ETSI TS 124 080 V6.2.0 (2004-12)

Technical Specification

**Digital cellular telecommunications system (Phase 2+);
Universal Mobile Telecommunications System (UMTS);
Mobile radio Layer 3 supplementary service specification;
Formats and coding
(3GPP TS 24.080 version 6.2.0 Release 6)**

Reference
RTS/TSGN-0424080v620

Keywords
GSM, UMTS

ETSI

650 Route des Lucioles
F-06921 Sophia Antipolis Cedex - FRANCE

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

Siret N° 348 623 562 00017 - NAF 742 C
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° 7803/88

Important notice

Individual copies of the present document can be downloaded from:
http://www.etsi.org

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

Users of the present document should be aware that the document may be subject to revision or change of status. Information on the current status of this and other ETSI documents is available at
http://portal.etsi.org/tb/status/status.asp

If you find errors in the present document, please send your comment to one of the following services:
http://portal.etsi.org/chaircor/ETSI_support.asp

Copyright Notification

Intellectual Property Rights

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

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

Foreword

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

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

The cross reference between GSM, UMTS, 3GPP and ETSI identities can be found under http://webapp.etsi.org/key/queryform.asp .

Contents

Foreword

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

The present document defines the coding of information necessary for support of supplementary service operation on the mobile radio interface layer 3 within the 3GPP system.1

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

Version x.y.z

where:

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

1 Scope

The present document contains the coding of information necessary for support of supplementary service operation on the mobile radio interface layer 3.

Clause 2 gives the functional definitions and contents of messages for call independent supplementary service operations. Messages necessary for support of call related supplementary service operations are defined in TS 24.008.

Clause 3 gives the general format and coding for messages used for call independent supplementary service and the format and coding of information elements used for both call related and call independent supplementary service operations.

Clause 4 gives the specification of the call related and call independent supplementary service operations.

1.1 References

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

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

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

[2] 3GPP TS 22.024: "Description of Charge Advice Information (CAI)".

[3] 3GPP TS 44.006: "Mobile Station - Base Station System (MS - BSS) interface Data Link (DL) layer specification".

[4] 3GPP TS 24.007: "Mobile radio interface signalling layer 3; General aspects".

[5] 3GPP TS 24.008: "Mobile radio interface layer 3 specification".

[6] 3GPP TS 24.010: "Mobile radio interface layer 3; Supplementary services specification; General aspects".

[7] 3GPP TS 24.080: "Mobile radio interface layer 3 supplementary services specification; Formats and coding".

[8] 3GPP TS 24.090: "Unstructured supplementary services operation - Stage 3".

[9] 3GPP TS 29.002: "Mobile Application Part (MAP) specification".

[10] 3GPPTS 29.011: "Signalling interworking for supplementary services".

[11] ITU-T Recommendation X.680: "Information technology – Abstract Syntax Notation One (ASN.1): Specification of basic notation".

[11b] ITU-T Recommendation X.681: "Information technology – Abstract Syntax Notation One (ASN.1): Information object specification".

[12] ITU-T Recommendation X.690: "Information technology – ASN.1 encoding rules: Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER)".

[13] ITU-T Recommendation X.880: "Data networks and open system communication - Open System Interconnection - Service definitions - Remote operations: Concepts, model and notation".

1.2 Abbreviations

Abbreviations used in the present document are listed in TR 21.905.

2 Message functional definitions and contents

2.1 General

This clause defines the structure of the messages of the layer 3 protocol defined in TS 24.080. These messages are standard L3 messages as defined in TS 24.007.

Each definition includes:

a) a brief description of the message;

b) a table listing the information elements in the order of their appearance in the message. In a sequence of consecutive IEs with half octet length, the first IE occupies bits 1 to 4 of octet N, the second bits 5 to 8 of octet N, the third bits 1 to 4 of octet N+1 etc..

   For each IE the table indicates:

   1) the information element identifier, in hexadecimal notation, if the IE has format T, TV or TLV. If the IEI has half octet length, it is specified by a notation representing the IEI as a hexadecimal digit followed by a "-" (example: B-);

   2) the name of the IE (which gives an idea of the semantics of the element), which is used in this and other specifications as a reference to the IE within the message;

   3) the name of the type of the IE (which indicates the coding of the value part of the IE), and a reference to a description of the value part of the IE;

   4) the presence requirement indication (M, C or O) for the IE, as defined in TS 24.007;

   5) the format of the IE (T, V, TV, LV, TLV) as defined in TS 24.007;

   6) the length of the IE (or permissible range of lengths), in octets, in the message, where "?" means that the maximum length of the IE is only constrained by the link layer protocol, and in the case of the facility IE by possible further considerations specified in TS 24.010. This indication is non-normative.

c) Subclauses specifying conditions for IEs with presence requirement C or O in the relevant message. Together with other conditions specified in TS 24.080, TS 24.010 or TS 24.08x and 24.09x-series this defines when the IE shall be included or not, what non-presence of such IEs means, and (for IEs with presence requirement C) the static conditions for presence and/or non-presence of the IEs (see TS 24.007).

2.2 Messages for supplementary services control

Table 2.1 summarizes the messages for call independent supplementary services control (see TS 24.010 for a detailed description of call independent supplementary service messages).

Table 2.1: Messages for call independent supplementary service control

Messages for supplementary service control	Reference
FACILITY	2.3
REGISTER	2.4
RELEASE COMPLETE	2.5

2.3 Facility

This message is sent by the mobile station or the network to request or acknowledge a supplementary service. It is used when information is to be conveyed and the transaction already exists, but is not to be released. The supplementary service to be invoked, and its associated parameters, are specified in the Facility information element (see table 2.2).

Table 2.2: FACILITY message content

IEI	Information element	Type / Reference	Presence	Format	Length
	Supplementary service protocol discriminator	Protocol discriminator 3.2	M	V	1/2
	Transaction identifier	Transaction identifier 3.3	M	V	1/2
	Facility message type	Message type 3.4	M	V	1
	Facility	Facility 3.6	M	LV	2-?

2.4 Register

2.4.1 Register (network to MS direction)

This message is sent by the network to the mobile station to assign a new transaction identifier for call independent supplementary service control and to request or acknowledge a supplementary service (see table 2.3).

Table 2.3: REGISTER message content (network to MS direction)

IEI	Information element	Type / Reference	Presence	Format	Length
	Supplementary service protocol discriminator	Protocol discriminator 3.2	M	V	1/2
	Transaction identifier	Transaction identifier 3.3	M	V	1/2
	Register message type	Message type 3.4	M	V	1
1C	Facility	Facility 3.6	M	TLV	2-?

2.4.2 Register (MS to network direction)

This message is sent by the mobile station to the network to assign a new transaction identifier for call independent supplementary service control and to request or acknowledge a supplementary service (see table 2.4).

Table 2.4: REGISTER message content (MS to network direction)

IEI	Information element	Type / Reference	Presence	Format	Length
	Supplementary service protocol discriminator	Protocol discriminator 3.2	M	V	1/2
	Transaction identifier	Transaction identifier 3.3	M	V	1/2
	Register message type	Message type 3.4	M	V	1
1C	Facility	Facility 3.6	M	TLV	2-?
7F	SS version	SS version indicator 3.7.2	O	TLV	3

2.4.2.1 SS version

This information element shall be included if the supplementary service operation being invoked is implemented according to the phase 2 or higher protocol version.

2.5 Release complete

This message is sent by the mobile station or the network to release a transaction used for call independent supplementary service control. It may also request or acknowledge a supplementary service (see table 2.5).

Table 2.5: RELEASE COMPLETE message content

IEI	Information element	Type / Reference	Presence	Format	Length
	Supplementary service protocol discriminator	Protocol discriminator 3.2	M	V	1/2
	Transaction identifier	Transaction identifier 3.3	M	V	1/2
	Release Complete message type	Message type 3.4	M	V	1
08	Cause	Cause TS 24.008	O	TLV	4-32
1C	Facility	Facility 3.6	O	TLV	2-?

2.5.1 Cause

This information element shall be included when the functional handling of the Cause IE is specified in the service description or TS 29.011. If the functional handling of the Cause IE is not specified, the receiving entity may ignore the IE.

2.5.2 Facility

This information element shall be included as required by the service description and the procedures defined in TS 24.010.

3 General message format and information elements coding

The figures and text in this clause describe message contents. Within each octet, the bit designated "bit 1" is transmitted first, followed by bits 2, 3, 4, etc. Similarly, the octet shown at the top of each figure is sent first.

3.1 Overview

Within the layer 3 protocol defined in TS 24.080, every message is a standard L3 message as defined in TS 24.007. This means that the message consists of the following parts:

a) protocol discriminator;

b) transaction identifier;

c) message type;

d) other information elements, as required.

Unless specified otherwise, a particular information element may be present only once in a given message.

When a field extends over more than one octet, the order of bit values progressively decreases as the octet number increases. The least significant bit of the field is represented by the lowest numbered bit of the highest numbered octet of the field.

3.2 Protocol discriminator

The Protocol Discriminator (PD) and its use are defined in TS 24.007. TS 24.080 defines the protocols relating to the PD values:

1 0 1 1 supplementary services (call independent).

3.3 Transaction identifier

For general rules, format and coding of transaction identifier values, see TS 24.008.

3.4 Message type

The message type IE and its use are defined in TS 24.007. Table 3.1 defines the value part of the message type IE used in the supplementary service protocol.

Table 3.1: Message types

8	7	6	5	4	3	2	1	Message types
x	x	1	0	.	.	.	.	Clearing messages:
				1	0	1	0	- RELEASE COMPLETE
x	x	1	1	.	.	.	.	Miscellaneous message group:
				1	0	1	0	- FACILITY
				1	0	1	1	- REGISTER

For messages transmitted via CS domain the following applies:

- When the radio connection started with a core network node of earlier than R99, bit 8 shall be set to 0 and bit 7 is reserved for the send sequence number in messages sent from the mobile station. In messages sent from the network, bits 7 and 8 are coded with a "0". See 3GPP TS 24.007 [4].
- When the radio connection started with a core network node of R'99 or later, bits 7 and 8 are reserved for the send sequence number in messages sent from the mobile station. In messages sent from the network, bits 7 and 8 are coded with a "0". See 3GPP TS 24.007 [4].

For messages transmitted via PS domain the following applies:

- No sequence number shall be used. Bits 7 and 8 shall be coded with a "0". See 3GPP TS 24.007 [4]. If a sequence number is received on the network side it shall be ignored.

3.5 Other information elements

These information elements are coded according to the general coding rules as defined in TS 24.008.

Table 3.2 contains the code-points allocated to the information elements used in messages defined in this specification. All IEs are defined in TS 24.008, but the content of the Facility and SS version indicator IEs are defined within this specification.

Table 3.2: Information elements specific to call independent supplementary service control

8 7 6 5 4 3 2 1	**Reference (IE content)**

0	Type 3 and 4 information elements	
0 0 0 1 0 0 0	Cause	TS 24.008
0 0 1 1 1 0 0	Facility	3.6
1 1 1 1 1 1 1	SS version indicator	3.8.2

3.6 Facility information element

The purpose of the Facility information element is to indicate the invocation and operation of supplementary services, identified by the corresponding operation code within the Facility information element.

The Facility information element is coded as shown in figure 3.1 and tables 3.3 to 3.17.

The Facility is a type 4 information element with no upper length limit except that given by the maximum number of octets in a L3 message, see 3GPP TS 44.006.

8	7	6	5	4	3	2	1	
0	0	0	1	1	1	0	0	octet 1
Facility IEI								
Length of Facility contents								octet 2
Component(s) (note)								octet 3 etc.
NOTE: One or more components may be included depending on specific service requirements.								

Figure 3.1: Facility information element

3.6.1 Component (octet 3 etc.)

This subclause provides the formats and encoding of components in the Facility information element. Formats and encoding methods make use of and is a subset of ITU-T Rec. Q.773 (Transaction Capabilities formats and Encoding) and T/S 43/BB. The used part of ITU-T Rec. Q.773 respectively T/S 43/BB is almost the same as the Component Portion of TC messages. The only difference is that returnResultNotLast is not used.

This subclause is further based on:

- ITU-T Rec. X.680 (Abstract Syntax Notation One (ASN.1): Specification of Basic Notation);
- ITU-T Rec. X.690 (Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER));

and is consistent with these ITU-T Recommendations.

The parameters in tables 3.3 to 3.6 may be one of the following:

- a Sequence of Parameters;
- a Set of Parameters;
- a specific Parameter with its own tag (i.e. not part of a Sequence or Set);
- nothing at all (i.e. absent).

NOTE: Concerning the general rules for encoding (structure of encoding, identifier octets, length octets, etc.) see ITU-T Rec. X.690. For these general rules the same exceptions apply as stated in TS 29.002. This holds also for tables 3.3 to 3.6.

Table 3.3: Invoke component

Invoke component	Reference	Mandatory indication
Component type tag Component length	3.6.2 X.690	M
Invoke ID tag Invoke ID length Invoke ID	3.6.3 X.690 3.6.3	M
Linked ID tag Linked ID length Linked ID	3.6.3 X.690 3.6.3	O
Operation Code tag Operation Code length Operation Code	3.6.4 X.690 3.6.4	M
Parameters	4	O

Table 3.4: Return Result component

Return Result component	Reference	Mandatory indication
Component type tag Component length	3.6.2 X.690	M
Invoke ID tag Invoke ID length Invoke ID	3.6.3 X.690 3.6.3	M
Sequence tag Sequence length	3.6.5 X.690	O (note)
Operation Code tag Operation Code length Operation Code	3.6.4 X.690 3.6.4	O (note)
Parameters	4	O (note)
NOTE: Omitted if the Return Result component does not include any parameters.		

Table 3.5: Return Error component

Return Error component	Reference	Mandatory indication
Component type tag Component length	3.6.2 X.690	M
Invoke ID tag Invoke ID length Invoke ID	3.6.3 X.690 3.6.3	M
Error Code tag Error Code length Error Code	3.6.6 X.690 3.6.6	M
Parameters	4	O

Table 3.6: Reject component

Reject component	Reference	Mandatory indication
Component type tag Component length	3.6.2 X.690	M
Invoke ID tag (note) Invoke ID length Invoke ID	3.6.3 X.690 3.6.3	M
Problem Code tag Problem Code length Problem Code	3.6.7 X.690 3.6.7	M
NOTE: If the Invoke ID is not available, Universal Null (table 3.9) with length = 0 shall be used.		

3.6.2 Component type tag

The Component type tag is coded context-specific, constructor as indicated in table 3.7.

Table 3.7: Coding of Component type tag

Component type tag	8 7 6 5 4 3 2 1
Invoke	1 0 1 0 0 0 0 1
Return Result	1 0 1 0 0 0 1 0
Return Error	1 0 1 0 0 0 1 1
Reject	1 0 1 0 0 1 0 0

3.6.3 Component ID tag

The term Component ID refers to the Invoke ID or the Linked ID. The Component ID tag is coded as shown in table 3.8.

Table 3.8: Coding of Component ID tag

Component ID tag	8 7 6 5 4 3 2 1
Invoke ID	0 0 0 0 0 0 1 0
Linked ID (note)	1 0 0 0 0 0 0 0
NOTE: This tag differs from the Invoke ID tag, which is coded as a Universal INTEGER, in order to distinguish it from the following tag (Operation Code) which is also coded as a Universal INTEGER.	

The length of a Component ID is 1 octet.

An Invoke Component has one or two Component IDs: an Invoke ID, and if it is desired to associate the Invoke with a previous Invoke, then the Linked ID is provided in addition to the Invoke ID.

Return Result and Return Error Components have one Component ID, called an Invoke ID which is the reflection of the Invoke ID of the Invoke Component to which they are responding.

The Reject Component uses as its Invoke ID, the Invoke ID in the Component being rejected. If this ID is unavailable (e.g. due to mutilation of the message not detected by lower layers), then the Invoke ID tag is replaced with a universal NULL tag as shown in table 3.9. Universal NULL has always length = 0.

Any kind of component, except a reject component, may be rejected.

Table 3.9: Coding of NULL tag

	8 7 6 5 4 3 2 1
NULL tag	0 0 0 0 0 1 0 1

If an Invoke containing both Invoke and Linked IDs is being rejected, only the Invoke ID is used in the Reject Component.

3.6.4 Operation Code

Each Operation is assigned an Operation Code to identify it. An Operation Code follows an Operation Code tag and Operation Code length. The Operation Code tag is coded as shown in table 3.10.

Table 3.10: Coding of Operation Code tag

	8 7 6 5 4 3 2 1
Operation Code tag	0 0 0 0 0 0 1 0

The Operation Codes for the different Operations are defined in subclause 4.5.

3.6.5 Sequence and Set tags

When there is more than one parameter in a Component (applicable to all Component types), they follow the Sequence or Set tag, which are coded universal, constructor as shown in table 3.11.

Table 3.11: Coding of Sequence and set tags

Sequence and set tags	8 7 6 5 4 3 2 1
Sequence tag	0 0 1 1 0 0 0 0
Set tag	0 0 1 1 0 0 0 1

3.6.6 Error Code

Each Error is assigned a value (Error Code) to identify it.

An Error Code follows an Error Code tag and Error Code length. The Error Code tag is coded as shown in table 3.12.

Table 3.12: Coding of Error Code tag

	8 7 6 5 4 3 2 1
Error Code tag	0 0 0 0 0 0 1 0

The Error Codes for the different Errors are defined in subclause 4.5.

3.6.7 Problem Code

The Problem Code consists of one of the four elements: General Problem, Invoke Problem, Return Result Problem or Return Error Problem. The tags for these elements are coded as shown in table 3.13.

Table 3.13: Coding of Problem tags

Problem tags	8 7 6 5 4 3 2 1
General Problem tag	1 0 0 0 0 0 0 0
Invoke Problem tag	1 0 0 0 0 0 0 1
Return Result Problem tag	1 0 0 0 0 0 1 0
Return Error Problem tag	1 0 0 0 0 0 1 1

The Problem Codes for the different Problems are shown in tables 3.14 to 3.17.

Table 3.14: Coding of General Problem Codes

General Problem Codes	8 7 6 5 4 3 2 1
Unrecognized Component	0 0 0 0 0 0 0 0
Mistyped Component	0 0 0 0 0 0 0 1
Badly Structured Component	0 0 0 0 0 0 1 0

Table 3.15: Coding of Invoke Problem Codes

Invoke Problem Codes	8 7 6 5 4 3 2 1
Duplicate Invoke ID	0 0 0 0 0 0 0 0
Unrecognized Operation	0 0 0 0 0 0 0 1
Mistyped Parameter	0 0 0 0 0 0 1 0
Resource Limitation	0 0 0 0 0 0 1 1
Initiating Release	0 0 0 0 0 1 0 0
Unrecognized Linked ID	0 0 0 0 0 1 0 1
Linked Response Unexpected	0 0 0 0 0 1 1 0
Unexpected Linked Operation	0 0 0 0 0 1 1 1

Table 3.16: Coding of Return Result Problem Codes

Return Result Problem Codes	8 7 6 5 4 3 2 1
Unrecognized Invoke ID	0 0 0 0 0 0 0 0
Return Result Unexpected	0 0 0 0 0 0 0 1
Mistyped Parameter	0 0 0 0 0 0 1 0

Table 3.17: Coding of Return Error Problem Codes

Return Error Problem Codes	8 7 6 5 4 3 2 1
Unrecognized Invoke ID	0 0 0 0 0 0 0 0
Return Error Unexpected	0 0 0 0 0 0 0 1
Unrecognized Error	0 0 0 0 0 0 1 0
Unexpected Error	0 0 0 0 0 0 1 1
Mistyped Parameter	0 0 0 0 0 1 0 0

3.7 Version handling for supplementary services

3.7.1 Supplementary service screening indicator

The purpose of the supplementary service screening indicator is to allow the network to asses the capabilities of the MS in advance of a network initiated SS activity. The SS screening indicator is sent in the mobile station classmark 2 as defined in TS 24.008. The handling of the SS screening indicator is described in TS 24.010.

8	7	6 5	4 3 2 1
(note)	(note)	SS screening indicator	(note)

NOTE: Values not relevant to supplementary services.

Figure 3.2: Coding of SS screening indicator in mobile station classmark 2

Table 3.18: Coding of SS screening indicator in mobile station classmark 2

SS screening indicator in mobile station classmark 2	6	5
default value of phase 1	0	0
capability of handling of ellipsis notation and phase 2 error handling (note 1)	0	1
for future use (note 2)	1	0
for future use (note 2)	1	1

NOTE 1: Ellipsis notation is described in TS 24.010 and TS 29.002. SS Error handling is described in TS 24.010.

NOTE 2: The network shall interpret these values the same as "01".

3.7.2 Supplementary service version indicator

The purpose of the supplementary service version indicator is to allow the network to select the correct version of a protocol for a specific supplementary service. The SS version indicator is included in messages as defined in TS 24.008 and TS 24.080. The coding described in table 3.19 refers to the first octet received in the SS version indicator. Any other octets received shall be ignored. The handling of the SS version indicator is described in TS 24.010.

Table 3.19: Coding of SS version indicator

SS version indicator	8 7 6 5 4 3 2 1
phase 2 service, ellipsis notation, and phase 2 error handling is supported (note 1)	0 0 0 0 0 0 0 0
SS-Protocol version 3 is supported, and phase 2 error handling is supported (note 1) all other values are for future use (note 2)	0 0 0 0 0 0 0 1
NOTE 1: Ellipsis notation is described in TS 24.010 and TS 29.002. SS Error handling is described in TS 24.010. NOTE 2: The network shall interpret all higher values of the SS version indicator the same as "00000001".	

4 Supplementary services operation specifications

4.1 General

This clause specifies the abstract syntax for the Supplementary Service protocol using the Abstract Syntax Notation One (ASN.1), defined in ITU-T Rec. X.680 and X.681.

The mapping of OPERATION and ERROR to components is defined in clause 3 of this specification.

The encoding rules which are applicable to the defined abstract syntax are the Basic Encoding Rules for Abstract Syntax Notation One, defined in ITU-T Rec. X.690 with the same exceptions as stated in TS 29.002. For each Supplementary Service parameter which has to be transferred by a Supplementary Service 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, the first letter of the first word is lower-case and the first letter of the following words are capitalized (e.g. "bearer service" is mapped to "bearerService"). In addition some words may be abbreviated as follows:

- ms mobile subscriber;
- ss supplementary services;
- cug closed user group.

The ASN.1 data type which follows the words "ARGUMENT", "PARAMETER" or "RESULT" (for information objects of class "OPERATION" and "ERROR") is always optional from a syntactic point of view. However, except specific mention, it has to be considered as mandatory from a semantic point of view. When in an invoke component, a mandatory element is missing in any component or inner data structure, a reject component is returned with the problem code "Mistyped Parameter". 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; the associated type of error is "DataMissing".

In case an element is defined as mandatory in the protocol description (TS 24.080 including imports from TS 29.002), but is not present according to the service description (stage 1 to stage 3), the ASN.1 protocol description takes precedence over the diagrams in the TS 24.08x and 24.09x-series of technical specifications.

When possible operations and errors are imported from TS 29.002 thereby making the MSC transparent to most of the messages sent to or from the MS.

Timer values for operations which require timers are shown as ASN.1 comments.

The extension marker "…" shall be used in the same way as described in TS 29.002 and shall be supported on the radio interface by the MS and the network for all operations defined in this specification including those imported from TS 29.002.

4.2 Operation types

Table 4.1 summarizes the operations defined for supplementary services in this specification and shows which of these operations are call related and call independent. The terms "call related" and "call independent" are defined in TS 24.010.

Table 4.1: Relevance of supplementary service operations

Operation name	Call related SS	Call independent SS
registerSS	-	+
eraseSS	-	+
activateSS	-	+
deactivateSS	-	+
interrogateSS	-	+
registerPassword	-	+
getPassword	-	+
processUnstructuredSS-Data	+	+
forwardCheckSS-Indication	-	+
processUnstructuredSS-Request	-	+
unstructuredSS-Request	-	+
unstructuredSS-Notify	-	+
forwardChargeAdvice	+	-
notifySS	+	-
forwardCUG-Info	+	-
buildMPTY	+	-
holdMPTY	+	-
retrieveMPTY	+	-
splitMPTY	+	-
explicitCT	+	-
accessRegisterCCEntry	+	-
eraseCCEntry	-	+
callDeflection	+	-
userUserService	+	-
lcs-LocationNotification	-	+
lcs-MOLR	-	+
lcs-AreaEventRequest	-	+
lcs-AreaEventReport	-	+
Lcs-AreaEventCancellation	-	+
NOTE: The processUnstructuredSS-Data operation may be used call related by a GSM Phase 1 MS.		

The following ASN.1 module defines operations by allocating them a local value. For the involved operations the same local values as in MAP are allocated.

```
SS-Operations {
   itu-t identified-organization (4) etsi (0) mobileDomain (0) gsm-Access (2) modules (3)
   ss-Operations (0) version9 (9)}

DEFINITIONS ::=

BEGIN

EXPORTS

-- exports operations

-- operations defined in this specification
processUnstructuredSS-Data, notifySS, forwardChargeAdvice, forwardCUG-Info, buildMPTY, holdMPTY,
retrieveMPTY, splitMPTY, explicitCT, accessRegisterCCEntry, callDeflection, userUserService,
lcs-LocationNotification, lcs-MOLR, lcs-AreaEventRequest, lcs-AreaEventReport, lcs-
AreaEventCancellation;

IMPORTS

OPERATION FROM
Remote-Operations-Information-Objects {
    joint-iso-itu-t remote-operations(4)
  informationObjects(5) version1(0)}

-- The MAP operations:
-- registerSS, eraseSS, activateSS, deactivateSS, interrogateSS, registerPassword,
-- getPassword, processUnstructuredSS-Request, unstructuredSS-Request, unstructuredSS-Notify
-- forwardCheckSS-Indication
-- are imported from MAP-Operations in SS-Protocol module.

-- imports SS-data types
NotifySS-Arg,
ForwardChargeAdviceArg,
ForwardCUG-InfoArg,
```

```
SS-UserData,
AccessRegisterCCEntryArg,
CallDeflectionArg,
UserUserServiceArg,
LocationNotificationArg,
LocationNotificationRes,
LCS-MOLRArg,
LCS-MOLRRes,
LCS-AreaEventRequestArg,
LCS-AreaEventReportArg,
LCS-AreaEventCancellationArg

FROM SS-DataTypes {
   itu-t identified-organization (4) etsi (0) mobileDomain (0) gsm-Access (2) modules (3)
   ss-DataTypes (2) version9 (9)}

-- imports MAP-SS-data types
RegisterCC-EntryRes
FROM MAP-SS-DataTypes {
   itu-t identified-organization (4) etsi (0) mobileDomain (0)
   gsm-Network (1) modules (3) map-SS-DataTypes (14) version9 (9)}

-- imports MAP-errors
illegalSS-Operation, ss-ErrorStatus, ss-NotAvailable, ss-SubscriptionViolation,
ss-Incompatibility, systemFailure, facilityNotSupported, callBarred, unexpectedDataValue,
shortTermDenial, longTermDenial, dataMissing, forwardingViolation, forwardingFailed,
positionMethodFailure
FROM MAP-Errors {
   itu-t identified-organization (4) etsi (0) mobileDomain (0) gsm-Network (1) modules (3)
   map-Errors (10) version9 (9)}

-- imports SS-Errors
resourcesNotAvailable, maxNumberOfMPTY-ParticipantsExceeded, deflectionToServedSubscriber,
invalidDeflectedToNumber, specialServiceCode, rejectedByUser, rejectedByNetwork
FROM SS-Errors {
   itu-t identified-organization (4) etsi (0) mobileDomain (0) gsm-Access (2) modules (3)
   ss-Errors (1) version9 (9)}
;

-- operations definition

processUnstructuredSS-Data  OPERATION ::= { -- Timer T(PUSSD)= 15s to 30s
    ARGUMENT    SS-UserData
    RESULT      SS-UserData
                -- optional
    ERRORS      {
                systemFailure |
                unexpectedDataValue}
    CODE        local:19 }

notifySS    OPERATION ::= {
    ARGUMENT    NotifySS-Arg
    CODE        local:16 }

forwardChargeAdvice     OPERATION ::= { -- Timer T(AoC)= 1s to 40s
    ARGUMENT    ForwardChargeAdviceArg
    RETURN RESULT TRUE
    CODE        local:125 }

forwardCUG-Info     OPERATION ::= {
    ARGUMENT    ForwardCUG-InfoArg
    CODE        local:120 }

buildMPTY   OPERATION ::= { -- Timer T(BuildMPTY)= 5s to 30s
    RETURN RESULT TRUE
    ERRORS  {
            illegalSS-Operation |
            ss-ErrorStatus |
            ss-NotAvailable |
            ss-Incompatibility |
            systemFailure |
            resourcesNotAvailable |
            maxNumberOfMPTY-ParticipantsExceeded}
    CODE    local:124 }

holdMPTY    OPERATION ::= { -- Timer T(HoldMPTY)= 5s to 30s
    RETURN RESULT TRUE
    ERRORS  {
```

```
            illegalSS-Operation |
            ss-ErrorStatus |
            ss-Incompatibility |
            facilityNotSupported |
            systemFailure}
    CODE    local:123 }

retrieveMPTY    OPERATION ::= { -- Timer T(RetrieveMPTY)= 5s to 30s
    RETURN RESULT TRUE
    ERRORS  {
            illegalSS-Operation |
            ss-ErrorStatus |
            ss-Incompatibility |
            facilityNotSupported |
            systemFailure}
    CODE    local:122 }

splitMPTY   OPERATION ::= { -- Timer T(SplitMPTY)= 5s to 30s
    RETURN RESULT TRUE
    ERRORS  {
            illegalSS-Operation |
            ss-ErrorStatus |
            ss-Incompatibility |
            facilityNotSupported |
            systemFailure}
    CODE    local:121 }

explicitCT  OPERATION ::= { -- Timer T(ECT)= 5s to 15s
    RETURN RESULT TRUE
    ERRORS  {
            illegalSS-Operation |
            ss-ErrorStatus |
            ss-NotAvailable |
            ss-Incompatibility |
            facilityNotSupported |
            systemFailure |
            resourcesNotAvailable |
            callBarred}
    CODE    local:126 }

accessRegisterCCEntry    OPERATION ::= { -- Timer T(AccRegCCEntry)= 30s
    ARGUMENT    AccessRegisterCCEntryArg
    RESULT      RegisterCC-EntryRes
    ERRORS      {
                systemFailure |
                dataMissing |
                unexpectedDataValue |
                callBarred |
                illegalSS-Operation |
                ss-ErrorStatus |
                ss-Incompatibility |
                shortTermDenial |
                longTermDenial |
                facilityNotSupported}
    CODE        local:119 }

-- the timer value is defined by T308, see also in TS 24.008 for definition of timer T308

callDeflection  OPERATION ::= { -- Timer T(CD)= 30s
    ARGUMENT    CallDeflectionArg
    RETURN RESULT TRUE
    ERRORS      {
                illegalSS-Operation |
                ss-ErrorStatus |
                ss-NotAvailable |
                ss-Incompatibility |
                facilityNotSupported |
                systemFailure |
                resourcesNotAvailable |
                forwardingViolation |
                callBarred |
                deflectionToServedSubscriber |
                invalidDeflectedToNumber |
                specialServiceCode |
                forwardingFailed}
    CODE        local:117 }

-- the timer value is defined by T305, see also in TS 24.008 for definition of timer T305
```

```
-- extensionContainer shall not be used with this operation

userUserService     OPERATION ::= { -- Timer T(UUS3)= 10s
    ARGUMENT    UserUserServiceArg
    RETURN RESULT TRUE
    ERRORS      {
                illegalSS-Operation |
                ss-ErrorStatus |
                ss-NotAvailable |
                ss-Incompatibility |
                facilityNotSupported |
                systemFailure |
                resourcesNotAvailable |
                rejectedByNetwork |
                rejectedByUser}
    CODE        local:118 }

-- The timer value for UUS3 is 10s; it is applicable only if UUS3 is activated by FACILITY
-- message. If UUS service (UUS1, UUS2 or UUS3) is activated by SETUP message, no timers are
-- needed. In those cases Return Result or Return Error must be received within certain call
-- control messages, see 3GPP TS 24.087.
-- extensionContainer shall not be used with this operation.


lcs-LocationNotification    OPERATION ::= { -- Timer T(LCSN)= 10s to 20s
    ARGUMENT    LocationNotificationArg
    RESULT      LocationNotificationRes
    ERRORS      {
                systemFailure |
                unexpectedDataValue}
    CODE        local:116 }


lcs-MOLR    OPERATION ::= { -- Timer T(LCSL)= 10s to 300s
    ARGUMENT    LCS-MOLRArg
    RESULT      LCS-MOLRRes
    ERRORS      {
                systemFailure |
                unexpectedDataValue |
                dataMissing |
                facilityNotSupported |
                ss-SubscriptionViolation |
                positionMethodFailure}
    CODE        local:115 }


lcs-AreaEventRequest    OPERATION ::= { -- Timer T(LCSN)= 10s to 20s
    ARGUMENT    LCS-AreaEventRequestArg
    RETURN RESULT TRUE
    ERRORS      {
                systemFailure |
                facilityNotSupported |
                unexpectedDataValue}
    CODE        local:114 }


lcs-AreaEventReport OPERATION ::= { -- Timer T(LCSL)= 10s to 300s
    ARGUMENT    LCS-AreaEventReportArg
    RETURN RESULT TRUE
    ERRORS      {
                systemFailure |
                unexpectedDataValue |
                facilityNotSupported}
    CODE        local:113 }


lcs-AreaEventCancellation   OPERATION ::= { -- Timer T(LCSN)= 10s to 20s
    ARGUMENT    LCS-AreaEventCancellationArg
    RETURN RESULT TRUE
    ERRORS      {
                systemFailure |
                facilityNotSupported |
                unexpectedDataValue}
    CODE        local:112 }


END
```

4.2.1 Void

4.2.2 Operations description

For each operation this subclause provides a brief prose description.

4.2.2.1 registerSS (MS --> network)

This operation is invoked by an MS to register data related to a supplementary service in the network. When no BasicService parameter is provided, the registration applies to all provisioned and applicable basic services.

4.2.2.2 eraseSS (MS --> network)

This operation is invoked by an MS to erase data related to a supplementary service in the network. When no BasicService parameter is provided, the erasure applies to all provisioned and applicable basic services.

4.2.2.3 activateSS (MS --> network)

This operation is invoked by an MS to request the network for a supplementary service activation. When no BasicService parameter is provided, the activation applies to all provisioned and applicable basic services.

4.2.2.4 deactivateSS (MS --> network)

This operation is invoked by an MS to request the network for a supplementary service deactivation. When no BasicService parameter is provided, the deactivation applies to all provisioned and applicable basic services.

4.2.2.5 interrogateSS (MS --> network)

This operation is invoked by an MS to request the network for a supplementary service interrogation. When no BasicService parameter is provided, the interrogation applies to all provisioned and applicable basic services.

4.2.2.6 notifySS (network --> MS)

This operation is invoked by the network to forward a supplementary service notification towards a mobile subscriber.

4.2.2.7 registerPassword (MS --> network)

This operation is invoked by an MS to register a new password related to the management by the subscriber himself of subscription data in the HLR. The operation "Register password" will be successful if the subscriber can provide the old password, the new password and the new password again as results of 3 subsequent operations "Get password".

4.2.2.8 getPassword (network --> MS)

This operation is invoked by the network to request a password from the mobile subscriber. It may be used to allow the registration of a new password or the management of subscription data by the subscriber himself (e.g. modification of call barring activation status).

4.2.2.9 processUnstructuredSS-Data (MS --> network)

This operation is invoked by an MS to relay unstructured information in order to allow end to end SS operation between the MS and the network following specific rules (e.g. embedding of keypad commands). The operation is used in order to provide backward compatibility (see TS 24.090).

4.2.2.10 processUnstructuredSS-Request (MS --> network)

This operation is invoked by an MS to start an unstructured supplementary service data application in the network.

4.2.2.11 unstructuredSS-Request (network --> MS)

This operation is invoked by the network to request unstructured information from the MS in order to perform an unstructured supplementary service data application.

4.2.2.12 unstructuredSS-Notify (network --> MS)

This operation is invoked by the network to give an unstructured supplementary service notification to the mobile user.

4.2.2.13 forwardCheckSSIndication (network --> MS)

This operation is invoked by the network to indicate to the mobile subscriber that the status of supplementary services may not be correct in the network. The procedures for initiating ForwardCheckSSIndication are specified in TS 29.002.

4.2.2.14 forwardChargeAdvice (network --> MS)

This operation is invoked by the network to forward Advice of Charge information to the mobile subscriber.

4.2.2.15 buildMPTY (MS --> network)

This operation is invoked by an MS to request the network to connect calls in a multi party call.

4.2.2.16 holdMPTY (MS --> network)

This operation is invoked by an MS to put the MS-connection to a multi party call (invoked by that MS) on hold.

4.2.2.17 retrieveMPTY (MS --> network)

This operation is invoked by an MS to request retrieval of a multi party call held by that MS.

4.2.2.18 splitMPTY (MS --> network)

This operation is invoked by an MS to request a private communication with one of the remote parties in a multi party call invoked by that MS.

4.2.2.19 forwardCUG-Info (MS --> network)

This operation is used by an MS to explicitly invoke a CUG call.

4.2.2.20 explicitCT (MS --> Network)

This operation is invoked by an MS to request the network to connect the two calls of the subscriber.

4.2.2.21 accessRegisterCCEntry (MS --> Network)

This operation is invoked by an MS to activate a CCBS request in the network.

4.2.2.22 callDeflection (MS --> Network)

This operation is invoked by an MS to request the network to deflect the incoming call to a specified destination.

4.2.2.23 userUserService (MS --> Network, Network --> MS)

This operation is invoked by an MS to request the network to allow an MS to send/receive information to/from another subscriber in association with a call.

4.2.2.24 lcs-LocationNotification (network --> MS)

This operation is invoked by the network to request a verification from the mobile subscriber for the attempted location request or to notify the subscriber about authorized location request.

4.2.2.25 lcs-MOLR (MS --> Network)

This operation is invoked by an MS to request the network to start location procedure, which is used to provide the MS location estimate, location assistance data or deciphering keys for broadcast assistance data.

4.2.2.26 lcs-AreaEventRequest (network --> MS)

This operation is invoked by the network to request a mobile to start the deferred MT-LR Area Event procedure.

4.2.2.27 lcs-AreaEventReport (MS --> network)

This operation is invoked by an MS to respond that the requested Area Event has occurred.

4.2.2.28 lcs-AreaEventCancellation (network --> MS)

This operation is invoked by the network to request a mobile to cancel the deferred MT-LR Area Event procedure.

4.3 Errors

4.3.1 Errors ASN.1 specification

The following ASN.1 module provides an ASN.1 specification of errors. Errors from MAP are imported in the SS-Protocol module in subclause 4.5. The module defines errors by allocating them a local value. For the involved errors the same local values as in MAP are allocated.

```
SS-Errors {
   itu-t identified-organization (4) etsi (0) mobileDomain (0) gsm-Access (2) modules (3)
   ss-Errors (1) version9 (9)}

DEFINITIONS ::=

BEGIN

IMPORTS

ERROR FROM
Remote-Operations-Information-Objects {joint-iso-itu-t remote-operations(4)
  informationObjects(5) version1(0)};

-- The MAP errors
-- unknownSubscriber, bearerServiceNotProvisioned, teleserviceNotProvisioned,
-- illegalSS-Operation, ss-ErrorStatus, ss-NotAvailable, ss-SubscriptionViolation,
-- ss-Incompatibility, systemFailure, dataMissing, unexpectedDataValue, facilityNotSupported,
-- pw-RegistrationFailure, negativePW-Check, callBarred, numberOfPW-AttemptsViolation,
-- absentSubscriber, illegalSubscriber, illegalEquipment, ussd-Busy, unknownAlphabet,
-- forwardingViolation, forwardingFailed
-- are imported from MAP-Errors in SS-Protocol module.

-- errors definition
resourcesNotAvailable  ERROR ::= {
    CODE    local:127 }
maxNumberOfMPTY-ParticipantsExceeded  ERROR ::= {
    CODE    local:126 }
invalidDeflectedToNumber  ERROR ::= {
    CODE    local:125 }
specialServiceCode  ERROR ::= {
    CODE    local:124 }
deflectionToServedSubscriber  ERROR ::= {
    CODE    local:123 }
rejectedByNetwork  ERROR ::= {
```

```
    CODE    local:122 }
rejectedByUser  ERROR ::= {
    CODE    local:121 }

END
```

4.3.2 Errors description

For each error this subclause provides a brief prose description.

4.3.2.1 unknownSubscriber

This error is returned by the network when it is requested to perform an operation concerning an unknown subscriber.

4.3.2.2 bearerServiceNotProvisioned

This error is returned by the network when it is requested to perform an operation on a supplementary service and not even a subset of the requested bearer service group has been subscribed to.

4.3.2.3 teleServiceNotProvisioned

This error is returned by the network when it is requested to perform an operation on a supplementary service and not even a subset of the requested teleservice group has been subscribed to.

4.3.2.4 illegalSS-Operation

This error is returned by the network when it is requested to perform an illegal operation which is defined as not applicable for the relevant supplementary service(s) (e.g. registration request for a service which must be registered by the administration). For the definition of the allowed operations for the individual supplementary services, see TS 24.08x and 24.09x-series of technical specifications.

4.3.2.5 ss-ErrorStatus

This error is returned by the network when it is requested to perform an operation which is not compatible with the current status of the relevant supplementary service. The current status may be given as additional information by use of the SS-parameter.

4.3.2.6 ss-NotAvailable

This error is returned by the network when it is requested to perform an operation on a supplementary service which is not available in the current location area.

4.3.2.7 ss-SubscriptionViolation

This error is returned by the network when it is requested to perform an operation on a supplementary service, transgressing the subscription restrictions. The nature of the restriction or the transgressed options may be sent as parameters.

4.3.2.8 ss-Incompatibility

This error is returned by the network when it is requested for a supplementary service operation incompatible with the status of an other supplementary service or with the teleservice or bearer service for which the operation is requested. This error shall only be used if the operation is not compatible for even a subset of the teleservice group or bearer service group specified in the request. The identity and status of the conflicting service may also be indicated. The additional information may contain the SS-code parameter, the Basic Service Group parameter and the SS-status parameter.

4.3.2.9 systemFailure

This error is returned by the network, when it cannot perform an operation because of a failure in the network.

4.3.2.10 dataMissing

This error is returned by the network when an optional parameter is missing in an invoke component or an inner data structure, while it is required by the context of the request.

4.3.2.11 unexpectedDataValue

This error is returned by the network when it receives a parameter with an unexpected value, without type violation.

4.3.2.12 passwordRegistrationFailure

This error is returned when a password registration procedure fails because of abnormal subscriber inputs. A more specific diagnostic may be passed as error parameter and indicates situations such as:

- invalid password format;
- new passwords mismatch.

4.3.2.13 negativePasswordCheck

This error is returned to indicate the negative result of a password check because the subscriber has not provided the required password or has provided a password which does not match the valid one.

4.3.2.14 facilityNotSupported

This error is returned by the network receiving a request about a facility which is not supported in the PLMN.

4.3.2.15 resourcesNotAvailable

This error is returned by the network to the MS if temporarily there are no resources to support e.g. a multi party call available in the network.

4.3.2.16 maxNumberOfMPTY-ParticipantsExceeded

This error is returned by the network to the MS if the request must be rejected because the number of subscribers to join a multi party call would exceed the maximum value.

4.3.2.17 callBarred

This error is returned by the network to the MS when call independent subscriber control procedures are barred by the operator. The parameter "operator barring" shall be included.

4.3.2.18 numberOfPW-AttemptsViolation

This error is returned by the network to the MS when the maximum number of wrong password attempts is exceeded.

4.3.2.19 absentSubscriber

This error is returned when the subscriber has activated the detach service or the system detects the absence condition. This error is not used on the radio interface but only between network entities.

4.3.2.20 illegalSubscriber

This error is returned when illegality of the access has been established by use of authentication procedure. This error is not used on the radio interface but only between network entities.

4.3.2.21 illegalEquipment

This error is returned when the IMEI check procedure has shown that the IMEI is blacklisted or not white—listed. This error is not used on the radio interface but only between network entities.

4.3.2.22 ussd-Busy

This error is returned by the MS to the network when the MS is not able to process the unstructured supplementary service data operation due to an on-going MMI input of the user or an already existing call independent supplementary service transaction.

4.3.2.23 unknownAlphabet

This error is returned by the MS or the network when the alphabet/language used for the unstructured supplementary service data operation is not known by the network or the MS.

4.3.2.24 invalidDeflectedToNumber

This error is returned if the requested deflected-to number is invalid.

4.3.2.25 specialServiceCode

This error is returned if diversion to a special service code was requested.

4.3.2.26 deflectionToServedSubscriber

This error is returned if a diversion to the served subscriber's number was requested.

4.3.2.27 rejectedByNetwork

This error is returned by the network when the network rejects User-to-User Signalling service request.

4.3.2.28 rejectedByUser

This error is returned by the remote party when the remote party rejects User-to-User Signalling service request.

4.3.2.29 positionMethodFailure

This error is returned by the network when the network is unable to obtain any of the location information requested or none of the information obtained satisfies the requested LCS QoS or if requested LCS assistance data could not be transferred or requested deciphering keys for broadcast assistance data could not be returned.

4.4 Data types and identifiers

4.4.1 General

The data types used in the SS protocol specifications are described in the ASN.1 module provided in subclause 4.4.2, while subclause 4.4.3 provides an overview of the identifiers used in SS ASN.1 specifications.

Named values have been defined in the following module for the upper boundaries of the value ranges associated to several sub-type specifications.

4.4.2 ASN.1 data types

This subclause provides an ASN.1 module defining the abstract data types in operations and errors specification. Only data types which are specific for this specification are defined. All other data types are imported from MAP together with the import of operations and errors.

```
SS-DataTypes {
   itu-t identified-organization (4) etsi (0) mobileDomain (0) gsm-Access (2) modules (3)
   ss-DataTypes (2) version9 (9)}

DEFINITIONS

IMPLICIT TAGS ::=

BEGIN

-- exports all data types defined in this module

IMPORTS

SS-Code
FROM MAP-SS-Code {
   itu-t identified-organization (4) etsi (0) mobileDomain (0) gsm-Network (1) modules (3)
   map-SS-Code (15) version9 (9)}

-- imports MAP-SS-DataTypes
SS-Status, USSD-DataCodingScheme, USSD-String, CCBS-Feature
-- USSD-DataCodingScheme, USSD-String were introduced because of CNAP.
FROM MAP-SS-DataTypes {
   itu-t identified-organization (4) etsi (0) mobileDomain (0) gsm-Network (1) modules (3)
   map-SS-DataTypes (14) version9 (9)}

GSN-Address,
CUG-Index,
NotificationToMSUser
FROM MAP-MS-DataTypes {
   itu-t identified-organization (4) etsi (0) mobileDomain (0) gsm-Network (1) modules (3)
   map-MS-DataTypes (11) version9 (9)}

maxSignalInfoLength,
ISDN-AddressString,
ISDN-SubaddressString,
AlertingPattern,
LCSClientExternalID,
AddressString,
LCSServiceTypeID
FROM MAP-CommonDataTypes {
   itu-t identified-organization (4) etsi (0) mobileDomain (0) gsm-Network (1) modules (3)
   map-CommonDataTypes (18) version9 (9)}

LocationType,
DeferredLocationEventType,
LCSClientName,
LCS-QoS,
Horizontal-Accuracy,
ResponseTime,
Ext-GeographicalInformation,
SupportedGADShapes,
Add-GeographicalInformation,
LCSRequestorID,
LCS-ReferenceNumber,
LCSCodeword,
AreaEventInfo
FROM MAP-LCS-DataTypes {
   itu-t identified-organization (4) etsi (0) mobileDomain (0)
   gsm-Network (1) modules (3) map-LCS-DataTypes (25) version9 (9)}

;

-- data types definition

SS-UserData ::= IA5String (SIZE (1.. maxSignalInfoLength))

NotifySS-Arg ::= SEQUENCE{
    ss-Code                     [1]     SS-Code OPTIONAL,
    ss-Status                   [4]     SS-Status OPTIONAL,
    ss-Notification             [5]     SS-Notification OPTIONAL,
    callIsWaiting-Indicator     [14]    NULL OPTIONAL,
    callOnHold-Indicator        [15]    CallOnHold-Indicator OPTIONAL,
    mpty-Indicator              [16]    NULL OPTIONAL,
    cug-Index                   [17]    CUG-Index OPTIONAL,
    clirSuppressionRejected     [18]    NULL OPTIONAL,
    ... ,
    ect-Indicator               [19]    ECT-Indicator OPTIONAL,
```

```
    nameIndicator               [20]    NameIndicator OPTIONAL,
    ccbs-Feature                [21]    CCBS-Feature OPTIONAL,
    alertingPattern             [22]    AlertingPattern OPTIONAL,
    multicall-Indicator         [23]    Multicall-Indicator OPTIONAL}

--  The nameIndicator is defined because of CNAP.

Multicall-Indicator ::= ENUMERATED {
        nbr-SNexceeded (0),
        nbr-Userexceeded (1)}

ForwardChargeAdviceArg ::= SEQUENCE{
    ss-Code                     [0]     SS-Code,
    chargingInformation         [1]     ChargingInformation,
    ...}

SS-Notification ::= OCTET STRING (SIZE (1))

--    Bit 8 7 6 5 4 00000 (Unused)

--  Bit 3   Call is forwarded indication to A-subscriber
--      (calling subscriber)
--  0   No information content
--  1   Outgoing call has been forwarded to C

--  Bit 2   Call is forwarded indication to B-subscriber
--      (forwarding subscriber)
--  0   No information content
--  1   Incoming call has been forwarded to C

--  Bit 1   Call is forwarded indication to C-subscriber
--      (forwarded-to subscriber)
--  0   No information content
--  1   Incoming call is a forwarded call

ChargingInformation ::= SEQUENCE{
    e1  [1] E1 OPTIONAL,
    e2  [2] E2 OPTIONAL,
    e3  [3] E3 OPTIONAL,
    e4  [4] E4 OPTIONAL,
    e5  [5] E5 OPTIONAL,
    e6  [6] E6 OPTIONAL,
    e7  [7] E7 OPTIONAL,
    ...}

E1 ::= INTEGER (0..max10TimesUnitsPerTime)
max10TimesUnitsPerTime INTEGER ::= 8191

E2 ::= INTEGER (0..max10TimesTimeInterval)
max10TimesTimeInterval INTEGER ::= 8191

E3 ::= INTEGER (0..max100TimesScalingFactor)
max100TimesScalingFactor INTEGER ::= 8191

E4 ::= INTEGER (0..max10TimesIncrement)
max10TimesIncrement INTEGER ::= 8191

E5 ::= INTEGER (0..max10TimesIncrementPerDataInterval)
max10TimesIncrementPerDataInterval INTEGER ::= 8191

E6 ::= INTEGER (0..maxNumberOfSegmentsPerDataInterval)
maxNumberOfSegmentsPerDataInterval INTEGER ::= 8191

E7 ::= INTEGER (0..max10TimesInitialTime)
max10TimesInitialTime INTEGER ::= 8191

CallOnHold-Indicator    ::= ENUMERATED {
            callRetrieved (0),
            callOnHold (1)}

ForwardCUG-InfoArg ::= SEQUENCE {
    cug-Index           [0] CUG-Index OPTIONAL,
    suppressPrefCUG     [1] NULL OPTIONAL,
    suppressOA          [2] NULL OPTIONAL,
    ...}

ECT-Indicator   ::= SEQUENCE {
    ect-CallState       [0] ECT-CallState,
```

```
    rdn [1] RDN OPTIONAL,
    ...}

ECT-CallState   ::= ENUMERATED {
        alerting (0),
        active (1)}

    NameIndicator ::= SEQUENCE {
        callingName     [0] Name OPTIONAL,
        ...}

    Name ::= CHOICE {
        namePresentationAllowed     [0] NameSet,
        presentationRestricted      [1] NULL,
        nameUnavailable             [2] NULL,
        namePresentationRestricted  [3] NameSet}

    NameSet ::= SEQUENCE {
        dataCodingScheme        [0] USSD-DataCodingScheme,
        lengthInCharacters      [1] INTEGER,
        nameString              [2] USSD-String,
        ...}

-- NameIndicator, Name and NameSet are defined because of CNAP.
-- The USSD-DataCodingScheme shall indicate use of the default alphabet through the
-- following encoding:
--    bit  7   6   5   4   3   2   1   0
--       | 0   0   0   0 | 1   1   1   1|

RDN ::= CHOICE {
    presentationAllowedAddress              [0] RemotePartyNumber,
    presentationRestricted                  [1] NULL,
    numberNotAvailableDueToInterworking     [2] NULL,
    presentationRestrictedAddress           [3] RemotePartyNumber}

RemotePartyNumber   ::= SEQUENCE {
    partyNumber             [0] ISDN-AddressString,
    partyNumberSubaddress   [1] ISDN-SubaddressString OPTIONAL,
    ...}

AccessRegisterCCEntryArg    ::= SEQUENCE {
    ...}

CallDeflectionArg   ::= SEQUENCE {
    deflectedToNumber       [0] AddressString,
    deflectedToSubaddress   [1] ISDN-SubaddressString OPTIONAL,
    ...}

UserUserServiceArg ::= SEQUENCE {
    uUS-Service     [0] UUS-Service,
    uUS-Required    [1] BOOLEAN,
    ... }

UUS-Service ::= ENUMERATED {
    uUS1 (1),
    uUS2 (2),
    uUS3 (3),
    ... }

-- exception handling:
-- In case of UUS-Service with any other value, indicated as "UUS required",
-- but not understood by the MS, the call will be cleared.

LocationNotificationArg ::= SEQUENCE {
    notificationType    [0] NotificationToMSUser,
    locationType        [1] LocationType,
    lcsClientExternalID [2] LCSClientExternalID     OPTIONAL,
    lcsClientName       [3] LCSClientName           OPTIONAL,
    ... ,
    lcsRequestorID      [4] LCSRequestorID          OPTIONAL,
    lcsCodeword         [5] LCSCodeword             OPTIONAL,
    lcsServiceTypeID    [6] LCSServiceTypeID        OPTIONAL }
-- exception handling:
-- At reception of an unrecognised notificationType value the receiver shall reject the
-- operation with a return error cause of unexpected data value.
-- At reception of an unrecognised locationType value the receiver shall reject the
-- operation with a return error cause of unexpected data value.
```

```
LocationNotificationRes ::= SEQUENCE {
    verificationResponse        [0] VerificationResponse OPTIONAL,
    ...}

VerificationResponse::= ENUMERATED {
    permissionDenied    (0),
    permissionGranted   (1),
    ... }

-- exception handling:
-- an unrecognized value shall be treated the same as value 0 (permissionDenied)

LCS-MOLRArg ::= SEQUENCE {
    molr-Type           [0] MOLR-Type,
    locationMethod      [1] LocationMethod          OPTIONAL,
    lcs-QoS             [2] LCS-QoS                 OPTIONAL,
    lcsClientExternalID [3] LCSClientExternalID     OPTIONAL,
    mlc-Number          [4] ISDN-AddressString      OPTIONAL,
    gpsAssistanceData   [5] GPSAssistanceData       OPTIONAL,
    ...,
    supportedGADShapes  [6] SupportedGADShapes      OPTIONAL,
    lcsServiceTypeID    [7] LCSServiceTypeID        OPTIONAL }
-- The parameter locationMethod shall be included if and only if the molr-Type is set to value
-- deCipheringKeys or assistanceData.
-- The parameter gpsAssistanceData shall be included if and only if the molr-Type is set to value
-- assistanceData and locationMethod is set to value assistedGPS.

MOLR-Type::= ENUMERATED {
    locationEstimate            (0),
    assistanceData              (1),
    deCipheringKeys             (2),
    ... }
-- exception handling:
-- an unrecognized value shall be rejected by the receiver with a return error cause of
-- unexpected data value.

LocationMethod::= ENUMERATED {
    msBasedEOTD         (0),
    msAssistedEOTD      (1),
    assistedGPS         (2),
    ...,
    msBasedOTDOA        (3)
    }
-- exception handling:
-- When this parameter is received with value msBasedEOTD or msAssistedEOTD and the MS
-- is camped on an UMTS Service Area then the receiver shall reject it
-- with a return error cause of unexpected data value.
-- When this parameter is received with value msBasedOTDOA and the MS
-- is camped on a GSM Cell then the receiver shall reject it with a return error cause of
-- unexpected data value.
-- an unrecognized value shall be rejected by the receiver with a return error cause of
-- unexpected data value.

GPSAssistanceData::= OCTET STRING (SIZE (1..38))
-- Octets 1 to 38 are coded in the same way as the octets 3 to 7+2n of Requested GPS Data IE
-- in 3GPP TS 49.031.

LCS-MOLRRes::= SEQUENCE {
    locationEstimate        [0] Ext-GeographicalInformation     OPTIONAL,
    decipheringKeys         [1] DecipheringKeys                 OPTIONAL,
    ...,
    add-LocationEstimate    [2] Add-GeographicalInformation         OPTIONAL}
-- Parameters locationEstimate or add-LocationEstimate (one but not both)
-- shall be included if and only if the
-- molr-Type in LocationRequestArg was set to value locationEstimate.
-- Parameter add-LocationEstimate shall not be included if the supportedGADShapes
-- parameter was not received in the LCS-MOLRArg.
-- The locationEstimate and the add-locationEstimate parameters shall not be sent if
-- the supportedGADShapes parameter has been received in LCS-MOLRArg
-- and the shape encoded in locationEstimate or add-LocationEstimate is not marked
-- as supported in supportedGADShapes. In such a case LCS-MOLRArg
-- shall be rejected with error FacilityNotSupported with additional indication
-- shapeOfLocationEstimateNotSupported.
-- Parameter decipheringKeys shall be included if and only if the molr-Type
-- in LocationRequestArg was set to value deCipheringKeys.
```

```
DecipheringKeys::= OCTET STRING (SIZE (15))
-- Octets in DecipheringKeys are coded in the same way as the octets 3 to 17 of Deciphering Key IE
-- in 3GPP TS 49.031. I.e. these octets contain Current Deciphering Key, Next Deciphering Key and
-- Ciphering Key Flag.
LCS-AreaEventRequestArg ::= SEQUENCE {
    referenceNumber             [0] LCS-ReferenceNumber,
    h-gmlc-address              [1] GSN-Address,
    deferredLocationEventType   [3] DeferredLocationEventType,
    areaEventInfo               [4] AreaEventInfo,
    ... }

-- the msAvailableValue in the DeferredLocationEventType is not applicable for this procedure

LCS-AreaEventReportArg  ::= SEQUENCE {
    referenceNumber             [0] LCS-ReferenceNumber,
    h-gmlc-address              [1] GSN-Address,
    ... }

LCS-AreaEventCancellationArg    ::= SEQUENCE {
    referenceNumber             [0] LCS-ReferenceNumber,
    h-gmlc-address              [1] GSN-Address,
    ... }

END
```

4.4.3 Identifiers definition

The parameters which are described in the following subclauses correspond to the identifiers used in operation and error descriptions.

4.4.3.1 chargingInformation

The chargingInformation identifier refers to the necessary information for the Advice of Charge supplementary service (see TS 22.024).

4.4.3.2 e1

The e1 identifier refers to 10 times the number of LPLMN units per time interval in connection with the Advice of Charge supplementary service, see TS 22.024.

4.4.3.3 e2

The e2 identifier refers to 10 times the length of the time interval in seconds in connection with the Advice of Charge supplementary service, see TS 22.024.

4.4.3.4 e3

The e3 identifier refers to 100 times the scaling factor to convert from LPLMN units to HPLMN units in connection with the Advice of Charge supplementary service, see TS 22.024.

4.4.3.5 e4

The e4 identifier refers to 10 times the LPLMN increment in connection with the Advice of Charge supplementary service, see TS 22.024.

4.4.3.6 e5

The e5 identifier refers to 10 times the number of LPLMN units incremented per data interval in connection with the Advice of Charge supplementary service, see TS 22.024.

4.4.3.7 e6

The e6 identifier refers to the number of segments per data interval in connection with the Advice of Charge supplementary service, see TS 22.024.

4.4.3.8 e7

The e7 identifier refers to 10 times the length of the initial time interval in seconds in connection with the Advice of Charge supplementary service, see TS 22.024.

4.4.3.9 ss-Code

The ss-Code identifier refers to the code which identify a supplementary service or a group of supplementary services.

4.4.3.10 ss-Notification

The ss-Notification identifier refers to one or several supplementary service notifications which have to be forwarded to a mobile subscriber.

4.4.3.11 ss-Status

The ss-Status identifier refers to the status of a supplementary service.

4.4.3.12 callIsWaiting-Indicator

The callIsWaiting-Indicator identifier refers to the indication given to the mobile station that the call is waiting.

4.4.3.13 callOnhold-Indicator

The callOnHold-Indicator identifier refers to the indication given to the mobile station that the call has been put on hold or has been retrieved.

4.4.3.14 mpty-Indicator

The mpty-Indicator identifier refers to the indication given to the mobile station that the multi party call has been invoked.

4.4.3.15 forwardCUG-InfoArg

The forwardCUG-InfoArg identifier refers to the indication given from the mobile subscriber to the network in connection with explicit invocation of a CUG call.

4.4.3.16 cug-Index

The cug-Index identifier refers to the index of a CUG given in an explicit invocation of a CUG call.

4.4.3.17 suppressPrefCUG

The suppressPrefCUG identifier refers to the mobile subscribers request to the network to prohibit the use of the preferential CUG.

4.4.3.18 suppressOA

The suppressOA identifier refers to the mobile subscribers request to the network to prohibit the use of the subscriber option "OA allowed".

4.4.3.19 clirSuppressionRejected

The clirSuppressionRejected identifier refers to the indication given to the mobile station that the CLIR suppression request has been rejected.

4.4.3.20 ect-Indicator

The ect-Indicator identifier refers to the indication given to the mobile station that the call was transferred.

4.4.3.21 ect-CallState

The ect-CallState identifier refers to the state of the call to the other remote party in which Explicit Call Transfer was invoked.

4.4.3.22 rdn

The Rdn identifier refers to the line identity information of the other remote party.

4.4.3.23 presentationAllowedAddress

The presentationAllowedAddress identifier refers to the line identity of the other remote party that is allowed to be presented.

4.4.3.24 presentationRestricted

The presentationRestricted identifier refers to the restriction of presentation of the line identity of the other remote party.

Also, the identifier refers to the restriction of presentation of the name identity of the calling party to the called party.

4.4.3.25 numberNotAvailableDueToInterworking

The numberNotAvailableDueToInterworking identifier refers to the unavailability of the line identity of the other remote party.

4.4.3.26 presentationRestrictedAddress

The presentationRestrictedAddress identifier refers to the line identity of the other remote party which presentation restriction is overridden.

4.4.3.27 partyNumber

The partyNumber identifier refers to the remote party number.

4.4.3.28 partyNumberSubaddress

The partyNumberSubaddress identifier refers to remote party number subaddress.

4.4.3.29 nameIndicator

The nameIndicator identifier refers to the indication given to the mobile station that the name presentation has been invoked.

4.4.3.30 namePresentationAllowed

The namePresentationAllowed identifier refers to the presentation of the calling party"s name identity to the called party.

4.4.3.31 nameUnavailable

The nameUnavailable identifier refers to the unavailability of the calling party"s name identity to be offered to the called party.

4.4.3.32 namePresentationRestricted

The namePresentationRestricted identifier refers to the calling party"s name identity to be offered to the called party with which presentation restriction is overridden.

4.4.3.33 deflectedToNumber

The DeflectedToNumber identifier refers to a party an incoming shall be deflected to.

4.4.3.34 deflectedToSubaddress

The DeflectedToSubaddress identifier refers to a subaddress an incoming call shall be deflected to.

4.4.3.35 uUS-Service

The uUS-Service identifier refers to the UUS service (service 1, service 2 or service 3) to be requested.

4.4.3.36 uUS-Required

The uUS-Required identifier refers to the option ("UUS required" or "UUS not required") given when requesting the UUS service.

4.4.3.37 locationNotificationArg

The locationNotificationArg identifier refers to the location notification request which is sent to the MS by the network.

4.4.3.38 notificationType

The notificationType identifier refers to the type of location notification (notification or privacy verification).

4.4.3.39 locationNotificationRes

The locationNotificationRes identifier refers to the location notification response which is sent to the network by the MS.

4.4.3.40 verificationResponse

The VerificationResponse identifier refers to the privacy verification response given by the MS user.

4.4.3.41 lcs-MOLRArg

The lcs-MOLRArg identifier refers to the MO-LR request parameters which are sent to the network by the MS.

4.4.3.42 molr-Type

The molr-Type identifier refers to the type of MO-LR.

4.4.3.43 locationMethod

The locationMethod identifier refers to the location method, for which assistance data is requested by the MS.

4.4.3.44 gpsAssistanceData

The gpsAssistanceData identifier refers to the indication, which GPS assistance data is requested by the MS.

4.4.3.45 lcs-MOLRRes

The lcs-MOLRRes identifier refers to the MO-LR response parameters which are sent to the MS by the network.

4.4.3.46 decipheringKeys

The decipheringKeys identifier refers to the set of deciphering keys, that contains Current Deciphering Key, Next Deciphering Key and Ciphering Key Flag.

4.4.3.47 multicall-Indicator

The multicall-Indicator identifier refers to the indication given to the mobile station that the number of active bearers has exceeded the maximum number.

4.5 Operations and errors implementation

For the actual implementation of supplementary services, operations and errors have to be defined by value. The following ASN.1 module, imports operation from the ASN.1 module described in subclause 4.2 and operations and errors from MAP.

```
SS-Protocol {
   itu-t identified-organization (4) etsi (0) mobileDomain (0)
   gsm-Access (2) modules (3) ss-Protocol (3) version9 (9)}

DEFINITIONS ::=

BEGIN

IMPORTS

OPERATION
FROM Remote-Operations-Information-Objects {
   joint-iso-itu-t remote-operations(4) informationObjects(5) version1(0)}

-- imports operations

-- imports operation from MAP-MobileServiceOperations
forwardCheckSS-Indication
FROM MAP-MobileServiceOperations {
   itu-t identified-organization (4) etsi (0) mobileDomain (0) gsm-Network (1) modules (3)
   map-MobileServiceOperations (5) version9 (9)}

-- imports operations from MAP-SupplementaryServiceOperations
registerSS, eraseSS, activateSS, deactivateSS, interrogateSS, registerPassword, getPassword,
processUnstructuredSS-Request, unstructuredSS-Request, unstructuredSS-Notify, eraseCC-Entry
FROM MAP-SupplementaryServiceOperations {
   itu-t identified-organization (4) etsi (0) mobileDomain (0) gsm-Network (1) modules (3)
   map-SupplementaryServiceOperations (8) version9 (9)}

-- imports operations from SS-Operations
processUnstructuredSS-Data, notifySS, forwardChargeAdvice, buildMPTY, holdMPTY, retrieveMPTY,
splitMPTY, explicitCT, forwardCUG-Info, accessRegisterCCEntry, callDeflection, userUserService,
lcs-LocationNotification, lcs-MOLR, lcs-AreaEventRequest, lcs-AreaEventReport, lcs-
AreaEventCancellation
FROM SS-Operations {
   itu-t identified-organization (4) etsi (0) mobileDomain (0) gsm-Access (2) modules (3)
   ss-Operations (0) version9 (9)}

;

Supported-SS-Operations OPERATION ::= {forwardCheckSS-Indication | registerSS | eraseSS |
    activateSS | deactivateSS | interrogateSS | registerPassword | getPassword |
    processUnstructuredSS-Request | unstructuredSS-Request | unstructuredSS-Notify | eraseCC-Entry |
    processUnstructuredSS-Data | notifySS | forwardChargeAdvice | buildMPTY | holdMPTY |
    retrieveMPTY | splitMPTY | explicitCT | forwardCUG-Info | accessRegisterCCEntry |
    callDeflection | userUserService | lcs-LocationNotification | lcs-MOLR | lcs-AreaEventRequest |
    lcs-AreaEventReport | lcs-AreaEventCancellation }



END
```

Annex A (informative): Expanded ASN.1 Module "SS-Protocol"

```
--         Expanded ASN1 Module 'SS-Operations'
--SIEMENS ASN.1 Compiler       R5.70  (Production_5.70)
--            Date: 2004-03-19 Time: 12:34:55

SS-Operations{ 0 identified-organization (4) etsi (0) mobileDomain (0) gsm-Access (2) modules (3)
ss-Operations (0) version9 (9) }

DEFINITIONS

::=

BEGIN

EXPORTS
   processUnstructuredSS-Data,
   notifySS,
   forwardChargeAdvice,
   forwardCUG-Info,
   buildMPTY,
   holdMPTY,
   retrieveMPTY,
   splitMPTY,
   explicitCT,
   accessRegisterCCEntry,
   callDeflection,
   userUserService,
   lcs-LocationNotification,
   lcs-MOLR,
   lcs-AreaEventRequest,
   lcs-AreaEventReport,
   lcs-AreaEventCancellation;


processUnstructuredSS-Data OPERATION  ::=  {
   ARGUMENT     IA5String  ( SIZE( 1 .. 200 ) )
   RESULT     IA5String  ( SIZE( 1 .. 200 ) )
   ERRORS     {
      systemFailure |
      unexpectedDataValue }
   CODE     local      : 19
   }

notifySS   OPERATION  ::=  {
   ARGUMENT     SEQUENCE {
      ss-Code                   [1] IMPLICIT OCTET STRING ( SIZE( 1 ) ) OPTIONAL,
      ss-Status                 [4] IMPLICIT OCTET STRING ( SIZE( 1 ) ) OPTIONAL,
      ss-Notification           [5] IMPLICIT OCTET STRING ( SIZE( 1 ) ) OPTIONAL,
      callIsWaiting-Indicator   [14] IMPLICIT NULL OPTIONAL,
      callOnHold-Indicator      [15] IMPLICIT ENUMERATED {
         callRetrieved    ( 0 ),
         callOnHold       ( 1 ) } OPTIONAL,
      mpty-Indicator            [16] IMPLICIT NULL OPTIONAL,
      cug-Index                 [17] IMPLICIT INTEGER ( 0 .. 32767 ) OPTIONAL,
      clirSuppressionRejected   [18] IMPLICIT NULL OPTIONAL,
      ... ,
      ect-Indicator             [19] IMPLICIT SEQUENCE {
         ect-CallState   [0] IMPLICIT ENUMERATED {
            alerting    ( 0 ),
            active      ( 1 ) },
         rdn             [1] CHOICE {
            presentationAllowedAddress              [0] IMPLICIT SEQUENCE {
               partyNumber             [0] IMPLICIT OCTET STRING ( SIZE( 1 .. 20 ) ) ( SIZE( 1 .. 9
) ),
               partyNumberSubaddress   [1] IMPLICIT OCTET STRING ( SIZE( 1 .. 21 ) ) OPTIONAL,
               ... },
            presentationRestricted                  [1] IMPLICIT NULL,
            numberNotAvailableDueToInterworking     [2] IMPLICIT NULL,
            presentationRestrictedAddress           [3] IMPLICIT SEQUENCE {
               partyNumber             [0] IMPLICIT OCTET STRING ( SIZE( 1 .. 20 ) ) ( SIZE( 1 .. 9
) ),
```

```
               partyNumberSubaddress   [1] IMPLICIT OCTET STRING ( SIZE( 1 .. 21 ) ) OPTIONAL,
               ... }} OPTIONAL,
         ... } OPTIONAL,
      nameIndicator             [20] IMPLICIT SEQUENCE {
         callingName   [0] CHOICE {
            namePresentationAllowed        [0] IMPLICIT SEQUENCE {
               dataCodingScheme     [0] IMPLICIT OCTET STRING ( SIZE( 1 ) ),
               lengthInCharacters   [1] IMPLICIT INTEGER,
               nameString           [2] IMPLICIT OCTET STRING ( SIZE( 1 .. 160 ) ),
               ... },
            presentationRestricted         [1] IMPLICIT NULL,
            nameUnavailable                [2] IMPLICIT NULL,
            namePresentationRestricted     [3] IMPLICIT SEQUENCE {
               dataCodingScheme     [0] IMPLICIT OCTET STRING ( SIZE( 1 ) ),
               lengthInCharacters   [1] IMPLICIT INTEGER,
               nameString           [2] IMPLICIT OCTET STRING ( SIZE( 1 .. 160 ) ),
               ... }} OPTIONAL,
         ... } OPTIONAL,
      ccbs-Feature              [21] IMPLICIT SEQUENCE {
         ccbs-Index               [0] IMPLICIT INTEGER ( 1 .. 5 ) OPTIONAL,
         b-subscriberNumber       [1] IMPLICIT OCTET STRING ( SIZE( 1 .. 20 ) ) ( SIZE( 1 .. 9 ) )
OPTIONAL,
         b-subscriberSubaddress   [2] IMPLICIT OCTET STRING ( SIZE( 1 .. 21 ) ) OPTIONAL,
         basicServiceGroup        [3] CHOICE {
            bearerService     [2] IMPLICIT OCTET STRING ( SIZE( 1 ) ),
            teleservice       [3] IMPLICIT OCTET STRING ( SIZE( 1 ) )} OPTIONAL,
         ... } OPTIONAL,
      alertingPattern           [22] IMPLICIT OCTET STRING ( SIZE( 1 ) ) OPTIONAL,
      multicall-Indicator       [23] IMPLICIT ENUMERATED {
         nbr-SNexceeded      ( 0 ),
         nbr-Userexceeded    ( 1 ) } OPTIONAL}
   CODE     local      : 16
   }

forwardChargeAdvice OPERATION  ::=  {
   ARGUMENT     SEQUENCE {
      ss-Code               [0] IMPLICIT OCTET STRING ( SIZE( 1 ) ),
      chargingInformation   [1] IMPLICIT SEQUENCE {
         e1         [1] IMPLICIT INTEGER ( 0 .. 8191 ) OPTIONAL,
         e2         [2] IMPLICIT INTEGER ( 0 .. 8191 ) OPTIONAL,
         e3         [3] IMPLICIT INTEGER ( 0 .. 8191 ) OPTIONAL,
         e4         [4] IMPLICIT INTEGER ( 0 .. 8191 ) OPTIONAL,
         e5         [5] IMPLICIT INTEGER ( 0 .. 8191 ) OPTIONAL,
         e6         [6] IMPLICIT INTEGER ( 0 .. 8191 ) OPTIONAL,
         e7         [7] IMPLICIT INTEGER ( 0 .. 8191 ) OPTIONAL,
         ... },
      ... }
   RETURN  RESULT     TRUE
   CODE     local      : 125
   }

forwardCUG-Info OPERATION  ::=  {
   ARGUMENT     SEQUENCE {
      cug-Index         [0] IMPLICIT INTEGER ( 0 .. 32767 ) OPTIONAL,
      suppressPrefCUG   [1] IMPLICIT NULL OPTIONAL,
      suppressOA        [2] IMPLICIT NULL OPTIONAL,
      ... }
   CODE     local      : 120
   }

buildMPTY  OPERATION  ::=  {
   RETURN  RESULT     TRUE
   ERRORS     {
      illegalSS-Operation |
      ss-ErrorStatus |
      ss-NotAvailable |
      ss-Incompatibility |
      systemFailure |
      resourcesNotAvailable |
      maxNumberOfMPTY-ParticipantsExceeded }
   CODE     local      : 124
   }

holdMPTY   OPERATION  ::=  {
   RETURN  RESULT     TRUE
   ERRORS     {
      illegalSS-Operation |
      ss-ErrorStatus |
```

```
      ss-Incompatibility |
      facilityNotSupported |
      systemFailure }
   CODE     local      : 123
   }

retrieveMPTY OPERATION  ::=  {
   RETURN  RESULT     TRUE
   ERRORS     {
      illegalSS-Operation |
      ss-ErrorStatus |
      ss-Incompatibility |
      facilityNotSupported |
      systemFailure }
   CODE     local      : 122
   }

splitMPTY  OPERATION  ::=  {
   RETURN  RESULT     TRUE
   ERRORS     {
      illegalSS-Operation |
      ss-ErrorStatus |
      ss-Incompatibility |
      facilityNotSupported |
      systemFailure }
   CODE     local      : 121
   }

explicitCT OPERATION  ::=  {
   RETURN  RESULT     TRUE
   ERRORS     {
      illegalSS-Operation |
      ss-ErrorStatus |
      ss-NotAvailable |
      ss-Incompatibility |
      facilityNotSupported |
      systemFailure |
      resourcesNotAvailable |
      callBarred }
   CODE     local      : 126
   }

accessRegisterCCEntry OPERATION  ::=  {
   ARGUMENT     SEQUENCE {
      ... }
   RESULT     SEQUENCE {
      ccbs-Feature   [0] IMPLICIT SEQUENCE {
         ccbs-Index               [0] IMPLICIT INTEGER ( 1 .. 5 ) OPTIONAL,
         b-subscriberNumber       [1] IMPLICIT OCTET STRING ( SIZE( 1 .. 20 ) ) ( SIZE( 1 .. 9 ) )
OPTIONAL,
         b-subscriberSubaddress   [2] IMPLICIT OCTET STRING ( SIZE( 1 .. 21 ) ) OPTIONAL,
         basicServiceGroup        [3] CHOICE {
            bearerService     [2] IMPLICIT OCTET STRING ( SIZE( 1 ) ),
            teleservice       [3] IMPLICIT OCTET STRING ( SIZE( 1 ) )} OPTIONAL,
         ... } OPTIONAL,
      ... }
   ERRORS     {
      systemFailure |
      dataMissing |
      unexpectedDataValue |
      callBarred |
      illegalSS-Operation |
      ss-ErrorStatus |
      ss-Incompatibility |
      shortTermDenial |
      longTermDenial |
      facilityNotSupported }
   CODE     local      : 119
   }

callDeflection OPERATION  ::=  {
   ARGUMENT     SEQUENCE {
      deflectedToNumber       [0] IMPLICIT OCTET STRING ( SIZE( 1 .. 20 ) ),
      deflectedToSubaddress   [1] IMPLICIT OCTET STRING ( SIZE( 1 .. 21 ) ) OPTIONAL,
      ... }
   RETURN  RESULT     TRUE
   ERRORS     {
      illegalSS-Operation |
```

```
      ss-ErrorStatus |
      ss-NotAvailable |
      ss-Incompatibility |
      facilityNotSupported |
      systemFailure |
      resourcesNotAvailable |
      forwardingViolation |
      callBarred |
      deflectionToServedSubscriber |
      invalidDeflectedToNumber |
      specialServiceCode |
      forwardingFailed }
   CODE     local      : 117
   }

userUserService OPERATION  ::=  {
   ARGUMENT     SEQUENCE {
      uUS-Service    [0] IMPLICIT ENUMERATED {
         uUS1        ( 1 ),
         uUS2        ( 2 ),
         uUS3        ( 3 ),
         ...  },
      uUS-Required   [1] IMPLICIT BOOLEAN,
      ... }
   RETURN  RESULT     TRUE
   ERRORS     {
      illegalSS-Operation |
      ss-ErrorStatus |
      ss-NotAvailable |
      ss-Incompatibility |
      facilityNotSupported |
      systemFailure |
      resourcesNotAvailable |
      rejectedByNetwork |
      rejectedByUser }
   CODE     local      : 118
   }

lcs-LocationNotification OPERATION  ::=  {
   ARGUMENT     SEQUENCE {
      notificationType      [0] IMPLICIT ENUMERATED {
         notifyLocationAllowed                              ( 0 ),
         notifyAndVerify-LocationAllowedIfNoResponse        ( 1 ),
         notifyAndVerify-LocationNotAllowedIfNoResponse     ( 2 ),
         ... ,
         locationNotAllowed                                 ( 3 ) },
      locationType          [1] IMPLICIT SEQUENCE {
         locationEstimateType        [0] IMPLICIT ENUMERATED {
            currentLocation              ( 0 ),
            currentOrLastKnownLocation   ( 1 ),
            initialLocation              ( 2 ),
            ... ,
            activateDeferredLocation     ( 3 ),
            cancelDeferredLocation       ( 4 ) },
         ... ,
         deferredLocationEventType   [1] IMPLICIT BIT STRING {
            msAvailable  (0 ),
            enteringIntoArea  (1 ),
            leavingFromArea  (2 ),
            beingInsideArea  (3 )} ( SIZE( 1 .. 16 ) ) OPTIONAL},
      lcsClientExternalID   [2] IMPLICIT SEQUENCE {
         externalAddress      [0] IMPLICIT OCTET STRING ( SIZE( 1 .. 20 ) ) ( SIZE( 1 .. 9 ) )
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,
      lcsClientName         [3] IMPLICIT SEQUENCE {
         dataCodingScheme      [0] IMPLICIT OCTET STRING ( SIZE( 1 ) ),
```

```
         nameString            [2] IMPLICIT OCTET STRING ( SIZE( 1 .. 160 ) ) ( SIZE( 1 .. 63 ) ),
         ... ,
         lcs-FormatIndicator   [3] IMPLICIT ENUMERATED {
            logicalName      ( 0 ),
            e-mailAddress    ( 1 ),
            msisdn           ( 2 ),
            url              ( 3 ),
            sipUrl           ( 4 ),
            ...  } OPTIONAL} OPTIONAL,
      ... ,
      lcsRequestorID        [4] IMPLICIT SEQUENCE {
         dataCodingScheme      [0] IMPLICIT OCTET STRING ( SIZE( 1 ) ),
         requestorIDString     [1] IMPLICIT OCTET STRING ( SIZE( 1 .. 160 ) ) ( SIZE( 1 .. 63 ) ),
         ... ,
         lcs-FormatIndicator   [2] IMPLICIT ENUMERATED {
            logicalName      ( 0 ),
            e-mailAddress    ( 1 ),
            msisdn           ( 2 ),
            url              ( 3 ),
            sipUrl           ( 4 ),
            ...  } OPTIONAL} OPTIONAL,
      lcsCodeword           [5] IMPLICIT SEQUENCE {
         dataCodingScheme    [0] IMPLICIT OCTET STRING ( SIZE( 1 ) ),
         lcsCodewordString   [1] IMPLICIT OCTET STRING ( SIZE( 1 .. 160 ) ) ( SIZE( 1 .. 20 ) ),
         ... } OPTIONAL,
      lcsServiceTypeID      [6] IMPLICIT INTEGER ( 0 .. 127 ) OPTIONAL}
   RESULT     SEQUENCE {
      verificationResponse   [0] IMPLICIT ENUMERATED {
         permissionDenied     ( 0 ),
         permissionGranted    ( 1 ),
         ...  } OPTIONAL,
      ... }
   ERRORS     {
      systemFailure |
      unexpectedDataValue }
   CODE     local      : 116
   }

lcs-MOLR   OPERATION  ::=  {
   ARGUMENT     SEQUENCE {
      molr-Type             [0] IMPLICIT ENUMERATED {
         locationEstimate    ( 0 ),
         assistanceData      ( 1 ),
         deCipheringKeys     ( 2 ),
         ...  },
      locationMethod        [1] IMPLICIT ENUMERATED {
         msBasedEOTD       ( 0 ),
         msAssistedEOTD    ( 1 ),
         assistedGPS       ( 2 ),
         ... ,
         msBasedOTDOA      ( 3 ) } OPTIONAL,
      lcs-QoS               [2] IMPLICIT SEQUENCE {
         horizontal-accuracy         [0] IMPLICIT OCTET STRING ( SIZE( 1 ) ) OPTIONAL,
         verticalCoordinateRequest   [1] IMPLICIT NULL OPTIONAL,
         vertical-accuracy           [2] IMPLICIT OCTET STRING ( SIZE( 1 ) ) OPTIONAL,
         responseTime                [3] IMPLICIT SEQUENCE {
            responseTimeCategory   ENUMERATED {
               lowdelay         ( 0 ),
               delaytolerant    ( 1 ),
               ...  },
            ... } OPTIONAL,
         extensionContainer          [4] IMPLICIT SEQUENCE {
            privateExtensionList   [0] IMPLICIT SEQUENCE  ( SIZE( 1 .. 10 ) ) OF
               SEQUENCE {
                  extId      MAP-EXTENSION .&extensionId  ( {
                     ,
                     ...} ) ,
                  extType    MAP-EXTENSION .&ExtensionType  ( {
                     ,
                     ...} { @extId   }  )  OPTIONAL} OPTIONAL,
            pcs-Extensions         [1] IMPLICIT SEQUENCE {
               ... } OPTIONAL,
            ... } OPTIONAL,
         ... } OPTIONAL,
      lcsClientExternalID   [3] IMPLICIT SEQUENCE {
         externalAddress      [0] IMPLICIT OCTET STRING ( SIZE( 1 .. 20 ) ) ( SIZE( 1 .. 9 ) )
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,
      mlc-Number            [4] IMPLICIT OCTET STRING ( SIZE( 1 .. 20 ) ) ( SIZE( 1 .. 9 ) )
OPTIONAL,
      gpsAssistanceData     [5] IMPLICIT OCTET STRING ( SIZE( 1 .. 38 ) ) OPTIONAL,
      ... ,
      supportedGADShapes    [6] IMPLICIT BIT STRING {
         ellipsoidPoint  (0 ),
         ellipsoidPointWithUncertaintyCircle  (1 ),
         ellipsoidPointWithUncertaintyEllipse  (2 ),
         polygon  (3 ),
         ellipsoidPointWithAltitude  (4 ),
         ellipsoidPointWithAltitudeAndUncertaintyElipsoid  (5 ),
         ellipsoidArc  (6 )} ( SIZE( 7 .. 16 ) ) OPTIONAL,
      lcsServiceTypeID      [7] IMPLICIT INTEGER ( 0 .. 127 ) OPTIONAL}
   RESULT     SEQUENCE {
      locationEstimate       [0] IMPLICIT OCTET STRING ( SIZE( 1 .. 20 ) ) OPTIONAL,
      decipheringKeys        [1] IMPLICIT OCTET STRING ( SIZE( 15 ) ) OPTIONAL,
      ... ,
      add-LocationEstimate   [2] IMPLICIT OCTET STRING ( SIZE( 1 .. 91 ) ) OPTIONAL}
   ERRORS     {
      systemFailure |
      unexpectedDataValue |
      dataMissing |
      facilityNotSupported |
      ss-SubscriptionViolation |
      positionMethodFailure }
   CODE     local      : 115
   }

lcs-AreaEventRequest OPERATION  ::=  {
   ARGUMENT     SEQUENCE {
      referenceNumber             [0] IMPLICIT OCTET STRING ( SIZE( 1 ) ),
      h-gmlc-address              [1] IMPLICIT OCTET STRING ( SIZE( 5 .. 17 ) ),
      deferredLocationEventType   [3] IMPLICIT BIT STRING {
         msAvailable  (0 ),
         enteringIntoArea  (1 ),
         leavingFromArea  (2 ),
         beingInsideArea  (3 )} ( SIZE( 1 .. 16 ) ),
      areaEventInfo               [4] IMPLICIT SEQUENCE {
         areaDefinition   [0] IMPLICIT SEQUENCE {
            areaList   [0] IMPLICIT SEQUENCE  ( SIZE( 1 .. 10 ) ) OF
               SEQUENCE {
                  areaType             [0] IMPLICIT ENUMERATED {
                     countryCode       ( 0 ),
                     plmnId            ( 1 ),
                     locationAreaId    ( 2 ),
                     routingAreaId     ( 3 ),
                     cellGlobalId      ( 4 ),
                     ...  },
                  areaIdentification   [1] IMPLICIT OCTET STRING ( SIZE( 2 .. 7 ) ),
                  ... },
            ... },
         occurrenceInfo   [1] IMPLICIT ENUMERATED {
            oneTimeEvent         ( 0 ),
            multipleTimeEvent    ( 1 ),
            ...  } OPTIONAL,
         intervalTime     [2] IMPLICIT INTEGER ( 1 .. 32767 ) OPTIONAL,
         ... },
      ... }
   RETURN  RESULT     TRUE
   ERRORS     {
      systemFailure |
      facilityNotSupported |
      unexpectedDataValue }
   CODE     local      : 114
   }
```

```
lcs-AreaEventReport OPERATION  ::=  {
   ARGUMENT     SEQUENCE {
      referenceNumber   [0] IMPLICIT OCTET STRING ( SIZE( 1 ) ),
      h-gmlc-address    [1] IMPLICIT OCTET STRING ( SIZE( 5 .. 17 ) ),
      ... }
   RETURN  RESULT     TRUE
   ERRORS     {
      systemFailure |
      unexpectedDataValue |
      facilityNotSupported }
   CODE     local      : 113
   }

lcs-AreaEventCancellation OPERATION  ::=  {
   ARGUMENT     SEQUENCE {
      referenceNumber   [0] IMPLICIT OCTET STRING ( SIZE( 1 ) ),
      h-gmlc-address    [1] IMPLICIT OCTET STRING ( SIZE( 5 .. 17 ) ),
      ... }
   RETURN  RESULT     TRUE
   ERRORS     {
      systemFailure |
      facilityNotSupported |
      unexpectedDataValue }
   CODE     local      : 112
   }

END
```

Annex B (informative): Change history

Change history						
TSG CN#	**Spec**	**Version**	**CR**	**<Phase>**	**New Version**	**Subject/Comment**
Apr 1999	GSM 04.80	7.0.0				Transferred to 3GPP CN1
CN#03	24.080				3.0.0	Approved at CN#03
CN#06	24.080	3.0.0	001	R99	3.1.0	Addition of LCS operations
CN#07	24.80	3.1.0	002r1	R99	3.2.0	Correction to Location Notification Type and LCS-MOLR errors
CN#08	24.080	3.2.0	004	R99	3.3.0	Correction of definition of Deflected-to number
CN#09	24.080	3.3.0	005r1	R99	3.4.0	Message type: Alignment to 24.007 and 24.008
CN#09	24.080	3.3.0	006	R99	3.4.0	Addition of error type description for PositionMethodFailure
	24.080	3.4.0		R99	3.4.1	Update of Annex A
CN#11	24.080	3.4.1		Rel-4	4.0.0	Release 4 after CN#11
CN#12	24.080	4.0.0	008	Rel-4	4.1.0	Add support in DTAP for all shapes defined in 23.032
CN#12	24.080	4.0.0	009	Rel-4	4.1.0	OTDOA location method to be added
CN#12	24.080	4.0.0	011r1	Rel-4	4.1.0	Addition of the description for Multicall missing from 24.080
CN#14	24.080	4.1.0	014	Rel-4	4.2.0	Message type: completion of alignment to 24.007 and 24.008
Cn~15	24.080	4.2.0	015r1	Rel-5	5.0.0	Introduction of the 'Requestor ID'
CN#16	24.080	5.0.0	023	Rel-5	5.1.0	LCS: error handling if shape not supported by MS
CN#16	24.080	5.0.0	018	Rel-5	5.1.0	LCS: Error handling if wrong method requested in LCS-MOLR
CN#16	24.080	5.0.0	016	Rel-5	5.1.0	LCS: Codeword and Service Type
CN#16	24.080	5.0.0	020	Rel-5	5.1.0	Correction of Object Identifiers for ASN.1 modules
CN#17	24.080	5.1.0	019r3	Rel-5	5.2.0	Compatible upgrade to ASN.1:1997 of 24.080
CN#17	24.080	5.1.0	024	Rel-5	5.2.0	Correction of references to FACILITY information element
CN#18	24.080	5.2.0	026r2	Rel-5	5.3.0	LCS: Exception handling for positioning methods MS Assisted E-OTD and MS Assisted OTDOA
CN#21	24.080	5.3.0	029	Rel-5	5.4.0	Reduce maximum length of 'LCS Requestor ID' and 'LCS Codeword'.
CN#21	24.080	5.3.0		Rel-5	5.4.0	A wrong SS version indicator link corrected in table 2.4.
CN#22	24.080	5.4.0	031r2	Rel-6	6.0.0	Deferred MT-LR Area Event
CN#23	24.080	6.0.0	034	Rel-6	6.1.0	Removal of R-GMLC Address
CN#23	24.080	6.0.0	033r1	Rel-6	6.1.0	MO-LR Service Identity support
CN#26	24.080	6.1.0	042r1	Rel-6	6.2.0	Sequence numbering for SS via PS
CN#26	24.080	6.1.0	038	Rel-6	6.2.0	Correction of setting for timer T(LCSL)

History

Document history		
V6.2.0	December 2004	Publication