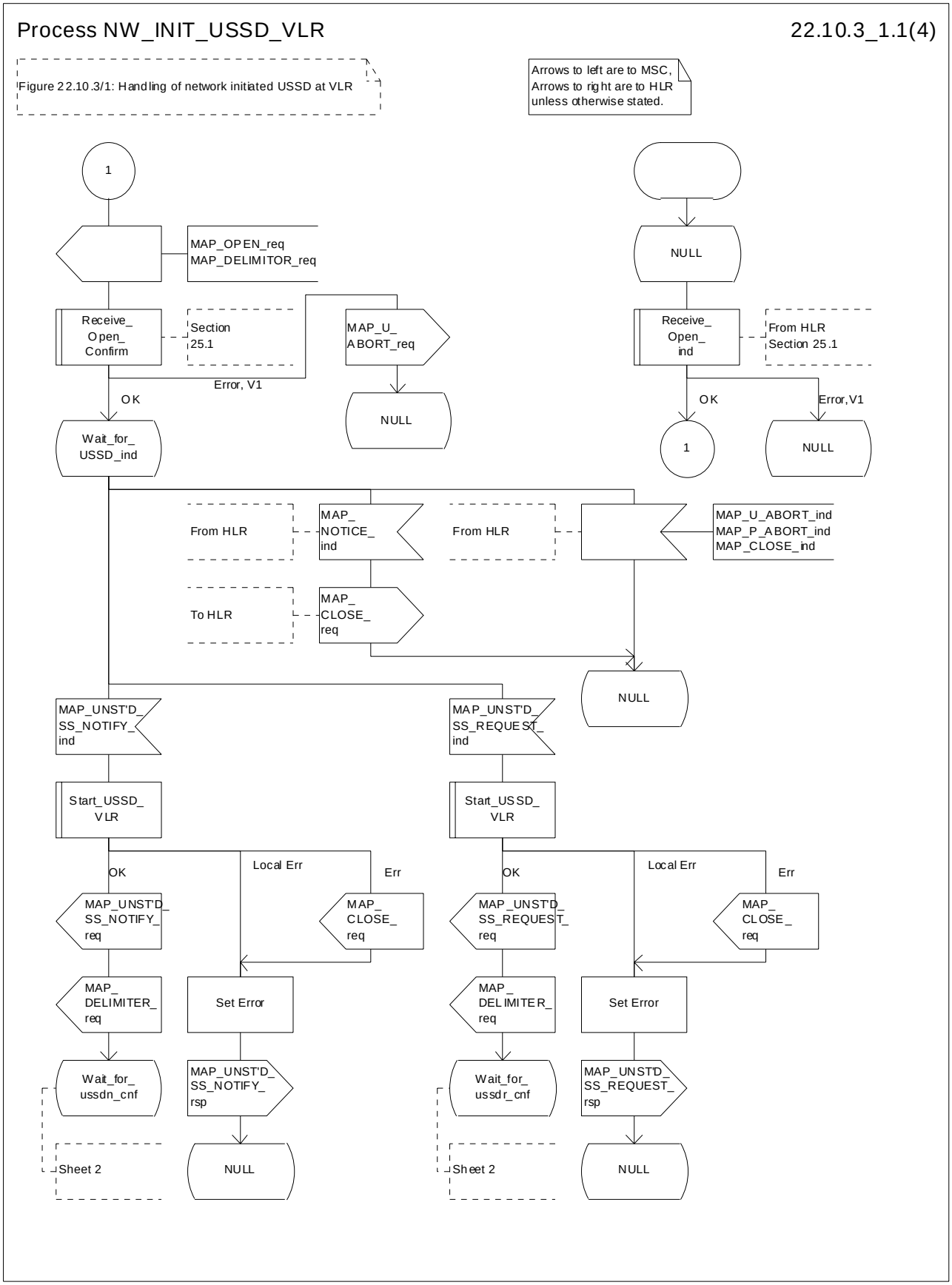


Figure 22.10.3/1 (sheet 1 of 4): Procedure NI_USSD_VLR

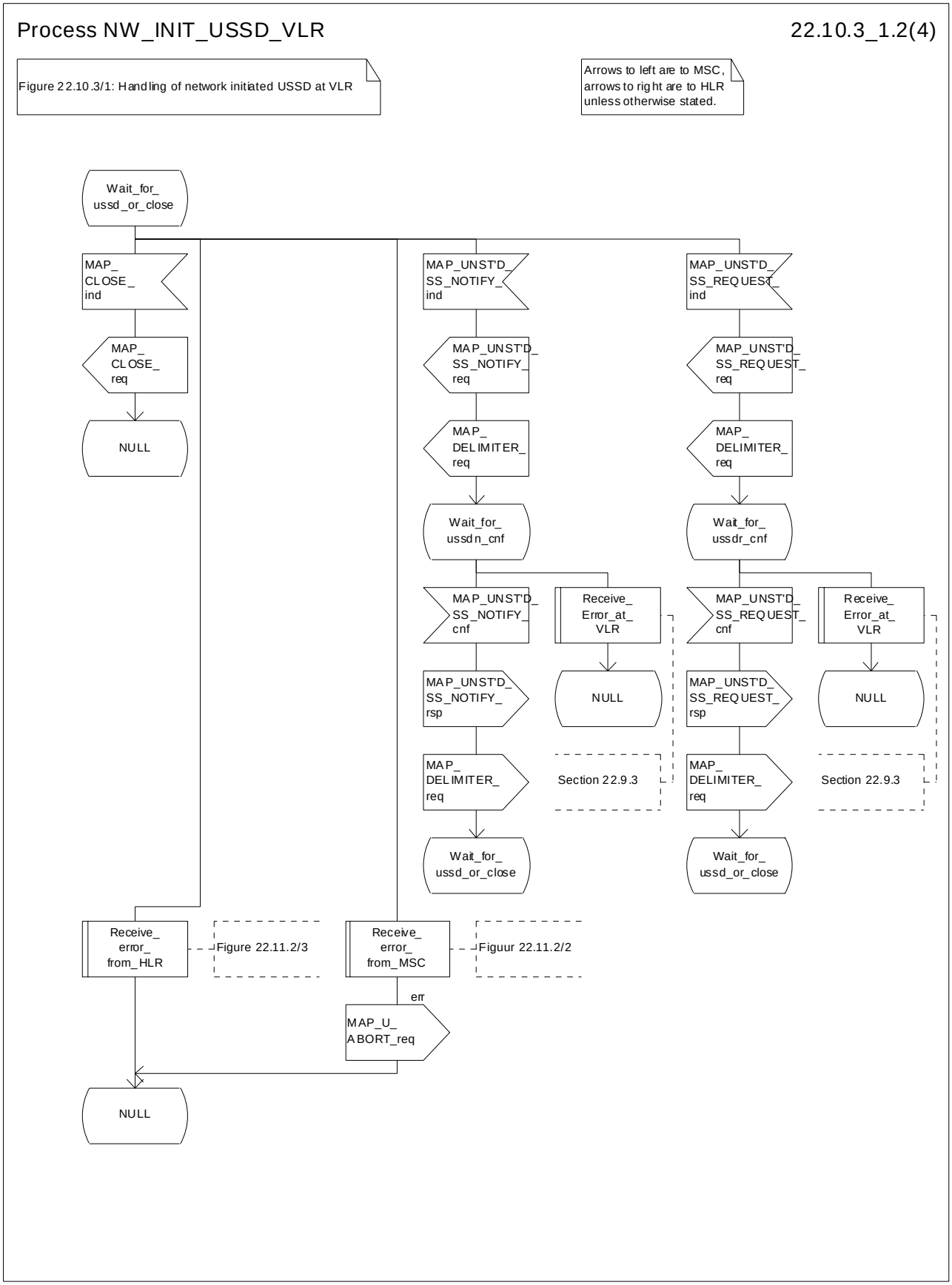


Figure 22.10.3/1 (sheet 2 of 4): Procedure NI_USSD_VLR

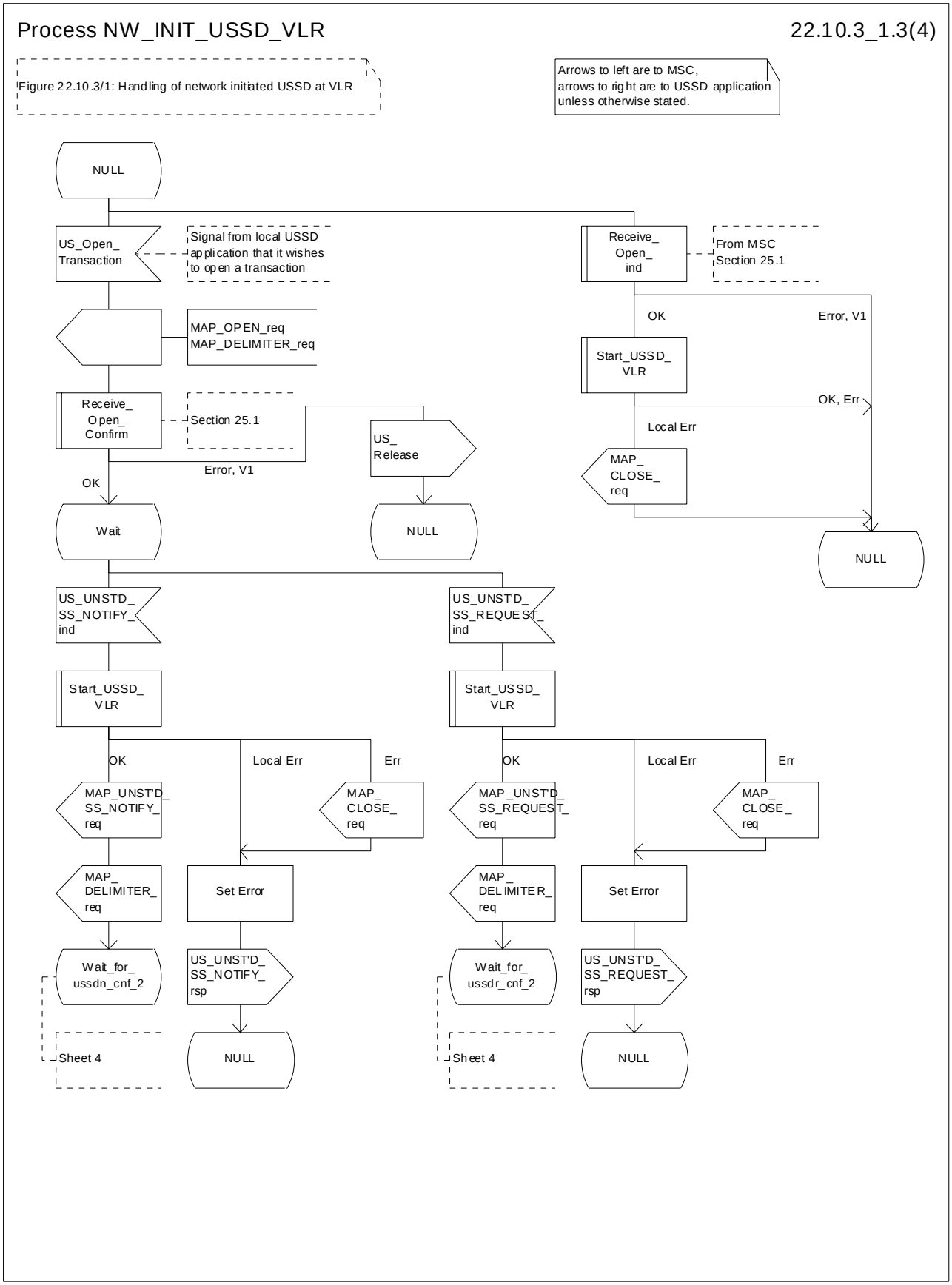


Figure 22.10.3/1 (sheet 3 of 4): Procedure NI_USSD_VLR

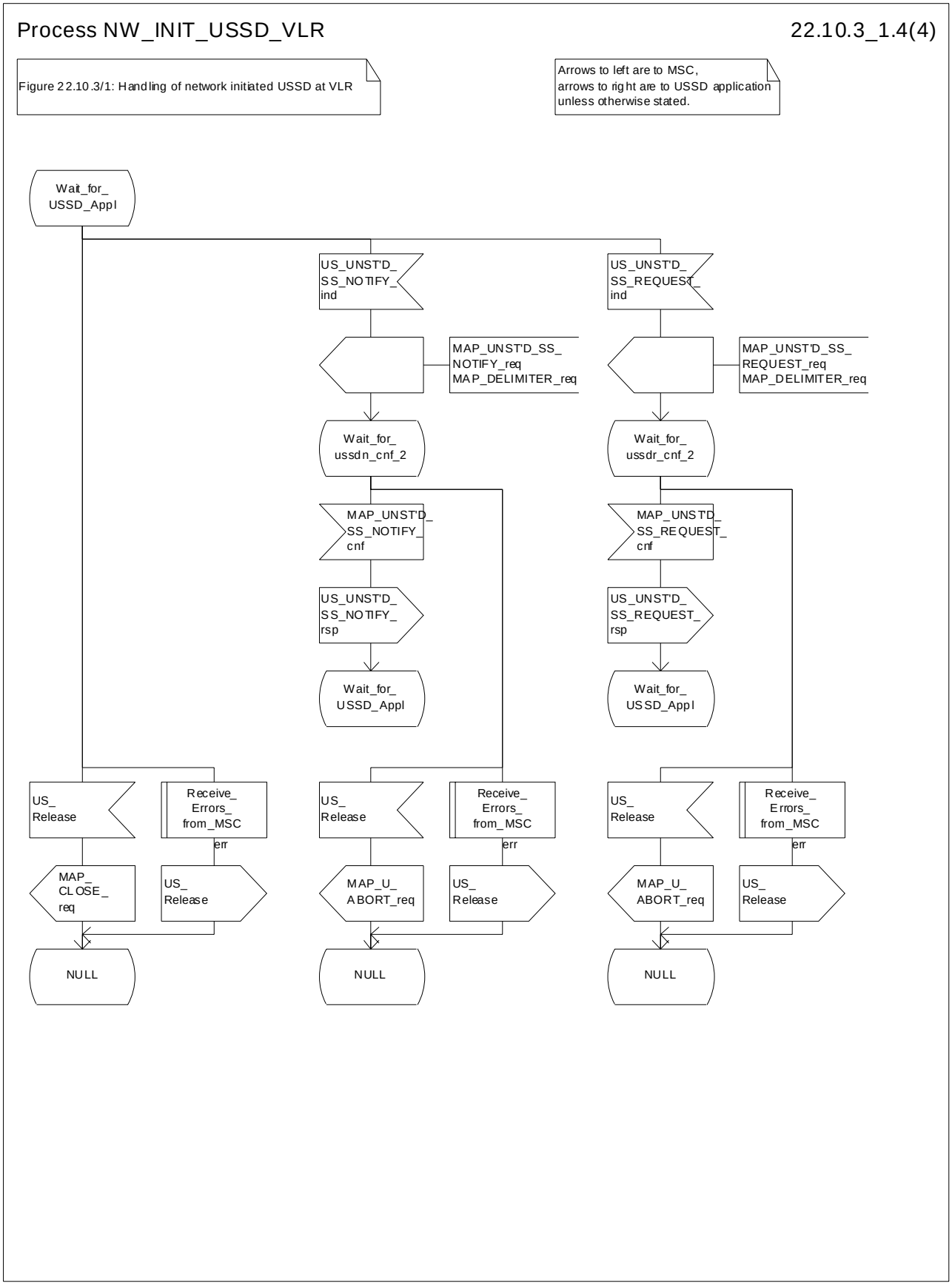


Figure 22.10.3/1 (sheet 4 of 4): Procedure NI_USSD_VLR

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.

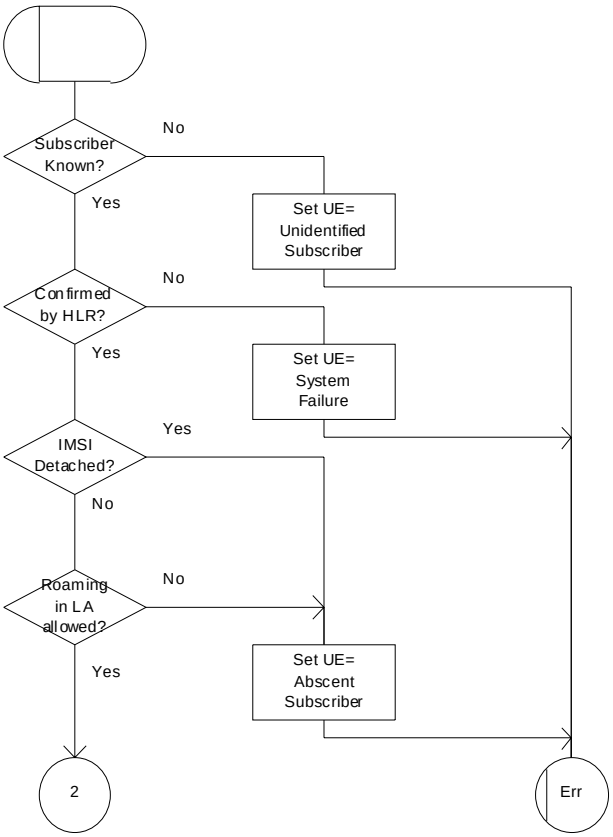


Figure 22.10.3/2 (sheet 1 of 2): Macro Start_USSD_VLR

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.

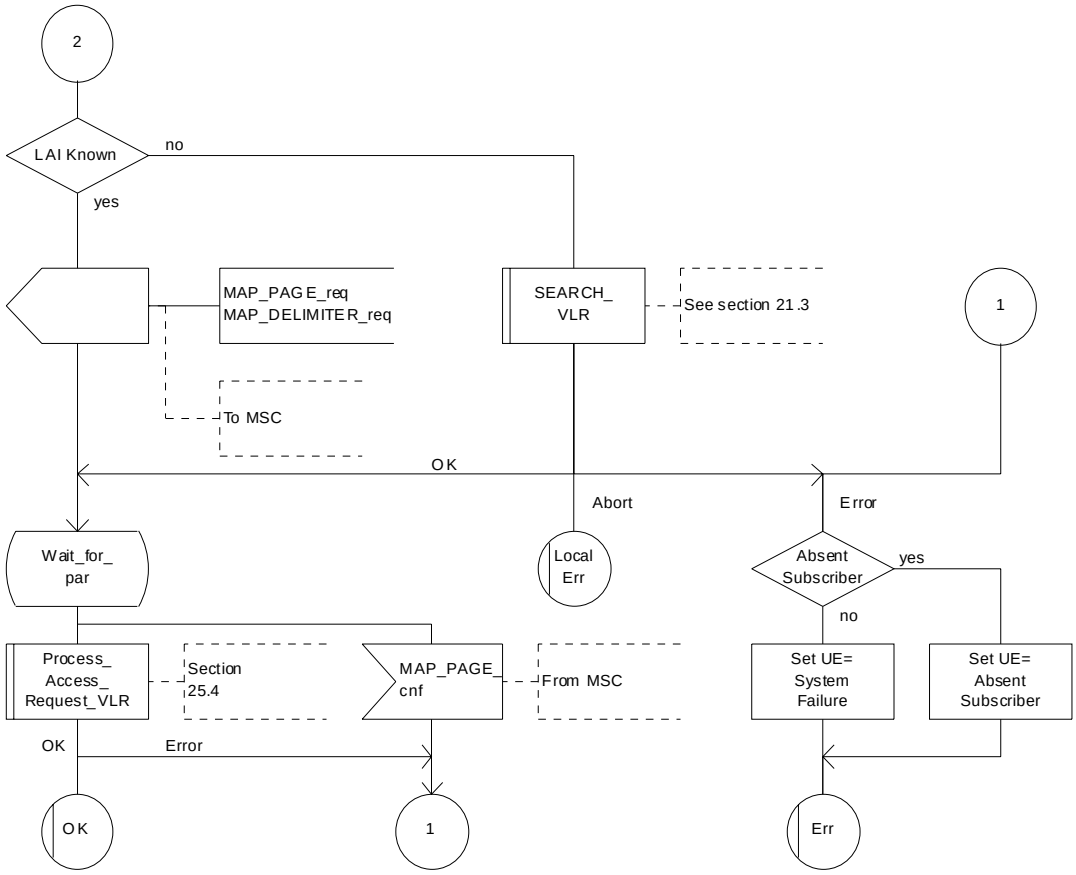


Figure 22.10.3/2 (sheet 2 of 2): Macro Start_USSD_VLR

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 and send the message received from the gsmSCF or secondary HLR or USSD application 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 HLR is shown in figure 22.10.4/1 and 22.10.4/2.

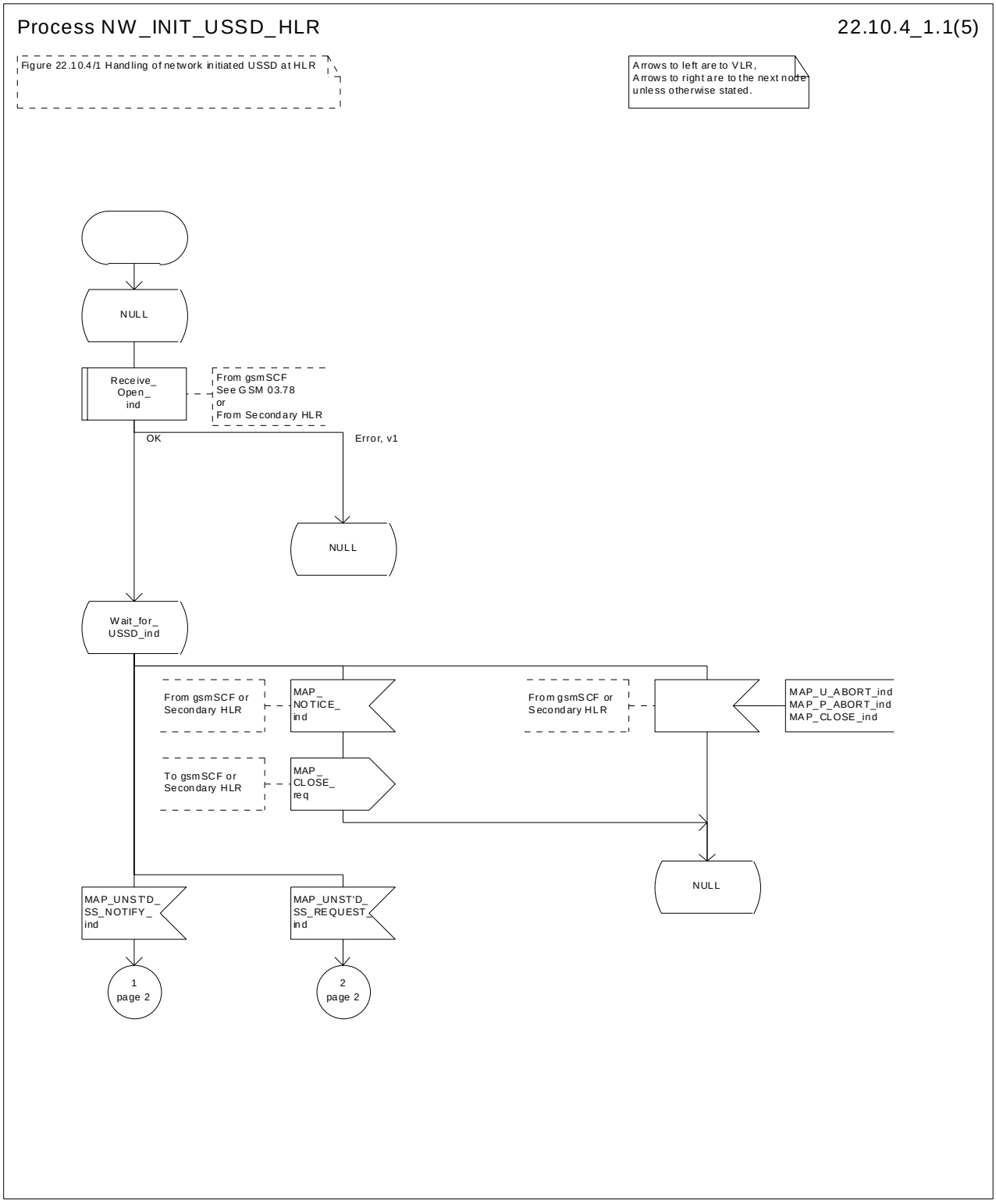


Figure 22.10.4/1 (sheet 1 of 5): Procedure NI_USSD_HLR

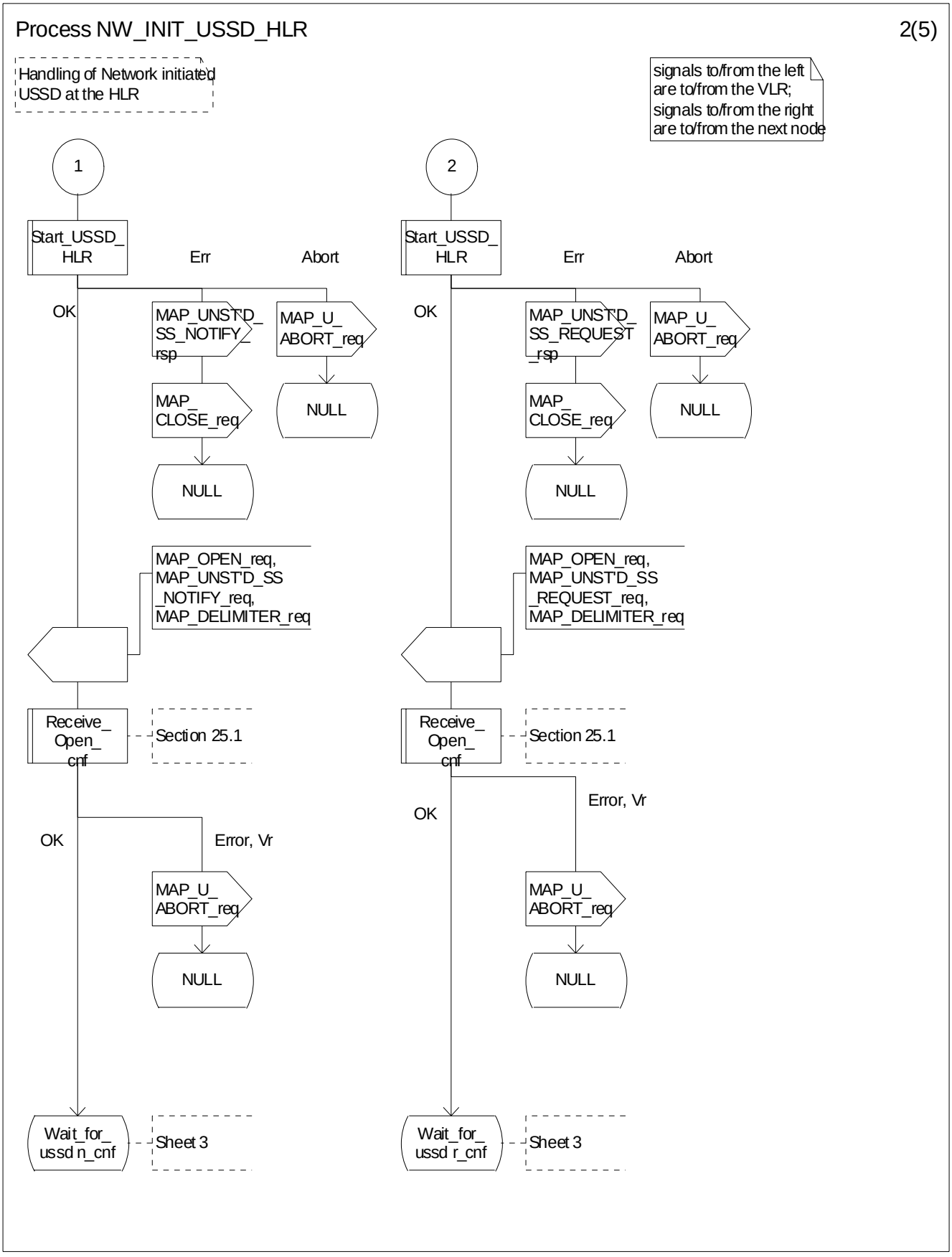


Figure 22.10.4/1 (sheet 2 of 5): Procedure NI_USSD_HLR

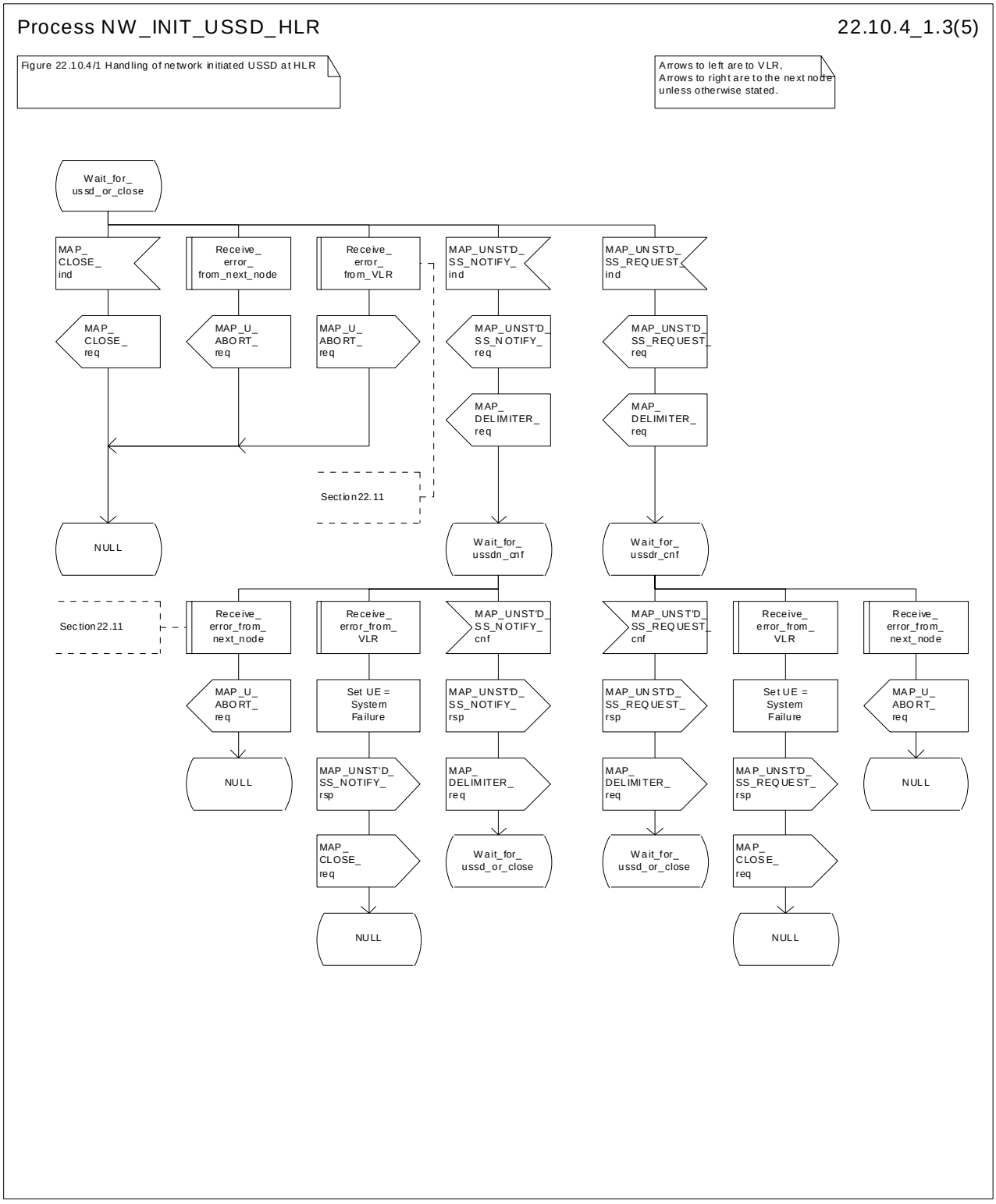


Figure 22.10.4/1 (sheet 3 of 5): Procedure NI_USSD_HLR

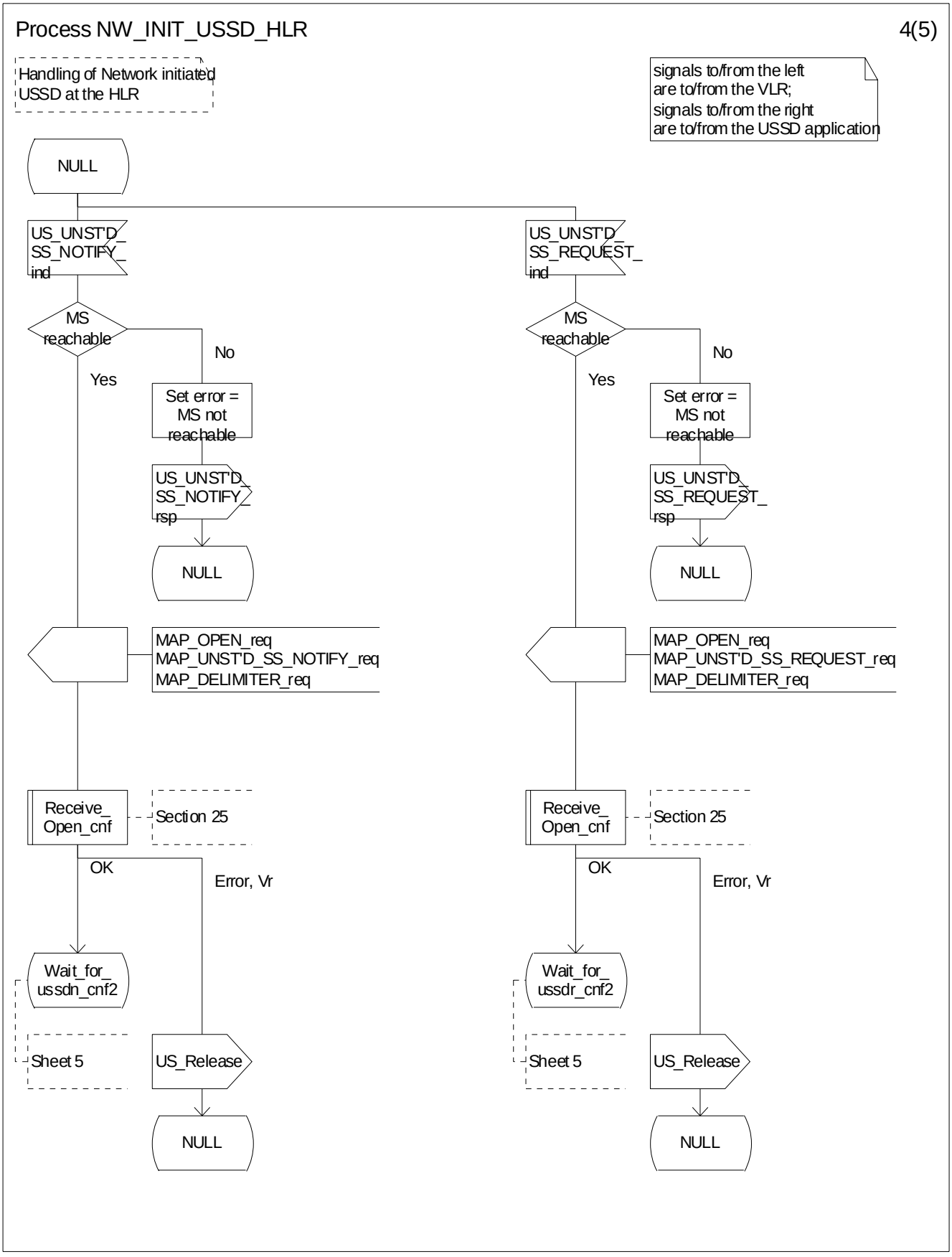


Figure 22.10.4/1 (sheet 4 of 5): Procedure NI_USSD_HLR

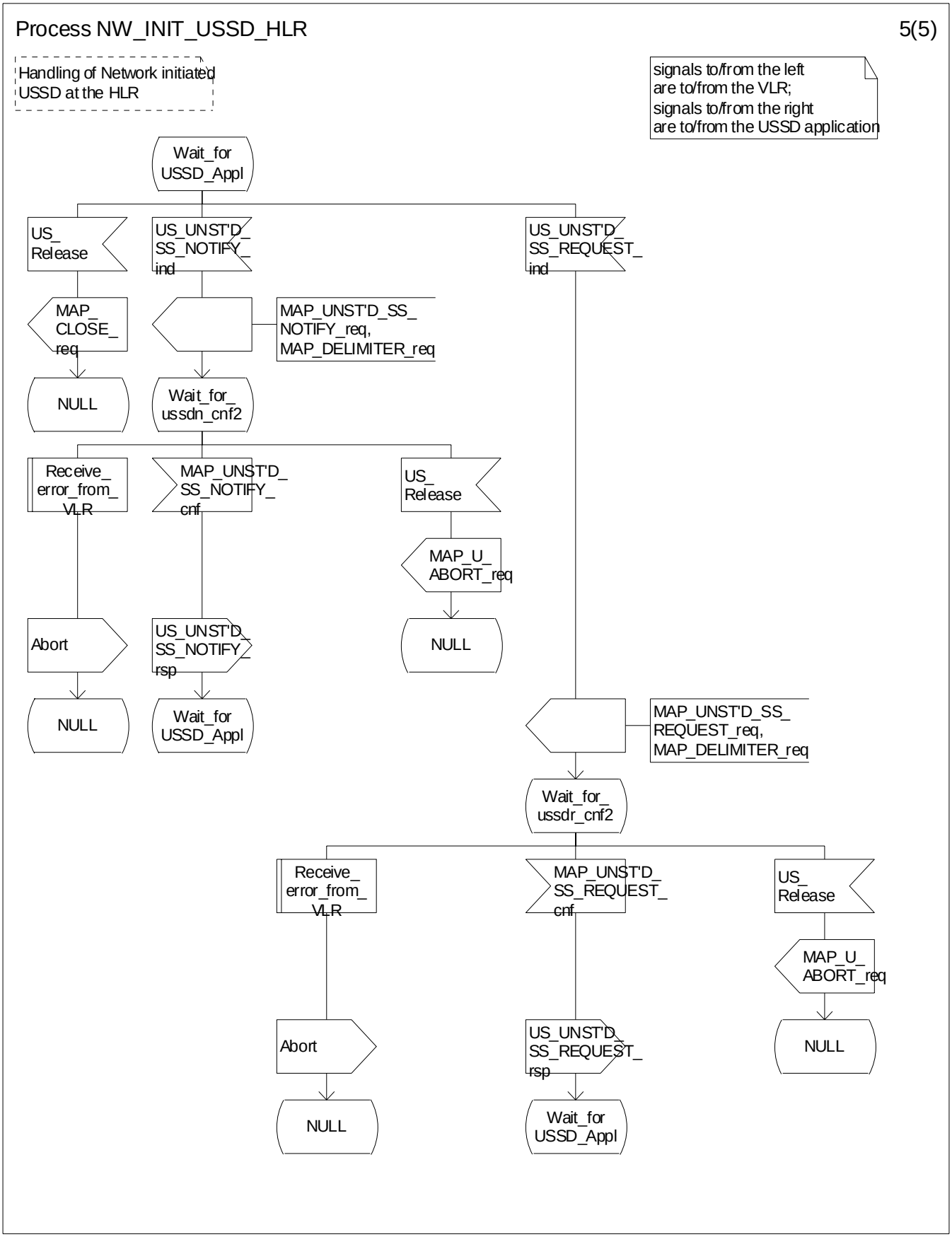


Figure 22.10.4/1 (sheet 5 of 5): Procedure NI_USSD_HLR

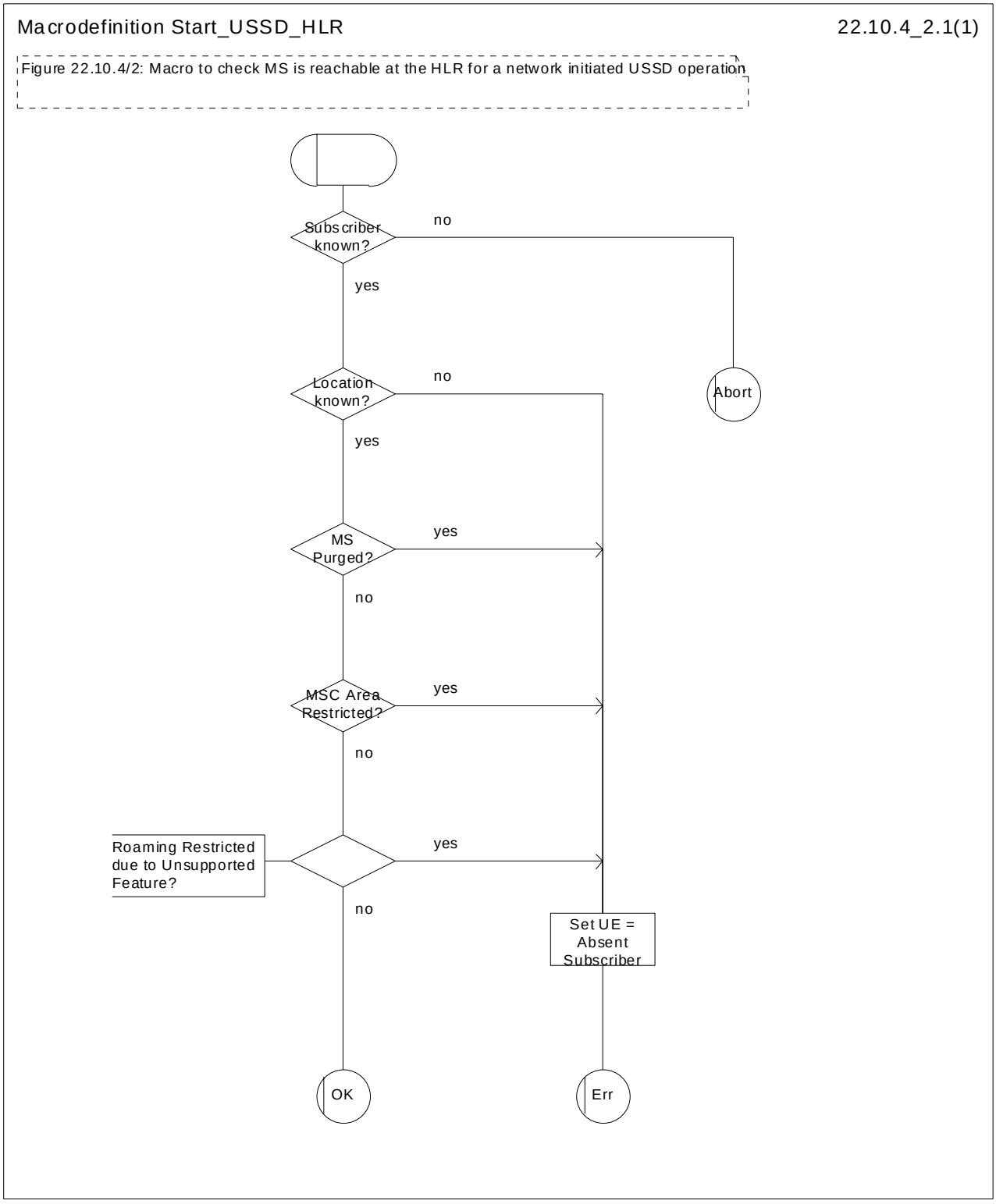


Figure 22.10.4/2: Macro Start_USSD_HLR

22.10.5 Procedure in the gsmSCF and secondary HLR

The procedure is invoked by an USSD application local to the gsmSCF/secondary HLR. It may start by using either the MAP_UNSTRUCTURED_SS_REQUEST or MAP_UNSTRUCTURED_SS_NOTIFY service.

In both cases the gsmSCF will initiate a MAP dialogue with the HLR and send the message received from the USSD application to the HLR.

Following transfer of the message the gsmSCF will wait for a confirmation from the HLR. This will be relayed to the USSD application..

Following this, the gsmSCF/secondary HLR may receive further UNSTRUCTURED_SS_REQUEST or UNSTRUCTURED_SS_NOTIFY requests, or may receive a Release from the USSD application.

In the event of an error, the MAP dialogue with the HLR shall be released as shown in the diagram.

The procedure in the gsmSCF and secondary HLR is shown in figure 22.10.5/1.

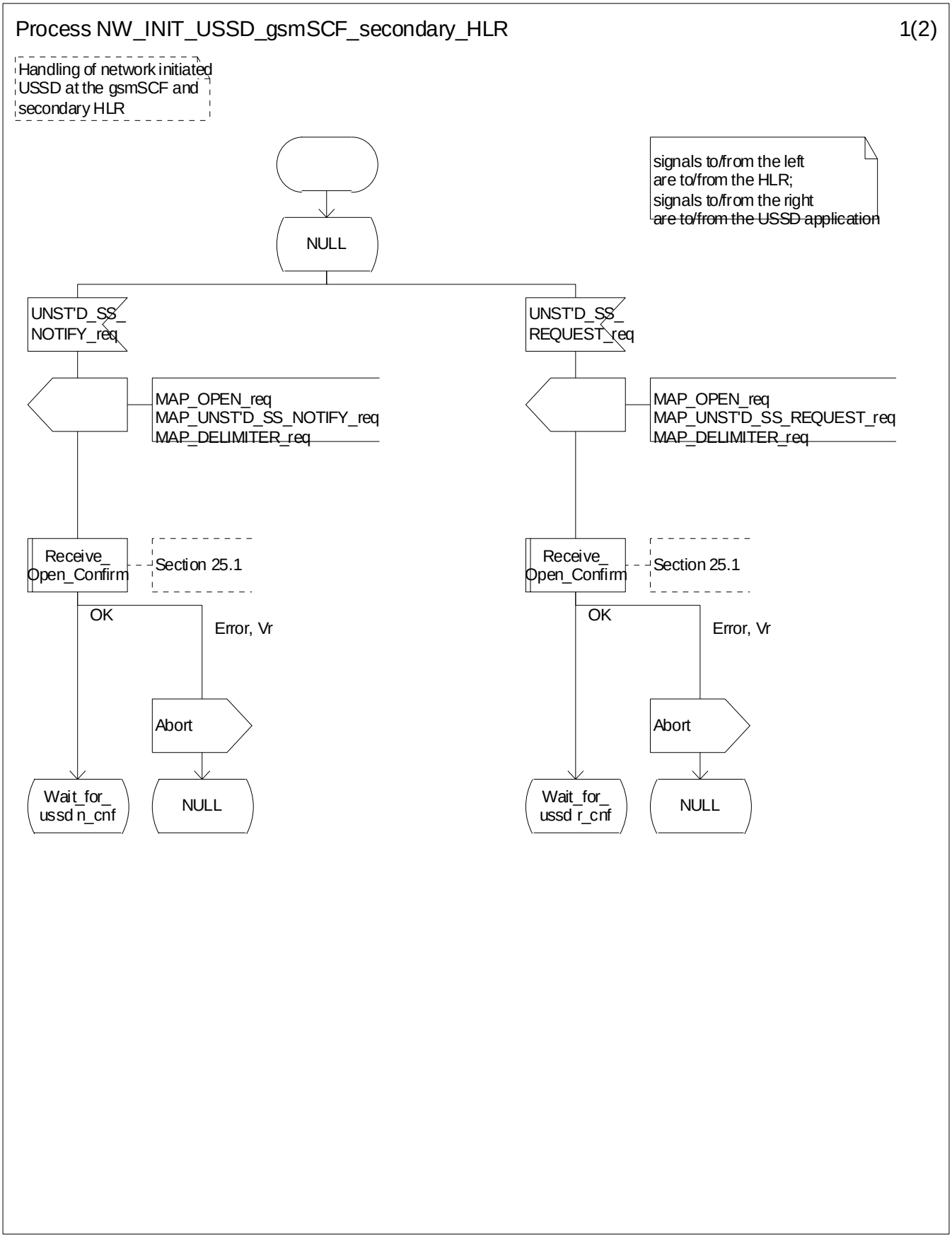


Figure 22.10.5/1 (sheet 1 of 2): Procedure NI_USSD_gsmSCF_secondary_HLR

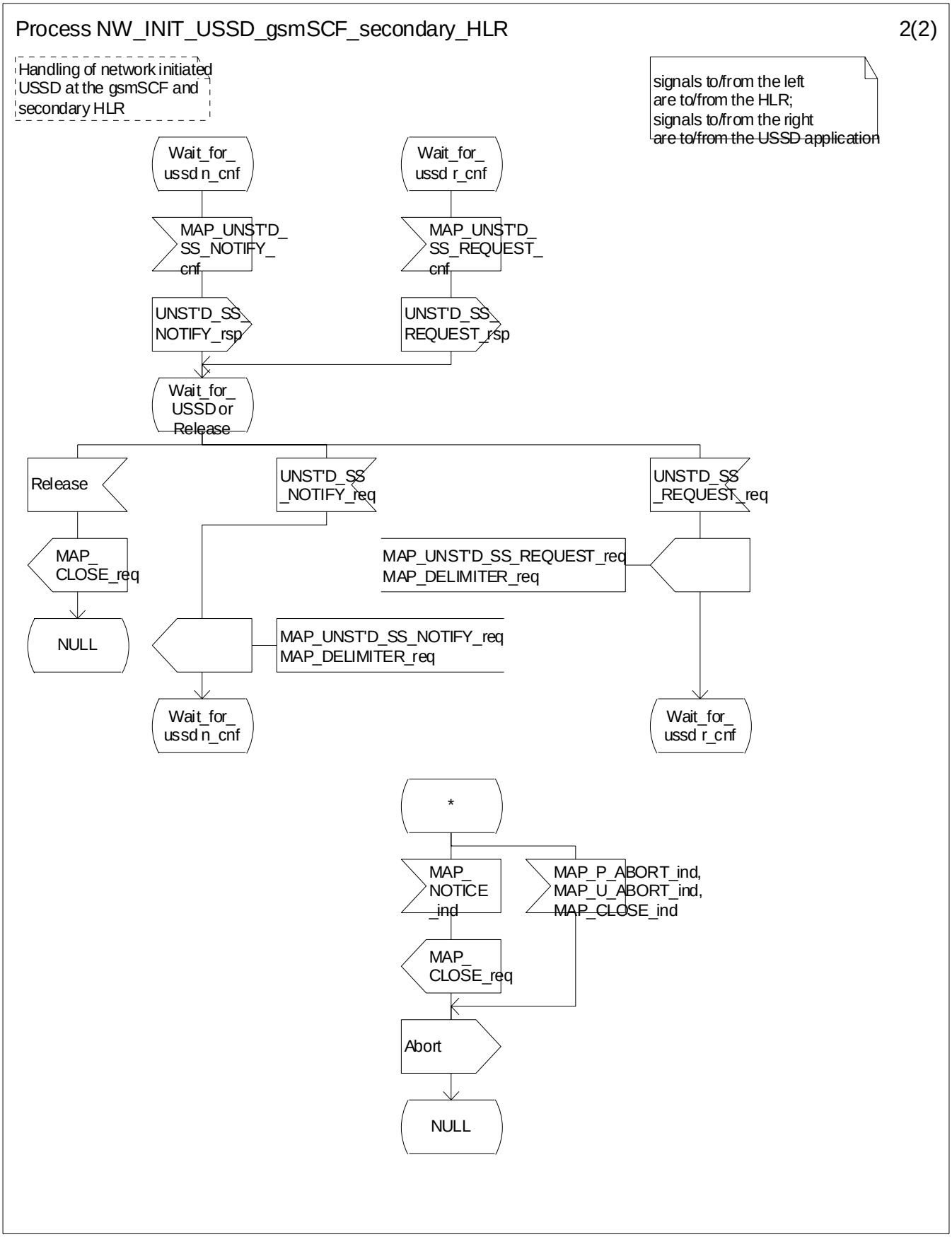


Figure 22.10.5/1 (sheet 2 of 2): Procedure NI_USSD_gsmSCF_secondary_HLR

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.

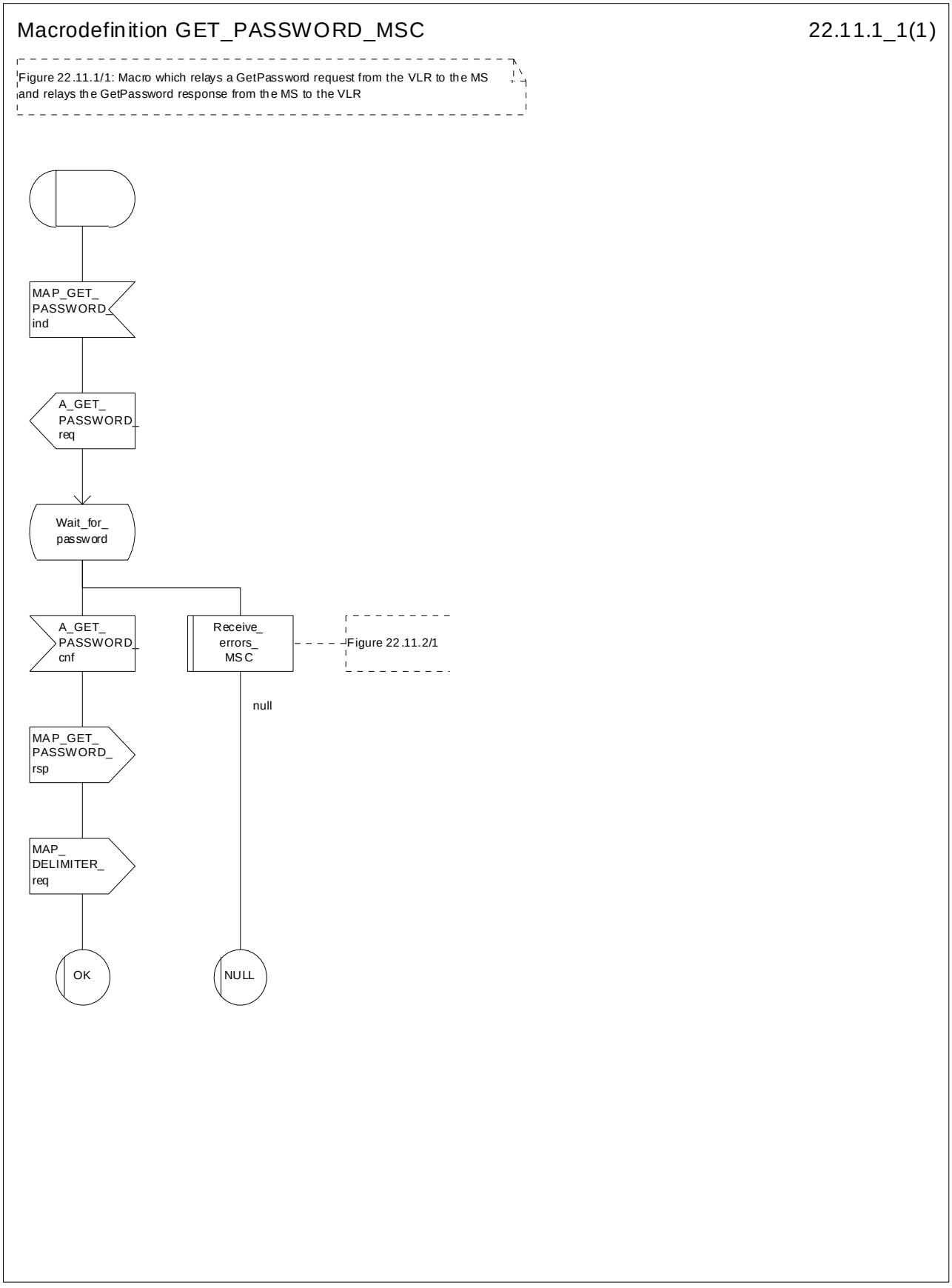


Figure 22.11.1/1: Macro Get_PW_MSC

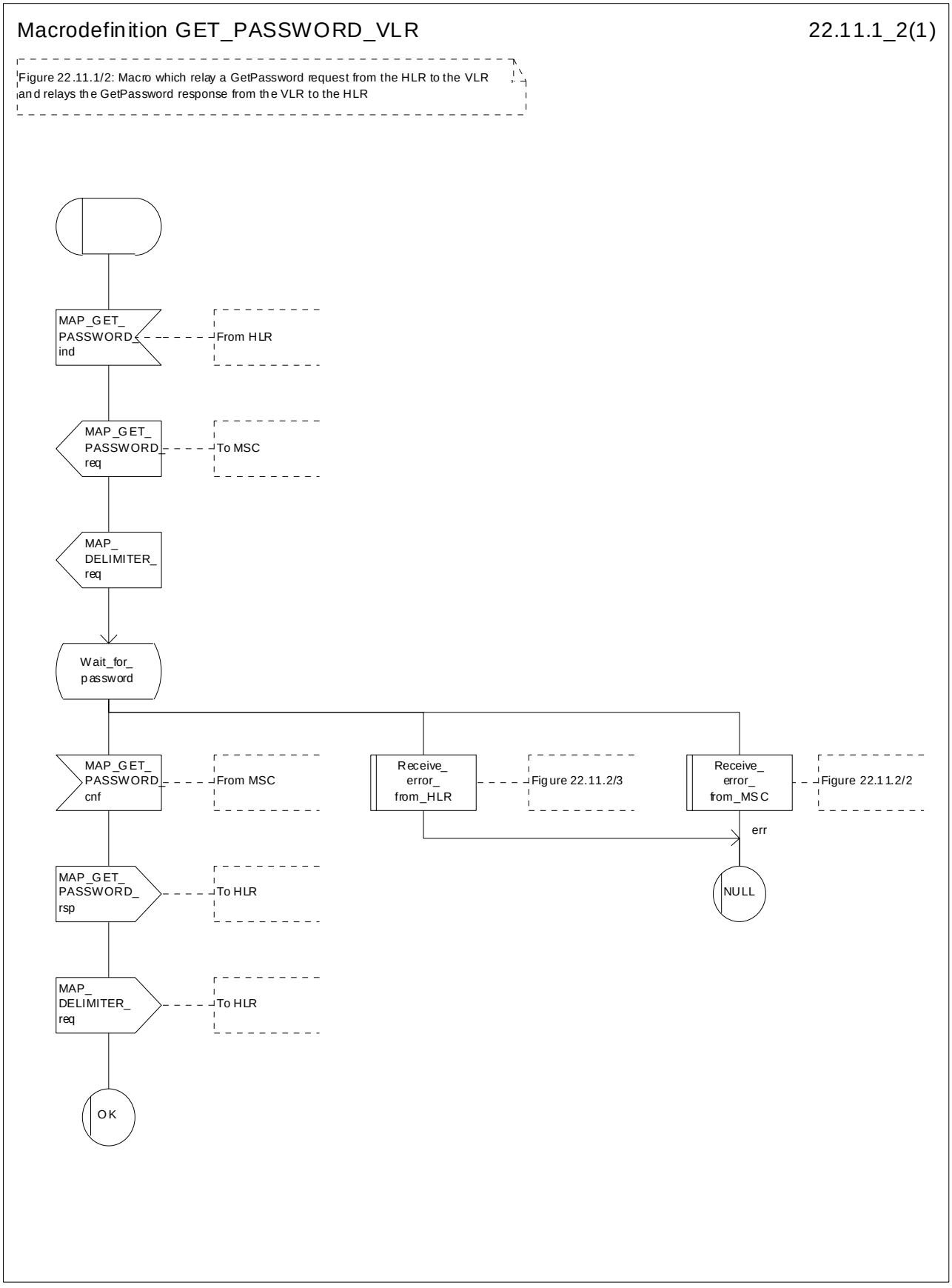


Figure 22.11.1/2: Macro Get_PW_VLR

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

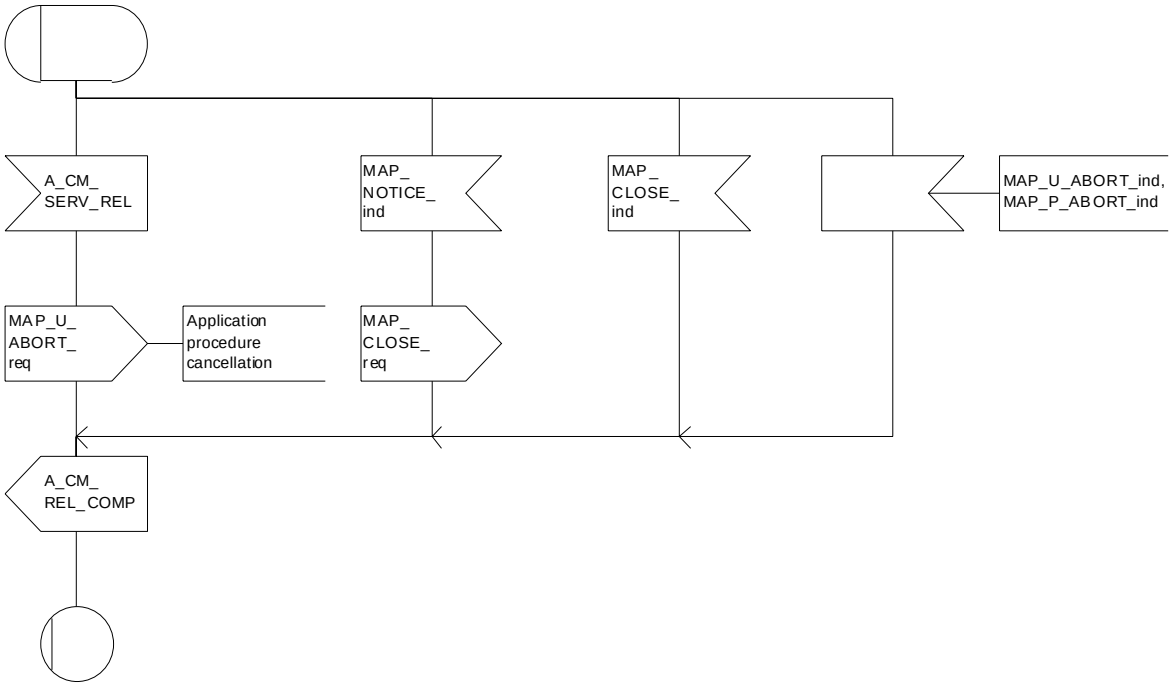


Figure 22.11.2/1: Macro Receive_Errors_MSC

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

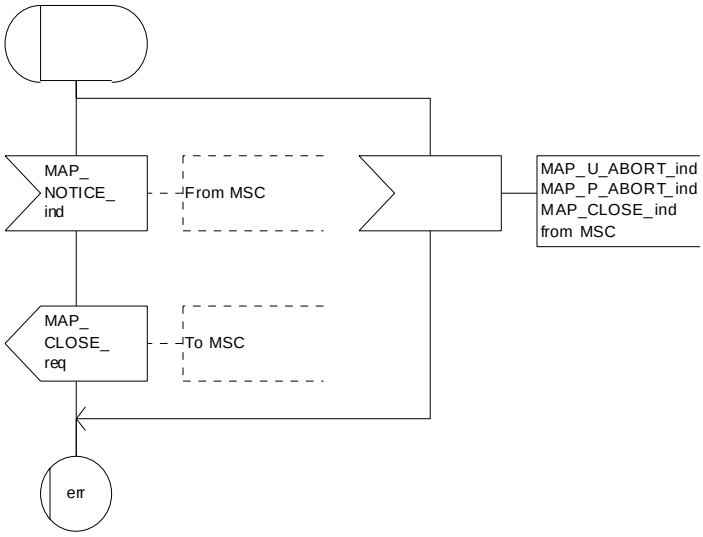


Figure 22.11.2/2: Macro Receive_Error_from_MSC

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

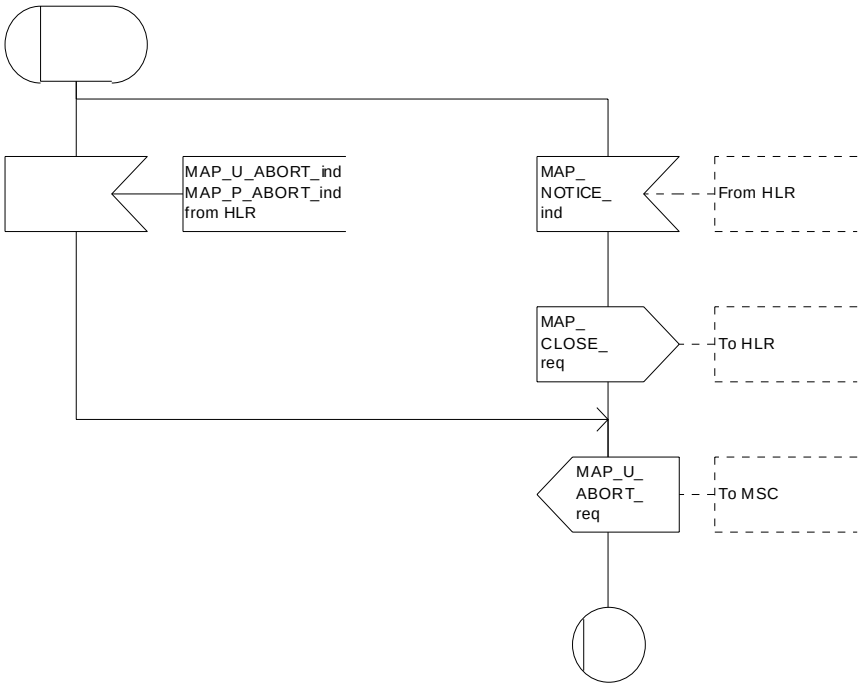


Figure 22.11.2/3: Macro Receive_Errors_HLR

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

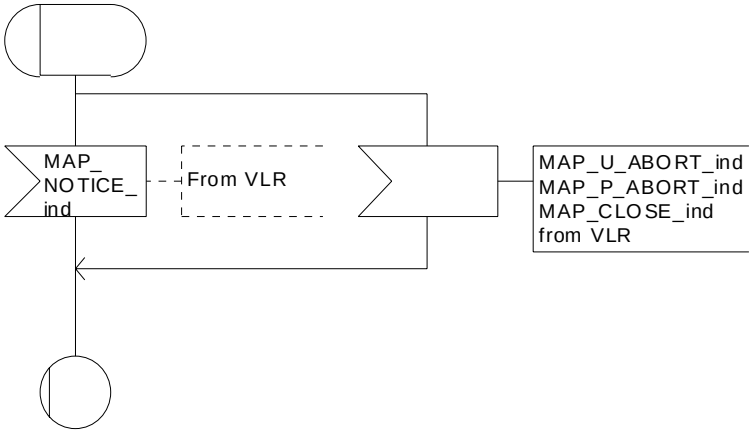


Figure 22.11.2/4: Macro Receive_error_from_VLR

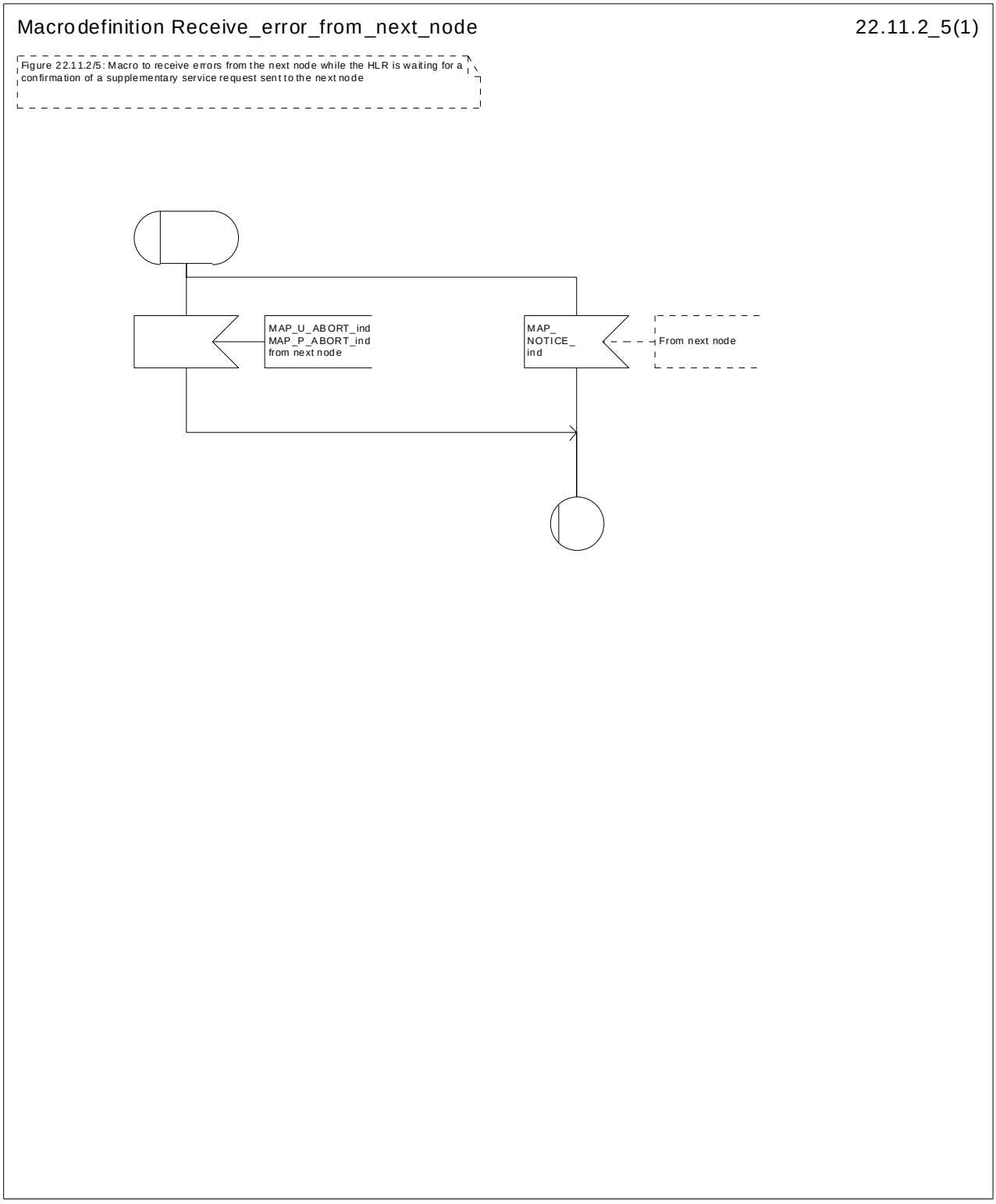


Figure 22.11.2/5: Macro Receive_error_from_next_node

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:

Figure 22.12.1/1: Interfaces and services for supplementary service invocation notification

(1) MAP-SUPPLEMENTARY-SERVICE-INVOCATION-NOTIFICATION (MSC to gsmSCF).

(2) MAP-SUPPLEMENTARY-SERVICE-INVOCATION-NOTIFICATION-ACK (gsmSCF to MSC).

MAP-SUPPLEMENTARY-SERVICE-INVOCATION-NOTIFICATION (defined in clauses 8 and 25).

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 via 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 of or receipt of a TCAP P-ABORT primitive. This is illustrated in figure 22.12.2/1.

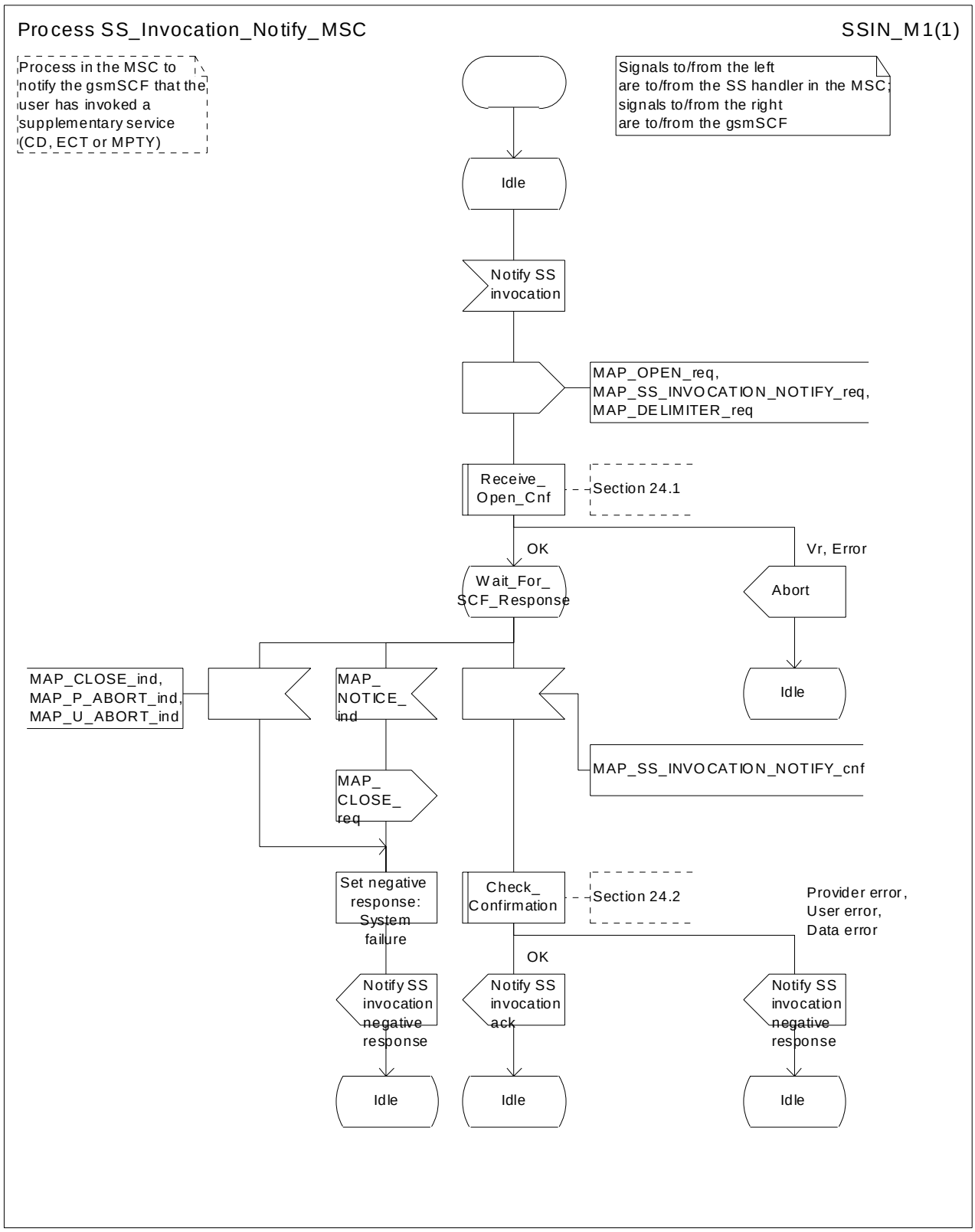


Figure 22.12.2/1: Process SS_Invocation_Notify_MSC (sheet 1 of 1)

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 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 22.12.3/1.

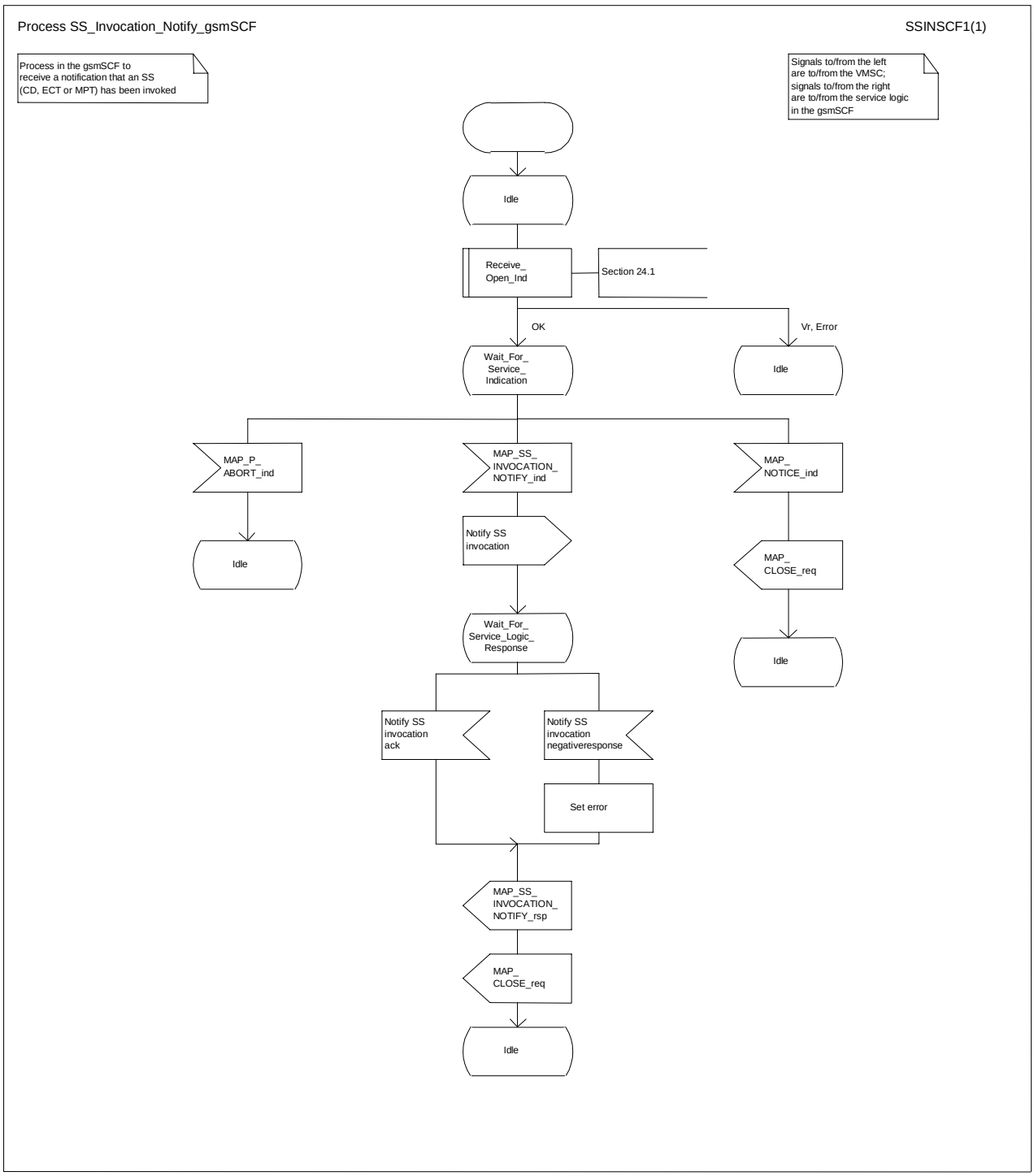


Figure 22.12.3/1: Process SS_Invocation_Notify_gsmSCF (sheet 1 of 1)

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.

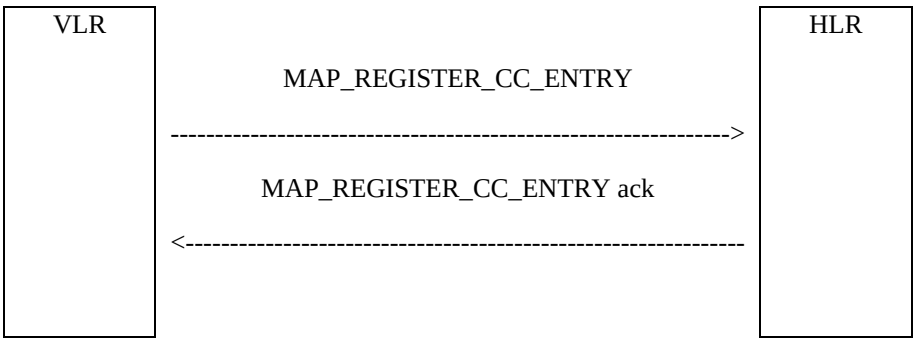


Figure 22.13.1/1: Message flow to activate a CCBS request

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 clause; the definitions of these macros can be found as follows:

Receive_Open_Cnf see clause 25.1.2;

Check_Confirmation see clause 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.

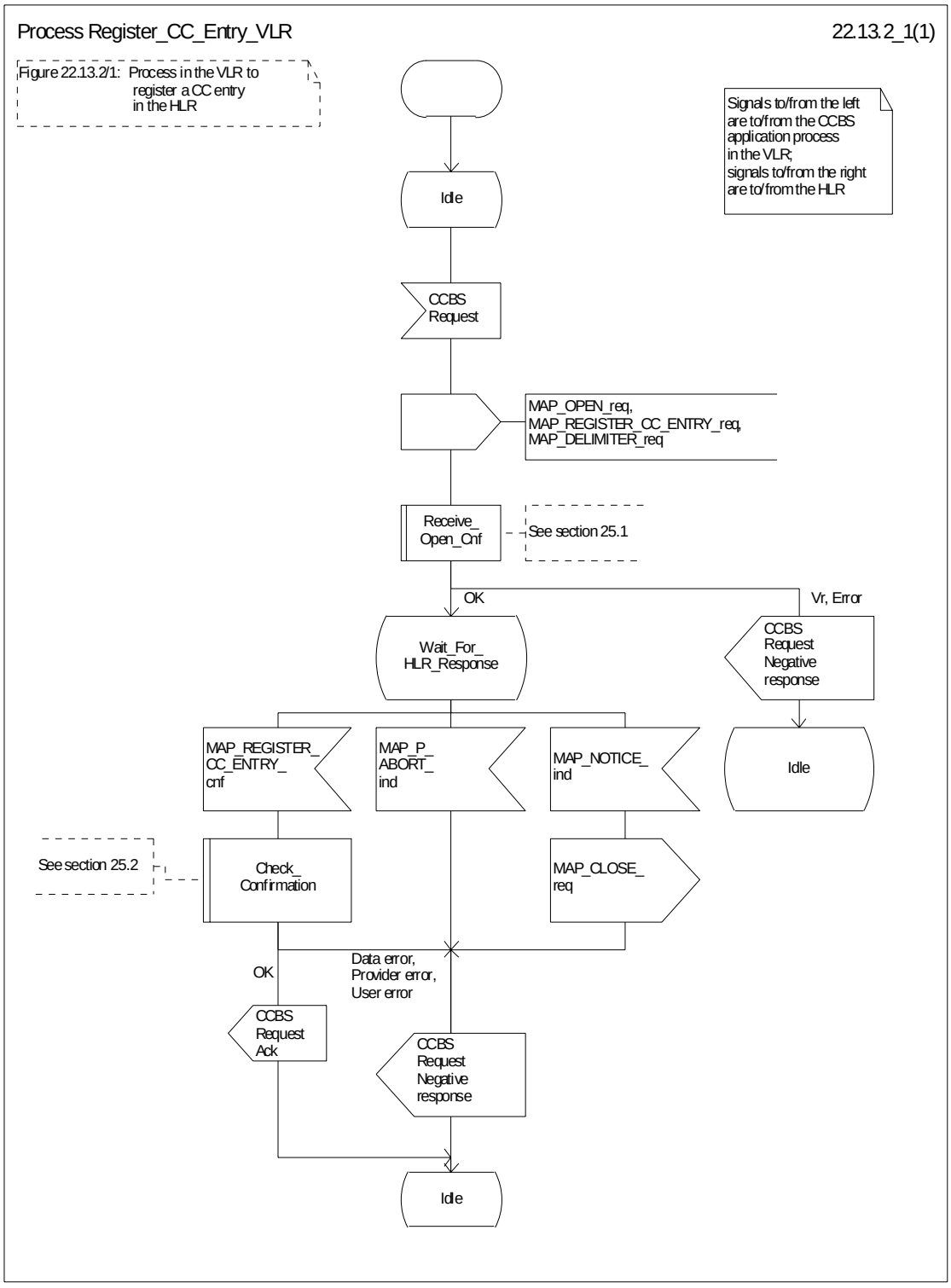


Figure 22.13.2/1: Process Register_CC_Entry_VLR

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.

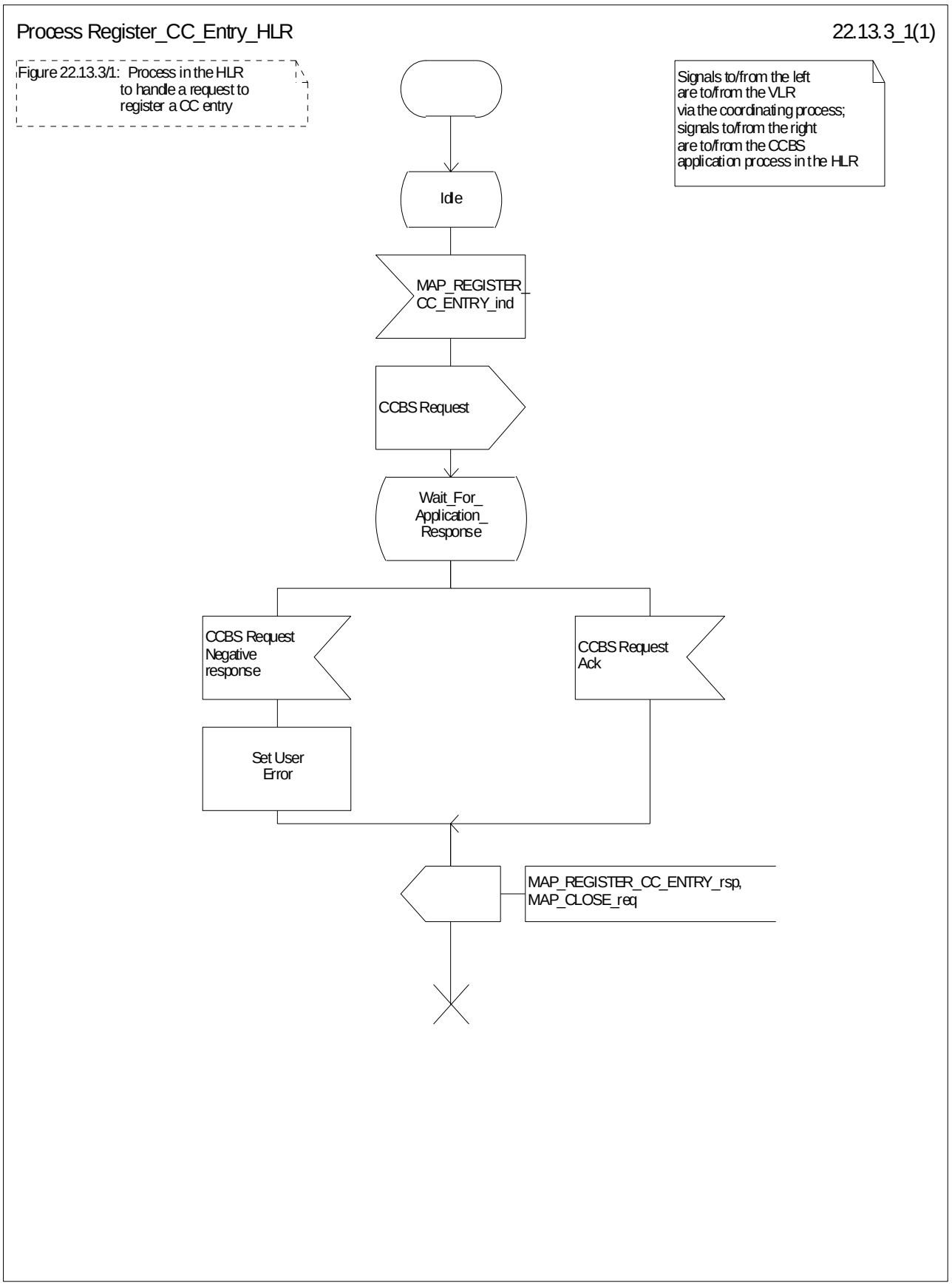


Figure 22.13.3/1: Process Register_CC_Entry_HLR

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.

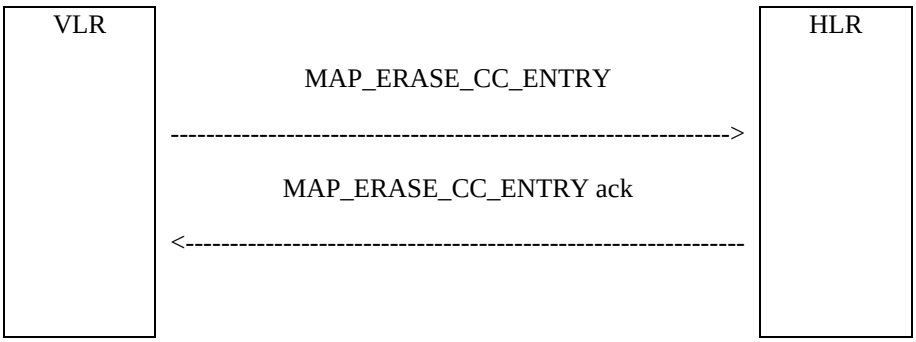


Figure 22.14.1/1: Message flow to deactivate a CCBS request

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 clause; the definitions of these macros can be found as follows:

Receive_Open_Cnf see clause 25.1.2;

Check_Confirmation see clause 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.

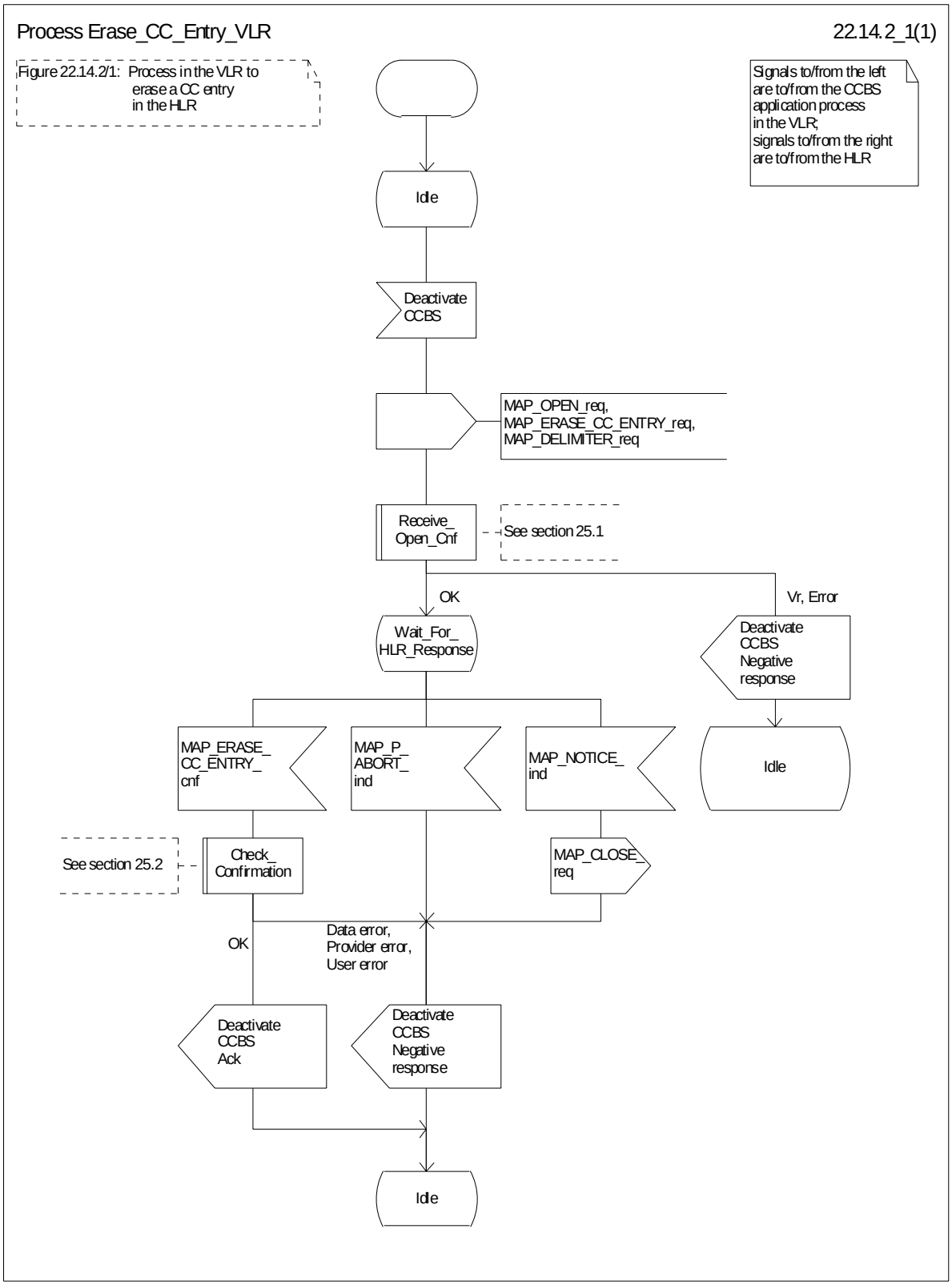


Figure 22.14.2/1: Process Erase_CC_Entry_VLR

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.

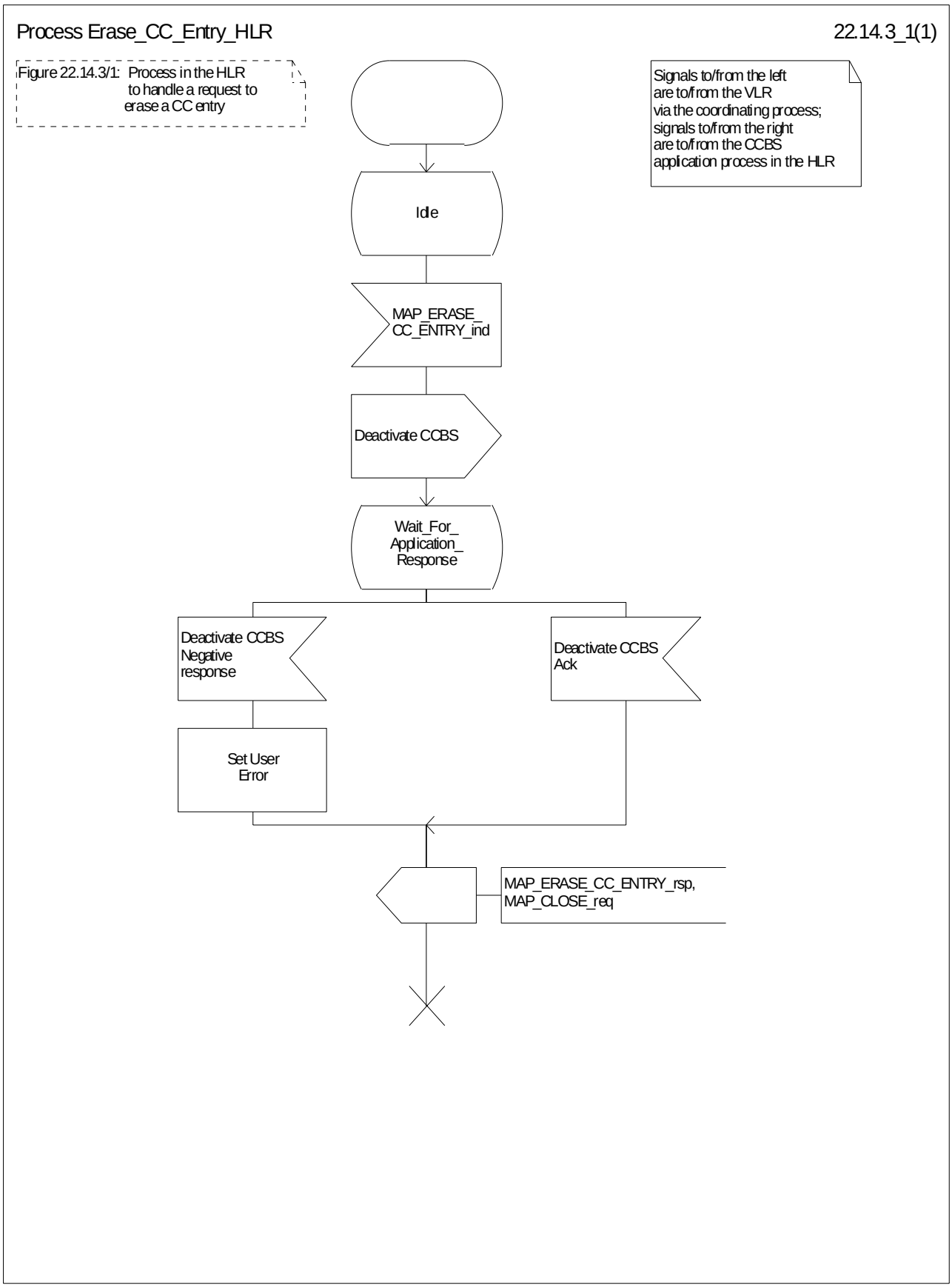


Figure 22.14.3/1: Process Erase_CC_Entry_HLR

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 (3GPP TS 24.008) 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 clause 23.2.1);
- if the A_RP_SM_MEMORY_AVAILABLE indication is received, the process SC_Alert_MSC is initiated (see clause 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.

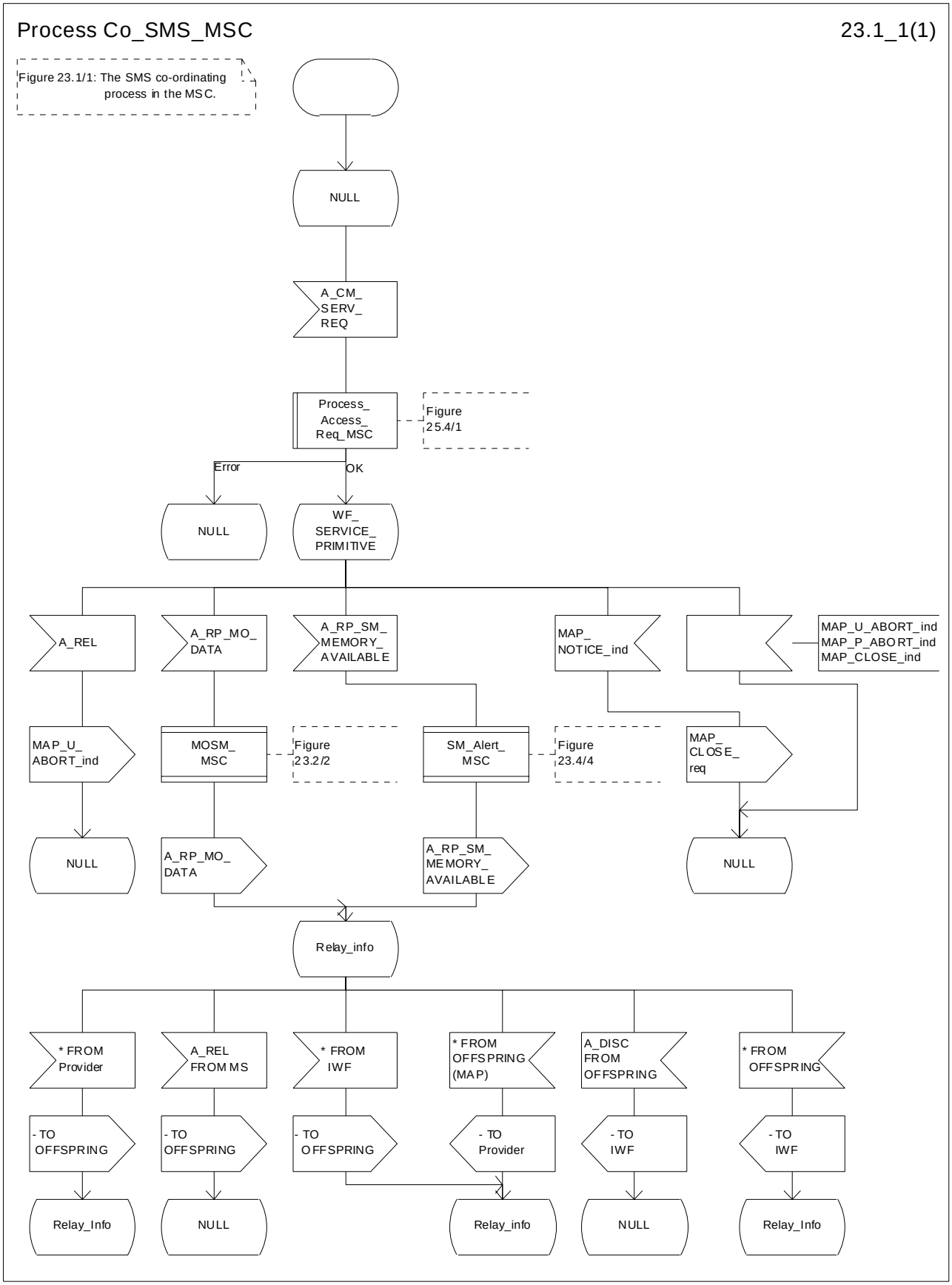


Figure 23.1/1: Process Co_SMS_MSC

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_SM_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.

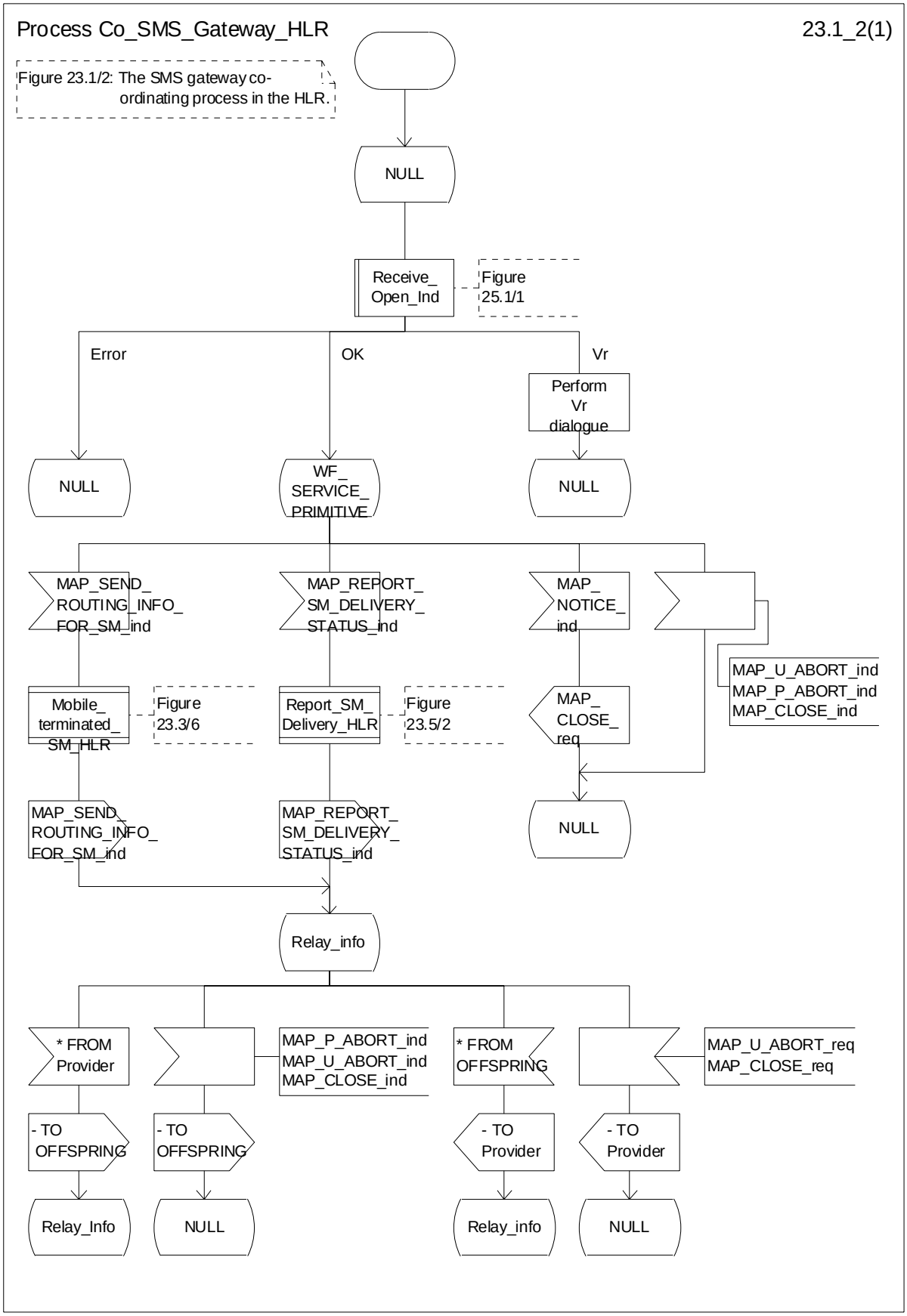


Figure 23.1/2: Process Co_SMS_Gateway_HLR

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 clause 23.2.4);
- if the A_RP_SM_MEMORY_AVAILABLE indication is received, the process SC_Alert_SGSN is initiated (see clause 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.

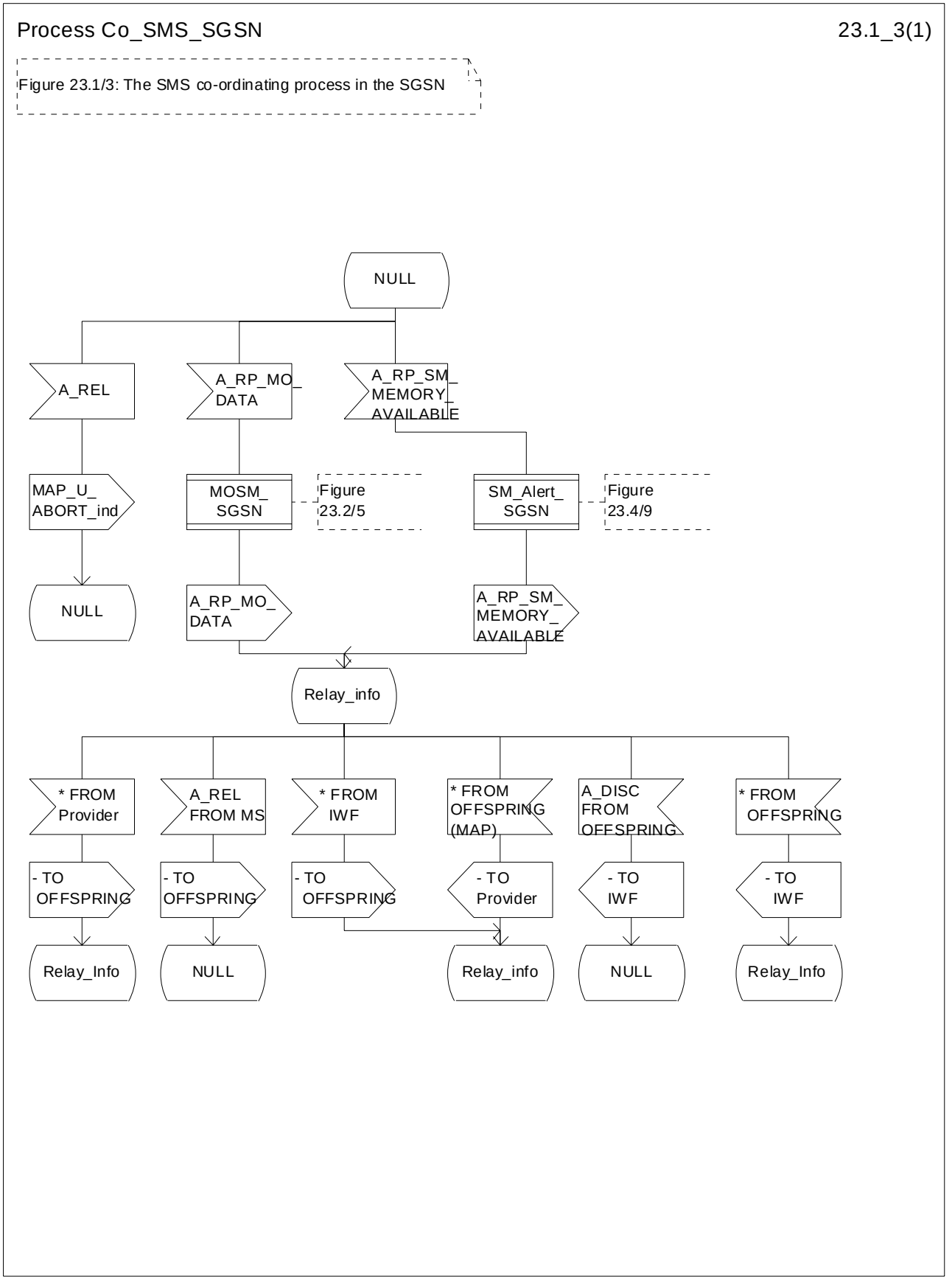


Figure 23.1/3: Process Co_SMS_SGSN

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.

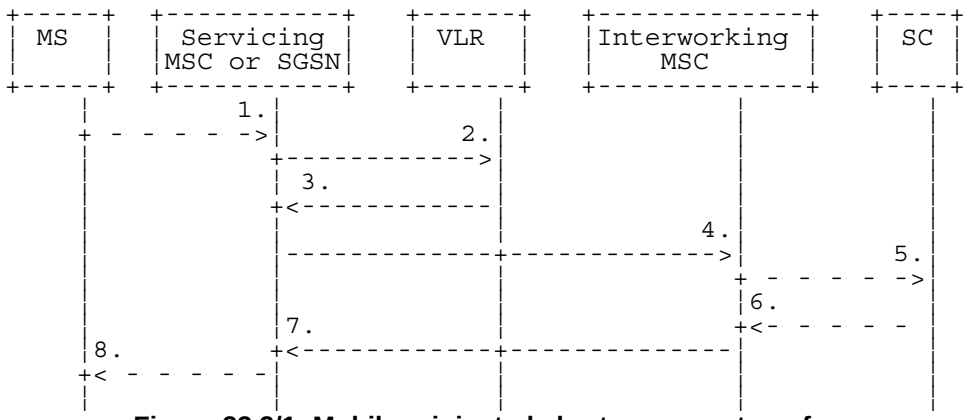


Figure 23.2/1: Mobile originated short message transfer

1) Short Message (3GPP TS 24.011 [37]).

2) MAP_SEND_INFO_FOR_MO_SMS (*).

3) MAP_SEND_INFO_FOR_MO_SMS_ACK (*).

4) MAP_MO_FORWARD_SHORT_MESSAGE.

5) Short message (3GPP TS 23.040[26]).

6) Short message Acknowledgement (3GPP TS 23.040[26]).

7) MAP_MO_FORWARD_SHORT_MESSAGE_ACK.

8) Short Message Acknowledgement (3GPP TS 24.011 [37]).

(*) Messages 2) and 3) are not used by SGSN.

In addition the following MAP services are used:

MAP_PROCESS_ACCESS_REQUEST	(see clause 8.3); (*)
MAP_AUTHENTICATE	(see clause 8.5); (*)
MAP_SET_CIPHERING_MODE	(see clause 8.6); (*)
MAP_PROVIDE_IMSI	(see clause 8.9); (*)
MAP_CHECK_IMEI	(see clause 8.7);
MAP_FORWARD_NEW_TMSI	(see clause 8.9); (*)
MAP_TRACE_SUBSCRIBER_ACTIVITY	(see clause 9.1); (*)
MAP_READY_FOR_SM	(see clause 12.4).

(*) Those messages are not used by SGSN.

23.2.1 Procedure in the serving MSC

Any CAMEL-specific handling defined in this subclause is omitted if the MSC does not support CAMEL control of MO SMS or if the subscriber does not have a subscription for CAMEL control of MO SMS .

When the MSC receives the short message from the A-interface, it sends a MAP_SEND_INFO_FOR_MO_SMS request to the VLR and waits for a response. Whilethe MSC is waiting for a response from the VLR:

- if the VLR aborts, or prematurely closes, the dialogue, the MSC reports to the gsmSCF that the short message submission has failed and sends an A_RP_ERROR with error cause "Network out of order" to the MS, and the process terminates;
- if it receives a MAP_CONTINUE_CAMEL_SMS_HANDLING indication, it checks the indication.
  - if the indication is badly formed, the MSC sends an A_RP_ERROR with error cause "Network out of order" to the MS and aborts the dialogue with the VLR, and the process terminates;
  - if the indication is OK, the MSC calls the procedure CAMEL_O_SMS_INIT and tests the result.
    - if the result was "SMS_Aborted", the MSC aborts the dialogue with the VLR, and the process terminates;
    - if the result was "Release_SMS", the MSC returns an A_RP_ERROR, with an error cause as instructed by the gsmSCF, to the MS and aborts the dialogue with the VLR, and the process terminates;
    - if the result was "Redirect SMS", the MSC modifies the data for the submitted short message as instructed by the gsmSCF, sends to the VLR a MAP_SEND_INFO_FOR_MO_SMS request and waits for a response;
    - if the result was "Continue", the MSC sends to the VLR a MAP_SEND_INFO_FOR_MO_SMS request and waits for a response. The handling for this request is shown in the procedure CAMEL_MO_SMS_VLR (see 3GPP TS 23.078 [98]).
- if it receives a MAP_SEND_INFO_FOR_MO_SMS confirmation from VLR, it checks the confirmation.
  - if the confirmation includes an error, the MSC reports to the gsmSCF that the short message submission has failed and sends an A_RP_ERROR with the appropriate error cause to the MS, and the process terminates;
  - if the confirmation indicates a successful result, the MSC checks whether the MSC is also the SMS-IWMSC.
    - if the MSC is separate from the SMS-IWMSC, MSC handling continues as described below under the heading "Serving MSC is separate from SMS-IWMSC".
    - if the MSC is also the SMS-IWMSC, the MSC handling continues as described below under the heading "Serving MSC is SMS-IWMSC";

Serving MSC is separate from SMS-IWMSC

The MSC checks whether the MAP_OPEN request and the MAP_MO_FORWARD_SHORT_MESSAGE request can be sent in a single message signal unit through the lower layers of the protocol.

- if the two requests can be grouped in a single TC message, the MSC requests a dialogue with the SMS-IWMSC, including the MAP_MO_FORWARD_SHORT_MESSAGE request;
  - if the dialogue opening is successful, the MSC waits for the response from the SMS-IWMSC;
  - if the macro Receive_Open_Cnf takes the "Error" exit, the MSC reports to the gsmSCF that the short message submission has failed and returns an A_RP_ERROR with cause "Network out of order" to the MS, and the process terminates;
  - if the macro Receive_Open_Cnf takes the "Vr" exit, the MSC handles the dialogue according to the specification for the earlier version of the protocol and checks the result.
    - if the submission was successful, the MSC reports to the gsmSCF that the short message submission was successful and returns an A_RP_ACK to the MS, and the process terminates;

- if the submission failed, the MSC reports to the gsmSCF that the short message submission has failed and returns an A_RP_ERROR with the appropriate error cause to the MS, and the process terminates.

- if the two requests cannot be grouped in a single TC message, the MSC requests a dialogue with the SMS-IWMSC, omitting the MAP_MO_FORWARD_SHORT_MESSAGE request;

  - if the dialogue opening is successful, the MSC sends a MAP_MO_FORWARD_SHORT_MESSAGE request to the SMS-IWMSC, and waits for the response from the SMS-IWMSC;

  - if the macro Receive_Open_Cnf takes the "Error" exit, the MSC reports to the gsmSCF that the short message submission has failed and returns an A_RP_ERROR with cause "Network out of order" to the MS, and the process terminates;

  - if the macro Receive_Open_Cnf takes the "Vr" exit, the MSC handles the dialogue according to the specification for the earlier version of the protocol, and checks the result.

    - if the submission was successful, the MSC reports to the gsmSCF that the short message submission was successful and returns an A_RP_ACK to the MS, and the process terminates;

    - if the submission failed, the MSC reports to the gsmSCF that the short message submission has failed and returns an A_RP_ERROR with the appropriate error cause to the MS, and the process terminates.

- if the MSC receives a MAP_MO_FORWARD_SHORT_MESSAGE confirmation from the SMS-IWMSC, it checks the content of the confirmation;

  - if the confirmation indicates that the submission of the short message was successful, the MSC reports to the gsmSCF that the short message submission was successful and returns an A_RP_ACK to the MS, and the process terminates;

  - if the confirmation indicates that the submission of the short message failed, the MSC reports to the gsmSCF that the short message submission has failed and returns an A_RP_ERROR with the appropriate error cause to the MS, and the process terminates;

- if the dialogue with the SMS-IWMSC fails, the MSC reports to the gsmSCF that the short message submission has failed and returns an A_RP_ERROR with cause "Network out of order" to the MS, and the process terminates.

Serving MSC is SMS-IWMSC

The MSC sends an SC_RP_MO_DATA request to the Short Message Service Centre (SMSC), and waits for the response.

- if the MSC receives an error response from the SMSC, it reports to the gsmSCF that the short message submission has failed and returns an A_RP_ERROR with the appropriate error cause to the MS, and the process terminates;

- if the SMSC aborts the dialogue, the MSC reports to the gsmSCF that the short message submission has failed and returns an A_RP_ERROR with cause "Network out of order" to the MS, and the process terminates;

- if the MSC receives a positive response from the SMSC, it reports to the gsmSCF that the short message submission was successful and returns an A_RP_ACK to the MS, and the process terminates.

The mobile originated short message service process in the MSC is shown in figure 23.2/2.

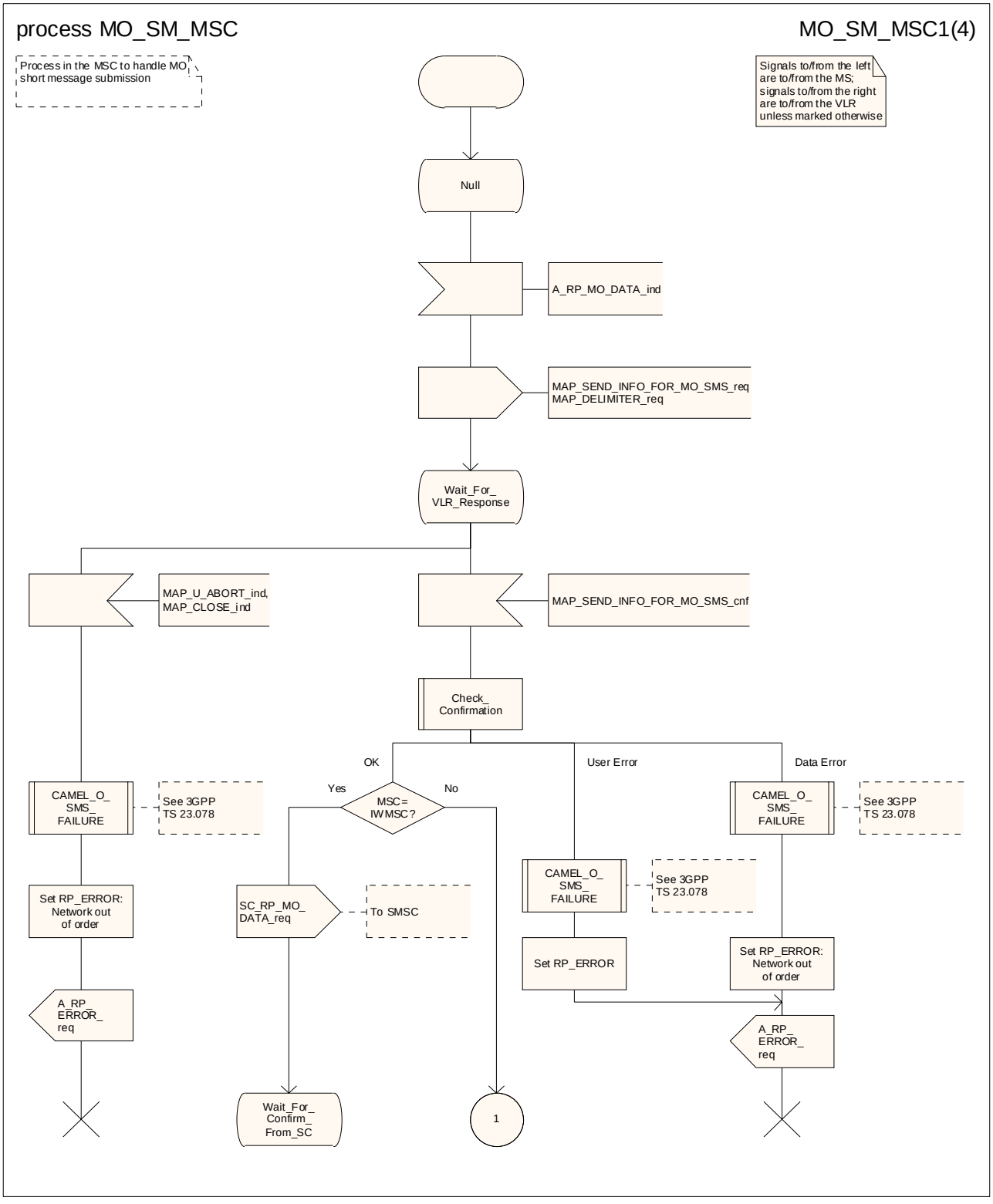


Figure 23.2/2 (sheet 1 of 4): Process MO_SM_MSC

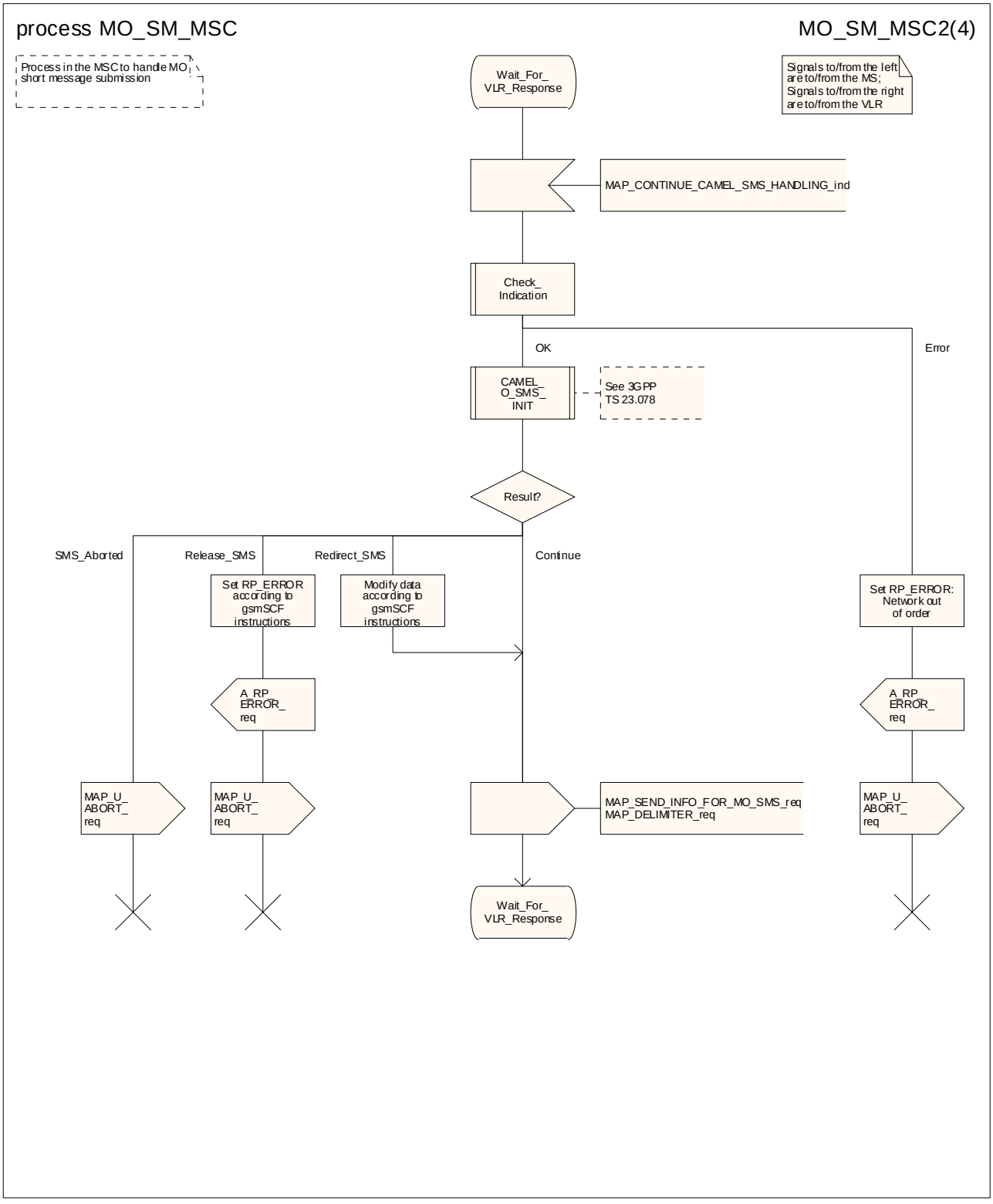


Figure 23.2/2 (sheet 2 of 4): Process MO_SM_MSC

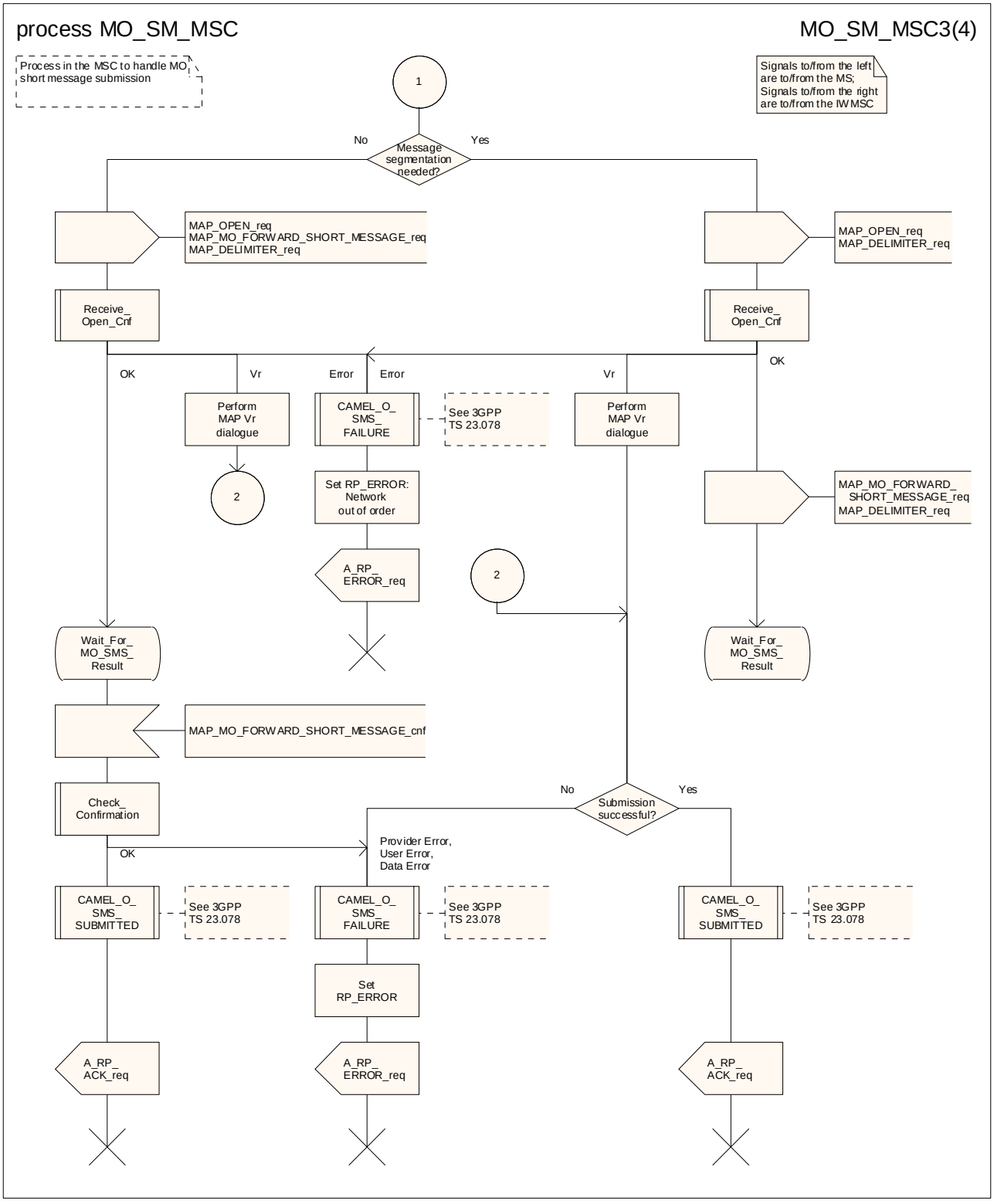


Figure 23.2/2 (sheet 3 of 4): Process MO_SM_MSC

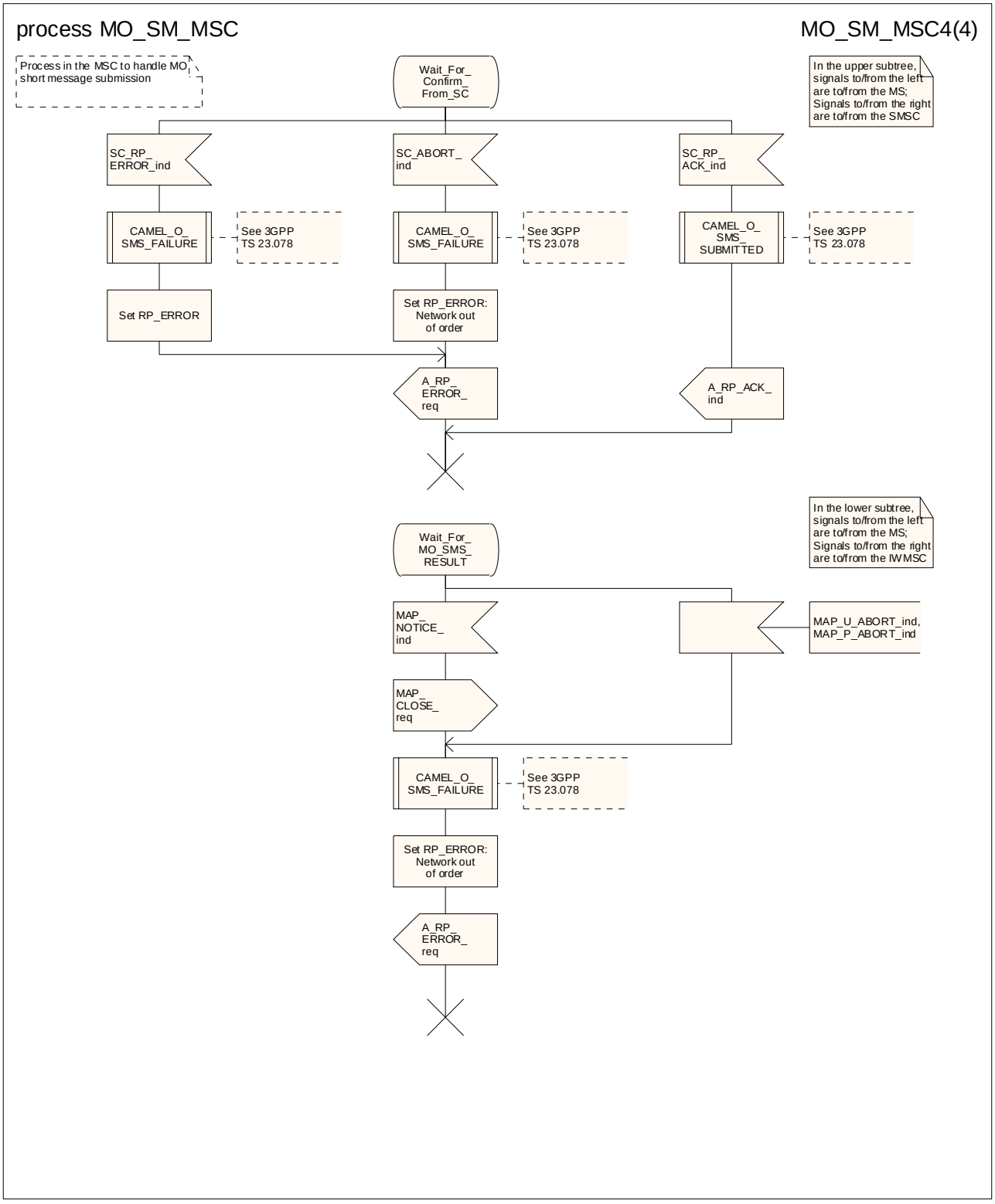


Figure 23.2/2 (sheet 4 of 4): Process MO_SM_MSC

23.2.2 Procedure in the VLR

Any CAMEL-specific handling defined in this subclause is omitted if the VLR does not support CAMEL control of MO SMS.

The process is triggered by a dialogue opening request followed by a MAP_PROCESS_ACCESS_REQUEST including a CM service type Short Message Service.

If the macro Process_Access_Request_VLR takes the "OK" exit, the VLR waits for a MAP_SEND_INFO_FOR_MO_SMS indication from the MSC.

- If the MSC aborts the dialogue, the process returns to the Null state;
- if the indication is badly formed, the VLR returns a MAP_SEND_INFO_FOR_MO_SMS response containing the appropriate user error;
- if the indication is OK, the VLR checks whether the submission of the short message is allowed
  - if MO SMS is not provisioned, VLR returns a MAP_SEND_INFO_FOR_MO_SMS response containing the user error "Teleservice not provisioned";
  - if the submission of the short message is prevented by Operator Determined Barring of all outgoing calls, the VLR returns a MAP_SEND_INFO_FOR_MO_SMS response containing the user error "Call barred" with barring cause "Operator barring";- if the submission of the short message is prevented by supplementary service barring of all outgoing calls, the VLR returns a MAP_SEND_INFO_FOR_MO_SMS response containing the user error "Call barred" with barring cause "Barring service active";
  - the VLR calls the procedure CAMEL_MO_SMS_VLR and checks the result.
    - if the result is "Fail", the process returns to the Null state;
    - if the result is "Pass", the VLR continues to check the subscription information.
  - if the submission of the short message is prevented by Operator Determined Barring (other than barring of all outgoing calls), the VLR returns a MAP_SEND_INFO_FOR_MO_SMS response containing the user error "Call barred" with barring cause "Operator barring";
  - if the submission of the short message is prevented by supplementary service barring (other than barring of all outgoing calls), the VLR returns a MAP_SEND_INFO_FOR_MO_SMS response containing the user error "Call barred" with barring cause "Barring service active";
  - if the submission of the short message is allowed, the VLR returns a MAP_SEND_INFO_FOR_MO_SMS response containing the MSISDN of the requesting subscriber..

When the VLR has returned the MAP_SEND_INFO_FOR_MO_SMS response, the process returns to the Null state.

The mobile originated short message transfer process in the VLR is shown in figure 23.2/3.

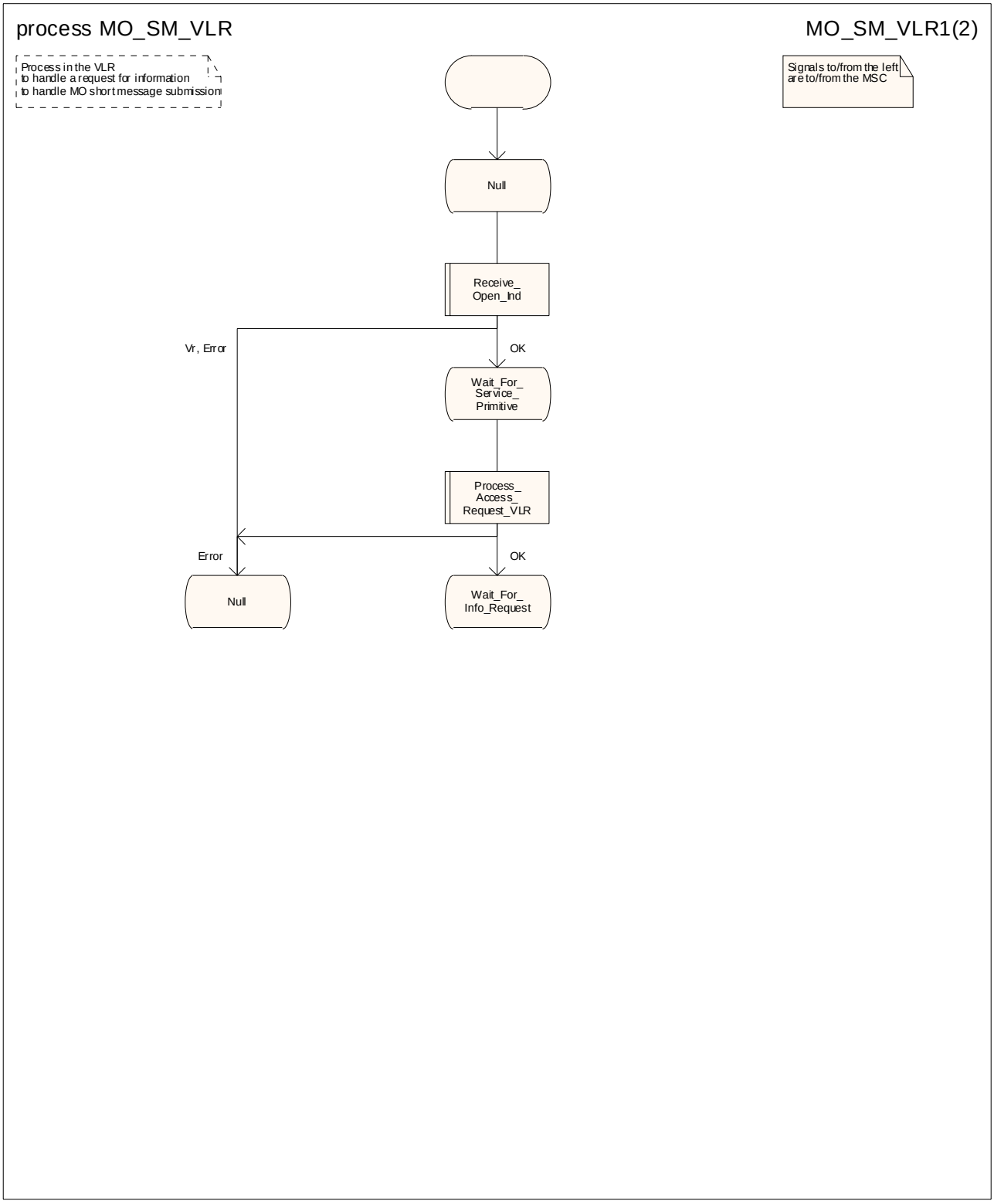


Figure 23.2/3(sheet 1 of 2): Process MOSM_VLR

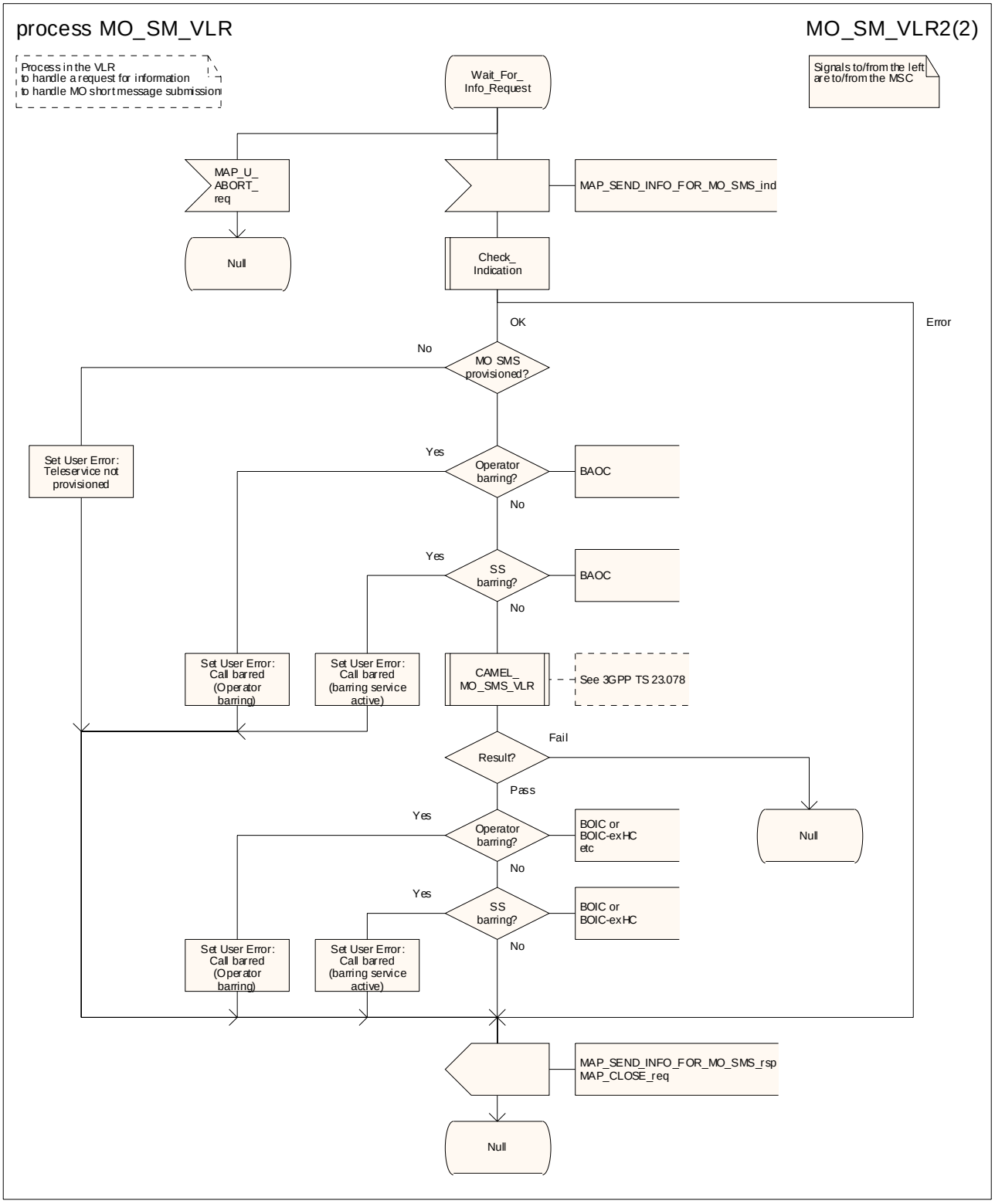


Figure 23.2/3(sheet 2 of 2): Process MO_SM_VLR

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.

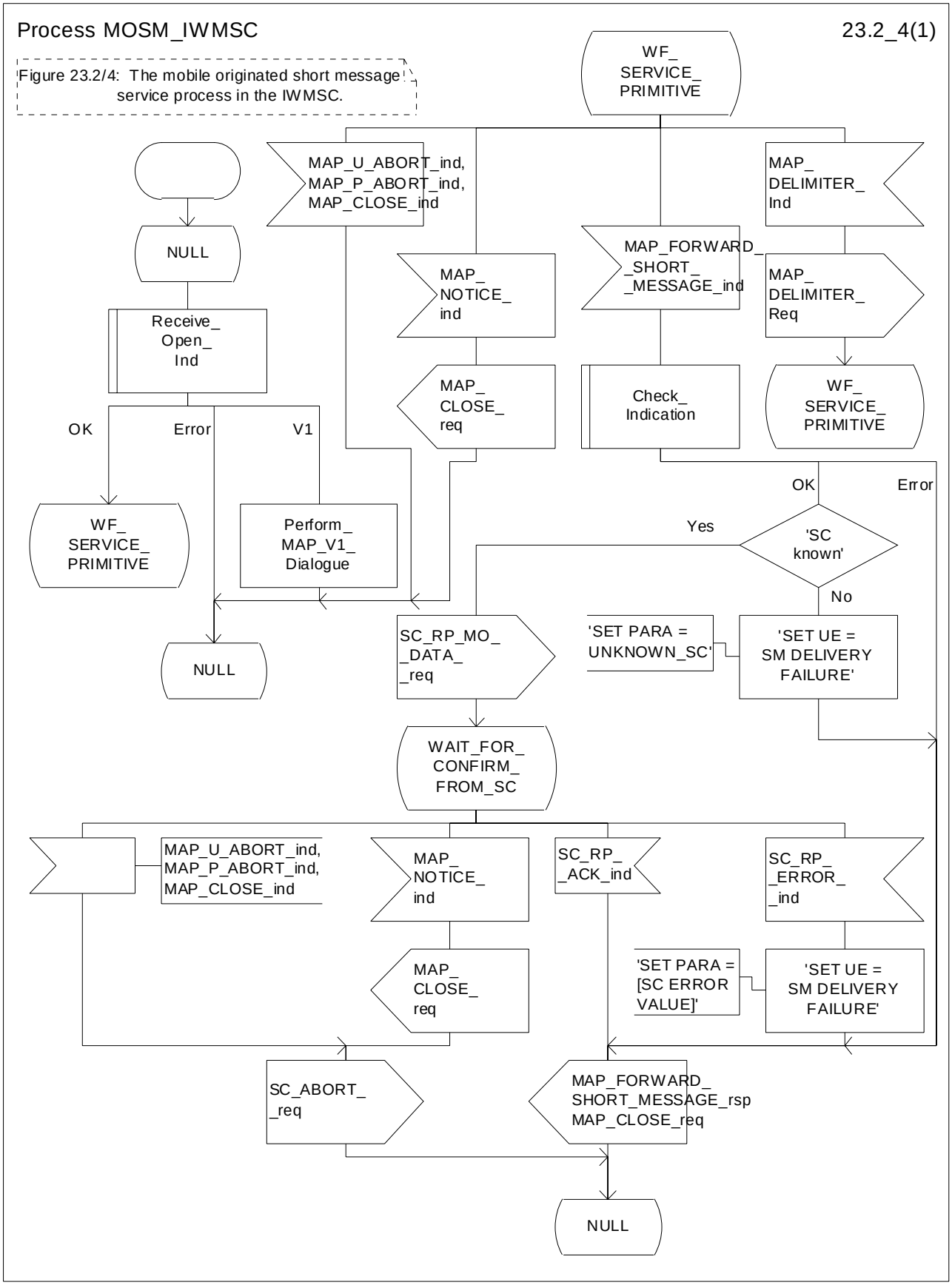


Figure 23.2/4: Process MOSM_IWMSC

23.2.4 Procedure in the SGSN

Any CAMEL-specific handling defined in this subclause is omitted if the SGSN does not support CAMEL control of MO SMS, or if the subscriber is not a CAMEL subscriber.

The process is triggered by a short message received from the MS over the Gb interface.

If the MO SMS service is not provisioned, the SGSN returns a Gb_RP_ERROR with error cause "Requested facility not subscribed", and the process returns to the Null state.

If the MO SMS service is provisioned, the SGSN checks whether Operator Determined Barring of all outgoing calls is in force.

- if Operator Determined Barring would prevent the submission of the short message, the SGSN returns a Gb_RP_ERROR with error cause "Operator determined barring" to the MS, and the process returns to the Null state;

- if Operator Determined Barring would not prevent the submission of the short message, the SGSN handling continues.

The SGSN calls the procedure CAMEL_O_SMS_INIT and tests the result.

- if the result was "SMS_Aborted", the process returns to the Null state;

- if the result was "Release_SMS", the SGSN returns a Gb_RP_ERROR, with an error cause as instructed by the gsmSCF, to the MS, and the process returns to the Null state;

- if the result was "Redirect_SMS", the SGSN modifies the data for the submitted short message as instructed by the gsmSCF, and the MSC handling continues;

- if the result was "Continue", the SGSN handling continues.

The SGSN checks whether Operator Determined Barring of outgoing calls (other than barring of all outgoing calls) would prevent the submission of the short message.

- if Operator Determined Barring would prevent the submission of the short message, the SGSN reports to the gsmSCF that the short message submission has failed and returns a Gb_RP_ERROR with error cause "Operator determined barring" to the MS, and the process returns to the Null state;

- if Operator Determined Barring would not prevent the submission of the short message, the SGSN handling continues.

The SGSN checks whether the MAP_OPEN request and the MAP_MO_FORWARD_SHORT_MESSAGE request can be sent in a single message signal unit through the lower layers of the protocol.

- if the two requests can be grouped in a single TC message, the SGSN requests a dialogue with the SMS-IWMSC, including the MAP_MO_FORWARD_SHORT_MESSAGE request;

    - if the dialogue opening is successful, the SGSN waits for the response from the SMS-IWMSC;

    - if the macro Receive_Open_Cnf takes the "Error" exit, the SGSN reports to the gsmSCF that the short message submission has failed and returns a Gb_RP_ERROR with cause "Network out of order" to the MS, and the process returns to the Null state.

    - if the macro Receive_Open_Cnf takes the "Vr" exit, the SGSN handles the dialogue according to the specification for the earlier version of the protocol and checks the result.

        - if the submission was successful, the SGSN reports to the gsmSCF that the short message submission was successful and returns a Gb_RP_ACK to the MS, and the process returns to the Null state;

        - if the submission failed, the SGSN reports to the gsmSCF that the short message submission has failed and returns a Gb_RP_ERROR with the appropriate error cause to the MS, and the process returns to the Null state.

- if the two requests cannot be grouped in a single TC message, the SGSN requests a dialogue with the SMS-IWMSC, omitting the MAP_MO_FORWARD_SHORT_MESSAGE request;

- if the dialogue opening is successful, the SGSN sends a MAP_MO_FORWARD_SHORT_MESSAGE request to the SMS-IWMSC, and waits for the response from the SMS-IWMSC;

- if the macro Receive_Open_Cnf takes the "Error" exit, the SGSN reports to the gsmSCF that the short message submission has failed and returns a Gb_RP_ERROR with cause "Network out of order" to the MS, and the process returns to the Null state.

- if the macro Receive_Open_Cnf takes the "Vr" exit, the SGSN handles the dialogue according to the specification for the earlier version of the protocol and checks the result.

  - if the submission was successful, the SGSN reports to the gsmSCF that the short message submission was successful and returns a Gb_RP_ACK to the MS, and the process returns to the Null state;

  - if the submission failed, the SGSN reports to the gsmSCF that the short message submission has failed and returns a Gb_RP_ERROR with the appropriate error cause to the MS, and the process returns to the Null state.

- if the SGSN receives a MAP_MO_FORWARD_SHORT_MESSAGE confirmation from the SMS-IWMSC, it checks the content of the confirmation;

  - if the confirmation indicates that the submission of the short message was successful, the SGSN reports to the gsmSCF that the short message submission was successful and returns a Gb_RP_ACK to the MS, and the process returns to the Null state;

  - if the confirmation indicates that the submission of the short message failed, the SGSN reports to the gsmSCF that the short message submission has failed and returns a Gb_RP_ERROR with the appropriate error cause to the MS, and the process returns to the Null state;

- if the dialogue with the SMS-IWMSC fails, the SGSN reports to the gsmSCF that the short message submission has failed and returns an A_RP_ERROR with cause "Network out of order" to the MS, and the process returns to the Null state.

The mobile originated short message service process in the SGSN is shown in figure 23.2/5.

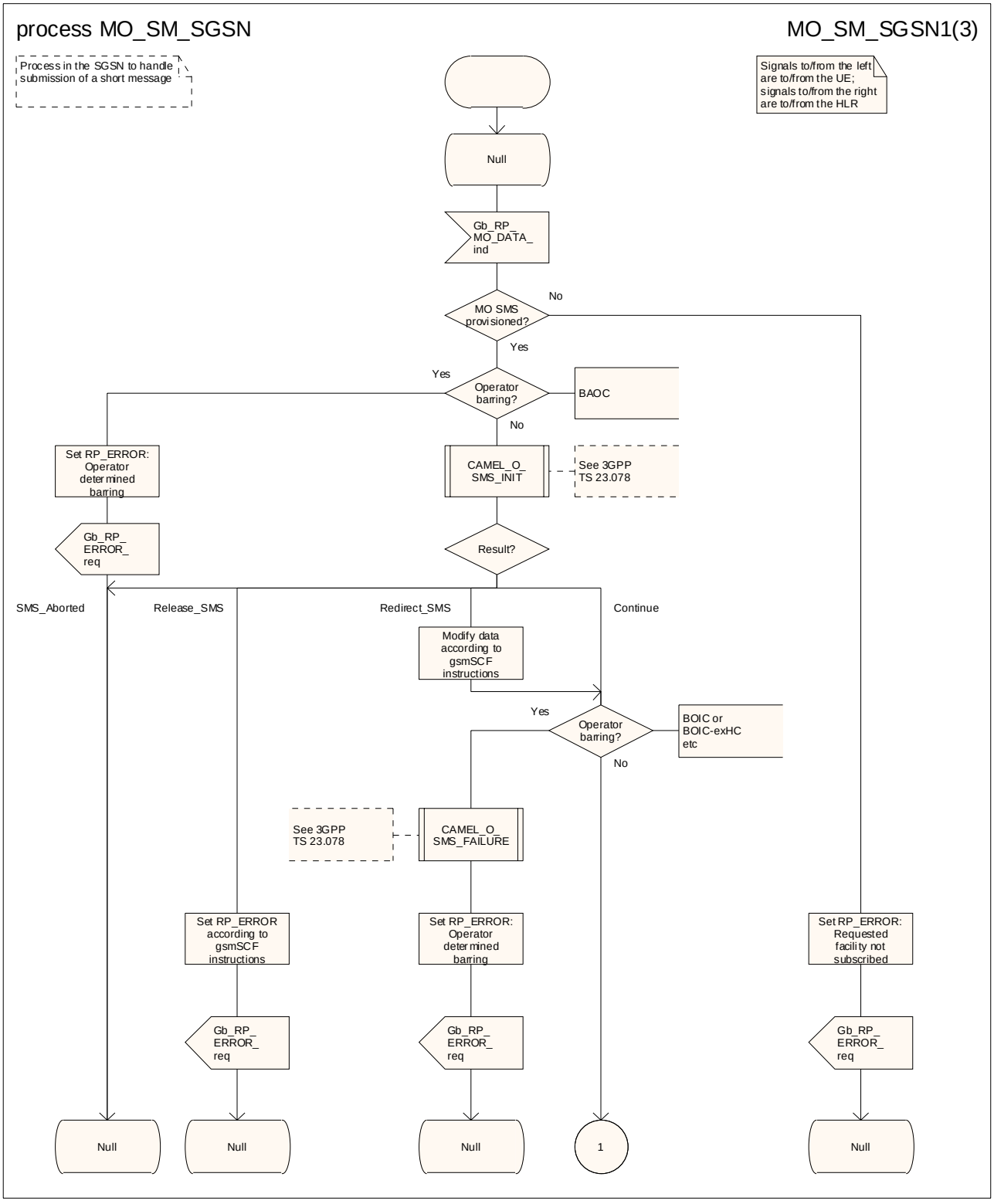


Figure 23.2/5 (sheet 1 of 3): Process MO_SM_SGSN

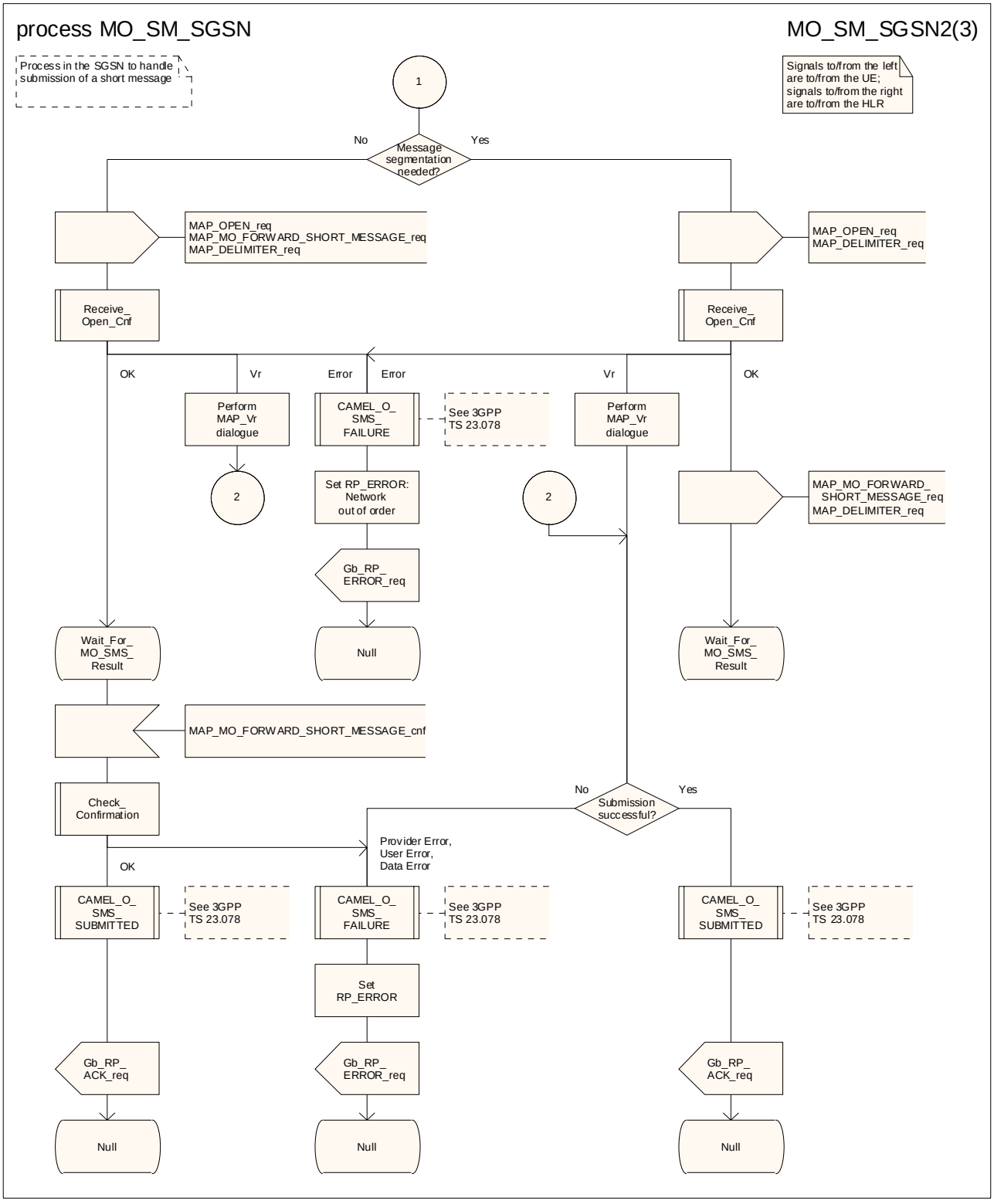


Figure 23.2/5 (sheet 2 of 3): Process MO_SM_SGSN

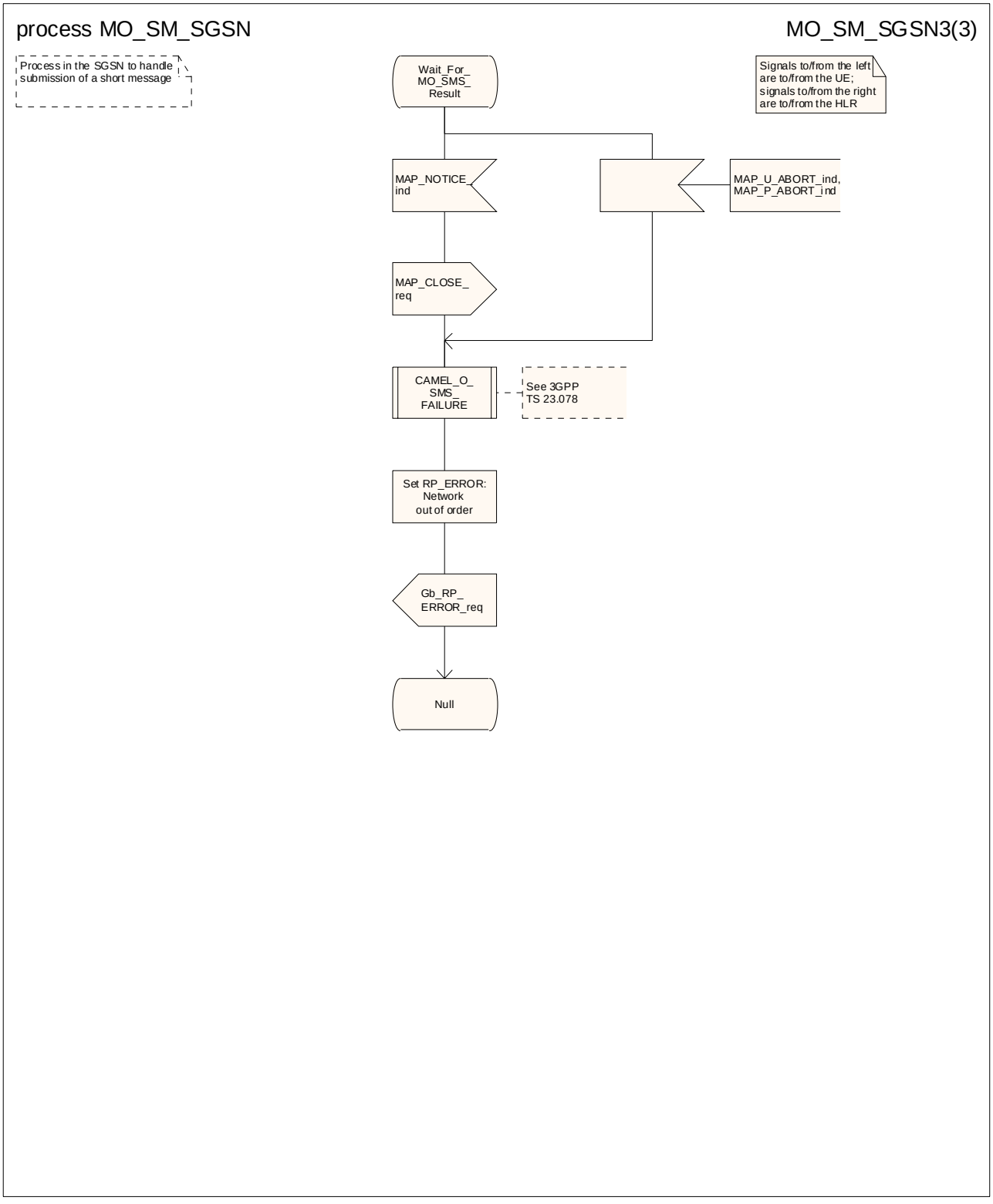


Figure 23.2/5 (sheet 3 of 3): Process MO_SM_SGSN

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.

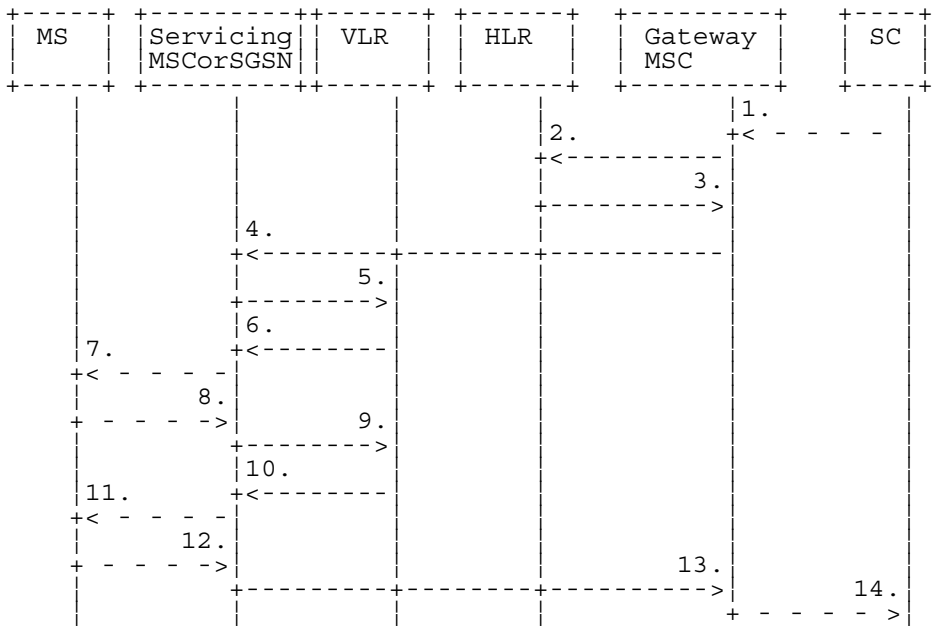


Figure 23.3/1: Mobile terminated short message service procedures

1) Short Message (3GPP TS 23.040[26]).
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 (3GPP TS 24.008).

8) Page response (3GPP TS 24.008).

9) MAP_PROCESS_ACCESS_REQUEST_ACK and MAP_SEARCH_FOR_MOBILE_SUBSCRIBER_ACK (*).

10) MAP_SEND_INFO_FOR_MT_SMS_ACK (*).

11) Short Message (3GPP TS 24.011 [37]).

12) Short Message Acknowledgement (3GPP TS 24.011 [37]).

13) MAP_MT_FORWARD_SHORT_MESSAGE_ACK.

14) Short Message Acknowledgement (3GPP TS 23.040[26]).

(*) Messages 5), 6), 9), and 10) are not used by SGSN.

The mobile terminated short message procedure for multiple short message transfer is shown in figure 23.3/2.

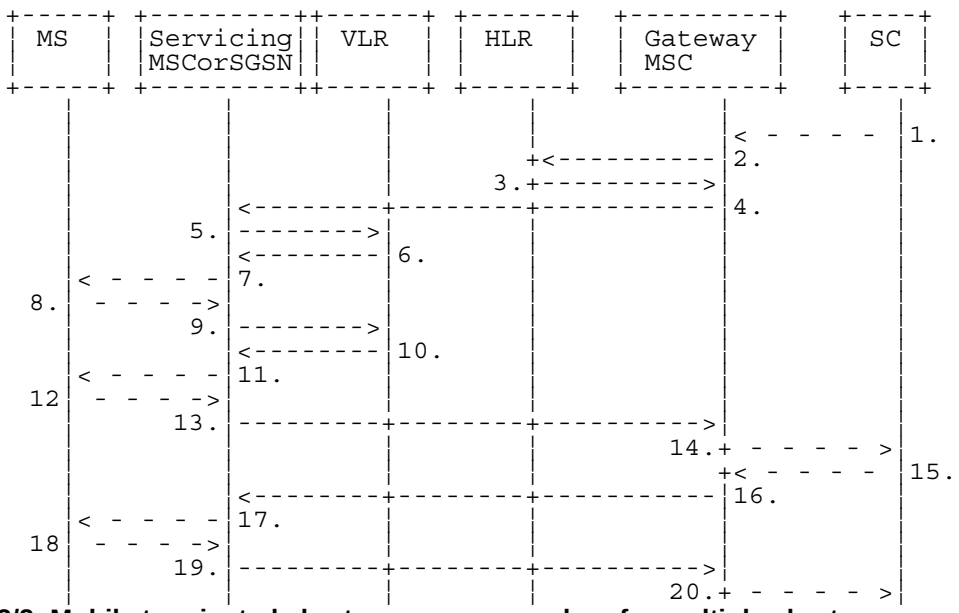


Figure 23.3/2: Mobile terminated short message procedure for multiple short message transfer

1) Short Message (3GPP TS 23.040[26]).

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 (3GPP TS 24.008).

9) MAP_PROCESS_ACCESS_REQUEST_ACK and
MAP_SEARCH_FOR_MOBILE_SUBSCRIBER_ACK (*).

10) MAP_SEND_INFO_FOR_MT_SMS_ACK (*).

11) Short Message (3GPP TS 24.011 [37]).

12) Short Message Acknowledgement (3GPP TS 24.011 [37]).

13) MAP_MT_FORWARD_SHORT_MESSAGE_ACK.

14) Short Message Acknowledgement (3GPP TS 23.040[26]).

15) Short Message (3GPP TS 23.040[26]).

16) MAP_MT_FORWARD_SHORT_MESSAGE (note 2).

17) Short Message (3GPP TS 24.011 [37]).

18) Short Message Acknowledgement (3GPP TS 24.011 [37]).

19) MAP_MT_FORWARD_SHORT_MESSAGE_ACK.

20) Short Message Acknowledgement (3GPP TS 23.040[26]).

(*) 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.

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 clause 8.3); (*)
MAP_PAGE	(see clause 8.2); (*)
MAP_SEARCH_FOR_MS	(see clause 8.2); (*)
MAP_AUTHENTICATE	(see clause 8.5); (*)
MAP_SET_CIPHERING_MODE	(see clause 8.6); (*)
MAP_CHECK_IMEI	(see clause 8.7);
MAP_FORWARD_NEW_TMSI	(see clause 8.9); (*)
MAP_REPORT_SM_DELIVERY_STATUS	(see clause 12.3);
MAP_INFORM_SERVICE_CENTRE	see clause 12.6);
MAP_TRACE_SUBSCRIBER_ACTIVITY	(see clause 9.1); (*)
MAP_READY_FOR_SM	(see clause 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 clause 25.3) reporting the successful outcome of the procedure;
- search procedure invocation (see clause 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 clause 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 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 clause 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.

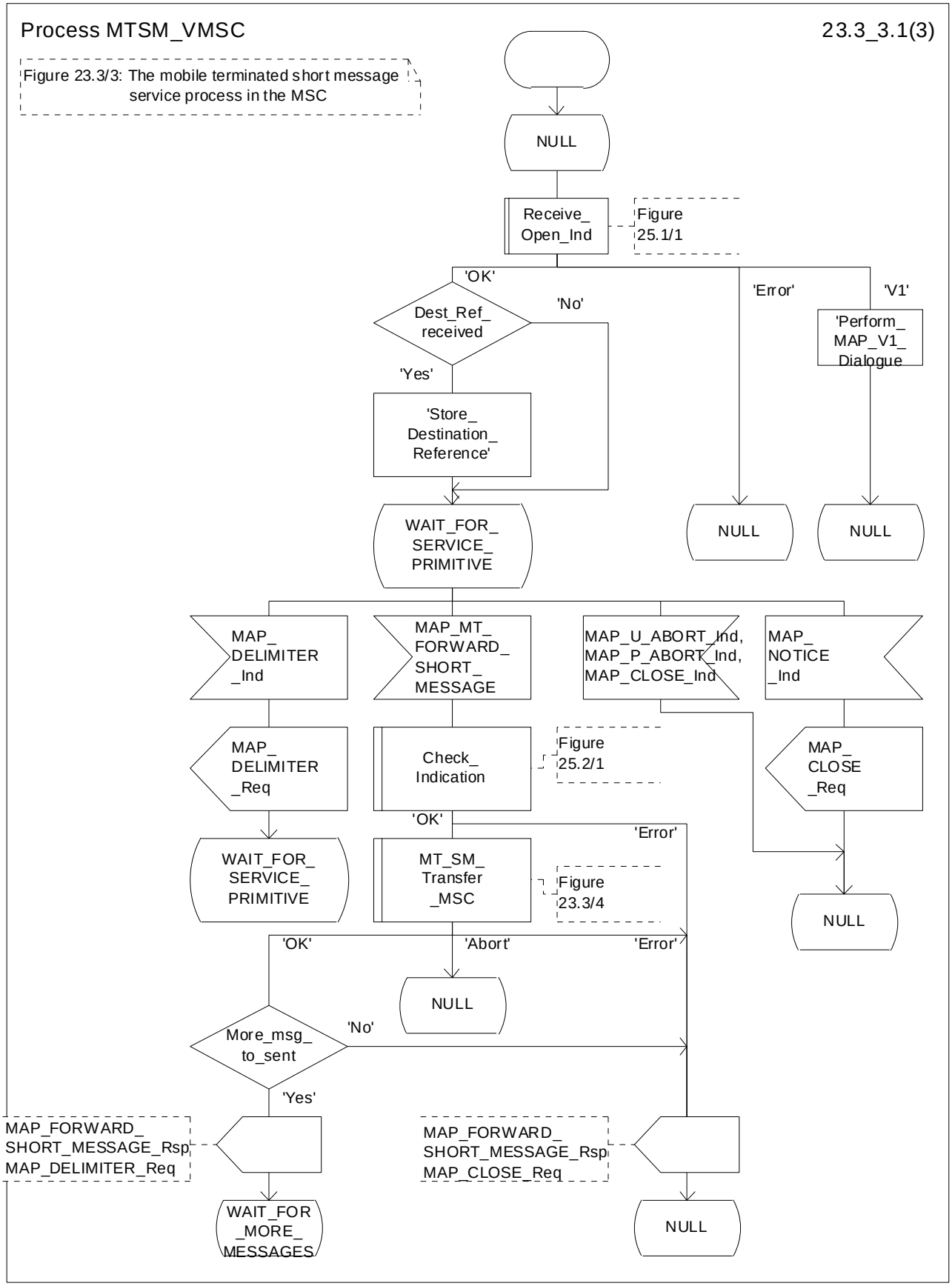


Figure 23.3/3 (sheet 1 of 3): Procedure MTSM_VMSC

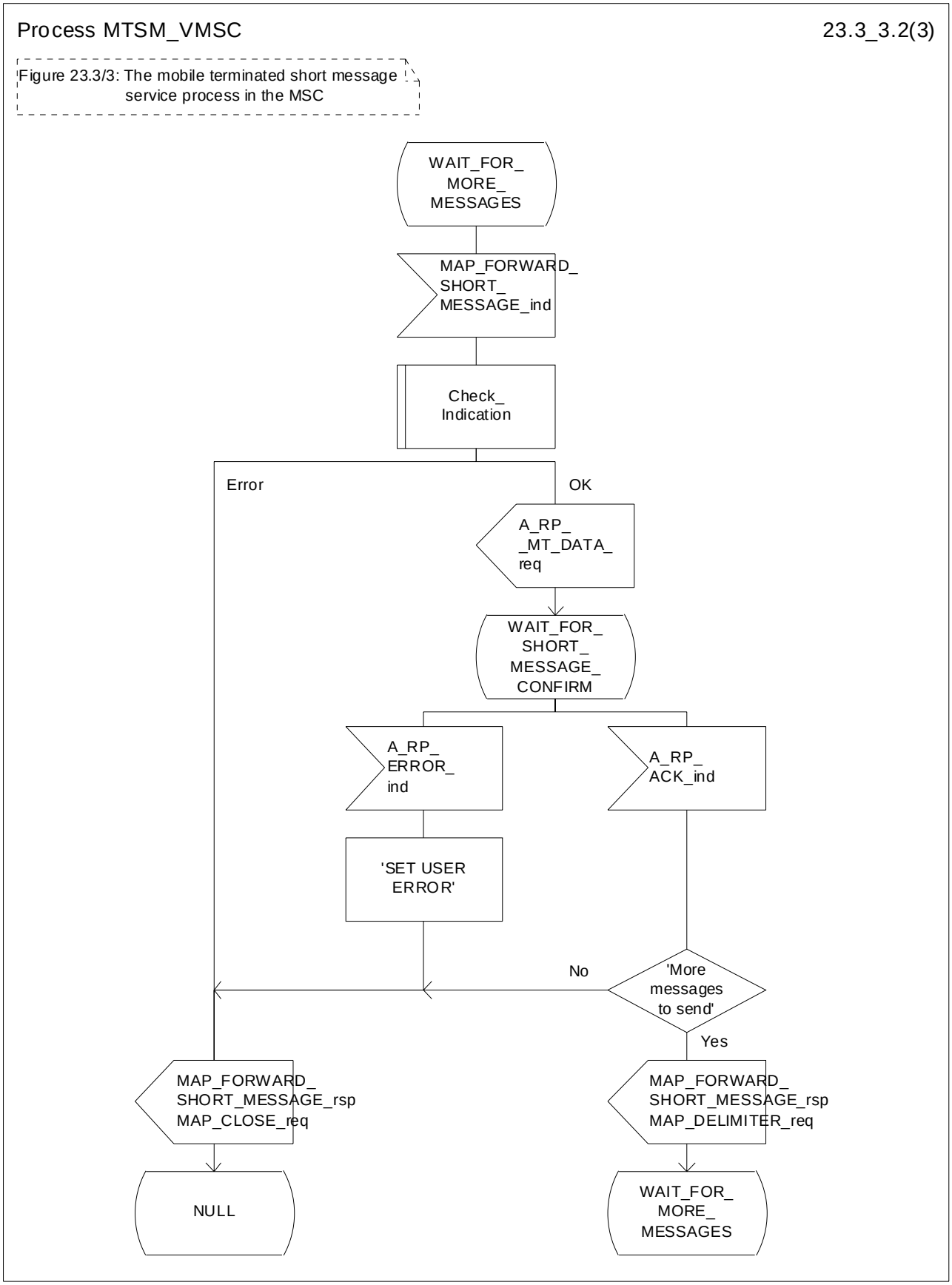


Figure 23.3/3 (sheet 2 of 3): Procedure MTSM_VMSC

Process MTSM_VMSC 23.3_3.3(3)

Figure 23.3/3: The mobile terminated short message service process in the MSC

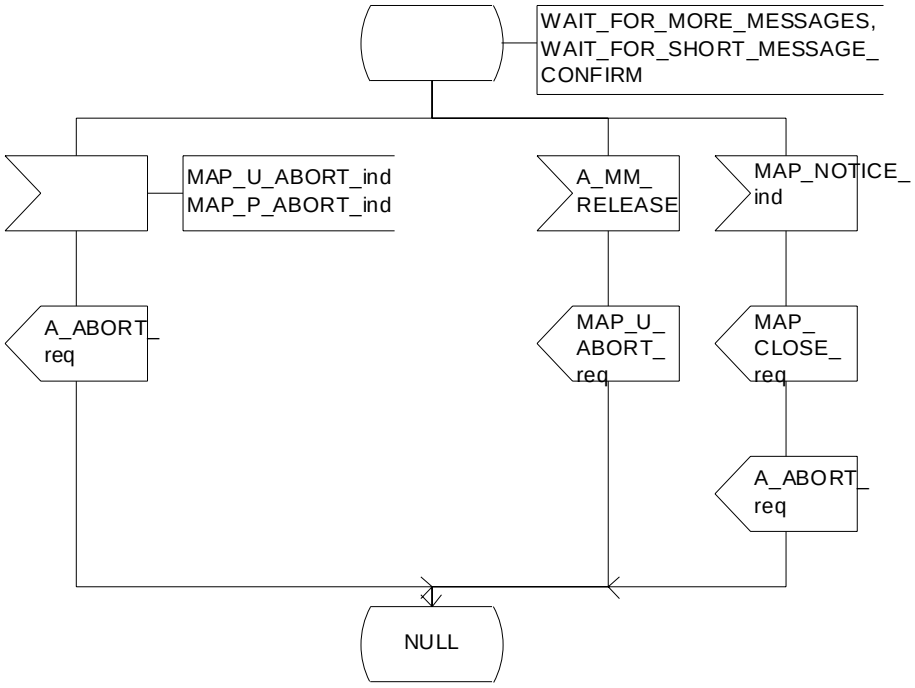


Figure 23.3/3 (sheet 3 of 3): Procedure MTSM_VMSC

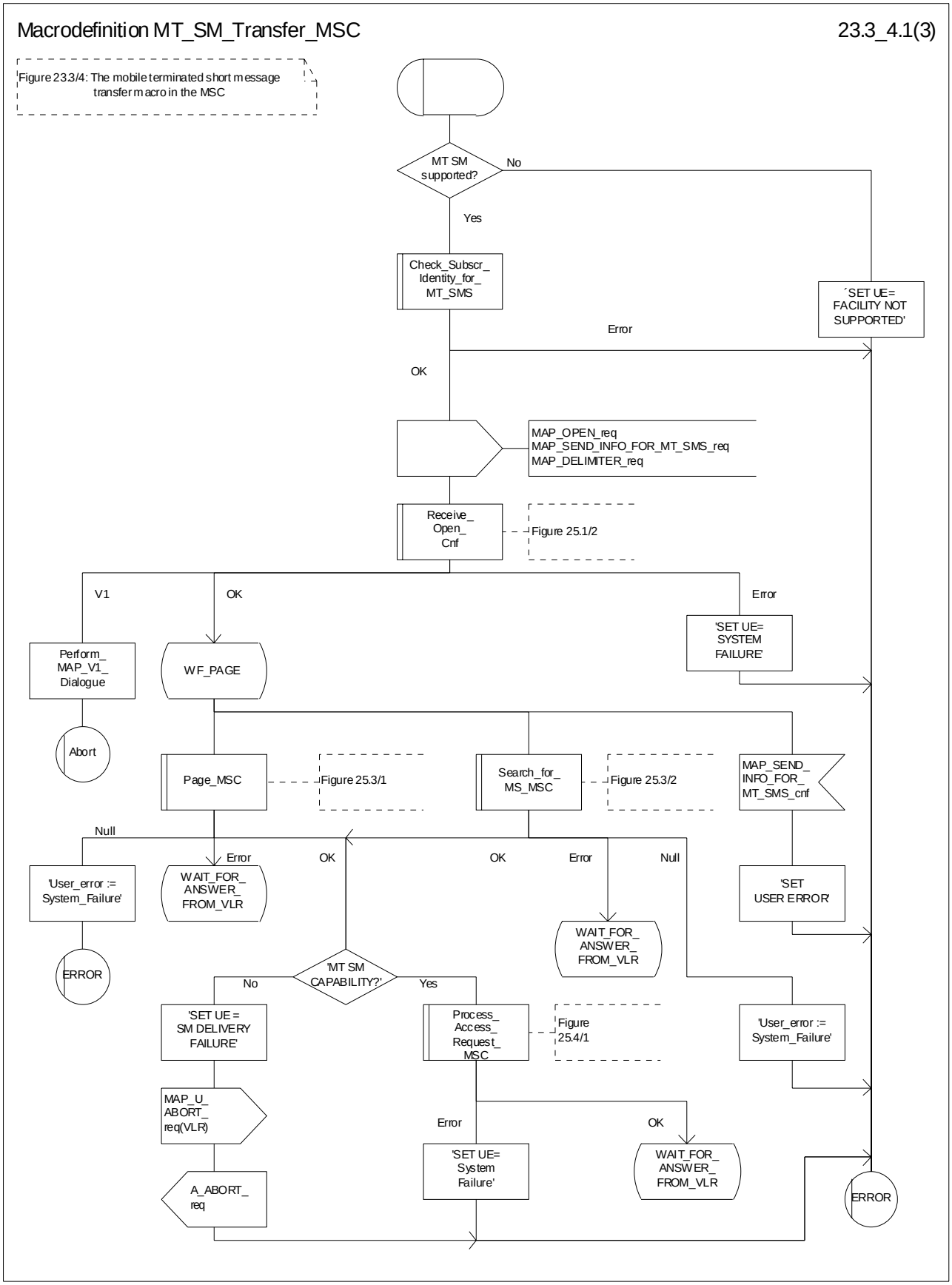


Figure 23.3/4 (sheet 1 of 3): Macro MT_SM_Transfer_MSC

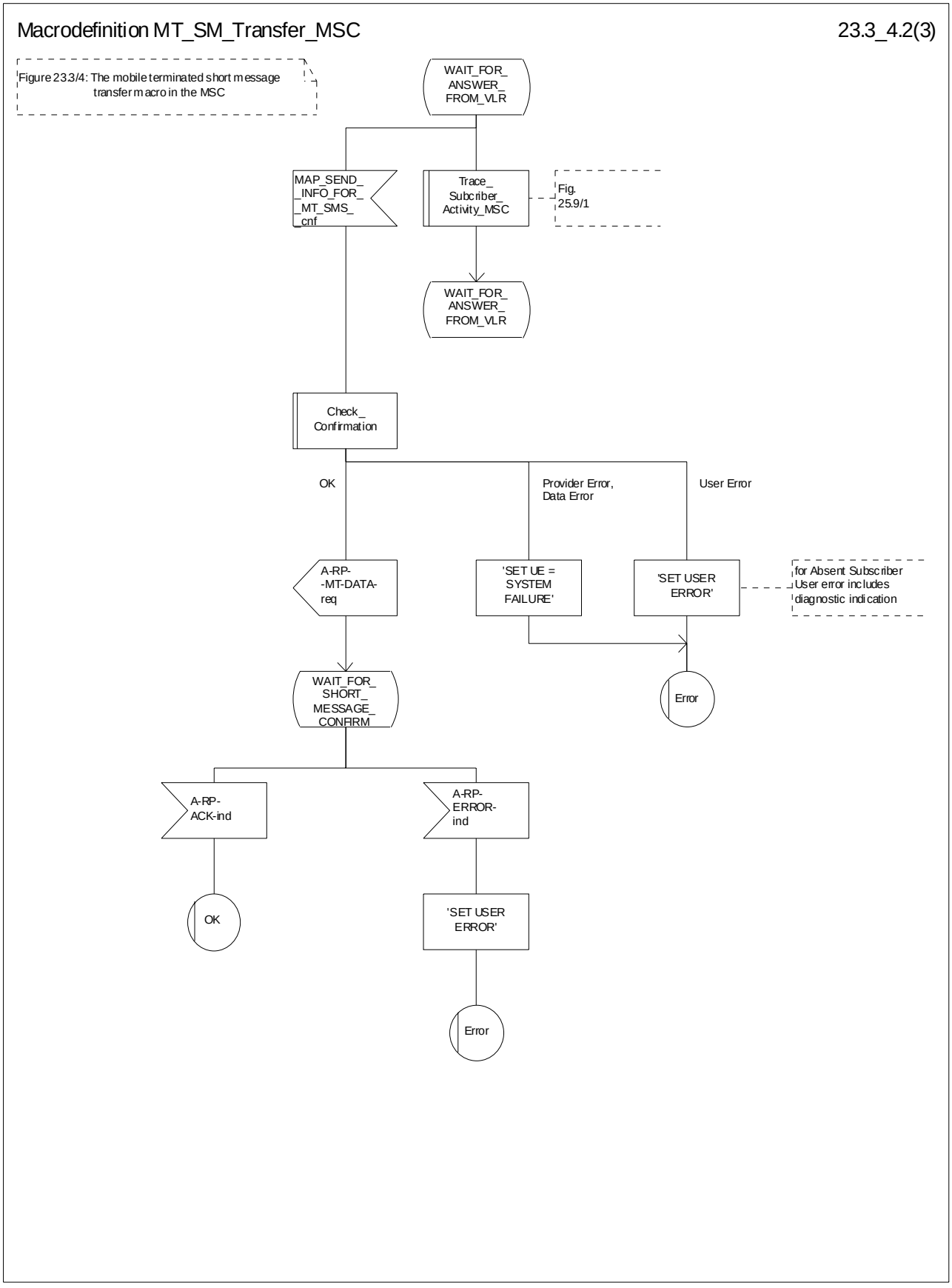


Figure 23.3/4 (sheet 2 of 3): Macro MT_SM_Transfer_MSC

Macrodefinition MT_SM_Transfer_MSC 23.3_4.3(3)

Figure 23.3/4: The mobile terminated short message transfer macro in the MSC

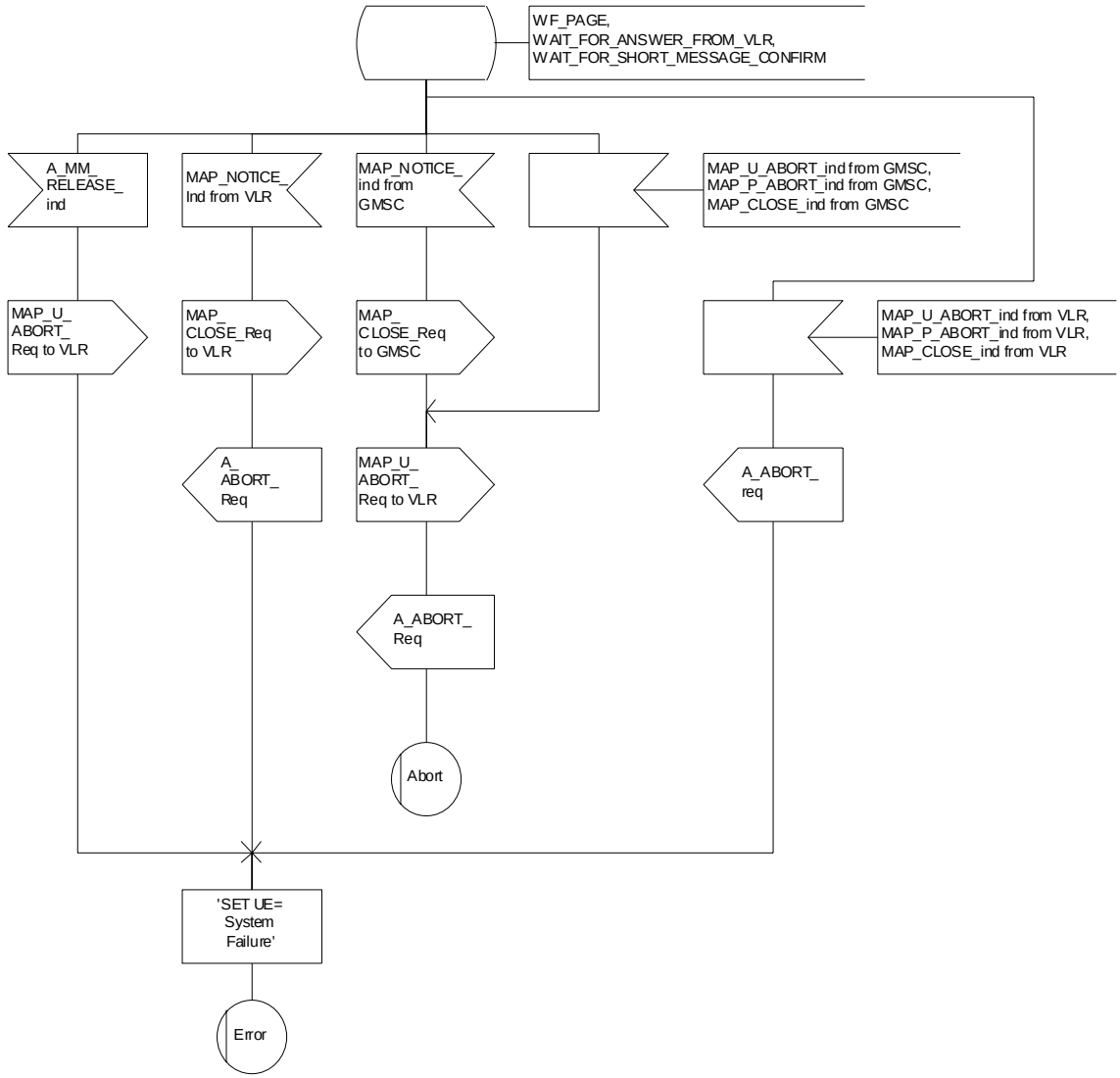


Figure 23.3/4 (sheet 3 of 3): Macro MT_SM_Transfer_MSC

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.

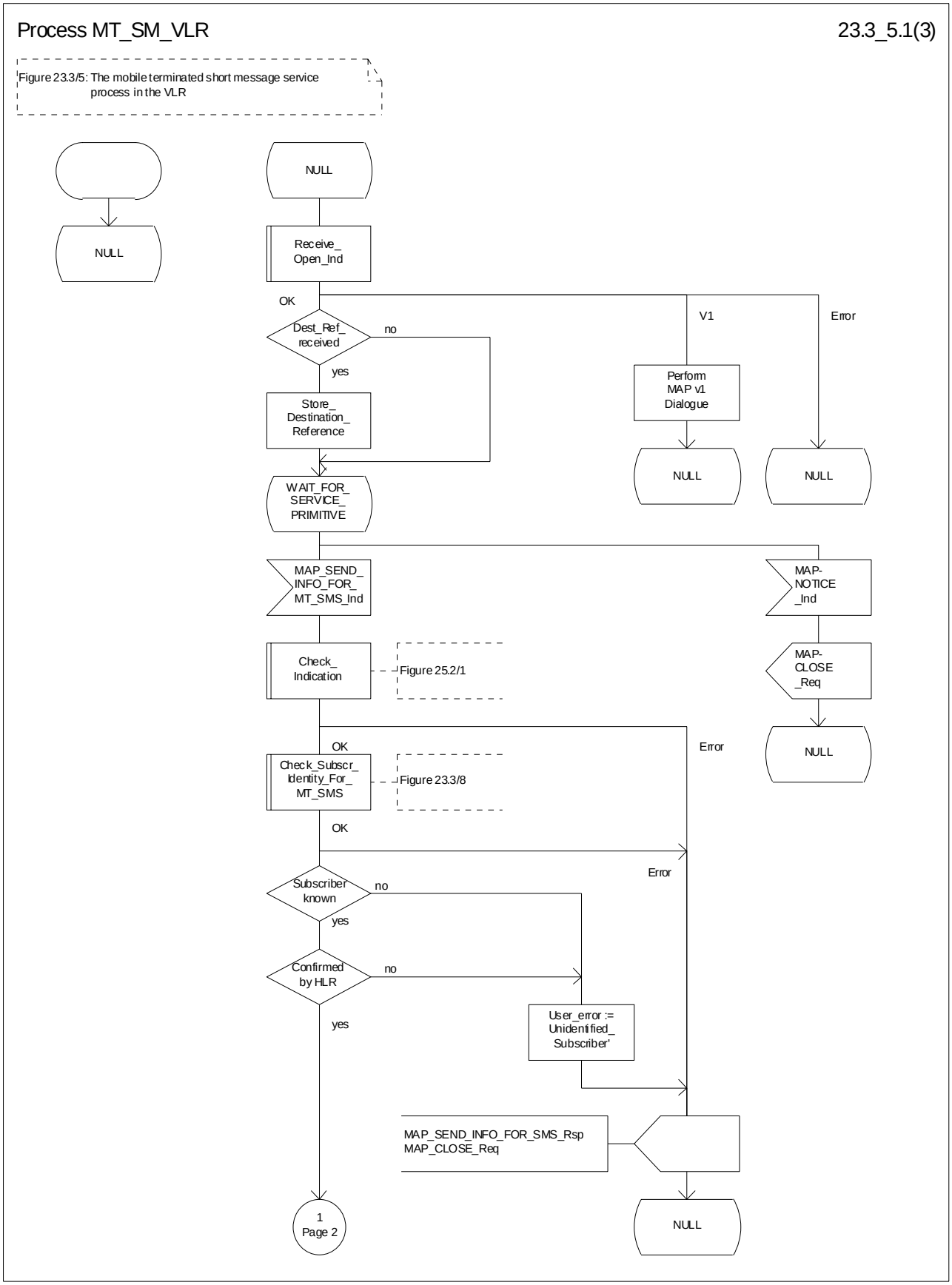


Figure 23.3/5 (sheet 1 of 3): Process MT_SM_VLR

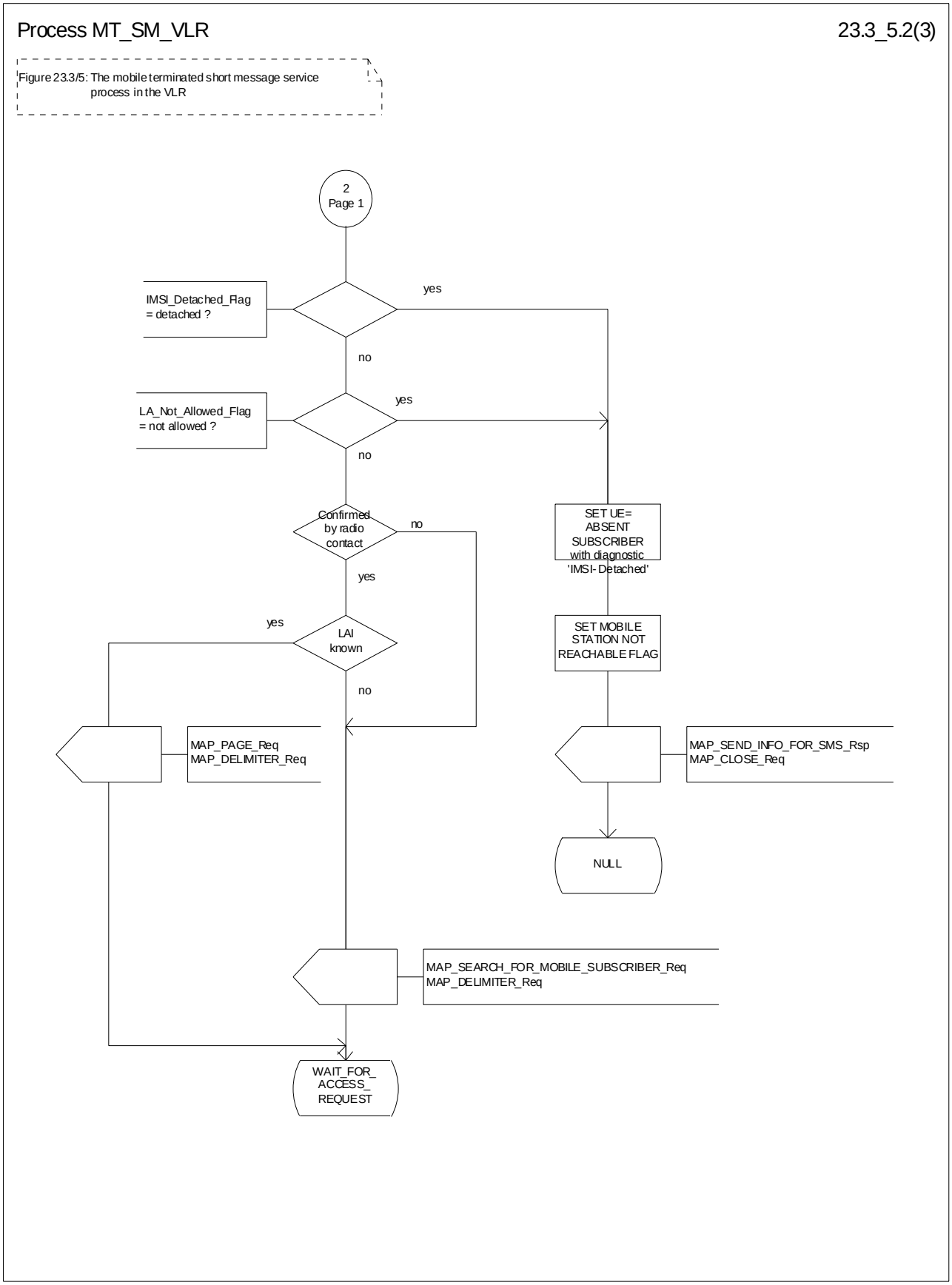


Figure 23.3/5 (sheet 2 of 3): Process MT_SM_VLR

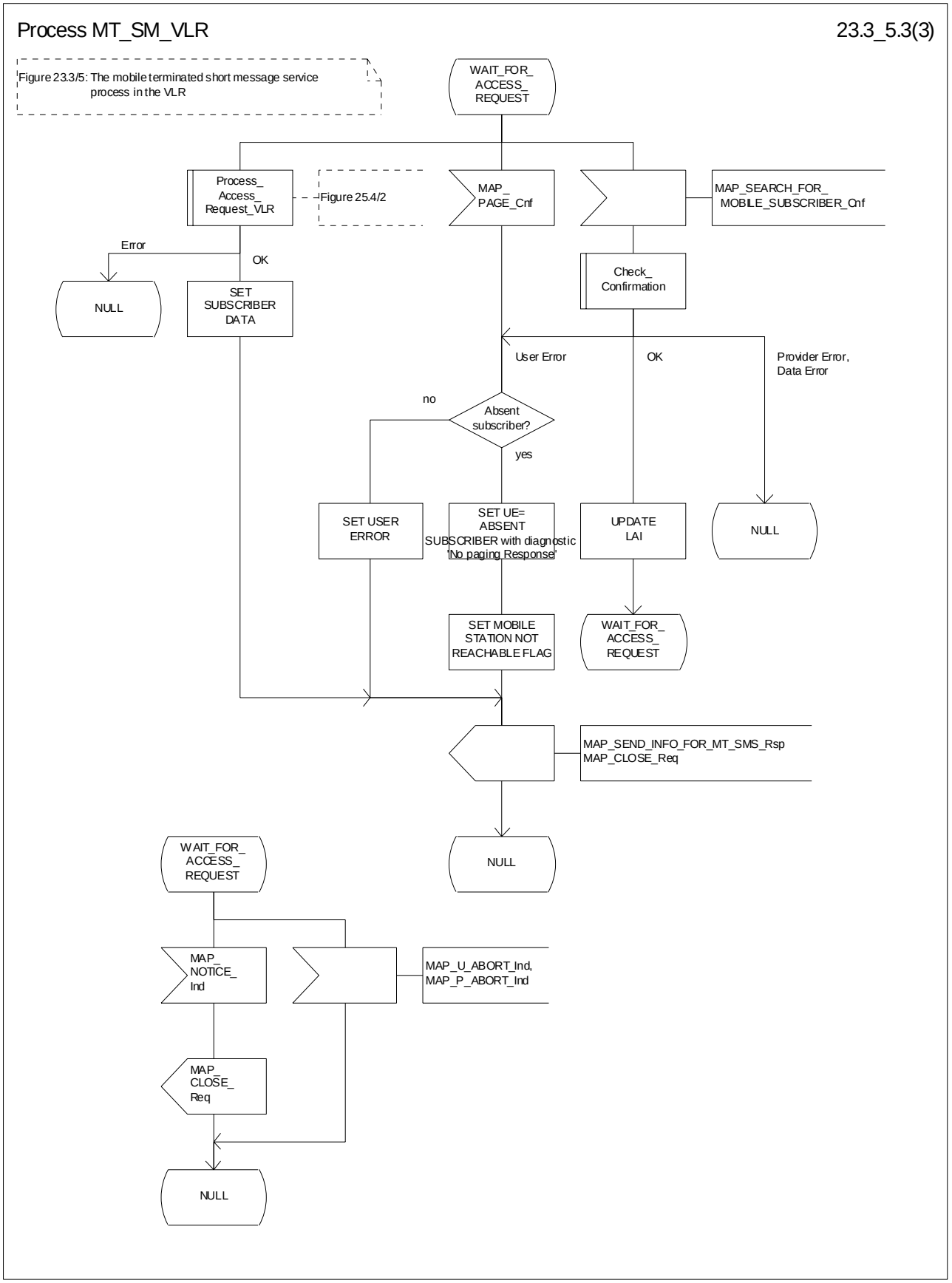


Figure 23.3/5 (sheet 3 to 3): Process MT_SM_VLR

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 routeing 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.

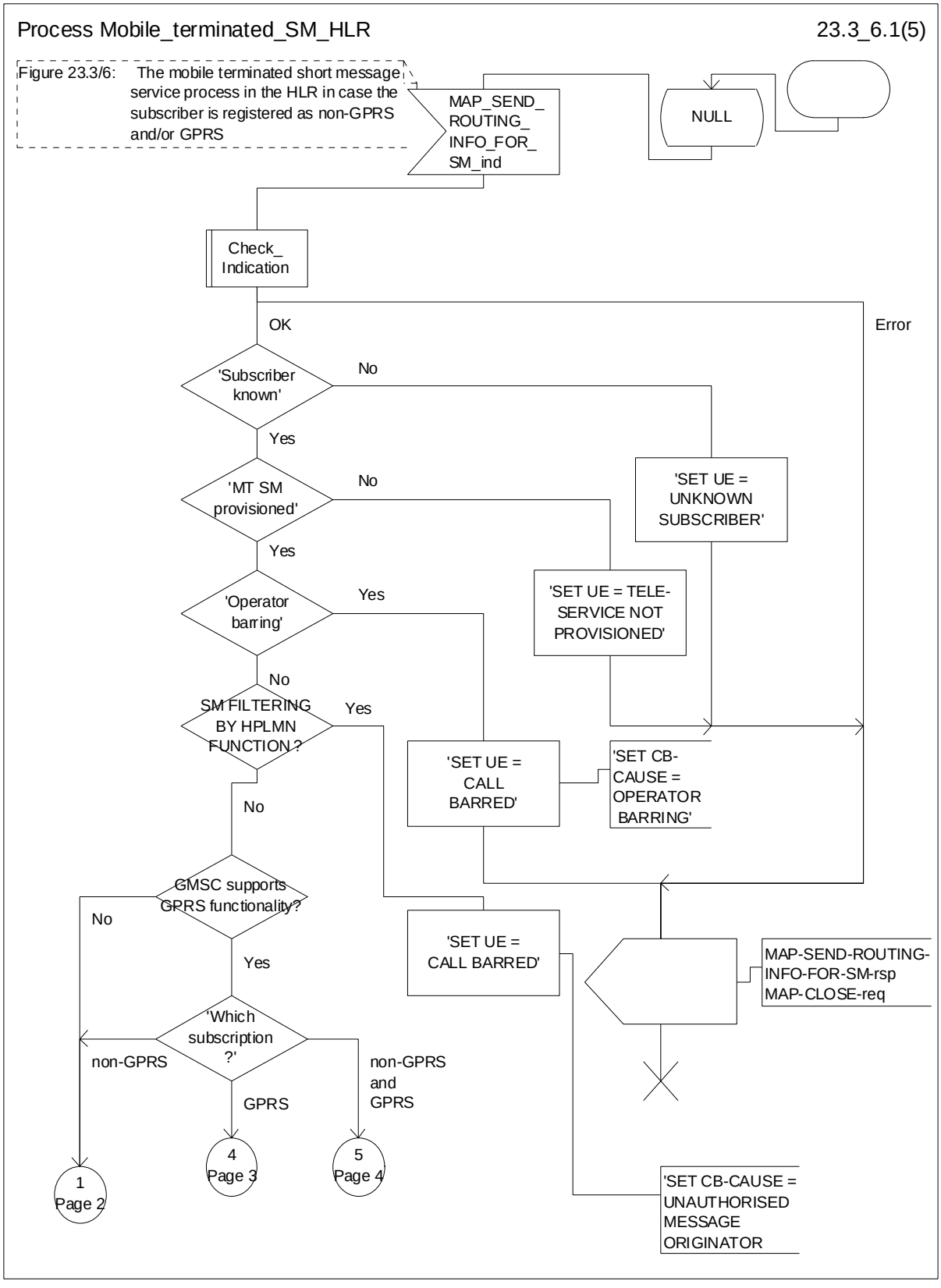


Figure 23.3/6 (sheet 1 of 5): Process Mobile_terminated_SM_HLR

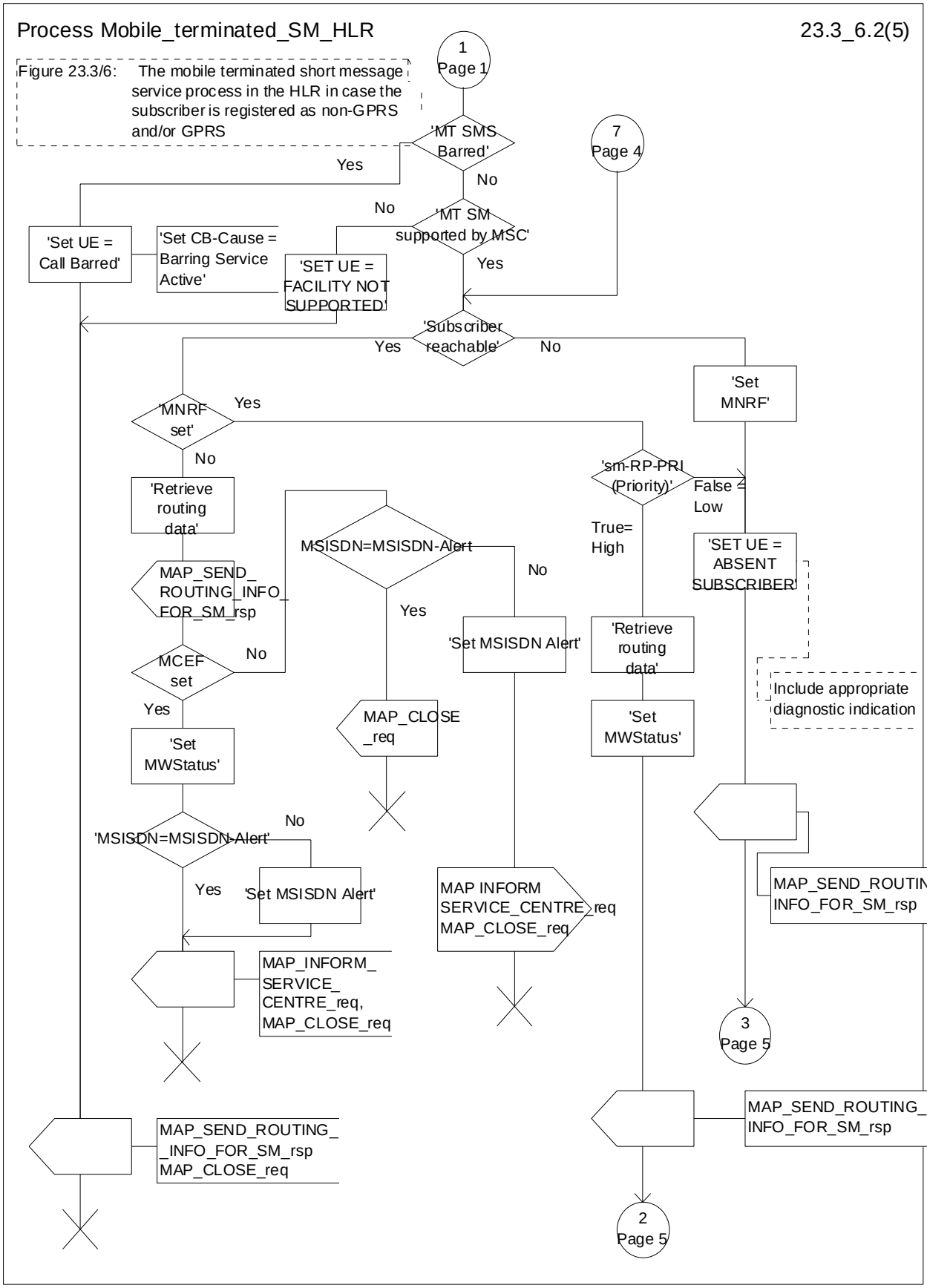


Figure 23.3/6 (sheet 2 of 5): Process Mobile_terminated_SM_HLR

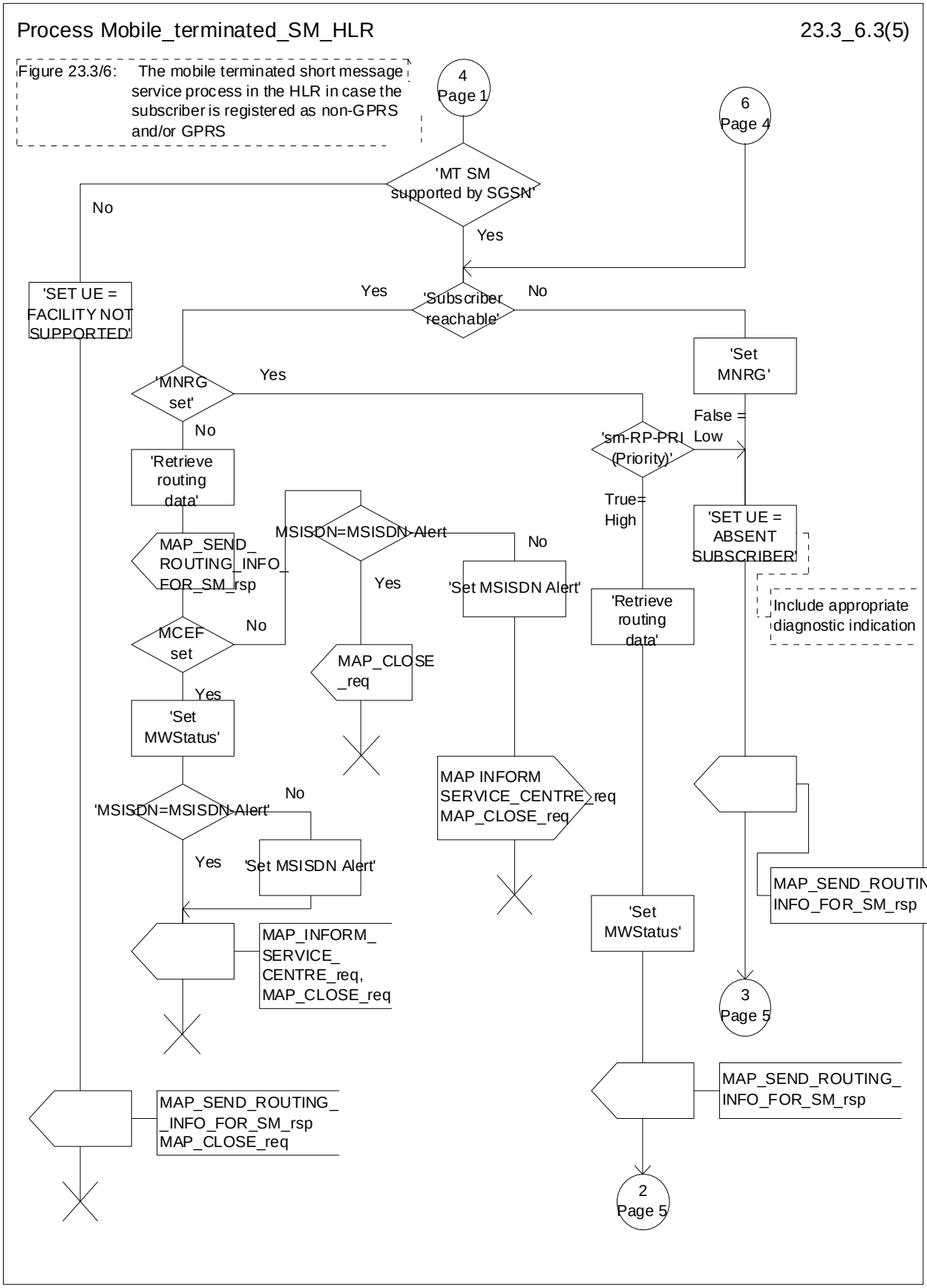


Figure 23.3/6 (sheet 3 of 5): Process Mobile_terminated_SM_HLR

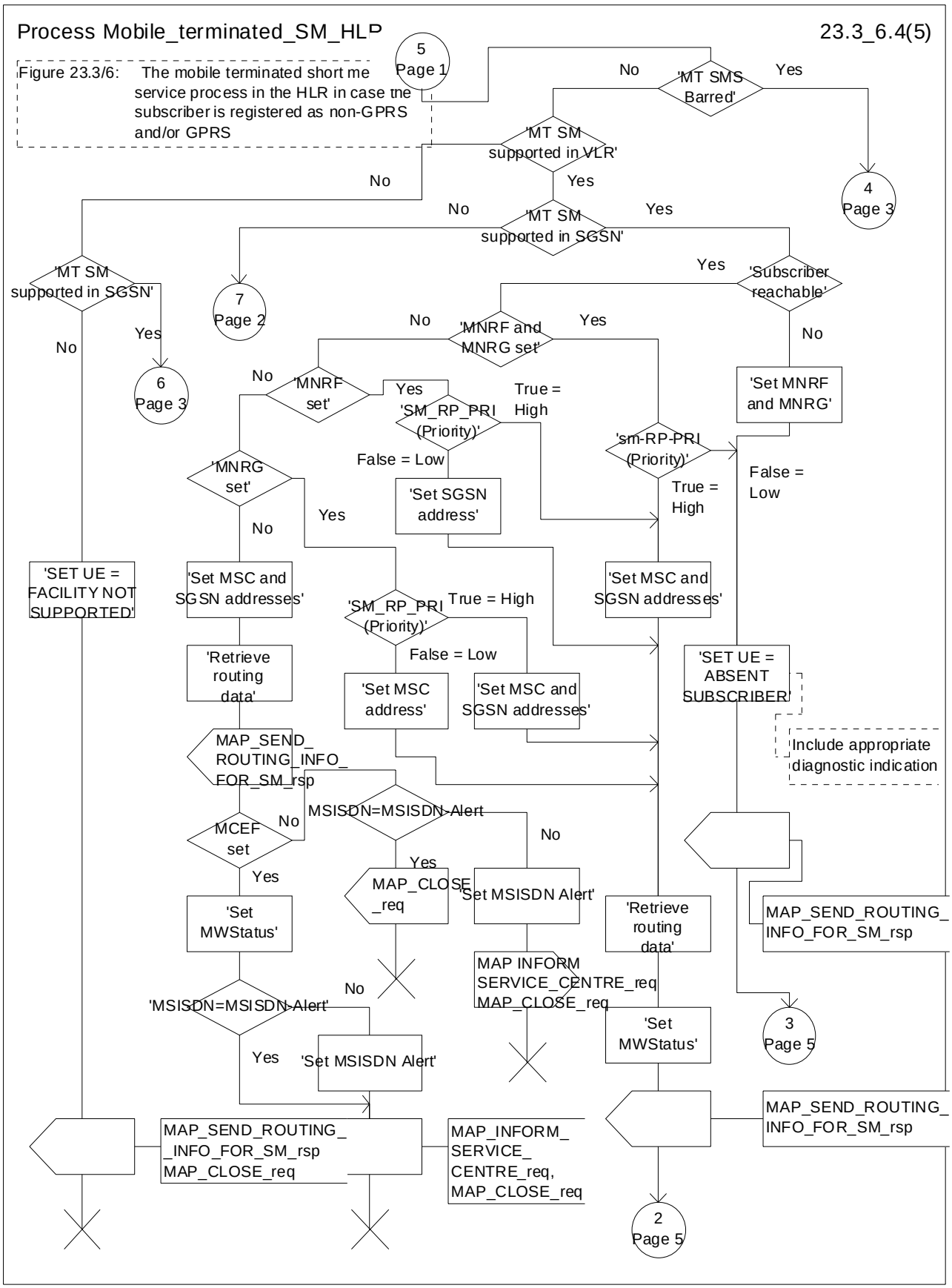


Figure 23.3/6 (sheet 4 of 5): Process Mobile_terminated_SM_HLR

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

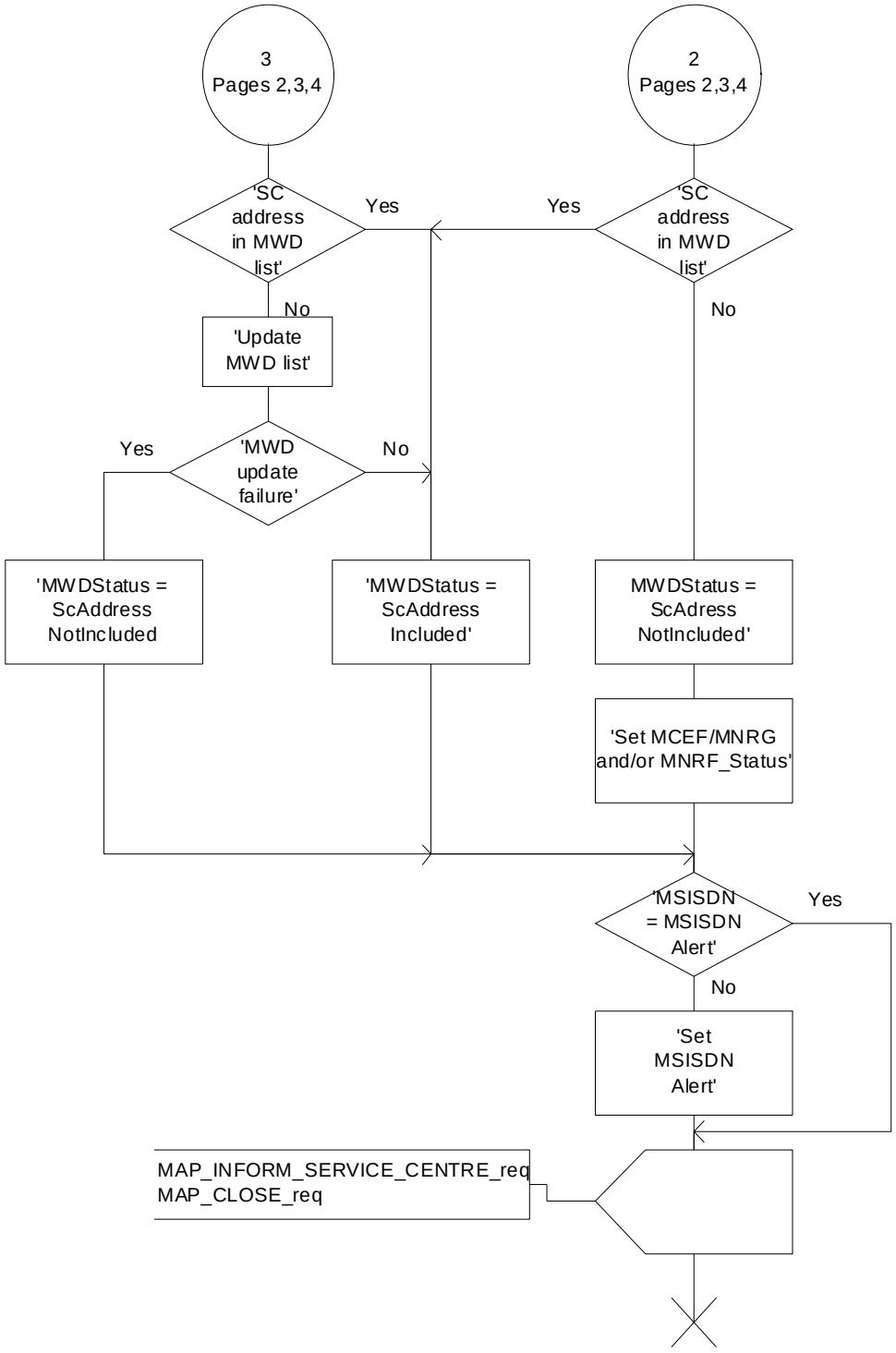


Figure 23.3/6 (sheet 5 of 5): Process Mobile_terminated_SM_HLR

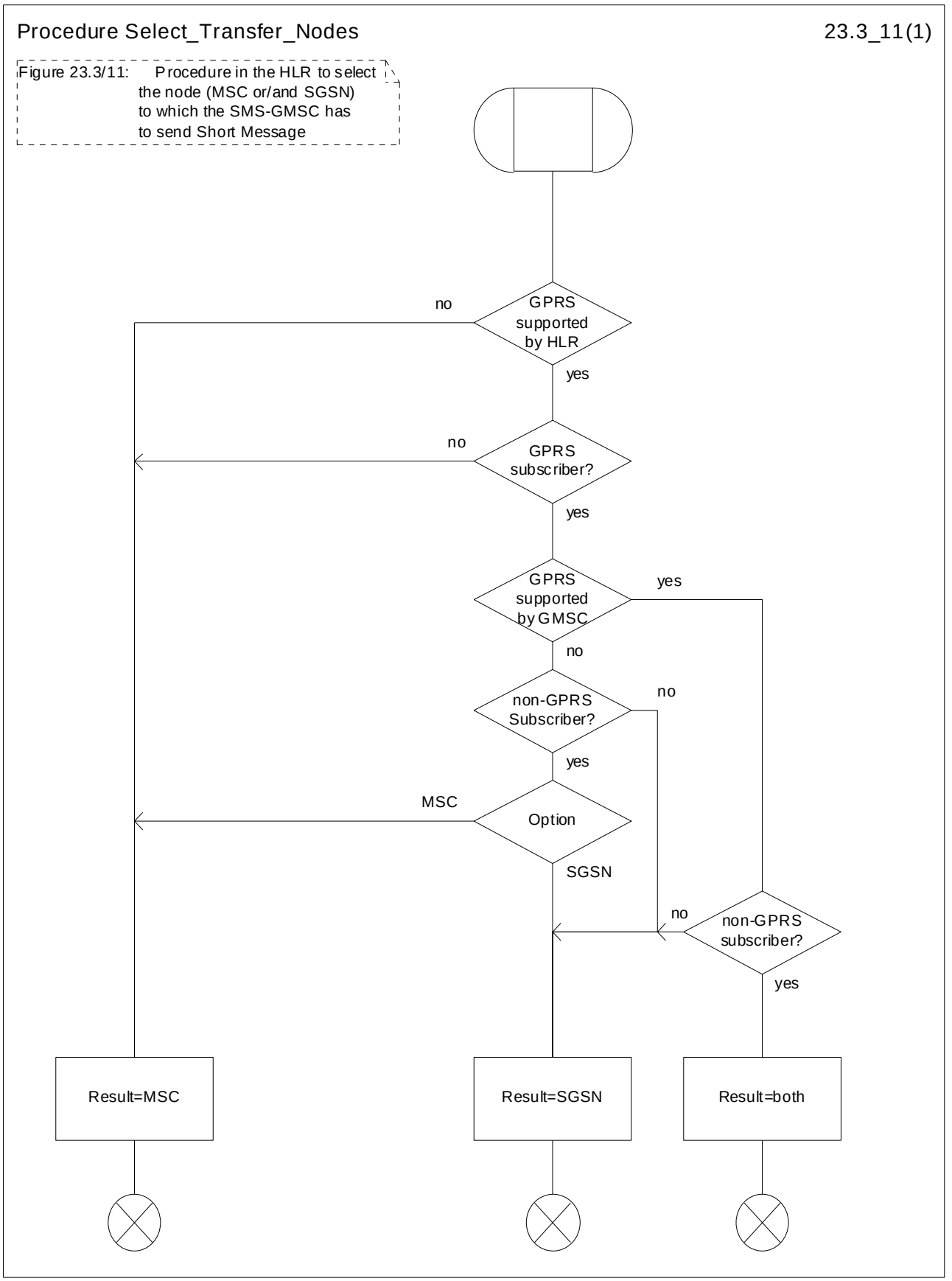


Figure 23.3/11: Procedure Select_Transfer_Nodes

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 3GPP TS 23.040[26].

- 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 clause 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 3GPP TS 23.040[26]. 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

SM Delivery Failure with indication: equipment Not SM Equipped

otherwise, the mapping of the MAP error causes and the RP_ERROR causes is performed (see 3GPP TS 23.040[26]) 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 3GPP TS 23.040[26]. 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 3GPP TS 23.040[26]) 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 clause 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 3GPP TS 23.040[26] and 3GPP TS 24.011 [37].

The unidentified subscriber error is indicated to the SC as absent subscriber with diagnostic information set to 'Unidentified subscriber' as described in 3GPP TS 23.040[26].

Note that the indication, on which number belongs the SGSN and MSC, received from the HLR at routing information result (see clause 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 clause 23.3.1.

The mobile terminated short message transfer procedure in the gateway MSC is shown in figure 23.3/7.

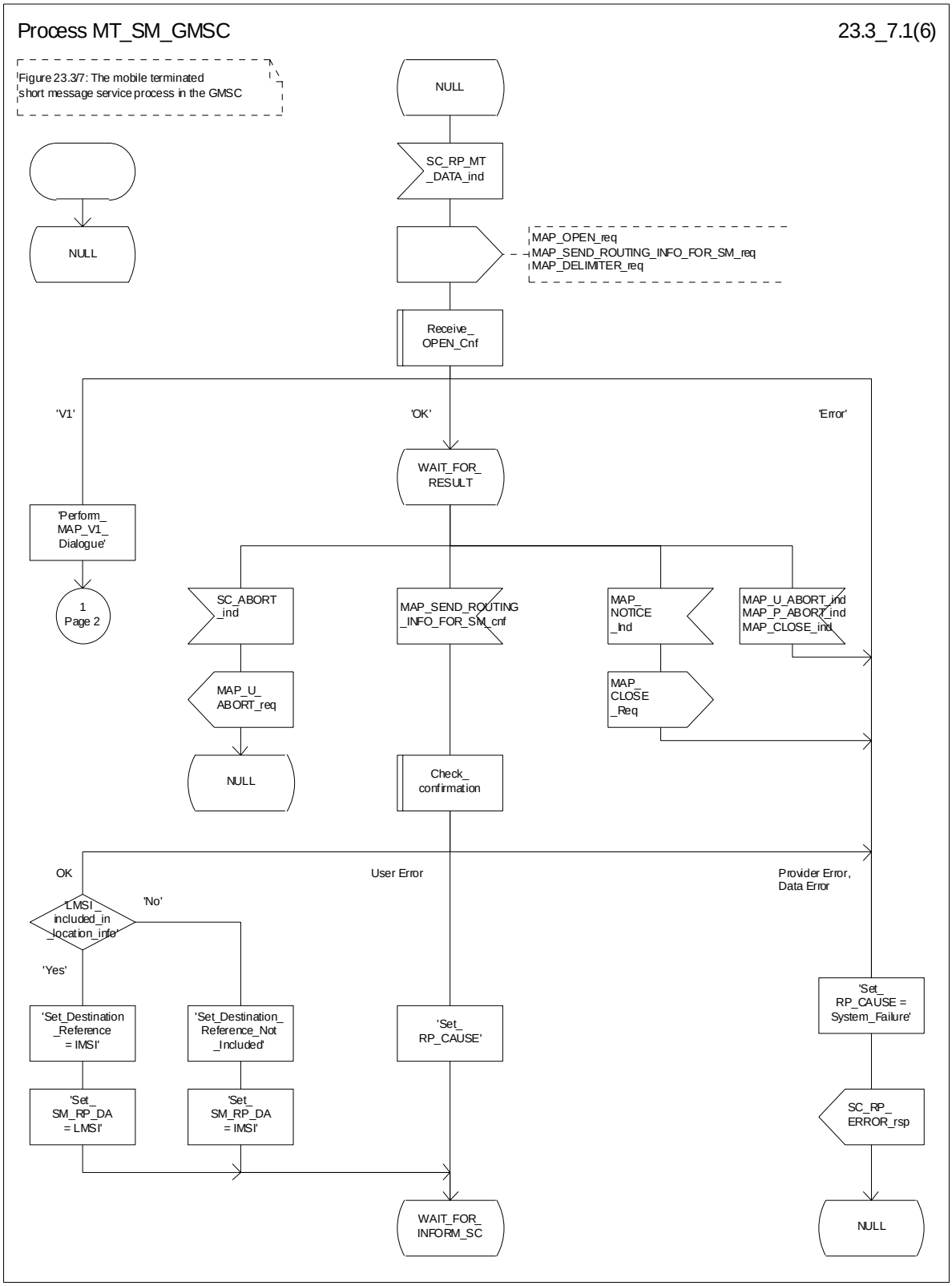


Figure 23.3/7 (sheet 1 of 6): Procedure MT_SM_GMSC

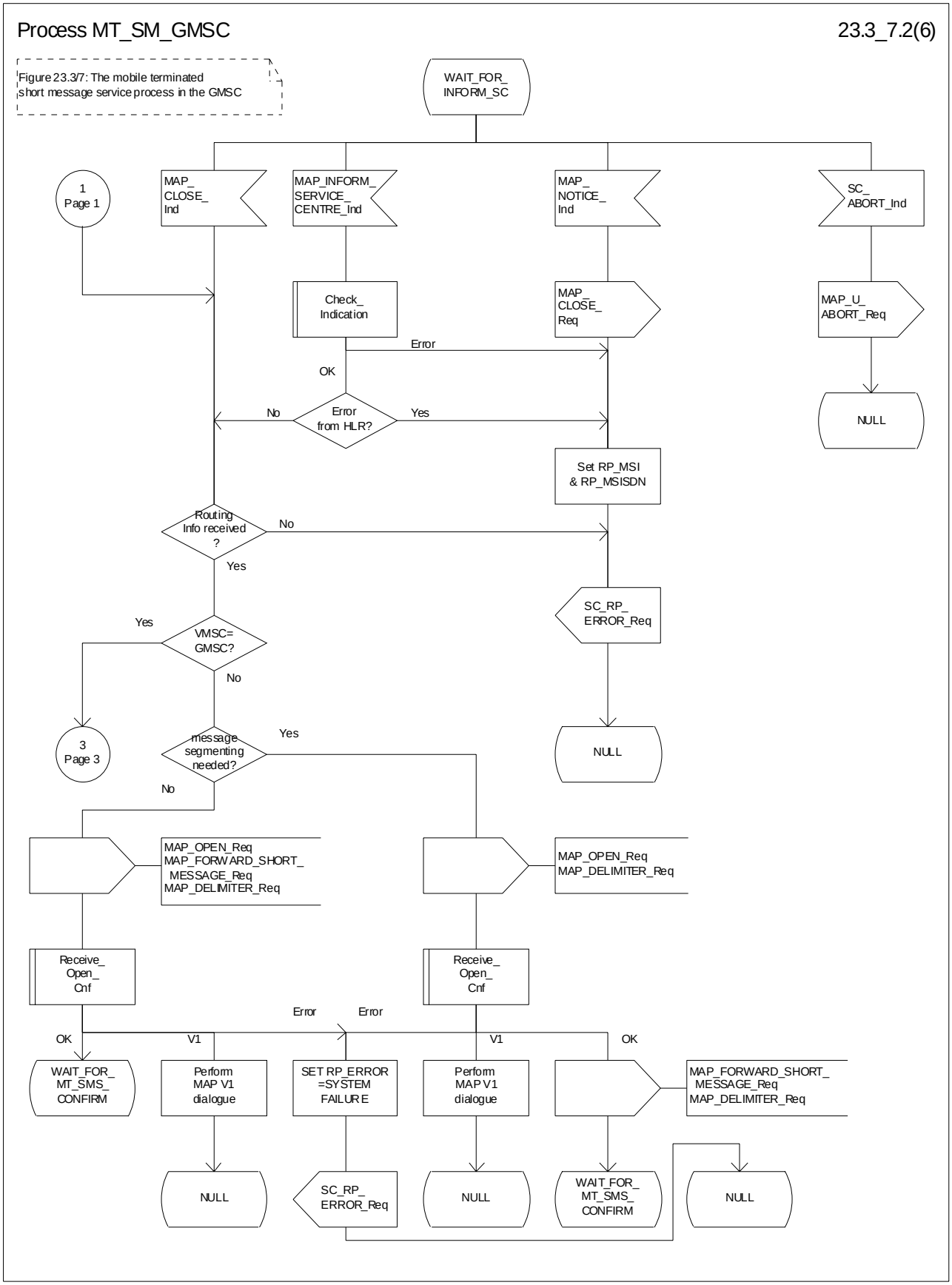


Figure 23.3/7 (sheet 2 to 6): Procedure MT_SM_GMSC

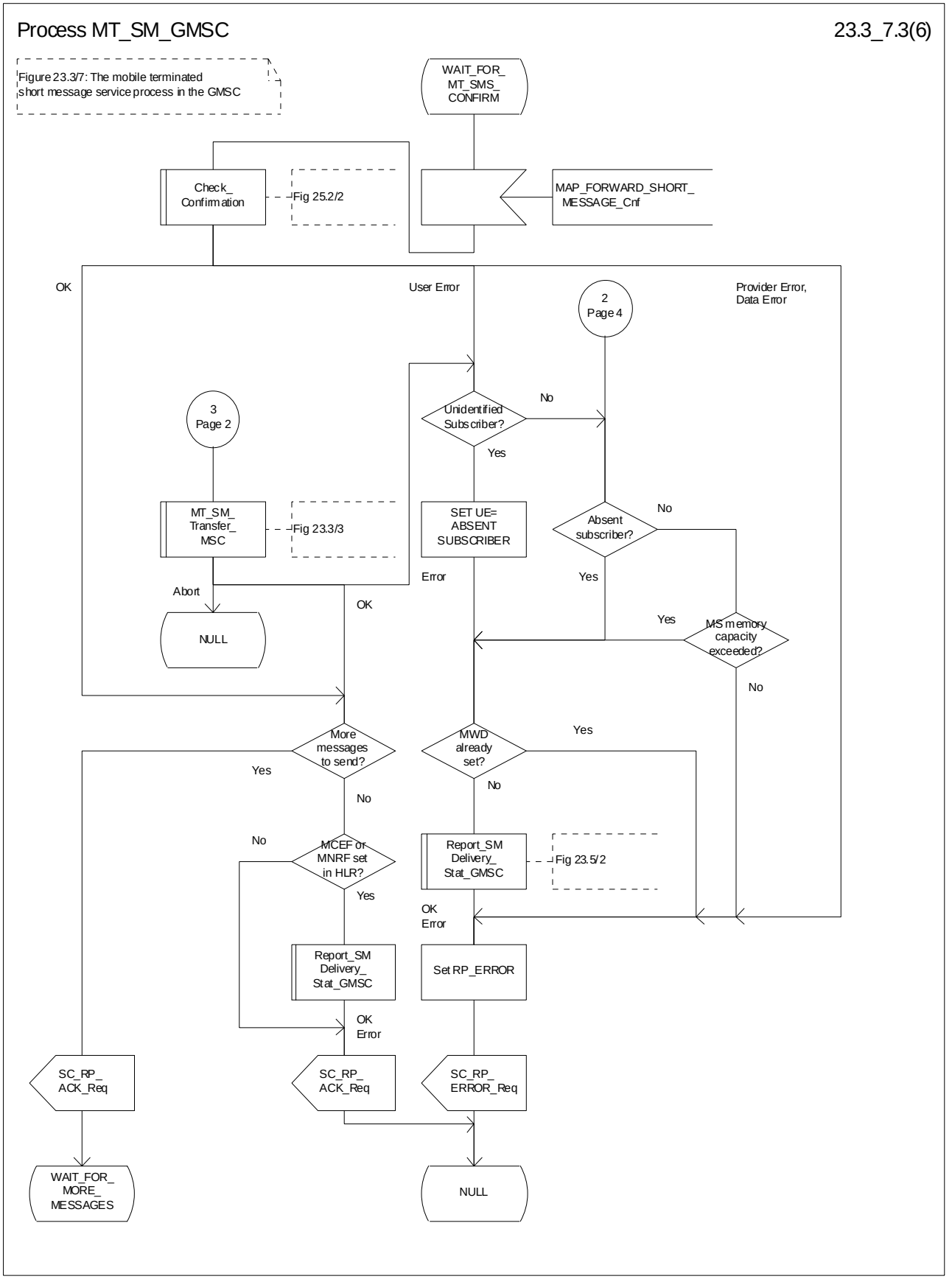


Figure 23.3/7 (sheet 3 of 6): Procedure MT_SM_GMSC

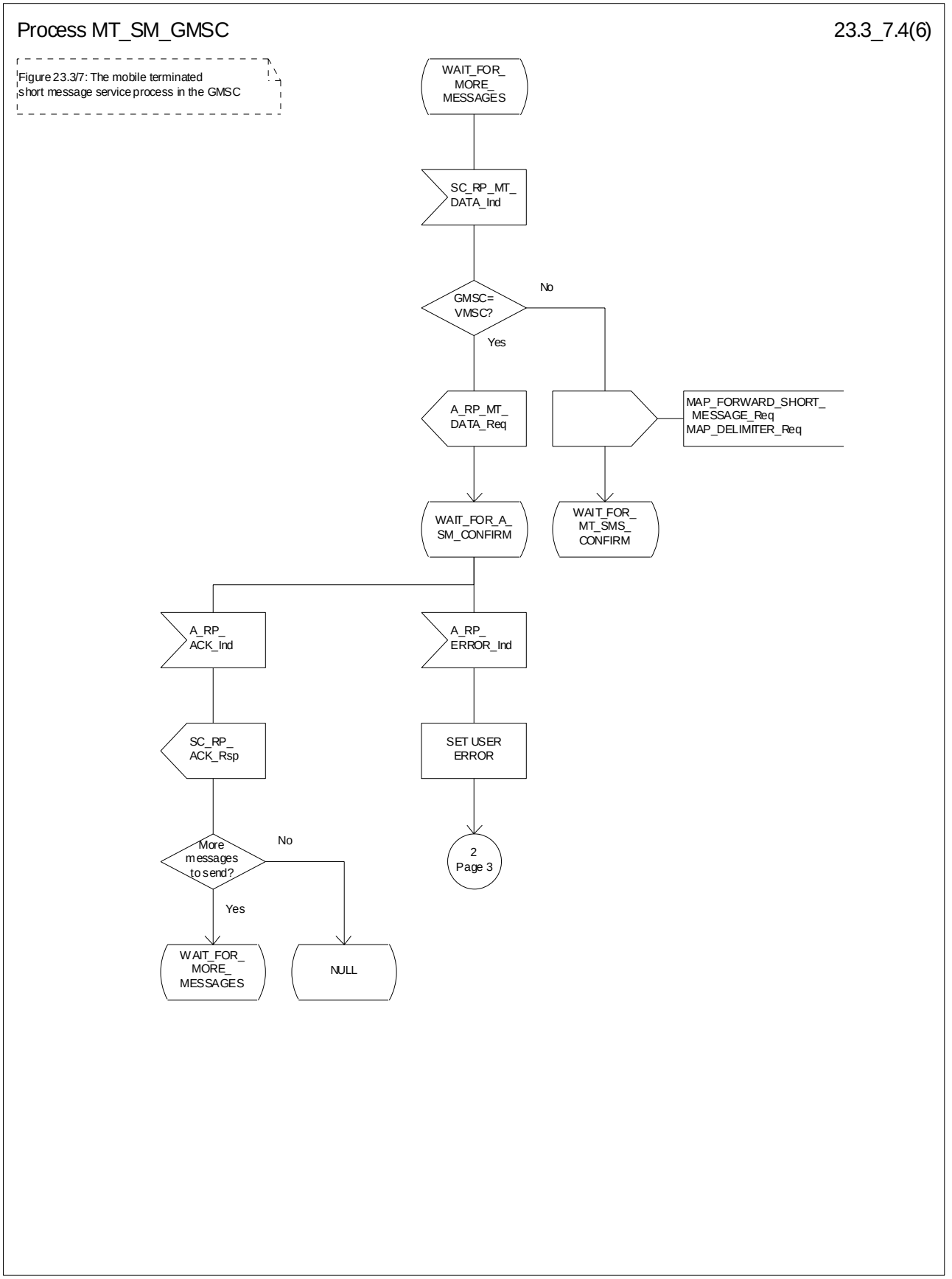


Figure 23.3/7 (sheet 4 of 6): Procedure_MT_SM_GMSC

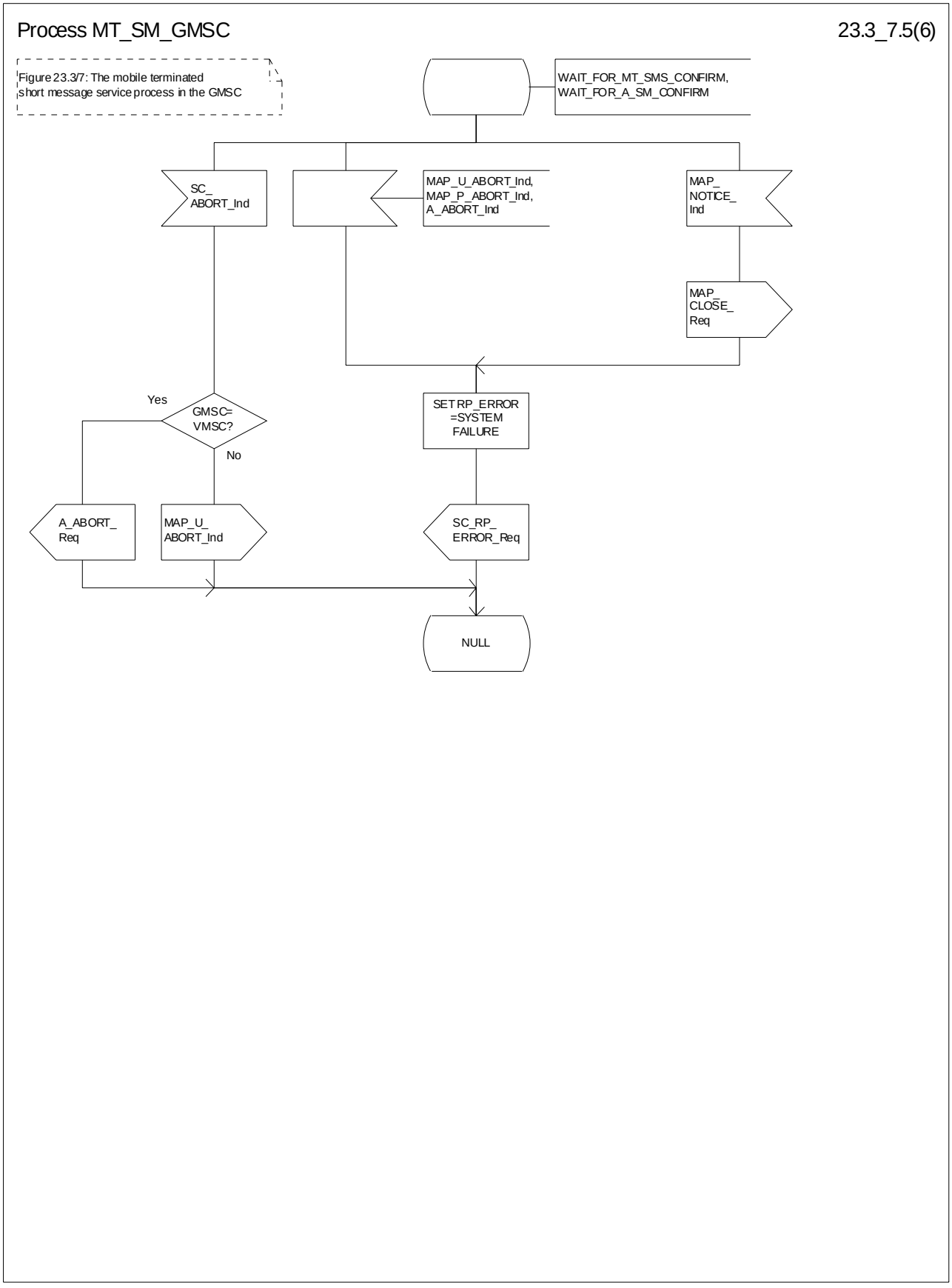


Figure 23.3/7 (sheet 5 to 6): Procedure MT_SM_GMSC

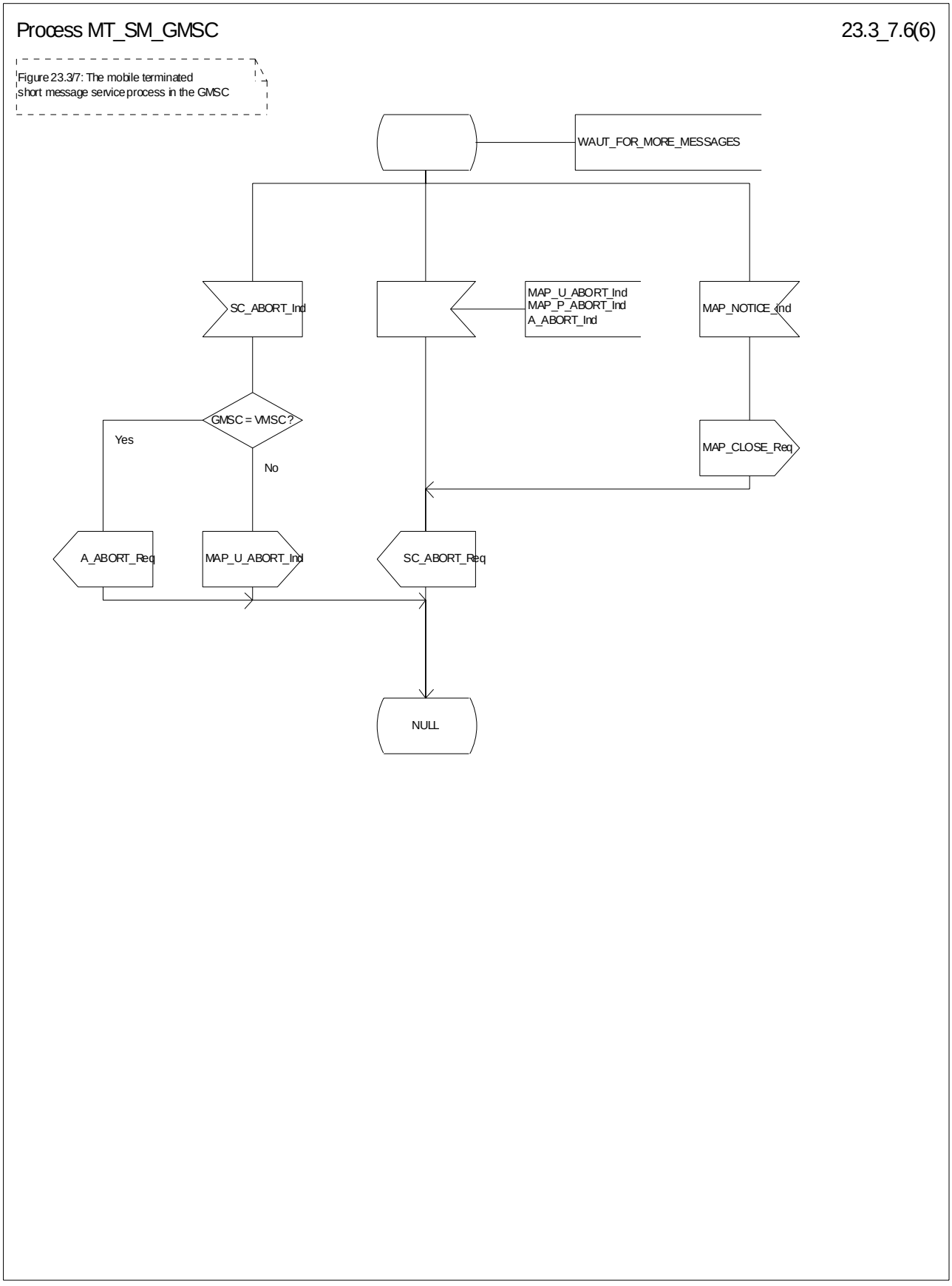


Figure 23.3/7 (sheet 6 of 6): Procedure MT_SM_GMSC

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

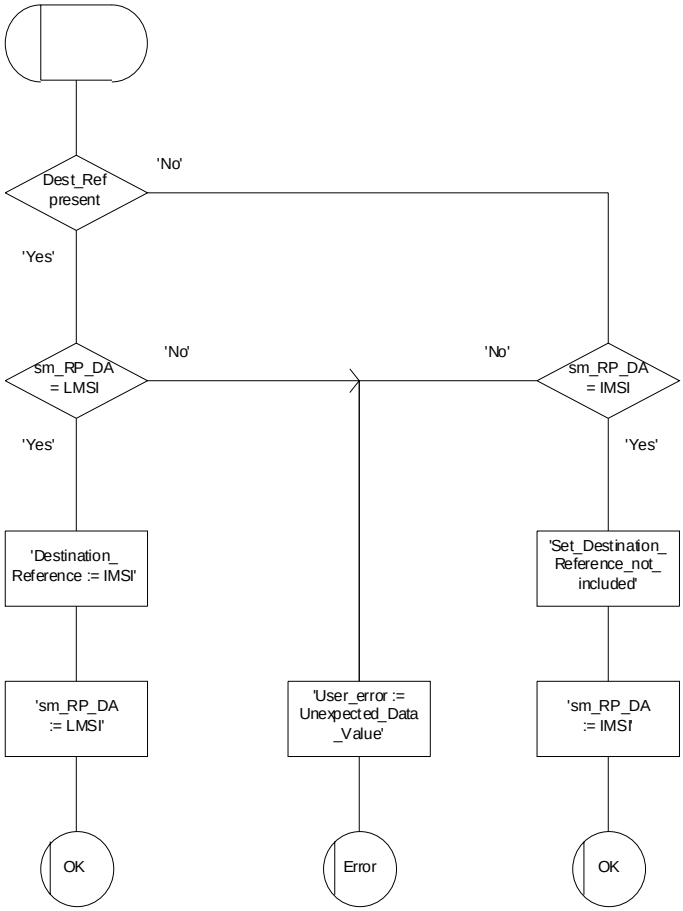


Figure 23.3/8: Macro Check_Subscr_Identity_For_MT_SMS

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 clause 25.3).Otherwise the search procedure is invoked (see clause 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 clauses 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 clause 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.

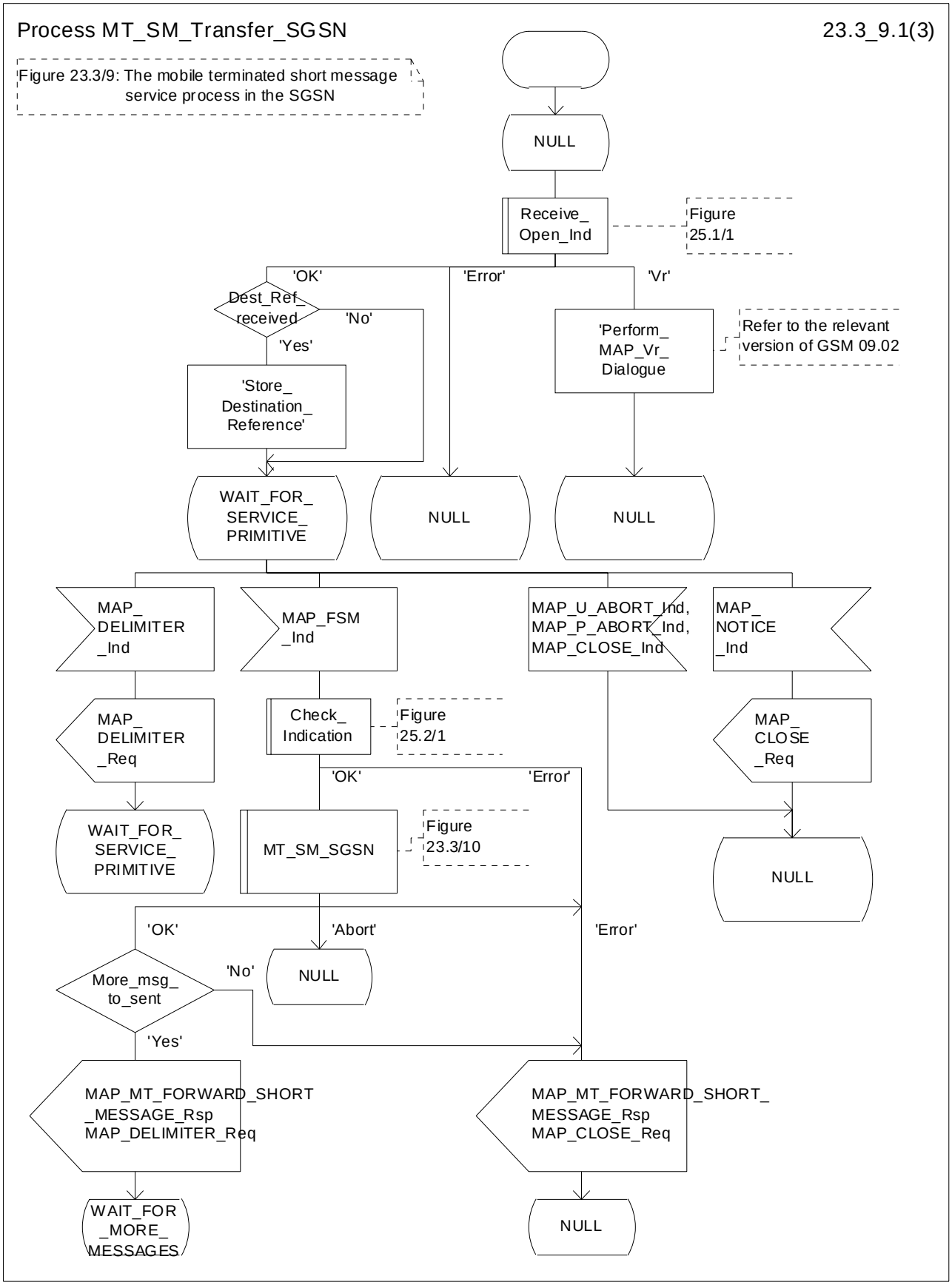


Figure 23.3/9 (sheet 1 of 3): Procedure MT_SM_Transfer_SGSN

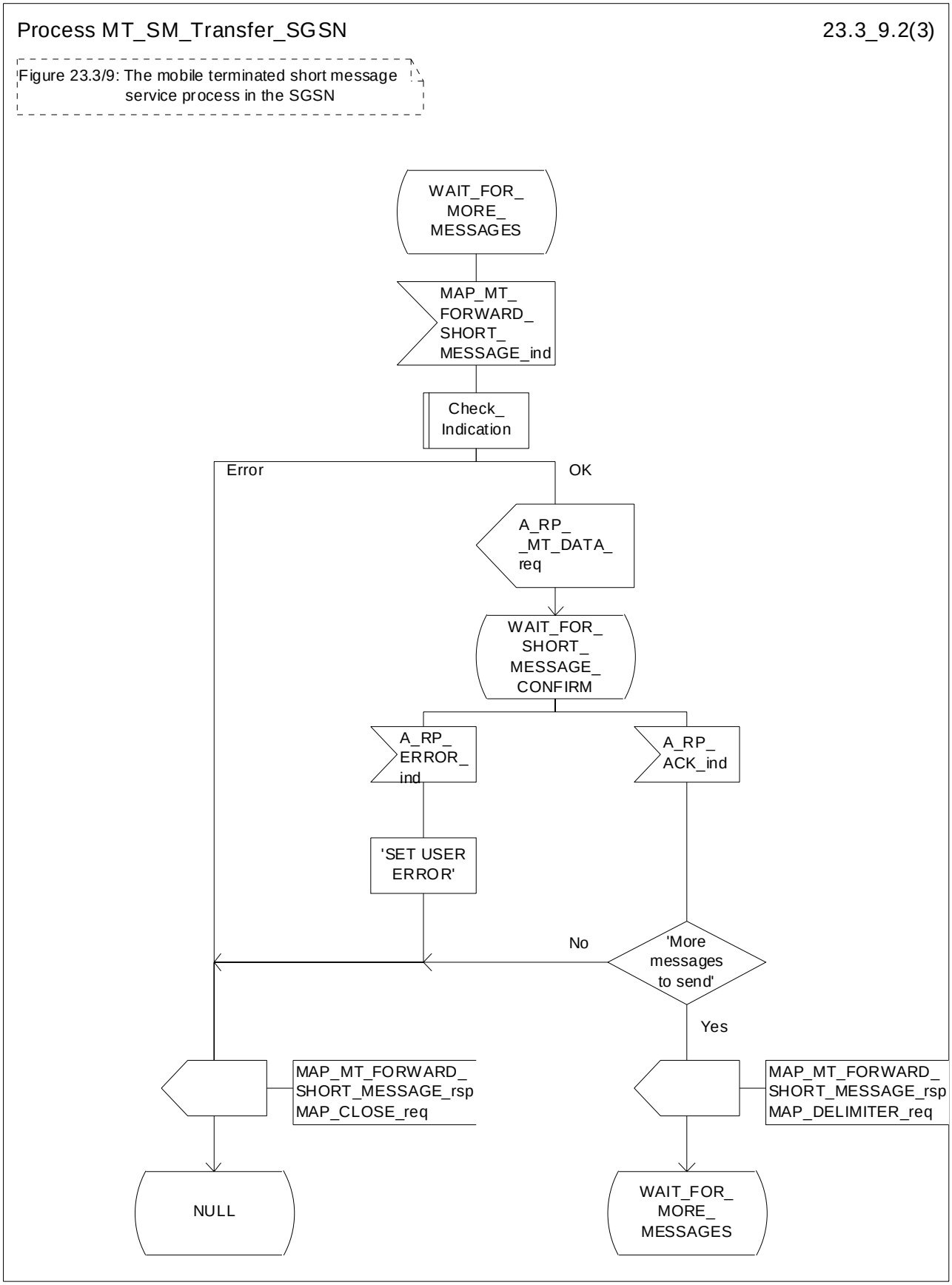


Figure 23.3/9 (sheet 2 of 3): Procedure MT_SM_Transfer_SGSN

Process MT_SM_Transfer_SGSN 23.3_9.3(3)

Figure 23.3/9: The mobile terminated short message service process in the SGSN

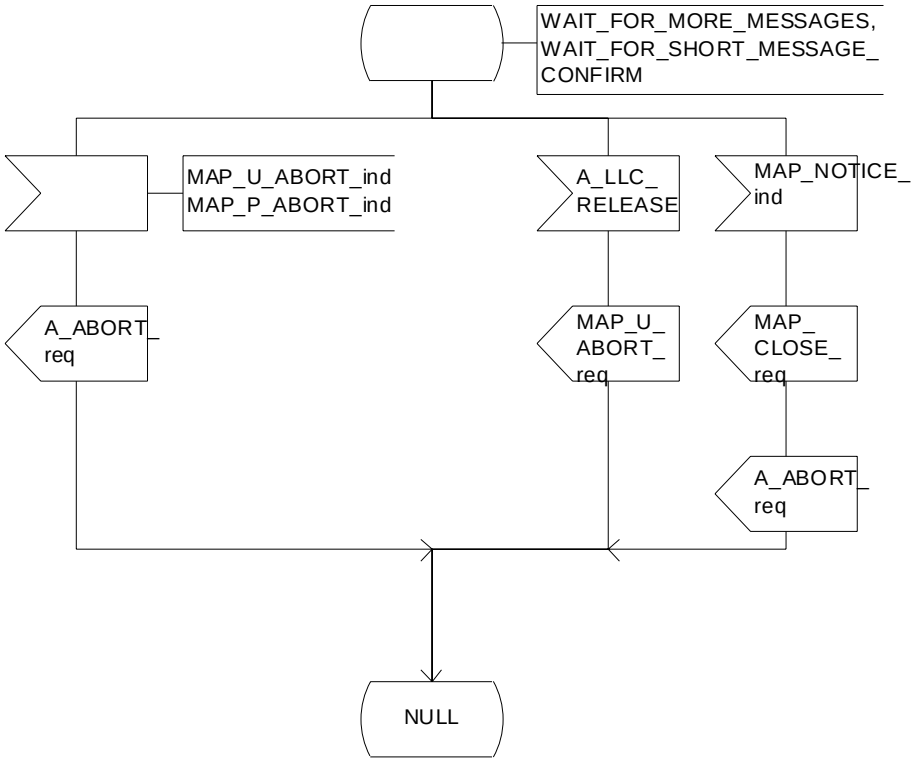


Figure 23.3/9 (sheet 3 of 3): Procedure MT_SM_Transfer_SGSN

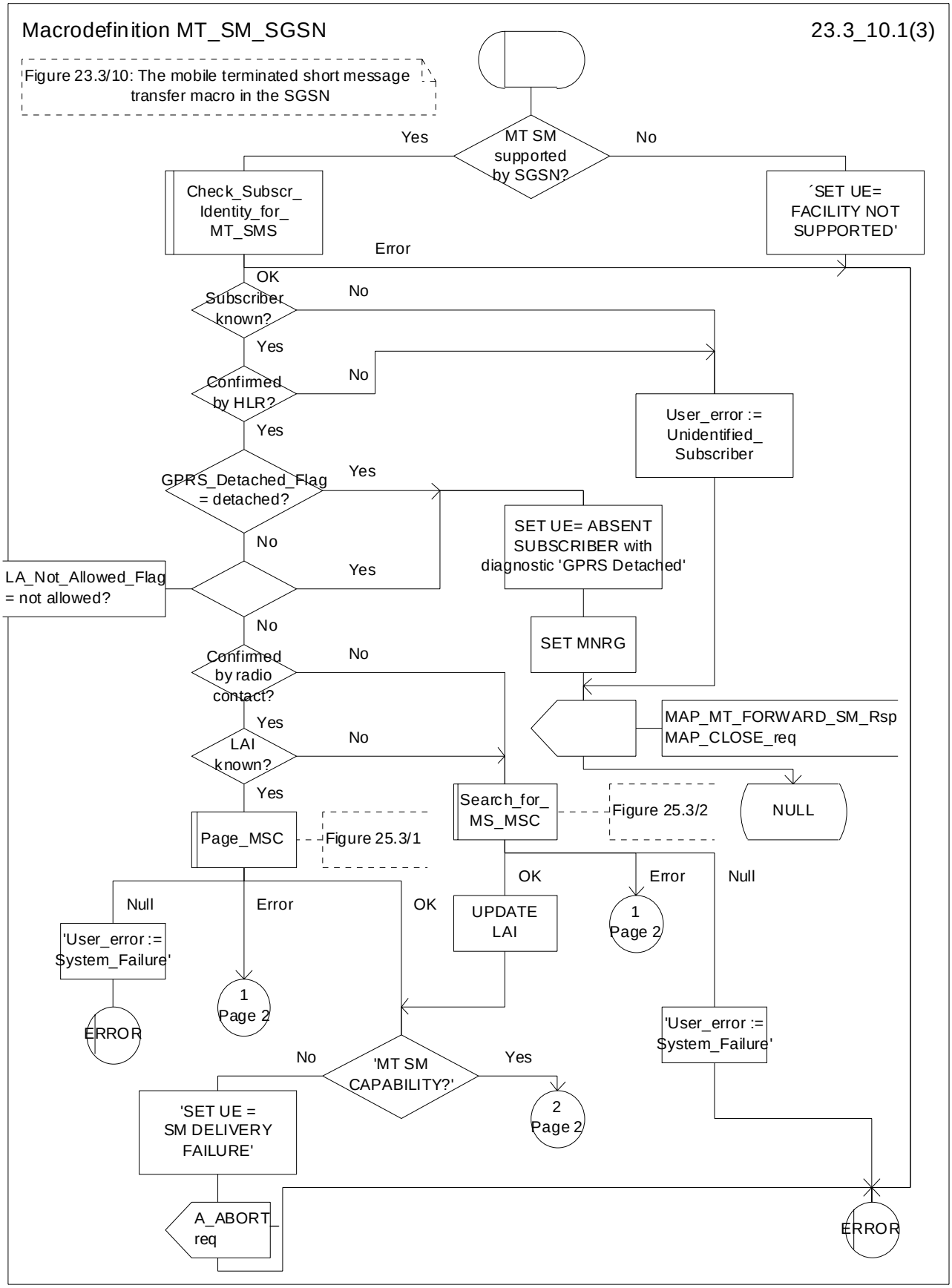


Figure 23.3/10 (sheet 1 of 3): Macro MT_SM_SGSN

Macrodefinition MT_SM_SGSN 23.3_10.2(3)

Figure 23.3/10: The mobile terminated short message transfer macro in the SGSN

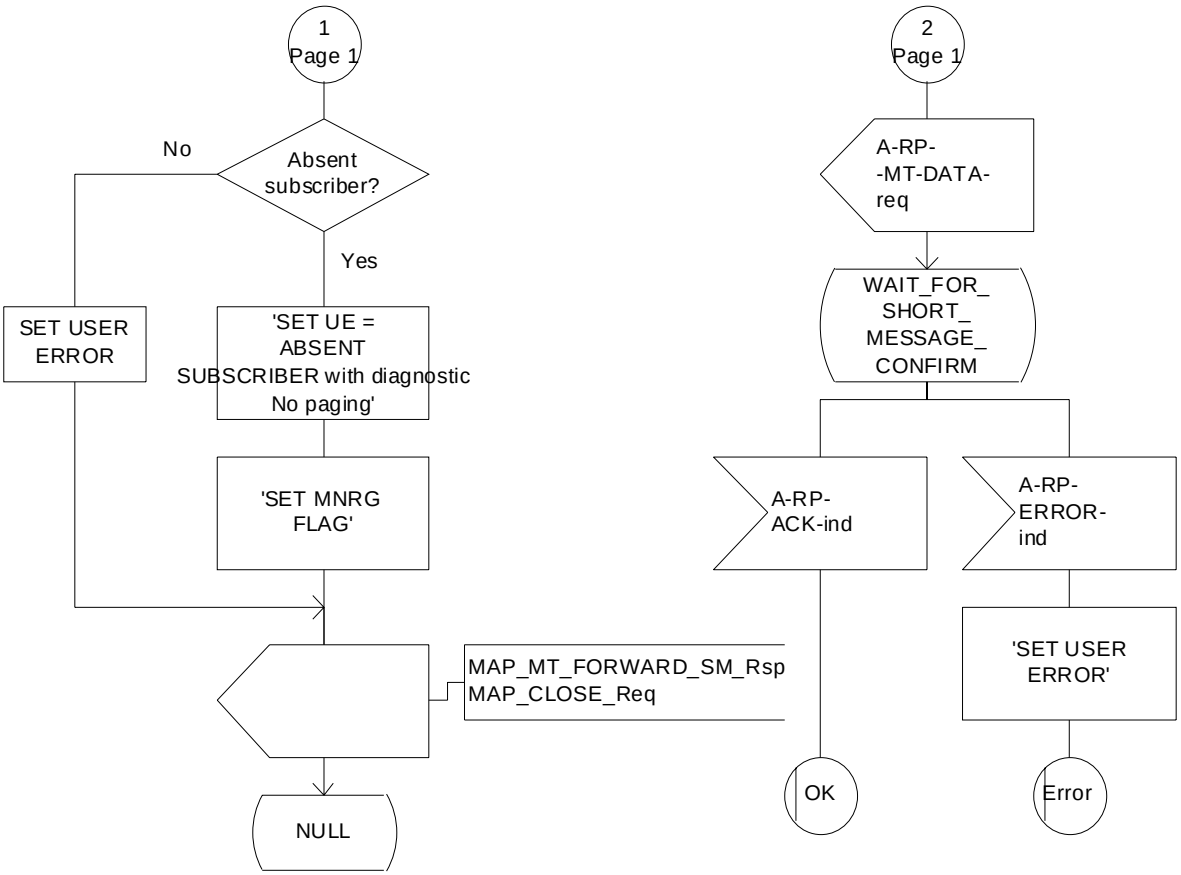


Figure 23.3/10 (sheet 2 of 3): Macro MT_SM_SGSN

Macrodefinition MT_SM_SGSN 23.3_10.3(3)

Figure 23.3/10: The mobile terminated short message transfer macro in the SGSN

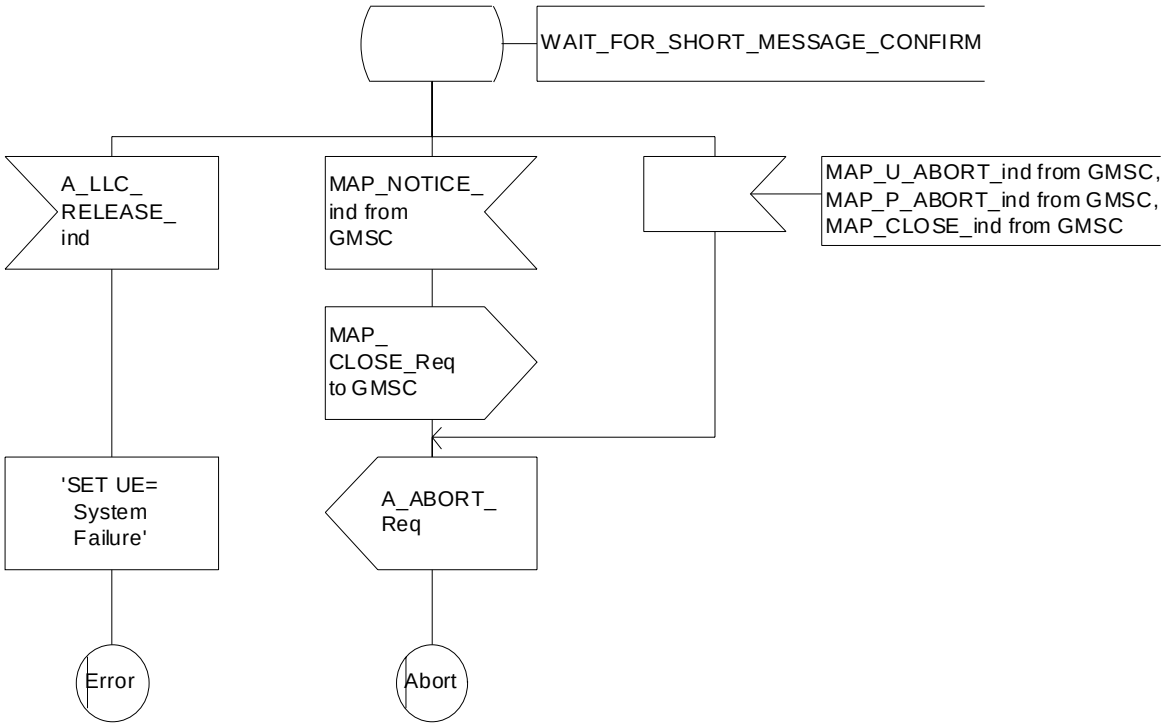


Figure 23.3/10 (sheet 3 of 3): Macro MT_SM_SGSN

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.

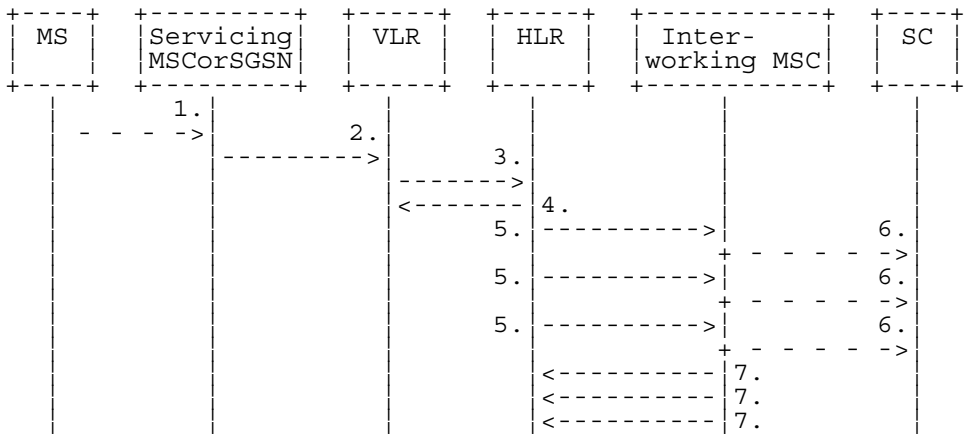


Figure 23.4/1: Short message alert procedure (Mobile is present)

1) CM Service Request (**), Page response or Location Updating (3GPP TS 24.008).
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 (3GPP TS 23.040[26]).
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.

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.

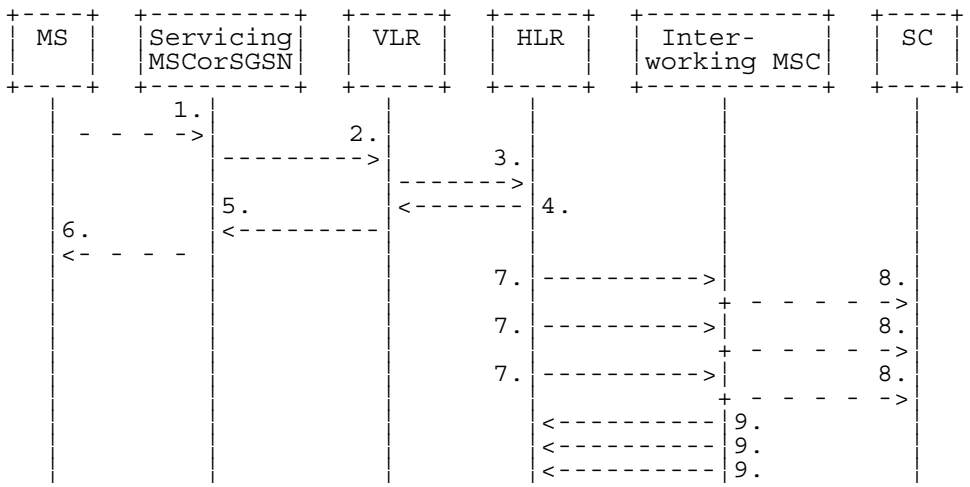


Figure 23.4/2: Short message alert procedure (MS memory capacity available)

1) SM memory capacity available (3GPP TS 24.011 [37]).

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) (3GPP TS 24.011 [37]).

7) MAP_ALERT_SERVICE_CENTRE (note).

8) Alert Service Centre (3GPP TS 23.040[26]).

9) MAP_ALERT_SERVICE_CENTRE_ACK.

NOTE: 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.

In addition the following MAP services are used in the MS memory available case:

MAP_PROCESS_ACCESS_REQUEST	(see clause 8.3); (*)
MAP_AUTHENTICATE	(see clause 8.5); (*)
MAP_SET_CIPHERING_MODE	(see clause 8.6); (*)
MAP_PROVIDE_IMSI	(see clause 8.9); (*)
MAP_CHECK_IMEI	(see clause 8.7);
MAP_FORWARD_NEW_TMSI	(see clause 8.9); (*)
MAP_TRACE_SUBSCRIBER_ACTIVITY	(see clause 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.

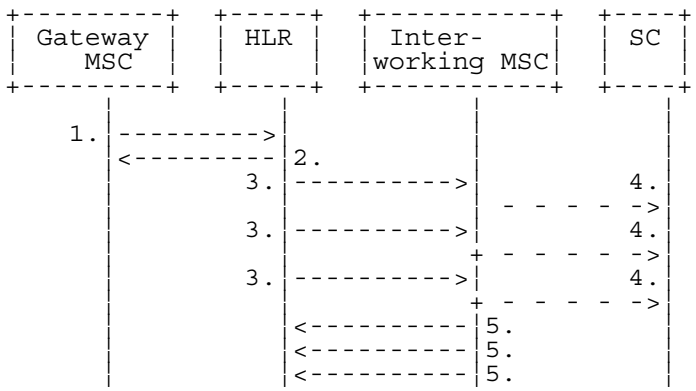


Figure 23.4/3: Short message alert procedure (Successful transfer after polling)

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 (3GPP TS 23.040[26]).

5) MAP_ALERT_SERVICE_CENTRE_ACK.

NOTE: To all Service Centres in the Message Waiting List.

23.4.1 Procedures in the Servicing MSC

The activation of the MAP_PROCESS_ACCESS_REQUEST service is described in the clause 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 acknowledgement. The MSC sends the corresponding message to the MS;
- negative acknowledgement, 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.

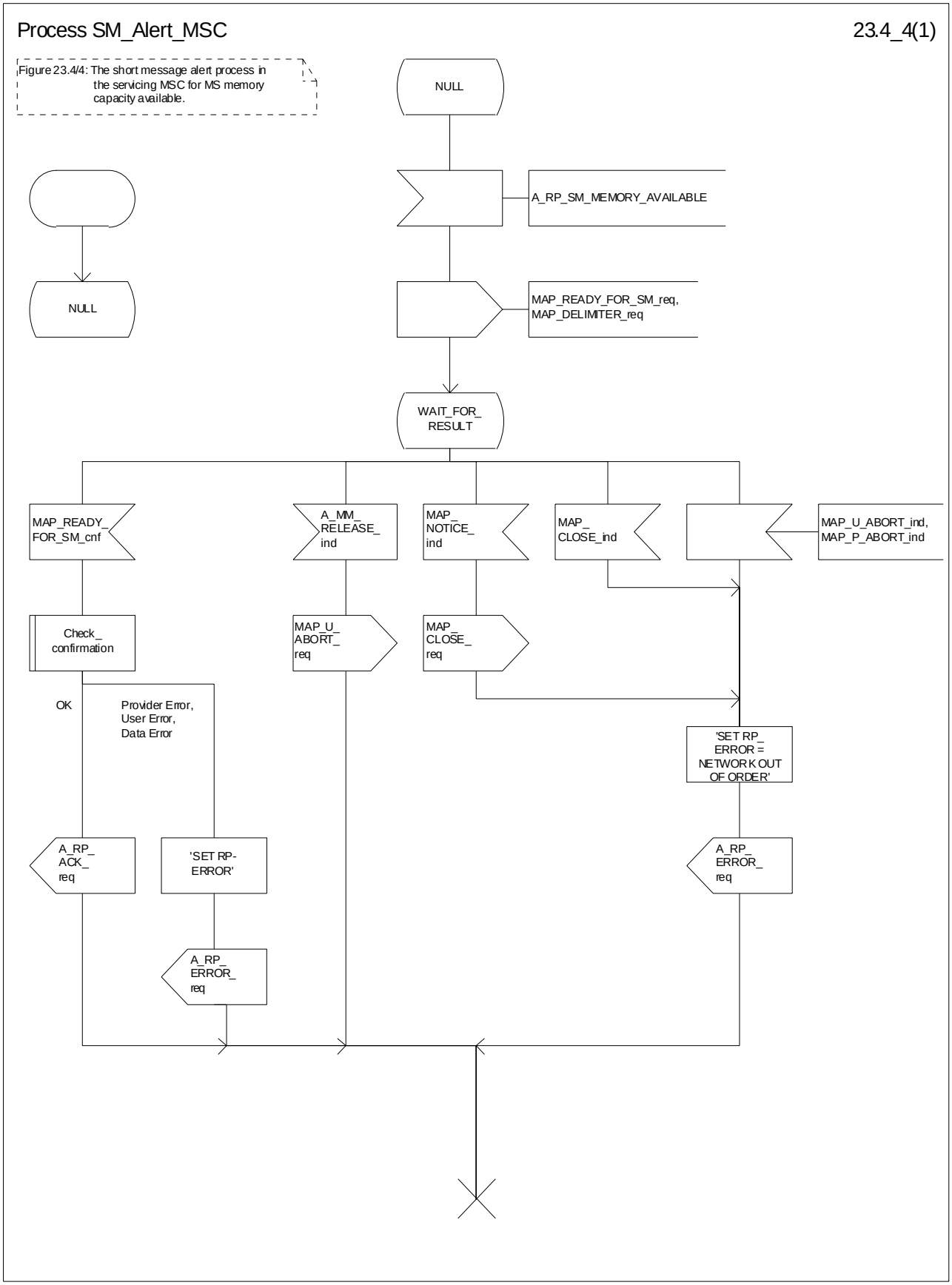


Figure 23.4/4: Procedure SM_Alert_MSC

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 clause 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 acknowledgement.

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 acknowledgement 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.

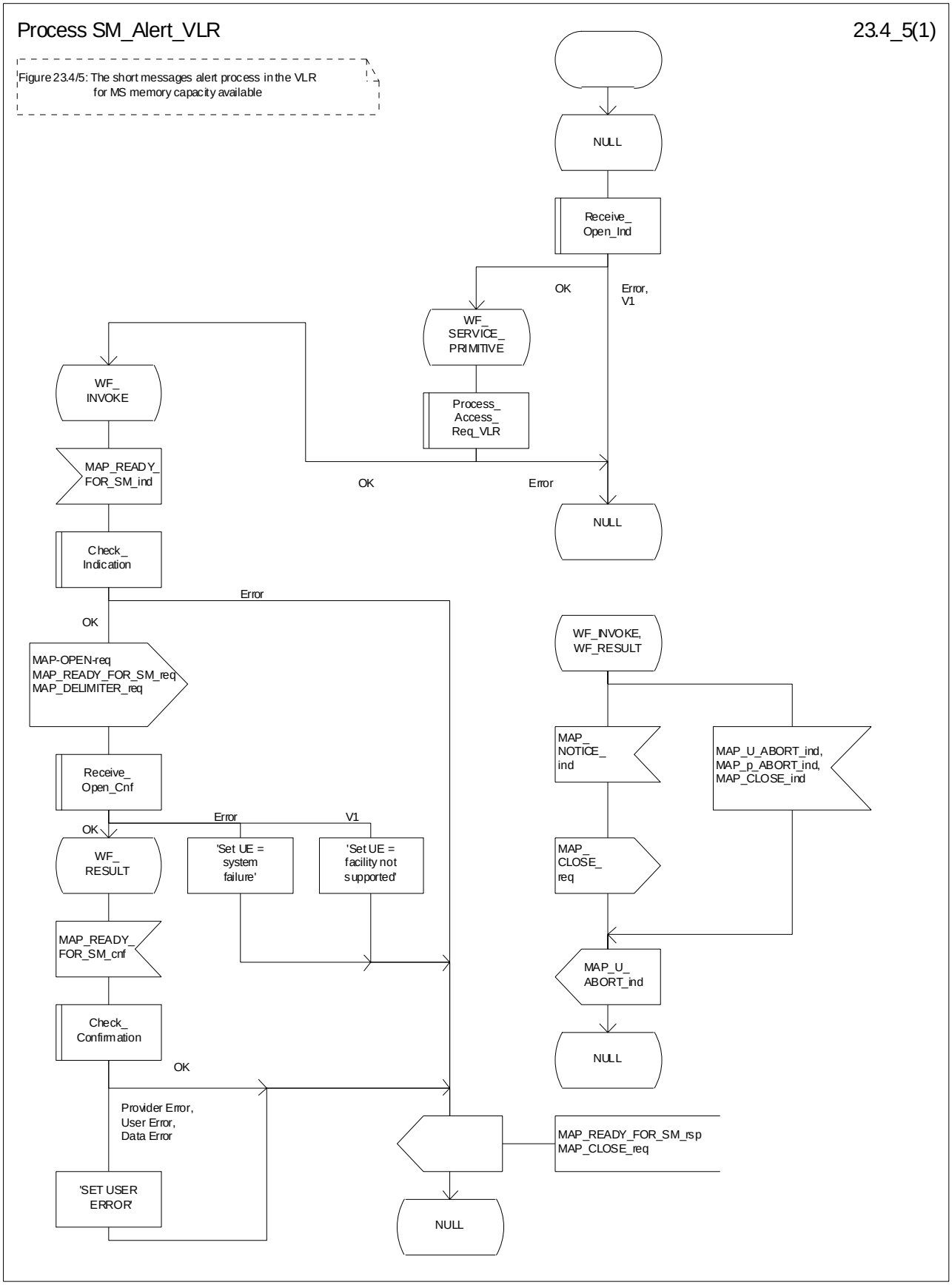


Figure 23.4/5: Procedure SM_Alert_VLR

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 clause 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 GPRS location 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 clause 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 clause 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.

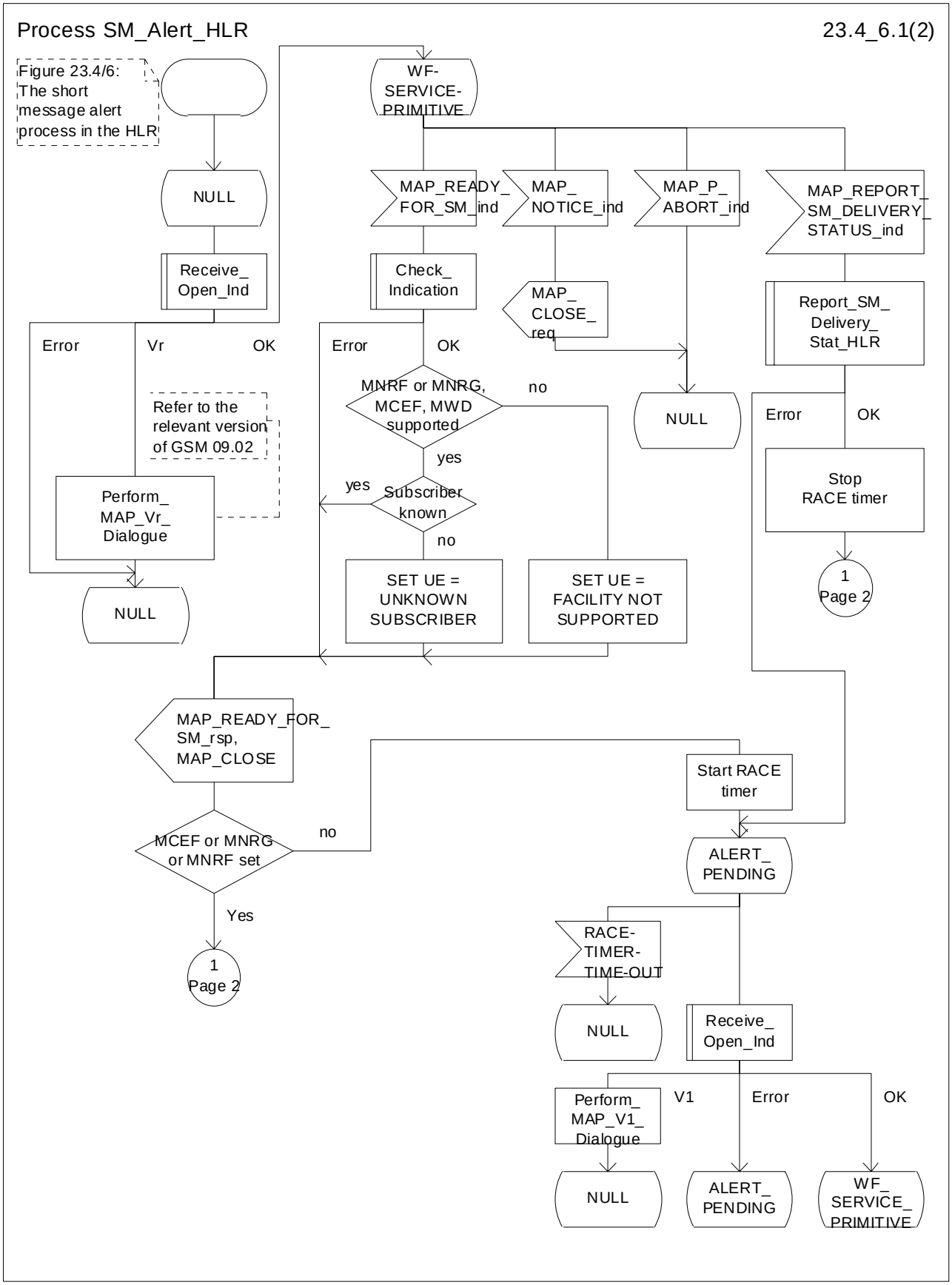


Figure 23.4/6 (sheet 1 of 2): Process SM_Alert_HLR

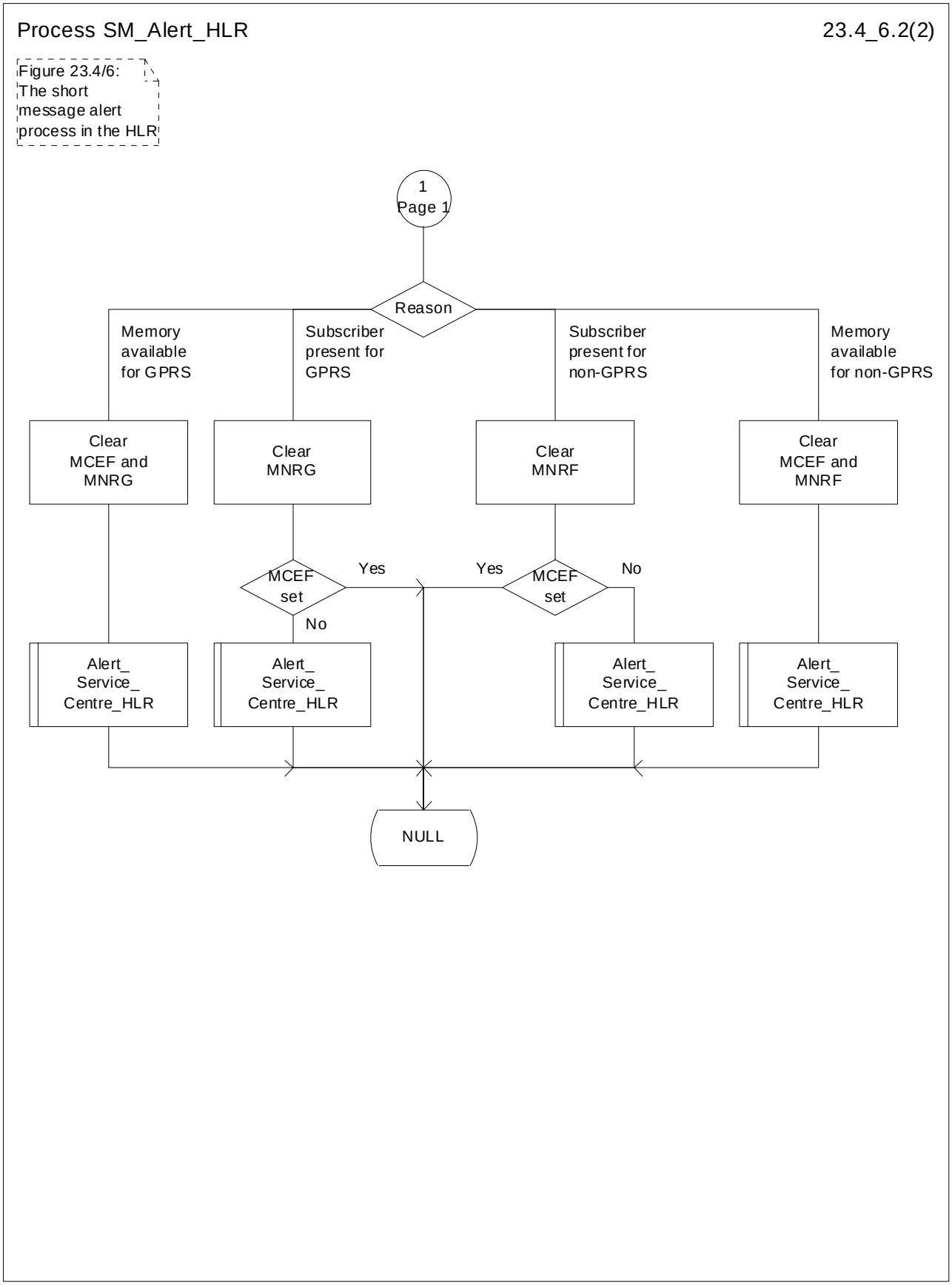


Figure 23.4/6 (sheet 2 of 2): Process SM_Alert_HLR

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.

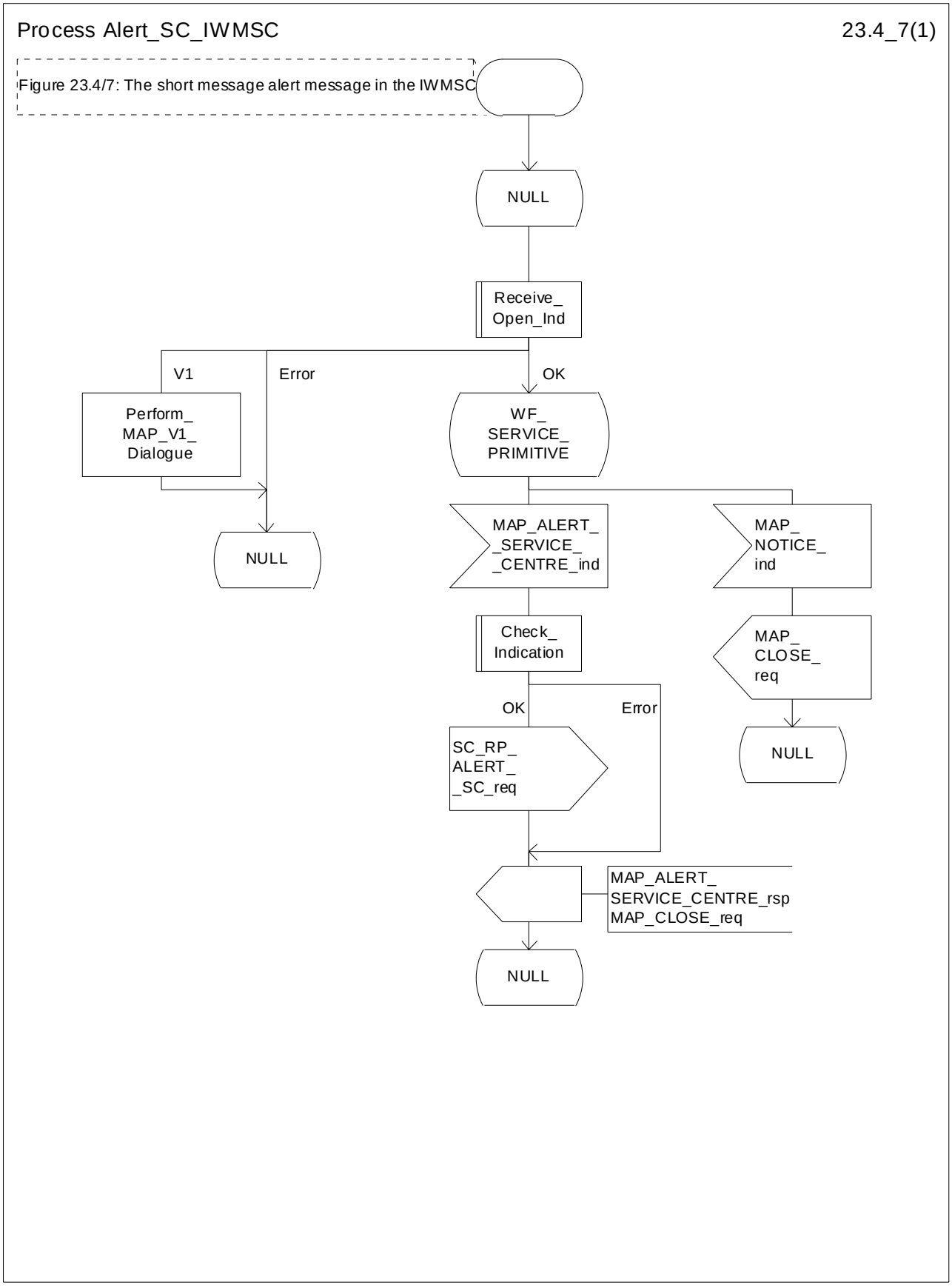


Figure 23.4/7: Process Alert_SC_IWMSC

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 (3GPP TS 24.008), 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 clause 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 acknowledgement. The SGSN sends the corresponding message to the MS;
- negative acknowledgement, 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.

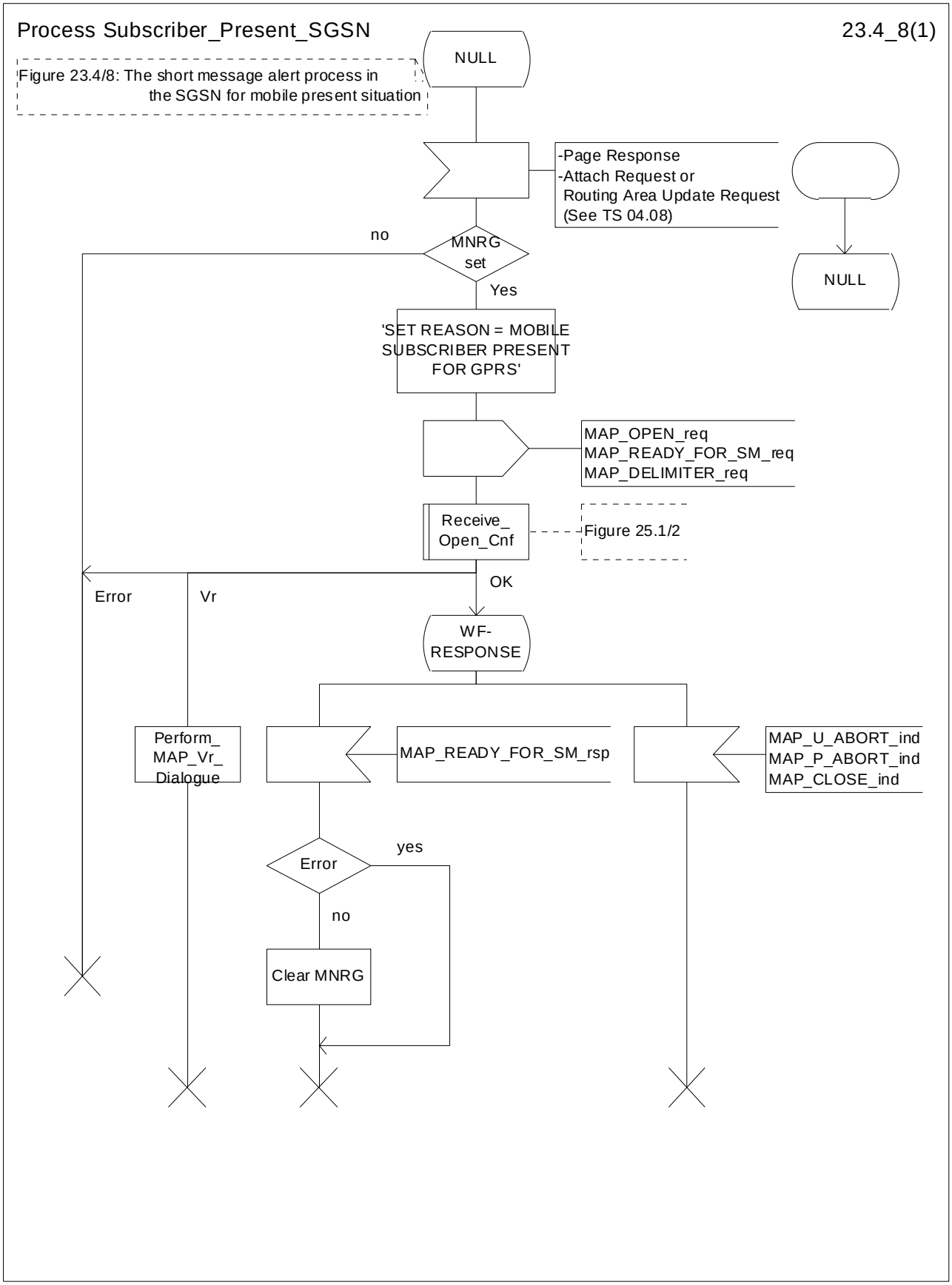


Figure 23.4/8: Process Subscriber_Present_SGSN

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 clause 23.4.

The SM delivery status report procedure is shown in figure 23.5/1.

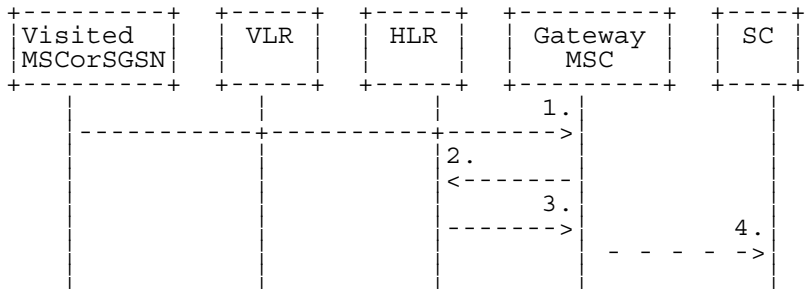


Figure 23.5/1: Short message delivery status report procedure

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 (3GPP TS 23.040[26]).

23.5.1 Procedures in the HLR

When the HLR receives a MAP_REPORT_SM_DELIVERY_STATUS indication, it acts as described in the clause 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.

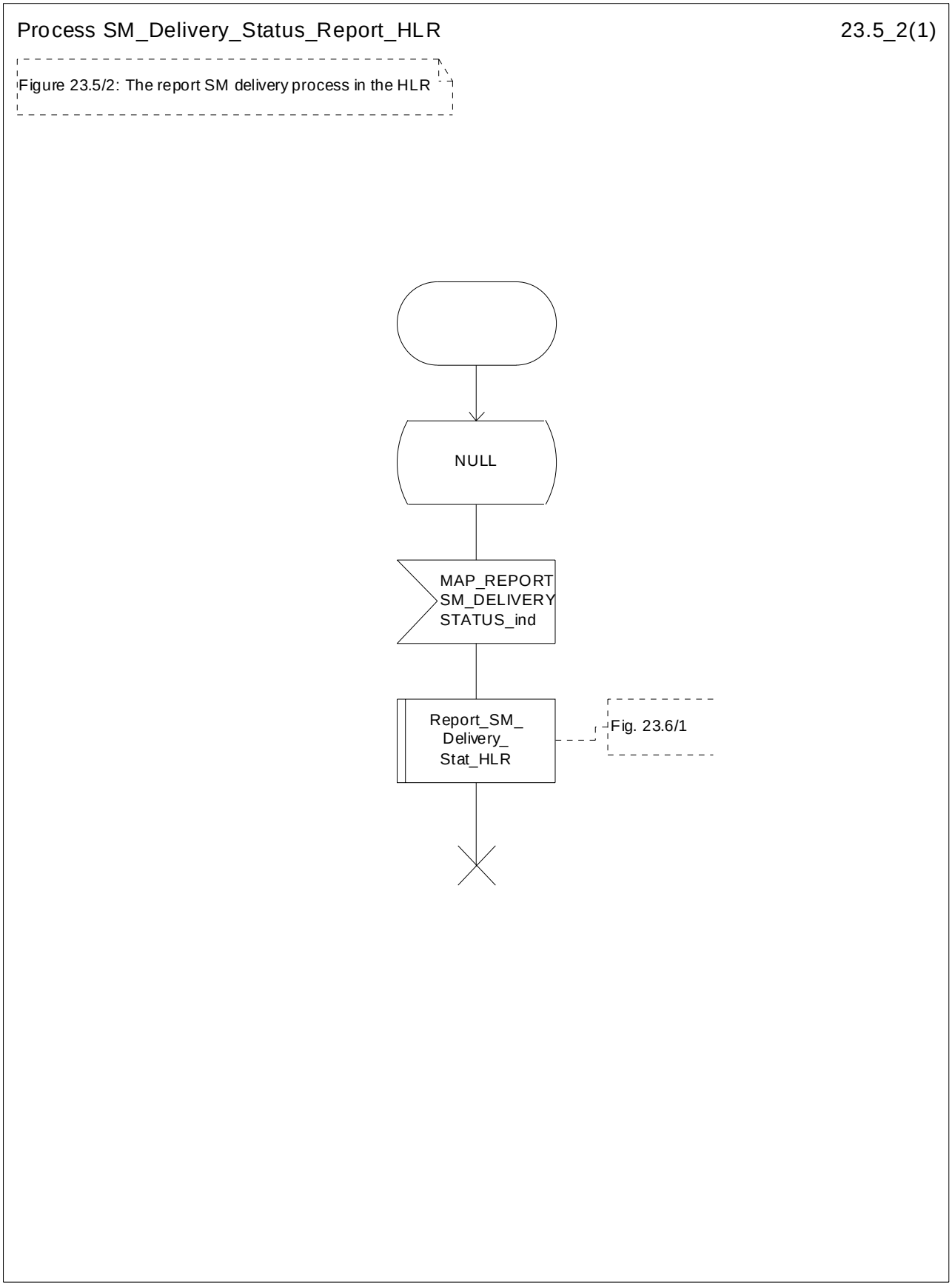


Figure 23.5/2: Process SM_Delivery_Status_Report_HLR

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 clause 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.

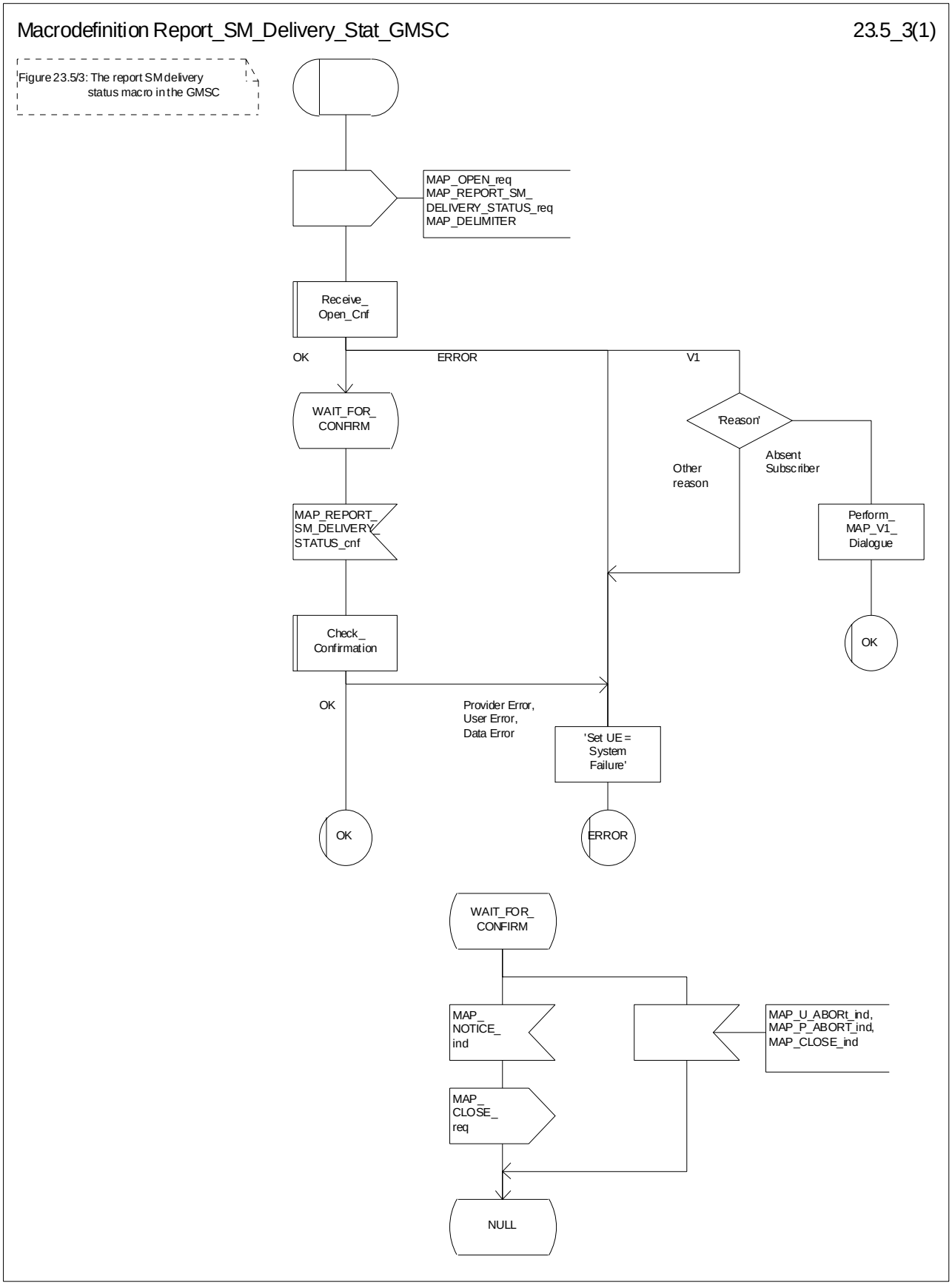


Figure 23.5/3: Macro Report_SM_Delivery_Stat_GMSC

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 responses 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 recognised 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 clause 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.

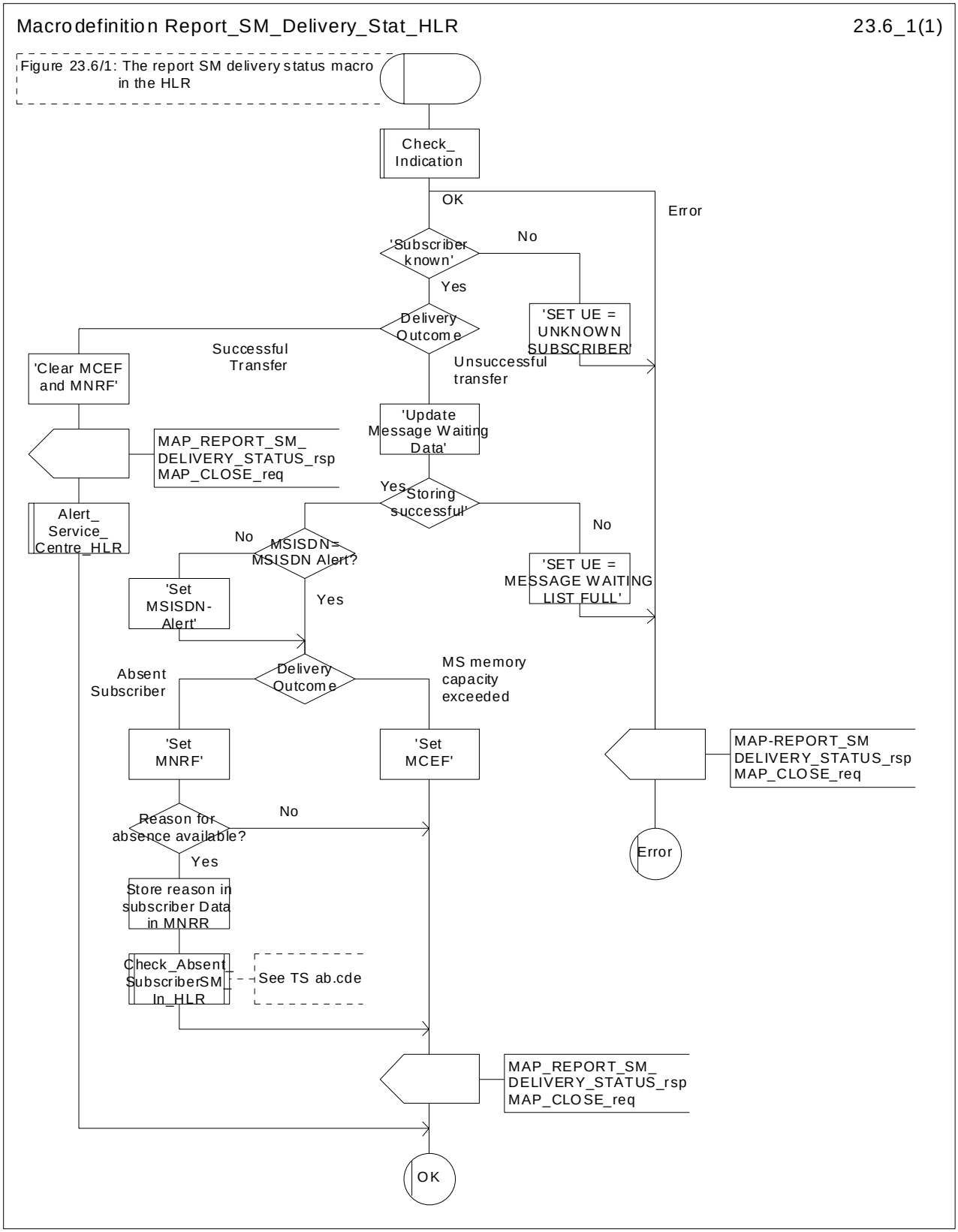


Figure 23.6/1: Macro Report_SM_Delivery_Stat_HLR

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 3GPP TS 23.060[104] [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 macros not defined in this clause; the definition of these macros can be found as follows:

Receive_Open_Ind see clause 25.1.1;

Check_Indication see clause 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 waits 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.

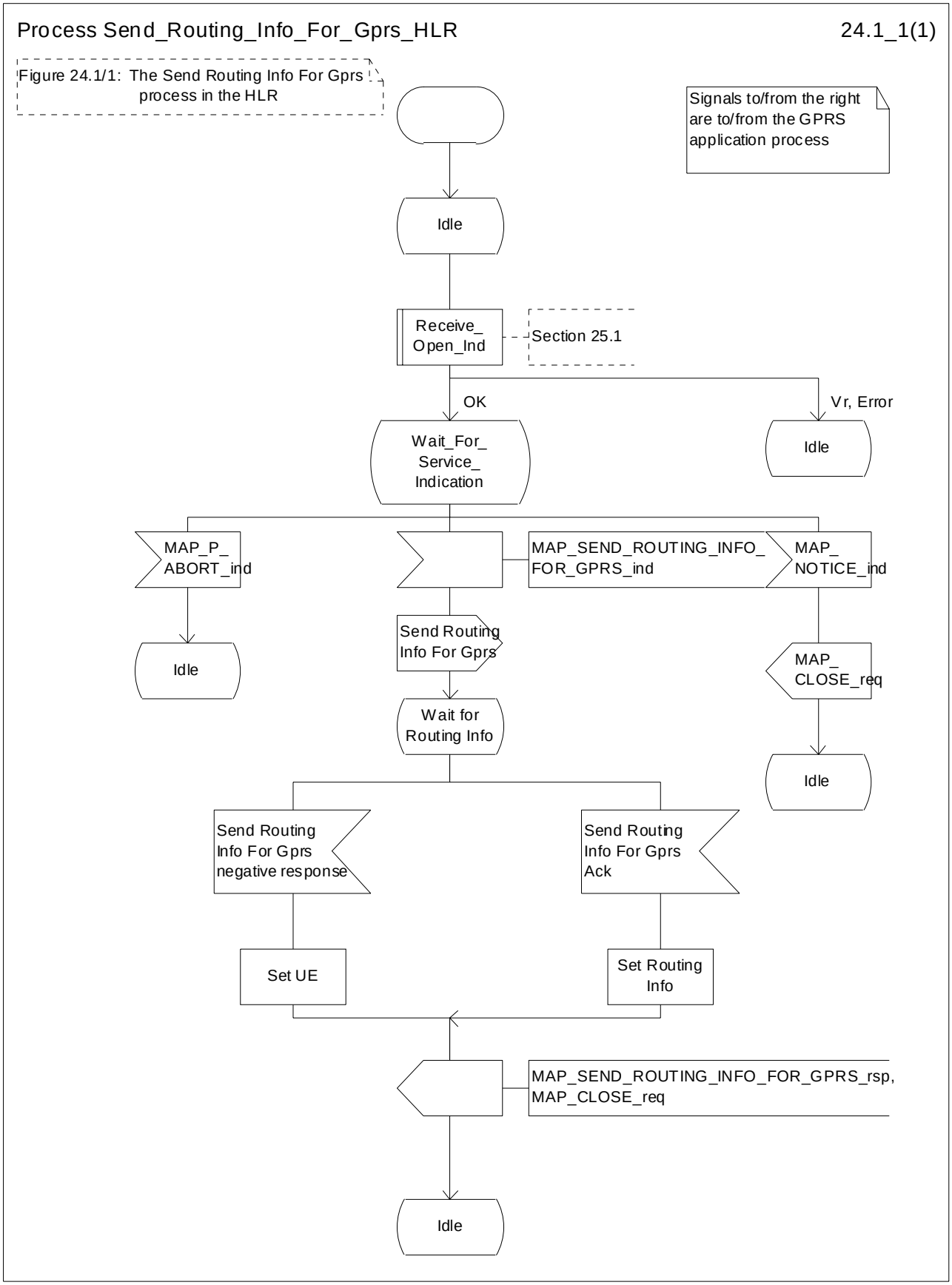


Figure 24.1/1: Process Send Routing Info For Gprs_HLR

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 routeing 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.

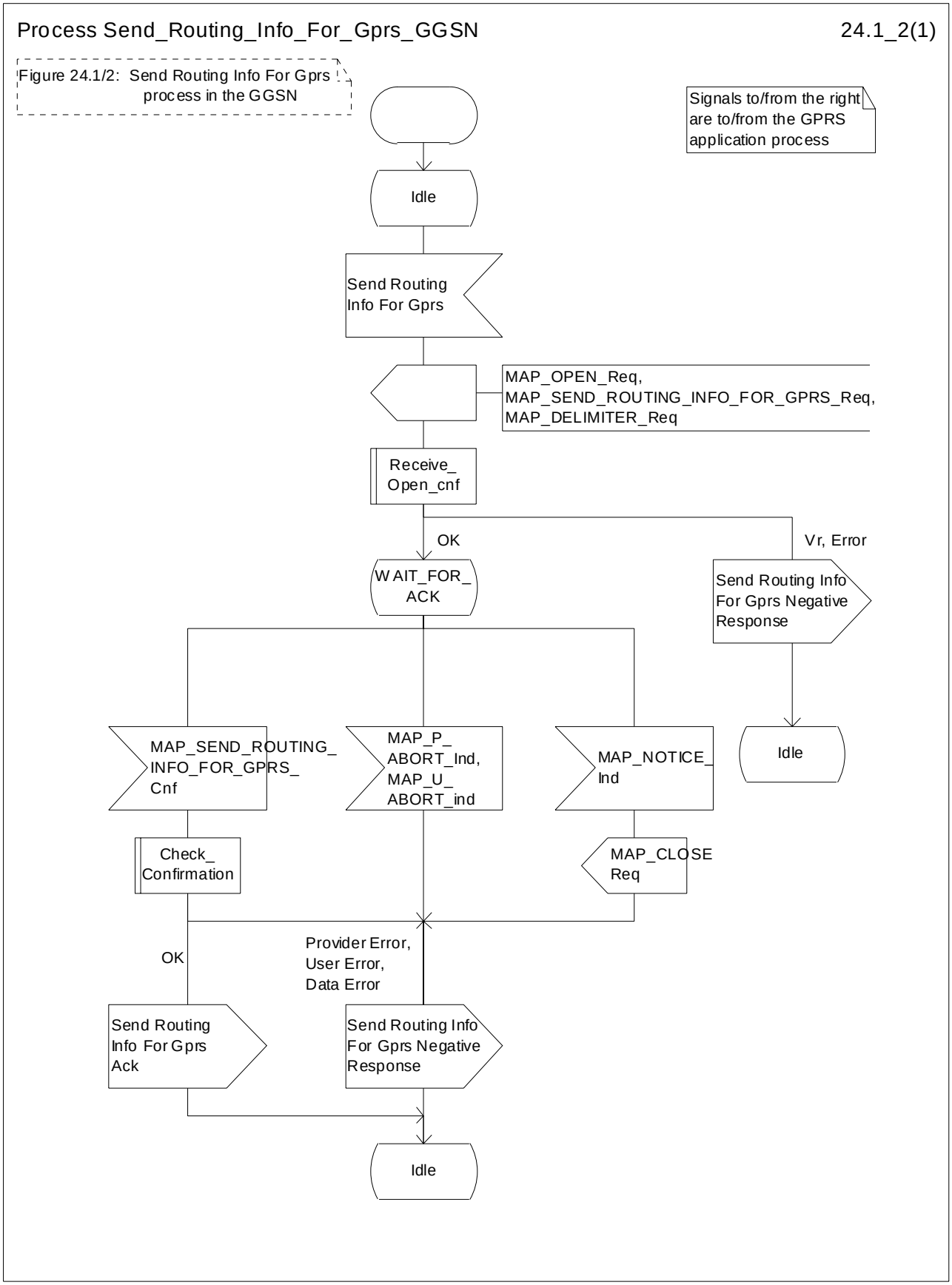


Figure 24.1/2: Process Send_Routing_Info_For_Gprs_GGSN

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 macros not defined in this clause; the definition of these macros can be found as follows:

Receive_Open_Ind	see clause 25.1.1;
Check Indication	see clause 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 waits 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.

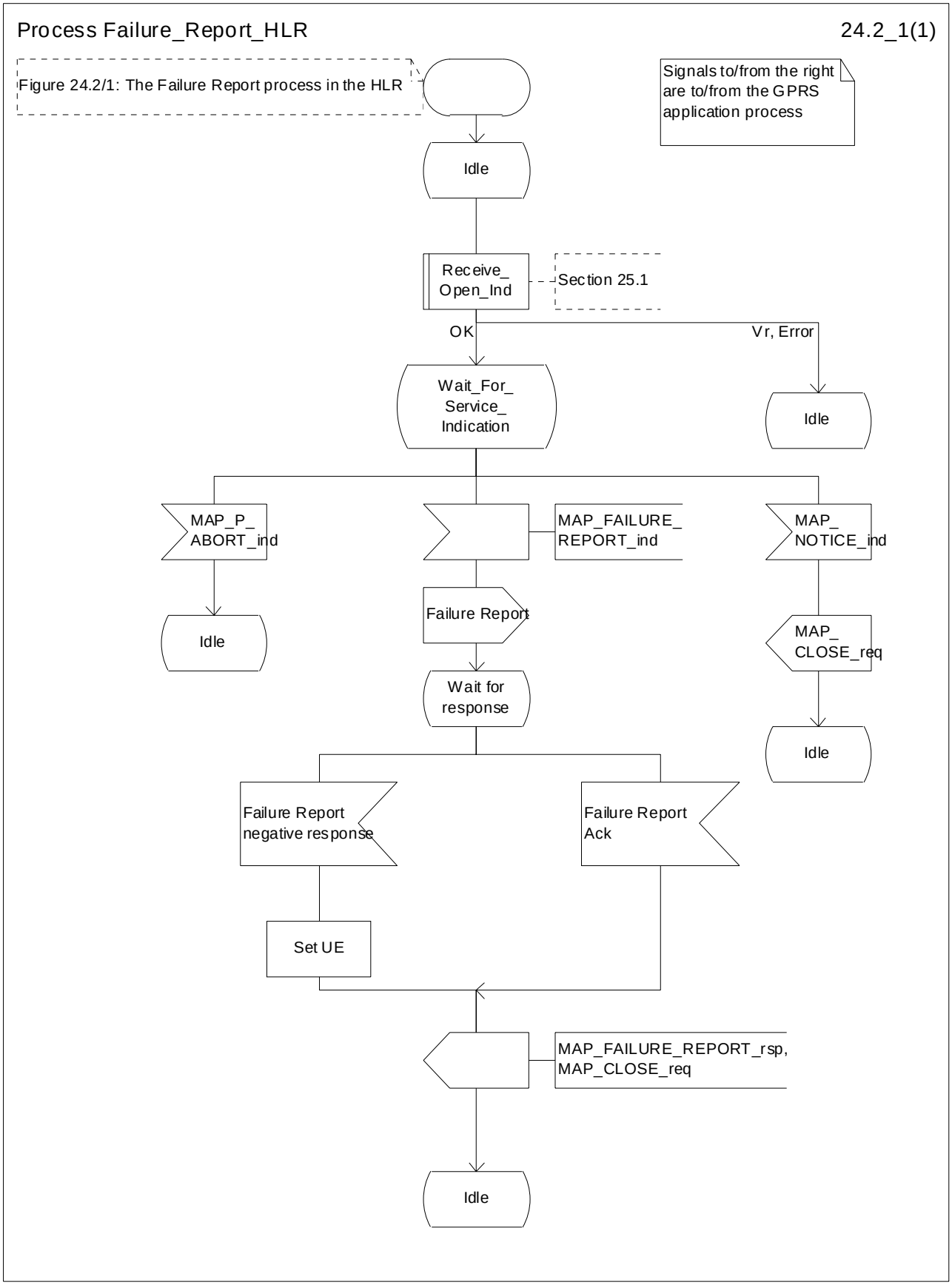


Figure 24.2/1: Process Failure_Report_HLR

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.

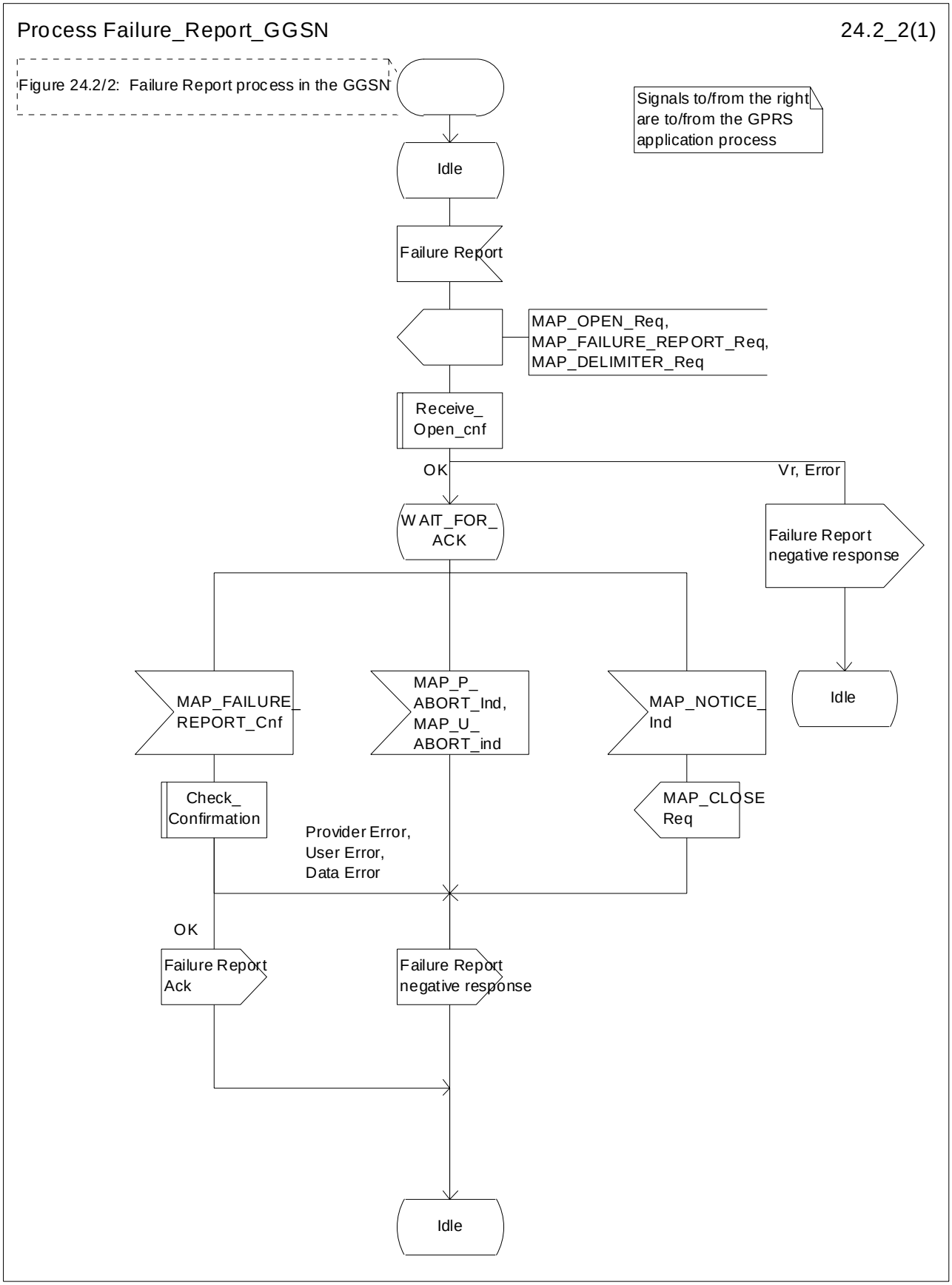


Figure 24.2/2: Process Failure_Report_GGSN

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 macros not defined in this clause; the definition of these macros can be found as follows:

Receive_Open_Ind	see clause 25.1.1;
Check_Indication	see clause 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 waits 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.

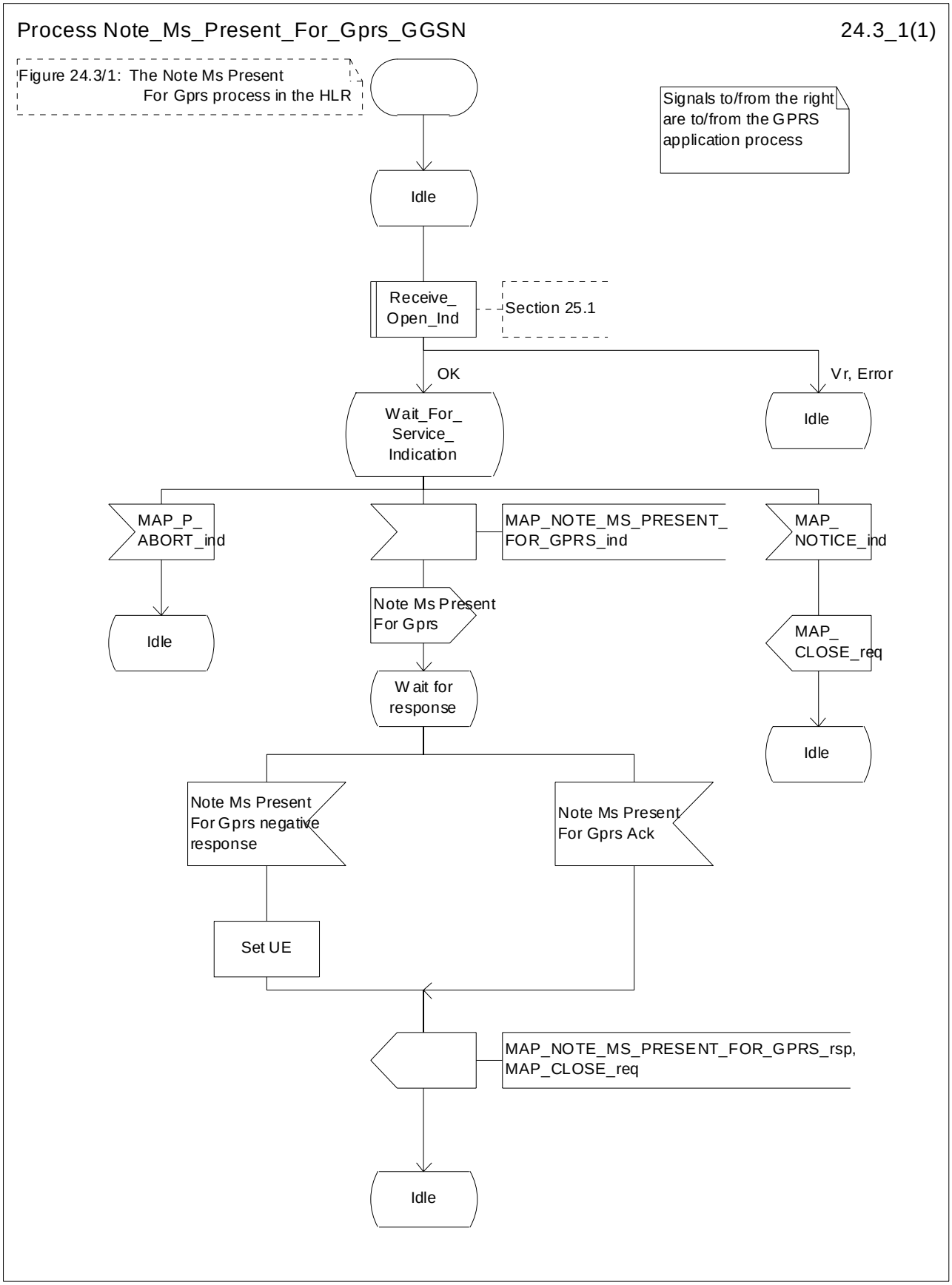


Figure 24.3/1: Process Note_Ms_Present_For_Gprs_GGSN

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.

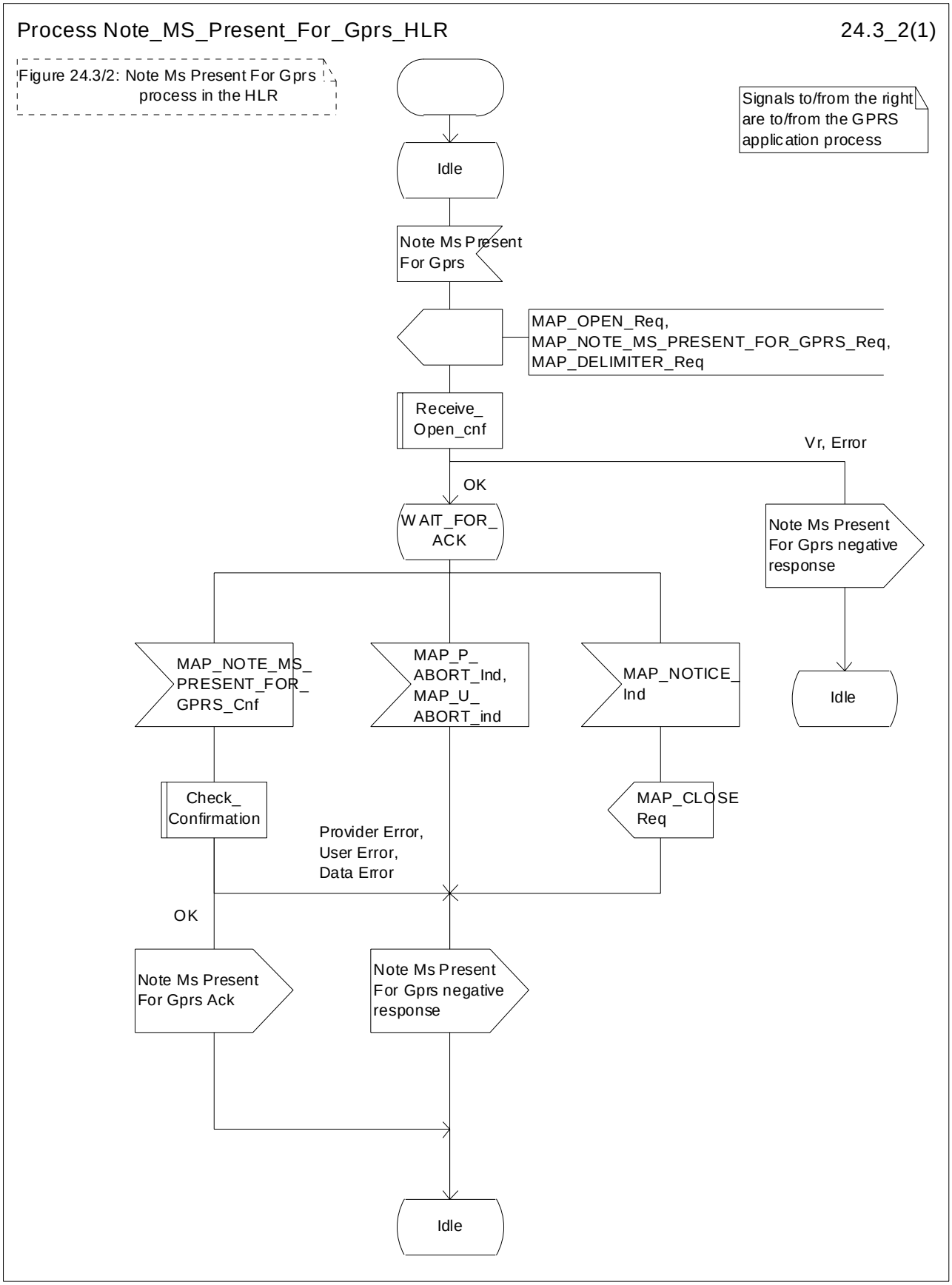


Figure 24.3/2: Process Note_Ms_Present_For_Gprs_HLR

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.

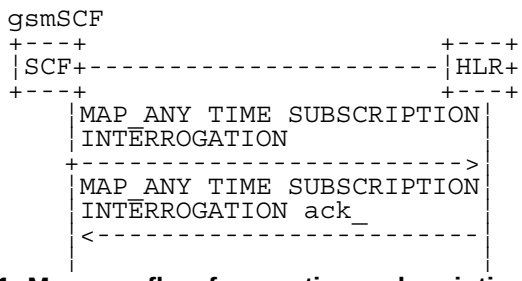


Figure 24A.1/1: Message flow for any time subscription interrogation

The following MAP services are used to retrieve requested information:

MAP_ANY_TIME_SUBSCRIPTION_INTERROGATION see clause 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 clause; the definitions of these macros can be found as follows:

Receive_Open_Ind see clause 25.1.1;

Successful outcome

When the MAP process receives a MAP_OPEN indication with the application context anyTimeInformationHandlng, 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 3GPP 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.

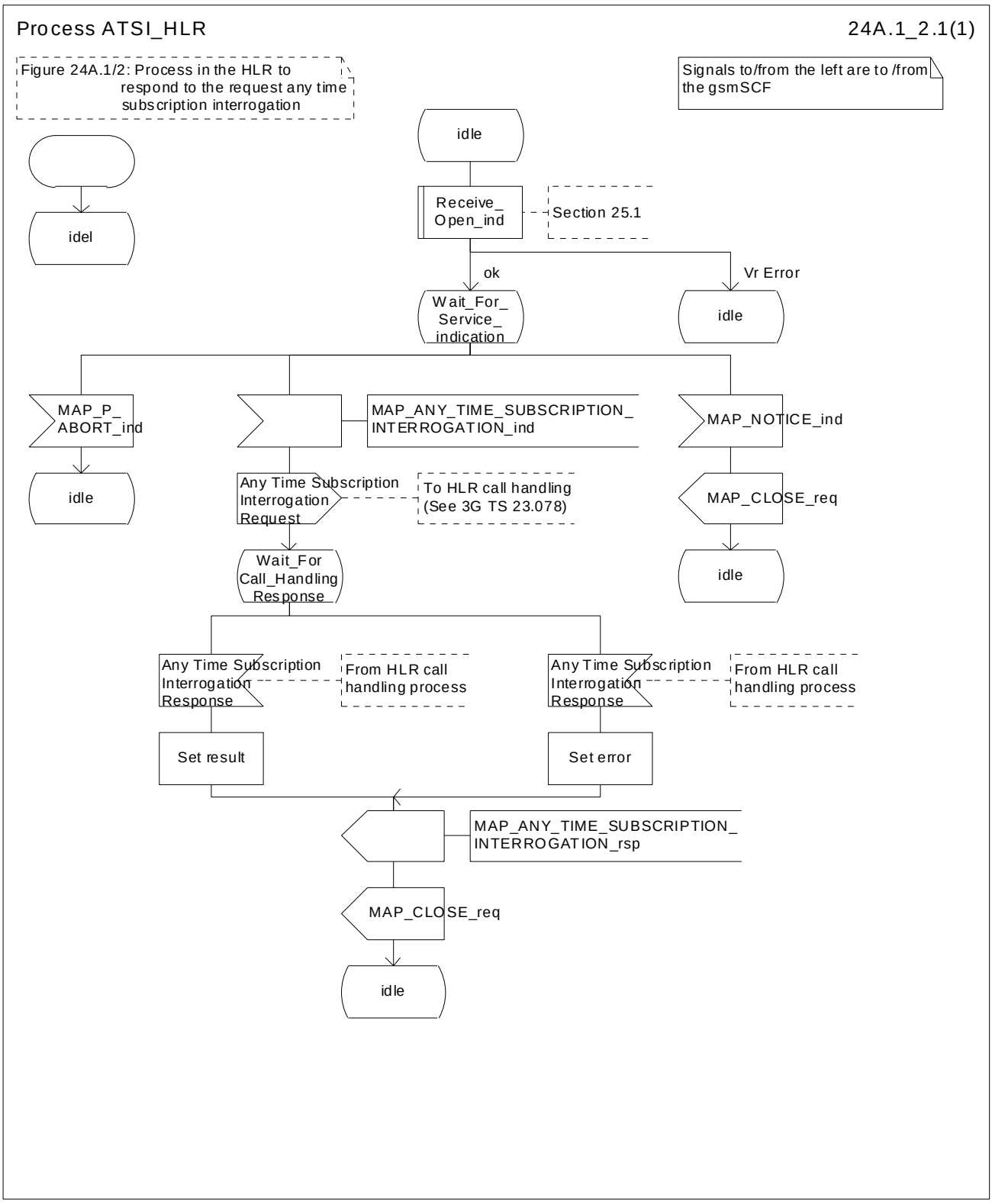


Figure 24A.1/2: Process ATSI_HLR

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

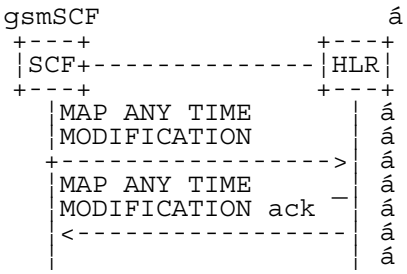


Figure 24A.2/1: Message flow for any time modification

The following MAP services are used to modify subscription information:

MAP_ANY_TIME_MODIFICATIONsee clause 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 clause; the definitions of these macros can be found as follows:

Receive_Open_Ind see clause 25.1.1;

Insert_Subs_Data_Stand_Alone_HLR see clause 25.7.2;

Successful outcome

When the MAP process receives a MAP_OPEN indication with the application context anyTimeInfromationHandling, 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 3GPP 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.

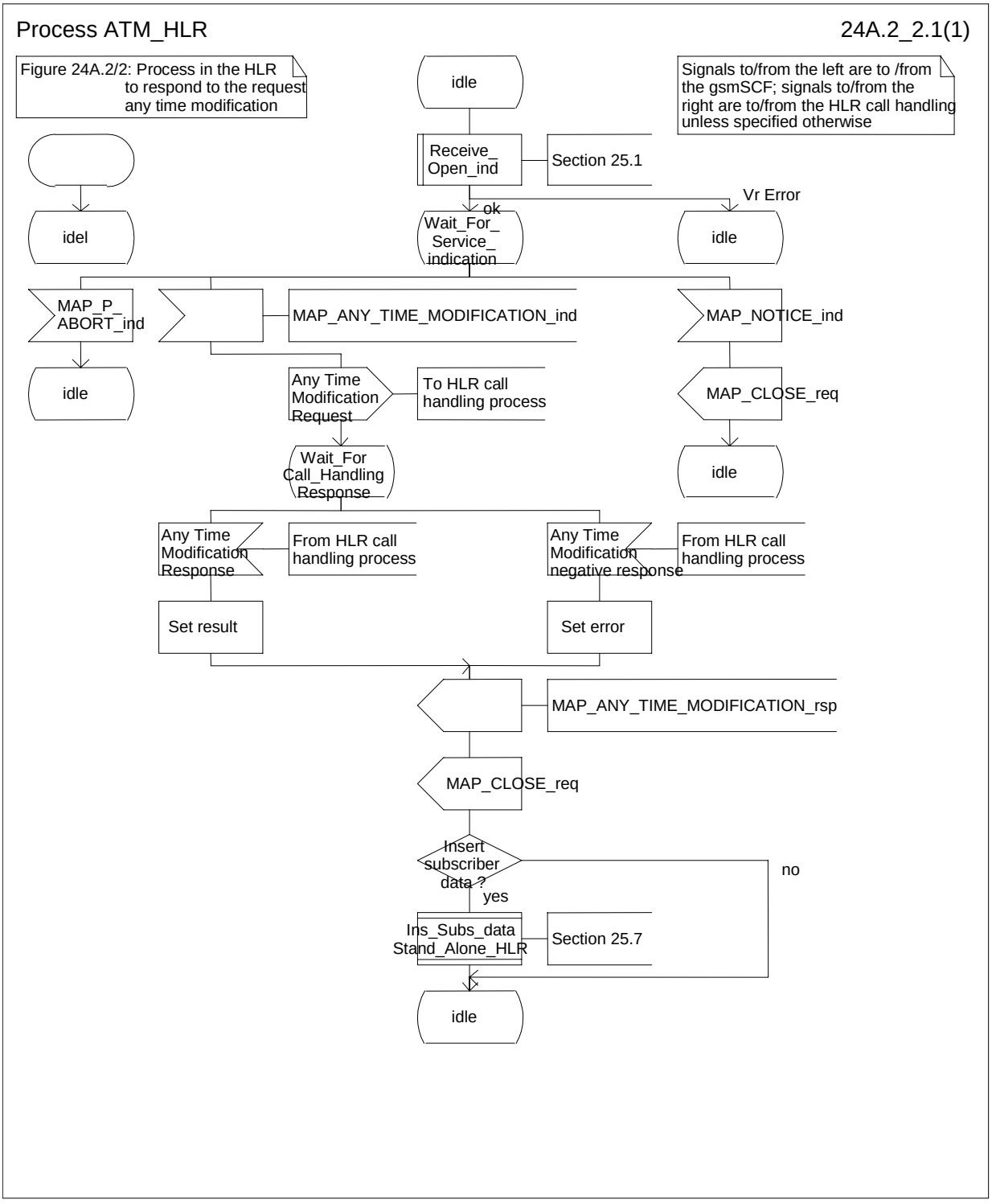


Figure 24A.2/2: Process ATM_HLR

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 3GPP 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:

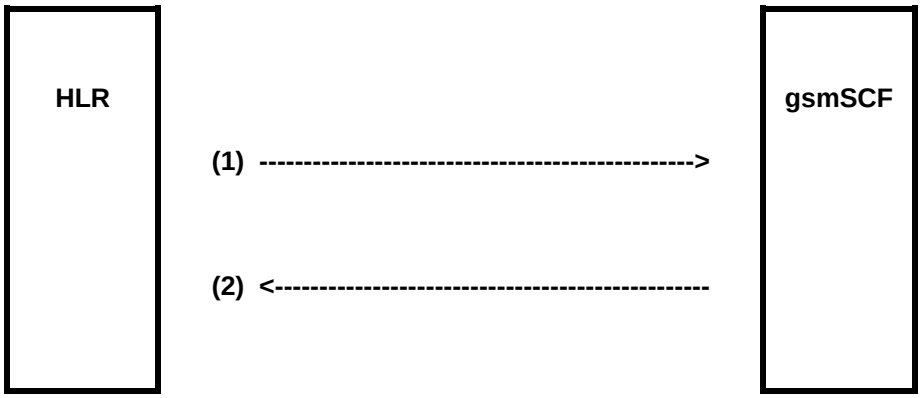


Figure 24A.3/1: Interfaces and services for subscriber data modification notification

(1) MAP-NOTE_SUBSCRIBER_DATA_MODIFIED (HLR to gsmSCF).
(2) MAP-NOTE_SUBSCRIBER_DATA_MODIFIED-ACK (gsmSCF to HLR).

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

The MAP process in the HLR to send modified data to the gsmSCF is shown in figure 24A.3/2. The MAP process invokes macros not defined in this clause; the definitions of these macros can be found as follows:

Receive_Open_Cnf see clause 25.1.2;

Check_Confirmation see clause 25.2.2.

Successful Outcome

When the MAP process receives a Notify Subscriber Data Change request from the process in the HLR, it requests a dialogue with the gsmSCF whose identity is contained in the Note Subscriber Data Modified request by sending a MAP_OPEN service request, notifies modified subscriber data to the gsmSCF using a MAP_NOTE_SUBSCRIBER_DATA_MODIFIED 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 gsmSCF.

If the HLR notices after receiving a Notify Subscriber Data Change request that the segmentation is needed the HLR does not set the “All Information Sent” indicator. Otherwise the indicator is set and the process returns to the Wait for SCF response state.

If the MAP process receives a MAP_NOTE_SUBSCRIBER_DATA_MODIFIED service confirm from the gsmSCF, 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 Notify Subscriber Data Modified ack to the process in the HLR 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 HLR does not set the indicator and sends MAP_NOTE_SUBSCRIBER_DATA_MODIFIED service request to the gsmSCF. Otherwise the indicator is set and the MAP_NOTE_SUBSCRIBER_DATA_MODIFIED service request is sent to the gsmSCF.

Dialogue opening failure

If the macro Receive_Open_Cnf indicates that the dialogue with the gsmSCF could not be opened or that the dialogue can be opened only at an earlier version, the MAP process sends a Notify Subscriber Data Modified negative response indicating system failure to the process in the HLR and returns to the idle state.

Error in MAP_NOTE_SUBSCRIBER_DATA_MODIFIED confirm

If the MAP_NOTE_SUBSCRIBER_DATA_MODIFIED service confirm contains a user error or a provider error, the MAP process sends a Notify Subscriber Data Change negative response to the process in the HLR and returns to the idle state.

Abort of gsmSCF dialogue

After the dialogue with the gsmSCF has been established, the MAP service provider may abort the dialogue by issuing a MAP_P_ABORT indication, or the gsmSCF may send a MAP_CLOSE indication. In either of these cases, the MAP process sends a Notify Subscriber Data Change negative response to the 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 gsmSCF, sends a Notify Subscriber Data Change negative response indicating system failure to the process in the HLR and returns to the idle state.

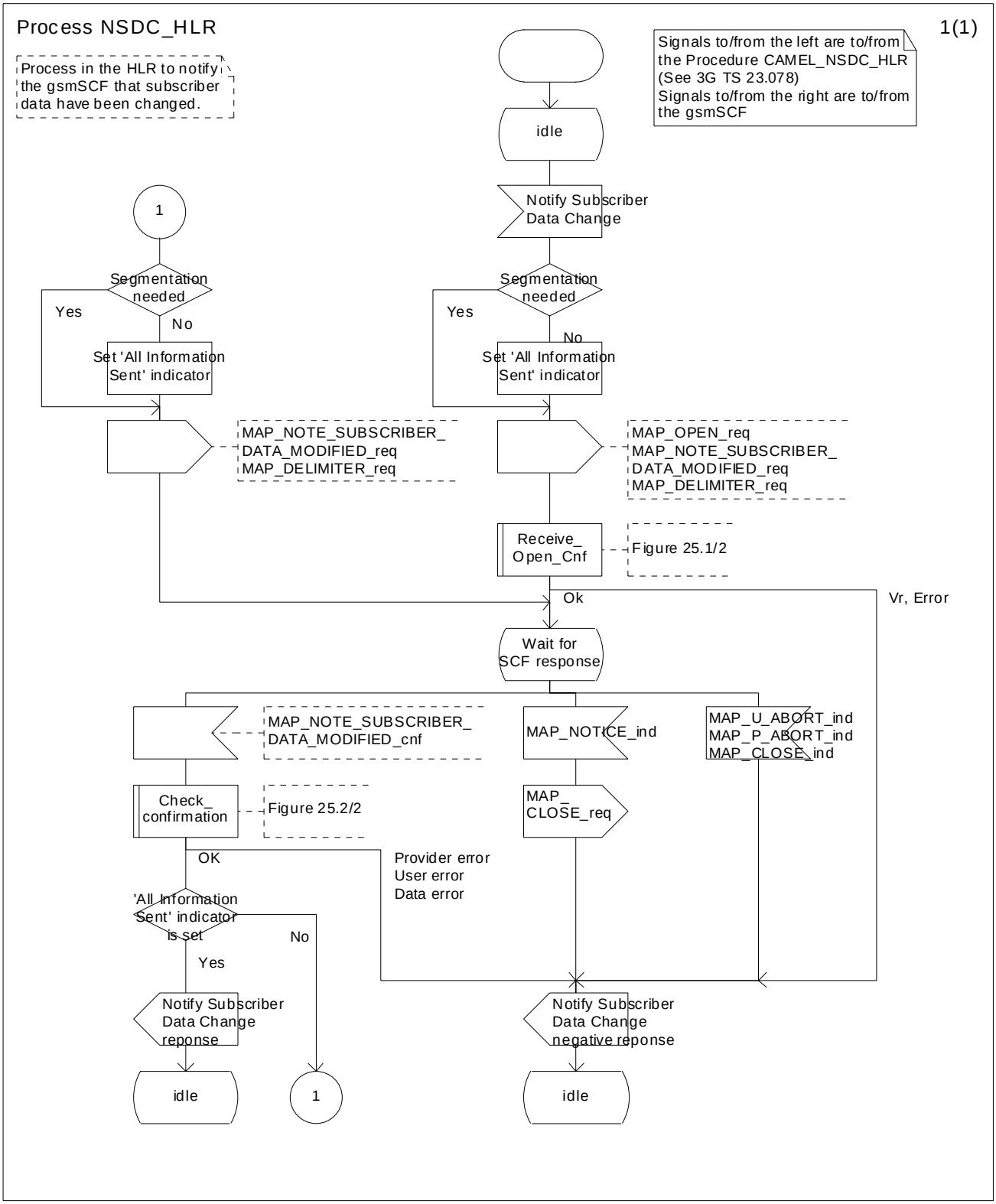


Figure 24A.3/2 Process Subscriber_Data_Modification_Notification_HLR (sheet 1 of 1)

24A.3.2.2 Process in the gsmSCF

The MAP process in the gsmSCF to handle a notification to the gsmSCF of change of subscriber data resume is shown in figure 24A.3/3. The MAP process invokes a macro not defined in this clause; the definition of this macro can be found as follows:

Receive_Open_Ind see clause 25.1.1;

Successful outcome

When the MAP process receives a MAP_OPEN indication with the application context noteSubscriberDataModified, 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_SUBSCRIBER_DATA_MODIFIED service indication is received, the MAP process checks if the "All Information Sent" indicator is set and if so it sends a Subscriber Data Changed request including all the stored data to the process in the gsmSCF, and waits for a response. The Subscriber Data Changed request contains the parameters received in the MAP_NOTE_SUBSCRIBER_DATA_MODIFIED service indication. If the "All Information Sent" indicator is not set, the received data is stored and the MAP process constructs an empty MAP_NOTE_SUBSCRIBER_DATA_MODIFIED service response, sends it to the HLR and returns to the Wait for response state.

If the process in the gsmSCF returns a negative response, the MAP process constructs a MAP_NOTE_SUBSCRIBER_DATA_MODIFIED service response, 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.

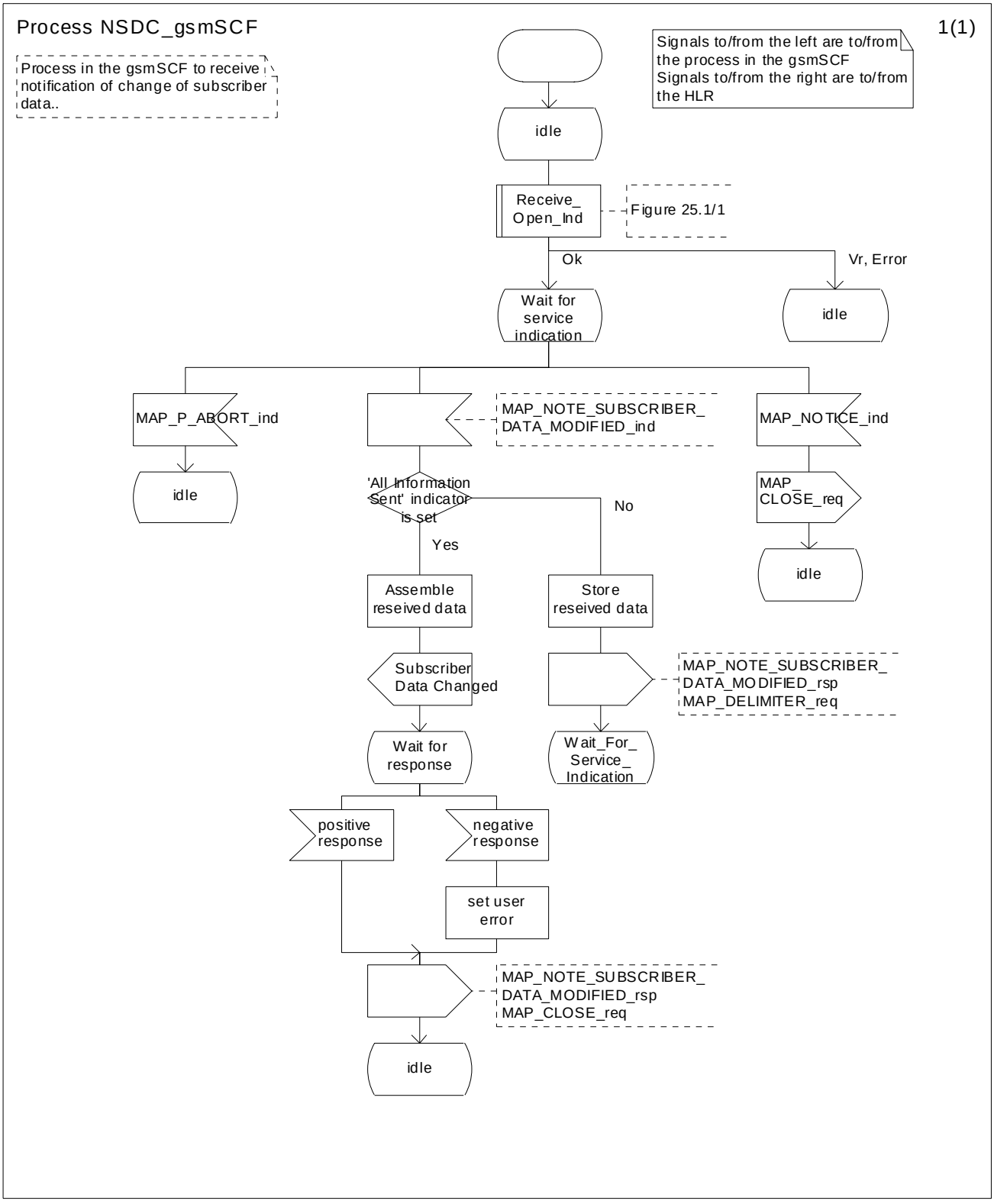


Figure 24A.3/3: Process Subscriber_Data_Modification_Notification_gsmSCF (sheet 1 of 1)

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. If the outcome of this check is an error, the MAP_OPEN responds 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" (Vr < Vn), 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.

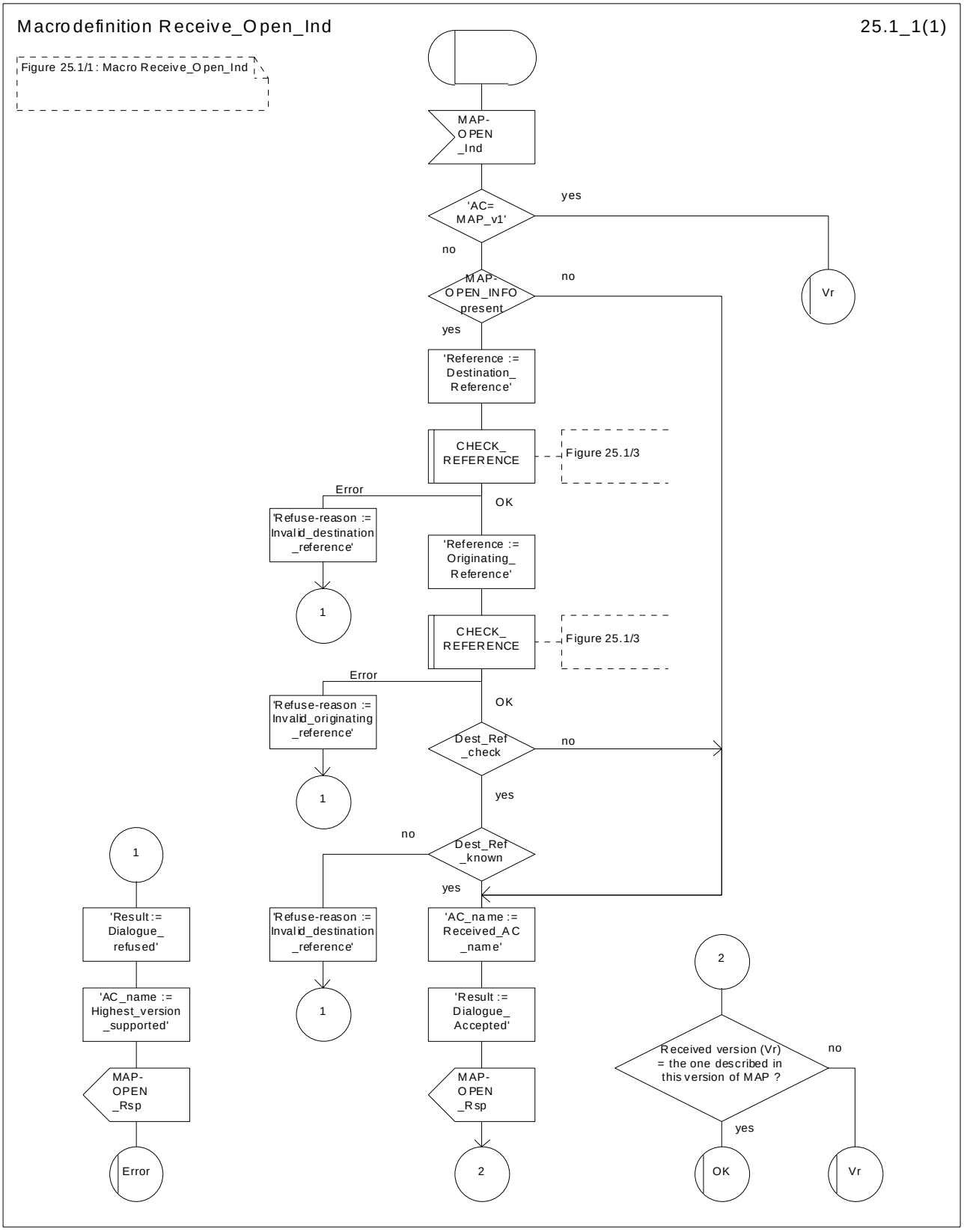


Figure 25.1/1: Macro Receive_Open_Ind

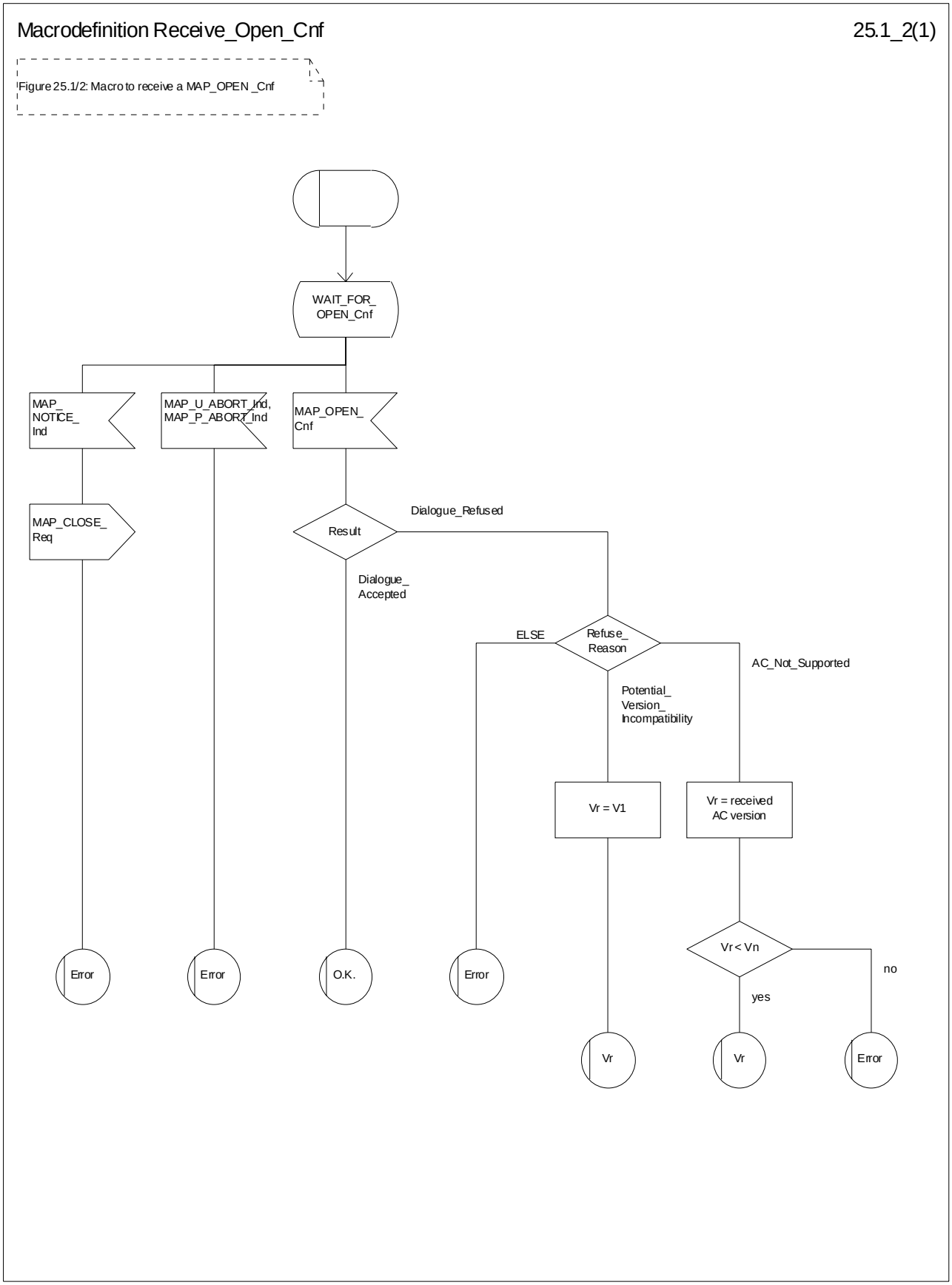


Figure 25.1/2: Macro Receive_Open_Cnf

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

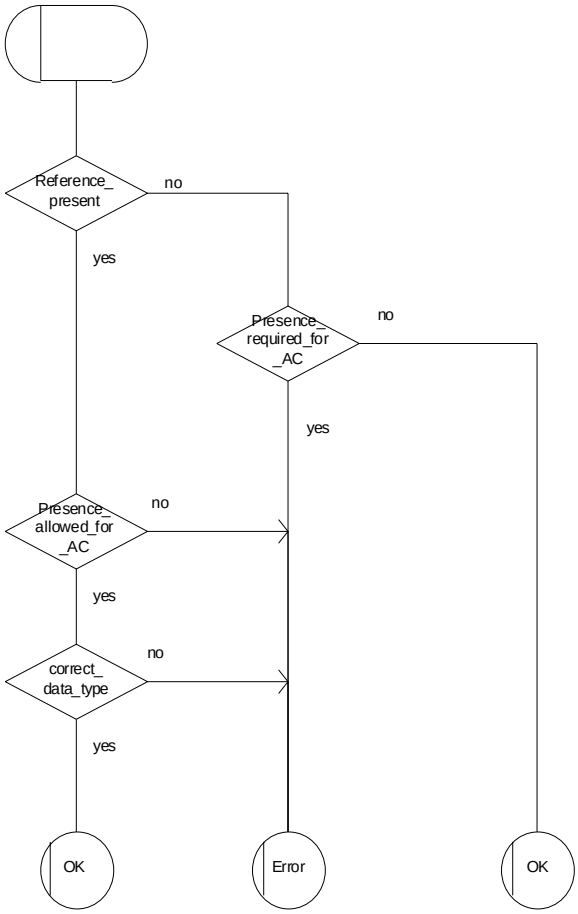


Figure 25.1/3: Macro CHECK_REFERENCE

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

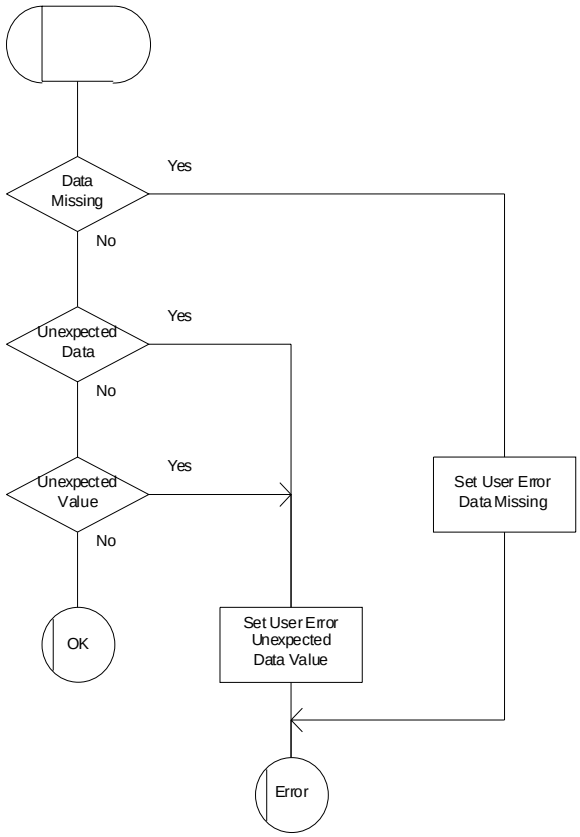


Figure 25.2/1: Macro Check_Indication

Macrodefinition Check_Confirmation 25.2_2(1)

Figure 25.2/2: Macro to check the parameters of a confirmation primitive

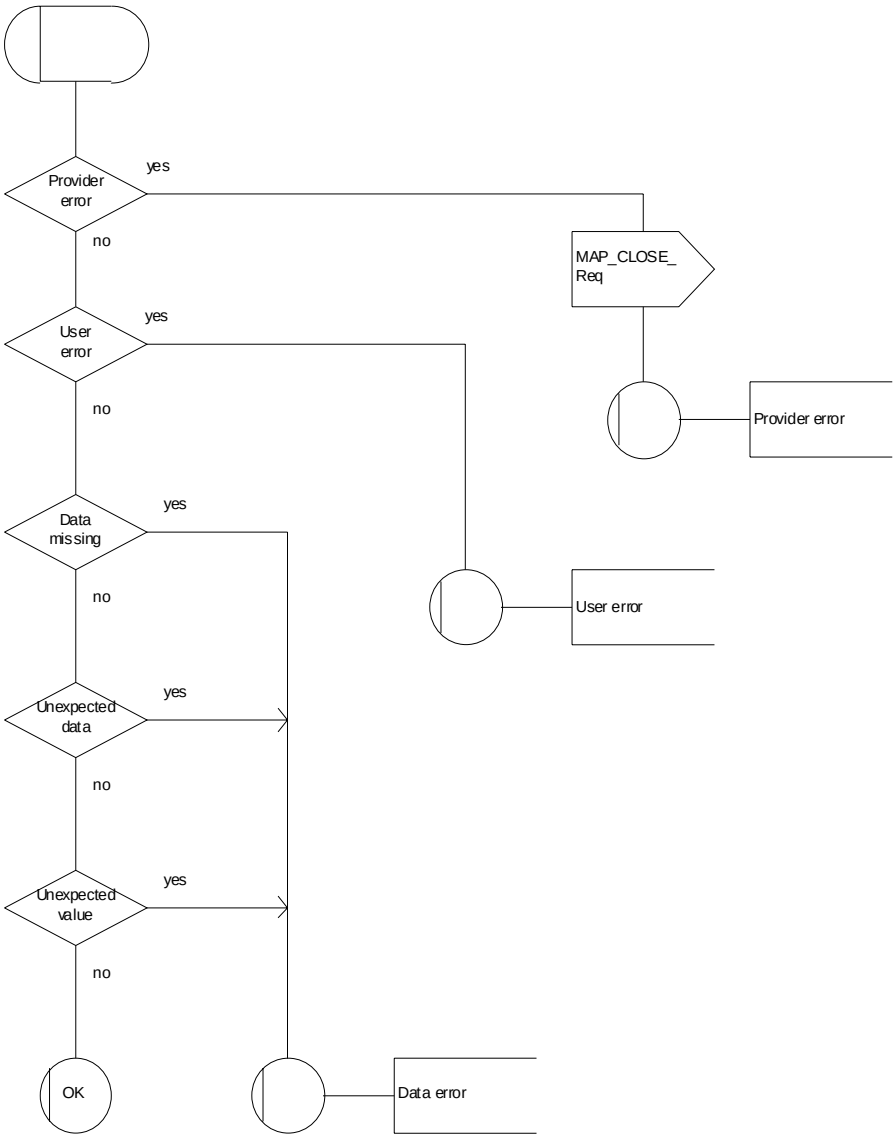


Figure 25.2/2: Macro Check_Confirmation

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 clause 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 3GPP TS 22.001 [2] 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 3GPP TS 24.008). 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 clause 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 3GPP TS 22.001 [2] 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 3GPP TS 24.008). There are several possible outcomes of paging:

- the MS responds to paging, causing a MAP_SEARCH_FOR_MS response containing the IMSI and current location area identification of the called MS to be returned to the VLR. The access connection status will be set accordingly (i.e. no RR-connection, in which case other values are not significant), and the macro terminates with successful outcome.

- the MS responds with a channel request containing an establishment cause which is not "answer to paging". The MSC sends a MAP_SEARCH_FOR_MS response primitive with user error "Busy Subscriber" before the macro terminates with unsuccessful outcome. This will give priority to the mobile originating request. Alternatively, as an implementation option, the MSC may treat this as a response to paging, which will give priority to the mobile terminating request.

- there is no response from the MS. The MSC sends a MAP_SEARCH_FOR_MS response primitive with user error "Absent Subscriber" before the macro terminates with unsuccessful outcome.

- the call handling connection or MAP transaction on which the call, SMS or unstructured SS transaction is waiting for delivery, is released before a response is received from the MS (indicated in the SDL by the input signal I-REL). The MAP transaction with the VLR will be released in this case by a MAP_U_ABORT request, and the unsuccessful macro termination will indicate transaction termination.

- the MAP transaction with the VLR may be released by receiving a MAP_U_ABORT or MAP_P_ABORT indication. The call handling connection or MAP transaction on which the call, SMS or unstructured SS transaction is waiting for delivery, is released (indicated in the SDL by the output signal I-REL), and the unsuccessful macro termination will indicate transaction termination.

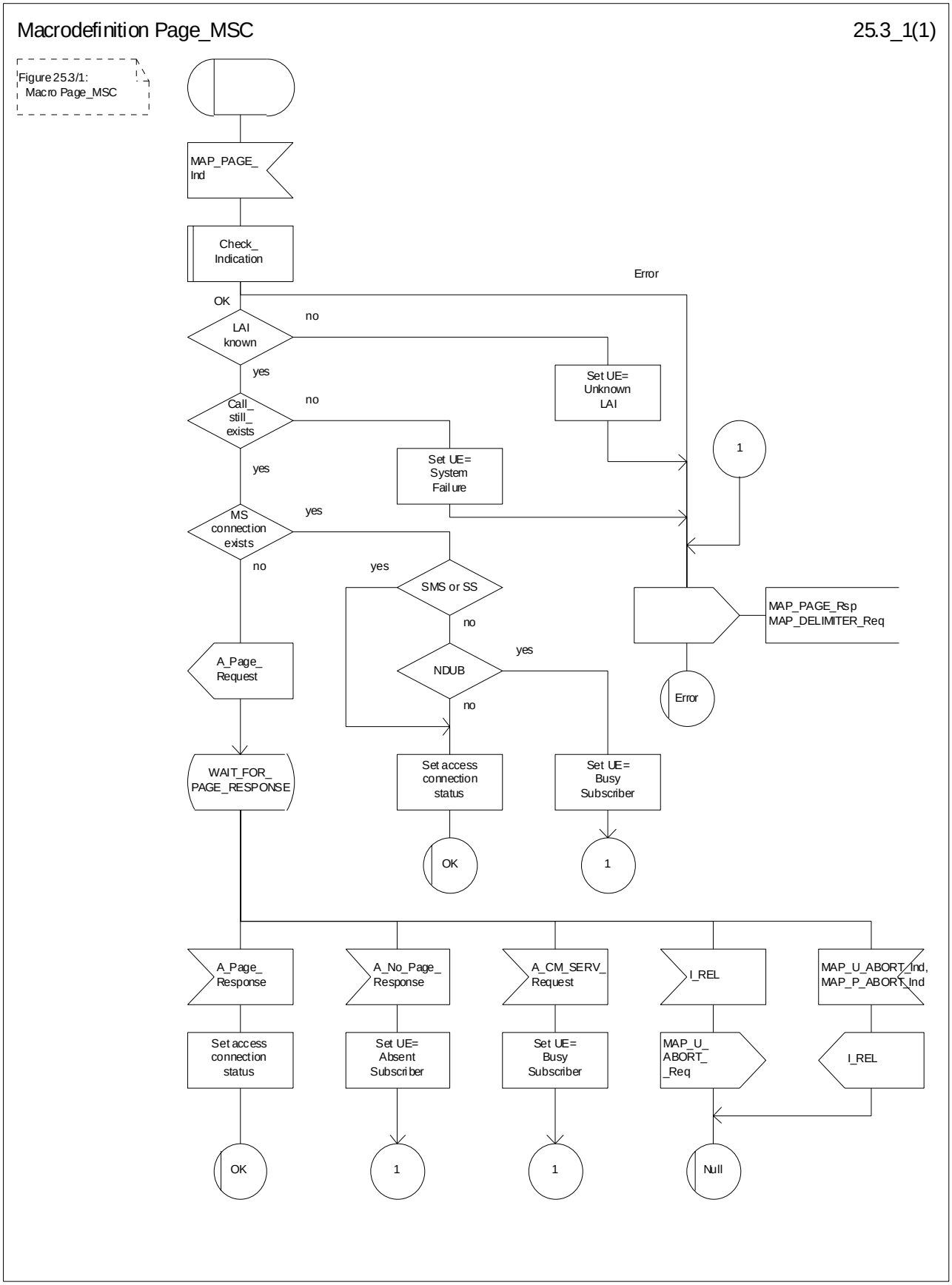


Figure 25.3/1: Macro Page_MSC

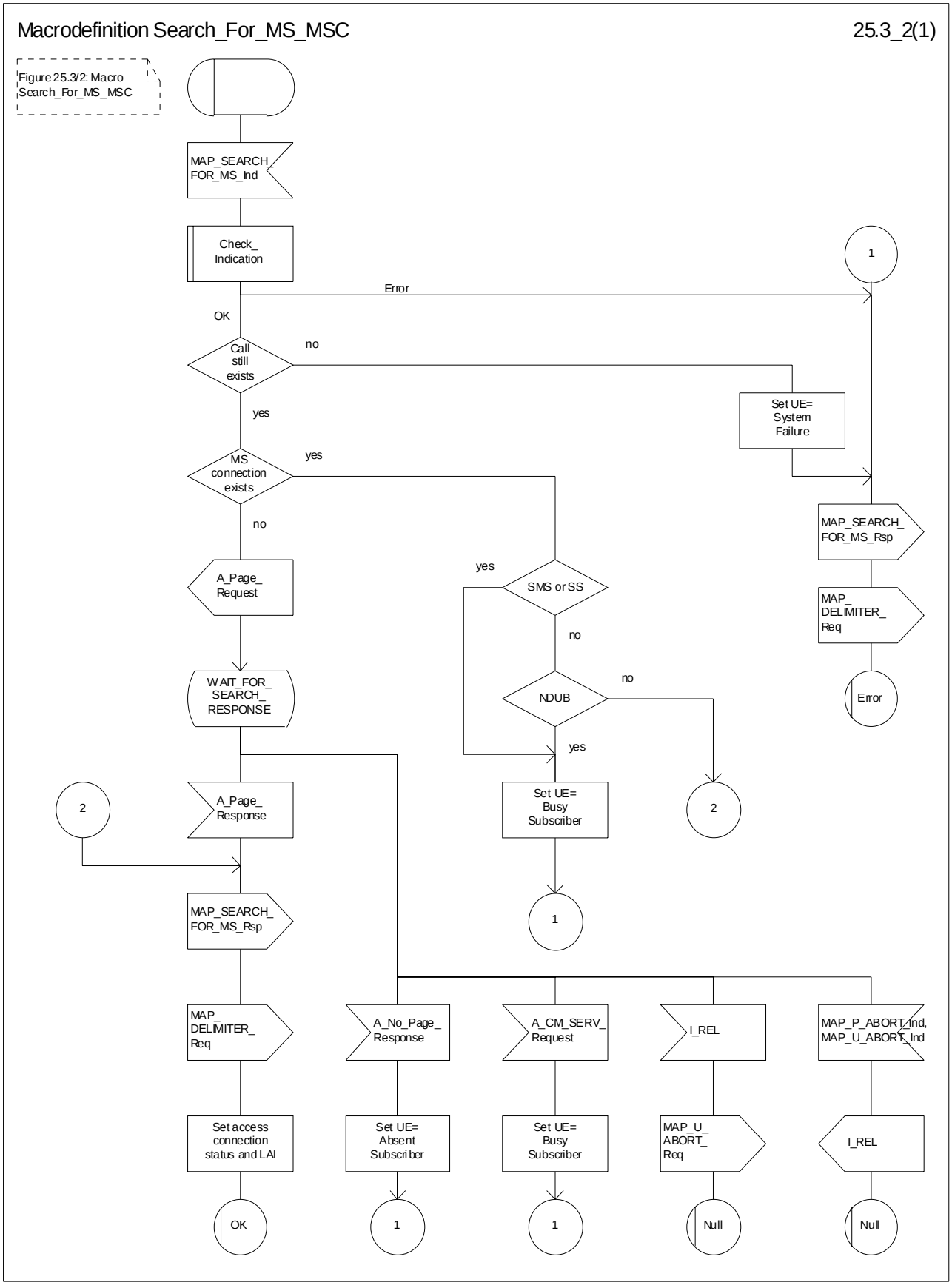


Figure 25.3/2: Macro Search_for_MS_MSC

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, clause 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 clause 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 clause 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 clause 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 clause 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 clause 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 3GPP TS 29.010 [58]);

- 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 3GPP TS 29.010 [58]), 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 3GPP TS 29.010 [58]), 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 3GPP TS 29.010 [58]);

- 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 3GPP TS 29.010 [58]), 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 3GPP TS 29.010 [58]).

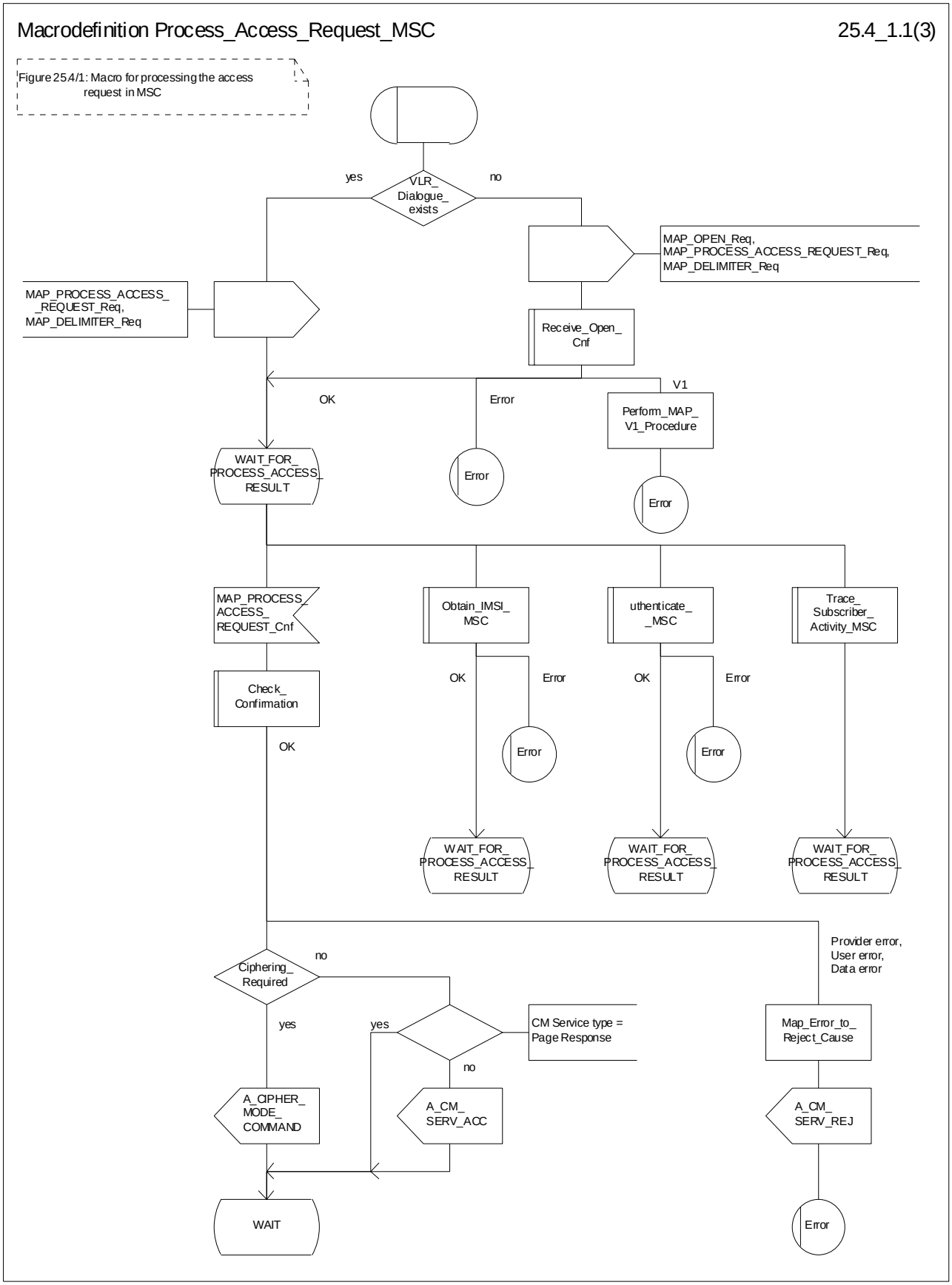


Figure 25.4/1 (sheet 1 of 3): Macro Process_Access_Request_MSC

Macrodefinition Process_Access_Request_MSC 25.4_1.2(3)

Figure 25.4/1: Macro for processing the access request in MSC

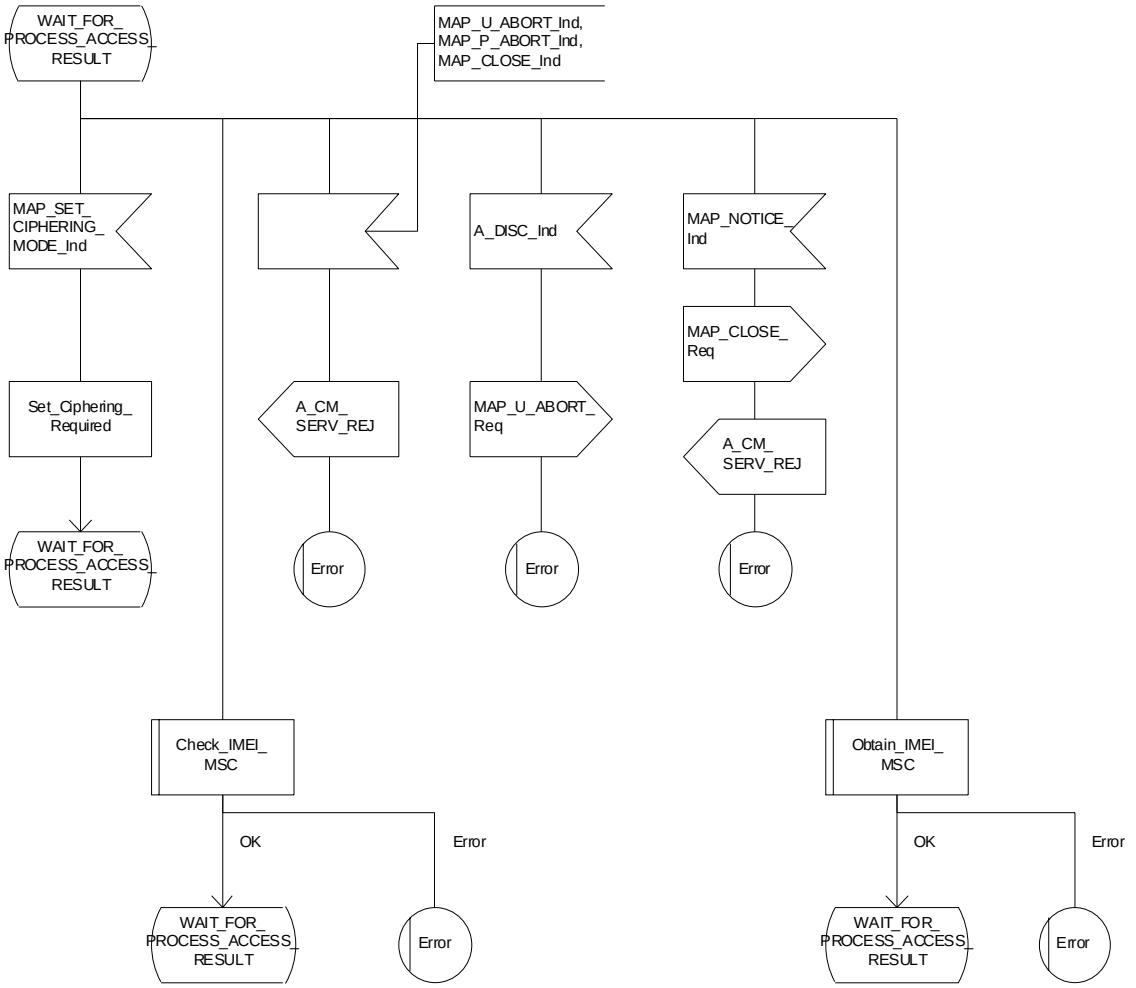


Figure 25.4/1 (sheet 2 of 3): Macro Process_Access_Request_MSC

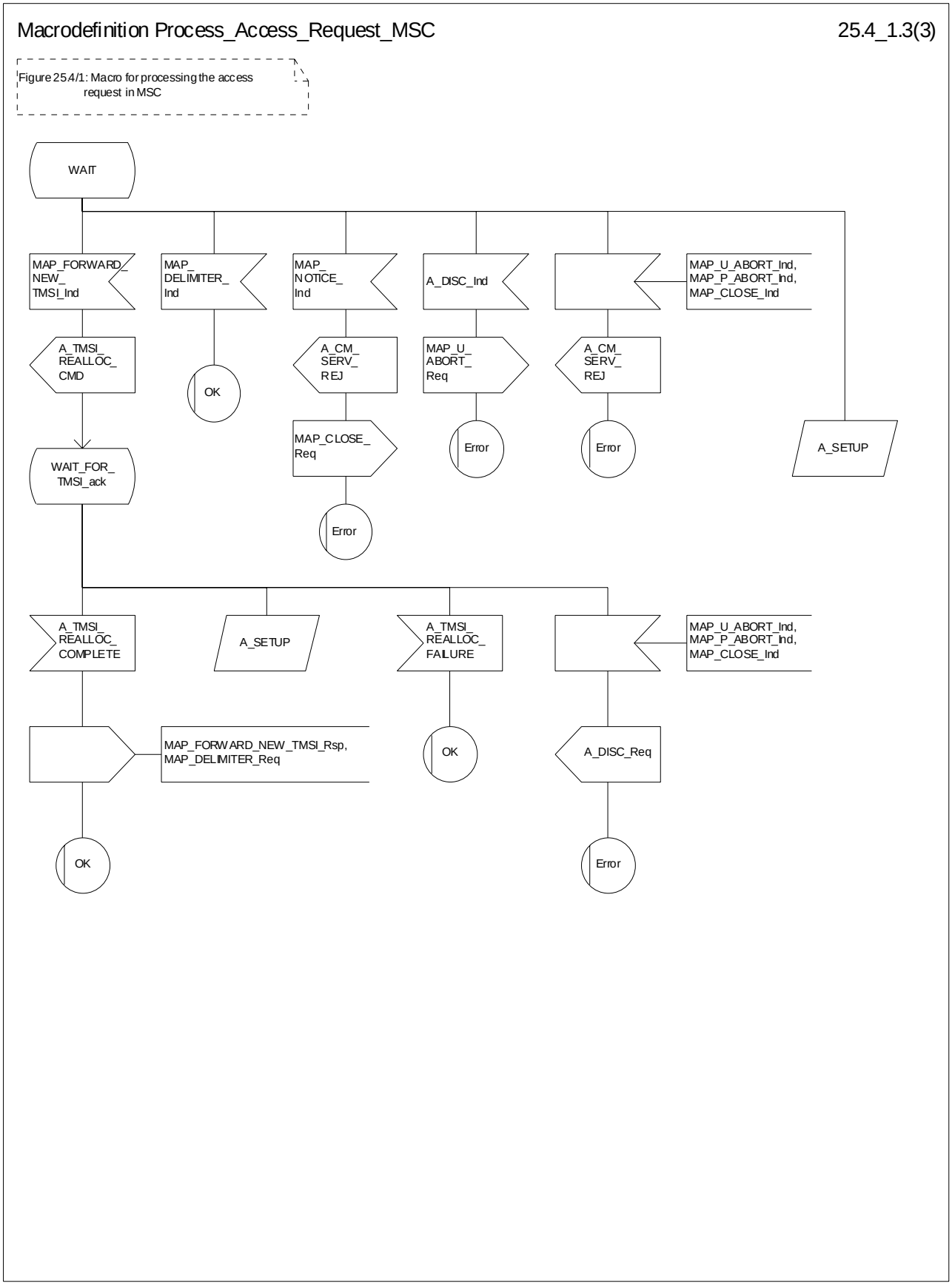


Figure 25.4/1 (sheet 3 of 3): Macro Process_Access_Request_MSC

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 clause 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 recognise 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, clause 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 clause 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 clause 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 clause 25.10).

At next, tracing is invoked if required by the operator (macro Trace_Subscriber_Activity_VLR, see clause 25.9). Thereafter,

if ciphering is not required, IMEI checking is invoked if required by the operator (see macro Check_IMEI_VLR defined in clause 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 clause 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 clause 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.

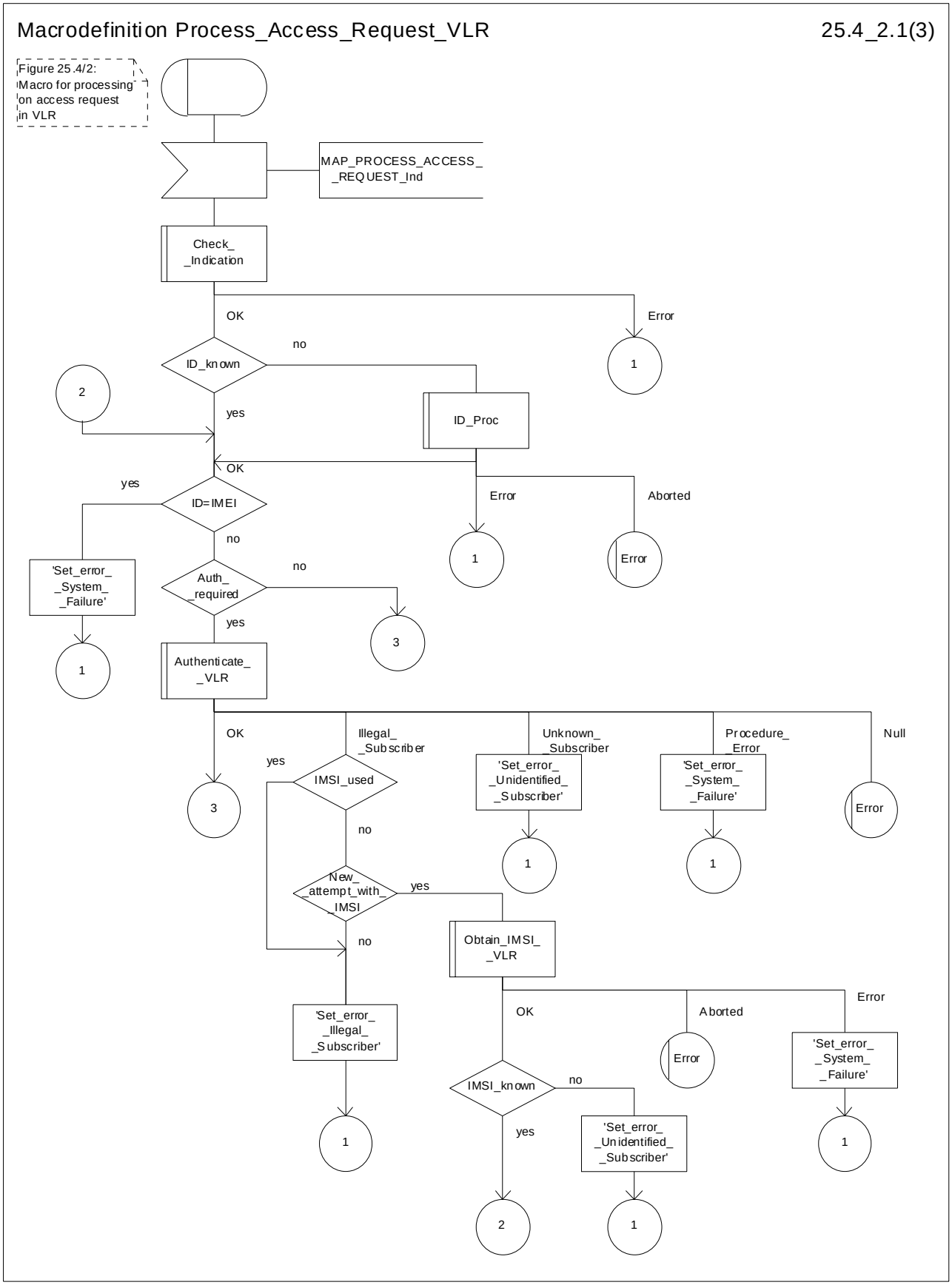


Figure 25.4/2 (sheet 1 of 3): Macro Process_Access_Request_VLR

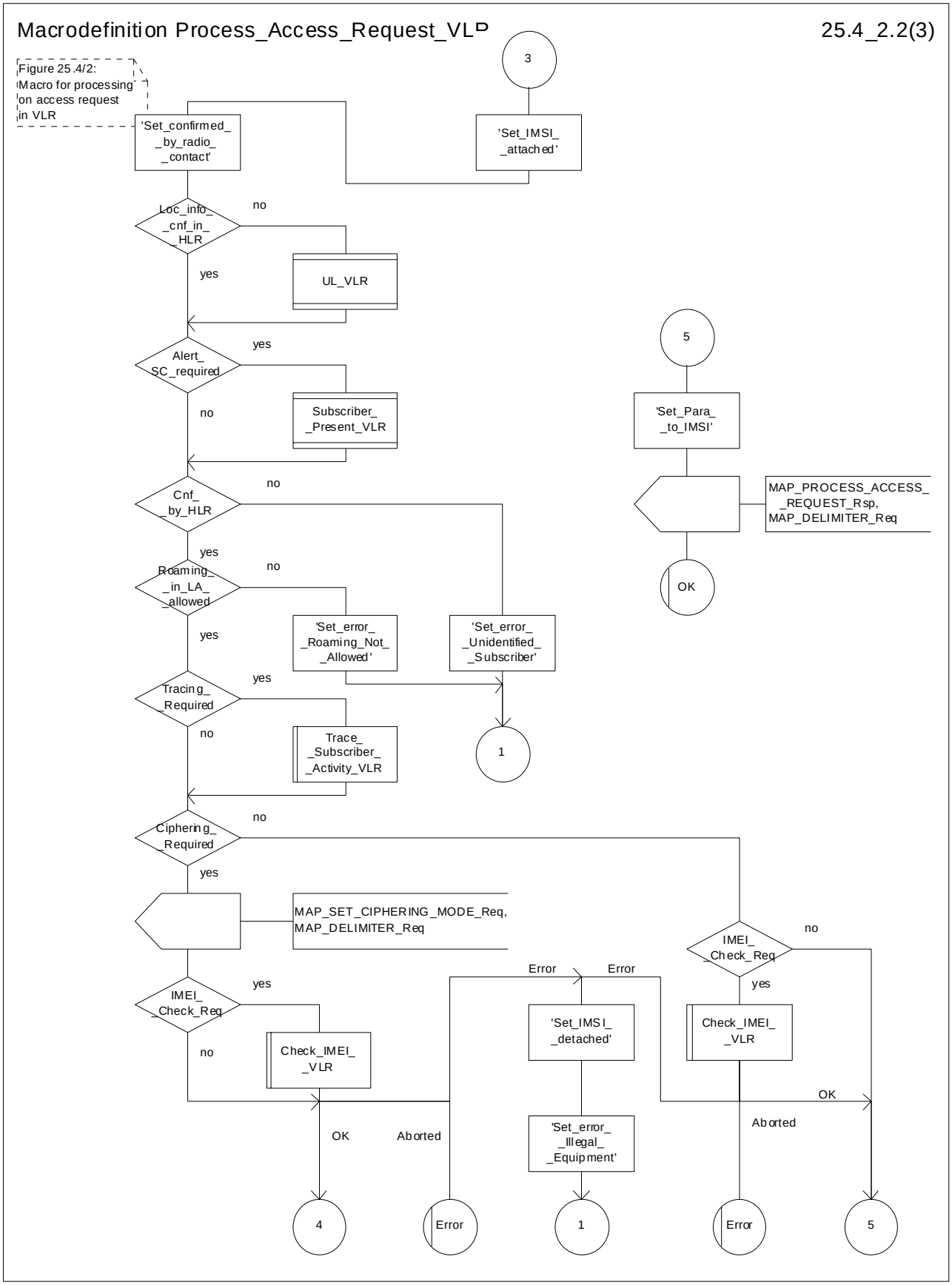


Figure 25.4/2 (sheet 2 of 3): Macro Process_Access_Request_VLR

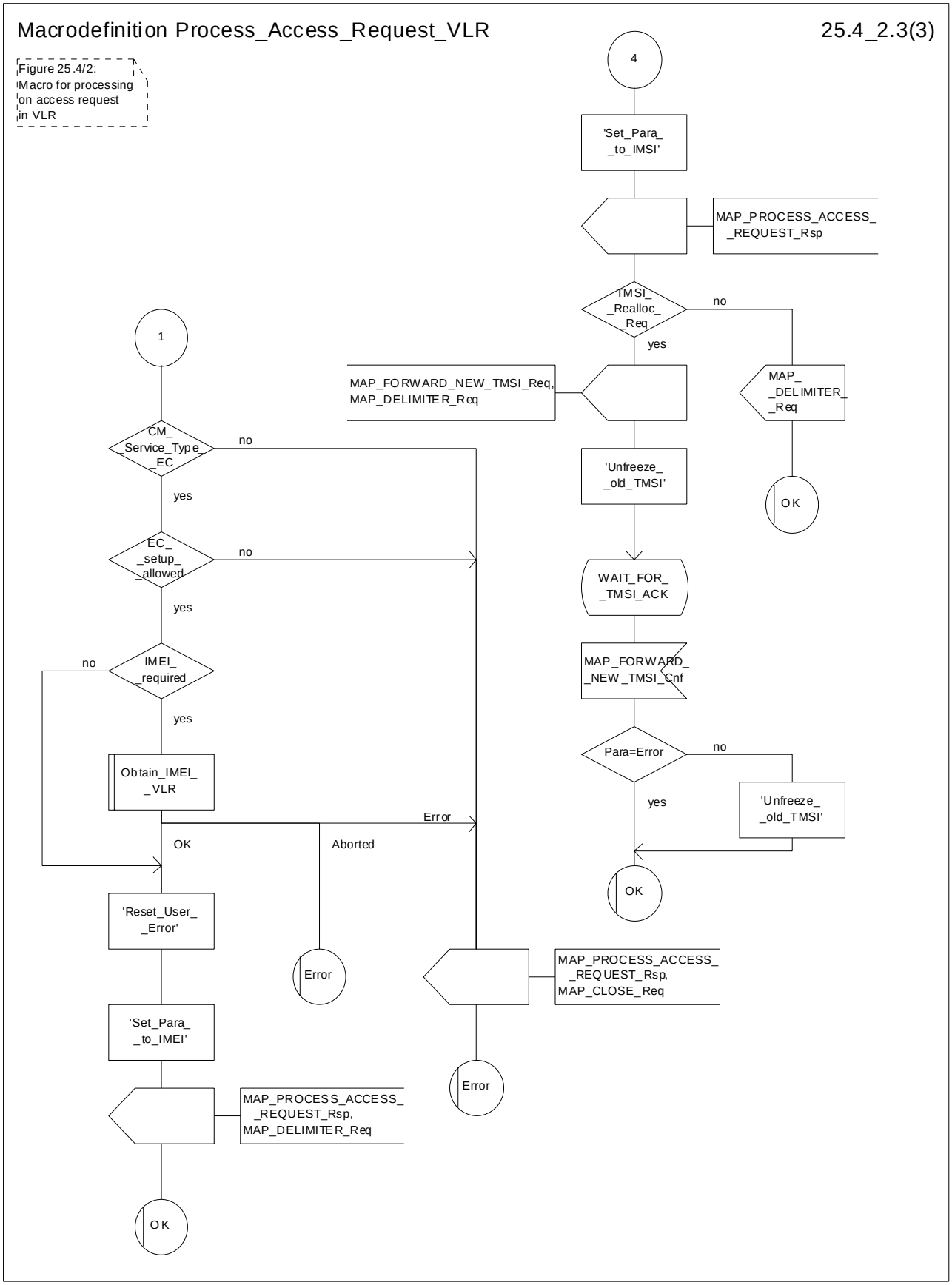


Figure 25.4/2 (sheet 3 of 3): Macro Process_Access_Request_VLR

Macrodefinition ID_Proc_VLR 25.4_3(1)

Figure 25.4/3

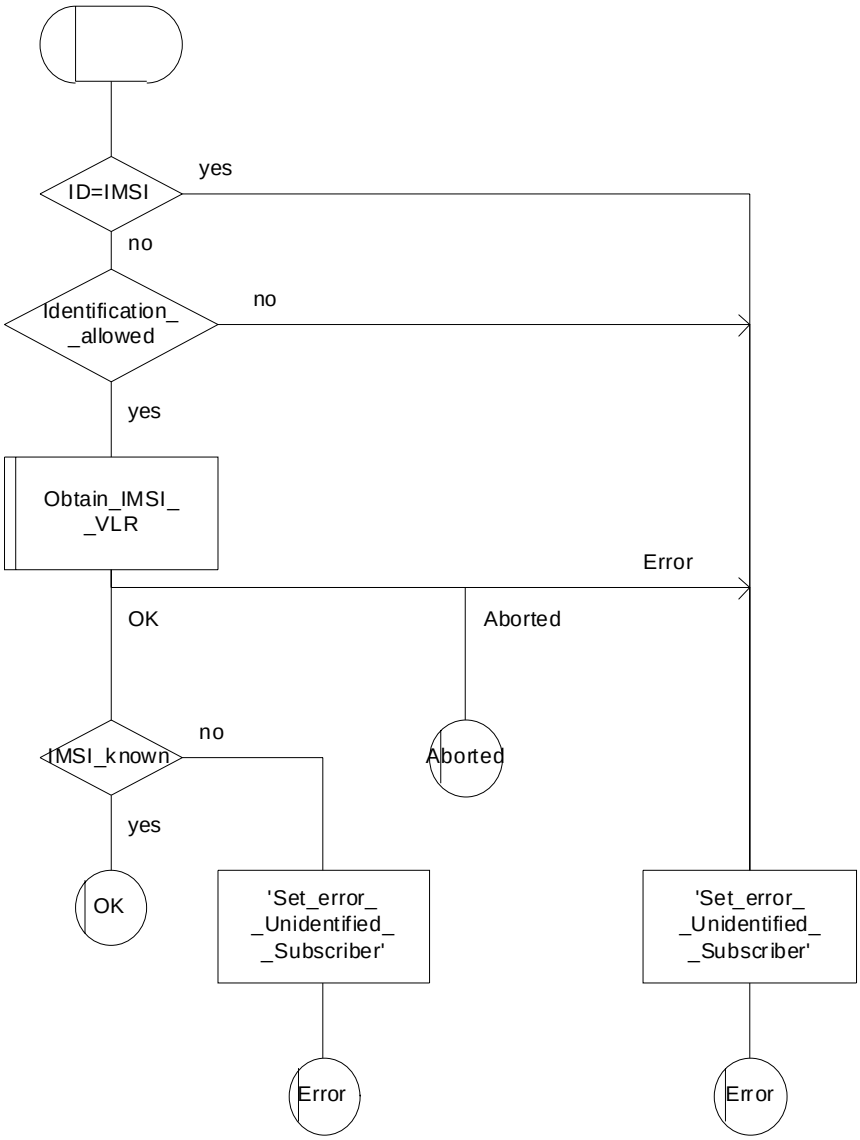


Figure 25.4/3: Macro ID_Proc_VLR

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 vectors 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 vectors 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, CKSN or KSI, and possibly AUTN 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), or whether the received RES is identical to the stored XRES. If this is not the case, the "Illegal Subscriber" exit is used. If the SRES values or RES and XRES 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 authentication vectors 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, transferred by means of the TC-RESULT-L service, 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_AUTHENTIFICATION_INFO_Request with no parameter part. 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 clause 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 by means of the TC-RESULT-L service.

- if segmentation of the response message is required and allowed, a MAP_SEND_AUTHENTICATION_INFO_response transferred by means of the TC-RESULT-L service, 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

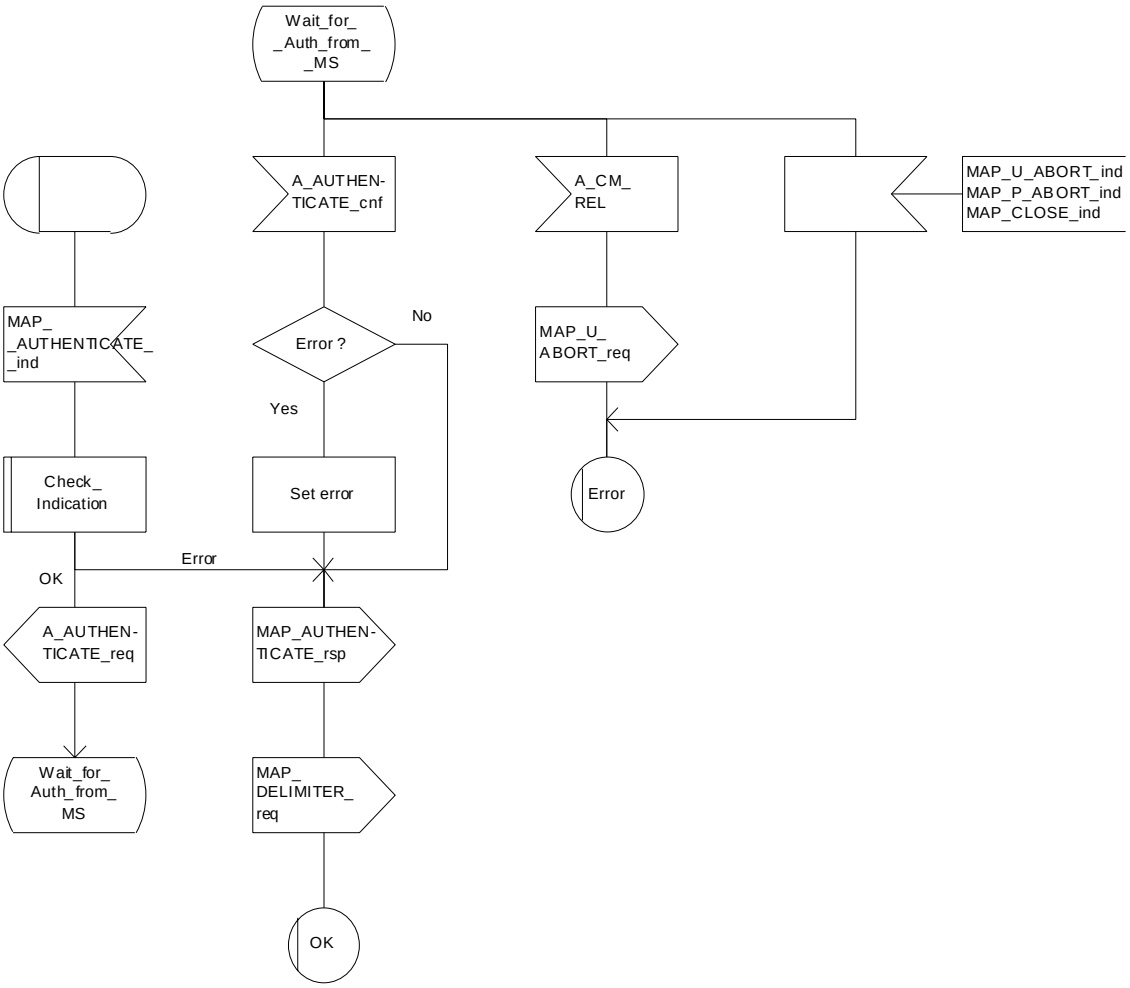


Figure 25.5/1: Macro Authenticate_MSC

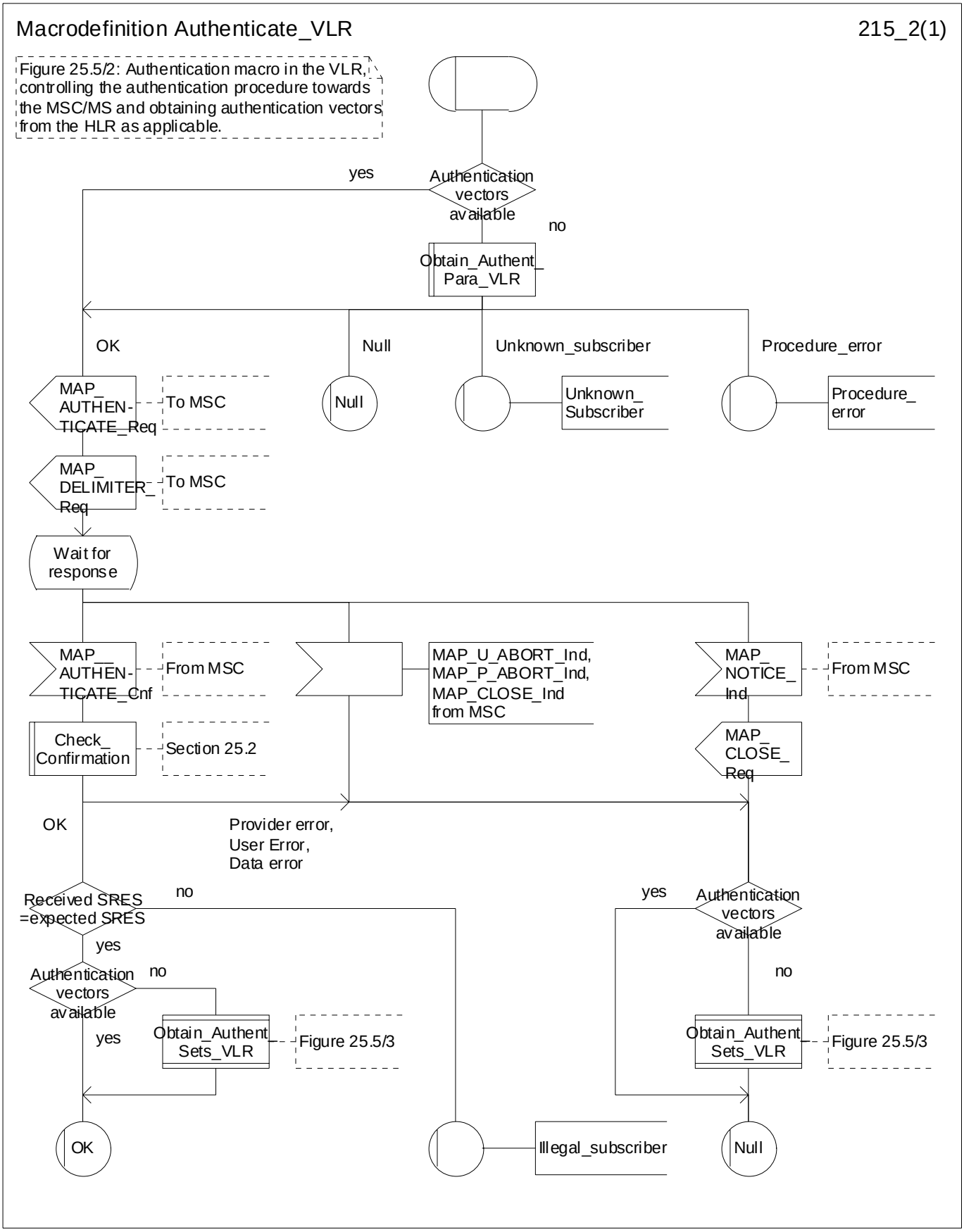


Figure 25.5/2: Macro Authenticate_VLR

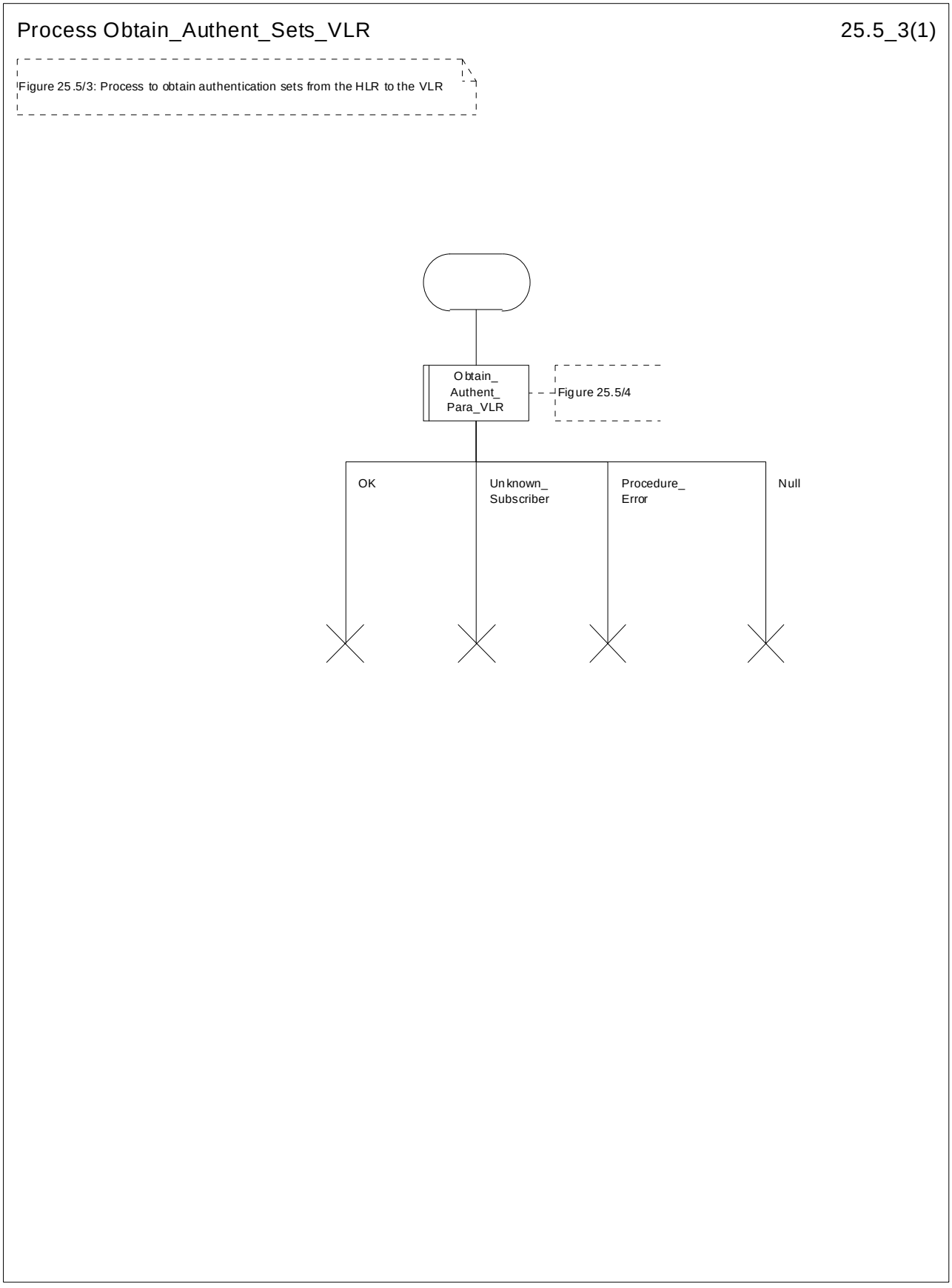


Figure 25.5/3: Process Obtain_Authentication_Sets_VLR

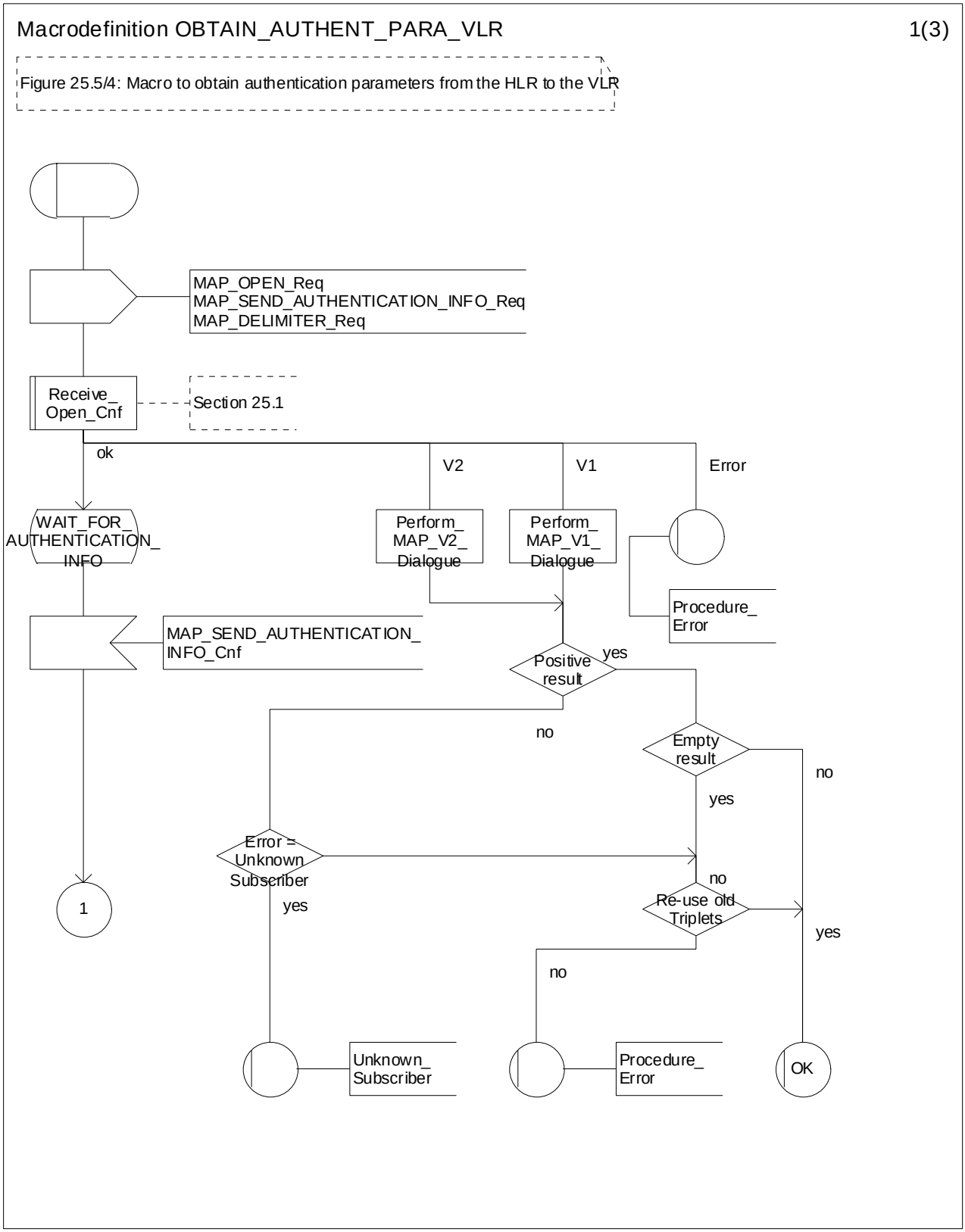


Figure 25.5/4 (sheet 1 of 3): Macro Obtain_Authent_Para_VLR

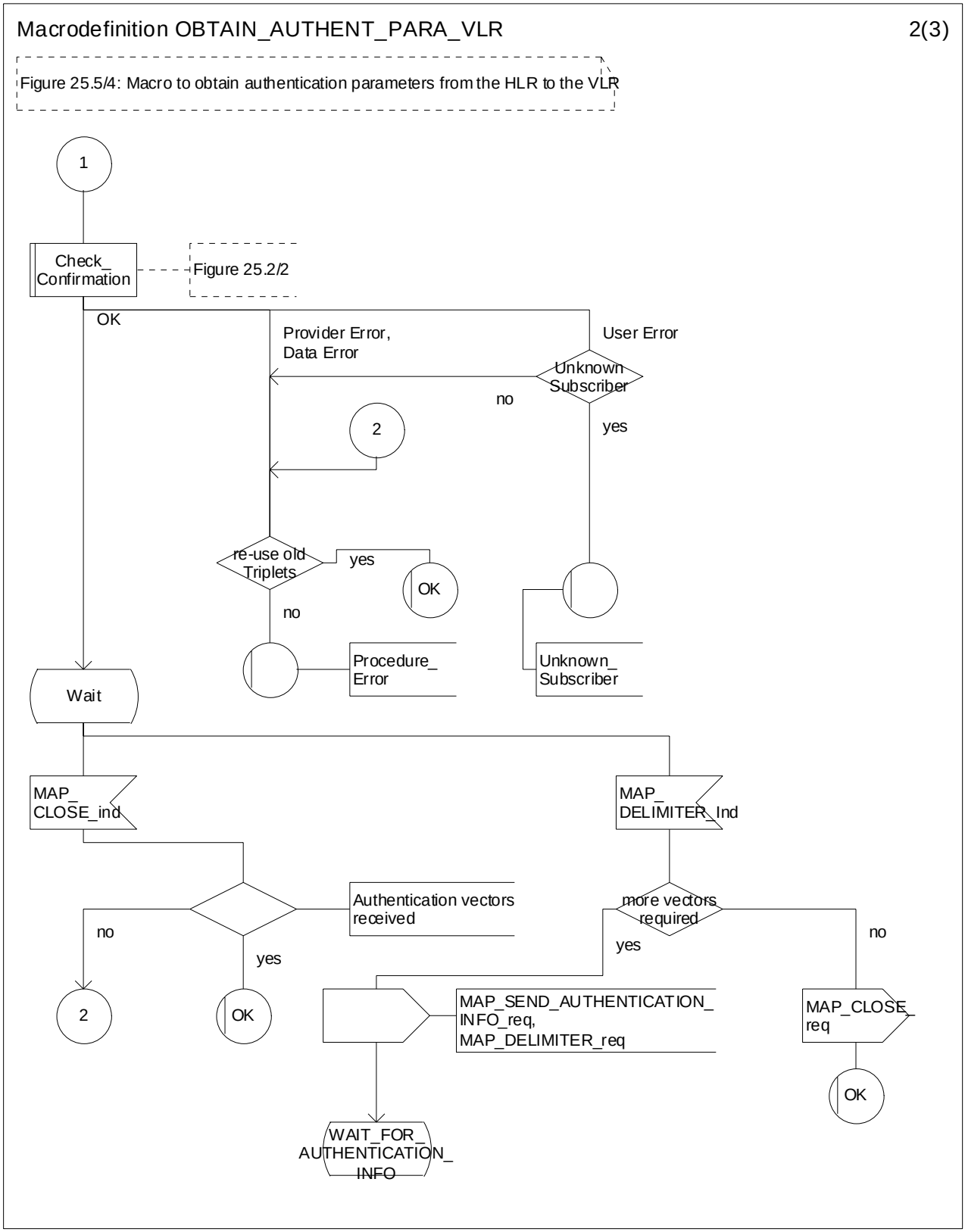


Figure 25.5/4 (sheet 2 of 3): Macro Obtain_Authent_Para_VLR

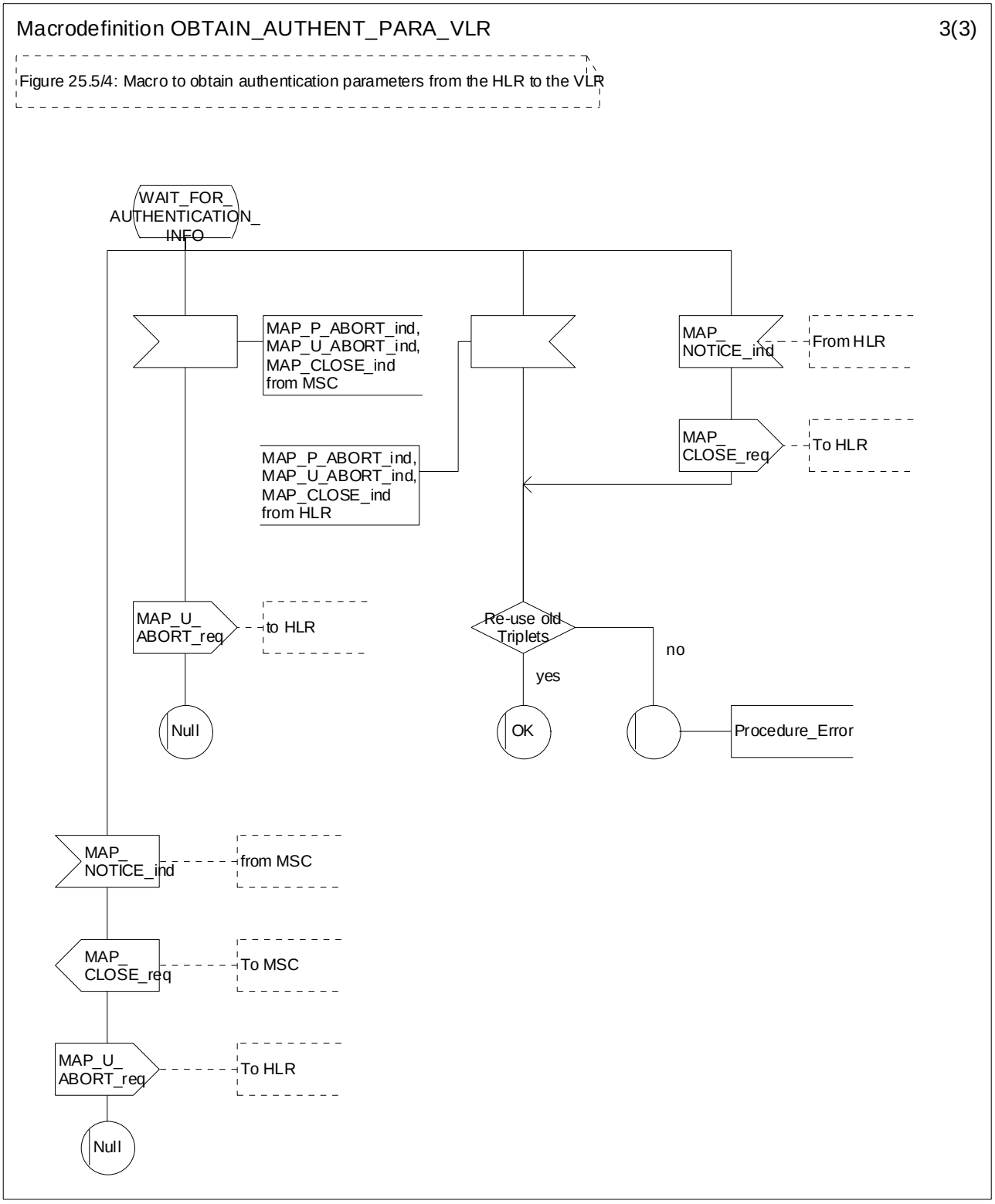


Figure 25.5/4 (sheet 3 of 3): Macro Obtain_Authent_Para_VLR

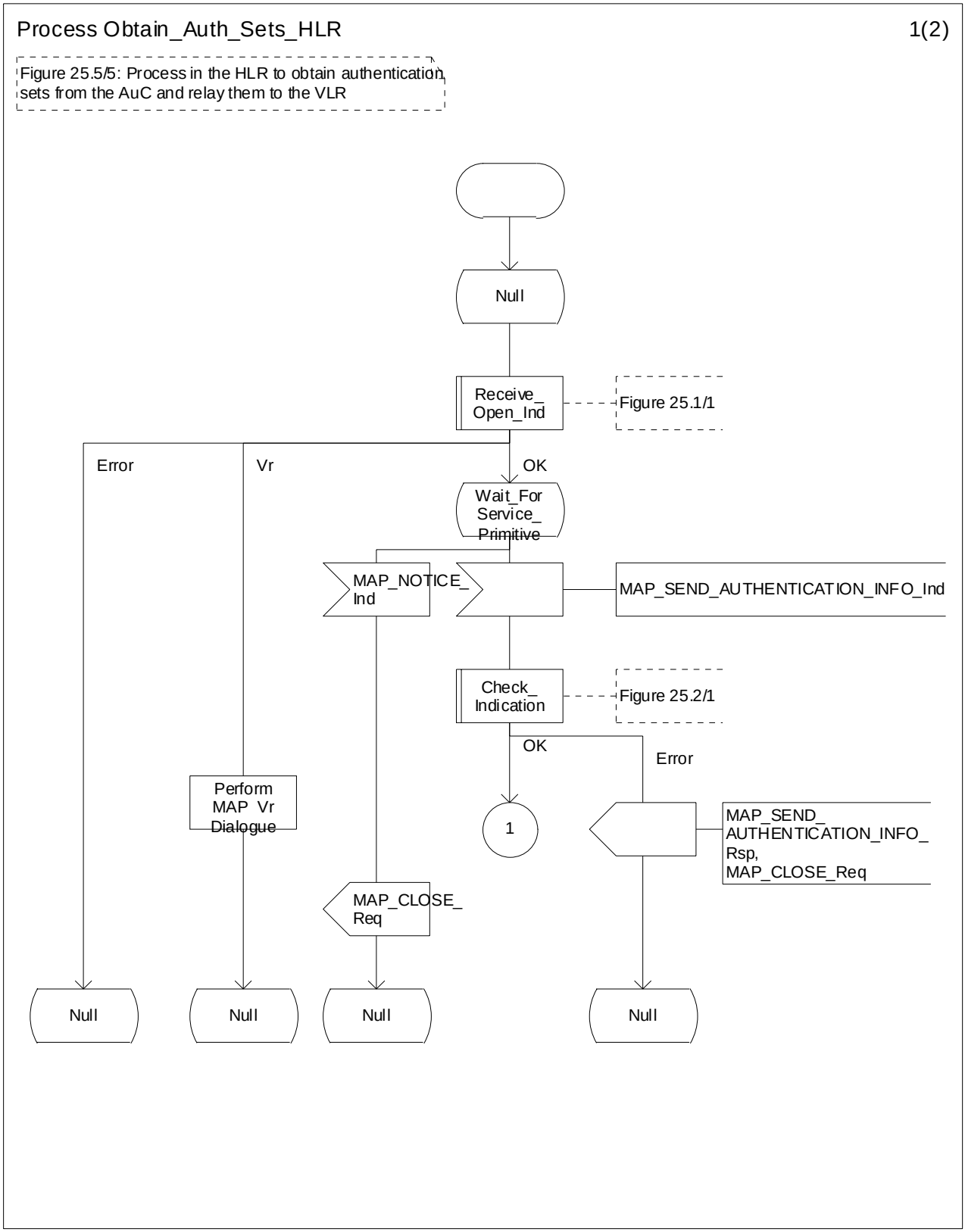


Figure 25.5/5 (sheet 1 of 2): Process Obtain_Auth_Sets_HLR

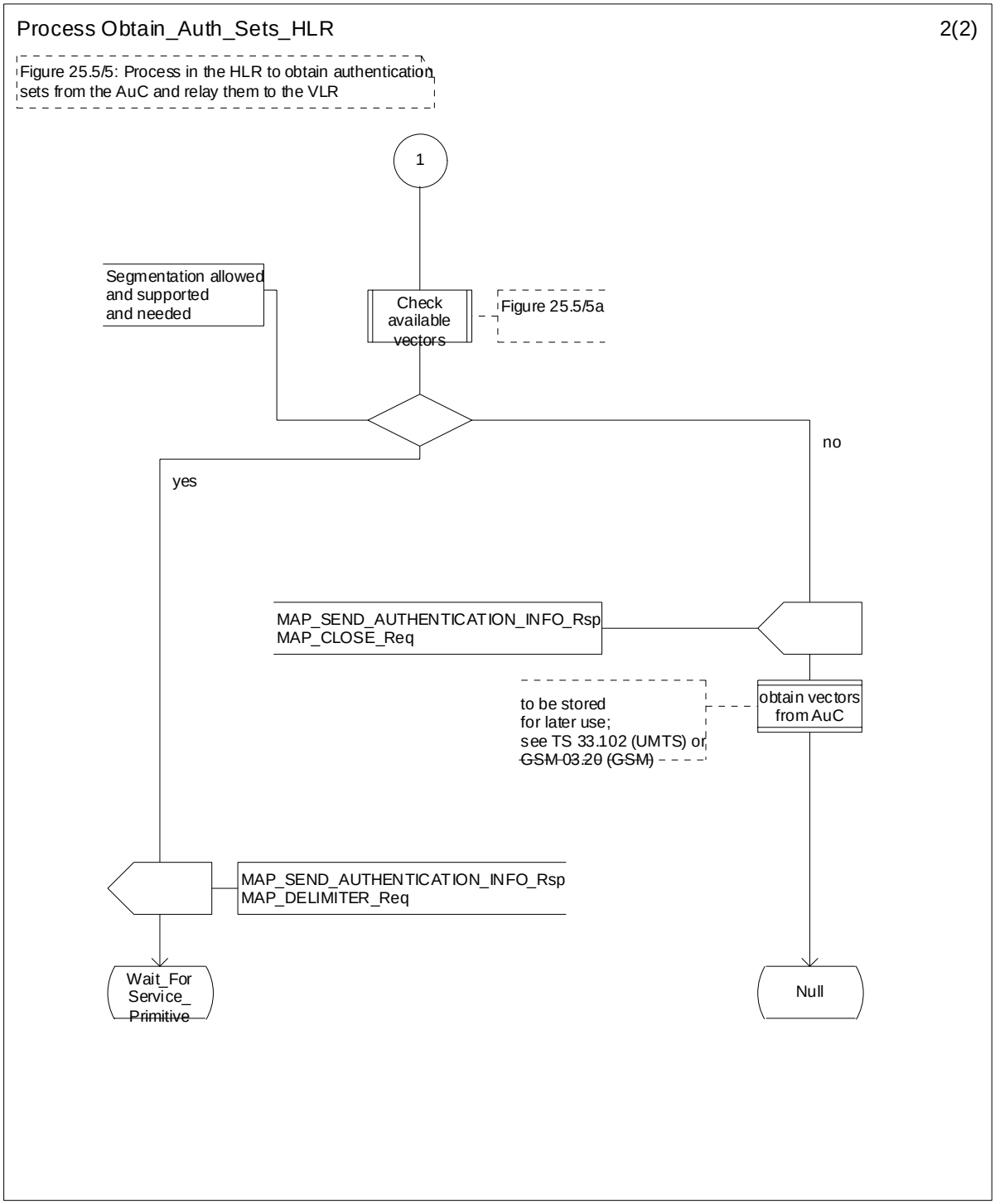


Figure 25.5/5 (sheet 2 of 2): Process Obtain_Auth_Sets_HLR

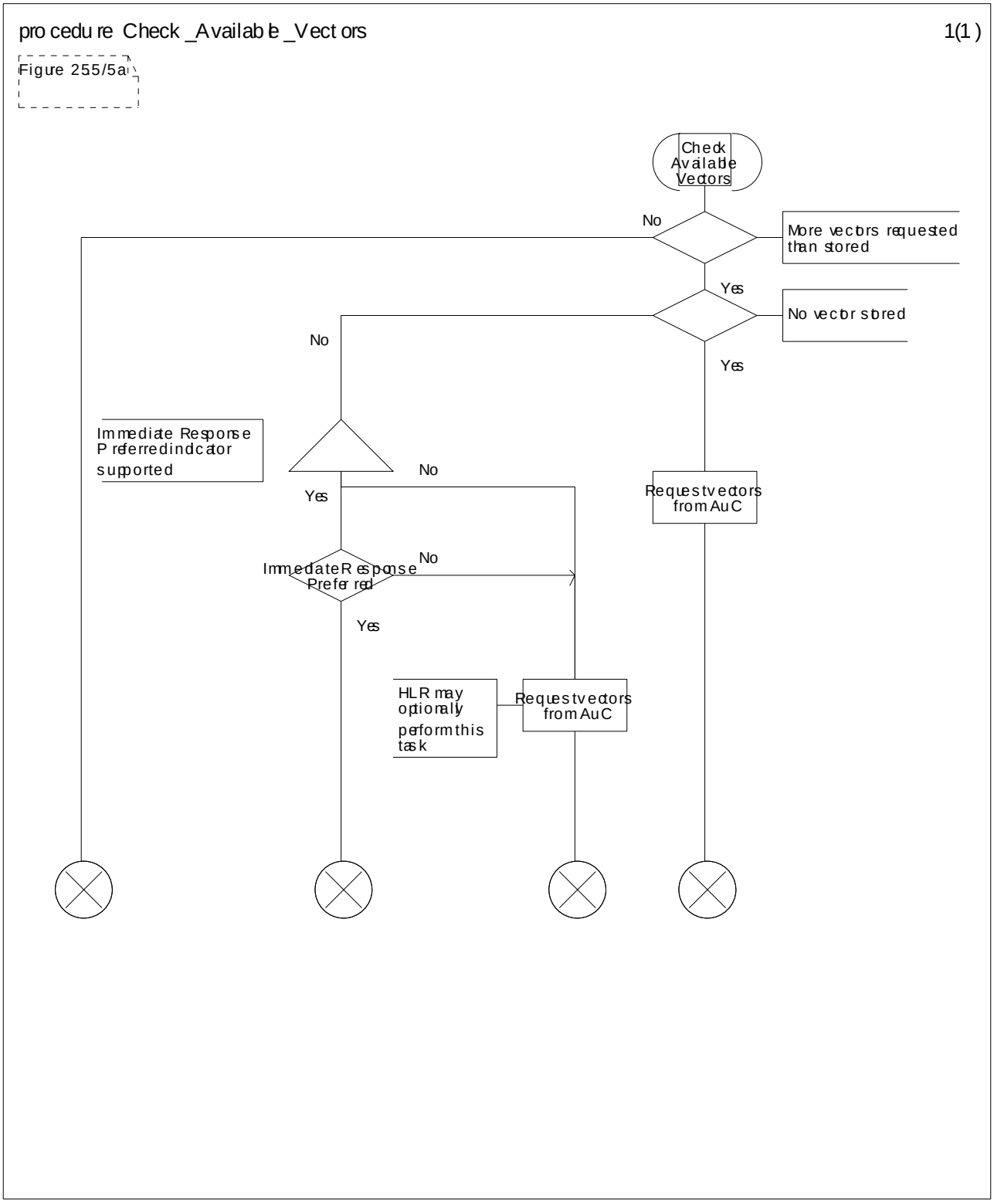


Figure 25.5/6: Procedure Check_Available_Vectors

25.5.6 Process Obtain_Authent_Para_SGSN

For authentication procedure description see 3GPP TS 23.060[104] and 3GPP TS 24.008.

This Process is used by the SGSN to request authentication vectors from the HLR.

If the SGSN does not know the subscriber's HLR address (e.g. no IMSI translation exists), the Authentication Parameter negative response with error “Unknown HLR” is returned to the requesting process.

Otherwise, 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 2dialogue is to be used, the SGSN performs the equivalent MAP version 1 or 2dialogue. 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 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, transferred by means of the TC-RESULT-L service, 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_AUTHENTIFICATION_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(3)

Figure 25.5/6: Process to obtain authentication parameters from the HLR to the SGSN

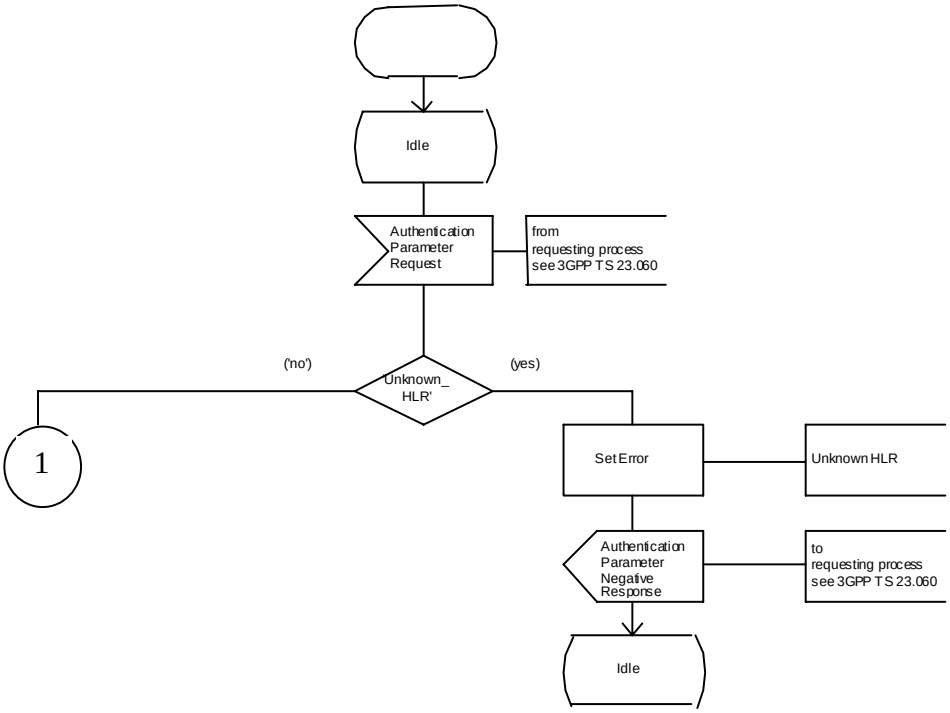


Figure 25.5/6 (sheet 1 of 3): Process Obtain_Authen_Para_SGSN

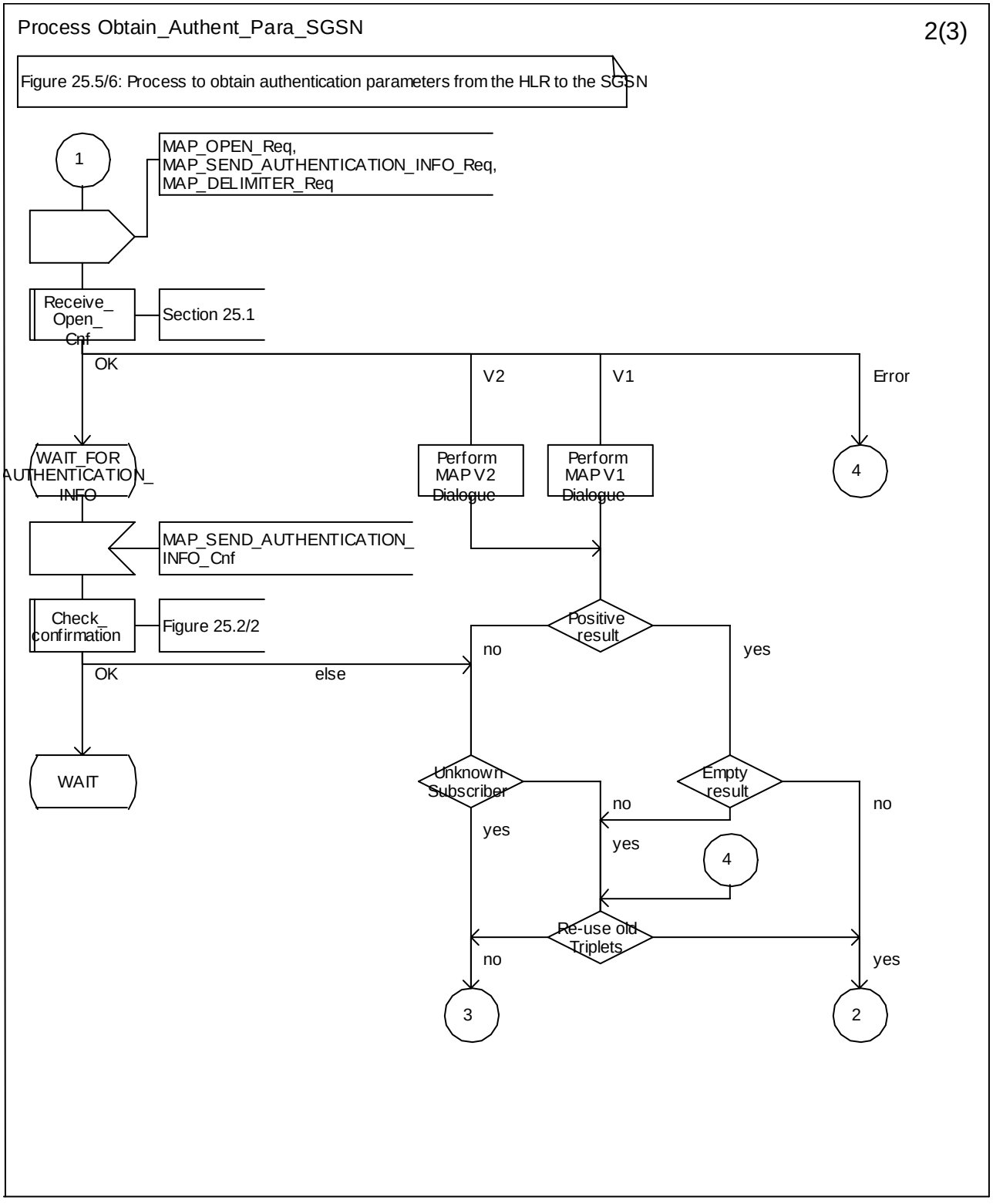


Figure 25.5/6 (sheet 2 of 3): Process Obtain_Authen_Para_SGSN

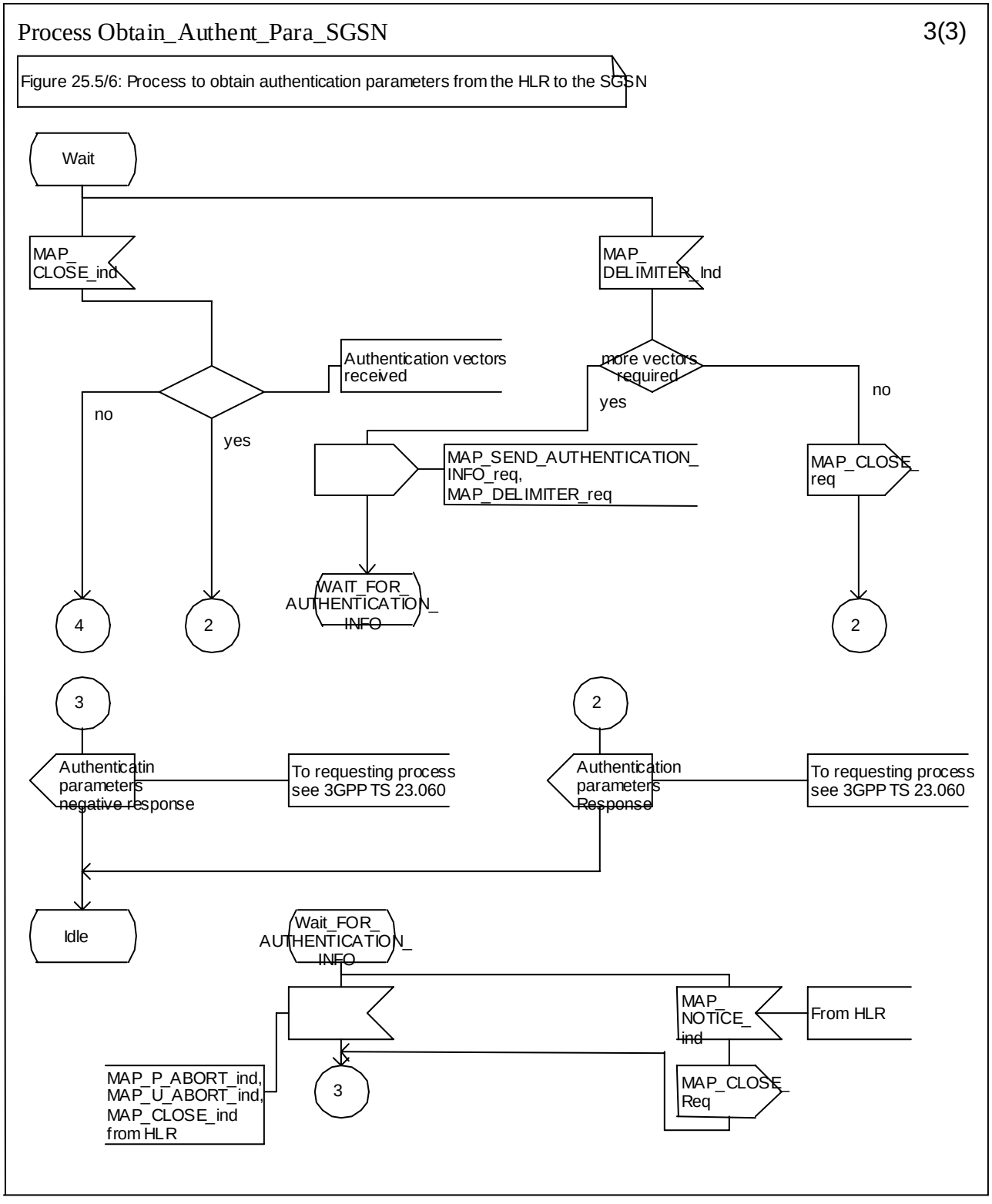


Figure 25.5/6 (sheet 3 of 3): Process Obtain_Authen_Para_SGSN

25.5.7 Process Authentication_Failure_Report

25.5.7.1 General

The Authentication Failure Report procedure is used to notify a HLR about the occurrence of an authentication failure in the SGSN or VLR.

The procedure is shown in figure 25.5/7.

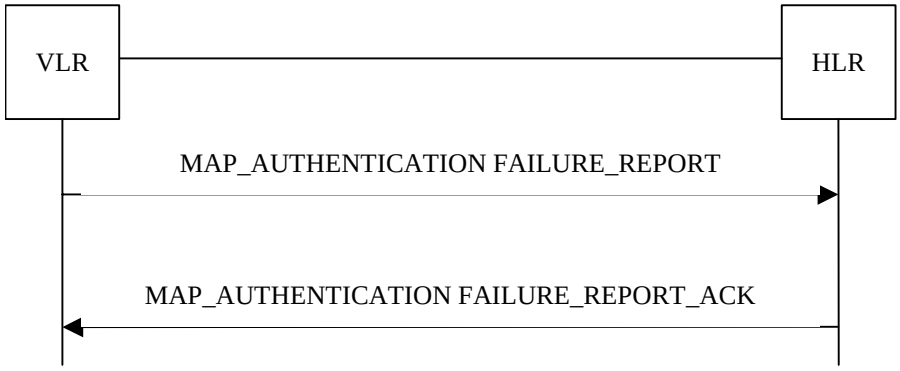


Figure 25.5/7: Message Flows to Authentication Failure Report

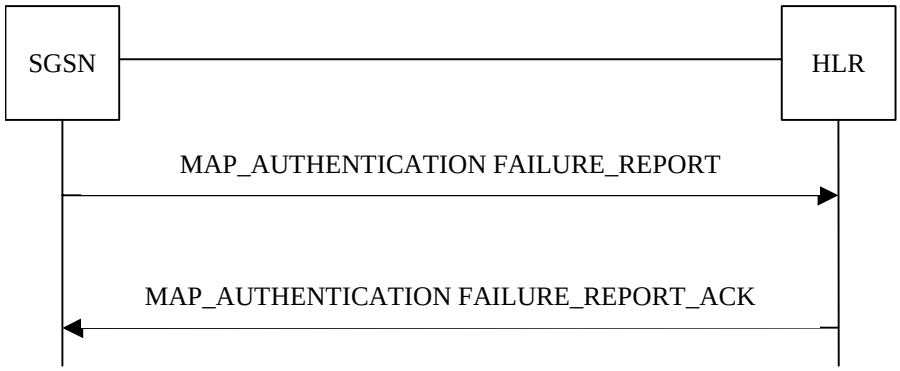


Figure 25.6/7: Message Flows to Authentication Failure Report

25.5.7.2 Process in the VLR

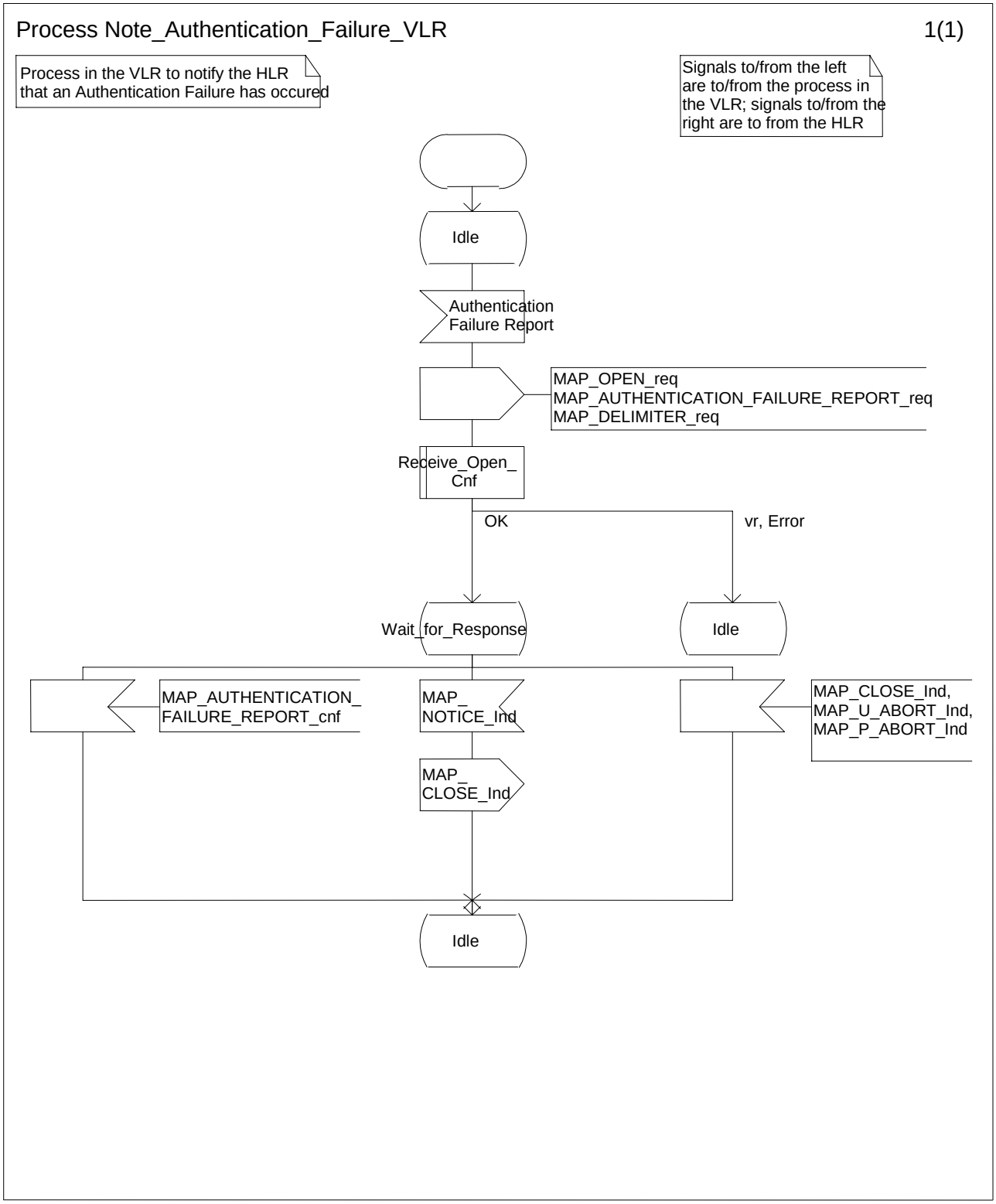


Figure 25.6/8: Process Note_Authentication_Failure_VLR

25.5.7.3 Process in the SGSN

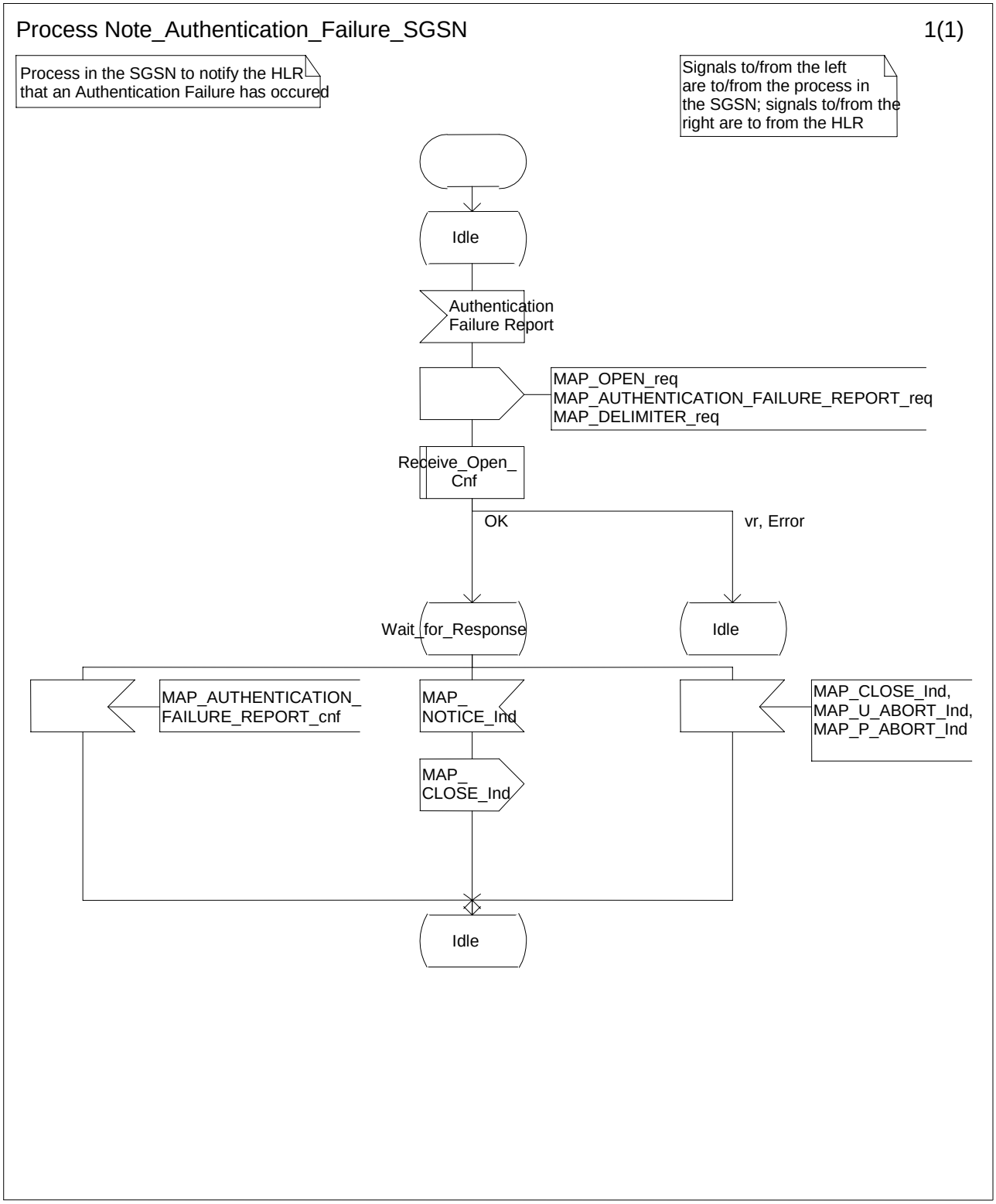


Figure 25.6/9: Process Note_Authentication_Failure_SGSN

25.5.7.4 Process in the HLR

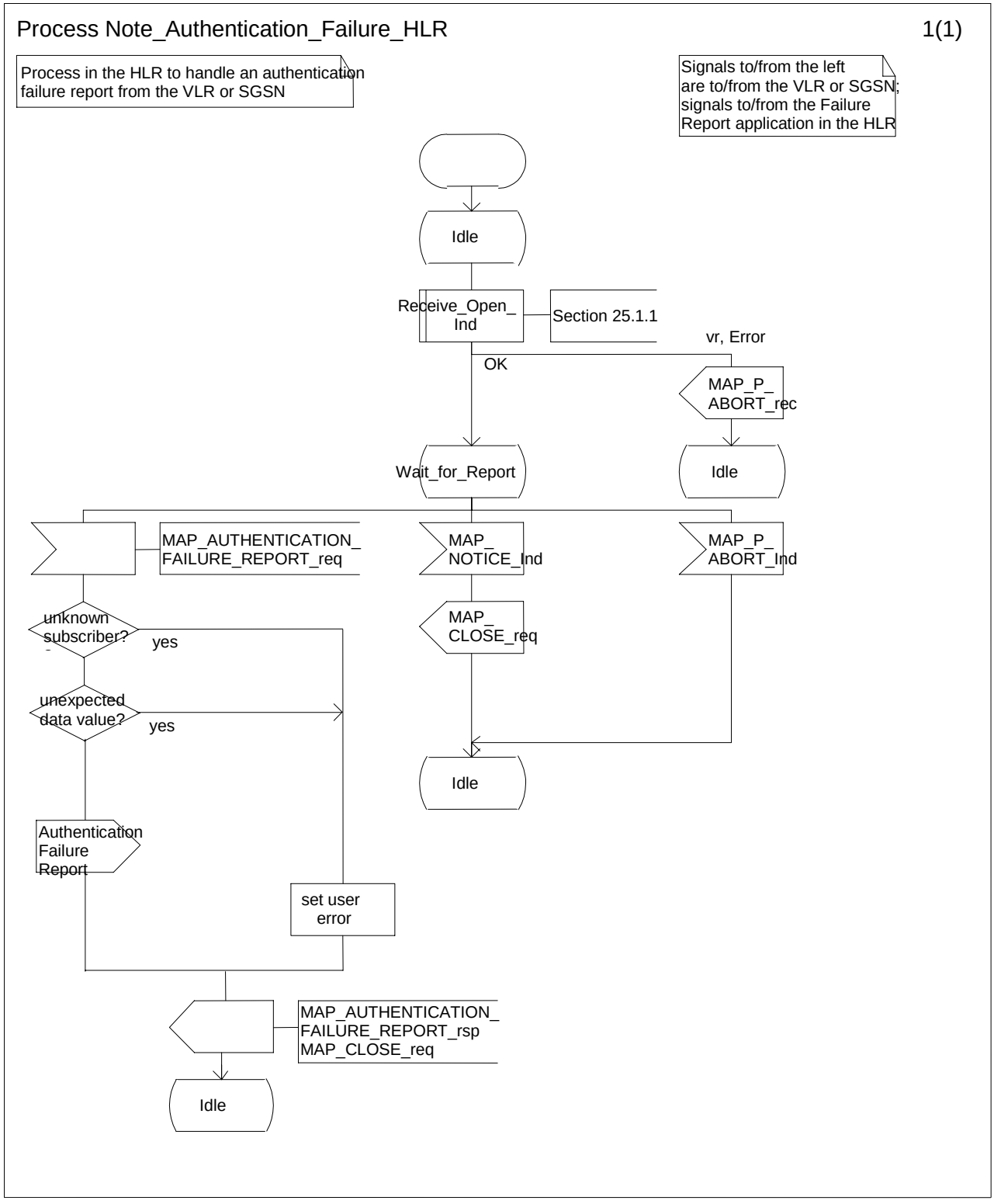


Figure 25.6/10: Process Note_Authentication_Failure_HLR

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, clause 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 3GPP TS 22.016 [7];

- 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.

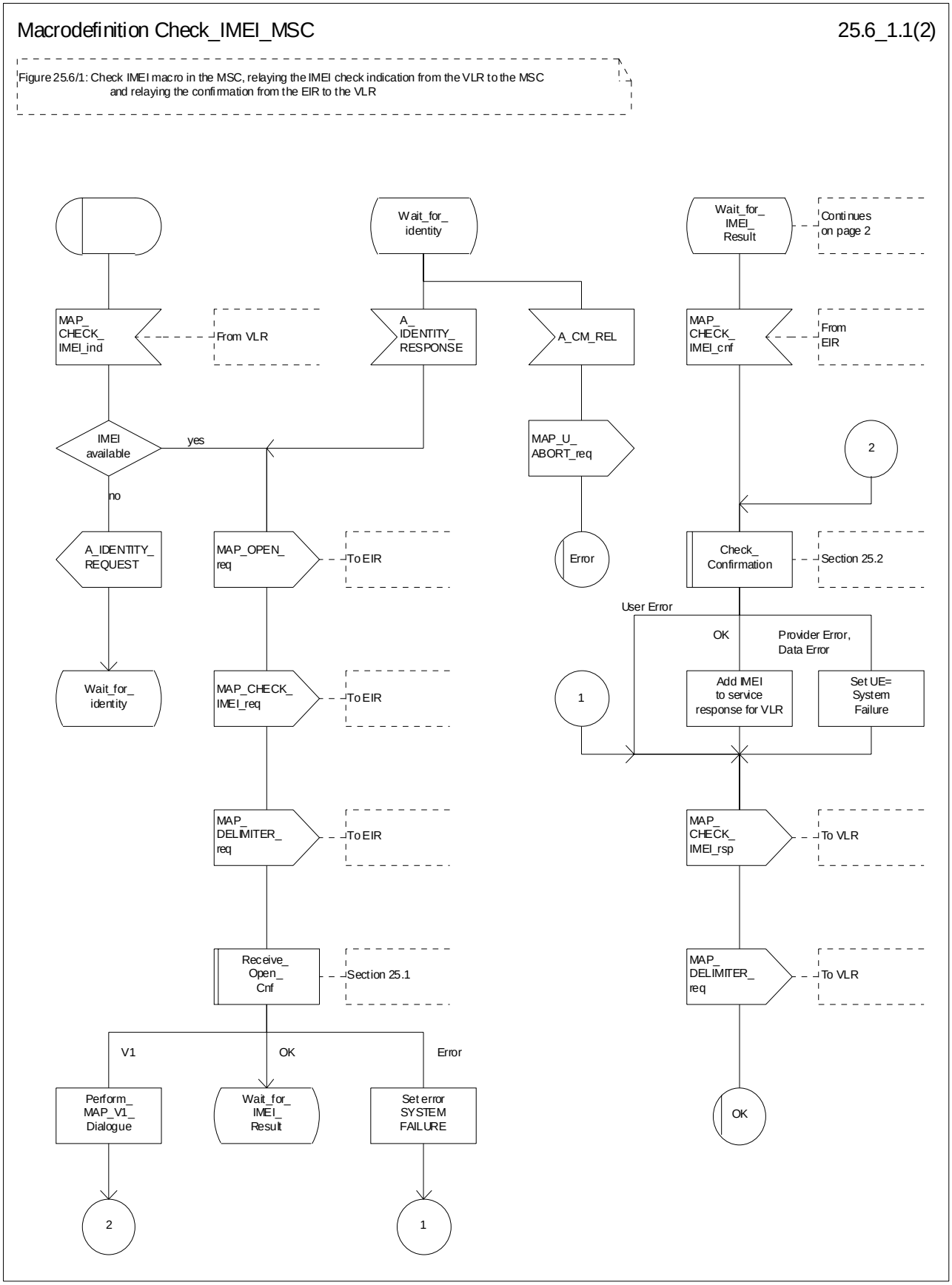


Figure 25.6/1 (sheet 1 of 2): Process Check_IMEI_MSC

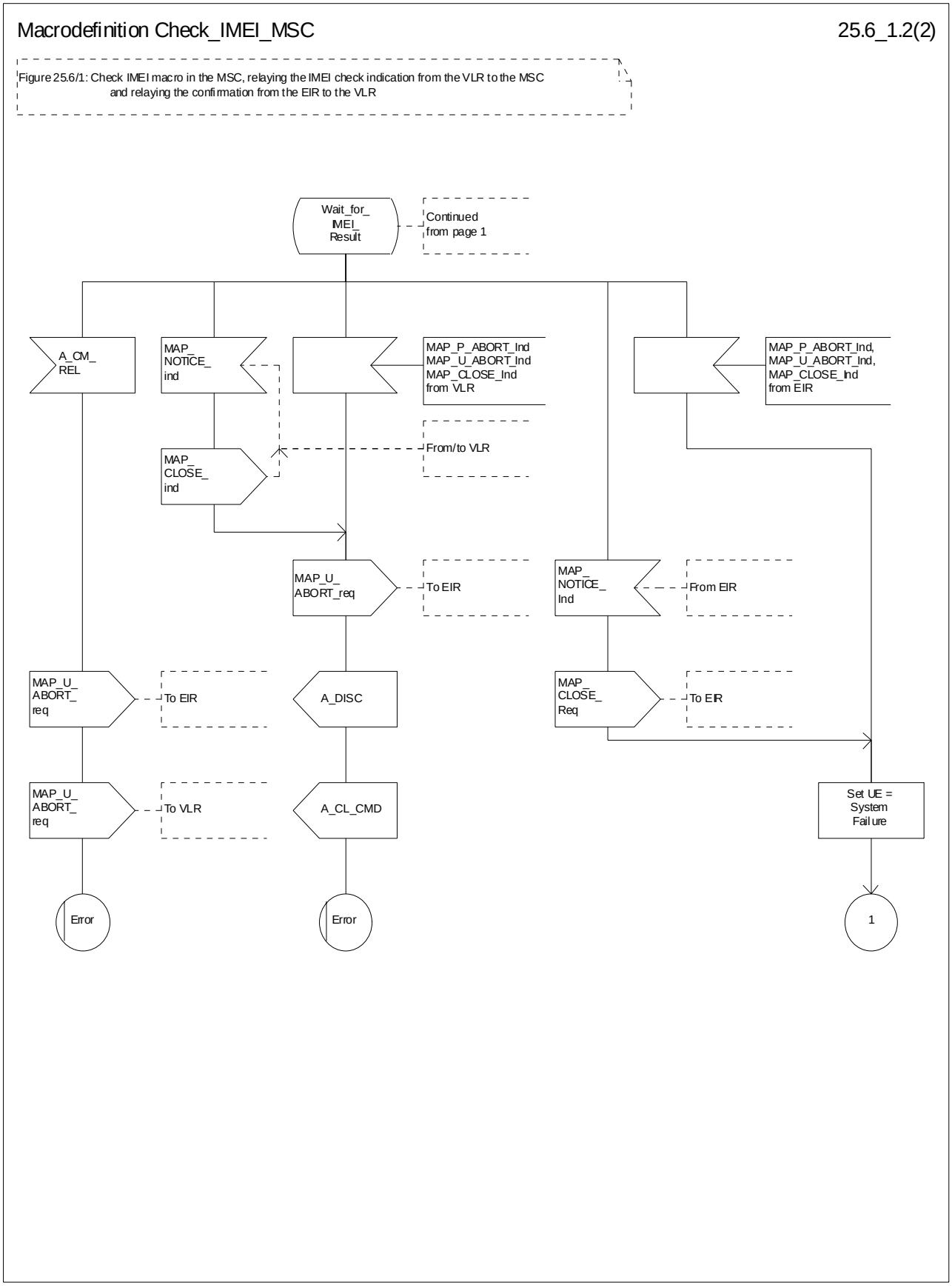


Figure 25.6/1 (sheet 2 of 2): Process Check_IMEI_MSC

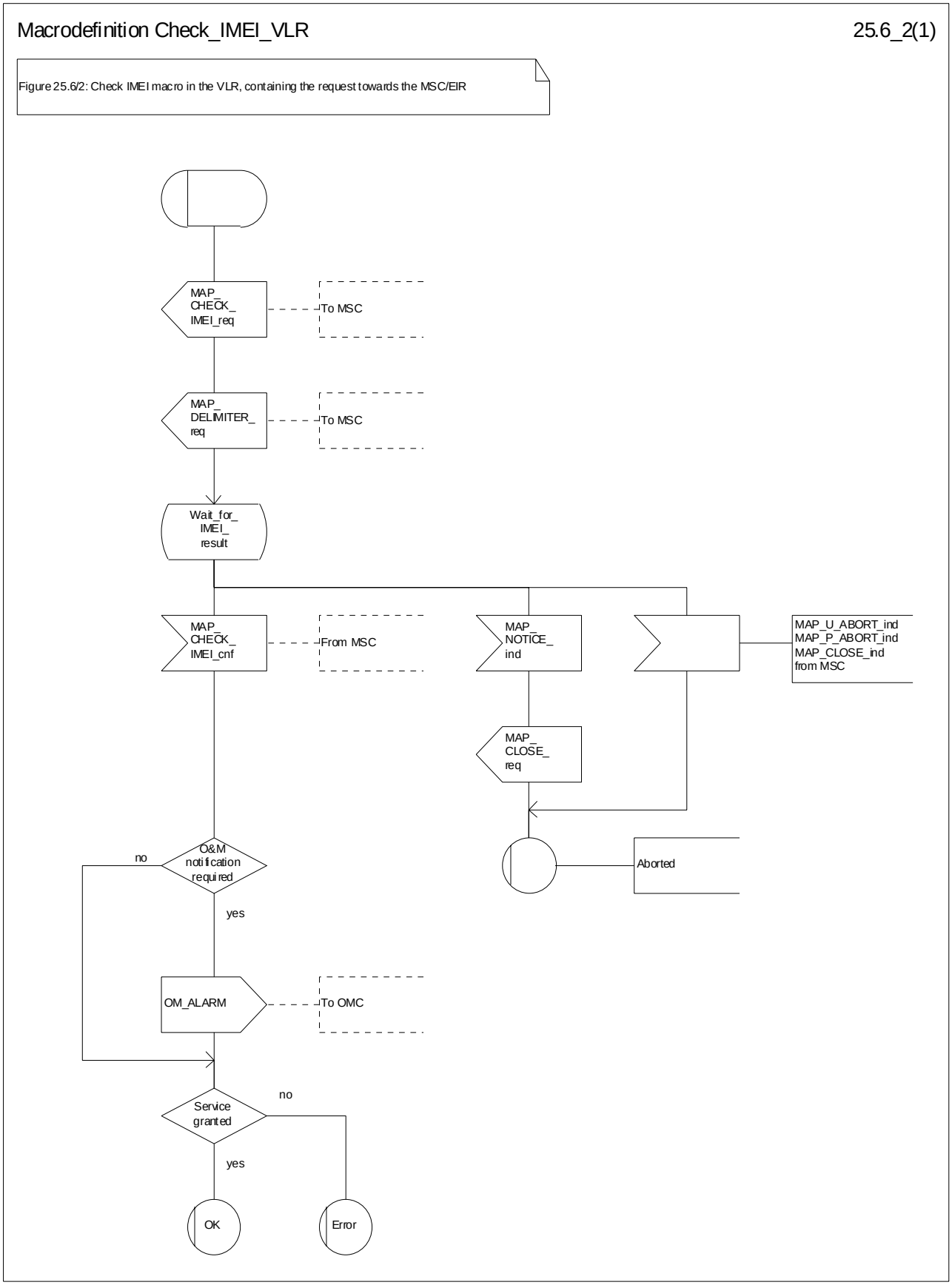


Figure 25.6/2: Process Check_IMEI_VLR

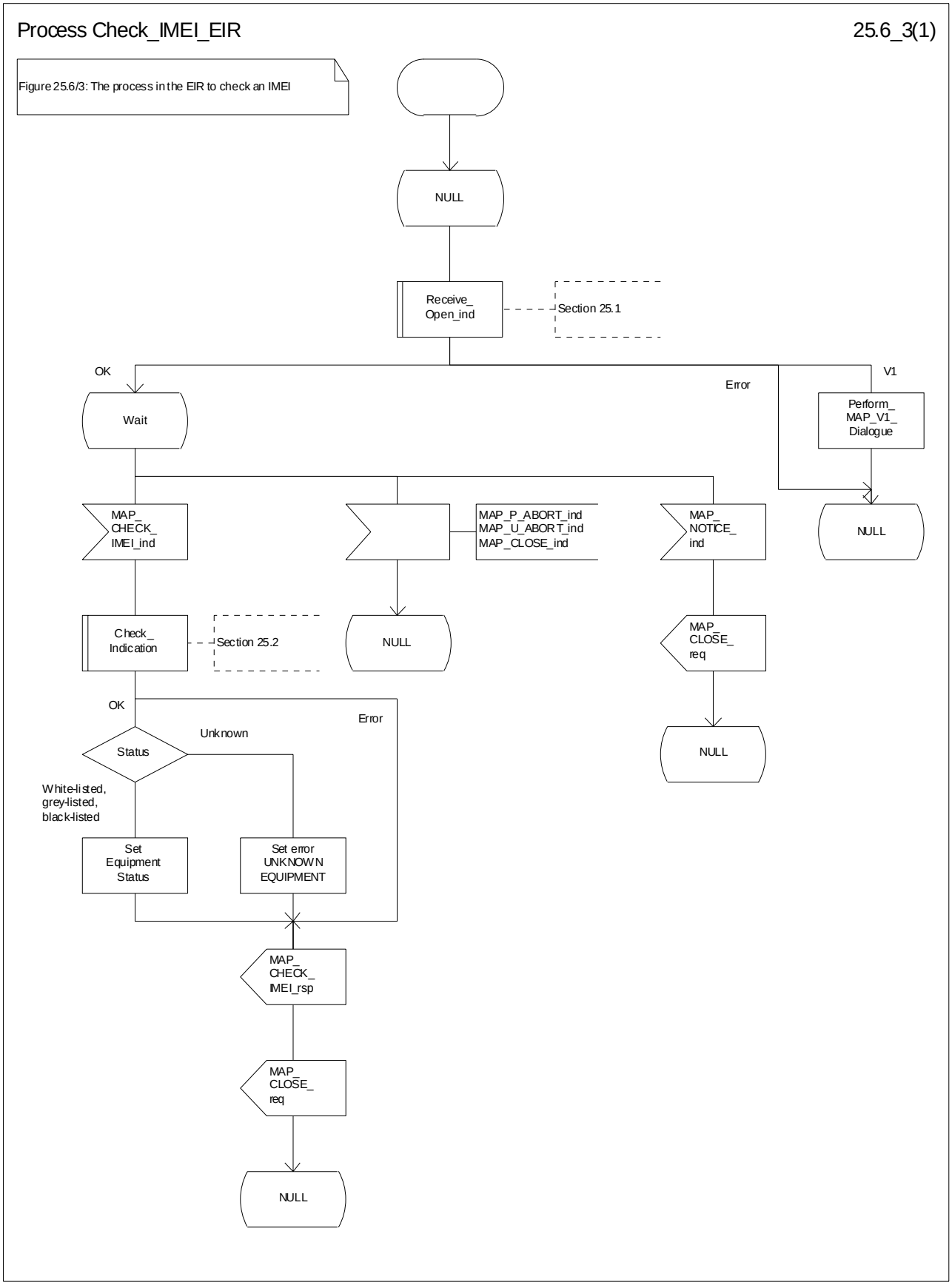


Figure 25.6/3: Process Check_IMEI_EIR

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

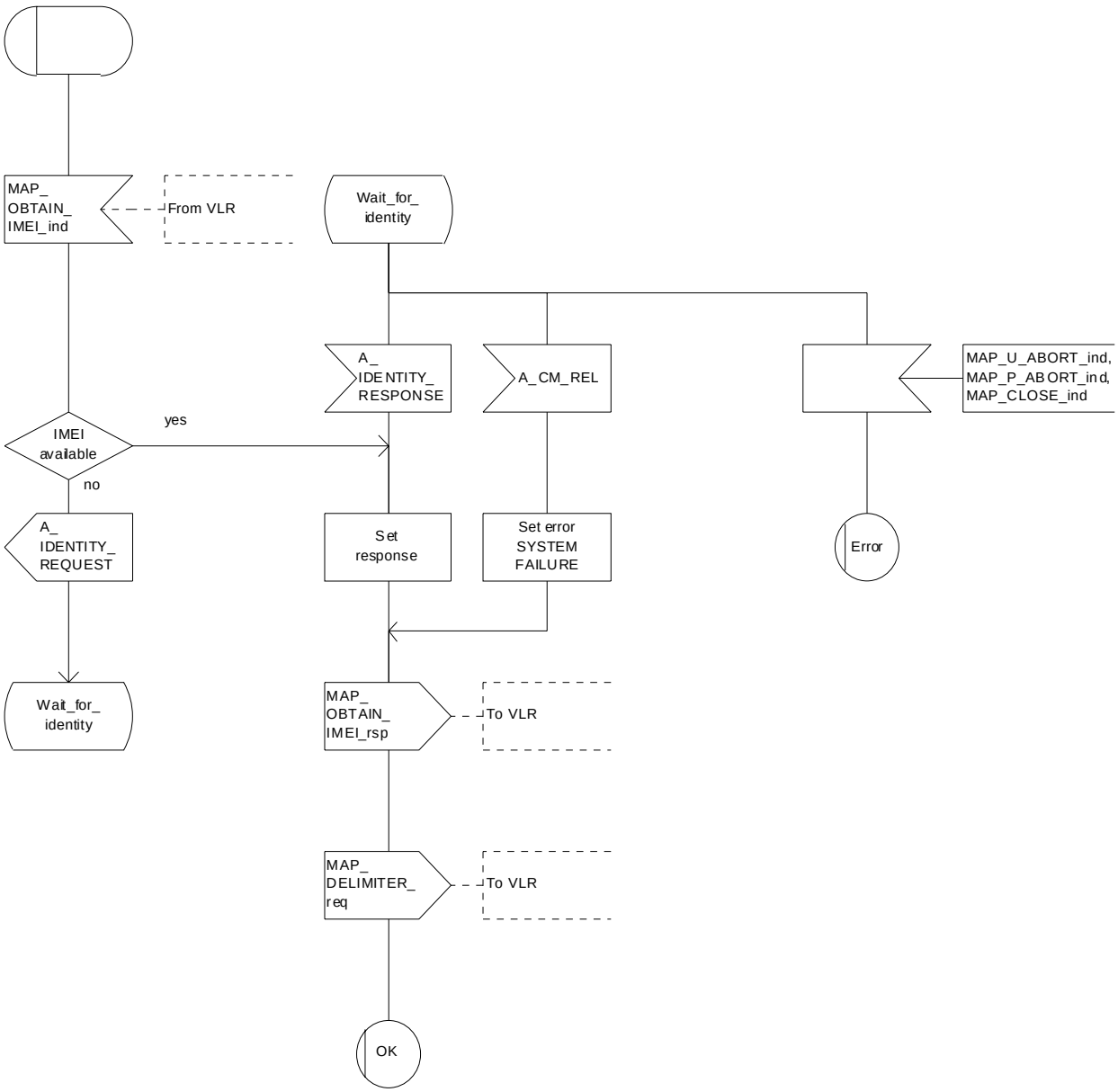


Figure 25.6/4: Process Obtain_IMEI_MSC

Macrodefinition Obtain_IMEI_VLR 25.6_5(1)

Figure 25.6/5: Obtain IMEI macro in the VLR, controlling the request towards the MSC

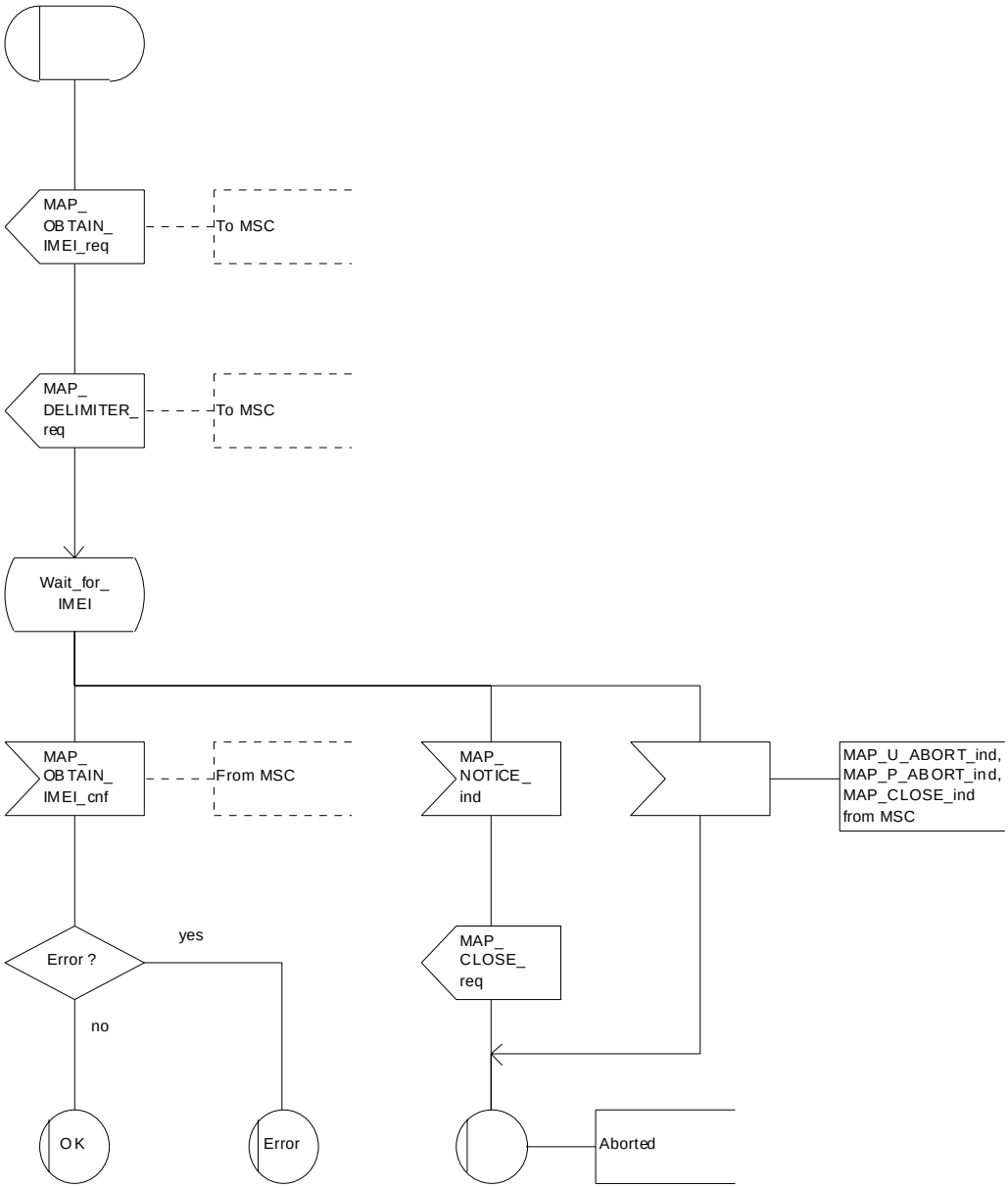


Figure 25.6/5: Process Obtain_IMEI_VLR

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.

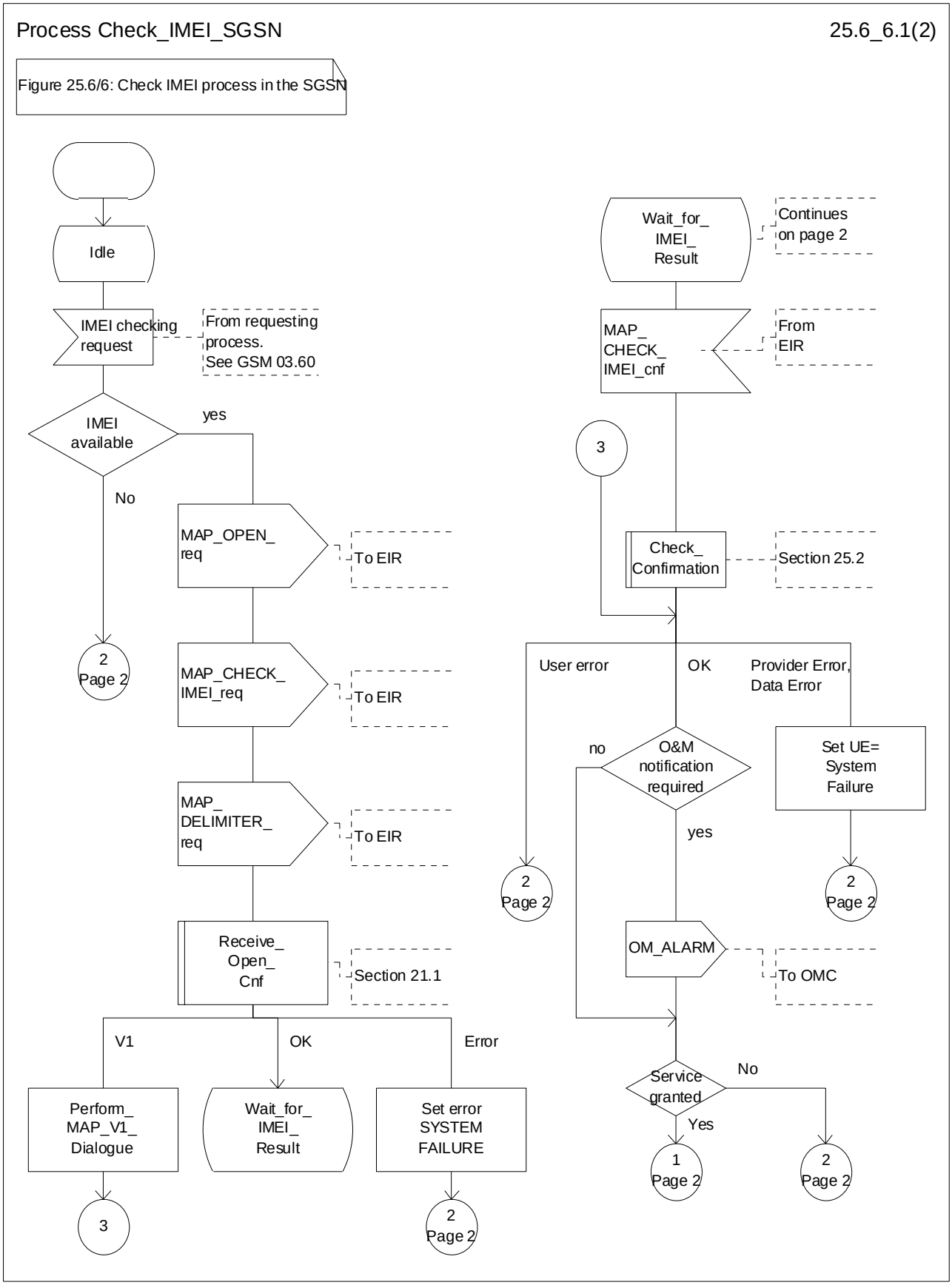


Figure 25.6/6 (sheet 1 of 2): Process Check_IMEI_SGSN

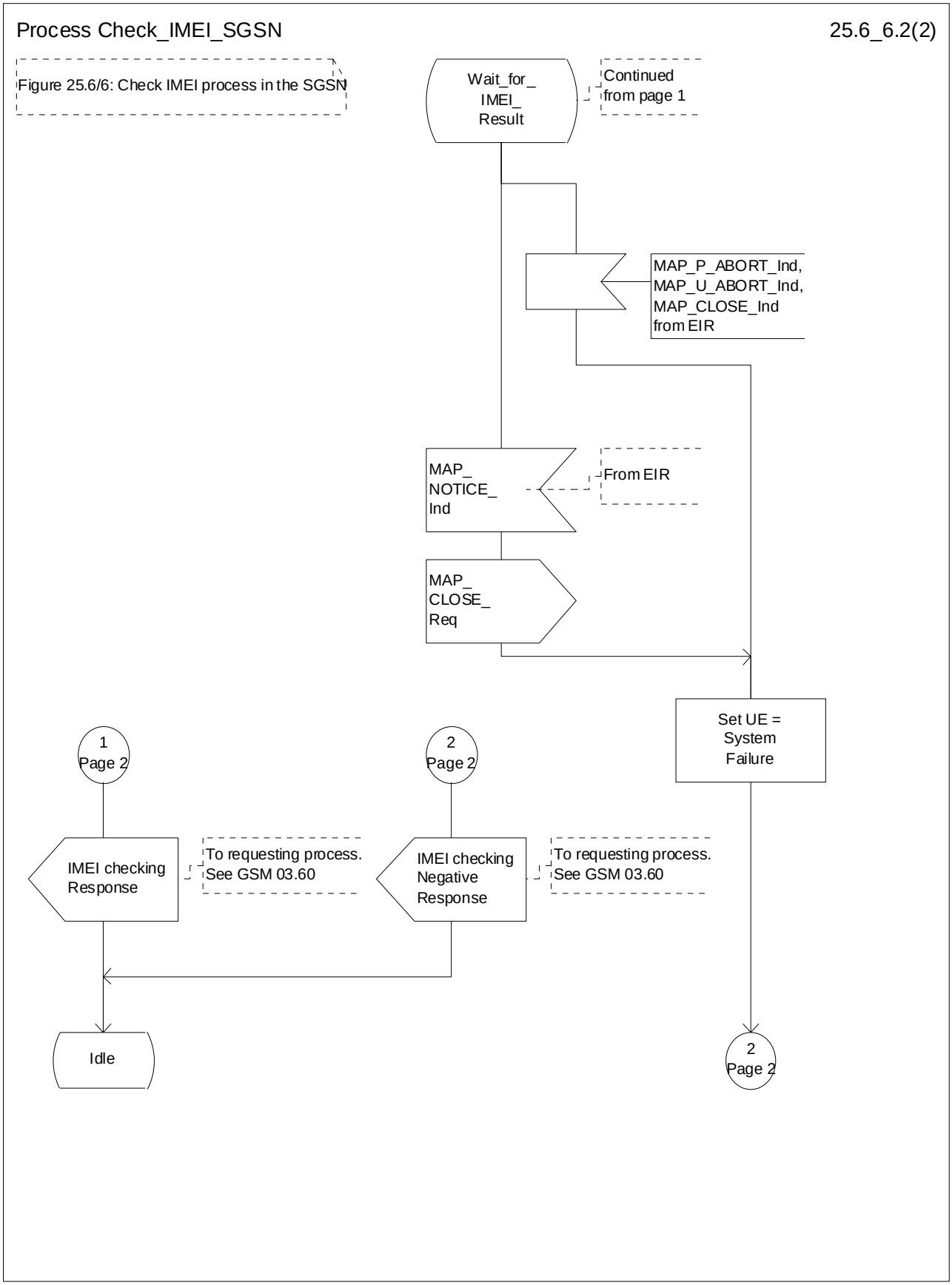


Figure 25.6/6 (sheet 2 of 2): Process Check_IMEI_SGSN

25.7 Insert Subscriber Data Macros

25.7.1 Macro Insert_Subs_Data_VLR

This macro describes the reception of the InsertSubscriberData service indication. This macro is used by any procedure that triggers the reception of subscriber data (e.g. Update Location or Restore Data).

If the VLR does not support any basic or supplementary service or the network feature Operator Determined Barring, or there is a problem with Regional Subscription Data then it reports it to the HLR.

If the entire MSC area is restricted due to regional subscription this is reported to the HLR.

The SDL diagram is shown in figure 25.7/1.

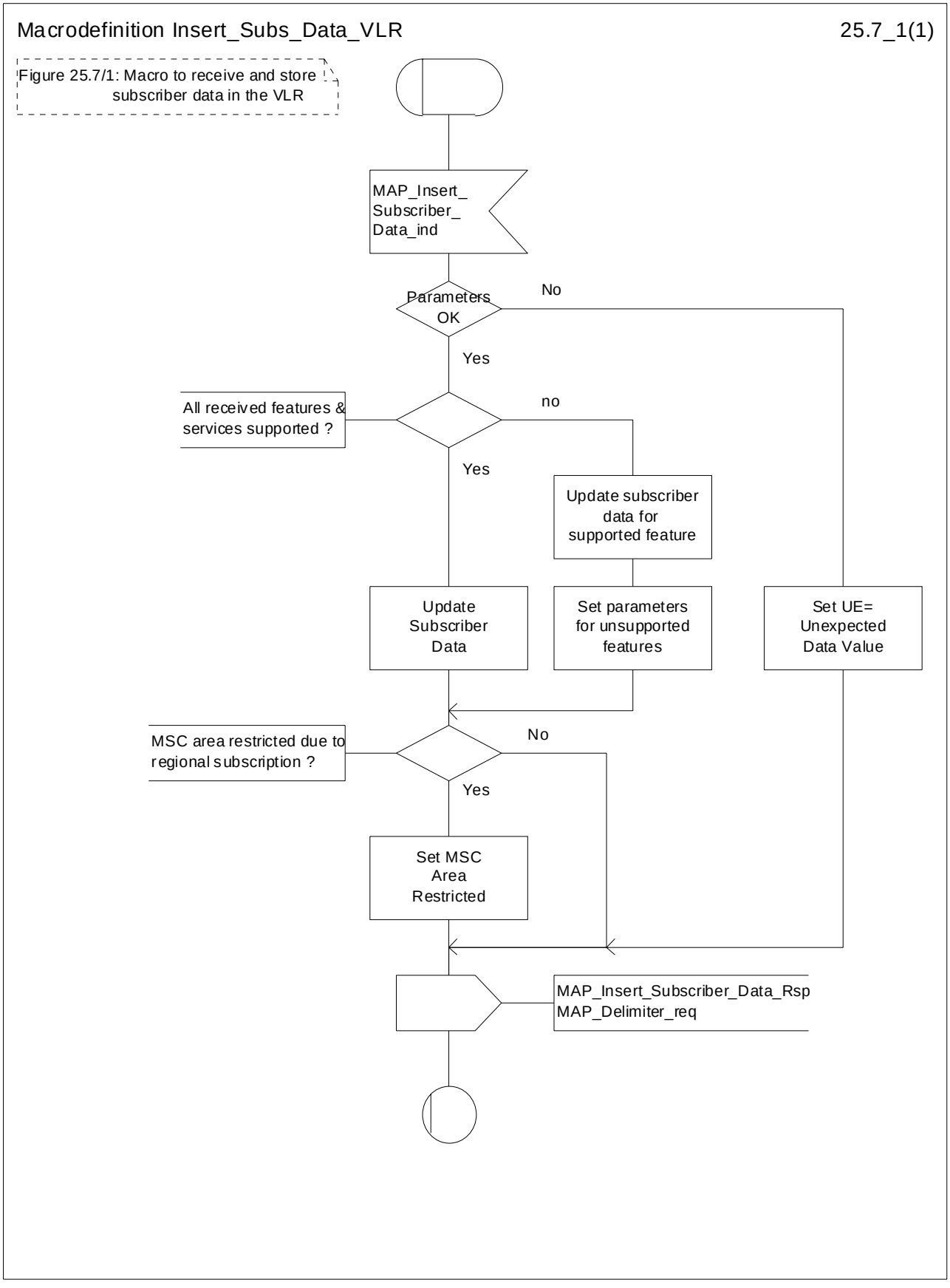


Figure 25.7/1: Macro Insert_Subs_Data_VLR

25.7.2 Process Insert_Subs_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_Subs_Data" (see clause 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_Subs_Data_Cnf" and “Wait_for_Insert_GPRS_Subs_Data_Cnf” (see clauses 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 3GPP 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.

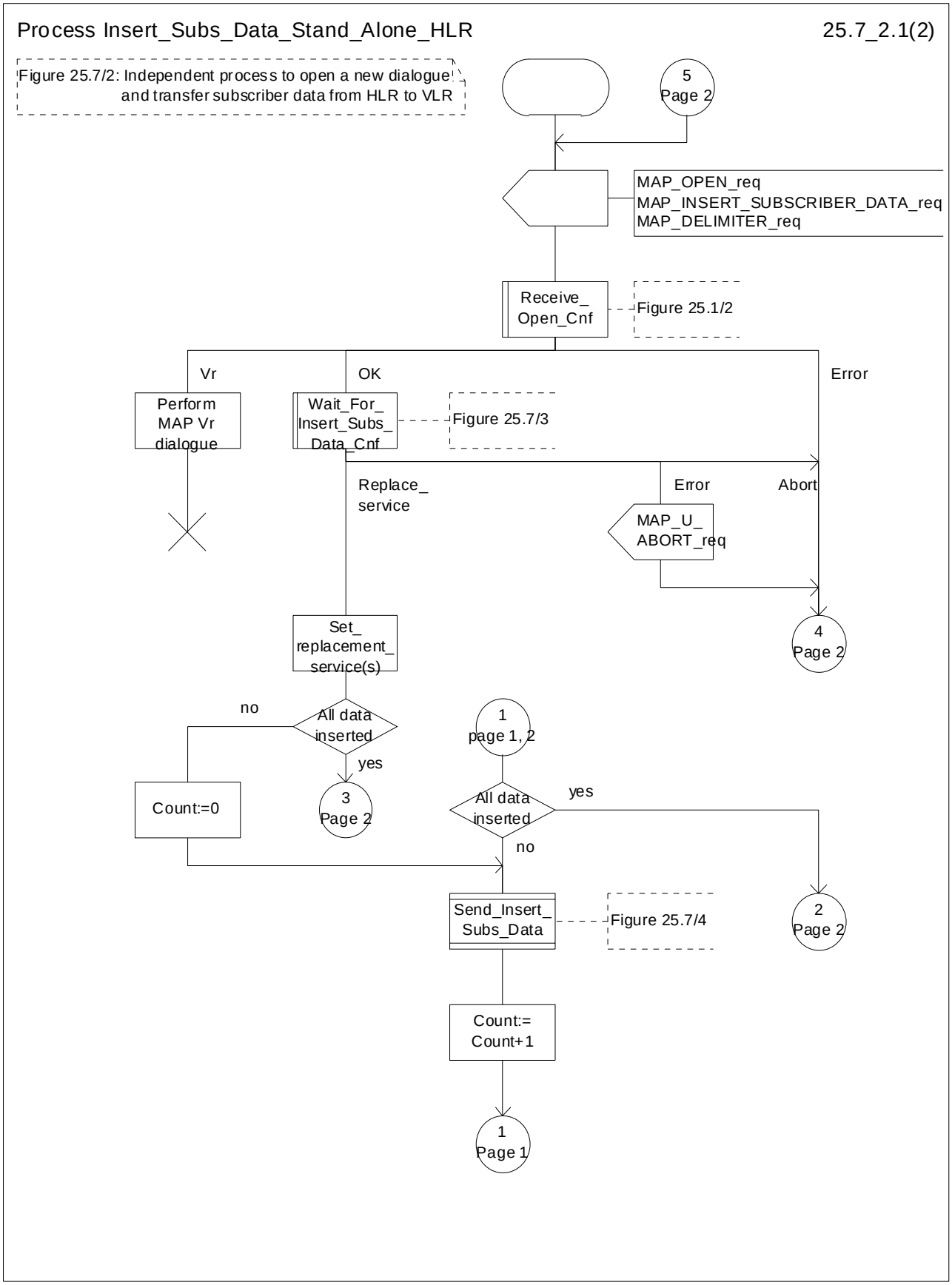


Figure 25.7/2 (sheet 1 of 2): Process Insert_Subs_Data_Stand_Alone_HLR

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

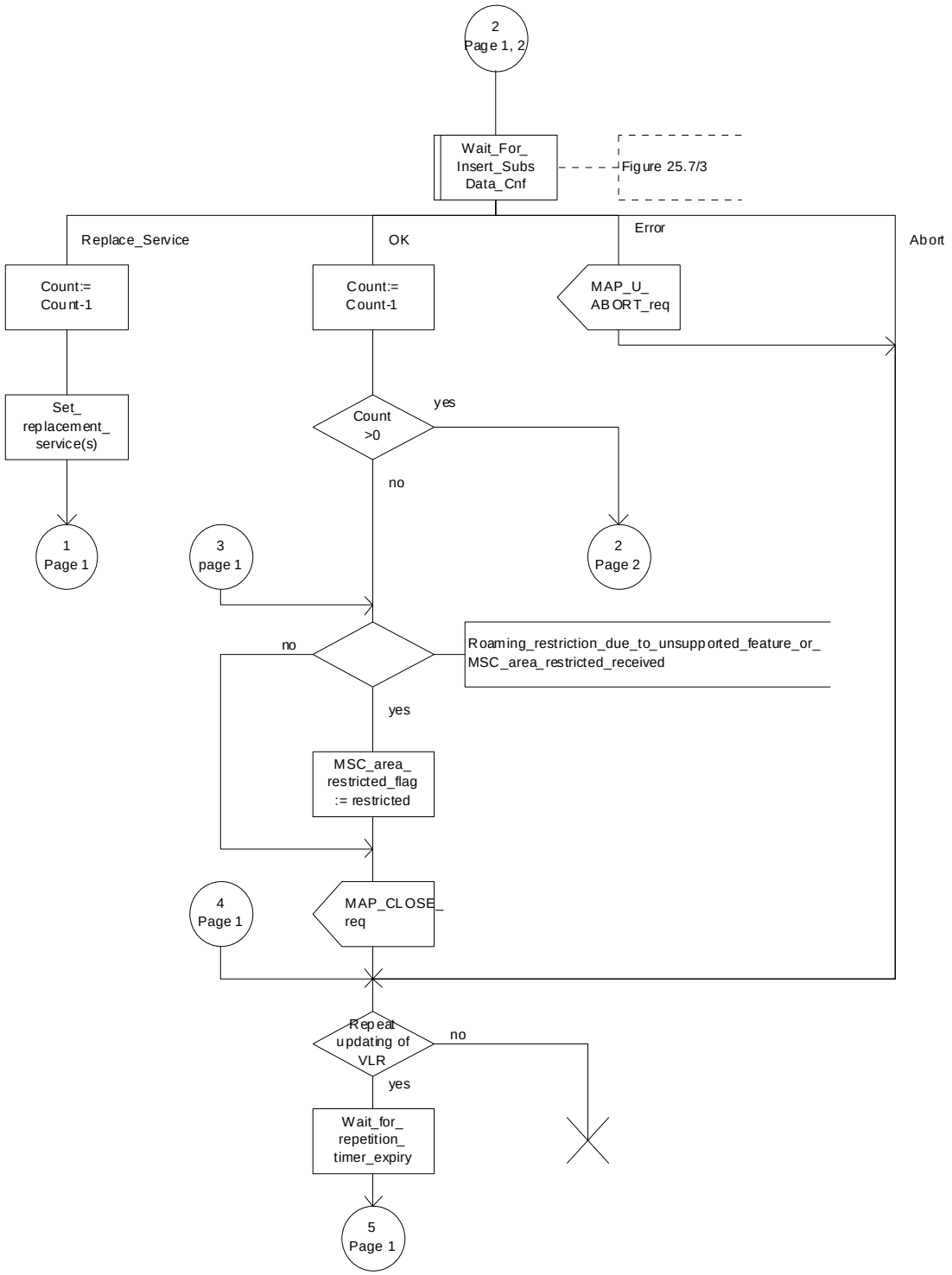


Figure 25.7/2 (sheet 2 of 2): Process Insert_Subs_Data_Stand_Alone_HLR

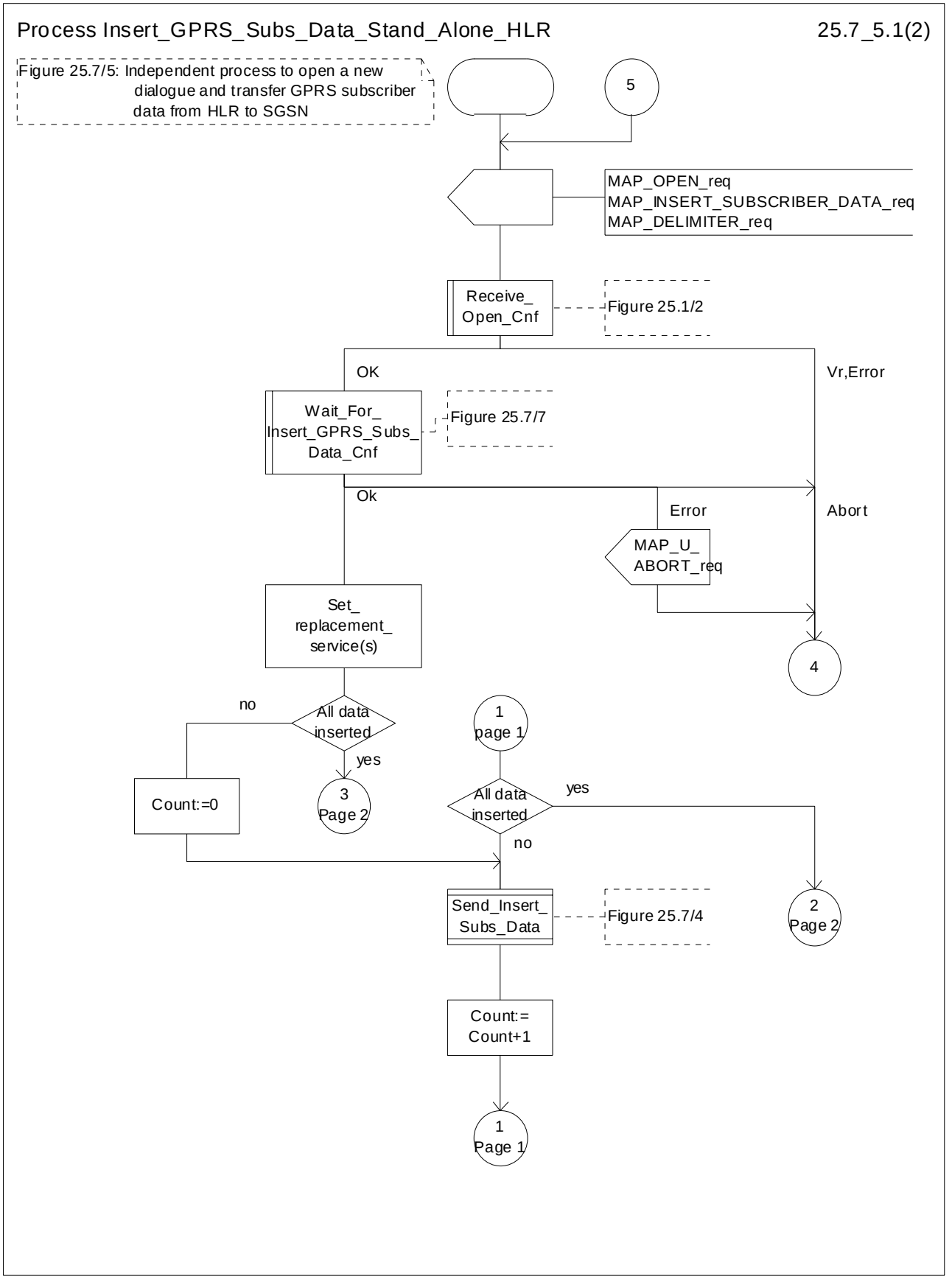


Figure 25.7/5 (sheet 1 of 2): Process Insert_GPRS_Subs_Data_Stand_Alone_HLR

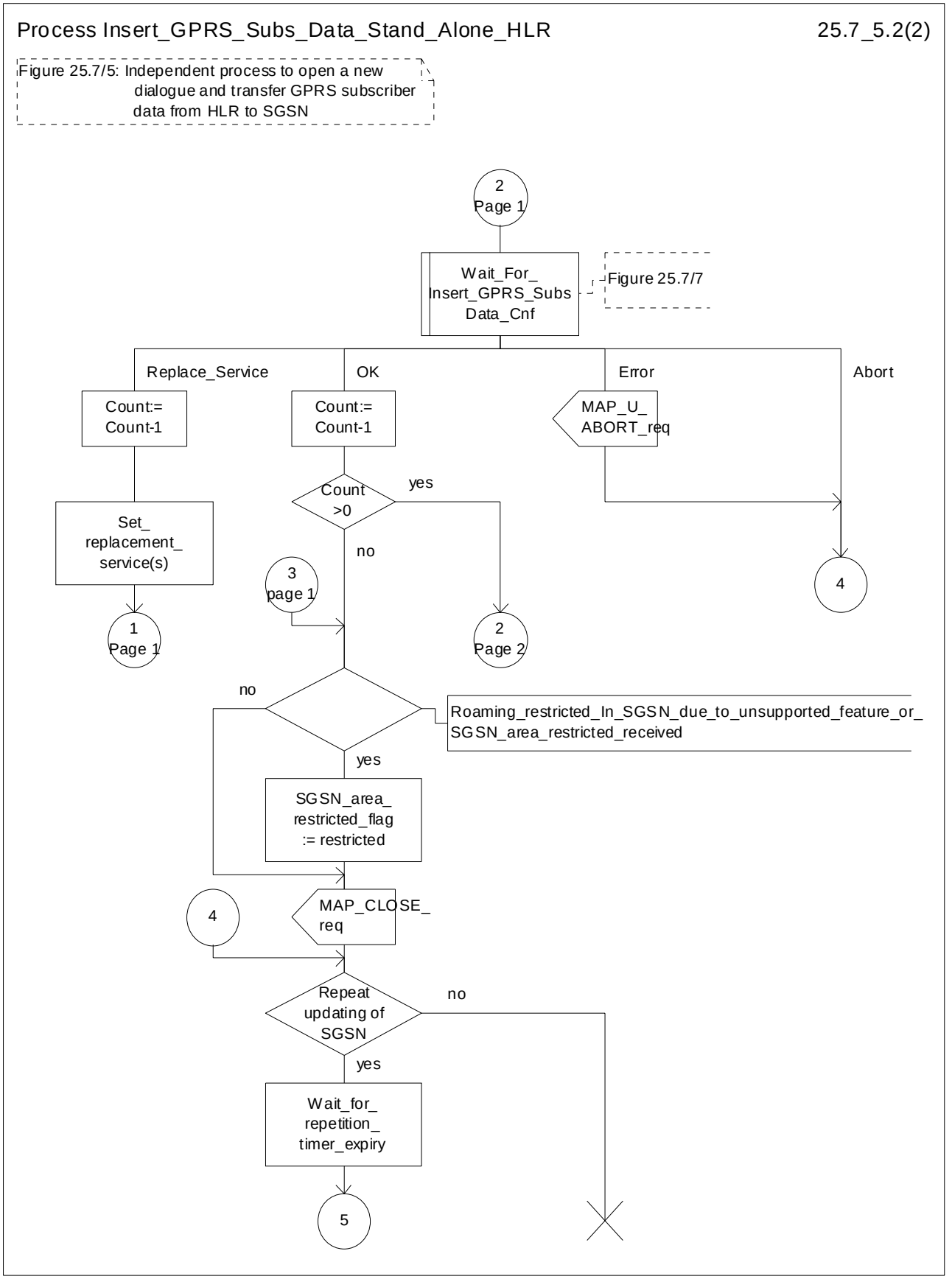


Figure 25.7/5 (sheet 2 of 2): Process Insert_GPRS_Subs_Data_Stand_Alone_HLR

25.7.3 Macro Wait_for_Insert_Subs_Data_Cnf

This macro is used by any process or macro that describes the handling of the reception of the Insert_Subscriber_Data service in HLR 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.

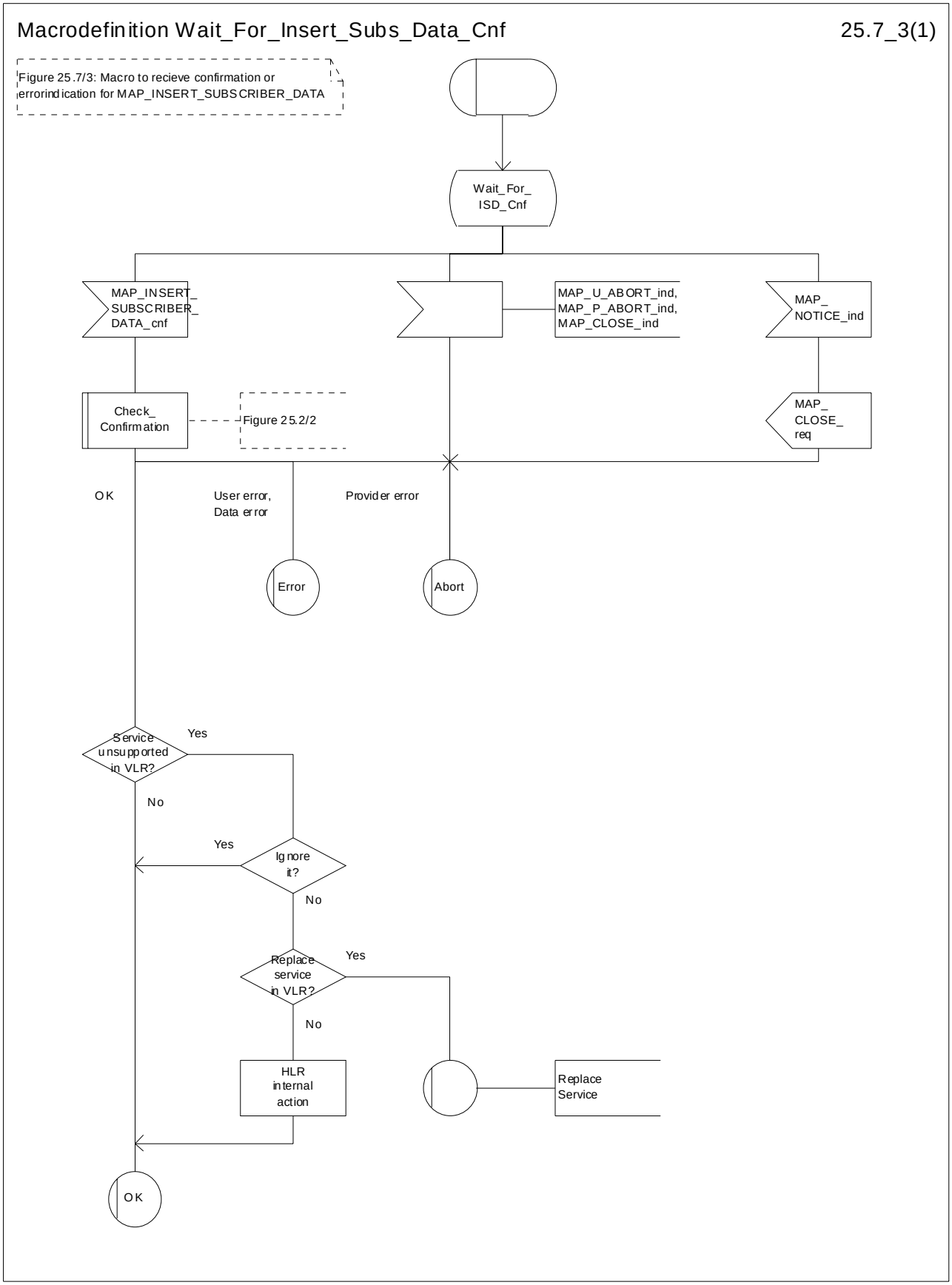


Figure 25.7/3: Macro Wait_for_Insert_Subs_Data_Cnf

25.7.4 Process Send_Insert_Subs_Data

This process is used by any process or macro where the Insert_Subscriber_Data request is sent to VLR or to SGSN.

The SDL diagram is shown in figure 25.7/4.

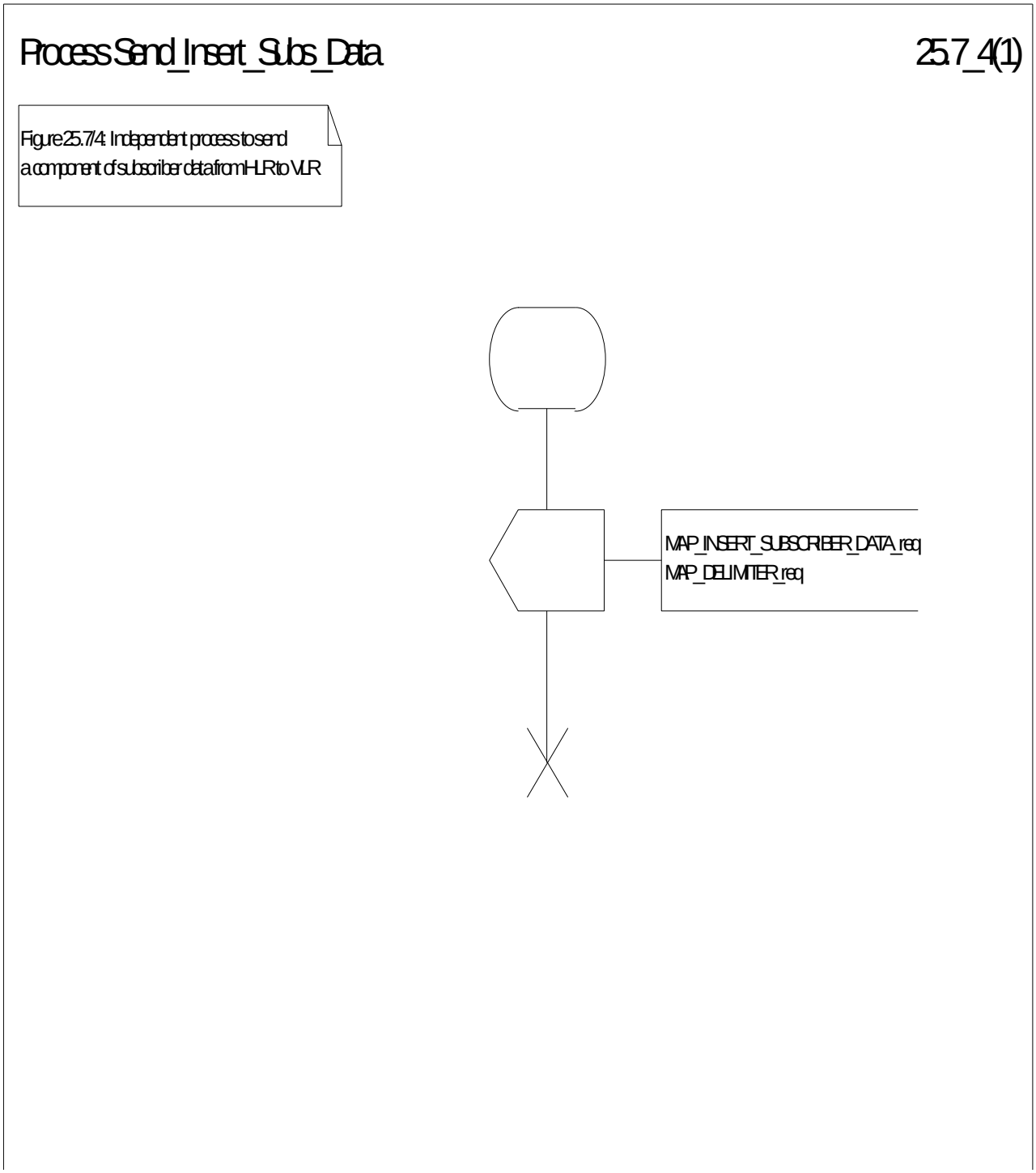


Figure 25.7/4: Process Send_Insert_Subs_Data

25.7.5 Macro Insert_Subs_Data_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.

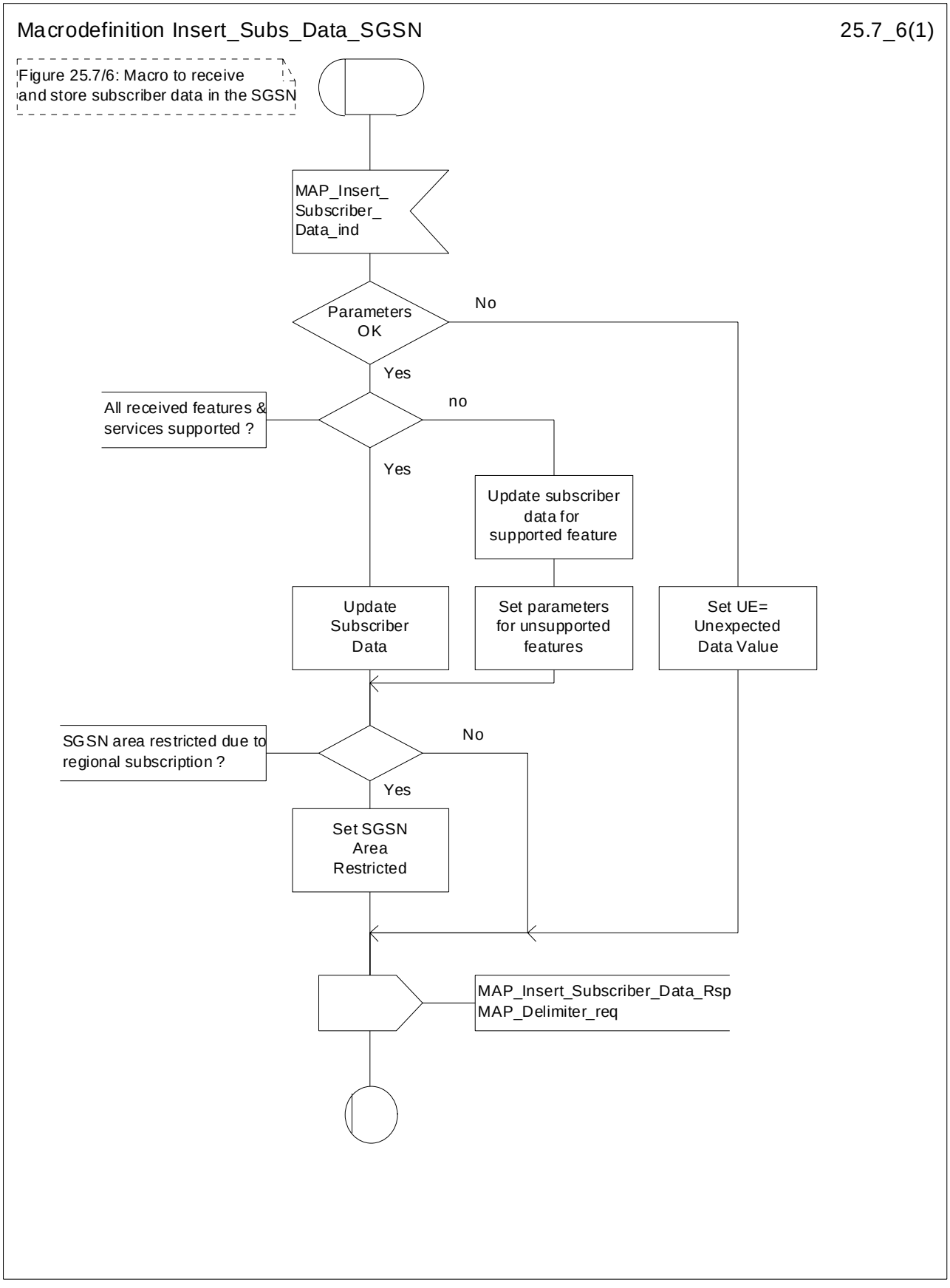


Figure 25.7/6: Macro Insert_Subs_Data_SGSN

25.7.6 Macro Wait_for_Insert_GPRS_Subs_Data_Cnf

This macro is used by any process or macro that describes the handling of the reception of the Insert_Subscriber_Data service in HLR 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.

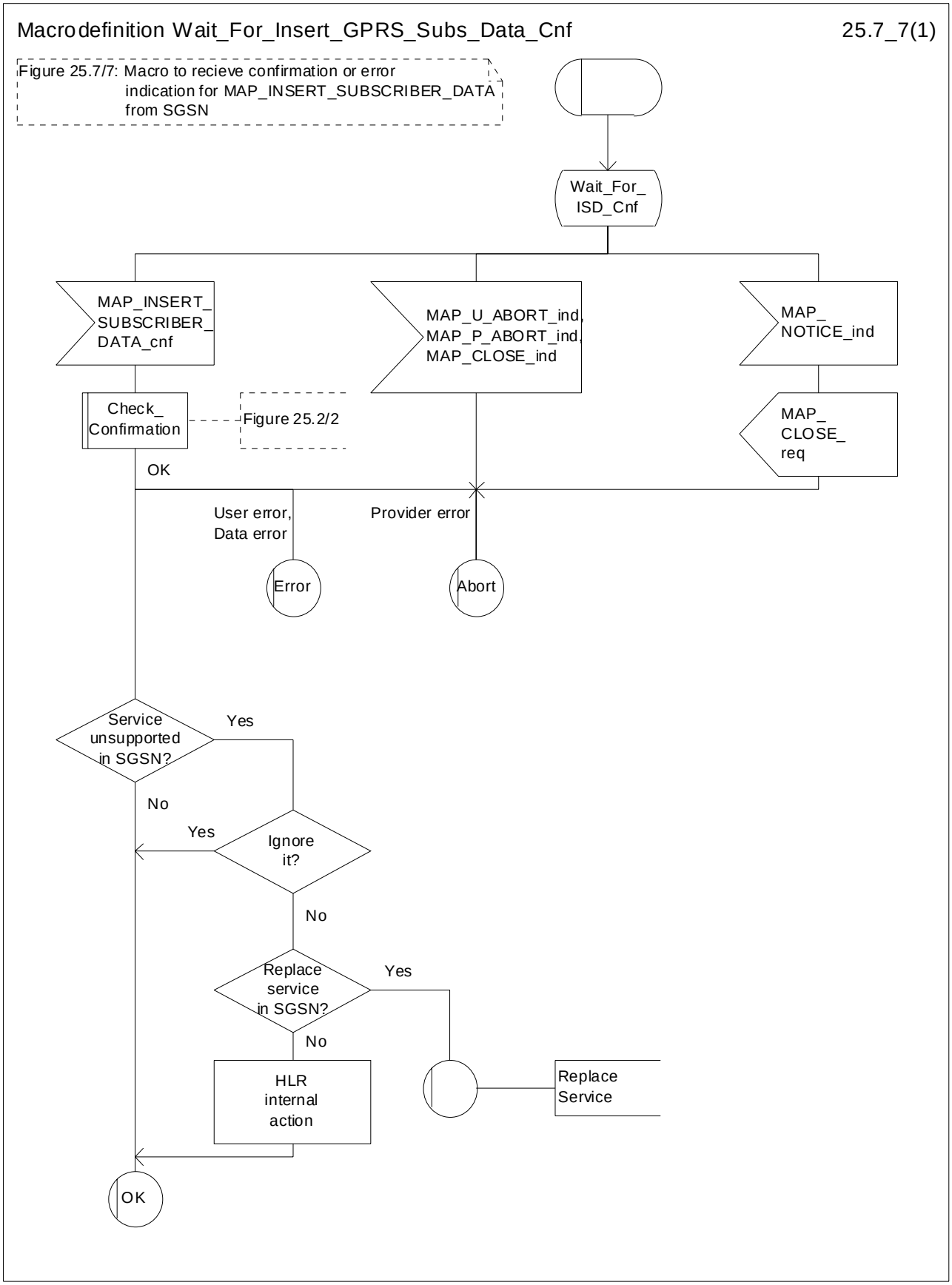


Figure 25.7/7: Macro Wait_for_Insert_GPRS_Subs_Data_Cnf

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.

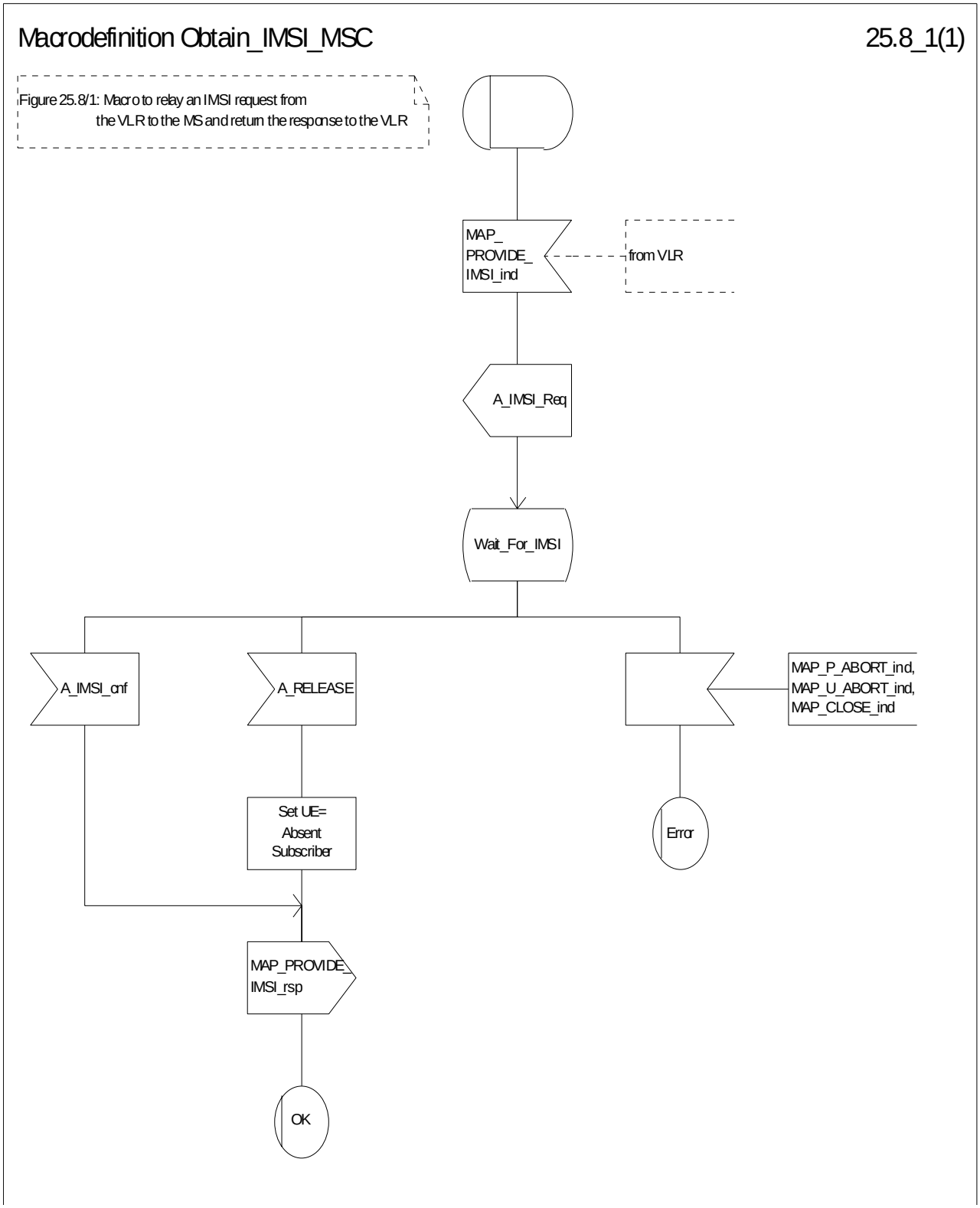


Figure 25.8/1: Macro Obtain_IMSI_MSC

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.

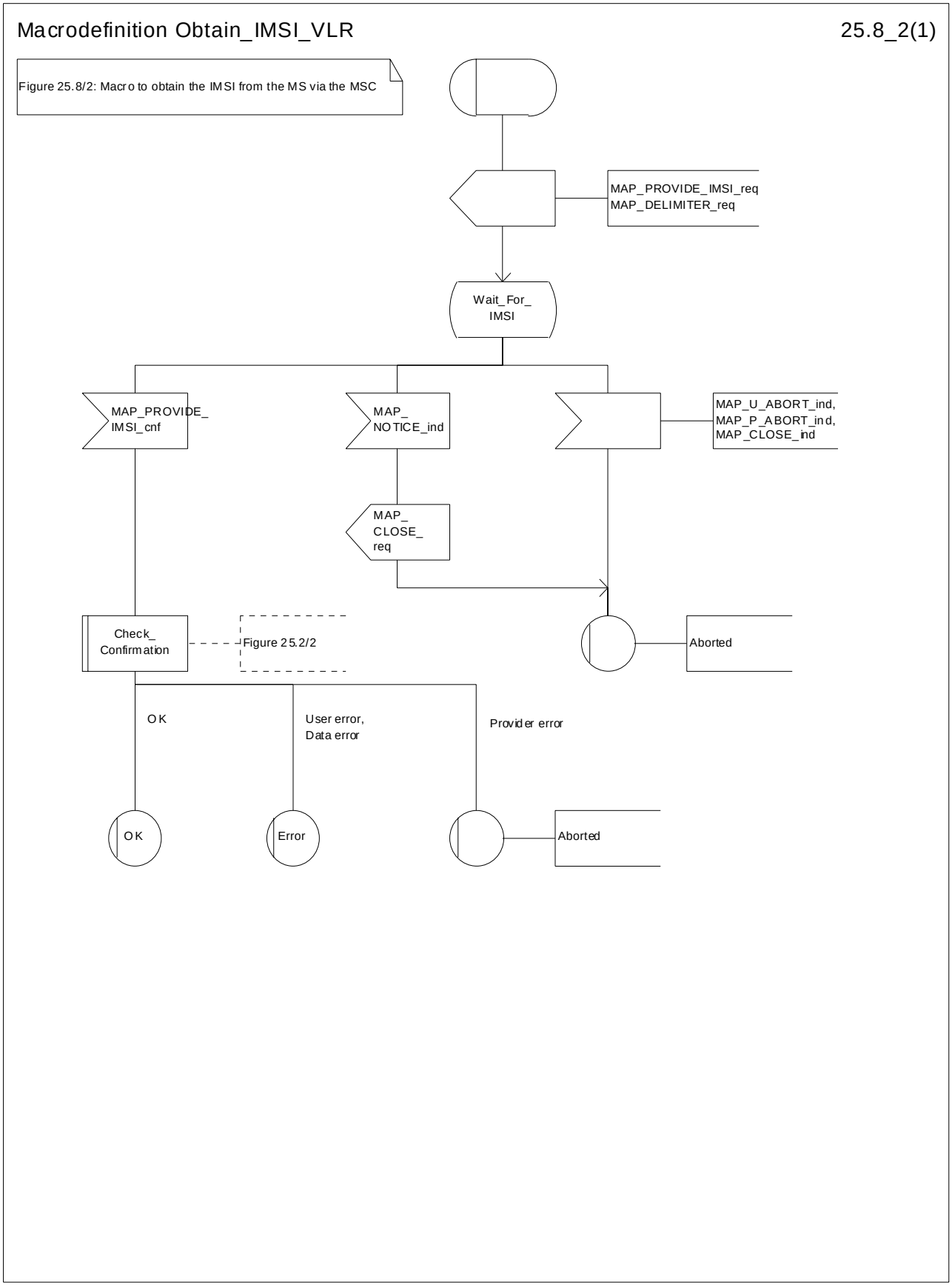


Figure 25.8/2: Macro Obtain_IMSI_VLR

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.

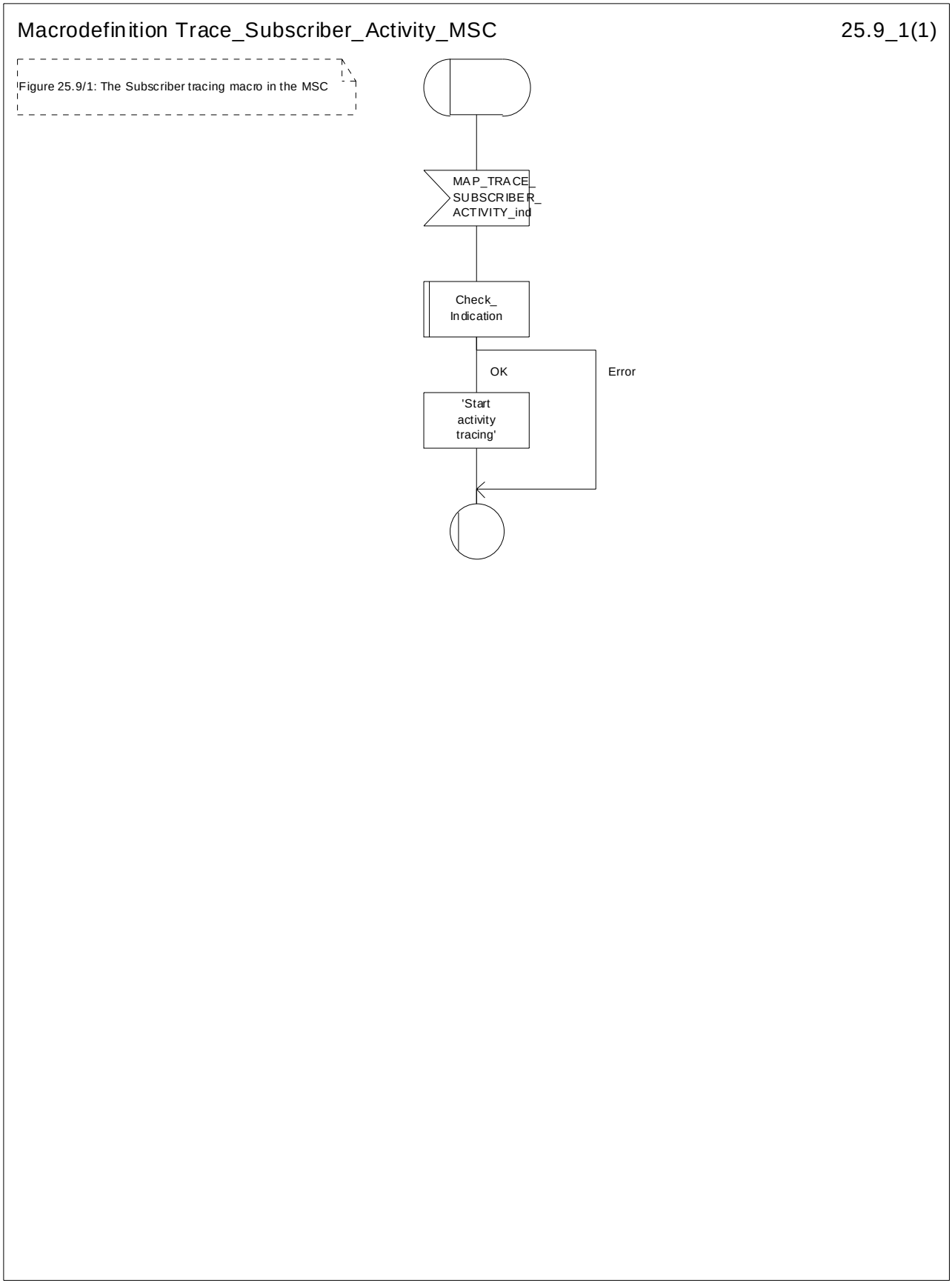


Figure 25.9/1: Macro Trace_Subscriber_Activity_MSC

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

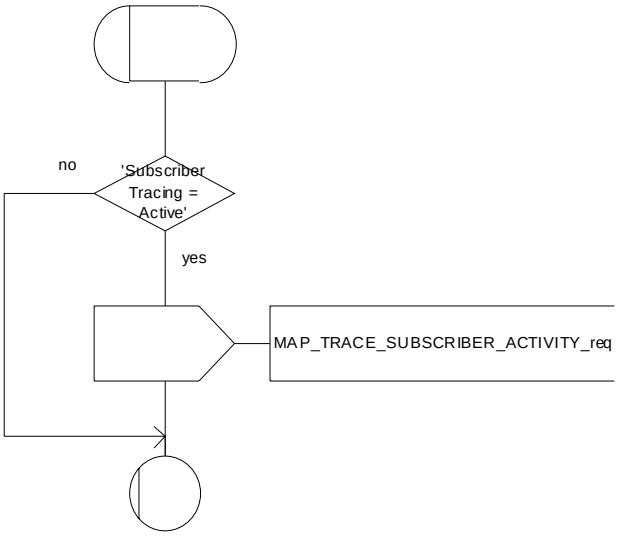


Figure 25.9/2: Macro Trace_Subscriber_Activity_VLR

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 are detected, the tracing is set active and a positive acknowledgement 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

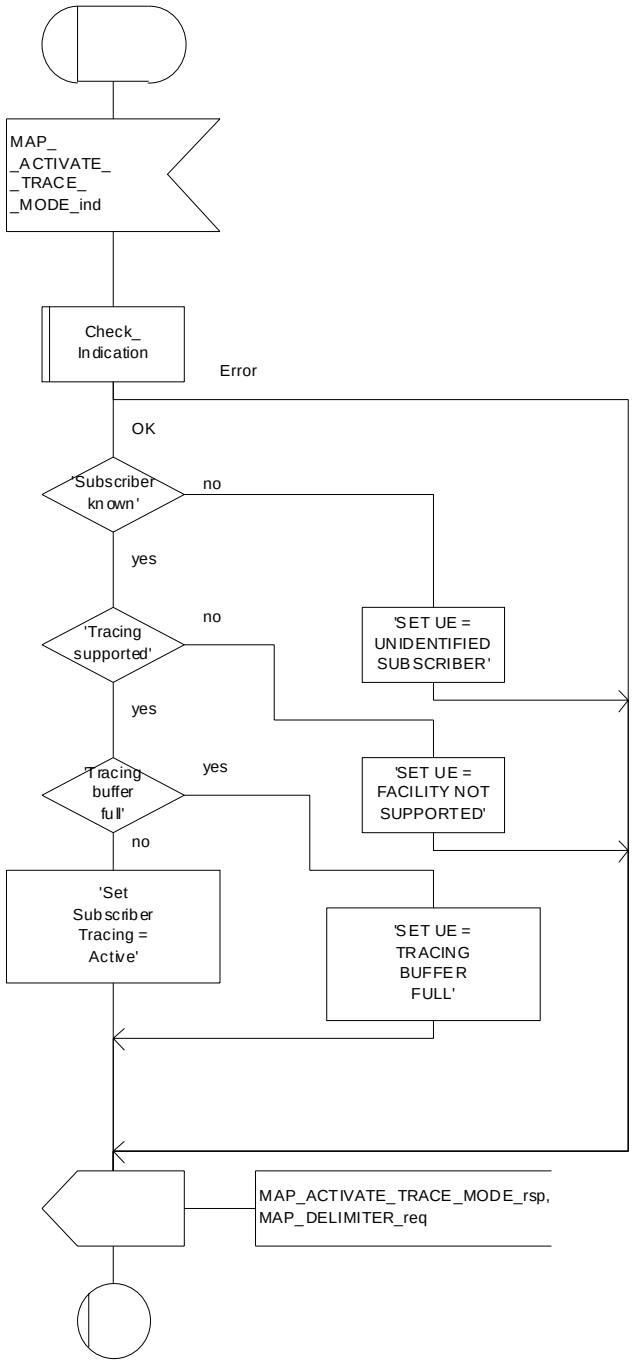


Figure 25.9/3: Macro Activate_Tracing_VLR

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 acknowledgement, 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 to 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.

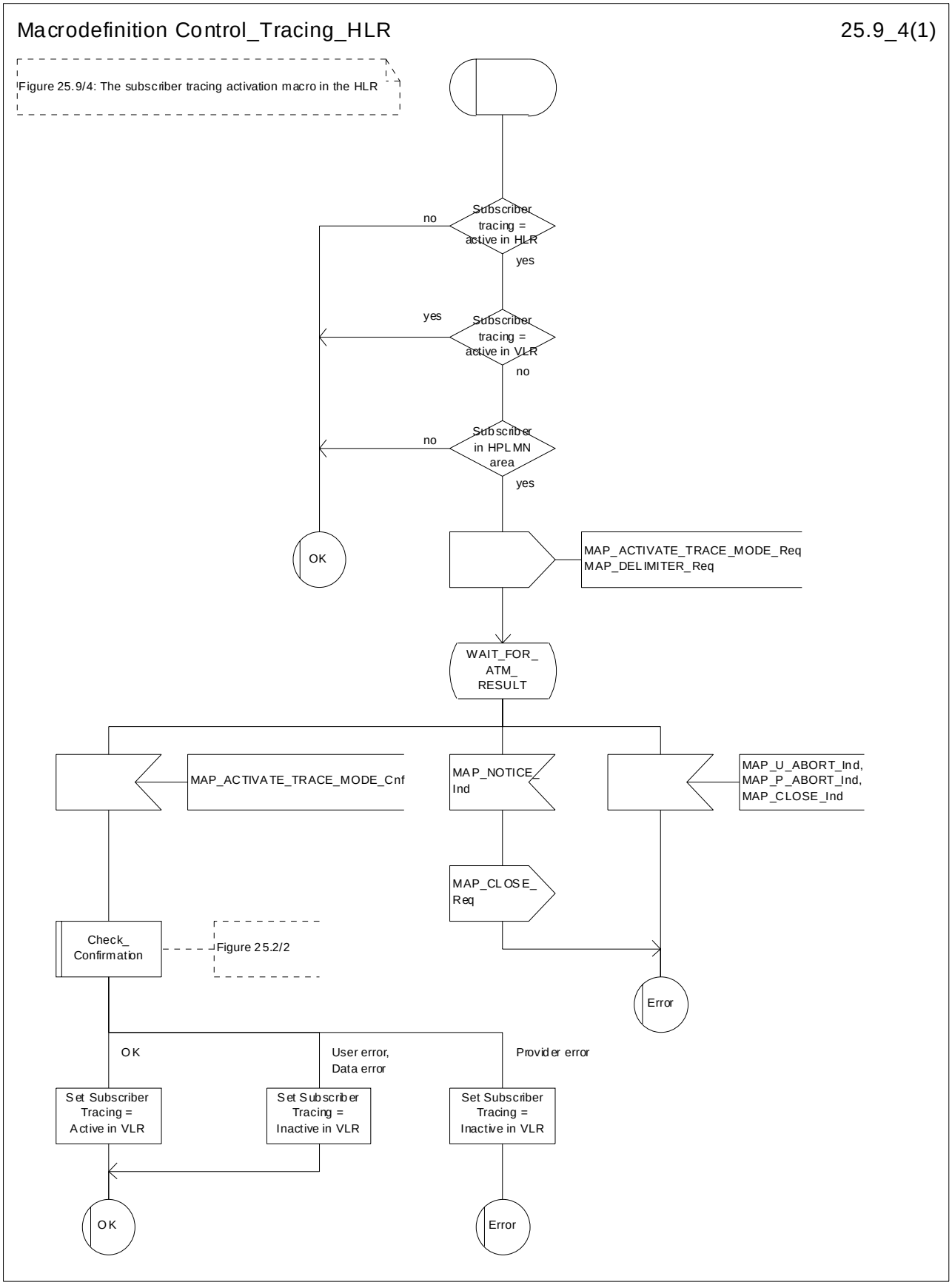


Figure 25.9/4: Macro Control_Tracing_HLR

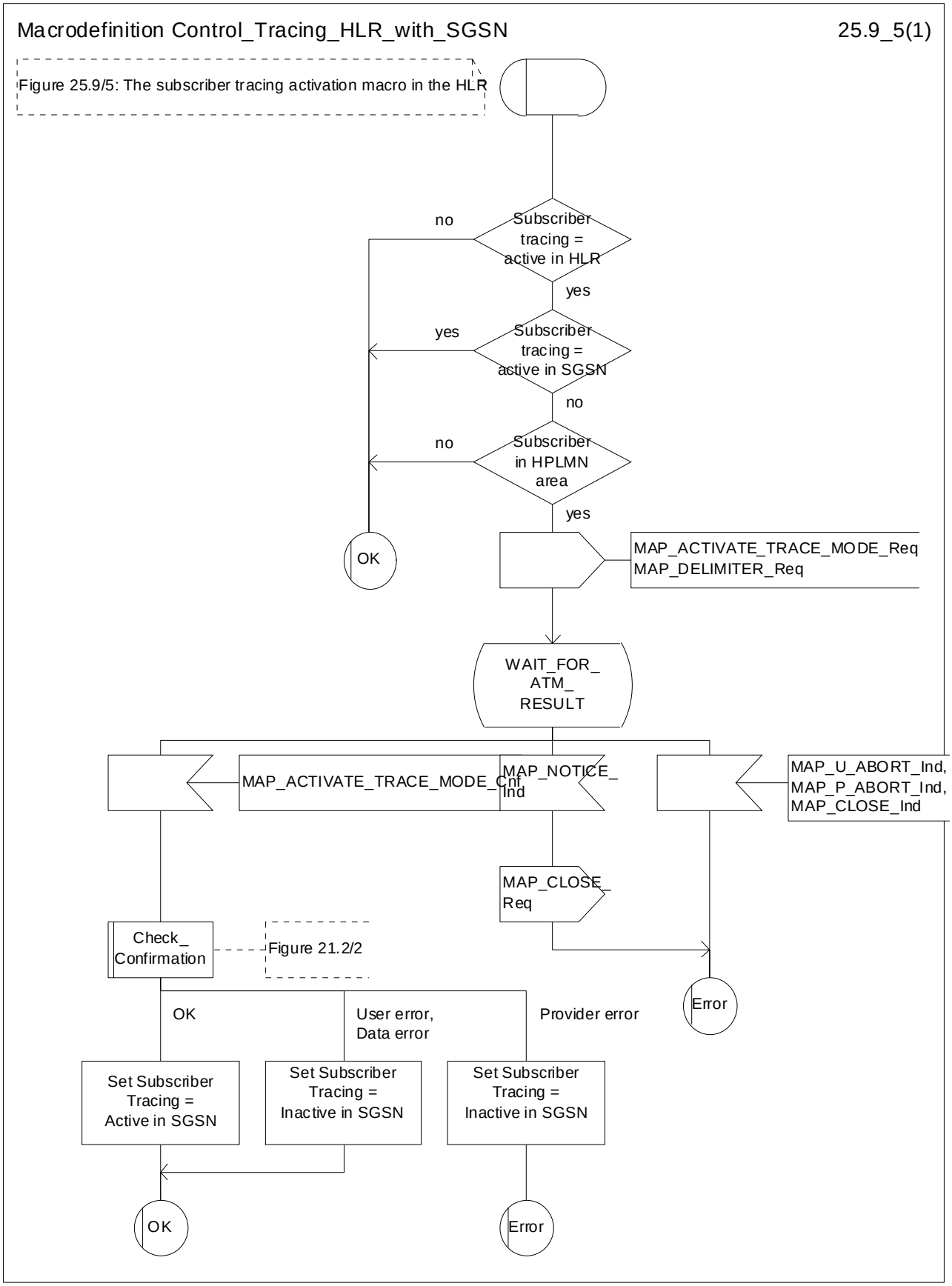


Figure 25.9/5: Macro Control_Tracing_HLR_with_SGSN

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

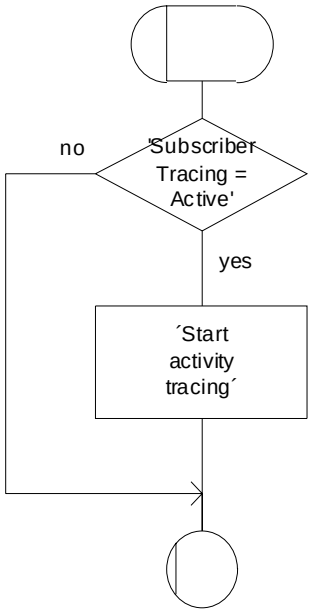


Figure 25.9/6: Macro Trace_Subscriber_Activity_SGSN

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 are detected, the tracing is set active and a positive acknowledgement 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

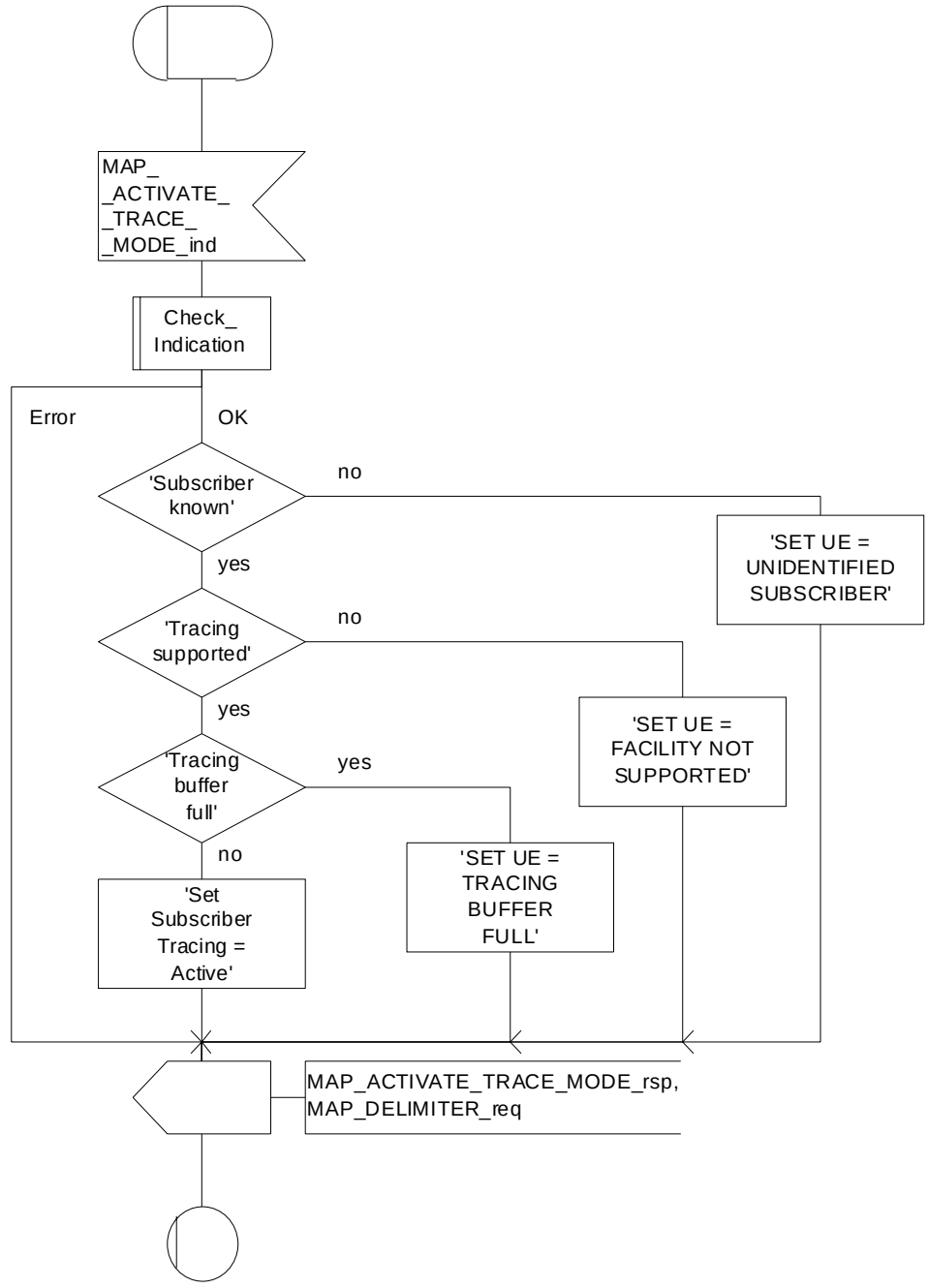


Figure 25.9/7: Macro Activate_Tracing_SGSN

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 clause 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.

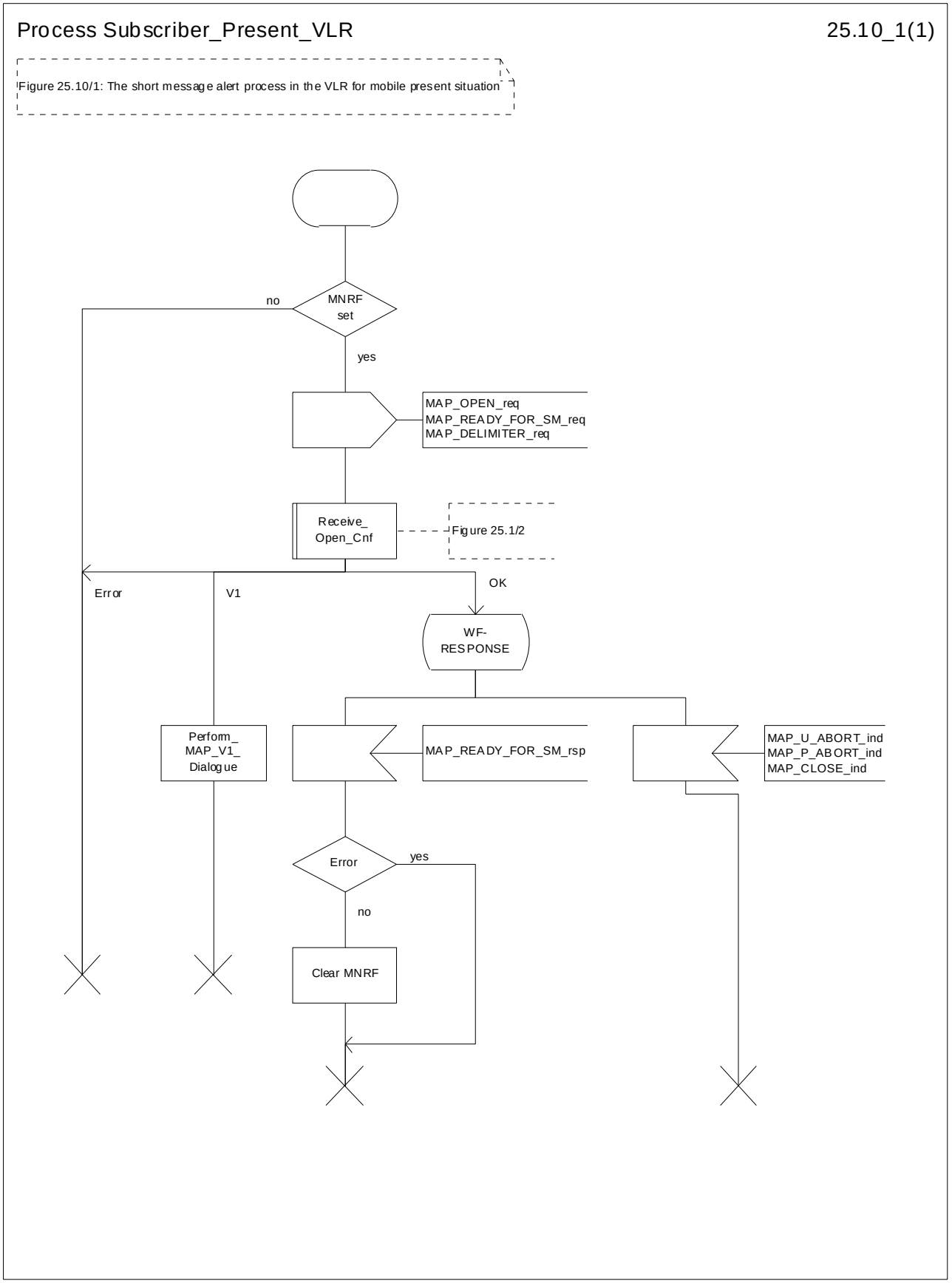


Figure 25.10/1: Process Subscriber_Present_VLR

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 acknowledgement 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

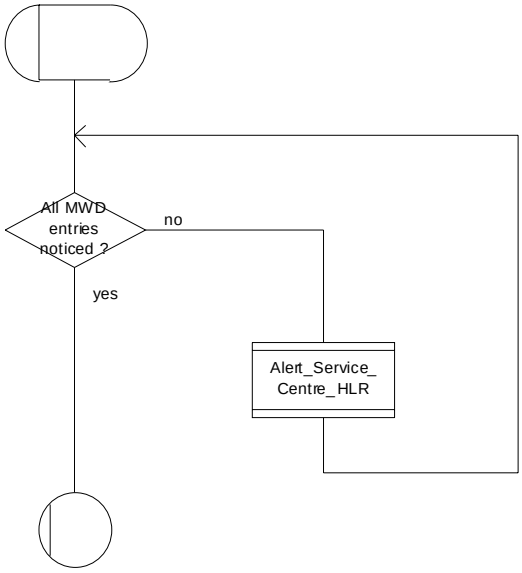


Figure 25.10/2: Macro Alert_Service_Centre_HLR

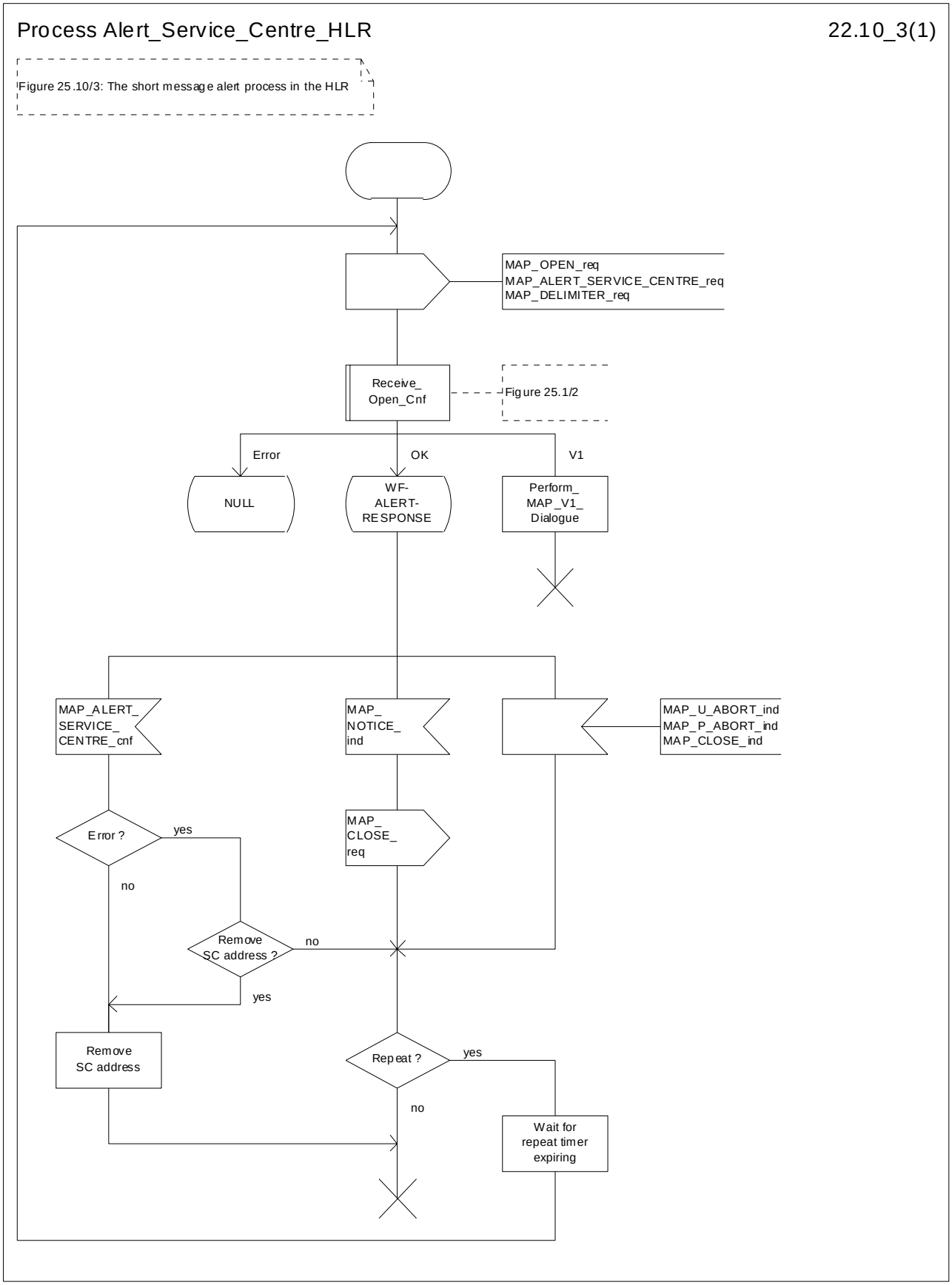


Figure 25.10/3: Process Alert_Service_Centre_HLR

25.10.3 The Mobile Subscriber is present

When receiving Page response, Attach request or Routing area update request messages (3GPP TS 24.008), 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.

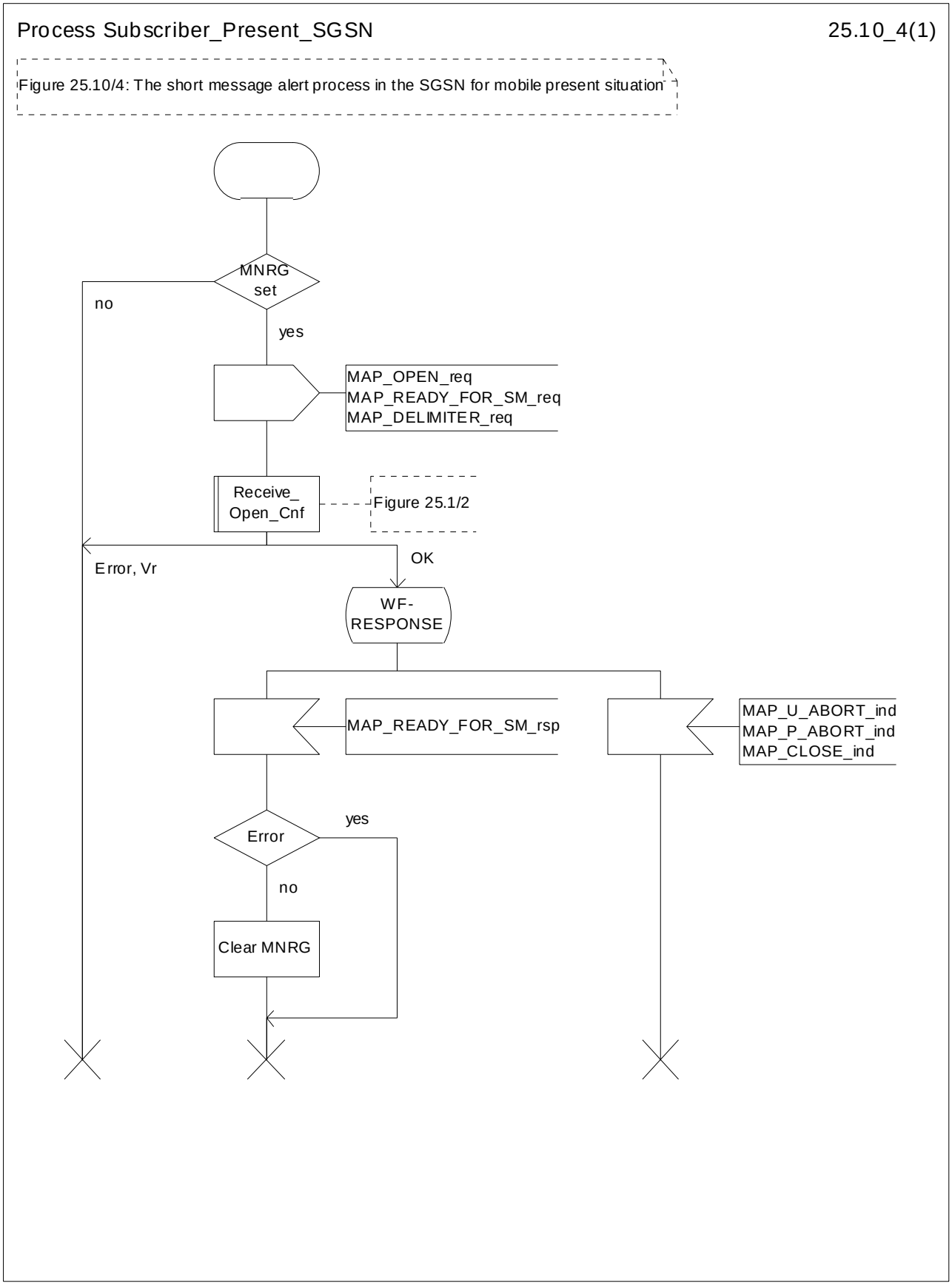


Figure 25.10/4: Process Subscriber_Present_SGSN

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 R5.70 Cross Reference Listing for MAP-Protocol 2004-06-09 15:32:55 PAGE 1

&extensionId.....identifier of Fieldspec
DEFINED in MAP-ExtensionDataTypes : 25
USED in MAP-ExtensionDataTypes : 47

&ExtensionType.....identifier of Fieldspec
DEFINED in MAP-ExtensionDataTypes : 24
USED in MAP-ExtensionDataTypes : 49

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 : 375

AbsentSubscriber.....type reference ERROR
DEFINED in MAP-Errors : 279
USED in MAP-Protocol : 135 375
USED in MAP-MobileServiceOpera : 93 441
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 : 48

absentSubscriber.....identifier of Named Number, 1
DEFINED in MAP-SM-DataTypes : 168

absentSubscriberDiagnosticSM.....identifier of [0] AbsentSubscriberDiagnosticSM
DEFINED in MAP-SM-DataTypes : 147

absentSubscriberDiagnosticSM.....identifier of AbsentSubscriberDiagnosticSM
DEFINED in MAP-ER-DataTypes : 156

AbsentSubscriberDiagnosticSM.....type reference INTEGER
DEFINED in MAP-ER-DataTypes : 166
USED in MAP-MS-DataTypes : 196 1595
USED in MAP-SM-DataTypes : 40 147 160
USED in MAP-ER-DataTypes : 43 156 161

absentSubscriberParam.....identifier of AbsentSubscriberParam
DEFINED in MAP-Errors : 281

AbsentSubscriberParam.....type reference SEQUENCE
DEFINED in MAP-ER-DataTypes : 243
USED in MAP-Errors : 125 281
USED in MAP-ER-DataTypes : 34

absentSubscriberReason.....identifier of [0] AbsentSubscriberReason
DEFINED in MAP-ER-DataTypes : 246

AbsentSubscriberReason.....type reference ENUMERATED
DEFINED in MAP-ER-DataTypes : 248
USED in MAP-ER-DataTypes : 246

absentsubscriberSM.....value reference AbsentSubscriberSM, CHOICE VALUE
DEFINED in MAP-Protocol : 421

AbsentSubscriberSM.....type reference ERROR
DEFINED in MAP-Errors : 414
USED in MAP-Protocol : 156 421
USED in MAP-ShortMessageServic : 40 79 109
USED in MAP-Errors : 80

absentSubscriberSM-Param.....identifier of AbsentSubscriberSM-Param
DEFINED in MAP-Errors : 416

AbsentSubscriberSM-Param.....type reference SEQUENCE
DEFINED in MAP-ER-DataTypes : 155
USED in MAP-Errors : 135 416
USED in MAP-ER-DataTypes : 42

accepted.....identifier of Named Number, 0
DEFINED in MAP-CH-DataTypes : 401

accessNetworkProtocolId.....identifier of AccessNetworkProtocolId
DEFINED in MAP-CommonDataTypes : 239

AccessNetworkProtocolId.....type reference ENUMERATED
DEFINED in MAP-CommonDataTypes : 255
USED in MAP-CommonDataTypes : 239

AccessNetworkSignalInfo.....type reference SEQUENCE
DEFINED in MAP-CommonDataTypes : 238
USED in MAP-MS-DataTypes : 165 408 467 521 568 574 579 593
USED in MAP-CommonDataTypes : 23

accessOutsideLSAsAllowed.....identifier of Named Number, 0
DEFINED in MAP-MS-DataTypes : 850

accessOutsideLSAsRestricted.....identifier of Named Number, 1
DEFINED in MAP-MS-DataTypes : 851

activate.....identifier of Named Number, 1
DEFINED in MAP-MS-DataTypes : 1901

activateSS.....value reference ActivateSS, CHOICE VALUE
DEFINED in MAP-Protocol : 247

ActivateSS.....type reference OPERATION
DEFINED in MAP-SupplementaryServi : 121
USED in MAP-Protocol : 71 247
USED in MAP-SupplementaryServi : 15

activateTraceMode.....value reference ActivateTraceMode, CHOICE VALUE
DEFINED in MAP-Protocol : 224

ActivateTraceMode.....type reference OPERATION
DEFINED in MAP-OperationAndMainte : 50
USED in MAP-Protocol : 46 224
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

active.....identifier of Named Number, 2
DEFINED in MAP-SS-DataTypes : 288

additionalAbsentSubscriberDiagnosticSM..identifier of [5] AbsentSubscriberDiagnosticSM
DEFINED in MAP-SM-DataTypes : 160

additionalAbsentSubscriberDiagnosticSM..identifier of [0] AbsentSubscriberDiagnosticSM
DEFINED in MAP-ER-DataTypes : 161

additionalSignalInfo.....identifier of [17] Ext-ExternalSignalInfo
DEFINED in MAP-CH-DataTypes : 113

additionalSignalInfo.....identifier of [14] Ext-ExternalSignalInfo
DEFINED in MAP-CH-DataTypes : 214

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 : 100
USED in MAP-MS-DataTypes : 161 1875
USED in MAP-CommonDataTypes : 16 144 150 376
USED in MAP-OM-DataTypes : 21 40
USED in MAP-SS-DataTypes : 44 75 301

USED in MAP-SM-DataTypes : 31 55 135 140 145 179

USED in MAP-LCS-DataTypes : 28 106

Add-GeographicalInformation.....type reference OCTET STRING
DEFINED in MAP-LCS-DataTypes : 261
USED in MAP-LCS-DataTypes : 24 190 296

add-LocationEstimate.....identifier of [2] Add-GeographicalInformation
DEFINED in MAP-LCS-DataTypes : 190

add-LocationEstimate.....identifier of [8] Add-GeographicalInformation
DEFINED in MAP-LCS-DataTypes : 296

AgeIndicator.....type reference OCTET STRING
DEFINED in MAP-MS-DataTypes : 231
USED in MAP-MS-DataTypes : 229 703

ageOfLocationEstimate.....identifier of [0] AgeOfLocationInformation
DEFINED in MAP-LCS-DataTypes : 187

ageOfLocationEstimate.....identifier of [6] AgeOfLocationInformation
DEFINED in MAP-LCS-DataTypes : 293

ageOfLocationInformation.....identifier of AgeOfLocationInformation
DEFINED in MAP-MS-DataTypes : 1706

AgeOfLocationInformation.....type reference INTEGER
DEFINED in MAP-CommonDataTypes : 495
USED in MAP-MS-DataTypes : 178 1706
USED in MAP-CommonDataTypes : 58
USED in MAP-LCS-DataTypes : 34 187 293

alertingCategory-1.....value reference AlertingPattern, '00000100'B
DEFINED in MAP-CommonDataTypes : 283

alertingCategory-2.....value reference AlertingPattern, '00000101'B
DEFINED in MAP-CommonDataTypes : 284

alertingCategory-3.....value reference AlertingPattern, '00000110'B
DEFINED in MAP-CommonDataTypes : 285

alertingCategory-4.....value reference AlertingPattern, '00000111'B
DEFINED in MAP-CommonDataTypes : 286

alertingCategory-5.....value reference AlertingPattern, '00001000'B
DEFINED in MAP-CommonDataTypes : 287

alertingLevel-0.....value reference AlertingPattern, '00000000'B
DEFINED in MAP-CommonDataTypes : 277

alertingLevel-1.....value reference AlertingPattern, '00000001'B
DEFINED in MAP-CommonDataTypes : 278

alertingLevel-2.....value reference AlertingPattern, '00000010'B
DEFINED in MAP-CommonDataTypes : 279

AlertingPattern.....type reference OCTET STRING
DEFINED in MAP-CommonDataTypes : 264
USED in MAP-CommonDataTypes : 26 277 278 279 283 284 285 286 287
USED in MAP-CH-DataTypes : 71 110 211 391
USED in MAP-SS-DataTypes : 50 225

alertingPattern.....identifier of [14] AlertingPattern
DEFINED in MAP-CH-DataTypes : 110

alertingPattern.....identifier of [12] AlertingPattern
DEFINED in MAP-CH-DataTypes : 211

alertingPattern.....identifier of [5] AlertingPattern
DEFINED in MAP-CH-DataTypes : 391

alertingPattern.....identifier of AlertingPattern
DEFINED in MAP-SS-DataTypes : 225

alertReason.....identifier of AlertReason
DEFINED in MAP-SM-DataTypes : 198

AlertReason.....type reference ENUMERATED
DEFINED in MAP-SM-DataTypes : 210
USED in MAP-SM-DataTypes : 27 198

alertReasonIndicator.....identifier of NULL

DEFINED in MAP-SM-DataTypes : 199

alertServiceCentre.....value reference AlertServiceCentre, CHOICE VALUE
DEFINED in MAP-Protocol : 267

AlertServiceCentre.....type reference OPERATION
DEFINED in MAP-ShortMessageServic : 123
USED in MAP-Protocol : 91 267
USED in MAP-ShortMessageServic : 17

alertServiceCentreArg.....identifier of AlertServiceCentreArg
DEFINED in MAP-ShortMessageServic : 125

AlertServiceCentreArg.....type reference SEQUENCE
DEFINED in MAP-SM-DataTypes : 177
USED in MAP-ShortMessageServic : 53 125
USED in MAP-SM-DataTypes : 22

allAdditionalInfoTransferSS.....value reference SS-Code, '10000000'B
DEFINED in MAP-SS-Code : 107

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

allAsynchronousServices.....value reference BearerServiceCode, '01100000'B
DEFINED in MAP-BS-Code : 95

allBarringSS.....value reference SS-Code, '10010000'B
DEFINED in MAP-SS-Code : 117

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 : 153

allChargingSS.....value reference SS-Code, '01110000'B
DEFINED in MAP-SS-Code : 99

allCommunityOfInterest-SS.....value reference SS-Code, '01100000'B
DEFINED in MAP-SS-Code : 93

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 : 934

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 : 1243

allIC-CallsBarred.....identifier of Named Number, 19
DEFINED in MAP-MS-DataTypes : 941

allInformationSent.....identifier of [4] NULL
DEFINED in MAP-MS-DataTypes : 1912

allInformationSent.....identifier of [11] NULL
DEFINED in MAP-CH-DataTypes : 235

allLCSPrivacyException.....value reference SS-Code, '10110000'B
DEFINED in MAP-SS-Code : 159

allLineIdentificationSS.....value reference SS-Code, '00010000'B
DEFINED in MAP-SS-Code : 25

allLSAData.....identifier of NULL
DEFINED in MAP-MS-DataTypes : 1250

allMOLR-SS.....value reference SS-Code, '11000000'B
DEFINED in MAP-SS-Code : 171

allMultiPartySS.....value reference SS-Code, '01010000'B
DEFINED in MAP-SS-Code : 87

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 : 925

allowedGSM-Algorithms.....identifier of [4] AllowedGSM-Algorithms
DEFINED in MAP-MS-DataTypes : 412

AllowedGSM-Algorithms.....type reference OCTET STRING
DEFINED in MAP-MS-DataTypes : 422
USED in MAP-MS-DataTypes : 412 473

allowedGSM-Algorithms.....identifier of [9] AllowedGSM-Algorithms
DEFINED in MAP-MS-DataTypes : 473

allowedUMTS-Algorithms.....identifier of [5] AllowedUMTS-Algorithms
DEFINED in MAP-MS-DataTypes : 413

AllowedUMTS-Algorithms.....type reference SEQUENCE
DEFINED in MAP-MS-DataTypes : 427
USED in MAP-MS-DataTypes : 413 474

allowedUMTS-Algorithms.....identifier of [10] AllowedUMTS-Algorithms
DEFINED in MAP-MS-DataTypes : 474

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 : 136

allPLMN-specificTS.....value reference TeleserviceCode, '11010000'B
DEFINED in MAP-TS-Code : 71

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 : 384

anyTimeInterrogation.....value reference AnyTimeInterrogation, CHOICE VALUE
DEFINED in MAP-Protocol : 276

AnyTimeInterrogation.....type reference OPERATION
DEFINED in MAP-MobileServiceOpera : 244
USED in MAP-Protocol : 31 276
USED in MAP-MobileServiceOpera : 27

anyTimeInterrogationArg.....identifier of AnyTimeInterrogationArg
DEFINED in MAP-MobileServiceOpera : 246

AnyTimeInterrogationArg.....type reference SEQUENCE
DEFINED in MAP-MS-DataTypes : 1762
USED in MAP-MobileServiceOpera : 151 246
USED in MAP-MS-DataTypes : 102

anyTimeInterrogationRes.....identifier of AnyTimeInterrogationRes
DEFINED in MAP-MobileServiceOpera : 248

AnyTimeInterrogationRes.....type reference SEQUENCE
DEFINED in MAP-MS-DataTypes : 1769
USED in MAP-MobileServiceOpera : 152 248
USED in MAP-MS-DataTypes : 103

anyTimeModification.....value reference AnyTimeModification, CHOICE VALUE
DEFINED in MAP-Protocol : 281

AnyTimeModification.....type reference OPERATION
DEFINED in MAP-MobileServiceOpera : 275
USED in MAP-Protocol : 33 281
USED in MAP-MobileServiceOpera : 31

anyTimeModificationArg.....identifier of AnyTimeModificationArg
DEFINED in MAP-MobileServiceOpera : 277

AnyTimeModificationArg.....type reference SEQUENCE
DEFINED in MAP-MS-DataTypes : 1855
USED in MAP-MobileServiceOpera : 147 277
USED in MAP-MS-DataTypes : 108

anyTimeModificationRes.....identifier of AnyTimeModificationRes
DEFINED in MAP-MobileServiceOpera : 279

AnyTimeModificationRes.....type reference SEQUENCE
DEFINED in MAP-MS-DataTypes : 1865
USED in MAP-MobileServiceOpera : 148 279
USED in MAP-MS-DataTypes : 109

anyTimeSubscriptionInterrogation.value reference AnyTimeSubscriptionInterrogation, CHOICE VALUE
DEFINED in MAP-Protocol : 280

AnyTimeSubscriptionInterrogation.....type reference OPERATION
DEFINED in MAP-MobileServiceOpera : 258
USED in MAP-Protocol : 32 280
USED in MAP-MobileServiceOpera : 30

anyTimeSubscriptionInterrogationArg.....identifier of AnyTimeSubscriptionInterrogationArg
DEFINED in MAP-MobileServiceOpera : 260

AnyTimeSubscriptionInterrogationArg.....type reference SEQUENCE
DEFINED in MAP-MS-DataTypes : 1777
USED in MAP-MobileServiceOpera : 145 260
USED in MAP-MS-DataTypes : 106

anyTimeSubscriptionInterrogationRes.....identifier of AnyTimeSubscriptionInterrogationRes
DEFINED in MAP-MobileServiceOpera : 262

AnyTimeSubscriptionInterrogationRes.....type reference SEQUENCE
DEFINED in MAP-MS-DataTypes : 1785
USED in MAP-MobileServiceOpera : 146 262

USED in MAP-MS-DataTypes : 107

an-APDU.....identifier of AccessNetworkSignalInfo
DEFINED in MAP-MS-DataTypes : 408

an-APDU.....identifier of [2] AccessNetworkSignalInfo
DEFINED in MAP-MS-DataTypes : 467

an-APDU.....identifier of [2] AccessNetworkSignalInfo
DEFINED in MAP-MS-DataTypes : 521

an-APDU.....identifier of [3] AccessNetworkSignalInfo
DEFINED in MAP-MS-DataTypes : 568

an-APDU.....identifier of AccessNetworkSignalInfo
DEFINED in MAP-MS-DataTypes : 574

an-APDU.....identifier of AccessNetworkSignalInfo
DEFINED in MAP-MS-DataTypes : 579

an-APDU.....identifier of AccessNetworkSignalInfo
DEFINED in MAP-MS-DataTypes : 593

aocc.....value reference SS-Code, '01110010'B
DEFINED in MAP-SS-Code : 104

aoci.....value reference SS-Code, '01110001'B
DEFINED in MAP-SS-Code : 102

apn.....identifier of [20] APN
DEFINED in MAP-MS-DataTypes : 750

APN.....type reference OCTET STRING
DEFINED in MAP-MS-DataTypes : 822
USED in MAP-MS-DataTypes : 750

asciCallReference.....identifier of [20] ASCI-CallReference
DEFINED in MAP-MS-DataTypes : 482

asciCallReference.....identifier of ASCI-CallReference
DEFINED in MAP-GR-DataTypes : 51

ASCI-CallReference.....type reference TBCD-STRING
DEFINED in MAP-CommonDataTypes : 307
USED in MAP-MS-DataTypes : 182 482
USED in MAP-CommonDataTypes : 41
USED in MAP-GR-DataTypes : 26 51

assumedIdle.....identifier of [0] NULL
DEFINED in MAP-MS-DataTypes : 1749

ati-NotAllowed.....value reference ATI-NotAllowed, CHOICE VALUE
DEFINED in MAP-Protocol : 387

ATI-NotAllowed.....type reference ERROR
DEFINED in MAP-Errors : 323
USED in MAP-Protocol : 142 387
USED in MAP-MobileServiceOpera : 90 251
USED in MAP-Errors : 55

ati-NotAllowedParam.....identifier of ATI-NotAllowedParam
DEFINED in MAP-Errors : 325

ATI-NotAllowedParam.....type reference SEQUENCE
DEFINED in MAP-ER-DataTypes : 279
USED in MAP-Errors : 132 325
USED in MAP-ER-DataTypes : 39

atm-NotAllowed.....value reference ATM-NotAllowed, CHOICE VALUE
DEFINED in MAP-Protocol : 391

ATM-NotAllowed.....type reference ERROR
DEFINED in MAP-Errors : 334
USED in MAP-Protocol : 167 391
USED in MAP-MobileServiceOpera : 96 281
USED in MAP-Errors : 59

atm-NotAllowedParam.....identifier of ATM-NotAllowedParam
DEFINED in MAP-Errors : 336

ATM-NotAllowedParam.....type reference SEQUENCE
DEFINED in MAP-ER-DataTypes : 287

USED in MAP-Errors : 147 336

USED in MAP-ER-DataTypes : 55

atsi-NotAllowed.....value reference ATSI-NotAllowed, CHOICE VALUE
DEFINED in MAP-Protocol : 390

ATSI-NotAllowed.....type reference ERROR
DEFINED in MAP-Errors : 329
USED in MAP-Protocol : 166 390
USED in MAP-MobileServiceOpera : 95 264
USED in MAP-Errors : 58

atsi-NotAllowedParam.....identifier of ATSI-NotAllowedParam
DEFINED in MAP-Errors : 331

ATSI-NotAllowedParam.....type reference SEQUENCE
DEFINED in MAP-ER-DataTypes : 283
USED in MAP-Errors : 146 331
USED in MAP-ER-DataTypes : 54

attach.....identifier of Named Number, 1
DEFINED in MAP-MS-DataTypes : 811

attachChangeOfPosition.....identifier of Named Number, 2
DEFINED in MAP-MS-DataTypes : 812

authenticationFailureReport.....value reference AuthenticationFailureReport, CHOICE VALUE
DEFINED in MAP-Protocol : 200

AuthenticationFailureReport.....type reference OPERATION
DEFINED in MAP-MobileServiceOpera : 367
USED in MAP-Protocol : 23 200
USED in MAP-MobileServiceOpera : 46

authenticationFailureReportArg.....identifier of AuthenticationFailureReportArg
DEFINED in MAP-MobileServiceOpera : 369

AuthenticationFailureReportArg.....type reference SEQUENCE
DEFINED in MAP-MS-DataTypes : 364
USED in MAP-MobileServiceOpera : 133 369
USED in MAP-MS-DataTypes : 45

authenticationFailureReportRes.....identifier of AuthenticationFailureReportRes
DEFINED in MAP-MobileServiceOpera : 371

AuthenticationFailureReportRes.....type reference SEQUENCE
DEFINED in MAP-MS-DataTypes : 370
USED in MAP-MobileServiceOpera : 134 371
USED in MAP-MS-DataTypes : 46

AuthenticationQuintuplet.....type reference SEQUENCE
DEFINED in MAP-MS-DataTypes : 319
USED in MAP-MS-DataTypes : 311

authenticationSetList.....identifier of AuthenticationSetList
DEFINED in MAP-MS-DataTypes : 296

AuthenticationSetList.....type reference CHOICE
DEFINED in MAP-MS-DataTypes : 303
USED in MAP-MS-DataTypes : 296 673

authenticationSetList.....identifier of AuthenticationSetList
DEFINED in MAP-MS-DataTypes : 673

AuthenticationTriplet.....type reference SEQUENCE
DEFINED in MAP-MS-DataTypes : 313
USED in MAP-MS-DataTypes : 308

autn.....identifier of AUTN
DEFINED in MAP-MS-DataTypes : 324

AUTN.....type reference OCTET STRING
DEFINED in MAP-MS-DataTypes : 354
USED in MAP-MS-DataTypes : 324

automaticFacsimileGroup3.....value reference TeleserviceCode, '01100010'B
DEFINED in MAP-TS-Code : 50

autonomousSelfLocation.....value reference SS-Code, '11000010'B
DEFINED in MAP-SS-Code : 175

AUTS.....type reference OCTET STRING

DEFINED in MAP-MS-DataTypes : 356

USED in MAP-MS-DataTypes : 669

auts.....identifier of AUTS
DEFINED in MAP-MS-DataTypes : 669

a-side.....identifier of Named Number, 0
DEFINED in MAP-CH-DataTypes : 364

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 : 130

baoc.....value reference SS-Code, '10010010'B
DEFINED in MAP-SS-Code : 121

barringOfIncomingCalls.....value reference SS-Code, '10011001'B
DEFINED in MAP-SS-Code : 128

barringOfOutgoingCalls.....value reference SS-Code, '10010001'B
DEFINED in MAP-SS-Code : 119

barringServiceActive.....identifier of Named Number, 0
DEFINED in MAP-ER-DataTypes : 108

basicCall.....identifier of Named Number, 0
DEFINED in MAP-CH-DataTypes : 122

basicISTSupported.....identifier of Named Number, 0
DEFINED in MAP-MS-DataTypes : 236

basicSelfLocation.....value reference SS-Code, '11000001'B
DEFINED in MAP-SS-Code : 173

basicService.....identifier of Ext-BasicServiceCode
DEFINED in MAP-MS-DataTypes : 989

basicService.....identifier of Ext-BasicServiceCode
DEFINED in MAP-MS-DataTypes : 1048

basicService.....identifier of Ext-BasicServiceCode
DEFINED in MAP-MS-DataTypes : 1091

basicService.....identifier of [1] Ext-BasicServiceCode
DEFINED in MAP-MS-DataTypes : 1873

basicService.....identifier of [1] Ext-BasicServiceCode
DEFINED in MAP-MS-DataTypes : 1884

basicService.....identifier of [5] Ext-BasicServiceCode
DEFINED in MAP-CH-DataTypes : 155

basicService.....identifier of BasicServiceCode
DEFINED in MAP-SS-DataTypes : 74

basicService.....identifier of BasicServiceCode
DEFINED in MAP-SS-DataTypes : 100

basicService.....identifier of BasicServiceCode
DEFINED in MAP-SS-DataTypes : 157

basicService.....identifier of BasicServiceCode
DEFINED in MAP-SS-DataTypes : 186

basicService.....identifier of BasicServiceCode
DEFINED in MAP-ER-DataTypes : 130

BasicServiceCode.....type reference CHOICE
DEFINED in MAP-CommonDataTypes : 427
USED in MAP-CommonDataTypes : 48
USED in MAP-SS-DataTypes : 49 74 100 157 186 210 265
USED in MAP-ER-DataTypes : 74 130

basicServiceCriteria.....identifier of [1] BasicServiceCriteria
DEFINED in MAP-MS-DataTypes : 1388

basicServiceCriteria.....identifier of [0] BasicServiceCriteria

DEFINED in MAP-MS-DataTypes : 1396

BasicServiceCriteria.....type reference SEQUENCE OF
DEFINED in MAP-MS-DataTypes : 1416
USED in MAP-MS-DataTypes : 72 1388 1396

basicServiceGroup.....identifier of [9] Ext-BasicServiceCode
DEFINED in MAP-CH-DataTypes : 104

basicServiceGroup.....identifier of [1] Ext-BasicServiceCode
DEFINED in MAP-CH-DataTypes : 226

basicServiceGroup.....identifier of [3] BasicServiceCode
DEFINED in MAP-SS-DataTypes : 210

basicServiceGroupList.....identifier of Ext-BasicServiceGroupList
DEFINED in MAP-MS-DataTypes : 1066

basicServiceGroupList.....identifier of Ext-BasicServiceGroupList
DEFINED in MAP-MS-DataTypes : 1113

basicServiceGroupList.....identifier of BasicServiceGroupList
DEFINED in MAP-SS-DataTypes : 165

basicServiceGroupList.....identifier of [2] BasicServiceGroupList
DEFINED in MAP-SS-DataTypes : 217

BasicServiceGroupList.....type reference SEQUENCE OF
DEFINED in MAP-SS-DataTypes : 264
USED in MAP-SS-DataTypes : 165 217

basicServiceList.....identifier of [1] BasicServiceList
DEFINED in MAP-MS-DataTypes : 1210

BasicServiceList.....type reference SEQUENCE OF
DEFINED in MAP-MS-DataTypes : 1256
USED in MAP-MS-DataTypes : 1210

bearerService.....identifier of [2] BearerServiceCode
DEFINED in MAP-CommonDataTypes : 428

BearerServiceCode.....type reference OCTET STRING
DEFINED in MAP-BS-Code : 11
USED in MAP-CommonDataTypes : 70 428
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 : 886

BearerServiceList.....type reference SEQUENCE OF
DEFINED in MAP-MS-DataTypes : 908
USED in MAP-MS-DataTypes : 886 1194

bearerServiceList.....identifier of [2] BearerServiceList
DEFINED in MAP-MS-DataTypes : 1194

bearerServiceNotProvisioned.....value reference BearerServiceNotProvisioned, CHOICE VALUE
DEFINED in MAP-Protocol : 350

BearerServiceNotProvisioned.....type reference ERROR
DEFINED in MAP-Errors : 239
USED in MAP-Protocol : 128 350
USED in MAP-MobileServiceOpera : 97 268 285
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 : 241

BearerServNotProvParam.....type reference SEQUENCE
DEFINED in MAP-ER-DataTypes : 227
USED in MAP-Errors : 120 241
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 : 132

blackListed.....identifier of Named Number, 1
DEFINED in MAP-MS-DataTypes : 682

boic.....value reference SS-Code, '10010011'B
DEFINED in MAP-SS-Code : 123

boicExHC.....value reference SS-Code, '10010100'B
DEFINED in MAP-SS-Code : 125

bothMSCAndSGSN.....identifier of Named Number, 0
DEFINED in MAP-MS-DataTypes : 732

broadcastInitEntitlement.....identifier of NULL
DEFINED in MAP-MS-DataTypes : 1666

broadcastService.....identifier of Named Number, 0
DEFINED in MAP-CommonDataTypes : 381

bssmap-ServiceHandover.....identifier of [9] BSSMAP-ServiceHandover
DEFINED in MAP-MS-DataTypes : 418

bssmap-ServiceHandover.....identifier of [13] BSSMAP-ServiceHandover
DEFINED in MAP-MS-DataTypes : 479

bssmap-ServiceHandover.....identifier of BSSMAP-ServiceHandover
DEFINED in MAP-MS-DataTypes : 489

BSSMAP-ServiceHandover.....type reference OCTET STRING
DEFINED in MAP-MS-DataTypes : 496
USED in MAP-MS-DataTypes : 418 479 489

BSSMAP-ServiceHandoverInfo.....type reference SEQUENCE
DEFINED in MAP-MS-DataTypes : 488
USED in MAP-MS-DataTypes : 486

bssmap-ServiceHandoverList.....identifier of [10] BSSMAP-ServiceHandoverList
DEFINED in MAP-MS-DataTypes : 420

bssmap-ServiceHandoverList.....identifier of [15] BSSMAP-ServiceHandoverList
DEFINED in MAP-MS-DataTypes : 481

BSSMAP-ServiceHandoverList.....type reference SEQUENCE OF
DEFINED in MAP-MS-DataTypes : 485
USED in MAP-MS-DataTypes : 420 481

busy.....identifier of Named Number, 1
DEFINED in MAP-CH-DataTypes : 131

busy.....identifier of Named Number, 2
DEFINED in MAP-CH-DataTypes : 374

busySubscriber.....value reference BusySubscriber, CHOICE VALUE
DEFINED in MAP-Protocol : 376

BusySubscriber.....type reference ERROR
DEFINED in MAP-Errors : 286
USED in MAP-Protocol : 136 376
USED in MAP-CallHandlingOperat : 41 97 190
USED in MAP-Errors : 46

busySubscriberParam.....identifier of BusySubscriberParam
DEFINED in MAP-Errors : 288

BusySubscriberParam.....type reference SEQUENCE
DEFINED in MAP-ER-DataTypes : 261
USED in MAP-Errors : 126 288
USED in MAP-ER-DataTypes : 35

b-side.....identifier of Named Number, 1
DEFINED in MAP-CH-DataTypes : 365

b-subscriberNumber.....identifier of [1] ISDN-AddressString
DEFINED in MAP-SS-DataTypes : 208

b-subscriberNumber.....identifier of [5] ISDN-AddressString

DEFINED in MAP-SS-DataTypes : 281

b-subscriberSubaddress.....identifier of [2] ISDN-SubaddressString
DEFINED in MAP-SS-DataTypes : 209

b-Subscriber-Address.....identifier of [3] ISDN-AddressString
DEFINED in MAP-CH-DataTypes : 285

callBarred.....value reference CallBarred, CHOICE VALUE
DEFINED in MAP-Protocol : 378

CallBarred.....type reference ERROR
DEFINED in MAP-Errors : 296
USED in MAP-Protocol : 138 378
USED in MAP-MobileServiceOpera : 99 270 287
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 : 49

callBarredParam.....identifier of CallBarredParam
DEFINED in MAP-Errors : 298

CallBarredParam.....type reference CHOICE
DEFINED in MAP-ER-DataTypes : 100
USED in MAP-Errors : 128 298
USED in MAP-ER-DataTypes : 15

callBarringCause.....identifier of CallBarringCause
DEFINED in MAP-ER-DataTypes : 101

CallBarringCause.....type reference ENUMERATED
DEFINED in MAP-ER-DataTypes : 107
USED in MAP-ER-DataTypes : 101 112

callBarringCause.....identifier of CallBarringCause
DEFINED in MAP-ER-DataTypes : 112

callBarringData.....identifier of [2] CallBarringData
DEFINED in MAP-MS-DataTypes : 1787

CallBarringData.....type reference SEQUENCE
DEFINED in MAP-MS-DataTypes : 1821
USED in MAP-MS-DataTypes : 1787

CallBarringFeature.....type reference SEQUENCE
DEFINED in MAP-SS-DataTypes : 156
USED in MAP-SS-DataTypes : 154

callBarringFeatureList.....identifier of Ext-CallBarFeatureList
DEFINED in MAP-MS-DataTypes : 1040

callBarringFeatureList.....identifier of Ext-CallBarFeatureList
DEFINED in MAP-MS-DataTypes : 1822

callBarringFeatureList.....identifier of [1] Ext-CallBarFeatureList
DEFINED in MAP-MS-DataTypes : 1951

callBarringFeatureList.....identifier of CallBarringFeatureList
DEFINED in MAP-SS-DataTypes : 150

CallBarringFeatureList.....type reference SEQUENCE OF
DEFINED in MAP-SS-DataTypes : 153
USED in MAP-SS-DataTypes : 150

callBarringInfo.....identifier of [1] Ext-CallBarInfo
DEFINED in MAP-MS-DataTypes : 973

callBarringInfo.....identifier of [1] CallBarringInfo
DEFINED in MAP-SS-DataTypes : 87

CallBarringInfo.....type reference SEQUENCE
DEFINED in MAP-SS-DataTypes : 148
USED in MAP-SS-DataTypes : 87

callBarringInfoFor-CSE.....identifier of [1] Ext-CallBarringInfoFor-CSE
DEFINED in MAP-MS-DataTypes : 1909

callBarringInfoFor-CSE.....identifier of [1] Ext-CallBarringInfoFor-CSE
DEFINED in MAP-MS-DataTypes : 1939

CallDirection.....type reference OCTET STRING
DEFINED in MAP-CH-DataTypes : 292
USED in MAP-CH-DataTypes : 284

callDiversionTreatmentIndicator.....identifier of [20] CallDiversionTreatmentIndicator
DEFINED in MAP-CH-DataTypes : 116

CallDiversionTreatmentIndicator.....type reference OCTET STRING
DEFINED in MAP-CH-DataTypes : 140
USED in MAP-CH-DataTypes : 116

calledPartySS-InteractionViolation.....identifier of Named Number, 7
DEFINED in MAP-ER-DataTypes : 126

callForwardingData.....identifier of [1] CallForwardingData
DEFINED in MAP-MS-DataTypes : 1786

CallForwardingData.....type reference SEQUENCE
DEFINED in MAP-MS-DataTypes : 1815
USED in MAP-MS-DataTypes : 1786

callInfo.....identifier of [1] ExternalSignalInfo
DEFINED in MAP-CH-DataTypes : 387

callInfo.....identifier of [3] ExternalSignalInfo
DEFINED in MAP-SS-DataTypes : 314

callOriginator.....identifier of [8] NULL
DEFINED in MAP-GR-DataTypes : 123

callOutcome.....identifier of [1] CallOutcome
DEFINED in MAP-CH-DataTypes : 359

CallOutcome.....type reference ENUMERATED
DEFINED in MAP-CH-DataTypes : 371
USED in MAP-CH-DataTypes : 359

callReferenceNumber.....identifier of [7] CallReferenceNumber
DEFINED in MAP-CH-DataTypes : 102

CallReferenceNumber.....type reference OCTET STRING
DEFINED in MAP-CH-DataTypes : 127
USED in MAP-CH-DataTypes : 22 102 207 225

callReferenceNumber.....identifier of [9] CallReferenceNumber
DEFINED in MAP-CH-DataTypes : 207

callReferenceNumber.....identifier of [0] CallReferenceNumber
DEFINED in MAP-CH-DataTypes : 225

callrelated.....value reference SS-Code, '10110010'B
DEFINED in MAP-SS-Code : 163

callReportdata.....identifier of [2] CallReportData
DEFINED in MAP-CH-DataTypes : 348

CallReportData.....type reference SEQUENCE
DEFINED in MAP-CH-DataTypes : 357
USED in MAP-CH-DataTypes : 348

callTerminationIndicator.....identifier of [2] CallTerminationIndicator
DEFINED in MAP-CH-DataTypes : 424

CallTerminationIndicator.....type reference ENUMERATED
DEFINED in MAP-CH-DataTypes : 437
USED in MAP-CH-DataTypes : 424

callToClientNotSetup.....identifier of Named Number, 2
DEFINED in MAP-ER-DataTypes : 348

callTypeCriteria.....identifier of [2] CallTypeCriteria
DEFINED in MAP-MS-DataTypes : 1389

CallTypeCriteria.....type reference ENUMERATED
DEFINED in MAP-MS-DataTypes : 1427
USED in MAP-MS-DataTypes : 1389

callunrelated.....value reference SS-Code, '10110011'B
DEFINED in MAP-SS-Code : 166

call-Direction.....identifier of [2] CallDirection

DEFINED in MAP-CH-DataTypes : 284

camelBusy.....identifier of [1] NULL
DEFINED in MAP-MS-DataTypes : 1750

camelCapabilityHandling.....identifier of [1] CamelCapabilityHandling
DEFINED in MAP-MS-DataTypes : 776

camelCapabilityHandling.....identifier of [1] CamelCapabilityHandling
DEFINED in MAP-MS-DataTypes : 1281

camelCapabilityHandling.....identifier of [0] CamelCapabilityHandling
DEFINED in MAP-MS-DataTypes : 1340

CamelCapabilityHandling.....type reference INTEGER
DEFINED in MAP-MS-DataTypes : 1458
USED in MAP-MS-DataTypes : 71 776 1281 1340 1473 1548

camelCapabilityHandling.....identifier of [1] CamelCapabilityHandling
DEFINED in MAP-MS-DataTypes : 1473

camelCapabilityHandling.....identifier of [0] CamelCapabilityHandling
DEFINED in MAP-MS-DataTypes : 1548

camelInfo.....identifier of [11] CamelInfo
DEFINED in MAP-CH-DataTypes : 106

CamelInfo.....type reference SEQUENCE
DEFINED in MAP-CH-DataTypes : 256
USED in MAP-CH-DataTypes : 106

camelRoutingInfo.....identifier of [8] CamelRoutingInfo
DEFINED in MAP-CH-DataTypes : 264

CamelRoutingInfo.....type reference SEQUENCE
DEFINED in MAP-CH-DataTypes : 266
USED in MAP-CH-DataTypes : 264

camelSubscriptionInfoWithdraw.....identifier of [9] NULL
DEFINED in MAP-MS-DataTypes : 1218

camel-invoked.....identifier of Named Number, 1
DEFINED in MAP-SS-DataTypes : 320

camel-SubscriptionInfo.....identifier of [4] CAMEL-SubscriptionInfo
DEFINED in MAP-MS-DataTypes : 1789

CAMEL-SubscriptionInfo.....type reference SEQUENCE
DEFINED in MAP-MS-DataTypes : 1837
USED in MAP-MS-DataTypes : 1789 1867 1911

camel-SubscriptionInfo.....identifier of [1] CAMEL-SubscriptionInfo
DEFINED in MAP-MS-DataTypes : 1867

camel-SubscriptionInfo.....identifier of [3] CAMEL-SubscriptionInfo
DEFINED in MAP-MS-DataTypes : 1911

cancellationType.....identifier of CancellationType
DEFINED in MAP-MS-DataTypes : 251

CancellationType.....type reference ENUMERATED
DEFINED in MAP-MS-DataTypes : 256
USED in MAP-MS-DataTypes : 251

cancelLocation.....value reference CancelLocation, CHOICE VALUE
DEFINED in MAP-Protocol : 182

CancelLocation.....type reference OPERATION
DEFINED in MAP-MobileServiceOpera : 188
USED in MAP-Protocol : 13 182
USED in MAP-MobileServiceOpera : 16

cancelLocationArg.....identifier of CancelLocationArg
DEFINED in MAP-MobileServiceOpera : 190

CancelLocationArg.....type reference [3] SEQUENCE
DEFINED in MAP-MS-DataTypes : 249
USED in MAP-MobileServiceOpera : 115 190
USED in MAP-MS-DataTypes : 18

cancelLocationRes.....identifier of CancelLocationRes
DEFINED in MAP-MobileServiceOpera : 192

CancelLocationRes.....type reference SEQUENCE
DEFINED in MAP-MS-DataTypes : 263
USED in MAP-MobileServiceOpera : 116 192
USED in MAP-MS-DataTypes : 19

category.....identifier of [2] Category
DEFINED in MAP-MS-DataTypes : 884

Category.....type reference OCTET STRING
DEFINED in MAP-MS-DataTypes : 901
USED in MAP-MS-DataTypes : 884

CauseValue.....type reference OCTET STRING
DEFINED in MAP-MS-DataTypes : 1445
USED in MAP-MS-DataTypes : 1436 1439

ccbsIdle.....identifier of Named Number, 1
DEFINED in MAP-CH-DataTypes : 337

ccbsNotIdle.....identifier of Named Number, 0
DEFINED in MAP-CH-DataTypes : 336

ccbsNotReachable.....identifier of Named Number, 2
DEFINED in MAP-CH-DataTypes : 338

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

ccbs-Busy.....identifier of [1] NULL
DEFINED in MAP-ER-DataTypes : 265

ccbs-Call.....identifier of [15] NULL
DEFINED in MAP-CH-DataTypes : 111

ccbs-Call.....identifier of [13] NULL
DEFINED in MAP-CH-DataTypes : 212

ccbs-Data.....identifier of [1] CCBS-Data
DEFINED in MAP-SS-DataTypes : 307

CCBS-Data.....type reference SEQUENCE
DEFINED in MAP-SS-DataTypes : 310
USED in MAP-SS-DataTypes : 307

ccbs-Feature.....identifier of [2] CCBS-Feature
DEFINED in MAP-CH-DataTypes : 388

CCBS-Feature.....type reference SEQUENCE
DEFINED in MAP-SS-DataTypes : 206
USED in MAP-CH-DataTypes : 58 388
USED in MAP-SS-DataTypes : 36 202 311 325

ccbs-Feature.....identifier of [0] CCBS-Feature
DEFINED in MAP-SS-DataTypes : 311

ccbs-Feature.....identifier of [0] CCBS-Feature
DEFINED in MAP-SS-DataTypes : 325

ccbs-FeatureList.....identifier of [2] CCBS-FeatureList
DEFINED in MAP-SS-DataTypes : 196

CCBS-FeatureList.....type reference SEQUENCE OF
DEFINED in MAP-SS-DataTypes : 201
USED in MAP-SS-DataTypes : 196

ccbs-Index.....identifier of [0] CCBS-Index
DEFINED in MAP-SS-DataTypes : 207

CCBS-Index.....type reference INTEGER
DEFINED in MAP-SS-DataTypes : 213
USED in MAP-SS-DataTypes : 207 330

ccbs-Index.....identifier of [1] CCBS-Index
DEFINED in MAP-SS-DataTypes : 330

ccbs-Indicators.....identifier of [11] CCBS-Indicators
DEFINED in MAP-CH-DataTypes : 162

CCBS-Indicators.....type reference SEQUENCE

DEFINED in MAP-CH-DataTypes : 177
USED in MAP-CH-DataTypes : 162

ccbs-Monitoring.....identifier of [2] ReportingState
DEFINED in MAP-CH-DataTypes : 318

ccbs-Possible.....identifier of [0] NULL
DEFINED in MAP-CH-DataTypes : 178

ccbs-Possible.....identifier of [8] NULL
DEFINED in MAP-CH-DataTypes : 232

ccbs-Possible.....identifier of [0] NULL
DEFINED in MAP-ER-DataTypes : 264

ccbs-RequestState.....identifier of [6] CCBS-RequestState
DEFINED in MAP-SS-DataTypes : 282

CCBS-RequestState.....type reference ENUMERATED
DEFINED in MAP-SS-DataTypes : 285
USED in MAP-SS-DataTypes : 282

ccbs-SubscriberStatus.....identifier of [0] CCBS-SubscriberStatus
DEFINED in MAP-CH-DataTypes : 331

CCBS-SubscriberStatus.....type reference ENUMERATED
DEFINED in MAP-CH-DataTypes : 335
USED in MAP-CH-DataTypes : 331 353

ccbs-SubscriberStatus.....identifier of [0] CCBS-SubscriberStatus
DEFINED in MAP-CH-DataTypes : 353

cd.....value reference SS-Code, '00100100'B
DEFINED in MAP-SS-Code : 60

cellGlobalIdOrServiceAreaIdFixedLength..identifier of [0] CellGlobalIdOrServiceAreaIdFixedLength
DEFINED in MAP-CommonDataTypes : 393

CellGlobalIdOrServiceAreaIdFixedLength..type reference OCTET STRING
DEFINED in MAP-CommonDataTypes : 396
USED in MAP-CommonDataTypes : 393

cellGlobalIdOrServiceAreaIdOrLAI.....identifier of [3] CellGlobalIdOrServiceAreaIdOrLAI
DEFINED in MAP-MS-DataTypes : 1710

CellGlobalIdOrServiceAreaIdOrLAI.....type reference CHOICE
DEFINED in MAP-CommonDataTypes : 392
USED in MAP-MS-DataTypes : 172 1710
USED in MAP-CommonDataTypes : 45

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 : 305

chargeableECT-Barred.....identifier of Named Number, 10
DEFINED in MAP-MS-DataTypes : 935

chargingCharacteristics.....identifier of [18] ChargingCharacteristics
DEFINED in MAP-MS-DataTypes : 707

ChargingCharacteristics.....type reference OCTET STRING
DEFINED in MAP-MS-DataTypes : 846
USED in MAP-MS-DataTypes : 707 754

checkIMEI.....value reference CheckIMEI, CHOICE VALUE
DEFINED in MAP-Protocol : 205

CheckIMEI.....type reference OPERATION
DEFINED in MAP-MobileServiceOpera : 380
USED in MAP-Protocol : 24 205

USED in MAP-MobileServiceOpera : 49

chosenChannel.....identifier of [4] ExternalSignalInfo
DEFINED in MAP-CH-DataTypes : 286

chosenChannel.....identifier of [1] ExternalSignalInfo
DEFINED in MAP-CH-DataTypes : 306

chosenChannel.....identifier of [0] ExternalSignalInfo
DEFINED in MAP-CH-DataTypes : 311

chosenChannelInfo.....identifier of [0] ChosenChannelInfo
DEFINED in MAP-MS-DataTypes : 550

ChosenChannelInfo.....type reference OCTET STRING
DEFINED in MAP-MS-DataTypes : 555
USED in MAP-MS-DataTypes : 550

ChosenEncryptionAlgorithm.....type reference OCTET STRING
DEFINED in MAP-MS-DataTypes : 542
USED in MAP-MS-DataTypes : 531

ChosenIntegrityProtectionAlgorithm.....type reference OCTET STRING
DEFINED in MAP-MS-DataTypes : 535
USED in MAP-MS-DataTypes : 530

chosenRadioResourceInformation.....identifier of [6] ChosenRadioResourceInformation
DEFINED in MAP-MS-DataTypes : 525

ChosenRadioResourceInformation.....type reference SEQUENCE
DEFINED in MAP-MS-DataTypes : 549
USED in MAP-MS-DataTypes : 525 582

chosenRadioResourceInformation.....identifier of [3] ChosenRadioResourceInformation
DEFINED in MAP-MS-DataTypes : 582

chosenSpeechVersion.....identifier of [1] ChosenSpeechVersion
DEFINED in MAP-MS-DataTypes : 551

ChosenSpeechVersion.....type reference OCTET STRING
DEFINED in MAP-MS-DataTypes : 559
USED in MAP-MS-DataTypes : 551

cipheringAlgorithm.....identifier of CipheringAlgorithm
DEFINED in MAP-GR-DataTypes : 53

CipheringAlgorithm.....type reference OCTET STRING
DEFINED in MAP-GR-DataTypes : 101
USED in MAP-GR-DataTypes : 53

ck.....identifier of CK
DEFINED in MAP-MS-DataTypes : 322

ck.....identifier of CK
DEFINED in MAP-MS-DataTypes : 337

CK.....type reference OCTET STRING
DEFINED in MAP-MS-DataTypes : 350
USED in MAP-MS-DataTypes : 322 337

cksn.....identifier of Cksn
DEFINED in MAP-MS-DataTypes : 333

Cksn.....type reference OCTET STRING
DEFINED in MAP-MS-DataTypes : 358
USED in MAP-MS-DataTypes : 333

clientIdentity.....identifier of LCSClientExternalID
DEFINED in MAP-MS-DataTypes : 1148

clientNotInMSPrivacyExceptionList.....identifier of Named Number, 1
DEFINED in MAP-ER-DataTypes : 347

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 : 172

CliRestrictionOption.....type reference
ENUMERATED

DEFINED in MAP-SS-DataTypes : 175
USED in MAP-SS-DataTypes : 29 172 192

cliRestrictionOption.....identifier of CliRestrictionOption
DEFINED in MAP-SS-DataTypes : 192

clir-invoked.....identifier of Named Number, 0
DEFINED in MAP-SS-DataTypes : 319

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 : 95
USED in MAP-GR-DataTypes : 52

collectedInfo.....identifier of Named Number, 2
DEFINED in MAP-MS-DataTypes : 1368

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

completed.....identifier of Named Number, 3
DEFINED in MAP-SS-DataTypes : 289

completeDataListIncluded.....identifier of NULL
DEFINED in MAP-MS-DataTypes : 760

completeDataListIncluded.....identifier of NULL
DEFINED in MAP-MS-DataTypes : 866

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 : 361

ContextId.....type reference INTEGER
DEFINED in MAP-MS-DataTypes : 757
USED in MAP-MS-DataTypes : 745 1247

contextIdList.....identifier of ContextIdList
DEFINED in MAP-MS-DataTypes : 1244

ContextIdList.....type reference SEQUENCE OF
DEFINED in MAP-MS-DataTypes : 1246
USED in MAP-MS-DataTypes : 1244

Continue.....type reference SEQUENCE
DEFINED in TCAPMessages : 69
USED in TCAPMessages : 55

continueCall.....identifier of Named Number, 0
DEFINED in MAP-MS-DataTypes : 1451

continueTransaction.....identifier of Named Number, 0
DEFINED in MAP-MS-DataTypes : 803

continueTransaction.....identifier of Named Number, 0

DEFINED in MAP-MS-DataTypes : 1508

continue-ME.....identifier of [APPLICATION 5] IMPLICIT Continue
DEFINED in TCAPMessages : 55

controllingMSC.....identifier of Named Number, 4
DEFINED in MAP-CommonDataTypes : 353

csiActive.....identifier of [2] NULL
DEFINED in MAP-MS-DataTypes : 1342

csi-Active.....identifier of [4] NULL
DEFINED in MAP-MS-DataTypes : 779

csi-Active.....identifier of [4] NULL
DEFINED in MAP-MS-DataTypes : 1284

csi-Active.....identifier of [1] NULL
DEFINED in MAP-MS-DataTypes : 1312

csi-Active.....identifier of [4] NULL
DEFINED in MAP-MS-DataTypes : 1476

csi-Active.....identifier of [3] NULL
DEFINED in MAP-MS-DataTypes : 1521

csi-Active.....identifier of [2] NULL
DEFINED in MAP-MS-DataTypes : 1550

cs-AllocationRetentionPriority.....identifier of [29] CS-AllocationRetentionPriority
DEFINED in MAP-MS-DataTypes : 705

CS-AllocationRetentionPriority.....type reference OCTET STRING
DEFINED in MAP-MS-DataTypes : 712
USED in MAP-MS-DataTypes : 705

cug.....value reference SS-Code, '01100001'B
DEFINED in MAP-SS-Code : 96

cugIC-CallBarred.....identifier of Named Number, 1
DEFINED in MAP-MS-DataTypes : 1077

cugOG-CallBarred.....identifier of Named Number, 2
DEFINED in MAP-MS-DataTypes : 1078

cugSubscriptionFlag.....identifier of [6] NULL
DEFINED in MAP-CH-DataTypes : 152

CUG-CheckInfo.....type reference SEQUENCE
DEFINED in MAP-CH-DataTypes : 86
USED in MAP-CH-DataTypes : 96 151 229

cug-CheckInfo.....identifier of [1] CUG-CheckInfo
DEFINED in MAP-CH-DataTypes : 96

cug-CheckInfo.....identifier of [3] CUG-CheckInfo
DEFINED in MAP-CH-DataTypes : 151

cug-CheckInfo.....identifier of [4] CUG-CheckInfo
DEFINED in MAP-CH-DataTypes : 229

CUG-Feature.....type reference SEQUENCE
DEFINED in MAP-MS-DataTypes : 1090
USED in MAP-MS-DataTypes : 1083

cug-FeatureList.....identifier of CUG-FeatureList
DEFINED in MAP-MS-DataTypes : 1055

CUG-FeatureList.....type reference SEQUENCE OF
DEFINED in MAP-MS-DataTypes : 1082
USED in MAP-MS-DataTypes : 1055

cug-Index.....identifier of CUG-Index
DEFINED in MAP-MS-DataTypes : 1063

CUG-Index.....type reference INTEGER
DEFINED in MAP-MS-DataTypes : 1070
USED in MAP-MS-DataTypes : 75 1063 1092

cug-Info.....identifier of [2] CUG-Info
DEFINED in MAP-MS-DataTypes : 974

CUG-Info.....type reference SEQUENCE
DEFINED in MAP-MS-DataTypes : 1053
USED in MAP-MS-DataTypes : 76 974

cug-Interlock.....identifier of CUG-Interlock
DEFINED in MAP-MS-DataTypes : 1064

CUG-Interlock.....type reference OCTET STRING
DEFINED in MAP-MS-DataTypes : 1073
USED in MAP-MS-DataTypes : 77 1064
USED in MAP-CH-DataTypes : 42 87

cug-Interlock.....identifier of CUG-Interlock
DEFINED in MAP-CH-DataTypes : 87

cug-OutgoingAccess.....identifier of NULL
DEFINED in MAP-CH-DataTypes : 88

cug-Reject.....value reference CUG-Reject, CHOICE VALUE
DEFINED in MAP-Protocol : 382

CUG-Reject.....type reference ERROR
DEFINED in MAP-Errors : 311
USED in MAP-Protocol : 141 382
USED in MAP-CallHandlingOperat : 46 100
USED in MAP-Errors : 52

cug-RejectCause.....identifier of CUG-RejectCause
DEFINED in MAP-ER-DataTypes : 118

CUG-RejectCause.....type reference ENUMERATED
DEFINED in MAP-ER-DataTypes : 122
USED in MAP-ER-DataTypes : 118

cug-RejectParam.....identifier of CUG-RejectParam
DEFINED in MAP-Errors : 313

CUG-RejectParam.....type reference SEQUENCE
DEFINED in MAP-ER-DataTypes : 117
USED in MAP-Errors : 131 313
USED in MAP-ER-DataTypes : 16

CUG-Subscription.....type reference SEQUENCE
DEFINED in MAP-MS-DataTypes : 1062
USED in MAP-MS-DataTypes : 1060

cug-SubscriptionList.....identifier of CUG-SubscriptionList
DEFINED in MAP-MS-DataTypes : 1054

CUG-SubscriptionList.....type reference SEQUENCE OF
DEFINED in MAP-MS-DataTypes : 1059
USED in MAP-MS-DataTypes : 1054

currentLocation.....identifier of [3] NULL
DEFINED in MAP-MS-DataTypes : 1701

currentLocation.....identifier of Named Number, 0
DEFINED in MAP-LCS-DataTypes : 95

currentLocationRetrieved.....identifier of [8] NULL
DEFINED in MAP-MS-DataTypes : 1716

currentOrLastKnownLocation.....identifier of Named Number, 1
DEFINED in MAP-LCS-DataTypes : 96

currentPassword.....identifier of Password
DEFINED in MAP-SupplementaryServi : 238

currentSecurityContext.....identifier of [2] CurrentSecurityContext
DEFINED in MAP-MS-DataTypes : 297

CurrentSecurityContext.....type reference CHOICE
DEFINED in MAP-MS-DataTypes : 327
USED in MAP-MS-DataTypes : 297

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 : 123

dataMissing.....value reference DataMissing, CHOICE VALUE
DEFINED in MAP-Protocol : 329

DataMissing.....type reference ERROR
DEFINED in MAP-Errors : 169
USED in MAP-Protocol : 117 329
USED in MAP-MobileServiceOpera : 83 183 195 205 215 239 252 265 282
303 317 344 363 387 399 410 429 443
457 471 482
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 : 171

DataMissingParam.....type reference SEQUENCE
DEFINED in MAP-ER-DataTypes : 181
USED in MAP-Errors : 111 171
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 : 1900

deactivateSS.....value reference DeactivateSS, CHOICE VALUE
DEFINED in MAP-Protocol : 248

DeactivateSS.....type reference OPERATION
DEFINED in MAP-SupplementaryServi : 141
USED in MAP-Protocol : 72 248
USED in MAP-SupplementaryServi : 16

deactivateTraceMode.....value reference DeactivateTraceMode, CHOICE VALUE
DEFINED in MAP-Protocol : 225

DeactivateTraceMode.....type reference OPERATION
DEFINED in MAP-OperationAndMainte : 64
USED in MAP-Protocol : 47 225
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 : 1303

defaultCallHandling.....identifier of [1] DefaultCallHandling
DEFINED in MAP-MS-DataTypes : 1360

DefaultCallHandling.....type reference ENUMERATED
DEFINED in MAP-MS-DataTypes : 1450
USED in MAP-MS-DataTypes : 70 1303 1360 1568

defaultCallHandling.....identifier of [1] DefaultCallHandling
DEFINED in MAP-MS-DataTypes : 1568

DefaultGPRS-Handling.....type reference ENUMERATED
DEFINED in MAP-MS-DataTypes : 802
USED in MAP-MS-DataTypes : 797

defaultPriority.....identifier of EMLPP-Priority
DEFINED in MAP-CommonDataTypes : 437

defaultPriority.....identifier of [7] EMLPP-Priority
DEFINED in MAP-SS-DataTypes : 79

defaultPriority.....identifier of EMLPP-Priority
DEFINED in MAP-SS-DataTypes : 167

defaultPriority.....identifier of [1] EMLPP-Priority
DEFINED in MAP-SS-DataTypes : 195

defaultSessionHandling.....identifier of [3] DefaultGPRS-Handling
DEFINED in MAP-MS-DataTypes : 797

defaultSMS-Handling.....identifier of [3] DefaultSMS-Handling
DEFINED in MAP-MS-DataTypes : 1494

DefaultSMS-Handling.....type reference ENUMERATED
DEFINED in MAP-MS-DataTypes : 1507
USED in MAP-MS-DataTypes : 1494

delaytolerant.....identifier of Named Number, 1
DEFINED in MAP-LCS-DataTypes : 169

deleted.....identifier of Named Number, 6
DEFINED in MAP-SS-DataTypes : 292

deleteSubscriberData.....value reference DeleteSubscriberData, CHOICE VALUE
DEFINED in MAP-Protocol : 211

DeleteSubscriberData.....type reference OPERATION
DEFINED in MAP-MobileServiceOpera : 403
USED in MAP-Protocol : 26 211
USED in MAP-MobileServiceOpera : 53

deleteSubscriberDataArg.....identifier of DeleteSubscriberDataArg
DEFINED in MAP-MobileServiceOpera : 405

DeleteSubscriberDataArg.....type reference SEQUENCE
DEFINED in MAP-MS-DataTypes : 1208
USED in MAP-MobileServiceOpera : 138 405
USED in MAP-MS-DataTypes : 56

deleteSubscriberDataRes.....identifier of DeleteSubscriberDataRes
DEFINED in MAP-MobileServiceOpera : 407

DeleteSubscriberDataRes.....type reference SEQUENCE

DEFINED in MAP-MS-DataTypes : 1261

USED in MAP-MobileServiceOpera : 139 407
USED in MAP-MS-DataTypes : 57

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 : 1387

DestinationNumberCriteria.....type reference SEQUENCE
DEFINED in MAP-MS-DataTypes : 1400
USED in MAP-MS-DataTypes : 1387

destinationNumberLengthList.....identifier of [2] DestinationNumberLengthList
DEFINED in MAP-MS-DataTypes : 1403

DestinationNumberLengthList.....type reference SEQUENCE OF
DEFINED in MAP-MS-DataTypes : 1413
USED in MAP-MS-DataTypes : 1403

destinationNumberList.....identifier of [1] DestinationNumberList
DEFINED in MAP-MS-DataTypes : 1402

DestinationNumberList.....type reference SEQUENCE OF
DEFINED in MAP-MS-DataTypes : 1408
USED in MAP-MS-DataTypes : 1402

DestTransactionID.....type reference [APPLICATION 9] IMPLICIT TransactionID
DEFINED in TCAPMessages : 98
USED in TCAPMessages : 65 70 74

diagnosticInfo.....identifier of SignalInfo
DEFINED in MAP-ER-DataTypes : 151

dialledNumber.....identifier of ISDN-AddressString
DEFINED in MAP-MS-DataTypes : 1300

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 : 350

doublyChargeableECT-Barred.....identifier of Named Number, 13
DEFINED in MAP-MS-DataTypes : 938

downlinkAttached.....identifier of [5] NULL
DEFINED in MAP-GR-DataTypes : 120

dp-AnalysedInfoCriteriaList.....identifier of [0] DP-AnalysedInfoCriteriaList
DEFINED in MAP-MS-DataTypes : 1280

DP-AnalysedInfoCriteriaList.....type reference SEQUENCE OF
DEFINED in MAP-MS-DataTypes : 1294
USED in MAP-MS-DataTypes : 1280

DP-AnalysedInfoCriterium.....type reference SEQUENCE
DEFINED in MAP-MS-DataTypes : 1299
USED in MAP-MS-DataTypes : 1295

dtid.....identifier of DestTransactionID
DEFINED in TCAPMessages : 65

dtid.....identifier of DestTransactionID
DEFINED in TCAPMessages : 70

dtid.....identifier of DestTransactionID
DEFINED in TCAPMessages : 74

dualCommunication.....identifier of [7] NULL
DEFINED in MAP-GR-DataTypes : 122

duplicateInvokeID.....identifier of Named Number, 0
DEFINED in TCAPMessages : 183

d-csi.....identifier of Named Number, 3
DEFINED in MAP-MS-DataTypes : 1232

d-CSI.....identifier of [9] D-CSI
DEFINED in MAP-MS-DataTypes : 1277

D-CSI.....type reference SEQUENCE
DEFINED in MAP-MS-DataTypes : 1279
USED in MAP-MS-DataTypes : 65 1277 1840
USED in MAP-CH-DataTypes : 44 237 279

d-csi.....identifier of Named Number, 8
DEFINED in MAP-MS-DataTypes : 1813

d-CSI.....identifier of [2] D-CSI
DEFINED in MAP-MS-DataTypes : 1840

d-csi.....identifier of [12] D-CSI
DEFINED in MAP-CH-DataTypes : 237

d-csi.....identifier of [5] D-CSI
DEFINED in MAP-CH-DataTypes : 279

ect.....value reference SS-Code, '00110001'B
DEFINED in MAP-SS-Code : 66

eir.....identifier of Named Number, 6
DEFINED in MAP-CommonDataTypes : 355

ellipsoidArc.....identifier of Named Number, 6
DEFINED in MAP-LCS-DataTypes : 181

ellipsoidPoint.....identifier of Named Number, 0
DEFINED in MAP-LCS-DataTypes : 175

ellipsoidPointWithAltitude.....identifier of Named Number, 4
DEFINED in MAP-LCS-DataTypes : 179

ellipsoidPointWithAltitudeAndUncertaintyidentifier of Named Number, 5
DEFINED in MAP-LCS-DataTypes : 180

ellipsoidPointWithUncertaintyCircle.....identifier of Named Number, 1
DEFINED in MAP-LCS-DataTypes : 176

ellipsoidPointWithUncertaintyEllipse....identifier of Named Number, 2
DEFINED in MAP-LCS-DataTypes : 177

emergencyCallOrigination.....identifier of Named Number, 0
DEFINED in MAP-LCS-DataTypes : 303

emergencyCallRelease.....identifier of Named Number, 1
DEFINED in MAP-LCS-DataTypes : 304

emergencyCalls.....value reference TeleserviceCode, '00010010'B
DEFINED in MAP-TS-Code : 42

emergencyServices.....identifier of Named Number, 0
DEFINED in MAP-LCS-DataTypes : 112

emlpp.....value reference SS-Code, '10100001'B
DEFINED in MAP-SS-Code : 156

emlpp-Info.....identifier of [4] EMLPP-Info
DEFINED in MAP-MS-DataTypes : 976

EMLPP-Info.....type reference SEQUENCE
DEFINED in MAP-CommonDataTypes : 435
USED in MAP-MS-DataTypes : 175 976
USED in MAP-CommonDataTypes : 50

EMLPP-Priority.....type reference INTEGER
DEFINED in MAP-CommonDataTypes : 441
USED in MAP-CommonDataTypes : 51 436 437 447 448 449 450 451 452 453
USED in MAP-SS-DataTypes : 51 79 167 194 195
USED in MAP-GR-DataTypes : 25 56

enabling.....identifier of Named Number, 1
DEFINED in MAP-MS-DataTypes : 1433

encryptionAlgorithm.....identifier of [1] ChosenEncryptionAlgorithm
DEFINED in MAP-MS-DataTypes : 531

encryptionAlgorithms.....identifier of [1] PermittedEncryptionAlgorithms
DEFINED in MAP-MS-DataTypes : 429

encryptionInfo.....identifier of [1] EncryptionInformation
DEFINED in MAP-MS-DataTypes : 410

encryptionInfo.....identifier of [6] EncryptionInformation
DEFINED in MAP-MS-DataTypes : 471

EncryptionInformation.....type reference OCTET STRING
DEFINED in MAP-MS-DataTypes : 644
USED in MAP-MS-DataTypes : 410 471

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 : 250

enterNewPW-Again.....identifier of Named Number, 2
DEFINED in MAP-SS-DataTypes : 251

enterPW.....identifier of Named Number, 0
DEFINED in MAP-SS-DataTypes : 249

equipmentNotSM-Equipped.....identifier of Named Number, 2
DEFINED in MAP-ER-DataTypes : 143

equipmentProtocolError.....identifier of Named Number, 1
DEFINED in MAP-ER-DataTypes : 142

equipmentStatus.....identifier of EquipmentStatus
DEFINED in MAP-MobileServiceOpera : 384

EquipmentStatus.....type reference ENUMERATED
DEFINED in MAP-MS-DataTypes : 680
USED in MAP-MobileServiceOpera : 135 384
USED in MAP-MS-DataTypes : 49

eraseCC-Entry.....value reference EraseCC-Entry, CHOICE VALUE
DEFINED in MAP-Protocol : 257

EraseCC-Entry.....type reference OPERATION
DEFINED in MAP-SupplementaryServi : 268
USED in MAP-Protocol : 81 257
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 : 328
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 : 333
USED in MAP-SupplementaryServi : 73 272
USED in MAP-SS-DataTypes : 40

eraseSS.....value reference EraseSS, CHOICE VALUE

DEFINED in MAP-Protocol : 246

EraseSS.....type reference OPERATION
DEFINED in MAP-SupplementaryServi : 104
USED in MAP-Protocol : 70 246
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 : 219

ets-300356.....identifier of Named Number, 1
DEFINED in MAP-CommonDataTypes : 230

eventMet.....identifier of [0] MM-Code
DEFINED in MAP-MS-DataTypes : 1925

eventReportData.....identifier of [1] EventReportData
DEFINED in MAP-CH-DataTypes : 347

EventReportData.....type reference SEQUENCE
DEFINED in MAP-CH-DataTypes : 352
USED in MAP-CH-DataTypes : 347

extendedRoutingInfo.....identifier of ExtendedRoutingInfo
DEFINED in MAP-CH-DataTypes : 150

ExtendedRoutingInfo.....type reference CHOICE
DEFINED in MAP-CH-DataTypes : 262
USED in MAP-CH-DataTypes : 150

extensibleCallBarredParam.....identifier of ExtensibleCallBarredParam
DEFINED in MAP-ER-DataTypes : 103

ExtensibleCallBarredParam.....type reference SEQUENCE
DEFINED in MAP-ER-DataTypes : 111
USED in MAP-ER-DataTypes : 103

extensibleSystemFailureParam.....identifier of ExtensibleSystemFailureParam
DEFINED in MAP-ER-DataTypes : 172

ExtensibleSystemFailureParam.....type reference SEQUENCE
DEFINED in MAP-ER-DataTypes : 176
USED in MAP-ER-DataTypes : 172

extensionContainer.....identifier of ExtensionContainer
DEFINED in MAP-MS-DataTypes : 213

extensionContainer.....identifier of ExtensionContainer
DEFINED in MAP-MS-DataTypes : 220

extensionContainer.....identifier of ExtensionContainer
DEFINED in MAP-MS-DataTypes : 246

extensionContainer.....identifier of ExtensionContainer
DEFINED in MAP-MS-DataTypes : 252

extensionContainer.....identifier of ExtensionContainer
DEFINED in MAP-MS-DataTypes : 264

extensionContainer.....identifier of ExtensionContainer
DEFINED in MAP-MS-DataTypes : 271

extensionContainer.....identifier of ExtensionContainer
DEFINED in MAP-MS-DataTypes : 277

extensionContainer.....identifier of ExtensionContainer
DEFINED in MAP-MS-DataTypes : 288

extensionContainer.....identifier of [3] ExtensionContainer
DEFINED in MAP-MS-DataTypes : 298

extensionContainer.....identifier of ExtensionContainer
DEFINED in MAP-MS-DataTypes : 367

extensionContainer.....identifier of ExtensionContainer
DEFINED in MAP-MS-DataTypes : 371

extensionContainer.....identifier of ExtensionContainer

DEFINED in MAP-MS-DataTypes : 384

extensionContainer.....identifier of [1] ExtensionContainer
DEFINED in MAP-MS-DataTypes : 391

extensionContainer.....identifier of ExtensionContainer
DEFINED in MAP-MS-DataTypes : 402

extensionContainer.....identifier of [3] ExtensionContainer
DEFINED in MAP-MS-DataTypes : 415

extensionContainer.....identifier of [2] ExtensionContainer
DEFINED in MAP-MS-DataTypes : 430

extensionContainer.....identifier of [8] ExtensionContainer
DEFINED in MAP-MS-DataTypes : 476

extensionContainer.....identifier of [4] ExtensionContainer
DEFINED in MAP-MS-DataTypes : 526

extensionContainer.....identifier of [2] ExtensionContainer
DEFINED in MAP-MS-DataTypes : 532

extensionContainer.....identifier of [5] ExtensionContainer
DEFINED in MAP-MS-DataTypes : 570

extensionContainer.....identifier of [0] ExtensionContainer
DEFINED in MAP-MS-DataTypes : 575

extensionContainer.....identifier of [0] ExtensionContainer
DEFINED in MAP-MS-DataTypes : 584

extensionContainer.....identifier of [0] ExtensionContainer
DEFINED in MAP-MS-DataTypes : 594

extensionContainer.....identifier of [0] ExtensionContainer
DEFINED in MAP-MS-DataTypes : 598

extensionContainer.....identifier of [2] ExtensionContainer
DEFINED in MAP-MS-DataTypes : 662

extensionContainer.....identifier of ExtensionContainer
DEFINED in MAP-MS-DataTypes : 674

extensionContainer.....identifier of [14] ExtensionContainer
DEFINED in MAP-MS-DataTypes : 691

extensionContainer.....identifier of [21] ExtensionContainer
DEFINED in MAP-MS-DataTypes : 751

extensionContainer.....identifier of [2] ExtensionContainer
DEFINED in MAP-MS-DataTypes : 765

extensionContainer.....identifier of [2] ExtensionContainer
DEFINED in MAP-MS-DataTypes : 771

extensionContainer.....identifier of [2] ExtensionContainer
DEFINED in MAP-MS-DataTypes : 777

extensionContainer.....identifier of [4] ExtensionContainer
DEFINED in MAP-MS-DataTypes : 798

extensionContainer.....identifier of [3] ExtensionContainer
DEFINED in MAP-MS-DataTypes : 862

extensionContainer.....identifier of [3] ExtensionContainer
DEFINED in MAP-MS-DataTypes : 872

extensionContainer.....identifier of ExtensionContainer
DEFINED in MAP-MS-DataTypes : 921

extensionContainer.....identifier of [0] ExtensionContainer
DEFINED in MAP-MS-DataTypes : 982

extensionContainer.....identifier of [9] ExtensionContainer
DEFINED in MAP-MS-DataTypes : 998

extensionContainer.....identifier of ExtensionContainer
DEFINED in MAP-MS-DataTypes : 1041

extensionContainer.....identifier of ExtensionContainer
DEFINED in MAP-MS-DataTypes : 1050

extensionContainer.....identifier of [0] ExtensionContainer
DEFINED in MAP-MS-DataTypes : 1056

extensionContainer.....identifier of [0] ExtensionContainer
DEFINED in MAP-MS-DataTypes : 1067

extensionContainer.....identifier of ExtensionContainer
DEFINED in MAP-MS-DataTypes : 1094

extensionContainer.....identifier of [5] ExtensionContainer
DEFINED in MAP-MS-DataTypes : 1114

extensionContainer.....identifier of [3] ExtensionContainer
DEFINED in MAP-MS-DataTypes : 1133

extensionContainer.....identifier of [2] ExtensionContainer
DEFINED in MAP-MS-DataTypes : 1153

extensionContainer.....identifier of [0] ExtensionContainer
DEFINED in MAP-MS-DataTypes : 1181

extensionContainer.....identifier of [7] ExtensionContainer
DEFINED in MAP-MS-DataTypes : 1199

extensionContainer.....identifier of [6] ExtensionContainer
DEFINED in MAP-MS-DataTypes : 1219

extensionContainer.....identifier of ExtensionContainer
DEFINED in MAP-MS-DataTypes : 1263

extensionContainer.....identifier of [1] ExtensionContainer
DEFINED in MAP-MS-DataTypes : 1268

extensionContainer.....identifier of [2] ExtensionContainer
DEFINED in MAP-MS-DataTypes : 1282

extensionContainer.....identifier of ExtensionContainer
DEFINED in MAP-MS-DataTypes : 1304

extensionContainer.....identifier of ExtensionContainer
DEFINED in MAP-MS-DataTypes : 1309

extensionContainer.....identifier of [0] ExtensionContainer
DEFINED in MAP-MS-DataTypes : 1320

extensionContainer.....identifier of ExtensionContainer
DEFINED in MAP-MS-DataTypes : 1338

extensionContainer.....identifier of [2] ExtensionContainer
DEFINED in MAP-MS-DataTypes : 1361

extensionContainer.....identifier of [4] ExtensionContainer
DEFINED in MAP-MS-DataTypes : 1392

extensionContainer.....identifier of [2] ExtensionContainer
DEFINED in MAP-MS-DataTypes : 1474

extensionContainer.....identifier of [4] ExtensionContainer
DEFINED in MAP-MS-DataTypes : 1495

extensionContainer.....identifier of [1] ExtensionContainer
DEFINED in MAP-MS-DataTypes : 1519

extensionContainer.....identifier of ExtensionContainer
DEFINED in MAP-MS-DataTypes : 1546

extensionContainer.....identifier of [2] ExtensionContainer
DEFINED in MAP-MS-DataTypes : 1569

extensionContainer.....identifier of [3] ExtensionContainer
DEFINED in MAP-MS-DataTypes : 1589

extensionContainer.....identifier of [3] ExtensionContainer
DEFINED in MAP-MS-DataTypes : 1596

extensionContainer.....identifier of [3] ExtensionContainer
DEFINED in MAP-MS-DataTypes : 1605

extensionContainer.....identifier of [1] ExtensionContainer
DEFINED in MAP-MS-DataTypes : 1610

extensionContainer.....identifier of [3] ExtensionContainer

DEFINED in MAP-MS-DataTypes : 1619

extensionContainer.....identifier of [0] ExtensionContainer
DEFINED in MAP-MS-DataTypes : 1623

extensionContainer.....identifier of ExtensionContainer
DEFINED in MAP-MS-DataTypes : 1637

extensionContainer.....identifier of ExtensionContainer
DEFINED in MAP-MS-DataTypes : 1644

extensionContainer.....identifier of ExtensionContainer
DEFINED in MAP-MS-DataTypes : 1661

extensionContainer.....identifier of ExtensionContainer
DEFINED in MAP-MS-DataTypes : 1667

extensionContainer.....identifier of [3] ExtensionContainer
DEFINED in MAP-MS-DataTypes : 1682

extensionContainer.....identifier of ExtensionContainer
DEFINED in MAP-MS-DataTypes : 1687

extensionContainer.....identifier of [2] ExtensionContainer
DEFINED in MAP-MS-DataTypes : 1693

extensionContainer.....identifier of [2] ExtensionContainer
DEFINED in MAP-MS-DataTypes : 1699

extensionContainer.....identifier of [4] ExtensionContainer
DEFINED in MAP-MS-DataTypes : 1711

extensionContainer.....identifier of [2] ExtensionContainer
DEFINED in MAP-MS-DataTypes : 1766

extensionContainer.....identifier of ExtensionContainer
DEFINED in MAP-MS-DataTypes : 1771

extensionContainer.....identifier of [3] ExtensionContainer
DEFINED in MAP-MS-DataTypes : 1781

extensionContainer.....identifier of [7] ExtensionContainer
DEFINED in MAP-MS-DataTypes : 1792

extensionContainer.....identifier of [6] ExtensionContainer
DEFINED in MAP-MS-DataTypes : 1801

extensionContainer.....identifier of [0] ExtensionContainer
DEFINED in MAP-MS-DataTypes : 1818

extensionContainer.....identifier of ExtensionContainer
DEFINED in MAP-MS-DataTypes : 1826

extensionContainer.....identifier of ExtensionContainer
DEFINED in MAP-MS-DataTypes : 1834

extensionContainer.....identifier of [13] ExtensionContainer
DEFINED in MAP-MS-DataTypes : 1851

extensionContainer.....identifier of [5] ExtensionContainer
DEFINED in MAP-MS-DataTypes : 1861

extensionContainer.....identifier of [2] ExtensionContainer
DEFINED in MAP-MS-DataTypes : 1868

extensionContainer.....identifier of [7] ExtensionContainer
DEFINED in MAP-MS-DataTypes : 1879

extensionContainer.....identifier of [6] ExtensionContainer
DEFINED in MAP-MS-DataTypes : 1889

extensionContainer.....identifier of [3] ExtensionContainer
DEFINED in MAP-MS-DataTypes : 1896

extensionContainer.....identifier of ExtensionContainer
DEFINED in MAP-MS-DataTypes : 1913

extensionContainer.....identifier of ExtensionContainer
DEFINED in MAP-MS-DataTypes : 1917

extensionContainer.....identifier of [6] ExtensionContainer

DEFINED in MAP-MS-DataTypes : 1930

extensionContainer.....identifier of ExtensionContainer
DEFINED in MAP-MS-DataTypes : 1934

extensionContainer.....identifier of [3] ExtensionContainer
DEFINED in MAP-MS-DataTypes : 1946

extensionContainer.....identifier of [5] ExtensionContainer
DEFINED in MAP-MS-DataTypes : 1955

extensionContainer.....identifier of ExtensionContainer
DEFINED in MAP-CommonDataTypes : 200

extensionContainer.....identifier of ExtensionContainer
DEFINED in MAP-CommonDataTypes : 226

extensionContainer.....identifier of ExtensionContainer
DEFINED in MAP-CommonDataTypes : 242

extensionContainer.....identifier of [1] ExtensionContainer
DEFINED in MAP-CommonDataTypes : 360

extensionContainer.....identifier of [1] ExtensionContainer
DEFINED in MAP-CommonDataTypes : 377

extensionContainer.....identifier of ExtensionContainer
DEFINED in MAP-CommonDataTypes : 438

extensionContainer.....identifier of [4] ExtensionContainer
DEFINED in MAP-CommonDataTypes : 461

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

extensionContainer.....identifier of ExtensionContainer
DEFINED in MAP-CH-DataTypes : 89

extensionContainer.....identifier of [13] ExtensionContainer
DEFINED in MAP-CH-DataTypes : 108

extensionContainer.....identifier of [0] ExtensionContainer
DEFINED in MAP-CH-DataTypes : 158

extensionContainer.....identifier of [2] ExtensionContainer
DEFINED in MAP-CH-DataTypes : 180

extensionContainer.....identifier of [7] ExtensionContainer
DEFINED in MAP-CH-DataTypes : 194

extensionContainer.....identifier of [11] ExtensionContainer
DEFINED in MAP-CH-DataTypes : 209

extensionContainer.....identifier of ExtensionContainer
DEFINED in MAP-CH-DataTypes : 221

extensionContainer.....identifier of [7] ExtensionContainer
DEFINED in MAP-CH-DataTypes : 231

extensionContainer.....identifier of [3] ExtensionContainer
DEFINED in MAP-CH-DataTypes : 243

extensionContainer.....identifier of ExtensionContainer
DEFINED in MAP-CH-DataTypes : 253

extensionContainer.....identifier of ExtensionContainer
DEFINED in MAP-CH-DataTypes : 259

extensionContainer.....identifier of [1] ExtensionContainer
DEFINED in MAP-CH-DataTypes : 269

extensionContainer.....identifier of [2] ExtensionContainer
DEFINED in MAP-CH-DataTypes : 275

extensionContainer.....identifier of [7] ExtensionContainer
DEFINED in MAP-CH-DataTypes : 289

extensionContainer.....identifier of [1] ExtensionContainer
DEFINED in MAP-CH-DataTypes : 301

extensionContainer.....identifier of [2] ExtensionContainer
DEFINED in MAP-CH-DataTypes : 307

extensionContainer.....identifier of [1] ExtensionContainer
DEFINED in MAP-CH-DataTypes : 312

extensionContainer.....identifier of [3] ExtensionContainer
DEFINED in MAP-CH-DataTypes : 319

extensionContainer.....identifier of [1] ExtensionContainer
DEFINED in MAP-CH-DataTypes : 332

extensionContainer.....identifier of [3] ExtensionContainer
DEFINED in MAP-CH-DataTypes : 349

extensionContainer.....identifier of [1] ExtensionContainer
DEFINED in MAP-CH-DataTypes : 354

extensionContainer.....identifier of [2] ExtensionContainer
DEFINED in MAP-CH-DataTypes : 360

extensionContainer.....identifier of [0] ExtensionContainer
DEFINED in MAP-CH-DataTypes : 382

extensionContainer.....identifier of [6] ExtensionContainer
DEFINED in MAP-CH-DataTypes : 392

extensionContainer.....identifier of [1] ExtensionContainer
DEFINED in MAP-CH-DataTypes : 397

extensionContainer.....identifier of [1] ExtensionContainer
DEFINED in MAP-CH-DataTypes : 418

extensionContainer.....identifier of [3] ExtensionContainer
DEFINED in MAP-CH-DataTypes : 425

extensionContainer.....identifier of [1] ExtensionContainer
DEFINED in MAP-CH-DataTypes : 430

extensionContainer.....identifier of ExtensionContainer
DEFINED in MAP-CH-DataTypes : 434

extensionContainer.....identifier of [4] ExtensionContainer
DEFINED in MAP-SS-DataTypes : 279

extensionContainer.....identifier of ExtensionContainer
DEFINED in MAP-SS-DataTypes : 296

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 : 173

extensionContainer.....identifier of ExtensionContainer

DEFINED in MAP-SM-DataTypes : 185

extensionContainer.....identifier of ExtensionContainer
DEFINED in MAP-SM-DataTypes : 202

extensionContainer.....identifier of ExtensionContainer
DEFINED in MAP-SM-DataTypes : 206

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 : 90

extensionContainer.....identifier of [2] ExtensionContainer
DEFINED in MAP-LCS-DataTypes : 58

extensionContainer.....identifier of [2] ExtensionContainer
DEFINED in MAP-LCS-DataTypes : 64

extensionContainer.....identifier of [1] ExtensionContainer
DEFINED in MAP-LCS-DataTypes : 70

extensionContainer.....identifier of [8] ExtensionContainer
DEFINED in MAP-LCS-DataTypes : 84

extensionContainer.....identifier of [4] ExtensionContainer
DEFINED in MAP-LCS-DataTypes : 145

extensionContainer.....identifier of [1] ExtensionContainer
DEFINED in MAP-LCS-DataTypes : 188

extensionContainer.....identifier of ExtensionContainer
DEFINED in MAP-LCS-DataTypes : 312

extensionContainer.....identifier of ExtensionContainer
DEFINED in MAP-ER-DataTypes : 93

extensionContainer.....identifier of ExtensionContainer
DEFINED in MAP-ER-DataTypes : 113

extensionContainer.....identifier of ExtensionContainer
DEFINED in MAP-ER-DataTypes : 119

extensionContainer.....identifier of ExtensionContainer
DEFINED in MAP-ER-DataTypes : 152

extensionContainer.....identifier of ExtensionContainer
DEFINED in MAP-ER-DataTypes : 159

extensionContainer.....identifier of ExtensionContainer
DEFINED in MAP-ER-DataTypes : 178

extensionContainer.....identifier of ExtensionContainer
DEFINED in MAP-ER-DataTypes : 182

extensionContainer.....identifier of ExtensionContainer
DEFINED in MAP-ER-DataTypes : 186

extensionContainer.....identifier of ExtensionContainer
DEFINED in MAP-ER-DataTypes : 190

extensionContainer.....identifier of ExtensionContainer
DEFINED in MAP-ER-DataTypes : 194

extensionContainer.....identifier of ExtensionContainer
DEFINED in MAP-ER-DataTypes : 198

extensionContainer.....identifier of ExtensionContainer

DEFINED in MAP-ER-DataTypes : 212

extensionContainer.....identifier of ExtensionContainer
DEFINED in MAP-ER-DataTypes : 216

extensionContainer.....identifier of ExtensionContainer
DEFINED in MAP-ER-DataTypes : 220

extensionContainer.....identifier of ExtensionContainer
DEFINED in MAP-ER-DataTypes : 224

extensionContainer.....identifier of ExtensionContainer
DEFINED in MAP-ER-DataTypes : 228

extensionContainer.....identifier of ExtensionContainer
DEFINED in MAP-ER-DataTypes : 232

extensionContainer.....identifier of ExtensionContainer
DEFINED in MAP-ER-DataTypes : 236

extensionContainer.....identifier of ExtensionContainer
DEFINED in MAP-ER-DataTypes : 240

extensionContainer.....identifier of ExtensionContainer
DEFINED in MAP-ER-DataTypes : 244

extensionContainer.....identifier of ExtensionContainer
DEFINED in MAP-ER-DataTypes : 262

extensionContainer.....identifier of ExtensionContainer
DEFINED in MAP-ER-DataTypes : 268

extensionContainer.....identifier of ExtensionContainer
DEFINED in MAP-ER-DataTypes : 272

extensionContainer.....identifier of ExtensionContainer
DEFINED in MAP-ER-DataTypes : 276

extensionContainer.....identifier of ExtensionContainer
DEFINED in MAP-ER-DataTypes : 280

extensionContainer.....identifier of ExtensionContainer
DEFINED in MAP-ER-DataTypes : 284

extensionContainer.....identifier of ExtensionContainer
DEFINED in MAP-ER-DataTypes : 288

extensionContainer.....identifier of ExtensionContainer
DEFINED in MAP-ER-DataTypes : 292

extensionContainer.....identifier of ExtensionContainer
DEFINED in MAP-ER-DataTypes : 296

extensionContainer.....identifier of ExtensionContainer
DEFINED in MAP-ER-DataTypes : 300

extensionContainer.....identifier of ExtensionContainer
DEFINED in MAP-ER-DataTypes : 304

extensionContainer.....identifier of ExtensionContainer
DEFINED in MAP-ER-DataTypes : 308

extensionContainer.....identifier of ExtensionContainer
DEFINED in MAP-ER-DataTypes : 315

extensionContainer.....identifier of ExtensionContainer
DEFINED in MAP-ER-DataTypes : 319

extensionContainer.....identifier of ExtensionContainer
DEFINED in MAP-ER-DataTypes : 323

extensionContainer.....identifier of ExtensionContainer
DEFINED in MAP-ER-DataTypes : 327

extensionContainer.....identifier of ExtensionContainer
DEFINED in MAP-ER-DataTypes : 337

extensionContainer.....identifier of [1] ExtensionContainer
DEFINED in MAP-ER-DataTypes : 342

extensionContainer.....identifier of [1] ExtensionContainer
DEFINED in MAP-ER-DataTypes : 357

extensionContainer.....identifier of ExtensionContainer
DEFINED in MAP-ER-DataTypes : 375

extensionContainer.....identifier of ExtensionContainer
DEFINED in MAP-ER-DataTypes : 379

extensionContainer.....identifier of ExtensionContainer
DEFINED in MAP-ER-DataTypes : 384

ExtensionContainer.....type reference SEQUENCE
DEFINED in MAP-ExtensionDataTypes : 38
USED in MAP-MS-DataTypes : 191 213 220 246 252 264 271 277 288
298 367 371 384 391 402 415 430 476
526 532 570 575 584 594 598 662 674
691 751 765 771 777 798 862 872 921
982 998 1041 1050 1056 1067 1094 1114 1133
1153 1181 1199 1219 1263 1268 1282 1304 1309
1320 1338 1361 1392 1474 1495 1519 1546 1569
1589 1596 1605 1610 1619 1623 1637 1644 1661
1667 1682 1687 1693 1699 1711 1766 1771 1781
1792 1801 1818 1826 1834 1851 1861 1868 1879
1889 1896 1913 1917 1930 1934 1946 1955
USED in MAP-CommonDataTypes : 81 200 226 242 360 377 438 461
USED in MAP-OM-DataTypes : 27 41 51 57 61
USED in MAP-CH-DataTypes : 79 89 108 158 180 194 209 221 231
243 253 259 269 275 289 301 307 312
319 332 349 354 360 382 392 397 418
425 430 434
USED in MAP-SS-DataTypes : 60 279 296
USED in MAP-SM-DataTypes : 45 56 82 88 110 116 124 129 149
173 185 202 206
USED in MAP-GR-DataTypes : 42 58 63 68 72 82 90
USED in MAP-LCS-DataTypes : 41 58 64 70 84 145 188 312
USED in MAP-ER-DataTypes : 85 93 113 119 152 159 178 182 186
190 194 198 212 216 220 224 228 232
236 240 244 262 268 272 276 280 284
288 292 296 300 304 308 315 319 323
327 337 342 357 375 379 384
USED in MAP-ExtensionDataTypes : 16

ExtensionSet.....information object set reference MAP-EXTENSION, Information Object Set
DEFINED in MAP-ExtensionDataTypes : 54
USED in MAP-ExtensionDataTypes : 48 50

externalAddress.....identifier of [0] AddressString
DEFINED in MAP-CommonDataTypes : 376

ExternalClient.....type reference SEQUENCE
DEFINED in MAP-MS-DataTypes : 1147
USED in MAP-MS-DataTypes : 1138

externalClientList.....identifier of [1] ExternalClientList
DEFINED in MAP-MS-DataTypes : 1129

ExternalClientList.....type reference SEQUENCE OF
DEFINED in MAP-MS-DataTypes : 1137
USED in MAP-MS-DataTypes : 1129

ExternalSignalInfo.....type reference SEQUENCE
DEFINED in MAP-CommonDataTypes : 195
USED in MAP-CommonDataTypes : 21
USED in MAP-CH-DataTypes : 66 105 203 204 282 283 286 287 288
305 306 311 387
USED in MAP-SS-DataTypes : 54 314 315

extId.....identifier of InformationObjectClassFieldType
DEFINED in MAP-ExtensionDataTypes : 47

extType.....identifier of InformationObjectClassFieldType
DEFINED in MAP-ExtensionDataTypes : 49

Ext-BasicServiceCode.....type reference CHOICE
DEFINED in MAP-CommonDataTypes : 431
USED in MAP-MS-DataTypes : 173 989 1048 1086 1091 1257 1417 1873 1884
USED in MAP-CommonDataTypes : 49
USED in MAP-CH-DataTypes : 70 104 155 226

Ext-BasicServiceGroupList.....type reference SEQUENCE OF
DEFINED in MAP-MS-DataTypes : 1085
USED in MAP-MS-DataTypes : 1066 1113

ext-BearerService.....identifier of [2] Ext-BearerServiceCode

DEFINED in MAP-CommonDataTypes : 432

Ext-BearerServiceCode.....type reference OCTET STRING
DEFINED in MAP-BS-Code : 25
USED in MAP-MS-DataTypes : 150 909
USED in MAP-CommonDataTypes : 71 432

Ext-CallBarFeatureList.....type reference SEQUENCE OF
DEFINED in MAP-MS-DataTypes : 1044
USED in MAP-MS-DataTypes : 1040 1822 1951

Ext-CallBarInfo.....type reference SEQUENCE
DEFINED in MAP-MS-DataTypes : 1038
USED in MAP-MS-DataTypes : 973

Ext-CallBarringFeature.....type reference SEQUENCE
DEFINED in MAP-MS-DataTypes : 1047
USED in MAP-MS-DataTypes : 1045

Ext-CallBarringInfoFor-CSE.....type reference SEQUENCE
DEFINED in MAP-MS-DataTypes : 1949
USED in MAP-MS-DataTypes : 1909 1939

Ext-ExternalSignalInfo.....type reference SEQUENCE
DEFINED in MAP-CommonDataTypes : 221
USED in MAP-CommonDataTypes : 22
USED in MAP-CH-DataTypes : 67 113 214

Ext-ForwardingInfoFor-CSE.....type reference SEQUENCE
DEFINED in MAP-MS-DataTypes : 1942
USED in MAP-MS-DataTypes : 1908 1938

Ext-ForwFeature.....type reference SEQUENCE
DEFINED in MAP-MS-DataTypes : 988
USED in MAP-MS-DataTypes : 986

Ext-ForwFeatureList.....type reference SEQUENCE OF
DEFINED in MAP-MS-DataTypes : 985
USED in MAP-MS-DataTypes : 981 1816 1944

Ext-ForwInfo.....type reference SEQUENCE
DEFINED in MAP-MS-DataTypes : 979
USED in MAP-MS-DataTypes : 972

Ext-ForwOptions.....type reference OCTET STRING
DEFINED in MAP-MS-DataTypes : 1002
USED in MAP-MS-DataTypes : 996

Ext-GeographicalInformation.....type reference OCTET STRING
DEFINED in MAP-LCS-DataTypes : 201
USED in MAP-LCS-DataTypes : 22 186 292

Ext-NoRepCondTime.....type reference INTEGER
DEFINED in MAP-MS-DataTypes : 1031
USED in MAP-MS-DataTypes : 997 1877

ext-ProtocolId.....identifier of Ext-ProtocolId
DEFINED in MAP-CommonDataTypes : 222

Ext-ProtocolId.....type reference ENUMERATED
DEFINED in MAP-CommonDataTypes : 229
USED in MAP-CommonDataTypes : 222

ext-QoS-Subscribed.....identifier of [0] Ext-QoS-Subscribed
DEFINED in MAP-MS-DataTypes : 753

Ext-QoS-Subscribed.....type reference OCTET STRING
DEFINED in MAP-MS-DataTypes : 839
USED in MAP-MS-DataTypes : 58 753

Ext-SS-Data.....type reference SEQUENCE
DEFINED in MAP-MS-DataTypes : 1109
USED in MAP-MS-DataTypes : 975

Ext-SS-Info.....type reference CHOICE
DEFINED in MAP-MS-DataTypes : 971
USED in MAP-MS-DataTypes : 969

Ext-SS-InfoFor-CSE.....type reference CHOICE
DEFINED in MAP-MS-DataTypes : 1937
USED in MAP-MS-DataTypes : 1866

Ext-SS-InfoList.....type reference SEQUENCE OF
DEFINED in MAP-MS-DataTypes : 968
USED in MAP-MS-DataTypes : 892

Ext-SS-Status.....type reference OCTET STRING
DEFINED in MAP-CommonDataTypes : 471
USED in MAP-MS-DataTypes : 181 990 1049 1111 1124 1180 1874 1885
USED in MAP-CommonDataTypes : 55 458

ext-Teleservice.....identifier of [3] Ext-TeleserviceCode
DEFINED in MAP-CommonDataTypes : 433

Ext-TeleserviceCode.....type reference OCTET STRING
DEFINED in MAP-TS-Code : 20
USED in MAP-MS-DataTypes : 155 914
USED in MAP-CommonDataTypes : 65 433
USED in MAP-GR-DataTypes : 31 50

facilityNotSupParam.....identifier of FacilityNotSupParam
DEFINED in MAP-Errors : 183

FacilityNotSupParam.....type reference SEQUENCE
DEFINED in MAP-ER-DataTypes : 189
USED in MAP-Errors : 113 183
USED in MAP-ER-DataTypes : 23

facilityNotSupported.....value reference FacilityNotSupported, CHOICE VALUE
DEFINED in MAP-Protocol : 331

FacilityNotSupported.....type reference ERROR
DEFINED in MAP-Errors : 181
USED in MAP-Protocol : 119 331
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

facsimileGroup4.....value reference TeleserviceCode, '01100011'B
DEFINED in MAP-TS-Code : 51

failure.....identifier of Named Number, 1
DEFINED in MAP-CH-DataTypes : 373

failureCause.....identifier of FailureCause
DEFINED in MAP-MS-DataTypes : 366

FailureCause.....type reference ENUMERATED
DEFINED in MAP-MS-DataTypes : 374
USED in MAP-MS-DataTypes : 366

failureReport.....value reference FailureReport, CHOICE VALUE
DEFINED in MAP-Protocol : 310

FailureReport.....type reference OPERATION
DEFINED in MAP-MobileServiceOpera : 449
USED in MAP-Protocol : 35 310
USED in MAP-MobileServiceOpera : 64

failureReportArg.....identifier of FailureReportArg
DEFINED in MAP-MobileServiceOpera : 451

FailureReportArg.....type reference SEQUENCE
DEFINED in MAP-MS-DataTypes : 1601
USED in MAP-MobileServiceOpera : 155 451
USED in MAP-MS-DataTypes : 120

failureReportRes.....identifier of FailureReportRes
DEFINED in MAP-MobileServiceOpera : 453

FailureReportRes.....type reference SEQUENCE
DEFINED in MAP-MS-DataTypes : 1608
USED in MAP-MobileServiceOpera : 156 453
USED in MAP-MS-DataTypes : 121

foreignNumberPortedToForeignNetwork.....identifier of Named Number, 2
DEFINED in MAP-CH-DataTypes : 171

forwardAccessSignalling.....value reference ForwardAccessSignalling, CHOICE VALUE
DEFINED in MAP-Protocol : 192

ForwardAccessSignalling.....type reference OPERATION
DEFINED in MAP-MobileServiceOpera : 333
USED in MAP-Protocol : 20 192
USED in MAP-MobileServiceOpera : 41

forwardAccessSignalling-Arg.....identifier of ForwardAccessSignalling-Arg
DEFINED in MAP-MobileServiceOpera : 335

ForwardAccessSignalling-Arg.....type reference [3] SEQUENCE
DEFINED in MAP-MS-DataTypes : 407
USED in MAP-MobileServiceOpera : 125 335
USED in MAP-MS-DataTypes : 32

forwardCheckSS-Indication.....value reference ForwardCheckSS-Indication, CHOICE VALUE
DEFINED in MAP-Protocol : 217

ForwardCheckSS-Indication.....type reference OPERATION
DEFINED in MAP-MobileServiceOpera : 420
USED in MAP-Protocol : 28 217
USED in MAP-MobileServiceOpera : 57

forwarded.....identifier of Named Number, 0
DEFINED in MAP-MS-DataTypes : 1428

forwardedToNumber.....identifier of [5] ISDN-AddressString
DEFINED in MAP-MS-DataTypes : 991

forwardedToNumber.....identifier of [3] AddressString
DEFINED in MAP-MS-DataTypes : 1875

forwardedToNumber.....identifier of [5] ISDN-AddressString
DEFINED in MAP-CH-DataTypes : 188

forwardedToNumber.....identifier of [4] AddressString
DEFINED in MAP-SS-DataTypes : 75

forwardedToNumber.....identifier of [5] ISDN-AddressString
DEFINED in MAP-SS-DataTypes : 102

forwardedToSubaddress.....identifier of [8] ISDN-SubaddressString
DEFINED in MAP-MS-DataTypes : 995

forwardedToSubaddress.....identifier of [4] ISDN-SubaddressString
DEFINED in MAP-MS-DataTypes : 1876

forwardedToSubaddress.....identifier of [4] ISDN-SubaddressString
DEFINED in MAP-CH-DataTypes : 192

forwardedToSubaddress.....identifier of [6] ISDN-SubaddressString
DEFINED in MAP-SS-DataTypes : 76

forwardedToSubaddress.....identifier of [8] ISDN-SubaddressString
DEFINED in MAP-SS-DataTypes : 103

forwardGroupCallSignalling.....value reference ForwardGroupCallSignalling, CHOICE VALUE
DEFINED in MAP-Protocol : 297

ForwardGroupCallSignalling.....type reference OPERATION
DEFINED in MAP-Group-Call-Operati : 67
USED in MAP-Protocol : 101 297
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 : 123

forwardingData.....identifier of ForwardingData
DEFINED in MAP-CH-DataTypes : 185

ForwardingData.....type reference SEQUENCE
DEFINED in MAP-CH-DataTypes : 187

USED in MAP-CH-DataTypes : 185 227 267

forwardingData.....identifier of [2] ForwardingData
DEFINED in MAP-CH-DataTypes : 227

forwardingData.....identifier of ForwardingData
DEFINED in MAP-CH-DataTypes : 267

forwardingFailed.....value reference ForwardingFailed, CHOICE VALUE
DEFINED in MAP-Protocol : 379

ForwardingFailed.....type reference ERROR
DEFINED in MAP-Errors : 306
USED in MAP-Protocol : 140 379
USED in MAP-CallHandlingOperat : 45 125
USED in MAP-Errors : 51

forwardingFailedParam.....identifier of ForwardingFailedParam
DEFINED in MAP-Errors : 308

ForwardingFailedParam.....type reference SEQUENCE
DEFINED in MAP-ER-DataTypes : 275
USED in MAP-Errors : 130 308
USED in MAP-ER-DataTypes : 38

ForwardingFeature.....type reference SEQUENCE
DEFINED in MAP-SS-DataTypes : 99
USED in MAP-SS-DataTypes : 97

forwardingFeatureList.....identifier of Ext-ForwFeatureList
DEFINED in MAP-MS-DataTypes : 981

forwardingFeatureList.....identifier of Ext-ForwFeatureList
DEFINED in MAP-MS-DataTypes : 1816

forwardingFeatureList.....identifier of [1] Ext-ForwFeatureList
DEFINED in MAP-MS-DataTypes : 1944

forwardingFeatureList.....identifier of ForwardingFeatureList
DEFINED in MAP-SS-DataTypes : 92

ForwardingFeatureList.....type reference SEQUENCE OF
DEFINED in MAP-SS-DataTypes : 95
USED in MAP-SS-DataTypes : 92 218

forwardingFeatureList.....identifier of [3] ForwardingFeatureList
DEFINED in MAP-SS-DataTypes : 218

forwardingInfo.....identifier of [0] Ext-ForwInfo
DEFINED in MAP-MS-DataTypes : 972

forwardingInfo.....identifier of [0] ForwardingInfo
DEFINED in MAP-SS-DataTypes : 86

ForwardingInfo.....type reference SEQUENCE
DEFINED in MAP-SS-DataTypes : 90
USED in MAP-SS-DataTypes : 86

forwardingInfoFor-CSE.....identifier of [0] Ext-ForwardingInfoFor-CSE
DEFINED in MAP-MS-DataTypes : 1908

forwardingInfoFor-CSE.....identifier of [0] Ext-ForwardingInfoFor-CSE
DEFINED in MAP-MS-DataTypes : 1938

forwardingInterrogationRequired.....identifier of [4] NULL
DEFINED in MAP-CH-DataTypes : 156

forwardingOptions.....identifier of [6] Ext-ForwOptions
DEFINED in MAP-MS-DataTypes : 996

forwardingOptions.....identifier of [6] ForwardingOptions
DEFINED in MAP-CH-DataTypes : 193

forwardingOptions.....identifier of [6] ForwardingOptions
DEFINED in MAP-SS-DataTypes : 104

ForwardingOptions.....type reference OCTET STRING
DEFINED in MAP-SS-DataTypes : 124
USED in MAP-CH-DataTypes : 56 193
USED in MAP-SS-DataTypes : 31 104

forwardingReason.....identifier of [8] ForwardingReason

DEFINED in MAP-CH-DataTypes : 103

ForwardingReason.....type reference ENUMERATED
DEFINED in MAP-CH-DataTypes : 129
USED in MAP-CH-DataTypes : 103

forwardingViolation.....value reference ForwardingViolation, CHOICE VALUE
DEFINED in MAP-Protocol : 381

ForwardingViolation.....type reference ERROR
DEFINED in MAP-Errors : 301
USED in MAP-Protocol : 139 381
USED in MAP-CallHandlingOperat : 44 101
USED in MAP-Errors : 50

forwardingViolationParam.....identifier of ForwardingViolationParam
DEFINED in MAP-Errors : 303

ForwardingViolationParam.....type reference SEQUENCE
DEFINED in MAP-ER-DataTypes : 271
USED in MAP-Errors : 129 303
USED in MAP-ER-DataTypes : 37

freezeP-TMSI.....identifier of [1] NULL
DEFINED in MAP-MS-DataTypes : 276

freezeTMSI.....identifier of [0] NULL
DEFINED in MAP-MS-DataTypes : 275

frozen.....identifier of Named Number, 5
DEFINED in MAP-SS-DataTypes : 291

FTN-AddressString.....type reference AddressString
DEFINED in MAP-CommonDataTypes : 149
USED in MAP-MS-DataTypes : 164 1000
USED in MAP-CommonDataTypes : 19
USED in MAP-CH-DataTypes : 65 196
USED in MAP-SS-DataTypes : 47 107

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 : 190
USED in MAP-SS-DataTypes : 219

genericServiceInfo.....identifier of [4] GenericServiceInfo
DEFINED in MAP-SS-DataTypes : 219

geodeticInformation.....identifier of [7] GeodeticInformation
DEFINED in MAP-MS-DataTypes : 1715

GeodeticInformation.....type reference OCTET STRING
DEFINED in MAP-MS-DataTypes : 1733
USED in MAP-MS-DataTypes : 1715

geographicalInformation.....identifier of [0] GeographicalInformation
DEFINED in MAP-MS-DataTypes : 1707

GeographicalInformation.....type reference OCTET STRING
DEFINED in MAP-MS-DataTypes : 1723
USED in MAP-MS-DataTypes : 92 1707

getPassword.....value reference GetPassword, CHOICE VALUE
DEFINED in MAP-Protocol : 255

GetPassword.....type reference OPERATION

DEFINED in MAP-SupplementaryServi : 234
USED in MAP-Protocol : 78 255
USED in MAP-SupplementaryServi : 22 232

ggsn-Address.....identifier of [1] GSN-Address
DEFINED in MAP-MS-DataTypes : 1587

ggsn-Address.....identifier of [1] GSN-Address
DEFINED in MAP-MS-DataTypes : 1594

ggsn-Address.....identifier of [2] GSN-Address
DEFINED in MAP-MS-DataTypes : 1604

ggsn-Address.....identifier of [0] GSN-Address
DEFINED in MAP-MS-DataTypes : 1609

ggsn-Address.....identifier of [2] GSN-Address
DEFINED in MAP-MS-DataTypes : 1618

ggsn-Number.....identifier of [2] ISDN-AddressString
DEFINED in MAP-MS-DataTypes : 1588

ggsn-Number.....identifier of [1] ISDN-AddressString
DEFINED in MAP-MS-DataTypes : 1603

GlobalCellId.....type reference OCTET STRING
DEFINED in MAP-CommonDataTypes : 335
USED in MAP-MS-DataTypes : 171 464 565
USED in MAP-CommonDataTypes : 37

gmlc-List.....identifier of [0] GMLC-List
DEFINED in MAP-MS-DataTypes : 719

GMLC-List.....type reference SEQUENCE OF
DEFINED in MAP-MS-DataTypes : 724
USED in MAP-MS-DataTypes : 719

gmlc-List.....identifier of Named Number, 0
DEFINED in MAP-MS-DataTypes : 1157

gmlc-ListWithdraw.....identifier of [13] NULL
DEFINED in MAP-MS-DataTypes : 1224

gmlc-Restriction.....identifier of [0] GMLC-Restriction
DEFINED in MAP-MS-DataTypes : 1149

GMLC-Restriction.....type reference ENUMERATED
DEFINED in MAP-MS-DataTypes : 1156
USED in MAP-MS-DataTypes : 1149

gmscCamelSubscriptionInfo.....identifier of [0] GmscCamelSubscriptionInfo
DEFINED in MAP-CH-DataTypes : 268

GmscCamelSubscriptionInfo.....type reference SEQUENCE
DEFINED in MAP-CH-DataTypes : 272
USED in MAP-CH-DataTypes : 268

gmsc-Address.....identifier of [6] ISDN-AddressString
DEFINED in MAP-CH-DataTypes : 101

gmsc-Address.....identifier of [8] ISDN-AddressString
DEFINED in MAP-CH-DataTypes : 206

gprsConnectionSuspended.....identifier of NULL
DEFINED in MAP-ER-DataTypes : 310

GPRSDataList.....type reference SEQUENCE OF
DEFINED in MAP-MS-DataTypes : 739
USED in MAP-MS-DataTypes : 764

gprsDataList.....identifier of [1] GPRSDataList
DEFINED in MAP-MS-DataTypes : 764

gprsEnhancementsSupportIndicator.....identifier of [3] NULL
DEFINED in MAP-MS-DataTypes : 394

gprsNodeIndicator.....identifier of [5] NULL
DEFINED in MAP-SM-DataTypes : 90

gprsSubscriptionData.....identifier of [16] GPRSSubscriptionData
DEFINED in MAP-MS-DataTypes : 695

GPRSSubscriptionData.....type reference SEQUENCE
DEFINED in MAP-MS-DataTypes : 759
USED in MAP-MS-DataTypes : 695

gprsSubscriptionDataWithdraw.....identifier of [10] GPRSSubscriptionDataWithdraw
DEFINED in MAP-MS-DataTypes : 1221

GPRSSubscriptionDataWithdraw.....type reference CHOICE
DEFINED in MAP-MS-DataTypes : 1242
USED in MAP-MS-DataTypes : 1221

gprsSubscriptionUnknown.....identifier of Named Number, 1
DEFINED in MAP-ER-DataTypes : 204

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 : 793
USED in MAP-MS-DataTypes : 789

gprs-CamelTDPDataList.....identifier of [0] GPRS-CamelTDPDataList
DEFINED in MAP-MS-DataTypes : 775

GPRS-CamelTDPDataList.....type reference SEQUENCE OF
DEFINED in MAP-MS-DataTypes : 788
USED in MAP-MS-DataTypes : 775

gprs-CSI.....identifier of [0] GPRS-CSI
DEFINED in MAP-MS-DataTypes : 769

GPRS-CSI.....type reference SEQUENCE
DEFINED in MAP-MS-DataTypes : 774
USED in MAP-MS-DataTypes : 769 1847

gprs-csi.....identifier of Named Number, 7
DEFINED in MAP-MS-DataTypes : 1236

gprs-CSI.....identifier of Named Number, 4
DEFINED in MAP-MS-DataTypes : 1809

gprs-CSI.....identifier of [9] GPRS-CSI
DEFINED in MAP-MS-DataTypes : 1847

gprs-TriggerDetectionPoint.....identifier of [0] GPRS-TriggerDetectionPoint
DEFINED in MAP-MS-DataTypes : 794

GPRS-TriggerDetectionPoint.....type reference ENUMERATED
DEFINED in MAP-MS-DataTypes : 810
USED in MAP-MS-DataTypes : 794

greyListed.....identifier of Named Number, 2
DEFINED in MAP-MS-DataTypes : 683

groupCallNumber.....identifier of ISDN-AddressString
DEFINED in MAP-GR-DataTypes : 62

groupId.....identifier of GroupId
DEFINED in MAP-MS-DataTypes : 1660

groupid.....identifier of GroupId
DEFINED in MAP-MS-DataTypes : 1665

GroupId.....type reference TBCD-STRING
DEFINED in MAP-MS-DataTypes : 1670
USED in MAP-MS-DataTypes : 1660 1665

groupKey.....identifier of [1] Kc
DEFINED in MAP-GR-DataTypes : 55

groupKeyNumber.....identifier of [0] GroupKeyNumber
DEFINED in MAP-GR-DataTypes : 54

GroupKeyNumber.....type reference INTEGER
DEFINED in MAP-GR-DataTypes : 93
USED in MAP-GR-DataTypes : 54

gsmSCF-Address.....identifier of [2] ISDN-AddressString

DEFINED in MAP-MS-DataTypes : 796

gsmSCF-Address.....identifier of ISDN-AddressString
DEFINED in MAP-MS-DataTypes : 1302

gsmSCF-Address.....identifier of ISDN-AddressString
DEFINED in MAP-MS-DataTypes : 1319

gsmSCF-Address.....identifier of [0] ISDN-AddressString
DEFINED in MAP-MS-DataTypes : 1359

gsmSCF-Address.....identifier of [2] ISDN-AddressString
DEFINED in MAP-MS-DataTypes : 1493

gsmSCF-Address.....identifier of [0] ISDN-AddressString
DEFINED in MAP-MS-DataTypes : 1518

gsmSCF-Address.....identifier of [0] ISDN-AddressString
DEFINED in MAP-MS-DataTypes : 1567

gsmSCF-Address.....identifier of [3] ISDN-AddressString
DEFINED in MAP-MS-DataTypes : 1765

gsmSCF-Address.....identifier of [2] ISDN-AddressString
DEFINED in MAP-MS-DataTypes : 1780

gsmSCF-Address.....identifier of [1] ISDN-AddressString
DEFINED in MAP-MS-DataTypes : 1857

gsm-0408.....identifier of Named Number, 1
DEFINED in MAP-CommonDataTypes : 215

gsm-0806.....identifier of Named Number, 2
DEFINED in MAP-CommonDataTypes : 216

gsm-0806.....identifier of Named Number, 1
DEFINED in MAP-CommonDataTypes : 256

gsm-BearerCapability.....identifier of [5] ExternalSignalInfo
DEFINED in MAP-CH-DataTypes : 203

gsm-BearerCapability.....identifier of [0] ExternalSignalInfo
DEFINED in MAP-CH-DataTypes : 282

gsm-BSSMAP.....identifier of Named Number, 3
DEFINED in MAP-CommonDataTypes : 217

gsm-SecurityContextData.....identifier of [0] GSM-SecurityContextData
DEFINED in MAP-MS-DataTypes : 328

GSM-SecurityContextData.....type reference SEQUENCE
DEFINED in MAP-MS-DataTypes : 331
USED in MAP-MS-DataTypes : 328

GSN-Address.....type reference OCTET STRING
DEFINED in MAP-MS-DataTypes : 397
USED in MAP-MS-DataTypes : 29 383 1587 1593 1594 1604 1609 1617 1618

guidanceInfo.....identifier of GuidanceInfo
DEFINED in MAP-SupplementaryServi : 236

GuidanceInfo.....type reference ENUMERATED
DEFINED in MAP-SS-DataTypes : 248
USED in MAP-SupplementaryServi : 67 236
USED in MAP-SS-DataTypes : 25

handoverNumber.....identifier of [0] ISDN-AddressString
DEFINED in MAP-MS-DataTypes : 519

handoverNumber.....identifier of ISDN-AddressString
DEFINED in MAP-MS-DataTypes : 619

highLayerCompatibility.....identifier of [6] ExternalSignalInfo
DEFINED in MAP-CH-DataTypes : 288

hlr.....identifier of Named Number, 1
DEFINED in MAP-CommonDataTypes : 350

HLR-Id.....type reference IMSI
DEFINED in MAP-CommonDataTypes : 324
USED in MAP-CommonDataTypes : 329

hlr-List.....identifier of HLR- List

DEFINED in MAP-MS-DataTypes : 1631

HLR-List.....type reference SEQUENCE OF
DEFINED in MAP-CommonDataTypes : 328
USED in MAP-MS-DataTypes : 168 1631
USED in MAP-CommonDataTypes : 35

hlr-Number.....identifier of ISDN-AddressString
DEFINED in MAP-MS-DataTypes : 244

hlr-Number.....identifier of ISDN-AddressString
DEFINED in MAP-MS-DataTypes : 401

hlr-Number.....identifier of ISDN-AddressString
DEFINED in MAP-MS-DataTypes : 1630

hlr-Number.....identifier of ISDN-AddressString
DEFINED in MAP-MS-DataTypes : 1642

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 : 1158

horizontal-accuracy.....identifier of [0] Horizontal-Accuracy
DEFINED in MAP-LCS-DataTypes : 142

Horizontal-Accuracy.....type reference OCTET STRING
DEFINED in MAP-LCS-DataTypes : 148
USED in MAP-LCS-DataTypes : 20 142

ho-NumberNotRequired.....identifier of NULL
DEFINED in MAP-MS-DataTypes : 465

identity.....identifier of Identity
DEFINED in MAP-MS-DataTypes : 250

Identity.....type reference CHOICE
DEFINED in MAP-CommonDataTypes : 297
USED in MAP-MS-DataTypes : 170 250
USED in MAP-CommonDataTypes : 32

ik.....identifier of IK
DEFINED in MAP-MS-DataTypes : 323

ik.....identifier of IK
DEFINED in MAP-MS-DataTypes : 338

IK.....type reference OCTET STRING
DEFINED in MAP-MS-DataTypes : 352
USED in MAP-MS-DataTypes : 323 338

illegalEquipment.....value reference IllegalEquipment, CHOICE VALUE
DEFINED in MAP-Protocol : 349

IllegalEquipment.....type reference ERROR
DEFINED in MAP-Errors : 233
USED in MAP-Protocol : 127 349
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 : 235

IllegalEquipmentParam.....type reference SEQUENCE
DEFINED in MAP-ER-DataTypes : 223
USED in MAP-Errors : 119 235
USED in MAP-ER-DataTypes : 29

illegalSS-Operation.....value reference IllegalSS-Operation, CHOICE VALUE
DEFINED in MAP-Protocol : 401

IllegalSS-Operation.....type reference ERROR
DEFINED in MAP-Errors : 347
USED in MAP-Protocol : 143 401
USED in MAP-MobileServiceOpera : 100 271 288
USED in MAP-SupplementaryServi : 40 100 117 134 154 172 261 278
USED in MAP-Errors : 63

illegalSS-OperationParam.....identifier of IllegalSS-OperationParam
DEFINED in MAP-Errors : 349

IllegalSS-OperationParam.....type reference SEQUENCE
DEFINED in MAP-ER-DataTypes : 291
USED in MAP-Errors : 148 349
USED in MAP-ER-DataTypes : 56

illegalSubscriber.....value reference IllegalSubscriber, CHOICE VALUE
DEFINED in MAP-Protocol : 348

IllegalSubscriber.....type reference ERROR
DEFINED in MAP-Errors : 227
USED in MAP-Protocol : 126 348
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 : 229

IllegalSubscriberParam.....type reference SEQUENCE
DEFINED in MAP-ER-DataTypes : 219
USED in MAP-Errors : 118 229
USED in MAP-ER-DataTypes : 28

imei.....identifier of IMEI
DEFINED in MAP-MobileServiceOpera : 382

IMEI.....type reference TBCD-STRING
DEFINED in MAP-CommonDataTypes : 317
USED in MAP-MobileServiceOpera : 167 382
USED in MAP-CommonDataTypes : 34
USED in MAP-LCS-DataTypes : 30 81 289

imei.....identifier of [5] IMEI
DEFINED in MAP-LCS-DataTypes : 81

imei.....identifier of [2] IMEI
DEFINED in MAP-LCS-DataTypes : 289

immediateResponsePreferred.....identifier of [1] NULL
DEFINED in MAP-MS-DataTypes : 660

imsi.....identifier of IMSI
DEFINED in MAP-OperationAndMainte : 81

imsi.....identifier of IMSI
DEFINED in MAP-MS-DataTypes : 208

imsi.....identifier of IMSI
DEFINED in MAP-MS-DataTypes : 268

imsi.....identifier of IMSI
DEFINED in MAP-MS-DataTypes : 292

imsi.....identifier of IMSI
DEFINED in MAP-MS-DataTypes : 365

imsi.....identifier of IMSI
DEFINED in MAP-MS-DataTypes : 381

imsi.....identifier of [4] IMSI
DEFINED in MAP-MS-DataTypes : 469

imsi.....identifier of [0] IMSI
DEFINED in MAP-MS-DataTypes : 657

imsi.....identifier of [0] IMSI
DEFINED in MAP-MS-DataTypes : 689

imsi.....identifier of [0] IMSI
DEFINED in MAP-MS-DataTypes : 1209

imsi.....identifier of [0] IMSI
DEFINED in MAP-MS-DataTypes : 1586

imsi.....identifier of [0] IMSI
DEFINED in MAP-MS-DataTypes : 1602

imsi.....identifier of [0] IMSI

DEFINED in MAP-MS-DataTypes : 1616

imsi.....identifier of IMSI
DEFINED in MAP-MS-DataTypes : 1635

imsi.....identifier of [0] IMSI
DEFINED in MAP-MS-DataTypes : 1679

imsi.....identifier of IMSI
DEFINED in MAP-MS-DataTypes : 1906

imsi.....identifier of [1] IMSI
DEFINED in MAP-MS-DataTypes : 1926

IMSI.....type reference TBCD-STRING
DEFINED in MAP-CommonDataTypes : 294
USED in MAP-OperationAndMainte : 43 81
USED in MAP-MS-DataTypes : 166 208 268 292 365 381 469 657 689
1209 1586 1602 1616 1635 1679 1906 1926
USED in MAP-CommonDataTypes : 30 298 302 314 324 371
USED in MAP-OM-DataTypes : 22 37 55
USED in MAP-CH-DataTypes : 68 146 199 228 316 346 386 417 429
USED in MAP-SS-DataTypes : 48 270
USED in MAP-SM-DataTypes : 34 80 112 133 197
USED in MAP-GR-DataTypes : 24 67 76
USED in MAP-LCS-DataTypes : 31 78 288

imsi.....identifier of IMSI
DEFINED in MAP-CommonDataTypes : 298

imsi.....identifier of IMSI
DEFINED in MAP-CommonDataTypes : 302

imsi.....identifier of [0] IMSI
DEFINED in MAP-CommonDataTypes : 314

imsi.....identifier of [0] IMSI
DEFINED in MAP-CommonDataTypes : 371

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 : 146

imsi.....identifier of [0] IMSI
DEFINED in MAP-CH-DataTypes : 199

imsi.....identifier of [3] IMSI
DEFINED in MAP-CH-DataTypes : 228

imsi.....identifier of [0] IMSI
DEFINED in MAP-CH-DataTypes : 316

imsi.....identifier of [0] IMSI
DEFINED in MAP-CH-DataTypes : 346

imsi.....identifier of [0] IMSI
DEFINED in MAP-CH-DataTypes : 386

imsi.....identifier of [0] IMSI
DEFINED in MAP-CH-DataTypes : 417

imsi.....identifier of [0] IMSI
DEFINED in MAP-CH-DataTypes : 429

imsi.....identifier of [0] IMSI
DEFINED in MAP-SS-DataTypes : 270

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 : 197

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 : 78

imsi.....identifier of [1] IMSI
DEFINED in MAP-LCS-DataTypes : 288

imsiDetach.....identifier of Named Number, 0
DEFINED in MAP-ER-DataTypes : 249

imsiDetached.....identifier of Named Number, 1
DEFINED in MAP-MS-DataTypes : 1756

imsiUnknown.....identifier of Named Number, 0
DEFINED in MAP-ER-DataTypes : 203

imsi-WithLMSI.....identifier of IMSI-WithLMSI
DEFINED in MAP-CommonDataTypes : 299

IMSI-WithLMSI.....type reference SEQUENCE
DEFINED in MAP-CommonDataTypes : 301
USED in MAP-CommonDataTypes : 299

incomingCallsBarredWithinCUG.....identifier of Named Number, 0
DEFINED in MAP-ER-DataTypes : 123

incompatibleTerminal.....value reference IncompatibleTerminal, CHOICE VALUE
DEFINED in MAP-Protocol : 332

IncompatibleTerminal.....type reference ERROR
DEFINED in MAP-Errors : 187
USED in MAP-Protocol : 161 332
USED in MAP-CallHandlingOperat : 48 187
USED in MAP-Errors : 18

incompatibleTerminalParam.....identifier of IncompatibleTerminalParam
DEFINED in MAP-Errors : 189

IncompatibleTerminalParam.....type reference SEQUENCE
DEFINED in MAP-ER-DataTypes : 326
USED in MAP-Errors : 138 189
USED in MAP-ER-DataTypes : 46

inconsistentMeasurementData.....identifier of Named Number, 3
DEFINED in MAP-ER-DataTypes : 364

incorrectTransactionPortion.....identifier of Named Number, 3
DEFINED in TCAPMessages : 106

informationNotAvailable.....value reference InformationNotAvailable, CHOICE VALUE
DEFINED in MAP-Protocol : 392

InformationNotAvailable.....type reference ERROR
DEFINED in MAP-Errors : 339
USED in MAP-Protocol : 168 392
USED in MAP-MobileServiceOpera : 105 273 292
USED in MAP-Errors : 60

informationNotAvailableParam.....identifier of InformationNotAvailableParam
DEFINED in MAP-Errors : 341

InformationNotAvailableParam.....type reference SEQUENCE
DEFINED in MAP-ER-DataTypes : 303
USED in MAP-Errors : 151 341
USED in MAP-ER-DataTypes : 59

informPreviousNetworkEntity.....identifier of [11] NULL
DEFINED in MAP-MS-DataTypes : 216

informPreviousNetworkEntity.....identifier of [1] NULL
DEFINED in MAP-MS-DataTypes : 387

informServiceCentre.....value reference InformServiceCentre, CHOICE VALUE
DEFINED in MAP-Protocol : 266

InformServiceCentre.....type reference OPERATION
DEFINED in MAP-ShortMessageServic : 132
USED in MAP-Protocol : 92 266
USED in MAP-ShortMessageServic : 18

informServiceCentreArg.....identifier of InformServiceCentreArg
DEFINED in MAP-ShortMessageServic : 134

InformServiceCentreArg.....type reference SEQUENCE
DEFINED in MAP-SM-DataTypes : 182
USED in MAP-ShortMessageServic : 54 134
USED in MAP-SM-DataTypes : 23

inhibiting.....identifier of Named Number, 0
DEFINED in MAP-MS-DataTypes : 1432

initialLocation.....identifier of Named Number, 2
DEFINED in MAP-LCS-DataTypes : 97

initiatingRelease.....identifier of Named Number, 4
DEFINED in TCAPMessages : 187

insertSubscriberData.....value reference InsertSubscriberData, CHOICE VALUE
DEFINED in MAP-Protocol : 210

InsertSubscriberData.....type reference OPERATION
DEFINED in MAP-MobileServiceOpera : 392
USED in MAP-Protocol : 25 210
USED in MAP-MobileServiceOpera : 52

insertSubscriberDataArg.....identifier of InsertSubscriberDataArg
DEFINED in MAP-MobileServiceOpera : 394

InsertSubscriberDataArg.....type reference SEQUENCE
DEFINED in MAP-MS-DataTypes : 688
USED in MAP-MobileServiceOpera : 136 394
USED in MAP-MS-DataTypes : 53

insertSubscriberDataRes.....identifier of InsertSubscriberDataRes
DEFINED in MAP-MobileServiceOpera : 396

InsertSubscriberDataRes.....type reference SEQUENCE
DEFINED in MAP-MS-DataTypes : 1192
USED in MAP-MobileServiceOpera : 137 396
USED in MAP-MS-DataTypes : 54

insufficientMeasurementData.....identifier of Named Number, 2
DEFINED in MAP-ER-DataTypes : 363

insufficientResources.....identifier of Named Number, 1
DEFINED in MAP-ER-DataTypes : 362

integrityProtectionAlgorithm.....identifier of [0] ChosenIntegrityProtectionAlgorithm
DEFINED in MAP-MS-DataTypes : 530

integrityProtectionAlgorithms.....identifier of [0] PermittedIntegrityProtectionAlgorithms
DEFINED in MAP-MS-DataTypes : 428

integrityProtectionInfo.....identifier of [0] IntegrityProtectionInformation
DEFINED in MAP-MS-DataTypes : 409

integrityProtectionInfo.....identifier of [5] IntegrityProtectionInformation
DEFINED in MAP-MS-DataTypes : 470

IntegrityProtectionInformation.....type reference OCTET STRING
DEFINED in MAP-MS-DataTypes : 634
USED in MAP-MS-DataTypes : 409 470

interCUG-Restrictions.....identifier of InterCUG-Restrictions
DEFINED in MAP-MS-DataTypes : 1093

InterCUG-Restrictions.....type reference OCTET STRING
DEFINED in MAP-MS-DataTypes : 1097
USED in MAP-MS-DataTypes : 78 1093

internationalECT-Barred.....identifier of Named Number, 11
DEFINED in MAP-MS-DataTypes : 936

internationalOGCallsBarred.....identifier of Named Number, 1
DEFINED in MAP-MS-DataTypes : 926

internationalOGCallsNotToHPLMN-CountryBaidentifier of Named Number, 2

DEFINED in MAP-MS-DataTypes : 927

interrogateSS.....value reference InterrogateSS, CHOICE VALUE
DEFINED in MAP-Protocol : 249

InterrogateSS.....type reference OPERATION
DEFINED in MAP-SupplementaryServi : 160
USED in MAP-Protocol : 73 249
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 : 215
USED in MAP-SupplementaryServi : 63 164
USED in MAP-SS-DataTypes : 19

interrogationType.....identifier of [3] InterrogationType
DEFINED in MAP-CH-DataTypes : 98

InterrogationType.....type reference ENUMERATED
DEFINED in MAP-CH-DataTypes : 121
USED in MAP-CH-DataTypes : 98

interzonalECT-Barred.....identifier of Named Number, 12
DEFINED in MAP-MS-DataTypes : 937

interzonalOGCallsAndInternationalOGCallsidentifier of Named Number, 8
DEFINED in MAP-MS-DataTypes : 930

interzonalOGCallsBarred.....identifier of Named Number, 6
DEFINED in MAP-MS-DataTypes : 928

interzonalOGCallsNotToHPLMN-CountryBarreidentifier of Named Number, 7
DEFINED in MAP-MS-DataTypes : 929

intraCUG-Options.....identifier of IntraCUG-Options
DEFINED in MAP-MS-DataTypes : 1065

IntraCUG-Options.....type reference ENUMERATED
DEFINED in MAP-MS-DataTypes : 1075
USED in MAP-MS-DataTypes : 79 1065

invalidFormat.....identifier of Named Number, 1
DEFINED in MAP-ER-DataTypes : 136

invalidSME-Address.....identifier of Named Number, 5
DEFINED in MAP-ER-DataTypes : 146

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 : 143

USED in MAP-OperationAndMainte : 42 79
USED in MAP-MS-DataTypes : 162 210 211 244 269 270 382 401 519
566 619 725 796 883 991 1300 1302 1319
1359 1409 1493 1518 1567 1588 1603 1630 1642
1708 1714 1765 1780 1857 1907 1927
USED in MAP-CommonDataTypes : 17 372
USED in MAP-CH-DataTypes : 63 95 101 157 163 184 188 200 201
206 220 233 285 300 389
USED in MAP-SS-DataTypes : 45 102 208 226 271 281 312
USED in MAP-SM-DataTypes : 32 53 86 98 99 139 144 172 178
183
USED in MAP-GR-DataTypes : 23 62
USED in MAP-LCS-DataTypes : 29 56 68 75 79 287 290 291 314

isdn-BearerCapability.....identifier of [1] ExternalSignalInfo
DEFINED in MAP-CH-DataTypes : 283

ISDN-SubaddressString.....type reference OCTET STRING
DEFINED in MAP-CommonDataTypes : 155
USED in MAP-MS-DataTypes : 163 995 1876
USED in MAP-CommonDataTypes : 20
USED in MAP-CH-DataTypes : 64 192
USED in MAP-SS-DataTypes : 46 76 103 209

istAlert.....value reference IST-Alert, CHOICE VALUE
DEFINED in MAP-Protocol : 239

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 : 702

istAlertTimer.....identifier of [14] IST-AlertTimerValue
DEFINED in MAP-CH-DataTypes : 165

istAlertTimer.....identifier of [0] IST-AlertTimerValue
DEFINED in MAP-CH-DataTypes : 422

istCommand.....value reference IST-Command, CHOICE VALUE
DEFINED in MAP-Protocol : 240

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 : 237

istInformationWithdraw.....identifier of [14] NULL
DEFINED in MAP-MS-DataTypes : 1225

istInformationWithdraw.....identifier of [1] NULL
DEFINED in MAP-CH-DataTypes : 423

istSupportIndicator.....identifier of [1] IST-SupportIndicator
DEFINED in MAP-MS-DataTypes : 223

istSupportIndicator.....identifier of [18] IST-SupportIndicator
DEFINED in MAP-CH-DataTypes : 114

IST-Alert.....type reference OPERATION
DEFINED in MAP-CallHandlingOperat : 192
USED in MAP-Protocol : 62 239
USED in MAP-CallHandlingOperat : 21

IST-AlertArg.....type reference SEQUENCE
DEFINED in MAP-CH-DataTypes : 416
USED in MAP-CallHandlingOperat : 70 194
USED in MAP-CH-DataTypes : 33

IST-AlertRes.....type reference SEQUENCE
DEFINED in MAP-CH-DataTypes : 421
USED in MAP-CallHandlingOperat : 71 196
USED in MAP-CH-DataTypes : 34

IST-AlertTimerValue.....type reference INTEGER

DEFINED in MAP-MS-DataTypes : 716
USED in MAP-MS-DataTypes : 82 702
USED in MAP-CH-DataTypes : 48 165 422

IST-Command.....type reference OPERATION
DEFINED in MAP-CallHandlingOperat : 205
USED in MAP-Protocol : 63 240
USED in MAP-CallHandlingOperat : 22

IST-CommandArg.....type reference SEQUENCE
DEFINED in MAP-CH-DataTypes : 428
USED in MAP-CallHandlingOperat : 72 207
USED in MAP-CH-DataTypes : 35

IST-CommandRes.....type reference SEQUENCE
DEFINED in MAP-CH-DataTypes : 433
USED in MAP-CallHandlingOperat : 73 209
USED in MAP-CH-DataTypes : 36

IST-SupportIndicator.....type reference ENUMERATED
DEFINED in MAP-MS-DataTypes : 235
USED in MAP-MS-DataTypes : 26 223
USED in MAP-CH-DataTypes : 47 114

kc.....identifier of Kc
DEFINED in MAP-MS-DataTypes : 316

kc.....identifier of Kc
DEFINED in MAP-MS-DataTypes : 332

Kc.....type reference OCTET STRING
DEFINED in MAP-MS-DataTypes : 346
USED in MAP-MS-DataTypes : 50 316 332
USED in MAP-GR-DataTypes : 36 55

keepCCBS-CallIndicator.....identifier of [1] NULL
DEFINED in MAP-CH-DataTypes : 179

keyStatus.....identifier of [2] KeyStatus
DEFINED in MAP-MS-DataTypes : 411

KeyStatus.....type reference ENUMERATED
DEFINED in MAP-MS-DataTypes : 455
USED in MAP-MS-DataTypes : 411

ksi.....identifier of KSI
DEFINED in MAP-MS-DataTypes : 339

KSI.....type reference OCTET STRING
DEFINED in MAP-MS-DataTypes : 361
USED in MAP-MS-DataTypes : 339

laiFixedLength.....identifier of [1] LAIFixedLength
DEFINED in MAP-CommonDataTypes : 394

LAIFixedLength.....type reference OCTET STRING
DEFINED in MAP-CommonDataTypes : 412
USED in MAP-CommonDataTypes : 394

lawfulInterceptServices.....identifier of Named Number, 3
DEFINED in MAP-LCS-DataTypes : 115

lcsClientDialedByMS.....identifier of [2] AddressString
DEFINED in MAP-LCS-DataTypes : 106

LCSClientExternalID.....type reference SEQUENCE
DEFINED in MAP-CommonDataTypes : 375
USED in MAP-MS-DataTypes : 179 1148
USED in MAP-CommonDataTypes : 59
USED in MAP-LCS-DataTypes : 35 105

lcsClientExternalID.....identifier of [1] LCSClientExternalID
DEFINED in MAP-LCS-DataTypes : 105

LCSClientInternalID.....type reference ENUMERATED
DEFINED in MAP-CommonDataTypes : 380
USED in MAP-MS-DataTypes : 180 1143
USED in MAP-CommonDataTypes : 60
USED in MAP-LCS-DataTypes : 36 107

lcsClientInternalID.....identifier of [3] LCSClientInternalID

DEFINED in MAP-LCS-DataTypes : 107

lcsClientName.....identifier of [4] LCSClientName
DEFINED in MAP-LCS-DataTypes : 108

LCSClientName.....type reference SEQUENCE
DEFINED in MAP-LCS-DataTypes : 122
USED in MAP-LCS-DataTypes : 18 108

lcsClientType.....identifier of [0] LCSClientType
DEFINED in MAP-LCS-DataTypes : 104

LCSClientType.....type reference ENUMERATED
DEFINED in MAP-LCS-DataTypes : 111
USED in MAP-LCS-DataTypes : 104

lcsInformation.....identifier of [22] LCSInformation
DEFINED in MAP-MS-DataTypes : 701

LCSInformation.....type reference SEQUENCE
DEFINED in MAP-MS-DataTypes : 718
USED in MAP-MS-DataTypes : 701

lcsLocationInfo.....identifier of [1] LCSLocationInfo
DEFINED in MAP-LCS-DataTypes : 63

LCSLocationInfo.....type reference SEQUENCE
DEFINED in MAP-LCS-DataTypes : 67
USED in MAP-LCS-DataTypes : 63 286

lcsLocationInfo.....identifier of LCSLocationInfo
DEFINED in MAP-LCS-DataTypes : 286

lcs-ClientID.....identifier of [0] LCS-ClientID
DEFINED in MAP-LCS-DataTypes : 76

LCS-ClientID.....type reference SEQUENCE
DEFINED in MAP-LCS-DataTypes : 103
USED in MAP-LCS-DataTypes : 76 285

lcs-ClientID.....identifier of LCS-ClientID
DEFINED in MAP-LCS-DataTypes : 285

lcs-Event.....identifier of LCS-Event
DEFINED in MAP-LCS-DataTypes : 284

LCS-Event.....type reference ENUMERATED
DEFINED in MAP-LCS-DataTypes : 302
USED in MAP-LCS-DataTypes : 284

lcs-Priority.....identifier of [6] LCS-Priority
DEFINED in MAP-LCS-DataTypes : 82

LCS-Priority.....type reference OCTET STRING
DEFINED in MAP-LCS-DataTypes : 136
USED in MAP-LCS-DataTypes : 82

LCS-PrivacyClass.....type reference SEQUENCE
DEFINED in MAP-MS-DataTypes : 1122
USED in MAP-MS-DataTypes : 1118

lcs-PrivacyExceptionList.....identifier of [1] LCS-PrivacyExceptionList
DEFINED in MAP-MS-DataTypes : 720

LCS-PrivacyExceptionList.....type reference SEQUENCE OF
DEFINED in MAP-MS-DataTypes : 1117
USED in MAP-MS-DataTypes : 720

lcs-QoS.....identifier of [7] LCS-QoS
DEFINED in MAP-LCS-DataTypes : 83

LCS-QoS.....type reference SEQUENCE
DEFINED in MAP-LCS-DataTypes : 141
USED in MAP-LCS-DataTypes : 19 83

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 : 212

lmsi.....identifier of LMSI
DEFINED in MAP-MS-DataTypes : 1636

lmsi.....identifier of [1] LMSI
DEFINED in MAP-MS-DataTypes : 1680

lmsi.....identifier of LMSI
DEFINED in MAP-CommonDataTypes : 303

LMSI.....type reference OCTET STRING
DEFINED in MAP-CommonDataTypes : 333
USED in MAP-MS-DataTypes : 169 212 1636 1680
USED in MAP-CommonDataTypes : 36 303
USED in MAP-CH-DataTypes : 69 202 317
USED in MAP-SM-DataTypes : 35 87 134
USED in MAP-LCS-DataTypes : 32 69 80

lmsi.....identifier of [4] LMSI
DEFINED in MAP-CH-DataTypes : 202

lmsi.....identifier of [1] LMSI
DEFINED in MAP-CH-DataTypes : 317

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 : 69

lmsi.....identifier of [4] LMSI
DEFINED in MAP-LCS-DataTypes : 80

lmu-Indicator.....identifier of [21] NULL
DEFINED in MAP-MS-DataTypes : 700

locationEstimate.....identifier of Ext-GeographicalInformation
DEFINED in MAP-LCS-DataTypes : 186

locationEstimate.....identifier of [5] Ext-GeographicalInformation
DEFINED in MAP-LCS-DataTypes : 292

locationEstimateType.....identifier of [0] LocationEstimateType
DEFINED in MAP-LCS-DataTypes : 91

LocationEstimateType.....type reference ENUMERATED
DEFINED in MAP-LCS-DataTypes : 94
USED in MAP-LCS-DataTypes : 91

locationInformation.....identifier of [0] LocationInformation
DEFINED in MAP-MS-DataTypes : 1691

locationInformation.....identifier of [0] NULL
DEFINED in MAP-MS-DataTypes : 1697

LocationInformation.....type reference SEQUENCE
DEFINED in MAP-MS-DataTypes : 1705
USED in MAP-MS-DataTypes : 98 1691 1928

locationInformation.....identifier of [3] LocationInformation
DEFINED in MAP-MS-DataTypes : 1928

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 : 1709

LocationNumber.....type reference OCTET STRING
DEFINED in MAP-MS-DataTypes : 1745
USED in MAP-MS-DataTypes : 1709

locationProcedureNotCompleted.....identifier of Named Number, 4
DEFINED in MAP-ER-DataTypes : 365

locationProcedureNotSupportedByTargetMS.identifier of Named Number, 5
DEFINED in MAP-ER-DataTypes : 366

locationType.....identifier of LocationType
DEFINED in MAP-LCS-DataTypes : 74

LocationType.....type reference SEQUENCE
DEFINED in MAP-LCS-DataTypes : 90
USED in MAP-LCS-DataTypes : 17 74

longForwardedToNumber.....identifier of [10] FTN-AddressString
DEFINED in MAP-MS-DataTypes : 1000

longForwardedToNumber.....identifier of [8] FTN-AddressString
DEFINED in MAP-CH-DataTypes : 196

longForwardedToNumber.....identifier of [9] FTN-AddressString
DEFINED in MAP-SS-DataTypes : 107

longFTN-Supported.....identifier of [4] NULL
DEFINED in MAP-MS-DataTypes : 225

longFTN-Supported.....identifier of [4] NULL
DEFINED in MAP-MS-DataTypes : 1782

longFTN-Supported.....identifier of [6] NULL
DEFINED in MAP-MS-DataTypes : 1862

longFTN-Supported.....identifier of [21] NULL
DEFINED in MAP-CH-DataTypes : 117

longFTN-Supported.....identifier of [18] NULL
DEFINED in MAP-CH-DataTypes : 217

longFTN-Supported.....identifier of [9] NULL
DEFINED in MAP-SS-DataTypes : 81

longFTN-Supported.....identifier of [4] NULL
DEFINED in MAP-SS-DataTypes : 188

LongSignalInfo.....type reference OCTET STRING
DEFINED in MAP-CommonDataTypes : 245
USED in MAP-CommonDataTypes : 240

longTermDenial.....value reference LongTermDenial, CHOICE VALUE
DEFINED in MAP-Protocol : 413

LongTermDenial.....type reference ERROR
DEFINED in MAP-Errors : 392
USED in MAP-Protocol : 160 413
USED in MAP-SupplementaryServi : 54 265
USED in MAP-Errors : 74

longTermDenialParam.....identifier of LongTermDenialParam
DEFINED in MAP-Errors : 394

LongTermDenialParam.....type reference SEQUENCE
DEFINED in MAP-ER-DataTypes : 333
USED in MAP-Errors : 140 394
USED in MAP-ER-DataTypes : 48

lowdelay.....identifier of Named Number, 0
DEFINED in MAP-LCS-DataTypes : 168

lowerLayerCompatibility.....identifier of [5] ExternalSignalInfo
DEFINED in MAP-CH-DataTypes : 287

lsaActiveModeIndicator.....identifier of [2] NULL
DEFINED in MAP-MS-DataTypes : 861

lsaAttributes.....identifier of [1] LSAAttributes
DEFINED in MAP-MS-DataTypes : 860

LSAAttributes.....type reference OCTET STRING
DEFINED in MAP-MS-DataTypes : 878
USED in MAP-MS-DataTypes : 860

LSAData.....type reference SEQUENCE
DEFINED in MAP-MS-DataTypes : 858
USED in MAP-MS-DataTypes : 854

LSADataList.....type reference SEQUENCE OF

DEFINED in MAP-MS-DataTypes : 853
USED in MAP-MS-DataTypes : 871

lsaDataList.....identifier of [2] LSADataList
DEFINED in MAP-MS-DataTypes : 871

lsaIdentity.....identifier of [0] LSAIdentity
DEFINED in MAP-MS-DataTypes : 859

LSAIdentity.....type reference OCTET STRING
DEFINED in MAP-MS-DataTypes : 875
USED in MAP-MS-DataTypes : 55 859 1254 1713

lsaIdentityList.....identifier of LSAIdentityList
DEFINED in MAP-MS-DataTypes : 1251

LSAIdentityList.....type reference SEQUENCE OF
DEFINED in MAP-MS-DataTypes : 1253
USED in MAP-MS-DataTypes : 1251

lsaInformation.....identifier of [25] LSAInformation
DEFINED in MAP-MS-DataTypes : 699

LSAInformation.....type reference SEQUENCE
DEFINED in MAP-MS-DataTypes : 865
USED in MAP-MS-DataTypes : 699

lsaInformationWithdraw.....identifier of [12] LSAInformationWithdraw
DEFINED in MAP-MS-DataTypes : 1223

LSAInformationWithdraw.....type reference CHOICE
DEFINED in MAP-MS-DataTypes : 1249
USED in MAP-MS-DataTypes : 1223

LSAOnlyAccessIndicator.....type reference ENUMERATED
DEFINED in MAP-MS-DataTypes : 849
USED in MAP-MS-DataTypes : 870

lsaOnlyAccessIndicator.....identifier of [1] LSAOnlyAccessIndicator
DEFINED in MAP-MS-DataTypes : 870

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 : 151
USED in MAP-CommonDataTypes : 72

MAP-CallHandlingOperations.....module reference
DEFINED in MAP-CallHandlingOperat : 1
USED in MAP-Protocol : 64

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 : 168
USED in MAP-OperationAndMainte : 44
USED in MAP-MS-DataTypes : 187
USED in MAP-OM-DataTypes : 23
USED in MAP-CH-DataTypes : 75
USED in MAP-SS-DataTypes : 56
USED in MAP-SM-DataTypes : 36
USED in MAP-GR-DataTypes : 27
USED in MAP-LCS-DataTypes : 37
USED in MAP-ER-DataTypes : 76

MAP-Errors.....module reference
DEFINED in MAP-Errors : 1
USED in MAP-Protocol : 173
USED in MAP-MobileServiceOpera : 109
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 : 156
USED in MAP-MS-DataTypes : 197
USED in MAP-SM-DataTypes : 41

MAP-EXTENSION.....information object class reference CLASS
DEFINED in MAP-ExtensionDataTypes : 23
USED in MAP-ExtensionDataTypes : 47 49 54

MAP-ExtensionDataTypes.....module reference
DEFINED in MAP-ExtensionDataTypes : 1
USED in MAP-MS-DataTypes : 192
USED in MAP-CommonDataTypes : 82
USED in MAP-OM-DataTypes : 28
USED in MAP-CH-DataTypes : 80
USED in MAP-SS-DataTypes : 61
USED in MAP-SM-DataTypes : 46
USED in MAP-GR-DataTypes : 43
USED in MAP-LCS-DataTypes : 43
USED in MAP-ER-DataTypes : 86

MAP-Group-Call-Operations.....module reference
DEFINED in MAP-Group-Call-Operati : 1
USED in MAP-Protocol : 103

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 : 111

MAP-MobileServiceOperations.....module reference
DEFINED in MAP-MobileServiceOpera : 1
USED in MAP-Protocol : 41

MAP-MS-DataTypes.....module reference
DEFINED in MAP-MS-DataTypes : 1
USED in MAP-MobileServiceOpera : 163
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 : 49

MAP-Protocol.....module reference
DEFINED in MAP-Protocol : 1

MAP-ShortMessageServiceOperations.....module reference
DEFINED in MAP-ShortMessageServic : 1
USED in MAP-Protocol : 94

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 : 146
USED in MAP-CommonDataTypes : 77
USED in MAP-SS-DataTypes : 66
USED in MAP-ER-DataTypes : 81

MAP-SS-DataTypes.....module reference
DEFINED in MAP-SS-DataTypes : 1
USED in MAP-SupplementaryServi : 74
USED in MAP-Errors : 103
USED in MAP-MS-DataTypes : 141
USED in MAP-CH-DataTypes : 59
USED in MAP-LCS-DataTypes : 49

USED in MAP-ER-DataTypes : 69

MAP-SupplementaryServiceOperations.....module reference
DEFINED in MAP-SupplementaryServi : 1
USED in MAP-Protocol : 82

MAP-TS-Code.....module reference
DEFINED in MAP-TS-Code : 1
USED in MAP-MS-DataTypes : 156
USED in MAP-CommonDataTypes : 66
USED in MAP-GR-DataTypes : 32

matchType.....identifier of [0] MatchType
DEFINED in MAP-MS-DataTypes : 1401

MatchType.....type reference ENUMERATED
DEFINED in MAP-MS-DataTypes : 1431
USED in MAP-MS-DataTypes : 1401

maxAddressLength.....value reference INTEGER, 20
DEFINED in MAP-CommonDataTypes : 141
USED in MAP-CommonDataTypes : 100

maxAdd-GeographicalInformation.....value reference INTEGER, 91
DEFINED in MAP-LCS-DataTypes : 279
USED in MAP-LCS-DataTypes : 261

maxEventSpecification.....value reference INTEGER, 2
DEFINED in MAP-SS-DataTypes : 303
USED in MAP-SS-DataTypes : 300

maxExt-GeographicalInformation.....value reference INTEGER, 20
DEFINED in MAP-LCS-DataTypes : 257
USED in MAP-LCS-DataTypes : 201

maxFTN-AddressLength.....value reference INTEGER, 15
DEFINED in MAP-CommonDataTypes : 153
USED in MAP-CommonDataTypes : 150

maximumentitledPriority.....identifier of EMLPP-Priority
DEFINED in MAP-CommonDataTypes : 436

maximumEntitledPriority.....identifier of [0] EMLPP-Priority
DEFINED in MAP-SS-DataTypes : 194

maxISDN-AddressLength.....value reference INTEGER, 9
DEFINED in MAP-CommonDataTypes : 147
USED in MAP-CommonDataTypes : 18 144

maxISDN-SubaddressLength.....value reference INTEGER, 21
DEFINED in MAP-CommonDataTypes : 193
USED in MAP-CommonDataTypes : 156

maxLongSignalInfoLength.....value reference INTEGER, 2560
DEFINED in MAP-CommonDataTypes : 247
USED in MAP-CommonDataTypes : 245

MaxMC-Bearers.....type reference INTEGER
DEFINED in MAP-CommonDataTypes : 464
USED in MAP-CommonDataTypes : 53 459
USED in MAP-SS-DataTypes : 52 197

maxNameStringLength.....value reference INTEGER, 63
DEFINED in MAP-LCS-DataTypes : 134
USED in MAP-LCS-DataTypes : 132

maxNrOfRABs.....value reference INTEGER, 255
DEFINED in MAP-MS-DataTypes : 626
USED in MAP-MS-DataTypes : 624

maxNumOfBasicServiceGroups.....value reference INTEGER, 13
DEFINED in MAP-SS-DataTypes : 267
USED in MAP-SS-DataTypes : 96 153 264

maxNumOfBasicServices.....value reference INTEGER, 70
DEFINED in MAP-MS-DataTypes : 1259
USED in MAP-MS-DataTypes : 1256

maxNumOfBearerServices.....value reference INTEGER, 50
DEFINED in MAP-MS-DataTypes : 911
USED in MAP-MS-DataTypes : 908

maxNumOfCamelBasicServiceCriteria.....value reference INTEGER, 5

DEFINED in MAP-MS-DataTypes : 1425
USED in MAP-MS-DataTypes : 1416

maxNumOfCamelDestinationNumberLengths...value reference INTEGER, 3
DEFINED in MAP-MS-DataTypes : 1423
USED in MAP-MS-DataTypes : 1413

maxNumOfCamelDestinationNumbers.....value reference INTEGER, 10
DEFINED in MAP-MS-DataTypes : 1421
USED in MAP-MS-DataTypes : 1408

maxNumOfCamelSSEvents.....value reference INTEGER, 10
DEFINED in MAP-MS-DataTypes : 1334
USED in MAP-MS-DataTypes : 1323

maxNumOfCamelTDPData.....value reference INTEGER, 10
DEFINED in MAP-MS-DataTypes : 1354
USED in MAP-MS-DataTypes : 74 788 1347 1379 1382 1485 1556

maxNumOfCAMEL-O-CauseValueCriteria.....value reference INTEGER, 5
DEFINED in MAP-MS-DataTypes : 1441
USED in MAP-MS-DataTypes : 1435

maxNumOfCAMEL-T-CauseValueCriteria.....value reference INTEGER, 5
DEFINED in MAP-MS-DataTypes : 1443
USED in MAP-MS-DataTypes : 1438

maxNumOfCCBS-Requests.....value reference INTEGER, 5
DEFINED in MAP-SS-DataTypes : 204
USED in MAP-SS-DataTypes : 201 213

maxNumOfCUG.....value reference INTEGER, 10
DEFINED in MAP-MS-DataTypes : 1080
USED in MAP-MS-DataTypes : 1059

maxNumOfDP-AnalysedInfoCriteria.....value reference INTEGER, 10
DEFINED in MAP-MS-DataTypes : 1297
USED in MAP-MS-DataTypes : 1294

maxNumOfEncryptionInfo.....value reference INTEGER, 100
DEFINED in MAP-MS-DataTypes : 652
USED in MAP-MS-DataTypes : 644

maxNumOfExternalClient.....value reference INTEGER, 5
DEFINED in MAP-MS-DataTypes : 1140
USED in MAP-MS-DataTypes : 1137

maxNumOfExt-BasicServiceGroups.....value reference INTEGER, 32
DEFINED in MAP-MS-DataTypes : 1088
USED in MAP-MS-DataTypes : 985 1044 1082 1085

maxNumOfGMLC.....value reference INTEGER, 5
DEFINED in MAP-MS-DataTypes : 728
USED in MAP-MS-DataTypes : 724

maxNumOfHLR-Id.....value reference INTEGER, 50
DEFINED in MAP-CommonDataTypes : 331
USED in MAP-CommonDataTypes : 328

maxNumOfIntegrityInfo.....value reference INTEGER, 100
DEFINED in MAP-MS-DataTypes : 642
USED in MAP-MS-DataTypes : 634

maxNumOfISDN-AddressDigits.....value reference INTEGER, 15
DEFINED in MAP-MS-DataTypes : 1419
USED in MAP-MS-DataTypes : 1414

maxNumOfLSAs.....value reference INTEGER, 20
DEFINED in MAP-MS-DataTypes : 856
USED in MAP-MS-DataTypes : 853 1253

maxNumOfMC-Bearers.....value reference INTEGER, 7
DEFINED in MAP-CommonDataTypes : 468
USED in MAP-CommonDataTypes : 464 466

maxNumOfMobilityTriggers.....value reference INTEGER, 10
DEFINED in MAP-MS-DataTypes : 1529
USED in MAP-MS-DataTypes : 1526

maxNumOfMOLR-Class.....value reference INTEGER, 3
DEFINED in MAP-MS-DataTypes : 1176

USED in MAP-MS-DataTypes : 1173

maxNumOfPDP-Contexts.....value reference INTEGER, 50
DEFINED in MAP-MS-DataTypes : 742
USED in MAP-MS-DataTypes : 739 757 1246

maxNumOfPLMNClient.....value reference INTEGER, 5
DEFINED in MAP-MS-DataTypes : 1145
USED in MAP-MS-DataTypes : 1142

maxNumOfPrivacyClass.....value reference INTEGER, 4
DEFINED in MAP-MS-DataTypes : 1120
USED in MAP-MS-DataTypes : 1117

maxNumOfPrivateExtensions.....value reference INTEGER, 10
DEFINED in MAP-ExtensionDataTypes : 52
USED in MAP-ExtensionDataTypes : 43

maxNumOfRadioResources.....value reference INTEGER, 7
DEFINED in MAP-MS-DataTypes : 516
USED in MAP-MS-DataTypes : 507

maxNumOfRelocationNumber.....value reference INTEGER, 7
DEFINED in MAP-MS-DataTypes : 628
USED in MAP-MS-DataTypes : 613 616

maxNumOfServiceHandovers.....value reference INTEGER, 7
DEFINED in MAP-MS-DataTypes : 494
USED in MAP-MS-DataTypes : 485

maxNumOfSS.....value reference INTEGER, 30
DEFINED in MAP-SS-DataTypes : 259
USED in MAP-MS-DataTypes : 136 968
USED in MAP-SS-DataTypes : 32 256 261

maxNumOfTeleservices.....value reference INTEGER, 20
DEFINED in MAP-MS-DataTypes : 916
USED in MAP-MS-DataTypes : 913

maxNumOfVBSGroupIds.....value reference INTEGER, 50
DEFINED in MAP-MS-DataTypes : 1655
USED in MAP-MS-DataTypes : 1649

maxNumOfVGCSGroupIds.....value reference INTEGER, 50
DEFINED in MAP-MS-DataTypes : 1657
USED in MAP-MS-DataTypes : 1652

maxNumOfZoneCodes.....value reference INTEGER, 10
DEFINED in MAP-MS-DataTypes : 1190
USED in MAP-MS-DataTypes : 63 1184

maxPermittedEncryptionAlgorithmsLength..value reference INTEGER, 9
DEFINED in MAP-MS-DataTypes : 453
USED in MAP-MS-DataTypes : 445

maxPermittedIntegrityProtectionAlgorithmvalue reference INTEGER, 9
DEFINED in MAP-MS-DataTypes : 442
USED in MAP-MS-DataTypes : 434

maxSignalInfoLength.....value reference INTEGER, 200
DEFINED in MAP-CommonDataTypes : 206
USED in MAP-CommonDataTypes : 25 204

maxUSSD-StringLength.....value reference INTEGER, 160
DEFINED in MAP-SS-DataTypes : 242
USED in MAP-SS-DataTypes : 238

mc.....value reference SS-Code, '01000101'B
DEFINED in MAP-SS-Code : 84

mcef-Set.....identifier of Named Number, 2
DEFINED in MAP-SM-DataTypes : 191

mci.....value reference SS-Code, '00010101'B
DEFINED in MAP-SS-Code : 36

MC-Bearers.....type reference INTEGER
DEFINED in MAP-CommonDataTypes : 466
USED in MAP-CommonDataTypes : 54 460
USED in MAP-SS-DataTypes : 53 80 168 198 199

mc-SS-Info.....identifier of [28] MC-SS-Info

DEFINED in MAP-MS-DataTypes : 704

MC-SS-Info.....type reference SEQUENCE
DEFINED in MAP-CommonDataTypes : 456
USED in MAP-MS-DataTypes : 176 704
USED in MAP-CommonDataTypes : 52

memoryAvailable.....identifier of Named Number, 1
DEFINED in MAP-SM-DataTypes : 212

memoryCapacityExceeded.....identifier of Named Number, 0
DEFINED in MAP-SM-DataTypes : 167

memoryCapacityExceeded.....identifier of Named Number, 0
DEFINED in MAP-ER-DataTypes : 141

MessageType.....type reference CHOICE
DEFINED in TCAPMessages : 51
USED in TCAPMessages : 47

messageWaitingListFull.....value reference MessageWaitingListFull, CHOICE VALUE
DEFINED in MAP-Protocol : 420

MessageWaitingListFull.....type reference ERROR
DEFINED in MAP-Errors : 409
USED in MAP-Protocol : 155 420
USED in MAP-ShortMessageServic : 39 121
USED in MAP-Errors : 79

messageWaitListFullParam.....identifier of MessageWaitListFullParam
DEFINED in MAP-Errors : 411

MessageWaitListFullParam.....type reference SEQUENCE
DEFINED in MAP-ER-DataTypes : 314
USED in MAP-Errors : 134 411
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 : 56

mlc-Number.....identifier of ISDN-AddressString
DEFINED in MAP-LCS-DataTypes : 75

MM-Code.....type reference OCTET STRING
DEFINED in MAP-MS-DataTypes : 1531
USED in MAP-MS-DataTypes : 1527 1925

mm-EventNotSupported.....value reference MM-EventNotSupported, CHOICE VALUE
DEFINED in MAP-Protocol : 432

MM-EventNotSupported.....type reference ERROR
DEFINED in MAP-Errors : 448
USED in MAP-Protocol : 169 432
USED in MAP-MobileServiceOpera : 94 485
USED in MAP-Errors : 92

mm-EventNotSupported-Param.....identifier of MM-EventNotSupported-Param
DEFINED in MAP-Errors : 450

MM-EventNotSupported-Param.....type reference SEQUENCE
DEFINED in MAP-ER-DataTypes : 378
USED in MAP-Errors : 145 450
USED in MAP-ER-DataTypes : 53

mnrf-Set.....identifier of Named Number, 1
DEFINED in MAP-SM-DataTypes : 190

mnrg-Set.....identifier of Named Number, 3
DEFINED in MAP-SM-DataTypes : 192

mobileNotReachableReason.....identifier of [2] AbsentSubscriberDiagnosticSM

DEFINED in MAP-MS-DataTypes : 1595

mobilityTriggers.....identifier of MobilityTriggers
DEFINED in MAP-MS-DataTypes : 1516

MobilityTriggers.....type reference SEQUENCE OF
DEFINED in MAP-MS-DataTypes : 1526
USED in MAP-MS-DataTypes : 1516

ModificationInstruction.....type reference ENUMERATED
DEFINED in MAP-MS-DataTypes : 1899
USED in MAP-MS-DataTypes : 1878 1888 1894 1895

modificationRequestFor-CB-Info.....identifier of [3] ModificationRequestFor-CB-Info
DEFINED in MAP-MS-DataTypes : 1859

ModificationRequestFor-CB-Info.....type reference SEQUENCE
DEFINED in MAP-MS-DataTypes : 1882
USED in MAP-MS-DataTypes : 1859

modificationRequestFor-CF-Info.....identifier of [2] ModificationRequestFor-CF-Info
DEFINED in MAP-MS-DataTypes : 1858

ModificationRequestFor-CF-Info.....type reference SEQUENCE
DEFINED in MAP-MS-DataTypes : 1871
USED in MAP-MS-DataTypes : 1858

modificationRequestFor-CSI.....identifier of [4] ModificationRequestFor-CSI
DEFINED in MAP-MS-DataTypes : 1860

ModificationRequestFor-CSI.....type reference SEQUENCE
DEFINED in MAP-MS-DataTypes : 1892
USED in MAP-MS-DataTypes : 1860

modifyCSI-State.....identifier of [2] ModificationInstruction
DEFINED in MAP-MS-DataTypes : 1895

modifyNotificationToCSE.....identifier of [6] ModificationInstruction
DEFINED in MAP-MS-DataTypes : 1878

modifyNotificationToCSE.....identifier of [5] ModificationInstruction
DEFINED in MAP-MS-DataTypes : 1888

modifyNotificationToCSE.....identifier of [1] ModificationInstruction
DEFINED in MAP-MS-DataTypes : 1894

MOLR-Class.....type reference SEQUENCE
DEFINED in MAP-MS-DataTypes : 1178
USED in MAP-MS-DataTypes : 1174

molr-List.....identifier of [2] MOLR-List
DEFINED in MAP-MS-DataTypes : 721

MOLR-List.....type reference SEQUENCE OF
DEFINED in MAP-MS-DataTypes : 1173
USED in MAP-MS-DataTypes : 721

monitoringMode.....identifier of [0] MonitoringMode
DEFINED in MAP-CH-DataTypes : 358

MonitoringMode.....type reference ENUMERATED
DEFINED in MAP-CH-DataTypes : 363
USED in MAP-CH-DataTypes : 358

moreMessagesToSend.....identifier of NULL
DEFINED in MAP-SM-DataTypes : 123

mo-forwardSM.....value reference MO-ForwardSM, CHOICE VALUE
DEFINED in MAP-Protocol : 263

MO-ForwardSM.....type reference OPERATION
DEFINED in MAP-ShortMessageServic : 81
USED in MAP-Protocol : 88 263
USED in MAP-ShortMessageServic : 14

mo-forwardSM-Arg.....identifier of MO-ForwardSM-Arg
DEFINED in MAP-ShortMessageServic : 83

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 : 305

msc-Number.....identifier of [1] ISDN-AddressString
DEFINED in MAP-MS-DataTypes : 210

msc-Number.....identifier of [6] ISDN-AddressString
DEFINED in MAP-MS-DataTypes : 1714

msc-Number.....identifier of [1] ISDN-AddressString
DEFINED in MAP-CH-DataTypes : 200

msc-Number.....identifier of [0] ISDN-AddressString
DEFINED in MAP-SM-DataTypes : 98

msc-Number.....identifier of ISDN-AddressString
DEFINED in MAP-LCS-DataTypes : 68

msisdn.....identifier of ISDN-AddressString
DEFINED in MAP-OperationAndMainte : 79

msisdn.....identifier of [1] ISDN-AddressString
DEFINED in MAP-MS-DataTypes : 883

msisdn.....identifier of ISDN-AddressString
DEFINED in MAP-MS-DataTypes : 1907

msisdn.....identifier of [2] ISDN-AddressString
DEFINED in MAP-MS-DataTypes : 1927

msisdn.....identifier of [1] ISDN-AddressString
DEFINED in MAP-CommonDataTypes : 372

msisdn.....identifier of [0] ISDN-AddressString
DEFINED in MAP-CH-DataTypes : 95

msisdn.....identifier of [12] ISDN-AddressString
DEFINED in MAP-CH-DataTypes : 163

msisdn.....identifier of [2] ISDN-AddressString
DEFINED in MAP-CH-DataTypes : 201

msisdn.....identifier of [9] ISDN-AddressString
DEFINED in MAP-CH-DataTypes : 233

msisdn.....identifier of [0] ISDN-AddressString
DEFINED in MAP-SS-DataTypes : 226

msisdn.....identifier of [1] ISDN-AddressString
DEFINED in MAP-SS-DataTypes : 271

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 : 178

msisdn.....identifier of [3] ISDN-AddressString
DEFINED in MAP-LCS-DataTypes : 79

msisdn.....identifier of [0] ISDN-AddressString
DEFINED in MAP-LCS-DataTypes : 287

msNotReachable.....identifier of NULL
DEFINED in MAP-MS-DataTypes : 1643

msPurged.....identifier of Named Number, 0

DEFINED in MAP-MS-DataTypes : 1755

ms-Present.....identifier of Named Number, 0
DEFINED in MAP-SM-DataTypes : 211

mt-forwardSM.....value reference MT-ForwardSM, CHOICE VALUE
DEFINED in MAP-Protocol : 264

MT-ForwardSM.....type reference OPERATION
DEFINED in MAP-ShortMessageServic : 93
USED in MAP-Protocol : 89 264
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

multicallBearerInfo.....identifier of [3] MulticallBearerInfo
DEFINED in MAP-MS-DataTypes : 522

MulticallBearerInfo.....type reference INTEGER
DEFINED in MAP-MS-DataTypes : 616
USED in MAP-MS-DataTypes : 522

multipleBearerNotSupported.....identifier of NULL
DEFINED in MAP-MS-DataTypes : 523

multipleBearerRequested.....identifier of [3] NULL
DEFINED in MAP-MS-DataTypes : 468

multipleECT-Barred.....identifier of Named Number, 14
DEFINED in MAP-MS-DataTypes : 939

multiPTY.....value reference SS-Code, '01010001'B
DEFINED in MAP-SS-Code : 90

mw-Status.....identifier of MW-Status
DEFINED in MAP-SM-DataTypes : 184

MW-Status.....type reference BIT STRING
DEFINED in MAP-SM-DataTypes : 188
USED in MAP-SM-DataTypes : 184

m-csi.....identifier of Named Number, 6
DEFINED in MAP-MS-DataTypes : 1235

m-CSI.....identifier of [5] M-CSI
DEFINED in MAP-MS-DataTypes : 1273

M-CSI.....type reference SEQUENCE
DEFINED in MAP-MS-DataTypes : 1515
USED in MAP-MS-DataTypes : 1273 1850

m-CSI.....identifier of Named Number, 7
DEFINED in MAP-MS-DataTypes : 1812

m-CSI.....identifier of [12] M-CSI
DEFINED in MAP-MS-DataTypes : 1850

NAEA-CIC.....type reference OCTET STRING
DEFINED in MAP-CommonDataTypes : 363
USED in MAP-CommonDataTypes : 40 359

naea-PreferredCI.....identifier of [15] NAEA-PreferredCI
DEFINED in MAP-MS-DataTypes : 693

NAEA-PreferredCI.....type reference SEQUENCE
DEFINED in MAP-CommonDataTypes : 358
USED in MAP-MS-DataTypes : 174 693

USED in MAP-CommonDataTypes : 39

USED in MAP-CH-DataTypes : 72 160

naea-PreferredCI.....identifier of [10] NAEA-PreferredCI
DEFINED in MAP-CH-DataTypes : 160

naea-PreferredCIC.....identifier of [0] NAEA-CIC
DEFINED in MAP-CommonDataTypes : 359

nameString.....identifier of [2] NameString
DEFINED in MAP-LCS-DataTypes : 124

NameString.....type reference USSD-String
DEFINED in MAP-LCS-DataTypes : 132
USED in MAP-LCS-DataTypes : 124

na-ESRD.....identifier of [3] ISDN-AddressString
DEFINED in MAP-LCS-DataTypes : 290

na-ESRK.....identifier of [4] ISDN-AddressString
DEFINED in MAP-LCS-DataTypes : 291

na-ESRK.....identifier of [0] ISDN-AddressString
DEFINED in MAP-LCS-DataTypes : 314

na-ESRK-Request.....identifier of [0] NULL
DEFINED in MAP-ExtensionDataTypes : 67

nbrSB.....identifier of [2] MaxMC-Bearers
DEFINED in MAP-CommonDataTypes : 459

nbrSB.....identifier of [3] MaxMC-Bearers
DEFINED in MAP-SS-DataTypes : 197

nbrSN.....identifier of [5] MC-Bearers
DEFINED in MAP-SS-DataTypes : 199

nbrUser.....identifier of [3] MC-Bearers
DEFINED in MAP-CommonDataTypes : 460

nbrUser.....identifier of [8] MC-Bearers
DEFINED in MAP-SS-DataTypes : 80

nbrUser.....identifier of [5] MC-Bearers
DEFINED in MAP-SS-DataTypes : 168

nbrUser.....identifier of [4] MC-Bearers
DEFINED in MAP-SS-DataTypes : 198

negativePW-Check.....value reference NegativePW-Check, CHOICE VALUE
DEFINED in MAP-Protocol : 409

NegativePW-Check.....type reference ERROR
DEFINED in MAP-Errors : 383
USED in MAP-Protocol : 151 409
USED in MAP-SupplementaryServi : 46 138 157 229
USED in MAP-Errors : 71

netDetNotReachable.....identifier of NotReachableReason
DEFINED in MAP-MS-DataTypes : 1751

networkAccessMode.....identifier of [24] NetworkAccessMode
DEFINED in MAP-MS-DataTypes : 698

NetworkAccessMode.....type reference ENUMERATED
DEFINED in MAP-MS-DataTypes : 731
USED in MAP-MS-DataTypes : 698

networkNode-AreaRestricted.....identifier of Named Number, 0
DEFINED in MAP-MS-DataTypes : 1203

networkNode-Number.....identifier of [1] ISDN-AddressString
DEFINED in MAP-SM-DataTypes : 86

NetworkResource.....type reference ENUMERATED
DEFINED in MAP-CommonDataTypes : 348
USED in MAP-CommonDataTypes : 38
USED in MAP-ER-DataTypes : 75 170 177

networkResource.....identifier of NetworkResource
DEFINED in MAP-ER-DataTypes : 170

networkResource.....identifier of NetworkResource

DEFINED in MAP-ER-DataTypes : 177

networkSignalInfo.....identifier of [10] ExternalSignalInfo
DEFINED in MAP-CH-DataTypes : 105

networkSignalInfo.....identifier of [6] ExternalSignalInfo
DEFINED in MAP-CH-DataTypes : 204

networkSignalInfo.....identifier of [4] ExternalSignalInfo
DEFINED in MAP-SS-DataTypes : 315

new.....identifier of Named Number, 1
DEFINED in MAP-MS-DataTypes : 457

newPassword.....identifier of Password
DEFINED in MAP-SupplementaryServi : 221

newPasswordsMismatch.....identifier of Named Number, 2
DEFINED in MAP-ER-DataTypes : 137

noAdditionalInformation.....identifier of Named Number, 0
DEFINED in MAP-ER-DataTypes : 346

noCUG-Restrictions.....identifier of Named Number, 0
DEFINED in MAP-MS-DataTypes : 1076

noGroupCallNbParam.....identifier of NoGroupCallNbParam
DEFINED in MAP-Errors : 423

NoGroupCallNbParam.....type reference SEQUENCE
DEFINED in MAP-ER-DataTypes : 322
USED in MAP-Errors : 137 423
USED in MAP-ER-DataTypes : 45

noGroupCallNumberAvailable.....value reference NoGroupCallNumberAvailable, CHOICE VALUE
DEFINED in MAP-Protocol : 396

NoGroupCallNumberAvailable.....type reference ERROR
DEFINED in MAP-Errors : 421
USED in MAP-Protocol : 158 396
USED in MAP-Group-Call-Operati : 26 53
USED in MAP-Errors : 83

noHandoverNumberAvailable.....value reference NoHandoverNumberAvailable, CHOICE VALUE
DEFINED in MAP-Protocol : 358

NoHandoverNumberAvailable.....type reference ERROR
DEFINED in MAP-Errors : 254
USED in MAP-Protocol : 130 358
USED in MAP-MobileServiceOpera : 91 319
USED in MAP-Errors : 36

noPageResponse.....identifier of Named Number, 2
DEFINED in MAP-ER-DataTypes : 251

noReply.....identifier of Named Number, 2
DEFINED in MAP-CH-DataTypes : 132

noReplyConditionTime.....identifier of [7] Ext-NoRepCondTime
DEFINED in MAP-MS-DataTypes : 997

noReplyConditionTime.....identifier of [5] Ext-NoRepCondTime
DEFINED in MAP-MS-DataTypes : 1877

noReplyConditionTime.....identifier of [5] NoReplyConditionTime
DEFINED in MAP-SS-DataTypes : 77

NoReplyConditionTime.....type reference INTEGER
DEFINED in MAP-SS-DataTypes : 83
USED in MAP-SS-DataTypes : 30 77 105

noReplyConditionTime.....identifier of [7] NoReplyConditionTime
DEFINED in MAP-SS-DataTypes : 105

noResponseFromBusyMS.....identifier of Named Number, 3
DEFINED in MAP-CH-DataTypes : 404

noResponseFromFreeMS.....identifier of Named Number, 2
DEFINED in MAP-CH-DataTypes : 403

noRoamingNbParam.....identifier of NoRoamingNbParam

DEFINED in MAP-Errors : 276

NoRoamingNbParam.....type reference SEQUENCE
DEFINED in MAP-ER-DataTypes : 239
USED in MAP-Errors : 123 276
USED in MAP-ER-DataTypes : 33

noRoamingNumberAvailable.....value reference NoRoamingNumberAvailable, CHOICE VALUE
DEFINED in MAP-Protocol : 374

NoRoamingNumberAvailable.....type reference ERROR
DEFINED in MAP-Errors : 274
USED in MAP-Protocol : 134 374
USED in MAP-CallHandlingOperat : 39 116
USED in MAP-Errors : 45

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 : 377

NoSubscriberReply.....type reference ERROR
DEFINED in MAP-Errors : 291
USED in MAP-Protocol : 137 377
USED in MAP-CallHandlingOperat : 42 98
USED in MAP-Errors : 47

noSubscriberReplyParam.....identifier of NoSubscriberReplyParam
DEFINED in MAP-Errors : 293

NoSubscriberReplyParam.....type reference SEQUENCE
DEFINED in MAP-ER-DataTypes : 267
USED in MAP-Errors : 127 293
USED in MAP-ER-DataTypes : 36

NoteMM-Event.....type reference OPERATION
DEFINED in MAP-MobileServiceOpera : 476
USED in MAP-Protocol : 37
USED in MAP-MobileServiceOpera : 70

noteMM-EventArg.....identifier of NoteMM-EventArg
DEFINED in MAP-MobileServiceOpera : 478

NoteMM-EventArg.....type reference SEQUENCE
DEFINED in MAP-MS-DataTypes : 1923
USED in MAP-MobileServiceOpera : 159 478
USED in MAP-MS-DataTypes : 128

noteMM-EventRes.....identifier of NoteMM-EventRes
DEFINED in MAP-MobileServiceOpera : 480

NoteMM-EventRes.....type reference SEQUENCE
DEFINED in MAP-MS-DataTypes : 1933
USED in MAP-MobileServiceOpera : 160 480
USED in MAP-MS-DataTypes : 129

noteMsPresentForGprs.....value reference NoteMsPresentForGprs, CHOICE VALUE
DEFINED in MAP-Protocol : 314

NoteMsPresentForGprs.....type reference OPERATION
DEFINED in MAP-MobileServiceOpera : 463
USED in MAP-Protocol : 36 314
USED in MAP-MobileServiceOpera : 67

noteMsPresentForGprsArg.....identifier of NoteMsPresentForGprsArg
DEFINED in MAP-MobileServiceOpera : 465

NoteMsPresentForGprsArg.....type reference SEQUENCE
DEFINED in MAP-MS-DataTypes : 1615
USED in MAP-MobileServiceOpera : 157 465
USED in MAP-MS-DataTypes : 124

noteMsPresentForGprsRes.....identifier of NoteMsPresentForGprsRes
DEFINED in MAP-MobileServiceOpera : 467

NoteMsPresentForGprsRes.....type reference SEQUENCE
DEFINED in MAP-MS-DataTypes : 1622
USED in MAP-MobileServiceOpera : 158 467

USED in MAP-MS-DataTypes : 125

noteSubscriberDataModified.....value reference NoteSubscriberDataModified, CHOICE VALUE
DEFINED in MAP-Protocol : 285

NoteSubscriberDataModified.....type reference OPERATION
DEFINED in MAP-MobileServiceOpera : 296
USED in MAP-Protocol : 38 285
USED in MAP-MobileServiceOpera : 34

noteSubscriberDataModifiedArg.....identifier of NoteSubscriberDataModifiedArg
DEFINED in MAP-MobileServiceOpera : 298

NoteSubscriberDataModifiedArg.....type reference SEQUENCE
DEFINED in MAP-MS-DataTypes : 1905
USED in MAP-MobileServiceOpera : 149 298
USED in MAP-MS-DataTypes : 112

noteSubscriberDataModifiedRes.....identifier of NoteSubscriberDataModifiedRes
DEFINED in MAP-MobileServiceOpera : 300

NoteSubscriberDataModifiedRes.....type reference SEQUENCE
DEFINED in MAP-MS-DataTypes : 1916
USED in MAP-MobileServiceOpera : 150 300
USED in MAP-MS-DataTypes : 113

notForwarded.....identifier of Named Number, 1
DEFINED in MAP-MS-DataTypes : 1429

notificationToCSE.....identifier of [3] NULL
DEFINED in MAP-MS-DataTypes : 778

notificationToCSE.....identifier of [3] NULL
DEFINED in MAP-MS-DataTypes : 1283

notificationToCSE.....identifier of [0] NULL
DEFINED in MAP-MS-DataTypes : 1311

notificationToCSE.....identifier of [1] NULL
DEFINED in MAP-MS-DataTypes : 1341

notificationToCSE.....identifier of [3] NULL
DEFINED in MAP-MS-DataTypes : 1475

notificationToCSE.....identifier of [2] NULL
DEFINED in MAP-MS-DataTypes : 1520

notificationToCSE.....identifier of [1] NULL
DEFINED in MAP-MS-DataTypes : 1549

notificationToCSE.....identifier of NULL
DEFINED in MAP-MS-DataTypes : 1817

notificationToCSE.....identifier of NULL
DEFINED in MAP-MS-DataTypes : 1825

notificationToCSE.....identifier of NULL
DEFINED in MAP-MS-DataTypes : 1833

notificationToCSE.....identifier of [2] NULL
DEFINED in MAP-MS-DataTypes : 1945

notificationToCSE.....identifier of [4] NULL
DEFINED in MAP-MS-DataTypes : 1954

notificationToMSUser.....identifier of [0] NotificationToMSUser
DEFINED in MAP-MS-DataTypes : 1125

notificationToMSUser.....identifier of [1] NotificationToMSUser
DEFINED in MAP-MS-DataTypes : 1150

NotificationToMSUser.....type reference ENUMERATED
DEFINED in MAP-MS-DataTypes : 1164
USED in MAP-MS-DataTypes : 80 1125 1150

notifyAndVerify-LocationAllowedIfNoRespoidentifier of Named Number, 1
DEFINED in MAP-MS-DataTypes : 1166

notifyAndVerify-LocationNotAllowedIfNoReidentifier of Named Number, 2
DEFINED in MAP-MS-DataTypes : 1167

notifyLocationAllowed.....identifier of Named Number, 0

DEFINED in MAP-MS-DataTypes : 1165

notKnownToBePorted.....identifier of Named Number, 0
DEFINED in MAP-CH-DataTypes : 169

notProvidedFromVLR.....identifier of [2] NULL
DEFINED in MAP-MS-DataTypes : 1752

notReachable.....identifier of Named Number, 0
DEFINED in MAP-CH-DataTypes : 130

NotReachableReason.....type reference ENUMERATED
DEFINED in MAP-MS-DataTypes : 1754
USED in MAP-MS-DataTypes : 1751

notRegistered.....identifier of Named Number, 3
DEFINED in MAP-MS-DataTypes : 1758

not-derivable.....identifier of NULL
DEFINED in TCAPMessages : 168

npdbMismatch.....identifier of Named Number, 2
DEFINED in MAP-ER-DataTypes : 206

numberChanged.....value reference NumberChanged, CHOICE VALUE
DEFINED in MAP-Protocol : 339

NumberChanged.....type reference ERROR
DEFINED in MAP-Errors : 205
USED in MAP-Protocol : 121 339
USED in MAP-CallHandlingOperat : 36 93
USED in MAP-Errors : 23

numberChangedParam.....identifier of NumberChangedParam
DEFINED in MAP-Errors : 207

NumberChangedParam.....type reference SEQUENCE
DEFINED in MAP-ER-DataTypes : 211
USED in MAP-Errors : 115 207
USED in MAP-ER-DataTypes : 26

NumberOfForwarding.....type reference INTEGER
DEFINED in MAP-CH-DataTypes : 92
USED in MAP-CH-DataTypes : 20 97

numberOfForwarding.....identifier of [2] NumberOfForwarding
DEFINED in MAP-CH-DataTypes : 97

numberOfPW-AttemptsViolation.....value reference NumberOfPW-AttemptsViolation, CHOICE VALUE
DEFINED in MAP-Protocol : 410

NumberOfPW-AttemptsViolation.....type reference ERROR
DEFINED in MAP-Errors : 385
USED in MAP-Protocol : 152 410
USED in MAP-SupplementaryServi : 47 139 158 230
USED in MAP-Errors : 72

numberOfRequestedVectors.....identifier of NumberOfRequestedVectors
DEFINED in MAP-MS-DataTypes : 282

numberOfRequestedVectors.....identifier of NumberOfRequestedVectors
DEFINED in MAP-MS-DataTypes : 658

NumberOfRequestedVectors.....type reference INTEGER
DEFINED in MAP-MS-DataTypes : 665
USED in MAP-MS-DataTypes : 282 658

numberPortabilityStatus.....identifier of [13] NumberPortabilityStatus
DEFINED in MAP-CH-DataTypes : 164

NumberPortabilityStatus.....type reference ENUMERATED
DEFINED in MAP-CH-DataTypes : 168
USED in MAP-CH-DataTypes : 164

odb.....identifier of [2] NULL
DEFINED in MAP-MS-DataTypes : 1797

odb-Data.....identifier of [8] ODB-Data
DEFINED in MAP-MS-DataTypes : 893

ODB-Data.....type reference SEQUENCE
DEFINED in MAP-MS-DataTypes : 918

USED in MAP-MS-DataTypes : 60 893 1832

odb-Data.....identifier of ODB-Data
DEFINED in MAP-MS-DataTypes : 1832

odb-GeneralData.....identifier of ODB-GeneralData
DEFINED in MAP-MS-DataTypes : 919

ODB-GeneralData.....type reference BIT STRING
DEFINED in MAP-MS-DataTypes : 924
USED in MAP-MS-DataTypes : 919 1196

odb-GeneralData.....identifier of [4] ODB-GeneralData
DEFINED in MAP-MS-DataTypes : 1196

odb-HPLMN-Data.....identifier of ODB-HPLMN-Data
DEFINED in MAP-MS-DataTypes : 920

ODB-HPLMN-Data.....type reference BIT STRING
DEFINED in MAP-MS-DataTypes : 957
USED in MAP-MS-DataTypes : 920

odb-Info.....identifier of [3] ODB-Info
DEFINED in MAP-MS-DataTypes : 1788

ODB-Info.....type reference SEQUENCE
DEFINED in MAP-MS-DataTypes : 1831
USED in MAP-MS-DataTypes : 1788 1910

odb-Info.....identifier of [2] ODB-Info
DEFINED in MAP-MS-DataTypes : 1910

old.....identifier of Named Number, 0
DEFINED in MAP-MS-DataTypes : 456

omc-Id.....identifier of [3] AddressString
DEFINED in MAP-OM-DataTypes : 40

onlyMSC.....identifier of Named Number, 1
DEFINED in MAP-MS-DataTypes : 733

onlySGSN.....identifier of Named Number, 2
DEFINED in MAP-MS-DataTypes : 734

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 : 109

operatorDeterminedBarring.....identifier of Named Number, 1
DEFINED in MAP-MS-DataTypes : 906

operatorDeterminedBarring.....identifier of Named Number, 3
DEFINED in MAP-ER-DataTypes : 98

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 : 215

or-Capability.....identifier of [5] OR-Phase
DEFINED in MAP-CH-DataTypes : 100

or-Interrogation.....identifier of [4] NULL
DEFINED in MAP-CH-DataTypes : 99

or-Interrogation.....identifier of [10] NULL
DEFINED in MAP-CH-DataTypes : 208

or-NotAllowed.....value reference OR-NotAllowed, CHOICE VALUE
DEFINED in MAP-Protocol : 380

OR-NotAllowed.....type reference ERROR
DEFINED in MAP-Errors : 316

USED in MAP-Protocol : 133 380

USED in MAP-CallHandlingOperat : 34 91 114 126
USED in MAP-Errors : 44

or-NotAllowedParam.....identifier of OR-NotAllowedParam
DEFINED in MAP-Errors : 318

OR-NotAllowedParam.....type reference SEQUENCE
DEFINED in MAP-ER-DataTypes : 193
USED in MAP-Errors : 124 318
USED in MAP-ER-DataTypes : 24

OR-Phase.....type reference INTEGER
DEFINED in MAP-CH-DataTypes : 125
USED in MAP-CH-DataTypes : 100

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 : 173

OverrideCategory.....type reference ENUMERATED
DEFINED in MAP-SS-DataTypes : 180
USED in MAP-SS-DataTypes : 28 173

overrideDisabled.....identifier of Named Number, 1
DEFINED in MAP-SS-DataTypes : 182

overrideEnabled.....identifier of Named Number, 0
DEFINED in MAP-SS-DataTypes : 181

ownNumberPortedOut.....identifier of Named Number, 1
DEFINED in MAP-CH-DataTypes : 170

o-andM-HPLMN.....identifier of Named Number, 1
DEFINED in MAP-CommonDataTypes : 382

o-andM-VPLMN.....identifier of Named Number, 2
DEFINED in MAP-CommonDataTypes : 383

O-BcsmCamelTDPCriteriaList.....type reference SEQUENCE OF
DEFINED in MAP-MS-DataTypes : 1379
USED in MAP-MS-DataTypes : 66 1271 1839
USED in MAP-CH-DataTypes : 45 277

O-BcsmCamelTDPData.....type reference SEQUENCE
DEFINED in MAP-MS-DataTypes : 1356
USED in MAP-MS-DataTypes : 1348

o-BcsmCamelTDPDataList.....identifier of O-BcsmCamelTDPDataList
DEFINED in MAP-MS-DataTypes : 1337

O-BcsmCamelTDPDataList.....type reference SEQUENCE OF
DEFINED in MAP-MS-DataTypes : 1347
USED in MAP-MS-DataTypes : 1337

O-BcsmCamelTDP-Criteria.....type reference SEQUENCE
DEFINED in MAP-MS-DataTypes : 1385
USED in MAP-MS-DataTypes : 1380

o-BcsmCamelTDP-CriteriaList.....identifier of [4] O-BcsmCamelTDPCriteriaList
DEFINED in MAP-MS-DataTypes : 1271

o-BcsmCamelTDP-CriteriaList.....identifier of [1] O-BcsmCamelTDPCriteriaList
DEFINED in MAP-MS-DataTypes : 1839

o-BcsmCamelTDP-CriteriaList.....identifier of [3] O-BcsmCamelTDPCriteriaList
DEFINED in MAP-CH-DataTypes : 277

o-BcsmTriggerDetectionPoint.....identifier of O-BcsmTriggerDetectionPoint
DEFINED in MAP-MS-DataTypes : 1357

O-BcsmTriggerDetectionPoint.....type reference ENUMERATED
DEFINED in MAP-MS-DataTypes : 1367
USED in MAP-MS-DataTypes : 1357 1386

o-BcsmTriggerDetectionPoint.....identifier of O-BcsmTriggerDetectionPoint
DEFINED in MAP-MS-DataTypes : 1386

o-CauseValueCriteria.....identifier of [3] O-CauseValueCriteria
DEFINED in MAP-MS-DataTypes : 1391

O-CauseValueCriteria.....type reference SEQUENCE OF
DEFINED in MAP-MS-DataTypes : 1435
USED in MAP-MS-DataTypes : 1391

o-csi.....identifier of Named Number, 0
DEFINED in MAP-MS-DataTypes : 1229

o-CSI.....identifier of [0] O-CSI
DEFINED in MAP-MS-DataTypes : 1267

O-CSI.....type reference SEQUENCE
DEFINED in MAP-MS-DataTypes : 1336
USED in MAP-MS-DataTypes : 64 1267 1838
USED in MAP-CH-DataTypes : 43 230 274

o-CSI.....identifier of Named Number, 0
DEFINED in MAP-MS-DataTypes : 1805

o-CSI.....identifier of [0] O-CSI
DEFINED in MAP-MS-DataTypes : 1838

o-CSI.....identifier of [5] O-CSI
DEFINED in MAP-CH-DataTypes : 230

o-CSI.....identifier of [1] O-CSI
DEFINED in MAP-CH-DataTypes : 274

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

password.....identifier of Password
DEFINED in MAP-MS-DataTypes : 1823

password.....identifier of [3] Password
DEFINED in MAP-MS-DataTypes : 1886

password.....identifier of [2] Password
DEFINED in MAP-MS-DataTypes : 1952

Password.....type reference NumericString
DEFINED in MAP-SS-DataTypes : 244
USED in MAP-SupplementaryServi : 66 221 238
USED in MAP-MS-DataTypes : 140 1823 1886 1952
USED in MAP-SS-DataTypes : 24

pcs-Extensions.....identifier of [1] PCS-Extensions
DEFINED in MAP-ExtensionDataTypes : 40

PCS-Extensions.....type reference SEQUENCE
DEFINED in MAP-ExtensionDataTypes : 62
USED in MAP-ExtensionDataTypes : 40

pdp-Address.....identifier of [17] PDP-Address
DEFINED in MAP-MS-DataTypes : 747

PDP-Address.....type reference OCTET STRING
DEFINED in MAP-MS-DataTypes : 828
USED in MAP-MS-DataTypes : 747

pdp-ChargingCharacteristics.....identifier of [1] ChargingCharacteristics
DEFINED in MAP-MS-DataTypes : 754

PDP-Context.....type reference SEQUENCE
DEFINED in MAP-MS-DataTypes : 744
USED in MAP-MS-DataTypes : 740

pdp-ContextChangeOfPosition.....identifier of Named Number, 14
DEFINED in MAP-MS-DataTypes : 815

pdp-ContextEstablishment.....identifier of Named Number, 11
DEFINED in MAP-MS-DataTypes : 813

pdp-ContextEstablishmentAcknowledgement.identifier of Named Number, 12
DEFINED in MAP-MS-DataTypes : 814

pdp-ContextId.....identifier of ContextId
DEFINED in MAP-MS-DataTypes : 745

pdp-Type.....identifier of [16] PDP-Type
DEFINED in MAP-MS-DataTypes : 746

PDP-Type.....type reference OCTET STRING
DEFINED in MAP-MS-DataTypes : 825
USED in MAP-MS-DataTypes : 746

permanent.....identifier of Named Number, 0
DEFINED in MAP-SS-DataTypes : 176

PermittedEncryptionAlgorithms.....type reference OCTET STRING
DEFINED in MAP-MS-DataTypes : 444
USED in MAP-MS-DataTypes : 429

PermittedIntegrityProtectionAlgorithms..type reference OCTET STRING
DEFINED in MAP-MS-DataTypes : 433
USED in MAP-MS-DataTypes : 428

phase1.....identifier of Named Number, 0
DEFINED in MAP-MS-DataTypes : 1465

phase2.....identifier of Named Number, 1
DEFINED in MAP-MS-DataTypes : 1466

phase3.....identifier of Named Number, 2
DEFINED in MAP-MS-DataTypes : 1467

plmn.....identifier of Named Number, 0
DEFINED in MAP-CommonDataTypes : 349

plmnClientList.....identifier of [2] PLMNClientList
DEFINED in MAP-MS-DataTypes : 1131

PLMNClientList.....type reference SEQUENCE OF
DEFINED in MAP-MS-DataTypes : 1142
USED in MAP-MS-DataTypes : 1131

plmnoperator.....value reference SS-Code, '10110100'B
DEFINED in MAP-SS-Code : 168

plmnOperatorServices.....identifier of Named Number, 2
DEFINED in MAP-LCS-DataTypes : 114

plmnRoamingNotAllowed.....identifier of Named Number, 0
DEFINED in MAP-ER-DataTypes : 97

plmn-SpecificBarringType1.....identifier of Named Number, 0
DEFINED in MAP-MS-DataTypes : 958

plmn-SpecificBarringType2.....identifier of Named Number, 1
DEFINED in MAP-MS-DataTypes : 959

plmn-SpecificBarringType3.....identifier of Named Number, 2
DEFINED in MAP-MS-DataTypes : 960

plmn-SpecificBarringType4.....identifier of Named Number, 3
DEFINED in MAP-MS-DataTypes : 961

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 : 137

plmn-specificSS-2.....value reference SS-Code, '11110010'B
DEFINED in MAP-SS-Code : 138

plmn-specificSS-3.....value reference SS-Code, '11110011'B
DEFINED in MAP-SS-Code : 139

plmn-specificSS-4.....value reference SS-Code, '11110100'B
DEFINED in MAP-SS-Code : 140

plmn-specificSS-5.....value reference SS-Code, '11110101'B
DEFINED in MAP-SS-Code : 141

plmn-specificSS-6.....value reference SS-Code, '11110110'B
DEFINED in MAP-SS-Code : 142

plmn-specificSS-7.....value reference SS-Code, '11110111'B
DEFINED in MAP-SS-Code : 143

plmn-specificSS-8.....value reference SS-Code, '11111000'B
DEFINED in MAP-SS-Code : 144

plmn-specificSS-9.....value reference SS-Code, '11111001'B
DEFINED in MAP-SS-Code : 145

plmn-specificSS-A.....value reference SS-Code, '11111010'B
DEFINED in MAP-SS-Code : 146

plmn-specificSS-B.....value reference SS-Code, '11111011'B
DEFINED in MAP-SS-Code : 147

plmn-specificSS-C.....value reference SS-Code, '11111100'B
DEFINED in MAP-SS-Code : 148

plmn-specificSS-D.....value reference SS-Code, '11111101'B

DEFINED in MAP-SS-Code : 149

plmn-specificSS-E.....value reference SS-Code, '11111110'B
DEFINED in MAP-SS-Code : 150

plmn-specificSS-F.....value reference SS-Code, '11111111'B
DEFINED in MAP-SS-Code : 151

plmn-specificTS-1.....value reference TeleserviceCode, '11010001'B
DEFINED in MAP-TS-Code : 72

plmn-specificTS-2.....value reference TeleserviceCode, '11010010'B
DEFINED in MAP-TS-Code : 73

plmn-specificTS-3.....value reference TeleserviceCode, '11010011'B
DEFINED in MAP-TS-Code : 74

plmn-specificTS-4.....value reference TeleserviceCode, '11010100'B
DEFINED in MAP-TS-Code : 75

plmn-specificTS-5.....value reference TeleserviceCode, '11010101'B
DEFINED in MAP-TS-Code : 76

plmn-specificTS-6.....value reference TeleserviceCode, '11010110'B
DEFINED in MAP-TS-Code : 77

plmn-specificTS-7.....value reference TeleserviceCode, '11010111'B
DEFINED in MAP-TS-Code : 78

plmn-specificTS-8.....value reference TeleserviceCode, '11011000'B
DEFINED in MAP-TS-Code : 79

plmn-specificTS-9.....value reference TeleserviceCode, '11011001'B
DEFINED in MAP-TS-Code : 80

plmn-specificTS-A.....value reference TeleserviceCode, '11011010'B
DEFINED in MAP-TS-Code : 81

plmn-specificTS-B.....value reference TeleserviceCode, '11011011'B
DEFINED in MAP-TS-Code : 82

plmn-specificTS-C.....value reference TeleserviceCode, '11011100'B
DEFINED in MAP-TS-Code : 83

plmn-specificTS-D.....value reference TeleserviceCode, '11011101'B
DEFINED in MAP-TS-Code : 84

plmn-specificTS-E.....value reference TeleserviceCode, '11011110'B
DEFINED in MAP-TS-Code : 85

plmn-specificTS-F.....value reference TeleserviceCode, '11011111'B
DEFINED in MAP-TS-Code : 86

polygon.....identifier of Named Number, 3
DEFINED in MAP-LCS-DataTypes : 178

positionMethodFailure.....value reference PositionMethodFailure, CHOICE VALUE
DEFINED in MAP-Protocol : 427

PositionMethodFailure.....type reference ERROR
DEFINED in MAP-Errors : 438
USED in MAP-Protocol : 164 427
USED in MAP-LocationServiceOpe : 31 82
USED in MAP-Errors : 88

positionMethodFailure-Diagnostic.....identifier of [0] PositionMethodFailure-Diagnostic
DEFINED in MAP-ER-DataTypes : 356

PositionMethodFailure-Diagnostic.....type reference ENUMERATED
DEFINED in MAP-ER-DataTypes : 360
USED in MAP-ER-DataTypes : 356

positionMethodFailure-Param.....identifier of PositionMethodFailure-Param
DEFINED in MAP-Errors : 440

PositionMethodFailure-Param.....type reference SEQUENCE
DEFINED in MAP-ER-DataTypes : 355
USED in MAP-Errors : 143 440
USED in MAP-ER-DataTypes : 51

positionMethodNotAvailableInLocationAreaidentifier of Named Number, 8
DEFINED in MAP-ER-DataTypes : 369

positionMethodNotAvailableInNetwork.....identifier of Named Number, 7
DEFINED in MAP-ER-DataTypes : 368

preferentialCUG-Indicator.....identifier of CUG-Index
DEFINED in MAP-MS-DataTypes : 1092

premiumRateEntertainementOGCallsBarred..identifier of Named Number, 4
DEFINED in MAP-MS-DataTypes : 932

premiumRateInformationOGCallsBarred.....identifier of Named Number, 3
DEFINED in MAP-MS-DataTypes : 931

prepareGroupCall.....value reference PrepareGroupCall, CHOICE VALUE
DEFINED in MAP-Protocol : 294

PrepareGroupCall.....type reference OPERATION
DEFINED in MAP-Group-Call-Operati : 46
USED in MAP-Protocol : 99 294
USED in MAP-Group-Call-Operati : 13

prepareGroupCallArg.....identifier of PrepareGroupCallArg
DEFINED in MAP-Group-Call-Operati : 48

PrepareGroupCallArg.....type reference SEQUENCE
DEFINED in MAP-GR-DataTypes : 49
USED in MAP-Group-Call-Operati : 31 48
USED in MAP-GR-DataTypes : 14

prepareGroupCallRes.....identifier of PrepareGroupCallRes
DEFINED in MAP-Group-Call-Operati : 50

PrepareGroupCallRes.....type reference SEQUENCE
DEFINED in MAP-GR-DataTypes : 61
USED in MAP-Group-Call-Operati : 32 50
USED in MAP-GR-DataTypes : 15

prepareHandover.....value reference PrepareHandover, CHOICE VALUE
DEFINED in MAP-Protocol : 189

PrepareHandover.....type reference OPERATION
DEFINED in MAP-MobileServiceOpera : 310
USED in MAP-Protocol : 17 189
USED in MAP-MobileServiceOpera : 38

prepareHO-Arg.....identifier of PrepareHO-Arg
DEFINED in MAP-MobileServiceOpera : 312

PrepareHO-Arg.....type reference [3] SEQUENCE
DEFINED in MAP-MS-DataTypes : 463
USED in MAP-MobileServiceOpera : 123 312
USED in MAP-MS-DataTypes : 33

prepareHO-Res.....identifier of PrepareHO-Res
DEFINED in MAP-MobileServiceOpera : 314

PrepareHO-Res.....type reference [3] SEQUENCE
DEFINED in MAP-MS-DataTypes : 518
USED in MAP-MobileServiceOpera : 124 314
USED in MAP-MS-DataTypes : 34

prepareSubsequentHandover.....value reference PrepareSubsequentHandover, CHOICE VALUE
DEFINED in MAP-Protocol : 193

PrepareSubsequentHandover.....type reference OPERATION
DEFINED in MAP-MobileServiceOpera : 337
USED in MAP-Protocol : 21 193
USED in MAP-MobileServiceOpera : 42

prepareSubsequentHO-Arg.....identifier of PrepareSubsequentHO-Arg
DEFINED in MAP-MobileServiceOpera : 339

PrepareSubsequentHO-Arg.....type reference [3] SEQUENCE
DEFINED in MAP-MS-DataTypes : 564
USED in MAP-MobileServiceOpera : 130 339
USED in MAP-MS-DataTypes : 35

prepareSubsequentHO-Res.....identifier of PrepareSubsequentHO-Res
DEFINED in MAP-MobileServiceOpera : 341

PrepareSubsequentHO-Res.....type reference [3] SEQUENCE
DEFINED in MAP-MS-DataTypes : 573

USED in MAP-MobileServiceOpera : 129 341

USED in MAP-MS-DataTypes : 36

pre-pagingSupported.....identifier of [19] NULL
DEFINED in MAP-CH-DataTypes : 115

pre-pagingSupported.....identifier of [17] NULL
DEFINED in MAP-CH-DataTypes : 216

priority.....identifier of [2] EMLPP-Priority
DEFINED in MAP-GR-DataTypes : 56

priorityLevel0.....value reference EMLPP-Priority, 0
DEFINED in MAP-CommonDataTypes : 449

priorityLevel1.....value reference EMLPP-Priority, 1
DEFINED in MAP-CommonDataTypes : 450

priorityLevel2.....value reference EMLPP-Priority, 2
DEFINED in MAP-CommonDataTypes : 451

priorityLevel3.....value reference EMLPP-Priority, 3
DEFINED in MAP-CommonDataTypes : 452

priorityLevel4.....value reference EMLPP-Priority, 4
DEFINED in MAP-CommonDataTypes : 453

priorityLevelA.....value reference EMLPP-Priority, 6
DEFINED in MAP-CommonDataTypes : 447

priorityLevelB.....value reference EMLPP-Priority, 5
DEFINED in MAP-CommonDataTypes : 448

privacyOverride.....identifier of [1] NULL
DEFINED in MAP-LCS-DataTypes : 77

privacyOverrideNotApplicable.....identifier of Named Number, 3
DEFINED in MAP-ER-DataTypes : 349

PrivateExtension.....type reference SEQUENCE
DEFINED in MAP-ExtensionDataTypes : 46
USED in MAP-ExtensionDataTypes : 15 44

privateExtensionList.....identifier of [0] PrivateExtensionList
DEFINED in MAP-ExtensionDataTypes : 31

privateExtensionList.....identifier of [0] PrivateExtensionList
DEFINED in MAP-ExtensionDataTypes : 39

PrivateExtensionList.....type reference SEQUENCE OF
DEFINED in MAP-ExtensionDataTypes : 43
USED in MAP-ExtensionDataTypes : 31 39

problem.....identifier of CHOICE
DEFINED in TCAPMessages : 169

processAccessSignalling.....value reference ProcessAccessSignalling, CHOICE VALUE
DEFINED in MAP-Protocol : 191

ProcessAccessSignalling.....type reference OPERATION
DEFINED in MAP-MobileServiceOpera : 329
USED in MAP-Protocol : 19 191
USED in MAP-MobileServiceOpera : 40

processAccessSignalling-Arg.....identifier of ProcessAccessSignalling-Arg
DEFINED in MAP-MobileServiceOpera : 331

ProcessAccessSignalling-Arg.....type reference [3] SEQUENCE
DEFINED in MAP-MS-DataTypes : 578
USED in MAP-MobileServiceOpera : 126 331
USED in MAP-MS-DataTypes : 37

processGroupCallSignalling.....value reference ProcessGroupCallSignalling, CHOICE VALUE
DEFINED in MAP-Protocol : 296

ProcessGroupCallSignalling.....type reference OPERATION
DEFINED in MAP-Group-Call-Operati : 63
USED in MAP-Protocol : 100 296
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 : 86
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 : 250

ProcessUnstructuredSS-Request.....type reference OPERATION
DEFINED in MAP-SupplementaryServi : 175
USED in MAP-Protocol : 74 250
USED in MAP-SupplementaryServi : 18

protocolId.....identifier of ProtocolId
DEFINED in MAP-CommonDataTypes : 196

ProtocolId.....type reference ENUMERATED
DEFINED in MAP-CommonDataTypes : 214
USED in MAP-CommonDataTypes : 196

provideRoamingNumber.....value reference ProvideRoamingNumber, CHOICE VALUE
DEFINED in MAP-Protocol : 232

ProvideRoamingNumber.....type reference OPERATION
DEFINED in MAP-CallHandlingOperat : 103
USED in MAP-Protocol : 55 232
USED in MAP-CallHandlingOperat : 14

provideRoamingNumberArg.....identifier of ProvideRoamingNumberArg
DEFINED in MAP-CallHandlingOperat : 106

ProvideRoamingNumberArg.....type reference SEQUENCE
DEFINED in MAP-CH-DataTypes : 198
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 : 219
USED in MAP-CallHandlingOperat : 57 108
USED in MAP-CH-DataTypes : 17

provideSIWFSNumber.....value reference ProvideSIWFSNumber, CHOICE VALUE
DEFINED in MAP-Protocol : 234

ProvideSIWFSNumber.....type reference OPERATION
DEFINED in MAP-CallHandlingOperat : 130
USED in MAP-Protocol : 57 234
USED in MAP-CallHandlingOperat : 16

provideSIWFSNumberArg.....identifier of ProvideSIWFSNumberArg
DEFINED in MAP-CallHandlingOperat : 132

ProvideSIWFSNumberArg.....type reference SEQUENCE
DEFINED in MAP-CH-DataTypes : 281
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 : 299
USED in MAP-CallHandlingOperat : 61 134
USED in MAP-CH-DataTypes : 24

provideSubscriberInfo.....value reference ProvideSubscriberInfo, CHOICE VALUE
DEFINED in MAP-Protocol : 272

ProvideSubscriberInfo.....type reference OPERATION
DEFINED in MAP-MobileServiceOpera : 233
USED in MAP-Protocol : 30 272
USED in MAP-MobileServiceOpera : 24

provideSubscriberInfoArg.....identifier of ProvideSubscriberInfoArg
DEFINED in MAP-MobileServiceOpera : 235

ProvideSubscriberInfoArg.....type reference SEQUENCE
DEFINED in MAP-MS-DataTypes : 1678

USED in MAP-MobileServiceOpera : 143 235

USED in MAP-MS-DataTypes : 95

provideSubscriberInfoRes.....identifier of ProvideSubscriberInfoRes
DEFINED in MAP-MobileServiceOpera : 237

ProvideSubscriberInfoRes.....type reference SEQUENCE
DEFINED in MAP-MS-DataTypes : 1685
USED in MAP-MobileServiceOpera : 144 237
USED in MAP-MS-DataTypes : 96

provideSubscriberLocation.....value reference ProvideSubscriberLocation, CHOICE VALUE
DEFINED in MAP-Protocol : 318

ProvideSubscriberLocation.....type reference OPERATION
DEFINED in MAP-LocationServiceOpe : 66
USED in MAP-Protocol : 108 318
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 : 73
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 : 185
USED in MAP-LocationServiceOpe : 44 70
USED in MAP-LCS-DataTypes : 14

provisionedSS.....identifier of [7] Ext-SS-InfoList
DEFINED in MAP-MS-DataTypes : 892

purgedMS.....identifier of Named Number, 3
DEFINED in MAP-ER-DataTypes : 253

purgeMS.....value reference PurgeMS, CHOICE VALUE
DEFINED in MAP-Protocol : 183

PurgeMS.....type reference OPERATION
DEFINED in MAP-MobileServiceOpera : 198
USED in MAP-Protocol : 14 183
USED in MAP-MobileServiceOpera : 17

purgeMS-Arg.....identifier of PurgeMS-Arg
DEFINED in MAP-MobileServiceOpera : 200

PurgeMS-Arg.....type reference [3] SEQUENCE
DEFINED in MAP-MS-DataTypes : 267
USED in MAP-MobileServiceOpera : 117 200
USED in MAP-MS-DataTypes : 20

purgeMS-Res.....identifier of PurgeMS-Res
DEFINED in MAP-MobileServiceOpera : 202

PurgeMS-Res.....type reference SEQUENCE
DEFINED in MAP-MS-DataTypes : 274
USED in MAP-MobileServiceOpera : 118 202
USED in MAP-MS-DataTypes : 21

pvlr.....identifier of Named Number, 3
DEFINED in MAP-CommonDataTypes : 352

pw-RegistrationFailure.....value reference PW-RegistrationFailure, CHOICE VALUE
DEFINED in MAP-Protocol : 408

PW-RegistrationFailure.....type reference ERROR
DEFINED in MAP-Errors : 379
USED in MAP-Protocol : 150 408
USED in MAP-SupplementaryServi : 45 228
USED in MAP-Errors : 70

pw-RegistrationFailureCause.....identifier of PW-RegistrationFailureCause
DEFINED in MAP-Errors : 381

PW-RegistrationFailureCause.....type reference ENUMERATED
DEFINED in MAP-ER-DataTypes : 134

USED in MAP-Errors : 108 381

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 : 367

qos-Subscribed.....identifier of [18] QoS-Subscribed
DEFINED in MAP-MS-DataTypes : 748

QoS-Subscribed.....type reference OCTET STRING
DEFINED in MAP-MS-DataTypes : 836
USED in MAP-MS-DataTypes : 81 748

quintupletList.....identifier of [1] QuintupletList
DEFINED in MAP-MS-DataTypes : 305

QuintupletList.....type reference SEQUENCE OF
DEFINED in MAP-MS-DataTypes : 310
USED in MAP-MS-DataTypes : 305

rab-Id.....identifier of [12] RAB-Id
DEFINED in MAP-MS-DataTypes : 478

rab-Id.....identifier of RAB-Id
DEFINED in MAP-MS-DataTypes : 490

rab-Id.....identifier of RAB-Id
DEFINED in MAP-MS-DataTypes : 512

rab-Id.....identifier of RAB-Id
DEFINED in MAP-MS-DataTypes : 620

RAB-Id.....type reference INTEGER
DEFINED in MAP-MS-DataTypes : 624
USED in MAP-MS-DataTypes : 478 490 512 569 583 620

RadioResource.....type reference SEQUENCE
DEFINED in MAP-MS-DataTypes : 510
USED in MAP-MS-DataTypes : 508

radioResourceInformation.....identifier of [6] RadioResourceInformation
DEFINED in MAP-MS-DataTypes : 414

radioResourceInformation.....identifier of [7] RadioResourceInformation
DEFINED in MAP-MS-DataTypes : 472

radioResourceInformation.....identifier of RadioResourceInformation
DEFINED in MAP-MS-DataTypes : 511

RadioResourceInformation.....type reference OCTET STRING
DEFINED in MAP-MS-DataTypes : 630
USED in MAP-MS-DataTypes : 414 472 511

radioResourceList.....identifier of [7] RadioResourceList
DEFINED in MAP-MS-DataTypes : 417

radioResourceList.....identifier of [11] RadioResourceList
DEFINED in MAP-MS-DataTypes : 475

RadioResourceList.....type reference SEQUENCE OF
DEFINED in MAP-MS-DataTypes : 507
USED in MAP-MS-DataTypes : 417 475

ranap-ServiceHandover.....identifier of [8] RANAP-ServiceHandover
DEFINED in MAP-MS-DataTypes : 419

ranap-ServiceHandover.....identifier of [14] RANAP-ServiceHandover
DEFINED in MAP-MS-DataTypes : 480

RANAP-ServiceHandover.....type reference OCTET STRING
DEFINED in MAP-MS-DataTypes : 500
USED in MAP-MS-DataTypes : 419 480

rand.....identifier of RAND
DEFINED in MAP-MS-DataTypes : 314

rand.....identifier of RAND
DEFINED in MAP-MS-DataTypes : 320

RAND.....type reference OCTET STRING
DEFINED in MAP-MS-DataTypes : 342
USED in MAP-MS-DataTypes : 314 320 668

rand.....identifier of RAND
DEFINED in MAP-MS-DataTypes : 668

readyForSM.....value reference ReadyForSM, CHOICE VALUE
DEFINED in MAP-Protocol : 268

ReadyForSM.....type reference OPERATION
DEFINED in MAP-ShortMessageServic : 136
USED in MAP-Protocol : 93 268
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 : 196
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 : 205
USED in MAP-ShortMessageServic : 56 140
USED in MAP-SM-DataTypes : 25

reason.....identifier of CHOICE
DEFINED in TCAPMessages : 75

recall.....identifier of Named Number, 1
DEFINED in MAP-SS-DataTypes : 287

regionalSubscNotSupported.....identifier of Named Number, 3
DEFINED in MAP-MS-DataTypes : 1206

regionalSubscriptionData.....identifier of [10] ZoneCodeList
DEFINED in MAP-MS-DataTypes : 895

regionalSubscriptionIdentifier.....identifier of [5] ZoneCode
DEFINED in MAP-MS-DataTypes : 1215

regionalSubscriptionResponse.....identifier of [5] RegionalSubscriptionResponse
DEFINED in MAP-MS-DataTypes : 1197

RegionalSubscriptionResponse.....type reference ENUMERATED
DEFINED in MAP-MS-DataTypes : 1202
USED in MAP-MS-DataTypes : 1197 1262

regionalSubscriptionResponse.....identifier of [0] RegionalSubscriptionResponse
DEFINED in MAP-MS-DataTypes : 1262

registerCC-Entry.....value reference RegisterCC-Entry, CHOICE VALUE
DEFINED in MAP-Protocol : 256

RegisterCC-Entry.....type reference OPERATION
DEFINED in MAP-SupplementaryServi : 251
USED in MAP-Protocol : 80 256
USED in MAP-SupplementaryServi : 24

registerCC-EntryArg.....identifier of RegisterCC-EntryArg
DEFINED in MAP-SupplementaryServi : 253

RegisterCC-EntryArg.....type reference SEQUENCE
DEFINED in MAP-SS-DataTypes : 305
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 : 324
USED in MAP-SupplementaryServi : 71 255

USED in MAP-SS-DataTypes : 38

registerPassword.....value reference RegisterPassword, CHOICE VALUE
DEFINED in MAP-Protocol : 254

RegisterPassword.....type reference OPERATION
DEFINED in MAP-SupplementaryServi : 217
USED in MAP-Protocol : 77 254
USED in MAP-SupplementaryServi : 21

registerSS.....value reference RegisterSS, CHOICE VALUE
DEFINED in MAP-Protocol : 245

RegisterSS.....type reference OPERATION
DEFINED in MAP-SupplementaryServi : 87
USED in MAP-Protocol : 69 245
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 : 72
USED in MAP-SupplementaryServi : 60 89
USED in MAP-SS-DataTypes : 14

registrationAllCF-Barred.....identifier of Named Number, 24
DEFINED in MAP-MS-DataTypes : 946

registrationCFNotToHPLMN-Barred.....identifier of Named Number, 25
DEFINED in MAP-MS-DataTypes : 947

registrationInternationalCF-Barred.....identifier of Named Number, 28
DEFINED in MAP-MS-DataTypes : 950

registrationInterzonalCFNotToHPLMN-Barreidentifier of Named Number, 27
DEFINED in MAP-MS-DataTypes : 949

registrationInterzonalCF-Barred.....identifier of Named Number, 26
DEFINED in MAP-MS-DataTypes : 948

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 : 402

releaseCall.....identifier of Named Number, 1
DEFINED in MAP-MS-DataTypes : 1452

releaseGroupCall.....identifier of [2] NULL
DEFINED in MAP-GR-DataTypes : 89

releaseTransaction.....identifier of Named Number, 1
DEFINED in MAP-MS-DataTypes : 804

releaseTransaction.....identifier of Named Number, 1
DEFINED in MAP-MS-DataTypes : 1509

RelocationNumber.....type reference SEQUENCE
DEFINED in MAP-MS-DataTypes : 618
USED in MAP-MS-DataTypes : 614

relocationNumberList.....identifier of [1] RelocationNumberList
DEFINED in MAP-MS-DataTypes : 520

RelocationNumberList.....type reference SEQUENCE OF
DEFINED in MAP-MS-DataTypes : 613
USED in MAP-MS-DataTypes : 520

remoteUserFree.....value reference RemoteUserFree, CHOICE VALUE
DEFINED in MAP-Protocol : 238

RemoteUserFree.....type reference OPERATION
DEFINED in MAP-CallHandlingOperat : 179
USED in MAP-Protocol : 61 238
USED in MAP-CallHandlingOperat : 20

remoteUserFreeArg.....identifier of RemoteUserFreeArg

DEFINED in MAP-CallHandlingOperat : 181

RemoteUserFreeArg.....type reference SEQUENCE
DEFINED in MAP-CH-DataTypes : 385
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
DEFINED in MAP-CH-DataTypes : 395
USED in MAP-CallHandlingOperat : 69 183
USED in MAP-CH-DataTypes : 32

replaceB-Number.....identifier of [4] NULL
DEFINED in MAP-CH-DataTypes : 390

ReportingState.....type reference ENUMERATED
DEFINED in MAP-CH-DataTypes : 322
USED in MAP-CH-DataTypes : 318

reportSM-DeliveryStatus.....value reference ReportSM-DeliveryStatus, CHOICE VALUE
DEFINED in MAP-Protocol : 265

ReportSM-DeliveryStatus.....type reference OPERATION
DEFINED in MAP-ShortMessageServic : 111
USED in MAP-Protocol : 90 265
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 : 171
USED in MAP-ShortMessageServic : 52 115
USED in MAP-SM-DataTypes : 21

request.....identifier of Named Number, 0
DEFINED in MAP-SS-DataTypes : 286

requestedBasicServiceViolatesCUG-Constraidentifier of Named Number, 5
DEFINED in MAP-ER-DataTypes : 125

requestedCAMEL-SubscriptionInfo.....identifier of [3] RequestedCAMEL-SubscriptionInfo
DEFINED in MAP-MS-DataTypes : 1798

RequestedCAMEL-SubscriptionInfo.....type reference ENUMERATED
DEFINED in MAP-MS-DataTypes : 1804
USED in MAP-MS-DataTypes : 1798 1893

requestedCamel-SubscriptionInfo.....identifier of [0] RequestedCAMEL-SubscriptionInfo
DEFINED in MAP-MS-DataTypes : 1893

requestedInfo.....identifier of [2] RequestedInfo
DEFINED in MAP-MS-DataTypes : 1681

RequestedInfo.....type reference SEQUENCE
DEFINED in MAP-MS-DataTypes : 1696
USED in MAP-MS-DataTypes : 1681 1764

requestedInfo.....identifier of [1] RequestedInfo
DEFINED in MAP-MS-DataTypes : 1764

requestedSS-Info.....identifier of [1] SS-ForBS-Code
DEFINED in MAP-MS-DataTypes : 1796

requestedSubscriptionInfo.....identifier of [1] RequestedSubscriptionInfo
DEFINED in MAP-MS-DataTypes : 1779

RequestedSubscriptionInfo.....type reference SEQUENCE
DEFINED in MAP-MS-DataTypes : 1795
USED in MAP-MS-DataTypes : 1779

reset.....value reference Reset, CHOICE VALUE
DEFINED in MAP-Protocol : 216

Reset.....type reference OPERATION
DEFINED in MAP-MobileServiceOpera : 416
USED in MAP-Protocol : 27 216
USED in MAP-MobileServiceOpera : 56

resetArg.....identifier of ResetArg
DEFINED in MAP-MobileServiceOpera : 418

ResetArg.....type reference SEQUENCE
DEFINED in MAP-MS-DataTypes : 1629
USED in MAP-MobileServiceOpera : 140 418
USED in MAP-MS-DataTypes : 87

resourceLimitation.....value reference ResourceLimitation, CHOICE VALUE
DEFINED in MAP-Protocol : 333

ResourceLimitation.....type reference ERROR
DEFINED in MAP-Errors : 192
USED in MAP-Protocol : 157 333
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 : 194

ResourceLimitationParam.....type reference SEQUENCE
DEFINED in MAP-ER-DataTypes : 318
USED in MAP-Errors : 136 194
USED in MAP-ER-DataTypes : 44

responseTime.....identifier of [3] ResponseTime
DEFINED in MAP-LCS-DataTypes : 144

ResponseTime.....type reference SEQUENCE
DEFINED in MAP-LCS-DataTypes : 162
USED in MAP-LCS-DataTypes : 21 144

responseTimeCategory.....identifier of ResponseTimeCategory
DEFINED in MAP-LCS-DataTypes : 163

ResponseTimeCategory.....type reference ENUMERATED
DEFINED in MAP-LCS-DataTypes : 167
USED in MAP-LCS-DataTypes : 163

restoreData.....value reference RestoreData, CHOICE VALUE
DEFINED in MAP-Protocol : 219

RestoreData.....type reference OPERATION
DEFINED in MAP-MobileServiceOpera : 422
USED in MAP-Protocol : 29 219
USED in MAP-MobileServiceOpera : 58

restoreDataArg.....identifier of RestoreDataArg
DEFINED in MAP-MobileServiceOpera : 424

RestoreDataArg.....type reference SEQUENCE
DEFINED in MAP-MS-DataTypes : 1634
USED in MAP-MobileServiceOpera : 141 424
USED in MAP-MS-DataTypes : 88

restoreDataRes.....identifier of RestoreDataRes
DEFINED in MAP-MobileServiceOpera : 426

RestoreDataRes.....type reference SEQUENCE
DEFINED in MAP-MS-DataTypes : 1641
USED in MAP-MobileServiceOpera : 142 426
USED in MAP-MS-DataTypes : 89

restrictedArea.....identifier of Named Number, 2
DEFINED in MAP-MS-DataTypes : 1757

restrictedArea.....identifier of Named Number, 1

DEFINED in MAP-ER-DataTypes : 250

result-RR.....identifier of SEQUENCE
DEFINED in TCAPMessages : 146

resumeCallHandling.....value reference ResumeCallHandling, CHOICE VALUE
DEFINED in MAP-Protocol : 233

ResumeCallHandling.....type reference OPERATION
DEFINED in MAP-CallHandlingOperat : 118
USED in MAP-Protocol : 56 233
USED in MAP-CallHandlingOperat : 15

resumeCallHandlingArg.....identifier of ResumeCallHandlingArg
DEFINED in MAP-CallHandlingOperat : 120

ResumeCallHandlingArg.....type reference SEQUENCE
DEFINED in MAP-CH-DataTypes : 224
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 : 252
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 : 661

Re-synchronisationInfo.....type reference SEQUENCE
DEFINED in MAP-MS-DataTypes : 667
USED in MAP-MS-DataTypes : 661

RNCId.....type reference OCTET STRING
DEFINED in MAP-MS-DataTypes : 601
USED in MAP-MS-DataTypes : 466 567

roamingNotAllowed.....value reference RoamingNotAllowed, CHOICE VALUE
DEFINED in MAP-Protocol : 347

RoamingNotAllowed.....type reference ERROR

DEFINED in MAP-Errors : 223

USED in MAP-Protocol : 125 347
USED in MAP-MobileServiceOpera : 89 186 229
USED in MAP-Errors : 29

roamingNotAllowedCause.....identifier of RoamingNotAllowedCause
DEFINED in MAP-ER-DataTypes : 92

RoamingNotAllowedCause.....type reference ENUMERATED
DEFINED in MAP-ER-DataTypes : 96
USED in MAP-ER-DataTypes : 92

roamingNotAllowedParam.....identifier of RoamingNotAllowedParam
DEFINED in MAP-Errors : 225

RoamingNotAllowedParam.....type reference SEQUENCE
DEFINED in MAP-ER-DataTypes : 91
USED in MAP-Errors : 117 225
USED in MAP-ER-DataTypes : 14

roamingNumber.....identifier of ISDN-AddressString
DEFINED in MAP-CH-DataTypes : 184

roamingNumber.....identifier of ISDN-AddressString
DEFINED in MAP-CH-DataTypes : 220

roamingOutsidePLMNICountryIC-CallsBarredidentifier of Named Number, 21
DEFINED in MAP-MS-DataTypes : 943

roamingOutsidePLMNIC-CallsBarred.....identifier of Named Number, 20
DEFINED in MAP-MS-DataTypes : 942

roamingOutsidePLMNOG-CallsBarred.....identifier of Named Number, 18
DEFINED in MAP-MS-DataTypes : 940

roamingOutsidePLMN-Barred.....identifier of Named Number, 22
DEFINED in MAP-MS-DataTypes : 944

roamingOutsidePLMN-CountryBarred.....identifier of Named Number, 23
DEFINED in MAP-MS-DataTypes : 945

roamingRestrictedInSgsnDueToUnsupportedFidentifier of [23] NULL
DEFINED in MAP-MS-DataTypes : 696

roamingRestrictedInSgsnDueToUnsuppportedidentifier of [11] NULL
DEFINED in MAP-MS-DataTypes : 1222

roamingRestrictionDueToUnsupportedFeaturidentifier of [9] NULL
DEFINED in MAP-MS-DataTypes : 894

roamingRestrictionDueToUnsupportedFeaturidentifier of [4] NULL
DEFINED in MAP-MS-DataTypes : 1214

routeSelectFailure.....identifier of Named Number, 4
DEFINED in MAP-MS-DataTypes : 1370

RoutingInfo.....type reference CHOICE
DEFINED in MAP-CH-DataTypes : 183
USED in MAP-CH-DataTypes : 263

routingInfo.....identifier of RoutingInfo
DEFINED in MAP-CH-DataTypes : 263

routingInfoForLCS-Arg.....identifier of RoutingInfoForLCS-Arg
DEFINED in MAP-LocationServiceOpe : 54

RoutingInfoForLCS-Arg.....type reference SEQUENCE
DEFINED in MAP-LCS-DataTypes : 55
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 : 61
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 : 356

ruf-Outcome.....identifier of [0] RUF-Outcome
DEFINED in MAP-CH-DataTypes : 396

RUF-Outcome.....type reference ENUMERATED
DEFINED in MAP-CH-DataTypes : 400
USED in MAP-CH-DataTypes : 396

sai-Present.....identifier of [9] NULL
DEFINED in MAP-MS-DataTypes : 1717

sc-AddressNotIncluded.....identifier of Named Number, 0
DEFINED in MAP-SM-DataTypes : 189

sc-Congestion.....identifier of Named Number, 4
DEFINED in MAP-ER-DataTypes : 145

segmentationProhibited.....identifier of NULL
DEFINED in MAP-MS-DataTypes : 287

segmentationProhibited.....identifier of NULL
DEFINED in MAP-MS-DataTypes : 659

selectedGSM-Algorithm.....identifier of [2] SelectedGSM-Algorithm
DEFINED in MAP-MS-DataTypes : 581

SelectedGSM-Algorithm.....type reference OCTET STRING
DEFINED in MAP-MS-DataTypes : 587
USED in MAP-MS-DataTypes : 581

selectedLSA-Id.....identifier of [5] LSAIdentity
DEFINED in MAP-MS-DataTypes : 1713

selectedRab-Id.....identifier of [4] RAB-Id
DEFINED in MAP-MS-DataTypes : 569

selectedRab-Id.....identifier of [4] RAB-Id
DEFINED in MAP-MS-DataTypes : 583

selectedUMTS-Algorithms.....identifier of [5] SelectedUMTS-Algorithms
DEFINED in MAP-MS-DataTypes : 524

SelectedUMTS-Algorithms.....type reference SEQUENCE
DEFINED in MAP-MS-DataTypes : 529
USED in MAP-MS-DataTypes : 524 580

selectedUMTS-Algorithms.....identifier of [1] SelectedUMTS-Algorithms
DEFINED in MAP-MS-DataTypes : 580

sendAuthenticationInfo.....value reference SendAuthenticationInfo, CHOICE VALUE
DEFINED in MAP-Protocol : 199

SendAuthenticationInfo.....type reference OPERATION
DEFINED in MAP-MobileServiceOpera : 350
USED in MAP-Protocol : 22 199
USED in MAP-MobileServiceOpera : 45

sendAuthenticationInfoArg.....identifier of SendAuthenticationInfoArg
DEFINED in MAP-MobileServiceOpera : 352

SendAuthenticationInfoArg.....type reference SEQUENCE
DEFINED in MAP-MS-DataTypes : 656
USED in MAP-MobileServiceOpera : 131 352
USED in MAP-MS-DataTypes : 43

sendAuthenticationInfoRes.....identifier of SendAuthenticationInfoRes
DEFINED in MAP-MobileServiceOpera : 359

SendAuthenticationInfoRes.....type reference [3] SEQUENCE
DEFINED in MAP-MS-DataTypes : 672
USED in MAP-MobileServiceOpera : 132 359
USED in MAP-MS-DataTypes : 44

sendEndSignal.....value reference SendEndSignal, CHOICE VALUE
DEFINED in MAP-Protocol : 190

SendEndSignal.....type reference OPERATION
DEFINED in MAP-MobileServiceOpera : 322
USED in MAP-Protocol : 18 190
USED in MAP-MobileServiceOpera : 39

sendEndSignal-Arg.....identifier of SendEndSignal-Arg
DEFINED in MAP-MobileServiceOpera : 324

SendEndSignal-Arg.....type reference [3] SEQUENCE
DEFINED in MAP-MS-DataTypes : 592
USED in MAP-MobileServiceOpera : 127 324
USED in MAP-MS-DataTypes : 38

sendEndSignal-Res.....identifier of SendEndSignal-Res
DEFINED in MAP-MobileServiceOpera : 326

SendEndSignal-Res.....type reference SEQUENCE
DEFINED in MAP-MS-DataTypes : 597
USED in MAP-MobileServiceOpera : 128 326
USED in MAP-MS-DataTypes : 39

sendGroupCallEndSignal.....value reference SendGroupCallEndSignal, CHOICE VALUE
DEFINED in MAP-Protocol : 295

SendGroupCallEndSignal.....type reference OPERATION
DEFINED in MAP-Group-Call-Operati : 56
USED in MAP-Protocol : 102 295
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 : 184

SendIdentification.....type reference OPERATION
DEFINED in MAP-MobileServiceOpera : 209
USED in MAP-Protocol : 15 184
USED in MAP-MobileServiceOpera : 18

sendIdentificationArg.....identifier of SendIdentificationArg
DEFINED in MAP-MobileServiceOpera : 211

SendIdentificationArg.....type reference SEQUENCE
DEFINED in MAP-MS-DataTypes : 280
USED in MAP-MobileServiceOpera : 119 211
USED in MAP-MS-DataTypes : 22

sendIdentificationRes.....identifier of SendIdentificationRes
DEFINED in MAP-MobileServiceOpera : 213

SendIdentificationRes.....type reference [3] SEQUENCE
DEFINED in MAP-MS-DataTypes : 291
USED in MAP-MobileServiceOpera : 120 213
USED in MAP-MS-DataTypes : 23

sendIMSI.....value reference SendIMSI, CHOICE VALUE
DEFINED in MAP-Protocol : 226

SendIMSI.....type reference OPERATION

DEFINED in MAP-OperationAndMainte : 77

USED in MAP-Protocol : 48 226
USED in MAP-OperationAndMainte : 15

sendRoutingInfo.....value reference SendRoutingInfo, CHOICE VALUE
DEFINED in MAP-Protocol : 231

SendRoutingInfo.....type reference OPERATION
DEFINED in MAP-CallHandlingOperat : 80
USED in MAP-Protocol : 54 231
USED in MAP-CallHandlingOperat : 13

sendRoutingInfoArg.....identifier of SendRoutingInfoArg
DEFINED in MAP-CallHandlingOperat : 83

SendRoutingInfoArg.....type reference SEQUENCE
DEFINED in MAP-CH-DataTypes : 94
USED in MAP-CallHandlingOperat : 54 83
USED in MAP-CH-DataTypes : 14

sendRoutingInfoForGprs.....value reference SendRoutingInfoForGprs, CHOICE VALUE
DEFINED in MAP-Protocol : 306

SendRoutingInfoForGprs.....type reference OPERATION
DEFINED in MAP-MobileServiceOpera : 435
USED in MAP-Protocol : 34 306
USED in MAP-MobileServiceOpera : 61

sendRoutingInfoForGprsArg.....identifier of SendRoutingInfoForGprsArg
DEFINED in MAP-MobileServiceOpera : 437

SendRoutingInfoForGprsArg.....type reference SEQUENCE
DEFINED in MAP-MS-DataTypes : 1585
USED in MAP-MobileServiceOpera : 153 437
USED in MAP-MS-DataTypes : 116

sendRoutingInfoForGprsRes.....identifier of SendRoutingInfoForGprsRes
DEFINED in MAP-MobileServiceOpera : 439

SendRoutingInfoForGprsRes.....type reference SEQUENCE
DEFINED in MAP-MS-DataTypes : 1592
USED in MAP-MobileServiceOpera : 154 439
USED in MAP-MS-DataTypes : 117

sendRoutingInfoForLCS.....value reference SendRoutingInfoForLCS, CHOICE VALUE
DEFINED in MAP-Protocol : 319

SendRoutingInfoForLCS.....type reference OPERATION
DEFINED in MAP-LocationServiceOpe : 52
USED in MAP-Protocol : 109 319
USED in MAP-LocationServiceOpe : 14

sendRoutingInfoForSM.....value reference SendRoutingInfoForSM, CHOICE VALUE
DEFINED in MAP-Protocol : 262

SendRoutingInfoForSM.....type reference OPERATION
DEFINED in MAP-ShortMessageServic : 66
USED in MAP-Protocol : 87 262
USED in MAP-ShortMessageServic : 13

sendRoutingInfoRes.....identifier of SendRoutingInfoRes
DEFINED in MAP-CallHandlingOperat : 85

SendRoutingInfoRes.....type reference [3] SEQUENCE
DEFINED in MAP-CH-DataTypes : 145
USED in MAP-CallHandlingOperat : 55 85
USED in MAP-CH-DataTypes : 15

sendSubscriberData.....identifier of [0] NULL
DEFINED in MAP-MS-DataTypes : 228

serviceCentreAddress.....identifier of [2] AddressString
DEFINED in MAP-SM-DataTypes : 55

serviceCentreAddress.....identifier of AddressString
DEFINED in MAP-SM-DataTypes : 145

serviceCentreAddress.....identifier of AddressString
DEFINED in MAP-SM-DataTypes : 179

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 : 905

serviceIndicator.....identifier of [2] ServiceIndicator
DEFINED in MAP-SS-DataTypes : 313

ServiceIndicator.....type reference BIT STRING
DEFINED in MAP-SS-DataTypes : 318
USED in MAP-SS-DataTypes : 313

serviceKey.....identifier of [1] ServiceKey
DEFINED in MAP-MS-DataTypes : 795

serviceKey.....identifier of ServiceKey
DEFINED in MAP-MS-DataTypes : 1301

serviceKey.....identifier of ServiceKey
DEFINED in MAP-MS-DataTypes : 1358

ServiceKey.....type reference INTEGER
DEFINED in MAP-MS-DataTypes : 1365
USED in MAP-MS-DataTypes : 69 795 1301 1358 1492 1517 1566 1924

serviceKey.....identifier of [1] ServiceKey
DEFINED in MAP-MS-DataTypes : 1492

serviceKey.....identifier of ServiceKey
DEFINED in MAP-MS-DataTypes : 1517

serviceKey.....identifier of ServiceKey
DEFINED in MAP-MS-DataTypes : 1566

serviceKey.....identifier of ServiceKey
DEFINED in MAP-MS-DataTypes : 1924

setReportingState.....value reference SetReportingState, CHOICE VALUE
DEFINED in MAP-Protocol : 236

SetReportingState.....type reference OPERATION
DEFINED in MAP-CallHandlingOperat : 153
USED in MAP-Protocol : 59 236
USED in MAP-CallHandlingOperat : 18

setReportingStateArg.....identifier of SetReportingStateArg
DEFINED in MAP-CallHandlingOperat : 155

SetReportingStateArg.....type reference SEQUENCE
DEFINED in MAP-CH-DataTypes : 315
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 : 330
USED in MAP-CallHandlingOperat : 65 157
USED in MAP-CH-DataTypes : 28

sgsn-Address.....identifier of GSN-Address
DEFINED in MAP-MS-DataTypes : 383

sgsn-Address.....identifier of [0] GSN-Address
DEFINED in MAP-MS-DataTypes : 1593

sgsn-Address.....identifier of [1] GSN-Address
DEFINED in MAP-MS-DataTypes : 1617

sgsn-CAMEL-SubscriptionInfo.....identifier of [17] SGSN-CAMEL-SubscriptionInfo
DEFINED in MAP-MS-DataTypes : 706

SGSN-CAMEL-SubscriptionInfo.....type reference SEQUENCE
DEFINED in MAP-MS-DataTypes : 768
USED in MAP-MS-DataTypes : 706

sgsn-Capability.....identifier of [0] SGSN-Capability
DEFINED in MAP-MS-DataTypes : 386

SGSN-Capability.....type reference SEQUENCE

DEFINED in MAP-MS-DataTypes : 389
USED in MAP-MS-DataTypes : 386

sgsn-Number.....identifier of [1] ISDN-AddressString
DEFINED in MAP-MS-DataTypes : 270

sgsn-Number.....identifier of ISDN-AddressString
DEFINED in MAP-MS-DataTypes : 382

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 : 412

ShortTermDenial.....type reference ERROR
DEFINED in MAP-Errors : 387
USED in MAP-Protocol : 159 412
USED in MAP-SupplementaryServi : 53 264
USED in MAP-Errors : 73

shortTermDenialParam.....identifier of ShortTermDenialParam
DEFINED in MAP-Errors : 389

ShortTermDenialParam.....type reference SEQUENCE
DEFINED in MAP-ER-DataTypes : 330
USED in MAP-Errors : 139 389
USED in MAP-ER-DataTypes : 47

signalInfo.....identifier of SignalInfo
DEFINED in MAP-CommonDataTypes : 197

SignalInfo.....type reference OCTET STRING
DEFINED in MAP-CommonDataTypes : 204
USED in MAP-CommonDataTypes : 24 197 223
USED in MAP-SM-DataTypes : 33 109 115 122 128
USED in MAP-ER-DataTypes : 73 151

signalInfo.....identifier of SignalInfo
DEFINED in MAP-CommonDataTypes : 223

signalInfo.....identifier of LongSignalInfo
DEFINED in MAP-CommonDataTypes : 240

sIWFSNumber.....identifier of [0] ISDN-AddressString
DEFINED in MAP-CH-DataTypes : 300

sIWFSSignallingModify.....value reference SIWFSSignallingModify, CHOICE VALUE
DEFINED in MAP-Protocol : 235

SIWFSSignallingModify.....type reference OPERATION
DEFINED in MAP-CallHandlingOperat : 141
USED in MAP-Protocol : 58 235
USED in MAP-CallHandlingOperat : 17

sIWFSSignallingModifyArg.....identifier of SIWFSSignallingModifyArg
DEFINED in MAP-CallHandlingOperat : 143

SIWFSSignallingModifyArg.....type reference SEQUENCE
DEFINED in MAP-CH-DataTypes : 304
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 : 310
USED in MAP-CallHandlingOperat : 63 145
USED in MAP-CH-DataTypes : 26

slr-ArgExtensionContainer.....identifier of [7] SLR-ArgExtensionContainer
DEFINED in MAP-LCS-DataTypes : 294

SLR-ArgExtensionContainer.....type reference SEQUENCE

DEFINED in MAP-ExtensionDataTypes : 30

USED in MAP-LCS-DataTypes : 42 294
USED in MAP-ExtensionDataTypes : 17

slr-Arg-PCS-Extensions.....identifier of [1] SLR-Arg-PCS-Extensions
DEFINED in MAP-ExtensionDataTypes : 32

SLR-Arg-PCS-Extensions.....type reference SEQUENCE
DEFINED in MAP-ExtensionDataTypes : 65
USED in MAP-ExtensionDataTypes : 32

SMS-CAMEL-TDP-Data.....type reference SEQUENCE
DEFINED in MAP-MS-DataTypes : 1490
USED in MAP-MS-DataTypes : 1486

sms-CAMEL-TDP-DataList.....identifier of [0] SMS-CAMEL-TDP-DataList
DEFINED in MAP-MS-DataTypes : 1472

SMS-CAMEL-TDP-DataList.....type reference SEQUENCE OF
DEFINED in MAP-MS-DataTypes : 1485
USED in MAP-MS-DataTypes : 1472

sms-CollectedInfo.....identifier of Named Number, 1
DEFINED in MAP-MS-DataTypes : 1500

sms-CSI.....identifier of [1] SMS-CSI
DEFINED in MAP-MS-DataTypes : 770

sms-csi.....identifier of Named Number, 5
DEFINED in MAP-MS-DataTypes : 1234

sms-CSI.....identifier of [6] SMS-CSI
DEFINED in MAP-MS-DataTypes : 1274

SMS-CSI.....type reference SEQUENCE
DEFINED in MAP-MS-DataTypes : 1471
USED in MAP-MS-DataTypes : 770 1274 1848

sms-CSI.....identifier of Named Number, 5
DEFINED in MAP-MS-DataTypes : 1810

sms-CSI.....identifier of [10] SMS-CSI
DEFINED in MAP-MS-DataTypes : 1848

sms-TriggerDetectionPoint.....identifier of [0] SMS-TriggerDetectionPoint
DEFINED in MAP-MS-DataTypes : 1491

SMS-TriggerDetectionPoint.....type reference ENUMERATED
DEFINED in MAP-MS-DataTypes : 1499
USED in MAP-MS-DataTypes : 1491

sm-DeliveryFailure.....value reference SM-DeliveryFailure, CHOICE VALUE
DEFINED in MAP-Protocol : 419

SM-DeliveryFailure.....type reference ERROR
DEFINED in MAP-Errors : 405
USED in MAP-Protocol : 154 419
USED in MAP-ShortMessageServic : 38 91 108
USED in MAP-Errors : 78

sm-DeliveryFailureCause.....identifier of SM-DeliveryFailureCause
DEFINED in MAP-Errors : 407

SM-DeliveryFailureCause.....type reference SEQUENCE
DEFINED in MAP-ER-DataTypes : 149
USED in MAP-Errors : 109 407
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 : 166
USED in MAP-SM-DataTypes : 26 146 157

SM-EnumeratedDeliveryFailureCause.....type reference ENUMERATED
DEFINED in MAP-ER-DataTypes : 140
USED in MAP-ER-DataTypes : 150

sm-EnumeratedDeliveryFailureCause.....identifier of SM-EnumeratedDeliveryFailureCause
DEFINED in MAP-ER-DataTypes : 150

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
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 : 222

solsaSupportIndicator.....identifier of NULL
DEFINED in MAP-MS-DataTypes : 390

specificCSIDeletedList.....identifier of [14] SpecificCSI-Withdraw
DEFINED in MAP-MS-DataTypes : 1853

specificCSI-Withdraw.....identifier of [15] SpecificCSI-Withdraw
DEFINED in MAP-MS-DataTypes : 1226

SpecificCSI-Withdraw.....type reference BIT STRING
DEFINED in MAP-MS-DataTypes : 1228
USED in MAP-MS-DataTypes : 1226 1853

sres.....identifier of SRES
DEFINED in MAP-MS-DataTypes : 315

SRES.....type reference OCTET STRING
DEFINED in MAP-MS-DataTypes : 344
USED in MAP-MS-DataTypes : 315

ss-AccessBarred.....identifier of Named Number, 5
DEFINED in MAP-MS-DataTypes : 933

ss-CamelData.....identifier of SS-CamelData
DEFINED in MAP-MS-DataTypes : 1308

SS-CamelData.....type reference SEQUENCE
DEFINED in MAP-MS-DataTypes : 1317
USED in MAP-MS-DataTypes : 1308

ss-Code.....identifier of SS- Code

DEFINED in MAP-SupplementaryServi : 219

ss-Code.....identifier of SS-Code
DEFINED in MAP-MS-DataTypes : 980

ss-Code.....identifier of SS-Code
DEFINED in MAP-MS-DataTypes : 1039

ss-Code.....identifier of SS-Code
DEFINED in MAP-MS-DataTypes : 1110

ss-Code.....identifier of SS-Code
DEFINED in MAP-MS-DataTypes : 1123

ss-Code.....identifier of SS-Code
DEFINED in MAP-MS-DataTypes : 1179

ss-Code.....identifier of [0] SS-Code
DEFINED in MAP-MS-DataTypes : 1872

ss-Code.....identifier of [0] SS-Code
DEFINED in MAP-MS-DataTypes : 1883

ss-Code.....identifier of [0] SS-Code
DEFINED in MAP-MS-DataTypes : 1943

ss-Code.....identifier of [0] SS-Code
DEFINED in MAP-MS-DataTypes : 1950

ss-Code.....identifier of [0] SS-Code
DEFINED in MAP-CommonDataTypes : 457

ss-Code.....identifier of SS-Code
DEFINED in MAP-SS-DataTypes : 73

ss-Code.....identifier of SS-Code
DEFINED in MAP-SS-DataTypes : 91

ss-Code.....identifier of SS-Code
DEFINED in MAP-SS-DataTypes : 149

ss-Code.....identifier of SS-Code
DEFINED in MAP-SS-DataTypes : 162

ss-Code.....identifier of SS-Code
DEFINED in MAP-SS-DataTypes : 185

ss-Code.....identifier of [0] SS-Code
DEFINED in MAP-SS-DataTypes : 306

ss-Code.....identifier of [0] SS-Code
DEFINED in MAP-SS-DataTypes : 329

ss-Code.....identifier of [0] SS-Code
DEFINED in MAP-SS-DataTypes : 334

SS-Code.....type reference OCTET STRING
DEFINED in MAP-SS-Code : 11
USED in MAP-SupplementaryServi : 78 219
USED in MAP-MS-DataTypes : 145 980 1039 1110 1123 1179 1323 1872 1883
1943 1950
USED in MAP-CommonDataTypes : 76 457
USED in MAP-SS-DataTypes : 65 73 91 149 162 185 257 272 306
329 334
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 84 87 90
93 96 99 102 104 107 110 112 114
117 119 121 123 125 128 130 132 136
137 138 139 140 141 142 143 144 145
146 147 148 149 150 151 153 156 159
161 163 166 168 171 173 175 178
USED in MAP-ER-DataTypes : 80 129

ss-Code.....identifier of [1] SS-Code
DEFINED in MAP-ER-DataTypes : 129

ss-csi.....identifier of Named Number, 1
DEFINED in MAP-MS-DataTypes : 1230

ss-CSI.....identifier of [2] SS-CSI

DEFINED in MAP-MS-DataTypes : 1270

SS-CSI.....type reference SEQUENCE
DEFINED in MAP-MS-DataTypes : 1307
USED in MAP-MS-DataTypes : 68 1270 1849

ss-CSI.....identifier of Named Number, 6
DEFINED in MAP-MS-DataTypes : 1811

ss-CSI.....identifier of [11] SS-CSI
DEFINED in MAP-MS-DataTypes : 1849

ss-Data.....identifier of [3] Ext-SS-Data
DEFINED in MAP-MS-DataTypes : 975

ss-Data.....identifier of [3] SS-Data
DEFINED in MAP-SS-DataTypes : 88

SS-Data.....type reference SEQUENCE
DEFINED in MAP-SS-DataTypes : 161
USED in MAP-SS-DataTypes : 33 88

ss-ErrorStatus.....value reference SS-ErrorStatus, CHOICE VALUE
DEFINED in MAP-Protocol : 402

SS-ErrorStatus.....type reference ERROR
DEFINED in MAP-Errors : 353
USED in MAP-Protocol : 144 402
USED in MAP-MobileServiceOpera : 101 290
USED in MAP-SupplementaryServi : 41 101 118 135 155 262 279
USED in MAP-Errors : 64

ss-Event.....identifier of [2] SS-Code
DEFINED in MAP-SS-DataTypes : 272

ss-EventList.....identifier of SS-EventList
DEFINED in MAP-MS-DataTypes : 1318

SS-EventList.....type reference SEQUENCE OF
DEFINED in MAP-MS-DataTypes : 1323
USED in MAP-MS-DataTypes : 1318

ss-EventSpecification.....identifier of [3] SS-EventSpecification
DEFINED in MAP-SS-DataTypes : 278

SS-EventSpecification.....type reference SEQUENCE OF
DEFINED in MAP-SS-DataTypes : 300
USED in MAP-SS-DataTypes : 278

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 : 184
USED in MAP-SupplementaryServi : 62 106 123 143 162
USED in MAP-MS-DataTypes : 139 1796
USED in MAP-SS-DataTypes : 18

ss-Incompatibility.....value reference SS-Incompatibility, CHOICE VALUE
DEFINED in MAP-Protocol : 405

SS-Incompatibility.....type reference ERROR
DEFINED in MAP-Errors : 370
USED in MAP-Protocol : 147 405
USED in MAP-MobileServiceOpera : 103 291
USED in MAP-SupplementaryServi : 44 102 137 263
USED in MAP-Errors : 67

ss-IncompatibilityCause.....identifier of SS-IncompatibilityCause
DEFINED in MAP-Errors : 372

SS-IncompatibilityCause.....type reference SEQUENCE
DEFINED in MAP-ER-DataTypes : 128

USED in MAP-Errors : 107 372

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 : 85
USED in MAP-SupplementaryServi : 61 91 108 125 145
USED in MAP-SS-DataTypes : 15 262

ss-InfoFor-CSE.....identifier of [0] Ext-SS-InfoFor-CSE
DEFINED in MAP-MS-DataTypes : 1866

SS-InfoList.....type reference SEQUENCE OF
DEFINED in MAP-SS-DataTypes : 261
USED in MAP-SS-DataTypes : 27

ss-InvocationNotification.....value reference SS-InvocationNotification, CHOICE VALUE
DEFINED in MAP-Protocol : 289

SS-InvocationNotification.....type reference OPERATION
DEFINED in MAP-SupplementaryServi : 240
USED in MAP-Protocol : 79 289
USED in MAP-SupplementaryServi : 23

ss-InvocationNotificationArg.....identifier of SS-InvocationNotificationArg
DEFINED in MAP-SupplementaryServi : 242

SS-InvocationNotificationArg.....type reference SEQUENCE
DEFINED in MAP-SS-DataTypes : 269
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 : 295
USED in MAP-SupplementaryServi : 69 244
USED in MAP-SS-DataTypes : 35

ss-List.....identifier of [3] SS-List
DEFINED in MAP-MS-DataTypes : 1195

ss-List.....identifier of [2] SS-List
DEFINED in MAP-MS-DataTypes : 1213

ss-List.....identifier of [1] SS-List
DEFINED in MAP-CH-DataTypes : 154

SS-List.....type reference SEQUENCE OF
DEFINED in MAP-SS-DataTypes : 256
USED in MAP-MS-DataTypes : 138 1195 1213
USED in MAP-CH-DataTypes : 57 154
USED in MAP-SS-DataTypes : 26

ss-NotAvailable.....value reference SS-NotAvailable, CHOICE VALUE
DEFINED in MAP-Protocol : 403

SS-NotAvailable.....type reference ERROR
DEFINED in MAP-Errors : 358
USED in MAP-Protocol : 145 403
USED in MAP-MobileServiceOpera : 102 272
USED in MAP-SupplementaryServi : 42 173
USED in MAP-Errors : 65

ss-NotAvailableParam.....identifier of SS-NotAvailableParam
DEFINED in MAP-Errors : 360

SS-NotAvailableParam.....type reference SEQUENCE
DEFINED in MAP-ER-DataTypes : 295
USED in MAP-Errors : 149 360

USED in MAP-ER-DataTypes : 57

ss-Status.....identifier of SS-Status
DEFINED in MAP-Errors : 355

ss-Status.....identifier of [4] Ext-SS-Status
DEFINED in MAP-MS-DataTypes : 990

ss-Status.....identifier of [4] Ext-SS-Status
DEFINED in MAP-MS-DataTypes : 1049

ss-Status.....identifier of [4] Ext-SS-Status
DEFINED in MAP-MS-DataTypes : 1111

ss-Status.....identifier of Ext-SS-Status
DEFINED in MAP-MS-DataTypes : 1124

ss-Status.....identifier of Ext-SS-Status
DEFINED in MAP-MS-DataTypes : 1180

ss-Status.....identifier of [2] Ext-SS-Status
DEFINED in MAP-MS-DataTypes : 1874

ss-Status.....identifier of [2] Ext-SS-Status
DEFINED in MAP-MS-DataTypes : 1885

ss-Status.....identifier of [1] Ext-SS-Status
DEFINED in MAP-CommonDataTypes : 458

ss-Status.....identifier of [4] SS-Status
DEFINED in MAP-SS-DataTypes : 101

SS-Status.....type reference OCTET STRING
DEFINED in MAP-SS-DataTypes : 109
USED in MAP-Errors : 102 355
USED in MAP-SS-DataTypes : 16 101 158 163 191 216 335
USED in MAP-ER-DataTypes : 68 131

ss-Status.....identifier of [4] SS-Status
DEFINED in MAP-SS-DataTypes : 158

ss-Status.....identifier of [4] SS-Status
DEFINED in MAP-SS-DataTypes : 163

ss-Status.....identifier of SS-Status
DEFINED in MAP-SS-DataTypes : 191

ss-Status.....identifier of [0] SS-Status
DEFINED in MAP-SS-DataTypes : 216

ss-Status.....identifier of [1] SS-Status
DEFINED in MAP-SS-DataTypes : 335

ss-Status.....identifier of [4] SS-Status
DEFINED in MAP-ER-DataTypes : 131

ss-SubscriptionOption.....identifier of SS-SubscriptionOption
DEFINED in MAP-MS-DataTypes : 1112

ss-SubscriptionOption.....identifier of SS-SubscriptionOption
DEFINED in MAP-SS-DataTypes : 164

SS-SubscriptionOption.....type reference CHOICE
DEFINED in MAP-SS-DataTypes : 171
USED in MAP-MS-DataTypes : 137 1112
USED in MAP-SS-DataTypes : 17 164

ss-SubscriptionViolation.....value reference SS-SubscriptionViolation, CHOICE VALUE
DEFINED in MAP-Protocol : 404

SS-SubscriptionViolation.....type reference ERROR
DEFINED in MAP-Errors : 364
USED in MAP-Protocol : 146 404
USED in MAP-MobileServiceOpera : 104 289
USED in MAP-SupplementaryServi : 43 136 156 227
USED in MAP-Errors : 66

ss-SubscriptionViolationParam.....identifier of SS-SubscriptionViolationParam
DEFINED in MAP-Errors : 366

SS-SubscriptionViolationParam.....type reference SEQUENCE
DEFINED in MAP-ER-DataTypes : 299

USED in MAP-Errors : 150 366

USED in MAP-ER-DataTypes : 58

startMonitoring.....identifier of Named Number, 1
DEFINED in MAP-CH-DataTypes : 324

stateAttributes.....identifier of [5] StateAttributes
DEFINED in MAP-GR-DataTypes : 84

StateAttributes.....type reference SEQUENCE
DEFINED in MAP-GR-DataTypes : 119
USED in MAP-GR-DataTypes : 84

statusReport.....value reference StatusReport, CHOICE VALUE
DEFINED in MAP-Protocol : 237

StatusReport.....type reference OPERATION
DEFINED in MAP-CallHandlingOperat : 167
USED in MAP-Protocol : 60 237
USED in MAP-CallHandlingOperat : 19

statusReportArg.....identifier of StatusReportArg
DEFINED in MAP-CallHandlingOperat : 169

StatusReportArg.....type reference SEQUENCE
DEFINED in MAP-CH-DataTypes : 345
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 : 381
USED in MAP-CallHandlingOperat : 67 171
USED in MAP-CH-DataTypes : 30

stopMonitoring.....identifier of Named Number, 0
DEFINED in MAP-CH-DataTypes : 323

storedMSISDN.....identifier of ISDN-AddressString
DEFINED in MAP-SM-DataTypes : 172

storedMSISDN.....identifier of ISDN-AddressString
DEFINED in MAP-SM-DataTypes : 183

subBusyForMT-SMS-Param.....identifier of SubBusyForMT-SMS-Param
DEFINED in MAP-Errors : 402

SubBusyForMT-SMS-Param.....type reference SEQUENCE
DEFINED in MAP-ER-DataTypes : 307
USED in MAP-Errors : 133 402
USED in MAP-ER-DataTypes : 40

subscriberBusyForMT-SMS.....value reference SubscriberBusyForMT-SMS, CHOICE VALUE
DEFINED in MAP-Protocol : 418

SubscriberBusyForMT-SMS.....type reference ERROR
DEFINED in MAP-Errors : 400
USED in MAP-Protocol : 153 418
USED in MAP-ShortMessageServic : 37 107
USED in MAP-Errors : 77

SubscriberData.....type reference SEQUENCE
DEFINED in MAP-MS-DataTypes : 882
USED in MAP-MS-DataTypes : 59 690

subscriberDataStored.....identifier of [1] AgeIndicator
DEFINED in MAP-MS-DataTypes : 229

SubscriberId.....type reference CHOICE
DEFINED in MAP-CommonDataTypes : 313
USED in MAP-CommonDataTypes : 33

subscriberIdentity.....identifier of [0] SubscriberIdentity
DEFINED in MAP-MS-DataTypes : 1763

subscriberIdentity.....identifier of [0] SubscriberIdentity
DEFINED in MAP-MS-DataTypes : 1778

subscriberIdentity.....identifier of [0] SubscriberIdentity
DEFINED in MAP-MS-DataTypes : 1856

SubscriberIdentity.....type reference CHOICE
DEFINED in MAP-CommonDataTypes : 370
USED in MAP-MS-DataTypes : 177 1763 1778 1856
USED in MAP-CommonDataTypes : 42
USED in MAP-LCS-DataTypes : 33 57 62

subscriberInfo.....identifier of SubscriberInfo
DEFINED in MAP-MS-DataTypes : 1686

SubscriberInfo.....type reference SEQUENCE
DEFINED in MAP-MS-DataTypes : 1690
USED in MAP-MS-DataTypes : 97 1686 1770
USED in MAP-CH-DataTypes : 40 153

subscriberInfo.....identifier of SubscriberInfo
DEFINED in MAP-MS-DataTypes : 1770

subscriberInfo.....identifier of [7] SubscriberInfo
DEFINED in MAP-CH-DataTypes : 153

subscriberLocationReport.....value reference SubscriberLocationReport, CHOICE VALUE
DEFINED in MAP-Protocol : 320

SubscriberLocationReport.....type reference OPERATION
DEFINED in MAP-LocationServiceOpe : 84
USED in MAP-Protocol : 110 320
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 : 283
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 : 311
USED in MAP-LocationServiceOpe : 46 88
USED in MAP-LCS-DataTypes : 16

subscriberNotMemberOfCUG.....identifier of Named Number, 1
DEFINED in MAP-ER-DataTypes : 124

subscriberNotSC-Subscriber.....identifier of Named Number, 6
DEFINED in MAP-ER-DataTypes : 147

subscriberState.....identifier of [1] SubscriberState
DEFINED in MAP-MS-DataTypes : 1692

subscriberState.....identifier of [1] NULL
DEFINED in MAP-MS-DataTypes : 1698

SubscriberState.....type reference CHOICE
DEFINED in MAP-MS-DataTypes : 1748
USED in MAP-MS-DataTypes : 99 1692

subscriberStatus.....identifier of [3] SubscriberStatus
DEFINED in MAP-MS-DataTypes : 885

SubscriberStatus.....type reference ENUMERATED
DEFINED in MAP-MS-DataTypes : 904
USED in MAP-MS-DataTypes : 61 885

subscriptionWithdraw.....identifier of Named Number, 1
DEFINED in MAP-MS-DataTypes : 258

subsequentHandoverFailure.....value reference SubsequentHandoverFailure, CHOICE VALUE
DEFINED in MAP-Protocol : 360

SubsequentHandoverFailure.....type reference ERROR
DEFINED in MAP-Errors : 256
USED in MAP-Protocol : 131 360
USED in MAP-MobileServiceOpera : 92 346
USED in MAP-Errors : 37

success.....identifier of Named Number, 0
DEFINED in MAP-CH-DataTypes : 372

successfulTransfer.....identifier of Named Number, 2
DEFINED in MAP-SM-DataTypes : 169

SuperChargerInfo.....type reference CHOICE
DEFINED in MAP-MS-DataTypes : 227
USED in MAP-MS-DataTypes : 224 393

superChargerSupportedInHLR.....identifier of [27] AgeIndicator
DEFINED in MAP-MS-DataTypes : 703

superChargerSupportedInServingNetworkEntidentifier of [3] SuperChargerInfo
DEFINED in MAP-MS-DataTypes : 224

superChargerSupportedInServingNetworkEntidentifier of [2] SuperChargerInfo
DEFINED in MAP-MS-DataTypes : 393

supportedCamelPhases.....identifier of [0] SupportedCamelPhases
DEFINED in MAP-MS-DataTypes : 219

supportedCamelPhases.....identifier of [4] SupportedCamelPhases
DEFINED in MAP-MS-DataTypes : 395

supportedCamelPhases.....identifier of [6] SupportedCamelPhases
DEFINED in MAP-MS-DataTypes : 1198

SupportedCamelPhases.....type reference BIT STRING
DEFINED in MAP-MS-DataTypes : 1464
USED in MAP-MS-DataTypes : 73 219 395 1198 1790 1791 1929
USED in MAP-CH-DataTypes : 41 213 257

supportedCAMELPhases.....identifier of [5] SupportedCamelPhases
DEFINED in MAP-MS-DataTypes : 1929

supportedCamelPhases.....identifier of SupportedCamelPhases
DEFINED in MAP-CH-DataTypes : 257

supportedCamelPhasesInGMSC.....identifier of [15] SupportedCamelPhases
DEFINED in MAP-CH-DataTypes : 213

supportedCCBS-Phase.....identifier of [16] SupportedCCBS-Phase
DEFINED in MAP-CH-DataTypes : 112

SupportedCCBS-Phase.....type reference INTEGER
DEFINED in MAP-CH-DataTypes : 134
USED in MAP-CH-DataTypes : 112

supportedGADShapes.....identifier of [9] SupportedGADShapes
DEFINED in MAP-LCS-DataTypes : 86

SupportedGADShapes.....type reference BIT STRING
DEFINED in MAP-LCS-DataTypes : 174
USED in MAP-LCS-DataTypes : 23 86

supportedSGSN-CAMEL-Phases.....identifier of [6] SupportedCamelPhases
DEFINED in MAP-MS-DataTypes : 1791

supportedSGSN-CAMEL-Phases.....identifier of [5] NULL
DEFINED in MAP-MS-DataTypes : 1800

supportedVLR-CAMEL-Phases.....identifier of [5] SupportedCamelPhases
DEFINED in MAP-MS-DataTypes : 1790

supportedVLR-CAMEL-Phases.....identifier of [4] NULL
DEFINED in MAP-MS-DataTypes : 1799

suppressionOfAnnouncement.....identifier of [12] SuppressionOfAnnouncement
DEFINED in MAP-CH-DataTypes : 107

SuppressionOfAnnouncement.....type reference NULL
DEFINED in MAP-CH-DataTypes : 119
USED in MAP-CH-DataTypes : 21 107 205

suppressionOfAnnouncement.....identifier of [7] SuppressionOfAnnouncement
DEFINED in MAP-CH-DataTypes : 205

suppress-T-CSI.....identifier of NULL
DEFINED in MAP-CH-DataTypes : 258

suspended.....identifier of Named Number, 4
DEFINED in MAP-SS-DataTypes : 290

systemFailure.....value reference
SystemFailure, CHOICE VALUE

DEFINED in MAP-Protocol : 328

SystemFailure.....type reference ERROR
DEFINED in MAP-Errors : 164
USED in MAP-Protocol : 116 328
USED in MAP-MobileServiceOpera : 82 182 226 250 316 362 374 386 428 442 456 470
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 : 166

SystemFailureParam.....type reference CHOICE
DEFINED in MAP-ER-DataTypes : 169
USED in MAP-Errors : 110 166
USED in MAP-ER-DataTypes : 20

targetCellId.....identifier of [0] GlobalCellId
DEFINED in MAP-MS-DataTypes : 464

targetCellId.....identifier of [0] GlobalCellId
DEFINED in MAP-MS-DataTypes : 565

targetCellOutsideGCA-Param.....identifier of TargetCellOutsideGCA-Param
DEFINED in MAP-Errors : 260

TargetCellOutsideGCA-Param.....type reference SEQUENCE
DEFINED in MAP-ER-DataTypes : 383
USED in MAP-Errors : 152 260
USED in MAP-ER-DataTypes : 60

targetCellOutsideGroupCallArea.....value reference TargetCellOutsideGroupCallArea, CHOICE VALUE
DEFINED in MAP-Protocol : 362

TargetCellOutsideGroupCallArea.....type reference ERROR
DEFINED in MAP-Errors : 258
USED in MAP-Protocol : 170 362
USED in MAP-MobileServiceOpera : 106 320
USED in MAP-Errors : 38

targetMS.....identifier of [1] SubscriberIdentity
DEFINED in MAP-LCS-DataTypes : 57

targetMS.....identifier of [0] SubscriberIdentity
DEFINED in MAP-LCS-DataTypes : 62

targetMSC-Number.....identifier of [1] ISDN-AddressString
DEFINED in MAP-MS-DataTypes : 566

targetMSsubscribedService.....identifier of Named Number, 4
DEFINED in MAP-CommonDataTypes : 385

targetRNCId.....identifier of [1] RNCId
DEFINED in MAP-MS-DataTypes : 466

targetRNCId.....identifier of [2] RNCId
DEFINED in MAP-MS-DataTypes : 567

TBCD-STRING.....type reference OCTET STRING
DEFINED in MAP-CommonDataTypes : 90
USED in MAP-MS-DataTypes : 183 1670
USED in MAP-CommonDataTypes : 27 294 307 317

tBusy.....identifier of Named Number, 13
DEFINED in MAP-MS-DataTypes : 1575

TCAPMessages.....module reference
DEFINED in TCAPMessages : 1
USED in MAP-MobileServiceOpera : 79
USED in MAP-OperationAndMainte : 20
USED in MAP-CallHandlingOperat : 27
USED in MAP-SupplementaryServi : 30
USED in MAP-ShortMessageServic : 24

USED in MAP-Group-Call-Operati : 21

USED in MAP-LocationServiceOpe : 20
USED in MAP-Errors : 99

telephony.....value reference TeleserviceCode, '00010001'B
DEFINED in MAP-TS-Code : 41

teleservice.....identifier of [3] TeleserviceCode
DEFINED in MAP-CommonDataTypes : 429

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 : 64 429
USED in MAP-TS-Code : 38 40 41 42 44 45 46 48 49
50 51 55 58 67 68 69 71 72
73 74 75 76 77 78 79 80 81
82 83 84 85 86

teleserviceList.....identifier of [6] TeleserviceList
DEFINED in MAP-MS-DataTypes : 889

TeleserviceList.....type reference SEQUENCE OF
DEFINED in MAP-MS-DataTypes : 913
USED in MAP-MS-DataTypes : 889 1193

teleserviceList.....identifier of [1] TeleserviceList
DEFINED in MAP-MS-DataTypes : 1193

teleserviceNotProvisioned.....value reference TeleserviceNotProvisioned, CHOICE VALUE
DEFINED in MAP-Protocol : 352

TeleserviceNotProvisioned.....type reference ERROR
DEFINED in MAP-Errors : 245
USED in MAP-Protocol : 129 352
USED in MAP-MobileServiceOpera : 98 269 286
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 : 247

TeleservNotProvParam.....type reference SEQUENCE
DEFINED in MAP-ER-DataTypes : 231
USED in MAP-Errors : 121 247
USED in MAP-ER-DataTypes : 31

temporaryDefaultAllowed.....identifier of Named Number, 2
DEFINED in MAP-SS-DataTypes : 178

temporaryDefaultRestricted.....identifier of Named Number, 1
DEFINED in MAP-SS-DataTypes : 177

termAttemptAuthorized.....identifier of Named Number, 12
DEFINED in MAP-MS-DataTypes : 1573

terminateAllCallActivities.....identifier of Named Number, 1
DEFINED in MAP-CH-DataTypes : 439

terminateCallActivityReferred.....identifier of Named Number, 0
DEFINED in MAP-CH-DataTypes : 438

tif-csi.....identifier of Named Number, 2
DEFINED in MAP-MS-DataTypes : 1231

tif-CSI.....identifier of [3] NULL
DEFINED in MAP-MS-DataTypes : 1272

tif-CSI.....identifier of Named Number, 3
DEFINED in MAP-MS-DataTypes : 1808

tif-CSI.....identifier of [7] NULL
DEFINED in MAP-MS-DataTypes : 1845

tif-CSI-NotificationToCSE.....identifier of [8] NULL
DEFINED in MAP-MS-DataTypes : 1846

tmsi.....identifier of TMSI

DEFINED in MAP-MS-DataTypes : 281

TMSI.....type reference OCTET STRING
DEFINED in MAP-CommonDataTypes : 311
USED in MAP-MS-DataTypes : 167 281
USED in MAP-CommonDataTypes : 31 315

tmsi.....identifier of [1] TMSI
DEFINED in MAP-CommonDataTypes : 315

tNoAnswer.....identifier of Named Number, 14
DEFINED in MAP-MS-DataTypes : 1576

tooManyZoneCodes.....identifier of Named Number, 1
DEFINED in MAP-MS-DataTypes : 1204

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 : 369

TracingBufferFull.....type reference ERROR
DEFINED in MAP-Errors : 266
USED in MAP-Protocol : 132 369
USED in MAP-OperationAndMainte : 29 62
USED in MAP-Errors : 41

tracingBufferFullParam.....identifier of TracingBufferFullParam
DEFINED in MAP-Errors : 268

TracingBufferFullParam.....type reference SEQUENCE
DEFINED in MAP-ER-DataTypes : 235
USED in MAP-Errors : 122 268
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, '11000011'B
DEFINED in MAP-SS-Code : 178

translatedB-Number.....identifier of [3] ISDN-AddressString
DEFINED in MAP-CH-DataTypes : 389

translatedB-Number.....identifier of [1] ISDN-AddressString
DEFINED in MAP-SS-DataTypes : 312

tripletList.....identifier of [0] TripletList
DEFINED in MAP-MS-DataTypes : 304

TripletList.....type reference SEQUENCE OF
DEFINED in MAP-MS-DataTypes : 307
USED in MAP-MS-DataTypes : 304

ts3G-25413.....identifier of Named Number, 2
DEFINED in MAP-CommonDataTypes : 257

T-BcsmCamelTDPData.....type reference SEQUENCE
DEFINED in MAP-MS-DataTypes : 1564
USED in MAP-MS-DataTypes : 1557

t-BcsmCamelTDPDataList.....identifier of T-BcsmCamelTDPDataList
DEFINED in MAP-MS-DataTypes : 1545

T-BcsmCamelTDPDataList.....type reference SEQUENCE OF
DEFINED in MAP-MS-DataTypes : 1556

USED in MAP-MS-DataTypes : 1545

t-BcsmTriggerDetectionPoint.....identifier of T-BcsmTriggerDetectionPoint
DEFINED in MAP-MS-DataTypes : 1565

T-BcsmTriggerDetectionPoint.....type reference ENUMERATED
DEFINED in MAP-MS-DataTypes : 1572
USED in MAP-MS-DataTypes : 84 1395 1565

T-BCSM-CAMEL-TDP-Criteria.....type reference SEQUENCE
DEFINED in MAP-MS-DataTypes : 1394
USED in MAP-MS-DataTypes : 1383

t-BCSM-CAMEL-TDP-CriteriaList.....identifier of [8] T-BCSM-CAMEL-TDP-CriteriaList
DEFINED in MAP-MS-DataTypes : 1276

T-BCSM-CAMEL-TDP-CriteriaList.....type reference SEQUENCE OF
DEFINED in MAP-MS-DataTypes : 1382
USED in MAP-MS-DataTypes : 67 1276 1842 1844
USED in MAP-CH-DataTypes : 46 278

t-BCSM-CAMEL-TDP-CriteriaList.....identifier of [4] T-BCSM-CAMEL-TDP-CriteriaList
DEFINED in MAP-MS-DataTypes : 1842

t-BCSM-CAMEL-TDP-CriteriaList.....identifier of [4] T-BCSM-CAMEL-TDP-CriteriaList
DEFINED in MAP-CH-DataTypes : 278

t-BCSM-TriggerDetectionPoint.....identifier of T-BcsmTriggerDetectionPoint
DEFINED in MAP-MS-DataTypes : 1395

t-CauseValueCriteria.....identifier of [1] T-CauseValueCriteria
DEFINED in MAP-MS-DataTypes : 1397

T-CauseValueCriteria.....type reference SEQUENCE OF
DEFINED in MAP-MS-DataTypes : 1438
USED in MAP-MS-DataTypes : 1397

t-csi.....identifier of Named Number, 8
DEFINED in MAP-MS-DataTypes : 1237

T-CSI.....type reference SEQUENCE
DEFINED in MAP-MS-DataTypes : 1544
USED in MAP-MS-DataTypes : 83 1275 1841 1843
USED in MAP-CH-DataTypes : 49 273

t-CSI.....identifier of Named Number, 1
DEFINED in MAP-MS-DataTypes : 1806

t-CSI.....identifier of [3] T-CSI
DEFINED in MAP-MS-DataTypes : 1841

t-CSI.....identifier of [0] T-CSI
DEFINED in MAP-CH-DataTypes : 273

udubFromBusyMS.....identifier of Named Number, 5
DEFINED in MAP-CH-DataTypes : 406

udubFromFreeMS.....identifier of Named Number, 4
DEFINED in MAP-CH-DataTypes : 405

umts-SecurityContextData.....identifier of [1] UMTS-SecurityContextData
DEFINED in MAP-MS-DataTypes : 329

UMTS-SecurityContextData.....type reference SEQUENCE
DEFINED in MAP-MS-DataTypes : 336
USED in MAP-MS-DataTypes : 329

unauthorisedMessageOriginator.....identifier of [1] NULL
DEFINED in MAP-ER-DataTypes : 115

unauthorizedLCSClient.....value reference UnauthorizedLCSClient, CHOICE VALUE
DEFINED in MAP-Protocol : 426

UnauthorizedLCSClient.....type reference ERROR
DEFINED in MAP-Errors : 433
USED in MAP-Protocol : 163 426
USED in MAP-LocationServiceOpe : 30 81
USED in MAP-Errors : 87

unauthorizedLCSClient-Diagnostic.....identifier of [0] UnauthorizedLCSClient-Diagnostic
DEFINED in MAP-ER-DataTypes : 341

UnauthorizedLCSClient-Diagnostic.....type reference
ENUMERATED

DEFINED in MAP-ER-DataTypes : 345
USED in MAP-ER-DataTypes : 341

unauthorizedLCSClient-Param.....identifier of UnauthorizedLCSClient-Param
DEFINED in MAP-Errors : 435

UnauthorizedLCSClient-Param.....type reference SEQUENCE
DEFINED in MAP-ER-DataTypes : 340
USED in MAP-Errors : 142 435
USED in MAP-ER-DataTypes : 50

unauthorizedRequestingNetwork.....value reference UnauthorizedRequestingNetwork, CHOICE VALUE
DEFINED in MAP-Protocol : 425

UnauthorizedRequestingNetwork.....type reference ERROR
DEFINED in MAP-Errors : 428
USED in MAP-Protocol : 162 425
USED in MAP-LocationServiceOpe : 29 64 80 95
USED in MAP-Errors : 86

unauthorizedRequestingNetwork-Param.....identifier of UnauthorizedRequestingNetwork-Param
DEFINED in MAP-Errors : 430

UnauthorizedRequestingNetwork-Param.....type reference SEQUENCE
DEFINED in MAP-ER-DataTypes : 336
USED in MAP-Errors : 141 430
USED in MAP-ER-DataTypes : 49

undetermined.....identifier of Named Number, 0
DEFINED in MAP-ER-DataTypes : 135

unexpectedDataParam.....identifier of UnexpectedDataParam
DEFINED in MAP-Errors : 177

UnexpectedDataParam.....type reference SEQUENCE
DEFINED in MAP-ER-DataTypes : 185
USED in MAP-Errors : 112 177
USED in MAP-ER-DataTypes : 22

unexpectedDataValue.....value reference UnexpectedDataValue, CHOICE VALUE
DEFINED in MAP-Protocol : 330

UnexpectedDataValue.....type reference ERROR
DEFINED in MAP-Errors : 175
USED in MAP-Protocol : 118 330
USED in MAP-MobileServiceOpera : 84 184 196 206 227 240 253 266 283
304 318 343 364 375 400 411 430 444
458 472 483
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 : 214

UnidentifiedSubParam.....type reference SEQUENCE
DEFINED in MAP-ER-DataTypes : 215
USED in MAP-Errors : 116 214
USED in MAP-ER-DataTypes : 27

unidentifiedSubscriber.....value reference UnidentifiedSubscriber, CHOICE VALUE
DEFINED in MAP-Protocol : 341

UnidentifiedSubscriber.....type reference ERROR
DEFINED in MAP-Errors : 212
USED in MAP-Protocol : 123 341
USED in MAP-MobileServiceOpera : 87 216 401 412
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
USED in TCAPMessages : 52

universal.....value reference SS-Code, '10110001'B
DEFINED in MAP-SS-Code : 161

unknownAlphabet.....value reference UnknownAlphabet, CHOICE VALUE
DEFINED in MAP-Protocol : 406

UnknownAlphabet.....type reference ERROR
DEFINED in MAP-Errors : 375
USED in MAP-Protocol : 148 406
USED in MAP-SupplementaryServi : 48 184 200 214
USED in MAP-Errors : 68

unknownEquipment.....value reference UnknownEquipment, CHOICE VALUE
DEFINED in MAP-Protocol : 342

UnknownEquipment.....type reference ERROR
DEFINED in MAP-Errors : 218
USED in MAP-Protocol : 124 342
USED in MAP-MobileServiceOpera : 88 388
USED in MAP-Errors : 26

unknownMSC.....value reference UnknownMSC, CHOICE VALUE
DEFINED in MAP-Protocol : 340

UnknownMSC.....type reference ERROR
DEFINED in MAP-Errors : 210
USED in MAP-Protocol : 122 340
USED in MAP-MobileServiceOpera : 86 345
USED in MAP-Errors : 24

unknownOrUnreachableLCSClient.....value reference UnknownOrUnreachableLCSClient, CHOICE VALUE
DEFINED in MAP-Protocol : 428

UnknownOrUnreachableLCSClient.....type reference ERROR
DEFINED in MAP-Errors : 443
USED in MAP-Protocol : 165 428
USED in MAP-LocationServiceOpe : 33 96
USED in MAP-Errors : 89

unknownOrUnreachableLCSClient-Param.....identifier of UnknownOrUnreachableLCSClient-Param
DEFINED in MAP-Errors : 445

UnknownOrUnreachableLCSClient-Param.....type reference SEQUENCE
DEFINED in MAP-ER-DataTypes : 374
USED in MAP-Errors : 144 445
USED in MAP-ER-DataTypes : 52

unknownServiceCentre.....identifier of Named Number, 3
DEFINED in MAP-ER-DataTypes : 144

unknownSubscriber.....value reference UnknownSubscriber, CHOICE VALUE
DEFINED in MAP-Protocol : 338

UnknownSubscriber.....type reference ERROR
DEFINED in MAP-Errors : 199
USED in MAP-Protocol : 120 338
USED in MAP-MobileServiceOpera : 85 185 207 228 254 267 284 305 365
376 431 445 459 473 484
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 : 200

UnknownSubscriberDiagnostic.....type reference ENUMERATED
DEFINED in MAP-ER-DataTypes : 202
USED in MAP-ER-DataTypes : 200

unknownSubscriberParam.....identifier of UnknownSubscriberParam
DEFINED in MAP-Errors : 201

UnknownSubscriberParam.....type reference SEQUENCE
DEFINED in MAP-ER-DataTypes : 197
USED in MAP-Errors : 114 201
USED in MAP-ER-DataTypes : 25

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 : 253

UnstructuredSS-Notify.....type reference OPERATION
DEFINED in MAP-SupplementaryServi : 203
USED in MAP-Protocol : 76 253
USED in MAP-SupplementaryServi : 20

unstructuredSS-Request.....value reference UnstructuredSS-Request, CHOICE VALUE
DEFINED in MAP-Protocol : 252

UnstructuredSS-Request.....type reference OPERATION
DEFINED in MAP-SupplementaryServi : 187
USED in MAP-Protocol : 75 252
USED in MAP-SupplementaryServi : 19

updateGprsLocation.....value reference UpdateGprsLocation, CHOICE VALUE
DEFINED in MAP-Protocol : 302

UpdateGprsLocation.....type reference OPERATION
DEFINED in MAP-MobileServiceOpera : 220
USED in MAP-Protocol : 16 302
USED in MAP-MobileServiceOpera : 21

updateGprsLocationArg.....identifier of UpdateGprsLocationArg
DEFINED in MAP-MobileServiceOpera : 222

UpdateGprsLocationArg.....type reference SEQUENCE
DEFINED in MAP-MS-DataTypes : 380
USED in MAP-MobileServiceOpera : 121 222
USED in MAP-MS-DataTypes : 24

updateGprsLocationRes.....identifier of UpdateGprsLocationRes
DEFINED in MAP-MobileServiceOpera : 224

UpdateGprsLocationRes.....type reference SEQUENCE
DEFINED in MAP-MS-DataTypes : 400
USED in MAP-MobileServiceOpera : 122 224
USED in MAP-MS-DataTypes : 25

updateLocation.....value reference UpdateLocation, CHOICE VALUE
DEFINED in MAP-Protocol : 181

UpdateLocation.....type reference OPERATION
DEFINED in MAP-MobileServiceOpera : 176
USED in MAP-Protocol : 12 181
USED in MAP-MobileServiceOpera : 15

updateLocationArg.....identifier of UpdateLocationArg

DEFINED in MAP-MobileServiceOpera : 178

UpdateLocationArg.....type reference SEQUENCE
DEFINED in MAP-MS-DataTypes : 207
USED in MAP-MobileServiceOpera : 113 178
USED in MAP-MS-DataTypes : 16

updateLocationRes.....identifier of UpdateLocationRes
DEFINED in MAP-MobileServiceOpera : 180

UpdateLocationRes.....type reference SEQUENCE
DEFINED in MAP-MS-DataTypes : 243
USED in MAP-MobileServiceOpera : 114 180
USED in MAP-MS-DataTypes : 17

updateProcedure.....identifier of Named Number, 0
DEFINED in MAP-MS-DataTypes : 257

uplinkAttached.....identifier of [6] NULL
DEFINED in MAP-GR-DataTypes : 121

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 : 88

uplinkRequest.....identifier of [0] NULL
DEFINED in MAP-GR-DataTypes : 87

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 : 221
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 : 407

USSD-Busy.....type reference ERROR
DEFINED in MAP-Errors : 377
USED in MAP-Protocol : 149 407
USED in MAP-SupplementaryServi : 49 201 215
USED in MAP-Errors : 69

ussd-DataCodingScheme.....identifier of USSD-DataCodingScheme
DEFINED in MAP-SS-DataTypes : 222

ussd-DataCodingScheme.....identifier of USSD-DataCodingScheme
DEFINED in MAP-SS-DataTypes : 229

USSD-DataCodingScheme.....type reference OCTET STRING
DEFINED in MAP-SS-DataTypes : 233
USED in MAP-SS-DataTypes : 22 222 229
USED in MAP-LCS-DataTypes : 47 123

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 : 228
USED in MAP-SupplementaryServi : 65 179 191
USED in MAP-SS-DataTypes : 21

ussd-String.....identifier of USSD-String
DEFINED in MAP-SS-DataTypes : 223

ussd-String.....identifier of USSD-String
DEFINED in MAP-SS-DataTypes : 230

USSD-String.....type reference OCTET STRING
DEFINED in MAP-SS-DataTypes : 238
USED in MAP-SS-DataTypes : 23 223 230
USED in MAP-LCS-DataTypes : 48 132

uui.....identifier of [1] UUI
DEFINED in MAP-CH-DataTypes : 241

UUI.....type reference OCTET STRING
DEFINED in MAP-CH-DataTypes : 249
USED in MAP-CH-DataTypes : 241

uuIndicator.....identifier of [0] UUIndicator
DEFINED in MAP-CH-DataTypes : 240

UUIndicator.....type reference OCTET STRING
DEFINED in MAP-CH-DataTypes : 246
USED in MAP-CH-DataTypes : 240

uus1.....value reference SS-Code, '10000001'B
DEFINED in MAP-SS-Code : 110

uus2.....value reference SS-Code, '10000010'B
DEFINED in MAP-SS-Code : 112

uus3.....value reference SS-Code, '10000011'B
DEFINED in MAP-SS-Code : 114

uusCFInteraction.....identifier of [2] NULL
DEFINED in MAP-CH-DataTypes : 242

uu-Data.....identifier of [10] UU-Data
DEFINED in MAP-CH-DataTypes : 234

UU-Data.....type reference SEQUENCE
DEFINED in MAP-CH-DataTypes : 239
USED in MAP-CH-DataTypes : 234

valueAddedServices.....identifier of Named Number, 1
DEFINED in MAP-LCS-DataTypes : 113

VBSDataList.....type reference SEQUENCE OF
DEFINED in MAP-MS-DataTypes : 1649
USED in MAP-MS-DataTypes : 896

vbsGroupIndication.....identifier of [7] NULL
DEFINED in MAP-MS-DataTypes : 1216

vbsSubscriptionData.....identifier of [11] VBSDataList
DEFINED in MAP-MS-DataTypes : 896

verticalCoordinateRequest.....identifier of [1] NULL
DEFINED in MAP-LCS-DataTypes : 143

vertical-accuracy.....identifier of [2] Vertical-Accuracy
DEFINED in MAP-LCS-DataTypes : 144

Vertical-Accuracy.....type reference OCTET STRING
DEFINED in MAP-LCS-DataTypes : 155
USED in MAP-LCS-DataTypes : 144

VGCSDataList.....type reference SEQUENCE OF
DEFINED in MAP-MS-DataTypes : 1652
USED in MAP-MS-DataTypes : 897

vgcsGroupIndication.....identifier of [8] NULL
DEFINED in MAP-MS-DataTypes : 1217

vgcsSubscriptionData.....identifier of [12] VGCSDataList

DEFINED in MAP-MS-DataTypes : 897

vlr.....identifier of Named Number, 2
DEFINED in MAP-CommonDataTypes : 351

vlrCamelSubscriptionInfo.....identifier of [13] VlrCamelSubscriptionInfo
DEFINED in MAP-MS-DataTypes : 898

VlrCamelSubscriptionInfo.....type reference SEQUENCE
DEFINED in MAP-MS-DataTypes : 1266
USED in MAP-MS-DataTypes : 898

vlr-Capability.....identifier of [6] VLR-Capability
DEFINED in MAP-MS-DataTypes : 215

VLR-Capability.....type reference SEQUENCE
DEFINED in MAP-MS-DataTypes : 218
USED in MAP-MS-DataTypes : 215 1639

vlr-Capability.....identifier of [6] VLR-Capability
DEFINED in MAP-MS-DataTypes : 1639

vlr-Number.....identifier of ISDN-AddressString
DEFINED in MAP-MS-DataTypes : 211

vlr-Number.....identifier of [0] ISDN-AddressString
DEFINED in MAP-MS-DataTypes : 269

vlr-number.....identifier of [1] ISDN-AddressString
DEFINED in MAP-MS-DataTypes : 1708

vmsc.....identifier of Named Number, 5
DEFINED in MAP-CommonDataTypes : 354

vmsc-Address.....identifier of [2] ISDN-AddressString
DEFINED in MAP-CH-DataTypes : 157

voiceBroadcastCall.....value reference TeleserviceCode, '10010010'B
DEFINED in MAP-TS-Code : 69

VoiceBroadcastData.....type reference SEQUENCE
DEFINED in MAP-MS-DataTypes : 1664
USED in MAP-MS-DataTypes : 1650

voiceGroupCall.....value reference TeleserviceCode, '10010001'B
DEFINED in MAP-TS-Code : 68

VoiceGroupCallData.....type reference SEQUENCE
DEFINED in MAP-MS-DataTypes : 1659
USED in MAP-MS-DataTypes : 1653

vplmnAddressAllowed.....identifier of [19] NULL
DEFINED in MAP-MS-DataTypes : 749

vt-BCSM-CAMEL-TDP-CriteriaList.....identifier of [6] T-BCSM-CAMEL-TDP-CriteriaList
DEFINED in MAP-MS-DataTypes : 1844

vt-csi.....identifier of Named Number, 4
DEFINED in MAP-MS-DataTypes : 1233

vt-CSI.....identifier of [7] T-CSI
DEFINED in MAP-MS-DataTypes : 1275

vt-CSI.....identifier of Named Number, 2
DEFINED in MAP-MS-DataTypes : 1807

vt-CSI.....identifier of [5] T-CSI
DEFINED in MAP-MS-DataTypes : 1843

whiteListed.....identifier of Named Number, 0
DEFINED in MAP-MS-DataTypes : 681

wrongNetworkSignature.....identifier of Named Number, 1
DEFINED in MAP-MS-DataTypes : 376

wrongPasswordAttemptsCounter.....identifier of WrongPasswordAttemptsCounter
DEFINED in MAP-MS-DataTypes : 1824

WrongPasswordAttemptsCounter.....type reference INTEGER
DEFINED in MAP-MS-DataTypes : 1829
USED in MAP-MS-DataTypes : 1824 1887 1953

wrongPasswordAttemptsCounter.....identifier of [4] WrongPasswordAttemptsCounter
DEFINED in MAP-MS-DataTypes : 1887

wrongPasswordAttemptsCounter.....identifier of [3] WrongPasswordAttemptsCounter
DEFINED in MAP-MS-DataTypes : 1953

wrongUserResponse.....identifier of Named Number, 0
DEFINED in MAP-MS-DataTypes : 375

xres.....identifier of XRES
DEFINED in MAP-MS-DataTypes : 321

XRES.....type reference OCTET STRING
DEFINED in MAP-MS-DataTypes : 348
USED in MAP-MS-DataTypes : 321

ZoneCode.....type reference OCTET STRING
DEFINED in MAP-MS-DataTypes : 1187
USED in MAP-MS-DataTypes : 1185 1215

ZoneCodeList.....type reference SEQUENCE OF
DEFINED in MAP-MS-DataTypes : 1184
USED in MAP-MS-DataTypes : 62 895

zoneCodesConflict.....identifier of Named Number, 2
DEFINED in MAP-MS-DataTypes : 1205

Annex B (informative): Fully expanded ASN.1 sources for abstract syntaxes of MAP

Annex B is not part of the standard, it is included for information purposes only.

For every (Value)Assignment in the root ASN.1 module all the used defined types and defined values, which are defined within the ASN.1 module or imported from ASN.1 modules, are replaced by the constructs this type or value is composed of.

The fully expanded ASN.1 root module is itself a correct and equivalent representation of the MAP-Protocol.

It allows to see at all the parameters, including all nested ones for a specific operationcode or errorcode at once.

Note that for those operations which use a result without parameters the keyword RESULT is not shown. Empty results are only defined in the ASN.1 description in clause 17.

B.1 Fully Expanded ASN.1 Source of MAP-Protocol/TCAPMessages

```
--          Expanded ASN1 Module 'MAP-Protocol'
--SIEMENS ASN.1 Compiler       R5.70  (Production_5.70)
--           Date: 2004-06-09 Time: 15:32:55

MAP-Protocol{ 0 identified-organization (4) etsi (0) mobileDomain (0) gsm-Network (1) modules (3) map-Protocol (4) version6 (6) }

DEFINITIONS

::=

BEGIN

updateLocation OPERATION
  ARGUMENT
    updateLocationArg SEQUENCE {
      imsi                    OCTET STRING ( SIZE( 3 .. 8 ) ),
      msc-Number                  [1] IMPLICIT OCTET STRING ( SIZE( 1 .. 20 ) ) ( SIZE( 1 .. 9 ) ),
      vlr-Number                OCTET STRING ( SIZE( 1 .. 20 ) ) ( SIZE( 1 .. 9 ) ),
      lmsi                    [10] IMPLICIT 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,
        longFTN-Supported                       [4] IMPLICIT NULL OPTIONAL} OPTIONAL,
      informPreviousNetworkEntity   [11] IMPLICIT NULL 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 ) OPTIONAL,
      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( 16 ) ),
              ... }} OPTIONAL,
        currentSecurityContext   [2] CHOICE {
          gsm-SecurityContextData     [0] IMPLICIT SEQUENCE {
            kc        OCTET STRING ( SIZE( 8 ) ),
            cksn      OCTET STRING ( SIZE( 1 ) ),
            ... },
          umts-SecurityContextData    [1] IMPLICIT SEQUENCE {
            ck        OCTET STRING ( SIZE( 16 ) ),
            ik        OCTET STRING ( SIZE( 16 ) ),
            ksi       OCTET STRING ( SIZE( 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 {
      -- dataMissing -- localValue : 35,
      -- unidentifiedSubscriber -- localValue : 5}
   ::= localValue : 55

prepareHandover OPERATION
    ARGUMENT
      prepareHO-Arg [3] IMPLICIT SEQUENCE {
        targetCellId              [0] IMPLICIT OCTET STRING ( SIZE( 5 .. 7 ) ) OPTIONAL,
        ho-NumberNotRequired        NULL OPTIONAL,
        targetRNCId               [1] IMPLICIT OCTET STRING ( SIZE( 7 ) ) OPTIONAL,
        an-APDU                   [2] IMPLICIT SEQUENCE {
          accessNetworkProtocolId   ENUMERATED {
            gsm-0806      ( 1 ),
            ts3G-25413    ( 2 ),
            ...  },
          signalInfo              OCTET STRING ( SIZE( 1 .. 2560 ) ),
          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,
        multipleBearerRequested     [3] IMPLICIT NULL OPTIONAL,
        imsi                      [4] IMPLICIT OCTET STRING ( SIZE( 3 .. 8 ) ) OPTIONAL,
        integrityProtectionInfo     [5] IMPLICIT OCTET STRING ( SIZE( 18 .. 100 ) ) OPTIONAL,
        encryptionInfo            [6] IMPLICIT OCTET STRING ( SIZE( 18 .. 100 ) ) OPTIONAL,
        radioResourceInformation    [7] IMPLICIT OCTET STRING ( SIZE( 3 .. 13 ) ) OPTIONAL,
        allowedGSM-Algorithms       [9] IMPLICIT OCTET STRING ( SIZE( 1 ) ) OPTIONAL,
        allowedUMTS-Algorithms      [10] IMPLICIT SEQUENCE {
          integrityProtectionAlgorithms   [0] IMPLICIT OCTET STRING ( SIZE( 1 .. 9 ) ) OPTIONAL,
          encryptionAlgorithms            [1] IMPLICIT OCTET STRING ( SIZE( 1 .. 9 ) ) 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,
      radioResourceList           [11] IMPLICIT SEQUENCE  ( SIZE( 1 .. 7 ) ) OF
        SEQUENCE {
          radioResourceInformation   OCTET STRING ( SIZE( 3 .. 13 ) ),
          rab-Id                     INTEGER ( 1 .. 255 ),
          ... } 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,
      ... ,
      rab-Id                     [12] IMPLICIT INTEGER ( 1 .. 255 ) OPTIONAL,
      bssmap-ServiceHandover       [13] IMPLICIT OCTET STRING ( SIZE( 1 ) ) OPTIONAL,
      ranap-ServiceHandover        [14] IMPLICIT OCTET STRING ( SIZE( 1 ) ) OPTIONAL,
      bssmap-ServiceHandoverList   [15] IMPLICIT SEQUENCE  ( SIZE( 1 .. 7 ) ) OF
        SEQUENCE {
          bssmap-ServiceHandover   OCTET STRING ( SIZE( 1 ) ),
          rab-Id                   INTEGER ( 1 .. 255 ),
          ... } OPTIONAL,
      asciCallReference           [20] IMPLICIT OCTET STRING ( SIZE( 1 .. 8 ) ) OPTIONAL}
  RESULT
    prepareHO-Res [3] IMPLICIT SEQUENCE {
      handoverNumber                 [0] IMPLICIT OCTET STRING ( SIZE( 1 .. 20 ) ) ( SIZE( 1 .. 9 ) ) OPTIONAL,
      relocationNumberList           [1] IMPLICIT SEQUENCE  ( SIZE( 1 .. 7 ) ) OF
        SEQUENCE {
          handoverNumber   OCTET STRING ( SIZE( 1 .. 20 ) ) ( SIZE( 1 .. 9 ) ),
          rab-Id           INTEGER ( 1 .. 255 ),
          ... } OPTIONAL,
      an-APDU                        [2] IMPLICIT SEQUENCE {
        accessNetworkProtocolId   ENUMERATED {
          gsm-0806      ( 1 ),
          ts3G-25413    ( 2 ),
          ...  },
        signalInfo               OCTET STRING ( SIZE( 1 .. 2560 ) ),
        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,
      multicallBearerInfo              [3] IMPLICIT INTEGER ( 1 .. 7 ) OPTIONAL,
      multipleBearerNotSupported       NULL OPTIONAL,
      selectedUMTS-Algorithms          [5] IMPLICIT SEQUENCE {
        integrityProtectionAlgorithm   [0] IMPLICIT OCTET STRING ( SIZE( 1 ) ) OPTIONAL,
        encryptionAlgorithm            [1] IMPLICIT OCTET STRING ( SIZE( 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,
      chosenRadioResourceInformation   [6] IMPLICIT SEQUENCE {
        chosenChannelInfo     [0] IMPLICIT OCTET STRING ( SIZE( 1 ) ) OPTIONAL,
        chosenSpeechVersion   [1] IMPLICIT OCTET STRING ( SIZE( 1 ) ) 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,
      ... }
  ERRORS {
    -- systemFailure -- localValue : 34,
    -- dataMissing -- localValue : 35,
    -- unexpectedDataValue -- localValue : 36,
    -- noHandoverNumberAvailable -- localValue : 25,
    -- targetCellOutsideGroupCallArea -- localValue : 42}
 ::= localValue : 68

sendEndSignal OPERATION
  ARGUMENT
    sendEndSignal-Arg [3] IMPLICIT SEQUENCE {
      an-APDU             SEQUENCE {
        accessNetworkProtocolId   ENUMERATED {
          gsm-0806      ( 1 ),
          ts3G-25413    ( 2 ),
          ...  },
        signalInfo              OCTET STRING ( SIZE( 1 .. 2560 ) ),
        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,
      ... }
  RESULT
    sendEndSignal-Res 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,
      ... }
 ::= localValue : 29

processAccessSignalling OPERATION
  ARGUMENT
    processAccessSignalling-Arg [3] IMPLICIT SEQUENCE {
      an-APDU                    SEQUENCE {
        accessNetworkProtocolId   ENUMERATED {
          gsm-0806      ( 1 ),
          ts3G-25413    ( 2 ),
          ... },
        signalInfo              OCTET STRING ( SIZE( 1 .. 2560 ) ),
        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,
        ... },
      selectedUMTS-Algorithms         [1] IMPLICIT SEQUENCE {
        integrityProtectionAlgorithm   [0] IMPLICIT OCTET STRING ( SIZE( 1 ) ) OPTIONAL,
        encryptionAlgorithm           [1] IMPLICIT OCTET STRING ( SIZE( 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,
      selectedGSM-Algorithm           [2] IMPLICIT OCTET STRING ( SIZE( 1 ) ) OPTIONAL,
      chosenRadioResourceInformation   [3] IMPLICIT SEQUENCE {
        chosenChannelInfo     [0] IMPLICIT OCTET STRING ( SIZE( 1 ) ) OPTIONAL,
        chosenSpeechVersion   [1] IMPLICIT OCTET STRING ( SIZE( 1 ) ) OPTIONAL,
        ... } OPTIONAL,
      selectedRab-Id                  [4] IMPLICIT INTEGER ( 1 .. 255 ) 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,
      ... }
 ::= localValue : 33

forwardAccessSignalling OPERATION
  ARGUMENT
    forwardAccessSignalling-Arg [3] IMPLICIT SEQUENCE {
      an-APDU                    SEQUENCE {
        accessNetworkProtocolId   ENUMERATED {
          gsm-0806      ( 1 ),
          ts3G-25413    ( 2 ),
          ... },
        signalInfo              OCTET STRING ( SIZE( 1 .. 2560 ) ),
        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,
          ... },
        integrityProtectionInfo      [0] IMPLICIT OCTET STRING ( SIZE( 18 .. 100 ) ) OPTIONAL,
        encryptionInfo               [1] IMPLICIT OCTET STRING ( SIZE( 18 .. 100 ) ) OPTIONAL,
        keyStatus                    [2] IMPLICIT ENUMERATED {
          old        ( 0 ),
          new        ( 1 ),
          ...  } OPTIONAL,
        allowedGSM-Algorithms        [4] IMPLICIT OCTET STRING ( SIZE( 1 ) ) OPTIONAL,
        allowedUMTS-Algorithms       [5] IMPLICIT SEQUENCE {
          integrityProtectionAlgorithms   [0] IMPLICIT OCTET STRING ( SIZE( 1 .. 9 ) ) OPTIONAL,
          encryptionAlgorithms           [1] IMPLICIT OCTET STRING ( SIZE( 1 .. 9 ) ) 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,
        radioResourceInformation     [6] IMPLICIT OCTET STRING ( SIZE( 3 .. 13 ) ) 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,
        ... ,
        radioResourceList            [7] IMPLICIT SEQUENCE  ( SIZE( 1 .. 7 ) ) OF
          SEQUENCE {
            radioResourceInformation   OCTET STRING ( SIZE( 3 .. 13 ) ),
            rab-Id                     INTEGER ( 1 .. 255 ),
            ... } OPTIONAL,
        bssmap-ServiceHandover       [9] IMPLICIT OCTET STRING ( SIZE( 1 ) ) OPTIONAL,
        ranap-ServiceHandover        [8] IMPLICIT OCTET STRING ( SIZE( 1 ) ) OPTIONAL,
        bssmap-ServiceHandoverList   [10] IMPLICIT SEQUENCE  ( SIZE( 1 .. 7 ) ) OF
          SEQUENCE {
            bssmap-ServiceHandover   OCTET STRING ( SIZE( 1 ) ),
            rab-Id                   INTEGER ( 1 .. 255 ),
            ... } OPTIONAL}
 ::= localValue : 34

prepareSubsequentHandover OPERATION
  ARGUMENT
    prepareSubsequentHO-Arg [3] IMPLICIT SEQUENCE {
      targetCellId         [0] IMPLICIT OCTET STRING ( SIZE( 5 .. 7 ) ) OPTIONAL,
      targetMSC-Number     [1] IMPLICIT OCTET STRING ( SIZE( 1 .. 20 ) ) ( SIZE( 1 .. 9 ) ),
      targetRNCId          [2] IMPLICIT OCTET STRING ( SIZE( 7 ) ) OPTIONAL,
      an-APDU              [3] IMPLICIT SEQUENCE {
        accessNetworkProtocolId   ENUMERATED {
          gsm-0806      ( 1 ),
          ts3G-25413    ( 2 ),
          ...  },
        signalInfo               OCTET STRING ( SIZE( 1 .. 2560 ) ),
        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,
        selectedRab-Id       [4] IMPLICIT INTEGER ( 1 .. 255 ) 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,
        ... }
    RESULT
      prepareSubsequentHO-Res [3] IMPLICIT SEQUENCE {
        an-APDU              SEQUENCE {
          accessNetworkProtocolId   ENUMERATED {
            gsm-0806      ( 1 ),
            ts3G-25413    ( 2 ),
            ...  },
          signalInfo               OCTET STRING ( SIZE( 1 .. 2560 ) ),
          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,
        ... }
    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 ) ),
          auts       OCTET STRING ( SIZE( 14 ) ),
          ... } 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( 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,
      ... }
  ERRORS {
    -- systemFailure -- localValue : 34,
    -- dataMissing -- localValue : 35,
    -- unexpectedDataValue -- localValue : 36,
    -- unknownSubscriber -- localValue : 1}
 ::= localValue : 56

authenticationFailureReport OPERATION
  ARGUMENT
    authenticationFailureReportArg SEQUENCE {
      imsi               OCTET STRING ( SIZE( 3 .. 8 ) ),
      failureCause       ENUMERATED {
        wrongUserResponse       ( 0 ),
        wrongNetworkSignature   ( 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,
      ... }
  RESULT
    authenticationFailureReportRes 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,
    -- unexpectedDataValue -- localValue : 36,
    -- unknownSubscriber -- localValue : 1}
 ::= localValue : 15

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,
                ... ,
                longForwardedToNumber   [10] IMPLICIT OCTET STRING ( SIZE( 1 .. 20 ) ) ( SIZE( 1 .. 15 ) ) 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,
              ... } 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 {
            maximumentitledPriority   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 ),
            premiumRateEntertainementOGCallsBarred  (4 ),
            ss-AccessBarred  (5 ),
            allECT-Barred  (9 ),
            chargeableECT-Barred  (10 ),
            internationalECT-Barred  (11 ),
            interzonalECT-Barred  (12 ),
            doublyChargeableECT-Barred  (13 ),
```

```
      multipleECT-Barred  (14 ),
      roamingOutsidePLMNOG-CallsBarred  (18 ),
      allIC-CallsBarred  (19 ),
      roamingOutsidePLMNIC-CallsBarred  (20 ),
      roamingOutsidePLMNICountryIC-CallsBarred  (21 ),
      roamingOutsidePLMN-Barred  (22 ),
      roamingOutsidePLMN-CountryBarred  (23 ),
      registrationAllCF-Barred  (24 ),
      registrationCFNotToHPLMN-Barred  (25 ),
      registrationInterzonalCF-Barred  (26 ),
      registrationInterzonalCFNotToHPLMN-Barred  (27 ),
      registrationInternationalCF-Barred  (28 )} ( 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,
    ... } 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,
  vgcsSubscriptionData                    [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,
  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,
      ... },
    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    [0] IMPLICIT NULL OPTIONAL,
```

```
    csi-Active          [1] IMPLICIT NULL 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,
    csi-Active          [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 ) ),
        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,
          ... } OPTIONAL,
    camelCapabilityHandling   [1] IMPLICIT INTEGER ( 1 .. 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,
    notificationToCSE         [3] IMPLICIT NULL OPTIONAL,
    csi-Active                [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   [0] IMPLICIT 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,
      camelCapabilityHandling       [1] IMPLICIT INTEGER ( 1 .. 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,
      notificationToCSE             [3] IMPLICIT NULL OPTIONAL,
      csi-Active                    [4] IMPLICIT NULL 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,
  gprsSubscriptionData                      [16] IMPLICIT SEQUENCE {
    completeDataListIncluded    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,
        ... ,
        ext-QoS-Subscribed          [0] IMPLICIT OCTET STRING ( SIZE( 1 .. 9 ) ) OPTIONAL,
        pdp-ChargingCharacteristics  [1] IMPLICIT OCTET STRING ( SIZE( 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,
  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 ) ),
        lsaAttributes           [1] IMPLICIT OCTET STRING ( SIZE( 1 ) ),
        lsaActiveModeIndicator  [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,
    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,
  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 ) ),
    notificationToMSUser   [0] IMPLICIT ENUMERATED {
      notifyLocationAllowed                          ( 0 ),
      notifyAndVerify-LocationAllowedIfNoResponse      ( 1 ),
      notifyAndVerify-LocationNotAllowedIfNoResponse   ( 2 ),
      ...  } OPTIONAL,
    externalClientList     [1] IMPLICIT SEQUENCE  ( SIZE( 0 .. 5 ) ) OF
      SEQUENCE {
        clientIdentity        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,
          ... },
        gmlc-Restriction       [0] IMPLICIT ENUMERATED {
          gmlc-List       ( 0 ),
          home-Country    ( 1 ),
          ...  } OPTIONAL,
        notificationToMSUser   [1] IMPLICIT ENUMERATED {
          notifyLocationAllowed                          ( 0 ),
          notifyAndVerify-LocationAllowedIfNoResponse      ( 1 ),
          notifyAndVerify-LocationNotAllowedIfNoResponse   ( 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,
    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,
  mc-SS-Info                            [28] IMPLICIT SEQUENCE {
    ss-Code             [0] IMPLICIT OCTET STRING ( SIZE( 1 ) ),
    ss-Status           [1] IMPLICIT OCTET STRING ( SIZE( 1 .. 5 ) ),
    nbrSB               [2] IMPLICIT INTEGER ( 2 .. 7 ),
    nbrUser             [3] IMPLICIT INTEGER ( 1 .. 7 ),
    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,
  cs-AllocationRetentionPriority                [29] IMPLICIT OCTET STRING ( SIZE( 1 ) ) OPTIONAL,
  sgsn-CAMEL-SubscriptionInfo                   [17] 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,
          ... } OPTIONAL,
      camelCapabilityHandling   [1] IMPLICIT INTEGER ( 1 .. 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,
      notificationToCSE         [3] IMPLICIT NULL OPTIONAL,
      csi-Active                [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,
              ... } OPTIONAL,
          camelCapabilityHandling   [1] IMPLICIT INTEGER ( 1 .. 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,
          notificationToCSE         [3] IMPLICIT NULL OPTIONAL,
          csi-Active                [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,
      chargingCharacteristics                 [18] IMPLICIT OCTET STRING ( SIZE( 2 ) ) 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 ),
        premiumRateEntertainementOGCallsBarred  (4 ),
        ss-AccessBarred  (5 ),
        allECT-Barred  (9 ),
        chargeableECT-Barred  (10 ),
        internationalECT-Barred  (11 ),
        interzonalECT-Barred  (12 ),
```

```
        doublyChargeableECT-Barred  (13 ),
        multipleECT-Barred  (14 ),
        roamingOutsidePLMNOG-CallsBarred  (18 ),
        allIC-CallsBarred  (19 ),
        roamingOutsidePLMNIC-CallsBarred  (20 ),
        roamingOutsidePLMNICountryIC-CallsBarred  (21 ),
        roamingOutsidePLMN-Barred  (22 ),
        roamingOutsidePLMN-CountryBarred  (23 ),
        registrationAllCF-Barred  (24 ),
        registrationCFNotToHPLMN-Barred  (25 ),
        registrationInterzonalCF-Barred  (26 ),
        registrationInterzonalCFNotToHPLMN-Barred  (27 ),
        registrationInternationalCF-Barred  (28 )} ( 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,
      roamingRestrictedInSgsnDueToUnsuppportedFeature   [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,
        specificCSI-Withdraw                [15] IMPLICIT BIT STRING {
          o-csi  (0 ),
          ss-csi  (1 ),
          tif-csi  (2 ),
          d-csi  (3 ),
          vt-csi  (4 ),
          sms-csi  (5 ),
          m-csi  (6 ),
          gprs-csi  (7 ),
          t-csi  (8 )} ( SIZE( 8 .. 32 ) ) 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,
      longFTN-Supported                        [4] IMPLICIT NULL 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,
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,
gmsc-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,
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,
      longFTN-Supported                [21] IMPLICIT NULL 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,
            ... ,
            longForwardedToNumber   [8] IMPLICIT OCTET STRING ( SIZE( 1 .. 20 ) ) ( SIZE( 1 .. 15 ) ) 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,
            ... ,
            longForwardedToNumber   [8] IMPLICIT OCTET STRING ( SIZE( 1 .. 20 ) ) ( SIZE( 1 .. 15 ) ) 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} 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   [0] IMPLICIT 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,
          camelCapabilityHandling       [1] IMPLICIT INTEGER ( 1 .. 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,
          notificationToCSE             [3] IMPLICIT NULL OPTIONAL,
          csi-Active                    [4] 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,
        ... }} 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,
      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,
          cellGlobalIdOrServiceAreaIdOrLAI   [3] CHOICE {
            cellGlobalIdOrServiceAreaIdFixedLength     [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,
      currentLocationRetrieved        [8] IMPLICIT NULL OPTIONAL,
      sai-Present                     [9] IMPLICIT NULL 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,
  ... ,
  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,
      longFTN-Supported           [18] 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,
      ... }
  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,
    ... ,
    longForwardedToNumber   [8] IMPLICIT OCTET STRING ( SIZE( 1 .. 20 ) ) ( SIZE( 1 .. 15 ) ) 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   [0] IMPLICIT 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,
      camelCapabilityHandling       [1] IMPLICIT INTEGER ( 1 .. 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,
      notificationToCSE             [3] IMPLICIT NULL OPTIONAL,
      csi-Active                    [4] IMPLICIT NULL 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,
        ... }
  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,
        ... } 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,
      ... }
  RESULT
    provideSIWFSNumberRes SEQUENCE {
      sIWFSNumber          [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,
      ... }
  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,
      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,
      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 ),
      ududFromFreeMS          ( 4 ),
      ududFromBusyMS          ( 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,
      nbrUser                 [8] IMPLICIT INTEGER ( 1 .. 7 ) OPTIONAL,
      longFTN-Supported       [9] IMPLICIT NULL 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,
            ... ,
            longForwardedToNumber   [9] IMPLICIT OCTET STRING ( SIZE( 1 .. 20 ) ) ( SIZE( 1 .. 15 ) ) 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,
        nbrUser                 [5] IMPLICIT INTEGER ( 1 .. 7 ) 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,
      ... ,
      longFTN-Supported   [4] IMPLICIT NULL 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,
            ... ,
            longForwardedToNumber   [9] IMPLICIT OCTET STRING ( SIZE( 1 .. 20 ) ) ( SIZE( 1 .. 15 ) ) 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,
        nbrUser                [5] IMPLICIT INTEGER ( 1 .. 7 ) 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,
```

```
      ... ,
      longFTN-Supported   [4] IMPLICIT NULL 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,
            ... ,
            longForwardedToNumber   [9] IMPLICIT OCTET STRING ( SIZE( 1 .. 20 ) ) ( SIZE( 1 .. 15 ) ) 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,
        nbrUser                 [5] IMPLICIT INTEGER ( 1 .. 7 ) 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,
      ... ,
      longFTN-Supported   [4] IMPLICIT NULL 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,
            ... ,
            longForwardedToNumber [9] IMPLICIT OCTET STRING ( SIZE( 1 .. 20 ) ) ( SIZE( 1 .. 15 ) ) 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,
        nbrUser               [5] IMPLICIT INTEGER ( 1 .. 7 ) 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,
      ... ,
      longFTN-Supported  [4] IMPLICIT NULL 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,
            ... ,
            longForwardedToNumber   [9] IMPLICIT OCTET STRING ( SIZE( 1 .. 20 ) ) ( SIZE( 1 .. 15 ) ) 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,
            nbrSB                   [3] IMPLICIT INTEGER ( 2 .. 7 ) OPTIONAL,
            nbrUser                 [4] IMPLICIT INTEGER ( 1 .. 7 ) OPTIONAL,
            nbrSN                   [5] IMPLICIT INTEGER ( 1 .. 7 ) 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,
      ... }
  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,
      ... ,
      currentLocation       [3] 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,
    ... }
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,
        cellGlobalIdOrServiceAreaIdOrLAI   [3] CHOICE {
          cellGlobalIdOrServiceAreaIdFixedLength     [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,
        currentLocationRetrieved          [8] IMPLICIT NULL OPTIONAL,
        sai-Present                       [9] IMPLICIT NULL 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,
        ... ,
        currentLocation       [3] IMPLICIT NULL 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,
          cellGlobalIdOrServiceAreaIdOrLAI   [3] CHOICE {
            cellGlobalIdOrServiceAreaIdFixedLength     [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,
            currentLocationRetrieved        [8] IMPLICIT NULL OPTIONAL,
            sai-Present                     [9] IMPLICIT NULL 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,
            ... ,
            longFTN-Supported   [4] IMPLICIT NULL 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,
    longFTN-Supported           [4] IMPLICIT NULL 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,
          ... ,
          longForwardedToNumber   [10] IMPLICIT OCTET STRING ( SIZE( 1 .. 20 ) ) ( SIZE( 1 .. 15 ) ) 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,
```

```
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 ) ) OPTIONAL,
  wrongPasswordAttemptsCounter   INTEGER ( 0 .. 4 ) 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,
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 ),
      premiumRateEntertainementOGCallsBarred  (4 ),
      ss-AccessBarred  (5 ),
      allECT-Barred  (9 ),
      chargeableECT-Barred  (10 ),
      internationalECT-Barred  (11 ),
      interzonalECT-Barred  (12 ),
      doublyChargeableECT-Barred  (13 ),
      multipleECT-Barred  (14 ),
      roamingOutsidePLMNOG-CallsBarred  (18 ),
      allIC-CallsBarred  (19 ),
      roamingOutsidePLMNIC-CallsBarred  (20 ),
      roamingOutsidePLMNICountryIC-CallsBarred  (21 ),
      roamingOutsidePLMN-Barred  (22 ),
      roamingOutsidePLMN-CountryBarred  (23 ),
      registrationAllCF-Barred  (24 ),
      registrationCFNotToHPLMN-Barred  (25 ),
      registrationInterzonalCF-Barred  (26 ),
      registrationInterzonalCFNotToHPLMN-Barred  (27 ),
      registrationInternationalCF-Barred  (28 )} ( 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,
    d-CSI                       [2] IMPLICIT SEQUENCE {
      dp-AnalysedInfoCriteriaList   [0] IMPLICIT 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,
      camelCapabilityHandling       [1] IMPLICIT INTEGER ( 1 .. 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,
      notificationToCSE             [3] IMPLICIT NULL OPTIONAL,
      csi-Active                    [4] IMPLICIT NULL OPTIONAL,
      ... } OPTIONAL,
    t-CSI                       [3] 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    [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,
  vt-CSI                        [5] 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   [6] 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                       [7] IMPLICIT NULL OPTIONAL,
    tif-CSI-NotificationToCSE     [8] IMPLICIT NULL OPTIONAL,
    gprs-CSI                      [9] 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,
          ... } OPTIONAL,
      camelCapabilityHandling   [1] IMPLICIT INTEGER ( 1 .. 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,
      notificationToCSE         [3] IMPLICIT NULL OPTIONAL,
      csi-Active                [4] IMPLICIT NULL OPTIONAL,
      ... } OPTIONAL,
    sms-CSI                       [10] 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,
        ... } OPTIONAL,
    camelCapabilityHandling   [1] IMPLICIT INTEGER ( 1 .. 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,
    notificationToCSE         [3] IMPLICIT NULL OPTIONAL,
    csi-Active                [4] IMPLICIT NULL OPTIONAL,
    ... } OPTIONAL,
  ss-CSI                        [11] 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,
      ... },
    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    [0] IMPLICIT NULL OPTIONAL,
    csi-Active           [1] IMPLICIT NULL OPTIONAL} OPTIONAL,
  m-CSI                         [12] 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,
          csi-Active           [3] IMPLICIT NULL OPTIONAL,
          ... } 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,
        ... ,
        specificCSIDeletedList          [14] IMPLICIT BIT STRING {
          o-csi  (0 ),
          ss-csi  (1 ),
          tif-csi  (2 ),
          d-csi  (3 ),
          vt-csi  (4 ),
          sms-csi  (5 ),
          m-csi  (6 ),
          gprs-csi  (7 ),
          t-csi  (8 )} ( SIZE( 8 .. 32 ) ) 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-CF-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-CB-Info  [3] 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,
    password                    [3] IMPLICIT NumericString ( FROM ("0"|"1"|"2"|"3"|"4"|"5"|"6"|"7"|"8"|"9" )) (SIZE( 4 ) ) OPTIONAL,
    wrongPasswordAttemptsCounter  [4] IMPLICIT INTEGER ( 0 .. 4 ) OPTIONAL,
    modifyNotificationToCSE       [5] IMPLICIT ENUMERATED {
      deactivate    ( 0 ),
      activate      ( 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,
    ... } OPTIONAL,
  modificationRequestFor-CSI      [4] IMPLICIT SEQUENCE {
    requestedCamel-SubscriptionInfo  [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 ) },
    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,
  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,
      longFTN-Supported              [6] IMPLICIT NULL 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,
              ... ,
              longForwardedToNumber   [10] IMPLICIT OCTET STRING ( SIZE( 1 .. 20 ) ) ( SIZE( 1 .. 15 ) ) OPTIONAL},
          notificationToCSE       [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,
          ... },
        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 ) ) OPTIONAL,
```

```
        wrongPasswordAttemptsCounter   [3] IMPLICIT INTEGER ( 0 .. 4 ) OPTIONAL,
        notificationToCSE              [4] IMPLICIT NULL 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,
        ... }} OPTIONAL,
      camel-SubscriptionInfo   [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,
    d-CSI                       [2] IMPLICIT SEQUENCE {
      dp-AnalysedInfoCriteriaList   [0] IMPLICIT 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,
      camelCapabilityHandling       [1] IMPLICIT INTEGER ( 1 .. 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,
      notificationToCSE             [3] IMPLICIT NULL OPTIONAL,
      csi-Active                    [4] IMPLICIT NULL OPTIONAL,
      ... } OPTIONAL,
    t-CSI                       [3] 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    [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,
  vt-CSI                        [5] 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   [6] 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                        [7] IMPLICIT NULL OPTIONAL,
  tif-CSI-NotificationToCSE      [8] IMPLICIT NULL OPTIONAL,
  gprs-CSI                       [9] 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,
        ... } OPTIONAL,
    camelCapabilityHandling   [1] IMPLICIT INTEGER ( 1 .. 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,
    notificationToCSE         [3] IMPLICIT NULL OPTIONAL,
    csi-Active                [4] IMPLICIT NULL OPTIONAL,
    ... } OPTIONAL,
  sms-CSI                        [10] 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,
          ... } OPTIONAL,
    camelCapabilityHandling   [1] IMPLICIT INTEGER ( 1 .. 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,
    notificationToCSE         [3] IMPLICIT NULL OPTIONAL,
    csi-Active                [4] IMPLICIT NULL OPTIONAL,
    ... } OPTIONAL,
  ss-CSI                        [11] 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,
      ... },
    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    [0] IMPLICIT NULL OPTIONAL,
    csi-Active           [1] IMPLICIT NULL OPTIONAL} OPTIONAL,
  m-CSI                         [12] 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,
          csi-Active           [3] IMPLICIT NULL OPTIONAL,
          ... } 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,
        ... ,
        specificCSIDeletedList          [14] IMPLICIT BIT STRING {
          o-csi  (0 ),
          ss-csi  (1 ),
          tif-csi  (2 ),
          d-csi  (3 ),
          vt-csi  (4 ),
          sms-csi  (5 ),
          m-csi  (6 ),
          gprs-csi  (7 ),
          t-csi  (8 )} ( SIZE( 8 .. 32 ) ) 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 {
    -- 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 ) ),
      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,
          ... ,
          longForwardedToNumber   [10] IMPLICIT OCTET STRING ( SIZE( 1 .. 20 ) ) ( SIZE( 1 .. 15 ) ) OPTIONAL},
      notificationToCSE       [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,
    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 ) ) OPTIONAL,
      wrongPasswordAttemptsCounter   [3] IMPLICIT INTEGER ( 0 .. 4 ) OPTIONAL,
      notificationToCSE             [4] IMPLICIT NULL 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,
      ... } OPTIONAL,
    odb-Info                [2] 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 ),
          premiumRateEntertainementOGCallsBarred  (4 ),
          ss-AccessBarred  (5 ),
          allECT-Barred  (9 ),
          chargeableECT-Barred  (10 ),
          internationalECT-Barred  (11 ),
          interzonalECT-Barred  (12 ),
          doublyChargeableECT-Barred  (13 ),
```

```
        multipleECT-Barred  (14 ),
        roamingOutsidePLMNOG-CallsBarred  (18 ),
        allIC-CallsBarred  (19 ),
        roamingOutsidePLMNIC-CallsBarred  (20 ),
        roamingOutsidePLMNICountryIC-CallsBarred  (21 ),
        roamingOutsidePLMN-Barred  (22 ),
        roamingOutsidePLMN-CountryBarred  (23 ),
        registrationAllCF-Barred  (24 ),
        registrationCFNotToHPLMN-Barred  (25 ),
        registrationInterzonalCF-Barred  (26 ),
        registrationInterzonalCFNotToHPLMN-Barred  (27 ),
        registrationInternationalCF-Barred  (28 )} ( 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   [3] 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,
    d-CSI                         [2] IMPLICIT SEQUENCE {
      dp-AnalysedInfoCriteriaList   [0] IMPLICIT 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,
      camelCapabilityHandling       [1] IMPLICIT INTEGER ( 1 .. 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,
      notificationToCSE           [3] IMPLICIT NULL OPTIONAL,
      csi-Active                  [4] IMPLICIT NULL OPTIONAL,
      ... } OPTIONAL,
    t-CSI                         [3] 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    [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,
    vt-CSI                        [5] 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   [6] 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                       [7] IMPLICIT NULL OPTIONAL,
  tif-CSI-NotificationToCSE       [8] IMPLICIT NULL OPTIONAL,
  gprs-CSI                      [9] 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,
        ... } OPTIONAL,
```

```
    camelCapabilityHandling   [1] IMPLICIT INTEGER ( 1 .. 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,
    notificationToCSE         [3] IMPLICIT NULL OPTIONAL,
    csi-Active                [4] IMPLICIT NULL OPTIONAL,
    ... } OPTIONAL,
  sms-CSI                     [10] 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,
        ... } OPTIONAL,
    camelCapabilityHandling   [1] IMPLICIT INTEGER ( 1 .. 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,
    notificationToCSE         [3] IMPLICIT NULL OPTIONAL,
    csi-Active                [4] IMPLICIT NULL OPTIONAL,
    ... } OPTIONAL,
  ss-CSI                      [11] 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,
      ... },
    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    [0] IMPLICIT NULL OPTIONAL,
        csi-Active           [1] IMPLICIT NULL OPTIONAL} OPTIONAL,
      m-CSI                       [12] 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,
        csi-Active           [3] IMPLICIT NULL OPTIONAL,
        ... } 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,
      ... ,
      specificCSIDeletedList          [14] IMPLICIT BIT STRING {
        o-csi  (0 ),
        ss-csi  (1 ),
        tif-csi  (2 ),
        d-csi  (3 ),
        vt-csi  (4 ),
        sms-csi  (5 ),
        m-csi  (6 ),
        gprs-csi  (7 ),
        t-csi  (8 )} ( SIZE( 8 .. 32 ) ) OPTIONAL} OPTIONAL,
    allInformationSent       [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,
    ... }
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 {
    -- dataMissing -- localValue : 35,
    -- 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,
      ... ,
      b-subscriberNumber      [5] IMPLICIT OCTET STRING ( SIZE( 1 .. 20 ) ) ( SIZE( 1 .. 9 ) ) OPTIONAL,
      ccbs-RequestState       [6] IMPLICIT ENUMERATED {
        request      ( 0 ),
        recall       ( 1 ),
        active       ( 2 ),
        completed    ( 3 ),
        suspended    ( 4 ),
        frozen       ( 5 ),
        deleted      ( 6 ) } 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 ) ),
      asciCallReference    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,
      ... ,
      stateAttributes          [5] IMPLICIT SEQUENCE {
        downlinkAttached    [5] IMPLICIT NULL OPTIONAL,
        uplinkAttached      [6] IMPLICIT NULL OPTIONAL,
        dualCommunication   [7] IMPLICIT NULL OPTIONAL,
        callOriginator      [8] IMPLICIT NULL 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,
      informPreviousNetworkEntity   [1] IMPLICIT NULL 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,
      ... ,
      supportedGADShapes   [9] IMPLICIT BIT STRING {
        ellipsoidPoint  (0 ),
        ellipsoidPointWithUncertaintyCircle  (1 ),
        ellipsoidPointWithUncertaintyEllipse  (2 ),
        polygon  (3 ),
        ellipsoidPointWithAltitude  (4 ),
        ellipsoidPointWithAltitudeAndUncertaintyElipsoid  (5 ),
        ellipsoidArc  (6 )} ( SIZE( 7 .. 16 ) ) 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,
      ... ,
      add-LocationEstimate    [2] IMPLICIT OCTET STRING ( SIZE( 1 .. 91 ) ) 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,
      slr-ArgExtensionContainer  [7] IMPLICIT SEQUENCE {
        privateExtensionList    [0] IMPLICIT SEQUENCE  ( SIZE( 1 .. 10 ) ) OF
          SEQUENCE {
            extId     MAP-EXTENSION .&extensionId  ( {
              ,
              ...} ) ,
            extType   MAP-EXTENSION .&ExtensionType  ( {
              ,
              ...} { @extId  } )  OPTIONAL} OPTIONAL,
        slr-Arg-PCS-Extensions  [1] IMPLICIT SEQUENCE {
          ... ,
          na-ESRK-Request  [0] IMPLICIT NULL OPTIONAL} OPTIONAL,
        ... } OPTIONAL,
      ... ,
      add-LocationEstimate      [8] IMPLICIT OCTET STRING ( SIZE( 1 .. 91 ) ) 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,
      ... ,
      na-ESRK             [0] IMPLICIT OCTET STRING ( SIZE( 1 .. 20 ) ) ( SIZE( 1 .. 9 ) ) 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

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 ),
        ... ,
        npdbMismatch              ( 2 ) } 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

targetCellOutsideGroupCallArea ERROR
  PARAMETER
    targetCellOutsideGCA-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 : 42

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 ),
         positionMethodNotAvailableInNetwork        ( 7 ),
         positionMethodNotAvailableInLocationArea   ( 8 ),
         ...  } 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

unknownOrUnreachableLCSClient ERROR
  PARAMETER
    unknownOrUnreachableLCSClient-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       R5.70  (Production_5.70)
--           Date: 2004-06-09 Time: 15:33:12

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:
Void

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	REV	PHASE	CAT	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
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 RoamingRestrictionDueTo UnsupportedFeature	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
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 3GPP 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
			3.3.0					New version created to fix a CR implementation error	3.3.1	
07	N2B000436	29.002	3.3.1	048	5	R99	B	Introduction of Multicall	3.4.0	Multicall
07	N2B000319	29.002	3.3.1	059	1	R99	B	Alternative solution for ALR	3.4.0	CAMEL phase 3
07	N2B000461	29.002	3.3.1	063	4	R99	B	MNP Database Mismatch	3.4.0	MNP
07	N2B000375	29.002	3.3.1	066	5	R99	B	Addition of the FTN-AddressString	3.4.0	Call Forwarding Enhancements
07	N2B000456	29.002	3.3.1	079	4	R99	C	Correction of SS Invocation Notification for CCBS	3.4.0	CAMEL Phase 3
07	N2A000023	29.002	3.3.1	080		R99	F	Corrections to ATSI, ATM, NCSD	3.4.0	CAMEL Phase 3
07	N2B000046	29.002	3.3.1	083		R99	A	Privacy notification/verification for call related privacy class	3.4.0	Location Services (LCS)
07	N2B000142	29.002	3.3.1	084	2	R99	B	Addition of CS Allocation/retention priority	3.4.0	QoS enhancements
07	N2B000144	29.022	3.3.1	086	1	R99	D	Editorial cleanup of 29.002	3.4.0	TEI
07	N2B000100	29.002	3.3.1	087		R99	A	Correction of LSA information	3.4.0	SoLSA
07	N2B000067	29.002	3.3.1	089		R99	F	Security interworking between release 99 and pre-99 MSC/VLRs	3.4.0	Security
07	N2B000113	29.002	3.3.1	090	1	R99	B	Improving GPRS charging efficiency	3.4.0	GPRS
07	N2B000120	29.002	3.3.1	094	2	R99	C	QoS-Subscribed field enhancements	3.4.0	QoS enhancements
07	N2B000322	29.002	3.3.1	095	1	R99	C	RANAP support on the E-interface	3.4.0	Handover
07	N2B000191	29.002	3.3.1	099		R99	B	UMTS Authentication	3.4.0	Security
07	N2B000466	29.002	3.3.1	100	5	R99	C	Support of 3G Handover, including Multicall	3.4.0	Multicall
07	N2B000372	29.002	3.3.1	101	1	R99	B	Introduction of Service Area Identification	3.4.0	TEI
07	N2B000380	29.002	3.3.1	102	2	R99	F	Clarification on Authentication Info Retrieval	3.4.0	Security
07	N2B000330	29.002	3.3.1	103	1	R99	B	Addition of UMTS security to MAP B interface	3.4.0	Security
07	N2B000244	29.002	3.3.1	104		R99	F	Re-Synchronisation Info	3.4.0	UMTS Security

07	N2B000324	29.002	3.3.1	105	1	R99	C	Introduction of additional service parameters for inter-system handover	3.4.0	Handover
07	N2B000281	29.002	3.3.1	107		R99	D	Removal of architectural information from clause 4	3.4.0	TEI
07	N2-000454	29.002	3.3.1	110	1	R99	B	Introduction of Authentication Failure Report	3.4.0	Security
07	N2B000357	29.002	3.3.1	111		R99	B	Use of MAP private extensions to implement region-specific requirements	3.4.0	TEI
07	N2B000470	29.002	3.3.1	112		R99	A	Prioritisation of MAP application context related to VGCS/VBS	3.4.0	ASCI Phase 2
07	N2B000472	29.002	3.3.1	113		R99	F	Correction of SS-Codes for LCS	3.4.0	LCS
08	N4-000098	29.002	3.4.0	115	1	R99	F	Minor corrections to CAMEL3 NSDC/ATM/ATSI information flows	3.5.0	CAMEL Phase 3
08	N4-000094	29.002	3.4.0	117	1	R99	A	Using DSD to delete CCBS-B from the subscriber	3.5.0	CCBS
08	N4-000089	29.002	3.4.0	118	1	R99	F	Indication in PRN of support of Long FTNs	3.5.0	CF enhancements
08	N4-000073	29.002	3.4.0	120	1	R99	F	QoS-Subscribed field enhancements	3.5.0	QoS enhancements
08	N4-000050	29.002	3.4.0	121		R99	F	Correction of introduction of additional service parameters for inter-system handover	3.5.0	Handover/Relocation
08	N4-000100	29.002	3.4.0	122	2	R99	C	Proposed information flow on NSDC	3.5.0	CAMEL Phase 3
08	N4-000321	29.002	3.4.0	124	3	R99	C	CAMEL Subscription Info	3.5.0	CAMEL Phase 3
08	N4-000068	29.002	3.4.0	125		R99	A	Clarification to GMLC List definition	3.5.0	LCS
08	N4-000320	29.002	3.4.0	127	1	R99	F	Optionality of parameters in d-csi and in sms-csi	3.5.0	CAMEL Phase 3
08	N4-000209	29.002	3.4.0	130		R99	F	Version 3 tags for handover messages	3.5.0	Handover
08	N4-000211	29.002	3.4.0	132		R99	A	Correction of version handling at dialogue establishment	3.5.0	TEI
08	N4-000357	29.002	3.4.0	133	1	R99	F	Various corrections and/or cleanup to 29.002	3.5.0	TEI
08	N4-000217	29.002	3.4.0	134		R99	A	Correction of errors in Figure 25.1/1: Macro Receive_Open_Ind	3.5.0	TEI
08	N4-000326	29.002	3.4.0	135	1	R99	B	Addition of charging characteristics per PDP context	3.5.0	TEI
08	N4-000264	29.002	3.4.0	138		R99	F	Clarification of SAI-ack segmentation procedure	3.5.0	Security
08	N4-000392	29.002	3.4.0	139	1	R99	A	Indication of unsupported position method	3.5.0	LCS
08	N4-000276	29.002	3.4.0	141		R99	A	Clarification for ReportSM-DeliveryStatus operation	3.5.0	GPRS
08	N4-000349	29.002	3.4.0	142	1	R99	C	Addition of a parameter in the subsequent Handover from UMTS to GSM with Multicall	3.5.0	Multicall
08	N4-000278	29.002	3.4.0	143		R99	D	Editorial correction to MSC-A handover SDLs	3.5.0	Multicall
08	N4-000378	29.002	3.4.0	144	1	R99	A	Use of NAM parameter with MAP-INSERT-SUBSCRIBER-DATA service between HLR and SGSN	3.5.0	GPRS
08	N4-000293	29.002	3.4.0	145		R99	F	Addition of state attributes in Forward group call signalling	3.5.0	ASCI
08	N4-000294	29.002	3.4.0	146		R99	F	New user error 'target cell outside group call area' in MAP Prepare Handover message	3.5.0	ASCI
08	N4-000374	29.002	3.4.0	149		R99	A	Correction to the description of MAP-MO-Forward-Short-Message service	3.5.0	TEI
-		29.002	3.5.0			R99		Ver 3.5.1 created to allow for automatic update of Section 17 and Annexes A and B	3.5.1	-
-		29.002	3.5.1			R99		Ver 3.5.2 created to correct Table of Contents problem and missing headers.	3.5.2	-
09	N4-000428	29.002	3.5.2	151		R99	F	AUTS and AUTN parameter length	3.6.0	Security
09	N4-000743	29.002	3.5.2	154	3	R99	F	Clarification on Authentication Failure Report ack	3.6.0	Security

09	N4-000481	29.002	3.5.2	156		R99	F	Aligning 29.002 with 25.413 (UTRAN Iu Interface RANAP Signalling)	3.6.0	Handover
09	N4-000490	29.002	3.5.2	157		R99	A	Deletion of informative Annexe C	3.6.0	TEI
09	N4-000665	29.002	3.5.2	162	1	R99	F	Correction on Location Information	3.6.0	CAMEL phase 3
09	N4-000597	29.002	3.5.2	165		R99	F	Removal of LSAIdentity from NoteMM-EventArg	3.6.0	CAMEL phase 3
09	N4-000776	29.002	3.5.2	173	2	R99	F	Optionality of parameters in GPRS-CSI	3.6.0	CAMEL phase 3
09	N4-000787	29.002	3,5,2	175	1	R99	A	Correction to QoS indication	3.6.0	LCS
09	N4-000746	29.002	3.5.2	177	1	R99	F	Clarification of use of Radio Resource Information	3.6.0	Handover
09	N4-000748	29.002	3.5.2	179	2	R99	F	Correction to MSC-A handover SDLs	3.6.0	TEI
09	N4-000771	29.002	3.5.2	183		R99	F	LCS Support for CAMEL Phase 3	3.6.0	LCS
09	N4-000749	29.002	3.5.2	185	1	R99	F	Correction to MSC-A handover SDLs	3.6.0	TEI
09	N4-000778	29.002	3.5.2	187		R99	F	Clarification for segmentation of D-CSI and SMS-CSI	3.6.0	CAMEL phase 3
10	N4-000911	29.002	3.5.1	167	3	R99	A	Corrections and clarifications for USSD procedures on the HLR - gsmSCF interface	3.7.0	USSD
10	N4-000907	29.002	3.6.0	190	1	R99	F	Corrections of ISD data structure for CAMEL phase 3	3.7.0	CAMEL Phase 3
10	N4-001068	29.002	3.6.0	192	2	R99	F	USSD corrections for Follow Me	3.7.0	USSD
10	N4-001070	29.002	3.6.0	195	1	R99	F	GSM to 3G Handover: MAP parameter Target Cell ID	3.7.0	Handover
10	N4-000920	29.002	3.6.0	197		R99	F	ASN.1 description of targetCellId	3.7.0	Handover
10	N4-001072	29.002	3.6.0	199	1	R99	F	IMSI in MAP_PREPARE_HANDOVER	3.7.0	Handover
10	N4-001075	29.002	3.6.0	207	1	R99	F	Alignment of the Target RNC-ID	3.7.0	Handover
10	N4-001079	29.002	3.6.0	209		R99	F	Transport of long RANAP messages on MAP-E interface	3.7.0	Handover
10	N4-001088	29.002	3.6.0	210	1	R99	F	Export of GSN-Address data type	3.7.0	CAMEL Phase 3
-	-	29.002	3.7.0	-	-	R99	-	Automatic update of Annexes A and B	3.7.1	-
-	-	29.002	3.7.1	-	-	R99	-	Corrupted figures in chapter 25.9 fixed	3.7.2	-
11	N4-010035	29.002	3.7.2	205	1	R99	A	Correction to LCS application context	3.8.0	LCS
11	N4-010032	29.002	3.7.2	216		R99	F	Correction to maximum numbers of RAB's	3.8.0	Multicall
11	N4-010057	29.002	3.7.2	223		R99	A	Failure of Update GPRS Location when HLR is not reachable	3.8.0	GPRS R97
11	N4-010170	29.002	3.7.2	228		R99	F	Adding EXPORT definition for LSAIdentity	3.8.0	CAMEL 3
11	N4-010171	29.002	3.7.2	229		R99	F	Removing duplicate parameters from SS-CSI	3.8.0	CAMEL 3
11	N4-010172	29.002	3.7.2	230		R99	F	Correction to description of SS-CSI in HLR to VLR information flow	3.8.0	CAMEL 3
11	N4-010364	29.002	3.7.2	249		R99	F	GSM to UMTS handover: addition of MAP parameter RNC ID	3.8.0	Handover
11	N4-010392	29.002	3.7.2	251		R99	F	Clarification on the use of multicall bearer information	3.8.0	Multicall

11	N4-010427	29.002	3.7.2	257		R99	F	Adding EXPORT definition for GeographicalInformation	3.8.0	CAMEL 3
11	N4-010445	29.002	3.7.2	259		R99	A	Failure of Authentication Parameter GPRS when HLR is not reachable	3.8.0	GPRS R97
11	N4-010484	29.002	3.7.2	261	1	R99	F	Correction to D-CSI	3.8.0	CAMEL 3
12	N4-010727	29.002	3.8.0	225	4	R99	F	Addition of selected UMTS algorithm indication to the handover procedures	3.9.0	Handover
12	N4-010729	29.002	3.8.0	226	5	R99	F	Addition of allowed GSM algorithms indication to the handover procedures	3.9.0	Handover
12	N4-010732	29.002	3.8.0	242	4	R99	F	Addition of allowed UMTS algorithm indication to the handover procedures	3.9.0	Handover
12	N4-010734	29.002	3.8.0	243	4	R99	F	Addition of selected GSM algorithm indication to the handover procedures	3.9.0	Handover
12	N4-010738	29.002	3.8.0	253	2	R99	F	Addition of radio resource list to the handover procedures	3.9.0	Multicall
12	NP-010247	29.002	3.8.0	255	3	R99	F	Addition of GSM channel type and GSM chosen channel indication to handover procedures	3.9.0	Handover
12	N4-010786	29.002	3.8.0	263	3	R99	F	Add support in MAP for all shapes defined in 23.032	3.9.0	LCS
12	N4-010550	29.002	3.8.0	265		R99	A	Add support in MAP for Ellipsoid Point	3.9.0	LCS
12	N4-010632	29.002	3.8.0	269	1	R99	F	Correction to description of RNCId parameter	3.9.0	Handover
12	N4-010634	29.002	3.8.0	271	1	R99	F	Correction to Encryption Information and Integrity Protection Information parameters	3.9.0	Handover
12	N4-010766	29.002	3.8.0	278	3	R99	F	Essential drawbacks on service due to introduction of Super-Charger function	3.9.0	TEI
12	N4-010740	29.002	3.8.0	282	1	R99	F	Introduction of selected Rab-id to the Process Access Signalling operation	3.9.0	Multicall
12	N4-010672	29.002	3.8.0	284		R99	F	Mistake in the definition of AFR Application Context	3.9.0	SEC
13	N4-011016	29.002	3.9.0	303	3	R99	F	Minimum MAP application context for intersystem MSC handover from GSM to UMTS	3.10.0	Handover
13	N4-010973	29.002	3.9.0	305		R99	F	Addition of data type definitions to EXPORT statements for the usage in CAP	3.10.0	CAMEL3
13	N4-011018	29.002	3.9.0	308	2	R99	F	Minimum MAP application context for intersystem MSC handover from UMTS to GSM	3.10.0	Handover
14	N4-011030	29.002	3.10.0	312		R99	A	Clarification on LCS parameters in MAP	3.11.0	LCS
14	N4-011044	29.002	3.10.0	315		R99	F	Alignment of SDL with text for procedure Process_Components in the MAP protocol machine	3.11.0	TEI
14	N4-011197	29.002	3.10.0	317	1	R99	F	Indication of deletion of CSI in Notify Subscriber Data Change	3.11.0	CAMEL Phase 3
14	N4-011073	29.002	3.10.0	319		R99	F	Correct length of Add-GeographicalInformation	3.11.0	LCS

14	N4-011090	29.002	3.10.0	321		R99	F	Clarify encoding of RNC Id	3.11.0	Handover
14	N4-011093	29.002	3.10.0	323		R99	F	Clarify encoding of RANAP parameters in MAP	3.11.0	Handover
14	N4-011226	29.002	3.10.0	330	1	R99	F	Clarification of methodology for maintaining data consistency in Supercharger	3.11.0	TEI
14	N4-011172	29.002	3.10.0	333		R99	F	Addition of RAB ID to Prepare Handover procedure	3.11.0	Multicall
14	N4-011174	29.002	3.10.0	335		R99	F	Correction to the Allowed GSM Algorithms parameter	3.11.0	Handover
14	N4-011189	29.002	3.10.0	338		R99	F	CUG-Info is not exported from 29.002	3.11.0	CAMEL3
14	N4-011208	29.002	3.10.0	340		R99	F	Clarification on NSCD when data is withdrawn	3.11.0	CAMEL phase 3
14	N4-011210	29.002	3.10.0	342		R99	F	Clarification of sending CAMEL information in stand alone ISD case	3.11.0	CAMEL phase 3
14	N4-011272	29.002	3.10.0	346		R99	F	ASN.1 correction	3.11.0	CAMEL
14	N4-011432	29.002	3.10.0	353	1	R99	F	Minimum MAP application context for G2G inter-MSC handover	3.11.0	Handover
14	N4-011438	29.002	3.10.0	358	2	R99	F	Alignment of parameter lengths with those prescribed in 08.08	3.11.0	TEI
14	N4-011393	29.002	3.10.0	363		R99	F	Syntax error in the ATM result and ATSI result	3.11.0	CAMEL phase 3
15	N4-020209	29.002	3.11.0	371	1	R99	F	Inclusion of complete ODB data in ATSI and NSDC	3.12.0	CAMEL3
15	N4-020246	29.002	3.11.0	388	1	R99	F	Incomplete description of Restore Data parameters	3.12.0	TEI
15	N4-020181	29.002	3.11.0	401		R99	A	Clarification on CODEC-Info	3.12.0	TEI
15	N4-020225	29.002	3.11.0	405		R99	A	ODB alignment	3.12.0	TEI
16	N4-020505	29.002	3.12.0	426	1	R99	F	LCS : on error handling if shape not supported by GMLC	3.13.0	LCS1
16	N4-020620	29.002	3.12.0	451		R99	F	Addition of Radio Resource List to the Forward Access Signalling operation	3.13.0	Multicall
16	N4-020639	29.002	3.12.0	458		R99	F	Clarification on Resume Call Handling	3.13.0	TEI
16	N4-020744	29.002	3.12.0	438	2	R99	F	Clarification on SendAuthenticationInfo	3.13.0	TEI
16	N4-020775	29.002	3.12.0	444	2	R99	F	Addition of Service Handover parameters to MAP Handover messages	3.13.0	Handover
17	N4-021043	29.002	3.13.0	477		R99	F	Clarifications on Send Identification	3.14.0	TEI
18	N4-021288	29.002	3.14.0	494	1	R99	F	Correction to segmentation of O-CSI and T-CSI	3.15.0	CAMEL3
18	N4-021416	29.002	3.14.0	506		R99	F	ODB correction	3.15.0	CAMEL3
18	N4-021571	29.002	3.14.0	514	2	R99	F	Correction to the Service Handover parameters	3.15.0	Multicall
19	N4-030298	29.002	3.15.0	530	1	R99	F	Correction to the definitions of Radio Resource List and BSSMAP Service Handover List	3.16.0	Multicall
19	N4-030302	29.002	3.15.0	538	2	R99	F	Handover of Group Calls where MSC-B has bearer established	3.16.0	TEI

19	N4-030284	29.002	3.15.0	548	1	R99	A	Change of SS-Code List description for Insert Subscriber Data	3.16.0	TEI
19	NP-030068	29.002	3.15.0	542	3	R99	F	Correction to interactions between CAMEL control of MO SMS and barring	3.16.0	CAMEL3
20	NP-030214	29.002	3.16.0	630		R99	F	Missing SMSs over MSC even if the MS is capable of such sending	3.17.0	TEI
21	N4-0301050	29.002	3.17.0	661	1	R99	A	Correction of encoding description of Group-Id	3.18.0	ASCI
23	N4-040330	29.002	3.18.0	705	2	R99	A	Correction to Insert Subscriber Data message for LCS SS	3.19.0	LCS
24	N4-040517	29.002	3.19.0	712	1	R99	F	Introduction of North American Interim Location Based Routing of Emergency Call	3.20.0	LCS