

Draft Standard ECMA-QSIG-SMS  
December 2000

**ECMA**

Standardizing Information and Communication Systems

---

---

---

**Private Integrated Services Network  
(PISN) –  
Inter-Exchange Signalling Protocol -  
Short Message Service Supplementary  
Service**

---

---

First Draft – October 2000



**ECMA**

Standardizing Information and Communication Systems

---

---

---

**Private Integrated Services Network  
(PISN) –  
Inter-Exchange Signalling Protocol -  
Short Message Service Supplementary  
Service**

---

---

**(QSIG-SMS)**



### **Brief History**

This Standard is one of a series of ECMA Standards defining services and signalling protocols applicable to Private Integrated Services Digital Networks (PISNs). The series uses ISDN concepts as developed by ITU-T and conforms to the framework of International Standards on Open Systems Interconnection as defined by ISO/ IEC. It has been produced under ITSTC work item XXX and under ETSI work item XXX.

This particular Standard specifies the signalling protocol for use at the Q reference point in support of the Short Message Service supplementary service. The protocol defined in this Standard forms part of the PSS1 protocol (informally known as QSIG).

This Standard is based upon the practical experience of ECMA member companies and the results of their active and continuous participation in the work of ISO/ IEC JTC1, ITU-T, ETSI and other international and national standardization bodies. It represents a pragmatic and widely based consensus.

### **Drafting Statements**

First draft was edited by Lars Kiessling and Georg Mayer, Siemens AG, Germany ([Lars.Kiessling@icn.siemens.de](mailto:Lars.Kiessling@icn.siemens.de), [Georg.Mayer@icn.siemens.de](mailto:Georg.Mayer@icn.siemens.de)).







## Table of contents

<b>1</b>	<b>Scope</b>	<b>1</b>
<b>2</b>	<b>Conformance</b>	<b>1</b>
<b>3</b>	<b>References (normative)</b>	<b>1</b>
<b>4</b>	<b>Definitions</b>	<b>2</b>
4.1	External Definitions	2
4.2	Other Definitions	2
4.2.1	Receiving User	2
4.2.2	Receiving User PINX	2
4.2.3	Sending User	2
4.2.4	Sending User PINX	2
4.2.5	Service Centre	2
4.2.6	Sending User Message Centre	2
4.2.7	Short Message Waiting Data	3
4.2.8	Receiving User Message Centre	3
<b>5</b>	<b>Acronyms</b>	<b>3</b>
<b>6</b>	<b>Signalling Protocol for the support of SS-SMS</b>	<b>3</b>
6.1	SS-SMS Description	3
6.2	SS-SMS operational requirements	3
6.2.1	Provision/Withdrawal	3
	Requirements on a Sending User PINX	3
6.2.2	Requirements on a Sending User Message Centre	3
6.2.3	Requirements on a Service Centre	3
6.2.4	Requirements on a Receiving User PINX	4
6.2.5	Requirements on a Receiving User Message Centre	4
6.3	SS-SMS coding requirements	4
6.3.1	Operations	4
6.3.2	Information Elements	16
6.3.3	Messages	16
6.4	SS-SMS State Definitions	16
6.4.1	States at a Sending User PINX and at the Sending User Message Centre	16
6.4.2	States at a Service Centre	16
6.4.3	States at a Receiving User PINX	17
6.4.4	States at a Receiving User Message Centre	17
6.5	SS-SMS signalling procedures	17
6.5.1	Actions at a Sending User PINX/ Sending User Message Centre	17
6.5.2	Actions at a Sending User Message Centre	20
6.5.3	Actions at a Service Centre	20
6.5.4	Actions at a Receiving User PINX	24
6.5.5	Actions at a Receiving User Message Centre	26
6.6	SS-SMS impact on interworking with public ISDNs	27

6.7	SS-SMS impact on interworking with non-ISDNs	27
6.8	Protocol Interactions between SS-SMS and other supplementary services and ANFs	27
6.8.1	Calling Line Identification Presentation (SS-CLIP)	28
6.8.2	Connected Line Identification Presentation (SS-COLP)	28
6.8.3	Calling/ Connected Line Identification Restriction (SS-CLIR)	28
6.8.4	Calling Name Identification Presentation (SS-CNIP)	28
6.8.5	Calling/ Connected Name Identification Restriction (SS-CNIR)	28
6.8.6	Connected Name Identification Presentation (SS-CONP)	28
6.8.7	Completion of Calls to Busy Subscriber (SS-CCBS)	28
6.8.8	Completion of Calls on No Reply (SS-CCNR)	28
6.8.9	Call Transfer (CT)	28
6.8.10	Call Forwarding Unconditional (SS-CFU)	28
6.8.11	Call Forwarding Busy (SS-CFB)	28
6.8.12	Call Forwarding No Reply (SS-CFNR)	28
6.8.13	Call Deflection (SS-CD)	28
6.8.14	Path Replacement (ANF-PR)	28
6.8.15	Call Offer (SS-CO)	28
6.8.16	Call Intrusion (SS-CI)	28
6.8.17	Do Not Disturb (SS-DND)	28
6.8.18	Do Not Disturb Override (SS-DNDO)	28
6.8.19	Advice of charge (SS-AOC)	28
6.8.20	Recall (SS-RE)	28
6.8.21	Call Interception (ANF-CINT)	28
6.8.22	Transit Counter (ANF-TC)	29
6.8.23	Route Restriction Class (ANF-RRC)	29
6.8.24	Message Waiting Indication (SS-MWI)	29
6.8.25	Cordless Terminal Location Registration (SS-CTLR)	29
6.8.26	Cordless Terminal Mobility Incoming Call (SS-CTMI)	29
6.8.27	Cordless Terminal Mobility Outgoing Call (SS-CTMO)	29
6.8.28	Authentication of a CTM user (SS-CTAT)	29
6.8.29	Authentication of the PISN (SS-CTAN)	29
6.8.30	Private User Mobility Incoming Call (ANF-PUMI)	29
6.8.31	Private User Mobility Outgoing Call (ANF-PUMO)	29
6.8.32	Private User Mobility Registration (SS-PUMR)	29
6.8.33	Common Information (ANF-CMN)	29
6.8.34	Call Priority Interruption (Protection) (SS-CPI(P))	29
6.8.35	Single Step Call Transfer (SS-SSCT)	29
6.8.36	Simple Dialog (SS-SD)	29
6.8.37	Call Identification and Call Linkage (ANF-CIDL)	29
6.9	SS-SMS Parameter values (Timers)	29
	Timer T1	29
	Timer T2	30
	Timer T3	30
	Timer T4	30
	Timer T5	30
	Timer T6	30
	Timer T7	30

Timer T8	30
<b>ANNEX E</b>	<b>31</b>
<b>Description of APDU elements</b>	<b>31</b>
<b>1. Elements in smsSubmit invoke APDU</b>	<b>31</b>
1.1 messageReference	31
1.2 smsSubmitParameter	31
1.2.1 protocolIdentifier	31
1.2.2 validityPeriod	31
1.2.2 statusReportRequest	31
1.2.3 replyPath	31
1.2.4 rejectDuplicates	31
1.3 userData	32
1.3.1 userDataHeader	32
1.3.2 class	33
1.3.3 compressed	34
1.3.4 shortMessageText	34
<b>2. Elements in smsSubmit return result APDU</b>	<b>34</b>
2.1 serviceCentreTimeStamp	34
2.2 protocolIdentifier	34
2.3 userData	34
<b>3. Elements in smsSubmit return error APDU</b>	<b>34</b>
3.1 failureCause	34
3.2 serviceCentreTimeStamp	35
3.3 protocolIdentifier	35
3.4 userData	35
<b>4. Elements in smsDeliver invoke APDU</b>	<b>35</b>
4.1 originatingName	35
4.2 smsDeliverParameter	35
4.2.1 protocolIdentifier	35
4.2.2 serviceCentreTimeStamp	35
4.2.3 priority	35
4.2.4 moreMessagesToSend	35
4.2.5 statusReportIndication	35
4.2.6 replyPath	35
4.3 userData	36
<b>5. Elements in smsDeliver return result APDU</b>	<b>36</b>
5.1 protocolIdentifier	36
5.2 userData	36
<b>6. Elements in smsDeliver return error APDU</b>	<b>36</b>
6.1 failureCause	36

6.2	protocolIdentifier	36
6.3	userData	36
6.4	scAddressSaved	36
<b>7.</b>	<b>Elements in smsStatusReport invoke APDU</b>	<b>36</b>
7.1	messageReference	36
7.2	serviceCentreTimeStamp	37
7.3	dischargeTime	37
7.4	recipientAddress	37
7.5	recipientName	37
7.6	status	37
7.7	priority	37
7.8	moreMessagesToSend	37
7.9	statusReportQualifier	37
7.10	protocolIdentifier	37
7.11	userData	37
<b>8.</b>	<b>Elements in smsStatusReport return result APDU</b>	<b>37</b>
8.1	protocolIdentifier	37
8.2	userData	37
<b>9.</b>	<b>Elements in smsStatusReport return error APDU</b>	<b>38</b>
9.1	failureCause	38
9.2	protocolIdentifier	38
9.3	userData	38
9.4	scAddressSaved	38
<b>10.</b>	<b>Elements in smsCommand invoke APDU</b>	<b>38</b>
10.1	messageReference	38
10.2	messageNumber	38
10.2	protocolIdentifier	38
10.3	commandType	38
10.4	commandData	38
10.5	statusReportRequest	38
<b>11.</b>	<b>Elements in smsCommand return result APDU</b>	<b>38</b>
11.1	serviceCentreTimeStamp	38
11.2	protocolIdentifier	38
11.3	userData	39
<b>12.</b>	<b>Elements in smsCommand return error APDU</b>	<b>39</b>
12.1	failureCause	39
12.2.	serviceCentreTimeStamp	39
12.3	protocolIdentifier	39
12.4	userData	39
1.8	Originating-Address	39
1.11	Receiving User	39
1.14	Sending User	39

<b>ANNEX F</b>	<b>40</b>
<b>1. User Data Headers</b>	<b>40</b>
1.1 Concatenated Short Message	40
1.2 SMSC Control Parameters	40
1.3 Application port addressing	40
1.4 User Data Header Source Indicator	40
1.5 Wireless Control Message Protocol	40
<b>2. Special Short Messages</b>	<b>41</b>
2.1 SMS and Internet Electronic Mail interworking	41
2.2 SMS Compression	41



## 1 Scope

This Standard specifies the signalling protocol for the support of the Short Message Service supplementary service (SS-SMS) at the Q reference point between Private Integrated Services Network Exchanges (PINXs) connected together within a Private Integrated Services Network (PISN).

The Short Message Service is a supplementary service which permits a served user to send a Short Message of limited size to another user in the same the same PISN or another network.

The Q reference point is defined in ISO/IEC 11579-1.

Service specifications are produced in three stages and according to the method specified in ETS 300 387. This Standard contains the stage 3 specification for the Q reference point and satisfies the requirements identified by the stage 1 and stage 2 specifications in ECMA-XXX.

Editor's Note:

The Standard number has to be replaced by the SS-SMS Stage1/2 number.

The signalling protocol for SS-SMS operates on top of the signalling protocol for basic circuit switched call control, as specified in ECMA-143 and uses certain aspects of the generic procedures for the control of supplementary services specified in ECMA-165.

This Standard also specifies additional signalling protocol requirements for the support of interactions at the Q reference point between SS-SMS and other supplementary services and ANFs.

This Standard is applicable to PINXs which can be interconnected to form a PISN.

## 2 Conformance

In order to conform to this Standard, a PINX shall satisfy the requirements identified in the Protocol Implementation Conformance Statement (PICS) proforma in Annex A.

## 3 References (normative)

The following standards contain provisions which, through references in this text, constitute provisions of this Standard. All standards are subject to revision, and parties to agreements based on this Standard are encouraged to investigate the possibility of applying the most recent editions of the standards indicated below.

In the case of references to ECMA Standards that are aligned with ISO/IEC International Standards, the number of the appropriate ISO/IEC International Standard is given in brackets after the ECMA reference.

ECMA-142	Private Integrated Services Network – Circuit-mode 64 kbit/s Bearer Services – Service Description, Functional Capabilities and Information Flows (International Standard ISO/IEC 11574)
ISO/IEC 11579-1	Information technology – Telecommunications and information exchange between systems – Private Integrated Services Network – Part 1: Reference configuration for PISN Exchanges (PINX)
ETSI TS 100 900	Digital cellular telecommunications systems (Phase 2+); Alphabets and language-specific information (1998)(GSM 03.38)
ETSI TS 100 901	Digital cellular telecommunications systems (Phase 2+); Technical realization of the Short Message Service (SMS) (1998) (GSM 03.40)
ETSI TS 100 942	Digital cellular telecommunications systems (Phase 2+); Point-to-Point (PP) Short Message Service (SMS) support on mobile radio interface (1999)(GSM 04.11)
ETSI TS 100 974	Digital cellular telecommunications systems (Phase 2+); Mobile Application Part (MAP) specification (1999)(GSM 09.02)
ETSI TS 101 032	Digital cellular telecommunications systems (Phase 2+); Compression algorithm for text messaging services (1998) (GSM 03.42)

ETS 300 387	Private Telecommunication Network (PTN); Method for the specification of basic and supplementary services (1994)
ITU-T Rec. I.112	Vocabulary of terms for ISDNs (1993)
ITU-T Rec. I.210	Principles of telecommunication services supported by an ISDN and the means to describe them (1993)
ITU-T Rec. Z.100	Specification and description language (1993)

## 4 Definitions

For the purpose of this Standard the following definitions apply.

### 4.1 External Definitions

This Standard uses the following terms defined in other documents:

- Application Protocol Data Unit (APDU) (ECMA-165)
- Call, Basic Call (ECMA-165)
- Call Independent (ECMA-165)
- End PINX (ECMA-165)
- Private Integrated Services Network (PISN) (ISO/IEC 11579-1)
- Private Integrated Services Network Exchange (PINX) (ISO/IEC 11579-1)
- Short Message ECMA-XXX
- Status Report ECMA-XXX
- Command ECMA-XXX

Editor's Note:

Replace Standard number with SS-SMS Stage 1&2 Standard number.

- Signalling (ITU-T Rec. I.112)
- Supplementary Service (ITU-T Rec. I.210)
- User (ECMA-142)

### 4.2 Other Definitions

#### 4.2.1 Receiving User

The user receiving Short Messages.

#### 4.2.2 Receiving User PINX

The Receiving User PINX is the PINX serving the Receiving User.

#### 4.2.3 Sending User

The user sending Short Messages or Commands and receiving Status Reports.

#### 4.2.4 Sending User PINX

The Sending User PINX is the PINX serving the Sending User.

#### 4.2.5 Service Centre

The Service Centre is a store and forward unit receiving Short Messages and Commands from a Sending User PINX or Sending User Message Centre and forwarding the Short Messages to the Receiving User PINX.

#### 4.2.6 Sending User Message Centre

The Message Centre serving the Sending User.

#### 4.2.7 Short Message Waiting Data

The ability of a Sending User PINX to store the address of a Service Centre and to inform the Service Centre whenever a Receiving User becomes available again or has memory available again.

#### 4.2.8 Receiving User Message Centre

The Message Centre serving the Receiving User.

## 5 Acronyms

APDU	Application Protocol Data Unit
ASN.1	Abstract Syntax Notation One
ISDN	Integrated Services Digital Network
PICS	Protocol Implementation Conformance Statement
PINX	Private Integrated Services Network Exchange
PISN	Private Integrated Services Network
SDL	Specification and Description Language
SM	Short Message
SMWD	Short Message Waiting Data
SS	Supplementary Service
SS-SMS	Supplementary Service Short Message Service
TE	Terminal Equipment

## 6 Signalling Protocol for the support of SS-SMS

### 6.1 SS-SMS Description

Short Message Service is a supplementary service which is offered to a user in a PISN and which enables the user to send and receive Short Messages to and from another user in a PISN or in another (e.g. GSM) network.

The PISN transfers the Short Message from the Sending User to an SC and from the SC to the Receiving User.

### 6.2 SS-SMS operational requirements

#### 6.2.1 Provision/Withdrawal

Provision and withdrawal shall be in accordance with 6.2.1 from ECMA-XXX.

Editors note:

The Standard number has to be replaced with the number ascribed to the Standard SS-SMS.

#### 6.2.2 Requirements on a Sending User PINX

Generic procedures for the call independent control (connection oriented) of supplementary services, as specified in ECMA-165 for an Originating-PINX, shall apply.

#### 6.2.3 Requirements on a Sending User Message Centre

Generic procedures for the call independent control (connection oriented) of supplementary services, as specified in ECMA-165 for an Originating-PINX, shall apply.

#### 6.2.4 Requirements on a Service Centre

Generic procedures for the call independent control (connection oriented) of supplementary services, as specified in ECMA-165 for an Originating-PINX and for a Terminating-PINX, shall apply.

### 6.2.5 Requirements on a Receiving User PINX

Generic procedures for the call independent control (connection oriented) of supplementary services, as specified in ECMA-165 for an Terminating-PINX , shall apply

### 6.2.6 Requirements on a Receiving User Message Centre

Generic procedures for the call independent control (connection oriented) of supplementary services, as specified in ECMA-165 for an Terminating-PINX, shall apply.

## 6.3 SS-SMS coding requirements

### 6.3.1 Operations

The following operations defined in Abstract Syntax Notation One (ASN.1) in table 1 shall apply.

**Table 1 - Operations in Support of SS-SMS**

```
DEFINITIONS EXPLICIT TAGS ::=
BEGIN
IMPORTS
    OPERATION,
    ERROR
FROM Remote-Operation-Notation
    {joint-iso-ccitt(2) remote-operations(4) notation(0)}
    Extension
FROM Manufacturer-specific-service-extension-definition
    {iso(1) standard(0) pss1-generic-procedures(11582) msi-definition(0)}
    Name
FROM Name-Operations
    {iso(1) standard(0) pss1-name(13868) name-operations(0)}
    supplementaryServiceInteractionNotAllowed
FROM General-Error-List
    {ccitt(0) recommendation (0) q950 general-error-list(1)}
    PartyNumber
FROM Addressing-Data-Elements
    {iso(1) standard(0) pss1-generic-procedures(11582) addressing-data-elements (9)};

--TYPE DEFINITIONS FOR SMS OPERATIONS FOLLOW
SmsDeliver ::= OPERATION --sent from the Service Centre
    ARGUMENT          SmsDeliverArg
    RESULT             SmsDeliverRes
    ERRORS             {smsDeliverError,
                        unspecified}
SmsSubmit ::= OPERATION --sent from the TE
    ARGUMENT          SmsSubmitArg
    RESULT             SmsSubmitRes
```

	ERRORS	{smsSubmitError, unspecified}	
smsStatusReport ::=	OPERATION		
	ARGUMENT	SmsStatusReportArg	
	RESULT	SmsStatusReportRes	
	ERRORS	{smsStatusReportError, unspecified}	
smsCommand ::=	OPERATION		
	ARGUMENT	SmsCommandArg	
	RESULT	SmsCommandRes	
	ERRORS	{smsCommandError, unspecified}	
scAlert ::=	OPERATION		
	ARGUMENT	ScAlertArg	
	RESULT	DummyRes	
	ERRORS	{unspecified}	
<b>--TYPE DEFINITIONS FOR SMS DATA TYPES FOLLOW</b>			
SmsSubmitArg ::=	SEQUENCE	{	
destinationAddress		PartyNumber,	
originatingAddress		PartyNumber,	
message Reference		IMPLICIT MessageReference,	
smSubmitParameter		IMPLICIT SmSubmitParameter,	
userData		IMPLICIT UserData,	
argumentExtension		ArgumentExtension	OPTIONAL}
SmsSubmitRes ::=	SEQUENCE	{	
serviceCentreTimeStamp		IMPLICIT ServiceCentreTimeStamp,	
protocolIdentifier		[2] IMPLICIT ProtocolIdentifier	OPTIONAL,
userData		[3] IMPLICIT UserData	OPTIONAL,
argumentExtension		ArgumentExtension	OPTIONAL}
SmsDeliverArg ::=	SEQUENCE	{	
originatingAddress		PartyNumber,	
destinationAddress		PartyNumber,	
originatingName		Name	OPTIONAL,

smDeliverParameter	[5] IMPLICIT SmDeliverParameter,	
userData	IMPLICIT UserData,	
argumentExtension	ArgumentExtension	OPTIONAL}
SmsDeliverRes ::= SEQUENCE	{	
smsDeliverResponseChoice	IMPLICIT SmsDeliverResChoice,	
argumentExtension	ArgumentExtension	OPTIONAL}
SmsStatusReportArg ::= SEQUENCE	{	
messageReference	IMPLICIT MessageReference,--previously submitted SUBMIT	
serviceCentreTimeStamp	IMPLICIT ServiceCentreTimeStamp,	
dischargeTime	IMPLICIT DischargeTime,	
recipientAddress	PartyNumber,	
recipientName	Name	OPTIONAL,
destinationAddress	PartyNumber,	
status	[5] IMPLICIT Status,	
priority	[6] IMPLICIT BOOLEAN DEFAULT FALSE,	
moreMessagesToSend	[8] IMPLICIT BOOLEAN DEFAULT FALSE,	
statusReportQualifier	[9] IMPLICIT BOOLEAN DEFAULT FALSE,	
protocolIdentifier	[10] IMPLICIT ProtocolIdentifier	OPTIONAL,
userData	[11] IMPLICIT UserData	OPTIONAL,
argumentExtension	ArgumentExtension	OPTIONAL}
SmsStatusReportRes ::= SEQUENCE	{	
smsStatusReportResponseChoice	IMPLICIT SmsStatusReportResponseChoice,	
argumentExtension	ArgumentExtension	OPTIONAL}
SmsCommandArg ::= SEQUENCE	{	
destinationAddress	PartyNumber,	
messageReference	IMPLICIT MessageReference,	
messageNumber	IMPLICIT MessageReference, --which SM in the SC to --operate on, only necessary for Command that works on a --specific Short Message, else it shall be ignored	
protocolIdentifier	IMPLICIT ProtocolIdentifier,	
commandType	IMPLICIT CommandType,	

commandData	[2] IMPLICIT CommandData	OPTIONAL,
statusReportRequest	[3] IMPLICIT BOOLEAN DEFAULT FALSE	OPTIONAL,
argumentExtension	ArgumentExtension	OPTIONAL}
SmsCommandRes ::= SEQUENCE	{	
serviceCentreTimeStamp	IMPLICIT ServiceCentreTimeStamp,	
protocolIdentifier	[2] IMPLICIT ProtocolIdentifier	OPTIONAL,
userData	[3] IMPLICIT UserData	OPTIONAL,
argumentExtension	ArgumentExtension	OPTIONAL}
ScAlertArg ::= SEQUENCE	{	
originatingAddress	PartyNumber,	
argumentExtension	ArgumentExtension	OPTIONAL}
DummyRes ::= CHOICE{		
null	NULL,	
extension	[1] IMPLICIT Extension,	
multipleExtension	[2] IMPLICIT SEQUENCE OF Extension}	
SmSubmitParameter ::= SEQUENCE	{	
protocolIdentifier	IMPLICIT ProtocolIdentifier,	
validityPeriod	ValidityPeriod	OPTIONAL,
statusReportRequest	[4] IMPLICIT BOOLEAN DEFAULT FALSE,	
replyPath	[5] IMPLICIT BOOLEAN DEFAULT FALSE,	
rejectDuplicates	[6] IMPLICIT BOOLEAN DEFAULT FALSE}	
SmDeliverParameter ::= SEQUENCE	{	
protocolIdentifier	IMPLICIT ProtocolIdentifier,	
serviceCentreTimeStamp	IMPLICIT ServiceCentreTimeStamp,	
priority	[0] IMPLICIT BOOLEAN DEFAULT FALSE,	
moreMessagesToSend	[1] IMPLICIT BOOLEAN DEFAULT FALSE,	
statusReportIndication	[2] IMPLICIT BOOLEAN DEFAULT FALSE,	
replyPath	[3] IMPLICIT BOOLEAN DEFAULT FALSE}	
SmsDeliverResChoice ::= CHOICE	{	
Null,		
[0] IMPLICIT ProtocolIdentifier,		
[1] IMPLICIT UserData,		

```
[2] IMPLICIT SEQUENCE { ProtocolIdentifier,
                        UserData} }
```

```
SmsStatusReportResponseChoice ::= CHOICE {
    Null,
    [0] IMPLICIT ProtocolIdentifier,
    [1] IMPLICIT UserData,
    [2] IMPLICIT SEQUENCE { ProtocolIdentifier,
                            UserData} }
```

```
MessageReference ::= INTEGER(0..255)
```

```
ArgumentExtension ::= CHOICE{
    [0] IMPLICIT Extension,
    [1] IMPLICIT SEQUENCE OF Extension}
```

```
ProtocolIdentifier ::= INTEGER {
    noIw (0),
    --the following protocolIdentifier settings do not indicate interworking but the protocol between the SM
    --entities
    telex (1),
    group3Telefax (2),
    group4Telefax (3),
    voiceTelephone (4),
    eRMES (5),
    nationalPagingSystem (6),
    videotex (7),
    teletexUnspecifiedCarrier (8),
    teletexPSPDN (9),
    teletexCSPDN (10),
    teletexAnalogPSTN (11),
    teletexDigitalISDN (12),
    uci (13),
    messageHandlingFacility (16),
    x400MessageHandlingSystem (17),
    internetElectronicMail (18),
    gsmMobileStation (31),
    --the following protocolIdentifiers indicate intworking
```

iwImplicit	(32),
iwTelex	(33),
iwTelefaxGroup3	(34),
iwTelefaxGroup4	(35),
iwVoiceTelephone	(36),
iwERMES	(37),
iwNationalPagingSystem	(38),
iwVideotex	(39),
iwTeletexCarrierUnspecified	(40),
iwTeletexPSPDN	(41),
iwTeletexCSPDN	(42),
iwTeletexAnaloguePSTN	(43),
iwTeletexDigitalISDN	(44),
iwUCI	(45),
iwMessageHandlingFacility	(48),
iwX400MessageHandlingSystem	(49),
iwInternetElectronicMail	(50),
iwGSMMobileStation	(63),
shortMessageType0	(64),
replaceShortMessageType1	(65),
replaceShortMessageType2	(66),
replaceShortMessageType3	(67),
replaceShortMessageType4	(68),
replaceShortMessageType5	(69),
replaceShortMessageType6	(70),
replaceShortMessageType7	(71),
returnCallMessage	(95),
Editor's note:	
Can codepoint 127 be used in a PISN?	
simDataDownload	(127)} (0..127)
ServiceCentreTimeStamp	::= GeneralizedTime(Size(12..19)) --this date and time representation follows ISO 8601
DischargeTime	::= GeneralizedTime(Size(12..19)) --this date and time representation follows ISO 8601
ValidityPeriod	::= CHOICE{

```

validityPeriodRel    [0] IMPLICIT ValidityPeriodRel,
validityPeriodAbs    [1] IMPLICIT ValidityPeriodAbs,
validityPeriodEnh    [2] IMPLICIT ValidityPeriodEnh}

ValidityPeriodAbs ::= GeneralizedTime(Size(12..19))
                    --this date and time representation follows ISO 8601

ValidityPeriodRel ::= INTEGER(0..255)
                    --coding rules:
                    --   value           | ValidityPeriodValue
                    --   0 to 143       | (value+1) x 5 minutes (i.e. 5 mins. intervals up to 12 hours)
                    --   144 to 167     | 12 hours + ((value - 143) x 30 minutes)
                    --   168 to 196     | (value-166) x 1 day
                    --   197 to 255     | (value-192) x 1 weeks

ValidityPeriodEnh ::= SEQUENCE{
    singleShotSM      IMPLICIT BOOLEAN DEFAULT FALSE,
    enhancedVP        IMPLICIT EnhancedVP}

EnhanceVP ::= CHOICE{
    validityPeriodRel [0] IMPLICIT ValidityPeriodRel,
    validityPeriodSec [1] IMPLICIT INTEGER(0..255)      --relative time in seconds, time value
                    --of 0 seconds is undefined
    validityPeriodSemi [2] IMPLICIT ValidityPeriodSemi}

ValidityPeriodSemi ::= OCTET STRING(3)
                    --Validity Period is relative in semi-octet representation, see ETSI TS 100 901, section 9.1.2.3
                    --and section 9.2.3.11

```

Editor's Note:

Is it possible to define an OCTET STRING like that? There are three octets which are divided in six semi-octets. Each semi-octet represents a digit.

Octet1	h1 h1 h1 h1 h2 h2 h2 h2	hence	0010 0100 = 24 hours
Octet2	m1 m1 m1 m1 m2 m2 m2 m2		0011 0000 = 30 minutes
Octet3	s1 s1 s1 s1 s2 s2 s2 s2		0101 0111 = 57 seconds

```

UserData ::= SEQUENCE{
    userDataHeader    [0] IMPLICIT UserDataHeader          OPTIONAL,
    class              [1] IMPLICIT INTEGER (0..3)          OPTIONAL,
    compressed         [2] IMPLICIT BOOLEAN DEFAULT FALSE,

```

shortMessageText	ShortMessageText}
ShortMessageText ::=	CHOICE{
iA5Coded	[0] IMPLICIT IA5String(0..160),
octetCoded	[1] IMPLICIT OCTET STRING(0..140),
uniCoded	[2] IMPLICIT SmsDataBMPString,
compressedCoded	[3] IMPLICIT OCTET STRING(0..140)}
	--compressedCoded data contain Compressed Data Header and
	--Footer according to GSM 03.42
Editor's note: should Data Coding Scheme be defined as an Integer and the text itself as an Octet String which would allow extensibility of the DCS?	
SmsDataBMPString ::=	OCTET STRING(SIZE(0..140))
	--shall be used according to ISO/IEC 10646-1
	-- coded as a BMP String according to
	-- ITU-T Rec. X.690
Status ::=	INTEGER{
	--SM Transaction completed
smReceivedBySME	(0),
smForwardedButSCUnableToConfirmDelivery	(1),
smReplacedByTheSC	(2),
	--errors below are temporary, SC still trying to transfer SM
tempCongestion	(32),
tempSMEBusy	(33),
tempNoResponseFromSME	(34),
tempServiceRejected	(35),
tempQualityOfServiceNotAvailable	(36),
tempErrorInSME	(37),
	--errors below are permanent, SC is not making any more transfer attempts
remoteProcedureError	(64),
incompatibleDestination	(65),
connectionRejectedBySME	(66),
notObtainable	(67),
qualityOfServiceNotAvailable	(68),
noInterworkingAvailable	(69),
iwValidityPeriodExpired	(70),
smDeletedByOriginatingSME	(71),
smDeletedBySCAdministration	(72),
smDoesNotExist	(73),
	--errors below are temporary, SC is not making any more transfer attempts

congestion	(96),
sMEBusy	(97),
noResponseFromSME	(98),
serviceRejected	(99),
qualityOfServiceNotAvailable	(100),
errorInSME	(101)} (0..255)
CommandType ::= INTEGER{	
Enquiry	(0), --SRR=1
CancelSRR	(1), --SRR=0
DeletePreviouslySubmittedSM	(2), --SRR=0
EnableSRRrelatingToPreviouslySubmittedSM	(3)} (0..255) --SRR=0
Editors Note:	
Command Data contains the data necessary for executing the command. There are no further details given in GSM 03.40.	
--CommandData ::= OCTET STRING	
Editor's Note: Do we need all the following Error-Causes?	
FailureCause ::=INTEGER{	
--ProtocolIdentifier-Errors	
telematicInterworkingNotSupported	(128),
shortMessageType0NotSupported	(129),
canNotReplaceShortMessage	(130),
unspecifiedProtocolIdentifierError	(143),
--Data-Coding-Scheme-Errors	
alphabetNotSupported	(144),
messageClassNotSupported	(145),
unspecifiedDcsError	(159),
--Command-Errors	
commandCanNotBeActioned	(160),
commandUnsupported	(161),
unspecifiedCommandError	(175),
--other Errors	
pduNotSupported	(176),
--SC Errors	
scBusy	(192),
noScSubscription	(193),

scSystemFailure	(194),
invalidSmeAddress	(195),
destinationSmeBarred	(196),
smRejectedDuplicateSm	(197),
validityPeriodFormatNotSupported	(198),
validityPeriodNotSupported	(199),
--MS Errors	
simSmsStorageFull	(208),
noSmsStorageCapabilityinSIM	(209),
errorInTE	(210),
memoryCapacityExceeded	(211),
simApplicationToolkitBusy	(212),
simDataDownloadError	(213),
--	
unspecifiedErrorCause	(255)
-- all other values are reserved}	(0..255)
UserDataHeader ::= SEQUENCE OF UserDataHeaderChoice	
UserDataHeaderChoice ::= CHOICE{	
smscControlParameterHeader	[0] IMPLICIT SmscControlParameterHeader,
concatenated8BitSMHeader	[1] IMPLICIT Concatenated8BitSMHeader,
concatenated16BitSMHeader	[2] IMPLICIT Concatenated16BitSMHeader,
applicationPort8BitHeader	[3] IMPLICIT ApplicationPort8BitHeader,
applicationPort16BitHeader	[4] IMPLICIT ApplicationPort16BitHeader,
dataHeaderSourceIndicator	[5] IMPLICIT DataHeaderSourceIndicator,
wirelessControlHeader	[6] IMPLICIT WirelessControlHeader,
genericUserValue	[99] IMPLICIT GenericUserValue}
Editor's note: The UserDataHeaderChoice should be extensible – due to ASN.1 a CHOICE can be extended at any time – if existing implementations have problems with this CHOICE-extensibility, the ArgumentExtension should become a part of the CHOICE.	
SmscControlParameterHeader ::= BIT STRING {	
SRforTransactionCompleted (0), SRforPermanentError (1), SRforTempErrorSCnotTrying (2),	
SRforTempErrorSCstillTrying (3), CancelSRRforConcatenatedSM(6), IncludeOrigUDHintoSR	
(7)}(SIZE(1..8))	
Editor's Note	
CancelSRRforConcatenatedSM: a SR generated by this Short Message, due to a permanent error or last temporary error, cancels the SRR of the rest of the Short Messages in a Concatenated Short Message. This feature can only	

be used where the SC is aware of the segmentation and is therefore an implementation matter. Description of the SmscControlParameters in ETSI TS 100 901, section 9.2.3.24.5.

Bit 4 and 5 are reserved and not included in the BIT STRING. Is that possible or do they have to be included?

```
Concatenated8BitSMHeader ::= SEQUENCE{
    concatenated8BitSMReferenceNumber      IMPLICIT INTEGER(0..255),
    maximumNumberOf8BitSMInConcatenatedSM IMPLICIT INTEGER(0..255),
    sequenceNumberOf8BitSM                IMPLICIT INTEGER(0..255)}
```

```
Concatenated16BitSMHeader ::= SEQUENCE{
    concatenated16BitSMReferenceNumber      IMPLICIT INTEGER(0..65536),
    maximumNumberOf16BitSMInConcatenatedSM IMPLICIT INTEGER(0..255),
    sequenceNumberOf16BitSM                IMPLICIT INTEGER(0..255)}
```

```
ApplicationPort8BitHeader ::= SEQUENCE{
    destination8BitPort      IMPLICIT INTEGER(0..255),
    originator8BitPort      IMPLICIT INTEGER(0..255)}
--      Value (port number)  Meaning
--      0 – 239              Reserved
--      240 – 255           Available for allocations by applications
```

```
ApplicatonPort16BitHeader ::= SEQUENCE{
    destination16BitPort      IMPLICIT INTEGER(0..65536),
    originator16BitPort      IMPLICIT INTEGER(0..65536)}
--      Value (port number)  Meaning
--      0 – 15999            As allocated by IANA (http://www.IANA.com)
--      16000 – 16999       Available for allocation by applications
--      17000 – 65535       Reserved
```

```
DataHeaderSourceIndicator ::= INTEGER{
    OriginalSender      01,      --valid in case of Status Report
    OriginalReceiver   02,      --valid in case of Status Report
    SMSC               03}(0..255) --can occur in any message or report
```

```
WirelessControlHeader ::= OCTET STRING
```

```
GenericUserValue ::= SEQUENCE{
    parameterValue      IMPLICIT INTEGER(0..255),
    genericUserData     OCTET STRING}
```

SmsDeliverError ::=	ERROR		
	PARAMETER SEQUENCE{		
failureCause	FailureCause,		
protocolIdentifier	[0] IMPLICIT ProtocolIdentifier	OPTIONAL,	
userData	[1] IMPLICIT userData	OPTIONAL,	
scAddressSaved	[2] IMPLICIT BOOLEAN DEFAULT FALSE}		
SmsSubmitError	ERROR		
	PARAMETER SEQUENCE{		
failureCause	FailureCause,		
serviceCentreTimeStamp	IMPLICIT ServiceCentreTimeStamp,		
protocolIdentifier	[0] IMPLICIT ProtocolIdentifier	OPTIONAL,	
userData	[1] IMPLICIT UserData	OPTIONAL}	
SmsStatusReportError	ERROR		
	PARAMETER SEQUENCE{		
FailureCause	FailureCause,		
protocolIdentifier	[0] IMPLICIT ProtocolIdentifier	OPTIONAL,	
userData	[1] IMPLICIT userData	OPTIONAL,	
ScAddressSaved	[2] IMPLICIT BOOLEAN DEFAULT FALSE}		
SmsCommandError	ERROR		
	PARAMETER SEQUENCE{		
failureCause	FailureCause,		
serviceCentreTimeStamp	IMPLICIT ServiceCentreTimeStamp,		
protocolIdentifier	[0] IMPLICIT ProtocolIdentifier	OPTIONAL,	
userData	[1] IMPLICIT UserData	OPTIONAL}	
smsSubmit	SmsSubmit ::= localvalue {(\$a)}		
smsDeliver	SmsDeliver ::= localvalue {(\$b)}		
smsStatusReport	smsStatusReport ::= localvalue {(\$c)}		
smsCommand	smsCommand ::= localvalue {(\$d)}		
scAlert	scAlert ::= localvalue {(\$e)}		
smsDeliverError	SmsDeliverError ::= localvalue{(\$A)}		
smsSubmitError	SmsSubmitError ::= localvalue {(\$B)}		
smsStatusReportError	SmsStatusReportError ::= localvalue{(\$C)}		
smsCommandError	SmsCommandError ::= localvalue {(\$D)}		

Editors Note:

The Operation- and Error-Values have to be replaced by valid values

## **6.3.2 Information Elements**

### **6.3.2.1 Facility information element**

The operations defined in 6.3.1 for the support of SS-SMS shall be coded in the Facility information element in accordance with ECMA-165.

When conveying the invoke APDU of the operations defined in 6.3.1 the destinationEntity data element of the NFE shall contain value endPINX. The Interpretation APDU in the Facility information element shall be omitted or have the value “rejectAnyUnrecognizedInvokeAPDU (0)”.

### **6.3.2.2 Other information elements**

Any other information elements shall be coded in accordance with ECMA-143.

## **6.3.3 Messages**

The Facility information element shall be conveyed in messages as specified in clause 10 of ECMA-165.

## **6.4 SS-SMS State Definitions**

### **6.4.1 States at a Sending User PINX and at the Sending User Message Centre**

The procedures at the Sending User PINX/ Sending User Message Centre are written in terms of the following conceptual states existing within the SS-SMS control entity in that Sending User PINX/ Sending User Message Centre in association with a particular send Short Message request or send Command request from the Sending User.

#### **6.4.1.1 SMS-Send-Idle**

SS-SMS is not operating.

#### **6.4.1.2 SMS-Send-Submit-Wait**

A smsSubmit invoke APDU has been sent to the Service Centre. The Sending User PINX/Sending User Message Centre is waiting for a response.

#### **6.4.1.3 SMS-Send-Command-Wait**

The Sending User-PINX/Sending User Message Centre has received a command request from the Sending User, has sent a smsCommand invoke APDU to the Service Centre and is waiting for receipt of a smsCommand return result, return error or reject APDU.

### **6.4.2 States at a Service Centre**

The procedures at the Service Centre are written in terms of the following conceptual states existing within the SS-SMS control entity in that Service Centre.

#### **6.4.2.1 SMS-SC-Idle**

SMS is not operating

#### **6.4.2.2 SMS-SC-Deliver-Wait**

The Service Centre has received an smsSubmit invoke APDU from the Sending User PINX/Sending User Message Centre, saved the Short Message, sent a smsDeliver invoke APDU to the Receiving User PINX and is waiting for receipt of a smsDeliver return result, return error or reject APDU.

#### **6.4.2.3 SMS-SC-SR-Wait**

The Service Centre has sent an smsStatusReport invoke APDU to the Sending User PINX/Sending User Message Centre and is waiting for receipt of a smsStatusReport return result, return error or reject APDU.

*NOTE*

*This state may exist parallel and independently of the normal states for Short Messages.*

**6.4.2.4 SMS-SC-SR-Idle**

The Service Centre is waiting for an internal request to send a Status Report.

*NOTE*

*This state may exist parallel and independently of the normal states for Short Messages.*

**6.4.2.5 SMS-SC-Await-Alert**

The Service Centre has received an smsDeliver return error APDU with failureCause “memoryCapacityExceeded” or “simSmsStorageFull” or with an additional Cause Information Element and is now waiting for receipt of a scAlert invoke APDU from the Receiving User PINX.

**6.4.3 States at a Receiving User PINX**

The procedures at the Receiving User PINX are written in terms of the following conceptual states existing within the SS-SMS control entity in that PINX.

**6.4.3.1 SMS-Rec-User-case-Idle**

SMS is not operating.

**6.4.3.2 SMS-Rec-User-case-AlertWait**

The Receiving User PINX unsuccessfully attempted to deliver a Short Message to a terminal and is waiting for an internal indication that the Receiving User is available again for further smsDeliver invoke APDUs.

**6.4.3.3 SMS-Rec-User-case-AlertResp**

The Receiving User PINX has sent an scAlert invoke APDU to the Service Centre and is now waiting for receipt of a scAlert return result, return error or reject APDU from the Service Centre.

**6.4.3.4 SMS-Rec-MC-case-Idle**

The Receiving User PINX has received APDUs from the Service Centre and forwarded them to the Receiving User Message Centre or has received APDUs from the Receiving User Message Centre and forwarded them to the Service Centre.

**6.4.4 States at a Receiving User Message Centre**

The procedures at the Sending User Message Centre are written in terms of the following conceptual states existing within the SS-SMS control entity in that Message Centre.

**6.4.4.1 SMS-Rec-MC-Idle**

SMS is not operating.

**6.4.4.2 SMS-Rec-MC-AWait**

The Message Centre has received a smsDeliver reject APDU and is waiting for the Receiving User to become available or having memory available again.

**6.4.4.3 SMS-Rec-MC-AResp**

The Message Centre has sent a scAlert invoke APDU and is waiting for receipt of a scAlert return result, return error or reject APDU from the Receiving User PINX.

**6.5 SS-SMS signalling procedures**

References in this clause to protocol states refer to basic call protocol states defined in ECMA-143.

**6.5.1 Actions at a Sending User PINX/ Sending User Message Centre**

*NOTE*

*The APDU elements are described in annex E.*

All invoke, return error, return result and reject APDUs shall be transported using the Call Reference of an already existing call independent or call related signalling connection. If no such connection exists, and no call has to be established, the Sending User PINX/Sending User Message Centre shall set up a call independent signalling connection in accordance with the procedures described in 7.3 in ECMA-165.

The Sending User PINX/Sending User Message Centre is responsible for the clearing of this call independent signalling connection.

### **6.5.1.1 Normal Procedures**

#### **6.5.1.1.1 Short Message**

On request of the Sending User to send a Short Message the Sending User PINX/ Sending User Message Centre shall

- 1) check if the Sending User is permitted to use the SS-SMS; if so
- 2) generate a smsSubmit invoke APDU, based on the Short Message elements received from the Sending User which shall include the following mandatory elements:
  - the Party Number of the Receiving User in element destinationAddress,
  - the Party Number of the Sending User in element originating Address,
  - a Message Reference in element messageReference which is allocated by the Sending User PINX for each new Short Message that is sent (see Annex E for further details)
  - Short Message specific Parameters in element smsSubmitParameters (see Annex E for further details)
  - the Short Message Text and related information in element userData (see Annex E for further details)
- 3) send the smsSubmit invoke APDU to the Service Centre,
- 4) start timer T1 and enter state SMS-Send-Submit-wait.

On receipt in state SMS-Send-Submit-wait of a smsSubmit return result APDU the Sending User PINX shall

- 1) stop timer T1;
- 2) send an indication to the Sending User that the submission of the Short Message was successful and
- 3) enter state SMS-Send-Idle.

#### **6.5.1.1.2 Command**

On request of the Sending User to send a Command the Sending User PINX/ Sending User Message Centre shall

- 1) check if the Sending User is permitted to use the SMS, if so
- 2) generate a smsCommand invoke APDU based on the Command information received from the Sending User, which shall include the following mandatory elements:
  - the Party Number of the Receiving User of the Short Message to which the Command refers in element destinationAddress,
  - a Message Reference in element messageReference which is allocated by the Sending User PINX/Sending User Message Centre for each new Command that is sent (see Annex E for further details),
  - the Message Reference of the Short Message to which the Command refers in element messageNumber,
  - the Protocol Identifier identifying the above layer protocol in element protocolIdentifier (see Annex E for further details)
  - the Command Type in element commandType (see Annex E for further details)
  - optional elements as described in 6.3.1.
- 3) send the smsCommand invoke APDU to the Service Centre,
- 4) start timer T2 and enter state SMS-Send-Command-wait.

On receipt in state SMS-Send-Command-wait of a smsCommand return result APDU the Sending User PINX/ Sending User Message Centre shall

- 1) stop timer T2,
- 2) send an indication to the Sending User that the submission of the Command was successful and
- 3) enter state SMS-Send-Idle.

#### **6.5.1.1.3 Status Report**

On receipt in state SMS-Send-Idle of a smsStatusReport invoke APDU the Sending User PINX/ Sending User Message Centre shall

- 1) send a positive result indication to the Sending User,
- 2) send a smsStatusReport return result to the Service Centre and
- 3) enter state SMS-Send-Idle.

#### **6.5.1.2 Exceptional procedures**

The Sending User PINX/ Sending User Message Centre shall return an error indication to the Sending User if

- 1) the Sending User is not permitted to use the SS-SMS;
- 2) the smsSubmit/smsCommand elements are incorrect or if mandatory elements are missing.

##### **6.5.1.2.1 Short Message**

On receipt in state SMS-Send-Submit-wait of a smsSubmit reject or return error APDU the Sending User PINX/ Sending User Message Centre shall

- 1) stop timer T1,
- 2) send an indication including the error reason to the Sending User and
- 3) enter state SMS-Send-Idle.

On expiry of timer T1 the Sending User PINX/ Sending User Message Centre shall either

- re-send the smsSubmit invoke ADPU, start timer T1 and enter state SMS-Send-Submit-wait or
- send an indication including the error reason to the Sending User and enter state SMS-Send-Idle.

#### *NOTE*

*The number of times the Sending User PINX may repeat the smsSubmit is an implementation matter.*

##### **6.5.1.2.2 Command**

On receipt in state SMS-Send-Command-wait of a smsCommand reject or return error APDU the Sending User PINX/ Sending User Message Centre shall

- 1) stop timer T2;
- 2) send an indication including the error reason to the Sending User and
- 3) enter state SMS-Send-Idle.

On expiry of timer T2 the Sending User PINX/ Sending User Message Centre shall either

- 1) re-send the smsCommand invoke APDU, start timer T2 and enter state SMS-Send-Command-wait or
- 2) send an indication including the error reason to the Sending User and enter state SMS-Send-Idle.

#### *NOTE*

*The number of times the Sending User PINX may repeat the smsCommand is an implementation matter.*

### 6.5.1.2.3 Status Report

On receipt in state SMS-Send-Idle of a smsStatusReport invoke APDU the Sending User PINX/Sending User Message Centre shall in case of an error send a smsStatusReport return error APDU with the mandatory elements

- Failure Cause transported in element failureCause,
- indication if the Service Centre address could be saved transported in scAddressSaved to the Service Centre and enter state SMS-Send-Idle.

### 6.5.2 Actions at a Sending User Message Centre

The procedures for the Sending User Message Centre are as described in 6.5.1 for the Sending User PINX.

### 6.5.3 Actions at a Service Centre

All invoke, return error, return result and reject APDUs shall be transported using the Call Reference of an already existing call independent or call related signalling connection. If no such connection exists, and no call has to be established, the Service Centre shall set up a call independent signalling connection in accordance with the procedures described in 7.3 in ECMA-165. The Service Centre is responsible for the clearing of this call independent signalling connection.

#### NOTE

*The APDU elements are described in annex E.*

### 6.5.3.1 Normal procedures

#### 6.5.3.1.1 Short Message

On receipt in state SMS-SC-Idle of a smsSubmit invoke APDU from the Sending User PINX/Sending User Message Centre the Service Centre shall

- 1) check if the received smsSubmit invoke APDU contains a Short Message with the same messageReference and destinationAddress as a previously received Short Message from the same originatingAddress.  
In case such a Short Message exists and the rejectDuplicates APDU element is set to FALSE or in case that the messageReference is different to the messageReference of the previously received Short Message the Service Centre shall
- 2) check the APDU element Protocol Identifier and
  - if it is set to Replace Short Message check the originating address and replace any existing stored Short Message having the same Protocol Identifier Code and originating address with the new Short Message and other parameter values. If there is no message to be replaced the Service Centre shall store the Short Message in the normal way.
  - if no Replace Short Message code is present the Service Centre shall store the Short Message locally,
- 3) analyse and store the smsSubmit invoke APDU,
- 4) assign and store a serviceCentreTimeStamp for the Short Message,
- 5) send an smsSubmit return result APDU to the Sending User PINX/Sending User Message Centre,
- 6) send an smsDeliver invoke APDU to the Receiving User PINX using the call reference of a call-independent-signalling-connection. The smsDeliver invoke APDU contains the received elements from the smsSubmit invoke ADPU and the following additional elements:
  - optionally, the originatingName of the Sending User if available and not restricted;
  - serviceCentreTimeStamp;
  - priority;
  - moreMessagesToSend: set to TRUE if there are more Short Messages waiting in that Service Centre for that particular destinationAddress;

statusReportIndication: set to TRUE if the statusReportRequest information element is set to TRUE in the original smsSubmit invoke APDU;

replyPath: set to TRUE if the Service Centre supports the Reply Path.

7) start timer T3 and enter state SMS-SC-Deliver-Wait.

On receipt in state SMS-SC-Deliver-Wait of a smsDeliver return result APDU the Service Centre shall

1) stop timer T3 and

2) generate an internal request to send a Status Report with value “smReceivedBySME” if the internal StatusReportRequest bit is set to TRUE and, if included, according to the settings of the element smscControlParameterHeader and

3) enter state SMS-SC-Idle.

Editor’s Note:

Is it necessary to include the mapping between the smsSubmit and smsDeliver APDUs?

Editor’s Note:

Do we have to include: *ME Data download, ME de-personalization Short Message* as Protocol Identifier?

#### 6.5.3.1.2 Command

On receipt in any state except SMS-SC-Idle of an smsCommand invoke APDU the Service Centre shall

- identify a specific locally stored Short Message by the received smsCommand invoke APDU elements
  - originatingAddress
  - messageNumber, including the messageReference of the stored Short Message
- execute the requested commandType on this message, i.e. for commandType
  - “Enquiry” set the internal field StatusReportRequest to TRUE;
  - “CancelSRR” set the internal field StatusReportRequest to FALSE;
  - “EnableSRRrelatingToPreviouslySubmittedSM” set the internal field StatusReportRequest to TRUE;
  - “DeletePreviouslySubmittedSM” delete the identified Short Message;

afterwards send an smsCommand return result APDU to the Sending User PINX/Sending User Message Centre and generate an internal request to send a Status Report if the StatusReportRequest bit is set to TRUE for commandType

- “Enquiry” with a value according to the actual status of the Short Message
- “DeletePreviouslySubmittedSM” with value “smDeletedByOriginatingSME”

and maintain the current state.

#### 6.5.3.1.3 Status-Report

##### NOTE

*The Status Report-related states within the Service Centre are independent of those for the normal Short Message, i.e. parallel states may exist.*

In state SMS-SC-SR-Idle on an internal request to send a Status Report for a specific Short Message if the StatusReportRequest bit is set to TRUE and according to the settings in smscControlParameterHeader, if available, the Service Centre shall send a smsStatusReport invoke APDU to the Sending User PINX/Sending User Message Centre using the following smsStatusReport APDU elements:

- messageReference: value contained in the previously received smsSubmit or smsCommand invoke APDU. If the Status Report is the result of an smsCommand where the Command Type was “Enquiry” the messageReference used in the Status Report shall be the messageNumber contained in the smsCommand invoke APDU (i.e. the messageReference of the previously submitted Short Message to which the “Enquiry” refers).
- dischargeTime: time at which a previously submitted smsSubmit was successfully delivered to or attempted to deliver to the Receiving User or disposed of by the SC,
- serviceCentreTimeStamp: the serviceCentreTimeStamp of the assigned to the original smsSubmit invoke APDU

and other elements as described in 6.3.1, start timer T6 and enter state SMS-SC-SR-Wait.

On receipt in state SMS-SC-SR-Wait of a smsStatusReport return result APDU the Service Centre shall stop timer T6 and enter state SMS-SC-SR-Idle.

### 6.5.3.2 Exceptional procedures

Editors Note:

Is it necessary to describe all relevant error cases here or shall they be included in the annex?

#### 6.5.3.2.1 Short Message

On receipt in state SMS-SC-Idle of a smsSubmit invoke APDU from the Sending User PINX/Sending User Message Centre the Service Centre shall

- 1) check if the received smsSubmit invoke APDU contains a Short Message with the messageReference and destinationAddress of a previously received Short Message from the same originatingAddress.  
In case such a Short Message exists and the rejectDuplicates APDU element is set to TRUE or in case that the messageReference is the same as the messageReference of the previously received Short Message the Service Centre shall send an smsSubmit return error APDU to the Sending User PINX/Sending User Message Centre with failureCause “smRejectedDuplicateSM”, discard the smsSubmit invoke APDU and enter state SMS-SC-Idle.
- 2) check the APDU element Protocol Identifier and  
if it indicates interworking the Service Centre is requested to convert the Short Message into a form suited for that device type. These procedures are outside the scope of this standard. Following these procedures the Service Centre shall enter state SMS-SC-Idle. If interworking is not supported the Service Centre shall return a smsSubmit return error APDU with failureCause “telematicInterworkingNotSupported” and enter state SMS-SC-Idle.

Editors Note:

failureCause “cannotReplaceShortMessage” exists but it is unclear when this can happen.

On receipt in state SMS-SC-Del-Wait of a smsDeliver return error APDU the Service Centre shall check the APDU element failureCause.

If it contains the value “memoryCapacityExceeded” or “simSmsStorageFull” or if an additional Cause Information Element has been received, the Service Centre shall

stop timer T3,

send an internal request with value “errorInSme” for a Status Report if the internal StatusReportRequest bit is set to TRUE and, if available, according to the settings in smscControlParameterHeader,

start timer T4 if the smsDeliver return error APDU element “scAddressSaved” is set to TRUE and

enter state SMS-SC-Await-Alert.

On receipt of a different failureCause the Service Centre shall

- stop timer T3,
- delete the Short Message,
- send an internal request for a Status Report if the internal StatusReportRequest bit is set to TRUE and, if available, according to the setting of smscControlParameterHeader and
- enter state SMS-SC-Idle.

On receipt in state SMS-SC-Del-Wait of a smsDeliver reject APDU the Service Centre shall

- stop timer T3,
- delete the Short Message,
- send an internal request for a Status Report with value “connectionRejectedBySME” if the internal StatusReportRequest bit is set to TRUE and, if available, according to the settings of smscControlParameterHeader, and
- enter state SMS-SC-Idle.

On expiry of timer T3 in state SMS-SC-Deliver-Wait the Service Centre may

- either re-send the smsDeliver invoke APDU to the Receiving User PINX, start timer T3, send an internal request for a Status Report with value “noResponseFromSME” if StatusReportRequest is set to TRUE and, if available, according to the settings of smscControlParameterHeader, enter state SMS-SC-Deliver-Wait
- or delete the Short Message, send an internal request for a Status Report with value “smDeletedByScAdministration” if StatusReportRequest bit is set to TRUE and enter state SMS-SC-Idle.

On receipt in state SMS-SC-Await-Alert of a scAlert invoke APDU from the Receiving User PINX the Service Centre shall

- stop timer T4 if running, check the scAlert invoke APDU and depending on the outcome send an scAlert return result or return error APDU to the Receiving User PINX. If the scAlert is valid the SC shall
  - send the smsDeliver invoke APDU to the Receiving User PINX,
  - start timer T3 and
  - enter state SMS-SC-Deliver-Wait.

On expiry of timer T4 in state SMS-SC-Await-Alert the Service Centre shall

- send the smsDeliver invoke APDU to the Receiving User PINX,
- start timer T3 and
- enter state SMS-SC-Deliver-Wait.

**NOTE**

*The number of times the Service Centre may repeat the delivery attempt for a Short Message is an implementation matter.*

On receipt in state SMS-SC-Await-Alert of an internal indication that the Validity Period for a Short Message expired the Service Centre shall

- stop timer T4 if running and can, as an implementation matter, either
  - start timer T3, send the smsDeliver invoke APDU to the Receiving User PINX with priority set to TRUE and enter state SMS-SC-Deliver-Wait or
  - delete the Short Message, send an internal request for a Status Report with value “smValidityPeriodExpired” if the internal StatusReportRequest bit is set to TRUE and enter state SMS-SC-Idle.

#### 6.5.3.2.2 Command

On receipt in any state of an smsCommand invoke APDU, if no Short Message can be identified, the Service Centre shall

- return an smsCommand return error APDU to the Sending User PINX/Sending User Message Centre with failureCause “commandCannotBeActioned”,
- generate an internal request to send a Status Report with value “smDoesNotExist” if the StatusReportRequest bit is set to TRUE and
- maintain the current state

#### 6.5.3.2.3 Status Report

On receipt in state SMS-SC-SR-Wait of an smsStatusReport return error APDU or an smsStatusReport reject APDU or on expiry of timer T6 the Service Centre shall stop timer T6 (if running) and enter the SMS-SC-SR-Idle and may afterwards

- re-send the smsStatusReport invoke APDU according to the procedures described in section 6.5.3.1.1.

### 6.5.4 Actions at a Receiving User PINX

#### NOTE

*The APDU elements are described in annex E.*

Due to internal administration, the Receiving User PINX shall, upon starting operation for a specific user,

- either enter state SM-Rec-MC-case-Idle if the Short Messages are stored and managed at a Message Centre
- or enter state SM-Rec- -User-case-Idle if the Short Messages are stored and managed locally, either by the Receiving User or by the Receiving User PINX.

#### 6.5.4.1 Normal procedures

In state SM-Rec-MC-case-Idle upon receipt of a smsDeliver return result , reject, return error APDU or a scAlert invoke APDU from the Receiving User Message Centre the Receiving User PINX shall send these APDUs unchanged on the call reference of a call-independent-signalling-connection to the Service Centre.

In state SM-Rec-MC-case Idle upon receipt of a smsDeliver invoke APDU, a scAlert return result, return error or reject APDU the Receiving User PINX shall send these APDUs unchanged on the call reference of a call independent signalling connection to the Receiving User Message Centre.

On receipt in state SMS-Rec-User-case-Idle of a smsDeliver invoke APDU from the Service Centre the Receiving User PINX shall attempt to deliver the SM to the Receiving User. If the SM can successfully be delivered the Receiving User PINX shall send a smsDeliver return result APDU to the Service Centre and enter state SMS-Rec-User-case-Idle.

If element protocolIdentifier is set to “shortMessageType0” the Receiving User PINX must acknowledge the receipt of the Short Message but may discard its contents

#### 6.5.4.2 Exceptional procedures

If the attempt by the Receiving User PINX to deliver a Short Message received from the Service Centre to the Receiving User is not successful, the Receiving User PINX shall

- if SMWD is not implemented,
  - send to the Service Centre a smsDeliver return error APDU with the following elements
    - failureCause set to “errorInMS”,
    - optionally protocolIdentifier as received in the original smsDeliver invoke APDU,
    - optionally userData as received in the original smsDeliver invoke APDU,
    - scAddressSaved set to FALSE and

enter state SMS-Rec-User-case-Idle.

- if SMWD is implemented

save the Service Centre Address as indicated in the Calling Party Number Information Element of the call-independent-signalling-connection on which the smsDeliver invoke APDU was received,

send to the Service Centre a smsDeliver return error APDU with the following elements

- failureCause set to “errorInMS”,
- optionally protocolIdentifier as received in the smsDeliver invoke APDU,
- optionally userData as received in the smsDeliver invoke APDU,
- scAddressSaved set to ONE and

enter state SMS-Rec-User-case-AlertWait.

On receipt in state SMS-Rec-User-case-AlertWait of an internal indication that the user is reachable or that the user has memory available again the Receiving User PINX shall send an scAlert invoke APDU to the Service Centre, start timer T7 and enter state SMS-Rec-User-case-AlertResp.

On receipt in state SMS-Rec-User-case-AlertWait of a smsDeliver invoke APDU with element priority set to TRUE the Receiving User PINX shall attempt to deliver the SM to the Receiving User.

If the SM can be delivered the Receiving User PINX shall

- 1) return an smsDeliver return result APDU to the Service Centre,
- 2) send an scAlert invoke APDU to the Service Centre with element originatingAddress set to the Party Number of the Receiving User,
- 3) start timer T7 and enter state SMS-Rec-User-Alert-Resp.

If the SM can not be delivered then, if the SC Address is not yet stored in SMWD, the SC Address as indicated in the Calling Party Number Information element of the call-independent-signalling-connection shall be saved in SMWD and the Receiving User PINX shall send an smsDeliver return error APDU to the Service Centre with scAddressSaved set to TRUE.

On receipt in state SMS-Rec-User-case-AlertWait of an smsDeliver invoke APDU the Receiving User PINX shall,

- 1) if the SC Address is not saved already in SMWD, save it,
- 2) return a smsDeliver return error APDU to the Service Centre with scAddressSaved set to TRUE and
- 3) enter state SMS-Rec-User-case-AlertWait.

On receipt in state SMS-Rec-User-case-AlertResp of an scAlert return result APDU the Receiving User PINX shall stop timer T7 and enter state SMS-Rec-User-case-Idle.

On receipt in state SMS-Rec-User-case-AlertResp of an scAlert return error or reject APDU the Receiving User PINX shall stop timer T7 and may optionally

- enter state SMS-Rec-User-case-Idle or
- send an scAlert invoke APDU to the Service Centre, start timer T7 and enter state SMS-Rec-User-case-AlertResp.

On receipt in state SMS-Rec-User-case-AlertResp of an expiry of timer T7 the Receiving User PINX may either

- enter state SMS-Rec-User-case-Idle or
- send an scAlert invoke APDU to the Service Centre, start timer T7 and enter state SMS-Rec-User-case-AlertResp.

## 6.5.5 Actions at a Receiving User Message Centre

### NOTE

*The APDU elements are described in annex E.*

### 6.5.5.1 Normal procedures

On receipt in state SMS-Rec-MC-Idle of a smsDeliver invoke APDU from the Receiving User PINX the Receiving User Message Centre shall check the APDU element protocolIdentifier. If it is set to

- shortMessageType0 the Receiving User Message Centre shall send a smsDeliver return result APDU to the Service Centre but may discard the contents of the Short Message.
- replaceShortMessage the Receiving User Message Centre shall check the originatingAddress and replace any existing stored message having the same protocolIdentifier code and originatingAddress with the new Short Message and other APDU elements. If there is no message to be replaced the Receiving User Message Centre shall store the message in the normal way. The Receiving User Message Centre may also check the Service Centre address as indicated in the Calling Party Number Information element of the call-independent-signalling-connection as well as the originatingAddress. However, in a network which has multiple SCs it is possible for a Replace Message type for a SM to be sent via different SCs and so it is recommended that the SC address should not be checked by the Receiving User Message Centre unless the application specifically requires such a check. The Replace Short Message feature is optional but if implemented it shall be performed as described above. If a Replace Short Message Type code is not present then the Receiving User Message Centre will store the message in the normal way.

If the Short Message is saved the Receiving User Message Centre shall send a smsDeliver return result APDU to the Sending User PINX, indicate the reception of a new Short Message to the Receiving User using the SS-MWI and enter state SMS-Rec-MC-Idle.

### 6.5.5.2 Exceptional procedures

On receipt in state SMS-Rec-MC-Idle of a smsDeliver invoke APDU from the Receiving User PINX if it is not possible for the Receiving User Message Centre to save the SM it shall, if SMWD is implemented,

- 1) save the SC address as indicated in the Calling Party Number Information element of the call-independent-signalling-connection,
- 2) return an smsDeliver return error APDU to the Service Centre with failureCause “memoryCapacityExceeded”, in the case of SMWD being implemented the element scAddressStored set to TRUE and
- 3) enter state SMS-Rec-MC-AWait.

If SMWD is not implemented the Receiving User Message Centre shall return an smsDeliver return error APDU to the Receiving User PINX with failureCause “memoryCapacityExceeded” and enter state SMS-Rec-MC-Idle.

On receipt in state SMS-Rec-MC-AWait of an internal indication that the user is reachable or that the user has memory available again the Receiving User Message Centre shall

- 1) send an scAlert invoke APDU to the Service Centre,
- 2) start timer T8 and enter state SMS-Rec-MC-AResp.

On receipt in state SMS-Rec-MC-AWait of a priority smsDeliver invoke APDU the Receiving User Message Centre shall attempt to save the SM. If the SM can be saved the Receiving User Message Centre shall

- 1) return an smsDeliver return result APDU to the Receiving User PINX,
- 2) send an scAlert invoke APDU to the Receiving User PINX,
- 3) start timer T8 and enter state SMS-Rec-MC-AResp.

If the SM can not be saved then, if the SC Address is not yet stored in SMWD, the SC address as indicated in the Calling Party Number Information element of the call-independent-signalling-connection shall be saved in SMWD, the Receiving User Message Centre shall

- 1) send an smsDeliver return error APDU to the Receiving User with scAddressStored set to TRUE and
- 2) enter state SMS-Rec-MC-AWait.

On receipt in state SMS-Rec-MC-AWait of an smsDeliver invoke APDU the Receiving User Message Centre shall

- 1) if the SC Address is not saved already in SMWD save it as indicated in the Calling Party Number Information element of the call-independent-signalling-connection,
- 2) return an smsDeliver return error APDU with scAddressStored set to TRUE to the Receiving User PINX and
- 3) enter state SMS-Rec-MC-AWait.

On receipt in state SMS-Rec-MC-AResp of an scAlert return result APDU the Receiving User Message Centre shall

- 1) stop timer T8
- 2) delete the SC Address from the SMWD field and
- 3) enter state SMS-Rec-MC-Idle.

On receipt in state SMS-Rec-MC-AResp of an scAlert return error or reject APDU the Receiving User Message Centre shall stop timer T8 and may optionally

- enter state SMS-Rec-MC-Idle or
- send an scAlert invoke APDU to the Service Centre, start timer T8 and enter state SMS-Rec-MC-AResp.

On expiry of timer T8 in state SMS-Rec-MC-AResp the Receiving User Message Centre may either

- enter state SMS-Rec-MC-Idle or
- send an scAlert invoke APDU to the Service Centre, start timer T8 and enter state SMS-Rec-MC-AResp.

## **6.6 SS-SMS impact on interworking with public ISDNs**

## **6.7 SS-SMS impact on interworking with non-ISDNs**

## **6.8 Protocol Interactions between SS-SMS and other supplementary services and ANFs**

This clause specifies protocol interactions with other supplementary services and ANFs for which stage 3 standards had been published at the time of publication of this Standard. For interactions with supplementary services and ANFs for which stage 3 standards are published subsequent to the publication of this Standard, see those other stage 3 standards.

### *NOTE 1*

*Simultaneous conveyance of APDUs for SS-SMS and another supplementary service or ANF in the same message, each in accordance with the requirements of its respective stage 3 standard, does not, on its own, constitute a protocol interaction.*

### *NOTE 2*

*Additional interactions that have no impact on the signalling protocol at the Q reference point can be found in the relevant stage 1 specification.*

- 6.8.1 Calling Line Identification Presentation (SS-CLIP)**  
No protocol interaction.
- 6.8.2 Connected Line Identification Presentation (SS-COLP)**  
No protocol interaction.
- 6.8.3 Calling/ Connected Line Identification Restriction (SS-CLIR)**  
No protocol interaction.
- 6.8.4 Calling Name Identification Presentation (SS-CNIP)**  
No protocol interaction.
- 6.8.5 Calling/ Connected Name Identification Restriction (SS-CNIR)**  
No protocol interaction.
- 6.8.6 Connected Name Identification Presentation (SS-CONP)**  
No protocol interaction.
- 6.8.7 Completion of Calls to Busy Subscriber (SS-CCBS)**  
No protocol interaction.
- 6.8.8 Completion of Calls on No Reply (SS-CCNR)**  
No protocol interaction.
- 6.8.9 Call Transfer (CT)**  
No protocol interaction.
- 6.8.10 Call Forwarding Unconditional (SS-CFU)**  
No protocol interaction.
- 6.8.11 Call Forwarding Busy (SS-CFB)**  
No protocol interaction.
- 6.8.12 Call Forwarding No Reply (SS-CFNR)**  
No protocol interaction.
- 6.8.13 Call Deflection (SS-CD)**  
No protocol interaction.
- 6.8.14 Path Replacement (ANF-PR)**  
No protocol interaction.
- 6.8.15 Call Offer (SS-CO)**  
No protocol interaction.
- 6.8.16 Call Intrusion (SS-CI)**  
No protocol interaction.
- 6.8.17 Do Not Disturb (SS-DND)**  
No protocol interaction.
- 6.8.18 Do Not Disturb Override (SS-DNDO)**  
No protocol interaction.
- 6.8.19 Advice of charge (SS-AOC)**  
No protocol interaction.
- 6.8.20 Recall (SS-RE)**  
No protocol interaction.
- 6.8.21 Call Interception (ANF-CINT)**  
No protocol interaction.

**6.8.22 Transit Counter (ANF-TC)**

No protocol interaction.

**6.8.23 Route Restriction Class (ANF-RRC)**

No protocol interaction.

**6.8.24 Message Waiting Indication (SS-MWI)**

The Receiving User Message Centre shall, upon receipt and storage of a smsDeliver invoke APDU, send a mwiActivate invoke APDU with element basicService set to “shortMessageService” to the Receiving User PINX.

**6.8.25 Cordless Terminal Location Registration (SS-CTLR)**

No protocol interaction.

**6.8.26 Cordless Terminal Mobility Incoming Call (SS-CTMI)**

No protocol interaction

**6.8.27 Cordless Terminal Mobility Outgoing Call (SS-CTMO)**

No protocol interaction.

**6.8.28 Authentication of a CTM user (SS-CTAT)**

No protocol interaction.

**6.8.29 Authentication of the PISN (SS-CTAN)**

No protocol interaction.

**6.8.30 Private User Mobility Incoming Call (ANF-PUMI)**

No protocol interaction.

**6.8.31 Private User Mobility Outgoing Call (ANF-PUMO)**

No protocol interaction.

**6.8.32 Private User Mobility Registration (SS-PUMR)**

No protocol interaction.

**6.8.33 Common Information (ANF-CMN)**

No protocol interaction.

**6.8.34 Call Priority Interruption (Protection) (SS-CPI(P))**

No protocol interaction.

**6.8.35 Single Step Call Transfer (SS-SSCT)**

No protocol interaction.

**6.8.36 Simple Dialog (SS-SD)**

No protocol interaction.

**6.8.37 Call Identification and Call Linkage (ANF-CIDL)**

No protocol interaction.

**6.9 SS-SMS Parameter values (Timers)**

Editors Note:

Values for timers still to be defined.

**Timer T1**

Timer T1 shall operate at the Sending User PINX during state SMS-Send-Submit-wait. Its purpose is to protect against an absence of response to smsSubmit invoke APDU sent during the call.

Timer T1 shall have a value in the range XX.

### **Timer T2**

Timer T2 shall operate at the Sending User PINX during state SMS-Send-Command-wait. Its purpose is to protect against an absence of response to smsCommand invoke APDU sent during the call.

Timer T2 shall have a value in the range XX.

### **Timer T3**

Timer T3 shall operate at the Service Centre during state SMS-SC-Deliver-Wait. Its purpose is to protect against absence of response to smsDeliver invoke APDU sent during the call.

Timer T3 shall have a value in the range XX.

### **Timer T4**

Timer T4 may operate optionally at the Service Centre, if Short Message Waiting Data is not implemented, during state SMS-SC-Await-Alert. Its purpose is to ensure the automatic repetition of the delivery attempt of a Short Message .

Timer T4 shall have a value in the range XX.

### **Timer T5**

Timer T5 shall operate at the Service Centre during state SMS-SC-Prio-Wait. Its purpose is to protect against absence of response to smsDeliver APDU sent during the call.

Timer T5 shall have a value in the range XX.

### **Timer T6**

Timer T6 shall operate at the Service Centre during state SMS-SC-SR-Wait. Its purpose is to protect against absence of response to smsStatusReport invoke APDU sent during the call.

Timer T6 shall have a value in the range XX.

### **Timer T7**

Timer T7 shall operate at the Receiving PINX in the case where the Short Messages are sent to a Terminal. It shall operate during state SMS-Rec-User-case-AlertResp. Its purpose is to protect against an absence of response to scAlert invoke APDU sent during the call.

Timer T7 shall have a value in the range XX.

### **Timer T8**

Timer T8 shall operate at the Receiving User Message Centre during state SMS-Rec-MC-AResp. Its purpose is to protect against absence of response to scAlert invoke APDU sent during the call.

Timer T8 shall have a value in the range XX.

## ANNEX E

### Description of APDU elements

#### 1. Elements in smsSubmit invoke APDU

##### 1.1 messageReference

The messageReference is a value between 0 and 255 which is incremented by one and allocated to each Short Message that is sent by the Sending User. If the value is 255 and the messageReference is incremented, it starts again with value 0.

If the Sending User wants to overwrite an already sent Short Message, the new Short Message shall be assigned the same messageReference as the already sent one. Additionally element rejectDuplicates within smsSubmitParameter of the smsSubmit invoke APDU shall be set to TRUE.

##### 1.2 smsSubmitParameter

###### 1.2.1 protocolIdentifier

The protocolIdentifier indicates either a higher layer protocol being used or interworking with a certain type of telematic device.

For the straightforward case of simple Short Message transfer the element protocolIdentifier is set to "noIw" (0).

For protocolIdentifier values between (1) and (31) it indicates the protocol used between the Sending and Receiving Short Message entity.

If the protocolIdentifier contains values between (32) and (63) it indicates interworking with a certain type of telematic device and requests the Service Centre to convert the SM into a form suited for that device type. If the destination network is ISDN the SC must also select the correct service indicators for connecting to a device of that type.

If the protocolIdentifier contains a "replaceShortMessageTypeX" then the Service Centre on receipt of such a protocolIdentifier will check the originatingAddress and replace any existing Short Message having the same protocolIdentifier value and originatingAddress with the new Short Message and other elements. If there is no message to be replaced the Service Centre shall store the new Short Message in the normal way.

If a "replaceShortMessageTypeX" type code is not present the Service Centre shall store the message in the normal way.

###### 1.2.2 validityPeriod

The Validity Period enables the Sending User to indicate the time period during which the Sending User considers the Short Message to be valid, i.e. for how long the SC shall guarantee its existence in the SC memory before delivery to the Receiving User has been carried out.

###### 1.2.3 statusReportRequest

If statusReportRequest is set to TRUE a Status Report is requested from the Sending User for this particular Short Message.

###### 1.2.4 replyPath

If replyPath is set to TRUE in a smsSubmit invoke APDU the Sending User requests the Service Centre to handle a Short Message sent in reply to this smsSubmit via the same Service Centre.

Editors Note:

The description of the replyPath procedures still have to be included!

###### 1.2.5 rejectDuplicates

Reject Duplicates indicates to the SC whether it shall accept or reject a Short Message with the same messageReference and destinationAddress as a previously submitted Short Message from the same

originatingAddress. If the Short Message is rejected an appropriate failureCause will be returned to the Sending User in the smsDeliver return error APDU.

### **1.3 userData**

#### **1.3.1 userDataHeader**

If a userDataHeader is present it indicates a special Short Message. The userDataHeader may contain one or more of the following headers.

##### **1.3.1.1 smscControlParameterHeader**

This header makes it possible to expand the userDataHeader in a flexible way and controls the Service Centre with regard to the request for Status Reports. The Sending User may request a Status Report for

- Transaction completed (bit 0),
- permanent Error (bit 1)
- temporary Error, SC not trying anymore (bit 2),
- temporary Error, SC still trying (bit 3),
- cancel a Status Report Request for concatenated Short Messages (bit 6) and
- request the original userDataHeader to be included into the Status Report (bit 7).

If bit 6 is set to TRUE then a Status Report generated by this Short Message due to a permanent error or last temporary error cancels the statusReportRequest of the rest of the Short Messages in a concatenated Short Message. This feature can only be used where the Service Centre is aware of the segmentation and is therefore an implementation matter.

The smscControlParameterHeader element is only enabled if the internal statusReportRequest bit of the Short Message is set to TRUE.

In the case of concatenated Short Messages smscControlParameterHeaders must be present in every Short Message making up that concatenated Short Message.

##### **1.3.1.2 concatenated8BitSMHeader**

Longer messages (which exceed the length of one Short Message) can be formed by concatenating several Short Messages.

The concatenated8BitSMHeader contains a concatenated8BitSMReferenceNumber, a maximumNumberOf8BitSMInConcatenatedSM and a sequenceNumberOf8BitSM element.

The concatenated8BitSMReferenceNumber element contains a modulo 255 counter and indicates the Reference Number of a particular concatenated Short Message. This Reference Number shall remain constant for every Short Message which makes up a particular concatenated Short Message.

The maximumNumberOf8BitSMInConcatenatedSM element is a value in the range 0 to 255 and indicates the total number of Short Messages within the concatenated Short Message. The value shall start at 1 and remain constant for every Short Message which makes up the concatenated Short Message.

The sequenceNumberOf8BitSM element is a value in the range 0 to 255 and indicates the sequence number of a particular Short Message within the concatenated Short Message. The value shall start at 1 and increment by one for every Short Message sent within the concatenated Short Message.

##### **1.3.1.3 concatenated16BitSMHeader**

The concatenated16BitSMHeader is an enhanced variant of the concatenated8BitSMHeader. The enhancement is a 16-bit Reference Number instead of the short 8-bit Reference Number. The elements are as described in 1.3.1.2 apart from the concatenated8BitSMReferenceNumber which is replaced by a concatenated16BitSMReferenceNumber.

The concatenated16BitSMReferenceNumber element contains a modulo 65536 counter and indicates the Reference Number of a particular concatenated Short Message. This Reference Number shall remain constant for every Short Message which makes up a particular concatenated Short Message.

#### **1.3.1.4 applicationPort8BitHeader**

The applicationPort8BitHeader element allows Short Messages to be routed directly to one of multiple applications in the Terminal Equipment. The applicationPort8BitHeader contains a destination8BitPort and an originator8BitPort element.

The destination8BitPort element is a value in the range 0 to 255 and indicates the receiving port, i.e. application, in the receiving device.

The originating8BitPort is a value in the range 0 to 255 and indicates the sending port, i.e. application, in the sending device.

Port numbers in the range 0 to 239 are reserved and port numbers in the range 240 to 255 are available for allocation by applications.

In the case of concatenated Short Messages the applicationPort8BitHeader shall be included in every Short Message making up the concatenated Short Message.

#### **1.3.1.5 applicationPort16BitHeader**

The applicationPort16BitHeader element allows Short Messages to be routed directly to one of multiple applications in the Terminal Equipment. The applicationPort16BitHeader contains a destination16BitPort and an originator16BitPort element.

The destination16BitPort element is a value in the range 0 to 65535 and indicates the receiving port, i.e. application, in the receiving device.

The originating16BitPort is a value in the range 0 to 65535 and indicates the sending port, i.e. application, in the sending device.

Port numbers in the range 0 to 15999 are as allocated by IANA (<http://www.IANA.com>), port numbers in the range 16000 to 16999 are available for allocation by applications and all other values are reserved.

In the case of concatenated Short Messages the applicationPort16BitHeader shall be included in every Short Message making up the concatenated Short Message.

#### **1.3.1.6 dataHeaderSourceIndicator**

The dataHeaderSourceIndicator element is used to separate the userDataHeader of the original message, a userDataHeader created by the Service Centre and a userDataHeader provided by the Receiving User. The dataHeaderSourceIndicator is placed in front of the content inserted by the source.

The dataHeaderSourceIndicator may contain the values originalSender (1), originalReceiver (2) or SMSC (3).

#### **1.3.1.7 wirelessControlHeader**

The wirelessControlHeader element is used to transport Wireless-Control-Message-Protocol (WCMP) messages. The OCTET STRING associated with the wirelessControlHeader shall contain a WCMP protocol data unit.

In the case of concatenated Short Messages the wirelessControlHeader shall be included in every Short Message making up the concatenated Short Message.

#### **1.3.1.8 genericUserValue**

The genericUserValue element is reserved for further extensions introduced in GSM 03.40.

### **1.3.2 class**

The class element indicates how to handle a received Short Message with respect to displaying, storing and acknowledging.

If element class is set to 0 the Receiving User PINX/ Receiving User Message Centre shall display the Short Message immediately and send an smsDeliver return result APDU to the Service Centre when the Short Message has successfully reached the Receiving User PINX/Receiving User Message Centre, irrespective of whether there is memory available or not. The Short Message shall not automatically be stored.

If element class is set to 1 the Receiving User PINX/Receiving User Message Centre shall send a smsDeliver return result APDU to the Service Centre when the Short Message has reached the Receiving User PINX/ Receiving User Message Centre and can be stored.

Editors Note:

Is it necessary to include the description of class 2?

### 1.3.3 **compressed**

Indicates that the text contained in shortMessageText is compressed using the standard compression algorithm. The compression algorithm is described in ETSI TS 101 032 (GSM 03.42).

### 1.3.4 **shortMessageText**

The shortMessageText element contains one of the following elements

#### 1.3.4.1 **iA5Coded**

The iA5Coded element contains between 0 and 160 byte Short Message Text.

#### 1.3.4.2 **octetCoded**

The octetCoded element contains between 0 and 140 byte Short Message Text.

#### 1.3.4.3 **uniCoded**

The uniCoded element contains between 0 and 140 byte Short Message Text coded according to ISO/IEC 10646-1.

#### 1.3.4.4 **compressedCoded**

The compressedCoded element contains between 0 and 140 byte compressed Short Message Text including the Compressed Data Header and Footer described in GSM 03.42.

## 2. **Elements in smsSubmit return result APDU**

### 2.1 **serviceCentreTimeStamp**

The serviceCentreTimeStamp indicates the time of arrival of a Short Message at the Service Centre. The Service-Centre-Time-Stamp, and any other times that are defined in this specification, represent the time local to the sending entity.

The date and time representation follows ISO 8601.

The serviceCentreTimeStamp contains the time of arrival with an accuracy of a second. If two or more messages arrive at the Service Centre at the same time the SC shall modify the serviceCentreTimeStamp in such a way that:

- all messages to one Receiving User contain different time stamps,
- the modification of the time stamps is kept to a minimum.

### 2.2 **protocolIdentifier**

As described in 1.2.1 in Annex E.

### 2.3 **userData**

The userData element in a smsSubmit return result APDU is only available for use by the Service Centre. The elements shall be used as described in 1.3 in Annex E.

It is possible for the Service Centre to include further userDataHeader in the smsSubmit return result APDU by adding the dataHeaderSourceIndicator and other userDataHeader elements.

## 3. **Elements in smsSubmit return error APDU**

### 3.1 **failureCause**

The failureCause element contains the reason for the smsSubmit failure. All possible failureCauses are contained in 6.3.1.

Editors Note:

The occurrence of the different error causes and their assignment to possible error scenarios still have to be defined

### 3.2 **serviceCentreTimeStamp**

As described in 2.1 of Annex E.

### 3.3 **protocolIdentifier**

As described in 1.2.1 in Annex E.

### 3.4 **userData**

The userData element in a smsSubmit return error APDU is only available for use by the Service Centre. The elements shall be used as described in 1.3 in Annex E.

It is possible for the Service Centre to include further userDataHeader in the smsSubmit return result APDU by adding the dataHeaderSourceIndicator and other userDataHeader elements.

## 4. **Elements in smsDeliver invoke APDU**

### 4.1 **originatingName**

Name of the Sending User if available to the Service Centre and not restricted.

### 4.2 **smsDeliverParameter**

#### 4.2.1 **protocolIdentifier**

Editors Note:

How is it possible to describe the following text for QSIG?

If the protocolIdentifier is set to values between (32) and (63) it indicates that the Short Message Entity is a telematic device of the indicated type.

If the protocolIdentifier is set to values between (1) and (31) it identifies the application layer protocol used between the Short Message entity and the Mobile Station.

If the protocolIdentifier contains a “replaceShortMessageTypeX” then the Receiving User PINX/ Receiving User Message Centre on receipt of such a protocolIdentifier will check the originatingAddress and replace any existing Short Message having the same protocolIdentifier value and originatingAddress with the new Short Message and other elements. If there is no message to be replaced the Receiving User PINX/ Receiving User Message Centre shall store the new Short Message in the normal way.

If a “replaceShortMessageTypeX” type code is not present the Receiving User PINX/ Receiving User Message Centre shall store the message in the normal way.

#### 4.2.2 **serviceCentreTimeStamp**

As described in 2.1 of Annex E.

#### 4.2.3 **priority**

If the priority element is set to FALSE the message transfer should be stopped if the SC address is already contained in the SMWD field.

#### 4.2.4 **moreMessagesToSend**

If the moreMessagesToSend element is set to TRUE it indicates that there are more messages waiting for this Receiving User in that particular Service Centre.

#### 4.2.5 **statusReportIndication**

If the statusReportIndication element is set to TRUE it indicates that a Status Report will be returned to the Sending User of the Short Message.

#### 4.2.6 **replyPath**

If the replyPath is set to TRUE it indicates that the reply path for the Short Message exists, e.g. that the Receiving User may reply via this Service Centre although it may not be the default Service Centre.

Editors Note:

The reply path procedures still have to be included in the standard draft.

#### 4.3 **userData**

The userData element contains the userData element received from the Sending User PINX/Sending User Message Centre as included in the smsSubmit invoke APDU as described in 1.3 in Annex E.

It is possible for the Service Centre to include further userDataHeader in the smsDeliver invoke APDU by adding the dataHeaderSourceIndicator and other userDataHeader elements.

### 5. **Elements in smsDeliver return result APDU**

The smsDeliver return result APDU shall either contain one or both of the following elements.

#### 5.1 **protocolIdentifier**

As described in 4.2.1 in Annex E.

#### 5.2 **userData**

The userData element in a smsDeliver return result APDU is only available for use by the Receiving User PINX/Receiving User Message Centre.

It is possible for the Receiving User PINX/ Receiving User Message Centre to include further userDataHeader in the smsDeliver return result APDU by adding the dataHeaderSourceIndicator and other userDataHeader elements.

### 6. **Elements in smsDeliver return error APDU**

#### 6.1 **failureCause**

The failureCause element contains the reason for the smsDeliver failure. All possible failureCauses are contained in 6.3.1.

Editors Note:

The occurrence of the different error causes and their assignment to possible error scenarios still have to be defined

#### 6.2 **protocolIdentifier**

As described in 1.2.1 in Annex E.

#### 6.3 **userData**

The userData element in a smsDeliver return error APDU is only available for use by the Receiving User PINX/Receiving User Message Centre. The userData element shall be used according to 1.3 in Annex E.

It is possible for the Receiving User PINX/ Receiving User Message Centre to include further userDataHeader in the smsDeliver return error APDU by adding the dataHeaderSourceIndicator and other userDataHeader elements.

#### 6.4 **scAddressSaved**

If the scAddressSaved element is set to TRUE it indicates that the Receiving User PINX/ Receiving User Message Centre stored the Service Centre address, hence the Service Centre does not have to repeat the delivery procedure periodically.

### 7. **Elements in smsStatusReport invoke APDU**

#### 7.1 **messageReference**

The messageReference in a smsStatusReport invoke APDU shall contain the messageReference of the smsSubmit or smsCommand invoke APDU to which the Status Report refers. If the Status Report refers to a smsCommand invoke APDU an the commandTyp was an "Enquiry" the messageReference element shall contain the messageNumber contained in that smsCommand invoke APDU (e.g. the messageReference of the previously received Short Message to which the Enquiry refers).

## **7.2 serviceCentreTimeStamp**

The `serviceCentreTimeStamp` element shall contain the `serviceCentreTimeStamp` of the Short Message to which the Status Report refers. This will allow the Sending User to associate a Short Message with a Status Report by correlating the `serviceCentreTimeStamps`.

## **7.3 dischargeTime**

The `dischargeTime` element indicates the time at which a previously submitted Short Message was successfully delivered to the Receiving User or attempted to deliver to the Receiving User or disposed of by the SC. The representation of the `dischargeTime` follows ISO 8601.

## **7.4 recipientAddress**

The `recipientAddress` element contains the address of the Receiving User of the previously submitted Short Message.

## **7.5 recipientName**

The `recipientName` element contains the name of the Receiving User of the previously submitted Short Message if it is available to the Service Centre and not restricted.

## **7.6 status**

The `status` element indicates the status of a previously submitted Short Message and certain Commands for which a StatusReport has been requested. All possible values are described in 6.3.1 and the concerned procedures.

## **7.7 priority**

As described in 4.2.3 in Annex E.

## **7.8 moreMessagesToSend**

As described in 4.2.4 in Annex E.

## **7.9 statusReportQualifier**

If the `statusReportQualifier` is set to FALSE the Status Report is the result of a Short Message, if it is set to TRUE the Status Report is the result of a Command, e.g. a Enquiry.

## **7.10 protocolIdentifier**

The `protocolIdentifier` element in a `smsStatusReport` invoke APDU shall contain the same settings as received in the related `smsSubmit` invoke APDU, described in 1.2.1 in Annex E.

## **7.11 userData**

The `userData` element in a `smsStatusReport` invoke APDU contains information related to a `smsDeliver` invoke APDU, can contain information transported in the `userData` of a `smsDeliver` return result APDU and information inserted by the Service Centre.

The Service Centre can include further `userDataHeader` in the `smsStatusReport` invoke APDU by adding the `dataHeaderSourceIndicator` and other `userDataHeader` elements.

The default content of the `userDataHeader` in a `smsStatusReport` invoke APDU is the `userDataHeader` copied from the `smsDeliver` return result APDU.

## **8. Elements in smsStatusReport return result APDU**

The `smsStatusReport` return result APDU shall either contain one or both of the following elements.

### **8.1 protocolIdentifier**

As described in 5.1 in Annex E.

### **8.2 userData**

As described in 5.2 in Annex E.

## 9. Elements in smsStatusReport return error APDU

### 9.1 failureCause

As described in 6.1 of Annex E.

### 9.2 protocolIdentifier

As described in 6.2 of Annex E.

### 9.3 userData

As described in 6.3 of Annex E.

### 9.4 scAddressSaved

As described in 6.4 of Annex E.

## 10. Elements in smsCommand invoke APDU

### 10.1 messageReference

The messageReference is a value between 0 and 255 which is incremented by one and allocated to each Command that is sent by the Sending User. If the value is 255 and the messageReference is incremented, it starts again with value 0.

### 10.2 messageNumber

The messageNumber element contains the messageReference of the Short Message to which the Command refers, e.g. the messageReference of a previously submitted Short Message. The messageNumber is only necessary for Commands that operate on a specific Short Message, else it shall be ignored.

### 10.2 protocolIdentifier

As described in 1.2.1 of Annex E.

### 10.3 commandType

The commandType element specifies which operation is to be performed on a Short Message. The commandType can be set to “Enquiry” (0), “CancelSRR” (1), “DeletePreviouslySubmittedSM” (2) and “EnableSRRrelatingToPreviouslySubmittedSM” (3).

If the commandType is set to “Enquiry” and the operation was successful the Service Centre will send a Status Report for the Short Message to which the “Enquiry” refers. In the case where the SC has a number of Short Messages which have the same messageReference, the same destinationAddress and have come from the same originatingAddress the Service Centre will send a Status Report for each Short Message.

In the case where the commandType is set to “DeletePreviouslySubmittedSM” (2) the Service Centre will send a Status Report indicating that the Short Message has been deleted if the statusReportRequest in the original smsSubmit invoke APDU was set to TRUE.

### 10.4 commandData

The commandData element contains data related to the operation requested by the Sending User.

Editors Note:

There are no further details given in GSM 03.40 for the usage and content of commandData

### 10.5 statusReportRequest

The statusReportRequest element indicates whether or not the Command requests a Status Report. It shall be set TRUE for commandType “Enquiry”, otherwise it shall be set FALSE.

## 11. Elements in smsCommand return result APDU

### 11.1 serviceCentreTimeStamp

As described in 2.1 of Annex E.

### 11.2 protocolIdentifier

As described in 2.2 of Annex E.

**11.3 userData**

As described in 2.3 of Annex E.

**12. Elements in smsCommand return error APDU**

**12.1 failureCause**

As described in 3.1 of Annex E.

**12.2. serviceCentreTimeStamp**

As described in 3.2 of Annex E.

**12.3 protocolIdentifier**

As described in 3.3 of Annex E.

**12.4 userData**

As described in 3.4 of Annex E.

**1.8 Originating-Address**

Address (e.g. E.164, PNP) of the Sending User.

**1.11 Receiving User**

The terminating user, i.e. the user receiving a Short Message.

**1.14 Sending User**

The originating user, i.e. the user sending a SM.

## ANNEX F

### 1. User Data Headers

#### 1.1 Concatenated Short Message

It shall be possible for the user to send several Short Messages which together form a longer message. To achieve that, the Concatenated Short Message information must be present in all Short Messages that together form the Concatenated Short Message. The Concatenated Short Message information shall contain:

- the Concatenated Short Message Reference,
- the Maximum-Number-of-concatenated-SMs and
- the Sequence-Number of the actual Short Message.

The elements in the Short Messages forming a Concatenated Short Message should remain unchanged apart from the Short-Message-Text and the Sequence-Number. The Sequence-Number and the Maximum-Number-of-concatenated-SMs are used to assemble the Short-Message-Text again at the Receiving User side. The Concatenated Short Message Reference is used to differentiate between different Concatenated Short Messages.

#### 1.2 SMSC Control Parameters

This information element enables the Sending User to determine in more detail in which cases a Status Report shall be returned from the SC. The following settings are possible:

- Status Report for Short Message transaction completed;
- Status Report for permanent error when SC is not making any more transfer attempts;
- Status Report for temporary error when SC is not making any more transfer attempts;
- Status Report for temporary error when SC is still trying to transfer Short Message;
- cancel all Status Report requests generated by the Short Messages in a Concatenated Short Message.

All above mentioned settings can be used independently of each other, hence several Status Reports can be sent for the same Short Message. The Status Report Request bit must be set in order for the SMSC Control Parameters to be enabled.

#### 1.3 Application port addressing

This information element allows Short Messages to be routed to one of multiple applications in the terminal equipment, using a method similar to TCP/UDP ports in a TCP/IP network. An application is uniquely identified by the pair of Destination/ Originating Address and the port address. The port addressing is transparent to the transport and also useful in Status Reports.

#### 1.4 User Data Header Source Indicator

This information element is used to separate the User Data Header of the original message, a User Data Header created by the SMSC and a UDH provided by the original receiving entity. The source indicator is placed in front of the content inserted by the source. The indicated content (one or more Information Elements) ends at the next UDH-Source Indicator or at the end of the UDH. The Separator is intended to be used especially in Status Report, but can also be used by the SMSC to add information into a Short Message. The default content for a UDH in a smsDeliver is the headers inserted by the sending device and the default content for a UDH in a smsStatusReport is the headers copied from the smsDeliver-Report.

#### 1.5 Wireless Control Message Protocol

The Wireless Control Message Protocol (WCMP) is part of the WAP suit of protocols; an open standard specified by the WAP-Forum Ltd.

The protocol specifies a set of messages that can be user by the receiver to notify the sender if an error occurs. This can be due to routing problems, no application listening at the destination port number, or due to insufficient buffer capacity. The error messages can be used by the sender to avoid retransmitting packets, that can not be properly handled at the receiver. WCMP can also be used for diagnostics and informational purposes. WCMP messages are usually generated by a datagram transport layer or a management entity.

## **2. Special Short Messages**

### **2.1 SMS and Internet Electronic Mail interworking**

The interworking between Internet electronic mail and SMS is offered in both directions. The formatting rules for transferring email in either direction are described in 3.8 in ETSI TS 100 901 (GSM 03.40) and shall be followed by a user of a private network wishing to send Internet electronic mail using SMS.

### **2.2 SMS Compression**

Short Messages may be compressed in accordance with the compression algorithm described in ETSI TS 101 032 (GSM 03.42) and according to the rules described in 3.9 in ETSI TS 100 901 (GSM 03.40).

REFMERGEFORMATLINKMERGEFORMATSEQARABICLINKMERGEFORMATREFLINKMERGEFO  
RMATREFREFLINKMERGEFORMATSEQARABICLINKMERGEFORMATREFLINKMERGEFORMATREFLINK  
MERGEFORMATREFLINKMERGEFORMATREFREFLINKMERGEFORMATSEQARABICLINKMERG  
EFORMATREFREFMERGEFORMATLINKMERGEFORMATSEQARABICLINKMERGEFORMATREFM  
ERGEFORMATLINKMERGEFORMATREFREFMERGEFORMATLINKMERGEFORMATSEQARABICLINKMER  
GEFORMATREF

Free printed copies can be ordered from:

**ECMA**

114 Rue du Rhône

CH-1204 Geneva

Switzerland

Fax: +41 22 849.60.01

Email: [documents@ecma.ch](mailto:documents@ecma.ch)

Files of this Standard can be freely downloaded from the ECMA web site ([www.ecma.ch](http://www.ecma.ch)). This site gives full information on ECMA, ECMA activities, ECMA Standards and Technical Reports.

**ECMA**  
114 Rue du Rhône  
CH-1204 Geneva  
Switzerland

See inside cover page for obtaining further soft or hard copies.