3GPP TSG CN Plenary Meeting #20　　　　　　　　　　　　　　　**NP-030187**
4th - 6th June 2003. HÄMEENLINNA, Finland.

Source:	**CN2**
Title:	**CR on Rel-5 Work Item CAMEL4**
Agenda item:	**8.3**
Document for:	**APPROVAL**

<u>Introduction:</u>

This document contains 1 CR on Rel-5 WI CAMEL4. This CRs has been agreed by TSG CN WG2 as alternative proposal to document NP-030186 ("Health warning for Calling Party Number length in IDP SMS") and is forwarded to TSG CN Plenary meeting #20 for decision and approval.

Spec	CR	Rev	Doc-2nd-Level	Phase	Subject	Cat	Ver_C
29.078	324	2	N2-030314	Rel-5	ASN.1 change for Calling Party Number length in IDP SMS	F	5.3.0

CR-Form-v7

CHANGE REQUEST

⌘ **29.078** CR **324** ⌘rev **2** ⌘ Current version: **5.3.0** ⌘

For **HELP** *on using this form, see bottom of this page or look at the pop-up text over the ⌘ symbols.*

Proposed change affects: UICC apps⌘ ☐   ME ☐ Radio Access Network ☐ Core Network **X**

Title: ⌘	ASN.1 change for Calling Party Number length in InitialDPSMS and ConnectSMS	
Source: ⌘	Siemens AG	
Work item code: ⌘	CAMEL4	*Date:* ⌘ 23/05/2003
Category: ⌘ **F**		*Release:* ⌘ Rel-5

Use one of the following categories:
F (correction)
A (corresponds to a correction in an earlier release)
B (addition of feature),
C (functional modification of feature)
D (editorial modification)
Detailed explanations of the above categories can
be found in 3GPP TR 21.900.

Use one of the following releases:
2 (GSM Phase 2)
R96 (Release 1996)
R97 (Release 1997)
R98 (Release 1998)
R99 (Release 1999)
Rel-4 (Release 4)
Rel-5 (Release 5)
Rel-6 (Release 6)

Reason for change: ⌘	When the MSC or SGSN initiates a CAMEL dialogue with the SCP for a Mobile Terminated SMS, it includes the Calling Party Number in the InitialDPSMS Operation. The data type used to carry the Calling Party Number is ISDN-AddressString. ISDN-AddressString may have a legth of nine OCTETs, of which one OCTET is used for header information, which leaves eight OCTETs for address digits, which corresponds with a maximum of 16 address digits. The Calling Party Number for a Mobile Terminated SMS is carried in the TP-Originating-Address from SMS-Deliver-TPDU. SMS-Deliver-TPDU is contained in sm-RP-UI. Refer to 3GPP TS 23.040. The TP-Originating-Address from SMS-Deliver-TPDU is encoded in accordance with the formatting of Address Fields (refer TS 23.040, section 9.2.3.7). Section 9.1.2.5 in TS 23.040 specifies the formatting of Address Fields. The maximum length of the full address field (Address-Length, Type-of-Address and Address-Value) is 12 octets, which leaves 10 Octets available for Address digits, which may carry 20 digits, which is four digits more than can be conveyed in ISDN-AddressString. Hence, a maximum length Calling Party Number for MT-SMS can't be conveyed in CAP InitialDPSMS and ConnectSMS.
Summary of change: ⌘	Introduce a new data type to allocate all the digits. To do this, AddressString data type is imported from MAP and set the length of eleven octets.
Consequences if not approved: ⌘	Complete CallingPartyNumber may not be informed to the gsmSCF in MT-SMS. This would cause malfunction in the service logic, e.g. screening.
Clauses affected: ⌘	5, 7

	Y	N			
Other specs affected:	⌘		X	Other core specifications	⌘
			X	Test specifications	
			X	O&M Specifications	
Other comments:	⌘				

*** First modified section ***

5 Common CAP Types

5.1 Data types

```
CAP-datatypes {itu-t(0) identified-organization(4) etsi(0) mobileDomain(0) umts-network(1)
modules(3) cap-datatypes(52) version4(3)}

DEFINITIONS IMPLICIT TAGS ::= BEGIN

IMPORTS

    Duration,
    Integer4,
    Interval,
    LegID,
    ServiceKey
FROM CS1-DataTypes {itu-t(0) identified-organization(4) etsi(0) inDomain(1) in-network(1)
modules(0) cs1-datatypes(2) version1(0)}

    BothwayThroughConnectionInd,
    CriticalityType,
    MiscCallInfo
FROM CS2-datatypes {itu-t(0) identified-organization(4) etsi(0) inDomain(1) in-network(1)
cs2(20) modules(0) in-cs2-datatypes(0) version1(0)}

    AddressString,
    IMSI,
    ISDN-AddressString,
    Ext-BasicServiceCode,
    NAEA-CIC
FROM MAP-CommonDataTypes {itu-t(0) identified-organization(4) etsi(0) mobileDomain(0)
gsm-Network(1) modules(3) map-CommonDataTypes(18) version8(8)}

SMS-AddressString ::= AddressString (SIZE (1 .. maxSMS-AddressStringLength))
-- This data type is used to transport CallingPartyNumber for MT-SMS.
-- If this data type is used for MO-SMS, the maximum number of digits shall be 16.

maxSMS-AddressStringLength ::= INTEGER 11
```

*** Next modified section ***

7 SMS Control

This clause defines the protocol used for CAMEL control of MO SMS and MT SMS. CAMEL control of MO SMS uses version 3 of the application context, and CAMEL control of MT SMS uses version 4 of the application context.

7.1 SMS operations and arguments

```
CAP-SMS-ops-args {itu-t(0) identified-organization(4) etsi(0) mobileDomain(0) umts-network(1)
modules(3) cap-SMS-ops-args(105) version4(3)}

DEFINITIONS IMPLICIT TAGS::= BEGIN

-- This module contains the operations and operation arguments used for the
-- smsSSF- gsmSCF interface, for the control of MO-SMS and MT-SMS.

-- The table in subclause 2.1 lists the specifications that contain the modules
-- that are used by CAP.

IMPORTS
```

```
    errortypes,
    datatypes,
    operationcodes,
    classes,
    ros-InformationObjects,
    tc-Messages
FROM CAP-object-identifiers {itu-t(0) identified-organization(4) etsi(0) mobileDomain(0)
umts-network(1) modules(3) cap-object-identifiers(100) version4(3)}

    OPERATION
FROM Remote-Operations-Information-Objects ros-InformationObjects

    ServiceKey
FROM CS1-DataTypes {itu-t(0) identified-organization(4) etsi(0) inDomain(1) in-network(1)
modules(0) cs1-datatypes(2) version1(0)}

    MiscCallInfo
FROM CS2-datatypes {itu-t(0) identified-organization(4) etsi(0) inDomain(1) in-network(1)
cs2(20) modules(0) in-cs2-datatypes (0) version1(0)}

    IMEI,
    IMSI,
    ISDN-AddressString
FROM MAP-CommonDataTypes {itu-t(0) identified-organization(4) etsi(0) mobileDomain(0)
gsm-Network(1) modules(3) map-CommonDataTypes(18) version8(8)}

    GPRSMSClass,
    LocationInformation,
    MS-Classmark2
FROM MAP-MS-DataTypes {itu-t(0) identified-organization(4) etsi(0) mobileDomain(0)
gsm-Network(1) modules(3) map-MS-DataTypes(11) version8(8)}

    PARAMETERS-BOUND
FROM CAP-classes classes

    opcode-connectSMS,
    opcode-continueSMS,
    opcode-eventReportSMS,
    opcode-furnishChargingInformationSMS,
    opcode-initialDPSMS,
    opcode-releaseSMS,
    opcode-requestReportSMSEvent,
    opcode-resetTimerSMS
FROM CAP-operationcodes operationcodes

    CalledPartyBCDNumber {},
    EventSpecificInformationSMS,
    EventTypeSMS,
    Extensions {},
    FCISMSBillingChargingCharacteristics,
    LocationInformationGPRS,
    RPCause,
    SMS-AddressString,
    SMSEvent,
    TimeAndTimezone {},
    TimerID,
    TimerValue,
    TPDataCodingScheme,
    TPProtocolIdentifier,
    TPShortMessageSpecificInfo,
    TPValidityPeriod
FROM CAP-datatypes datatypes

    missingCustomerRecord,
    missingParameter,
    parameterOutOfRange,
    systemFailure,
    taskRefused,
    unexpectedComponentSequence,
    unexpectedDataValue,
    unexpectedParameter
FROM CAP-errortypes errortypes

    CallReferenceNumber
FROM MAP-CH-DataTypes {itu-t(0) identified-organization(4) etsi(0) mobileDomain(0)
gsm-Network(1) modules(3) map-CH-DataTypes(13) version8(8)}

    ;
```

```
connectSMS {PARAMETERS-BOUND : bound} OPERATION ::= {
    ARGUMENT        ConnectSMSArg {bound}
    RETURN RESULT   FALSE
    ERRORS          {missingParameter |
                    parameterOutOfRange |
                    systemFailure |
                    taskRefused |
                    unexpectedComponentSequence |
                    unexpectedDataValue |
                    unexpectedParameter}
    CODE            opcode-connectSMS}
-- Direction: gsmSCF -> gsmSSF or gprsSSF, Timer: T_consms
-- This operation is used to request the smsSSF to perform the SMS processing
-- actions to route
-- or forward a short message to a specified destination.

ConnectSMSArg {PARAMETERS-BOUND : bound} ::= SEQUENCE {
    callingPartysNumber          [0] SMS-AddressStringISDN-AddressString     OPTIONAL,
    destinationSubscriberNumber  [1] CalledPartyBCDNumber {bound}            OPTIONAL,
    sMSCAddress                  [2] ISDN-AddressString                     OPTIONAL,
    extensions                   [10] Extensions {bound}                    OPTIONAL,
    ...
    }

continueSMS OPERATION ::= {
    RETURN RESULT   FALSE
    ALWAYS RESPONDS FALSE
    CODE            opcode-continueSMS}
-- Direction: gsmSCF -> smsSSF, Timer: T_cuesms
-- This operation is used to request the smsSSF to proceed with
-- Short Message processing at the DP at which it previously suspended
-- Short Message processing to await gsmSCF instructions (i.e. proceed
-- to the next Point in Association in the SMS FSM). The smsSSF
-- continues SMS processing without substituting new data from the gsmSCF.

eventReportSMS {PARAMETERS-BOUND : bound} OPERATION ::= {
    ARGUMENT        EventReportSMSArg {bound}
    RETURN RESULT   FALSE
    ALWAYS RESPONDS FALSE
    CODE            opcode-eventReportSMS}
-- Direction: gsmSSF or gprsSSF -> gsmSCF, Timer: T_erbsms
-- This operation is used to notify the gsmSCF of a Short Message related event (FSM events
-- such as submission, delivery or failure) previously requested by the gsmSCF in a
-- RequestReportSMSEvent operation.

EventReportSMSArg {PARAMETERS-BOUND : bound} ::= SEQUENCE {
    eventTypeSMS                    [0] EventTypeSMS,
    eventSpecificInformationSMS     [1] EventSpecificInformationSMS          OPTIONAL,
    miscCallInfo                    [2] MiscCallInfo DEFAULT {messageType request},
    extensions                      [10] Extensions {bound}                 OPTIONAL,
    ...
    }

furnishChargingInformationSMS {PARAMETERS-BOUND : bound} OPERATION ::= {
    ARGUMENT        FurnishChargingInformationSMSArg {bound}
    RETURN RESULT   FALSE
    ERRORS          {missingParameter |
                    taskRefused |
                    unexpectedComponentSequence |
                    unexpectedDataValue |
                    unexpectedParameter}
    CODE            opcode-furnishChargingInformationSMS}
-- Direction: gsmSCF -> gsmSSF or gprsSSF, Timer: T_fcisms
-- This operation is used to request the smsSSF to generate, register a charging record
-- or to include some information in the default SM record. The registered charging record is
-- intended for off line charging of the Short Message.

FurnishChargingInformationSMSArg {PARAMETERS-BOUND : bound} ::=
    FCISMSBillingChargingCharacteristics {bound}

initialDPSMS {PARAMETERS-BOUND : bound} OPERATION ::= {
    ARGUMENT        InitialDPSMSArg {bound}
    RETURN RESULT   FALSE
    ERRORS          {missingCustomerRecord |
                    missingParameter |
                    parameterOutOfRange |
```

```
                        systemFailure |
                        taskRefused |
                        unexpectedComponentSequence |
                        unexpectedDataValue |
                        unexpectedParameter}
        CODE            opcode-initialDPSMS}
-- Direction: gsmSSF or gprsSSF -> gsmSCF, Timer: T_idpsms

-- This operation is used after a TDP to indicate request for service.

InitialDPSMSArg {PARAMETERS-BOUND : bound} ::= SEQUENCE {
    serviceKey                          [0] ServiceKey,
    destinationSubscriberNumber         [1] CalledPartyBCDNumber {bound}           OPTIONAL,
    callingPartyNumber                  [2] SMS-AddressStringISDN-AddressString
    OPTIONAL,
    eventTypeSMS                        [3] EventTypeSMS                           OPTIONAL,
    iMSI                                [4] IMSI                                   OPTIONAL,
    locationInformationMSC              [5] LocationInformation                    OPTIONAL,
    locationInformationGPRS             [6] LocationInformationGPRS                OPTIONAL,
    sMSCAddress                         [7] ISDN-AddressString                     OPTIONAL,
    timeAndTimezone                     [8] TimeAndTimezone {bound}                OPTIONAL,
    tPShortMessageSpecificInfo          [9] TPShortMessageSpecificInfo             OPTIONAL,
    tPProtocolIdentifier                [10] TPProtocolIdentifier                  OPTIONAL,
    tPDataCodingScheme                  [11] TPDataCodingScheme                    OPTIONAL,
    tPValidityPeriod                    [12] TPValidityPeriod                      OPTIONAL,
    extensions                          [13] Extensions {bound}                    OPTIONAL,
    ...,
    smsReferenceNumber                  [14] CallReferenceNumber                   OPTIONAL,
    mscAddress                          [15] ISDN-AddressString                    OPTIONAL,
    sgsn-Number                         [16] ISDN-AddressString                    OPTIONAL,
    ms-Classmark2                       [17] MS-Classmark2                         OPTIONAL,
    gPRSMSClass                         [18] GPRSMSClass                           OPTIONAL,
    iMEI                                [19] IMEI                                  OPTIONAL,
    calledPartyNumber                   [20] ISDN-AddressString                    OPTIONAL
    }

releaseSMS OPERATION ::= {
    ARGUMENT        ReleaseSMSArg
    RETURN RESULT   FALSE
    ALWAYS RESPONDS FALSE
    CODE            opcode-releaseSMS}
-- Direction: gsmSCF -> gsmSSF or gprsSSF, Timer: T_relsms

-- This operation is used to prevent an attempt to submit or deliver a short message.

ReleaseSMSArg ::= RPCause

requestReportSMSEvent {PARAMETERS-BOUND : bound} OPERATION ::= {
    ARGUMENT        RequestReportSMSEventArg {bound}
    RETURN RESULT   FALSE
    ERRORS          {missingParameter |
                    parameterOutOfRange |
                    systemFailure |
                    taskRefused |
                    unexpectedComponentSequence |
                    unexpectedDataValue |
                    unexpectedParameter}
    CODE            opcode-requestReportSMSEvent}
-- Direction: gsmSCF -> gsmSSF or gprsSSF, Timer: T_rrbsms

-- This operation is used to request the gsmSSF or gprsSSF to monitor for a
-- Short Message related event (FSM events such as submission, delivery or failure)
-- and to send a notification to the gsmSCF when the event is detected.

RequestReportSMSEventArg {PARAMETERS-BOUND : bound} ::= SEQUENCE {
    sMSEvents                           [0] SEQUENCE SIZE (1..bound.&numOfSMSEvents) OF SMSEvent,
    extensions                          [10] Extensions {bound}                    OPTIONAL,
    ...
    }
-- Indicates the Short Message related events(s) for notification.

resetTimerSMS {PARAMETERS-BOUND : bound} OPERATION ::= {
    ARGUMENT        ResetTimerSMSArg {bound}
    RETURN RESULT   FALSE
    ERRORS          {missingParameter |
                    parameterOutOfRange |
                    taskRefused |
                    unexpectedComponentSequence |
                    unexpectedDataValue |
                    unexpectedParameter}
```

```
    CODE              opcode-resetTimerSMS}
-- Direction: gsmSCF -> smsSSF, Timer: T_{rtsms}
-- This operation is used to request the smsSSF to refresh an application
-- timer in the smsSSF.

ResetTimerSMSArg {PARAMETERS-BOUND : bound} ::= SEQUENCE {
    timerID                          [0] TimerID DEFAULT tssf,
    timervalue                       [1] TimerValue,
    extensions                       [2] Extensions {bound}                OPTIONAL,
    ...
    }

END
```

*** End of document ***