

EUROPEAN
TELECOMMUNICATION
STANDARD

ETS 300 974

April 2000

Tenth Edition

Source: SMG

Reference: RE/SMG-030902QR9

ICS: 33.020

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



**Digital cellular telecommunications system (Phase 2+);
Mobile Application Part (MAP) specification
(GSM 09.02 version 5.14.1 Release 1996)**

ETSI

European Telecommunications Standards Institute

ETSI Secretariat

Postal address: F-06921 Sophia Antipolis CEDEX - FRANCE

Office address: 650 Route des Lucioles - Sophia Antipolis - Valbonne - FRANCE

Internet: secretariat@etsi.fr - <http://www.etsi.org>

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

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

© European Telecommunications Standards Institute 2000. All rights reserved.

Contents

Intellectual Property Rights	19
Foreword.....	19
1 Scope	21
1.1 Normative references	21
1.2 Abbreviations	27
2 Configuration of the mobile network.....	27
2.1 The entities of the mobile system	27
2.1.1 The Home Location Register (HLR)	27
2.1.2 The Visitor Location Register (VLR)	27
2.1.3 The Mobile-services Switching Centre (MSC)	28
2.1.4 The Base Station System (BSS)	28
2.1.5 The Gateway MSC (GMSC)	28
2.1.6 The SMS Gateway MSC	28
2.1.7 The SMS Interworking MSC	28
2.1.8 The VBS/VGCS Anchor MSC	29
2.1.9 The Equipment Identity Register (EIR)	29
2.1.10 The GSM Service Control Function (gsmSCF)	29
2.1.11 The Group Call Register (GCR)	29
2.2 Configuration of a Public Land Mobile Network (PLMN)	29
2.3 Interconnection between PLMNs	29
2.4 The interfaces within the mobile service	29
2.4.1 Interface between the HLR and the VLR (D-interface)	29
2.4.2 Interface between the HLR and the gsmSCF (J-interface)	30
2.4.3 Interface between the VLR and its associated MSC(s) (B-interface)	30
2.4.4 Interface between VLRs (G-interface)	30
2.4.5 Interface between the HLR and the MSC (C-interface)	30
2.4.6 Interface between MSCs (E-interface)	30
2.4.7 Interface between the MSC and Base Station Systems (A-interface)	30
2.4.8 Interface between MSC and EIR (F-interface)	30
2.4.9 Interface between VBS/VGCS Anchor MSC and GCR (I-interface)	30
2.5 Splitting of the data storage	31
3 Overload and compatibility overview	32
3.1 Overload control	32
3.1.1 Overload control for MSC (outside MAP)	32
3.1.2 Overload control for MAP entities	32
3.1.3 Congestion control for Signalling System No. 7	35
3.2 Compatibility	35
3.2.1 General	35
3.2.2 Strategy for selecting the Application Context (AC) version	35
3.2.2.1 Proposed method	35
3.2.2.2 Managing the version look-up table	36
3.2.2.3 Optimizing the method	37
4 Requirements concerning the use of SCCP and TC	38
4.1 Use of SCCP	38
4.1.1 SCCP Class	38
4.1.2 Sub-System Number (SSN)	38
4.1.3 SCCP addressing	38
4.1.3.1 Introduction	38
4.1.3.2 The Mobile-services Switching Centre (MSC)	39
4.1.3.2.1 MSC interaction during handover	39
4.1.3.2.2 MSC for short message routing	39
4.1.3.3 The Home Location Register (HLR)	39

	4.1.3.3.1	During call set-up	39
	4.1.3.3.2	Before location updating completion	40
	4.1.3.3.3	After location updating completion	40
	4.1.3.3.4	VLR restoration	40
	4.1.3.4	The Visitor Location Register (VLR).....	41
	4.1.3.4.1	Inter-VLR information retrieval	41
	4.1.3.4.2	HLR request	41
	4.1.3.5	The Interworking MSC (IWMSC) for Short Message Service	41
	4.1.3.6	The Equipment Identity Register (EIR)	41
	4.1.3.7	Summary table	41
4.2	Use of TC		42
5	General on MAP services		44
5.1	Terminology and definitions		44
5.2	Modelling principles		44
5.3	Common MAP services.....		44
	5.3.1	MAP-OPEN service	45
	5.3.2	MAP-CLOSE service	47
	5.3.3	MAP-DELIMITER service	48
	5.3.4	MAP-U-ABORT service	48
	5.3.5	MAP-P-ABORT service	49
	5.3.6	MAP-NOTICE service.....	50
5.4	Sequencing of services		50
5.5	General rules for mapping of services onto TC.....		52
	5.5.1	Mapping of common services.....	52
	5.5.2	Mapping of user specific services.....	53
5.6	Definition of parameters		54
	5.6.1	Common parameters.....	56
	5.6.1.1	Invoke Id.....	56
	5.6.1.2	Linked Id.....	56
	5.6.1.3	Provider error	56
	5.6.1.4	User error	56
	5.6.2	Numbering and identification parameter.....	58
	5.6.2.1	IMSI	58
	5.6.2.2	TMSI.....	58
	5.6.2.3	IMEI	58
	5.6.2.4	Previous location area Id.....	58
	5.6.2.5	Stored location area Id.....	59
	5.6.2.6	Current location area Id.....	59
	5.6.2.7	Target location area Id	59
	5.6.2.8	Target cell Id	59
	5.6.2.9	[Spare].....	59
	5.6.2.10	Originating entity number	59
	5.6.2.11	MSC number	59
	5.6.2.12	Target MSC number	59
	5.6.2.13	HLR number	59
	5.6.2.14	VLR number	59
	5.6.2.15	HLR Id	59
	5.6.2.16	LMSI.....	59
	5.6.2.17	MS ISDN	59
	5.6.2.18	OMC Id.....	59
	5.6.2.19	Roaming number	59
	5.6.2.20	[Spare].....	60
	5.6.2.21	Handover number	60
	5.6.2.22	Forwarded-to number.....	60
	5.6.2.23	Forwarded-to subaddress	60
	5.6.2.24	Called number.....	60
	5.6.2.25	Calling number	60
	5.6.2.26	Originally dialled number.....	60
	5.6.2.27	Service centre address	60
	5.6.2.28	Zone Code	60
	5.6.2.29	MSISdn-Alert	60

	5.6.2.30	Location Information	60
	5.6.2.31	GMSC Address	60
	5.6.2.32	VMSC Address	60
	5.6.2.33	Group Id	60
5.6.3		Subscriber management parameters	61
	5.6.3.1	Category	61
	5.6.3.2	Equipment status	61
	5.6.3.3	Extensible Bearer service	61
	5.6.3.4	Extensible Teleservice	61
	5.6.3.5	Extensible Basic Service Group	61
	5.6.3.6	GSM bearer capability	61
	5.6.3.7	Subscriber Status	61
	5.6.3.8	CUG Outgoing Access indicator	61
	5.6.3.9	Operator Determined Barring General Data	61
	5.6.3.10	ODB HPLMN Specific Data	62
	5.6.3.11	Regional Subscription Data	62
	5.6.3.12	Regional Subscription Response	62
	5.6.3.13	Roaming Restriction Due To Unsupported Feature	62
	5.6.3.14	Extensible SS-Info	62
	5.6.3.15	Extensible Forwarding information	62
	5.6.3.16	Extensible Forwarding feature	62
	5.6.3.17	Extensible SS-Status	63
	5.6.3.18	Extensible Forwarding Options	63
	5.6.3.19	Extensible No reply condition timer	63
	5.6.3.20	Extensible Call barring information	63
	5.6.3.21	Extensible Call barring feature	63
	5.6.3.22	CUG info	63
	5.6.3.23	CUG subscription	63
	5.6.3.24	CUG interlock	63
	5.6.3.25	CUG index	63
	5.6.3.26	CUG feature	64
	5.6.3.27	Inter CUG options	64
	5.6.3.28	Intra CUG restrictions	64
	5.6.3.29	Extensible SS-Data	64
	5.6.3.30	Subscriber State	64
	5.6.3.31	Requested Info	64
	5.6.3.32	Suppression of Announcement	64
	5.6.3.33	Suppress T-CSI	65
	5.6.3.34	GMSC CAMEL Subscription Info	65
	5.6.3.35	VLR CAMEL Subscription Info	65
	5.6.3.36	Supported CAMEL Phases	65
	5.6.3.37	CUG Subscription Flag	65
	5.6.3.38	CAMEL Subscription Info Withdraw	65
	5.6.3.39	Voice Group Call Service (VGCS) Data	65
	5.6.3.40	Voice Broadcast Service (VBS) Data	65
5.6.4		Supplementary services parameters	65
	5.6.4.1	SS-Code	65
	5.6.4.2	SS-Status	65
	5.6.4.3	SS-Data	66
	5.6.4.4	Override Category	66
	5.6.4.5	CLI Restriction Option	66
	5.6.4.6	Forwarding Options	66
	5.6.4.7	No reply condition timer	66
	5.6.4.8 - 5.6.4.14	[Spare]	66
	5.6.4.15	Forwarding information	66
	5.6.4.16	Forwarding feature	66
	5.6.4.17	[Spare]	67
	5.6.4.18	Call barring information	67
	5.6.4.19	Call barring feature	67
	5.6.4.20	New password	67
	5.6.4.21	Current password	67
	5.6.4.22	Guidance information	67
	5.6.4.23	[Spare]	67

	5.6.4.24	SS-Info	67
	5.6.4.25 - 5.6.4.35	[Spare]	68
	5.6.4.36	USSD Data Coding Scheme	68
	5.6.4.37	USSD String	68
	5.6.4.38	Bearer service	68
	5.6.4.39	Teleservice	68
	5.6.4.40	Basic Service Group	68
	5.6.4.41	eMLPP information	68
5.6.5	Call parameters		68
	5.6.5.1	Call reference number	68
	5.6.5.2	Interrogation type	68
	5.6.5.3	OR interrogation	69
	5.6.5.4	OR capability	69
	5.6.5.5	Forwarding reason	69
	5.6.5.6	Forwarding interrogation required	69
	5.6.5.7	O-CSI	69
5.6.6	Radio parameters		69
	5.6.6.1 - 5.6.6.6	[Spare]	69
	5.6.6.7	HO-Number Not Required	69
5.6.7	Authentication parameters		69
	5.6.7.1	Authentication set list	69
	5.6.7.2	Rand	69
	5.6.7.3	Sres	69
	5.6.7.4	Kc	69
	5.6.7.5	[Spare]	70
	5.6.7.6	Cksn	70
	5.6.7.7	Ciphering mode	70
5.6.8	Short message parameters		70
	5.6.8.1	SM-RP-DA	70
	5.6.8.2	SM-RP-OA	70
	5.6.8.3	MWD status	70
	5.6.8.4	SM-RP-UI	70
	5.6.8.5	SM-RP-PRI	70
	5.6.8.6	SM Delivery Outcome	70
	5.6.8.7	More Messages To Send	71
	5.6.8.8	Alert Reason	71
	5.6.8.9	Absent Subscriber Diagnostic SM	71
5.6.9	Access and signalling system related parameters		71
	5.6.9.1	BSS-apdu	71
	5.6.9.2	CM service type	71
	5.6.9.3	Access connection status	71
	5.6.9.4	External Signal Information	71
	5.6.9.5	Access signalling information	71
	5.6.9.6	Location update type	71
	5.6.9.7	Protocol ID	72
	5.6.9.8	Network signal information	72
5.6.10	System operations parameters		72
	5.6.10.1	Network resources	72
	5.6.10.2	Trace reference	73
	5.6.10.3	Trace type	73
5.7	Representation of a list of a basic parameter in service-primitives		73
6	Mobility services		74
6.1	Location management services		74
6.1.1	MAP_UPDATE_LOCATION_AREA service		74
	6.1.1.1	Definition	74
	6.1.1.2	Service primitives	74
	6.1.1.3	parameter definitions and use	74
6.1.2	MAP_UPDATE_LOCATION service		75
	6.1.2.1	Definition	75
	6.1.2.2	Service primitives	75
	6.1.2.3	Parameter definitions and use	75
6.1.3	MAP_CANCEL_LOCATION service		76

	6.1.3.1	Definition.....	76
	6.1.3.2	Service primitives.....	76
	6.1.3.3	Parameter definitions and use.....	77
6.1.4		MAP_SEND_IDENTIFICATION service.....	77
	6.1.4.1	Definition.....	77
	6.1.4.2	Service primitives.....	77
	6.1.4.3	Parameter definitions and use.....	77
6.1.5		MAP_DETACH_IMSI service.....	78
	6.1.5.1	Definition.....	78
	6.1.5.2	Service primitives.....	78
	6.1.5.3	Parameter definitions and use.....	78
6.1.6		MAP_PURGE_MS service.....	79
	6.1.6.1	Definition.....	79
	6.1.6.2	Service primitives.....	79
	6.1.6.3	Parameter definitions and use.....	79
6.2		Paging and search.....	79
	6.2.1	MAP_PAGE service.....	79
		6.2.1.1 Definition.....	79
		6.2.1.2 Service primitives.....	80
		6.2.1.3 Parameter definitions and use.....	80
	6.2.2	MAP_SEARCH_FOR_MS service.....	80
		6.2.2.1 Definition.....	80
		6.2.2.2 Service primitives.....	81
		6.2.2.3 Parameter definitions and use.....	81
6.3		Access management services.....	81
	6.3.1	MAP_PROCESS_ACCESS_REQUEST service.....	81
		6.3.1.1 Definition.....	81
		6.3.1.2 Service primitives.....	82
		6.3.1.3 Parameter definitions and use.....	82
6.4		Handover services.....	83
	6.4.1	MAP_PREPARE_HANDOVER service.....	83
		6.4.1.1 Definition.....	83
		6.4.1.2 Service primitives.....	83
		6.4.1.3 Parameter use.....	83
	6.4.2	MAP_SEND_END_SIGNAL service.....	84
		6.4.2.1 Definition.....	84
		6.4.2.2 Service primitives.....	84
		6.4.2.3 Parameter use.....	84
	6.4.3	MAP_PROCESS_ACCESS_SIGNALLING service.....	85
		6.4.3.1 Definition.....	85
		6.4.3.2 Service primitives.....	85
		6.4.3.3 Parameter use.....	85
	6.4.4	MAP_FORWARD_ACCESS_SIGNALLING service.....	85
		6.4.4.1 Definition.....	85
		6.4.4.2 Service primitives.....	85
		6.4.4.3 Parameter use.....	86
	6.4.5	MAP_PREPARE_SUBSEQUENT_HANDOVER service.....	86
		6.4.5.1 Definition.....	86
		6.4.5.2 Service primitives.....	86
		6.4.5.3 Parameter use.....	86
	6.4.6	MAP_ALLOCATE_HANDOVER_NUMBER service.....	87
		6.4.6.1 Definition.....	87
		6.4.6.2 Service primitives.....	87
		6.4.6.3 Parameter use.....	87
	6.4.7	MAP_SEND_HANDOVER_REPORT service.....	87
		6.4.7.1 Definition.....	87
		6.4.7.2 Service primitives.....	87
		6.4.7.3 Parameter use.....	88
6.5		Authentication management services.....	88
	6.5.1	MAP_AUTHENTICATE service.....	88
		6.5.1.1 Definition.....	88
		6.5.1.2 Service primitives.....	88
		6.5.1.3 Parameter use.....	88

6.5.2	MAP_SEND_AUTHENTICATION_INFO service	89
6.5.2.1	Definition	89
6.5.2.2	Service primitives	89
6.5.2.3	Parameter use.....	89
6.6	Security management services	90
6.6.1	MAP_SET_CIPHERING_MODE service.....	90
6.6.1.1	Definitions	90
6.6.1.2	Service primitives	90
6.6.1.3	Parameter use.....	90
6.7	International mobile equipment identities management services.....	90
6.7.1	MAP_CHECK_IMEI service.....	90
6.7.1.1	Definition	90
6.7.1.2	Service primitives	90
6.7.1.3	Parameter use.....	91
6.7.2	MAP_OBTAIN_IMEI service.....	91
6.7.2.1	Definition	91
6.7.2.2	Service primitives	92
6.7.2.3	Parameter use.....	92
6.8	Subscriber management services.....	92
6.8.1	MAP-INSERT-SUBSCRIBER-DATA service.....	92
6.8.1.1	Definition	92
6.8.1.2	Service primitives	93
6.8.1.3	Parameter use.....	93
6.8.1.4	Supported CAMEL Phases	98
6.8.1.4	Basic service information related to supplementary services	99
6.8.2	MAP-DELETE-SUBSCRIBER-DATA service	99
6.8.2.1	Definition	99
6.8.2.2	Service primitives	100
6.8.2.3	Parameter use.....	100
6.9	Identity management services.....	101
6.9.1	MAP-PROVIDE-IMSI service.....	101
6.9.1.1	Definition	101
6.9.1.2	Service primitives	102
6.9.1.3	Parameter use.....	102
6.9.2	MAP-FORWARD-NEW-TMSI service	102
6.9.2.1	Definition	102
6.9.2.2	Service primitives	102
6.9.2.3	Parameter use.....	102
6.10	Fault recovery services	102
6.10.1	MAP_RESET service.....	102
6.10.1.1	Definition	102
6.10.1.2	Service primitives	103
6.10.1.3	Parameter definition and use	103
6.10.2	MAP_FORWARD_CHECK_SS_INDICATION service.....	103
6.10.2.1	Definition	103
6.10.2.2	Service primitives	103
6.10.2.3	Parameter definition and use	103
6.10.3	MAP_RESTORE_DATA service.....	103
6.10.3.1	Definition	103
6.10.3.2	Service primitives	104
6.10.3.3	Parameter definitions and use	104
6.11	Subscriber Information services.....	105
6.11.1	MAP-ANY-TIME-INTERROGATION service	105
6.11.1.1	Definition	105
6.11.1.2	Service primitives	105
6.11.1.3	Parameter definition and use	105
6.11.2	MAP-PROVIDE-SUBSCRIBER-Info service	106
6.11.2.1	Definition	106
6.11.2.2	Service primitives	106
6.11.2.3	Parameter definition and use	106
7	Operation and maintenance services	107

7.1	Subscriber tracing services.....	107
7.1.1	MAP-ACTIVATE-TRACE-MODE service.....	107
7.1.1.1	Definition.....	107
7.1.1.2	Service primitives.....	107
7.1.1.3	Parameter use.....	107
7.1.2	MAP-DEACTIVATE-TRACE-MODE service.....	108
7.1.2.1	Definition.....	108
7.1.2.2	Service primitives.....	108
7.1.2.3	Parameter use.....	108
7.1.3	MAP-TRACE-SUBSCRIBER-ACTIVITY service.....	108
7.1.3.1	Definition.....	108
7.1.3.2	Service primitives.....	109
7.1.3.3	Parameter use.....	109
7.2	Other operation and maintenance services.....	109
7.2.1	MAP-SEND-IMSI service.....	109
7.2.1.1	Definition.....	109
7.2.1.2	Service primitives.....	109
7.2.1.3	Parameter use.....	110
8	Call handling services.....	110
8.1	MAP_SEND_ROUTING_INFORMATION service.....	110
8.1.1	Definition.....	110
8.1.2	Service primitives.....	110
8.1.3	Parameter use.....	111
8.2	MAP_PROVIDE_ROAMING_NUMBER service.....	114
8.2.1	Definition.....	114
8.2.2	Service primitives.....	114
8.2.3	Parameter use.....	114
8.3	MAP_RESUME_CALL_HANDLING service.....	116
8.3.1	Definition.....	116
8.3.2	Service primitives.....	116
8.3.3	Parameter use.....	116
9	Supplementary services related services.....	117
9.1	MAP_REGISTER_SS service.....	117
9.1.1	Definition.....	117
9.1.2	Service primitives.....	117
9.1.3	Parameter use.....	117
9.2	MAP_ERASE_SS service.....	118
9.2.1	Definition.....	118
9.2.2	Service primitives.....	119
9.2.3	Parameter use.....	119
9.3	MAP_ACTIVATE_SS service.....	120
9.3.1	Definition.....	120
9.3.2	Service primitives.....	120
9.3.3	Parameter use.....	120
9.4	MAP_DEACTIVATE_SS service.....	121
9.4.1	Definitions.....	121
9.4.2	Service primitives.....	122
9.4.3	Parameter use.....	122
9.5	MAP_INTERROGATE_SS service.....	123
9.5.1	Definitions.....	123
9.5.2	Service primitives.....	123
9.5.3	Parameter use.....	123
9.6	MAP_INVOKE_SS service.....	126
9.6.1	Definitions.....	126
9.6.2	Service primitives.....	126
9.6.3	Parameter use.....	126
9.7	MAP_REGISTER_PASSWORD service.....	127
9.7.1	Definitions.....	127
9.7.2	Service primitives.....	127
9.7.3	Parameter use.....	127
9.8	MAP_GET_PASSWORD service.....	127

	9.8.1	Definitions	127
	9.8.2	Service primitives.....	128
	9.8.3	Parameter use	128
9.9		MAP_PROCESS_UNSTRUCTURED_SS_REQUEST service	128
	9.9.1	Definitions	128
	9.9.2	Service primitives.....	128
	9.9.3	Parameter use	129
9.10		MAP_UNSTRUCTURED_SS_REQUEST service.....	129
	9.10.1	Definitions	129
	9.10.2	Service primitives.....	129
	9.10.3	Parameter use	130
9.11		MAP_UNSTRUCTURED_SS_NOTIFY service	130
	9.11.1	Definitions	130
	9.11.2	Service primitives.....	130
	9.11.3	Parameter use	131
10		Short message service management services.....	131
10.1		MAP-SEND-ROUTING-INFO-FOR-SM service.....	131
	10.1.1	Definition	131
	10.1.2	Service primitives.....	132
	10.1.3	Parameter use	132
10.2		MAP-MO-FORWARD-SHORT-MESSAGE service	133
	10.2.1	Definition	133
	10.2.2	Service primitives.....	133
	10.2.3	Parameter use	133
10.3		MAP-REPORT-SM-DELIVERY-STATUS service	134
	10.3.1	Definition	134
	10.3.2	Service primitives.....	134
	10.3.3	Parameter use	134
10.4		MAP-READY-FOR-SM service	135
	10.4.1	Definition	135
	10.4.2	Service primitives.....	136
	10.4.3	Parameter use	136
10.5		MAP-ALERT-SERVICE-CENTRE service	136
	10.5.1	Definition	136
	10.5.2	Service primitives.....	137
	10.5.3	Parameter use	137
10.6		MAP-INFORM-SERVICE-CENTRE service.....	137
	10.6.1	Definition	137
	10.6.2	Service primitives.....	138
	10.6.3	Parameter use	138
10.7		MAP-SEND-INFO-FOR-MT-SMS service.....	138
	10.7.1	Definition	138
	10.7.2	Service primitives.....	138
	10.7.3	Parameter use	138
10.8		MAP-SEND-INFO-FOR-MO-SMS service	139
	10.8.1	Definition	139
	10.8.2	Service primitives.....	139
	10.8.3	Parameter use	139
10.9		MAP-MT-FORWARD-SHORT-MESSAGE service.....	140
	10.9.1	Definition	140
	10.9.2	Service primitives.....	140
	10.9.3	Parameter use	140
11		General	142
	11.1	Overview.....	142
	11.2	Underlying services	142
	11.3	Model.....	142
	11.4	Conventions	142
12		Elements of procedure.....	143
	12.1	Dialogue establishment.....	143
	12.1.1	Handling of unknown operations	143

12.1.2	Receipt of a MAP-OPEN request primitive.....	144
12.1.3	Receipt of a TC-BEGIN indication.....	144
12.1.4	Receipt of a MAP-OPEN response	147
12.1.5	Receipt of the first TC-CONTINUE ind.....	147
12.1.6	Receipt of a TC-END ind.....	147
12.1.7	Receipt of a TC-U-ABORT ind.....	147
12.1.8	Receipt of a TC-P-ABORT ind	148
12.2	Dialogue continuation.....	148
12.2.1	Sending entity.....	148
12.2.2	Receiving entity	148
12.3	Dialogue termination	148
12.3.1	Receipt of a MAP-CLOSE request	148
12.3.2	Receipt of a TC-END indication	149
12.4	User Abort.....	149
12.4.1	MAP-U-ABORT request	149
12.4.2	TC-U-ABORT ind	149
12.5	Provider Abort.....	149
12.5.1	MAP PM error situation	149
12.5.2	TC-P-ABORT ind.....	149
12.5.3	TC-U-ABORT ind	150
12.6	Procedures for MAP specific services	150
12.6.1	Service invocation	150
12.6.2	Service invocation receipt.....	150
12.6.3	Service response.....	151
12.6.4	Receipt of a response	152
12.6.4.1	Receipt of a TC-RESULT-NL indication	152
12.6.4.2	Receipt of a TC-RESULT-L indication.....	152
12.6.4.3	Receipt of a TC-U-ERROR indication	152
12.6.4.4	Receipt of a TC-INVOKE indication.....	153
12.6.4.5	Receipt of a TC-U-REJECT indication	153
12.6.4.6	Receipt of a TC-L-REJECT indication.....	153
12.6.4.7	Receipt of a TC-L-CANCEL indication.....	154
12.6.4.8	Receipt of a TC-NOTICE indication.....	154
12.6.5	Other events.....	154
12.6.5.1	Receipt of a TC-U-REJECT.....	154
12.6.5.2	Receipt of a TC-R-REJECT indication	155
12.6.5.3	Receipt of a TC-L-REJECT indication	155
12.6.6	Parameter checks	155
12.6.7	Returning state machines to idle	155
12.6.8	Load control.....	156
13	Mapping on to TC services.....	156
13.1	Dialogue control.....	156
13.1.1	Directly mapped parameters	156
13.1.2	Use of other parameters of dialogue handling primitives	156
13.1.2.1	Dialogue Id.....	156
13.1.2.2	Application-context-name	156
13.1.2.3	User information	156
13.1.2.4	Component present	156
13.1.2.5	Termination.....	156
13.1.2.6	P-Abort-Cause	156
13.1.2.7	Quality of service	157
13.2	Service specific procedures	157
13.2.1	Directly mapped parameters	157
13.2.2	Use of other parameters of component handling primitives.....	157
13.2.2.1	Dialogue Id.....	157
13.2.2.2	Class.....	157
13.2.2.3	Linked Id	158
13.2.2.4	Operation	158
13.2.2.5	Error.....	159
13.2.2.6	Parameters	159
13.2.2.7	Time out.....	159
13.2.2.8	Last component	159

	13.2.2.9	Problem code	159
	13.2.2.9.1	Mapping to MAP User Error	159
	13.2.2.9.2	Mapping to MAP Provider Error parameter	159
	13.2.2.9.3	Mapping to diagnostic parameter	160
13.3	SDL descriptions		160
14	Abstract syntax of the MAP protocol		186
14.1	General		186
	14.1.1	Encoding rules	186
	14.1.2	Use of TC	186
	14.1.2.1	Use of Global Operation and Error codes defined outside MAP	187
	14.1.3	Use of information elements defined outside MAP	187
	14.1.4	Compatibility considerations	188
	14.1.5	Structure of the Abstract Syntax of MAP	188
	14.1.6	Application Contexts	189
14.2	Operation packages		191
	14.2.1	General aspects	191
	14.2.2	Packages specifications	192
	14.2.2.1	Location updating	192
	14.2.2.2	Location cancellation	192
	14.2.2.3	Roaming number enquiry	192
	14.2.2.4	Information retrieval	192
	14.2.2.5	Inter-VLR information retrieval	192
	14.2.2.6	IMSI retrieval	193
	14.2.2.7	Call control transfer	193
	14.2.2.8 - 14.2.2.9	[Spare]	193
	14.2.2.10	Interrogation	193
	14.2.2.11	[Spare]	193
	14.2.2.12	Handover Control	193
	14.2.2.13	Subscriber Data management stand alone	194
	14.2.2.14	Equipment management	194
	14.2.2.15	Subscriber data management	194
	14.2.2.16	Location register restart	194
	14.2.2.17	Tracing stand-alone	195
	14.2.2.18	Functional SS handling	195
	14.2.2.19	Tracing	195
	14.2.2.20	Binding	195
	14.2.2.21	Unstructured SS handling	195
	14.2.2.22	MO Short message relay services	196
	14.2.2.23	Short message gateway services	196
	14.2.2.24	MT Short message relay services	196
	14.2.2.25	[Spare]	197
	14.2.2.26	Message waiting data management	197
	14.2.2.27	Alerting	197
	14.2.2.28	Data restoration	197
	14.2.2.29	Purging	197
	14.2.2.30	Subscriber information enquiry	197
	14.2.2.31	Any time information enquiry	198
14.3	Application contexts		199
	14.3.1	General aspects	199
	14.3.2	Application context definitions	200
	14.3.2.1	[Spare]	200
	14.3.2.2	Location Updating	200
	14.3.2.3	Location Cancellation	200
	14.3.2.4	Roaming number enquiry	200
	14.3.2.5	[Spare]	201
	14.3.2.6	Location Information Retrieval	201
	14.3.2.7	Call control transfer	201
	14.3.2.8 - 14.3.2.10	[Spare]	201
	14.3.2.11	Location registers restart	201
	14.3.2.12	Handover control	201

	14.3.2.13	IMSI Retrieval	202
	14.3.2.14	Equipment Management.....	202
	14.3.2.15	Information retrieval.....	202
	14.3.2.16	Inter-VLR information retrieval.....	202
	14.3.2.17	Stand Alone Subscriber Data Management	202
	14.3.2.18	Tracing.....	203
	14.3.2.19	Network functional SS handling.....	203
	14.3.2.20	Network unstructured SS handling	203
	14.3.2.21	Short Message Gateway.....	203
	14.3.2.22	Mobile originating Short Message Relay	204
	14.3.2.23	[Spare]	204
	14.3.2.24	Short message alert.....	204
	14.3.2.25	Short message waiting data management	204
	14.3.2.26	Mobile terminating Short Message Relay	204
	14.3.2.27	MS purging	205
	14.3.2.28	Subscriber information enquiry	205
	14.3.2.29	Any time information enquiry	205
	14.3.3	ASN.1 Module for application-context-names	206
14.4		MAP Dialogue Information	207
14.5		MAP operation and error codes	210
14.6		MAP operation and error types	215
	14.6.1	Mobile Service Operations	215
	14.6.2	Operation and Maintenance Operations	219
	14.6.3	Call Handling Operations.....	221
	14.6.4	Supplementary service operations	223
	14.6.5	Short message service operations	227
	14.6.6	Errors.....	230
14.7		MAP constants and data types	235
	14.7.1	Mobile Service data types	235
	14.7.2	Operation and maintenance data types.....	245
	14.7.3	Call handling data types	246
	14.7.4	Supplementary service data types	249
	14.7.5	Supplementary service codes	252
	14.7.6	Short message data types.....	254
	14.7.7	Error data types	256
	14.7.8	Common data types	260
	14.7.9	Teleservice Codes.....	264
	14.7.10	Bearer Service Codes	266
	14.7.11	Extension data types	268
15		General on MAP user procedures.....	269
15.1		Introduction	269
15.2		Common aspects of user procedure descriptions	269
	15.2.1	General conventions.....	269
	15.2.2	Naming conventions.....	269
	15.2.3	Convention on primitives parameters.....	271
		15.2.3.1 Open service.....	271
		15.2.3.2 Close service	271
	15.2.4	Version handling at dialogue establishment.....	271
		15.2.4.1 Behaviour at the initiating side	271
		15.2.4.2 Behaviour at the responding side	271
	15.2.5	Abort Handling.....	272
	15.2.6	SDL conventions	272
15.3		Interaction between MAP Provider and MAP Users	272
16		Mobility procedures	273
16.1		Location management Procedures.....	273
	16.1.1	Location updating	278
		16.1.1.1 General	278
		16.1.1.2 Detailed procedure in the MSC.....	282
		16.1.1.3 Detailed procedure in the VLR.....	286
		16.1.1.4 Detailed procedure in the HLR	298
		16.1.1.5 Send Identification	301

	16.1.1.5.1	General.....	301
	16.1.1.5.2	Detailed procedure in the VLR	301
	16.1.1.5.3	Detailed procedure in the PVLR.....	301
	16.1.1.6	The Process Update Location VLR.....	303
	16.1.1.7	The Process Subscriber Present HLR	305
16.1.2		Location Cancellation.....	307
	16.1.2.1	General	307
	16.1.2.2	Detailed procedure in the HLR	307
	16.1.2.3	Detailed procedure in the VLR	308
16.1.3		Detach IMSI	311
	16.1.3.1	General	311
	16.1.3.2	Detailed procedure in the MSC	311
	16.1.3.3	Detailed procedure in the VLR	311
16.1.4		Purge MS	314
	16.1.4.1	General	314
	16.1.4.2	Detailed procedure in the VLR	314
	16.1.4.3	Detailed procedure in the HLR	314
16.2		Handover procedure.....	318
	16.2.1	General	318
	16.2.2	Handover procedure in MSC-A.....	321
	16.2.2.1	Basic handover	321
	16.2.2.2	Handling of access signalling.....	321
	16.2.2.3	Other procedures in stable handover situation	322
	16.2.2.4	Subsequent handover	322
	16.2.2.5	SDL Diagrams.....	322
	16.2.3	Handover procedure in MSC-B.....	335
	16.2.3.1	Basic handover	335
	16.2.3.2	Allocation of handover number	335
	16.2.3.3	Handling of access signalling.....	335
	16.2.3.4	Other procedures in stable handover situation	335
	16.2.3.5	Subsequent handover	335
	16.2.3.6	SDL Diagrams.....	336
	16.2.4	Handover error handling macro	348
	16.2.5	Handover procedure in VLR	350
	16.2.5.1	Allocation of handover number	350
	16.2.5.2	SDL Diagrams.....	350
16.3		Fault recovery procedures.....	353
	16.3.1	VLR fault recovery procedures	353
	16.3.2	HLR fault recovery procedures	355
	16.3.3	VLR restoration: the restore data procedure in the HLR	361
16.4		Macro Insert_Subst_Data_Framed_HLR.....	363
17		Operation and maintenance procedures	365
	17.1	General.....	365
	17.1.1	Tracing Co-ordinator for the VLR.....	365
	17.1.2	Subscriber Data Management Co-ordinator for the VLR.....	367
	17.2	Tracing procedures	369
	17.2.1	Procedures in the HLR	370
	17.2.1.1	Subscriber tracing activation procedure.....	370
	17.2.1.2	Subscriber tracing deactivation procedure.....	373
	17.2.2	Procedures in the VLR.....	376
	17.2.2.1	Subscriber tracing activation procedure.....	376
	17.2.2.2	Subscriber tracing deactivation procedure.....	378
	17.2.2.3	Subscriber tracing procedure	380
	17.2.3	Procedures in the MSC.....	380
	17.2.3.1	Subscriber tracing procedure	380
	17.3	Subscriber data management procedures	380
	17.3.1	Procedures in the HLR	381
	17.3.1.1	Subscriber deletion procedure	381
	17.3.1.2	Subscriber data modification procedure	383
	17.3.2	Procedures in the VLR.....	386
	17.3.2.1	Subscriber deletion procedure	386
	17.3.2.2	Subscriber data modification procedure	386

17.4	Subscriber Identity procedure	389
17.4.1	Subscriber identity procedure in the HLR	389
17.4.2	Subscriber identity procedure in the VLR	391
18	Call handling procedures	393
18.1	General	393
18.2	Retrieval of routing information	393
18.2.1	General	393
18.2.2	Process in the GMSC	395
18.2.3	Procedures in the HLR	399
18.2.4	Process in the VLR to provide a roaming number	405
18.2.5	Process in the VLR to restore subscriber data	407
18.2.6	Process in the VLR to provide subscriber information	409
18.2.7	Process in the HLR for Any Time Interrogation	411
18.2.7.1	Process in the gsmSCF	411
18.2.8	Process in the HLR	411
18.3	Transfer of call handling	415
18.3.1	General	415
18.3.2	Process in the VMSC	415
18.3.3	Process in the GMSC	418
19	Supplementary services procedures	420
19.0	Functional supplementary service processes	420
19.0.1	Functional supplementary service process co-ordinator for MSC	420
19.0.2	Functional supplementary service process co-ordinator for VLR	422
19.0.3	Functional supplementary service process co-ordinator for HLR	425
19.1	Registration procedure	428
19.1.1	General	428
19.1.2	Procedures in the MSC	429
19.1.3	Procedures in the VLR	431
19.1.4	Procedures in the HLR	434
19.2	Erasure procedure	437
19.2.1	General	437
19.2.2	Procedures in the MSC	437
19.2.3	Procedures in the VLR	437
19.2.4	Procedures in the HLR	438
19.3	Activation procedure	438
19.3.1	General	438
19.3.2	Procedures in the MSC	439
19.3.3	Procedures in the VLR	441
19.3.4	Procedures in the HLR	444
19.4	Deactivation procedure	447
19.4.1	General	447
19.4.2	Procedures in the MSC	447
19.4.3	Procedures in the VLR	448
19.4.4	Procedures in the HLR	448
19.5	Interrogation procedure	448
19.5.1	General	448
19.5.2	Procedures in the MSC	449
19.5.3	Procedures in the VLR	449
19.5.4	Procedures in the HLR	453
19.6	Invocation procedure	455
19.6.1	General	455
19.6.2	Procedures in the MSC	456
19.6.3	Procedures in the VLR	459
19.7	Password registration procedure	461
19.7.1	General	461
19.7.2	Procedures in the MSC	462
19.7.3	Procedures in the VLR	462
19.7.4	Procedures in the HLR	462
19.8	Mobile Initiated USSD procedure	465
19.8.1	General	465
19.8.2	Procedures in the MSC	465

	19.8.3	Procedures in the VLR.....	469
	19.8.4	Procedures in the HLR	474
19.9		Network initiated USSD procedure.....	477
	19.9.1	General	477
	19.9.2	Procedure in the MSC.....	477
	19.9.3	Procedure in the VLR.....	482
	19.9.4	Procedure in the HLR	489
19.10		Common macros for clause 19	492
	19.10.1	SS Password handling macros.....	492
	19.10.2	SS Error handling macros.....	495
20		Short message service procedures	499
20.1		General.....	499
	20.1.1	Mobile originated short message service Co-ordinator for the MSC	499
	20.1.2	Short message Gateway Co-ordinator for the HLR	501
20.2		The mobile originated short message transfer procedure	503
	20.2.1	Procedure in the servicing MSC	504
	20.2.2	Procedure in the VLR.....	508
	20.2.3	Procedure in the interworking MSC	510
20.3		The mobile terminated short message transfer procedure	512
	20.3.1	Procedure in the Servicing MSC.....	514
	20.3.2	Procedures in the VLR.....	523
	20.3.3	Procedures in the HLR	527
	20.3.4	Procedures in the gateway MSC	531
20.4		The Short Message Alert procedure	540
	20.4.1	Procedures in the Servicing MSC	542
	20.4.2	Procedures in the VLR.....	544
		20.4.2.1 The Mobile Subscriber is present.....	544
		20.4.2.2 The Mobile Equipment has memory available	544
	20.4.3	Procedures in the HLR	546
	20.4.4	Procedures in the Interworking MSC	548
20.5		The SM delivery status report procedure	550
	20.5.1	Procedures in the HLR	550
	20.5.2	Procedures in the gateway MSC	552
20.6		Common procedures for the short message clause	554
	20.6.1	The macro Report_SM_Delivery_Stat_HLR	554
21		General macro description.....	556
21.1		MAP open macros	556
	21.1.1	Macro Receive_Open_Ind	556
	21.1.2	Macro Receive_Open_Cnf	557
21.2		Macros to check the content of indication and confirmation primitives	560
	21.2.1	Macro Check_Indication	560
	21.2.2	Macro Check_Confirmation	560
21.3		The page and search macros	563
	21.3.1	Macro PAGE_MSC	563
	21.3.2	Macro Search_For_MS_MSC.....	564
21.4		Macros for handling an Access Request.....	568
	21.4.1	Macro Process_Access_Request_MSC.....	568
	21.4.2	Macro Process_Access_Request_VLR.....	574
	21.4.3	Macro Identification Procedure	576
21.5		Authentication macros and processes	581
	21.5.1	Macro Authenticate_MSC	581
	21.5.2	Macro Authenticate_VLR	581
	21.5.3	Process Obtain_Authentication_Sets_VLR	581
	21.5.4	Macro Obtain_Authent_Para_VLR	582
	21.5.5	Process Obtain_Auth_Sets_HLR	583
21.6		IMEI Handling Macros	590
	21.6.1	Macro Check_IMEI_MSC	590
	21.6.2	Macro Check_IMEI_VLR	590
	21.6.3	Process Check_IMEI_EIR	591
	21.6.4	Macro Obtain_IMEI_MSC	591
	21.6.5	Macro Obtain_IMEI_VLR	592

21.7	Insert Subscriber Data Macros	599
21.7.1	Macro Insert_SubData_VLR.....	599
21.7.2	Process Insert_SubData_Stand_Alone_HLR.....	601
21.7.3	Macro Wait_for_Insert_SubData_Cnf	604
21.7.4	Process Send_Insert_SubData	606
21.8	Request IMSI Macros	607
21.8.1	Macro Obtain_IMSI_MSC.....	607
21.8.2	Macro Obtain_IMSI_VLR.....	608
21.9	Tracing macros	609
21.9.1	Macro Trace_Subscriber_Activity_MSC.....	609
21.9.2	Macro Trace_Subscriber_Activity_VLR.....	610
21.9.3	Macro Activate_Tracing_VLR.....	611
21.9.4	Macro Control_Tracing_HLR.....	612
21.10	Short Message Alert procedures	614
21.10.1	Subscriber_Present_VLR process	614
21.10.2	Macro Alert_Service_Centre_HLR	615
Annex A (informative):	Cross-reference for abstract syntaxes of MAP	617
Annex B (informative):	Fully expanded ASN.1 sources for abstract syntaxes of MAP	671
B.1	Fully Expanded ASN.1 Source of MAP-Protocol/TCAPMessages.....	671
B.2	Fully Expanded ASN.1 Source of MAP-DialogueInformation.....	706
Annex C (informative):	Formal protocol incompatibilities between versions 1 & 2 of MAP	710
C.1	Introduction.....	710
C.2	Deletion of operations and errors	710
C.2.1	Deletion of operation DeregisterMobileSubscriber	710
C.2.2	Deletion of operation RegisterChargingInfo.....	710
C.2.3	Deletion of operation ForwardSS-Notification.....	710
C.2.4	Deletion of operations used only on the B-interface	710
C.2.5	Deletion of error InsufficientBearerCapabilities.....	710
C.3	Deletion of errors for operations.....	710
C.3.1	Error NegativePW-Check for operation RegisterSS	710
C.3.2	Error NegativePW-Check for operation EraseSS	711
C.3.3	Error NegativePW-Check for operation InterrogateSS	711
C.3.4	Error CUG-Reject for operation SendRoutingInfoForSM.....	711
C.4	Changes to definitions of data types	711
C.4.1	CUG-Feature	711
C.4.2	CUG-FeatureList.....	711
C.4.3	CUG-Info.....	711
C.4.4	CUG-RejectCause	711
C.4.5	IMSI.....	711
C.4.6	ISDN-AddressString.....	711
C.4.7	Password	712
C.4.8	RequestParameter	712
C.4.9	RequestParameterList	712
C.4.10	SentParameter.....	712
C.4.11	SentParameterList	712
C.4.12	SS-Data	712
C.4.13	SS-Info	712
C.4.14	SS-InfoList	712
C.4.15	SS-SubscriptionOption.....	712
C.4.16	SubscriberData	713
C.5	Changes to parameters of errors	713
C.5.1	CUG-Reject.....	713
C.5.2	SS-SubscriptionViolation.....	713

C.6	Changes to parameters of operations	713
C.6.1	InsertSubscriberData.....	713
C.6.2	RegisterSS	713
C.6.3	SendParameters	714
C.6.4	SendRoutingInfoForSM.....	714
C.7	Changes to results of operations	714
C.7.1	ActivateSS	714
C.7.2	DeactivateSS.....	714
C.7.3	EraseSS	714
C.7.4	GetPassword	715
C.7.5	InterrogateSS	715
C.7.6	RegisterSS	715
C.7.7	SendParameters	715
C.7.8	SendRoutingInfoForSM.....	715
C.8	Changes to errors of operations	716
C.8.1	ActivateSS	716
C.8.2	DeactivateSS.....	716
C.8.3	EraseSS	716
C.8.4	RegisterSS	716
C.8.5	SendRoutingInfo.....	716
Annex D (informative):	Change Request History	717
History		719

Intellectual Property Rights

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

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

Foreword

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

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

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

- 5 Indicates GSM Phase 2+ Release 1996
- x the second digit is incremented for all changes of substance, i.e. technical enhancements, corrections, updates, etc.
- y the third digit is incremented when editorial only changes have been incorporated in the specification.

Proposed transposition dates	
Date of adoption of this EN:	31 March 2000
Date of latest announcement of this ETS (doa):	30 June 2000
Date of latest publication of new National Standard or endorsement of this ETS (dop/e):	31 December 2000
Date of withdrawal of any conflicting National Standard (dow):	31 December 2000

Blank page

1 Scope

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

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

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

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

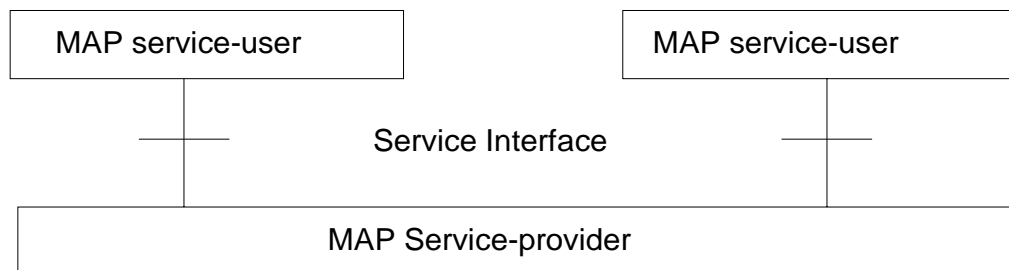


Figure 1.1/1: Modelling principles

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

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

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

1.1 Normative references

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

- | | |
|-----|--|
| [1] | GSM 01.04: "Digital cellular telecommunications system (Phase 2+); Abbreviations and acronyms". |
| [2] | GSM 02.01: "Digital cellular telecommunications system (Phase 2+); Principles of telecommunication services supported by a GSM Public Land Mobile Network (PLMN)". |
| [3] | GSM 02.02: "Digital cellular telecommunications system (Phase 2+); Bearer Services (BS) Supported by a GSM Public Land Mobile Network (PLMN)". |
| [4] | GSM 02.03: "Digital cellular telecommunications system (Phase 2+); Teleservices Supported by a GSM Public Land Mobile Network (PLMN)". |
| [5] | GSM 02.04: "Digital cellular telecommunications system (Phase 2+); General on supplementary services". |
| [6] | GSM 02.09: "Digital cellular telecommunications system; Security aspects". |

- [7] GSM 02.16: "Digital cellular telecommunications system; International Mobile station Equipment Identities (IMEI)".
- [8] GSM 02.41: "Digital cellular telecommunications system (Phase 2+); Operator determined barring".
- [9] GSM 02.81: "Digital cellular telecommunications system; Line identification supplementary services - Stage 1".
- [10] GSM 02.82: "Digital cellular telecommunications system; Call Forwarding (CF) supplementary services - Stage 1".
- [11] GSM 02.83 : "Digital cellular telecommunications system; Call Waiting (CW) and Call Hold (HOLD) supplementary services - Stage 1".
- [12] GSM 02.84: "Digital cellular telecommunications system; Multi Party (MPTY) supplementary services - Stage 1".
- [13] GSM 02.85: "Digital cellular telecommunications system; Closed User Group (CUG) supplementary services - Stage 1".
- [14] GSM 02.86: "Digital cellular telecommunications system; Advice of charge (AoC) supplementary services - Stage 1".
- [15] GSM 02.88: "Digital cellular telecommunications system; Call Barring (CB) supplementary services - Stage 1".
- [16] GSM 02.90: "Digital cellular telecommunication system (Phase 2+); Unstructured supplementary services operation - Stage 1".
- [17] GSM 03.03 (ETS 300 927): "Digital cellular telecommunications system (Phase 2+); Numbering, addressing and identification".
- [18] GSM 03.04: "Digital cellular telecommunications system; Signalling requirements relating to routing of calls to mobile subscribers".
- [19] GSM 03.07: "Digital cellular telecommunications system (Phase 2+); Restoration procedures".
- [20] GSM 03.08: "Digital cellular telecommunications system (Phase 2+); Organisation of subscriber data".
- [21] GSM 03.09: "Digital cellular telecommunications system (Phase 2+; Handover procedures".
- [22] GSM 03.11: "Digital cellular telecommunications system; Technical realization of supplementary services".
- [23] GSM 03.12: "Digital cellular telecommunications system; Location registration procedures".
- [24] GSM 03.20: "Digital cellular telecommunications system; Security related network functions".
- [25] GSM 03.38: "Digital cellular telecommunications system (Phase 2+); Alphabets and language specific information for GSM".

- [26] GSM 03.40: "Digital cellular telecommunications system (Phase 2+); Technical realization of the Short Message Service (SMS) Point to Point (PP)".
- [27] GSM 03.81: "Digital cellular telecommunications system; Line identification supplementary services - Stage 2".
- [28] GSM 03.82: "Digital cellular telecommunications system; Call Forwarding (CF) supplementary services - Stage 2".
- [29] GSM 03.83: "Digital cellular telecommunications system; Call Waiting (CW) and Call Hold (HOLD) supplementary services - Stage 2".
- [30] GSM 03.84: "Digital cellular telecommunications system; Multi Party (MPTY) supplementary services - Stage 2".
- [31] GSM 03.85: "Digital cellular telecommunications system; Closed User Group (CUG) supplementary services - Stage 2".
- [32] GSM 03.86: "Digital cellular telecommunications system; Advice of Charge (AoC) supplementary services - Stage 2".
- [33] GSM 03.88: "Digital cellular telecommunications system; Call Barring (CB) supplementary services - Stage 2".
- [34] GSM 03.90: "Digital cellular telecommunications system; Unstructured supplementary services operation - Stage 2".
- [35] GSM 04.08: "Digital cellular telecommunications system (Phase 2+); Mobile radio interface layer 3 specification".
- [36] GSM 04.10: "Digital cellular telecommunications system; Mobile radio interface layer 3 Supplementary services specification General aspects".
- [37] GSM 04.11: "Digital cellular telecommunications system (Phase 2+); Point-to-Point (PP) Short Message Service (SMS) support on mobile radio interface".
- [38] GSM 04.80: "Digital cellular telecommunications system (Phase 2+); Mobile radio interface layer 3 supplementary services specification Formats and coding".
- [39] GSM 04.81: "Digital cellular telecommunications system; Line identification supplementary services - Stage 3".
- [40] GSM 04.82: "Digital cellular telecommunications system ; Call Forwarding (CF) supplementary services - Stage 3".
- [41] GSM 04.83: "Digital cellular telecommunications system; Call Waiting (CW) and Call Hold (HOLD) supplementary services - Stage 3".
- [42] GSM 04.84: "Digital cellular telecommunications system; Multi Party (MPTY) supplementary services - Stage 3".
- [43] GSM 04.85: "Digital cellular telecommunications system; Closed User Group (CUG) supplementary services - Stage 3".

- [44] GSM 04.86: "Digital cellular telecommunications system; Advice of Charge (AoC) supplementary services - Stage 3".
- [45] GSM 04.88: "Digital cellular telecommunications system; Call Barring (CB) supplementary services - Stage 3".
- [46] GSM 04.90: "Digital cellular telecommunications system; Unstructured supplementary services operation - Stage 3".
- [47] GSM 08.02: "Digital cellular telecommunications system; Base Station System - Mobile-services Switching Centre (BSS - MSC) interface Interface principles".
- [48] GSM 08.06: "Digital cellular telecommunications system; Signalling transport mechanism specification for the Base Station System - Mobile-services Switching Centre (BSS - MSC) interface".
- [49] GSM 08.08: "Digital cellular telecommunications system (Phase 2+); Mobile Switching Centre - Base Station System (MSC - BSS) interface Layer 3 specification".
- [49a] GSM 08.08: "Digital cellular telecommunications system (Phase 1); Mobile Switching Centre - Base Station System (MSC - BSS) interface Layer 3 specification".
- [50] GSM 09.01: "Digital cellular telecommunications system; General network interworking scenarios".
- [51] GSM 09.02: "Digital cellular telecommunications system (Phase 1); Mobile Application Part (MAP) specification".
- [52] GSM 09.03: "Digital cellular telecommunications system; Signalling requirements on interworking between the Integrated Services Digital Network (ISDN) or Public Switched Telephone Network (PSTN) and the Public Land Mobile Network (PLMN)".
- [53] GSM 09.04: "Digital cellular telecommunications system; Interworking between the Public Land Mobile Network (PLMN) and the Circuit Switched Public Data Network (CSPDN)".
- [54] GSM 09.05: "Digital cellular telecommunications system; Interworking between the Public Land Mobile Network (PLMN) and the Packet Switched Public Data Network (PSPDN) for Packet Assembly/Disassembly facility (PAD) access".
- [55] GSM 09.06: "Digital cellular telecommunications system (Phase 2+); Interworking between a Public Land Mobile Network (PLMN) and a Packet Switched Public Data Network/Integrated Services Digital Network (PSPDN/ISDN) for the support of packet switched data transmission services".
- [56] GSM 09.07: "Digital cellular telecommunications system (Phase 2+); General requirements on interworking between the Public Land Mobile Network (PLMN) and the Integrated Services Digital Network (ISDN) or Public Switched Telephone Network (PSTN)".
- [57] GSM 09.08: "Digital cellular telecommunications system; Application of the Base Station System Application Part (BSSAP) on the E-interface".

- [58] GSM 09.10: "Digital cellular telecommunications system (Phase 2+); Information element mapping between Mobile Station - Base Station System and BSS - Mobile-services Switching Centre (MS - BSS - MSC) Signalling procedures and the Mobile Application Part (MAP)".
- [59] GSM 09.11: "Digital cellular telecommunications system (Phase 2+); Signalling interworking for supplementary services".
- [60] GSM 09.90: "Digital cellular telecommunications system (Phase 2+); Interworking between Phase 1 infrastructure and Phase 2 Mobile Stations (MS)".
- [61] GSM 12.08: "Digital cellular telecommunications system (Phase 2); Subscriber and Equipment Trace".
- [62] ETS 300 102-1 (1990): "Integrated Services Digital Network (ISDN); User-network interface layer 3 specifications for basic call control".
- [63] ETS 300 136 (1992): "Integrated Services Digital Network (ISDN); Closed User Group (CUG) supplementary service description".
- [64] ETS 300 138 (1992): "Integrated Services Digital Network (ISDN); Closed User Group (CUG) supplementary service Digital Subscriber Signalling System No.one (DSS1) protocol".
- [65] ETS 300 287: "Integrated Services Digital Network (ISDN); CCITT Signalling System No.7 Transaction Capabilities Application Part (TCAP)(version 2)".
- [66] ETR 060: "Signalling Protocols and Switching (SPS); Guide-lines for using Abstract Syntax Notation One (ASN.1) in telecommunication application protocols".
- [67] CCITT Recommendation E.164: "Numbering plan for the ISDN era".
- [68] CCITT Recommendation E.212: "Identification plan for land mobile stations".
- [69] CCITT Recommendation E.213: "Telephone and ISDN numbering plan for land mobile stations".
- [70] CCITT Recommendation E.214: "Structuring of the land mobile global title for the signalling connection control part".
- [71] CCITT Recommendation Q.669: "Interworking between the Digital Subscriber Signalling System Layer 3 protocol and the Signalling System No.7 ISDN User part".
- [72] CCITT Recommendation Q.711: "Specifications of Signalling System No.7; Functional description of the signalling connection control part".
- [73] CCITT Recommendation Q.712: "Definition and function of SCCP messages".
- [74] CCITT Recommendation Q.713: "Specifications of Signalling System No.7; SCCP formats and codes".
- [75] CCITT Recommendation Q.714: "Specifications of Signalling System No.7; Signalling connection control part procedures".

- [76] CCITT Recommendation Q.716: "Specifications of Signalling System No.7; Signalling connection control part (SCCP) performances".
- [77] CCITT Recommendation Q.721 (1988): "Specifications of Signalling System No.7; Functional description of the Signalling System No.7 Telephone user part".
- [78] CCITT Recommendation Q.722 (1988): "Specifications of Signalling System No.7; General function of Telephone messages and signals".
- [79] CCITT Recommendation Q.723 (1988): "Specifications of Signalling System No.7; Formats and codes".
- [80] CCITT Recommendation Q.724 (1988): "Specifications of Signalling System No.7; Signalling procedures".
- [81] CCITT Recommendation Q.725 (1988): "Specifications of Signalling System No.7; Signalling performance in the telephone application".
- [82] CCITT Recommendation Q.761 (1988): "Specifications of Signalling System No.7; Functional description of the ISDN user part of Signalling System No.7".
- [83] CCITT Recommendation Q.762 (1988): "Specifications of Signalling System No.7; General function of messages and signals".
- [84] CCITT Recommendation Q.763 (1988): "Specifications of Signalling System No.7; Formats and codes".
- [85] CCITT Recommendation Q.764 (1988): "Specifications of Signalling System No.7; Signalling procedures".
- [86] CCITT Recommendation Q.767: "Specifications of Signalling System No.7; Application of the ISDN user part of CCITT signalling System No.7 for international ISDN interconnections".
- [87] CCITT Recommendation Q.771: "Specifications of Signalling System No.7; Functional description of transaction capabilities".
- [88] CCITT Recommendation Q.772: "Specifications of Signalling System No.7; Transaction capabilities information element definitions".
- [89] CCITT Recommendation Q.773: "Specifications of Signalling System No.7; Transaction capabilities formats and encoding".
- [90] CCITT Recommendation Q.774: "Specifications of Signalling System No.7; Transaction capabilities procedures".
- [91] CCITT Recommendation Q.775: "Specifications of Signalling System No.7; Guide-lines for using transaction capabilities".
- [92] CCITT Recommendation X.200: "Reference Model of Open systems interconnection for CCITT Applications".
- [93] CCITT Recommendation X.208 (1988): "Specification of Abstract Syntax Notation One (ASN.1)".

- [94] CCITT Recommendation X.209 (1988): "Specification of basic encoding rules for Abstract Syntax Notation One (ASN.1)".
- [95] CCITT Recommendation X.210: "Open systems interconnection layer service definition conventions".
- [96] GSM 09.02: "Digital cellular telecommunications system (Phase 2); Mobile Application Part (MAP) specification".
- [97] GSM 03.18: "Digital cellular telecommunications system (Phase 2+); Basic Call Handling".
- [98] GSM 03.78: "Digital cellular telecommunications system (Phase 2+); Customised Applications for Mobile network Enhanced Logic (CAMEL) - Stage 2".
- [99] GSM 03.79: "Digital cellular telecommunications system (Phase 2+); Support of Optimal Routeing (SOR) - Stage 2".

1.2 Abbreviations

Abbreviations used in this ETS are listed in GSM 01.04.

2 Configuration of the mobile network

2.1 The entities of the mobile system

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

2.1.1 The Home Location Register (HLR)

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

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

- IMSI;
- MSISDN.

The data base contains other information such as:

- location information (VLR number);
- basic telecommunication services subscription information;
- service restrictions (e.g. roaming limitation);
- supplementary services; the tables contain the parameters attached to these services.

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

2.1.2 The Visitor Location Register (VLR)

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

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

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

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

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

2.1.3 The Mobile-services Switching Centre (MSC)

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

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

2.1.4 The Base Station System (BSS)

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

2.1.5 The Gateway MSC (GMSC)

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

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

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

See also GSM 03.04.

2.1.6 The SMS Gateway MSC

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

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

2.1.7 The SMS Interworking MSC

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

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

2.1.8 The VBS/VGCS Anchor MSC

The voice broadcast/group call anchor MSC obtains from the associated GCR all relevant attributes and controls in turn all cells and dispatchers belonging to a given group call.

NOTE: Interactions between MSCs are not scope of this release of the specification; i.e. the "relay MSC function" which is needed to implement inter-MSC group calls will be specified in the next release of this standard

2.1.9 The Equipment Identity Register (EIR)

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

2.1.10 The GSM Service Control Function (gsmSCF)

This functional entity contains the CAMEL service logic to implement OSS. It interfaces with the gsmSSF and the HLR; see also TS GSM 03.78.

2.1.11 The Group Call Register (GCR)

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

2.2 Configuration of a Public Land Mobile Network (PLMN)

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

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

2.3 Interconnection between PLMNs

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

2.4 The interfaces within the mobile service

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

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

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

2.4.2 Interface between the HLR and the gsmSCF (J-interface)

This interface is used by the gsmSCF to request information from the HLR. Support of the gsmSCF-HLR interface is a network operator option. As a network operator option, the HLR may refuse to provide the information requested by the gsmSCF.

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

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

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

2.4.4 Interface between VLRs (G-interface)

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

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

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

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

2.4.6 Interface between MSCs (E-interface)

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

This interface is also used to forward short messages.

The application of this interface for inter-MSC VBS/VGCS calls will be specified in the next release of this standard.

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

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

The BSS-MSC interface carries information concerning:

- BSS management;
- call handling;
- location management.

2.4.8 Interface between MSC and EIR (F-interface)

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

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

This is an internal interface.

2.5 Splitting of the data storage

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

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

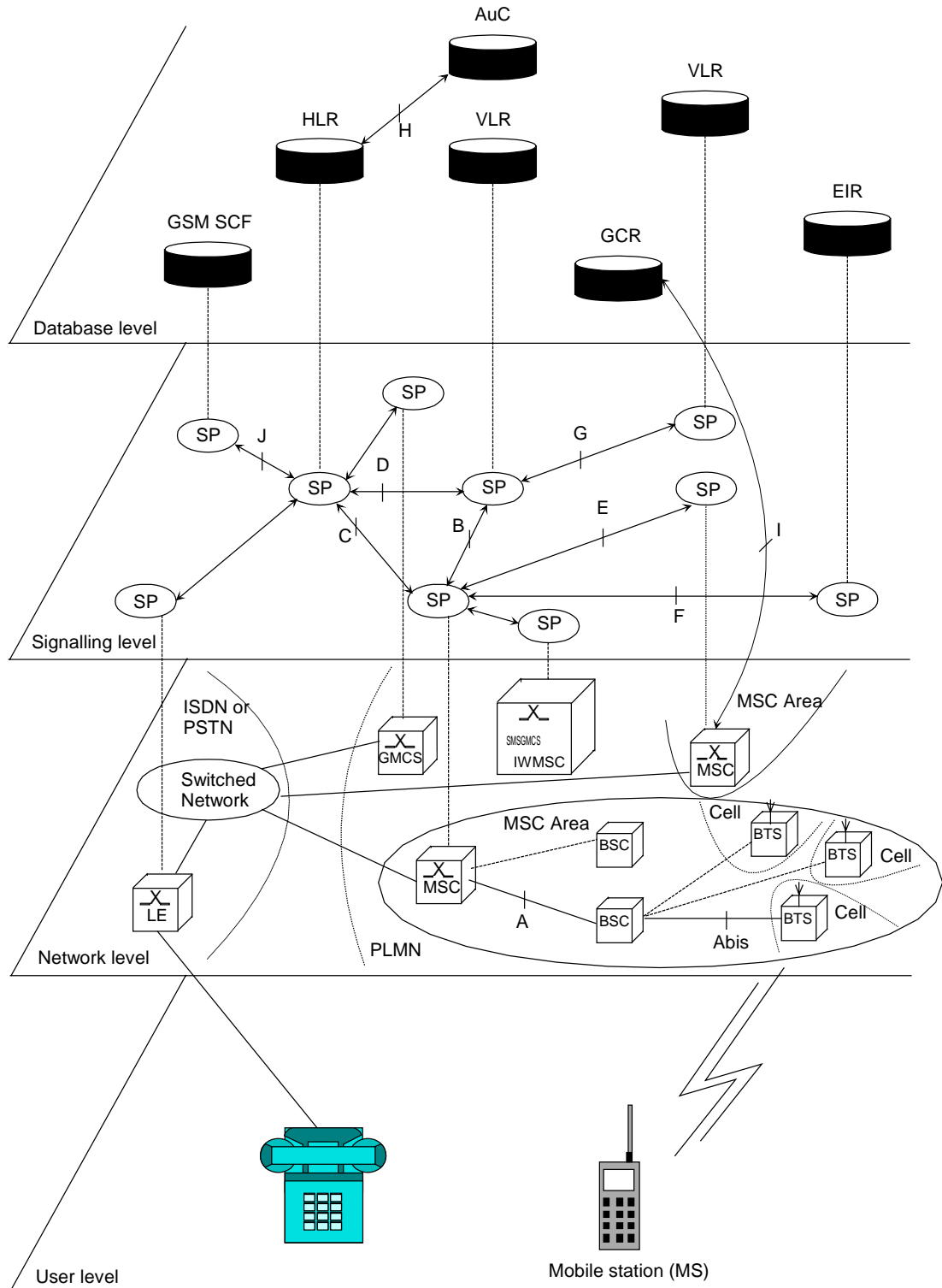


Figure 2.2/1: Configuration of a PLMN

3 Overload and compatibility overview

3.1 Overload control

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

3.1.1 Overload control for MSC (outside MAP)

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

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

3.1.2 Overload control for MAP entities

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

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

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

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

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

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

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

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

Responder = MSC/VLR		Initiating	Entity
Priority			high
	<u>Handover</u>		
	handoverControl (prepareHandover/v2), (performHandover/v1)	MSC	
	<u>Mobility and Location Register Management</u>		
	locationCancel (cancelLocation)	HLR	
	reset (reset)	HLR	
	interVlrInfoRetrieval (sendIdentification/v2), (sendParameters/v1)	VLR	
	subscriberDataMngt (insertSubscriberData), (deleteSubscriberData)	HLR	
	tracing (activateTraceMode), (deactivateTraceMode)	HLR	
	<u>Short Message Service</u>		
	shortMsgMO-Relay (MO-ForwardSM v3) (forwardSM v1/v2)	MSC	
	shortMsgMT-Relay (MT-ForwardSM v3) (forwardSM	MSC	
	shortMsgAlert (alertServiceCentre/v2), (alertServiceCentreWithoutResult/v1)	HLR	v1/v2)
	<u>Mobile Terminating Traffic</u>		
	roamingNbEnquiry (provideRoamingNumber)	HLR	
	callControlTransfer (resumeCallHandling)	MSC	
	subscriberInfoEnquiry (provideSubscriberInformation)	HLR	
	<u>Network-Initiated</u>		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.

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

3.1.3 Congestion control for Signalling System No. 7

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

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

3.2 Compatibility

3.2.1 General

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

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

When starting a signalling procedure, the MAP-user supplies an application-context-name to the MAP-provider. This name refers to the set of application layer communication capabilities required for this dialogue. This refers to the required TC facilities (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 functionalities (if possible).

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

3.2.2 Strategy for selecting the Application Context (AC) version

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

3.2.2.1 Proposed method

A table (table 1) may be set up by administrative action to define the highest application context (AC) version supported by each destination; a destination may be another node within the same or a different PLMN, or another PLMN considered as a single entity. The destination may be defined by an E.164 number or an E.214 number derived from an IMSI. The table also includes the date when each destination is expected to be able to handle at least one AC of the 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 which has an entry in table 1. For a given entity, the entry in table 2 may be a single application context version or a vector of different versions applying to different application contexts for that entity. Table 2 is managed as described in subclause 3.2.2.2.

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

- a) the version shown in table 1 is "version 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 subclause 3.2.2.2;
- c) when the PLMN operator declares that an entity (single node or entire PLMN) has been upgraded to support all the MAP version 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.

3.2.2.2 Managing the version look-up table

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

IF the entity number is known

THEN

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

ELSE

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

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

IF the destination address is known and not timed-out

THEN

It retrieves the appropriate AC-name and uses it

IF the dialogue is accepted by the peer

THEN

It does not modify the version control table

ELSE (this should never occur)

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

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

ELSE

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

IF the dialogue is accepted by the peer

THEN

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

ELSE

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

IF the destination node was not known

THEN

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

ELSE

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

3.2.2.3 Optimizing the method

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

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

for HLR:

- Subscriber data modification (stand alone);

for VLR:

- Data Restoration.

4 Requirements concerning the use of SCCP and TC

4.1 Use of SCCP

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

4.1.1 SCCP Class

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

4.1.2 Sub-System Number (SSN)

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

4.1.3 SCCP addressing

4.1.3.1 Introduction

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

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

- 1) Intra-PLMN addressing
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);
 - Routing indicator = 0 (Routing on global title);
 - b) Calling Party Address
 - SSN indicator = 1 (MAP SSNs always included);
 - Point code indicator = 0;
 - Global title indicator = 0100 (Global title includes translation type, numbering plan, encoding scheme and nature of address indicator);
 - the translation type field will be coded "00000000" (Not used);
 - Routing indicator = 0 (Routing on Global Title).

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

- E.212 numbering plan
An E.212 number must not be included as Global Title in an SCCP UNITDATA message. The translation of an E.212 number into a Mobile Global Title is applicable in a dialogue initiating VLR if the routing information towards the HLR is derived from the subscriber's IMSI. When an MS moves from one VLR service area to another, the new VLR may derive the address of the previous VLR from the Location Area Identification provided by the MS in the location registration request. The PLMN where the previous VLR is located is identified by the E.212 numbering plan elements of the Location Area Identification, i.e. the Mobile Country Code (MCC) and the Mobile Network Code (MNC).
- E.214 and E.164 numbering plans
Only address information belonging to either E.214 or E.164 numbering plan is allowed to be included as Global Title in the Called and Calling Party Address.
If the Calling Party Address associated with the dialogue initiating message contains a Global Title, the sending network entity shall include its E.164 entity number.
When receiving an SCCP UNITDATA message, SCCP shall accept either of the valid numbering plans in the Called Party Address and in the Calling Party Address.
When receiving an N-UNITDATA-REQUEST primitive from TC, SCCP shall accept an E.164 number or an E.214 number in the Called Address and in the Calling Address.

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

- the Mobile-services Switching Centre (MSC);
- the Home location Register (HLR);
- the Visitor Location Register (VLR);
- the Gateway Mobile-services Switching Centre (GMSC);
- the Interworking Mobile-services Switching Centre (IWMSC).

4.1.3.2 The Mobile-services Switching Centre (MSC)

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

4.1.3.2.1 MSC interaction during handover

The address is derived from the target Cellid.

4.1.3.2.2 MSC for short message routing

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

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

4.1.3.3 The Home Location Register (HLR)

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

4.1.3.3.1 During call set-up

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

If the calling subscriber is a fixed network subscriber, the interrogation can be initiated from the Gateway MSC of the home PLMN in the general case. If the topology of the network allows it, the interrogation

could be initiated from any Signalling Point which has MAP capabilities, e.g. local exchange, outgoing International Switching Centre (ISC), etc.

4.1.3.3.2 Before location updating completion

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

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

When answering to the VLR the HLR should insert its E.164 address in the Calling Party Address of the SCCP message containing the first responding CONTINUE message.

NOTE: As PCS 1900 networks do not use E.214 addresses, the use of an E.214 address may lead to interworking problems between GSM and PCS 1900 networks.

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

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

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

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

4.1.3.3.3 After location updating completion

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

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

4.1.3.3.4 VLR restoration

If a roaming number is requested by the HLR for an IMSI that has no data record in the interrogated VLR, the VLR provides the roaming number in the dialogue terminating message. Subsequently the VLR must retrieve the authentication data from the MS's HLR, if required, and must then trigger the restore data procedure. For this purpose, the VLR has to initiate in succession two independent dialogues with the

MS's HLR. The MTP and SCCP address information needed for routing 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 routing information towards the HLR.

4.1.3.4 The Visitor Location Register (VLR)

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

4.1.3.4.1 Inter-VLR information retrieval

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

4.1.3.4.2 HLR request

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

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

4.1.3.5 The Interworking MSC (IWMSC) for Short Message Service

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

4.1.3.6 The Equipment Identity Register (EIR)

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

4.1.3.7 Summary table

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

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

to from	fixed network	HLR	VLR	MSC	EIR	gsmSCF
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
visitor location register	---	I:SPC/GT E:GT T:MGT/MSISDN /HLR NUMBER (note)	I:SPC/GT E:GT T:VLR NUMBER	---	---	---
mobile- services switchin g 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	---
gsm Service Control Function	---	I:SPC/GT E:GT T:MSISDN	---	---	---	---

I: Intra-PLMN

E: Extra(Inter)-PLMN

T: Address Type

GT: Global Title

MGT: E.214 Mobile Global Title

SPC: Signalling Point Code

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

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

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

Table 4.1/1

4.2 Use of TC

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

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

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

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

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

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

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

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

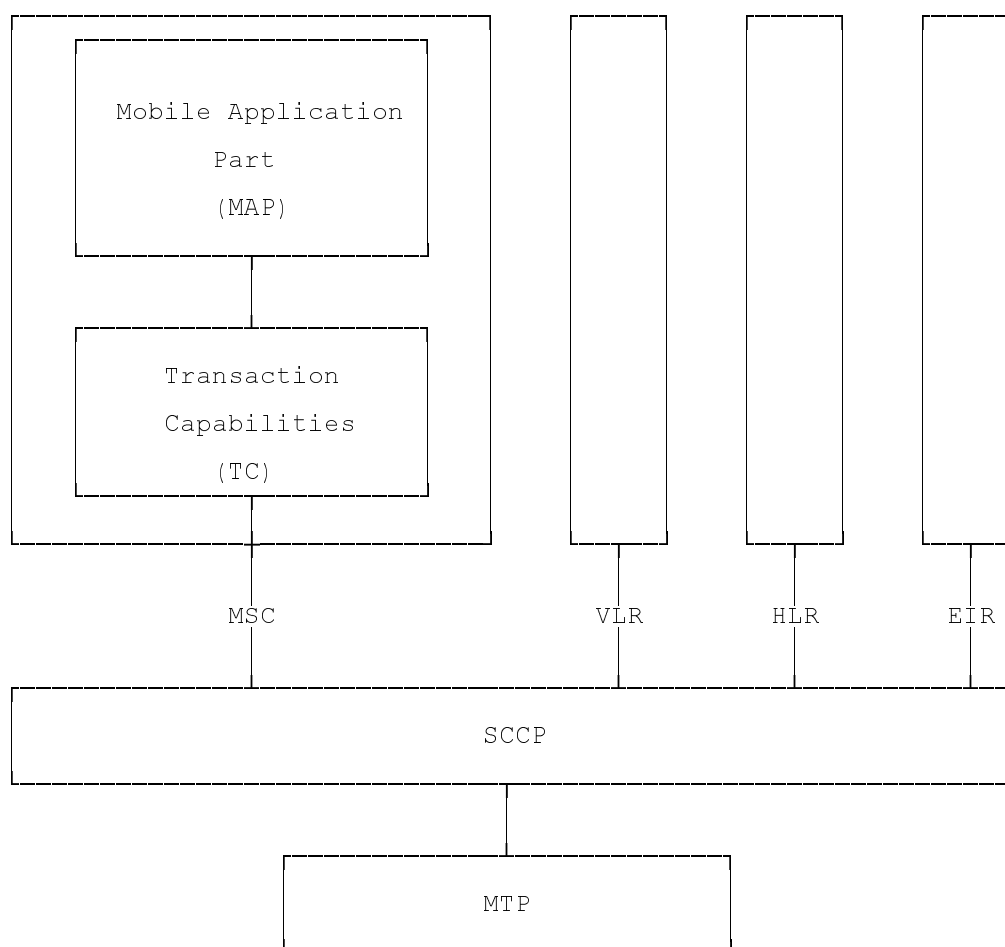


Figure 4.2/1: Facilities for supporting the Mobile Application Part in Signalling System No.7

5 General on MAP services

5.1 Terminology and definitions

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

5.2 Modelling principles

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

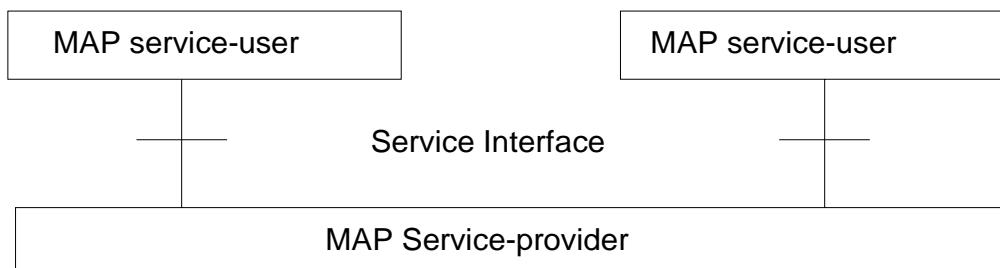


Figure 5.2/1: Modelling principles

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

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

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

MAP-ServicePrimitiveName type

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

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

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

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

5.3 Common MAP services

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

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

For these purposes the following common services are defined:

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

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

- M** the inclusion of the parameter is mandatory. The M category can be used for any primitive type and specifies that the corresponding parameter must be present in the indicated primitive type;
- O** the inclusion of the parameter is a service-provider option. The O category can be used in indication and confirm type primitives and is used for parameters that may optionally be included by the service-provider;
- U** the inclusion of the parameter is a service-user option. The U category can be used in request and response type primitives. The inclusion of the corresponding parameter is the choice of the service-user;
- C** the inclusion of the parameter is conditional. The C category can be used for the following purposes:
- to indicate that if the parameter is received from another entity it must be included for the service being considered;
 - to indicate that the service user must decide whether to include the parameter, based on the context on which the service is used;
 - to indicate that one of a number of mutually exclusive parameters must be included (e.g. parameters indicating a positive result versus parameters indicating a negative result);
 - to indicate that a service user optional parameter (marked "U") or a conditional parameter (marked "C") presented by the service user in a request or response type primitive is to be presented to the service user in the corresponding indication or confirm type primitive;
- (=) when appended to one of the above, this symbol means that the parameter takes the same value as the parameter appearing immediately to its left;

blank the parameter is not present.

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

5.3.1 MAP-OPEN service

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

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

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

Application context name:

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

Destination address:

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

Destination-reference:

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

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

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

Table 5.3/2: Use of the destination reference

Originating address:

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

Originating-reference:

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

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

Table 5.3/3: Use of the originating reference

Specific information:

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

Responding address:

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

Result:

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

Refuse reason:

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

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

5.3.2 MAP-CLOSE service

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

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

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

Release method:

This parameter can take the following two values:

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

Specific information:

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

5.3.3 MAP-DELIMITER service

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

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

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

Parameters	Request	Indication

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

5.3.4 MAP-U-ABORT service

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

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

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

User reason:

This parameter can take the following values:

- resource limitation (congestion);
the requested user resource is unavailable due to congestion;
- resource unavailable;
the requested user resource is unavailable for reasons other than congestion;
- application procedure cancellation;
the procedure is cancelled for reason detailed in the diagnostic information parameter;
- procedure error;
processing of the procedure is terminated for procedural reasons.

Diagnostic information:

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

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

Table 5.3/7: User reason and diagnostic information

Specific information:

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

5.3.5 MAP-P-ABORT service

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

Parameters		Indication
Provider reason		M
Source		M

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

Provider reason:

This parameter indicates the reason for aborting the MAP dialogue:

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

Source:

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

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

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

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

5.3.6 MAP-NOTICE service

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

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

Parameters		Indication
Problem diagnostic		M

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

Problem diagnostic:

This parameter can take one of the following values:

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

5.4 Sequencing of services

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

Opening:

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

Continuing:

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

Closing:

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

Aborting:

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

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

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

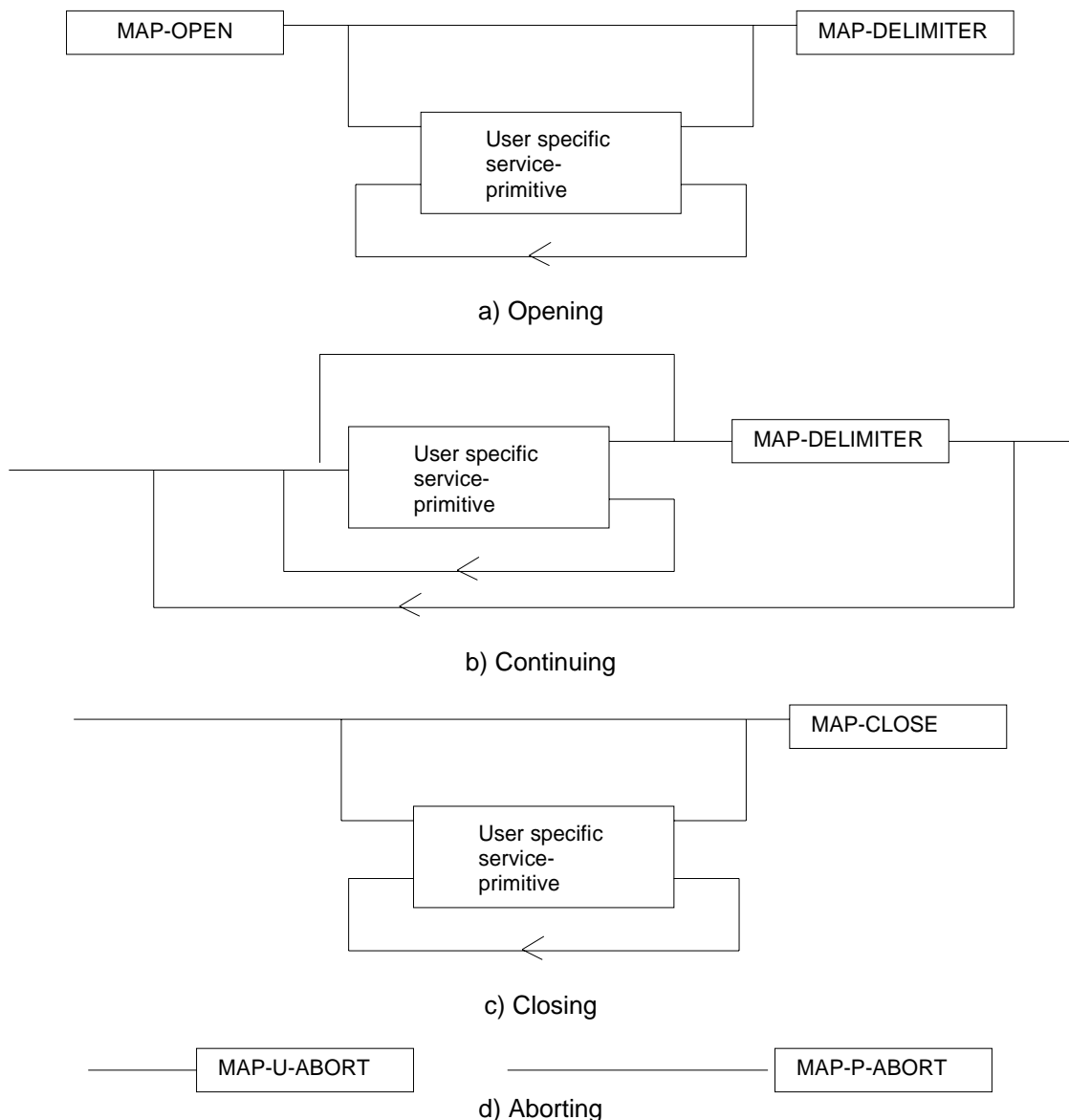


Figure 5.4/1: Sequencing of services

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

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

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

5.5 General rules for mapping of services onto TC

5.5.1 Mapping of common services

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

Protocol machine description is given in clauses 11 to 14.

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

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

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

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

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

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

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

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

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

NOTE 6: Only if opening sequence is pending.

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

5.5.2 Mapping of user specific services

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

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

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

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

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

Notes to tables 5.10 and 5.11:

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

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

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

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

5.6 Definition of parameters

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

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

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

Access connection status	5.6.9.3	Kc	5.6.7.4
Access signalling information	5.6.9.5	Linked Id	5.6.1.2
Alert Reason	5.6.8.8	LMSI	5.6.2.16
Authentication set list	5.6.7.1	Location Information	5.6.2.30
Basic Service Group	5.6.4.40	Location update type	5.6.9.6
Bearer service	5.6.4.38	More Messages To Send	5.6.8.7
BSS-apdu	5.6.9.1	MS ISDN	5.6.2.17
Call barring feature	5.6.4.19	MSC number	5.6.2.11
Call barring information	5.6.4.18	MSISdn-Alert	5.6.2.29
Call reference	5.6.5.1	MWD status	5.6.8.3
Called number	5.6.2.24	Network resources	5.6.10.1
Calling number	5.6.2.25	Network signal information	5.6.9.8
CAMEL Subscription Info Withdraw	5.6.3.38	New password	5.6.4.20
Category	5.6.3.1	No reply condition timer	5.6.4.7
Ciphering mode	5.6.7.7	ODB General Data	5.6.3.9
Cksn	5.6.7.5	ODB HPLMN Specific Data	5.6.3.10
CLI Restriction	5.6.4.5	OMC Id	5.6.2.18
CM service type	5.6.9.2	Originally dialled number	5.6.2.26
CUG feature	5.6.3.26	Originating entity number	5.6.2.10
CUG index	5.6.3.25	Override Category	5.6.4.4
CUG info	5.6.3.22	Previous location area Id	5.6.2.4
CUG interlock	5.6.3.24	Protocol Id	5.6.9.7
CUG Outgoing Access indicator	5.6.3.8	Provider error	5.6.1.3
CUG subscription	5.6.3.23	Rand	5.6.7.2
CUG Subscription Flag	5.6.3.37	Regional Subscription Data	5.6.3.11
Current location area Id	5.6.2.6	Regional Subscription Response	5.6.3.12
Current password	5.6.4.21	Requested Info	5.6.3.31
eMLPP Information	5.6.4.41	Roaming number	5.6.2.19
Equipment status	5.6.3.2	Roaming Restriction Due To	5.6.3.13
		Unsupported Feature	
Extensible Basic Service Group	5.6.3.5	Service centre address	5.6.2.27
Extensible Bearer service	5.6.3.3	SM Delivery Outcome	5.6.8.6
Extensible Call barring feature	5.6.3.21	SM-RP-DA	5.6.8.1
Extensible Call barring information	5.6.3.20	SM-RP-OA	5.6.8.2
Extensible Forwarding feature	5.6.3.16	SM-RP-PRI	5.6.8.5
Extensible Forwarding info	5.6.3.15	SM-RP-UI	5.6.8.4
Extensible Forwarding Options	5.6.3.18	Sres	5.6.7.3
Extensible No reply condition timer	5.6.3.19	SS-Code	5.6.4.1
Extensible SS-Data	5.6.3.29	SS-Data	5.6.4.3
Extensible SS-Info	5.6.3.14	SS-Info	5.6.4.24
Extensible SS-Status	5.6.3.17	SS-Status	5.6.4.2
Extensible Teleservice	5.6.3.4	Stored location area Id	5.6.2.5
External Signal Information	5.6.9.4	Subscriber State	5.6.3.30
Forwarded-to number	5.6.2.22	Subscriber Status	5.6.3.7
Forwarded-to subaddress	5.6.2.23	Supported CAMEL Phases	5.6.3.36
		Suppress T-CSI	5.6.3.33
Forwarding feature	5.6.4.16	Suppression of Announcement	5.6.3.32
Forwarding information	5.6.4.15	Target cell Id	5.6.2.8
Forwarding Options	5.6.4.6	Target location area Id	5.6.2.7
GMSC CAMEL Subscription Info	5.6.3.34	Target MSC number	5.6.2.12
Group Id	5.6.2.33	Teleservice	5.6.4.39
GSM bearer capability	5.6.3.6	TMSI	5.6.2.2
Guidance information	5.6.4.22	Trace reference	5.6.10.2
Handover number	5.6.2.21	Trace type	5.6.10.3
HLR Id	5.6.2.15	User error	5.6.1.4
HLR number	5.6.2.13	USSD Data Coding Scheme	5.6.4.36
HO-Number Not Required	5.6.6.7	USSD String	5.6.4.37
IMEI	5.6.2.3	VBS Data	5.6.3.40
IMSI	5.6.2.1	VGCS Data	5.6.3.39
Inter CUG options	5.6.3.27	VLR CAMEL Subscription Info	5.6.3.35
Intra CUG restrictions	5.6.3.28	VLR number	5.6.2.14
Invoke Id	5.6.1.1	Zone Code	5.6.2.28

5.6.1 Common parameters

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

5.6.1.1 Invoke Id

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

5.6.1.2 Linked Id

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

5.6.1.3 Provider error

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

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

5.6.1.4 User error

This parameter can take values as follows:

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

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

- b) Identification or numbering problem:
- unknown subscriber, i.e. no such subscription exists;
 - number changed, i.e. the subscription does not exist for that number any more;
 - unknown MSC;
 - unidentified subscriber, i.e. if the subscriber is not contained in the database and it has not or cannot be established whether or not a subscription exists;
 - unallocated roaming number;
 - unknown 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;
 - subsequent handover failure, i.e. handover to a third MSC failed for some reason.
- e) Operation and maintenance problem:
- tracing buffer full, i.e. tracing cannot be performed because the tracing capacity is exceeded.
- f) Call set-up problem:
- no roaming number available, i.e. a roaming number cannot be allocated because all available numbers are in use;
 - absent subscriber, i.e. the subscriber has activated the detach service or the system detects the absence condition.;
 - busy subscriber;
 - 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.
 - 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.

For definition of these errors see GSM 04.80.

- h) Short message problem:
- SM delivery failure with detailed reason as follows:
 - memory capacity exceeded;
 - MS protocol error;
 - MS not equipped;
 - unknown service centre (SC);
 - SC congestion;
 - invalid SME address;
 - subscriber is not an SC subscriber;
 - and possibly detailed diagnostic information, coded as specified in TS GSM 03.40, under SMS-SUBMIT-REPORT and SMS-DELIVERY-REPORT. If the SM entity which returns the SM Delivery Failure error includes detailed diagnostic information, it shall be forwarded in the MAP_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 GSM 03.40.
 - Absent Subscriber SM, i.e. the mobile terminated short message transfer cannot be completed because the network cannot contact the subscriber. Diagnostic information regarding the reason for the subscriber's absence may be included with this error.

5.6.2 Numbering and identification parameter

5.6.2.1 IMSI

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

5.6.2.2 TMSI

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

5.6.2.3 IMEI

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

5.6.2.4 Previous location area Id

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

5.6.2.5 Stored location area Id

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

5.6.2.6 Current location area Id

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

5.6.2.7 Target location area Id

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

5.6.2.8 Target cell Id

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

5.6.2.9 [Spare]

5.6.2.10 Originating entity number

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

5.6.2.11 MSC number

This parameter refers to the ISDN number of an MSC.

5.6.2.12 Target MSC number

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

5.6.2.13 HLR number

This parameter refers to the ISDN number of an HLR.

5.6.2.14 VLR number

This parameter refers to the ISDN number of a VLR.

5.6.2.15 HLR Id

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

5.6.2.16 LMSI

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

5.6.2.17 MS ISDN

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

5.6.2.18 OMC Id

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

5.6.2.19 Roaming number

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

5.6.2.20 [Spare]

5.6.2.21 Handover number

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

5.6.2.22 Forwarded-to number

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

5.6.2.23 Forwarded-to subaddress

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

5.6.2.24 Called number

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

5.6.2.25 Calling number

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

5.6.2.26 Originally dialled number

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

5.6.2.27 Service centre address

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

5.6.2.28 Zone Code

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

5.6.2.29 MSISdn-Alert

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

5.6.2.30 Location Information

This parameter indicates the location of the served subscriber as defined in GSM 03.18.

5.6.2.31 GMSC Address

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

5.6.2.32 VMSC Address

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

5.6.2.33 Group Id

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

5.6.3 Subscriber management parameters

5.6.3.1 Category

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

5.6.3.2 Equipment status

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

5.6.3.3 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 TS GSM 02.02. This parameter is used only for subscriber profile management. Extensible Bearer service values include all values defined for a Bearer service parameter (5.6.4.38).

5.6.3.4 Extensible Teleservice

This parameter may refer to a single teleservice, a set of teleservices or to all teleservices as defined in TS GSM 02.03. This parameter is used only for subscriber profile management. Extensible Teleservice values include all values defined for a Teleservice parameter (5.6.4.39).

5.6.3.5 Extensible Basic Service Group

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

5.6.3.6 GSM bearer capability

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

5.6.3.7 Subscriber Status

This parameter refers to the barring status of the subscriber:

- service granted;
- Operator Determined Barring.

5.6.3.8 CUG Outgoing Access indicator

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

5.6.3.9 Operator Determined Barring General Data

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

- All outgoing calls barred;
- International outgoing calls barred;
- International outgoing calls except those to the home PLMN country barred;
- 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.

5.6.3.10 ODB HPLMN Specific Data

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

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

5.6.3.11 Regional Subscription Data

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

5.6.3.12 Regional Subscription Response

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

5.6.3.13 Roaming Restriction Due To Unsupported Feature

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

5.6.3.14 Extensible SS-Info

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

- extensible forwarding information (see subclause 5.6.3.15);
- call barring information (see subclause 5.6.3.20);
- extensible CUG info (see subclause 5.6.3.22);
- extensible SS-Data (see subclause 5.6.3.29).

5.6.3.15 Extensible Forwarding information

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

- the SS-Code of the relevant call forwarding service (see subclause 5.6.4.1);
- if required, a list of extensible forwarding feature parameters (see subclause 5.6.3.16).
The list may contain one item per Basic Service Group.

5.6.3.16 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 subclause 5.6.3.5);
- extensible SS-Status (see subclause 5.6.3.17);
- forwarded-to number (see subclause 5.6.2.22);
- forwarded-to subaddress (see subclause 5.6.2.23);
- extensible forwarding options (see subclause 5.6.3.18);
- extensible no reply condition timer (see subclause 5.6.4.19).

5.6.3.17 Extensible SS-Status

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

5.6.3.18 Extensible Forwarding Options

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

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

5.6.3.19 Extensible No reply condition timer

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

5.6.3.20 Extensible Call barring information

This parameter contains for each call barring service:

- SS-Code (see subclause 5.6.4.1);
- a list of extensible call barring feature parameters (see subclause 5.6.3.21).
The list may contain one item per Basic Service Group.

5.6.3.21 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 subclause 5.6.3.5);
- provisioned SS-Status (see subclause 5.6.3.17).

5.6.3.22 CUG info

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

- CUG subscriptionList;
- CUG featureList.

5.6.3.23 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.

5.6.3.24 CUG interlock

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

5.6.3.25 CUG index

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

5.6.3.26 CUG feature

This parameter contains two parameters which 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 TS GSM 02.85 for meaning of this parameter.

5.6.3.27 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).

5.6.3.28 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.

5.6.3.29 Extensible SS-Data

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

- SS-Code (see subclause 5.6.4.1);
- Extensible SS-Status (if applicable) (see subclause 5.6.3.17);
- Extensible Override subscription option (if applicable) (see subclause 5.6.3.30);
- Extensible CLI Restriction (if applicable) (see subclause 5.6.3.31);
- Extensible Basic Service Group Code (see subclause 5.6.3.5).

5.6.3.30 Subscriber State

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

5.6.3.31 Requested Info

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

5.6.3.32 Suppression of Announcement

This parameter indicates if the announcement or tones shall be suppressed as defined in GSM 03.78.

5.6.3.33 Suppress T-CSI

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

5.6.3.34 GMSC CAMEL Subscription Info

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

5.6.3.35 VLR CAMEL Subscription Info

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

5.6.3.36 Supported CAMEL Phases

This parameter indicates which phases of CAMEL are supported.

5.6.3.37 CUG Subscription Flag

This parameter indicates a that a subscriber with a T-CSI also has a CUG subscription. It is defined in TS GSM 03.78.

5.6.3.38 CAMEL Subscription Info Withdraw

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

5.6.3.39 Voice Group Call Service (VGCS) Data

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

5.6.3.40 Voice Broadcast Service (VBS) Data

This parameter refers to one or more groups a subscriber may be 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.

5.6.4 Supplementary services parameters

5.6.4.1 SS-Code

This parameter may refer to one supplementary service or a set of supplementary services as defined in TS GSM 02.04. For MAP Release '96 this includes:

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

5.6.4.2 SS-Status

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

5.6.4.3 SS-Data

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

- SS-Code (see subclause 5.6.4.1);
- SS-Status (if applicable) (see subclause 5.6.4.2);
- Override subscription option (see subclause 5.6.4.4);
- CLI Restriction (see subclause 5.6.4.5);
- Basic Service Group Code (see subclause 5.6.4.40).

5.6.4.4 Override Category

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

- Enabled;
- Disabled.

5.6.4.5 CLI Restriction Option

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

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

5.6.4.6 Forwarding Options

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

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

5.6.4.7 No reply condition timer

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

5.6.4.8 - 5.6.4.14 [Spare]

5.6.4.15 Forwarding information

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

- the SS-Code of the relevant call forwarding service (see subclause 5.6.4.1);
- if required, a list of forwarding feature parameters (see subclause 5.6.4.16).
The list may contain one item per Basic Service Group.

5.6.4.16 Forwarding feature

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

- Basic Service Group (see subclause 5.6.4.40);
- SS-Status (see subclause 5.6.4.2);
- forwarded-to number (see subclause 5.6.2.22);
- forwarded-to subaddress (see subclause 5.6.2.23);
- forwarding options (see subclause 5.6.4.6);

- no reply condition timer (see subclause 5.6.4.7).

5.6.4.17 [Spare]

5.6.4.18 Call barring information

This parameter contains for each call barring service:

- SS-Code (see subclause 5.6.4.1);
- a list of call barring feature parameters (see subclause 5.6.4.19).
The list may contain one item per Basic Service Group.

5.6.4.19 Call barring feature

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

- Basic Service Group (see subclause 5.6.4.40);
- SS-Status (see subclause 5.6.4.2).

5.6.4.20 New password

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

5.6.4.21 Current password

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

5.6.4.22 Guidance information

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

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

5.6.4.23 [Spare]

5.6.4.24 SS-Info

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

- forwarding information (see subclause 5.6.4.15);
- call barring information (see subclause 5.6.4.18);
- CUG info (see subclause 5.6.4.8);
- SS-Data (see subclause 5.6.4.3).
- eMLPP information (see subclause 5.6.4.41).

5.6.4.25 - 5.6.4.35 [Spare]**5.6.4.36 USSD Data Coding Scheme**

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

5.6.4.37 USSD String

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

5.6.4.38 Bearer service

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

5.6.4.39 Teleservice

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

5.6.4.40 Basic Service Group

This parameter refers to the Basic Service Group either as a bearer service (see subclause 5.6.4.38) or a teleservice (see subclause 5.6.4.39). 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.

5.6.4.41 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.

5.6.5 Call parameters**5.6.5.1 Call reference number**

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

5.6.5.2 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).

5.6.5.3 OR interrogation

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

5.6.5.4 OR capability

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

5.6.5.5 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.

5.6.5.6 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.

5.6.5.7 O-CSI

This parameter identifies the subscriber as having originating CAMEL services as defined in TS GSM 03.78

5.6.6 Radio parameters

5.6.6.1 - 5.6.6.6 [Spare]

5.6.6.7 HO-Number Not Required

This parameter indicates that no handover number allocation is necessary.

5.6.7 Authentication parameters

5.6.7.1 Authentication set list

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

- Rand;
- Sres;
- Kc.

5.6.7.2 Rand

This parameter represents a random number used for authentication.

5.6.7.3 Sres

This parameter represents the response to an authentication request.

5.6.7.4 Kc

This parameter refers to a key used for ciphering purposes.

5.6.7.5 [Spare]**5.6.7.6 Cksn**

This parameter refers to a ciphering key sequence number.

5.6.7.7 Ciphering mode

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

- no encryption;
- identification of specific ciphering algorithm.

5.6.8 Short message parameters**5.6.8.1 SM-RP-DA**

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

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

5.6.8.2 SM-RP-OA

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

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

5.6.8.3 MWD status

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

5.6.8.4 SM-RP-UI

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

5.6.8.5 SM-RP-PRI

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

5.6.8.6 SM Delivery Outcome

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

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

5.6.8.7 More Messages To Send

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

5.6.8.8 Alert Reason

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

- MS present;
- Memory Available.

5.6.8.9 Absent Subscriber Diagnostic SM

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

5.6.9 Access and signalling system related parameters

5.6.9.1 BSS-apdu

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

5.6.9.2 CM service type

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

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

5.6.9.3 Access connection status

This parameter represents the following access connection status information:

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

5.6.9.4 External Signal Information

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

5.6.9.5 Access signalling information

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

5.6.9.6 Location update type

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

5.6.9.7 Protocol ID

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

The following values are defined:

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

5.6.9.8 Network signal information

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

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

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

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

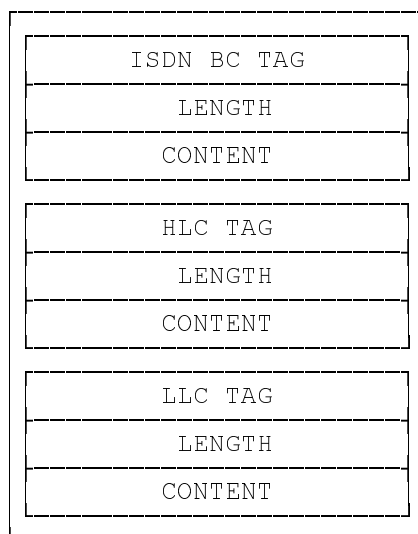


Figure 5.6/1: Network signal information parameter

5.6.10 System operations parameters

5.6.10.1 Network resources

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

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

- EIR;
- radio sub-system.

5.6.10.2 Trace reference

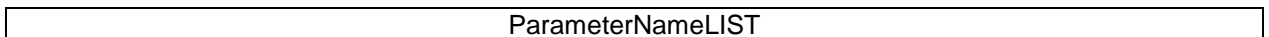
This parameter represents a reference associated with a tracing request. The parameter is managed by OMC.

5.6.10.3 Trace type

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

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

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



in the tables where ParameterName refers to one of the parameters defined in subclause 5.6. This corresponds to the following construction rule:

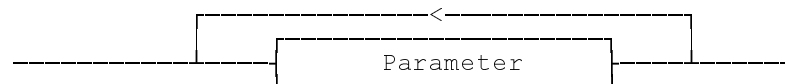


Figure 5.7/1: Construction of Lists

6 Mobility services

6.1 Location management services

6.1.1 MAP_UPDATE_LOCATION_AREA service

6.1.1.1 Definition

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

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

6.1.1.2 Service primitives

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

Table 6.1/1: MAP_UPDATE_LOCATION_AREA

6.1.1.3 parameter definitions and use

Invoke Id

See definition in subclause 5.6.1.

Target location area Id

See definition in subclause 5.6.2.

Location update type

See definition in subclause 5.6.9.

IMSI

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

TMSI

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

Previous location area Id

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

CKSN

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

User error

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

- unknown subscriber;
This cause is used if the subscriber is not known in the VLR and even a correlated request to the subscriber's HLR gives a negative result (i.e. the IMSI is not allocated to a subscriber).
- unknown location area;
This cause is used if the target location area identity given is not known in the VLR.
- roaming not allowed;
This cause is used if the MS is not allowed to roam into the target location area indicated in the MAP_UPDATE_LOCATION_AREA Req. The cause will be qualified according to the roaming restriction reason, i.e. one of "National Roaming Not Allowed", "PLMN Not Allowed", "Location Area Not Allowed", or "Operator Determined Barring".
- illegal subscriber;
This error is sent if a correlated authentication procedure has not authenticated the subscriber.
- illegal equipment;
This error is sent if an IMEI check failed, i.e. the IMEI is blacklisted or not white-listed.
- system failure;
- unexpected data value.

Provider error

For definition of provider errors see subclause 5.6.1.

6.1.2 MAP_UPDATE_LOCATION service

6.1.2.1 Definition

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

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

6.1.2.2 Service primitives

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

Table 6.1/2: MAP_UPDATE_LOCATION

6.1.2.3 Parameter definitions and use

Invoke Id

See definition in subclause 5.6.1.

IMSI

See definition in subclause 5.6.2.

MSC Address

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

VLR number

See definition in subclause 5.6.2.

LMSI

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

HLR number

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

User error

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

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

Provider error

For definition of provider errors see subclause 5.6.1.

6.1.3 MAP_CANCEL_LOCATION service

6.1.3.1 Definition

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

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

6.1.3.2 Service primitives

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

Table 6.1/3: MAP_CANCEL_LOCATION

6.1.3.3 Parameter definitions and use

Invoke Id

See definition in subclause 5.6.1.

IMSI

See definition in subclause 5.6.2.

LMSI

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

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

User error

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

- unexpected data value.

Provider error

For definition of provider errors see subclause 5.6.1.

6.1.4 MAP_SEND_IDENTIFICATION service

6.1.4.1 Definition

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

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

6.1.4.2 Service primitives

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

Table 6.1/4: MAP_SEND_IDENTIFICATION

6.1.4.3 Parameter definitions and use

Invoke Id

See definition in subclause 5.6.1.

TMSI

See definition in subclause 5.6.2.

IMSI

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

Authentication set

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

User error

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

- unidentified subscriber.

Provider error

For definition of provider errors see subclause 5.6.1.

6.1.5 MAP_DETACH_IMSI service

6.1.5.1 Definition

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

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

6.1.5.2 Service primitives

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

Table 6.1/5: MAP_DETACH_IMSI

6.1.5.3 Parameter definitions and use

Invoke Id

See definition in subclause 5.6.1.

IMSI

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

TMSI

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

6.1.6 MAP_PURGE_MS service

6.1.6.1 Definition

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

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

6.1.6.2 Service primitives

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

Table 6.1/6: MAP_PURGE_MS

6.1.6.3 Parameter definitions and use

Invoke ID

See definition in subclause 5.6.1.

IMSI

See definition in subclause 5.6.2.

VLR number

See definition in subclause 5.6.2.

Provider error

See definition of provider errors in subclause 5.6.1.

6.2 Paging and search

6.2.1 MAP_PAGE service

6.2.1.1 Definition

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

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

6.2.1.2 Service primitives

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

Table 6.2/1: MAP_PAGE

6.2.1.3 Parameter definitions and use

Invoke Id

See definition in subclause 5.6.1.

IMSI

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

Stored location area Id

See definition in subclause 5.6.2.

TMSI

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

User error

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

- absent subscriber;
- unknown location area;
- busy subscriber;
- system failure;
This corresponds to the case where there is no call associated with the MAP_PAGE service, i.e. if the call has been released but the dialogue to the VLR has not been aborted.
- unexpected data value.

Provider error

See definition in subclause 5.6.1.

6.2.2 MAP_SEARCH_FOR_MS service

6.2.2.1 Definition

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

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

6.2.2.2 Service primitives

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

Table 6.2/2: MAP_SEARCH_FOR_MS

6.2.2.3 Parameter definitions and use

Invoke Id

See definition in subclause 5.6.1.

IMSI

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

Current location area Id

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

User error

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

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

Provider error

See definition in subclause 5.6.1.

6.3 Access management services

6.3.1 MAP_PROCESS_ACCESS_REQUEST service

6.3.1.1 Definition

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

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

6.3.1.2 Service primitives

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

Table 6.3/1: MAP_PROCESS_ACCESS_REQUEST**6.3.1.3 Parameter definitions and use**Invoke Id

See definition in subclause 5.6.1.

CM service type

See definition in subclause 5.6.9.

Access connection status

See definition in subclause 5.6.9.

Current Location Area Id

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

TMSI

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

Cksn

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

IMSI

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

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

IMEI

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

MSISDN

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

User error

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

- unidentified subscriber;
- illegal subscriber;
This error is sent if a correlated authentication procedure has not authenticated the subscriber.
- illegal equipment;
This error is sent if an IMEI check failed, i.e. the IMEI is blacklisted or not white-listed.
- roaming not allowed;
This cause is used after VLR restart if the subscriber has no subscription for the current location area, e.g. due to regional subscription. The cause will be qualified by "location area not allowed" or "national roaming not allowed", respectively.
- unknown location area;
- system failure;
- unexpected data value.

Provider error

For definition of provider errors see subclause 5.6.1.

6.4 Handover services

6.4.1 MAP_PREPARE_HANDOVER service

6.4.1.1 Definition

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

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

6.4.1.2 Service primitives

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

Table 6.4/1: MAP_PREPARE_HANDOVER

6.4.1.3 Parameter use

Invoke Id

For definition of this parameter see subclause 5.6.1.

Target Cell Id

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

HO-Number Not Required

For definition of this parameter see subclause 5.6.6.

BSS-APDU

For definition of this parameter see subclause 5.6.9.

Handover Number

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

User error

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

- No handover number available;
- System failure;
- Unexpected data value;
- DataMissing.

Provider error

See definition of provider errors in subclause 5.6.1.

6.4.2 MAP_SEND_END_SIGNAL service

6.4.2.1 Definition

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

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

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

6.4.2.2 Service primitives

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

Table 6.4/2: MAP_SEND_END_SIGNAL

6.4.2.3 Parameter use

Invoke Id

For definition of this parameter see subclause 5.6.1.

BSS-APDU

For definition of this parameter see subclause 5.6.9.

Provider error

For definition of this parameter see subclause 5.6.1.

6.4.3 MAP_PROCESS_ACCESS_SIGNALLING service

6.4.3.1 Definition

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

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

6.4.3.2 Service primitives

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

Table 6.4/3: MAP_PROCESS_ACCESS_SIGNALLING

6.4.3.3 Parameter use

Invoke Id

For definition of this parameter see subclause 5.6.1.

BSS-APDU

For definition of this parameter see subclause 5.6.9.

6.4.4 MAP_FORWARD_ACCESS_SIGNALLING service

6.4.4.1 Definition

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

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

6.4.4.2 Service primitives

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

Table 6.4/4: MAP_FORWARD_ACCESS_SIGNALLING

6.4.4.3 Parameter use

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

Invoke Id

For definition of this parameter see subclause 5.6.1.

BSS-APDU

For definition of this parameter see subclause 5.6.9.

6.4.5 MAP_PREPARE_SUBSEQUENT_HANDOVER service

6.4.5.1 Definition

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

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

6.4.5.2 Service primitives

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

Table 6.4/5: MAP_PREPARE_SUBSEQUENT_HANDOVER

6.4.5.3 Parameter use

Invoke Id

For definition of this parameter see subclause 5.6.1.

Target Cell Id

For definition of this parameter see subclause 5.6.2.

Target MSC Number

For definition of this parameter see subclause 5.6.2.

BSS-APDU

For definition of this parameter see subclause 5.6.9.

User error

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

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

Provider error

For definition of this parameter see subclause 5.6.1.

6.4.6 MAP_ALLOCATE_HANOVER_NUMBER service

6.4.6.1 Definition

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

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

6.4.6.2 Service primitives

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

Table 6.4/6: MAP_ALLOCATE_HANOVER_NUMBER

6.4.6.3 Parameter use

Invoke Id

For definition of this parameter see subclause 5.6.1.

User error

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

- No handover number available.

Provider error

For definition of this parameter see subclause 5.6.1.

6.4.7 MAP_SEND_HANOVER_REPORT service

6.4.7.1 Definition

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

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

6.4.7.2 Service primitives

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

Table 6.4/7: MAP_SEND_HANOVER_REPORT

6.4.7.3 Parameter use

Invoke Id

For definition of this parameter see subclause 5.6.1.

Handover Number

For definition of this parameter see subclause 5.6.2.

Linked Id

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

Provider error

For definition of this parameter see subclause 5.6.1.

6.5 Authentication management services

6.5.1 MAP_AUTHENTICATE service

6.5.1.1 Definition

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

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

6.5.1.2 Service primitives

The service primitives are shown in table 6.5/1

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

Table 6.5/1: MAP_AUTHENTICATE parameters

6.5.1.3 Parameter use

Invoke id

See subclause 5.6.1 for the use of this parameter.

RAND

See subclause 5.6.7 for the use of this parameter.

CKSN

See subclause 5.6.7 for the use of this parameter.

SRES

See subclause 5.6.7 for the use of this parameter.

Provider error

See subclause 5.6.1 for the use of this parameter.

6.5.2 MAP_SEND_AUTHENTICATION_INFO service

6.5.2.1 Definition

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

If the HLR cannot provide the VLR with triplets, an empty response is returned. The VLR may then re-use old authentication triplets, except where this is forbidden under the conditions specified in GSM 03.20 [24].

If the VLR 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 shall fail.

Security related network functions are further described in GSM 03.20.

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

6.5.2.2 Service primitives

The service primitives are shown in table 6.5/2.

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

Table 6.5/2: MAP_SEND_AUTHENTICATION_PARAMETERS parameters

6.5.2.3 Parameter use

Invoke id

See subclause 5.6.1 for the use of this parameter.

IMSI

See subclause 5.6.2 for the use of this parameter.

AuthenticationSetList

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

User error

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

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

Provider error

See subclause 5.6.1 for the use of this parameter.

6.6 Security management services

6.6.1 MAP_SET_CIPHERING_MODE service

6.6.1.1 Definitions

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

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

6.6.1.2 Service primitives

The service primitives are shown in table 6.6/1

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

Table 6.6/1: MAP_SET_CIPHERING_MODE parameters

6.6.1.3 Parameter use

Invoke id

See subclause 5.6.1 for the use of this parameter.

Ciphering mode

See subclause 5.6.7 for the use of this parameter.

Kc

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

6.7 International mobile equipment identities management services

6.7.1 MAP_CHECK_IMEI service

6.7.1.1 Definition

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

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

6.7.1.2 Service primitives

The service primitives are shown in table 6.7/1.

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

Table 6.7/1: MAP_CHECK_IMEI parameters

6.7.1.3 Parameter use

Invoke id

See subclause 5.6.1 for the use of this parameter.

IMEI

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

Equipment status

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

User error

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

- unknown 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 subclause 5.6.1 for the use of this parameter.

6.7.2 MAP_OBTAIN_IMEI service

6.7.2.1 Definition

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

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

6.7.2.2 Service primitives

The service primitives are shown in table 6.7/2.

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

Table 6.7/2: MAP_OBTAIN_IMEI parameters

6.7.2.3 Parameter useInvoke id

See subclause 5.6.1 for the use of this parameter.

IMEI

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

User error

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

Provider error

See subclause 5.6.1 for the use of this parameter.

6.8 Subscriber management services**6.8.1 MAP-INSERT-SUBSCRIBER-DATA service****6.8.1.1 Definition**

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

- the operator has changed the subscription of one or more supplementary services, basic services or data of a subscriber. Note that in case of withdrawal of a Basic or Supplementary service this primitive shall not be used;
- the operator has applied, changed or removed Operator Determined Barring;
- the subscriber has changed data concerning one or more supplementary services by using a subscriber procedure;
- the HLR provides the VLR with subscriber parameters at location updating of a subscriber or at restoration. In this case, this service is used to indicate explicitly that a supplementary service is not provisioned, if the supplementary service specification requires it. The only supplementary services which have this requirement are the CLIR and COLR services.

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

6.8.1.2 Service primitives

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

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

6.8.1.3 Parameter use

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

IMSI

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

MSISDN

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

Category

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

Subscriber Status

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

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

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

Bearer service List

A list of Extensible Bearer service parameters (Extensible Bearer service is defined in subclause 5.6). 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.

Teleservice List

A list of Extensible Teleservice parameters (Extensible Teleservice is defined in subclause 5.6). 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).

If the VLR 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).

Forwarding information List

A list of Extensible Forwarding information parameters (Extensible Forwarding information is defined in subclause 5.6). It includes Call Forwarding services either at location updating or at restoration or when they are changed. Each 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 subclause 5.6).

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

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 forwarded-to number and, if available, the forwarded-to subaddress. In other states the forwarded-to number and, if applicable, the forwarded-to subaddress shall not be included. For call forwarding unconditional the forwarded-to number and, if applicable, the forwarded-to subaddress shall not be included. If the VLR does not receive a forwarded-to subaddress then it shall assume that a forwarded-to subaddress has not been registered.

The 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).

Call barring information List

A list of Extensible Call barring information parameters (Extensible Call barring information is defined in subclause 5.6). It includes Call Barring services either at location updating or at restoration or when they are changed. Each 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 subclause 5.6).

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

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).

CUG information List

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

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

SS-Data List

A list of Extensible SS-Data parameters (Extensible SS-Data is defined in subclause 5.6). 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 subclause 6.8.1.4.

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

Operator Determined Barring General data

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

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

Operator Determined Barring HPLMN data

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

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

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 subclause 5.6). 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.

Roaming Restriction Due To Unsupported Feature

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

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

Regional Subscription Data

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

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

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

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

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

- Too Many Zone Codes: more than 10 Zone Codes are to be stored in the VLR;
- Regional Subscription Not Supported by the VLR;
- Zone Codes Conflict: the VLR detects that the zone codes indicate conflicting service permission for a location area.

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

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

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

Voice Broadcast Data

This parameter contains a list of group id's a user might have subscribed to; (VBS-Data is defined in subclause 5.6). 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-data 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. Voice Group Call Data

This parameter contains a list of group id's a user might have subscribed to; see subclause 5.6.

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.

SS-Code List

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

Regional Subscription Response

If included in the response this parameter indicates one of:

- MSC Area Restricted entirely because of regional subscription;
- Too Many Zone Codes to be inserted;
- Zone Codes Conflict;
- Regional Subscription not Supported by the VLR.

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

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. 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. The entire set of CAMEL Subscription Info is sent. If a set of CAMEL Subscription Info is already stored in the VLR it is replaced by the received data.

Supported CAMEL Phases

The use of this parameter and the requirements for its presence are specified in GSM 03.78. A VLR not supporting any CAMEL-Phase may omit this parameter.

User error

Only one of the following values is applicable:

- Unidentified subscriber;
- Data missing;
- Unexpected data value.

6.8.1.4 Basic service information related to supplementary services

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

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

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

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

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

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

- the supplementary service is applicable to at least one basic service in the Basic Service Group; and
- the subscriber has a subscription to at least one basic service in the Basic Service Group.

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

6.8.2 MAP-DELETE-SUBSCRIBER-DATA service

6.8.2.1 Definition

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

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

6.8.2.2 Service primitives

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

Table 6.8/2: MAP-DELETE-SUBSCRIBER-DATA**6.8.2.3 Parameter use**

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

Basic service List

A list of Extensible Basic service parameters (Extensible Basic service is defined in subclause 5.6). It is used when one, several or all basic services are to be withdrawn from the subscriber. If the VLR receives a value for an Extensible Basic Service which it does not support, it shall ignore that value.

SS-Code List

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

There are three possible options:

- deletion of basic service(s);
The parameter Basic service List is only included.
- deletion of supplementary service(s);
The parameter SS-Code List is only included.
- deletion of basic and supplementary services;
Both Basic service List and SS-Code List are included.

Roaming Restriction Due To Unsupported Feature

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

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

CAMEL Subscription Info Withdraw

This parameter is used to indicate that CAMEL Subscription Info shall be deleted from the VLR. All CAMEL Subscription Info for the subscriber shall be deleted.

Regional Subscription Identifier

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

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

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

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

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

VBS Group Indication

Contains an indication (flag) which is used if all Group Id's 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.

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.

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.

Regional Subscription Response

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

- MSC Area Restricted;
- Regional Subscription Not Supported.

User error

Only one of the following values is applicable:

- Unidentified subscriber;
- Data missing;
- Unexpected data value.

6.9 Identity management services

6.9.1 MAP-PROVIDE-IMSI service

6.9.1.1 Definition

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

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

6.9.1.2 Service primitives

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

Table 6.9/1: MAP-PROVIDE-IMSI**6.9.1.3 Parameter use**

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

IMSI

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

User error

Only one of the following values is applicable:

- Absent subscriber.

6.9.2 MAP-FORWARD-NEW-TMSI service**6.9.2.1 Definition**

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

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

6.9.2.2 Service primitives

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

Table 6.9/2: MAP-FORWARD-NEW-TMSI**6.9.2.3 Parameter use**

The parameter TMSI is described in subclause 5.6.

6.10 Fault recovery services**6.10.1 MAP_RESET service****6.10.1.1 Definition**

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

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

6.10.1.2 Service primitives

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

Table 6.10/1: MAP_RESET

6.10.1.3 Parameter definition and use

Invoke Id

See definition in subclause 5.6.1.

HLR number

See definition in subclause 5.6.2.

HLR Id LIST

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

6.10.2 MAP_FORWARD_CHECK_SS_INDICATION service

6.10.2.1 Definition

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

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

6.10.2.2 Service primitives

Parameter name	Request	Indication
Invoke Id	M	M(=)

Table 6.10/2: MAP_FORWARD_CHECK_SS_INDICATION

6.10.2.3 Parameter definition and use

Invoke Id

See definition in subclause 5.6.1.

6.10.3 MAP_RESTORE_DATA service

6.10.3.1 Definition

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

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

6.10.3.2 Service primitives

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

Table 6.10/3: MAP_RESTORE_DATA

6.10.3.3 Parameter definitions and use

Invoke Id

See definition in subclause 5.6.1.

IMSI

See definition in subclause 5.6.2.

LMSI

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

HLR number

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

MS Not Reachable Flag

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

User error

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

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

Provider error

For definition of provider errors see subclause 5.6.1.

6.11 Subscriber Information services

6.11.1 MAP-ANY-TIME-INTERROGATION service

6.11.1.1 Definition

This service is used by the gsmSCF, to request information (e.g. subscriber state and location) from the HLR at any time.

6.11.1.2 Service primitives

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

Table 6.11/1: Any_Time_Interrogation

6.11.1.3 Parameter definition and use

All parameters are described in subclause 5.6.

The HLR may be able to use the value of the parameter gsmSCF-address to screen an MAP_Any_Time_Interrogation indication.

The use of the parameters and the requirements for their presence are specified in GSM 03.78.

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 subclause 5.6.1.

6.11.2 MAP-PROVIDE-SUBSCRIBER-Info service**6.11.2.1 Definition**

This service is used to request information (e.g. subscriber state and location) from the VLR at any time.

6.11.2.2 Service primitives

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

Table 6.11/2: Provide_Subscriber_Information

6.11.2.3 Parameter definition and use

All parameters are defined in section 5.6. The use of these parameters and the requirements for their presence are specified in GSM 03.18

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 subclause 5.6.1.

7 Operation and maintenance services

7.1 Subscriber tracing services

7.1.1 MAP-ACTIVATE-TRACE-MODE service

7.1.1.1 Definition

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

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

7.1.1.2 Service primitives

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

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

7.1.1.3 Parameter use

Invoke id

See definition in subclause 5.6.1.

IMSI

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

Trace reference

See definition in subclause 5.6.10.

Trace type

See definition in subclause 5.6.10.

OMC Id

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

User error

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

- Unidentified Subscriber;
- Facility Not Supported;
- Tracing Buffer Full;
- System Failure;
- Unexpected Data Value;
- Data missing.

Provider error

For definition of provider errors see subclause 5.6.1.

7.1.2 MAP-DEACTIVATE-TRACE-MODE service

7.1.2.1 Definition

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

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

7.1.2.2 Service primitives

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

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

7.1.2.3 Parameter use

Invoke id

See definition in subclause 5.6.1.

IMSI

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

Trace reference

See definition in subclause 5.6.10.

User error

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

- Unidentified Subscriber;
- Facility Not Supported;
- System Failure;
- Unexpected Data Value;
- Data missing.

Provider error

For definition of provider errors see subclause 5.6.1.

7.1.3 MAP-TRACE-SUBSCRIBER-ACTIVITY service

7.1.3.1 Definition

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

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

7.1.3.2 Service primitives

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

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

7.1.3.3 Parameter use

Invoke id

See definition in subclause 5.6.1.

IMSI

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

Trace reference

See definition in subclause 5.6.10.

Trace type

See definition in subclause 5.6.10.

OMC Id

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

7.2 Other operation and maintenance services

7.2.1 MAP-SEND-IMSI service

7.2.1.1 Definition

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

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

7.2.1.2 Service primitives

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

Table 7.2/1: MAP-SEND-IMSI

7.2.1.3 Parameter use

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

User error

Only one of the following values is applicable:

- Unknown subscriber;
- Unexpected data value;
- Data missing.

8 Call handling services

8.1 MAP_SEND_ROUTING_INFORMATION service

8.1.1 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 8.1/1.

8.1.2 Service primitives

Parameter name	Request	Indication	Response	Confirm
Invoke Id	M	M(=)	M(=)	M(=)
Interrogation Type	M	M(=)		
GMSC Address	M	M(=)		
MSISDN	M	M(=)		
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(=)		
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(=)
User error			C	C(=)
SS-List			U	C(=)
Provider error				O

Table 8.1/1: MAP_SEND_ROUTING_INFORMATION parameters

8.1.3 Parameter use

See subclause 5.6 for a definition of the parameters used in addition to the following. Note that:

- a conditional parameter whose use is defined only in GSM 03.78 shall be absent if the sending entity does not support CAMEL;
- a conditional parameter whose use is defined only in GSM 03.79 shall be absent if the sending entity does not support optimal routeing;
- a conditional parameter whose use is defined only in GSM 03.78 & GSM 03.79 shall be absent if the sending entity supports neither CAMEL nor optimal routeing.

Interrogation Type

See GSM 03.79 [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.

OR Interrogation

See GSM 03.79 [99] for the use of this parameter and the conditions for its presence.

OR Capability

See GSM 03.79 [99] for the use of this parameter and the conditions for its presence.

CUG Interlock

See GSM 03.18 [97] for the use of this parameter and the conditions for its presence.

CUG Outgoing Access

See GSM 03.18 [97] for the use of this parameter and the conditions for its presence.

Number of Forwarding

See GSM 03.18 [97] for the use of this parameter and the conditions for its presence.

Network Signal Info

See GSM 03.18 [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 GSM 03.78

T-CSI Suppression

The use of this parameter and the requirements for its presence are specified in GSM 03.78

Suppression Of Announcement

The use of this parameter and the requirements for its presence are specified in GSM 03.78

Call Reference Number

The use of this parameter and the conditions for its presence are specified in GSM 03.78 [98] and GSM 03.79 [99].

Forwarding Reason

See GSM 03.79 [99] for the use of this parameter and the conditions for its presence.

Basic Service Group

See GSM 03.79 [99] for the use of this parameter and the conditions for its presence.

IMSI

See GSM 03.18 [97] for the use of this parameter and the conditions for its presence.

MSRN

See GSM 03.18 [97] and GSM 03.79 [99] for the use of this parameter and the conditions for its presence.

Forwarding Data

This parameter includes the forwarded-to number, the forwarding option Notification to calling party and the forwarding reason, and can include the forwarded-to subaddress. See GSM 03.18 [97] and GSM 03.79 [99] for the conditions for the presence of its components.

Forwarding Interrogation Required

See GSM 03.79 [99] for the use of this parameter and the conditions for its presence.

VMSC address

See GSM 03.79 [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 GSM 03.78

Location Information

The use of this parameter and the requirements for its presence are specified in GSM 03.78

Subscriber State

The use of this parameter and the requirements for its presence are specified in GSM 03.78

CUG Subscription Flag

The use of this parameter and the requirements for its presence are specified in GSM 03.78.

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 GSM 03.78.

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.

User error

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

- Unknown Subscriber;
- Number changed;
- Call Barred;
This error will indicate that either incoming calls are barred for this MS or that calls are barred due to Operator Determined Barring (see GSM 02.41 for a definition of this network feature).
- CUG Reject;
The value of this error cause will indicate the reason for CUG Reject.
- Bearer Service Not Provisioned;
- Teleservice Not Provisioned;
A subscription check has been performed and the call has not passed the check due to incompatibility with regard to the requested service. Depending on the nature of the incompatibility, either of these messages will be returned.
- Facility Not Supported;
- Absent Subscriber;
This indicates that the location of the MS is not known (either the station is not registered and there is no location information available or the Provide Roaming Number procedure fails due to IMSI detached flag being set), 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.
- 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 subclause 5.6 for a definition of these errors.

Provider error

These are defined in subclause 5.6.

8.2 MAP_PROVIDE_ROAMING_NUMBER service

8.2.1 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 8.2/1.

8.2.2 Service primitives

Parameter name	Request	Indication	Response	Confirm
Invoke Id	M	M(=)	M(=)	M(=)
IMSI	M	M(=)		
MSC Number	M	M(=)		
MSISDN	U	C(=)		
LMSI	C	C(=)		
GSM Bearer Capability	C	C(=)		
Network Signal Info	C	C(=)		
Suppression Of Announcement	C	C(=)		
Call Reference Number	C	C(=)		
GMSC Address	C	C(=)		
OR Interrogation	C	C(=)		
Roaming Number			C	C(=)
User error			C	C(=)
Provider error				O

Table 8.2/1: MAP_PROVIDE_ROAMING_NUMBER parameters

8.2.3 Parameter use

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

- a conditional parameter whose use is defined only in GSM 03.78 shall be absent if the sending entity does not support CAMEL;
- a conditional parameter whose use is defined only in GSM 03.79 shall be absent if the sending entity does not support optimal routing;
- a conditional parameter whose use is defined only in GSM 03.78 & GSM 03.79 shall be absent if the sending entity supports neither CAMEL nor optimal routing.

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 GSM 03.18 [97] for the use of this parameter and the conditions for its presence.

LMSI

See GSM 03.18 [97] for the use of this parameter and the conditions for its presence.

GSM Bearer Capability

See GSM 03.18 [97] for the use of this parameter and the conditions for its presence.

This information is passed according to the rules specified in TS GSM 09.07.

There may be two GSM Bearer Capabilities supplied.

Network Signal Info

See GSM 03.18 [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 GSM 03.78.

Call Reference Number

The use of this parameter and the conditions for its presence are specified in GSM 03.78 [98] and GSM 03.79 [99].

GMSC Address

The use of this parameter and the conditions for its presence are specified in GSM 03.78 [98] and GSM 03.79 [99].

OR Interrogation

See GSM 03.79 [99] for the use of this parameter and the conditions for its presence.

Roaming Number

See GSM 03.18 [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 routing.
- Facility Not Supported;
- System Failure;
- Data Missing;
- Unexpected Data Value.

See subclause 5.6 for a definition of these reasons.

Provider error

These are defined in subclause 5.6.

8.3 MAP_RESUME_CALL_HANDLING service

8.3.1 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 8.3/1.

8.3.2 Service primitives

Parameter name	Request	Indication	Response	Confirm
Invoke Id	M	M(=)	M(=)	M(=)
Call Reference Number	M	M(=)		
Basic Service Group	M	M(=)		
IMSI	M	M(=)		
Forwarding Data	M	M(=)		
CUG Interlock	C	C(=)		
CUG Outgoing Access	C	C(=)		
O-CSI	C	C(=)		
User error			C	C(=)
Provider error				O

Table 8.3/1: MAP_RESUME_CALL_HANDLING parameters

8.3.3 Parameter use

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

Call Reference Number

See GSM 03.79 [99] for the use of this parameter.

Basic Service Group

See GSM 03.79 [99] for the use of this parameter.

IMSI

This is the IMSI of the forwarding Subscriber.

Forwarding Data

Includes the forwarded-to number, the forwarding reason, an indication of whether the calling party is to be notified that the call has been forwarded and possibly a forwarded-to subaddress.

CUG Interlock

See GSM 03.79 [99] for the use of this parameter and the conditions for its presence.

CUG Outgoing Access

See GSM 03.79 [99] for the use of this parameter and the conditions for its presence.

O-CSI

See GSM 03.79 [99] for the use of this parameter and the conditions for its presence.

For CAMEL phase 1, the O-CSI shall contain only one set of O-BCSM TDP data.

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.

Provider error

These are defined in subclause 5.6.

9 Supplementary services related services

9.1 MAP_REGISTER_SS service

9.1.1 Definition

This service is used between the MSC and the VLR and between the VLR and the HLR to register data related to a supplementary service. The VLR will relay the message to the HLR.

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

9.1.2 Service primitives

The service primitives are shown in table 9.1/1.

Parameter name	Request	Indication	Response	Confirm
Invoke id	M	M(=)	M(=)	M(=)
SS-Code	M	M(=)		
Basic service	C	C(=)		
Forwarded-to number with subaddress	C	C(=)		
No reply condition time	C	C(=)		
Forwarding information			C	C(=)
User error			C	C(=)
Provider error				O

Table 9.1/1: MAP_REGISTER_SS parameters

9.1.3 Parameter use

Invoke id

See subclause 5.6.1 for the use of this parameter.

SS-Code

This parameter indicates the supplementary service which the mobile subscriber wants to register.

Basic service

This parameter indicates for which basic service group the supplementary service is to be registered. If it is not included, the registration request applies to all basic services.

Forwarded-to number with subaddress

This parameter is obligatory if the registration applies to one or more call forwarding supplementary services. It can optionally include a sub-address.

No reply condition time

This parameter is included if the registration applies to the Call Forwarding on No Reply supplementary service (or a superset of this service) and the mobile subscriber supplies a value for this time.

Forwarding information

This parameter is returned by the responder at successful outcome of the service, if the registration request concerned one or a group of Call Forwarding supplementary services.

User error

This parameter is sent by the responder upon unsuccessful outcome of the service, and then takes one of the following values defined in subclause 5.6.1:

- System failure;
- Data missing;
- Unexpected data value;
- Call Barred;
- Bearer service not provisioned;
This error is returned only if not even a subset of the requested bearer service group has been subscribed to.
- Teleservice not provisioned;
This error is returned only if not even a subset of the requested teleservice group has been subscribed to.
- Illegal SS operation;
- SS error status;
- SS incompatibility.

Provider error

See subclause 5.6.1 for the use of this parameter.

9.2 MAP_ERASE_SS service

9.2.1 Definition

This service is used between the MSC and the VLR and between the VLR and the HLR to erase data related to a supplementary service. The VLR will relay the message to the HLR.

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

9.2.2 Service primitives

The service primitives are shown in table 9.2/1.

Parameter name	Request	Indication	Response	Confirm
Invoke id	M	M(=)	M(=)	M(=)
SS-Code	M	M(=)		
Basic service	C	C(=)		
Forwarding information			C	C(=)
User error			C	C(=)
Provider error				O

Table 9.2/1: MAP_ERASE_SS parameters

9.2.3 Parameter use

Invoke id

See subclause 5.6.1 for the use of this parameter.

SS-Code

This parameter indicates the supplementary service which the mobile subscriber wants to erase.

Basic service

This parameter indicates for which basic service group the supplementary service should be erased. If it is not included, the erasure request applies to all basic services.

Forwarding information

This parameter is returned by the responder at successful outcome of the service, if the erasure request concerned one or a group of Call Forwarding supplementary services.

User error

This parameter is sent by the responder upon unsuccessful outcome of the service, and then takes one of the following values, defined in subclause 5.6.1:

- System failure;
- Data Missing;
- Unexpected data value;
- Bearer service not provisioned;
This error is returned only if not even a subset of the requested bearer service group has been subscribed to.
- Teleservice not provisioned;
This error is returned only if not even a subset of the requested teleservice group has been subscribed to.
- Call Barred;
- Illegal SS operation;
- SS error status.

Provider error

See subclause 5.6.1 for the use of this parameter.

9.3 MAP_ACTIVATE_SS service

9.3.1 Definition

This service is used between the MSC and the VLR and between the VLR and the HLR to activate a supplementary service. The VLR will relay the message to the HLR.

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

9.3.2 Service primitives

The service primitives are shown in table 9.3/1.

Parameter name	Request	Indication	Response	Confirm
Invoke id	M	M(=)	M(=)	M(=)
SS-Code	M	M(=)		
Basic service	C	C(=)		
Forwarding information			C	C(=)
Call barring information			C	C(=)
SS-Data			C	C(=)
User error			C	C(=)
Provider error				O

Table 9.3/1: MAP_ACTIVATE_SS parameters

9.3.3 Parameter use

Invoke id

See subclause 5.6.1 for the use of this parameter.

SS-Code

This parameter indicates the supplementary service which the mobile subscriber wants to activate.

Basic service

This parameter indicates for which basic service groups the requested supplementary service(s) should be activated. If it is not included, the activation request applies to all basic services.

Forwarding information

This parameter is returned by the responder at successful outcome of the service, if the activation request concerned Call Forwarding.

Call barring information

This parameter is returned by the responder at successful outcome of the service, if the activation request concerned Call Barring.

SS-Data

This parameter is returned by the responder at successful outcome of the service, if the activation request concerned for example Call Waiting.

User error

This parameter is sent by the responder upon unsuccessful outcome of the service, and then takes one of the following values, defined in subclause 5.6.1:

- System failure;
- Data Missing;
- Unexpected data value;
- Bearer service not provisioned;
This error is returned only if not even a subset of the requested bearer service group has been subscribed to.
- Teleservice not provisioned;
This error is returned only if not even a subset of the requested teleservice group has been subscribed to.
- Call Barred;
- Illegal SS operation;
- SS error status;
- SS subscription violation;
- SS incompatibility;
- Negative PW check;
- Number Of PW Attempts Violation.

Provider error

See subclause 5.6.1 for the use of this parameter.

9.4 MAP_DEACTIVATE_SS service

9.4.1 Definitions

This service is used between the MSC and the VLR and between the VLR and the HLR to deactivate a supplementary service. The VLR will relay the message to the HLR.

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

9.4.2 Service primitives

The service primitives are shown in table 9.4/1.

Parameter name	Request	Indication	Response	Confirm
Invoke id	M	M(=)	M(=)	M(=)
SS-Code	M	M(=)		
Basic service	C	C(=)		
Forwarding information			C	C(=)
Call barring information			C	C(=)
SS-Data			C	C(=)
User error			C	C(=)
Provider error				O

Table 9.4/1: MAP_DEACTIVATE_SS parameters

9.4.3 Parameter use

Invoke id

See subclause 5.6.1 for the use of this parameter.

SS-Code

This parameter indicates the supplementary service which the mobile subscriber wants to deactivate.

Basic service

This parameter indicates for which basic service group the requested supplementary service(s) should be deactivated. If it is not included the deactivation request applies to all basic services.

Forwarding information

This parameter is returned by the responder at successful outcome of the service, if the deactivation request concerned one or a group of Call Forwarding supplementary services.

Call barring information

This parameter is returned by the responder at successful outcome of the service, if the activation request concerned one or a group of Call Barring supplementary services.

SS-Data

This parameter is returned by the responder at successful outcome of the service, for example if the deactivation request concerned the Call Waiting supplementary service.

User error

This parameter is sent by the responder upon unsuccessful outcome of the service, and then takes one of the following values, defined in subclause 5.6.1:

- System failure;
- Data Missing;
- Unexpected data value;
- Bearer service not provisioned;
This error is returned only if not even a subset of the requested bearer service group has been subscribed to.

- Teleservice not provisioned;
This error is returned only if not even a subset of the requested teleservice group has been subscribed to.
- Call Barred;
- Illegal SS operation;
- SS error status;
- SS subscription violation;
- Negative PW check;
- Number Of PW Attempts Violation.

Provider error

See subclause 5.6.1 for the use of this parameter.

9.5 MAP_INTERROGATE_SS service

9.5.1 Definitions

This service is used between the MSC and the VLR and between the VLR and the HLR to retrieve information related to a supplementary service. The VLR will relay the message to the HLR if necessary.

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

9.5.2 Service primitives

The service primitives are shown in table 9.5/1.

Parameter name	Request	Indication	Response	Confirm
Invoke id	M	M(=)	M(=)	M(=)
SS-Code	M	M(=)		
Basic service	C	C(=)		
SS-Status			C	C(=)
Basic service Group LIST			C	C(=)
Forwarding feature LIST			C	C(=)
CLI restriction Info			C	C(=)
User error			C	C(=)
Provider error				O

Table 9.5/1: MAP_INTERROGATE_SS parameters

9.5.3 Parameter use

For additional information on parameter use refer to the GSM 04.8x and 04.9x-series of technical specifications.

Invoke id

See subclause 5.6.1 for the use of this parameter.

SS-Code

The mobile subscriber can only interrogate a single supplementary service per service request.

Basic service

This parameter indicates for which basic service group the given supplementary service is interrogated. If it is not included, the interrogation request applies to all basic services.

SS-Status

This parameter is included by the responder if:

- the interrogated supplementary service can only be subscribed for all applicable basic services simultaneously; or
- the interrogated supplementary service is not active for any of the interrogated basic services.

Basic service group LIST

This parameter LIST is used to include one or a series of basic service groups for which the interrogated supplementary service is active. If the interrogated supplementary service is not active for any of the interrogated (and provisioned) basic service groups, the SS-Status parameter is returned.

Forwarding feature LIST

The forwarding feature parameter is described in subclause 5.6.4. A list of one or more forwarding features is returned by the responder when the interrogation request applied to Call Forwarding supplementary service.

If no basic service code parameter is provided within this sequence, the forwarding feature parameter applies to all provisioned basic services.

CLI restriction Info

The CLI-RestrictionInfo parameter is returned by the responder when the interrogation request applies to the CLIR supplementary service.

User error

This error is sent by the responder upon unsuccessful outcome of the interrogation service, and then takes one of the following values, defined in subclause 5.6.1:

- System failure;
- Data Missing;
- Unexpected data value;
- Bearer Service not provisioned;
This error is returned only if not even a subset of the interrogated bearer services are provided.
- Teleservice not provisioned;
This error is returned only if not even a subset of the interrogated teleservices are provided.
- Call Barred;
- Illegal SS operation;
- SS not available.

Provider error

See subclause 5.6.1 for the use of this parameter.

9.6 MAP_INVOKE_SS service

9.6.1 Definitions

This service is used between the MSC and the VLR to check the subscriber's subscription to a given supplementary service in the VLR, in connection with in-call invocation of that supplementary service, i.e. after the call set-up phase is finished. For supplementary service invocation during call set-up phase, please refer to the call handling descriptions.

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

9.6.2 Service primitives

The service primitives are shown in table 9.6/1.

Parameter name	Request	Indication	Response	Confirm
Invoke id	M	M(=)	M(=)	M(=)
SS-Code	M	M(=)		
Basic service	C	C(=)		
User error			C	C(=)
Provider error				O

Table 9.6/1: MAP_INVOKE_SS parameters

9.6.3 Parameter use

Invoke id

See subclause 5.6.1 for the use of this parameter.

SS-Code

This SS-Code can only refer to a single supplementary service, e.g. the Call Hold or Multi Party supplementary services.

Basic service

This parameter indicates for which basic service the supplementary service invocation is required.

User error

This parameter is sent by the responder upon unsuccessful outcome of the service, and then takes one of the following values:

- System Failure;
- Data Missing;
- Unexpected data value;
- Call Barred;
- Illegal SS operation;
- SS error status;
- SS not available.

Provider error

See subclause 5.6.1 for the use of this parameter.

9.7 MAP_REGISTER_PASSWORD service

9.7.1 Definitions

This service is used between the MSC and the VLR and between the VLR and the HLR if the mobile subscriber requests to register a new password. The VLR will relay the message to the HLR.

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

9.7.2 Service primitives

The service primitives are shown in table 9.7/1.

Parameter name	Request	Indication	Response	Confirm
Invoke id	M	M(=)	M(=)	M(=)
SS-Code	M	M(=)		
New password			C	C(=)
User error			C	C(=)
Provider error				O

Table 9.7/1: MAP_REGISTER_PASSWORD parameters

9.7.3 Parameter use

Invoke id

See subclause 5.6.1 for the use of this parameter.

SS-Code

This parameter indicates for which supplementary service(s) the password should be registered.

New Password

See subclause 5.6.4 for the use of this parameter.

User error

This parameter is sent by the responder upon unsuccessful outcome of the service, and then takes one of the following values, defined in subclause 5.6.1:

- System failure;
- Data Missing;
- Unexpected data value;
- Call Barred;
- SS subscription violation;
- Password registration failure;
- Negative PW check;
- Number Of PW Attempts Violation.

Provider error

See subclause 5.6.1 for the use of this parameter.

9.8 MAP_GET_PASSWORD service

9.8.1 Definitions

This service is used between the HLR and the VLR and between the VLR and the MSC when the HLR receives a request from the mobile subscriber for an operation on a supplementary service which requires a password from the subscriber. The VLR will relay the message to the MSC.

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

9.8.2 Service primitives

The service primitives are shown in table 9.8/1.

Parameter name	Request	Indication	Response	Confirm
Invoke id	M	M(=)	M(=)	M(=)
Linked id	C	C(=)		
Guidance info	M	M(=)		
Current password			M	M(=)
Provider error				O

Table 9.8/1: MAP_GET_PASSWORD parameters

9.8.3 Parameter use

Invoke id

See subclause 5.6.1 for the use of this parameter.

Linked Id

See subclause 5.6.1 for the use of this parameter. If the MAP GET PASSWORD service is used in conjunction with the MAP REGISTER PASSWORD service, this parameter must be present; otherwise it must be absent.

Guidance info

See subclause 5.6.4 for the use of this parameter.

Current password

See subclause 5.6.4 for the use of this parameter.

Provider error

See subclause 5.6.1 for the use of this parameter.

9.9 MAP_PROCESS_UNSTRUCTURED_SS_REQUEST service

9.9.1 Definitions

This service is used between the MSC and the VLR and between the VLR and the HLR to relay information in order to allow unstructured supplementary service operation.

The MAP_PROCESS_UNSTRUCTURED_SS_REQUEST service is a confirmed service using the primitives from table 9.9/1.

9.9.2 Service primitives

Parameter name	Request	Indication	Response	Confirm
Invoke id	M	M(=)	M(=)	M(=)
USSD Data Coding Scheme	M	M(=)	C	C(=)
USSD String	M	M(=)	C	C(=)
User error			C	C(=)
Provider error				O

Table 9.9/1: MAP_PROCESS_UNSTRUCTURED_SS_REQUEST parameters

9.9.3 Parameter use

Invoke id

See subclause 5.6.1 for the use of this parameter.

USSD Data Coding Scheme:

See subclause 5.6.4 for the use of this parameter. The presence of the parameter in the response is dependent on the unstructured supplementary service application. If this parameter is present, then the USSD String parameter has to be present.

USSD String:

See subclause 5.6.1 for the use of this parameter. The presence of the parameter in the response is dependent on the unstructured supplementary service application. If this parameter is present, then the USSD Data Coding Scheme parameter has to be present.

User error

This parameter is sent by the responder upon unsuccessful outcome of the service, and then takes one of the following values defined in subclause 5.6.1:

- System failure;
- Data missing;
- Unexpected data value;
This error is returned by the responder if it is not able to deal with the contents of the USSD string.
- Call Barred;
- Unknown Alphabet.

Provider error

See subclause 5.6.1 for the use of this parameter.

9.10 MAP_UNSTRUCTURED_SS_REQUEST service

9.10.1 Definitions

This service is used between the HLR and the VLR and between the VLR and the MSC when the invoking entity requires information from the mobile user, in connection with unstructured supplementary service handling.

The MAP_UNSTRUCTURED_SS_REQUEST service is a confirmed service using the primitives from table 9.10/1.

9.10.2 Service primitives

The service primitives are shown in table 9.10/1.

Parameter name	Request	Indication	Response	Confirm
Invoke id	M	M(=)	M(=)	M(=)
USSD Data Coding Scheme	M	M(=)	C	C(=)
USSD String	M	M(=)	C	C(=)
User error			C	C(=)
Provider error				O

Table 9.10/1: MAP_UNSTRUCTURED_SS_REQUEST parameters

9.10.3 Parameter use

Invoke id

See subclause 5.6.1 for the use of this parameter.

USSD Data Coding Scheme:

See subclause 5.6.4 for the use of this parameter. The presence of the parameter in the response is dependent on the mobile user's MMI input. If this parameter is present, then the USSD String parameter has to be present.

USSD String:

See subclause 5.6.1 for the use of this parameter. The presence of the parameter in the response is dependent on the mobile user's MMI input. If this parameter is present, then the USSD Data Coding Scheme parameter has to be present.

User error

This parameter is sent by the responder upon unsuccessful outcome of the service, and then takes one of the following values defined in subclause 5.6.1:

- System failure;
- Data missing;
- Unexpected data value;
This error is returned by the responder if it is not able to deal with the contents of the USSD string.
- 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 subclause 5.6.1 for the use of this parameter.

9.11 MAP_UNSTRUCTURED_SS_NOTIFY service

9.11.1 Definitions

This service is used between the HLR and the VLR and between the VLR and the MSC when the invoking entity requires a notification to be sent to the mobile user, in connection with unstructured supplementary services handling.

The MAP_UNSTRUCTURED_SS_NOTIFY service is a confirmed service using the primitives from table 9.11/1.

9.11.2 Service primitives

The service primitives are shown in table 9.11/1.

Parameter name	Request	Indication	Response	Confirm
Invoke id	M	M(=)	M(=)	M(=)
USSD Data Coding Scheme	M	M(=)		
USSD String	M	M(=)		
User error			C	C(=)
Provider error				O

Table 9.11/1: MAP_UNSTRUCTURED_SS_NOTIFY parameters

9.11.3 Parameter use

Invoke id

See subclause 5.6.1 for the use of this parameter.

USSD Data Coding Scheme:

See subclause 5.6.4 for the use of this parameter.

USSD String:

See subclause 5.6.1 for the use of this parameter.

User error

This parameter is sent by the responder upon unsuccessful outcome of the service, and then takes one of the following values defined in subclause 5.6.1:

- System failure;
- Data missing;
- Unexpected data value;
This error is returned by the responder if it is not able to deal with the contents of the USSD string.
- 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 subclause 5.6.1 for the use of this parameter.

10 Short message service management services

10.1 MAP-SEND-ROUTING-INFO-FOR-SM service

10.1.1 Definition

This service is used between the gateway MSC and the HLR to retrieve the routing information needed for routing the short message to the servicing MSC.

The MAP-SEND-ROUTING-INFO-FOR-SM is a confirmed service using the primitives from table 10.1/1.

10.1.2 Service primitives

The service primitives are shown in table 10.1/1.

Parameter name	Request	Indication	Response	Confirm
Invoke Id	M	M(=)	M(=)	M(=)
MSISDN	M	M(=)		
SM-RP-PRI	M	M(=)		
Service Centre Address	M	M(=)		
IMSI			C	C(=)
MSC Number			C	C(=)
LMSI			C	C(=)
User error			C	C(=)
Provider error				O

Table 10.1/1: MAP-SEND-ROUTING-INFO-FOR-SM

10.1.3 Parameter use

Invoke id:

See definition in subclause 5.6.1.

MSISDN:

See definition in subclause 5.6.2.

SM-RP-PRI:

See definition in subclause 5.6.8.

Service Centre Address:

See definition in subclause 5.6.2.

IMSI:

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

MSC Number:

See definition in subclause 5.6.2. This parameter is provided in a successful response.

LMSI:

See definition in subclause 5.6.2. It is an operator option to provide this parameter from the VLR; it is mandatory for the HLR to include the LMSI in a successful response, if the VLR has used the LMSI.

User error:

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

- Unknown subscriber;
- Call Barred;
- Teleservice Not Provisioned;
- Absent Subscriber;
- Facility Not Supported;
- System failure;
- Unexpected Data Value;
- Data missing.

Provider error:

For definition of provider errors see subclause 5.6.1.

10.2 MAP-MO-FORWARD-SHORT-MESSAGE service

10.2.1 Definition

This service is used between the serving MSC and the gateway MSC to forward mobile originated short messages.

The MAP-MO-FORWARD-SHORT-MESSAGE service is a confirmed service using the service primitives given in table 10.2/1.

10.2.2 Service primitives

The service primitives are shown in table 10.2/1.

Parameter name	Request	Indication	Response	Confirm
Invoke Id	M	M(=)	M(=)	M(=)
SM RP DA	M	M(=)		
SM RP OA	M	M(=)		
SM RP UI	M	M(=)	C	C(=)
User error			C	C(=)
Provider error				O

Table 10.2/1: MAP-MO-FORWARD-SHORT-MESSAGE

10.2.3 Parameter use

Invoke id:

See definition in subclause 5.6.1.

SM RP DA:

See definition in subclause 5.6.8.

In the mobile originated SM transfer this parameter contains the Service Centre address received from the mobile station.

SM RP OA:

See definition in subclause 5.6.8.

The MSISDN received from the VLR is inserted in this parameter in the mobile originated SM transfer.

SM RP UI:

See definition in subclause 5.6.8. The short message transfer protocol data unit received from the Service Centre is inserted in this parameter.

User error:

The following errors defined in subclause 5.6.1 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 subclause 5.6.1.

10.3 MAP-REPORT-SM-DELIVERY-STATUS service**10.3.1 Definition**

This service is used between the gateway MSC and the HLR. The MAP-REPORT-SM-DELIVERY-STATUS service is used to set the Message Waiting Data into the HLR or to inform the HLR of successful SM transfer after polling. This service is invoked by the gateway MSC.

The MAP-REPORT-SM-DELIVERY-STATUS service is a confirmed service using the service primitives given in table 10.3/1.

10.3.2 Service primitives

The service primitives are shown in table 10.3/1.

Parameter name	Request	Indication	Response	Confirm
Invoke Id	M	M(=)	M(=)	M(=)
MSISDN	M	M(=)		
Service Centre Address	M	M(=)		
SM Delivery Outcome	M	M(=)		
Absent Subscriber	C	C(=)		
Diagnostic SM				
MSISdn-Alert			C	C(=)
User error			C	C(=)
Provider error				O

Table 10.3/1: MAP-REPORT-SM-DELIVERY-STATUS

10.3.3 Parameter use**Invoke id:**

See definition in subclause 5.6.1.

MSISDN:

See definition in subclause 5.6.2.

Service Centre Address:

See definition in subclause 5.6.2.

SM Delivery Outcome:

See definition in subclause 5.6.8. This parameter indicates the status of the mobile terminated SM delivery.

Absent Subscriber Diagnostic SM:

See definition in subclause 5.6.8.

MSIsdn-Alert:

See definition in subclause 5.6.2. 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 subclause 5.6.1 may be used, depending on the nature of the fault:

- Unknown Subscriber;
- Message Waiting List Full;
- Unexpected Data Value;
- Data missing.

Provider error:

For definition of provider errors see subclause 5.6.1.

10.4 MAP-READY-FOR-SM service

10.4.1 Definition

This service is used between the MSC and VLR and as well between the VLR and the HLR. The MSC initiates this service if a subscriber indicates memory available situation. The VLR uses the service to indicate this to the HLR.

The VLR initiates this service if a subscriber, whose message waiting flag is active in the VLR, has radio contact in the MSC.

The MAP-READY-FOR-SM service is a confirmed service using the primitives from table 10.4/1.

10.4.2 Service primitives

The service primitives are shown in table 10.4/1.

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

Table 10.4/1: MAP-READY-FOR-SM

10.4.3 Parameter use

Invoke id:

See definition in subclause 5.6.1.

IMSI:

See definition in subclause 5.6.2. The IMSI is used always between the VLR and the HLR. Between the MSC and the VLR the identification can be either IMSI or TMSI.

TMSI:

See definition in subclause 5.6.2. The identification can be either IMSI or TMSI between MSC and VLR.

Alert Reason:

See definition in subclause 5.6.8. This parameter indicates if the mobile subscriber is present or the MS has memory available.

User error:

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

- Unknown Subscriber;
- Facility Not Supported;
- System Failure;
- Unexpected Data Value;
- Data missing;

Provider error:

For definition of provider errors see subclause 5.6.1.

10.5 MAP-ALERT-SERVICE-CENTRE service

10.5.1 Definition

This service is used between the HLR and the interworking MSC. The HLR initiates this service, if the HLR detects that a subscriber, whose MSISDN is in the Message Waiting Data file, is active or the MS has memory available.

The MAP-ALERT-SERVICE-CENTRE service is a confirmed service using the primitives from table 10.5/1.

10.5.2 Service primitives

The service primitives are shown in table 10.5/1.

Parameter name	Request	Indication	Response	Confirm
Invoke Id	M	M(=)	M(=)	M(=)
MSIsdn-Alert	M	M(=)		
Service Centre Address	M	M(=)		
User error			C	C(=)
Provider error				O

Table 10.5/1: MAP-ALERT-SERVICE-CENTRE

10.5.3 Parameter use

Invoke id:

See definition in subclause 5.6.1.

MSIsdn-Alert:

See definition in subclause 5.6.2. The provided MSISDN shall be the one which is stored in the Message Waiting Data file.

Service Centre Address:

See definition in subclause 5.6.2.

User error:

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

- System Failure;
- Unexpected Data Value;
- Data missing.

Provider error:

For definition of provider errors see subclause 5.6.1.

10.6 MAP-INFORM-SERVICE-CENTRE service

10.6.1 Definition

This service is used between the HLR and the gateway MSC to inform the Service Centre which MSISDN number is stored in the Message Waiting Data file. If the stored MSISDN number is not the same than the one received from the gateway MSC in the MAP-SEND-ROUTING-INFO-FOR-SM service primitive the stored MSISDN number is included in the message.

Additionally the status of MCEF and MNRF flags and the inclusion of the particular Service Centre address in the Message Waiting Data list is informed to the gateway MSC when appropriate.

The MAP-INFORM-SERVICE-CENTRE service is a non-confirmed service using the primitives from table 10.6/1.

10.6.2 Service primitives

The service primitives are shown in table 10.6/1.

Parameter name	Request	Indication
Invoke Id	M	M(=)
MSIsdn-Alert	C	C(=)
MWD Status	C	C(=)

Table 10.6/1: MAP-INFORM-SERVICE-CENTRE**10.6.3 Parameter use****Invoke id:**

See definition in subclause 5.6.1.

MSIsdn-Alert:

See definition in subclause 5.6.2 This parameter refers to the MSISDN stored in a Message Waiting Data file in the HLR.

MWD Status:

See definition in subclause 5.6.8. This parameter indicates the status of the MCEF and MNRF flags and the status of the particular SC address presence in the Message Waiting Data list.

10.7 MAP-SEND-INFO-FOR-MT-SMS service**10.7.1 Definition**

This service is used between the MSC and the VLR. The service is invoked by the MSC receiving an mobile terminated short message to request subscriber related information from the VLR.

The MAP-SEND-INFO-FOR-MT-SMS service is a confirmed service using the primitives from table 10.7/1.

10.7.2 Service primitives

The service primitives are shown in table 10.7/1.

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

Table 10.7/1: MAP-SEND-INFO-FOR-MT-SMS**10.7.3 Parameter use****Invoke id:**

See definition in subclause 5.6.1.

SM RP DA:

See definition in subclause 5.6.8. This parameter shall contain either an IMSI or a LMSI.

MSISDN:

See definition in subclause 5.6.2.

User error:

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

- Unknown subscriber;
- Unidentified Subscriber;
- Absent subscriber;
- Unexpected Data Value;
- Data Missing;
- Illegal subscriber;
- Illegal equipment;
- Subscriber busy for MT SMS;
- System Failure.

Provider error:

For definition of provider errors see subclause 5.6.1.

10.8 MAP-SEND-INFO-FOR-MO-SMS service

10.8.1 Definition

This service is used between the MSC and the VLR. The service is invoked by the MSC which has to handle a mobile originated short message request to request the subscriber related information from the VLR.

The MAP-SEND-INFO-FOR-MO-SMS service is a confirmed service using the primitives from table 10.8/1.

10.8.2 Service primitives

The service primitives are shown in table 10.8/1.

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

Table 10.8/1: MAP-SEND-INFO-FOR-MO-SMS

10.8.3 Parameter use

Invoke id:

See definition in subclause 5.6.1.

Service Centre Address:

See definition in subclause 5.6.2.

MSISDN:

See definition in subclause 5.6.2.

User error:

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

- Teleservice Not Provisioned;
- Call Barred;
- Unexpected Data Value;
- Data Missing.

Provider error:

For definition of provider errors see subclause 5.6.1.

10.9 MAP-MT-FORWARD-SHORT-MESSAGE service**10.9.1 Definition**

This service is used between the gateway MSC and the servicing MSC to forward mobile mobile terminated short messages.

The MAP-MT-FORWARD-SHORT-MESSAGE service is a confirmed service using the service primitives given in table 10.9/1.

10.9.2 Service primitives

The service primitives are shown in table 10.9/1.

Parameter name	Request	Indication	Response	Confirm
Invoke Id	M	M(=)	M(=)	M(=)
SM RP DA	M	M(=)		
SM RP OA	M	M(=)		
SM RP UI	M	M(=)	C	C(=)
More Messages To Send	C	C(=)		
User error			C	C(=)
Provider error				O

Table 10.9/1: MAP-MT-FORWARD-SHORT-MESSAGE

10.9.3 Parameter use**Invoke id:**

See definition in subclause 5.6.1.

SM RP DA:

See definition in subclause 5.6.8. This parameter can contain either an IMSI or a LMSI. 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 subclause 5.6.8. 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 subclause 5.6.8. 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 acknowledgment from the MSC to the Service Centre.

More Messages To Send:

See definition in subclause 5.6.8. The information from the MMS indication received from the Service Centre is inserted in this parameter.

User error:

The following errors defined in subclause 5.6.1 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 subclause 5.6.1.

11 General

11.1 Overview

Clause 11 to 14 specify the protocol elements to be used to provide the MAP services described in clause 5.

Clause 12 specifies the elements of procedures for the MAP protocol. Clause 13 specifies the mapping on to TC service primitives. Clause 14 specifies the application contexts, operation packages and abstract syntaxes for the MAP protocol as well as the encoding rules to be applied.

11.2 Underlying services

The MAP protocol relies on the services provided by the Transaction Capabilities (TC) of signalling system number 7, as referenced in clause 4.

11.3 Model

The MAP Protocol Machine (MAP PM) can be modelled as a collection of service state machines (SSMs) - one per MAP specific service invoked - coordinated by a MAP dialogue control function with its one state machine: MAP dialogue state machine (DSM). There are two types of Service State Machines: Requesting Service State Machines (RSM) and Performing Service State Machines (PSM).

A new invocation of a MAP PM is employed on the receipt of a MAP-OPEN request primitive or a TC-BEGIN indication primitive. Each invocation controls exactly one MAP dialogue. For each MAP specific service invoked during a dialogue, a MAP RSM is created at the requestor's side and a MAP PSM is created at the performer's side.

This modelling is used only to facilitate understanding and the MAP behaviour descriptions and is not intended to suggest any implementation. SDL descriptions are organized according to this model.

How the MAP-service-user and the MAP refer to a MAP dialogue (i.e. a MAP PM invocation) is a local implementation matter.

How TC dialogue identifiers are assigned to a MAP PM invocation is also a local implementation matter.

11.4 Conventions

The behaviour of the MAP PM depends on the application-context-name associated with the dialogue. One major difference is that the MAP requests the transfer of the application-context-name by TC only for those contexts which do not belong to the so-called "version one context set".

The "version one context set" is a set of application-contexts which model the behaviour of a MAP V1 implementation according to the latest phase 1 version of GSM 09.02. This set is defined in clause 12.

The procedures described in clause 12 are used when the application-context-name does not refer to a dialogue between an MSC and its VLR. When the application-context-name refers to a dialogue between an MSC and its VLR the MAP PM procedures are a local implementation matter.

12 Elements of procedure

12.1 Dialogue establishment

The establishment of a MAP dialogue involves two MAP-service-users, one that is the dialogue-initiator and one that is the dialogue-responder.

This procedure is driven by the following signals:

- a MAP-OPEN request primitive from the dialogue-initiator;
- a TC-BEGIN indication primitive occurring at the responding side;
- a MAP-OPEN response primitive from the dialogue-responder;
- the first TC-CONTINUE indication primitive occurring at the initiating side;

and under specific conditions:

- a TC-END indication primitive occurring at the initiating side;
- a TC-U-ABORT indication primitive occurring at the initiating side;
- a TC-P-ABORT indication primitive occurring at the initiating side.

12.1.1 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 16-21 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.

12.1.2 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 subclause 12.6 for a description of the associated SSMS.

On receipt of the MAP-DELIMITER request primitive the MAP PM shall issue a TC-BEGIN request primitive. 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.

12.1.3 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 12.1/1 (note 1).

NOTE 1: In some cases, it may be necessary to analyse the operation argument.

Then:

- a) if no application-context-name can be derived (i.e. the operation code does not exist in MAP V1 specifications), the MAP PM shall issue a TC-U-ABORT request primitive (note 2). The local MAP-User is not informed.

- b) if an application-context-name can be derived and if it is acceptable from a load control point of view, the MAP PM shall:
- i) if this primitive requests the beginSubscriberActivity operation, the MAP PM shall check whether more components have been received associated with this operation. If more components are present, the MAP PM shall issue a MAP-OPEN indication primitive with the version 1 application-context-name "networkFunctionalSsContext-v1". The Destination-reference shall include the IMSI taken from the argument of the beginSubscriberActivity operation; the Originating-reference shall cover the originatingEntityNumber.

A beginSubscriberActivity operation that is not associated with any other Component shall be rejected by the MAP PM by issuing a TC-U-ABORT request primitive (note 2). The local MAP-User shall not be informed.

- ii) otherwise, the MAP PM shall issue a MAP-OPEN indication primitive with the version 1 application-context-name set according to table 12.1/1. DestinationReference and OriginatingReference must not be included in the MAP-OPEN indication primitive.

Then the MAP PM shall function in a way that the dialogue responding MAP behaves as specified in the GSM phase 1 protocol (latest version of TS GSM 09.02 phase 1).

NOTE 2: If no AARQ apdu was included in the BEGIN message, TC (Component Sub-layer) will not include an AARE apdu or an ABRT apdu in a TR-U-ABORT request primitive that is to be issued on receipt of a TC-U-ABORT request primitive from the local MAP service provider.

- c) if an application-context-name can be derived but if it is not acceptable from a load control point of view, the MAP PM shall ignore this dialogue request and not inform the MAP-user;
- if a version 1 application-context-name is included, the MAP PM shall issue a TC-U-ABORT request primitive with abort-reason "User-specific" and user-information "MAP-ProviderAbortInfo" indicating "abnormalDialogue". The local MAP-user shall not be informed.
- if an application-context-name different from version 1 is included in the primitive and if User-information is present, the User-information must constitute a syntactically correct MAP-OPEN dialogue PDU. Otherwise a TC-U-ABORT request primitive with abort-reason "User-specific" and user-information "MAP-ProviderAbortInfo" indicating "abnormalDialogue" shall be issued and the local MAP-user shall not be informed.
- if no User-information is present it is checked whether presence of User Information in the TC-BEGIN indication primitive is required for the received application-context-name. If User Information is required but not present, a TC-U-ABORT request primitive with abort-reason "User-specific" and user-information "MAP-ProviderAbortInfo" indicating "abnormalDialogue" shall be issued. The local MAP-user shall not be informed.
- if an application-context-name different from version 1 is received in a syntactically correct TC-BEGIN indication primitive but is not acceptable from a load control point of view, the MAP PM shall ignore this dialogue request. The MAP-user is not informed.
- if an application-context-name different from version 1 is received in a syntactically correct TC-BEGIN indication primitive and if it is acceptable from a load control point of view, the MAP PM shall check whether the application-context-name is supported.

NOTE 3: Unknown application-context-names are treated like unsupported ones.

If it is, the MAP PM shall issue a MAP-OPEN indication primitive with all parameters (application-context-name included) set according to the value of the corresponding parameter of the TC-BEGIN indication primitive.

The MAP PM shall then process any other indication primitives received from TC as described in subclause 12.6. Once all the received components have been processed, the MAP PM shall inform the local MAP service user by a MAP-DELIMITER indication primitive.

If the TC-BEGIN indication primitive is not associated with any component, the MAP PM shall inform the MAP User by a MAP-DELIMITER indication primitive.

Once all the received primitives have been processed, the MAP PM does not accept any primitive from the provider and waits for a MAP-OPEN response primitive from its user.

- if an application-context-name different from version 1 is received in a syntactically correct TC-BEGIN indication primitive and if it is acceptable from a load control point of view but the application-context-name is not supported, the MAP PM shall issue a TC-U-ABORT request primitive with abort-reason indicating "application-context-not-supported". If an alternative application-context-name cannot be offered, the received application-context-name shall be returned in the TC-U-ABORT Req primitive.

In the following cases an alternative application-context can be offered and its name included in the TC-U-ABORT Req primitive:

- a) if an application-context of version 2 or higher is requested, but only version 1 application-context supported, then the v1 application context shall be returned;
- b) if an application-context of version 3 or higher is requested, but only version 2 application-context supported, then the v2 application context shall be returned.
- 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.

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 14 and allocated to each application-context used for the MAP.

NOTE 2: The choice between the application contexts is based on the parameters received in the operation.

Table 12.1/1: Mapping of V1 operation codes on to application-context-names

12.1.4 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 subclause 12.6. The MAP PM shall then issue a TC-CONTINUE request primitive after it receives the MAP-DELIMITER request primitive if no MAP-CLOSE request primitive has been received, otherwise it shall issue a TC-END request primitive. In both cases the MAP-Accept PDU (if any) is included in the user-information parameter of the TC primitive.

If the dialogue is not associated with a version 1 application context, the MAP PM shall include the application-context-name in the TC primitive.

If no MAP-CLOSE request has been received, the MAP PM waits for a request primitive from its user or an indication primitive from TC.

On receipt of a MAP-OPEN response primitive indicating that the dialogue is not accepted, the MAP PM shall build a MAP-Refuse PDU and request its transfer using the TC-U-ABORT req primitive (abort reason = user specific).

12.1.5 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 subclause 12.6, and then waits for a request primitive from its user or an indication primitive from TC, otherwise it shall issue a TC-U-ABORT request primitive with a MAP-providerAbort PDU indicating "abnormal dialogue" and a MAP-P-ABORT indication primitive with the "provider-reason" parameter indicating "abnormal dialogue".

12.1.6 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 subclause 12.6; then it shall issue a MAP-CLOSE indication primitive and return to idle all state machines associated with the dialogue.

12.1.7 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 subclause 12.4.2).

12.1.8 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 subclause 12.5.2).

12.2 Dialogue continuation

Once established the dialogue is said to be in a continuation phase.

Both MAP users can request the transfer of MAP APDUs until one of them requests the termination of the dialogue.

12.2.1 Sending entity

The MAP PM shall accept any MAP specific service request or response primitives and process them as described in subclause 12.6.

On receipt of a MAP-DELIMITER request primitive, the MAP PM shall issue a TC-CONTINUE request primitive.

12.2.2 Receiving entity

On receipt of a TC-CONTINUE indication primitive the MAP PM shall accept zero, one or several TC component handling indication primitives and process them as described in subclause 12.6.

12.3 Dialogue termination

Both the dialogue-initiator and the dialogue-responder have the ability to request the termination of a dialogue after it has been established.

The dialogue termination procedure is driven by the following events:

- a MAP-CLOSE request primitive;
- a TC-END indication primitive.

12.3.1 Receipt of a MAP-CLOSE request

On receipt of a MAP-CLOSE request primitive, the MAP PM shall issue a TC-END request primitive and, if applicable, return to idle the associated active SSMS. Note that if the release method parameter of the MAP-CLOSE request indicates "normal" the TC-END request primitive will trigger the transmission of components associated with any user specific request or response primitives which may have been issued after the last MAP-DELIMITER request.

12.3.2 Receipt of a TC-END indication

On receipt of a TC-END indication primitive, the MAP shall accept any component handling indication primitives and process them as described in subclause 12.6.

Once all the received primitives have been processed, the MAP PM shall return to idle the associated SSMs and issue a MAP-CLOSE indication primitive.

12.4 User Abort

Both the dialogue-initiator and the dialogue-responder have the ability to abort a dialogue at any time.

The user abort procedure is driven by one of the following events:

- a MAP-U-ABORT request primitive;
- a TC-U-ABORT indication primitive carrying a MAP-user-abort PDU.

12.4.1 MAP-U-ABORT request

On receipt of a MAP-U-ABORT request the MAP PM shall construct a MAP-user-abort PDU from the user-reason and diagnostic parameters and issue a TC-U-ABORT request primitive. All state machines associated with the dialogue are returned to idle.

12.4.2 TC-U-ABORT ind

On receipt of a TC-U-ABORT indication carrying a MAP-user-abort PDU, the MAP PM shall issue a MAP-U-ABORT indication primitive. The user-reason and diagnostic information elements are mapped to the corresponding parameters of the MAP-U-ABORT indication primitive.

All state machines associated with the dialogue are returned to idle.

12.5 Provider Abort

The MAP has the ability to abort a dialogue at both the dialogue-initiator side and the dialogue-responder side.

The provider abort procedure is driven by one of the following events:

- a MAP PM error situation;
- a TC-P-ABORT indication primitive;
- a TC-U-ABORT indication primitive carrying a MAP-abort PDU.

12.5.1 MAP PM error situation

In the case of an abnormal situation detected at the MAP level during an established dialogue, the MAP PM shall:

- issue a MAP-P-ABORT indication primitive with the appropriate value of the provider-reason parameter;
- construct a MAP-abort PDU from the value of these parameters and request its transfer using a TC-U-ABORT request primitive.

12.5.2 TC-P-ABORT ind

On receipt of a TC-P-ABORT indication, the MAP PM shall issue a MAP-P-ABORT indication primitive.

All state machines associated with the dialogue are returned to idle.

12.5.3 TC-U-ABORT ind

On receipt of a TC-U-ABORT indication carrying a MAP-abort PDU, the MAP PM shall issue a MAP-P-ABORT indication primitive, with the appropriate value of the provider-reason parameter. The source parameter shall indicate "MAP-provider".

All state machines associated with the dialogue are returned to idle.

12.6 Procedures for MAP specific services

This subclause describes the MAP procedures for MAP specific services.

These procedures are driven by the following types of events:

- a MAP specific request or a MAP specific MAP response primitive;
- a component handling primitive from TC.

A Service State Machine is activated on receipt of one of the following signals:

- a MAP request primitive, which activates a requesting SSM;
- a TC-INVOKE indication primitive without linked identifier, which activates a responding SSM.

For component handling primitives there are two types of events:

- events which activate a Service State Machine or which can be related to an existing one;
The procedure elements driven by these events are described in subclauses 12.6.1 to 12.6.4.
- events which cannot be related to a Service State Machine.
The procedure elements driven by these events are described in subclause 12.6.5.

12.6.1 Service invocation

The MAP specific procedures are initiated by the MAP request primitives.

On receipt of a MAP request primitive, the MAP PM shall build an operation argument from the parameters received in the request primitive and request the invocation of the associated operation using the TC-INVOKE procedure. If a linked ID parameter is inserted in the primitive this indicates a child service and implies that the operation on which the service is mapped is linked to the operation on which the parent service is mapped.

The mapping of MAP specific services on to remote operations is given in table 13.2/1.

12.6.2 Service invocation receipt

On receipt of a TC-INVOKE indication primitive, the MAP PM shall:

- if the invoke ID is already in use by an active service, request the transfer of a reject component using the TC-U-REJECT request primitive with the appropriate problem code (duplicated invokeID) and issue a MAP-NOTICE indication primitive with a diagnostic parameter set to "abnormal event received from the peer";
- if the operation code does not correspond to an operation supported by the application-context, request the transfer of a reject component using the TC-U-REJECT request primitive, with the appropriate problem code (unrecognized operation), and -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.

12.6.3 Service response

For user confirmed services, the MAP PM shall accept a MAP response primitive and shall:

- if no error indication is included in the primitive and the service maps on to a class 1 or 3 operation, construct a result information element from the parameters received and request its transfer using the TC-RESULT-L service and optionally the TC-RESULT-NL service.

The TC-RESULT-NL services shall be used when the user specific parameters of the response primitives cannot be transferred in a single signalling frame and no segmenting mechanism is available from the underlying layers. The MAP PM shall issue one or several TC-RESULT-NL request primitives followed by a TC-RESULT-L primitive. The user parameters shall be split so that each portion contains sufficient information to construct a value compatible with the type defined for the result of the associated operation.

- if no error indication is included in the primitive and the service response maps on to a class 4 linked operation, construct an operation argument from the parameters received and request its transfer using the TC-INVOKE service for this class 4 linked operation. The operation to be invoked is deduced from the value of the result parameter of the service primitive;
- if an error indication is included in the primitive and the service maps on to a class 1 or 2 operation, either issue a TC-U-REJECT request primitive if the user error parameter indicates "resource limitation" or "initiating release", or construct an error parameter from the parameters received and request its transfer using the TC-U-ERROR request primitive. The error code should be the one associated with the value of the user error parameter of the response primitive.

NOTE: The only user errors that a MAP user can generate in addition to the list of errors attached to the operation which is associated with the service are: resource limitation and initiating release. Any other abnormal situation is detected either by the TC entity or by the MAP entity.

- if an error indication is received and the operation maps on to a class 3 operation, or if no error indication is received but the service maps on to a class 2 operation which has no class 4 linked operation, return the local service state machine to idle without requesting any service from TC.

12.6.4 Receipt of a response

A component handling indication primitive is considered as driving a response for a confirmed service if the invoke ID parameter value matches the one stored for the service, or if the linked ID parameter value matches the one stored for the service and the operation invoked is a class 4 operation. On receipt of a response (except a TC-L-CANCEL indication) for an unconfirmed service the MAP PM shall issue a MAP-NOTICE indication primitive with the appropriate provider error (return result unexpected or return error unexpected).

12.6.4.1 Receipt of a TC-RESULT-NL indication

If the type of the partial result parameter is not compatible with the one defined for the complete result of this operation, request the transfer of a reject component using the TC-U-REJECT request primitive, with the appropriate problem code (mistyped parameter) and issue a confirm primitive with the provider error parameter set to "invalid response received". The MAP PM shall also issue a TC-U-CANCEL request primitive so that all subsequent result components for this operation are discarded by TC.

Otherwise, store the value of the partial result parameter and wait for subsequent TC-RESULT-NL indication primitives until a TC-RESULT-L indication primitive is received.

12.6.4.2 Receipt of a TC-RESULT-L indication

If the type of the result parameter is not the one defined for the result of this operation, request the transfer of a reject component using the TC-U-REJECT request primitive, with the appropriate problem code (mistyped parameter), and issue a confirm primitive with the provider error parameter set to "invalid response received".

If the type of the result parameter is correct but does not contain all the information elements required by the service associated with the invocation, issue a confirm primitive with the provider error parameter set to "invalid response received".

NOTE 1: These checks are only relevant when there is not a one-to-one mapping between a service and an operation.

If the type of the result parameter is correct but contains information elements which are not relevant for the service associated with the invocation are missing, issue a confirm primitive with the provider error parameter set to "invalid response received".

NOTE 2: These checks are only relevant when there is not a one-to-one mapping between a service and an operation.

Otherwise, issue a MAP confirm primitive to the MAP-service-user mapping the result parameter of the TC-RESULT-L primitive on to the MAP specific parameters.

If partial results have been previously received, the value of the partial result parameters shall also be taken into account before performing the three previous checks.

12.6.4.3 Receipt of a TC-U-ERROR indication

If the error code is not defined for the MAP or is not one associated with the operation referred to by the invoke identifier, request the transfer of a reject component using the TC-U-REJECT request primitive, with the appropriate problem code (unrecognized error or unexpected error), and issue a confirm primitive with the provider error parameter set to "invalid response received".

If the type of the error parameter is not the one defined for this error, request the transfer of a reject component using the TC-U-REJECT request primitive, with the appropriate problem code (mistyped parameter), and issue a confirm primitive with the provider error parameter set to "invalid response received".

If the type of the error parameter is correct but does not contain all the information elements required by the service associated with the invocation, issue a confirm primitive with the provider error parameter set to "invalid response received".

NOTE 1: In some cases, it may be necessary to analyse the operation argument.

If the type of the error parameter is correct but its value includes information elements which are not relevant for the service associated with the invocation, issue a confirm primitive with the provider error parameter set to "invalid response received".

NOTE 2: In some cases, it may be necessary to analyse the operation argument.

Otherwise, issue a MAP confirm primitive to the MAP-service-user with the user error parameter set according to the received error code. If applicable the error parameter is mapped to the diagnostic parameter.

12.6.4.4 Receipt of a TC-INVOKE indication

A TC-INVOKE indication primitive is considered as carrying a possible response to a specific service if the linked ID refers to an active specific service and the associated operation is a class 4 operation. Note that the presence of a linked ID parameter in a TC-INVOKE primitive requesting a non class 4 operation indicates a child service whose procedures are the same as the procedures for the parent service.

On receipt of a TC-INVOKE indication confirming an active service, the MAP PM shall:

- if the operation code is not defined for MAP and the dialogue version is at least 3, issue a TC-U-REJECT request primitive with the appropriate problem code (unrecognized 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 (unrecognized operation, linked response unexpected or unexpected linked operation). If the service is confirmed, the MAP shall also issue a Confirm primitive with provider error indication "unexpected response from the peer", otherwise it may issue a MAP-NOTICE indication primitive with an appropriate diagnostic "abnormal event received from the peer".
- otherwise issue a confirm primitive mapping the operation argument parameter to the user specific parameters and setting the result parameter according to the operation code of the linked operation.

12.6.4.5 Receipt of a TC-U-REJECT indication

On receipt of a TC-U-REJECT indication primitive which affects a pending service, the MAP PM shall issue a MAP confirm primitive to the MAP-service-user with the appropriate value of the provider error or user error parameter.

The mapping of TC invoke problem codes on to MAP Provider Error and MAP User Error parameter values is described in clause 13.

12.6.4.6 Receipt of a TC-L-REJECT indication

This event occurs when the local TC detects a protocol error in an incoming component which affects an active specific service.

On receipt of a TC-L-REJECT indicating "return result problem, unexpected return result", the MAP shall issue a confirm primitive with the parameter provider error indicating "unexpected response from the peer".

On receipt of a TC-L-REJECT indicating "return error problem, unexpected error result", the MAP shall issue a confirm primitive with the parameter provider error indicating "unexpected response from the peer".

Note that when the problem code indicates a general problem, it is considered that the event cannot be related to an existing SSM even if the invoke Id is provided by TC. This is because whether the invoke Id refers to a local or remote invocation is ambiguous. The behaviour of the MAP PM in such a case is described in subclause 12.6.5.3.

12.6.4.7 Receipt of a TC-L-CANCEL indication

On receipt of a TC-L-CANCEL indication, the MAP PM shall:

- if the associated operation is a class 1 operation, issue a confirm primitive with the provider error cause indicating "no response from the peer";
- if the associated operation is a class 2 operation and no linked operations are defined for this operation, issue a confirm primitive without parameter (i.e. indicating implicitly the successful completion of the service);
- if the associated operation is a class 2 operation and has linked operations but none of them has been invoked, issue a confirm primitive with the provider error parameter indicating "service completion failure";
- if the associated operation is a class 2 operation and a linked operation invocation has already been received in response to this operation, ignore the primitive;
- if the associated operation is a class 3 operation, issue a confirm primitive with the provider error cause indicating "service completion failure";
- if the associated operation is a class 4 operation, ignore the primitive.

NOTE: When a TC-L-CANCEL ind primitive is received before the dialogue has been confirmed (i.e. no backward message is received by the dialogue initiator node), the MAP PM shall first issue a MAP-OPEN Cnf primitive with the result parameter indicating "accepted" (which means that the dialogue is considered as being implicitly accepted). Then, as indicated above, the TC-L-CANCEL Indication is interpreted according to the class of the operation to which it refers.

12.6.4.8 Receipt of a TC-NOTICE indication

If a TC-NOTICE indication primitive is received before the dialogue has been confirmed (i.e. no backward message is received by the dialogue initiator node), the MAP PM shall issue a MAP-OPEN Cnf primitive with the result parameter indicating Refused and a refuse reason Remote node not reachable".

If a TC-NOTICE indication primitive is received after the dialogue has been confirmed, the MAP PM shall issue a MAP-NOTICE indication to the user, with a problem diagnostic indicating "message cannot be delivered to the peer".

12.6.5 Other events

This subclause describes the behaviour of the MAP PM on receipt of a component handling indication primitive which cannot be related to any service or which does not affect a pending one. The MAP user is only informed that an abnormal event occurred during the associated dialogue. It is up to the MAP user to abort, continue or terminate the dialogue.

12.6.5.1 Receipt of a TC-U-REJECT

On receipt of a TC-U-REJECT indication primitive which does not affect an active SSM (i.e. indicating a return result or return error problem), the MAP PM shall issue a MAP-NOTICE indication primitive with the diagnostic parameter set to "response rejected by the peer".

This is also applicable for invoke problems related to a class 4 linked operation.

12.6.5.2 Receipt of a TC-R-REJECT indication

On receipt of a TC-R-REJECT indication (i.e. when a protocol error has been detected by the peer TC entity) which does not affect an active SSM, the MAP PM shall either discard this indication or issue a MAP-NOTICE indication primitive with the provider error indicating "abnormal event detected by the peer".

In case of notification, it is up to the MAP user to continue, abort or terminate the dialogue. Note also that for MAP V1 the reject component is received in an END message and therefore the dialogue is terminated anyway.

12.6.5.3 Receipt of a TC-L-REJECT indication

On receipt of a TC-L-REJECT indication primitive (i.e. when a protocol error has been detected by the local TC entity) which cannot be related to an active SSM, the MAP PM shall either discard this indication or issue a MAP-NOTICE indication primitive with the provider error indicating "abnormal event received from the peer".

In case of notification, it is up to the MAP user to continue, or to terminate the dialogue and implicitly trigger the transmission of the reject component or to abort the dialogue.

12.6.6 Parameter checks

As described in the previous subclauses, the MAP PM performs a set of checks to ensure the correctness of the information elements received; these are:

- check if the syntax and encoding (note) of the operation argument, result or error parameter are correct.

NOTE: Depending on the implementation, encoding problems on the TC user portion may be detected at TC level or by the MAP user. In the second case the problem is reported in a similar manner to a syntactical problem.

The syntax shall be considered incorrect if a mandatory information element is missing in any constructed element or if the value of an information element is out of the range defined for the type it is supposed to belong to;

- if there is not a one-to-one mapping between a service and an operation:
 - i) check if the value of the information elements (generally a single one) permits the MAP PM to determine the service associated with the operation invocation;
 - ii) check that there are no information elements which are irrelevant for the indication or a confirm primitive to be issued;
- check if all the information elements required to build an indication or a confirm primitive are available.

However some additional checks may have to be performed by the MAP user (see clause 15).

12.6.7 Returning state machines to idle

Unlike TC invocation state machines, service state machines exist at both requestor and performer side.

A service state machine at the requestor side is returned to idle when the MAP-specific confirm primitive is issued or when the dialogue terminates.

A service state machine at the performer side is returned to idle on receipt of a MAP-specific response primitive from the MAP user, when the dialogue terminates or at expiry of an implementation dependent watch-dog timer which is started when the state machine is created.

12.6.8 Load control

As stated in the previous subclauses, before issuing a MAP-OPEN indication primitive the MAP PM performs a check to verify if there are sufficient resources to open the dialogue taking into account possible overload conditions.

The decision is based on the priority allocated to the application-context whose name is explicitly included in the TC-BEGIN indication primitive or implied by the first operation invocation when V1 contexts are in use. How a V1 application-context-name is derived from an operation code is described in table 12.1/1.

The priority level allocated to each application-context is described in clause 3 tables 3.1/1 and 3.1/2.

13 Mapping on to TC services

13.1 Dialogue control

Dialogue control services are mapped to TC dialogue handling services. The TC-UNI service is not used by the MAP PM.

13.1.1 Directly mapped parameters

The following parameters of the MAP-OPEN request and indication primitives are directly mapped on to the corresponding parameters of the TC-BEGIN primitives:

- destination address;
- originating address.

13.1.2 Use of other parameters of dialogue handling primitives

13.1.2.1 Dialogue Id

The value of this parameter is associated with the MAP PM invocation in an implementation dependent manner.

13.1.2.2 Application-context-name

The application-context-name parameter of a MAP primitive is mapped to the application-context-name parameter of TC dialogue handling primitives according to the rules described in subclause 12.1.

13.1.2.3 User information

The user information parameter of TC dialogue primitives is used to carry the MAP dialogue APDUs.

13.1.2.4 Component present

This parameter is used by the MAP PM as described in CCITT Recommendation Q.771. It is not visible to the MAP user.

13.1.2.5 Termination

The value of this parameter of the TC-END request primitive is set by the MAP PM on the basis of the release method parameter of the MAP-CLOSE request primitive, except when the dialogue state machine is in the state DIALOGUE INITIATED, in which case the Termination parameter shall always indicate "pre-arranged end".

13.1.2.6 P-Abort-Cause

Values of the P-abort-cause parameter are mapped to the values of the provider-reason parameter of the MAP-P-ABORT indication primitive according to table 13.1/1, except in the dialogue initiated phase for the "incorrectTransactionPortion" and "noCommonDialoguePortion" values which are mapped to the "potential incompatibility problem" value of the refuse-reason parameter of the MAP-OPEN cnf primitive. The source parameter in the MAP-P-ABORT ind takes the value "TC problem".

13.1.2.7 Quality of service

The quality of service of TC request primitives is set by the MAP as shown below.

- Return option: "Return message on error" or "Discard message on error" as required by the network operator;
- Sequence control: "Sequence guaranteed" or "Sequence result not guaranteed" as required by the network operator;

"Sequence guaranteed" shall be used when a segmented result is to be transferred (e.g. subscriber data in response to SendParameters). It may also be appropriate to use Sequence guaranteed when a series of InsertSubscriberData, ProcessAccessSignalling or ForwardAccessSignalling operations is used.

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.

TC P-Abort cause	MAP provider-reason
unrecognized message type	provider malfunction
unrecognized transaction Id	supporting dialogue released
badlyFormattedTransactionPortion	provider malfunction
incorrectTransactionPortion	provider malfunction (note)
resourceLimitation	resource limitation
abnormalDialogue	provider malfunction
noCommonDialoguePortion	version incompatibility

NOTE: Or version incompatibility in the dialogue initiated phase.

Table 13.1/1: Mapping of P-Abort cause in TC-P-ABORT indication on to provider-reason in MAP-P-ABORT indication

13.2 Service specific procedures

Specific services are mapped to TC component handling services.

13.2.1 Directly mapped parameters

The Invoke Id parameter of the MAP request and indication primitive is directly mapped on to the Invoke Id parameter of the component handling primitives.

13.2.2 Use of other parameters of component handling primitives

13.2.2.1 Dialogue Id

The value of this parameter is associated with the MAP PM invocation in an implementation dependent manner.

13.2.2.2 Class

The value of this parameter is set by the MAP PM according to the type of the operation to be invoked.

13.2.2.3 Linked Id

When a service response is mapped to a class 4 operation, the value of this parameter is set by the MAP PM and corresponds to the value assigned by the user to the initial service request (i.e. the value of the invoke ID parameter of the request primitive). Otherwise if such a parameter is included in MAP request/indication primitives it is directly mapped to the linked ID parameter of the associated TC-INVOKE request/indication primitives.

13.2.2.4 Operation

When mapping a request primitive on to a Remote Operations PDU (invoke), the MAP PM shall set the operation code according to the mapping described in table 13.2/1.

When mapping a response primitive on to a Remote Operations service, the MAP PM shall set the operation code of the TC-RESULT-L/NL primitive (if required) to the same value as the one received at invocation time.

MAP-SERVICE	operation
MAP-ACTIVATE-SS	activateSS
MAP-ACTIVATE-TRACE-MODE	activateTraceMode
MAP-ALERT-SERVICE-CENTRE	alertServiceCentre
MAP-ANY-TIME-INTERROGATION	anyTimeInterrogaton
MAP-CANCEL-LOCATION	cancelLocation
MAP-CHECK-IMEI	checkIMEI
MAP-DEACTIVATE-SS	deactivateSS
MAP-DEACTIVATE-TRACE-MODE	deactivateTraceMode
MAP-DELETE-SUBSCRIBER-DATA	deleteSubscriberData
MAP-ERASE-SS	eraseSS
MAP-FORWARD-ACCESS-SIGNALLING	forwardAccessSignalling
MAP-FORWARD-CHECK-SS-INDICATION	forwardCheckSsIndication
MAP-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-PREPARE-HANDOVER	prepareHandover
MAP-PREPARE-SUBSEQUENT-HANDOVER	prepareSubsequentHandover
MAP-PROCESS-ACCESS-SIGNALLING	processAccessSignalling
MAP-PROCESS-UNSTRUCTURED-SS-REQUEST	processUnstructuredSS-Request
MAP-PROVIDE-ROAMING-NUMBER	provideRoamingNumber
MAP-PROVIDE-SUBSCRIBER-INFO	provideSubscriberInfo
MAP-PURGE-MS	purgeMS
MAP-READY-FOR-SM	readyForSM
MAP-REGISTER-PASSWORD	registerPassword
MAP-REGISTER-SS	registerSS
MAP-REPORT-SM-DELIVERY-STATUS	reportSmDeliveryStatus
MAP-RESET	reset
MAP-RESTORE-DATA	restoreData
MAP-SEND-END-SIGNAL	sendEndSignal
MAP-SEND-AUTHENTICATION-INFO	sendAuthenticationInfo
MAP-SEND-IMSI	sendIMSI
MAP-SEND-IDENTIFICATION	sendIdentification
MAP-SEND-ROUTING-INFO-FOR-SM	sendRoutingInfoForSM
MAP-SEND-ROUTING-INFORMATION	sendRoutingInfo
MAP-UNSTRUCTURED-SS-NOTIFY	unstructuredSS-Notify
MAP-UNSTRUCTURED-SS-REQUEST	unstructuredSS-Request
MAP-UPDATE-LOCATION	updateLocation

Table 13.2/1: Mapping of MAP specific services on to MAP operations

13.2.2.5 Error

The error parameter in a TC-U-ERROR indication primitive is mapped to the user error parameter in the MAP confirm primitive of the service associated with the operation to which the error is attached.

The user error parameter in MAP response primitives is mapped to the error parameter of the TC-U-ERROR request primitive, except for "initiating-release" and "resource-limitation" which are mapped to the problem code parameter of the TC-U-REJECT request primitive.

13.2.2.6 Parameters

The parameters of MAP specific request and indication primitives are mapped to the argument parameter of TC-INVOKE primitives.

The parameters of MAP specific response and confirm primitives are mapped to the result parameter of TC-RESULT-L primitives, the parameter of TC-U-ERROR primitives or the argument of TC-INVOKE primitives when mapping on linked class 4 operations is used.

13.2.2.7 Time out

The value of this parameter is set by the MAP PM according to the type of operation invoked.

13.2.2.8 Last component

This parameter is used by the MAP PM as described in CCITT Recommendation Q.711. It is not visible from the MAP user.

13.2.2.9 Problem code

13.2.2.9.1 Mapping to MAP User Error

The following values of the user error parameter are mapped as follows to values of the TC problem code parameter. These values are generated by the MAP user. This mapping is valid from the TC-U-REJECT indication primitive to the MAP confirm service primitive and from the MAP response service primitive to the TC-U-REJECT request primitive.

MAP User Error	TC problem code
resource limitation	resource limitation
initiating release	initiating release

Table 13.2/2: Mapping of MAP User Error parameter on to TC problem code in TC-U-REJECT primitives

13.2.2.9.2 Mapping to MAP Provider Error parameter

The following values of the TC problem code parameter of the TC-U-REJECT indication primitive are mapped as follows to values of the MAP Provider Error parameter of the MAP confirm primitive.

TC problem code	MAP Provider Error
duplicated invoke Id	duplicated invoke id
unrecognized operation	service not supported
mistyped parameter	mistyped parameter

Table 13.2/3: Mapping of TC problem code in TC-U-REJECT on to MAP Provider Error parameter

The following values of the problem code parameters of the TC-L-REJECT primitive are mapped to values of the provider error parameter of the MAP confirm primitive as follows:

TC problem code	MAP Provider Error
return result unexpected	unexpected response from the peer
return error unexpected	unexpected response from the peer

Table 13.2/4: Mapping of TC problem code in TC-L-REJECT on to MAP Provider Error parameter

13.2.2.9.3 Mapping to diagnostic parameter

The following values of the problem code parameter of the TC-R-REJECT and TC-U-REJECT primitive are mapped to values of the diagnostic parameter of the MAP-NOTICE indication primitive as follows:

TC problem code	MAP diagnostic
General problem	
abnormal event detected by the peer	
Invoke problem	
- unrecognized linked ID	- abnormal event detected by the peer
- linked response unexpected	- response rejected by the peer
- unexpected linked operation	- response rejected by the peer
Return result problem	
- unrecognized invoke ID	- response rejected by the peer
- return result unexpected	- response rejected by the peer
- mistyped parameter	- response rejected by the peer
Return error problem	
- unrecognized invoke ID	- response rejected by the peer
- return error unexpected	- response rejected by the peer
- unrecognized error	- response rejected by the peer
- unexpected error	- response rejected by the peer
- mistyped parameter	- response rejected by the peer

Table 13.2/5: Mapping of TC problem code of TC-R-REJECT and TC-U-REJECT on to diagnostic parameter

The following values of the problem code parameter of the TC-L-REJECT primitive are mapped to values of the diagnostic parameter of the MAP-NOTICE indication primitive as follows:

TC problem code	MAP diagnostic
General problems:	- abnormal event received from the peer
Invoke problem:	
- unrecognized linked ID	- abnormal event received from the peer
Return result problem:	
- unrecognized invoke ID	- abnormal event received from the peer
Return error problem:	
- unrecognized invoke ID	- abnormal event received from the peer

Table 13.2/6: Mapping of TC problem code of TC-L-REJECT on to diagnostic parameter

13.3 SDL descriptions

The following SDL specification describes a system which includes three blocks: MAP-user, MAP-provider and TC.

Such a system resides in each network component supporting MAP and communicates with its peers via the lower layers of the signalling network which are part of the environment.

Only the MAP-provider is fully described in this subclause. The various type of processes which form the MAP-User block and the TC block are described respectively in clauses 15 to 21 of this ETS and in CCITT Recommendation Q.774.

The MAP-Provider block communicates with the MAP_USER via two channels U1 and U2. Via U1 the MAP-provider receives the MAP request and response primitives. Via U2 it sends the MAP indication and confirm primitives.

The MAP-Provider block communicates with TC via two channels P1 and P2. Via P1 the MAP-Provider sends all the TC request primitives. Via P2 it receives all the TC indication primitives.

The MAP-Provider block is composed of the four following types of processes:

- a) MAP_DSM: This type of process handles a dialogue. There exists one process instance per MAP dialogue.
- b) LOAD_CTRL: This type of process is in charge of load control. There is only one instance of this process in each system.
- c) PERFORMING_MAP_SSM: This type of process 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 organized 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.

Figure 13.2/1: System MAP_STACK

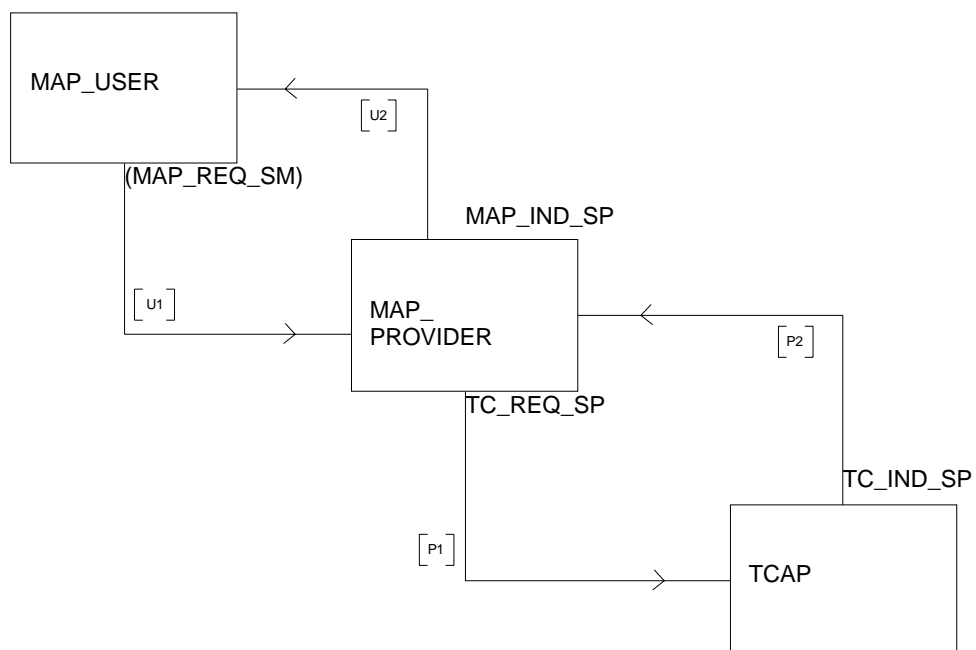


Figure 13.2/2: Block MAPPROVIDER

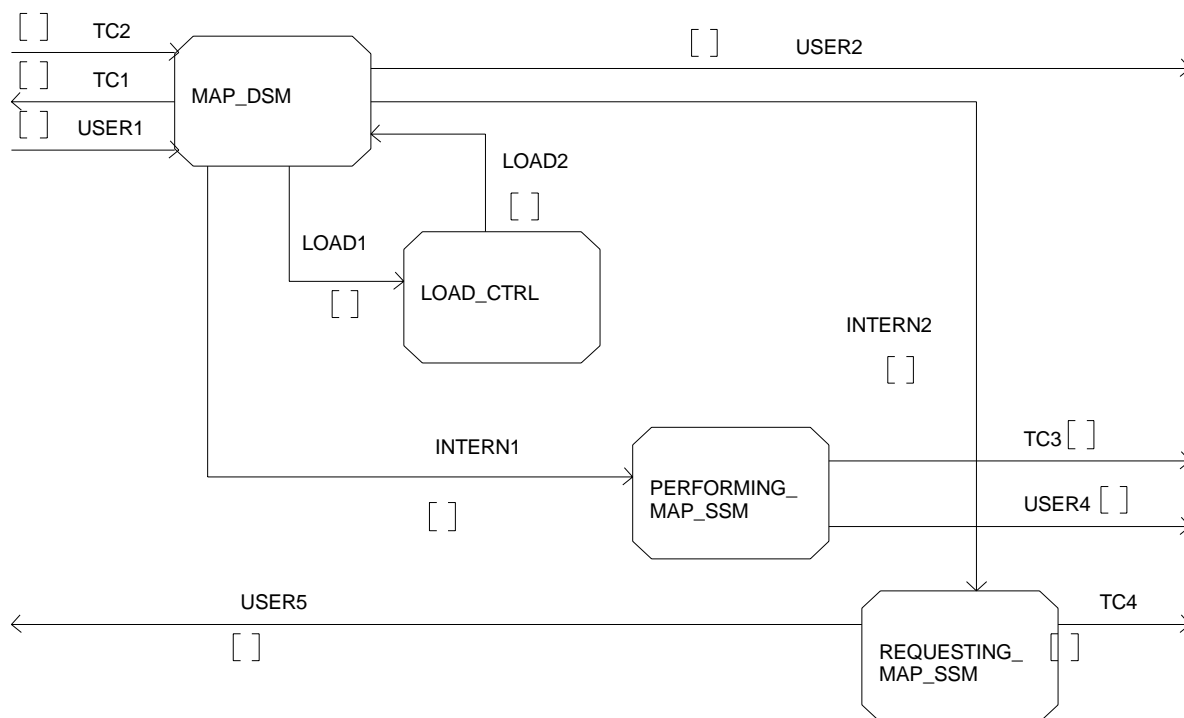


Figure 13.2/3 (sheet 1 of 11): Process MAP_DSM

Process MAP_DSM

132_31(11)



Comment 'MAP Dialoges State Maschine':
DCL
COMPONENTS_PRESENT, INVOKEID_ACTIVE, LAST_COMPONENT, OP_EXIST BOOLEAN,
OP_CODE INTEGER;

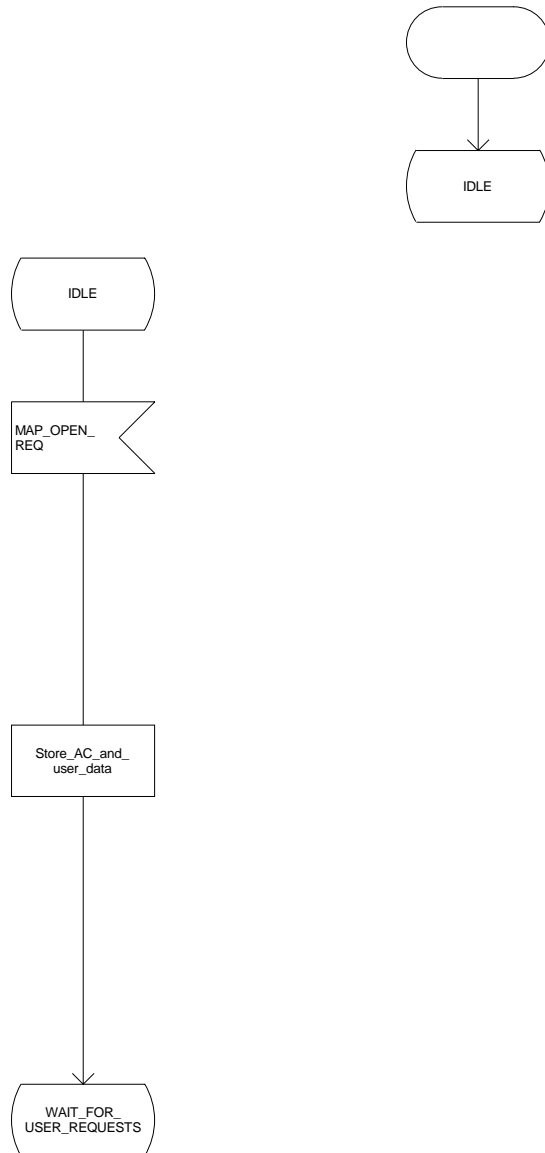


Figure 13.2/3 (sheet 2 of 11): Process MAP_DSM

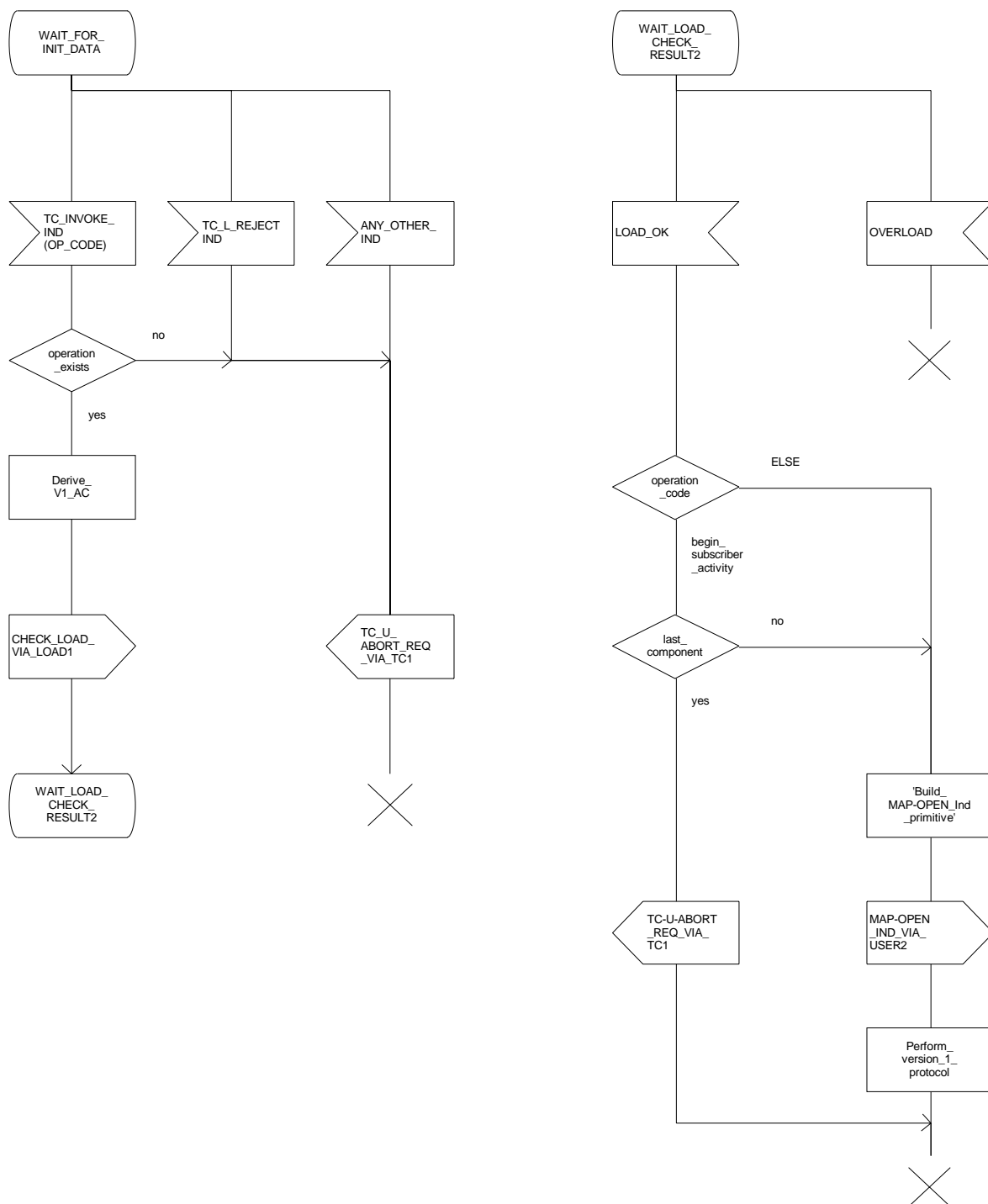


Figure 13.2/3 (sheet 3 of 11): Process MAP_DSM

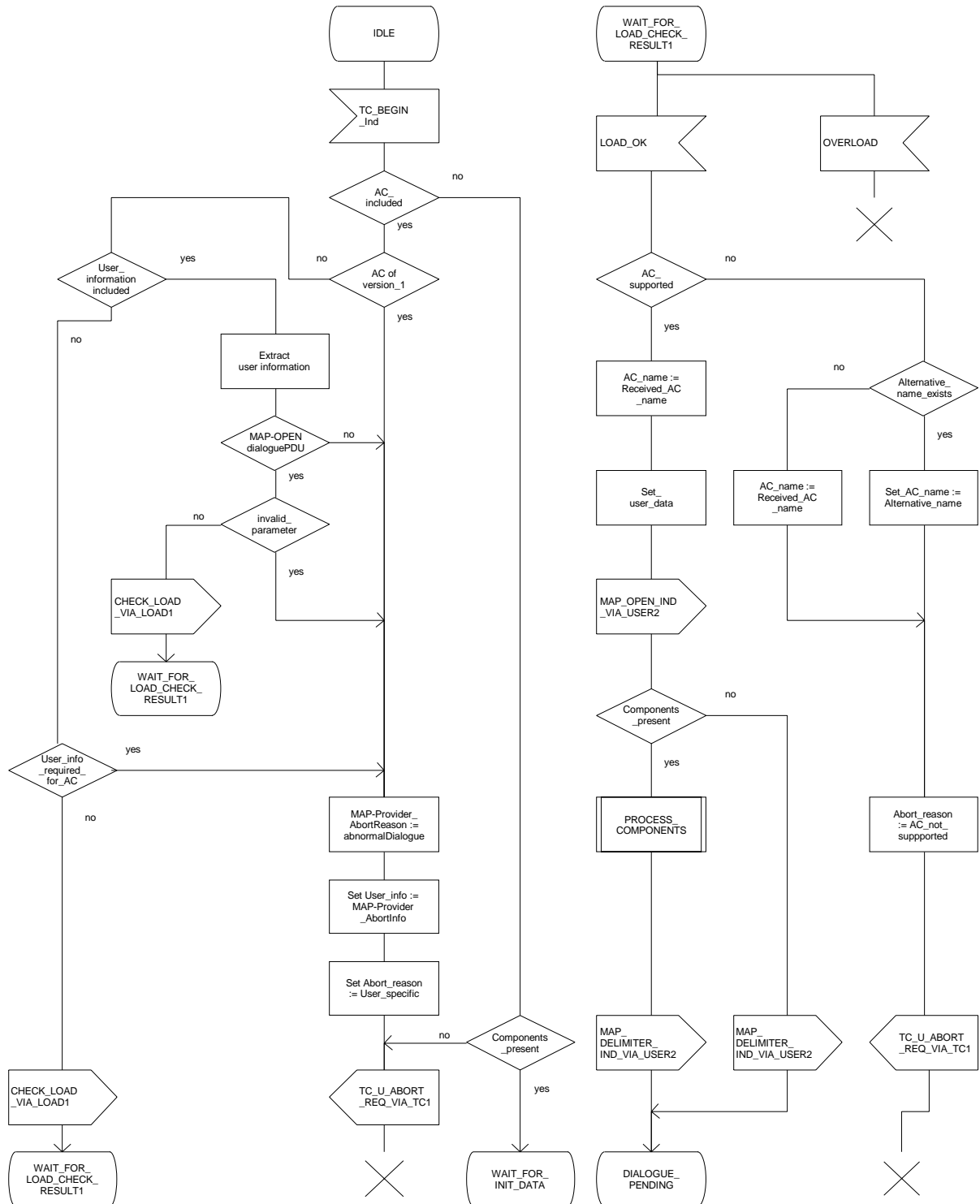


Figure 13.2/3 (sheet 4 of 11): Process MAP_DSM

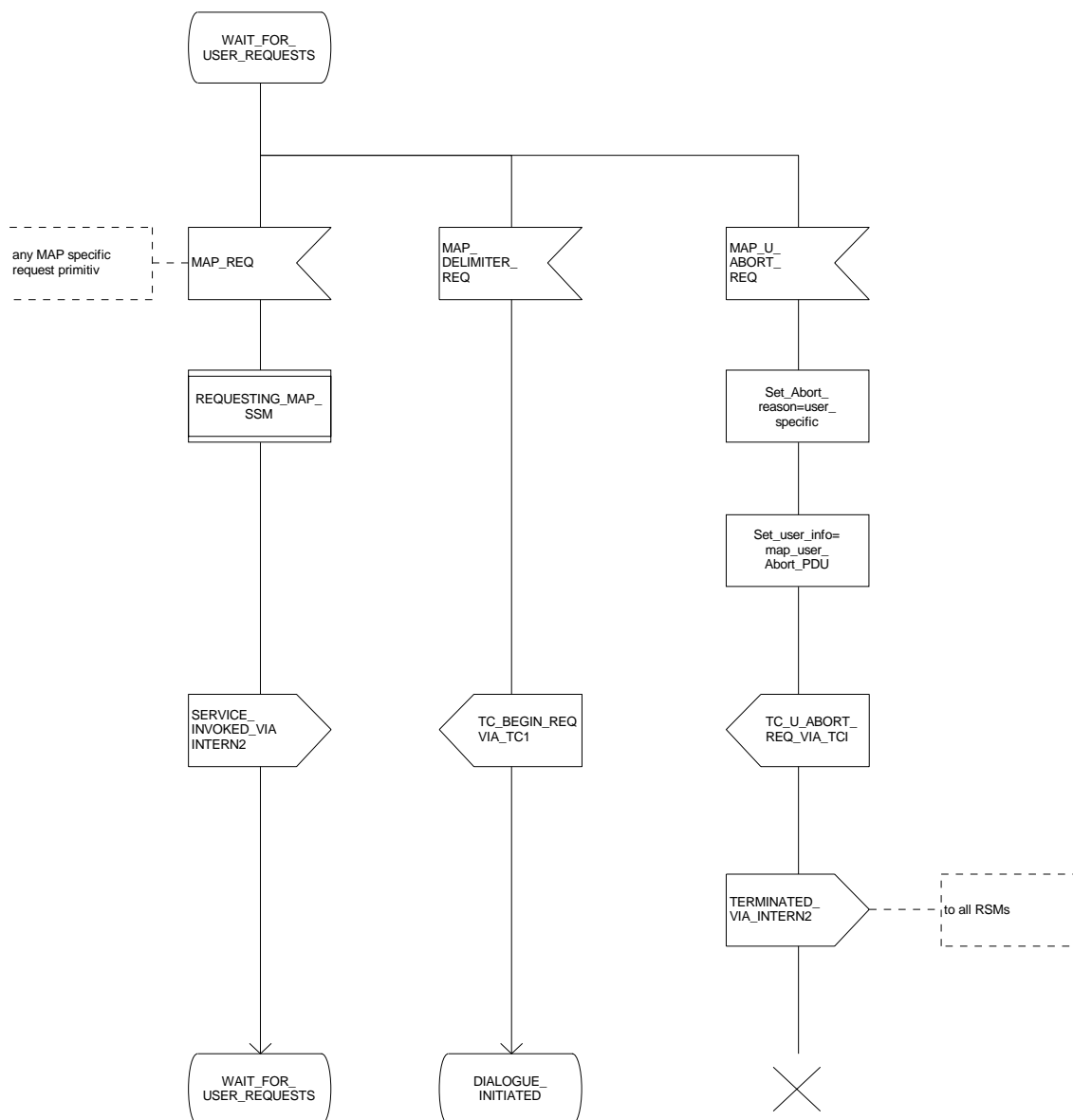


Figure 13.2/3 (sheet 5 of 11): Process MAP_DSM

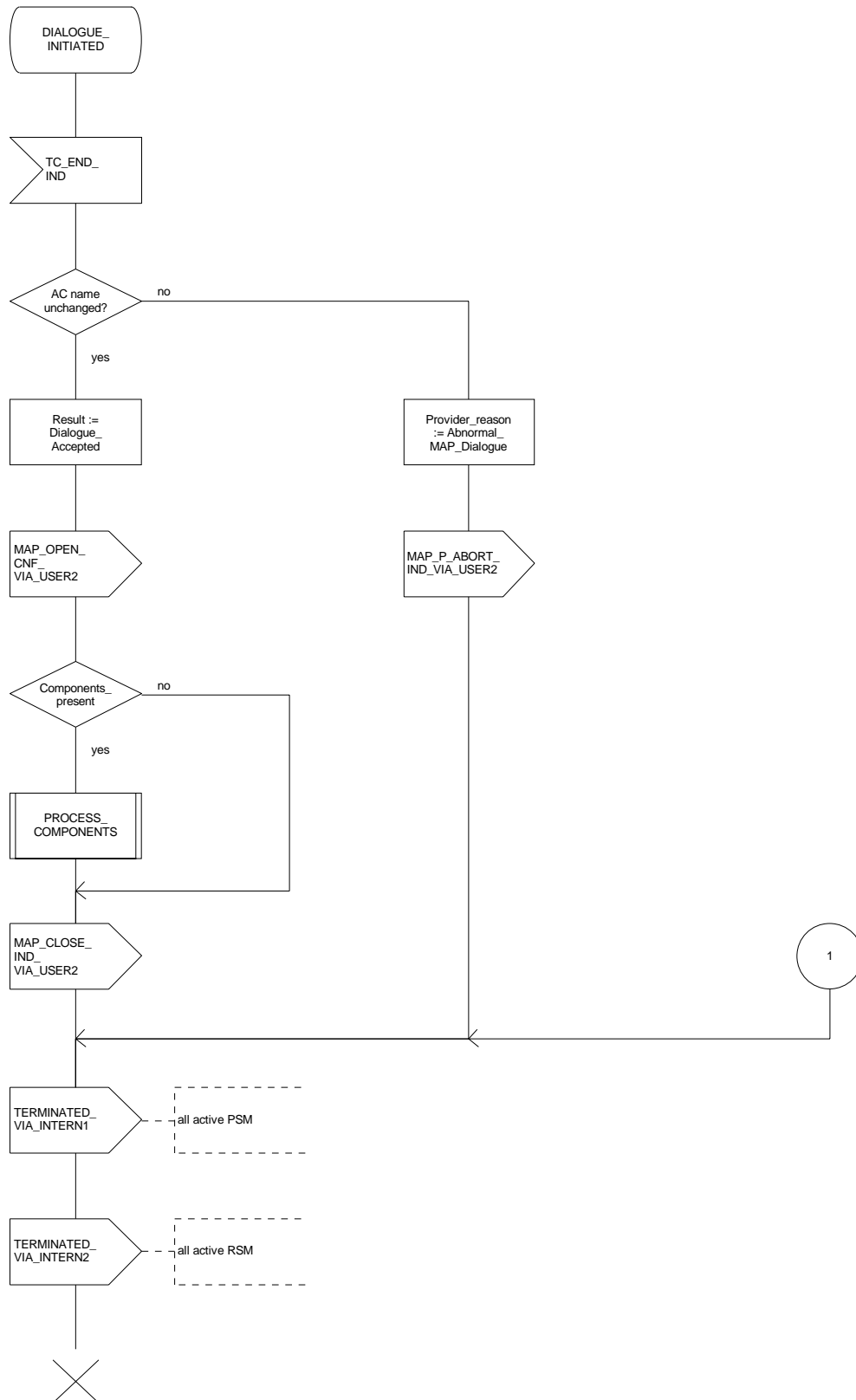


Figure 13.2/3 (sheet 6 of 11): Process MAP_DSM

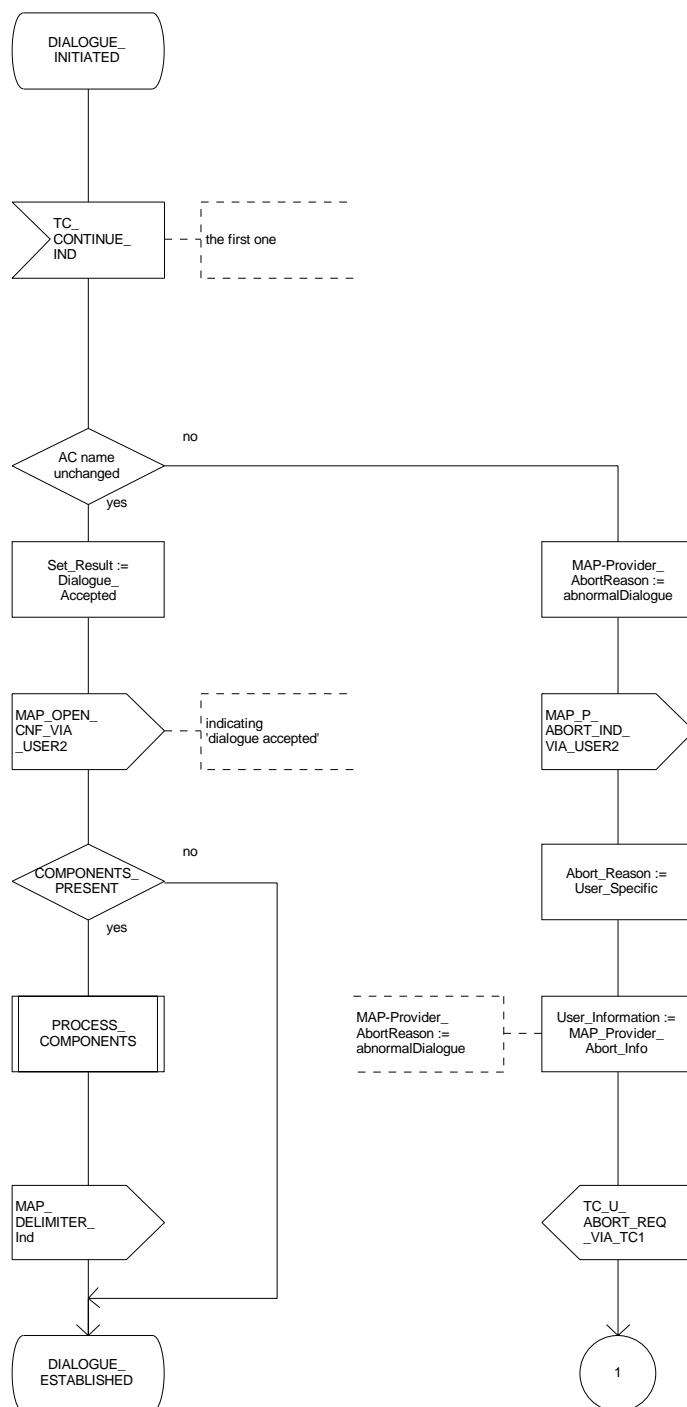


Figure 13.2/3 (sheet 7 of 11): Process MAP_DSM

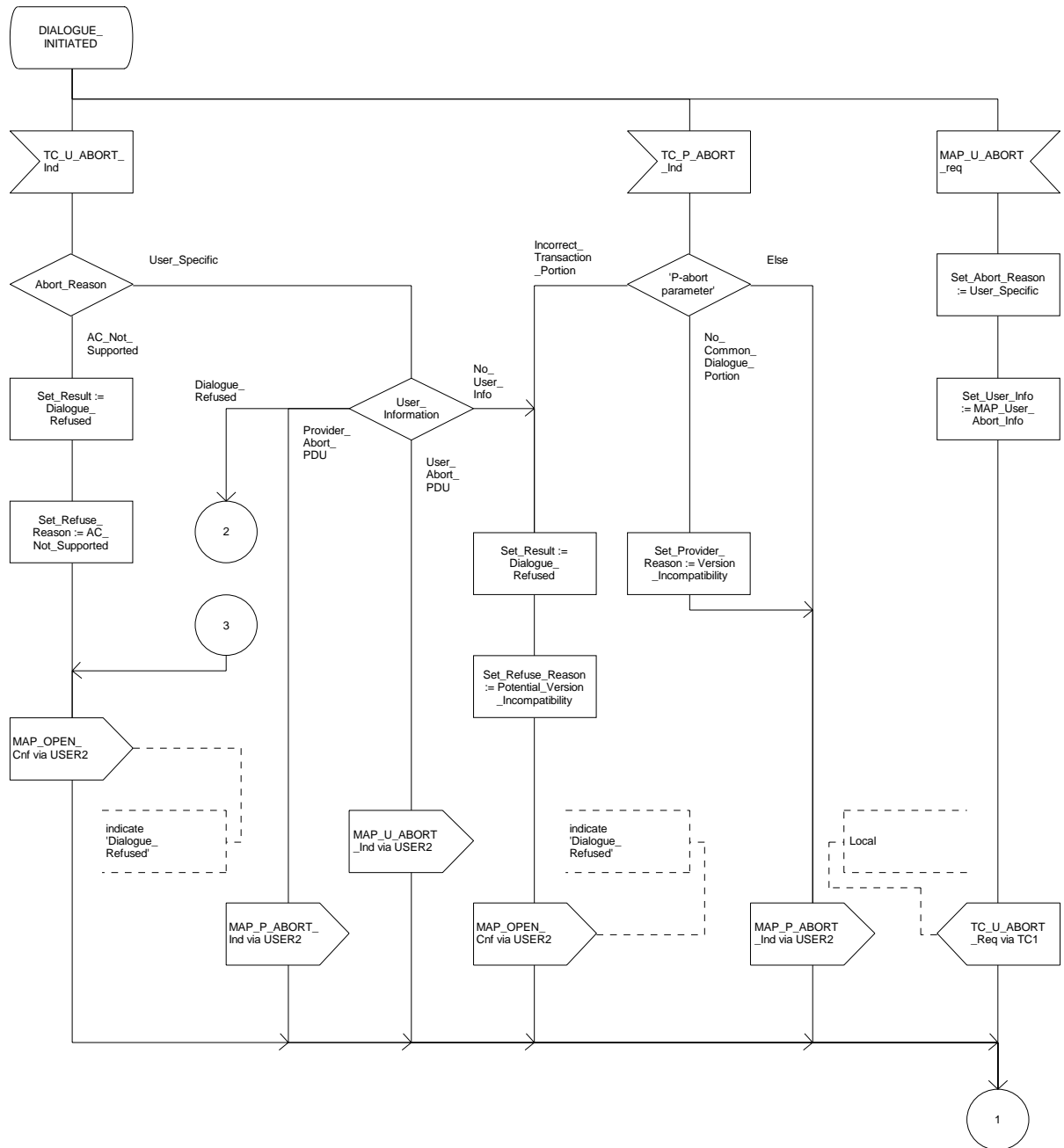


Figure 13.2/3 (sheet 8 of 11): Process MAP_DSM

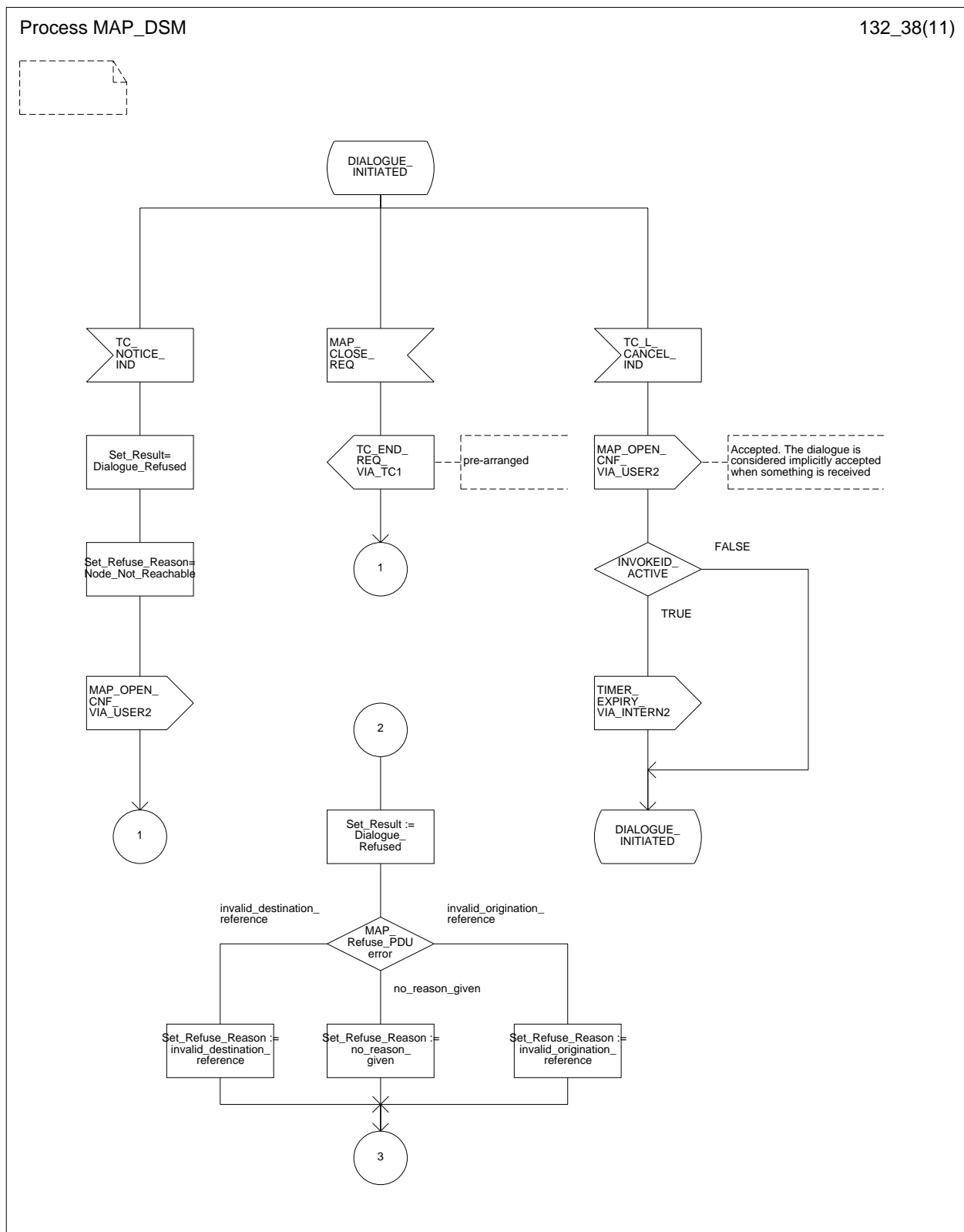


Figure 13.2/3 (sheet 9 of 11): Process MAP_DSM

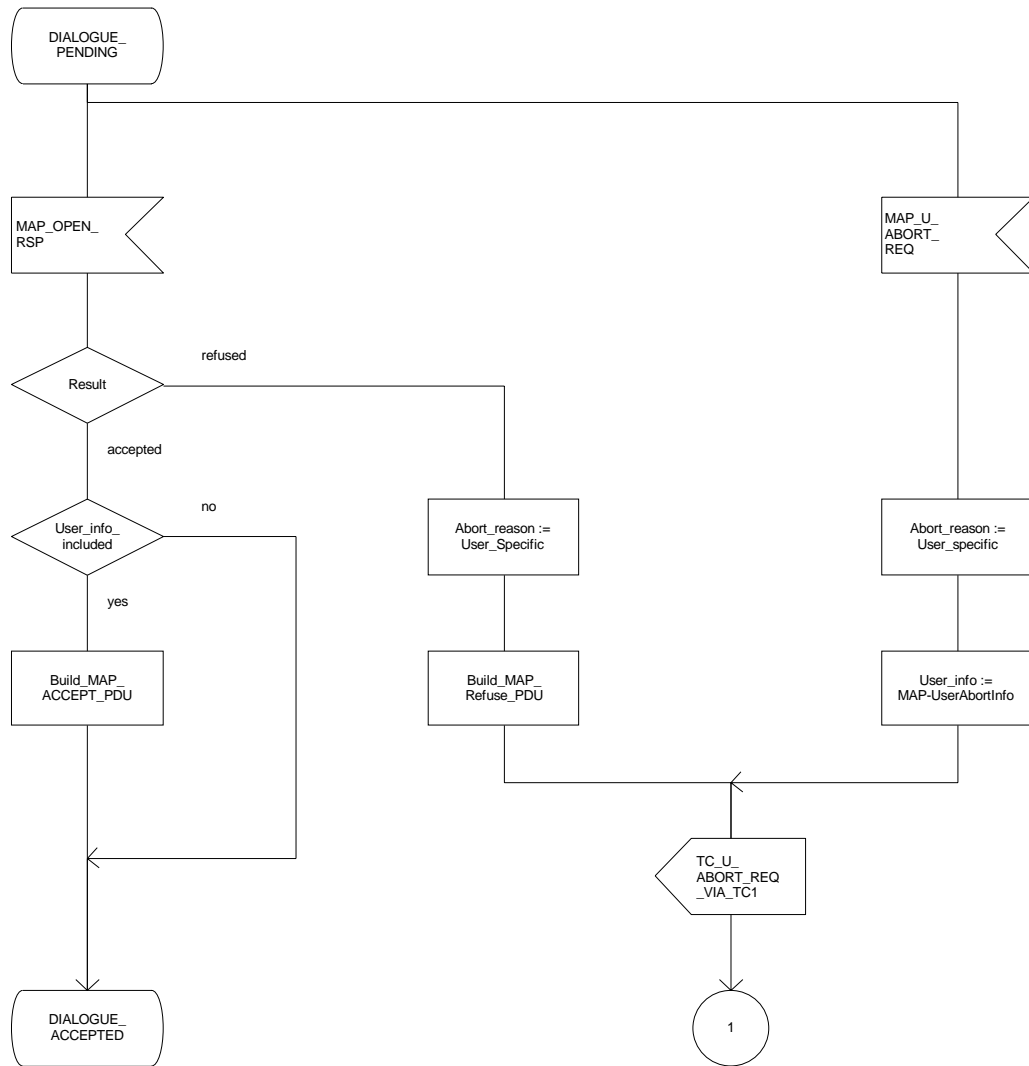


Figure 13.2/3 (sheet 10 of 11): Process MAP_DSM

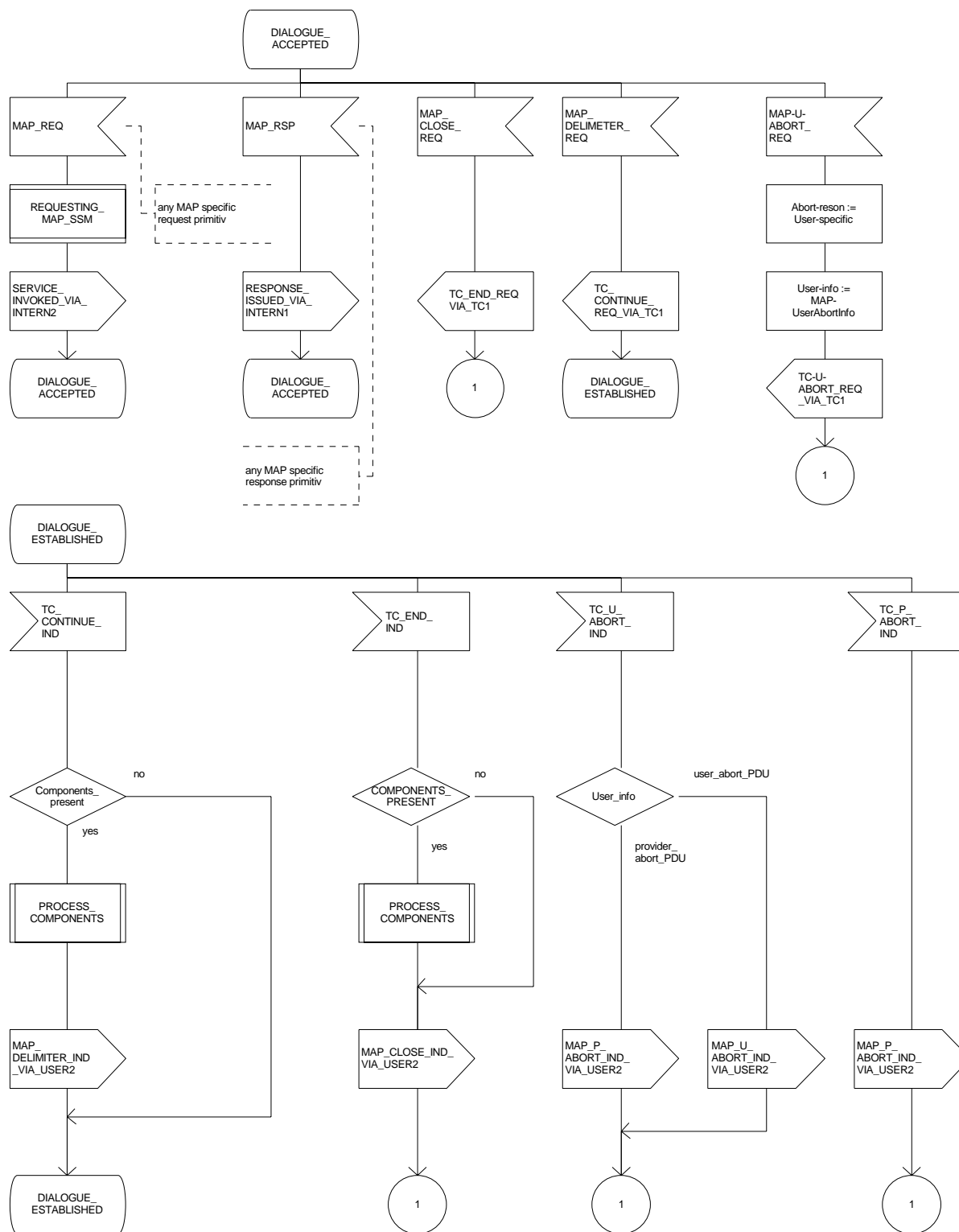


Figure 13.2/3 (sheet 11 of 11): Process MAP_DSM

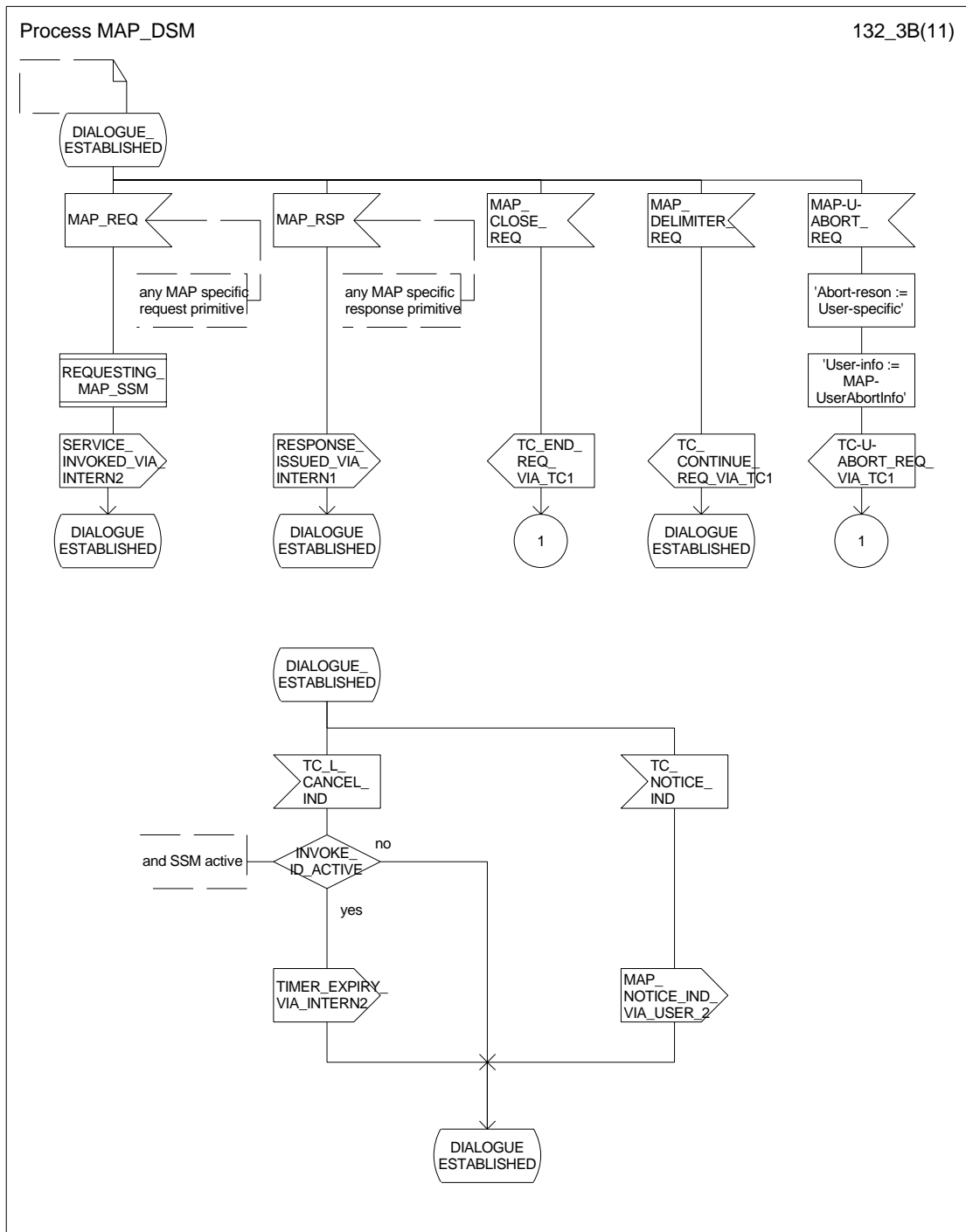


Figure 13.2/4 (sheet 1 of 4): Procedure PROCESS_COMPONENTS

Comments: Components from TCAP:
DCL
OP_CODE INTERGER;
OP_EXIST, LAST_COMPONENT, INVOKEID_ASS, LINKEDID_PRES, LINKEDID_ASS BOOLEAN;

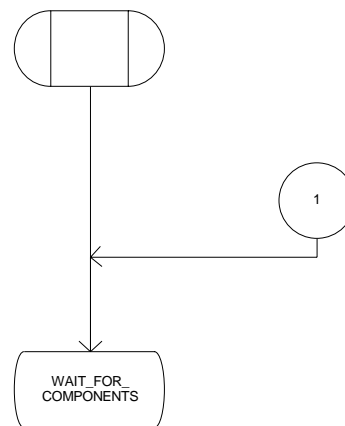


Figure 13.2/4 (sheet 2 of 4): Procedure PROCESS_COMPONENTS

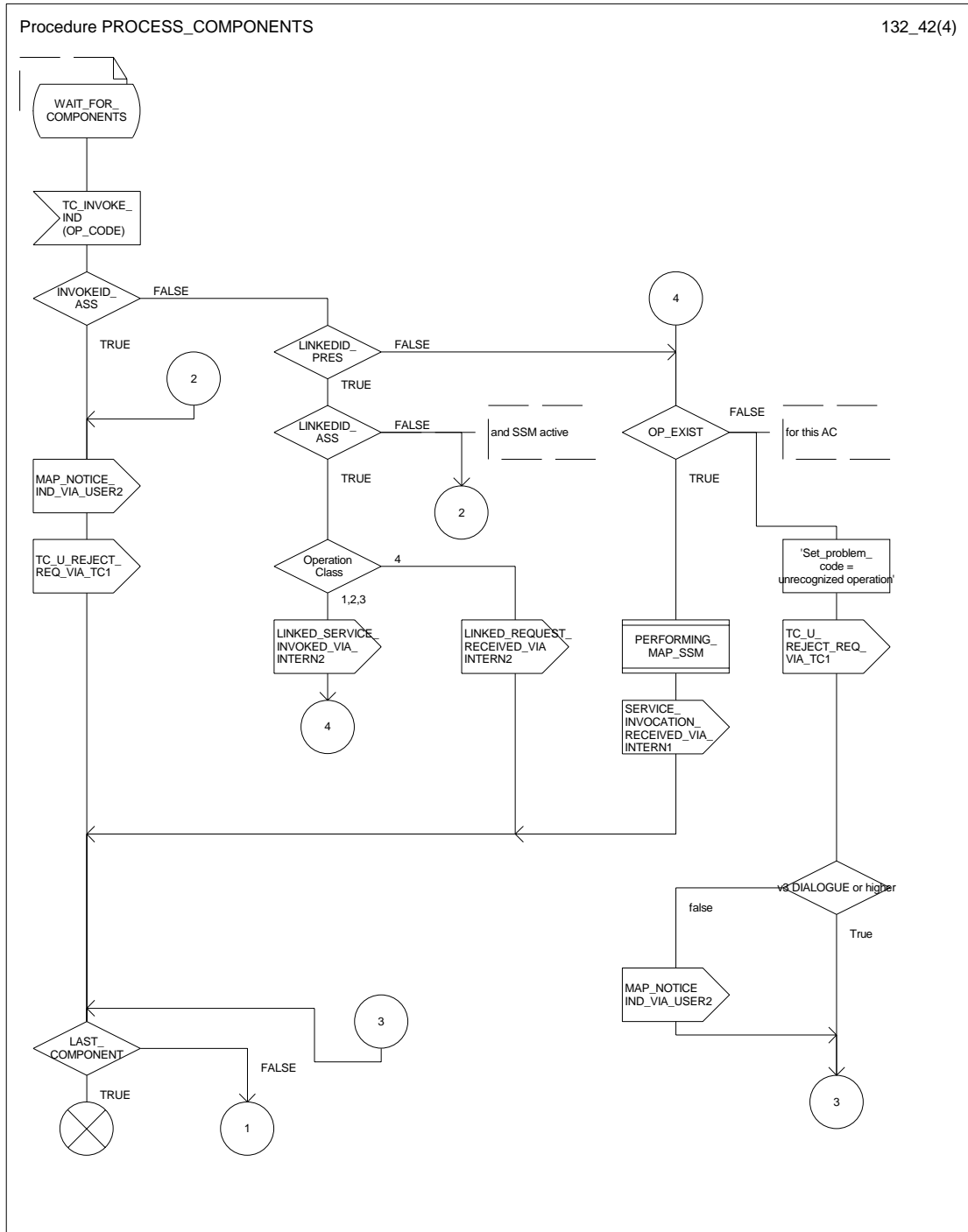


Figure 13.2/4 (sheet 3 of 4): Procedure PROCESS_COMPONENTS

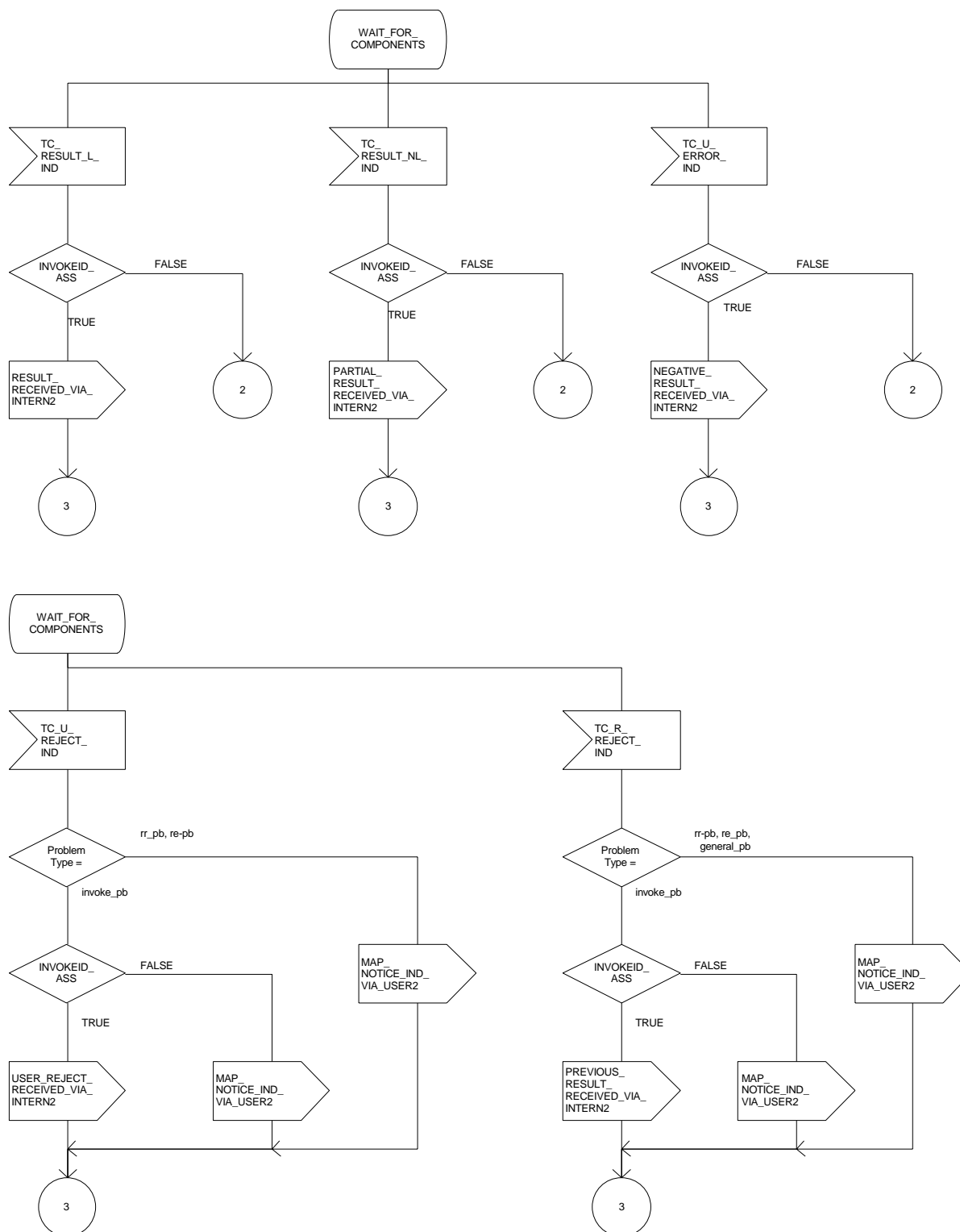


Figure 13.2/4 (sheet 4 of 4): Procedure PROCESS_COMPONENTS

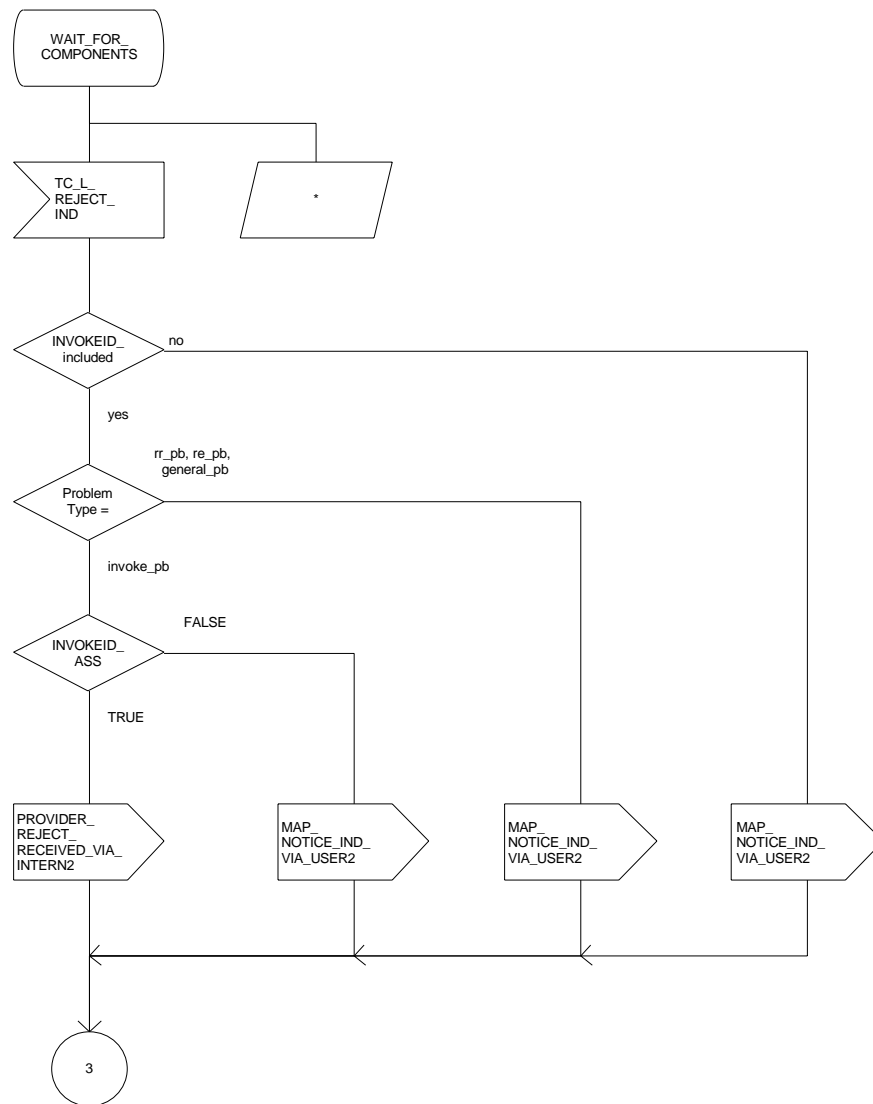


Figure 13.2/5: Process LOAD_CTRL

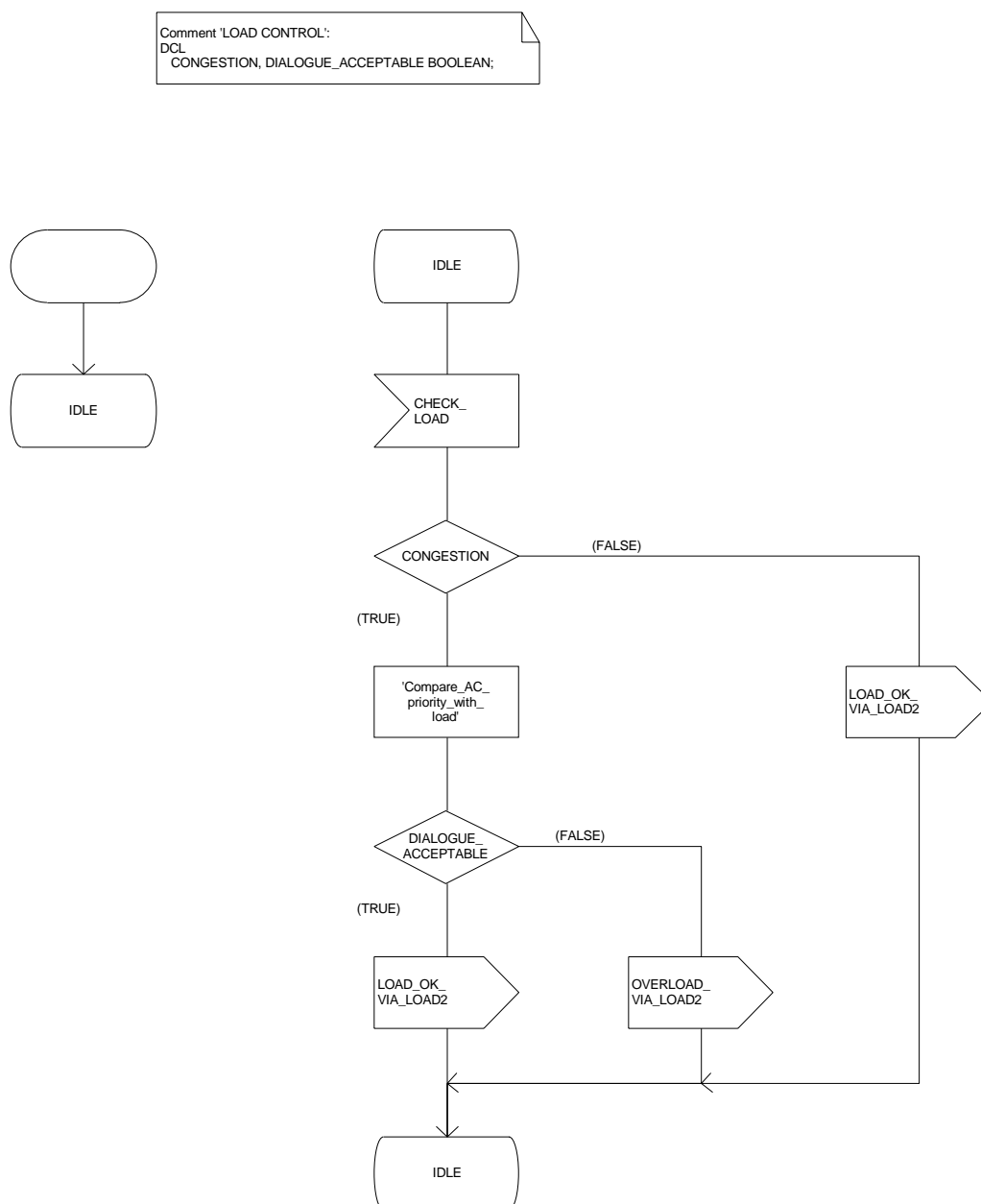


Figure 13.2/6 (sheet 1 of 3): Process PERFORMING_MAP_SSM

Comment 'MAP Service State Machine':
DCL
ARGUMENT_CORRECT, USER_ERROR_PRESENT,
SPECIFIC_ERROR_LINKED_REQUEST, CNF BOOLEAN,
OP_CLASS INTEGER,
TIMER_GUARD_TIMER COMMENT 'expires if MAP user does not respond';

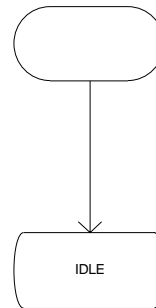


Figure 13.2/6 (sheet 2 of 3): Process PERFORMING_MAP_SSM

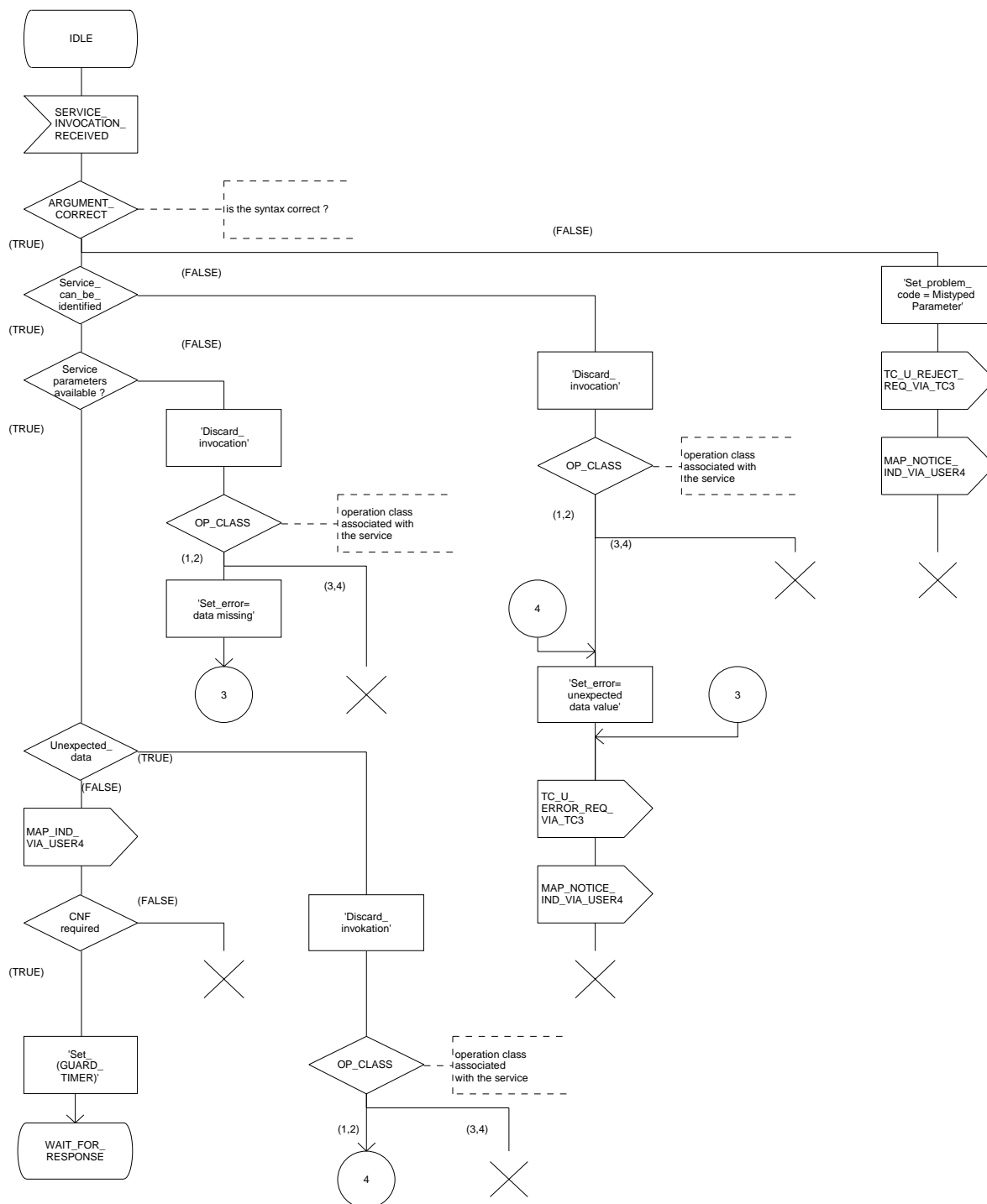


Figure 13.2/6 (sheet 3 of 3): Process PERFORMING_MAP_SSM

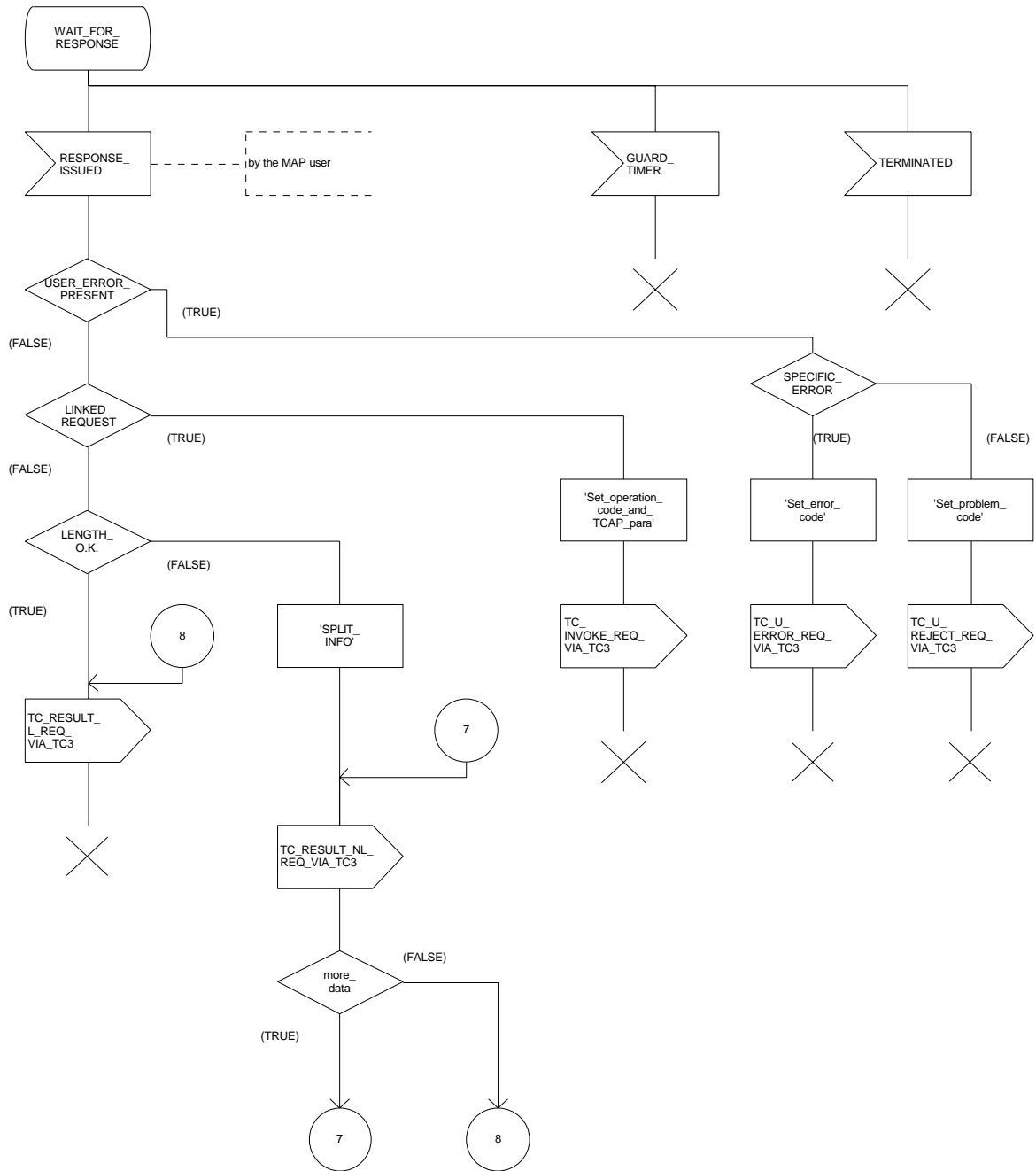


Figure 13.2/7 (sheet 1 of 4): Process REQUESTING_MAP_SSM

Comment 'MAP Service State Maschine':

DCL

ARGUMENT_CORRECT, ERROR_CODE_CORRECT, LINKED_REQ_DEF, SYNTAX_CORRECT,
MAP_INITIATED, CNF, LINKED_OPERATION_ALLOWED BOOLEAN,
OP_CLASS INTEGER;

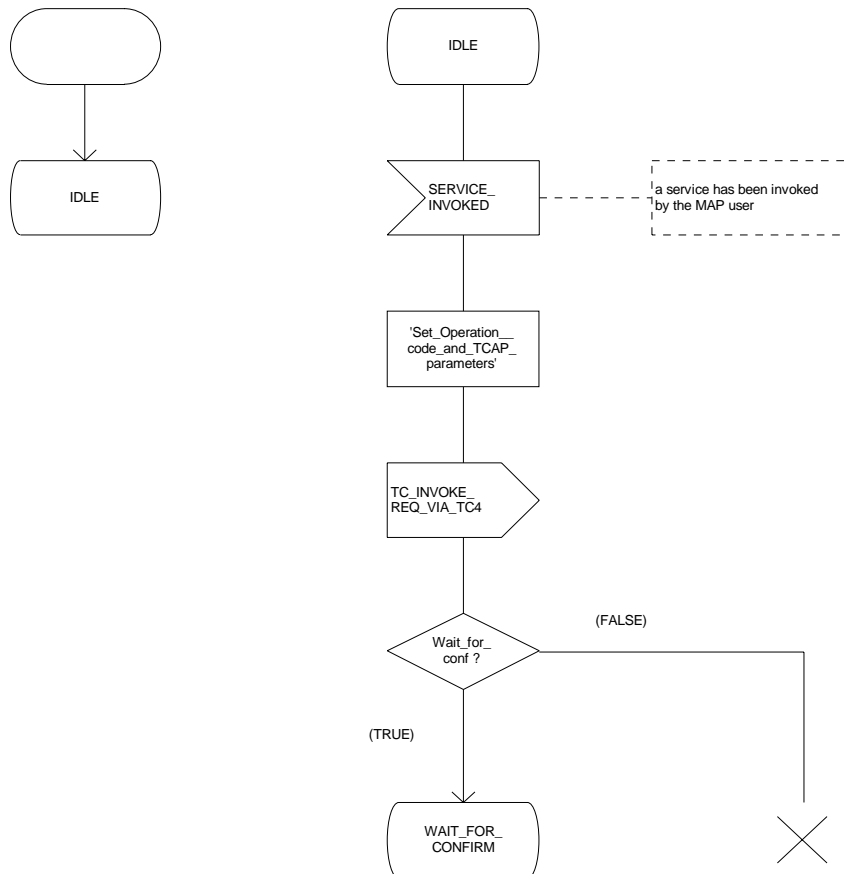


Figure 13.2/7 (sheet 2 of 4): Process REQUESTING_MAP_SSM

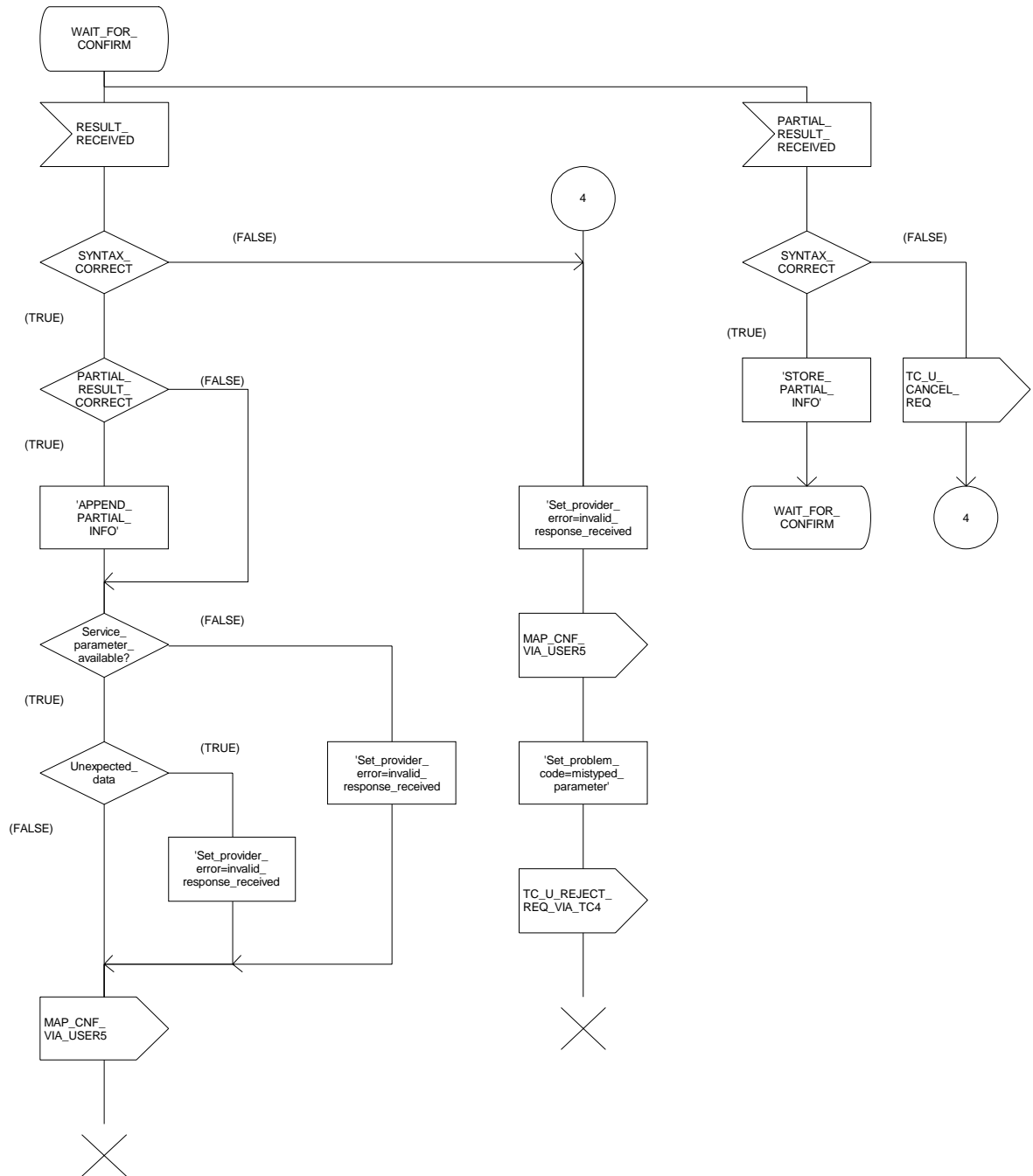


Figure 13.2/7 (sheet 3 of 4): Process REQUESTING_MAP_SSM

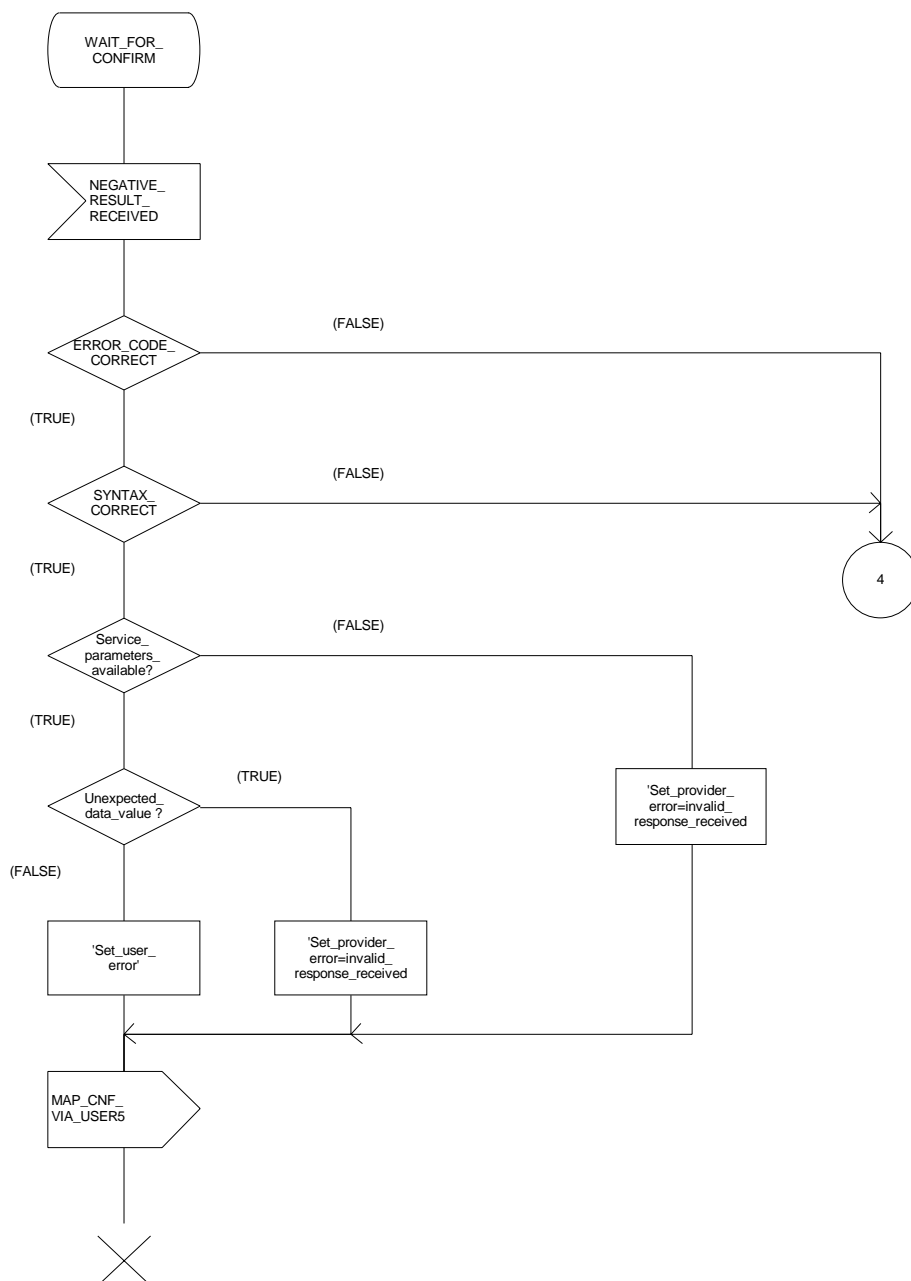
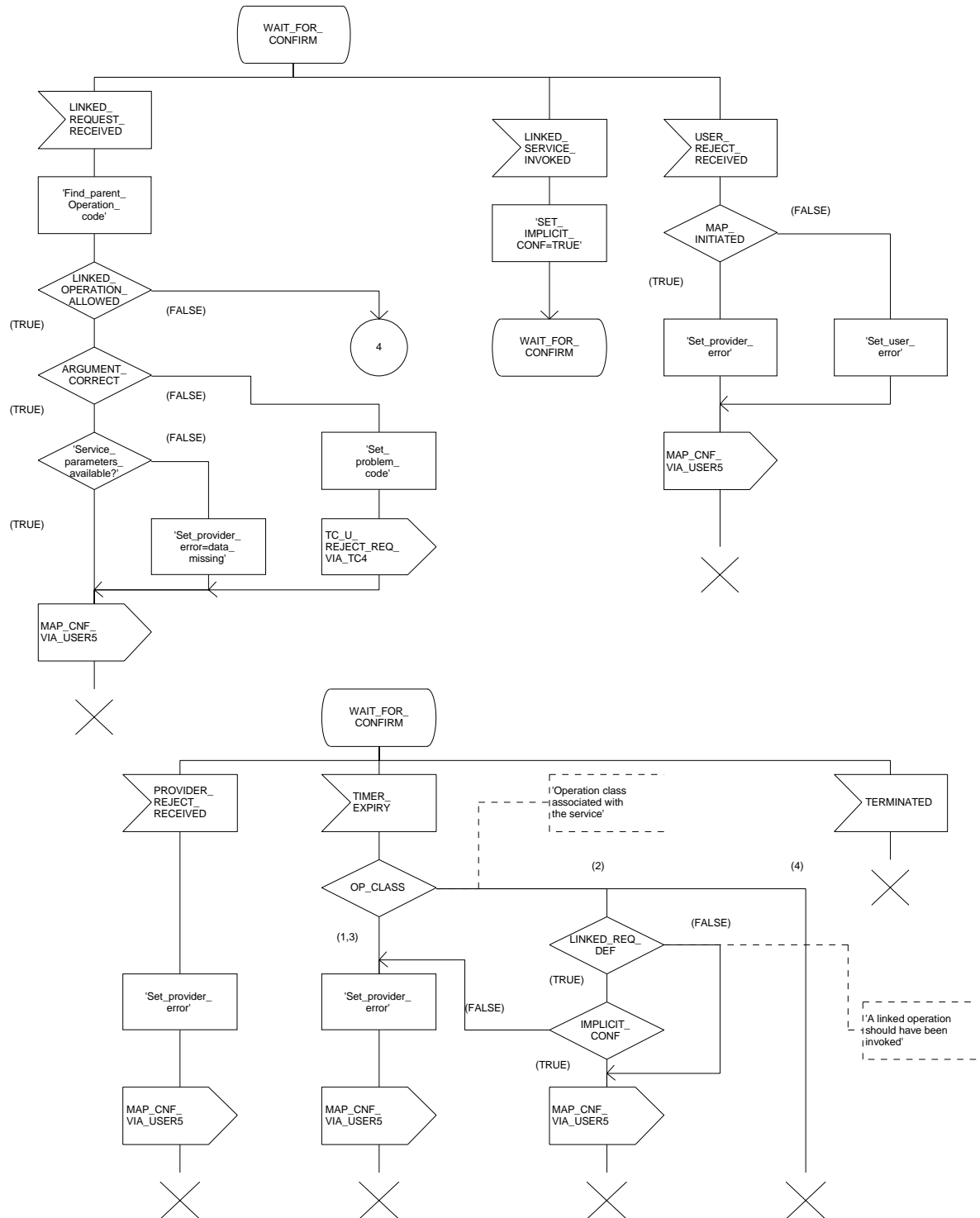


Figure 13.2/7 (sheet 4 of 4): Process REQUESTING_MAP_SSM



14 Abstract syntax of the MAP protocol

14.1 General

This subclause specifies the Abstract Syntaxes for the Mobile Application Part as well as the associated set of Operations and Errors, using the Abstract Syntax Notation One (ASN.1), defined in CCITT Recommendation X.208 (1988) or X.680 (1994) with additions as defined in subclause 14.1.4 on Compatibility Considerations and the OPERATION and ERROR external MACROs, defined in CCITT Recommendation Q.773.

The Abstract Syntax is defined for all interfaces specified in subclause 2.4 except for the A- and B-interfaces.

The Mobile Application Part protocol is defined by two Abstract Syntaxes:

- one Abstract Syntax which encompass all Operations; and
- Errors identified by the various MAP subsystem numbers.

This Abstract Syntax represents the set of values each of which is a value of the ASN.1 type TCAPMessages. MessageType as defined in CCITT Recommendation Q.773 with the ANY DEFINED BY sections resolved by the operation and error codes included in the ASN.1 module MAP-Protocol. However, only the subset of this abstract syntax which is required by the procedures defined for an entity needs to be supported:

- one Abstract Syntax identified by the OBJECT IDENTIFIER value MAP-DialogueInformation.map-DialogueAS.

This Abstract Syntax represents the set of values each of which is a value of the ASN.1 type MAP-DialogueInformation.MAP-DialoguePDU. Such a value of the ASN.1 single-ASN.1-type element is contained within the user-information element of the TCAPMessages.DialoguePortion ASN.1 type. This Abstract Syntax name is to be used as a direct reference.

14.1.1 Encoding rules

The encoding rules which are applicable to the defined Abstract Syntaxes are the Basic Encoding Rules for Abstract Syntax Notation One, defined in CCITT Recommendation X.690 with the same exceptions as in CCITT Recommendation Q.773 section 4 Message Representation.

When the definite form is used for length encoding, a data value of length less than 128 octets must have the length encoded in the short form.

When the long form is employed to code a length, the minimum number of octets shall be used to code the length field.

OCTET STRING values and BIT STRING values must be encoded in a primitive form.

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.

14.1.2 Use of TC

The mapping of OPERATION and ERROR to TC components is defined in ETS 300 287 (version 2) which is based on CCITT Recommendation Q.773 (1992).

NOTE 1: The class of an operation is not stated explicitly but is specified as well in the ASN.1 operation type definition.

Class 1: RESULT and ERROR appear in ASN.1 operation type definition.

Class 2: only ERROR appears in ASN.1 operation type definition.

Class 3: only RESULT appears in ASN.1 operation type definition.

Class 4: both RESULT and ERROR do not appear in ASN.1 operation type definition.

The ASN.1 data type which follows the keywords "ARGUMENT", "PARAMETER" or "RESULT" (for OPERATION and ERROR) is always optional from a syntactic point of view. However, except when specifically mentioned 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 comment. The Timer Value Ranges are:

s	= from 3 seconds to 10 seconds;
m	= from 15 seconds to 30 seconds;
ml	= from 1 minute to 10 minutes;
l	= from 28 hours to 38 hours.

14.1.2.1 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 GSM 09.02, encoded as either an Object Identifier (as defined in CCITT Recommendation X.690 (1994)) or an integer value (as defined in section 14.5). 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 GSM 09.02
- 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 section 12.1.1

14.1.3 Use of information elements defined outside MAP

An information element or a set of information elements (messages) transparently carried in the Mobile Application Part but defined in other recommendation/technical specifications are handled in one of the following ways:

- i) The contents of each information element (without the octets encoding the identifier and the length in the recommendation/technical specification where it is defined) is carried as the value of an ASN.1 NamedType derived from the OCTET STRING data type. Additionally, the internal structure may be explained by means of comments. In case of misalignment the referred to recommendation/technical specification takes precedence.
- ii) The complete information element (including the octets encoding the identifier and the length in the recommendation/technical specification where it is defined) or set of information elements and the identity of the associated protocol are carried as the value of the ExternalSignalInfo data type defined in this ETS. Where more than one information element is carried, the information elements are sent contiguously with no filler octets between them.

14.1.4 Compatibility considerations

The following ASN.1 modules 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 subclause 14.1.1 apply to every element of the whole Transfer Syntax especially to the ASN.1 type EXTERNAL.

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.

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.

14.1.5 Structure of the Abstract Syntax of MAP

For each MAP parameter which has to be transferred by a MAP Protocol Data Unit (MAP message), there is a PDU field (an ASN.1 NamedType) whose ASN.1 identifier has the same name as the corresponding parameter, except for the differences required by the ASN.1 notation (blanks between words are removed or replaced by hyphen, the first letter of the first word is lower-case and the first letter of the following words are capitalized, e.g. "no reply condition time" is mapped to "noReplyConditionTime"). Additionally some words may be abbreviated as follows:

bs	basic service
ch	call handling
cug	closed user group
ho	handover
ic	incoming call
id	identity
info	information
ms	mobile service
oc	outgoing call
om	operation & maintenance
pw	Password
sm	short message service
ss	supplementary service

The MAP protocol is composed of several ASN.1 modules dealing with either operations, errors, data types, and, if applicable, split into those dealing with mobile services, call handling services, supplementary services and short message services. For operations and errors no values are assigned, but only the operation and error types in order to allow use of the defined types also by other protocols (e.g. TS GSM 04.80). The values (operation codes and error codes) are defined in a separate module. The ASN.1 source lines are preceded by line-numbers at the left margin in order to enable the usage of the cross-reference in appendix A.

The module containing the definition of the operation packages for MAP is:

1. MAP-OperationPackages.

The module containing the definition of the application contexts for MAP is:

2. MAP-ApplicationContexts.

The module containing the data types for the Abstract Syntax to be used for TCAPMessages.DialoguePortion for MAP is:

3. MAP-DialogueInformation.

The module containing the operation codes and error codes for MAP is:

4. MAP-Protocol.

The modules containing all operation type definitions for MAP are:

5. MAP-MobileServiceOperations;
6. MAP-OperationAndMaintenanceOperations;
7. MAP-CallHandlingOperations;
8. MAP-SupplementaryServiceOperations;
9. MAP-ShortMessageServiceOperations.

The module containing all error type definitions for MAP is:

10. MAP-Errors.

Modules containing all data type definitions for MAP are:

11. MAP-MS-DataTypes;
12. MAP-OM-DataTypes;
13. MAP-CH-DataTypes;
14. MAP-SS-DataTypes;
15. MAP-SS-Code;
16. MAP-SM-DataTypes;
17. MAP-ER-DataTypes;
18. MAP-CommonDataTypes;
19. MAP-TS-Code;
20. MAP-BS-Code;
21. MAP-ExtensionDataTypes.

References are made also to modules defined outside of this ETS. They are defined in the technical specification Mobile Services Domain and technical specification Transaction Capability respectively:

MobileDomainDefinitions;
TCAPMessages;
DialoguePDUs.

14.1.6 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 sections 14.6 & 14.7 relates only to the ACs in this table.

AC Name	AC Version	Operations Used	Comments *
locationCancellationContext	v2	cancelLocation	
equipmentMngtContext	v2	checkIMEI	
imsiRetrievalContext	v2	sendIMSI	
infoRetrievalContext	v2	sendAuthenticationInfo	
interVlrInfoRetrievalContext	v2	sendIdentification	
handoverControlContext	v2	prepareHandover forwardAccessSignalling sendEndSignal processAccessSignalling prepareSubsequentHandover	
mwdMngtContext	v2	readyForSM	
msPurgingContext	v2	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	
networkLocUpContext	v3	updateLocation forwardCheckSs-Indication restoreData insertSubscriberData activateTraceMode	the syntax is the same in v1 & v2
subscriberDataMngtContext	v3	insertSubscriberData deleteSubscriberData	
roamingNumberEnquiryContext	v3	provideRoamingNumber	
locationInfoRetrievalContext	v3	sendRoutingInfo	
callControlTransferContext	v3	resumeCallHandling	
subscriberInfoEnquiryContext	v3	provideSubscriberInfo	
anyTimeEnquiryContext	v3	anyTimeInterrogation	

NOTE (*): The syntax of the operations is not the same in previous versions unless explicitly stated

14.2 Operation packages

14.2.1 General aspects

This subclause describes the operation-packages which are used to build the application-contexts defined in subclause 14.3.

Each operation-package is a specification of the roles of a pair of communicating objects (i.e. a pair of MAP-Providers), in term of operations which they can invoke of each other.

The grouping of operations into one or several packages does not necessarily imply any grouping in term of Application Service Elements.

The following ASN.1 MACRO is used to describe operation-packages in this subclause:

```
OPERATION-PACKAGE MACRO ::=
BEGIN
TYPE NOTATION ::= Symmetric | ConsumerInvokes SupplierInvokes |
empty
VALUE NOTATION ::= value(VALUE OBJECT IDENTIFIER)
Symmetric ::= "OPERATIONS" "{" OperationList "}"
ConsumerInvokes ::= "CONSUMER INVOKES" "{" OperationList "}"
SupplierInvokes ::= "SUPPLIER INVOKES" "{" OperationList "}" | empty
OperationList ::= Operation | OperationList "," Operation
Operation ::= value(OPERATION)
END
```

Since the application-context definitions provided in subclause 14.3 use only an informal description technique, only the type notation is used in the following subclauses to define operation-packages.

The following definitions are used throughout this subclause ($n \geq 2$):

- 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 $n \geq 2$.

For each operation package which is not vn-only ($n \geq 2$) and which does not include only v(n-1)-operations, there is a v(n-1)-equivalent package. Except when a definition is explicitly provided in the following subclauses, the v(n-1)-equivalent package includes the v(n-1)-equivalent operations of the operations which belong to this package.

14.2.2 Packages specifications

14.2.2.1 Location updating

This operation package includes the operations required for location management procedures between HLR and VLR.

```
LocationUpdatingPackage-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 subclause 14.2.1.

14.2.2.2 Location cancellation

This operation package includes the operations required for location cancellation and MS purging procedures between HLR and VLR.

```
LocationCancellationPackage-v2 ::= OPERATION-PACKAGE
-- Supplier is VLR if Consumer is HLR
CONSUMER INVOKES {
    cancelLocation}
```

The v1-equivalent package can be determined according to the rules described in subclause 14.2.1.

14.2.2.3 Roaming number enquiry

This operation package includes the operations required for roaming number enquiry procedures between HLR and VLR.

```
RoamingNumberEnquiryPackage-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 subclause 14.2.1.

14.2.2.4 Information retrieval

This operation package includes the operation required for the authentication information retrieval procedure between HLR and VLR.

```
InfoRetrievalPackage-v2 ::= OPERATION-PACKAGE
-- Supplier is HLR if Consumer is VLR
CONSUMER INVOKES {
    sendAuthenticationInfo}
```

The v1-equivalent package is defined as follows:

```
InfoRetrievalPackage-v1 ::= OPERATION-PACKAGE
-- Supplier is HLR or VLR if Consumer is VLR
CONSUMER INVOKES {
    sendParameters}
```

14.2.2.5 Inter-VLR information retrieval

This operation package includes the operations required for inter VLR information retrieval procedures.

```
InterVlrInfoRetrievalPackage-v2 ::= OPERATION-PACKAGE
-- Supplier is VLR if Consumer is VLR
CONSUMER INVOKES {
    sendIdentification}
```


The v1-equivalent package is : InfoRetrievalPackage-v1

14.2.2.6 IMSI retrieval

This operation package includes the operation required for the IMSI retrieval procedure between HLR and VLR.

```
IMSIRetrievalPackage-v2 ::= OPERATION-PACKAGE
-- Supplier is HLR if Consumer is VLR
CONSUMER INVOKES {
    sendIMSI}
```

This package is v2 only.

14.2.2.7 Call control transfer

This operation package includes the operation required for the call control transfer procedure between VMSC and GMSC.

```
CallControlTransferPackage-v3 ::= OPERATION-PACKAGE
-- Supplier is GMSC if Consumer is VMSC
CONSUMER INVOKES {
    resumeCallHandling}
```

This package is v3 only.

14.2.2.8 - 14.2.2.9 [Spare]

14.2.2.10 Interrogation

This operation package includes the operations required for interrogation procedures between MSC and HLR.

```
InterrogationPackage-v3 ::= OPERATION-PACKAGE
-- Supplier is HLR if Consumer is MSC
CONSUMER INVOKES {
    sendRoutingInfo}
```

The v1-equivalent and v2-equivalent packages can be determined according to the rules described in subclause 14.2.1.

14.2.2.11 [Spare]

14.2.2.12 Handover Control

This operation package includes the operations required for handover procedures between MSCs.

```
HandoverControlPackage-v2 ::= OPERATION-PACKAGE
-- Supplier is MSCB if Consumer is MSCA
CONSUMER INVOKES {
    prepareHandover,
    forwardAccessSignalling}
SUPPLIER INVOKES {
    sendEndSignal,
    processAccessSignalling,
    prepareSubsequentHandover}
```

The v1-equivalent package is defined as follows.

```
HandoverControlPackage-v1 ::= OPERATION-PACKAGE
-- Supplier is MSCB if Consumer is MSCA
CONSUMER INVOKES {
    performHandover,
    forwardAccessSignalling,
    traceSubscriberActivity}
SUPPLIER INVOKES {
    sendEndSignal,
    noteInternalHandover,
    processAccessSignalling,
    performSubsequentHandover}
```

14.2.2.13 Subscriber Data management stand alone

This operation package includes the operations required for stand alone subscriber data management procedures between HLR and VLR.

```
SubscriberDataMngtStandAlonePackage-v3 ::= OPERATION-PACKAGE
-- Supplier is VLR if Consumer is HLR
CONSUMER INVOKES {
    insertSubscriberData,
    deleteSubscriberData}
```

The v1-equivalent and v2-equivalent packages can be determined according to the rules described in subclause 14.2.1.

14.2.2.14 Equipment management

This operation package includes the operations required for equipment management procedures between EIR and MSC.

```
EquipmentMngtPackage-v2 ::= OPERATION-PACKAGE
-- Supplier is EIR if Consumer is MSC
CONSUMER INVOKES {
    checkIMEI}
```

The v1-equivalent package can be determined according to the rules described in subclause 14.2.1.

14.2.2.15 Subscriber data management

This operation package includes the operations required for subscriber data management procedures between HLR and VLR.

```
SubscriberDataMngtPackage-v3 ::= OPERATION-PACKAGE
-- Supplier is VLR if Consumer is HLR
CONSUMER INVOKES {
    insertSubscriberData}
```

The v1-equivalent and v2-equivalent packages can be determined according to the rules described in subclause 14.2.1.

14.2.2.16 Location register restart

This operation package includes the operations required for location register restart procedures between HLR and VLR.

```
ResetPackage-v2 ::= OPERATION-PACKAGE
-- Supplier is VLR if Consumer is HLR
CONSUMER INVOKES {
    reset}
```

The v1-equivalent package can be determined according to the rules described in subclause 14.2.1.

14.2.2.17 Tracing stand-alone

This operation package includes the operations required for stand alone tracing procedures between HLR and VLR.

```
TracingStandAlonePackage-v3 ::= OPERATION-PACKAGE
-- Supplier is VLR if Consumer is HLR
CONSUMER INVOKES {
    activateTraceMode,
    deactivateTraceMode}
```

The v1-equivalent and v2-equivalent packages can be determined according to the rules described in subclause 14.2.1.

14.2.2.18 Functional SS handling

This operation package includes the operations required for functional supplementary services procedures between VLR and HLR.

```
FunctionalSsPackage-v2 ::= OPERATION-PACKAGE
-- Supplier is HLR if Consumer is VLR
CONSUMER INVOKES {
    registerSS,
    eraseSS,
    activateSS,
    deactivateSS,
    registerPassword,
    interrogateSS}
SUPPLIER INVOKES {
    getPassword}
```

The v1-equivalent package can be determined according to the rules described in subclause 14.2.1.

14.2.2.19 Tracing

This operation package includes the operations required for tracing procedures between HLR and VLR.

```
TracingPackage-v3 ::= OPERATION-PACKAGE
-- Supplier is VLR if Consumer is HLR
CONSUMER INVOKES {
    activateTraceMode}
```

The v1-equivalent and v2-equivalent packages can be determined according to the rules described in subclause 14.2.1.

14.2.2.20 Binding

This operation package includes the operation required to initialize a supplementary service procedure between VLR and HLR.

```
BindingPackage-v1 ::= OPERATION-PACKAGE
-- Supplier is HLR if Consumer is VLR
CONSUMER INVOKES {
    beginSubscriberActivity}
```

This package is v1 only.

14.2.2.21 Unstructured SS handling

This operation package includes the operations required for unstructured supplementary services procedures between VLR and HLR.

```
UnstructuredSsPackage-v2 ::= OPERATION-PACKAGE
-- Supplier is HLR if Consumer is VLR
CONSUMER INVOKES {
    processUnstructuredSS-Request}
SUPPLIER INVOKES {
    unstructuredSS-Request,
    unstructuredSS-Notify}
```

The v1-equivalent package is defined as follows:

```
UnstructuredSsPackage-v1 ::= OPERATION-PACKAGE
-- Supplier is HLR if Consumer is VLR
CONSUMER INVOKES {
    processUnstructuredSS-Data}
```

14.2.2.22 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.

```
MOShortMsgRelayPackage-v3 ::= OPERATION-PACKAGE
-- Supplier is IWMSC if Consumer is MSC
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 if Consumer is GMSC
CONSUMER INVOKES {
    forwardSM}
```

The v1-equivalent package can be determined according to the rules described in subclause 14.2.1.

14.2.2.23 Short message gateway services

This operation package includes the operations required for short message service gateway procedures between MSC and HLR.

```
ShortMsgGatewayPackage-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 subclause 14.2.1

The v1-equivalent package is defined as follows:

```
ShortMsgGatewayPackage-v1 ::= OPERATION-PACKAGE
-- Supplier is HLR if Consumer is GMSC
CONSUMER INVOKES {
    sendRoutingInfoForSM
    reportSMDeliveryStatus}
```

14.2.2.24 MT Short message relay services

This operation package includes the operations required for short message relay service procedures between GMSC and MSC.

```
MTShortMsgRelayPackage-v3 ::= OPERATION-PACKAGE
-- Supplier is MSC if Consumer is GMSC
CONSUMER INVOKES {
    MT-forwardSM}
```

The v2-equivalent package is: **ShortMsgRelayPackage-v2**

14.2.2.25 [Spare]

14.2.2.26 Message waiting data management

This operation package includes the operations required for short message waiting data procedures between HLR and VLR.

```
MwdMngtPackage-v2 ::= OPERATION-PACKAGE
-- Supplier is HLR if Consumer is VLR
CONSUMER INVOKES {
    readyForSM}
```

The v1-equivalent package is defined as follows:

```
MwdMngtPackage-v1 ::= OPERATION-PACKAGE
-- Supplier is HLR if Consumer is VLR
CONSUMER INVOKES {
    noteSubscriberPresent}
```

14.2.2.27 Alerting

This operation package includes the operations required for alerting between HLR and IWMSC.

```
AlertingPackage-v2 ::= OPERATION-PACKAGE
-- Supplier is IWMSC if Consumer is HLR
CONSUMER INVOKES {
    alertServiceCentre}
```

The v1-equivalent package is defined as follows.

```
AlertingPackage-v1 ::= OPERATION-PACKAGE
-- Supplier is IWMSC if Consumer is HLR
CONSUMER INVOKES {
    alertServiceCentreWithoutResult}
```

14.2.2.28 Data restoration

This operation package includes the operations required for VLR data restoration between HLR and VLR.

```
DataRestorationPackage-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 subclause 14.2.1.

The v1-equivalent package is: InfoRetrievalPackage-v1

14.2.2.29 Purging

This operation package includes the operations required for purging between HLR and VLR.

```
PurgingPackage-v2 ::= OPERATION-PACKAGE
-- Supplier is HLR if Consumer is VLR
CONSUMER INVOKES {
    purgeMS}
```

This Package is v2 only.

14.2.2.30 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.

14.2.2.31 Any time information enquiry

This operation package includes the operations required for any time information enquiry procedures between gsmSCF and HLR.

<pre>AnyTimeInformationEnquiryPackage-v3 ::= OPERATION-PACKAGE -- Supplier is HLR if Consumer is gsmSCF CONSUMER INVOKES { anyTimeInterrogation}</pre>

This package is v3 only.

14.3 Application contexts

14.3.1 General aspects

An application-context is assigned for each dialogue established by a MAP-user. In this ETS each application-context is assigned a name which is supplied in the MAP-OPEN Req primitive by the MAP-User and transmitted to the peer under certain circumstances.

The following ASN.1 MACRO is used to describe the main aspects of application-contexts in the following subclauses:

```
APPLICATION-CONTEXT MACRO ::=
BEGIN
TYPE NOTATION ::= Symmetric | InitiatorConsumerOf
ResponderConsumerOf | empty
VALUE NOTATION ::= value(VALUE OBJECT IDENTIFIER)
Symmetric ::= "OPERATIONS OF" "{" PackageList "}"
InitiatorConsumerOf ::= "INITIATOR CONSUMER OF" "{" PackageList "}"
ResponderConsumerOf ::= "RESPONDER CONSUMER OF" "{" PackageList "}"
| empty
PackageList ::= Package | PackageList "," Package
Package ::= value(OPERATION-PACKAGE)
| type -- shall reference a package type
END
```

The following definitions are used throughout this subclause:

- v1-application-context: An application-context which contains only v1-packages and uses only TC v1 facilities;
- v1 context set: the set of v1-application-contexts defined in this ETS.
- vn-application-context ($n \geq 2$): 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 $n \geq 2$.

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 subclause 14.1.1.

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 subclause 14.4.

14.3.2 Application context definitions**14.3.2.1 [Spare]****14.3.2.2 Location Updating**

This application context is used between HLR and VLR for location updating procedures.

```
networkLocUpContext-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)}
```

14.3.2.3 Location Cancellation

This application context is used between HLR and VLR for location cancellation procedures.

```
locationCancellationContext-v2 APPLICATION-CONTEXT
-- Responder is VLR if Initiator is HLR
INITIATOR CONSUMER OF {
    LocationCancellationPackage-v2}
::= {map-ac locationCancel(2) version2(2)}
```

The following application-context-name is assigned to the v1-equivalent application-context:

```
map-ac locationCancel(2) version1(1)
```

14.3.2.4 Roaming number enquiry

This application context is used between HLR and VLR for roaming number enquiry procedures.

```
roamingNumberEnquiryContext-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)}
```


14.3.2.5 [Spare]

14.3.2.6 Location Information Retrieval

This application-context is used between GMSC and HLR when retrieving location information.

```
locationInfoRetrievalContext-v3 APPLICATION-CONTEXT
-- Responder is HLR 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)}
```

14.3.2.7 Call control transfer

This application context is used for the call control transfer procedure between the VMSC and the GMSC.

```
callControlTransferContext-v3 APPLICATION-CONTEXT
-- Responder is GMSC if Initiator is VMSC
INITIATOR CONSUMER OF {
    CallControlTransferPackage-v3}
::= {map-ac callControlTransfer(6) version3(3)}
```

This application-context is v3 only.

14.3.2.8 - 14.3.2.10 [Spare]

14.3.2.11 Location registers restart

This application context is used for location register restart procedures.

```
resetContext-v2 APPLICATION-CONTEXT
-- Responder is VLR if Initiator is HLR
INITIATOR CONSUMER OF {
    ResetPackage-v2}
::= {map-ac reset(10) version2(2)}
```

The following application-context-name is assigned to the v1-equivalent application-context:

```
{map-ac reset(10) version1(1)}
```

14.3.2.12 Handover control

This application context is used for handover procedures between MSCs.

```
handoverControlContext-v2 APPLICATION-CONTEXT
-- Responder is MSCB if Initiator is MSCA
INITIATOR CONSUMER OF {
    HandoverControlPackage-v2}
::= {map-ac handoverControl(11) version2(2)}
```

The following application-context-name is assigned to the v1-equivalent application-context:

```
{map-ac handoverControl(11) version1(1)}
```

14.3.2.13 IMSI Retrieval

This application context is used for IMSI retrieval between HLR and VLR.

```
imsiRetrievalContext-v2 APPLICATION-CONTEXT
-- Responder is HLR if Initiator is VLR
INITIATOR CONSUMER OF {
    IMSIRetrievalPackage-v2}
::= {map-ac imsiRetrieval(26) version2(2)}
```

This application-context is v2 only.

14.3.2.14 Equipment Management

This application context is used for equipment checking between MSC and EIR:

```
equipmentMngtContext-v2 APPLICATION-CONTEXT
-- Responder is EIR if Initiator is MSC
INITIATOR CONSUMER OF {
    EquipmentMngtPackage-v2}
::= {map-ac equipmentMngt(13) version2(2)}
```

The following application-context-name is assigned to the v1-equivalent application-context:

```
{map-ac equipmentMngt(13) version1(1)}
```

14.3.2.15 Information retrieval

This application context is used for authentication information retrieval between HLR and VLR.

```
infoRetrievalContext-v2 APPLICATION-CONTEXT
-- Responder is HLR if Initiator is VLR
INITIATOR CONSUMER OF {
    InfoRetrievalPackage-v2}
::= {map-ac infoRetrieval(14) version2(2)}
```

The following application-context-name is assigned to the v1-equivalent application-context:

```
-- Responder is HLR if Initiator is VLR
{map-ac infoRetrieval(14) version1(1)}
```

14.3.2.16 Inter-VLR information retrieval

This application context is used for information retrieval between VLRs.

```
interVlrInfoRetrievalContext-v2 APPLICATION-CONTEXT
-- Responder is VLR if Initiator is VLR
INITIATOR CONSUMER OF {
    InterVlrInfoRetrievalPackage-v2}
::= {map-ac interVlrInfoRetrieval(15) version2(2)}
```

The v1-equivalent application-context is:

```
-- Responder is VLR if Initiator is VLR
{map-ac infoRetrieval(14) version1(1)}
```

14.3.2.17 Stand Alone Subscriber Data Management

This application context is used for stand alone subscriber data management between HLR and VLR:

```
subscriberDataMngtContext-v3 APPLICATION-CONTEXT
-- Responder is VLR 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)}
```

14.3.2.18 Tracing

This application context is used for stand alone tracing control procedures:

```
tracingContext-v3 APPLICATION-CONTEXT
-- Responder is VLR 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)}
```

14.3.2.19 Network functional SS handling

This application context is used for functional-like SS handling procedures between VLR and HLR.

```
networkFunctionalSsContext-v2 APPLICATION-CONTEXT
-- Responder is HLR, Initiator is VLR
INITIATOR CONSUMER OF {
    FunctionalSsPackage-v2}
::= {map-ac networkFunctionalSs(18) version2(2)}
```

The v1-equivalent application-context is defined as follows:

```
networkFunctionalSsContext-v1 APPLICATION-CONTEXT
-- Responder is HLR, Initiator is VLR
INITIATOR CONSUMER OF {
    FunctionalSsPackage-v1,
    UnstructuredSsPackage-v1,
    BindingPackage-v1}
::= {map-ac networkFunctionalSs(18) version1(1)}
```

14.3.2.20 Network unstructured SS handling

This application context is used for handling stimuli-like procedures between HLR and VLR.

```
networkUnstructuredSsContext-v2 APPLICATION-CONTEXT
-- Responder is HLR, Initiator is VLR
-- Responder is VLR, Initiator is HLR
OPERATIONS OF {
    UnstructuredSsPackage-v2}
::= {map-ac networkUnstructuredSs(19) version2(2)}
```

The following application-context-name is assigned to the v1-equivalent application-context:

```
{map-ac networkFunctionalSs(18) version1(1)}
```

14.3.2.21 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)}
```

14.3.2.22 Mobile originating Short Message Relay

This application context is used for mobile originating short message relay procedures.

```
shortMsgMO-RelayContext-v3 APPLICATION-CONTEXT
-- Responder is IWMSC if Initiator is MSC
INITIATOR CONSUMER OF {
    MShortMsgRelayPackage-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)}
```

14.3.2.23 [Spare]

14.3.2.24 Short message alert

This application context is used for short message alerting procedures.

```
shortMsgAlertContext-v2 APPLICATION-CONTEXT
-- Responder is IWMSC if Initiator is HLR
INITIATOR CONSUMER OF {
    AlertingPackage-v2}
::= {map-ac shortMsgAlert(23) version2(2)}
```

The following application-context-name is symbolically assigned to the v1-equivalent application-context:

```
{map-ac shortMsgAlert(23) version1(1)}
```

14.3.2.25 Short message waiting data management

This application context is used for short message waiting data management procedures.

```
mwdMngtContext-v2 APPLICATION-CONTEXT
-- Responder is HLR if Initiator is VLR
INITIATOR CONSUMER OF {
    MwdMngtPackage-v2}
::= {map-ac mwdMngt(24) version2(2)}
```

The following application-context-name is assigned to the v1-equivalent application-context:

```
{map-ac mwdMngt(24) version1(1)}
```

14.3.2.26 Mobile terminating Short Message Relay

This application context is used for mobile terminating short message relay procedures.

```
shortMsgMT-RelayContext-v3 APPLICATION-CONTEXT
-- Responder is MSC 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)}
```

14.3.2.27 MS purging

This application context is used between HLR and VLR for MS purging procedures.

```
msPurgingContext-v2 APPLICATION-CONTEXT
-- Responder is HLR if Initiator is VLR
INITIATOR CONSUMER OF {
    purgingPackage-v2}
::= {map-ac msPurging(27) version2(2)}
```

This application-context is v2 only.

14.3.2.28 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.

14.3.2.29 Any time information enquiry

This application context is used between gsmSCF and HLR for any time information enquiry procedures.

```
anyTimeInfoEnquiryContext-v3 APPLICATION-CONTEXT
-- Responder is HLR if Initiator is gsmSCF
INITIATOR CONSUMER OF {
    AnyTimeInformationEnquiryPackage-v3}
::= {map-ac anyTimeInfoEnquiry(29) version3(3)}
```

This application-context is v3 only.

14.3.3 ASN.1 Module for application-context-names

The following ASN.1 module summarizes the application-context-name assigned to MAP application-contexts.

```

1 MAP-ApplicationContexts {
2   ccitt identified-organization (4) etsi (0) mobileDomain (0)
3   gsm-Network (1) modules (3) map-ApplicationContexts (2) version3 (3)}
4
5 DEFINITIONS
6 ::=
7
8 BEGIN
9
10
11
12 -- EXPORTS everything
13
14
15 IMPORTS
16   gsm-NetworkId,
17   ac-Id
18 FROM MobileDomainDefinitions {
19   ccitt (0) identified-organization (4) etsi (0) mobileDomain (0)
20   mobileDomainDefinitions (0) version1 (1)}
21 ;
22
23 -- application-context-names
24
25 map-ac OBJECT IDENTIFIER ::= {gsm-NetworkId ac-Id}
26
27 networkLocUpContext-v3 OBJECT IDENTIFIER ::=
28   {map-ac networkLocUp(1) version3(3)}
29
30 locationCancellationContext-v2 OBJECT IDENTIFIER ::=
31   {map-ac locationCancel(2) version2(2)}
32
33 roamingNumberEnquiryContext-v3 OBJECT IDENTIFIER ::=
34   {map-ac roamingNbEnquiry(3) version3(3)}
35
36 locationInfoRetrievalContext-v3 OBJECT IDENTIFIER ::=
37   {map-ac locInfoRetrieval(5) version3(3)}
38
39 resetContext-v2 OBJECT IDENTIFIER ::=
40   {map-ac reset(10) version2(2)}
41
42 handoverControlContext-v2 OBJECT IDENTIFIER ::=
43   {map-ac handoverControl(11) version2(2)}
44
45 equipmentMngtContext-v2 OBJECT IDENTIFIER ::=
46   {map-ac equipmentMngt(13) version2(2)}
47
48 infoRetrievalContext-v2 OBJECT IDENTIFIER ::=
49   {map-ac infoRetrieval(14) version2(2)}
50
51 interVlrInfoRetrievalContext-v2 OBJECT IDENTIFIER ::=
52   {map-ac interVlrInfoRetrieval(15) version2(2)}
53
54 subscriberDataMngtContext-v3 OBJECT IDENTIFIER ::=
55   {map-ac subscriberDataMngt(16) version3(3)}
56
57 tracingContext-v3 OBJECT IDENTIFIER ::=
58   {map-ac tracing(17) version3(3)}
59
60 networkFunctionalSsContext-v2 OBJECT IDENTIFIER ::=
61   {map-ac networkFunctionalSs(18) version2(2)}
62
63 networkUnstructuredSsContext-v2 OBJECT IDENTIFIER ::=
64   {map-ac networkUnstructuredSs(19) version2(2)}
65
66 shortMsgGatewayContext-v3 OBJECT IDENTIFIER ::=
67   {map-ac shortMsgGateway(20) version3(3)}
68
69 shortMsgMO-RelayContext-v3 OBJECT IDENTIFIER ::=
70   {map-ac shortMsgMO-Relay(21) version3(3)}
71

```

```

72 shortMsgAlertContext-v2 OBJECT IDENTIFIER ::=
73     {map-ac shortMsgAlert(23) version2(2)}
74
75 mwdMngtContext-v2 OBJECT IDENTIFIER ::=
76     {map-ac mwdMngt(24) version2(2)}
77
78 shortMsgMT-RelayContext-v3 OBJECT IDENTIFIER ::=
79     {map-ac shortMsgMT-Relay(25) version3(3)}
80
81 imsiRetrievalContext-v2 OBJECT IDENTIFIER ::=
82     {map-ac imsiRetrieval(26) version2(2)}
83
84 msPurgingContext-v2 OBJECT IDENTIFIER ::=
85     {map-ac msPurging(27) version2(2)}
86
87 subscriberInfoEnquiryContext-v3 OBJECT IDENTIFIER ::=
88     {map-ac subscriberInfoEnquiry(28) version3(3)}
89
90 anyTimeInfoEnquiryContext-v3 OBJECT IDENTIFIER ::=
91     {map-ac anyTimeInfoEnquiry(29) version3(3)}
92
93 callControlTransferContext-v3 OBJECT IDENTIFIER ::=
94     {map-ac callControlTransfer(6) version3(3)}
95
96 -- The following Object Identifiers are reserved for application-
97 -- contexts existing in previous versions of the protocol
98
99 -- AC Name & Version                                Object Identifier
100 --
101 -- networkLocUpContext-v1                            map-ac networkLocUp (1)          version1 (1)
102 -- networkLocUpContext-v2                            map-ac networkLocUp (1)          version2 (2)
103 -- locationCancellationContext-v1                    map-ac locationCancellation (2)   version1 (1)
104 -- roamingNumberEnquiryContext-v1                    map-ac roamingNumberEnquiry (3)   version1 (1)
105 -- roamingNumberEnquiryContext-v2                    map-ac roamingNumberEnquiry (3)   version2 (2)
106 -- locationInfoRetrievalContext-v1                   map-ac locationInfoRetrieval (5)   version1 (1)
107 -- locationInfoRetrievalContext-v2                   map-ac locationInfoRetrieval (5)   version2 (2)
108 -- resetContext-v1                                   map-ac reset (10)                version1 (1)
109 -- handoverControlContext-v1                         map-ac handoverControl (11)       version1 (1)
110 -- equipmentMngtContext-v1                           map-ac equipmentMngt (13)         version1 (1)
111 -- infoRetrievalContext-v1                           map-ac infoRetrieval (14)         version1 (1)
112 -- subscriberDataMngtContext-v1                     map-ac subscriberDataMngt (16)     version1 (1)
113 -- subscriberDataMngtContext-v2                     map-ac subscriberDataMngt (16)     version2 (2)
114 -- tracingContext-v1                                 map-ac tracing (17)              version1 (1)
115 -- tracingContext-v2                                 map-ac tracing (17)              version2 (2)
116 -- networkFunctionalSsContext-v1                     map-ac networkFunctionalSs (18)    version1 (1)
117 -- shortMsgGatewayContext-v1                        map-ac shortMsgGateway (20)       version1 (1)
118 -- shortMsgGatewayContext-v2                        map-ac shortMsgGateway (20)       version2 (2)
119 -- shortMsgRelayContext-v1                          map-ac shortMsgRelay (21)         version1 (1)
120 -- shortMsgAlertContext-v1                          map-ac shortMsgAlert (23)         version1 (1)
121 -- mwdMngtContext-v1                                map-ac mwdMngt (24)              version1 (1)
122 -- shortMsgMT-RelayContext-v2                       map-ac shortMsgMT-Relay (25)      version2 (2)
123
124
125 END

```

1 14.4 MAP Dialogue Information

```

2 MAP-DialogueInformation {
3     ccitt identified-organization (4) etsi (0) mobileDomain (0)
4     gsm-Network (1) modules (3) map-DialogueInformation (3) version3 (3)}
5
6 DEFINITIONS
7
8 IMPLICIT TAGS
9
10 ::=
11
12 BEGIN
13
14 EXPORTS
15     map-DialogueAS,
16     MAP-DialoguePDU
17 ;
18
19 IMPORTS
20     gsm-NetworkId,
21     as-Id
22 FROM MobileDomainDefinitions {
23     ccitt (0) identified-organization (4) etsi (0) mobileDomain (0)

```

```

24     mobileDomainDefinitions (0) version1 (1)}
25
26     AddressString
27 FROM MAP-CommonDataTypes {
28     ccitt identified-organization (4) etsi (0) mobileDomain (0)
29     gsm-Network(1) modules (3) map-CommonDataTypes (18) version3 (3)}
30
31     ExtensionContainer
32 FROM MAP-ExtensionDataTypes {
33     ccitt identified-organization (4) etsi (0) mobileDomain (0)
34     gsm-Network (1) modules (3) map-ExtensionDataTypes (21) version3 (3)}
35
36 ;
37
38
39 -- abstract syntax name for MAP-DialoguePDU
40

```

```

41 map-DialogueAS OBJECT IDENTIFIER ::=
42     {gsm-NetworkId as-Id map-DialoguePDU (1) version1 (1)}
43

```

```

44 MAP-DialoguePDU ::= CHOICE {
45     map-open                [0] MAP-OpenInfo,
46     map-accept              [1] MAP-AcceptInfo,
47     map-close               [2] MAP-CloseInfo,
48     map-refuse              [3] MAP-RefuseInfo,
49     map-userAbort           [4] MAP-UserAbortInfo,
50     map-providerAbort       [5] MAP-ProviderAbortInfo}
51

```

```

52 MAP-OpenInfo ::= SEQUENCE {
53     destinationReference    [0] AddressString        OPTIONAL,
54     originationReference    [1] AddressString        OPTIONAL,
55     ...,
56     extensionContainer      ExtensionContainer        OPTIONAL
57     -- extensionContainer must not be used in version 2
58 }
59

```

```

60 MAP-AcceptInfo ::= SEQUENCE {
61     ...,
62     extensionContainer      ExtensionContainer        OPTIONAL
63     -- extensionContainer must not be used in version 2
64 }
65

```

```

66 MAP-CloseInfo ::= SEQUENCE {
67     ...,
68     extensionContainer      ExtensionContainer        OPTIONAL
69     -- extensionContainer must not be used in version 2
70 }
71

```

```

72 MAP-RefuseInfo ::= SEQUENCE {
73     reason Reason,
74     ...,
75     extensionContainer      ExtensionContainer        OPTIONAL
76     -- extensionContainer must not be used in version 2
77 }
78

```

```

79 Reason ::= ENUMERATED {
80     noReasonGiven (0),
81     invalidDestinationReference (1),
82     invalidOriginatingReference (2)}
83

```

```

84 MAP-UserAbortInfo ::= SEQUENCE {
85     map-UserAbortChoice     MAP-UserAbortChoice,
86     ...,
87     extensionContainer      ExtensionContainer        OPTIONAL
88     -- extensionContainer must not be used in version 2
89 }
90

```

```

91 MAP-UserAbortChoice ::= CHOICE {
92     userSpecificReason      [0] NULL,
93     userResourceLimitation  [1] NULL,
94     resourceUnavailable     [2] ResourceUnavailableReason,
95     applicationProcedureCancellation [3] ProcedureCancellationReason}
96

```

```

97 ResourceUnavailableReason ::= ENUMERATED {
98     shortTermResourceLimitation (0),
99     longTermResourceLimitation (1)}
100

```



```
101 ProcedureCancellationReason ::= ENUMERATED {
102     handoverCancellation (0),
103     radioChannelRelease (1),
104     networkPathRelease (2),
105     callRelease (3),
106     associatedProcedureFailure (4),
107     tandemDialogueRelease (5),
108     remoteOperationsFailure (6)}
109
110 MAP-ProviderAbortInfo ::= SEQUENCE {
111     map-ProviderAbortReason      MAP-ProviderAbortReason,
112     ...,
113     extensionContainer           ExtensionContainer           OPTIONAL
114     -- extensionContainer must not be used in version 2
115     }
116
117 MAP-ProviderAbortReason ::= ENUMERATED {
118     abnormalDialogue (0),
119     invalidPDU (1)}
120
121 END
```

14.5 MAP operation and error codes

```
1 MAP-Protocol {
2   ccitt identified-organization (4) etsi (0) mobileDomain (0)
3   gsm-Network (1) modules (3) map-Protocol (4) version3 (3)}
4
5 DEFINITIONS
6
7 ::=
8
9 BEGIN
10
11 IMPORTS
12     UpdateLocation,
13     CancelLocation,
14     PurgeMS,
15     SendIdentification,
16     PrepareHandover,
17     SendEndSignal,
18     ProcessAccessSignalling,
19     ForwardAccessSignalling,
20     PrepareSubsequentHandover,
21     SendAuthenticationInfo,
22     CheckIMEI,
23     InsertSubscriberData,
24     DeleteSubscriberData,
25     Reset,
26     ForwardCheckSS-Indication,
27     RestoreData,
28     ProvideSubscriberInfo,
29     AnyTimeInterrogation
30
31 FROM MAP-MobileServiceOperations {
32   ccitt identified-organization (4) etsi (0) mobileDomain (0)
33   gsm-Network (1) modules (3) map-MobileServiceOperations (5)
34   version3 (3)}
35
36     ActivateTraceMode,
37     DeactivateTraceMode,
38     SendIMSI
39 FROM MAP-OperationAndMaintenanceOperations {
40   ccitt identified-organization (4) etsi (0) mobileDomain (0)
41   gsm-Network (1) modules (3) map-OperationAndMaintenanceOperations (6)
42   version3 (3)}
43
44     SendRoutingInfo,
45     ProvideRoamingNumber,
46     ResumeCallHandling
47 FROM MAP-CallHandlingOperations {
48   ccitt identified-organization (4) etsi (0) mobileDomain (0)
49   gsm-Network (1) modules (3) map-CallHandlingOperations (7)
50   version3 (3)}
51
52     RegisterSS,
53     EraseSS,
54     ActivateSS,
55     DeactivateSS,
56     InterrogateSS,
57     ProcessUnstructuredSS-Request,
58     UnstructuredSS-Request,
59     UnstructuredSS-Notify,
60     RegisterPassword,
61     GetPassword
62 FROM MAP-SupplementaryServiceOperations {
63   ccitt identified-organization (4) etsi (0) mobileDomain (0)
64   gsm-Network (1) modules (3) map-SupplementaryServiceOperations (8)
65   version3 (3)}
66
67     SendRoutingInfoForSM,
68     MO-ForwardSM,
69     MT-ForwardSM,
70     ReportSM-DeliveryStatus,
71     AlertServiceCentre,
72     InformServiceCentre,
73     ReadyForSM
74 FROM MAP-ShortMessageServiceOperations {
75   ccitt identified-organization (4) etsi (0) mobileDomain (0)
76   gsm-Network (1) modules (3) map-ShortMessageServiceOperations (9)
77   version3 (3)}
78
```

```

79      SystemFailure,
80      DataMissing,
81      UnexpectedDataValue,
82      FacilityNotSupported,
83      UnknownSubscriber,
84      NumberChanged,
85      UnknownMSC,
86      UnidentifiedSubscriber,
87      UnknownEquipment,
88      RoamingNotAllowed,
89      IllegalSubscriber,
90      IllegalEquipment,
91      BearerServiceNotProvisioned,
92      TeleserviceNotProvisioned,
93      NoHandoverNumberAvailable,
94      SubsequentHandoverFailure,
95      TracingBufferFull,
96      OR-NotAllowed,
97      NoRoamingNumberAvailable,
98      AbsentSubscriber,
99      BusySubscriber,
100     NoSubscriberReply,
101     CallBarred,
102     ForwardingViolation,
103     ForwardingFailed,
104     CUG-Reject,
105     ATI-NotAllowed,
106     IllegalSS-Operation,
107     SS-ErrorStatus,
108     SS-NotAvailable,
109     SS-SubscriptionViolation,
110     SS-Incompatibility,
111     UnknownAlphabet,
112     USSD-Busy,
113     PW-RegistrationFailure,
114     NegativePW-Check,
115     NumberOfPW-AttemptsViolation,
116     SubscriberBusyForMT-SMS,
117     SM-DeliveryFailure,
118     MessageWaitingListFull,
119     AbsentSubscriberSM
120 FROM MAP-Errors {
121     ccitt identified-organization (4) etsi (0) mobileDomain (0)
122     gsm-Network (1) modules (3) map-Errors (10) version3 (3)}
123 ;
124
125
126 -- location registration operation codes
127
128 updateLocation UpdateLocation ::= localValue 2
129 cancelLocation CancelLocation ::= localValue 3
130 purgeMS PurgeMS ::= localValue 67
131 sendIdentification SendIdentification ::= localValue 55
132
133
134 -- handover operation codes
135
136 prepareHandover PrepareHandover ::= localValue 68
137 sendEndSignal SendEndSignal ::= localValue 29
138 processAccessSignalling ProcessAccessSignalling ::= localValue 33
139 forwardAccessSignalling ForwardAccessSignalling ::= localValue 34
140 prepareSubsequentHandover PrepareSubsequentHandover ::=
141     localValue 69
142
143
144 -- authentication operation codes
145
146 sendAuthenticationInfo SendAuthenticationInfo ::= localValue 56
147
148
149 -- IMEI MANAGEMENT operation codes
150
151 checkIMEI CheckIMEI ::= localValue 43
152
153
154 -- subscriber management operation codes
155

```

```

156 insertSubscriberData InsertSubscriberData ::= localValue 7
157 deleteSubscriberData DeleteSubscriberData ::= localValue 8
158
159
160 -- fault recovery operation codes
161
162 reset Reset ::= localValue 37
163 forwardCheckSS-Indication ForwardCheckSS-Indication ::=
164     localValue 38
165 restoreData RestoreData ::= localValue 57
166
167
168 -- operation and maintenance operation codes
169
170 activateTraceMode ActivateTraceMode ::= localValue 50
171 deactivateTraceMode DeactivateTraceMode ::= localValue 51
172 sendIMSI SendIMSI ::= localValue 58
173
174
175 -- call handling operation codes
176
177 sendRoutingInfo SendRoutingInfo ::= localValue 22
178 provideRoamingNumber ProvideRoamingNumber ::= localValue 4
179 resumeCallHandling ResumeCallHandling ::= localValue 6
180
181
182 -- supplementary service handling operation codes
183
184 registerSS RegisterSS ::= localValue 10
185 eraseSS EraseSS ::= localValue 11
186 activateSS ActivateSS ::= localValue 12
187 deactivateSS DeactivateSS ::= localValue 13
188 interrogateSS InterrogateSS ::= localValue 14
189 processUnstructuredSS-Request ProcessUnstructuredSS-Request ::=
190     localValue 59
191 unstructuredSS-Request UnstructuredSS-Request ::= localValue 60
192 unstructuredSS-Notify UnstructuredSS-Notify ::= localValue 61
193 registerPassword RegisterPassword ::= localValue 17
194 getPassword GetPassword ::= localValue 18
195
196
197 -- short message service operation codes
198
199 sendRoutingInfoForSM SendRoutingInfoForSM ::= localValue 45
200 mo-forwardSM MO-ForwardSM ::= localValue 46
201 mt-forwardSM MT-ForwardSM ::= localValue 44
202 reportSM-DeliveryStatus ReportSM-DeliveryStatus ::= localValue 47
203 informServiceCentre InformServiceCentre ::= localValue 63
204 alertServiceCentre AlertServiceCentre ::= localValue 64
205 readyForSM ReadyForSM ::= localValue 66
206
207 -- provide subscriber info operation codes
208
209 provideSubscriberInfo ProvideSubscriberInfo ::= localValue 70
210
211 -- any time interrogation operation codes
212
213 anyTimeInterrogation AnyTimeInterrogation ::= localValue 71
214
215 -- generic error codes
216
217 systemFailure SystemFailure ::= localValue 34
218 dataMissing DataMissing ::= localValue 35
219 unexpectedDataValue UnexpectedDataValue ::= localValue 36
220 facilityNotSupported FacilityNotSupported ::= localValue 21
221
222
223 -- identification and numbering error codes
224
225 unknownSubscriber UnknownSubscriber ::= localValue 1
226 numberChanged NumberChanged ::= localValue 44
227 unknownMSC UnknownMSC ::= localValue 3
228 unidentifiedSubscriber UnidentifiedSubscriber ::= localValue 5
229 unknownEquipment UnknownEquipment ::= localValue 7
230
231
232 -- subscription error codes
233

```

```

234 roamingNotAllowed RoamingNotAllowed ::= localValue 8
235 illegalSubscriber IllegalSubscriber ::= localValue 9
236 illegalEquipment IllegalEquipment ::= localValue 12
237 bearerServiceNotProvisioned BearerServiceNotProvisioned ::=
238     localValue 10
239 teleserviceNotProvisioned TeleserviceNotProvisioned ::=
240     localValue 11
241
242
243 -- handover error codes
244
245 noHandoverNumberAvailable NoHandoverNumberAvailable ::=
246     localValue 25
247 subsequentHandoverFailure SubsequentHandoverFailure ::=
248     localValue 26
249
250
251 -- operation and maintenance error codes
252
253 tracingBufferFull TracingBufferFull ::= localValue 40
254
255
256 -- call handling error codes
257
258 noRoamingNumberAvailable NoRoamingNumberAvailable ::= localValue 39
259 absentSubscriber AbsentSubscriber ::= localValue 27
260 busySubscriber BusySubscriber ::= localValue 45
261 noSubscriberReply NoSubscriberReply ::= localValue 46
262 callBarred CallBarred ::= localValue 13
263 forwardingFailed ForwardingFailed ::= localValue 47
264 or-NotAllowed OR-NotAllowed ::= localValue 48
265 forwardingViolation ForwardingViolation ::= localValue 14
266 cug-Reject CUG-Reject ::= localValue 15
267
268
269 -- any time interrogation error codes
270 ati-NotAllowed ATI-NotAllowed ::= localValue 49
271
272
273 -- supplementary service error codes
274
275 illegalSS-Operation IllegalSS-Operation ::= localValue 16
276 ss-ErrorStatus SS-ErrorStatus ::= localValue 17
277 ss-NotAvailable SS-NotAvailable ::= localValue 18
278 ss-SubscriptionViolation SS-SubscriptionViolation ::= localValue 19
279 ss-Incompatibility SS-Incompatibility ::= localValue 20
280 unknownAlphabet UnknownAlphabet ::= localValue 71
281 ussd-Busy USSD-Busy ::= localValue 72
282 pw-RegistrationFailure PW-RegistrationFailure ::= localValue 37
283 negativePW-Check NegativePW-Check ::= localValue 38
284 numberOfPW-AttemptsViolation NumberOfPW-AttemptsViolation ::=
285     localValue 43
286
287
288 -- short message service error codes
289
290 subscriberBusyForMT-SMS SubscriberBusyForMT-SMS ::= localValue 31
291 sm-DeliveryFailure SM-DeliveryFailure ::= localValue 32
292 messageWaitingListFull MessageWaitingListFull ::= localValue 33
293 absentsubscriberSM AbsentSubscriberSM ::= localValue 6
294
295
296 -- The following operation codes are reserved for operations
297 -- existing in previous versions of the protocol
298
299 -- Operation Name          AC used          Oper. Code
300 --
301 -- sendParameters           map-ac infoRetrieval (14) version1 (1)    localValue 9
302 -- processUnstructuredSS-Data map-ac networkFunctionalSs (18) version1 (1) localValue 19
303 -- performHandover          map-ac handoverControl (11) version1 (1)    localValue 28
304 -- performSubsequentHandover map-ac handoverControl (11) version1 (1)    localValue 30
305 -- noteInternalHandover     map-ac handoverControl (11) version1 (1)    localValue 35
306 -- noteSubscriberPresent     map-ac mwdMngt (24) version1 (1)            localValue 48
307 -- alertServiceCentreWithoutResult map-ac shortMsgAlert (23) version1 (1)    localValue 49
308 -- traceSubscriberActivity   map-ac handoverControl (11) version1 (1)    localValue 52
309 -- beginSubscriberActivity   map-ac networkFunctionalSs (18) version1 (1) localValue 54
310
311 -- The following error codes are reserved for errors

```

312 -- existing in previous versions of the protocol

313

314 -- **Error Name**

AC used

Error Code

315 --

316 -- unknownBaseStation

map-ac handoverControl (11) version1 (1)

localValue 2

317 -- invalidTargetBaseStation

map-ac handoverControl (11) version1 (1)

localValue 23

318 -- noRadioResourceAvailable

map-ac handoverControl (11) version1 (1)

localValue 24

319

320

321 END

14.6 MAP operation and error types

14.6.1 Mobile Service Operations

```

1  MAP-MobileServiceOperations {
2      ccitt identified-organization (4) etsi (0) mobileDomain (0)
3      gsm-Network (1) modules (3) map-MobileServiceOperations (5)
4      version3 (3)}
5
6  DEFINITIONS
7
8  ::=
9
10 BEGIN
11
12 EXPORTS
13
14     -- location registration operations
15     UpdateLocation,
16     CancelLocation,
17     PurgeMS,
18     SendIdentification,
19
20     -- subscriber information enquiry operations
21     ProvideSubscriberInfo,
22
23     -- any time information enquiry operations
24     AnyTimeInterrogation,
25
26     -- handover operations
27     PrepareHandover,
28     SendEndSignal,
29     ProcessAccessSignalling,
30     ForwardAccessSignalling,
31     PrepareSubsequentHandover,
32
33     -- authentication management operations
34     SendAuthenticationInfo,
35
36     -- IMEI management operations
37     CheckIMEI,
38
39     -- subscriber management operations
40     InsertSubscriberData,
41     DeleteSubscriberData,
42
43     -- fault recovery operations
44     Reset,
45     ForwardCheckSS-Indication,
46     RestoreData
47 ;
48
49 IMPORTS
50     OPERATION
51 FROM TCAPMessages {
52     ccitt recommendation q 773 modules (2) messages (1) version2 (2)}
53
54     SystemFailure,
55     DataMissing,
56     UnexpectedDataValue,
57     UnknownSubscriber,
58     UnknownMSC,
59     UnidentifiedSubscriber,
60     UnknownEquipment,
61     RoamingNotAllowed,
62     ATI-NotAllowed,
63     NoHandoverNumberAvailable,
64     SubsequentHandoverFailure
65 FROM MAP-Errors {
66     ccitt identified-organization (4) etsi (0) mobileDomain (0)
67     gsm-Network (1) modules (3) map-Errors (10) version3 (3)}
68
69     UpdateLocationArg,
70     UpdateLocationRes,
71     CancelLocationArg,
72     PurgeMS-Arg,
73     SendIdentificationRes,
74     PrepareHO-Arg,
75     PrepareHO-Res,

```

```

76     PrepareSubsequentHO-Arg,
77     SendAuthenticationInfoArg,
78     SendAuthenticationInfoRes,
79     EquipmentStatus,
80     InsertSubscriberDataArg,
81     InsertSubscriberDataRes,
82     DeleteSubscriberDataArg,
83     DeleteSubscriberDataRes,
84     ResetArg,
85     RestoreDataArg,
86     RestoreDataRes,
87     ProvideSubscriberInfoArg,
88     ProvideSubscriberInfoRes,
89     AnyTimeInterrogationArg,
90     AnyTimeInterrogationRes
91
92 FROM MAP-MS-DataTypes {
93     ccitt identified-organization (4) etsi (0) mobileDomain (0)
94     gsm-Network (1) modules (3) map-MS-DataTypes (11) version3 (3)}
95
96     ExternalSignalInfo,
97     TMSI,
98     IMEI
99 FROM MAP-CommonDataTypes {
100     ccitt identified-organization (4) etsi (0) mobileDomain (0)
101     gsm-Network (1) modules (3) map-CommonDataTypes (18) version3 (3)}
102 ;
103
104

```

-- 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	
ERRORS {	
DataMissing,	
UnexpectedDataValue}	

PurgeMS ::= OPERATION	--Timer m
ARGUMENT	
purgeMS-Arg	PurgeMS-Arg
RESULT	

SendIdentification ::= OPERATION	--Timer s
ARGUMENT	
tmsi	TMSI
RESULT	
sendIdentificationRes	SendIdentificationRes
ERRORS {	
DataMissing,	
UnidentifiedSubscriber}	

-- subscriber information enquiry operations

ProvideSubscriberInfo ::= OPERATION	--Timer m
ARGUMENT	
provideSubscriberInfoArg	ProvideSubscriberInfoArg
RESULT	
provideSubscriberInfoRes	ProvideSubscriberInfoRes
ERRORS {	
DataMissing,	
UnexpectedDataValue}	

-- any time information enquiry operations


```

154 AnyTimeInterrogation ::= OPERATION                                --Timer m
155     ARGUMENT
156         anyTimeInterrogationArg    AnyTimeInterrogationArg
157     RESULT
158         anyTimeInterrogationRes    AnyTimeInterrogationRes
159     ERRORS {
160         SystemFailure,
161         ATI-NotAllowed,
162         DataMissing,
163         UnexpectedDataValue,
164         UnknownSubscriber}
165 -- handover operations
166
167 PrepareHandover ::= OPERATION                                --Timer m
168     ARGUMENT
169         prepareHO-Arg              PrepareHO-Arg
170     RESULT
171         prepareHO-Res              PrepareHO-Res
172     ERRORS {
173         SystemFailure,
174         DataMissing,
175         UnexpectedDataValue,
176         NoHandoverNumberAvailable}
177
178 SendEndSignal ::= OPERATION                                --Timer l
179     ARGUMENT
180         bss-APDU                    ExternalSignalInfo
181     RESULT
182
183 ProcessAccessSignalling ::= OPERATION                    --Timer s
184     ARGUMENT
185         bss-APDU                    ExternalSignalInfo
186
187 ForwardAccessSignalling ::= OPERATION                    --Timer s
188     ARGUMENT
189         bss-APDU                    ExternalSignalInfo
190
191 PrepareSubsequentHandover ::= OPERATION                    --Timer m
192     ARGUMENT
193         prepareSubsequentHO-Arg      PrepareSubsequentHO-Arg
194     RESULT
195         bss-APDU                    ExternalSignalInfo
196     ERRORS {
197         UnexpectedDataValue,
198         DataMissing,
199         UnknownMSC,
200         SubsequentHandoverFailure}
201
202 -- authentication management operations
203
204 SendAuthenticationInfo ::= OPERATION                    --Timer m
205     ARGUMENT
206         sendAuthenticationInfoArg      SendAuthenticationInfoArg
207     RESULT
208         sendAuthenticationInfoRes      SendAuthenticationInfoRes
209         -- optional
210     ERRORS {
211         SystemFailure,
212         DataMissing,
213         UnexpectedDataValue,
214         UnknownSubscriber}
215
216 -- IMEI management operations
217
218 CheckIMEI ::= OPERATION                                --Timer m
219     ARGUMENT
220         imei                          IMEI
221     RESULT
222         equipmentStatus                EquipmentStatus
223     ERRORS {
224         SystemFailure,
225         DataMissing,
226         UnknownEquipment}
227
228 -- subscriber management operations
229

```

```

230 InsertSubscriberData ::= OPERATION                                --Timer m
231     ARGUMENT
232         insertSubscriberDataArg      InsertSubscriberDataArg
233     RESULT
234         insertSubscriberDataRes      InsertSubscriberDataRes
235         -- optional
236     ERRORS {
237         DataMissing,
238         UnexpectedDataValue,
239         UnidentifiedSubscriber}
240
241 DeleteSubscriberData ::= OPERATION                                --Timer m
242     ARGUMENT
243         deleteSubscriberDataArg      DeleteSubscriberDataArg
244     RESULT
245         deleteSubscriberDataRes      DeleteSubscriberDataRes
246         -- optional
247     ERRORS {
248         DataMissing,
249         UnexpectedDataValue,
250         UnidentifiedSubscriber}
251
252 -- fault recovery operations
253
254 Reset ::= OPERATION                                              --Timer m
255     ARGUMENT
256         resetArg                      ResetArg
257
258 ForwardCheckSS-Indication ::= OPERATION                          --Timer s
259
260 RestoreData ::= OPERATION                                        --Timer m
261     ARGUMENT
262         restoreDataArg                RestoreDataArg
263     RESULT
264         restoreDataRes                RestoreDataRes
265     ERRORS {
266         SystemFailure,
267         DataMissing,
268         UnexpectedDataValue,
269         UnknownSubscriber}
270
271 END

```

14.6.2 Operation and Maintenance Operations

```

1 MAP-OperationAndMaintenanceOperations {
2   ccitt identified-organization (4) etsi (0) mobileDomain (0)
3   gsm-Network (1) modules (3) map-OperationAndMaintenanceOperations (6)
4   version3 (3)}
5
6 DEFINITIONS
7
8 ::=
9
10 BEGIN
11
12 EXPORTS
13   ActivateTraceMode,
14   DeactivateTraceMode,
15   SendIMSI
16 ;
17
18 IMPORTS
19   OPERATION
20 FROM TCAPMessages {
21   ccitt recommendation q 773 modules (2) messages (1) version2 (2)}
22
23   SystemFailure,
24   DataMissing,
25   UnexpectedDataValue,
26   FacilityNotSupported,
27   UnknownSubscriber,
28   UnidentifiedSubscriber,
29   TracingBufferFull
30 FROM MAP-Errors {
31   ccitt identified-organization (4) etsi (0) mobileDomain (0)
32   gsm-Network (1) modules (3) map-Errors (10) version3 (3)}
33
34   ActivateTraceModeArg,
35   ActivateTraceModeRes,
36   DeactivateTraceModeArg,
37   DeactivateTraceModeRes
38 FROM MAP-OM-DataTypes {
39   ccitt identified-organization (4) etsi (0) mobileDomain (0)
40   gsm-Network (1) modules (3) map-OM-DataTypes (12) version3 (3)}
41
42   ISDN-AddressString,
43   IMSI
44 FROM MAP-CommonDataTypes {
45   ccitt identified-organization (4) etsi (0) mobileDomain (0)
46   gsm-Network (1) modules (3) map-CommonDataTypes (18) version3 (3)}
47 ;
48
49
50 ActivateTraceMode ::= OPERATION --Timer m
51   ARGUMENT
52     activateTraceModeArg      ActivateTraceModeArg
53   RESULT
54     activateTraceModeRes      ActivateTraceModeRes
55     -- optional
56   ERRORS {
57     SystemFailure,
58     DataMissing,
59     UnexpectedDataValue,
60     FacilityNotSupported,
61     UnidentifiedSubscriber,
62     TracingBufferFull}
63
64 DeactivateTraceMode ::= OPERATION --Timer m
65   ARGUMENT
66     deactivateTraceModeArg    DeactivateTraceModeArg
67   RESULT
68     deactivateTraceModeRes    DeactivateTraceModeRes
69     -- optional
70   ERRORS {
71     SystemFailure,
72     DataMissing,
73     UnexpectedDataValue,
74     FacilityNotSupported,
75     UnidentifiedSubscriber}
76

```

```
77 SendIMSI ::= OPERATION --Timer m
78     ARGUMENT
79         msisdn                ISDN-AddressString
80     RESULT
81         imsi                IMSI
82     ERRORS {
83         DataMissing,
84         UnexpectedDataValue,
85         UnknownSubscriber}
86
87 END
```

14.6.3 Call Handling Operations

```

1 MAP-CallHandlingOperations {
2   ccitt identified-organization (4) etsi (0) mobileDomain (0)
3   gsm-Network (1) modules (3) map-CallHandlingOperations (7)
4   version3 (3)}
5
6 DEFINITIONS
7
8 ::=
9
10 BEGIN
11
12 EXPORTS
13   SendRoutingInfo,
14   ProvideRoamingNumber,
15   ResumeCallHandling
16 ;
17
18 IMPORTS
19   OPERATION
20 FROM TCAPMessages {
21   ccitt recommendation q 773 modules (2) messages (1) version2 (2)}
22
23   SystemFailure,
24   DataMissing,
25   UnexpectedDataValue,
26   FacilityNotSupported,
27   OR-NotAllowed,
28   UnknownSubscriber,
29   NumberChanged,
30   BearerServiceNotProvisioned,
31   TeleserviceNotProvisioned,
32   NoRoamingNumberAvailable,
33   AbsentSubscriber,
34   BusySubscriber,
35   NoSubscriberReply,
36   CallBarred,
37   ForwardingViolation,
38   ForwardingFailed,
39   CUG-Reject
40 FROM MAP-Errors {
41   ccitt identified-organization (4) etsi (0) mobileDomain (0)
42   gsm-Network (1) modules (3) map-Errors (10) version3 (3)}
43   SendRoutingInfoArg,
44   SendRoutingInfoRes,
45   ProvideRoamingNumberArg,
46   ProvideRoamingNumberRes,
47   ResumeCallHandlingArg,
48   ResumeCallHandlingRes
49 FROM MAP-CH-DataTypes {
50   ccitt identified-organization (4) etsi (0) mobileDomain (0)
51   gsm-Network (1) modules (3) map-CH-DataTypes (13) version3 (3)}
52
53 ;
54
55 SendRoutingInfo ::= OPERATION --Timer m
56   ARGUMENT
57     sendRoutingInfoArg          SendRoutingInfoArg
58   RESULT
59     sendRoutingInfoRes          SendRoutingInfoRes
60   ERRORS {
61     SystemFailure,
62     DataMissing,
63     UnexpectedDataValue,
64     FacilityNotSupported,
65     OR-NotAllowed,
66     UnknownSubscriber,
67     NumberChanged,
68     BearerServiceNotProvisioned,
69     TeleserviceNotProvisioned,
70     AbsentSubscriber,
71     BusySubscriber,
72     NoSubscriberReply,
73     CallBarred,
74     CUG-Reject,
75     ForwardingViolation}
76

```

```
77 ProvideRoamingNumber ::= OPERATION                                --Timer m
78     ARGUMENT
79         provideRoamingNumberArg      ProvideRoamingNumberArg
80     RESULT
81         provideRoamingNumberRes      ProvideRoamingNumberRes
82     ERRORS {
83         SystemFailure,
84         DataMissing,
85         UnexpectedDataValue,
86         FacilityNotSupported,
87         OR-NotAllowed,
88         AbsentSubscriber,
89         NoRoamingNumberAvailable}
90
91 ResumeCallHandling ::= OPERATION                                --Timer m
92     ARGUMENT
93         resumeCallHandlingArg        ResumeCallHandlingArg
94     RESULT
95         resumeCallHandlingRes        ResumeCallHandlingRes
96     ERRORS {
97         ForwardingFailed,
98         OR-NotAllowed,
99         UnexpectedDataValue}
100
101 END
```

14.6.4 Supplementary service operations

```

1  MAP-SupplementaryServiceOperations {
2      ccitt identified-organization (4) etsi (0) mobileDomain (0)
3      gsm-Network (1) modules (3) map-SupplementaryServiceOperations (8)
4      version3 (3)}
5
6  DEFINITIONS
7
8  ::=
9
10 BEGIN
11
12 EXPORTS
13     RegisterSS,
14     EraseSS,
15     ActivateSS,
16     DeactivateSS,
17     InterrogateSS,
18     ProcessUnstructuredSS-Request,
19     UnstructuredSS-Request,
20     UnstructuredSS-Notify,
21     RegisterPassword,
22     GetPassword
23 ;
24
25 IMPORTS
26     OPERATION
27 FROM TCAPMessages {
28     ccitt recommendation q 773 modules (2) messages (1) version2 (2)}
29
30     SystemFailure,
31     DataMissing,
32     UnexpectedDataValue,
33     BearerServiceNotProvisioned,
34     TeleserviceNotProvisioned,
35     CallBarred,
36     IllegalSS-Operation,
37     SS-ErrorStatus,
38     SS-NotAvailable,
39     SS-SubscriptionViolation,
40     SS-Incompatibility,
41     PW-RegistrationFailure,
42     NegativePW-Check,
43     NumberOfPW-AttemptsViolation,
44     UnknownAlphabet,
45     USSD-Busy,
46     AbsentSubscriber,
47     IllegalSubscriber,
48     IllegalEquipment
49 FROM MAP-Errors {
50     ccitt identified-organization (4) etsi (0) mobileDomain (0)
51     gsm-Network (1) modules (3) map-Errors (10) version3 (3)}
52
53     RegisterSS-Arg,
54     SS-Info,
55     SS-ForBS-Code,
56     InterrogateSS-Res,
57     USSD-Arg,
58     USSD-Res,
59     Password,
60     GuidanceInfo
61 FROM MAP-SS-DataTypes {
62     ccitt identified-organization (4) etsi (0) mobileDomain (0)
63     gsm-Network (1) modules (3) map-SS-DataTypes (14) version3 (3)}
64
65     SS-Code
66 FROM MAP-SS-Code {
67     ccitt identified-organization (4) etsi (0) mobileDomain (0)
68     gsm-Network (1) modules (3) map-SS-Code (15) version3 (3)}
69 ;
70
71
72 -- supplementary service handling operations
73

```

```

74 RegisterSS ::= OPERATION                                --Timer m
75     ARGUMENT
76         registerSS-Arg                                RegisterSS-Arg
77     RESULT
78         ss-Info                                        SS-Info
79         -- optional
80     ERRORS {
81         SystemFailure,
82         DataMissing,
83         UnexpectedDataValue,
84         BearerServiceNotProvisioned,
85         TeleserviceNotProvisioned,
86         CallBarred,
87         IllegalSS-Operation,
88         SS-ErrorStatus,
89         SS-Incompatibility}
90
91 EraseSS ::= OPERATION                                --Timer m
92     ARGUMENT
93         ss-ForBS                                        SS-ForBS-Code
94     RESULT
95         ss-Info                                        SS-Info
96         -- optional
97     ERRORS {
98         SystemFailure,
99         DataMissing,
100        UnexpectedDataValue,
101        BearerServiceNotProvisioned,
102        TeleserviceNotProvisioned,
103        CallBarred,
104        IllegalSS-Operation,
105        SS-ErrorStatus
106    }
107
108 ActivateSS ::= OPERATION                                --Timer m
109     ARGUMENT
110         ss-ForBS                                        SS-ForBS-Code
111     RESULT
112         ss-Info                                        SS-Info
113         -- optional
114     ERRORS {
115         SystemFailure,
116         DataMissing,
117         UnexpectedDataValue,
118         BearerServiceNotProvisioned,
119         TeleserviceNotProvisioned,
120         CallBarred,
121         IllegalSS-Operation,
122         SS-ErrorStatus,
123         SS-SubscriptionViolation,
124         SS-Incompatibility,
125         NegativePW-Check,
126         NumberOfPW-AttemptsViolation}
127
128 DeactivateSS ::= OPERATION                                --Timer m
129     ARGUMENT
130         ss-ForBS                                        SS-ForBS-Code
131     RESULT
132         ss-Info                                        SS-Info
133         -- optional
134     ERRORS {
135         SystemFailure,
136         DataMissing,
137         UnexpectedDataValue,
138         BearerServiceNotProvisioned,
139         TeleserviceNotProvisioned,
140         CallBarred,
141         IllegalSS-Operation,
142         SS-ErrorStatus,
143         SS-SubscriptionViolation,
144         NegativePW-Check,
145         NumberOfPW-AttemptsViolation}
146

```



```

147 InterrogateSS ::= OPERATION                                --Timer m
148     ARGUMENT
149         ss-ForBS                                SS-ForBS-Code
150     RESULT
151         interrogateSS-Res                        InterrogateSS-Res
152     ERRORS {
153         SystemFailure,
154         DataMissing,
155         UnexpectedDataValue,
156         BearerServiceNotProvisioned,
157         TeleserviceNotProvisioned,
158         CallBarred,
159         IllegalSS-Operation,
160         SS-NotAvailable}
161
162 ProcessUnstructuredSS-Request ::= OPERATION                --Timer 10 minutes
163     ARGUMENT
164         ussd-Arg                                USSD-Arg
165     RESULT
166         ussd-Res                                USSD-Res
167     ERRORS {
168         SystemFailure,
169         DataMissing,
170         UnexpectedDataValue,
171         UnknownAlphabet,
172         CallBarred}
173
174 UnstructuredSS-Request ::= OPERATION                        --Timer m1
175     ARGUMENT
176         ussd-Arg                                USSD-Arg
177     RESULT
178         ussd-Res                                USSD-Res
179         -- optional
180     ERRORS {
181         SystemFailure,
182         DataMissing,
183         UnexpectedDataValue,
184         AbsentSubscriber,
185         IllegalSubscriber,
186         IllegalEquipment,
187         UnknownAlphabet,
188         USSD-Busy}
189
190 UnstructuredSS-Notify ::= OPERATION                        --Timer m1
191     ARGUMENT
192         ussd-Arg                                USSD-Arg
193     RESULT
194     ERRORS {
195         SystemFailure,
196         DataMissing,
197         UnexpectedDataValue,
198         AbsentSubscriber,
199         IllegalSubscriber,
200         IllegalEquipment,
201         UnknownAlphabet,
202         USSD-Busy}
203
204 RegisterPassword ::= OPERATION                            --Timer m1
205     ARGUMENT
206         ss-Code                                SS-Code
207     RESULT
208         newPassword                            Password
209     ERRORS {
210         SystemFailure,
211         DataMissing,
212         UnexpectedDataValue,
213         CallBarred,
214         SS-SubscriptionViolation,
215         PW-RegistrationFailure,
216         NegativePW-Check,
217         NumberOfPW-AttemptsViolation}
218     LINKED {
219         GetPassword}
220

```

221	GetPassword ::= OPERATION	--Timer m
222	ARGUMENT	
223	guidanceInfo	GuidanceInfo
224	RESULT	
225	currentPassword	Password
226		
227	END	

14.6.5 Short message service operations

```

1  MAP-ShortMessageServiceOperations {
2      ccitt identified-organization (4) etsi (0) mobileDomain (0)
3      gsm-Network (1) modules (3) map-ShortMessageServiceOperations (9)
4      version3 (3)}
5
6  DEFINITIONS
7
8  ::=
9
10 BEGIN
11
12 EXPORTS
13     SendRoutingInfoForSM,
14     MO-ForwardSM,
15     MT-ForwardSM,
16     ReportSM-DeliveryStatus,
17     AlertServiceCentre,
18     InformServiceCentre,
19     ReadyForSM
20 ;
21
22 IMPORTS
23     OPERATION
24 FROM TCAPMessages {
25     ccitt recommendation q 773 modules (2) messages (1) version2 (2)}
26
27     SystemFailure,
28     DataMissing,
29     UnexpectedDataValue,
30     FacilityNotSupported,
31     UnknownSubscriber,
32     UnidentifiedSubscriber,
33     IllegalSubscriber,
34     IllegalEquipment,
35     TeleserviceNotProvisioned,
36     AbsentSubscriber,
37     CallBarred,
38     SubscriberBusyForMT-SMS,
39     SM-DeliveryFailure,
40     MessageWaitingListFull,
41     AbsentSubscriberSM
42 FROM MAP-Errors {
43     ccitt identified-organization (4) etsi (0) mobileDomain (0)
44     gsm-Network (1) modules (3) map-Errors (10) version3 (3)}
45
46     RoutingInfoForSM-Arg,
47     RoutingInfoForSM-Res,
48     MO-ForwardSM-Arg,
49     MO-ForwardSM-Res,
50     MT-ForwardSM-Arg,
51     MT-ForwardSM-Res,
52     ReportSM-DeliveryStatusArg,
53     ReportSM-DeliveryStatusRes,
54     AlertServiceCentreArg,
55     InformServiceCentreArg,
56     ReadyForSM-Arg
57 FROM MAP-SM-DataTypes {
58     ccitt identified-organization (4) etsi (0) mobileDomain (0)
59     gsm-Network (1) modules (3) map-SM-DataTypes (16) version3 (3)}
60
61
62
63 ;
64
65
    
```

```

66 SendRoutingInfoForSM ::= OPERATION                                --Timer m
67     ARGUMENT
68         routingInfoForSM-Arg          RoutingInfoForSM-Arg
69     RESULT
70         routingInfoForSM-Res          RoutingInfoForSM-Res
71     ERRORS {
72         SystemFailure,
73         DataMissing,
74         UnexpectedDataValue,
75         FacilityNotSupported,
76         UnknownSubscriber,
77         TeleserviceNotProvisioned,
78         CallBarred,
79         AbsentSubscriberSM}
80
81 MO-ForwardSM ::= OPERATION                                --Timer m1
82     ARGUMENT
83         mo-forwardSM-Arg              MO-ForwardSM-Arg
84     RESULT
85         mo-forwardSM-Res              MO-ForwardSM-Res
86         -- optional
87     ERRORS {
88         SystemFailure,
89         UnexpectedDataValue,
90         FacilityNotSupported,
91         SM-DeliveryFailure}
92
93 MT-ForwardSM ::= OPERATION                                --Timer m1
94     ARGUMENT
95         mt-forwardSM-Arg              MT-ForwardSM-Arg
96     RESULT
97         mt-forwardSM-Res              MT-ForwardSM-Res
98         -- optional
99     ERRORS {
100         SystemFailure,
101         DataMissing,
102         UnexpectedDataValue,
103         FacilityNotSupported,
104         UnidentifiedSubscriber,
105         IllegalSubscriber,
106         IllegalEquipment,
107         SubscriberBusyForMT-SMS,
108         SM-DeliveryFailure,
109         AbsentSubscriberSM}
110
111 ReportSM-DeliveryStatus ::= OPERATION                    --Timer s
112     ARGUMENT
113         reportSM-DeliveryStatusArg    ReportSM-DeliveryStatusArg
114     RESULT
115         reportSM-DeliveryStatusRes    ReportSM-DeliveryStatusRes
116         -- optional
117     ERRORS {
118         DataMissing,
119         UnexpectedDataValue,
120         UnknownSubscriber,
121         MessageWaitingListFull}
122
123 AlertServiceCentre ::= OPERATION                        --Timer s
124     ARGUMENT
125         alertServiceCentreArg          AlertServiceCentreArg
126     RESULT
127     ERRORS {
128         SystemFailure,
129         DataMissing,
130         UnexpectedDataValue}
131
132 InformServiceCentre ::= OPERATION                       --Timer s
133     ARGUMENT
134         informServiceCentreArg          InformServiceCentreArg
135

```

```
136 ReadyForSM ::= OPERATION --Timer m
137     ARGUMENT
138         readyForSM-Arg          ReadyForSM-Arg
139     RESULT
140     ERRORS {
141         DataMissing,
142         UnexpectedDataValue,
143         FacilityNotSupported,
144         UnknownSubscriber}
145
146 END
```

14.6.6 Errors

```
1 MAP-Errors {
2   ccitt identified-organization (4) etsi (0) mobileDomain (0)
3   gsm-Network (1) modules (3) map-Errors (10) version3 (3)}
4
5 DEFINITIONS
6
7 ::=
8
9 BEGIN
10
11 EXPORTS
12
13   -- generic errors
14   SystemFailure,
15   DataMissing,
16   UnexpectedDataValue,
17   FacilityNotSupported,
18
19   -- identification and numbering errors
20   UnknownSubscriber,
21   NumberChanged,
22   UnknownMSC,
23   UnidentifiedSubscriber,
24   UnknownEquipment,
25
26   -- subscription errors
27   RoamingNotAllowed,
28   IllegalSubscriber,
29   IllegalEquipment,
30   BearerServiceNotProvisioned,
31   TeleserviceNotProvisioned,
32
33   -- handover errors
34   NoHandoverNumberAvailable,
35   SubsequentHandoverFailure,
36
37   -- operation and maintenance errors
38   TracingBufferFull,
39
40   -- call handling errors
41   OR-NotAllowed,
42   NoRoamingNumberAvailable,
43   BusySubscriber,
44   NoSubscriberReply,
45   AbsentSubscriber,
46   CallBarred,
47   ForwardingViolation,
48   ForwardingFailed,
49   CUG-Reject,
50
51   -- any time interrogation errors
52   ATI-NotAllowed,
53
54   -- supplementary service errors
55   IllegalSS-Operation,
56   SS-ErrorStatus,
57   SS-NotAvailable,
58   SS-SubscriptionViolation,
59   SS-Incompatibility,
60   UnknownAlphabet,
61   USSD-Busy,
62   PW-RegistrationFailure,
63   NegativePW-Check,
64   NumberOfPW-AttemptsViolation,
65
66   -- short message service errors
67   SubscriberBusyForMT-SMS,
68   SM-DeliveryFailure,
69   MessageWaitingListFull,
70   AbsentSubscriberSM
71 ;
72
73 IMPORTS
74   ERROR
75 FROM TCAPMessages {
76   ccitt recommendation q 773 modules (2) messages (1) version2 (2)}
77
78   SS-Status
```

```

79 FROM MAP-SS-DataTypes {
80   ccitt identified-organization (4) etsi (0) mobileDomain (0)
81   gsm-Network (1) modules (3) map-SS-DataTypes (14) version3 (3)}
82
83   SS-IncompatibilityCause,
84   PW-RegistrationFailureCause,
85   SM-DeliveryFailureCause,
86   SystemFailureParam,
87   DataMissingParam,
88   UnexpectedDataParam,
89   FacilityNotSupParam,
90   UnknownSubscriberParam,
91   NumberChangedParam,
92   UnidentifiedSubParam,
93   RoamingNotAllowedParam,
94   IllegalSubscriberParam,
95   IllegalEquipmentParam,
96   BearerServNotProvParam,
97   TeleservNotProvParam,
98   TracingBufferFullParam,
99   NoRoamingNbParam,
100  OR-NotAllowedParam,
101  AbsentSubscriberParam,
102  BusySubscriberParam,
103  NoSubscriberReplyParam,
104  CallBarredParam,
105  ForwardingViolationParam,
106  ForwardingFailedParam,
107  CUG-RejectParam,
108  ATI-NotAllowedParam,
109  SubBusyForMT-SMS-Param,
110  MessageWaitListFullParam,
111  AbsentSubscriberSM-Param
112
113 FROM MAP-ER-DataTypes {
114   ccitt identified-organization (4) etsi (0) mobileDomain (0)
115   gsm-Network (1) modules (3) map-ER-DataTypes (17) version3 (3)}
116 ;
117
118 -- generic errors
119
120
121 SystemFailure ::= ERROR
122   PARAMETER
123     systemFailureParam          SystemFailureParam
124     -- optional
125
126 DataMissing ::= ERROR
127   PARAMETER
128     dataMissingParam            DataMissingParam
129     -- optional
130     -- dataMissingParam must not be used in version <3
131
132 UnexpectedDataValue ::= ERROR
133   PARAMETER
134     unexpectedDataParam          UnexpectedDataParam
135     -- optional
136     -- unexpectedDataParam must not be used in version <3
137
138 FacilityNotSupported ::= ERROR
139   PARAMETER
140     facilityNotSupParam          FacilityNotSupParam
141     -- optional
142     -- facilityNotSupParam must not be used in version <3
143
144 -- identification and numbering errors
145
146
147 UnknownSubscriber ::= ERROR
148   PARAMETER
149     unknownSubscriberParam       UnknownSubscriberParam
150     -- optional
151     -- unknownSubscriberParam must not be used in version <3
152
153 NumberChanged ::= ERROR
154   PARAMETER
155     numberChangedParam           NumberChangedParam
156     -- optional
157

```

158	UnknownMSC ::= ERROR
159	
160	UnidentifiedSubscriber ::= ERROR
161	PARAMETER
162	unidentifiedSubParam UnidentifiedSubParam
163	-- optional
164	-- unidentifiedSubParam must not be used in version <3
165	
166	UnknownEquipment ::= ERROR
167	
168	
169	-- subscription errors
170	
171	RoamingNotAllowed ::= ERROR
172	PARAMETER
173	roamingNotAllowedParam RoamingNotAllowedParam
174	
175	IllegalSubscriber ::= ERROR
176	PARAMETER
177	illegalSubscriberParam IllegalSubscriberParam
178	-- optional
179	-- illegalSubscriberParam must not be used in version <3
180	
181	IllegalEquipment ::= ERROR
182	PARAMETER
183	illegalEquipmentParam IllegalEquipmentParam
184	-- optional
185	-- illegalEquipmentParam must not be used in version <3
186	
187	BearerServiceNotProvisioned ::= ERROR
188	PARAMETER
189	bearerServNotProvParam BearerServNotProvParam
190	-- optional
191	-- bearerServNotProvParam must not be used in version <3
192	
193	TeleserviceNotProvisioned ::= ERROR
194	PARAMETER
195	teleservNotProvParam TeleservNotProvParam
196	-- optional
197	-- teleservNotProvParam must not be used in version <3
198	
199	
200	-- handover errors
201	
202	NoHandoverNumberAvailable ::= ERROR
203	
204	SubsequentHandoverFailure ::= ERROR
205	
206	
207	-- operation and maintenance errors
208	
209	TracingBufferFull ::= ERROR
210	PARAMETER
211	tracingBufferFullParam TracingBufferFullParam
212	-- optional
213	
214	
215	-- call handling errors
216	
217	NoRoamingNumberAvailable ::= ERROR
218	PARAMETER
219	noRoamingNbParam NoRoamingNbParam
220	-- optional
221	
222	AbsentSubscriber ::= ERROR
223	PARAMETER
224	absentSubscriberParam AbsentSubscriberParam
225	-- optional
226	
227	-- absentSubscriberParam must not be used in version <3
228	BusySubscriber ::= ERROR
229	PARAMETER
230	busySubscriberParam BusySubscriberParam
231	-- optional
232	
233	NoSubscriberReply ::= ERROR
234	PARAMETER
235	noSubscriberReplyParam NoSubscriberReplyParam

236	-- optional	
237		
238	CallBarred ::= ERROR	
239	PARAMETER	
240	callBarredParam	CallBarredParam
241	-- optional	
242		
243	ForwardingViolation ::= ERROR	
244	PARAMETER	
245	forwardingViolationParam	ForwardingViolationParam
246	-- optional	
247		
248	ForwardingFailed ::= ERROR	
249	PARAMETER	
250	forwardingFailedParam	ForwardingFailedParam
251	-- optional	
252		
253	CUG-Reject ::= ERROR	
254	PARAMETER	
255	cug-RejectParam	CUG-RejectParam
256	-- optional	
257		
258	OR-NotAllowed ::= ERROR	
259	PARAMETER	
260	or-NotAllowedParam	OR-NotAllowedParam
261	-- optional	
262		
263		
264	-- any time interrogation errors	
265	ATI-NotAllowed ::= ERROR	
266	PARAMETER	
267	ati-NotAllowedParam	ATI-NotAllowedParam
268	-- optional	
269		
270		
271	-- supplementary service errors	
272		
273	IllegalSS-Operation ::= ERROR	
274		
275	SS-ErrorStatus ::= ERROR	
276	PARAMETER	
277	ss-Status	SS-Status
278	-- optional	
279		
280	SS-NotAvailable ::= ERROR	
281		
282	SS-SubscriptionViolation ::= ERROR	
283		
284	SS-Incompatibility ::= ERROR	
285	PARAMETER	
286	ss-IncompatibilityCause	SS-IncompatibilityCause
287	-- optional	
288		
289	UnknownAlphabet ::= ERROR	
290		
291	USSD-Busy ::= ERROR	
292		
293	PW-RegistrationFailure ::= ERROR	
294	PARAMETER	
295	pw-RegistrationFailureCause	PW-RegistrationFailureCause
296		
297	NegativePW-Check ::= ERROR	
298		
299	NumberOfPW-AttemptsViolation ::= ERROR	
300		
301		
302	-- short message service errors	
303		
304	SubscriberBusyForMT-SMS ::= ERROR	
305	PARAMETER	
306	subBusyForMT-SMS-Param	SubBusyForMT-SMS-Param
307	-- optional	
308		
309	SM-DeliveryFailure ::= ERROR	
310	PARAMETER	
311	sm-DeliveryFailureCause	SM-DeliveryFailureCause
312		

```
313 MessageWaitingListFull ::= ERROR
314     PARAMETER
315         messageWaitListFullParam      MessageWaitListFullParam
316         -- optional
317
318
319 AbsentSubscribersSM ::= ERROR
320     PARAMETER
321         absentSubscriberSM-Param      AbsentSubscriberSM-Param
322         -- optional
323
324 END
```

14.7 MAP constants and data types

14.7.1 Mobile Service data types

```

1 MAP-MS-DataTypes {
2   ccitt identified-organization (4) etsi (0) mobileDomain (0)
3   gsm-Network (1) modules (3) map-MS-DataTypes (11) version3 (3)}
4
5 DEFINITIONS
6
7 IMPLICIT TAGS
8
9 ::=
10
11 BEGIN
12
13 EXPORTS
14
15   -- location registration types
16   UpdateLocationArg,
17   UpdateLocationRes,
18   CancelLocationArg,
19   PurgeMS-Arg,
20   SendIdentificationRes,
21
22   -- handover types
23   PrepareHO-Arg,
24   PrepareHO-Res,
25   PrepareSubsequentHO-Arg,
26
27   -- authentication management types
28   SendAuthenticationInfoArg,
29   SendAuthenticationInfoRes,
30
31   -- security management types
32   EquipmentStatus,
33
34   -- subscriber management types
35   InsertSubscriberDataArg,
36   InsertSubscriberDataRes,
37   DeleteSubscriberDataArg,
38   DeleteSubscriberDataRes,
39   SubscriberData,
40   ODB-Data,
41   SubscriberStatus,
42   ZoneCodeList,
43   maxNumOfZoneCodes,
44   O-CSI,
45   ServiceKey,
46   DefaultCallHandling,
47   SupportedCamelPhases,
48   maxNumOfCamelTDPData,
49   CUG-Index,
50   CUG-Interlock,
51   InterCUG-Restrictions,
52   IntraCUG-Options,
53
54   -- fault recovery types
55   ResetArg,
56   RestoreDataArg,
57   RestoreDataRes,
58
59   -- subscriber information enquiry types
60   ProvideSubscriberInfoArg,
61   ProvideSubscriberInfoRes,
62   SubscriberInfo,
63   LocationInformation,
64   SubscriberState,
65
66   -- any time information enquiry types
67   AnyTimeInterrogationArg,
68   AnyTimeInterrogationRes
69
70 ;
71
72 IMPORTS
73   maxNumOfSS,
74   SS-SubscriptionOption,
75   SS-List

```

```

76 FROM MAP-SS-DataTypes {
77     ccitt identified-organization (4) etsi (0) mobileDomain (0)
78     gsm-Network (1) modules (3) map-SS-DataTypes (14) version3 (3)}
79
80     SS-Code
81 FROM MAP-SS-Code {
82     ccitt identified-organization (4) etsi (0) mobileDomain (0)
83     gsm-Network (1) modules (3) map-SS-Code (15) version3 (3)}
84
85     Ext-BearerServiceCode
86 FROM MAP-BS-Code {
87     ccitt identified-organization (4) etsi (0) mobileDomain (0)
88     gsm-Network (1) modules (3) map-BS-Code (20) version3 (3)}
89
90     Ext-TeleserviceCode
91 FROM MAP-TS-Code {
92     ccitt identified-organization (4) etsi (0) mobileDomain (0)
93     gsm-Network (1) modules (3) map-TS-Code (19) version3 (3)}
94
95
96     ISDN-AddressString,
97     ISDN-SubaddressString,
98     ExternalSignalInfo,
99     IMSI,
100    HLR-List,
101    LMSI,
102    GlobalCellId,
103    CellIdOrLAI,
104    Ext-BasicServiceCode
105
106 FROM MAP-CommonDataTypes {
107     ccitt identified-organization (4) etsi (0) mobileDomain (0)
108     gsm-Network (1) modules (3) map-CommonDataTypes (18) version3 (3)}
109
110     ExtensionContainer
111 FROM MAP-ExtensionDataTypes {
112     ccitt identified-organization (4) etsi (0) mobileDomain (0)
113     gsm-Network (1) modules (3) map-ExtensionDataTypes (21) version3 (3)}
114
115 ;
116
117 -- location registration types
118

```

UpdateLocationArg ::= SEQUENCE {			
imsi	IMSI,		
msc-Number	[1] ISDN-AddressString,	vlr-Number	
ISDN-AddressString,			
lmsi	[10] LMSI	OPTIONAL,	
extensionContainer	ExtensionContainer	OPTIONAL,	
...			
}			

UpdateLocationRes ::= SEQUENCE {			
hlr-Number	ISDN-AddressString,		
extensionContainer	ExtensionContainer	OPTIONAL,	
...			
}			

CancelLocationArg ::= CHOICE {	
imsi	IMSI,
imsi-WithLMSI	IMSI-WithLMSI}

PurgeMS-Arg ::= SEQUENCE {	
imsi	IMSI,
vlr-Number	ISDN-AddressString,
...	
}	

IMSI-WithLMSI ::= SEQUENCE {	
imsi	IMSI,
lmsi	LMSI,
-- a special value 00000000 indicates that the LMSI is not in use	
...	
}	

SendIdentificationRes ::= SEQUENCE {			
imsi	IMSI,		
authenticationSetList	AuthenticationSetList	OPTIONAL,	
...			
}			

```

154 AuthenticationSetList ::= SEQUENCE SIZE (1..5) OF
155     AuthenticationSet
156
157 AuthenticationSet ::= SEQUENCE {
158     rand                RAND,
159     sres                SRES,
160     kc                 Kc,
161     ...}
162
163 RAND ::= OCTET STRING (SIZE (16))
164
165 SRES ::= OCTET STRING (SIZE (4))
166
167 Kc ::= OCTET STRING (SIZE (8))
168
169 -- handover types
170
171 PrepareHO-Arg ::= SEQUENCE {
172     targetCellId        GlobalCellId                OPTIONAL,
173     ho-NumberNotRequired NULL                        OPTIONAL,
174     bss-APDU            ExternalSignalInfo           OPTIONAL,
175     ...}
176
177 PrepareHO-Res ::= SEQUENCE {
178     handoverNumber      ISDN-AddressString           OPTIONAL,
179     bss-APDU            ExternalSignalInfo           OPTIONAL,
180     ...}
181
182 PrepareSubsequentHO-Arg ::= SEQUENCE {
183     targetCellId        GlobalCellId,
184     targetMSC-Number    ISDN-AddressString,
185     bss-APDU            ExternalSignalInfo,
186     ...}
187
188 -- authentication management types
189
190 SendAuthenticationInfoArg ::= IMSI
191
192 SendAuthenticationInfoRes ::= AuthenticationSetList
193
194 -- security management types
195
196
197 EquipmentStatus ::= ENUMERATED {
198     whiteListed (0),
199     blackListed (1),
200     greyListed (2)}
201
202 -- subscriber management types
203
204
205 InsertSubscriberDataArg ::= SEQUENCE {
206     imsi                [0] IMSI                OPTIONAL,
207     COMPONENTS OF      SubscriberData,
208     extensionContainer  [14] ExtensionContainer  OPTIONAL,
209     ...}
210
211 SubscriberData ::= SEQUENCE {
212     msisdN              [1] ISDN-AddressString   OPTIONAL,
213     category            [2] Category            OPTIONAL,
214     subscriberStatus    [3] SubscriberStatus    OPTIONAL,
215     bearerServiceList   [4] BearerServiceList   OPTIONAL,
216     -- The exception handling for reception of unsupported / not allocated
217     -- bearerServiceCodes is defined in section 6.8.1
218     teleserviceList     [6] TeleserviceList      OPTIONAL,
219     -- The exception handling for reception of unsupported / not allocated
220     -- teleserviceCodes is defined in section 6.8.1
221     provisionedSS       [7] Ext-SS-InfoList      OPTIONAL,
222     odb-Data            [8] ODB-Data            OPTIONAL,
223     roamingRestrictionDueToUnsupportedFeature [9] NULL        OPTIONAL,
224     regionalSubscriptionData [10] ZoneCodeList   OPTIONAL,
225     vbsSubscriptionData [11] VBSDataList        OPTIONAL,
226     vgcsSubscriptionData [12] VGCSDataList      OPTIONAL,
227     vlrCamelSubscriptionInfo [13] VlrCamelSubscriptionInfo OPTIONAL
228 }
229
230 Category ::= OCTET STRING (SIZE (1))
231 -- The internal structure is defined in CCITT Rec Q.763.

```

232	
233	SubscriberStatus ::= ENUMERATED {
234	serviceGranted (0),
235	operatorDeterminedBarring (1)}
236	
237	BearerServiceList ::= SEQUENCE SIZE (1..maxNumOfBearerServices) OF
238	Ext-BearerServiceCode
239	
240	maxNumOfBearerServices INTEGER ::= 50
241	
242	TeleserviceList ::= SEQUENCE SIZE (1..maxNumOfTeleservices) OF
243	Ext-TeleserviceCode
244	
245	maxNumOfTeleservices INTEGER ::= 20
246	
247	ODB-Data ::= SEQUENCE {
248	odb-GeneralData ODB-GeneralData,
249	odb-HPLMN-Data ODB-HPLMN-Data OPTIONAL,
250	extensionContainer ExtensionContainer OPTIONAL,
251	...}
252	
253	ODB-GeneralData ::= BIT STRING {
254	allOG-CallsBarred (0),
255	internationalOGCallsBarred (1),
256	internationalOGCallsNotToHPLMN-CountryBarred (2),
257	interzonalOGCallsBarred (6),
258	interzonalOGCallsNotToHPLMN-CountryBarred (7),
259	interzonalOGCallsAndInternationalOGCallsNotToHPLMN-CountryBarred (8),
260	premiumRateInformationOGCallsBarred (3),
261	premiumRateEntertainmentOGCallsBarred (4),
262	ss-AccessBarred (5),
263	allECT-Barred (9),
264	chargeableECT-Barred (10),
265	internationalECT-Barred (11),
266	interzonalECT-Barred (12),
267	doublyChargeableECT-Barred (13),
268	multipleECT-Barred (14)} (SIZE (15..32))
269	-- exception handling: reception of unknown bit assignments in the
270	-- ODB-GeneralData type shall be treated like unsupported ODB-GeneralData
271	
272	ODB-HPLMN-Data ::= BIT STRING {
273	plmn-SpecificBarringType1 (0),
274	plmn-SpecificBarringType2 (1),
275	plmn-SpecificBarringType3 (2),
276	plmn-SpecificBarringType4 (3)} (SIZE (4..32))
277	-- exception handling: reception of unknown bit assignments in the
278	-- ODB-HPLMN-Data type shall be treated like unsupported ODB-HPLMN-Data
279	
280	Ext-SS-InfoList ::= SEQUENCE SIZE (1..maxNumOfSS) OF
281	Ext-SS-Info
282	
283	Ext-SS-Info ::= CHOICE {
284	forwardingInfo [0] Ext-ForwInfo,
285	callBarringInfo [1] Ext-CallBarInfo,
286	cug-Info [2] CUG-Info,
287	ss-Data [3] Ext-SS-Data,
288	emlpp-Info [4] EMLPP-Info}
289	
290	EMLPP-Info ::= SEQUENCE {
291	maximumentitledPriority EMLPP-Priority,
292	defaultPriority EMLPP-Priority,
293	extensionContainer ExtensionContainer OPTIONAL,
294	...}
295	
296	EMLPP-Priority ::= INTEGER (0..15)
297	-- The mapping from the values A,B,0,1,2,3,4 to the integer-value is
298	-- specified as follows where A is the highest and 4 is the lowest
299	-- priority level
300	-- the integer values 7-15 are spare and shall be mapped to value 4
301	
302	priorityLevelA EMLPP-Priority ::= 6
303	priorityLevelB EMLPP-Priority ::= 5
304	priorityLevel0 EMLPP-Priority ::= 0
305	priorityLevel1 EMLPP-Priority ::= 1
306	priorityLevel2 EMLPP-Priority ::= 2
307	priorityLevel3 EMLPP-Priority ::= 3
308	priorityLevel4 EMLPP-Priority ::= 4
309	

```

310 Ext-ForwInfo ::= SEQUENCE {
311     ss-Code                      SS-Code,
312     forwardingFeatureList        Ext-ForwFeatureList,
313     extensionContainer            [0] ExtensionContainer OPTIONAL,
314     ...}
315
316 Ext-ForwFeatureList ::=
317     SEQUENCE SIZE (1..maxNumOfExt-BasicServiceGroups) OF
318         Ext-ForwFeature
319
320 Ext-ForwFeature ::= SEQUENCE {
321     basicService                  Ext-BasicServiceCode OPTIONAL,
322     ss-Status                    [4] Ext-SS-Status,
323     forwardedToNumber            [5] ISDN-AddressString OPTIONAL,
324     forwardedToSubaddress        [8] ISDN-SubaddressString OPTIONAL,
325     forwardingOptions            [6] Ext-ForwOptions OPTIONAL,
326     noReplyConditionTime        [7] Ext-NoRepCondTime OPTIONAL,
327     extensionContainer            [9] ExtensionContainer OPTIONAL,
328     ...}
329
330 Ext-SS-Status ::= OCTET STRING (SIZE (1..5))
331
332     -- OCTET 1:
333     --
334     -- bits 8765: 0000 (unused)
335     -- bits 4321: Used to convey the "P bit","R bit","A bit" and "Q bit",
336     --             representing supplementary service state information
337     --             as defined in TS GSM 03.11
338
339     -- bit 4: "Q bit"
340
341     -- bit 3: "P bit"
342
343     -- bit 2: "R bit"
344
345     -- bit 1: "A bit"
346
347     -- OCTETS 2-5: reserved for future use. They shall be discarded if
348     -- received and not understood.
349
350
351 Ext-ForwOptions ::= OCTET STRING (SIZE (1..5))
352
353     -- OCTET 1:
354     --
355     -- bit 8: notification to forwarding party
356     -- 0 no notification
357     -- 1 notification
358
359     -- bit 7: 0 (unused)
360
361     -- bit 6: notification to calling party
362     -- 0 no notification
363     -- 1 notification
364
365     -- bit 5: 0 (unused)
366
367     -- bits 43: forwarding reason
368     -- 00 ms not reachable
369     -- 01 ms busy
370     -- 10 no reply
371     -- 11 unconditional
372
373     -- bits 21: 00 (unused)
374
375     -- OCTETS 2-5: reserved for future use. They shall be discarded if
376     -- received and not understood.
377
378 Ext-NoRepCondTime ::= INTEGER (1..100)
379     -- Only values 5-30 are used.
380     -- Values in the ranges 1-4 and 31-100 are reserved for future use
381     -- If received:
382     --     values 1-4 shall be mapped on to value 5
383     --     values 31-100 shall be mapped on to value 30
384

```

385	Ext-CallBarInfo ::= SEQUENCE {		
386	ss-Code	SS-Code,	
387	callBarringFeatureList	Ext-CallBarFeatureList,	
388	extensionContainer	ExtensionContainer	OPTIONAL,
389	...}		
390			
391	Ext-CallBarFeatureList ::=		
392	SEQUENCE SIZE (1..maxNumOfExt-BasicServiceGroups) OF		
393	Ext-CallBarringFeature		
394			
395	Ext-CallBarringFeature ::= SEQUENCE {		
396	basicService	Ext-BasicServiceCode	OPTIONAL,
397	ss-Status	[4] Ext-SS-Status,	
398	extensionContainer	ExtensionContainer	OPTIONAL,
399	...}		
400			
401	CUG-Info ::= SEQUENCE {		
402	cug-SubscriptionList	CUG-SubscriptionList,	
403	cug-FeatureList	CUG-FeatureList	OPTIONAL,
404	extensionContainer	[0] ExtensionContainer	OPTIONAL,
405	...}		
406			
407	CUG-SubscriptionList ::= SEQUENCE SIZE (0..maxNumOfCUG) OF		
408	CUG-Subscription		
409			
410	CUG-Subscription ::= SEQUENCE {		
411	cug-Index	CUG-Index,	
412	cug-Interlock	CUG-Interlock,	
413	intraCUG-Options	IntraCUG-Options,	
414	basicServiceGroupList	Ext-BasicServiceGroupList	OPTIONAL,
415	extensionContainer	[0] ExtensionContainer	OPTIONAL,
416	...}		
417			
418	CUG-Index ::= INTEGER (0..32767)		
419	-- The internal structure is defined in ETS 300 138.		
420			
421	CUG-Interlock ::= OCTET STRING (SIZE (4))		
422			
423	IntraCUG-Options ::= ENUMERATED {		
424	noCUG-Restrictions (0),		
425	cugIC-CallBarred (1),		
426	cugOG-CallBarred (2)}		
427			
428	maxNumOfCUG INTEGER ::= 10		
429			
430	CUG-FeatureList ::= SEQUENCE SIZE (1..maxNumOfExt-BasicServiceGroups) OF		
431	CUG-Feature		
432			
433	Ext-BasicServiceGroupList ::= SEQUENCE SIZE (1..maxNumOfExt-BasicServiceGroups)		
434	OF		
435	Ext-BasicServiceCode		
436			
437	maxNumOfExt-BasicServiceGroups INTEGER ::= 32		
438			
439	CUG-Feature ::= SEQUENCE {		
440	basicService	Ext-BasicServiceCode	OPTIONAL,
441	preferentialCUG-Indicator	CUG-Index	OPTIONAL,
442	interCUG-Restrictions	InterCUG-Restrictions,	
443	extensionContainer	ExtensionContainer	OPTIONAL,
444	...}		
445			
446	InterCUG-Restrictions ::= OCTET STRING (SIZE (1))		
447			
448	-- bits 876543: 000000 (unused)		
449	-- Exception handling:		
450	-- bits 876543 shall be ignored if received and not understood		
451			
452	-- bits 21		
453	-- 00 CUG only facilities		
454	-- 01 CUG with outgoing access		
455	-- 10 CUG with incoming access		
456	-- 11 CUG with both outgoing and incoming access		
457			


```

458 Ext-SS-Data ::= SEQUENCE {
459     ss-Code                SS-Code,
460     ss-Status              [4] Ext-SS-Status,
461     ss-SubscriptionOption  SS-SubscriptionOption    OPTIONAL,
462     basicServiceGroupList  Ext-BasicServiceGroupList OPTIONAL,
463     extensionContainer     [5] ExtensionContainer    OPTIONAL,
464     ...}
465
466 ZoneCodeList ::= SEQUENCE SIZE (1..maxNumOfZoneCodes)
467                 OF ZoneCode
468
469 ZoneCode ::= OCTET STRING (SIZE (2))
470 -- internal structure is defined in TS GSM 03.03
471
472 maxNumOfZoneCodes INTEGER ::= 10
473
474 InsertSubscriberDataRes ::= SEQUENCE {
475     teleserviceList        [1] TeleserviceList        OPTIONAL,
476     bearerServiceList      [2] BearerServiceList      OPTIONAL,
477     ss-List                [3] SS-List                OPTIONAL,
478     odb-GeneralData        [4] ODB-GeneralData        OPTIONAL,
479     regionalSubscriptionResponse [5]
480         RegionalSubscriptionResponse    OPTIONAL,
481     supportedCamelPhases   [6] SupportedCamelPhases   OPTIONAL,
482     extensionContainer     [7] ExtensionContainer     OPTIONAL,
483     ...}
484
485 RegionalSubscriptionResponse ::= ENUMERATED {
486     msc-AreaRestricted      (0),
487     tooManyZoneCodes       (1),
488     zoneCodesConflict      (2),
489     regionalSubscNotSupported (3)}
490
491 DeleteSubscriberDataArg ::= SEQUENCE {
492     imsi                  [0] IMSI,
493     basicServiceList      [1] BasicServiceList        OPTIONAL,
494     -- The exception handling for reception of unsupported / not allocated
495     -- basicServiceCodes is defined in section 6.8.2
496     ss-List              [2] SS-List                OPTIONAL,
497     roamingRestrictionDueToUnsupportedFeature [4] NULL    OPTIONAL,
498     regionalSubscriptionIdentifier [5] ZoneCode        OPTIONAL,
499     vbsGroupIndication    [7] NULL                  OPTIONAL,
500     vgcsGroupIndication   [8] NULL                  OPTIONAL,
501     camelSubscriptionInfoWithdraw [9] NULL            OPTIONAL,
502     extensionContainer    [6] ExtensionContainer     OPTIONAL,
503     ...}
504
505 BasicServiceList ::= SEQUENCE SIZE (1..maxNumOfBasicServices) OF
506                     Ext-BasicServiceCode
507
508 maxNumOfBasicServices INTEGER ::= 70
509
510 DeleteSubscriberDataRes ::= SEQUENCE {
511     regionalSubscriptionResponse [0]
512         RegionalSubscriptionResponse    OPTIONAL,
513     extensionContainer          [1]
514         ExtensionContainer              OPTIONAL,
515     ...}
516
517 VlrCamelSubscriptionInfo ::= SEQUENCE {
518     o-CSI [0] O-CSI                OPTIONAL,
519     extensionContainer [1] ExtensionContainer    OPTIONAL,
520     ...}
521
522 O-CSI ::= SEQUENCE {
523     o-BcsmCamelTDPDataList O-BcsmCamelTDPDataList,
524     extensionContainer     ExtensionContainer        OPTIONAL,
525     ...}
526
527 O-BcsmCamelTDPDataList ::= SEQUENCE SIZE (1..maxNumOfCamelTDPData) OF
528     O-BcsmCamelTDPData
529
530 maxNumOfCamelTDPData INTEGER ::= 10

```

```

531 O-BcsmCamelTDPData ::= SEQUENCE {
532     o-BcsmTriggerDetectionPoint      O-BcsmTriggerDetectionPoint,
533     serviceKey                       ServiceKey,
534     gsmSCF-Address                   [0] ISDN-AddressString,
535     defaultCallHandling               [1] DefaultCallHandling,
536     extensionContainer                 [2] ExtensionContainer      OPTIONAL,
537     ...}
538
539 ServiceKey ::= INTEGER (0..2147483647)
540
541 O-BcsmTriggerDetectionPoint ::= ENUMERATED {
542     collectedInfo (2),
543     ... }
544 -- exception handling:
545 -- For O-BcsmCamelTDPData sequences containing this parameter with any
546 -- other value than the ones listed the receiver shall ignore the whole
547 -- O-BcsmCamelTDPDataSequence.
548
549 DefaultCallHandling ::= ENUMERATED {
550     continueCall (0) ,
551     releaseCall (1) ,
552     ...}
553 -- exception handling:
554 -- reception of values in range 2-31 shall be treated as "continueCall"
555 -- reception of values greater than 31 shall be treated as "releaseCall"
556
557 SupportedCamelPhases ::= BIT STRING {
558     phase1 (0) } (SIZE (1..16))
559
560 -- fault recovery types
561
562
563 ResetArg ::= SEQUENCE {
564     hlr-Number          ISDN-AddressString,
565     hlr-List             HLR-List      OPTIONAL,
566     ...}
567
568 RestoreDataArg ::= SEQUENCE {
569     imsi                IMSI,
570     lmsi                LMSI      OPTIONAL,
571     extensionContainer   ExtensionContainer  OPTIONAL,
572     ...}
573
574 RestoreDataRes ::= SEQUENCE {
575     hlr-Number          ISDN-AddressString,
576     msNotReachable      NULL      OPTIONAL,
577     extensionContainer   ExtensionContainer  OPTIONAL,
578     ...}
579
580 -- VBS/VGCS types
581 VBSDataList ::= SEQUENCE SIZE (1..maxNumOfVBSGroupIds)
582                OF VoiceBroadcastData
583
584 VGCSDataList ::= SEQUENCE SIZE (1..maxNumOfVGCSGroupIds)
585                OF VoiceGroupCallData
586
587 maxNumOfVBSGroupIds  INTEGER ::= 50
588
589 maxNumOfVGCSGroupIds INTEGER ::= 50
590
591 VoiceGroupCallData ::= SEQUENCE {
592     groupId             GroupId,
593     extensionContainer   ExtensionContainer  OPTIONAL,
594     ...}
595
596 VoiceBroadcastData ::= SEQUENCE {
597     groupId             GroupId,
598     broadcastInitEntitlement  NULL      OPTIONAL,
599     extensionContainer   ExtensionContainer  OPTIONAL,
600     ...}
601
602 GroupId ::= OCTET STRING (SIZE (3))
603     -- Refers to the Group Identification as specified in GSM TS 03.03
604     -- and 03.68/ 03.69
605
606 -- provide subscriber info types
607

```

```

608 ProvideSubscriberInfoArg ::= SEQUENCE {
609     imsi      [0] IMSI,
610     lmsi      [1] LMSI
611     requestedInfo      [2] RequestedInfo,
612     extensionContainer [3] ExtensionContainer OPTIONAL,
613     ...}
614
615 ProvideSubscriberInfoRes ::= SEQUENCE {
616     subscriberInfo      SubscriberInfo,
617     extensionContainer  ExtensionContainer OPTIONAL,
618     ...}
619
620 SubscriberInfo ::= SEQUENCE {
621     locationInformation [0] LocationInformation OPTIONAL,
622     subscriberState    [1] SubscriberState   OPTIONAL,
623     extensionContainer [2] ExtensionContainer  OPTIONAL,
624     ...}
625
626 RequestedInfo ::= SEQUENCE {
627     locationInformation [0] NULL OPTIONAL,
628     subscriberState    [1] NULL OPTIONAL,
629     extensionContainer [2] ExtensionContainer OPTIONAL,
630     ...}
631
632 LocationInformation ::= SEQUENCE {
633     ageOfLocationInformation AgeOfLocationInformation OPTIONAL,
634     geographicalInformation  [0] GeographicalInformation OPTIONAL,
635     vlr-number              [1] ISDN-AddressString   OPTIONAL,
636     locationNumber          [2] LocationNumber      OPTIONAL,
637     cellIdOrLAI             [3] CellIdOrLAI         OPTIONAL,
638     extensionContainer       [4] ExtensionContainer  OPTIONAL,
639     ...}
640
641 AgeOfLocationInformation ::= INTEGER (0..32767)
642 -- the value represents the elapsed time in minutes since the last
643 -- network contact of the mobile station (i.e. the actuality of the
644 -- location information).
645 -- value "0" indicates that the MS is currently in contact with the
646 -- network
647 -- value "32767" indicates that the location information is at least
648 -- 32767 minutes old
649
650 GeographicalInformation ::= OCTET STRING (SIZE (8))
651 -- Refers to geographical Information defined in GSM 03.32.
652 -- Only the description of an ellipsoid point with uncertainty circle
653 -- as specified in GSM 03.32 is allowed to be used
654 -- The internal structure according to GSM 03.32 is as follows:
655 --     Type of shape (ellipsoid point with uncertainty circle) 1 octet
656 --     Degrees of Latitude                                     3 octets
657 --     Degrees of Longitude                                   3 octets
658 --     Uncertainty code                                       1 octet
659
660 LocationNumber ::= OCTET STRING (SIZE (2..10))
661 -- the internal structure is defined in CCITT Rec Q.763
662
663 SubscriberState ::= CHOICE {
664     assumedIdle [0] NULL,
665     camelBusy   [1] NULL,
666     netDetNotReachable NotReachableReason,
667     notProvidedFromVLR [2] NULL}
668
669 NotReachableReason ::= ENUMERATED {
670     msPurged (0),
671     imsiDetached (1),
672     restrictedArea (2),
673     notRegistered (3)}
674
675 -- any time interrogation info types
676
677 AnyTimeInterrogationArg ::= SEQUENCE {
678     subscriberIdentity [0] SubscriberIdentity,
679     requestedInfo      [1] RequestedInfo,
680     gsmSCF-Address     [3] ISDN-AddressString,
681     extensionContainer [2] ExtensionContainer OPTIONAL,
682     ...}
683

```

```
684 AnyTimeInterrogationRes ::= SEQUENCE {
685     subscriberInfo      SubscriberInfo,
686     extensionContainer   ExtensionContainer OPTIONAL,
687     ...}
688
689 SubscriberIdentity ::= CHOICE {
690     imsi      [0] IMSI,
691     msisdn    [1] ISDN-AddressString
692 }
693
694 END
```

14.7.2 Operation and maintenance data types

```

1 MAP-OM-DataTypes {
2   ccitt identified-organization (4) etsi (0) mobileDomain (0)
3   gsm-Network (1) modules (3) map-OM-DataTypes (12) version3 (3)}
4
5 DEFINITIONS
6
7 IMPLICIT TAGS
8
9 ::=
10
11 BEGIN
12
13 EXPORTS
14   ActivateTraceModeArg,
15   ActivateTraceModeRes,
16   DeactivateTraceModeArg,
17   DeactivateTraceModeRes
18 ;
19
20 IMPORTS
21   AddressString,
22   IMSI
23 FROM MAP-CommonDataTypes {
24   ccitt identified-organization (4) etsi (0) mobileDomain (0)
25   gsm-Network (1) modules (3) map-CommonDataTypes (18) version3 (3)}
26
27   ExtensionContainer
28 FROM MAP-ExtensionDataTypes {
29   ccitt identified-organization (4) etsi (0) mobileDomain (0)
30   gsm-Network (1) modules (3) map-ExtensionDataTypes (21) version3 (3)}
31
32
33 ;
34
35
36 ActivateTraceModeArg ::= SEQUENCE {
37   imsi                [0] IMSI                OPTIONAL,
38   traceReference      [1] TraceReference,
39   traceType           [2] TraceType,
40   omc-Id              [3] AddressString        OPTIONAL,
41   extensionContainer  [4] ExtensionContainer  OPTIONAL,
42   ...}
43
44 TraceReference ::= OCTET STRING (SIZE (1..2))
45
46 TraceType ::= INTEGER
47   (0..255)
48   -- Trace types are fully defined in TS GSM 12.08.
49
50 ActivateTraceModeRes ::= SEQUENCE {
51   extensionContainer  [0] ExtensionContainer  OPTIONAL,
52   ...}
53
54 DeactivateTraceModeArg ::= SEQUENCE {
55   imsi                [0] IMSI                OPTIONAL,
56   traceReference      [1] TraceReference,
57   extensionContainer  [2] ExtensionContainer  OPTIONAL,
58   ...}
59
60 DeactivateTraceModeRes ::= SEQUENCE {
61   extensionContainer  [0] ExtensionContainer  OPTIONAL,
62   ...}
63
64 END

```

14.7.3 Call handling data types

```

1 MAP-CH-DataTypes {
2   ccitt identified-organization (4) etsi (0) mobileDomain (0)
3   gsm-Network (1) modules (3) map-CH-DataTypes (13) version3 (3)}
4
5 DEFINITIONS
6
7 IMPLICIT TAGS
8
9 ::=
10
11 BEGIN
12
13 EXPORTS
14   SendRoutingInfoArg,
15   SendRoutingInfoRes,
16   ProvideRoamingNumberArg,
17   ProvideRoamingNumberRes,
18   ResumeCallHandlingArg,
19   ResumeCallHandlingRes,
20   NumberOfForwarding,
21   SuppressionOfAnnouncement,
22   CallReferenceNumber
23 ;
24
25 IMPORTS
26   maxNumOfCamelTDPData,
27   SubscriberInfo,
28   ServiceKey,
29   DefaultCallHandling,
30   SupportedCamelPhases,
31   CUG-Interlock,
32   O-CSI
33 FROM MAP-MS-DataTypes {
34   ccitt identified-organization (4) etsi (0) mobileDomain (0)
35   gsm-Network (1) modules (3) map-MS-DataTypes (11) version3 (3)}
36
37   ForwardingOptions,
38   SS-List
39 FROM MAP-SS-DataTypes {
40   ccitt identified-organization (4) etsi (0) mobileDomain (0)
41   gsm-Network (1) modules (3) map-SS-DataTypes (14) version3 (3)}
42
43   ISDN-AddressString,
44   ISDN-SubaddressString,
45   ExternalSignalInfo,
46   IMSI,
47   LMSI,
48   Ext-BasicServiceCode
49 FROM MAP-CommonDataTypes {
50   ccitt identified-organization (4) etsi (0) mobileDomain (0)
51   gsm-Network (1) modules (3) map-CommonDataTypes (18) version3 (3)}
52
53   ExtensionContainer
54 FROM MAP-ExtensionDataTypes {
55   ccitt identified-organization (4) etsi (0) mobileDomain (0)
56   gsm-Network (1) modules (3) map-ExtensionDataTypes (21) version3 (3)}
57 ;
58
59
60 CUG-CheckInfo ::= SEQUENCE {
61   cug-Interlock          CUG-Interlock,
62   cug-OutgoingAccess     NULL,
63   extensionContainer     ExtensionContainer
64   ...}
65
66 NumberOfForwarding ::= INTEGER (1..5)
67

```

```

68 SendRoutingInfoArg ::= SEQUENCE {
69     msisdn                [0] ISDN-AddressString,
70     cug-CheckInfo         [1] CUG-CheckInfo          OPTIONAL,
71     numberOfForwarding    [2] NumberOfForwarding      OPTIONAL,
72     interrogationType     [3] InterrogationType,
73     or-Interrogation      [4] NULL                    OPTIONAL,
74     or-Capability         [5] OR-Phase                OPTIONAL,
75     gsmc-Address          [6] ISDN-AddressString,
76     callReferenceNumber   [7] CallReferenceNumber      OPTIONAL,
77     forwardingReason      [8] ForwardingReason        OPTIONAL,
78     basicServiceGroup     [9] Ext-BasicServiceCode     OPTIONAL,
79     networkSignalInfo     [10] ExternalSignalInfo      OPTIONAL,
80     camelInfo             [11] CamelInfo               OPTIONAL,
81     suppressionOfAnnouncement [12] SuppressionOfAnnouncement OPTIONAL,
82     extensionContainer     [13] ExtensionContainer      OPTIONAL,
83     ...}
84
85 SuppressionOfAnnouncement ::= NULL
86
87 InterrogationType ::= ENUMERATED {
88     basicCall (0),
89     forwarding (1)}
90
91 OR-Phase ::= INTEGER (1..127)
92
93 CallReferenceNumber ::= OCTET STRING (SIZE (1..8))
94
95 ForwardingReason ::= ENUMERATED {
96     notReachable (0),
97     busy (1),
98     noReply (2)}
99
100
101 SendRoutingInfoRes ::= [3] SEQUENCE {
102     imsi                [9] IMSI                      OPTIONAL,
103     -- IMSI must be present if SendRoutingInfoRes is not segmented.
104     -- If the TC-Result-NL segmentation option is taken the IMSI must be
105     -- present in one segmented transmission of SendRoutingInfoRes.
106     extendedRoutingInfo ExtendedRoutingInfo          OPTIONAL,
107     cug-CheckInfo        [3] CUG-CheckInfo            OPTIONAL,
108     cugSubscriptionFlag  [6] NULL                     OPTIONAL,
109     subscriberInfo       [7] SubscriberInfo           OPTIONAL,
110     ss-List              [1] SS-List                  OPTIONAL,
111     basicService         [5] Ext-BasicServiceCode     OPTIONAL,
112     forwardingInterrogationRequired [4] NULL           OPTIONAL,
113     vmsc-Address         [2] ISDN-AddressString       OPTIONAL,
114     extensionContainer    [0] ExtensionContainer      OPTIONAL,
115     ...}
116
117 RoutingInfo ::= CHOICE {
118     roamingNumber        ISDN-AddressString,
119     forwardingData       ForwardingData}
120
121 ForwardingData ::= SEQUENCE {
122     forwardedToNumber     [5] ISDN-AddressString      OPTIONAL,
123     forwardedToSubaddress [4] ISDN-SubaddressString  OPTIONAL,
124     forwardingOptions     [6] ForwardingOptions       OPTIONAL,
125     extensionContainer    [7] ExtensionContainer      OPTIONAL,
126     ...}
127
128 ProvideRoamingNumberArg ::= SEQUENCE {
129     imsi                [0] IMSI,
130     msc-Number          [1] ISDN-AddressString,
131     msisdn              [2] ISDN-AddressString        OPTIONAL,
132     lmsi                [4] LMSI                      OPTIONAL,
133     gsm-BearerCapability [5] ExternalSignalInfo       OPTIONAL,
134     networkSignalInfo   [6] ExternalSignalInfo       OPTIONAL,
135     suppressionOfAnnouncement [7] SuppressionOfAnnouncement OPTIONAL,
136     gsmc-Address        [8] ISDN-AddressString        OPTIONAL,
137     callReferenceNumber [9] CallReferenceNumber        OPTIONAL,
138     or-Interrogation    [10] NULL                     OPTIONAL,
139     extensionContainer  [11] ExtensionContainer        OPTIONAL,
140     ...}
141
142 ProvideRoamingNumberRes ::= SEQUENCE {
143     roamingNumber        ISDN-AddressString,
144     extensionContainer    ExtensionContainer          OPTIONAL,
145     ...}
146
  
```

```

147
148 ResumeCallHandlingArg ::= SEQUENCE {
149     callReferenceNumber      [0] CallReferenceNumber,
150     basicServiceGroup        [1] Ext-BasicServiceCode,
151     forwardingData           [2] ForwardingData,
152     imsi                     [3] IMSI,
153     cug-CheckInfo            [4] CUG-CheckInfo           OPTIONAL,
154     o-CSI                    [5] O-CSI                   OPTIONAL,
155     extensionContainer        [7] ExtensionContainer      OPTIONAL,
156     ...}
157
158 ResumeCallHandlingRes ::= SEQUENCE {
159     extensionContainer        ExtensionContainer           OPTIONAL,
160     ...}
161
162 CamelInfo ::= SEQUENCE {
163     supportedCamelPhases      SupportedCamelPhases,
164     suppress-T-CSI           NULL                      OPTIONAL,
165     extensionContainer        ExtensionContainer          OPTIONAL,
166     ...}
167
168 ExtendedRoutingInfo ::= CHOICE {
169     routingInfo               RoutingInfo,
170     camelRoutingInfo          [8] CamelRoutingInfo}
171
172 CamelRoutingInfo ::= SEQUENCE {
173     forwardingData            ForwardingData             OPTIONAL,
174     gsmcCamelSubscriptionInfo [0] GsmcCamelSubscriptionInfo,
175     extensionContainer        [1] ExtensionContainer      OPTIONAL,
176     ...}
177
178 GsmcCamelSubscriptionInfo ::= SEQUENCE {
179     t-CSI                     [0] T-CSI                   OPTIONAL,
180     o-CSI                     [1] O-CSI                   OPTIONAL,
181     extensionContainer        [2] ExtensionContainer      OPTIONAL,
182     ...}
183
184 T-CSI ::= SEQUENCE {
185     t-BcsmCamelTDPDataList    T-BcsmCamelTDPDataList,
186     extensionContainer        ExtensionContainer           OPTIONAL,
187     ...}
188
189 T-BcsmCamelTDPDataList ::= SEQUENCE SIZE (1..maxNumOfCamelTDPData) OF
190     T-BcsmCamelTDPData
191
192 T-BcsmCamelTDPData ::= SEQUENCE {
193     t-BcsmTriggerDetectionPoint T-BcsmTriggerDetectionPoint,
194     serviceKey                  ServiceKey,
195     gsmSCF-Address              [0] ISDN-AddressString,
196     defaultCallHandling         [1] DefaultCallHandling,
197     extensionContainer          [2] ExtensionContainer      OPTIONAL,
198     ...}
199
200 T-BcsmTriggerDetectionPoint ::= ENUMERATED {
201     termAttemptAuthorized(12),
202     ...}
203 -- exception handling:
204 -- For T-BcsmCamelTDPData sequences containing this parameter with any other
205 -- value than the ones listed the receiver shall ignore the whole
206 -- T-BcsmCamelTDPData sequence.
207
208 END

```


14.7.4 Supplementary service data types

```

1 MAP-SS-DataTypes {
2   ccitt identified-organization (4) etsi (0) mobileDomain (0)
3   gsm-Network (1) modules (3) map-SS-DataTypes (14) version3 (3)}
4
5 DEFINITIONS
6
7 IMPLICIT TAGS
8
9 ::=
10
11 BEGIN
12
13 EXPORTS
14   RegisterSS-Arg,
15   SS-Info,
16   SS-Status,
17   SS-SubscriptionOption,
18   SS-ForBS-Code,
19   InterrogateSS-Res,
20   USSD-Arg,
21   USSD-Res,
22   Password,
23   GuidanceInfo,
24   SS-List,
25   SS-InfoList,
26   OverrideCategory,
27   CliRestrictionOption,
28   NoReplyConditionTime,
29   ForwardingOptions,
30   maxNumOfSS,
31   SS-Data,
32   USSD-DataCodingScheme,
33   USSD-String
34 ;
35
36 IMPORTS
37   AddressString,
38   ISDN-AddressString,
39   ISDN-SubaddressString,
40   BasicServiceCode
41 FROM MAP-CommonDataTypes {
42   ccitt identified-organization (4) etsi (0) mobileDomain (0)
43   gsm-Network (1) modules (3) map-CommonDataTypes (18) version3 (3)}
44
45   SS-Code
46 FROM MAP-SS-Code {
47   ccitt identified-organization (4) etsi (0) mobileDomain (0)
48   gsm-Network (1) modules (3) map-SS-Code (15) version3 (3)}
49 ;
50
51
52 RegisterSS-Arg ::= SEQUENCE{
53   ss-Code                SS-Code,
54   basicService            BasicServiceCode                OPTIONAL,
55   forwardedToNumber       [4] AddressString                OPTIONAL,
56   forwardedToSubaddress   [6] ISDN-SubaddressString        OPTIONAL,
57   noReplyConditionTime    [5] NoReplyConditionTime          OPTIONAL,
58   ...}
59
60 NoReplyConditionTime ::= INTEGER (5..30)
61
62 SS-Info ::= CHOICE {
63   forwardingInfo          [0] ForwardingInfo,
64   callBarringInfo        [1] CallBarringInfo,
65   ss-Data                 [3] SS-Data}
66
67 ForwardingInfo ::= SEQUENCE {
68   ss-Code                SS-Code                OPTIONAL,
69   forwardingFeatureList   ForwardingFeatureList,
70   ...}
71
72 ForwardingFeatureList ::=
73   SEQUENCE SIZE (1..maxNumOfBasicServiceGroups) OF
74   ForwardingFeature
75

```

```

76 ForwardingFeature ::= SEQUENCE {
77     basicService          BasicServiceCode          OPTIONAL,
78     ss-Status             [4] SS-Status             OPTIONAL,
79     forwardedToNumber     [5] ISDN-AddressString     OPTIONAL,
80     forwardedToSubaddress [8] ISDN-SubaddressString  OPTIONAL,
81     forwardingOptions     [6] ForwardingOptions      OPTIONAL,
82     noReplyConditionTime  [7] NoReplyConditionTime   OPTIONAL,
83     ...}

```

```

85 SS-Status ::= OCTET STRING (SIZE (1))
86
87 -- bits 8765: 0000 (unused)
88 -- bits 4321: Used to convey the "P bit","R bit","A bit" and "Q bit",
89 --             representing supplementary service state information
90 --             as defined in TS GSM 03.11
91
92 -- bit 4: "Q bit"
93
94 -- bit 3: "P bit"
95
96 -- bit 2: "R bit"
97
98 -- bit 1: "A bit"

```

```

100 ForwardingOptions ::= OCTET STRING (SIZE (1))
101
102 -- bit 8: notification to forwarding party
103 -- 0 no notification
104 -- 1 notification
105
106 -- bit 7: 0 (unused)
107
108 -- bit 6: notification to calling party
109 -- 0 no notification
110 -- 1 notification
111
112 -- bit 5: 0 (unused)
113
114 -- bits 43: forwarding reason
115 -- 00 ms not reachable
116 -- 01 ms busy
117 -- 10 no reply
118 -- 11 unconditional
119 -- bits 21: 00 (unused)

```

```

121 CallBarringInfo ::= SEQUENCE {
122     ss-Code          SS-Code          OPTIONAL,
123     callBarringFeatureList CallBarringFeatureList,
124     ...}

```

```

126 CallBarringFeatureList ::=
127     SEQUENCE SIZE (1..maxNumOfBasicServiceGroups) OF
128         CallBarringFeature

```

```

130 CallBarringFeature ::= SEQUENCE {
131     basicService          BasicServiceCode          OPTIONAL,
132     ss-Status             [4] SS-Status             OPTIONAL,
133     ...}

```

```

135 SS-Data ::= SEQUENCE {
136     ss-Code          SS-Code          OPTIONAL,
137     ss-Status        [4] SS-Status     OPTIONAL,
138     ss-SubscriptionOption SS-SubscriptionOption OPTIONAL,
139     basicServiceGroupList BasicServiceGroupList OPTIONAL,
140     ...}

```

```

142 SS-SubscriptionOption ::= CHOICE {
143     cliRestrictionOption [2] CliRestrictionOption,
144     overrideCategory     [1] OverrideCategory}

```

```

146 CliRestrictionOption ::= ENUMERATED {
147     permanent (0),
148     temporaryDefaultRestricted (1),
149     temporaryDefaultAllowed (2)}

```

```

151 OverrideCategory ::= ENUMERATED {
152     overrideEnabled (0),
153     overrideDisabled (1)}

```

```

155 SS-ForBS-Code ::= SEQUENCE {
156     ss-Code                SS-Code,
157     basicService           BasicServiceCode          OPTIONAL,
158     ...}
159
160 Cli-RestrictionInfo ::= SEQUENCE {
161     ss-Status              SS-Status,
162     cliRestrictionOption   CliRestrictionOption      OPTIONAL,
163     ...}
164
165 InterrogateSS-Res ::= CHOICE {
166     ss-Status              [0] SS-Status,
167     basicServiceGroupList [2] BasicServiceGroupList,
168     forwardingFeatureList [3] ForwardingFeatureList,
169     cli-RestrictionInfo    [4] Cli-RestrictionInfo}
170
171 USSD-Arg ::= SEQUENCE {
172     ussd-DataCodingScheme  USSD-DataCodingScheme,
173     ussd-String            USSD-String,
174     ...}
175
176 USSD-Res ::= SEQUENCE {
177     ussd-DataCodingScheme  USSD-DataCodingScheme,
178     ussd-String            USSD-String,
179     ...}
180
181 USSD-DataCodingScheme ::= OCTET STRING (SIZE (1))
182     -- The structure of the USSD-DataCodingScheme is defined by
183     -- the Cell Broadcast Data Coding Scheme as described in
184     -- TS GSM 03.38
185
186 USSD-String ::= OCTET STRING (SIZE (1..maxUSSD-StringLength))
187     -- The structure of the contents of the USSD-String is dependent
188     -- on the USSD-DataCodingScheme as described in TS GSM 03.38.
189
190 maxUSSD-StringLength INTEGER ::= 160
191
192 Password ::= NumericString
193     (FROM ("0"|"1"|"2"|"3"|"4"|"5"|"6"|"7"|"8"|"9"))
194     (SIZE (4))
195
196 GuidanceInfo ::= ENUMERATED {
197     enterPW (0),
198     enterNewPW (1),
199     enterNewPW-Again (2)}
200     -- How this information is really delivered to the subscriber
201     -- (display, announcement, ...) is not part of this
202     -- specification.
203
204 SS-List ::= SEQUENCE SIZE (1..maxNumOfSS) OF
205     SS-Code
206
207 maxNumOfSS INTEGER ::= 30
208
209 SS-InfoList ::= SEQUENCE SIZE (1..maxNumOfSS) OF
210     SS-Info
211
212 BasicServiceGroupList ::= SEQUENCE SIZE (1..maxNumOfBasicServiceGroups) OF
213     BasicServiceCode
214
215 maxNumOfBasicServiceGroups INTEGER ::= 13
216
217 END

```

14.7.5 Supplementary service codes

```

1 MAP-SS-Code {
2   ccitt-identified-organization (4) etsi (0) mobileDomain (0)
3   gsm-Network (1) modules (3) map-SS-Code (15) version3 (3)}
4
5 DEFINITIONS
6
7 ::=
8
9 BEGIN
10
11 SS-Code ::= OCTET STRING (SIZE (1))
12   -- This type is used to represent the code identifying a single
13   -- supplementary service, a group of supplementary services, or
14   -- all supplementary services. The services and abbreviations
15   -- used are defined in TS GSM 02.04. The internal structure is
16   -- defined as follows:
17   --
18   -- bits 87654321: group (bits 8765), and specific service
19   -- (bits 4321)
20
21 allSS SS-Code ::= '00000000'B
22   -- reserved for possible future use
23   -- all SS
24
25 allLineIdentificationSS SS-Code ::= '00010000'B
26   -- reserved for possible future use
27   -- all line identification SS
28 clip SS-Code ::= '00010001'B
29   -- calling line identification presentation
30 clir SS-Code ::= '00010010'B
31   -- calling line identification restriction
32 colp SS-Code ::= '00010011'B
33   -- connected line identification presentation
34 colr SS-Code ::= '00010100'B
35   -- connected line identification restriction
36 mci SS-Code ::= '00010101'B
37   -- reserved for possible future use
38   -- malicious call identification
39
40
41 allForwardingSS SS-Code ::= '00100000'B
42   -- all forwarding SS
43 cfu SS-Code ::= '00100001'B
44   -- call forwarding unconditional
45 allCondForwardingSS SS-Code ::= '00101000'B
46   -- all conditional forwarding SS
47 cfb SS-Code ::= '00101001'B
48   -- call forwarding on mobile subscriber busy
49 cfny SS-Code ::= '00101010'B
50   -- call forwarding on no reply
51 cfnc SS-Code ::= '00101011'B
52   -- call forwarding on mobile subscriber not reachable
53
54 allCallOfferingSS SS-Code ::= '00110000'B
55   -- reserved for possible future use
56   -- all call offering SS includes also all forwarding SS
57 ect SS-Code ::= '00110001'B
58   -- explicit call transfer
59 mah SS-Code ::= '00110010'B
60   -- reserved for possible future use
61   -- mobile access hunting
62
63 allCallCompletionSS SS-Code ::= '01000000'B
64   -- reserved for possible future use
65   -- all Call completion SS
66 cw SS-Code ::= '01000001'B
67   -- call waiting
68 hold SS-Code ::= '01000010'B
69   -- call hold
70 ccbs SS-Code ::= '01000011'B
71   -- reserved for possible future use
72   -- completion of call to busy subscribers
73

```

```

74 allMultiPartySS SS-Code ::= '01010000'B
75 -- reserved for possible future use
76 -- all multiparty SS
77 multiPTY SS-Code ::= '01010001'B
78 -- multiparty
79
80 allCommunityOfInterest-SS SS-Code ::= '01100000'B
81 -- reserved for possible future use
82 -- all community of interest SS
83 cug SS-Code ::= '01100001'B
84 -- closed user group
85
86 allChargingSS SS-Code ::= '01110000'B
87 -- reserved for possible future use
88 -- all charging SS
89 aoci SS-Code ::= '01110001'B
90 -- advice of charge information
91 aocc SS-Code ::= '01110010'B
92 -- advice of charge charging
93
94 allAdditionalInfoTransferSS SS-Code ::= '10000000'B
95 -- reserved for possible future use
96 -- all additional information transfer SS
97 uus SS-Code ::= '10000001'B
98 -- reserved for possible future use
99 -- UUS user-to-user signalling
100
101 allBarringSS SS-Code ::= '10010000'B
102 -- all barring SS
103 barringOfOutgoingCalls SS-Code ::= '10010001'B
104 -- barring of outgoing calls
105 baoc SS-Code ::= '10010010'B
106 -- barring of all outgoing calls
107 boic SS-Code ::= '10010011'B
108 -- barring of outgoing international calls
109 boicExHC SS-Code ::= '10010100'B
110 -- barring of outgoing international calls except those directed
111 -- to the home PLMN
112 barringOfIncomingCalls SS-Code ::= '10011001'B
113 -- barring of incoming calls
114 baic SS-Code ::= '10011010'B
115 -- barring of all incoming calls
116 bicRoam SS-Code ::= '10011011'B
117 -- barring of incoming calls when roaming outside home PLMN
118 -- Country
119
120 allPLMN-specificSS SS-Code ::= '11110000'B
121 plmn-specificSS-1 SS-Code ::= '11110001'B
122 plmn-specificSS-2 SS-Code ::= '11110010'B
123 plmn-specificSS-3 SS-Code ::= '11110011'B
124 plmn-specificSS-4 SS-Code ::= '11110100'B
125 plmn-specificSS-5 SS-Code ::= '11110101'B
126 plmn-specificSS-6 SS-Code ::= '11110110'B
127 plmn-specificSS-7 SS-Code ::= '11110111'B
128 plmn-specificSS-8 SS-Code ::= '11111000'B
129 plmn-specificSS-9 SS-Code ::= '11111001'B
130 plmn-specificSS-A SS-Code ::= '11111010'B
131 plmn-specificSS-B SS-Code ::= '11111011'B
132 plmn-specificSS-C SS-Code ::= '11111100'B
133 plmn-specificSS-D SS-Code ::= '11111101'B
134 plmn-specificSS-E SS-Code ::= '11111110'B
135 plmn-specificSS-F SS-Code ::= '11111111'B
136
137 allCallPrioritySS SS-Code ::= '10100000'B
138 -- reserved for possible future use
139 -- all call priority SS
140 emlpp SS-Code ::= '10100001'B
141 -- enhanced Multilevel Precedence Pre-emption (EMLPP) service
142
143 END

```

14.7.6 Short message data types

```

1 MAP-SM-DataTypes {
2   ccitt identified-organization (4) etsi (0) mobileDomain (0)
3   gsm-Network (1) modules (3) map-SM-DataTypes (16) version3 (3)}
4
5 DEFINITIONS
6
7 IMPLICIT TAGS
8
9 ::=
10
11 BEGIN
12
13 EXPORTS
14   RoutingInfoForSM-Arg,
15   RoutingInfoForSM-Res,
16   MO-ForwardSM-Arg,
17   MO-ForwardSM-Res,
18   MT-ForwardSM-Arg,
19   MT-ForwardSM-Res,
20   ReportSM-DeliveryStatusArg,
21   ReportSM-DeliveryStatusRes,
22   AlertServiceCentreArg,
23   InformServiceCentreArg,
24   ReadyForSM-Arg,
25   SM-DeliveryOutcome,
26   AlertReason
27 ;
28
29 IMPORTS
30   AddressString,
31   ISDN-AddressString,
32   SignalInfo,
33   IMSI,
34   LMSI
35 FROM MAP-CommonDataTypes {
36   ccitt identified-organization (4) etsi (0) mobileDomain (0)
37   gsm-Network (1) modules (3) map-CommonDataTypes (18) version3 (3)}
38
39   AbsentSubscriberDiagnosticSM
40 FROM MAP-ER-DataTypes {
41   ccitt identified-organization (4) etsi (0) mobileDomain (0)
42   gsm-Network (1) modules (3) map-ER-DataTypes (17) version3 (3)}
43
44   ExtensionContainer
45 FROM MAP-ExtensionDataTypes {
46   ccitt identified-organization (4) etsi (0) mobileDomain (0)
47   gsm-Network (1) modules (3) map-ExtensionDataTypes (21) version3 (3)}
48 ;
49
50

```

RoutingInfoForSM-Arg ::= SEQUENCE {			
msisdn	[0] ISDN-AddressString,		
sm-RP-PRI	[1] BOOLEAN,		
serviceCentreAddress	[2] AddressString,		
extensionContainer	[6] ExtensionContainer	OPTIONAL,	
...}			

RoutingInfoForSM-Res ::= SEQUENCE {			
imsi	IMSI,		
locationInfoWithLMSI	[0] LocationInfoWithLMSI,		
extensionContainer	[4] ExtensionContainer	OPTIONAL,	
...}			

LocationInfoWithLMSI ::= SEQUENCE {			
msc-Number	[1] ISDN-AddressString,		
lmsi	LMSI	OPTIONAL,	
extensionContainer	ExtensionContainer	OPTIONAL,	
...}			

MO-ForwardSM-Arg ::= SEQUENCE {			
sm-RP-DA	SM-RP-DA,		
sm-RP-OA	SM-RP-OA,		
sm-RP-UI	SignalInfo,		
extensionContainer	ExtensionContainer	OPTIONAL,	
...}			

```

77 MO-ForwardSM-Res ::= SEQUENCE {
78     sm-RP-UI                               SignalInfo,                OPTIONAL,
79     extensionContainer                      ExtensionContainer          OPTIONAL,
80     ...}
81
82 MT-ForwardSM-Arg ::= SEQUENCE {
83     sm-RP-DA                               SM-RP-DA,
84     sm-RP-OA                               SM-RP-OA,
85     sm-RP-UI                               SignalInfo,
86     moreMessagesToSend                     NULL,                        OPTIONAL,
87     extensionContainer                      ExtensionContainer          OPTIONAL,
88     ...}
89
90 MT-ForwardSM-Res ::= SEQUENCE {
91     sm-RP-UI                               SignalInfo                OPTIONAL,
92     extensionContainer                      ExtensionContainer          OPTIONAL,
93     ...}
94
95 SM-RP-DA ::= CHOICE {
96     imsi                                   [0] IMSI,
97     lmsi                                   [1] LMSI,
98     serviceCentreAddressDA                [4] AddressString,
99     noSM-RP-DA                           [5] NULL}
100
101 SM-RP-OA ::= CHOICE {
102     msisdn                                [2] ISDN-AddressString,
103     serviceCentreAddressOA                [4] AddressString,
104     noSM-RP-OA                           [5] NULL}
105
106 ReportSM-DeliveryStatusArg ::= SEQUENCE {
107     msisdn                                ISDN-AddressString,
108     serviceCentreAddress                  AddressString,
109     sm-DeliveryOutcome                    SM-DeliveryOutcome,
110     absentSubscriberDiagnosticSM           [0] AbsentSubscriberDiagnosticSM
111                                           OPTIONAL,
112     extensionContainer                    [1] ExtensionContainer          OPTIONAL,
113     ...}
114
115 SM-DeliveryOutcome ::= ENUMERATED {
116     memoryCapacityExceeded (0),
117     absentSubscriber (1),
118     successfulTransfer (2)}
119
120 ReportSM-DeliveryStatusRes ::= SEQUENCE {
121     storedMSISDN                          ISDN-AddressString        OPTIONAL,
122     extensionContainer                      ExtensionContainer          OPTIONAL,
123     ...}
124
125 AlertServiceCentreArg ::= SEQUENCE {
126     msisdn                                ISDN-AddressString,
127     serviceCentreAddress                  AddressString,
128     ...}
129
130 InformServiceCentreArg ::= SEQUENCE {
131     storedMSISDN                          ISDN-AddressString        OPTIONAL,
132     mw-Status                             MW-Status                OPTIONAL,
133     extensionContainer                      ExtensionContainer          OPTIONAL,
134     ...}
135
136 MW-Status ::= BIT STRING {
137     sc-AddressNotIncluded (0),
138     mnrf-Set (1),
139     mcef-Set (2)} (SIZE (6..16))
140 -- exception handling:
141 -- bits 3 to 15 shall be ignored if received and not understood
142
143 ReadyForSM-Arg ::= SEQUENCE {
144     imsi                                   [0] IMSI,
145     alertReason                           AlertReason,
146     ...}
147
148 AlertReason ::= ENUMERATED {
149     ms-Present (0),
150     memoryAvailable (1)}
151
152 END

```

14.7.7 Error data types

```

1 MAP-ER-DataTypes {
2   ccitt identified-organization (4) etsi (0) mobileDomain (0)
3   gsm-Network (1) modules (3) map-ER-DataTypes (17) version3 (3)}
4
5 DEFINITIONS
6
7 IMPLICIT TAGS
8
9 ::=
10
11 BEGIN
12
13 EXPORTS
14   RoamingNotAllowedParam,
15   CallBarredParam,
16   CUG-RejectParam,
17   SS-IncompatibilityCause,
18   PW-RegistrationFailureCause,
19   SM-DeliveryFailureCause,
20   SystemFailureParam,
21   DataMissingParam,
22   UnexpectedDataParam,
23   FacilityNotSupParam,
24   OR-NotAllowedParam,
25   UnknownSubscriberParam,
26   NumberChangedParam,
27   UnidentifiedSubParam,
28   IllegalSubscriberParam,
29   IllegalEquipmentParam,
30   BearerServNotProvParam,
31   TeleservNotProvParam,
32   TracingBufferFullParam,
33   NoRoamingNbParam,
34   AbsentSubscriberParam,
35   BusySubscriberParam,
36   NoSubscriberReplyParam,
37   ForwardingViolationParam,
38   ForwardingFailedParam,
39   ATI-NotAllowedParam,
40   SubBusyForMT-SMS-Param,
41   MessageWaitListFullParam,
42   AbsentSubscriberSM-Param,
43   AbsentSubscriberDiagnosticSM
44
45 ;
46
47 IMPORTS
48   SS-Status
49 FROM MAP-SS-DataTypes {
50   ccitt identified-organization (4) etsi (0) mobileDomain (0)
51   gsm-Network (1) modules (3) map-SS-DataTypes (14) version3 (3)}
52
53   SignalInfo,
54   BasicServiceCode,
55   NetworkResource
56 FROM MAP-CommonDataTypes {
57   ccitt identified-organization (4) etsi (0) mobileDomain (0)
58   gsm-Network (1) modules (3) map-CommonDataTypes (18) version3 (3)}
59
60   SS-Code
61 FROM MAP-SS-Code {
62   ccitt identified-organization (4) etsi (0) mobileDomain (0)
63   gsm-Network (1) modules (3) map-SS-Code (15) version3 (3)}
64
65   ExtensionContainer
66 FROM MAP-ExtensionDataTypes {
67   ccitt identified-organization (4) etsi (0) mobileDomain (0)
68   gsm-Network (1) modules (3) map-ExtensionDataTypes (21) version3 (3)}
69 ;
70
71 RoamingNotAllowedParam ::= SEQUENCE {
72   roamingNotAllowedCause      RoamingNotAllowedCause,
73   extensionContainer          ExtensionContainer OPTIONAL,
74   ...}
75

```



```

76 RoamingNotAllowedCause ::= ENUMERATED {
77     plmnRoamingNotAllowed (0),
78     operatorDeterminedBarring (3)}
79
80 CallBarredParam ::= CHOICE {
81     callBarringCause          CallBarringCause,
82     -- call BarringCause must not be used in version 3
83     extensibleCallBarredParam ExtensibleCallBarredParam
84     -- extensibleCallBarredParam must not be used in version <3
85 }
86
87 CallBarringCause ::= ENUMERATED {
88     barringServiceActive (0),
89     operatorBarring (1)}
90
91 ExtensibleCallBarredParam ::= SEQUENCE {
92     callBarringCause          CallBarringCause          OPTIONAL,
93     extensionContainer        ExtensionContainer        OPTIONAL,
94     ...}
95
96 CUG-RejectParam ::= SEQUENCE {
97     cug-RejectCause          CUG-RejectCause          OPTIONAL,
98     extensionContainer        ExtensionContainer        OPTIONAL,
99     ...}
100
101 CUG-RejectCause ::= ENUMERATED {
102     incomingCallsBarredWithinCUG (0),
103     subscriberNotMemberOfCUG (1),
104     requestedBasicServiceViolatesCUG-Constraints (5),
105     calledPartySS-InteractionViolation (7)}
106
107 SS-IncompatibilityCause ::= SEQUENCE {
108     ss-Code                  [1] SS-Code                OPTIONAL,
109     basicService              BasicServiceCode           OPTIONAL,
110     ss-Status                 [4] SS-Status              OPTIONAL,
111     ...}
112
113 PW-RegistrationFailureCause ::= ENUMERATED {
114     undetermined (0),
115     invalidFormat (1),
116     newPasswordsMismatch (2)}
117
118
119 SM-EnumeratedDeliveryFailureCause ::= ENUMERATED {
120     memoryCapacityExceeded (0),
121     equipmentProtocolError (1),
122     equipmentNotSM-Equipped (2),
123     unknownServiceCentre (3),
124     sc-Congestion (4),
125     invalidSME-Address (5),
126     subscriberNotSC-Subscriber (6)}
127
128 SM-DeliveryFailureCause ::= SEQUENCE {
129     sm-EnumeratedDeliveryFailureCause
130     SM-EnumeratedDeliveryFailureCause,
131     diagnosticInfo          SignalInfo                OPTIONAL,
132     extensionContainer        ExtensionContainer        OPTIONAL,
133     ...}
134
135 AbsentSubscriberSM-Param ::= SEQUENCE {
136     absentSubscriberDiagnosticSM AbsentSubscriberDiagnosticSM OPTIONAL,
137     extensionContainer          ExtensionContainer        OPTIONAL,
138     ...}
139
140 AbsentSubscriberDiagnosticSM ::= INTEGER (0..255)
141 -- AbsentSubscriberDiagnosticSM values are defined in ETS 300 536 (GSM 03.40)
142
143 SystemFailureParam ::= CHOICE {
144     networkResource          NetworkResource,
145     -- networkResource must not be used in version 3
146     extensibleSystemFailureParam ExtensibleSystemFailureParam
147     -- extensibleSystemFailureParam must not be used in version <3
148 }
149

```

150	ExtensibleSystemFailureParam ::= SEQUENCE {		
151	networkResource	NetworkResource	OPTIONAL,
152	extensionContainer	ExtensionContainer	OPTIONAL,
153	...		
154			
155	DataMissingParam ::= SEQUENCE {		
156	extensionContainer	ExtensionContainer	OPTIONAL,
157	...		
158			
159	UnexpectedDataParam ::= SEQUENCE {		
160	extensionContainer	ExtensionContainer	OPTIONAL,
161	...		
162			
163	FacilityNotSupParam ::= SEQUENCE {		
164	extensionContainer	ExtensionContainer	OPTIONAL,
165	...		
166			
167	OR-NotAllowedParam ::= SEQUENCE {		
168	extensionContainer	ExtensionContainer	OPTIONAL,
169	...		
170			
171	UnknownSubscriberParam ::= SEQUENCE {		
172	extensionContainer	ExtensionContainer	OPTIONAL,
173	...		
174			
175	NumberChangedParam ::= SEQUENCE {		
176	extensionContainer	ExtensionContainer	OPTIONAL,
177	...		
178			
179	UnidentifiedSubParam ::= SEQUENCE {		
180	extensionContainer	ExtensionContainer	OPTIONAL,
181	...		
182			
183	IllegalSubscriberParam ::= SEQUENCE {		
184	extensionContainer	ExtensionContainer	OPTIONAL,
185	...		
186			
187	IllegalEquipmentParam ::= SEQUENCE {		
188	extensionContainer	ExtensionContainer	OPTIONAL,
189	...		
190			
191	BearerServNotProvParam ::= SEQUENCE {		
192	extensionContainer	ExtensionContainer	OPTIONAL,
193	...		
194			
195	TeleservNotProvParam ::= SEQUENCE {		
196	extensionContainer	ExtensionContainer	OPTIONAL,
197	...		
198			
199	TracingBufferFullParam ::= SEQUENCE {		
200	extensionContainer	ExtensionContainer	OPTIONAL,
201	...		
202			
203	NoRoamingNbParam ::= SEQUENCE {		
204	extensionContainer	ExtensionContainer	OPTIONAL,
205	...		
206			
207	AbsentSubscriberParam ::= SEQUENCE {		
208	extensionContainer	ExtensionContainer	OPTIONAL,
209	...		
210			
211	BusySubscriberParam ::= SEQUENCE {		
212	extensionContainer	ExtensionContainer	OPTIONAL,
213	...		
214			
215	NoSubscriberReplyParam ::= SEQUENCE {		
216	extensionContainer	ExtensionContainer	OPTIONAL,
217	...		
218			
219	ForwardingViolationParam ::= SEQUENCE {		
220	extensionContainer	ExtensionContainer	OPTIONAL,
221	...		
222			
223	ForwardingFailedParam ::= SEQUENCE {		
224	extensionContainer	ExtensionContainer	OPTIONAL,
225	...		
226			

```
227 ATI-NotAllowedParam ::= SEQUENCE {  
228     extensionContainer      ExtensionContainer      OPTIONAL,  
229     ...}  
230  
231 SubBusyForMT-SMS-Param ::= SEQUENCE {  
232     extensionContainer      ExtensionContainer      OPTIONAL,  
233     ...}  
234  
235 MessageWaitListFullParam ::= SEQUENCE {  
236     extensionContainer      ExtensionContainer      OPTIONAL,  
237     ...}  
238  
239 END
```

14.7.8 Common data types

```

1 MAP-CommonDataTypes {
2     ccitt identified-organization (4) etsi (0) mobileDomain (0)
3     gsm-Network (1) modules (3) map-CommonDataTypes (18) version3 (3)}
4
5 DEFINITIONS
6
7 IMPLICIT TAGS
8
9 ::=
10
11 BEGIN
12
13 EXPORTS
14
15     -- general data types and values
16     AddressString,
17     ISDN-AddressString,
18     ISDN-SubaddressString,
19     ExternalSignalInfo,
20     SignalInfo,
21     maxSignalInfoLength,
22
23     -- data types for numbering and identification
24     IMSI,
25     TMSI,
26     SubscriberId,
27     IMEI,
28     HLR-List,
29     LMSI,
30     GlobalCellId,
31     NetworkResource,
32
33     -- data types for CAMEL
34     CellIdOrLAI,
35
36     -- data types for subscriber management
37     BasicServiceCode,
38     Ext-BasicServiceCode
39 ;
40
41 IMPORTS
42     TeleserviceCode,
43     Ext-TeleserviceCode
44 FROM MAP-TS-Code {
45     ccitt identified-organization (4) etsi (0) mobileDomain (0)
46     gsm-Network (1) modules (3) map-TS-Code (19) version3 (3)}
47
48     BearerServiceCode,
49     Ext-BearerServiceCode
50 FROM MAP-BS-Code {
51     ccitt identified-organization (4) etsi (0) mobileDomain (0)
52     gsm-Network (1) modules (3) map-BS-Code (20) version3 (3)}
53
54     ExtensionContainer
55 FROM MAP-ExtensionDataTypes {
56     ccitt identified-organization (4) etsi (0) mobileDomain (0)
57     gsm-Network (1) modules (3) map-ExtensionDataTypes (21) version3 (3)}
58 ;
59
60
61 -- general data types
62
63 TBCD-STRING ::= OCTET STRING
64     -- This type (Telephony Binary Coded Decimal String) is used to
65     -- represent several digits from 0 through 9, *, #, a, b, c, two
66     -- digits per octet, each digit encoded 0000 to 1001 (0 to 9),
67     -- 1010 (*), 1011 (#), 1100 (a), 1101 (b) or 1110 (c); 1111 used
68     -- as filler when there is an odd number of digits.
69
70     -- bits 8765 of octet n encoding digit 2n
71     -- bits 4321 of octet n encoding digit 2(n-1) +1
72

```

```

73 AddressString ::= OCTET STRING (SIZE (1..maxAddressLength))
74 -- This type is used to represent a number for addressing
75 -- purposes. It is composed of
76 -- a) one octet for nature of address, and numbering plan
77 -- indicator.
78 -- b) digits of an address encoded as TBCD-String.
79
80 -- a) The first octet includes a one bit extension indicator, a
81 -- 3 bits nature of address indicator and a 4 bits numbering
82 -- plan indicator, encoded as follows:
83
84 -- bit 8: 1 (no extension)
85
86 -- bits 765: nature of address indicator
87 -- 000 unknown
88 -- 001 international number
89 -- 010 national significant number
90 -- 011 network specific number
91 -- 100 subscriber number
92 -- 101 reserved
93 -- 110 abbreviated number
94 -- 111 reserved for extension
95
96 -- bits 4321: numbering plan indicator
97 -- 0000 unknown
98 -- 0001 ISDN/Telephony Numbering Plan (Rec CCITT E.164)
99 -- 0010 spare
100 -- 0011 data numbering plan (CCITT Rec X.121)
101 -- 0100 telex numbering plan (CCITT Rec F.69)
102 -- 0101 spare
103 -- 0110 land mobile numbering plan (CCITT Rec E.212)
104 -- 0111 spare
105 -- 1000 national numbering plan
106 -- 1001 private numbering plan
107 -- 1111 reserved for extension
108
109 -- all other values are reserved.
110
111 -- b) The following octets representing digits of an address
112 -- encoded as a TBCD-STRING.
113
114 maxAddressLength INTEGER ::= 20
115
116 ISDN-AddressString ::=
117     AddressString (SIZE (1..maxISDN-AddressLength))
118     -- This type is used to represent ISDN numbers.
119
120 maxISDN-AddressLength INTEGER ::= 9
121

```

```

122 ISDN-SubaddressString ::=
123     OCTET STRING (SIZE (1..maxISDN-SubaddressLength))
124     -- This type is used to represent ISDN subaddresses.
125     -- It is composed of
126     -- a) one octet for type of subaddress and odd/even indicator.
127     -- b) 20 octets for subaddress information.
128
129     -- a) The first octet includes a one bit extension indicator, a
130     --     3 bits type of subaddress and a one bit odd/even indicator,
131     --     encoded as follows:
132
133     -- bit 8: 1 (no extension)
134
135     -- bits 765: type of subaddress
136     --     000 NSAP (X.213/ISO 8348 AD2)
137     --     010 User Specified
138     --     All other values are reserved
139
140     -- bit 4: odd/even indicator
141     --     0 even number of address signals
142     --     1 odd number of address signals
143     -- The odd/even indicator is used when the type of subaddress
144     -- is "user specified" and the coding is BCD.
145
146     -- bits 321: 000 (unused)
147
148     -- b) Subaddress information.
149     -- The NSAP X.213/ISO8348AD2 address shall be formatted as specified
150     -- by octet 4 which contains the Authority and Format Identifier
151     -- (AFI). The encoding is made according to the "preferred binary
152     -- encoding" as defined in X.213/ISO834AD2. For the definition
153     -- of this type of subaddress, see CCITT Rec I.334.
154
155     -- For User-specific subaddress, this field is encoded according
156     -- to the user specification, subject to a maximum length of 20
157     -- octets. When interworking with X.25 networks BCD coding should
158     -- be applied.

```

```

159
160 maxISDN-SubaddressLength INTEGER ::= 21

```

```

161
162 ExternalSignalInfo ::= SEQUENCE {
163     protocolId          ProtocolId,
164     signalInfo          SignalInfo,
165     -- Information about the internal structure is given in
166     -- subclause 5.6.9.
167     extensionContainer  ExtensionContainer OPTIONAL,
168     -- extensionContainer must not be used in version 2
169     ...}

```

```

170
171 SignalInfo ::= OCTET STRING (SIZE (1..maxSignalInfoLength))

```

```

172
173 maxSignalInfoLength INTEGER ::= 200
174 -- This NamedValue represents the theoretical maximum number of
175 -- octets which are available to carry a single data type,
176 -- without requiring segmentation to cope with the network layer
177 -- service. However, the actual maximum size available for a data
178 -- type may be lower, especially when other information elements
179 -- have to be included in the same component.

```

```

180
181 ProtocolId ::= ENUMERATED {
182     gsm-0408 (1),
183     gsm-0806 (2),
184     gsm-BSSMAP (3),
185     -- Value 3 is reserved and must not be used
186     ets-300102-1 (4)}

```

```

187
188 -- data types for numbering and identification

```

```

189
190 IMSI ::= TB CD-STRING (SIZE (3..8))
191 -- digits of MCC, MNC, MSIN are concatenated in this order.

```

```

192
193 TMSI ::= OCTET STRING (SIZE (1..4))

```

```

194
195 SubscriberId ::= CHOICE {
196     imsi                [0] IMSI,
197     tmsi                [1] TMSI}

```

```

200 IMEI ::= TBCD-STRING (SIZE (8))
201     -- Refers to International Mobile Station Equipment Identity
202     -- and Software Version Number (SVN) defined in TS GSM 03.03.
203     -- If the SVN is not present the last octet shall contain the
204     -- digit 0 and a filler.
205     -- If present the SVN shall be included in the last octet.
206
207 HLR-Id ::= IMSI
208     -- leading digits of IMSI, i.e. (MCC, MNC, leading digits of
209     -- MSIN) forming HLR Id defined in TS GSM 03.03.
210
211 HLR-List ::= SEQUENCE SIZE (1..maxNumOfHLR-Id) OF
212             HLR-Id
213
214 maxNumOfHLR-Id INTEGER ::= 50
215
216 LMSI ::= OCTET STRING (SIZE (4))
217
218 GlobalCellId ::= OCTET STRING (SIZE (5..7))
219     -- Refers to Cell Global Identification defined in TS GSM 03.03.
220     -- Octets are coded according to TS GSM 04.08.
221     -- The internal structure is defined as follows:
222     -- Mobile Country Code:      3 digits according to CCITT Rec E.212
223     --                           1 digit filler (1111)
224     -- Mobile Network Code:      2 digits according to CCITT Rec E.212
225     -- Location Area Code:      2 octets according to TS GSM 04.08
226     -- Cell Identity:           2 octets (CI) according to TS GSM 04.08
227
228 NetworkResource ::= ENUMERATED {
229     plmn (0),
230     hlr (1),
231     vlr (2),
232     pvlr (3),
233     controllingMSC (4),
234     vmsc (5),
235     eir (6),
236     rss (7)}
237
238 -- data types for CAMEL
239
240 CellIdOrLAI ::= CHOICE {
241     cellIdFixedLength [0] CellIdFixedLength,
242     laiFixedLength [1] LAIFixedLength}
243
244 CellIdFixedLength ::= OCTET STRING (SIZE (7))
245     -- Refers to Cell Global Identification defined in TS GSM 03.03.
246     -- Octets are coded according to TS GSM 04.08.
247     -- The internal structure is defined as follows:
248     -- Mobile Country Code:      3 digits according to CCITT Rec E.212
249     --                           1 digit filler (1111)
250     -- Mobile Network Code:      2 digits according to CCITT Rec E.212
251     -- Location Area Code:      2 octets according to TS GSM 04.08
252     -- Cell Identity:           2 octets (CI) according to TS GSM 04.08
253
254 LAIFixedLength ::= OCTET STRING (SIZE (5))
255     -- Refers to Location Area Identification defined in TS GSM 03.03.
256     -- Octets are coded according to TS GSM 04.08.
257     -- The internal structure is defined as follows:
258     -- Mobile Country Code:      3 digits according to CCITT Rec E.212
259     --                           1 digit filler (1111)
260     -- Mobile Network Code:      2 digits according to CCITT Rec E.212
261     -- Location Area Code:      2 octets according to TS GSM 04.08
262
263 -- data types for subscriber management
264
265 BasicServiceCode ::= CHOICE {
266     bearerService [2] BearerServiceCode,
267     teleservice [3] TeleserviceCode}
268
269 Ext-BasicServiceCode ::= CHOICE {
270     ext-BearerService [2] Ext-BearerServiceCode,
271     ext-Teleservice [3] Ext-TeleserviceCode}
272
273 END

```

14.7.9 Teleservice Codes

```

1 MAP-TS-Code {
2   ccitt-identified-organization (4) etsi (0) mobileDomain (0)
3   gsm-Network (1) modules (3) map-TS-Code (19) version3 (3)}
4
5 DEFINITIONS
6
7 ::=
8
9 BEGIN
10
11 TeleserviceCode ::= OCTET STRING (SIZE (1))
12   -- This type is used to represent the code identifying a single
13   -- teleservice, a group of teleservices, or all teleservices. The
14   -- services are defined in TS GSM 02.03.
15   -- The internal structure is defined as follows:
16
17   -- bits 87654321: group (bits 8765) and specific service
18   -- (bits 4321)
19
20 Ext-TeleserviceCode ::= OCTET STRING (SIZE (1..5))
21   -- This type is used to represent the code identifying a single
22   -- teleservice, a group of teleservices, or all teleservices. The
23   -- services are defined in TS GSM 02.03.
24   -- The internal structure is defined as follows:
25
26   -- OCTET 1:
27   -- bits 87654321: group (bits 8765) and specific service
28   -- (bits 4321)
29
30   -- OCTETS 2-5: reserved for future use. If received the
31   -- Ext-TeleserviceCode shall be
32   -- treated according to the exception handling defined for the
33   -- operation that uses this type.
34
35   -- Ext-TeleserviceCode includes all values defined for TeleserviceCode.
36
37
38 allTeleservices TeleserviceCode ::= '00000000'B
39
40 allSpeechTransmissionServices TeleserviceCode ::= '00010000'B
41 telephony TeleserviceCode ::= '00010001'B
42 emergencyCalls TeleserviceCode ::= '00010010'B
43
44 allShortMessageServices TeleserviceCode ::= '00100000'B
45 shortMessageMT-PP TeleserviceCode ::= '00100001'B
46 shortMessageMO-PP TeleserviceCode ::= '00100010'B
47
48 allFacsimileTransmissionServices TeleserviceCode ::= '01100000'B
49 facsimileGroup3AndAlterSpeech TeleserviceCode ::= '01100001'B
50 automaticFacsimileGroup3 TeleserviceCode ::= '01100010'B
51 facsimileGroup4 TeleserviceCode ::= '01100011'B
52
53 -- The following non-hierarchical Compound Teleservice Groups
54 -- are defined in TS GSM 02.30:
55 allDataTeleservices TeleserviceCode ::= '01110000'B
56   -- covers Teleservice Groups 'allFacsimileTransmissionServices'
57   -- and 'allShortMessageServices'
58 allTeleservices-ExeptsSMS TeleserviceCode ::= '10000000'B
59   -- covers Teleservice Groups 'allSpeechTransmissionServices' and
60   -- 'allFacsimileTransmissionServices'
61 --
62 -- Compound Teleservice Group Codes are only used in call
63 -- independent supplementary service operations, i.e. they
64 -- are not used in InsertSubscriberData or in
65 -- DeleteSubscriberData messages.
66
67 allVoiceGroupCallServices TeleserviceCode ::= '10010000'B
68
69 voiceGroupCall TeleserviceCode ::= '10010001'B
70 voiceBroadcastCall TeleserviceCode ::= '10010010'B
71

```


72	allPLMN-specificTS	TeleserviceCode ::= '11010000'B
73	plmn-specificTS-1	TeleserviceCode ::= '11010001'B
74	plmn-specificTS-2	TeleserviceCode ::= '11010010'B
75	plmn-specificTS-3	TeleserviceCode ::= '11010011'B
76	plmn-specificTS-4	TeleserviceCode ::= '11010100'B
77	plmn-specificTS-5	TeleserviceCode ::= '11010101'B
78	plmn-specificTS-6	TeleserviceCode ::= '11010110'B
79	plmn-specificTS-7	TeleserviceCode ::= '11010111'B
80	plmn-specificTS-8	TeleserviceCode ::= '11011000'B
81	plmn-specificTS-9	TeleserviceCode ::= '11011001'B
82	plmn-specificTS-A	TeleserviceCode ::= '11011010'B
83	plmn-specificTS-B	TeleserviceCode ::= '11011011'B
84	plmn-specificTS-C	TeleserviceCode ::= '11011100'B
85	plmn-specificTS-D	TeleserviceCode ::= '11011101'B
86	plmn-specificTS-E	TeleserviceCode ::= '11011110'B
87	plmn-specificTS-F	TeleserviceCode ::= '11011111'B

88
89 END

14.7.10 Bearer Service Codes

```

1 MAP-BS-Code {
2   ccitt-identified-organization (4) etsi (0) mobileDomain (0)
3   gsm-Network (1) modules (3) map-BS-Code (20) version3 (3)}
4
5 DEFINITIONS
6
7 ::=
8
9 BEGIN
10
11 BearerServiceCode ::= OCTET STRING (SIZE (1))
12   -- This type is used to represent the code identifying a single
13   -- bearer service, a group of bearer services, or all bearer
14   -- services. The services are defined in TS GSM 02.02.
15   -- The internal structure is defined as follows:
16   --
17   -- plmn-specific bearer services:
18   -- bits 87654321: defined by the HPLMN operator
19
20   -- rest of bearer services:
21   -- bit 8: 0 (unused)
22   -- bits 7654321: group (bits 7654), and rate, if applicable
23   -- (bits 321)
24
25 Ext-BearerServiceCode ::= OCTET STRING (SIZE (1..5))
26   -- This type is used to represent the code identifying a single
27   -- bearer service, a group of bearer services, or all bearer
28   -- services. The services are defined in TS GSM 02.02.
29   -- The internal structure is defined as follows:
30   --
31   -- OCTET 1:
32   -- plmn-specific bearer services:
33   -- bits 87654321: defined by the HPLMN operator
34   --
35   -- rest of bearer services:
36   -- bit 8: 0 (unused)
37   -- bits 7654321: group (bits 7654), and rate, if applicable
38   -- (bits 321)
39
40   -- OCTETS 2-5: reserved for future use. If received the
41   -- Ext-TeleserviceCode shall be
42   -- treated according to the exception handling defined for the
43   -- operation that uses this type.
44
45   -- Ext-BearerServiceCode includes all values defined for BearerServiceCode.
46
47
48
49 allBearerServices BearerServiceCode ::= '00000000'B
50
51 allDataCDA-Services BearerServiceCode ::= '00010000'B
52 dataCDA-300bps BearerServiceCode ::= '00010001'B
53 dataCDA-1200bps BearerServiceCode ::= '00010010'B
54 dataCDA-1200-75bps BearerServiceCode ::= '00010011'B
55 dataCDA-2400bps BearerServiceCode ::= '00010100'B
56 dataCDA-4800bps BearerServiceCode ::= '00010101'B
57 dataCDA-9600bps BearerServiceCode ::= '00010110'B
58 general-dataCDA BearerServiceCode ::= '00010111'B
59
60 allDataCDS-Services BearerServiceCode ::= '00011000'B
61 dataCDS-1200bps BearerServiceCode ::= '00011010'B
62 dataCDS-2400bps BearerServiceCode ::= '00011100'B
63 dataCDS-4800bps BearerServiceCode ::= '00011101'B
64 dataCDS-9600bps BearerServiceCode ::= '00011110'B
65 general-dataCDS BearerServiceCode ::= '00011111'B
66
67 allPadAccessCA-Services BearerServiceCode ::= '00100000'B
68 padAccessCA-300bps BearerServiceCode ::= '00100001'B
69 padAccessCA-1200bps BearerServiceCode ::= '00100010'B
70 padAccessCA-1200-75bps BearerServiceCode ::= '00100011'B
71 padAccessCA-2400bps BearerServiceCode ::= '00100100'B
72 padAccessCA-4800bps BearerServiceCode ::= '00100101'B
73 padAccessCA-9600bps BearerServiceCode ::= '00100110'B
74 general-padAccessCA BearerServiceCode ::= '00100111'B
75

```

```

76 allDataPDS-Services      BearerServiceCode ::= '00101000'B
77 dataPDS-2400bps          BearerServiceCode ::= '00101100'B
78 dataPDS-4800bps          BearerServiceCode ::= '00101101'B
79 dataPDS-9600bps          BearerServiceCode ::= '00101110'B
80 general-dataPDS          BearerServiceCode ::= '00101111'B
81
82 allAlternateSpeech-DataCDA BearerServiceCode ::= '00110000'B
83
84 allAlternateSpeech-DataCDS BearerServiceCode ::= '00111000'B
85
86 allSpeechFollowedByDataCDA BearerServiceCode ::= '01000000'B
87
88 allSpeechFollowedByDataCDS BearerServiceCode ::= '01001000'B
89
90 -- The following non-hierarchical Compound Bearer Service
91 -- Groups are defined in TS GSM 02.30:
92 allDataCircuitAsynchronous BearerServiceCode ::= '01010000'B
93 -- covers "allDataCDA-Services", "allAlternateSpeech-DataCDA" and
94 -- "allSpeechFollowedByDataCDA"
95 allAsynchronousServices    BearerServiceCode ::= '01100000'B
96 -- covers "allDataCDA-Services", "allAlternateSpeech-DataCDA",
97 -- "allSpeechFollowedByDataCDA" and "allPadAccessCDA-Services"
98 allDataCircuitSynchronous  BearerServiceCode ::= '01011000'B
99 -- covers "allDataCDS-Services", "allAlternateSpeech-DataCDS" and
100 -- "allSpeechFollowedByDataCDS"
101 allSynchronousServices     BearerServiceCode ::= '01101000'B
102 -- covers "allDataCDS-Services", "allAlternateSpeech-DataCDS",
103 -- "allSpeechFollowedByDataCDS" and "allDataPDS-Services"
104 --
105 -- Compound Bearer Service Group Codes are only used in call
106 -- independent supplementary service operations, i.e. they
107 -- are not used in InsertSubscriberData or in
108 -- DeleteSubscriberData messages.
109
110 allPLMN-specificBS        BearerServiceCode ::= '11010000'B
111 plmn-specificBS-1         BearerServiceCode ::= '11010001'B
112 plmn-specificBS-2         BearerServiceCode ::= '11010010'B
113 plmn-specificBS-3         BearerServiceCode ::= '11010011'B
114 plmn-specificBS-4         BearerServiceCode ::= '11010100'B
115 plmn-specificBS-5         BearerServiceCode ::= '11010101'B
116 plmn-specificBS-6         BearerServiceCode ::= '11010110'B
117 plmn-specificBS-7         BearerServiceCode ::= '11010111'B
118 plmn-specificBS-8         BearerServiceCode ::= '11011000'B
119 plmn-specificBS-9         BearerServiceCode ::= '11011001'B
120 plmn-specificBS-A         BearerServiceCode ::= '11011010'B
121 plmn-specificBS-B         BearerServiceCode ::= '11011011'B
122 plmn-specificBS-C         BearerServiceCode ::= '11011100'B
123 plmn-specificBS-D         BearerServiceCode ::= '11011101'B
124 plmn-specificBS-E         BearerServiceCode ::= '11011110'B
125 plmn-specificBS-F         BearerServiceCode ::= '11011111'B
126
127 END

```

14.7.11 Extension data types

```

1 MAP-ExtensionDataTypes {
2   ccitt identified-organization (4) etsi (0) mobileDomain (0)
3   gsm-Network (1) modules (3) map-ExtensionDataTypes (21) version3 (3)}
4
5 DEFINITIONS
6
7 IMPLICIT TAGS
8
9 ::=
10
11 BEGIN
12
13 EXPORTS
14
15     PrivateExtension,
16     ExtensionContainer;
17
18
19 -- IOC for private MAP extensions
20
21
22 MAP-EXTENSION ::= CLASS {
23     &ExtensionType                                OPTIONAL,
24     &extensionId                                OBJECT IDENTIFIER }
25     -- The length of the Object Identifier shall not exceed 16 octets and the
26     -- number of components of the Object Identifier shall not exceed 16
27
28
29
30 -- data types
31
32 ExtensionContainer ::= SEQUENCE {
33     privateExtensionList    [0]PrivateExtensionList    OPTIONAL,
34     pcs-Extensions          [1]PCS-Extensions          OPTIONAL,
35     ...}
36
37 PrivateExtensionList ::= SEQUENCE SIZE (1..maxNumOfPrivateExtensions) OF
38     PrivateExtension
39
40 PrivateExtension ::= SEQUENCE {
41     extId                                MAP-EXTENSION.&extensionId
42                                     ({ExtensionSet}),
43     extType                                MAP-EXTENSION.&ExtensionType
44                                     ({ExtensionSet}{@extId})    OPTIONAL}
45
46 maxNumOfPrivateExtensions    INTEGER ::= 10
47
48 ExtensionSet                                MAP-EXTENSION ::=
49     {...
50     -- ExtensionSet is the set of all defined private extensions
51     }
52
53     -- Unsupported private extensions shall be discarded if received.
54
55
56 PCS-Extensions ::= SEQUENCE {
57     ...}
58
59 END
60
61
62
63

```

15 General on MAP user procedures

15.1 Introduction

Clauses 15 to 21 describe the use of MAP services for GSM signalling procedures. GSM signalling procedures may involve one or several interfaces running one or several application protocols. This ETS addresses only the signalling procedures which require at least the use of one MAP service.

When a signalling procedure takes place in the network, an application process invocation is created in each system component involved. Part of the application process invocation acts as a MAP user and handles one or several MAP dialogues. For each dialogue it employs an instance of the MAP service provider. It may also use other communication services to exchange information on other interfaces, but detailed description of these aspects is outside the scope of this ETS.

15.2 Common aspects of user procedure descriptions

15.2.1 General conventions

For each signalling procedure this ETS provides a brief textual overview accompanied by a flow diagram which represent the functional interactions between system components. Functional interactions are labelled using the MAP service name when the interaction results from a service request or by this service name followed by the symbol "ack" when this interaction results from a service response.

For each of the system components involved, this ETS also provides a detailed textual description of the application process behaviour as well as an SDL diagram. SDL diagrams describe the sequence of events, as seen by the MAP-User, which occurs at MAP service provider boundaries as well as external events which occur at other interfaces and which impact on the previous sequence.

External events do not necessarily correspond to the messages of other protocols used in the system component. The MAP-user procedures are described as if a set of interworking functions (IWF) between the MAP-user and the other protocol entities was implemented (see figure 15.2/1). Such interworking functions are assumed to perform either an identity mapping or some processing or translation as required to eliminate information irrelevant to the MAP-user.

The mapping of service primitives on to protocol elements is described in clauses 11 to 14.

GSM signalling procedures are built from one or more sub-procedures (e.g. authentication, ciphering,). Sub-procedures from which signalling procedures are built are represented using SDL MACRO descriptions.

In case of any discrepancy between the textual descriptions and the SDL descriptions, the latter take precedence.

15.2.2 Naming conventions

Events related to MAP are represented by MAP service primitives. The signal names used in the SDL diagrams are derived from the service primitive names defined in clauses 5 to 10, with some lexical transformations for readability and parsability purposes (blanks between words are replaced by underscores, the first letter of each word is capitalized).

Events received and sent on other interfaces are named by appending the message or signal name to a symbol representing the interface type, with some lexical transformations for readability and parsability purposes (blanks between words are replaced by underscores, the first letter of each word is capitalized).

The following symbols are used to represent the interface types:

"I": For interfaces to the fixed network. "I" stands for ISUP interface.
 "A": For interfaces to BSS (i.e. A-interfaces);
 "OM": For network management interfaces (communication with OMC, MML interface, ...);
 "SC": For interfaces to a Service Centre;
 "HO_CA": For internal interfaces to the Handover Control Application.
 "US": For a local USSD application.

These naming conventions can be summarized by the following BNF description:

```

<Event_Name>          ::= <MAP_Primitive> | <External_Event>

<MAP_Primitive>       ::= <MAP_Open> | <MAP_Close> | <MAP_U_Abort> | <MAP_P_Abort> |
                          <MAP_Specific> | <MAP_Notice>

<MAP_Open>            ::= MAP_Open_Req | MAP_Open_Ind | MAP_Open_Rsp | MAP_Open_Cnf

<MAP_Close>           ::= MAP_Close_Req | MAP_Close_Ind

<MAP_U_Abort>         ::= MAP_U_Abort_Req | MAP_U_Abort_Ind

<MAP_P_Abort>         ::= MAP_P_Abort_Ind

<MAP_Notice>          ::= MAP_Notice_Ind

<MAP_Specific>        ::= <MAP_Req> | <MAP_Ind> | <MAP_Rsp> | <MAP_Cnf>

<MAP_Req>             ::= MAP_<Service_Name>_Req

<MAP_Ind>             ::= MAP_<Service_Name>_Ind

<MAP_Rsp>            ::= MAP_<Service_Name>_Rsp

<MAP_Cnf>            ::= MAP_<Service_Name>_Cnf

<External_Event>      ::= <Interface_Type>_<External_Signal>

<Interface_Type>      ::= I | A | OM | SC | HO AC | US

<External_Signal>     ::= <Lexical_Unit>

<Service_Name>        ::= <Lexical_Unit>

<Lexical_Unit>        ::= <Lexical_Component> | <Lexical_Unit>_<Lexical_Component>

<Lexical_Component>   ::= <Upper_Case_Letter><Letter_Or_Digit_List>

<Letter_Or_Digit_List> ::= <Letter_Or_Digit> | <Letter_Or_Digit_List><Letter_Or_Digit>

<Letter_Or_Digit>     ::= <Letter> | <Digit>

<Letter>              ::= <Lower_Case_Letter> | <Upper_Case_Letter>

<Upper_Case_Letter>   ::= A|B|C|D|E|F|G|H|I|J|K|L|M|N|O|P|Q|R|S|T|U|V|W|X|Y|Z

<Lower_Case_Letter>   ::= a|b|c|d|e|f|g|h|i|j|k|l|m|n|o|p|q|r|s|t|u|v|w|x|y|z

<Digit>               ::= 1|2|3|4|5|6|7|8|9|0
  
```

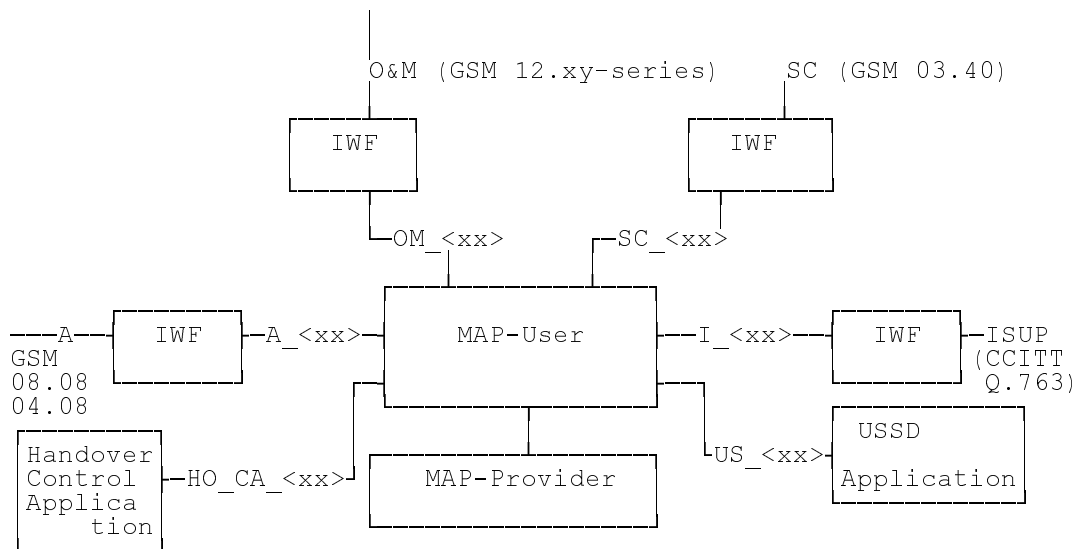


Figure 15.2/1: Interfaces applicable to the MAP-User

15.2.3 Convention on primitives parameters

15.2.3.1 Open service

When the originating and destination reference parameters shall be included in the MAP-OPEN request primitive, their value are indicated as a comment to the signal which represents this primitive.

15.2.3.2 Close service

When a pre-arranged released is requested, a comment is attached to the signal which represents the MAP-CLOSE request primitive. In the absence of comment, a normal release is assumed.

15.2.4 Version handling at dialogue establishment

Unless explicitly indicated in subsequent subclauses, the following principles regarding version handling procedures at dialogue establishment are applied by the MAP-user:

15.2.4.1 Behaviour at the initiating side

When a MAP user signalling procedure has to be executed, the MAP-user issues a MAP-OPEN request primitive with an appropriate application-context-name. If several names are supported (i.e. several versions) a suitable one is selected using the procedures described in clause 3.

If version 2 is selected and a MAP-CLOSE Confirm primitive in response to the MAP-OPEN request is received with a result parameter set to "refused" and a diagnostic parameter indicating "application-context-not-supported" or "potential incompatibility problem", the MAP-User issues a new MAP-OPEN request primitive with the equivalent version one context. This is informally represented in the SDL diagrams by a task symbol indicating "Perform Vr procedure".

If version 3 is selected and a MAP-CLOSE Confirm primitive in response to the MAP-OPEN request is received with a result parameter set to "refused" and a diagnostic parameter indicating "application-context-not-supported" or "potential incompatibility problem", the MAP-User issues a new MAP-OPEN request primitive with the equivalent version one or version two context. This is informally represented in the SDL diagrams by task symbols indicating "Perform Vr procedure".

15.2.4.2 Behaviour at the responding side

On receipt of a MAP-OPEN indication primitive, the MAP-User analyses the application-context-name.

If it refers to a version one context, the associated V1 procedure is executed; if it refers to a version two context, the associated V2 procedure is executed, otherwise the associated V3 procedure is executed.

15.2.5 Abort Handling

Unless explicitly indicated in subsequent subclauses, the following principles are applied by the MAP-user regarding abort handling procedures:

On receipt of a MAP-P-ABORT indication or MAP-U-ABORT Indication primitive from any MAP-provider invocation, the MAP-User issues a MAP-U-ABORT Request primitive to each MAP-provider invocation associated with the same user procedure.

If applicable a decision is made to decide if the affected user procedure has to be retried or not.

15.2.6 SDL conventions

The MAP SDLs make use of a number of SDL concepts and conventions, where not all of them may be widely known. Therefore, this subclause outlines the use of a few concepts and conventions to improve understanding of the MAP SDLs.

The MAP User SDLs make use of SDL Processes, Procedures and Macros. Processes are independent from each other even if one process starts another one: The actions of both of them have no ordering in time. SDL Procedures and Macros are just used to ease writing of the specification: They contain parts of a behaviour used in several places, and the corresponding Procedure/Macro definition has to be expanded at the position of the Procedure/Macro call.

All Processes are started at system initialization and live forever, unless process creation/termination is indicated explicitly (i.e. a process is created by some other process).

The direction of Input/Output Signals in the SDL graphs is used to indicate the entity to which/from which communication is directed. If a process A communicates in parallel with processes B and C, all Inputs/Outputs to/from B are directed to one side, whereas communication with C is directed to the other side. However, there has been no formal convention used that communication to a certain entity (e.g. a HLR) will always be directed to a certain side (e.g. right).

In each state all those Input Signals are listed, which result in an action and/or state change. If an Input Signal is not listed in a state, receipt of this input should lead to an implicit consumption without any action or state change (according to the SDL rules). This implicit consumption is mainly used for receipt of the MAP DELIMITER indication and for receipt of a MAP CLOSE indication, except for a premature MAP CLOSE.

15.3 Interaction between MAP Provider and MAP Users

Each MAP User is defined by at least one SDL process. On the dialogue initiating side the MAP User will create a new instance of a MAP Provider implicit by issuing a MAP-OPEN request. This instance corresponds to a TC Dialogue and lives as long as the dialogue exists (see also subclause 11.3). There is a fix relation between MAP User and this Provider instance, i.e. all MAP service primitives from the MAP User for this dialogue are sent to this instance and all TC components received by this MAP Provider are mapped onto service primitives sent to this MAP User.

On the receiving side a MAP Provider instance is created implicit by receipt of a TC BEGIN indication. The corresponding MAP User is determined by the Application Context name included in this primitive, i.e. each Application Context is associated with one and only one MAP User. An instance of this User will be created implicit by receiving a MAP-OPEN indication. Note that in some cases there exist several SDL Processes for one MAP User (Application Context), e.g. the processes Register_SS_HLR, Erase_SS_HLR, Activate_SS_HLR, Deactivate_SS_HLR, Interrogate_SS_HLR, and Register_Password for the AC Network_Functional_SS_Handling. In these cases, a coordinator process is introduced acting as a MAP User, which in turn starts a sub-process depending on the first MAP service primitive received.

16 Mobility procedures

16.1 Location management Procedures

This subclause comprises a number of processes to handle the mobile nature of the subscriber. The processes will be addressed by SCCP Sub-System Number (MSC, VLR or HLR) and the Application Context. The following processes are defined in this subclause:

Process Update Location Area:

Initiator: Update_Location_Area_MSC, subclause 16.1.1.2;
Responder: Update_Location_Area_VLR, subclause 16.1.1.3;

Process Update Location:

Initiator: Update_Location_Area_VLR, subclause 16.1.1.3, or
Update_Location_VLR, subclause 16.1.1.6;
Responder: Update_Location_HLR, subclause 16.1.1.4;

Process Send Identification:

Initiator: Update_Location_Area_VLR, subclause 16.1.1.3;
Responder: Send_Identification_VLR, subclause 16.1.1.5;

Process Subscriber Present HLR:

Initiator: Subscriber_Present_HLR, subclause 16.1.1.7;
Responder: Short_Message_Alert_IWMSC, subclause 20.4.3;

Process Cancel Location:

Initiator: Cancel_Location_HLR, subclause 16.1.2.2;
Responder: Cancel_Location_VLR, subclause 16.1.2.3;

Process Detach IMSI:

Initiator: Detach_IMSI_MSC, subclause 16.3.2;
Responder: Detach_IMSI_VLR, subclause 16.3.3.

As both the Update Location Area and the Detach IMSI processes use the same application context name, the MAP Provider cannot distinguish between them. Therefore, a Location Management Coordinator Process will act as one user for this application context. This process (one in MSC, one in VLR) will create the Update Location Area or the Detach IMSI process, depending on the first service primitive received in the respective dialogue.

Additionally, a Location Management Coordinator process in the HLR coordinates the two application processes "Update Location HLR" (subclause 16.1.1.4) and "RESTORE_DATA_HLR" (subclause 16.3.3) that are addressed by the same application context.

Location Management Coordinator MSC

On receipt of a request for location updating from the A-interface, the Location Management Coordinator in the MSC will:

- create the process Update_Location_Area_MSC in case the updating type indicated in the A-interface primitive indicates normal updating, periodic updating or IMSI Attach;
- create the process Detach_IMSI_MSC in case the updating type indicated in the A-interface primitive indicates IMSI Detach.

The respective primitive is then forwarded to the created process. Henceforth, the coordinator will relay all service primitives from provider to the user and vice versa, until a request or indication for dialogue termination is received. This last primitive will be relayed, too, before the Coordinator process returns to idle state.

Location Management Coordinator VLR

On receipt of a dialogue request for the Location Management Application Context (see Receive_Open_Ind macro in subclause 21.1), the Location_Management_Coordinator will:

- terminate the procedure in case of parameter problems or if the MSC indicated version Vr protocol;
or
- continue as below, if the dialogue is accepted.

Depending on the first service primitive received from the MAP Provider in this dialogue, the user process is created:

- Update_Location_Area_VLR in case the primitive is a MAP_UPDATE_LOCATION_AREA indication;
- Detach_IMSI_VLR in case the primitive is a MAP_DETACH_IMSI indication.

In case a MAP_U_ABORT, MAP_P_ABORT or a premature MAP_CLOSE indication is received instead, the process returns to idle state. If a MAP_NOTICE indication is received, the dialogue towards the MSC is aborted and the process returns to idle state.

After creation of the user process the service primitive received from the provider is passed to the user process. Henceforth, the coordinator will relay all service primitives from provider to the user and vice versa, until a request or indication for dialogue termination is received. This last primitive will be relayed, too, before the Coordinator process returns to idle state.

Location Management Coordinator HLR

On receipt of a dialogue request for the Location Management Application Context (see Receive_Open_Ind macro in subclause 21.1), the Location_Management_Coordinator will:

- terminate the process in case of parameter problems; or
- revert to MAP version Vr protocol if the VLR requests version Vr protocol; or
- continue as described in the following, if the dialogue is accepted.

The user process is created depending on the first service primitive received from the MAP service provider within this dialogue:

- Update_Location_HLR if the primitive is a MAP_UPDATE_LOCATION indication;
- RESTORE_DATA_HLR if the primitive is a MAP_RESTORE_DATA indication.

If a MAP_NOTICE indication is received instead, the dialogue towards the MSC is terminated and the process returns to idle state.

After creation of the user process the service primitive received from the MAP service-provider is passed to the user process. Henceforth, the coordinator will relay all service primitives from MAP service-provider to the MAP service-user and vice versa, until a request or indication for dialogue termination is received. This last primitive will be relayed, too, before the Coordinator process returns to idle state.

Figure 16.1/1: Process Location_Management_Coordinator_MSC

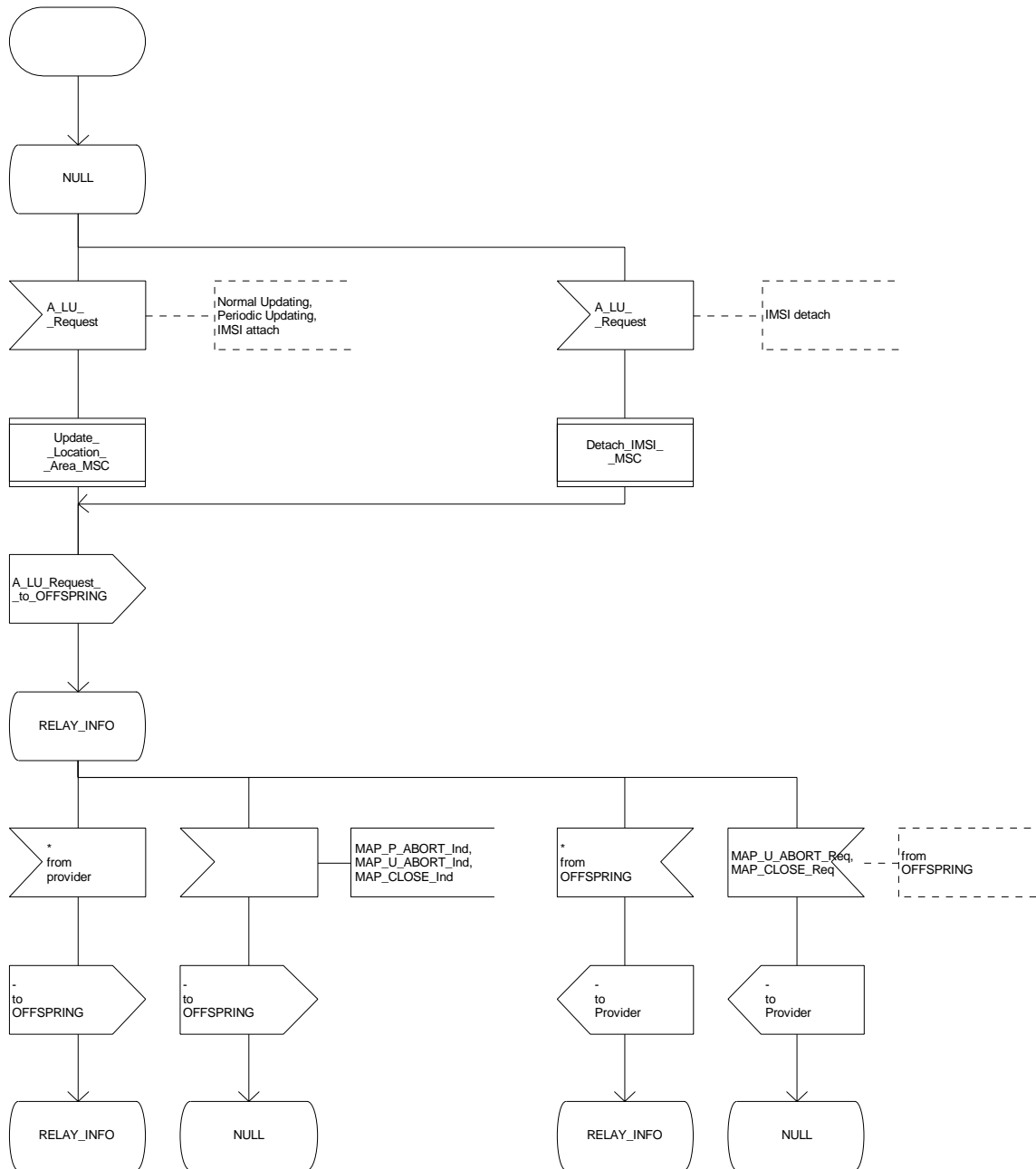


Figure 16.1/1

Figure 16.1/2: Process Location_Management_Coordinator_VLR

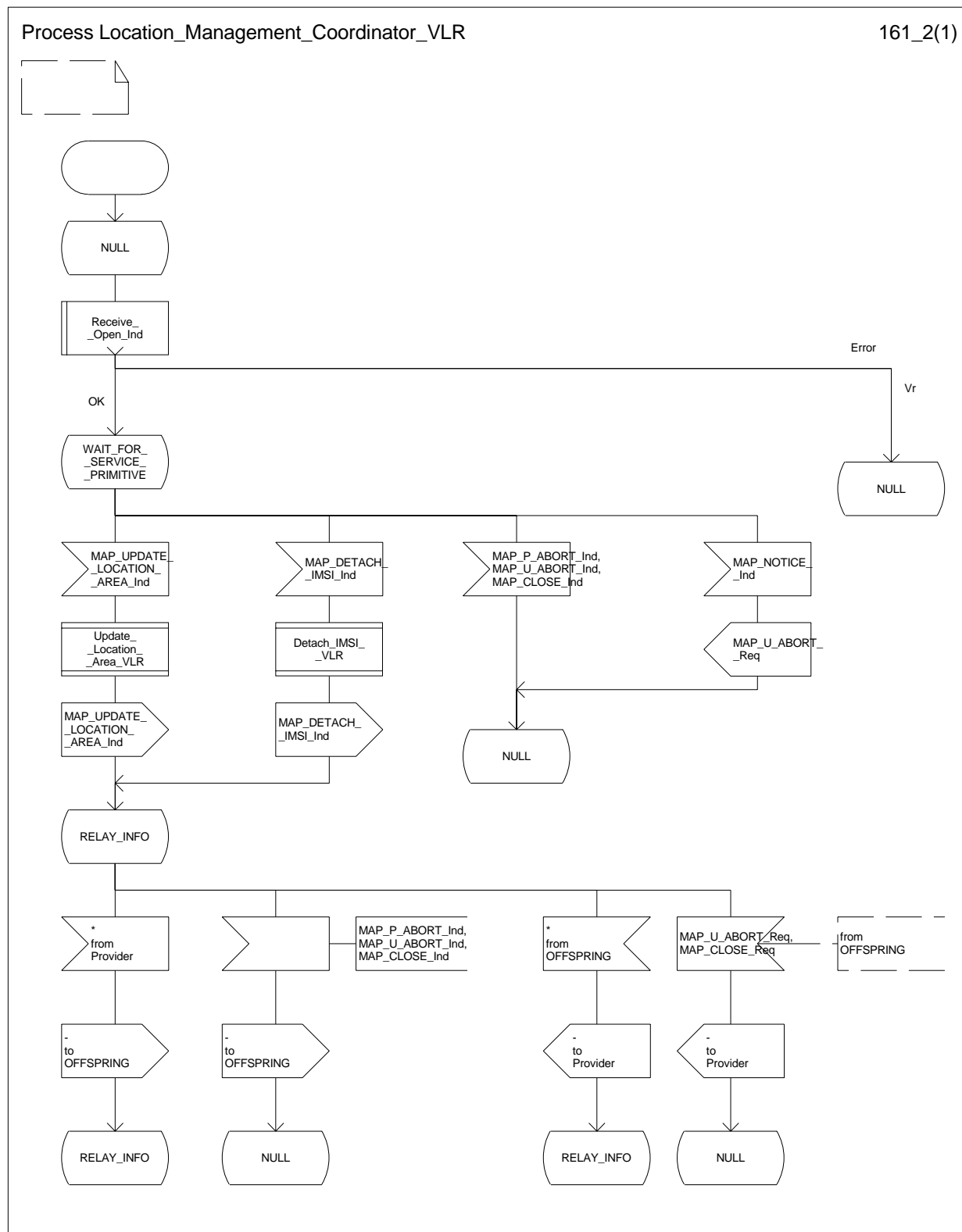
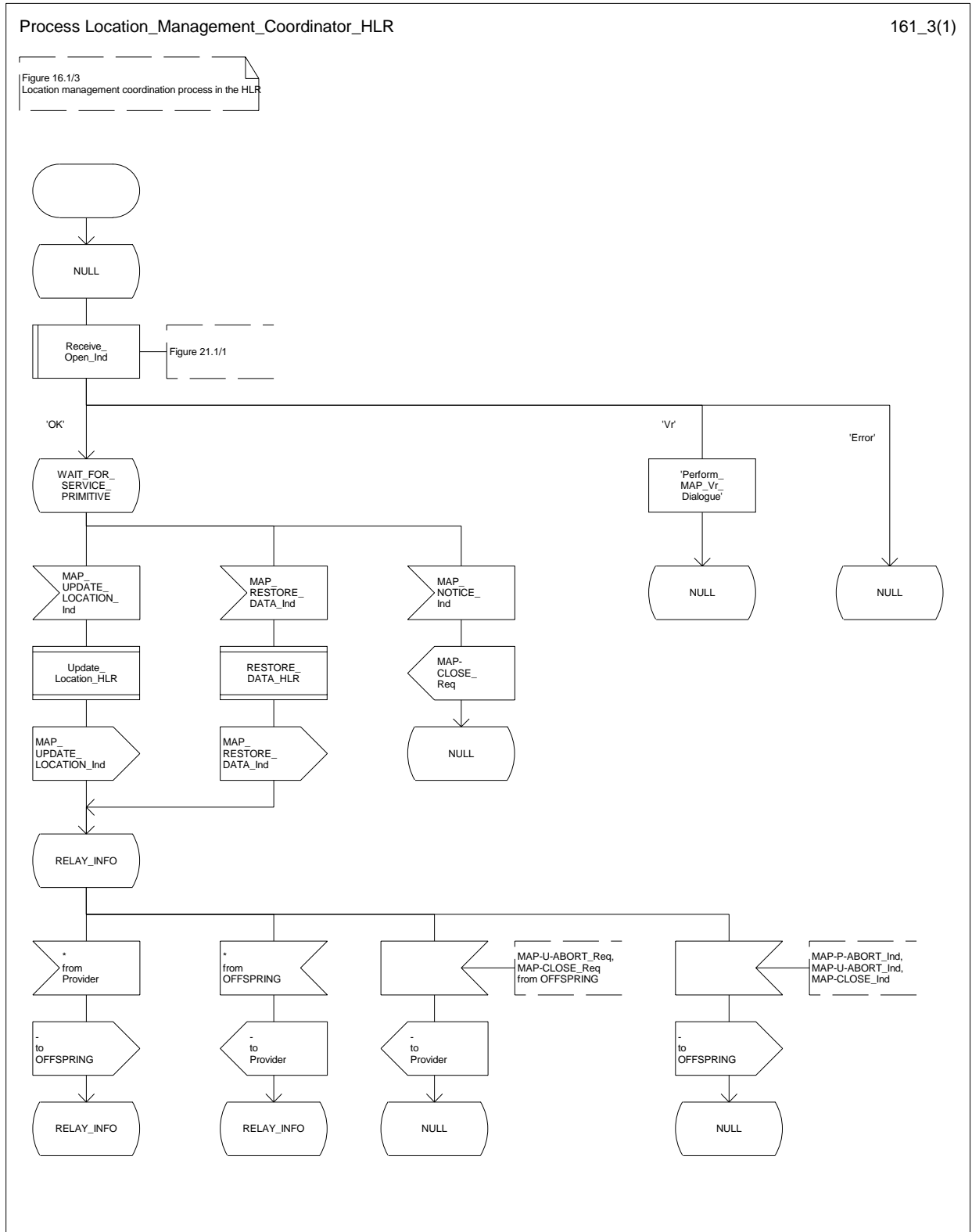


Figure 16.1/3: Process Location_Management_Coordinator_HLR



16.1.1 Location updating

16.1.1.1 General

The location updating procedure is used to update the location information held in the network. This location information is used to route incoming calls, short messages and unstructured supplementary service data to the roaming subscriber. Additionally, this procedure is used to provide the VLR with the information that a subscriber already registered, but being detached, is reachable again (IMSI Attach, see GSM 03.12). The use of this Detach / Attach feature is optional for the network operator.

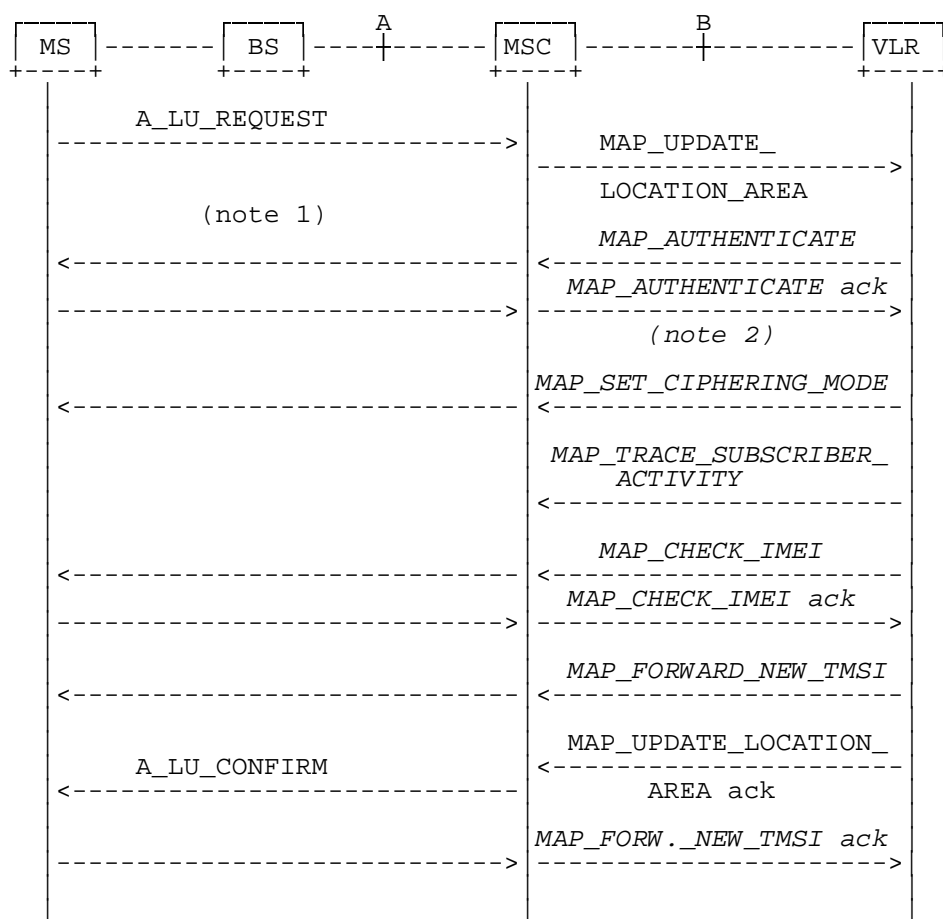
To minimize the updates of the subscriber's HLR, the HLR holds only information about the VLR and MSC the subscriber is attached to. The VLR contains more detailed location information, i.e. the location area the subscriber is actually roaming in. Therefore, the VLR needs to be updated at each location area change (see figure 16.1.1/1 for this procedure), whereas the HLR needs updating only in the following cases:

- when the subscriber registers in a new VLR, i.e. the VLR has no data for that subscriber;
- when the subscriber registers in a new location area of the same VLR and new routing information is to be provided to the HLR (change of MSC area);
- if the indicator "Confirmed by HLR" or the indicator "Location Information Confirmed in HLR" is set to "Not Confirmed" because of HLR or VLR restoration, and the VLR receives an indication that the subscriber is present.

If a mobile subscriber registers in a visitor location register (VLR) not holding any information about this subscriber and is identified by a temporary mobile subscriber identity (TMSI) allocated by a previous visitor location register (PVLR), if the PVLR identity can be derived from LAI the new VLR must obtain the IMSI from PVLR to identify the HLR to be updated (see figure 16.1.1/2). If the IMSI cannot be retrieved from PVLR, it is requested from the MS (see figure 16.1.1/3).

The following MAP services are invoked by the location update procedure:

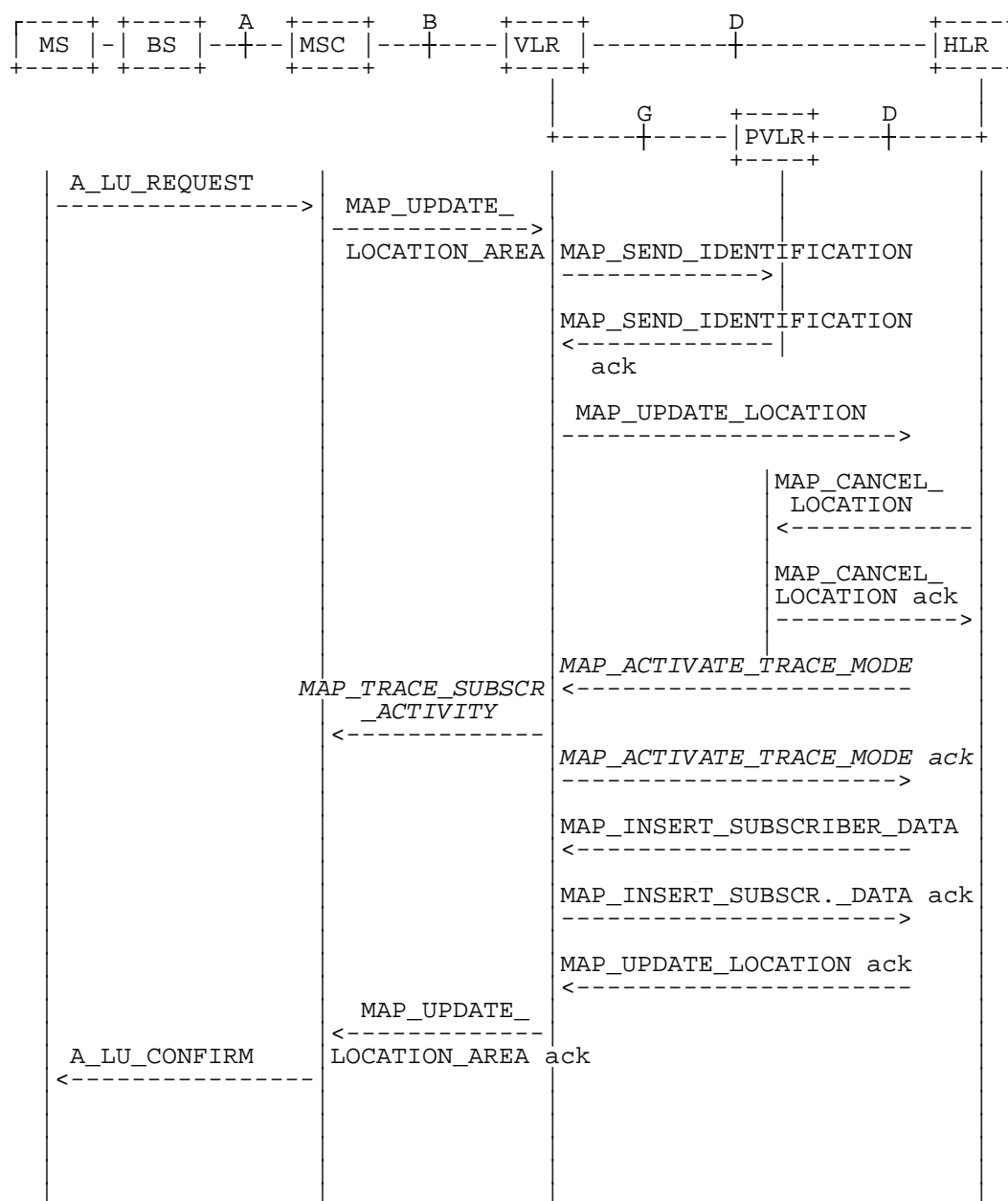
MAP_UPDATE_LOCATION_AREA (see subclause 6.1);
MAP_UPDATE_LOCATION (see subclause 6.1);
MAP_CANCEL_LOCATION (see subclause 6.1);
MAP_INSERT_SUBSCRIBER_DATA (see subclause 6.8);
MAP_SEND_IDENTIFICATION (see subclause 6.1);
MAP_PROVIDE_IMSI (see subclause 6.9);
MAP_AUTHENTICATE (see subclause 6.5);
MAP_SET_CIPHERING_MODE (see subclause 6.6);
MAP_FORWARD_NEW_TMSI (see subclause 6.9);
MAP_CHECK_IMEI (see subclause 6.7);
MAP_ACTIVATE_TRACE_MODE (see subclause 7.2);
MAP_TRACE_SUBSCRIBER_ACTIVITY (see subclause 7.2).



NOTE 1: For details of the procedure on the radio path, see GSM 04.08. The services shown in dotted lines indicate the trigger provided by the signalling on the radio path, and the signalling triggered on the radio path.

NOTE 2: Optional services are printed in *italics*.

Figure 16.1.1/1: Interface and services for location updating when roaming within an visitor location registers area (without need to update HLR)



NOTE: The optional procedures in figure 16.1.1/1 apply here respectively.

Figure 16.1.1/2: Interface and services for location updating when changing the VLR area

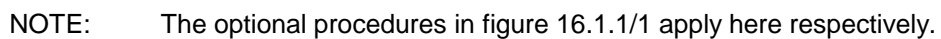


Figure 16.1.1/3: Interface and services for location updating involving both a VLR and an HLR, when IMSI can not be retrieved from the previous VLR

16.1.1.2 Detailed procedure in the MSC

Figure 16.1.1/4 shows the MSC process for location register updating, containing macro calls for:

Receive_Open_Cnf	subclause 21.1;
Authenticate_MSC	subclause 21.5;
Check_IMEI_MSC	subclause 21.6;
Obtain_IMSI_MSC	subclause 21.8;
Trace_Subscriber_Activity_MSC	subclause 21.9.

For structuring purposes, the second part of the process is placed into the macro Update Location Completion MSC, which is specific to this process (see figure 16.1.1/5).

When the MSC receives an A_LU_REQUEST (normal location updating, periodic location updating or IMSI attach) for a subscriber via the radio path, the MSC opens a dialogue to the VLR (MAP_OPEN request without any user specific parameters) and sends a MAP_UPDATE_LOCATION_AREA request, containing the parameters provided in the A_LU_REQUEST by the MS or BSS (for the parameter mapping see GSM 09.10).

If the dialogue is rejected or the VLR indicates a fallback to the version Vr procedure (see Receive_Open_Cnf macro in subclause 21.1), the MSC will send an A_LU_Rej towards the MS and terminate the procedure.

If the dialogue is accepted, the VLR will process this updating request, invoking optionally the MAP_PROVIDE_IMSI, MAP_TRACE_SUBSCRIBER_ACTIVITY, MAP_CHECK_IMEI or the MAP_AUTHENTICATE services first (see subclause 16.1.1.3 for initiation conditions, subclause 21 for macros defining the handling of services in the MSC). For these macros there are two possible outcomes:

- a positive outcome, in which case the process continues waiting for the MAP_UPDATE_LOCATION_AREA confirmation; or
- an error is reported, in which case the process terminates (not applicable for Trace_Subscriber_Activity_MSC, which has only a positive outcome).

After receiving the MAP_UPDATE_LOCATION_AREA indication and handling these optional services, the VLR will decide whether a new TMSI need to be allocated to the subscriber or not.

Updating without TMSI reallocation

If the VLR does not reallocate the TMSI, the MSC will receive a MAP_UPDATE_LOCATION_AREA confirmation next (figure 16.1.1/4).

- if there are no parameters with this primitive, updating was successful and a confirmation will be sent to the MS;
- if there is an error cause contained in the received primitive, this cause will be mapped to the corresponding cause in the confirmation sent to the MS (see GSM 09.10 for the mapping of messages and causes).

Updating including TMSI reallocation

This case is covered by the macro Update Location Completion MSC given in figure 16.1.1/5. The MSC will upon receipt of a MAP_SET_CIPHERING_MODE request send a ciphering command towards BSS/MS. Thereafter, the MAP_FORWARD_NEW_TMSI indication and the MAP_UPDATE_LOCATION_AREA confirmation are received in arbitrary order, causing a confirmation on the radio path containing both new LAI and new TMSI. If the MAP_UPDATE_LOCATION_AREA confirmation contains any error, the updating request is rejected towards the MS:

- the MS will confirm receipt of the new TMSI, resulting in an empty MAP_FORWARD_NEW_TMSI response terminating the dialogue;
- if there is no confirmation received from the A-interface, the dialogue is terminated locally.

Before receiving a MAP_UPDATE_LOCATION_AREA confirmation, the MSC may receive a MAP_CHECK_IMEI indication. Handling of this indication, comprising IMEI request towards the MS and IMEI checking request towards the EIR, is given in the macro description in subclause 21.6. The result may either be to return to the state Wait for TMSI or to return to terminate.

Forwarding the Check SS Indication

When the VLR receives a MAP_FORWARD_CHECK_SS_INDICATION_Ind during the Update LOCATION Area process, this indication is relayed to the MS (see GSM 09.11 for detailed interworking) and the MSC remains in the current state.

Abort handling

If the VLR receives a MAP_U_ABORT, a MAP_P_ABORT or a premature MAP_CLOSE indication from the VLR during the location update process, the MSC terminates the process by sending an A_LU_CONFIRM containing the error cause Updating Failure to the MS. If the MSC had already confirmed the location update towards the MS, the process terminates without notification towards the A-interface.

If the MSC receives a MAP_NOTICE indication, it issues a MAP_CLOSE and terminates the A-interface dialogue, and the process terminates.

When the procedure is terminated abnormally on the radio path, the dialogue towards the VLR is aborted with the appropriate diagnostic information, and the procedure terminates.

Figure 16.1.1/4: Process Update_Location_Area_MSC

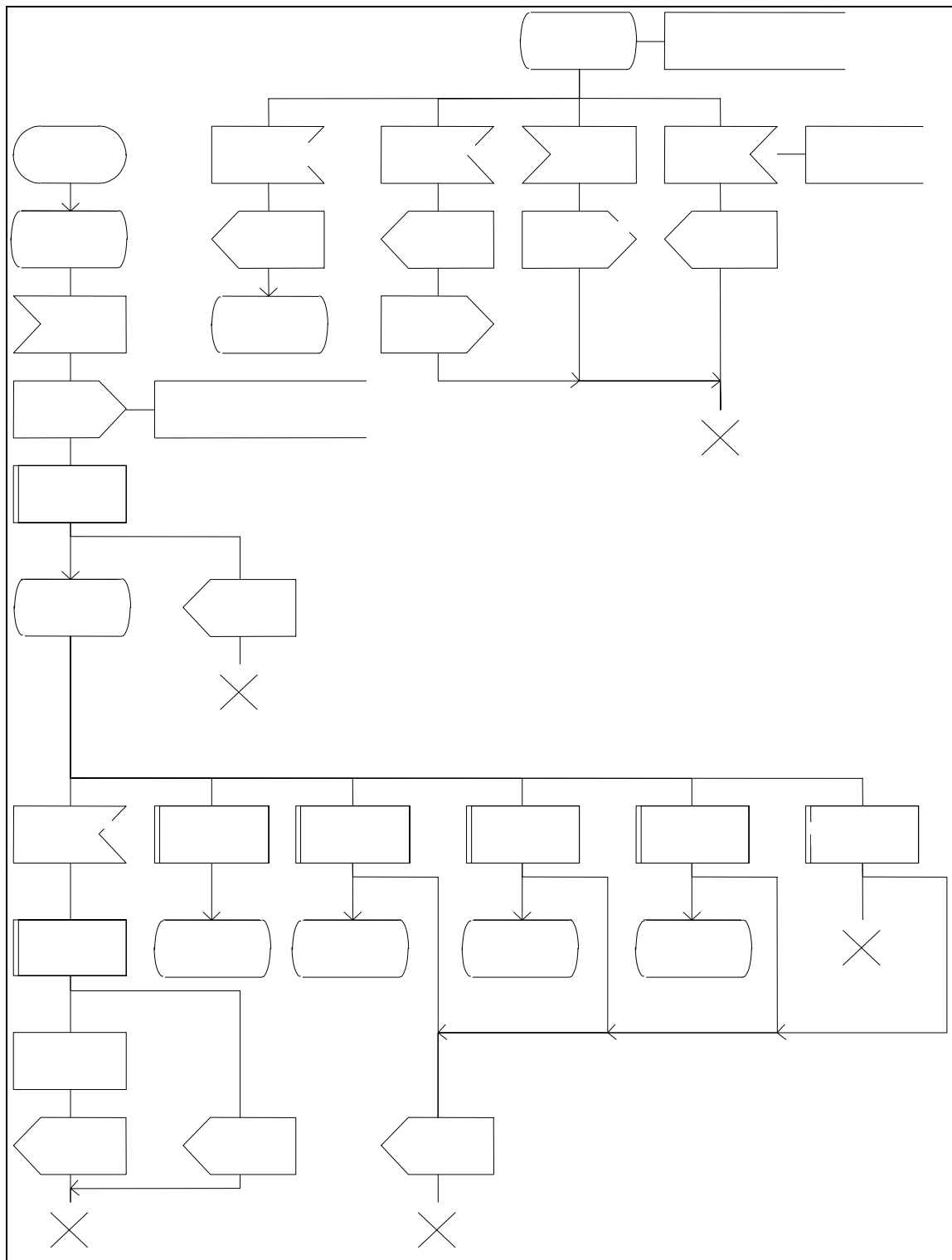
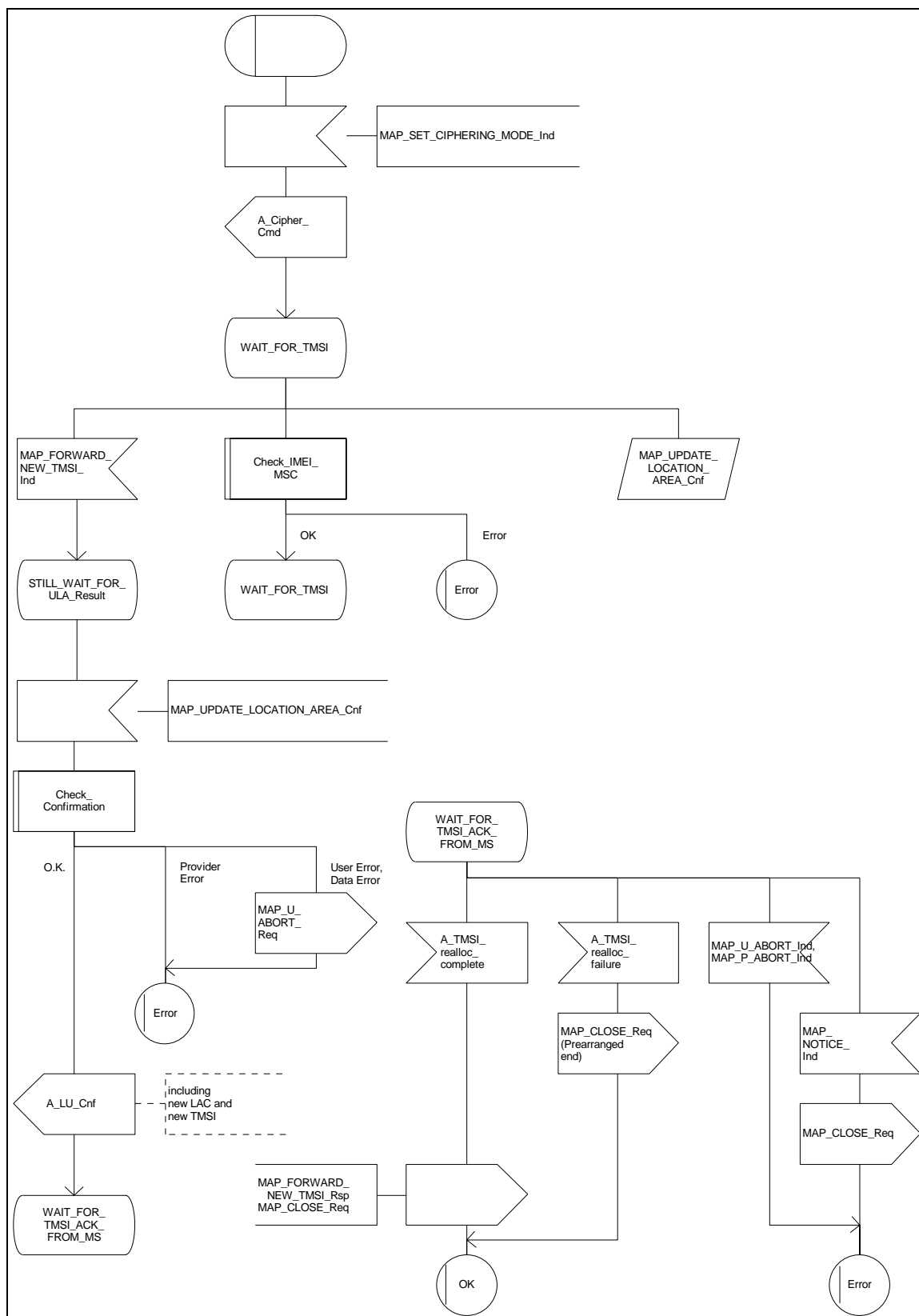


Figure 16.1.1/5: Macro Update_Location_Completion_MSC



16.1.1.3 Detailed procedure in the VLR

Figure 16.1.1/6 shows the process for location updating in the VLR. The following general macros are used:

Receive_Open_Ind	subclause 21.1;
Receive_Open_Cnf	subclause 21.1;
Authenticate_VLR	subclause 21.5;
Check_IMEI_VLR	subclause 21.6;
Insert_Subscriber_Data_VLR	subclause 21.7;
Obtain_IMSI_VLR to request the IMSI for the subscriber	subclause 21.8;
Activate_Tracing_VLR and Trace_Subscriber_Activity_VLR	subclause 21.9;
Subscriber_Present_VLR	subclause 21.10.

Additionally, the process specific macro

Location_Update_Completion_VLR, for optional initiation of Ciphering and TMSI reallocation as for acknowledgement of the MAP_UPDATE_LOCATION_AREA service, see figure 16.1.1/7,

and the optional process specific macro

VLR_Update_HLR to update the HLR and download subscriber data from there, see figure 16.1.1/8,

are invoked by this process.

Process Initiation

The location area updating process will be activated by receiving a MAP_UPDATE_LOCATION_AREA indication from the MSC. If there are parameter errors in the indication, the process is terminated with the appropriate error sent in the MAP_UPDATE_LOCATION_AREA response to the MSC. Else, The behaviour will depend on the subscriber identity received, either an IMSI or an TMSI.

Updating using IMSI

If the subscriber identity is an IMSI, the VLR checks whether the subscriber is unknown (i.e. no IMSI record). If so, the indicator "Location Information Confirmed in HLR" is set to "Not Confirmed" to initiate HLR updating later on. If the IMSI is known, the VLR checks whether the previous location area identification (LAI) provided in the primitive received from the MSC belongs to this VLR. If it does not, the indicator "Location Information Confirmed in HLR" is set to "Not Confirmed" to initiate HLR updating later on. The process may continue in both cases with the authentication check (see below).

Updating using TMSI

If the subscriber identity is a TMSI, the VLR checks whether the previous location area identification (LAI) provided in the primitive received from MSC belongs to an area of this VLR:

- if so, the TMSI will be checked. In case of location area change within a VLR, the TMSI should be known and the process may continue with the authentication check. Additionally, the indicator "Location Information Confirmed in HLR" is set to "Not confirmed" and the trace activity status is checked in case the target Location Area Id belongs to a new MSC.
- if the TMSI is not known or the subscriber data stored are incomplete, e.g. because the new LA belongs to a different VLR or due to VLR restoration, the indicator "Confirmed by VLR" is set to "Not Confirmed" to initiate HLR updating later on.

If the subscriber has not already been registered in the VLR, i.e. the previous LAI belongs to a different VLR, the indicators "Confirmed by HLR" and "Location Information Confirmed in HLR" are set to "Not Confirmed" and the VLR checks whether the identity of the Previous VLR (PVLR) is derivable from the previous LAI:

- if so, the IMSI and authentication parameters are requested from that VLR using the MAP_SEND_IDENTIFICATION service (see sheet 3 of figure 16.1.1/6), containing the subscriber's TMSI.
- if the dialogue is rejected by the PVLR, the process continues requesting the IMSI from the MS. In case the PVLR reverts to the MAP version Vr dialogue, the VLR will perform the respective procedure of version Vr, too, with outcomes as for the current MAP version dialogue. Else, the process waits for the respective MAP_SEND_IDENTIFICATION response from the PVLR:
 - if the IMSI is received in that primitive, the process continues with the authentication check;
 - if the IMSI is not received from the previous VLR for any reason, the dialogue to the PVLR is terminated and the IMSI will be requested from the MS;
 - if a MAP_NOTICE indication is received from the PVLR, the dialogue will be terminated by sending a MAP_CLOSE indication, and the process continues requesting the IMSI from the MS;
 - if a MAP_P_ABORT or MAP_U_ABORT indication is received from the MSC while waiting for the MAP_SEND_IDENTIFICATION response, the process is terminated;
 - if a MAP_NOTICE indication is received from the MSC while waiting for the MAP_SEND_IDENTIFICATION response, the dialogue with the PVLR will be aborted by sending a MAP_U_ABORT indication (Remote Operations Failure), the dialogue with the MSC will be terminated by sending a MAP_CLOSE and the process terminates;
- if the identity of the previous VLR cannot be derived, the process continues by requesting the IMSI from the MS.

Requesting IMSI from the MS

For requesting the IMSI from the MS, the macro Obtain_IMSI_VLR described in subclause 21.8 is invoked (see figure 16.1.1/6 sheet 3). The outcome will be:

- OK, i.e. receipt of IMSI, in which case the process continues with the authentication check described below; or
- receipt of an Absent Subscriber error, indicating that the MS did not respond. In this case the System Failure error is reported in the MAP_UPDATE_LOCATION_AREA response towards the MSC and the updating process is terminated;
- aborted, i.e. the MSC dialogue has been released while waiting for the IMSI. In this case the updating process is terminated, too.

Authentication check

After a subscriber identity has been received, either in the service indication or by an explicit request procedure, the VLR checks whether authentication of this identity is required (see figure 16.1.1/6 sheet 2). If so, the authentication macro described in subclause 21.5 is invoked. The outcome of this macro can be:

- OK, i.e. the subscriber has been authenticated successfully, in which case the process is continued by setting the indicator "Confirmed by Radio Contact" to "Confirmed" and updating the location information held in the register. Thereafter,
 - if one or both of the indicators "Confirmed by HLR" and "Location Information Confirmed in HLR" is set to "Not Confirmed", HLR updating is invoked first;
 - otherwise the process continues with the Location Update Completion VLR macro described below, and the register is updated after successful completion of this macro.

- Illegal subscriber, i.e. there was a mismatch between expected and received SRES. The VLR checks whether authentication had been performed using the TMSI, in which case a new authentication attempt with IMSI may be started (VLR operator option).
 - if so, the process continues by requesting the IMSI from the MS;
 - else, the Illegal Subscriber error is reported in the MAP_UPDATE_LOCATION_AREA response.
- Unknown Subscriber, i.e. the IMSI given is unknown in the HLR. In this case, the subscriber data are deleted in the VLR and the same error is returned in the MAP_UPDATE_LOCATION_AREA response.
- Procedure error, i.e. the authentication process was unsuccessful for some other reason, e.g. because of a failure while requesting authentication information from the HLR. In this case the System Failure error is reported in the MAP_UPDATE_LOCATION_AREA response.
- Null, indicating impossible dialogue continuation (e.g. termination of the radio path), and leading to procedure termination without any further action.

Updating the HLR

If the HLR is to be updated, the VLR_Update_HLR macro described below is performed, with one of the following results (see sheet 4 of figure 16.1.1/6):

- OK, if HLR updating has been completed successfully. The response will contain the HLR number as parameter. Next, the Location_Update_Completion VLR macro is invoked (checking amongst others the roaming restrictions and regional subscription data), and upon successful outcome of this macro the register is updated and the process terminates.
- Roaming Not Allowed, qualified by PLMN Roaming Not Allowed if the location information indicates a PLMN for which the subscriber has no subscription or if the subscribers HLR cannot be reached (e.g. SS7 links to the subscribers HPLMN do not yet exist). In this case, the error Roaming Not Allowed qualified by PLMN Roaming Not Allowed is sent in the MAP_UPDATE_LOCATION_AREA response. The Subscriber Data are deleted in the VLR.
- if Roaming Not Allowed was qualified by the parameter Operator Determined Barring, the same value is sent in the MAP_UPDATE_LOCATION_AREA response to the MSC. The subscriber data are deleted in the VLR.
- Unknown Subscriber, if the subscriber is not known in the HLR. In this case, the subscriber data are deleted in the VLR, and the same error is sent in the MAP_UPDATE_LOCATION_AREA response.
- Procedure error, if there occurs some other error during HLR updating (e.g. abort of the connection to HLR):
 - if the VLR can proceed in stand alone mode (VLR operator option), the Location Update Completion VLR macro is invoked to complete the VLR updating, and the indicator "Confirmed by HLR" remains unchanged;
 - otherwise, the System Failure error is sent in the MAP_UPDATE_LOCATION_AREA response.
- Aborted, indicating that during HLR updating the MSC dialogue has been terminated. In this case, the updating process terminates without any further action.

The macro Location Update Completion VLR

This macro completes the VLR updating process. First, the VLR checks whether there is a roaming restriction for the subscriber (see figure 16.1.1/7):

- if the target LA is not allowed for the subscriber due to national roaming restrictions, the error Roaming Not Allowed with cause National Roaming Not Allowed is returned in the MAP_UPDATE_LOCATION_AREA response towards the MSC.

The subscriber data are not deleted from VLR, to avoid unnecessary HLR updating when roaming into other LAs of the same MSC. An indication that the subscriber is not allowed to roam is set in the VLR (LA Not Allowed Flag set to not allowed). As a consequence the subscriber is not reachable (checked for MTC, SMS and MT USSD) and cannot perform outgoing actions (checked in Access Management).

- if the target LA is not allowed for the subscriber because of regional subscription data (Zone Code List) or Roaming Restriction Due To Unsupported Feature stored in the VLR, the error Roaming Not Allowed with cause Location Area Not Allowed is returned towards the MSC in the MAP_UPDATE_LOCATION_AREA response.

Also in this case the subscriber data are not deleted from VLR, to avoid unnecessary HLR updating when roaming into other LAs of the same MSC. The LA Not Allowed Flag is set to not allowed in the VLR.

- if, after check of possible roaming restrictions, the subscriber is allowed to roam in the target LA, the LA Not Allowed Flag is set to allowed (if necessary), the IMSI Detached Flag is set to attached and the process SUBSCRIBER_PRESENT_VLR is started; this may inform the HLR that the subscriber is present again to retry an SMS delivery (see subclause 16.1.1.7). Thereafter, the VLR checks whether TMSI reallocation is required.
 - if so, the VLR sends a MAP_SET_CIPHERING_MODE request containing:
 - Ciphering Mode (version 1 GSM); and
 - Kc, the cipher key to be used.
- if IMEI checking is required by the operator, the VLR will invoke the CHECK_IMEI_VLR macro (see subclause 21.6) to initiate both requesting IMEI from the MS and checking of this IMEI towards the EIR. As result either the service is granted, with process continuation as given below, or the service is rejected, in which case the VLR marks the subscriber as detached and returns an Illegal Equipment error in the MAP_UPDATE_LOCATION_AREA response before the process terminates.
 - the VLR then sends a MAP_FORWARD_NEW_TMSI request containing the new TMSI, and the MAP_UPDATE_LOCATION_AREA response containing no parameters. The process will thereafter wait for the MAP_FORWARD_NEW_TMSI confirm. If this indicates a negative outcome, or if a MAP_P_ABORT or a MAP_U_ABORT primitive is received, the old TMSI is frozen. Subsequent accesses of the MS shall be accepted with both old or new TMSI.
- if TMSI reallocation is not required, the VLR invokes the CHECK_IMEI_VLR macro (see subclause 21.6) to initiate both requesting IMEI from the MS and checking of this IMEI towards the EIR, if IMEI Checking is required by the operator. As a result, either the service is granted, in which case the MAP_UPDATE_LOCATION_AREA response is sent without any parameters, or the service is rejected, in which case an Illegal Equipment error is returned in the MAP_UPDATE_LOCATION_AREA response, before the process terminates.

In all cases where the VLR sends a MAP_UPDATE_LOCATION_AREA response to the MSC, the dialogue towards the MSC is terminated by a MAP_CLOSE request with parameter Release Method indicating Normal Release.

The macro VLR Update HLR

This macro is invoked by the VLR process for location updating or by some other process handling the first subscriber access to the network after a register failure in order to perform HLR updating. If the VLR does not know the subscribers HLR (e.g. no IMSI translation exists as there are not yet any SS7 links to the subscribers HPLMN), the error Roaming Not Allowed with cause PLMN Roaming Not Allowed is returned.

If the subscribers HLR can be reached, the VLR opens a dialogue towards the HLR (see figure 16.1.1/8) by sending a MAP_OPEN request without any user specific parameters, together with a MAP_UPDATE_LOCATION request containing the parameters

- IMSI, identifying the subscriber;
- Location Info, containing the MSC number;

- VLR Number, the E.164 address of the VLR, to be used by the HLR when addressing the VLR henceforth (e.g. when requesting an MSRN);
- the LMSI as an VLR operator option; this is a subscriber identification local to the VLR, used for fast data base access.

In case the HLR rejects dialogue opening (see subclause 21.1), the VLR will terminate the procedure indicating procedure error. If the HLR indicates version 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 subclause 21.7;

NOTE: The HLR may repeat this service several times depending on the amount of data to be transferred to the VLR and to replace subscription data in case they are not supported by the VLR.

- a MAP_ACTIVATE_TRACE_MODE indication, handled by the macro Activate_Tracing_VLR defined in subclause 21.9;
- a MAP_FORWARD_CHECK_SS_INDICATION_ind. This indication will be relayed to the MSC without any change of the current state.
- the MAP_UPDATE_LOCATION confirmation:
 - if this confirmation contains the HLR Number, this indicates that the HLR has passed all information and that updating has been successfully completed. The VLR is updated using the parameters provided in the service and needed by the VLR. If certain parameters are not needed in the VLR, e.g. because some service is not supported, the corresponding data may be discarded. The VLR sets the "Confirmed by HLR" and "Location information confirmed in HLR" indicators to "Confirmed" to indicate successful subscriber data updating;
 - if the confirmation contains an User error cause (Unknown Subscriber, Roaming Not Allowed or some other), the process calling the macro continues accordingly. In the last case, the subscriber data are marked as incomplete by setting the indicators "Confirmed by HLR" and "Location information confirmed in HLR" to "Not Confirmed". The same holds if there is a Provider error or a Data error in the confirmation;
- a MAP_P_ABORT, MAP_U_ABORT, or MAP_CLOSE indication. In these cases, the subscriber data are marked to be incomplete and the process continues as in the case of an error reported by the HLR;
- a MAP_NOTICE indication. Then, the dialogue towards the HLR is terminated, the subscriber data are marked to be incomplete and the process continues as in the case of an error reported by the HLR;
- if during HLR updating the VLR receives a MAP_P_ABORT, MAP_U_ABORT or a MAP_CLOSE indication concerning the MSC dialogue, the process is terminated by sending a MAP_U_ABORT request towards the HLR, and subscriber data are marked to be incomplete;
- if during HLR updating the VLR receives a MAP_NOTICE indication concerning the MSC dialogue, the dialogue with the MSC is terminated by sending a MAP_CLOSE, the dialogue with the HLR is terminated by sending a MAP_U_ABORT, subscriber data are marked to be incomplete and the process is terminated.

Abort Handling

If the VLR receives a MAP_NOTICE indication from the MSC while waiting for a MAP service primitive, the VLR will terminate the MSC dialogue by sending a MAP_CLOSE and any pending HLR dialogue by sending a MAP_U_ABORT (Remote Operations Failure), and the process is terminated.

Figure 16.1.1/6 (sheet 1 of 4): Process Update_Location_Area_VLR

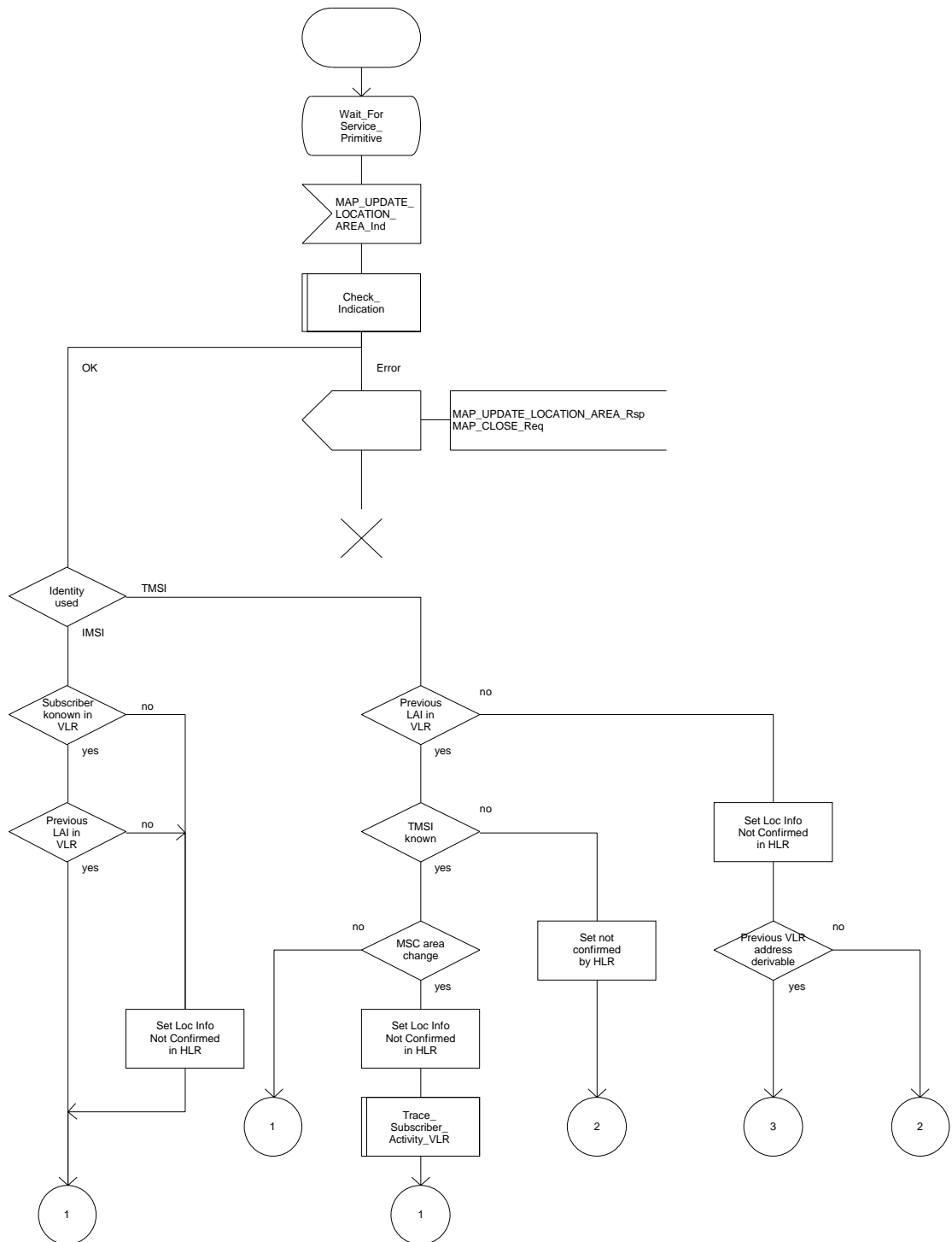


Figure 16.1.1/6 (sheet 2 of 4): Process Update_Location_Area_VLR

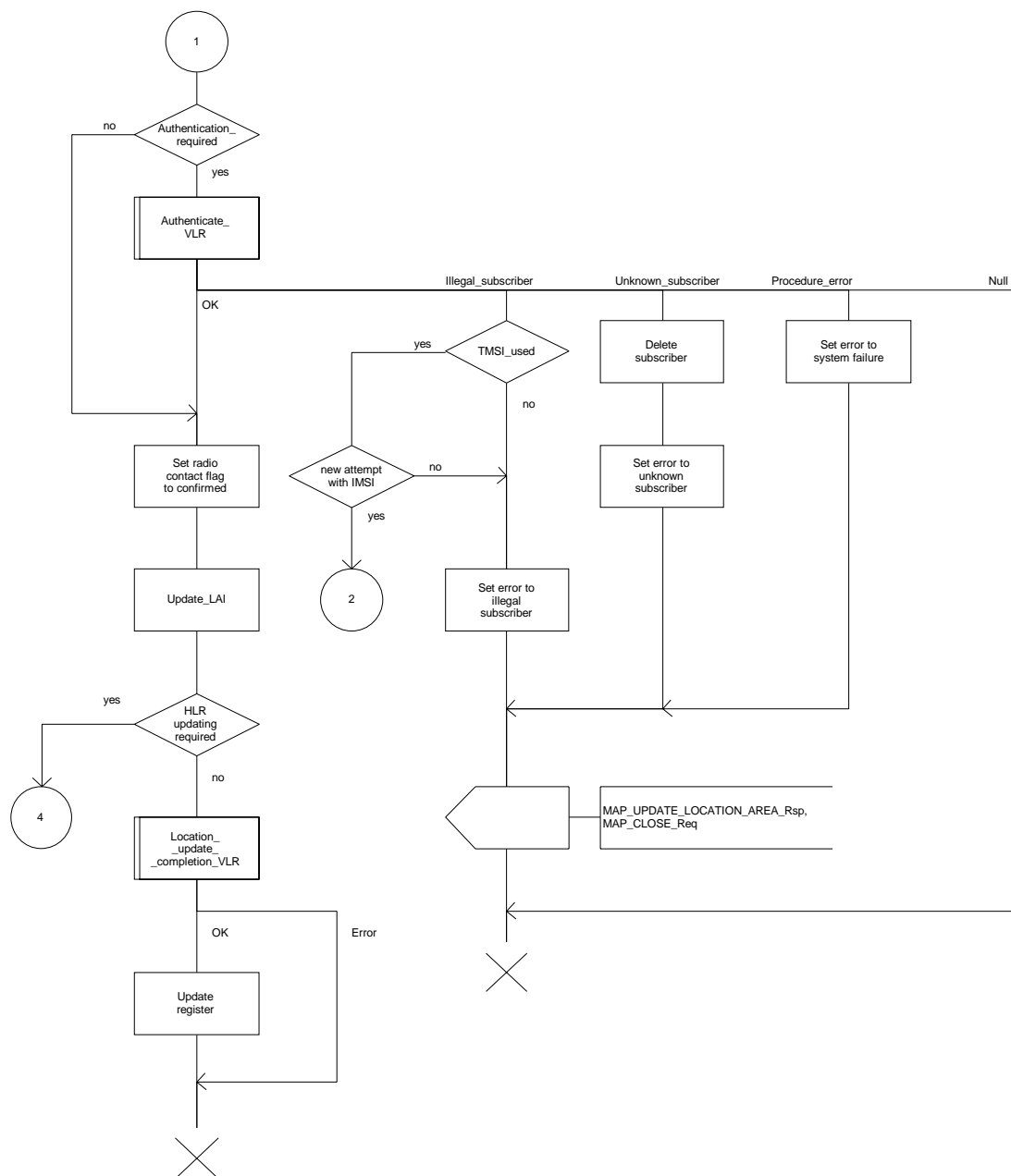


Figure 16.1.1/6 (sheet 3 of 4): Process Update_Location_Area_VLR

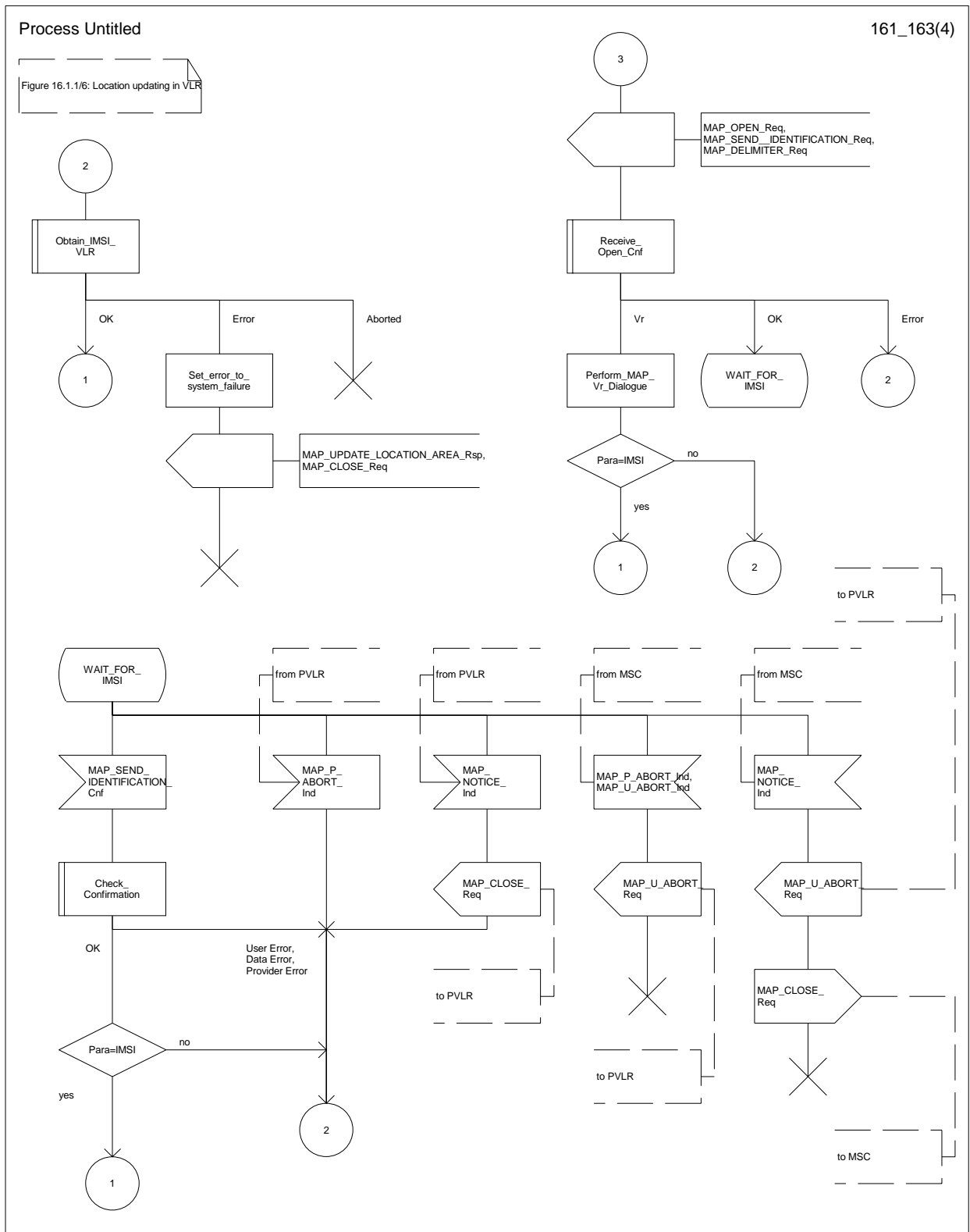


Figure 16.1.1/6 (sheet 4 of 4): Process Update_Location_Area_VLR

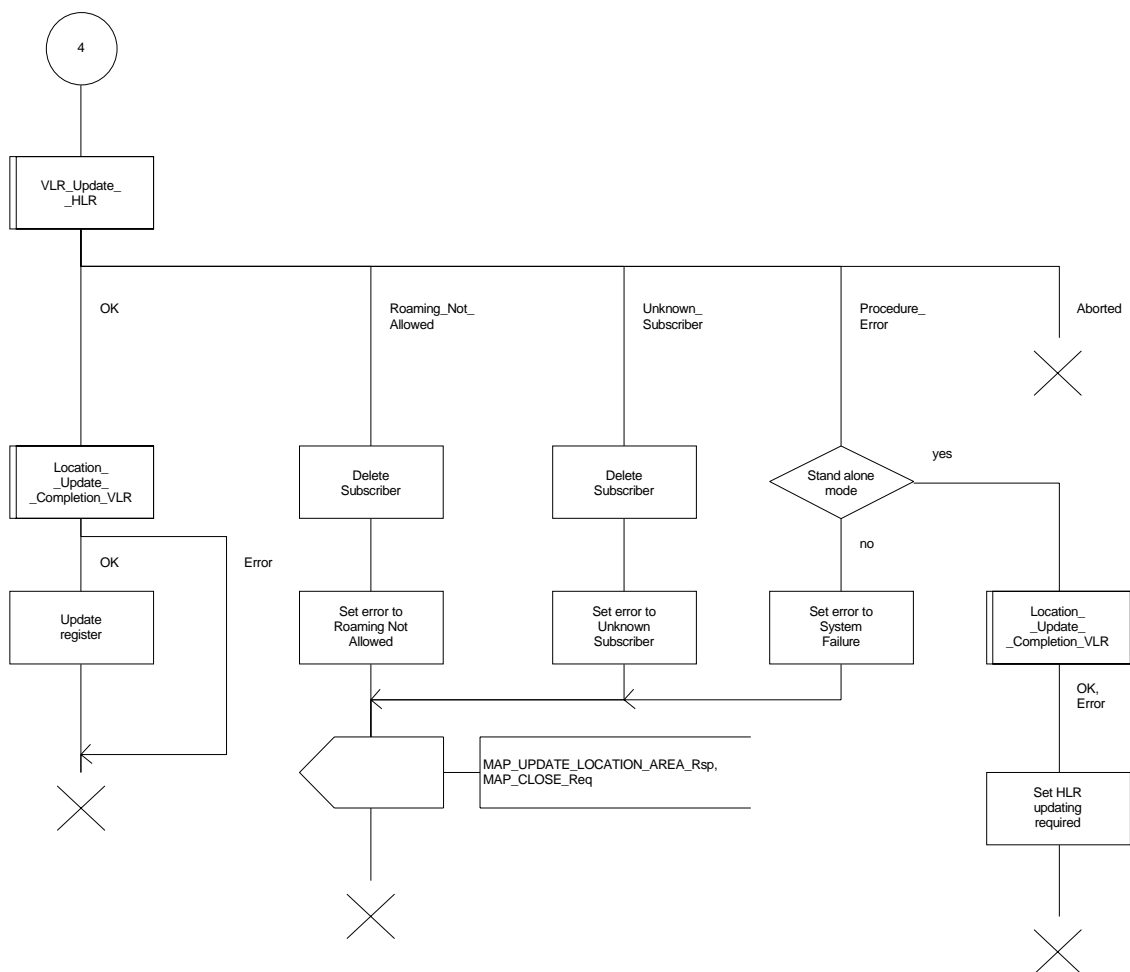


Figure 16.1.1/7: Macro Location_Update_Completion_VLR

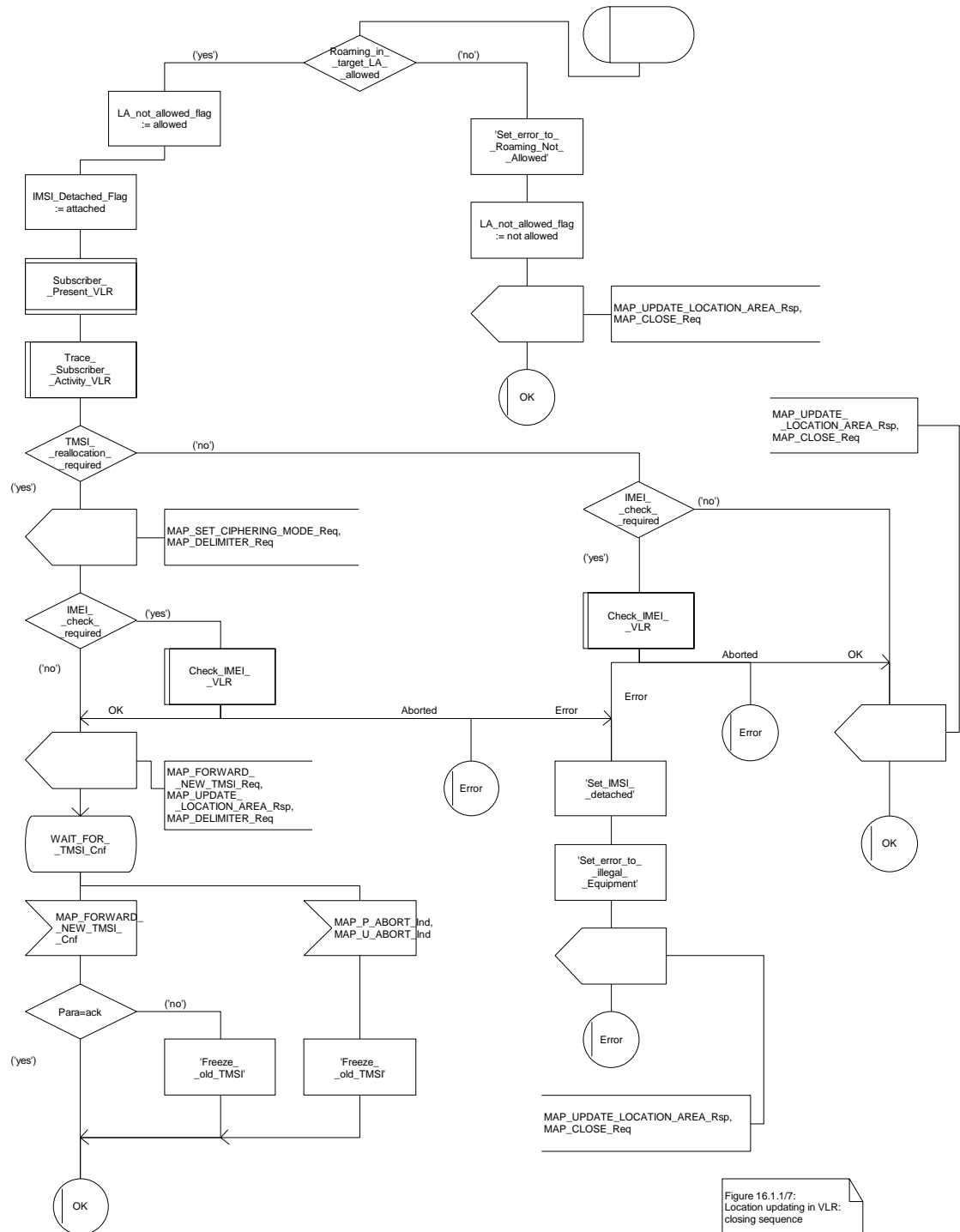


Figure 16.1.1/7: Location updating in VLR: closing sequence

Figure 16.1.1/8 (sheet 1 of 2): Macro VLR_Update_HLR

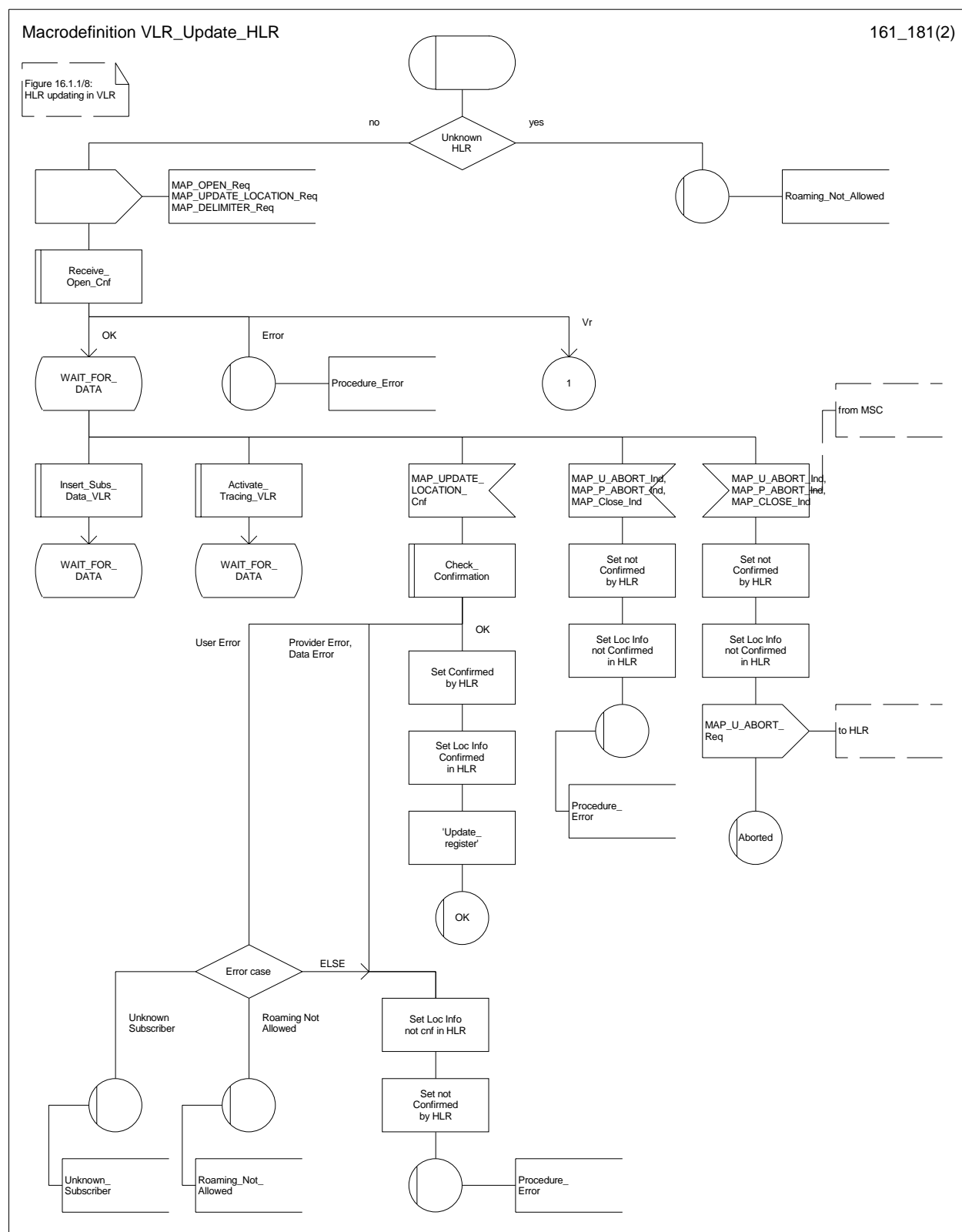
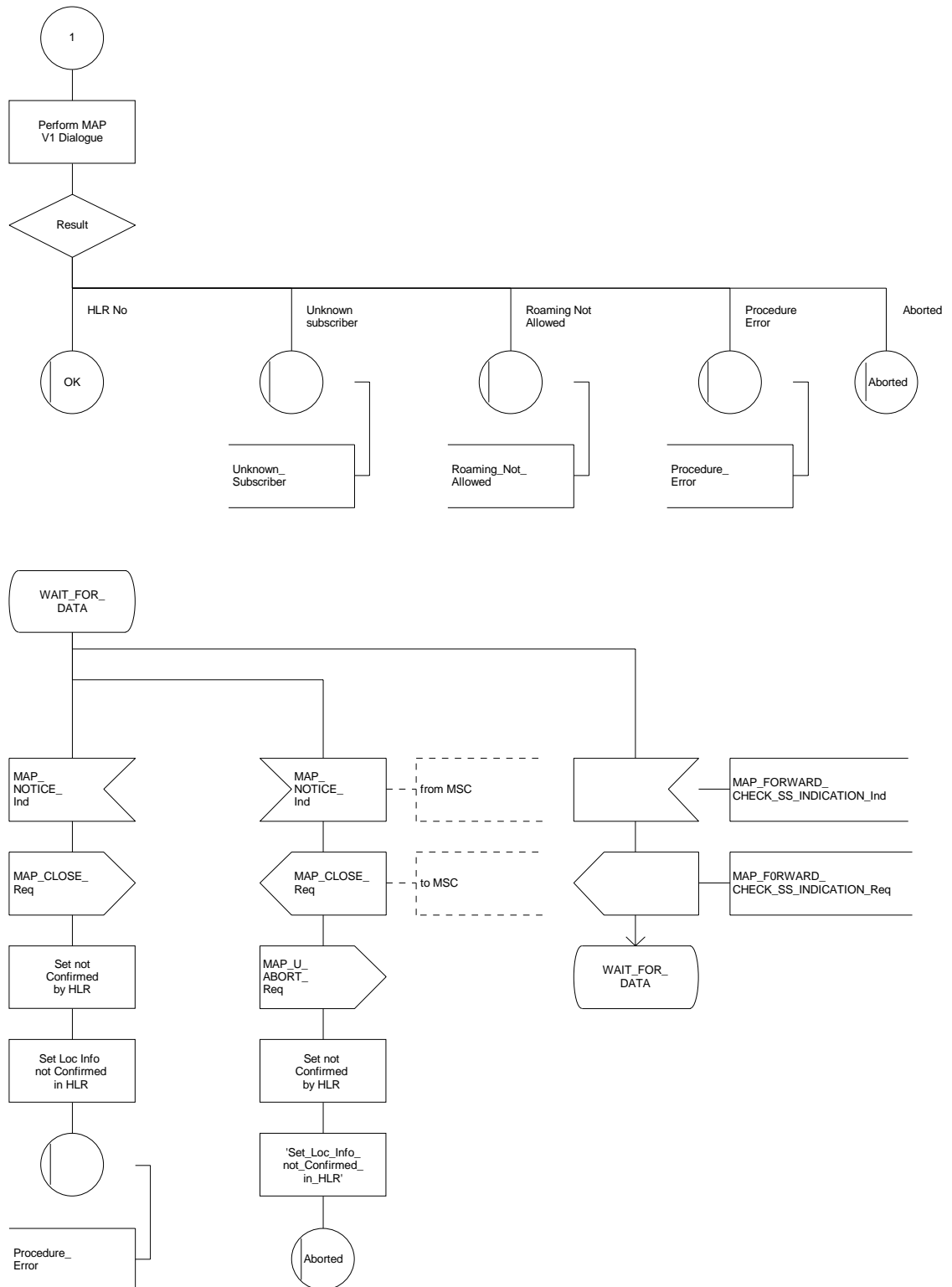


Figure 16.1.1/8 (sheet 2 of 2): Macro VLR_Update_HLR



16.1.1.4 Detailed procedure in the HLR

The following macros are used by this process:

- Receive_Open_Ind, defined in subclause 21.1;
- Check_indication, defined in subclause 21.2;
- Insert_Subs_Data_Framed_HLR, described in subclause 16.4.1;
- Control_Tracing_HLR, described in subclause 21.9;

and the processes Cancel_Location_HLR (see subclause 16.1.2) and Subscriber_Present_HLR (see subclause 16.1.1.7) are invoked.

The location updating process in the HLR is activated by receipt of a MAP_UPDATE_LOCATION indication (see figure 16.1.1/9):

- if there is a parameter problem in the indication, the error Unexpected Data Value is returned in the MAP_UPDATE_LOCATION response (see Check_indication macro defined in subclause 21.2); if the subscriber is not known in the HLR, the error Unknown Subscriber is returned in the response. In either case the process terminates;
- tracing shall be set to deactive in the VLR
- if the VLR address received in the MAP_UPDATE_LOCATION indication differs from the one actually stored against the subscriber, the Cancel_Location_HLR process is started to cancel the subscriber data in the stored VLR (see subclause 16.1.2).

The next action will be to check whether the subscriber is allowed to roam into the PLMN indicated by the VLR Number given in the MAP_UPDATE_LOCATION indication:

- if the subscriber is not allowed to roam into the PLMN, the error Roaming not Allowed with cause PLMN Roaming Not Allowed is returned in the MAP_UPDATE_LOCATION response, and the routing information stored (VLR number, MSC Number, LMSI) is deleted (deregistration);
- otherwise the HLR database will be updated with information received in the indication. The HLR sets the "MS purged" flag to False and checks whether tracing is required for that subscriber. This is handled by the macro Control_Tracing_HLR described in subclause 21.9.

Thereafter, the macro Insert_Subs_Data_Framed_HLR described in subclause 16.4.1 is invoked. The outcome of this macro may be:

- aborted, in which case the process terminates;
- error, in which case the error System Failure is returned in the MAP_UPDATE_LOCATION response and the process terminates;
- OK, indicating successful outcome of downloading the subscriber data to the VLR.

The SUBSCRIBER_PRESENT_HLR process is then started to alert the Short Message Service Centre, if required (see subclause 16.1.7). Additionally, the MAP_FORWARD_CHECK_SS_INDICATION request is sent to inform the subscriber about an uncertain state of his SS-Data if this is needed due to previous HLR restoration (use of this service may be omitted as an HLR operator option).

Finally the HLR number is returned in the MAP_UPDATE_LOCATION response.

In all cases where the HLR sends a MAP_UPDATE_LOCATION response to the VLR, the dialogue towards the VLR is terminated by a MAP_CLOSE request with parameter Release Method indicating Normal Release.

Figure 16.1.1/9 (sheet 1 of 2): Process Update_Location_HLR

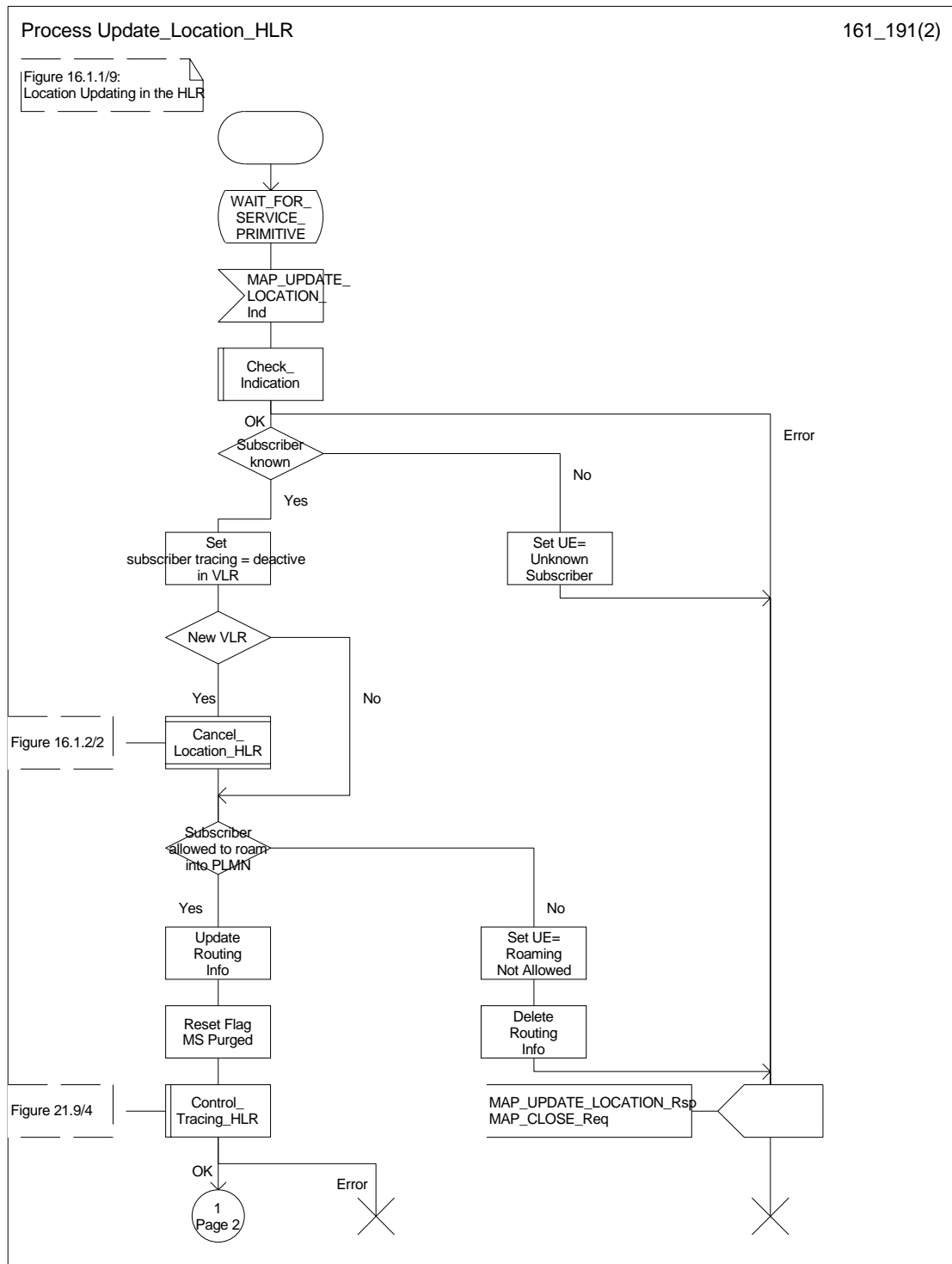
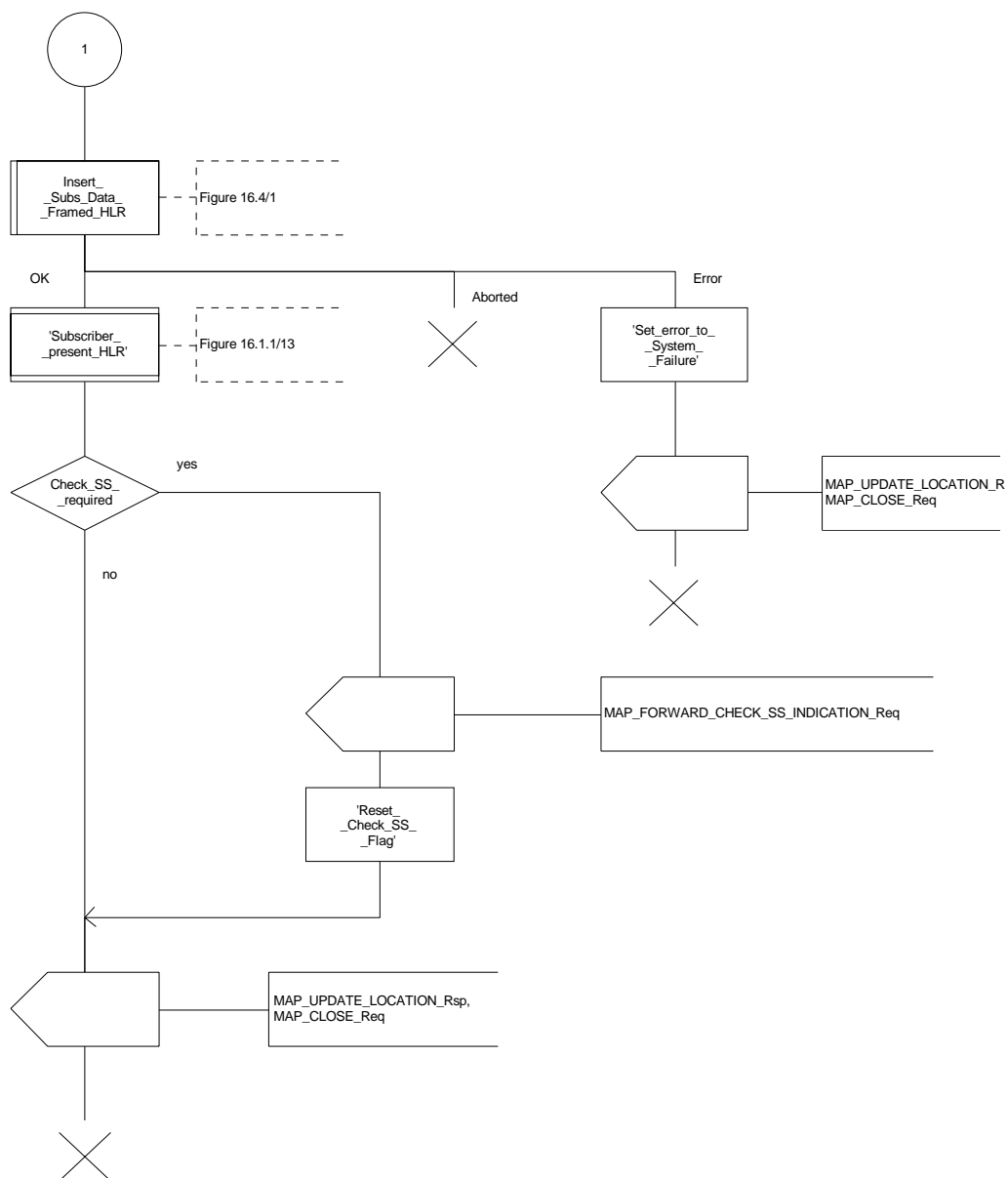


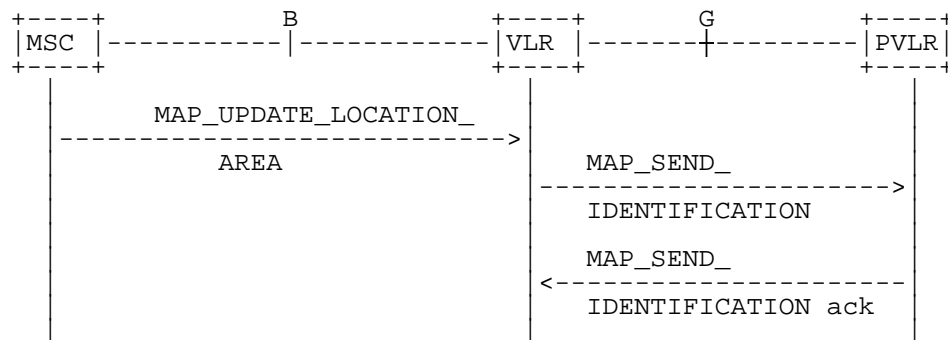
Figure 16.1.1/9 (sheet 2 of 2): Process Update_Location_HLR



16.1.1.5 Send Identification

16.1.1.5.1 General

This service is invoked by a VLR when it receives a MAP_UPDATE_LOCATION_AREA indication containing a LAI indicating that the subscriber was registered in a different VLR (henceforth called the Previous VLR, PVLR). If the identity of the PVLR is derivable for the VLR (usually if both are within the same network), the IMSI and authentication sets are requested from the PVLR (see subclause 16.1.1.3), using the service described in subclause 6.1.4.



NOTE: The service shown in dotted lines indicates the trigger provided by other MAP signalling.

Figure 16.1.1/10: Interface and services for Send Identification

16.1.1.5.2 Detailed procedure in the VLR

The VLR procedure is part of the location area updating process described in subclause 16.1.1.3, see also figure 16.1.1/6 sheet 3.

16.1.1.5.3 Detailed procedure in the PVLR

On receipt of a dialogue request for the Send Identification procedure, (see Receive_Open_Ind macro in subclause 21.1), the PVLR will:

- terminate the procedure in case of parameter problems;
- revert to the MAP version Vr procedure in case the VLR indicated version Vr protocol; or
- continue as below, if the dialogue is accepted.

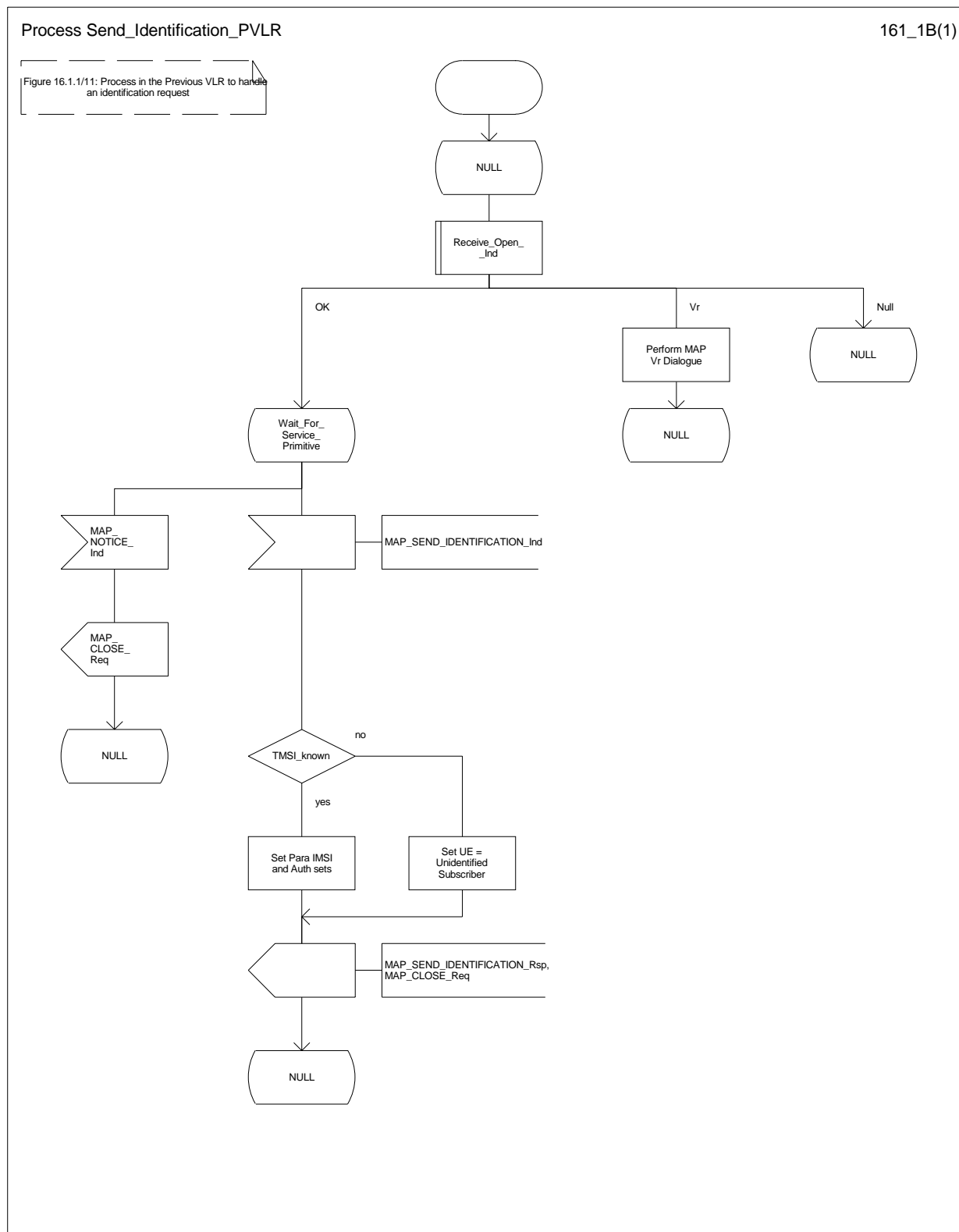
If the PVLR process receives a MAP_NOTICE indication, it terminates the dialogue by sending a MAP_CLOSE request.

If the PVLR process receives a MAP_SEND_IDENTIFICATION indication from the VLR (see figure 16.1.1/11), it checks whether the subscriber identity provided is known:

- if so, the IMSI and - if available - authentication parameters for the subscriber are returned in the MAP_SEND_IDENTIFICATION response;
- if not, the error Unidentified Subscriber is returned in the MAP_SEND_IDENTIFICATION response.

In all cases where the PVLR sends a MAP_SEND_IDENTIFICATION response to the VLR, the dialogue towards the VLR is terminated by a MAP_CLOSE request with parameter Release Method indicating Normal Release.

Figure 16.1.1/11: Process Send_Identification_PVLR



16.1.1.6 The Process Update Location VLR

This process is started by some other MAP user process in case the HLR need to be updated due to previous network failure. It is invoked when the subscriber accesses the network, e.g. for mobile originated call set-up, response to paging or supplementary services handling. Here, location updating consists only of invoking the macro VLR_Update_HLR described above (see subclause 16.1.1.3), which performs HLR updating and downloading of subscriber data.

If updating is successful (OK) the HLR Number is received in the MAP_UPDATE_LOCATION confirm primitive and the process terminates.

If one of the errors Roaming not Allowed or Unknown Subscriber is received instead, all subscriber data are deleted from the VLR before the process terminates.

In case some other error occurs during HLR updating, the process simply terminates. Note, in all error cases the initiating restoration flags in VLR remain false, therefore a new HLR updating attempt will be started later on.

NOTE: This process will be performed independent from the calling process, no coordination is required.

Figure 16.1.1/12: Process UL_VLR

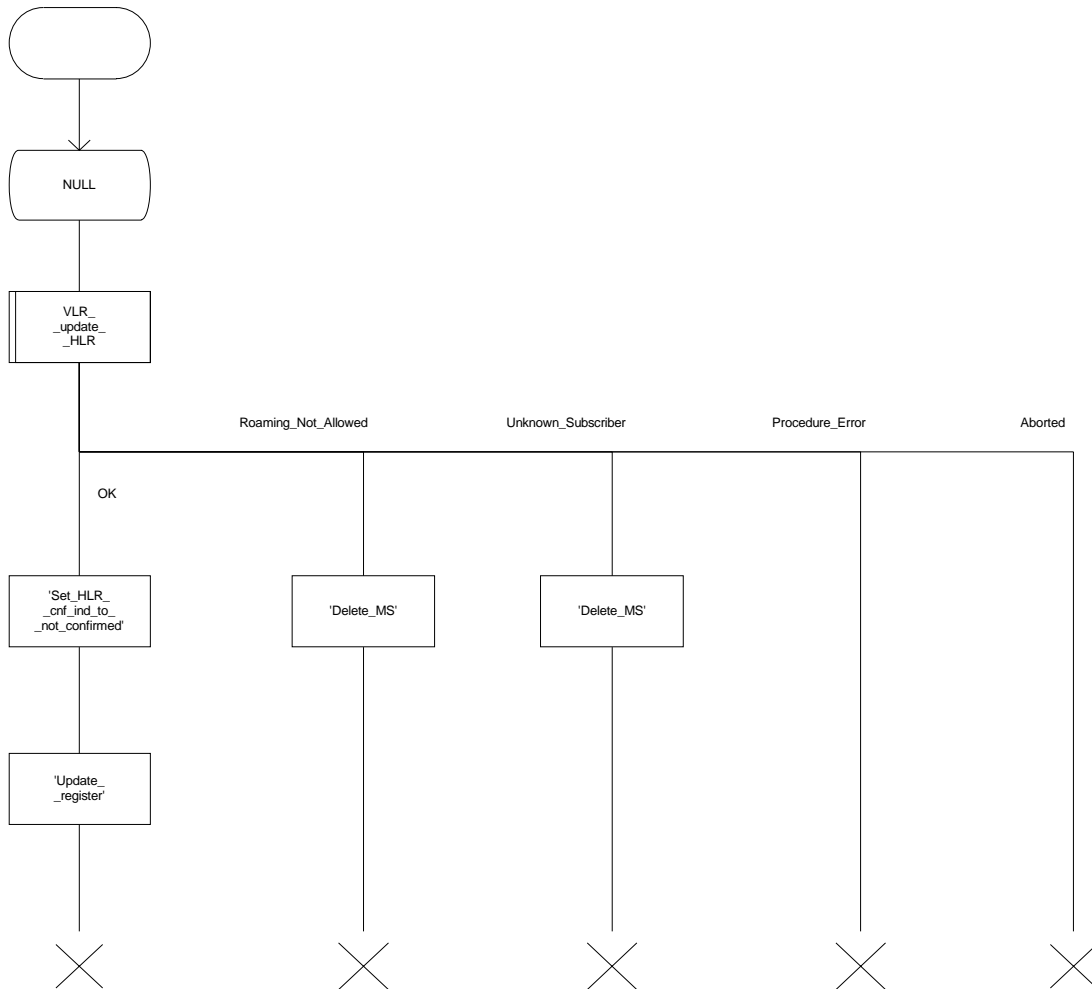
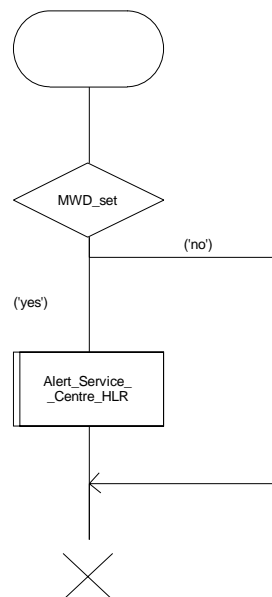


Figure 16.1.1/12:
Location updating initiated
by VLR: VLR process

16.1.1.7 The Process Subscriber Present HLR

The process Subscriber Present HLR is started by the location updating process in HLR to perform actions required for short message alerting. The process checks the Message Waiting Data flag, and if this is set, the macro Alert_Service_Centre_HLR defined in subclause 21.10 is invoked. This macro will alert all service centres from which there are short messages waiting for this subscriber.

Figure 16.1.1/13: Process Subscriber_Present_HLR



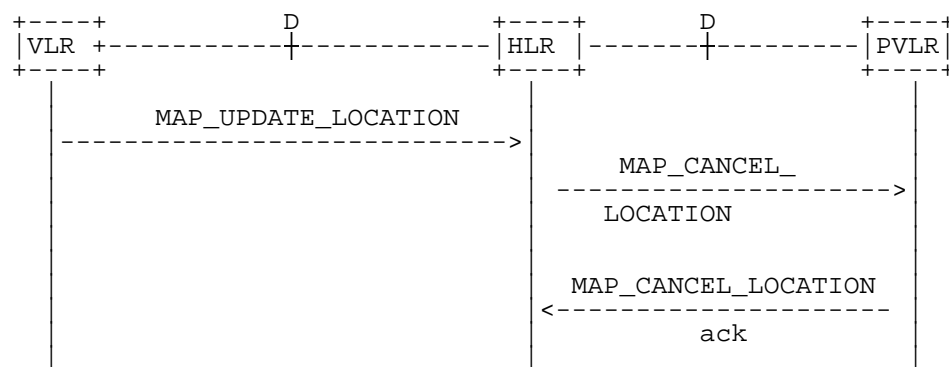
16.1.2 Location Cancellation

16.1.2.1 General

The purpose of this process is to delete a subscriber's record from a previous visitor location register after she has registered with a new visitor location register. The procedure may also be used if the subscriber's record is to be deleted for other operator determined purposes, e.g. withdrawal of subscription, imposition of roaming restrictions or modifications to the subscription which result in roaming restrictions. Location cancellation can be used to enforce location updating including updating of subscriber data in the VLR at the next subscriber access.

In all cases, the process is performed independently of the invoking process (e.g. Location Updating).

The service as described in subclause 6.1.3 is invoked when an HLR receives a MAP_UPDATE_LOCATION indication from a VLR other than that stored in its table for this subscriber. Additionally the service may be invoked by operator intervention. The MAP_CANCEL_LOCATION service is in any case invoked towards the VLR whose identity is contained in the HLR table.



NOTE: The service shown in dotted lines indicates the trigger provided by other MAP signalling.

Figure 16.1.2/1: Interface and services for Location Cancellation

16.1.2.2 Detailed procedure in the HLR

The location cancellation process is started by an external process as stated above. The HLR opens a dialogue with the VLR whose identity is contained in the HLR table (MAP_OPEN request without any user specific parameters), sending the MAP_CANCEL_LOCATION request primitive (see figure 16.1.2/2), containing the parameters:

- IMSI, to identify the subscriber to be deleted from that VLR;
- LMSI, which is included if available in the HLR.

The HLR then waits for the MAP_OPEN confirmation (see macro Receive_Open_Cnf, subclause 21.1), indicating either:

- reject of the dialogue (process terminates);
- reversion to version Vr (process will be performed according to MAP version Vr); or
- dialogue acceptance.

When the VLR accepts the dialogue, it will return a MAP_CANCEL_LOCATION confirmation, containing:

- no parameter, indicating successful outcome of the procedure;
- a user error, provider error or a data error indicating unsuccessful outcome of the procedure.

In case of unsuccessful outcome or if a MAP_P_ABORT indication has been received, the HLR may repeat the MAP_CANCEL_LOCATION request later, where the number of repeat attempts and time in between are HLR operator options, depending on the error returned by the VLR.

16.1.2.3 Detailed procedure in the VLR

Opening of the dialogue is described in the macro Receive_Open_Ind in subclause 21.1, with outcomes:

- reversion to version Vr procedure;
- procedure termination; or
- dialogue acceptance, with processing as below.

If the VLR process receives a MAP_NOTICE indication, it terminates the dialogue by sending a MAP_CLOSE request.

If the VLR process receives a MAP_CANCEL_LOCATION indication from the HLR (see figure 16.1.2/3), the parameters are checked first (macro Check_Indication, see subclause 21.2). In case of parameter problems the appropriate error is sent in the MAP_CANCEL_LOCATION response.

If the MAP_CANCEL_LOCATION indication contains both the IMSI and the LMSI, the VLR checks whether the stored IMSI matches the received IMSI. If it does not, the VLR attempts to process the request using the IMSI received from the HLR to define the subscriber record to be deleted.

Thereafter the VLR checks whether the subscriber identity provided is known in the VLR:

- if so, the data of the subscriber are deleted from VLR table and a MAP_CANCEL_LOCATION response is returned without any parameters;
- if not, location cancellation is regarded as being successful, too, and the MAP_CANCEL_LOCATION response is returned without any parameters.

In either case, after sending the MAP_CANCEL_LOCATION response the VLR process releases any TMSI which may be associated with the IMSI of the subscriber, terminates the dialogue (MAP_CLOSE with Release Method Normal Release) and returns to the idle state.

Figure 16.1.2/2: Process Cancel_Location_HLR

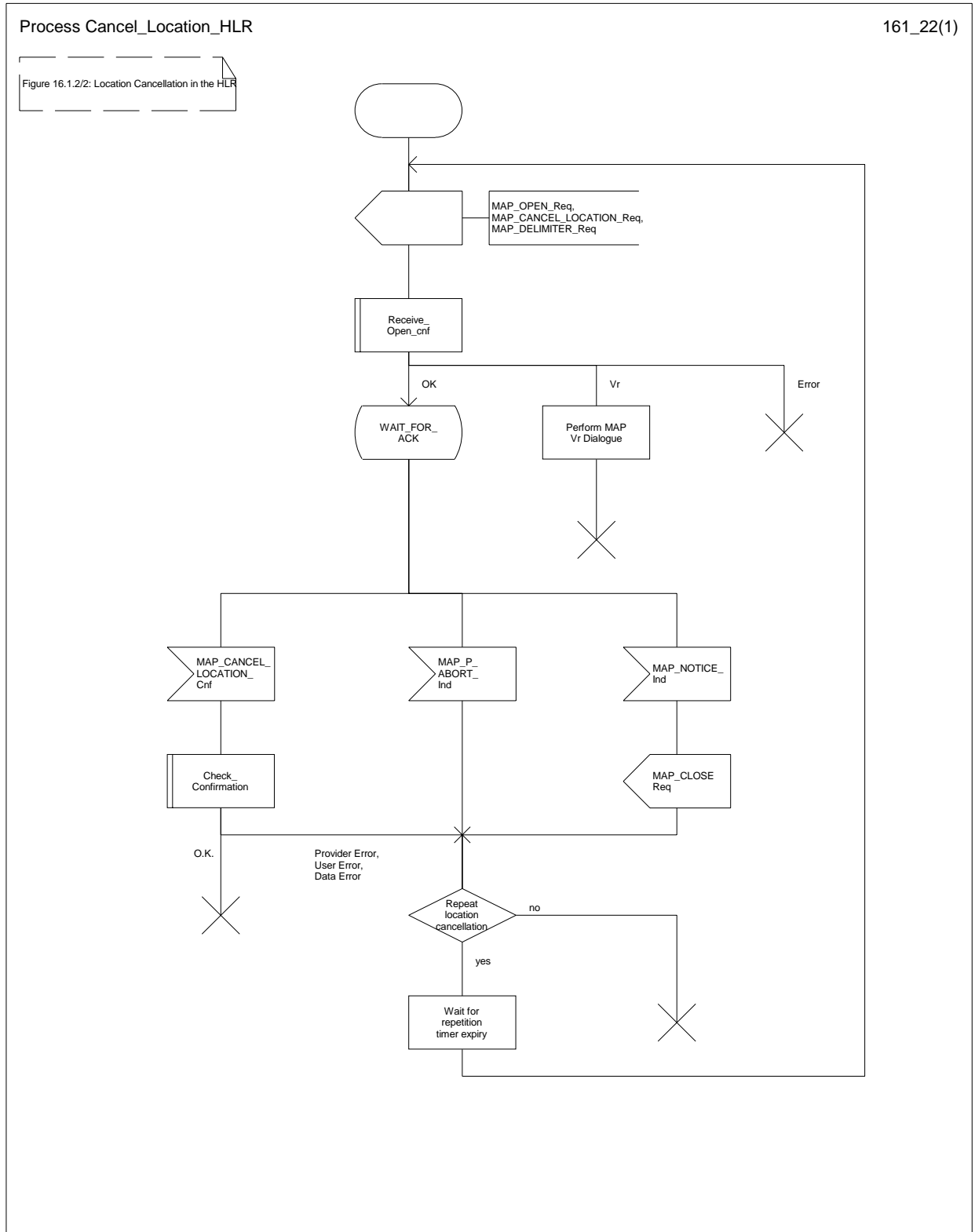
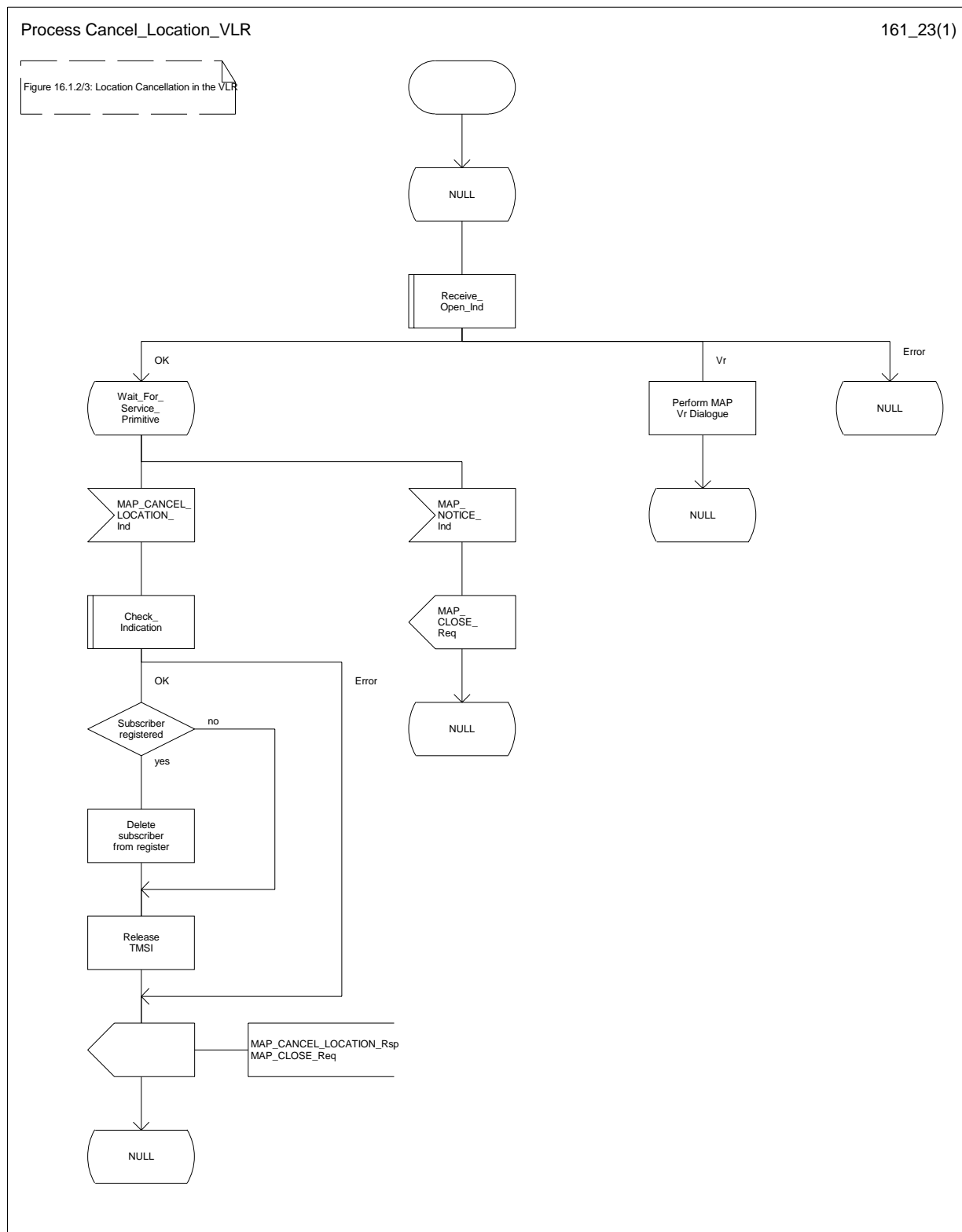


Figure 16.1.2/3: Process Cancel_Location_VLR

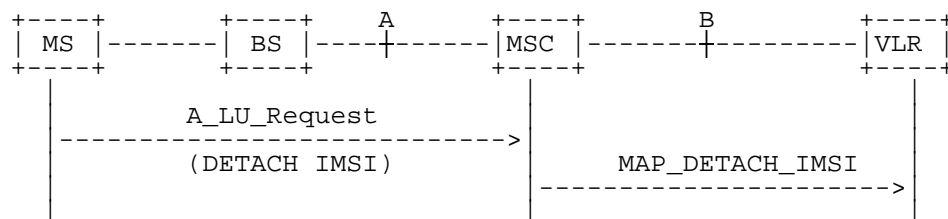


16.1.3 Detach IMSI

16.1.3.1 General

On receipt of an A_LU_REQUEST (DETACH IMSI) indication from the radio interface this procedure invokes the MAP_DETACH_IMSI service described in subclause 6.1.5 in order to inform the visitor location register that a subscriber is no longer reachable (see figure 16.1.3/1), e.g. due to switched off station. This information is used by the VLR to reject mobile terminating calls or short messages without sending page messages on the radio path. The service is unconfirmed as it is likely that the MS is switched off before receiving a confirmation.

The detach IMSI feature is optional for the network operator. The MS is informed by the network whether detach IMSI is to be used or not.



NOTE: The service shown in dotted lines indicates the trigger provided by the radio interface (see GSM 09.10).

Figure 16.1.3/1: Interface and services for MAP_DETACH_IMSI

16.1.3.2 Detailed procedure in the MSC

The MAP_DETACH_IMSI service is invoked by the MSC when receiving an A_LU_Request (DETACH IMSI) for a subscriber (see figure 16.1.3/2).

The MSC will open the dialogue to the VLR with a MAP_OPEN request containing no user specific parameters. The MAP_DETACH_IMSI request will contain the following parameter received from the radio side (for the mapping see GSM 09.10):

- Subscriber Id, being either a TMSI or an IMSI.

The MSC then waits for the MAP_OPEN confirmation (see macro Receive_Open_Cnf, subclause 21.1), indicating either:

- reject of dialogue (process terminates);
- reversion to version Vr(process terminates); or
- dialogue acceptance.

Thereafter, the dialogue is terminated locally by the MSC (MAP_CLOSE request with Release Method Prearranged End).

16.1.3.3 Detailed procedure in the VLR

When the VLR receives a MAP_DETACH_IMSI indication (see figure 16.1.3/3), it first checks the indication data (macro Check_Indication, see subclause 21.2). Thereafter it is checked whether the subscriber is known:

- if the subscriber is unknown the VLR ignores the indication;
- if the subscriber is known in the VLR, the IMSI detached flag is set.

The VLR process will terminate the dialogue locally (MAP_CLOSE request with Release Method Prearranged End).

Figure 16.1.3/2: Process Detach_IMSI_MSC

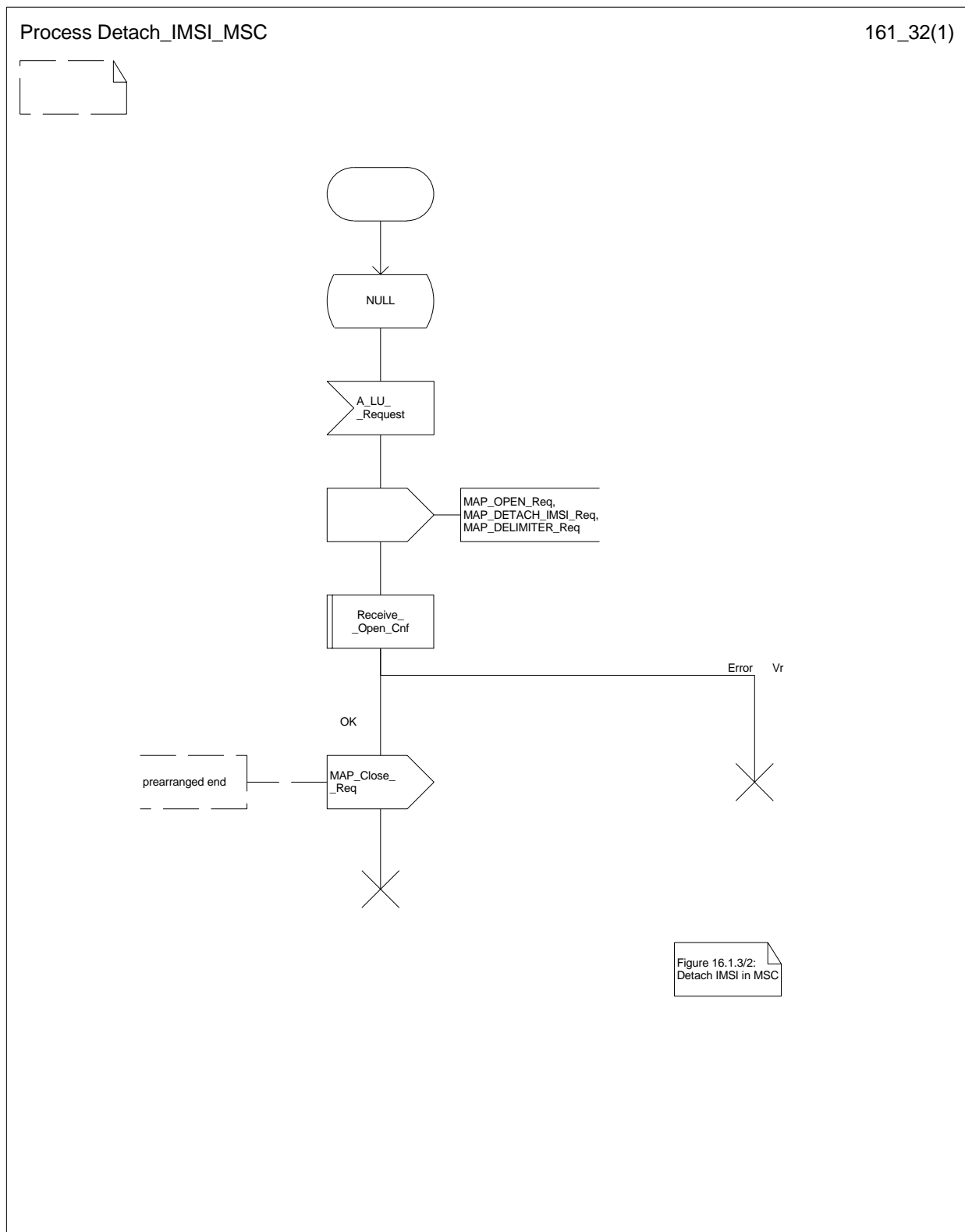


Figure 16.1.3/3: Process Detach_IMSI_VLR

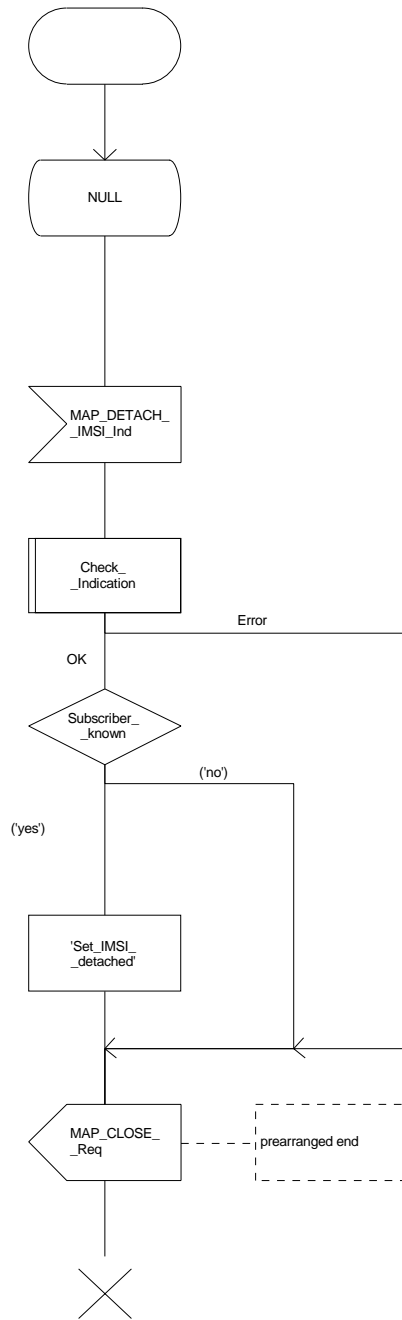


Figure 16.1.3/3:
Detach IMSI in VLR

16.1.4 Purge MS

16.1.4.1 General

When the VLR receives an indication on the O&M interface that the MS record is to be purged (either because of administrative action or because the MS has been inactive for an extended period), this procedure invokes the MAP_PURGE_MS service described in subclause 6.1.6 to request the HLR to set the "MS purged" flag for the MS so that any request for routing information for a mobile terminated call or a mobile terminated short message will be treated as if the MS is not reachable. The message flow is shown in figure 16.1.4/1.

It is optional for the network operator to delete MS records from the VLR, but if the option is used the VLR shall notify the HLR when a record has been deleted.

The O&M process in the VLR must ensure that during the MS purging procedure any other attempt to access the MS record is blocked, to maintain consistency of data.

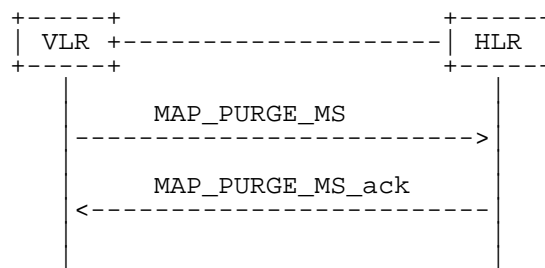


Figure 16.1.4/1: Interface and services for MAP_PURGE_MS

16.1.4.2 Detailed procedure in the VLR

When the VLR receives an indication from O&M that an MS record is to be purged, it invokes the MAP_PURGE_MS service (see figure 16.1.4/2).

The VLR opens the dialogue to the HLR with a MAP_OPEN request containing no user specific parameters. The MAP_PURGE_MS request contains the IMSI of the MS which is to be purged and the VLR number.

The VLR then waits for the MAP_OPEN confirmation (see macro Receive_Open_Cnf, subclause 21.1), indicating one of:

- rejection of the dialogue (process terminates);
- reversion to version one (process terminates);
- dialogue acceptance.

If the HLR accepts the dialogue it returns a MAP_PURGE_MS confirmation, containing no parameter, indicating successful outcome of the procedure.

If a MAP_PURGE_MS confirmation containing a provider error, data error or user error, or a MAP_P_ABORT, MAP_NOTICE or premature MAP_CLOSE indication, has been received, the failure is reported to the O&M interface. Successful outcome of the procedure leads to deletion of the subscriber data and freezing of the TMSI, and is reported to the O&M interface.

16.1.4.3 Detailed procedure in the HLR

Opening of the dialogue is described in the macro Receive_Open_Ind in subclause 21.1. The possible outcomes are:

- termination of the procedure if the AC indicates a version 1 dialogue, as this procedure is defined only for version 2;
- termination of the procedure if there is an error;
- dialogue acceptance, in which case the procedure is as described below.

If the HLR receives a MAP_NOTICE indication, it terminates the dialogue by sending a MAP_CLOSE request.

If the HLR receives a MAP_PURGE_MS indication (see figure 16.1.4/3), it first checks the indication data (macro Check_Indication, see subclause 21.2). If there is a parameter error the HLR terminates the dialogue by sending a MAP_CLOSE request (local termination). If there is no parameter error the HLR then checks whether the subscriber is known.

- if the subscriber is unknown, the HLR reports an error to the O&M interface, and terminates the dialogue by sending a MAP_CLOSE request (local termination);
- if the subscriber is known, the HLR checks whether the purging notification came from the VLR where the MS was last registered:
 - if the received VLR number and the stored VLR number match, the HLR sets the "MS purged" flag for the subscriber and sends a MAP_PURGE_MS response containing an empty result to indicate successful outcome;
 - if the received VLR number and the stored VLR number do not match, the HLR sends a MAP_PURGE_MS response containing an empty result to indicate successful outcome. Since the MS is known by the HLR to be in a different VLR area, it is not appropriate to block mobile terminated calls or short messages to the MS, but the VLR which initiated the purging procedure can safely purge its record for the MS.

In either cases of successful termination the HLR terminates the dialogue by sending a MAP_CLOSE request.

Figure 16.1.4/2: Process Purge_MS_VLR

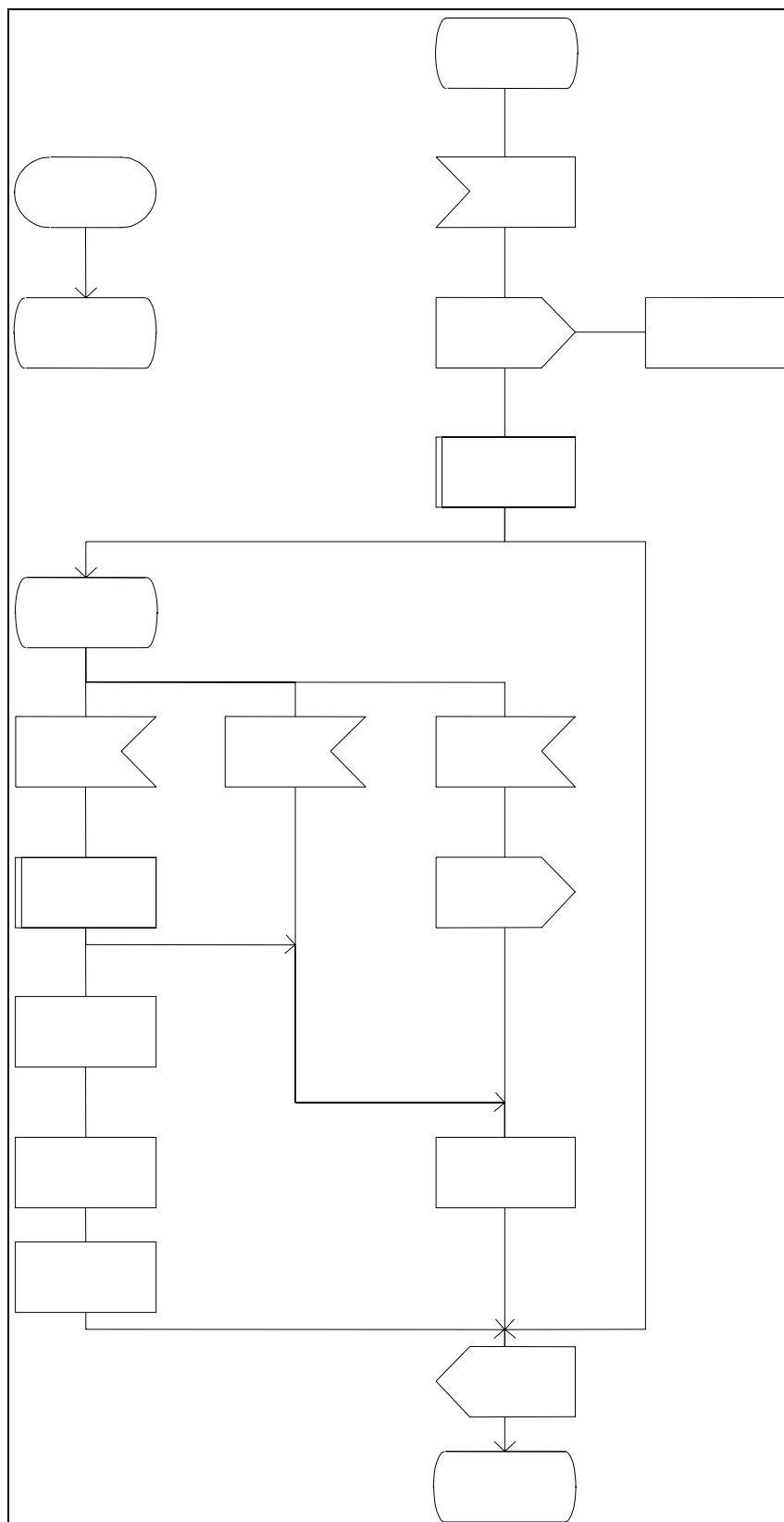
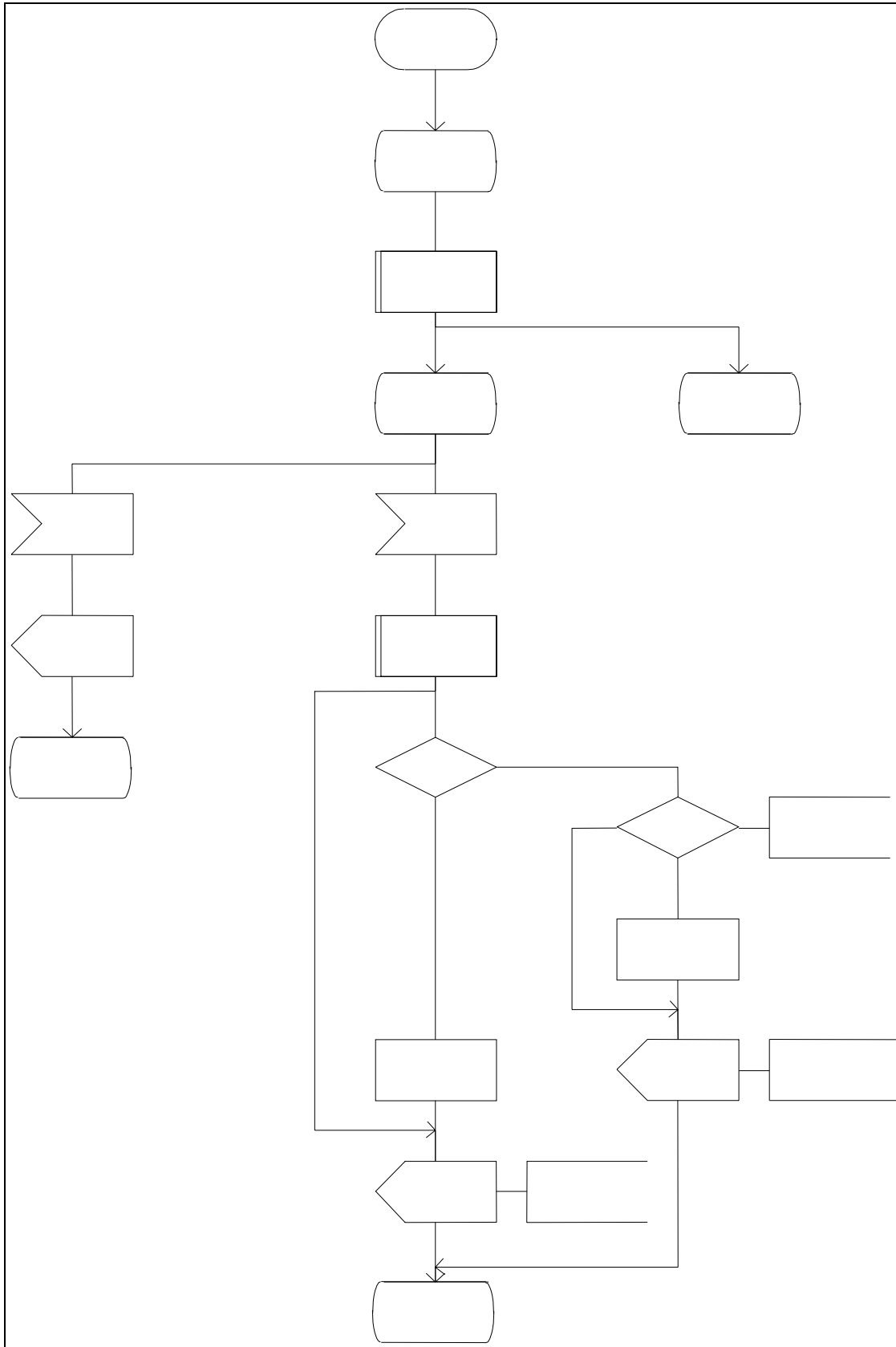


Figure 16.1.4/3: Process Purge_MS_HLR



16.2 Handover procedure

16.2.1 General

The handover between different MSCs is called Inter-MSC handover. The interfaces involved for Inter-MSC handover are shown in figure 16.2/1. Following two Inter-MSC handover procedures apply:

1) Basic Inter-MSC handover:

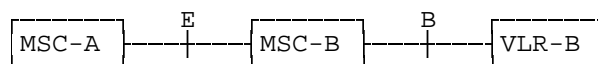
The call is handed over from the controlling MSC, called MSC-A to another MSC, called MSC-B (figure 16.2/1a).

Figure 16.2/2 shows a successful handover between MSC-A and MSC-B including a request for handover number allocation by MSC-B to VLR-B.

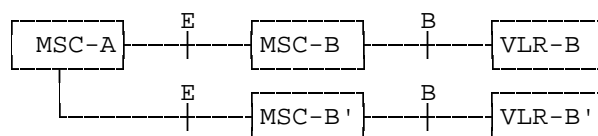
2) Subsequent Inter-MSC handover:

After the call has been handed over from MSC-A to MSC-B, a handover to either MSC-A (figure 16.2/1a) or to a third MSC (MSC-B') (figure 16.2/1b) is necessary in order to continue the connection.

Figure 16.2/3 shows a successful subsequent handover.



a) Basic handover procedure MSC-A to MSC-B and subsequent handover procedure MSC-B to MSC-A.



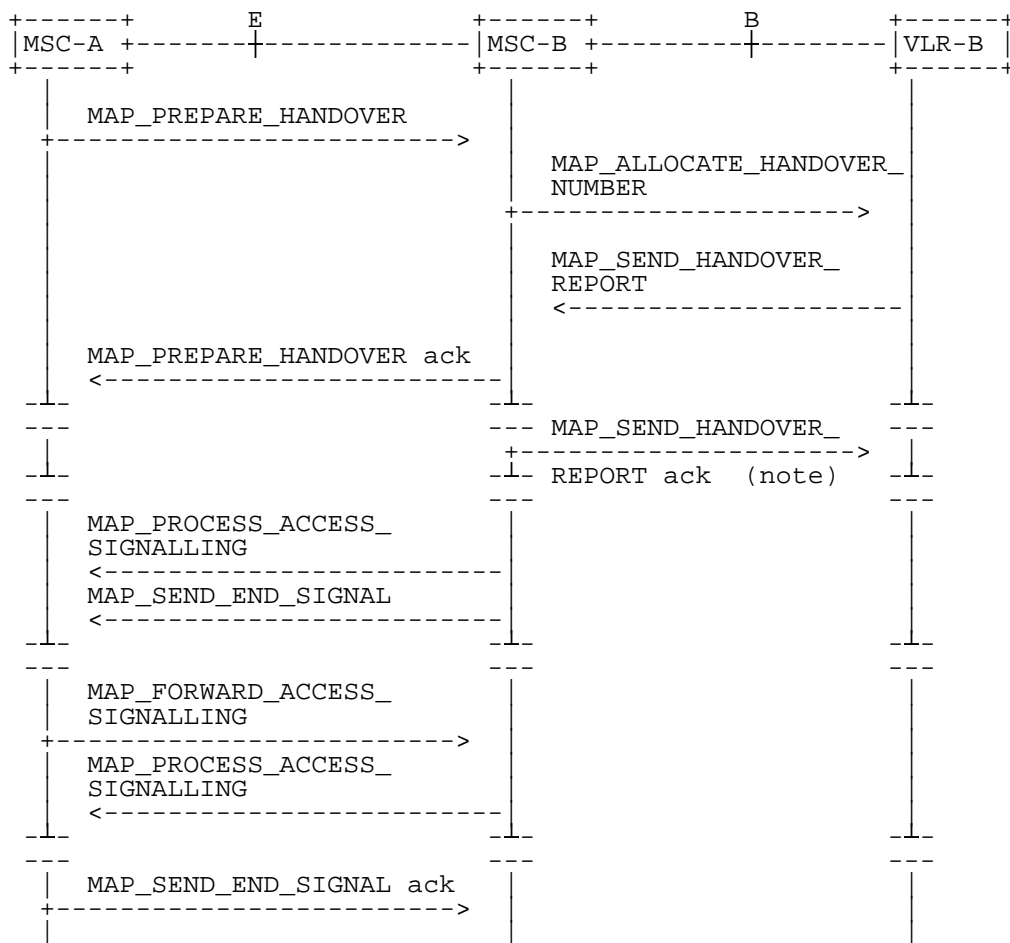
b) Subsequent handover procedure MSC-B to MSC-B'.

Figure 16.2/1: Interface structure for handover

The MAP handover procedures achieve the functionality required to set up an MSC-MSC dialogue, to optionally allocate a handover number and to transport BSSAP messages.

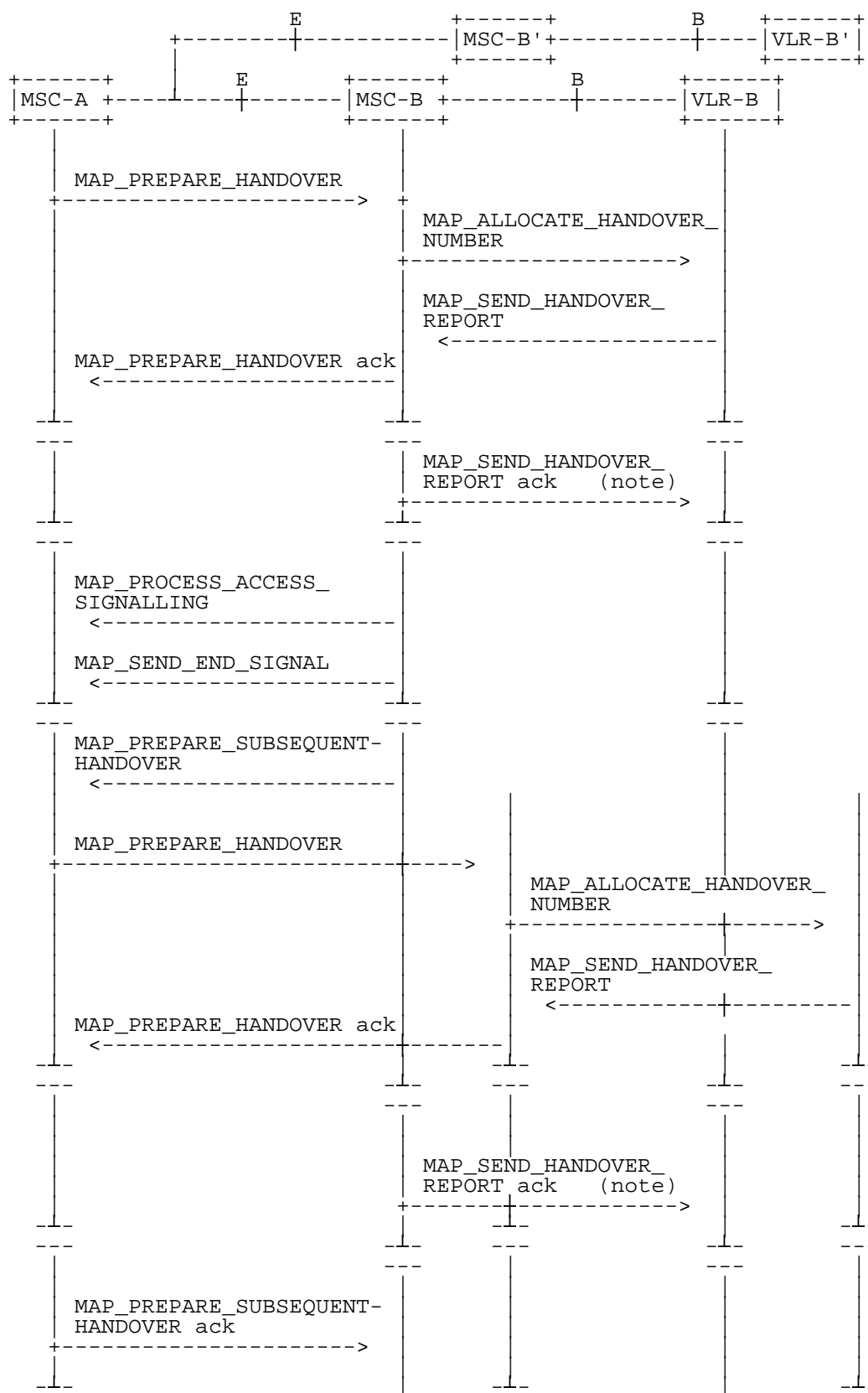
The transported BSSAP messages are controlled and handled by the Handover Control Application in the MSCs. This information will be transparent to the MAP protocol. If the MSC receives via the MAP protocol BSSAP messages, this information will be forwarded to the Handover Control Application (shown in the handover SDL diagrams with the internal HO_CA signalling, it is an internal process in the MSC) and vice versa if the Handover Control Application requires the sending of BSSAP messages via the MAP protocol.

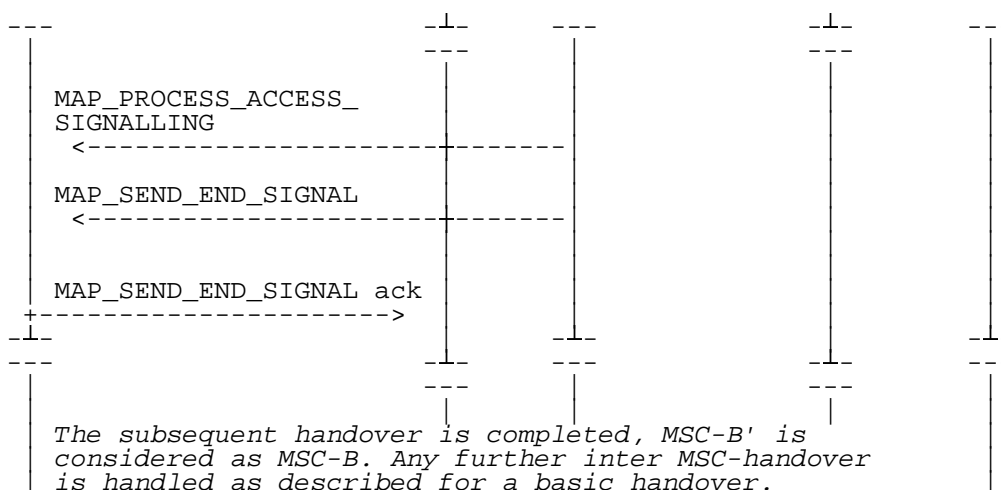
For detailed interworking between the A-interface and MAP procedures, see GSM 03.09 and GSM 09.10.



NOTE: This can be sent at any time after the connection between MSC-A and MSC-B is established.

Figure 16.2/2: Example of a successful basic handover procedure to MSC-B





NOTE: This can be sent at any time after the connection between MSC-A and MSC-B is established

Figure 16.2/3: Example of a handover towards a third MSC

16.2.2 Handover procedure in MSC-A

This subclause describes the handover procedure in MSC-A, including the request for a basic handover to another MSC (MSC-B), subsequent handover to a third MSC (MSC-B') or back to the controlling MSC (MSC-A).

16.2.2.1 Basic handover

When MSC-A has decided that a call has to be handed over to MSC-B, the Handover Control Application in MSC-A requests the MAP application to initiate the MAP_PREPARE_HANDOVER request to MSC-B.

MSC-A opens the dialogue to MSC-B with a MAP_OPEN request containing no user specific parameters and sends a MAP_PREPARE_HANDOVER request. This request may optionally contain an indication that a handover number allocation is not required, targetCellId, for compatibility reasons, and all information required by MSC-B to allocate the necessary radio resources.

If MSC-B accepts the dialogue, it returns a MAP_PREPARE_HANDOVER confirmation containing a handover number, unless the request has included the HO-NumberNotRequired parameter, and BSSAP information which is forwarded to and handled by the Handover Control Application in MSC-A.

Optionally MSC-A can receive, after a MAP_PREPARE_HANDOVER confirmation, a MAP_PROCESS_ACCESS_SIGNALLING indication containing BSSAP information.

When the connection has been established between the MS and MSC-B, MSC-A will be informed by a MAP_SEND_END_SIGNAL indication.

When MSC-A wants to clear the connection with BSS-B, an indication from the Handover Control Application is received in the Map Application to send the MAP_SEND_END-SIGNAL response to MSC-B to close the MAP dialogue.

MSC-A may abort the handover procedure at any time (e.g. if the call is cleared).

16.2.2.2 Handling of access signalling

If required, the Handover Control Application in MSC-A requests the MAP application to invoke the MAP_FORWARD_ACCESS_SIGNALLING request containing the information to be transferred to the A-interface of MSC-B (e.g. call control information).

MAP_FORWARD_ACCESS_SIGNALLING is a non-confirmed service.

MSC-B will then forward the required information to the Handover Control Application. The MAP_FORWARD_ACCESS_SIGNALLING is composed in such a way that the information can be passed transparently to the A-interface for call control and mobility management information. Any response received in MSC-B from the A-interface that should be brought to MSC-A will require a new independent request from the Handover Control Application in MSC-B to MSC-A by invoking a MAP_PROCESS_ACCESS_SIGNALLING request.

16.2.2.3 Other procedures in stable handover situation

During a call and after handover, a number of procedures between MSC-A and BSS-B controlled by or reported to MSC-A may be initiated in both directions by invoking a MAP_FORWARD_ACCESS_SIGNALLING request and reception of a MAP_PROCESS_ACCESS_SIGNALLING indication.

16.2.2.4 Subsequent handover

When MSC-A receives a MAP_PREPARE_SUBSEQUENT_HANOVER request, it will start the procedure of handing the call over to a third MSC (MSC-B'), or back to the controlling MSC (MSC-A). If the new handover procedure towards MSC-B' or MSC-A is successful, the handover control application in MSC-A will request the release of the dialogue towards MSC-B by sending the MAP_SEND_END_SIGNAL confirmation.

16.2.2.5 SDL Diagrams

The SDL diagrams on the following pages describe the user processes in MSC-A for the procedures described in this subclause.

The services used are defined in subclause 6.4.

NOTE: The message primitives HO_CA_MESSAGE used in the SDL-Diagrams are used to show the internal co-ordination between the MAP application and the Handover Control Application. For a detailed description of the co-ordination between the applications for the handover procedure, see GSM 03.09.

Note that in case of reception of errors from the MSCs (see the Handover error handling macro), the MAP user reports them to the Handover Control Application and does not take any action except in cases explicitly mentioned in the SDL diagrams.

Figure 16.2.2/1 (sheet 1 of 12): Process MSC_A_HO

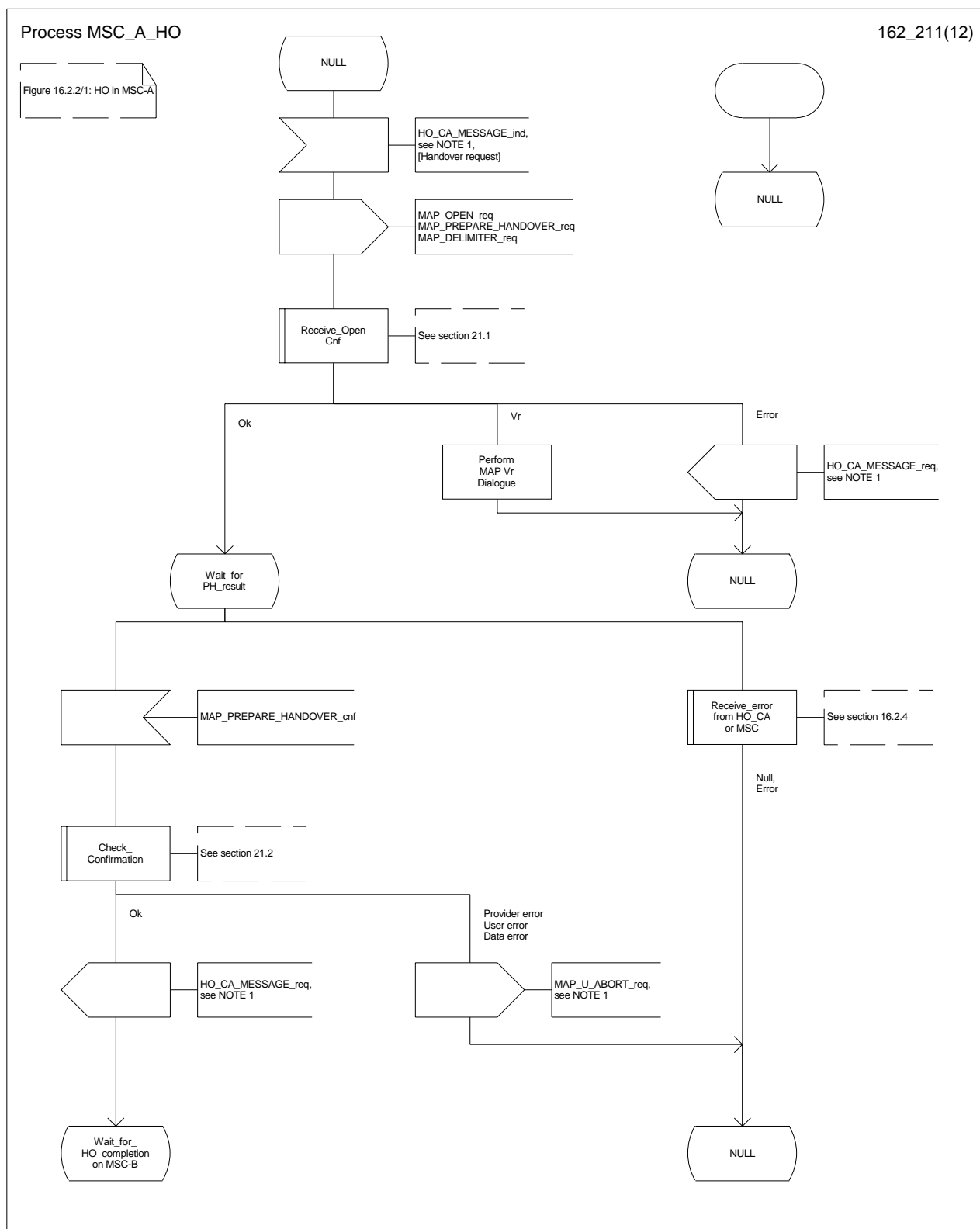


Figure 16.2.2/1 (sheet 2 of 12): Process MSC_A_HO

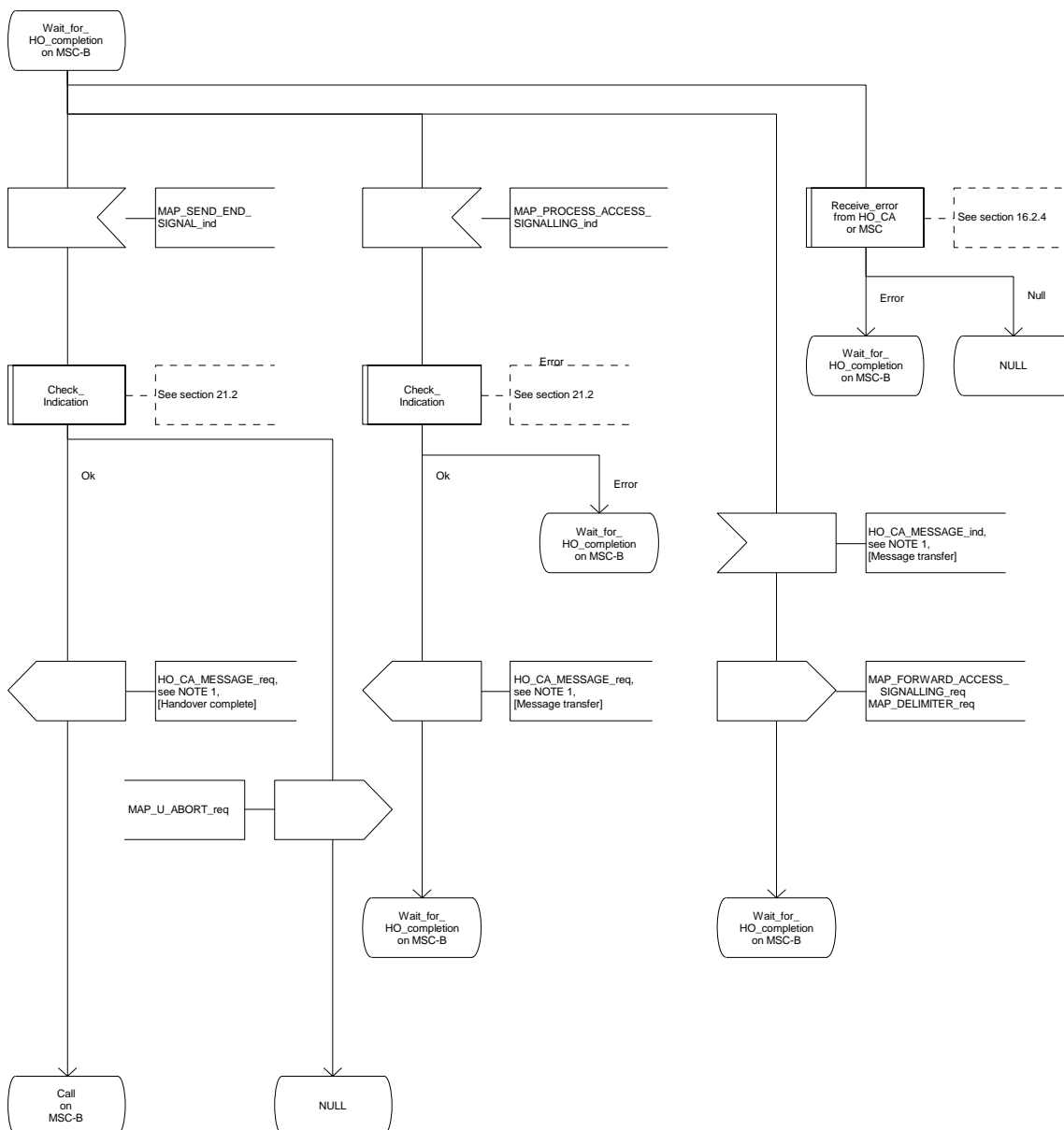


Figure 16.2.2/1 (sheet 3 of 12): Process MSC_A_HO

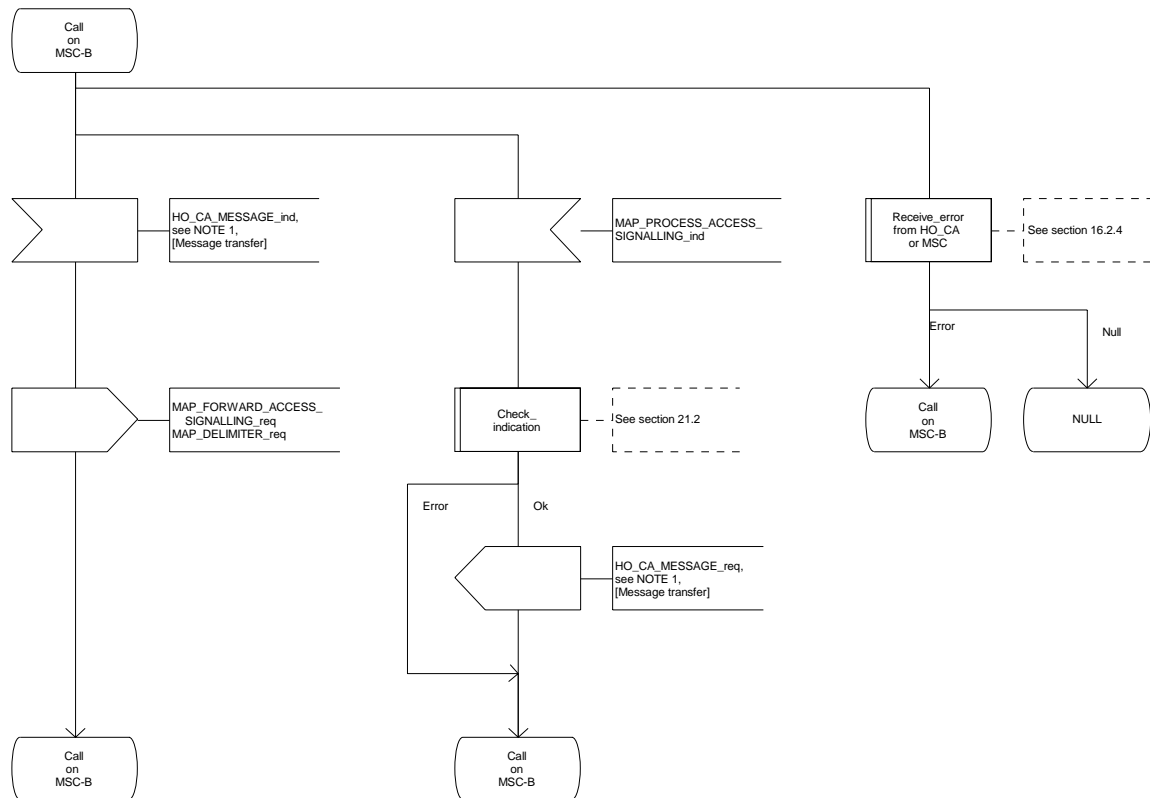


Figure 16.2.2/1 (sheet 4 of 12): Process MSC_A_HO

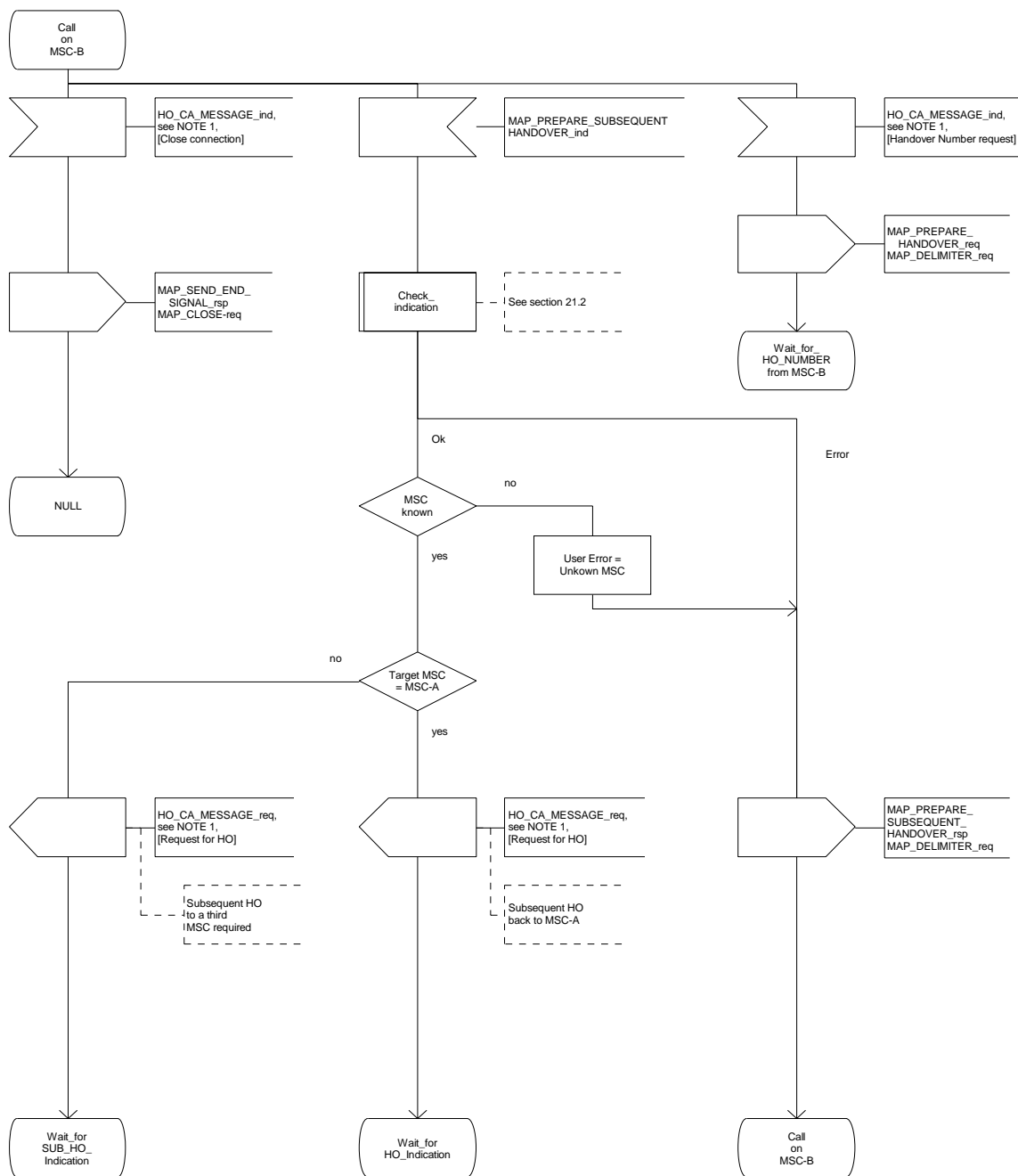


Figure 16.2.2/1 (sheet 5 of 12): Process MSC_A_HO

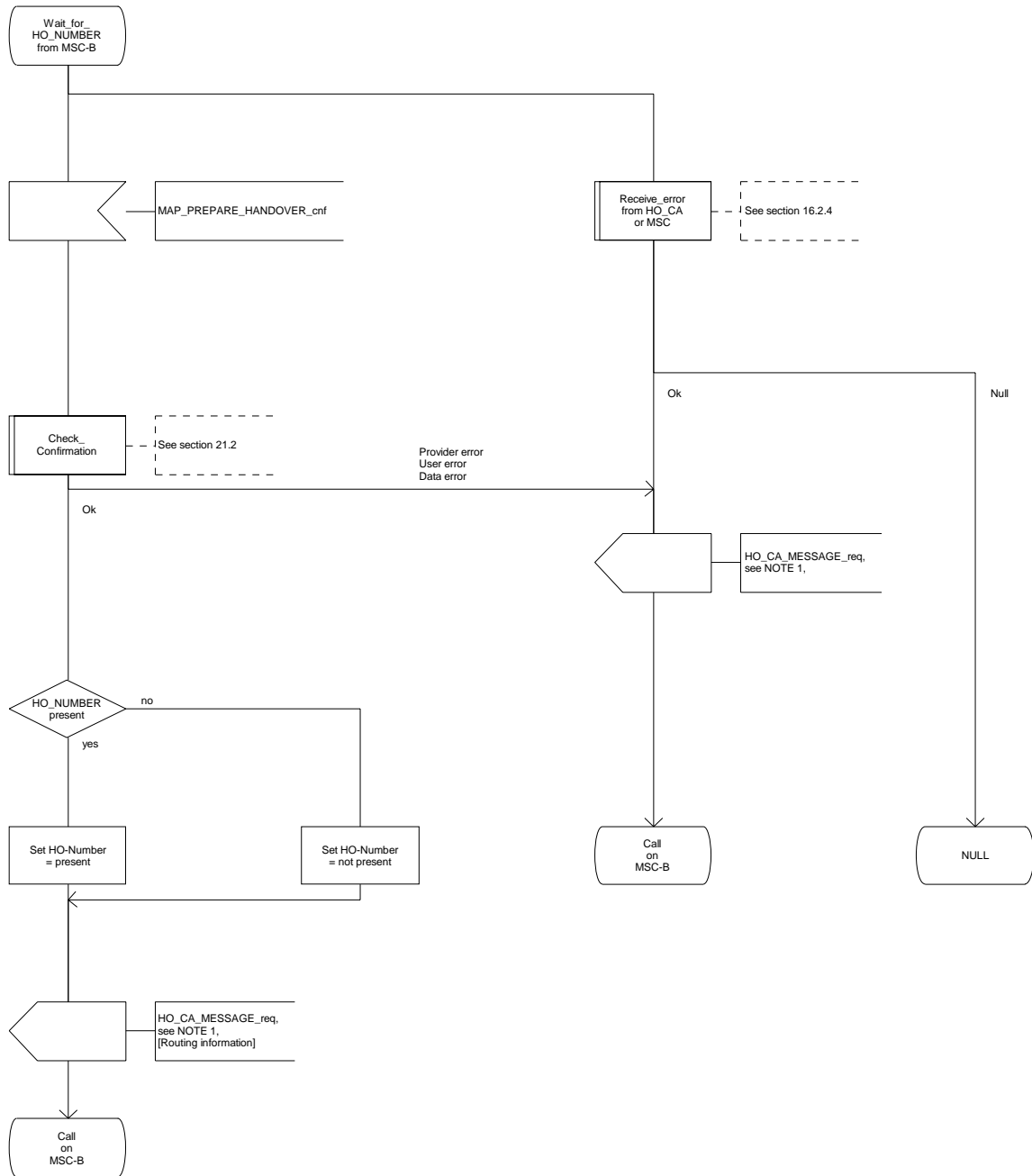


Figure 16.2.2/1 (sheet 6 of 12): Process MSC_A_HO

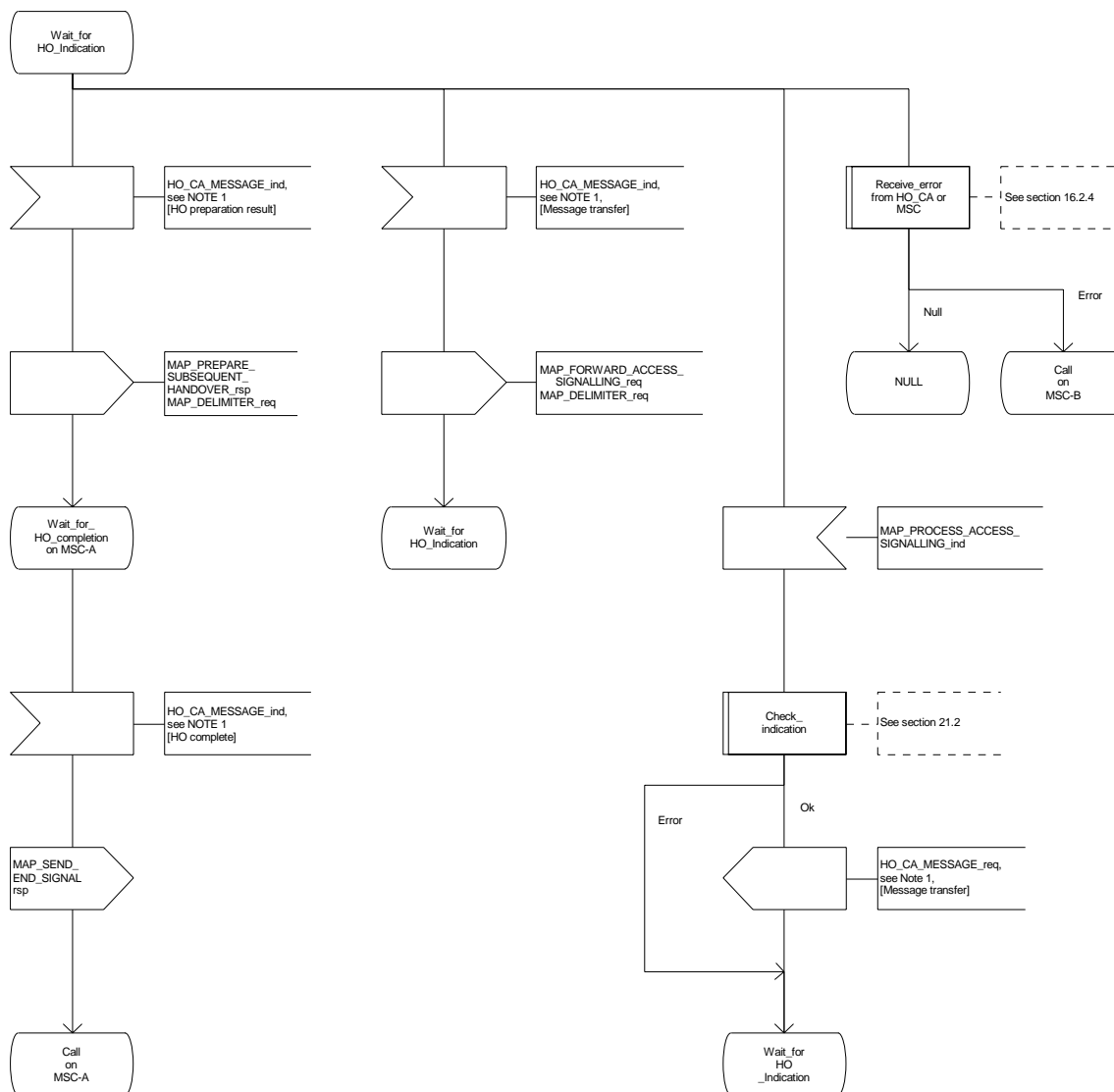


Figure 16.2.2/1 (sheet 7 of 12): Process MSC_A_HO

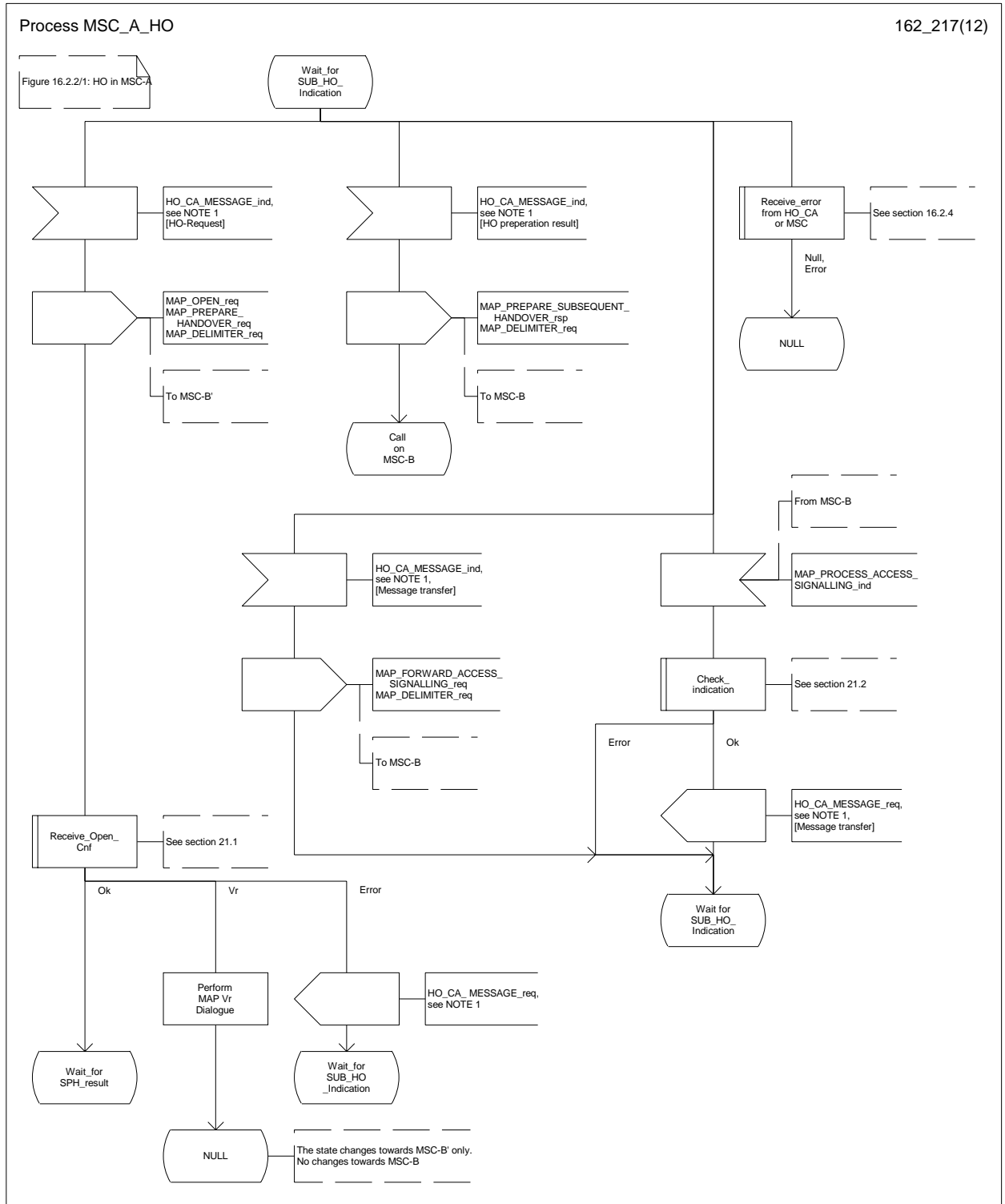


Figure 16.2.2/1 (sheet 8 of 12): Process MSC_A_HO

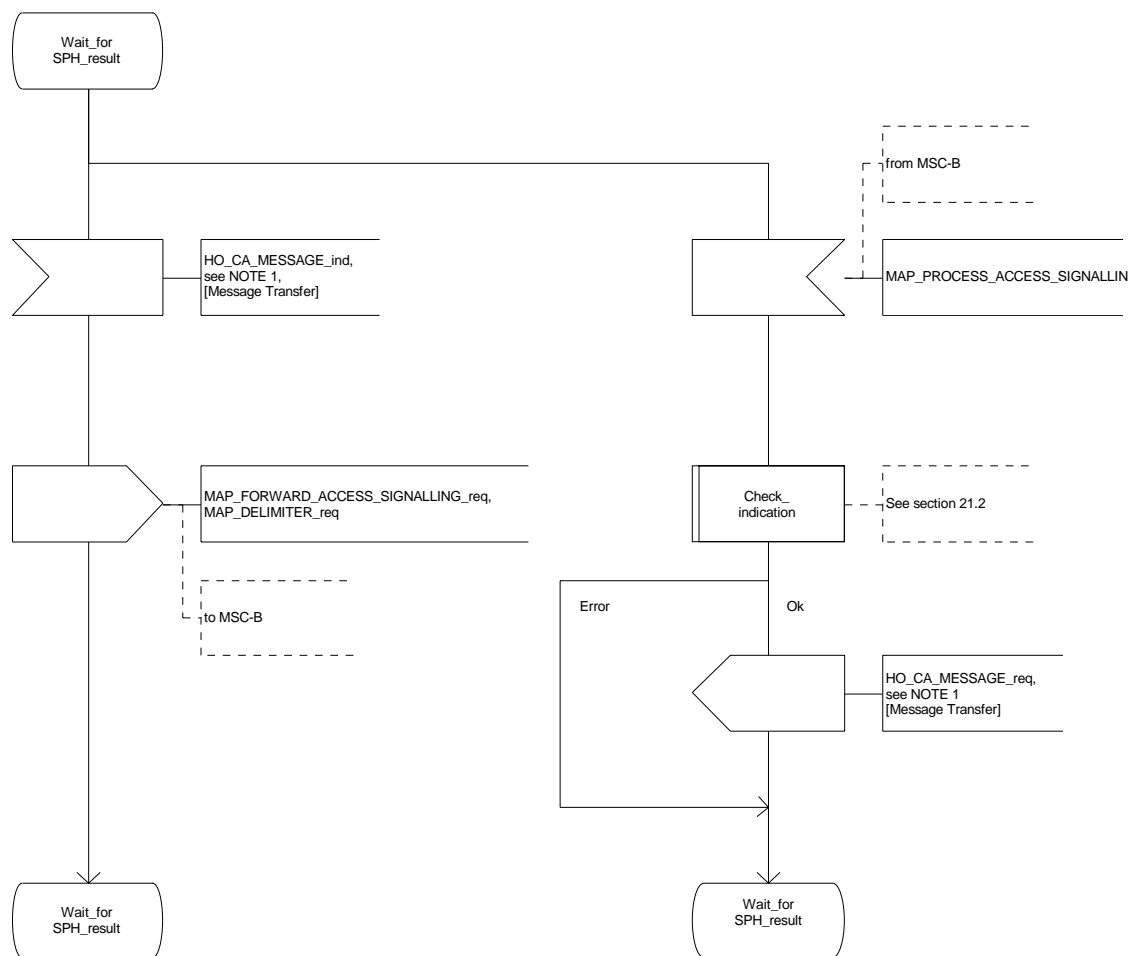


Figure 16.2.2/1 (sheet 9 of 12): Process MSC_A_HO

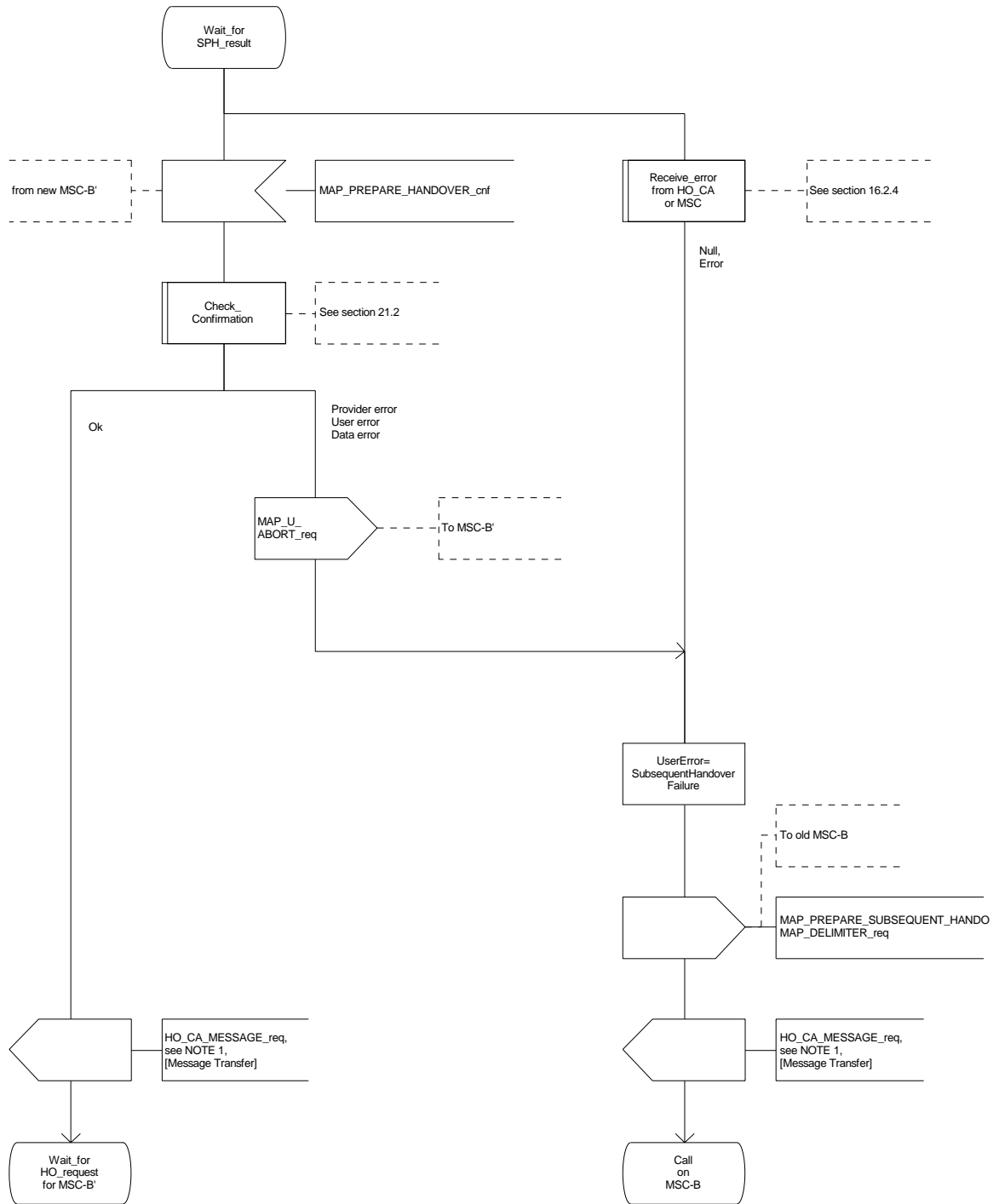


Figure 16.2.2/1 (sheet 10 of 12): Process MSC_A_HO

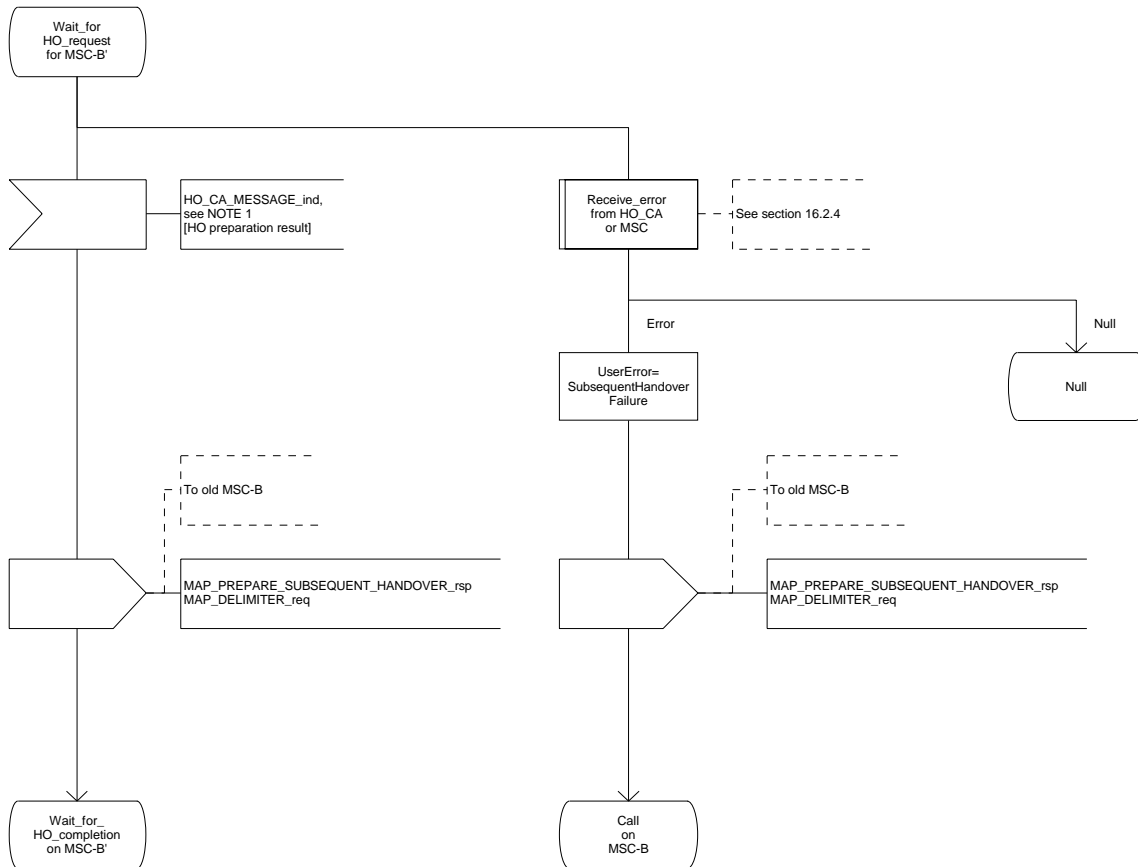


Figure 16.2.2/1 (sheet 11 of 12): Process MSC_A_HO

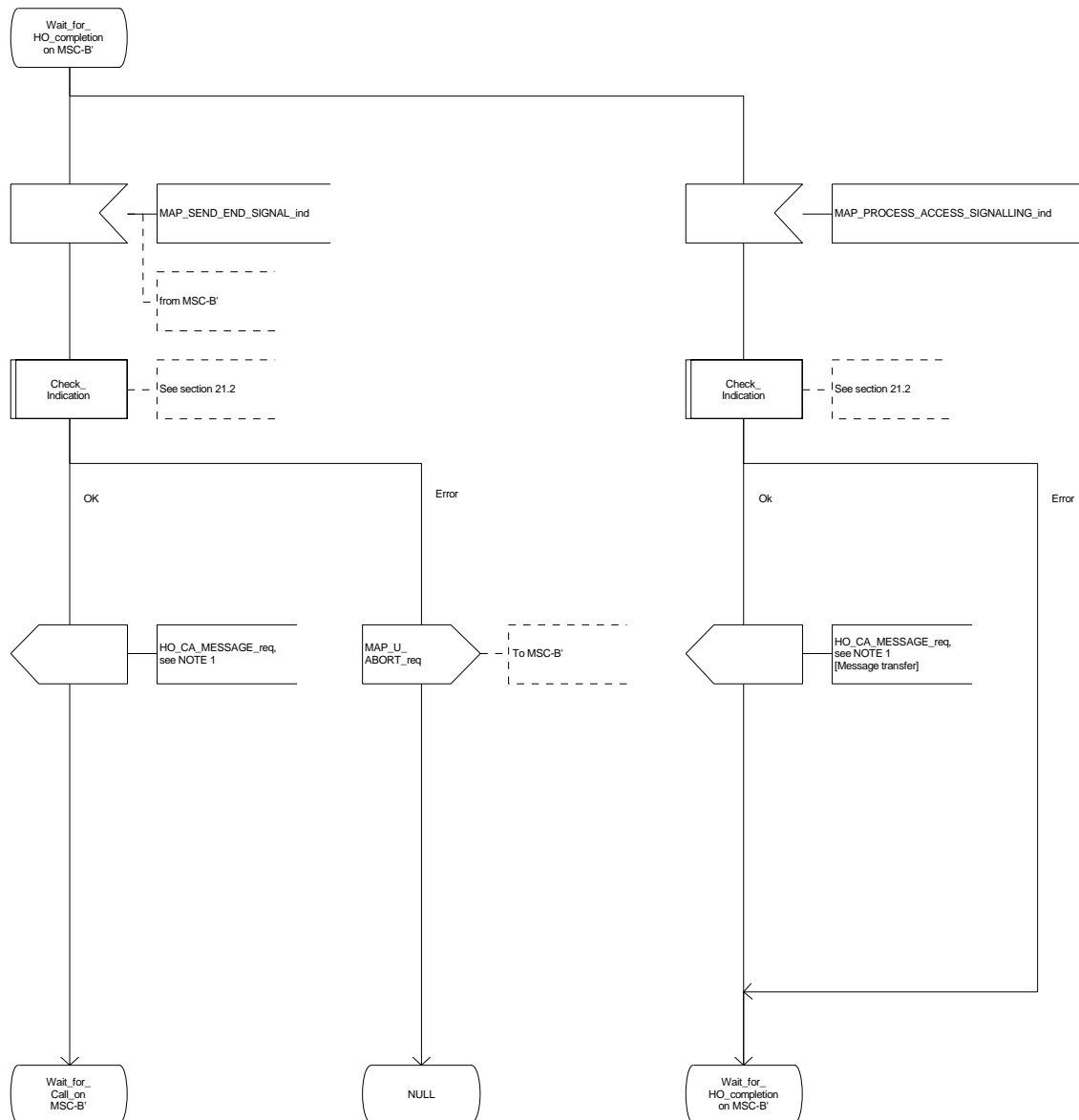
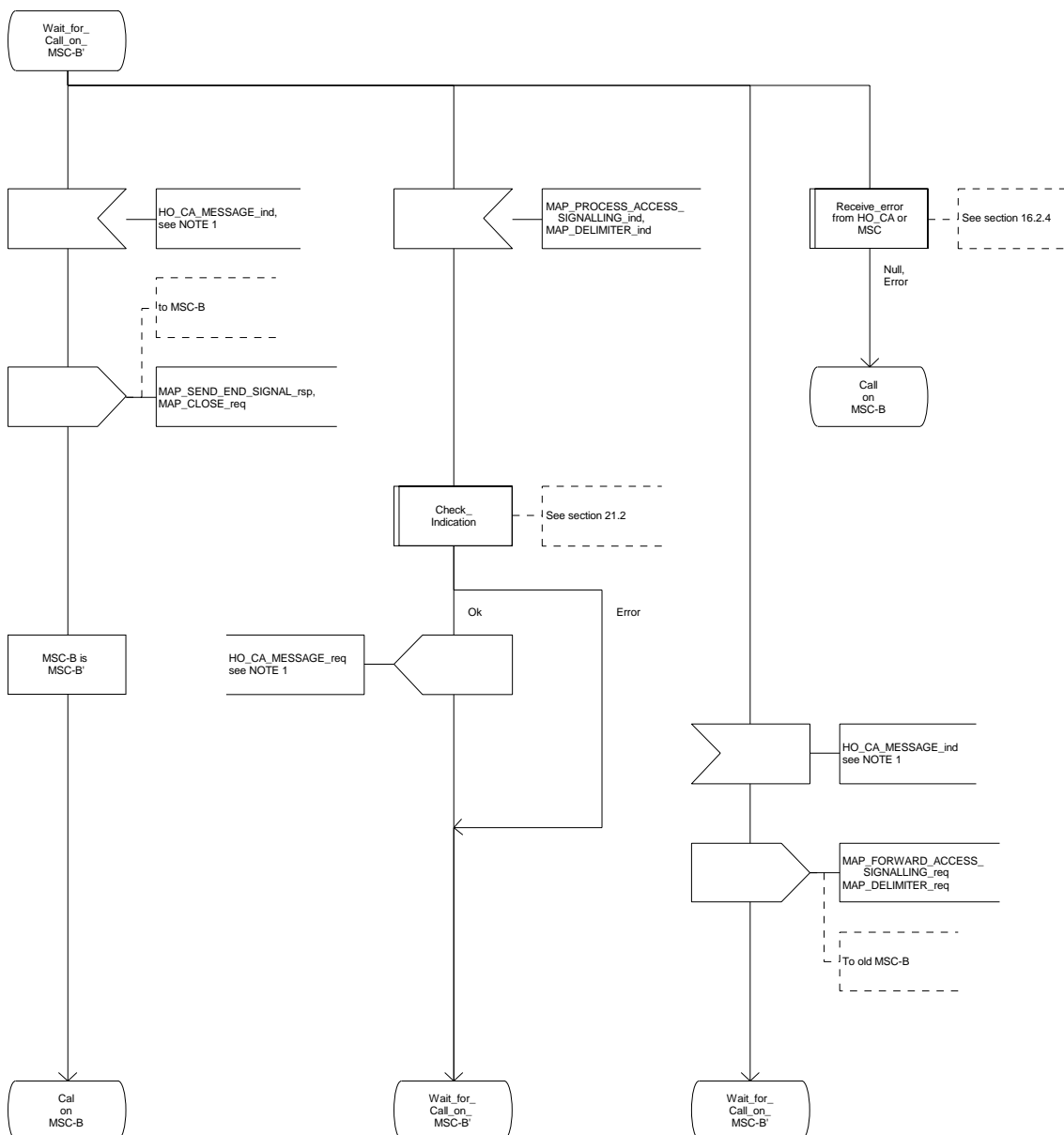


Figure 16.2.2/1 (sheet 12 of 12): Process MSC_A_HO



16.2.3 Handover procedure in MSC-B

This subclause describes the handover procedure in MSC-B, including the request for a handover from another MSC (MSC-A), subsequent handover to a third MSC (MSC-B') or back to the controlling MSC (MSC-A).

16.2.3.1 Basic handover

Opening of the dialogue is described in the macro Receive_Open_Ind in subclause 21.1.

When MSC-B process receives a MAP_PREPARE_HANOVER indication from MSC-A, MSC-B requests its associated VLR to provide a handover number, unless the parameter HO-NumberNotRequired is received in the indication.

When the connection between the MS and MSC-B is established on MSC-B, the Handover Control Application will request the MAP application to indicate this event to MSC-A by invoking the MAP_SEND_END_SIGNAL request. When a call is released, MSC-A will inform MSC-B by MAP_SEND_END_SIGNAL response and the MAP dialogue between MSC-A and MSC-B is closed.

16.2.3.2 Allocation of handover number

When a handover number is required, a MAP_ALLOCATE_HANOVER_NUMBER request will be sent to the VLR. The handover number is received in the MAP_SEND_HANOVER_REPORT request, and will be included in the MAP_PREPARE_HANOVER response to MSC-A.

As soon as the call from MSC-A using the handover number arrives in MSC-B, MSC-B shall release the handover number in the VLR using the MAP_SEND_HANOVER_REPORT response.

16.2.3.3 Handling of access signalling

If required by the Handover Control Application, MSC-B invokes the MAP_PROCESS_ACCESS_SIGNALLING request containing the information received on the A-interface that should be transferred to MSC-A (e.g. call control information).

MAP_PROCESS_ACCESS_SIGNALLING is a non-confirmed service and any response from MSC-A will require a MAP_FORWARD_ACCESS_SIGNALLING request.

16.2.3.4 Other procedures in stable handover situation

During a call and after handover, a number of procedures between MSC-A and BSS-B controlled by or reported to MSC-A may be initiated by involving access signalling transfer in both directions.

16.2.3.5 Subsequent handover

The procedure is used when the Handover Control Application in MSC-B has decided that a call is to be handed over to another MSC (either back to the controlling MSC (MSC-A) or to a third MSC (MSC-B')).

After the MAP_PREPARE_SUBSEQUENT_HANOVER response is received from MSC-A, MSC-B will await the disconnection of the call. Once the disconnect is complete, MSC-B will inform its VLR by invoking the MAP_SEND_HANOVER_REPORT confirmation. VLR-B will then release the allocated handover number.

The subsequent handover procedure is shown in figure 16.2/3.

16.2.3.6 SDL Diagrams

The SDL diagrams on the following pages describe the user process in MSC-B for the procedures described in this subclause.

The services used are defined in subclause 6.4.

NOTE 1: The message primitives HO_CA_MESSAGE in the SDL-diagrams are used to show the internal co-ordination between the MAP application and the Handover Control Application. For a detailed description of the co-ordination between the applications for the handover procedure, see GSM 03.09.

NOTE 2: The order in the SDL diagrams to allocate first the handover number and then the radio resources is not binding.

Figure 16.2.3/1 (sheet 1 of 11): Process MSC_B_HO

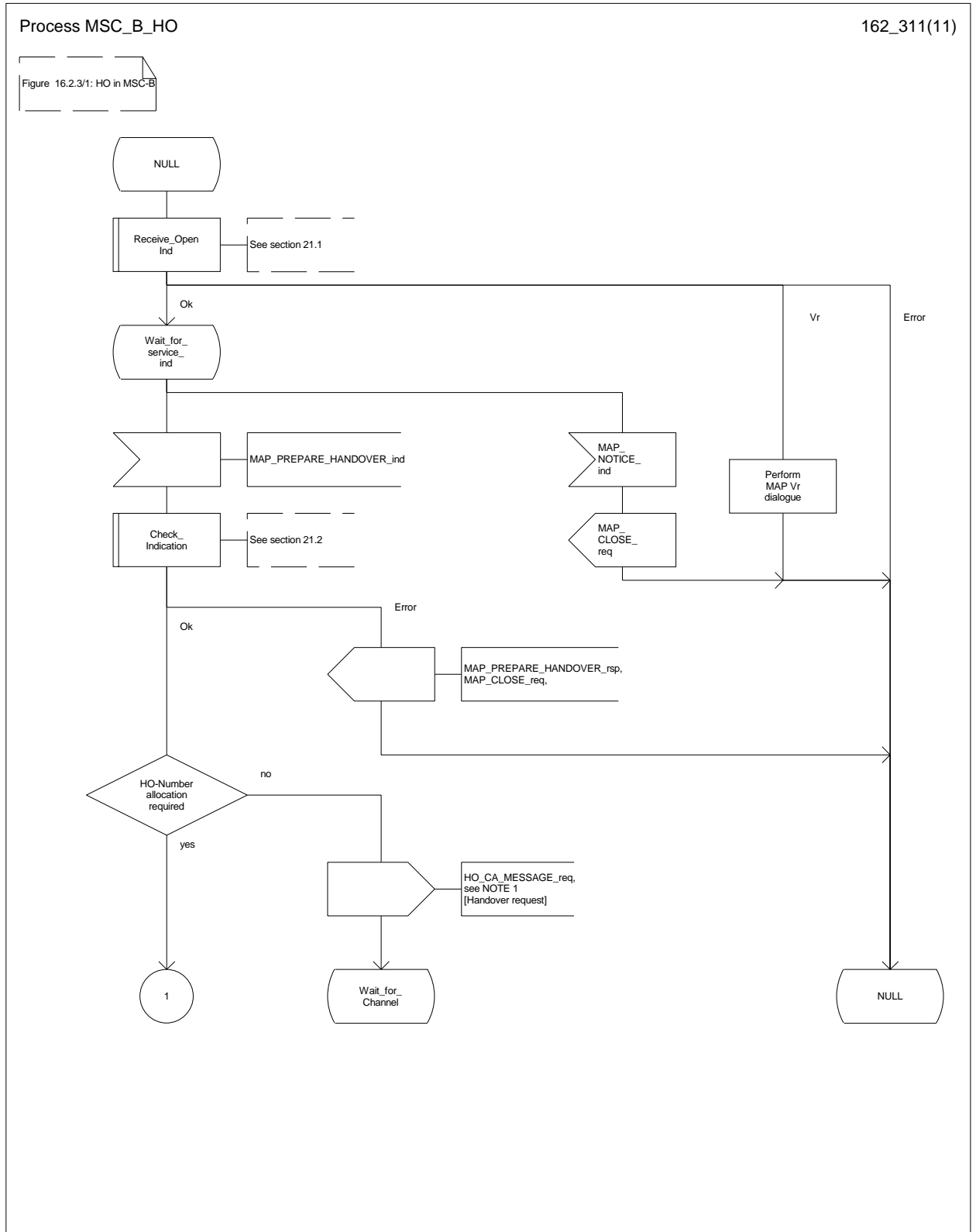


Figure 16.2.3/1 (sheet 2 of 11): Process MSC_B_HO

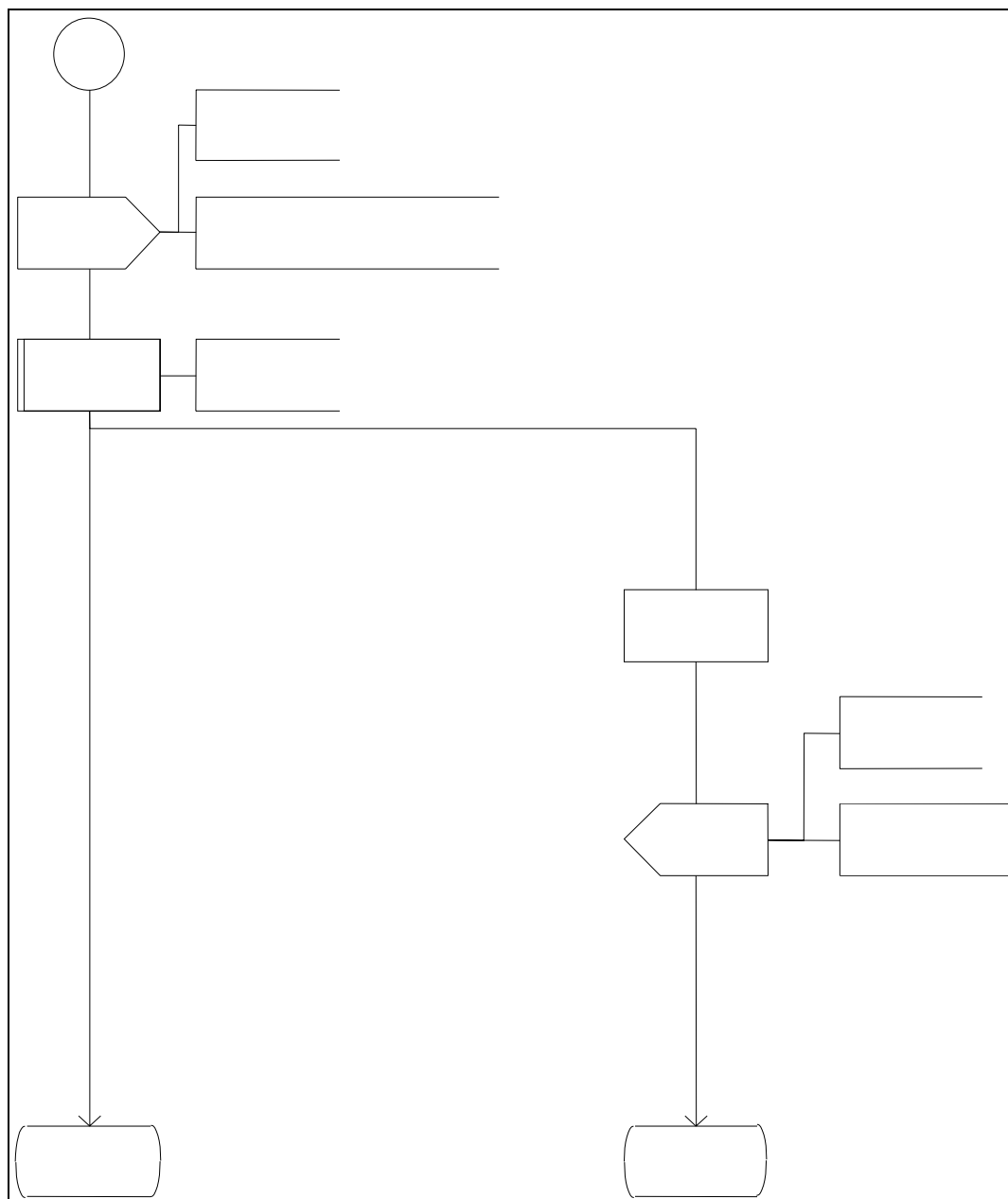


Figure 16.2.3/1 (sheet 3 of 11): Process MSC_B_HO

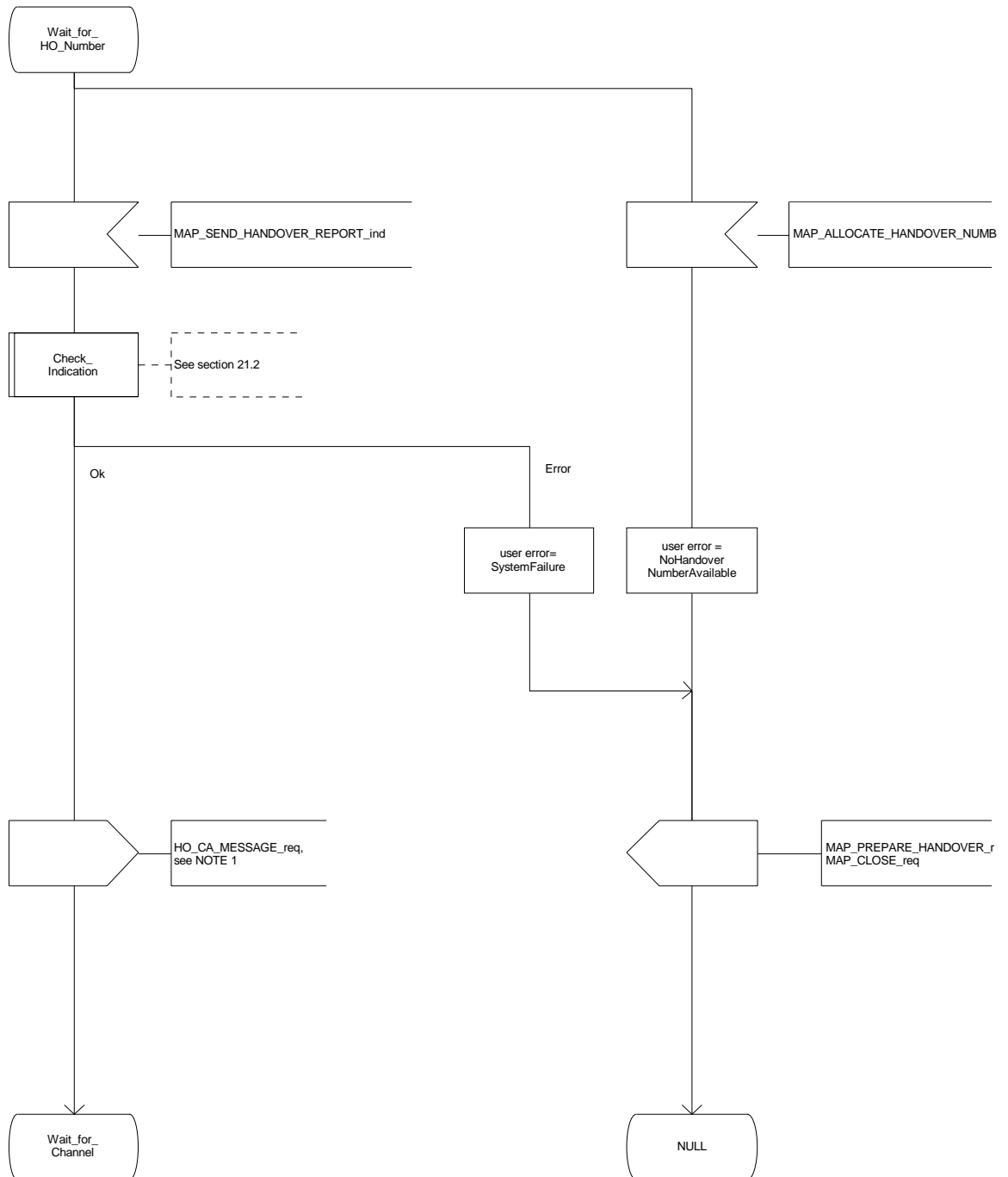


Figure 16.2.3/1 (sheet 4 of 11): Process MSC_B_HO

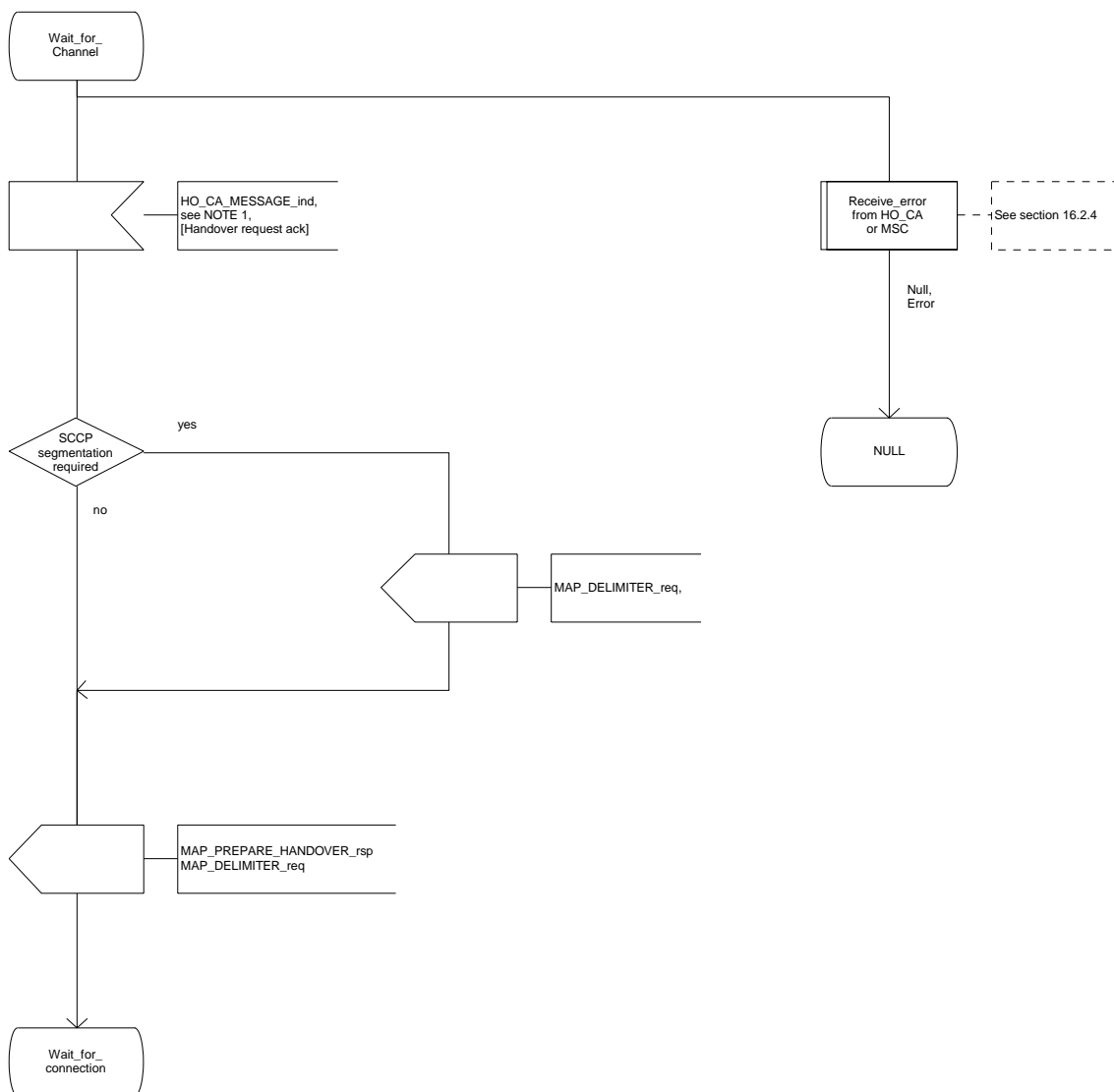


Figure 16.2.3/1 (sheet 5 of 11): Process MSC_B_HO

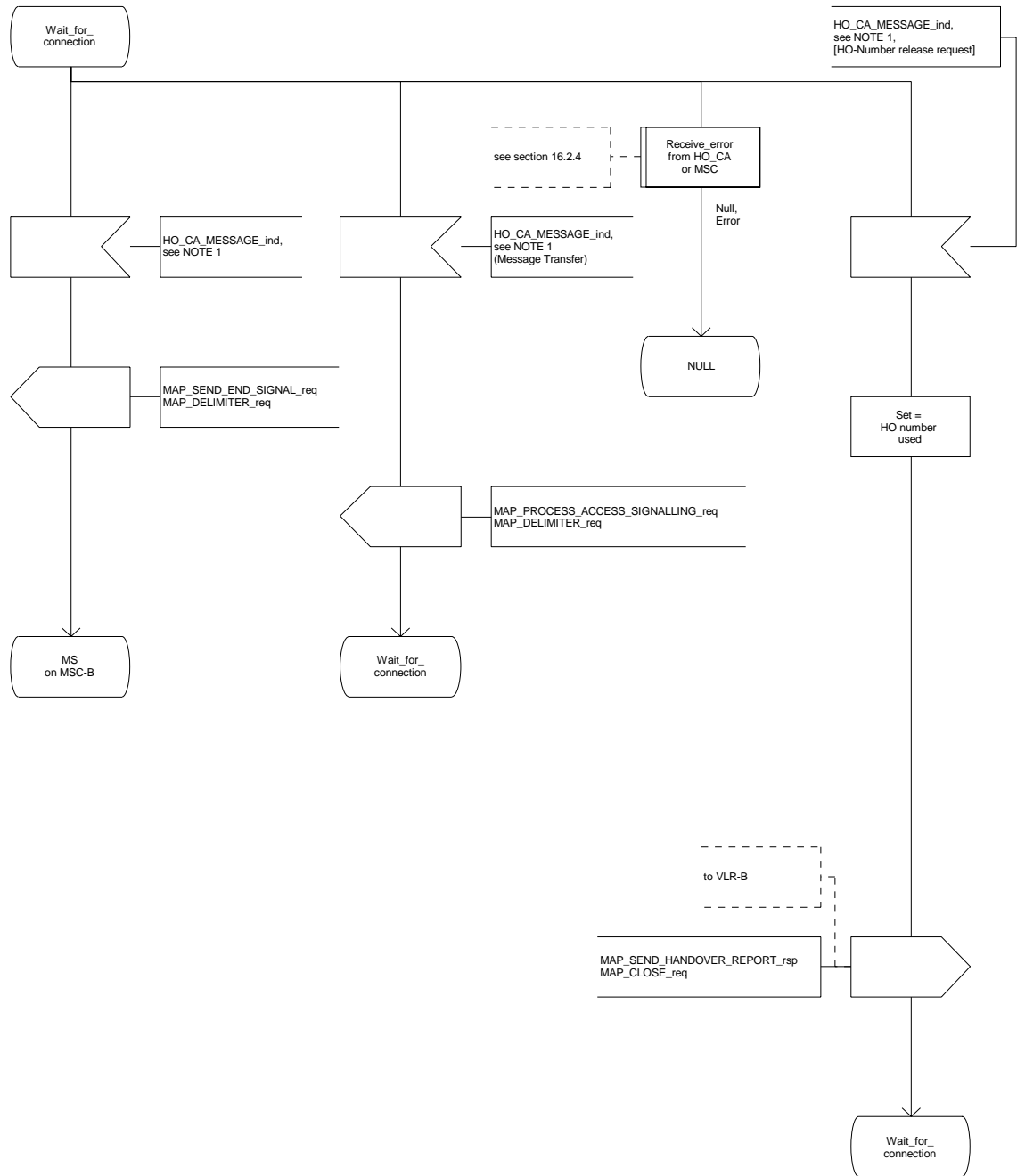


Figure 16.2.3/1 (sheet 6 of 11): Process MSC_B_HO

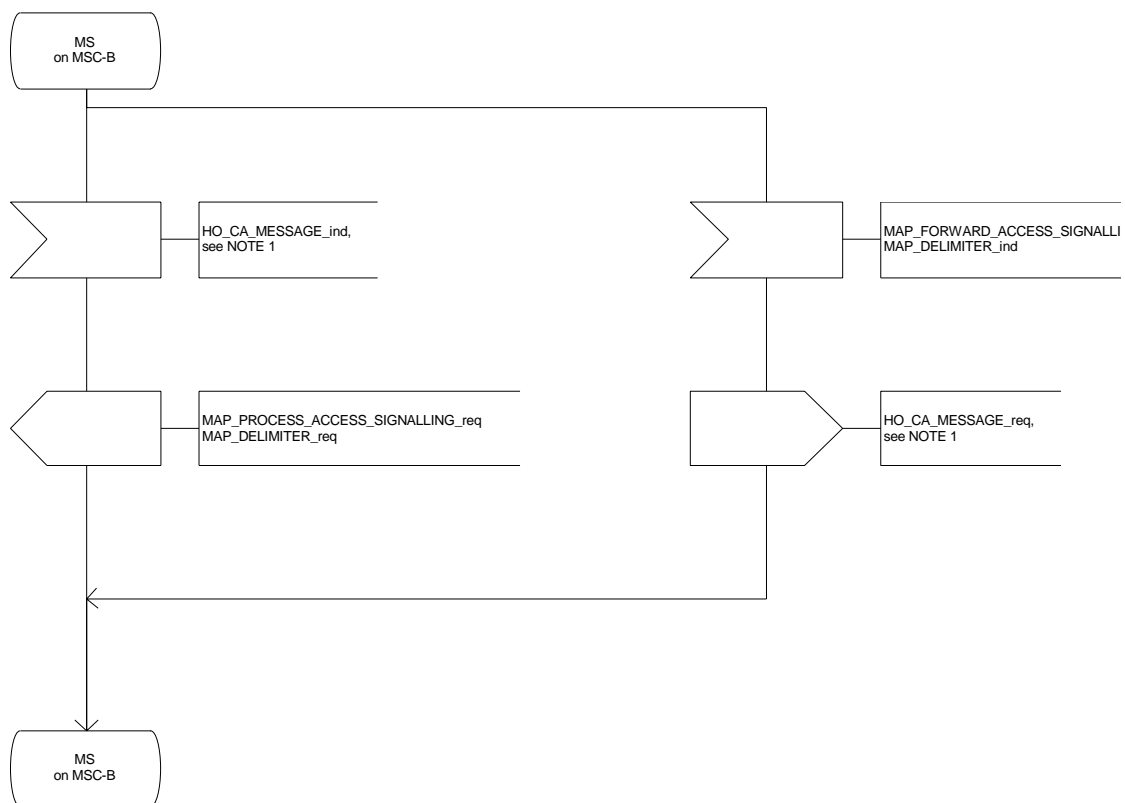


Figure 16.2.3/1 (sheet 7 of 11): Process MSC_B_HO

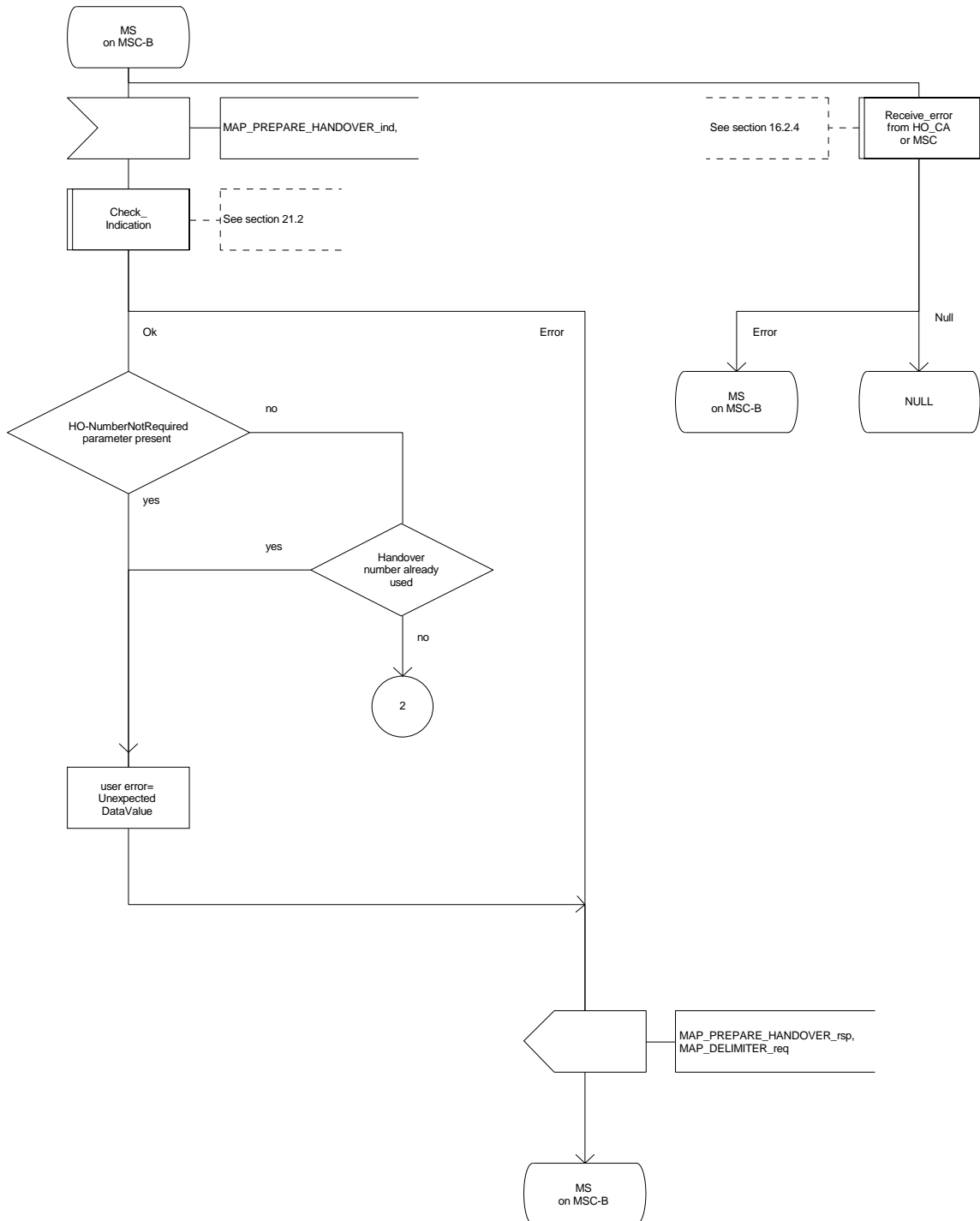


Figure 16.2.3/1 (sheet 8 of 11): Process MSC_B_HO

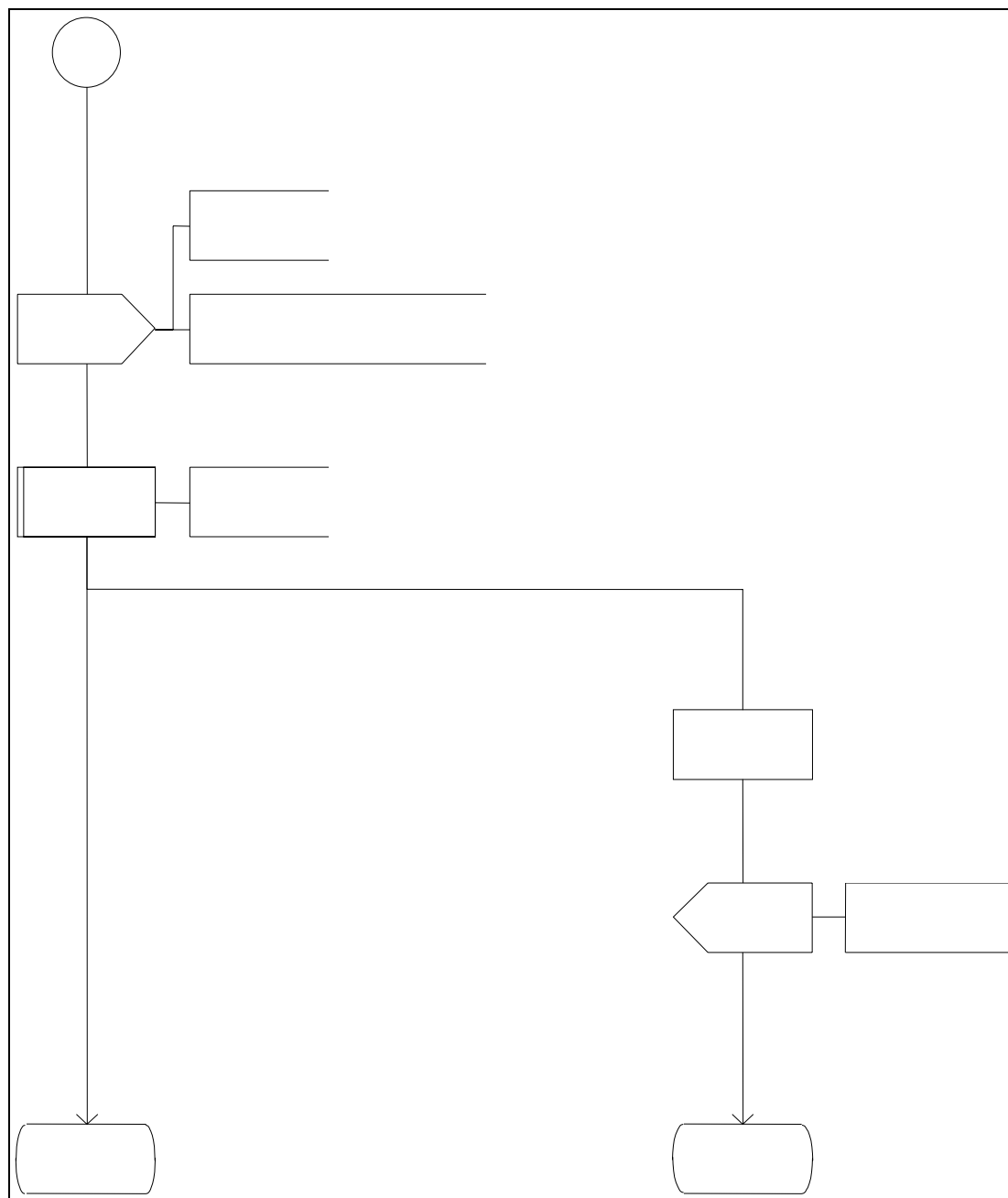


Figure 16.2.3/1 (sheet 9 of 11): Process MSC_B_HO

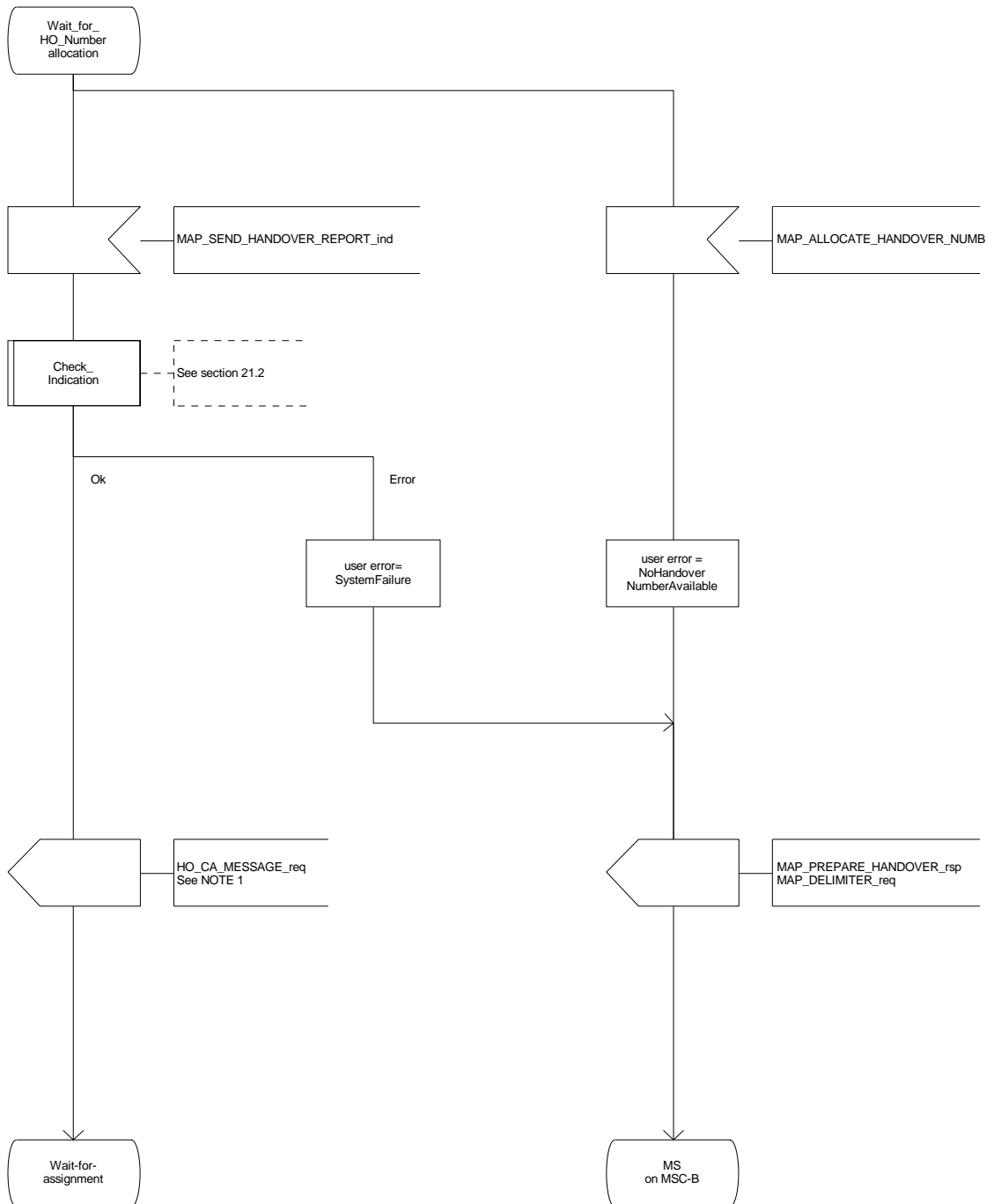


Figure 16.2.3/1 (sheet 10 of 11): Process MSC_B_HO

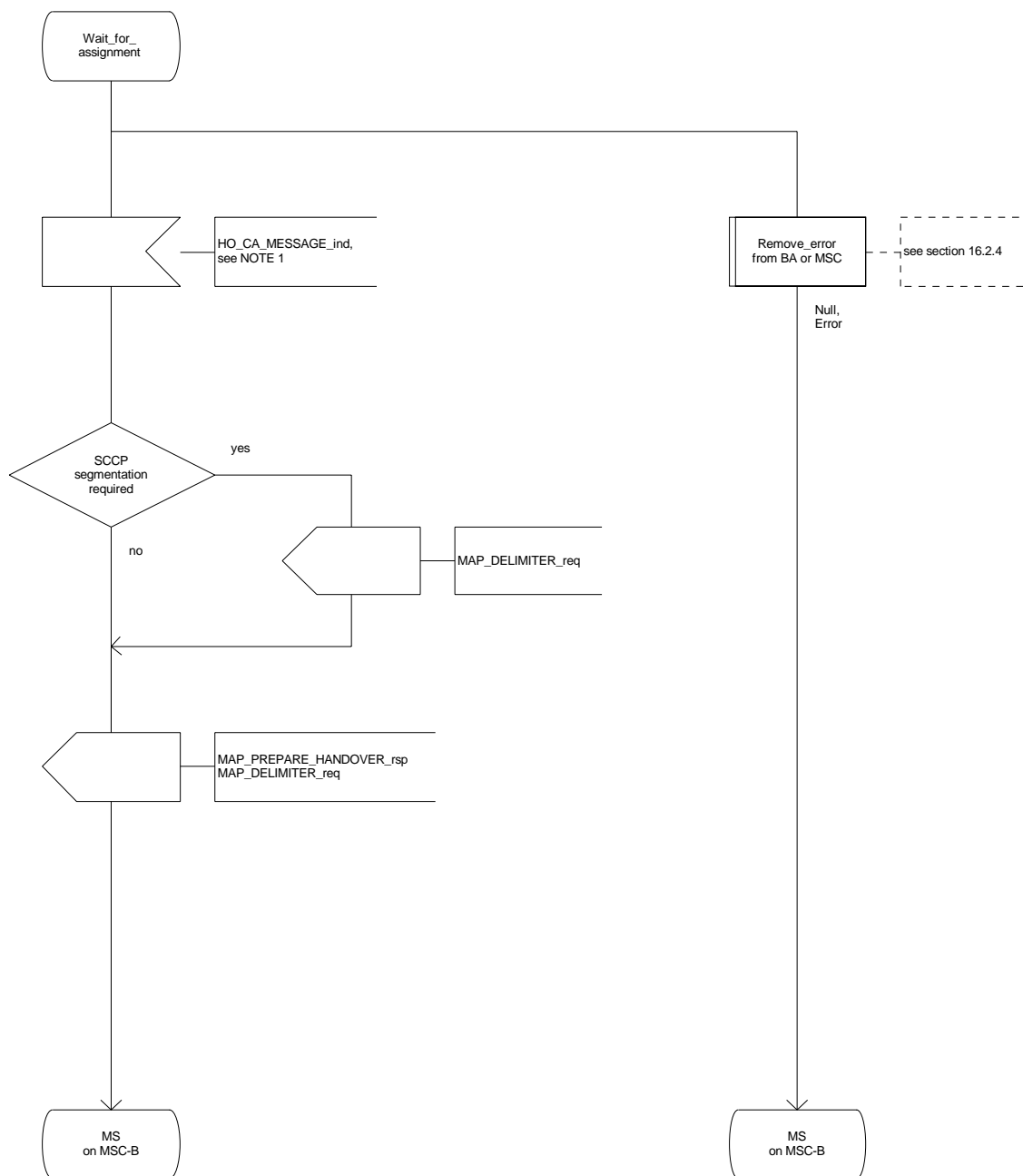


Figure 16.2.3/1 (sheet 11 of 11): Process MSC_B_HO

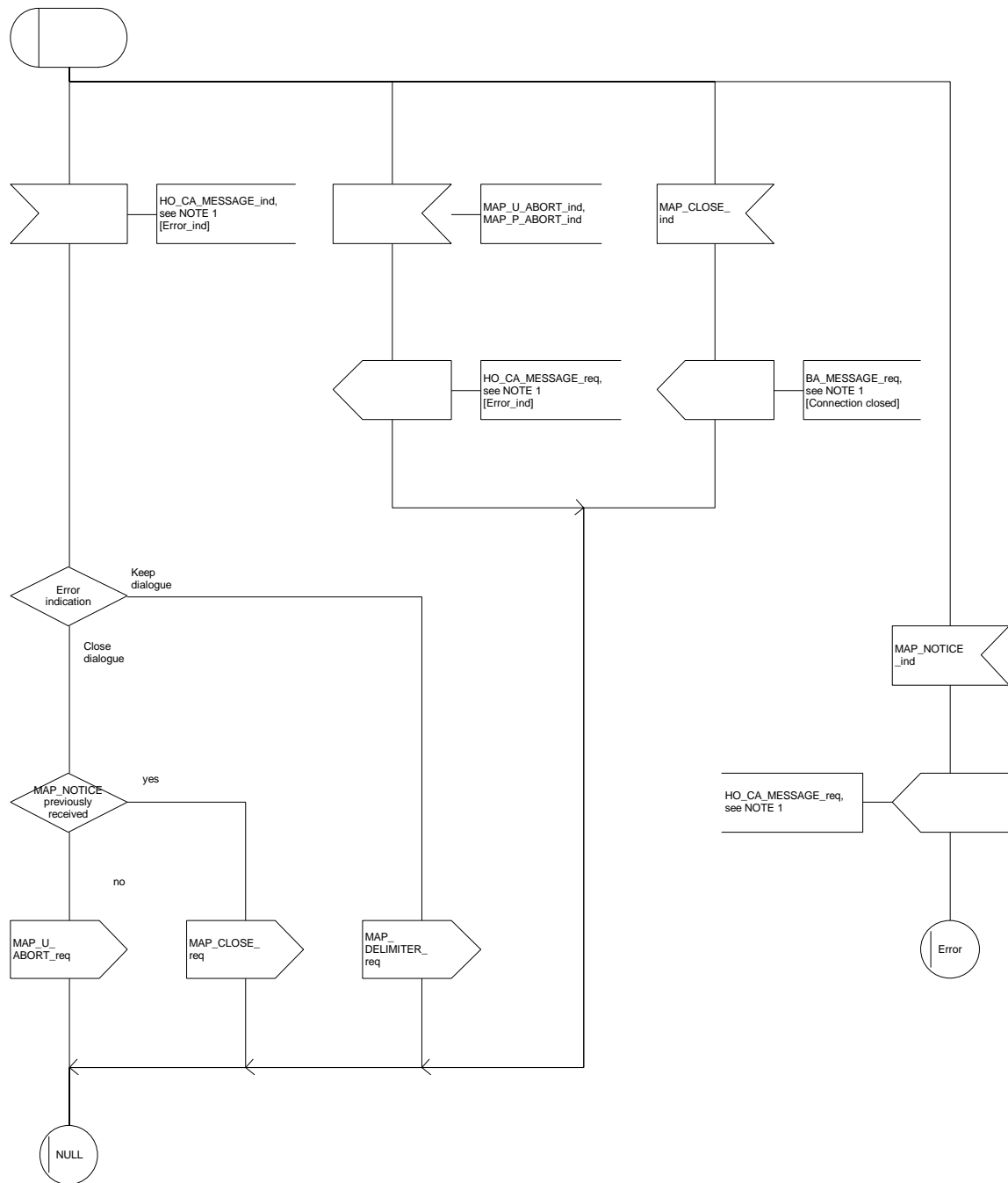


16.2.4 Handover error handling macro

This macro is used for the handover procedures to receive errors from the MSCs and from the Handover Control Application at any state of a handover process.

If a MAP_NOTICE indication is received, the Handover Control Application is informed and the actual situation is kept and the Handover Control Application decides how the handover process should continue. In all other cases the MSC is returned to a "NULL" state.

Figure 16.2.4/1: Macro Receive_error_from_HO_CA_or_MSC



16.2.5 Handover procedure in VLR

16.2.5.1 Allocation of handover number

When receiving the MAP_ALLOCATE_HANDOVER_NUMBER indication, the VLR will determine whether a handover number is available. If no handover number is available, this will be indicated by a MAP_ALLOCATE_HANDOVER_NUMBER response with the appropriate error.

The handover number allocated will otherwise be returned to MSC-B in the MAP_SEND_HANDOVER_REPORT request.

The handover number will be reserved until a MAP_SEND_HANDOVER_REPORT confirmation is received from MSC-B.

16.2.5.2 SDL Diagrams

The SDL diagrams on the following pages describe the user processes in VLR for the procedures described in this subclause.

The services used are defined in subclause 6.4.

Figure 16.2.5/1 (sheet 1 of 2): Process VLR_B_HO

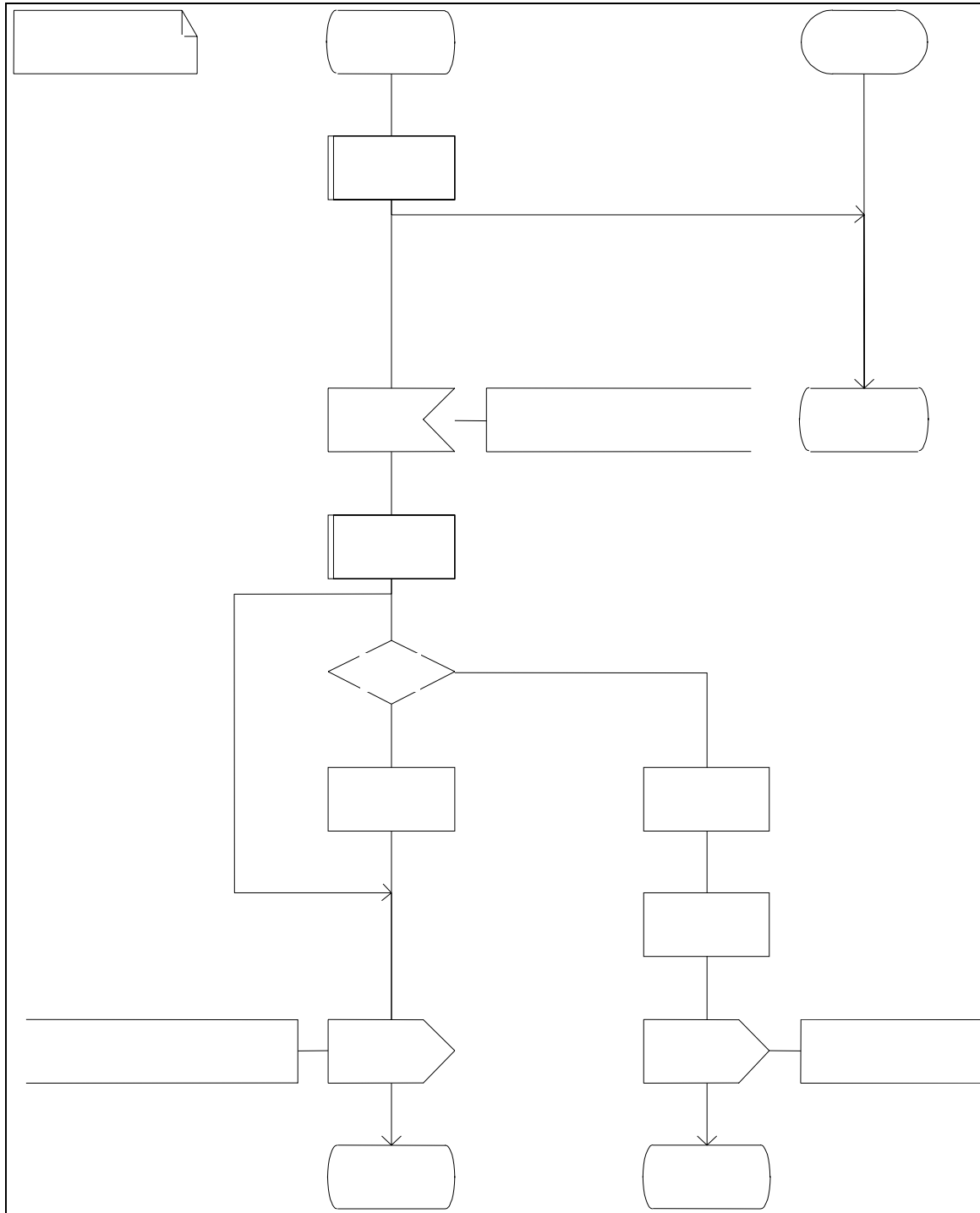
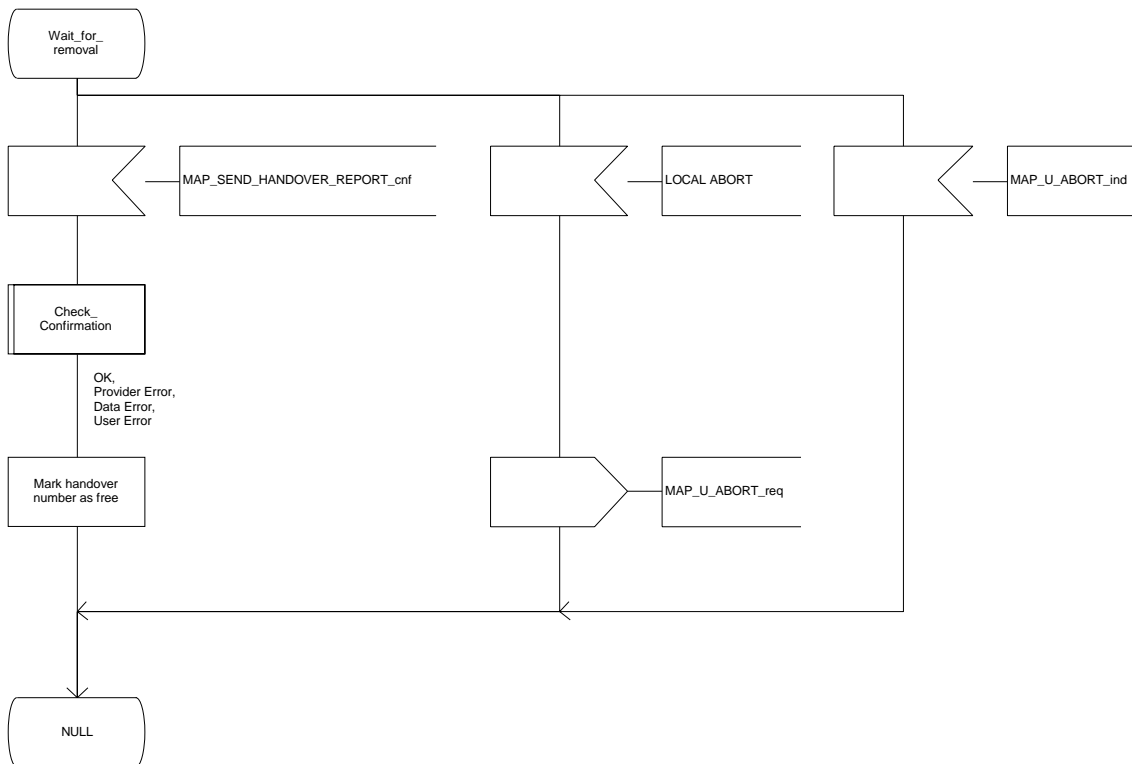


Figure 16.2.5/1 (sheet 2 of 2): Process VLR_B_HO

Figure 16.2.5/1: (sheet 2 of 2)



16.3 Fault recovery procedures

After a fault of a location register, the fault recovery procedures ensure that the subscriber data in the VLR become consistent with the subscriber data that are stored in the HLR for the MS concerned and that the location information in HLR and VLR reflect accurately the current location of the MS.

The detailed specification of fault recovery procedures of location registers is given in GSM 03.07.

16.3.1 VLR fault recovery procedures

The following processes are involved with the restoration of one IMSI record in the VLR:

- In case of a location registration request from the MS:
Update_Location_Area_VLR subclause 16.1.1.3;
Update_Location_HLR subclause 16.1.1.4.
- In case of a mobile terminated call:
PRN_VLR subclause 18.2.4;
RESTORE_DATA_VLR subclause 18.2.4;
RESTORE_DATA_HLR subclause 16.3.3;
ICS_VLR subclause 18.3.3.

After a restart, the VLR shall erase all IMSI records affected by the failure and shall cause all affected TMSIs and all affected LMSIs to become invalid. There will be no subscriber data or location information stored for an affected MS until after the VLR has received either a MAP_PROVIDE_ROAMING_NUMBER indication or a MAP_UPDATE_LOCATION_AREA indication for that MS. Restoration of subscriber data in the VLR is triggered individually for each IMSI record by receipt of either of these indications.

Reception of either a MAP_UPDATE_LOCATION_AREA indication or a MAP_PROVIDE_ROAMING_NUMBER indication with an IMSI that is unknown in the VLR causes creation of a skeleton IMSI record that is marked as:

- not confirmed by radio contact by the indicator "Confirmed by Radio Contact" (The function of this indicator is described in GSM 03.07), and
- not confirmed by HLR by the indicator "Confirmed by HLR" (The function of this indicator is described in GSM 03.07).

A third indicator "Location Information Confirmed in HLR" is allocated to each IMSI record in the VLR (The function of this indicator is described in GSM 03.07).

The indicator "Location Information Confirmed in HLR" shall be checked whenever authenticated radio contact with an MS has been established. The status "Not Confirmed" of this indicator shall force the VLR to invoke the MAP_UPDATE_LOCATION service but it shall never cause rejection of a mobile originated request. The status is changed from "Not Confirmed" to "Confirmed" only after successful completion of a MAP_UPDATE_LOCATION procedure for the MS concerned.

If the VLR serves only one MSC, the indicator "Location Information Confirmed in HLR" is only relevant to the HLR restoration procedure and an initial value must be assigned when an IMSI record is created in the VLR:

- if the IMSI record was created due to a roaming number request, the initial value must be set to "Confirmed";
- if reception of a MAP_UPDATE_LOCATION_AREA indication causes creation of the IMSI record, the initial value must be "Not Confirmed".

If the VLR serves more than one MSC, the indicator "Location Information Confirmed in HLR" is used in the VLR restoration procedure as well as in the HLR restoration procedure. When an IMSI record is created in the VLR, the indicator must be set to "Not Confirmed".

VLR restoration triggered by a location registration request

Upon receipt of a MAP_UPDATE_LOCATION_AREA indication, the VLR retrieves authentication data from the HLR by using the MAP_SEND_AUTHENTICATION_INFO service if authentication is required and if no authentication data are available in the VLR for the IMSI concerned (see figure 16.1.1/6).

Receipt of a MAP_UPDATE_LOCATION_AREA indication for an MS whose IMSI is unknown in the VLR or whose data stored in the VLR are marked as "Not Confirmed" by the indicator "Confirmed by HLR" and/or by the indicator "Location Information Confirmed in HLR" forces the VLR to invoke the MAP_UPDATE_LOCATION service after successful authentication, if required. The location updating procedure is performed as described in subclause 16.1.

Any other mobile originated request from an MS whose IMSI is unknown in the VLR or whose subscriber data stored in the VLR are marked as "Not Confirmed" by the indicator "Confirmed by HLR" shall be rejected with error cause "Unidentified Subscriber". This causes the MS to trigger the location registration procedure.

After successful completion of the MAP_UPDATE_LOCATION procedure, the indicators "Confirmed by HLR" and "Location Information Confirmed in HLR" are set to "Confirmed".

The indicator "Confirmed by Radio Contact" is set to "Confirmed" when the radio contact with the MS is authenticated.

VLR restoration triggered by a roaming number request

Figure 16.3/1 illustrates the signalling sequence for restoration of an IMSI record in the VLR triggered by a mobile terminating call set-up.

Upon receipt of a MAP_PROVIDE_ROAMING_NUMBER indication for an IMSI that is unknown in the VLR and for which authentication is required, the VLR retrieves authentication data from the HLR by using the MAP_SEND_AUTHENTICATION_INFO service after an MSRN has been sent to the HLR in the MAP_PROVIDE_ROAMING_NUMBER response.

Receipt of a MAP_PROVIDE_ROAMING_NUMBER indication for an MS whose IMSI is unknown in the VLR or whose data record in the VLR is marked as "Not Confirmed" by the indicator "Confirmed by HLR" forces the VLR to request subscriber data from the HLR by sending a MAP_RESTORE_DATA request which triggers one or more INSERT_SUBSCRIBER_DATA operations from the HLR. The MAP_RESTORE_DATA request may also be used to send the LMSI to the HLR.

The MAP_RESTORE_DATA process in the VLR is described in subclause 18.2.4.

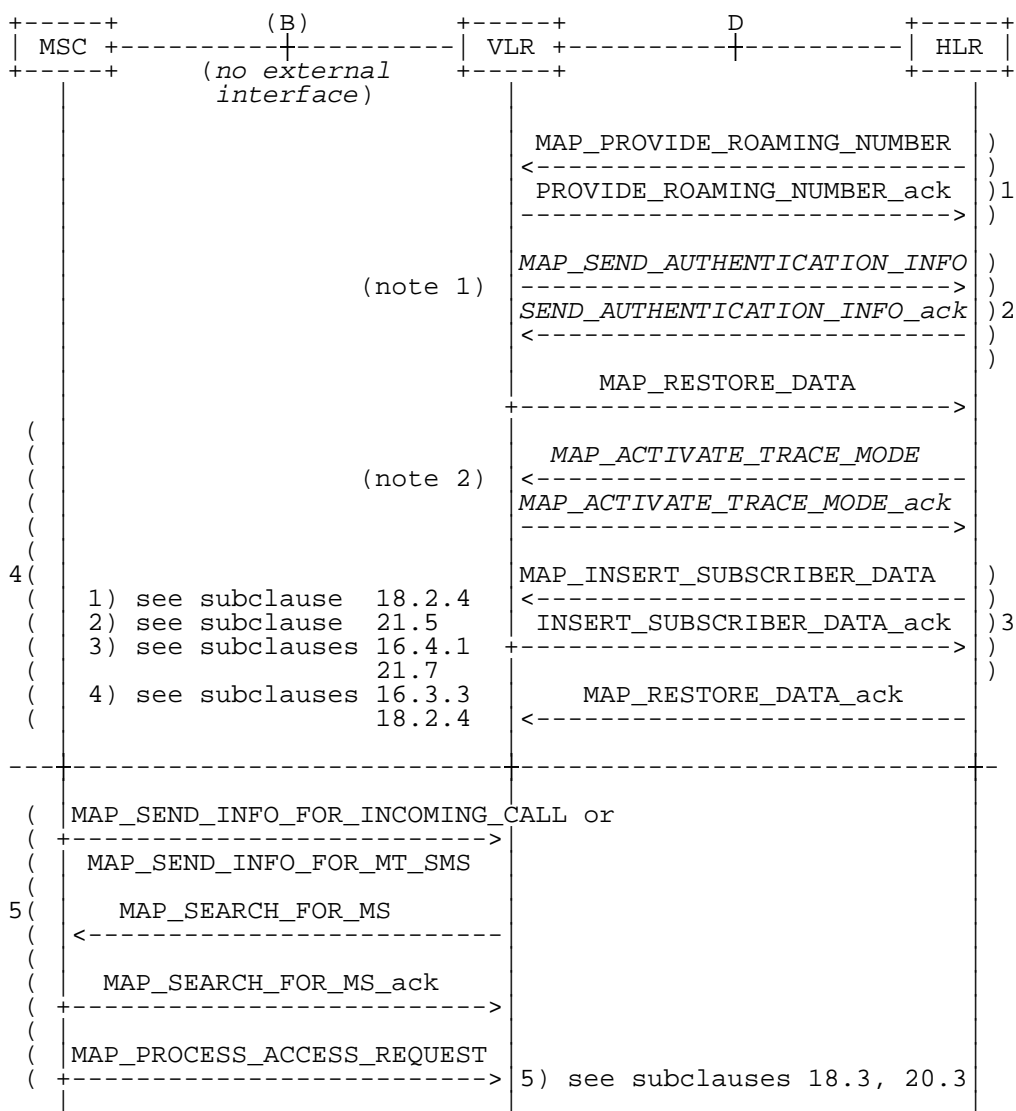
The MAP_RESTORE_DATA process in the HLR is described in subclause 16.3.3.

After successful completion of the MAP_RESTORE_DATA procedure, the indicator "Confirmed by HLR" is set to "Confirmed".

If restoration of an IMSI record was triggered by a MAP_PROVIDE_ROAMING_NUMBER indication (i.e. by a mobile terminating call), the VLR has no valid Location Area Identity information for the MS concerned before successful establishment of the first authenticated radio contact. Upon receipt of a MAP_SEND_INFO_FOR_INCOMING_CALL indication from the MSC (see 5 in figure 16.3/1) for an MS whose subscriber data are marked as "Confirmed" by the indicator "Confirmed by HLR" but not confirmed by radio contact, the VLR shall invoke a "MAP_SEARCH_FOR_MS" instead of a "MAP_PAGE".

A MAP_SEARCH_FOR_MS shall also be performed if the VLR receives a MAP_SEND_INFO_FOR_MT_SMS indication from the MSC for an MS whose IMSI record is marked as "Confirmed" by the indicator "Confirmed by HLR" but not confirmed by radio contact.

The indicator "Confirmed by Radio Contact" is set to "Confirmed" when authenticated radio contact caused by a mobile originated or a mobile terminated activity is established.



NOTE 1: If authentication required.

NOTE 2: If subscriber tracing active in HLR.

Figure 16.3/1: Procedures related to restoration of VLR in case of mobile terminated call set-up

16.3.2 HLR fault recovery procedures

The following processes are involved with the restart of the HLR:

- HLR_RESTART subclause 16.3.2;
- REC_RESET_IN_VLR subclause 16.3.2.

In the case of a location registration request from the MS, the following processes are involved with the HLR restoration procedure:

- Update_Location_Area_VLR subclause 16.1.1.3;
- Update_Location_HLR subclause 16.1.1.4.

In the case of a mobile originated service request, the

- Macro Process_Access_Request_VLR subclause 21.4.2; and the
- Process Update_Location_HLR subclause 16.1.1.4,

are involved with the HLR restoration procedure.

For the HLR, periodic back-up of data to non-volatile memory is mandatory.

Data that have been changed in the period of time after the last back-up storage and before the restart of the HLR cannot be recovered by reload from the non-volatile memory. Therefore, a restoration procedure is triggered individually for each IMSI record that has been affected by the HLR fault at the first authenticated radio contact that is established with the MS concerned.

The HLR restoration procedure forces updating of MSC number, VLR number and, if provided by the VLR, LMSI in the HLR. Consistency of subscriber data that are stored in the VLR for an MS that has been affected by a HLR fault with the subscriber data stored in the HLR for this MS will be achieved.

As an implementation option, a notification can be forwarded to the MS to alert the subscriber to check the parameters for supplementary services that allow subscriber controlled input (MAP_FORWARD_CHECK_SS_INDICATION service). If the VLR receives this notification from the HLR it shall forward the notification to the MS.

Figure 16.3/2 illustrates the signalling sequence for HLR restoration.

After a restart, the home location register performs the following actions for the subscriber data records that have been affected by the HLR fault (see figure 16.3/3):

- reload all data from the non-volatile back-up;
- if the MAP_FORWARD_CHECK_SS_INDICATION service is implemented, mark each subscriber record "SS Check Required" by setting the "Check SS" indicator;
- set subscriber tracing deactive in the VLR for each of its Mss;
- reset the "MS Purged" flag for each of its MSs;
- send a MAP_RESET request to the VLRs where its MSs are located (see figure 16.3/4).

The MAP_RESET request contains the HLR number and optionally the HLR Identity List.

When receiving a MAP_RESET indication, the VLR will derive all involved MSs of that HLR either from the HLR Identity List (if present), or from the HLR number. The VLR will then mark these MSs with the indicator "Location Information Confirmed in HLR" set to "Not Confirmed" and will deactivate all subscriber tracings for these Mss (see figure 16.3/5).

The status "Not Confirmed" of the indicator "Location Information Confirmed in HLR" forces the VLR to invoke the MAP_UPDATE_LOCATION service after establishment of authenticated radio contact with the MS concerned.

The MAP_UPDATE_LOCATION procedure is performed as described in subclause 16.1.

After receipt of the MAP_UPDATE_LOCATION acknowledge containing the HLR number, the status of the indicator "Location Information Confirmed in HLR" is changed to "Confirmed".

If the MAP_UPDATE_LOCATION procedure is unsuccessful for any reason, the status of the indicator "Location Information Confirmed in HLR" remains unchanged except for the case that the IMSI record in the VLR is deleted because either of the errors "Unknown Subscriber" or "Roaming Not Allowed" has been received from the HLR in response to a MAP_UPDATE_LOCATION request.

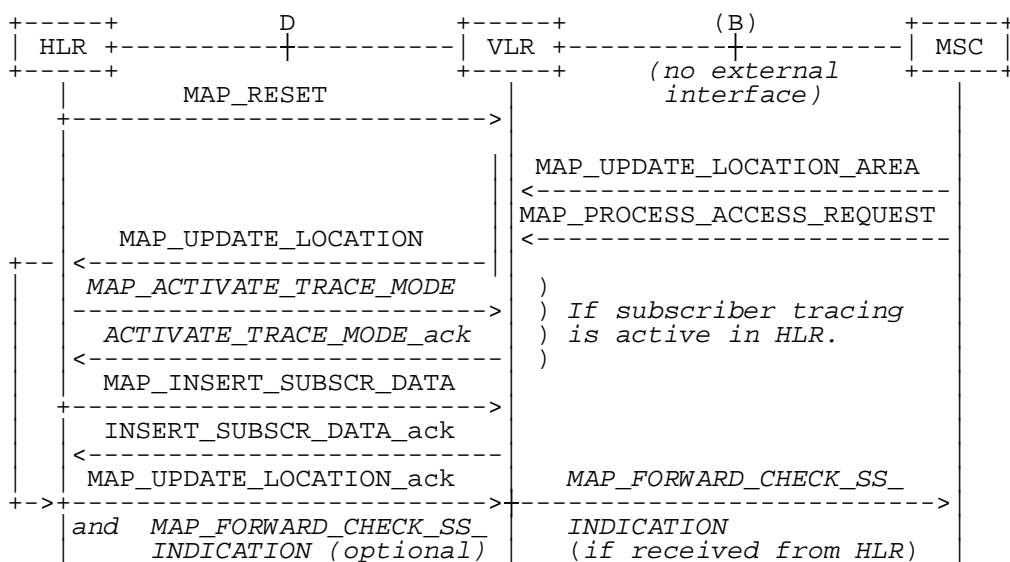


Figure 16.3/2: Procedures related to restoration of HLR

Figure 16.3/3: Process HLR_RESTART

Process HLR_RESTART

163_3(1)

Figure 16.3/3 Restoration of the HLR
Application process in the HLR for HLR Restart

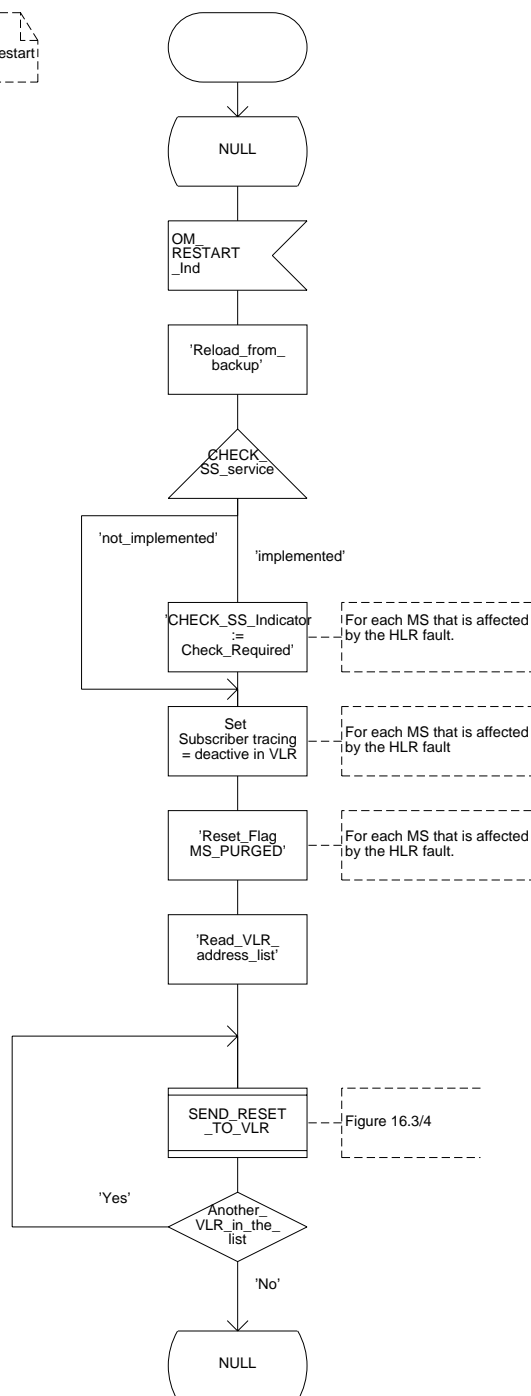
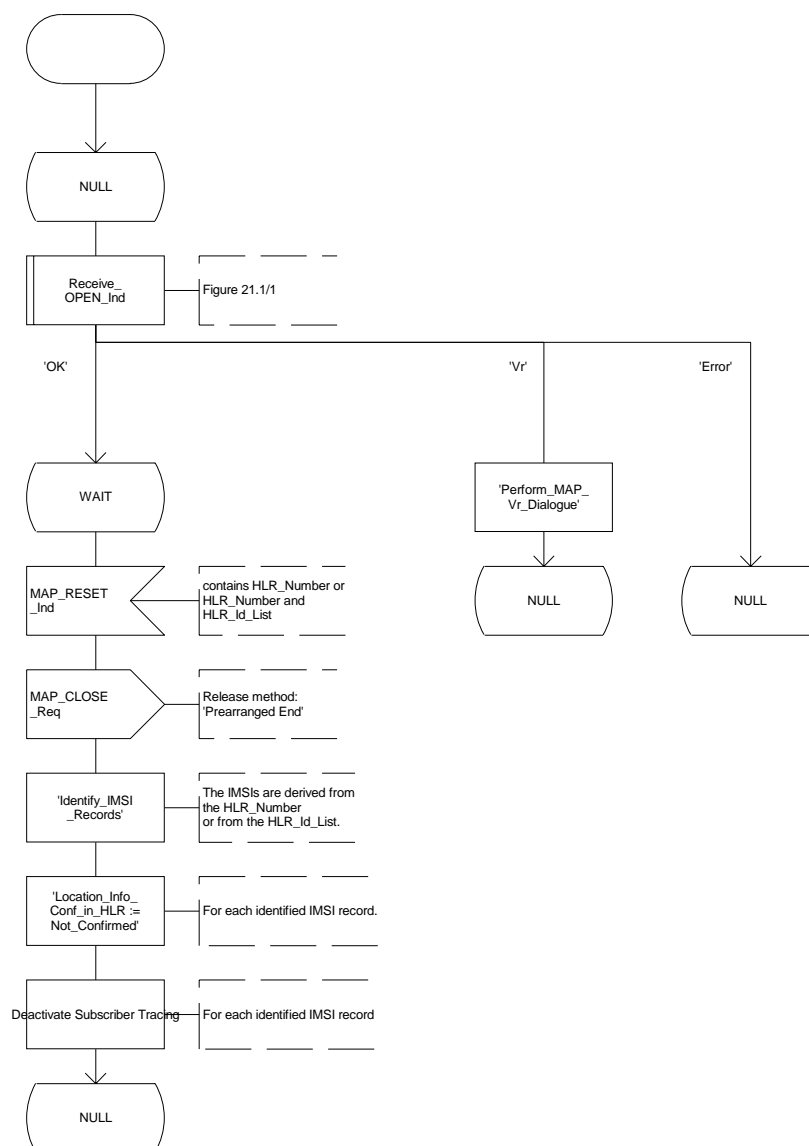


Figure 16.3/5: Process REC_RESET_IN_VLR

Process REC_RESET_IN_VLR

163_5(1)

Figure 16.3/5 Restoration of the HLR - Application process in the VLR for reception of the RESET message from HLR



16.3.3 VLR restoration: the restore data procedure in the HLR

The MAP_RESTORE_DATA procedure in the HLR (Process RESTORE_DATA_HLR) is described in this subclause; the corresponding procedure in the VLR (RESTORE_DATA_VLR) is described in subclause 18.2.4.

The process RESTORE_DATA_HLR makes use of the following macros:

- Receive_Open_Ind subclause 21.1.1;
- Check_Indication subclause 21.2.1;
- Insert_Subs_Data_Framed_HLR subclause 16.4.1.

The MAP_RESTORE_DATA service is invoked by the VLR after provision of a roaming number in response to a MAP_PROVIDE_ROAMING_NUMBER indication for an unidentified MS (i.e. IMSI unknown in VLR), or for a known MS whose IMSI record is marked as "Not Confirmed" by the indicator "Confirmed by HLR" (see 4 in figure 16.3/1). The process RESTORE_DATA_VLR is shown in figure 18.2/6.

The restore data process in the HLR is activated by receipt of a MAP_RESTORE_DATA indication from the VLR (see figure 16.3/6). If there is a parameter problem in the indication, either of the errors "Unexpected Data Value" or "Data Missing" is returned in the MAP_RESTORE_DATA response; if the subscriber is not known in the HLR, the error "Unknown Subscriber" is returned in the MAP_RESTORE_DATA response. In all of these cases the process in the HLR terminates.

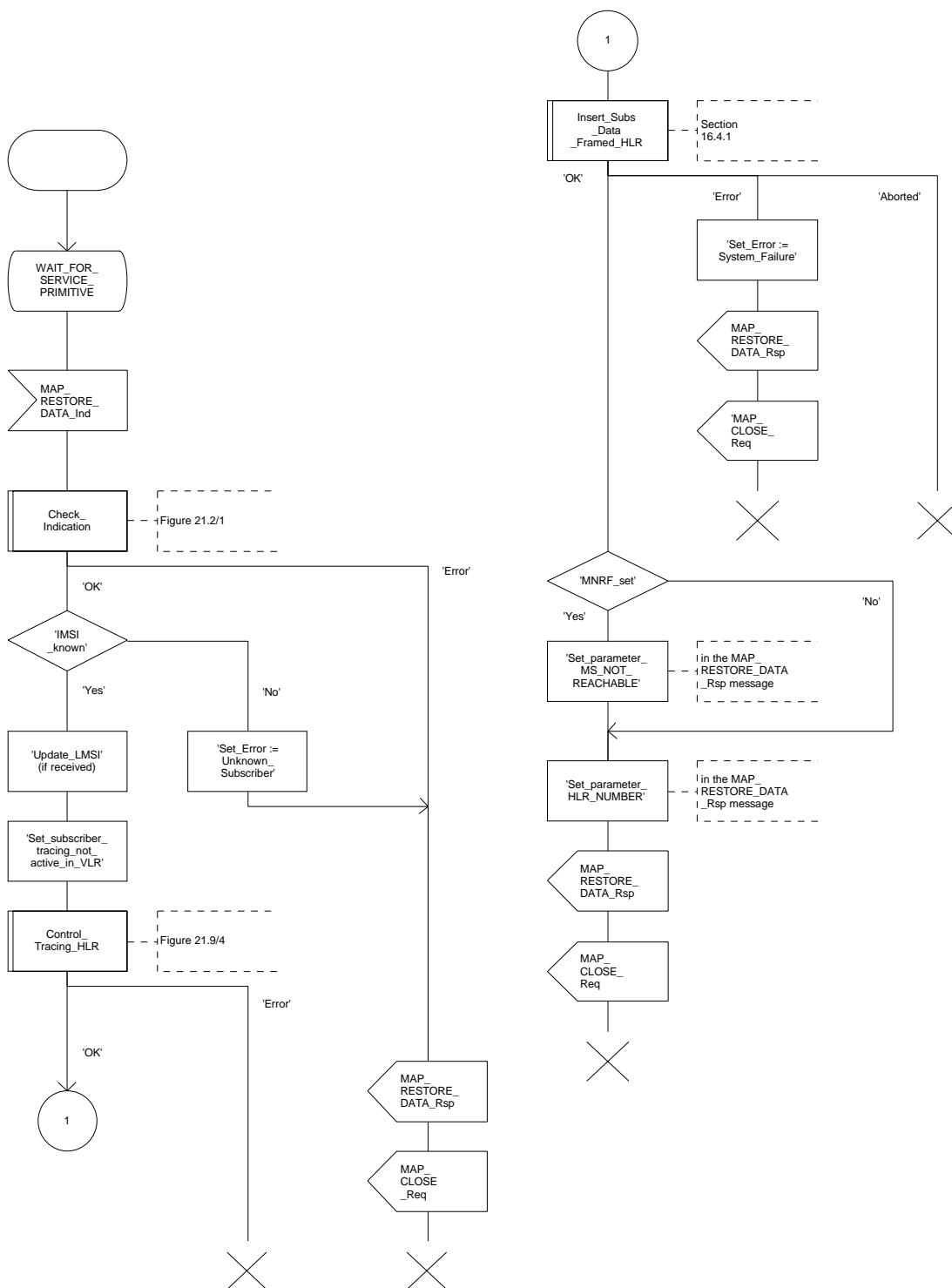
If the MAP_RESTORE_DATA indication is accepted and if the LMSI is received, the HLR updates the LMSI for the IMSI received in the MAP_RESTORE_DATA indication. For this IMSI the HLR sets "subscriber-tracing-not-active-in-VLR" and checks whether tracing is required. This check is handled by the macro "Control_Tracing_HLR" that is described in subclause 21.9. Thereafter, the macro "Insert_Subs_Data_Framed_HLR" that is described in subclause 16.4.1 is invoked. The outcome of the macro Insert_Subs_Data_Framed_HLR is one of:

- abort, in which case the process terminates;
- error, in which case the HLR returns the error "System Failure" in the MAP_RESTORE_DATA response, and the process terminates;
- OK, indicating successful outcome of downloading the subscriber data to the VLR.

After successful completion of the framed MAP_INSERT_SUBSCRIBER_DATA procedure, the HLR Number and, if applicable, the "MS Not Reachable Flag" which is used for SMS, are provided in the MAP_RESTORE_DATA response.

Upon receipt of the MAP_RESTORE_DATA confirmation, the VLR behaves as described in subclause 18.2.4, figure 18.2/6.

Figure 16.3/6: Process RESTORE_DATA_HLR



16.4 Macro Insert_Subs_Data_Framed_HLR

This macro is used by any procedure invoked in HLR which requires the transfer of subscriber data by means of the InsertSubscriberData operation (e.g. Update Location or Restore Data).

The invocation of the operation is done in a dialogue already opened by the framing procedure. Therefore the latter is the one that handles the reception of the open indication and sends the dialogue close request.

The macro calls the process "Send_Insert_Subs_Data" (see subclause 21.7.4) as many times as it is needed for transferring all subscriber data. This process call is meant to describe two possible behaviours of HLR to handle service requests and confirmations:

- either the HLR handles requests and confirmations in parallel; or
- the HLR sends the next request only after receiving the confirmation to the previous one.

Another call is done to the macro "Wait_for_Insert_Subscriber_Data" (see subclause 21.7.3). There the reception and handling of the service confirmations is described.

If certain services required for a subscriber are not supported by the VLR (e.g. Advice of Charge Charging Level), this may result in one of the following outcomes:

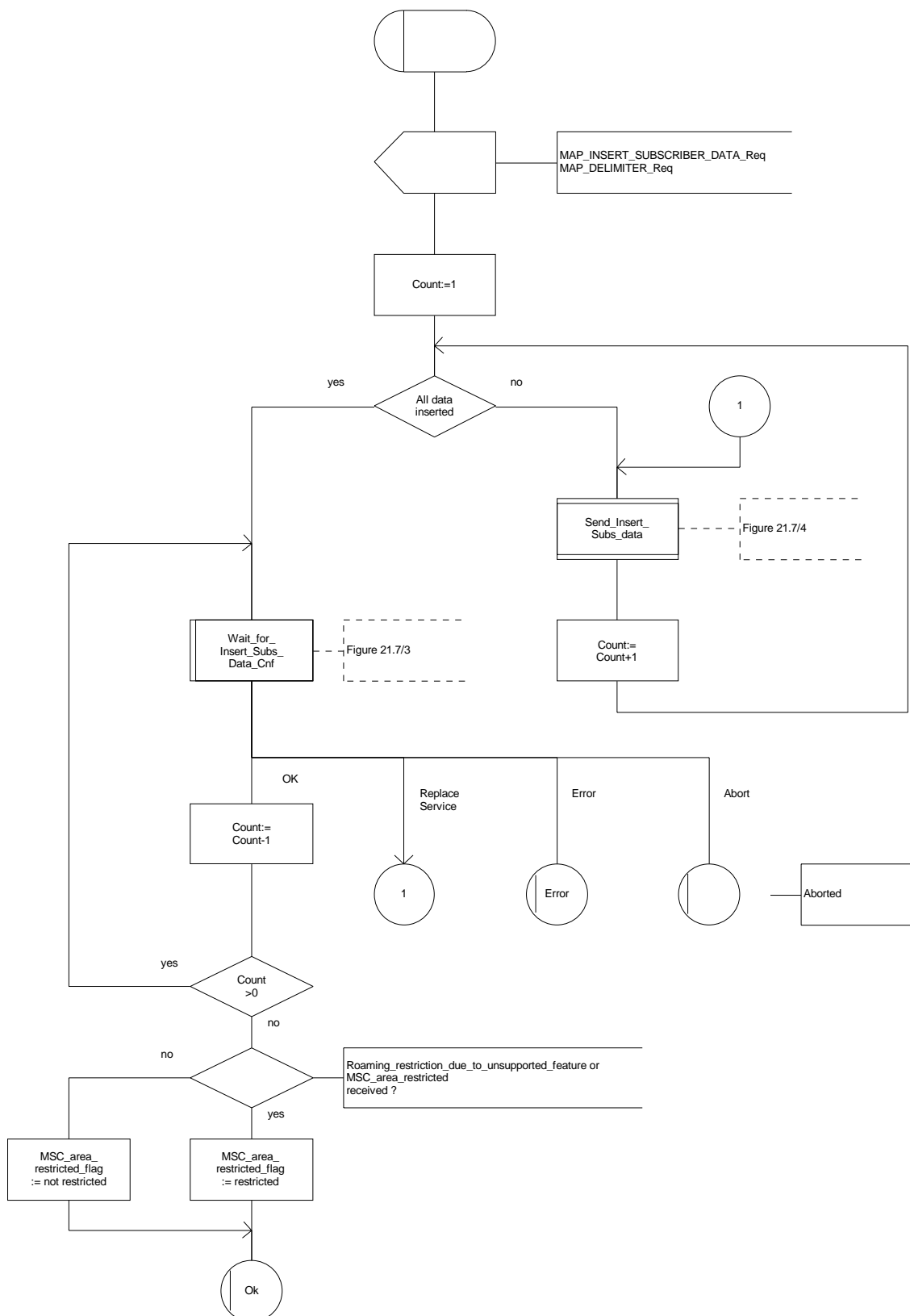
- The HLR stores and sends "Roaming Restriction Due To Unsupported Feature" in a subsequent MAP_INSERT_SUBSCRIBER_DATA service. If "Roaming Restriction Due To Unsupported Feature" is stored in the HLR, the "MSC Area Restricted Flag" shall be set to "restricted". This will prevent MT calls, MT SM and MT USSD from being forwarded to the MSC/VLR;
- The HLR stores and sends other induced subscriber data (e.g. a specific barring program) in a subsequent MAP_INSERT_SUBSCRIBER_DATA service. This will cause rejection of mobile originated service requests, except emergency calls.

When the VLR receives regional subscription data (Zone Code List) it may respond with "MSC Area Restricted" in the MAP_INSERT_SUBSCRIBER_DATA response. In this case the "MSC Area Restricted Flag" shall be set to "restricted" in the HLR. This will prevent MT calls, MT SM and MT USSD from being forwarded to the MSC/VLR.

If the HLR neither stores "Roaming Restriction Due To Unsupported Feature" nor receives "MSC Area Restricted" in the MAP_INSERT_SUBSCRIBER_DATA response, the "MSC Area Restricted Flag" in the HLR shall be set to "not restricted".

The SDL diagram is shown in figure 16.4/1.

Figure 16.4/1: Macro Insert_Subs_Data_Framed_HLR



17 Operation and maintenance procedures

17.1 General

The Operation and Maintenance procedures are needed for operating and maintaining the GSM PLMN network.

The following procedures exist for operation and maintenance purposes:

- i) Tracing procedures;
- ii) Subscriber Data Management procedures;
- iii) Subscriber Identity procedures.

The following application contexts refer to complex MAP Users consisting of several processes:

- subscriberDataManagementContext;
- tracingContext.

These two application contexts need a co-ordinating process in the VLR as described in the following subclauses.

17.1.1 Tracing Co-ordinator for the VLR

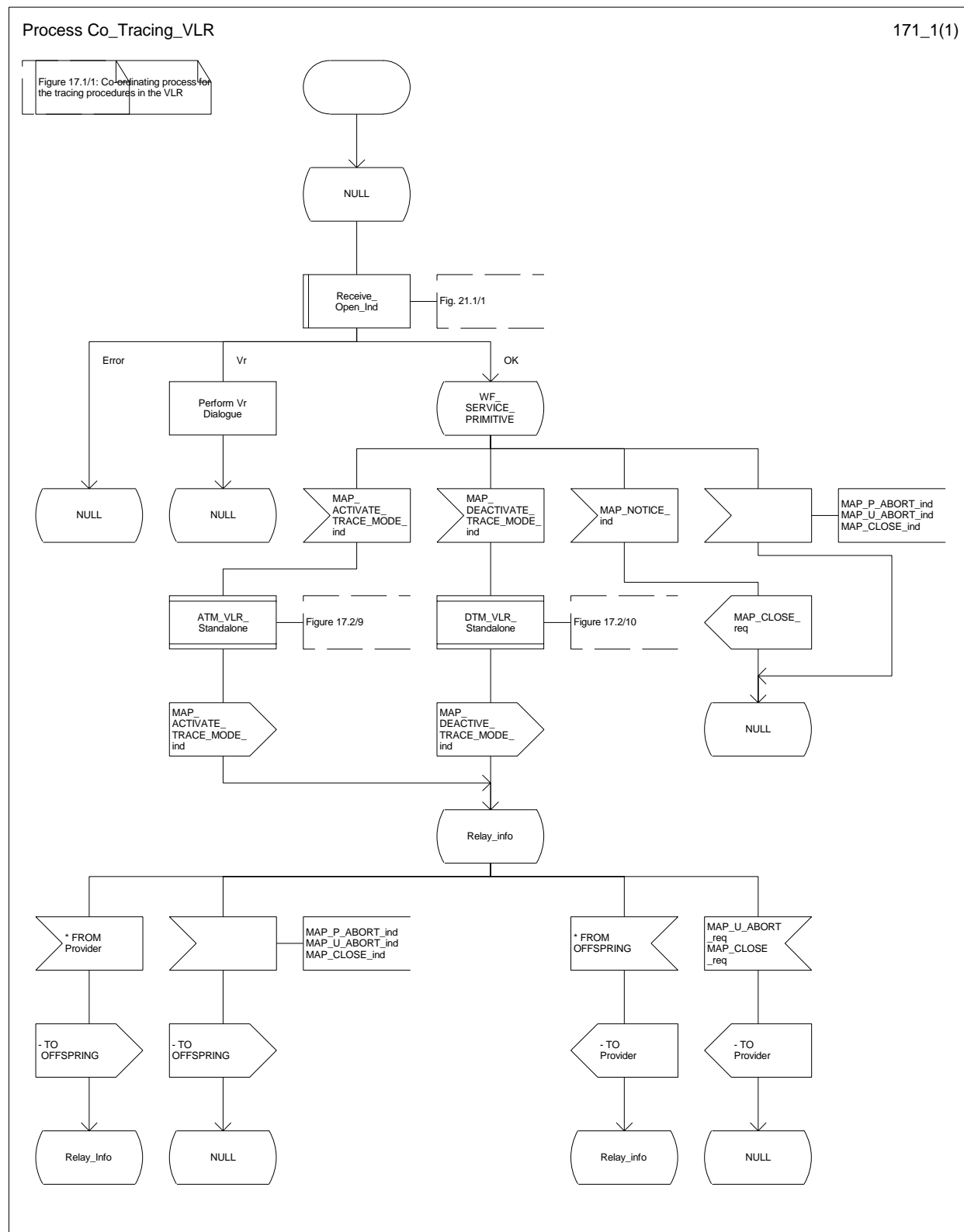
The MAP_OPEN indication opens the dialogue for the stand-alone tracing procedure when the application context tracingContext is received. If that service is successful, the Co-ordinator can receive the first service primitive from the MAP_PM. Depending on the received primitive, the user process is created 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 17.1/1.

Figure 17.1/1: Process Co_Tracing_VLR



17.1.2 **Subscriber Data Management Co-ordinator for the VLR**

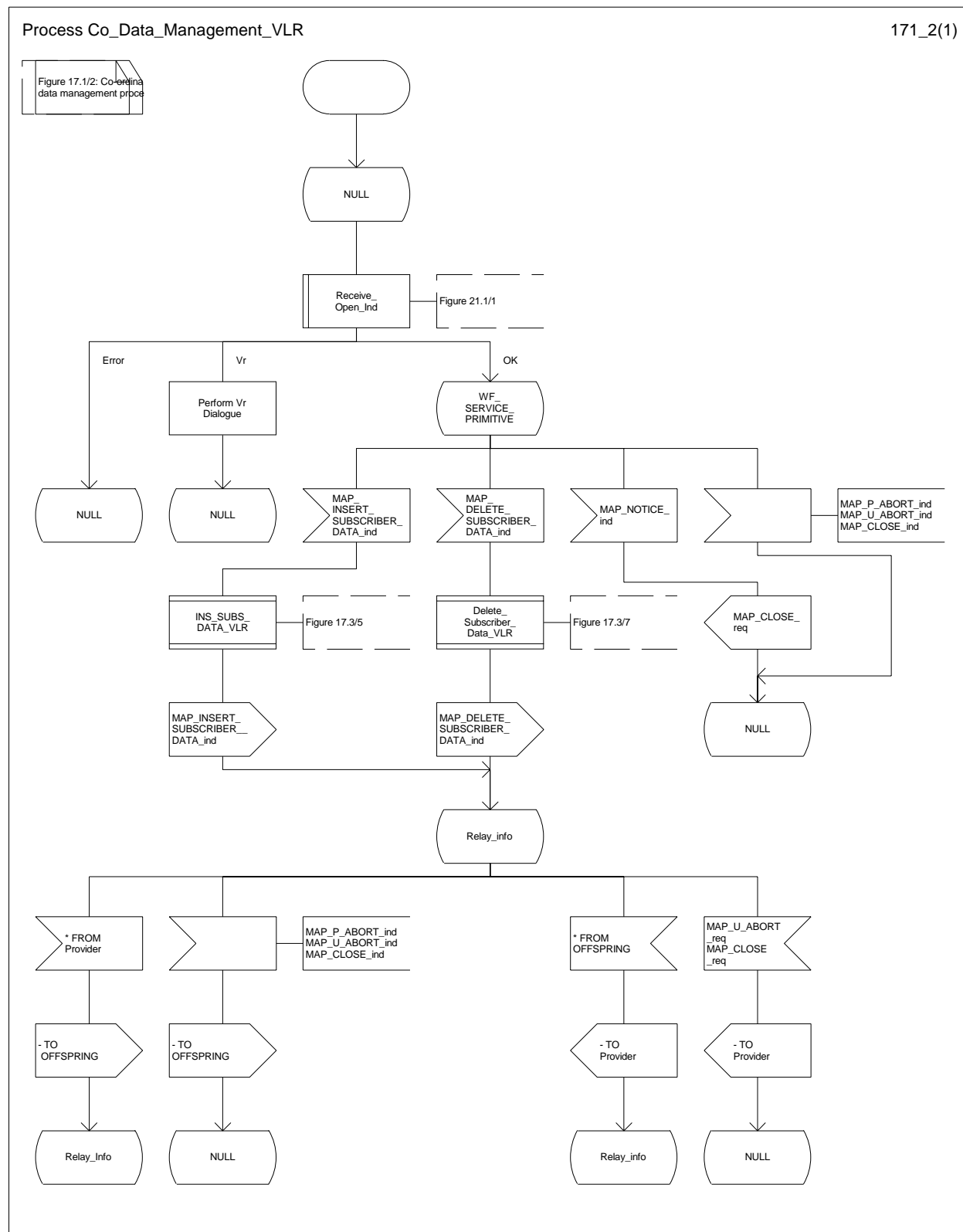
The MAP_OPEN indication opens the dialogue for the stand-alone subscriber data management procedure when the application context subscriberDataManagementContex is received. If that service is successful, the Co-ordinator can receive the first service primitive from the MAP_PM. Depending on the received 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 17.1/2.

Figure 17.1/2: Process Co_Data_Management_VLR

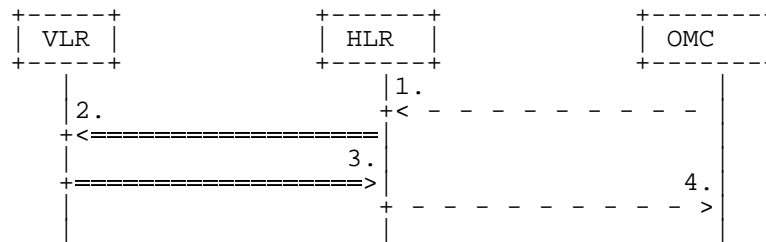


17.2 Tracing procedures

Three type of tracing procedures exist:

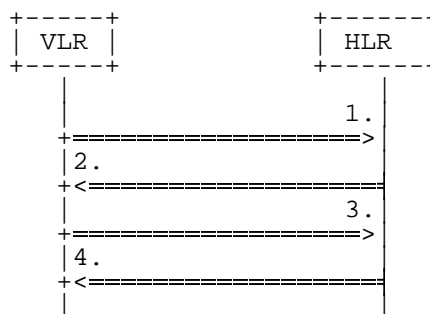
- i) Subscriber tracing management procedures;
- ii) Subscriber tracing procedures;
- iii) Event tracing procedures.

The subscriber tracing management procedures are used for management of the status and the type of the tracing. The subscriber tracing activation procedure is used at location updating or data restoration when the trace mode of a subscriber is set active in the HLR or, as a stand alone procedure, when the subscriber is already registered and the trace mode becomes active in the HLR. The procedures for providing a trace request to the VLR are shown in figures 17.2/1 and 17.2/2.



- 1) Subscriber Tracing Activation
- 2) MAP_ACTIVATE_TRACE_MODE
- 3) MAP_ACTIVATE_TRACE_MODE_ACK
- 4) Subscriber Tracing Activation Accepted

Figure 17.2/1: Stand alone subscriber tracing activation procedure

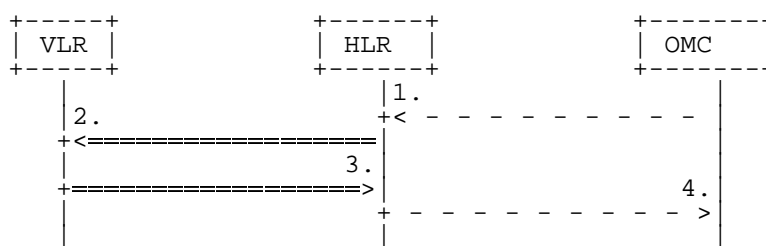


- 1) MAP_UPDATE_LOCATION or MAP_RESTORE_DATA
- 2) MAP_ACTIVATE_TRACE_MODE
- 3) MAP_ACTIVATE_TRACE_MODE_ACK
- 4) MAP_UPDATE_LOCATION_ACK or MAP_RESTORE_DATA_ACK

Figure 17.2/2: Subscriber tracing activation procedure at location updating or data restoration

The HLR sends the trace request (IMSI, trace reference, trace type and identity of the OMC) to the VLR in a MAP_ACTIVATE_TRACE_MODE request. The receipt of this primitive is acknowledged. The acknowledge primitive will indicate that the trace request is accepted by the VLR. If the request is not accepted, the reason will be reported to the HLR.

The subscriber tracing deactivation procedure is used when the trace request of a subscriber is to be cancelled in the VLR. The procedure is shown in figure 17.2/3.

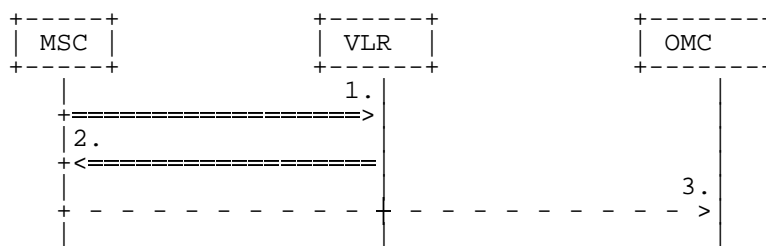


- 1) Subscriber Tracing Deactivation
- 2) MAP_DEACTIVATE_TRACE_MODE
- 3) MAP_DEACTIVATE_TRACE_MODE_ACK
- 4) Subscriber Tracing Deactivation Accepted

Figure 17.2/3: Subscriber tracing deactivation procedure

The HLR sends a MAP_DEACTIVATE_TRACE_MODE request to the VLR. The VLR will acknowledge the deactivation. The acknowledge primitive will indicate that the trace request has been deleted by the VLR. If the deactivation is not accepted, the reason will be reported to the HLR.

The subscriber tracing procedures are used when the VLR detects any subscriber related activity for which the trace mode is activated, e.g. receives the MAP_PROCESS_ACCESS_REQUEST indication. The procedure is shown in figure 17.2/4.



- 1) MAP_PROCESS_ACCESS_REQUEST, MAP_UPDATE_LOCATION_AREA, MAP_TRACE_SUBSCRIBER_ACTIVITY
- 2) Subscriber tracing information
- 3) Subscriber tracing information

Figure 17.2/4: Subscriber tracing procedure in the servicing MSC

The VLR will generate the MAP_TRACE_SUBSCRIBER_ACTIVITY indication. The receiving MSC will send the trace record to the OMC.

[Figure numbers 17.2/5 and 17.2/6 are spare.]

17.2.1 Procedures in the HLR

17.2.1.1 Subscriber tracing activation procedure

When receiving the subscriber tracing mode activation command for a subscriber from the OMC, the HLR will activate tracing, if the subscriber is known and registered in the HLR and the subscriber is roaming in the home PLMN area. The MAP_ACTIVATE_TRACE_MODE request is sent to the VLR where the subscriber is registered.

If the MAP_ACTIVATE_TRACE_MODE confirmation is received indicating an error situation, the errors are mapped to the OMC interface. The activation request may also be repeated; the number of repeat attempts and the time in between are HLR operator options, depending on the error returned by the VLR.

If the subscriber is known in the HLR, but is deregistered or roaming outside the home PLMN area, the subscriber tracing status is activated in the HLR, but the VLR is not updated.

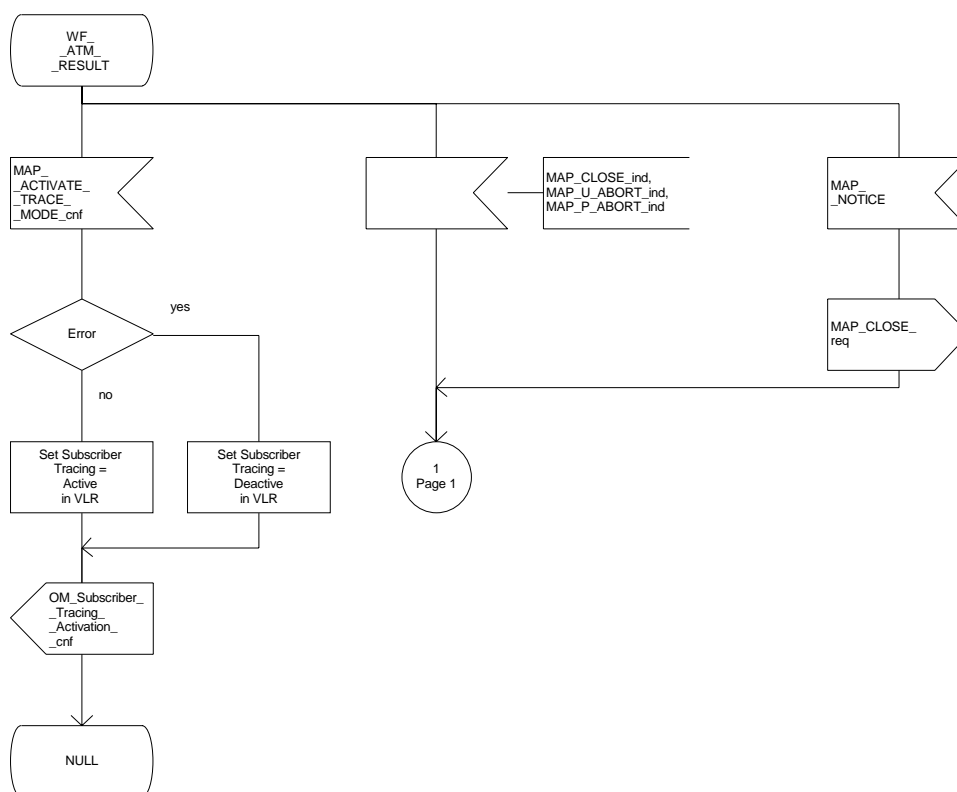
When receiving a request for location updating or data restoration while the subscriber trace mode is active, the macro Control_Tracing_HLR (see figure 21.9/4) shall be initiated by the location updating process in the HLR.

The subscriber tracing activation process in the HLR is shown in figure 17.2/7.

Figure 17.2/7 (sheet 1 of 2): Process ATM_HLR

Figure 17.2/7 (sheet 2 of 2): Process ATM_HLR

Figure 17.2/7: The subscriber tracing activation process in the HLR



17.2.1.2 Subscriber tracing deactivation procedure

When receiving the subscriber trace mode deactivation command for a subscriber from the OMC, the HLR will send the MAP_DEACTIVATE_TRACE_MODE request to the VLR where the subscriber is registered, if the trace mode activation has been carried out. The subscriber tracing in HLR is set to a deactive state.

If the operation is successful, the HLR will set the subscriber tracing in VLR to a deactive state.

If the MAP_DEACTIVATE_TRACE_MODE confirmation is received indicating an error situation, the errors are mapped to the OMC interface. The deactivation request may be also repeated; the number of repeat attempts and the time in between are HLR operator options, depending on the error returned by the VLR.

The subscriber tracing deactivation procedure is shown in figure 17.2/8.

Figure 17.2/8 (sheet 1 of 2): Process DTM_HLR

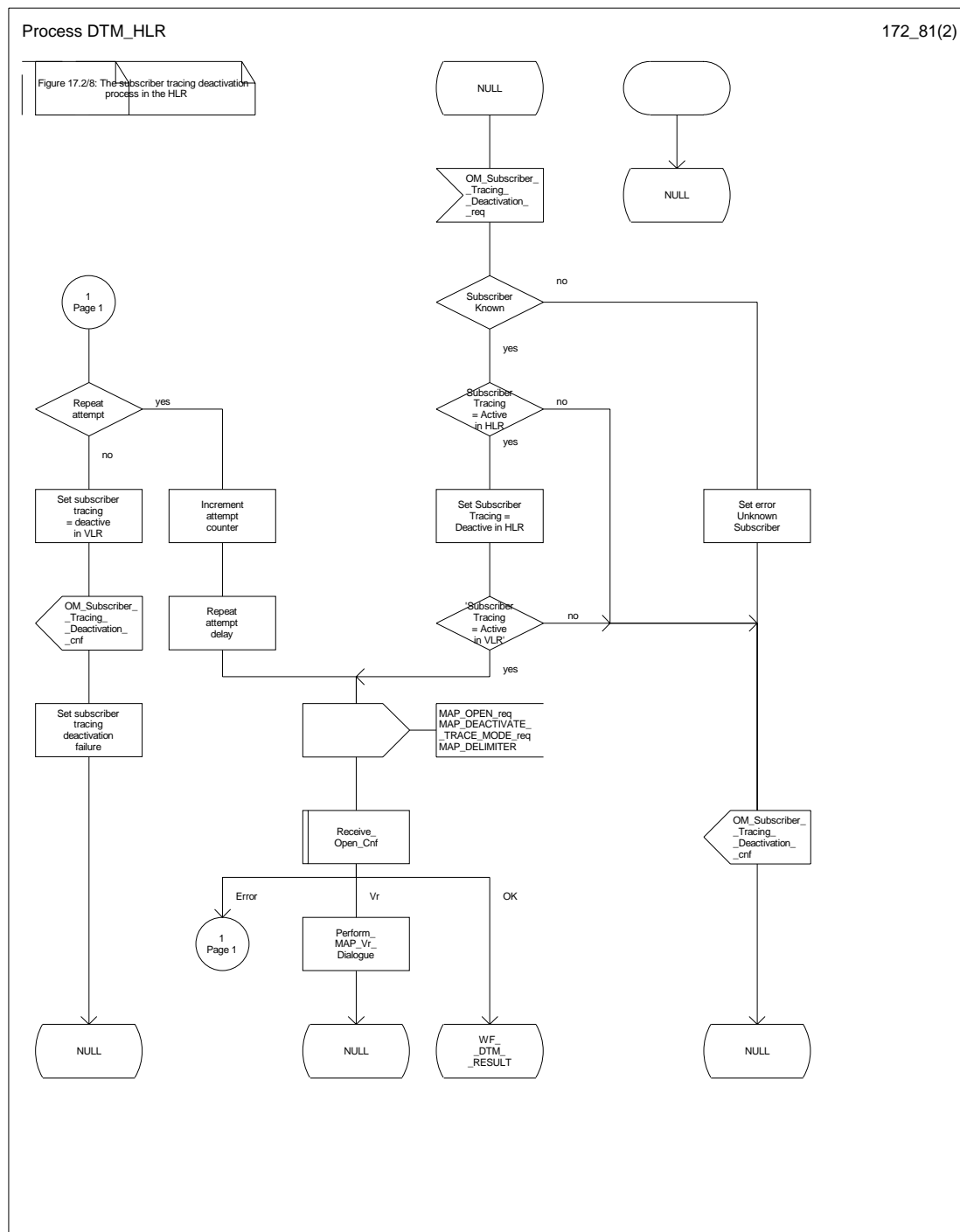
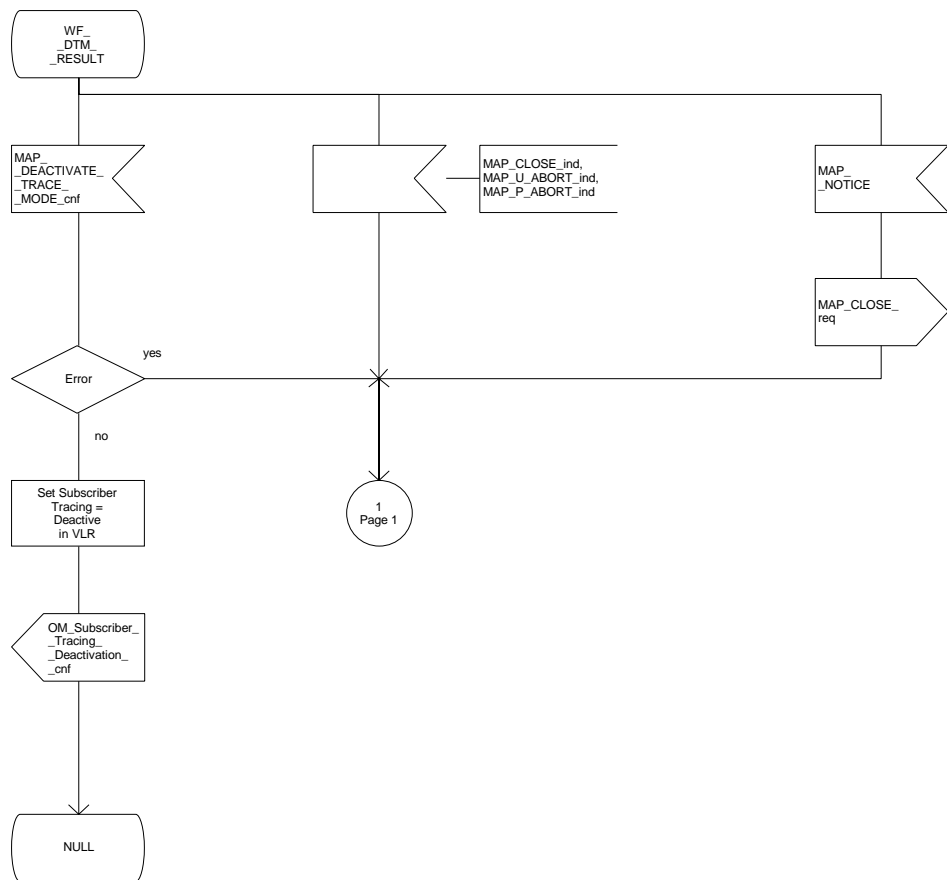


Figure 17.2/8 (sheet 2 of 2): Process DTM_HLR

Figure 17.2/8: The subscriber tracing deactivation process in the HLR



17.2.2 Procedures in the VLR

The VLR is involved in the following tracing procedures:

- i) Subscriber tracing activation procedure;
- ii) Subscriber tracing deactivation procedure;
- iii) Subscriber tracing procedure.

17.2.2.1 Subscriber tracing activation procedure

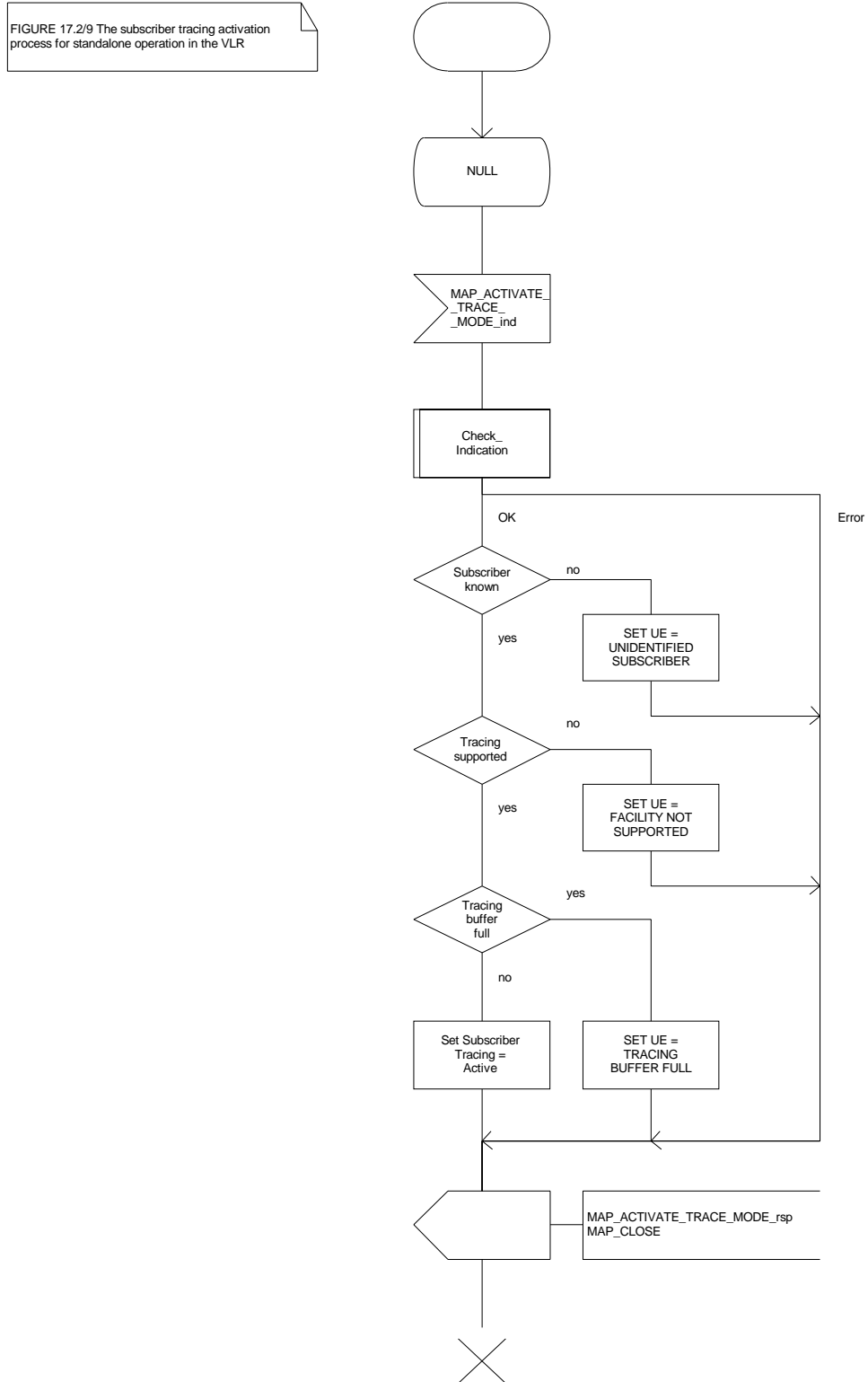
When receiving a MAP_ACTIVATE_TRACE_MODE indication, the VLR will check the parameters and data in the primitive. Data errors are reported as an unexpected data value error or as a data missing error depending on the nature of the error.

If the subscriber is known, the tracing facility is supported and the tracing capacity is not exceeded, the successful report is sent in the MAP_ACTIVATE_TRACE_MODE response primitive.

The MAP_ACTIVATE_TRACE_MODE indication primitive may be received during a location updating or data restoration procedure, so the location updating or restore data process shall use the macro Activate_Tracing_VLR (see figure 21.9/3).

The subscriber tracing activation process in the VLR is shown in figure 17.2/9.

Figure 17.2/9: Process ATM_VLR_Standalone



17.2.2.2 Subscriber tracing deactivation procedure

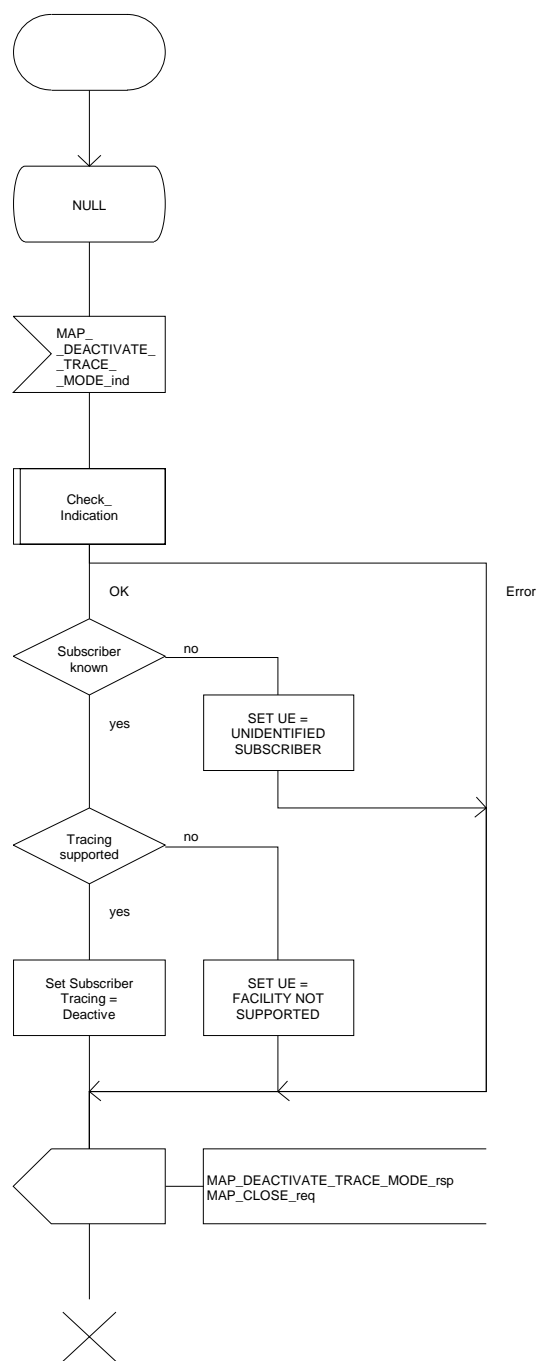
When receiving a MAP_DEACTIVATE_TRACE_MODE indication, the VLR will check the parameters and data in the primitive. Data errors are reported as an unexpected data value error or as a data missing error depending on the nature of the error.

If the subscriber is known and the tracing facility is supported, the successful report is sent in the MAP_DEACTIVATE_TRACE_MODE response primitive.

The subscriber tracing deactivation procedure in the VLR is shown in figure 17.2/10.

Figure 17.2/10: Process DTM_VLR_Standalone

Figure 17.2/10: The subscriber tracing deactivation process in the VLR



17.2.2.3 Subscriber tracing procedure

When the VLR receives a MAP_PROCESS_ACCESS_REQUEST 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 21.9/2.

17.2.3 Procedures in the MSC

The MSC is involved in the following tracing procedure:

- i) Subscriber tracing procedure.

17.2.3.1 Subscriber tracing procedure

When receiving the MAP_TRACE_SUBSCRIBER_ACTIVITY indication from the VLR, the MSC stores trace reference, trace type and the identity of the OMC in charge of the trace, and the MSC starts to collect the trace information. The MSC will send the trace record to the OMC.

The macro Trace_Subscriber_Activity_MSC is shown in figure 21.9/1.

17.3 Subscriber data management procedures

Two types of subscriber data management procedures exist in the Mobile Application Part

- i) Subscriber Deletion;
- ii) Subscriber Data Modification.

No requirements have been identified for the Subscriber creation and subscriber data interrogation procedures.

The subscriber deletion and subscriber data modification procedures are initiated by the OMC (see figures 17.3/1 and 17.3/2).

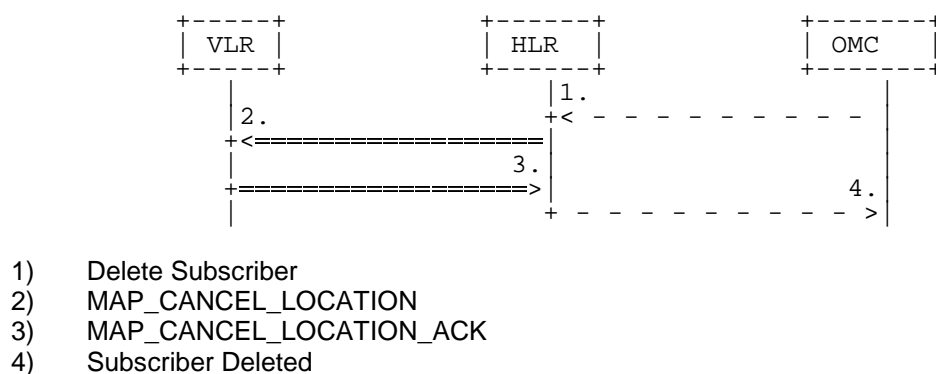


Figure 17.3/1: Subscriber deletion procedure

In the subscriber deletion procedure the subscriber data should be removed from the VLR and from the HLR. The HLR uses the MAP_CANCEL_LOCATION service.

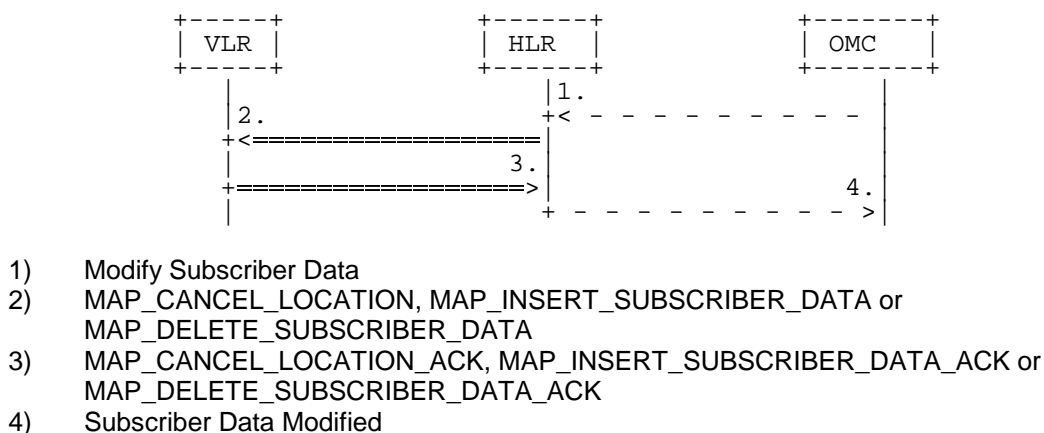


Figure 17.3/2: Subscriber data modification procedure

In the subscriber data modification procedure the subscriber data is modified in the HLR and when necessary also in the VLR. The HLR initiates either the MAP_INSERT_SUBSCRIBER_DATA, MAP_DELETE_SUBSCRIBER_DATA or MAP_CANCEL_LOCATION service depending on the modified data.

17.3.1 Procedures in the HLR

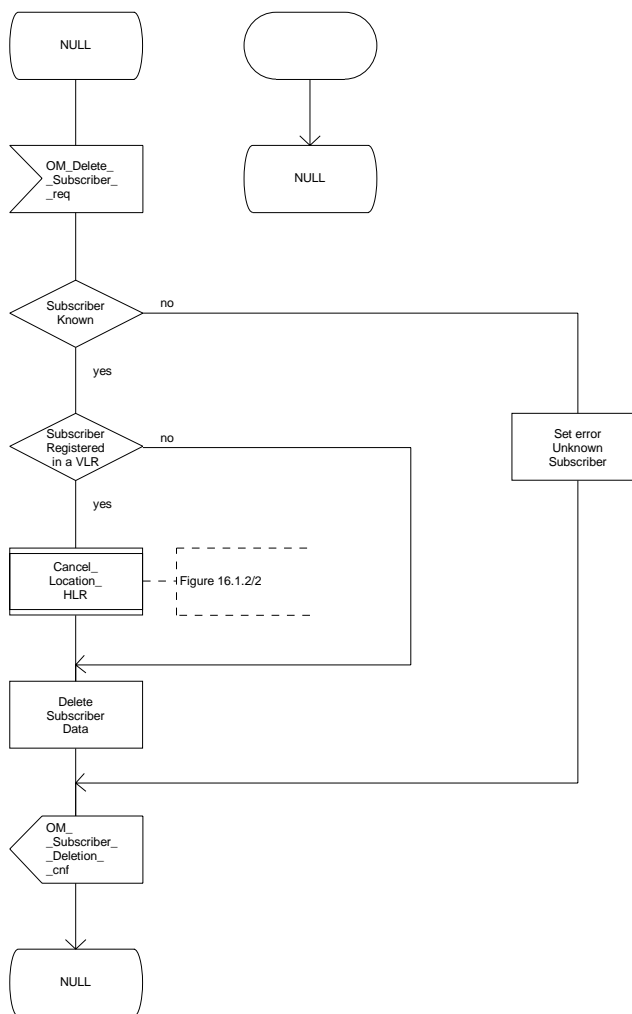
17.3.1.1 Subscriber deletion procedure

When the subscriber deletion request is received from the OMC, the HLR shall delete the subscriber data from the HLR and initiate the MAP_CANCEL_LOCATION request to the VLR where the subscriber is registered.

The subscriber deletion procedure in the HLR is shown in the figure 17.3/3.

Figure 17.3/3: Process Delete_Subscriber_HLR

Figure 17.3/3: The subscriber deletion process in the HLR



17.3.1.2 **Subscriber data modification procedure**

The OMC can modify the subscriber data in several different ways. The modifications can be categorized in five groups:

- a) HLR internal modification, no effect in the VLR;
- b) data shall be modified both in the HLR and VLR;
- c) withdrawal of a basic service or a supplementary service;
- d) modification affects on the roaming of the subscriber and the subscriber shall be removed from the VLR data base;
- e) authentication algorithm or authentication key of the subscriber is modified.

In case "b" the MAP_INSERT_SUBSCRIBER_DATA service is initiated in the HLR.

In case "c" the MAP_DELETE_SUBSCRIBER_DATA service is initiated in the HLR.

In cases "d" and "e" the MAP_CANCEL_LOCATION service is initiated in the HLR.

If the result of a primitive received from the VLR is unsuccessful, the HLR may initiate re-attempts; the number of repeat attempts and the time in between are HLR operator options, depending on the error returned by the VLR.

The subscriber data modification procedure in the HLR is shown in the figures 17.3/4, 17.3/5 and 21.7/2.

Figure 17.3/4: Process Modify_Data_HLR

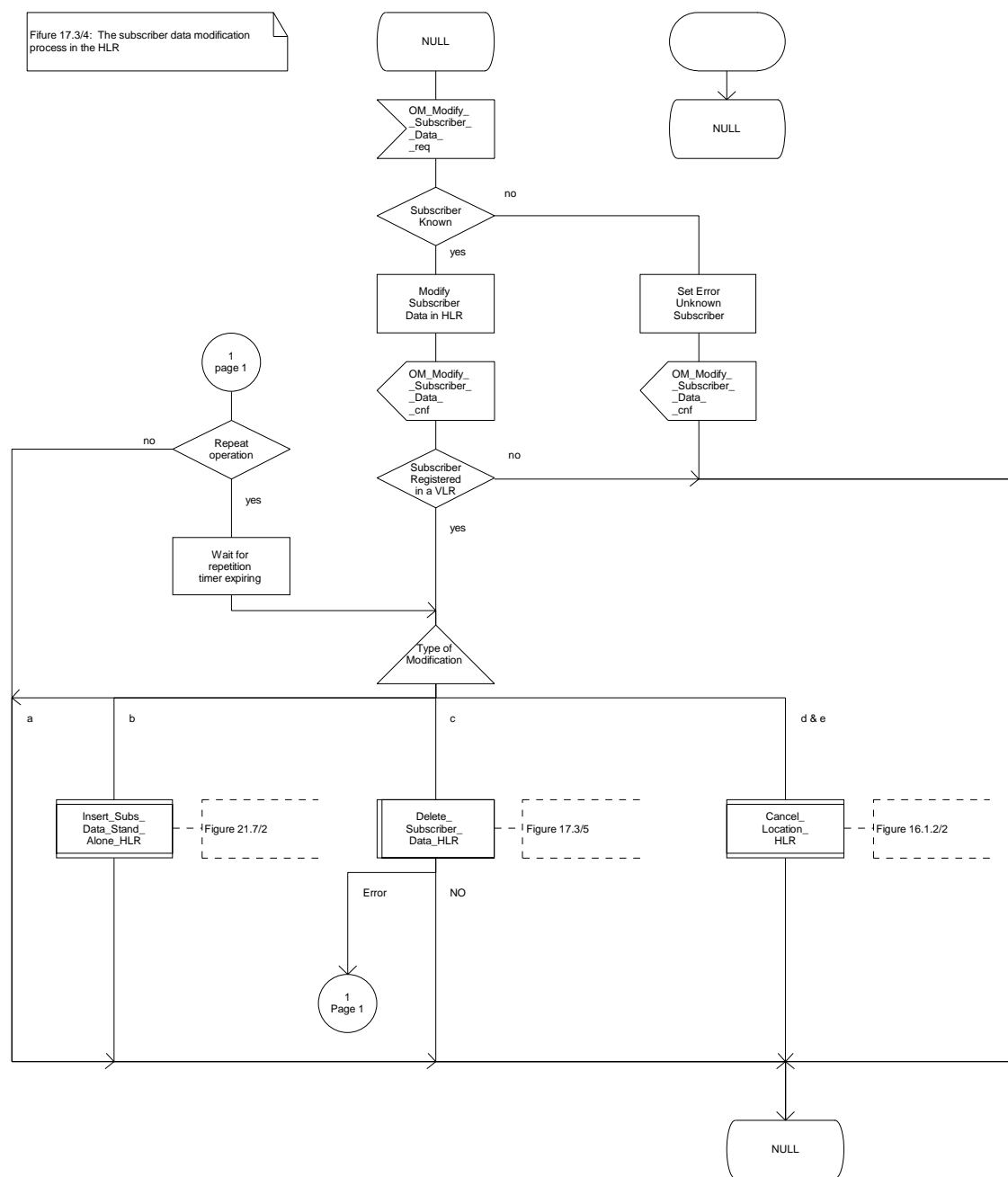
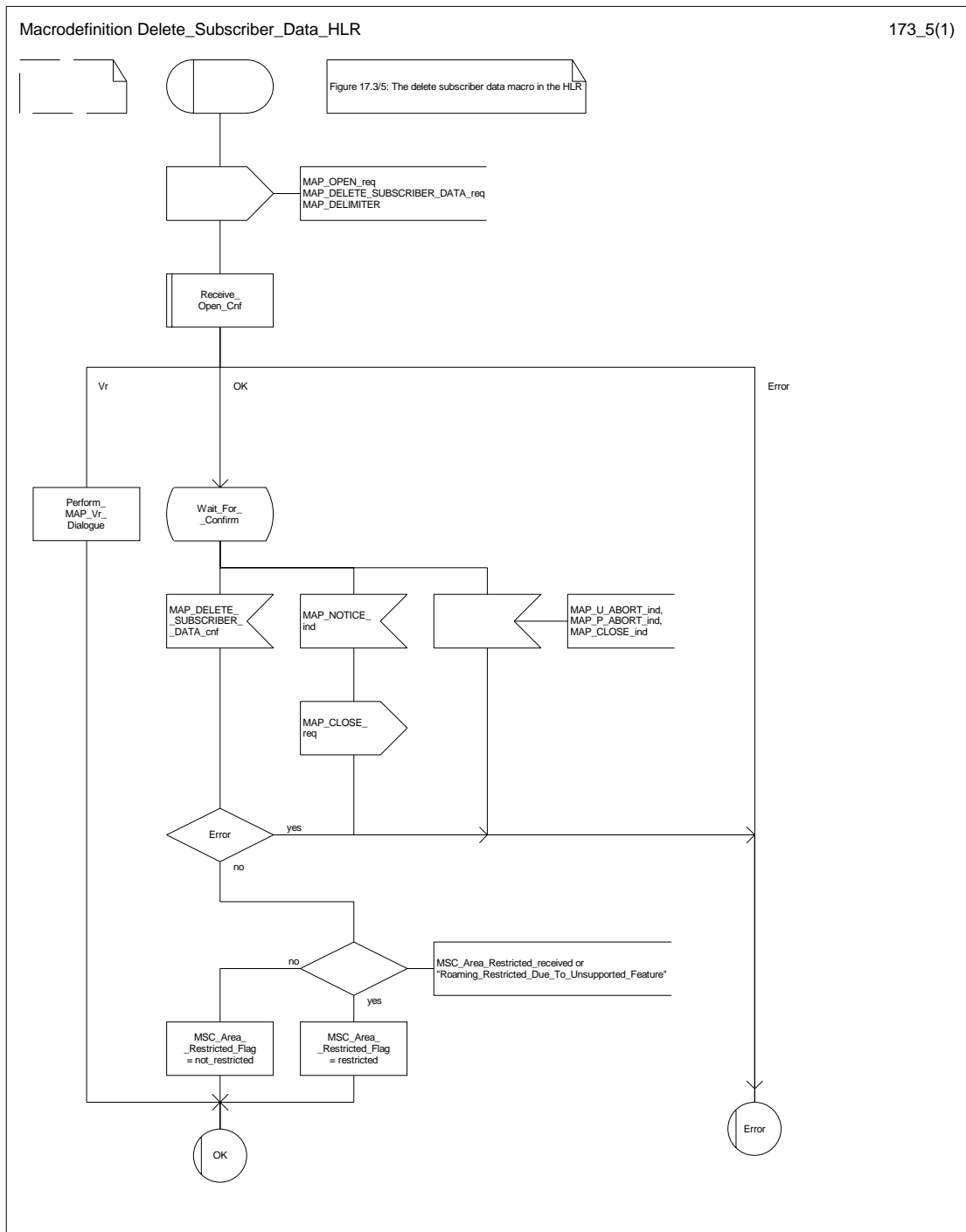


Figure 17.3/5: Macro Delete_Subscriber_Data_HLR



17.3.2 Procedures in the VLR

17.3.2.1 Subscriber deletion procedure

The subscriber deletion procedure in the VLR is described in the subclause 16.1.

17.3.2.2 Subscriber data modification procedure

When receiving either the MAP_INSERT_SUBSCRIBER_DATA indication or the MAP_DELETE_SUBSCRIBER_DATA indication, the VLR check the parameters and data in the primitive. Data errors are reported as an unexpected data value error or a data missing error depending on the nature of the error.

After receiving the first MAP_INSERT_SUBSCRIBER_DATA indication, the VLR will check the IMSI that is included in the primitive. If the IMSI is unknown, the error "Unidentified subscriber" is returned.

If the VLR does not support received basic or supplementary services or the network feature Operator Determined Barring, or there is a problem with Regional Subscription Data then it reports it to the HLR.

If the entire MSC area is restricted due to regional subscription, this is reported to the HLR.

If the updating of the subscriber data is not possible, the VLR will initiate the MAP_U_ABORT request primitive. If the updating is successful, the MAP_CLOSE indication is received from the HLR.

The subscriber data modification procedure in the VLR is shown in the figures 17.3/6, 17.3/7 and 21.7/1.

Figure 17.3/6: Process INS_SUBS_DATA_VLR

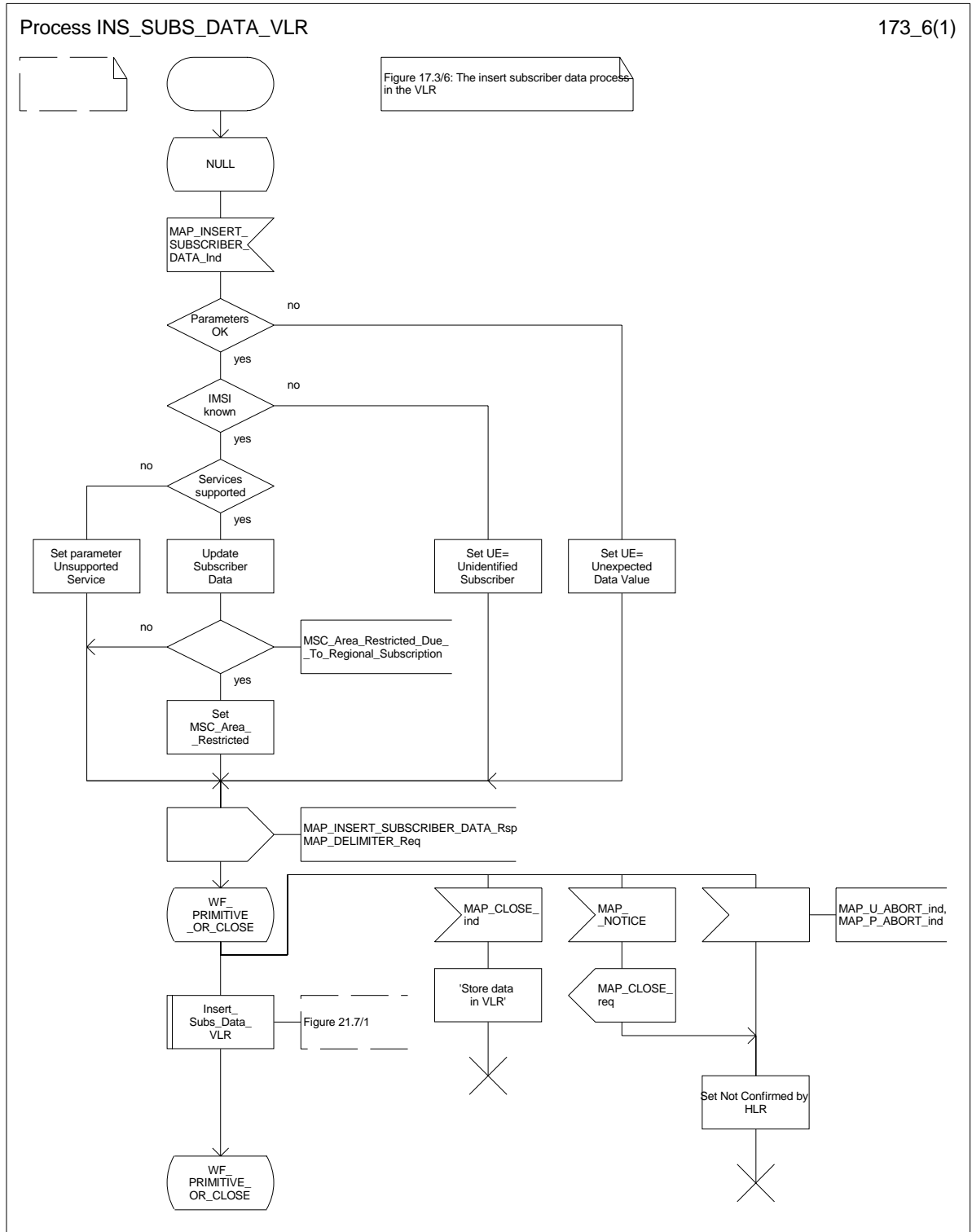
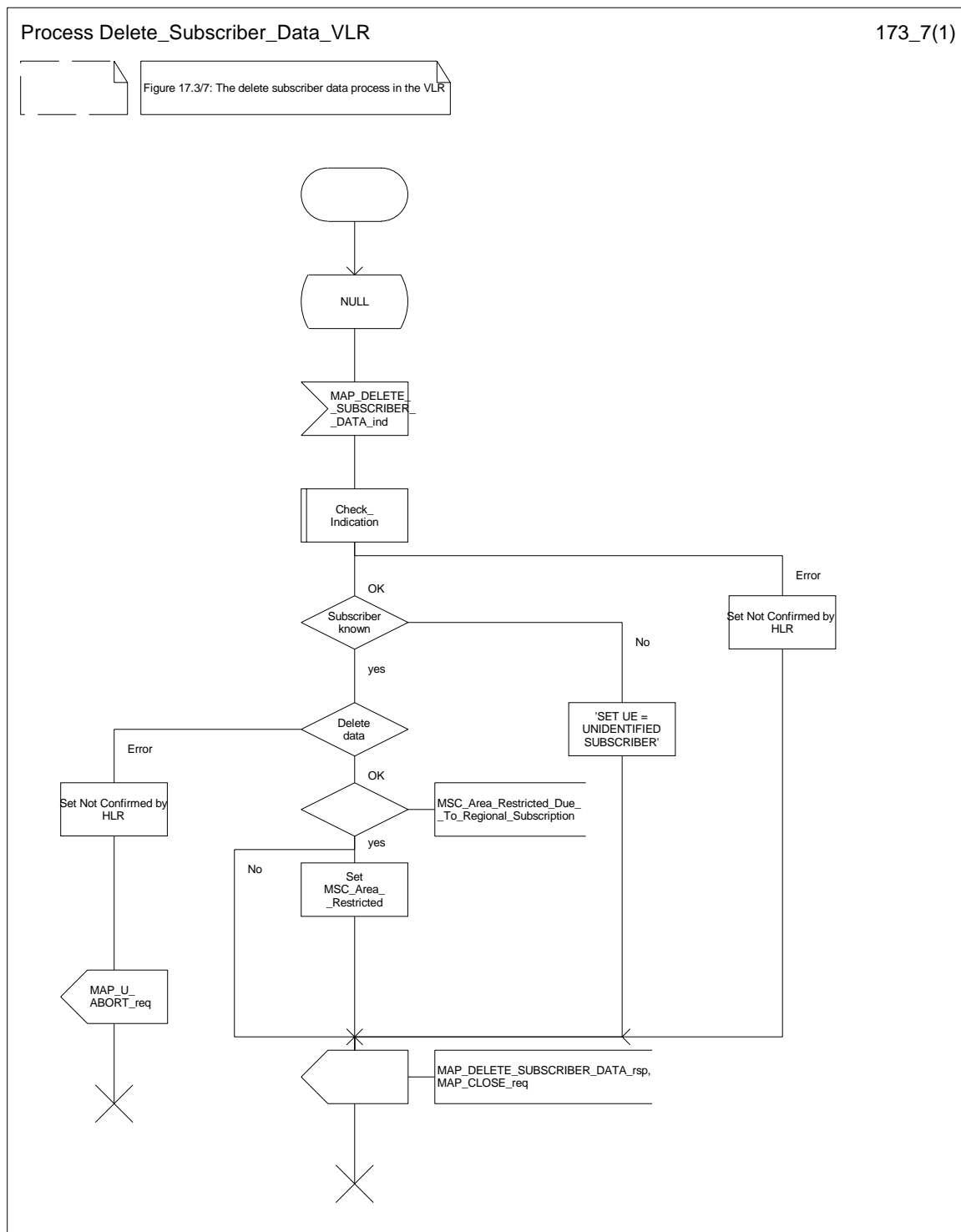


Figure 17.3/7: Process Delete_Subscriber_Data_VLR



17.4 Subscriber Identity procedure

In the subscriber identity procedure the IMSI of the subscriber is retrieved from the HLR. The procedure is shown in figure 17.4/1.

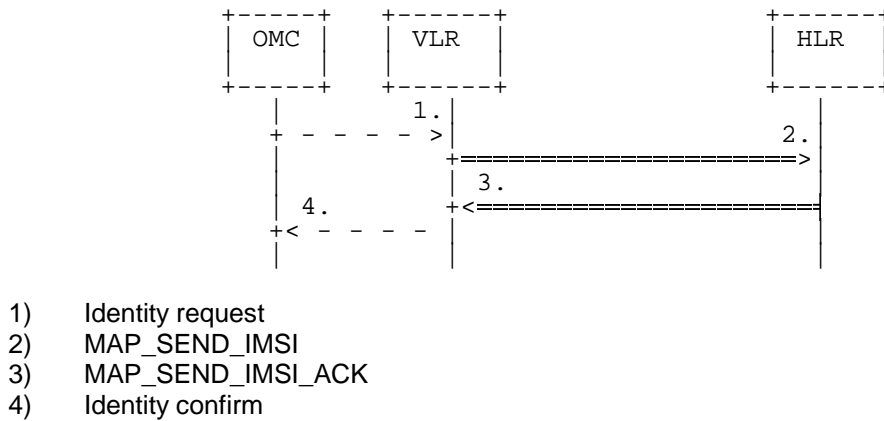


Figure 17.4/1: The subscriber identity procedure

17.4.1 Subscriber identity procedure in the HLR

Opening of the dialogue is described in the macro `Receive_Open_Ind` in subclause 21.1, with outcomes:

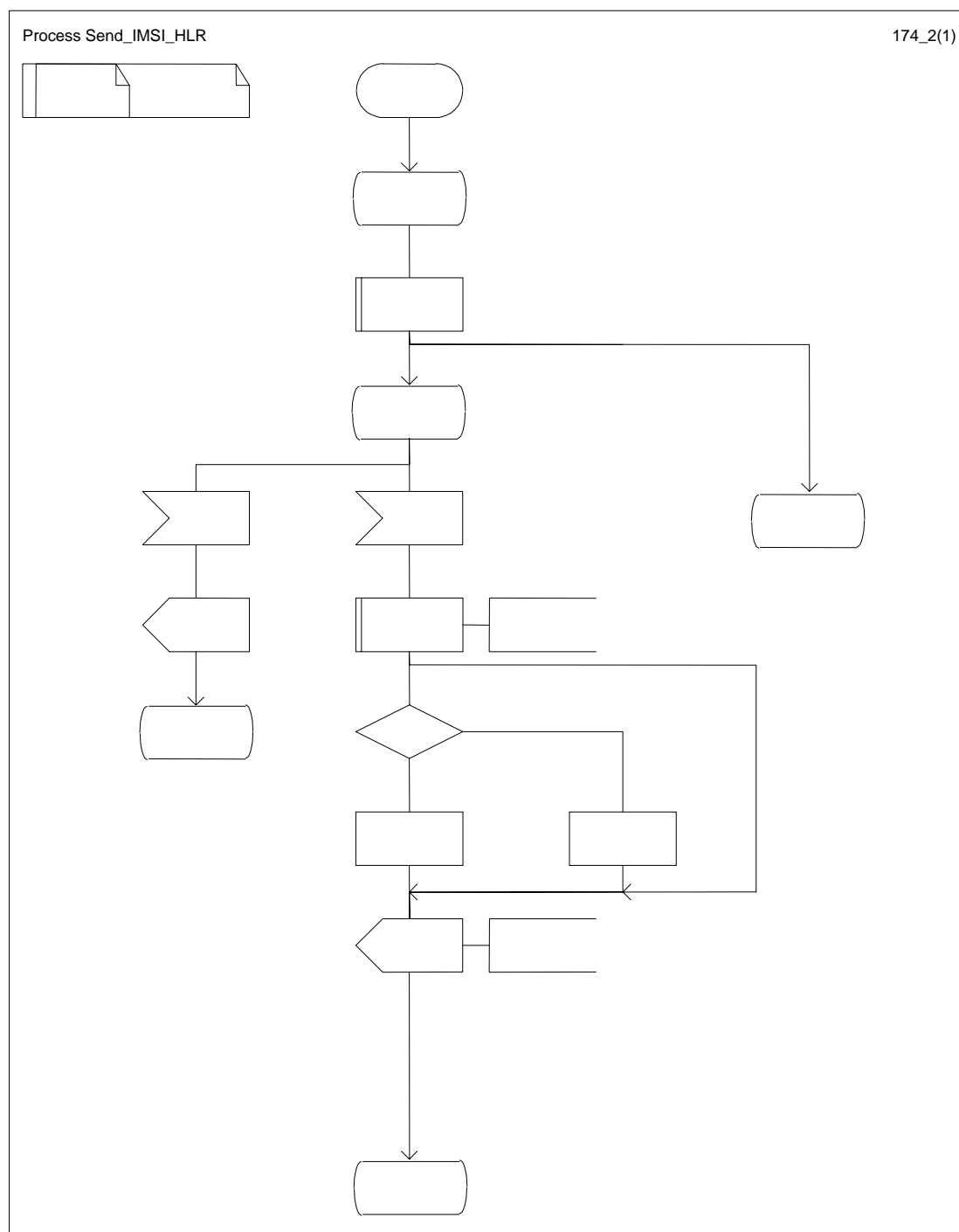
- procedure termination; or
- dialogue acceptance, with proceeding as below.

When receiving the `MAP_SEND_IMSI` indication, the HLR will check the parameters and data in the primitive. Data errors are reported as an unexpected data value error or a data missing error depending on the nature of the error.

If the subscriber is known in the HLR, the IMSI is fetched from the database and sent to the VLR. If the MSISDN cannot be identified, unknown subscriber indication is passed to the VLR.

The subscriber identity procedure in the HLR is shown in figure 17.4/2.

Figure 17.4/2: Process Send_IMSI_HLR

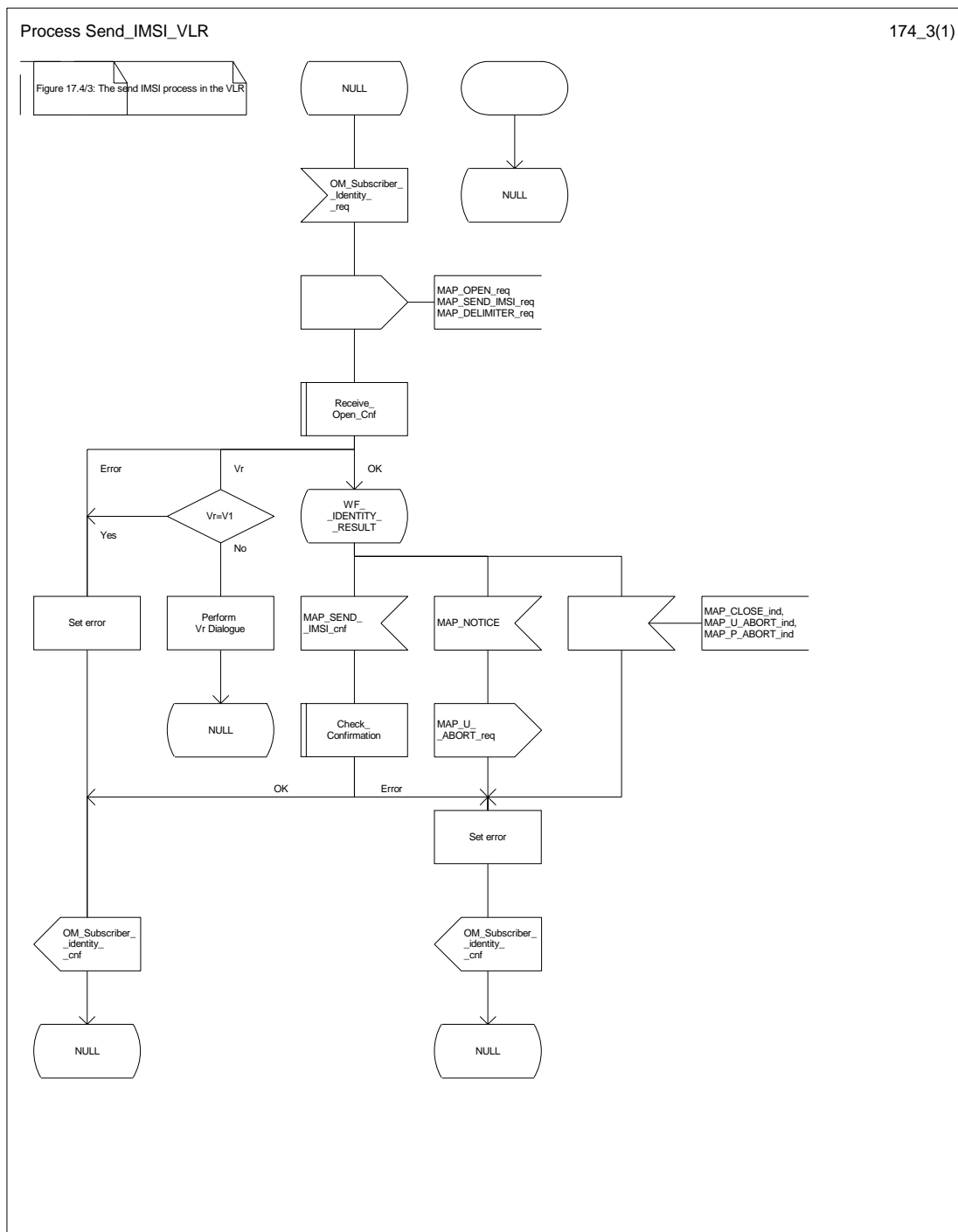


17.4.2 Subscriber identity procedure in the VLR

When the IMSI request is received from the OMC, the VLR will send the MAP_SEND_IMSI request to the HLR. The contents of the response is sent to the OMC.

The subscriber identity procedure in the VLR is shown in figure 17.4/3.

Figure 17.4/3: Process Send_IMSI_VLR



18 Call handling procedures

18.1 General

The MAP call handling procedures are used to retrieve routing information to handle a mobile terminating call and to transfer control of a call back to the GMSC if the call is to be forwarded.

The procedures to handle a mobile originating call and a mobile terminating call after the call has arrived at the destination MSC do not require any signalling over a MAP interface. These procedures are specified in GSM 03.18 [97].

The stage 2 specification for the retrieval of routing information to handle a mobile terminating call is in GSM 03.18 [97]; modifications to this procedure for CAMEL are specified in GSM 03.78 [98], for optimal routing of a basic mobile-to-mobile call in GSM 03.79 [99]. The interworking between the MAP signalling procedures and the call handling procedures for each entity (GMSC, HLR and VLR) is shown by the transfer of signals between these procedures.

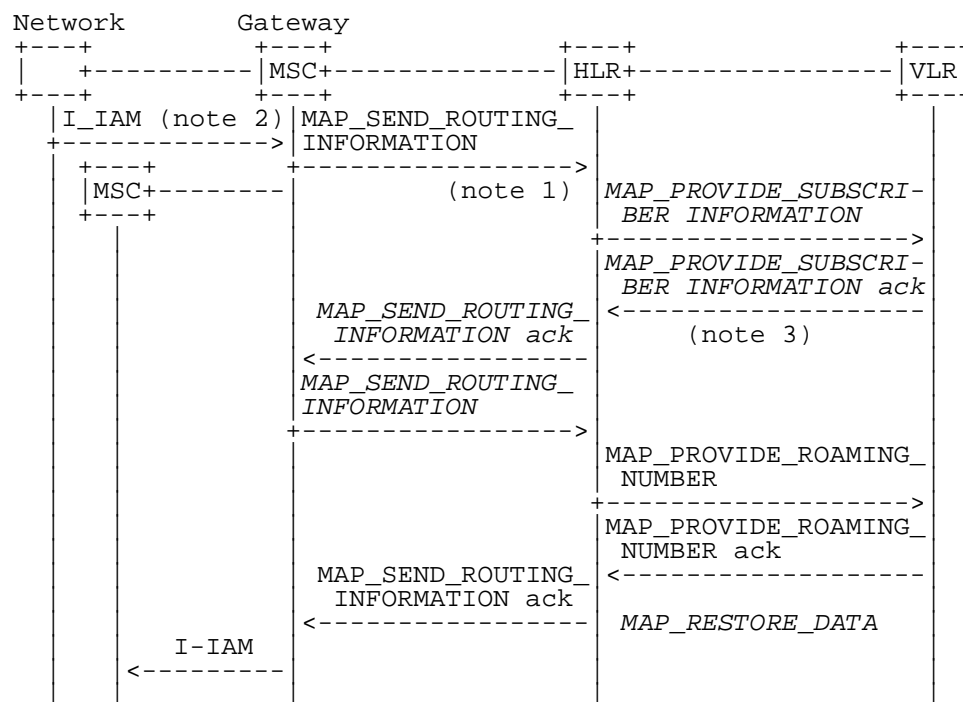
The stage 2 specification for the transfer of control of a call back to the GMSC if the call is to be forwarded is in GSM 03.79 [99]. The interworking between the MAP signalling procedures and the call handling procedures for each entity (VMSC and GMSC) is shown by the transfer of signals between these procedures.

The interworking between the call handling procedures and signalling protocols other than MAP is shown in GSM 03.18, GSM 03.78 and GSM 03.79.

18.2 Retrieval of routing information

18.2.1 General

The message flows for successful retrieval of routing information for a mobile terminating call are shown in figure 18.2/1 (mobile terminating call which has not been optimally routed) and 18.2/2 (mobile-to-mobile call which has been optimally routed).



Notes:

xxx = Optional Procedure

NOTE 1: This service may also be used by an ISDN exchange for obtaining routing information from the HLR.

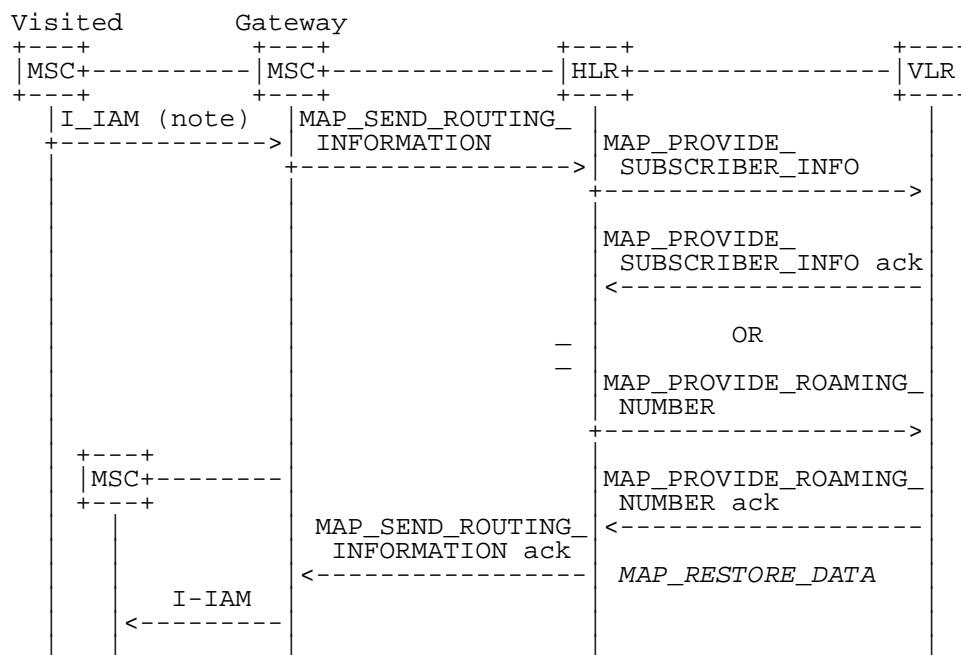
NOTE 2: TUP or ISUP may be used in signalling between MSCs, depending on the network type between the MSCs. For further details on the TUP and ISUP procedures refer to the following ITU-T Recommendations and ETSI specification:

Q.721-725 - Telephone User Part (TUP);

ETS 300 356-1 - Integrated Services Digital Network (ISDN); Signalling System No.7; ISDN User Part (ISUP) version 2 for the international interface; Part 1: Basic services.

NOTE 3: As a network operator option, the HLR sends MAP_PROVIDE_SUBSCRIBER_INFORMATION to the VLR. For further details on the CAMEL procedures refer to GSM TS 03.78;

Figure 18.2/1: Message flow for retrieval of routing information (non-optimally routed call)



Notes:

xxx = Optional Procedure

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.

TUP or ISUP may be used in signalling between MSCs, depending on the network type between the MSCs. For further details on the TUP and ISUP procedures refer to the following CCITT Recommendations & ETSI specification:

Q.721-725 - Telephone User Part (TUP);
 ETS 300 356-1 - Integrated Services Digital Network (ISDN); Signalling System No.7;
 ISDN User Part (ISUP) version 2 for the international interface; Part 1: Basic services.

Figure 18.2/2: Message flow for retrieval of routeing information (optimally routed call)

The following MAP services are used to retrieve routing information:

MAP_SEND_ROUTING_INFORMATION	see subclause 8.1;
MAP_PROVIDE_ROAMING_NUMBER	see subclause 8.2;
MAP_PROVIDE_SUBSCRIBER_INFO	see subclause 6.11.2;
MAP_RESTORE_DATA	see subclause 6.10.3.

18.2.2 Process in the GMSC

The MAP process in the GMSC to retrieve routeing information for a mobile terminating call is shown in figure 18.2/3. The MAP process invokes macros not defined in this subclause; the definitions of these macros can be found as follows:

Receive_Open_Cnf	see subclause 21.1.2;
Check_Confirmation	see subclause 21.2.2.

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] or [96] 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 18.2/3 (sheet 1 of 2): Process SRI_GMSC

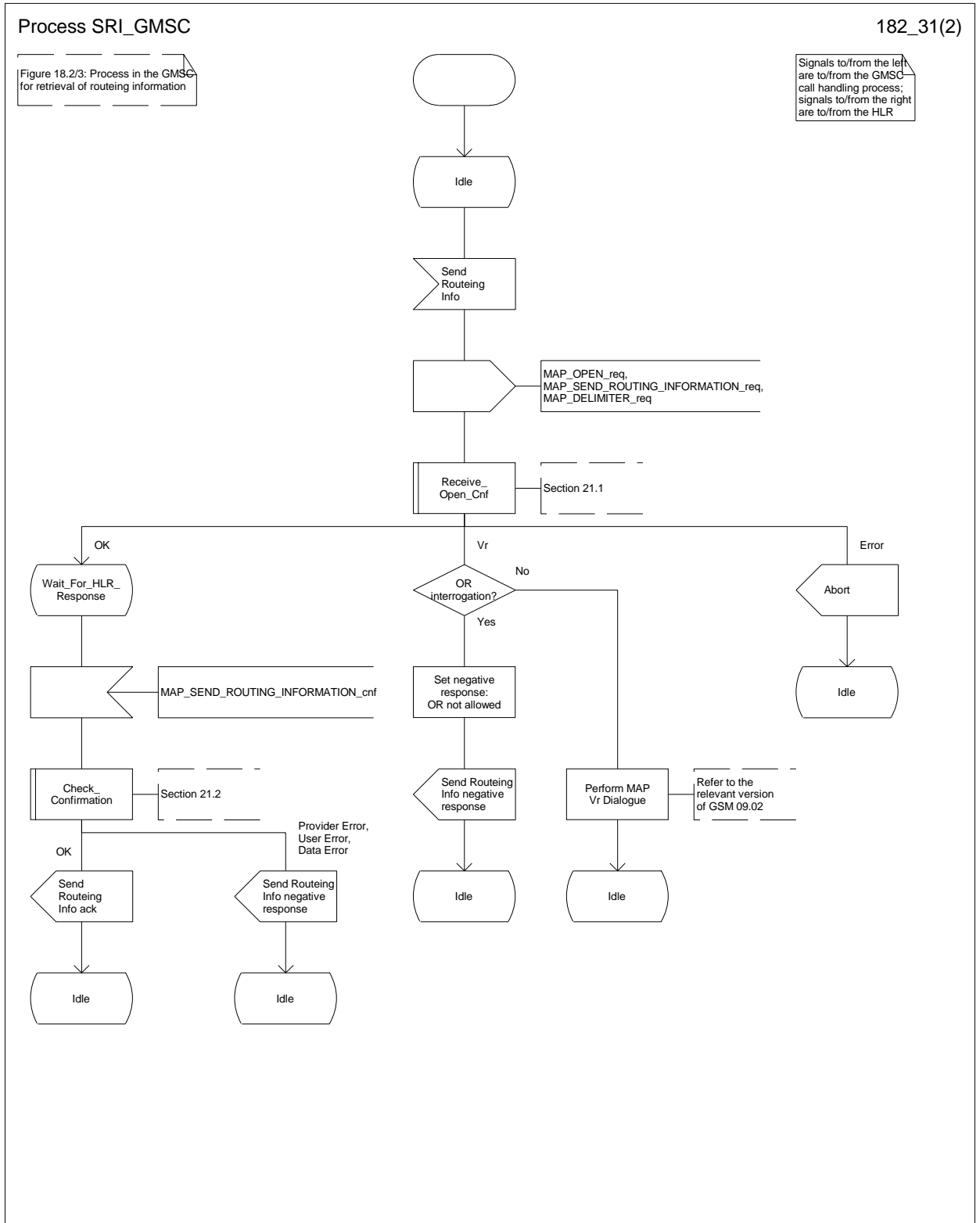
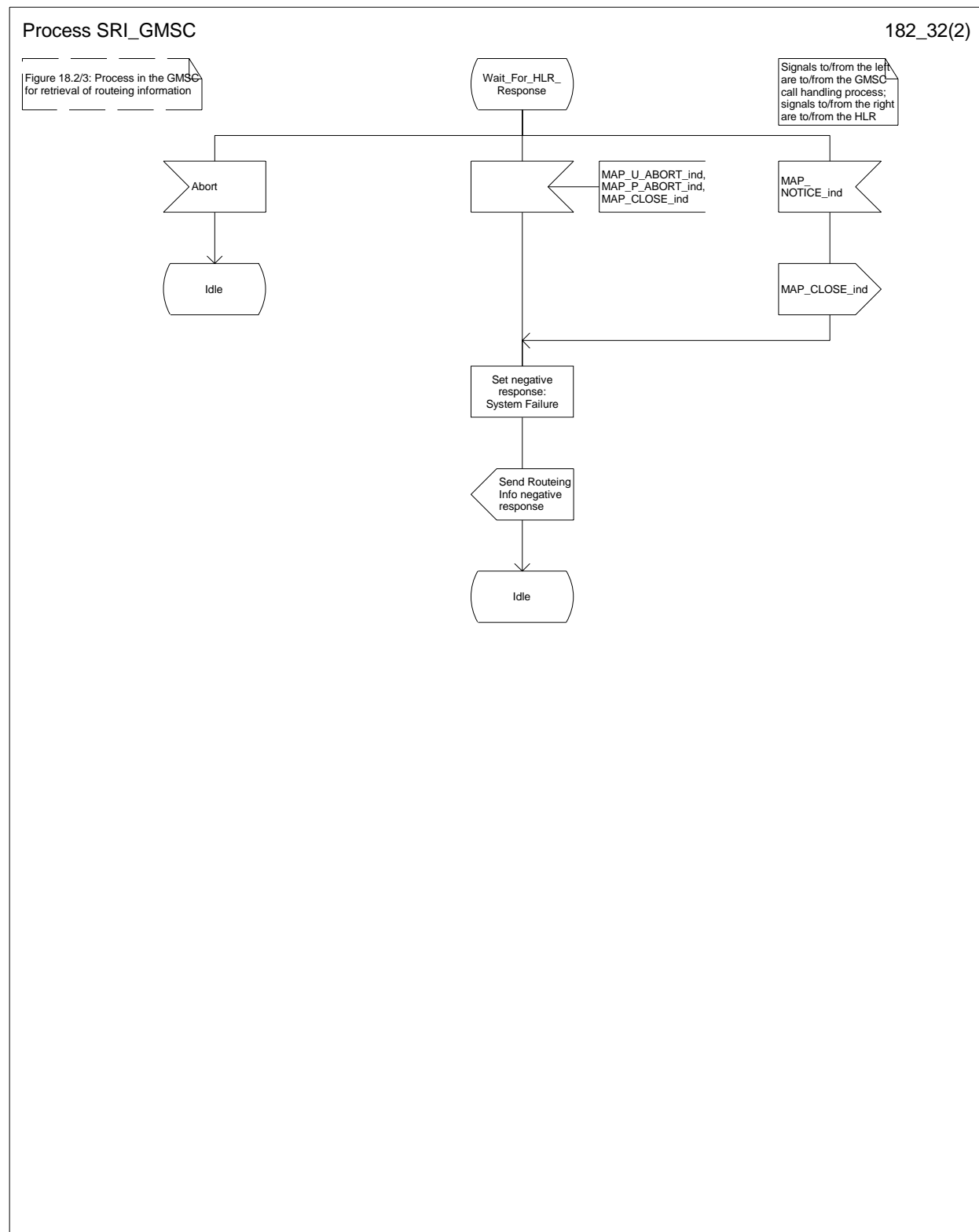


Figure 18.2/3 (sheet 2 of 2): Process SRI_GMSC



18.2.3 Procedures in the HLR

The MAP process in the HLR to retrieve routing information for a mobile terminating call is shown in figure 18.2/4. The MAP process invokes macros not defined in this subclause; the definitions of these macros can be found as follows:

Receive_Open_Ind	see subclause 21.1.1;
Receive_Open_Cnf	see subclause 21.1.2;
Check_Confirmation	see subclause 21.2.2.

Successful outcome

When the MAP process receives a MAP_OPEN indication with the application context locInfoRetrieval, it checks it by invoking the macro Receive_Open_Ind.

If the macro takes the OK exit, the MAP process waits for a service indication.

If a MAP_SEND_ROUTING_INFORMATION service indication is received, the MAP process sends a Send Routing Info request to the call handling process in the HLR, and waits for a response. The Send Routing Info request contains the parameters received in the MAP_SEND_ROUTING_INFORMATION service indication.

If the call handling process in the HLR returns a Send Routing Info ack, the MAP process constructs a MAP_SEND_ROUTING_INFORMATION service response containing the routing information contained in the Send Routing Info ack, constructs a MAP_CLOSE service request, sends them to the GMSC and returns to the idle state. If the MAP_SEND_ROUTING_INFORMATION response cannot be carried in a single TC-Result component, it is carried in one or more TC-Result-NL components (each sent in a TC-CONTINUE), followed by a TC-Result-L component in a TC-END message.

If the call handling process in the HLR returns a Provide Subscriber Info request, the MAP process requests a dialogue with the VLR whose identity is contained in the Provide Subscriber Info request by sending a MAP_OPEN service request, requests the subscriber status using a MAP_PROVIDE_SUBSCRIBER_INFO service request, and invokes the macro Receive_Open_Cnf to wait for the response to the dialogue opening request.

If the macro takes the OK exit, the MAP process waits for the response from the VLR.

If the MAP process receives a MAP_PROVIDE_SUBSCRIBER_INFO service confirm, it invokes the macro Check_Confirmation to check the content of the confirm.

If the Check_Confirmation macro takes the OK exit, the MAP process sends a Provide Subscriber Info ack containing the information received in the MAP_PROVIDE_SUBSCRIBER_INFO service confirm to the call handling process in the HLR, and waits for a response. The handling of the response from the call handling process in the HLR is described above.

If the MAP_PROVIDE_SUBSCRIBER_INFO service confirm contains a provider error or a data error, the MAP process sends a Provide Subscriber Info negative response indicating the type of error to the call handling process in the HLR, and waits for a response. The handling of the response from the call handling process in the HLR is described above.

NOTE: The 'User Error' exit from the macro Check_Confirmation is shown for formal completeness; the MAP_PROVIDE_SUBSCRIBER_INFO_cnf primitive cannot contain a user error.

If the call handling process in the HLR returns a Provide Roaming Number request, the MAP process requests a dialogue with the VLR whose identity is contained in the Provide Roaming Number request by sending a MAP_OPEN service request, requests a roaming number using a MAP_PROVIDE_ROAMING_NUMBER service request, and invokes the macro Receive_Open_Cnf to wait for the response to the dialogue opening request.

If the macro takes the OK exit, the MAP process waits for the response from the VLR.

If the MAP process receives a MAP_PROVIDE_ROAMING_NUMBER service confirm, it invokes the macro Check_Confirmation to check the content of the confirm.

If the Check_Confirmation macro takes the OK exit, the MAP process sends a Provide Roaming Number ack containing the MSRN received in the MAP_PROVIDE_ROAMING_NUMBER service confirm to the call handling process in the HLR, and waits for a response. The handling of the response from the call handling process in the HLR is described above.

If the MAP_PROVIDE_ROAMING_NUMBER service confirm contains a user error or a provider error, or the macro Check_Confirmation indicates that there is a data error, the MAP process sends a Provide Roaming Number negative response indicating the type of error to the call handling process in the HLR, and waits for a response. The handling of the response from the call handling process in the HLR is described above.

Negative response from HLR call handling process

If the call handling process in the HLR returns a negative response, either before or after a dialogue with the VLR to obtain a roaming number, the MAP process constructs a MAP_SEND_ROUTING_INFORMATION service response containing the appropriate error, constructs a MAP_CLOSE service request, sends them to the GMSC and returns to the idle state.

Earlier version MAP Provide Roaming Number dialogue with the VLR

If the macro Receive_Open_Cnf takes the Vr exit after the MAP process has requested opening of a Provide Roaming Number dialogue with the VLR, the MAP process checks whether this is an OR interrogation (indicated by the inclusion of the OR interrogation parameter in the MAP_PROVIDE_ROAMING_NUMBER service request).

If this is not an OR interrogation, the HLR performs the earlier version MAP dialogue as specified in [51] or [96], relays the result of the dialogue to the HLR call handling process, and waits for a response. The handling of the response from the call handling process in the HLR is described above.

If this is an OR interrogation, the MAP process sends a Provide Roaming Number negative response indicating OR not allowed to the call handling process in the HLR and waits for a response. The handling of the response from the call handling process in the HLR is described above.

Failure of Provide Subscriber Info dialogue with the VLR

If the Receive_Open_Cnf macro takes the Vr exit or the Error exit after the MAP process has requested opening of a Provide Subscriber Info dialogue with the VLR, the MAP process sends a Provide Subscriber Info negative response indicating system failure to the call handling process in the HLR, and waits for a response. The handling of the response from the call handling process in the HLR is described above.

Failure of Provide Roaming Number dialogue with the VLR

If the Receive_Open_Cnf macro takes the Error exit after the MAP process has requested opening of a Provide Roaming Number dialogue with the VLR, the MAP process sends a Provide Roaming Number negative response indicating system failure to the call handling process in the HLR, and waits for a response. The handling of the response from the call handling process in the HLR is described above.

If the MAP process receives a MAP_U_ABORT, a MAP_P_ABORT or a premature MAP_CLOSE from the MAP provider, it sends a Provide Roaming Number negative response indicating system failure to the call handling process in the HLR, and waits for a response. The handling of the response from the call handling process in the HLR is described above.

If the MAP process receives a MAP_NOTICE from the MAP provider, it returns a MAP_CLOSE request to the MAP provider, sends a Provide Roaming Number negative response indicating system failure to the call handling process in the HLR, and waits for a response. The handling of the response from the call handling process in the HLR is described above.

Earlier version MAP dialogue with the GMSC

If the macro Receive_Open_Ind takes the Vr exit, the the HLR performs the earlier version MAP dialogue as specified in [51] or [96] 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 18.2/4 (sheet 1 of 3): Process SRI_HLR

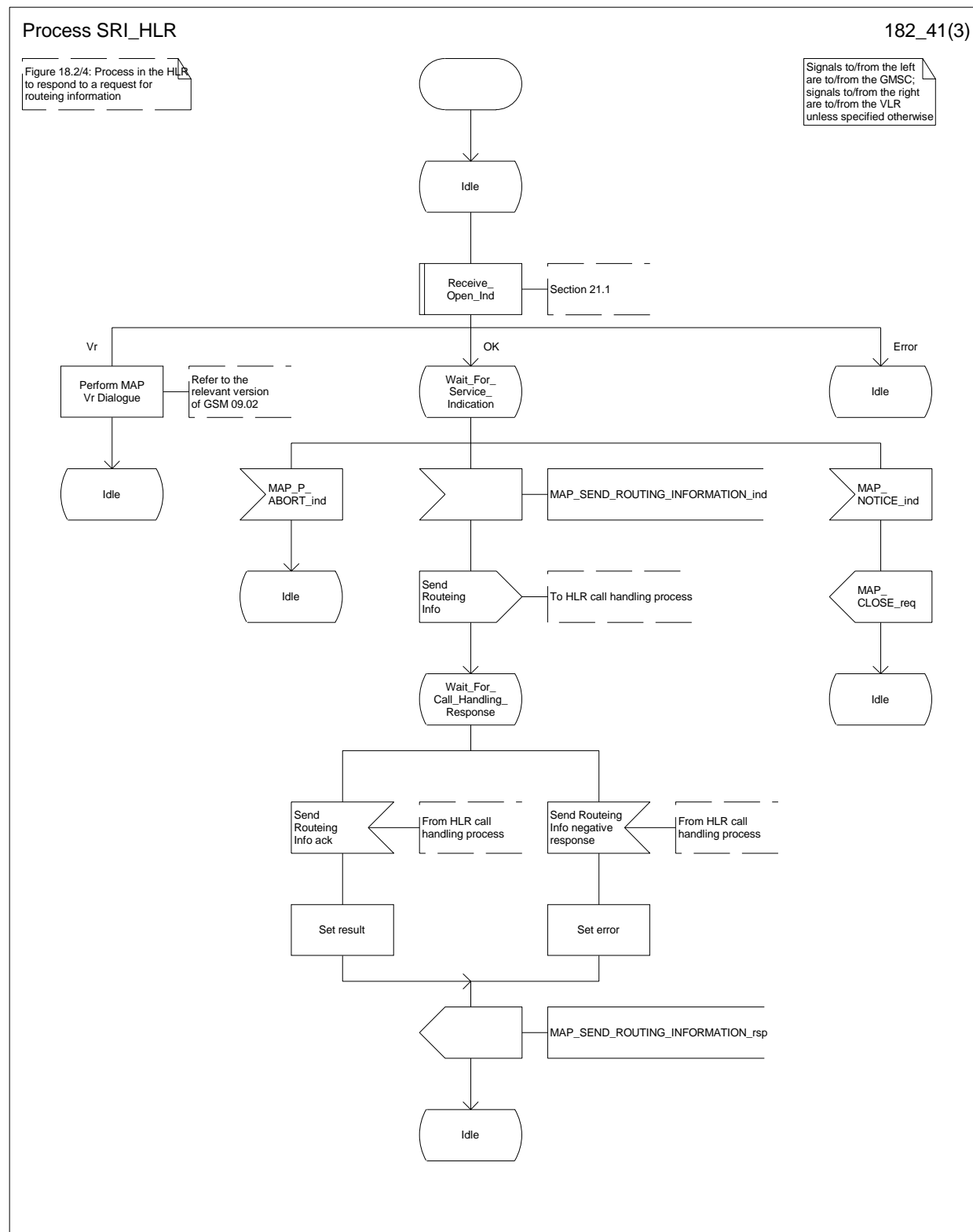


Figure 18.2/4 (sheet 2 of 3): Process SRI_HLR

Process SRI_HLR

182_42(3)

Figure 18.2/4: Process in the HLR to respond to a request for routing information

Signals to/from the left are to/from the GMSC; signals to/from the right are to/from the VLR unless specified otherwise

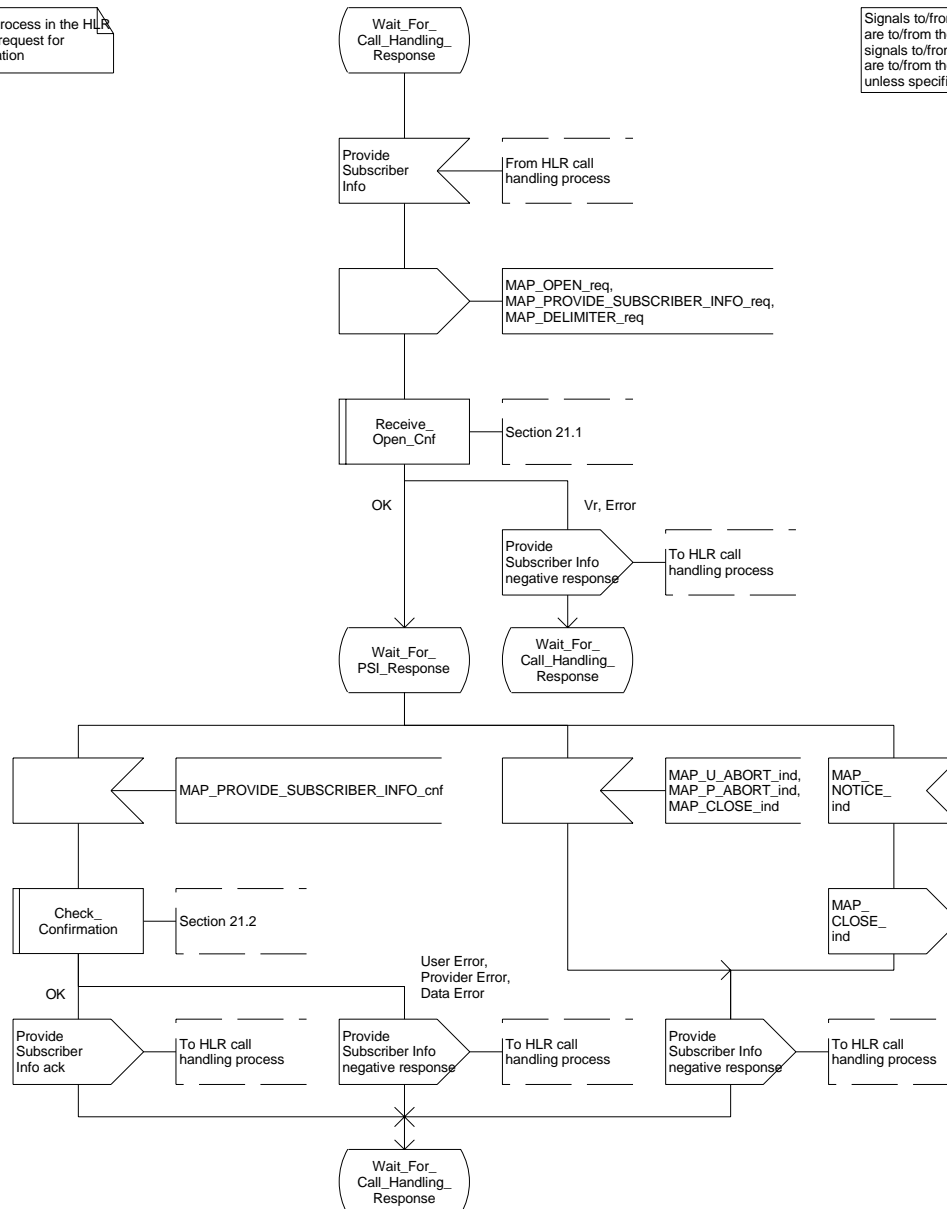
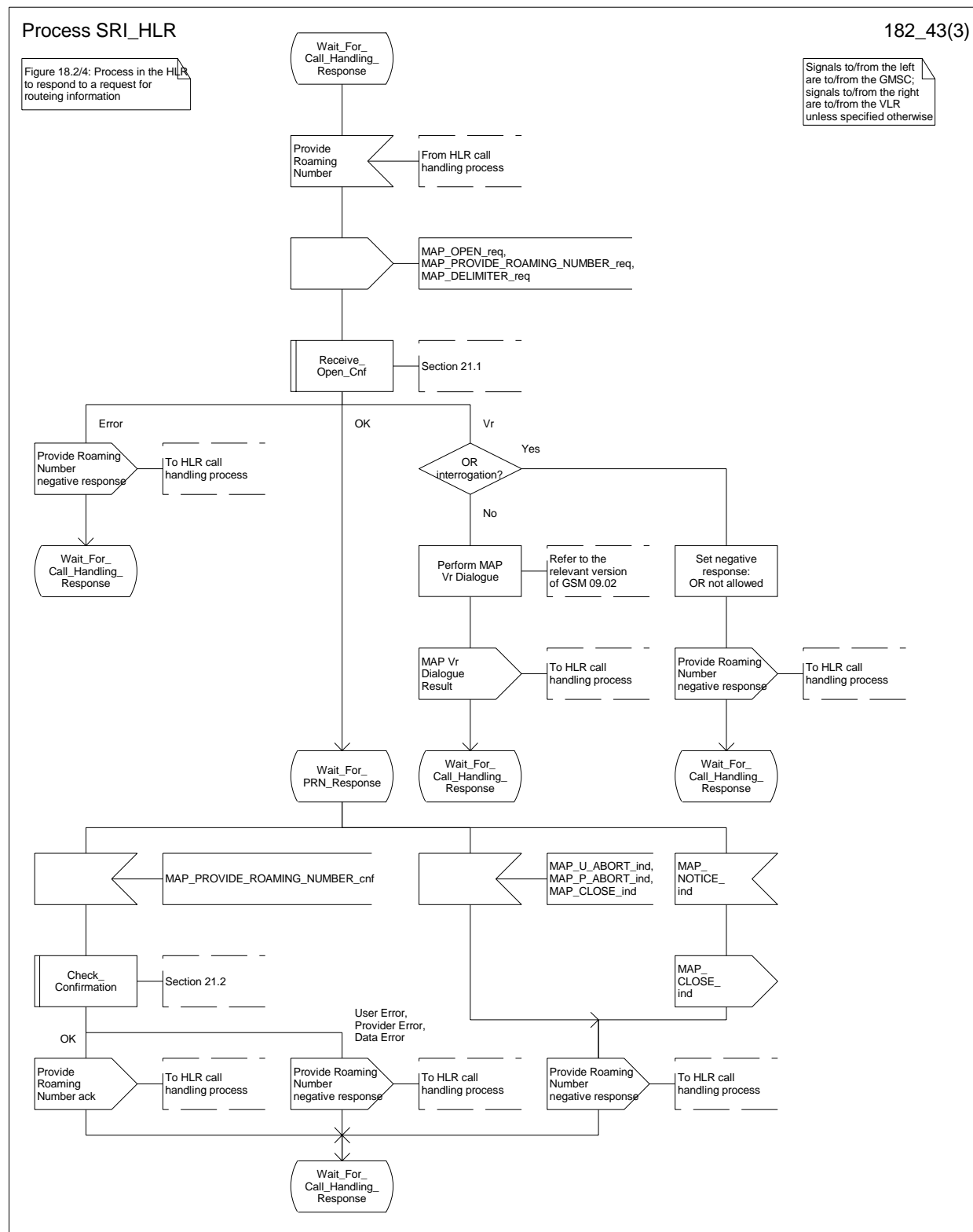


Figure 18.2/4 (sheet 3 of 3): Process SRI_HLR



18.2.4 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 18.2/5. The MAP process invokes a macro not defined in this subclause; the definition of this macro can be found as follows:

Receive_Open_Ind see subclause 21.1.1;

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 Routing Info ack, constructs a MAP_CLOSE service request, sends them to the HLR and returns to the idle state.

Earlier version MAP dialogue with the HLR

If the macro `Receive_Open_Ind` takes the `Vr` exit, the the VLR performs the earlier version MAP dialogue as specified in [51] or [96] 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.

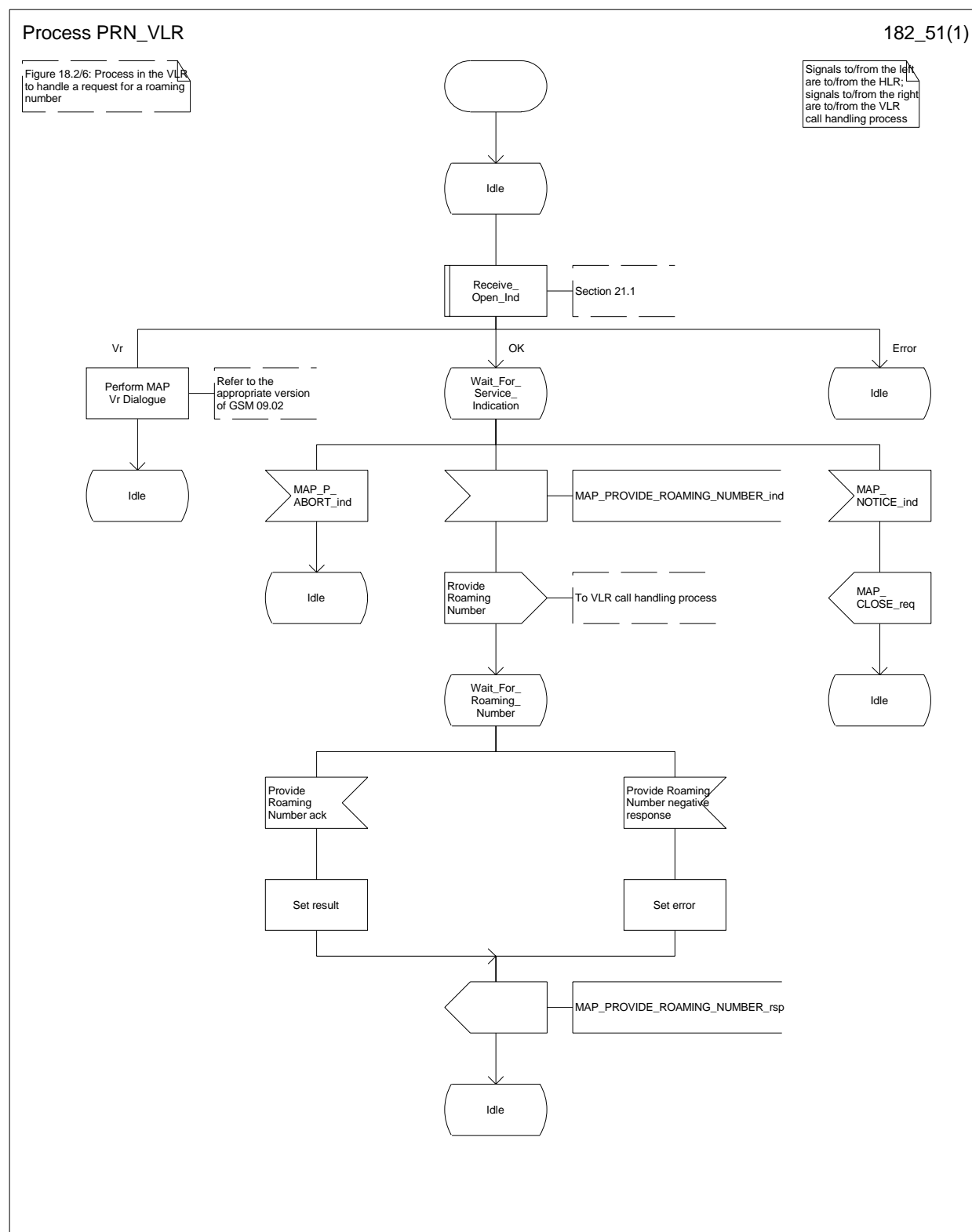
If the MAP provider sends a MAP_P_ABORT while the MAP process is waiting for a service indication, the MAP process returns to the idle state.

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 18.2/5: Process PRN_VLR



18.2.5 Process in the VLR to restore subscriber data

The MAP process in the HLR to restore subscriber data is shown in figure 18.2/6. The MAP process invokes macros not defined in this subclause; the definitions of these macros can be found as follows:

Receive_Open_Cnf	see subclause 21.1.2;
Check_Confirmation	see subclause 21.2.2;
Insert_Subscriber_Data_VLR	see subclause 21.7.1;
Activate_Tracing_VLR	see subclause 21.9.3.

Successful outcome

When the MAP process receives a Restore Data request from the data restoration process in the VLR, it requests a dialogue with the HLR whose identity is contained in the Restore Data request by sending a MAP_OPEN service request, requests data restoration using a MAP_RESTORE_DATA service request and invokes the macro Receive_Open_Cnf to wait for the response to the dialogue opening request. If the dialogue opening is successful, the MAP process waits for a response from the HLR.

The VLR may receive a MAP_INSERT_SUBSCRIBER_DATA service indication from the HLR; this is handled by the macro Insert_Subscriber_Data_VLR as described in subclause 21.7.1, and the MAP process waits for a further response from the HLR.

The VLR may receive a MAP_ACTIVATE_TRACE_MODE service indication from the HLR; this is handled by the macro Activate_Tracing_VLR as described in subclause 21.9.3, 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.

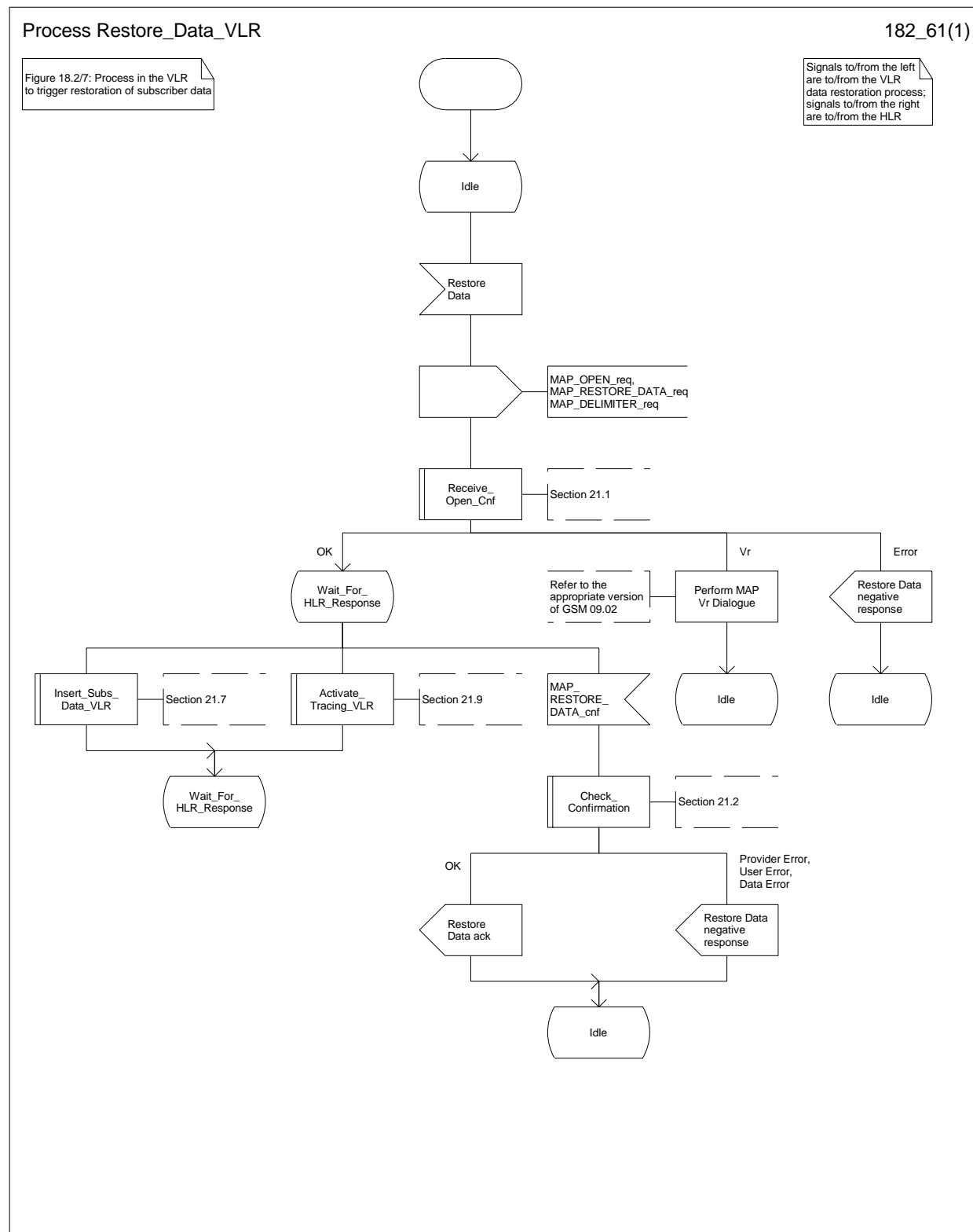
Earlier version MAP dialogue with the HLR

If the macro Receive_Open_Cnf takes the Vr exit, the VLR performs the earlier MAP version dialogue as specified in [51] or [96] and the process terminates.

Dialogue opening failure

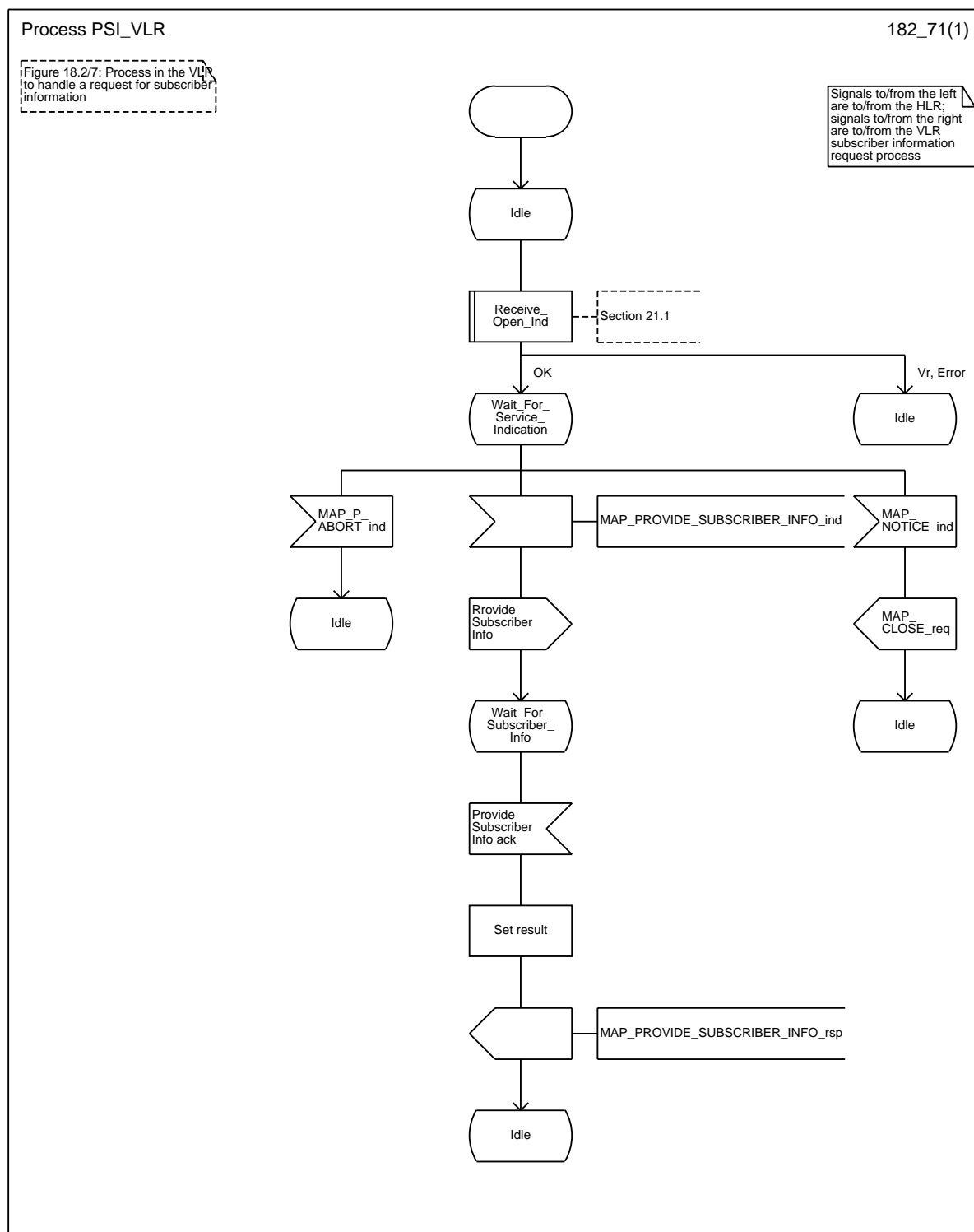
If the macro Receive_Open_Cnf indicates that the dialogue with the HLR could not be opened, the MAP process sends a negative response indicating system failure to the data restoration process in the GMSC and returns to the idle state.

Figure 18.2/6: Process Restore_Data_VLR



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 18.2/7: Process PSI_VLR



18.2.7 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 18.2/8.

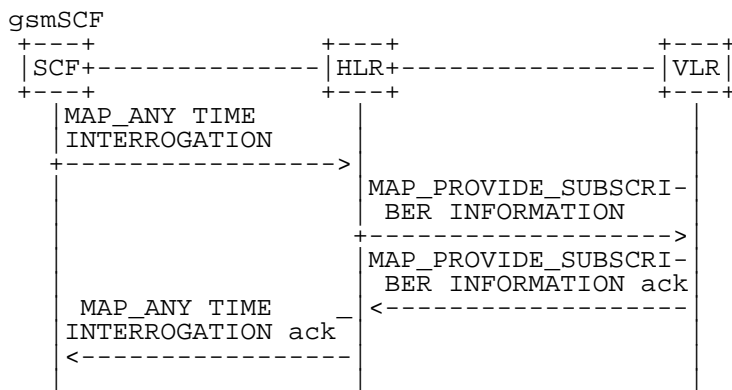


Figure 18.2/8: Message flow for any time interrogation

The following MAP services are used to retrieve routing information:

MAP_ANY_TIME_INTERROGATION see subclause 6.11.1;
 MAP_PROVIDE_SUBSCRIBER_INFO see subclause 6.11.2;

18.2.7.1 Process in the gsmSCF

Out of the scope of the MAP specification.

18.2.8 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 18.2/8. The MAP process invokes macros not defined in this subclause; the definitions of these macros can be found as follows:

Receive_Open_Ind see subclause 21.1.1;
 Receive_Open_Cnf see subclause 21.1.2;
 Check_Confirmation see subclause 21.2.2.

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 GSM 03.78), 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 18.2/9 (sheet 1 of 2): Process ATI_HLR (New)

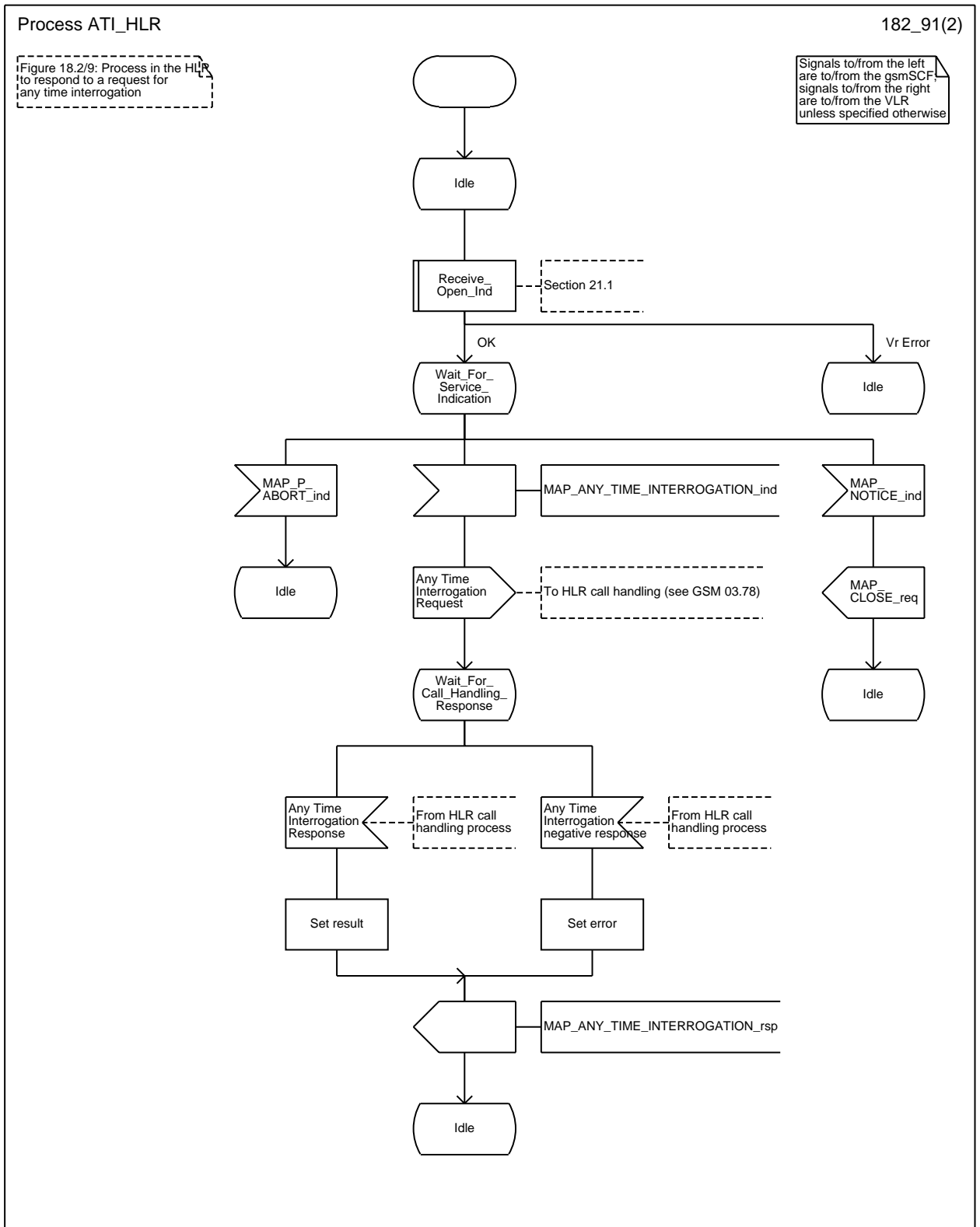
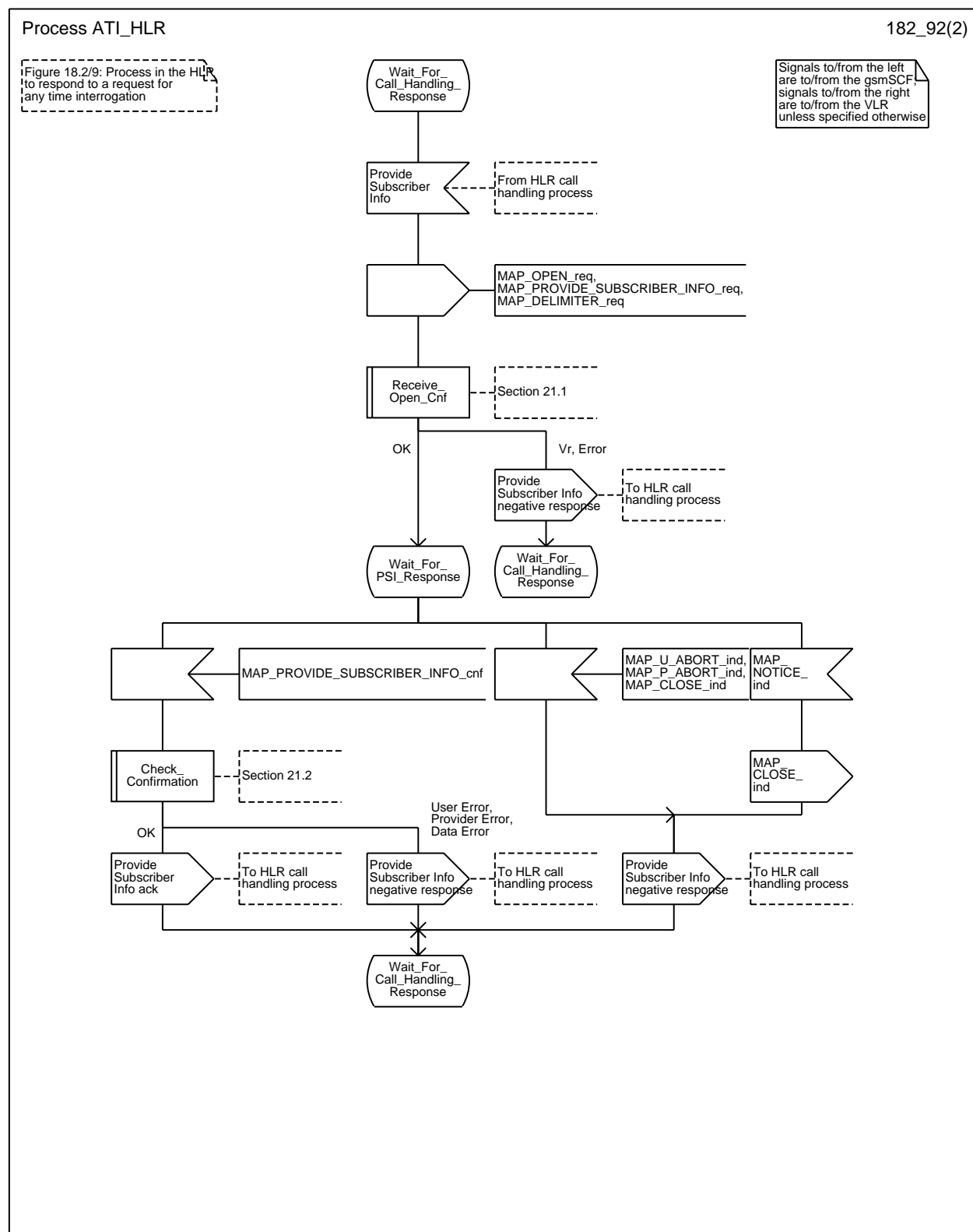


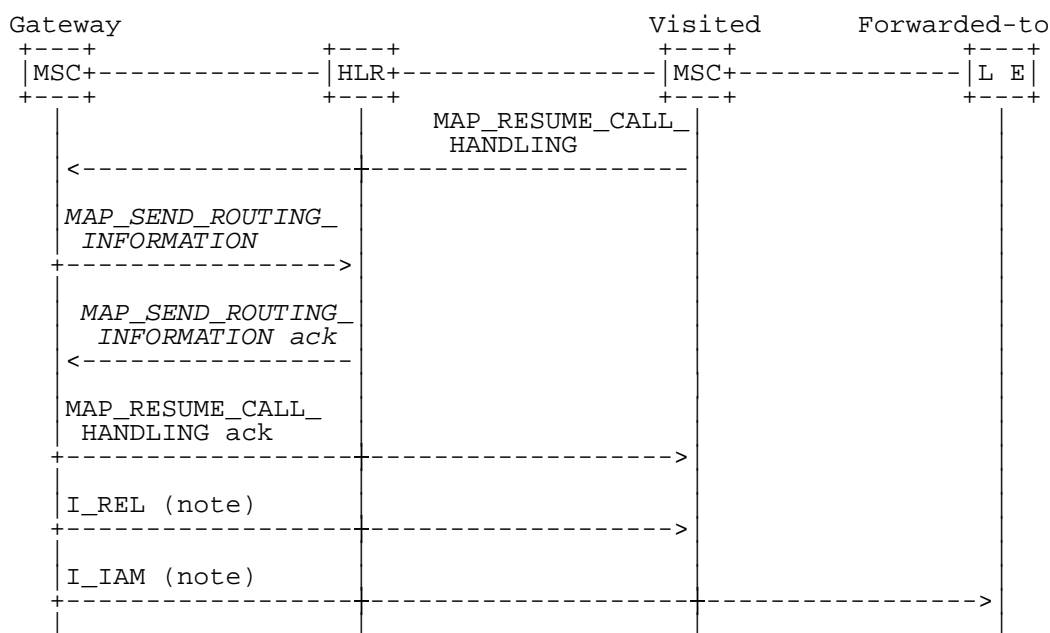
Figure 18.2/9 (sheet 2 of 2): Process ATI_HLR (New)



18.3 Transfer of call handling

18.3.1 General

The message flow for successful transfer of call handling to forward a call is shown in figure 18.3/1.



NOTES:

xxx = Optional Procedure

TUP or ISUP may be used in signalling between MSCs, depending on the network type between the MSCs. For further details on the TUP and ISUP procedures refer to the following CCITT Recommendations & ETSI specification:

Q.721-725 - Telephone User Part (TUP);
 ETS 300 356-1 - Integrated Services Digital Network (ISDN); Signalling System No.7;
 ISDN User Part (ISUP) version 2 for the international interface; Part 1: Basic services.

Figure 18.3/1: Message flow for transfer of call handling

If the HLR indicated in the response to the original request for routing information that forwarding interrogation is required, the GMSC executes the Send Routing Information procedure with the HLR to obtain forwarding information; otherwise the GMSC uses the forwarding data which were sent in the MAP_RESUME_CALL_HANDLING req/ind.

18.3.2 Process in the VMSC

The MAP process in the VMSC to retrieve routing information for a mobile terminating call is shown in figure 18.3/2. The MAP process invokes macros not defined in this subclause; the definitions of these macros can be found as follows:

Receive_Open_Cnf	see subclause 21.1.2;
Check_Confirmation	see subclause 21.2.2.

Successful Outcome

When the MAP process receives a Resume Call Handling request from the call handling process in the VMSC, it requests a dialogue with the GMSC whose identity is contained in the Resume Call Handling request by sending a MAP_OPEN service request, requests routing information using a MAP_RESUME_CALL_HANDLING service request and invokes the macro Receive_Open_Cnf to wait for the response to the dialogue opening request. If the dialogue opening is successful, the MAP process waits for a response from the GMSC.

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 sends a Resume Call Handling ack 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 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.

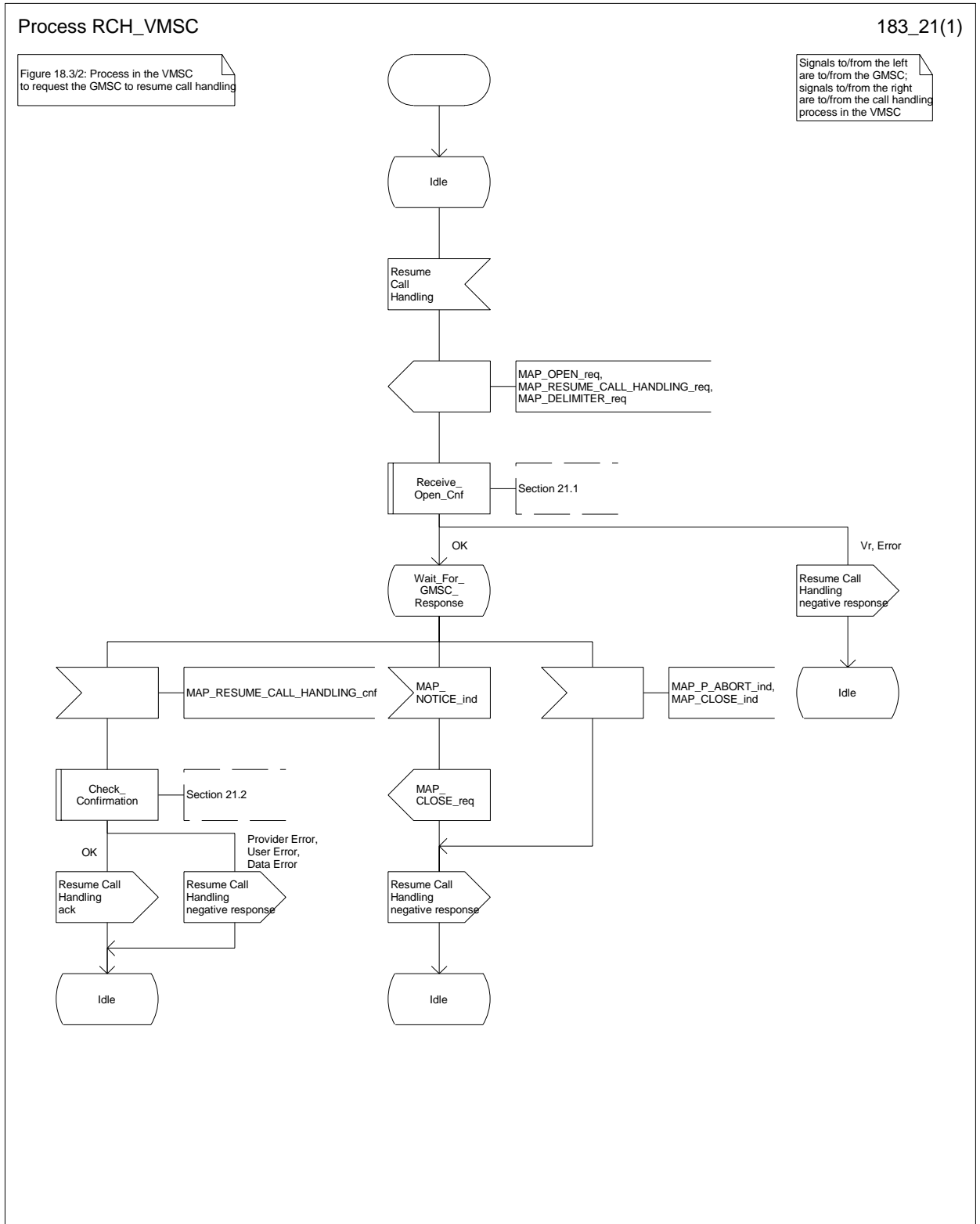
NOTE: the 'Data Error' exit from the macro Check_Confirmation is shown for formal completeness; the result is empty, so the MAP_PROVIDE_SUBSCRIBER_INFO_cnf primitive cannot contain a data error.]

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 18.3/2: Process RCH_VMSC



18.3.3 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 18.3/3. The MAP process invokes a macro not defined in this subclause; the definition of this macro can be found as follows:

Receive_Open_Ind see subclause 21.1.1;

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 sends a Resume Call Handling request 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 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 HLR and returns to the idle state.

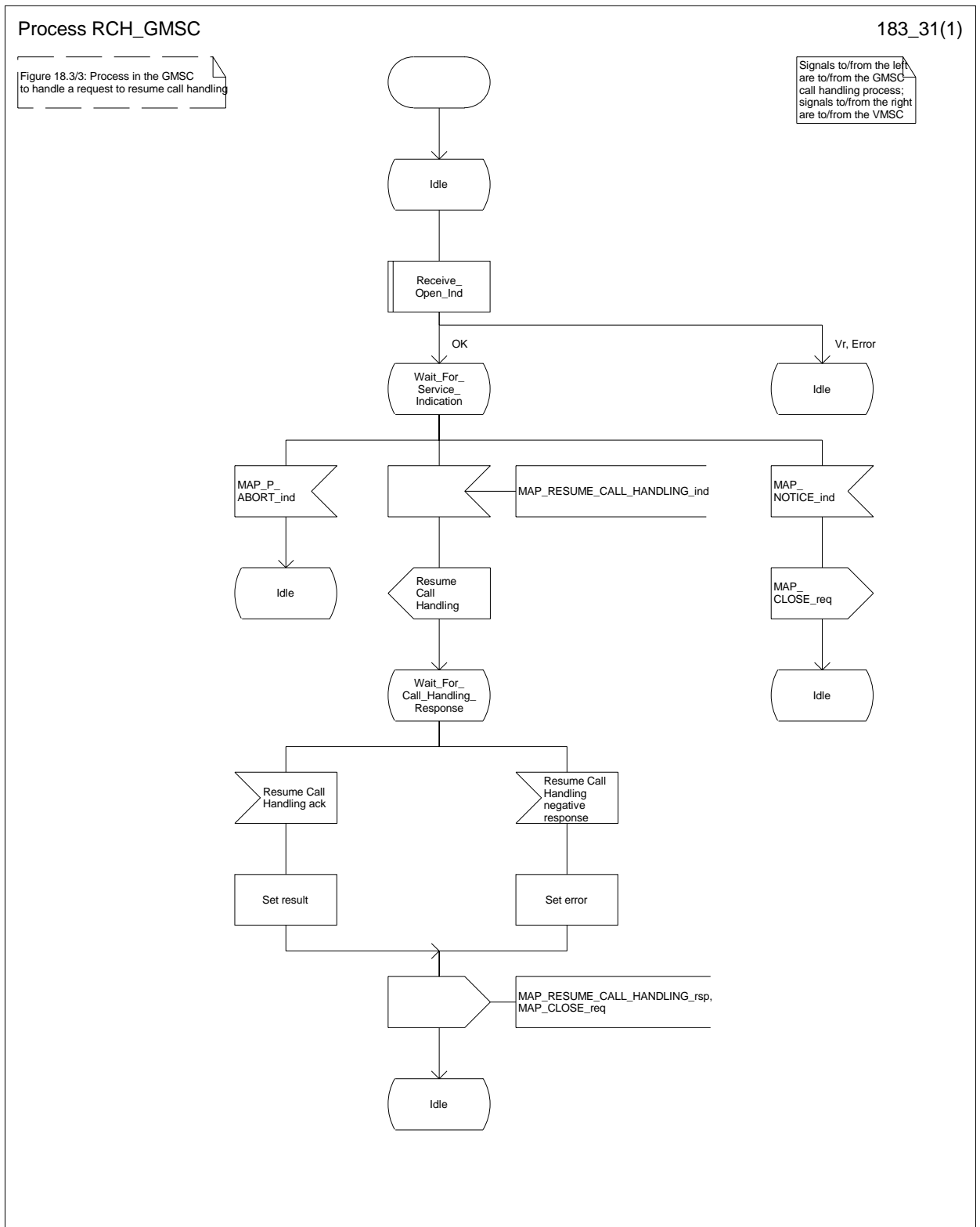
Failure of dialogue opening with the VMSC

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 18.3/3: Process RCH_GMSC



19 Supplementary services procedures

The following application contexts exist for handling of supplementary services:

- accessUnstructuredSsContext;
- accessFunctionalSsContext.

The accessUnstructuredSsContext refers to a simple MAP users, for which the corresponding MAP process can be identified by the MAP-Provider directly.

However, the accessFunctionalSsContext refers to a complex MAP-User consisting of several processes. For this user, a process co-ordinator is defined for each network entity, in order to identify the correct process to invoke. These processes open and validate the dialogue, then invoke the necessary operation-specific process. These processes are described below.

19.0 Functional supplementary service processes

19.0.1 Functional supplementary service process co-ordinator for MSC

Upon receipt of a CM-Service request with CM-service type = SS, the MSC initiates the process access request procedure towards the VLR as described in clause 21 of this ETS.

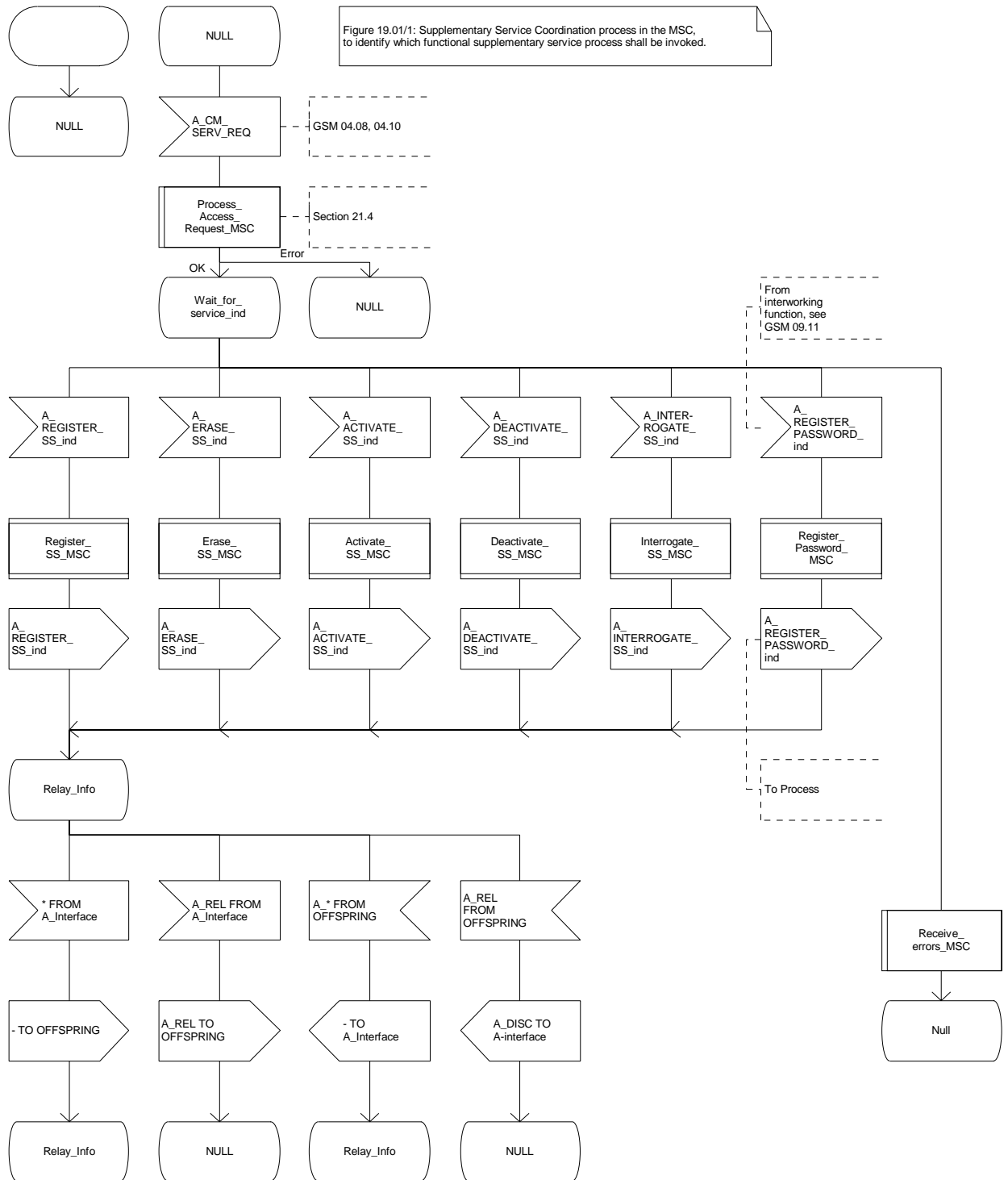
Once a CM connection is established, the MSC can handle supplementary service indications from the MS. Table 19.0/1 shows the co-ordinating process' reaction on receipt of specific SS service indications on the air interface. After the relevant process is invoked, the received air interface service indication is sent to that process. The creation of service requests on the basis of air interface messages is described in GSM 09.11.

Service indication received	Process invoked
A_REGISTER_SS_ind	REGISTER_SS_MSC
A_ERASE_SS_ind	ERASE_SS_MSC
A_ACTIVATE_SS_ind	ACTIVATE_SS_MSC
A_DEACTIVATE_SS_ind	DEACTIVATE_SS_MSC
A_INTERROGATE_SS_ind	INTERROGATE_SS_MSC
A_REGISTER_PASSWORD	REGISTER_PASSWORD_MSC

Table 19.0/1: Relationship between received service indication and invoked process in the MSC

Figure 19.0/1 shows the co-ordinating process in the MSC.

Figure 19.0/1



19.0.2 Functional supplementary service process co-ordinator for VLR

Any functional SS process in the VLR starts by the VLR receiving the MAP_PROCESS_ACCESS_REQUEST indication. The VLR then acts as described in clause 21 of this ETS.

If the Process Access Request was successful, the VLR can handle supplementary service indications from the MSC. Table 19.0/2 shows the co-ordinating process' reaction on receipt of specific SS service indications from the MSC. After the relevant process is invoked, the received service indication is sent to that process, and the co-ordinating process terminates.

Service indication received	Process invoked
MAP_REGISTER_SS_ind	REGISTER_SS_VLR
MAP_ERASE_SS_ind	ERASE_SS_VLR
MAP_ACTIVATE_SS_ind	ACTIVATE_SS_VLR
MAP_DEACTIVATE_SS_ind	DEACTIVATE_SS_VLR
MAP_INTERROGATE_SS_ind	INTERROGATE_SS_VLR
MAP_REGISTER_PASSWORD	REGISTER_PASSWORD_VLR

Table 19.0/2: Relationship between received service indication and invoked process in the VLR

Figure 19.0/2 shows the co-ordinating process in the VLR.

Figure 19.0/2 (sheet 1 of 2)

Figure 19.0/2: Supplementary Service Coordination process in the VLR, to open and process the access request from the MSC, and then identify which functional supplementary service process shall be invoked.

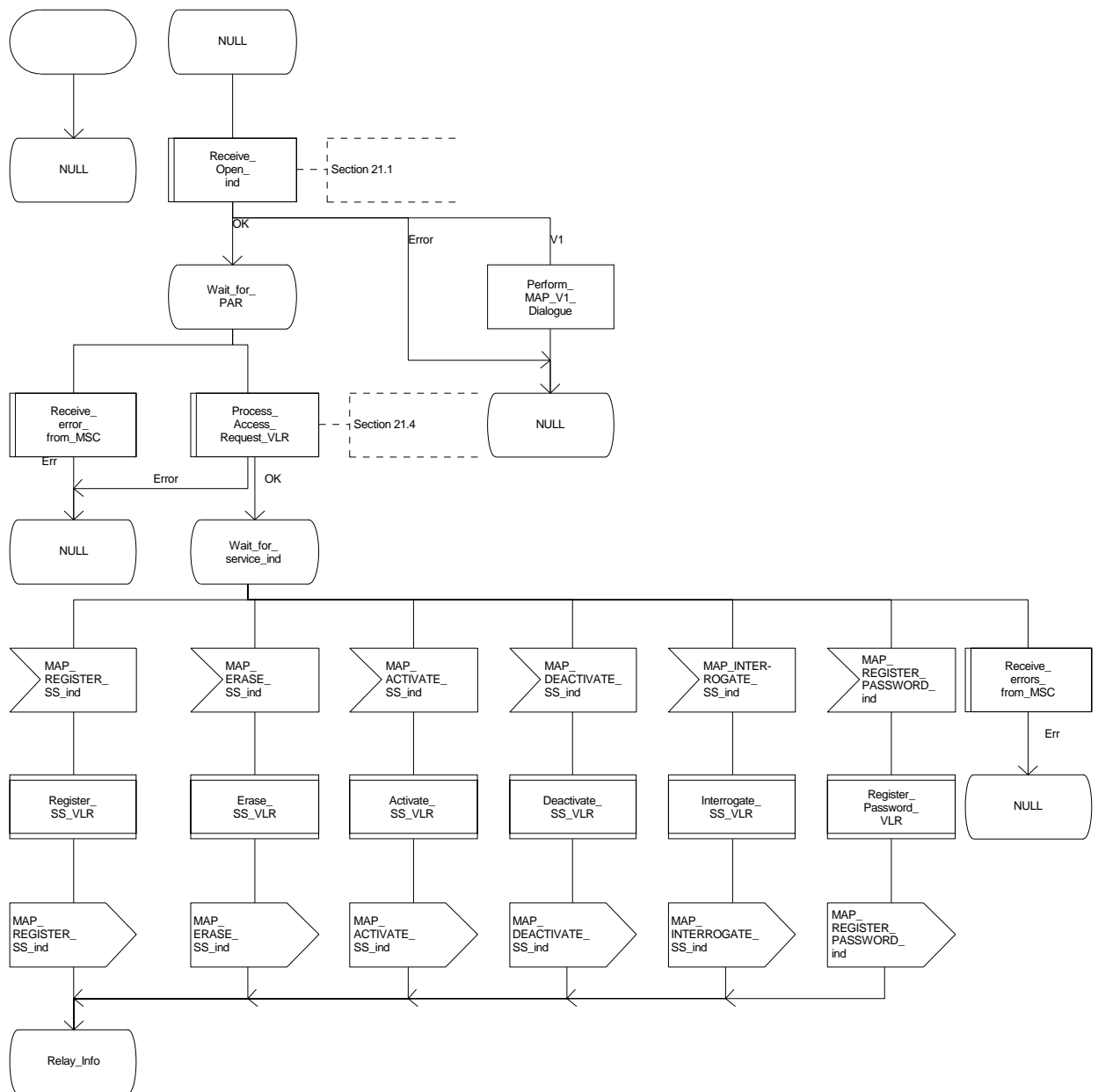
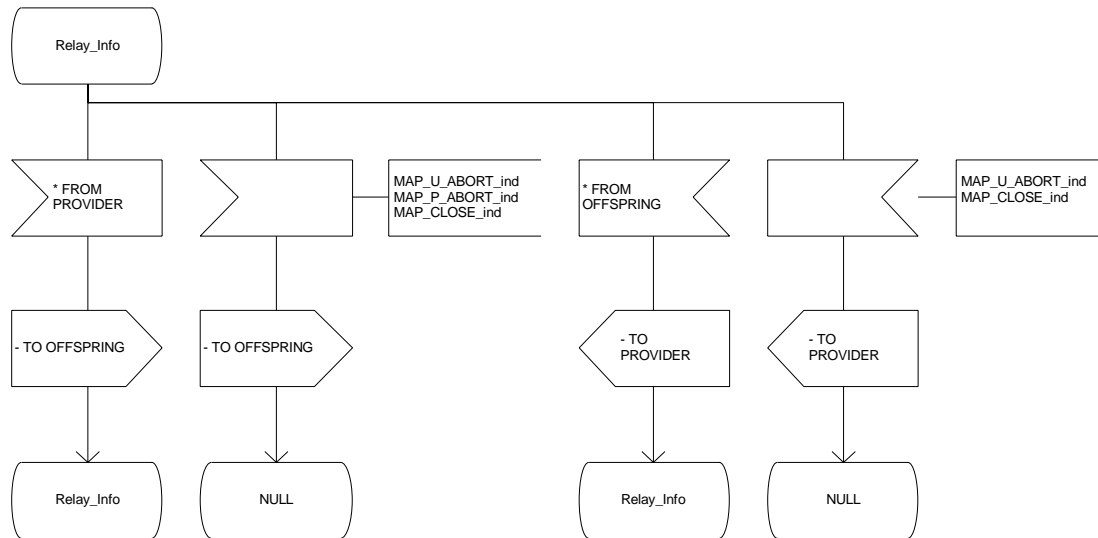


Figure 19.0/2 (sheet 2 of 2)



19.0.3 Functional supplementary service process co-ordinator for HLR

Any functional SS process in the HLR starts by the HLR receiving a MAP-OPEN service indication. If that service is successful, the HLR can handle supplementary service indications from the VLR. Table 19.0/3 shows the co-ordinating process' reaction on receipt of specific SS service indications from the VLR. After the relevant process is invoked, the received service indication is sent to that process, and the co-ordinating process terminates.

Service indication received	Process invoked
MAP_REGISTER_SS_ind	REGISTER_SS_HLR
MAP_ERASE_SS_ind	ERASE_SS_HLR
MAP_ACTIVATE_SS_ind	ACTIVATE_SS_HLR
MAP_DEACTIVATE_SS_ind	DEACTIVATE_SS_HLR
MAP_INTERROGATE_SS_ind	INTERROGATE_SS_HLR
MAP_REGISTER_PASSWORD	REGISTER_PASSWORD_HLR

Table 19.0/3: Relationship between received service indication and invoked process in the HLR.

Figure 19.0/3 shows the co-ordinating process in the HLR.

Figure 19.0/3 (sheet 1 of 2)

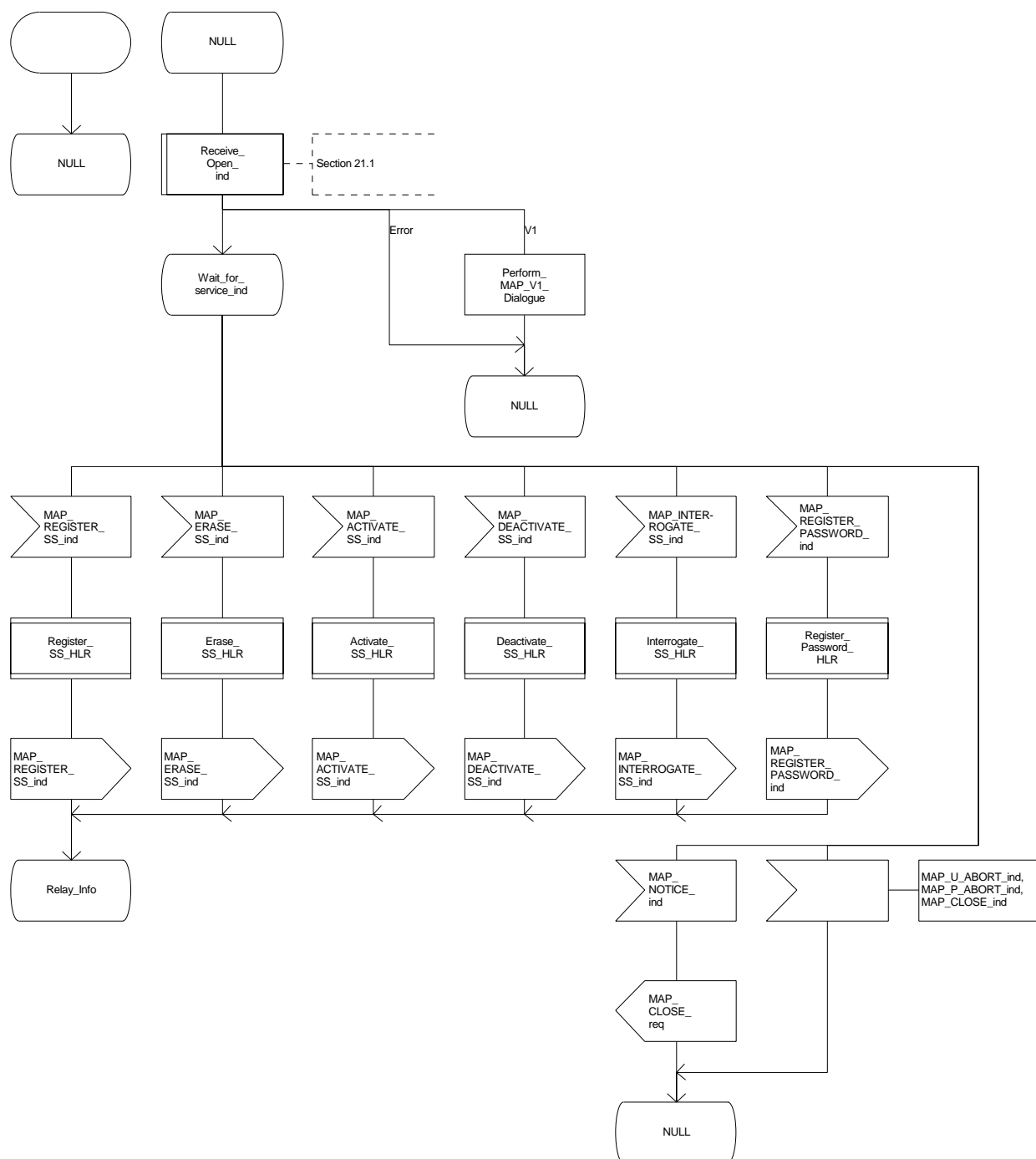
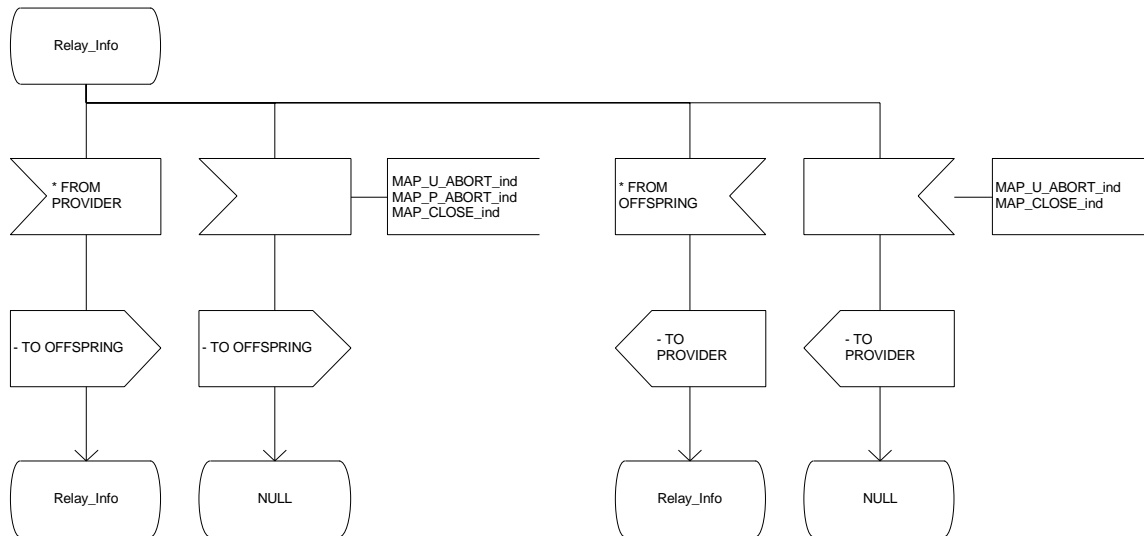


Figure 19.0/3 (sheet 2 of 2)

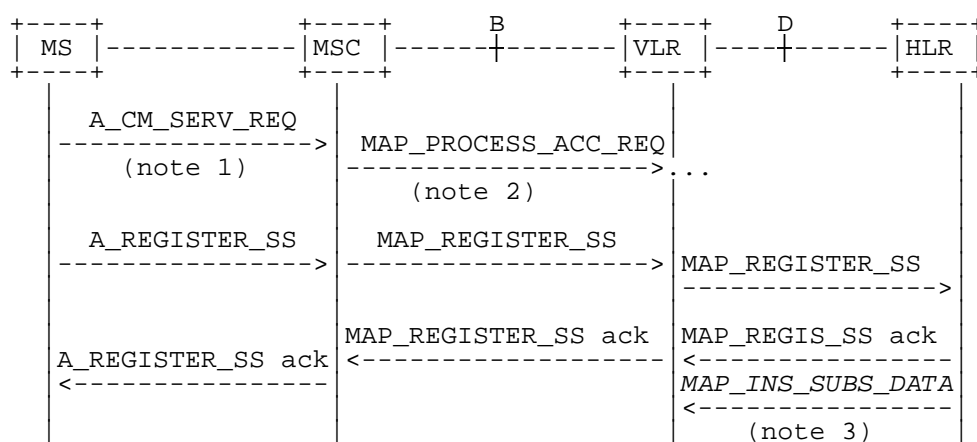


19.1.1 General

The registration procedure is shown in figure 19.1.1/1.

The following services may be used:

MAP_PROCESS_ACCESS_REQUEST	(defined in clauses 6 and 21);
MAP_TRACE_SUBSCRIBER_ACTIVITY	(defined in clauses 7 and 21);
MAP_PROVIDE_IMSI	(defined in clauses 6 and 21);
MAP_FORWARD_NEW_TMSI	(defined in clauses 6 and 21);
MAP_AUTHENTICATE	(defined in clauses 6 and 21);
MAP_SET_CIPHERING_MODE	(defined in clauses 6 and 21);
MAP_CHECK_IMEI	(defined in clauses 6 and 21);
MAP_READY_FOR_SM	(defined in clauses 10 and 21);
MAP_INSERT_SUBSCRIBER_DATA	(defined in clauses 6 and 21);
MAP_REGISTER_SS	(defined in clause 9).



NOTE 1: For details of the procedure on the radio path, see GSM 04.08, 04.10, 04.8x and 04.9x. Services shown in dotted lines indicate the trigger provided by the signalling on the radio path, and the signalling triggered on the radio path.

NOTE 2: For details on the Process Access Request procedure, please refer to clause 21 in this ETS.

NOTE 3: Services printed in *italics* are optional.

Figure 19.1.1/1: Interfaces and services for supplementary service registration

19.1.2 Procedures in the MSC

Supplementary service registration

The A_REGISTER_SS service indication received by the MAP user in the MSC contains the SS-Code and any parameters that are related to the supplementary service.

The MAP user transfers the received information to the VLR in the MAP_REGISTER_SS request without checking the contents of the service indication. Rules for the mapping are described in GSM 09.11.

The MSC then awaits the receipt of the MAP_REGISTER_SS confirm from the VLR. The outcome of the procedure is reported to the MS in the A_REGISTER_SS response message as described in GSM 04.8x, 04.9x and 09.11. Finally the SS-connection is released.

For call independent SS operations, each message shall only contain a single component. Messages which contain more than one component will be stopped at the air interface (as specified in GSM 09.11).

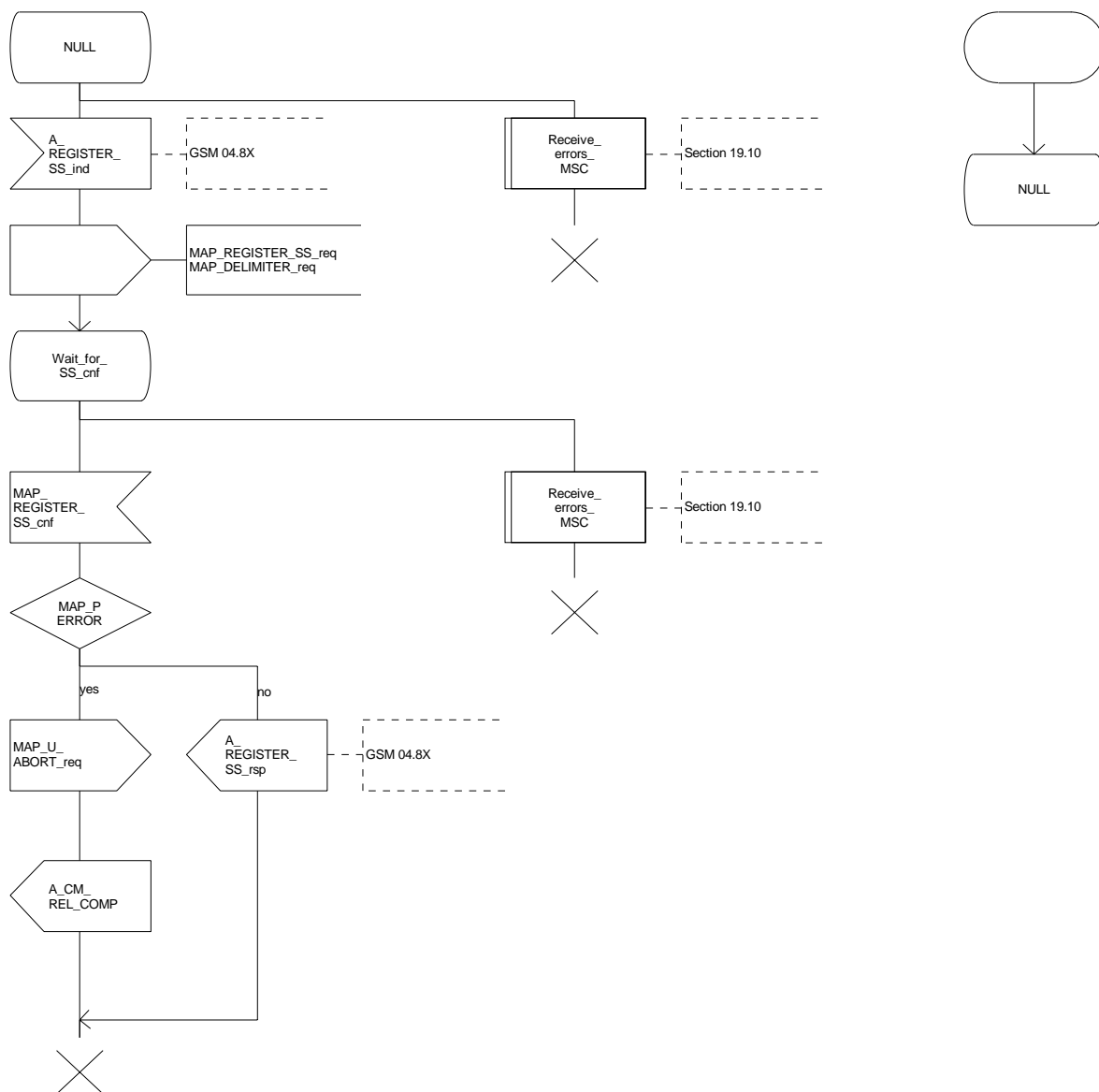
Error handling

If at any time during the supplementary service part of this procedure a MAP_P_ABORT, MAP_U_ABORT, MAP_NOTICE or unexpected MAP_CLOSE indication is received from the VLR concerning the process, a CM_RELEASE_COMPLETE indication is sent to the MS (as specified in GSM 09.11). Upon receipt of a MAP_NOTICE indication from the VLR, the MSC must close the VLR dialogue by sending a MAP_CLOSE request. The process is then terminated.

If an A_CM_RELEASE indication is received from the MS, all open transactions shall be released using the MAP_U_ABORT request indicating application procedure cancellation, and the process is terminated.

The registration procedure in the MSC is shown in figure 19.1.2/1.

Figure 19.1.2/1



19.1.3 Procedures in the VLR

Supplementary service registration

When receiving the MAP_REGISTER_SS indication, the MAP user in the VLR transfers the information to the HLR in the MAP_REGISTER_SS request without checking the contents of the service indication.

The VLR then awaits the receipt of the MAP_REGISTER_SS confirm from the HLR. The MAP user in the VLR shall transfer the information contained in this primitive to the MSC in the MAP_REGISTER_SS response without checking its contents.

For call independent SS operations, each message shall only contain a single component. Messages which contain more than one component will be stopped at the air interface (as specified in GSM 09.11).

Error handling

If at any time during this procedure a MAP_P_ABORT, MAP_U_ABORT, MAP_NOTICE or unexpected MAP_CLOSE indication is received from the MSC concerning the process, a MAP_U_ABORT request indicating application procedure cancellation is sent to the HLR (if a connection exists). If a MAP_NOTICE indication was received from the MSC, that dialogue must be closed by sending a MAP_CLOSE request towards the MSC. The process is terminated.

If a MAP_P_ABORT, MAP_U_ABORT or MAP_CLOSE indication is received from the HLR, a MAP_U_ABORT request shall be sent to the MSC terminating the process. If a MAP_NOTICE indication was received from the HLR, that dialogue must be closed by sending a MAP_CLOSE request towards the HLR. The process terminates.

The registration procedure in the VLR is shown in figure 19.1.3/1.

Figure 19.1.3/1 (sheet 1 of 2)

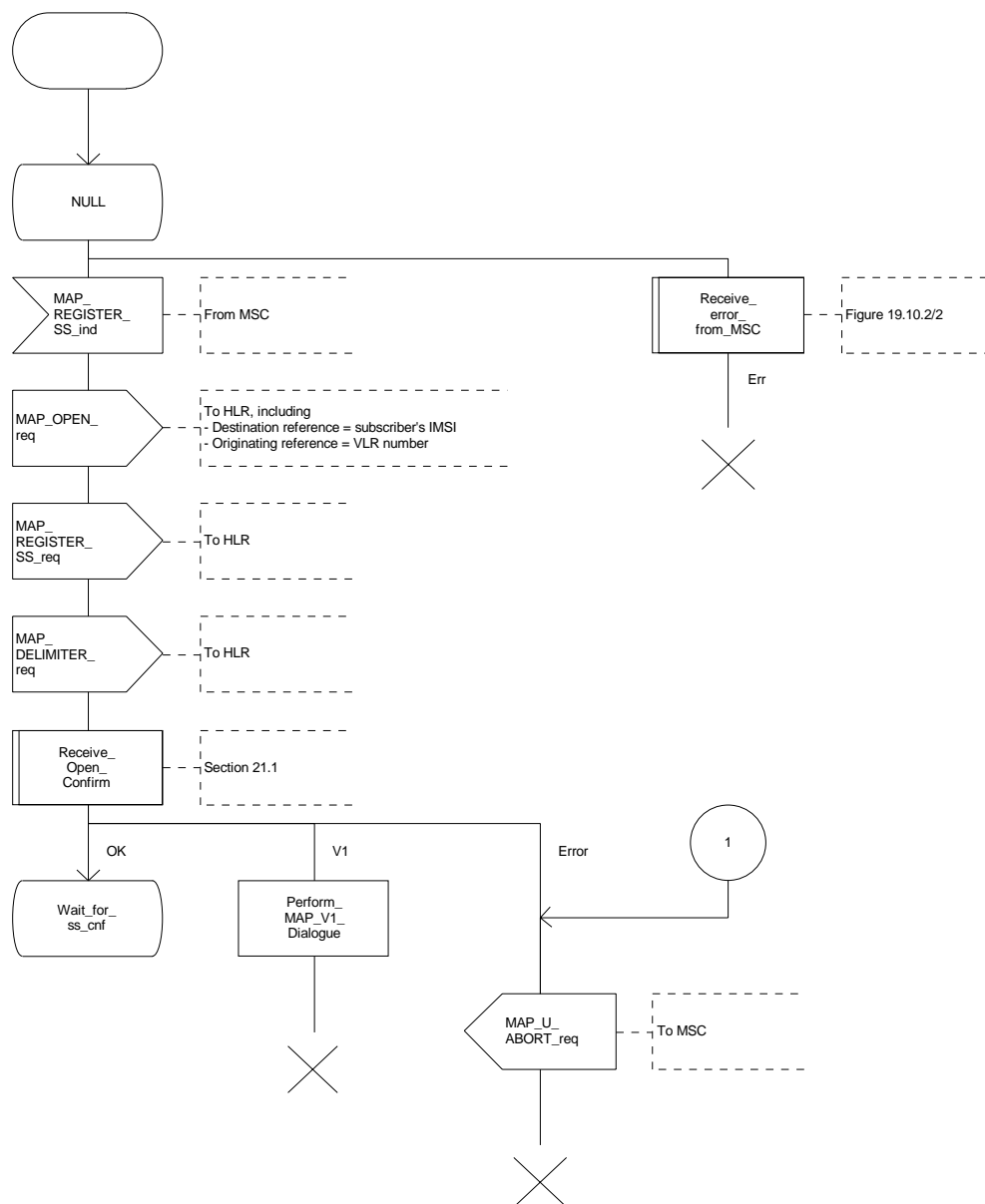
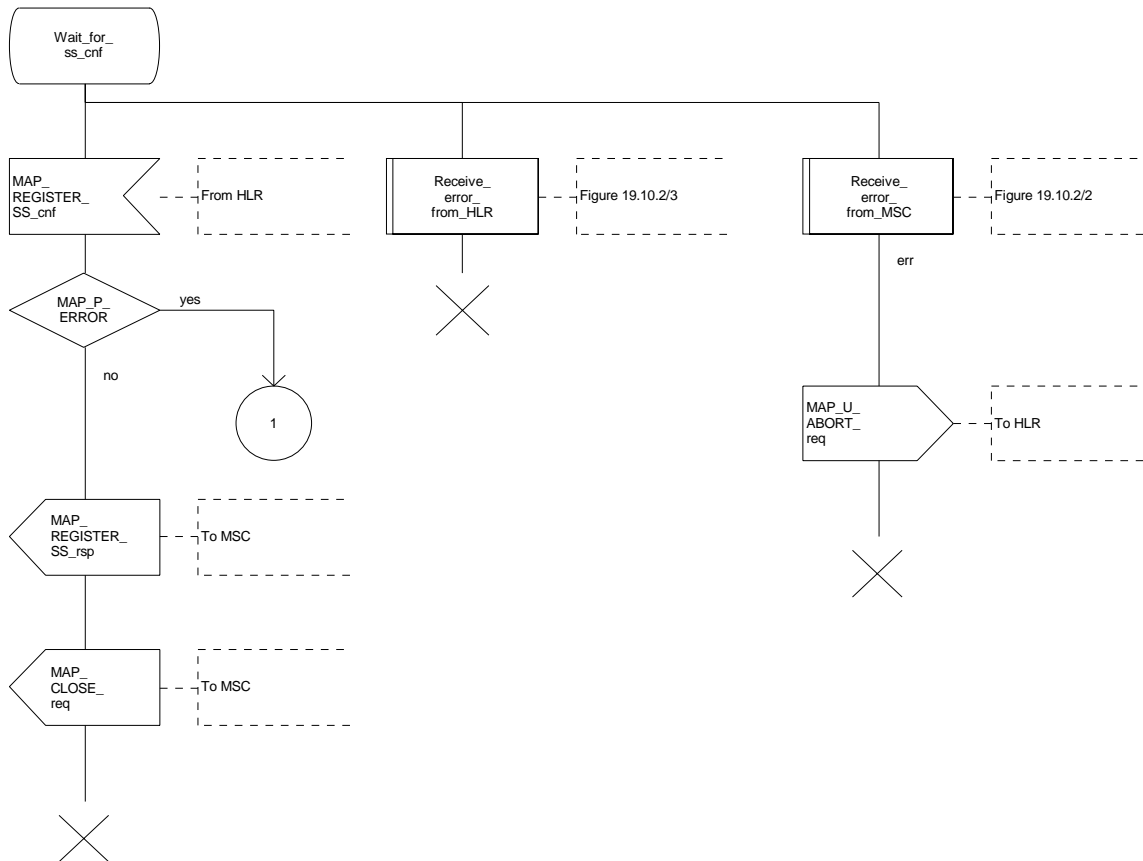


Figure 19.1.3/1 (sheet 2 of 2)



19.1.4 Procedures in the HLR

The procedure in the HLR is initiated when it receives a MAP_REGISTER_SS indication.

The HLR acts as follows:

- if the operator has barred the subscriber from access to supplementary services, the Call Barred error should be returned to the VLR. The parameter "operatorBarring" shall be included with the error.

The supplementary service request shall then be processed according to GSM 03.11 and the 03.8x and 03.9x-series of technical specifications. This handling may lead to either a successful result, a partially successful result, or an error being returned.

For call independent SS operations, each message shall only contain a single component. Messages which contain more than one component will be stopped at the air interface (as specified in GSM 09.11):

- if the VLR is to be updated after the supplementary service registration, the MAP_INSERT_SUBS_DATA_HLR process shall be initiated;
- if at any time during this procedure a MAP_P_ABORT, MAP_U_ABORT or MAP_CLOSE indication concerning the process is received from the VLR, the process is terminated. If a MAP_NOTICE indication is received, a MAP_CLOSE request indicating sent towards the VLR.

The registration procedure in the HLR is shown in figure 19.1.4/1.

Figure 19.1.4/1 (sheet 1 of 2)

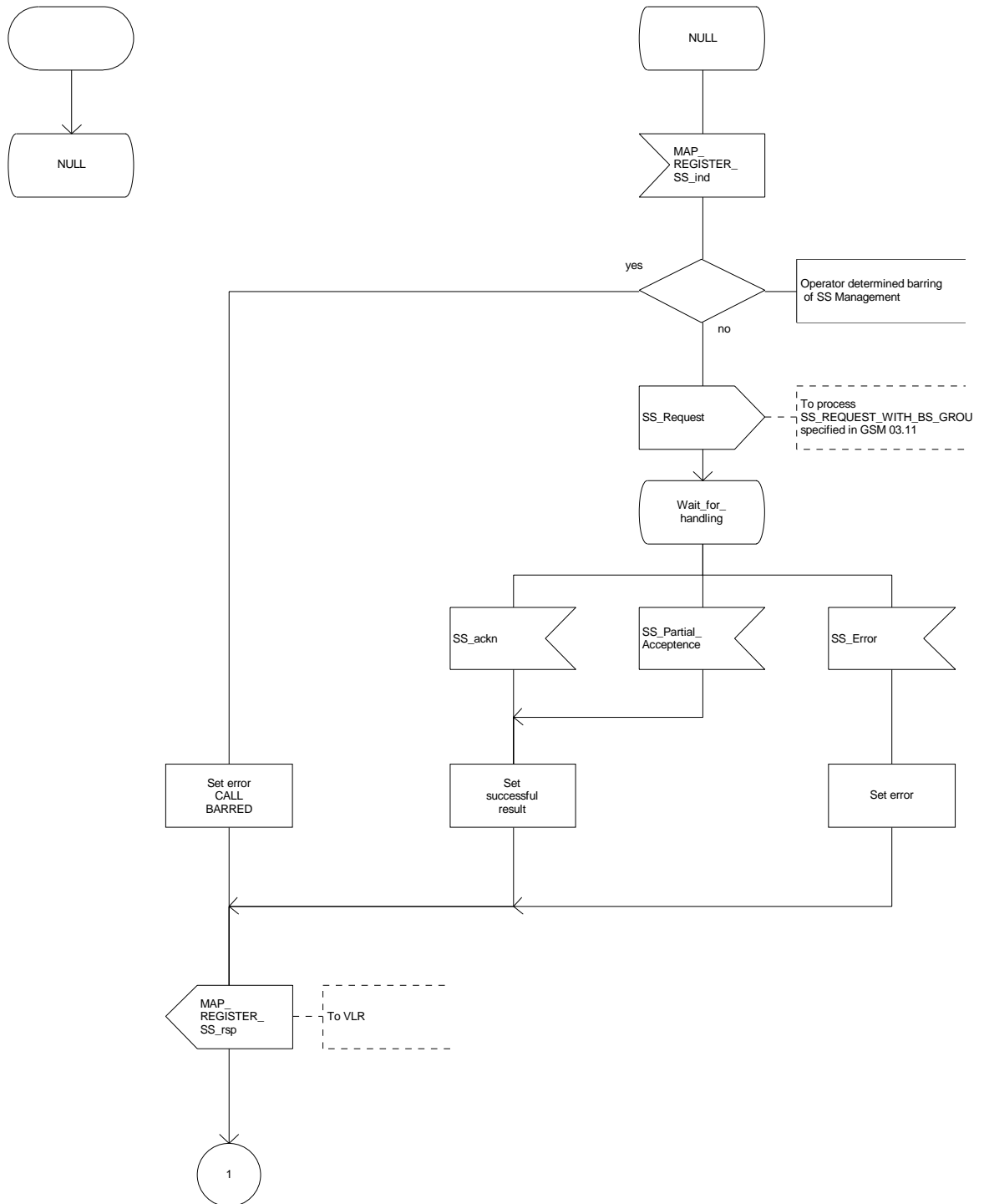
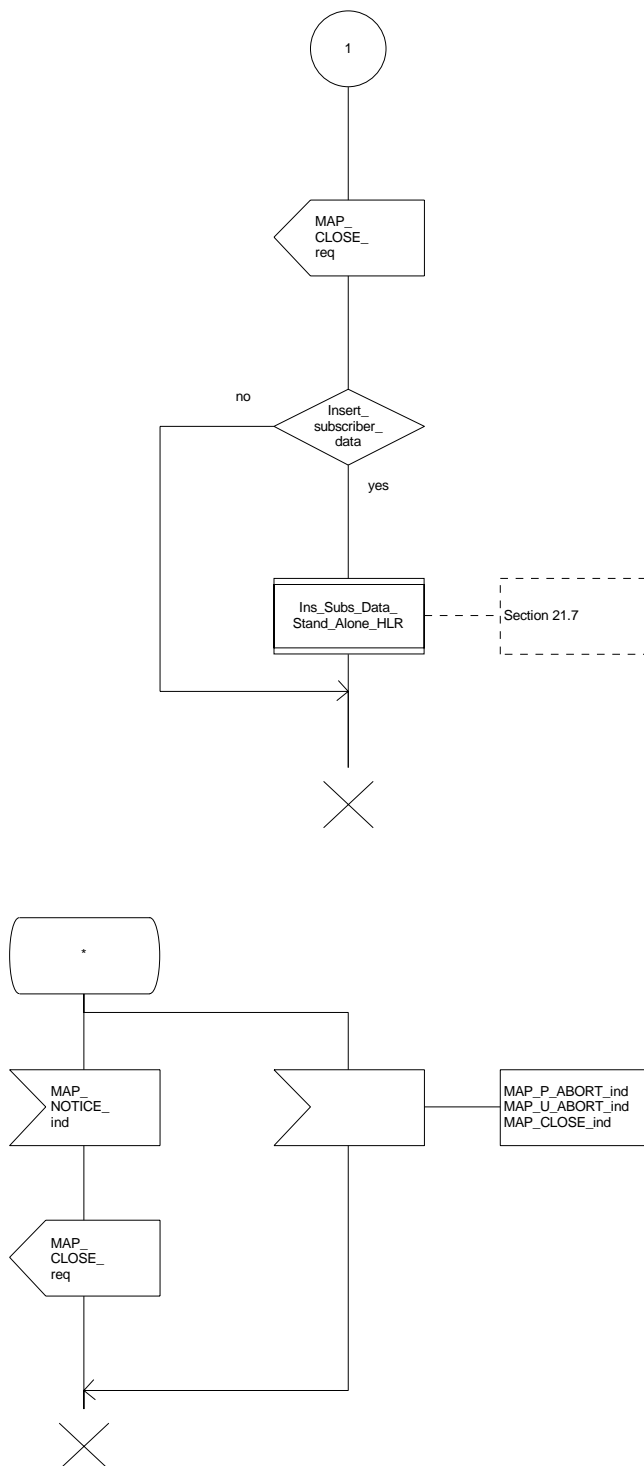


Figure 19.1.4/1 (sheet 2 of 2)



19.2 Erasure procedure

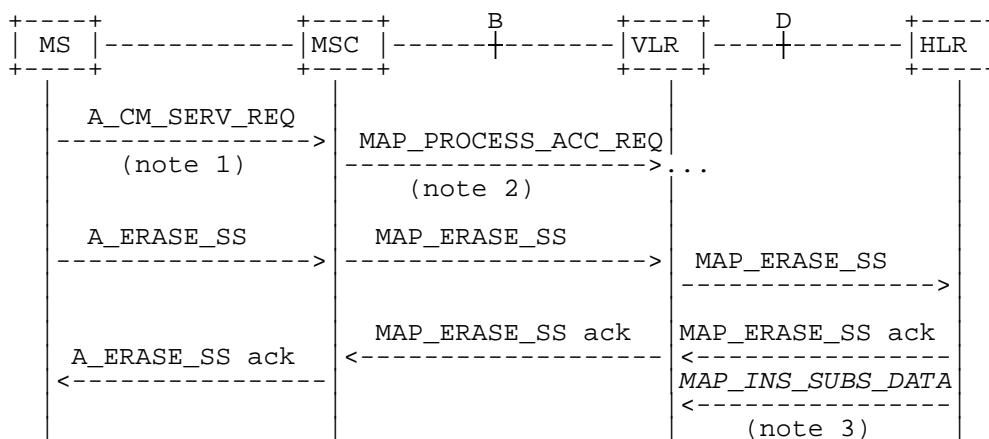
19.2.1 General

The erasure procedure is used to erase data related to a supplementary service in the HLR. The erasure procedure is a fully transparent communication between the MS and the HLR, except that some services may be invoked as a result of the procedure, as described in the subclauses below.

The erasure procedure is shown in figure 19.2.1/1.

The following services may be used:

MAP_PROCESS_ACCESS_REQUEST	(defined in subclauses 6 and 21);
MAP_TRACE_SUBSCRIBER_ACTIVITY	(defined in clauses 7 and 21);
MAP_PROVIDE_IMSI	(defined in clauses 6 and 21);
MAP_FORWARD_NEW_TMSI	(defined in clauses 6 and 21);
MAP_AUTHENTICATE	(defined in clauses 6 and 21);
MAP_SET_CIPHERING_MODE	(defined in clauses 6 and 21);
MAP_CHECK_IMEI	(defined in clauses 6 and 21);
MAP_READY_FOR_SM	(defined in clauses 10 and 21);
MAP_INSERT_SUBSCRIBER_DATA	(defined in clauses 6 and 21);
MAP_ERASE_SS	(defined in clause 9).



NOTE 1: For details of the procedure on the radio path, see GSM 04.08, 04.10, 04.8x and 04.9x. Services shown in dotted lines indicate the trigger provided by the signalling on the radio path, and the signalling triggered on the radio path.

NOTE 2: For details on the Process Access Request procedure, please refer to clause 21 in this ETS.

NOTE 3: Services printed in *italics* are optional.

Figure 19.2.1/1: Interfaces and services for supplementary service erasure

19.2.2 Procedures in the MSC

The MSC procedures for erasure are identical to those specified for registration in subclause 19.1.2. The text and diagrams in subclause 19.1.2 apply with all references to registration changed to erasure.

19.2.3 Procedures in the VLR

The VLR procedures for erasure are identical to those specified for registration in subclause 19.1.3. The text and diagrams in subclause 19.1.3 apply with all references to registration changed to erasure.

19.2.4 Procedures in the HLR

The HLR procedure for erasure is identical to those specified for registration in subclause 19.1.4. The text and diagrams in subclause 19.1.4 apply with all references to registration changed to erasure.

19.3 Activation procedure

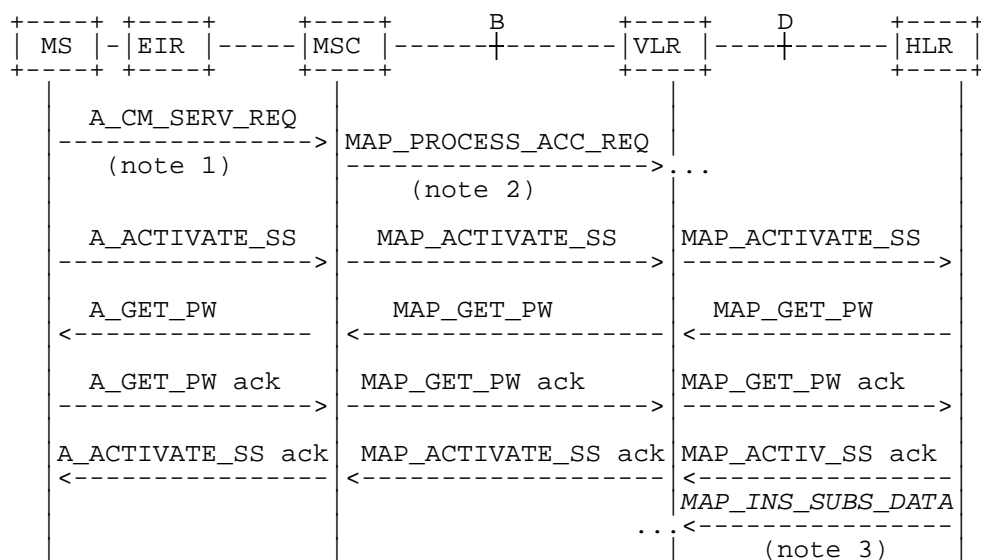
19.3.1 General

The activation procedure is used to activate a supplementary service in the HLR. The activation procedure is a fully transparent communication between the MS and the HLR, except that some services may be invoked as a result of the procedure, as described in the subclauses below.

The activation procedure is shown in figure 19.3.1/1.

The following services may be used:

MAP_PROCESS_ACCESS_REQUEST	(defined in clauses 6 and 21);
MAP_TRACE_SUBSCRIBER_ACTIVITY	(defined in clauses 7 and 21);
MAP_PROVIDE_IMSI	(defined in clauses 6 and 21);
MAP_FORWARD_NEW_TMSI	(defined in clauses 6 and 21);
MAP_AUTHENTICATE	(defined in clauses 6 and 21);
MAP_SET_CIPHERING_MODE	defined in clauses 6 and 21);
MAP_CHECK_IMEI	(defined in clauses 6 and 21);
MAP_READY_FOR_SM	(defined in clauses 10 and 21);
MAP_GET_PASSWORD	(defined in clause 9);
MAP_INSERT_SUBSCRIBER_DATA	(defined in clauses 6 and 21);
MAP_ACTIVATE_SS	defined in clause 9).



NOTE 1: For details of the procedure on the radio path, see GSM 04.08, 04.10, 04.8x and 04.9x. Services shown in dotted lines indicate the trigger provided by the signalling on the radio path, and the signalling triggered on the radio path.

NOTE 2: For details on the Process Access Request procedure, please refer to clause 21 in this ETS.

NOTE 3: Services printed in *italics* are optional.

Figure 19.3.1/1: Interfaces and services for supplementary service activation

19.3.2 Procedures in the MSC

The A_ACTIVATE_SS service indication received by the MAP user in the MSC contains the SS-Code and any parameters related to the supplementary service.

The MSC transfers the received information to the VLR in the MAP_ACTIVATE_SS request without checking the contents of the service indication. Rules for the mapping are described in GSM 09.11.

The MAP user may subsequently receive the MAP_GET_PASSWORD indication from the VLR. Upon receipt of this indication, the MSC sends the A_GET_PASSWORD message towards the MS and then awaits the response from the MS. When an A_GET_PASSWORD confirm message is received from the MS, the MSC initiates the MAP_GET_PASSWORD response towards the VLR without checking further the contents of the indication. Also see GSM 09.11.

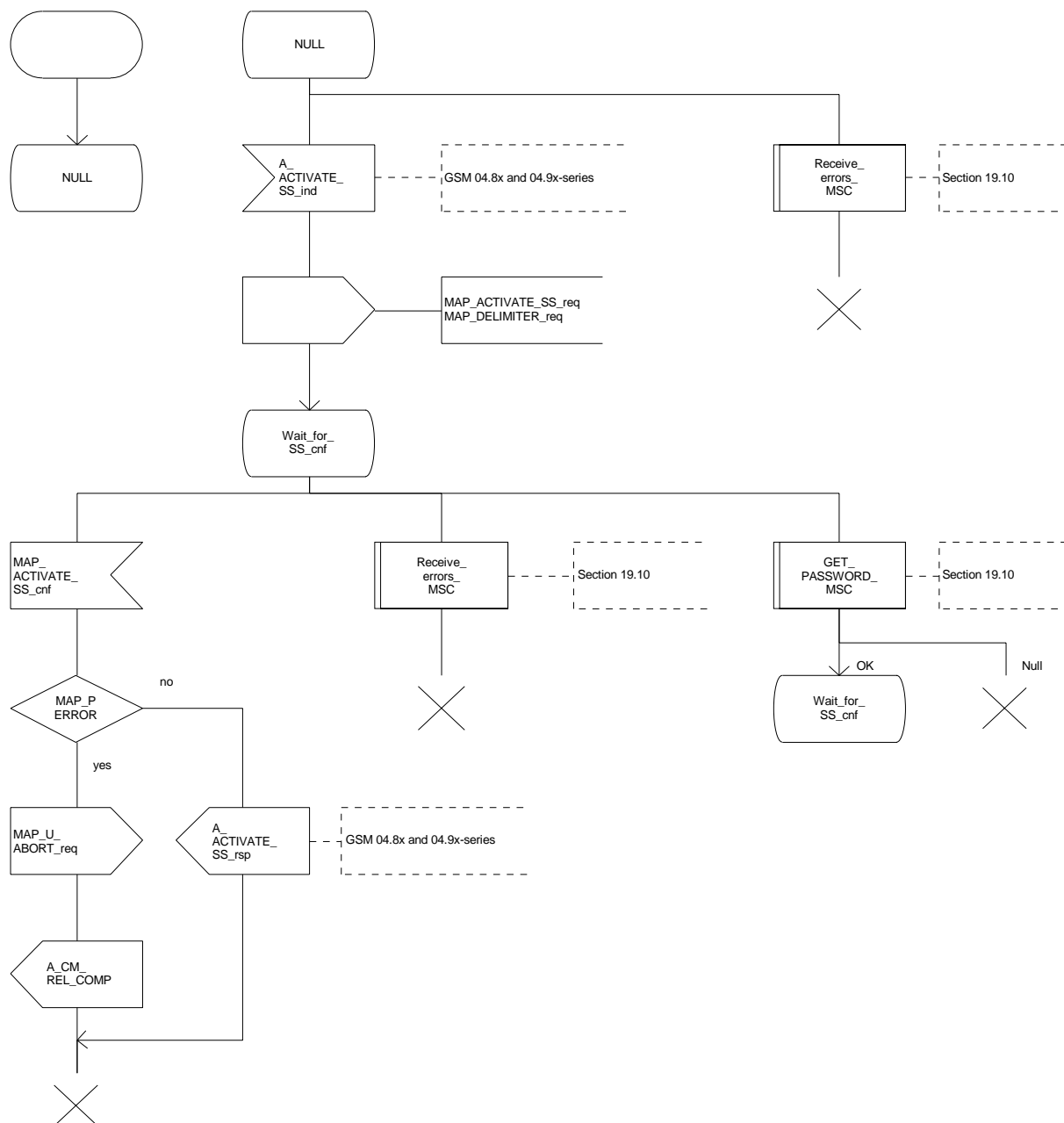
The MSC will receive a MAP_ACTIVATE_SS confirm from the VLR. The outcome of the procedure is reported to the MS in the A_ACTIVATE_SS response message, see GSM 04.8x, 04.9x and 09.11. Finally the SS connection is released.

For call independent SS operations, each message shall only contain a single component. Messages which contain more than one component will be stopped at the air interface (as specified in GSM 09.11).

The handling of MAP_P_ABORT, MAP_U_ABORT, MAP_NOTICE and unexpected MAP_CLOSE or A_CM_RELEASE in this procedure is identical to the handling in the Registration procedure in the MSC, see subclause 19.1.2 of this ETS.

The activation procedure in the MSC is shown in figure 19.3.2/1.

Figure 19.3.2/1



19.3.3 Procedures in the VLR

Supplementary service activation

When receiving the MAP_ACTIVATE_SS indication, the MAP user in the VLR transfers the information to the HLR in the MAP_ACTIVATE_SS request without checking the contents of the service indication.

The VLR may then receive the MAP_GET_PASSWORD indication. This information is transferred to the MSC in the MAP_GET_PASSWORD request. If a MAP_GET_PASSWORD confirm primitive is received from the MSC, the VLR initiates the MAP_GET_PASSWORD response towards the HLR.

The VLR will receive the MAP_ACTIVATE_SS confirm from the HLR. The MAP user in the VLR shall transfer the information contained in this primitive to the MSC in the MAP_ACTIVATE_SS response without checking its contents.

For call independent SS operations, each message shall only contain a single component. Messages which contain more than one component will be stopped at the air interface (as specified in GSM 09.11).

Error handling

The handling of MAP_P_ABORT, MAP_U_ABORT, MAP_NOTICE and unexpected MAP_CLOSE in this procedure is identical to the handling in the Registration procedure in the VLR, see subclause 19.1.3 of this ETS.

The activation procedure in the VLR is shown in figure 19.3.3/1.

Figure 19.3.3/1 (sheet 1 of 2)

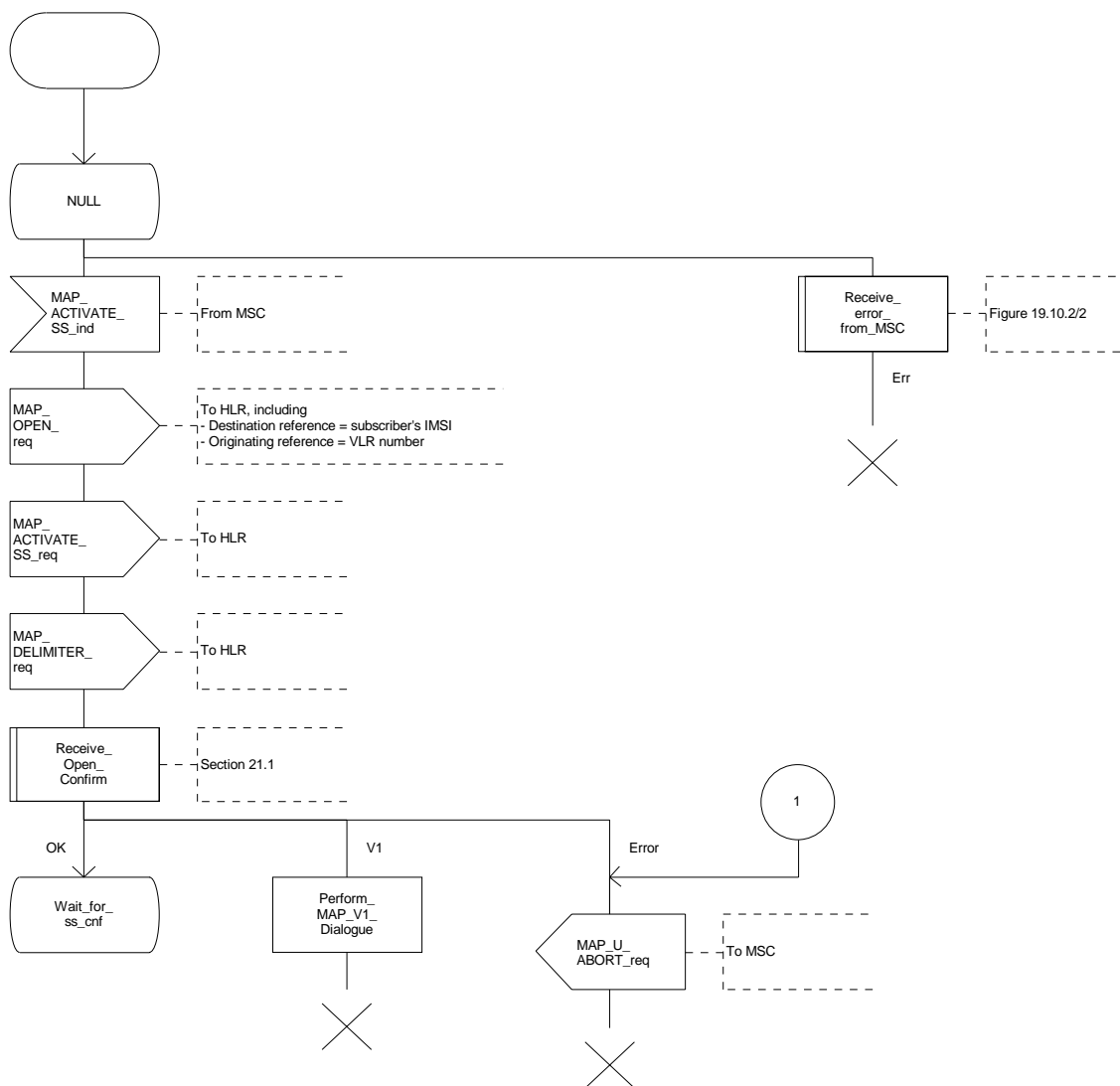
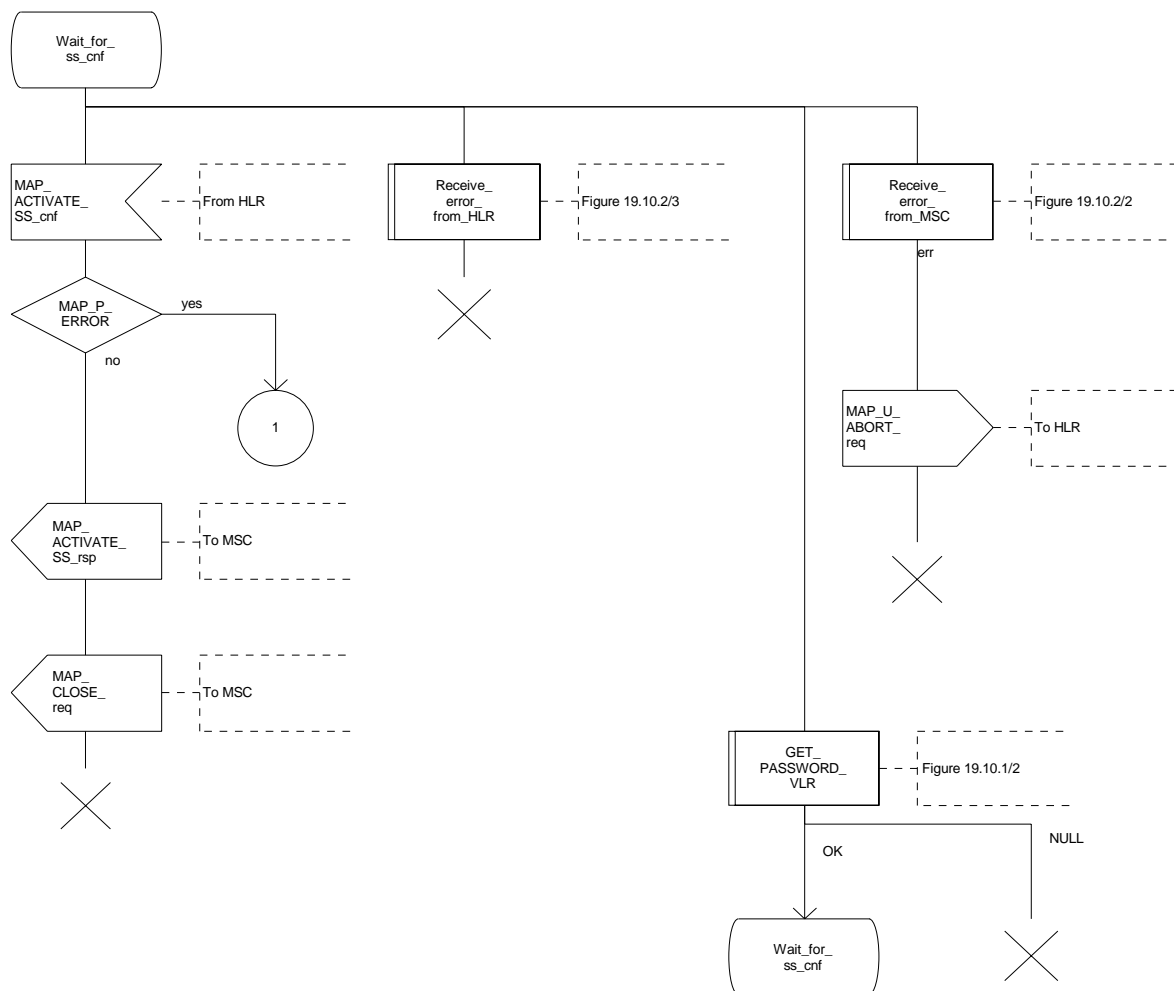


Figure 19.3.3/1 (sheet 2 of 2)



19.3.4 Procedures in the HLR

The procedure in the HLR is initiated when it receives a MAP_ACTIVATE_SS indication.

The HLR acts as follows:

- if the operator has barred the subscriber from access to supplementary services, the Call Barred error should be returned to the VLR. The parameter "operatorBarring" shall be included with the error.

The supplementary service request shall then be processed according to GSM 03.11 and the 03.8x and 03.9x-series of technical specifications. This handling may lead to either a successful result, a partially successful result, or an error being returned.

During the handling of activation, the get password procedure may be initiated (as specified in GSM 03.11). This will involve the sending of a MAP_GET_PASSWORD request to the VLR.

For call independent SS operations, each message shall only contain a single component. Messages which contain more than one component will be stopped at the air interface (as specified in GSM 09.11):

- if the VLR is to be updated after the supplementary service activation, the MAP_INSERT_SUBS_DATA_HLR process is initiated;
- handling of receipt of MAP_P_ABORT, MAP_U_ABORT or MAP_CLOSE indications from the VLR is identical to their handling in the registration procedure, see subclause 19.1.4 above.

The activation procedure in the HLR is shown in figure 19.3.4/1.

Figure 19.3.4/1 (sheet 1 of 2)

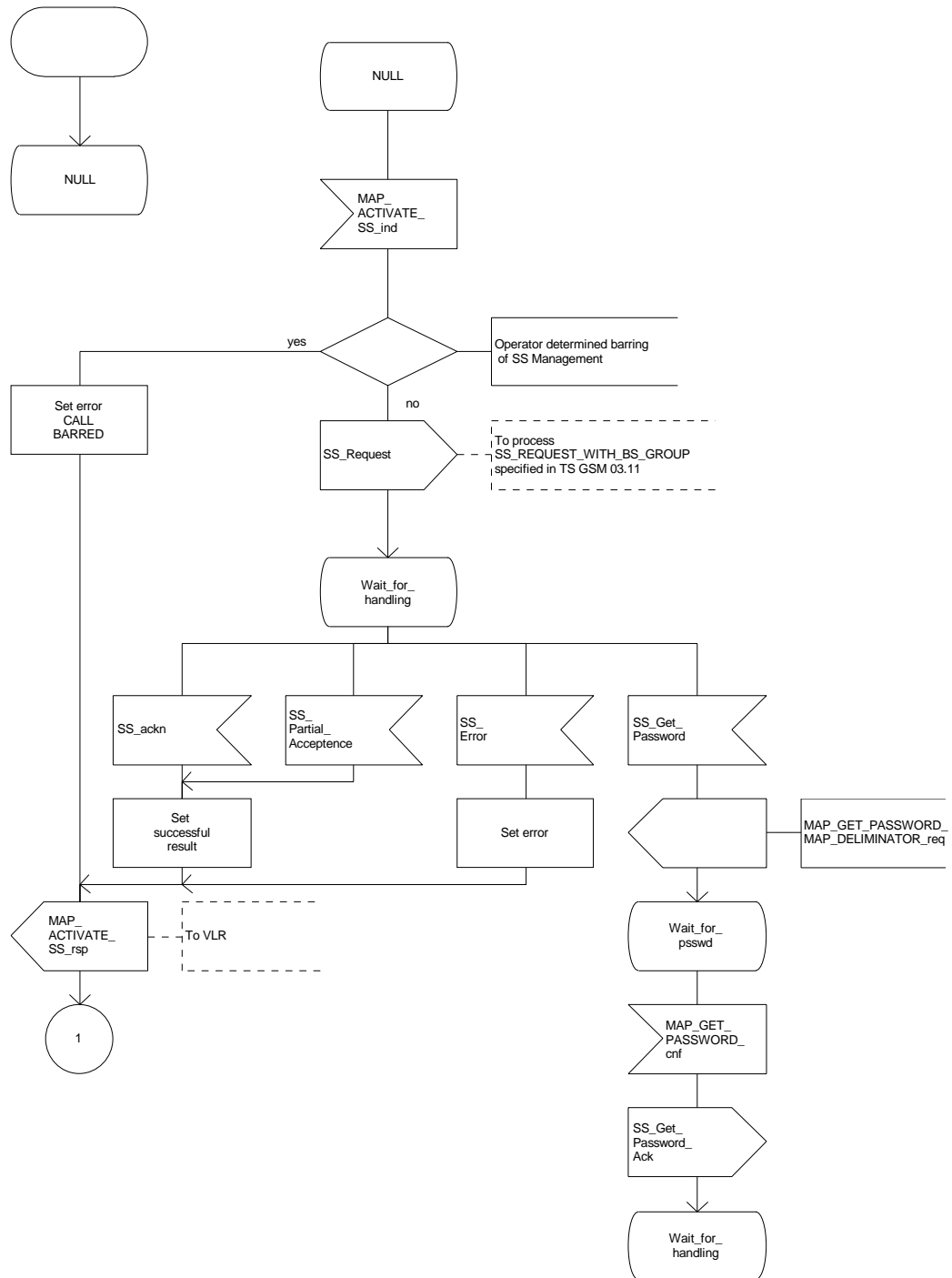
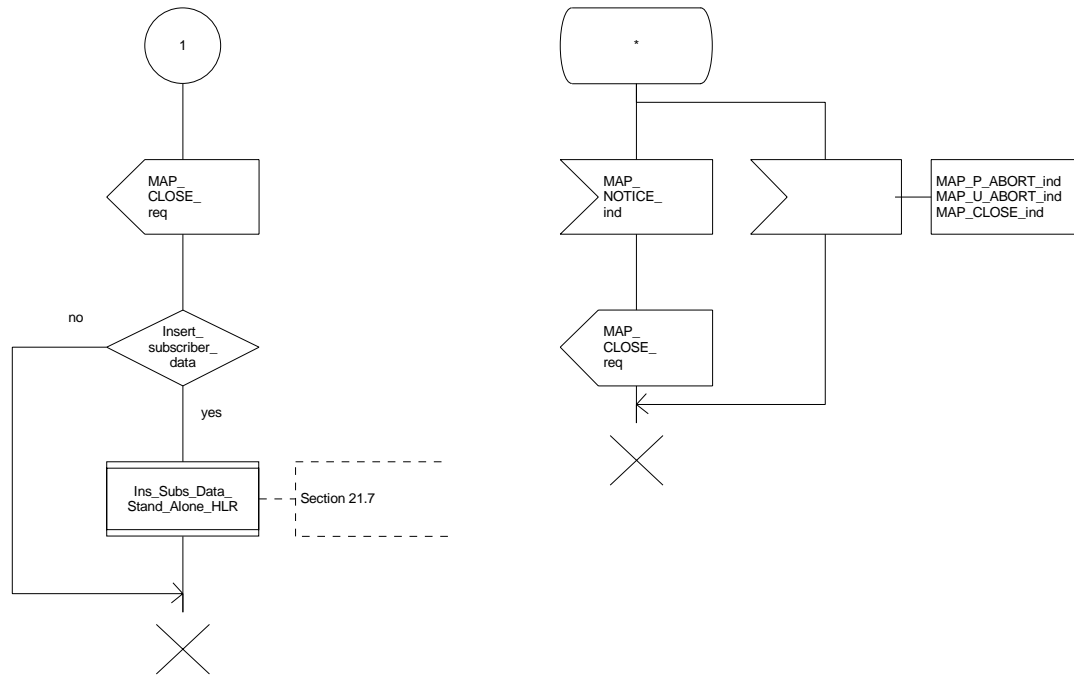


Figure 19.3.4/1 (sheet 2 of 2)



19.4 Deactivation procedure

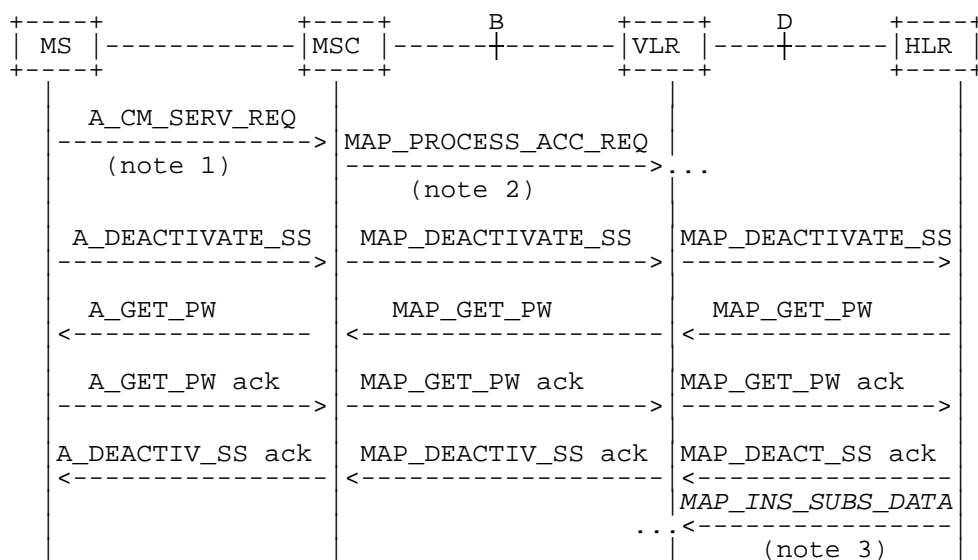
19.4.1 General

The deactivation procedure is used to deactivate a supplementary service in the HLR. The deactivation procedure is a fully transparent communication between the MS and the HLR, except that some services may be invoked as a result of the procedure, as described in the subclauses below.

The deactivation procedure is shown in figure 19.4.1/1.

The following services may be used:

MAP_PROCESS_ACCESS_REQUEST	(defined in clauses 6 and 21);
MAP_TRACE_SUBSCRIBER_ACTIVITY	(defined in clauses 7 and 21);
MAP_PROVIDE_IMSI	(defined in clauses 6 and 21);
MAP_FORWARD_NEW_TMSI	(defined in clauses 6 and 21);
MAP_AUTHENTICATE	(defined in clauses 6 and 21);
MAP_SET_CIPHERING_MODE	(defined in clauses 6 and 21);
MAP_CHECK_IMEI	(defined in clauses 6 and 21);
MAP_READY_FOR_SM	(defined in clauses 10 and 21);
MAP_GET_PASSWORD	(defined in clause 9);
MAP_INSERT_SUBSCRIBER_DATA	(defined in clauses 6 and 21);
MAP_DEACTIVATE_SS	(defined in clause 9).



NOTE 1: For details of the procedure on the radio path, see GSM 04.08, 04.10, 04.8x and 04.9x. Services shown in dotted lines indicate the trigger provided by the signalling on the radio path, and the signalling triggered on the radio path.

NOTE 2: For details on the Process Access Request procedure, please refer to clause 21 in this ETS.

NOTE 3: Services printed in *italics* are optional.

Figure 19.4.1/1: Interfaces and services for supplementary service deactivation

19.4.2 Procedures in the MSC

The MSC procedures for deactivation are identical to those specified for activation in subclause 19.3.2. The text and diagrams in subclause 19.3.2 apply with all references to activation changed to deactivation.

19.4.3 Procedures in the VLR

The VLR procedures for deactivation are identical to those specified for activation in subclause 19.3.3. The text and diagrams in subclause 19.3.3 apply with all references to activation changed to deactivation.

19.4.4 Procedures in the HLR

The HLR procedures for deactivation are identical to those specified for activation in subclause 19.3.4. The text and diagrams in subclause 19.3.4 apply with all references to activation changed to deactivation.

19.5 Interrogation procedure

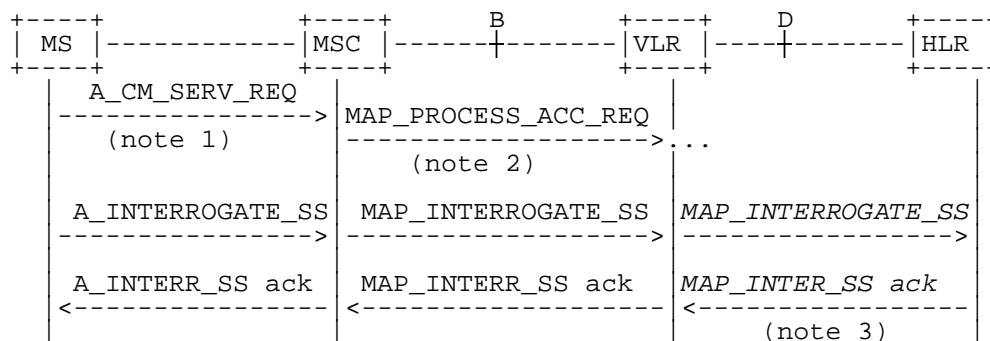
19.5.1 General

The interrogation procedure is used to retrieve information related to a supplementary service from the VLR or the HLR. It is the VLR which decides whether an interrogation request should be forwarded to the HLR or not. Some non-supplementary service related services may be invoked as a result of the procedure, as described in the subclauses below.

The interrogation procedure is shown in figure 19.5.1/1.

The following services may be used:

MAP_PROCESS_ACCESS_REQUEST	(defined in clauses 6 and 21);
MAP_TRACE_SUBSCRIBER_ACTIVITY	(defined in clauses 7 and 21);
MAP_PROVIDE_IMSI	(defined in clauses 6 and 21);
MAP_FORWARD_NEW_TMSI	(defined in clauses 6 and 21);
MAP_AUTHENTICATE	(defined in clauses 6 and 21);
MAP_SET_CIPHERING_MODE	(defined in clauses 6 and 21);
MAP_CHECK_IMEI	(defined in clauses 6 and 21);
MAP_READY_FOR_SM	(defined in clauses 10 and 21);
MAP_INTERROGATE_SS	(defined in clause 9).



NOTE 1: For details of the procedure on the radio path, see GSM 04.08, 04.10, 04.8x and 04.9x. Services shown in dotted lines indicate the trigger provided by the signalling on the radio path, and the signalling triggered on the radio path.

NOTE 2: For details on the Process Access Request procedure, please refer to clause 21 in this ETS.

NOTE 3: Services printed in italics are optional.

Figure 19.5.1/1: Interfaces and services for supplementary service interrogation

19.5.2 Procedures in the MSC

The MSC procedures for interrogation are identical to those specified for registration in subclause 19.1.2. The text and diagrams in subclause 19.1.2 apply with all references to registration changed to interrogation.

19.5.3 Procedures in the VLR

Supplementary service interrogation

When receiving the MAP_INTERROGATE_SS indication, the MAP user acts as follows:

- if the operator has barred the subscriber from access to supplementary services, the error Call Barred is returned to the MSC. The parameter "operatorBarring" shall be included with the error.

The interrogation is either answered by the VLR or by the HLR, depending on the service interrogated.

a) Interrogation to be handled by the VLR

The supplementary service request shall then be processed according to GSM 03.11 and the 03.8x and 03.9x-series of technical specifications. This handling may lead to either a successful result, a partially successful result, or an error being returned.

For call independent SS operations, each message shall only contain a single component. Messages which contain more than one component will be stopped at the air interface (as specified in GSM 09.11).

b) Interrogation to be handled by HLR

If the interrogation is to be handled by the HLR, on receiving the MAP_INTERROGATE_SS indication, the MAP user in the VLR transfers the information to the HLR in the MAP_INTERROGATE_SS request without further checking the contents of the service indication.

The VLR will receive the MAP_INTERROGATE_SS confirm from the HLR. The MAP user in the VLR shall transfer the information contained in this primitive to the MSC in the MAP_INTERROGATE_SS response without checking its contents.

For call independent SS operations, each message shall only contain a single component. Messages which contain more than one component will be stopped at the air interface (as specified in GSM 09.11).

Error handling

Handling of MAP_P_ABORT, MAP_U_ABORT, MAP_NOTICE and unexpected MAP_CLOSE in this procedure is identical to the handling in the Registration procedure in the VLR, subclause 19.1.3. The Interrogation procedure is described in figure 19.5.3/1.

Figure 19.5.3/1 (sheet 1 of 3)

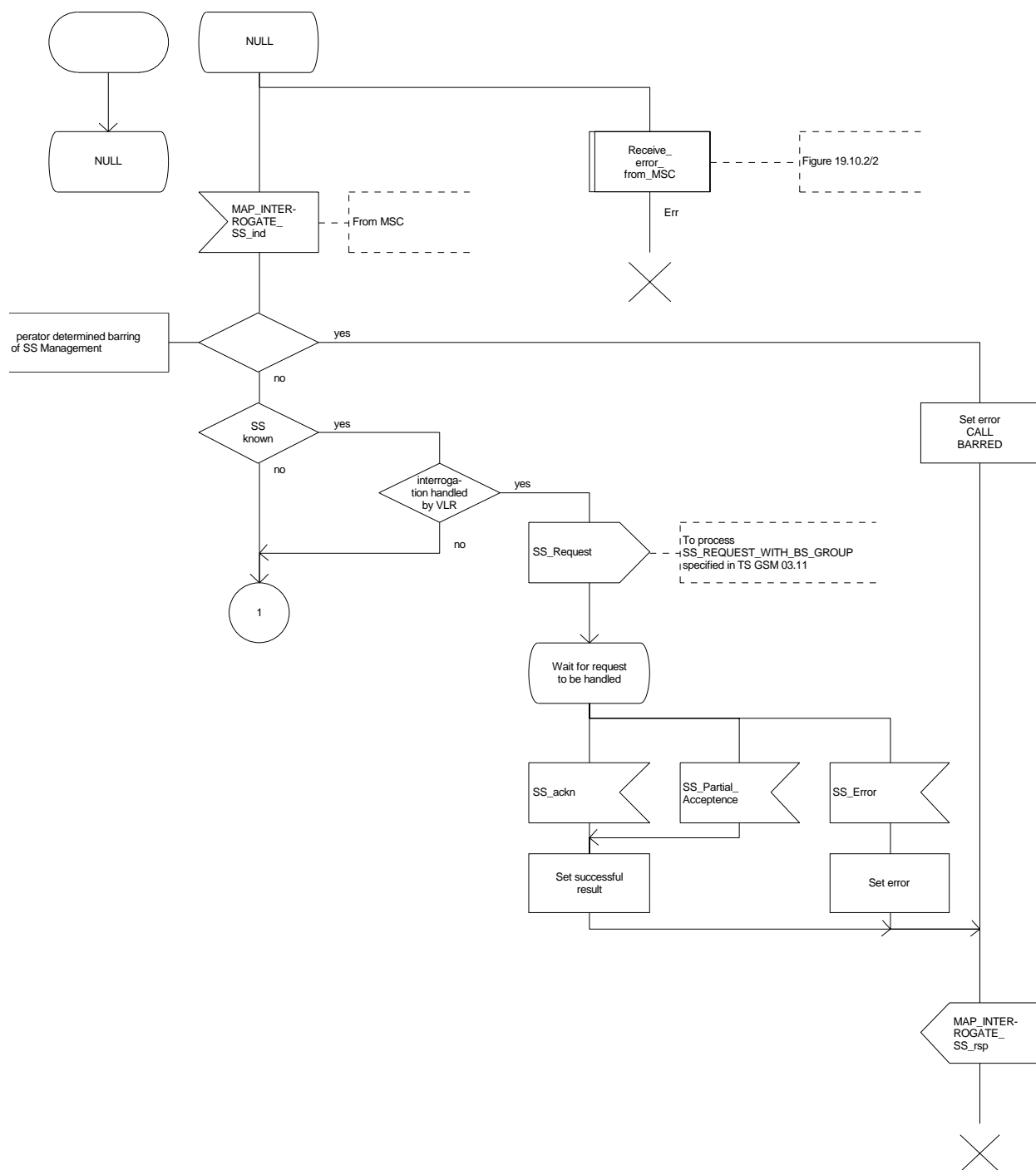


Figure 19.5.3/1 (sheet 2 of 3)

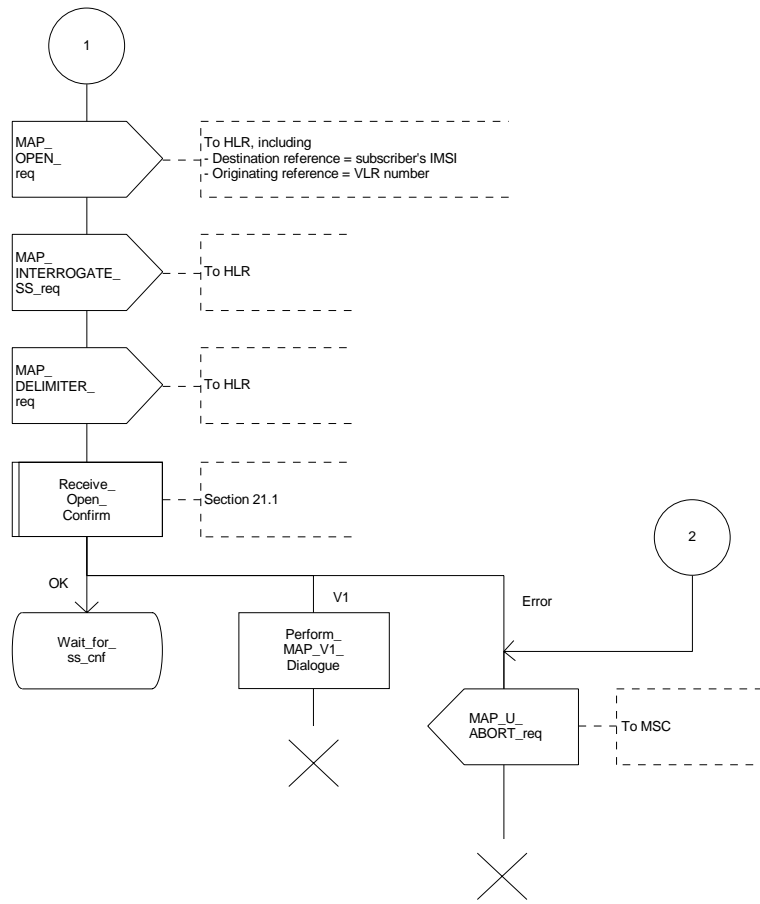
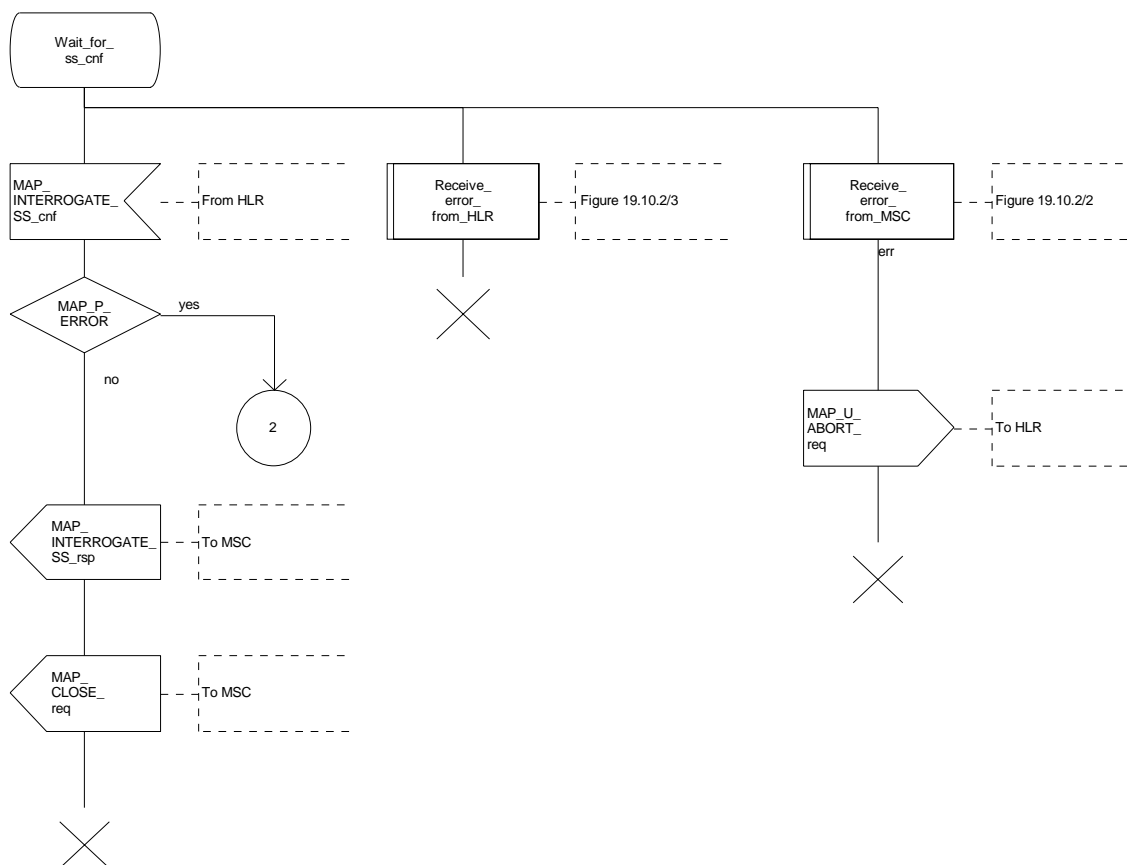


Figure 19.5.3/1 (sheet 3 of 3)



19.5.4 Procedures in the HLR

When receiving the MAP_INTERROGATE_SS indication, the MAP user acts as follows:

- if the operator has barred the subscriber from access to supplementary services, the error Call Barred is returned to the MSC. The parameter "operatorBarring" shall be included with the error;
- if the supplementary service is not supported in HLR the error Unexpected Data Value is returned to the VLR.

The interrogation is either answered by the VLR or by the HLR, depending on the service interrogated.

a) Interrogation to be handled by the VLR

If the interrogation procedure should have been answered by the VLR, then the HLR assumes that the VLR does not support the interrogated supplementary service, and returns the SS Not Available error to the VLR.

b) Interrogation to be handled by HLR

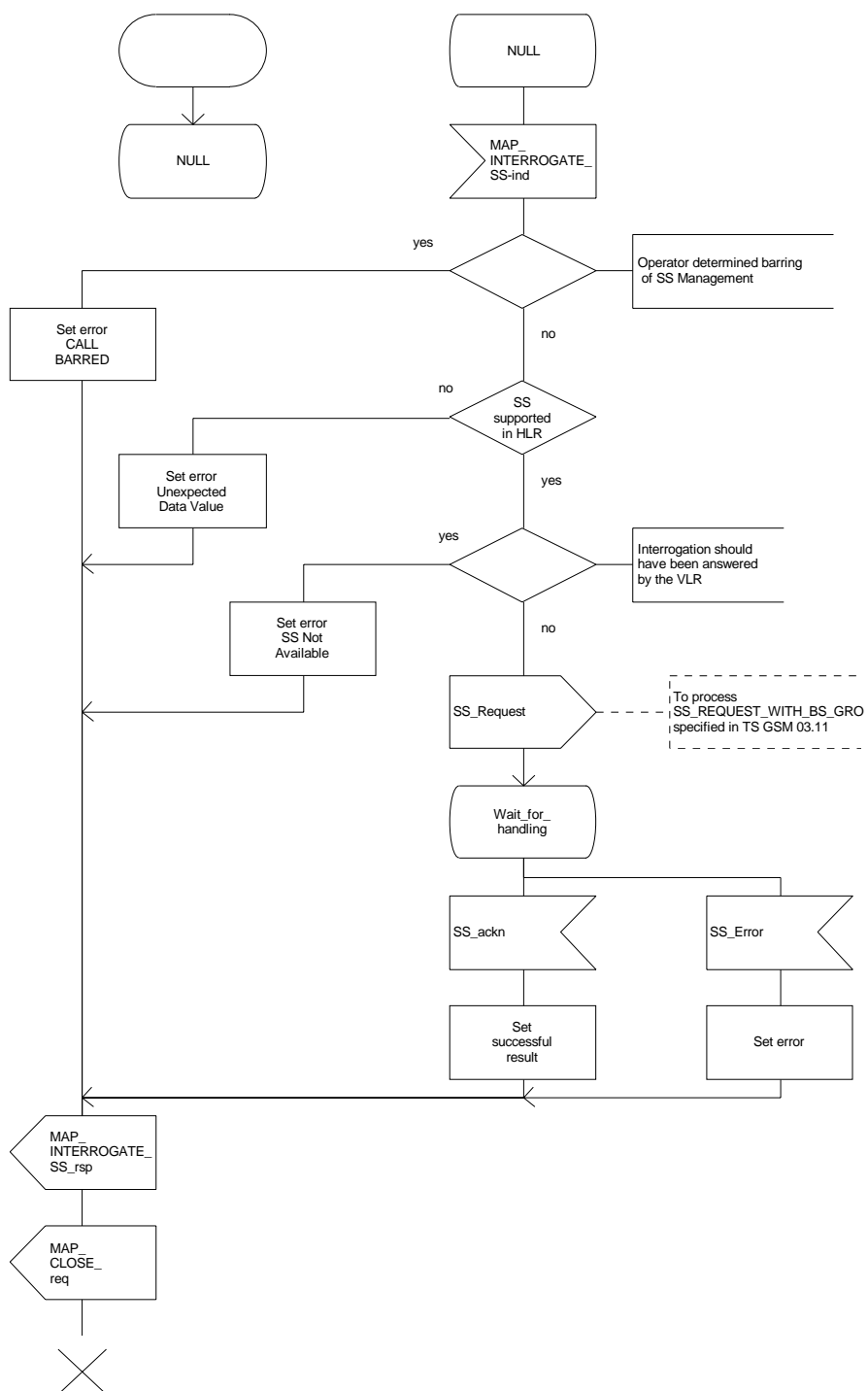
The supplementary service request shall be processed according to GSM 03.11 and the 03.8x and 03.9x-series of technical specifications. This handling may lead to either a successful result or an error being returned.

For call independent SS operations, each message shall only contain a single component.

Error handling

Handling of MAP_P_ABORT, MAP_U_ABORT, MAP_NOTICE and unexpected MAP_CLOSE in this procedure is identical to the handling in the Registration procedure in the VLR, subclause 19.1.3. The Interrogation procedure is described in figure 19.5.4/1.

Figure 19.5.4/1



19.6 Invocation procedure

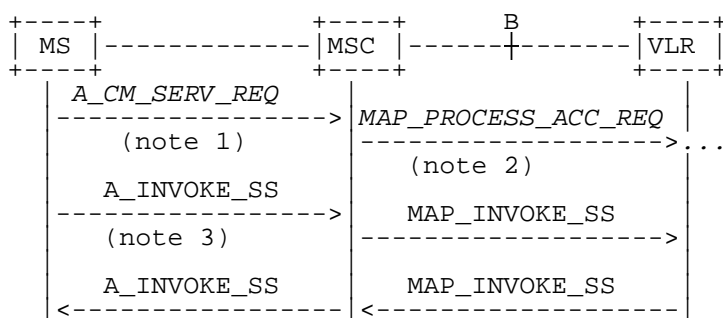
19.6.1 General

The invocation procedure is used to check subscription data in the VLR for certain supplementary services which are invoked after the call set-up phase is finished. For invocation of supplementary services which are invoked during the call set-up phase, please refer to the Call Handling procedure descriptions.

The invocation procedure is shown in figure 19.6.1/1. Note that some optional services may be invoked in connection with this procedure, as described in the subclause below.

The following services are used:

MAP_PROCESS_ACCESS_REQUEST	(defined in clauses 6 and 21);
MAP_TRACE_SUBSCRIBER_ACTIVITY	(defined in clauses 7 and 21);
MAP_PROVIDE_IMSI	(defined in clauses 6 and 21);
MAP_FORWARD_NEW_TMSI	(defined in clauses 6 and 21);
MAP_AUTHENTICATE	(defined in clauses 6 and 21);
MAP_SET_CIPHERING_MODE	(defined in clauses 6 and 21);
MAP_CHECK_IMEI	(defined in clauses 6 and 21);
MAP_READY_FOR_SM	(defined in clauses 10 and 21);
MAP_INVOKE_SS	(defined in clause 9).



NOTE 1: For details of the procedure on the radio path, see GSM 04.08, 04.10, 04.8x and 04.9x. Services shown in dotted lines indicate the trigger provided by the signalling on the radio path, and the signalling triggered on the radio path.

NOTE 2: For details on the Process Access Request procedure, please refer to clause 21 in this ETS.

NOTE 3: A_INVOKESS is a generic message to illustrate any supplementary service invocation request message on the air interface, e.g. BuildMPTY, see GSM 04.80.

Figure 19.6.1/1: Interfaces and services for supplementary service invocation

19.6.2 Procedures in the MSC

Process access request

Before the Call Hold or Multi-Party supplementary services can be invoked, a CC connection must be established between the MS and the MSC as described in GSM 04.08 and the Call Handling procedure descriptions within this ETS.

When an A_INVOKE_SS request message arrives at the MSC during a call (as described in GSM 04.10, 04.8x and 04.9x-series of technical specifications), then if control of subscription to the invoked supplementary service is required, the MSC initiates the process access request procedure towards the VLR as described in clause 21 of this ETS.

Supplementary service invocation

If the Process Access Request procedure towards the VLR is successful, the MSC shall forward a MAP_INVOKE_SS service request towards the VLR. This request shall contain the SS-Code of the supplementary service to be invoked, and possibly the Basic service code. Mapping from the A_INVOKE_SS to this service request is described in GSM 09.11.

The MSC will receive a MAP_INVOKE_SS confirm from the VLR. If the outcome of the service is successful (i.e. the service confirm is empty), the MSC will invoke the requested supplementary service as described in GSM 02.8x-series, 03.8x and 03.9x-series of technical specifications. If the outcome of the service is unsuccessful, the MSC shall send an appropriate A_INVOKE_SS response towards the MS. The structure of this message is described in GSM 09.11 and 04.8x and 04.9x-series of technical specifications.

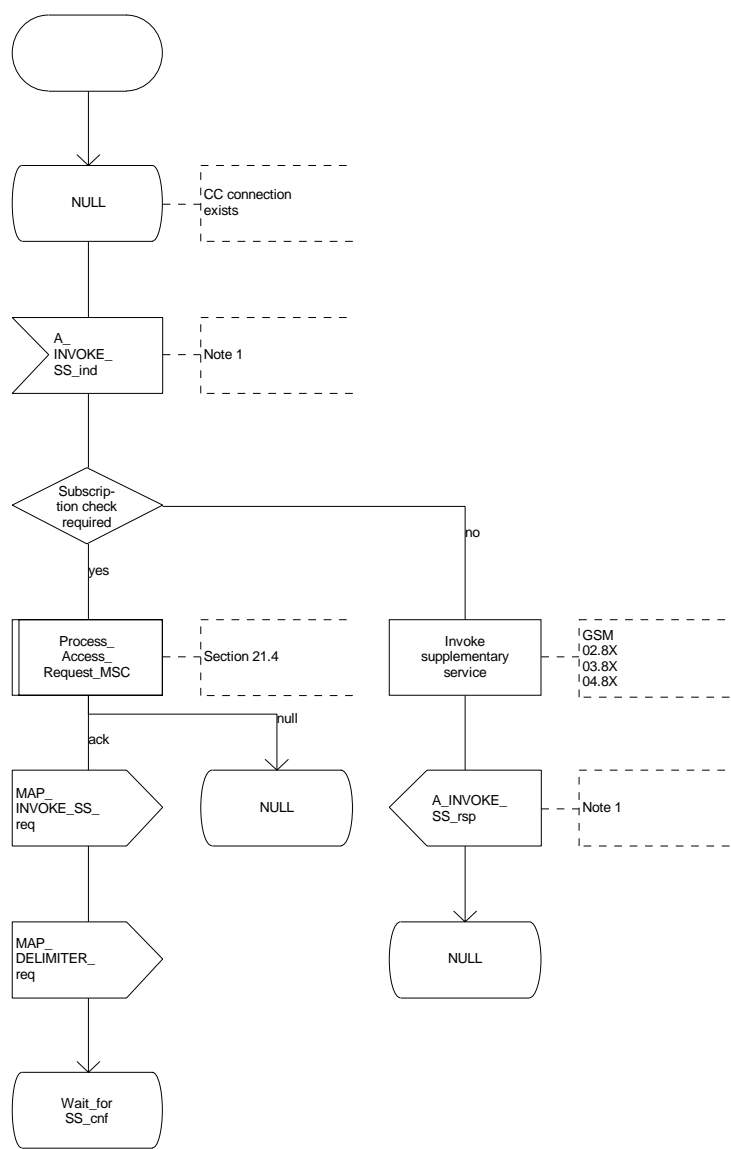
Error handling

If at any time during this procedure a MAP_P_ABORT, MAP_U_ABORT, MAP_NOTICE or MAP_CLOSE indication concerning the process is received from the VLR, the process is terminated. If a MAP_NOTICE indication was received from the VLR, the VLR dialogue must also be aborted by sending a MAP_U_ABORT request indicating Procedure error towards the VLR. Possible signalling to the MS is described in GSM 04.10.

If an A_CM_RELEASE indication is received from the MS, all open transactions are released using the MAP_U_ABORT request indicating application procedure cancellation; the process terminates.

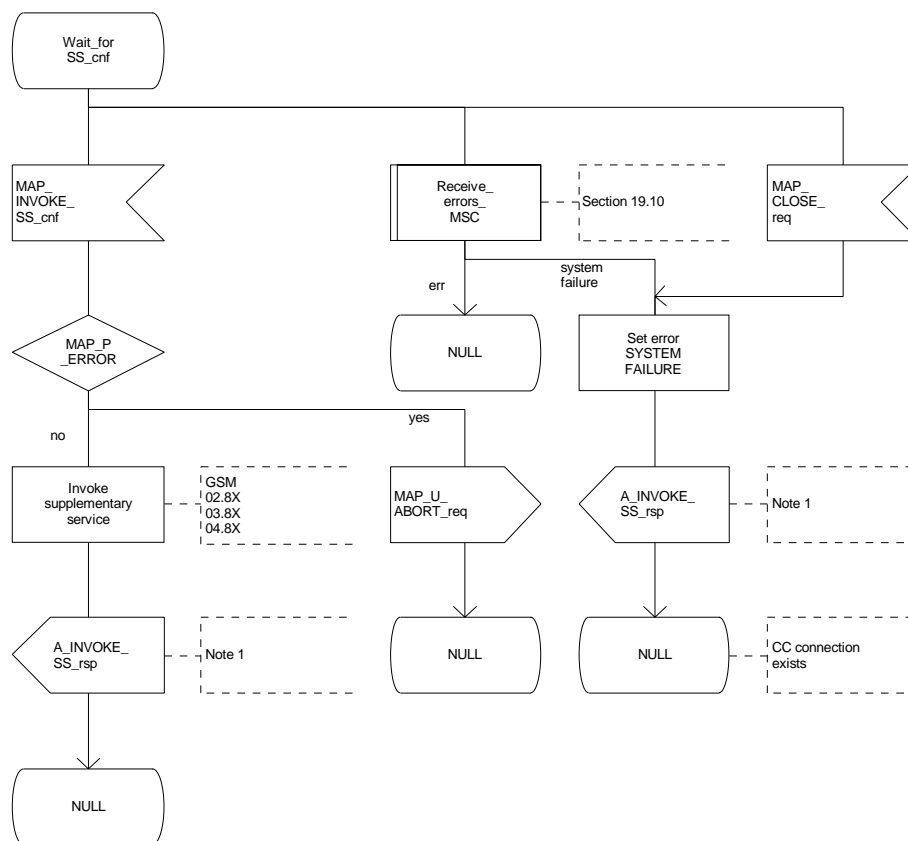
The invocation procedure in the MSC is shown in figure 19.6.2/1.

Figure 19.6.2/1 (sheet 1 of 2)



Note 1: Fictitious signal to indicate receipt/sending of SS invocation invoke component on the air interface (eg. BuildMPY). Described in GSM 04.8X and 09.11.

Figure 19.6.2/1 (sheet 2 of 2)



Note 1: Fictitious signal to indicate receipt/ sending of SS invocation invoke component on the air interface, (eg. BuildMPY). Described in GSM 04.8X and 09.11.

19.6.3 Procedures in the VLR

Process Access Request

When receiving the MAP_PROCESS_ACCESS_REQUEST indication, the VLR acts as described in clause 21 of this ETS.

Supplementary service invocation

When receiving the MAP_INVOKE_SS indication, the MAP user acts as follows:

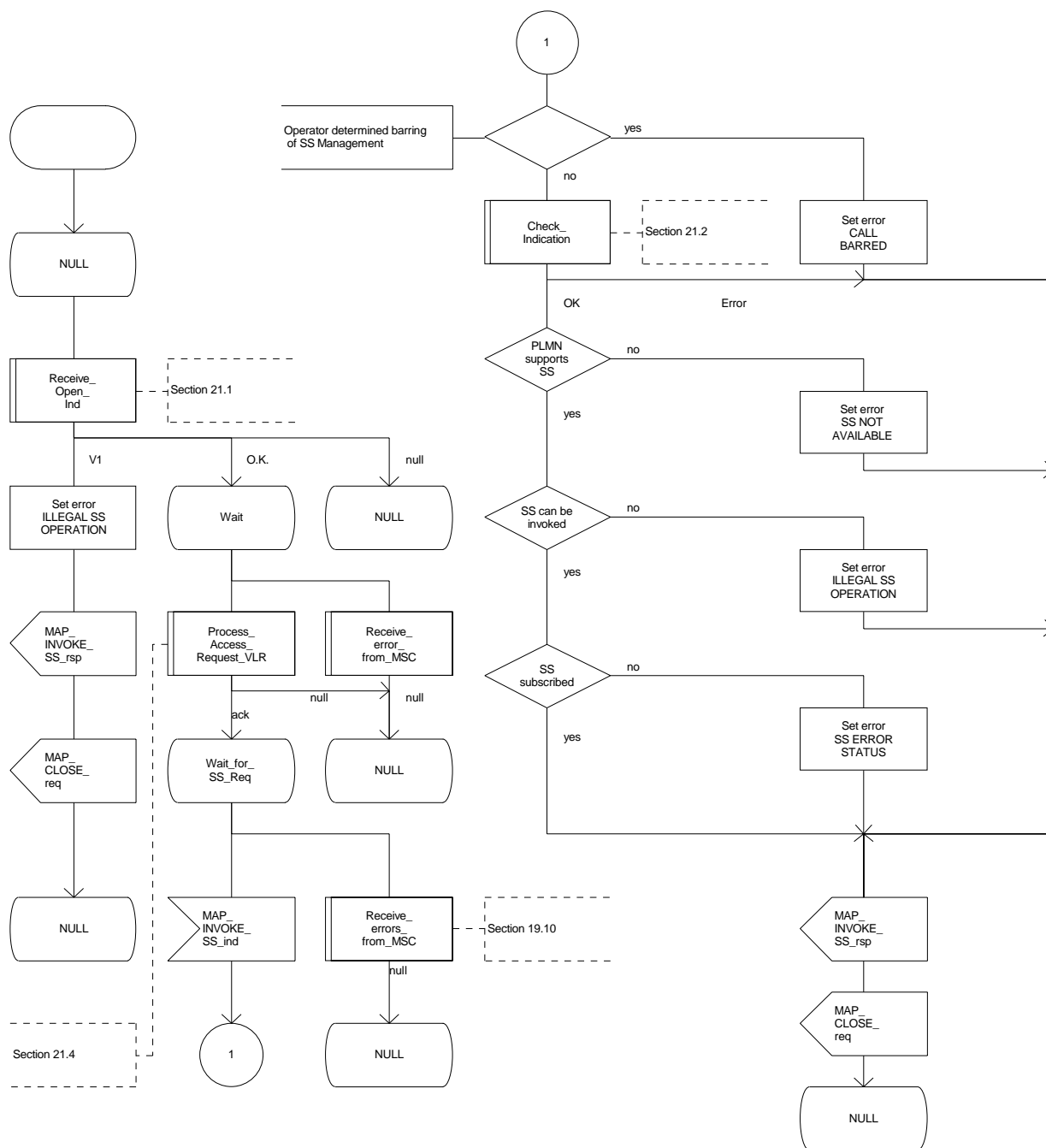
- if the operator has barred the subscriber from access to supplementary services, the error "Call Barred" is returned to the MSC. The parameter "operatorBarring" shall be included with the error;
- if any irrelevant information elements (according to the service description) or invalid information element values are present in the service request, then the unexpected data value error is returned to the MSC in the MAP_INVOKE_SS response;
- if the VLR does not support the invoked supplementary service then the VLR shall respond with the SS Not Available error;
- if the requested supplementary service cannot be invoked by subscriber actions, then the VLR shall respond with the Illegal SS Operation error;
- if the subscriber is not provided with (i.e. subscribed to) the requested supplementary service, then the SS error status error (possibly including the SS-Status as parameter) is returned to the MSC in the MAP_INVOKE_SS response.

If all checks are passed the VLR returns an empty MAP_INVOKE_SS response to the MSC, thus indicating that the invocation request was accepted.

If at any time during this procedure a MAP_P_ABORT, MAP_U_ABORT, MAP_NOTICE or unexpected MAP_CLOSE indication concerning the process is received from the MSC, the process terminates. If a MAP_NOTICE indication was received from the MSC, that dialogue must be aborted by sending a MAP_U_ABORT request indicating Procedure error towards the MSC. The process terminates.

The invocation procedure in the VLR is shown in figure 19.6.3/1.

Figure 19.6.3/1



19.7 Password registration procedure

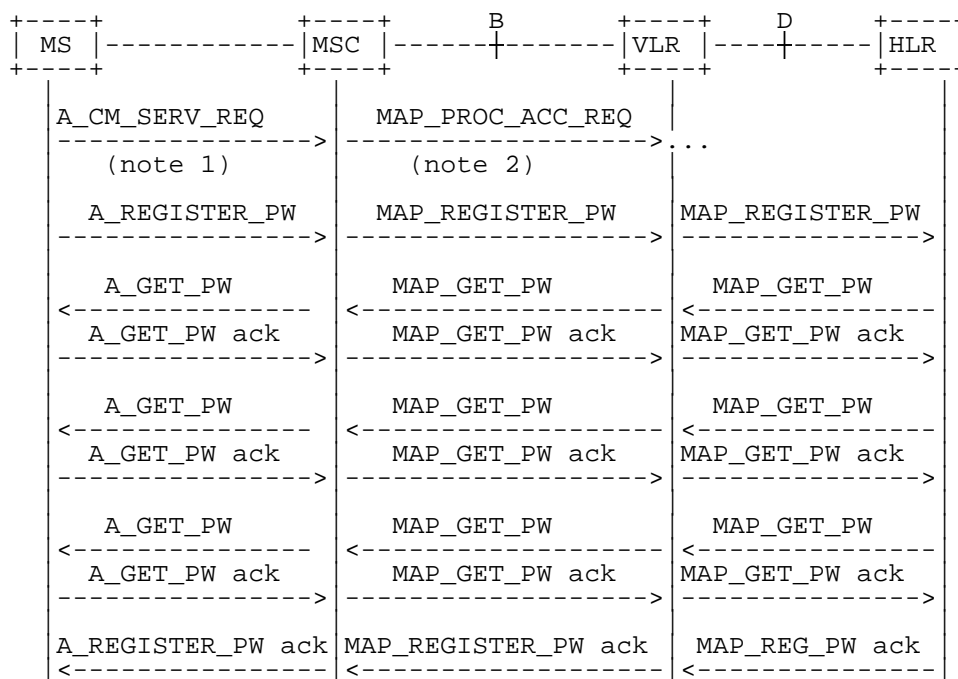
19.7.1 General

The password registration procedure is used to register a password in the HLR. The password registration procedure is a fully transparent communication between the MS and the HLR, except that some services may be invoked as a result of the procedure, as described below.

The password registration procedure is shown in figure 19.7.1/1.

The following services may be used:

MAP_PROCESS_ACCESS_REQUEST	(defined in clauses 6 and 21);
MAP_TRACE_SUBSCRIBER_ACTIVITY	(defined in clauses 7 and 21);
MAP_PROVIDE_IMSI	(defined in clauses 6 and 21);
MAP_FORWARD_NEW_TMSI	(defined in clauses 6 and 21);
MAP_AUTHENTICATE	(defined in clauses 6 and 21);
MAP_SET_CIPHERING_MODE	(defined in clauses 6 and 21);
MAP_CHECK_IMEI	(defined in clauses 6 and 21);
MAP_READY_FOR_SM	(defined in clauses 10 and 21);
MAP_GET_PASSWORD	(defined in clause 9).



NOTE 1: For details of the procedure on the radio path, see GSM 04.08, 04.10, 04.8x and 04.9x. Services shown in dotted lines are triggers/ triggered signalling on the radio path.

NOTE 2: For details on the Process Access Request procedure, please refer to clause 21 in this ETS.

NOTE 3: Use of each of the three MAP_GET_PASSWORD operations is described in subclause 19.7.4.

Figure 19.7.1/1: Interfaces and services for supplementary service password registration

19.7.2 Procedures in the MSC

The password registration procedure in the MSC is identical to that for activation specified in subclause 19.3.2. All the text and diagrams in subclause 19.3.2 apply with all references to activation changed to password registration.

19.7.3 Procedures in the VLR

The password registration procedure in the VLR is identical to that for activation specified in subclause 19.3.3. All the text and diagrams in subclause 19.3.3 apply with all references to activation changed to password registration.

19.7.4 Procedures in the HLR

The procedure in the HLR is initiated when it receives a MAP_REGISTER_PASSWORD indication.

The HLR acts as follows:

- if the operator has barred the subscriber for access to supplementary services, the Call Barred error is returned to the VLR. The parameter "operatorBarring" shall be included with the error;
- if any irrelevant information elements (according to the service description) or invalid information element values are present, then the unexpected data value error is returned to the VLR in the response. This error should thus be returned if the SS-Code provided by the mobile subscriber is not allocated.

The HLR shall then process the MAP_REGISTER_PASSWORD indication as specified in GSM 03.11. During the handling of password registration, the password procedure will be initiated (as specified in GSM 03.11) This will involve the sending of MAP_GET_PASSWORD requests to the VLR.

- Handling of receipt of MAP_P_ABORT, MAP_U_ABORT or MAP_CLOSE indications from the VLR is identical to their handling in the registration procedure, see subclause 19.1.4 above.

The password registration procedure in the HLR is shown in figure 19.7.4/1.

Figure 19.7.4/1 (sheet 1 of 2)

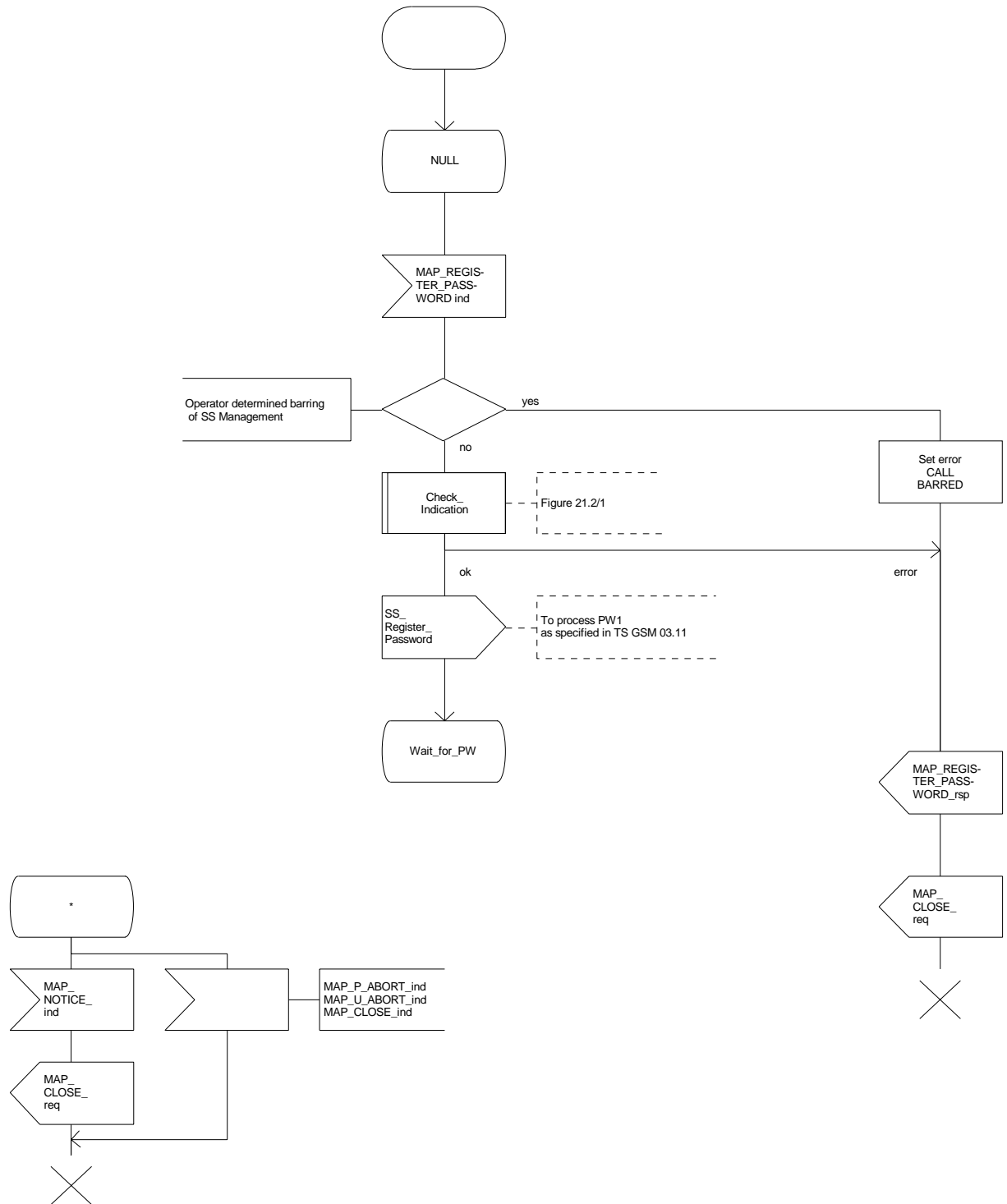
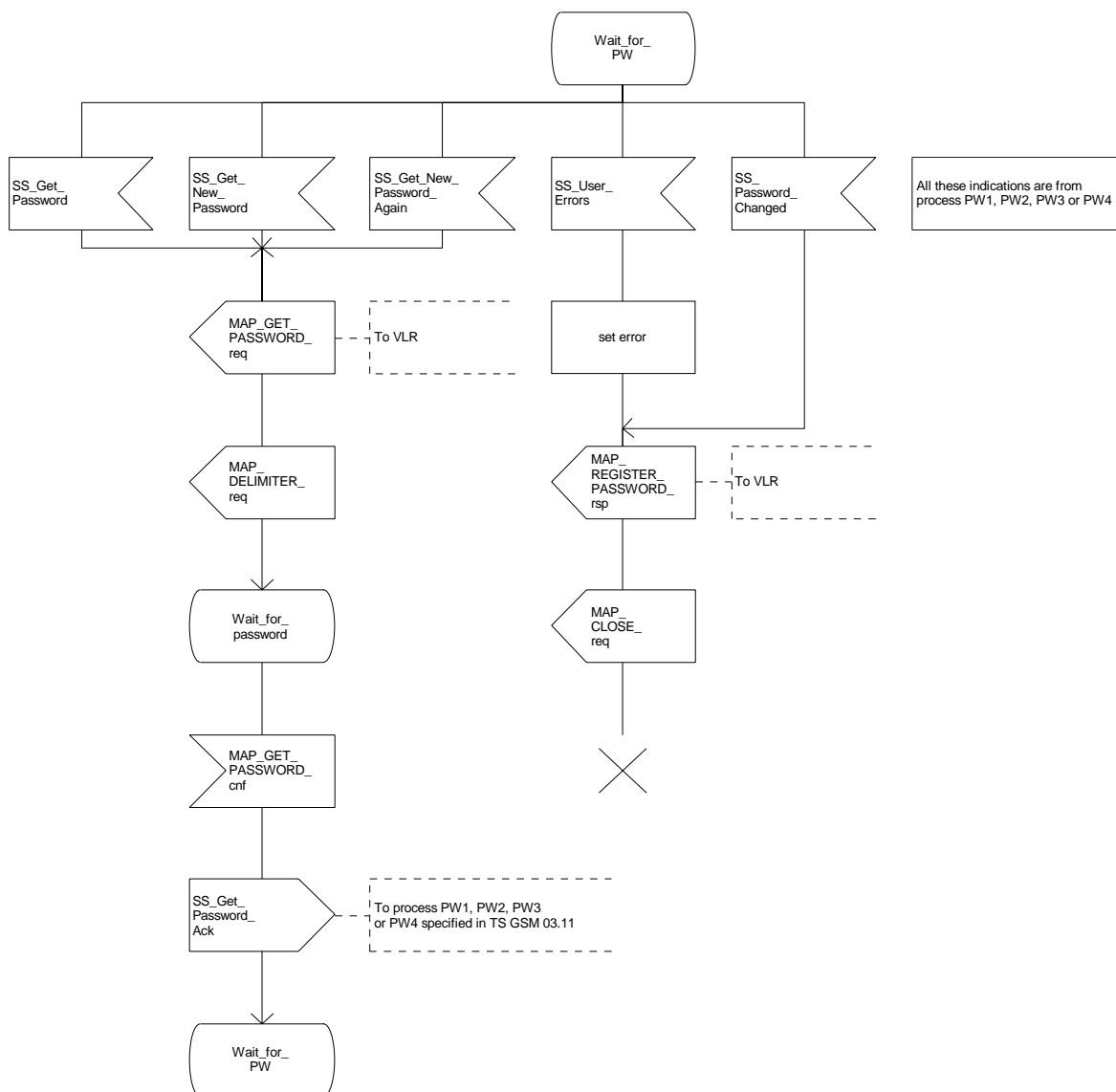


Figure 19.7.4/1 (sheet 2 of 2)



19.8 Mobile Initiated USSD procedure

19.8.1 General

The procedure supports supplementary service signalling procedures which can allow PLMN specific services to be introduced.

The message flow for the procedure can be found in GSM 03.90.

The following services may be used:

MAP_PROCESS_ACCESS_REQUEST	(defined in clauses 6 and 21);
MAP_TRACE_SUBSCRIBER_ACTIVITY	(defined in clauses 7 and 21);
MAP_PROVIDE_IMSI	(defined in clauses 6 and 21);
MAP_FORWARD_NEW_TMSI	(defined in clauses 6 and 21);
MAP_AUTHENTICATE	(defined in clauses 6 and 21);
MAP_SET_CIPHERING_MODE	(defined in clauses 6 and 21);
MAP_CHECK_IMEI	(defined in clauses 6 and 21);
MAP_READY_FOR_SM	(defined in clauses 10 and 21);
MAP_UNSTRUCTURED_SS_REQUEST	(defined in clause 9);
MAP_UNSTRUCTURED_SS_NOTIFY	(defined in clause 9).

The following service is certainly used:

MAP_PROCESS_UNSTRUCTURED_SS_REQUEST (defined in clause 9).

19.8.2 Procedures in the MSC

Before the Process Unstructured SS Request service can be invoked, a call independent CM connection must be created between the MS and the MSC.

Once a CM-connection is established, the MSC may handle the A_PROCESS_UNSTRUCTURED_SS_REQUEST from the MS. This message contains information input by the user, the message may be fed to an application contained locally in the MSC or to the VLR. The rules for determining this are specified in GSM 03.90.

1) Message Destined for VLR

If the message is destined for the VLR then the MSC shall transfer the message to the VLR using the mapping specified in detail in GSM 09.11.

The MSC may subsequently receive one or more MAP_UNSTRUCTURED_SS_REQUEST or MAP_UNSTRUCTURED_SS_NOTIFY indications from the VLR. These shall be sent transparently to the MS. When a confirmation is received from the MS this shall be returned to the VLR.

When the MSC receives a MAP_PROCESS_UNSTRUCTURED_SS_REQUEST confirmation from the VLR then it shall pass this to the MS and initiate release of the CM connection.

2) Message Destined for Local Application

If the message is destined for the local USSD application then the MSC shall transfer the message to the application.

The MSC may subsequently receive one or more requests from the application which correspond to the MAP_UNSTRUCTURED_SS_REQUEST or MAP_UNSTRUCTURED_SS_NOTIFY indications. These shall be sent transparently to the MS. When a confirmation is received from the MS this shall be returned to the application.

When the MSC receives the result of the original operation from the application then it shall pass this to the MS and initiate release of the CM connection.

Error Handling

Both the MS and the VLR or USSD Application may initiate release of the CM-connection at any time. This is handled as shown in the diagrams.

The procedure in the MSC is shown in figure 19.8.2/1.

Figure 19.8.2/1 (sheet 1 of 2)

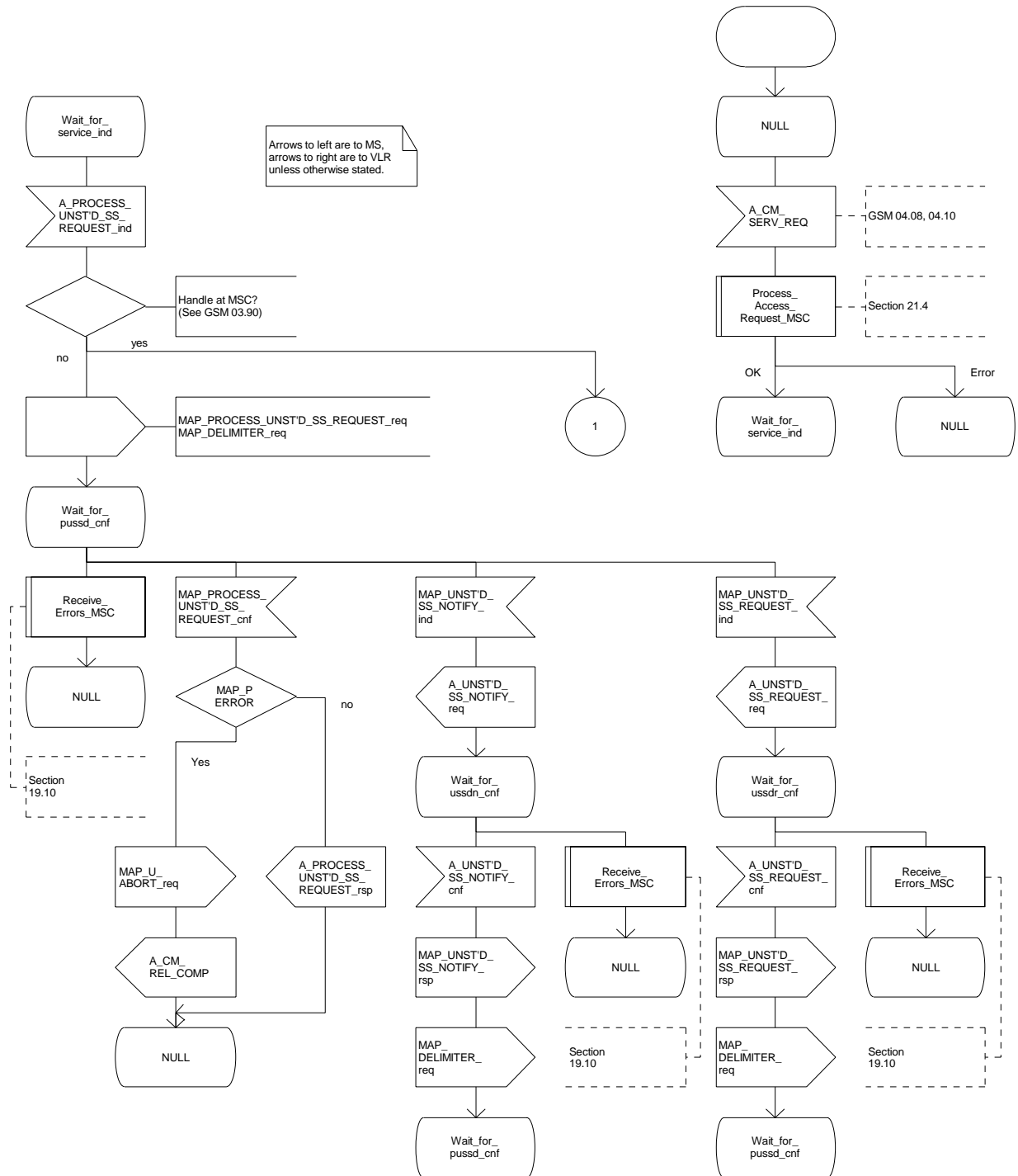
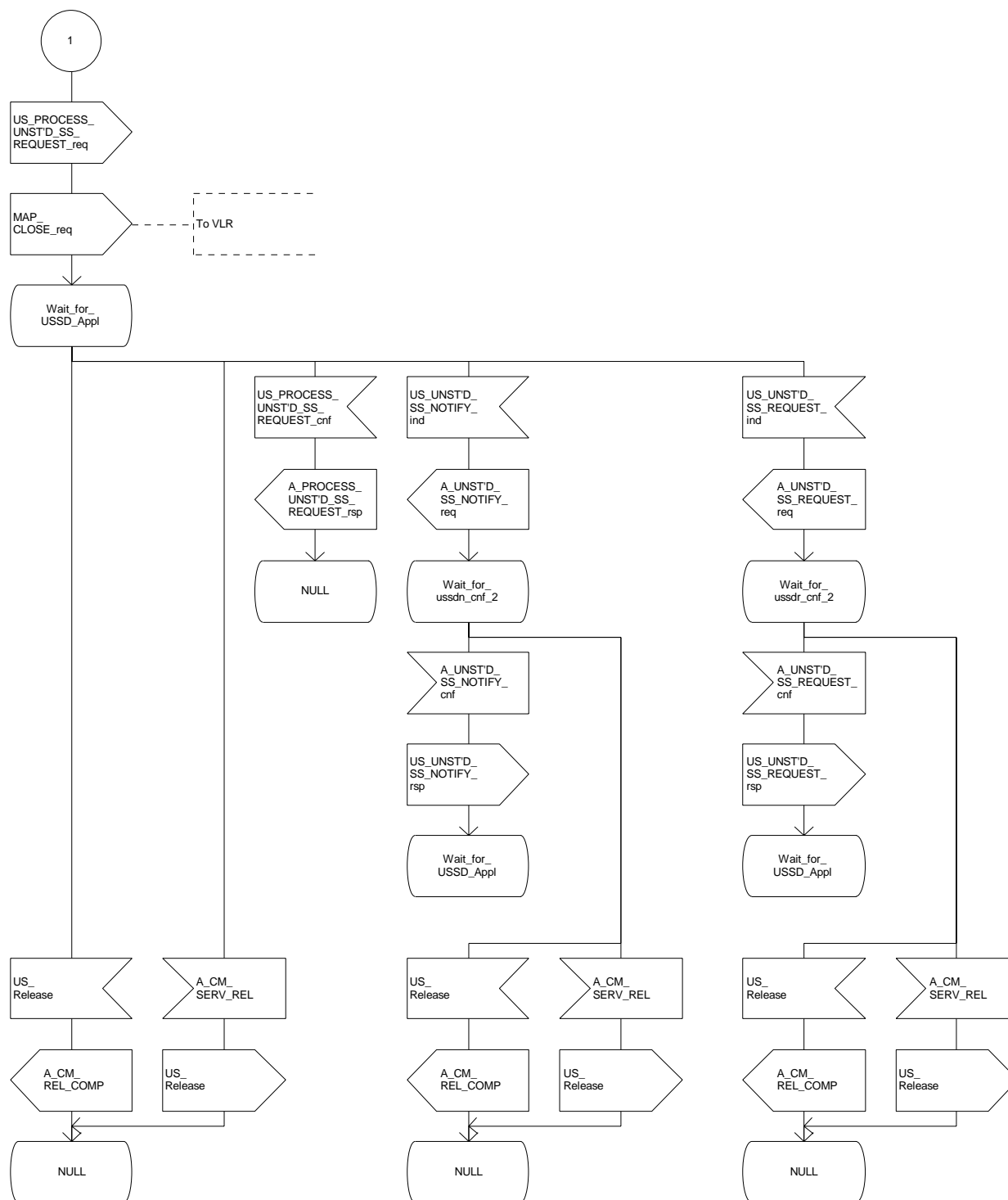


Figure 19.8.2/1 (sheet 2 of 2)

Arrows to left are to MS,
arrows to right are to USSD application
unless otherwise stated.



19.8.3 Procedures in the VLR

The initiation of the process is shown in subclause 19.0.2.

Once a MAP dialogue is established, the VLR may handle the MAP_PROCESS_UNSTRUCTURED_SS_REQUEST from the MSC. This message contains information input by the user, the message may be fed to an application contained locally in the VLR or to the HLR. The rules for determining this are specified in GSM 03.90.

Message Destined for HLR

If the message is destined for the HLR then the VLR shall transfer the message transparently to the HLR.

The VLR may subsequently receive one or more MAP_UNSTRUCTURED_SS_REQUEST or MAP_UNSTRUCTURED_SS_NOTIFY indications from the HLR. These shall be sent transparently to the MSC. When a confirmation is received from the MSC this shall be returned to the HLR.

When the VLR receives a MAP_PROCESS_UNSTRUCTURED_SS_REQUEST confirmation from the HLR then it shall pass this to the MS and close the MAP provider service.

Message Destined for Local Application

If the message is destined for the local USSD application then the VLR shall transfer the message to the application.

The VLR may subsequently receive one or more requests from the application which correspond to the MAP_UNSTRUCTURED_SS_REQUEST or MAP_UNSTRUCTURED_SS_NOTIFY indications. These shall be sent transparently to the MSC. When a confirmation is received from the MSC this shall be returned to the application.

When the VLR receives the result of the original operation from the application then it shall pass this to the MSC and initiate release of the CM connection.

Error Handling

Both the MSC and the HLR or USSD Application may initiate release of the MAP service at any time. This is handled as shown in the diagrams.

The procedure in the VLR is shown in figures 19.8.3/1 and 19.8.3/2.

Figure 19.8.3/1 (sheet 1 of 3)

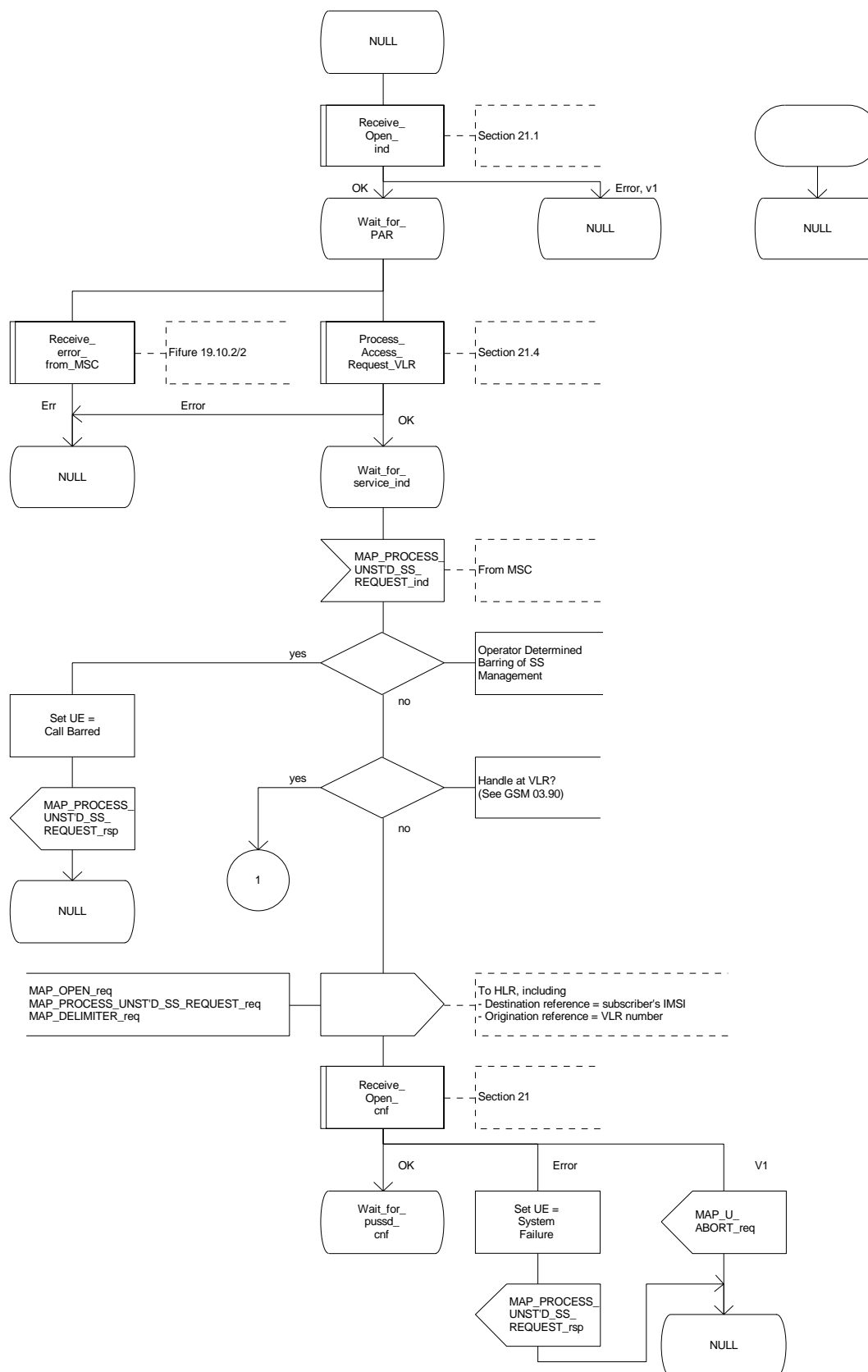


Figure 19.8.3/1 (sheet 2 of 3)

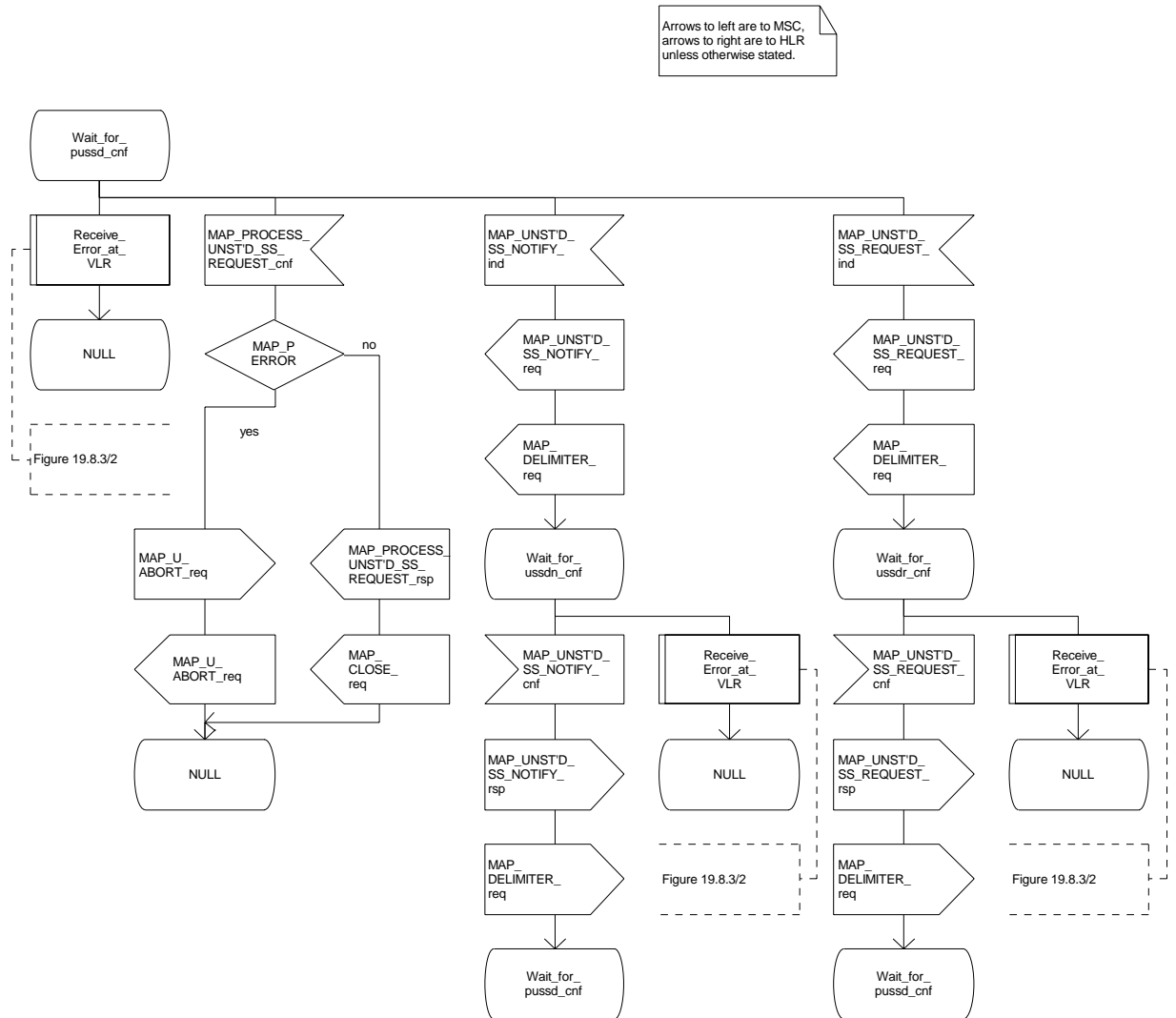


Figure 19.8.3/1 (sheet 3 of 3)

Arrows to left are to MSC,
arrows to right are to USSD application
unless otherwise stated.

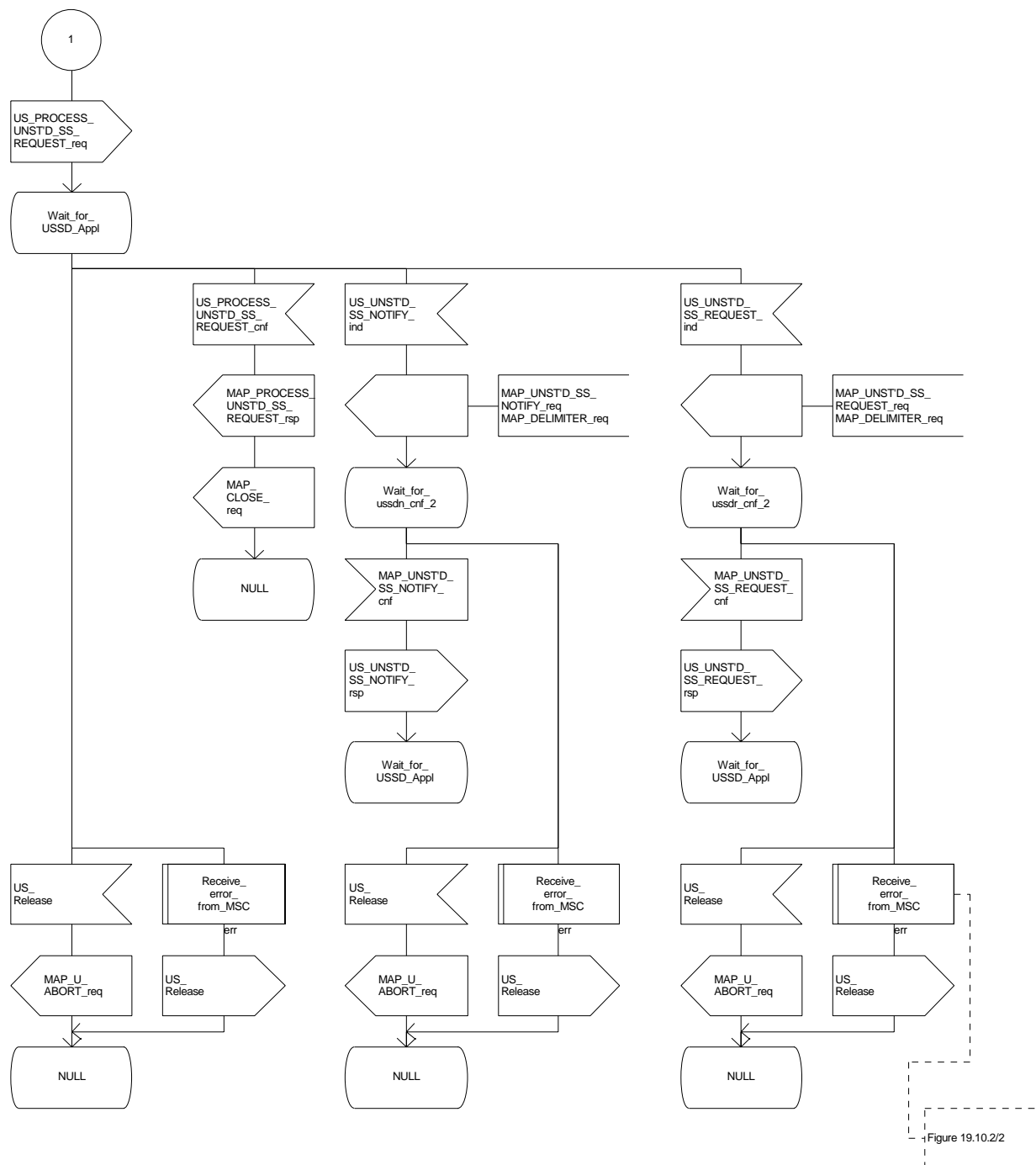
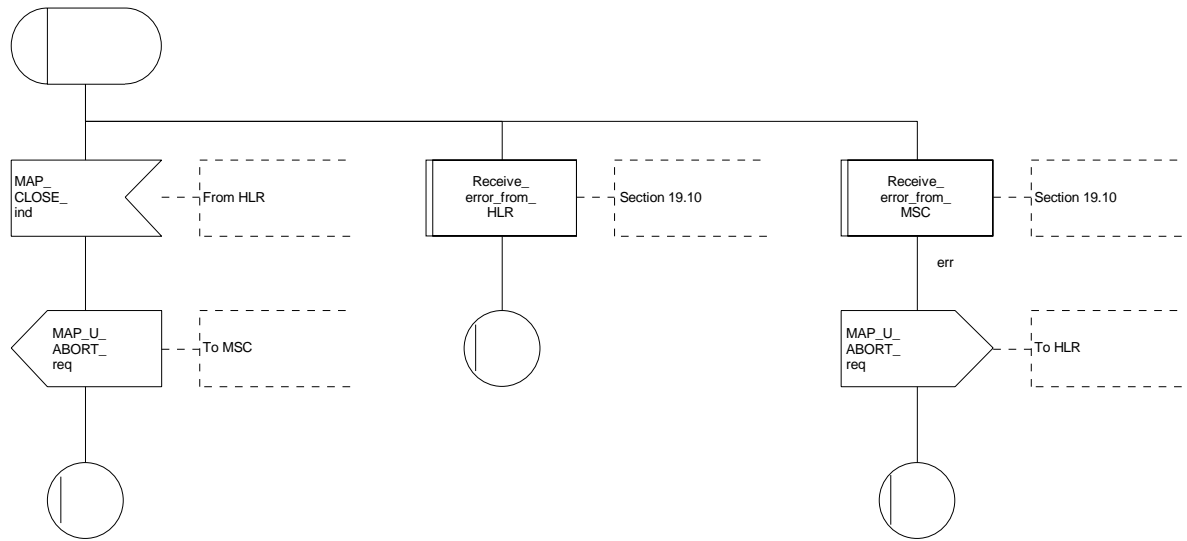


Figure 19.10.2/2

Figure 19.8.3/2



19.8.4 Procedures in the HLR

The initiation of the process is shown in subclause 19.0.3.

Once a MAP dialogue is established, the HLR may handle the MAP_PROCESS_UNSTRUCTURED_SS_REQUEST from the VLR. This message contains information input by the user. If the alphabet used for the message is understood then the message shall be fed to an application contained locally in the HLR. If the alphabet is not understood then the error "UnknownAlphabet" shall be returned.

The HLR may subsequently receive one or more requests from the application which correspond to the MAP_UNSTRUCTURED_SS_REQUEST or MAP_UNSTRUCTURED_SS_NOTIFY indications. These shall be sent transparently to the VLR. When a confirmation is received from the VLR this shall be returned to the application.

When the HLR receives the result of the original operation from the application then it shall pass this to the VLR and initiate release of the CM connection.

Error Handling

Both the VLR and the USSD Application may initiate release of the MAP service at any time. This is handled as shown in the diagrams.

The procedure in the HLR is shown in figure 19.8.4/1.

Figure 19.8.4/1 (sheet 1 of 2)

Arrows to left are to VLR unless otherwise stated.

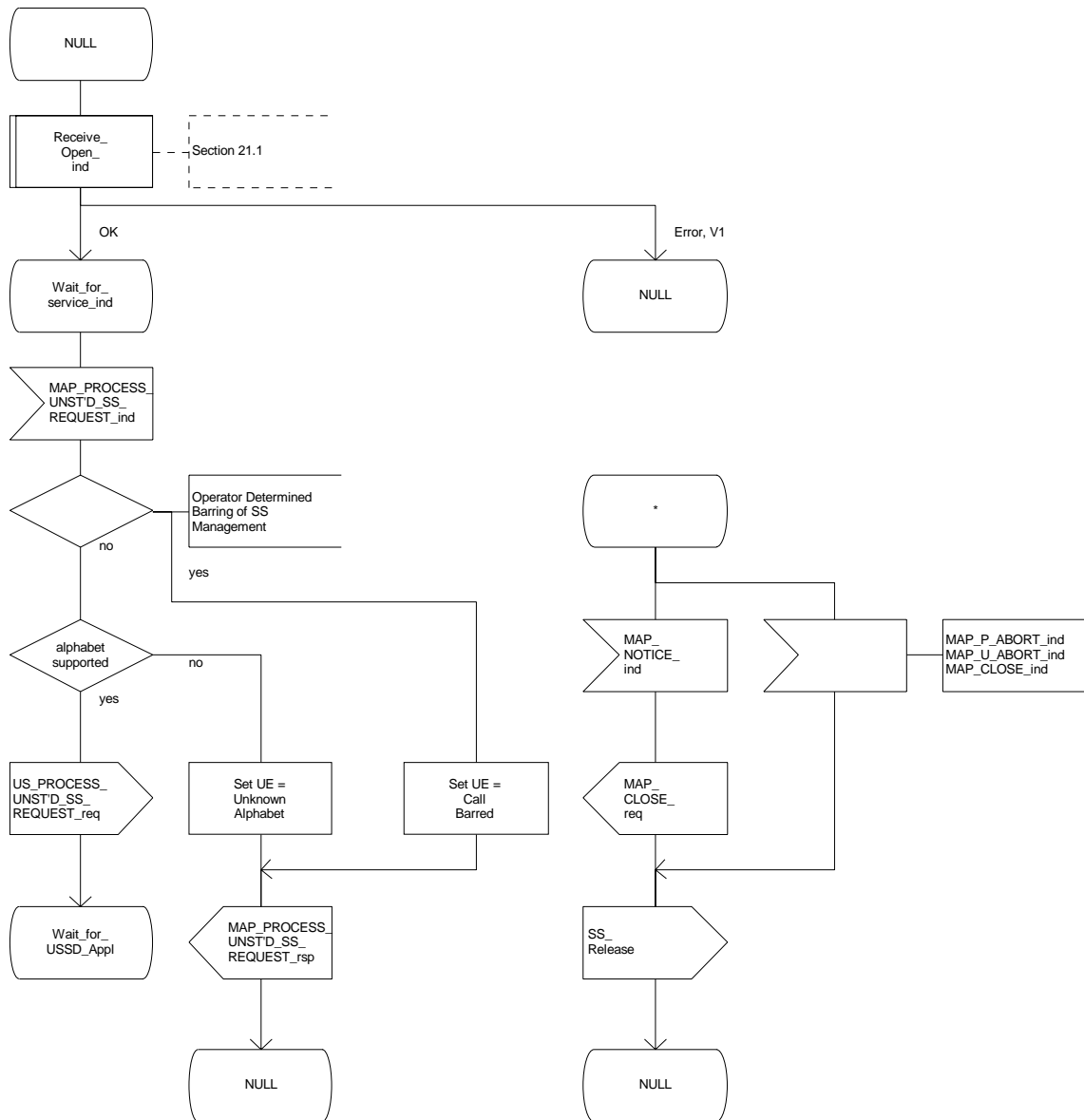
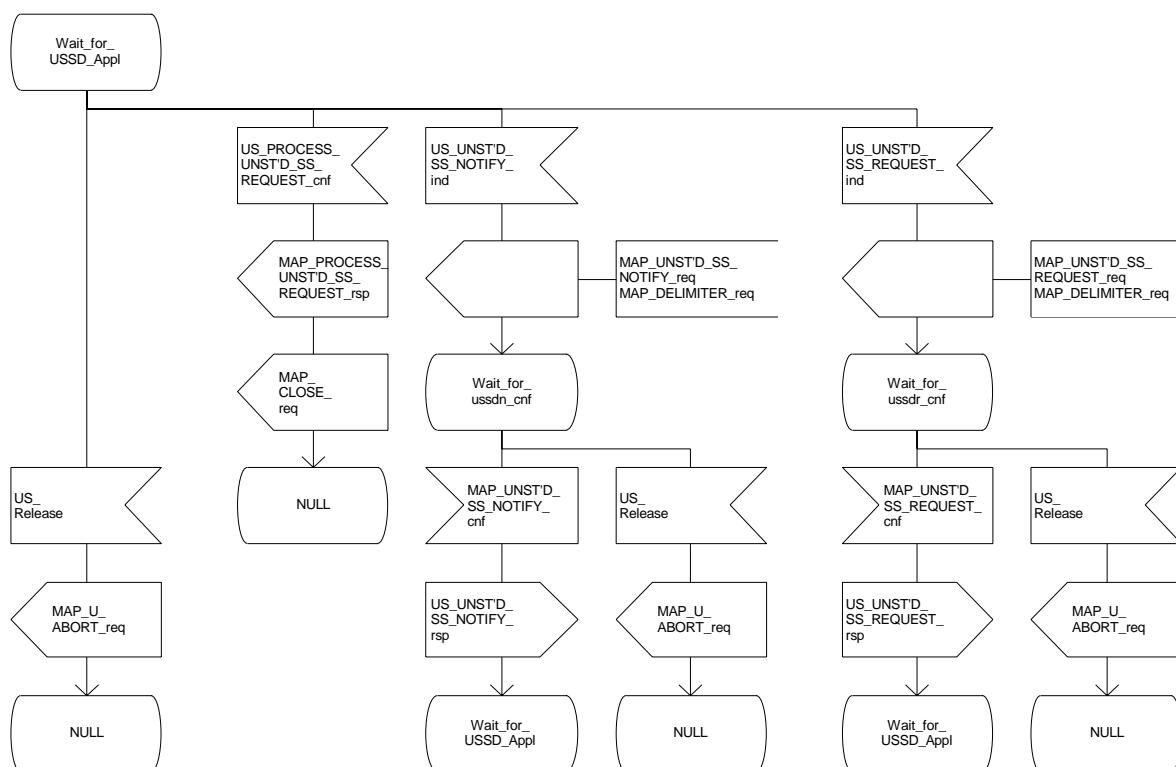


Figure 19.8.4/1 (sheet 2 of 2)

Arrows to left are to VLR,
arrows to right are to USSD application
unless otherwise stated.



19.9 Network initiated USSD procedure

19.9.1 General

The procedure supports supplementary service signalling procedures which can allow PLMN specific services to be introduced.

The message flow for the procedure can be found in GSM 03.90.

The following services may be used:

MAP_PAGE	(defined in clauses 6 and 21);
MAP_SEARCH_FOR_MOBILE_SUBSCRIBER	(defined in clauses 6 and 21);
MAP_PROCESS_ACCESS_REQUEST	(defined in clauses 6 and 21);
MAP_AUTHENTICATE	(defined in clauses 6 and 21);
MAP_SET_CIPHERING_MODE	(defined in clauses 6 and 21);
MAP_FORWARD_NEW_TMSI	(defined in clauses 6 and 21);
MAP_READY_FOR_SM	(defined in clauses 10 and 21).

At least one of the following services will certainly be used, and both may be used:

MAP_UNSTRUCTURED_SS_REQUEST	(defined in clause 9);
MAP_UNSTRUCTURED_SS_NOTIFY	(defined in clause 9).

19.9.2 Procedure in the MSC

The procedure may be invoked either by the VLR or by a USSD application local to the MSC. They may start by using either the MAP_UNSTRUCTURED_SS_REQUEST or MAP_UNSTRUCTURED_SS_NOTIFY service. If the request is initiated by a local USSD application then the MSC will open a dialogue with the HLR.

In both cases the MSC will initiate a CM connection to the MS (using the page or search macros defined in subclause 21.3). Once the connection is successfully established the message received from the VLR or USSD application will be sent to the MS using the mapping specified in GSM 09.11.

Following transfer of the message the MSC will wait for a confirmation from the MS. This will be sent to the VLR or USSD application as appropriate.

Following this, the MSC may receive further uses of the MAP_UNSTRUCTURED_SS_REQUEST or MAP_UNSTRUCTURED_SS_NOTIFY services, or may receive an indication to release the connection to the MS.

In the event of an error, the connection to the MS shall be released, and the MAP process with the VLR shall be aborted as shown in the diagram.

The procedure in the MSC is shown in figure 19.9.2/1.

Figure 19.9.2/1 (sheet 1 of 4)

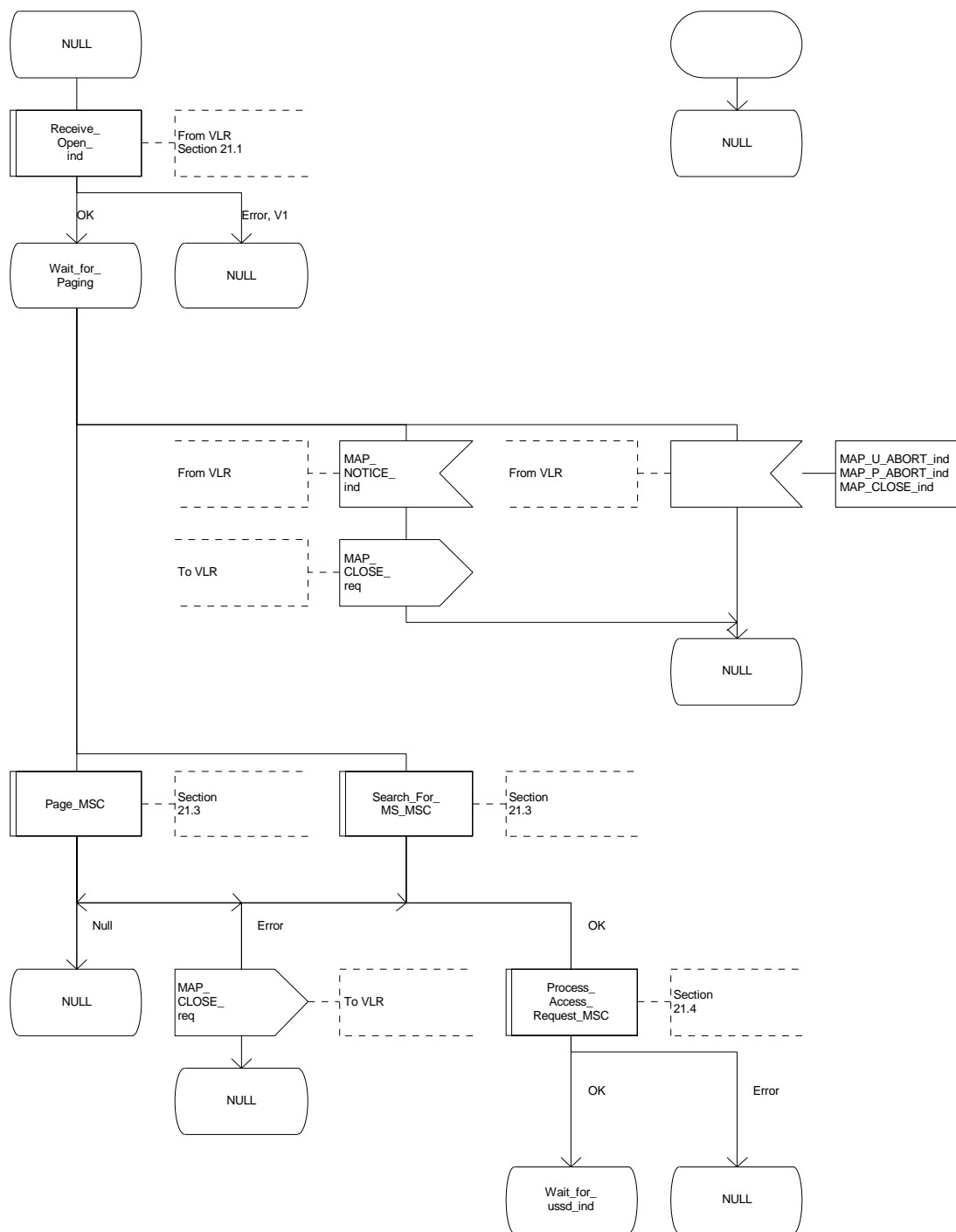


Figure 19.9.2/1 (sheet 2 of 4)

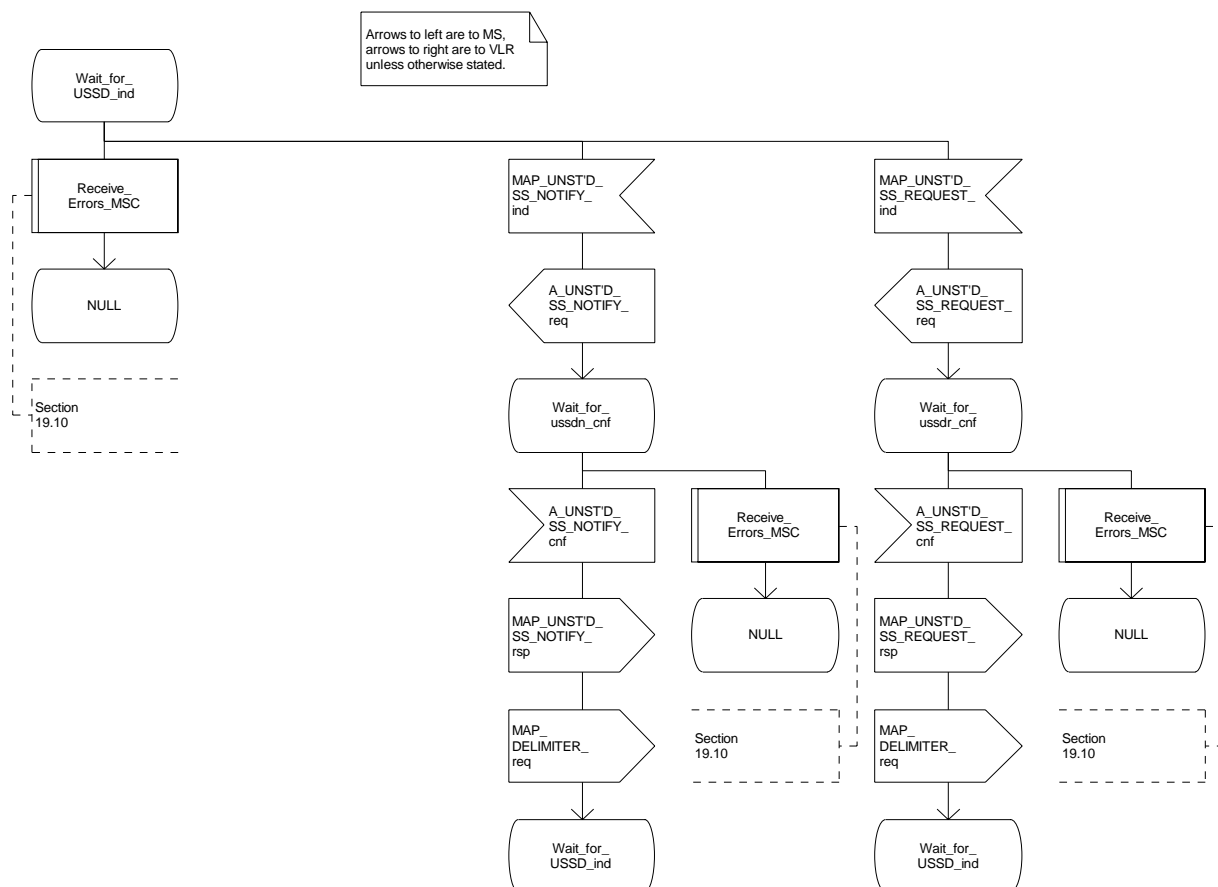


Figure 19.9.2/1 (sheet 3 of 4)

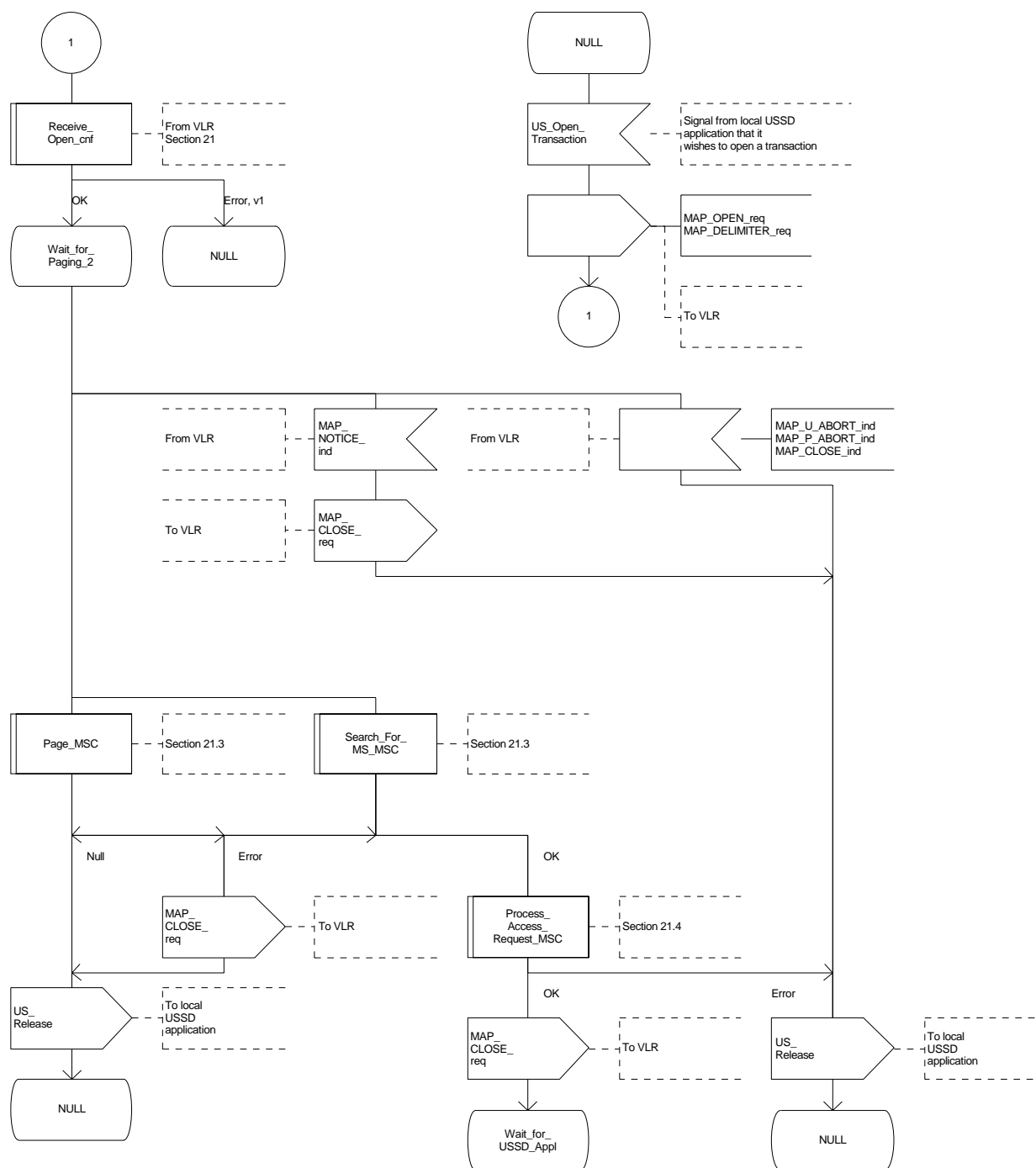
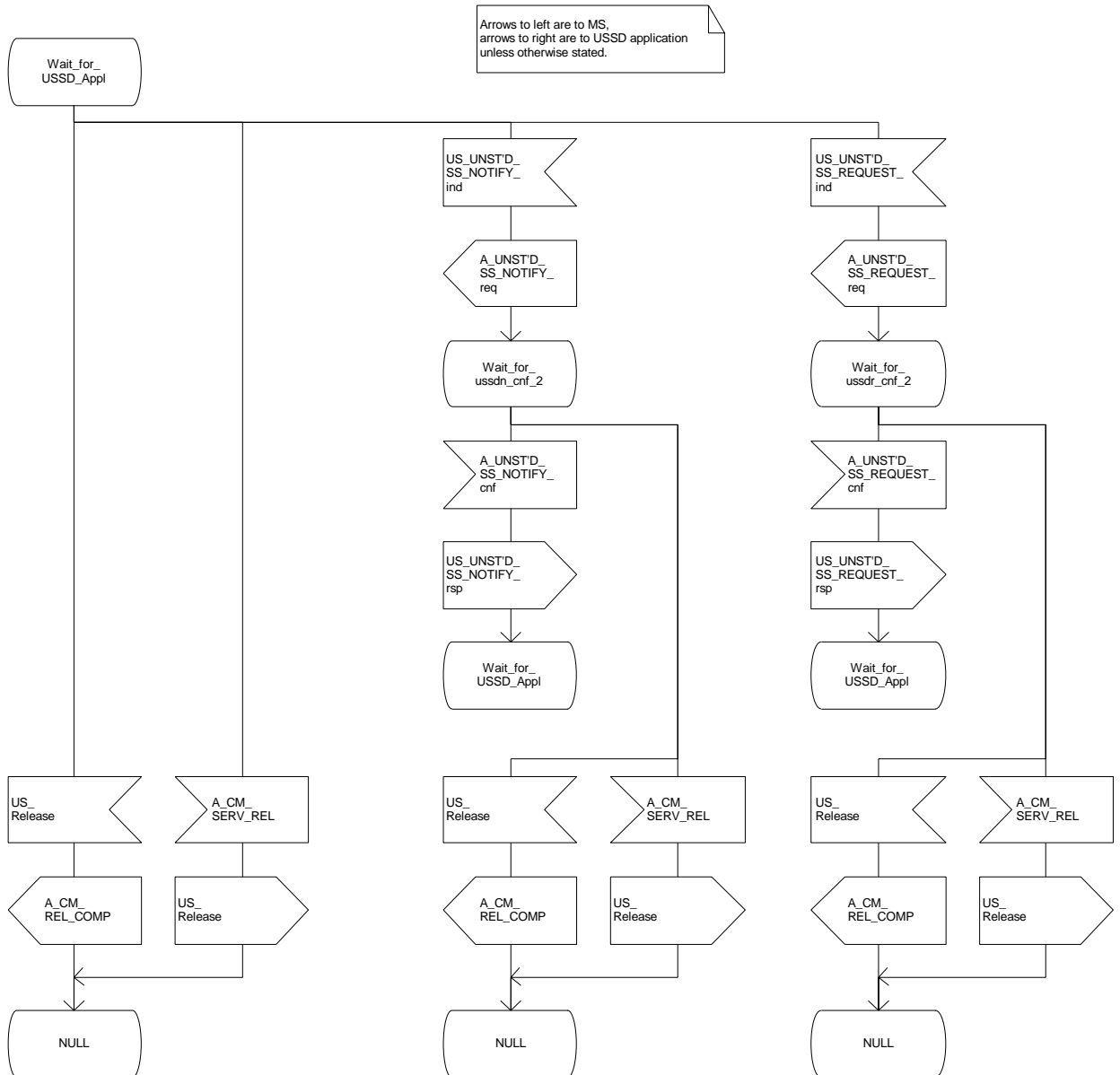


Figure 19.9.2/1 (sheet 4 of 4)



19.9.3 Procedure in the VLR

The procedure may be invoked either by the HLR or by a USSD application local to the VLR. They may start by using either the MAP_UNSTRUCTURED_SS_REQUEST or MAP_UNSTRUCTURED_SS_NOTIFY service.

In both cases the VLR will first initiate a MAP dialogue with the MSC. When the indication for the unstructured SS request or notify is received then the macro Start_USSD_VLR will be used to page the MS and open a CM connection. Once the CM connection is successfully established the indication received from the HLR or USSD application will be sent to the MSC.

Following transfer of the message the VLR will wait for a confirmation from the MSC. This will be sent to the HLR or USSD application as appropriate.

Following this, the VLR may receive further uses of the MAP_UNSTRUCTURED_SS_REQUEST or MAP_UNSTRUCTURED_SS_NOTIFY services, or may receive a MAP_CLOSE_ind.

In the event of an error, the MAP process with the MSC shall be released, and if necessary the MAP process with the HLR shall be aborted as shown in the diagram.

The procedure in the VLR is shown in figure 19.9.3/1.

MSC Initiated USSD

If a USSD application in the MSC wishes to use the network initiated USSD procedure, and a connection to the MS does not exist then it shall open a dialogue to the VLR. This dialogue will automatically lead to the VLR performing page and search using the macro Start_USSD_VLR.

Macro Start_USSD_VLR

This macro is used to initiate a CM connection with the MS for transfer of network initiated unstructured SS data.

It first checks for correct data in the VLR. If a problem is found then "Err" is returned.

A page or search procedure (as appropriate) will then be used to contact the MS. Following successful page or search the macro Process_Access_Request_VLR specified in subclause 21.4 will be used to handle the CM connection establishment.

The macro is shown in figure 19.9.3/2.

Figure 19.9.3/1 (sheet 1 of 4)

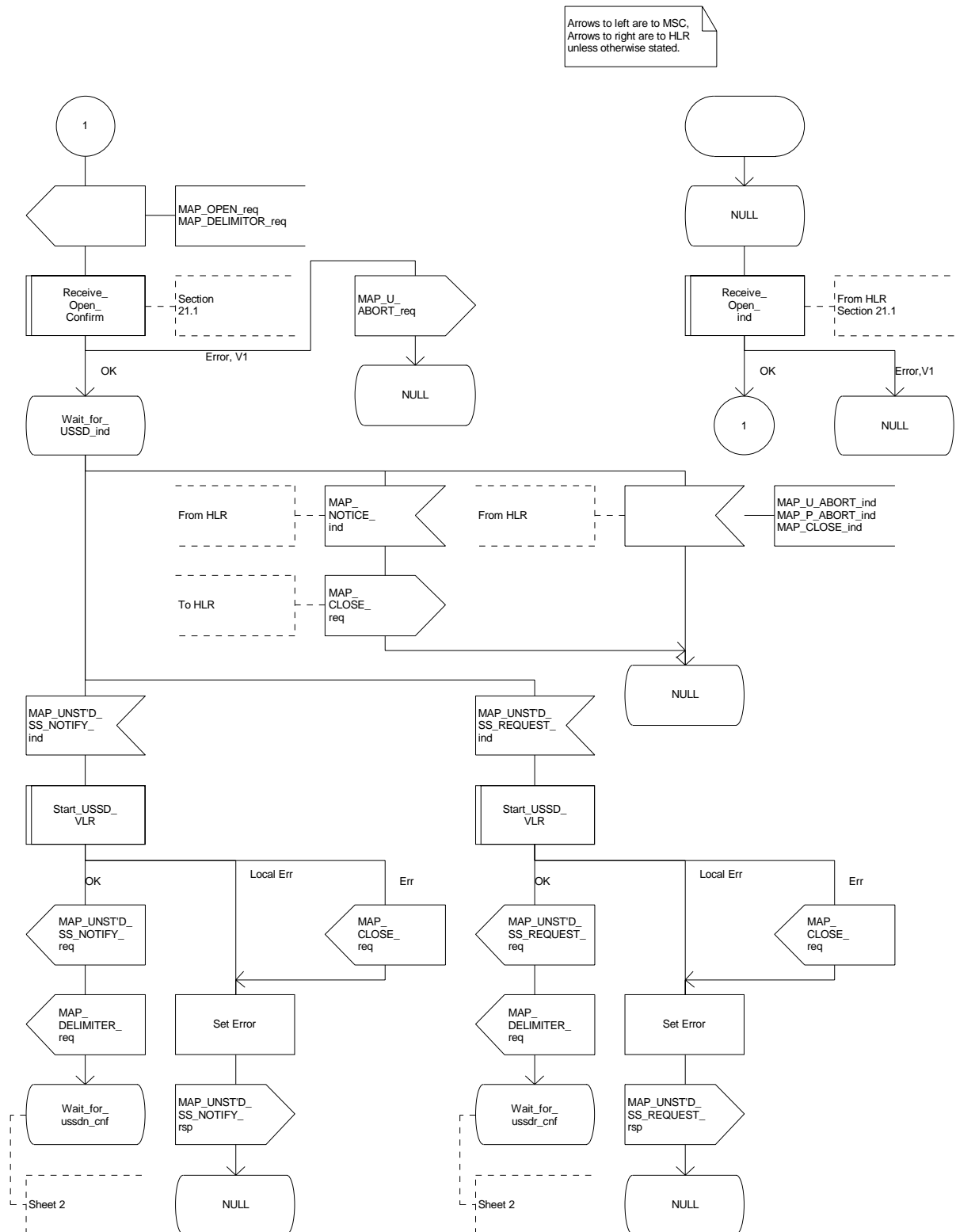


Figure 19.9.3/1 (sheet 2 of 4)

Arrows to left are to MSC,
arrows to right are to HLR
unless otherwise stated.

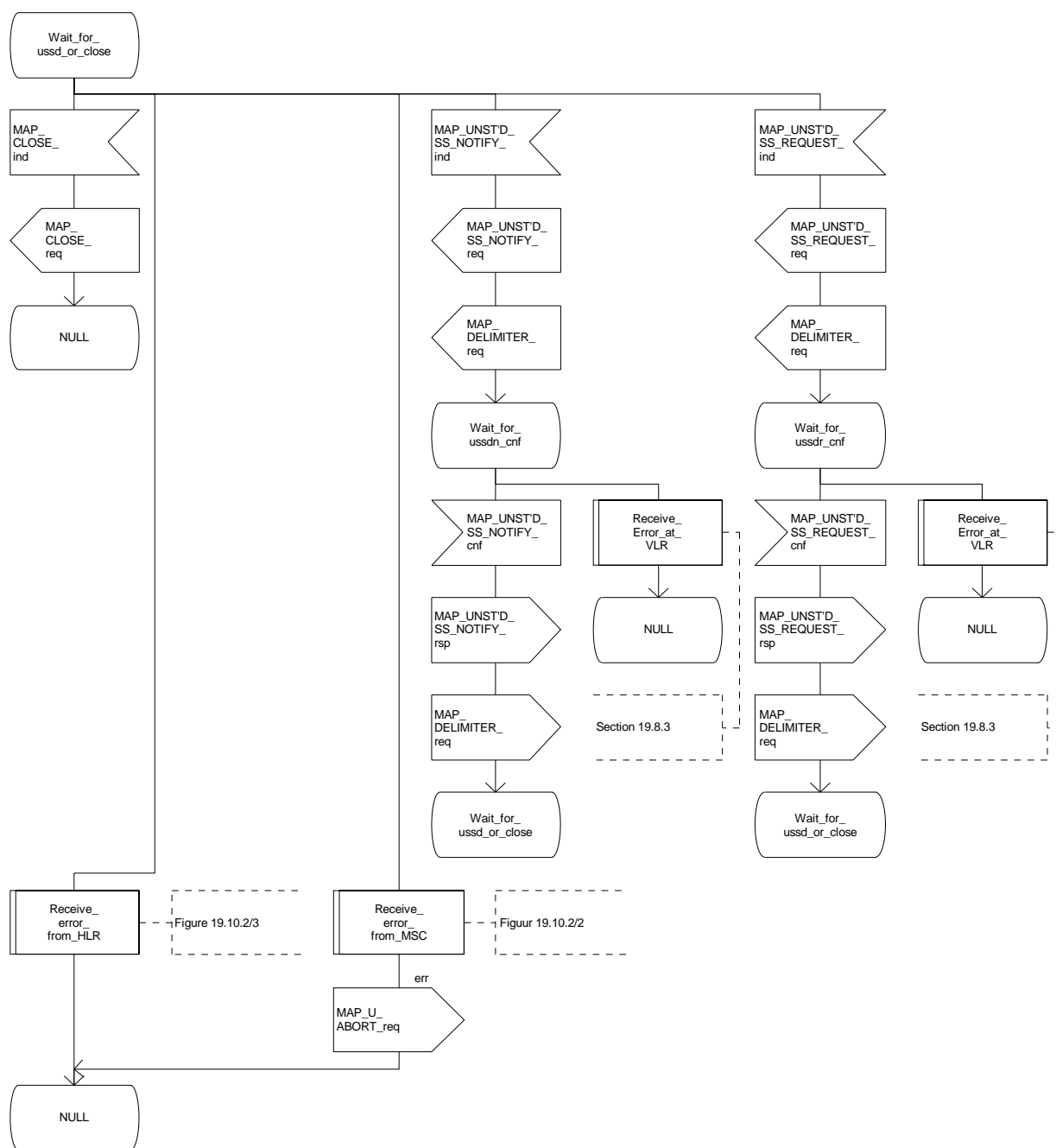


Figure 19.9.3/1 (sheet 3 of 4)

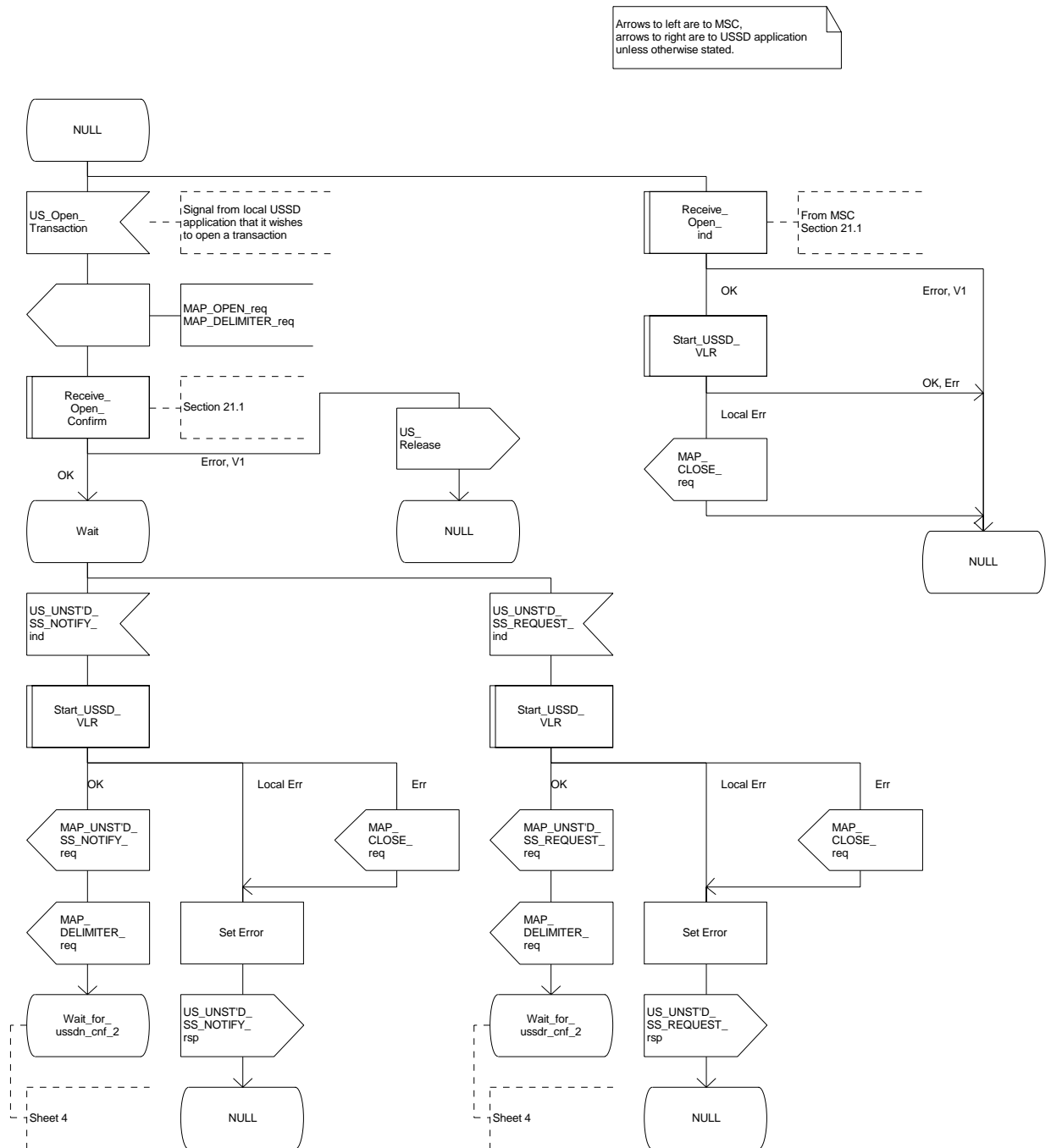


Figure 19.9.3/1 (sheet 4 of 4)

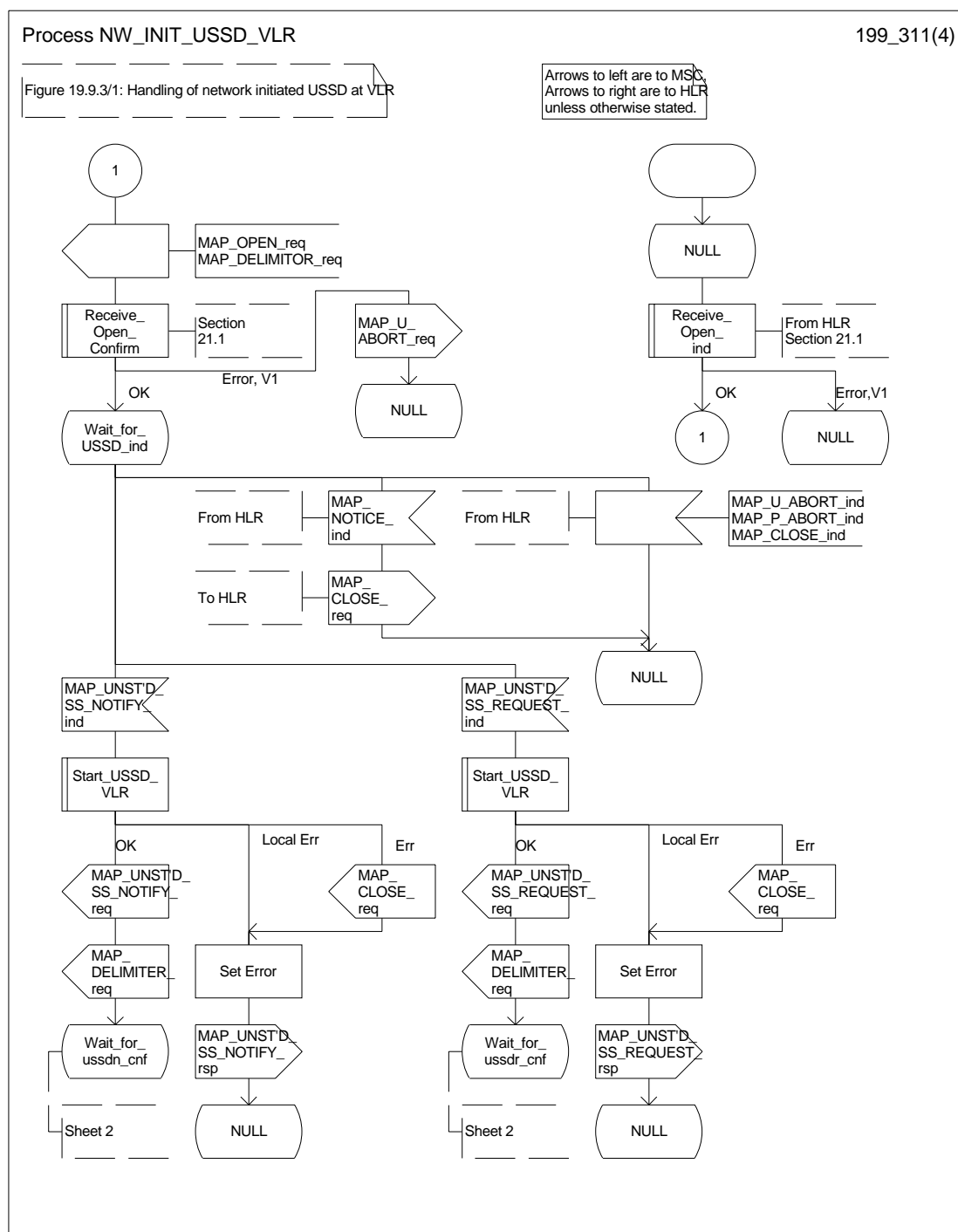


Figure 19.9.3/2 (sheet 1 of 2)

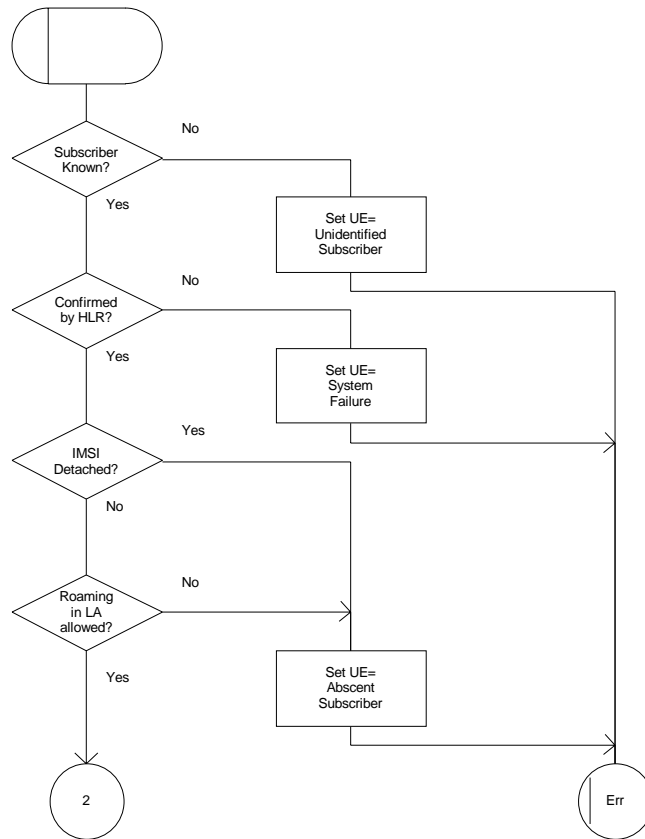
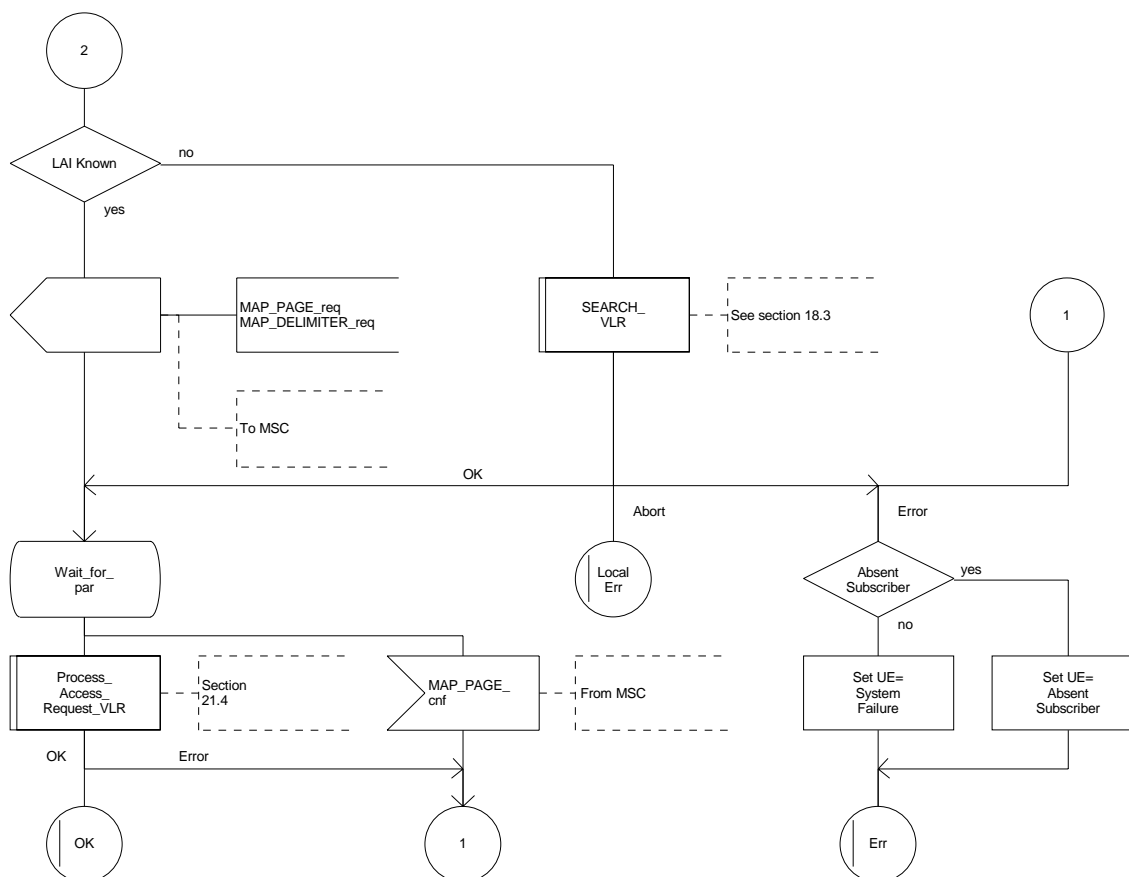


Figure 19.9.3/2 (sheet 2 of 2)



19.9.4 Procedure in the HLR

The procedure may be invoked by the USSD application local to the HLR. It may start by using either the MAP_UNSTRUCTURED_SS_REQUEST or MAP_UNSTRUCTURED_SS_NOTIFY service.

In both cases the HLR will first check whether the MS is reachable (i.e. there is a VLR identity stored in the subscriber record, the MS record is not marked as purged and the MS record is not marked "MSC Area Restricted").

If the MS is reachable, the HLR will initiate a MAP dialogue with the VLR. Once the dialogue is successfully established the message received from the USSD application will be sent to the MSC.

Following transfer of the message the HLR will wait for a confirmation from the MSC. This will be sent to the USSD application.

Following this, the HLR may receive further uses of the MAP_UNSTRUCTURED_SS_REQUEST or MAP_UNSTRUCTURED_SS_NOTIFY services, or may receive a MAP_CLOSE_ind.

In the event of an error, the MAP process with the VLR shall be released as shown in the diagram.

The procedure in the HLR is shown in figure 19.9.4/1.

Figure 19.9.4/1 (sheet 1 of 2)

Arrows to left are to VLR,
arrows to right are to USSD application
unless otherwise stated.

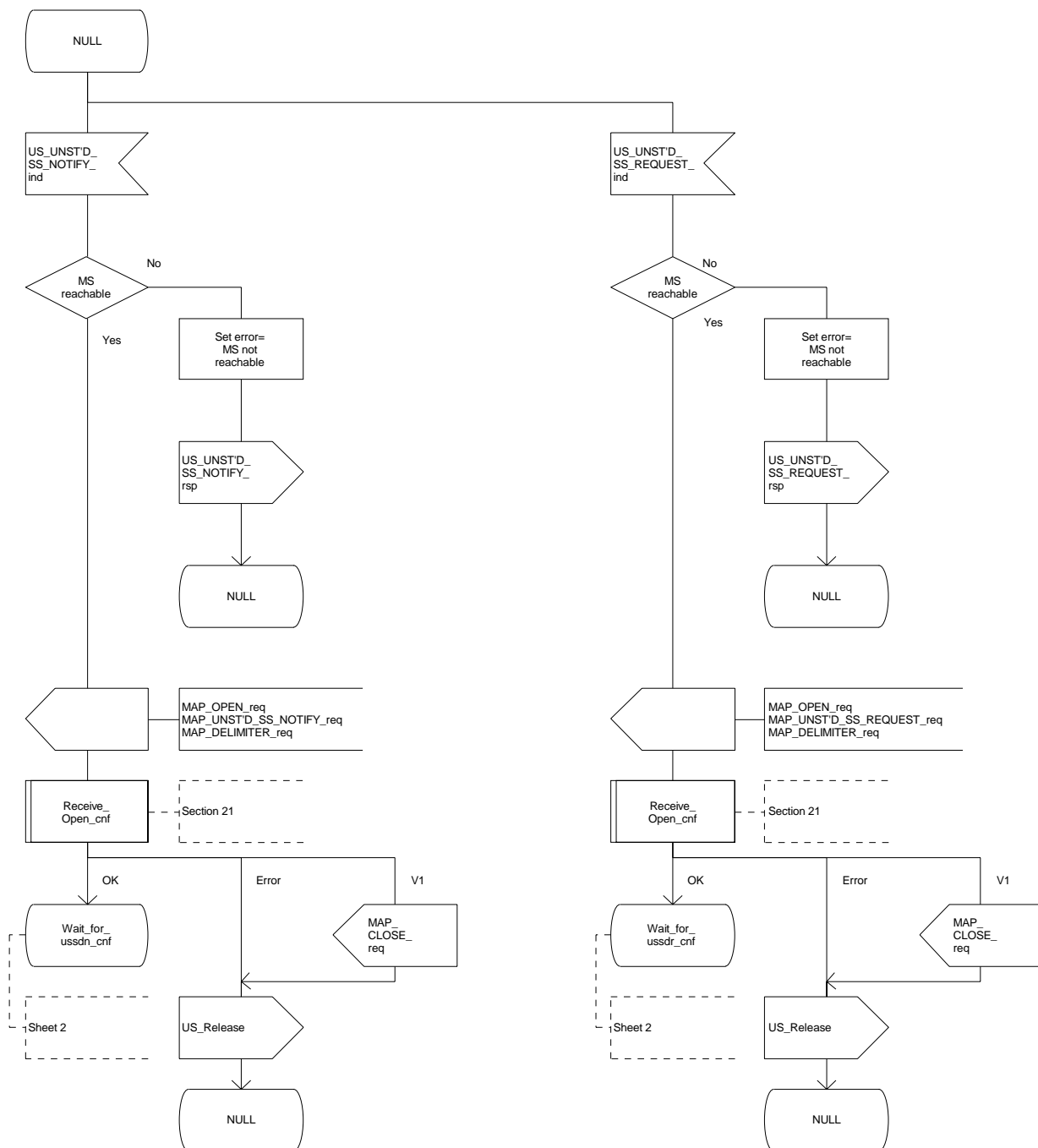
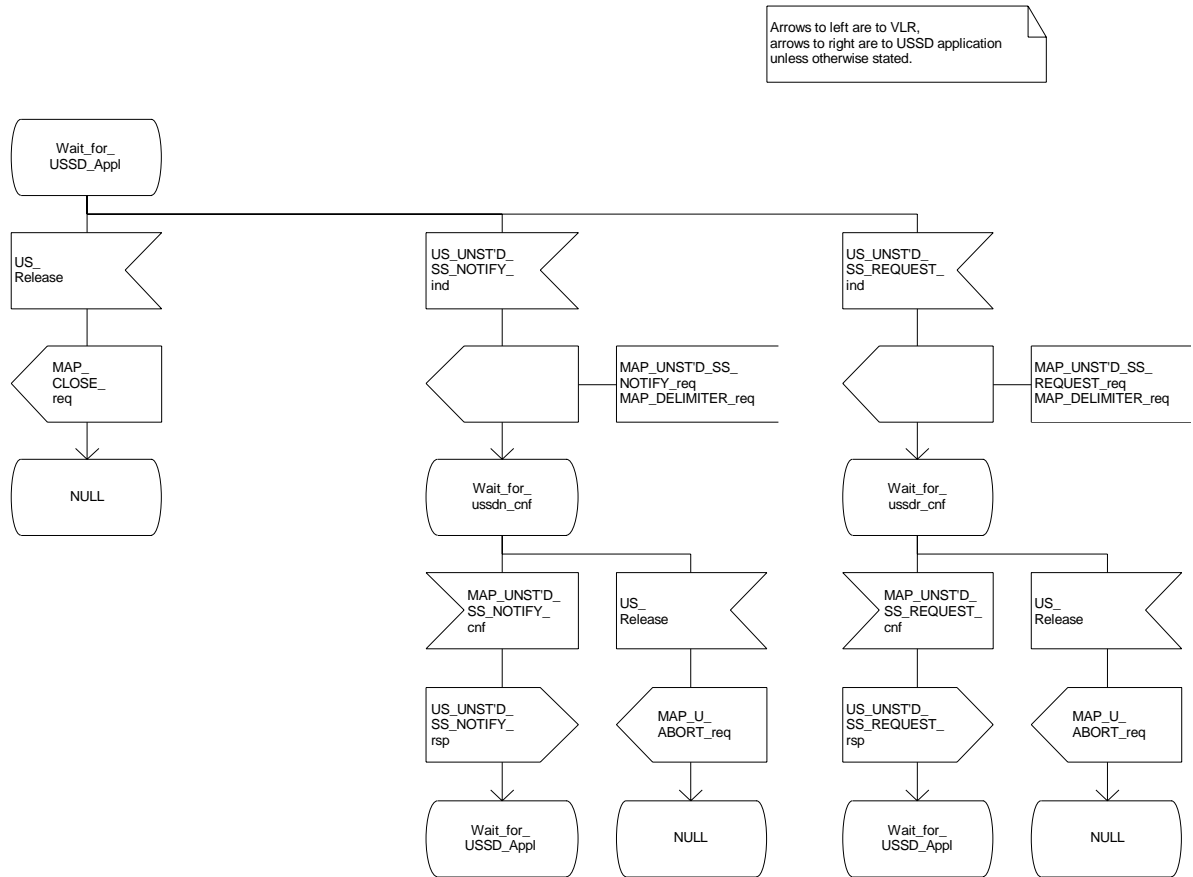


Figure 19.9.4/1 (sheet 2 of 2)



19.10 Common macros for clause 19

The following macros are used for the description of more than one of the supplementary service processes described in clause 19:

19.10.1 SS Password handling macros

Macro Get_Password_MSC

This macro is used by the MSC to relay a request for password from the VLR to the MS, and to relay a response from the MS back to the VLR. The macro is described in figure 19.10.1/1.

Macro Get_Password_VLR

This macro is used by the VLR to relay a request for password from the HLR to the MSC, and to relay a response from the MSC back to the HLR. The macro is described in figure 19.10.1/2.

Figure 19.10.1/1

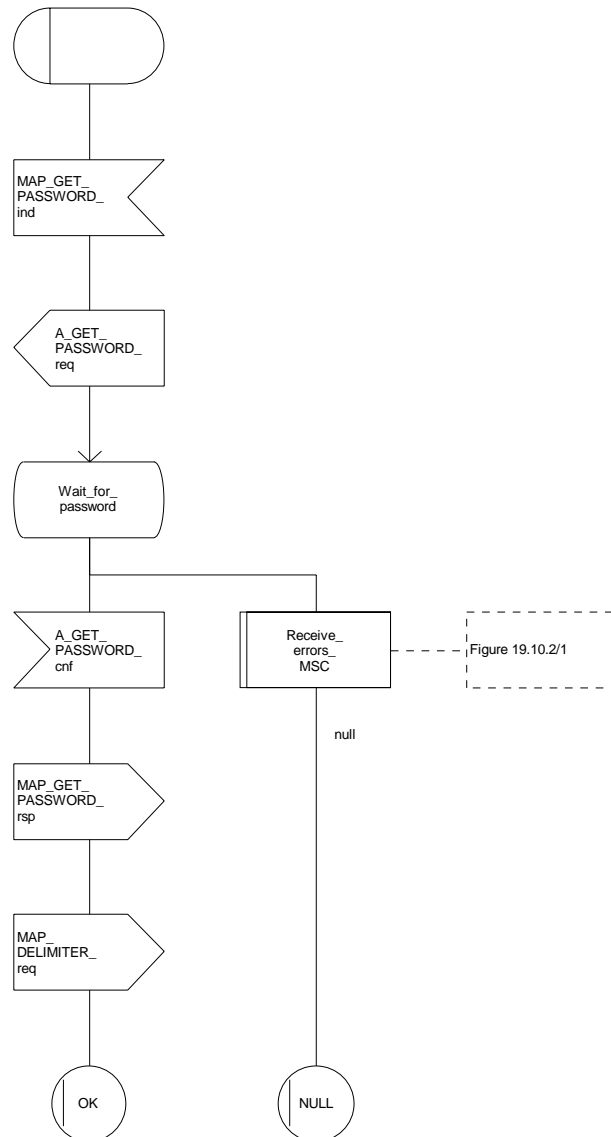
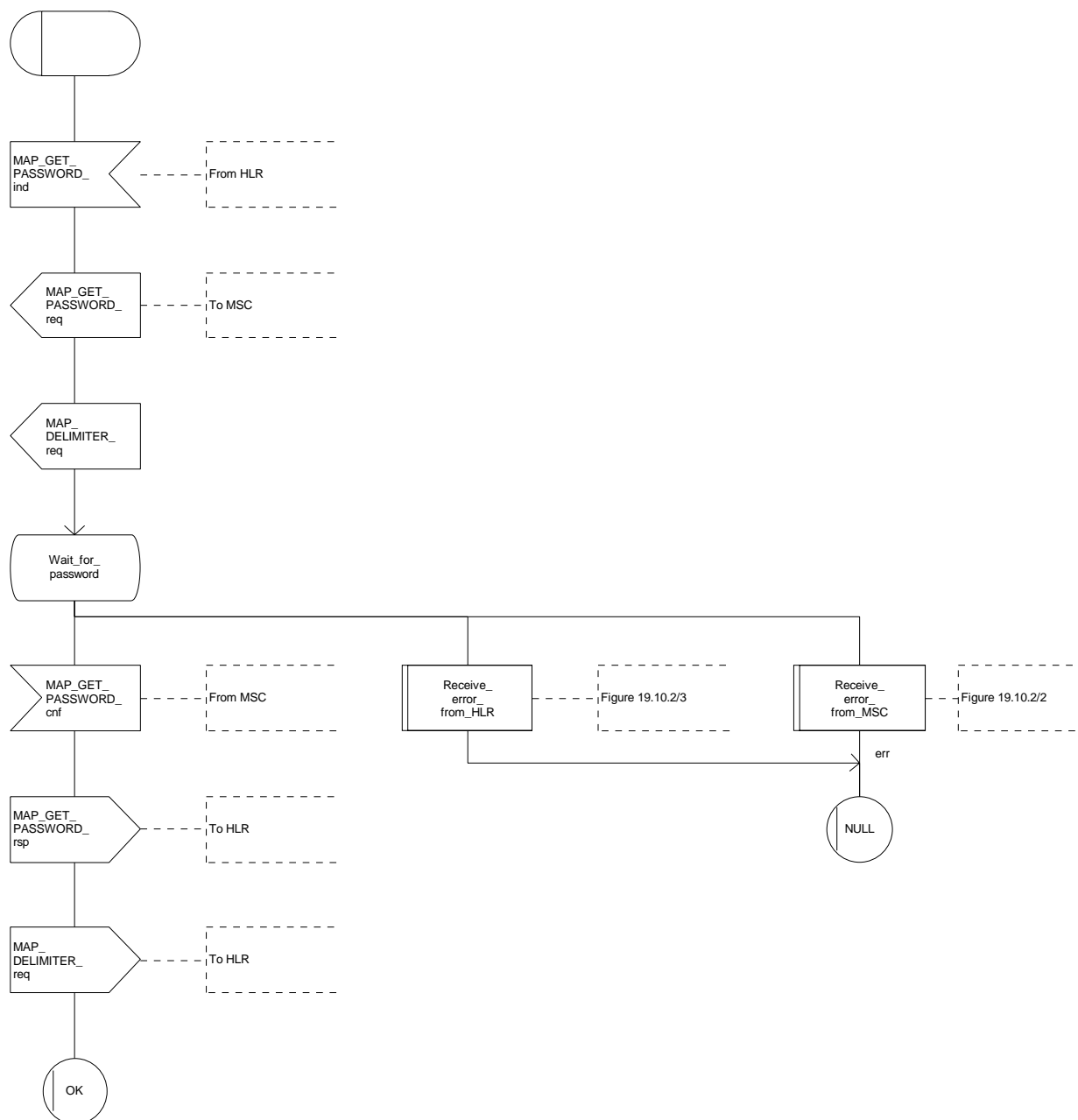


Figure 19.10.1/2



19.10.2 SS Error handling macros

Macro Receive_errors_MSC

This macro is used by the MSC to receive signals which should lead to failure if received in any state of a supplementary service process. If the air interface connection is released by the MS, the communication towards the VLR is aborted, and the MSC should return to a stable "NULL" state. If a MAP_NOTICE indication is received from the VLR, or the VLR aborts or unexpectedly closes the connection, then the air interface connection shall be released. The macro is described in figure 19.10.2/1.

Macro Receive_error_from_MSC

This macro is used by the VLR to receive signals from the MSC which should lead to failure if received in any state of a supplementary service process. If a MAP_NOTICE indication is received from the MSC, that connection is closed before the only outcome of the macro, "err" is reported back to the calling process. The macro is described in figure 19.10.2/2.

Macro Receive_error_from_HLR

This macro is used by the VLR to receive signals from the HLR which should lead to failure if received in any state of a supplementary service process. If a MAP_NOTICE indication is received from the MSC, that connection is closed. The macro is described in figure 19.10.2/3.

Figure 19.10.2/1

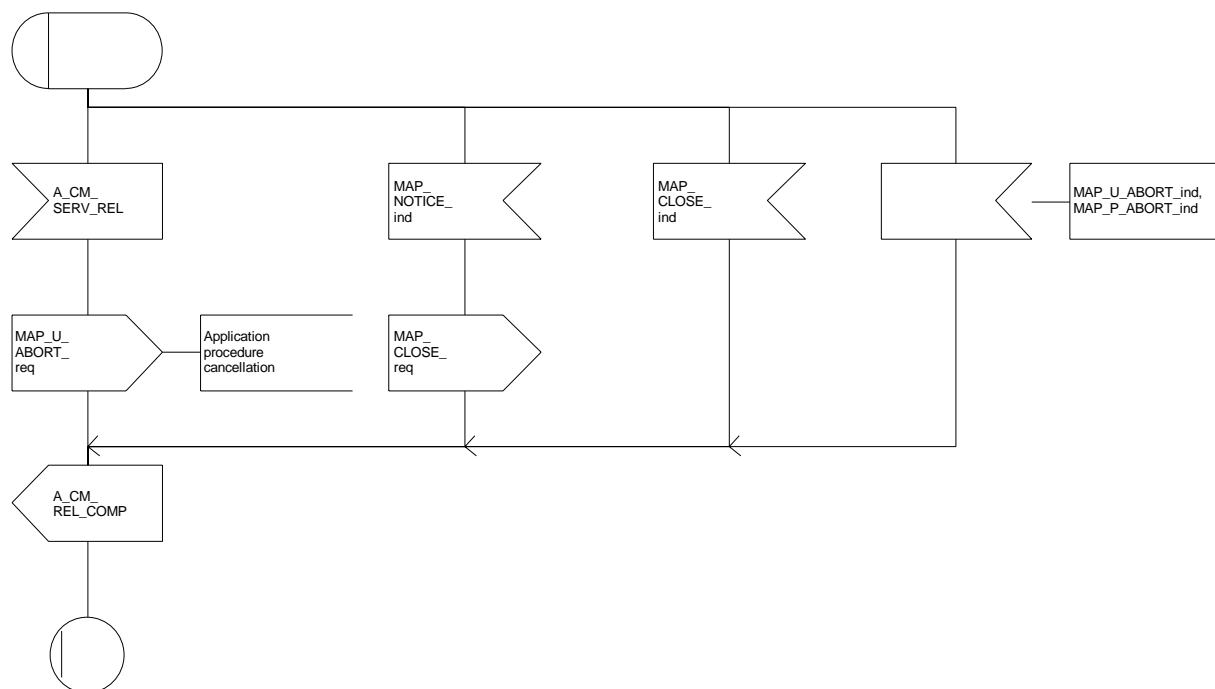


Figure 19.10.2/2

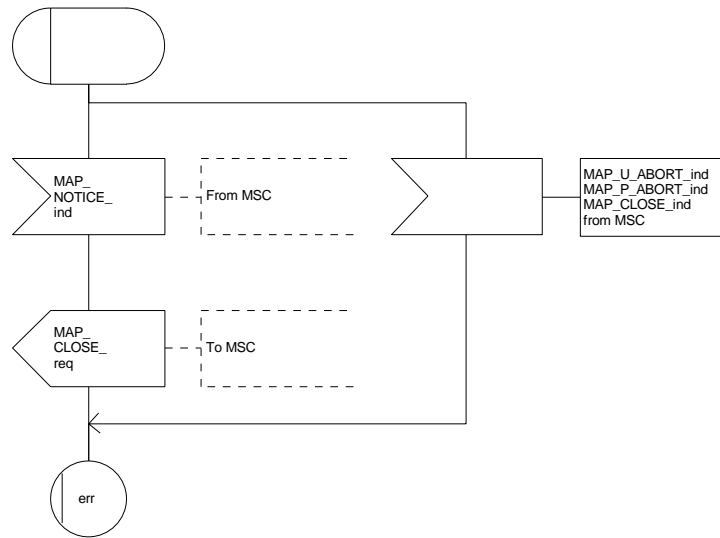
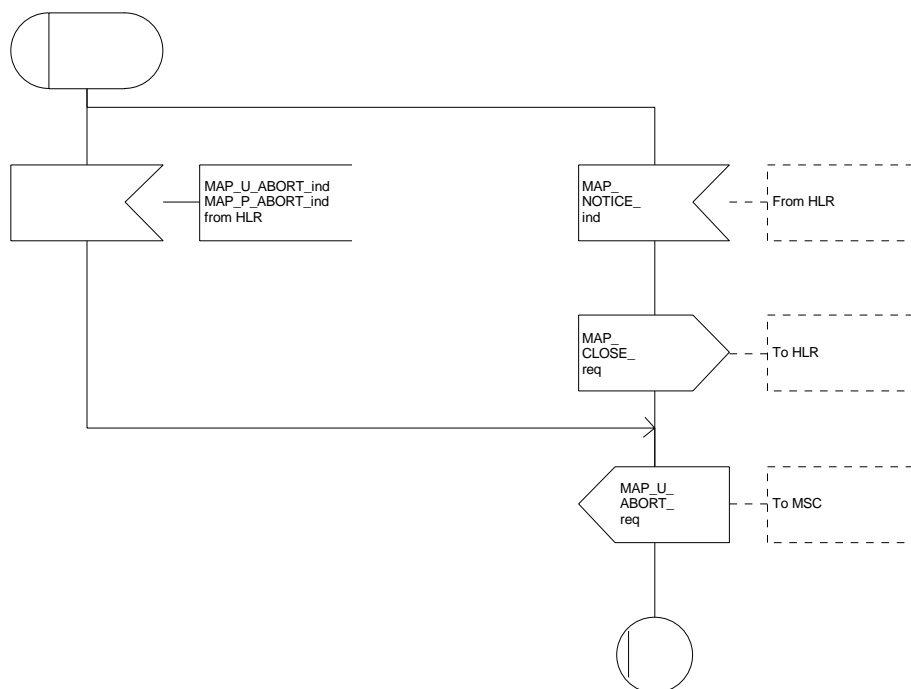


Figure 19.10.2/3



20 Short message service procedures

20.1 General

The short message service procedures are used to control both mobile originated and mobile terminated short message transfer.

Four procedures exist for short message services:

- mobile originated short message service transfer;
- mobile terminated short message service transfer;
- short message alert procedure;
- short message waiting data set procedure.

The following application context refers to a complex MAP user consisting of several processes:

- shortMessageGatewayContext.

This application context needs a co-ordinating process in the HLR. Additionally a Co-ordinator has to be defined for the mobile originated situation in the MSC, because the A_CM_SERV_REQ message does not distinguish between mobile originated short message transfer and the short message alert procedures.

20.1.1 Mobile originated short message service Co-ordinator for the MSC

The A_CM_SERV_REQ message (GSM 04.08) is received from the A-interface containing the CM service type. This parameter indicates mobile originated short message service. The service MAP_PROCESS_ACCESS_REQUEST is started.

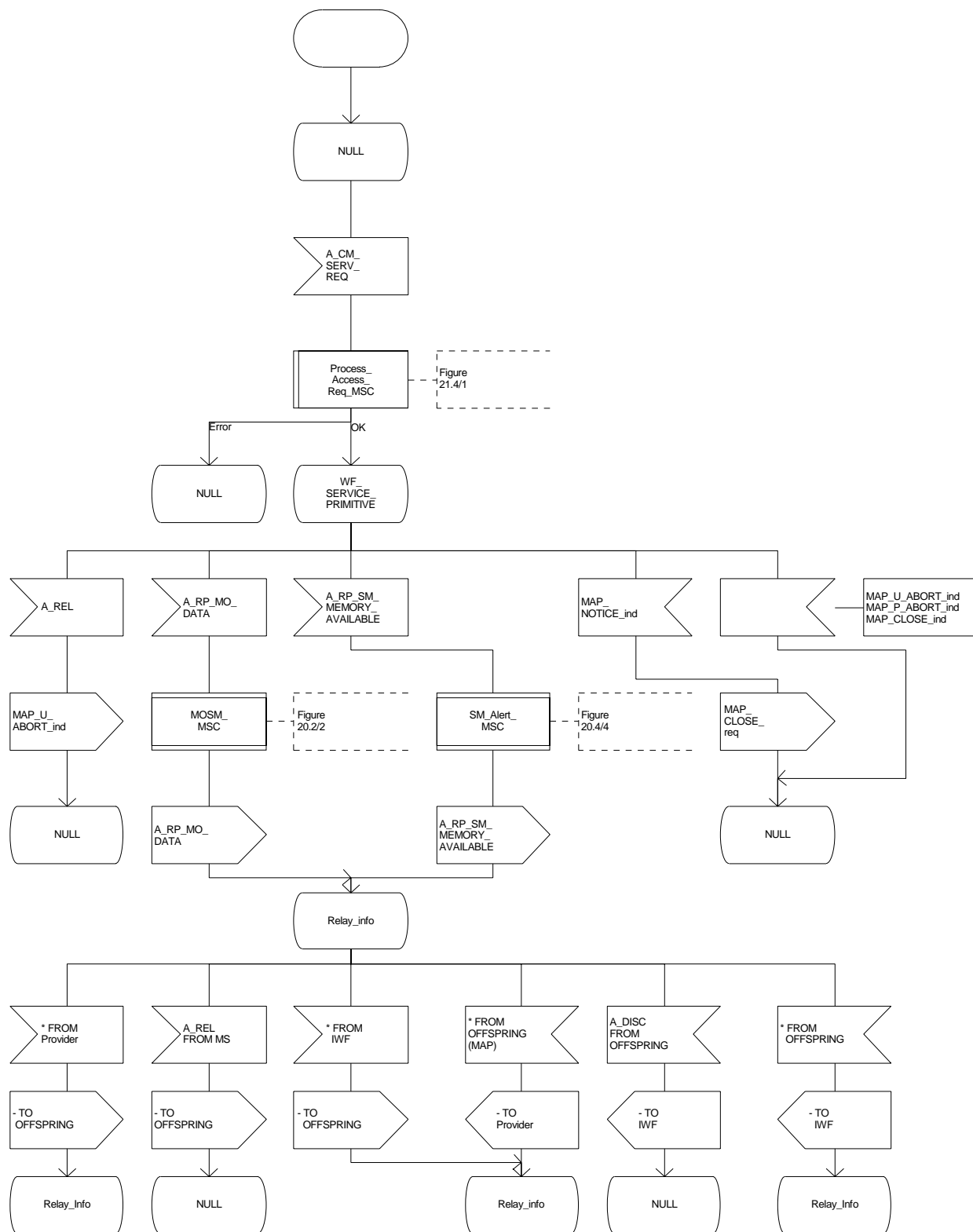
If the MAP_PROCESS_ACCESS_REQUEST service ends successfully, the MS initiates mobile originated short message transfer or alerting indication. Depending on the situation, the appropriate process is initiated as follows:

- if the A_RP_MO_DATA indication is received, the process MOSM_MSC is initiated (see subclause 20.2.1);
- if the A_RP_SM_MEMORY_AVAILABLE indication is received, the process SC_Alert_MSC is initiated (see subclause 20.4.1).

After creation of the user process the Co-ordinator relays the messages between the A-interface and the invoked process until a request or an indication for dialogue termination is received.

The SMS process Co-ordinator is shown in the figure 20.1/1.

Figure 20.1/1



20.1.2 Short message Gateway Co-ordinator for the HLR

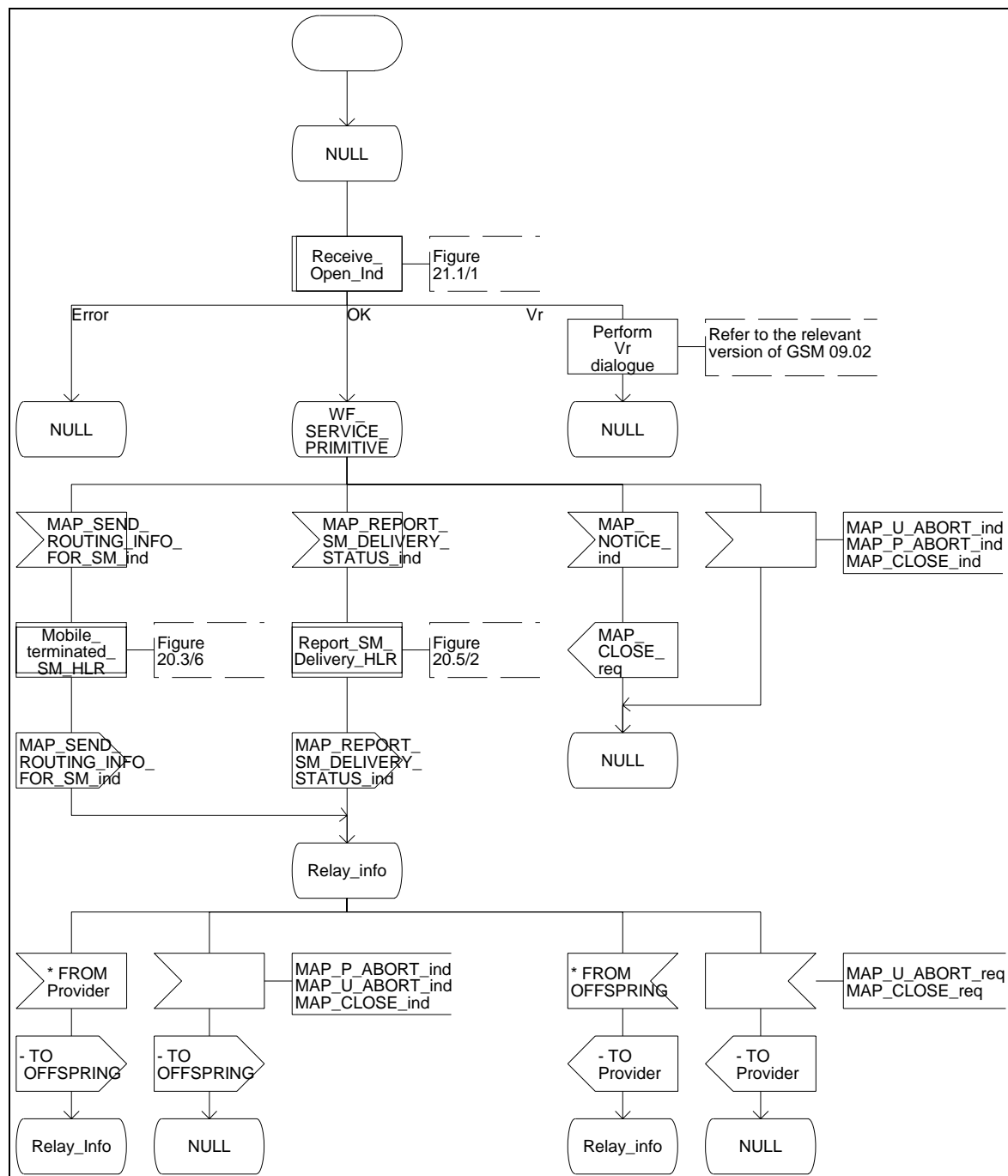
The MAP_OPEN indication opens a dialogue for the short message procedure between the gateway MSC and the HLR when the application context shortMessageGatewayContext is received. If that service is successful, the Co-ordinator can receive the first service primitive from the MAP_PM. Depending on the received primitive, the user process is created as follows:

- if the MAP_SEND_ROUTING_INFO_FOR_SM indication is received, the process Mobile_Terminated_MS_HLR is created;
- if the MAP_REPORT_SM_DELIVERY_STATUS indication is received, the process Report_SM_delivery_stat_HLR is created.

After creation of the user processs the Co-ordinator relays the messages between the MAP_PM and the invoked process until a request or an indication for dialogue termination is received.

The SM Gateway Co-ordinator is shown in the figure 20.1/2.

Figure 20.1/2



20.2 The mobile originated short message transfer procedure

The mobile originated short message service procedure is used to forward short message from a mobile subscriber to a Service Centre. The mobile originated short message service procedure is shown in figure 20.2/1.

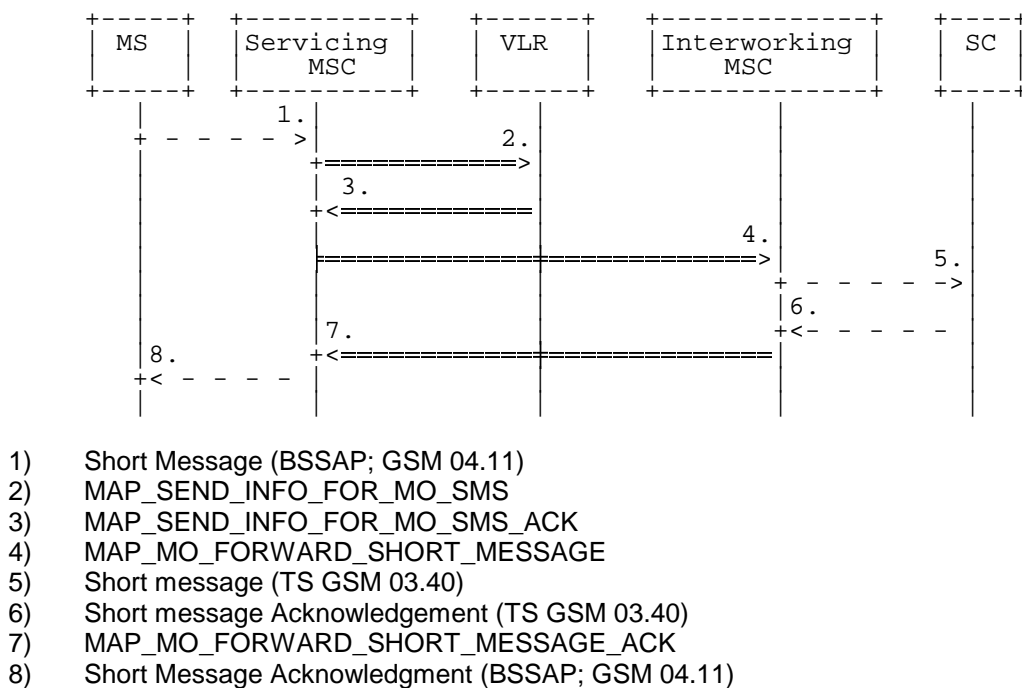


Figure 20.2/1: Mobile originated short message transfer

In addition the following MAP services are used:

MAP_PROCESS_ACCESS_REQUEST	(see subclause 6.3);
MAP_AUTHENTICATE	(see subclause 6.5);
MAP_SET_CIPHERING_MODE	(see subclause 6.6);
MAP_PROVIDE_IMSI	(see subclause 6.9);
MAP_CHECK_IMEI	(see subclause 6.7);
MAP_FORWARD_NEW_TMSI	(see subclause 6.9);
MAP_TRACE_SUBSCRIBER_ACTIVITY	(see subclause 7.1);
MAP_READY_FOR_SM	(see subclause 10.4).

20.2.1 Procedure in the servicing MSC

The activation of the MAP_PROCESS_ACCESS_REQUEST service is described in the subclause 21.4.1.

When receiving the short message from the A-interface, the MSC sends the MAP_SEND_INFO_FOR_MO_SMS request to the VLR. As a response the MSC will receive the MAP_SEND_INFO_FOR_MO_SMS confirmation from VLR indicating that:

- the service ends successfully. If the MSC is not itself the IWMSC, the short message transmission towards the IWMSC is initiated using the MAP_MO_FORWARD_SHORT_MESSAGE request;
- the service ends unsuccessfully. The error cause in the MAP_SEND_INFO_FOR_MO_SMS confirmation indicates the reason for the unsuccessful end. The mapping between MAP error causes and RP_ERROR causes is described in TS GSM 03.40.

If there are data errors in the MAP_SEND_INFO_FOR_MO_SMS confirmation, or there is an operation failure in MAP, the RP_ERROR cause network out of order is forwarded to the mobile station.

If the service MAP_MO_FORWARD_SHORT_MESSAGE is started, the MSC will check whether the grouping of MAP_OPEN request and MAP_MO_FORWARD_SHORT_MESSAGE request needs segmentation. If this is the case then the MAP_OPEN request primitive shall be sent first without any associated MAP service request primitive and the dialogue confirmation must be received before the MAP_MO_FORWARD_SHORT_MESSAGE request is sent. As a response to the procedure, the servicing MSC will receive the MAP_MO_FORWARD_SHORT_MESSAGE confirmation from the IWMSC indicating that:

- the short message has been successfully delivered to the Service Centre. The acknowledgement is sent to the mobile station;
- one of several error cases has occurred. The mapping between MAP error causes and RP_ERROR causes is described in TS GSM 03.40. The appropriate indication is provided to the mobile station.

If the procedure failed, a provider error or an abort indication is received. The RP_ERROR cause network out of order is provided to the mobile station.

If the MSC itself is the interworking MSC, the short message is forwarded to the Service Centre. In that case the service MAP_MO_FORWARD_SHORT_MESSAGE is not initiated. The acknowledge message from the Service Centre is forwarded to the mobile station (TS GSM 3.40, TS GSM 4.11).

The mobile originated short message service procedure is shown in figure 20.2/2.

Figure 20.2/2 (sheet 1 of 3)

Process MOSM_MSC

202_21(3)

Figure 20.2/2: The mobile originated short message service process in the MSC.

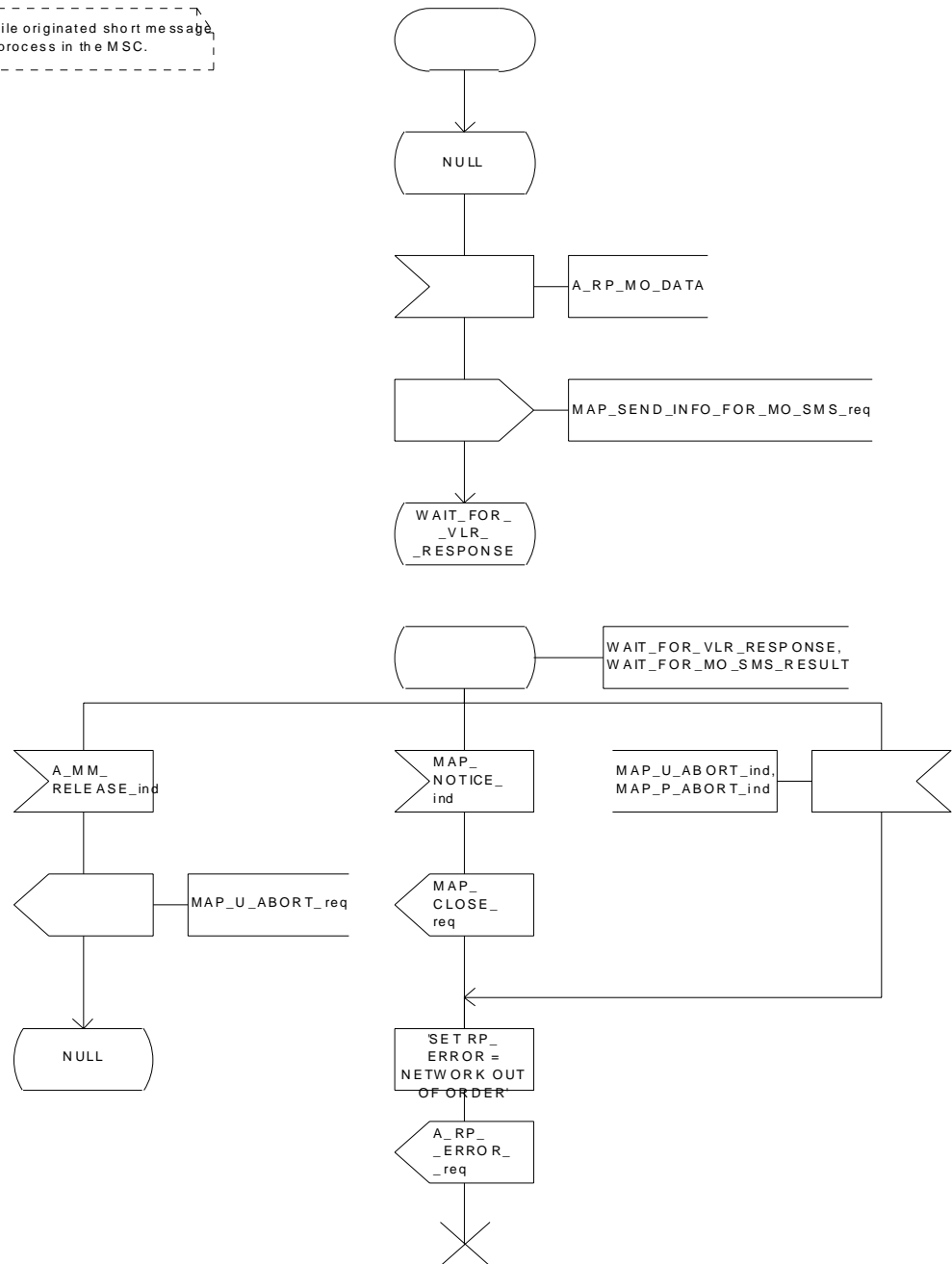


Figure 20.2/2 (sheet 2 of 3)

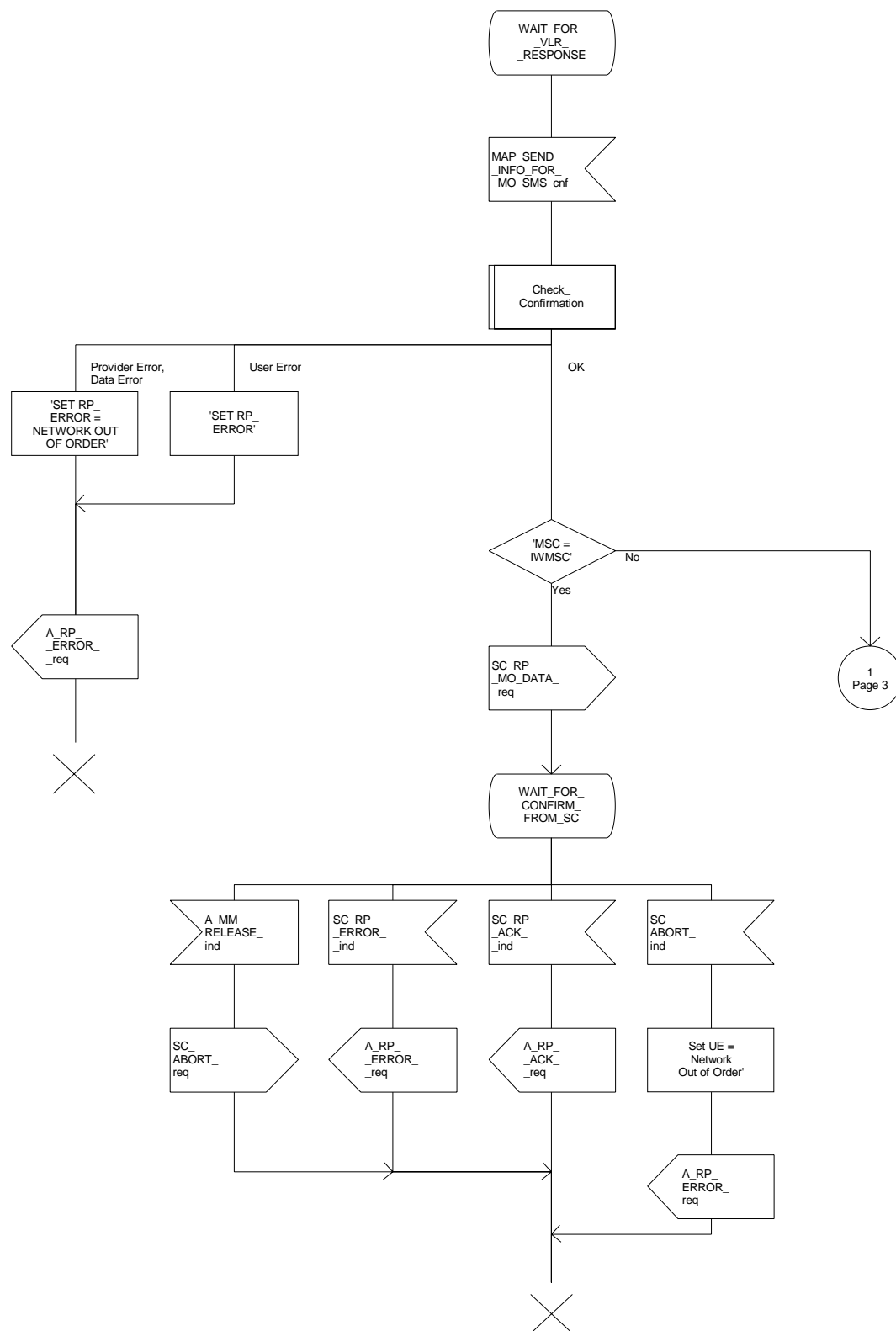
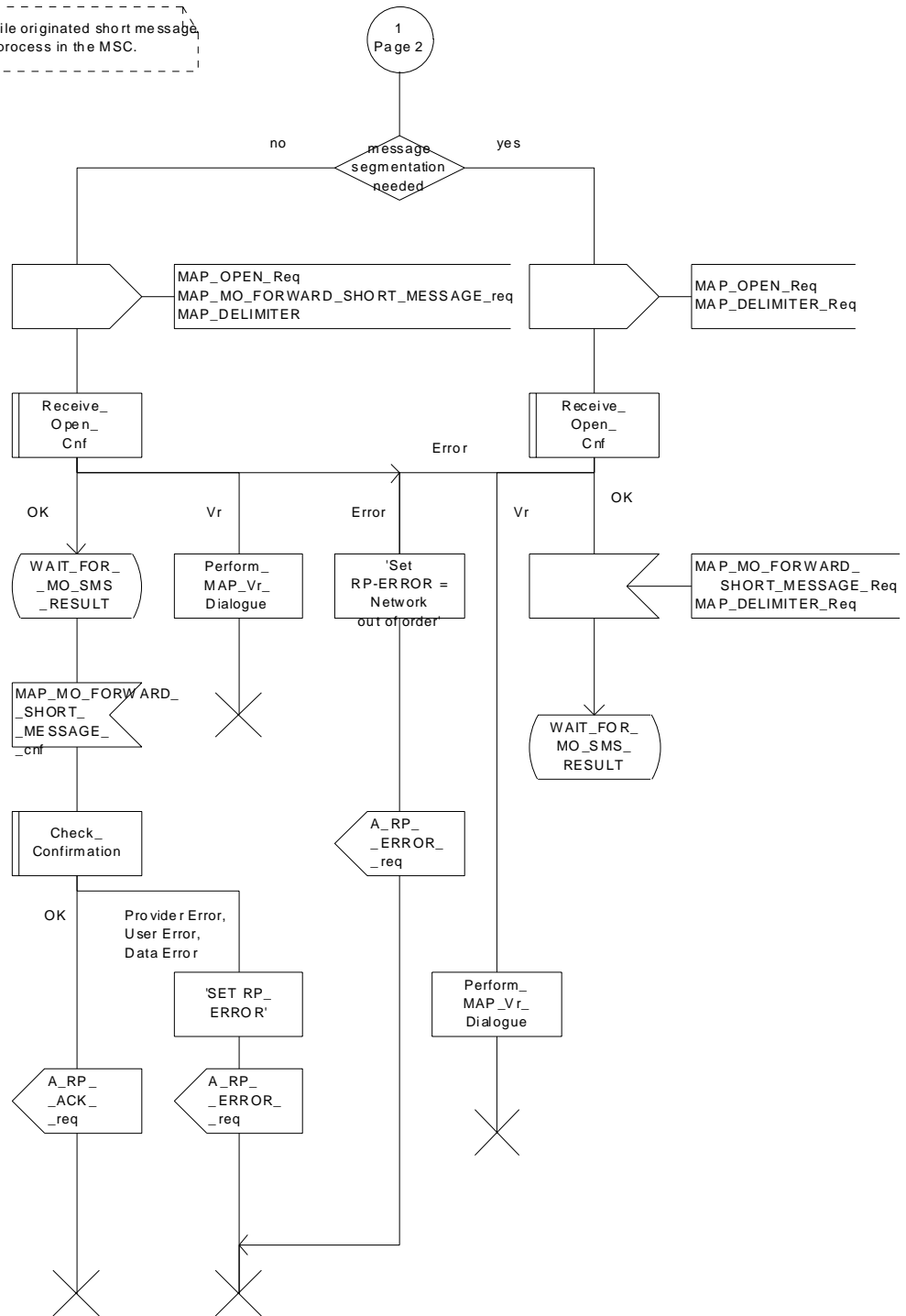


Figure 20.2/2 (sheet 3 of 3)

Process MOSM_MSC

202_23(3)

Figure 20.2/2: The mobile originated short message service process in the MSC.



20.2.2 Procedure in the VLR

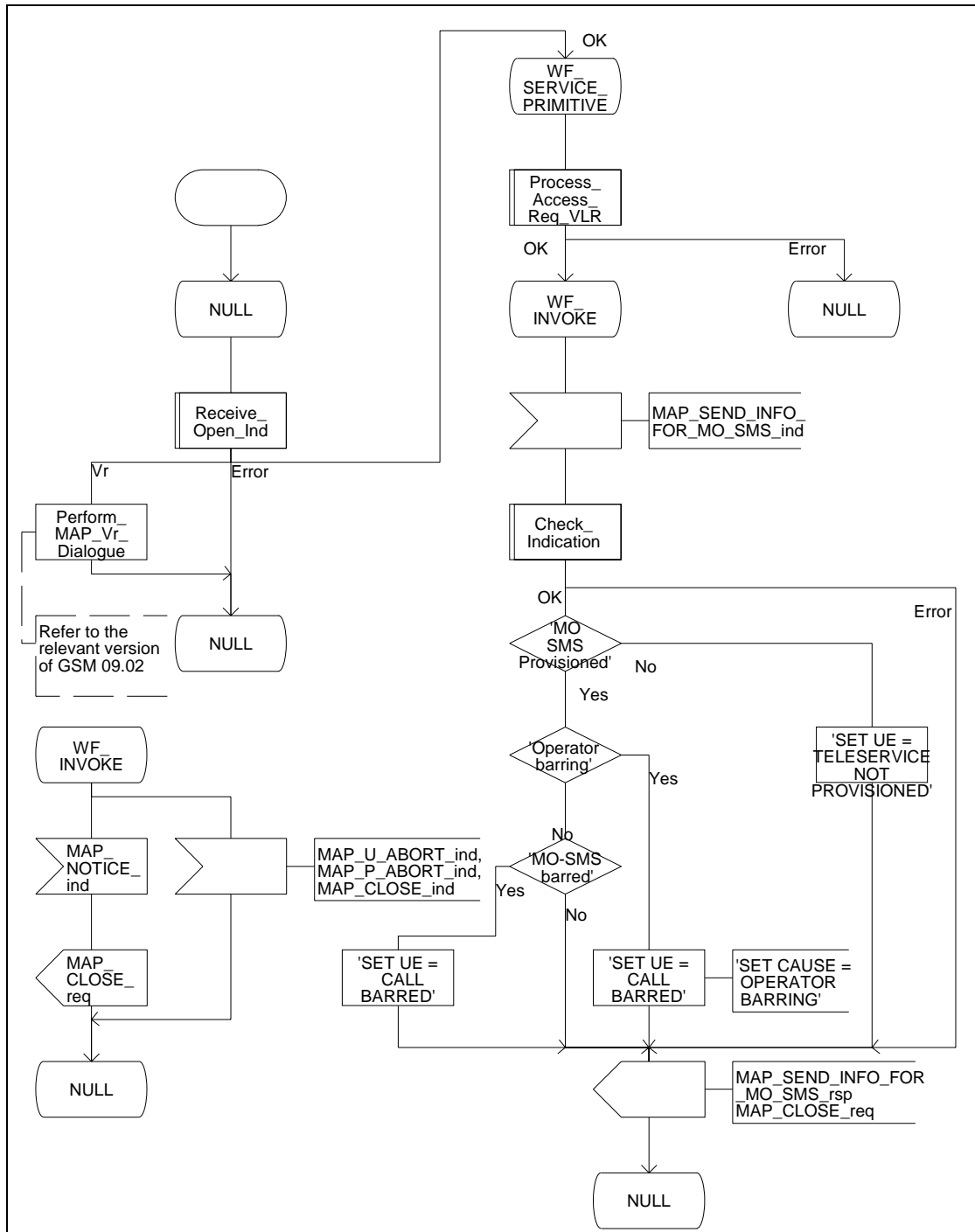
The MAP_PROCESS_ACCESS_REQUEST indication starts the MAP_PROCESS_ACCESS_REQUEST service in the VLR. The application context in the MAP_OPEN indication is mobile originated short message transfer.

If the service MAP_PROCESS_ACCESS_REQUEST is successful, the VLR waits for the next message from the MSC. When receiving the MAP_SEND_INFO_FOR_MO_SMS indication, the VLR acts as follows:

- if there is incompatibility in the subscription check, the error teleservice not provisioned is returned to the MSC;
- if the short message transfer would contravene operator determined barring, the call barred error with cause operator barring is returned;
- if the short message transfer would contravene the supplementary service call barring conditions in the VLR, the call barred error with cause barring service active is returned.

When the mobile subscriber has passed all checks, the MAP_SEND_INFO_FOR_MO_SMS response is initiated and the procedure is terminated in the VLR. The mobile originated short message transfer procedure in the VLR is shown in figure 20.2/3.

Figure 20.2/3



20.2.3 Procedure in the interworking MSC

This procedure applies only when the IWMSC is not the servicing MSC.

When receiving a MAP_OPEN indication primitive that is not associated with any MAP service indication primitive and if the dialogue is accepted, the MAP service-user in the interworking MSC issues a MAP_DELIMITER request primitive in order to trigger the local MAP service-provider to confirm the dialogue. Then a MAP_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.

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:

- if a short message is accepted by the Service Centre, an acknowledgement is sent back to the servicing MSC;
- if the Service Centre is not identified, the SM Delivery Failure error is returned to the servicing MSC;
- if the Service Centre returns an error indication, the SM Delivery Failure error is returned to the servicing MSC with the error cause and any diagnostic information received from the Service Centre;
- if the short message cannot be forwarded to the Service Centre or the procedure towards the Service Centre fails for some reason, a system failure error is sent to the servicing MSC.

The mobile originated short message service transfer in the IWMSC is shown in figure 20.2/4.

Figure 20.2/4

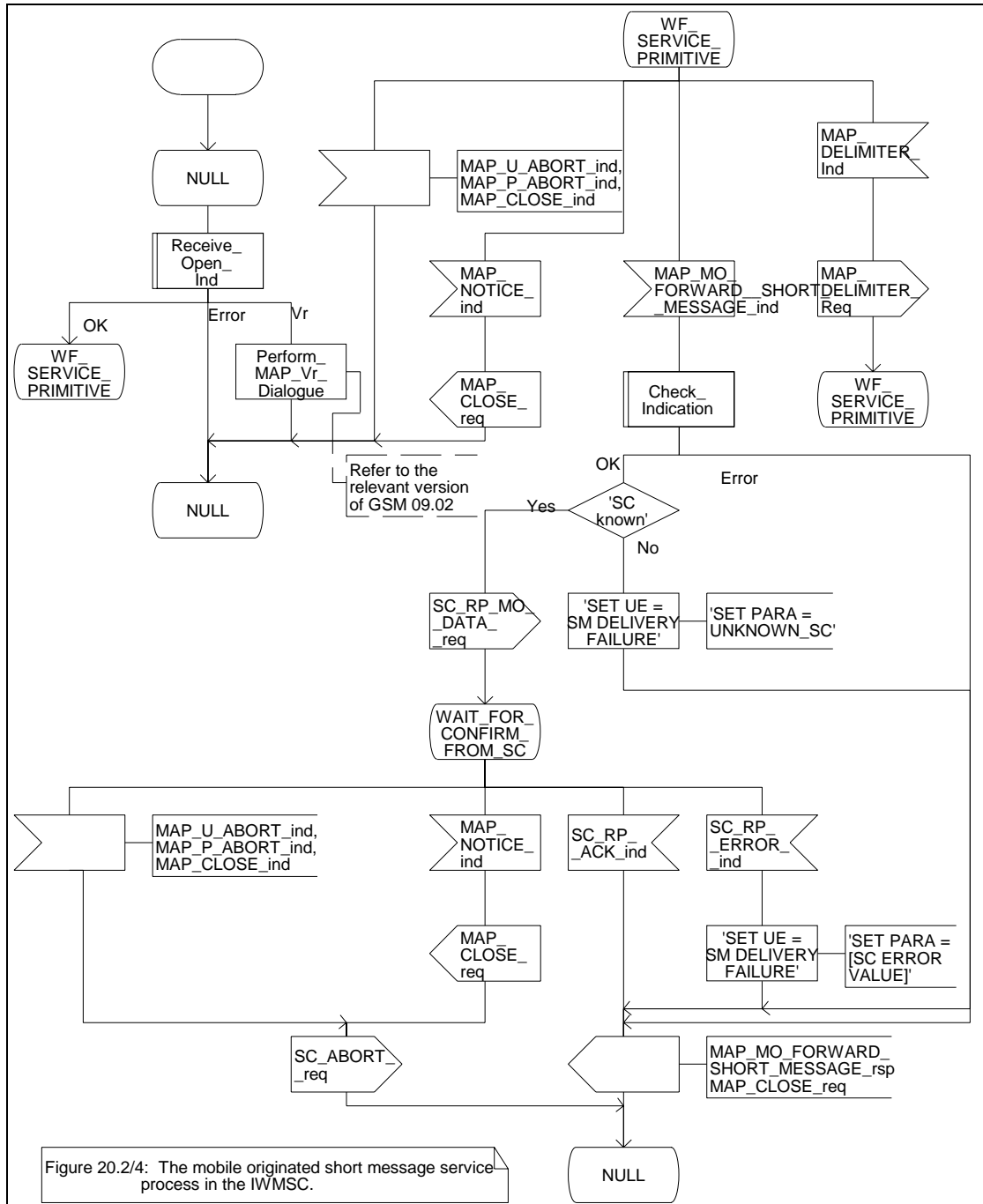
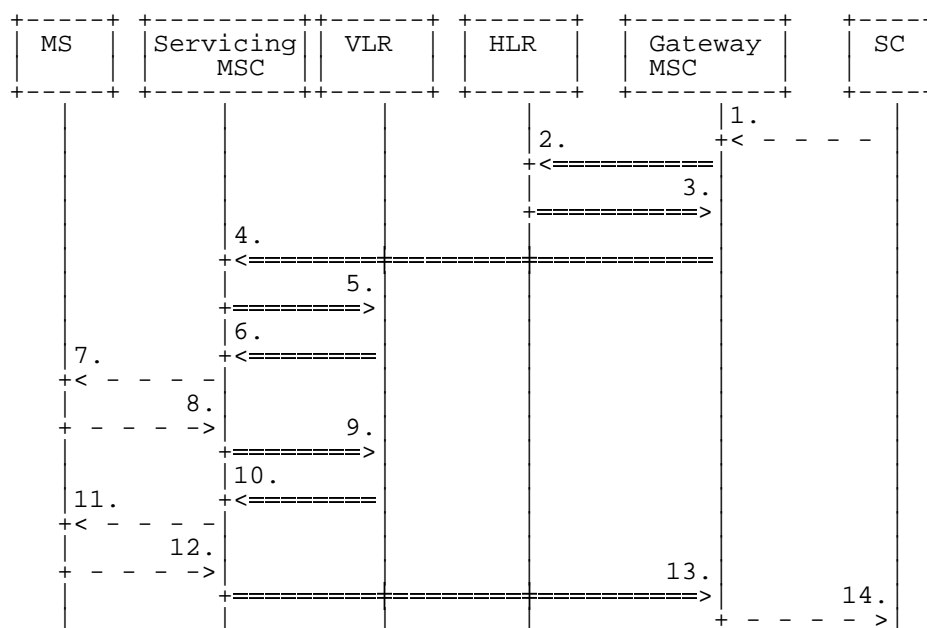


Figure 20.2/4: The mobile originated short message service process in the IWMSC.

20.3 The mobile terminated short message transfer procedure

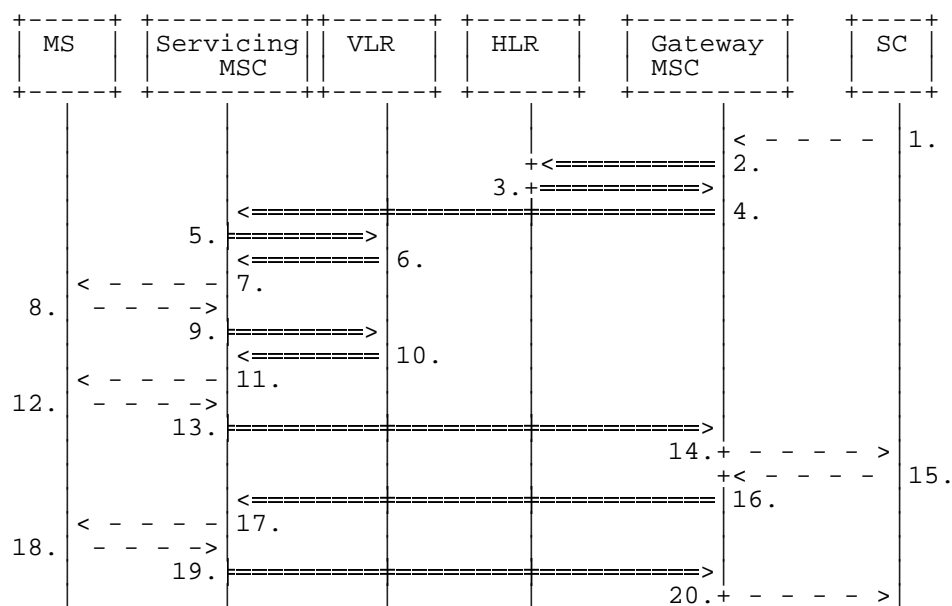
The mobile terminated short message transfer procedure is used for forwarding a short message or several short messages from a Service Centre to a mobile subscriber. The mobile terminated short message procedure for a single short message transfer is shown in figure 20.3/1.



- 1) Short Message (GSM 03.40)
- 2) MAP_SEND_ROUTING_INFO_FOR_SM
- 3) MAP_SEND_ROUTING_INFO_FOR_SM_ACK
- 4) MAP_MT_FORWARD_SHORT_MESSAGE
- 5) MAP_SEND_INFO_FOR_MT_SMS
- 6) MAP_PAGE/MAP_SEARCH_FOR_MOBILE_SUBSCRIBER
- 7) Page (BSSAP; TS GSM 04.08)
- 8) Page response (BSSAP; TS GSM 04.08)
- 9) MAP_PROCESS_ACCESS_REQUEST_ACK and MAP_SEARCH_FOR_MOBILE_SUBSCRIBER_ACK
- 10) MAP_SEND_INFO_FOR_MT_SMS_ACK
- 11) Short Message (BSSAP; TS GSM 04.11)
- 12) Short Message Acknowledgement (BSSAP; TS GSM 04.11)
- 13) MAP_MT_FORWARD_SHORT_MESSAGE_ACK
- 14) Short Message Acknowledgment (GSM 03.40)

Figure 20.3/1: Mobile terminated short message service procedures

The mobile terminated short message procedure for multiple short message transfer is shown in figure 20.3/2.



- 1) Short Message (GSM 03.40)
- 2) MAP_SEND_ROUTING_INFO_FOR_SM
- 3) MAP_SEND_ROUTING_INFO_FOR_SM_ACK
- 4) MAP_MT_FORWARD_SHORT_MESSAGE (note 1)
- 5) MAP_SEND_INFO_FOR_MT_SMS
- 6) MAP_PAGE/MAP_SEARCH_FOR_MOBILE_SUBSCRIBER
- 7) Page (BSSAP; TS GSM 08.08)
- 8) Page response (BSSAP; TS GSM 04.08)
- 9) MAP_PROCESS_ACCESS_REQUEST_ACK and MAP_SEARCH_FOR_MOBILE_SUBSCRIBER_ACK
- 10) MAP_SEND_INFO_FOR_MT_SMS_ACK
- 11) Short Message (BSSAP; TS GSM 04.11)
- 12) Short Message Acknowledgment (BSSAP; TS GSM 04.11)
- 13) MAP_MT_FORWARD_SHORT_MESSAGE_ACK
- 14) Short Message Acknowledgment (TS GSM 03.40)
- 15) Short Message (TS GSM 03.40)
- 16) MAP_MT_FORWARD_SHORT_MESSAGE (note 2)
- 17) Short Message (BSSAP; TS GSM 04.11)
- 18) Short Message Acknowledgment (BSSAP; TS GSM 04.11)
- 19) MAP_MT_FORWARD_SHORT_MESSAGE_ACK
- 20) Short Message Acknowledgment (TS GSM 03.40)

NOTE 1: The More Messages To Send flag is TRUE.

NOTE 2: The More Messages To Send flag is FALSE

Figure 20.3/2: Mobile terminated short message procedure for multiple short message transfer.

In the multiple short message transfer the service MAP_MT_FORWARD_SHORT_MESSAGE can be used several times. However, the short message transfer is always acknowledged to the Service Centre before the next short message is sent.

In addition the following MAP services are used:

MAP_PROCESS_ACCESS_REQUEST	(see subclause 6.3);
MAP_PAGE	(see subclause 6.2);
MAP_SEARCH_FOR_MS	(see subclause 6.2);
MAP_AUTHENTICATE	(see subclause 6.5);
MAP_SET_CIPHERING_MODE	(see subclause 6.6);
MAP_CHECK_IMEI	(see subclause 6.7);
MAP_FORWARD_NEW_TMSI	(see subclause 6.9);
MAP_REPORT_SM_DELIVERY_STATUS	(see subclause 10.3);
MAP_INFORM_SERVICE_CENTRE	see subclause 10.6);
MAP_TRACE_SUBSCRIBER_ACTIVITY	(see subclause 7.1);
MAP_READY_FOR_SM	(see subclause 10.4).

20.3.1 Procedure in the Servicing MSC

When initiating the dialogue with the servicing MSC, the SMS Gateway MSC must provide the IMSI of the subscriber to whom the short message is directed.

The IMSI can be included either in the Destination Reference of the MAP_OPEN indication received from the SMS Gateway MSC or in the sm-RP-DA information field of the MAP_MT_FORWARD_SHORT_MESSAGE indication.

When receiving a MAP_OPEN indication primitive that is not associated with any MAP service indication primitive and if the dialogue is accepted, the MAP service-user in the servicing MSC issues a MAP_DELIMITER request primitive in order to trigger the local MAP service-provider to confirm the dialogue.

When receiving the first MAP_MT_FORWARD_SHORT_MESSAGE indication from the gateway MSC, the servicing MSC sends the MAP_SEND_INFO_FOR_MT_SMS request primitive to the VLR, if the MAP service primitive is accepted and if short message service is supported in the servicing MSC.

The MAP_MT_FORWARD_SHORT_MESSAGE indication primitive is checked by the macro "Check_Indication". If the received MAP service primitive contains errors, the service is aborted and an unexpected data value error or data missing error is returned to the GMSC.

If the MSC does not support the short message service, the service is aborted in the servicing MSC and the error "Facility Not Supported" is returned to the GMSC.

The subscriber identity information that may be included in the MAP_OPEN indication primitive and in the MAP service indication primitive is checked by the macro "Check_Subscr_Identity_For_MT_SMS" as follows.

If a Destination Reference has been received in the MAP_OPEN indication, an LMSI must be present in the sm-RP-DA information field of the MAP_MT_FORWARD_SHORT_MESSAGE indication. The LMSI shall be included in the sm-RP-DA information field of the MAP_SEND_INFO_FOR_MT_SMS request sent to the VLR; the associated MAP_OPEN request must contain a Destination Reference that carries an IMSI.

Otherwise, if the IMSI is included in the sm-RP-DA information field of the MAP_MT_FORWARD_SHORT_MESSAGE indication, it is mapped into the sm-RP-DA information field of the MAP_SEND_INFO_FOR_MT_SMS request that is sent to the VLR. In this case, the IMSI is not accompanied by an LMSI and neither the MAP_OPEN indication received from the gateway MSC nor the MAP_OPEN request sent to the VLR shall include a Destination Reference.

If a Destination Reference has been received in the servicing MSC and the sm-RP-DA information field of the MAP_MT_FORWARD_SHORT_MESSAGE indication does not include an LMSI or if no Destination Reference has been received and the sm-RP-DA information field does not cover an IMSI the service is aborted in the servicing MSC and the error "Unexpected Data Value" is returned to the SMS GMSC.

The following responses to the MAP_SEND_INFO_FOR_MT_SMS request may be received from the VLR:

- unidentified subscriber or system failure error. The error code is forwarded to the GMSC;
- absent subscriber error. The absent subscriber_SM error is forwarded to the GMSC with the absent subscriber diagnostic indication set to 'IMSI Detached';
- unknown subscriber error. The system failure indication is provided to the GMSC;
- data missing or unexpected data value error. The system failure indication is provided to the GMSC;
- a provider error or an abort indication. The system failure indication is provided to the GMSC;
- subscriber busy for MT SMS. The error code is forwarded to the GMSC;
- paging procedure invocation (see subclause 21.3) reporting the successful outcome of the procedure;
- search procedure invocation (see subclause 21.3) reporting the successful outcome of the procedure.

The result of the paging or the search procedure is processed as follows:

- if the procedure is completed successfully, the MSC will send the MAP_PROCESS_ACCESS_REQUEST request to the VLR (see subclause 21.4);
- if the procedure is completed successfully, but the MS has no mobile terminated short message transfer capability, the procedure is terminated and SM delivery failure indication with cause "equipment not SM equipped" is provided to the GMSC;
- if the procedure ends unsuccessfully, the termination of the procedure is awaited from the VLR. The absent subscriber_SM error is forwarded to the GMSC with the absent subscriber diagnostic indication set to 'No Paging Response', but the other error causes are reported as a system failure indication.

If the short message transfer is aborted for any reason, the dialogue with the VLR is aborted. If the procedure with the VLR is aborted by the VLR or by the provider, a system failure indication is provided to the GMSC.

The unsuccessful outcome of the MAP_PROCESS_ACCESS_REQUEST service is reported by using the system failure error to the GMSC.

When the service MAP_PROCESS_ACCESS_REQUEST is carried out, the MSC will receive the MAP_SEND_INFO_FOR_MT_SMS confirmation indicating:

- the unsuccessful outcome of the procedure. The error indication received from the VLR is forwarded to the GMSC;
- the successful outcome of the procedure. The MSC initiates forwarding of the short message to the MS.

If the primitive itself is badly formatted or data is missing, the system failure error is sent to the GMSC.

If forwarding of the short message is initiated, the MSC awaits the result before one of the following responses is sent back to the GMSC:

- an acknowledge if the short message has been successfully delivered to the mobile subscriber;
- an SM delivery failure error containing a parameter indicating either of the following: there is a MS protocol error or the MS memory capacity is exceeded; detailed diagnostic information (see subclause 5.6.1.4) may also be carried;
- a system failure error if the delivery procedure is aborted.

If the More Messages To Send flag was FALSE or the service MAP_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 17.

The mobile terminated short message transfer procedure in the servicing MSC is shown in figures 20.3/3 and 20.3/4. The page and search procedures are shown in figure 21.3/1 and 21.3/2.

Figure 20.3/3 (sheet 1 of 3): Procedure MT_SM_Transfer_MSC

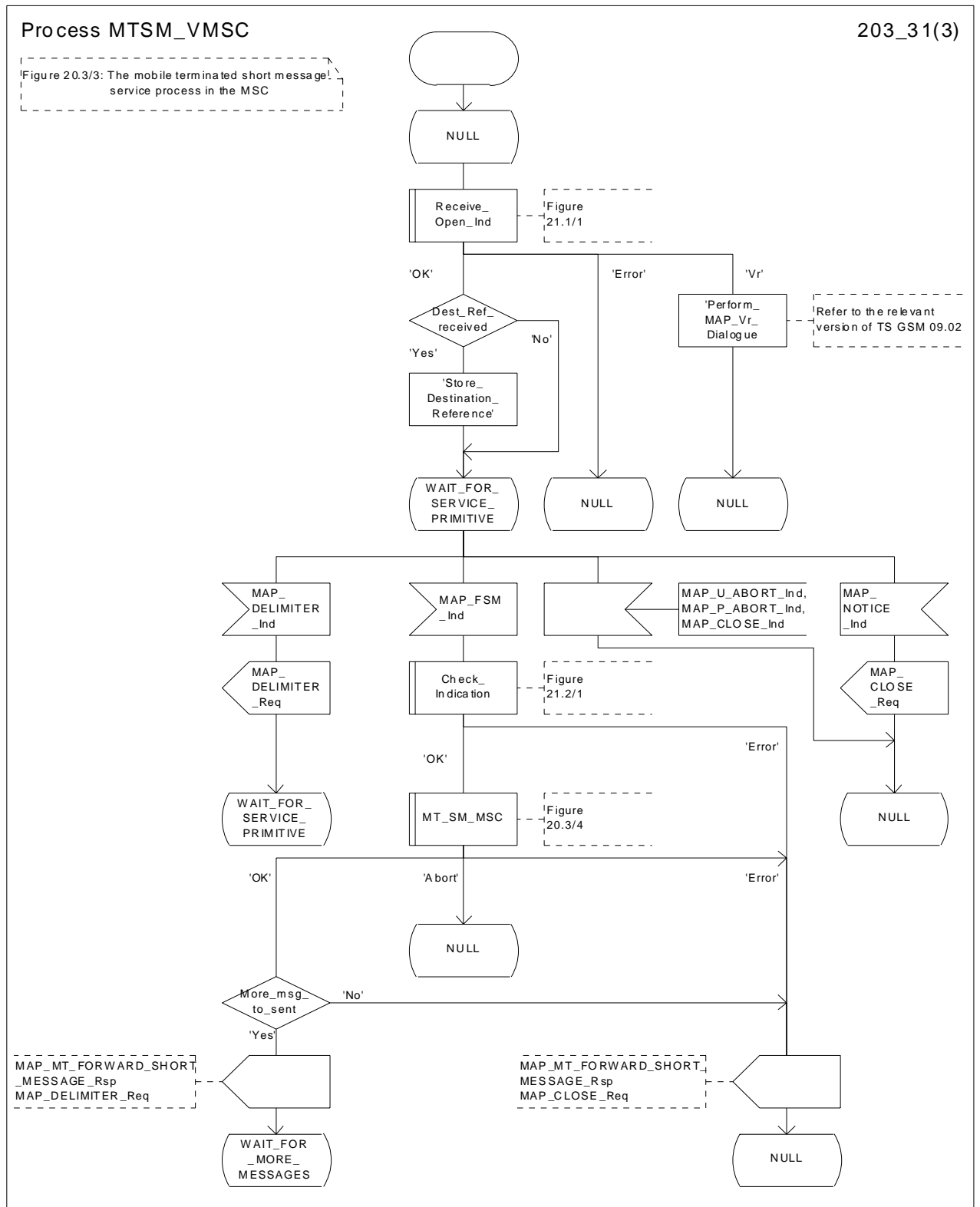


Figure 20.3/3 (sheet 2 of 3): Procedure MT_SM_Transfer_MSC

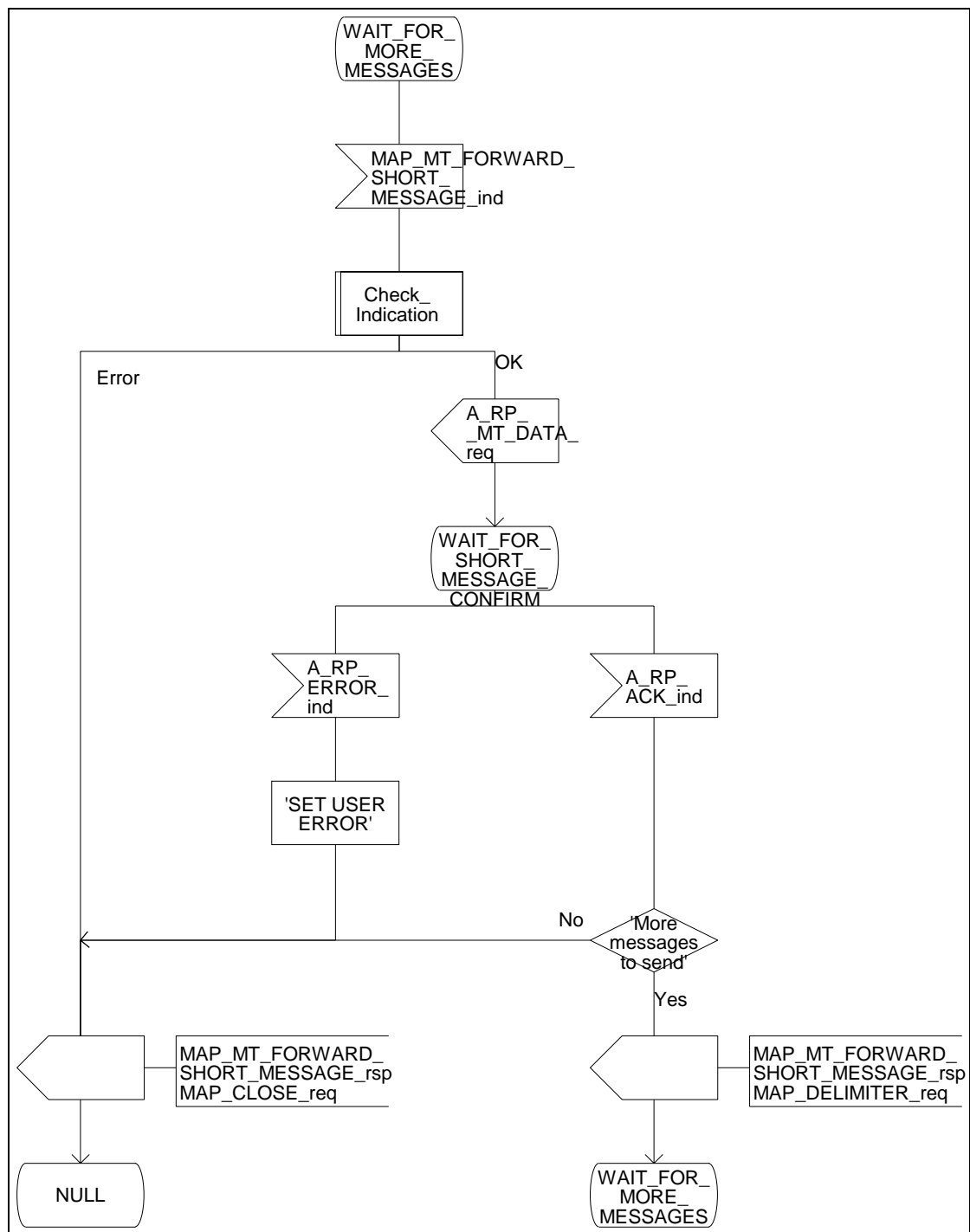


Figure 20.3/3 (sheet 3 of 3): Procedure MT_SM_Transfer_MSC

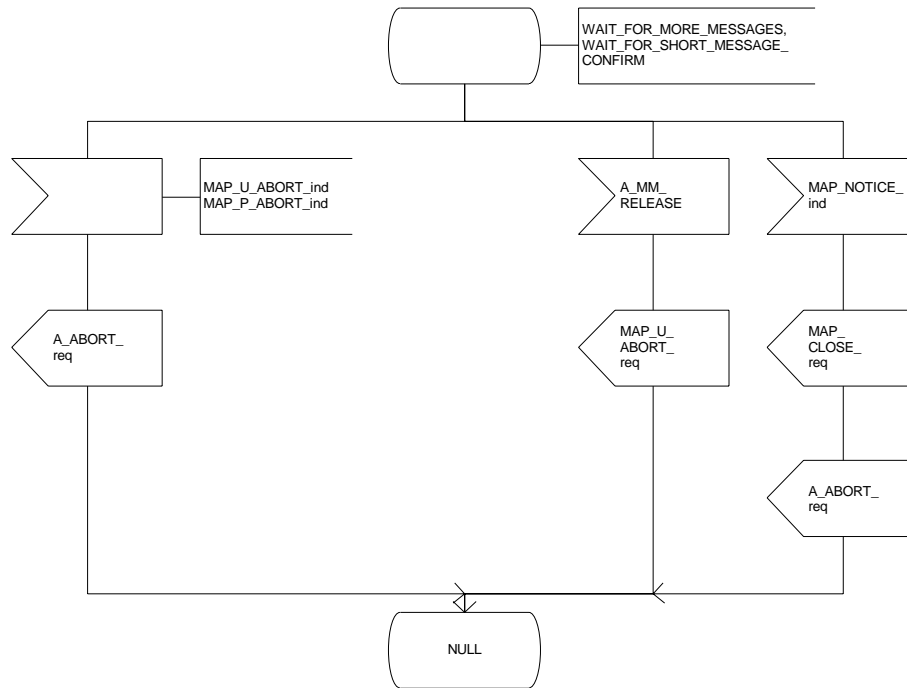


Figure 20.3/4 (sheet 1 of 3): Macro MT_SM_MSC

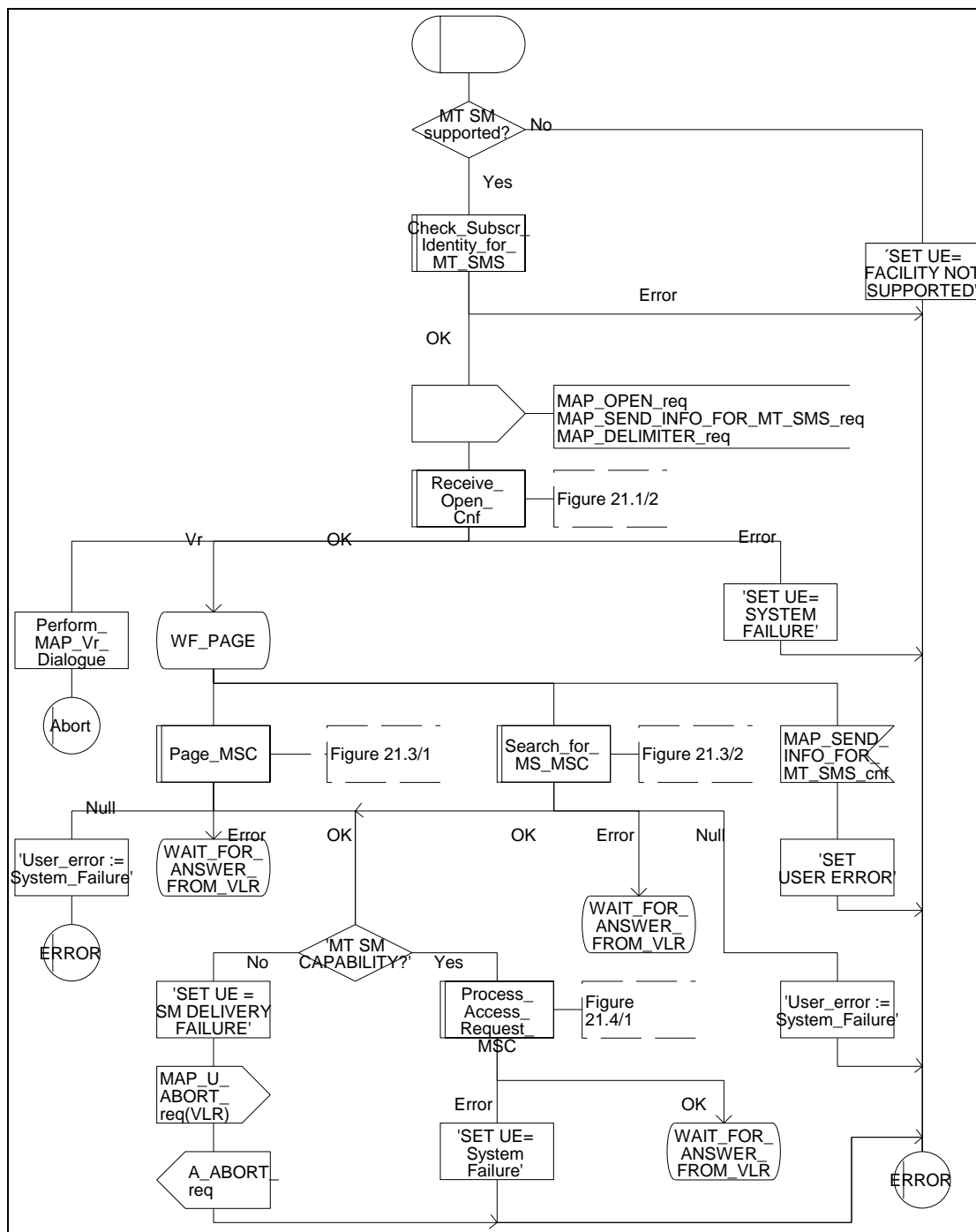


Figure 20.3/4 (sheet 2 of 3): Macro MT_SM_MSC

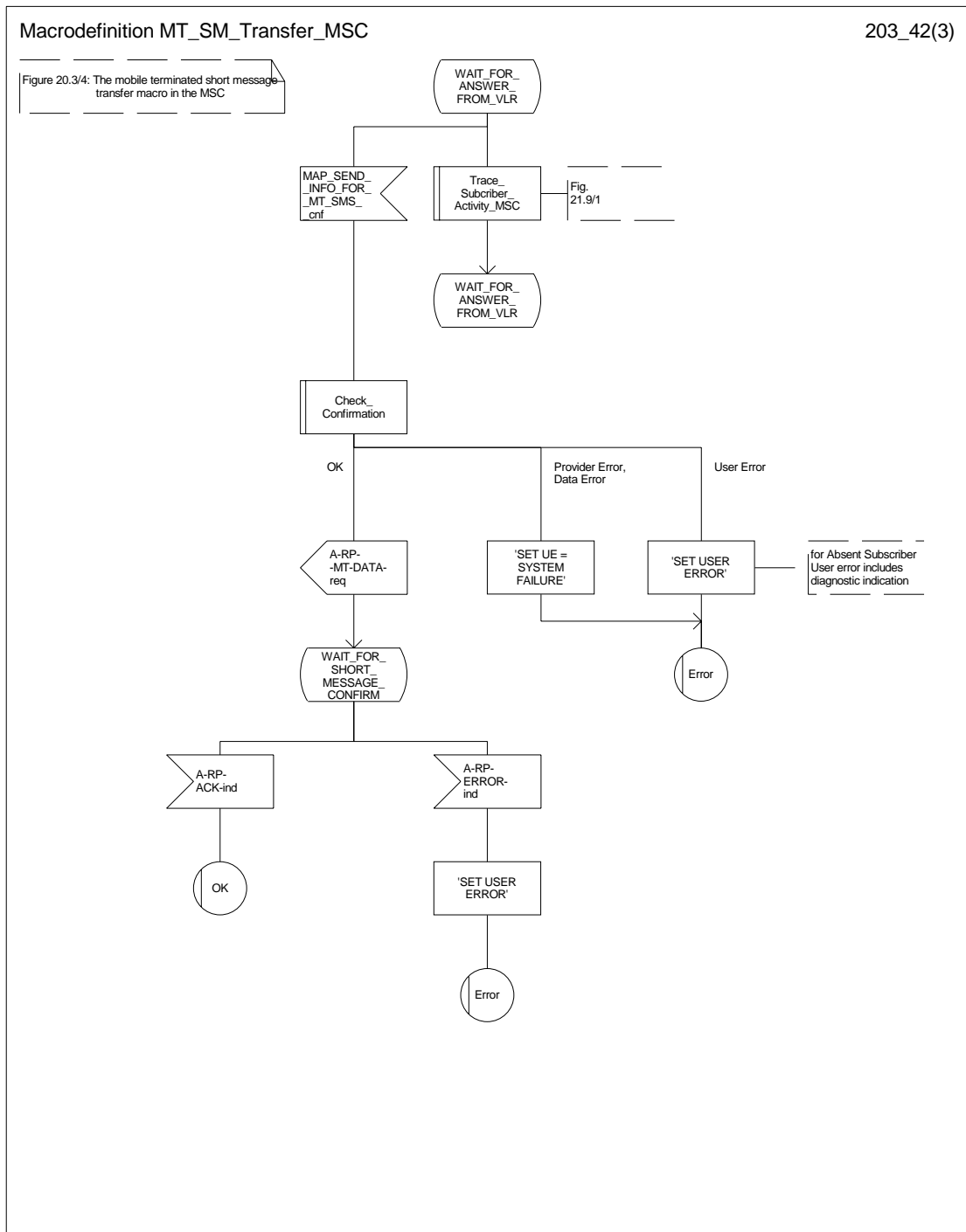
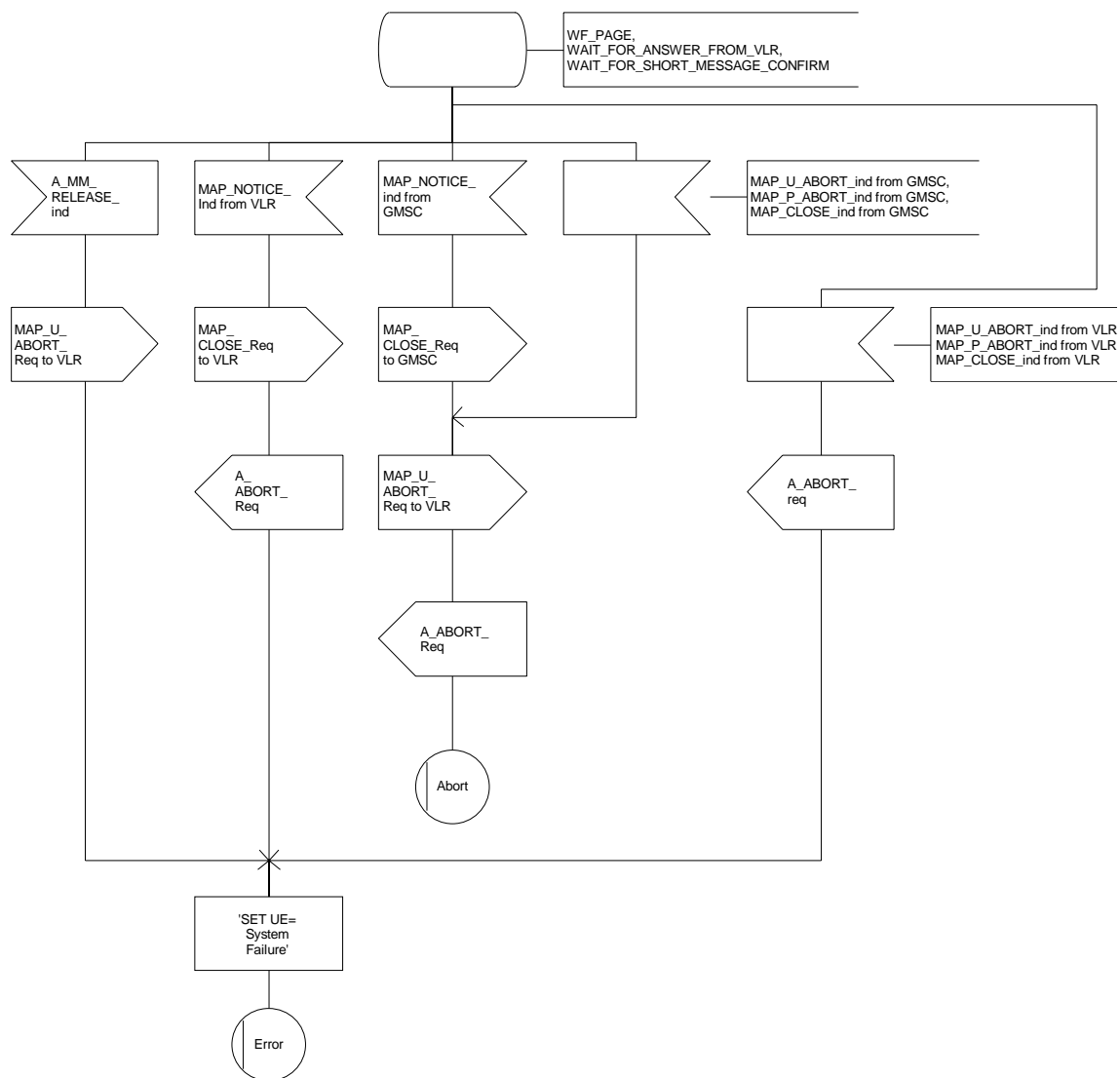


Figure 20.3/4 (sheet 3 of 3): Macro MT_SM_MSC



20.3.2 Procedures in the VLR

When receiving the MAP_SEND_INFO_FOR_MT_SMS indication, the VLR will act as follows:

- the parameters and data in the primitive are checked by the macro "Check_Indication". A data failure is reported as an unexpected data value error or a data missing error depending on the nature of the failure;
- for mobile terminated short message the mobile subscriber is identified either by the IMSI only or by the IMSI accompanied by the LMSI. The subscriber identity information that may be included in the MAP_OPEN indication primitive and in the MAP service indication primitive is checked by the macro "Check_Subscr_Identity_For_MT_SMS". In the first case, the IMSI is included in the sm-RP-DA information field and the Destination Reference must not be present in the MAP_OPEN primitive. In the latter case the IMSI must be obtained from the Destination Reference of the MAP_OPEN indication primitive and an LMSI must be present in the sm-RP-DA information field of the MAP_SEND_INFO_FOR_MT_SMS indication. If the mobile subscriber is unknown, the unidentified subscriber error is returned;
- if the "Confirmed by HLR" indicator is set to "Not Confirmed", the unidentified subscriber error is returned;
- if the IMSI Detached Flag is set to detached or the LA Not Allowed Flag is set to not allowed in the VLR, an absent subscriber error 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'. 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 20.3/5.

Figure 20.3/5 (sheet 1 of 3): Process MT_SM_VLR

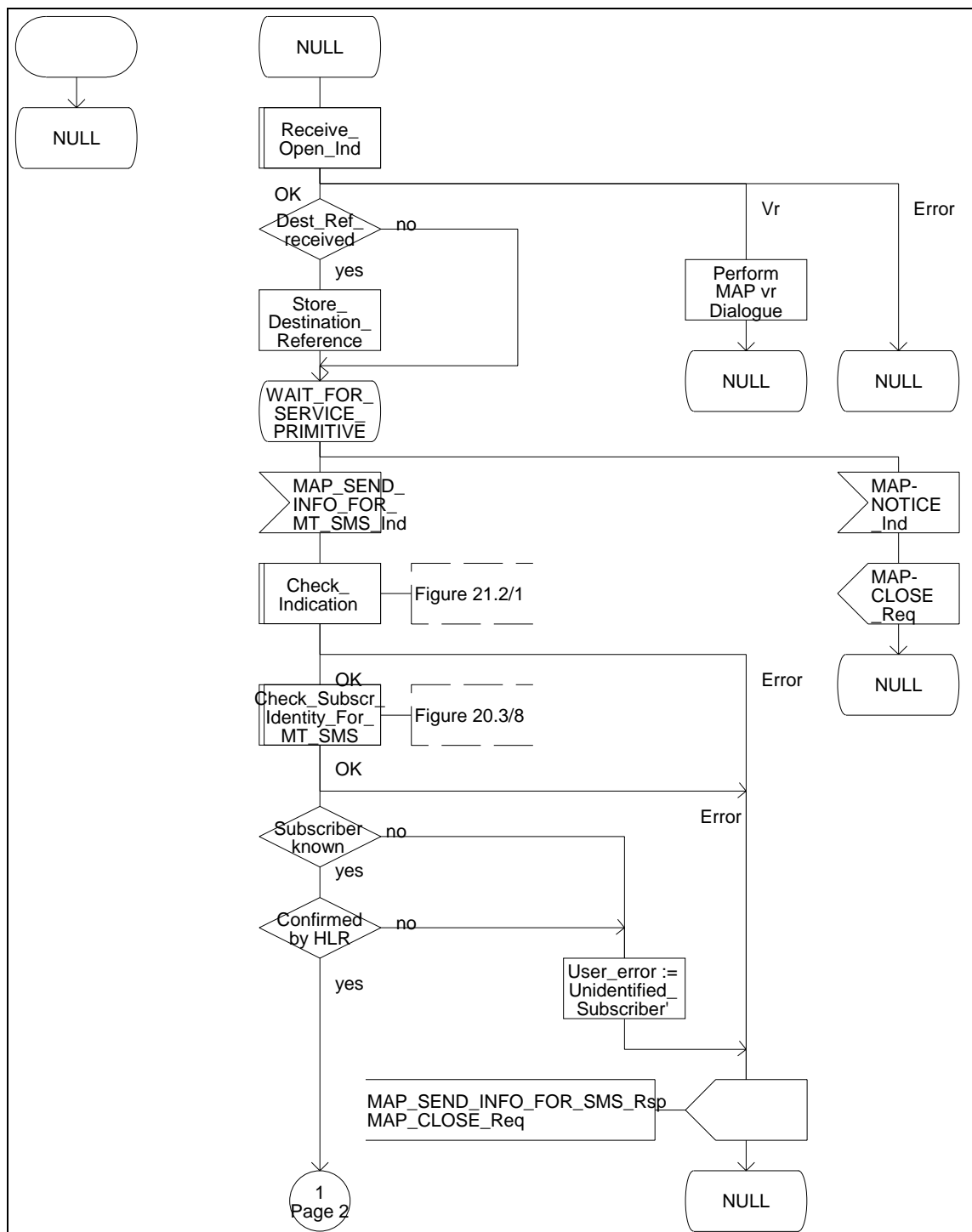


Figure 20.3/5 (sheet 2 of 3): Process MT_SM_VLR

Process MT_SM_VLR

203_52(3)

Figure 20.3/5: The mobile terminated short message service process in the VLR

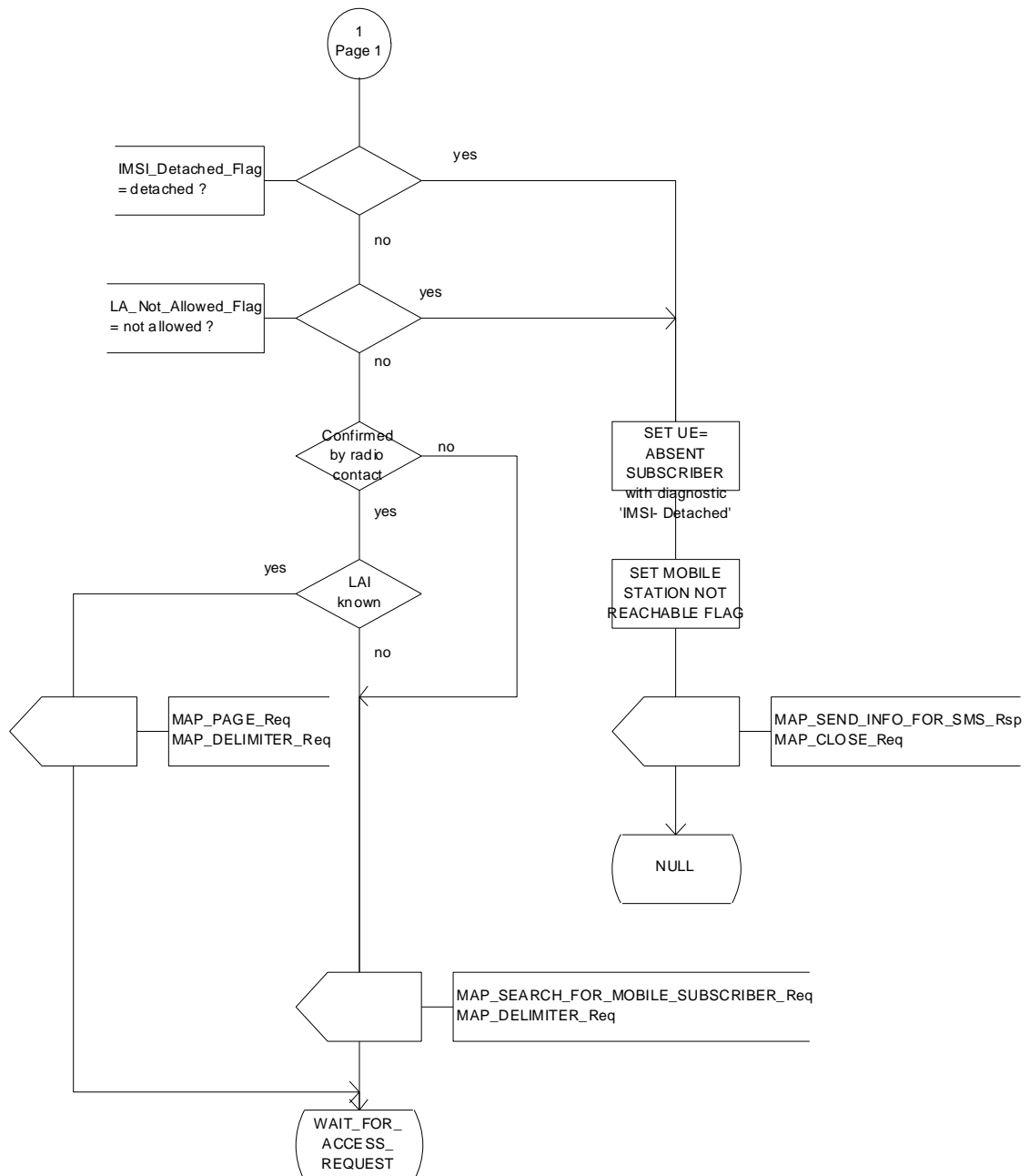
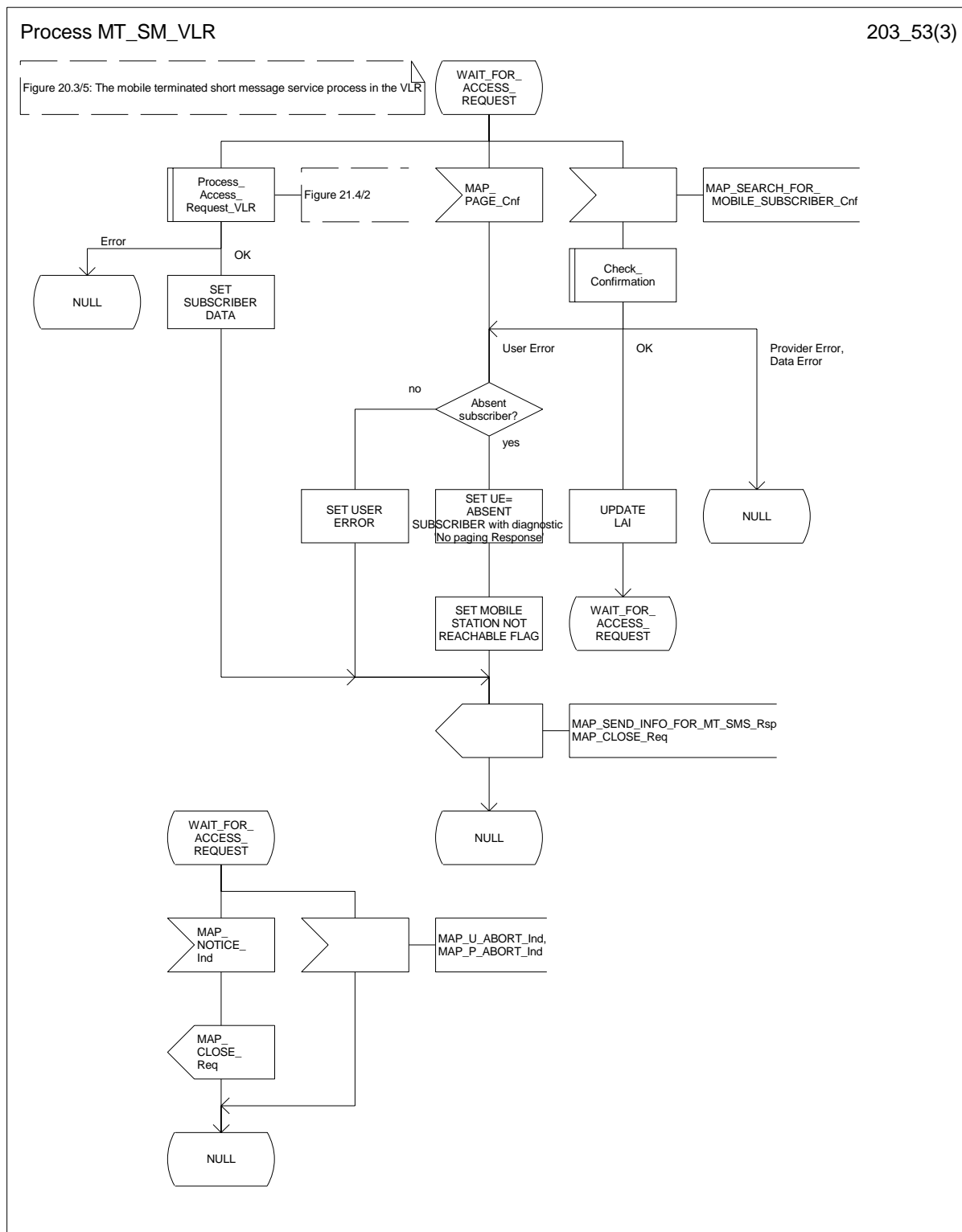


Figure 20.3/5 (sheet 3 of 3): Process MT_SM_VLR



20.3.3 Procedures in the HLR

The MAP_SEND_ROUTING_INFO_FOR_SM indication is received from the GMSC. The following error cases are reported to the GMSC in the MAP_SEND_ROUTING_INFO_FOR_SM response as an unsuccessful outcome of the procedure:

- if the necessary parameters and data are not present in the primitive or they are badly formatted, the data missing or unexpected data value error is returned;
- if the mobile subscriber is unknown, i.e. it cannot be identified from the MSISDN given, an unknown subscriber error is returned;
- if the short message transfer would contravene operator determined barring, the call barred error with cause operator barring is returned;
- if the short message transfer would contravene the supplementary service barring, the call barred error with cause barring service active is returned;
- if the mobile subscription identified by the given MSISDN number does not include the short message service, the teleservice not provisioned error is returned;
- if the location registration of the mobile subscriber shows that the visited PLMN does not support the MT short message service, the facility not supported error is returned;
- if no MSC identity is stored for the mobile subscriber or the "MSC Area Restricted Flag" is set or the "MS purged" flag is set, i.e. the MS is not reachable, the MSISDN-Alert and the SC address are included in the MWD (if possible), 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', 'Roaming Restricted' or 'MS-Purged'.

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 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 is retrieved as described below;
- if the priority is high (SM_RP_PRI = True) and the MNRF is set, the HLR will send the acknowledge primitive containing the routing information to the gateway MSC. In addition the service MAP_INFORM_SERVICE_CENTRE including the parameter MW Status is invoked to indicate whether or not the SC address is already included in the MWD list.

If the MSISDN-Alert number of the mobile subscriber stored in the MWD is not the same as that received in the MAP_SEND_ROUTING_INFO_FOR_SM indication, the HLR will include in the MAP_INFORM_SERVICE_CENTRE request to the GMSC the MSISDN-Alert number stored.

The MAP_INFORM_SERVICE_CENTRE request is sent also when the MCEF and/or MNRF are set but the routing information is still sent to the GMSC. The status of the flags is indicated in the parameter MW Status.

The routing information is included in a MAP_SEND_ROUTING_INFO_FOR_SM response as follows:

- the IMSI will be returned to the GMSC together with the MSC number and may be optionally accompanied by the LMSI.

The mobile terminated short message transfer procedure in the HLR is shown in figure 20.3/6.

Figure 20.3/6 (sheet 1 of 3): Process MT_SM_HLR

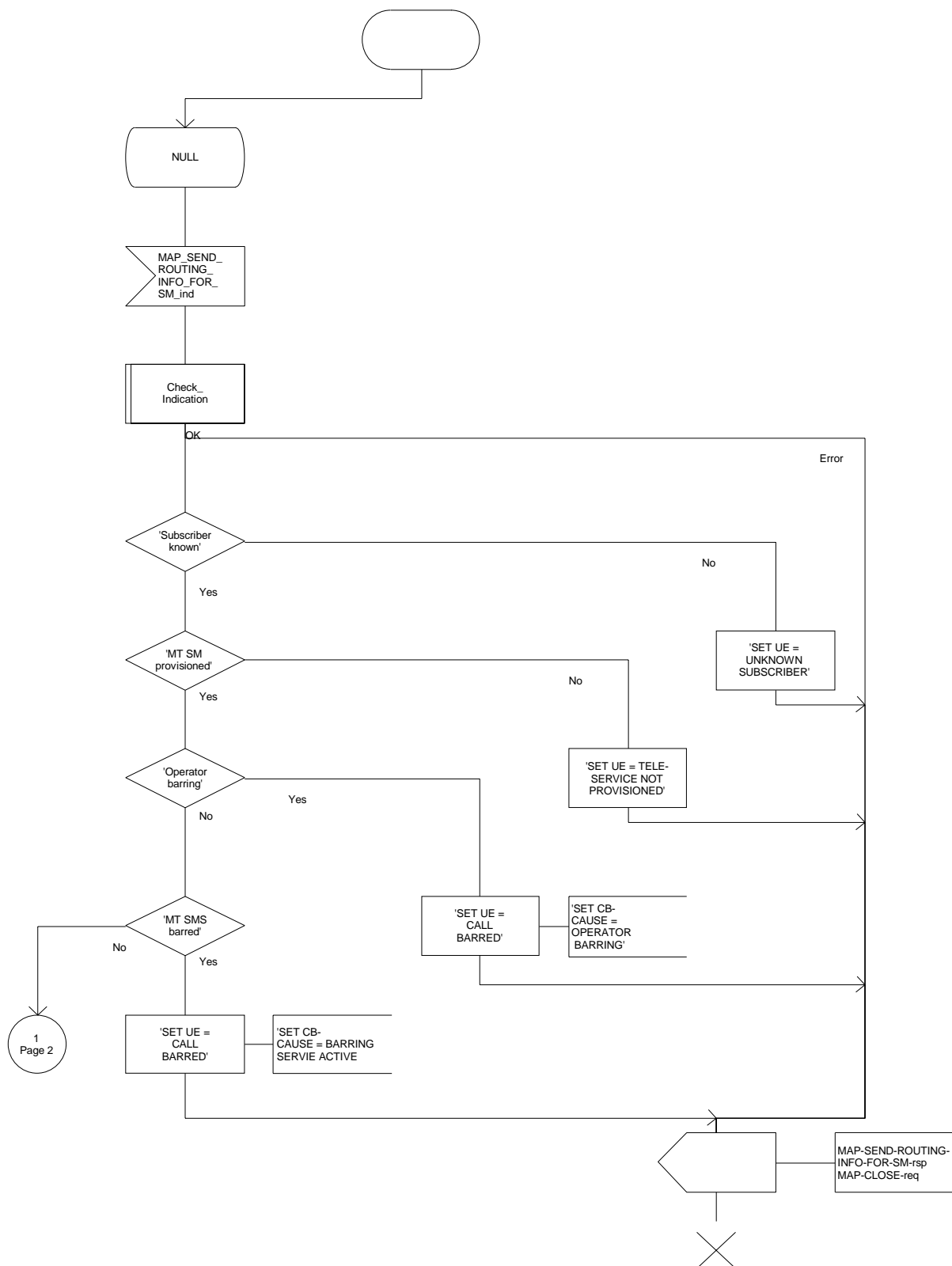


Figure 20.3/6 (sheet 2 of 3): Process MT_SM_HLR

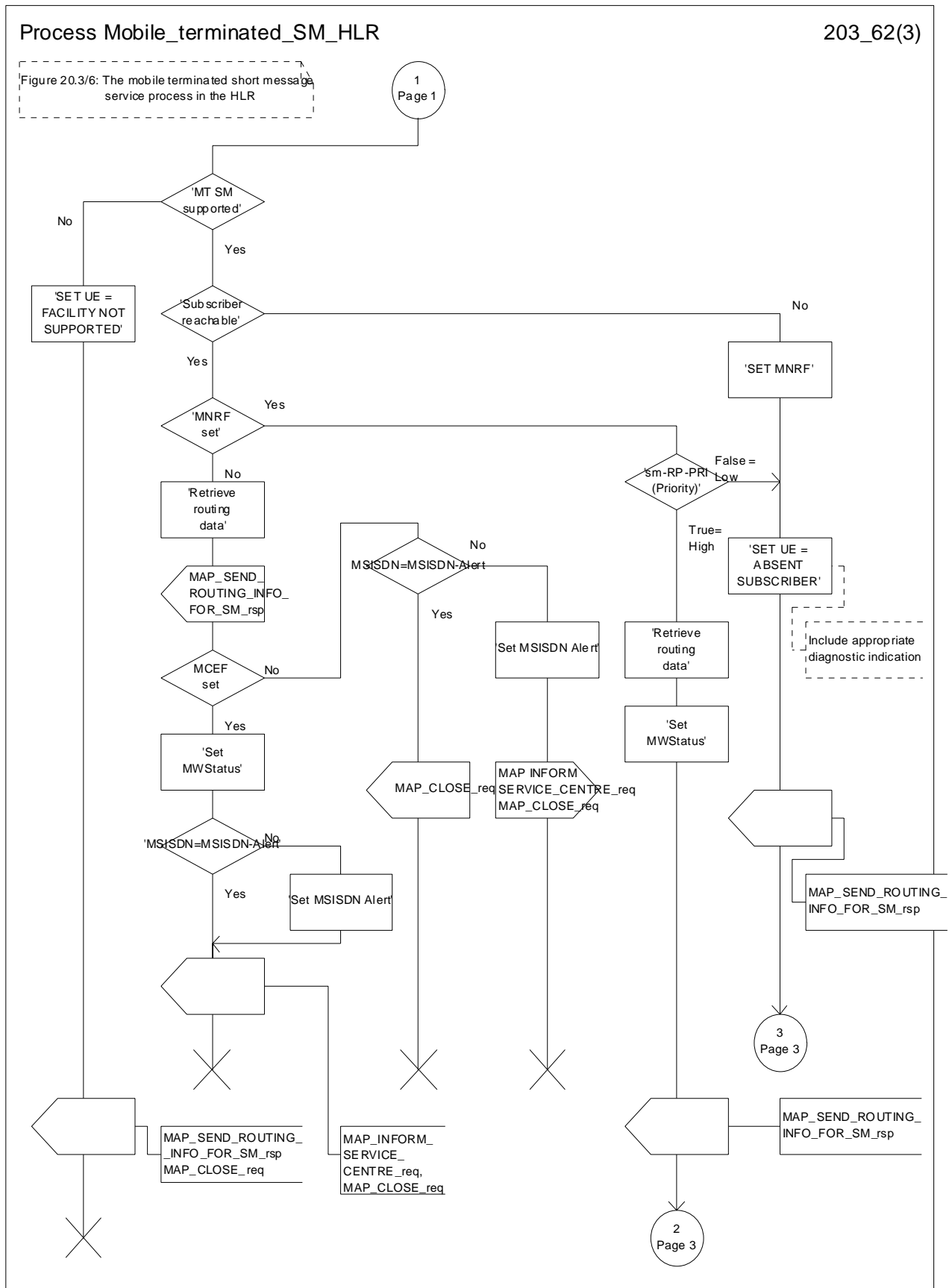
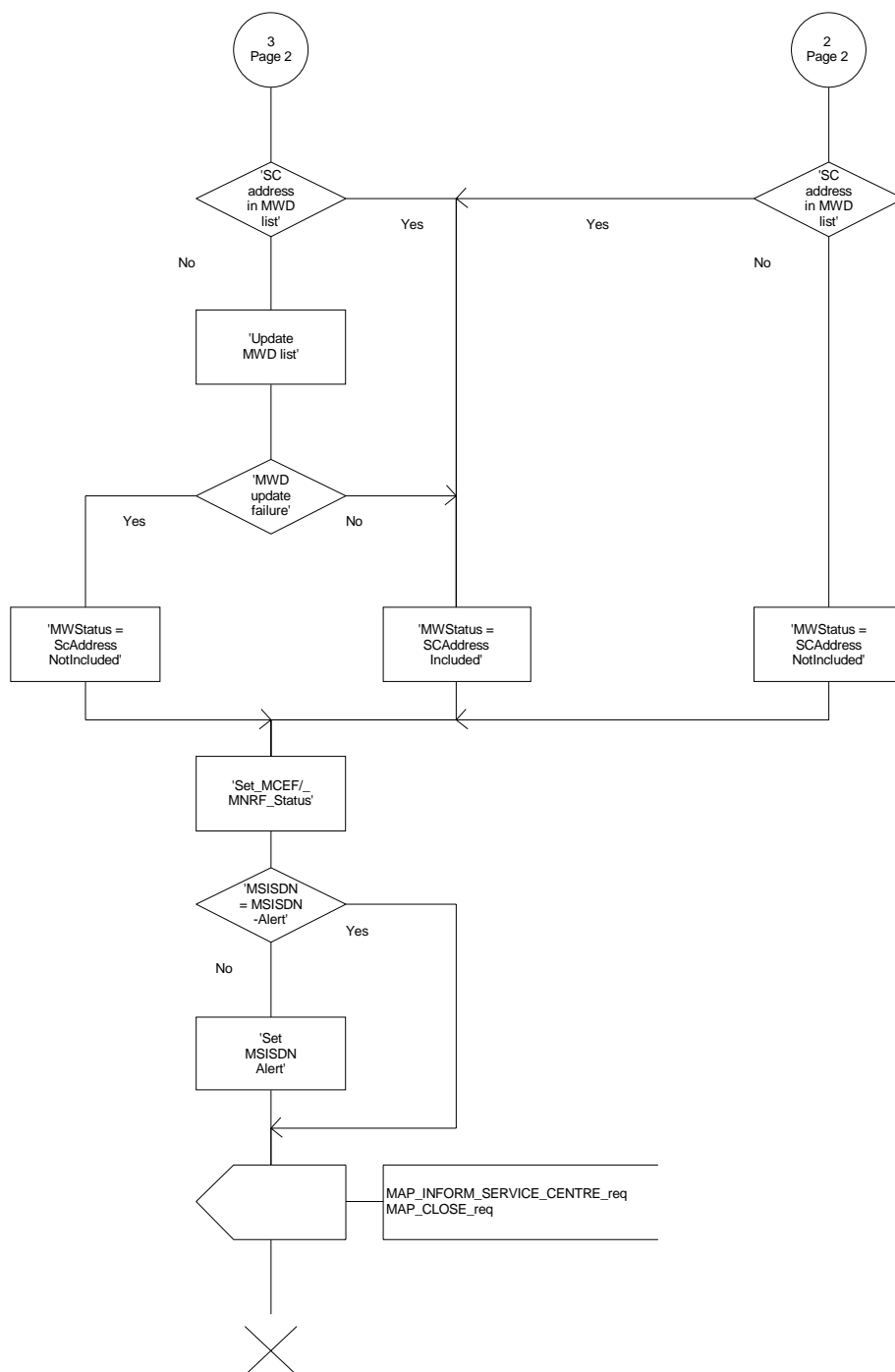


Figure 20.3/6 (sheet 3 of 3): Process MT_SM_HLR



20.3.4 Procedures in the gateway MSC

The short message handling function of the GMSC will request routing information when a mobile terminated short message is received from a Service Centre. The GMSC sends the MAP_SEND_ROUTING_INFO_FOR_SM request to the HLR containing the subscriber data of the mobile subscriber.

As an outcome of the procedure the MAP_SEND_ROUTING_INFO_FOR_SM confirmation is received indicating:

- an unsuccessful event indication containing an error;
The mapping between the MAP error causes and the RP_ERROR causes is explained in TS GSM 03.40.
- a successful event indication containing following parameters:
 - an IMSI optionally accompanied by an LMSI; and
 - a routing address (a servicing MSC address).

The GMSC may also receive a MAP_INFORM_SERVICE_CENTRE indication after the MAP_SEND_ROUTING_INFO_FOR_SM confirmation. The parameter MW Status in the message indicates whether or not the Service Centre address is stored in the Message Waiting Data. It also indicates the status of the MCEF and MNRF flags in the HLR.

If the MSISDN-Alert stored in the MWD data is not the same as the one sent to the HLR, the MSISDN-Alert is received in the MAP_INFORM_SERVICE_CENTRE indication. This MSISDN number shall be transferred in a delivery failure report to the SC.

In the abnormal end or in the provider error case the system failure error is provided to the SC.

The forward short message procedure is initiated when the GMSC has obtained the routing information needed to forward a mobile terminated short message to the servicing MSC. If an LMSI has been provided in the MAP_SEND_ROUTING_INFO_FOR_SM confirmation, it can be included in the sm-RP-DA information field of the first MAP_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 shall contain the IMSI and the Destination Reference in the MAP_OPEN request shall not be present. The Service Centre address is sent in the parameter SM_RP_OA. The More Messages To Send flag is set to TRUE or FALSE depending on the information received from the Service Centre.

If the GMSC is the servicing MSC then the MAP service is not initiated. The procedure in the Servicing MSC is described in subclause 20.3.1 and in the figure 20.3/4.

If the grouping of MAP_OPEN request and MAP_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. The mapping of the MAP error causes and the RP_ERROR causes is explained in TS GSM 03.40. The appropriate error indication is sent to the SC.

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, 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 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. The gateway MSC may also invoke the procedure when the transfer was successful, if the MNRF and/or MCEF flags were set in the HLR. This procedure is described in detail in subclause 20.5.

Unexpected data value, system failure and unidentified subscriber errors are indicated as a system failure to the SC. Other errors are indicated using appropriate cause values and diagnostic information between the GMSC and the SC as described in TS GSM 03.40 and GSM 04.11.

If there are more short messages to send in the Service Centre and the previous short message transfer succeeded, then the gateway MSC awaits the next short message.

When receiving the next short message from the SC, the gateway MSC sets the More Messages To Send flag according to the information received and starts the service MAP_MT_FORWARD_SHORT_MESSAGE again.

If the gateway MSC is the servicing MSC, then the short message transfer to mobile subscriber is started as described in the subclause 20.3.1.

The mobile terminated short message transfer procedure in the gateway MSC is shown in figure 20.3/7.

Figure 20.3/7 (sheet 1 of 6)

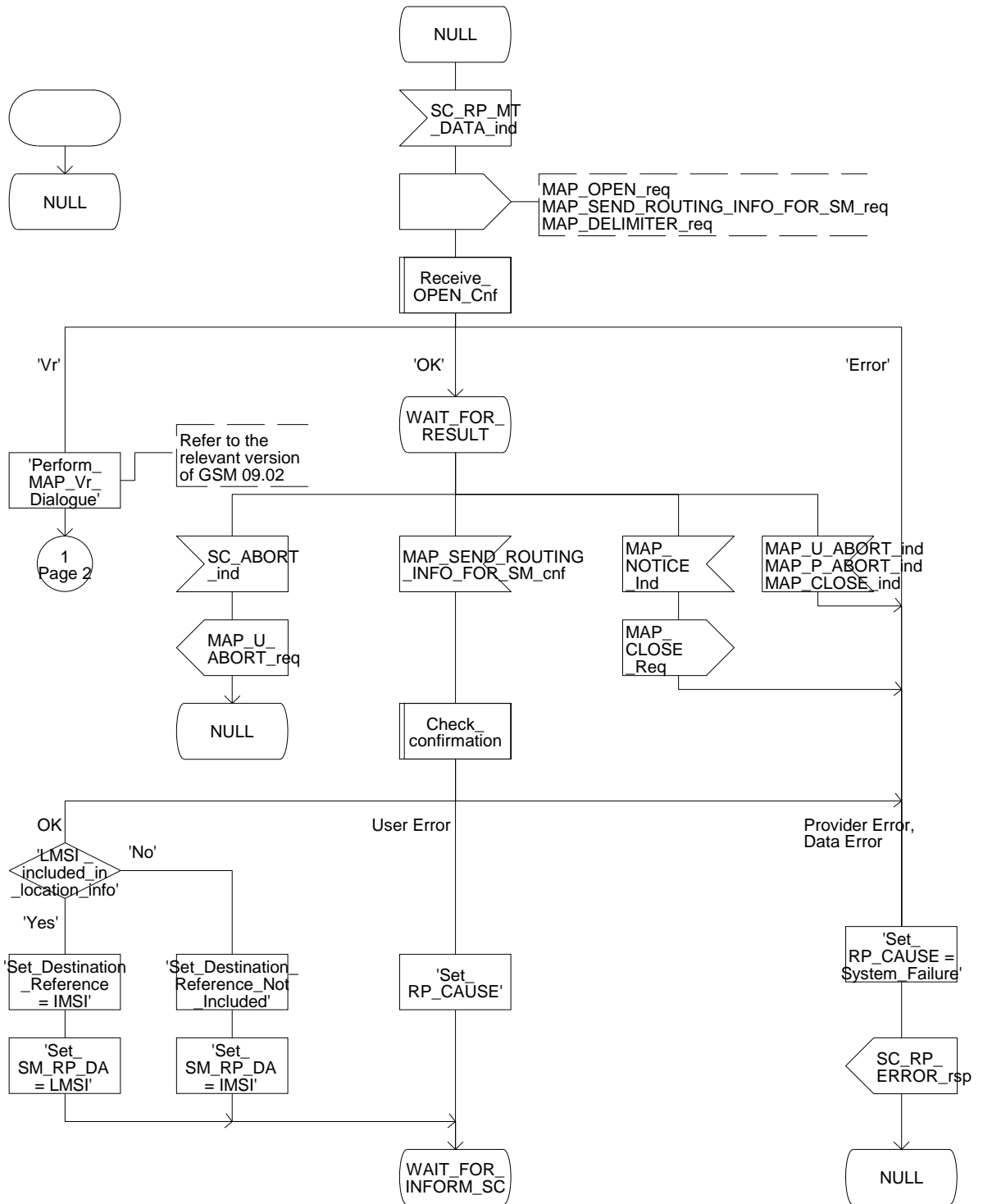


Figure 20.3/7 (sheet 2 of 6)

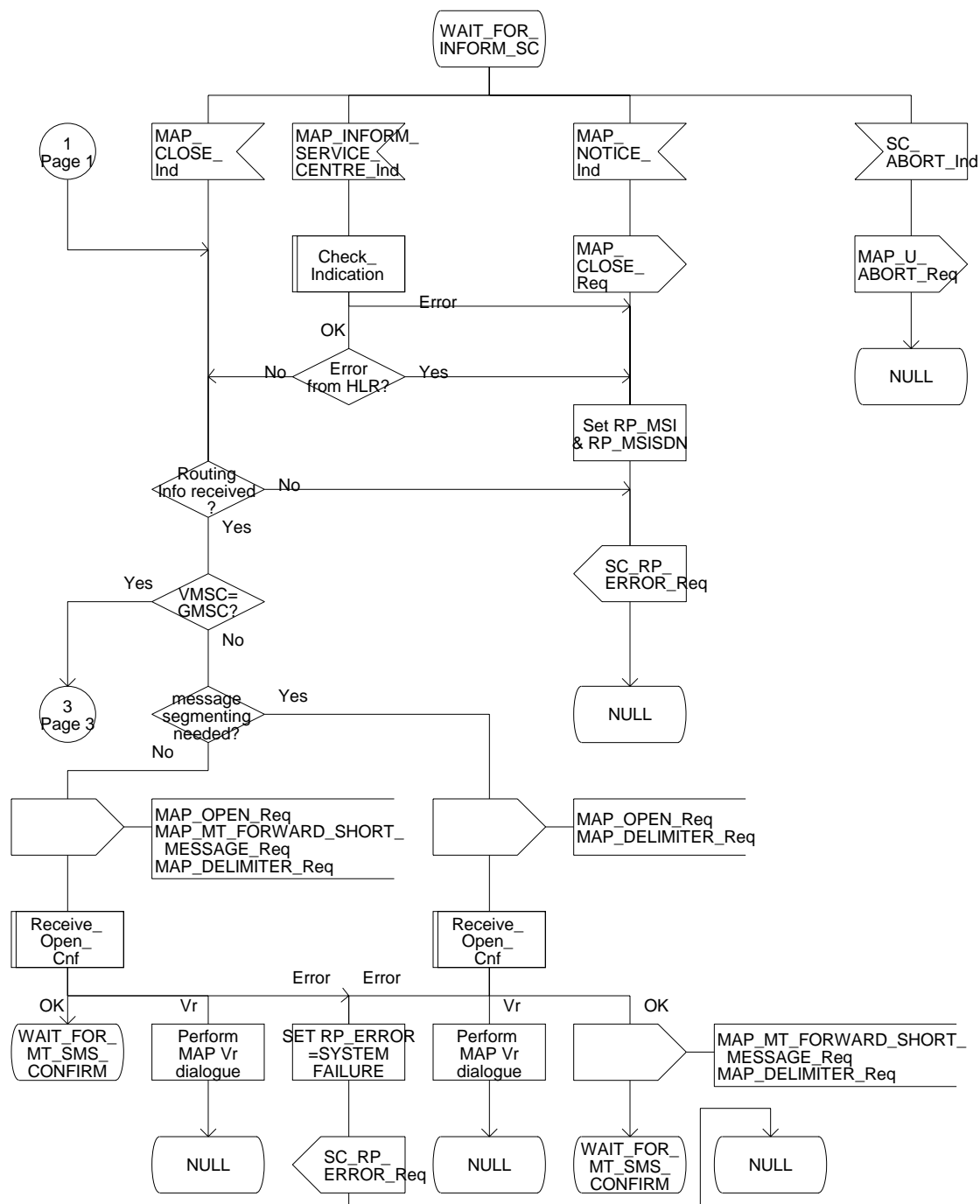


Figure 20.3/7 (sheet 3 of 6)

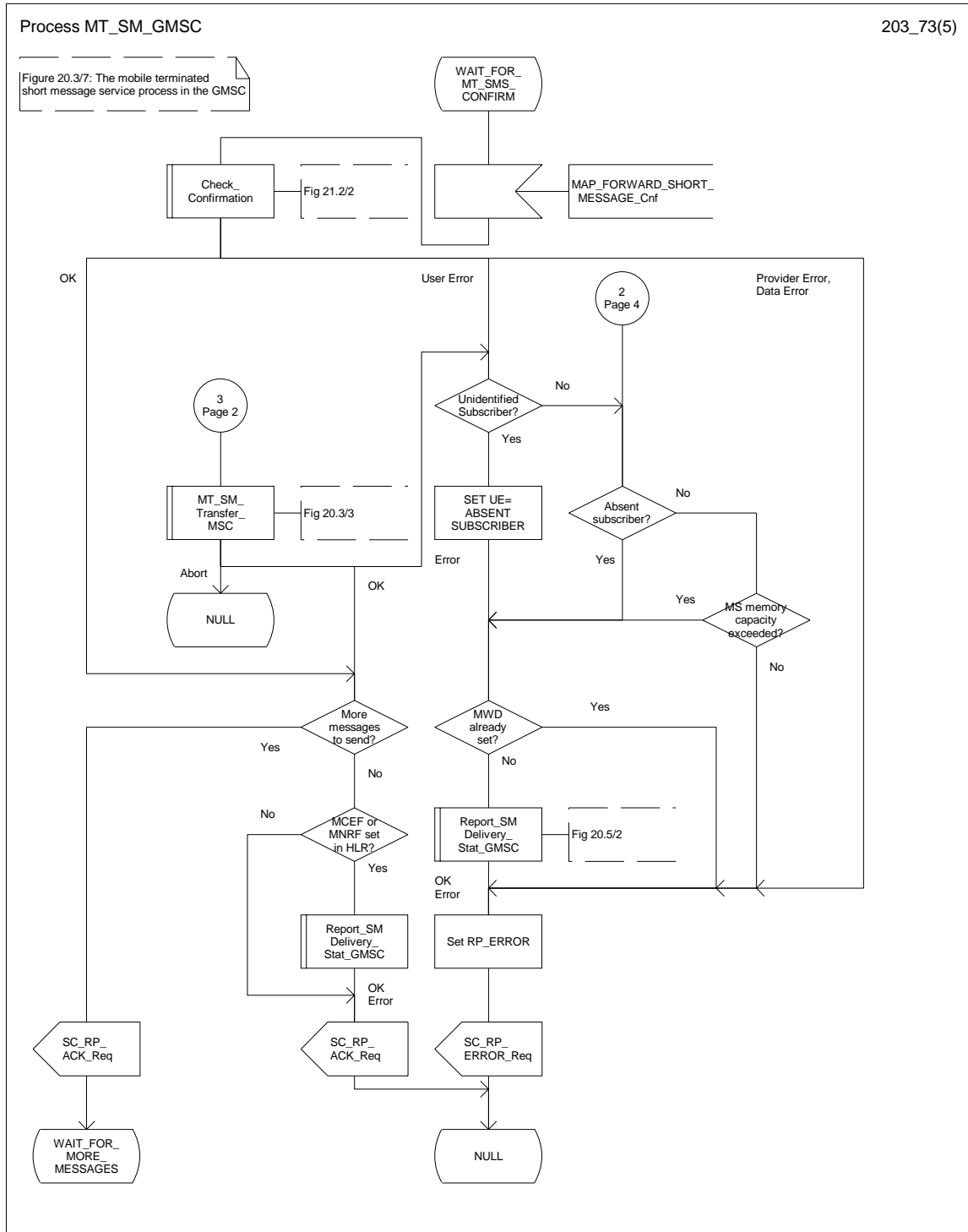


Figure 20.3/7 (sheet 4 of 6)

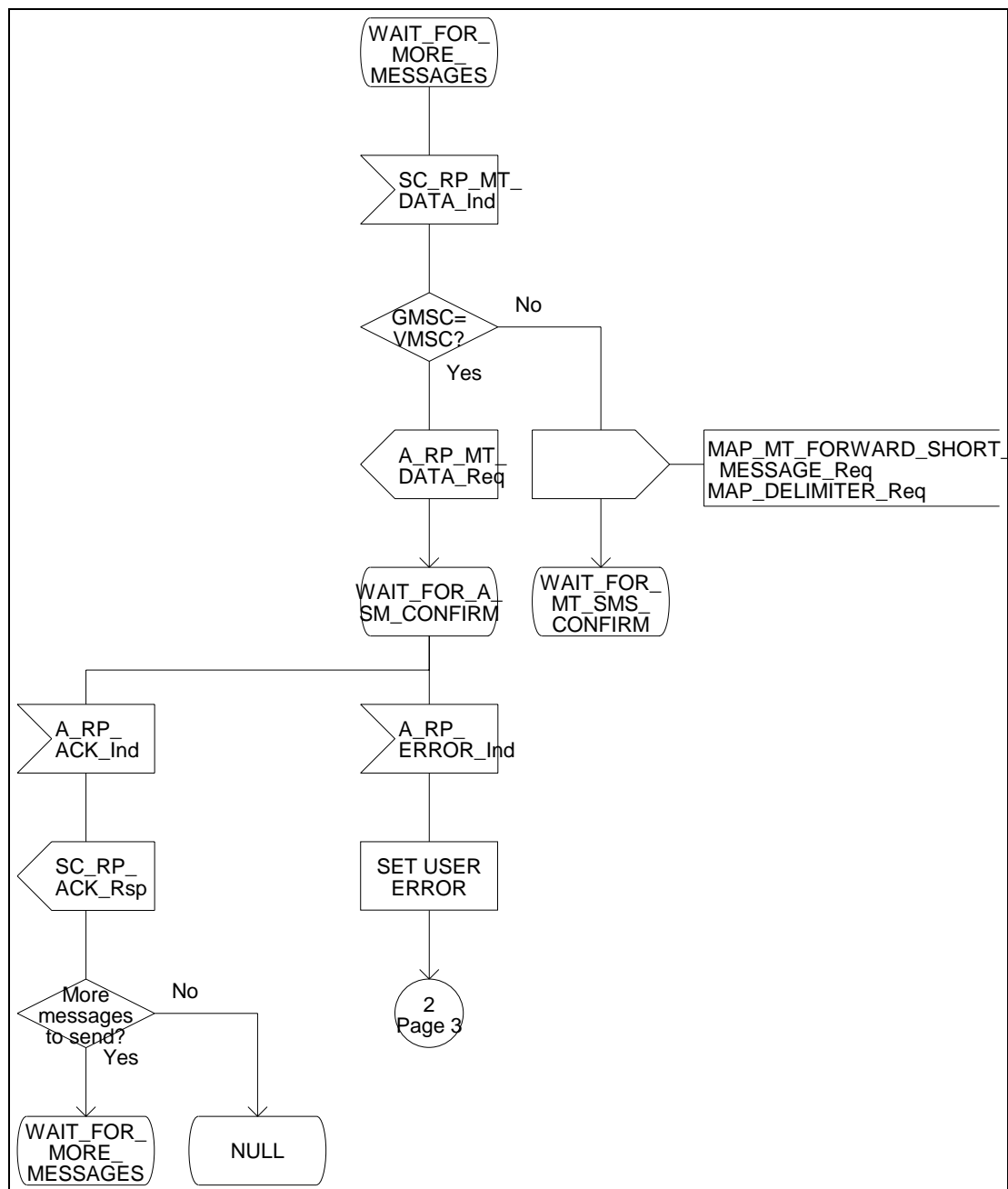


Figure 20.3/7 (sheet 5 of 6)

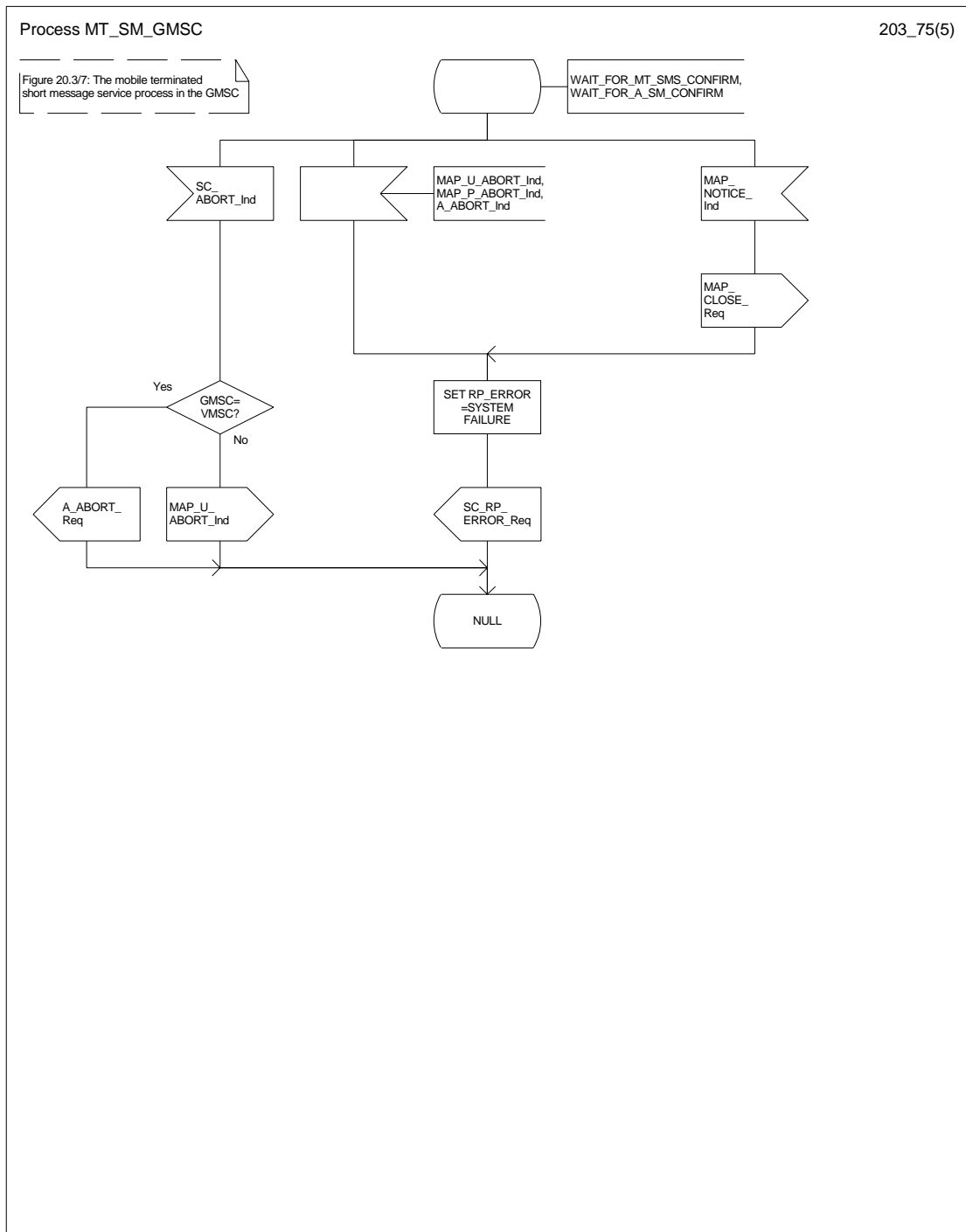


Figure 20.3/7 (sheet 6 of 6)

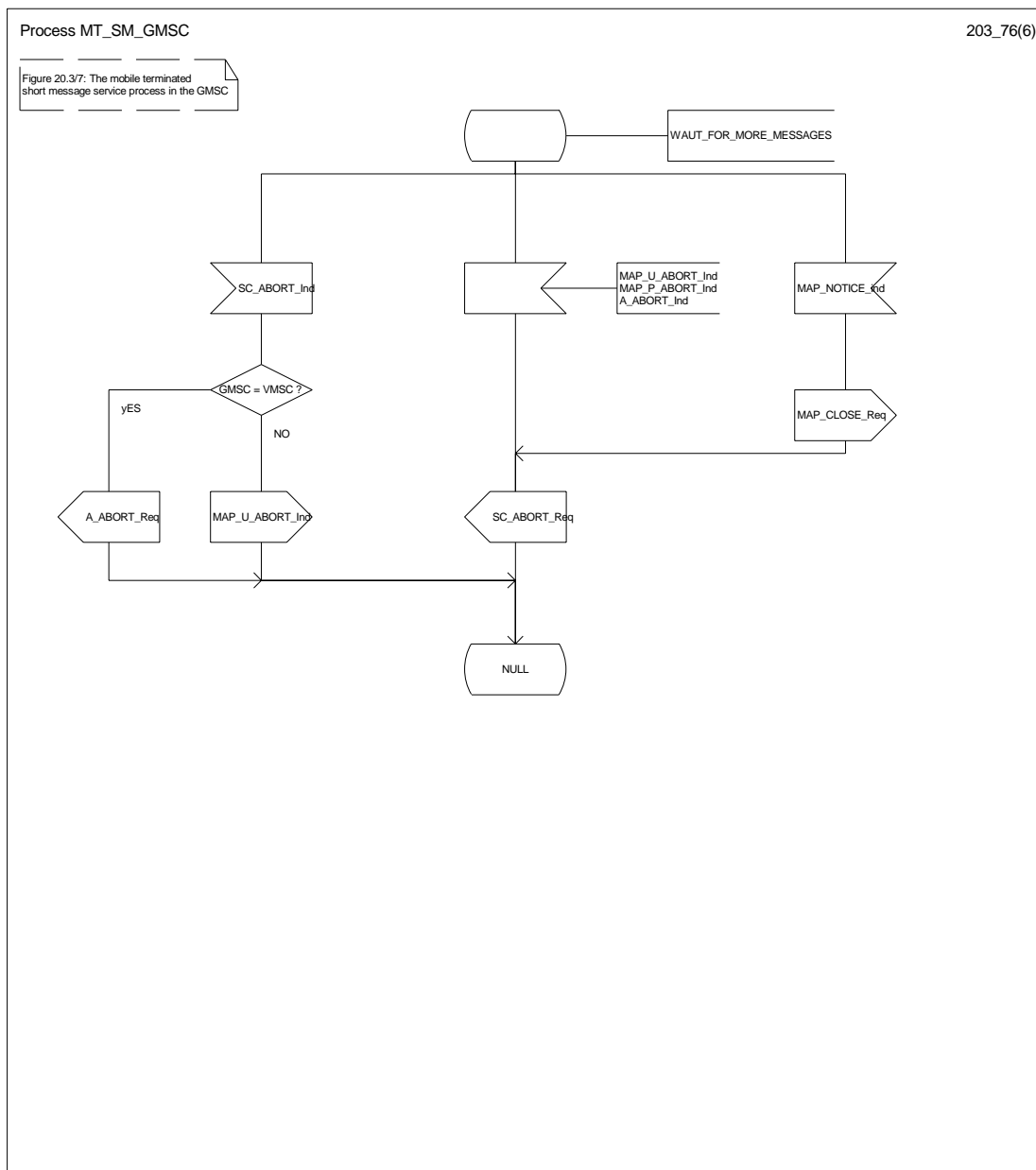
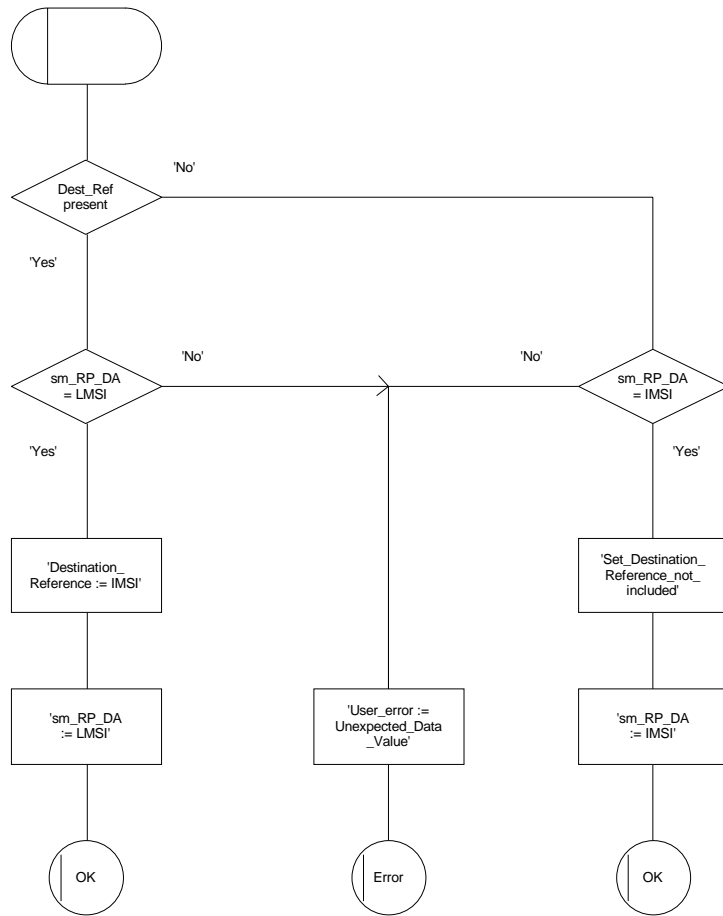


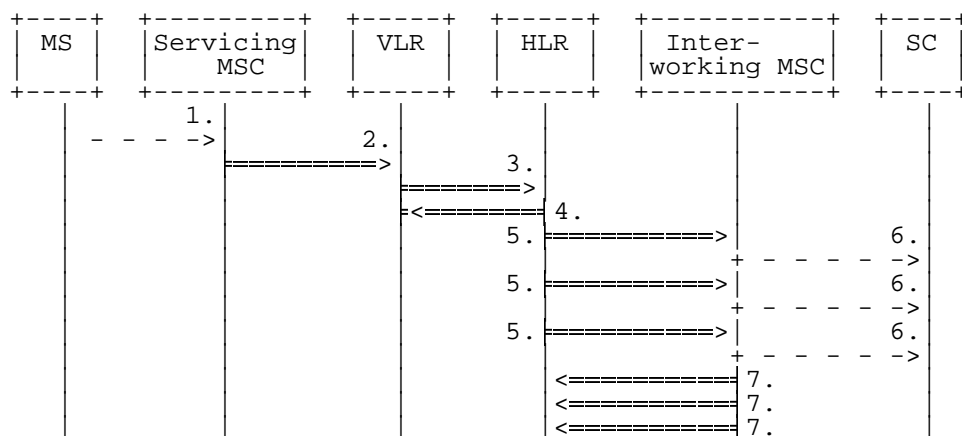
Figure 20.3/8



20.4 The Short Message Alert procedure

The Short Message Alert procedure is used for alerting the Service Centre when the mobile subscriber is active after a short message transfer has failed because the mobile subscriber is not reachable or when the MS has indicated that it has memory capacity to accept a short message.

The Short Message Alert procedure for the case when the mobile subscriber was not reachable is shown in figure 20.4/1.



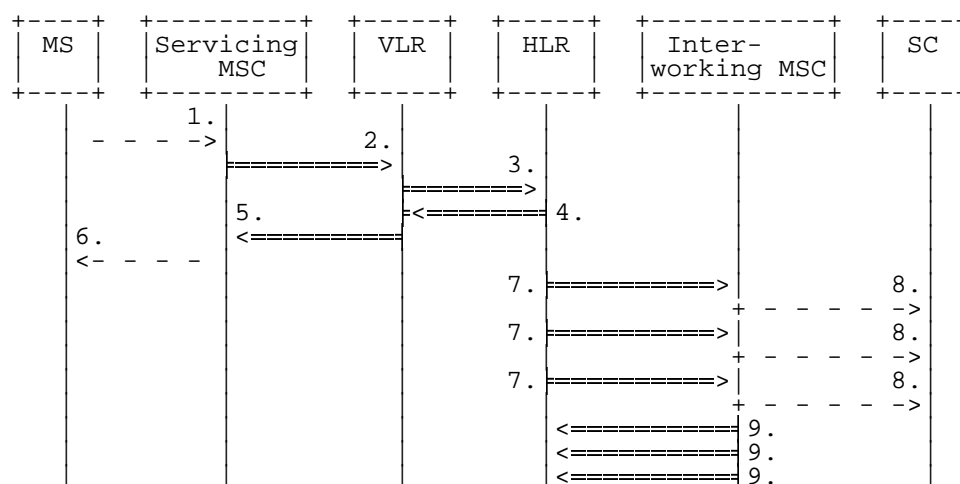
- 1) CM Service Request, Page response or Location Updating (BSSAP; GSM 04.08)
- 2) MAP_PROCESS_ACCESS_REQUEST / MAP_UPDATE_LOCATION_AREA
- 3) MAP_READY_FOR_SM (Mobile Present) / MAP_UPDATE_LOCATION / Supplementary Service Control Request
- 4) MAP_READY_FOR_SM_ACK
- 5) MAP_ALERT_SERVICE_CENTRE (notes 1 and 2)
- 6) Alert Service Centre (GSM 03.40)
- 7) MAP_ALERT_SERVICE_CENTRE_ACK

NOTE 1: To all Service Centres in the Message Waiting List.

NOTE 2: The HLR initiates the MAP_ALERT_SERVICE_CENTRE service only if the MS Memory Capacity Exceeded flag is clear.

Figure 20.4/1: Short message alert procedure (Mobile is present)

The Short Message Alert procedure for the case where the MS indicates that it has memory capacity to accept one or more short messages is shown in figure 20.4/2.



- 1) SM memory capacity available (BSSAP; GSM 04.11)
- 2) MAP_READY_FOR_SM (Memory Available)
- 3) MAP_READY_FOR_SM (Memory Available)
- 4) MAP_READY_FOR_SM_ACK
- 5) MAP_READY_FOR_SM_ACK
- 6) SM memory capacity available (Acknowledge) (BSSAP; GSM 04.11)
- 7) MAP_ALERT_SERVICE_CENTRE (note)
- 8) Alert Service Centre (GSM 03.40)
- 9) MAP_ALERT_SERVICE_CENTRE_ACK

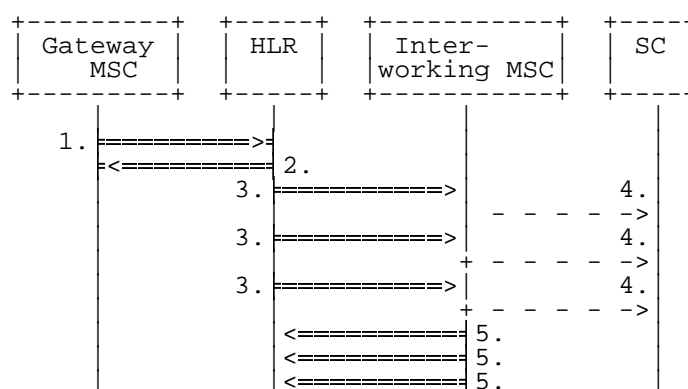
NOTE: To all Service Centres in the Message Waiting List.

Figure 20.4/2: Short message alert procedure (MS memory capacity available)

In addition the following MAP services are used in the MS memory available case:

MAP_PROCESS_ACCESS_REQUEST	(see subclause 6.3);
MAP_AUTHENTICATE	(see subclause 6.5);
MAP_SET_CIPHERING_MODE	(see subclause 6.6);
MAP_PROVIDE_IMSI	(see subclause 6.9);
MAP_CHECK_IMEI	(see subclause 6.7);
MAP_FORWARD_NEW_TMSI	(see subclause 6.9);
MAP_TRACE_SUBSCRIBER_ACTIVITY	(see subclause 7.1).

The Short Message Alert procedure when the MS indicates successful transfer after polling is shown in figure 20.4/3.



- 1) MAP_REPORT_SM_DELIVERY_STATUS (Successful Transfer)
- 2) MAP_REPORT_SM_DELIVERY_STATUS_ACK
- 3) MAP_ALERT_SERVICE_CENTRE (note)
- 4) Alert Service Centre (GSM 03.40)
- 5) MAP_ALERT_SERVICE_CENTRE_ACK

NOTE: To all Service Centres in the Message Waiting List.

Figure 20.4/3: Short message alert procedure (Successful transfer after polling)

20.4.1 Procedures in the Servicing MSC

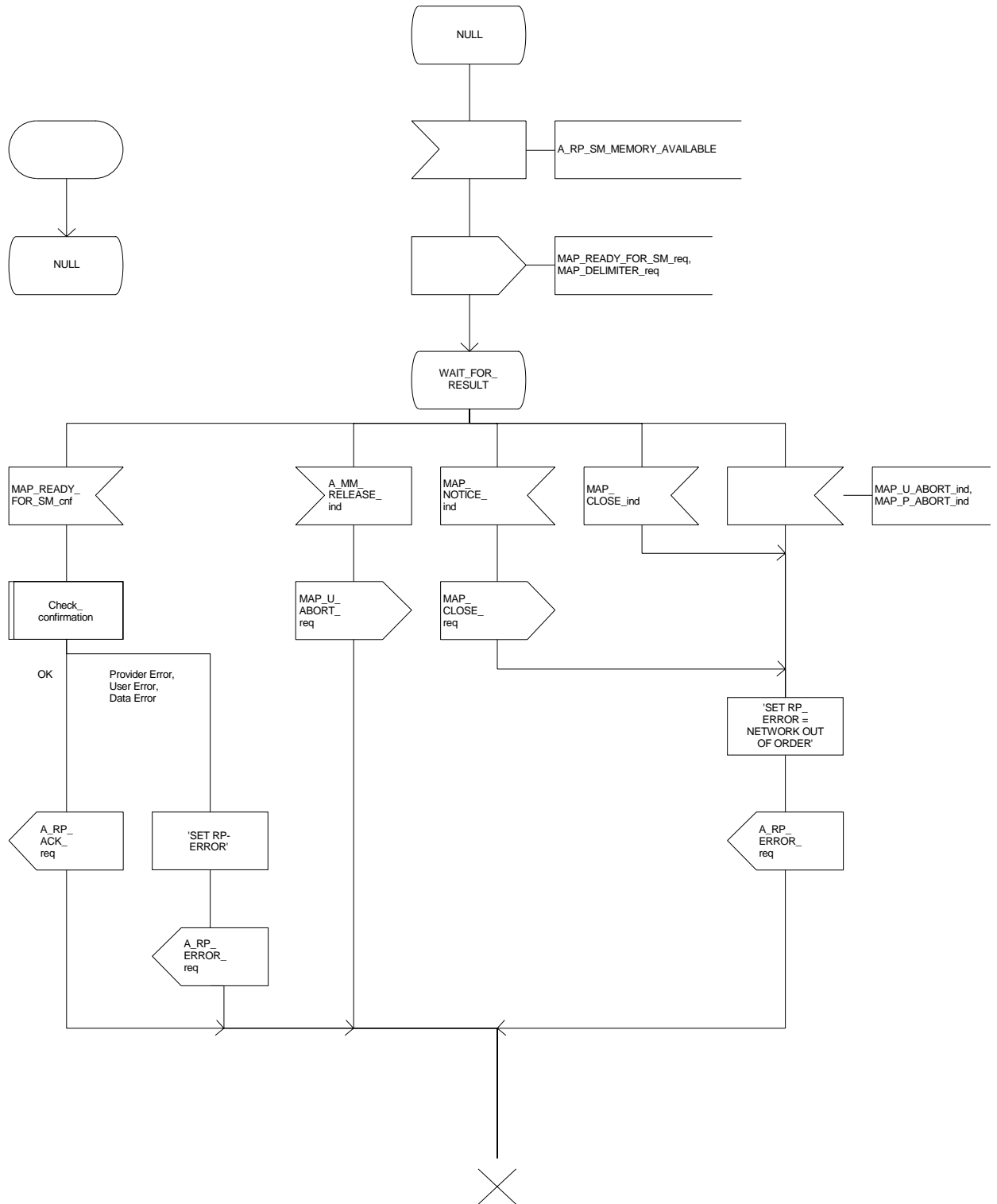
The activation of the MAP_PROCESS_ACCESS_REQUEST service is described in the subclause 20.6.2.

After receiving the SM memory capacity available indication, the servicing MSC sends the MAP_READY_FOR_SM request to the VLR indicating memory available. The outcome of that procedure is one of the following:

- successful acknowledgment. The MSC sends the corresponding message to the MS;
- negative acknowledgment, where the error causes are treated as follows:
 - unexpected data value, data missing and system failure errors are reported as network out of order error to the MS;
 - facility not supported is reported as requested facility not implemented error to the MS;
- procedure failure, which is reported as network out of order error to the MS if a connection to the MS still exists.

The short message alert procedure in the MSC for the MS memory capacity available case is shown in figure 20.4/4.

Figure 20.4/4



20.4.2 Procedures in the VLR

20.4.2.1 The Mobile Subscriber is present

When receiving the MAP_PROCESS_ACCESS_REQUEST indication, MAP_UPDATE_LOCATION_AREA indication while the MS not reachable flag (MNRF) is set, the VLR will send the MAP_READY_FOR_SM request towards the HLR. The Alert Reason is set to indicate that the mobile subscriber is present. If the authentication procedure is initiated and it fails, the VLR will not initiate the service. The process in VLR is described in detail in the subclause 21.10.

20.4.2.2 The Mobile Equipment has memory available

The MAP_PROCESS_ACCESS_REQUEST indication starts the MAP_PROCESS_ACCESS_REQUEST service in the VLR. The application context in the MAP_OPEN indication refers to the short message alerting procedure.

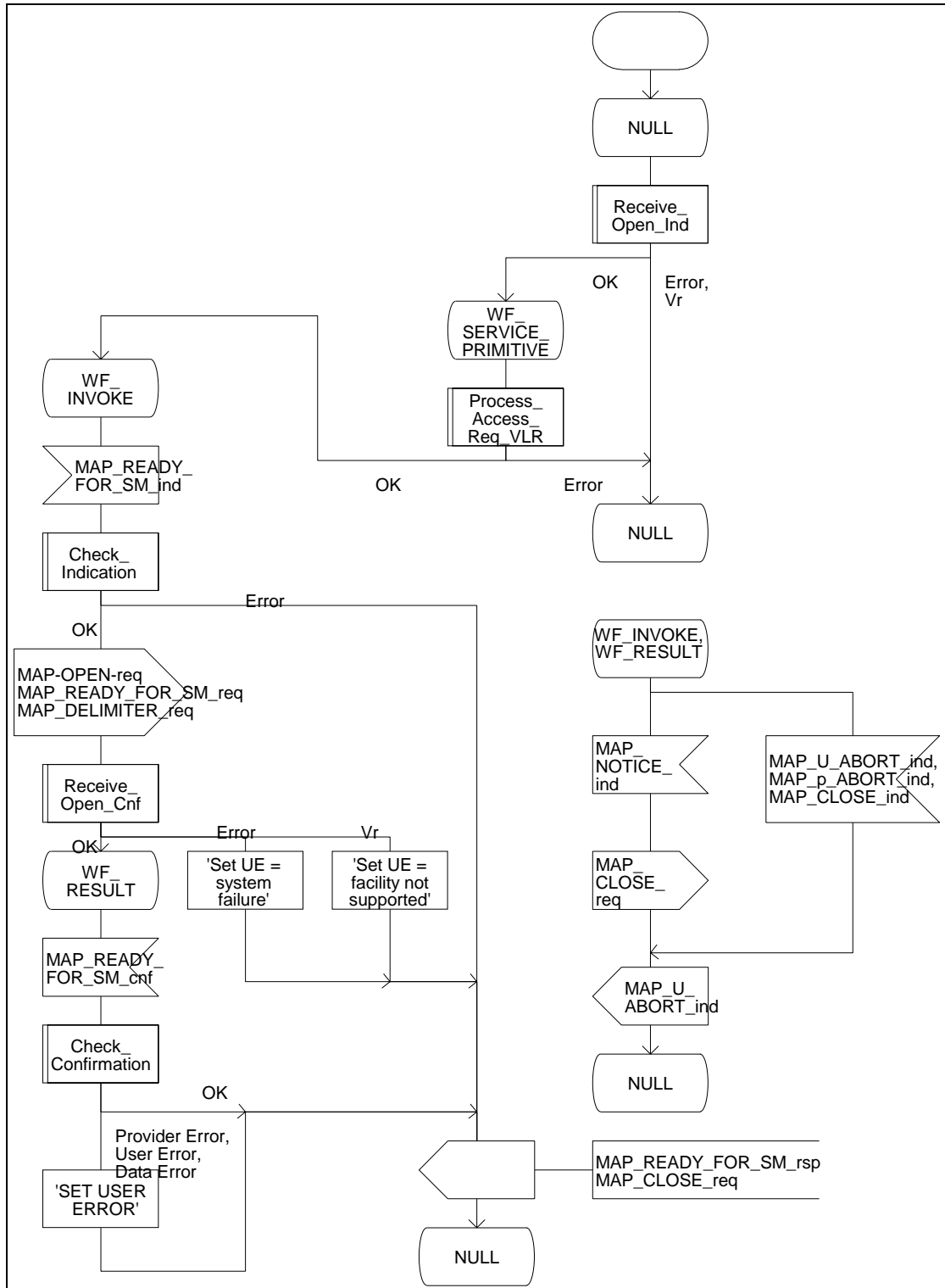
If the service MAP_PROCESS_ACCESS_REQUEST is successful, the VLR waits for the next message from the MSC. When receiving the MAP_READY_FOR_SM indication from the MSC, the VLR will check the contents. Data errors are reported to the MSC as an unexpected data value or data missing error, depending on the error. If the primitive passes the data check, the VLR forwards it to the HLR and awaits an acknowledgment.

When receiving the MAP_READY_FOR_SM confirmation from the HLR and the Alert Reason is MS memory available, the VLR will act as follows:

- the MAP_READY_FOR_SM response is sent to the MSC as follows:
 - an acknowledge in the positive case;
 - system failure error, if unexpected data value, data missing, or unknown subscriber errors are received, otherwise the error cause received from the HLR;
 - a facility not supported error, if the HLR supports MAP Vr only;
 - procedure failure is reported as a system failure error.

The short message alert procedure in the VLR is shown in figures 20.4/5.

Figure 20.4/5



20.4.3 Procedures in the HLR

When receiving the MAP_READY_FOR_SM indication, the HLR will check the contents. Data errors are reported to the VLR as an unexpected data value or a data missing error depending on the error. If the HLR does not support the MNRF, MCEF, and MWD a facility not supported error is reported to the VLR. If the IMSI is unknown an unknown subscriber error is reported to the VLR. Otherwise an acknowledgement is returned to the VLR.

If neither the MS not reachable flag (MNRF) or the memory capacity exceeded flag (MCEF) is set, the HLR sets a timer and waits for it to expire. This ensures that in the race situation the MAP_REPORT_SM_DELIVERY_STATUS service (as described in the subclause 20.6) for the same subscriber can be carried out when delayed in the GMSC.

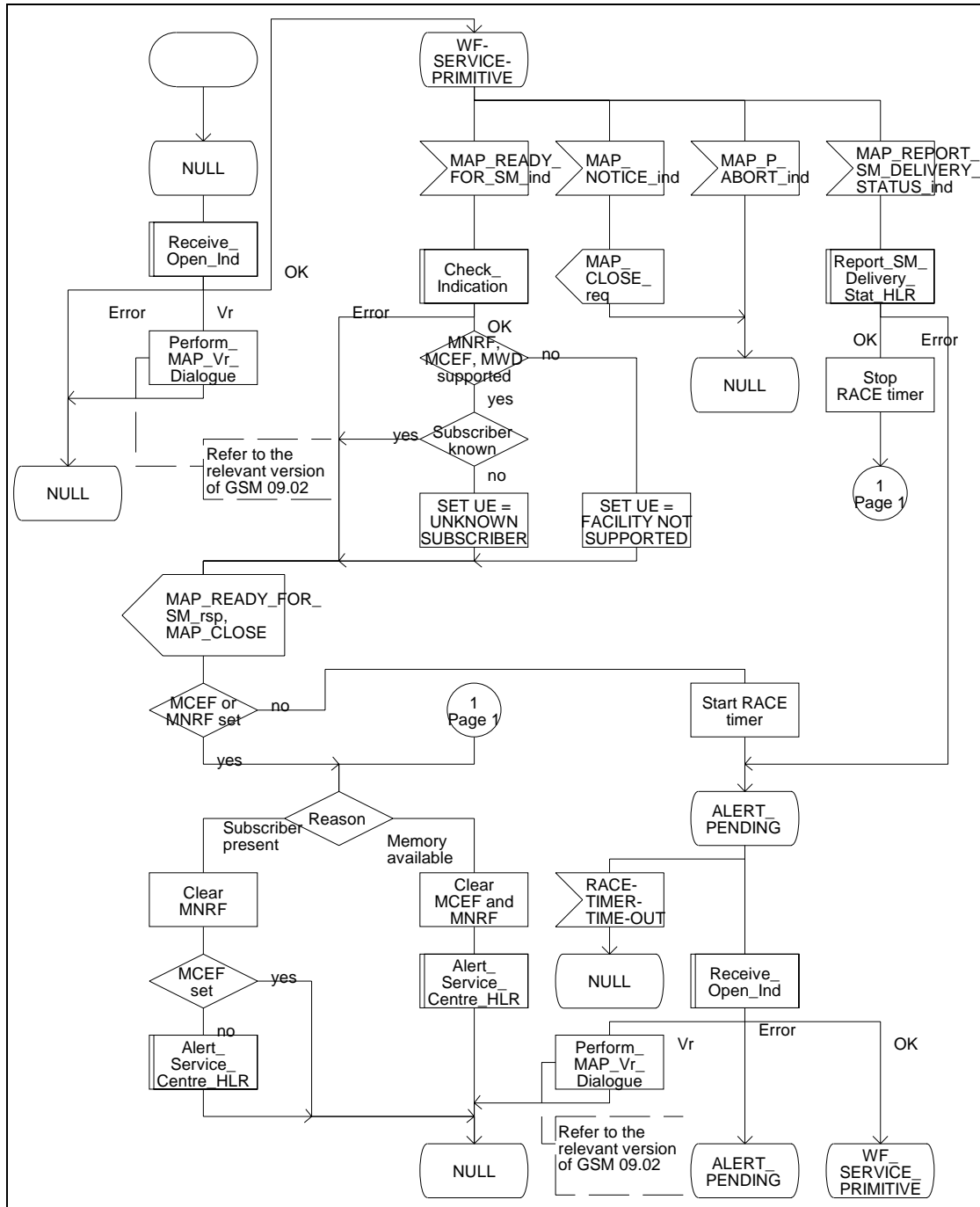
If the Alert Reason indicates the mobile present situation, or when the update location procedure has been successfully completed or Supplementary Service Control request is received, the MS not reachable flag is cleared and the service centre alert procedure is initiated. If the memory capacity exceeded flag is set, the MS not reachable flag is and stored reason for absence are cleared but the alert procedure is not started.

If the Alert Reason indicates the memory available situation, the HLR initiates the alert procedure. The MS not reachable and memory capacity available flags are cleared.

If the MAP_REPORT_SM_DELIVERY_STATUS indication is received and it indicates the successful transfer of the mobile terminated short message, the HLR initiates the alert procedure described in the subclause 21.10 and clears MCEF and MNRF flags and stored reason for absence.

The short message alert procedure in the HLR is shown in figures 20.4/6 and 21.10/2.

Figure 20.4/6



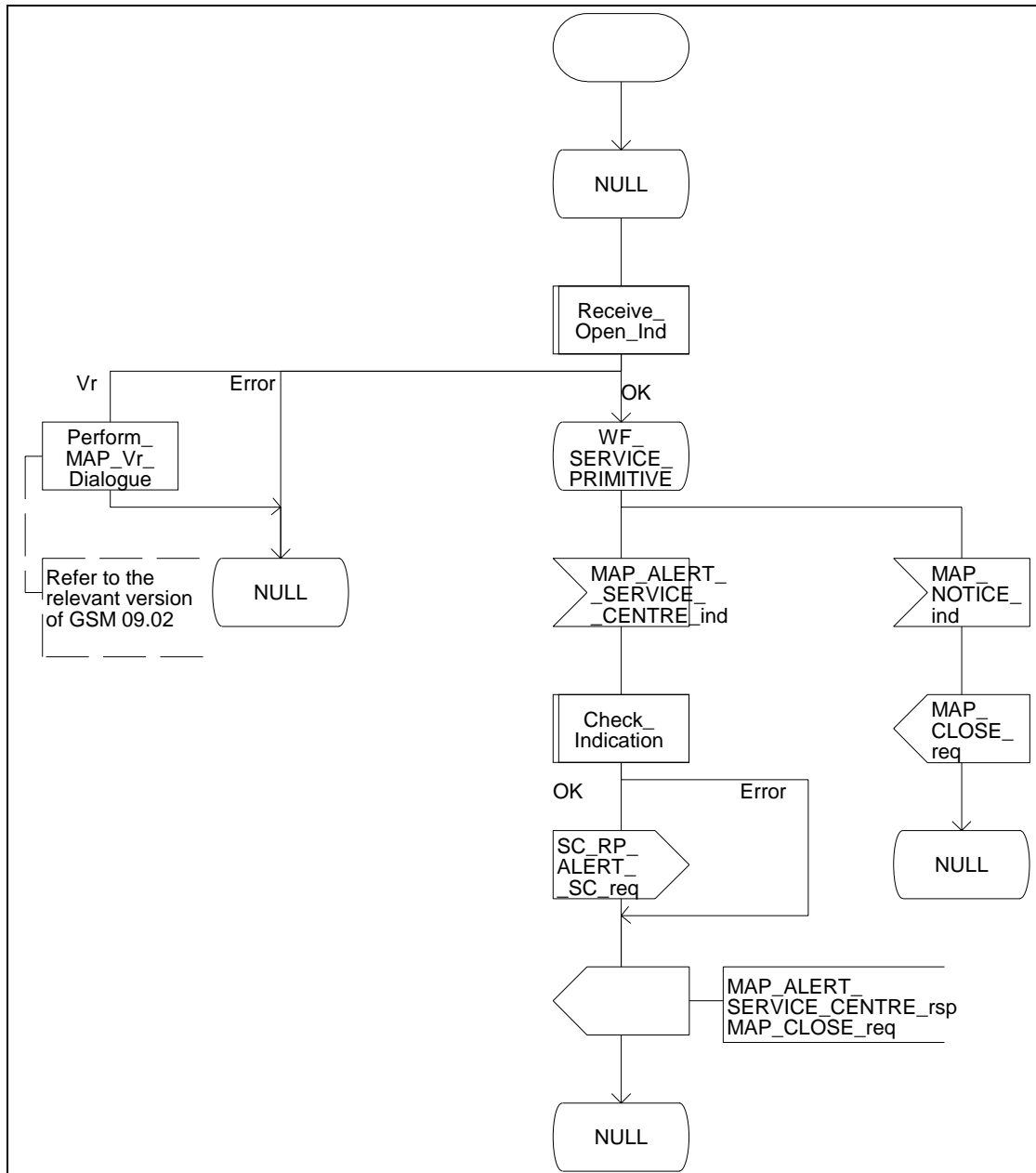
20.4.4 Procedures in the Interworking MSC

When a MAP_ALERT_SERVICE_CENTRE indication is correctly received by the IWMSC, the IWMSC will forward the alerting to the given Service Centre if possible.

Data errors are reported to the HLR as an unexpected data value or a data missing error depending on the error.

The short message alert procedure is shown in figure 20.4/7.

Figure 20.4/7



20.5 The SM delivery status report procedure

The SM delivery status report procedure is used to set the Service Centre address into the message waiting list in the HLR because the subscriber is absent or unidentified or the memory capacity is exceeded. The procedure sets the memory capacity exceeded flag in the HLR if the MS memory does not have room for more messages or the MS not reachable flag in the case of unidentified or absent subscriber.

Additionally the procedure is used to report the HLR about the successful transfer after the Service Centre has polled the subscriber. This procedure is described also in the subclause 20.4.

The SM delivery status report procedure is shown in figure 20.5/1.

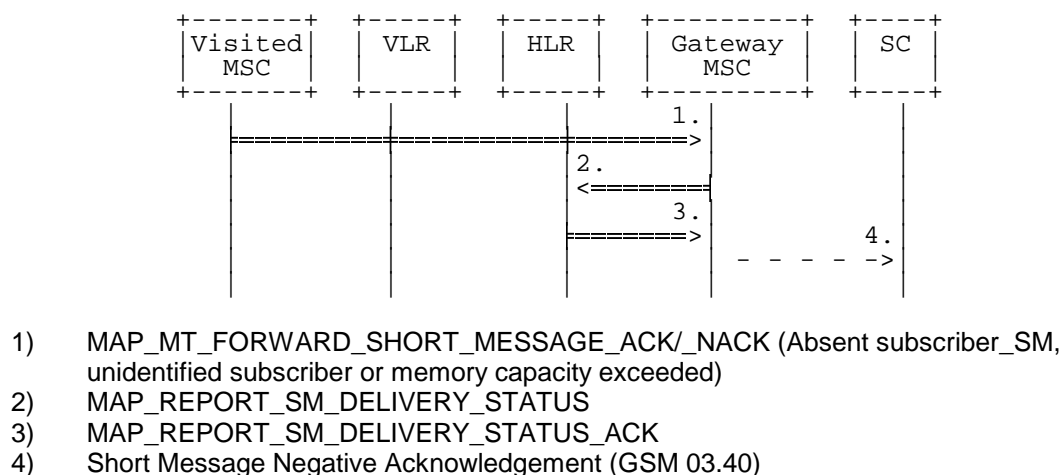


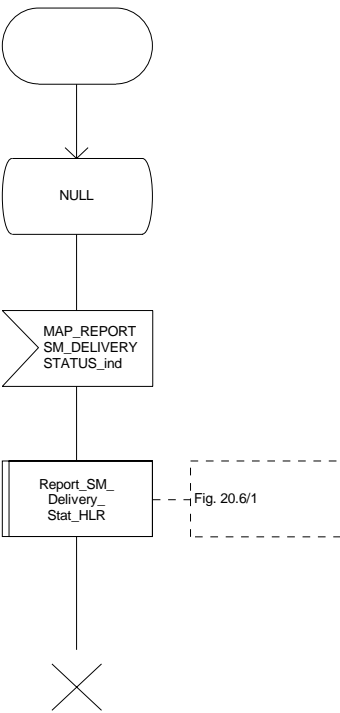
Figure 20.5/1: Short message delivery status report procedure

20.5.1 Procedures in the HLR

When the HLR receives a MAP_REPORT_SM_DELIVERY_STATUS indication, it acts as described in the subclause 20.6, macro Report_SM_Delivery_Stat_HLR.

The short message delivery status report process in the HLR is shown in figure 20.5/2.

Figure 20.5/2



20.5.2 Procedures in the gateway MSC

The GMSC invokes the short message delivery status report procedure if an absent subscriber_SM indication or unidentified subscriber indication or SM delivery failure error indicating MS memory capacity exceeded is received from the servicing MSC during a mobile terminated short message transfer, and the HLR has not indicated that the SC address is included in the MWD. The unidentified subscriber indication is however processed as the absent subscriber_SM indication.

The service is invoked also when the HLR has indicated that either of the flags MCEF or MNRF is set and the SM delivery was successful or, in case of subsequent SM, the last SM delivery was successful.

The reason for unsuccessful or successful delivery of the short message is included in the SM Delivery Outcome in the MAP_REPORT_SM_DELIVERY_STATUS request. 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.

The GMSC sends the MAP_REPORT_SM_DELIVERY_STATUS request to the HLR. As a response the GMSC will receive the MAP_REPORT_SM_DELIVERY_STATUS confirmation reporting:

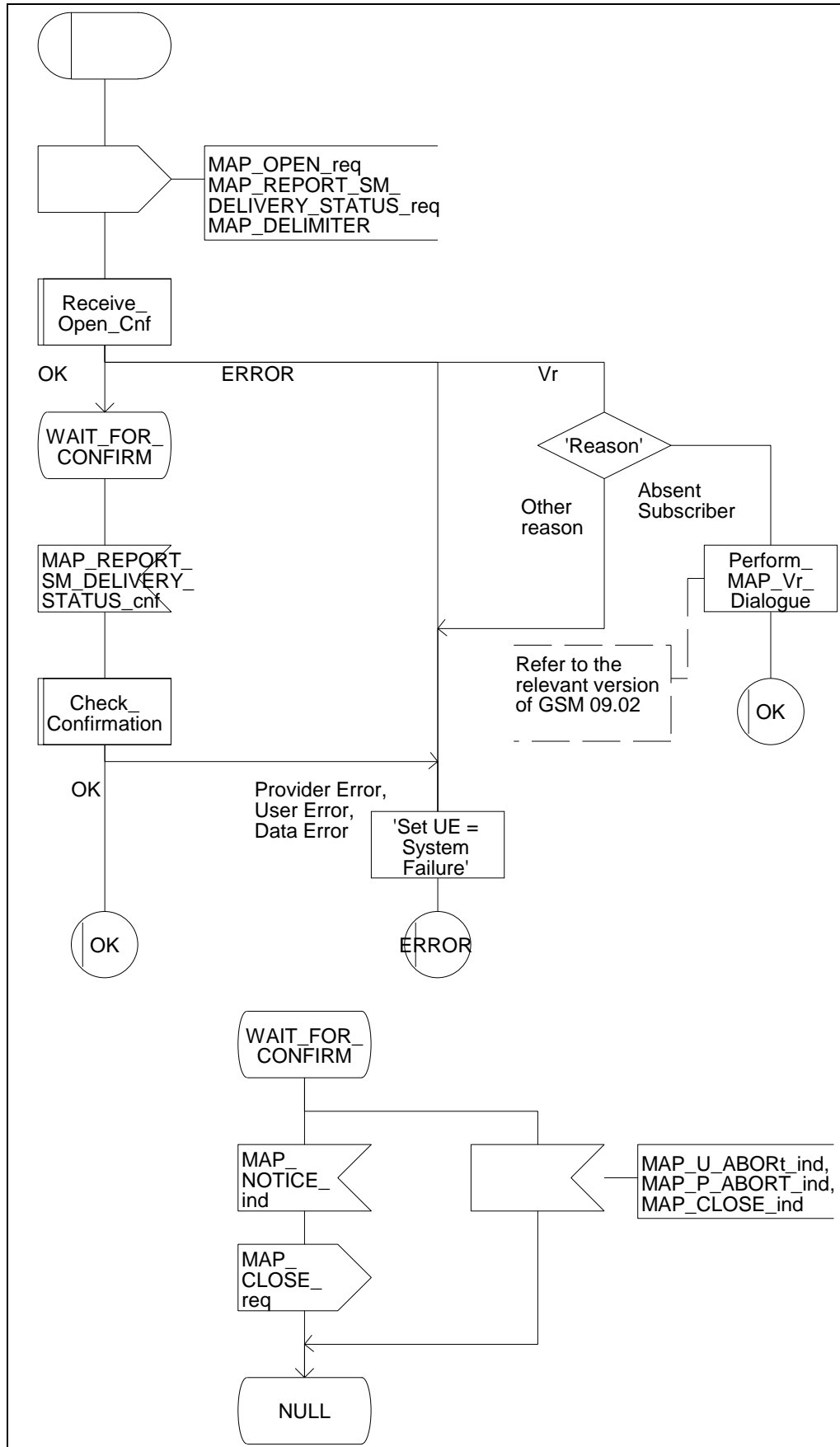
- successful outcome of the procedure. The acknowledge primitive may contain the MSISDN-Alert number which is stored in the MWD List in the HLR;
- unsuccessful outcome of the procedure. The system failure indication is forwarded to the SC.

A provider error is indicated as a system failure to the SC.

The procedure towards the Service Centre may also be aborted. If so the operation towards the HLR is also aborted.

The short message delivery status report procedure in the GMSC is shown in figure 20.5/3.

Figure 20.5/3



20.6 Common procedures for the short message clause

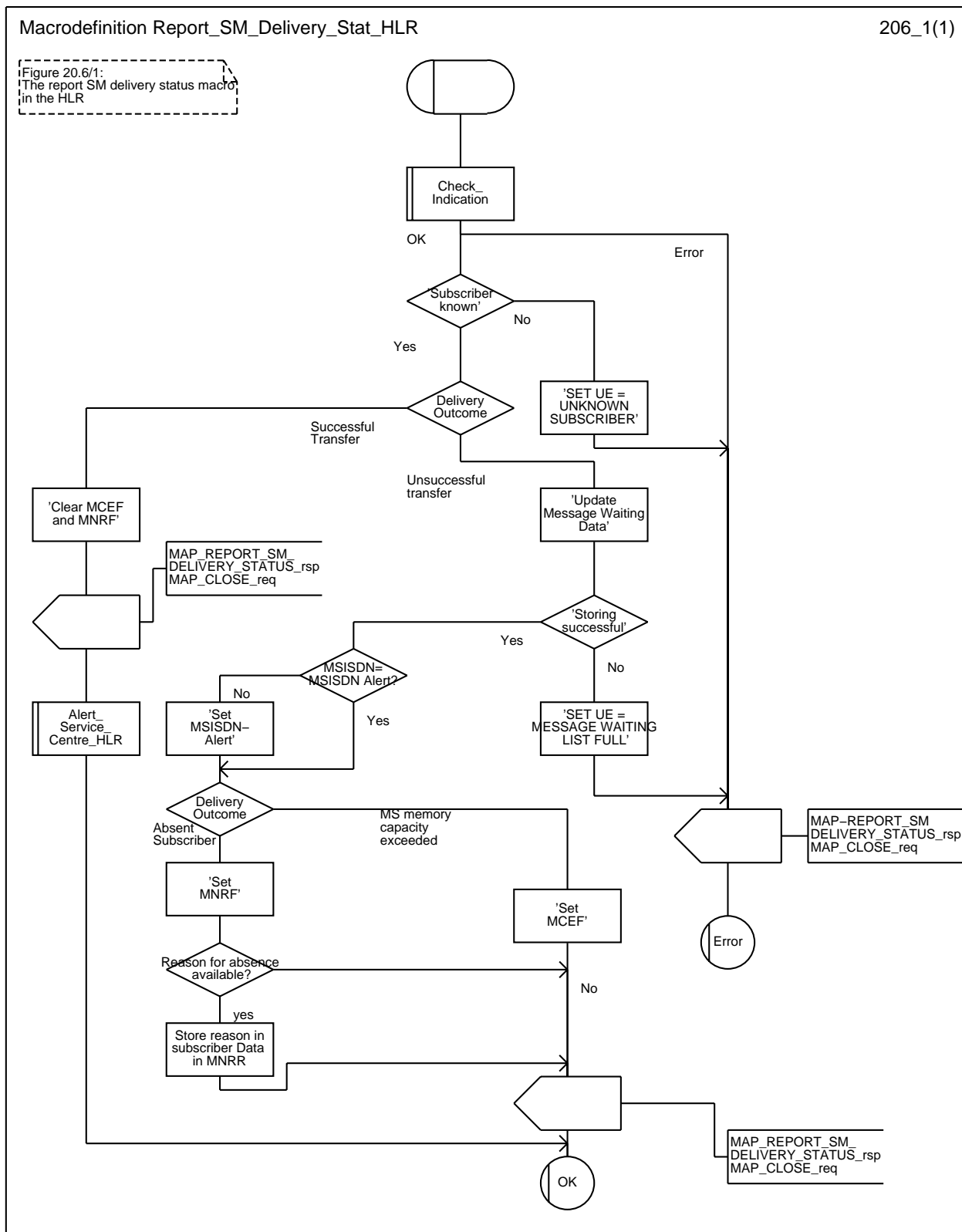
20.6.1 The macro Report_SM_Delivery_Stat_HLR

This macro is used when the HLR receives a MAP_REPORT_SM_DELIVERY_STATUS indication from the GMSC. The HLR responds to the indication as follows:

- if invalid data content is detected, an unexpected data value error or a data missing error is returned to the GMSC;
- if the MSISDN number provided is not recognized by the HLR, an unknown subscriber error is returned to the GMSC;
- if the MAP_REPORT_SM_DELIVERY_STATUS indication reports a successful SM delivery, the Service Centres in the Message Waiting list are alerted as described in the subclause 21.10;
- if the SM Delivery Outcome reports unsuccessful delivery and the inclusion of the SC address in the MWD is not possible, a message waiting list full error is returned to the GMSC;
- if the SM Delivery Outcome reports unsuccessful delivery and the message waiting list is not full, the given Service Centre address is inserted and an acknowledgement is sent to the GMSC. If the MSISDN-Alert stored in the subscriber data is not the same as that received in the MAP_REPORT_SM_DELIVERY_STATUS indication, the MSISDN-Alert is sent in a response primitive to the GMSC;
- if the SM Delivery Outcome is MS memory capacity exceeded the HLR sets the memory capacity exceeded flag in the subscriber data and resets the MRNF;
- if the SM Delivery Outcome is absent subscriber the HLR sets the 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.

The short message delivery status report macro in the HLR is shown in figure 20.6/1.

Figure 20.6/1: Macro Report_SM_Delivery_Stat_HLR



21 General macro description

21.1 MAP open macros

21.1.1 Macro Receive_Open_Ind

This macro is used by a MAP service-user procedure when a peer entity requests opening of a dialogue.

If the application context received in the MAP-OPEN indication primitive indicates a context name of the MAP version one context set, the macro takes the Vr exit..

If an application-context different from version 1 is received, the presence of MAP_OPEN information is checked. If no MAP_OPEN information has been received, the MAP_OPEN response with:

- Result set to Dialogue Accepted; and
- Application Context Name set to the received value,

is returned

If the received version (Vr) is the one described in this version of MAP, the macro takes the OK exit, otherwise it takes the Vr exit..

If MAP_OPEN information is received, the macro "CHECK_REFERENCE" is called in order to check whether the received values for Destination Reference and Originating Reference correspond with the requirements of the received application-context-name. The outcome of this check is an error, the MAP_OPEN response with:

- Result set to Dialogue Refused;
- Refuse Reason set to Invalid Destination Reference or Invalid Originating Reference;
- Application Context Name set to the highest version supported,

is returned and the macro takes the error exit.

If the data values received for Destination Reference and Originating Reference are accepted for the associated application-context-name it is checked whether the Destination Reference is known if this check is required by the process that calls the macro.

If the Destination Reference (e.g. a subscribers IMSI) is unknown, the MAP_OPEN response with

- Result set to Dialogue Refused;
- Refuse Reason set to Invalid Destination Reference;
- Application Context Name set to the highest version supported,

is returned and the macro takes the error exit.

Else, if the Destination Reference is accepted or if no check is required, the MAP_OPEN response with

- Result set to Dialogue Accepted; and
- Application Context Name set to the received value,

is returned and

If the received version (Vr) is the one described in this version of MAP, the macro takes the OK exit, otherwise it takes the Vr exit.

21.1.2 Macro Receive_Open_Cnf

This macro is used by a user procedure after it requested opening of a dialogue towards a peer entity.

On receipt of a MAP_OPEN Confirmation with a "Result" parameter indicating "Dialogue Accepted", the macro takes the OK exit.

If the "Result" parameter indicates "Dialogue Refused", the "Refuse-reason" parameter is examined. If the "Refuse-reason" parameter indicates "Potential Version Incompatibility", the macro terminates in a way that causes restart of the dialogue by using the version 1 protocol.

If the "Refuse-reason" parameter indicates "Application Context Not Supported" and if the received Application Context Name indicates "Version Vr" ($V_r < V_n$), the macro terminates in a way that causes restart of the dialogue by using the version Vr protocol. Otherwise, the macro takes the Error exit.

If the "Refuse-reason" parameter indicates neither "Potential Version Incompatibility" nor "Application Context Not Supported", the macro takes the Error exit.

If a MAP_U_ABORT, a MAP_P_ABORT or a MAP_NOTICE Indication is received, the macro takes the Error exit.

Figure 21.1/1

Figure 21.1/2

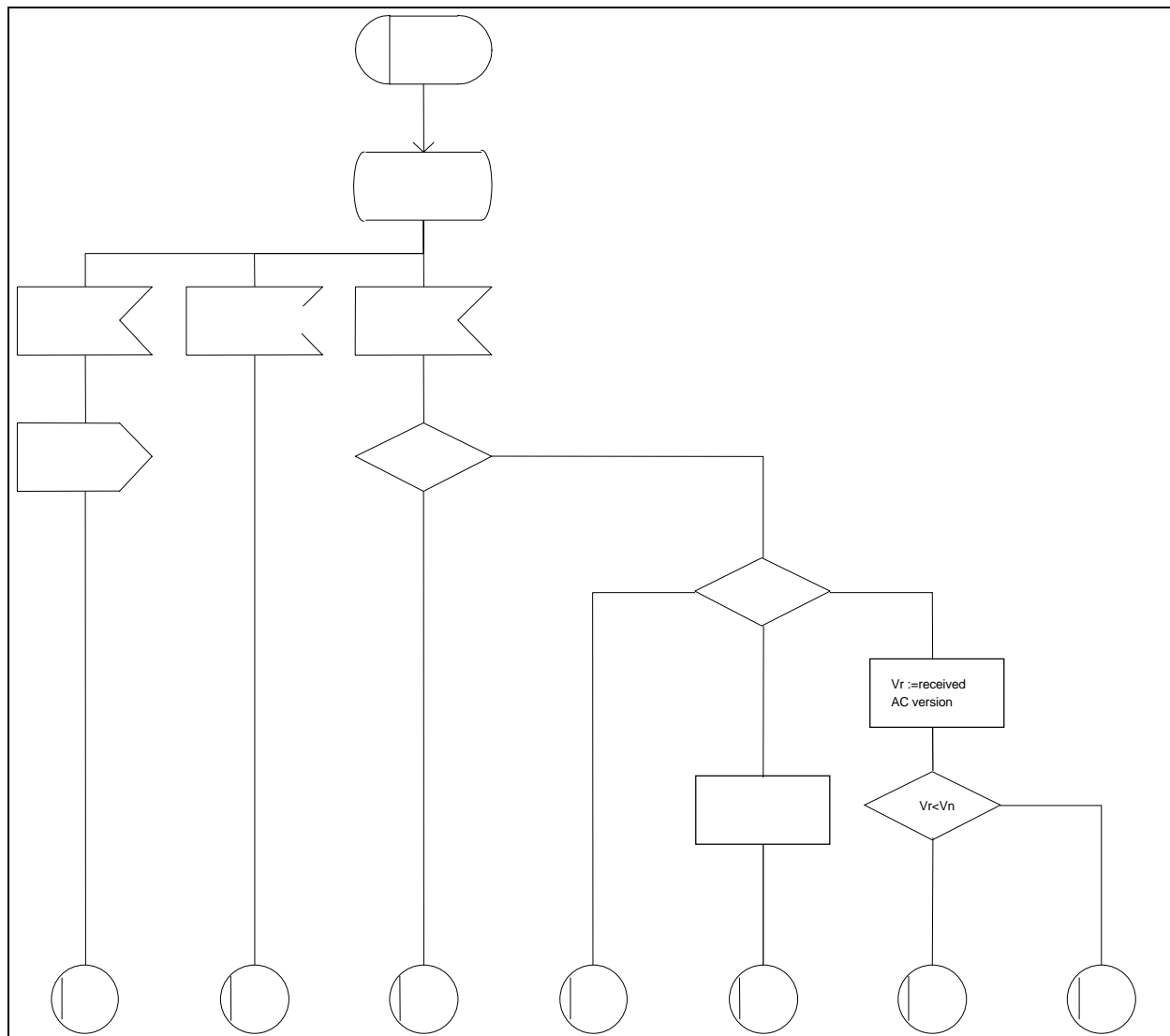
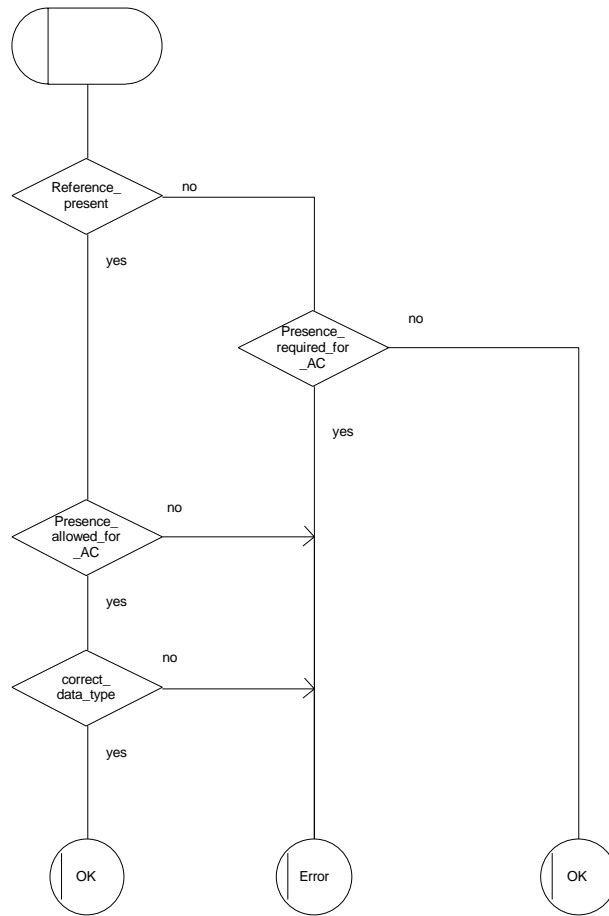


Figure 21.1/3



21.2 Macros to check the content of indication and confirmation primitives

21.2.1 Macro Check_Indication

If a parameter required by the application is missing from the indication, the macro takes the error exit, with a user error of "Data Missing".

If a parameter not expected by the application is present in the indication, or an expected parameter has a value not in the set of values permitted by the application, the macro takes the error exit, with a user error of "Unexpected Data Value".

Otherwise the macro takes the "OK" exit.

The macro is shown in figure 21.2/1.

21.2.2 Macro Check_Confirmation

If the confirmation contains a provider error the macro issues a MAP CLOSE request and takes the provider error exit.

Otherwise, if the confirmation contains a user error the macro takes the user error exit.

Otherwise, if a parameter required by the application is missing from the confirmation, or a parameter not expected by the application is present in the confirmation, or an expected parameter has a value not in the set of values permitted by the application, the macro takes the data error exit.

Otherwise the macro takes the "OK" exit.

The macro is shown in figure 21.2/2.

Figure 21.2/1

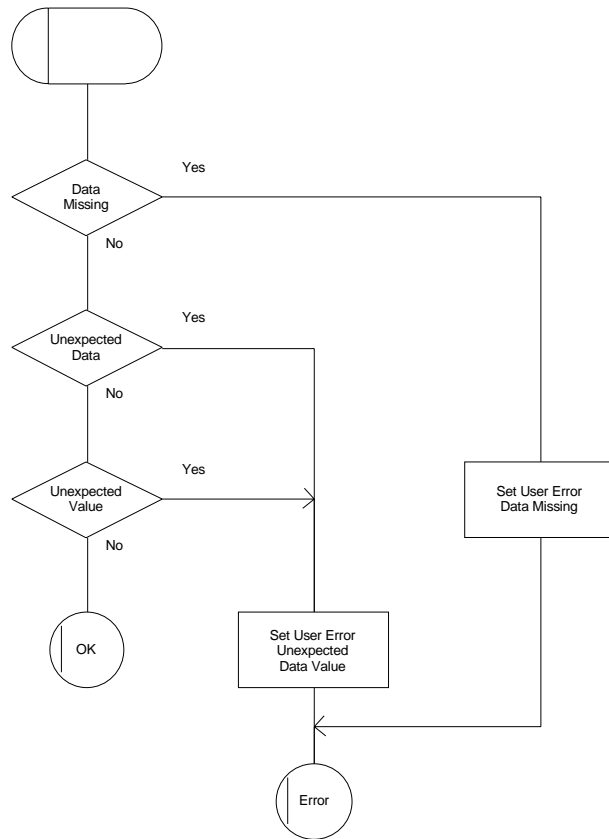
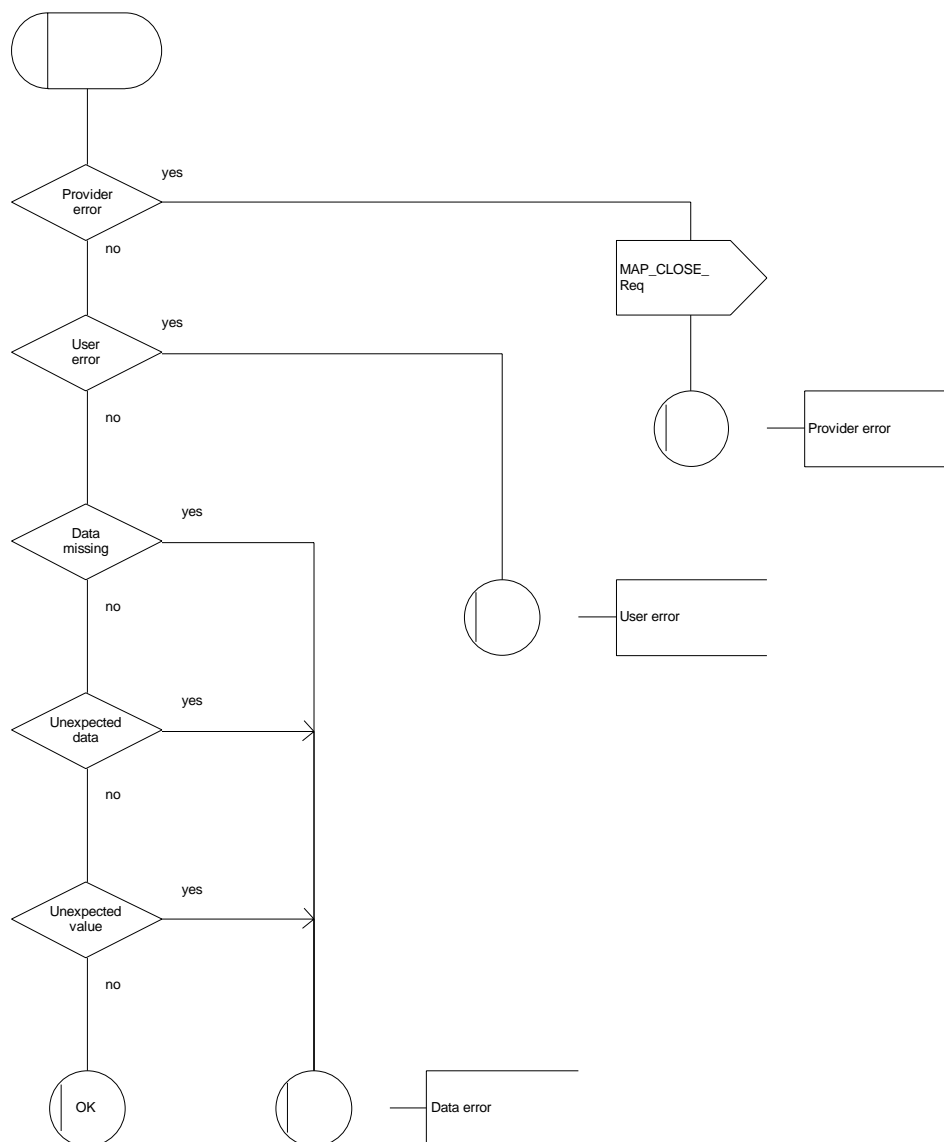


Figure 21.2/2



21.3 The page and search macros

21.3.1 Macro PAGE_MSC

This macro (see figure 21.3/1) is called if a mobile terminating call set-up, an unstructured SS notification, a network-initiated unstructured SS request or a mobile terminating short message is to be delivered to the MS and the current location area identity of the MS is known in the VLR.

When the MSC receives a MAP_PAGE indication, parameter checks are performed first (macro Check_Indication, see subclause 21.2). If parameter errors are detected, the MSC returns a MAP_PAGE response containing the appropriate error cause and the macro terminates with unsuccessful outcome.

Thereafter, several checks on the indication content are performed. The macro terminates by returning the MAP_PAGE response with error:

Unknown Location Area if the LAI is not known in the MSC;

System Failure if the call has been released by the calling subscriber or the SMS or SS transaction for this subscriber has been released by the originating entity in the meantime.

Next, the MSC checks if an MM-connection over the radio link already exists for the given IMSI. If so,

- in the case of mobile terminating call set-up the MSC determines whether the busy condition can be established (see GSM 02.01 for a definition of busy states). If the MSC determines that the MS is busy, it returns a MAP_PAGE response with error Busy Subscriber, qualified by either More Calls Allowed or No More Calls Allowed. The macro then terminates with unsuccessful outcome.
- if the service requested is short message service or an unstructured SS notification or network-initiated unstructured SS request, or if the service is mobile terminating call set-up, but the existing connection is for signalling purposes only (i.e. a service different from call set-up), the access connection status is set according to the characteristics of the existing connection (i.e. RR-connection established, ciphering mode on/off, MM-connection existing and authenticated or not), and the macro terminates with successful outcome.

If no MM-connection for the given IMSI exists, paging is initiated at the radio interface within all cells of the location area indicated by the VLR. If the VLR provided the TMSI, the MSC uses it to identify the MS at the radio interface; otherwise the MSC uses the IMSI. The IMSI will also be used to determine the page group (see GSM 04.08). There are several possible outcomes of paging:

- the MS responds to paging, causing the access connection status to be set accordingly (i.e. no RR-connection, in which case other values are not significant), and the macro terminates with successful outcome;
- the MS responds with a channel request containing an establishment cause which is not "answer to paging". The MSC sends a MAP_PAGE response primitive with user error Busy Subscriber before the macro terminates with unsuccessful outcome. This will give priority to the mobile originating request. Alternatively, as an implementation option, the MSC may treat this as a response to paging, which will give priority to the mobile terminating request.
- there is no response from the MS. The MSC sends a MAP_PAGE response primitive with user error Absent Subscriber before the macro terminates with unsuccessful outcome;
- the call handling connection or MAP transaction on which the call, SMS or unstructured SS transaction is waiting for delivery, is released before a response is received from the MS (indicated in the SDL by the input signal I-REL). The MAP transaction with the VLR will be released in this case by a MAP_U_ABORT request, and the unsuccessful macro termination will indicate transaction termination.

- the MAP transaction with the VLR may be released by receiving a MAP_U_ABORT or MAP_P_ABORT indication. The call handling connection or MAP transaction on which the call, SMS or unstructured SS transaction is waiting for delivery, is released (indicated in the SDL by the output signal I-REL), and the unsuccessful macro termination will indicate transaction termination.

21.3.2 Macro Search_For_MS_MSC

This macro (see figure 21.3/2) is called if a mobile terminating call set-up, an unstructured SS notification, a network-initiated unstructured SS request or a mobile terminating short message is to be delivered to the MS and the current location area identity of the MS is not known in VLR.

When the MSC receives a MAP_SEARCH_FOR_MS Indication, parameter checks are performed first (macro Check_indication, see subclause 21.2). If parameter errors are detected, the MSC returns a MAP_SEARCH_FOR_MS response containing the appropriate error cause and the macro terminates with unsuccessful outcome.

Thereafter, the MSC checks whether the call or the SMS or SS transaction still exists in the MSC. If the call or the SMS or SS transaction has been released, the MSC returns a MAP_SEARCH_FOR_MS response with error System Failure and the macro terminates with unsuccessful outcome.

Next, the MSC checks if an MM-connection over the radio link already exists for the given IMSI. If so,

- in the case of mobile terminating call set-up the MSC determines whether the busy condition can be established (see GSM 02.01 for a definition of busy states). If the MSC determines that the MS is busy, it returns a MAP_SEARCH_FOR_MS response with error Busy Subscriber, qualified by either More Calls Allowed or No More Calls Allowed. The macro then terminates with unsuccessful outcome.
- if the service requested is short message service or an unstructured SS notification or network-initiated unstructured SS request, or if the service is mobile terminating call set-up, but the existing connection is for signalling purposes only (i.e. a service different from call set-up), a MAP_SEARCH_FOR_MS response containing the IMSI and current location area identification of the called MS is returned to the VLR. The access connection status is set according to the characteristics of the existing connection (i.e. RR-connection established, ciphering mode on/off, MM-connection existing and authenticated or not), and the macro terminates with successful outcome.

If no MM-connection for the given IMSI exists, paging is initiated at the radio interface within all cells of all location areas of the VLR, using the IMSI to identify the subscriber and the page group (see GSM 04.08). There are several possible outcomes of paging:

- the MS responds to paging, causing a MAP_SEARCH_FOR_MS response containing the IMSI and current location area identification of the called MS to be returned to the VLR. The access connection status will be set accordingly (i.e. no RR-connection, in which case other values are not significant), and the macro terminates with successful outcome.
- the MS responds with a channel request containing an establishment cause which is not "answer to paging". The MSC sends a MAP_SEARCH_FOR_MS response primitive with user error "Busy Subscriber" before the macro terminates with unsuccessful outcome. This will give priority to the mobile originating request. Alternatively, as an implementation option, the MSC may treat this as a response to paging, which will give priority to the mobile terminating request.
- there is no response from the MS. The MSC sends a MAP_SEARCH_FOR_MS response primitive with user error "Absent Subscriber" before the macro terminates with unsuccessful outcome.

- the call handling connection or MAP transaction on which the call, SMS or unstructured SS transaction is waiting for delivery, is released before a response is received from the MS (indicated in the SDL by the input signal I-REL). The MAP transaction with the VLR will be released in this case by a MAP_U_ABORT request, and the unsuccessful macro termination will indicate transaction termination.
- the MAP transaction with the VLR may be released by receiving a MAP_U_ABORT or MAP_P_ABORT indication. The call handling connection or MAP transaction on which the call, SMS or unstructured SS transaction is waiting for delivery, is released (indicated in the SDL by the output signal I-REL), and the unsuccessful macro termination will indicate transaction termination.

Figure 21.3/1

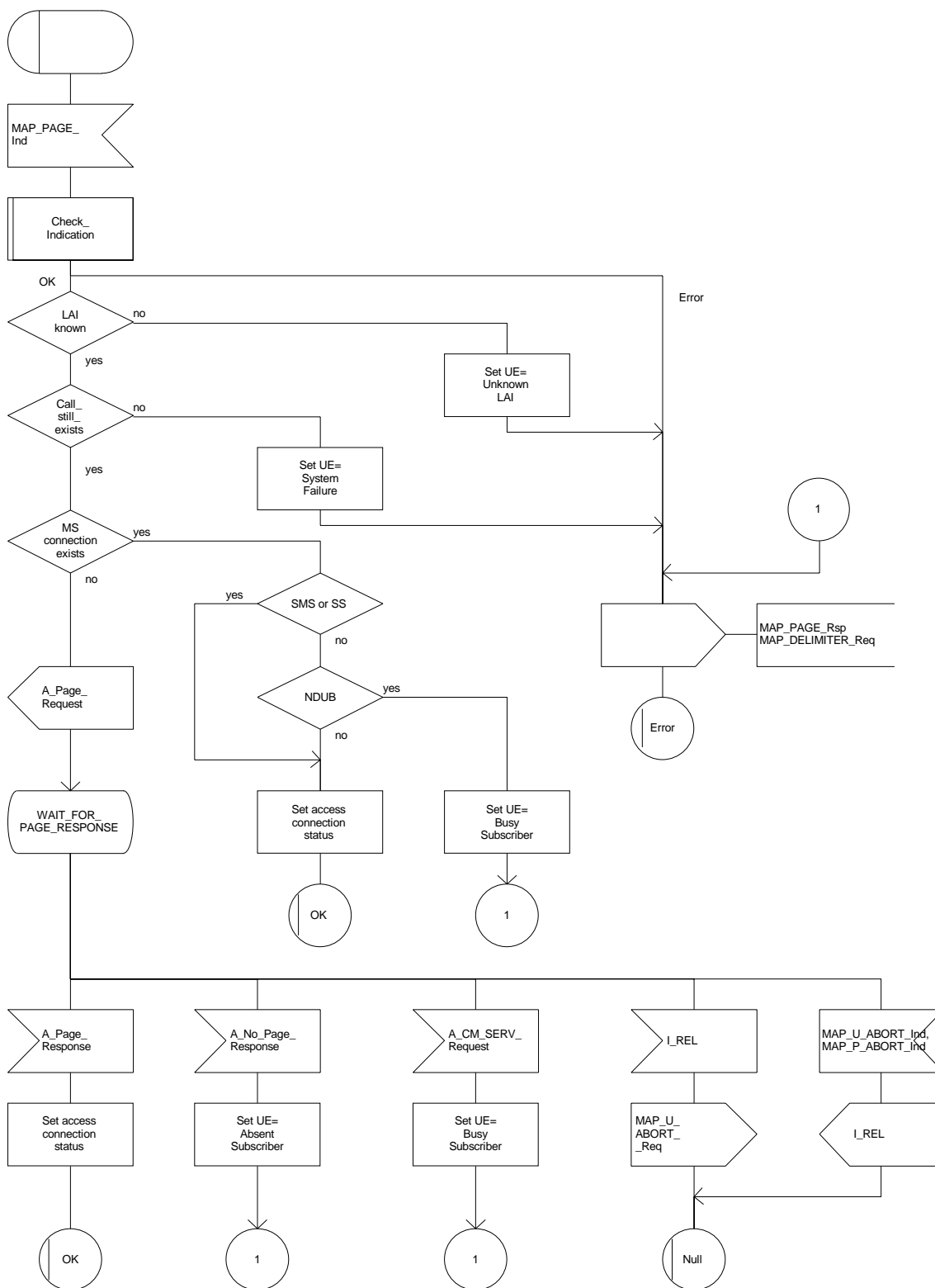
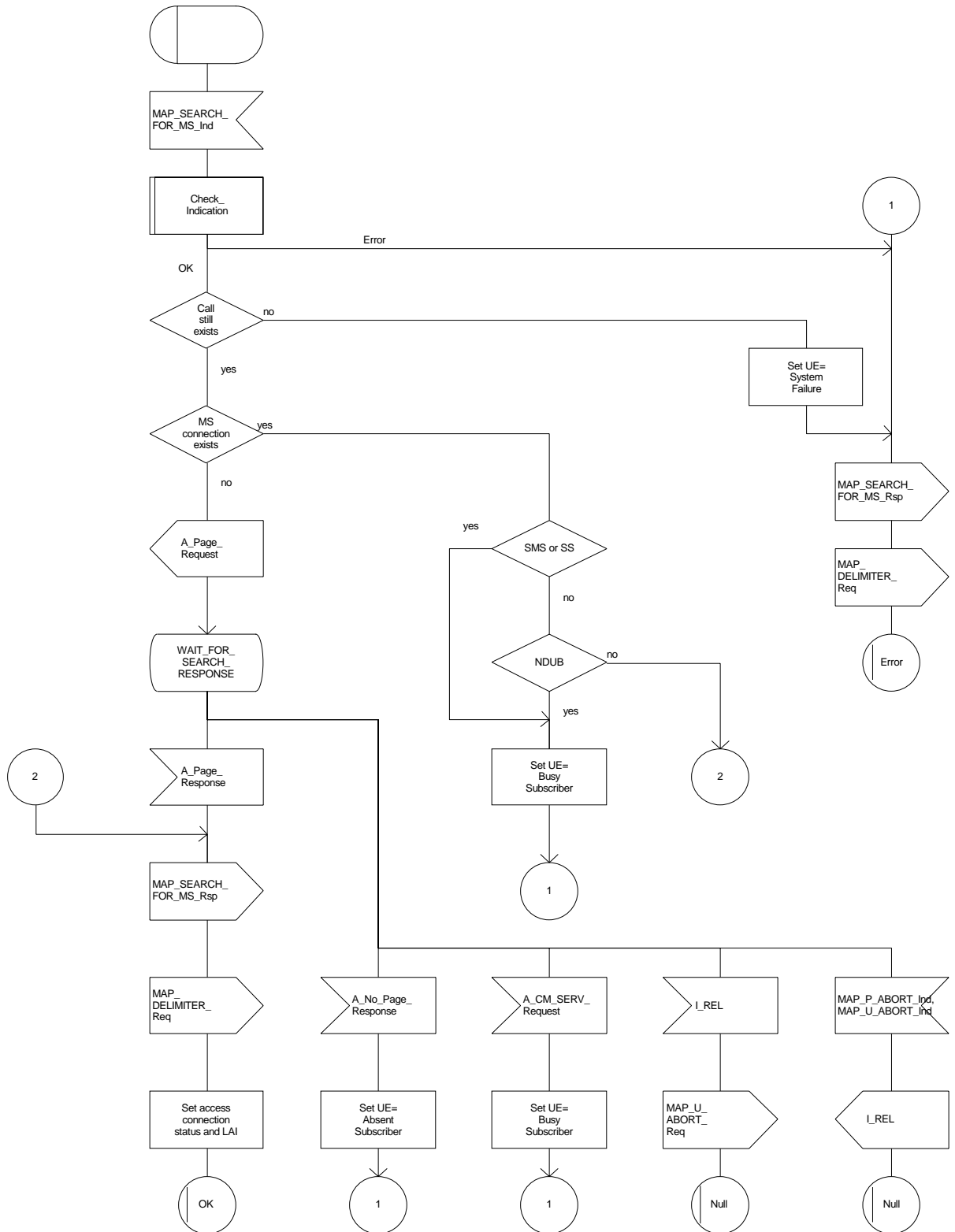


Figure 21.3/2



21.4 **Macros for handling an Access Request**

These macros are invoked when a MS accesses the network, e.g. to set up an outgoing call or when responding to paging. The macro handles identification and authentication of the mobile subscriber as well as invocation of security related features (see GSM 02.09).

21.4.1 **Macro Process_Access_Request_MSC**

This macro is invoked by any procedure receiving an access request from the MS, e.g. the page response at mobile terminating call set-up or the request for outgoing call set-up.

If no dialogue with the VLR exists (e.g. within the procedure for outgoing call set-up), the MSC will open a dialogue towards the VLR by sending a MAP_OPEN request without any user specific parameters.

In any case, the parameters received from the MS are mapped to a MAP_PROCESS_ACCESS_REQUEST request primitive, containing:

- the received subscriber identification (IMSI, TMSI) or - in case of emergency call set-up - an IMEI;
- the CM service type, indicating the type of request;
- the status of the access connection, i.e. whether a connection to this MS already exists and if so, whether it is already authenticated and ciphered;
- the current location area id of the MS; and
- the CKSN received from the MS.

If opening of the dialogue was required, the MSC will wait for the dialogue confirmation (see macro Receive_Open_Confirmation, subclause 21.1), leading either to:

- immediate unsuccessful exit from the macro, in case no dialogue is possible;
- reversion to MAP version one dialogue if indicated by the VLR. The macro terminates with unsuccessful outcome, as the complete dialogue will be covered by the version one procedure, so that no further action from the calling process is required;
- continuation as given below, if the dialogue is accepted by the VLR.

The MSC waits then for the MAP_PROCESS_ACCESS_REQUEST confirmation. In between, several other indications may be received from the VLR:

- the MSC may receive a MAP_PROVIDE_IMSI indication, handled by the macro Obtain_IMSI_MSC defined in subclause 21.8. In case of positive outcome, the procedure continues waiting for the MAP_PROCESS_ACCESS_REQUEST confirmation, else the macro terminates with unsuccessful outcome;
- the MSC may receive a MAP_AUTHENTICATE indication, handled by the macro Authenticate_MSC defined in subclause 21.5. In case of positive outcome, the procedure continues waiting for the MAP_PROCESS_ACCESS_REQUEST confirmation, else the macro terminates with unsuccessful outcome;
- the MSC may receive a MAP_TRACE_SUBSCRIBER_ACTIVITY indication, handled by the macro Trace_Subscriber_Activity_MSC defined in subclause 21.9;

- the MSC may receive a MAP_SET_CIPHERING_MODE indication, which will be stored for initiating ciphering later on;
- the MSC may receive a MAP_CHECK_IMEI indication, handled by the macro Check_IMEI_MSC defined in subclause 21.6. In case of positive outcome, the procedure continues waiting for the MAP_PROCESS_ACCESS_REQUEST confirmation, else the macro terminates with unsuccessful outcome;
- the MSC may receive a MAP_Obtain_IMEI indication, handled by the macro Obtain_IMEI_MSC defined in subclause 21.6. In case of positive outcome, the procedure continues waiting for the MAP_PROCESS_ACCESS_REQUEST confirmation, else the macro terminates with unsuccessful outcome;
- the MSC may receive a MAP_U_ABORT or MAP_P_ABORT indication, or a premature MAP_CLOSE indication from the VLR. In all these cases, the macro terminates with unsuccessful outcome, after sending the appropriate reject towards the MS (see GSM 09.10);
- the MSC may receive a MAP_NOTICE indication from the VLR. In this case, the dialogue towards the VLR is terminated by a MAP_CLOSE primitive, the appropriate reject is sent towards the MS (see GSM 09.10), and the macro terminates with unsuccessful outcome;
- the MSC may receive an indication for release of the radio path, in which case the dialogue towards the VLR will be terminated by a MAP_U_ABORT primitive, containing the diagnostic information Radio Channel Release.

When the MAP_PROCESS_ACCESS_REQUEST confirmation is received, the parameters of this primitive are checked first. In case of unsuccessful outcome of the service, the MAP User Error received is mapped onto the appropriate radio interface message (see GSM 09.10), before the macro terminates with unsuccessful outcome.

In case of positive outcome of the service, ciphering is initiated on the radio path, if this had been requested by the VLR (see above). Otherwise, if the access request was not triggered by a page response from the MS, the access request is accepted explicitly by sending a CM_Service_Accept message to the MS. If the access request was triggered by a page response from the MS then no CM Service Accept message is sent.

After ciphering has been initiated, the MSC will wait for the MAP_FORWARD_NEW_TMSI indication from the VLR. While waiting, the MSC may receive:

- a MAP_U_ABORT or MAP_P_ABORT indication, or a premature MAP_CLOSE indication from the VLR. In these cases, the macro terminates with unsuccessful outcome, after sending a release request towards the MS (see GSM 09.10);
- a MAP_NOTICE indication from the VLR. In this case, the dialogue towards the VLR is terminated by a MAP_CLOSE primitive, the appropriate reject is sent towards the MS (see GSM 09.10), and the macro terminates with unsuccessful outcome;
- an indication for release of the radio path, in which case the dialogue towards the VLR will be terminated by a MAP_U_ABORT primitive, containing the diagnostic information Radio Channel Release;
- a MAP_DELIMITER request from the VLR. This will be taken as a successful outcome of the macro (i.e. the VLR did not require TMSI reallocation), and it terminates successfully;
- an A_SETUP request from the MS. This will be saved for handling by the procedure which invoked the macro Process_Access_Request_MSC after the macro has terminated.

When the MAP_FORWARD_NEW_TMSI indication is received in the MSC, the TMSI Reallocation Command is sent to the MS, and the MSC waits for an acknowledgement from the MS. In case a positive acknowledgement is received, the MSC sends an empty MAP_FORWARD_NEW_TMSI response primitive to the VLR and terminates successfully. Else, the dialogue is terminated locally (MAP_CLOSE_Req with Release method Prearranged End) without any further action.

If the MSC receives an A_SETUP request while it is waiting for the TMSI acknowledgement from the MS, the A_SETUP is saved for handling by the procedure which invoked the macro Process_Access_Request_MSC after the macro has terminated.

If the dialogue is aborted by the VLR while waiting for the TMSI acknowledgement from the MS, the MSC regards the access request to be failed and terminates with unsuccessful outcome, after sending a release request towards the MS (see GSM 09.10).

Figure 21.4/1 (sheet 1 of 3)

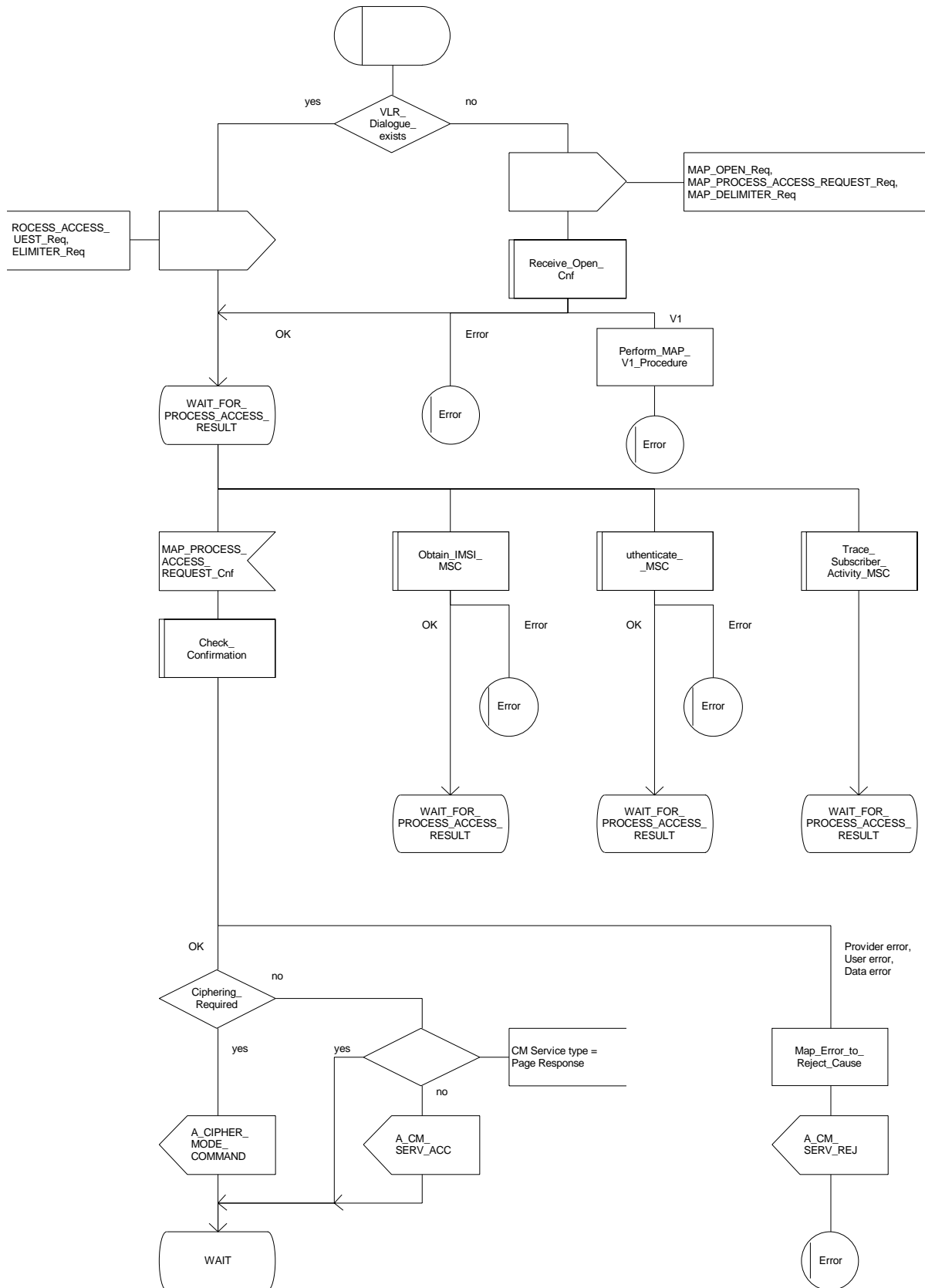


Figure 21.4/1 (sheet 2 of 3)

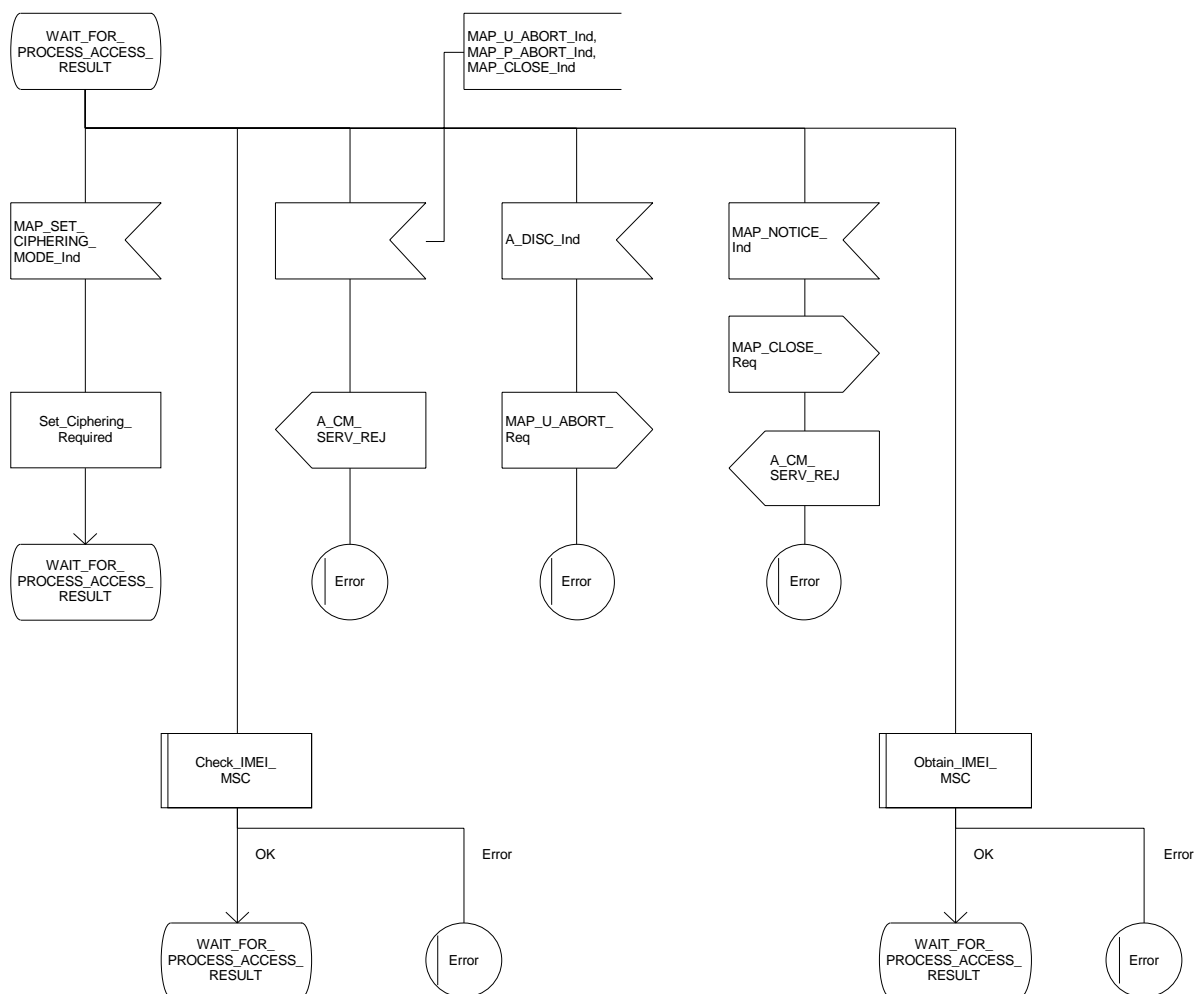
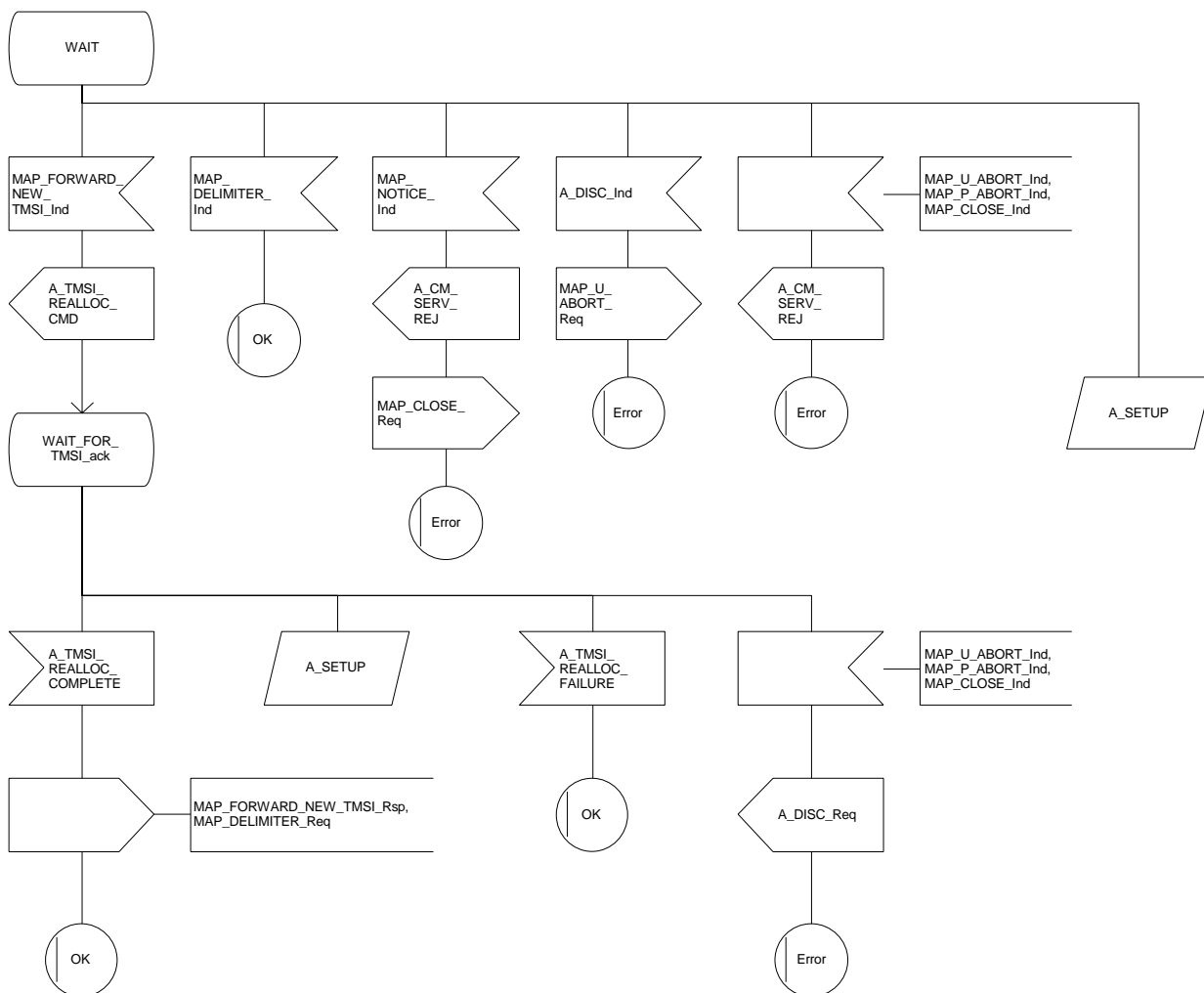


Figure 21.4/1 (sheet 3 of 3)



21.4.2 Macro Process_Access_Request_VLR

When the VLR receives a MAP_PROCESS_ACCESS_REQUEST indication, the VLR will check this indication first (macro Check_Indication, see subclause 21.2). In case of negative outcome, the macro will proceed with the error handling described below.

If the indication data are correct, it is checked first whether the subscriber identification (IMSI or TMSI) is known if included:

- if the identification is not known, the IMSI may be requested from the MS, described in the macro Identification_Procedure (see below) with outcome:
 - OK, if a IMSI known in the VLR has been received;
 - Error, if the VLR did not recognize the subscriber's identity. The macro will proceed with the error handling described below;
 - Aborted, if the transaction to the MSC is released. The macro will terminate immediately with unsuccessful.

In case the identity received is an IMEI, the error System Failure is set and the macro proceeds with the error handling described below.

NOTE: Emergency Call with IMEI may be accepted within the error handling phase.

For a known subscriber the authentication check is performed next (see macro Authenticate_VLR, subclause 21.5), if required. If a negative result is received, the VLR proceeds on receipt of user error:

- illegal subscriber depending on the identity used for authentication;

In case IMSI is already used or no new authentication attempt with IMSI shall not be performed (operator option), the error Illegal Subscriber is set and the macro proceeds with the error handling described below.

If a new authentication attempt with IMSI shall be performed, the IMSI is requested from the MS (macro Obtain_IMSI_VLR, see subclause 21.8):

 - the authentication will be performed again if a IMSI known in the VLR is received;
 - the error Unidentified Subscriber is set and the macro proceeds with the error handling described below, if the IMSI received is unknown in VLR;
 - if the IMSI request procedure fails for any other reason, the error System Failure is set and the macro proceeds with the error handling described below;
 - if the dialogue has been aborted during the IMSI request, the macro terminates immediately with unsuccessful outcome;
- unknown subscriber by setting the error Unidentified Subscriber and proceeding with the error handling described below.

NOTE: This can occur only in case of data inconsistency between HLR and VLR;

- procedure error by setting the error System Failure and proceeding with the error handling described below;
- null (i.e. the dialogue towards the MSC is terminated) by terminating immediately with unsuccessful outcome.

The MS access is accepted if no authentication is required or after successful authentication. Then, the indicator "Confirmed by Radio Contact" is set to "Confirmed". If the indicator "Location Information Confirmed in HLR" is set to "Not Confirmed", HLR updating will be started as an independent process (Update_Location_VLR, see subclause 16.1.1.6).

If the indicator "Confirmed by HLR" is set to "Not Confirmed", the error Unidentified Subscriber is set and the macro proceeds with the error handling described below.

If roaming is not allowed in the location area indicated in the Current Location Area Id parameter, the error Roaming Not Allowed qualified by the roaming restriction reason is set and the macro proceeds with the error handling described below.

In case roaming is allowed, the IMSI is set to attached and the process for notifying the HLR that the subscriber is present is started if required (Subscriber Present VLR, see subclause 21.10).

At next, tracing is invoked if required by the operator (macro Trace_Subscriber_Activity_VLR, see subclause 21.9). Thereafter,

if ciphering is not required, IMEI checking is invoked if required by the operator (see macro Check_IMEI_VLR defined in subclause 21.6).

The error Illegal Equipment is set in case of unsuccessful outcome of the IMEI check, the subscriber is marked as detached and the macro proceeds with the error handling described below.

The macro terminates immediately with unsuccessful outcome if the MSC dialogue has been released during the IMEI check.

Else, the macro terminates successfully by returning the MAP_PROCESS_ACCESS_REQUEST response containing the IMSI to indicate acceptance of the MS access.

if ciphering is required, the MAP_SET_CIPHERING_MODE request containing:

- the cipher mode indicating the cipher algorithm required; and
- the cipher key to be used;

is sent to the MSC.

As a further operator option, IMEI checking may be performed next.

The error Illegal Equipment is set in case of unsuccessful outcome of the IMEI check, the subscriber is marked as detached and the macro proceeds with the error handling described below.

The macro terminates immediately with unsuccessful outcome if the MSC dialogue has been released during the IMEI check.

Else, the macro terminates successfully by returning the MAP_PROCESS_ACCESS_REQUEST response containing the IMSI to indicate acceptance of the MS access.

IF no TMSI reallocation is required (again an operator option), the macro terminates thereafter. Else, TMSI reallocation is performed by sending a MAP_FORWARD_NEW_TMSI request, containing the new TMSI as parameter. The old TMSI will be frozen until an acknowledgement from the MS has been received. Before the macro terminates, the VLR will wait for the MAP_FORWARD_NEW_TMSI response, containing no parameters if reallocation has been confirmed by the MS, or a Provider Error, otherwise, in which case the old TMSI is kept frozen to avoid double allocation. In this case, both the old as the new TMSI are subsequently regarded valid when used by the MS.

Error handling

In case some error is detected during handling the access request, a respective error has been set. Before returning this error cause to the MSC in a MAP_PROCESS_ACCESS_REQUEST response, it need to be checked whether this access is for emergency call set-up, as this will require extra treatment.

If the CM Service type given in the MAP_PROCESS_ACCESS_REQUEST indication is emergency call set-up, it is checked whether EC set-up in the particular error situation is permitted (operator option). If so, it is checked whether the IMEI is required, and if so the IMEI is requested from the MS (macro Obtain_IMEI_VLR, see subclause 21.6).

The macro will terminate immediately with unsuccessful outcome if the MSC transaction has been aborted during the IMEI retrieval.

In case of an error reported back from IMEI retrieval, MAP_PROCESS_ACCESS_REQUEST response containing the error cause set previously is returned to the MSC, the dialogue is closed (MAP_CLOSE request indicating normal release) and the macro terminates with unsuccessful outcome.

When a subscriber identity required by the operator (IMSI or IMEI) is available, the user error set previously is deleted, the respective identity is returned in the MAP_PROCESS_ACCESS_REQUEST response to indicate acceptance of emergency call, and the macro terminates with successful outcome.

In all other cases, the MAP_PROCESS_ACCESS_REQUEST response containing the error cause set previously is returned to the MSC, the dialogue is closed (MAP_CLOSE request indicating normal release) and the macro terminates with unsuccessful outcome.

21.4.3 Macro Identification Procedure

This macro is invoked by the macro Process_Access_Request_VLR in case the subscribers identity is not known in the VLR.

If the identity received from the MS is an IMSI, the error Unidentified Subscriber will be set and reported back to the calling macro (to be sent in the MAP_PROCESS_ACCESS_REQUEST response). The same error is used in case a TMSI was received from the MS, but the operator does not allow open identification of the MS.

If open identification of the MS is allowed, the macro Obtain_IMSI_VLR is invoked, requesting the subscribers IMSI from the MS (see subclause 21.8), with outcome

OK, in which case it is checked whether for the IMSI received there exists a subscriber record in the VLR. If so, the macro terminates successfully, else the error Unidentified Subscriber will be set and reported back to the calling macro.

Error, in which case the error System Failure will be set and reported back to the calling macro.

Aborted, i.e. the MSC transaction is released, in which the macro terminates accordingly.

Figure 21.4/2 (sheet 1 of 3)

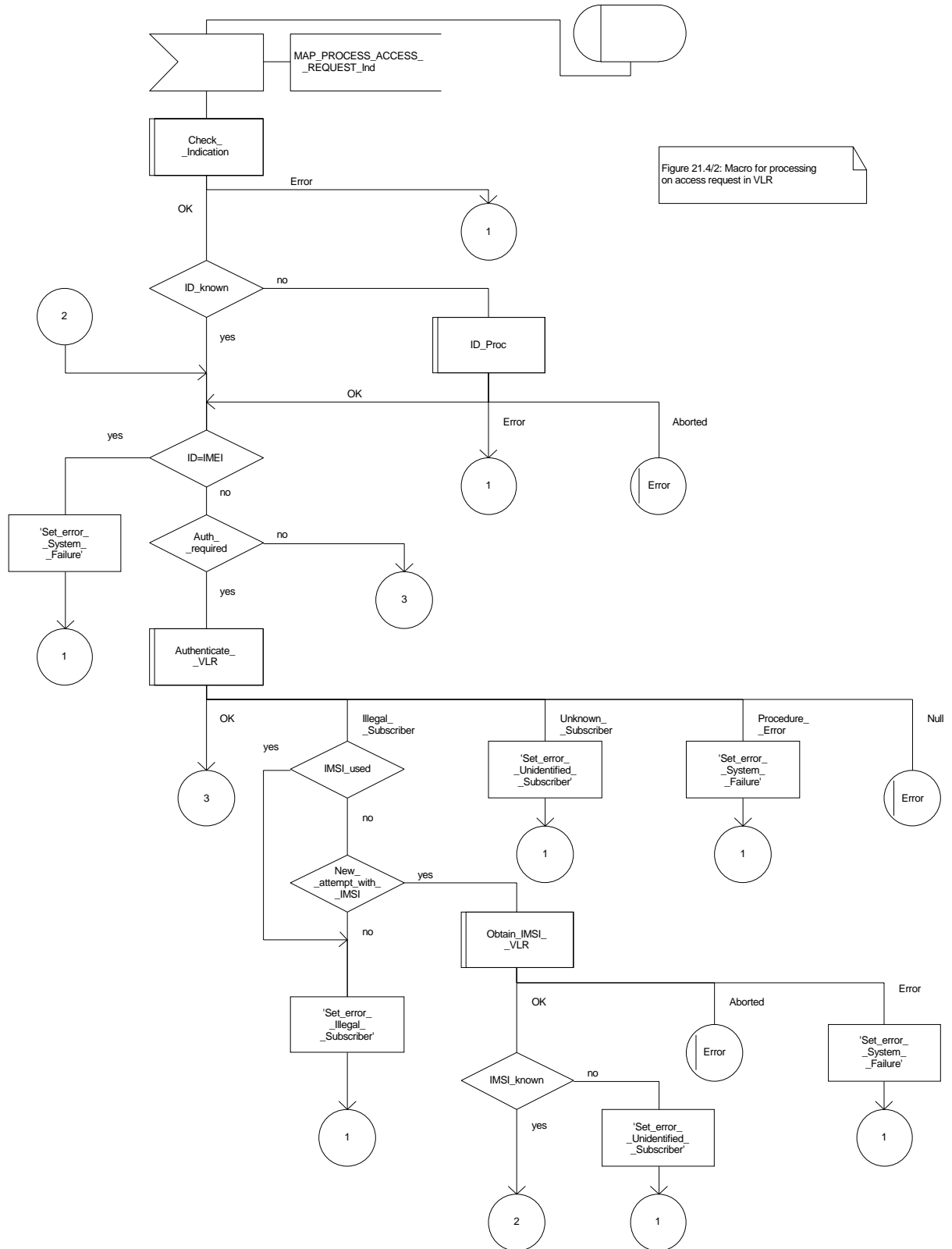


Figure 21.4/2 (sheet 2 of 3)

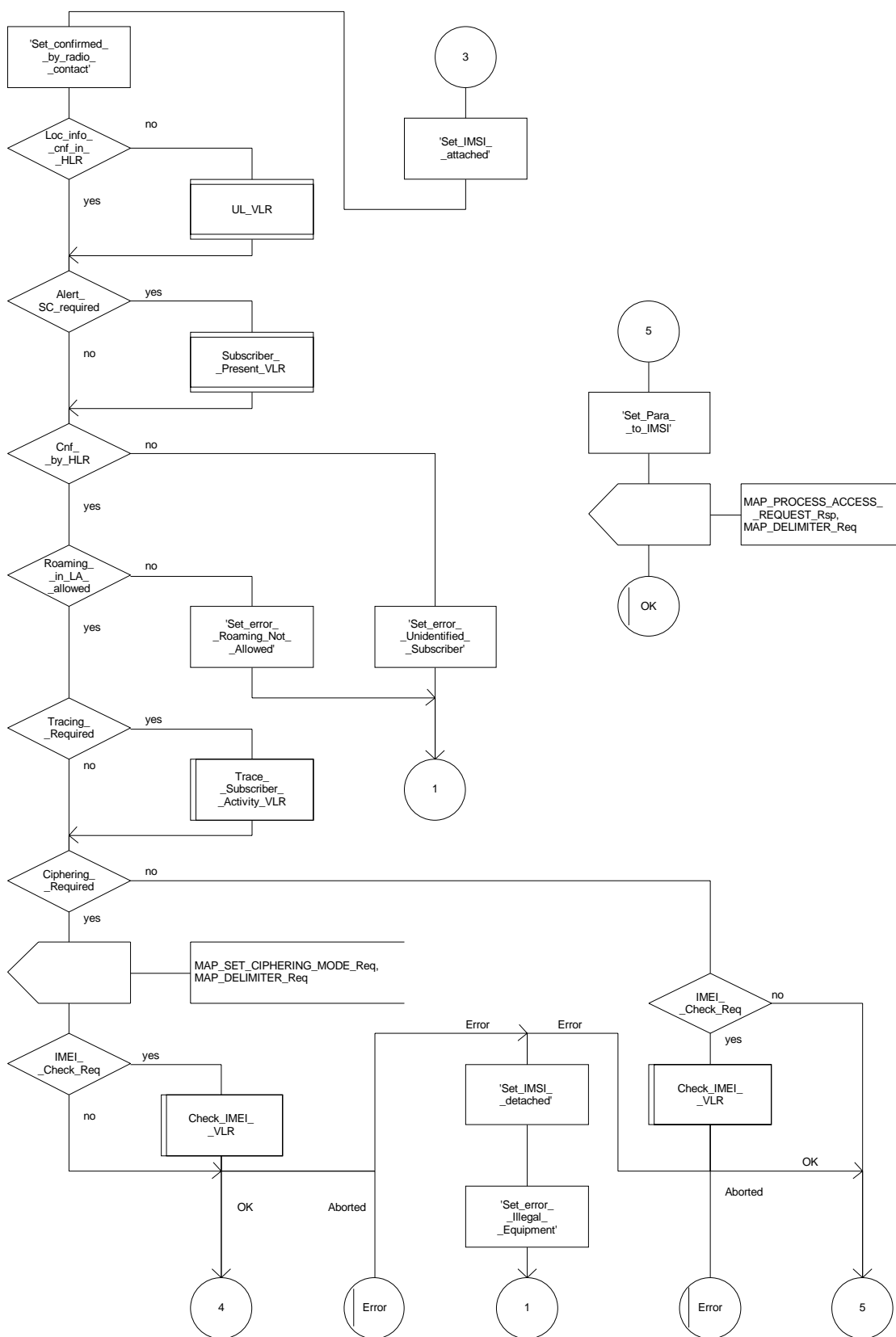


Figure 21.4/2 (sheet 3 of 3)

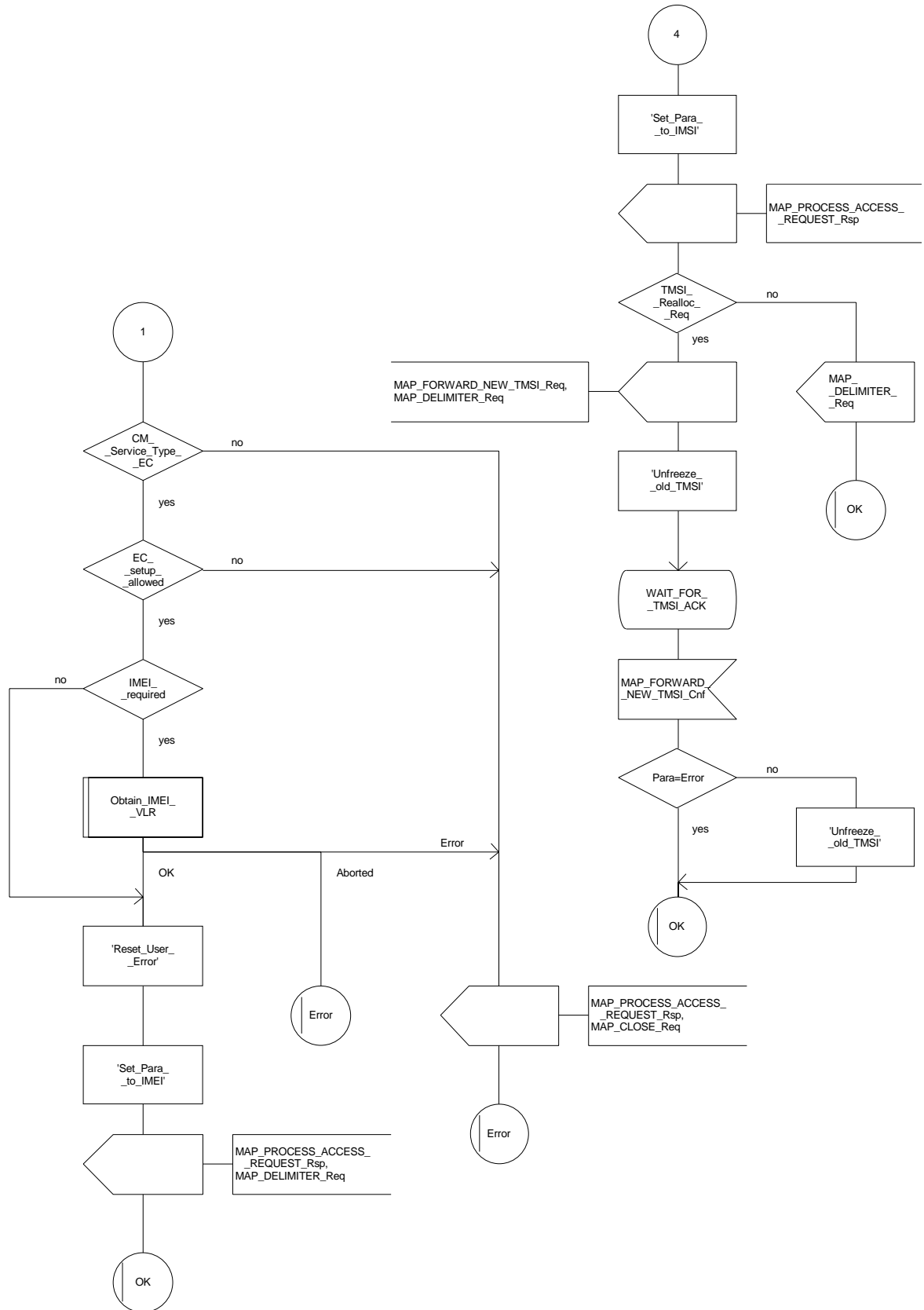
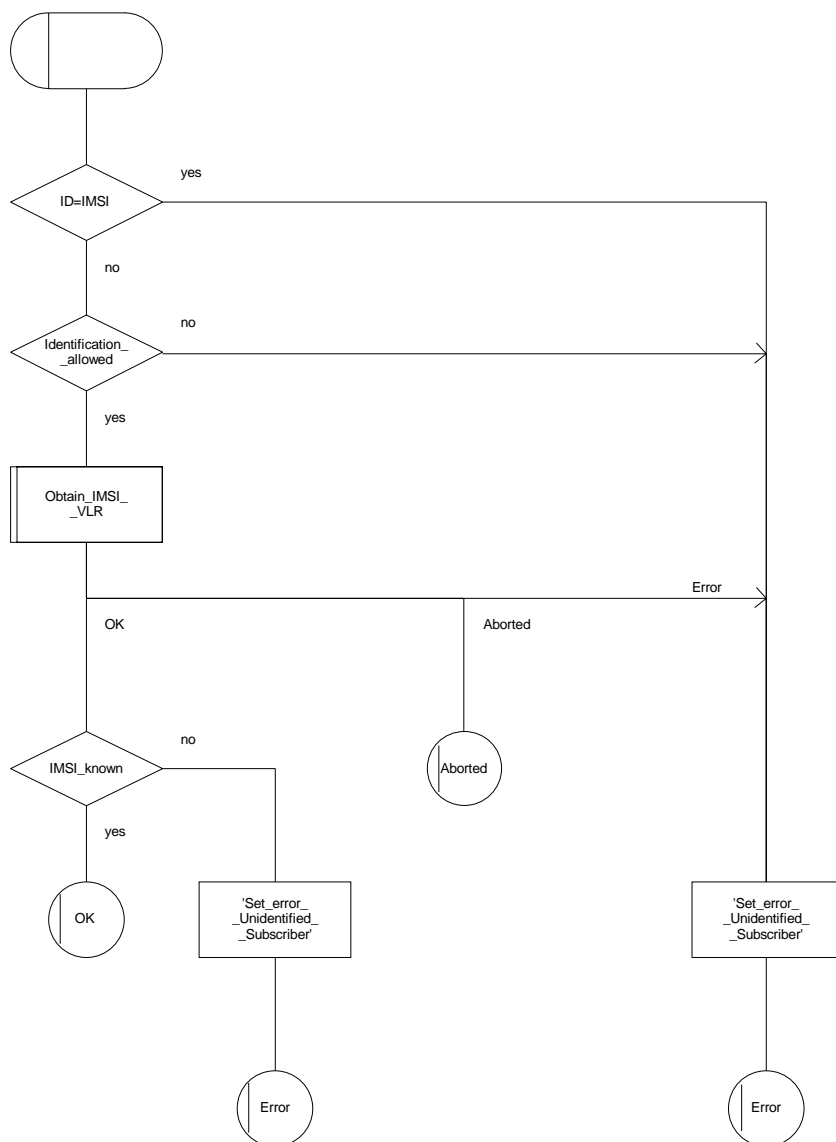


Figure 21.4/3



21.5 Authentication macros and processes

The following macros are used in the GSM network in order to enable authentication of a mobile subscriber.

21.5.1 Macro Authenticate_MSC

This macro is used by the MSC to relay a request for authentication transparently from the VLR to the MS, wait for a response from the MS and to relay the response from the MS back to the VLR. If, while the MSC is waiting for the authentication response, the air interface connection is released or a MAP_U_ABORT, MAP_P_ABORT or MAP_CLOSE indication is received from the VLR, then necessary connections are released and the "Error" exit is used. The macro is described in figure 21.5/1.

21.5.2 Macro Authenticate_VLR

This macro is used by the VLR to control the authentication of a subscriber. The macro proceeds as follows:

- if there are not enough authentication triplets in the VLR to perform the authentication, then the macro "Obtain_Authent_Para_VLR" described below is invoked. If this macro fails, then the corresponding error (Unknown Subscriber or Procedure Error) is returned to the calling process;
- if there are enough authentication triplets in the VLR, or the Obtain_Authent_Para_VLR macro was successful, then a MAP_AUTHENTICATE request is sent to the MSC. This request contains the RAND and CKSN parameters as indicated in the service description;
- the VLR then waits for a response from the MSC;
- if a MAP_U_ABORT, MAP_P_ABORT or MAP_CLOSE indication is received from the MSC in this wait state, the VLR checks whether authentication sets are available. If no sets are available the process Obtain_Authent_Sets_VLR is invoked to fetch authentication sets from the HLR. The "Null" exit is then used;
- if a MAP_NOTICE indication is received from the MSC in this wait state, the VLR closes the dialogue with the MSC, then checks whether authentication sets are available. If no sets are available the process Obtain_Authent_Sets_VLR is invoked to fetch authentication sets from the HLR. The "Null" exit is then used;
- if a MAP_AUTHENTICATE confirmation is received by the VLR, it checks whether the received Signed Result (SRES) is identical to the stored one (see GSM 03.20). If this is not the case, the "Illegal Subscriber" exit is used. If the SRES values are identical, then the "OK" exit is used;
- before exit, the VLR may fetch a new set of triplets from the HLR. This is done by initiating a separate Obtain_Authent_Sets_VLR process described below.

The macro is described in figure 21.5/2.

21.5.3 Process Obtain_Authentication_Sets_VLR

This process is initiated by the VLR to fetch triplets from a subscriber's HLR in a stand-alone, independent manner. The Obtain_Authent_Para_VLR macro described below is simply called; the process is described in figure 21.5/3.

21.5.4 Macro Obtain_Authent_Para_VLR

This macro is used by the VLR to request authentication triplets from the HLR. The macro proceeds as follows:

- a connection is opened, and a MAP_SEND_AUTHENTICATION_INFO request sent to the HLR;
- if the HLR indicates that a MAP version 1 dialogue is to be used, the VLR performs the equivalent MAP version 1 dialogue. which can return a positive result containing authentication sets, an empty positive result, or an error;
- if the dialogue opening fails, the "Procedure Error" exit is used. Otherwise, the VLR waits for the response from the HLR;
- if a MAP_SEND_AUTHENTICATION_INFO confirmation is received from the HLR, the VLR checks the received data.

One of the following positive responses may be received from a MAP version 1 or MAP version 2 dialogue with the HLR:

- Authentication triplets, in which case the outcome is successful;
- Empty response, in which case the VLR may re-use old triplets, if allowed by the PLMN operator.

If the VLR cannot re-use old triplets (or no such triplets are available) then the "Procedure Error" exit is used.

If the outcome was successful or re-use of old parameters in the VLR is allowed, then the "OK" exit is used.

If an "Unknown Subscriber" error is included in the MAP_SEND_AUTHENTICATION_INFO confirm or is returned by the MAP version 1 dialogue, then the "Unknown Subscriber" exit is used.

- if a MAP-U-ABORT, MAP_P_ABORT, MAP_NOTICE or unexpected MAP_CLOSE service indication is received from the MSC, then open connections are terminated, and the macro takes the "Null" exit;
- if a MAP-U-ABORT, MAP_P_ABORT or unexpected MAP_CLOSE service indication is received from the HLR, then the VLR checks whether old authentication parameters can be re-used. If old parameters cannot be re-used the macro takes the "Procedure Error" exit; otherwise it takes the "OK" exit;
- if a MAP_NOTICE service indication is received from the HLR, then the dialogue with the HLR is closed. The VLR then checks whether old authentication parameters can be re-used. If old parameters cannot be re-used the macro takes the "Procedure Error" exit; otherwise it takes the "OK" exit.

The macro is described in figure 21.5/4.

21.5.5 **Process Obtain_Auth_Sets_HLR**

Opening of the dialogue is described in the macro Receive_Open_Ind in subclause 21.1, with outcomes:

- reversion to version one procedure;
- procedure termination; or
- dialogue acceptance, with proceeding as below.

This process is used by the HLR to obtain authentication triplets from the AuC, upon request from the VLR. The process acts as follows:

- a MAP_SEND_AUTHENTICATION_INFO indication is received by the HLR;
- the HLR checks the service indication for errors. If any, they are reported to the VLR in the MAP_SEND_AUTHENTICATION_INFO response. If no errors are detected, authentication triplets are fetched from the AuC. Further details are found in GSM 03.20;
- if errors are detected they are reported to the VLR in the MAP_SEND_AUTHENTICATION_INFO response. Otherwise the authentication triplets are returned.

The process is described in figure 21.5/5.

Figure 21.5/1

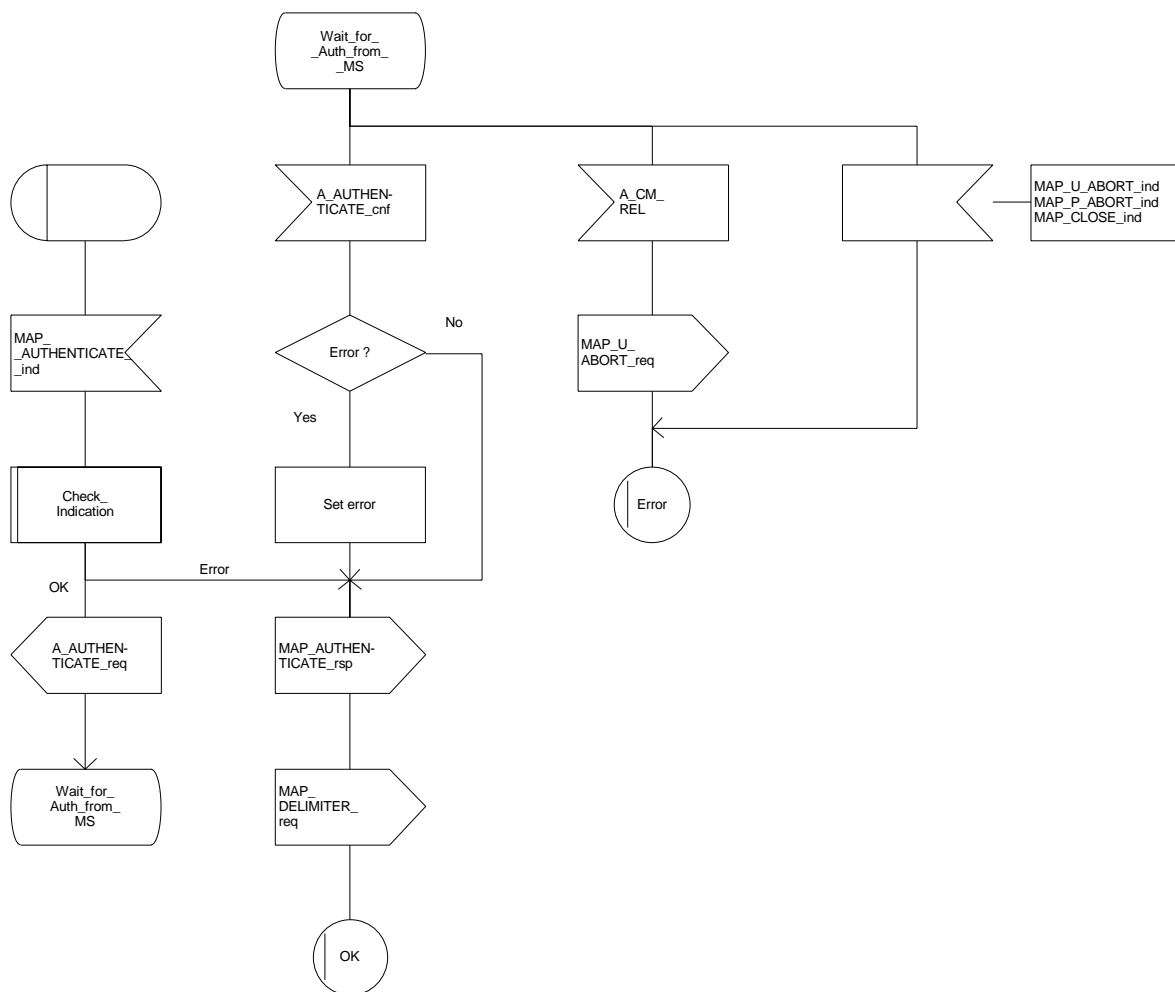


Figure 21.5/2

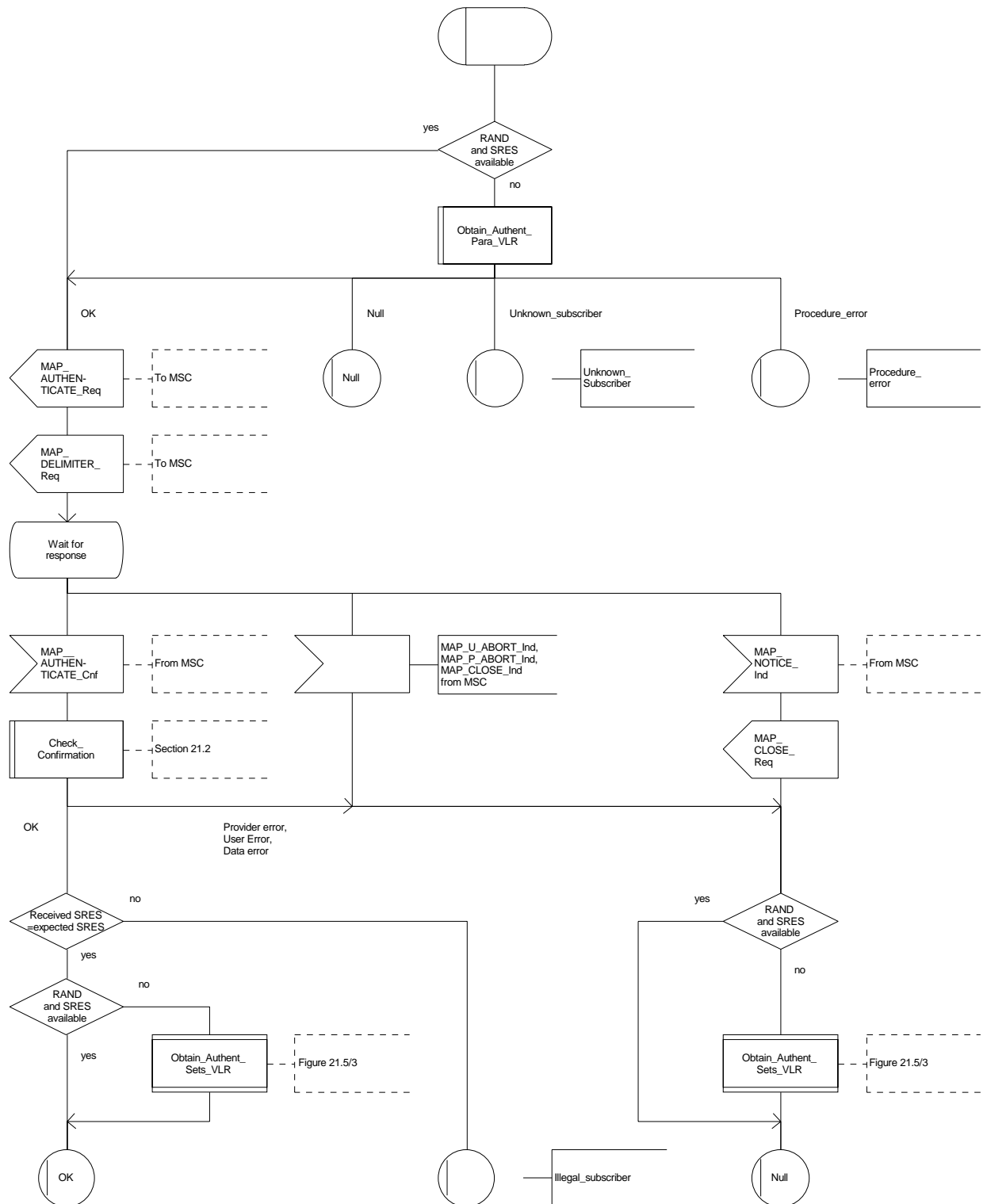


Figure 21.5/3

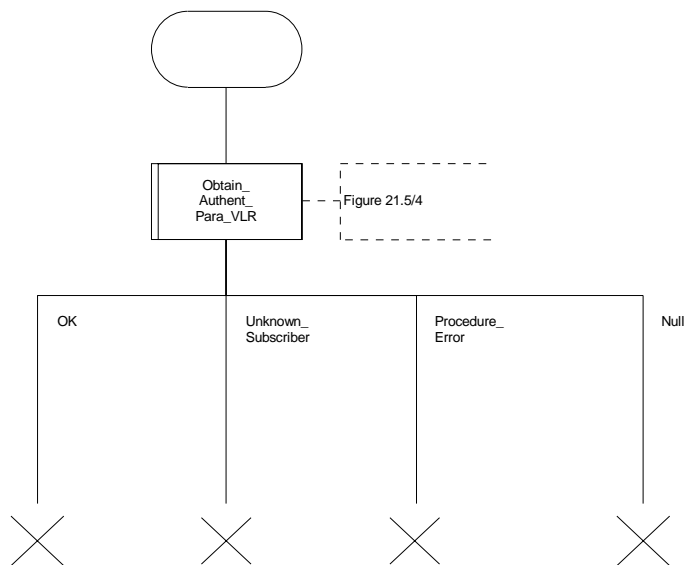


Figure 21.5/4 (sheet 1 of 2)

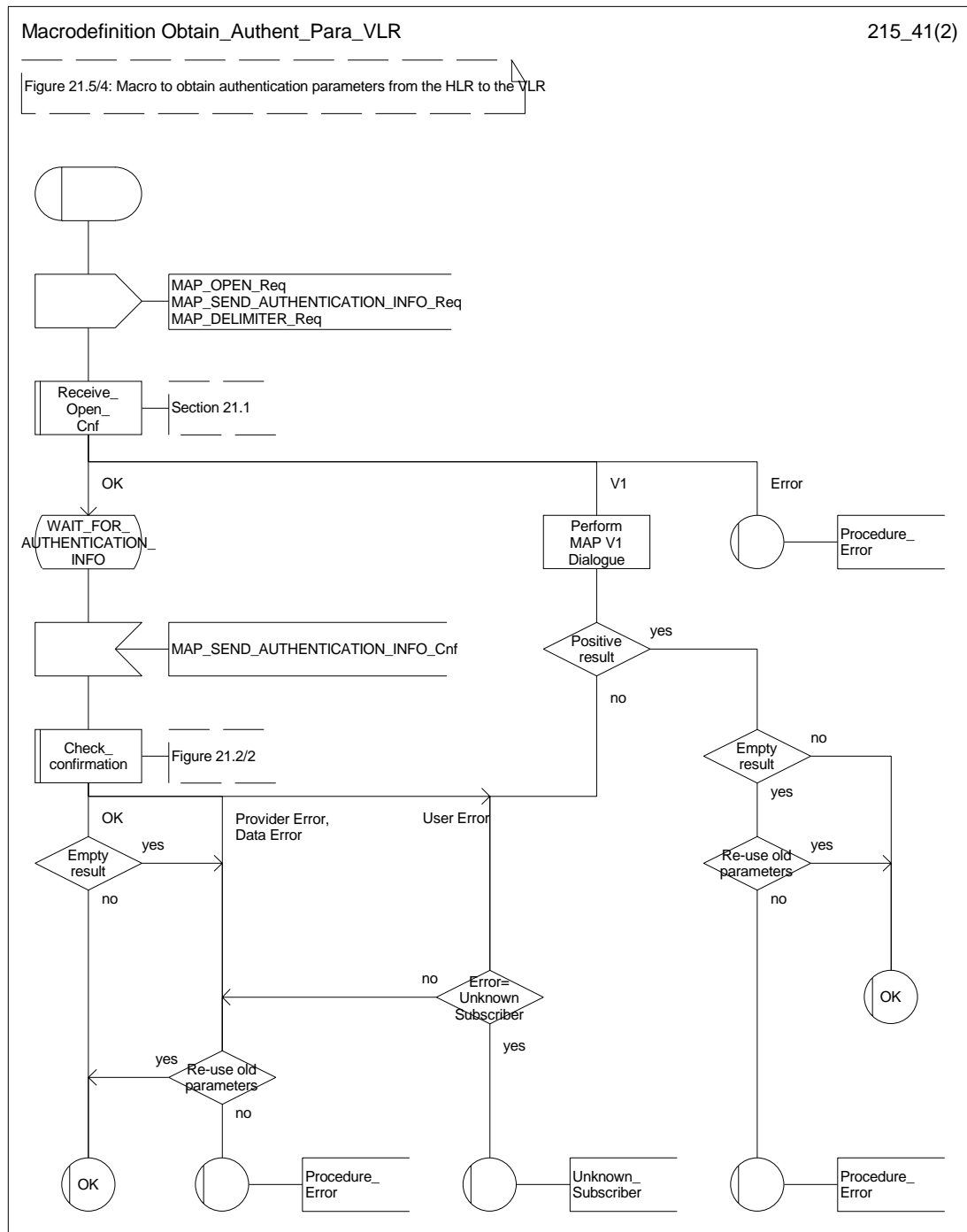


Figure 21.5/4 (sheet 2 of 2)

Macrodefinition Obtain_Authent_Para_VLR

215_42(2)

Figure 21.5/4: Macro to obtain authentication parameters from the HLR to the VLR

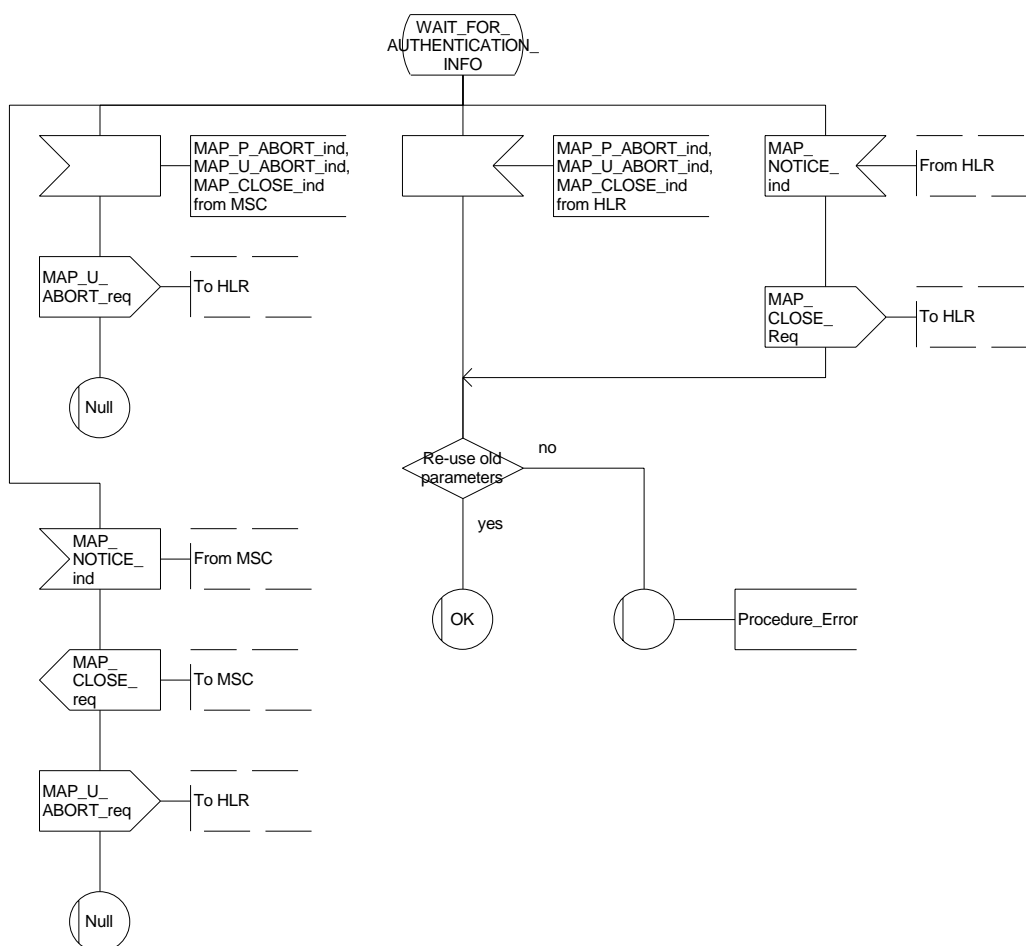
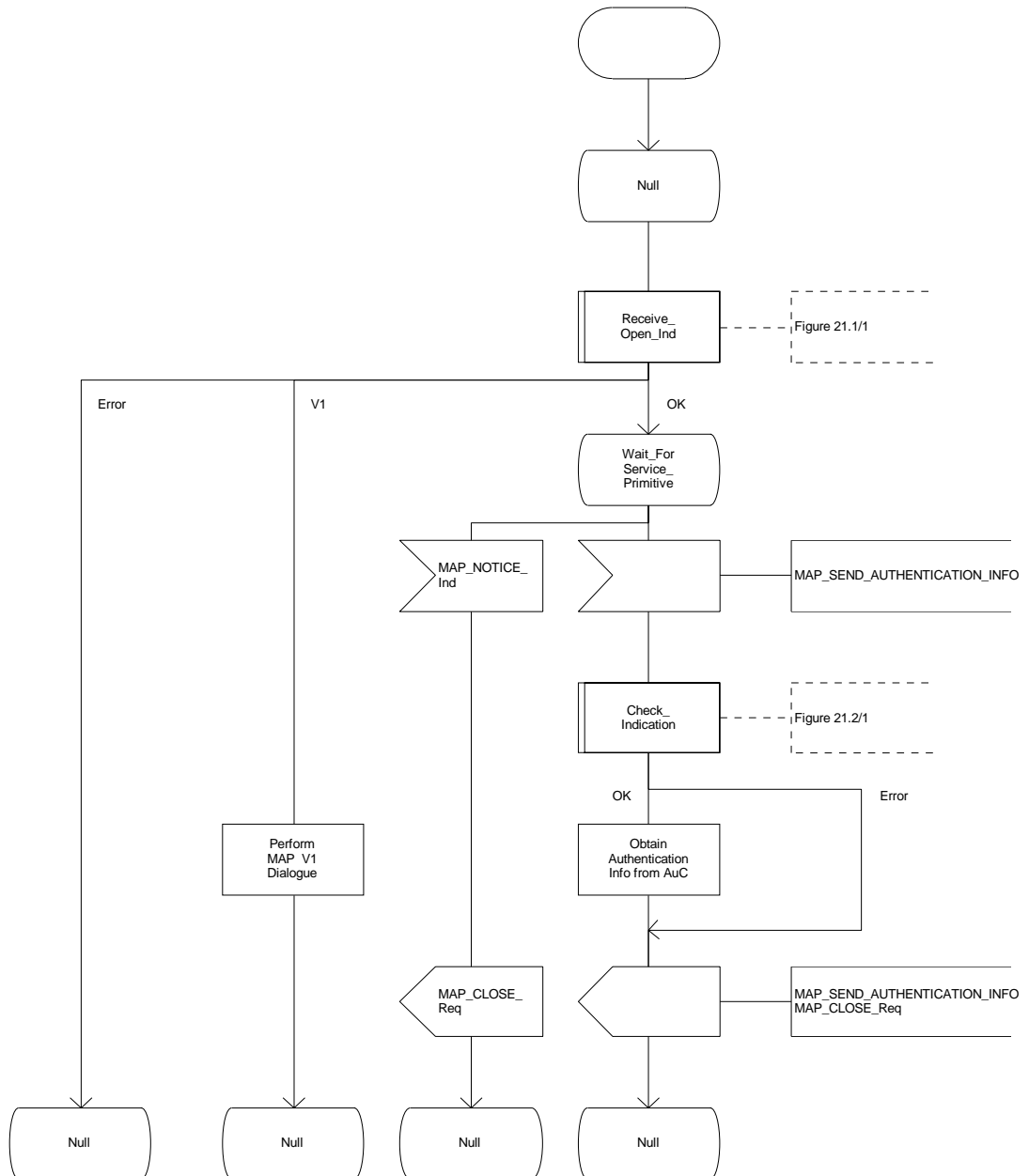


Figure 21.5/5



21.6 IMEI Handling Macros

The following macros are used in the GSM network in order to enable handling and checking of the mobile equipment identity.

21.6.1 Macro Check_IMEI_MSC

This macro is used by the MSC to receive a request from the VLR, relay it to the EIR, and pass the result from the EIR back to the VLR. The macro proceeds as follows:

- a MAP_CHECK_IMEI service indication containing only the Invoke Id is received from the VLR;
- if the IMEI is not available in the MSC, it is requested from the MS using the IDENTITY REQUEST message;
- if the MS releases the radio resources, a MAP_U_ABORT request indicating "Application procedure Cancellation" is sent to the VLR, and the "Error" exit of the macro is used;
- when the IMEI is known, a connection is set up towards the EIR, and a MAP_CHECK_IMEI service request is sent including the IMEI;
- if the opening of the dialogue fails, a System Failure is reported to the VLR. Otherwise, the MSC waits for a response from the EIR;
- when the MAP_CHECK_IMEI service confirm is received, it is checked for errors. Any errors discovered in the MSC lead to the System Failure error to be reported to the VLR in the MAP_CHECK_IMEI response. Any errors reported from the EIR are sent directly to the VLR in the MAP_CHECK_IMEI service response. If no errors are detected by or reported to the MSC, the IMEI is added to the MAP_CHECK_IMEI service response returned to the VLR. The "OK" exit is used in all cases;
- if a MAP_P_ABORT, MAP_U_ABORT, MAP_CLOSE or MAP_NOTICE service indication is received from the EIR, the MSC closes the transaction with the EIR (if necessary), reports a System Failure error back to the VLR in the MAP_CHECK_IMEI response, and uses the macro's "OK" exit;
- if a MAP_P_ABORT, MAP_U_ABORT, MAP_CLOSE or MAP_NOTICE indication is received from the VLR, the MSC closes the transaction with the VLR (if necessary) and aborts the connections towards the EIR and the MS; the macro takes the "Error" exit.

If the dialogue with the EIR drops back to version 1, the result or error returned by the EIR is checked. The use of the "Check_Confirmation" macro in the SDL diagram indicates that the checks carried out on the result returned by the EIR in a MAP v1 dialogue are functionally equivalent to those carried out on the parameters of the MAP_CHECK_IMEI confirm received from the EIR in a MAP v2 dialogue.

The macro is described in figure 21.6/1.

21.6.2 Macro Check_IMEI_VLR

This macro is used by the VLR to control the check of a mobile equipment's IMEI. The macro proceeds as follows:

- a MAP_CHECK_IMEI service request is sent to the MSC, including only the Invoke Id;
- the VLR then waits for the response from the MSC;

- if a MAP_CHECK_IMEI service confirm including either:
 - the IMEI and the Equipment Status; or
 - an error;
 is received, the VLR checks whether the response requires that an alarm be generated on the Operation and Maintenance interface. The criteria for such alarms are PLMN operator dependent;
- the VLR then checks whether the response from the MSC means that service is granted to the MS. The criteria for granting service depending on the equipment status or errors received in the MAP_CHECK_IMEI service response are also PLMN operator dependent;
- if a MAP_P_ABORT, MAP_U_ABORT, MAP_CLOSE or MAP_NOTICE indication is received from the MSC, then the MSC connection is closed (if necessary) and the macro takes the "Aborted" exit.

The macro is described in figure 21.6/2.

21.6.3 Process Check_IMEI_EIR

This process is used by the EIR to obtain the status of a piece of mobile equipment, upon request from the MSC. The process acts as follows:

- a MAP_OPEN service indication is received (macro Receive_Open_Ind, subclause 21.1.1). If the dialogue opening fails, the process terminates;
- otherwise, a MAP_CHECK_IMEI indication is received by the EIR, containing the IMEI to be checked;
- the EIR checks the service indication for errors. If there are any, they are reported to the MSC in the MAP-CHECK_IMEI response. If no errors are detected, the EIR data base function is interrogated for the status of the given equipment. Further details are found in GSM 02.16;
- the status of the equipment (white-listed, grey-listed, black-listed or unknown) is returned to the MSC in the MAP_CHECK_IMEI service response;
- if a MAP_U_ABORT, MAP_P_ABORT, MAP_NOTICE or MAP_CLOSE indication is received from the MSC at any time during this process, the process in the EIR terminates.

The process is described in figure 21.6/3.

21.6.4 Macro Obtain_IMEI_MSC

This macro is used by the MSC to respond to a request from the VLR to provide the IMEI. The macro proceeds as follows:

- a MAP_OBTAIN_IMEI service indication containing only the Invoke Id is received from the VLR;
- if the IMEI is not available in the MSC, it is requested from the MS using the IDENTITY REQUEST message;
- when the IMEI is known, it is returned to the VLR in the MAP_OBTAIN_IMEI service response. The macro terminates at the "OK" exit;
- if the IMEI cannot be obtained by the MSC, the System Failure error is reported back to the VLR in the MAP_OBTAIN_IMEI service response. The macro terminates at the "OK" exit;
- if a MAP_P_ABORT, MAP_U_ABORT or MAP_CLOSE indication is received from the VLR, the macro terminates at the "Error" exit.

The macro is described in figure 21.6/4.

21.6.5 Macro Obtain_IMEI_VLR

This macro is used by the VLR to obtain the IMEI from the MSC, e.g. to enable handling of emergency calls in case of authentication failure (in which case the IMEI may be used by some operators as an alternative to the IMSI). It proceeds as follows:

- the MAP_OBTAIN_IMEI service request is sent to the MSC, including only the Invoke Id;
- the VLR then waits for the response from the MSC;
- if the IMEI is received in the MAP_OBTAIN_IMEI service response, the macro terminates at the "OK" exit;
- if the System Failure error is reported in the MAP_OBTAIN_IMEI service response, the "Error" exit is used;
- if the MSC terminates the dialogue using a MAP_P_ABORT, MAP_U_ABORT, MAP_CLOSE or MAP_NOTICE service indication, the necessary connections are released, and the "Aborted" exit is used for termination of the macro.

The macro is shown in figure 21.6/5.

Figure 21.6/1 (sheet 1 of 2)

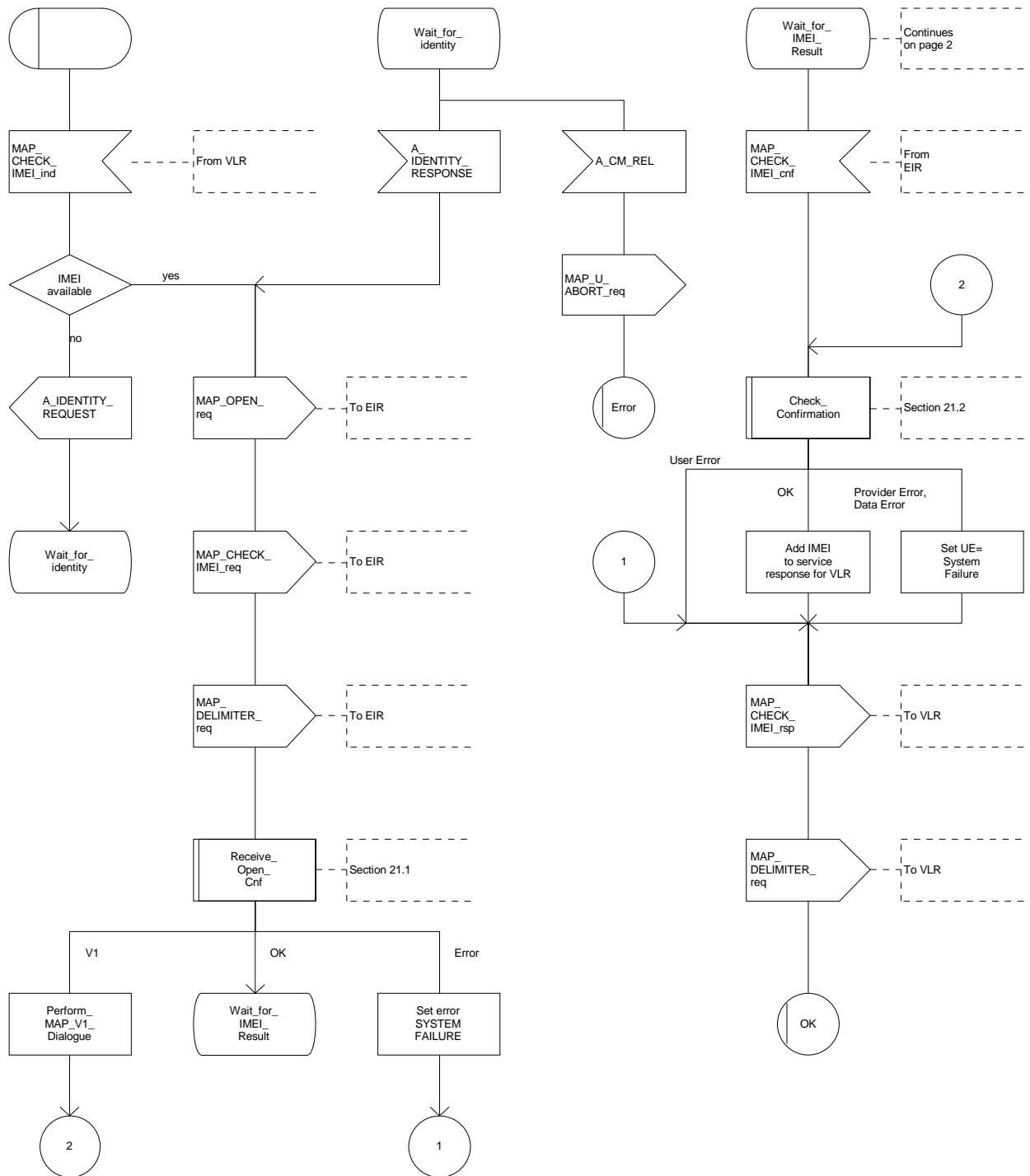


Figure 21.6/1 (sheet 2 of 2)

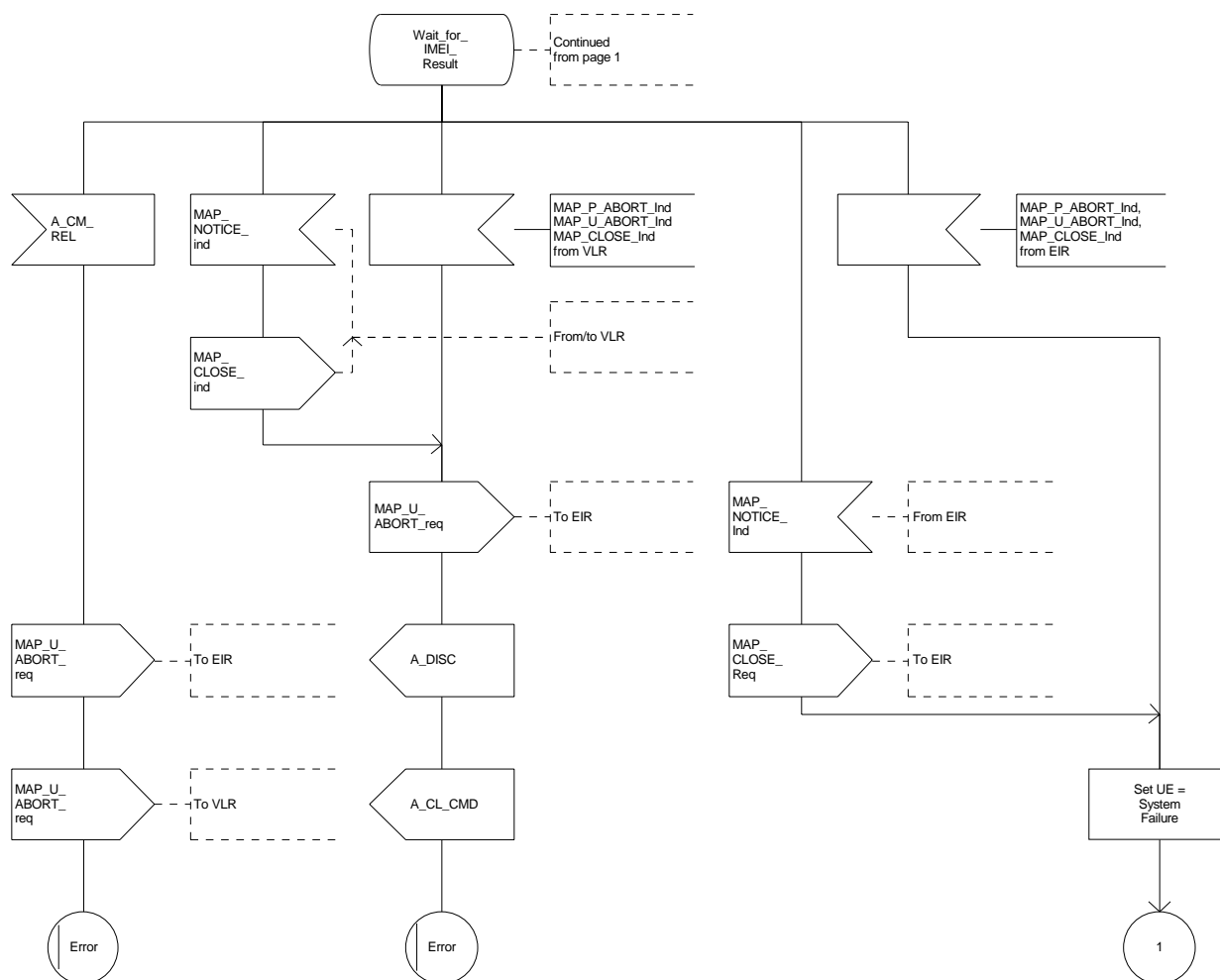


Figure 21.6/2

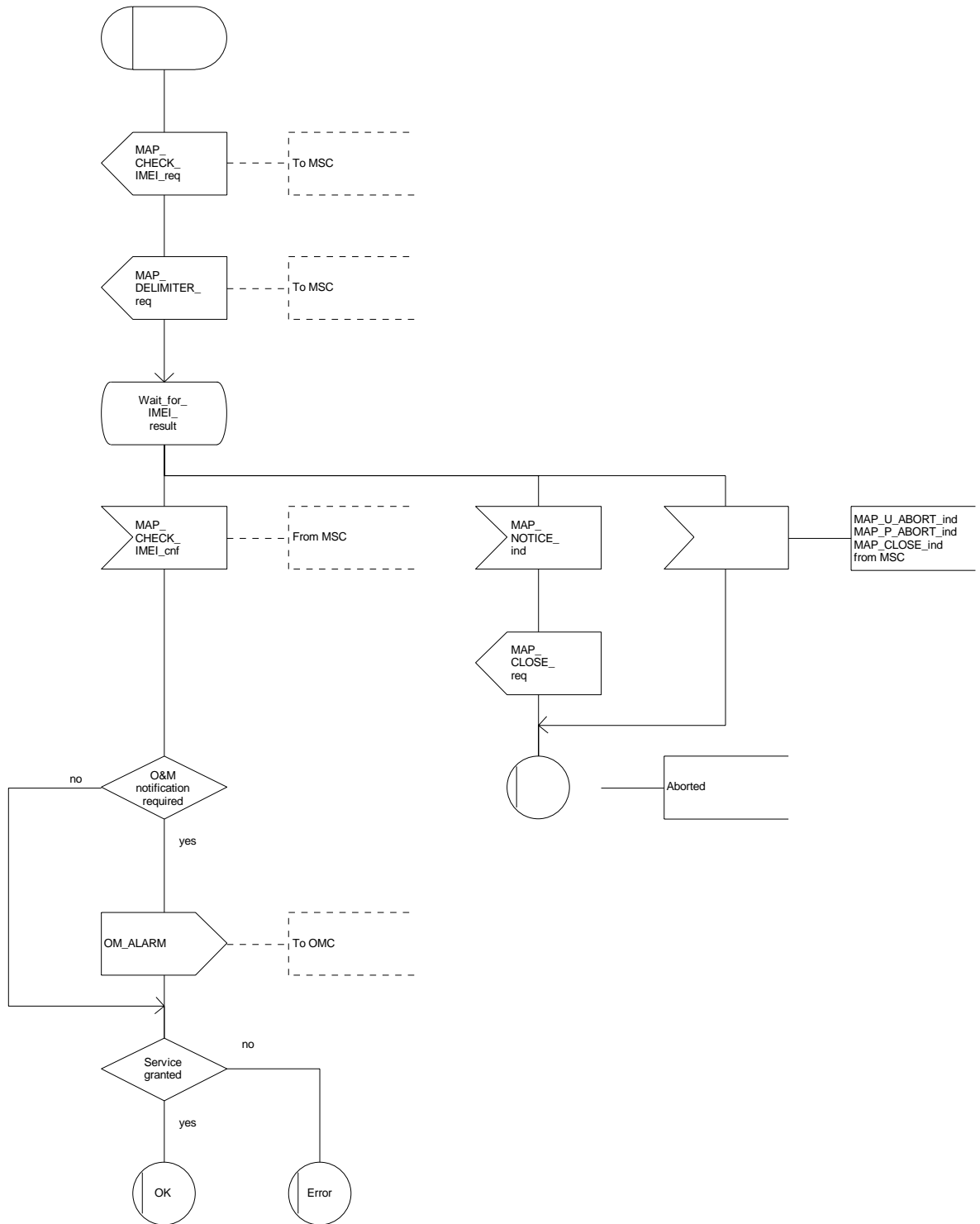


Figure 21.6/3

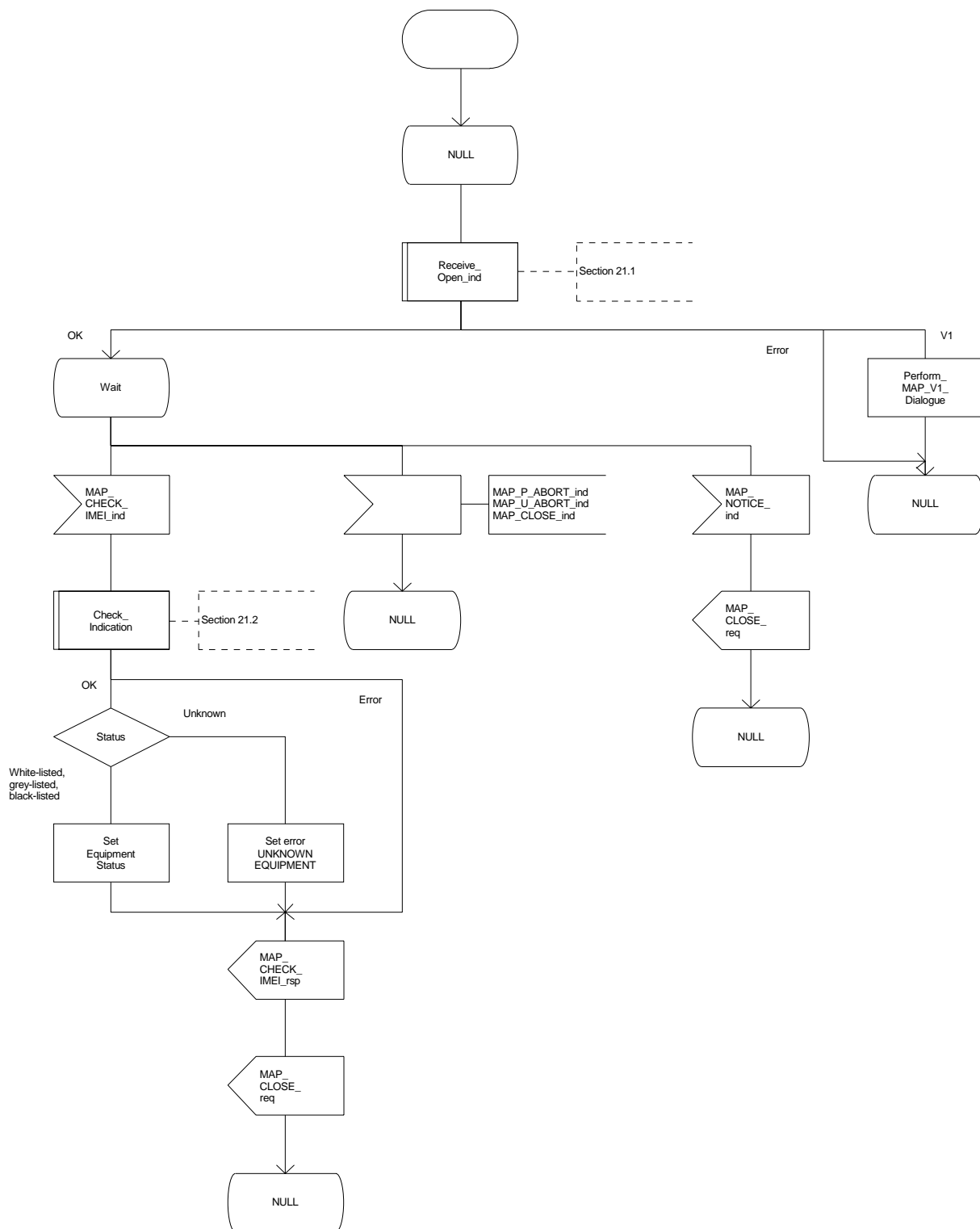


Figure 21.6/4

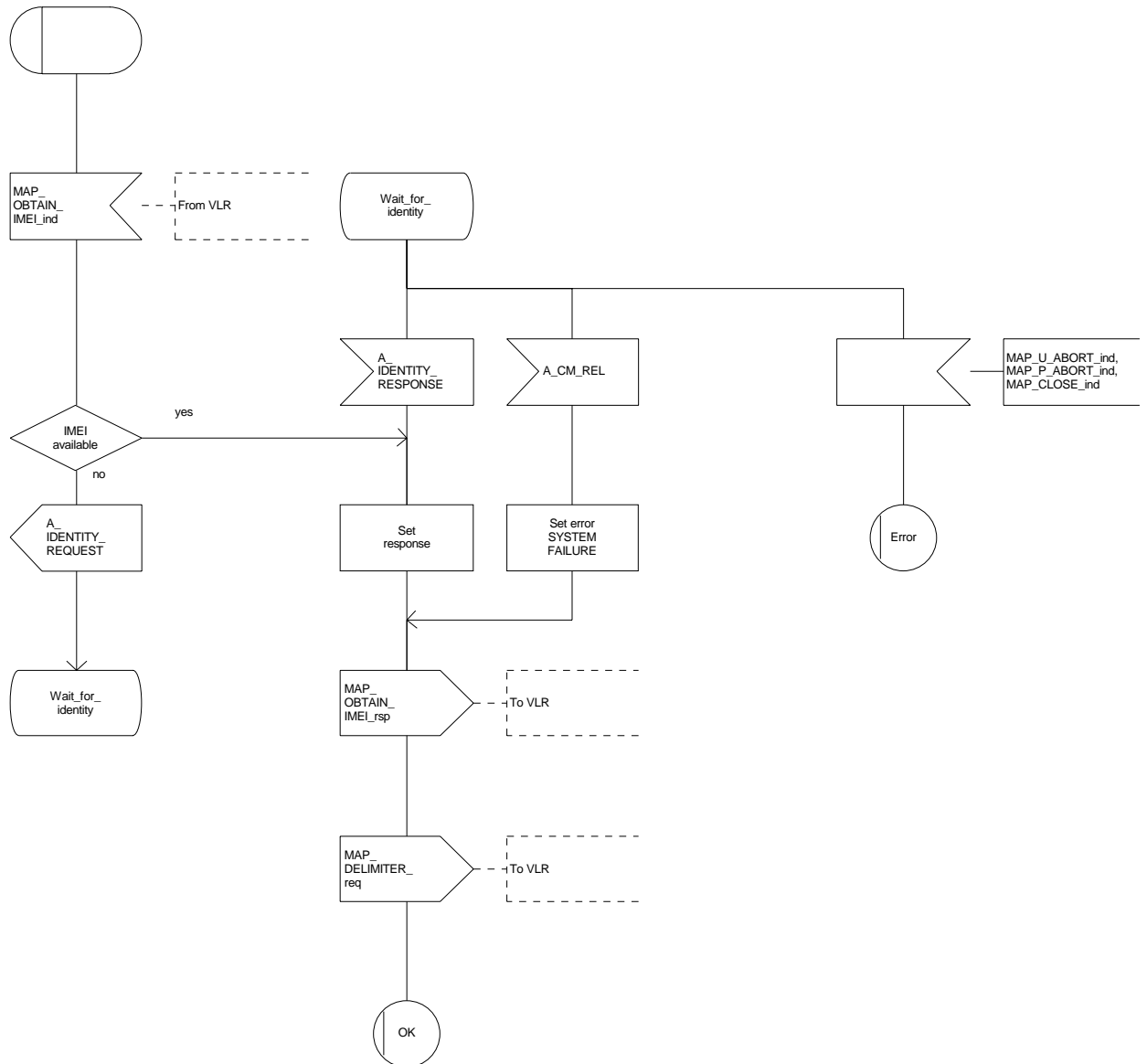
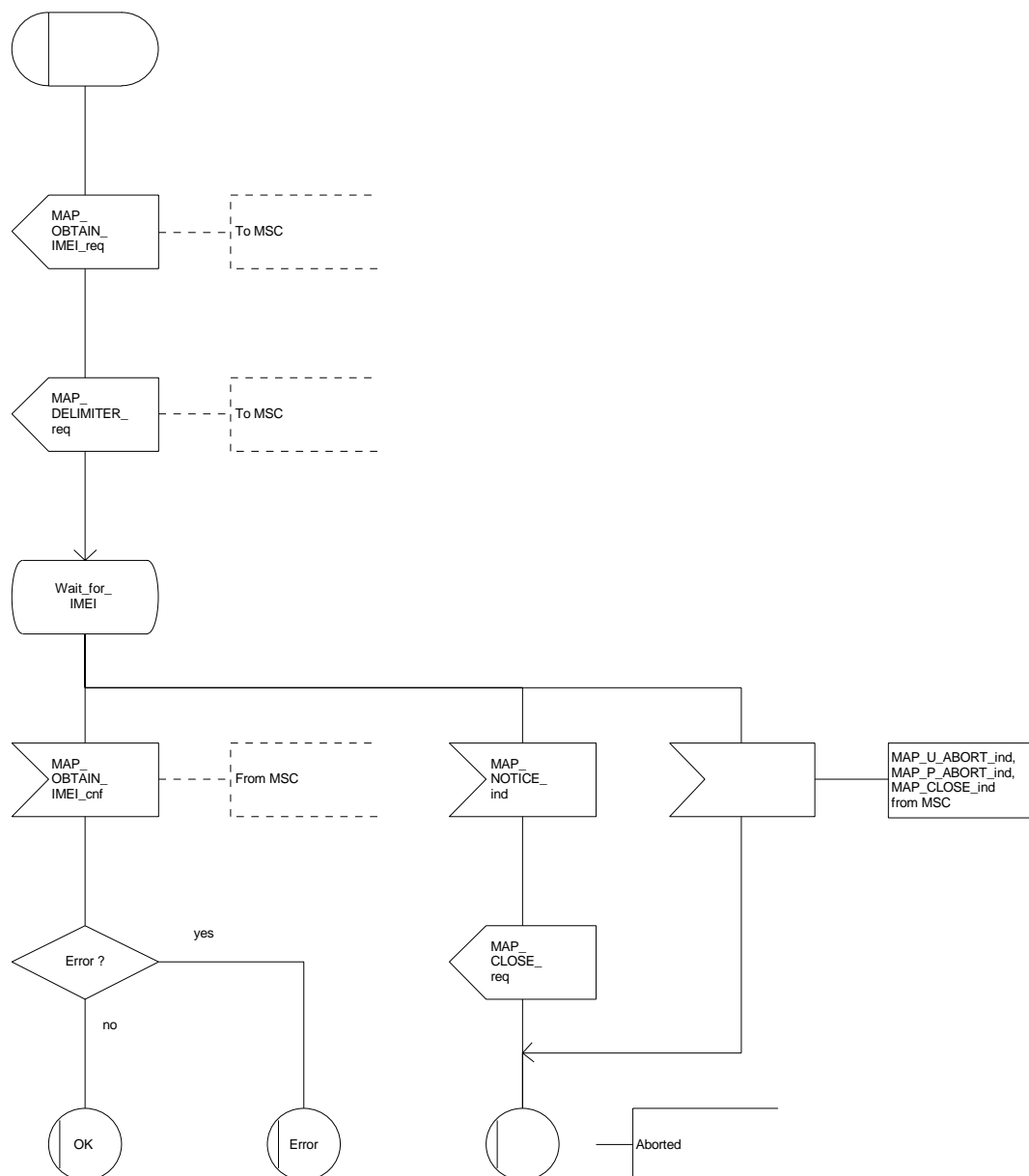


Figure 21.6/5



21.7 Insert Subscriber Data Macros

21.7.1 Macro Insert_SubData_VLR

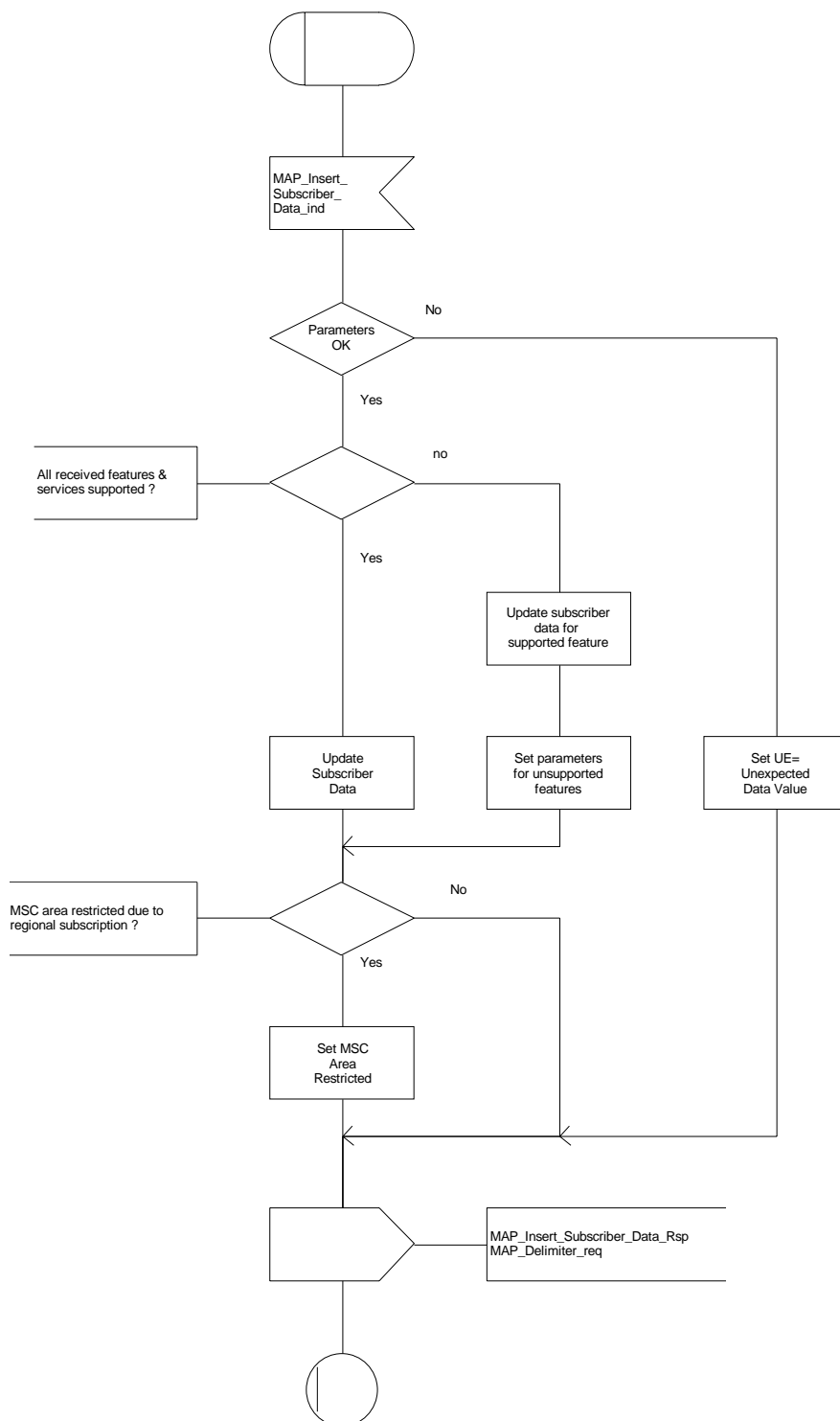
This macro describes the reception of the InsertSubscriberData service indication. This macro is used by any procedure that triggers the reception of subscriber data (e.g. Update Location or Restore Data).

If the VLR does not support any basic or supplementary service or the network feature Operator Determined Barring, or there is a problem with Regional Subscription Data then it reports it to the HLR.

If the entire MSC area is restricted due to regional subscription this is reported to the HLR.

The SDL diagram is shown in figure 21.7/1.

Figure 21.7/1



21.7.2 Process Insert_Subscriber_Data_Stand_Alone_HLR

This process is used by HLR to transfer subscriber data to VLR in a stand alone mode, i.e. in its own dialogue. This is done whenever a change of subscriber data is performed either by the operator or by the subscriber and this change has to be reported to VLR.

The process, after opening the dialogue with VLR, sends as many requests of the InsertSubscriberData service as necessary to transfer the subscriber data. The call to the process "Send_Insert_Subscriber_Data" (see subclause 21.7.4) is meant to describe two possible behaviours of the HLR when more than one service request has to be sent:

- either the HLR handles the requests and the confirmations in parallel; or
- the HLR sends every request after receiving the confirmation to the previous one.

The macro "Wait_for_Insert_Subscriber_Data_Cnf" (see subclause 21.7.3) is also called in order to handle every single confirmation.

If the result of a primitive received from the VLR is unsuccessful, the HLR may initiate re-attempts; the number of repeat attempts and the time in between are HLR operator options, depending on the error returned by the VLR.

If certain services required for a subscriber are not supported by the VLR (e.g. Advice of Charge Charging Level), this may result in one of the following outcomes:

- the HLR stores and sends "Roaming Restriction Due To Unsupported Feature" in a subsequent MAP_INSERT_SUBSCRIBER_DATA service. If "Roaming Restriction Due To Unsupported Feature" is stored in the HLR, the "MSC Area Restricted Flag" shall be set to "restricted". This will prevent MT calls, MT SM and MT USSD from being forwarded to the MSC/VLR.
- the HLR stores and sends other induced subscriber data (e.g. a specific barring program) in a subsequent MAP_INSERT_SUBSCRIBER_DATA service. This will cause rejection of mobile originated service requests, except emergency calls.

When the VLR receives regional subscription data (Zone Code List) it may respond with "MSC Area Restricted" in the MAP_INSERT_SUBSCRIBER_DATA response. In this case the "MSC Area Restricted Flag" shall be set to "restricted" in the HLR. This will prevent MT calls, MT SM and MT USSD from being forwarded to the MSC/VLR.

If the HLR does not store "Roaming Restriction Due To Unsupported Feature" as a consequence of the stand alone Insert Subscriber Data procedure and the HLR does not receive "MSC Area Restricted" in the MAP_INSERT_SUBSCRIBER_DATA response and "Roaming Restriction Due To Unsupported Feature" has not been stored in the HLR in the course of a previous subscriber data retrieval procedure, the "MSC Area Restricted Flag" in the HLR shall be set to "not restricted".

The SDL diagram is shown in figure 21.7/2.

Figure 21.7/2 (sheet 1 of 2)

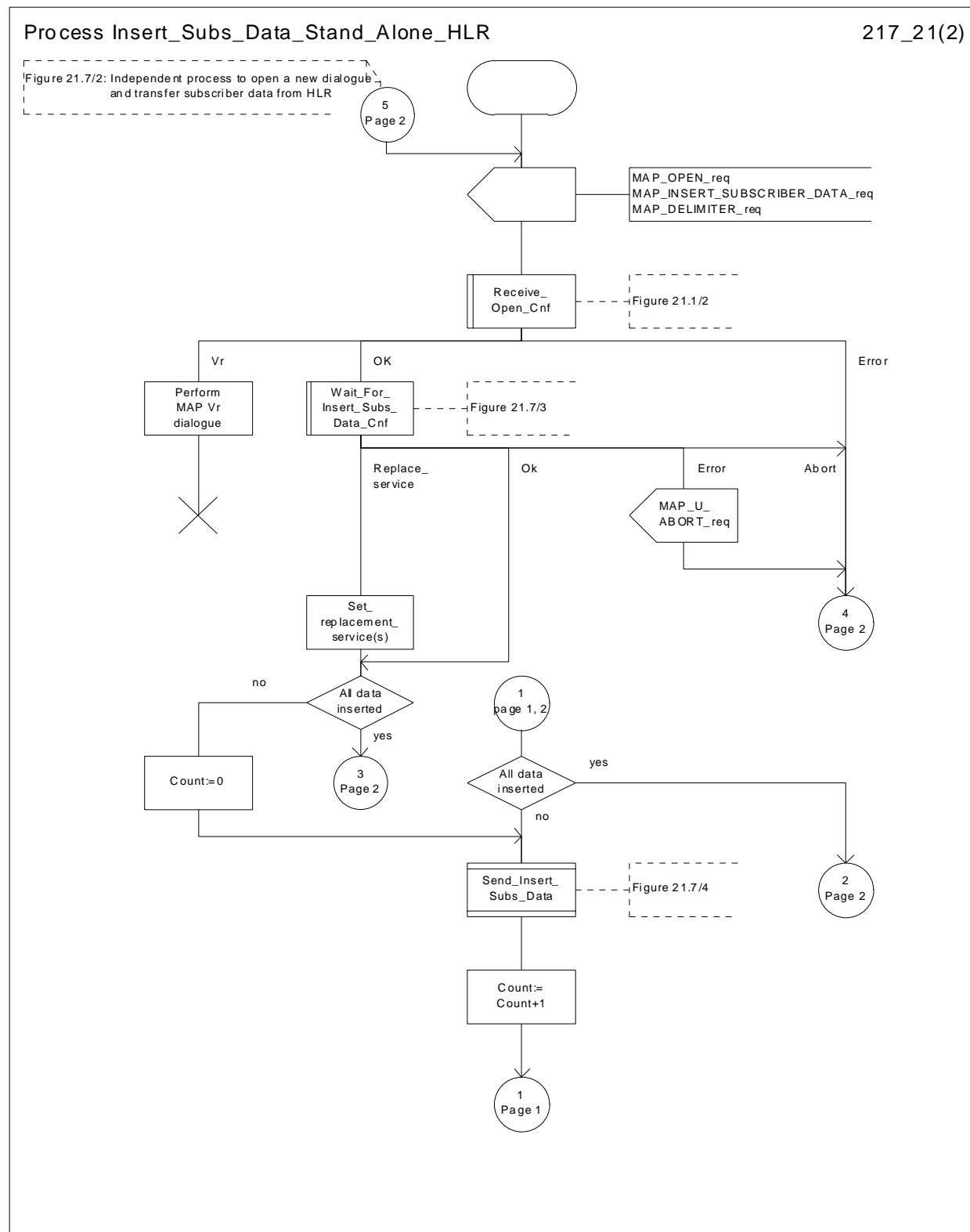
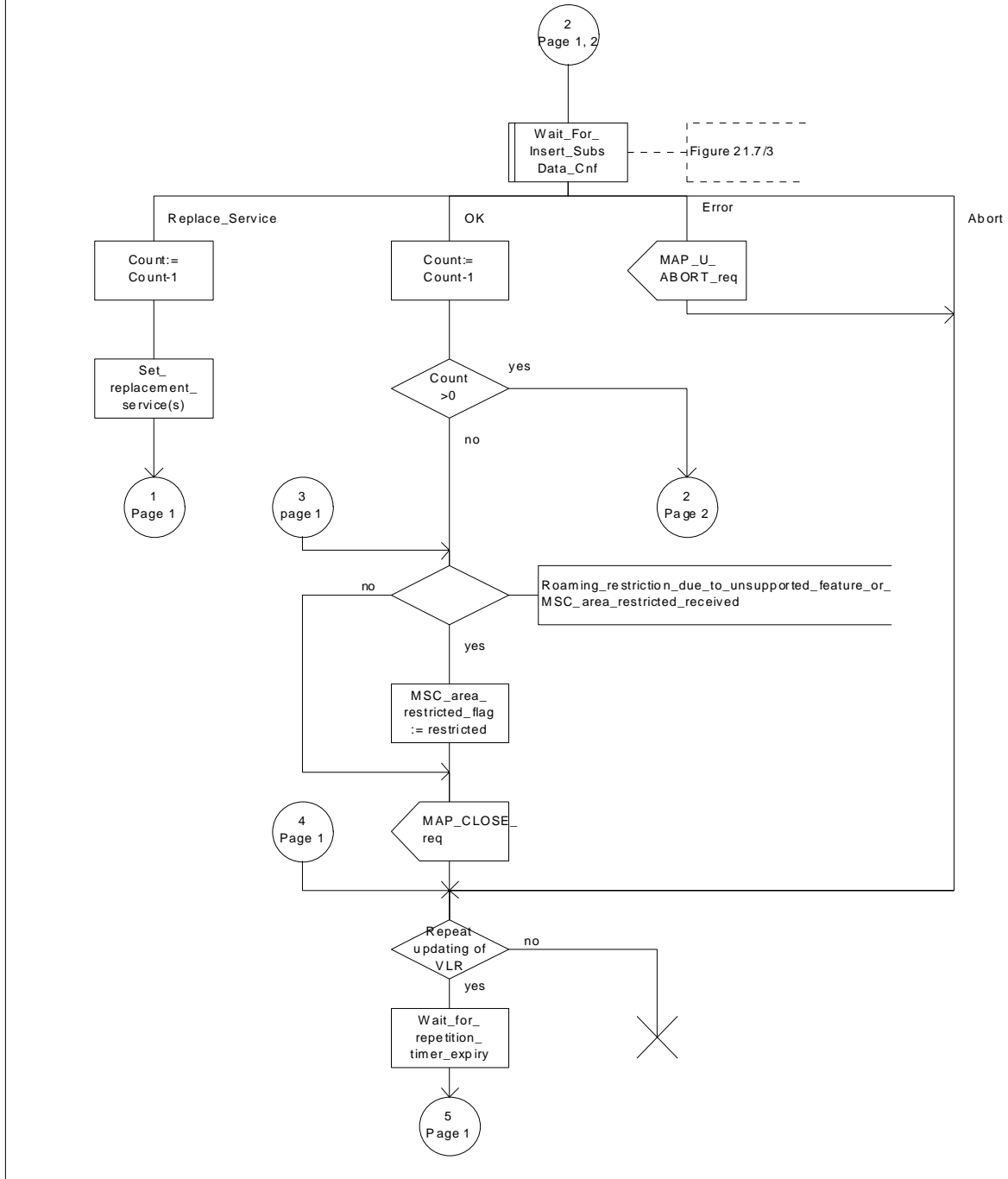


Figure 21.7/2 (sheet 2 of 2)

Process Insert_Subs_Data_Stand_Alone_HLR

217_22(2)

Figure 21.7/2: Independent process to open a new dialogue
and transfer subscriber data from HLR to VLR



21.7.3 Macro Wait_for_Insert_Subs_Data_Cnf

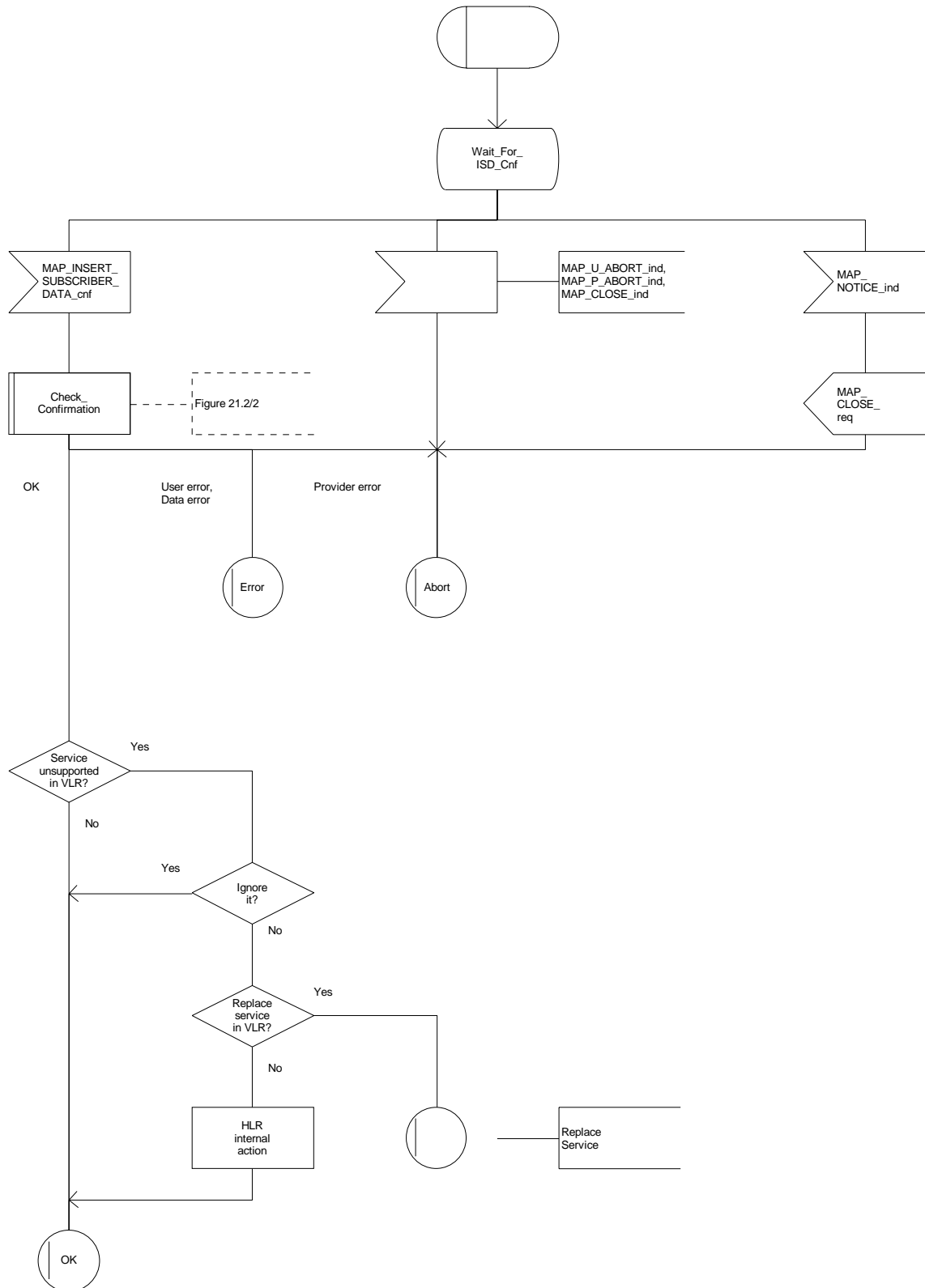
This macro is used by any process or macro that describes the handling of the reception of the Insert_Subscriber_Data service in HLR (e.g. Update Location or Restore Data).

If the VLR reports the non-support of some basic or supplementary service or the network feature Operator Determined Barring then three actions are possible:

- to ignore the information received;
- to replace the not supported service;
- or to perform any other internal action.

The SDL diagram is shown in figure 21.7/3.

Figure 21.7/3

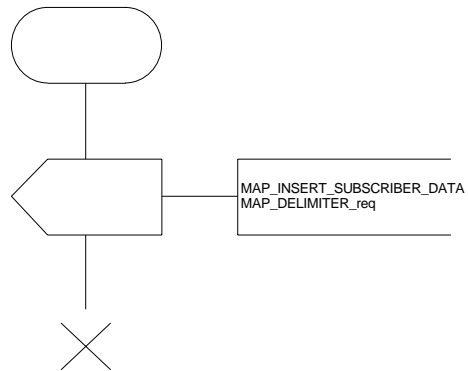


21.7.4 Process Send_Insert_Subs_Data

This process is used by any process or macro where the Insert_Subscriber_Data request is sent to VLR.

The SDL diagram is shown in figure 21.7/4.

Figure 21.7/4



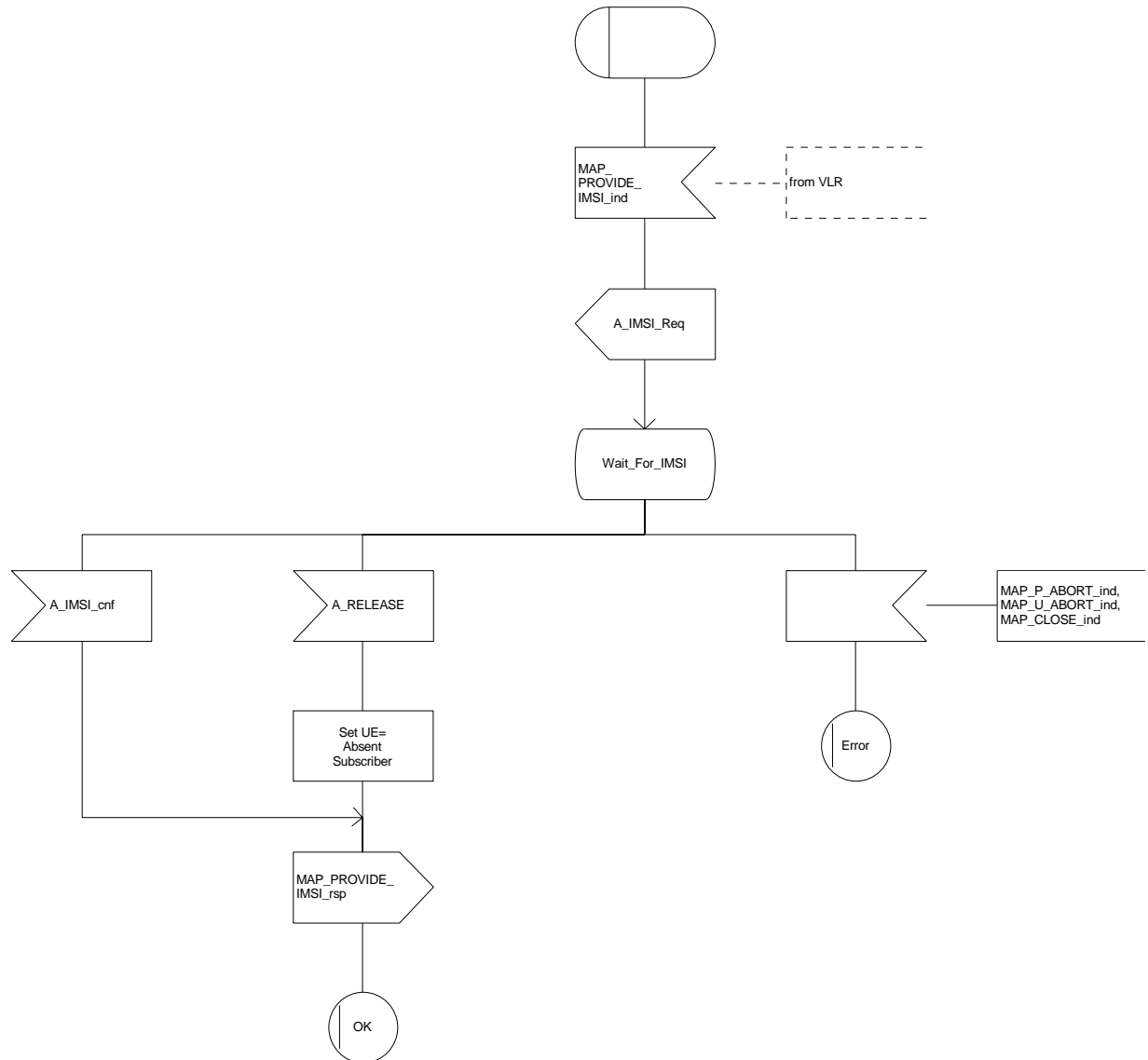
21.8 Request IMSI Macros

21.8.1 Macro Obtain_IMSI_MSC

This macro describes the handling of the request received from the VLR to provide the IMSI of a subscriber (e.g. at Location Updating).

The SDL diagram is shown in figure 21.8/1.

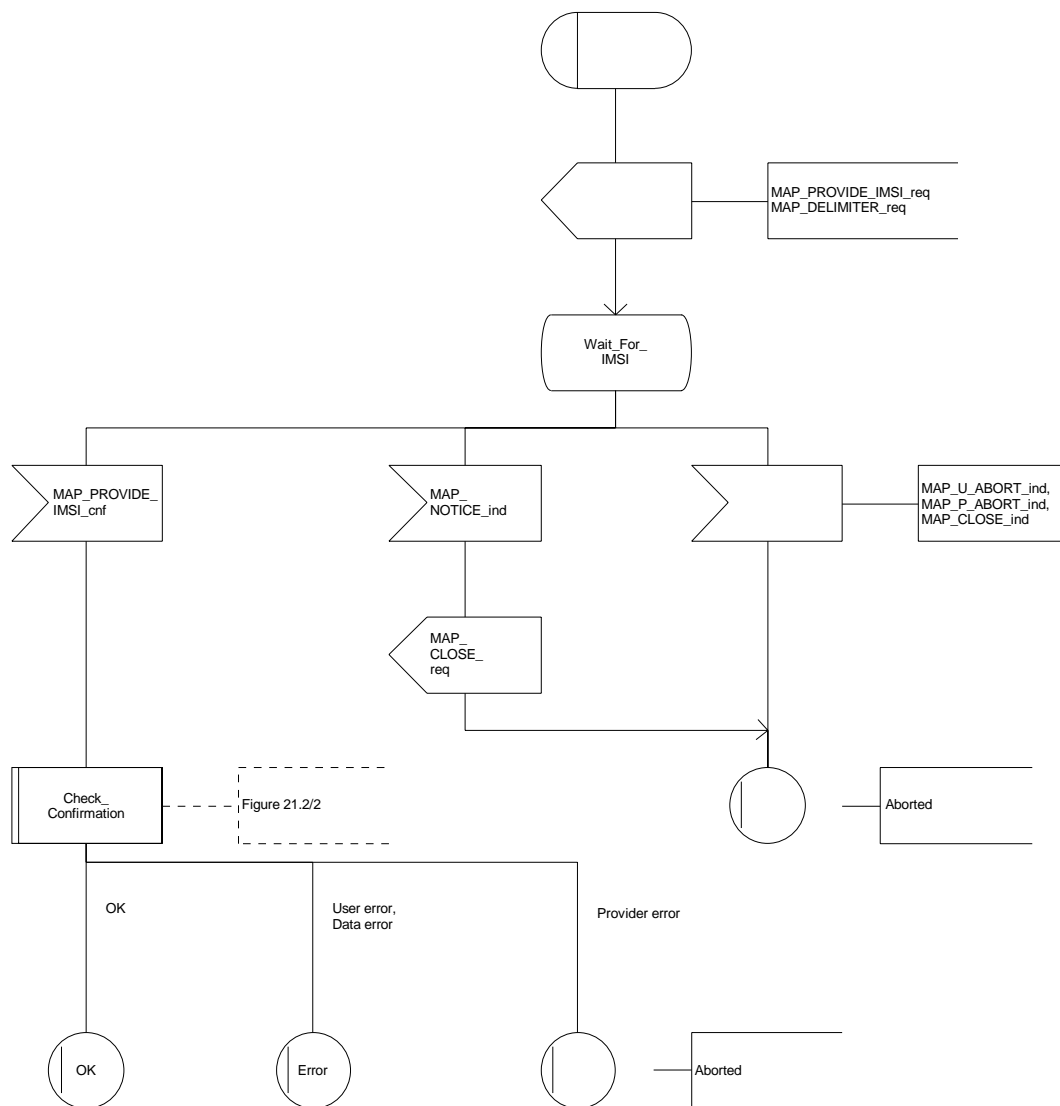
Figure 21.8/1



21.8.2 Macro Obtain_IMSI_VLR

This macro describes the way VLR requests the MSC the IMSI of a subscriber (e.g. at Location Updating).

The SDL diagram is shown in figure 21.8/2.

Figure 21.8/2

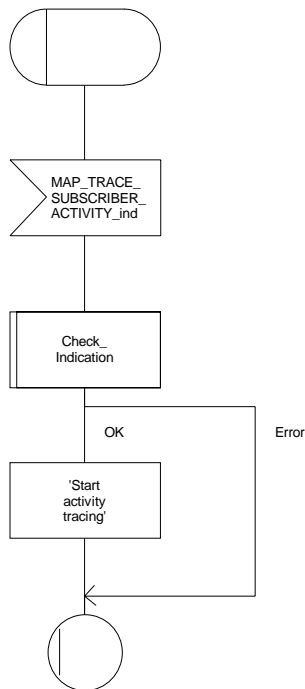
21.9 Tracing macros

21.9.1 Macro Trace_Subscriber_Activity_MSC

The Trace_Subscriber_Activity_MSC is invoked in the MSC, when the MSC receives the MAP_TRACE_SUBSCRIBER_ACTIVITY indication from the VLR. The data of the primitive is checked and the tracing in the MSC is started if the content includes no errors. No response is returned to the VLR.

The Trace_Subscriber_Activity_MSC macro is described in the figure 21.9/1.

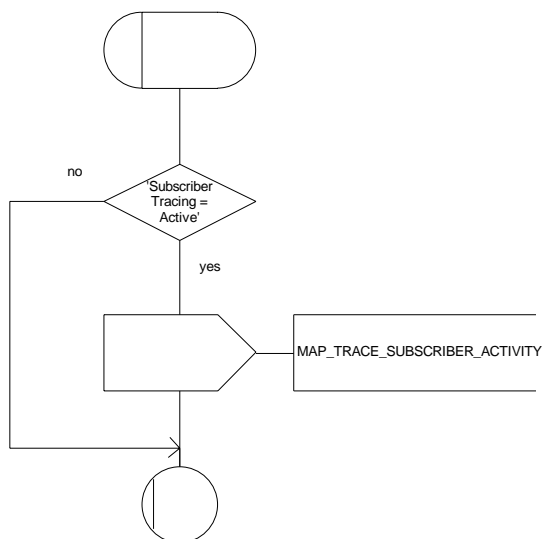
Figure 21.9/1



21.9.2 Macro Trace_Subscriber_Activity_VLR

The macro Trace_Subscriber_Activity_VLR is invoked, if the subscriber activity is detected by the VLR and the tracing is active. The VLR sends MAP_TRACE_SUBSCRIBER_ACTIVITY request to the MSC. No answer is awaited from the MSC.

The Trace_Subscriber_Activity_VLR macro is shown in the figure 21.9/2.

Figure 21.9/2

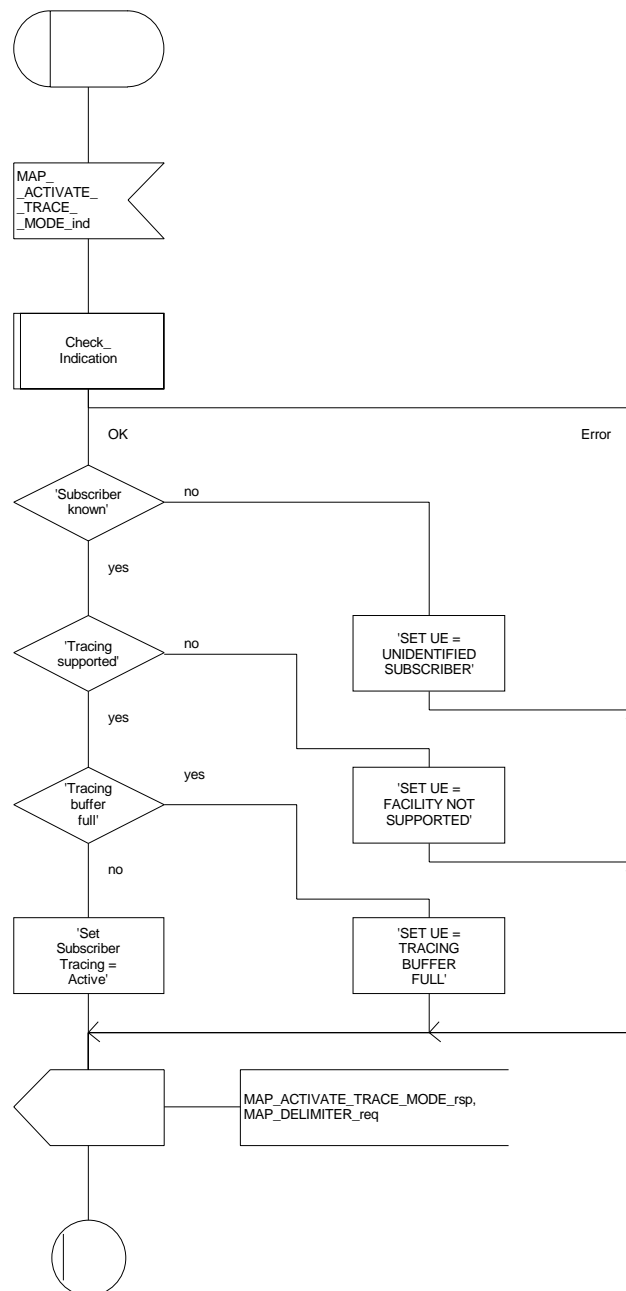
21.9.3 Macro Activate_Tracing_VLR

The Activate_Tracing_VLR macro is invoked, when the MAP_ACTIVATE_TRACE_MODE indication is received from the HLR. The primitive is processed in the VLR as follows:

- if the data contains errors, a data missing or unexpected data value indication is returned to the HLR;
- if the tracing is not supported, a facility not supported indication is returned to the HLR;
- if the tracing buffer does not have any space left for the data, a tracing buffer full indication is returned to the HLR;
- if no errors is detected, the tracing is set active and a positive acknowledge is returned to the HLR.

The Activate_Tracing_VLR macro is described in the figure 21.9/3.

Figure 21.9/3



21.9.4 Macro Control_Tracing_HLR

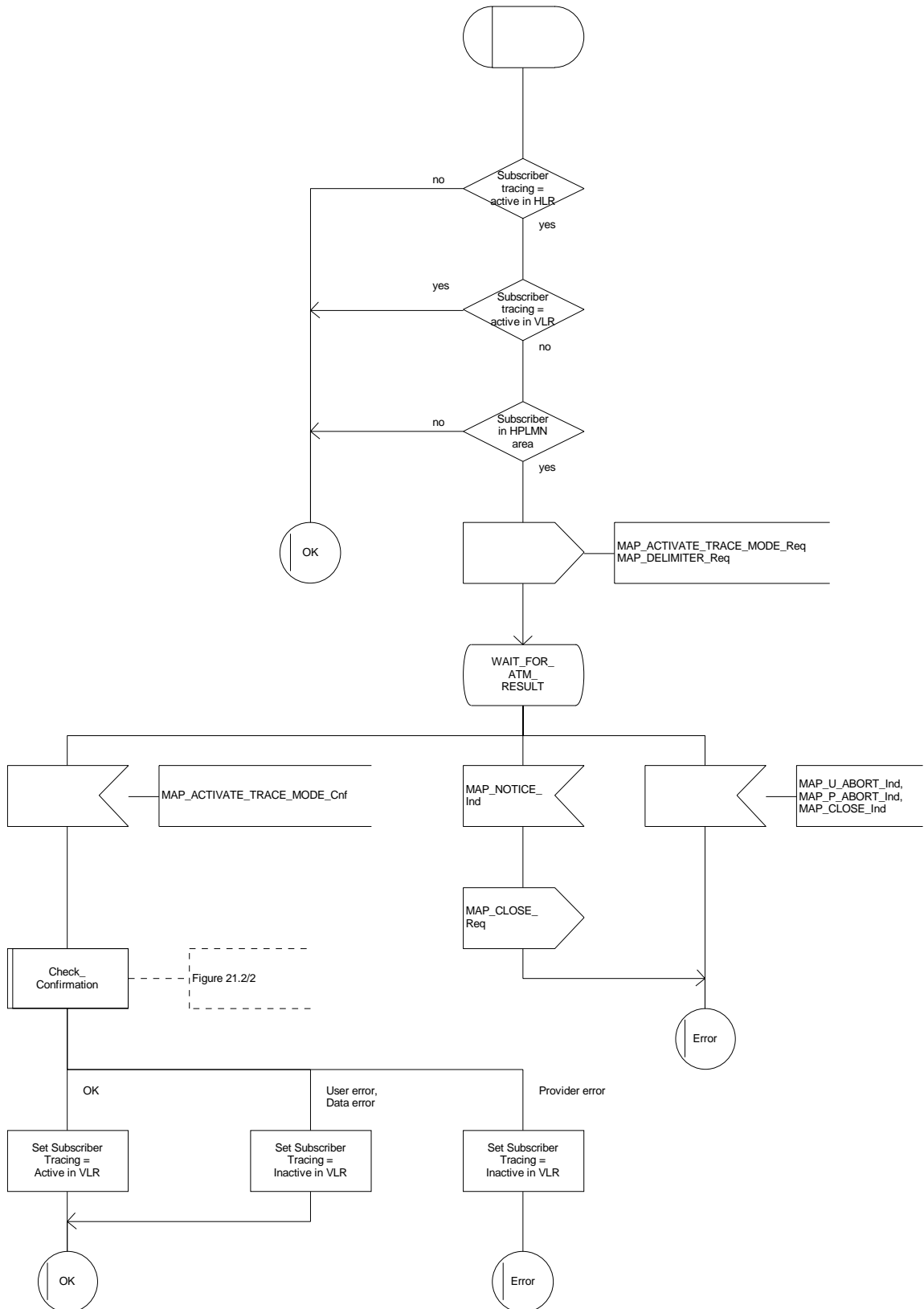
The Control_Tracing_HLR macro may be invoked in the HLR, if subscriber related activity is detected. If the tracing is active in the HLR and not active in the VLR, the MAP_ACTIVATE_TRACE_MODE request is sent to the VLR.

The MAP_ACTIVATE_TRACE_MODE confirmation from the VLR is processed as follows:

- if the primitive contains a successful acknowledge, the tracing in VLR is set active;
- if the primitive contains errors, the tracing in VLR is set deactive.

The Control_Tracing_HLR macro is shown in the figure 21.9/4.

Figure 21.9/4



21.10 Short Message Alert procedures

21.10.1 Subscriber_Present_VLR process

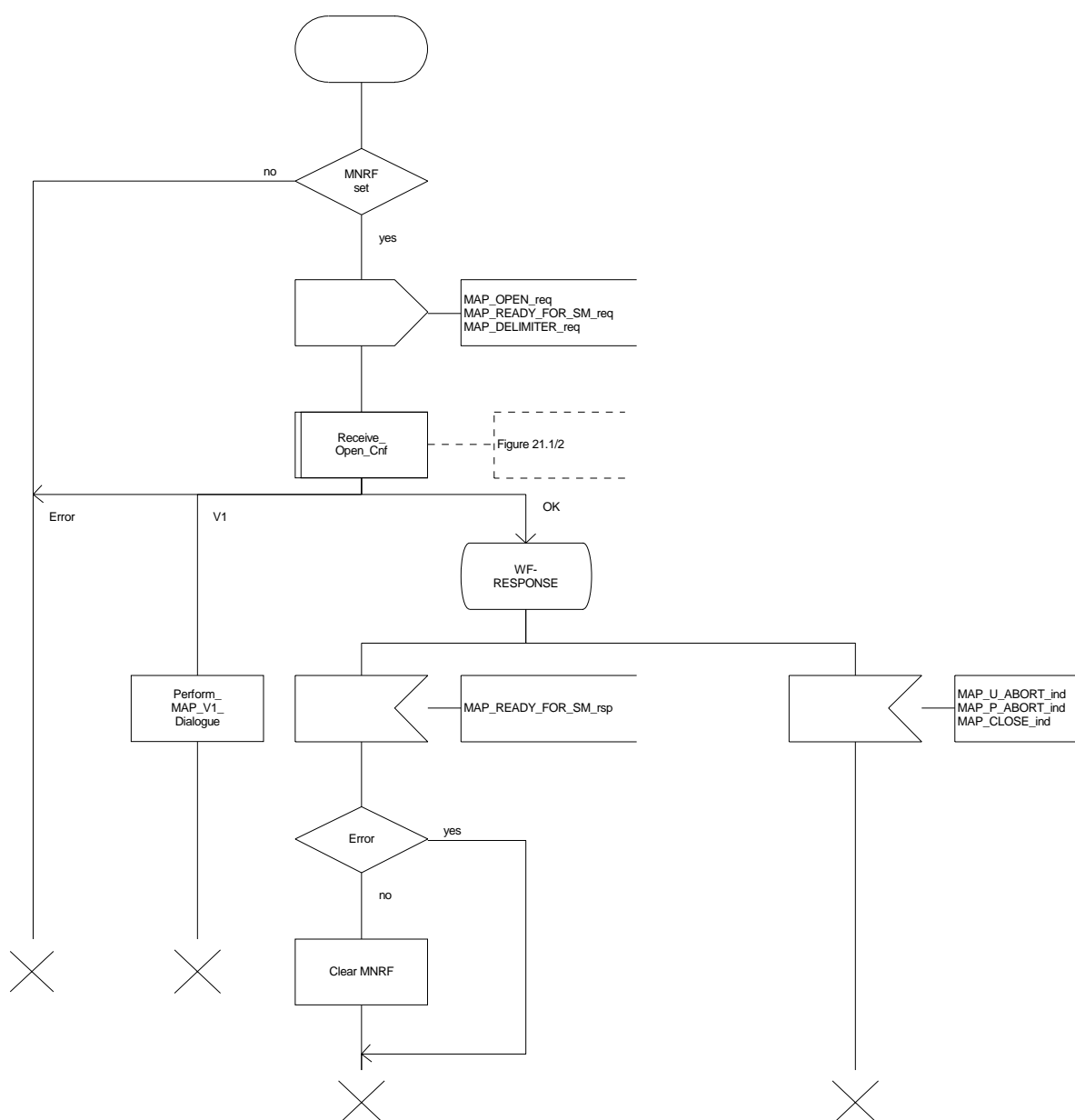
The Subscriber_Present_VLR process is invoked by the VLR, when the mobile subscriber becomes active and the MNRF flag is set. The general description of the short message alert procedures is in the subclause 20.4.

The VLR sends the MAP_READY_FOR_SM request to the HLR and waits for the HLR to answer. When receiving the answer, the VLR will act as follows:

- the MNRF flag is cleared if the procedure is successful;
- the MNRF flag is not cleared if the procedure is not successful.

The Subscriber_Present_VLR process is shown in the figure 21.10/1.

Figure 21.10/1



21.10.2 Macro Alert_Service_Centre_HLR

The Alert_Service_Centre_HLR macro is initiated when the HLR notices that the Service Centre(s) shall be alerted. The macro starts process Alert_Service_Centre_HLR for every SC address in the MWD list.

In the process Alert_Service_Centre_HLR the HLR sends MAP_ALERT_SERVICE_CENTRE request to the appropriate IWMSC. The MWD entry is deleted when the positive acknowledge is received from the IWMSC. The unsuccessful alert may be repeated. The MWD entry should be purged in the unsuccessful case, at least when a suitable time period has expired.

The Alert_Service_Centre_HLR macro is shown in the figure 21.10/2 and the Alert_Service_Centre_HLR process is shown in the figure 21.10/3.

Figure 21.10/2

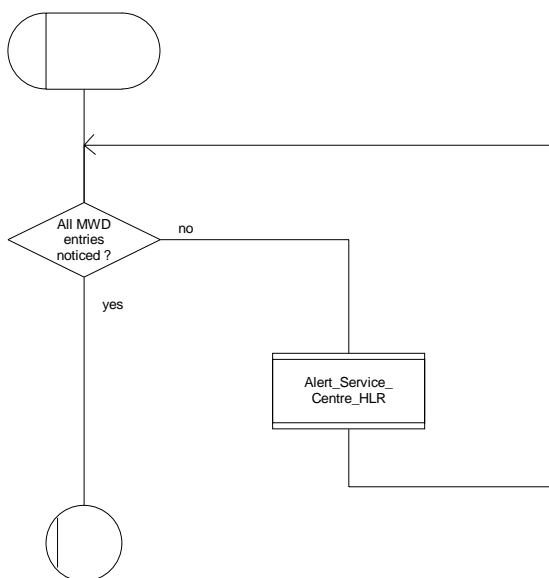
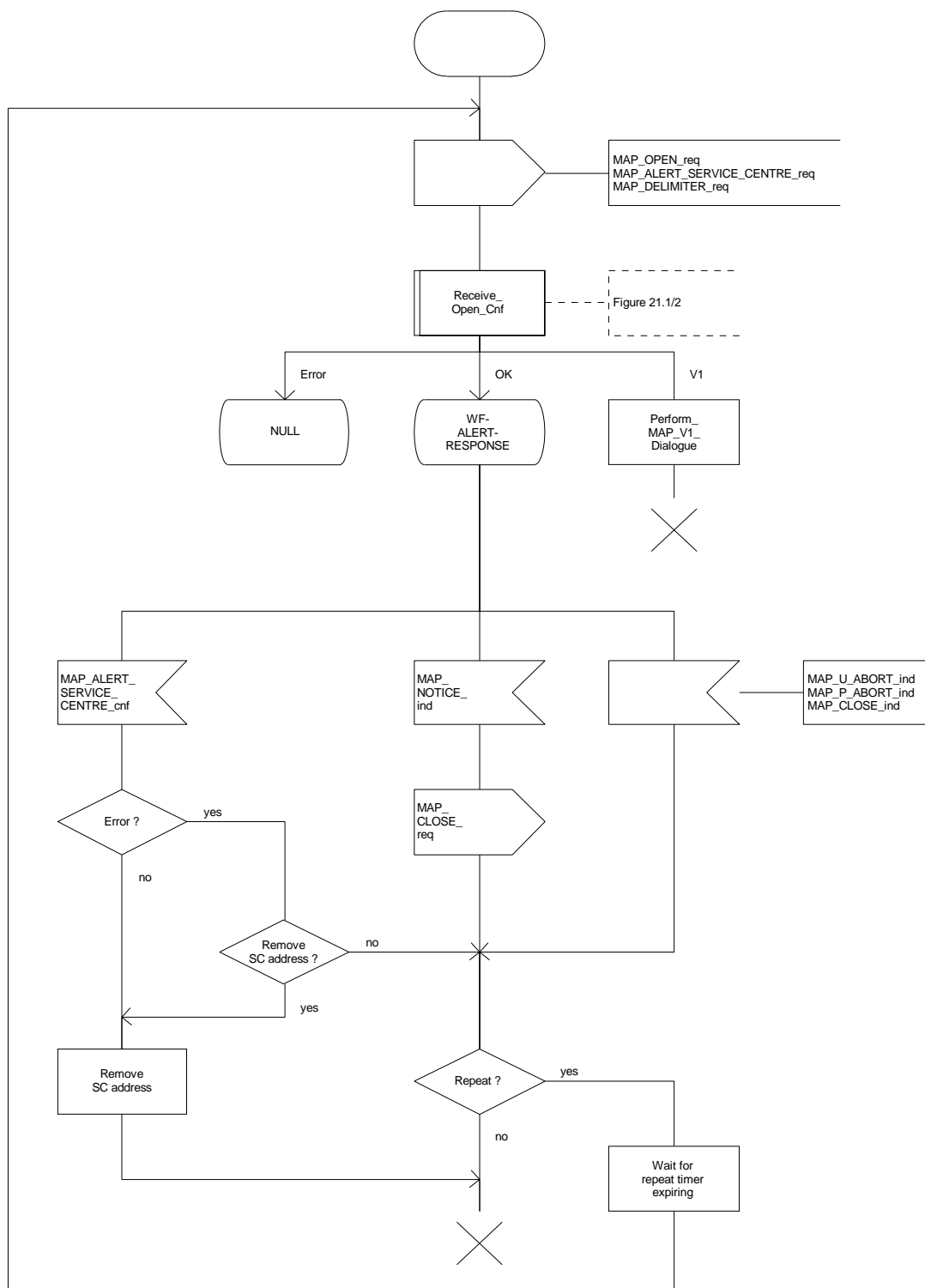


Figure 21.10/3



Annex A (informative): Cross-reference for abstract syntaxes of MAP

Annex A is not part of the standard, it is included for information purposes only.

For every ASN.1 item such as identifier, type-reference or value-reference the cross-reference allows to locate all occurrences by means of module-name and line numbers. For that purpose line numbers are printed at the left margin in front of each ASN.1 source line starting with 1 for every module.

The items are sorted alphabetically in the cross-reference in a case-insensitive manner. Occurrences of an item are its definition and all its usages such as in exports, imports or within a type or value assignment.

For every item additional information is provided such as kind of item (identifier, value reference, type reference), and tag, associated type and value if applicable.

The cross-reference for a root module includes all modules referred to directly or indirectly via imports. The cross-references for the root modules MAP-Protocol/TCAPMessages and MAP-DialoguePDU are included.

TAG P3.90M Cross Reference Listing for MAP-Protocol 98-07-23 09:08:55 1

```

&extensionId.....identifier of Fieldspec
  DEFINED in MAP-ExtensionDataTypes : 24
  USED in MAP-ExtensionDataTypes : 41

&ExtensionType.....identifier of Fieldspec
  DEFINED in MAP-ExtensionDataTypes : 23
  USED in MAP-ExtensionDataTypes : 43

abort.....identifier of [APPLICATION 7] IMPLICIT Abort
  DEFINED in TCAPMessages : 56

Abort.....type reference SEQUENCE
  DEFINED in TCAPMessages : 74
  USED in TCAPMessages : 56

absentSubscriber.....value reference AbsentSubscriber, CHOICE VALUE
  DEFINED in MAP-Protocol : 259

AbsentSubscriber.....type reference ERROR
  DEFINED in MAP-Errors : 222
  USED in MAP-Protocol : 98 259
  USED in MAP-CallHandlingOperat : 33 70 88
  USED in MAP-SupplementaryServi : 46 184 198
  USED in MAP-ShortMessageServic : 36
  USED in MAP-Errors : 45

absentSubscriber.....identifier of Named Number, 1
  DEFINED in MAP-SM-DataTypes : 117

absentSubscriberDiagnosticSM.....identifier of [0] AbsentSubscriberDiagnosticSM
  DEFINED in MAP-SM-DataTypes : 110

absentSubscriberDiagnosticSM.....identifier of AbsentSubscriberDiagnosticSM
  DEFINED in MAP-ER-DataTypes : 135

AbsentSubscriberDiagnosticSM.....type reference INTEGER
  DEFINED in MAP-ER-DataTypes : 139
  USED in MAP-SM-DataTypes : 39 110
  USED in MAP-ER-DataTypes : 43 135

absentSubscriberParam.....identifier of AbsentSubscriberParam
  DEFINED in MAP-Errors : 224

AbsentSubscriberParam.....type reference SEQUENCE
  DEFINED in MAP-ER-DataTypes : 206
  USED in MAP-Errors : 101 224
  USED in MAP-ER-DataTypes : 34

absentsubscriberSM.....value reference AbsentSubscriberSM, CHOICE VALUE
  DEFINED in MAP-Protocol : 293

AbsentSubscriberSM.....type reference ERROR
  DEFINED in MAP-Errors : 319
  USED in MAP-Protocol : 119 293
  USED in MAP-ShortMessageServic : 41 79 109
  
```

```

        USED in MAP-Errors                :      70

absentSubscriberSM-Param.....identifier of AbsentSubscriberSM-Param
    DEFINED in MAP-Errors                :      321

AbsentSubscriberSM-Param.....type reference SEQUENCE
    DEFINED in MAP-ER-DataTypes          :      134
    USED in MAP-Errors                  :      111      321
    USED in MAP-ER-DataTypes            :      42

activateSS.....value reference ActivateSS, CHOICE VALUE
    DEFINED in MAP-Protocol              :      186

ActivateSS.....type reference OPERATION
    DEFINED in MAP-SupplementaryServi :      108
    USED in MAP-Protocol              :      54      186
    USED in MAP-SupplementaryServi :      15

activateTraceMode.....value reference ActivateTraceMode, CHOICE VALUE
    DEFINED in MAP-Protocol              :      170

ActivateTraceMode.....type reference OPERATION
    DEFINED in MAP-OperationAndMainte :      50
    USED in MAP-Protocol              :      36      170
    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

AddressString.....type reference OCTET STRING
    DEFINED in MAP-CommonDataTypes       :      73
    USED in MAP-CommonDataTypes          :      16      117
    USED in MAP-OM-DataTypes             :      21      40
    USED in MAP-SS-DataTypes             :      37      55
    USED in MAP-SM-DataTypes             :      30      54      98      103      108      128

ageOfLocationInformation.....identifier of AgeOfLocationInformation
    DEFINED in MAP-MS-DataTypes          :      632

AgeOfLocationInformation.....type reference INTEGER
    DEFINED in MAP-MS-DataTypes          :      640
    USED in MAP-MS-DataTypes            :      632

alertReason.....identifier of AlertReason
    DEFINED in MAP-SM-DataTypes          :      146

AlertReason.....type reference ENUMERATED
    DEFINED in MAP-SM-DataTypes          :      149
    USED in MAP-SM-DataTypes            :      26      146

alertServiceCentre.....value reference AlertServiceCentre, CHOICE VALUE
    DEFINED in MAP-Protocol              :      204

AlertServiceCentre.....type reference OPERATION
    DEFINED in MAP-ShortMessageService :      123
    USED in MAP-Protocol              :      71      204
    USED in MAP-ShortMessageService :      17

alertServiceCentreArg.....identifier of AlertServiceCentreArg
    DEFINED in MAP-ShortMessageService :      125

AlertServiceCentreArg.....type reference SEQUENCE
    DEFINED in MAP-SM-DataTypes          :      126
    USED in MAP-ShortMessageService :      54      125
    USED in MAP-SM-DataTypes            :      22

allAdditionalInfoTransferSS.....value reference SS-Code, '10000000'B
    DEFINED in MAP-SS-Code              :      94

allAlternateSpeech-DataCDA.....value reference BearerServiceCode, '00110000'B
    DEFINED in MAP-BS-Code              :      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	: 101
allBearerServices.....value	reference BearerServiceCode, '00000000'B
DEFINED in MAP-BS-Code	: 49
allCallCompletionSS.....value	reference SS-Code, '01000000'B
DEFINED in MAP-SS-Code	: 63
allCallOfferingSS.....value	reference SS-Code, '00110000'B
DEFINED in MAP-SS-Code	: 54
allCallPrioritySS.....value	reference SS-Code, '10100000'B
DEFINED in MAP-SS-Code	: 137
allChargingSS.....value	reference SS-Code, '01110000'B
DEFINED in MAP-SS-Code	: 86
allCommunityOfInterest-SS.....value	reference SS-Code, '01100000'B
DEFINED in MAP-SS-Code	: 80
allCondForwardingSS.....value	reference SS-Code, '00101000'B
DEFINED in MAP-SS-Code	: 45
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	: 262
allFacsimileTransmissionServices.....value	reference TeleserviceCode, '01100000'B
DEFINED in MAP-TS-Code	: 48
allForwardingSS.....value	reference SS-Code, '00100000'B
DEFINED in MAP-SS-Code	: 41
allLineIdentificationSS.....value	reference SS-Code, '00010000'B
DEFINED in MAP-SS-Code	: 25
allMultiPartySS.....value	reference SS-Code, '01010000'B
DEFINED in MAP-SS-Code	: 74
allOG-CallsBarred.....identifier of Named Number, 0	
DEFINED in MAP-MS-DataTypes	: 253
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	: 120
allPLMN-specificTS.....value	reference TeleserviceCode, '11010000'B
DEFINED in MAP-TS-Code	: 72
allShortMessageServices.....value	reference TeleserviceCode, '00100000'B
DEFINED in MAP-TS-Code	: 44
allSpeechFollowedByDataCDA.....value	reference BearerServiceCode, '01000000'B
DEFINED in MAP-BS-Code	: 86
allSpeechFollowedByDataCDS.....value	reference BearerServiceCode, '01001000'B

```

DEFINED in MAP-BS-Code           :      88

allSpeechTransmissionServices.....value reference TeleserviceCode, '00010000'B
DEFINED in MAP-TS-Code           :      40

allSS.....value reference SS-Code, '00000000'B
DEFINED in MAP-SS-Code           :      21

allSynchronousServices.....value reference BearerServiceCode, '01101000'B
DEFINED in MAP-BS-Code           :     101

allTeleservices.....value reference TeleserviceCode, '00000000'B
DEFINED in MAP-TS-Code           :      38

allTeleservices-ExeptSMS.....value reference TeleserviceCode, '10000000'B
DEFINED in MAP-TS-Code           :      58

allVoiceGroupCallServices.....value reference TeleserviceCode, '10010000'B
DEFINED in MAP-TS-Code           :      67

anyTimeInterrogation.....value reference AnyTimeInterrogation, CHOICE VALUE
DEFINED in MAP-Protocol           :     213

AnyTimeInterrogation.....type reference OPERATION
DEFINED in MAP-MobileServiceOpera :     154
USED in MAP-Protocol              :      29   213
USED in MAP-MobileServiceOpera    :      24

anyTimeInterrogationArg.....identifier of AnyTimeInterrogationArg
DEFINED in MAP-MobileServiceOpera :     156

AnyTimeInterrogationArg.....type reference SEQUENCE
DEFINED in MAP-MS-DataTypes        :     676
USED in MAP-MobileServiceOpera    :      89   156
USED in MAP-MS-DataTypes          :      67

anyTimeInterrogationRes.....identifier of AnyTimeInterrogationRes
DEFINED in MAP-MobileServiceOpera :     158

AnyTimeInterrogationRes.....type reference SEQUENCE
DEFINED in MAP-MS-DataTypes        :     683
USED in MAP-MobileServiceOpera    :      90   158
USED in MAP-MS-DataTypes          :      68

aocc.....value reference SS-Code, '01110010'B
DEFINED in MAP-SS-Code           :      91

aoci.....value reference SS-Code, '01110001'B
DEFINED in MAP-SS-Code           :      89

assumedIdle.....identifier of [0] NULL
DEFINED in MAP-MS-DataTypes        :     663

ati-NotAllowed.....value reference ATI-NotAllowed, CHOICE VALUE
DEFINED in MAP-Protocol           :     270

ATI-NotAllowed.....type reference ERROR
DEFINED in MAP-Errors              :     265
USED in MAP-Protocol              :     105   270
USED in MAP-MobileServiceOpera    :      62   161
USED in MAP-Errors                :      52

ati-NotAllowedParam.....identifier of ATI-NotAllowedParam
DEFINED in MAP-Errors              :     267

ATI-NotAllowedParam.....type reference SEQUENCE
DEFINED in MAP-ER-DataTypes        :     226
USED in MAP-Errors                :     108   267
USED in MAP-ER-DataTypes          :      39

AuthenticationSet.....type reference SEQUENCE
DEFINED in MAP-MS-DataTypes        :     156
USED in MAP-MS-DataTypes          :     154

authenticationSetList.....identifier of AuthenticationSetList
DEFINED in MAP-MS-DataTypes        :     150

AuthenticationSetList.....type reference SEQUENCE OF
DEFINED in MAP-MS-DataTypes        :     153
USED in MAP-MS-DataTypes          :     150   191

automaticFacsimileGroup3.....value reference TeleserviceCode, '01100010'B
DEFINED in MAP-TS-Code           :      50

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         :    114

baoc.....value reference SS-Code, '10010010'B
  DEFINED in MAP-SS-Code         :    105

barringOfIncomingCalls.....value reference SS-Code, '10011001'B
  DEFINED in MAP-SS-Code         :    112

barringOfOutgoingCalls.....value reference SS-Code, '10010001'B
  DEFINED in MAP-SS-Code         :    103

barringServiceActive.....identifier of Named Number, 0
  DEFINED in MAP-ER-DataTypes     :     88

basicCall.....identifier of Named Number, 0
  DEFINED in MAP-CH-DataTypes     :     88

basicService.....identifier of Ext-BasicServiceCode
  DEFINED in MAP-MS-DataTypes     :    320

basicService.....identifier of Ext-BasicServiceCode
  DEFINED in MAP-MS-DataTypes     :    395

basicService.....identifier of Ext-BasicServiceCode
  DEFINED in MAP-MS-DataTypes     :    438

basicService.....identifier of [5] Ext-BasicServiceCode
  DEFINED in MAP-CH-DataTypes     :    111

basicService.....identifier of BasicServiceCode
  DEFINED in MAP-SS-DataTypes     :     54

basicService.....identifier of BasicServiceCode
  DEFINED in MAP-SS-DataTypes     :     77

basicService.....identifier of BasicServiceCode
  DEFINED in MAP-SS-DataTypes     :    131

basicService.....identifier of BasicServiceCode
  DEFINED in MAP-SS-DataTypes     :    157

basicService.....identifier of BasicServiceCode
  DEFINED in MAP-ER-DataTypes     :    109

BasicServiceCode.....type reference CHOICE
  DEFINED in MAP-CommonDataTypes  :    266
  USED in MAP-CommonDataTypes    :     37
  USED in MAP-SS-DataTypes       :     40     54     77     131     157     213
  USED in MAP-ER-DataTypes       :     54    109

basicServiceGroup.....identifier of [9] Ext-BasicServiceCode
  DEFINED in MAP-CH-DataTypes     :     78

basicServiceGroup.....identifier of [1] Ext-BasicServiceCode
  DEFINED in MAP-CH-DataTypes     :    150

basicServiceGroupList.....identifier of Ext-BasicServiceGroupList
  DEFINED in MAP-MS-DataTypes     :    413

basicServiceGroupList.....identifier of Ext-BasicServiceGroupList
  DEFINED in MAP-MS-DataTypes     :    460

basicServiceGroupList.....identifier of BasicServiceGroupList
  DEFINED in MAP-SS-DataTypes     :    139

basicServiceGroupList.....identifier of [2] BasicServiceGroupList
  DEFINED in MAP-SS-DataTypes     :    167

BasicServiceGroupList.....type reference SEQUENCE OF
  DEFINED in MAP-SS-DataTypes     :    212
  USED in MAP-SS-DataTypes       :    139    167

basicServiceList.....identifier of [1] BasicServiceList
  DEFINED in MAP-MS-DataTypes     :    491

BasicServiceList.....type reference SEQUENCE OF
  DEFINED in MAP-MS-DataTypes     :    503
  USED in MAP-MS-DataTypes       :    491

```

```

bearerService.....identifier of [2] BearerServiceCode
  DEFINED in MAP-CommonDataTypes      :    267

BearerServiceCode.....type reference OCTET STRING
  DEFINED in MAP-BS-Code                :    11
  USED in MAP-CommonDataTypes           :    48    267
  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           :    214

BearerServiceList.....type reference SEQUENCE OF
  DEFINED in MAP-MS-DataTypes           :    236
  USED in MAP-MS-DataTypes              :    214    474

bearerServiceList.....identifier of [2] BearerServiceList
  DEFINED in MAP-MS-DataTypes           :    474

VALUE bearerServiceNotProvisioned.....value reference BearerServiceNotProvisioned, CHOICE
  DEFINED in MAP-Protocol                :    237

BearerServiceNotProvisioned.....type reference ERROR
  DEFINED in MAP-Errors                  :    187
  USED in MAP-Protocol                   :    91    237
  USED in MAP-CallHandlingOperat        :    30    68
  USED in MAP-SupplementaryServi        :    33    84   101   118   138   156
  USED in MAP-Errors                     :    30

bearerServNotProvParam.....identifier of BearerServNotProvParam
  DEFINED in MAP-Errors                  :    189

BearerServNotProvParam.....type reference SEQUENCE
  DEFINED in MAP-ER-DataTypes            :    190
  USED in MAP-Errors                     :    96    189
  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                 :    116

blackListed.....identifier of Named Number, 1
  DEFINED in MAP-MS-DataTypes            :    198

boic.....value reference SS-Code, '10010011'B
  DEFINED in MAP-SS-Code                 :    107

boicExHC.....value reference SS-Code, '10010100'B
  DEFINED in MAP-SS-Code                 :    109

broadcastInitEntitlement.....identifier of NULL
  DEFINED in MAP-MS-DataTypes            :    597

bss-APDU.....identifier of ExternalSignalInfo
  DEFINED in MAP-MobileServiceOpera     :    181

bss-APDU.....identifier of ExternalSignalInfo
  DEFINED in MAP-MobileServiceOpera     :    186

bss-APDU.....identifier of ExternalSignalInfo
  DEFINED in MAP-MobileServiceOpera     :    190

bss-APDU.....identifier of ExternalSignalInfo
  DEFINED in MAP-MobileServiceOpera     :    196

bss-APDU.....identifier of ExternalSignalInfo
  DEFINED in MAP-MS-DataTypes            :    173

bss-APDU.....identifier of ExternalSignalInfo

```

```

DEFINED in MAP-MS-DataTypes      :      178

bss-APDU.....identifier of ExternalSignalInfo
  DEFINED in MAP-MS-DataTypes      :      184

busy.....identifier of Named Number, 1
  DEFINED in MAP-CH-DataTypes      :        97

busySubscriber.....value reference BusySubscriber, CHOICE VALUE
  DEFINED in MAP-Protocol          :      260

BusySubscriber.....type reference ERROR
  DEFINED in MAP-Errors            :      228
  USED in MAP-Protocol            :        99      260
  USED in MAP-CallHandlingOperat :        34      71
  USED in MAP-Errors              :        43

busySubscriberParam.....identifier of BusySubscriberParam
  DEFINED in MAP-Errors            :      230

BusySubscriberParam.....type reference SEQUENCE
  DEFINED in MAP-ER-DataTypes      :      210
  USED in MAP-Errors              :      102      230
  USED in MAP-ER-DataTypes        :        35

callBarred.....value reference CallBarred, CHOICE VALUE
  DEFINED in MAP-Protocol          :      262

CallBarred.....type reference ERROR
  DEFINED in MAP-Errors            :      238
  USED in MAP-Protocol            :      101      262
  USED in MAP-CallHandlingOperat :        36      73
  USED in MAP-SupplementaryServi :        35      86      103      120      140      158      172      213
  USED in MAP-ShortMessageServic :        37      78
  USED in MAP-Errors              :        46

callBarredParam.....identifier of CallBarredParam
  DEFINED in MAP-Errors            :      240

CallBarredParam.....type reference CHOICE
  DEFINED in MAP-ER-DataTypes      :        80
  USED in MAP-Errors              :      104      240
  USED in MAP-ER-DataTypes        :        15

callBarringCause.....identifier of CallBarringCause
  DEFINED in MAP-ER-DataTypes      :        81

CallBarringCause.....type reference ENUMERATED
  DEFINED in MAP-ER-DataTypes      :        87
  USED in MAP-ER-DataTypes        :        81      92

callBarringCause.....identifier of CallBarringCause
  DEFINED in MAP-ER-DataTypes      :        92

CallBarringFeature.....type reference SEQUENCE
  DEFINED in MAP-SS-DataTypes      :      130
  USED in MAP-SS-DataTypes        :      128

callBarringFeatureList.....identifier of Ext-CallBarFeatureList
  DEFINED in MAP-MS-DataTypes      :      386

callBarringFeatureList.....identifier of CallBarringFeatureList
  DEFINED in MAP-SS-DataTypes      :      123

CallBarringFeatureList.....type reference SEQUENCE OF
  DEFINED in MAP-SS-DataTypes      :      126
  USED in MAP-SS-DataTypes        :      123

callBarringInfo.....identifier of [1] Ext-CallBarInfo
  DEFINED in MAP-MS-DataTypes      :      284

callBarringInfo.....identifier of [1] CallBarringInfo
  DEFINED in MAP-SS-DataTypes      :        64

CallBarringInfo.....type reference SEQUENCE
  DEFINED in MAP-SS-DataTypes      :      121
  USED in MAP-SS-DataTypes        :        64

calledPartySS-InteractionViolation.....identifier of Named Number, 7
  DEFINED in MAP-ER-DataTypes      :      105

callReferenceNumber.....identifier of [7] CallReferenceNumber
  DEFINED in MAP-CH-DataTypes      :        76

CallReferenceNumber.....type reference OCTET STRING

```

DEFINED in MAP-CH-DataTypes	:	93			
USED in MAP-CH-DataTypes	:	22	76	137	149

callReferenceNumber.....identifier of [9] CallReferenceNumber
 DEFINED in MAP-CH-DataTypes : 137

callReferenceNumber.....identifier of [0] CallReferenceNumber
 DEFINED in MAP-CH-DataTypes : 149

camelBusy.....identifier of [1] NULL
 DEFINED in MAP-MS-DataTypes : 664

camelInfo.....identifier of [11] CamelInfo
 DEFINED in MAP-CH-DataTypes : 80

CamelInfo.....type reference SEQUENCE
 DEFINED in MAP-CH-DataTypes : 162
 USED in MAP-CH-DataTypes : 80

camelRoutingInfo.....identifier of [8] CamelRoutingInfo
 DEFINED in MAP-CH-DataTypes : 170

CamelRoutingInfo.....type reference SEQUENCE
 DEFINED in MAP-CH-DataTypes : 172
 USED in MAP-CH-DataTypes : 170

camelSubscriptionInfoWithdraw.....identifier of [9] NULL
 DEFINED in MAP-MS-DataTypes : 499

cancelLocation.....value reference CancelLocation, CHOICE VALUE
 DEFINED in MAP-Protocol : 129

CancelLocation.....type reference OPERATION
 DEFINED in MAP-MobileServiceOpera : 119
 USED in MAP-Protocol : 13 129
 USED in MAP-MobileServiceOpera : 16

cancelLocationArg.....identifier of CancelLocationArg
 DEFINED in MAP-MobileServiceOpera : 121

CancelLocationArg.....type reference CHOICE
 DEFINED in MAP-MS-DataTypes : 133
 USED in MAP-MobileServiceOpera : 71 121
 USED in MAP-MS-DataTypes : 18

category.....identifier of [2] Category
 DEFINED in MAP-MS-DataTypes : 212

Category.....type reference OCTET STRING
 DEFINED in MAP-MS-DataTypes : 229
 USED in MAP-MS-DataTypes : 212

ccbs.....value reference SS-Code, '01000011'B
 DEFINED in MAP-SS-Code : 70

cellIdFixedLength.....identifier of [0] CellIdFixedLength
 DEFINED in MAP-CommonDataTypes : 241

CellIdFixedLength.....type reference OCTET STRING
 DEFINED in MAP-CommonDataTypes : 244
 USED in MAP-CommonDataTypes : 241

cellIdOrLAI.....identifier of [3] CellIdOrLAI
 DEFINED in MAP-MS-DataTypes : 636

CellIdOrLAI.....type reference CHOICE
 DEFINED in MAP-CommonDataTypes : 240
 USED in MAP-MS-DataTypes : 103 636
 USED in MAP-CommonDataTypes : 34

cfb.....value reference SS-Code, '00101001'B
 DEFINED in MAP-SS-Code : 47

cfncrc.....value reference SS-Code, '00101011'B
 DEFINED in MAP-SS-Code : 51

cfnry.....value reference SS-Code, '00101010'B
 DEFINED in MAP-SS-Code : 49

cfu.....value reference SS-Code, '00100001'B
 DEFINED in MAP-SS-Code : 43

chargeableECT-Barred.....identifier of Named Number, 10
 DEFINED in MAP-MS-DataTypes : 263

checkIMEI.....value reference CheckIMEI, CHOICE VALUE
 DEFINED in MAP-Protocol : 151

CheckIMEI.....type reference OPERATION
 DEFINED in MAP-MobileServiceOpera : 219
 USED in MAP-Protocol : 22 151
 USED in MAP-MobileServiceOpera : 37

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 : 143

CliRestrictionOption.....type reference ENUMERATED
 DEFINED in MAP-SS-DataTypes : 146
 USED in MAP-SS-DataTypes : 27 143 162

cliRestrictionOption.....identifier of CliRestrictionOption
 DEFINED in MAP-SS-DataTypes : 162

Cli-RestrictionInfo.....type reference SEQUENCE
 DEFINED in MAP-SS-DataTypes : 160
 USED in MAP-SS-DataTypes : 169

cli-RestrictionInfo.....identifier of [4] Cli-RestrictionInfo
 DEFINED in MAP-SS-DataTypes : 169

collectedInfo.....identifier of Named Number, 2
 DEFINED in MAP-MS-DataTypes : 540

colp.....value reference SS-Code, '00010011'B
 DEFINED in MAP-SS-Code : 32

colr.....value reference SS-Code, '00010100'B
 DEFINED in MAP-SS-Code : 34

Component.....type reference CHOICE
 DEFINED in TCAPMessages : 124
 USED in TCAPMessages : 47 115

ComponentPortion.....type reference [APPLICATION 12] IMPLICIT SEQUENCE
 DEFINED in TCAPMessages : 115
 USED in TCAPMessages : 59 63 67 72

components.....identifier of ComponentPortion
 DEFINED in TCAPMessages : 59

components.....identifier of ComponentPortion
 DEFINED in TCAPMessages : 63

components.....identifier of ComponentPortion
 DEFINED in TCAPMessages : 67

components.....identifier of ComponentPortion
 DEFINED in TCAPMessages : 72

Continue.....type reference SEQUENCE
 DEFINED in TCAPMessages : 69
 USED in TCAPMessages : 55

continueCall.....identifier of Named Number, 0
 DEFINED in MAP-MS-DataTypes : 548

continue-ME.....identifier of [APPLICATION 5] IMPLICIT Continue
 DEFINED in TCAPMessages : 55

controllingMSC.....identifier of Named Number, 4
 DEFINED in MAP-CommonDataTypes : 233

cug.....value reference SS-Code, '01100001'B
 DEFINED in MAP-SS-Code : 83

cugIC-CallBarred.....identifier of Named Number, 1
 DEFINED in MAP-MS-DataTypes : 424

cugOG-CallBarred.....identifier of Named Number, 2
 DEFINED in MAP-MS-DataTypes : 425

cugSubscriptionFlag.....identifier of [6] NULL
 DEFINED in MAP-CH-DataTypes : 108

```

CUG-CheckInfo.....type reference SEQUENCE
  DEFINED in MAP-CH-DataTypes      :    60
  USED in MAP-CH-DataTypes         :    70   107   153

cug-CheckInfo.....identifier of [1] CUG-CheckInfo
  DEFINED in MAP-CH-DataTypes      :    70

cug-CheckInfo.....identifier of [3] CUG-CheckInfo
  DEFINED in MAP-CH-DataTypes      :   107

cug-CheckInfo.....identifier of [4] CUG-CheckInfo
  DEFINED in MAP-CH-DataTypes      :   153

CUG-Feature.....type reference SEQUENCE
  DEFINED in MAP-MS-DataTypes      :   437
  USED in MAP-MS-DataTypes         :   430

cug-FeatureList.....identifier of CUG-FeatureList
  DEFINED in MAP-MS-DataTypes      :   402

CUG-FeatureList.....type reference SEQUENCE OF
  DEFINED in MAP-MS-DataTypes      :   429
  USED in MAP-MS-DataTypes         :   402

cug-Index.....identifier of CUG-Index
  DEFINED in MAP-MS-DataTypes      :   410

CUG-Index.....type reference INTEGER
  DEFINED in MAP-MS-DataTypes      :   417
  USED in MAP-MS-DataTypes         :    49   410   439

cug-Info.....identifier of [2] CUG-Info
  DEFINED in MAP-MS-DataTypes      :   285

CUG-Info.....type reference SEQUENCE
  DEFINED in MAP-MS-DataTypes      :   400
  USED in MAP-MS-DataTypes         :   285

cug-Interlock.....identifier of CUG-Interlock
  DEFINED in MAP-MS-DataTypes      :   411

CUG-Interlock.....type reference OCTET STRING
  DEFINED in MAP-MS-DataTypes      :   420
  USED in MAP-MS-DataTypes         :    50   411
  USED in MAP-CH-DataTypes         :    31    61

cug-Interlock.....identifier of CUG-Interlock
  DEFINED in MAP-CH-DataTypes      :    61

cug-OutgoingAccess.....identifier of NULL
  DEFINED in MAP-CH-DataTypes      :    62

cug-Reject.....value reference CUG-Reject, CHOICE VALUE
  DEFINED in MAP-Protocol          :   266

CUG-Reject.....type reference ERROR
  DEFINED in MAP-Errors            :   253
  USED in MAP-Protocol            :   104   266
  USED in MAP-CallHandlingOperat :    39    74
  USED in MAP-Errors             :    49

cug-RejectCause.....identifier of CUG-RejectCause
  DEFINED in MAP-ER-DataTypes      :    97

CUG-RejectCause.....type reference ENUMERATED
  DEFINED in MAP-ER-DataTypes      :   101
  USED in MAP-ER-DataTypes         :    97

cug-RejectParam.....identifier of CUG-RejectParam
  DEFINED in MAP-Errors            :   255

CUG-RejectParam.....type reference SEQUENCE
  DEFINED in MAP-ER-DataTypes      :    96
  USED in MAP-Errors              :   107   255
  USED in MAP-ER-DataTypes         :    16

CUG-Subscription.....type reference SEQUENCE
  DEFINED in MAP-MS-DataTypes      :   409
  USED in MAP-MS-DataTypes         :   407

cug-SubscriptionList.....identifier of CUG-SubscriptionList
  DEFINED in MAP-MS-DataTypes      :   401

CUG-SubscriptionList.....type reference SEQUENCE OF

```

[illegible]

DEFINED in MAP-OperationAndMainte	:	64	
USED in MAP-Protocol	:	37	171
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 [1] DefaultCallHandling

DEFINED in MAP-MS-DataTypes	:	533	
-----------------------------	---	-----	--

DefaultCallHandling.....type reference ENUMERATED

DEFINED in MAP-MS-DataTypes	:	547	
USED in MAP-MS-DataTypes	:	46	533
USED in MAP-CH-DataTypes	:	29	196

defaultCallHandling.....identifier of [1] DefaultCallHandling

DEFINED in MAP-CH-DataTypes	:	196	
-----------------------------	---	-----	--

defaultPriority.....identifier of EMLPP-Priority

DEFINED in MAP-MS-DataTypes	:	291	
-----------------------------	---	-----	--

deleteSubscriberData.....value reference DeleteSubscriberData, CHOICE VALUE

DEFINED in MAP-Protocol	:	157	
-------------------------	---	-----	--

DeleteSubscriberData.....type reference OPERATION

DEFINED in MAP-MobileServiceOpera	:	242	
USED in MAP-Protocol	:	24	157
USED in MAP-MobileServiceOpera	:	41	

deleteSubscriberDataArg.....identifier of DeleteSubscriberDataArg

DEFINED in MAP-MobileServiceOpera	:	244	
-----------------------------------	---	-----	--

DeleteSubscriberDataArg.....type reference SEQUENCE

DEFINED in MAP-MS-DataTypes	:	489	
USED in MAP-MobileServiceOpera	:	82	244
USED in MAP-MS-DataTypes	:	37	

deleteSubscriberDataRes.....identifier of DeleteSubscriberDataRes

DEFINED in MAP-MobileServiceOpera	:	246	
-----------------------------------	---	-----	--

DeleteSubscriberDataRes.....type reference SEQUENCE

DEFINED in MAP-MS-DataTypes	:	508	
USED in MAP-MobileServiceOpera	:	83	246
USED in MAP-MS-DataTypes	:	38	

derivable.....identifier of InvokeIdType

DEFINED in TCAPMessages	:	167	
-------------------------	---	-----	--

DestTransactionID.....type reference [APPLICATION 9] IMPLICIT

TransactionID

DEFINED in TCAPMessages	:	98	
USED in TCAPMessages	:	65	70 74

diagnosticInfo.....identifier of SignalInfo

DEFINED in MAP-ER-DataTypes	:	130	
-----------------------------	---	-----	--

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

doublyChargeableECT-Barred.....identifier of Named Number, 13
  DEFINED in MAP-MS-DataTypes :      266

dtid.....identifier of DestTransactionID
  DEFINED in TCAPMessages     :      65

dtid.....identifier of DestTransactionID
  DEFINED in TCAPMessages     :      70

dtid.....identifier of DestTransactionID
  DEFINED in TCAPMessages     :      74

duplicateInvokeID.....identifier of Named Number, 0
  DEFINED in TCAPMessages     :      183

ect.....value reference SS-Code, '00110001'B
  DEFINED in MAP-SS-Code      :      57

eir.....identifier of Named Number, 6
  DEFINED in MAP-CommonDataTypes :      235

emergencyCalls.....value reference TeleserviceCode, '00010010'B
  DEFINED in MAP-TS-Code      :      42

emlpp.....value reference SS-Code, '10100001'B
  DEFINED in MAP-SS-Code      :      140

emlpp-Info.....identifier of [4] EMLPP-Info
  DEFINED in MAP-MS-DataTypes :      287

EMLPP-Info.....type reference SEQUENCE
  DEFINED in MAP-MS-DataTypes :      289
  USED in MAP-MS-DataTypes   :      287

EMLPP-Priority.....type reference INTEGER
  DEFINED in MAP-MS-DataTypes :      295
  USED in MAP-MS-DataTypes   :      290      291      301      302      303      304      305      306

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 :      198

enterNewPW-Again.....identifier of Named Number, 2
  DEFINED in MAP-SS-DataTypes :      199

enterPW.....identifier of Named Number, 0
  DEFINED in MAP-SS-DataTypes :      197

equipmentNotSM-Equipped.....identifier of Named Number, 2
  DEFINED in MAP-ER-DataTypes :      122

equipmentProtocolError.....identifier of Named Number, 1
  DEFINED in MAP-ER-DataTypes :      121

equipmentStatus.....identifier of EquipmentStatus
  DEFINED in MAP-MobileServiceOpera :      223

EquipmentStatus.....type reference ENUMERATED
  DEFINED in MAP-MS-DataTypes :      196
  USED in MAP-MobileServiceOpera :      79      223
  USED in MAP-MS-DataTypes   :      32

eraseSS.....value reference EraseSS, CHOICE VALUE
  DEFINED in MAP-Protocol     :      185

EraseSS.....type reference OPERATION
  DEFINED in MAP-SupplementaryServi :      91
  USED in MAP-Protocol           :      53      185
  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	:	186
extendedRoutingInfo.....	identifier of ExtendedRoutingInfo	
DEFINED in MAP-CH-DataTypes	:	106
ExtendedRoutingInfo.....	type reference CHOICE	
DEFINED in MAP-CH-DataTypes	:	168
USED in MAP-CH-DataTypes	:	106
extensibleCallBarredParam.....	identifier of ExtensibleCallBarredParam	
DEFINED in MAP-ER-DataTypes	:	83
ExtensibleCallBarredParam.....	type reference SEQUENCE	
DEFINED in MAP-ER-DataTypes	:	91
USED in MAP-ER-DataTypes	:	83
extensibleSystemFailureParam.....	identifier of ExtensibleSystemFailureParam	
DEFINED in MAP-ER-DataTypes	:	145
ExtensibleSystemFailureParam.....	type reference SEQUENCE	
DEFINED in MAP-ER-DataTypes	:	149
USED in MAP-ER-DataTypes	:	145
extensionContainer.....	identifier of ExtensionContainer	
DEFINED in MAP-MS-DataTypes	:	125
extensionContainer.....	identifier of ExtensionContainer	
DEFINED in MAP-MS-DataTypes	:	131
extensionContainer.....	identifier of [14] ExtensionContainer	
DEFINED in MAP-MS-DataTypes	:	207
extensionContainer.....	identifier of ExtensionContainer	
DEFINED in MAP-MS-DataTypes	:	249
extensionContainer.....	identifier of ExtensionContainer	
DEFINED in MAP-MS-DataTypes	:	292
extensionContainer.....	identifier of [0] ExtensionContainer	
DEFINED in MAP-MS-DataTypes	:	312
extensionContainer.....	identifier of [9] ExtensionContainer	
DEFINED in MAP-MS-DataTypes	:	326
extensionContainer.....	identifier of ExtensionContainer	
DEFINED in MAP-MS-DataTypes	:	387
extensionContainer.....	identifier of ExtensionContainer	
DEFINED in MAP-MS-DataTypes	:	397
extensionContainer.....	identifier of [0] ExtensionContainer	
DEFINED in MAP-MS-DataTypes	:	403
extensionContainer.....	identifier of [0] ExtensionContainer	
DEFINED in MAP-MS-DataTypes	:	414
extensionContainer.....	identifier of ExtensionContainer	
DEFINED in MAP-MS-DataTypes	:	441
extensionContainer.....	identifier of [5] ExtensionContainer	
DEFINED in MAP-MS-DataTypes	:	461
extensionContainer.....	identifier of [7] ExtensionContainer	
DEFINED in MAP-MS-DataTypes	:	480
extensionContainer.....	identifier of [6] ExtensionContainer	
DEFINED in MAP-MS-DataTypes	:	500
extensionContainer.....	identifier of ExtensionContainer	
DEFINED in MAP-MS-DataTypes	:	511
extensionContainer.....	identifier of [1] ExtensionContainer	
DEFINED in MAP-MS-DataTypes	:	516
extensionContainer.....	identifier of ExtensionContainer	
DEFINED in MAP-MS-DataTypes	:	521
extensionContainer.....	identifier of [2] ExtensionContainer	
DEFINED in MAP-MS-DataTypes	:	534
extensionContainer.....	identifier of ExtensionContainer	
DEFINED in MAP-MS-DataTypes	:	569
extensionContainer.....	identifier of ExtensionContainer	
DEFINED in MAP-MS-DataTypes	:	575

extensionContainer.....	identifier of ExtensionContainer
DEFINED in MAP-MS-DataTypes	: 592
extensionContainer.....	identifier of ExtensionContainer
DEFINED in MAP-MS-DataTypes	: 598
extensionContainer.....	identifier of [3] ExtensionContainer
DEFINED in MAP-MS-DataTypes	: 611
extensionContainer.....	identifier of ExtensionContainer
DEFINED in MAP-MS-DataTypes	: 616
extensionContainer.....	identifier of [2] ExtensionContainer
DEFINED in MAP-MS-DataTypes	: 622
extensionContainer.....	identifier of [2] ExtensionContainer
DEFINED in MAP-MS-DataTypes	: 628
extensionContainer.....	identifier of [4] ExtensionContainer
DEFINED in MAP-MS-DataTypes	: 637
extensionContainer.....	identifier of [2] ExtensionContainer
DEFINED in MAP-MS-DataTypes	: 680
extensionContainer.....	identifier of ExtensionContainer
DEFINED in MAP-MS-DataTypes	: 685
extensionContainer.....	identifier of ExtensionContainer
DEFINED in MAP-CommonDataTypes	: 167
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	: 63
extensionContainer.....	identifier of [13] ExtensionContainer
DEFINED in MAP-CH-DataTypes	: 82
extensionContainer.....	identifier of [0] ExtensionContainer
DEFINED in MAP-CH-DataTypes	: 114
extensionContainer.....	identifier of [7] ExtensionContainer
DEFINED in MAP-CH-DataTypes	: 125
extensionContainer.....	identifier of [11] ExtensionContainer
DEFINED in MAP-CH-DataTypes	: 139
extensionContainer.....	identifier of ExtensionContainer
DEFINED in MAP-CH-DataTypes	: 144
extensionContainer.....	identifier of [7] ExtensionContainer
DEFINED in MAP-CH-DataTypes	: 155
extensionContainer.....	identifier of ExtensionContainer
DEFINED in MAP-CH-DataTypes	: 159
extensionContainer.....	identifier of ExtensionContainer
DEFINED in MAP-CH-DataTypes	: 165
extensionContainer.....	identifier of [1] ExtensionContainer
DEFINED in MAP-CH-DataTypes	: 175
extensionContainer.....	identifier of [2] ExtensionContainer
DEFINED in MAP-CH-DataTypes	: 181
extensionContainer.....	identifier of ExtensionContainer
DEFINED in MAP-CH-DataTypes	: 186
extensionContainer.....	identifier of [2] ExtensionContainer
DEFINED in MAP-CH-DataTypes	: 197
extensionContainer.....	identifier of [6] ExtensionContainer
DEFINED in MAP-SM-DataTypes	: 55

extensionContainer.....	identifier of [4] ExtensionContainer
DEFINED in MAP-SM-DataTypes	: 61
extensionContainer.....	identifier of ExtensionContainer
DEFINED in MAP-SM-DataTypes	: 67
extensionContainer.....	identifier of ExtensionContainer
DEFINED in MAP-SM-DataTypes	: 74
extensionContainer.....	identifier of ExtensionContainer
DEFINED in MAP-SM-DataTypes	: 79
extensionContainer.....	identifier of ExtensionContainer
DEFINED in MAP-SM-DataTypes	: 87
extensionContainer.....	identifier of ExtensionContainer
DEFINED in MAP-SM-DataTypes	: 92
extensionContainer.....	identifier of [1] ExtensionContainer
DEFINED in MAP-SM-DataTypes	: 112
extensionContainer.....	identifier of ExtensionContainer
DEFINED in MAP-SM-DataTypes	: 122
extensionContainer.....	identifier of ExtensionContainer
DEFINED in MAP-SM-DataTypes	: 134
extensionContainer.....	identifier of ExtensionContainer
DEFINED in MAP-ER-DataTypes	: 73
extensionContainer.....	identifier of ExtensionContainer
DEFINED in MAP-ER-DataTypes	: 93
extensionContainer.....	identifier of ExtensionContainer
DEFINED in MAP-ER-DataTypes	: 98
extensionContainer.....	identifier of ExtensionContainer
DEFINED in MAP-ER-DataTypes	: 131
extensionContainer.....	identifier of ExtensionContainer
DEFINED in MAP-ER-DataTypes	: 136
extensionContainer.....	identifier of ExtensionContainer
DEFINED in MAP-ER-DataTypes	: 151
extensionContainer.....	identifier of ExtensionContainer
DEFINED in MAP-ER-DataTypes	: 155
extensionContainer.....	identifier of ExtensionContainer
DEFINED in MAP-ER-DataTypes	: 159
extensionContainer.....	identifier of ExtensionContainer
DEFINED in MAP-ER-DataTypes	: 163
extensionContainer.....	identifier of ExtensionContainer
DEFINED in MAP-ER-DataTypes	: 167
extensionContainer.....	identifier of ExtensionContainer
DEFINED in MAP-ER-DataTypes	: 171
extensionContainer.....	identifier of ExtensionContainer
DEFINED in MAP-ER-DataTypes	: 175
extensionContainer.....	identifier of ExtensionContainer
DEFINED in MAP-ER-DataTypes	: 179
extensionContainer.....	identifier of ExtensionContainer
DEFINED in MAP-ER-DataTypes	: 183
extensionContainer.....	identifier of ExtensionContainer
DEFINED in MAP-ER-DataTypes	: 187
extensionContainer.....	identifier of ExtensionContainer
DEFINED in MAP-ER-DataTypes	: 191
extensionContainer.....	identifier of ExtensionContainer
DEFINED in MAP-ER-DataTypes	: 195
extensionContainer.....	identifier of ExtensionContainer
DEFINED in MAP-ER-DataTypes	: 199
extensionContainer.....	identifier of ExtensionContainer
DEFINED in MAP-ER-DataTypes	: 203

	extensionContainer.....	identifier of ExtensionContainer							
	DEFINED in MAP-ER-DataTypes	:	207						
	extensionContainer.....	identifier of ExtensionContainer							
	DEFINED in MAP-ER-DataTypes	:	211						
	extensionContainer.....	identifier of ExtensionContainer							
	DEFINED in MAP-ER-DataTypes	:	215						
	extensionContainer.....	identifier of ExtensionContainer							
	DEFINED in MAP-ER-DataTypes	:	219						
	extensionContainer.....	identifier of ExtensionContainer							
	DEFINED in MAP-ER-DataTypes	:	223						
	extensionContainer.....	identifier of ExtensionContainer							
	DEFINED in MAP-ER-DataTypes	:	227						
	extensionContainer.....	identifier of ExtensionContainer							
	DEFINED in MAP-ER-DataTypes	:	231						
	extensionContainer.....	identifier of ExtensionContainer							
	DEFINED in MAP-ER-DataTypes	:	235						
	ExtensionContainer.....	type reference SEQUENCE							
	DEFINED in MAP-ExtensionDataTypes	:	32						
387	USED in MAP-MS-DataTypes	:	110	125	131	207	249	292	312 326
516			397	403	414	441	461	480	500 511
622			521	534	569	575	592	598	611 616
			628	637	680	685			
	USED in MAP-CommonDataTypes	:	54	167					
	USED in MAP-OM-DataTypes	:	27	41	51	57	61		
159	USED in MAP-CH-DataTypes	:	53	63	82	114	125	139	144 155
			165	175	181	186	197		
112	USED in MAP-SM-DataTypes	:	44	55	61	67	74	79	87 92
			122	134					
159	USED in MAP-ER-DataTypes	:	65	73	93	98	131	136	151 155
195			163	167	171	175	179	183	187 191
231			199	203	207	211	215	219	223 227
			235						
	USED in MAP-ExtensionDataTypes	:	16						
	ExtensionSet.....	information object set reference MAP-EXTENSION, Information							
Object Set	DEFINED in MAP-ExtensionDataTypes	:	48						
	USED in MAP-ExtensionDataTypes	:	42	44					
	ExternalSignalInfo.....	type reference SEQUENCE							
	DEFINED in MAP-CommonDataTypes	:	162						
	USED in MAP-MobileServiceOpera	:	96	181	186	190	196		
	USED in MAP-MS-DataTypes	:	98	173	178	184			
	USED in MAP-CommonDataTypes	:	19						
	USED in MAP-CH-DataTypes	:	45	79	133	134			
	extId.....	identifier of InformationObjectClassFieldType							
	DEFINED in MAP-ExtensionDataTypes	:	41						
	extType.....	identifier of InformationObjectClassFieldType							
	DEFINED in MAP-ExtensionDataTypes	:	43						
	Ext-BasicServiceCode.....	type reference CHOICE							
	DEFINED in MAP-CommonDataTypes	:	270						
	USED in MAP-MS-DataTypes	:	104	320	395	433	438	504	
	USED in MAP-CommonDataTypes	:	38						
	USED in MAP-CH-DataTypes	:	48	78	111	150			
	Ext-BasicServiceGroupList.....	type reference SEQUENCE OF							
	DEFINED in MAP-MS-DataTypes	:	432						
	USED in MAP-MS-DataTypes	:	413	460					
	ext-BearerService.....	identifier of [2] Ext-BearerServiceCode							
	DEFINED in MAP-CommonDataTypes	:	271						
	Ext-BearerServiceCode.....	type reference OCTET STRING							
	DEFINED in MAP-BS-Code	:	25						
	USED in MAP-MS-DataTypes	:	85	237					
	USED in MAP-CommonDataTypes	:	49	271					

Ext-CallBarFeatureList.....	type reference SEQUENCE OF			
DEFINED in MAP-MS-DataTypes	:	390		
USED in MAP-MS-DataTypes	:	386		
Ext-CallBarInfo.....	type reference SEQUENCE			
DEFINED in MAP-MS-DataTypes	:	384		
USED in MAP-MS-DataTypes	:	284		
Ext-CallBarringFeature.....	type reference SEQUENCE			
DEFINED in MAP-MS-DataTypes	:	394		
USED in MAP-MS-DataTypes	:	392		
Ext-ForwFeature.....	type reference SEQUENCE			
DEFINED in MAP-MS-DataTypes	:	319		
USED in MAP-MS-DataTypes	:	317		
Ext-ForwFeatureList.....	type reference SEQUENCE OF			
DEFINED in MAP-MS-DataTypes	:	315		
USED in MAP-MS-DataTypes	:	311		
Ext-ForwInfo.....	type reference SEQUENCE			
DEFINED in MAP-MS-DataTypes	:	309		
USED in MAP-MS-DataTypes	:	283		
Ext-ForwOptions.....	type reference OCTET STRING			
DEFINED in MAP-MS-DataTypes	:	350		
USED in MAP-MS-DataTypes	:	324		
Ext-NoRepCondTime.....	type reference INTEGER			
DEFINED in MAP-MS-DataTypes	:	377		
USED in MAP-MS-DataTypes	:	325		
Ext-SS-Data.....	type reference SEQUENCE			
DEFINED in MAP-MS-DataTypes	:	456		
USED in MAP-MS-DataTypes	:	286		
Ext-SS-Info.....	type reference CHOICE			
DEFINED in MAP-MS-DataTypes	:	282		
USED in MAP-MS-DataTypes	:	280		
Ext-SS-InfoList.....	type reference SEQUENCE OF			
DEFINED in MAP-MS-DataTypes	:	279		
USED in MAP-MS-DataTypes	:	220		
Ext-SS-Status.....	type reference OCTET STRING			
DEFINED in MAP-MS-DataTypes	:	329		
USED in MAP-MS-DataTypes	:	321	396	458
ext-Teleservice.....	identifier of [3] Ext-TeleserviceCode			
DEFINED in MAP-CommonDataTypes	:	272		
Ext-TeleserviceCode.....	type reference OCTET STRING			
DEFINED in MAP-TS-Code	:	20		
USED in MAP-MS-DataTypes	:	90	242	
USED in MAP-CommonDataTypes	:	43	272	
facilityNotSupParam.....	identifier of FacilityNotSupParam			
DEFINED in MAP-Errors	:	140		
FacilityNotSupParam.....	type reference SEQUENCE			
DEFINED in MAP-ER-DataTypes	:	162		
USED in MAP-Errors	:	89	140	
USED in MAP-ER-DataTypes	:	23		
facilityNotSupported.....	value reference FacilityNotSupported, CHOICE VALUE			
DEFINED in MAP-Protocol	:	220		
FacilityNotSupported.....	type reference ERROR			
DEFINED in MAP-Errors	:	138		
USED in MAP-Protocol	:	82	220	
USED in MAP-OperationAndMainte	:	26	60	74
USED in MAP-CallHandlingOperat	:	26	64	86
USED in MAP-ShortMessageServic	:	30	75	90 103 143
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		
forwardAccessSignalling.....	value reference ForwardAccessSignalling, CHOICE			
VALUE				
DEFINED in MAP-Protocol	:	139		

```

ForwardAccessSignalling.....type reference OPERATION
    DEFINED in MAP-MobileServiceOpera : 188
    USED in MAP-Protocol : 19 139
    USED in MAP-MobileServiceOpera : 30

forwardCheckSS-Indication.....value reference ForwardCheckSS-Indication, CHOICE
VALUE
    DEFINED in MAP-Protocol : 163

ForwardCheckSS-Indication.....type reference OPERATION
    DEFINED in MAP-MobileServiceOpera : 259
    USED in MAP-Protocol : 26 163
    USED in MAP-MobileServiceOpera : 45

forwardedToNumber.....identifier of [5] ISDN-AddressString
    DEFINED in MAP-MS-DataTypes : 322

forwardedToNumber.....identifier of [5] ISDN-AddressString
    DEFINED in MAP-CH-DataTypes : 122

forwardedToNumber.....identifier of [4] AddressString
    DEFINED in MAP-SS-DataTypes : 55

forwardedToNumber.....identifier of [5] ISDN-AddressString
    DEFINED in MAP-SS-DataTypes : 79

forwardedToSubaddress.....identifier of [8] ISDN-SubaddressString
    DEFINED in MAP-MS-DataTypes : 323

forwardedToSubaddress.....identifier of [4] ISDN-SubaddressString
    DEFINED in MAP-CH-DataTypes : 123

forwardedToSubaddress.....identifier of [6] ISDN-SubaddressString
    DEFINED in MAP-SS-DataTypes : 56

forwardedToSubaddress.....identifier of [8] ISDN-SubaddressString
    DEFINED in MAP-SS-DataTypes : 80

forwarding.....identifier of Named Number, 1
    DEFINED in MAP-CH-DataTypes : 89

forwardingData.....identifier of ForwardingData
    DEFINED in MAP-CH-DataTypes : 119

ForwardingData.....type reference SEQUENCE
    DEFINED in MAP-CH-DataTypes : 121
    USED in MAP-CH-DataTypes : 119 151 173

forwardingData.....identifier of [2] ForwardingData
    DEFINED in MAP-CH-DataTypes : 151

forwardingData.....identifier of ForwardingData
    DEFINED in MAP-CH-DataTypes : 173

forwardingFailed.....value reference ForwardingFailed, CHOICE VALUE
    DEFINED in MAP-Protocol : 263

ForwardingFailed.....type reference ERROR
    DEFINED in MAP-Errors : 248
    USED in MAP-Protocol : 103 263
    USED in MAP-CallHandlingOperat : 38 97
    USED in MAP-Errors : 48

forwardingFailedParam.....identifier of ForwardingFailedParam
    DEFINED in MAP-Errors : 250

ForwardingFailedParam.....type reference SEQUENCE
    DEFINED in MAP-ER-DataTypes : 222
    USED in MAP-Errors : 106 250
    USED in MAP-ER-DataTypes : 38

ForwardingFeature.....type reference SEQUENCE
    DEFINED in MAP-SS-DataTypes : 76
    USED in MAP-SS-DataTypes : 74

forwardingFeatureList.....identifier of Ext-ForwFeatureList
    DEFINED in MAP-MS-DataTypes : 311

forwardingFeatureList.....identifier of ForwardingFeatureList
    DEFINED in MAP-SS-DataTypes : 69

ForwardingFeatureList.....type reference SEQUENCE OF
    DEFINED in MAP-SS-DataTypes : 72
    USED in MAP-SS-DataTypes : 69 168

```

```

forwardingFeatureList.....identifier of [3] ForwardingFeatureList
  DEFINED in MAP-SS-DataTypes      :    168

forwardingInfo.....identifier of [0] Ext-ForwInfo
  DEFINED in MAP-MS-DataTypes      :    283

forwardingInfo.....identifier of [0] ForwardingInfo
  DEFINED in MAP-SS-DataTypes      :     63

ForwardingInfo.....type reference SEQUENCE
  DEFINED in MAP-SS-DataTypes      :     67
  USED in MAP-SS-DataTypes        :     63

forwardingInterrogationRequired.....identifier of [4] NULL
  DEFINED in MAP-CH-DataTypes      :    112

forwardingOptions.....identifier of [6] Ext-ForwOptions
  DEFINED in MAP-MS-DataTypes      :    324

forwardingOptions.....identifier of [6] ForwardingOptions
  DEFINED in MAP-CH-DataTypes      :    124

forwardingOptions.....identifier of [6] ForwardingOptions
  DEFINED in MAP-SS-DataTypes      :     81

ForwardingOptions.....type reference OCTET STRING
  DEFINED in MAP-SS-DataTypes      :    100
  USED in MAP-CH-DataTypes        :     37   124
  USED in MAP-SS-DataTypes        :     29   81

forwardingReason.....identifier of [8] ForwardingReason
  DEFINED in MAP-CH-DataTypes      :     77

ForwardingReason.....type reference ENUMERATED
  DEFINED in MAP-CH-DataTypes      :     95
  USED in MAP-CH-DataTypes        :     77

forwardingViolation.....value reference ForwardingViolation, CHOICE VALUE
  DEFINED in MAP-Protocol          :    265

ForwardingViolation.....type reference ERROR
  DEFINED in MAP-Errors            :    243
  USED in MAP-Protocol            :    102   265
  USED in MAP-CallHandlingOperat :     37   75
  USED in MAP-Errors              :     47

forwardingViolationParam.....identifier of ForwardingViolationParam
  DEFINED in MAP-Errors            :    245

ForwardingViolationParam.....type reference SEQUENCE
  DEFINED in MAP-ER-DataTypes      :    218
  USED in MAP-Errors              :    105   245
  USED in MAP-ER-DataTypes        :     37

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

geographicalInformation.....identifier of [0] GeographicalInformation
  DEFINED in MAP-MS-DataTypes      :    633

GeographicalInformation.....type reference OCTET STRING
  DEFINED in MAP-MS-DataTypes      :    649
  USED in MAP-MS-DataTypes        :    633

getPassword.....value reference GetPassword, CHOICE VALUE
  DEFINED in MAP-Protocol          :    194

GetPassword.....type reference OPERATION

```

```

    DEFINED in MAP-SupplementaryServi :    221
    USED in MAP-Protocol :    61    194
    USED in MAP-SupplementaryServi :    22    219

GlobalCellId.....type reference OCTET STRING
    DEFINED in MAP-CommonDataTypes :    218
    USED in MAP-MS-DataTypes :    102    171    182
    USED in MAP-CommonDataTypes :    30

gsmcCamelSubscriptionInfo.....identifier of [0] GsmcCamelSubscriptionInfo
    DEFINED in MAP-CH-DataTypes :    174

GsmcCamelSubscriptionInfo.....type reference SEQUENCE
    DEFINED in MAP-CH-DataTypes :    178
    USED in MAP-CH-DataTypes :    174

gsmc-Address.....identifier of [6] ISDN-AddressString
    DEFINED in MAP-CH-DataTypes :    75

gsmc-Address.....identifier of [8] ISDN-AddressString
    DEFINED in MAP-CH-DataTypes :    136

greyListed.....identifier of Named Number, 2
    DEFINED in MAP-MS-DataTypes :    199

groupId.....identifier of GroupId
    DEFINED in MAP-MS-DataTypes :    591

groupid.....identifier of GroupId
    DEFINED in MAP-MS-DataTypes :    596

GroupId.....type reference OCTET STRING
    DEFINED in MAP-MS-DataTypes :    601
    USED in MAP-MS-DataTypes :    591    596

gsmSCF-Address.....identifier of [0] ISDN-AddressString
    DEFINED in MAP-MS-DataTypes :    532

gsmSCF-Address.....identifier of [3] ISDN-AddressString
    DEFINED in MAP-MS-DataTypes :    679

gsmSCF-Address.....identifier of [0] ISDN-AddressString
    DEFINED in MAP-CH-DataTypes :    195

gsm-0408.....identifier of Named Number, 1
    DEFINED in MAP-CommonDataTypes :    182

gsm-0806.....identifier of Named Number, 2
    DEFINED in MAP-CommonDataTypes :    183

gsm-BearerCapability.....identifier of [5] ExternalSignalInfo
    DEFINED in MAP-CH-DataTypes :    133

gsm-BSSMAP.....identifier of Named Number, 3
    DEFINED in MAP-CommonDataTypes :    184

guidanceInfo.....identifier of GuidanceInfo
    DEFINED in MAP-SupplementaryServi :    223

GuidanceInfo.....type reference ENUMERATED
    DEFINED in MAP-SS-DataTypes :    196
    USED in MAP-SupplementaryServi :    60    223
    USED in MAP-SS-DataTypes :    23

handoverNumber.....identifier of ISDN-AddressString
    DEFINED in MAP-MS-DataTypes :    177

hlr.....identifier of Named Number, 1
    DEFINED in MAP-CommonDataTypes :    230

HLR-Id.....type reference IMSI
    DEFINED in MAP-CommonDataTypes :    207
    USED in MAP-CommonDataTypes :    212

hlr-List.....identifier of HLR-List
    DEFINED in MAP-MS-DataTypes :    563

HLR-List.....type reference SEQUENCE OF
    DEFINED in MAP-CommonDataTypes :    211
    USED in MAP-MS-DataTypes :    100    563
    USED in MAP-CommonDataTypes :    28

hlr-Number.....identifier of ISDN-AddressString
    DEFINED in MAP-MS-DataTypes :    129

```

```

hlr-Number.....identifier of ISDN-AddressString
  DEFINED in MAP-MS-DataTypes      :    562

hlr-Number.....identifier of ISDN-AddressString
  DEFINED in MAP-MS-DataTypes      :    573

hold.....value reference SS-Code, '01000010'B
  DEFINED in MAP-SS-Code           :     68

ho-NumberNotRequired.....identifier of NULL
  DEFINED in MAP-MS-DataTypes      :    172

illegalEquipment.....value reference IllegalEquipment, CHOICE VALUE
  DEFINED in MAP-Protocol          :    236

IllegalEquipment.....type reference ERROR
  DEFINED in MAP-Errors            :    181
  USED in MAP-Protocol            :    90    236
  USED in MAP-SupplementaryServi   :    48    186    200
  USED in MAP-ShortMessageServic   :    34    106
  USED in MAP-Errors              :     29

illegalEquipmentParam.....identifier of IllegalEquipmentParam
  DEFINED in MAP-Errors            :    183

IllegalEquipmentParam.....type reference SEQUENCE
  DEFINED in MAP-ER-DataTypes      :    186
  USED in MAP-Errors              :     95    183
  USED in MAP-ER-DataTypes        :     29

illegalSS-Operation.....value reference IllegalSS-Operation, CHOICE VALUE
  DEFINED in MAP-Protocol          :    275

IllegalSS-Operation.....type reference ERROR
  DEFINED in MAP-Errors            :    273
  USED in MAP-Protocol            :    106    275
  USED in MAP-SupplementaryServi   :     36     87    104    121    141    159
  USED in MAP-Errors              :     55

illegalSubscriber.....value reference IllegalSubscriber, CHOICE VALUE
  DEFINED in MAP-Protocol          :    235

IllegalSubscriber.....type reference ERROR
  DEFINED in MAP-Errors            :    175
  USED in MAP-Protocol            :     89    235
  USED in MAP-SupplementaryServi   :     47    185    199
  USED in MAP-ShortMessageServic   :     33    105
  USED in MAP-Errors              :     28

illegalSubscriberParam.....identifier of IllegalSubscriberParam
  DEFINED in MAP-Errors            :    177

IllegalSubscriberParam.....type reference SEQUENCE
  DEFINED in MAP-ER-DataTypes      :    182
  USED in MAP-Errors              :     94    177
  USED in MAP-ER-DataTypes        :     28

imei.....identifier of IMEI
  DEFINED in MAP-MobileServiceOpera :    221

IMEI.....type reference TB-CD-STRING
  DEFINED in MAP-CommonDataTypes    :    200
  USED in MAP-MobileServiceOpera    :     98    221
  USED in MAP-CommonDataTypes       :     27

imsi.....identifier of IMSI
  DEFINED in MAP-OperationAndMainte :     81

imsi.....identifier of IMSI
  DEFINED in MAP-MS-DataTypes       :    121

imsi.....identifier of IMSI
  DEFINED in MAP-MS-DataTypes       :    134

imsi.....identifier of IMSI
  DEFINED in MAP-MS-DataTypes       :    138

imsi.....identifier of IMSI
  DEFINED in MAP-MS-DataTypes       :    143

imsi.....identifier of IMSI
  DEFINED in MAP-MS-DataTypes       :    149

imsi.....identifier of [0] IMSI
  DEFINED in MAP-MS-DataTypes       :    205

```

```

imsi.....identifier of [0] IMSI
  DEFINED in MAP-MS-DataTypes      :      490

imsi.....identifier of IMSI
  DEFINED in MAP-MS-DataTypes      :      567

imsi.....identifier of [0] IMSI
  DEFINED in MAP-MS-DataTypes      :      608

imsi.....identifier of [0] IMSI
  DEFINED in MAP-MS-DataTypes      :      689

IMSI.....type reference TBCD-STRING
  DEFINED in MAP-CommonDataTypes   :      191
  USED in MAP-OperationAndMainte   :      43      81
  USED in MAP-MS-DataTypes         :      99     121     134     138     143     149     189     205

490
      567     608     689
  USED in MAP-CommonDataTypes     :      24     197     207
  USED in MAP-OM-DataTypes         :      22      37      55
  USED in MAP-CH-DataTypes         :      46     102     129     152
  USED in MAP-SM-DataTypes         :      33      59      96     145

imsi.....identifier of [0] IMSI
  DEFINED in MAP-CommonDataTypes   :      197

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      :      102

imsi.....identifier of [0] IMSI
  DEFINED in MAP-CH-DataTypes      :      129

imsi.....identifier of [3] IMSI
  DEFINED in MAP-CH-DataTypes      :      152

imsi.....identifier of IMSI
  DEFINED in MAP-SM-DataTypes      :      59

imsi.....identifier of [0] IMSI
  DEFINED in MAP-SM-DataTypes      :      96

imsi.....identifier of [0] IMSI
  DEFINED in MAP-SM-DataTypes      :      145

imsiDetached.....identifier of Named Number, 1
  DEFINED in MAP-MS-DataTypes      :      670

imsi-WithLMSI.....identifier of IMSI-WithLMSI
  DEFINED in MAP-MS-DataTypes      :      135

IMSI-WithLMSI.....type reference SEQUENCE
  DEFINED in MAP-MS-DataTypes      :      142
  USED in MAP-MS-DataTypes         :      135

incomingCallsBarredWithinCUG.....identifier of Named Number, 0
  DEFINED in MAP-ER-DataTypes      :      102

incorrectTransactionPortion.....identifier of Named Number, 3
  DEFINED in TCAPMessages          :      106

informServiceCentre.....value reference InformServiceCentre, CHOICE VALUE
  DEFINED in MAP-Protocol           :      203

InformServiceCentre.....type reference OPERATION
  DEFINED in MAP-ShortMessageServic :      132
  USED in MAP-Protocol              :      72     203
  USED in MAP-ShortMessageServic   :      18

informServiceCentreArg.....identifier of InformServiceCentreArg
  DEFINED in MAP-ShortMessageServic :      134

InformServiceCentreArg.....type reference SEQUENCE
  DEFINED in MAP-SM-DataTypes       :      131
  USED in MAP-ShortMessageServic    :      55     134
  USED in MAP-SM-DataTypes          :      23

initiatingRelease.....identifier of Named Number, 4
  DEFINED in TCAPMessages           :      187

```

```

insertSubscriberData.....value reference InsertSubscriberData, CHOICE VALUE
    DEFINED in MAP-Protocol : 156

InsertSubscriberData.....type reference OPERATION
    DEFINED in MAP-MobileServiceOpera : 231
    USED in MAP-Protocol : 23 156
    USED in MAP-MobileServiceOpera : 40

insertSubscriberDataArg.....identifier of InsertSubscriberDataArg
    DEFINED in MAP-MobileServiceOpera : 233

InsertSubscriberDataArg.....type reference SEQUENCE
    DEFINED in MAP-MS-DataTypes : 204
    USED in MAP-MobileServiceOpera : 80 233
    USED in MAP-MS-DataTypes : 35

insertSubscriberDataRes.....identifier of InsertSubscriberDataRes
    DEFINED in MAP-MobileServiceOpera : 235

InsertSubscriberDataRes.....type reference SEQUENCE
    DEFINED in MAP-MS-DataTypes : 472
    USED in MAP-MobileServiceOpera : 81 235
    USED in MAP-MS-DataTypes : 36

interCUG-Restrictions.....identifier of InterCUG-Restrictions
    DEFINED in MAP-MS-DataTypes : 440

InterCUG-Restrictions.....type reference OCTET STRING
    DEFINED in MAP-MS-DataTypes : 444
    USED in MAP-MS-DataTypes : 51 440

internationalECT-Barred.....identifier of Named Number, 11
    DEFINED in MAP-MS-DataTypes : 264

internationalOGCallsBarred.....identifier of Named Number, 1
    DEFINED in MAP-MS-DataTypes : 254

internationalOGCallsNotToHPLMN-CountryBaidentifier of Named Number, 2
    DEFINED in MAP-MS-DataTypes : 255

interrogateSS.....value reference InterrogateSS, CHOICE VALUE
    DEFINED in MAP-Protocol : 188

InterrogateSS.....type reference OPERATION
    DEFINED in MAP-SupplementaryServi : 147
    USED in MAP-Protocol : 56 188
    USED in MAP-SupplementaryServi : 17

interrogateSS-Res.....identifier of InterrogateSS-Res
    DEFINED in MAP-SupplementaryServi : 151

InterrogateSS-Res.....type reference CHOICE
    DEFINED in MAP-SS-DataTypes : 165
    USED in MAP-SupplementaryServi : 56 151
    USED in MAP-SS-DataTypes : 19

interrogationType.....identifier of [3] InterrogationType
    DEFINED in MAP-CH-DataTypes : 72

InterrogationType.....type reference ENUMERATED
    DEFINED in MAP-CH-DataTypes : 87
    USED in MAP-CH-DataTypes : 72

interzonalECT-Barred.....identifier of Named Number, 12
    DEFINED in MAP-MS-DataTypes : 265

interzonalOGCallsAndInternationalOGCallsidentifier of Named Number, 8
    DEFINED in MAP-MS-DataTypes : 258

interzonalOGCallsBarred.....identifier of Named Number, 6
    DEFINED in MAP-MS-DataTypes : 256

interzonalOGCallsNotToHPLMN-CountryBarreidentifier of Named Number, 7
    DEFINED in MAP-MS-DataTypes : 257

intraCUG-Options.....identifier of IntraCUG-Options
    DEFINED in MAP-MS-DataTypes : 412

IntraCUG-Options.....type reference ENUMERATED
    DEFINED in MAP-MS-DataTypes : 422
    USED in MAP-MS-DataTypes : 52 412

invalidFormat.....identifier of Named Number, 1
    DEFINED in MAP-ER-DataTypes : 115

```


	invalidSME-Address.....	identifier of Named Number, 5	
	DEFINED in MAP-ER-DataTypes	:	125
	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	:	116
	USED in MAP-OperationAndMainte	:	42 79
	USED in MAP-MS-DataTypes	:	96 123 123 129 139 177 183 211
322			532 562 573 634 679 690
	USED in MAP-CommonDataTypes	:	17
136	USED in MAP-CH-DataTypes	:	43 69 75 113 118 122 130 131
			143 195
	USED in MAP-SS-DataTypes	:	38 79
	USED in MAP-SM-DataTypes	:	31 52 65 102 107 121 127 132
	ISDN-SubaddressString.....	type reference OCTET STRING	
	DEFINED in MAP-CommonDataTypes	:	122
	USED in MAP-MS-DataTypes	:	97 323
	USED in MAP-CommonDataTypes	:	18
	USED in MAP-CH-DataTypes	:	44 123
	USED in MAP-SS-DataTypes	:	39 56 80
	kc.....	identifier of Kc	
	DEFINED in MAP-MS-DataTypes	:	159
	Kc.....	type reference OCTET STRING	
	DEFINED in MAP-MS-DataTypes	:	166
	USED in MAP-MS-DataTypes	:	159
	laiFixedLength.....	identifier of [1] LAIFixedLength	
	DEFINED in MAP-CommonDataTypes	:	242
	LAIFixedLength.....	type reference OCTET STRING	
	DEFINED in MAP-CommonDataTypes	:	254
	USED in MAP-CommonDataTypes	:	242
	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	:	124
	lmsi.....	identifier of LMSI	
	DEFINED in MAP-MS-DataTypes	:	144
	lmsi.....	identifier of LMSI	
	DEFINED in MAP-MS-DataTypes	:	568
	lmsi.....	identifier of [1] LMSI	
	DEFINED in MAP-MS-DataTypes	:	609

```

LMSI.....type reference OCTET STRING
  DEFINED in MAP-CommonDataTypes : 216
  USED in MAP-MS-DataTypes : 101 124 144 568 609
  USED in MAP-CommonDataTypes : 29
  USED in MAP-CH-DataTypes : 47 132
  USED in MAP-SM-DataTypes : 34 66 97

lmsi.....identifier of [4] LMSI
  DEFINED in MAP-CH-DataTypes : 132

lmsi.....identifier of LMSI
  DEFINED in MAP-SM-DataTypes : 66

lmsi.....identifier of [1] LMSI
  DEFINED in MAP-SM-DataTypes : 97

locationInformation.....identifier of [0] LocationInformation
  DEFINED in MAP-MS-DataTypes : 620

locationInformation.....identifier of [0] NULL
  DEFINED in MAP-MS-DataTypes : 626

LocationInformation.....type reference SEQUENCE
  DEFINED in MAP-MS-DataTypes : 631
  USED in MAP-MS-DataTypes : 63 620

locationInfoWithLMSI.....identifier of [0] LocationInfoWithLMSI
  DEFINED in MAP-SM-DataTypes : 60

LocationInfoWithLMSI.....type reference SEQUENCE
  DEFINED in MAP-SM-DataTypes : 64
  USED in MAP-SM-DataTypes : 60

locationNumber.....identifier of [2] LocationNumber
  DEFINED in MAP-MS-DataTypes : 635

LocationNumber.....type reference OCTET STRING
  DEFINED in MAP-MS-DataTypes : 659
  USED in MAP-MS-DataTypes : 635

mah.....value reference SS-Code, '00110010'B
  DEFINED in MAP-SS-Code : 59

MAP-BS-Code.....module reference
  DEFINED in MAP-BS-Code : 1
  USED in MAP-MS-DataTypes : 86
  USED in MAP-CommonDataTypes : 50

MAP-CallHandlingOperations.....module reference
  DEFINED in MAP-CallHandlingOperat : 1
  USED in MAP-Protocol : 47

MAP-CH-DataTypes.....module reference
  DEFINED in MAP-CH-DataTypes : 1
  USED in MAP-CallHandlingOperat : 49

MAP-CommonDataTypes.....module reference
  DEFINED in MAP-CommonDataTypes : 1
  USED in MAP-MobileServiceOpera : 99
  USED in MAP-OperationAndMainte : 44
  USED in MAP-MS-DataTypes : 106
  USED in MAP-OM-DataTypes : 23
  USED in MAP-CH-DataTypes : 49
  USED in MAP-SS-DataTypes : 41
  USED in MAP-SM-DataTypes : 35
  USED in MAP-ER-DataTypes : 56

MAP-Errors.....module reference
  DEFINED in MAP-Errors : 1
  USED in MAP-Protocol : 120
  USED in MAP-MobileServiceOpera : 65
  USED in MAP-OperationAndMainte : 30
  USED in MAP-CallHandlingOperat : 40
  USED in MAP-SupplementaryServi : 49
  USED in MAP-ShortMessageServic : 42

MAP-ER-DataTypes.....module reference
  DEFINED in MAP-ER-DataTypes : 1
  USED in MAP-Errors : 113
  USED in MAP-SM-DataTypes : 40

MAP-EXTENSION.....information object class reference CLASS
  DEFINED in MAP-ExtensionDataTypes : 22
  USED in MAP-ExtensionDataTypes : 41 43 48

```

```

MAP-ExtensionDataTypes.....module reference
    DEFINED in MAP-ExtensionDataTypes :    1
    USED in MAP-MS-DataTypes           :   111
    USED in MAP-CommonDataTypes        :    55
    USED in MAP-OM-DataTypes           :    28
    USED in MAP-CH-DataTypes           :    54
    USED in MAP-SM-DataTypes           :    45
    USED in MAP-ER-DataTypes           :    66

MAP-MobileServiceOperations.....module reference
    DEFINED in MAP-MobileServiceOpera :    1
    USED in MAP-Protocol               :    31

MAP-MS-DataTypes.....module reference
    DEFINED in MAP-MS-DataTypes        :    1
    USED in MAP-MobileServiceOpera     :    92
    USED in MAP-CH-DataTypes           :    33

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               :    39

MAP-Protocol.....module reference
    DEFINED in MAP-Protocol            :    1

MAP-ShortMessageServiceOperations.....module reference
    DEFINED in MAP-ShortMessageServic :    1
    USED in MAP-Protocol               :    74

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     :    66
    USED in MAP-MS-DataTypes           :    81
    USED in MAP-SS-DataTypes           :    46
    USED in MAP-ER-DataTypes           :    61

MAP-SS-DataTypes.....module reference
    DEFINED in MAP-SS-DataTypes        :    1
    USED in MAP-SupplementaryServi     :    61
    USED in MAP-Errors                 :    79
    USED in MAP-MS-DataTypes           :    76
    USED in MAP-CH-DataTypes           :    39
    USED in MAP-ER-DataTypes           :    49

MAP-SupplementaryServiceOperations.....module reference
    DEFINED in MAP-SupplementaryServi :    1
    USED in MAP-Protocol               :    62

MAP-TS-Code.....module reference
    DEFINED in MAP-TS-Code             :    1
    USED in MAP-MS-DataTypes           :    91
    USED in MAP-CommonDataTypes        :    44

maxAddressLength.....value reference INTEGER, 20
    DEFINED in MAP-CommonDataTypes     :   114
    USED in MAP-CommonDataTypes        :    73

maximumentitledPriority.....identifier of EMLPP-Priority
    DEFINED in MAP-MS-DataTypes        :   290

maxISDN-AddressLength.....value reference INTEGER, 9
    DEFINED in MAP-CommonDataTypes     :   120
    USED in MAP-CommonDataTypes        :   117

maxISDN-SubaddressLength.....value reference INTEGER, 21
    DEFINED in MAP-CommonDataTypes     :   160
    USED in MAP-CommonDataTypes        :   123

maxNumOfBasicServiceGroups.....value reference INTEGER, 13
    DEFINED in MAP-SS-DataTypes        :   215
    USED in MAP-SS-DataTypes           :    73   127   212

maxNumOfBasicServices.....value reference INTEGER, 70
    DEFINED in MAP-MS-DataTypes        :   506
    USED in MAP-MS-DataTypes           :   503

```

maxNumOfBearerServices.....value	reference	INTEGER, 50
DEFINED in MAP-MS-DataTypes	:	239
USED in MAP-MS-DataTypes	:	236
maxNumOfCamelTDPData.....value	reference	INTEGER, 10
DEFINED in MAP-MS-DataTypes	:	527
USED in MAP-MS-DataTypes	:	48 524
USED in MAP-CH-DataTypes	:	26 189
maxNumOfCUG.....value	reference	INTEGER, 10
DEFINED in MAP-MS-DataTypes	:	427
USED in MAP-MS-DataTypes	:	406
maxNumOfExt-BasicServiceGroups.....value	reference	INTEGER, 32
DEFINED in MAP-MS-DataTypes	:	435
USED in MAP-MS-DataTypes	:	316 391 429 432
maxNumOfHLR-Id.....value	reference	INTEGER, 50
DEFINED in MAP-CommonDataTypes	:	214
USED in MAP-CommonDataTypes	:	211
maxNumOfPrivateExtensions.....value	reference	INTEGER, 10
DEFINED in MAP-ExtensionDataTypes	:	46
USED in MAP-ExtensionDataTypes	:	37
maxNumOfSS.....value	reference	INTEGER, 30
DEFINED in MAP-SS-DataTypes	:	207
USED in MAP-MS-DataTypes	:	73 279
USED in MAP-SS-DataTypes	:	30 204 209
maxNumOfTeleservices.....value	reference	INTEGER, 20
DEFINED in MAP-MS-DataTypes	:	244
USED in MAP-MS-DataTypes	:	241
maxNumOfVBSGroupIds.....value	reference	INTEGER, 50
DEFINED in MAP-MS-DataTypes	:	586
USED in MAP-MS-DataTypes	:	580
maxNumOfVGCSGroupIds.....value	reference	INTEGER, 50
DEFINED in MAP-MS-DataTypes	:	588
USED in MAP-MS-DataTypes	:	583
maxNumOfZoneCodes.....value	reference	INTEGER, 10
DEFINED in MAP-MS-DataTypes	:	470
USED in MAP-MS-DataTypes	:	43 464
maxSignalInfoLength.....value	reference	INTEGER, 200
DEFINED in MAP-CommonDataTypes	:	173
USED in MAP-CommonDataTypes	:	21 171
maxUSSD-StringLength.....value	reference	INTEGER, 160
DEFINED in MAP-SS-DataTypes	:	190
USED in MAP-SS-DataTypes	:	186
mcef-Set.....identifier of Named Number, 2		
DEFINED in MAP-SM-DataTypes	:	140
mci.....value	reference	SS-Code, '00010101'B
DEFINED in MAP-SS-Code	:	36
memoryAvailable.....identifier of Named Number, 1		
DEFINED in MAP-SM-DataTypes	:	151
memoryCapacityExceeded.....identifier of Named Number, 0		
DEFINED in MAP-SM-DataTypes	:	116
memoryCapacityExceeded.....identifier of Named Number, 0		
DEFINED in MAP-ER-DataTypes	:	120
MessageType.....type	reference	CHOICE
DEFINED in TCAPMessages	:	51
USED in TCAPMessages	:	47
messageWaitingListFull.....value	reference	MessageWaitingListFull, CHOICE
VALUE		
DEFINED in MAP-Protocol	:	292
MessageWaitingListFull.....type	reference	ERROR
DEFINED in MAP-Errors	:	313
USED in MAP-Protocol	:	118 292
USED in MAP-ShortMessageServic	:	40 121
USED in MAP-Errors	:	69
messageWaitListFullParam.....identifier of MessageWaitListFullParam		
DEFINED in MAP-Errors	:	315

```

MessageWaitListFullParam.....type reference SEQUENCE
    DEFINED in MAP-ER-DataTypes      :    234
    USED in MAP-Errors               :    110    315
    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

mnrf-Set.....identifier of Named Number, 1
    DEFINED in MAP-SM-DataTypes      :    139

moreMessagesToSend.....identifier of NULL
    DEFINED in MAP-SM-DataTypes      :     86

mo-forwardSM.....value reference MO-ForwardSM, CHOICE VALUE
    DEFINED in MAP-Protocol           :    200

MO-ForwardSM.....type reference OPERATION
    DEFINED in MAP-ShortMessageServic :     81
    USED in MAP-Protocol              :     68    200
    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       :     70
    USED in MAP-ShortMessageServic    :     48    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       :     77
    USED in MAP-ShortMessageServic    :     49    85
    USED in MAP-SM-DataTypes          :     17

msc-AreaRestricted.....identifier of Named Number, 0
    DEFINED in MAP-MS-DataTypes       :    484

msc-Number.....identifier of [1] ISDN-AddressString
    DEFINED in MAP-MS-DataTypes       :    123

msc-Number.....identifier of [1] ISDN-AddressString
    DEFINED in MAP-CH-DataTypes       :    130

msc-Number.....identifier of [1] ISDN-AddressString
    DEFINED in MAP-SM-DataTypes       :     65

msisdn.....identifier of ISDN-AddressString
    DEFINED in MAP-OperationAndMainte :     79

msisdn.....identifier of [1] ISDN-AddressString
    DEFINED in MAP-MS-DataTypes       :    211

msisdn.....identifier of [1] ISDN-AddressString
    DEFINED in MAP-MS-DataTypes       :    690

msisdn.....identifier of [0] ISDN-AddressString
    DEFINED in MAP-CH-DataTypes       :     69

msisdn.....identifier of [2] ISDN-AddressString
    DEFINED in MAP-CH-DataTypes       :    131

msisdn.....identifier of [0] ISDN-AddressString
    DEFINED in MAP-SM-DataTypes       :     52

msisdn.....identifier of [2] ISDN-AddressString
    DEFINED in MAP-SM-DataTypes       :    102

msisdn.....identifier of ISDN-AddressString
    DEFINED in MAP-SM-DataTypes       :    107

msisdn.....identifier of ISDN-AddressString

```

```

DEFINED in MAP-SM-DataTypes      :      127

msNotReachable.....identifier of NULL
DEFINED in MAP-MS-DataTypes      :      574

msPurged.....identifier of Named Number, 0
DEFINED in MAP-MS-DataTypes      :      669

ms-Present.....identifier of Named Number, 0
DEFINED in MAP-SM-DataTypes      :      150

mt-forwardSM.....value reference MT-ForwardSM, CHOICE VALUE
DEFINED in MAP-Protocol          :      201

MT-ForwardSM.....type reference OPERATION
DEFINED in MAP-ShortMessageServic :      93
USED in MAP-Protocol             :      69      201
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      :      82
USED in MAP-ShortMessageServic :      50      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      :      90
USED in MAP-ShortMessageServic :      51      97
USED in MAP-SM-DataTypes      :      19

multipleECT-Barred.....identifier of Named Number, 14
DEFINED in MAP-MS-DataTypes      :      267

multiPTY.....value reference SS-Code, '01010001'B
DEFINED in MAP-SS-Code          :      77

mw-Status.....identifier of MW-Status
DEFINED in MAP-SM-DataTypes      :      133

MW-Status.....type reference BIT STRING
DEFINED in MAP-SM-DataTypes      :      137
USED in MAP-SM-DataTypes      :      133

negativePW-Check.....value reference NegativePW-Check, CHOICE VALUE
DEFINED in MAP-Protocol          :      283

NegativePW-Check.....type reference ERROR
DEFINED in MAP-Errors            :      297
USED in MAP-Protocol             :      114      283
USED in MAP-SupplementaryServi :      42      125      144      216
USED in MAP-Errors              :      63

netDetNotReachable.....identifier of NotReachableReason
DEFINED in MAP-MS-DataTypes      :      665

NetworkResource.....type reference ENUMERATED
DEFINED in MAP-CommonDataTypes  :      228
USED in MAP-CommonDataTypes  :      31
USED in MAP-ER-DataTypes      :      55      143      150

networkResource.....identifier of NetworkResource
DEFINED in MAP-ER-DataTypes      :      143

networkResource.....identifier of NetworkResource
DEFINED in MAP-ER-DataTypes      :      150

networkSignalInfo.....identifier of [10] ExternalSignalInfo
DEFINED in MAP-CH-DataTypes      :      79

networkSignalInfo.....identifier of [6] ExternalSignalInfo
DEFINED in MAP-CH-DataTypes      :      134

newPassword.....identifier of Password
DEFINED in MAP-SupplementaryServi :      208

newPasswordsMismatch.....identifier of Named Number, 2
DEFINED in MAP-ER-DataTypes      :      116

noCUG-Restrictions.....identifier of Named Number, 0
DEFINED in MAP-MS-DataTypes      :      423

```

```

noHandoverNumberAvailable.....value reference NoHandoverNumberAvailable, CHOICE
VALUE
    DEFINED in MAP-Protocol          :    245

NoHandoverNumberAvailable.....type reference ERROR
    DEFINED in MAP-Errors             :    202
    USED in MAP-Protocol              :    93    245
    USED in MAP-MobileServiceOpera   :    63    177
    USED in MAP-Errors                :    34

noReply.....identifier of Named Number, 2
    DEFINED in MAP-CH-DataTypes       :    98

noReplyConditionTime.....identifier of [7] Ext-NoRepCondTime
    DEFINED in MAP-MS-DataTypes       :    325

noReplyConditionTime.....identifier of [5] NoReplyConditionTime
    DEFINED in MAP-SS-DataTypes       :    57

NoReplyConditionTime.....type reference INTEGER
    DEFINED in MAP-SS-DataTypes       :    60
    USED in MAP-SS-DataTypes          :    28    57    82

noReplyConditionTime.....identifier of [7] NoReplyConditionTime
    DEFINED in MAP-SS-DataTypes       :    82

noRoamingNbParam.....identifier of NoRoamingNbParam
    DEFINED in MAP-Errors             :    219

NoRoamingNbParam.....type reference SEQUENCE
    DEFINED in MAP-ER-DataTypes       :    202
    USED in MAP-Errors                :    99    219
    USED in MAP-ER-DataTypes          :    33

noRoamingNumberAvailable.....value reference NoRoamingNumberAvailable, CHOICE
VALUE
    DEFINED in MAP-Protocol          :    258

NoRoamingNumberAvailable.....type reference ERROR
    DEFINED in MAP-Errors             :    217
    USED in MAP-Protocol              :    97    258
    USED in MAP-CallHandlingOperat   :    32    89
    USED in MAP-Errors                :    42

noSM-RP-DA.....identifier of [5] NULL
    DEFINED in MAP-SM-DataTypes       :    99

noSM-RP-OA.....identifier of [5] NULL
    DEFINED in MAP-SM-DataTypes       :    104

noSubscriberReply.....value reference NoSubscriberReply, CHOICE VALUE
    DEFINED in MAP-Protocol          :    261

NoSubscriberReply.....type reference ERROR
    DEFINED in MAP-Errors             :    233
    USED in MAP-Protocol              :    100    261
    USED in MAP-CallHandlingOperat   :    35    72
    USED in MAP-Errors                :    44

noSubscriberReplyParam.....identifier of NoSubscriberReplyParam
    DEFINED in MAP-Errors             :    235

NoSubscriberReplyParam.....type reference SEQUENCE
    DEFINED in MAP-ER-DataTypes       :    214
    USED in MAP-Errors                :    103    235
    USED in MAP-ER-DataTypes          :    36

notProvidedFromVLR.....identifier of [2] NULL
    DEFINED in MAP-MS-DataTypes       :    666

notReachable.....identifier of Named Number, 0
    DEFINED in MAP-CH-DataTypes       :    96

NotReachableReason.....type reference ENUMERATED
    DEFINED in MAP-MS-DataTypes       :    668
    USED in MAP-MS-DataTypes          :    665

notRegistered.....identifier of Named Number, 3
    DEFINED in MAP-MS-DataTypes       :    672

not-derivable.....identifier of NULL
    DEFINED in TCAPMessages           :    168

numberChanged.....value reference NumberChanged, CHOICE VALUE

```

DEFINED in MAP-Protocol	:	226	
NumberChanged.....	type reference	ERROR	
DEFINED in MAP-Errors	:	153	
USED in MAP-Protocol	:	84	226
USED in MAP-CallHandlingOperat	:	29	67
USED in MAP-Errors	:	21	
numberChangedParam.....	identifier of	NumberChangedParam	
DEFINED in MAP-Errors	:	155	
NumberChangedParam.....	type reference	SEQUENCE	
DEFINED in MAP-ER-DataTypes	:	174	
USED in MAP-Errors	:	91	155
USED in MAP-ER-DataTypes	:	26	
NumberOfForwarding.....	type reference	INTEGER	
DEFINED in MAP-CH-DataTypes	:	66	
USED in MAP-CH-DataTypes	:	20	71
numberOfForwarding.....	identifier of [2]	NumberOfForwarding	
DEFINED in MAP-CH-DataTypes	:	71	
numberOfPW-AttemptsViolation.....	value reference	NumberOfPW-AttemptsViolation, CHOICE	
VALUE			
DEFINED in MAP-Protocol	:	284	
NumberOfPW-AttemptsViolation.....	type reference	ERROR	
DEFINED in MAP-Errors	:	299	
USED in MAP-Protocol	:	115	284
USED in MAP-SupplementaryServi	:	43	126 145 217
USED in MAP-Errors	:	64	
odb-Data.....	identifier of [8]	ODB-Data	
DEFINED in MAP-MS-DataTypes	:	221	
ODB-Data.....	type reference	SEQUENCE	
DEFINED in MAP-MS-DataTypes	:	246	
USED in MAP-MS-DataTypes	:	40	221
odb-GeneralData.....	identifier of	ODB-GeneralData	
DEFINED in MAP-MS-DataTypes	:	247	
ODB-GeneralData.....	type reference	BIT STRING	
DEFINED in MAP-MS-DataTypes	:	252	
USED in MAP-MS-DataTypes	:	247	476
odb-GeneralData.....	identifier of [4]	ODB-GeneralData	
DEFINED in MAP-MS-DataTypes	:	476	
odb-HPLMN-Data.....	identifier of	ODB-HPLMN-Data	
DEFINED in MAP-MS-DataTypes	:	248	
ODB-HPLMN-Data.....	type reference	BIT STRING	
DEFINED in MAP-MS-DataTypes	:	271	
USED in MAP-MS-DataTypes	:	248	
omc-Id.....	identifier of [3]	AddressString	
DEFINED in MAP-OM-DataTypes	:	40	
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	:	89	
operatorDeterminedBarring.....	identifier of	Named Number, 1	
DEFINED in MAP-MS-DataTypes	:	234	
operatorDeterminedBarring.....	identifier of	Named Number, 3	
DEFINED in MAP-ER-DataTypes	:	78	
OrigTransactionID.....	type reference	[APPLICATION 8] IMPLICIT	
TransactionID			
DEFINED in TCAPMessages	:	97	
USED in TCAPMessages	:	61	69
or-Capability.....	identifier of [5]	OR-Phase	
DEFINED in MAP-CH-DataTypes	:	74	


```

or-Interrogation.....identifier of [4] NULL
  DEFINED in MAP-CH-DataTypes      :      73

or-Interrogation.....identifier of [10] NULL
  DEFINED in MAP-CH-DataTypes      :     138

or-NotAllowed.....value reference OR-NotAllowed, CHOICE VALUE
  DEFINED in MAP-Protocol          :     264

OR-NotAllowed.....type reference ERROR
  DEFINED in MAP-Errors            :     258
  USED in MAP-Protocol            :      96    264
  USED in MAP-CallHandlingOperat :      27    65    87    98
  USED in MAP-Errors              :      41

or-NotAllowedParam.....identifier of OR-NotAllowedParam
  DEFINED in MAP-Errors            :     260

OR-NotAllowedParam.....type reference SEQUENCE
  DEFINED in MAP-ER-DataTypes      :     166
  USED in MAP-Errors              :     100    260
  USED in MAP-ER-DataTypes        :      24

OR-Phase.....type reference INTEGER
  DEFINED in MAP-CH-DataTypes      :      91
  USED in MAP-CH-DataTypes        :      74

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      :     144

OverrideCategory.....type reference ENUMERATED
  DEFINED in MAP-SS-DataTypes      :     151
  USED in MAP-SS-DataTypes        :      26    144

overrideDisabled.....identifier of Named Number, 1
  DEFINED in MAP-SS-DataTypes      :     153

overrideEnabled.....identifier of Named Number, 0
  DEFINED in MAP-SS-DataTypes      :     152

O-BcsmCamelTDPData.....type reference SEQUENCE
  DEFINED in MAP-MS-DataTypes      :     529
  USED in MAP-MS-DataTypes        :     525

o-BcsmCamelTDPDataList.....identifier of O-BcsmCamelTDPDataList
  DEFINED in MAP-MS-DataTypes      :     520

O-BcsmCamelTDPDataList.....type reference SEQUENCE OF
  DEFINED in MAP-MS-DataTypes      :     524
  USED in MAP-MS-DataTypes        :     520

o-BcsmTriggerDetectionPoint.....identifier of O-BcsmTriggerDetectionPoint
  DEFINED in MAP-MS-DataTypes      :     530

O-BcsmTriggerDetectionPoint.....type reference ENUMERATED
  DEFINED in MAP-MS-DataTypes      :     539
  USED in MAP-MS-DataTypes        :     530

o-CSI.....identifier of [0] O-CSI
  DEFINED in MAP-MS-DataTypes      :     515

O-CSI.....type reference SEQUENCE
  DEFINED in MAP-MS-DataTypes      :     519
  USED in MAP-MS-DataTypes        :      44    515
  USED in MAP-CH-DataTypes        :      32    154    180

o-CSI.....identifier of [5] O-CSI
  DEFINED in MAP-CH-DataTypes      :     154

o-CSI.....identifier of [1] O-CSI
  DEFINED in MAP-CH-DataTypes      :     180

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.....type reference NumericString
DEFINED in MAP-SS-DataTypes      :     192
USED in MAP-SupplementaryServi  :     59    208    225
USED in MAP-SS-DataTypes        :     22

pcs-Extensions.....identifier of [1] PCS-Extensions
DEFINED in MAP-ExtensionDataTypes :     34

PCS-Extensions.....type reference SEQUENCE
DEFINED in MAP-ExtensionDataTypes :     56
USED in MAP-ExtensionDataTypes    :     34

permanent.....identifier of Named Number, 0
DEFINED in MAP-SS-DataTypes      :     147

phasel.....identifier of Named Number, 0
DEFINED in MAP-MS-DataTypes      :     556

plmn.....identifier of Named Number, 0
DEFINED in MAP-CommonDataTypes   :     229

plmnRoamingNotAllowed.....identifier of Named Number, 0
DEFINED in MAP-ER-DataTypes      :      77

plmn-SpecificBarringType1.....identifier of Named Number, 0
DEFINED in MAP-MS-DataTypes      :     272

plmn-SpecificBarringType2.....identifier of Named Number, 1
DEFINED in MAP-MS-DataTypes      :     273

plmn-SpecificBarringType3.....identifier of Named Number, 2
DEFINED in MAP-MS-DataTypes      :     274

plmn-SpecificBarringType4.....identifier of Named Number, 3
DEFINED in MAP-MS-DataTypes      :     275

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	: 121
plmn-specificSS-2.....value	reference SS-Code, '11110010'B
DEFINED in MAP-SS-Code	: 122
plmn-specificSS-3.....value	reference SS-Code, '11110011'B
DEFINED in MAP-SS-Code	: 123
plmn-specificSS-4.....value	reference SS-Code, '11110100'B
DEFINED in MAP-SS-Code	: 124
plmn-specificSS-5.....value	reference SS-Code, '11110101'B
DEFINED in MAP-SS-Code	: 125
plmn-specificSS-6.....value	reference SS-Code, '11110110'B
DEFINED in MAP-SS-Code	: 126
plmn-specificSS-7.....value	reference SS-Code, '11110111'B
DEFINED in MAP-SS-Code	: 127
plmn-specificSS-8.....value	reference SS-Code, '11111000'B
DEFINED in MAP-SS-Code	: 128
plmn-specificSS-9.....value	reference SS-Code, '11111001'B
DEFINED in MAP-SS-Code	: 129
plmn-specificSS-A.....value	reference SS-Code, '11111010'B
DEFINED in MAP-SS-Code	: 130
plmn-specificSS-B.....value	reference SS-Code, '11111011'B
DEFINED in MAP-SS-Code	: 131
plmn-specificSS-C.....value	reference SS-Code, '11111100'B
DEFINED in MAP-SS-Code	: 132
plmn-specificSS-D.....value	reference SS-Code, '11111101'B
DEFINED in MAP-SS-Code	: 133
plmn-specificSS-E.....value	reference SS-Code, '11111110'B
DEFINED in MAP-SS-Code	: 134
plmn-specificSS-F.....value	reference SS-Code, '11111111'B
DEFINED in MAP-SS-Code	: 135
plmn-specificTS-1.....value	reference TeleserviceCode, '11010001'B
DEFINED in MAP-TS-Code	: 73
plmn-specificTS-2.....value	reference TeleserviceCode, '11010010'B
DEFINED in MAP-TS-Code	: 74
plmn-specificTS-3.....value	reference TeleserviceCode, '11010011'B
DEFINED in MAP-TS-Code	: 75
plmn-specificTS-4.....value	reference TeleserviceCode, '11010100'B
DEFINED in MAP-TS-Code	: 76
plmn-specificTS-5.....value	reference TeleserviceCode, '11010101'B
DEFINED in MAP-TS-Code	: 77
plmn-specificTS-6.....value	reference TeleserviceCode, '11010110'B
DEFINED in MAP-TS-Code	: 78
plmn-specificTS-7.....value	reference TeleserviceCode, '11010111'B
DEFINED in MAP-TS-Code	: 79
plmn-specificTS-8.....value	reference TeleserviceCode, '11011000'B
DEFINED in MAP-TS-Code	: 80

```

plmn-specificTS-9.....value reference TeleserviceCode, '11011001'B
  DEFINED in MAP-TS-Code      :      81

plmn-specificTS-A.....value reference TeleserviceCode, '11011010'B
  DEFINED in MAP-TS-Code      :      82

plmn-specificTS-B.....value reference TeleserviceCode, '11011011'B
  DEFINED in MAP-TS-Code      :      83

plmn-specificTS-C.....value reference TeleserviceCode, '11011100'B
  DEFINED in MAP-TS-Code      :      84

plmn-specificTS-D.....value reference TeleserviceCode, '11011101'B
  DEFINED in MAP-TS-Code      :      85

plmn-specificTS-E.....value reference TeleserviceCode, '11011110'B
  DEFINED in MAP-TS-Code      :      86

plmn-specificTS-F.....value reference TeleserviceCode, '11011111'B
  DEFINED in MAP-TS-Code      :      87

preferentialCUG-Indicator.....identifier of CUG-Index
  DEFINED in MAP-MS-DataTypes  :      439

premiumRateEntertainmentOGCallsBarred..identifier of Named Number, 4
  DEFINED in MAP-MS-DataTypes  :      260

premiumRateInformationOGCallsBarred....identifier of Named Number, 3
  DEFINED in MAP-MS-DataTypes  :      259

prepareHandover.....value reference PrepareHandover, CHOICE VALUE
  DEFINED in MAP-Protocol      :      136

PrepareHandover.....type reference OPERATION
  DEFINED in MAP-MobileServiceOpera :      168
  USED in MAP-Protocol              :      16   136
  USED in MAP-MobileServiceOpera    :      27

prepareHO-Arg.....identifier of PrepareHO-Arg
  DEFINED in MAP-MobileServiceOpera :      170

PrepareHO-Arg.....type reference SEQUENCE
  DEFINED in MAP-MS-DataTypes        :      170
  USED in MAP-MobileServiceOpera     :      74   170
  USED in MAP-MS-DataTypes           :      23

prepareHO-Res.....identifier of PrepareHO-Res
  DEFINED in MAP-MobileServiceOpera :      172

PrepareHO-Res.....type reference SEQUENCE
  DEFINED in MAP-MS-DataTypes        :      176
  USED in MAP-MobileServiceOpera     :      75   172
  USED in MAP-MS-DataTypes           :      24

prepareSubsequentHandover.....value reference PrepareSubsequentHandover, CHOICE
VALUE
  DEFINED in MAP-Protocol            :      140

PrepareSubsequentHandover.....type reference OPERATION
  DEFINED in MAP-MobileServiceOpera :      192
  USED in MAP-Protocol              :      20   140
  USED in MAP-MobileServiceOpera    :      31

prepareSubsequentHO-Arg.....identifier of PrepareSubsequentHO-Arg
  DEFINED in MAP-MobileServiceOpera :      194

PrepareSubsequentHO-Arg.....type reference SEQUENCE
  DEFINED in MAP-MS-DataTypes        :      181
  USED in MAP-MobileServiceOpera     :      76   194
  USED in MAP-MS-DataTypes           :      25

priorityLevel0.....value reference EMLPP-Priority, 0
  DEFINED in MAP-MS-DataTypes        :      303

priorityLevel1.....value reference EMLPP-Priority, 1
  DEFINED in MAP-MS-DataTypes        :      304

priorityLevel2.....value reference EMLPP-Priority, 2
  DEFINED in MAP-MS-DataTypes        :      305

priorityLevel3.....value reference EMLPP-Priority, 3
  DEFINED in MAP-MS-DataTypes        :      306

priorityLevel4.....value reference EMLPP-Priority, 4
  DEFINED in MAP-MS-DataTypes        :      307

```

```

priorityLevelA.....value reference EMLPP-Priority, 6
    DEFINED in MAP-MS-DataTypes      :    301

priorityLevelB.....value reference EMLPP-Priority, 5
    DEFINED in MAP-MS-DataTypes      :    302

PrivateExtension.....type reference SEQUENCE
    DEFINED in MAP-ExtensionDataTypes :    40
    USED in MAP-ExtensionDataTypes   :    15    38

privateExtensionList.....identifier of [0] PrivateExtensionList
    DEFINED in MAP-ExtensionDataTypes :    33

PrivateExtensionList.....type reference SEQUENCE OF
    DEFINED in MAP-ExtensionDataTypes :    37
    USED in MAP-ExtensionDataTypes   :    33

problem.....identifier of CHOICE
    DEFINED in TCAPMessages          :    169

processAccessSignalling.....value reference ProcessAccessSignalling, CHOICE
VALUE
    DEFINED in MAP-Protocol          :    138

ProcessAccessSignalling.....type reference OPERATION
    DEFINED in MAP-MobileServiceOpera :    184
    USED in MAP-Protocol              :    18    138
    USED in MAP-MobileServiceOpera   :    29

processUnstructuredSS-Request.....value reference ProcessUnstructuredSS-Request, CHOICE
VALUE
    DEFINED in MAP-Protocol          :    189

ProcessUnstructuredSS-Request.....type reference OPERATION
    DEFINED in MAP-SupplementaryServi :    162
    USED in MAP-Protocol              :    57    189
    USED in MAP-SupplementaryServi   :    18

protocolId.....identifier of ProtocolId
    DEFINED in MAP-CommonDataTypes    :    163

ProtocolId.....type reference ENUMERATED
    DEFINED in MAP-CommonDataTypes    :    181
    USED in MAP-CommonDataTypes      :    163

provideRoamingNumber.....value reference ProvideRoamingNumber, CHOICE VALUE
    DEFINED in MAP-Protocol          :    178

ProvideRoamingNumber.....type reference OPERATION
    DEFINED in MAP-CallHandlingOperat :    77
    USED in MAP-Protocol              :    45    178
    USED in MAP-CallHandlingOperat   :    14

provideRoamingNumberArg.....identifier of ProvideRoamingNumberArg
    DEFINED in MAP-CallHandlingOperat :    79

ProvideRoamingNumberArg.....type reference SEQUENCE
    DEFINED in MAP-CH-DataTypes       :    128
    USED in MAP-CallHandlingOperat    :    45    79
    USED in MAP-CH-DataTypes         :    16

provideRoamingNumberRes.....identifier of ProvideRoamingNumberRes
    DEFINED in MAP-CallHandlingOperat :    81

ProvideRoamingNumberRes.....type reference SEQUENCE
    DEFINED in MAP-CH-DataTypes       :    142
    USED in MAP-CallHandlingOperat    :    46    81
    USED in MAP-CH-DataTypes         :    17

provideSubscriberInfo.....value reference ProvideSubscriberInfo, CHOICE
VALUE
    DEFINED in MAP-Protocol          :    209

ProvideSubscriberInfo.....type reference OPERATION
    DEFINED in MAP-MobileServiceOpera :    143
    USED in MAP-Protocol              :    28    209
    USED in MAP-MobileServiceOpera   :    21

provideSubscriberInfoArg.....identifier of ProvideSubscriberInfoArg
    DEFINED in MAP-MobileServiceOpera :    145

ProvideSubscriberInfoArg.....type reference SEQUENCE
    DEFINED in MAP-MS-DataTypes       :    607
    USED in MAP-MobileServiceOpera    :    87    145

```

```

        USED in MAP-MS-DataTypes          :      60

provideSubscriberInfoRes.....identifier of ProvideSubscriberInfoRes
    DEFINED in MAP-MobileServiceOpera    :      147

ProvideSubscriberInfoRes.....type reference SEQUENCE
    DEFINED in MAP-MS-DataTypes          :      614
        USED in MAP-MobileServiceOpera  :      88      147
        USED in MAP-MS-DataTypes        :      61

provisionedSS.....identifier of [7] Ext-SS-InfoList
    DEFINED in MAP-MS-DataTypes          :      220

purgeMS.....value reference PurgeMS, CHOICE VALUE
    DEFINED in MAP-Protocol              :      130

PurgeMS.....type reference OPERATION
    DEFINED in MAP-MobileServiceOpera    :      127
        USED in MAP-Protocol            :      14      130
        USED in MAP-MobileServiceOpera  :      17

purgeMS-Arg.....identifier of PurgeMS-Arg
    DEFINED in MAP-MobileServiceOpera    :      129

PurgeMS-Arg.....type reference SEQUENCE
    DEFINED in MAP-MS-DataTypes          :      137
        USED in MAP-MobileServiceOpera  :      72      129
        USED in MAP-MS-DataTypes        :      19

pvlr.....identifier of Named Number, 3
    DEFINED in MAP-CommonDataTypes       :      232

pw-RegistrationFailure.....value reference PW-RegistrationFailure, CHOICE
VALUE
    DEFINED in MAP-Protocol              :      282

PW-RegistrationFailure.....type reference ERROR
    DEFINED in MAP-Errors                :      293
        USED in MAP-Protocol            :      113      282
        USED in MAP-SupplementaryServi  :      41      215
        USED in MAP-Errors              :      62

pw-RegistrationFailureCause.....identifier of PW-RegistrationFailureCause
    DEFINED in MAP-Errors                :      295

PW-RegistrationFailureCause.....type reference ENUMERATED
    DEFINED in MAP-ER-DataTypes          :      113
        USED in MAP-Errors              :      84      295
        USED in MAP-ER-DataTypes        :      18

p-abortCause.....identifier of P-AbortCause
    DEFINED in TCAPMessages              :      76

P-AbortCause.....type reference [APPLICATION 10] IMPLICIT INTEGER
    DEFINED in TCAPMessages              :      102
        USED in TCAPMessages            :      76

rand.....identifier of RAND
    DEFINED in MAP-MS-DataTypes          :      157

RAND.....type reference OCTET STRING
    DEFINED in MAP-MS-DataTypes          :      162
        USED in MAP-MS-DataTypes        :      157

readyForSM.....value reference ReadyForSM, CHOICE VALUE
    DEFINED in MAP-Protocol              :      205

ReadyForSM.....type reference OPERATION
    DEFINED in MAP-ShortMessageServic    :      136
        USED in MAP-Protocol            :      73      205
        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          :      144
        USED in MAP-ShortMessageServic  :      56      138
        USED in MAP-SM-DataTypes        :      24

reason.....identifier of CHOICE
    DEFINED in TCAPMessages              :      75

regionalSubscNotSupported.....identifier of Named Number, 3
    DEFINED in MAP-MS-DataTypes          :      487

```

```

regionalSubscriptionData.....identifier of [10] ZoneCodeList
    DEFINED in MAP-MS-DataTypes      :    223

regionalSubscriptionIdentifier.....identifier of [5] ZoneCode
    DEFINED in MAP-MS-DataTypes      :    496

regionalSubscriptionResponse.....identifier of [5] RegionalSubscriptionResponse
    DEFINED in MAP-MS-DataTypes      :    477

RegionalSubscriptionResponse.....type reference ENUMERATED
    DEFINED in MAP-MS-DataTypes      :    483
    USED in MAP-MS-DataTypes        :    478    510

regionalSubscriptionResponse.....identifier of [0] RegionalSubscriptionResponse
    DEFINED in MAP-MS-DataTypes      :    509

registerPassword.....value reference RegisterPassword, CHOICE VALUE
    DEFINED in MAP-Protocol          :    193

RegisterPassword.....type reference OPERATION
    DEFINED in MAP-SupplementaryServi :    204
    USED in MAP-Protocol              :    60    193
    USED in MAP-SupplementaryServi    :    21

registerSS.....value reference RegisterSS, CHOICE VALUE
    DEFINED in MAP-Protocol          :    184

RegisterSS.....type reference OPERATION
    DEFINED in MAP-SupplementaryServi :    74
    USED in MAP-Protocol              :    52    184
    USED in MAP-SupplementaryServi    :    13

registerSS-Arg.....identifier of RegisterSS-Arg
    DEFINED in MAP-SupplementaryServi :    76

RegisterSS-Arg.....type reference SEQUENCE
    DEFINED in MAP-SS-DataTypes      :    52
    USED in MAP-SupplementaryServi    :    53    76
    USED in MAP-SS-DataTypes         :    14

reject.....identifier of [4] IMPLICIT Reject
    DEFINED in TCAPMessages          :    128

Reject.....type reference SEQUENCE
    DEFINED in TCAPMessages          :    165
    USED in TCAPMessages             :    128

releaseCall.....identifier of Named Number, 1
    DEFINED in MAP-MS-DataTypes      :    549

reportSM-DeliveryStatus.....value reference ReportSM-DeliveryStatus, CHOICE
VALUE
    DEFINED in MAP-Protocol          :    202

ReportSM-DeliveryStatus.....type reference OPERATION
    DEFINED in MAP-ShortMessageServic :    111
    USED in MAP-Protocol              :    70    202
    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      :    106
    USED in MAP-ShortMessageServic    :    52    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      :    120
    USED in MAP-ShortMessageServic    :    53    115
    USED in MAP-SM-DataTypes         :    21

requestedBasicServiceViolatesCUG-Constraidentifier of Named Number, 5
    DEFINED in MAP-ER-DataTypes      :    104

requestedInfo.....identifier of [2] RequestedInfo
    DEFINED in MAP-MS-DataTypes      :    610

RequestedInfo.....type reference SEQUENCE
    DEFINED in MAP-MS-DataTypes      :    625
    USED in MAP-MS-DataTypes        :    610    678

```

```

requestedInfo.....identifier of [1] RequestedInfo
    DEFINED in MAP-MS-DataTypes      :    678

reset.....value reference Reset, CHOICE VALUE
    DEFINED in MAP-Protocol          :    162

Reset.....type reference OPERATION
    DEFINED in MAP-MobileServiceOpera :    255
    USED in MAP-Protocol              :    25   162
    USED in MAP-MobileServiceOpera :    44

resetArg.....identifier of ResetArg
    DEFINED in MAP-MobileServiceOpera :    257

ResetArg.....type reference SEQUENCE
    DEFINED in MAP-MS-DataTypes      :    561
    USED in MAP-MobileServiceOpera :    84   257
    USED in MAP-MS-DataTypes      :    55

resourceLimitation.....identifier of Named Number, 4
    DEFINED in TCAPMessages          :    107

resourceLimitation.....identifier of Named Number, 3
    DEFINED in TCAPMessages          :    186

restoreData.....value reference RestoreData, CHOICE VALUE
    DEFINED in MAP-Protocol          :    165

RestoreData.....type reference OPERATION
    DEFINED in MAP-MobileServiceOpera :    261
    USED in MAP-Protocol              :    27   165
    USED in MAP-MobileServiceOpera :    46

restoreDataArg.....identifier of RestoreDataArg
    DEFINED in MAP-MobileServiceOpera :    263

RestoreDataArg.....type reference SEQUENCE
    DEFINED in MAP-MS-DataTypes      :    566
    USED in MAP-MobileServiceOpera :    85   263
    USED in MAP-MS-DataTypes      :    56

restoreDataRes.....identifier of RestoreDataRes
    DEFINED in MAP-MobileServiceOpera :    265

RestoreDataRes.....type reference SEQUENCE
    DEFINED in MAP-MS-DataTypes      :    572
    USED in MAP-MobileServiceOpera :    86   265
    USED in MAP-MS-DataTypes      :    57

restrictedArea.....identifier of Named Number, 2
    DEFINED in MAP-MS-DataTypes      :    671

result-RR.....identifier of SEQUENCE
    DEFINED in TCAPMessages          :    146

resumeCallHandling.....value reference ResumeCallHandling, CHOICE VALUE
    DEFINED in MAP-Protocol          :    179

ResumeCallHandling.....type reference OPERATION
    DEFINED in MAP-CallHandlingOperat :    91
    USED in MAP-Protocol              :    46   179
    USED in MAP-CallHandlingOperat :    15

resumeCallHandlingArg.....identifier of ResumeCallHandlingArg
    DEFINED in MAP-CallHandlingOperat :    93

ResumeCallHandlingArg.....type reference SEQUENCE
    DEFINED in MAP-CH-DataTypes      :    148
    USED in MAP-CallHandlingOperat :    47   93
    USED in MAP-CH-DataTypes      :    18

resumeCallHandlingRes.....identifier of ResumeCallHandlingRes
    DEFINED in MAP-CallHandlingOperat :    95

ResumeCallHandlingRes.....type reference SEQUENCE
    DEFINED in MAP-CH-DataTypes      :    158
    USED in MAP-CallHandlingOperat :    48   95
    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

roamingNotAllowed.....value reference RoamingNotAllowed, CHOICE VALUE
  DEFINED in MAP-Protocol : 234

RoamingNotAllowed.....type reference ERROR
  DEFINED in MAP-Errors : 171
  USED in MAP-Protocol : 88 234
  USED in MAP-MobileServiceOpera : 61 117
  USED in MAP-Errors : 27

roamingNotAllowedCause.....identifier of RoamingNotAllowedCause
  DEFINED in MAP-ER-DataTypes : 72

RoamingNotAllowedCause.....type reference ENUMERATED
  DEFINED in MAP-ER-DataTypes : 76
  USED in MAP-ER-DataTypes : 72

roamingNotAllowedParam.....identifier of RoamingNotAllowedParam
  DEFINED in MAP-Errors : 173

RoamingNotAllowedParam.....type reference SEQUENCE
  DEFINED in MAP-ER-DataTypes : 71
  USED in MAP-Errors : 93 173
  USED in MAP-ER-DataTypes : 14

roamingNumber.....identifier of ISDN-AddressString
  DEFINED in MAP-CH-DataTypes : 118

roamingNumber.....identifier of ISDN-AddressString
  DEFINED in MAP-CH-DataTypes : 143

roamingRestrictionDueToUnsupportedFeaturidentifier of [9] NULL
  DEFINED in MAP-MS-DataTypes : 222

roamingRestrictionDueToUnsupportedFeaturidentifier of [4] NULL
  DEFINED in MAP-MS-DataTypes : 495

RoutingInfo.....type reference CHOICE
  DEFINED in MAP-CH-DataTypes : 117
  USED in MAP-CH-DataTypes : 169

routingInfo.....identifier of RoutingInfo
  DEFINED in MAP-CH-DataTypes : 169

routingInfoForSM-Arg.....identifier of RoutingInfoForSM-Arg
  DEFINED in MAP-ShortMessageServic : 68

RoutingInfoForSM-Arg.....type reference SEQUENCE
  DEFINED in MAP-SM-DataTypes : 51
  USED in MAP-ShortMessageServic : 46 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      :      58
    USED in MAP-ShortMessageService  :      47      70
    USED in MAP-SM-DataTypes         :      15

rss.....identifier of Named Number, 7
    DEFINED in MAP-CommonDataTypes   :      236

sc-AddressNotIncluded.....identifier of Named Number, 0
    DEFINED in MAP-SM-DataTypes       :      138

sc-Congestion.....identifier of Named Number, 4
    DEFINED in MAP-ER-DataTypes       :      124

VALUE
sendAuthenticationInfo.....value reference SendAuthenticationInfo, CHOICE
    DEFINED in MAP-Protocol           :      146

SendAuthenticationInfo.....type reference OPERATION
    DEFINED in MAP-MobileServiceOpera :      205
    USED in MAP-Protocol              :      21      146
    USED in MAP-MobileServiceOpera    :      34

sendAuthenticationInfoArg.....identifier of SendAuthenticationInfoArg
    DEFINED in MAP-MobileServiceOpera :      207

SendAuthenticationInfoArg.....type reference IMSI
    DEFINED in MAP-MS-DataTypes       :      189
    USED in MAP-MobileServiceOpera    :      77      207
    USED in MAP-MS-DataTypes          :      28

sendAuthenticationInfoRes.....identifier of SendAuthenticationInfoRes
    DEFINED in MAP-MobileServiceOpera :      209

SendAuthenticationInfoRes.....type reference AuthenticationSetList
    DEFINED in MAP-MS-DataTypes       :      191
    USED in MAP-MobileServiceOpera    :      78      209
    USED in MAP-MS-DataTypes          :      29

sendEndSignal.....value reference SendEndSignal, CHOICE VALUE
    DEFINED in MAP-Protocol           :      137

SendEndSignal.....type reference OPERATION
    DEFINED in MAP-MobileServiceOpera :      179
    USED in MAP-Protocol              :      17      137
    USED in MAP-MobileServiceOpera    :      28

sendIdentification.....value reference SendIdentification, CHOICE VALUE
    DEFINED in MAP-Protocol           :      131

SendIdentification.....type reference OPERATION
    DEFINED in MAP-MobileServiceOpera :      132
    USED in MAP-Protocol              :      15      131
    USED in MAP-MobileServiceOpera    :      18

sendIdentificationRes.....identifier of SendIdentificationRes
    DEFINED in MAP-MobileServiceOpera :      136

SendIdentificationRes.....type reference SEQUENCE
    DEFINED in MAP-MS-DataTypes       :      148
    USED in MAP-MobileServiceOpera    :      73      136
    USED in MAP-MS-DataTypes          :      20

sendIMSI.....value reference SendIMSI, CHOICE VALUE
    DEFINED in MAP-Protocol           :      172

SendIMSI.....type reference OPERATION
    DEFINED in MAP-OperationAndMainte :      77
    USED in MAP-Protocol              :      38      172
    USED in MAP-OperationAndMainte    :      15

sendRoutingInfo.....value reference SendRoutingInfo, CHOICE VALUE
    DEFINED in MAP-Protocol           :      177

SendRoutingInfo.....type reference OPERATION
    DEFINED in MAP-CallHandlingOperat :      55
    USED in MAP-Protocol              :      44      177
    USED in MAP-CallHandlingOperat    :      13

sendRoutingInfoArg.....identifier of SendRoutingInfoArg
    DEFINED in MAP-CallHandlingOperat :      57

SendRoutingInfoArg.....type reference SEQUENCE
    DEFINED in MAP-CH-DataTypes       :      68

```

```

USED in MAP-CallHandlingOperat :    43    57
USED in MAP-CH-DataTypes       :    14

sendRoutingInfoForSM.....value reference SendRoutingInfoForSM, CHOICE VALUE
  DEFINED in MAP-Protocol       :    199

SendRoutingInfoForSM.....type reference OPERATION
  DEFINED in MAP-ShortMessageServic :    66
  USED in MAP-Protocol           :    67    199
  USED in MAP-ShortMessageServic :    13

sendRoutingInfoRes.....identifier of SendRoutingInfoRes
  DEFINED in MAP-CallHandlingOperat :    59

SendRoutingInfoRes.....type reference [3] SEQUENCE
  DEFINED in MAP-CH-DataTypes       :    101
  USED in MAP-CallHandlingOperat    :    44    59
  USED in MAP-CH-DataTypes         :    15

serviceCentreAddress.....identifier of [2] AddressString
  DEFINED in MAP-SM-DataTypes       :    54

serviceCentreAddress.....identifier of AddressString
  DEFINED in MAP-SM-DataTypes       :    108

serviceCentreAddress.....identifier of AddressString
  DEFINED in MAP-SM-DataTypes       :    128

serviceCentreAddressDA.....identifier of [4] AddressString
  DEFINED in MAP-SM-DataTypes       :    98

serviceCentreAddressOA.....identifier of [4] AddressString
  DEFINED in MAP-SM-DataTypes       :    103

serviceGranted.....identifier of Named Number, 0
  DEFINED in MAP-MS-DataTypes       :    233

serviceKey.....identifier of ServiceKey
  DEFINED in MAP-MS-DataTypes       :    531

ServiceKey.....type reference INTEGER
  DEFINED in MAP-MS-DataTypes       :    537
  USED in MAP-MS-DataTypes         :    45    531
  USED in MAP-CH-DataTypes         :    28    194

serviceKey.....identifier of ServiceKey
  DEFINED in MAP-CH-DataTypes       :    194

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

signalInfo.....identifier of SignalInfo
  DEFINED in MAP-CommonDataTypes    :    164

SignalInfo.....type reference OCTET STRING
  DEFINED in MAP-CommonDataTypes    :    171
  USED in MAP-CommonDataTypes       :    20    164
  USED in MAP-SM-DataTypes          :    32    73    78    85    91
  USED in MAP-ER-DataTypes          :    53    130

sm-DeliveryFailure.....value reference SM-DeliveryFailure, CHOICE VALUE
  DEFINED in MAP-Protocol           :    291

SM-DeliveryFailure.....type reference ERROR
  DEFINED in MAP-Errors             :    309
  USED in MAP-Protocol              :    117    291
  USED in MAP-ShortMessageServic    :    39    91    108
  USED in MAP-Errors                :    68

sm-DeliveryFailureCause.....identifier of SM-DeliveryFailureCause
  DEFINED in MAP-Errors             :    311

SM-DeliveryFailureCause.....type reference SEQUENCE
  DEFINED in MAP-ER-DataTypes       :    128
  USED in MAP-Errors                :    85    311
  USED in MAP-ER-DataTypes          :    19

sm-DeliveryOutcome.....identifier of SM-DeliveryOutcome
  DEFINED in MAP-SM-DataTypes       :    109

SM-DeliveryOutcome.....type reference ENUMERATED
  DEFINED in MAP-SM-DataTypes       :    115

```

```

USED in MAP-SM-DataTypes      :      25      109

SM-EnumeratedDeliveryFailureCause.....type reference ENUMERATED
  DEFINED in MAP-ER-DataTypes   :      119
  USED in MAP-ER-DataTypes      :      129

sm-EnumeratedDeliveryFailureCause.....identifier of SM-EnumeratedDeliveryFailureCause
  DEFINED in MAP-ER-DataTypes   :      129

sm-RP-DA.....identifier of SM-RP-DA
  DEFINED in MAP-SM-DataTypes   :      71

sm-RP-DA.....identifier of SM-RP-DA
  DEFINED in MAP-SM-DataTypes   :      83

SM-RP-DA.....type reference CHOICE
  DEFINED in MAP-SM-DataTypes   :      95
  USED in MAP-SM-DataTypes      :      71      83

sm-RP-OA.....identifier of SM-RP-OA
  DEFINED in MAP-SM-DataTypes   :      72

sm-RP-OA.....identifier of SM-RP-OA
  DEFINED in MAP-SM-DataTypes   :      84

SM-RP-OA.....type reference CHOICE
  DEFINED in MAP-SM-DataTypes   :      101
  USED in MAP-SM-DataTypes      :      72      84

sm-RP-PRI.....identifier of [1] BOOLEAN
  DEFINED in MAP-SM-DataTypes   :      53

sm-RP-UI.....identifier of SignalInfo
  DEFINED in MAP-SM-DataTypes   :      73

sm-RP-UI.....identifier of SignalInfo
  DEFINED in MAP-SM-DataTypes   :      78

sm-RP-UI.....identifier of SignalInfo
  DEFINED in MAP-SM-DataTypes   :      85

sm-RP-UI.....identifier of SignalInfo
  DEFINED in MAP-SM-DataTypes   :      91

sres.....identifier of SRES
  DEFINED in MAP-MS-DataTypes   :      158

SRES.....type reference OCTET STRING
  DEFINED in MAP-MS-DataTypes   :      164
  USED in MAP-MS-DataTypes      :      158

ss-AccessBarred.....identifier of Named Number, 5
  DEFINED in MAP-MS-DataTypes   :      261

ss-Code.....identifier of SS-Code
  DEFINED in MAP-SupplementaryServi :      206

ss-Code.....identifier of SS-Code
  DEFINED in MAP-MS-DataTypes   :      310

ss-Code.....identifier of SS-Code
  DEFINED in MAP-MS-DataTypes   :      385

ss-Code.....identifier of SS-Code
  DEFINED in MAP-MS-DataTypes   :      457

ss-Code.....identifier of SS-Code
  DEFINED in MAP-SS-DataTypes   :      53

ss-Code.....identifier of SS-Code
  DEFINED in MAP-SS-DataTypes   :      68

ss-Code.....identifier of SS-Code
  DEFINED in MAP-SS-DataTypes   :      122

ss-Code.....identifier of SS-Code
  DEFINED in MAP-SS-DataTypes   :      136

ss-Code.....identifier of SS-Code
  DEFINED in MAP-SS-DataTypes   :      156

SS-Code.....type reference OCTET STRING
  DEFINED in MAP-SS-Code        :      11
  USED in MAP-SupplementaryServi :      65      206
  USED in MAP-MS-DataTypes      :      80      310      385      457

```

43	USED in MAP-SS-DataTypes	:	45	53	68	122	136	156	205	
	USED in MAP-SS-Code	:	21	25	28	30	32	34	36	41
66			45	47	49	51	54	57	59	63
91			68	70	74	77	80	83	86	89
114			94	97	101	103	105	107	109	112
127			116	120	121	122	123	124	125	126
137			128	129	130	131	132	133	134	135
			140							
	USED in MAP-ER-DataTypes	:	60	108						
	ss-Code.....	identifier of [1] SS-Code								
	DEFINED in MAP-ER-DataTypes	:	108							
	ss-Data.....	identifier of [3] Ext-SS-Data								
	DEFINED in MAP-MS-DataTypes	:	286							
	ss-Data.....	identifier of [3] SS-Data								
	DEFINED in MAP-SS-DataTypes	:	65							
	SS-Data.....	type reference SEQUENCE								
	DEFINED in MAP-SS-DataTypes	:	135							
	USED in MAP-SS-DataTypes	:	31	65						
	ss-ErrorStatus.....	value reference SS-ErrorStatus, CHOICE VALUE								
	DEFINED in MAP-Protocol	:	276							
	SS-ErrorStatus.....	type reference ERROR								
	DEFINED in MAP-Errors	:	275							
	USED in MAP-Protocol	:	107	276						
	USED in MAP-SupplementaryServi	:	37	88	105	122	142			
	USED in MAP-Errors	:	56							
	ss-ForBS.....	identifier of SS-ForBS-Code								
	DEFINED in MAP-SupplementaryServi	:	93							
	ss-ForBS.....	identifier of SS-ForBS-Code								
	DEFINED in MAP-SupplementaryServi	:	110							
	ss-ForBS.....	identifier of SS-ForBS-Code								
	DEFINED in MAP-SupplementaryServi	:	130							
	ss-ForBS.....	identifier of SS-ForBS-Code								
	DEFINED in MAP-SupplementaryServi	:	149							
	SS-ForBS-Code.....	type reference SEQUENCE								
	DEFINED in MAP-SS-DataTypes	:	155							
	USED in MAP-SupplementaryServi	:	55	93	110	130	149			
	USED in MAP-SS-DataTypes	:	18							
	ss-Incompatibility.....	value reference SS-Incompatibility, CHOICE VALUE								
	DEFINED in MAP-Protocol	:	279							
	SS-Incompatibility.....	type reference ERROR								
	DEFINED in MAP-Errors	:	284							
	USED in MAP-Protocol	:	110	279						
	USED in MAP-SupplementaryServi	:	40	89	124					
	USED in MAP-Errors	:	59							
	ss-IncompatibilityCause.....	identifier of SS-IncompatibilityCause								
	DEFINED in MAP-Errors	:	286							
	SS-IncompatibilityCause.....	type reference SEQUENCE								
	DEFINED in MAP-ER-DataTypes	:	107							
	USED in MAP-Errors	:	83	286						
	USED in MAP-ER-DataTypes	:	17							
	ss-Info.....	identifier of SS-Info								
	DEFINED in MAP-SupplementaryServi	:	78							
	ss-Info.....	identifier of SS-Info								
	DEFINED in MAP-SupplementaryServi	:	95							
	ss-Info.....	identifier of SS-Info								
	DEFINED in MAP-SupplementaryServi	:	112							
	ss-Info.....	identifier of SS-Info								
	DEFINED in MAP-SupplementaryServi	:	132							
	SS-Info.....	type reference CHOICE								

DEFINED in MAP-SS-DataTypes	:	62				
USED in MAP-SupplementaryServi	:	54	78	95	112	132
USED in MAP-SS-DataTypes	:	15	210			
SS-InfoList.....	type reference SEQUENCE OF					
DEFINED in MAP-SS-DataTypes	:	209				
USED in MAP-SS-DataTypes	:	25				
ss-List.....	identifier of [3] SS-List					
DEFINED in MAP-MS-DataTypes	:	475				
ss-List.....	identifier of [2] SS-List					
DEFINED in MAP-MS-DataTypes	:	494				
ss-List.....	identifier of [1] SS-List					
DEFINED in MAP-CH-DataTypes	:	110				
SS-List.....	type reference SEQUENCE OF					
DEFINED in MAP-SS-DataTypes	:	204				
USED in MAP-MS-DataTypes	:	75	475	494		
USED in MAP-CH-DataTypes	:	38	110			
USED in MAP-SS-DataTypes	:	24				
ss-NotAvailable.....	value reference SS-NotAvailable, CHOICE VALUE					
DEFINED in MAP-Protocol	:	277				
SS-NotAvailable.....	type reference ERROR					
DEFINED in MAP-Errors	:	280				
USED in MAP-Protocol	:	108	277			
USED in MAP-SupplementaryServi	:	38	160			
USED in MAP-Errors	:	57				
ss-Status.....	identifier of SS-Status					
DEFINED in MAP-Errors	:	277				
ss-Status.....	identifier of [4] Ext-SS-Status					
DEFINED in MAP-MS-DataTypes	:	321				
ss-Status.....	identifier of [4] Ext-SS-Status					
DEFINED in MAP-MS-DataTypes	:	396				
ss-Status.....	identifier of [4] Ext-SS-Status					
DEFINED in MAP-MS-DataTypes	:	458				
ss-Status.....	identifier of [4] SS-Status					
DEFINED in MAP-SS-DataTypes	:	78				
SS-Status.....	type reference OCTET STRING					
DEFINED in MAP-SS-DataTypes	:	85				
USED in MAP-Errors	:	78	277			
USED in MAP-SS-DataTypes	:	16	78	132	137	161 166
USED in MAP-ER-DataTypes	:	48	110			
ss-Status.....	identifier of [4] SS-Status					
DEFINED in MAP-SS-DataTypes	:	132				
ss-Status.....	identifier of [4] SS-Status					
DEFINED in MAP-SS-DataTypes	:	137				
ss-Status.....	identifier of SS-Status					
DEFINED in MAP-SS-DataTypes	:	161				
ss-Status.....	identifier of [0] SS-Status					
DEFINED in MAP-SS-DataTypes	:	166				
ss-Status.....	identifier of [4] SS-Status					
DEFINED in MAP-ER-DataTypes	:	110				
ss-SubscriptionOption.....	identifier of SS-SubscriptionOption					
DEFINED in MAP-MS-DataTypes	:	459				
ss-SubscriptionOption.....	identifier of SS-SubscriptionOption					
DEFINED in MAP-SS-DataTypes	:	138				
SS-SubscriptionOption.....	type reference CHOICE					
DEFINED in MAP-SS-DataTypes	:	142				
USED in MAP-MS-DataTypes	:	74	459			
USED in MAP-SS-DataTypes	:	17	138			
ss-SubscriptionViolation.....	value reference SS-SubscriptionViolation, CHOICE					
VALUE						
DEFINED in MAP-Protocol	:	278				
SS-SubscriptionViolation.....	type reference ERROR					
DEFINED in MAP-Errors	:	282				

```

        USED in MAP-Protocol           :    109    278
        USED in MAP-SupplementaryServi :    39    123    143    214
        USED in MAP-Errors              :    58

storedMSISDN.....identifier of ISDN-AddressString
    DEFINED in MAP-SM-DataTypes        :    121

storedMSISDN.....identifier of ISDN-AddressString
    DEFINED in MAP-SM-DataTypes        :    132

subBusyForMT-SMS-Param.....identifier of SubBusyForMT-SMS-Param
    DEFINED in MAP-Errors              :    306

SubBusyForMT-SMS-Param.....type reference SEQUENCE
    DEFINED in MAP-ER-DataTypes        :    230
    USED in MAP-Errors                 :    109    306
    USED in MAP-ER-DataTypes          :    40

subscriberBusyForMT-SMS.....value reference SubscriberBusyForMT-SMS, CHOICE
VALUE
    DEFINED in MAP-Protocol            :    290

SubscriberBusyForMT-SMS.....type reference ERROR
    DEFINED in MAP-Errors              :    304
    USED in MAP-Protocol               :    116    290
    USED in MAP-ShortMessageServic    :    38    107
    USED in MAP-Errors                 :    67

SubscriberData.....type reference SEQUENCE
    DEFINED in MAP-MS-DataTypes        :    210
    USED in MAP-MS-DataTypes          :    39    206

SubscriberId.....type reference CHOICE
    DEFINED in MAP-CommonDataTypes    :    196
    USED in MAP-CommonDataTypes       :    26

subscriberIdentity.....identifier of [0] SubscriberIdentity
    DEFINED in MAP-MS-DataTypes        :    677

SubscriberIdentity.....type reference CHOICE
    DEFINED in MAP-MS-DataTypes        :    688
    USED in MAP-MS-DataTypes          :    677

subscriberInfo.....identifier of SubscriberInfo
    DEFINED in MAP-MS-DataTypes        :    615

SubscriberInfo.....type reference SEQUENCE
    DEFINED in MAP-MS-DataTypes        :    619
    USED in MAP-MS-DataTypes          :    62    615    684
    USED in MAP-CH-DataTypes          :    27    109

subscriberInfo.....identifier of SubscriberInfo
    DEFINED in MAP-MS-DataTypes        :    684

subscriberInfo.....identifier of [7] SubscriberInfo
    DEFINED in MAP-CH-DataTypes        :    109

subscriberNotMemberOfCUG.....identifier of Named Number, 1
    DEFINED in MAP-ER-DataTypes        :    103

subscriberNotSC-Subscriber.....identifier of Named Number, 6
    DEFINED in MAP-ER-DataTypes        :    126

subscriberState.....identifier of [1] SubscriberState
    DEFINED in MAP-MS-DataTypes        :    621

subscriberState.....identifier of [1] NULL
    DEFINED in MAP-MS-DataTypes        :    627

SubscriberState.....type reference CHOICE
    DEFINED in MAP-MS-DataTypes        :    662
    USED in MAP-MS-DataTypes          :    64    621

subscriberStatus.....identifier of [3] SubscriberStatus
    DEFINED in MAP-MS-DataTypes        :    213

SubscriberStatus.....type reference ENUMERATED
    DEFINED in MAP-MS-DataTypes        :    232
    USED in MAP-MS-DataTypes          :    41    213

subsequentHandoverFailure.....value reference SubsequentHandoverFailure, CHOICE
VALUE
    DEFINED in MAP-Protocol            :    247

SubsequentHandoverFailure.....type reference ERROR

```

```

DEFINED in MAP-Errors : 204
USED in MAP-Protocol : 94 247
USED in MAP-MobileServiceOpera : 64 201
USED in MAP-Errors : 35

successfulTransfer.....identifier of Named Number, 2
DEFINED in MAP-SM-DataTypes : 118

supportedCamelPhases.....identifier of [6] SupportedCamelPhases
DEFINED in MAP-MS-DataTypes : 479

SupportedCamelPhases.....type reference BIT STRING
DEFINED in MAP-MS-DataTypes : 555
USED in MAP-MS-DataTypes : 47 479
USED in MAP-CH-DataTypes : 30 163

supportedCamelPhases.....identifier of SupportedCamelPhases
DEFINED in MAP-CH-DataTypes : 163

suppressionOfAnnouncement.....identifier of [12] SuppressionOfAnnouncement
DEFINED in MAP-CH-DataTypes : 81

SuppressionOfAnnouncement.....type reference NULL
DEFINED in MAP-CH-DataTypes : 85
USED in MAP-CH-DataTypes : 21 81 135

suppressionOfAnnouncement.....identifier of [7] SuppressionOfAnnouncement
DEFINED in MAP-CH-DataTypes : 135

suppress-T-CSI.....identifier of NULL
DEFINED in MAP-CH-DataTypes : 164

systemFailure.....value reference SystemFailure, CHOICE VALUE
DEFINED in MAP-Protocol : 217

SystemFailure.....type reference ERROR
DEFINED in MAP-Errors : 121
USED in MAP-Protocol : 79 217
USED in MAP-MobileServiceOpera : 54 113 160 174 212 225 267
USED in MAP-OperationAndMainte : 23 57 71
USED in MAP-CallHandlingOperat : 23 61 83
USED in MAP-SupplementaryServi : 30 81 98 115 135 153 168 181

195

USED in MAP-ShortMessageServic : 27 72 88 100 128
USED in MAP-Errors : 14

systemFailureParam.....identifier of SystemFailureParam
DEFINED in MAP-Errors : 123

SystemFailureParam.....type reference CHOICE
DEFINED in MAP-ER-DataTypes : 142
USED in MAP-Errors : 86 123
USED in MAP-ER-DataTypes : 20

targetCellId.....identifier of GlobalCellId
DEFINED in MAP-MS-DataTypes : 171

targetCellId.....identifier of GlobalCellId
DEFINED in MAP-MS-DataTypes : 182

targetMSC-Number.....identifier of ISDN-AddressString
DEFINED in MAP-MS-DataTypes : 183

TBCD-STRING.....type reference OCTET STRING
DEFINED in MAP-CommonDataTypes : 63
USED in MAP-CommonDataTypes : 191 200

TCAPMessages.....module reference
DEFINED in TCAPMessages : 1
USED in MAP-MobileServiceOpera : 51
USED in MAP-OperationAndMainte : 20
USED in MAP-CallHandlingOperat : 20
USED in MAP-SupplementaryServi : 27
USED in MAP-ShortMessageServic : 24
USED in MAP-Errors : 75

telephony.....value reference TeleserviceCode, '00010001'B
DEFINED in MAP-TS-Code : 41

teleservice.....identifier of [3] TeleserviceCode
DEFINED in MAP-CommonDataTypes : 268

TeleserviceCode.....type reference OCTET STRING
DEFINED in MAP-TS-Code : 11

```


	USED in MAP-CommonDataTypes	:	42	268						
49	USED in MAP-TS-Code	:	38	40	41	42	44	45	46	48
			50	51	55	58	67	69	70	72
73			74	75	76	77	78	79	80	81
82			83	84	85	86	87			
	teleserviceList.....	identifier of [6] TeleserviceList								
	DEFINED in MAP-MS-DataTypes	:	217							
	TeleserviceList.....	type reference SEQUENCE OF								
	DEFINED in MAP-MS-DataTypes	:	241							
	USED in MAP-MS-DataTypes	:	217	473						
	teleserviceList.....	identifier of [1] TeleserviceList								
	DEFINED in MAP-MS-DataTypes	:	473							
	teleserviceNotProvisioned.....	value reference TeleserviceNotProvisioned, CHOICE								
VALUE	DEFINED in MAP-Protocol	:	239							
	TeleserviceNotProvisioned.....	type reference ERROR								
	DEFINED in MAP-Errors	:	193							
	USED in MAP-Protocol	:	92	239						
	USED in MAP-CallHandlingOperat	:	31	69						
	USED in MAP-SupplementaryServi	:	34	85	102	119	139	157		
	USED in MAP-ShortMessageServic	:	35	77						
	USED in MAP-Errors	:	31							
	teleservNotProvParam.....	identifier of TeleservNotProvParam								
	DEFINED in MAP-Errors	:	195							
	TeleservNotProvParam.....	type reference SEQUENCE								
	DEFINED in MAP-ER-DataTypes	:	194							
	USED in MAP-Errors	:	97	195						
	USED in MAP-ER-DataTypes	:	31							
	temporaryDefaultAllowed.....	identifier of Named Number, 2								
	DEFINED in MAP-SS-DataTypes	:	149							
	temporaryDefaultRestricted.....	identifier of Named Number, 1								
	DEFINED in MAP-SS-DataTypes	:	148							
	termAttemptAuthorized.....	identifier of Named Number, 12								
	DEFINED in MAP-CH-DataTypes	:	201							
	tmsi.....	identifier of TMSI								
	DEFINED in MAP-MobileServiceOpera	:	134							
	TMSI.....	type reference OCTET STRING								
	DEFINED in MAP-CommonDataTypes	:	194							
	USED in MAP-MobileServiceOpera	:	97	134						
	USED in MAP-CommonDataTypes	:	25	198						
	tmsi.....	identifier of [1] TMSI								
	DEFINED in MAP-CommonDataTypes	:	198							
	tooManyZoneCodes.....	identifier of Named Number, 1								
	DEFINED in MAP-MS-DataTypes	:	485							
	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	:	253							
	TracingBufferFull.....	type reference ERROR								
	DEFINED in MAP-Errors	:	209							
	USED in MAP-Protocol	:	95	253						

[illegible]

```

UnidentifiedSubscriber.....type reference ERROR
    DEFINED in MAP-Errors           :    160
    USED in MAP-Protocol            :    86    228
    USED in MAP-MobileServiceOpera :    59    139    240    251
    USED in MAP-OperationAndMainte :    28    61    75
    USED in MAP-ShortMessageServic :    32    104
    USED in MAP-Errors              :    23

unidirectional.....identifier of [APPLICATION 1] IMPLICIT
Unidirectional
    DEFINED in TCAPMessages         :    52

Unidirectional.....type reference SEQUENCE
    DEFINED in TCAPMessages         :    58
    USED in TCAPMessages            :    52

unknownAlphabet.....value reference UnknownAlphabet, CHOICE VALUE
    DEFINED in MAP-Protocol         :    280

UnknownAlphabet.....type reference ERROR
    DEFINED in MAP-Errors           :    289
    USED in MAP-Protocol            :    111    280
    USED in MAP-SupplementaryServi :    44    171    187    201
    USED in MAP-Errors              :    60

unknownEquipment.....value reference UnknownEquipment, CHOICE VALUE
    DEFINED in MAP-Protocol         :    229

UnknownEquipment.....type reference ERROR
    DEFINED in MAP-Errors           :    166
    USED in MAP-Protocol            :    87    229
    USED in MAP-MobileServiceOpera :    60    227
    USED in MAP-Errors              :    24

unknownMSC.....value reference UnknownMSC, CHOICE VALUE
    DEFINED in MAP-Protocol         :    227

UnknownMSC.....type reference ERROR
    DEFINED in MAP-Errors           :    158
    USED in MAP-Protocol            :    85    227
    USED in MAP-MobileServiceOpera :    58    200
    USED in MAP-Errors              :    22

unknownServiceCentre.....identifier of Named Number, 3
    DEFINED in MAP-ER-DataTypes     :    123

unknownSubscriber.....value reference UnknownSubscriber, CHOICE VALUE
    DEFINED in MAP-Protocol         :    225

UnknownSubscriber.....type reference ERROR
    DEFINED in MAP-Errors           :    147
    USED in MAP-Protocol            :    83    225
    USED in MAP-MobileServiceOpera :    57    116    164    215    270
    USED in MAP-OperationAndMainte :    27    85
    USED in MAP-CallHandlingOperat :    28    66
    USED in MAP-ShortMessageServic :    31    76    120    144
    USED in MAP-Errors              :    20

unknownSubscriberParam.....identifier of UnknownSubscriberParam
    DEFINED in MAP-Errors           :    149

UnknownSubscriberParam.....type reference SEQUENCE
    DEFINED in MAP-ER-DataTypes     :    170
    USED in MAP-Errors              :    90    149
    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

VALUE
unstructuredSS-Notify.....value reference UnstructuredSS-Notify, CHOICE
  DEFINED in MAP-Protocol      :    192

UnstructuredSS-Notify.....type reference OPERATION
  DEFINED in MAP-SupplementaryServi :    190
  USED in MAP-Protocol          :    59    192
  USED in MAP-SupplementaryServi :    20

unstructuredSS-Request.....value reference UnstructuredSS-Request, CHOICE
VALUE
  DEFINED in MAP-Protocol      :    191

UnstructuredSS-Request.....type reference OPERATION
  DEFINED in MAP-SupplementaryServi :    174
  USED in MAP-Protocol          :    58    191
  USED in MAP-SupplementaryServi :    19

updateLocation.....value reference UpdateLocation, CHOICE VALUE
  DEFINED in MAP-Protocol      :    128

UpdateLocation.....type reference OPERATION
  DEFINED in MAP-MobileServiceOpera :    107
  USED in MAP-Protocol          :    12    128
  USED in MAP-MobileServiceOpera :    15

updateLocationArg.....identifier of UpdateLocationArg
  DEFINED in MAP-MobileServiceOpera :    109

UpdateLocationArg.....type reference SEQUENCE
  DEFINED in MAP-MS-DataTypes      :    120
  USED in MAP-MobileServiceOpera   :    69    109
  USED in MAP-MS-DataTypes         :    16

updateLocationRes.....identifier of UpdateLocationRes
  DEFINED in MAP-MobileServiceOpera :    111

UpdateLocationRes.....type reference SEQUENCE
  DEFINED in MAP-MS-DataTypes      :    128
  USED in MAP-MobileServiceOpera   :    70    111
  USED in MAP-MS-DataTypes         :    17

ussd-Arg.....identifier of USSD-Arg
  DEFINED in MAP-SupplementaryServi :    164

ussd-Arg.....identifier of USSD-Arg
  DEFINED in MAP-SupplementaryServi :    176

ussd-Arg.....identifier of USSD-Arg
  DEFINED in MAP-SupplementaryServi :    192

USSD-Arg.....type reference SEQUENCE
  DEFINED in MAP-SS-DataTypes      :    171
  USED in MAP-SupplementaryServi   :    57    164    176    192
  USED in MAP-SS-DataTypes         :    20

ussd-Busy.....value reference USSD-Busy, CHOICE VALUE
  DEFINED in MAP-Protocol          :    281

USSD-Busy.....type reference ERROR
  DEFINED in MAP-Errors            :    291
  USED in MAP-Protocol            :    112    281
  USED in MAP-SupplementaryServi   :    45    188    202
  USED in MAP-Errors              :    61

ussd-DataCodingScheme.....identifier of USSD-DataCodingScheme
  DEFINED in MAP-SS-DataTypes      :    172

ussd-DataCodingScheme.....identifier of USSD-DataCodingScheme
  DEFINED in MAP-SS-DataTypes      :    177

USSD-DataCodingScheme.....type reference OCTET STRING
  DEFINED in MAP-SS-DataTypes      :    181
  USED in MAP-SS-DataTypes         :    32    172    177

ussd-Res.....identifier of USSD-Res
  DEFINED in MAP-SupplementaryServi :    166

ussd-Res.....identifier of USSD-Res

```

```

DEFINED in MAP-SupplementaryServi : 178

USSD-Res.....type reference SEQUENCE
  DEFINED in MAP-SS-DataTypes : 176
  USED in MAP-SupplementaryServi : 58 166 178
  USED in MAP-SS-DataTypes : 21

ussd-String.....identifier of USSD-String
  DEFINED in MAP-SS-DataTypes : 173

ussd-String.....identifier of USSD-String
  DEFINED in MAP-SS-DataTypes : 178

USSD-String.....type reference OCTET STRING
  DEFINED in MAP-SS-DataTypes : 186
  USED in MAP-SS-DataTypes : 33 173 178

uus.....value reference SS-Code, '10000001'B
  DEFINED in MAP-SS-Code : 97

VBSDDataList.....type reference SEQUENCE OF
  DEFINED in MAP-MS-DataTypes : 580
  USED in MAP-MS-DataTypes : 224

vbsGroupIndication.....identifier of [7] NULL
  DEFINED in MAP-MS-DataTypes : 497

vbsSubscriptionData.....identifier of [11] VBSDDataList
  DEFINED in MAP-MS-DataTypes : 224

VGCSDataList.....type reference SEQUENCE OF
  DEFINED in MAP-MS-DataTypes : 583
  USED in MAP-MS-DataTypes : 225

vgcsGroupIndication.....identifier of [8] NULL
  DEFINED in MAP-MS-DataTypes : 498

vgcsSubscriptionData.....identifier of [12] VGCSDataList
  DEFINED in MAP-MS-DataTypes : 225

vlr.....identifier of Named Number, 2
  DEFINED in MAP-CommonDataTypes : 231

vlrCamelSubscriptionInfo.....identifier of [13] VlrCamelSubscriptionInfo
  DEFINED in MAP-MS-DataTypes : 226

VlrCamelSubscriptionInfo.....type reference SEQUENCE
  DEFINED in MAP-MS-DataTypes : 514
  USED in MAP-MS-DataTypes : 226

vlr-Number.....identifier of ISDN-AddressString
  DEFINED in MAP-MS-DataTypes : 123

vlr-Number.....identifier of ISDN-AddressString
  DEFINED in MAP-MS-DataTypes : 139

vlr-number.....identifier of [1] ISDN-AddressString
  DEFINED in MAP-MS-DataTypes : 634

vmsc.....identifier of Named Number, 5
  DEFINED in MAP-CommonDataTypes : 234

vmsc-Address.....identifier of [2] ISDN-AddressString
  DEFINED in MAP-CH-DataTypes : 113

voiceBroadcastCall.....value reference TeleserviceCode, '10010010'B
  DEFINED in MAP-TS-Code : 70

VoiceBroadcastData.....type reference SEQUENCE
  DEFINED in MAP-MS-DataTypes : 595
  USED in MAP-MS-DataTypes : 581

voiceGroupCall.....value reference TeleserviceCode, '10010001'B
  DEFINED in MAP-TS-Code : 69

VoiceGroupCallData.....type reference SEQUENCE
  DEFINED in MAP-MS-DataTypes : 590
  USED in MAP-MS-DataTypes : 584

whiteListed.....identifier of Named Number, 0
  DEFINED in MAP-MS-DataTypes : 197

ZoneCode.....type reference OCTET STRING
  DEFINED in MAP-MS-DataTypes : 467
  USED in MAP-MS-DataTypes : 465 496

```

```
ZoneCodeList.....type reference SEQUENCE OF
  DEFINED in MAP-MS-DataTypes      :      464
  USED in MAP-MS-DataTypes         :      42   223

zoneCodesConflict.....identifier of Named Number, 2
  DEFINED in MAP-MS-DataTypes      :      486
```

Annex B (informative): Fully expanded ASN.1 sources for abstract syntaxes of MAP

Annex B is not part of the standard, it is included for information purposes only.

For every (Value)Assignment in the root ASN.1 module all the used defined types and defined values, which are defined within the ASN.1 module or imported from ASN.1 modules, are replaced by the constructs this type or value is composed of.

The fully expanded ASN.1 root module is itself a correct and equivalent representation of the MAP-Protocol.

It allows to see at all the parameters, including all nested ones for a specific operationcode or errorcode at once.

Note that for those operations which use a result without parameters the keyword RESULT is not shown. Empty results are only defined in the ASN.1 description in clause 14.

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

```
-- Expanded ASN1 Module 'MAP-Protocol'
--SIEMENS ASN.1 Compiler      P3.90M (04-07-00-00-64-00)
--      Date: 98-07-23 Time: 09:08:56

MAP-Protocol { 0 identified-organization (4) etsi (0) mobileDomain (0) gsm-Network (1) modules
(3) map-Protocol (4) version3 (3) }

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,
            ... }
      }
  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 CHOICE {
    imsi      OCTET STRING ( SIZE ( 3 .. 8 ) ),
    imsi-WithLMSI SEQUENCE {
      imsi      OCTET STRING ( SIZE ( 3 .. 8 ) ),
      lmsi      OCTET STRING ( SIZE ( 4 ) ),
      ... } }
ERRORS {
  -- dataMissing -- localValue : 35,
  -- unexpectedDataValue -- localValue : 36}
::= localValue : 3

```

purgeMS OPERATION

```

ARGUMENT
  purgeMS-Arg SEQUENCE {
    imsi      OCTET STRING ( SIZE ( 3 .. 8 ) ),
    vlr-Number OCTET STRING ( SIZE ( 1 .. 20 ) ) ( SIZE ( 1 .. 9 ) ),
    ... }
::= localValue : 67

```

sendIdentification OPERATION

```

ARGUMENT
  tmsi      OCTET STRING ( SIZE ( 1 .. 4 ) )
RESULT
  sendIdentificationRes SEQUENCE {
    imsi      OCTET STRING ( SIZE ( 3 .. 8 ) ),
    authenticationSetList SEQUENCE SIZE ( 1 .. 5 ) OF
      SEQUENCE {
        rand      OCTET STRING ( SIZE ( 16 ) ),
        sres      OCTET STRING ( SIZE ( 4 ) ),
        kc      OCTET STRING ( SIZE ( 8 ) ),
        ... } OPTIONAL,
    ... }
ERRORS {
  -- dataMissing -- localValue : 35,
  -- unidentifiedSubscriber -- localValue : 5}
::= localValue : 55

```

prepareHandover OPERATION

```

ARGUMENT
  prepareHO-Arg SEQUENCE {
    targetCellId OCTET STRING ( SIZE ( 5 .. 7 ) ) OPTIONAL,
    ho-NumberNotRequired NULL OPTIONAL,
    bss-APDU SEQUENCE {
      protocolId ENUMERATED {
        gsm-0408 ( 1 ),
        gsm-0806 ( 2 ),
        gsm-BSSMAP ( 3 ),
        ets-300102-1 ( 4 ) },
      signalInfo OCTET STRING ( SIZE ( 1 .. 200 ) ),
      extensionContainer SEQUENCE {
        privateExtensionList [ 0 ] IMPLICIT SEQUENCE SIZE ( 1 .. 10 ) OF
          SEQUENCE {
            extId      MAP-EXTENSION .&extensionId ( {
              '... } ) ,
            extType     MAP-EXTENSION .&ExtensionType ( {
              '... } { @extId } ) OPTIONAL} OPTIONAL,
            pcs-Extensions [ 1 ] IMPLICIT SEQUENCE {
              ... } OPTIONAL,
              ... } OPTIONAL,
              ... } OPTIONAL,
              ... }
RESULT
  prepareHO-Res SEQUENCE {
    handoverNumber OCTET STRING ( SIZE ( 1 .. 20 ) ) ( SIZE ( 1 .. 9 ) ) OPTIONAL,
    bss-APDU SEQUENCE {
      protocolId ENUMERATED {
        gsm-0408 ( 1 ),
        gsm-0806 ( 2 ),
        gsm-BSSMAP ( 3 ),
        ets-300102-1 ( 4 ) },
      signalInfo OCTET STRING ( SIZE ( 1 .. 200 ) ),
      extensionContainer SEQUENCE {
        privateExtensionList [ 0 ] IMPLICIT SEQUENCE SIZE ( 1 .. 10 ) OF
          SEQUENCE {
            extId      MAP-EXTENSION .&extensionId ( {
              '... } ) ,
            extType     MAP-EXTENSION .&ExtensionType ( {
              '... } { @extId } ) OPTIONAL} OPTIONAL,
            pcs-Extensions [ 1 ] IMPLICIT SEQUENCE {

```



```

        ... } OPTIONAL,
        ... } OPTIONAL,
        ... } OPTIONAL,
    ... }
ERRORS {
    -- systemFailure -- localValue : 34,
    -- dataMissing -- localValue : 35,
    -- unexpectedDataValue -- localValue : 36,
    -- noHandoverNumberAvailable -- localValue : 25}
 ::= localValue : 68

sendEndSignal OPERATION
ARGUMENT
    bss-APDU SEQUENCE {
        protocolId ENUMERATED {
            gsm-0408 (1 ),
            gsm-0806 (2 ),
            gsm-BSSMAP (3 ),
            ets-300102-1 (4 )},
        signalInfo OCTET STRING ( SIZE (1 .. 200 ) ),
        extensionContainer SEQUENCE {
            privateExtensionList [0] IMPLICIT SEQUENCE SIZE (1 .. 10 ) OF
                SEQUENCE {
                    extId MAP-EXTENSION .&extensionId ( {
                        '...} ) ,
                    extType MAP-EXTENSION .&ExtensionType ( {
                        '...} { @extId } ) OPTIONAL} OPTIONAL,
            pcs-Extensions [1] IMPLICIT SEQUENCE {
                ... } OPTIONAL,
                ... } OPTIONAL,
            ... }
        ::= localValue : 29

processAccessSignalling OPERATION
ARGUMENT
    bss-APDU SEQUENCE {
        protocolId ENUMERATED {
            gsm-0408 (1 ),
            gsm-0806 (2 ),
            gsm-BSSMAP (3 ),
            ets-300102-1 (4 )},
        signalInfo OCTET STRING ( SIZE (1 .. 200 ) ),
        extensionContainer SEQUENCE {
            privateExtensionList [0] IMPLICIT SEQUENCE SIZE (1 .. 10 ) OF
                SEQUENCE {
                    extId MAP-EXTENSION .&extensionId ( {
                        '...} ) ,
                    extType MAP-EXTENSION .&ExtensionType ( {
                        '...} { @extId } ) OPTIONAL} OPTIONAL,
            pcs-Extensions [1] IMPLICIT SEQUENCE {
                ... } OPTIONAL,
                ... } OPTIONAL,
            ... }
        ::= localValue : 33

forwardAccessSignalling OPERATION
ARGUMENT
    bss-APDU SEQUENCE {
        protocolId ENUMERATED {
            gsm-0408 (1 ),
            gsm-0806 (2 ),
            gsm-BSSMAP (3 ),
            ets-300102-1 (4 )},
        signalInfo OCTET STRING ( SIZE (1 .. 200 ) ),
        extensionContainer SEQUENCE {
            privateExtensionList [0] IMPLICIT SEQUENCE SIZE (1 .. 10 ) OF
                SEQUENCE {
                    extId MAP-EXTENSION .&extensionId ( {
                        '...} ) ,
                    extType MAP-EXTENSION .&ExtensionType ( {
                        '...} { @extId } ) OPTIONAL} OPTIONAL,
            pcs-Extensions [1] IMPLICIT SEQUENCE {
                ... } OPTIONAL,
                ... } OPTIONAL,
            ... }
        ::= localValue : 34

prepareSubsequentHandover OPERATION
ARGUMENT

```

```

prepareSubsequentHO-Arg SEQUENCE {
    targetCellId OCTET STRING ( SIZE ( 5 .. 7 ) ),
    targetMSC-Number OCTET STRING ( SIZE ( 1 .. 20 ) ) ( SIZE ( 1 .. 9 ) ),
    bss-APDU SEQUENCE {
        protocolId ENUMERATED {
            gsm-0408 ( 1 ),
            gsm-0806 ( 2 ),
            gsm-BSSMAP ( 3 ),
            ets-300102-1 ( 4 )},
        signalInfo OCTET STRING ( SIZE ( 1 .. 200 ) ),
        extensionContainer SEQUENCE {
            privateExtensionList [0] IMPLICIT SEQUENCE SIZE ( 1 .. 10 ) OF
                SEQUENCE {
                    extId MAP-EXTENSION .&extensionId ( {
                        '...' } ),
                    extType MAP-EXTENSION .&ExtensionType ( {
                        '...' { @extId } ) OPTIONAL} OPTIONAL,
            pcs-Extensions [1] IMPLICIT SEQUENCE {
                ... } OPTIONAL,
                ... } OPTIONAL,
            ... },
        ... }
RESULT
    bss-APDU SEQUENCE {
        protocolId ENUMERATED {
            gsm-0408 ( 1 ),
            gsm-0806 ( 2 ),
            gsm-BSSMAP ( 3 ),
            ets-300102-1 ( 4 )},
        signalInfo OCTET STRING ( SIZE ( 1 .. 200 ) ),
        extensionContainer SEQUENCE {
            privateExtensionList [0] IMPLICIT SEQUENCE SIZE ( 1 .. 10 ) OF
                SEQUENCE {
                    extId MAP-EXTENSION .&extensionId ( {
                        '...' } ),
                    extType MAP-EXTENSION .&ExtensionType ( {
                        '...' { @extId } ) OPTIONAL} OPTIONAL,
            pcs-Extensions [1] IMPLICIT SEQUENCE {
                ... } OPTIONAL,
                ... } OPTIONAL,
            ... }
ERRORS {
    -- unexpectedDataValue -- localValue : 36,
    -- dataMissing -- localValue : 35,
    -- unknownMSC -- localValue : 3,
    -- subsequentHandoverFailure -- localValue : 26}
::= localValue : 69

sendAuthenticationInfo OPERATION
ARGUMENT
    sendAuthenticationInfoArg OCTET STRING ( SIZE ( 3 .. 8 ) )
RESULT
    sendAuthenticationInfoRes SEQUENCE SIZE ( 1 .. 5 ) OF
        SEQUENCE {
            rand OCTET STRING ( SIZE ( 16 ) ),
            sres OCTET STRING ( SIZE ( 4 ) ),
            kc OCTET STRING ( SIZE ( 8 ) ),
            ... }
ERRORS {
    -- systemFailure -- localValue : 34,
    -- dataMissing -- localValue : 35,
    -- unexpectedDataValue -- localValue : 36,
    -- unknownSubscriber -- localValue : 1}
::= localValue : 56

checkIMEI OPERATION
ARGUMENT
    imei OCTET STRING ( SIZE ( 8 ) )
RESULT
    equipmentStatus ENUMERATED {
        whiteListed ( 0 ),
        blackListed ( 1 ),
        greyListed ( 2 )}
ERRORS {
    -- systemFailure -- localValue : 34,
    -- dataMissing -- localValue : 35,
    -- unknownEquipment -- localValue : 7}
::= localValue : 43

insertSubscriberData OPERATION
ARGUMENT

```

[illegible]

```

        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,
            ... } 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,
                    ... } } OPTIONAL,
        }
    odb-Data [8] IMPLICIT SEQUENCE {
        odb-GeneralData BIT STRING {
            allOG-CallsBarred (0 ),
            internationalOGCallsBarred (1 ),
            internationalOGCallsNotToHPLMN-CountryBarred (2 ),
            interzonalOGCallsBarred (6 ),
            interzonalOGCallsNotToHPLMN-CountryBarred (7 ),
            interzonalOGCallsAndInternationalOGCallsNotToHPLMN-CountryBarred (8 ),
            premiumRateInformationOGCallsBarred (3 ),
            premiumRateEntertainmentOGCallsBarred (4 ),
            ss-AccessBarred (5 ),
            allECT-Barred (9 ),
            chargeableECT-Barred (10 ),
            internationalECT-Barred (11 ),
            interzonalECT-Barred (12 ),
            doublyChargeableECT-Barred (13 ),
            multipleECT-Barred (14 )} ( SIZE (15 .. 32 ) ),
        odb-HPLMN-Data BIT STRING {
            plmn-SpecificBarringType1 (0 ),
            plmn-SpecificBarringType2 (1 ),
            plmn-SpecificBarringType3 (2 ),
            plmn-SpecificBarringType4 (3 )} ( SIZE (4 .. 32 ) ) OPTIONAL,
        extensionContainer SEQUENCE {
            privateExtensionList [0] IMPLICIT SEQUENCE SIZE (1 .. 10 ) OF
                SEQUENCE {
                    extId      MAP-EXTENSION .&extensionId ( {
                        '... } ) ,
                    extType      MAP-EXTENSION .&ExtensionType ( {
                        '... } { @extId } ) OPTIONAL } OPTIONAL,
                    pcs-Extensions [1] IMPLICIT SEQUENCE {
                        ... } OPTIONAL,
                        ... } OPTIONAL,
                        ... } OPTIONAL,
                    ... } OPTIONAL,
                }
            roamingRestrictionDueToUnsupportedFeature [9] IMPLICIT NULL OPTIONAL,
            regionalSubscriptionData [10] IMPLICIT SEQUENCE SIZE (1 .. 10 ) OF
                OCTET STRING ( SIZE (2 ) ) OPTIONAL,
            vbsSubscriptionData [11] IMPLICIT SEQUENCE SIZE (1 .. 50 ) OF
                SEQUENCE {
                    groupId      OCTET STRING ( SIZE (3 ) ),
                    broadcastInitEntitlement NULL OPTIONAL,
                    extensionContainer SEQUENCE {
                        privateExtensionList [0] IMPLICIT SEQUENCE SIZE (1 .. 10 ) OF
                            SEQUENCE {
                                extId      MAP-EXTENSION .&extensionId ( {
                                    '... } ) ,
                                extType      MAP-EXTENSION .&ExtensionType ( {
                                    '... } { @extId } ) OPTIONAL } OPTIONAL,
                                pcs-Extensions [1] IMPLICIT SEQUENCE {
                                    ... } OPTIONAL,
                                    ... } OPTIONAL,
                                    ... } OPTIONAL,
                                ... } OPTIONAL,
                            }
                        vgcSubscriptionData [12] IMPLICIT SEQUENCE SIZE (1 .. 50 ) OF
                            SEQUENCE {
                                groupId      OCTET STRING ( SIZE (3 ) ),
                                extensionContainer SEQUENCE {
                                    privateExtensionList [0] IMPLICIT SEQUENCE SIZE (1 .. 10 ) OF

```

```

SEQUENCE {
    extId      MAP-EXTENSION .&extensionId ( {
        '...' ) ,
    extType    MAP-EXTENSION .&ExtensionType ( {
        '...' { @extId } ) OPTIONAL} OPTIONAL,
    pcs-Extensions [1] IMPLICIT SEQUENCE {
        ... } OPTIONAL,
    ... } OPTIONAL,
    ... } OPTIONAL,
vlrCamelSubscriptionInfo [13] IMPLICIT SEQUENCE {
    o-CSI      [0] IMPLICIT SEQUENCE {
        o-BcsmCamelTDPDataList SEQUENCE SIZE (1 .. 10) OF
            SEQUENCE {
                o-BcsmTriggerDetectionPoint ENUMERATED {
                    collectedInfo (2),
                    ... },
                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,
                                                ... } 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,
                                                            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,
                                                                                    ... } OPTIONAL,
                                                                ... }
                                                                    RESULT
insertSubscriberDataRes SEQUENCE {
    teleserviceList [1] IMPLICIT SEQUENCE SIZE (1 .. 20) OF
        OCTET STRING (SIZE (1 .. 5)) OPTIONAL,
    bearerServiceList [2] IMPLICIT SEQUENCE SIZE (1 .. 50) OF
        OCTET STRING (SIZE (1 .. 5)) OPTIONAL,
    ss-List [3] IMPLICIT SEQUENCE SIZE (1 .. 30) OF
        OCTET STRING (SIZE (1)) OPTIONAL,

```

```

odb-GeneralData [4] IMPLICIT BIT STRING {
    allOG-CallsBarred (0 ),
    internationalOGCallsBarred (1 ),
    internationalOGCallsNotToHPLMN-CountryBarred (2 ),
    interzonalOGCallsBarred (6 ),
    interzonalOGCallsNotToHPLMN-CountryBarred (7 ),
    interzonalOGCallsAndInternationalOGCallsNotToHPLMN-CountryBarred (8 ),
    premiumRateInformationOGCallsBarred (3 ),
    premiumRateEntertainmentOGCallsBarred (4 ),
    ss-AccessBarred (5 ),
    allECT-Barred (9 ),
    chargeableECT-Barred (10 ),
    internationalECT-Barred (11 ),
    interzonalECT-Barred (12 ),
    doublyChargeableECT-Barred (13 ),
    multipleECT-Barred (14 )} ( SIZE (15 .. 32 ) ) OPTIONAL,
regionalSubscriptionResponse [5] IMPLICIT ENUMERATED {
    msc-AreaRestricted (0 ),
    tooManyZoneCodes (1 ),
    zoneCodesConflict (2 ),
    regionalSubscNotSupported (3 )} OPTIONAL,
supportedCamelPhases [6] IMPLICIT BIT STRING {
    phase1 (0 )} ( 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,
            ... }
RESULT
    deleteSubscriberDataRes SEQUENCE {
        regionalSubscriptionResponse [0] IMPLICIT ENUMERATED {
            msc-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,
                    ... }
        ... }
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,
                camellInfo [11] IMPLICIT SEQUENCE {
                    supportedCamelPhases BIT STRING {
                        phase1 ( 0 )} ( 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,
                                                        ... } 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 {

```

```

) ) OPTIONAL,
forwardedToNumber [5] IMPLICIT OCTET STRING ( SIZE ( 1 .. 20 ) ) ( SIZE ( 1 .. 9
forwardedToSubaddress [4] IMPLICIT OCTET STRING ( SIZE ( 1 .. 21 ) ) OPTIONAL,
forwardingOptions [6] IMPLICIT OCTET STRING ( SIZE ( 1 ) ) OPTIONAL,
extensionContainer [7] IMPLICIT SEQUENCE {
  privateExtensionList [0] IMPLICIT SEQUENCE SIZE ( 1 .. 10 ) OF
    SEQUENCE {
      extId      MAP-EXTENSION .&extensionId ( {
        '...} ) ,
      extType    MAP-EXTENSION .&ExtensionType ( {
        '...} { @extId } ) OPTIONAL} OPTIONAL,
  pcs-Extensions [1] IMPLICIT SEQUENCE {
    ... } OPTIONAL,
    ... } OPTIONAL,
    ... }},
camelRoutingInfo [8] IMPLICIT SEQUENCE {
  forwardingData SEQUENCE {
    forwardedToNumber [5] IMPLICIT OCTET STRING ( SIZE ( 1 .. 20 ) ) ( SIZE ( 1 .. 9
) ) OPTIONAL,
    forwardedToSubaddress [4] IMPLICIT OCTET STRING ( SIZE ( 1 .. 21 ) ) OPTIONAL,
    forwardingOptions [6] IMPLICIT OCTET STRING ( SIZE ( 1 ) ) OPTIONAL,
    extensionContainer [7] IMPLICIT SEQUENCE {
      privateExtensionList [0] IMPLICIT SEQUENCE SIZE ( 1 .. 10 ) OF
        SEQUENCE {
          extId      MAP-EXTENSION .&extensionId ( {
            '...} ) ,
          extType    MAP-EXTENSION .&ExtensionType ( {
            '...} { @extId } ) OPTIONAL} OPTIONAL,
          pcs-Extensions [1] IMPLICIT SEQUENCE {
            ... } OPTIONAL,
            ... } OPTIONAL,
            ... } OPTIONAL,
            gsmcCamelSubscriptionInfo [0] IMPLICIT SEQUENCE {
              t-CSI [0] IMPLICIT SEQUENCE {
                t-BcsmCamelTDPDataList SEQUENCE SIZE ( 1 .. 10 ) OF
                  SEQUENCE {
                    t-BcsmTriggerDetectionPoint ENUMERATED {
                      termAttemptAuthorized ( 12 ),
                      ... },
                    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,
                                  ... } OPTIONAL,
                                  o-CSI [1] IMPLICIT SEQUENCE {
                                    o-BcsmCamelTDPDataList SEQUENCE SIZE ( 1 .. 10 ) OF
                                      SEQUENCE {
                                        o-BcsmTriggerDetectionPoint ENUMERATED {
                                          collectedInfo ( 2 ),
                                          ... },
                                        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,
                        ... } 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 [1] IMPLICIT SEQUENCE {
                privateExtensionList [0] IMPLICIT SEQUENCE SIZE (1 .. 10 ) OF
                    SEQUENCE {
                        extId      MAP-EXTENSION .&extensionId ( {
                            '...'} ) ,
                        extType    MAP-EXTENSION .&ExtensionType ( {
                            '...' { @extId } ) OPTIONAL} OPTIONAL,
                        pcs-Extensions [1] IMPLICIT SEQUENCE {
                            ... } OPTIONAL,
                            ... } OPTIONAL,
                        ... } OPTIONAL,
                    ... } OPTIONAL,
            cug-CheckInfo [3] IMPLICIT SEQUENCE {
                cug-Interlock OCTET STRING ( SIZE (4 ) ),
                cug-OutgoingAccess NULL OPTIONAL,
                extensionContainer SEQUENCE {
                    privateExtensionList [0] IMPLICIT SEQUENCE SIZE (1 .. 10 ) OF
                        SEQUENCE {
                            extId      MAP-EXTENSION .&extensionId ( {
                                '...'} ) ,
                            extType    MAP-EXTENSION .&ExtensionType ( {
                                '...' { @extId } ) OPTIONAL} OPTIONAL,
                                pcs-Extensions [1] IMPLICIT SEQUENCE {
                                    ... } OPTIONAL,
                                    ... } OPTIONAL,
                                ... } OPTIONAL,
                            ... } OPTIONAL,
                cugSubscriptionFlag [6] IMPLICIT NULL OPTIONAL,
                subscriberInfo [7] IMPLICIT SEQUENCE {
                    locationInformation [0] IMPLICIT SEQUENCE {
                        ageOfLocationInformation INTEGER ( 0 .. 32767 ) OPTIONAL,
                        geographicalInformation [0] IMPLICIT OCTET STRING ( SIZE (8 ) ) OPTIONAL,
                        vlr-number [1] IMPLICIT OCTET STRING ( SIZE (1 .. 20 ) ) ( SIZE (1 .. 9 ) )
OPTIONAL,
                        locationNumber [2] IMPLICIT OCTET STRING ( SIZE (2 .. 10 ) ) OPTIONAL,
                        cellIdOrLAI [3] CHOICE {
                            cellIdFixedLength [0] IMPLICIT OCTET STRING ( SIZE (7 ) ),
                            laiFixedLength [1] IMPLICIT OCTET STRING ( SIZE (5 ) ) } OPTIONAL,
                        extensionContainer [4] IMPLICIT SEQUENCE {
                            privateExtensionList [0] IMPLICIT SEQUENCE SIZE (1 .. 10 ) OF

```

```

SEQUENCE {
    extId      MAP-EXTENSION .&extensionId ( {
        '...'} ) ,
    extType    MAP-EXTENSION .&ExtensionType ( {
        '...'} { @extId } ) OPTIONAL} OPTIONAL,
    pcs-Extensions [1] IMPLICIT SEQUENCE {
        ... } OPTIONAL,
    ... } OPTIONAL,
    ... } OPTIONAL,
    subscriberState [1] CHOICE {
        assumedIdle [0] IMPLICIT NULL,
        camelBusy [1] IMPLICIT NULL,
        netDetNotReachable ENUMERATED {
            msPurged (0 ),
            imsiDetached (1 ),
            restrictedArea (2 ),
            notRegistered (3 )},
        notProvidedFromVLR [2] IMPLICIT NULL} OPTIONAL,
    extensionContainer [2] IMPLICIT SEQUENCE {
        privateExtensionList [0] IMPLICIT SEQUENCE SIZE (1 .. 10 ) OF
            SEQUENCE {
                extId      MAP-EXTENSION .&extensionId ( {
                    '...'} ) ,
                extType    MAP-EXTENSION .&ExtensionType ( {
                    '...'} { @extId } ) OPTIONAL} OPTIONAL,
                pcs-Extensions [1] IMPLICIT SEQUENCE {
                    ... } OPTIONAL,
                    ... } OPTIONAL,
                    ... } OPTIONAL,
            ss-List [1] IMPLICIT SEQUENCE SIZE (1 .. 30 ) OF
                OCTET STRING ( SIZE (1 ) ) OPTIONAL,
            basicService [5] CHOICE {
                ext-BearerService [2] IMPLICIT OCTET STRING ( SIZE (1 .. 5 ) ),
                ext-Teleservice [3] IMPLICIT OCTET STRING ( SIZE (1 .. 5 ) )} OPTIONAL,
                forwardingInterrogationRequired [4] IMPLICIT NULL OPTIONAL,
                vmsc-Address [2] IMPLICIT OCTET STRING ( SIZE (1 .. 20 ) ) ( SIZE (1 .. 9 ) ) OPTIONAL,
                extensionContainer [0] IMPLICIT SEQUENCE {
                    privateExtensionList [0] IMPLICIT SEQUENCE SIZE (1 .. 10 ) OF
                        SEQUENCE {
                            extId      MAP-EXTENSION .&extensionId ( {
                                '...'} ) ,
                            extType    MAP-EXTENSION .&ExtensionType ( {
                                '...'} { @extId } ) OPTIONAL} OPTIONAL,
                            pcs-Extensions [1] IMPLICIT SEQUENCE {
                                ... } OPTIONAL,
                                ... } OPTIONAL,
                                ... } OPTIONAL,
                        ... }
                    }
            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,
    gsmc-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,
                ... }
    ... }
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 ) ),
            basicServiceGroup [1] CHOICE {
                ext-BearerService [2] IMPLICIT OCTET STRING ( SIZE (1 .. 5 ) ),
                ext-Teleservice [3] IMPLICIT OCTET STRING ( SIZE (1 .. 5 ) )},
            forwardingData [2] IMPLICIT SEQUENCE {

```

```

OPTIONAL,
forwardedToNumber [5] IMPLICIT OCTET STRING ( SIZE (1 .. 20 ) ) ( SIZE (1 .. 9 ) )
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,
        ... },
imsi [3] IMPLICIT OCTET STRING ( SIZE (3 .. 8 ) ),
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 ),
                        ... },
                    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,
                                        ... } OPTIONAL,
                                    extensionContainer [7] IMPLICIT SEQUENCE {
                                        privateExtensionList [0] IMPLICIT SEQUENCE SIZE (1 .. 10 ) OF
                                            SEQUENCE {
                                                extId      MAP-EXTENSION .&extensionId ( {
                                                    '...'} ) ,
                                                extType    MAP-EXTENSION .&ExtensionType ( {
                                                    '...' { @extId } ) OPTIONAL} OPTIONAL,
                                                pcs-Extensions [1] IMPLICIT SEQUENCE {
                                                    ... } OPTIONAL,
                                                    ... } OPTIONAL,
                                                ... } OPTIONAL,
                                            }

```

```

... }
RESULT
  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}
::= localValue : 6

registerSS OPERATION
  ARGUMENT
    registerSS-Arg SEQUENCE {
      ss-Code      OCTET STRING ( SIZE (1 ) ),
      basicService CHOICE {
        bearerService [2] IMPLICIT OCTET STRING ( SIZE (1 ) ),
        teleservice [3] IMPLICIT OCTET STRING ( SIZE (1 ) )} OPTIONAL,
        forwardedToNumber [4] IMPLICIT OCTET STRING ( SIZE (1 .. 20 ) ) OPTIONAL,
        forwardedToSubaddress [6] IMPLICIT OCTET STRING ( SIZE (1 .. 21 ) ) OPTIONAL,
        noReplyConditionTime [5] IMPLICIT INTEGER ( 5 .. 30 ) OPTIONAL,
        '... }
  RESULT
    ss-Info CHOICE {
      forwardingInfo [0] IMPLICIT SEQUENCE {
        ss-Code      OCTET STRING ( SIZE (1 ) ) OPTIONAL,
        forwardingFeatureList SEQUENCE SIZE (1 .. 13 ) OF
          SEQUENCE {
            basicService CHOICE {
              bearerService [2] IMPLICIT OCTET STRING ( SIZE (1 ) ),
              teleservice [3] IMPLICIT OCTET STRING ( SIZE (1 ) )} OPTIONAL,
              ss-Status [4] IMPLICIT OCTET STRING ( SIZE (1 ) ) OPTIONAL,
              forwardedToNumber [5] IMPLICIT OCTET STRING ( SIZE (1 .. 20 ) ) ( SIZE (1 .. 9
            ) ) OPTIONAL,
              forwardedToSubaddress [8] IMPLICIT OCTET STRING ( SIZE (1 .. 21 ) ) OPTIONAL,
              forwardingOptions [6] IMPLICIT OCTET STRING ( SIZE (1 ) ) OPTIONAL,
              noReplyConditionTime [7] IMPLICIT INTEGER ( 5 .. 30 ) OPTIONAL,
              '... },
            '... },
            callBarringInfo [1] IMPLICIT SEQUENCE {
              ss-Code      OCTET STRING ( SIZE (1 ) ) OPTIONAL,
              callBarringFeatureList SEQUENCE SIZE (1 .. 13 ) OF
                SEQUENCE {
                  basicService CHOICE {
                    bearerService [2] IMPLICIT OCTET STRING ( SIZE (1 ) ),
                    teleservice [3] IMPLICIT OCTET STRING ( SIZE (1 ) )} OPTIONAL,
                    ss-Status [4] IMPLICIT OCTET STRING ( SIZE (1 ) ) OPTIONAL,
                    '... },
                    '... },
                    ss-Data [3] IMPLICIT SEQUENCE {
                      ss-Code      OCTET STRING ( SIZE (1 ) ) OPTIONAL,
                      ss-Status [4] IMPLICIT OCTET STRING ( SIZE (1 ) ) OPTIONAL,
                      ss-SubscriptionOption CHOICE {
                        cliRestrictionOption [2] IMPLICIT ENUMERATED {
                          permanent (0 ),
                          temporaryDefaultRestricted (1 ),
                          temporaryDefaultAllowed (2 )},
                        overrideCategory [1] IMPLICIT ENUMERATED {
                          overrideEnabled (0 ),
                          overrideDisabled (1 )} OPTIONAL,
                          basicServiceGroupList SEQUENCE SIZE (1 .. 13 ) OF
                            CHOICE {
                              bearerService [2] IMPLICIT OCTET STRING ( SIZE (1 ) ),
                              teleservice [3] IMPLICIT OCTET STRING ( SIZE (1 ) )} OPTIONAL,
                              '... }}
ERRORS {
  -- systemFailure -- localValue : 34,
  -- dataMissing -- localValue : 35,
  -- unexpectedDataValue -- localValue : 36,
  -- bearerServiceNotProvisioned -- localValue : 10,
  -- teleserviceNotProvisioned -- localValue : 11,
  -- callBarred -- localValue : 13,
  -- illegalSS-Operation -- localValue : 16,
  -- ss-ErrorStatus -- localValue : 17,

```



```

-- ss-Incompatibility -- localValue : 20}
::= localValue : 10

eraseSS OPERATION
  ARGUMENT
    ss-ForBS SEQUENCE {
      ss-Code OCTET STRING ( SIZE ( 1 ) ),
      basicService CHOICE {
        bearerService [2] IMPLICIT OCTET STRING ( SIZE ( 1 ) ),
        teleservice [3] IMPLICIT OCTET STRING ( SIZE ( 1 ) )} OPTIONAL,
      ... }
  RESULT
    ss-Info CHOICE {
      forwardingInfo [0] IMPLICIT SEQUENCE {
        ss-Code OCTET STRING ( SIZE ( 1 ) ) OPTIONAL,
        forwardingFeatureList SEQUENCE SIZE ( 1 .. 13 ) OF
          SEQUENCE {
            basicService CHOICE {
              bearerService [2] IMPLICIT OCTET STRING ( SIZE ( 1 ) ),
              teleservice [3] IMPLICIT OCTET STRING ( SIZE ( 1 ) )} OPTIONAL,
            ss-Status [4] IMPLICIT OCTET STRING ( SIZE ( 1 ) ) OPTIONAL,
            forwardedToNumber [5] IMPLICIT OCTET STRING ( SIZE ( 1 .. 20 ) ) ( SIZE ( 1 .. 9
          ) ) OPTIONAL,
            forwardedToSubaddress [8] IMPLICIT OCTET STRING ( SIZE ( 1 .. 21 ) ) OPTIONAL,
            forwardingOptions [6] IMPLICIT OCTET STRING ( SIZE ( 1 ) ) OPTIONAL,
            noReplyConditionTime [7] IMPLICIT INTEGER ( 5 .. 30 ) OPTIONAL,
            ... },
          ... },
      callBarringInfo [1] IMPLICIT SEQUENCE {
        ss-Code OCTET STRING ( SIZE ( 1 ) ) OPTIONAL,
        callBarringFeatureList SEQUENCE SIZE ( 1 .. 13 ) OF
          SEQUENCE {
            basicService CHOICE {
              bearerService [2] IMPLICIT OCTET STRING ( SIZE ( 1 ) ),
              teleservice [3] IMPLICIT OCTET STRING ( SIZE ( 1 ) )} OPTIONAL,
              ss-Status [4] IMPLICIT OCTET STRING ( SIZE ( 1 ) ) OPTIONAL,
              ... },
            ... },
      ss-Data [3] IMPLICIT SEQUENCE {
        ss-Code OCTET STRING ( SIZE ( 1 ) ) OPTIONAL,
        ss-Status [4] IMPLICIT OCTET STRING ( SIZE ( 1 ) ) OPTIONAL,
        ss-SubscriptionOption CHOICE {
          cliRestrictionOption [2] IMPLICIT ENUMERATED {
            permanent ( 0 ),
            temporaryDefaultRestricted ( 1 ),
            temporaryDefaultAllowed ( 2 )},
          overrideCategory [1] IMPLICIT ENUMERATED {
            overrideEnabled ( 0 ),
            overrideDisabled ( 1 )}} OPTIONAL,
          basicServiceGroupList SEQUENCE SIZE ( 1 .. 13 ) OF
            CHOICE {
              bearerService [2] IMPLICIT OCTET STRING ( SIZE ( 1 ) ),
              teleservice [3] IMPLICIT OCTET STRING ( SIZE ( 1 ) )} OPTIONAL,
              ... }
        }
    }
  ERRORS {
    -- systemFailure -- localValue : 34,
    -- dataMissing -- localValue : 35,
    -- unexpectedDataValue -- localValue : 36,
    -- bearerServiceNotProvisioned -- localValue : 10,
    -- teleserviceNotProvisioned -- localValue : 11,
    -- callBarred -- localValue : 13,
    -- illegalSS-Operation -- localValue : 16,
    -- ss-ErrorStatus -- localValue : 17}
  ::= localValue : 11

activateSS OPERATION
  ARGUMENT
    ss-ForBS SEQUENCE {
      ss-Code OCTET STRING ( SIZE ( 1 ) ),
      basicService CHOICE {
        bearerService [2] IMPLICIT OCTET STRING ( SIZE ( 1 ) ),
        teleservice [3] IMPLICIT OCTET STRING ( SIZE ( 1 ) )} OPTIONAL,
      ... }
  RESULT
    ss-Info CHOICE {
      forwardingInfo [0] IMPLICIT SEQUENCE {
        ss-Code OCTET STRING ( SIZE ( 1 ) ) OPTIONAL,
        forwardingFeatureList SEQUENCE SIZE ( 1 .. 13 ) OF
          SEQUENCE {
            basicService CHOICE {
              bearerService [2] IMPLICIT OCTET STRING ( SIZE ( 1 ) ),
              teleservice [3] IMPLICIT OCTET STRING ( SIZE ( 1 ) )} OPTIONAL,
            ss-Status [4] IMPLICIT OCTET STRING ( SIZE ( 1 ) ) OPTIONAL,
            forwardedToNumber [5] IMPLICIT OCTET STRING ( SIZE ( 1 .. 20 ) ) ( SIZE ( 1 .. 9
          ) ) OPTIONAL,
            forwardedToSubaddress [8] IMPLICIT OCTET STRING ( SIZE ( 1 .. 21 ) ) OPTIONAL,
            forwardingOptions [6] IMPLICIT OCTET STRING ( SIZE ( 1 ) ) OPTIONAL,
            noReplyConditionTime [7] IMPLICIT INTEGER ( 5 .. 30 ) OPTIONAL,
            ... },
          ... },
      callBarringInfo [1] IMPLICIT SEQUENCE {
        ss-Code OCTET STRING ( SIZE ( 1 ) ) OPTIONAL,
        callBarringFeatureList SEQUENCE SIZE ( 1 .. 13 ) OF
          SEQUENCE {
            basicService CHOICE {
              bearerService [2] IMPLICIT OCTET STRING ( SIZE ( 1 ) ),
              teleservice [3] IMPLICIT OCTET STRING ( SIZE ( 1 ) )} OPTIONAL,
              ss-Status [4] IMPLICIT OCTET STRING ( SIZE ( 1 ) ) OPTIONAL,
              ... },
            ... },
      ss-Data [3] IMPLICIT SEQUENCE {
        ss-Code OCTET STRING ( SIZE ( 1 ) ) OPTIONAL,
        ss-Status [4] IMPLICIT OCTET STRING ( SIZE ( 1 ) ) OPTIONAL,
        ss-SubscriptionOption CHOICE {
          cliRestrictionOption [2] IMPLICIT ENUMERATED {
            permanent ( 0 ),
            temporaryDefaultRestricted ( 1 ),
            temporaryDefaultAllowed ( 2 )},
          overrideCategory [1] IMPLICIT ENUMERATED {
            overrideEnabled ( 0 ),
            overrideDisabled ( 1 )}} OPTIONAL,
          basicServiceGroupList SEQUENCE SIZE ( 1 .. 13 ) OF
            CHOICE {
              bearerService [2] IMPLICIT OCTET STRING ( SIZE ( 1 ) ),
              teleservice [3] IMPLICIT OCTET STRING ( SIZE ( 1 ) )} OPTIONAL,
              ... }
        }
    }
  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

```

```

        forwardedToSubaddress [8] IMPLICIT OCTET STRING ( SIZE ( 1 .. 21 ) ) OPTIONAL,
        forwardingOptions [6] IMPLICIT OCTET STRING ( SIZE ( 1 ) ) OPTIONAL,
        noReplyConditionTime [7] IMPLICIT INTEGER ( 5 .. 30 ) OPTIONAL,
        ... },
    ... },
    callBarringInfo [1] IMPLICIT SEQUENCE {
        ss-Code      OCTET STRING ( SIZE ( 1 ) ) OPTIONAL,
        callBarringFeatureList SEQUENCE SIZE ( 1 .. 13 ) OF
            SEQUENCE {
                basicService CHOICE {
                    bearerService [2] IMPLICIT OCTET STRING ( SIZE ( 1 ) ),
                    teleservice [3] IMPLICIT OCTET STRING ( SIZE ( 1 ) ) } OPTIONAL,
                    ss-Status [4] IMPLICIT OCTET STRING ( SIZE ( 1 ) ) OPTIONAL,
                    ... },
                ... },
    ss-Data [3] IMPLICIT SEQUENCE {
        ss-Code      OCTET STRING ( SIZE ( 1 ) ) OPTIONAL,
        ss-Status [4] IMPLICIT OCTET STRING ( SIZE ( 1 ) ) OPTIONAL,
        ss-SubscriptionOption CHOICE {
            cliRestrictionOption [2] IMPLICIT ENUMERATED {
                permanent ( 0 ),
                temporaryDefaultRestricted ( 1 ),
                temporaryDefaultAllowed ( 2 ) },
            overrideCategory [1] IMPLICIT ENUMERATED {
                overrideEnabled ( 0 ),
                overrideDisabled ( 1 ) } } OPTIONAL,
        basicServiceGroupList SEQUENCE SIZE ( 1 .. 13 ) OF
            CHOICE {
                bearerService [2] IMPLICIT OCTET STRING ( SIZE ( 1 ) ),
                teleservice [3] IMPLICIT OCTET STRING ( SIZE ( 1 ) ) } OPTIONAL,
                ... } }
ERRORS {
    -- systemFailure -- localValue : 34,
    -- dataMissing -- localValue : 35,
    -- unexpectedDataValue -- localValue : 36,
    -- bearerServiceNotProvisioned -- localValue : 10,
    -- teleserviceNotProvisioned -- localValue : 11,
    -- callBarred -- localValue : 13,
    -- illegalSS-Operation -- localValue : 16,
    -- ss-ErrorStatus -- localValue : 17,
    -- ss-SubscriptionViolation -- localValue : 19,
    -- ss-Incompatibility -- localValue : 20,
    -- negativePW-Check -- localValue : 38,
    -- numberOfPW-AttemptsViolation -- localValue : 43}
::= localValue : 12

deactivateSS OPERATION
    ARGUMENT
        ss-ForBS SEQUENCE {
            ss-Code      OCTET STRING ( SIZE ( 1 ) ),
            basicService CHOICE {
                bearerService [2] IMPLICIT OCTET STRING ( SIZE ( 1 ) ),
                teleservice [3] IMPLICIT OCTET STRING ( SIZE ( 1 ) ) } OPTIONAL,
                ... }
    RESULT
        ss-Info CHOICE {
            forwardingInfo [0] IMPLICIT SEQUENCE {
                ss-Code      OCTET STRING ( SIZE ( 1 ) ) OPTIONAL,
                forwardingFeatureList SEQUENCE SIZE ( 1 .. 13 ) OF
                    SEQUENCE {
                        basicService CHOICE {
                            bearerService [2] IMPLICIT OCTET STRING ( SIZE ( 1 ) ),
                            teleservice [3] IMPLICIT OCTET STRING ( SIZE ( 1 ) ) } OPTIONAL,
                            ss-Status [4] IMPLICIT OCTET STRING ( SIZE ( 1 ) ) OPTIONAL,
                            forwardedToNumber [5] IMPLICIT OCTET STRING ( SIZE ( 1 .. 20 ) ) ( SIZE ( 1 .. 9
) ) OPTIONAL,
                            forwardedToSubaddress [8] IMPLICIT OCTET STRING ( SIZE ( 1 .. 21 ) ) OPTIONAL,
                            forwardingOptions [6] IMPLICIT OCTET STRING ( SIZE ( 1 ) ) OPTIONAL,
                            noReplyConditionTime [7] IMPLICIT INTEGER ( 5 .. 30 ) OPTIONAL,
                            ... },
                        ... },
            callBarringInfo [1] IMPLICIT SEQUENCE {
                ss-Code      OCTET STRING ( SIZE ( 1 ) ) OPTIONAL,
                callBarringFeatureList SEQUENCE SIZE ( 1 .. 13 ) OF
                    SEQUENCE {
                        basicService CHOICE {
                            bearerService [2] IMPLICIT OCTET STRING ( SIZE ( 1 ) ),
                            teleservice [3] IMPLICIT OCTET STRING ( SIZE ( 1 ) ) } OPTIONAL,
                            ss-Status [4] IMPLICIT OCTET STRING ( SIZE ( 1 ) ) OPTIONAL,
                            ... },
                        ... },
            ss-Data [3] IMPLICIT SEQUENCE {
                ss-Code      OCTET STRING ( SIZE ( 1 ) ) OPTIONAL,
                ss-Status [4] IMPLICIT OCTET STRING ( SIZE ( 1 ) ) OPTIONAL,
                ss-SubscriptionOption CHOICE {

```

```

        cliRestrictionOption [2] IMPLICIT ENUMERATED {
            permanent (0 ),
            temporaryDefaultRestricted (1 ),
            temporaryDefaultAllowed (2 )},
        overrideCategory [1] IMPLICIT ENUMERATED {
            overrideEnabled (0 ),
            overrideDisabled (1 )}} OPTIONAL,
        basicServiceGroupList SEQUENCE SIZE (1 .. 13 ) OF
        CHOICE {
            bearerService [2] IMPLICIT OCTET STRING ( SIZE (1 ) ),
            teleservice [3] IMPLICIT OCTET STRING ( SIZE (1 ) )} OPTIONAL,
        ... }}
ERRORS {
    -- systemFailure -- localValue : 34,
    -- dataMissing -- localValue : 35,
    -- unexpectedDataValue -- localValue : 36,
    -- bearerServiceNotProvisioned -- localValue : 10,
    -- teleserviceNotProvisioned -- localValue : 11,
    -- callBarred -- localValue : 13,
    -- illegalSS-Operation -- localValue : 16,
    -- ss-ErrorStatus -- localValue : 17,
    -- ss-SubscriptionViolation -- localValue : 19,
    -- negativePW-Check -- localValue : 38,
    -- numberOfPW-AttemptsViolation -- localValue : 43}
::= localValue : 13

interrogateSS OPERATION
    ARGUMENT
        ss-ForBS SEQUENCE {
            ss-Code OCTET STRING ( SIZE (1 ) ),
            basicService CHOICE {
                bearerService [2] IMPLICIT OCTET STRING ( SIZE (1 ) ),
                teleservice [3] IMPLICIT OCTET STRING ( SIZE (1 ) )} OPTIONAL,
            ... }
    RESULT
        interrogateSS-Res CHOICE {
            ss-Status [0] IMPLICIT OCTET STRING ( SIZE (1 ) ),
            basicServiceGroupList [2] IMPLICIT SEQUENCE SIZE (1 .. 13 ) OF
            CHOICE {
                bearerService [2] IMPLICIT OCTET STRING ( SIZE (1 ) ),
                teleservice [3] IMPLICIT OCTET STRING ( SIZE (1 ) )},
            forwardingFeatureList [3] IMPLICIT SEQUENCE SIZE (1 .. 13 ) OF
            SEQUENCE {
                basicService CHOICE {
                    bearerService [2] IMPLICIT OCTET STRING ( SIZE (1 ) ),
                    teleservice [3] IMPLICIT OCTET STRING ( SIZE (1 ) )} OPTIONAL,
                ss-Status [4] IMPLICIT OCTET STRING ( SIZE (1 ) ) OPTIONAL,
                forwardedToNumber [5] IMPLICIT OCTET STRING ( SIZE (1 .. 20 ) ) ( SIZE (1 .. 9 ) )
                OPTIONAL,
                forwardedToSubaddress [8] IMPLICIT OCTET STRING ( SIZE (1 .. 21 ) ) OPTIONAL,
                forwardingOptions [6] IMPLICIT OCTET STRING ( SIZE (1 ) ) OPTIONAL,
                noReplyConditionTime [7] IMPLICIT INTEGER ( 5 .. 30 ) OPTIONAL,
                ... },
            cli-RestrictionInfo [4] IMPLICIT SEQUENCE {
                ss-Status OCTET STRING ( SIZE (1 ) ),
                cliRestrictionOption ENUMERATED {
                    permanent (0 ),
                    temporaryDefaultRestricted (1 ),
                    temporaryDefaultAllowed (2 )} OPTIONAL,
                ... }}
    ERRORS {
        -- systemFailure -- localValue : 34,
        -- dataMissing -- localValue : 35,
        -- unexpectedDataValue -- localValue : 36,
        -- 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 ) ),
            ... }
    RESULT
        ussd-Res SEQUENCE {
            ussd-DataCodingScheme OCTET STRING ( SIZE (1 ) ),
            ussd-String OCTET STRING ( SIZE (1 .. 160 ) ),
            ... }
    ERRORS {
        -- systemFailure -- localValue : 34,
        -- dataMissing -- localValue : 35,

```

```

-- unexpectedDataValue -- localValue : 36,
-- unknownAlphabet -- localValue : 71,
-- callBarred -- localValue : 13}
::= localValue : 59

unstructuredSS-Request OPERATION
  ARGUMENT
    ussd-Arg SEQUENCE {
      ussd-DataCodingScheme OCTET STRING ( SIZE (1) ),
      ussd-String OCTET STRING ( SIZE (1 .. 160) ),
      ... }
  RESULT
    ussd-Res SEQUENCE {
      ussd-DataCodingScheme OCTET STRING ( SIZE (1) ),
      ussd-String OCTET STRING ( SIZE (1 .. 160) ),
      ... }
  ERRORS {
    -- systemFailure -- localValue : 34,
    -- dataMissing -- localValue : 35,
    -- unexpectedDataValue -- localValue : 36,
    -- absentSubscriber -- localValue : 27,
    -- illegalSubscriber -- localValue : 9,
    -- illegalEquipment -- localValue : 12,
    -- unknownAlphabet -- localValue : 71,
    -- ussd-Busy -- localValue : 72}
::= localValue : 60

unstructuredSS-Notify OPERATION
  ARGUMENT
    ussd-Arg SEQUENCE {
      ussd-DataCodingScheme OCTET STRING ( SIZE (1) ),
      ussd-String OCTET STRING ( SIZE (1 .. 160) ),
      ... }
  ERRORS {
    -- systemFailure -- localValue : 34,
    -- dataMissing -- localValue : 35,
    -- unexpectedDataValue -- localValue : 36,
    -- absentSubscriber -- localValue : 27,
    -- illegalSubscriber -- localValue : 9,
    -- illegalEquipment -- localValue : 12,
    -- unknownAlphabet -- localValue : 71,
    -- ussd-Busy -- localValue : 72}
::= localValue : 61

registerPassword OPERATION
  ARGUMENT
    ss-Code OCTET STRING ( SIZE (1) )
  RESULT
    newPassword NumericString ( FROM ("0"|"1"|"2"|"3"|"4"|"5"|"6"|"7"|"8"|"9" )|SIZE (4) )
  ERRORS {
    -- systemFailure -- localValue : 34,
    -- dataMissing -- localValue : 35,
    -- unexpectedDataValue -- localValue : 36,
    -- callBarred -- localValue : 13,
    -- ss-SubscriptionViolation -- localValue : 19,
    -- pw-RegistrationFailure -- localValue : 37,
    -- negativePW-Check -- localValue : 38,
    -- numberOfPW-AttemptsViolation -- localValue : 43}
  LINKED {
    -- getPassword -- localValue : 18}
::= localValue : 17

getPassword OPERATION
  ARGUMENT
    guidanceInfo ENUMERATED {
      enterPW (0),
      enterNewPW (1),
      enterNewPW-Again (2)}
  RESULT
    currentPassword NumericString ( FROM ("0"|"1"|"2"|"3"|"4"|"5"|"6"|"7"|"8"|"9" )|SIZE (4) )
::= localValue : 18

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,
    ... }
RESULT
routingInfoForSM-Res SEQUENCE {
    imsi OCTET STRING ( SIZE ( 3 .. 8 ) ),
    locationInfoWithLMSI [0] IMPLICIT SEQUENCE {
        msc-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,
                    ... },
            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,
                        ... }
RESULT
    mo-forwardSM-Res SEQUENCE {
        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,
    ... }
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,
    ... }
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 )} ( 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 )},
            ... }
    }
    ERRORS {
        -- dataMissing -- localValue : 35,
        -- unexpectedDataValue -- localValue : 36,
        -- facilityNotSupported -- localValue : 21,
        -- unknownSubscriber -- localValue : 1}
    ::= localValue : 66

provideSubscriberInfo OPERATION
    ARGUMENT
        provideSubscriberInfoArg SEQUENCE {
            imsi [0] IMPLICIT OCTET STRING ( SIZE (3 .. 8 ) ),
            lmsi [1] IMPLICIT OCTET STRING ( SIZE (4 ) ) OPTIONAL,

```

```

requestedInfo [2] IMPLICIT SEQUENCE {
    locationInformation [0] IMPLICIT NULL OPTIONAL,
    subscriberState [1] IMPLICIT NULL OPTIONAL,
    extensionContainer [2] IMPLICIT SEQUENCE {
        privateExtensionList [0] IMPLICIT SEQUENCE SIZE (1 .. 10 ) OF
            SEQUENCE {
                extId      MAP-EXTENSION .&extensionId ( {
                    '...'} ) ,
                extType    MAP-EXTENSION .&ExtensionType ( {
                    '...'} { @extId } ) OPTIONAL} OPTIONAL,
        pcs-Extensions [1] IMPLICIT SEQUENCE {
            ... } OPTIONAL,
            ... } OPTIONAL,
        ... },
extensionContainer [3] IMPLICIT SEQUENCE {
    privateExtensionList [0] IMPLICIT SEQUENCE SIZE (1 .. 10 ) OF
        SEQUENCE {
            extId      MAP-EXTENSION .&extensionId ( {
                '...'} ) ,
            extType    MAP-EXTENSION .&ExtensionType ( {
                '...'} { @extId } ) OPTIONAL} OPTIONAL,
        pcs-Extensions [1] IMPLICIT SEQUENCE {
            ... } OPTIONAL,
            ... } OPTIONAL,
        ... }
... }

RESULT
provideSubscriberInfoRes SEQUENCE {
    subscriberInfo SEQUENCE {
        locationInformation [0] IMPLICIT SEQUENCE {
            ageOfLocationInformation INTEGER ( 0 .. 32767 ) OPTIONAL,
            geographicalInformation [0] IMPLICIT OCTET STRING ( SIZE (8 ) ) OPTIONAL,
            vlr-number [1] IMPLICIT OCTET STRING ( SIZE (1 .. 20 ) ) ( SIZE (1 .. 9 ) )
OPTIONAL,

            locationNumber [2] IMPLICIT OCTET STRING ( SIZE (2 .. 10 ) ) OPTIONAL,
            cellIdOrLAI [3] CHOICE {
                cellIdFixedLength [0] IMPLICIT OCTET STRING ( SIZE (7 ) ),
                laiFixedLength [1] IMPLICIT OCTET STRING ( SIZE (5 ) ) } OPTIONAL,
            extensionContainer [4] IMPLICIT SEQUENCE {
                privateExtensionList [0] IMPLICIT SEQUENCE SIZE (1 .. 10 ) OF
                    SEQUENCE {
                        extId      MAP-EXTENSION .&extensionId ( {
                            '...'} ) ,
                        extType    MAP-EXTENSION .&ExtensionType ( {
                            '...'} { @extId } ) OPTIONAL} OPTIONAL,
                pcs-Extensions [1] IMPLICIT SEQUENCE {
                    ... } OPTIONAL,
                    ... } OPTIONAL,
                ... } OPTIONAL,
            subscriberState [1] CHOICE {
                assumedIdle [0] IMPLICIT NULL,
                camelBusy [1] IMPLICIT NULL,
                netDetNotReachable ENUMERATED {
                    msPurged (0 ),
                    imsiDetached (1 ),
                    restrictedArea (2 ),
                    notRegistered (3 ) },
                notProvidedFromVLR [2] IMPLICIT NULL} OPTIONAL,
            extensionContainer [2] IMPLICIT SEQUENCE {
                privateExtensionList [0] IMPLICIT SEQUENCE SIZE (1 .. 10 ) OF
                    SEQUENCE {
                        extId      MAP-EXTENSION .&extensionId ( {
                            '...'} ) ,
                        extType    MAP-EXTENSION .&ExtensionType ( {
                            '...'} { @extId } ) OPTIONAL} OPTIONAL,
                pcs-Extensions [1] IMPLICIT SEQUENCE {
                    ... } OPTIONAL,
                    ... } OPTIONAL,
                ... },
            extensionContainer SEQUENCE {
                privateExtensionList [0] IMPLICIT SEQUENCE SIZE (1 .. 10 ) OF
                    SEQUENCE {
                        extId      MAP-EXTENSION .&extensionId ( {
                            '...'} ) ,
                        extType    MAP-EXTENSION .&ExtensionType ( {
                            '...'} { @extId } ) OPTIONAL} OPTIONAL,

```



```

pcs-Extensions [1] IMPLICIT SEQUENCE {
    ... } OPTIONAL,
    ... } OPTIONAL,
... }
ERRORS {
-- dataMissing -- localValue : 35,
-- unexpectedDataValue -- localValue : 36}
 ::= localValue : 70

anyTimeInterrogation OPERATION
ARGUMENT
anyTimeInterrogationArg SEQUENCE {
subscriberIdentity [0] CHOICE {
imsi      [0] IMPLICIT OCTET STRING ( SIZE ( 3 .. 8 ) ),
msisdn    [1] IMPLICIT OCTET STRING ( SIZE ( 1 .. 20 ) ) ( SIZE ( 1 .. 9 ) ) },
requestedInfo [1] IMPLICIT SEQUENCE {
locationInformation [0] IMPLICIT NULL OPTIONAL,
subscriberState [1] IMPLICIT NULL OPTIONAL,
extensionContainer [2] IMPLICIT SEQUENCE {
privateExtensionList [0] IMPLICIT SEQUENCE SIZE ( 1 .. 10 ) OF
SEQUENCE {
extId          MAP-EXTENSION .&extensionId ( {
'...' } ),
extType       MAP-EXTENSION .&ExtensionType ( {
'...' } { @extId   } ) OPTIONAL} OPTIONAL,
pcs-Extensions [1] IMPLICIT SEQUENCE {
... } OPTIONAL,
... } OPTIONAL,
... },
gsmSCF-Address [3] IMPLICIT OCTET STRING ( SIZE ( 1 .. 20 ) ) ( SIZE ( 1 .. 9 ) ),
extensionContainer [2] IMPLICIT SEQUENCE {
privateExtensionList [0] IMPLICIT SEQUENCE SIZE ( 1 .. 10 ) OF
SEQUENCE {
extId          MAP-EXTENSION .&extensionId ( {
'...' } ),
extType       MAP-EXTENSION .&ExtensionType ( {
'...' } { @extId   } ) OPTIONAL} OPTIONAL,
pcs-Extensions [1] IMPLICIT SEQUENCE {
... } OPTIONAL,
... } OPTIONAL,
... }
RESULT
anyTimeInterrogationRes SEQUENCE {
subscriberInfo SEQUENCE {
locationInformation [0] IMPLICIT SEQUENCE {
ageOfLocationInformation INTEGER ( 0 .. 32767 ) OPTIONAL,
geographicalInformation [0] IMPLICIT OCTET STRING ( SIZE ( 8 ) ) OPTIONAL,
vlr-number [1] IMPLICIT OCTET STRING ( SIZE ( 1 .. 20 ) ) ( SIZE ( 1 .. 9 ) )
OPTIONAL,
locationNumber [2] IMPLICIT OCTET STRING ( SIZE ( 2 .. 10 ) ) OPTIONAL,
cellIdOrLAI [3] CHOICE {
cellIdFixedLength [0] IMPLICIT OCTET STRING ( SIZE ( 7 ) ),
laiFixedLength [1] IMPLICIT OCTET STRING ( SIZE ( 5 ) ) } OPTIONAL,
extensionContainer [4] IMPLICIT SEQUENCE {
privateExtensionList [0] IMPLICIT SEQUENCE SIZE ( 1 .. 10 ) OF
SEQUENCE {
extId          MAP-EXTENSION .&extensionId ( {
'...' } ),
extType       MAP-EXTENSION .&ExtensionType ( {
'...' } { @extId   } ) OPTIONAL} OPTIONAL,
pcs-Extensions [1] IMPLICIT SEQUENCE {
... } OPTIONAL,
... } OPTIONAL,
... } 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

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

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,
                        ... }
        ::= localValue : 1

numberChanged ERROR
PARAMETER
    numberChangedParam SEQUENCE {
        extensionContainer SEQUENCE {
            privateExtensionList [0] IMPLICIT SEQUENCE SIZE (1 .. 10 ) OF
                SEQUENCE {
                    extId      MAP-EXTENSION .&extensionId ( {
                        '...'} ) ,
                    extType    MAP-EXTENSION .&ExtensionType ( {
                        '...'} { @extId } ) OPTIONAL} OPTIONAL,
                    pcs-Extensions [1] IMPLICIT SEQUENCE {
                        ... } OPTIONAL,
                        ... } OPTIONAL,
                        ... }
        ::= localValue : 44

unknownMSC ERROR
::= localValue : 3

unidentifiedSubscriber ERROR
PARAMETER
    unidentifiedSubParam SEQUENCE {
        extensionContainer SEQUENCE {
            privateExtensionList [0] IMPLICIT SEQUENCE SIZE (1 .. 10 ) OF
                SEQUENCE {
                    extId      MAP-EXTENSION .&extensionId ( {
                        '...'} ) ,
                    extType    MAP-EXTENSION .&ExtensionType ( {
                        '...'} { @extId } ) OPTIONAL} OPTIONAL,

```

```

        pcs-Extensions [1] IMPLICIT SEQUENCE {
            ... } OPTIONAL,
            ... } OPTIONAL,
            ... }
 ::= localValue : 5

unknownEquipment ERROR
 ::= localValue : 7

roamingNotAllowed ERROR
    PARAMETER
        roamingNotAllowedParam SEQUENCE {
            roamingNotAllowedCause ENUMERATED {
                plmnRoamingNotAllowed (0 ),
                operatorDeterminedBarring (3 )},
            extensionContainer SEQUENCE {
                privateExtensionList [0] IMPLICIT SEQUENCE SIZE (1 .. 10 ) OF
                    SEQUENCE {
                        extId          MAP-EXTENSION .&extensionId ( {
                            '...'} ) ,
                        extType      MAP-EXTENSION .&ExtensionType ( {
                            '...'} { @extId } ) OPTIONAL} OPTIONAL,
                pcs-Extensions [1] IMPLICIT SEQUENCE {
                    ... } OPTIONAL,
                    ... } OPTIONAL,
                    ... }
 ::= localValue : 8

illegalSubscriber ERROR
    PARAMETER
        illegalSubscriberParam SEQUENCE {
            extensionContainer SEQUENCE {
                privateExtensionList [0] IMPLICIT SEQUENCE SIZE (1 .. 10 ) OF
                    SEQUENCE {
                        extId          MAP-EXTENSION .&extensionId ( {
                            '...'} ) ,
                        extType      MAP-EXTENSION .&ExtensionType ( {
                            '...'} { @extId } ) OPTIONAL} OPTIONAL,
                pcs-Extensions [1] IMPLICIT SEQUENCE {
                    ... } OPTIONAL,
                    ... } OPTIONAL,
                    ... }
 ::= localValue : 9

illegalEquipment ERROR
    PARAMETER
        illegalEquipmentParam SEQUENCE {
            extensionContainer SEQUENCE {
                privateExtensionList [0] IMPLICIT SEQUENCE SIZE (1 .. 10 ) OF
                    SEQUENCE {
                        extId          MAP-EXTENSION .&extensionId ( {
                            '...'} ) ,
                        extType      MAP-EXTENSION .&ExtensionType ( {
                            '...'} { @extId } ) OPTIONAL} OPTIONAL,
                pcs-Extensions [1] IMPLICIT SEQUENCE {
                    ... } OPTIONAL,
                    ... } OPTIONAL,
                    ... }
 ::= localValue : 12

bearerServiceNotProvisioned ERROR
    PARAMETER
        bearerServNotProvParam SEQUENCE {
            extensionContainer SEQUENCE {
                privateExtensionList [0] IMPLICIT SEQUENCE SIZE (1 .. 10 ) OF
                    SEQUENCE {
                        extId          MAP-EXTENSION .&extensionId ( {
                            '...'} ) ,
                        extType      MAP-EXTENSION .&ExtensionType ( {
                            '...'} { @extId } ) OPTIONAL} OPTIONAL,
                pcs-Extensions [1] IMPLICIT SEQUENCE {
                    ... } OPTIONAL,
                    ... } OPTIONAL,
                    ... }
 ::= localValue : 10

teleserviceNotProvisioned ERROR

```

```

PARAMETER
    teleservNotProvParam SEQUENCE {
        extensionContainer SEQUENCE {
            privateExtensionList [0] IMPLICIT SEQUENCE SIZE (1 .. 10 ) OF
                SEQUENCE {
                    extId      MAP-EXTENSION .&extensionId ( {
                        '...' ) ,
                    extType    MAP-EXTENSION .&ExtensionType ( {
                        '...' { @extId } ) OPTIONAL} OPTIONAL,
                    pcs-Extensions [1] IMPLICIT SEQUENCE {
                        ... } OPTIONAL,
                        ... } OPTIONAL,
                        ... }
                ::= localValue : 11

noHandoverNumberAvailable ERROR
    ::= localValue : 25

subsequentHandoverFailure ERROR
    ::= localValue : 26

tracingBufferFull ERROR
    PARAMETER
        tracingBufferFullParam SEQUENCE {
            extensionContainer SEQUENCE {
                privateExtensionList [0] IMPLICIT SEQUENCE SIZE (1 .. 10 ) OF
                    SEQUENCE {
                        extId      MAP-EXTENSION .&extensionId ( {
                            '...' ) ,
                        extType    MAP-EXTENSION .&ExtensionType ( {
                            '...' { @extId } ) OPTIONAL} OPTIONAL,
                        pcs-Extensions [1] IMPLICIT SEQUENCE {
                            ... } OPTIONAL,
                            ... } OPTIONAL,
                            ... }
                    ::= localValue : 40

noRoamingNumberAvailable ERROR
    PARAMETER
        noRoamingNbParam SEQUENCE {
            extensionContainer SEQUENCE {
                privateExtensionList [0] IMPLICIT SEQUENCE SIZE (1 .. 10 ) OF
                    SEQUENCE {
                        extId      MAP-EXTENSION .&extensionId ( {
                            '...' ) ,
                        extType    MAP-EXTENSION .&ExtensionType ( {
                            '...' { @extId } ) OPTIONAL} OPTIONAL,
                        pcs-Extensions [1] IMPLICIT SEQUENCE {
                            ... } OPTIONAL,
                            ... } OPTIONAL,
                            ... }
                    ::= localValue : 39

absentSubscriber ERROR
    PARAMETER
        absentSubscriberParam SEQUENCE {
            extensionContainer SEQUENCE {
                privateExtensionList [0] IMPLICIT SEQUENCE SIZE (1 .. 10 ) OF
                    SEQUENCE {
                        extId      MAP-EXTENSION .&extensionId ( {
                            '...' ) ,
                        extType    MAP-EXTENSION .&ExtensionType ( {
                            '...' { @extId } ) OPTIONAL} OPTIONAL,
                        pcs-Extensions [1] IMPLICIT SEQUENCE {
                            ... } OPTIONAL,
                            ... } OPTIONAL,
                            ... }
                    ::= 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,
            ... }
 ::= 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,
                                    ... }
                                }
                            }
                    }
                ::= 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 : 47

```

```

        ... } 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

illegalSS-Operation ERROR
 ::= localValue : 16

ss-ErrorStatus ERROR
PARAMETER
    ss-Status OCTET STRING ( SIZE (1 ) )
 ::= localValue : 17

ss-NotAvailable ERROR
 ::= localValue : 18

ss-SubscriptionViolation ERROR
 ::= localValue : 19

ss-Incompatibility ERROR
PARAMETER
    ss-IncompatibilityCause SEQUENCE {
        ss-Code [1] IMPLICIT OCTET STRING ( SIZE (1 ) ) OPTIONAL,
        basicService CHOICE {
            bearerService [2] IMPLICIT OCTET STRING ( SIZE (1 ) ),
            teleservice [3] IMPLICIT OCTET STRING ( SIZE (1 ) ) } OPTIONAL,
        ss-Status [4] IMPLICIT OCTET STRING ( SIZE (1 ) ) OPTIONAL,
    }

```

```

... }
::= localValue : 20

unknownAlphabet ERROR
::= localValue : 71

ussd-Busy ERROR
::= localValue : 72

pw-RegistrationFailure ERROR
  PARAMETER
    pw-RegistrationFailureCause ENUMERATED {
      undetermined (0 ),
      invalidFormat (1 ),
      newPasswordsMismatch (2 )}
    ::= localValue : 37

negativePW-Check ERROR
::= localValue : 38

numberOfPW-AttemptsViolation ERROR
::= localValue : 43

subscriberBusyForMT-SMS ERROR
  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,
              ... }
            ::= 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,
              ... }
      ::= localValue : 6
    }
  END
```

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

```
-- Expanded ASN1 Module 'MAP-DialogueInformation'
--SIEMENS ASN.1 Compiler      P3.90M (04-07-00-00-64-00)
--      Date: 98-07-23 Time: 09:09:07

MAP-DialogueInformation { 0 identified-organization (4) etsi (0) mobileDomain (0) gsm-Network (1)
modules (3) map-DialogueInformation (3) version3 (3) }

DEFINITIONS

 ::=

BEGIN

EXPORTS
    map-DialogueAS,
    MAP-DialoguePDU
;

map-DialogueAS OBJECT IDENTIFIER ::= { ccitt (0) identified-organization (4) etsi (0)
mobileDomain (0) gsm-Network (1) 1 map-DialoguePDU (1) version1 (1) }

MAP-DialoguePDU ::= CHOICE {
    map-open [0] IMPLICIT SEQUENCE {
        destinationReference [0] IMPLICIT OCTET STRING ( SIZE (1 .. 20 ) ) OPTIONAL,
        originationReference [1] IMPLICIT OCTET STRING ( SIZE (1 .. 20 ) ) OPTIONAL,
        ... ,
        extensionContainer SEQUENCE {
            privateExtensionList [0] IMPLICIT SEQUENCE SIZE (1 .. 10 ) OF
                SEQUENCE {
                    extId      MAP-EXTENSION .&extensionId ( {
                        '
                        ... } ) ,
                    extType    MAP-EXTENSION .&ExtensionType ( {
                        '
                        ... } { @extId } ) OPTIONAL} OPTIONAL,
                    pcs-Extensions [1] IMPLICIT SEQUENCE {
                        ... } OPTIONAL,
                    ... } OPTIONAL},
        map-accept [1] IMPLICIT SEQUENCE {
            ... ,
            extensionContainer SEQUENCE {
                privateExtensionList [0] IMPLICIT SEQUENCE SIZE (1 .. 10 ) OF
                    SEQUENCE {
                        extId      MAP-EXTENSION .&extensionId ( {
                            '
                            ... } ) ,
                        extType    MAP-EXTENSION .&ExtensionType ( {
                            '
                            ... } { @extId } ) OPTIONAL} OPTIONAL,
                        pcs-Extensions [1] IMPLICIT SEQUENCE {
                            ... } OPTIONAL,
                            ... } OPTIONAL},
            map-close [2] IMPLICIT SEQUENCE {
                ... ,
                extensionContainer SEQUENCE {
                    privateExtensionList [0] IMPLICIT SEQUENCE SIZE (1 .. 10 ) OF
                        SEQUENCE {
                            extId      MAP-EXTENSION .&extensionId ( {
                                '
                                ... } ) ,
                            extType    MAP-EXTENSION .&ExtensionType ( {
                                '
                                ... } { @extId } ) OPTIONAL} OPTIONAL,
                            pcs-Extensions [1] IMPLICIT SEQUENCE {
                                ... } OPTIONAL,
                                ... } OPTIONAL},
                map-refuse [3] IMPLICIT SEQUENCE {
                    reason      ENUMERATED {
                        noReasonGiven (0 ),
                        invalidDestinationReference (1 ),
                        invalidOriginatingReference (2 )},
                    ... ,
                    extensionContainer SEQUENCE {
                        privateExtensionList [0] IMPLICIT SEQUENCE SIZE (1 .. 10 ) OF
                            SEQUENCE {
                                extId      MAP-EXTENSION .&extensionId ( {
                                    '
                                    ... } ) ,
                                extType    MAP-EXTENSION .&ExtensionType ( {
                                    '
                                    ... } { @extId } ) OPTIONAL} OPTIONAL,
```

```

    pcs-Extensions [1] IMPLICIT SEQUENCE {
        ... } OPTIONAL,
        ... } OPTIONAL},
map-userAbort [4] IMPLICIT SEQUENCE {
    map-UserAbortChoice CHOICE {
        userSpecificReason [0] IMPLICIT NULL,
        userResourceLimitation [1] IMPLICIT NULL,
        resourceUnavailable [2] IMPLICIT ENUMERATED {
            shortTermResourceLimitation (0 ),
            longTermResourceLimitation (1 )},
        applicationProcedureCancellation [3] IMPLICIT ENUMERATED {
            handoverCancellation (0 ),
            radioChannelRelease (1 ),
            networkPathRelease (2 ),
            callRelease (3 ),
            associatedProcedureFailure (4 ),
            tandemDialogueRelease (5 ),
            remoteOperationsFailure (6 )}},
    ... ,
    extensionContainer SEQUENCE {
        privateExtensionList [0] IMPLICIT SEQUENCE SIZE (1 .. 10 ) OF
            SEQUENCE {
                extId      MAP-EXTENSION .&extensionId ( {
                    '...'} ) ,
                extType    MAP-EXTENSION .&ExtensionType ( {
                    '...'} { @extId } ) OPTIONAL} OPTIONAL,
        pcs-Extensions [1] IMPLICIT SEQUENCE {
            ... } OPTIONAL,
            ... } OPTIONAL},
map-providerAbort [5] IMPLICIT SEQUENCE {
    map-ProviderAbortReason ENUMERATED {
        abnormalDialogue (0 ),
        invalidPDU (1 )},
    ... ,
    extensionContainer SEQUENCE {
        privateExtensionList [0] IMPLICIT SEQUENCE SIZE (1 .. 10 ) OF
            SEQUENCE {
                extId      MAP-EXTENSION .&extensionId ( {
                    '...'} ) ,
                extType    MAP-EXTENSION .&ExtensionType ( {
                    '...'} { @extId } ) OPTIONAL} OPTIONAL,
        pcs-Extensions [1] IMPLICIT SEQUENCE {
            ... } OPTIONAL,
            ... } OPTIONAL}}

MAP-OpenInfo ::= SEQUENCE {
    destinationReference [0] IMPLICIT OCTET STRING ( SIZE (1 .. 20 ) ) OPTIONAL,
    originationReference [1] IMPLICIT OCTET STRING ( SIZE (1 .. 20 ) ) OPTIONAL,
    ... ,
    extensionContainer SEQUENCE {
        privateExtensionList [0] IMPLICIT SEQUENCE SIZE (1 .. 10 ) OF
            SEQUENCE {
                extId      MAP-EXTENSION .&extensionId ( {
                    '...'} ) ,
                extType    MAP-EXTENSION .&ExtensionType ( {
                    '...'} { @extId } ) OPTIONAL} OPTIONAL,
        pcs-Extensions [1] IMPLICIT SEQUENCE {
            ... } OPTIONAL,
            ... } OPTIONAL}

MAP-AcceptInfo ::= SEQUENCE {
    ... ,
    extensionContainer SEQUENCE {
        privateExtensionList [0] IMPLICIT SEQUENCE SIZE (1 .. 10 ) OF
            SEQUENCE {
                extId      MAP-EXTENSION .&extensionId ( {
                    '...'} ) ,
                extType    MAP-EXTENSION .&ExtensionType ( {
                    '...'} { @extId } ) OPTIONAL} OPTIONAL,
        pcs-Extensions [1] IMPLICIT SEQUENCE {
            ... } OPTIONAL,
            ... } OPTIONAL}

MAP-CloseInfo ::= SEQUENCE {
    ... ,
    extensionContainer SEQUENCE {
        privateExtensionList [0] IMPLICIT SEQUENCE SIZE (1 .. 10 ) OF

```

```

SEQUENCE {
    extId      MAP-EXTENSION .&extensionId ( {
        '...} ) ,
    extType    MAP-EXTENSION .&ExtensionType ( {
        '...} { @extId } ) OPTIONAL} OPTIONAL,
pcs-Extensions [1] IMPLICIT SEQUENCE {
    ... } OPTIONAL,
... } OPTIONAL}

MAP-RefuseInfo ::= SEQUENCE {
    reason      ENUMERATED {
        noReasonGiven (0 ),
        invalidDestinationReference (1 ),
        invalidOriginatingReference (2 )},
    ... ,
    extensionContainer SEQUENCE {
        privateExtensionList [0] IMPLICIT SEQUENCE SIZE (1 .. 10 ) OF
            SEQUENCE {
                extId      MAP-EXTENSION .&extensionId ( {
                    '...} ) ,
                extType    MAP-EXTENSION .&ExtensionType ( {
                    '...} { @extId } ) OPTIONAL} OPTIONAL,
            pcs-Extensions [1] IMPLICIT SEQUENCE {
                ... } OPTIONAL,
            ... } OPTIONAL}

Reason ::= ENUMERATED {
    noReasonGiven (0 ),
    invalidDestinationReference (1 ),
    invalidOriginatingReference (2 )}

MAP-UserAbortInfo ::= SEQUENCE {
    map-UserAbortChoice CHOICE {
        userSpecificReason [0] IMPLICIT NULL,
        userResourceLimitation [1] IMPLICIT NULL,
        resourceUnavailable [2] IMPLICIT ENUMERATED {
            shortTermResourceLimitation (0 ),
            longTermResourceLimitation (1 )},
        applicationProcedureCancellation [3] IMPLICIT ENUMERATED {
            handoverCancellation (0 ),
            radioChannelRelease (1 ),
            networkPathRelease (2 ),
            callRelease (3 ),
            associatedProcedureFailure (4 ),
            tandemDialogueRelease (5 ),
            remoteOperationsFailure (6 )}},
    ... ,
    extensionContainer SEQUENCE {
        privateExtensionList [0] IMPLICIT SEQUENCE SIZE (1 .. 10 ) OF
            SEQUENCE {
                extId      MAP-EXTENSION .&extensionId ( {
                    '...} ) ,
                extType    MAP-EXTENSION .&ExtensionType ( {
                    '...} { @extId } ) OPTIONAL} OPTIONAL,
            pcs-Extensions [1] IMPLICIT SEQUENCE {
                ... } OPTIONAL,
            ... } OPTIONAL}

MAP-UserAbortChoice ::= CHOICE {
    userSpecificReason [0] IMPLICIT NULL,
    userResourceLimitation [1] IMPLICIT NULL,
    resourceUnavailable [2] IMPLICIT ENUMERATED {
        shortTermResourceLimitation (0 ),
        longTermResourceLimitation (1 )},
    applicationProcedureCancellation [3] IMPLICIT ENUMERATED {
        handoverCancellation (0 ),
        radioChannelRelease (1 ),
        networkPathRelease (2 ),
        callRelease (3 ),
        associatedProcedureFailure (4 ),
        tandemDialogueRelease (5 ),
        remoteOperationsFailure (6 )}}

ResourceUnavailableReason ::= ENUMERATED {
    shortTermResourceLimitation (0 ),
    longTermResourceLimitation (1 )}

ProcedureCancellationReason ::= ENUMERATED {
    handoverCancellation (0 ),

```

```
radioChannelRelease (1 ),
networkPathRelease (2 ),
callRelease (3 ),
associatedProcedureFailure (4 ),
tandemDialogueRelease (5 ),
remoteOperationsFailure (6 )}

MAP-ProviderAbortInfo ::= SEQUENCE {
    map-ProviderAbortReason ENUMERATED {
        abnormalDialogue (0 ),
        invalidPDU (1 )},
    ... ,
    extensionContainer SEQUENCE {
        privateExtensionList [0] IMPLICIT SEQUENCE SIZE (1 .. 10 ) OF
            SEQUENCE {
                extId      MAP-EXTENSION .&extensionId ( {
                    '...'} ) ,
                extType    MAP-EXTENSION .&ExtensionType ( {
                    '...'} { @extId } ) OPTIONAL} OPTIONAL,
        pcs-Extensions [1] IMPLICIT SEQUENCE {
            ... } OPTIONAL,
            ... } OPTIONAL}

MAP-ProviderAbortReason ::= ENUMERATED {
    abnormalDialogue (0 ),
    invalidPDU (1 )}

END
```

Annex C (informative): Formal protocol incompatibilities between versions 1 & 2 of MAP

C.1 Introduction

Annex C is not normative; it presents for information those parts of the MAP version 2 protocol which are not backward compatible with (i.e. a true superset of) the MAP version 1 protocol. For each incompatibility there is a commentary on the impact on the interworking of MAP version 1 and MAP version 2 entities.

SMG have decided that the MAP specification should include the operations and procedures used on the B interface (MSC/VLR) only for modelling purposes; the B interface cannot be implemented as an open interface. Hence any incompatibilities which affect operations used only on the B interface have no impact on the interworking of MAP version 1 and MAP version 2 entities.

C.2 Deletion of operations and errors

This subclause lists the operations and errors which have been completely removed from the MAP protocol.

C.2.1 Deletion of operation DeregisterMobileSubscriber

Although it is defined in the protocol in the MAP version 1 specification, this operation is not used (see subclause 1.2 of the latest phase 1 version of GSM 09.02).

C.2.2 Deletion of operation RegisterChargingInfo

There is no known implementation of MAP version 1 which supports this operation. The deletion has been approved by SMG.

C.2.3 Deletion of operation ForwardSS-Notification

There is no known implementation of MAP version 1 which supports this operation. The deletion has been approved by SMG.

C.2.4 Deletion of operations used only on the B-interface

The following operations (listed in alphabetical order) are not defined in the MAP version 2 protocol, because they are used only on the B-interface:

AllocateHandoverNumber; AttachIMSI; Authenticate; CompleteCall; DetachIMSI; ForwardNewTMSI; InvokeSS; Page; ProcessAccessRequest; ProvideIMSI; SearchForMobileSubscriber; SendHandoverReport; SendInfoForIncomingCall; SendInfoForOutgoingCall; SetCipheringMode; UpdateLocationArea.

C.2.5 Deletion of error InsufficientBearerCapabilities

This error is defined in the MAP version 1 protocol, but it is not specified for use with any operation.

C.3 Deletion of errors for operations

This subclause lists the cases where an error which is specified for use with an operation in the MAP version 1 specification is not specified for use with the same operation in the MAP version 2 specification.

C.3.1 Error NegativePW-Check for operation RegisterSS

Password checking is not used for the supplementary services to which registration applies.

C.3.2 Error NegativePW-Check for operation EraseSS

Password checking is not used for the supplementary services to which erasure applies.

C.3.3 Error NegativePW-Check for operation InterrogateSS

Password checking is not used for the interrogation of supplementary services.

C.3.4 Error CUG-Reject for operation SendRoutingInfoForSM

Closed User Group does not apply to the short message service.

C.4 Changes to definitions of data types

This subclause lists in alphabetical order those data types whose definitions have been directly or indirectly changed. For constructed data types, only the components which have been changed are mentioned. The commentary on the end effect of each change is given in subclauses C.6 (parameters of operations), C.7 (results of operations) or C.8 (errors of operations).

C.4.1 CUG-Feature

The type CUG-Feature was a sequence of four components; these have been deleted and replaced by three new components. It is used for the components of the type CUG-FeatureList.

C.4.2 CUG-FeatureList

The type CUG-FeatureList is a sequence of components of type CUG-Feature. It is used for a component of the type CUG-Info.

C.4.3 CUG-Info

The type CUG-Info is a sequence. One component of the sequence has been replaced by a component of a new type; the other component was a choice between a cug-Feature and a cug-FeatureList, and is now an optional cug-FeatureList. The type CUG-Info is used for a component of the type SS-Info.

C.4.4 CUG-RejectCause

The range of permitted values of the enumerated type CUG-RejectCause has been extended. The type is used for the parameter of the error CUG-Reject.

C.4.5 IMSI

The lower limit of the length of an IMSI has been increased from 2 octets to 3 octets. It is not possible to code a minimum length IMSI (MCC+MNC+MSIN) in 2 octets, so the theoretical lower limit of 2 octets should never be used by a MAP version 1 entity to send an IMSI; if it is, a MAP version 2 entity will treat it as a protocol error. Since this change has no practical impact it is not discussed further.

C.4.6 ISDN-AddressString

The upper limit of the length of an ISDN-AddressString has been reduced from 10 octets to 9 octets. The maximum length of an E.164 number is 15 digits; this can be encoded as a TBCD-string in 8 octets, plus a further octet to hold the type of number and number plan indicator. The cases where the ISDN-AddressString type was used in MAP version 1 to carry anything other than an E.164 number are described in subclause C.6; the other cases are not discussed further.

C.4.7 Password

In MAP version 1 the type Password was a choice between a printable string of length 4 to 8 octets or a numeric string of length 4 octets. It is now a numeric string of length 4 octets. The type Password is used for the result of the operation GetPassword.

C.4.8 RequestParameter

The enumerated type RequestParameter is no longer allowed to take the value requestCUG-Info. It is used as a component of the type RequestParameterList.

C.4.9 RequestParameterList

The type RequestParameterList is a sequence of components of type RequestParameter. The parameter of the operation SendParameters is a sequence of which one component is of type RequestParameterList.

C.4.10 SentParameter

The type SentParameter is a choice of which one component is of type SubscriberData. It is used as a component of the type SentParameterList.

C.4.11 SentParameterList

The type SentParameterList is a sequence whose components are of type SentParameter. The maximum number of components in the sequence has been reduced from 10 to 6.

The type SentParameterList is used for the result of the operation SendParameters.

C.4.12 SS-Data

The type SS-Data is a sequence of which one component is of type SS-SubscriptionOption. It is used for a component of the type SS-Info.

C.4.13 SS-Info

The type SS-Info is a choice of which one component is of type CUG-Info and another component is of type SS-Data. It is used for the result of the operations RegisterSS, EraseSS, ActivateSS and DeactivateSS, and for components of the type SS-InfoList.

C.4.14 SS-InfoList

The type SS-InfoList is a sequence of components of type SS-Info. It is used for a component of the type SubscriberData.

C.4.15 SS-SubscriptionOption

The type SS-SubscriptionOption was a choice from five components: perCallBasis (used for the CLIR supplementary service); notificationToHeldRetrievedParty (used for the Call Transfer supplementary service); userToUserServiceIndicator (used for the User to User Signalling supplementary service); maximumConfereesNumber (used for the Conference Calling supplementary service); and huntGroupAccessSelectionOrder (used for the Mobile Access Hunting supplementary service. It has been replaced by a choice from two components: cliRestrictionOption (used for the CLIR supplementary service); and overrideCategory (used for the CLIP and COLP supplementary services).

The Call Transfer, User to User Signalling, Conference Calling and Mobile Access Hunting supplementary services are not specified for GSM Phase 1 or GSM Phase 2, so data for these services should not be transferred in a dialogue involving a MAP version 1 entity. These cases will therefore not be discussed further.

The type SS-SubscriptionOption is used for a component of SS-Data and for the parameter of the error SS-SubscriptionViolation.

C.4.16 SubscriberData

The type SubscriberData is a sequence of which one component is of type SS-InfoList. Components of SubscriberData are used as a component of the parameter of the operation InsertSubscriberData; the type is also used for a component of the type SentParameter.

C.5 Changes to parameters of errors

This subclause lists in alphabetical order the errors whose parameters have changed.

C.5.1 CUG-Reject

The error CUG-Reject has an optional parameter of type CUG-RejectCause. The error CUG-Reject is used for the operation SendRoutingInfo.

C.5.2 SS-SubscriptionViolation

The error SS-SubscriptionViolation has an optional parameter of type SS-SubscriptionOption. The error SS-SubscriptionViolation is used for the operations ActivateSS, DeactivateSS, EraseSS and RegisterSS.

C.6 Changes to parameters of operations

This subclause lists in alphabetical order the operations whose parameters have changed, and gives a commentary on the effect of the changes on each operation.

C.6.1 InsertSubscriberData

The parameter of the operation InsertSubscriberData is a sequence of which one component is a sequence of components of SubscriberData. The components of SubscriberData which are affected by the changes listed in subclause C.4 are cug-Info and ss-SubscriptionOption.

The CUG supplementary service is not supported by MAP version 1; CUG-Info should therefore not be used as a component of SubscriberData in a dialogue involving a MAP version 1 entity.

The replacement of the perCallBasis (type BOOLEAN) subscription option by the cliRestrictionOption (type ENUMERATED) for the CLIR supplementary service means that full support for the CLIR supplementary service is not possible if either entity involved can support only MAP version 1.

C.6.2 RegisterSS

The forwardedToNumber component of the parameter of the operation RegisterSS had a maximum length of 10 octets in MAP version 1, as it was of the type ISDN-AddressString. In MAP version 2 the maximum length is 20 octets, as the type is AddressString. The maximum length (9 octets) of the ISDN-AddressString type in MAP version 2 may not be adequate to hold the forwardedToNumber, which is not necessarily an E.164 number; the user may enter the number using the digits for international access rather than the "+" key.

C.6.3 SendParameters

The operation SendParameters uses as its parameter a sequence of which one component is of type RequestParameter. The value requestCUG-Info can no longer be used for this component. The SendParameters operation is used only when interworking with a MAP version 1 entity, and MAP version 1 does not support the GSM Phase 2 CUG supplementary service, so the SendParameters operation should in any case not be used to request CUG information.

C.6.4 SendRoutingInfoForSM

The cug-Interlock component of the parameter of the operation SendRoutingInfoForSM has been deleted. Closed User Group does not apply to the short message service.

C.7 Changes to results of operations

This subclause lists in alphabetical order the operations whose results have changed, and gives a commentary on the effect of the changes on each operation.

C.7.1 ActivateSS

The result of the operation ActivateSS is of type SS-Info. Two data types used for components of SS-Info have suffered incompatible changes: CUG-Info and SS-SubscriptionOption.

The ActivateSS operation does not apply to the CUG supplementary service, so the cug-Info component of SS-Info should never be present in the result of the operation ActivateSS.

The ActivateSS operation does not apply to the CLIP, CLIR or COLP supplementary services, for which the type SS-SubscriptionOption is used, so the ss-SubscriptionOption component of SS-Info should never be present in the result of the operation ActivateSS.

C.7.2 DeactivateSS

The result of the operation DeactivateSS is of type SS-Info. Two data types used for components of SS-Info have suffered incompatible changes: CUG-Info and SS-SubscriptionOption.

The DeactivateSS operation does not apply to the CUG supplementary service, so the cug-Info component of SS-Info should never be present in the result of the operation DeactivateSS.

The DeactivateSS operation does not apply to the CLIP, CLIR or COLP supplementary services, for which the type SS-SubscriptionOption is used, so the ss-SubscriptionOption component of SS-Info should never be present in the result of the operation DeactivateSS.

C.7.3 EraseSS

The result of the operation EraseSS is of type SS-Info. Two data types used for components of SS-Info have suffered incompatible changes: CUG-Info and SS-SubscriptionOption.

The EraseSS operation does not apply to the CUG supplementary service, so the cug-Info component of SS-Info should never be present in the result of the operation EraseSS.

The EraseSS operation does not apply to the CLIP, CLIR or COLP supplementary services, for which the type SS-SubscriptionOption is used, so the ss-SubscriptionOption component of SS-Info should never be present in the result of the operation EraseSS.

C.7.4 GetPassword

The result of the operation GetPassword is of type Password. In MAP version 1 this was a choice between a printable string of length 4 to 8 octets or a numeric string of length 4 octets. It is now a numeric string of length 4 octets. The printable string option was never used in MAP version 1, as indicated by a comment in the ASN.1 in the latest phase 1 version of GSM 09.02.

C.7.5 InterrogateSS

The result of the InterrogateSS operation is a CHOICE; one of the components of the CHOICE is a list of basic services to which the supplementary service applies, which is used for the Call Barring supplementary service. In MAP version 1 this list can in principle have up to 70 members, the number of individual basic services. However Call Barring can apply to only 13 basic service groups. In MAP version 2 the length of the list of basic service codes which can be returned in the result of the InterrogateSS operation is reduced to 13 to reflect this.

C.7.6 RegisterSS

The result of the operation RegisterSS is of type SS-Info. Two data types used for components of SS-Info have suffered incompatible changes: CUG-Info and SS-SubscriptionOption.

The RegisterSS operation does not apply to the CUG supplementary service, so the cug-Info component of SS-Info should never be present in the result of the operation RegisterSS.

The RegisterSS operation does not apply to the CLIP, CLIR or COLP supplementary services, for which the type SS-SubscriptionOption is used, so the ss-SubscriptionOption component of SS-Info should never be present in the result of the operation RegisterSS.

C.7.7 SendParameters

The result of the operation SendParameters is of type SentParameterList, which is a sequence of components of type SentParameter. The maximum number of components in the sequence has been reduced from 10 to 6. MAP version 1 could in principle send 10 sets of CUG-Information, but the supplementary service Closed User Group is not defined for GSM Phase 1, and the MAP version 1 signalling protocol will not support Closed User Group as defined for GSM Phase 2, so a MAP version 1 entity should never request parameters for CUG. The maximum number of sent parameters therefore consists of an IMSI and 5 AuthenticationSets - a total of 6.

The type SentParameter is a choice of which one component is of type SubscriberData. The components of SubscriberData which are affected by the changes listed in subclause C.4 are CUG-Info and ss-SubscriptionOption.

The CUG supplementary service is not supported by MAP version 1; CUG-Info should therefore not be used as a component of SubscriberData in a dialogue involving a MAP version 1 entity.

The replacement of the perCallBasis (type BOOLEAN) subscription option by the cliRestrictionOption (type ENUMERATED) for the CLIR supplementary service means that full support for the CLIR supplementary service is not possible if either entity involved can support only MAP version 1.

C.7.8 SendRoutingInfoForSM

The result of the operation SendRoutingInfoForSM is a sequence of which one component was a choice between location information (optionally with an associated LMSI) and forwarding data; the choice of forwarding data has been removed. Call Forwarding does not apply to the short message service.

C.8 Changes to errors of operations

This subclause lists in alphabetical order the operations whose errors have changed, and gives a commentary on the effect of the changes on each operation.

C.8.1 ActivateSS

The definition of the type SS-SubscriptionOption used for the optional parameter of the error SS-SubscriptionViolation has been changed. However the only use defined for the error SS-SubscriptionViolation is when the user attempts to activate or deactivate a Call Barring supplementary service and the subscription option "Control by Service Provider" has been taken. The MAP version 1 protocol does not define this subscription option, so there is no case when the error SS-SubscriptionViolation will be used with the optional parameter.

C.8.2 DeactivateSS

The definition of the type SS-SubscriptionOption used for the optional parameter of the error SS-SubscriptionViolation has been changed. However the only use defined for the error SS-SubscriptionViolation is when the user attempts to activate or deactivate a Call Barring supplementary service and the subscription option "Control by Service Provider" has been taken. The MAP version 1 protocol does not define this subscription option, so there is no case when the error SS-SubscriptionViolation will be used with the optional parameter.

C.8.3 EraseSS

The definition of the type SS-SubscriptionOption used for the optional parameter of the error SS-SubscriptionViolation has been changed. However the only use defined for the error SS-SubscriptionViolation is when the user attempts to activate or deactivate a Call Barring supplementary service and the subscription option "Control by Service Provider" has been taken, so there is no case when the error SS-SubscriptionViolation will be used for the operation EraseSS.

C.8.4 RegisterSS

The definition of the type SS-SubscriptionOption used for the optional parameter of the error SS-SubscriptionViolation has been changed. However the only use defined for the error SS-SubscriptionViolation is when the user attempts to activate or deactivate a Call Barring supplementary service and the subscription option "Control by Service Provider" has been taken, so there is no case when the error SS-SubscriptionViolation will be used for the operation RegisterSS.

C.8.5 SendRoutingInfo

The definition of the type (CUG-RejectCause) used for the optional parameter of the error CUG-Reject has been changed. However the supplementary service Closed User Group is not defined for GSM Phase 1, and the MAP version 1 signalling protocol will not support Closed User Group as defined for GSM Phase 2, so the error CUG-Reject should not be used in a dialogue involving a MAP version 1 entity.

Annex D (informative): Change Request History

CR	SMG#	VERS	NEW_V ERS	SUBJECT
A018	S15		5.0.0	Export of the data type SubscriberData
A021	S15		5.0.0	Operations with empty result in the expanded ASN.1 source
A022	S15		5.0.0	Missing invoke Id in the service descriptions
A020	S15		5.0.0	Correction of MM-Release in SM_Alert_MSC
A019	S16		5.0.0	MSISDN parameter of the MAP_PROVIDE_ROAMING_NUMBER
A024	S16		5.0.0	Modification of the upper limit of Private tags numbers
A016	S16		5.0.0	Use of SCCP class zero
A025	S16		5.0.0	Editorial corrections to the USSD SDLs
A017	S16		5.0.0	Use of the SCCO return option
A035	S17		5.1.0	From "call restrictions" to "call barring"
A036	S17		5.1.0	Subscriber tracing related corrections and alignment to GSM 12.08
A034	S17		5.1.0	Not used value for LMSI in the Cancel_Location service
A038	S18	5.1.0	5.2.0	Introduction of ECT SS code
A038	S18	5.1.0	5.2.0	Introduction of ECT SS code
A043	S19	5.2.0	5.3.0	Timer for PrepareHandover operation
A044	S19	5.2.0	5.3.0	Exporting of data-types used in GSM12 series specifications
A045	S19	5.2.0	5.3.0	Checking of MSISDN in HLR for MT-SMG procedure
A046	S20	5.3.0	5.4.0	Timer values for mobile originated USSD
A049	s21	5.4.0	5.5.0	Indication of MAP_IMSI_ATTACH (phase 2+)
A031	s21	5.4.0	5.5.0	VBS, VGCS
A061	s21	5.4.0	5.5.0	Modification of MAP due to CAMEL ph 1
A054	s21	5.4.0	5.5.0	Transfer of SS codes from HLR to GMSC
A057	s21	5.4.0	5.5.0	HSCSD phase 2+
A052	s21	5.4.0	5.5.0	VGS/VGCS Routing
A051	s21	5.4.0	5.5.0	Transfer of new subscription data for the ASCI eMLPP supplementary service
A056	s21	5.4.0	5.5.0	Changes to MAP for Optimal Routeing
A059	s21	5.4.0	5.5.0	Setting of the 'non-confirmed by HLR'-flag (phase 2+)
A055	s21	5.4.0	5.5.0	Alignment with GSM 03.18
A053	s21	5.4.0	5.5.0	MAP extensibility (chapter 14 ASN.1)
A063	s21	5.4.0	5.5.0	RP-Ack data
A064	s21	5.4.0	5.5.0	Unknown operation
A070	s21	5.4.0	5.5.0	Introduction of new ODB categories for ECT
A067	s21	5.4.0	5.5.0	MAP v3 Fallback
A066	s21	5.4.0	5.5.0	Handling of empty constructors phase 2+
A048	s21	5.4.0	5.5.0	Enhancements to ODB (introduction of omitted categories)
A068	s21	5.4.0	5.5.0	SS handling for the ASCI group call teleservices
A062	s21	5.4.0	5.5.0	Support of SMS Absent subscriber
A085	s22	5.5.0	5.6.0	SCCP addressing for MAP
A081	s22	5.5.0	5.6.0	Deletion of errors in PSI
A083	s22	5.5.0	5.6.0	Clarify handling of result of ReportSMDeliveryStatus
A086	s22	5.5.0	5.6.0	Correction of editing error
A082	s22	5.5.0	5.6.0	Modifications due to SMS inconsistencies with GSM 03.40
A080	s22	5.5.0	5.6.0	Using TC-Result-NL to segment the SRI-ACK
A074	s22	5.5.0	5.6.0	Any Time Interrogation process description in HLR
A072	s22	5.5.0	5.6.0	Handling of Unknown Operation
A073	s22	5.5.0	5.6.0	Editorial clarification of handling of parameters for optional features
(continued)				

(concluded)

CR	SMG#	VERS	NEW_V ERS	SUBJECT
A075	s22	5.5.0	5.6.0	Alignment with CAMEL specs. 03.78 and 09.78
A077	s22	5.5.0	5.6.0	Transmission of gsmSCF-Address to HLR
A078	s22	5.5.0	5.6.0	New Maximum value for Age of Loc. Info
A079	s22	5.5.0	5.6.0	Editorial correction of AC table
A092	s23	5.6.0	5.7.0	ASCI ph1 option pruning (routing or VBS/VGCS calls) [R96]
A104	s23	5.6.0	5.7.0	ASN.1 definition of private extension
A090	s23	5.6.0	5.7.0	Re-introducing of BindingPackage-v1 to GSM 09.02 [R96]
A093	s23	5.6.0	5.7.0	Private extension in v2 SS operations (phase 2+)
A099	s23	5.6.0	5.7.0	Exception handling for Ext-BearerServiceCode/ Ext-TeleserviceCode
A102	s23	5.6.0	5.7.0	SSN reservation for CAP
A100	s23	5.6.0	5.7.0	Handling of PLMN specific SS and BS codes in the GMSC
A106	s24	5.7.0	5.8.0	Corrections
A107	s24	5.7.0	5.8.0	Addition of a context specific TAG for SendRoutingInfoRes
A112	s24	5.7.0	5.8.0	Correction due to GAD
A108	s24	5.7.0	5.8.0	object identifier values for proprietary extensions
A116	s25	5.8.0	5.9.0	Addition of extension marker
A117	s25	5.8.0	5.9.0	Alignment with GSM 03.32, Addition to CR 09.02 CR112
A113	s25	5.8.0	5.9.0	ASCI editorial corrections in section 2.1 R96
A118	s25	5.8.0	5.9.0	Definition of CAP SSN replaced by a reference to GSM 03.03
A110	s25	5.8.0	5.9.0	Alignment of parameter descriptions for ODB with ASN.1 definition
A128	s26	5.9.0	5.10.0	E.214 addressing of the HLR
A119	s26	5.9.0	5.10.0	Modification of CUG-info
A121	s26	5.9.0	5.10.0	Export of Parameters
A126	s26	5.9.0	5.10.0	Queuing of short messages at the VMSC
A137	s27	5.10.0	5.11.0	Extending the applicability of GMSC Address in ProvideRoamingNumber
A145	s27	5.10.0	5.11.0	Removal of redundant information from GSM 09.02
A148	s27	5.10.0	5.11.0	Clarification of limit on length of O-CSI in MAP_RESUME_CALL_HANDLING
A163	s28	5.11.0	5.12.0	Use of SCCP class 1
A166	s28	5.11.0	5.12.0	Corrections and updating of the 09.02
A169	s28	5.11.0	5.12.0	Optionality of sm-RP-UI in MO-ForwardSM-Res
A191	s28	5.11.0	5.12.0	Re-use of security related information
A220	s29	5.12.0	5.13.0	VBS data
A266	CN#06	5.13.0	5.14.0	Clarification on 'Supported CAMEL Phases' in ISD ack
		5.14.0	5.14.1	Version update to 5.14.1 for Publication

History

Document history			
August 1996	Publication of GSM 09.02 version 5.3.0		
November 1996	Unified Approval Procedure (First Edition)	UAP 59:	1996-11-25 to 1997-03-21
May 1997	First Edition		
May 1997	One-step Approval Procedure (Second Edition)	OAP 9742:	1997-05-23 to 1997-10-17
August 1997	One-step Approval Procedure (Third Edition)	OAP 9752	1997-08-29 to 1997-12-26
October 1997	Second Edition		
January 1998	Third Edition		
February 1998	One-step Approval Procedure (Fourth Edition)	OAP 9824	1998-02-13 to 1998-06-12
June 1998	One-step Approval Procedure (Fifth Edition)	OAP 9845:	1998-06-17 to 1998-11-13
July 1998	Fourth Edition		
August 1998	One-step Approval Procedure (Sixth Edition)	OAP 9852:	1998-08-28 to 1998-12-25
November 1998	One-step Approval Procedure (Seventh Edition)	OAP 9912:	1998-11-20 to 1999-03-19
December 1998	Fifth Edition		
January 1999	Sixth Edition		
March 1999	Seventh Edition		
April 1999	One Step Approval Procedure (Eighth Edition)	OAP 9931:	1999-04-02 to 1999-07-30
July 1999	One Step Approval Procedure (Ninth Edition)	OAP 9952:	1999-07-28 to 1999-11-26
August 1999	(Eighth Edition)		
December 1999	Ninth Edition		
December 1999	One Step Approval Procedure (Tenth Edition)	OAP 200013:	1999-12-01 to 2000-03-31
April 2000	Tenth Edition		