INTERNATIONAL TELECOMMUNICATION UNION

ITU-T

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

Q.1228

Fascicle 1/5

(09/97)

SERIES Q: SWITCHING AND SIGNALLING

Intelligent Network

Interface Recommendation for intelligent network Capability Set 2: Part 1

ITU-T Recommendation Q.1228 – Fascicle 1/5

(Previously CCITT Recommendation)

ITU-T RECOMMENDATION Q.1228

INTERFACE RECOMMENDATION FOR INTELLIGENT NETWORK CAPABILITY SET 2

Summary

Recommendation Q.1228 defines the Intelligent Network (IN) Application Protocol (INAP) for IN Capability Set 2 (IN CS-2). This Recommendation defines the INAP for IN CS-2 based upon the IN CS-1 refined (CS-1R) Q.1218 specification (1995) and the general rules for INAP provided in Recommendation Q.1208, consistent with the scope of IN CS-2 defined in Recommendation Q.1221.

Recommendation Q.1228 provides:

- general extensions to the CS-1R INAP in support of IN CS-2 target services including: SCF initiated trigger management, GVNS service support, explicit ISDN supplementary service feature interaction support, service compatibility checks, general User to Service Interaction (UTSI), Specialized Resource Function (SRF) script processing, enhanced security, extended BCSM support, message store and forward, SCF/SDF extensions in support of distributed data management;
- protocol support for Call Party Handling capabilities based, in part, on future study items identified in CS-1R;
- protocol support for the Service Control Function (SCF) to SCF and Service Data Function (SDF) to SDF functional relationships to support distributed service logic execution and distributed data functions;
- protocol support for the Call Unrelated Service Function (CUSF) to Service Control Function (SCF) to support non-call related interactions between users and the SCF;
- additional details on services assumed from lower layers and generic interface security;
- validated SDLs for SSF-related procedure handling based upon Z.100 object-oriented Specification and Description language.

Within the Q.122x Recommendation series, Recommendation Q.1228 describes the protocol realizing the Q.1224 distribution of Q.1223 Global Functional Plane functionality in a service and vendor/implementation independent manner, as constrained by the capabilities of the embedded base of evolvable network technology. This provides the flexibility to allocate distributed functionality into multiple physical network configurations, as described in Recommendation Q.1225, and to evolve IN from IN CS-2 to some future CS-N.

Source

ITU-T Recommendation Q.1228 was prepared by ITU-T Study Group 11 (1997-2000) and was approved under the WTSC Resolution No. 1 procedure on the 12th of September 1997.

CONTENTS

Page

PART 3

Page

Page

Page

Page

Page

Page

Page

Recommendation Q.1228

INTERFACE RECOMMENDATION FOR INTELLIGENT NETWORK CAPABILITY SET 2

(Geneva, 1997)

PART 1

1 Introduction

This Recommendation defines the INAP (Intelligent Network Application Protocol) required for support of Intelligent Network Capability Set 2. It supports interactions between the following Functional Entities (FEs), as defined in the IN functional model:

– Service Switching Function (SSF).

– Service Control Function (SCF).

– Specialized Resource Function (SRF).

– Service Data Function (SDF).

– Call Unrelated Service Function (CUSF).

The scope of this Recommendation is the further development of the INAP for both the Integrated Services Digital Network (ISDN) and Public Switched Telephone Network (PSTN).

It is intended as a guide to implementors and network operators to ensure interworking between different manufacturers equipment for all the IN CS-2 defined interfaces and between network operators for the internetwork interface.

As this Recommendation is intended for the early introduction of IN in the existing ISDN/PSTN, only simple solutions are assumed for solving the service interaction problems between IN and ISDN/PSTN.

NOTE – More sophisticated solutions for the service interactions between IN and the ISDN/PSTN environment should be studied in the scope of future versions of INAP and the ISDN/PSTN signalling standards.

2 General

2.1 Normative references

The following ITU-T Recommendations and other references contain provisions which, through references in this text, constitute provisions of this Recommendation. At the time of adoption of this ITU-T Recommendation, the reference editions indicated were valid. Recalling that all Recommendations and other material incorporated by referene herein are subject to future revision, all users of this Recommendation are therefore advised that changes in the reference text that constitue future decisions of the work of Organizations or Study Groups other than ITU Study Group 11, do not automatically apply as amended provisions of this Recommendation.

– CCITT Recommendation Q.29 (1988), *Causes of noise and ways of reducing noise in telephone exchanges.*

- ITU-T Recommendation Q.711 (1993), *Signalling System No. 7 – Functional description of the signalling connection control part.*
- ITU-T Recommendation Q.713 (1993), *Signalling System No. 7 – SCCP Formats and codes.*
- ITU-T Recommendation Q.715 (1996), *Signalling connection control part user guide.*
- ITU-T Recommendation Q.762 (1993), *General function of messages and signals of the ISDN User Part of Signalling System No. 7.*
- ITU-T Recommendation Q.763 (1993), *Formats and codes of the ISDN User Part of Signalling System No. 7.*
- ITU-T Recommendation Q.771 (1993), *Signalling System No. 7 – Functional description of transaction capabilities.*
- ITU-T Recommendation Q.772 (1993), *Signalling System No. 7 – Transaction capabilities information elements definitions.*
- ITU-T Recommendation Q.773 (1993), *Signalling System No. 7 – Transaction capabilities formats and encoding.*
- ITU-T Recommendation Q.774 (1993), *Signalling System No. 7 – Transaction capabilities procedures.*
- ITU-T Recommendation Q.775 (1993), *Signalling System No. 7 – Guidelines for using transaction capabilities.*
- ITU-T Recommendation Q.931 (1993), *ISDN user-network interface layer 3 specification for basic call control.*
- ITU-T Recommendation Q.932 (1993), *Generic procedures for the control of ISDN supplementary services.*
- ITU-T Recommendation Q.1290 (1993), *Glossary of terms used in the definition of intelligent networks.*
- ITU-T Recommendation X.500 (1997) | ISO/IEC 9594-1:1997, *Information technology – Open Systems Interconnection – The Directory: Overview of concepts, models and services.*
- ITU-T Recommendation X.501 (1997) | ISO/IEC 9594-2:1997, *Information technology – Open Systems Interconnection – The Directory: The Models.*
- ITU-T Recommendation X.509 (1997) | ISO/IEC 9594-8:1997, *Information technology – Open Systems Interconnection – The Directory – Authentication framework.*
- ITU-T Recommendation X.511 (1997) | ISO/IEC 9594-3:1997, *Information technology – Open Systems Interconnection – The Directory: Abstract service definition.*
- ITU-T Recommendation X.518 (1993) | ISO/IEC 9594-4:1995, *Information technology – Open Systems Interconnection – The Directory: Procedures for distributed operation.*
- ITU-T Recommendation X.519 (1993) | ISO/IEC 9594-5:1993, *Information technology – Open Systems Interconnection – The Directory: Protocol specifications.*
- ITU-T Recommendation X.680 (1994) | ISO/IEC 8824-1:1995, *Information technology – Abstract Syntax Notation One (ASN.1): Specification of basic notation.*
- ITU-T Recommendation X.681 (1994) | ISO/IEC 8824-2:1995, *Information technology – Abstract Syntax Notation One (ASN.1): Information object specification.*
- ITU-T Recommendation X.682 (1994) | ISO/IEC 8824-3:1995, *Information technology – Abstract Syntax Notation One (ASN.1): Constraint specification.*

- ITU-T Recommendation X.683 (1994) | ISO/IEC 8824-4:1995, *Information technology – Abstract Syntax Notation One (ASN.1): Parameterization of ASN.1 specifications.*
- ITU-T Recommendation X.690 (1994) | ISO/IEC 8825-1:1995, *Information technology – ASN.1 encoding rules: Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER).*
- ITU-T Recommendation X.830 (1995) | ISO/IEC 11586-1:1996, *Information technology – Open Systems Interconnection – Generic upper layers security: Overview, models and notation*.
- ITU-T Recommendation X.831 (1995) | ISO/IEC 11586-2:1996, *Information technology – Open Systems Interconnection – Generic upper layers security: Security Exchange Service Element (SESE) service definition*.
- ITU-T Recommendation X.832 (1995) | ISO/IEC 11586-3:1996, *Information technology – Open Systems Interconnection – Generic upper layers security: Security Exchange Service Element (SESE) protocol specification*.
- ITU-T Recommendation X.880 (1994) | ISO/IEC 13712-1:1995, *Information technology – Remote Operations: Concepts, model and notation.*
- Internet Engineering Task Force (IETF), Request For Comment (RFC) 2025: *The Simple Public-Key GSS-API Mechanism (SPKM), October 1996*. ftp://ds.internic.net/rfc/rfc2025.txt

2.2 Abbreviations and acronyms

This Recommendation uses the following abbreviations.

AC	Application Context
ACN	Application Context Negotiation
ACSE	Application Control Service Element
AD	Adjunct
ADSI	Analogue Display Service Interface Server
AE	Application Entity
AEI	Application Entity Invocation
AOC	Advice of Charge
APC	Apply Charging
APCI	Application Protocol Control Information
APDU	Application Protocol Data Unit
API	Application Programming Interface
APR	Apply Charging Report
ASE	Application Service Element
ASR	Automatic Speech Recognition
BCP	Basic Call Process
BCSM	Basic Call State Model
BCUP	Basic Call Unrelated Process

BCUSM	Basic Call Unrelated State Model
BGID	Business Group Identity
BRI	Basic Rate Interface
CAC	Carrier Access Code
CCAF	Call Control Agent Function
CCF	Call Control Function
CDP	Customized Dialling Plan
CHA	Component Handler
CID	Call Instance Data
CM	Call Manager
CMIS	Common Management Information System
CPH	Call Party Handling
CS	Call Segment
CS	Capability Set
CSA	Call Segment Association
CSM	Call Segment Model
CUSF	Call Unrelated Service Function
CVS	Connection View State
DAP	Directory Access Protocol
DET	Determination
DFP	Distributed Functional Plane
DHA	Dialogue Handler
DLE	Destination Local Exchange
DN	Directory Number
DN	Distinguished Name
DP	Detection Point
DSA	Directory System Agent
DSL	Distributed Service Logic
DSP	Directory System Protocol
DSS 1	Digital Subscriber Signalling No. 1
DTMF	Dual Tone Multi Frequency
DUA	Directory User Agent
EDP	Event Detection Point
EDP-N	Event Detection Point-Notification
EDP-R	Event Detection Point-Request
EUI	Extended User Interface Server

FCI	Furnish Charging Information
FEA	Functional Entity Action
FEAM	Functional Entity Access Manager
FIM	Feature Interactions Manager
FRL	Facility Restriction Level
FSM	Finite State Machine
GEN	Generation
GFP	Global Functional Plane
GSL	Global Service Logic
GVNS	Global Virtual Network Services
HLSIB	High Level Service Independent Block
IAF	Intelligent Access Function
IEC	International Electrotechnical Commission
IMT-2000	International Mobile Telecommunications-2000
IN	Intelligent Network
INAP	Intelligent Network Application Protocol
INCM	IN Conceptual Model
INDB	IN Data Base
INDBMS	IN Data Base Management System
IN-SM	IN Switching Manager
IN-SSM	IN Switching State Model
IP	Intelligent Peripheral
ISDN	Integrated Services Digital Network
ISDN-UP	ISDN User Part
ISO	International Organization for Standardization
ISUP	Integrated Services Digital Network-User Part
ISUP	ISDN-UP
ITU-T	International Telecommunication Union – Telecommunication Standardization Sector
LE	Local Exchange
MACF	Multiple Association Control Function
MSR	Message Storage and Retrieval
NAP	Network Access Point
NEF	Network Element Function
NFA	Network Functional Architecture
NM	Network Manager

NSAP	Network Service Access Point
OFC	Off-line Charging (billing/accounting information)
OLE	Originating Local Exchange
OLI	Originating Line Information
ONC	On-line Charging (user access information)
OSF	Operator System Function
OSI	Open Systems Interconnection
OUT	Output
PIC	Point in Call
PM	Personal Mobility
POC	Point of Control
POI	Point of Initiation
POR	Point of Return
POS	Point of Synchronisation
PRI	Primary Rate Interface
PSTN	Public Switched Telephone Network
PTNX	Private Telecommunications Network Exchange
RCP	Resource Control Part
RDN	Relative Distinguished Name
REG	Registration
RFP	Resource Function Part
RLF	Radio Link Function
ROA	Recognized Operating Agency
ROS	Remote Operations
ROSE	Remote Operations Service Element
SACF	Single Association Control Function
SAO	Single Association Object
SCE	Service Creation Environment
SCEF	Service Creation Environment Function
SCEP	Service Creation Environment Point
SCF	Service Control Function
SCF FSM	Service Control Function Finite State Machine
SCFID	Service Control Function Identifier
SCI	Send Charging Information
SCME	Service Control Function Management Entity
SCME FSM	Service Control Function Management Entity Finite State Machine

SCP	Service Control Point
SCSM	Service Control Function Call State Model
SDF FSM	Service Data Function Finite State Machine
SDF	Service Data Function
SDL	Specification and Description Language
SDME	Service Data Function Management Entity
SDP	Service Data Point
SDSM	Service Data Function Call State Model
SF	Service Feature
SIB	Service Independent Building Block
SL	Service Logic
SLCP	Service Logic Control Program
SLMP	Service Logic Management Program
SLP	Service Logic Processing Program
SLPI	Service Logic Processing Program Instance
SM	Service Manager
SMAF	Service Management Access Function
SMF	Service Management Function
SMP	Service Management Point
SMS	Service Management System
SN	Service Node
SRF	Specialized Resource Function
SRF FSM	Specialized Resource Function Finite State Machine
SRME	Specialized Resource Function Management Entity
SRSM	Specialized Resource Function Call State Model
SS	Service Subscriber
SS7	Signalling System No. 7
SSCP	Service Switching and Control Point
SSD	Service Support Data
SSF	Service Switching Function
SSF FSM	Service Switching Function Finite State Machine
SSME	Service Switching Function Management Entity
SSME FSM	Service Switching Function Management Entity Finite State Machine
SSP	Service Switching Point
STI	Service Trigger Information
SU	Service User

TC	Transaction Capabilities
TCAP	Transaction Capabilities Application Part
TDP	Trigger Detection Point
TDP-N	Trigger Detection Point- Notification
TDP-R	Trigger Detection Point-Request
TMN	Telecommunications Management Network
TTS	Text-to-Speech Synthesis
UPT	Universal Personal Telecommunication
VPN	Virtual Private Network
WCR	Wireless Call Related
WCU	Wireless Call Unrelated

2.3 Conventions

For the finite state machines found in clauses 11 through 15 inclusive, events are enumerated. The number of an event is prefixed with either the letter "E" (for external events) or "e" (for internal ones) and included in parentheses in the beginning of the event name. The scope of event names and numbers is defined by the state machine in which these events appear; the same applies to state names.

3 Interface recommendation for telecommunication services

3.1 General

3.1.1 Definition methodology

The definition of the protocol can be split into three sections:

– the definition of the SACF/MACF rules for the protocol ;

– the definition of the operations transferred between entities ;

– the definition of the actions taken at each entity.

The SACF/MACF rules are defined in prose. The operation definitions are in Abstract Syntax Notation One (ASN.1, see Recommendation X.680), and the actions are defined in terms of state transition diagrams. Further guidance on the actions to be performed on receipt of an operation can be gained from the description of the relevant information flow in Recommendation Q.1224.

The INAP is a ROSE user protocol (see Recommendatons X.219 and X.229). The ROSE protocol is contained within the component sublayer of TCAP (see Recommendations Q.771 to Q.775) and DSS 1 (Recommendation Q.932). At present the ROSE APDUs (Application Protocol Data Units) are conveyed in transaction sublayer messages in SS No. 7 and in the Q.931 REGISTER, FACILITY and call control messages in DSS 1. Other supporting protocols may be added at a later date.

The INAP (as a ROSE user) and the ROSE protocol have been specified using ASN.1 (see Recommendation X.680). The encoding of the resulting PDUs should use the Basic Encoding Rules (see Recommendation X.690).

3.1.2 Example physical scenarios

The protocol will support any mapping of functional to Physical Entities (PEs). It is the responsibility of network operators and equipment manufacturers to decide how to co-locate FEs to the best possible advantage as this may vary between manufacturers and between network operators. Therefore the protocol is defined assuming maximum distribution (i.e. one PE per FE).

The figures depicted in this subclause show how INAP would be supported in an SS No. 7 network environment. This does not imply that only SS No. 7 may be used as the network protocol to support INAP.

The interface between remotely located SCF and SDF will be INAP using TCAP which in turn, uses the services of the connectionless SCCP and MTP (see Figure 3-1). The SDF is responsible for any interworking to other protocols to access other types of networks.

When TCAP appears in one of the following figures, it shall be understood as representing the TCAP functionalities associated with a single dialogue and transaction (as opposed to a TCAP entity).

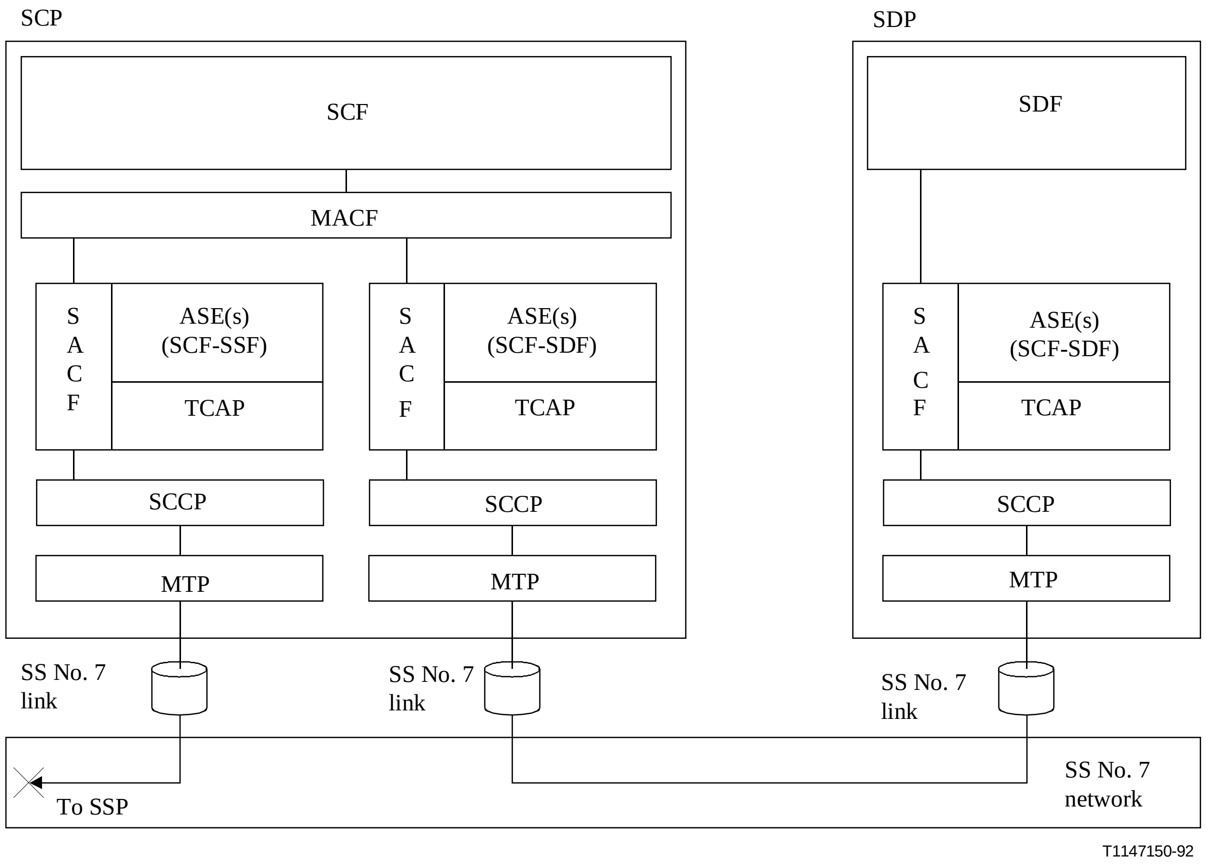


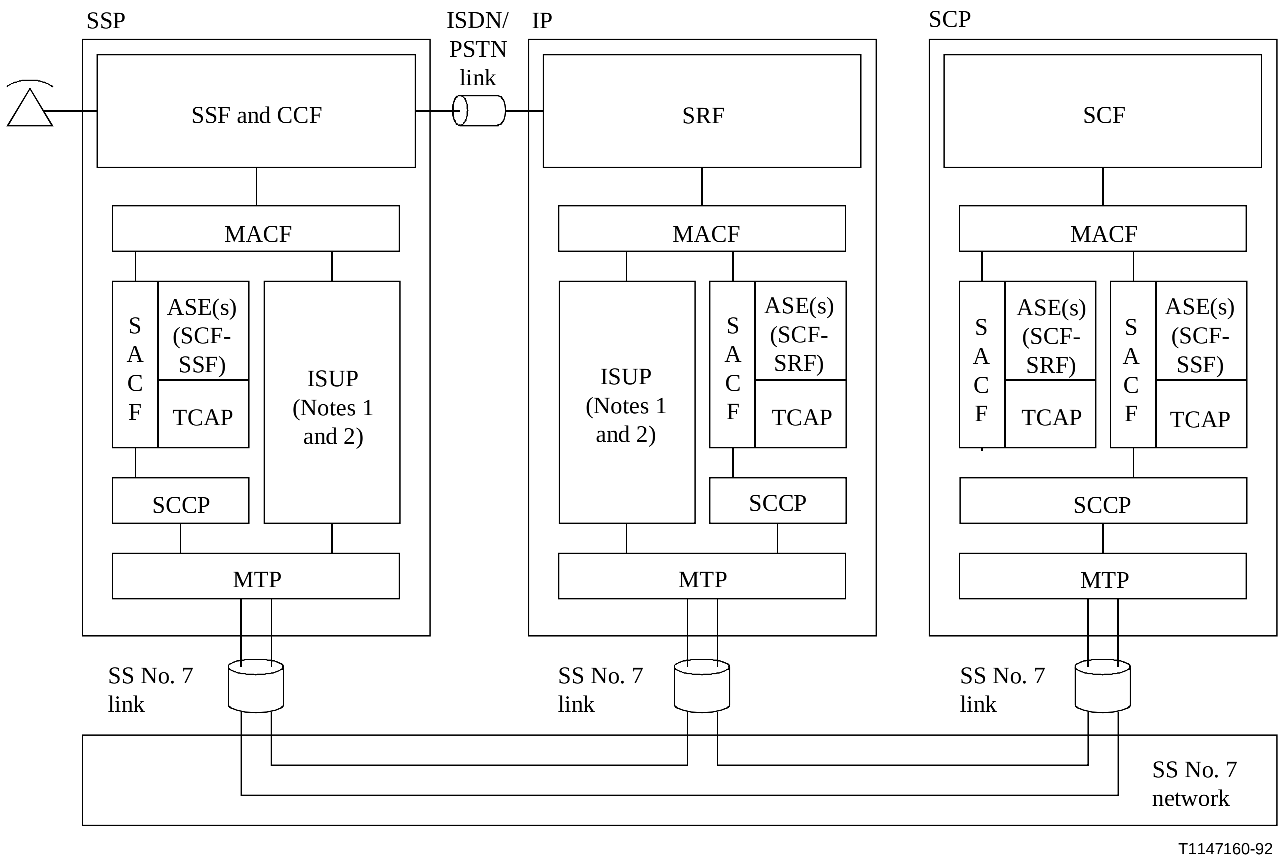
Figure 3-1/Q.1228 – Physical interface between SCP and SDP

If segmentation and re-assembly of INAP messages is required on the SCF-to-SDF interface (and on other interfaces, if needed) due to the length of messages, the segmentation and re-assembly procedure for SCCP connectionless messages, as specificed in Recommendation Q.714, should be used.

A number of example scenarios have been identified for support of the SCF, SSF and SRF functional entities as physical entities. These are illustrated as Figures 3-2 to 3-6. Each example is characterised by:

i) the method to support SCF-SRF relationship; and

ii) the type of signalling system between SSF and SRF.

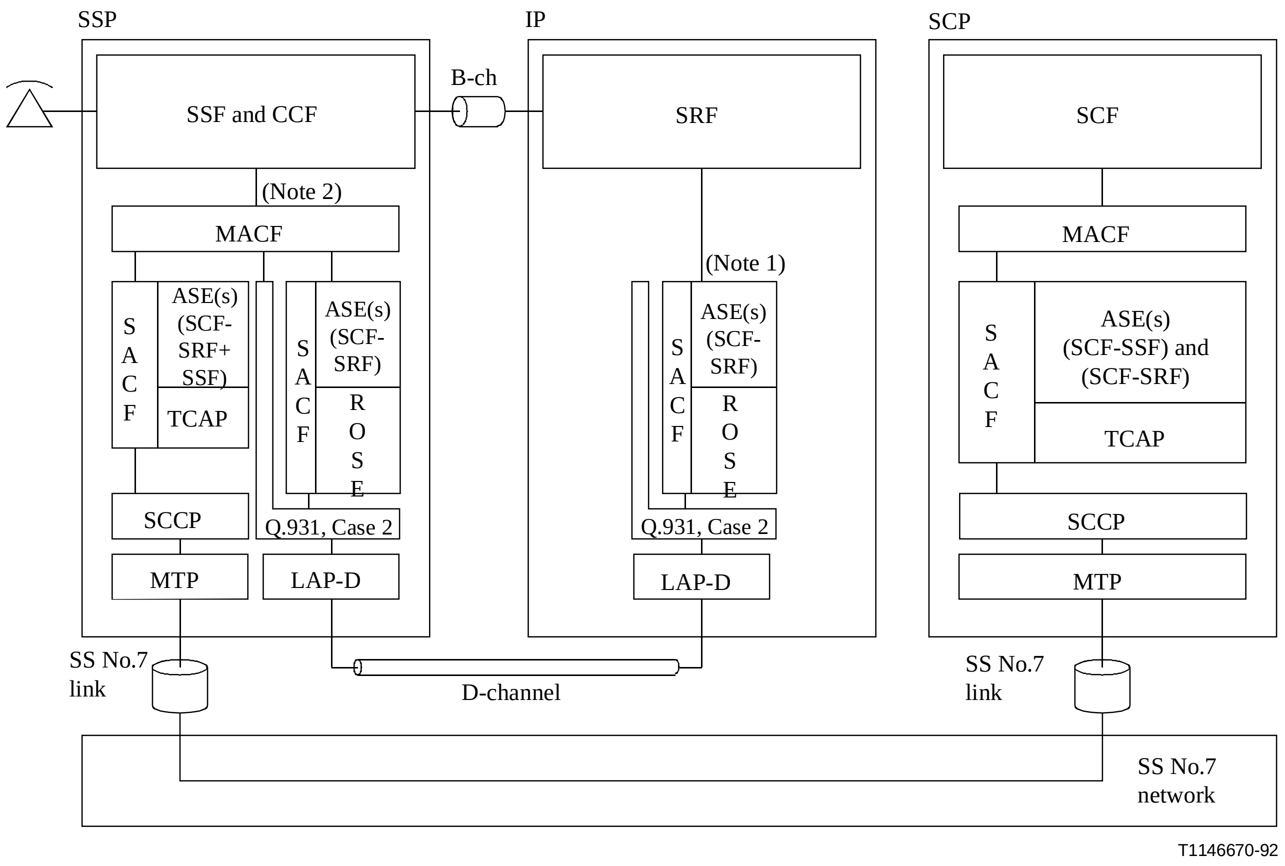


NOTE 1 – Transfer of correlation information needs to be supported. This may be supported in ISUP without introducing new ISUP parameter.

NOTE 2 – Other signalling systems may be used.

NOTE 3 – The IP can be integrated into a local exchange, or indirectly attached via a local exchange to the SSP that is interacting with the SCP.

Figure 3-2/Q.1228 – Example architecture for supporting SRF, Case 1 (SRF in IP connected to SSP and accessed by SCP through direct SS No. 7 connection)



NOTE 1 – Info flows between SCF and SRF are supported by this (ROSE) entity.

NOTE 2 – Relay function is provided either by MACF or by application process at SSP.

Figure 3-3/Q.1228 – Example architecture for supporting SRF, Case 2 (SRF in IP connected to SSP and accessed by SCP through D-channel via SSP)

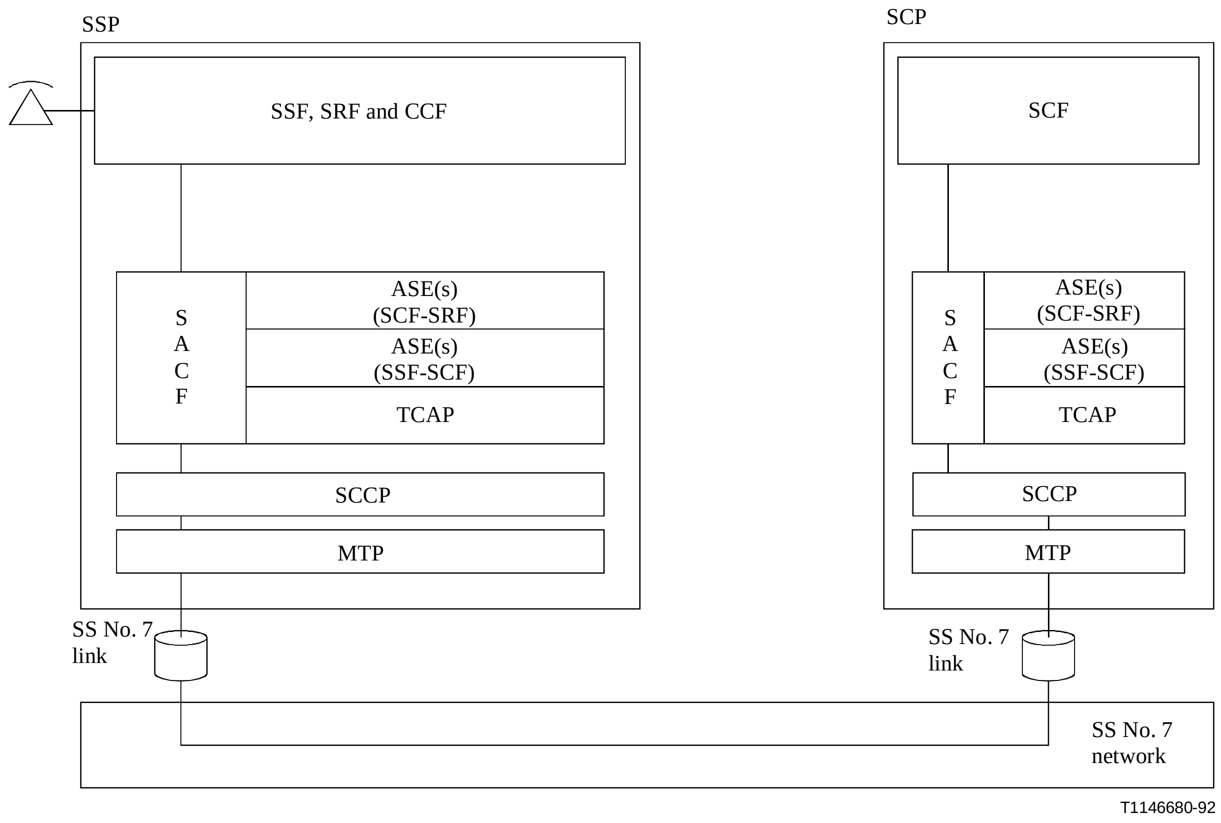
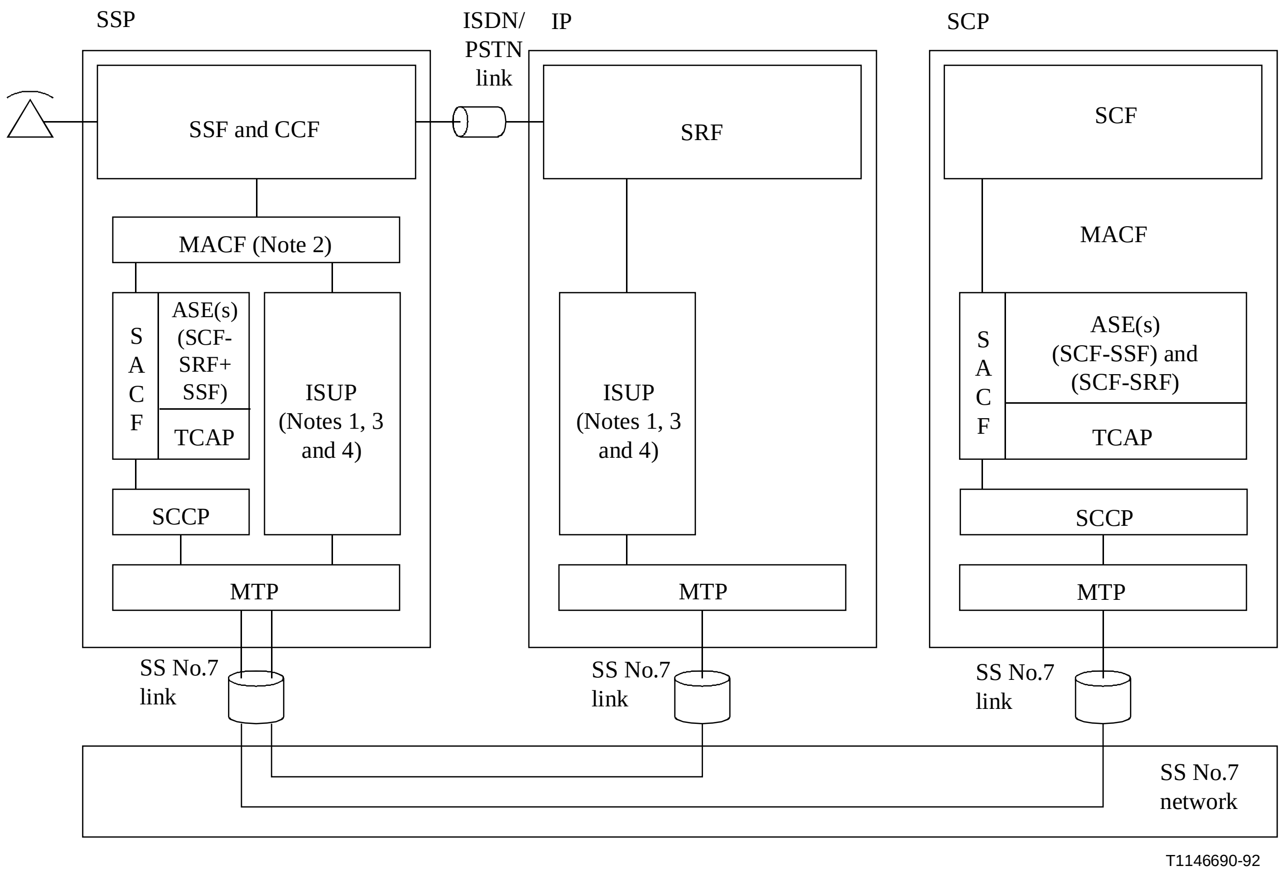


Figure 3-4/Q.1228 – Example architecture for supporting SRF, Case 3 (SRF in SSP and accessed via AP of SSP)



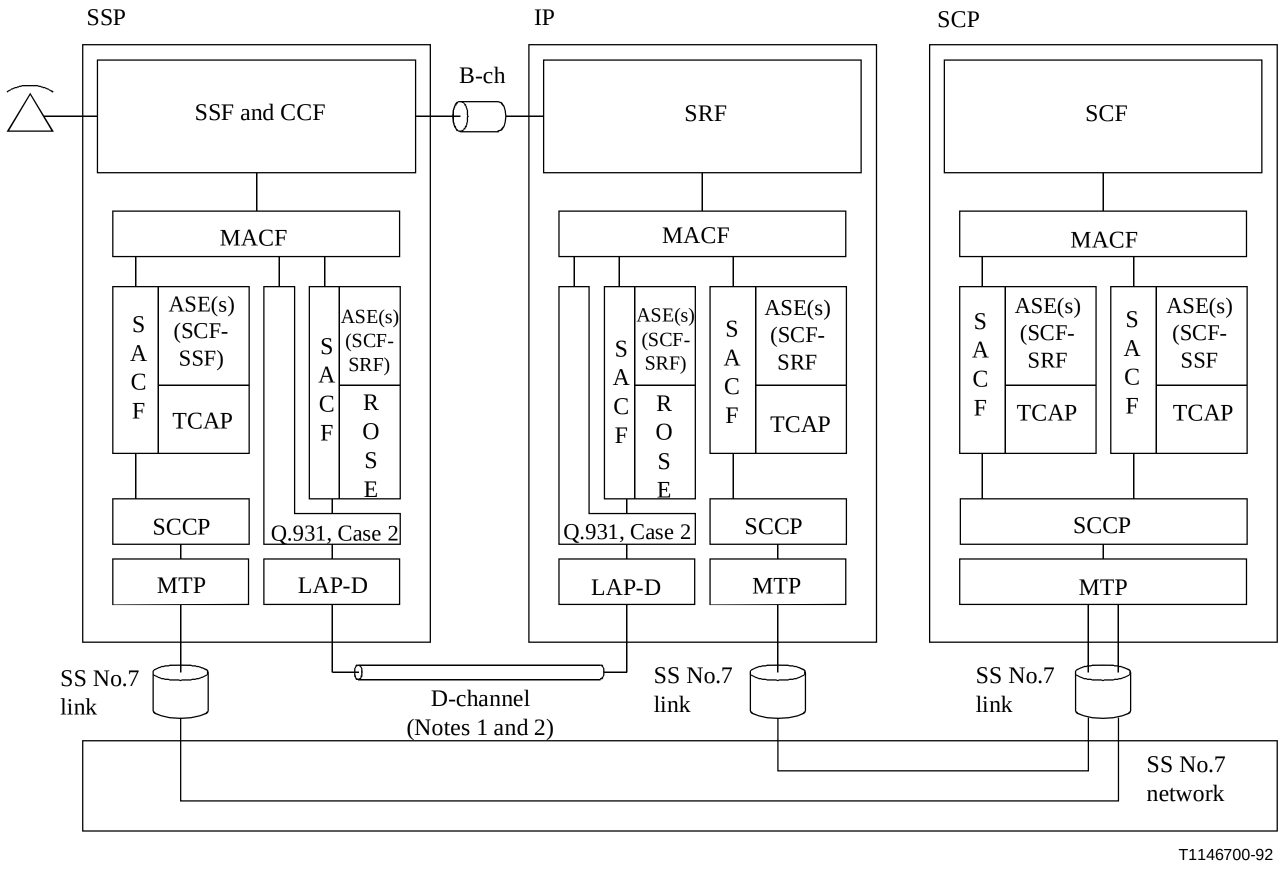
NOTE 1 – Info flows between SCF and SRF as well as connection control are directly supported by ISUP.

NOTE 2 – Relay function is provided either by MACF or by application process at SSP.

NOTE 3 – Assumes that ISUP porvides a means to transport ROSE information.

NOTE 4 – Other signalling systems may be used.

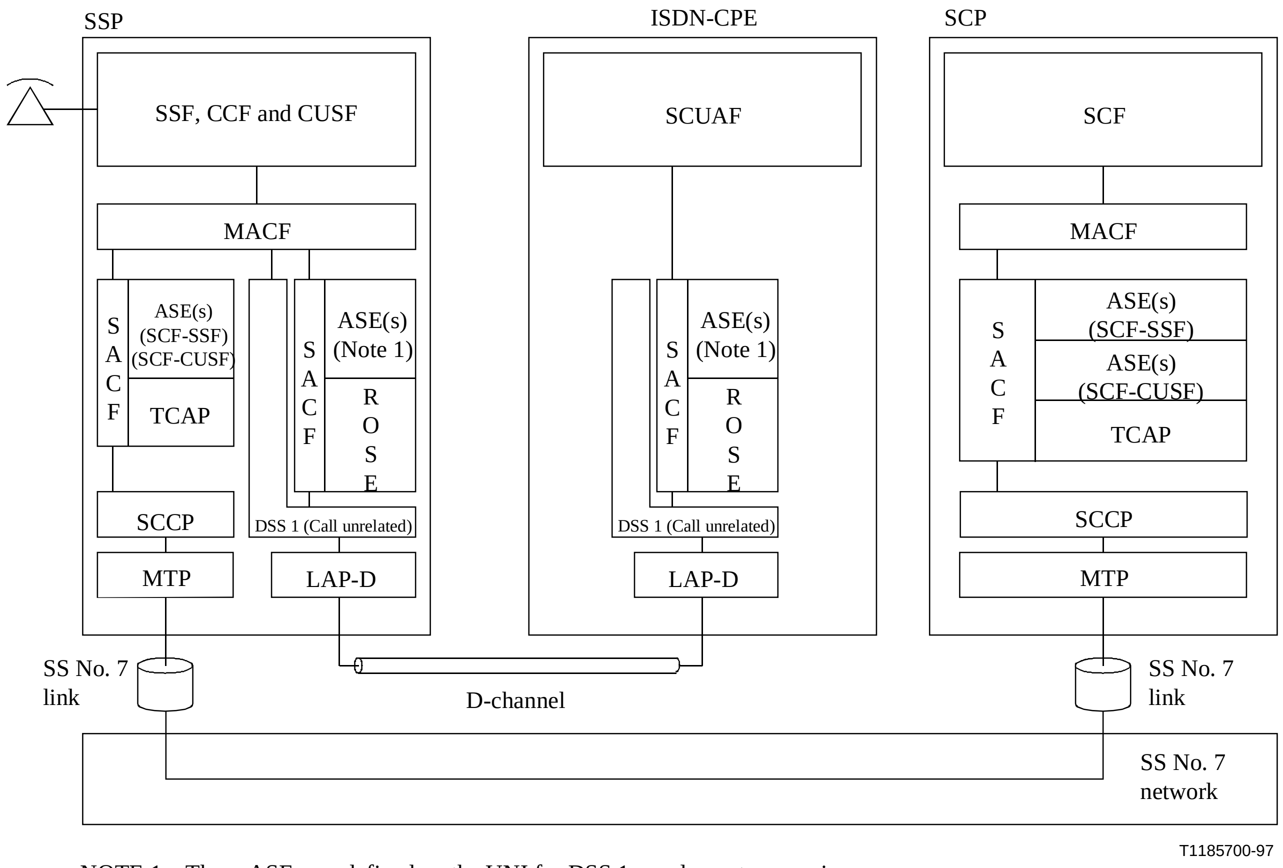
Figure 3-5/Q.1228 – Example architecture for supporting SRF, Case 4 (SRF in IP connected to SSP and accessed by SCP through ISUP via SSP)



NOTE 1 – Transfer of correlation information needs to be supported.
NOTE 2 – Other signalling systems may be used.

Figure 3-6/Q.1228 – Example architecture for supporting SRF, Case 5 (SRF in IP connected to SCP and SSP and accessed via both SS No. 7 and D-channel respectively)

As there may be several configurations for the SRF mapping, Figure 3-7 does not mention all possible architecture but a possible stack for the SSP with the CUSF, the CCF and the SSF.



NOTE 1 – These ASEs are defined on the UNI for DSS 1 supplementary services.
NOTE 2 – IP is omitted for simplicity, however SSP has ISUP/DSS 1 link to IP.

Figure 3-7/Q.1228 – Example architecture focusing on the CUSF (CUSF being mapped to SSP with SSF and CCF)

Table 3-1 summarises the selection of features for each figure.

Table 3-1/Q.1228

Type of signalling system between SSF and SRF	**Method to support SCF-SRF relationship**	
	Direct TCAP link	**Relay via SSP**
ISUP	Figure 3-2[a)]	Figure 3-5[d)]
DSS 1	Figure 3-6[e)]	Figure 3-3[b)]
Implementation dependent	As Figure 3-2 or 3-6 but with implementation dependent SCP-IP interface	Figure 3-4[c)]

Additional information related to each figure:

a) Figure 3-2/Q.1228: All associations are supported by SS No. 7, either TCAP or ISUP. In this case the IP is one of the network nodes.

b) Figure 3-3/Q.1228: IP can be accessed by DSS 1 only. The IP can be a physical entity residing outside the network.

c) Figure 3-4/Q.1228: SSP supports both CCF/SSF and SRF. The handling of SRF by SCF could be the same as the of Figure 3-3/Q.1228.

d) Figure 3-5/Q.1228: IP can be accessed by ISUP only. The handling of SRF by SCF could be the same as that of Figure 3-3/Q.1228.

e) Figure 3-6/Q.1228: The handling of SRF by SCF could be the same as that of Figure 3-2/Q.1228. Other types of signalling systems could be used.

3.1.2.1 "SCF-External SRF" Communication in the Relay Case

In the Relay case, when the SCF uses the *ConnectToResource* operation to connect to an External SRF, the SCF and the SRF embed the "User Interaction" operations exchanged with each other using the "Out-Channel Call Related User Interaction" operations: *SendSTUI*, *ReportUTSI* and *RequestReportUTSI.*

In this case, it is necessary to affect a new value of the *serviceIndicator* parameter for the "External SRF connection": *SRF_Connection.* As in CS-1, the "External SRF connection" is not modelled at the SSF level. Once receiving the *SendSTUI* (resp. *RequestReportUTSI*) operation from the SCF with a *serviceIndicator* parameter value set to *SRF_Connection*, the SSF checks only this parameter to decide that this operation is related to the "SCF-External SRF communication". The same processing applies for the *reportUTSI* operation in the "SRF to SCF" direction.

At a time, only one party (Called Party or Calling Party) can be connected to the SRF.

The following MSC illustrates the User Interaction in the Relay case (see Figure 3.8):

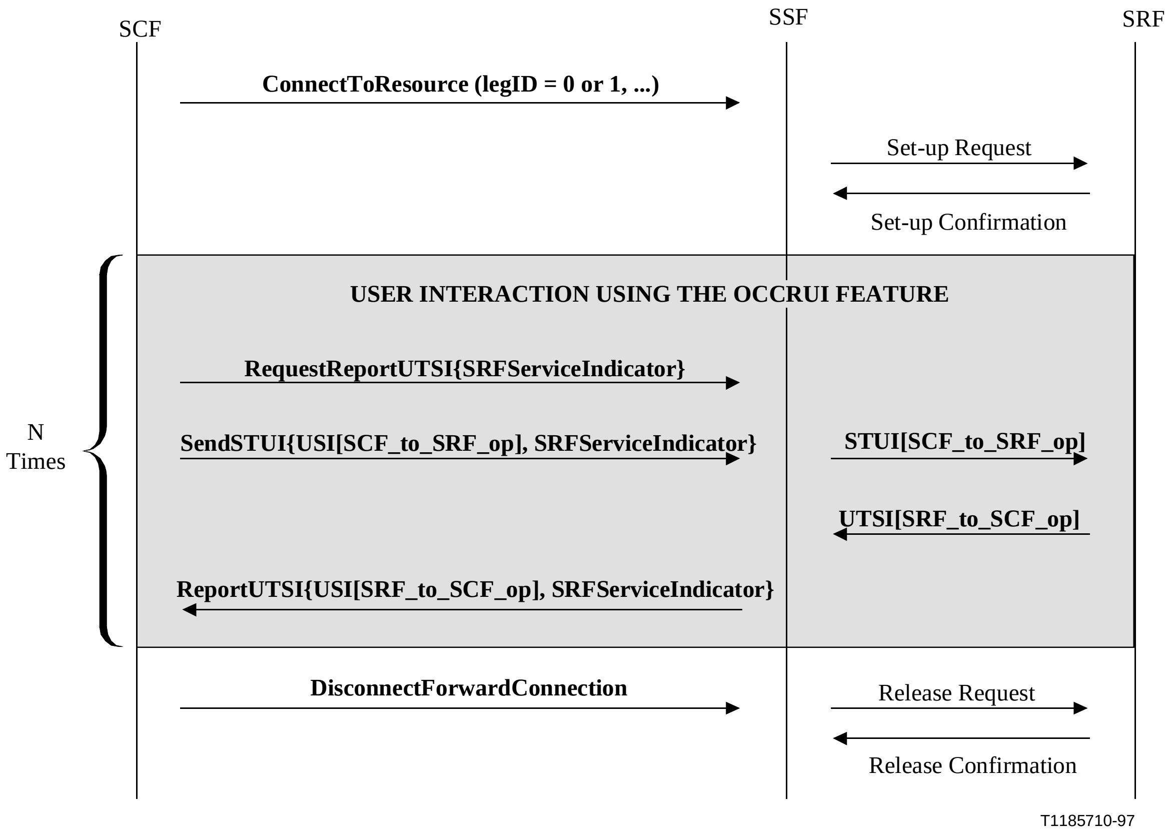

SCF
SSF
SRF
ConnectToResource (legID = 0 or 1, ...)
Set-up Request
Set-up Confirmation
USER INTERACTION USING THE OCCRUI FEATURE
N Times
RequestReportUTSI{SRFServiceIndicator}
SendSTUI{USI[SCF_to_SRF_op], SRFServiceIndicator}
STUI[SCF_to_SRF_op]
UTSI[SRF_to_SCF_op]
ReportUTSI{USI[SRF_to_SCF_op], SRFServiceIndicator}
DisconnectForwardConnection
Release Request
Release Confirmation
T1185710-97

Figure 3-8/Q.1228 – User Interaction in the Relay Case

3.1.3 INAP protocol architecture

Many of the terms used in this subclause are based on the OSI application layer structure as defined in ISO/IEC 9545.

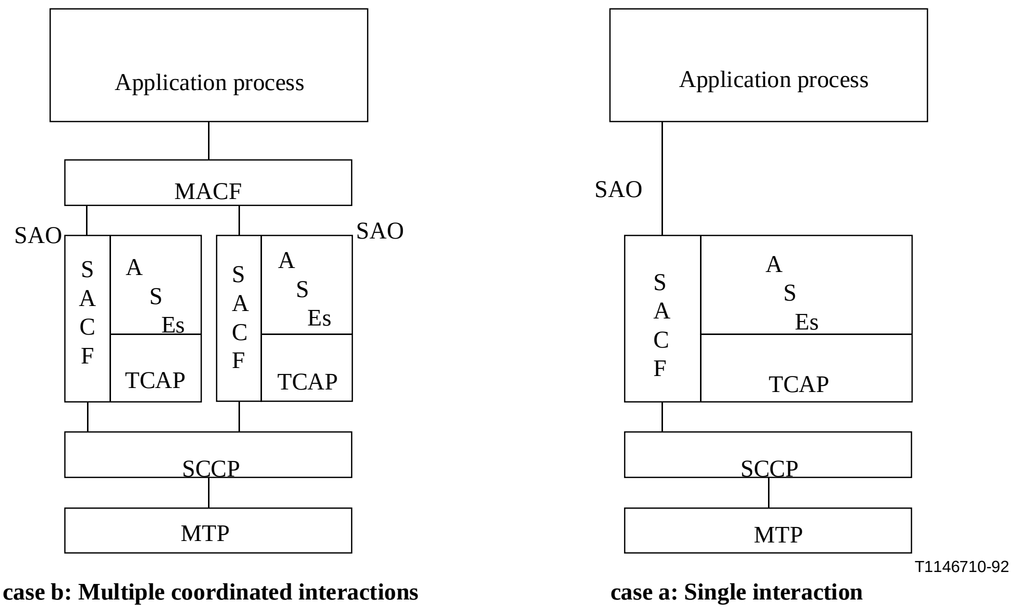
The INAP protocol architecture can be illustrated as shown in Figure 3-9.

A physical entity has either single interactions (case a) or multiple coordinated interactions (case b) with other physical entities.

In (case a), SACF provides a coordination function in using ASEs, which includes the ordering of operations supported by ASE(s), (based on the order of received primitives). The SAO represents the SACF plus a set of ASEs to be used over a single interaction between a pair of PEs.

In (case b), MACF provides a coordinating function among several SAOs, each of which interacts with an SAO in a remote PE.

Each ASE supports one or more operations. Description of each operation is tied with the action of corresponding FE modelling (see Recommendation Q.1214 and clause 3. Each operation is specified using the OPERATION macro described in Figure 3-9.



SACF	Single Association Control Function
MACF	Multiple Association Control Function
SAO	Single Association Object
ASE	Application Service Element
INAP	Intelligent Network Application Protocol

NOTE – INAP is the collection of specifications of all in ASEs.

Figure 3-9/Q.1228 – INAP protocol architecture

The use of the application context negotiation mechanism [as defined in the Q.770-series (*Transaction capabilities application part*)] allows the two communicating entities to identify exactly what their capabilities are and also what the capabilities required on the interface should be. This should be used to allow evolution through Intelligent Network capability sets.

If the indication of a specific application context is not supported by a pair of communicating FEs, some mechanism to pre-arrange the context must be supported.

3.1.3.1 INAP signalling congestion control for Signalling System No.7

The same type of procedure shall apply as defined for ISDN User Part signalling congestion control. The INAP procedures for signalling congestion control shall as far as possible be aligned with the ISDN User Part signalling congestion control procedures as specified in D.2.11/Q.767, i.e. on receipt of N-PCSTATE indication primitive with the information "signalling point congested" from SCCP, the INAP shall reduce the traffic load (e.g. InitialDP, AnalyzedInformation, or InitiateCallAttempts) into the affected direction in several steps.

The above procedure may only apply to traffic which uses MTP Point Code addressing in the affected direction.

3.1.4 INAP addressing

SCCP global title and MTP point code addressing [see Q.710-series (*Signalling connection control part*) and Q.700-series (*Message transfer part*)] ensure that PDUs reach their physical destination (i.e. the correct point code) regardless of which network it is in.

Within a node, it is the choice of the network operator/implementor as to which SSN or SSNs are assigned to INAP.

Regardless of the above, any addressing scheme supported by the SCCP may be used. See Figure 3-10.

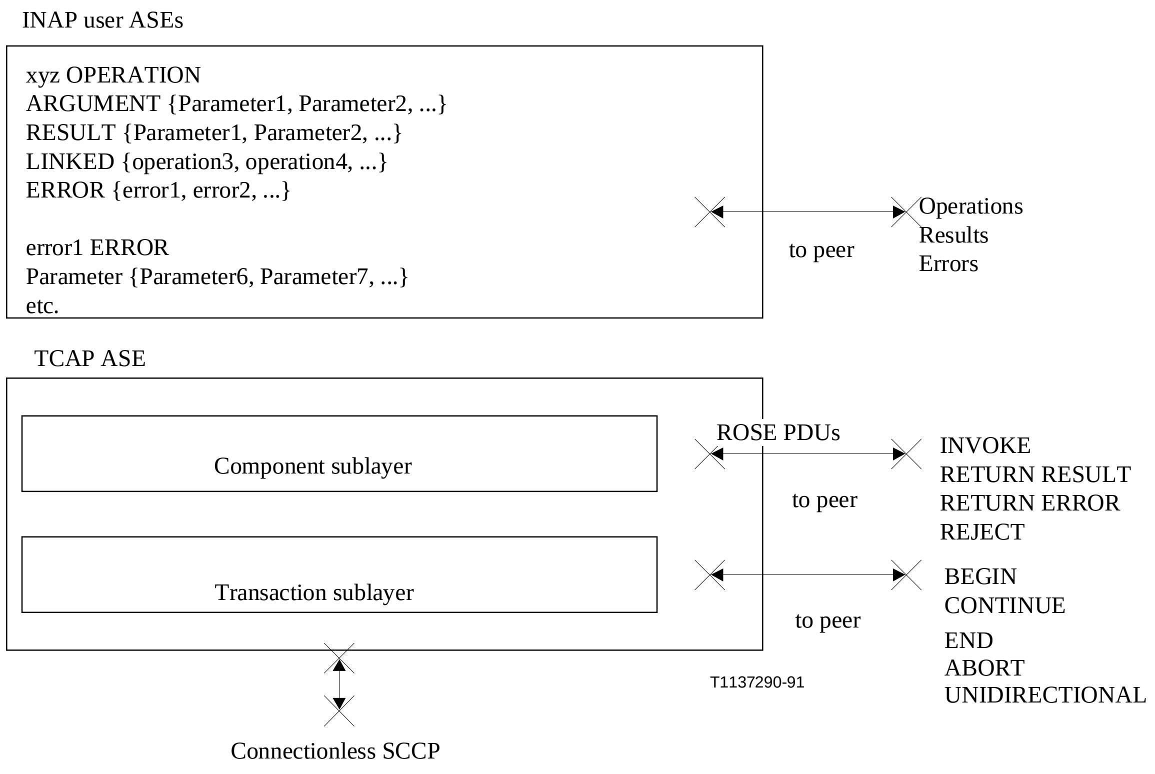


Figure 3-10/Q.1228 – Operation description

3.1.5 Relationship between Recommendation Q.1224 and this Recommendation

The following is a complete list of information flows. These map one-to-one with operations except where indicated.

Refer to 18.1 (Services Assumed from TCAP) to determine mapping of operations onto TCAP dialogue and component portions.

Rec. Q.1224 reference	**Information flow**	**Operation**
SCF-SSF		
12.4.3.1	Activate Service Filtering	Same
12.4.3.2	Activate Trigger Data	ManageTriggerData
12.4.3.3	Activate Trigger Data Confirmation	Return Result from ManageTriggerData
12.4.3.4	Activity Test	Same
12.4.3.5	Activity Test Response	Return Result from ActivityTest
12.4.3.6	Analyse Information	Same
12.4.3.7	Analysed Information	Same
12.4.3.8	Apply Charging	Same
12.4.3.9	Apply Charging Report	Same
12.4.3.10	Assist Request Instructions	Same
12.4.3.11	Authorize Termination	Same
12.4.3.12	Call Gap	Same

Rec. Q.1224 reference	**Information flow**	**Operation**
12.4.3.13	Call Information Report	Same
12.4.3.14	Call Information Request	Same
12.4.3.15	Cancel All Requests	Cancel (All Requests)
12.4.3.16	Cancel Status Report Request	Same
12.4.3.17	Collect Information	Same
12.4.3.18	Collected Information	Same
12.4.3.19	Connect	Same
12.4.3.20	Connect to Resource	Same
12.4.3.21	Continue	Same, ContinuewithArgument
12.4.3.22	Create Call Segment Association	Same
12.4.3.23	Create Call Segment Association Result	Return Result from CreateCallSegmentAssociation
12.4.3.24	Deactivate Trigger Data	ManageTriggerData
12.4.3.25	Deactivate Trigger Data Confirmation	Return Result from ManageTriggerData
12.4.3.26	Disconnect Forward Connection	Same, DFCwithArgument
12.4.3.27	Disconnect Leg	Same
12.4.3.28	Entity Released	Same
12.4.3.29	Establish Temporary Connection	Same
12.4.3.30	Event Notification Charging	Same
12.4.3.31	Event Report BCSM	Same
12.4.3.32	Event Report Facility	Same
12.4.3.33	Facility Selected And Available	Same
12.4.3.34	Furnish Charging Information	Same
12.4.3.35	Hold Call In Network	Same
12.4.3.36	Initial DP	Same
12.4.3.37	Initiate Call Attempt	Same
12.4.3.38	Merge Call Segments	Same
12.4.3.39	Move Call Segments	Same
12.4.3.40	Move Leg	Same
12.4.3.41	O_Abandon	Same
12.4.3.42	O_Answer	Same
12.4.3.43	O_Called_Party_Busy	Same
12.4.3.44	O_Disconnect	Same
12.4.3.45	O_MidCall	Same
12.4.3.46	O_No_Answer	Same
12.4.3.47	O_Suspended	Same
12.4.3.48	Origination Attempt	Same
12.4.3.49	Origination Attempt Authorized	Same
12.4.3.50	Reconnect	Same
12.4.3.51	Release Call	Same
12.4.3.52	Report UTSI	Same
12.4.3.53	Request Notification Charging Event	Same
12.4.3.54	Request Report BCSM Event	Same
12.4.3.55	Request Report Facility Event	Same
12.4.3.56	Request Report UTSI	Same

Rec. Q.1224 reference	Information flow	Operation
12.4.3.57	Request Status Report	RequestCurrentStatusReport RequestFirstStatusMatchReport RequestEveryStatusChangeReport
12.4.3.58	Reset Timer	Same
12.4.3.59	Route Select Failure	Same
12.4.3.60	Select Facility	Same
12.4.3.61	Select Route	Same
12.4.3.62	Send Charging Information	Same
12.4.3.63	Send Facility Information	Same
12.4.3.64	Send STUI	Same
12.4.3.65	Service Filtering Response	Same
12.4.3.66	Split Leg	Same
12.4.3.67	Status Report	Same, Return Result from requestCurrentStatusReport
12.4.3.68	T_Answer	Same
12.4.3.69	T_Busy	Same
12.4.3.70	T_Disconnect	Same
12.4.3.71	T_MidCall	Same
12.4.3.72	T_NoAnswer	Same
12.4.3.73	T_Suspended	Same
12.4.3.74	Termination Attempt	Same
12.4.3.75	Termination Attempt Authorized	termAttemptAuthorized
12.4.3.76	Trigger Data Status Report	Return Result from ManageTriggerData
12.4.3.77	Trigger Data Status Request	ManageTriggerData
SCF-SRF		
12.5.2.1	AssistRequestInstructions from SRF	AssistRequestInstructions
12.5.2.2	Cancel Announcement	Cancel (invokeID)
12.5.2.3	Collected User Information	Return Result from PromptAndCollectUserInformation
12.5.2.4	Message Received	Return Result from PromptAndReceiveMessage
12.5.2.5	Play Announcement	Same
12.5.2.6	Prompt And Collect User Information	Same
12.5.2.7	Prompt And Receive Message	Same
12.5.2.8	Script Close	Same
12.5.2.9	Script Event	Same
12.5.2.10	Script Information	Same
12.5.2.11	Script Run	Same
12.5.2.12	Specialized Resource Report	Same
SCF-SCF		
12.6.2.1	Activity Test	Activity Test
12.6.2.2	Activity Test Result	Return Result from ActivityTest
12.6.2.3	Additional Information Result	Return Result from ProvideUserInformation
12.6.2.4	Confirmed Notification Provided	Same
12.6.2.5	Confirmed Report Charging Information	Same

Rec. Q.1224 reference	Information flow	Operation
12.6.2.6	Establish Charging Record	Same
12.6.2.7	Handling Information Referral	Same
12.6.2.8	Handling Information Request	Same
12.6.2.9	Handling Information Result	Same
12.6.2.10	Network Capability Request	NetworkCapability
12.6.2.11	Network Capability Result	Return Result from NetworkCapability
12.6.2.12	Notification Provided	Same
12.6.2.13	Notification Provided Confirmation	Return Result from ConfirmedNotificationProvided
12.6.2.14	Provide User Information	Same
12.6.2.15	Report Charging Information	Same
12.6.2.16	Report Charging Information Confirmation	Return Result from ConfirmedReportChargingInformationConfirmation
12.6.2.17	Request Notification	Same
12.6.2.18	SCF Bind Request	SCF Bind
12.6.2.19	SCF Bind Result	Return Result from SCF Bind
12.6.2.20	SCF Unbind Request	SCF Unbind
SCF-CUSF		
12.7.2.1	Activation Received and Authorized	Same
12.7.2.2	Activity Test	Same
12.7.2.3	Activity Test Response	Return Result from ActivityTest
12.7.2.4	Association Release Requested	Same
12.7.2.5	Component Received	Same
12.7.2.6	Initiate Association	Same
12.7.2.7	Request Report BCUSM Event	Same
12.7.2.8	Release Association	Same
12.7.2.9	Send Component	Same
SCF-SDF		
12.8.2.1	Add Entry	Same
12.8.2.2	Add Entry Referral	Return Error from AddEntry
12.8.2.3	Add Entry Result	Return Result from AddEntry
12.8.2.4	Authenticate	Bind
12.8.2.5	Authenticate Result	Return Result from Bind
12.8.2.6	End Authenticated Relationship	Unbind
12.8.2.7	Execute	Same
12.8.2.8	Execute Referral	Return Error from Execute
12.8.2.9	Execute Result	Return Result from Execute
12.8.2.10	Modify Entry	Same
12.8.2.11	Modify Entry Referral	Return Error from ModifyEntry
12.8.2.12	Modify Entry Result	Return Result from ModifyEntry
12.8.2.13	Remove Entry	Same
12.8.2.14	Remove Entry Referral	Return Error from RemoveEntry
12.8.2.15	Remove Entry Result	Return Result from RemoveEntry

Rec. Q.1224 reference	Information flow	Operation
12.8.2.16	Search	Same
12.8.2.17	Search Referral	Return Error from Search
12.8.2.18	Search Result	Return Result from Search
SDF-SDF		
12.9.2.1	Authenticate	dSABind, dSAShadowBind
12.9.2.2	Authenticate Result	Return Result from dSABind or dSAShadowBind
12.9.2.3	Chaining Request	chained {OPERATION}
12.9.2.4	Chaining Result	Return Result from OPERATION
12.9.2.5	Copy Request	Coordinate Shadow Update Request Shadow Update
12.9.2.6	Copy Result	Return Result from Coordinate Shadow Update or from Request Shadow Update
12.9.2.7	End Authenticated Relationship	in-DSAUnbind, in-DSAShadowUnbind
12.9.2.8	Update Copy	Update Shadow
12.9.2.9	Update Copy Result	Return Result from Update Shadow

3.1.6 Compatibility mechanisms used for INAP

3.1.6.1 Introduction

This subclause specifies the compatibility mechanisms that shall be used to ensure consistent future versions of INAP.

There are three categories of compatibility:

– Minor changes to INAP in future standardized versions:

  A minor change can be defined as a change of a functionality which is not essential for the requested IN service. In case it is a modification of an existing function, it is acceptable that the addressed function is executed in either the older or the modified variant. If the change is purely additional, it is acceptable that it is not executed at all and that the peer Application Entity (AE) need not know about the effects of the change. For minor changes, a new AC is not required.

– Major changes to INAP in future standardized versions:

  A major change can be defined as a change of a functionality which is essential for the requested IN service. In case it is a modification of an existing function, both application entities shall have a shared knowledge about the addressed functional variant. If the change is purely additional, the requested IN service will not be provided if one of the application entities does not support the additional functionality. For major changes, a new AC is required.

– Network-specific changes to INAP:

  These additions may be of either the major or minor type for a service. No new AC is expected to be defined for this type of change. At the time of definition, the additions would not be expected to be included in identical form in future versions of Recommendations.

3.1.6.2 Definition of INAP compatibility mechanisms

3.1.6.2.1 Procedures for major additions to INAP

In order to support the introduction of major functional changes, the protocol allows a synchronisation between the two applications with regard to which functionality is to be performed. This synchronisation takes place before the new function is invoked in either application entity, in order to avoid complicated fall-back procedures. The solution chosen to achieve such a synchronisation is to use the AC negotiation procedures provided in Recommendation Q.773.

3.1.6.2.2 Procedures for minor additions to INAP

The extension mechanism marker shall be used for future standardized minor additions to INAP. This mechanism implements extensions differently by including an "extensions marker" in the type definition. The extensions are expressed by optional fields that are placed after the marker. When an entity receives unrecognised parameters that occur after the marker, they are ignored (see Recommendation X.68x).

3.1.6.2.3 Procedures for inclusion of network-specific additions to INAP

This mechanism is based on the ability to explicitly declare fields of any type via the Macro facility in ASN.1 at the outermost level of a type definition. It works by defining an "ExtensionField" that is placed at the end of the type definition. This extension field is defined as a set of extensions, where an extension can contain any type. Each extension is associated with a value that defines whether the terminating node should ignore the field if unrecognised, or reject the message, similar to the comprehension required mechanism described in the previous subclause. Refer to Recommendation Q.1400 for a definition of this mechanism.

3.2 SACF/MACF rules

3.2.1 Reflection of TCAP AC

TCAP Application Context negotiation rules require that the proposed AC, if acceptable, is reflected in the first backwards message.

If the AC is not acceptable, and the TC-User does not wish to continue the dialogue, it may provide an alternate AC to the initiator which can be used to start a new dialogue.

TCAP AC negotiation applies only to the SCF interfaces.

Refer to the Q.770-series (*Transaction capabilities application part*) for a more detailed description of the TCAP AC negotiation mechanism.

3.2.2 Sequential/parallel execution of operations

In some cases, it may be necessary to distinguish whether operations should be performed sequentially or in parallel (synchronised). Operations which may be synchronised are:

– charging operations may be synchronised with any other operation.

The method of indicating that operations are to be synchronised is to include them in the same message. Where one of the operations identified above must not be executed until some other operation has progressed to some extent or finished, the sending PE (usually SCP) can control this by sending the operations in two separate messages.

This method does not imply that all operations sent in the same message should be executed simultaneously, but simply that where it could make sense to do so (in the situations identified above) the operations should be synchronised.

In case of inconsistency between the above-mentioned generic rules and the FE-specific rules, as specified in clause 3, the FE-specific rules take precedence over the generic rules.

4 Common IN CS-2 Types

4.1 Data types

```
-- The Definition of Common Data Types Follows

IN-CS2-datatypes {itu-t recommendation q 1228 modules(0) in-cs2-datatypes (0) version1(0)}

DEFINITIONS IMPLICIT TAGS ::=

BEGIN

IMPORTS

tc-Messages, classes FROM  IN-CS2-object-identifiers
                { itu-t recommendation q 1228 module(0) in-cs2-object-identifiers(17) version1(0) }

        InvokeIdType
FROM TCAPMessages tc-Messages

        EXTENSION,
        PARAMETERS-BOUND,
        SupportedExtensions { }
FROM IN-CS2-classes classes
;

AccessCode  {PARAMETERS-BOUND : bound} ::= LocationNumber {bound}

-- An access code from a business group dialling plan attendant access codes, access codes to escape
-- to the public network, access code to access a private facility/network, and feature access codes.
-- Uses the LocationNumber format which is based on the Q.763 Location Number format.
-- The Nature of Address indicator field shall be set to "Spare" (value 00000000).
-- The Numbering Plan Indicator field shall be set to "Spare" (value 000).
-- Of local significance.

AccountNumber ::= NumericString (SIZE (1..151))

AChBillingChargingCharacteristics  {PARAMETERS-BOUND : bound} ::= OCTET STRING (SIZE
(bound.&minAChBillingChargingLength..bound.&maxAChBillingChargingLength))

-- The AChBillingChargingCharacteristics parameter specifies the charging related information
-- to be provided by the SSF and the conditions on which this information has to be reported
-- back to the SCF with the ApplyChargingReport operation.
-- Examples of charging related information to be provided by the SSF may be: bulk counter
-- values, costs, tariff change and time of charge, time stamps, durations, etc.
-- Examples of conditions on which the charging related information are to be reported may be:
-- threshold value reached, timer expiration, tariff change, end of connection configuration, etc.

ActionIndicator  ::= ENUMERATED {
activate        (1) ,
deactivate      (2) ,
retrieve        (3)
        }
-- indicates the action to be performed by the ManageTriggerData operation (activate, deactivate
-- or retrieve the status of a TDP.
```

```
ActionPerformed  ::= ENUMERATED {
activated                              (1) ,
deactivated                            (2) ,
alreadyActive                          (3) ,
alreadyInactive                        (4) ,
isActive                               (5) ,
isInactive                             (6)
        }
-- indicates the result of the operation ManageTriggerData
-- activated: response of activate TDP
-- deactivated: response of deactivate TDP
-- alreadyActive: response of activate TDP
-- alreadyInactive: response of deactivate TDP
-- isActive: response of retrieve status of TDP
-- isInactive: response of retrieve status of TDP

ActivableServices ::= BIT STRING {
        callingLineIdentificationPresentation (1),
        callingLineIdentificationRestriction (2),
        connectedLineIdentificationPresentation (3),
        connectedLineIdentificationRestriction (4),
        callForwardingOnNoReply (5),
        callForwardingUnconditional (6),
        callForwardingOnBusy (7),
        callForwardingOnNotReachable (8),
        reverseCharging (9),
        adviceOfChargeOnStart (10),
        adviceOfChargeAtEnd (11),
        adviceOfChargeDuringCall (12),
        timeDependentRouting (13),
        callingPartingDependentRouting (14),
        outgoingCallBarring (15),
        incomingCallBarring (16)
        }

AdditionalCallingPartyNumber  {PARAMETERS-BOUND : bound} ::= Digits {bound}

-- Indicates the Additional Calling Party Number. Refer to Rec. Q.763 for encoding.

AlertingPattern ::= OCTET STRING (SIZE(3))

-- Indicates a specific pattern that is used to alert a subscriber (e.g. distinctive ringing, tones, etc.).
-- Only applies if SSF is the terminating local exchange for the subscriber. Refer to the Q.931
-- Signal parameter for encoding.

ApplicationTimer ::=INTEGER (0..2047)

-- Used by the SCF to set a timer in the SSF. The timer is in seconds.

AssistingSSPIPRoutingAddress  {PARAMETERS-BOUND : bound} ::= Digits {bound}

-- Indicates the destination address of the SRF for the assist procedure.

BackwardGVNS  {PARAMETERS-BOUND : bound} ::= OCTET STRING (SIZE(
                                bound.&minBackwardGVNSLength..bound.&maxBackwardGVNSLength))

-- Indicates the GVNS Backward information. Refer to clause 6/Q.735 for encoding.
```

```
BackwardServiceInteractionInd ::= SEQUENCE {
        conferenceTreatmentIndicator          [1] OCTET STRING (SIZE(1))          OPTIONAL,
                -- acceptConferenceRequest 'xxxx xx01'B
                -- rejectConferenceRequest         'xxxx xx10'B
                -- network default is accept conference request,
        callCompletionTreatmentIndicator      [2] OCTET STRING (SIZE(1))          OPTIONAL
                -- acceptCallCompletionServiceRequest          'xxxx xx01'B,
                -- rejectCallCompletionServiceRequest          'xxxx xx10'B
                -- network default is accept call completion service request
        }

BCSMEvent  {PARAMETERS-BOUND : bound} ::= SEQUENCE {
        eventTypeBCSM                           [0] EventTypeBCSM,
        monitorMode                             [1] MonitorMode,
        legID                                   [2] LegID                          OPTIONAL,

        dpSpecificCriteria                      [30] DpSpecificCriteria {bound}    OPTIONAL
        }
-- Indicates the BCSM Event information for monitoring.

BCUSMEvent ::= SEQUENCE{
        eventType                               [0] EventTypeBCUSM,
        monitorMode                             [1] MonitorMode
        }

BearerCapabilities ::= BIT STRING {
        speech (0),
        bc64kbits (1),
        bc2x64kbits (2),
        bc384kbits (3),
        bc1536kbits (4),
        bc1920kbits (5),
        multirate (6),
        restrictedDigitalInfo (7),
        bc3-1khzAudio (8),
        bc7khzAudio (9),
        video (10)
        }
BearerCapability  {PARAMETERS-BOUND : bound} ::= CHOICE {
        bearerCap                   [0] OCTET STRING (SIZE(2..bound.&maxBearerCapabilityLength)),
        tmr                         [1] OCTET STRING (SIZE(1))
        }

-- Indicates the type of bearer capability connection to the user. For bearerCapability, either
-- DSS 1 (Rec. Q.931) or the ISUP User Service Information (Rec. Q.763) encoding can be used. Refer
-- to the Q.763 Transmission Medium Requirement parameter for tmr encoding.

BothwayThroughConnectionInd ::= ENUMERATED {
        bothwayPathRequired         (0),
        bothwayPathNotRequired      (1)
        }

CallConditions  {PARAMETERS-BOUND : bound} ::= CHOICE {
        userAbandon                             [0] NULL,
        callFailure                             [1] CauseValue,
        noReply                                 [2] INTEGER, -- time expressed in seconds
        callRelease                             [3] NULL,
        ss-invocation                           [4] InvokableService,
        creditLimitReached                      [5] INTEGER,
        callDuration                                [6] INTEGER,
        calledNumber                                [7] NumberMatch {bound},
        answeredCall                                [8] NULL
        }

CalledPartyBusinessGroupID ::= OCTET STRING

-- Indicates the business group of the called party. The value of this octet string is network-
-- operator specific.

CalledPartyNumber  {PARAMETERS-BOUND : bound} ::= OCTET STRING (SIZE
                                            (bound.&minCalledPartyNumberLength..
bound.&maxCalledPartyNumberLength))

-- Indicates the Called Party Number. Refer to Rec. Q.763 for encoding.

CalledPartySubaddress ::= OCTET STRING

-- Indicates the Called Party Subaddress. Refer to Rec. Q.931 for encoding.

CallIdentifier  ::= INTEGER (1..2147483647)

CallingPartyBusinessGroupID ::= OCTET STRING

-- Indicates the business group of the calling party. The value of this octet string is network-
-- operator specific.

CallingPartyNumber  {PARAMETERS-BOUND : bound} ::= OCTET STRING (SIZE (
                                            bound.&minCallingPartyNumberLength..
                                            bound.&maxCallingPartyNumberLength))

-- Indicates the Calling Party Number. Refer to Rec. Q.763 for encoding.

CallingPartySubaddress ::= OCTET STRING

-- Indicates the Calling Party Subaddress. Refer to Rec. Q.931 for encoding.

CallingPartysCategory ::= OCTET STRING (SIZE(1))

-- Indicates the type of calling party (e.g. operator, payphone, ordinary subscriber). Refer to Rec. Q.763
-- for encoding.

CallProcessingOperationCorrelationID ::=ENUMERATED {
        aLERTing(1),
        sETUP(5),
        cONNect(7),
        dISConnect(69),
        rELease(77),
        rELeaseCOMPlete(90),
        fACility(98)
        }

CallRecord  {PARAMETERS-BOUND : bound} ::= SEQUENCE {
        callDuration                                [0] Duration,
        callingPartyNumber                          [1] CallingPartyNumber {bound},
        calledPartyNumber                           [2] CalledPartyNumber {bound}
        }
```

```
CallResult  {PARAMETERS-BOUND : bound} ::= OCTET STRING (SIZE (bound.&minCallResultLength..
                                    bound.&maxCallResultLength))

-- This parameter provides the SCF with the charging related information previously requested
-- using the ApplyCharging operation. This shall include the partyToCharge parameter as
-- received in the related ApplyCharging operation to correlate the result to the request
-- The remaining content is network-operator specific.
-- Examples of charging related information to be provided by the SSF may be: bulk counter values,
-- costs, tariff change and time of change, time stamps, durations, etc.
-- Examples of conditions on which the charging related information are to be reported may be:
-- threshold value reached, timer expiration, tariff change, end of connection configuration, etc.

CallSegmentID  {PARAMETERS-BOUND : bound} ::= INTEGER (1..bound.&numOfCSs)

initialCallSegment INTEGER ::= 1
-- the initial call segment represents the call segment that was there when the CSA was created, ie. the CS where
-- the trigger took place or the CS that was created by an InitateCallAttempt within a TC-BEGIN message.

CallUnrelatedDpSpecificCommonParameters  {PARAMETERS-BOUND : bound} ::= SEQUENCE {
      serviceAddressInformation          [0] ServiceAddressInformation,
      callingPartyNumber                 [1] CallingPartyNumber {bound}      OPTIONAL,
      locationNumber                     [2] LocationNumber {bound}          OPTIONAL,
      terminalType                       [3] TerminalType                    DEFAULT isdn,
      extensions                         [4] SEQUENCE SIZE(1..bound.&numOfExtensions) OF
                                             ExtensionField {bound}          OPTIONAL
--    ...
      }

Carrier ::= OCTET STRING

-- Contains the carrier selection and carrier ID fields.
-- Carrier selection is one octet and is encoded as:
-- 00000000        No indication
-- 00000001        Selected carrier code pre-subscribed and not input by calling party
-- 00000010        Selected carrier identification code pre-subscribed and input by calling party
-- 00000011        Selected carrier identification code pre-subscribed, no indication of whether input by calling party
-- 00000100        Selected carrier identification code not pre-subscribed and input by calling party
-- 00000101
--    to           Spare
-- 11111110
-- 11111111        Reserved
--
-- Carrier ID has a one-octet field indicating the number of digits followed by the digits encoded using BCD
-- Detailed coding is for further study. It is of local significance and carrying it through the ISUP is for further study

Cause  {PARAMETERS-BOUND : bound} ::= OCTET STRING (SIZE (minCauseLength..
                                    bound.&maxCauseLength))
-- Indicates the cause for interface related information. Refer to the Q.763 Cause parameter for encoding.
-- For the use of cause and location values, refer to Rec. Q.850

CauseValue ::= OCTET STRING (SIZE (1)) --type extracted from Cause parameter in Rec. Q.763.

CGEncountered ::= ENUMERATED {
      noCGencountered(0),
      manualCGencountered(1),
      scpOverload(2)
      }

-- Indicates the type of automatic call gapping encountered, if any.
```

```
ChargeNumber  {PARAMETERS-BOUND : bound} ::= LocationNumber {bound}

-- Information sent in either direction indicating the chargeable number for the call and consisting
-- of the odd/even indicator, nature of address indicator, numbering plan indicator, and address signals.
-- Uses the LocationNumber format which is based on the Q.763 Location Number format
-- For example, the ChargeNumber may be a third party number to which a call is billed for the 3rd party billing
-- service. In this case, the calling party may request operator assistance to charge the call to,
-- for example, their home number.

ChargingEvent  {PARAMETERS-BOUND : bound} ::= SEQUENCE {
        eventTypeCharging          [0] EventTypeCharging {bound},
        monitorMode                [1] MonitorMode,
        legID                      [2] LegID                                  OPTIONAL
        }

-- This parameter indicates the charging event  type and corresponding
-- monitor mode and LedID

ChargingParameters   {PARAMETERS-BOUND : bound} ::= SEQUENCE {
        unitsPerInterval           [0] INTEGER (0..bound.&maxUnitsPerInterval),
        timePerInterval            [1] INTEGER (0..bound.&maxTimePerInterval),
        scalingFactor              [2] INTEGER (0..bound.&maxScalingFactor),
        initialUnitIncrement       [3] INTEGER (0..bound.&maxInitialUnitIncrement)         OPTIONAL,
        unitsPerDataInterval       [4] INTEGER (0..bound.&maxUnitsPerDataInterval)         OPTIONAL,
        segmentsPerDataInterval    [5] INTEGER (0..bound.&maxSegmentsPerDataInterval)      OPTIONAL,
        initialTimeInterval        [6] INTEGER (0..bound.&maxInitialTimeInterval)          OPTIONAL
        }

CollectedDigits ::= SEQUENCE {
        minimumNbOfDigits          [0] INTEGER (1..127)                DEFAULT 1,
        maximumNbOfDigits          [1] INTEGER (1..127),
        endOfReplyDigit            [2] OCTET STRING (SIZE (1..2))  OPTIONAL,
        cancelDigit                [3] OCTET STRING (SIZE (1..2))  OPTIONAL,
        startDigit                 [4] OCTET STRING (SIZE (1..2))  OPTIONAL,
        firstDigitTimeOut          [5] INTEGER (1..127)                OPTIONAL,
        interDigitTimeOut          [6] INTEGER (1..127)                OPTIONAL,
        errorTreatment             [7] ErrorTreatment              DEFAULT reportErrorToScf,
        interruptableAnnInd        [8] BOOLEAN                         DEFAULT TRUE,
        voiceInformation           [9] BOOLEAN                         DEFAULT FALSE,
        voiceBack                  [10] BOOLEAN                        DEFAULT FALSE
        }

-- The use of voiceBack is network-operator specific.
-- The endOfReplyDigit, cancelDigit, and startDigit parameters have been designated as OCTET STRING,
-- and are to be encoded as  BCD, one digit per octet only, contained
-- in the four least significant bits of each OCTET. The usage is service dependent.

CollectedInfo ::= CHOICE {
        collectedDigits            [0] CollectedDigits,
        iA5Information             [1] BOOLEAN
        }

Component ::= CHOICE {
        componentInfo              [0] OCTET STRING (SIZE(1..118)),

-- Contains the operation value (object identifier), error value, etc. within the UNI APDU, in addition also contains
-- the parameter set/sequence for the operation invocation/return result of return error/reject on UNI. See Rec. Q.932
-- for encoding

        relayedComponent           [1] EMBEDDED PDV
        }
```

```
-- If componentInfo is chosen, then it is necessary to use this parameter in sequence with ComponentType and
-- ComponentCorrelationID
-- If relayedComponent is chosen, then ComponentType and ComponentCorrelationID may not be used in the
-- sequence

ComponentCorrelationID ::= INTEGER

ComponentType ::= ENUMERATED {
        any (0),
        invoke (1),
        rResult (2),
        rError (3),
        rReject (4)
        }


ConnectedNumberTreatmentInd ::= ENUMERATED {
        noINImpact                  (0),
        presentationRestricted      (1),
        presentCalledINNumber       (2)
        }


Constraints ::= SEQUENCE {
        maximumNumberOfDigits       [1] INTEGER (1..127),
        minimumNumberOfDigits       [2] INTEGER (1..127),
        typeOfRequestedInfo         [3] InfoType DEFAULT numericString,
        numberOfAllowedRetries      [4] INTEGER (0..127) DEFAULT 0
        }


ControlConditionByCallParty ::= SEQUENCE {
        endOfMessageSendingDigit    [0] OCTET STRING (SIZE(1..2))              OPTIONAL,
        replayDigit                 [1] OCTET STRING (SIZE(1..2))              OPTIONAL
        }

ControlType ::= ENUMERATED {
        sCPOverloaded(0),
        manuallyInitiated(1),
        destinationOverload(2)
        -- other values for further study (FFS)
        }

CorrelationID {PARAMETERS-BOUND : bound} ::= Digits {bound}

-- used by SCF for correlation with a previous operation. Refer to clause 17 for a description of the procedures
-- associated with this parameter.

CounterAndValue ::= SEQUENCE {
                counterID           [0] CounterID,
                counterValue        [1] Integer4
                }

CounterID ::= INTEGER (0..99)

-- Indicates the counters to be incremented.
-- The counterIDs can be addressed by using the last digits of the dialled number.

CountersValue ::= SEQUENCE SIZE(0..numOfCounters) OF CounterAndValue
```

```
Credit {PARAMETERS-BOUND : bound} ::= CHOICE {
                currency                    CurrencyValue {bound},
                units                       CreditUnit
                                            }

CreditUnit ::= INTEGER (0..maxCreditUnit)

CriticalityType ::= ENUMERATED {
                ignore(0),
                abort(1)
                }

CSAID {PARAMETERS-BOUND : bound} ::= INTEGER (1..bound.&numOfCSAs)
-- Indicates the SSF CSA identifier

CurrencyID ::= PrintableString (SIZE (3) ) -- ISO 639 code

CurrencyValue {PARAMETERS-BOUND : bound} ::= SEQUENCE {
        currency                                    CurrencyID,
        amount                                      INTEGER (0..bound.&maxAmount)
        }

CutAndPaste ::= INTEGER (0..22)

-- Indicates the number of digits to be deleted. Refer to Rec. Q.1224 for additional information.

DateAndTime ::= OCTET STRING (SIZE(6))

-- Indicates, amongst others, the start time for activate service filtering. Coded as YYMMDDHHMMSS
-- with each digit coded BCD.
-- The first octet contains YY and the remaining items are sequenced following.
-- For example, 1993 September 30th, 12:15:01 would be encoded as:
-- Bits                              HGFE        DCBA
-- leading octet                       3           9
--                                     9           0
--                                     0           3
--                                     2           1
--                                     5           1
--                                     1           0

DestinationRoutingAddress {PARAMETERS-BOUND : bound} ::= SEQUENCE SIZE(1..3) OF
CalledPartyNumber {bound}

-- Indicates the list of Called Party Numbers (primary and alternates).

Digits {PARAMETERS-BOUND : bound} ::= OCTET STRING (SIZE
(bound.&minDigitsLength..bound.&maxDigitsLength))

-- Indicates the address signalling digits. Refer to the Q.763 Generic Number and  Generic Digits parameters
-- for encoding. The coding of the subfields 'NumberQualifier' in Generic Number and 'TypeOfDigits' in
-- Generic Digits is irrelevant to the INAP, the ASN.1 tags are sufficient to identify the parameter.
-- The ISUP format does not allow to exclude these subfields, therefore the value is network-operator specific.
-- The following parameters should use Generic Number:
-- CorrelationID for AssistRequestInstructions, AssistingSSPIPRoutingAddress for EstablishTemporaryConnection,
-- calledAddressValue for all occurrences,callingAddressValue for all occurrences.
-- The following parameters should use Generic Digits: prefix, all
-- other CorrelationID occurrences, dialledNumber filtering criteria, callingLineID filtering criteria, lineID for
-- ResourceIDType, digitResponse for ReceivedInformationArg, iNServiceControlLow / iNServiceControlHigh for
-- MidCallInfoType, iNServiceControlCode for MidCallInfo.
```

```
DisplayInformation {PARAMETERS-BOUND : bound} ::= IA5String (SIZE
                                        (bound.&minDisplayInformationLength..
                                        bound.&maxDisplayInformationLength))
-- Indicates the display information.
-- Delivery of DisplayInformation parameter to Private Networks cannot be guaranteed due to signalling
-- interworking problems, solutions are currently under study

DpSpecificCommonParameters {PARAMETERS-BOUND : bound} ::= SEQUENCE {
      serviceAddressInformation        [0] ServiceAddressInformation,
      bearerCapability                 [1] BearerCapability {bound}                  OPTIONAL,
      calledPartyNumber                [2] CalledPartyNumber {bound}                 OPTIONAL,
      callingPartyNumber               [3] CallingPartyNumber {bound}                OPTIONAL,
      callingPartysCategory            [4] CallingPartysCategory                     OPTIONAL,
      iPSSPCapabilities                [5] IPSSPCapabilities {bound}                 OPTIONAL,
      iPAvailable                      [6] IPAvailable {bound}                       OPTIONAL,
      iSDNAccessRelatedInformation     [7] ISDNAccessRelatedInformation              OPTIONAL,
      cGEncountered                    [8] CGEncountered                             OPTIONAL,
      locationNumber                   [9] LocationNumber {bound}                    OPTIONAL,
      serviceProfileIdentifier         [10] ServiceProfileIdentifier                 OPTIONAL,
      terminalType                     [11] TerminalType                             OPTIONAL,
      extensions                       [12] SEQUENCE SIZE(1..bound.&numOfExtensions)        OF
                                            ExtensionField {bound}                   OPTIONAL,
      chargeNumber                     [13] ChargeNumber {bound}                     OPTIONAL,
      servingAreaID                    [14] ServingAreaID {bound}                    OPTIONAL,
      serviceInteractionIndicators     [15] ServiceInteractionIndicators {bound}     OPTIONAL,
      iNServiceCompatibilityIndication [16] INServiceCompatibilityIndication {bound} OPTIONAL,
      serviceInteractionIndicatorsTwo  [17] ServiceInteractionIndicatorsTwo          OPTIONAL,
      uSIServiceIndicator              [18] USIServiceIndicator {bound}              OPTIONAL,
      uSIInformation                   [19] USIInformation {bound}                   OPTIONAL,
      forwardGVNS                      [20] ForwardGVNS {bound}                      OPTIONAL,
      createdCallSegmentAssociation    [21] CSAID {bound}                            OPTIONAL,
      ...
      }

-- OPTIONAL for iPSSPCapabilities, iPAvailable, and cGEncountered denotes network-operator specific use.
-- OPTIONAL for callingPartyNumber, and callingPartysCategory refer to clause 17 for
-- the trigger detection point processing rules to specify when these parameters are included in the
-- message. bearerCapability should be appropriately coded as speech.

DpSpecificCriteria {PARAMETERS-BOUND : bound} ::= CHOICE {
            numberOfDigits             [0] NumberOfDigits,
            applicationTimer           [1] ApplicationTimer,
            midCallControlInfo         [2] MidCallControlInfo {bound}
            }

-- The SCF may specify the number of digits to be collected by the SSF for the CollectedInfo event.
-- When all digits are collected, the SSF reports the event to the SCF.
-- The SCF may set a timer in the SSF for the No Answer event. If the user does not answer the call
-- within the allotted time, the SSF reports the event to the SCF

Duration ::= INTEGER (-2..86400)

-- Values are seconds

ElementaryMessageID ::= Integer4

Entry ::=CHOICE {
            agreements                 [0] OBJECT IDENTIFIER,
            networkSpecific            [1] Integer4
            }
```

```
ErrorTreatment ::= ENUMERATED {
            reportErrorToScf(0),
            help(1),
            repeatPrompt(2)
            }

-- reportErrorToScf means returning the "ImproperCallerResponse" error in the event of an error
-- condition during collection of user info.

EventSpecificInformationBCSM {PARAMETERS-BOUND : bound} ::= CHOICE {
      collectedInfoSpecificInfo        [0] SEQUENCE {
                                       calledPartynumber        [0] CalledPartyNumber {bound},
                                       ...
                                       },
      analysedInfoSpecificInfo         [1] SEQUENCE {
                                       calledPartynumber        [0] CalledPartyNumber {bound},
                                       ...
                                       },
      routeSelectFailureSpecificInfo   [2] SEQUENCE {
                                       failureCause             [0] Cause {bound}         OPTIONAL,
                                       ...
                                       },
      oCalledPartyBusySpecificInfo     [3] SEQUENCE {
                                       busyCause                [0] Cause {bound}         OPTIONAL,
                                       ...
                                       },
      oNoAnswerSpecificInfo            [4] SEQUENCE {
                                       -- no specific info defined --
                                       ...
                                       },
      oAnswerSpecificInfo              [5] SEQUENCE {

                                         backwardGVNS           [0] BackwardGVNS {bound}
                                                                                          OPTIONAL,
                                       ...
                                       },
      oMidCallSpecificInfo             [6] SEQUENCE {
                                       connectTime              [0] Integer4              OPTIONAL,
                                       oMidCallInfo             [1] MidCallInfo {bound}   OPTIONAL,
                                       ...
                                       },

      oDisconnectSpecificInfo          [7] SEQUENCE {
                                       releaseCause             [0] Cause {bound}
                                                                                          OPTIONAL,
                                       connectTime              [1] Integer4              OPTIONAL,
                                       ...
                                       },
      tBusySpecificInfo                [8] SEQUENCE {
                                       busyCause                [0] Cause {bound}
                                                                                          OPTIONAL,
                                       ...
                                       },
      tNoAnswerSpecificInfo            [9] SEQUENCE {
                                       -- no specific info defined --
                                       ...
                                       },
      tAnswerSpecificInfo              [10] SEQUENCE {
                                       -- no specific info defined --
                                       ...
                                       },
        tMidCallSpecificInfo            [11] SEQUENCE {
                                        connectTime             [0] Integer4            OPTIONAL,
                                        tMidCallInfo            [1] MidCallInfo {bound} OPTIONAL,
                                        ...
                                        },
        tDisconnectSpecificInfo         [12] SEQUENCE {
                                        releaseCause            [0] Cause {bound}       OPTIONAL,
                                        connectTime             [1] Integer4            OPTIONAL,
                                        ...
                                        },
        oTermSeizedSpecificInfo         [13] SEQUENCE {
                                        -- no specific info defined --
                                        ...
                                        },
        oSuspended                      [14] SEQUENCE {
                                        -- no specific info defined --
                                        ...
                                        },
        tSuspended                      [15] SEQUENCE {
                                        -- no specific info defined --
                                        ...
                                        },
        origAttemptAuthorized           [16] SEQUENCE {
                                        -- no specific info defined --
                                        ...
                                        },
        oReAnswer                       [17] SEQUENCE {
                                        -- no specific info defined --
                                        ...
                                        },
        tReAnswer                       [18] SEQUENCE {
                                        -- no specific info defined --
                                        ...
                                        },
        facilitySelectedAndAvailable    [19] SEQUENCE {
                                        -- no specific info defined --
                                        ...
                                        },
        callAccepted                    [20] SEQUENCE {
                                        -- no specific info defined --
                                        ...
                                        },
        oAbandon                        [21] SEQUENCE {
                                        abandonCause            [0] Cause {bound}       OPTIONAL,
                                        ...
                                        },
        tAbandon                        [22] SEQUENCE {
                                        abandonCause            [0] Cause {bound}       OPTIONAL,
                                        ...
                                        }
        }
-- Indicates the call related information specific to the event.
-- The connectTime indicates the duration between the received answer indication from the called party side
-- and the release of the connection for ODisconnect, OException, TDisconnect, or TException or between
-- the received answer indication from the called party side and the time of detection of the required
-- mid call event.
-- The unit for the connectTime is 100 milliseconds

EventSpecificInformationCharging {PARAMETERS-BOUND : bound} ::= OCTET STRING (SIZE
                                        (bound.&minEventSpecificInformationChargingLength..
                                         bound.&maxEventSpecificInformationChargingLength))
```

```
-- defined by network operator.
-- Indicates the charging related information specific to the event.
-- An example data type definition for this parameter is given below:
--              chargePulses                [0] Integer4,
--              chargeMessages              [1] OCTET STRING (SIZE (min..max))
EventTypeBCSM ::= ENUMERATED {
        origAttemptAuthorized(1),
        collectedInfo(2),
        analysedInformation(3),
        routeSelectFailure(4),
        oCalledPartyBusy(5),
        oNoAnswer(6),
        oAnswer(7),
        oMidCall(8),
        oDisconnect(9),
        oAbandon(10),
        termAttemptAuthorized(12),
        tBusy(13),
        tNoAnswer(14),
        tAnswer(15),
        tMidCall(16),
        tDisconnect(17),
        tAbandon(18),
        oTermSeized(19),
        oSuspended(20),
        tSuspended(21),
        origAttempt(22),
        termAttempt(23),
        oReAnswer(24),
        tReAnswer(25),
        facilitySelectedAndAvailable(26),
        callAccepted(27)
        }
-- Indicates the BCSM detection point event. Refer to Rec. Q.1224 for additional information on the events.
-- Values origAttemptAuthorized and termAttemptAuthorized can only be used for TDPs

EventTypeBCUSM ::= ENUMERATED{
                componentReceived(127),
                associationReleaseRequested(126)
                }

EventTypeCharging {PARAMETERS-BOUND : bound} ::= OCTET STRING (SIZE
                                    (bound.&minEventTypeChargingLength..
                                    bound.&maxEventTypeChargingLength))
-- This parameter indicates the charging event type. Its content is network-operator specific.
--
-- An example data type definition for this parameter is given below:
-- EventTypeCharging ::= ENUMERATED {
--                            chargePulses (0),
--                            chargeMessages (1)
--                            }
ExtensionField {PARAMETERS-BOUND : bound} ::= SEQUENCE {
        type                        EXTENSION.&id ({SupportedExtensions {bound}}),
                                    -- shall identify the value of an EXTENSION type
        criticality                 CriticalityType                 DEFAULT ignore,
        value                       [1] EXTENSION.&ExtensionType
                                        ({SupportedExtensions {bound}}{@type})
        }
--This parameter indicates an extension of an argument data type. Its content is network-operator specific
```

```
FacilityGroup ::= CHOICE {
        trunkGroupID                    [0] INTEGER,
        privateFacilityID               [1] INTEGER,
        huntGroup                       [2] OCTET STRING,
        routeIndex                      [3] OCTET STRING
        }
-- Indicates the particular group of facilities to route the call. huntGroup and routeIndex are encoded as
-- network-operator specific.

FacilityGroupMember ::= INTEGER

-- Indicates the specific member of a trunk group or multi-line hunt group.

FailureCause        ::= OCTET STRING
-- FailureCause is FFS. The coding should be specified to be able to handle unsuccessful situation
-- for TDP activation/deactivation.

FCIBillingChargingCharacteristics {PARAMETERS-BOUND : bound} ::= OCTET STRING (SIZE
                                    (bound.&minFCIBillingChargingLength..
                                     bound.&maxFCIBillingChargingLength))
-- This parameter indicates the billing and/or charging characteristics. Its content is network-operator specific.
-- An example datatype definition for this parameter is given below:
-- FCIBillingChargingCharacteristics ::= CHOICE {
--      completeChargingrecord          [0] OCTET STRING (SIZE (min..max)),
--      correlationID                   [1] CorrelationID,
--      scenario2Dot3                        [2] SEQUENCE {
--                                      chargeParty     [0] LegID               OPTIONAL,
--                                      chargeLevel     [1] OCTET STRING (SIZE (min..max))
--                                                                              OPTIONAL,
--                                      chargeItems     [2] SET OF Attribute    OPTIONAL
--                                      }
--      }
-- Depending on the applied charging scenario the following information elements can be included
-- (refer to Appendix II/Q.1214):
-- complete charging record (scenario 2.2)
-- charge party (scenario 2.3)
-- charge level (scenario 2.3)
-- charge items (scenario 2.3)
-- correlationID (scenario 2.4)

FeatureCode {PARAMETERS-BOUND : bound} ::= LocationNumber {bound}

-- The two-digit feature code preceded by "*" or "11".
-- Uses the LocationNumber format which is based on the Q.763 Location Number format.
-- The Nature of Address indicator field shall be set to "Spare" (value 00000000).
-- The Numbering Plan Indicator field shall be set to "Spare" (value 000)
-- Used for stimulus signalling (Rec. Q.932).

FeatureRequestIndicator ::= ENUMERATED {
        hold(0),
        retrieve(1),
        featureActivation(2),
        spare1(3),
        sparen(127)
        }

-- Indicates the feature activated (e.g. a switch-hook flash, feature activation). Spare values reserved
-- for future use.
```

```
FilteredCallTreatment {PARAMETERS-BOUND : bound} ::= SEQUENCE {
        sFBillingChargingCharacteristics   [0] SFBillingChargingCharacteristics {bound},
        informationToSend                  [1] InformationToSend {bound}           OPTIONAL,
        maximumNumberOfCounters            [2] MaximumNumberOfCounters             OPTIONAL,
        releaseCause                       [3] Cause {bound}                       OPTIONAL
        }

-- If releaseCause is not present, the default value is the same as the ISUP cause value decimal 31.
-- If informationToSend is present, the call will be released after the end of the announcement
-- with the indicated or default releaseCause.
-- If maximumNumberOfCounters is not present, ServiceFilteringResponse will be sent with
-- CountersValue::= SEQUENCE SIZE (0) OF CountersAndValue

FilteringCharacteristics ::= CHOICE {
        interval                           [0] INTEGER (1..32000),
        numberOfCalls                      [1] Integer4
        }

-- Indicates the severity of the filtering and the point in time when the ServiceFilteringResponse is to be sent.
-- If = interval, every interval of time the next call leads to an InitialDP and a ServiceFilteringResponse is sent to
-- the SCF. The interval is specified in seconds.
-- If = NumberOfCalls, every N calls the Nth call leads to an InitialDP and a ServiceFilteringResponse
-- is sent to the SCF.
-- If ActivateServiceFiltering implies several counters – filtering on several dialled number –
-- the numberOfCalls would include calls to all the dialled numbers.

FilteringCriteria {PARAMETERS-BOUND : bound} ::= CHOICE {
        dialledNumber                      [0] Digits {bound},
        callingLineID                      [1] Digits {bound},
        serviceKey                         [2] ServiceKey,
        addressAndService                  [30] SEQUENCE {
        calledAddressValue                 [0] Digits {bound},
        serviceKey                         [1] ServiceKey,
        callingAddressValue                [2] Digits {bound}                      OPTIONAL,
        locationNumber                     [3] LocationNumber {bound}              OPTIONAL
                                        }
        }

-- In case calledAddressValue is specified, the numbers to be filtered are from calledAddressValue
-- up to and including calledAddressValue + maximumNumberOfCounters –1.
-- The last two digits of calledAddressvalue cannot exceed 100 –maximumNumberOfCounters.

FilteringTimeOut ::= CHOICE {
        duration                           [0] Duration,
        stopTime                           [1] DateAndTime
        }

-- Indicates the maximum duration of the filtering. When the timer expires, a ServiceFilteringResponse
-- is sent to the SCF.

ForwardCallIndicators ::= OCTET STRING (SIZE(2))

-- Indicates the Forward Call Indicators. Refer to Rec. Q.763 for encoding

ForwardGVNS {PARAMETERS-BOUND : bound} ::= OCTET STRING (SIZE(
                                   bound.&minForwardGVNSLength..
                                   bound.&maxForwardGVNSLength))

-- Indicates the GVNS Forward information. Refer to clause 6/Q.735, for encoding.
```

```
ForwardingCondition ::= ENUMERATED {
        busy(0),
        noanswer(1),
        any(2)
        }

-- Indicates the condition that must be met to complete the connect.

ForwardServiceInteractionInd ::= SEQUENCE {
        conferenceTreatmentIndicator            [1] OCTET STRING (SIZE(1))          OPTIONAL,
                -- acceptConferenceRequest 'xxxx xx01',B
                -- rejectConferenceRequest          'xxxx xx10'B
                -- network default is accept conference request
        callDiversionTreatmentIndicator         [2] OCTET STRING (SIZE(1))          OPTIONAL,
                -- callDiversionAllowed             'xxxx xx01'B
                -- callDiversionNotAllowed          'xxxx xx10'B
                -- network default is Call Diversion allowed
        callOfferingTreatmentIndicator          [3] OCTET STRING (SIZE(1))          OPTIONAL
                -- callOfferingNotAllowed           'xxxx xx01'B,
                -- callOfferingAllowed              'xxxx xx10'B
                -- network default is Call Offering not allowed
        }

GapCriteria {PARAMETERS-BOUND : bound} ::= CHOICE {
        calledAddressValue                      [0] Digits {bound},
        gapOnService                            [2] GapOnService,
        gapAllInTraffic                         [3] NULL,
        calledAddressAndService                 [29] SEQUENCE {
                                                calledAddressValue          [0] Digits {bound},
                                                serviceKey                  [1] ServiceKey
                                                },
        callingAddressAndService                [30] SEQUENCE {
                                                callingAddressValue         [0] Digits {bound},
                                                serviceKey                  [1] ServiceKey,
                                                locationNumber              [2] LocationNumber {bound}
                                                                                    OPTIONAL
                                                }
        }

-- Both calledAddressValue and callingAddressValue can be
-- incomplete numbers, in the sense that a limited amount of digits can be given.
--
-- For the handling of numbers starting with the same digit string refer to the detailed procedure
-- of the CallGap operation in 17.12.

GapOnService ::= SEQUENCE {
        serviceKey                              [0] ServiceKey,
        dpCriteria                              [1] EventTypeBCSM                   OPTIONAL
        }
GapIndicators ::= SEQUENCE {
        duration                                [0] Duration,
        gapInterval                             [1] Interval
        }
-- Indicates the gapping characteristics. No gapping when gapInterval equals 0, and gap all calls when
-- gapInterval equals –1.
```

```
GapTreatment  {PARAMETERS-BOUND : bound} ::= CHOICE {
        informationToSend [0] InformationToSend {bound},
        releaseCause                    [1] Cause {bound},
        both                            [2] SEQUENCE {
                                        informationToSend       [0] InformationToSend {bound},
                                        releaseCause            [1] Cause {bound}
                                         }
        }

-- The default value for Cause is the same as in ISUP.

GenericName  {PARAMETERS-BOUND : bound} ::= OCTET STRING (SIZE(
                                    bound.&minGenericNameLength..
                                    bound.&maxGenericNameLength))

GenericNumber  {PARAMETERS-BOUND : bound} ::= OCTET STRING (SIZE(
                                    bound.&minGenericNumberLength..
                                    bound.&maxGenericNumberLength))

-- Refer to Q.763 Generic Number for encoding.

GenericNumbers {PARAMETERS-BOUND : bound} ::= SET SIZE(1..bound.&numOfGenericNumbers) OF
GenericNumber {bound}

HighLayerCompatibilities ::= BIT STRING {
        telephony (0),
        facsimileGroup2-3 (1),
        facsimileGroup4classeI (2),
        teletexMixedMode (3),
        teletexProcessableMode (4),
        teletexBasicMode (5),
        syntaxBasedVideotex (6),
        internationalVideotex (7),
        telexService (8),
        messageHandlingSystem (9),
        osiApplication (10),
        audioVisual (11)
        }
HighLayerCompatibility ::= OCTET STRING (SIZE (highLayerCompatibilityLength))

-- Indicates the teleservice. For encoding, DSS 1 (Rec.Q.931) is used.

HoldCause ::= OCTET STRING -- defined by network operator.

-- Indicates the cause for holding the call.

InbandInfo  {PARAMETERS-BOUND : bound} ::= SEQUENCE {
        messageID                       [0] MessageID {bound},
        numberOfRepetitions             [1] INTEGER (1..127)            OPTIONAL,
        duration                        [2] INTEGER (0..32767)          OPTIONAL,
        interval                        [3] INTEGER (0.. 32767)         OPTIONAL
        }

-- Interval is the time in seconds between each repeated announcement. Duration is the total
-- amount of time in seconds, including repetitions and intervals.
-- The end of announcement is either the end of duration or numberOfRepetitions, whatever comes first.
-- duration with value 0 indicates infinite duration
```

```
InformationToRecord {PARAMETERS-BOUND : bound} ::= SEQUENCE {
        messageID                       [0] ElementaryMessageID               OPTIONAL,
        messageDeletionTimeOut          [1] INTEGER (1..3600)                 OPTIONAL,
        timeToRecord                    [3] INTEGER (0..bound.&maxRecordingTime)    OPTIONAL,
        controlDigits                   [4] SEQUENCE {
                endOfRecordingDigit     [0] OCTET STRING (SIZE(1..2))         OPTIONAL,
                cancelDigit             [1] OCTET STRING (SIZE(1..2))         OPTIONAL,
                replayDigit             [2] OCTET STRING (SIZE(1..2))         OPTIONAL,
                restartRecordingDigit   [3] OCTET STRING (SIZE(1..2))         OPTIONAL,
                restartAllowed          [4] BOOLEAN                           DEFAULT FALSE,
                replayAllowed           [5] BOOLEAN                           DEFAULT FALSE
                }
        }

InformationToSend {PARAMETERS-BOUND : bound} ::= CHOICE {
        inbandInfo                      [0] InbandInfo {bound},
        tone                            [1] Tone,
        displayInformation              [2] DisplayInformation {bound}
        }

InfoToSend {PARAMETERS-BOUND : bound} ::= CHOICE {
        messageID                       [0] MessageID {bound},
        toneId                          [1] ToneId,
        displayInformation              [2] DisplayInformation {bound}
        }

InfoType ::= ENUMERATED {
        numericString (0),
        characterString (1),
        iA5String (2)
        }
INServiceCompatibilityIndication {PARAMETERS-BOUND : bound} ::= SEQUENCE SIZE
(1..bound.&numOfInServiceCompatibilityIndLength) OF Entry

INServiceCompatibilityResponse ::= Entry

Integer4 ::= INTEGER(0..2147483647)

InteractionStrategy ::= ENUMERATED {
        stopOnError (1),
        bestEffort (2)
}

Interval ::= INTEGER (-1..60000)

-- Units are milliseconds. A –1 value denotes infinite.

InvokableService ::= ENUMERATED {
        callingLineIdentificationRestriction (1),
        connectedLineIdentificationRestriction (2),
        callWaiting (3),
        callHold (4),
        reverseCharging (5),
        explicitCallTransfer (6),
        callCompletionOnBusySubscriber (7)
        }

InvokeID ::= InvokeIdType

-- Operation invoke identifier.
```

```
IPAvailable {PARAMETERS-BOUND : bound} ::= OCTET STRING (SIZE (
                                        bound.&minIPAvailableLength..bound.&maxIPAvailableLength))
-- defined by network operator.
-- Indicates that the resource is available.

IPRoutingAddress {PARAMETERS-BOUND : bound} ::= CalledPartyNumber {bound}

-- Indicates the routing address for the IP.

IPSSPCapabilities {PARAMETERS-BOUND : bound} ::= OCTET STRING (SIZE (
                                                bound.&minIPSSPCapabilitiesLength..
                                                bound.&maxIPSSPCapabilitiesLength))
-- defined by network operator.
-- Indicates the SRF resources available at the SSP.

ISDNAccessRelatedInformation ::= OCTET STRING

-- Indicates the destination user network interface related information. Refer to the Q.763 Access
-- Transport parameter for encoding.

Language ::= PrintableString (SIZE (3) ) -- ISO 639 codes only;


LegID ::= CHOICE {
        sendingSideID                   [0] LegType,
        receivingSideID                 [1] LegType
        }

-- Indicates a reference to a specific party in a call. OPTIONAL denotes network-operator specific use
-- with a choice of unilateral ID assignment or bilateral ID assignment.
-- OPTIONAL for LegID also denotes the following:
-- when only one party exists in the call, this parameter is not needed (as no ambiguity exists);
-- when more than one party exists in the call, one of the following alternatives applies:
--        1. LegID is present and indicates which party is concerned.
--        2. LegID is not present and a default value is assumed (e.g. calling party in the case of the
--          ApplyCharging operation).
-- Choice between these two alternatives is kept a network-operator option.

LegType ::= OCTET STRING (SIZE(1))

leg1 LegType ::= ’01’H

leg2 LegType ::= ’02’H

LocationNumber {PARAMETERS-BOUND : bound} ::= OCTET STRING (SIZE (
                                                bound.&minLocationNumberLength..
                                                bound.&maxLocationNumberLength))

-- Indicates the Location Number for the calling party. Refer to Rec. Q.763 (White Book) for encoding.

MailBoxID {PARAMETERS-BOUND : bound} ::= OCTET STRING (SIZE(
                                                bound.&minMailBoxIDLength..bound.&maxMailBoxIDLength))

MaximumNumberOfCounters ::= INTEGER (1..numOfCounters)

Media ::= ENUMERATED {
        voiceMail (0),
        faxGroup3 (1),
        faxGroup4 (2)
        }
```

```
Message ::= ENUMERATED{
        rELease(77),
        rELeaseCOMPlete(90),
        fACility(98)
}
-- Specifies the message to be used for sending the component.

MessageID {PARAMETERS-BOUND : bound} ::= CHOICE {
        elementaryMessageID             [0] Integer4,
        text                            [1] SEQUENCE {
                                        messageContent                  [0] IA5String (SIZE (
                                        bound.&minMessageContentLength..
                                        bound.&maxMessageContentLength)),
                                        attributes      [1] OCTET STRING (SIZE (
                                        bound.&minAttributesLength..
                                        bound.&maxAttributesLength))    OPTIONAL

                                         },
        elementaryMessageIDs            [29] SEQUENCE SIZE (1.. bound.&numOfMessageIDs) OF Integer4,
        variableMessage                 [30] SEQUENCE {
                                        elementaryMessageID             [0] Integer4,
                                        variableParts                   [1] SEQUENCE SIZE (1..5)
                                        OF VariablePart {bound}
                                        }
        }

-- OPTIONAL denotes network-operator specific use.

MidCallControlInfo {PARAMETERS-BOUND : bound} ::= SEQUENCE SIZE (
                bound.&minMidCallControlInfoNum ..
                bound.&maxMidCallControlInfoNum) OF SEQUENCE {
                        midCallInfoType         [0]     MidCallInfoType {bound},
                        midCallReportType       [1]     ENUMERATED {
                                                                inMonitoringState(0),
                                                                inAnyState(1)
                                                        } DEFAULT inMonitoringState
                        }

MidCallInfo {PARAMETERS-BOUND : bound} ::= SEQUENCE {
        iNServiceControlCode                    [0] Digits {bound}
        }

MidCallInfoType {PARAMETERS-BOUND : bound} ::= SEQUENCE {
        iNServiceControlCodeLow                 [0] Digits {bound},
        iNServiceControlCodeHigh                [1] Digits {bound}                      OPTIONAL
        }

MiscCallInfo ::= SEQUENCE {
        messageType                     [0] ENUMERATED {
                                                request(0),
                                                notification(1)
                                                },
dpAssignment                                    [1] ENUMERATED {
                                                individualLine(0),
                                                groupBased(1),
                                                officeBased(2)
                                                }                                       OPTIONAL
        }

-- Indicates detection point related information.
```

```
MonitorMode  ::= ENUMERATED {
        interrupted(0),
        notifyAndContinue(1),
        transparent(2)
        }

-- Indicates the event is relayed and/or processed by the SSP.
-- If this parameter is used in the context of charging events, the following definitions apply for the
-- handling of charging events:
-- Interrupted means that the SSF notifies the SCF of the charging event using
-- EventNotificationCharging, does not process the event but discards it.
-- NotifyAndContinue means that SSF notifies the SCF of the charging event using
-- EventNotificationCharging, and continues processing the event or signal without waiting for SCF instructions.
-- Transparent means that the SSF does not notify the SCF of the event. This value is used to end the monitoring
-- of a previously requested charging event. Previously requested charging events are monitored
-- until ended by a transparent monitor mode, or until the end of the connection configuration.
-- For the use of this parameter in the context of BCSM events, refer to clause 17.

Notification ::= ENUMERATED {
        userAbandon (0),
        callFailure (1),
        noReply (2),
        callRelease (3),
        ssInvocation (4),
        creditLimitReached (5),
        callDuration (6),
        calledNumber (7),
        answeredCall (8)
        }

NotificationInformation {PARAMETERS-BOUND : bound} ::= CHOICE {
        userAbandonSpecificInfo          [0] SEQUENCE {...},
        callFailureSpecificInfo          [1] SEQUENCE {
                                         failureCause      [0] Cause {bound}                OPTIONAL,
                                         ...},
        noReplySpecificInfo              [2] SEQUENCE {...},
        callReleaseSpecificInfo          [3] SEQUENCE {
                                         releaseCause      [0] Cause {bound}                OPTIONAL,
                                         timeStamp         [1] DateAndTime                  OPTIONAL,
                                         ...},
        ssInvocationSpecificInfo         [4] SEQUENCE {
                                         invokedService    [0] InvokableService,
                                         ...},
        creditLimitReachedSpecificInfo   [5] SEQUENCE {
                                         timeStamp         [0] DateAndTime                  OPTIONAL,
                                         ...},
        callDurationSpecificInfo         [6] SEQUENCE {
                                         timeStamp         [0] DateAndTime                  OPTIONAL,
                                         ...},
        calledNumberSpecificInfo         [7] SEQUENCE {
                                         calledNumber      [0] CalledPartyNumber {bound} OPTIONAL,
                                         ...},
        answeredCallSpecificInfo         [8] SEQUENCE {
                                         timeStamp         [0] DateAndTime                  OPTIONAL,
                                         ...}
        }
```

```
NumberingPlan ::= OCTET STRING (SIZE(1))

-- Indicates the numbering plan for collecting the user information. Refer to the Q.763 Numbering Plan
-- Indicator field for encoding.

NumberMatch {PARAMETERS-BOUND : bound} ::= CHOICE {
        initialMatch                    [0] CalledPartyNumber {bound},
        totalMatch                      [1] CalledPartyNumber {bound}
        }

NumberOfDigits ::= INTEGER (1..255)

-- Indicates the number of digits to be collected

OperationCode ::= CHOICE {
        globalCode                      OBJECT IDENTIFIER,
        local                           INTEGER
        }

-- contains the operation value, or error value (object identifier), or problem value of the FACILITY IE,
-- and the argument, the result, or the reject part of the same FACILITY IE that are received with DSS 1
-- message from the user. (see 8.2.2/Q.932 for encoding)

OriginalCalledPartyID {PARAMETERS-BOUND : bound} ::= OCTET STRING (SIZE
                                        (bound.&minOriginalCalledPartyIDLength..
                                         bound.&maxOriginalCalledPartyIDLength))

-- Indicates the original called number. Refer to the Q.763 Original Called Number for encoding.

ProfileIdentifier {PARAMETERS-BOUND : bound}  ::= CHOICE {
access [0]  CalledPartyNumber {bound},
group           [1]  FacilityGroup
 }

-- Please note that ’CalledPartyNumber’ is used to address a subscriber access line.
-- The data type was reused from the existing types to avoid the definition of a new one.

Reason {PARAMETERS-BOUND : bound} ::= OCTET STRING(SIZE(
                bound.&minReasonLength..bound.&maxReasonLength))

ReceivedInformation {PARAMETERS-BOUND : bound} ::= SEQUENCE SIZE (
                                        bound.&minReceivedInformationLength..
                                        bound.&maxReceivedInformationLength) OF IA5String
-- size limit to be added

ReceivedStatus ::=ENUMERATED {
        messageComplete (0),
        messageInterrupted (1),
        messageTimeOut (2)
        }

RecordedMessageID ::= Integer4

RedirectingPartyID {PARAMETERS-BOUND : bound} ::= OCTET STRING (SIZE (
                                        bound.&minRedirectingPartyIDLength..
                                        bound.&maxRedirectingPartyIDLength))

-- Indicates redirecting number. Refer to the Q.763 Redirecting number for encoding.
```

```
RedirectionInformation ::= OCTET STRING (SIZE(2))

-- Indicates redirection information. Refer to the Q.763 Redirection Information for encoding.

RegistratorIdentifier  ::= OCTET STRING


ReportCondition ::= ENUMERATED {
        statusReport(0),
        timerExpired(1),
        cancelled(2)
        }

-- ReportCondition specifies the cause of sending "StatusReport"operation to the SCF

RequestedInformationList {PARAMETERS-BOUND : bound} ::= SEQUENCE SIZE (1..numOfInfoItems) OF
RequestedInformation {bound}

RequestedInformationTypeList ::= SEQUENCE SIZE (1..numOfInfoItems) OF RequestedInformationType

RequestedInformation {PARAMETERS-BOUND : bound} ::= SEQUENCE {
        requestedInformationType            [0] RequestedInformationType,
        requestedInformationValue           [1] RequestedInformationValue {bound}
        }

RequestedInformationType ::= ENUMERATED {
        callAttemptElapsedTime(0),
        callStopTime(1),
        callConnectedElapsedTime(2),
        calledAddress(3),
        releaseCause(30)
        }

RequestedInformationValue {PARAMETERS-BOUND : bound} ::= CHOICE {
        callAttemptElapsedTimeValue     [0] INTEGER (0..255),
        callStopTimeValue               [1] DateAndTime,
        callConnectedElapsedTimeValue[2] Integer4,
        calledAddressValue              [3] Digits {bound},

        releaseCauseValue               [30] Cause {bound}
        }

-- The callAttemptElapsedTimeValue is specified in seconds. The unit for the
-- callConnectedElapsedTimeValue is 100 milliseconds

RequestedNotifications {PARAMETERS-BOUND : bound} ::= SET OF CallConditions {bound}

RequestedType ::= INTEGER (0 .. 127)

RequestedUTSI {PARAMETERS-BOUND : bound} ::= SEQUENCE {
        uSIServiceIndicator             [0] USIServiceIndicator {bound},
        uSImonitorMode                  [1] USIMonitorMode,
        legID                           [2] LegID               DEFAULT sendingSideID:leg1
        }


RequestedUTSIList {PARAMETERS-BOUND : bound} ::= SEQUENCE SIZE
                                        bound.&minRequestedUTSINum..
                                         bound.&maxRequestedUTSINum) OF  RequestedUTSI {bound}
```

```
ResourceID {PARAMETERS-BOUND : bound} ::= CHOICE {
        lineID                              [0] Digits {bound},
        facilityGroupID                     [1] FacilityGroup,
        facilityGroupMemberID               [2] INTEGER,
        trunkGroupID                        [3] INTEGER
        }
-- Indicates a logical identifier for the physical termination resource.

ResourceStatus ::= ENUMERATED {
        busy(0),
        idle(1)
        }

ResponseCondition ::= ENUMERATED {
        intermediateResponse(0),
        lastResponse(1)
-- additional values are for further study
        }

-- ResponseCondition is used to identify the reason why ServiceFilteringResponse operation is sent.
-- intermediateResponse identifies that service filtering is running and the interval time is expired and
-- a call is received, or that service filtering is running and the threshold value is reached.
-- lastResponse identifies that the duration time is expired and service filtering has been finished or
-- that the stop time is met and service filtering has been finished.

RouteList {PARAMETERS-BOUND : bound} ::= SEQUENCE SIZE(1..3) OF OCTET STRING (SIZE
(bound.&minRouteListLength..bound.&maxRouteListLength))

-- Indicates a list of trunk groups or a route index. See Rec. Q.1224 for additional information on this item.

RoutingAddress {PARAMETERS-BOUND : bound} ::= CHOICE {
        routingProhibited                       [0] NULL,
        destinationRoutingAddress               [1] DestinationRoutingAddress {bound}
        }

ScfAddress {PARAMETERS-BOUND : bound} ::= OCTET STRING (SIZE
(bound.&minScfAddressLength..bound.&maxScfAddressLength))
-- ISDN address

ScfID {PARAMETERS-BOUND : bound} ::= OCTET STRING (SIZE
        (bound.&minScfIDLength..bound.&maxScfIDLength))

-- defined by network operator.
-- Indicates the SCF identity.
-- Used to derive the INAP address of the SCF to establish a connection between a requesting FE
-- and the specified SCF.
-- When ScfID is used in an operation which may cross an internetwork boundary, its encoding must
-- be understood in both networks; this requires bilateral agreement on the encoding.
-- A possible encoding is the SCCP address of the SCF, as defined in 3.5/Q.713.
-- Other encoding schemes are also possible.

SCIBillingChargingCharacteristics {PARAMETERS-BOUND : bound} ::= OCTET STRING (SIZE (
                                        bound.&minSCIBillingChargingLength..
                                        bound.&maxSCIBillingChargingLength))

-- This parameter indicates the billing and/or charging characteristics. Its content is network-operator specific.
-- An example datatype definition for this parameter is given below:
-- SCIBillingChargingCharacteristics ::= CHOICE {
--      chargeLevel                 [0] OCTET STRING (SIZE (min..max),
--      chargePulses                [1] Integer4,
--	chargeMessages		[2] OCTET STRING (SIZE (min..max)
--	}
-- Depending on the applied charging scenario the following information elements
-- can be included (refer to Appendix II/Q.1214):
-- chargeLevel (scenario 3.2)
-- chargePulses (scenario 3.2)
-- chargeMessages (scenario 3.2)

ServiceAddressInformation ::= SEQUENCE {
	serviceKey		[0] ServiceKey				OPTIONAL,
	miscCallInfo		[1] MiscCallInfo,
	triggerType		[2] TriggerType				OPTIONAL
	}

-- Information that represents the result of trigger analysis and allows the SCF to choose the appropriate service logic

ServiceInteractionIndicators {PARAMETERS-BOUND : bound} ::= OCTET STRING (SIZE (
				bound.&minServiceInteractionIndicatorsLength..
				bound.&maxServiceInteractionIndicatorsLength))

-- Indicators which are exchanged between SSP and SCP to resolve interactions between IN-based services
-- and network-based services, respectively between different IN-based services.
-- The contents are network specific.
-- Note this parameter is kept in CS-2 for backward compatibility to CS-1R, for CS-2 see new
-- parameter ServiceInteractionIndicatorsTwo

ServiceInteractionIndicatorsTwo ::=	SEQUENCE {
	forwardServiceInteractionInd		[0] ForwardServiceInteractionInd		OPTIONAL,
		-- applicable to operations IDP, CON, ICA.
	backwardServiceInteractionInd		[1] BackwardServiceInteractionInd		OPTIONAL,
		-- applicable to operations IDP, CON, CTR, ETC.
	BothwayThroughConnectionInd		[2] BothwayThroughConnectionInd		OPTIONAL,
		-- applicable to operations CTR, ETC.
	suspendTimer				[3] SuspendTimer				OPTIONAL,
		-- applicable to operations CON, ICA.
	ConnectedNumberTreatmentInd		[4] ConnectedNumberTreatmentInd		OPTIONAL,
		-- applicable to operations CON, CTR, ETC.
	suppressCallDiversionNotification	[5] BOOLEAN				OPTIONAL,
		-- applicable to CON, ICA
	suppressCallTransferNotification	[6] BOOLEAN				OPTIONAL,
		-- applicable to CON, ICA
	allowCdINNoPresentationInd		[7] BOOLEAN				OPTIONAL,
		-- applicable to CON, ICA
		-- indicates whether the Number Presentation not allowed indicator of the ISUP
		-- "called IN number" shall be set to presentation allowed (TRUE) or presentation not allowed (FALSE)
	userDialogueDurationInd			[8] BOOLEAN				DEFAULT TRUE,
		-- applicable when interaction with the user is required, if the interaction
		-- TRUE means the user interaction may last longer than 90 seconds. Otherwise the
		-- indicator should be set to FALSE.
		-- used for delaying ISUP T9 timer.
	...
	}
-- Indicators which are exchanged between SSP and SCP to resolve interactions between IN-based services
-- and network-based services, respectively between different IN-based services.

ServiceKey ::= Integer4

-- Information that allows the SCF to choose the appropriate service logic.
```

```
ServiceProfileIdentifier ::= OCTET STRING

-- Indicates a particular ISDN terminal. Refer to Rec. Q.932 for encoding.

ServingAreaID {PARAMETERS-BOUND : bound} ::= LocationNumber {bound}

-- Identifies the local serving area where a network provider operates. Uses the LocationNumber
-- format which is based on the Q.763 Location Number format.
-- The Nature of Address indicator field shall be set to "Spare" (value 00000000).
-- The Numbering Plan Indicator field shall be set to "Spare" (value 000).
-- Defined by the network operator.

SFBillingChargingCharacteristics {PARAMETERS-BOUND : bound} ::= OCTET STRING (SIZE (
                                                bound.&minSFBillingChargingLength..
                                                bound.&maxSFBillingChargingLength))
-- This parameter indicates the billing and/or charging characteristics for filtered calls.
-- Its content is network-operator specific
SubscriberId {PARAMETERS-BOUND : bound} ::= GenericNumber {bound}

SupplementaryServices ::= BIT STRING {
        callingLineIdentificationPresentation (1),
        callingLineIdentificationRestriction (2),
        connectedLineIdentificationPresentation (3),
        connectedLineIdentificationRestriction (4),
        callForwardingOnNoReply (5),
        callForwardingUnconditional (6),
        callForwardingOnBusy (7),
        callForwardingOnNotReachable (8),
        callWaiting (9),
        callHold (10),
        reverseCharging (11),
        explicitCallTransfer (12),
        callCompletionOnBusySubscriber (13),
        adviceOfChargeOnStart (14),
        adviceOfChargeAtEnd (15),
        adviceOfChargeDuringCall (16),
        timeDependentRouting (17),
        callingPartingDependentRouting (18),
        outgoingCallBarring (19),
        incomingCallBarring (20)
        }

SuspendTimer ::= INTEGER (0..120) -- value in seconds

TargetLineIdentifier {PARAMETERS-BOUND : bound} ::= CHOICE {
        individual                      [0] CalledPartyNumber {bound},
        group                           [1] FacilityGroup
        }

TerminalType ::= ENUMERATED {
        unknown(0),
        dialPulse(1),
        dtmf(2),
        isdn(3),
        isdnNoDtmf(4),
        spare(16)
        }

-- Identifies the terminal type so that the SCF can specify, to the SRF, the appropriate type of capability
-- (voice recognition, DTMF, display capability, etc.). Since present signalling systems do not convey
-- terminal type, this parameter applies only at originating or terminating local exchanges.
```

```
TimerID ::= ENUMERATED {
        tssf(0)
        -- others ffs
        }

-- Indicates the timer to be reset.

TimerValue ::= Integer4

-- Indicates the timer value (in seconds).

Tone ::= SEQUENCE {
        toneID                          [0] Integer4,
        duration                        [1] Integer4        OPTIONAL
        }

-- The duration specifies the length of the tone in seconds, value 0 indicates infinite duration.

ToneId ::= CHOICE {
        local                           [0] Integer4,
        global                          [1] OBJECT IDENTIFIER
        }

TraceInformation {PARAMETERS-BOUND : bound} ::= SEQUENCE OF TraceItem { bound}

TraceItem {PARAMETERS-BOUND : bound} ::= SET {scf [0] ScfID { bound},...}

TravellingClassMark {PARAMETERS-BOUND : bound} ::= LocationNumber {bound}

-- Indicates travelling class mark information.
-- Uses the LocationNumber format which is based on the Q.763 Location Number format.
-- The Nature of Address indicator field shall be set to "Spare" (value 00000000).
-- The Numbering Plan Indicator field shall be set to "Spare" (value 000).
-- Maximum 2 digits.

TriggerDataIdentifier {PARAMETERS-BOUND : bound} ::= SEQUENCE {
        triggerID                       [0] EventTypeBCSM,
        profileIdentifier               [1] ProfileIdentifier {bound},
        extensions                      [2] SEQUENCE SIZE(1..bound.&numOfExtensions)   OF
                                            ExtensionField {bound}          OPTIONAL
        }

-- It is for further study whether all TDP types really apply


TriggerType ::= ENUMERATED {
                featureActivation(0),
                verticalServiceCode(1),
                customizedAccess(2),
                customizedIntercom(3),
                emergencyService(12),
                aFR(13),
                sharedIOTrunk(14),
                offHookDelay(17),
                channelSetupPRI(18),
                tNoAnswer(25),
                tBusy(26),
                oCalledPartyBusy(27),
                oNoAnswer(29),
                originationAttemptAuthorized(30),
                    oAnswer(31),
                    oDisconnect(32),
                    termAttemptAuthorized(33),
                    tAnswer(34),
                    tDisconnect(35)
                    -- Private (ffs)
                    }

-- The type of trigger which caused call suspension
-- 4-11: Reserved; 15,16: Reserved; 19-24: Reserved

UnavailableNetworkResource ::= ENUMERATED {
          unavailableResources(0),
          componentFailure(1),
          basicCallProcessingException(2),
          resourceStatusFailure(3),
          endUserFailure(4)
          }

-- Indicates the network resource that failed.

UserCredit {PARAMETERS-BOUND : bound} ::= Credit {bound}

UserInfo {PARAMETERS-BOUND : bound} ::= SEQUENCE OF UserInformation {bound}

UserInformation {PARAMETERS-BOUND : bound} ::= SEQUENCE {
          infoToSend                    [0] InfoToSend {bound},
          constraints                   [1] Constraints,
          errorInfo                     [2] InfoToSend {bound} OPTIONAL
          }

UserInteractionModes ::= BIT STRING {
          voiceMessage (0),
          tone (1),
          display (2)
          }

USIInformation {PARAMETERS-BOUND : bound} ::= OCTET STRING (SIZE (
                         bound.&minUSIInformationLength..bound.&maxUSIInformationLength))


USIMonitorMode ::= ENUMERATED {
          monitoringActive                        (0),
          monitoringInactive                      (1)
          }

--  Indicates if the monitoring relationship for the specified UTSI IE should be activated or deactivated.


USIServiceIndicator {PARAMETERS-BOUND : bound} ::= OCTET STRING (SIZE (
                                                  bound.&minUSIServiceIndicatorLength..
                                                  bound.&maxUSIServiceIndicatorLength))
```

```
VariablePart {PARAMETERS-BOUND : bound} ::= CHOICE {
        integer                         [0] Integer4,
        number                          [1] Digits { bound},          -- Generic digits
        time                            [2] OCTET STRING (SIZE(2)),   -- HH:MM, BCD coded
        date                            [3] OCTET STRING (SIZE(3)),   -- YYMMDD, BCD coded
        price                           [4] OCTET STRING (SIZE(4))
        }

-- Indicates the variable part of the message.
-- BCD coded variable parts are encoded as described in the examples below.
-- For example, time = 12:15 would be encoded as:
--      Bits                            HGFE        DCBA
--      leading octet                      2           1
--                                         5           1
-- date = 1993 September 30th would be encoded as:
--      Bits                            HGFE        DCBA
--      leading octet                      3           9
--                                         9           0
--                                         0           3

-- The Definition of range of constants Follows

highLayerCompatibilityLength                    INTEGER ::= 2
minCauseLength                                  INTEGER ::= 2
numOfCounters                                   INTEGER ::= 100
numOfInfoItems                                  INTEGER ::= 5

maxCreditUnit                                   INTEGER ::= 65536

END
```

4.2 Error types

```
IN-CS2-errortypes {itu-t recommendation q 1228 modules(0) in-cs2-errortypes (1) version1(0)}

-- This module contains the type definitions for the IN CS-2 errors.
-- Where a parameter of type CHOICE is tagged with a specific tag value, the tag is automatically
-- replaced with an EXPLICIT tag of the same value.

DEFINITIONS IMPLICIT TAGS ::=

BEGIN

IMPORTS
ros-InformationObjects, datatypes, errorcodes FROM IN-CS2-object-identifiers
                { itu-t recommendation q 1228 module(0) in-cs2-object-identifiers(17) version1(0) }
        ERROR
FROM Remote-Operations-Information-Objects ros-InformationObjects
        InvokeID,
        UnavailableNetworkResource
FROM IN-CS2-datatypes datatypes

        errcode-cancelled,
        errcode-cancelFailed,
        errcode-chainingRefused,
        errcode-eTCFailed,
        errcode-improperCallerResponse,
        errcode-missingCustomerRecord,
        errcode-missingParameter,
        errcode-parameterOutOfRange,
        errcode-requestedInfoError,
        errcode-systemFailure,
        errcode-taskRefused,
        errcode-unavailableResource,
        errcode-unexpectedComponentSequence,
        errcode-unexpectedDataValue,
        errcode-unexpectedParameter,
        errcode-unknownLegID,
        errcode-unknownRecordedMessageID,
        errcode-unknownResource,
        errcode-unknownSubscriber
FROM IN-CS2-errorcodes errorcodes;

-- TYPE DEFINITION FOR    IN CS-2    ERRORS FOLLOWS

cancelled ERROR ::= {
        CODE                     errcode-cancelled
        }
-- The operation has been cancelled.

cancelFailed ERROR ::= {
        PARAMETER                SEQUENCE {
                                 problem                  [0] ENUMERATED {
                                                          unknownOperation(0),
                                                          tooLate(1),
                                                          operationNotCancellable(2)
                                                          },
                                 operation                [1] InvokeID
                                 }
        CODE                     errcode-cancelFailed
        }
-- The operation failed to be cancelled.

chainingRefused ERROR ::= {
        CODE                     errcode-chainingRefused
        }


eTCFailed ERROR ::= {
        CODE                     errcode-eTCFailed
        }
-- The establish temporary connection failed.

improperCallerResponse ERROR ::= {
        CODE                     errcode-improperCallerResponse
        }
-- The caller response was not as expected.

missingCustomerRecord ERROR ::= {
        CODE                     errcode-missingCustomerRecord
        }
-- The Service Logic Program could not be found in the SCF.

missingParameter ERROR ::= {
        CODE                     errcode-missingParameter
        }
-- An expected optional parameter was not received.
```

```
parameterOutOfRange ERROR ::= {
        CODE                    errcode-parameterOutOfRange
        }
-- The parameter was not as expected (e.g. missing or out-of-range).

requestedInfoError ERROR ::= {
        PARAMETER               ENUMERATED {
                                unknownRequestedInfo(1),
                                requestedInfoNotAvailable(2)
                                -- other values FFS
                                }
        CODE                    errcode-requestedInfoError
        }

-- The requested information cannot be found.

systemFailure ERROR ::= {
        PARAMETER               UnavailableNetworkResource
        CODE                    errcode-systemFailure
        }
-- The operation could not be completed due to a system failure at the serving physical entity.

taskRefused ERROR ::= {
        PARAMETER               ENUMERATED {
                                generic(0),
                                unobtainable (1),
                                congestion(2)
                                --other values FFS
                                }
        CODE                    errcode-taskRefused
        }
-- An entity normally capable of the task requested cannot or chooses not to perform the task at this
-- time. This includes error situations like congestion and unobtainable address as used in e.g. the
-- connect operation.

unavailableResource ERROR ::= {
        CODE                    errcode-unavailableResource
        }
-- A requested resource is not available at the serving entity.

unexpectedComponentSequence ERROR ::= {
        CODE                    errcode-unexpectedComponentSequence
        }
-- An incorrect sequence of Components was received (e.g."DisconnectForwardConnection"
-- followed by "PlayAnnouncement").

unexpectedDataValue ERROR ::= {
        CODE                    errcode-unexpectedDataValue
        }
-- The data value was not as expected (e.g. routing number expected but billing number received)

unexpectedParameter ERROR ::= {
        CODE                    errcode-unexpectedParameter
        }
-- A parameter received was not expected.

unknownLegID ERROR ::= {
        CODE                    errcode-unknownLegID
        }
-- Leg not known to the SSF.
```

```
unknownResource ERROR ::= {
        CODE                        errcode-unknownResource
        }
-- Resource whose status is being requested is not known to the serving entity.


END
```

4.3 Operations codes

```
IN-CS2-operationcodes {itu-t recommendation q 1228 modules(0) in-cs2-operationcodes (2) version1(0)}

DEFINITIONS ::=

BEGIN

IMPORTS

ros-InformationObjects FROM  IN-CS2-object-identifiers
                {itu-t recommendation q 1228 module(0) in-cs2-object-identifiers(17) version1(0) }
        Code
FROM Remote-Operations-Information-Objects ros-InformationObjects
;

-- the operations are grouped by the identified operation packages.

-- SCF activation Package

        opcode-initialDP                                        Code ::= local : 0

-- Basic BCP DP Package

        opcode-originationAttemptAuthorized                     Code ::= local : 1
        opcode-collectedInformation                             Code ::= local : 2
        opcode-analysedInformation                              Code ::= local : 3
        opcode-routeSelectFailure                               Code ::= local : 4
        opcode-oCalledPartyBusy                                 Code ::= local : 5
        opcode-oNoAnswer                                        Code ::= local : 6
        opcode-oAnswer                                          Code ::= local : 7
        opcode-oDisconnect                                      Code ::= local : 8
        opcode-termAttemptAuthorized                            Code ::= local : 9
        opcode-tBusy                                            Code ::= local : 10
        opcode-tNoAnswer                                        Code ::= local : 11
        opcode-tAnswer                                          Code ::= local : 12
        opcode-tDisconnect                                      Code ::= local : 13

        opcode-facilitySelectedAndAvailable                     Code ::= local : 80
        opcode-originationAttempt                               Code ::= local : 81
        opcode-terminationAttempt                               Code ::= local : 82
        opcode-oAbandon                                         Code ::= local : 83


-- Advanced BCP DP Package

        opcode-oMidCall                                         Code ::= local : 14
        opcode-tMidCall                                         Code ::= local : 15

        opcode-oSuspended                                       Code ::= local : 84
        opcode-tSuspended                                       Code ::= local : 85
```

```
-- SCF/SRF activation of assist Package

          opcode-assistRequestInstructions                    Code ::= local : 16

-- Assist connection establishment Package

          opcode-establishTemporaryConnection                 Code ::= local : 17

-- Generic disconnect resource Package
          opcode-disconnectForwardConnection                  Code ::= local : 18
          opcode-dFCWithArgument                              Code ::= local : 86

-- Non-assisted connection establishment Package
          opcode-connectToResource                            Code ::= local : 19

-- Connect Package (elementary SSF function)
          opcode-connect                                      Code ::= local : 20

-- Call handling Package (elementary SSF function)

          opcode-holdCallInNetwork                            Code ::= local : 21
          opcode-releaseCall                                  Code ::= local : 22

-- BCSM Event handling Package

          opcode-requestReportBCSMEvent                       Code ::= local : 23
          opcode-eventReportBCSM                              Code ::= local : 24

-- Charging Event handling Package

          opcode-requestNotificationChargingEvent             Code ::= local : 25
          opcode-eventNotificationCharging                    Code ::= local : 26

-- SSF call processing Package

          opcode-collectInformation                           Code ::= local : 27
          opcode-analyseInformation                           Code ::= local : 28
          opcode-selectRoute                                  Code ::= local : 29
          opcode-selectFacility                               Code ::= local : 30
          opcode-continue                                     Code ::= local : 31
          opcode-authorizeTermination                         Code ::= local : 87

-- SCF call initiation Package

          opcode-initiateCallAttempt                          Code ::= local : 32

-- Timer Package

          opcode-resetTimer                                   Code ::= local : 33

-- Billing Package

          opcode-furnishChargingInformation                   Code ::= local : 34

-- Charging Package

          opcode-applyCharging                                Code ::= local : 35
          opcode-applyChargingReport                          Code ::= local : 36
```

```
-- Status reporting Package

        opcode-requestCurrentStatusReport                 Code ::= local : 37
        opcode-requestEveryStatusChangeReport             Code ::= local : 38
        opcode-requestFirstStatusMatchReport              Code ::= local : 39
        opcode-statusReport                               Code ::= local : 40

-- Traffic management Package

        opcode-callGap                                    Code ::= local : 41

-- Service management Package

        opcode-activateServiceFiltering                   Code ::= local : 42
        opcode-serviceFilteringResponse                   Code ::= local : 43

-- Call report Package

        opcode-callInformationReport                      Code ::= local : 44
        opcode-callInformationRequest                     Code ::= local : 45

-- Signalling control Package

        opcode-sendChargingInformation                    Code ::= local : 46

-- Specialized resource control Package

        opcode-playAnnouncement                           Code ::= local : 47
        opcode-promptAndCollectUserInformation            Code ::= local : 48
        opcode-specializedResourceReport                  Code ::= local : 49

-- Cancel Package

        opcode-cancel                                     Code ::= local : 53
        opcode-cancelStatusReportRequest                  Code ::= local : 54

-- Activity Test Package

        opcode-activityTest                               Code ::= local : 55

-- CPH Response Package

        opcode-continueWithArgument                       Code ::= local : 88
        opcode-createCallSegmentAssociation               Code ::= local : 89
        opcode-disconnectLeg                              Code ::= local : 90
        opcode-mergeCallSegments                          Code ::= local : 91
        opcode-moveCallSegments                           Code ::= local : 92
        opcode-moveLeg                                    Code ::= local : 93
        opcode-reconnect                                  Code ::= local : 94
        opcode-splitLeg                                   Code ::= local : 95

-- Exception Inform Package

        opcode-entityReleased                             Code ::= local : 96

-- Trigger Management Package

        opcode-manageTriggerData                          Code ::= local : 97
```

```
-- USI Handling Package

        opcode-requestReportUTSI                      Code ::= local : 98
        opcode-sendSTUI                               Code ::= local : 100
        opcode-reportUTSI                             Code ::= local : 101

-- Facility IE Handling Package

        opcode-sendFacilityInformation                Code ::= local : 102
        opcode-requestReportFacilityEvent             Code ::= local : 103
        opcode-eventReportFacility                    Code ::= local : 104

-- SRF/SCF interface
        opcode-promptAndReceiveMessage                Code ::= local : 107
        opcode-scriptInformation                      Code ::= local : 108
        opcode-scriptEvent                            Code ::= local : 109
        opcode-scriptRun                              Code ::= local : 110
        opcode-scriptClose                            Code ::= local : 111

-- SCF/SCF interface

        opcode-establishChargingRecord                Code ::= local : 112
        opcode-handlingInformationRequest             Code ::= local : 113
        opcode-handlingInformationResult              Code ::= local : 114
        opcode-networkCapability                      Code ::= local : 115
        opcode-notificationProvided                   Code ::= local : 116
        opcode-confirmedNotificationProvided          Code ::= local : 117
        opcode-provideUserInformation                 Code ::= local : 118
        opcode-confirmedReportChargingInformation     Code ::= local : 119
        opcode-reportChargingInformation              Code ::= local : 120
        opcode-requestNotification                    Code ::= local : 121

-- CUSF/SCF interface
        opcode-activationReceivedAndAuthorized        Code ::= local : 122
        opcode-initiateAssociation                    Code ::= local : 123
        opcode-associationReleaseRequested            Code ::= local : 124
        opcode-componentReceived                      Code ::= local : 125
        opcode-releaseAssociation                     Code ::= local : 126
        opcode-requestReportBCUSMEvent                Code ::= local : 127
        opcode-sendComponent                          Code ::= local : 130

END
```

4.4 Error codes

```
IN-CS2-errorcodes { itu-t recommendation q 1228 modules(0) in-cs2-errorcodes (3) version1(0)}

DEFINITIONS ::=

BEGIN

IMPORTS

ros-InformationObjects FROM  IN-CS2-object-identifiers
            { itu-t recommendation q 1228 module(0) in-cs2-object-identifiers(17) version1(0) }
    Code
FROM Remote-Operations-Information-Objects ros-InformationObjects
;
        errcode-cancelled                             Code ::= local : 0
        errcode-cancelFailed                          Code ::= local : 1
        errcode-eTCFailed                                   Code ::= local : 3
        errcode-improperCallerResponse                      Code ::= local : 4
        errcode-missingCustomerRecord                       Code ::= local : 6
        errcode-missingParameter                            Code ::= local : 7
        errcode-parameterOutOfRange                         Code ::= local : 8
        errcode-requestedInfoError                          Code ::= local : 10
        errcode-systemFailure                               Code ::= local : 11
        errcode-taskRefused                                 Code ::= local : 12
        errcode-unavailableResource                         Code ::= local : 13
        errcode-unexpectedComponentSequence                 Code ::= local : 14
        errcode-unexpectedDataValue                         Code ::= local : 15
        errcode-unexpectedParameter                         Code ::= local : 16
        errcode-unknownLegID                                Code ::= local : 17
        errcode-unknownResource                             Code ::= local : 18

-- Error codes for the new IN CS-2 error types follows
        errcode-scfReferral                                 Code ::= local : 21
        errcode-scfTaskRefused                              Code ::= local : 22
        errcode-chainingRefused                             Code ::= local : 23


END
```

4.5 Classes

```
IN-CS2-classes { itu-t recommendation q 1228 modules(0) in-cs2-classes (4) version1(0)}

DEFINITIONS ::=

BEGIN

IMPORTS

        ROS-OBJECT-CLASS, CONTRACT, OPERATION-PACKAGE, Code, OPERATION,
        CONNECTION-PACKAGE
FROM Remote-Operations-Information-Objects ros-InformationObjects
        emptyBind, emptyUnbind
FROM Remote-Operations-Useful-Definitions ros-UsefulDefinitions

        id-package-emptyConnection,
        id-rosObject-scf,
        id-rosObject-cusf,
        id-rosObject-dssp,
        id-rosObject-srf,
        id-rosObject-ssf,
        ros-InformationObjects,
        ros-UsefulDefinitions,
        ssf-scf-Protocol,
        scf-cusf-Protocol,
        scf-scf-Protocol,
        scf-srf-Protocol,
        scf-sdf-Protocol,
        datatypes

FROM IN-CS2-object-identifiers {itu-t recommendation q 1228 modules(0) in-cs2-object-identifiers (17)
version1(0)}

        inCs2AssistHandoffSsfToScf,
        inCs2ScfToSsfDpSpecific,
        inCs2ScfToSsfGeneric,
        inCs2ScfToSsfStatusReporting,
        inCs2ScfToSsfTrafficManagement,
        inCs2SsfToScfDpSpecific,
        inCs2SsfToScfGeneric,
        inCs2SsfToScfServiceManagement

FROM IN-CS2-SSF-SCF-pkgs-contracts-acs ssf-scf-Protocol

        cusf-scf-contract,
        scf-cusf-contract
FROM IN-CS2-SCF-CUSF-pkgs-contracts-acs scf-cusf-Protocol

        dsspContract,
        scf-scfContract

FROM IN-CS2-SCF-SCF-pkgs-contracts-acs scf-scf-Protocol

        srf-scf-contract

FROM IN-CS2-SCF-SRF-pkgs-contracts-acs scf-srf-Protocol

        dapContract
        FROM IN-CS2-SCF-SDF-Protocol scf-sdf-Protocol
        CriticalityType

FROM IN-CS2-datatypes datatypes
;
ssf ROS-OBJECT-CLASS ::= {
        INITIATES               {inCs2SsfToScfGeneric|
                                inCs2SsfToScfDpSpecific|
                                inCs2AssistHandoffSsfToScf|
                                inCs2SsfToScfServiceManagement}
        RESPONDS                {inCs2ScfToSsfGeneric|
                                inCs2ScfToSsfDpSpecific|
                                inCs2ScfToSsfTrafficManagement|
                                inCs2SsfToScfServiceManagement|
                                inCs2ScfToSsfStatusReporting}
        ID                      id-rosObject-ssf}

srf ROS-OBJECT-CLASS ::= {
        INITIATES               {srf-scf-contract}
        ID                      id-rosObject-srf
        }

cusf ROS-OBJECT-CLASS ::= {
        INITIATES               {cusf-scf-contract}
        RESPONDS                {scf-cusf-contract }
        ID                      id-rosObject-cusf}

dssp  ROS-OBJECT-CLASS ::= {
        BOTH                    {dsspContract}
        ID                      id-rosObject-dssp
        }

scf ROS-OBJECT-CLASS ::= {
        INITIATES               {inCs2ScfToSsfGeneric|
                                inCs2ScfToSsfDpSpecific|
                                inCs2ScfToSsfTrafficManagement|
                                inCs2ScfToSsfServiceManagement|
                                inCs2ScfToSsfTriggerManagement
                                inCs2ScfToSsfStatusReporting |
-- scf to cusf contracts
                                          scf-cusf-contract |
-- scf to scf contracts
                                          scf-scfContract |
                                          dsspContract |
-- sdf to scf contracts
                                          dapContract
                                    }
      RESPONDS                            {inCs2SsfToScfGeneric|
                                          inCs2SsfToScfDpSpecific|
                                          inCs2AssistHandoffSsfToScf|
                                          inCs2SsfToScfServiceManagement|
-- cusf to scf contracts
                                          cusf-scf-contract |
-- srf to scf contracts
                                          srf-scf-contract |
-- scf to scf contracts
                                          scf-scfContract |
                                          dsspContract

                                          }
      ID                                  id-rosObject-scf}

EXTENSION ::= CLASS {
      &ExtensionType,
      &criticality                        CriticalityType DEFAULT ignore,
      &id                                 Code
      }
WITH SYNTAX {
      EXTENSION-SYNTAX                    &ExtensionType
      CRITICALITY                         &criticality
      IDENTIFIED BY                       &id
      }

-- Example of addition of an extension named 'Some Network Specific Indicator' of type
-- BOOLEAN, with criticality 'abort' and to be identified as extension number 1
-- Example of definition using the above information object class:
--
-- SomeNetworkSpecificIndicator  EXTENSION ::= {
--      EXTENSION-SYNTAX    BOOLEAN
--      CRITICALITY         abort
--      IDENTIFIED BY       local : 1
--      }

-- Example of transfer syntax, using the ExtensionField datatype as specified in 4.1.
-- Assuming the value of the extension is set to TRUE, the extensions parameter
-- becomes a Sequence of type INTEGER ::= 1, criticality ENUMERATED ::= 1 and value [1]
-- EXPLICIT BOOLEAN ::= TRUE.
--
-- Use of Q.1400 defined Extension is ffs
-- In addition the extension mechanism marker is used to identify the future minor additions to INAP.

firstExtension EXTENSION ::= {
      EXTENSION-SYNTAX                    NULL
      CRITICALITY                         ignore
      IDENTIFIED BY                       local:1
      }
-- firstExtension is just an example.
SupportedExtensions  {PARAMETERS-BOUND : bound}  EXTENSION ::= {firstExtension , ...
-- full set of network operator extensions --}
-- SupportedExtension is the full set of the network operator extensions.
```

```
UISCRIPT ::= CLASS {
        &SpecificInfo                   OPTIONAL,
        &Result                         OPTIONAL,
        &id                             Code
        }

WITH SYNTAX {
        [WITH-SPECIFICINFO              &SpecificInfo]
        [WITH-RESULT                    &Result]
        IDENTIFIED BY                   &id
}

firstScript UISCRIPT ::=
        {
        IDENTIFIED BY local:1
        }
-- firstScript is just an example.

SupportedUIScripts {PARAMETERS-BOUND : bound}  UISCRIPT ::= {firstScript , ...
-- full set of User Interaction script --}
-- SupportedUIScripts is the full set of User Interaction scripts.

inEmptyUnbind OPERATION ::= {
        RETURN RESULT                   FALSE
        ALWAYS RESPONDS                 FALSE }
emptyConnectionPackage CONNECTION-PACKAGE ::= {
        BIND                            emptyBind
        UNBIND                          inEmptyUnbind
        RESPONDER UNBIND                TRUE
        ID                              id-package-emptyConnection
        }
PARAMETERS-BOUND ::= CLASS
{
        &minAChBillingChargingLength                    INTEGER,
        &maxAChBillingChargingLength                    INTEGER,
        &minAttributesLength                            INTEGER,
        &maxAttributesLength                            INTEGER,
        &minBackwardGVNSLength                          INTEGER,
        &maxBackwardGVNSLength                          INTEGER,
        &maxBearerCapabilityLength                      INTEGER,
        &minCalledPartyNumberLength                     INTEGER,
        &maxCalledPartyNumberLength                     INTEGER,
        &minCallingPartyNumberLength                    INTEGER,
        &maxCallingPartyNumberLength                    INTEGER,
        &minCallResultLength                            INTEGER,
        &maxCallResultLength                            INTEGER,
        &maxCauseLength                                 INTEGER,
        &minDigitsLength                                INTEGER,
        &maxDigitsLength                                INTEGER,
        &minDisplayInformationLength                    INTEGER,
        &maxDisplayInformationLength                    INTEGER,
        &minEventSpecificInformationChargingLength      INTEGER,
        &maxEventSpecificInformationChargingLength      INTEGER,
        &minEventTypeChargingLength                     INTEGER,
        &maxEventTypeChargingLength                     INTEGER,
        &minFCIBillingChargingLength                    INTEGER,
        &maxFCIBillingChargingLength                    INTEGER,
        &minForwardGVNSLength                           INTEGER,
        &maxForwardGVNSLength                           INTEGER,
        &minGenericNameLength                           INTEGER,
        &maxGenericNameLength                          INTEGER,
        &minGenericNumberLength                        INTEGER,
        &maxGenericNumberLength                        INTEGER,
        &maxInitialTimeInterval                        INTEGER,
        &maxINServiceCompatibilityIndLength            INTEGER,
        &minIPAvailableLength                          INTEGER,
        &maxIPAvailableLength                          INTEGER,
        &minIPSSPCapabilitiesLength                    INTEGER,
        &maxIPSSPCapabilitiesLength                    INTEGER,
        &minLocationNumberLength                       INTEGER,
        &maxLocationNumberLength                       INTEGER,
        &minMailBoxIDLength                            INTEGER,
        &maxMailBoxIDLength                            INTEGER,
        &minMessageContentLength                       INTEGER,
        &maxMessageContentLength                       INTEGER,
        &minMidCallControlInfoNum                      INTEGER,
        &maxMidCallControlInfoNum                      INTEGER,
        &minOriginalCalledPartyIDLength                INTEGER,
        &maxOriginalCalledPartyIDLength                INTEGER,
        &minReasonLength                               INTEGER,
        &maxReasonLength                               INTEGER,
        &minReceivedInformationLength                  INTEGER,
        &maxReceivedInformationLength                  INTEGER,
        &maxRecordedMessageUnits                       INTEGER,
        &maxRecordingTime                              INTEGER,
        &minRedirectingPartyIDLength                   INTEGER,
        &maxRedirectingPartyIDLength                   INTEGER,
        &minRequestedUTSINum                           INTEGER,
        &maxRequestedUTSINum                           INTEGER,
        &minRouteListLength                            INTEGER,
        &maxRouteListLength                            INTEGER,
        &minScfIDLength                                INTEGER,
        &maxScfIDLength                                INTEGER,
        &minScfAddressLength                           INTEGER,
        &maxScfAddressLength                           INTEGER,
        &minSCIBillingChargingLength                   INTEGER,
        &maxSCIBillingChargingLength                   INTEGER,
        &minServiceInteractionIndicatorsLength         INTEGER,
        &maxServiceInteractionIndicatorsLength         INTEGER,
        &minSFBillingChargingLength                    INTEGER,
        &maxSFBillingChargingLength                    INTEGER,
        &minUSIInformationLength                       INTEGER,
        &maxUSIInformationLength                       INTEGER,
        &minUSIServiceIndicatorLength                  INTEGER,
        &maxUSIServiceIndicatorLength                  INTEGER,
        &numOfBCSMEvents                               INTEGER,
        &numOfBCUSMEvents                              INTEGER,
        &numOfChargingEvents                           INTEGER,
        &numOfCSAs                                     INTEGER,
        &numOfCSs                                      INTEGER,
        &numOfExtensions                               INTEGER,
        &numOfGenericNumbers                           INTEGER,
        &numOfInServiceCompatibilityIndLength          INTEGER,
        &numOfLegs                                     INTEGER,
        &numOfMessageIDs                               INTEGER,
        &maxAmount                                     INTEGER,
        &maxInitialUnitIncrement                       INTEGER,
        &maxScalingFactor                              INTEGER,
        &maxSegmentsPerDataInterval                    INTEGER,
        &maxTimePerInterval                            INTEGER,
        &maxUnitsPerDataInterval                       INTEGER,
        &maxUnitsPerInterval                                INTEGER,
        &ub-maxUserCredit                                   INTEGER,
        &ub-nbCall                                          INTEGER
}
WITH SYNTAX
{
        MINIMUM-FOR-ACH-BILLING-CHARGING        &minAChBillingChargingLength
        MAXIMUM-FOR-ACH-BILLING-CHARGING        &maxAChBillingChargingLength
        MINIMUM-FOR-ATTRIBUTES                  &minAttributesLength
        MAXIMUM-FOR-ATTRIBUTES                  &maxAttributesLength
        MAXIMUM-FOR-BACKWARD-GVNS               &minBackwardGVNSLength
        MAXIMUM-FOR-BACKWARD-GVNS               &maxBackwardGVNSLength
        MAXIMUM-FOR-BEARER-CAPABILITY           &maxBearerCapabilityLength
        MINIMUM-FOR-CALLED-PARTY-NUMBER         &minCalledPartyNumberLength
        MAXIMUM-FOR-CALLED-PARTY-NUMBER         &maxCalledPartyNumberLength
        MINIMUM-FOR-CALLING-PARTY-NUMBER        &minCallingPartyNumberLength
        MAXIMUM-FOR-CALLING-PARTY-NUMBER        &maxCallingPartyNumberLength
        MINIMUM-FOR-CALL-RESULT                 &minCallResultLength
        MAXIMUM-FOR-CALL-RESULT                 &maxCallResultLength
        MAXIMUM-FOR-CAUSE                       &maxCauseLength
        MINIMUM-FOR-DIGITS                      &minDigitsLength
        MAXIMUM-FOR-DIGITS                      &maxDigitsLength
        MINIMUM-FOR-DISPLAY                     &minDisplayInformationLength
        MAXIMUM-FOR-DISPLAY                     &maxDisplayInformationLength
        MINIMUM-FOR-EVENT-SPECIFIC-CHARGING  &minEventSpecificInformationChargingLength
        MAXIMUM-FOR-EVENT-SPECIFIC-CHARGING &maxEventSpecificInformationChargingLength
        MINIMUM-FOR-EVENT-TYPE-CHARGING         &minEventTypeChargingLength
        MAXIMUM-FOR-EVENT-TYPE-CHARGING         &maxEventTypeChargingLength
        MINIMUM-FOR-FCI-BILLING-CHARGING        &minFCIBillingChargingLength
        MAXIMUM-FOR-FCI-BILLING-CHARGING        &maxFCIBillingChargingLength
        MINIMUM-FOR-FORWARD-GVNS                &minForwardGVNSLength
        MAXIMUM-FOR-FORWARD-GVNS                &maxForwardGVNSLength
        MINIMUM-FOR-GENERIC-NAME                &minGenericNameLength
        MAXIMUM-FOR-GENERIC-NAME                &maxGenericNameLength
        MINIMUM-FOR-GENERIC-NUMBER              &minGenericNumberLength
        MAXIMUM-FOR-GENERIC-NUMBER              &maxGenericNumberLength
        MAXIMUM-FOR-INITIAL-TIME-INTERVAL       &maxInitialTimeInterval
        MAXIMUM-FOR-IN-SERVICE-COMPATIBILITY&maxINServiceCompatibilityIndLength
        MINIMUM-FOR-IP-AVAILABLE                &minIPAvailableLength
        MAXIMUM-FOR-IP-AVAILABLE                &maxIPAvailableLength
        MINIMUM-FOR-IP-SSP-CAPABILITIES         &minIPSSPCapabilitiesLength
        MAXIMUM-FOR-IP-SSP-CAPABILITIES         &maxIPSSPCapabilitiesLength
        MINIMUM-FOR-LOCATION-NUMBER             &minLocationNumberLength
        MAXIMUM-FOR-LOCATION-NUMBER             &maxLocationNumberLength
        MINIMUM-FOR-MAIL-BOX-ID                 &minMailBoxIDLength
        MAXIMUM-FOR-MAIL-BOX-ID                 &maxMailBoxIDLength
        MINIMUM-FOR-MESSAGE-CONTENT             &minMessageContentLength
        MAXIMUM-FOR-MESSAGE-CONTENT             &maxMessageContentLength
        MINIMUM-FOR-MID-CALL-CONTROL-INFO       &minMidCallControlInfoNum
        MAXIMUM-FOR-MID-CALL-CONTROL-INFO       &maxMidCallControlInfoNum
        MINIMUM-FOR-ORIGINAL-CALLED-PARTY-ID&minOriginalCalledPartyIDLength
        MAXIMUM-FOR-ORIGINAL-CALLED-PARTY-ID &maxOriginalCalledPartyIDLength
        MINIMUM-FOR-REASON                      &minReasonLength
        MAXIMUM-FOR-REASON                      &maxReasonLength
        MINIMUM-FOR-RECEIVED-INFORMATION        &minReceivedInformationLength
        MAXIMUM-FOR-RECEIVED-INFORMATION        &maxReceivedInformationLength
        MAXIMUM-FOR-RECORDED-MESSAGE-UNITS &maxRecordedMessageUnits
        MAXIMUM-FOR-RECORDING-TIME              &maxRecordingTime
        MINIMUM-FOR-REDIRECTING-ID              &minRedirectingPartyIDLength
        MAXIMUM-FOR-REDIRECTING-ID              &maxRedirectingPartyIDLength
        MINIMUM-FOR-REQUESTED-UTSI-NUM          &minRequestedUTSINum
        MAXIMUM-FOR-REQUESTED-UTSI-NUM           &maxRequestedUTSINum
        MINIMUM-FOR-ROUTE-LIST                   &minRouteListLength
        MAXIMUM-FOR-ROUTE-LIST                   &maxRouteListLength
        MINIMUM-FOR-SCF-ID                       &minScfIDLength
        MAXIMUM-FOR-SCF-ID                       &maxScfIDLength
        MINIMUM-FOR-SCF-ADDRESS                  &minScfAddressLength
        MAXIMUM-FOR-SCF-ADDRESS                  &maxScfAddressLength
        MINIMUM-FOR-SCI-BILLING-CHARGING         &minSCIBillingChargingLength
        MAXIMUM-FOR-SCI-BILLING-CHARGING         &maxSCIBillingChargingLength
        MINIMUM-FOR-SII                          &minServiceInteractionIndicatorsLength
        MAXIMUM-FOR-SII                          &maxServiceInteractionIndicatorsLength
        MINIMUM-FOR-SF-BILLING-CHARGING          &minSFBillingChargingLength
        MAXIMUM-FOR-SF-BILLING-CHARGING          &maxSFBillingChargingLength
        MINIMUM-FOR-USI-INFORMATION              &minUSIInformationLength
        MAXIMUM-FOR-USI-INFORMATION              &maxUSIInformationLength
        MINIMUM-FOR-USI-SERVICE-INDICATOR        &minUSIServiceIndicatorLength
        MAXIMUM-FOR-USI-SERVICE-INDICATOR        &maxUSIServiceIndicatorLength
        NUM-OF-BCSM-EVENT                        &numOfBCSMEvents
        NUM-OF-BCUSM-EVENT                       &numOfBCUSMEvents
        NUM-OF-CHARGING-EVENT                    &numOfChargingEvents
        NUM-OF-CSAS                              &numOfCSAs
        NUM-OF-CSS                               &numOfCSs
        NUM-OF-EXTENSIONS                        &numOfExtensions
        NUM-OF-GENERIC-NUMBERS                   &numOfGenericNumbers
        NUM-OF-IN-SERVICE-COMPATIBILITY-ID       &numOfInServiceCompatibilityIndLength
        NUM-OF-LEGS                              &numOfLegs
        NUM-OF-MESSAGE-IDS                       &numOfMessageIDs
        MAXIMUM-FOR-AMOUNT                       &maxAmount
        MAXIMUM-FOR-INITIAL-UNIT-INCREMENT       &maxInitialUnitIncrement
        MAXIMUM-FOR-SCALING-FACTOR               &maxScalingFactor
        MAXIMUM-FOR-SEGMENTS-PER-DATA-INTERVAL   &maxSegmentsPerDataInterval
        MAXIMUM-FOR-TIME-PER-INTERVAL            &maxTimePerInterval
        MAXIMUM-FOR-UNITS-PER-DATA-INTERVAL      &maxUnitsPerDataInterval
        MAXIMUM-FOR-UNITS-PER-INTERVAL           &maxUnitsPerInterval
        MAXIMUM-FOR-UB-USER-CREDIT               &ub-maxUserCredit
        MAXIMUM-FOR-UB-NB-CALL                   &ub-nbCall
}

-- The following instance of the parameter bound is just an example
networkSpecificBoundSet  PARAMETERS-BOUND ::=
{
        MINIMUM-FOR-ACH-BILLING-CHARGING             1        -- example value
        MAXIMUM-FOR-ACH-BILLING-CHARGING             5        -- example value
        MINIMUM-FOR-ATTRIBUTES                       1        -- example value
        MAXIMUM-FOR-ATTRIBUTES                       5        -- example value
        MAXIMUM-FOR-BACKWARD-GVNS                    1        -- example value
        MAXIMUM-FOR-BACKWARD-GVNS                    5        -- example value
        MAXIMUM-FOR-BEARER-CAPABILITY                5        -- example value
        MINIMUM-FOR-CALLED-PARTY-NUMBER              1        -- example value
        MAXIMUM-FOR-CALLED-PARTY-NUMBER              5        -- example value
        MINIMUM-FOR-CALLING-PARTY-NUMBER             1        -- example value
        MAXIMUM-FOR-CALLING-PARTY-NUMBER             5        -- example value
        MINIMUM-FOR-CALL-RESULT                      1        -- example value
        MAXIMUM-FOR-CALL-RESULT                      5        -- example value
        MAXIMUM-FOR-CAUSE                            4        -- example value
        MINIMUM-FOR-DIGITS                           1        -- example value
        MAXIMUM-FOR-DIGITS                           5        -- example value
        MINIMUM-FOR-DISPLAY                          1        -- example value
        MAXIMUM-FOR-DISPLAY                          5        -- example value
        MINIMUM-FOR-EVENT-SPECIFIC-CHARGING          1        -- example value
        MAXIMUM-FOR-EVENT-SPECIFIC-CHARGING          5        -- example value
```

MINIMUM-FOR-EVENT-TYPE-CHARGING	**1**	-- *example value*
MAXIMUM-FOR-EVENT-TYPE-CHARGING	**5**	-- *example value*
MINIMUM-FOR-FCI-BILLING-CHARGING	**1**	-- *example value*
MAXIMUM-FOR-FCI-BILLING-CHARGING	**5**	-- *example value*
MINIMUM-FOR-FORWARD-GVNS	**1**	-- *example value*
MAXIMUM-FOR-FORWARD-GVNS	**5**	-- *example value*
MINIMUM-FOR-GENERIC-NAME	**1**	-- *example value*
MAXIMUM-FOR-GENERIC-NAME	**5**	-- *example value*
MINIMUM-FOR-GENERIC-NUMBER	**1**	-- *example value*
MAXIMUM-FOR-GENERIC-NUMBER	**5**	-- *example value*
MAXIMUM-FOR-INITIAL-TIME-INTERVAL	**5**	-- *example value*
MAXIMUM-FOR-IN-SERVICE-COMPATIBILITY	**5**	-- *example value*
MINIMUM-FOR-IP-AVAILABLE	**1**	-- *example value*
MAXIMUM-FOR-IP-AVAILABLE	**5**	-- *example value*
MINIMUM-FOR-IP-SSP-CAPABILITIES	**1**	-- *example value*
MAXIMUM-FOR-IP-SSP-CAPABILITIES	**5**	-- *example value*
MINIMUM-FOR-LOCATION-NUMBER	**1**	-- *example value*
MAXIMUM-FOR-LOCATION-NUMBER	**5**	-- *example value*
MINIMUM-FOR-MAIL-BOX-ID	**1**	-- *example value*
MAXIMUM-FOR-MAIL-BOX-ID	**5**	-- *example value*
MINIMUM-FOR-MESSAGE-CONTENT	**1**	-- *example value*
MAXIMUM-FOR-MESSAGE-CONTENT	**5**	-- *example value*
MINIMUM-FOR-MID-CALL-CONTROL-INFO	**1**	-- *example value*
MAXIMUM-FOR-MID-CALL-CONTROL-INFO	**5**	-- *example value*
MINIMUM-FOR-ORIGINAL-CALLED-PARTY-ID	**1**	-- *example value*
MAXIMUM-FOR-ORIGINAL-CALLED-PARTY-ID	**5**	-- *example value*
MINIMUM-FOR-REASON	**1**	-- *example value*
MAXIMUM-FOR-REASON	**5**	-- *example value*
MINIMUM-FOR-RECEIVED-INFORMATION	**1**	-- *example value*
MAXIMUM-FOR-RECEIVED-INFORMATION	**5**	-- *example value*
MAXIMUM-FOR-RECORDED-MESSAGE-UNITS	**5**	-- *example value*
MAXIMUM-FOR-RECORDING-TIME	**5**	-- *example value*
MINIMUM-FOR-REDIRECTING-ID	**1**	-- *example value*
MAXIMUM-FOR-REDIRECTING-ID	**5**	-- *example value*
MINIMUM-FOR-REQUESTED-UTSI-NUM	**1**	-- *example value*
MAXIMUM-FOR-REQUESTED-UTSI-NUM	**5**	-- *example value*
MINIMUM-FOR-ROUTE-LIST	**1**	-- *example value*
MAXIMUM-FOR-ROUTE-LIST	**5**	-- *example value*
MINIMUM-FOR-SCF-ID	**1**	-- *example value*
MAXIMUM-FOR-SCF-ID	**5**	-- *example value*
MINIMUM-FOR-SCF-ADDRESS	**1**	-- *example value*
MAXIMUM-FOR-SCF-ADDRESS	**5**	-- *example value*
MINIMUM-FOR-SCI-BILLING-CHARGING	**1**	-- *example value*
MAXIMUM-FOR-SCI-BILLING-CHARGING	**5**	-- *example value*
MINIMUM-FOR-SII	**1**	-- *example value*
MAXIMUM-FOR-SII	**5**	-- *example value*
MINIMUM-FOR-SF-BILLING-CHARGING	**1**	-- *example value*
MAXIMUM-FOR-SF-BILLING-CHARGING	**5**	-- *example value*
MINIMUM-FOR-USI-INFORMATION	**1**	-- *example value*
MAXIMUM-FOR-USI-INFORMATION	**5**	-- *example value*
MINIMUM-FOR-USI-SERVICE-INDICATOR	**1**	-- *example value*
MAXIMUM-FOR-USI-SERVICE-INDICATOR	**5**	-- *example value*
NUM-OF-BCSM-EVENT	**4**	-- *example value*
NUM-OF-BCUSM-EVENT	**4**	-- *example value*
NUM-OF-CHARGING-EVENT	**4**	-- *example value*
NUM-OF-CSAS	**2**	-- *example value*
NUM-OF-CSS	**2**	-- *example value*
NUM-OF-EXTENSIONS	**1**	-- *example value*
NUM-OF-GENERIC-NUMBERS	**2**	-- *example value*

```
        NUM-OF-IN-SERVICE-COMPATIBILITY-ID          2       -- example value
        NUM-OF-LEGS                                 2       -- example value
        NUM-OF-MESSAGE-IDS                          2       -- example value
        NUM-OF-RECORDED-MESSAGE-IDS                 2       -- example value
        MAXIMUM-FOR-AMOUNT                          2       -- example value
        MAXIMUM-FOR-INITIAL-UNIT-INCREMENT          2       -- example value
        MAXIMUM-FOR-SCALING-FACTOR                  2       -- example value
        MAXIMUM-FOR-SEGMENTS-PER-DATA-INTERVAL      5       -- example value
        MAXIMUM-FOR-TIME-PER-INTERVAL               5       -- example value
        MAXIMUM-FOR-UNITS-PER-DATA-INTERVAL         5       -- example value
        MAXIMUM-FOR-UNITS-PER-INTERVAL              5       -- example value
        MAXIMUM-FOR-UB-USER-CREDIT                  5       -- example value
        MAXIMUM-FOR-UB-NB-CALL                      5       -- example value
}
END
```

4.6 Object identifiers

```
IN-CS2-object-identifiers {itu-t recommendation q 1228 modules(0) in-cs2-object-identifiers(17) version1(0)}
DEFINITIONS ::=

BEGIN
-- This module assigns object identifiers for Modules, Packages, Contracts and Application Context
-- for IN CS-2

-- For Modules from TCAP, ROS
tc-Messages                     OBJECT IDENTIFIER ::=
        {ccitt recommendation q 773 modules(2) messages(1) version3(3)}
tc-NotationExtensions           OBJECT IDENTIFIER ::=
        {ccitt recommendation q 775 modules(2)  notation-extension (4) version1(1)}
ros-InformationObjects          OBJECT IDENTIFIER ::=
        {joint-iso-ccitt remote-operations(4) informationObjects(5) version1(0)}
ros-genericPDUs                 OBJECT IDENTIFIER ::=
        {joint-iso-ccitt remote-operations(4) generic-ROS-PDUs(6) version1(0)}
ros-UsefulDefinitions           OBJECT IDENTIFIER ::=
        {joint-iso-ccitt remote-operations(4) useful-definitions(7) version1(0)}
sese-APDUs                      OBJECT IDENTIFIER ::=
        {joint-iso-ccitt genericULS(20) modules(1) seseAPDUs(6) }
guls-Notation                   OBJECT IDENTIFIER ::=
        {joint-iso-ccitt genericULS (20) modules (1) notation (1)}
guls-SecurityTransformations    OBJECT IDENTIFIER ::=
        {joint-iso-itu-t genericULS (20) modules (1) gulsSecurityTransformations (3) }
ds-UsefulDefinitions            OBJECT IDENTIFIER ::=
        {joint-iso-ccitt ds(5) module(1) usefulDefinitions(0) 3}
spkmGssTokens                   OBJECT IDENTIFIER ::=
        { iso(1) identified-organization(3) dod(6) internet(1) security(5) mechanisms(5) spkm(1)
spkmGssTokens(10)}

-- For IN-CS-1 Modules
contexts                        OBJECT IDENTIFIER ::=
        {itu-t recommendation q 1218 modules (0) contexts (8) selectedContexts (1) version (1)}

-- For IN CS-2 Modules
datatypes                       OBJECT IDENTIFIER ::=
        {itu-t recommendation q 1228 modules(0) in-cs2-datatypes (0) version1(0)}
errortypes                      OBJECT IDENTIFIER ::=
        {itu-t recommendation q 1228 modules(0) in-cs2-errortypes (1) version1(0)}
operationcodes                  OBJECT IDENTIFIER ::=
        {itu-t recommendation q 1228 modules(0) in-cs2-operationcodes (2) version1(0)}
errorcodes                      OBJECT IDENTIFIER ::=
        {itu-t recommendation q 1228 modules(0) in-cs2-errorcodes (3) version1(0)}
```

```
classes                         OBJECT IDENTIFIER ::=
        {itu-t recommendation q 1228 modules(0) in-cs2-classes (4) version1(0)}
ssf-scf-Operations              OBJECT IDENTIFIER ::=
        {itu-t recommendation q 1228 modules(0) in-cs2-ssf-scf-ops-args (5) version1(0)}
ssf-scf-Protocol                OBJECT IDENTIFIER ::=
        {itu-t recommendation q 1228 modules(0) in-cs2-ssf-scf-pkgs-contracts-acs (6) version1(0)}
scf-srf-Operations              OBJECT IDENTIFIER ::=
        {itu-t recommendation q 1228 modules(0) in-cs2-scf-srf-ops-args (7) version1(0)}
scf-srf-Protocol                OBJECT IDENTIFIER ::=
        {itu-t recommendation q 1228 modules(0) in-cs2-scf-srf-pkgs-contracts-acs(8) version1(0)}
sdf-InformationFramework        OBJECT IDENTIFIER ::=
        {itu-t recommendation q 1228 module(0) sdfInformationFramework(9) version1(0) }
sdf-BasicAccessControl          OBJECT IDENTIFIER ::=
        {itu-t recommendation q 1228 module(0) sdfBasicAccessControl(10) version1(0) }
scf-sdf-Operations              OBJECT IDENTIFIER ::=
        {itu-t recommendation q 1228 module(0) scf-sdf-operations(11) version1(0) }
scf-sdf-Protocol                OBJECT IDENTIFIER ::=
        {itu-t recommendation q 1218 modules(0) in-scf-sdf-protocol(12) version1(0)}
scf-scf-Operations              OBJECT IDENTIFIER ::=
        {itu-t recommendation q 1228 modules(0) in-cs2-scf-scf-ops-args (13) version1(0)}
scf-scf-Protocol                OBJECT IDENTIFIER ::=
        {itu-t recommendation q 1228 modules(0) in-cs2-scf-scf-pkgs-contracts-acs (14) version1(0)}
scf-cusf-Operations             OBJECT IDENTIFIER ::=
        {itu-t recommendation q 1228 modules(0) in-cs2-scf-cusf-ops-args (15) version1(0)}
scf-cusf-Protocol               OBJECT IDENTIFIER ::=
        {itu-t recommendation q 1228 modules(0) in-cs2-scf-cusf-pkgs-contracts-acs (16) version1(0)}
object-identifiers              OBJECT IDENTIFIER ::=
        {itu-t recommendation q 1228 modules(0) in-cs2-object-identifiers(17) version1(0)}
sdf-sdf-Protocol                OBJECT IDENTIFIER ::=
        {itu-t recommendation q 1228 module(0) in-cs2-sdf-sdf-Protocol(18) version1(0) }

id-cs2          OBJECT IDENTIFIER ::= {itu-t recommendation q 1228 cs2 (2)}

id-ac                                   OBJECT IDENTIFIER ::= {id-cs2 ac(3)}
id-as                                   OBJECT IDENTIFIER ::= {id-cs2 as(5)}
id-rosObject                            OBJECT IDENTIFIER ::= {id-cs2 rosObject(25)}
id-contract                             OBJECT IDENTIFIER ::= {id-cs2 contract(26)}
id-package                              OBJECT IDENTIFIER ::= {id-cs2 package(27)}

-- for ac, as, rosObject, contract and package, the values are identical to Q.1218

id-package-scf-scfConnection            OBJECT IDENTIFIER ::= {id-package 46}
id-package-dsspConnection               OBJECT IDENTIFIER ::= {id-package 47}

-- ROS Objects

id-rosObject-scf                        OBJECT IDENTIFIER ::= {id-rosObject 4}
id-rosObject-ssf                        OBJECT IDENTIFIER ::= {id-rosObject 5}
id-rosObject-srf                        OBJECT IDENTIFIER ::= {id-rosObject 6}
id-rosObject-cusf                       OBJECT IDENTIFIER ::= {id-rosObject 7}
id-rosObject-dssp                       OBJECT IDENTIFIER ::= {id-rosObject 8}
id-rosObject-sdf                        OBJECT IDENTIFIER ::= {id-rosObject 9}

id-rosObject-dua                        OBJECT IDENTIFIER ::= {id-rosObject 1}
id-rosObject-directory                  OBJECT IDENTIFIER ::= {id-rosObject 2}
id-rosObject-dapDSA                     OBJECT IDENTIFIER ::= {id-rosObject 3}

id-rosObject-dspDSA                     OBJECT IDENTIFIER ::= { id-rosObject 10 }
id-rosObject-initiatingConsumerDSA      OBJECT IDENTIFIER ::= { id-rosObject 11 }
id-rosObject-respondingSupplierDSA      OBJECT IDENTIFIER ::= { id-rosObject 12 }
```

```
id-rosObject-respondingConsumerDSA            OBJECT IDENTIFIER ::= { id-rosObject 13 }
id-rosObject-initiatingSupplierDSA            OBJECT IDENTIFIER ::= { id-rosObject 14 }

-- ssf/scf Application Contexts

id-ac-cs2-ssf-scfGenericAC                    OBJECT IDENTIFIER ::= {id-ac 4}
id-ac-cs2-ssf-scfDPSpecificAC                 OBJECT IDENTIFIER ::= {id-ac 5}
id-ac-cs2-ssf-scfAssistHandoffAC              OBJECT IDENTIFIER ::= {id-ac 6}
id-ac-cs2-ssf-scfServiceManagementAC          OBJECT IDENTIFIER ::= {id-ac 7}
id-ac-cs2-scf-ssfGenericAC                    OBJECT IDENTIFIER ::= {id-ac 8}
id-ac-cs2-scf-ssfDPSpecificAC                 OBJECT IDENTIFIER ::= {id-ac 9}
id-ac-cs2-scf-ssfTrafficManagementAC          OBJECT IDENTIFIER ::= {id-ac 10}
id-ac-cs2-scf-ssfServiceManagementAC          OBJECT IDENTIFIER ::= {id-ac 11}
id-ac-cs2-scf-ssfStatusReportingAC            OBJECT IDENTIFIER ::= {id-ac 12}
id-ac-cs2-scf-ssfTriggerManagementAC          OBJECT IDENTIFIER ::= {id-ac 13}

-- srf/scf Application Context
id-ac-srf-scf                                 OBJECT IDENTIFIER ::= {id-ac 14}

-- SCF-SDF Application Contexts
id-ac-indirectoryAccessAC                     OBJECT IDENTIFIER ::= {id-ac 1}
id-ac-indirectoryAccessWith3seAC              OBJECT IDENTIFIER ::= {id-ac 2}
id-ac-inExtendedDirectoryAccessAC             OBJECT IDENTIFIER ::= {id-ac 3}
id-ac-inExtendedDirectoryAccessWith3seAC      OBJECT IDENTIFIER ::= {id-ac 27}

-- SDF-SDF Application Contexts
id-ac-indirectorySystemAC                     OBJECT IDENTIFIER ::= { id-ac 15 }
id-ac-inShadowSupplierInitiatedAC             OBJECT IDENTIFIER ::= { id-ac 16 }
id-ac-inShadowConsumerInitiatedAC             OBJECT IDENTIFIER ::= { id-ac 17 }
id-ac-indirectorySystemWith3seAC              OBJECT IDENTIFIER ::= { id-ac 18 }
id-ac-inShadowSupplierInitiatedWith3seAC      OBJECT IDENTIFIER ::= { id-ac 19 }
id-ac-inShadowConsumerInitiatedWith3seAC      OBJECT IDENTIFIER ::= { id-ac 20 }

-- scf/scf Application Contexts
id-ac-scf-scfOperationsAC                     OBJECT IDENTIFIER ::= {id-ac 21}
id-ac-distributedSCFSystemAC                  OBJECT IDENTIFIER ::= {id-ac 22}
id-ac-scf-scfOperationsWith3seAC              OBJECT IDENTIFIER ::= {id-ac 23}
id-ac-distributedSCFSystemWith3seAC       OBJECT IDENTIFIER ::= {id-ac 24}

-- cusf/scf Application Contexts
id-ac-scf-cusf                                OBJECT IDENTIFIER ::= {id-ac 25}
id-ac-cusf-scf                                OBJECT IDENTIFIER ::= {id-ac 26}

-- ssf/scf Contracts
id-inCs2SsfToScfGeneric                       OBJECT IDENTIFIER ::= {id-contract 3}
id-inCs2SsfToScfDpSpecific                    OBJECT IDENTIFIER ::= {id-contract 4}
id-inCs2AssistHandoffSsfToScf                 OBJECT IDENTIFIER ::= {id-contract 5}
id-inCs2ScfToSsfGeneric                       OBJECT IDENTIFIER ::= {id-contract 6}
id-inCs2ScfToSsfDpSpecific                    OBJECT IDENTIFIER ::= {id-contract 7}
id-inCs2ScfToSsfTrafficManagement             OBJECT IDENTIFIER ::= {id-contract 8}
id-inCs2ScfToSsfServiceManagement             OBJECT IDENTIFIER ::= {id-contract 9}
id-inCs2SsfToScfServiceManagement             OBJECT IDENTIFIER ::= {id-contract 10}
id-inCs2ScfToSsfStatusReporting               OBJECT IDENTIFIER ::= {id-contract 11}
id-inCs2ScfToSsfTriggerManagement             OBJECT IDENTIFIER ::= {id-contract 12}

-- srf/scf Contracts
id-contract-srf-scf                           OBJECT IDENTIFIER ::= {id-contract 13}

-- SCF-SDF Contracts
id-contract-dap                               OBJECT IDENTIFIER ::= {id-contract 1}
id-contract-dapExecute                        OBJECT IDENTIFIER ::= {id-contract 2 }
```

```
-- SDF-SDF Contracts
id-contract-indsp                            OBJECT IDENTIFIER ::= { id-contract 14 }
id-contract-shadowConsumer                   OBJECT IDENTIFIER ::= { id-contract 15 }
id-contract-shadowSupplier                   OBJECT IDENTIFIER ::= { id-contract 17 }

-- scf/scf Contracts
id-contract-scf-scf                          OBJECT IDENTIFIER ::= {id-contract 18}
id-contract-dssp                             OBJECT IDENTIFIER ::= {id-contract 19}

-- cusf/scf Contracts
id-contract-scf-cusf                         OBJECT IDENTIFIER ::= {id-contract 20}
id-contract-cusf-scf                         OBJECT IDENTIFIER ::= {id-contract 21}

-- ssf/scf Operation Packages

id-package-scfActivation                     OBJECT IDENTIFIER ::= {id-package 11}
id-package-basicBCPDP                        OBJECT IDENTIFIER ::= {id-package 12}
id-package-advancedBCPDP                     OBJECT IDENTIFIER ::= {id-package 14}
id-package-srf-scfActivationOfAssist         OBJECT IDENTIFIER ::= {id-package 15}
id-package-assistConnectionEstablishment     OBJECT IDENTIFIER ::= {id-package 16}
id-package-genericDisconnectResource         OBJECT IDENTIFIER ::= {id-package 17}
id-package-nonAssistedConnectionEstablishment OBJECT IDENTIFIER ::= {id-package 18}
id-package-connect                           OBJECT IDENTIFIER ::= {id-package 19}
id-package-callHandling                      OBJECT IDENTIFIER ::= {id-package 20}
id-package-bcsmEventHandling                 OBJECT IDENTIFIER ::= {id-package 21}
id-package-dpSpecificEventHandling           OBJECT IDENTIFIER ::= {id-package 22}
id-package-chargingEventHandling             OBJECT IDENTIFIER ::= {id-package 23}
id-package-ssfCallProcessing                 OBJECT IDENTIFIER ::= {id-package 24}
id-package-scfCallInitiation                 OBJECT IDENTIFIER ::= {id-package 25}
id-package-timer                             OBJECT IDENTIFIER ::= {id-package 26}
id-package-billing                           OBJECT IDENTIFIER ::= {id-package 27}
id-package-charging                          OBJECT IDENTIFIER ::= {id-package 28}
id-package-trafficManagement                 OBJECT IDENTIFIER ::= {id-package 29}
id-package-serviceManagementActivate         OBJECT IDENTIFIER ::= {id-package 30}
id-package-serviceManagementResponse         OBJECT IDENTIFIER ::= {id-package 31}
id-package-callReport                        OBJECT IDENTIFIER ::= {id-package 32}
id-package-signallingControl                 OBJECT IDENTIFIER ::= {id-package 33}
id-package-activityTest                      OBJECT IDENTIFIER ::= {id-package 34}
id-package-statusReporting                   OBJECT IDENTIFIER ::= {id-package 35}
id-package-cancel                            OBJECT IDENTIFIER ::= {id-package 36}
id-package-cphResponse                       OBJECT IDENTIFIER ::= {id-package 37}
id-package-entityReleased                    OBJECT IDENTIFIER ::= {id-package 38}
id-package-triggerManagement                 OBJECT IDENTIFIER ::= {id-package 39}
id-package-uSIHandling                       OBJECT IDENTIFIER ::= {id-package 40}
id-package-facilityIEHandling                OBJECT IDENTIFIER ::= {id-package 41}

-- srf/scf Operation Packages
id-package-specializedResourceControl        OBJECT IDENTIFIER ::= { id-package 42}
id-package-srf-scfCancel                     OBJECT IDENTIFIER ::= { id-package 43}
id-package-messageControl                    OBJECT IDENTIFIER ::= { id-package 44}
id-package-scriptControl                     OBJECT IDENTIFIER ::= { id-package 45}

-- SCF-SDF Packages
id-package-search                            OBJECT IDENTIFIER ::= {id-package 2}
id-package-modify                            OBJECT IDENTIFIER ::= {id-package 3}
id-package-dapConnection                     OBJECT IDENTIFIER ::={id-package 10}
id-package-execute                           OBJECT IDENTIFIER ::={id-package 4 }
```

```
-- SDF-SDF Packages
id-package-dspConnection                 OBJECT IDENTIFIER ::= { id-package 47 }
id-package-inchainedModify               OBJECT IDENTIFIER ::= { id-package 48 }
id-package-inchainedSearch               OBJECT IDENTIFIER ::= { id-package 49 }
id-package-chainedExecute                OBJECT IDENTIFIER ::= { id-package 50 }
id-package-dispConnection                OBJECT IDENTIFIER ::= { id-package 51 }
id-package-shadowConsumer                OBJECT IDENTIFIER ::= { id-package 52 }
id-package-shadowSupplier                OBJECT IDENTIFIER ::= { id-package 53 }

-- scf/scf Operation Packages
id-package-handlingInformation           OBJECT IDENTIFIER ::= {id-package 54}
id-package-notification                  OBJECT IDENTIFIER ::= {id-package 55}
id-package-chargingInformation           OBJECT IDENTIFIER ::= {id-package 56}
id-package-userInformation               OBJECT IDENTIFIER ::= {id-package 57}
id-package-networkCapability             OBJECT IDENTIFIER ::= {id-package 58}
id-package-chainedSCFOperations          OBJECT IDENTIFIER ::= {id-package 59}

-- cusf/scf Operation Packages
id-package-emptyConnection               OBJECT IDENTIFIER ::= { id-package 60}
id-package-basic-cusf-scf                OBJECT IDENTIFIER ::= { id-package 61}
id-package-basic-scf-cusf                OBJECT IDENTIFIER ::= { id-package 62}


-- ssf/scf Abstract Syntaxes

id-as-ssf-scfGenericAS                   OBJECT IDENTIFIER ::= {id-as 4}
id-as-ssf-scfDpSpecificAS                OBJECT IDENTIFIER ::= {id-as 5}
id-as-assistHandoff-ssf-scfAS            OBJECT IDENTIFIER ::= {id-as 6}
id-as-scf-ssfGenericAS                   OBJECT IDENTIFIER ::= {id-as 7}
id-as-scf-ssfDpSpecificAS                OBJECT IDENTIFIER ::= {id-as 8}
id-as-scf-ssfTrafficManagementAS         OBJECT IDENTIFIER ::= {id-as 9}
id-as-scf-ssfServiceManagementAS         OBJECT IDENTIFIER ::= {id-as 10}
id-as-ssf-scfServiceManagementAS         OBJECT IDENTIFIER ::= {id-as 11}
id-as-scf-ssfStatusReportingAS           OBJECT IDENTIFIER ::= {id-as 12}
id-as-scf-ssfTriggerManagementAS         OBJECT IDENTIFIER ::= {id-as 13}


-- srf/scf Abstract Syntaxes
id-as-basic-srf-scf                      OBJECT IDENTIFIER ::= { id-as 14}
id-as-basic-scf-srf                      OBJECT IDENTIFIER ::= { id-as 15}

-- SCF-SDF Abstract Syntaxes
id-as-indirectoryOperationsAS            OBJECT IDENTIFIER ::= {id-as 1}
id-as-indirectoryBindingAS               OBJECT IDENTIFIER ::= {id-as 2}
id-as-inExtendedDirectoryOperationsAS    OBJECT IDENTIFIER ::= {id-as 3 }
id-as-inSESEAS                           OBJECT IDENTIFIER ::= {id-as 25 }

-- SDF-SDF Abstract Syntaxes
id-as-indirectorySystemAS                OBJECT IDENTIFIER ::= { id-as 16 }
id-as-indirectoryDSABindingAS            OBJECT IDENTIFIER ::= { id-as 17 }
id-as-indirectoryShadowAS                OBJECT IDENTIFIER ::= { id-as 18 }
id-as-indsaShadowBindingAS               OBJECT IDENTIFIER ::= { id-as 19 }

-- scf/scf Abstract Syntaxes
id-as-scf-scfOperationsAS                OBJECT IDENTIFIER ::= {id-as 20}
id-as-distributedSCFSystemAS             OBJECT IDENTIFIER ::= {id-as 21}
id-as-scf-scfBindingAS                   OBJECT IDENTIFIER ::= {id-as 22}
```

```
-- cusf/scf Abstract Syntaxes

id-as-basic-cusf-scf                OBJECT IDENTIFIER ::= { id-as 23}
id-as-basic-scf-cusf                OBJECT IDENTIFIER ::= { id-as 24}



-- Object Identifiers for SDF-SDF interface
-- useful definitions
in-ds        OBJECT IDENTIFIER ::= {itu-t recommendation q 1228 sdf-objects (10)}

id-avc                              OBJECT IDENTIFIER ::= {in-ds 29}
id-aca                              OBJECT IDENTIFIER ::= {in-ds 24}
id-soa                              OBJECT IDENTIFIER ::= {in-ds 21}

-- Object Identifiers for SDF-SDF interface

-- SDF Attributes
id-soa-methodRuleUse                OBJECT IDENTIFIER ::= {id-soa 1}
id-aca-prescriptiveACI              OBJECT IDENTIFIER ::= { id-aca 4 }
id-aca-entryACI                     OBJECT IDENTIFIER ::= { id-aca 5 }
id-aca-subentryACI                  OBJECT IDENTIFIER ::= { id-aca 6 }

-- SDF  Attribute Value Contexts
id-avc-assignment                   OBJECT IDENTIFIER ::= {id-avc 1}



END
```

5 SSF/SCF interface

5.1 Operations and arguments

```
IN-CS2-SSF-SCF-ops-args {itu-t recommendation q 1228 modules(0) in-cs2-ssf-scf-ops-args (5) version1(0)}

DEFINITIONS IMPLICIT TAGS ::=

BEGIN

IMPORTS
    errortypes, datatypes, operationcodes, classes, ros-InformationObjects
FROM IN-CS2-object-identifiers
    {itu-t recommendation q 1228 modules(0) in-cs2-object-identifiers(17) version1(0)}

    OPERATION

FROM Remote-Operations-Information-Objects ros-InformationObjects



    PARAMETERS-BOUND

FROM IN-CS2-classes classes

    opcode-activateServiceFiltering,
    opcode-activityTest,
    opcode-analysedInformation,
    opcode-analyseInformation,
    opcode-applyCharging,
    opcode-applyChargingReport,
    opcode-assistRequestInstructions,
    opcode-authorizeTermination,
    opcode-callGap,
    opcode-callInformationReport,
    opcode-callInformationRequest,
    opcode-cancel,
    opcode-cancelStatusReportRequest,
    opcode-collectedInformation,
    opcode-collectInformation,
    opcode-connect,
    opcode-connectToResource,
    opcode-continue,
    opcode-continueWithArgument,
    opcode-createCallSegmentAssociation,
    opcode-disconnectForwardConnection,
    opcode-dFCWithArgument,
    opcode-disconnectLeg,
    opcode-entityReleased,
    opcode-establishTemporaryConnection,
    opcode-eventNotificationCharging,
    opcode-eventReportBCSM,
    opcode-eventReportFacility,
    opcode-facilitySelectedAndAvailable,
    opcode-furnishChargingInformation,
    opcode-holdCallInNetwork,
    opcode-initialDP,
    opcode-initiateCallAttempt,
    opcode-manageTriggerData,
    opcode-mergeCallSegments,
    opcode-moveCallSegments,
    opcode-oAbandon,
    opcode-oAnswer,
    opcode-oCalledPartyBusy,
    opcode-oDisconnect,
    opcode-oMidCall,
    opcode-moveLeg,
    opcode-oNoAnswer,
    opcode-originationAttempt,
    opcode-originationAttemptAuthorized,
    opcode-oSuspended,
    opcode-reconnect,
    opcode-releaseCall,
    opcode-reportUTSI,
    opcode-requestCurrentStatusReport,
    opcode-requestEveryStatusChangeReport,
    opcode-requestFirstStatusMatchReport,
    opcode-requestNotificationChargingEvent,
    opcode-requestReportBCSMEvent,
    opcode-requestReportUTSI,
    opcode-requestReportFacilityEvent,
    opcode-resetTimer,
    opcode-routeSelectFailure,
    opcode-selectFacility,
    opcode-selectRoute,
    opcode-sendChargingInformation,
    opcode-sendFacilityInformation,
    opcode-sendSTUI,
    opcode-serviceFilteringResponse,
    opcode-splitLeg,
            opcode-statusReport,
            opcode-tAnswer,
            opcode-tBusy,
            opcode-tDisconnect,
            opcode-termAttemptAuthorized,
            opcode-terminationAttempt,
            opcode-tMidCall,
            opcode-tNoAnswer,
            opcode-tSuspended

    FROM IN-CS2-operationcodes operationcodes

            AccessCode {},
            ActionIndicator,
            ActionPerformed,
            AChBillingChargingCharacteristics {},
            AdditionalCallingPartyNumber {},
            AlertingPattern,
            ApplicationTimer,
            AssistingSSPIPRoutingAddress {},
            BackwardGVNS {},
            BCSMEvent  {},
            BearerCapability {},
            CalledPartyBusinessGroupID,
            CalledPartyNumber {},
            CalledPartySubaddress,
            CallingPartyBusinessGroupID,
            CallingPartyNumber {},
            CallingPartysCategory,
            CallingPartySubaddress,
            CallProcessingOperationCorrelationID,
            CallResult {},
            CallSegmentID {},
            Carrier,
            Cause {},
            CGEncountered,
            ChargeNumber {},
            ChargingEvent {},
            Component,
            ComponentCorrelationID,
            ComponentType,
            ControlType,
            CorrelationID {},
            CountersValue,
            CSAID {},
            CutAndPaste,
            DateAndTime,
            DestinationRoutingAddress {},
            Digits {},
            DisplayInformation {},
            DpSpecificCommonParameters {},
            Duration,
            EventSpecificInformationBCSM {},
            EventSpecificInformationCharging {},
            EventTypeBCSM,
            EventTypeCharging {},
            ExtensionField {},
            FacilityGroup,
            FacilityGroupMember,
            FCIBillingChargingCharacteristics {},
            FeatureCode {},
    FeatureRequestIndicator,
    FilteredCallTreatment {},
    FilteringCharacteristics,
    FilteringCriteria {},
    FilteringTimeOut,
    ForwardCallIndicators,
    ForwardGVNS {},
    ForwardingCondition,
    GapCriteria {},
    GapIndicators,
    GapTreatment {},
    GenericName {},
    GenericNumbers {},
    HighLayerCompatibility,
    HoldCause,
    initialCallSegment,
    INServiceCompatibilityIndication {},
    INServiceCompatibilityResponse,
    Integer4,
    InvokeID,
    IPAvailable {},
    IPRoutingAddress {},
    IPSSPCapabilities {},
    ISDNAccessRelatedInformation,
    LegID,
    leg1,
    LocationNumber {},
    MiscCallInfo,
    MonitorMode,
    NumberingPlan,
    OriginalCalledPartyID {},
    Reason {},
    RedirectingPartyID {},
    RedirectionInformation,
    RegistratorIdentifier,
    ReportCondition,
    RequestedInformationList {},
    RequestedInformationTypeList,
    RequestedUTSIList {},
    ResourceID {},
    ResourceStatus,
    ResponseCondition,
    RouteList {},
    ScfID {},
    SCIBillingChargingCharacteristics {},
    ServiceInteractionIndicators {},
    ServiceInteractionIndicatorsTwo,
    ServiceKey,
    ServiceProfileIdentifier,
    TerminalType,
    TimerID,
    TimerValue,
    TravellingClassMark {},
    TriggerDataIdentifier {},
    TriggerType,
    USIInformation {},
    USIServiceIndicator {}
```

```
FROM IN-CS2-datatypes datatypes

        cancelFailed,
        eTCFailed,
        improperCallerResponse,
        missingCustomerRecord,
        missingParameter,
        parameterOutOfRange,
        requestedInfoError,
        systemFailure,
        taskRefused,
        unavailableResource,
        unexpectedComponentSequence,
        unexpectedDataValue,
        unexpectedParameter,
        unknownLegID,
        unknownResource

FROM IN-CS2-errortypes errortypes
;

activateServiceFiltering {PARAMETERS-BOUND : bound} OPERATION ::= {
        ARGUMENT                ActivateServiceFilteringArg {bound}
        RETURN RESULT TRUE
        ERRORS                  {missingParameter |
                                parameterOutOfRange |
                                systemFailure |
                                taskRefused |
                                unexpectedComponentSequence |
                                unexpectedParameter
                                }
        CODE                    opcode-activateServiceFiltering
        }

-- Direction: SCF → SSF, Timer: T_asf
-- When receiving this operation, the SSF handles calls to destination in a specified manner
-- without  sending queries for every detected call. It is used for example for providing
-- televoting or mass calling services. Simple registration functionality (counters) and
-- announcement control may be  located at the SSF. The operation initializes the specified
-- counters in the SSF.

ActivateServiceFilteringArg {PARAMETERS-BOUND : bound} ::= SEQUENCE {
        filteredCallTreatment         [0] FilteredCallTreatment {bound},
        filteringCharacteristics      [1] FilteringCharacteristics,
        filteringTimeOut              [2] FilteringTimeOut ,
        filteringCriteria             [3] FilteringCriteria {bound},
        startTime                     [4] DateAndTime                  OPTIONAL,
        extensions                    [5] SEQUENCE SIZE(1..bound.&numOfExtensions) OF
                                          ExtensionField {bound}  OPTIONAL,
        ...
        }

activityTest OPERATION ::= {
        RETURN RESULT TRUE
        CODE                          opcode-activityTest
        }
-- Direction: SCF → SSF, Timer: T_at
-- This operation is used to check for the continued existence of a relationship between the SCF
-- and SSF. If the relationship is still in existence, then the SSF will respond. If no reply is
-- received, then the SCF will assume that the SSF has failed in some way and will take the
-- appropriate action.

analysedInformation {PARAMETERS-BOUND : bound} OPERATION ::= {
        ARGUMENT                    AnalysedInformationArg {bound}
        RETURN RESULT               FALSE
        ERRORS                      {missingCustomerRecord |
                                    missingParameter |
                                    parameterOutOfRange |
                                    systemFailure |
                                    taskRefused |
                                    unexpectedComponentSequence |
                                    unexpectedDataValue |
                                    unexpectedParameter}
        CODE                        opcode-analysedInformation
        }
-- Direction:  SSF → SCF, Timer: T_adi
-- This operation is used to indicate availability of routing address and call type. (DP  –
-- Analysed_Info).
-- For additional information on this operation and its use with open numbering plans, refer to
-- Rec.Q.1224.

AnalysedInformationArg {PARAMETERS-BOUND : bound}::= SEQUENCE {
        dpSpecificCommonParameters      [0] DpSpecificCommonParameters {bound},
        dialledDigits                   [1] CalledPartyNumber {bound}           OPTIONAL,
        callingPartyBusinessGroupID     [2] CallingPartyBusinessGroupID         OPTIONAL,
        callingPartySubaddress          [3] CallingPartySubaddress              OPTIONAL,
        callingFacilityGroup            [4] FacilityGroup                       OPTIONAL,
        callingFacilityGroupMember      [5] FacilityGroupMember                 OPTIONAL,
        originalCalledPartyID           [6] OriginalCalledPartyID {bound}       OPTIONAL,
        prefix                          [7] Digits {bound}                      OPTIONAL,
        redirectingPartyID              [8] RedirectingPartyID {bound}          OPTIONAL,
        redirectionInformation          [9] RedirectionInformation              OPTIONAL,
        routeList                       [10] RouteList {bound}                  OPTIONAL,
        travellingClassMark             [11] TravellingClassMark {bound}        OPTIONAL,
        extensions                      [12] SEQUENCE SIZE(1..bound.&numOfExtensions) OF
                                            ExtensionField {bound}              OPTIONAL,
        featureCode                     [13] FeatureCode {bound}                OPTIONAL,
        accessCode                      [14] AccessCode {bound}                 OPTIONAL,
        carrier                         [15] Carrier                            OPTIONAL,
        componentType                   [16] ComponentType                      OPTIONAL,
        component                       [17] Component                          OPTIONAL,
        componentCorrelationID          [18] ComponentCorrelationID             OPTIONAL,
        ...
        }

-- For the OPTIONAL parameters, refer to clause 17 for the trigger detection point processing rules
-- to specify when these parameters are included in the message.

analyseInformation {PARAMETERS-BOUND : bound} OPERATION ::= {
        ARGUMENT                        AnalyseInformationArg {bound}
        RETURN RESULT                   FALSE
        ERRORS                          {missingParameter |
                                        parameterOutOfRange |
                                        systemFailure |
                                        taskRefused |
                                        unexpectedComponentSequence |
                                        unexpectedDataValue |
                                        unexpectedParameter}
        CODE                            opcode-analyseInformation
        }
```

```
-- Direction: SCF → SSF, Timer: Tai
-- This operation is used to request the SSF to perform the originating basic call processing actions
-- to analyse destination information that is either collected from a calling party or provided by the SCF
-- (e.g. for number translation). This includes actions to validate the information according to an office
-- or customized dialling plan, and if valid, to determine call termination information, to include the called
-- party address, the type of call (e.g. intranetwork or internetwork), and carrier (if internetwork).
-- If the called party is not served by the SSF, the SSF also determines a route index based on the called
-- party address and class of service, where the route index points to a list of outgoing trunk groups.

AnalyseInformationArg {PARAMETERS-BOUND : bound} ::= SEQUENCE {
        destinationRoutingAddress          [0] DestinationRoutingAddress {bound},
        alertingPattern                    [1] AlertingPattern                     OPTIONAL,
        iSDNAccessRelatedInformation       [2] ISDNAccessRelatedInformation        OPTIONAL,
        originalCalledPartyID              [3] OriginalCalledPartyID {bound}       OPTIONAL,
        extensions                         [4] SEQUENCE SIZE(1..bound.&numOfExtensions)  OF
                                               ExtensionField {bound}              OPTIONAL,
        callingPartyNumber                 [5] CallingPartyNumber {bound}          OPTIONAL,
        callingPartysCategory              [6] CallingPartysCategory               OPTIONAL,
        calledPartyNumber                  [7] CalledPartyNumber {bound}           OPTIONAL,
        chargeNumber                       [8] ChargeNumber {bound}                OPTIONAL,
        travellingClassMark                [9] TravellingClassMark {bound}         OPTIONAL,
        carrier                            [10] Carrier                            OPTIONAL,
        serviceInteractionIndicators       [11] ServiceInteractionIndicators {bound}  OPTIONAL,
        iNServiceCompatibilityResponse     [12] INServiceCompatibilityResponse     OPTIONAL,
        forwardGVNS                        [13] ForwardGVNS {bound}                OPTIONAL,
        backwardGVNS                       [14] BackwardGVNS {bound}               OPTIONAL,
        serviceInteractionIndicatorsTwo    [15] ServiceInteractionIndicatorsTwo    OPTIONAL,
        correlationID                      [16] CorrelationID {bound}              OPTIONAL,
        scfID                              [17] ScfID {bound}                      OPTIONAL,
        callSegmentID                      [18] CallSegmentID {bound}              OPTIONAL,
        legToBeCreated                     [19] LegID                              OPTIONAL,
        ...
        }

applyCharging {PARAMETERS-BOUND : bound} OPERATION ::= {
        ARGUMENT                           ApplyChargingArg {bound}
        RETURN RESULT                      FALSE
        ERRORS                             {missingParameter |
                                           unexpectedComponentSequence |
                                           unexpectedParameter |
                                           unexpectedDataValue |
                                           parameterOutOfRange |
                                           systemFailure |
                                           taskRefused|
                                           unknownLegID}
        CODE                               opcode-applyCharging
        }
-- Direction: SCF → SSF, Timer: Tac
-- This operation is used for interacting from the SCF with the SSF charging mechanisms. The ApplyChargingReport
-- operation provides the feedback from the SSF to the SCF.

ApplyChargingArg {PARAMETERS-BOUND : bound} ::= SEQUENCE {
        aChBillingChargingCharacteristics  [0] AChBillingChargingCharacteristics {bound},
        partyToCharge                      [2] LegID                               OPTIONAL,
        extensions                         [3] SEQUENCE SIZE (1..bound.&numOfExtensions) OF
                                               ExtensionField {bound}              OPTIONAL,
        ...
        }
```

```
-- The partyToCharge parameter indicates the party in the call to which the ApplyCharging operation
-- should be applied. If it is not present, then it is applied to the A-party

applyChargingReport {PARAMETERS-BOUND : bound} OPERATION ::= {
        ARGUMENT                        ApplyChargingReportArg {bound}
        RETURN RESULT                   FALSE
        ERRORS                          {missingParameter |
                                        unexpectedComponentSequence |
                                        unexpectedParameter |
                                        unexpectedDataValue |
                                        parameterOutOfRange |
                                        systemFailure |
                                        taskRefused}
        CODE                            opcode-applyChargingReport
        }

-- Direction: SSF → SCF, Timer: Tacr
-- This operation is used by the SSF to report to the SCF the occurrence of a specific charging event
-- as requested by the SCF using the ApplyCharging operation

ApplyChargingReportArg {PARAMETERS-BOUND : bound} ::= CallResult {bound}
-- NOTE – When the SSF sends the ApplyChargingReport operation as the last event from the Call Segment, the
-- lastEventIndicator parameter such as the CallInformationReport operation is needed for indicating whether
-- the event is last to the SCF.  However, because there is no consideration for the parameter expansion in the
-- CS-1, this parameter cannot be added. There are two alternatives for the solution. One is to be included
-- into the CallResult parameter. And the other is to specify a new operation with this parameter. The latter is
-- ffs.


assistRequestInstructions {PARAMETERS-BOUND : bound} OPERATION ::= {
        ARGUMENT                        AssistRequestInstructionsArg {bound}
        RETURN RESULT                   FALSE
        ERRORS                          {missingCustomerRecord|
                                        missingParameter |
                                        systemFailure |
                                        taskRefused |
                                        unexpectedComponentSequence |
                                        unexpectedDataValue |
                                        unexpectedParameter}
        CODE                            opcode-assistRequestInstructions
        }
-- Direction: SSF → SCF or SRF → SCF, Timer: Tari
-- This operation is used when there is an assist or a hand-off procedure and may be sent by the SSF
-- or SRF to the SCF. This operation is sent by the assisting SSF to SCF, when the initiating SSF has
-- set up a connection to the SRF or to the assisting SSF as a result of receiving an EstablishTemporaryConnection
-- or Connect/SelectRoute operation (in the case of hand-off) from the SCF.
-- Refer to clause 17 for a description of the procedures associated with this operation.

AssistRequestInstructionsArg {PARAMETERS-BOUND : bound} ::= SEQUENCE {
        correlationID                   [0] CorrelationID {bound},
        iPAvailable                     [1] IPAvailable {bound}                 OPTIONAL,
        iPSSPCapabilities               [2] IPSSPCapabilities {bound}           OPTIONAL,
        extensions                      [3] SEQUENCE SIZE(1..bound.&numOfExtensions)  OF
                                            ExtensionField {bound}              OPTIONAL,
        ...
        }
-- OPTIONAL denotes network-operator specific use. The value of the correlationID may be the
-- Called Party Number supplied by the initiating SSF.
```

```
authorizeTermination {PARAMETERS-BOUND : bound} OPERATION ::= {
        ARGUMENT                            AuthorizeTerminationArg {bound}
        RETURN RESULT                       FALSE
        ERRORS                              {missingParameter |
                                            systemFailure |
                                            taskRefused |
                                            unexpectedComponentSequence |
                                            unexpectedDataValue |
                                            unexpectedParameter}
        CODE                                opcode-authorizeTermination
        }
```

-- Direction: SCF $\rightarrow$ SSF, Timer: T_{atr}
-- This operation is used to request the SSF to resume terminating call processing action at the
-- Authorize_Termination PIC of the call based on the information received from the SCF.
-- For additional information on this operation, refer to Rec. Q.1224.

```
AuthorizeTerminationArg {PARAMETERS-BOUND : bound} ::= SEQUENCE {
        alertingPattern                     [0] AlertingPattern                     OPTIONAL,
        callingPartyNumber                  [1] CallingPartyNumber { bound}         OPTIONAL,
        destinationNumberRoutingAddress     [2] CalledPartyNumber { bound}          OPTIONAL,
        displayInformation                  [3] DisplayInformation {bound}          OPTIONAL,
        iSDNAccessRelatedInformation        [4] ISDNAccessRelatedInformation        OPTIONAL,
        originalCalledPartyID               [5] OriginalCalledPartyID {bound}       OPTIONAL,
        travellingClassMark                 [6] TravellingClassMark {bound}         OPTIONAL,
        extensions                          [7] SEQUENCE SIZE(1..bound.&numOfExtensions)  OF
                                                ExtensionField {bound}              OPTIONAL,
        iNServiceCompatibilityResponse      [8] INServiceCompatibilityResponse      OPTIONAL,
        forwardGVNS                         [9] ForwardGVNS {bound}                 OPTIONAL,
        backwardGVNS                        [10] BackwardGVNS {bound}               OPTIONAL,
        legID                               [11] LegID                              OPTIONAL,
        ...
        }
```
-- OPTIONAL parameters are only provided if modifications are desired to basic call processing values.

```
callGap {PARAMETERS-BOUND : bound} OPERATION ::= {
        ARGUMENT                            CallGapArg {bound}
        RETURN RESULT                       FALSE
        ALWAYS RESPONDS                     FALSE
        CODE                                opcode-callGap
        }
```
-- Direction: SCF $\rightarrow$ SSF, Timer: T_{cg}
-- This operation is used to request the SSF to reduce the rate at which specific service requests are sent to
-- the SCF. Use of this operation by the SCF to gap queries and updates at the SDF is for further study.

```
CallGapArg {PARAMETERS-BOUND : bound} ::= SEQUENCE {
        gapCriteria                         [0] GapCriteria {bound},
        gapIndicators                       [1] GapIndicators,
        controlType                         [2] ControlType                 OPTIONAL,
        gapTreatment                        [3] GapTreatment {bound}        OPTIONAL,
        extensions                          [4] SEQUENCE SIZE(1..bound.&numOfExtensions)  OF
                                               ExtensionField {bound}       OPTIONAL,
        ...
        }
```
-- OPTIONAL denotes network-operator optional. If gapTreatment is not present, the SSF will use
-- a default treatment depending on network-operator implementation.

```
callInformationReport {PARAMETERS-BOUND : bound} OPERATION::= {
        ARGUMENT                        CallInformationReportArg { bound}
        RETURN RESULT                   FALSE
        ALWAYS RESPONDS                 FALSE
        CODE                            opcode-callInformationReport
        }

-- Direction: SSF → SCF, Timer: Tcirp
-- This operation is used to send specific call information for a single call to the SCF as requested by the SCF
-- in a previous CallInformationRequest.

CallInformationReportArg {PARAMETERS-BOUND : bound} ::= SEQUENCE {
        requestedInformationList        [0] RequestedInformationList {bound},
        correlationID                   [1] CorrelationID {bound}             OPTIONAL,
        extensions                      [2] SEQUENCE SIZE(1..bound.&numOfExtensions)   OF
                                            ExtensionField {bound}            OPTIONAL,
        legID                           [3] LegID                             OPTIONAL,
        lastEventIndicator              [4] BOOLEAN                           DEFAULT  FALSE,
        ...
        }
-- OPTIONAL denotes network-operator optional.
-- The lastEventIndicator parameter is set with 'TRUE' when the report is last in the Call Segment.
-- In the CS-1, the lastEventIndicator should not be sent, and the meaning of DEFAULT is not applied. The SCF
-- must decide whether the report is last without this parameter.

callInformationRequest {PARAMETERS-BOUND : bound} OPERATION ::= {
        ARGUMENT                        CallInformationRequestArg {bound}
        RETURN RESULT                   FALSE
        ERRORS                          {missingParameter |
                                        parameterOutOfRange |
                                        requestedInfoError |
                                        systemFailure |
                                        taskRefused |
                                        unexpectedComponentSequence |
                                        unexpectedDataValue |
                                        unexpectedParameter|
                                        unknownLegID}
        CODE                            opcode-callInformationRequest
        }

-- Direction: SCF → SSF, Timer: Tcirq
-- This operation is used to request the SSF to record specific information about a single call and report it to
-- the SCF (with a CallInformationReport operation).

CallInformationRequestArg {PARAMETERS-BOUND : bound} ::= SEQUENCE {
        requestedInformationTypeList    [0] RequestedInformationTypeList,
        correlationID                   [1] CorrelationID {bound}             OPTIONAL,
        extensions                      [2] SEQUENCE SIZE(1..bound.&numOfExtensions) OF
                                            ExtensionField {bound}            OPTIONAL,
        legID                           [3] LegID                             OPTIONAL,
        ...
        }
-- OPTIONAL denotes network-operator optional.
```

```
cancel {PARAMETERS-BOUND : bound} OPERATION ::= {
        ARGUMENT                        CancelArg {bound}
        RETURN RESULT                   FALSE
        ERRORS                          {cancelFailed |
                                        missingParameter |
                                        taskRefused}
        CODE                            opcode-cancel
        }
-- Direction: SCF → SSF, or SCF → SRF, Timer: Tcan
-- This operation cancels the correlated previous operation or all previous requests. The following operations can be
-- cancelled: PlayAnnouncement, PromptAndCollectUserInformation.

CancelArg {PARAMETERS-BOUND : bound} ::= CHOICE {
        invokeID                        [0] InvokeID,
        allRequests                     [1] NULL,
        callSegmentToCancel             [2] SEQUENCE {
                                        invokeID                    [0] InvokeID,
                                        callSegmentID               [1] CallSegmentID {bound}
                                        }
        }
-- The InvokeID has the same value as that which was used for the operation to be cancelled.

cancelStatusReportRequest {PARAMETERS-BOUND : bound} OPERATION::= {
        ARGUMENT                        CancelStatusReportRequestArg {bound}
        RETURN RESULT                   FALSE
        ERRORS                          {cancelFailed |
                                        missingParameter |
                                        taskRefused}
        CODE                            opcode-cancelStatusReportRequest
        }

-- Direction: SCF → SSF, Timer: Tcsr
-- This operation cancels the following processes: RequestFirstStatusMatchReport and
-- RequestEveryStatusChangeReport.

CancelStatusReportRequestArg {PARAMETERS-BOUND : bound} ::= SEQUENCE {
        resourceID                      [0] ResourceID {bound}              OPTIONAL,
        extensions                      [1] SEQUENCE SIZE(1..bound.&numOfExtensions)   OF
                                            ExtensionField {bound}          OPTIONAL,
        ...
        }

collectedInformation {PARAMETERS-BOUND : bound} OPERATION ::= {
        ARGUMENT                        CollectedInformationArg {bound}
        RETURN RESULT                   FALSE
        ERRORS                          {missingCustomerRecord |
                                        missingParameter |
                                        systemFailure |
                                        taskRefused |
                                        unexpectedComponentSequence |
                                        unexpectedDataValue |
                                        unexpectedParameter}
        CODE                            opcode-collectedInformation
        }
-- Direction: SSF → SCF, Timer: Tcdi
-- This operation is used to indicate availability of complete initial information package/dialling string from
-- originating party. (This event may have already occurred in the case of en bloc signalling, in which case
-- the waiting duration in this PIC is zero.) (DP  – Collected_Info). For additional information on this operation
-- and its use with open numbering plans, refer to Rec. Q.1224.
```

```
CollectedInformationArg {PARAMETERS-BOUND : bound} ::= SEQUENCE {
        dpSpecificCommonParameters          [0] DpSpecificCommonParameters {bound},
        dialledDigits                       [1] CalledPartyNumber {bound}               OPTIONAL,
        callingPartyBusinessGroupID         [2] CallingPartyBusinessGroupID             OPTIONAL,
        callingPartySubaddress              [3] CallingPartySubaddress                  OPTIONAL,
        callingFacilityGroup                [4] FacilityGroup                           OPTIONAL,
        callingFacilityGroupMember          [5] FacilityGroupMember                     OPTIONAL,
        originalCalledPartyID               [6] OriginalCalledPartyID { bound}          OPTIONAL,
        prefix                              [7] Digits { bound}                         OPTIONAL,
        redirectingPartyID                  [8] RedirectingPartyID { bound}             OPTIONAL,
        redirectionInformation              [9] RedirectionInformation                  OPTIONAL,
        travellingClassMark                 [10] TravellingClassMark { bound}           OPTIONAL,
        extensions                          [11] SEQUENCE SIZE(1..bound.&numOfExtensions) OF
                                                 ExtensionField {bound}                 OPTIONAL,
        featureCode                         [12] FeatureCode { bound}                   OPTIONAL,
        accessCode                          [13] AccessCode { bound}                    OPTIONAL,
        carrier                             [14] Carrier                                OPTIONAL,
        componentType                       [15] ComponentType                          OPTIONAL,
        component                           [16] Component                              OPTIONAL,
        componentCorrelationID              [17] ComponentCorrelationID                 OPTIONAL,
        ...
        }
```

-- For the OPTIONAL parameters, refer to clause 17 for the trigger detection point processing rules to specify
-- when these parameters are included in the message.

```
collectInformation {PARAMETERS-BOUND : bound} OPERATION::= {
        ARGUMENT                            CollectInformationArg { bound}
        RETURN RESULT                       FALSE
        ERRORS                              {missingParameter |
                                            parameterOutOfRange |
                                            systemFailure |
                                            taskRefused |
                                            unexpectedComponentSequence |
                                            unexpectedDataValue |
                                            unexpectedParameter}
        CODE                                opcode-collectInformation
        }
```

-- Direction: SCF → SSF, Timer: T_{ci}
-- This operation is used to request the SSF to perform the originating basic call processing actions to prompt
-- a calling party for destination information, then collect destination information according to a specified
-- numbering plan (e.g. for virtual private networks).

```
CollectInformationArg {PARAMETERS-BOUND : bound} ::= SEQUENCE {
        alertingPattern                     [0] AlertingPattern                         OPTIONAL,
        numberingPlan                       [1] NumberingPlan                   OPTIONAL,
        originalCalledPartyID               [2] OriginalCalledPartyID { bound}          OPTIONAL,
        travellingClassMark                 [3] TravellingClassMark { bound}            OPTIONAL,
        extensions                          [4] SEQUENCE SIZE(1..bound.&numOfExtensions)  OF
                                                ExtensionField {bound}                  OPTIONAL,
        callingPartyNumber                  [5] CallingPartyNumber { bound}             OPTIONAL,
        dialledDigits                       [6] CalledPartyNumber { bound}              OPTIONAL,
        serviceInteractionIndicators        [7] ServiceInteractionIndicators { bound}   OPTIONAL,
        iNServiceCompatibilityResponse      [8] INServiceCompatibilityResponse          OPTIONAL,
        forwardGVNS                         [9] ForwardGVNS { bound}                    OPTIONAL,
        backwardGVNS                        [10] BackwardGVNS { bound}                  OPTIONAL,
        serviceInteractionIndicatorsTwo     [11] ServiceInteractionIndicatorsTwo      OPTIONAL,
        callSegmentID                       [12] CallSegmentID {bound}                OPTIONAL,
        legToBeCreated                      [13] LegID                                OPTIONAL,
        ...
        }

connect {PARAMETERS-BOUND : bound} OPERATION::= {
        ARGUMENT                            ConnectArg {bound}
        RETURN RESULT                       FALSE
        ERRORS                              {missingParameter |
                                            parameterOutOfRange |
                                            systemFailure |
                                            taskRefused |
                                            unexpectedComponentSequence |
                                            unexpectedDataValue |
                                            unexpectedParameter}
        CODE                                opcode-connect
        }
```

-- Direction: SCF $\rightarrow$ SSF, Timer: T_{con}
-- This operation is used to request the SSF to perform the call processing actions to route or forward a call to
-- a specified destination. To do so, the SSF may or may not use destination information from the calling party
-- (e.g. dialled digits) and existing call set-up information (e.g. route index to a list of trunk groups), depending on
-- the information provided by the SCF.
--   – When address information is only included in the Connect operation, call processing resumes at PIC3 in
--   the O-BCSM.
--   – When address information and routing information is included, call processing resumes at PIC4.

```
ConnectArg {PARAMETERS-BOUND : bound} ::= SEQUENCE {
        destinationRoutingAddress           [0] DestinationRoutingAddress { bound},
        alertingPattern                     [1] AlertingPattern                       OPTIONAL,
        correlationID                       [2] CorrelationID { bound}                OPTIONAL,
        cutAndPaste                         [3] CutAndPaste                           OPTIONAL,
        forwardingCondition                 [4] ForwardingCondition                   OPTIONAL,
        iSDNAccessRelatedInformation        [5] ISDNAccessRelatedInformation          OPTIONAL,
        originalCalledPartyID               [6] OriginalCalledPartyID { bound}        OPTIONAL,
        routeList                           [7] RouteList { bound}                    OPTIONAL,
        scfID                               [8] ScfID { bound}                        OPTIONAL,
        travellingClassMark                 [9] TravellingClassMark { bound}          OPTIONAL,
        extensions                          [10] SEQUENCE SIZE(1..bound.&numOfExtensions) OF
                                                 ExtensionField {bound}               OPTIONAL,
        carrier                             [11] Carrier                              OPTIONAL,
        serviceInteractionIndicators        [26] ServiceInteractionIndicators { bound} OPTIONAL,
        callingPartyNumber                  [27] CallingPartyNumber { bound}          OPTIONAL,
        callingPartysCategory               [28] CallingPartysCategory                OPTIONAL,
        redirectingPartyID                  [29] RedirectingPartyID { bound}          OPTIONAL,
        redirectionInformation              [30] RedirectionInformation               OPTIONAL,
        displayInformation                  [12] DisplayInformation { bound}          OPTIONAL,
        forwardCallIndicators               [13] ForwardCallIndicators                OPTIONAL,
        genericNumbers                      [14] GenericNumbers { bound}              OPTIONAL,
        serviceInteractionIndicatorsTwo     [15] ServiceInteractionIndicatorsTwo      OPTIONAL,
        iNServiceCompatibilityResponse      [16] INServiceCompatibilityResponse       OPTIONAL,
        forwardGVNS                         [17] ForwardGVNS { bound}                 OPTIONAL,
        backwardGVNS                        [18] BackwardGVNS { bound}                OPTIONAL,
        chargeNumber                        [19] ChargeNumber { bound}                OPTIONAL,
        callSegmentID                       [20] CallSegmentID {bound}                OPTIONAL,
        legToBeCreated                      [21] LegID                                OPTIONAL,
        ...
        }
```

-- For alerting pattern, OPTIONAL denotes that this parameter only applies if SSF is the terminating local
-- exchange for the subscriber.

```
connectToResource {PARAMETERS-BOUND : bound} OPERATION::= {
        ARGUMENT                        ConnectToResourceArg { bound}
        RETURN RESULT                   FALSE
        ERRORS                          {missingParameter |
                                        systemFailure |
                                        taskRefused |
                                        unexpectedComponentSequence |
                                        unexpectedDataValue |
                                        unexpectedParameter|
                                        unknownLegID}
        CODE                            opcode-connectToResource
        }
```

-- Direction: SCF → SSF, Timer: T_{ctr}
-- This operation is used to connect a call from the SSP to the physical entity containing the SRF.
-- Refer to clause 17 for a description of the procedures associated with this operation.

```
ConnectToResourceArg {PARAMETERS-BOUND : bound} ::= SEQUENCE {
        resourceAddress                 CHOICE {
                ipRoutingAddress        [0] IPRoutingAddress { bound},
                legID                   [1] LegID,
                ipAddressAndLegID       [2] SEQUENCE {
                                         ipRoutingAddress         [0] IPRoutingAddress {bound},
                                         legID                    [1] LegID
                                         },
                none                    [3] NULL,
                callSegmentID           [5] CallSegmentID { bound} ,
                ipAddressAndCallSegment [6] SEQUENCE {
                                         ipRoutingAddress         [0] IPRoutingAddress {bound},
                                         callSegmentID            [1] CallSegmentID { bound}
                                         }
                },
        extensions                      [4] SEQUENCE SIZE(1..bound.&numOfExtensions)   OF
                                            ExtensionField {bound}                   OPTIONAL,
        serviceInteractionIndicators    [30] ServiceInteractionIndicators { bound}   OPTIONAL,
        serviceInteractionIndicatorsTwo [7] ServiceInteractionIndicatorsTwo          OPTIONAL,
        ...
        }

continue OPERATION::= {
        RETURN RESULT                   FALSE
        ALWAYS RESPONDS                 FALSE
        CODE                            opcode-continue
        }
```

-- Direction: SCF → SSF, Timer: T_{cue}
-- This operation is used to request the SSF to proceed with call processing at the DP at which it
-- previously suspended call processing to await SCF instructions (i.e. proceed to the next point
-- in call in the BCSM). The SSF continues call processing without substituting new data from SCF.
-- This operation is not valid for a single call segment CSA with more than 2 legs or a multi call segment CSA.

```
continueWithArgument {PARAMETERS-BOUND : bound} OPERATION::= {
        ARGUMENT                        ContinueWithArgumentArg { bound}
        RETURN RESULT                   FALSE
        ERRORS                          {missingParameter |
                                        unexpectedComponentSequence |
                                        unexpectedParameter |
                                        unexpectedDataValue
                                        }
        CODE                            opcode-continueWithArgument}
```

-- Direction: SCF → SSF, Timer: T_{cwa}
-- This operation is used to request the SSF to proceed with call processing at the DP where it previously
-- suspended call processing to await SCF instructions.
-- It is also used to provide additional service related information to a User (Called Party or Calling Party) whilst
-- the call processing proceeds.

```
ContinueWithArgumentArg {PARAMETERS-BOUND : bound} ::= SEQUENCE {
        legID                           [0] LegID                       DEFAULT
                                            sendingSideID:leg1,
        alertingPattern                 [1] AlertingPattern             OPTIONAL,
        genericName                     [2] GenericName { bound}        OPTIONAL,
        iNServiceCompatibilityResponse  [3] INServiceCompatibilityResponse  OPTIONAL,
        forwardGVNS                     [4] ForwardGVNS { bound}        OPTIONAL,
        backwardGVNS                    [5] BackwardGVNS { bound}       OPTIONAL,
        extensions                      [6] SEQUENCE SIZE (1..bound.&numOfExtensions) OF
                                            ExtensionField {bound}      OPTIONAL,
        serviceInteractionIndicatorsTwo [7] ServiceInteractionIndicatorsTwo  OPTIONAL,
        ...
        }

createCallSegmentAssociation {PARAMETERS-BOUND : bound} OPERATION ::= {
        ARGUMENT                        CreateCallSegmentAssociationArg{ bound}
        RESULT                          CreateCallSegmentAssociationResult { bound}
        ERRORS                          {missingParameter |
                                        systemFailure|
                                        taskRefused|
                                        unexpectedComponentSequence
                                        unexpectedDataValue |
                                        unexpectedParameter
                                        }
        CODE                            opcode-createCallSegmentAssociation
        }
```

-- Direction: SCF → SSF, Timer: T_{csa}
-- This operation is used to create a new CSA. The new CSA will not contain any Call Segments after creation.
-- The SSF is responsible for specifying a new CSA identifier for the created CSA which is unique within
-- the SSF.

```
CreateCallSegmentAssociationArg {PARAMETERS-BOUND : bound} ::= SEQUENCE {
        extensions                      [0] SEQUENCE SIZE {1..bound.&numOfExtensions) OF
                                            ExtensionField {bound} OPTIONAL,
        ...
        }
CreateCallSegmentAssociationResult {PARAMETERS-BOUND : bound} ::= SEQUENCE {
        newCallSegmentAssociation       [0] CSAID { bound},
        ...
        }
```

```
disconnectForwardConnection OPERATION ::= {
        RETURN RESULTFALSE               FALSE
        ERRORS                           {systemFailure |
                                         taskRefused |
                                         unexpectedComponentSequence }
        CODE                             opcode-disconnectForwardConnection
        }

-- Direction: SCF → SSF, Timer: Tdfc
-- This operation is used to disconnect a forward temporary connection or a connection to a resource.
-- Refer to clause 17 for a description of the procedures associated with this operation.
-- This operation is not valid for a single call segment CSA with more than 2 legs or a multi call segment CSA.

disconnectForwardConnectionWithArgument {PARAMETERS-BOUND : bound} OPERATION ::= {
        ARGUMENT                         DisconnectForwardConnectionWithArgumentArg { bound}
        RETURN RESULT                    FALSE
        ERRORS                           {missingParameter |
                                         {systemFailure |
                                         taskRefused |
                                         unexpectedComponentSequence }
                                         unexpectedDataValue |
                                         unexpectedParameter |
                                         unknownLegID}
        CODE                             opcode-dFCWithArgument
        }

-- Direction: SCF → SSF, Timer: Tdfcwa
-- This operation is used to disconnect a forward temporary connection or a connection to a resource.
-- Refer to clause 17 for a description of the procedures associated with this operation.

DisconnectForwardConnectionWithArgumentArg {PARAMETERS-BOUND : bound} ::= SEQUENCE {

        partyToDisconnect            CHOICE {
                                     legID                      [0] LegID,
                                     callSegmentID              [1] CallSegmentID { bound}
                                     },
        extensions                   [2] SEQUENCE SIZE (1..bound.&numOfExtensions)   OF
                                          ExtensionField {bound}             OPTIONAL,
        ...
        }


disconnectLeg {PARAMETERS-BOUND : bound} OPERATION ::= {
        ARGUMENT                     DisconnectLegArg { bound}
        RETURN RESULT TRUE
        ERRORS                       {missingParameter|
                                     systemFailure |
                                     taskRefused |
                                     unexpectedComponentSequence |
                                     unexpectedDataValue |
                                     unexpectedParameter|
                                     unknownLegID}
        CODE                         opcode-disconnectLeg
        }

-- Direction: SCF → SSF, Timer: Tdl
-- This operation  is issued by the SCF  to release a specific leg associated with the call and retain any
-- other legs not specified in the DisconnectLeg. Any leg may be disconnected, including the controlling
-- leg, without completely releasing all legs.
-- For additional information on this operation, refer to Rec. Q.1224.
```

```
DisconnectLegArg {PARAMETERS-BOUND : bound} ::= SEQUENCE {
        legToBeReleased                 [0] LegID,
        releaseCause                    [1] Cause { bound}                              OPTIONAL,
        extensions                      [2] SEQUENCE SIZE (1..bound.&numOfExtensions) OF
                                            ExtensionField {bound}                      OPTIONAL,
        ...
        }


entityReleased {PARAMETERS-BOUND : bound}  OPERATION ::= {
        ARGUMENT                        EntityReleasedArg { bound}
        RETURN RESULT                   FALSE
        ALWAYS RESPONDS                 FALSE
        CODE                            opcode-entityReleased
        }
```

-- Direction: SSF → SCF, Timer: T_{er}
-- This operation is used by SSF to inform the SCF of an error/exception

```
EntityReleasedArg {PARAMETERS-BOUND : bound}  ::= CHOICE {
        cSFailure                       [0] SEQUENCE{
                                        callSegmentID               [0] CallSegmentID { bound},
                                        reason                      [1] Reason { bound
        OPTIONAL,
                                        cause                       [2] Cause { bound}

        OPTIONAL                        },
                                        [1] SEQUENCE{
        bCSMFailure                     legID                       [0] LegID,
                                        reason                      [1] Reason { bound}
        OPTIONAL,
                                        cause                       [2] Cause { bound}
        OPTIONAL
                                        }
        }
establishTemporaryConnection {PARAMETERS-BOUND : bound} OPERATION ::= {
        ARGUMENT                        EstablishTemporaryConnectionArg { bound}
        RETURN RESULTFALSE              FALSE
        ERRORS                          {eTCFailed |
                                        missingParameter |
                                        systemFailure |
                                        taskRefused |
                                        unexpectedComponentSequence |
                                        unexpectedDataValue |
                                        unexpectedParameter|
                                        unknownLegID}
        CODE                            opcode-establishTemporaryConnection
        }
```

-- Direction: SCF → SSF, Timer: T_{etc}
-- This operation is used to create a connection to a resource for a limited period of time
-- (e.g. to play an announcement, to collect user information); it implies the use of the assist
-- procedure. Refer to clause 17 for a description of the procedures associated with this operation.

```
EstablishTemporaryConnectionArg {PARAMETERS-BOUND : bound} ::= SEQUENCE {
        assistingSSPIPRoutingAddress   [0] AssistingSSPIPRoutingAddress { bound},
        correlationID                  [1] CorrelationID { bound}                   OPTIONAL,
        partyToConnect                 CHOICE {
                                       legID            [2] LegID,
                                       callSegmentID    [7] CallSegmentID { bound}
                                       }                                            OPTIONAL,
        scfID                          [3] ScfID { bound}                           OPTIONAL,
        extensions                     [4] SEQUENCE SIZE(1..bound.&numOfExtensions)
                                          OF ExtensionField {bound}                 OPTIONAL,
        carrier                        [5] Carrier                                  OPTIONAL,
        serviceInteractionIndicators   [30] ServiceInteractionIndicators { bound}   OPTIONAL,
        serviceInteractionIndicatorsTwo [6] ServiceInteractionIndicatorsTwo         OPTIONAL,
        ...
        }

eventNotificationCharging {PARAMETERS-BOUND : bound} OPERATION ::= {
        ARGUMENT                       EventNotificationChargingArg { bound}
        RETURN RESULT                  FALSE
        ALWAYS RESPONDS                FALSE
        CODE                           opcode-eventNotificationCharging
        }
```
-- Direction: SSF $\rightarrow$ SCF, Timer: T_{enc}
-- This operation is used by the SSF to report to the SCF the occurrence of a specific charging event
-- type as previously requested by the SCF in a RequestNotificationChargingEvent operation.

```
EventNotificationChargingArg {PARAMETERS-BOUND : bound} ::= SEQUENCE {
        eventTypeCharging                    [0] EventTypeCharging { bound},
        eventSpecificInformationCharging     [1] EventSpecificInformationCharging { bound} OPTIONAL,
        legID                                [2] LegID                                   OPTIONAL,
        extensions                           [3] SEQUENCE SIZE(1..bound.&numOfExtensions)  OF
                                                 ExtensionField {bound}                  OPTIONAL,
        monitorMode                          [30] MonitorMode  DEFAULT notifyAndContinue,
        ...
        }
```
-- OPTIONAL denotes network-operator specific use.

```
eventReportBCSM {PARAMETERS-BOUND : bound} OPERATION ::= {
        ARGUMENT                             EventReportBCSMArg { bound}
        RETURN RESULT                        FALSE
        ALWAYS RESPONDS                      FALSE
        CODE                                 opcode-eventReportBCSM
        }
```

-- Direction: SSF $\rightarrow$ SCF, Timer: T_{erb}
-- This operation is used to notify the SCF of a call-related event (e.g. BCSM events such as busy or
-- no answer) previously requested by the SCF in a RequestReportBCSMEvent operation.

```
EventReportBCSMArg {PARAMETERS-BOUND : bound} ::= SEQUENCE {
        eventTypeBCSM                  [0] EventTypeBCSM,
        bcsmEventCorrelationID         [1] CorrelationID { bound}              OPTIONAL,
        eventSpecificInformationBCSM   [2] EventSpecificInformationBCSM { bound}  OPTIONAL,
        legID                          [3] LegID                               OPTIONAL,
        miscCallInfo                   [4] MiscCallInfo          DEFAULT       {messageType request},
        extensions                     [5] SEQUENCE SIZE(1..bound.&numOfExtensions)     OF
                                           ExtensionField {bound}              OPTIONAL,
        componentType                [6] ComponentType                    OPTIONAL,
        component                    [7] Component                        OPTIONAL,
        componentCorrelationID       [8] ComponentCorrelationID           OPTIONAL,
        ...
        }


eventReportFacility {PARAMETERS-BOUND : bound} OPERATION ::= {
        ARGUMENT                        EventReportFacilityArg { bound}
        RETURN RESULT                   FALSE
        ALWAYS RESPONDS                 FALSE
        CODE                            opcode-eventReportFacility
        }
```

-- Direction: SSF $\rightarrow$ SCF, Timer: T_{erf}
-- This operation is issued by the SSF to report the event to the SCF, that was previously requested by the
-- SCF, the CCF/SSF receives a DSS 1 message which contains a FACILITY IE. Criteria for the report, like
-- reception of the ReturnResult which is specified with ComponentType, is optionally checked
-- before issuing this operation.

```
EventReportFacilityArg {PARAMETERS-BOUND : bound} ::= SEQUENCE{
        componentType                   [0] ComponentType OPTIONAL,
        component                       [1] Component                       OPTIONAL,
        legID                           [2] LegID                           OPTIONAL,
        componentCorrelationID          [3]  ComponentCorrelationID         OPTIONAL,
        extensions                      [4] SEQUENCE SIZE(1..bound.&numOfExtensions)   OF
                                             ExtensionField {bound}         OPTIONAL,
        ...
        }
```
-- When the monitorDuration is over and the report condition specified with RequestReportFacilityEvent
-- was not met, component shall be absent.

```
facilitySelectedAndAvailable {PARAMETERS-BOUND : bound} OPERATION::= {
        ARGUMENT                        FacilitySelectedAndAvailableArg { bound}
        RETURN RESULT                   FALSE
        ERRORS                          {missingCustomerRecord |
                                        missingParameter |
                                        systemFailure |
                                        taskRefused |
                                        unexpectedComponentSequence |
                                        unexpectedDataValue |
                                        unexpectedParameter}
        CODE                            opcode-facilitySelectedAndAvailable
        }
```
-- Direction: SSF $\rightarrow$ SCF, Timer: T_{fs}
-- This operation is used for indication of a call termination attempt from the terminating half BCSM. (DP –
-- Facility_Selected_And_Available).
-- For additional information on this operation, refer to Rec. Q.1224.

```
FacilitySelectedAndAvailableArg {PARAMETERS-BOUND : bound} ::= SEQUENCE {
        dpSpecificCommonParameters      [0] DpSpecificCommonParameters { bound},
        calledPartyBusinessGroupID      [1] CalledPartyBusinessGroupID      OPTIONAL,
        calledPartySubaddress           [2] CalledPartySubaddress           OPTIONAL,
        callingPartyBusinessGroupID     [3] CallingPartyBusinessGroupID     OPTIONAL,
        callingPartyNumber              [4] CallingPartyNumber { bound}     OPTIONAL,
        originalCalledPartyID           [5] OriginalCalledPartyID { bound}  OPTIONAL,
        redirectingPartyID              [6] RedirectingPartyID { bound}     OPTIONAL,
        redirectionInformation          [7] RedirectionInformation          OPTIONAL,
        routeList                       [8] RouteList { bound}              OPTIONAL,
        travellingClassMark             [9] TravellingClassMark { bound}    OPTIONAL,
        extensions                   [10] SEQUENCE SIZE(1..bound.&numOfExtensions) OF
                                          ExtensionField {bound}        OPTIONAL,
        componentType                [11] ComponentType                 OPTIONAL,
        component                    [12] Component                     OPTIONAL,
        componentCorrelationID       [13] ComponentCorrelationID        OPTIONAL,
        ...
        }

-- For the OPTIONAL parameters, refer to clause 17 for the trigger detection point processing rules
-- to specify when these parameters are included in the message.

furnishChargingInformation {PARAMETERS-BOUND : bound} OPERATION ::= {
        ARGUMENT                     FurnishChargingInformationArg { bound}
        RETURN RESULT                FALSE
        ERRORS                       {missingParameter |
                                     taskRefused |
                                     unexpectedComponentSequence |
                                     unexpectedDataValue |
                                     unexpectedParameter}
        CODE                         opcode-furnishChargingInformation
        }

-- Direction: SCF → SSF, Timer: Tfci
-- This operation is used to request the SSF to generate, register a call record or to include some information
-- in the default call record. The registered call record is intended for off-line charging of the call.

FurnishChargingInformationArg {PARAMETERS-BOUND : bound} ::= FCIBillingChargingCharacteristics
{bound}

holdCallInNetwork  OPERATION ::= {
        ARGUMENT                     HoldCallInNetworkArg
        RETURN RESULT                FALSE
        ERRORS                       {missingParameter |
                                     systemFailure |
                                     taskRefused |
                                     unexpectedComponentSequence |
                                     unexpectedDataValue |
                                     unexpectedParameter}
        CODE                         opcode-holdCallInNetwork
        }
-- Direction: SCF → SSF, Timer: Thcn
-- This operation is used to provide the capability of queueing a call during the set-up phase (e.g. to provide
-- a call completion to busy, the call would be queued until the destination becomes free).
```

```
FurnishChargingInformationArg {PARAMETERS-BOUND : bound} ::= FCIBillingChargingCharacteristics
{bound}

holdCallInNetwork  OPERATION ::= {
        ARGUMENT                     HoldCallInNetworkArg
        RETURN RESULT                FALSE
        ERRORS                       {missingParameter |
                                     systemFailure |
                                     taskRefused |
                                     unexpectedComponentSequence |
                                     unexpectedDataValue |
                                     unexpectedParameter}
        CODE                         opcode-holdCallInNetwork
        }
```

```
-- holdcause is optional and denotes network-operator specific use.
```

```
initialDP {PARAMETERS-BOUND : bound}  OPERATION ::= {
        ARGUMENT                      InitialDPArg { bound}
        RETURN RESULT                 FALSE
        ERRORS                        {missingCustomerRecord |
                                      missingParameter |
                                      parameterOutOfRange |
                                      systemFailure |
                                      taskRefused |
                                      unexpectedComponentSequence |
                                      unexpectedDataValue |
                                      unexpectedParameter
                                      }
        CODE                          opcode-initialDP
        }
-- Direction: SSF → SCF, Timer: Tidp
-- This operation is used after a TDP to indicate request for service.

InitialDPArg {PARAMETERS-BOUND : bound} ::= SEQUENCE {
        serviceKey                          [0] ServiceKey                            OPTIONAL,
        dialledDigits                       [1] CalledPartyNumber { bound}            OPTIONAL,
        calledPartyNumber                   [2] CalledPartyNumber { bound}            OPTIONAL,
        callingPartyNumber                  [3] CallingPartyNumber { bound}           OPTIONAL,
        callingPartyBusinessGroupID         [4] CallingPartyBusinessGroupID           OPTIONAL,
        callingPartysCategory               [5] CallingPartysCategory                 OPTIONAL,
        callingPartySubaddress              [6] CallingPartySubaddress                OPTIONAL,
        cGEncountered                       [7] CGEncountered                         OPTIONAL,
        iPSSPCapabilities                   [8] IPSSPCapabilities { bound}            OPTIONAL,
        iPAvailable                         [9] IPAvailable { bound}                  OPTIONAL,
        locationNumber                      [10] LocationNumber { bound}              OPTIONAL,
        miscCallInfo                        [11] MiscCallInfo                         OPTIONAL,
        originalCalledPartyID               [12] OriginalCalledPartyID {bound}        OPTIONAL,
        serviceProfileIdentifier            [13] ServiceProfileIdentifier             OPTIONAL,
        terminalType                        [14] TerminalType                         OPTIONAL,
        extensions                          [15] SEQUENCE SIZE(1..bound.&numOfExtensions) OF
                                                 ExtensionField {bound}               OPTIONAL,
        triggerType                         [16] TriggerType                          OPTIONAL,
        highLayerCompatibility              [23] HighLayerCompatibility          OPTIONAL,
        serviceInteractionIndicators        [24] ServiceInteractionIndicators { bound}  OPTIONAL,
        additionalCallingPartyNumber        [25] AdditionalCallingPartyNumber { bound} OPTIONAL,
        forwardCallIndicators               [26] ForwardCallIndicators                OPTIONAL,
        bearerCapability                    [27] BearerCapability { bound}            OPTIONAL,
        eventTypeBCSM                       [28] EventTypeBCSM                        OPTIONAL,
        redirectingPartyID                  [29] RedirectingPartyID { bound}          OPTIONAL,
        redirectionInformation              [30] RedirectionInformation               OPTIONAL,
        cause                               [17] Cause { bound}                       OPTIONAL,
        componentType                       [18] ComponentType                        OPTIONAL,
        component                           [19] Component                            OPTIONAL,
        componentCorrelationID              [20] ComponentCorrelationID               OPTIONAL,
        iSDNAccessRelatedInformation        [21] ISDNAccessRelatedInformation         OPTIONAL,
        iNServiceCompatibilityIndication    [22] INServiceCompatibilityIndication { bound} OPTIONAL,
        genericNumbers                      [31] GenericNumbers { bound}              OPTIONAL,
        serviceInteractionIndicatorsTwo     [32] ServiceInteractionIndicatorsTwo      OPTIONAL,
        forwardGVNS                         [33] ForwardGVNS { bound}                 OPTIONAL,
        createdCallSegmentAssociation       [34] CSAID { bound}                       OPTIONAL,
        uSIServiceIndicator                 [35] USIServiceIndicator { bound}         OPTIONAL,
        uSIInformation                      [36] USIInformation { bound}              OPTIONAL,
        ...
        }
```

-- OPTIONAL for iPSSPCapabilities, iPAvailable, cGEncountered, and miscCallInfo denotes network-
-- operator specific use.
-- OPTIONAL for dialledDigits, callingPartyNumber, and callingPartysCategory refer to clause 17 for the trigger
-- detection point processing rules to specify when these parameters are included in the message.
-- OPTIONAL for terminalType indicates that this parameter applies only at originating or terminating
-- local exchanges if the SSF has this information.

```
initiateCallAttempt {PARAMETERS-BOUND : bound}  OPERATION ::= {
        ARGUMENT                        InitiateCallAttemptArg { bound}
        RETURN RESULT                   FALSE
        ERRORS                          {missingParameter |
                                        parameterOutOfRange |
                                        systemFailure |
                                        taskRefused |
                                        unexpectedComponentSequence |
                                        unexpectedDataValue |
                                        unexpectedParameter|
                                        unknownLegID
                                        }
        CODE                            opcode-initiateCallAttempt
        }
```

-- Direction: SCF $\rightarrow$ SSF, Timer: T_{ica}
-- This operation is used to request the SSF to create a new call to one call party using address
-- information provided by the SCF.

```
InitiateCallAttemptArg {PARAMETERS-BOUND : bound} ::= SEQUENCE {
        destinationRoutingAddress       [0] DestinationRoutingAddress { bound},
        alertingPattern                 [1] AlertingPattern                 OPTIONAL,
        iSDNAccessRelatedInformation    [2] ISDNAccessRelatedInformation    OPTIONAL,
        travellingClassMark             [3] TravellingClassMark { bound}    OPTIONAL,
        extensions                      [4] SEQUENCE SIZE(1..bound.&numOfExtensions)  OF
                                            ExtensionField {bound}          OPTIONAL,
        serviceInteractionIndicators    [29] ServiceInteractionIndicators { bound}  OPTIONAL,
        callingPartyNumber              [30] CallingPartyNumber { bound}    OPTIONAL,
        legToBeCreated                  [5] LegID DEFAULT  sendingSideID:leg1,
        newCallSegment                  [6] CallSegmentID { bound} DEFAULT initialCallSegment,
        iNServiceCompatibilityResponse  [7] INServiceCompatibilityResponse  OPTIONAL,
        serviceInteractionIndicatorsTwo [8] ServiceInteractionIndicatorsTwo OPTIONAL,
        ...
        }

manageTriggerData {PARAMETERS-BOUND : bound}  OPERATION ::= {
        ARGUMENT                        ManageTriggerDataArg { bound}
        RESULT                          ManageTriggerDataResultArg { bound}
        ERRORS                          {missingParameter |
                                        parameterOutOfRange |
                                        systemFailure |
                                        taskRefused |
                                        unexpectedComponentSequence |
                                        unexpectedDataValue |
                                        unexpectedParameter
                                        }
        CODE                            opcode-manageTriggerData
        }
```

-- Direction: SCF $\rightarrow$ SSF, Class 1, Timer: T_{mtd}
-- This operation is used to activate, deactivate or retrieve
-- the status of a trigger detection point linked to a subscriber profile known at the switch, e.g. related to an access line.

```
ManageTriggerDataArg {PARAMETERS-BOUND : bound}  ::= SEQUENCE {
        actionIndicator                    [0] ActionIndicator,
        triggerDataIdentifier              [1] TriggerDataIdentifier { bound},
        registratorIdentifier              [2] RegistratorIdentifier                OPTIONAL,
        extensions                         [3] SEQUENCE SIZE (1..bound.&numOfExtensions) OF
                                               ExtensionField {bound}               OPTIONAL,
        ...
        }


ManageTriggerDataResultArg {PARAMETERS-BOUND : bound}  ::= SEQUENCE {
        actionPerformed                    [0]  ActionPerformed,
        extensions                         [1] SEQUENCE SIZE (1..bound.&numOfExtensions) OF
                                               ExtensionField {bound}               OPTIONAL,
        ...
        }


mergeCallSegments {PARAMETERS-BOUND : bound} OPERATION ::= {
        ARGUMENT                           MergeCallSegmentsArg { bound}
        RETURN RESULT TRUE
        ERRORS                             {missingParameter |
                                           systemFailure |
                                           taskRefused |
                                           unexpectedComponentSequence |
                                           unexpectedDataValue |
                                           unexpectedParameter
                                           }
        CODE                               opcode-mergeCallSegments
        }
```

-- Direction: SCF → SSF, Timer: T_{mc}
-- This operation is issued by the SCF to merge two associated CSs with a single controlling leg into one
-- CS with that controlling leg.
-- For additional information on this operation, refer to Rec. Q.1224.

```
MergeCallSegmentsArg {PARAMETERS-BOUND : bound} ::= SEQUENCE {
        sourceCallSegment                  [0] CallSegmentID {bound},
        targetCallSegment                  [1] CallSegmentID {bound} DEFAULT initialCallSegment,
        extensions                         [2] SEQUENCE SIZE (1..bound.&numOfExtensions)
                                              OF ExtensionField {bound}             OPTIONAL,
        ...
        }

moveCallSegments {PARAMETERS-BOUND : bound} OPERATION ::= {
        ARGUMENT                           MoveCallSegmentsArg { bound}
        RETURN RESULT TRUE
        ERRORS                             {missingParameter |
                                           systemFailure |
                                           taskRefused |
                                           unexpectedComponentSequence |
                                           unexpectedDataValue |
                                           unexpectedParameter|
                                           unknownLegID
                                           }
        CODE                               opcode-moveCallSegments
        }
```
-- Direction: SCF → SSF, Timer T_{mcs}
-- This operation is used to merge two call segments into one.

```
MoveCallSegmentsArg {PARAMETERS-BOUND : bound} ::= SEQUENCE {
        targetCallSegmentAssociation    [0] CSAID { bound},
-- assignement of CSAID by SSF/SCF is ffs.
        callSegments                    [1] SEQUENCE SIZE (1..bound.&numOfCSs) OF SEQUENCE {
                sourceCallSegment       [0] CallSegmentID { bound}          DEFAULT initialCallSegment,
                newCallSegment          [1] CallSegmentID { bound}
                },
        legs                            [2] SEQUENCE SIZE (1..bound.&numOfLegs) OF SEQUENCE {
                sourceLeg               [0] LegID,
                newLeg                  [1] LegID
                },
        extensions                      [2] SEQUENCE SIZE (1..bound.&numOfExtensions)
                                            OF ExtensionField {bound}       OPTIONAL,
        ...
        }

moveLeg {PARAMETERS-BOUND : bound} OPERATION ::= {
        ARGUMENT                        MoveLegArg { bound}
        RETURN RESULT TRUE
        ERRORS                          {missingParameter |
                                        {systemFailure |
                                        taskRefused |
                                        unexpectedComponentSequence |
                                        unexpectedDataValue |
                                        unexpectedParameter|
                                        unknownLegID
                                        }
        CODE                            opcode-moveLeg
        }

-- Direction : SCF → SSF, Timer: Tml
-- This operation is issued by the SCF to move a leg from one CS to another with which it is associated.

MoveLegArg {PARAMETERS-BOUND : bound} ::=SEQUENCE {
        legIDToMove             [0] LegID,
        targetCallSegment       [1] CallSegmentID { bound} DEFAULT 1,
        extensions              [2] SEQUENCE SIZE (1..bound.&numOfExtensions)          OF
                                    ExtensionField {bound} OPTIONAL,
        ...
        }
-- For the OPTIONAL parameters, refer to clause 17 for the trigger
-- detection point processing rules to specify when these parameters are
-- included in the message.

oAbandon {PARAMETERS-BOUND : bound} OPERATION ::= {
        ARGUMENT                OAbandonArg { bound}
        RETURN RESULT           FALSE
        ERRORS                  {missingCustomerRecord |
                                missingParameter |
                                systemFailure |
                                taskRefused |
                                unexpectedComponentSequence |
                                unexpectedDataValue |
                                unexpectedParameter |
                                unknownLegID
                                }
        CODE                    opcode-oAbandon
        }
```

```
-- Direction: SSF → SCF, Timer: Tob
-- This operation is issued by the SSF after detecting a valid trigger condition at the O_Abandon DP or to
-- report an oAbandon event requested by the RequestReportBCSMEvent. For additional information on this
-- operation, refer to Rec. Q.1224.

OAbandonArg {PARAMETERS-BOUND : bound} ::= SEQUENCE {
        dpSpecificCommonParameters   [0] DpSpecificCommonParameters { bound},
        callSegmentID                [1] CallSegmentID { bound},
        releaseCause                 [2] Cause { bound}                           OPTIONAL,
        extensions                   [3] SEQUENCE SIZE(1..bound.&numOfExtensions) OF
                                         ExtensionField {bound}                   OPTIONAL,
        ...
        }
-- For the OPTIONAL parameters, refer to clause 17 for the trigger detection point rules to specify
-- when these parameters are included in the message.
-- Type definition for PointInCall is ffs. Use of T/EDP-R is ffs.


oAnswer {PARAMETERS-BOUND : bound}  OPERATION ::= {
        ARGUMENT                     OAnswerArg { bound}
        RETURN RESULT                FALSE
        ERRORS                       {missingCustomerRecord |
                                     missingParameter |
                                     parameterOutOfRange |
                                     systemFailure |
                                     taskRefused |
                                     unexpectedComponentSequence |
                                     unexpectedDataValue |
                                     unexpectedParameter
                                     }
        CODE                         opcode-oAnswer
        }

-- Direction: SSF → SCF, Timer: Toa
-- This operation is used for indication from the terminating half BCSM that the call is accepted and answered
-- by terminating party (e.g. terminating party goes off-hook, Q.931 Connect message received, ISDN-UP Answer
-- message received) (DP – O_Answer). For additional information on this operation, refer to Rec. Q.1224.

OAnswerArg {PARAMETERS-BOUND : bound} ::= SEQUENCE {
        dpSpecificCommonParameters   [0] DpSpecificCommonParameters { bound},
        callingPartyBusinessGroupID  [1] CallingPartyBusinessGroupID              OPTIONAL,
        callingPartySubaddress       [2] CallingPartySubaddress                   OPTIONAL,
        callingFacilityGroup         [3] FacilityGroup                            OPTIONAL,
        callingFacilityGroupMember   [4] FacilityGroupMember                      OPTIONAL,
        originalCalledPartyID        [5] OriginalCalledPartyID { bound}           OPTIONAL,
        redirectingPartyID           [6] RedirectingPartyID { bound}              OPTIONAL,
        redirectionInformation       [7] RedirectionInformation                   OPTIONAL,
        routeList                    [8] RouteList { bound}                       OPTIONAL,
        travellingClassMark          [9] TravellingClassMark { bound}             OPTIONAL,
        extensions                   [10] SEQUENCE SIZE(1..bound.&numOfExtensions)        OF
                                          ExtensionField {bound}                  OPTIONAL,
        ...
        }
-- For the OPTIONAL parameters, refer to clause 17 for the trigger detection point processing rules
-- to specify when these parameters are included in the message.
```

```
oCalledPartyBusy {PARAMETERS-BOUND : bound}  OPERATION ::= {
        ARGUMENT                        OCalledPartyBusyArg { bound}
        RETURN RESULT                   FALSE
        ERRORS                          {missingCustomerRecord |
                                        missingParameter |
                                        parameterOutOfRange |
                                        systemFailure |
                                        taskRefused |
                                        unexpectedComponentSequence |
                                        unexpectedDataValue |
                                        unexpectedParameter
                                        }
        CODE                            opcode-oCalledPartyBusy
        }
-- Direction: SSF → SCF, Timer: T_ob
-- This operation is used for Indication from the terminating half BCSM that the terminating party is busy
-- (DP  – O_Called_Party_Busy). For additional information on this operation, refer to Rec. Q.1224.

OCalledPartyBusyArg {PARAMETERS-BOUND : bound} ::= SEQUENCE {
        dpSpecificCommonParameters      [0] DpSpecificCommonParameters { bound},
        busyCause                       [1] Cause { bound}                          OPTIONAL,
        callingPartyBusinessGroupID     [2] CallingPartyBusinessGroupID             OPTIONAL,
        callingPartySubaddress          [3] CallingPartySubaddress                  OPTIONAL,
        callingFacilityGroup            [4] FacilityGroup                           OPTIONAL,
        callingFacilityGroupMember      [5] FacilityGroupMember                     OPTIONAL,
        originalCalledPartyID           [6] OriginalCalledPartyID { bound}          OPTIONAL,
        prefix                          [7] Digits { bound}                         OPTIONAL,
        redirectingPartyID              [8] RedirectingPartyID { bound}             OPTIONAL,
        redirectionInformation          [9] RedirectionInformation                  OPTIONAL,
        routeList                       [10] RouteList { bound}                     OPTIONAL,
        travellingClassMark             [11] TravellingClassMark { bound}           OPTIONAL,
        extensions                      [12] SEQUENCE SIZE(1..bound.&numOfExtensions)      OF
                                              ExtensionField {bound}                OPTIONAL,
        carrier                         [13] Carrier                                OPTIONAL,
        ...
        }
-- For the OPTIONAL parameters, refer to clause 17 for the trigger detection point processing rules
-- to specify when these parameters are included in the message.

oDisconnect {PARAMETERS-BOUND : bound}  OPERATION ::= {
        ARGUMENT                        ODisconnectArg { bound}
        RETURN RESULT                   FALSE
        ERRORS                          {missingCustomerRecord |
                                        missingParameter |
                                        parameterOutOfRange |
                                        systemFailure |
                                        taskRefused |
                                        unexpectedComponentSequence |
                                        unexpectedDataValue |
                                        unexpectedParameter
                                        }
        CODE                            opcode-oDisconnect
        }
-- Direction: SSF → SCF, Timer: T_od
-- This operation is used for a disconnect indication (e.g. on-hook, Q.931 Disconnect message, SS7 Release message)
-- is received from the originating party, or received from the terminating party via the terminating half BCSM.
-- (DP  – O_Disconnect). For additional information on this operation, refer to Rec.Q.1224.
```

```
ODisconnectArg {PARAMETERS-BOUND : bound} ::= SEQUENCE {
        dpSpecificCommonParameters   [0] DpSpecificCommonParameters { bound},
        callingPartyBusinessGroupID  [1] CallingPartyBusinessGroupID              OPTIONAL,
        callingPartySubaddress       [2] CallingPartySubaddress                   OPTIONAL,
        callingFacilityGroup         [3] FacilityGroup                            OPTIONAL,
        callingFacilityGroupMember   [4] FacilityGroupMember                      OPTIONAL,
        releaseCause                 [5] Cause { bound}                           OPTIONAL,
        routeList                    [6] RouteList { bound}                       OPTIONAL,
        extensions                   [7] SEQUENCE SIZE(1..bound.&numOfExtensions) OF
                                         ExtensionField {bound}                   OPTIONAL,
        carrier                      [8] Carrier                                  OPTIONAL,
        connectTime                  [9] Integer4                                 OPTIONAL,
        componentType                [10] ComponentType                           OPTIONAL,
        component                    [11] Component                               OPTIONAL,
        componentCorrelationID       [12] ComponentCorrelationID                  OPTIONAL,
        ...
        }

-- For the OPTIONAL parameters, refer to clause 17 for the trigger detection point processing rules
-- to specify when these parameters are included in the message.

oMidCall {PARAMETERS-BOUND : bound} OPERATION ::= {
        ARGUMENT                     MidCallArg { bound}
        RETURN RESULT                FALSE
        ERRORS                       {missingCustomerRecord |
                                     missingParameter |
                                     parameterOutOfRange |
                                     systemFailure |
                                     taskRefused |
                                     unexpectedComponentSequence |
                                     unexpectedDataValue |
                                     unexpectedParameter
                                     }
        CODE                         opcode-oMidCall
        }

-- Direction: SSF → SCF, Timer: Tomc
-- This operation is used to indicate a feature request is received from the originating party
-- (e.g. hook flash, ISDN feature activation, Q.931 HOLD or RETrieve message). (DP  – O_Mid_Call).
-- For additional information on this operation, refer to Rec. Q.1224.

MidCallArg {PARAMETERS-BOUND : bound} ::= SEQUENCE {
        dpSpecificCommonParameters   [0] DpSpecificCommonParameters { bound},
        calledPartyBusinessGroupID   [1] CalledPartyBusinessGroupID               OPTIONAL,
        calledPartySubaddress        [2] CalledPartySubaddress                    OPTIONAL,
        callingPartyBusinessGroupID  [3] CallingPartyBusinessGroupID              OPTIONAL,
        callingPartySubaddress       [4] CallingPartySubaddress                   OPTIONAL,
        featureRequestIndicator      [5] FeatureRequestIndicator                  OPTIONAL,
        extensions                   [6] SEQUENCE SIZE(1..bound.&numOfExtensions) OF
                                         ExtensionField {bound}                   OPTIONAL,
        carrier                      [7] Carrier    OPTIONAL,
        componentType                [8] ComponentType                            OPTIONAL,
        component                    [9] Component                                OPTIONAL,
        componentCorrelationID       [10] ComponentCorrelationID                  OPTIONAL,
        ...
        }

-- For the OPTIONAL parameters, refer to clause 17 for the trigger detection point processing rules
-- to specify when these parameters are included in the message.
```

```
oNoAnswer  {PARAMETERS-BOUND : bound} OPERATION ::= {
        ARGUMENT                        ONoAnswerArg { bound}
        RETURN RESULT                   FALSE
        ERRORS                          {missingCustomerRecord |
                                        missingParameter |
                                        parameterOutOfRange |
                                        systemFailure |
                                        taskRefused |
                                        unexpectedComponentSequence |
                                        unexpectedDataValue |
                                        unexpectedParameter
                                        }
        CODE                            opcode-oNoAnswer
        }
```

-- Direction: SSF $\rightarrow$ SCF, Timer: T_{ona}
-- This operation is used for indication from the terminating half BCSM that the terminating party does not
-- answer within a specified time period (DP – O_No_Answer). For additional information on this operation,
-- refer to Rec. Q.1224.

```
ONoAnswerArg {PARAMETERS-BOUND : bound} ::= SEQUENCE {
        dpSpecificCommonParameters      [0] DpSpecificCommonParameters { bound},
        callingPartyBusinessGroupID     [1] CallingPartyBusinessGroupID             OPTIONAL,
        callingPartySubaddress          [2] CallingPartySubaddress                  OPTIONAL,
        callingFacilityGroup            [3] FacilityGroup                           OPTIONAL,
        callingFacilityGroupMember      [4] FacilityGroupMember                     OPTIONAL,
        originalCalledPartyID           [5] OriginalCalledPartyID { bound}          OPTIONAL,
        prefix                          [6] Digits { bound}                         OPTIONAL,
        redirectingPartyID              [7] RedirectingPartyID { bound}             OPTIONAL,
        redirectionInformation          [8] RedirectionInformation                  OPTIONAL,
        routeList                       [9] RouteList { bound}                      OPTIONAL,
        travellingClassMark             [10] TravellingClassMark { bound}           OPTIONAL,
        extensions                      [11] SEQUENCE SIZE(1..bound.&numOfExtensions)      OF
                                             ExtensionField {bound}                 OPTIONAL,
        carrier                         [12] Carrier                                OPTIONAL,
        ...
        }
```

-- For the OPTIONAL parameters, refer to clause 17 for the trigger detection point processing rules
-- to specify when these parameters are included in the message.

```
originationAttempt {PARAMETERS-BOUND : bound}  OPERATION ::= {
        ARGUMENT                        OriginationAttemptArg { bound}
        RETURN RESULT                   FALSE
        ERRORS                          {missingCustomerRecord |
                                        missingParameter |
                                        systemFailure |
                                        taskRefused |
                                        unexpectedComponentSequence |
                                        unexpectedDataValue |
                                        unexpectedParameter
                                        }
        CODE                            opcode-originationAttempt
        }
```

-- Direction: SSF $\rightarrow$ SCF, Timer: T_{ora}
-- This operation is used for indication of a call origination attempt from the originating half BCSM. (DP –
-- Origination_Attempt).
-- For additional information on this operation, refer to Rec. Q.1224.

```
OriginationAttemptArg {PARAMETERS-BOUND : bound} ::= SEQUENCE {
        dpSpecificCommonParameters   [0] DpSpecificCommonParameters { bound},
        callingPartyBusinessGroupID  [1] CallingPartyBusinessGroupID            OPTIONAL,
        callingPartySubaddress       [2] CallingPartySubaddress                 OPTIONAL,
        callingFacilityGroup         [3] FacilityGroup                          OPTIONAL,
        callingFacilityGroupMember   [4] FacilityGroupMember                    OPTIONAL,
        carrier                      [5] Carrier                                OPTIONAL,
        travellingClassMark          [6] TravellingClassMark { bound}           OPTIONAL,
        extensions                   [7] SEQUENCE SIZE(1..bound.&numOfExtensions) OF
                                         ExtensionField {bound}                 OPTIONAL,
        componentType                [8] ComponentType                          OPTIONAL,
        component                    [9] Component                              OPTIONAL,
        componenttCorrelationID      [10] ComponentCorrelationID                OPTIONAL,
        ...
        }
```

-- For the OPTIONAL parameters, refer to clause 17 for the trigger detection point processing rules
-- to specify when these parameters are included in the message.

```
originationAttemptAuthorized {PARAMETERS-BOUND : bound}  OPERATION ::= {
        ARGUMENT                     OriginationAttemptAuthorizedArg { bound}
        RETURN RESULT                FALSE
        ERRORS                       {missingCustomerRecord |
                                     missingParameter |
                                     parameterOutOfRange |
                                     systemFailure |
                                     taskRefused |
                                     unexpectedComponentSequence |
                                     unexpectedDataValue |
                                     unexpectedParameter
                                     }
        CODE                         opcode-originationAttemptAuthorized
        }
```

-- Direction: SSF $\rightarrow$ SCF, Timer: T_{oaa}
-- This operation is used to Indicate the desire to place outgoing call (e.g. off-hook, Q.931 Setup message,
-- ISDN-UP IAM message) and authority/ability to place outgoing call verified (DP –
-- Origination_Attempt_Authorized).
-- For additional information on this operation, refer to Rec. Q.1224.

```
OriginationAttemptAuthorizedArg {PARAMETERS-BOUND : bound} ::= SEQUENCE {
        dpSpecificCommonParameters   [0] DpSpecificCommonParameters { bound},
        dialledDigits                [1] CalledPartyNumber { bound}             OPTIONAL,
        callingPartyBusinessGroupID  [2] CallingPartyBusinessGroupID            OPTIONAL,
        callingPartySubaddress       [3] CallingPartySubaddress                 OPTIONAL,
        callingFacilityGroup         [4] FacilityGroup                          OPTIONAL,
        callingFacilityGroupMember   [5] FacilityGroupMember                    OPTIONAL,
        travellingClassMark          [6] TravellingClassMark { bound}           OPTIONAL,
        extensions                   [7] SEQUENCE SIZE(1..bound.&numOfExtensions) OF
                                         ExtensionField {bound}                 OPTIONAL,
        carrier                      [8] Carrier                                OPTIONAL,
        componentType                [9] ComponentType                          OPTIONAL,
        component                    [10] Component                             OPTIONAL,
        componentCorrelationID       [11] ComponentCorrelationID                OPTIONAL,
        ...
        }
```

-- For the OPTIONAL parameters, refer to clause 17 for the trigger detection point processing rules
-- to specify when these parameters are included in the message.

```
oSuspended {PARAMETERS-BOUND : bound}  OPERATION ::= {
        ARGUMENT                        OSuspendedArg { bound}
        RETURN RESULT                   FALSE
        ERRORS                          {missingCustomerRecord |
                                        missingParameter |
                                        systemFailure |
                                        taskRefused |
                                        unexpectedComponentSequence |
                                        unexpectedDataValue |
                                        unexpectedParameter |
                                        unknownLegID
                                        }
        CODE                            opcode-oSuspended
        }
```

-- Direction: SSF → SCF, Timer: T_{os}
-- This operation is issued by the SSF after detecting a valid trigger condition at the O_Suspended DP or to
-- report an oSuspended event requested by the RequestReportBCSMEvent. For additional information on
-- this operation, refer to Rec. Q.1224.

```
OSuspendedArg {PARAMETERS-BOUND : bound} ::= SEQUENCE {
        dpSpecificCommonParameters      [0] DpSpecificCommonParameters { bound},
        legID                           [1] LegID                                       OPTIONAL,
        extensions                      [2] SEQUENCE SIZE(1..bound.&numOfExtensions)   OF
                                            ExtensionField {bound}                      OPTIONAL,
        ...
        }
```

-- For the OPTIONAL parameters, refer to clause 17 for the trigger detection point processing rules
-- to specify when these parameters are included in the message.
-- Modification to LegID is ffs. Use for T/EDP-R is ffs.

```
reconnect {PARAMETERS-BOUND : bound} OPERATION ::= {
        ARGUMENT                        ReconnectArg { bound}
        RETURN RESULT                   FALSE
        ERRORS                          {missingParameter |
                                        systemFailure |
                                        taskRefused |
                                        unexpectedComponentSequence |
                                        unexpectedDataValue |
                                        unexpectedParameter
                                        }
        CODE                            opcode-reconnect
        }
```

-- Direction: SCF → SSF, Timer: T_{re}
-- This operation is issued by the SCF to reestablish communication between the controlling leg and the
-- (held) passive leg(s). For additional information on this operation, refer to Rec. Q.1224.

```
ReconnectArg {PARAMETERS-BOUND : bound} ::= SEQUENCE {
        notificationDuration            [0] ApplicationTimer                            OPTIONAL,
        alertingPattern                 [1] AlertingPattern                             OPTIONAL,
        displayInformation              [2] DisplayInformation { bound}                 OPTIONAL,
        extensions                      [3] SEQUENCE SIZE(1..bound.&numOfExtensions)   OF
                                            ExtensionField {bound}                      OPTIONAL,
        callSegmentID                   [4] CallSegmentID {bound}                       OPTIONAL,
        ...
        }
```

```
-- For the OPTIONAL parameters, refer to clause 17 for the trigger detection point processing rules
-- to specify when these parameters are included in the message.

releaseCall {PARAMETERS-BOUND : bound}  OPERATION ::= {
        ARGUMENT                    ReleaseCallArg { bound}
        RETURN RESULT               FALSE
        ALWAYS RESPONDS             FALSE
        CODE                        opcode-releaseCall
        }

-- Direction: SCF → SSF, Timer: Trc
-- This operation is used to tear down an existing call at any phase of the call for all parties
-- involved in the call.

ReleaseCallArg {PARAMETERS-BOUND : bound} ::= CHOICE {
        initialCallSegment          Cause { bound},
        associatedCallSegment       [1] SEQUENCE {
                callSegment         [0] INTEGER (2..bound.&numOfCSs),
                releaseCause        [1] Cause { bound}                          OPTIONAL
                },
        allCallSegments             [2] SEQUENCE {
                 releaseCause       [0] Cause { bound}                          OPTIONAL
                }
        }

-- A default value of decimal 31 (normal unspecified) should be coded appropriately.

reportUTSI {PARAMETERS-BOUND : bound}  OPERATION ::= {
        ARGUMENT                    ReportUTSIArg  { bound}
        RETURN RESULT               FALSE
        ALWAYS RESPONDS             FALSE
        CODE                        opcode-reportUTSI
        }

-- Direction: SSF → SCF, Timer: Tru
-- This operation is issued by the SSF in the context of the USI feature. It is used to report the receipt
-- of a User to Service Information (UTSI) IE to the SCF.

ReportUTSIArg {PARAMETERS-BOUND : bound} ::= SEQUENCE {
        uSIServiceIndicator         [0] USIServiceIndicator { bound},
        legID                       [1] LegID                   DEFAULT receivingSideID:leg1,
        uSIInformation              [2] USIInformation { bound},
        extensions                  [3] SEQUENCE SIZE(1..bound.&numOfExtensions)  OF
                                        ExtensionField {bound}                  OPTIONAL,
        ...
        }

requestCurrentStatusReport {PARAMETERS-BOUND : bound} OPERATION ::= {
        ARGUMENT                    RequestCurrentStatusReportArg { bound}
        RESULT                      RequestCurrentStatusReportResultArg { bound}
        ERRORS                      {missingParameter |
                                    parameterOutOfRange |
                                    systemFailure |
                                    taskRefused |
                                    unexpectedComponentSequence |
                                    unexpectedDataValue |
                                    unexpectedParameter
                                    unknownResource
                                    }
        CODE                         opcode-requestCurrentStatusReport
        }

-- Direction: SCF → SSF, Timer: Trcs
-- This operation is used to request the SSF to report immediately the busy/idle status of a physical
-- termination resource.

RequestCurrentStatusReportArg {PARAMETERS-BOUND : bound} ::= ResourceID { bound}

RequestCurrentStatusReportResultArg {PARAMETERS-BOUND : bound} ::= SEQUENCE {
        resourceStatus               [0] ResourceStatus,
        resourceID                   [1] ResourceID { bound}                    OPTIONAL,
        extensions                   [2] SEQUENCE SIZE(1..bound.&numOfExtensions) OF
                                         ExtensionField {bound}                 OPTIONAL,
        ...
        }

requestEveryStatusChangeReport {PARAMETERS-BOUND : bound}  OPERATION ::= {
        ARGUMENT                     RequestEveryStatusChangeReportArg { bound}
        RETURN RESULT TRUE
        ERRORS                       {missingParameter |
                                     parameterOutOfRange |
                                     systemFailure |
                                     taskRefused |
                                     unexpectedComponentSequence |
                                     unexpectedDataValue |
                                     unexpectedParameter
                                     unknownResource
                                     }
        CODE                         opcode-requestEveryStatusChangeReport
        }

-- Direction: SCF → SSF, Timer: Tres
-- This operation is used to request the SSF to report every change of busy/idle status of a physical
-- termination resource.

RequestEveryStatusChangeReportArg {PARAMETERS-BOUND : bound} ::= SEQUENCE {
        resourceID                   [0] ResourceID { bound},
        correlationID                [1] CorrelationID { bound}                 OPTIONAL,
        monitorDuration              [2] Duration                               OPTIONAL,
        extensions                   [3] SEQUENCE SIZE(1..bound.&numOfExtensions) OF
                                         ExtensionField {bound}                 OPTIONAL,
        ...
        }

-- For correlationID OPTIONAL denotes network-operator optional.
-- monitorDuration is required if outside the context of a call. It is not expected if we are in the context
-- of a call, because in that case the end of the call implicitly means the end of the monitoring.

requestFirstStatusMatchReport {PARAMETERS-BOUND : bound}  OPERATION ::= {
        ARGUMENT                     RequestFirstStatusMatchReportArg { bound}
        RETURN RESULT TRUE
        ERRORS                       {missingParameter |
                                     parameterOutOfRange |
                                     systemFailure |
                                     taskRefused |
                                     unexpectedComponentSequence |
                                     unexpectedDataValue |
                                     unexpectedParameter
                                     unknownResource
                                     }
        CODE                               opcode-requestFirstStatusMatchReport
        }
```
-- Direction: SCF $\rightarrow$ SSF, Timer: T_{rfs}
-- This operation is used to request the SSF to report the first change busy/idle to the specified status of
-- a physical termination resource.

```
RequestFirstStatusMatchReportArg {PARAMETERS-BOUND : bound} ::= SEQUENCE {
     resourceID              [0] ResourceID { bound}                          OPTIONAL,
     resourceStatus          [1] ResourceStatus                               OPTIONAL,
     correlationID           [2] CorrelationID { bound}                       OPTIONAL,
     monitorDuration         [3] Duration                                     OPTIONAL,
     extensions              [4] SEQUENCE SIZE(1..bound.&numOfExtensions)     OF
                                  ExtensionField {bound}                      OPTIONAL,
     bearerCapability        [5] BearerCapability { bound}                    OPTIONAL,
     ...
     }
```
-- For correlationID OPTIONAL denotes network-operator optional.
-- monitorDuration is required if outside the context of a call. It is not expected if we are in the context
-- of a call, because in that case the end of the call implicitly means the end of the monitoring.

```
requestNotificationChargingEvent {PARAMETERS-BOUND : bound}  OPERATION ::= {
     ARGUMENT                           RequestNotificationChargingEventArg { bound}
     RETURN RESULT                      FALSE
     ERRORS                             {missingParameter |
                                        parameterOutOfRange |
                                        systemFailure |
                                        taskRefused |
                                        unexpectedComponentSequence |
                                        unexpectedDataValue |
                                        unexpectedParameter
                                        }
     CODE                               opcode-requestNotificationChargingEvent
     }
```

-- Direction: SCF $\rightarrow$ SSF, Timer: T_{rnc}
-- This operation is used by the SCF to instruct the SSF on how to manage the charging events
-- which are received from other FEs and not under control of the service logic instance.

```
RequestNotificationChargingEventArg {PARAMETERS-BOUND : bound} ::= SEQUENCE
SIZE(1..bound.&numOfChargingEvents) OF
          ChargingEvent {bound}

requestReportBCSMEvent {PARAMETERS-BOUND : bound}  OPERATION ::= {
     ARGUMENT                           RequestReportBCSMEventArg { bound}
     RETURN RESULT                      FALSE
     ERRORS                             {missingParameter |
                                        parameterOutOfRange |
                                        systemFailure |
                                        taskRefused |
                                        unexpectedComponentSequence |
                                        unexpectedDataValue |
                                        unexpectedParameter
                                        }
     CODE                               opcode-requestReportBCSMEvent
     }
```

```
-- Direction: SCF → SSF, Timer: Trrb
-- This operation is used to request the SSF to monitor for a call-related event (e.g. BCSM events such as
-- busy or no answer), then send a notification back to the SCF when the event is detected.
-- It is proposed that Event Detection Point (EDP) processing is always initiated by RequestReportBCSMEvent and the
-- EDP may be acknowledged with either an EventReportBCSM or by a DP-specific operations:
-- NOTE –The application context should identify whether BCSM Event Handling Package
-- is being used, or whether DP Specific Event Handling Package
-- is being used.
--      – for a particular IN, only one of the two alternatives identified by the respective Packages should be
--        selected (i.e. only one approach should be selected for a given application context).
--      – Every EDP must be explicitly armed by the SCF via a RequestReportBCSMEvent operation. No
--        implicit arming of EDPs at the SSF after reception of any operation (different from
--        RequestReportBCSMEvent) from the SCF is allowed.

RequestReportBCSMEventArg {PARAMETERS-BOUND : bound} ::= SEQUENCE {
      bcsmEvents                  [0] SEQUENCE SIZE(1..bound.&numOfBCSMEvents)     OF
                                       BCSMEvent {bound},
      bcsmEventCorrelationID      [1] CorrelationID { bound}                       OPTIONAL,
      extensions                  [2] SEQUENCE SIZE(1..bound.&numOfExtensions)OF
                                       ExtensionField {bound}                      OPTIONAL,
      ...
      }

-- Indicates the BCSM related events for notification.
-- For correlationID OPTIONAL denotes network-operator optional.

requestReportFacilityEvent {PARAMETERS-BOUND : bound} OPERATION ::= {
      ARGUMENT                    RequestReportFacilityEventArg { bound}
      RETURN RESULT               FALSE
      ERRORS                      {missingParameter |
                                  systemFailure |
                                  taskRefused |
                                  unexpectedComponentSequence |
                                  unexpectedDataValue |
                                  unexpectedParameter |
                                  unknownLegID
                                  }
      CODE                        opcode-requestReportFacilityEvent
}

-- Direction: SCF → SSF, Timer: Trrfe
-- This operation is issued by the SCF to request the SSF to report the SCF the event that the CCF/SSF
-- receives a DSS 1 message which contains a FACILITY IE during a BCSM being suspended at a DP.

RequestReportFacilityEventArg {PARAMETERS-BOUND : bound}        ::= SEQUENCE{
      componentTypes              [0] SEQUENCE SIZE(1..3) OF ComponentType DEFAULT {any},
      legID                       [1] LegID                                        OPTIONAL,
      componentCorrelationID      [2] ComponentCorrelationID                       OPTIONAL,
      monitorDuration             [3] Duration,
      extensions                  [4] SEQUENCE SIZE(1..bound.&numOfExtensions)OF
                                      ExtensionField {bound}                       OPTIONAL,
      ...
      }
-- componentTypes specifies the component types which should be reported to the SCF.
-- monitorDuration specifies the monitor duration.
```

```
requestReportUTSI  {PARAMETERS-BOUND : bound} OPERATION ::= {
        ARGUMENT                    RequestReportUTSIArg { bound}
        RETURN RESULT               FALSE
        ERRORS                      {missingParameter |
                                    systemFailure |
                                    taskRefused |
                                    unexpectedComponentSequence |
                                    unexpectedDataValue |
                                    unexpectedParameter
                                    }
        CODE                        opcode-requestReportUTSI
        }
```

-- Direction: SCF → SSF, Timer: T_{rru}
-- This operation is issued by the SCF in the context of the USI feature to request the SSF to monitor for
-- a User to Service Information (UTSI) information element, which are received from a user.

```
RequestReportUTSIArg {PARAMETERS-BOUND : bound} ::= SEQUENCE {
        requestedUTSIList           [0] RequestedUTSIList { bound},
        extensions                  [1] SEQUENCE SIZE(1..bound.&numOfExtensions) OF
                                        ExtensionField {bound}                  OPTIONAL,
        legID                       [2] LegID                                   OPTIONAL,
        ...
        }


resetTimer {PARAMETERS-BOUND : bound}  OPERATION ::= {
        ARGUMENT                    ResetTimerArg { bound}
        RETURN RESULT               FALSE
        ERRORS                      {missingParameter |
                                    parameterOutOfRange |
                                    taskRefused |
                                    unexpectedComponentSequence |
                                    unexpectedDataValue |
                                    unexpectedParameter
                                    }
        CODE                        opcode-resetTimer
        }
```

-- Direction: SCF → SSF, Timer: T_{rt}
-- This operation is used to request the SSF to refresh an application timer in the SSF.

```
ResetTimerArg {PARAMETERS-BOUND : bound} ::= SEQUENCE {
        timerID                     [0] TimerID                         DEFAULT tssf,
        timervalue                  [1] TimerValue,
        extensions                  [2] SEQUENCE SIZE(1..bound.&numOfExtensions)   OF
                                        ExtensionField {bound}          OPTIONAL,
        callSegmentID               [3] CallSegmentID { bound}          OPTIONAL,
        ...
        }

routeSelectFailure {PARAMETERS-BOUND : bound}  OPERATION ::= {
        ARGUMENT                    RouteSelectFailureArg { bound}
        RETURN RESULT               FALSE
        ERRORS                      {missingCustomerRecord |
                                    missingParameter |
                                    parameterOutOfRange |
                                    systemFailure |
                                    taskRefused |
                                    unexpectedComponentSequence |
                                    unexpectedDataValue |
                                    unexpectedParameter
                                    }
        CODE                               opcode-routeSelectFailure
        }
```

-- Direction: SSF → SCF, Timer: T_{rsf}
-- This operation is used to indicate that the SSP is unable to select a route (e.g. unable to determine a
-- correct route, no more routes on route list) or indication from the terminating half BCSM that a call
-- cannot be presented to the terminating party (e.g. network congestion) (DP – Route_Select_Failure).
-- For additional information on this operation, refer to Rec. Q.1224.

```
RouteSelectFailureArg {PARAMETERS-BOUND : bound} ::= SEQUENCE {
     dpSpecificCommonParameters   [0] DpSpecificCommonParameters { bound},
     dialledDigits                [1] CalledPartyNumber { bound}              OPTIONAL,
     callingPartyBusinessGroupID  [2] CallingPartyBusinessGroupID             OPTIONAL,
     callingPartySubaddress       [3] CallingPartySubaddress                  OPTIONAL,
     callingFacilityGroup         [4] FacilityGroup                           OPTIONAL,
     callingFacilityGroupMember   [5] FacilityGroupMember                     OPTIONAL,
     failureCause                 [6] Cause { bound}                          OPTIONAL,
     originalCalledPartyID        [7] OriginalCalledPartyID { bound}          OPTIONAL,
     prefix                       [8] Digits { bound}                         OPTIONAL,
     redirectingPartyID           [9] RedirectingPartyID { bound}             OPTIONAL,
     redirectionInformation       [10] RedirectionInformation                 OPTIONAL,
     routeList                    [11] RouteList { bound}                     OPTIONAL,
     travellingClassMark          [12] TravellingClassMark { bound}           OPTIONAL,
     extensions                   [13] SEQUENCE SIZE(1..bound.&numOfExtensions)     OF
                                       ExtensionField {bound}                 OPTIONAL,
     carrier                      [14] Carrier                                OPTIONAL,
     ...
     }
```

-- For the OPTIONAL parameters, refer to clause 17 for the trigger detection point processing
-- rules to specify when these parameters are included in the message.

```
selectFacility {PARAMETERS-BOUND : bound}  OPERATION ::= {
     ARGUMENT                           SelectFacilityArg { bound}
     RETURN RESULT                      FALSE
     ERRORS                             {missingParameter |
                                        parameterOutOfRange |
                                        systemFailure |
                                        taskRefused |
                                        unexpectedComponentSequence |
                                        unexpectedDataValue |
                                        unexpectedParameter
                                        }
     CODE                               opcode-selectFacility
     }
```

-- Direction: SCF → SSF, Timer: T_{sf}
-- This operation is used to request the SSF to perform the terminating basic call processing
-- actions to select the terminating line if it is idle, or select an idle line from a multi-line hunt
-- group, or select an idle trunk from a trunk group, as appropriate. If no idle line or trunk is
-- available, the SSF determines that the terminating facility is busy.

```
SelectFacilityArg {PARAMETERS-BOUND : bound} ::= SEQUENCE {
        alertingPattern                     [0] AlertingPattern OPTIONAL,
        destinationNumberRoutingAddress     [1] CalledPartyNumber { bound}              OPTIONAL,
        iSDNAccessRelatedInformation        [2] ISDNAccessRelatedInformation            OPTIONAL,
        calledFacilityGroup                 [3] FacilityGroup                           OPTIONAL,
        calledFacilityGroupMember           [4] FacilityGroupMember                     OPTIONAL,
        originalCalledPartyID               [5] OriginalCalledPartyID { bound}          OPTIONAL,
        extensions                          [6] SEQUENCE SIZE(1..bound.&numOfExtensions)  OF
                                                ExtensionField {bound}                  OPTIONAL,
        displayInformation                  [7] DisplayInformation { bound}             OPTIONAL,
        serviceInteractionIndicators        [8] ServiceInteractionIndicators { bound}   OPTIONAL,
        iNServiceCompatibilityResponse      [9] INServiceCompatibilityResponse          OPTIONAL,
        forwardGVNS                         [10] ForwardGVNS { bound}                   OPTIONAL,
        backwardGVNS                        [11] BackwardGVNS { bound}                  OPTIONAL,
        serviceInteractionIndicatorsTwo     [12] ServiceInteractionIndicatorsTwo        OPTIONAL,
        correlationID                       [13] CorrelationID { bound}                 OPTIONAL,
        scfID                               [14] ScfID { bound}                         OPTIONAL,
        callSegmentID                       [15] CallSegmentID {bound}                  OPTIONAL,
        legToBeCreated                      [16] LegID                                  OPTIONAL,
        ...
        }
```

-- *OPTIONAL parameters are only provided when modifiying basic call processing values.*

```
selectRoute {PARAMETERS-BOUND : bound}  OPERATION ::= {
        ARGUMENT                    SelectRouteArg { bound}
        RETURN RESULT               FALSE
        ERRORS                      {missingParameter |
                                    parameterOutOfRange |
                                    systemFailure |
                                    taskRefused |
                                    unexpectedComponentSequence |
                                    unexpectedDataValue |
                                    unexpectedParameter
                                    }
        CODE                        opcode-selectRoute
        }
```

-- *Direction: SCF $\rightarrow$ SSF, Timer: T_{sr}*
-- *This operation is used to request the SSF to perform the originating basic call processing actions to*
-- *determine routing information and select a route for a call, based either on call information available*
-- *to the SSF, or on call information provided by the SCF (e.g. for alternate routing), to include the*
-- *called party address, type of call, carrier, route index, and one or more alternate route indices.*
-- *Based on the routing information, the SSF attempts to select a primary route for the call, and if the*
-- *route is busy, attempts to select an alternate route. The SSF may fail to select a route for the call*
-- *if all routes are busy.*

```
SelectRouteArg {PARAMETERS-BOUND : bound} ::= SEQUENCE {
        destinationRoutingAddress       [0] DestinationRoutingAddress { bound},
        alertingPattern                 [1] AlertingPattern                             OPTIONAL,
        correlationID                   [2] CorrelationID { bound}                      OPTIONAL,
        iSDNAccessRelatedInformation    [3] ISDNAccessRelatedInformation                OPTIONAL,
        originalCalledPartyID           [4] OriginalCalledPartyID { bound}              OPTIONAL,
        routeList                       [5] RouteList { bound}                          OPTIONAL,
        scfID                           [6] ScfID { bound}                              OPTIONAL,
        travellingClassMark             [7] TravellingClassMark { bound}                OPTIONAL,
        extensions                      [8] SEQUENCE SIZE(1..bound.&numOfExtensions)    OF
                                            ExtensionField {bound}                      OPTIONAL,
        carrier                         [9] Carrier                                     OPTIONAL,
        serviceInteractionIndicators    10] ServiceInteractionIndicators { bound}       OPTIONAL,
        iNServiceCompatibilityResponse [11] INServiceCompatibilityResponse            OPTIONAL,
        forwardGVNS                    [12] ForwardGVNS { bound}                      OPTIONAL,
        backwardGVNS                   [13] BackwardGVNS { bound}                     OPTIONAL,
        serviceInteractionIndicatorsTwo [14] ServiceInteractionIndicatorsTwo          OPTIONAL,
        callSegmentID                  [15] CallSegmentID {bound}                     OPTIONAL,
        legToBeCreated                 [16] LegID                                     OPTIONAL,
        ...
        }
```

-- OPTIONAL parameters are only provided when modifiying basic call processing values.

```
sendChargingInformation {PARAMETERS-BOUND : bound}  OPERATION ::= {
        ARGUMENT                SendChargingInformationArg { bound}
        RETURN RESULT           FALSE
        ERRORS                  {missingParameter |
                                unexpectedComponentSequence |
                                unexpectedParameter |
                                parameterOutOfRange |
                                systemFailure |
                                taskRefused |
                                unknownLegID
                                }
        CODE                    opcode-sendChargingInformation
        }
```

-- Direction: SCF $\rightarrow$ SSF, Timer: T_{sci}
-- This operation is used to instruct the SSF on the charging information to send by the SSF.
-- The charging information can either be sent back by means of signalling or internal
-- if the SSF is located in the local exchange. In the local exchange,
-- this information may be used to update the charge meter or to create a standard call record.

```
SendChargingInformationArg {PARAMETERS-BOUND : bound} ::= SEQUENCE {
        sCIBillingChargingCharacteristics    [0] SCIBillingChargingCharacteristics { bound},
        partyToCharge                        [1] LegID,
        extensions                           [2] SEQUENCE SIZE(1..bound.&numOfExtensions)   OF
                                             ExtensionField {bound}                 OPTIONAL,
        ...
        }
```

```
sendFacilityInformation {PARAMETERS-BOUND : bound} OPERATION ::= {
        ARGUMENT                SendFacilityInformationArg { bound}
        RETURN RESULT           FALSE
        ERRORS                  {missingParameter |
                                unexpectedComponentSequence |
                                unexpectedParameter |
                                unexpectedDataValue |
                                systemFailure |
                                taskRefused |
                                unknownLegID
                                }
        CODE                    opcode-sendFacilityInformation
        }
```

-- Direction: SCF $\rightarrow$ SSF, Timer: T_{sfi}
-- This operation is issued by the SCF during a BCSM being suspended at a DP to request the CCF/SSF
-- sending a FACILITY IE to a user with a specified DSS 1 message.

```
SendFacilityInformationArg {PARAMETERS-BOUND : bound} ::= SEQUENCE{
        componentType                         [0] ComponentType,
        legID                                 [1] LegID                    OPTIONAL,
        componentCorrelationID                [2] ComponentCorrelationID   OPTIONAL,
        component                             [3] Component,
        callProcessingOperationCorrelationID  [4] CallProcessingOperationCorrelationID
                                                                           DEFAULT      fACility,
        extensions                            [5] SEQUENCE SIZE(1..bound.&numOfExtensions)  OF
                                                  ExtensionField {bound}   OPTIONAL,
        ...
        }
-- FACILITY IE will be delivered with the specified DSS 1 message.  The message is specified with the
-- callProcessingOperationCorrelationID


sendSTUI  {PARAMETERS-BOUND : bound} OPERATION ::= {
        ARGUMENT                    SendSTUIArg { bound}
        RETURN RESULT               FALSE
        ERRORS                      {missingParameter |
                                    parameterOutOfRange |
                                    unexpectedComponentSequence |
                                    unexpectedParameter |
                                    unexpectedDataValue |
                                    systemFailure |
                                    taskRefused |
                                    unknownLegID
                                    }
        CODE                        opcode-sendSTUI
        }

-- Direction: SCF → SSF, Timer: T_ss
-- This operation is issued by the SCF in the context of the USI feature. It is used to request the SSF
-- to send a Service to User Information (STUI) information element to the indicated user.

SendSTUIArg {PARAMETERS-BOUND : bound} ::= SEQUENCE {
        uSIServiceIndicator         [0] USIServiceIndicator { bound},
        legID                       [1] LegID                        DEFAULT sendingSideID:leg1,
        uSIInformation              [2] USIInformation { bound},
        extensions                  [3] SEQUENCE SIZE(1..bound.&numOfExtensions)  OF
                                        ExtensionField {bound}                OPTIONAL,
        ...
        }


serviceFilteringResponse {PARAMETERS-BOUND : bound}  OPERATION ::= {
        ARGUMENT                    ServiceFilteringResponseArg { bound}
        RETURN RESULT               FALSE
        ALWAYS RESPONDS             FALSE
        CODE                        opcode-serviceFilteringResponse
        }

-- Direction: SSF → SCF, Timer: T_sfr
-- This operation is used to send back to the SCF the values of counters specified in a previous
-- ActivatedServiceFiltering operation
```

```
ServiceFilteringResponseArg {PARAMETERS-BOUND : bound} ::= SEQUENCE {
        countersValue                 [0] CountersValue,
        filteringCriteria             [1] FilteringCriteria { bound},
        extensions                    [2] SEQUENCE SIZE(1..bound.&numOfExtensions) OF
                                          ExtensionField {bound}                 OPTIONAL,
        responseCondition             [3] ResponseCondition                      OPTIONAL,
        ...
        }

splitLeg {PARAMETERS-BOUND : bound}  OPERATION ::= {
        ARGUMENT                      SplitLegArg { bound}
        RETURN RESULT TRUE
        ERRORS                        {missingParameter |
                                      unexpectedComponentSequence |
                                      unexpectedParameter |
                                      unexpectedDataValue |
                                      systemFailure |
                                      taskRefused |
                                      unknownLegID
                                      }
        CODE                          opcode-splitLeg
        }
```

-- Direction: SCF $\rightarrow$ SSF, Timer: T_{sl}
-- This operation is issued by the SCF to separate one joined leg from a multi-way connection
-- or to interrupt the bearer connection between the involved legs of a single two party Call segment.

```
SplitLegArg {PARAMETERS-BOUND : bound} ::= SEQUENCE {
        legToBeSplit                  [0] LegID,
        newCallSegment                [1] INTEGER (2..bound.&numOfCSs),
        extensions                    [2] SEQUENCE SIZE (1..bound.&numOfExtensions) OF
                                         ExtensionField {bound}                  OPTIONAL,
        ...
        }


statusReport {PARAMETERS-BOUND : bound}  OPERATION ::= {
        ARGUMENT                      StatusReportArg { bound}
        RETURN RESULT                 FALSE
        ALWAYS RESPONDS               FALSE
        CODE                          opcode-statusReport
        }
```

-- Direction: SSF $\rightarrow$ SCF, Timer: T_{srp}
-- This operation is used as a response to RequestFirstStatusMatchReport or
-- RequestEveryStatusChangeReport operations.

```
StatusReportArg {PARAMETERS-BOUND : bound} ::= SEQUENCE {
        resourceStatus                [0] ResourceStatus                         OPTIONAL,
        correlationID                 [1] CorrelationID { bound}                 OPTIONAL,
        resourceID                    [2] ResourceID { bound}                    OPTIONAL,
        extensions                    [3] SEQUENCE SIZE(1..bound.&numOfExtensions) OF
                                          ExtensionField {bound}                 OPTIONAL,
        reportCondition               [4] ReportCondition                        OPTIONAL,
        ...
        }
```

-- For correlationID, OPTIONAL denotes network-operator optional.
-- resourceID is required when the SSF sends a report as an answer to a previous request when the
-- correlationID was present.

```
tAnswer {PARAMETERS-BOUND : bound}  OPERATION ::= {
        ARGUMENT                        TAnswerArg { bound}
        RETURN RESULT                   FALSE
        ERRORS                          {missingCustomerRecord |
                                        missingParameter |
                                        parameterOutOfRange |
                                        unexpectedComponentSequence |
                                        unexpectedParameter |
                                        unexpectedDataValue |
                                        systemFailure |
                                        taskRefused
                                        }
        CODE                            opcode-tAnswer
        }
```

-- Direction: SSF → SCF, Timer: T_{ta}
-- This operation is used to indicate that the call is accepted and answered by terminating party
-- (e.g. terminating party goes off-hook, Q.931 Connect message received, ISDN-UP Answer message
-- received) (DP – T_Answer). For additional information on this operation, refer to Rec. Q.1224.

```
TAnswerArg {PARAMETERS-BOUND : bound} ::= SEQUENCE {
        dpSpecificCommonParameters      [0] DpSpecificCommonParameters { bound},
        calledPartyBusinessGroupID      [1] CalledPartyBusinessGroupID              OPTIONAL,
        calledPartySubaddress           [2] CalledPartySubaddress                   OPTIONAL,
        calledFacilityGroup             [3] FacilityGroup                           OPTIONAL,
        calledFacilityGroupMember       [4] FacilityGroupMember                     OPTIONAL,
        extensions                      [5] SEQUENCE SIZE(1..bound.&numOfExtensions) OF
                                            ExtensionField {bound}                  OPTIONAL,
        componentType                   [6] ComponentType                           OPTIONAL,
        component                       [7] Component                               OPTIONAL,
        componentCorrelationID          [8] ComponentCorrelationID                  OPTIONAL,
        ...
        }

tBusy {PARAMETERS-BOUND : bound} OPERATION ::= {
        ARGUMENT                        TBusyArg { bound}
        RETURN RESULT                   FALSE
        ERRORS                          {missingCustomerRecord |
                                        missingParameter |
                                        parameterOutOfRange |
                                        unexpectedComponentSequence |
                                        unexpectedParameter |
                                        unexpectedDataValue |
                                        systemFailure |
                                        taskRefused }
        CODE                            opcode-tBusy
        }
```

-- Direction: SSF → SCF, Timer: T_{tb}
-- This operation is used to indicate all resources in group busy (DP– TBusy).
-- For additional information on this operation, refer to Rec. Q.1224.

```
TBusyArg {PARAMETERS-BOUND : bound} ::= SEQUENCE {
        dpSpecificCommonParameters      [0] DpSpecificCommonParameters { bound},
        busyCause                       [1] Cause { bound}                          OPTIONAL,
        calledPartyBusinessGroupID      [2] CalledPartyBusinessGroupID              OPTIONAL,
        calledPartySubaddress           [3] CalledPartySubaddress                   OPTIONAL,
        originalCalledPartyID           [4] OriginalCalledPartyID { bound}          OPTIONAL,
        redirectingPartyID              [5] RedirectingPartyID { bound}             OPTIONAL,
        redirectionInformation          [6] RedirectionInformation                  OPTIONAL,
        routeList                   [7] RouteList { bound}                          OPTIONAL,
        travellingClassMark         [8] TravellingClassMark { bound}                OPTIONAL,
        extensions                  [9] SEQUENCE SIZE(1..bound.&numOfExtensions)    OF
                                        ExtensionField {bound}                      OPTIONAL,
        ...
        }
```

-- For the OPTIONAL parameters, refer to clause 17 for the trigger detection point processing rules
-- to specify when these parameters are included in the message.

```
tDisconnect {PARAMETERS-BOUND : bound} OPERATION ::={
        ARGUMENT                    TDisconnectArg { bound}
        RETURN RESULT               FALSE
        ERRORS                      {missingCustomerRecord |
                                    missingParameter |
                                    parameterOutOfRange |
                                    unexpectedComponentSequence |
                                    unexpectedParameter |
                                    unexpectedDataValue |
                                    systemFailure |
                                    taskRefused }
        CODE                        opcode-tDisconnect
        }
```

-- Direction: SSF → SCF, Timer: T_{td}
-- This operation is used for a disconnect indication (e.g. on-hook, Q.931 Disconnect message,
-- SS7 Release message) is received from the terminating party, or received from the originating party
-- via the originating half BCSM. (DP – T_Disconnect). For additional information on this operation,
-- refer to Rec. Q.1224.

```
TDisconnectArg {PARAMETERS-BOUND : bound} ::= SEQUENCE {
        dpSpecificCommonParameters  [0] DpSpecificCommonParameters { bound},
        calledPartyBusinessGroupID  [1] CalledPartyBusinessGroupID              OPTIONAL,
        calledPartySubaddress       [2] CalledPartySubaddress                   OPTIONAL,
        calledFacilityGroup         [3] FacilityGroup                           OPTIONAL,
        calledFacilityGroupMember   [4] FacilityGroupMember                     OPTIONAL,
        releaseCause                [5] Cause { bound}                          OPTIONAL,
        extensions                  [6] SEQUENCE SIZE(1..bound.&numOfExtensions) OF
                                        ExtensionField {bound}                  OPTIONAL,
        connectTime                 [7] Integer4                                OPTIONAL,
        componentType               [8] ComponentType                           OPTIONAL,
        component                   [9] Component                               OPTIONAL,
        componentCorrelationID      [10] ComponentCorrelationID                 OPTIONAL,
        ...
        }

termAttemptAuthorized {PARAMETERS-BOUND : bound} OPERATION ::= {
        ARGUMENT                    TermAttemptAuthorizedArg { bound}
        RETURN RESULT               FALSE
        ERRORS                      {missingCustomerRecord |
                                    missingParameter |
                                    parameterOutOfRange |
                                    unexpectedComponentSequence |
                                    unexpectedParameter |
                                    unexpectedDataValue |
                                    systemFailure |
                                    taskRefused }
        CODE                        opcode-termAttemptAuthorized
        }
```

```
-- Direction: SSF → SCF, Timer: Ttaa
-- This operation is used for indication of incoming call received from originating half BCSM and authority
-- to route call to a specified terminating resource (or group) verified. (DP  – Termination_Authorized).
-- For additional information on this operation, refer to Rec. Q.1224.

TermAttemptAuthorizedArg {PARAMETERS-BOUND : bound} ::= SEQUENCE {
        dpSpecificCommonParameters  [0] DpSpecificCommonParameters { bound},
        calledPartyBusinessGroupID  [1] CalledPartyBusinessGroupID               OPTIONAL,
        calledPartySubaddress       [2] CalledPartySubaddress                    OPTIONAL,
        callingPartyBusinessGroupID [3] CallingPartyBusinessGroupID              OPTIONAL,
        originalCalledPartyID       [4] OriginalCalledPartyID { bound}           OPTIONAL,
        redirectingPartyID          [5] RedirectingPartyID { bound}              OPTIONAL,
        redirectionInformation      [6] RedirectionInformation                   OPTIONAL,
        routeList                   [7] RouteList { bound}                       OPTIONAL,
        travellingClassMark         [8] TravellingClassMark { bound}             OPTIONAL,
        extensions                  [9] SEQUENCE SIZE(1..bound.&numOfExtensions) OF
                                        ExtensionField {bound}                   OPTIONAL,
        callingPartySubaddress      [10] CallingPartySubaddress                  OPTIONAL,
        ...
        }
terminationAttempt {PARAMETERS-BOUND : bound} OPERATION ::= {
        ARGUMENT                    TerminationAttemptArg { bound}
        RETURN RESULT               FALSE
        ERRORS                      {missingCustomerRecord |
                                    missingParameter |
                                    parameterOutOfRange |
                                    unexpectedComponentSequence |
                                    unexpectedParameter |
                                    unexpectedDataValue |
                                    systemFailure |
                                    taskRefused }
        CODE                        opcode-terminationAttempt
        }

-- Direction: SSF → SCF, Timer: Ttra
-- This operation is used for indication of a call termination attempt from the terminating half BCSM. (DP –
-- Termination_Attempt).
-- For additional information on this operation, refer to Rec. Q.1224.

TerminationAttemptArg {PARAMETERS-BOUND : bound} ::= SEQUENCE {
        dpSpecificCommonParameters  [0] DpSpecificCommonParameters { bound},
        calledPartyBusinessGroupID  [1] CalledPartyBusinessGroupID                  OPTIONAL,
        calledPartySubaddress       [2] CalledPartySubaddress                       OPTIONAL,
        callingPartyBusinessGroupID [3] CallingPartyBusinessGroupID                 OPTIONAL,
        callingPartySubaddress      [4] CallingPartySubaddress                      OPTIONAL,
        originalCalledPartyID       [5] OriginalCalledPartyID { bound}              OPTIONAL,
        redirectingPartyID          [6] RedirectingPartyID { bound}                 OPTIONAL,
        redirectionInformation      [7] RedirectionInformation                      OPTIONAL,
        routeList                   [8] RouteList { bound}                          OPTIONAL,
        travellingClassMark         [9] TravellingClassMark { bound}                OPTIONAL,
        extensions                  [10] SEQUENCE SIZE(1..bound.&numOfExtensions)   OF
                                         ExtensionField {bound}                     OPTIONAL,
        ...
        }

-- For the OPTIONAL parameters, refer to clause 17 for the trigger detection point processing rules
-- to specify when these parameters are included in the message.
```

```
tMidCall {PARAMETERS-BOUND : bound} OPERATION ::= {
        ARGUMENT                    MidCallArg { bound}
        RETURN RESULT               FALSE
        ERRORS                      {missingCustomerRecord |
                                    missingParameter |
                                    parameterOutOfRange |
                                    unexpectedComponentSequence |
                                    unexpectedParameter |
                                    unexpectedDataValue |
                                    systemFailure |
                                    taskRefused }
        CODE                        opcode-tMidCall
        }
```

-- Direction: SSF $\rightarrow$ SCF, Timer: T_{tmc}
-- This operation is used to indicate that a feature request is received from the terminating party (e.g. hook
-- flash, ISDN feature activation Q.931 HOLD or RETrieve message). (DP – T_Mid_Call).
-- For additional information on this operation, refer to Rec. Q.1224.

```
tNoAnswer {PARAMETERS-BOUND : bound} OPERATION ::= {
        ARGUMENT                    TNoAnswerArg { bound}
        RETURN RESULT               FALSE
        ERRORS                      {missingCustomerRecord |
                                    missingParameter |
                                    parameterOutOfRange |
                                    unexpectedComponentSequence |
                                    unexpectedParameter |
                                    unexpectedDataValue |
                                    systemFailure |
                                    taskRefused }
        CODE                        opcode-tNoAnswer
        }
```

-- Direction: SSF $\rightarrow$ SCF, Timer: T_{tna}
-- This operation is used to indicate that the terminating party does not answer within a specified duration.
-- (DP – T_No_Answer). For additional information on this operation, refer to Rec. Q.1224.

```
TNoAnswerArg {PARAMETERS-BOUND : bound} ::= SEQUENCE {
        dpSpecificCommonParameters  [0] DpSpecificCommonParameters { bound},
        calledPartyBusinessGroupID  [1] CalledPartyBusinessGroupID              OPTIONAL,
        calledPartySubaddress       [2] CalledPartySubaddress                   OPTIONAL,
        calledFacilityGroup         [3] FacilityGroup                           OPTIONAL,
        calledFacilityGroupMember   [4] FacilityGroupMember                     OPTIONAL,
        originalCalledPartyID       [5] OriginalCalledPartyID { bound}          OPTIONAL,
        redirectingPartyID          [6] RedirectingPartyID { bound}             OPTIONAL,
        redirectionInformation      [7] RedirectionInformation                  OPTIONAL,
        travellingClassMark         [8] TravellingClassMark { bound}            OPTIONAL,
        extensions                  [9] SEQUENCE SIZE(1..bound.&numOfExtensions) OF
                                        ExtensionField {bound}                  OPTIONAL,
        componentType               [10] ComponentType                          OPTIONAL,
        component                   [11] Component                              OPTIONAL,
        componentCorrelationID      [12] ComponentCorrelationID                 OPTIONAL,
        ...
        }
```

```
tSuspended {PARAMETERS-BOUND : bound}  OPERATION ::= {
        ARGUMENT                    TSuspendedArg { bound}
        RETURN RESULT               FALSE
        ERRORS                      {missingCustomerRecord |
                                    missingParameter |
                                    systemFailure |
                                    taskRefused |
                                    unexpectedComponentSequence |
                                    unexpectedDataValue |
                                    unexpectedParameter
                                    }
        CODE                        opcode-tSuspended
        }

-- Direction: SSF → SCF, Timer: Tts
-- This operation is issued by the SSF after detecting a valid trigger condition at the T_Suspended DP or  to
-- report a tSuspended event requested by the RequestReportBCSMEvent. For additional information on
-- this operation, refer to Rec. Q.1224.

TSuspendedArg {PARAMETERS-BOUND : bound} ::= SEQUENCE {
        dpSpecificCommonParameters  [0] DpSpecificCommonParameters { bound},
        legID                       [1] LegID                                       OPTIONAL,
        extensions                  [2] SEQUENCE SIZE(1..bound.&numOfExtensions)OF
                                        ExtensionField {bound}                      OPTIONAL,
        ...
        }
-- For the OPTIONAL parameters, refer to clause 17 for the trigger detection point processing rules
-- to specify when these parameters are included in the message.
-- Use for T/EDP-R is ffs.

END
```

Table 5-1 below lists all operation timers and the value range for each timer. The definitive value for each operation timer may be network-specific and has to be defined by the network operator.

NOTE – The following value ranges do apply for operation specific timers in INAP:

short:	1-10 seconds.
medium:	1-60 seconds.
long:	1 second-30 minutes.
ffs:	For Further Study.

Table 5-1/Q.1228 – Operation timers and their value range

Operation Name	Timer	Value range
ActivateServiceFiltering	T_{asf}	Medium
ActivityTest	T_{at}	Short
AnalysedInformation	T_{adi}	Short
AnalyseInformation	T_{ai}	Short
ApplyCharging	T_{ac}	Short
ApplyChargingReport	T_{acr}	Short
AssistRequestInstructions	T_{ari}	Short
AuthorizeTermination	T_{atr}	Short

Table 5-1/Q.1228 – Operation timers and their value range *(continued)*

Operation Name	Timer	Value range
CallGap	T_{cg}	Short
CallInformationReport	T_{cirp}	Short
CallInformationRequest	T_{cirq}	Short
Cancel	T_{can}	Short
CancelStatusReportRequest	T_{csr}	Not specified in IN CS-2
CollectedInformation	T_{cdi}	Short
CollectInformation	T_{ci}	Medium
Connect	T_{con}	Short
ConnectToResource	T_{ctr}	Short
Continue	T_{cue}	Short
ContinueWithArgument	T_{cwa}	Short
CreateCallSegmentAssociation	T_{csa}	Short
DisconnectForwardConnection	T_{dfc}	Short
DisconnectForwardConnectionWithArgument	T_{dfcwa}	Short
DisconnectLeg	T_{dl}	Short
EntityRelease	T_{er}	Short
EstablishTemporaryConnection	T_{etc}	Medium
EventNotificationCharging	T_{enc}	Short
EventReportBCSM	T_{erb}	Short
EventReportFacility	T_{erf}	Short
FacilitySelectedAndAvailable	T_{fs}	Short
FurnishChargingInformation	T_{fci}	Short
HoldCallInNetwork	T_{hcn}	Not specified in IN CS-2
InitialDP	T_{idp}	Short
InitiateCallAttempt	T_{ica}	Short
ManageTriggerData	T_{mtd}	Medium
MergeCallSegments	T_{mc}	Short
MoveCallSegments	T_{mcs}	Short
MoveLeg	T_{ml}	Short
Oabandon	T_{ob}	Short
Oanswer	T_{oa}	Short
OcalledPartyBusy	T_{ob}	Short
Odisconnect	T_{od}	Short
OmidCall	T_{omc}	Short

Table 5-1/Q.1228 – Operation timers and their value range *(concluded)*

Operation Name	Timer	Value range
OnoAnswer	T_{ona}	Short
OriginationAttempt	T_{ora}	Short
OriginationAttemptAuthorized	T_{oaa}	Short
Osuspended	T_{os}	Short
Reconnect	T_{re}	Short
ReleaseCall	T_{rc}	Short
ReportUTSI	T_{ru}	Short
RequestCurrentStatusReport	T_{rcs}	Not specified in IN CS-2
RequestEveryStatusChangeReport	T_{res}	Short
RequestFirstStatusMatchReport	T_{rfs}	Short
RequestNotificationChargingEvent	T_{rnc}	Short
RequestReportBCSMEvent	T_{rrb}	Short
RequestReportFacilityEvent	T_{rrfe}	Short
RequestReportUTSI	T_{rru}	Short
ResetTimer	T_{rt}	Short
RouteSelectFailure	T_{rsf}	Short
SelectFacility	T_{sf}	Short
SelectRoute	T_{sr}	Short
SendChargingInformation	T_{sci}	Short
SendFacilityInformation	T_{sfi}	Short
SendSTUI	T_{ss}	Short
ServiceFilteringResponse	T_{sfr}	Short
SplitLeg	T_{sl}	Short
StatusReport	T_{srp}	Not specified in IN CS-2
Tanswer	T_{ta}	Short
Tbusy	T_{tb}	Short
Tdisconnect	T_{td}	Short
TermAttemptAuthorized	T_{taa}	Short
TerminationAttempt	T_{tra}	Short
TmidCall	T_{tmc}	Short
TnoAnswer	T_{tna}	Short
Tsuspended	T_{ts}	Short

5.2 SSF/SCF packages, contracts and Application Contexts

5.2.1 Protocol overview

The **inCs2SsfToScfGeneric** contract expresses the form of the service in which the SSF, a ROS-object of class **ssf**, initiates the generic triggering approach contract. A ROS-object of class **scf** responds to this contract.

```
inCs2SsfToScfGeneric CONTRACT ::= {
-- dialogue initiated by SSF with InitialDP Operation
        INITIATOR CONSUMER OF    {exceptionInformPackage  {networkSpecificBoundSet} |
                                 scfActivationPackage {networkSpecificBoundSet} }
        RESPONDER CONSUMER OF    {activityTestPackage|
                                 assistConnectionEstablishmentPackage {networkSpecificBoundSet} |
                                 bcsmEventHandlingPackage {networkSpecificBoundSet} |
                                 billingPackage {networkSpecificBoundSet} |
                                 callHandlingPackage {networkSpecificBoundSet} |
                                 callReportPackage {networkSpecificBoundSet} |
                                 cancelPackage {networkSpecificBoundSet} |
                                 chargingEventHandlingPackage {networkSpecificBoundSet} |
                                 chargingPackage {networkSpecificBoundSet} |
                                 connectPackage {networkSpecificBoundSet} |
                                 cphResponsePackage  {networkSpecificBoundSet} |
                                 facilityIEHandlingPackage {networkSpecificBoundSet} |
                                 genericDisconnectResourcePackage {networkSpecificBoundSet} |
                                 nonAssistedConnectionEstablishmentPackage
                                 {networkSpecificBoundSet} |
                                 signallingControlPackage {networkSpecificBoundSet} |
                                 specializedResourceControlPackage {networkSpecificBoundSet} |
                                 scriptControlPackage {networkSpecificBoundSet} |
                                 messageControlPackage {networkSpecificBoundSet} |
                                 ssfCallProcessingPackage {networkSpecificBoundSet} |
                                 statusReportingPackage {networkSpecificBoundSet} |
                                 timerPackage {networkSpecificBoundSet} |
                                 trafficManagementPackage {networkSpecificBoundSet} |
                                 uSIHandlingPackage {networkSpecificBoundSet}
                                 scfCallInitiationPackage {networkSpecificBoundSet}
                                 }
        ID                       id-inCs2SsfToScfGeneric
        }
```

The **inCs2SsfToScfDpSpecific** contract expresses the form of the service in which the SSF, a ROS-object of class **ssf**, initiates the DP specific triggering approach contract. A ROS-object of class **scf** responds to this contract.

```
inCs2SsfToScfDpSpecific CONTRACT ::= {
-- dialogue initiated by SSF with DP Specific Initial Operations
        INITIATOR CONSUMER OF    {advancedBCPDPPackage {networkSpecificBoundSet} |
                                 basicBCPDPPackage  {networkSpecificBoundSet} |
                                 exceptionInformPackage {networkSpecificBoundSet} }
        RESPONDER CONSUMER OF    {activityTestPackage|
                                 assistConnectionEstablishmentPackage {networkSpecificBoundSet} |
                                 billingPackage {networkSpecificBoundSet} |
                                 callHandlingPackage {networkSpecificBoundSet} |
                                 callReportPackage {networkSpecificBoundSet} |
                                 cancelPackage {networkSpecificBoundSet} |
                                 chargingEventHandlingPackage {networkSpecificBoundSet} |
                                 chargingPackage {networkSpecificBoundSet} |
                                 connectPackage {networkSpecificBoundSet} |
                                 cphResponsePackage  {networkSpecificBoundSet} |
                                 dpSpecificEventHandlingPackage {networkSpecificBoundSet} |
                                 facilityIEHandlingPackage {networkSpecificBoundSet} |
                                 genericDisconnectResourcePackage {networkSpecificBoundSet} |
                                 nonAssistedConnectionEstablishmentPackage
                                 {networkSpecificBoundSet} |
                                 signallingControlPackage {networkSpecificBoundSet} |
                                 specializedResourceControlPackage {networkSpecificBoundSet} |
                                 scriptControlPackage {networkSpecificBoundSet} |
                                 messageControlPackage {networkSpecificBoundSet} |
                                 ssfCallProcessingPackage {networkSpecificBoundSet} |
                                 statusReportingPackage {networkSpecificBoundSet} |
                                 timerPackage {networkSpecificBoundSet} |
                                 trafficManagementPackage {networkSpecificBoundSet} |
                                 uSIHandlingPackage {networkSpecificBoundSet}
                                 scfCallInitiationPackage {networkSpecificBoundSet}
                                 }
        ID                              id-inCs2SsfToScfDpSpecific
        }
```

The **inCs2AssistHandoffSsfToScf** contract expresses the form of the service in which the SSF, a ROS-object of class **ssf**, initiates the Assist or Hand-off contract. A ROS-object of class **scf** responds to this contract.

```
inCs2AssistHandoffSsfToScf CONTRACT ::= {
-- dialogue initiated by SSF with AssistRequestInstructions
    INITIATOR CONSUMER OF       {srf-scfActivationOfAssistPackage {networkSpecificBoundSet} }
    RESPONDER CONSUMER OF       {activityTestPackage|
                                    billingPackage {networkSpecificBoundSet} |
                                    callHandlingPackage {networkSpecificBoundSet} |
                                    cancelPackage {networkSpecificBoundSet} |
                                    chargingPackage {networkSpecificBoundSet} |
                                    genericDisconnectResourcePackage {networkSpecificBoundSet} |
                                    nonAssistedConnectionEstablishmentPackage
                                    {networkSpecificBoundSet} |
                                    specializedResourceControlPackage {networkSpecificBoundSet} |
                                    scriptControlPackage {networkSpecificBoundSet} |
                                    messageControlPackage {networkSpecificBoundSet} |
                                    statusReportingPackage {networkSpecificBoundSet} |
                                    timerPackage {networkSpecificBoundSet}
                                    }
    ID                              id-inCs2AssistHandoffSsfToScf
    }
```

The **inCs2ScfToSsfGeneric** contract expresses the form of the service in which the SCF, a ROS-object of class **scf**, initiates the generic messaging approach for the SCF Initiate Call Attempt contract. A ROS-object of class **ssf** responds to this contract.

```
inCs2ScfToSsfGeneric CONTRACT ::= {
-- dialogue initiated by SCF with InitiateCallAttempt, Generic Case
    INITIATOR CONSUMER OF       {activityTestPackage|
                                    assistConnectionEstablishmentPackage {networkSpecificBoundSet} |
                                    bcsmEventHandlingPackage {networkSpecificBoundSet} |
                                    billingPackage {networkSpecificBoundSet} |
                                    callHandlingPackage {networkSpecificBoundSet} |
                                    callReportPackage {networkSpecificBoundSet} |
                                    cancelPackage {networkSpecificBoundSet} |
                                    chargingPackage {networkSpecificBoundSet} |
                                    connectPackage {networkSpecificBoundSet} |
                                    cphResponsePackage  {networkSpecificBoundSet} |
                                    facilityIEHandlingPackage {networkSpecificBoundSet} |
                                    genericDisconnectResourcePackage {networkSpecificBoundSet} |
                                    nonAssistedConnectionEstablishmentPackage
                                    {networkSpecificBoundSet} |
                                    scfCallInitiationPackage {networkSpecificBoundSet} |
                                    signallingControlPackage {networkSpecificBoundSet} |
                                    specializedResourceControlPackage {networkSpecificBoundSet} |
                                    scriptControlPackage {networkSpecificBoundSet} |
                                    messageControlPackage {networkSpecificBoundSet} |
                                    ssfCallProcessingPackage {networkSpecificBoundSet} |
                                    statusReportingPackage {networkSpecificBoundSet} |
                                    timerPackage {networkSpecificBoundSet} |
                                    uSIHandlingPackage {networkSpecificBoundSet}
                                    }
    RESPONDER CONSUMER OF       {exceptionInformPackage {networkSpecificBoundSet} }
    ID                          id-inCs2ScfToSsfGeneric
    }
```

The **inCs2ScfToSsfDpSpecific** contract expresses the form of the service in which the SCF, a ROS-object of class **scf**, initiates the DP specific messaging approach for the SCF Initiate Call Attempt contract. A ROS-object of class **ssf** responds to this contract.

```
inCs2ScfToSsfDpSpecific CONTRACT ::= {
-- dialogue initiated by SCF with InitiateCallAttempt, DP Specific Case
    INITIATOR CONSUMER OF       {activityTestPackage|
                                assistConnectionEstablishmentPackage {networkSpecificBoundSet} |
                                billingPackage {networkSpecificBoundSet} |
                                callHandlingPackage {networkSpecificBoundSet} |
                                callReportPackage {networkSpecificBoundSet} |
                                cancelPackage {networkSpecificBoundSet} |
                                chargingEventHandlingPackage {networkSpecificBoundSet} |
                                chargingPackage {networkSpecificBoundSet} |
                                connectPackage {networkSpecificBoundSet} |
                                cphResponsePackage  {networkSpecificBoundSet} |
                                dpSpecificEventHandlingPackage {networkSpecificBoundSet} |
                                facilityIEHandlingPackage {networkSpecificBoundSet} |
                                genericDisconnectResourcePackage {networkSpecificBoundSet} |
                                nonAssistedConnectionEstablishmentPackage
                                {networkSpecificBoundSet} |
                                scfCallInitiationPackage {networkSpecificBoundSet} |
                                signallingControlPackage {networkSpecificBoundSet} |
                                specializedResourceControlPackage {networkSpecificBoundSet} |
                                scriptControlPackage {networkSpecificBoundSet} |
                                messageControlPackage {networkSpecificBoundSet} |
                                ssfCallProcessingPackage {networkSpecificBoundSet} |
                                statusReportingPackage {networkSpecificBoundSet} |
                                timerPackage {networkSpecificBoundSet} |
                                uSIHandlingPackage {networkSpecificBoundSet}
                                }
    RESPONDER CONSUMER OF       {exceptionInformPackage {networkSpecificBoundSet} }
    ID                          id-inCs2ScfToSsfDpSpecific
    }
```

The **inCs2ScfToSsfTrafficManagement** contract expresses the form of the service in which the SCF, a ROS-object of class **scf**, initiates the Traffic Management related contract. A ROS-object of class **ssf** responds to this contract.

```
inCs2ScfToSsfTrafficManagement CONTRACT ::= {
-- dialogue initiated by SCF with CallGap
        INITIATOR CONSUMER OF    {trafficManagementPackage {networkSpecificBoundSet}
                                 }
        ID                       id-inCs2ScfToSsfTrafficManagement
        }
```

The **inCs2ScfToSsfServiceManagement** contract expresses the form of the service in which the SCF, a ROS-object of class **scf**, initiates the Service Management related contract. A ROS-object of class **ssf**, in the context of a separate contract, responds to this initiation.

```
inCs2ScfToSsfTriggerManagement CONTRACT ::= {
-- dialogue initiated by SCF with Manage Trigger Data
        INITIATOR CONSUMER OF    {triggerManagementPackage {networkSpecificBoundSet}
                                 }
        ID                       id-inCs2ScfToSsfTriggerManagement
        }
```

The **inCs2SsfToScfServiceManagement** expresses the form of the service in which the SSF, a ROS-object of class **ssf**, initiates the Service Management related contract for reporting Service Management results.

```
inCs2SsfToScfServiceManagement CONTRACT ::= {
-- dialogue initiated/ended by SSF with ServiceFilteringResponse
        INITIATOR CONSUMER OF    {serviceManagementResponsePackage {networkSpecificBoundSet}
                                 }
        ID                       id-inCs2SsfToScfServiceManagement
        }
```

The **inCs2ScfToSsfStatusReporting** contract expresses the form of the service in which the SCF, a ROS-object of class **scf**, initiates the Status Reporting related contract. A ROS-object of class **ssf**, responds to this contract.

```
inCs2ScfToSsfStatusReporting CONTRACT ::= {
-- dialogue initiated by SCF with StatusReporting Operations
        INITIATOR CONSUMER OF    {cancelPackage {networkSpecificBoundSet} |
                                 statusReportingPackage {networkSpecificBoundSet}
                                 }
        ID                       id-inCs2ScfToSsfStatusReporting
        }
```

The **inCs2ScfToSsfTriggerManagement** contract expresses the form of the service in which the SCF, a ROS-object of class **scf**, initiates the Trigger Management related contract. A ROS-object of class **ssf**, in the context of a separate contract, responds to this initiation.

```
inCs2ScfToSsfTriggerManagement CONTRACT ::= {
-- dialogue initiated by SCF with Manage Trigger Data
        INITIATOR CONSUMER OF    {triggerManagementPackage {networkSpecificBoundSet}
                                 }
        ID                       id-inCs2ScfToSsfTriggerManagement
        }
```

SSF/SCF operation packages

The operation packages below are defined as information objects of class OPERATION-PACKAGE. The operations of these packages are defined in 5.1.

```
scfActivationPackage  {PARAMETERS-BOUND : bound}   OPERATION-PACKAGE ::= {
        CONSUMER INVOKES            {initialDP {bound}}
        ID                          id-package-scfActivation}

basicBCPDPPackage  {PARAMETERS-BOUND : bound}   OPERATION-PACKAGE ::= {
        CONSUMER INVOKES            {originationAttemptAuthorized {bound}|
                                    collectedInformation {bound}|
                                    analysedInformation {bound}| routeSelectFailure  {bound}|
                                    facilitySelectedAndAvailable  {bound}|
                                    oAbandon  {bound}| originationAttempt  {bound} |
                                    terminationAttempt  {bound} |
                                    oCalledPartyBusy  {bound} | oNoAnswer  {bound} |
                                    oAnswer {bound} |
                                    oDisconnect {bound} | termAttemptAuthorized {bound} |
                                    tBusy {bound} |
                                    tNoAnswer {bound} | tAnswer {bound} | tDisconnect {bound} }
        ID                          id-package-basicBCPDP}

advancedBCPDPPackage  {PARAMETERS-BOUND : bound}    OPERATION-PACKAGE ::= {
        CONSUMER INVOKES            {oMidCall {bound} | oSuspended {bound} |
                                    tMidCall {bound} | tSuspended{bound} }
        ID                          id-package-advancedBCPDP}

srf-scfActivationOfAssistPackage  {PARAMETERS-BOUND : bound}  OPERATION-PACKAGE ::= {
        CONSUMER INVOKES            {assistRequestInstructions {bound}}
        ID                          id-package-srf-scfActivationOfAssist}

assistConnectionEstablishmentPackage  {PARAMETERS-BOUND : bound}     OPERATION-PACKAGE ::= {
        CONSUMER INVOKES            {establishTemporaryConnection {bound}}
        ID                          id-package-assistConnectionEstablishment}

genericDisconnectResourcePackage  {PARAMETERS-BOUND : bound}     OPERATION-PACKAGE ::= {
        CONSUMER INVOKES            {disconnectForwardConnection |
                                    disconnectForwardConnectionWithArgument {bound}}
        ID                          id-package-genericDisconnectResource}

nonAssistedConnectionEstablishmentPackage  {PARAMETERS-BOUND : bound}  OPERATION-PACKAGE ::= {
        CONSUMER INVOKES            {connectToResource {bound}}
        ID                          id-package-nonAssistedConnectionEstablishment}

connectPackage  {PARAMETERS-BOUND : bound}  OPERATION-PACKAGE ::= {
        CONSUMER INVOKES            {connect {bound}}
        ID                          id-package-connect}

callHandlingPackage   {PARAMETERS-BOUND : bound} OPERATION-PACKAGE ::= {
        CONSUMER INVOKES            {holdCallInNetwork | releaseCall {bound}}
        ID                          id-package-callHandling}

bcsmEventHandlingPackage  {PARAMETERS-BOUND : bound}  OPERATION-PACKAGE ::= {
        CONSUMER INVOKES            {requestReportBCSMEvent {bound}}
        SUPPLIER INVOKES            {eventReportBCSM {bound}}
        ID                          id-package-bcsmEventHandling}

dpSpecificEventHandlingPackage  {PARAMETERS-BOUND : bound}     OPERATION-PACKAGE ::= {
        CONSUMER INVOKES            {requestReportBCSMEvent {bound}}
        SUPPLIER INVOKES            {originationAttemptAuthorized {bound} |
                                    collectedInformation {bound} |
                                    analysedInformation  {bound} | routeSelectFailure  {bound} |
                                    facilitySelectedAndAvailable  {bound} |
                                    oAbandon  {bound} | originationAttempt  {bound} |
                                    terminationAttempt  {bound} |
                                        oCalledPartyBusy  {bound} | oNoAnswer  {bound} |
                                        oAnswer  {bound} |
                                        oDisconnect  {bound} | termAttemptAuthorized  {bound} |
                                        tBusy  {bound} |
                                        tNoAnswer  {bound} | tAnswer  {bound} | tDisconnect  {bound} |
                                        oMidCall  {bound} | oSuspended  {bound} |
                                        tMidCall  {bound} | tSuspended {bound}
                                        }
     ID                                 id-package-dpSpecificEventHandling}

chargingEventHandlingPackage  {PARAMETERS-BOUND : bound} OPERATION-PACKAGE ::= {
     CONSUMER INVOKES                   {requestNotificationChargingEvent {bound}}
     SUPPLIER INVOKES                   {eventNotificationCharging {bound}}
     ID                                 id-package-chargingEventHandling}

ssfCallProcessingPackage {PARAMETERS-BOUND : bound}  OPERATION-PACKAGE ::= {
     CONSUMER INVOKES                   {collectInformation  {bound} | analyseInformation  {bound}|
                                        authorizeTermination  {bound}| selectRoute  {bound}|
                                        selectFacility  {bound}| continue}
     ID                                 id-package-ssfCallProcessing}

scfCallInitiationPackage {PARAMETERS-BOUND : bound} OPERATION-PACKAGE ::= {
     CONSUMER INVOKES                   {initiateCallAttempt {bound}}
     ID                                 id-package-scfCallInitiation}

timerPackage {PARAMETERS-BOUND : bound} OPERATION-PACKAGE ::= {
     CONSUMER INVOKES                   {resetTimer {bound}}
     ID                                 id-package-timer}

billingPackage {PARAMETERS-BOUND : bound} OPERATION-PACKAGE ::= {
     CONSUMER INVOKES                   {furnishChargingInformation {bound}}
     ID                                 id-package-billing}

chargingPackage {PARAMETERS-BOUND : bound} OPERATION-PACKAGE ::= {
     CONSUMER INVOKES                   {applyCharging {bound}}
     SUPPLIER INVOKES                   {applyChargingReport {bound}}
     ID                                 id-package-charging}

trafficManagementPackage {PARAMETERS-BOUND : bound} OPERATION-PACKAGE ::= {
     CONSUMER INVOKES                   {callGap {bound}}
     ID                                 id-package-trafficManagement}

serviceManagementActivatePackage {PARAMETERS-BOUND : bound} OPERATION-PACKAGE ::= {
     CONSUMER INVOKES                   {activateServiceFiltering {bound}}
     ID                                 id-package-serviceManagementActivate}

serviceManagementResponsePackage {PARAMETERS-BOUND : bound} OPERATION-PACKAGE ::= {
     CONSUMER INVOKES                   {serviceFilteringResponse {bound}}
     ID                                 id-package-serviceManagementResponse}

callReportPackage {PARAMETERS-BOUND : bound} OPERATION-PACKAGE ::= {
     CONSUMER INVOKES                   {callInformationRequest {bound}}
     SUPPLIER INVOKES                   {callInformationReport {bound}}
     ID                                 id-package-callReport}

signallingControlPackage {PARAMETERS-BOUND : bound} OPERATION-PACKAGE ::= {
     CONSUMER INVOKES                   {sendChargingInformation {bound}}
     ID                                 id-package-signallingControl}
```

```
activityTestPackage  OPERATION-PACKAGE ::= {
        CONSUMER INVOKES            {activityTest}
        ID                          id-package-activityTest}

statusReportingPackage {PARAMETERS-BOUND : bound} OPERATION-PACKAGE ::= {
        CONSUMER INVOKES            {requestCurrentStatusReport  {bound}|
                                    requestEveryStatusChangeReport {bound}|
                                    requestFirstStatusMatchReport {bound}}
        SUPPLIER INVOKES            {statusReport {bound}}
        ID                          id-package-statusReporting}

cancelPackage {PARAMETERS-BOUND : bound} OPERATION-PACKAGE ::= {
        CONSUMER INVOKES            {cancel {bound}| cancelStatusReportRequest {bound}}
        ID                          id-package-cancel}

cphResponsePackage  {PARAMETERS-BOUND : bound} OPERATION-PACKAGE ::= {
        CONSUMER INVOKES            {
                                    continueWithArgument {bound}| disconnectLeg  {bound}|
                                    mergeCallSegments  {bound}|
                                    moveCallSegments  {bound}|
                                    moveLeg  {bound}|
                                    createCallSegmentAssociation  {bound} |
                                    reconnect  {bound}|
                                    splitLeg {bound}
                                    }
        ID                          id-package-cphResponse}

exceptionInformPackage OPERATION-PACKAGE ::= {
        CONSUMER INVOKES            {entityReleased}
        ID                          id-package-entityReleased}

triggerManagementPackage   {PARAMETERS-BOUND : bound} OPERATION-PACKAGE ::= {
        CONSUMER INVOKES            {manageTriggerData {bound}}
        ID                          id-package-triggerManagement}

uSIHandlingPackage OPERATION-PACKAGE ::= {
        CONSUMER INVOKES            {requestReportUTSI | sendSTUI}
        SUPPLIER INVOKES            {reportUTSI}
        ID                          id-package-uSIHandling
        }

facilityIEHandlingPackage OPERATION-PACKAGE ::= {
        CONSUMER INVOKES            {requestReportFacilityEvent | sendFacilityInformation}
        SUPPLIER INVOKES            {eventReportFacility}
        ID                          id-package-facilityIEHandling
        }
```

Abstract syntax

This version of the INAP requires the support of two abstract syntaxes:

a) the abstract syntax of TC dialogue control protocol data units, **dialogue-abstract-syntax**, which is needed to establish the dialogues between FEs and specified in Recommendation Q.773;

b) the abstract syntax for conveying the protocol data units for invoking the operations involved in the operation packages specified in 5.2.2 and reporting their outcome.

The ASN.1 type from which the values of the last abstract syntax are derived is specified using the parameterized type **TCMessage {}** defined in Recommendation Q.773.

All these abstract syntaxes shall (as a minimum) be encoded according to the Basic ASN.1 encoding rules with the restrictions listed in Recommendation Q.773.

The SSF-SCF INAP Packages that realize the operation packages specified as above share the following abstract syntaxes. These are specified as information objects of the class ABSTRACT-SYNTAX.

```
ssf-scfGenericAbstractSyntax  ABSTRACT-SYNTAX ::= {
        GenericSSF-SCF-PDUs
        IDENTIFIED BY                  id-as-ssf-scfGenericAS}

GenericSSF-SCF-PDUs ::= TCMessage {{SsfToScfGenericInvokable},
                                   {SsfToScfGenericReturnable}}

SsfScfGenericInvokable  OPERATION ::= {
                                   activateServiceFiltering  {networkSpecificBoundSet} |
                                   activityTest |
                                   applyCharging  {networkSpecificBoundSet} |
                                   applyChargingReport  {networkSpecificBoundSet} |
                                   callInformationReport  {networkSpecificBoundSet} |
                                   callInformationRequest  {networkSpecificBoundSet} |
                                   cancel  {networkSpecificBoundSet} |
                                   cancelStatusReportRequest  {networkSpecificBoundSet} |
                                   collectInformation  {networkSpecificBoundSet} |
                                   connect  {networkSpecificBoundSet} | connectToResource
                                   {networkSpecificBoundSet} |
                                   disconnectForwardConnection |
                                   disconnectForwardConnectionWithArgument
                                   {networkSpecificBoundSet}|
                                   disconnectLeg {networkSpecificBoundSet} |
                                   entityReleased {networkSpecificBoundSet} |
                                   establishTemporaryConnection {networkSpecificBoundSet} |
                                   eventNotificationCharging  {networkSpecificBoundSet} |
                                   eventReportBCSM  {networkSpecificBoundSet} |
                                   eventReportFacility {networkSpecificBoundSet} |
                                   furnishChargingInformation  {networkSpecificBoundSet} |
                                   holdCallInNetwork |
                                   initialDP  {networkSpecificBoundSet} |
                                   mergeCallSegments {networkSpecificBoundSet} |
                                   moveCallSegments {networkSpecificBoundSet} |
                                   moveLeg {networkSpecificBoundSet} |
                                   createCallSegmentAssociation {networkSpecificBoundSet} |
                                   reconnect {networkSpecificBoundSet} |
                                   releaseCall {networkSpecificBoundSet} |
                                   reportUTSI {networkSpecificBoundSet} |
                                   requestCurrentStatusReport  {networkSpecificBoundSet} |
                                   requestEveryStatusChangeReport {networkSpecificBoundSet} |
                                   requestFirstStatusMatchReport  {networkSpecificBoundSet} |
                                   requestNotificationChargingEvent  {networkSpecificBoundSet} |
                                   requestReportBCSMEvent  {networkSpecificBoundSet} |
                                   requestReportFacilityEvent {networkSpecificBoundSet} |
                                   requestReportUTSI {networkSpecificBoundSet} |
                                   resetTimer  {networkSpecificBoundSet} |
                                   sendChargingInformation  {networkSpecificBoundSet} |
                                   sendFacilityInformation {networkSpecificBoundSet} |
                                   sendSTUI {networkSpecificBoundSet} |
                                   serviceFilteringResponse  {networkSpecificBoundSet} |
                                   splitLeg {networkSpecificBoundSet} |
                                   statusReport  {networkSpecificBoundSet} |
                                   playAnnouncement  {networkSpecificBoundSet} |
                                        promptAndCollectUserInformation  {networkSpecificBoundSet} |
                                        scriptClose  {networkSpecificBoundSet} |
                                        scriptEvent  {networkSpecificBoundSet} |
                                        scriptInformation  {networkSpecificBoundSet} |
                                        scriptRun  {networkSpecificBoundSet} |
                                        specializedResourceReport |
                                        promptAndReceiveMessage  {networkSpecificBoundSet}
                                        }

SsfScfGenericReturnable  OPERATION ::= {
                                        activateServiceFiltering  {networkSpecificBoundSet} |
                                        activityTest |
                                        applyCharging  {networkSpecificBoundSet} |
                                        applyChargingReport  {networkSpecificBoundSet} |
                                        callGap  {networkSpecificBoundSet} |
                                        callInformationRequest  {networkSpecificBoundSet} |
                                        cancel  {networkSpecificBoundSet} |
                                        cancelStatusReportRequest  {networkSpecificBoundSet} |
                                        collectInformation  {networkSpecificBoundSet} |
                                        connect  {networkSpecificBoundSet} |
                                        connectToResource  {networkSpecificBoundSet} |
                                        continue |
                                        continueWithArgument  {networkSpecificBoundSet}|
                                        disconnectForwardConnection |
                                        disconnectForwardConnectionWithArgument
                                        {networkSpecificBoundSet}|
                                        disconnectLeg {networkSpecificBoundSet}|
                                        establishTemporaryConnection {networkSpecificBoundSet}|
                                        furnishChargingInformation  {networkSpecificBoundSet}|
                                        holdCallInNetwork |
                                        initialDP  {networkSpecificBoundSet}|
                                        mergeCallSegments {networkSpecificBoundSet}|
                                        moveCallSegments {networkSpecificBoundSet}|
                                        moveLeg {networkSpecificBoundSet}|
                                        createCallSegmentAssociation {networkSpecificBoundSet}|
                                        reconnect {networkSpecificBoundSet}|
                                        releaseCall {networkSpecificBoundSet}|
                                        requestCurrentStatusReport  {networkSpecificBoundSet}|
                                        requestEveryStatusChangeReport {networkSpecificBoundSet}|
                                        requestFirstStatusMatchReport  {networkSpecificBoundSet}|
                                        requestNotificationChargingEvent  {networkSpecificBoundSet}|
                                        requestReportBCSMEvent  {networkSpecificBoundSet}|
                                        requestReportFacilityEvent {networkSpecificBoundSet}|
                                        requestReportUTSI {networkSpecificBoundSet}|
                                        resetTimer  {networkSpecificBoundSet}|
                                        sendChargingInformation {networkSpecificBoundSet}|
                                        sendFacilityInformation {networkSpecificBoundSet}|
                                        sendSTUI {networkSpecificBoundSet}|
                                        splitLeg {networkSpecificBoundSet}|
                                        playAnnouncement  {networkSpecificBoundSet}|
                                        promptAndCollectUserInformation  {networkSpecificBoundSet}|
                                        scriptClose  {networkSpecificBoundSet}|
                                        scriptInformation  {networkSpecificBoundSet}|
                                        scriptRun  {networkSpecificBoundSet}|
                                        promptAndReceiveMessage  {networkSpecificBoundSet}
                                        }

ssf-scfDpSpecificAbstractSyntax  ABSTRACT-SYNTAX ::= {
        DpSpecificSSF-SCF-PDUs
        IDENTIFIED BY                   id-as-ssf-scfDpSpecificAS}
```

```
DpSpecificSSF-SCF-PDUs ::= TCMessage {{SsfToScfDpSpecificInvokable},
                                      {SsfToScfDpSpecificReturnable}}

SsfToScfDpSpecificInvokable  OPERATION ::= {
                                          activateServiceFiltering  {networkSpecificBoundSet}|
                                          activityTest |
                                          analyseInformation  {networkSpecificBoundSet}|
                                          analysedInformation  {networkSpecificBoundSet}|
                                          applyCharging  {networkSpecificBoundSet}|
                                          applyChargingReport  {networkSpecificBoundSet}|
                                          assistRequestInstructions  {networkSpecificBoundSet}|
                                          callInformationReport  {networkSpecificBoundSet}|
                                          callInformationRequest  {networkSpecificBoundSet}|
                                          cancel  {networkSpecificBoundSet}|
                                          cancelStatusReportRequest  {networkSpecificBoundSet}|
                                          collectedInformation  {networkSpecificBoundSet}|
                                          collectInformation  {networkSpecificBoundSet}|
                                          connect  {networkSpecificBoundSet}|
                                          connectToResource  {networkSpecificBoundSet}|
                                          disconnectForwardConnection |
                                          disconnectForwardConnectionWithArgument
                                          {networkSpecificBoundSet}|
                                          disconnectLeg {networkSpecificBoundSet}|
                                          entityReleased {networkSpecificBoundSet}|
                                          establishTemporaryConnection {networkSpecificBoundSet}|
                                          eventNotificationCharging  {networkSpecificBoundSet}|
                                          eventReportFacility {networkSpecificBoundSet}|
                                          furnishChargingInformation  {networkSpecificBoundSet}|
                                          holdCallInNetwork |
                                          initiateCallAttempt {networkSpecificBoundSet}|
                                          mergeCallSegments {networkSpecificBoundSet}|
                                          moveCallSegments {networkSpecificBoundSet}|
                                          moveLeg {networkSpecificBoundSet}|
                                          oAbandon  {networkSpecificBoundSet}|
                                          oAnswer  {networkSpecificBoundSet}|
                                          oCalledPartyBusy  {networkSpecificBoundSet}|
                                          oDisconnect  {networkSpecificBoundSet}|
                                          oMidCall  {networkSpecificBoundSet}|
                                          oNoAnswer  {networkSpecificBoundSet}|
                                          createCallSegmentAssociation {networkSpecificBoundSet}|
                                          originationAttemptAuthorized  {networkSpecificBoundSet}|
                                          reconnect {networkSpecificBoundSet}|
                                          releaseCall {networkSpecificBoundSet} |
                                          reportUTSI {networkSpecificBoundSet}|
                                          requestCurrentStatusReport  {networkSpecificBoundSet}|
                                          requestEveryStatusChangeReport {networkSpecificBoundSet}|
                                          requestFirstStatusMatchReport  {networkSpecificBoundSet}|
                                          requestNotificationChargingEvent  {networkSpecificBoundSet}|
                                          requestReportBCSMEvent  {networkSpecificBoundSet}|
                                          requestReportFacilityEvent {networkSpecificBoundSet}|
                                          requestReportUTSI {networkSpecificBoundSet}|
                                          resetTimer  {networkSpecificBoundSet}|
                                          routeSelectFailure  {networkSpecificBoundSet}|
                                          selectFacility  {networkSpecificBoundSet}|
                                          selectRoute  {networkSpecificBoundSet}|
                                          sendChargingInformation  {networkSpecificBoundSet}|
                                          sendFacilityInformation {networkSpecificBoundSet}|
                                          sendSTUI {networkSpecificBoundSet}|
                                          serviceFilteringResponse  {networkSpecificBoundSet}|
                                          splitLeg {networkSpecificBoundSet}|
                                          statusReport  {networkSpecificBoundSet}|
                                          tAnswer  {networkSpecificBoundSet}|
                                          tBusy  {networkSpecificBoundSet}|
                                          tDisconnect  {networkSpecificBoundSet} |
                                          termAttemptAuthorized  {networkSpecificBoundSet}|
                                          tMidCall  {networkSpecificBoundSet}|
                                          tNoAnswer  {networkSpecificBoundSet} |
                                          playAnnouncement  {networkSpecificBoundSet}|
                                          promptAndCollectUserInformation  {networkSpecificBoundSet}|
                                          scriptClose  {networkSpecificBoundSet}|
                                          scriptEvent  {networkSpecificBoundSet}|
                                          scriptInformation  {networkSpecificBoundSet}|
                                          scriptRun  {networkSpecificBoundSet}|
                                          specializedResourceReport |
                                          promptAndReceiveMessage  {networkSpecificBoundSet}
                                          }

  SsfToScfDpSpecificReturnable  OPERATION ::= {
                                          activateServiceFiltering  {networkSpecificBoundSet}|
                                          activityTest |
                                          analyseInformation  {networkSpecificBoundSet}|
                                          analysedInformation  {networkSpecificBoundSet}|
                                          applyCharging  {networkSpecificBoundSet}|
                                          applyChargingReport  {networkSpecificBoundSet}|
                                          assistRequestInstructions  {networkSpecificBoundSet}|
                                          callGap  {networkSpecificBoundSet}|
                                          callInformationRequest  {networkSpecificBoundSet} |
                                          cancel  {networkSpecificBoundSet} |
                                          cancelStatusReportRequest {networkSpecificBoundSet} |
                                          collectedInformation  {networkSpecificBoundSet} |
                                          collectInformation  {networkSpecificBoundSet} |
                                          connect  {networkSpecificBoundSet} |
                                          connectToResource  {networkSpecificBoundSet} |
                                          continue |
                                          continueWithArgument {networkSpecificBoundSet} |
                                          disconnectForwardConnection |
                                          disconnectForwardConnectionWithArgument
                                          {networkSpecificBoundSet} |
                                          disconnectLeg {networkSpecificBoundSet} |
                                          establishTemporaryConnection {networkSpecificBoundSet} |
                                          furnishChargingInformation {networkSpecificBoundSet} |
                                          holdCallInNetwork |
                                          initiateCallAttempt  {networkSpecificBoundSet}|
                                          mergeCallSegments {networkSpecificBoundSet}|
                                          moveCallSegments {networkSpecificBoundSet}|
                                          moveLeg {networkSpecificBoundSet}|
                                          oAbandon  {networkSpecificBoundSet}|
                                          oAnswer  {networkSpecificBoundSet}|
                                          oCalledPartyBusy  {networkSpecificBoundSet}|
                                          oDisconnect  {networkSpecificBoundSet}|
                                          oMidCall  {networkSpecificBoundSet}|
                                          oNoAnswer  {networkSpecificBoundSet}|
                                          createCallSegmentAssociation  {networkSpecificBoundSet}|
                                          originationAttemptAuthorized  {networkSpecificBoundSet}|
                                          reconnect {networkSpecificBoundSet}|
                                          releaseCall {networkSpecificBoundSet}|
                                          requestCurrentStatusReport  {networkSpecificBoundSet}|
                                          requestEveryStatusChangeReport {networkSpecificBoundSet}|
                                          requestFirstStatusMatchReport  {networkSpecificBoundSet}|
                                          requestNotificationChargingEvent  {networkSpecificBoundSet}|
                                          requestReportBCSMEvent  {networkSpecificBoundSet}|
                                          requestReportFacilityEvent {networkSpecificBoundSet}|
                                          requestReportUTSI {networkSpecificBoundSet}|
                                          resetTimer  {networkSpecificBoundSet}|
                                          routeSelectFailure  {networkSpecificBoundSet}|
                                          selectFacility  {networkSpecificBoundSet}|
                                          selectRoute  {networkSpecificBoundSet}|
                                          sendChargingInformation {networkSpecificBoundSet}|
                                          sendFacilityInformation {networkSpecificBoundSet}|
                                          sendSTUI {networkSpecificBoundSet}|
                                          splitLeg {networkSpecificBoundSet}|
                                          tAnswer  {networkSpecificBoundSet}|
                                          tBusy  {networkSpecificBoundSet}|
                                          tDisconnect  {networkSpecificBoundSet}|
                                          termAttemptAuthorized  {networkSpecificBoundSet}|
                                          tMidCall  {networkSpecificBoundSet}|
                                          tNoAnswer  {networkSpecificBoundSet}|
                                          playAnnouncement  {networkSpecificBoundSet}|
                                          promptAndCollectUserInformation  {networkSpecificBoundSet}|
                                          scriptClose  {networkSpecificBoundSet}|
                                          scriptInformation  {networkSpecificBoundSet}|
                                          scriptRun  {networkSpecificBoundSet}|
                                          promptAndReceiveMessage  {networkSpecificBoundSet}
                                          }

  assistHandoff-ssf-scfAbstractSyntax  ABSTRACT-SYNTAX ::= {
       AssistHandoffSSF-SCF-PDUs
       IDENTIFIED BY                      id-as-assistHandoff-ssf-scfAS}
  AssistHandoffSSF-SCF-PDUs ::= TCMessage {{AssistHandoffSsfToScfInvokable},
                                          {AssistHandoffSsfToScfReturnable}}

  AssistHandoffSsfToScfInvokable OPERATION ::= {
                                          activityTest |
                                          applyCharging  {networkSpecificBoundSet}|
                                          applyChargingReport  {networkSpecificBoundSet}|
                                          assistRequestInstructions  {networkSpecificBoundSet}|
                                          cancel  {networkSpecificBoundSet}|
                                          cancelStatusReportRequest  {networkSpecificBoundSet}|
                                          connectToResource  {networkSpecificBoundSet}|
                                          disconnectForwardConnection |
                                          disconnectForwardConnectionWithArgument
                                          {networkSpecificBoundSet}|
                                          furnishChargingInformation  {networkSpecificBoundSet}|
                                          holdCallInNetwork |
                                          playAnnouncement  {networkSpecificBoundSet}|
                                          promptAndCollectUserInformation  {networkSpecificBoundSet}|
                                          requestCurrentStatusReport  {networkSpecificBoundSet}|
                                          requestEveryStatusChangeReport {networkSpecificBoundSet}|
                                          requestFirstStatusMatchReport  {networkSpecificBoundSet}|
                                          resetTimer  {networkSpecificBoundSet}|
                                          statusReport  {networkSpecificBoundSet}|
                                          scriptClose  {networkSpecificBoundSet}|
                                          scriptEvent  {networkSpecificBoundSet}|
                                          scriptInformation  {networkSpecificBoundSet}|
                                          scriptRun  {networkSpecificBoundSet}|
                                          specializedResourceReport |
                                          promptAndReceiveMessage {networkSpecificBoundSet}
                                          }
```

```
AssistHandoffSsfToScfReturnable OPERATION ::= {
                                             activityTest |
                                             applyCharging {networkSpecificBoundSet}|
                                             applyChargingReport {networkSpecificBoundSet}|
                                             assistRequestInstructions {networkSpecificBoundSet}|
                                             cancel {networkSpecificBoundSet}|
                                             cancelStatusReportRequest {networkSpecificBoundSet}|
                                             connectToResource {networkSpecificBoundSet}|
                                             disconnectForwardConnection |
                                             disconnectForwardConnectionWithArgument
                                             {networkSpecificBoundSet}|
                                             furnishChargingInformation {networkSpecificBoundSet}|
                                             holdCallInNetwork |
                                             playAnnouncement {networkSpecificBoundSet}|
                                             promptAndCollectUserInformation {networkSpecificBoundSet}|
                                             requestCurrentStatusReport {networkSpecificBoundSet}|
                                             requestEveryStatusChangeReport {networkSpecificBoundSet}|
                                             requestFirstStatusMatchReport {networkSpecificBoundSet}|
                                             resetTimer {networkSpecificBoundSet}|
                                             scriptClose {networkSpecificBoundSet}|
                                             scriptInformation {networkSpecificBoundSet}|
                                             scriptRun {networkSpecificBoundSet}|
                                             promptAndReceiveMessage {networkSpecificBoundSet}
                                             }

scf-ssfGenericAbstractSyntax  ABSTRACT-SYNTAX ::= {
      GenericSCF-SSF-PDUs
      IDENTIFIED BY                          id-as-scf-ssfGenericAS}

GenericSCF-SSF-PDUs ::= TCMessage {{ScfToSsfGenericInvokable}, {ScfToSsfGenericReturnable}}

ScfSsfGenericInvokable  OPERATION ::= {
                                             activateServiceFiltering {networkSpecificBoundSet}|
                                             activityTest |
                                             applyCharging {networkSpecificBoundSet}|
                                             applyChargingReport {networkSpecificBoundSet}|
                                             callInformationRequest {networkSpecificBoundSet}|
                                             cancel {networkSpecificBoundSet}|
                                             cancelStatusReportRequest {networkSpecificBoundSet}|
                                             collectInformation {networkSpecificBoundSet}|
                                             connect {networkSpecificBoundSet}|
                                             connectToResource {networkSpecificBoundSet}|
                                             continue |
                                             continueWithArgument{networkSpecificBoundSet}|
                                             disconnectForwardConnection |
                                             disconnectForwardConnectionWithArgument
                                             {networkSpecificBoundSet}|
                                             disconnectLeg {networkSpecificBoundSet}|
                                             establishTemporaryConnection {networkSpecificBoundSet}|
                                             furnishChargingInformation  {networkSpecificBoundSet}|
                                             holdCallInNetwork |
                                             initiateCallAttempt  {networkSpecificBoundSet}|
                                             mergeCallSegments {networkSpecificBoundSet}|
                                             moveCallSegments {networkSpecificBoundSet}|
                                             moveLeg {networkSpecificBoundSet}|
                                             createCallSegmentAssociation {networkSpecificBoundSet}|
                                             releaseCall {networkSpecificBoundSet}|
                                             reconnect {networkSpecificBoundSet}|
                                             requestCurrentStatusReport  {networkSpecificBoundSet}|
                                             requestEveryStatusChangeReport {networkSpecificBoundSet}|
                                             requestFirstStatusMatchReport  {networkSpecificBoundSet}|
                                             requestNotificationChargingEvent  {networkSpecificBoundSet}|
                                             requestReportBCSMEvent  {networkSpecificBoundSet}|
                                             requestReportFacilityEvent {networkSpecificBoundSet}|
                                             requestReportUTSI {networkSpecificBoundSet}|
                                             resetTimer  {networkSpecificBoundSet}|
                                             sendChargingInformation  {networkSpecificBoundSet}|
                                             sendFacilityInformation {networkSpecificBoundSet}|
                                             sendSTUI {networkSpecificBoundSet}|
                                             splitLeg  {networkSpecificBoundSet}|
                                             playAnnouncement  {networkSpecificBoundSet}|
                                             promptAndCollectUserInformation  {networkSpecificBoundSet}|
                                             scriptClose  {networkSpecificBoundSet}|
                                             scriptInformation  {networkSpecificBoundSet}|
                                             scriptRun  {networkSpecificBoundSet}|
                                             promptAndReceiveMessage  {networkSpecificBoundSet}
                                             }

ScfSsfGenericReturnable  OPERATION ::= {
                                             activateServiceFiltering  {networkSpecificBoundSet}|
                                             activityTest |
                                             applyCharging  {networkSpecificBoundSet}|
                                             applyChargingReport  {networkSpecificBoundSet}|
                                             callInformationReport  {networkSpecificBoundSet}|
                                             callInformationRequest  {networkSpecificBoundSet}|
                                             cancel  {networkSpecificBoundSet}|
                                             cancelStatusReportRequest  {networkSpecificBoundSet}|
                                             collectInformation  {networkSpecificBoundSet}|
                                             connect  {networkSpecificBoundSet}|
                                             connectToResource  {networkSpecificBoundSet}|
                                             disconnectForwardConnection |
                                             disconnectForwardConnectionWithArgument
                                             {networkSpecificBoundSet}|
                                             disconnectLeg {networkSpecificBoundSet}|
                                             entityReleased {networkSpecificBoundSet}|
                                             establishTemporaryConnection {networkSpecificBoundSet}|
                                             eventNotificationCharging {networkSpecificBoundSet} | resetTimer
                                             {networkSpecificBoundSet}|
                                             eventReportBCSM  {networkSpecificBoundSet}|
                                             eventReportFacility {networkSpecificBoundSet}|
                                             furnishChargingInformation  {networkSpecificBoundSet}|
                                             holdCallInNetwork |
                                             initiateCallAttempt  {networkSpecificBoundSet}|
                                             mergeCallSegments {networkSpecificBoundSet}|
                                             moveCallSegments {networkSpecificBoundSet}|
                                             moveLeg {networkSpecificBoundSet}|
                                             createCallSegmentAssociation {networkSpecificBoundSet}|
                                             reconnect {networkSpecificBoundSet}|
                                             reportUTSI {networkSpecificBoundSet}|
                                             requestCurrentStatusReport  {networkSpecificBoundSet}|
                                             requestEveryStatusChangeReport {networkSpecificBoundSet}|
                                             requestFirstStatusMatchReport  {networkSpecificBoundSet}|
                                             requestNotificationChargingEvent  {networkSpecificBoundSet}|
                                             requestReportBCSMEvent  {networkSpecificBoundSet}|
                                             requestReportFacilityEvent {networkSpecificBoundSet}|
                                             requestReportUTSI {networkSpecificBoundSet} |
                                             sendChargingInformation  {networkSpecificBoundSet}|
                                             sendFacilityInformation {networkSpecificBoundSet}|
                                             sendSTUI {networkSpecificBoundSet}|
                                             serviceFilteringResponse  {networkSpecificBoundSet}|
                                             splitLeg {networkSpecificBoundSet}|
                                             statusReport  {networkSpecificBoundSet}|
                                             playAnnouncement  {networkSpecificBoundSet}|
                                             promptAndCollectUserInformation  {networkSpecificBoundSet}|
                                             scriptClose  {networkSpecificBoundSet}|
                                             scriptEvent  {networkSpecificBoundSet}|
                                             scriptInformation  {networkSpecificBoundSet}|
                                             scriptRun  {networkSpecificBoundSet}|
                                             specializedResourceReport |
                                             promptAndReceiveMessage  {networkSpecificBoundSet}
                                             }


     scf-ssfDpSpecificAbstractSyntax  ABSTRACT-SYNTAX ::= {
             DpSpecificSCF-SSF-PDUs
             IDENTIFIED BY                   id-as-scf-ssfDpSpecificAS}

     DpSpecificSCF-SCF-PDUs ::= TCMessage {{ScfToSsfDpSpecificInvokable},
                                             {ScfToSsfDpSpecificReturnable}}

     ScfSsfDpSpecificInvokable  OPERATION ::= {
                                             activateServiceFiltering  {networkSpecificBoundSet}|
                                             activityTest |
                                             analyseInformation {networkSpecificBoundSet} |
                                             analysedInformation {networkSpecificBoundSet}|
                                             applyCharging {networkSpecificBoundSet} |
                                             applyChargingReport {networkSpecificBoundSet} |
                                             callInformationRequest {networkSpecificBoundSet} |
                                             cancel {networkSpecificBoundSet} |
                                             cancelStatusReportRequest {networkSpecificBoundSet} |
                                             collectedInformation {networkSpecificBoundSet} |
                                             collectInformation {networkSpecificBoundSet} |
                                             connect {networkSpecificBoundSet}|
                                             connectToResource {networkSpecificBoundSet} |
                                             continue |
                                             continueWithArgument {networkSpecificBoundSet}|
                                             disconnectForwardConnection |
                                             disconnectForwardConnectionWithArgument
                                             {networkSpecificBoundSet}|
                                             disconnectLeg {networkSpecificBoundSet}|
                                             establishTemporaryConnection {networkSpecificBoundSet}|
                                             furnishChargingInformation  {networkSpecificBoundSet}|
                                             holdCallInNetwork |
                                             initiateCallAttempt {networkSpecificBoundSet}|
                                             mergeCallSegments  {networkSpecificBoundSet}|
                                             moveCallSegments {networkSpecificBoundSet}|
                                             moveLeg {networkSpecificBoundSet}|
                                             oAbandon {networkSpecificBoundSet}|
                                             oAnswer {networkSpecificBoundSet}|
                                             oCalledPartyBusy {networkSpecificBoundSet}|
                                             oDisconnect {networkSpecificBoundSet}|
                                             oMidCall {networkSpecificBoundSet}|
                                             oNoAnswer {networkSpecificBoundSet}|
                                             createCallSegmentAssociation {networkSpecificBoundSet}|
                                             originationAttemptAuthorized  {networkSpecificBoundSet}|
                                             reconnect {networkSpecificBoundSet}|
                                             releaseCall {networkSpecificBoundSet}|
                                             requestCurrentStatusReport  {networkSpecificBoundSet}|
                                             requestEveryStatusChangeReport {networkSpecificBoundSet}|
                                             requestFirstStatusMatchReport  {networkSpecificBoundSet}|
                                             requestNotificationChargingEvent  {networkSpecificBoundSet}|
                                             requestReportBCSMEvent  {networkSpecificBoundSet}|
                                             requestReportFacilityEvent {networkSpecificBoundSet}|
                                             requestReportUTSI {networkSpecificBoundSet}|
                                             resetTimer  {networkSpecificBoundSet}|
                                             routeSelectFailure  {networkSpecificBoundSet}|
                                             selectFacility  {networkSpecificBoundSet}|
                                             selectRoute  {networkSpecificBoundSet}|
                                             sendChargingInformation  {networkSpecificBoundSet}|
                                             sendFacilityInformation {networkSpecificBoundSet}|
                                             sendSTUI {networkSpecificBoundSet}|
                                             splitLeg {networkSpecificBoundSet}|
                                             tAnswer  {networkSpecificBoundSet}|
                                             tBusy  {networkSpecificBoundSet}|
                                             tDisconnect  {networkSpecificBoundSet}|
                                             termAttemptAuthorized  {networkSpecificBoundSet}|
                                             tMidCall  {networkSpecificBoundSet}|
                                             tNoAnswer  {networkSpecificBoundSet}|
                                             playAnnouncement  {networkSpecificBoundSet}|
                                             promptAndCollectUserInformation  {networkSpecificBoundSet}|
                                             scriptClose  {networkSpecificBoundSet}|
                                             scriptInformation  {networkSpecificBoundSet}|
                                             scriptRun  {networkSpecificBoundSet}|
                                             promptAndReceiveMessage  {networkSpecificBoundSet}
                                             }

     ScfSsfDpSpecificReturnable  OPERATION ::= {
                                             activateServiceFiltering  {networkSpecificBoundSet}|
                                             activityTest |
                                             analyseInformation  {networkSpecificBoundSet}|
                                             analysedInformation  {networkSpecificBoundSet}|
                                             applyCharging  {networkSpecificBoundSet}|
                                             applyChargingReport  {networkSpecificBoundSet}|
                                             callInformationReport  {networkSpecificBoundSet}|
                                             callInformationRequest  {networkSpecificBoundSet}|
                                             cancel  {networkSpecificBoundSet}|
                                             cancelStatusReportRequest  {networkSpecificBoundSet}|
                                             collectedInformation  {networkSpecificBoundSet}|
                                             collectInformation  {networkSpecificBoundSet}|
                                             connect  {networkSpecificBoundSet}|
                                             connectToResource  {networkSpecificBoundSet}|
                                             disconnectForwardConnection |
                                             disconnectForwardConnectionWithArgument
                                             {networkSpecificBoundSet}|
                                             disconnectLeg {networkSpecificBoundSet}|
                                             entityReleased {networkSpecificBoundSet}|
                                             establishTemporaryConnection {networkSpecificBoundSet}|
                                             eventNotificationCharging  {networkSpecificBoundSet}|
                                             eventReportFacility {networkSpecificBoundSet}|
                                             furnishChargingInformation  {networkSpecificBoundSet}|
                                             holdCallInNetwork |
                                             initiateCallAttempt  {networkSpecificBoundSet}|
                                             initiateCallAttempt {networkSpecificBoundSet}|
                                             mergeCallSegments {networkSpecificBoundSet}|
                                             moveCallSegments {networkSpecificBoundSet}|
                                             moveLeg {networkSpecificBoundSet}|
                                             oAnswer  {networkSpecificBoundSet}|
                                             oCalledPartyBusy  {networkSpecificBoundSet}|
                                             oDisconnect  {networkSpecificBoundSet}|
                                             oMidCall  {networkSpecificBoundSet}|
                                             oAbandon {networkSpecificBoundSet}|
                                             oNoAnswer  {networkSpecificBoundSet}|
                                             createCallSegmentAssociation {networkSpecificBoundSet}|
                                             originationAttemptAuthorized  {networkSpecificBoundSet}|
                                             reconnect {networkSpecificBoundSet}|
                                             reportUTSI {networkSpecificBoundSet}|
                                             requestCurrentStatusReport  {networkSpecificBoundSet}|
                                             requestEveryStatusChangeReport {networkSpecificBoundSet}|
                                             requestFirstStatusMatchReport  {networkSpecificBoundSet}|
                                             requestNotificationChargingEvent  {networkSpecificBoundSet}|
                                             requestReportBCSMEvent  {networkSpecificBoundSet}|
                                             requestReportFacilityEvent {networkSpecificBoundSet}|
                                             requestReportUTSI {networkSpecificBoundSet}|
                                             resetTimer  {networkSpecificBoundSet}|
                                             routeSelectFailure  {networkSpecificBoundSet}|
                                             selectFacility  {networkSpecificBoundSet}|
                                             selectRoute  {networkSpecificBoundSet}|
                                             sendChargingInformation  {networkSpecificBoundSet}|
                                             sendFacilityInformation {networkSpecificBoundSet}|
                                             sendSTUI {networkSpecificBoundSet}|
                                             serviceFilteringResponse  {networkSpecificBoundSet}|
                                             splitLeg {networkSpecificBoundSet}|
                                             statusReport  {networkSpecificBoundSet}|
                                             tAnswer  {networkSpecificBoundSet}|
                                             tBusy  {networkSpecificBoundSet}|
                                             tDisconnect  {networkSpecificBoundSet}|
                                             termAttemptAuthorized  {networkSpecificBoundSet}|
                                             tMidCall  {networkSpecificBoundSet}|
                                             tNoAnswer  {networkSpecificBoundSet}|
                                             playAnnouncement  {networkSpecificBoundSet}|
                                             promptAndCollectUserInformation  {networkSpecificBoundSet}|
                                             scriptClose  {networkSpecificBoundSet}|
                                             scriptEvent  {networkSpecificBoundSet}|
                                             scriptInformation  {networkSpecificBoundSet}|
                                             scriptRun  {networkSpecificBoundSet}|
                                             specializedResourceReport |
                                             promptAndReceiveMessage  {networkSpecificBoundSet}
                                             }


     scf-ssfTrafficManagementAbstractSyntax  ABSTRACT-SYNTAX ::= {
             TrafficManagementSCF-SSF-PDUs
             IDENTIFIED BY                   id-as-scf-ssfTrafficManagementAS}

     TrafficManagementSCF-SSF-PDUs ::= TCMessage {{ScfToSsfTrafficManagementInvokable}}

     ScfToSsfTrafficManagementInvokable OPERATION ::= {
             callGap {networkSpecificBoundSet}
             }

     scf-ssfServiceManagementAbstractSyntax  ABSTRACT-SYNTAX ::= {
             ServiceManagementSCF-SSF-PDUs
             IDENTIFIED BY                   id-as-scf-ssfServiceManagementAS}

     ServiceManagementSCF-SSF-PDUs ::= TCMessage {{ScfToSsfServiceManagementInvokable},
                                             {ScfToSsfServiceManagementReturnable}}

     ScfToSsfServiceManagementInvokable OPERATION ::= {
             activateServiceFiltering {networkSpecificBoundSet}
             }

     ScfToSsfServiceManagementReturnable OPERATION ::= {
             activateServiceFiltering {networkSpecificBoundSet}
             }
```

```
ssf-scfServiceManagementAbstractSyntax  ABSTRACT-SYNTAX ::= {
        ServiceManagementSSF-SCF-PDUs
        IDENTIFIED BY                   id-as-ssf-scfServiceManagementAS}

ServiceManagementSSF-SCF-PDUs ::= TCMessage {{SsfToScfServiceManagementInvokable}}

SsfToScfServiceManagementInvokable OPERATION ::= {
        serviceFilteringResponse {networkSpecificBoundSet}
        }

scf-ssfStatusReportingAbstractSyntax  ABSTRACT-SYNTAX ::= {
        StatusReportingSCF-SSF-PDUs
        IDENTIFIED BY                   id-as-scf-ssfStatusReportingAS}

StatusReportingSCF-SSF-PDUs ::= TCMessage {{ScfToSsfStatusReportingInvokable},
                                        {ScfToSsfStatusReportingReturnable}}

ScfToSsfStatusReportingInvokable OPERATION  ::= {
                                        cancel  {networkSpecificBoundSet}|
                                        cancelStatusReportRequest  {networkSpecificBoundSet}|
                                        requestCurrentStatusReport  {networkSpecificBoundSet}|
                                        requestEveryStatusChangeReport {networkSpecificBoundSet}|
                                        requestFirstStatusMatchReport {networkSpecificBoundSet}
                                        }

ScfToSsfStatusReportingReturnable OPERATION  ::= {
                                        cancel  {networkSpecificBoundSet}|
                                        cancelStatusReportRequest  {networkSpecificBoundSet}|
                                        requestCurrentStatusReport  {networkSpecificBoundSet}|
                                        requestEveryStatusChangeReport {networkSpecificBoundSet}|
                                        requestFirstStatusMatchReport  {networkSpecificBoundSet}|
                                        statusReport {networkSpecificBoundSet}
                                        }

scf-ssfTriggerManagementAbstractSyntax  ABSTRACT-SYNTAX ::= {
        TriggerManagementSCF-SSF-PDUs
        IDENTIFIED BY                   id-as-scf-ssfTriggerManagementAS}

TriggerManagementSCF-SSF-PDUs ::= TCMessage {{ScfToSsfTriggerManagementInvokable},
                                        {ScfToSsfTriggerManagementReturnable}}

ScfToSsfTriggerManagementInvokable OPERATION ::= {
        manageTriggerData
        }

ScfToSsfTriggerManagementReturnable OPERATION ::= {
        manageTriggerData
        }
```

SSF-SCF Application Contexts

The **SSF to SCF** contracts are realized by four application contexts, **cs2ssf-scfGenericAC**, **cs2ssf-scfDPSpecificAC**, **cs2ssf-scfAssistHandoffAC** and **cs2ssf-scfServiceManagementAC**. These application contexts are specified as information objects of the class APPLICATION-CONTEXT.

```
cs2ssf-scfGenericAC  APPLICATION-CONTEXT ::= {
        CONTRACT                        inCs2SsfToScfGeneric
        DIALOGUE MODE                   structured
        ABSTRACT SYNTAXES               {dialogue-abstract-syntax |
                                        ssf-scfGenericAbstractSyntax}
        APPLICATION CONTEXT NAME        id-ac-cs2-ssf-scfGenericAC}

cs2ssf-scfDPSpecificAC  APPLICATION-CONTEXT ::= {
        CONTRACT                        inCs2SsfToScfDpSpecific
        DIALOGUE MODE                   structured
        ABSTRACT SYNTAXES               {dialogue-abstract-syntax |
                                        ssf-scfDpSpecificAbstractSyntax}
        APPLICATION CONTEXT NAME        id-ac-cs2-ssf-scfDPSpecificAC}

cs2ssf-scfAssistHandoffAC  APPLICATION-CONTEXT ::=      {
        CONTRACT                        inCs2AssistHandoffSsfToScf
        DIALOGUE MODE                   structured
        ABSTRACT SYNTAXES               {dialogue-abstract-syntax |
                                        assistHandoff-ssf-scfAbstractSyntax}
        APPLICATION CONTEXT NAME        id-ac-cs2-ssf-scfAssistHandoffAC}

cs2ssf-scfServiceManagementAC  APPLICATION-CONTEXT ::= {
        CONTRACT                        inCs2SsfToScfServiceManagement
        DIALOGUE MODE                   structured
        ABSTRACT SYNTAXES               {dialogue-abstract-syntax |
                                        ssf-scfServiceManagementAbstractSyntax}
        APPLICATION CONTEXT NAME        id-ac-cs2-ssf-scfServiceManagementAC}
```

The **SCF to SSF** contracts are realized by six application contexts, **cs2scf-ssfGenericAC**, **cs2scf-ssfDPSpecificAC**, **cs2scf-ssfTrafficManagementAC**, **cs2scf-ssfServiceManagementAC, cs2scf-ssfStatusReportingAC, and cs2scf-ssfTriggerManagementAC**. These application contexts are specified as information objects of the class APPLICATION-CONTEXT.

```
cs2scf-ssfGenericAC  APPLICATION-CONTEXT ::= {
        CONTRACT                        inCs2ScfToSsfGeneric
        DIALOGUE MODE                   structured
        ABSTRACT SYNTAXES               {dialogue-abstract-syntax |
                                        ssf-scfGenericAbstractSyntax}
        APPLICATION CONTEXT NAME        id-ac-cs2-scf-ssfGenericAC}

cs2scf-ssfDPSpecificAC  APPLICATION-CONTEXT ::= {
        CONTRACT                        inCs2ScfToSsfDpSpecific
        DIALOGUE MODE                   structured
        ABSTRACT SYNTAXES               {dialogue-abstract-syntax |
                                        scf-ssfDpSpecificAbstractSyntax}
        APPLICATION CONTEXT NAME        id-ac-cs2-scf-ssfDPSpecificAC}

cs2scf-ssfTrafficManagementAC  APPLICATION-CONTEXT ::= {
        CONTRACT                        inCs2ScfToSsfTrafficManagement
        DIALOGUE MODE                   structured
        ABSTRACT SYNTAXES               {dialogue-abstract-syntax |
                                        scf-ssfTrafficManagementAbstractSyntax}
        APPLICATION CONTEXT NAME        id-ac-cs2-scf-ssfTrafficManagementAC}

cs2scf-ssfServiceManagementAC  APPLICATION-CONTEXT ::= {
        CONTRACT                        inCs2ScfToSsfServiceManagement
        DIALOGUE MODE                   structured
        ABSTRACT SYNTAXES               {dialogue-abstract-syntax |
                                        scf-ssfServiceManagementAbstractSyntax}
        APPLICATION CONTEXT NAME        id-ac-cs2-scf-ssfServiceManagementAC}
```

```
cs2scf-ssfStatusReportingAC  APPLICATION-CONTEXT ::= {
        CONTRACT                          inCs2ScfToSsfStatusReporting
        DIALOGUE MODE                     structured
        ABSTRACT SYNTAXES                 {dialogue-abstract-syntax |
                                          scf-ssfStatusReportingAbstractSyntax}
        APPLICATION CONTEXT NAME          id-ac-cs2-scf-ssfStatusReportingAC}

cs2scf-ssfTriggerManagementAC  APPLICATION-CONTEXT ::= {
        CONTRACT                          inCs2ScfToSsfTriggerManagement
        DIALOGUE MODE                     structured
        ABSTRACT SYNTAXES                 {dialogue-abstract-syntax |
                                          scf-ssfTriggerManagementAbstractSyntax}
        APPLICATION CONTEXT NAME          id-ac-cs2-scf-ssfTriggerManagementAC}
```

5.2.2 SSF/SCF ASN.1 module

```
IN-CS2-SSF-SCF-pkgs-contracts-acs {itu-t recommendation q 1228 modules(0) in-cs2-ssf-scf-pkgs-contracts-acs
(6) version1(0)}

DEFINITIONS ::=

BEGIN

-- This module describes the operation-packages, contracts and application-contexts used
-- over the SSF-SCF interface.

IMPORTS


        PARAMETERS-BOUND,
        networkSpecificBoundSet
FROM IN-CS2-classes classes


        ROS-OBJECT-CLASS, CONTRACT, OPERATION-PACKAGE, OPERATION

FROM Remote-Operations-Information-Objects ros-InformationObjects

        TCMessage {}

FROM TCAPMessages tc-Messages

        APPLICATION-CONTEXT, dialogue-abstract-syntax

FROM TC-Notation-Extensions tc-NotationExtensions

        activateServiceFiltering {},
        activityTest,
        analysedInformation {},
        analyseInformation {},
        applyCharging {},
        applyChargingReport {},
        assistRequestInstructions {},
        authorizeTermination {},
        callGap {},
        callInformationReport {},
        callInformationRequest {},
        cancel {},
        cancelStatusReportRequest {},
        collectedInformation {},
        collectInformation {},
        connect {},
        connectToResource {},
        continue,
        continueWithArgument {},
        createCallSegmentAssociation {},
        disconnectForwardConnection,
        disconnectForwardConnectionWithArgument {},
        disconnectLeg {},
        entityReleased {},
        establishTemporaryConnection {},
        eventNotificationCharging {},
        eventReportBCSM {},
        eventReportFacility {},
        facilitySelectedAndAvailable {},
        furnishChargingInformation {},
        holdCallInNetwork,
        initialDP {},
        initiateCallAttempt {},
        manageTriggerData {},
        mergeCallSegments {},
        moveCallSegments {},
        moveLeg {},
        oAbandon {},
        oAnswer {},
        oCalledPartyBusy {},
        oDisconnect {},
        oMidCall {},
        oNoAnswer {},
        originationAttempt {},
        originationAttemptAuthorized {},
        oSuspended {},
        reconnect {},
        releaseCall {},
        reportUTSI {},
        requestCurrentStatusReport {},
        requestEveryStatusChangeReport {},
        requestFirstStatusMatchReport {},
        requestNotificationChargingEvent {},
        requestReportBCSMEvent {},
        requestReportUTSI {},
        requestReportFacilityEvent {},
        resetTimer {},
        routeSelectFailure {},
        selectFacility {},
        selectRoute {},
        sendChargingInformation {},
        sendFacilityInformation {},
        sendSTUI {},
        serviceFilteringResponse {},
        splitLeg {},
        statusReport {},
        tAnswer {},
        tBusy {},
        tDisconnect {},
        terminationAttempt {},
        termAttemptAuthorized {},
        tMidCall {},
        tNoAnswer {},
        tSuspended {}
```

```
FROM  IN-CS2-SSF-SCF-ops-args ssf-scf-Operations

        playAnnouncement {},
        promptAndCollectUserInformation {},
        promptAndReceiveMessage {},
        scriptClose {},
        scriptEvent {},
        scriptInformation {},
        scriptRun {},
        specializedResourceReport

FROM IN-CS2-SCF-SRF-ops-args scf-srf-Operations

        specializedResourceControlPackage {},
        scriptControlPackage {},
        messageControlPackage {}

FROM IN-CS2-SCF-SRF-pkgs-contracts-acs scf-srf-Protocol

        id-ac-cs2-ssf-scfGenericAC,
        id-ac-cs2-ssf-scfDPSpecificAC,
        id-ac-cs2-ssf-scfAssistHandoffAC,
        id-ac-cs2-ssf-scfServiceManagementAC,
        id-ac-cs2-scf-ssfGenericAC,
        id-ac-cs2-scf-ssfDPSpecificAC,
        id-ac-cs2-scf-ssfTrafficManagementAC,
        id-ac-cs2-scf-ssfServiceManagementAC,
        id-ac-cs2-scf-ssfStatusReportingAC,
        id-ac-cs2-scf-ssfTriggerManagementAC,
        id-inCs2SsfToScfGeneric,
        id-inCs2SsfToScfDpSpecific,
        id-inCs2AssistHandoffSsfToScf,
        id-inCs2ScfToSsfGeneric,
        id-inCs2ScfToSsfDpSpecific,
        id-inCs2ScfToSsfTrafficManagement,
        id-inCs2ScfToSsfServiceManagement,
        id-inCs2SsfToScfServiceManagement,
        id-inCs2ScfToSsfStatusReporting,
        id-inCs2ScfToSsfTriggerManagement,
        id-as-ssf-scfGenericAS,
        id-as-ssf-scfDpSpecificAS,
        id-as-assistHandoff-ssf-scfAS,
        id-as-scf-ssfGenericAS,
        id-as-scf-ssfDpSpecificAS,
        id-as-scf-ssfTrafficManagementAS,
        id-as-scf-ssfServiceManagementAS,
        id-as-ssf-scfServiceManagementAS,
        id-as-scf-ssfStatusReportingAS,
        id-as-scf-ssfTriggerManagementAS,
        id-package-scfActivation,
        id-package-basicBCPDP,
        id-package-advancedBCPDP,
        id-package-srf-scfActivationOfAssist,
        id-package-assistConnectionEstablishment,
        id-package-genericDisconnectResource,
        id-package-nonAssistedConnectionEstablishment,
        id-package-connect,
        id-package-callHandling,
        id-package-bcsmEventHandling,
        id-package-chargingEventHandling,
        id-package-ssfCallProcessing,
        id-package-scfCallInitiation,
        id-package-timer,
        id-package-billing,
        id-package-charging,
        id-package-trafficManagement,
        id-package-serviceManagementActivate,
        id-package-serviceManagementResponse,
        id-package-callReport,
        id-package-signallingControl,
        id-package-activityTest,
        id-package-statusReporting,
        id-package-cancel,
        id-package-cphResponse,
        id-package-entityReleased,
        id-package-triggerManagement,
        id-package-uSIHandling,
        id-package-facilityIEHandling,
        id-package-dpSpecificEventHandling,
        classes, ros-InformationObjects, tc-Messages, tc-NotationExtensions,
        ssf-scf-Operations, scf-srf-Operations, scf-srf-Protocol

FROM IN-CS2-object-identifiers {itu-t recommendation q 1228 modules(0) in-cs2-object-identifiers (17)
version1(0)}

;
-- Application Contexts

cs2ssf-scfGenericAC  APPLICATION-CONTEXT ::= {
        CONTRACT                        inCs2SsfToScfGeneric
        DIALOGUE MODE                   structured
        ABSTRACT SYNTAXES               {dialogue-abstract-syntax |
                                        ssf-scfGenericAbstractSyntax}
        APPLICATION CONTEXT NAME        id-ac-cs2-ssf-scfGenericAC}

cs2ssf-scfDPSpecificAC  APPLICATION-CONTEXT ::= {
        CONTRACT                        inCs2SsfToScfDpSpecific
        DIALOGUE MODE                   structured
        ABSTRACT SYNTAXES               {dialogue-abstract-syntax |
                                        ssf-scfDpSpecificAbstractSyntax}
        APPLICATION CONTEXT NAME        id-ac-cs2-ssf-scfDPSpecificAC}

cs2ssf-scfAssistHandoffAC  APPLICATION-CONTEXT ::=      {
        CONTRACT                        inCs2AssistHandoffSsfToScf
        DIALOGUE MODE                   structured
        ABSTRACT SYNTAXES               {dialogue-abstract-syntax |
                                        assistHandoff-ssf-scfAbstractSyntax}
        APPLICATION CONTEXT NAME        id-ac-cs2-ssf-scfAssistHandoffAC}

cs2ssf-scfServiceManagementAC  APPLICATION-CONTEXT ::= {
        CONTRACT                        inCs2SsfToScfServiceManagement
        DIALOGUE MODE                   structured
        ABSTRACT SYNTAXES               {dialogue-abstract-syntax |
                                        ssf-scfServiceManagementAbstractSyntax}
        APPLICATION CONTEXT NAME        id-ac-cs2-ssf-scfServiceManagementAC}

cs2scf-ssfGenericAC  APPLICATION-CONTEXT ::= {
        CONTRACT                        inCs2ScfToSsfGeneric
        DIALOGUE MODE                   structured
        ABSTRACT SYNTAXES               {dialogue-abstract-syntax |
                                        ssf-scfGenericAbstractSyntax}
        APPLICATION CONTEXT NAME        id-ac-cs2-scf-ssfGenericAC}
```

```
cs2scf-ssfDPSpecificAC  APPLICATION-CONTEXT ::= {
        CONTRACT                           inCs2ScfToSsfDpSpecific
        DIALOGUE MODE                      structured
        ABSTRACT SYNTAXES                  {dialogue-abstract-syntax |
                                           scf-ssfDpSpecificAbstractSyntax}
        APPLICATION CONTEXT NAME           id-ac-cs2-scf-ssfDPSpecificAC}

cs2scf-ssfTrafficManagementAC  APPLICATION-CONTEXT ::= {
        CONTRACT                           inCs2ScfToSsfTrafficManagement
        DIALOGUE MODE                      structured
        ABSTRACT SYNTAXES                  {dialogue-abstract-syntax |
                                           scf-ssfTrafficManagementAbstractSyntax}
        APPLICATION CONTEXT NAME           id-ac-cs2-scf-ssfTrafficManagementAC}

cs2scf-ssfServiceManagementAC  APPLICATION-CONTEXT ::= {
        CONTRACT                           inCs2ScfToSsfServiceManagement
        DIALOGUE MODE                      structured
        ABSTRACT SYNTAXES                  {dialogue-abstract-syntax |
                                           scf-ssfServiceManagementAbstractSyntax}
        APPLICATION CONTEXT NAME           id-ac-cs2-scf-ssfServiceManagementAC}

cs2scf-ssfStatusReportingAC  APPLICATION-CONTEXT ::= {
        CONTRACT                           inCs2ScfToSsfStatusReporting
        DIALOGUE MODE                      structured
        ABSTRACT SYNTAXES                  {dialogue-abstract-syntax |
                                           scf-ssfStatusReportingAbstractSyntax}
        APPLICATION CONTEXT NAME           id-ac-cs2-scf-ssfStatusReportingAC}

cs2scf-ssfTriggerManagementAC  APPLICATION-CONTEXT ::= {
        CONTRACT                           inCs2ScfToSsfTriggerManagement
        DIALOGUE MODE                      structured
        ABSTRACT SYNTAXES                  {dialogue-abstract-syntax |
                                           scf-ssfTriggerManagementAbstractSyntax}
        APPLICATION CONTEXT NAME           id-ac-cs2-scf-ssfTriggerManagementAC}


-- Contracts

inCs2SsfToScfGeneric CONTRACT ::= {
-- dialogue initiated by SSF with InitialDP Operation
        INITIATOR CONSUMER OF    {exceptionInformPackage  {networkSpecificBoundSet} |
                                 scfActivationPackage {networkSpecificBoundSet} }
        RESPONDER CONSUMER OF    {activityTestPackage|
                                 assistConnectionEstablishmentPackage {networkSpecificBoundSet} |
                                 bcsmEventHandlingPackage {networkSpecificBoundSet} |
                                 billingPackage {networkSpecificBoundSet} |
                                 callHandlingPackage {networkSpecificBoundSet} |
                                 callReportPackage {networkSpecificBoundSet} |
                                 cancelPackage {networkSpecificBoundSet} |
                                 chargingEventHandlingPackage {networkSpecificBoundSet} |
                                 chargingPackage {networkSpecificBoundSet} |
                                 connectPackage {networkSpecificBoundSet} |
                                 cphResponsePackage  {networkSpecificBoundSet} |
                                 facilityIEHandlingPackage {networkSpecificBoundSet} |
                                 genericDisconnectResourcePackage {networkSpecificBoundSet} |
                                 nonAssistedConnectionEstablishmentPackage
                                 {networkSpecificBoundSet} |
                                 signallingControlPackage {networkSpecificBoundSet} |
                                 specializedResourceControlPackage {networkSpecificBoundSet} |
                                 scriptControlPackage {networkSpecificBoundSet} |
                                        messageControlPackage {networkSpecificBoundSet} |
                                        ssfCallProcessingPackage {networkSpecificBoundSet} |
                                        statusReportingPackage {networkSpecificBoundSet} |
                                        timerPackage {networkSpecificBoundSet} |
                                        trafficManagementPackage {networkSpecificBoundSet} |
                                        uSIHandlingPackage {networkSpecificBoundSet} |
                                        scfCallInitiationPackage {networkSpecificBoundSet}
                                        }
     ID                                 id-inCs2SsfToScfGeneric
     }

inCs2SsfToScfDpSpecific CONTRACT ::= {
-- dialogue initiated by SSF with DP Specific Iniltial Operations
     INITIATOR CONSUMER OF              {advancedBCPDPPackage {networkSpecificBoundSet} |
                                        basicBCPDPPackage  {networkSpecificBoundSet} |
                                        exceptionInformPackage {networkSpecificBoundSet} }
     RESPONDER CONSUMER OF              {activityTestPackage|
                                        assistConnectionEstablishmentPackage {networkSpecificBoundSet} |
                                        billingPackage {networkSpecificBoundSet} |
                                        callHandlingPackage {networkSpecificBoundSet} |
                                        callReportPackage {networkSpecificBoundSet} |
                                        cancelPackage {networkSpecificBoundSet} |
                                        chargingEventHandlingPackage {networkSpecificBoundSet} |
                                        chargingPackage {networkSpecificBoundSet} |
                                        connectPackage {networkSpecificBoundSet} |
                                        cphResponsePackage  {networkSpecificBoundSet} |
                                        dpSpecificEventHandlingPackage {networkSpecificBoundSet} |
                                        facilityIEHandlingPackage {networkSpecificBoundSet} |
                                        genericDisconnectResourcePackage {networkSpecificBoundSet} |
                                        nonAssistedConnectionEstablishmentPackage
                                        {networkSpecificBoundSet} |
                                        signallingControlPackage {networkSpecificBoundSet} |
                                        specializedResourceControlPackage {networkSpecificBoundSet} |
                                        scriptControlPackage {networkSpecificBoundSet} |
                                        messageControlPackage {networkSpecificBoundSet} |
                                        ssfCallProcessingPackage {networkSpecificBoundSet} |
                                        statusReportingPackage {networkSpecificBoundSet} |
                                        timerPackage {networkSpecificBoundSet} |
                                        trafficManagementPackage {networkSpecificBoundSet} |
                                        uSIHandlingPackage {networkSpecificBoundSet} |
                                        scfCallInitiationPackage {networkSpecificBoundSet}
                                        }
     ID                                 id-inCs2SsfToScfDpSpecific
     }

inCs2AssistHandoffSsfToScf CONTRACT ::= {
-- dialogue initiated by SSF with AssistRequestInstructions
     INITIATOR CONSUMER OF              {srf-scfActivationOfAssistPackage {networkSpecificBoundSet} }
     RESPONDER CONSUMER OF              {activityTestPackage|
                                        billingPackage {networkSpecificBoundSet} |
                                        callHandlingPackage {networkSpecificBoundSet} |
                                        cancelPackage {networkSpecificBoundSet} |
                                        chargingPackage {networkSpecificBoundSet} |
                                        genericDisconnectResourcePackage {networkSpecificBoundSet} |
                                        nonAssistedConnectionEstablishmentPackage
                                        {networkSpecificBoundSet} |
                                        specializedResourceControlPackage {networkSpecificBoundSet} |
                                        scriptControlPackage {networkSpecificBoundSet} |
                                        messageControlPackage {networkSpecificBoundSet} |
                                        statusReportingPackage {networkSpecificBoundSet} |
                                        timerPackage {networkSpecificBoundSet}
                                        }
     ID                          id-inCs2AssistHandoffSsfToScf
     }

inCs2ScfToSsfGeneric CONTRACT ::= {
-- dialogue initiated by SCF with InitiateCallAttempt, Generic Case
     INITIATOR CONSUMER OF       {activityTestPackage|
                                        assistConnectionEstablishmentPackage {networkSpecificBoundSet} |
                                        bcsmEventHandlingPackage {networkSpecificBoundSet} |
                                        billingPackage {networkSpecificBoundSet} |
                                        callHandlingPackage {networkSpecificBoundSet} |
                                        callReportPackage {networkSpecificBoundSet} |
                                        cancelPackage {networkSpecificBoundSet} |
                                        chargingPackage {networkSpecificBoundSet} |
                                        connectPackage {networkSpecificBoundSet} |
                                        cphResponsePackage  {networkSpecificBoundSet} |
                                        facilityIEHandlingPackage {networkSpecificBoundSet} |
                                        genericDisconnectResourcePackage {networkSpecificBoundSet} |
                                        nonAssistedConnectionEstablishmentPackage
                                        {networkSpecificBoundSet} |
                                        scfCallInitiationPackage {networkSpecificBoundSet} |
                                        signallingControlPackage {networkSpecificBoundSet} |
                                        specializedResourceControlPackage {networkSpecificBoundSet} |
                                        scriptControlPackage {networkSpecificBoundSet} |
                                        messageControlPackage {networkSpecificBoundSet} |
                                        ssfCallProcessingPackage {networkSpecificBoundSet} |
                                        statusReportingPackage {networkSpecificBoundSet} |
                                        timerPackage {networkSpecificBoundSet} |
                                        uSIHandlingPackage {networkSpecificBoundSet} |
                                        scfCallInitiationPackage {networkSpecificBoundSet}
                                        }
     RESPONDER CONSUMER OF       {exceptionInformPackage {networkSpecificBoundSet} }
     ID                          id-inCs2ScfToSsfGeneric
     }

inCs2ScfToSsfDpSpecific CONTRACT ::= {
-- dialogue initiated by SCF with InitiateCallAttempt, DP Specific Case
     INITIATOR CONSUMER OF       {activityTestPackage|
                                        assistConnectionEstablishmentPackage {networkSpecificBoundSet} |
                                        billingPackage {networkSpecificBoundSet} |
                                        callHandlingPackage {networkSpecificBoundSet} |
                                        callReportPackage {networkSpecificBoundSet} |
                                        cancelPackage {networkSpecificBoundSet} |
                                        chargingEventHandlingPackage {networkSpecificBoundSet} |
                                        chargingPackage {networkSpecificBoundSet} |
                                        connectPackage {networkSpecificBoundSet} |
                                        cphResponsePackage  {networkSpecificBoundSet} |
                                        dpSpecificEventHandlingPackage {networkSpecificBoundSet} |
                                        facilityIEHandlingPackage {networkSpecificBoundSet} |
                                        genericDisconnectResourcePackage {networkSpecificBoundSet} |
                                        nonAssistedConnectionEstablishmentPackage
                                        {networkSpecificBoundSet} |
                                        scfCallInitiationPackage {networkSpecificBoundSet} |
                                        signallingControlPackage {networkSpecificBoundSet} |
                                        specializedResourceControlPackage {networkSpecificBoundSet} |
                                        scriptControlPackage {networkSpecificBoundSet} |
                                        messageControlPackage {networkSpecificBoundSet} |
                                        ssfCallProcessingPackage {networkSpecificBoundSet} |
                                        statusReportingPackage {networkSpecificBoundSet} |
                                        timerPackage {networkSpecificBoundSet} |
                                        uSIHandlingPackage {networkSpecificBoundSet} |
                                        scfCallInitiationPackage {networkSpecificBoundSet}
                                        }
        RESPONDER CONSUMER OF {exceptionInformPackage {networkSpecificBoundSet} }
        ID                              id-inCs2ScfToSsfDpSpecific
        }

inCs2ScfToSsfTrafficManagement CONTRACT ::= {
-- dialogue initiated by SCF with CallGap
        INITIATOR CONSUMER OF   {trafficManagementPackage {networkSpecificBoundSet}
                                        }
        ID                              id-inCs2ScfToSsfTrafficManagement
        }

inCs2ScfToSsfServiceManagement CONTRACT ::= {
-- dialogue initiated by SCF with ActivateServiceFiltering
        INITIATOR CONSUMER OF   {serviceManagementActivatePackage {networkSpecificBoundSet}
                                        }
        ID                              id-inCs2ScfToSsfServiceManagement
        }

inCs2SsfToScfServiceManagement CONTRACT ::= {
-- dialogue initiated/ended by SSF with ServiceFilteringResponse
        INITIATOR CONSUMER OF   {serviceManagementResponsePackage {networkSpecificBoundSet}
                                        }
        ID                              id-inCs2SsfToScfServiceManagement
        }

inCs2ScfToSsfStatusReporting CONTRACT ::= {
-- dialogue initiated by SCF with StatusReporting Operations
        INITIATOR CONSUMER OF   {cancelPackage {networkSpecificBoundSet} |
                                        statusReportingPackage {networkSpecificBoundSet}
                                        }
        ID                              id-inCs2ScfToSsfStatusReporting
        }
inCs2ScfToSsfTriggerManagement CONTRACT ::= {
-- dialogue initiated by SCF with Manage Trigger Data
        INITIATOR CONSUMER OF   {triggerManagementPackage {networkSpecificBoundSet}
                                        }
        ID                              id-inCs2ScfToSsfTriggerManagement
        }

-- Operation Packages

scfActivationPackage  {PARAMETERS-BOUND : bound}   OPERATION-PACKAGE ::= {
        CONSUMER INVOKES                {initialDP {bound}}
        ID                              id-package-scfActivation}

basicBCPDPPackage  {PARAMETERS-BOUND : bound}   OPERATION-PACKAGE ::= {
        CONSUMER INVOKES                {originationAttemptAuthorized {bound}|
                                        collectedInformation {bound}|
                                        analysedInformation {bound}| routeSelectFailure  {bound}|
                                        facilitySelectedAndAvailable  {bound}|
                                        oAbandon  {bound}| originationAttempt  {bound} |
                                        terminationAttempt  {bound} |
                                        oCalledPartyBusy  {bound} | oNoAnswer  {bound} |
                                oAnswer {bound} |
                                oDisconnect {bound} | termAttemptAuthorized {bound} |
                                tBusy {bound} |
                                tNoAnswer {bound} | tAnswer {bound} | tDisconnect {bound} }
    ID                          id-package-basicBCPDP}

advancedBCPDPPackage  {PARAMETERS-BOUND : bound}    OPERATION-PACKAGE ::= {
    CONSUMER INVOKES            {oMidCall {bound} | oSuspended {bound} |
                                tMidCall {bound} | tSuspended{bound} }
    ID                          id-package-advancedBCPDP}

srf-scfActivationOfAssistPackage  {PARAMETERS-BOUND : bound}  OPERATION-PACKAGE ::= {
    CONSUMER INVOKES            {assistRequestInstructions {bound}}
    ID                          id-package-srf-scfActivationOfAssist}

assistConnectionEstablishmentPackage  {PARAMETERS-BOUND : bound}    OPERATION-PACKAGE ::= {
    CONSUMER INVOKES            {establishTemporaryConnection {bound}}
    ID                          id-package-assistConnectionEstablishment}

genericDisconnectResourcePackage  {PARAMETERS-BOUND : bound}    OPERATION-PACKAGE ::= {
    CONSUMER INVOKES            {disconnectForwardConnection |
                                disconnectForwardConnectionWithArgument {bound}}
    ID                          id-package-genericDisconnectResource}

nonAssistedConnectionEstablishmentPackage  {PARAMETERS-BOUND : bound}  OPERATION-PACKAGE ::= {
    CONSUMER INVOKES            {connectToResource {bound}}
    ID                          id-package-nonAssistedConnectionEstablishment}

connectPackage  {PARAMETERS-BOUND : bound}  OPERATION-PACKAGE ::= {
    CONSUMER INVOKES            {connect {bound}}
    ID                          id-package-connect}

callHandlingPackage   {PARAMETERS-BOUND : bound} OPERATION-PACKAGE ::= {
    CONSUMER INVOKES            {holdCallInNetwork | releaseCall {bound}}
    ID                          id-package-callHandling}

bcsmEventHandlingPackage  {PARAMETERS-BOUND : bound}  OPERATION-PACKAGE ::= {
    CONSUMER INVOKES            {requestReportBCSMEvent {bound}}
    SUPPLIER INVOKES            {eventReportBCSM {bound}}
    ID                          id-package-bcsmEventHandling}

dpSpecificEventHandlingPackage  {PARAMETERS-BOUND : bound}    OPERATION-PACKAGE ::= {
    CONSUMER INVOKES            {requestReportBCSMEvent {bound}}
    SUPPLIER INVOKES            {originationAttemptAuthorized {bound} |
                                collectedInformation {bound} |
                                analysedInformation  {bound} | routeSelectFailure  {bound} |
                                facilitySelectedAndAvailable  {bound} |
                                oAbandon  {bound} | originationAttempt  {bound} |
                                terminationAttempt  {bound} |
                                oCalledPartyBusy  {bound} | oNoAnswer  {bound} |
                                oAnswer  {bound} |
                                oDisconnect  {bound} | termAttemptAuthorized  {bound} |
                                tBusy  {bound} |
                                tNoAnswer  {bound} | tAnswer  {bound} | tDisconnect  {bound} |
                                oMidCall  {bound} | oSuspended  {bound} |
                                tMidCall  {bound} | tSuspended {bound}
                                }
    ID                          id-package-dpSpecificEventHandling}
```

```
chargingEventHandlingPackage  {PARAMETERS-BOUND : bound} OPERATION-PACKAGE ::= {
        CONSUMER INVOKES            {requestNotificationChargingEvent {bound}}
        SUPPLIER INVOKES            {eventNotificationCharging {bound}}
        ID                          id-package-chargingEventHandling}

ssfCallProcessingPackage {PARAMETERS-BOUND : bound}  OPERATION-PACKAGE ::= {
        CONSUMER INVOKES            {collectInformation  {bound} | analyseInformation  {bound}|
                                    authorizeTermination  {bound}| selectRoute  {bound}|
                                    selectFacility  {bound}| continue}
        ID                          id-package-ssfCallProcessing}

scfCallInitiationPackage {PARAMETERS-BOUND : bound} OPERATION-PACKAGE ::= {
        CONSUMER INVOKES            {initiateCallAttempt {bound}}
        ID                          id-package-scfCallInitiation}

timerPackage {PARAMETERS-BOUND : bound} OPERATION-PACKAGE ::= {
        CONSUMER INVOKES            {resetTimer {bound}}
        ID                          id-package-timer}

billingPackage {PARAMETERS-BOUND : bound} OPERATION-PACKAGE ::= {
        CONSUMER INVOKES            {furnishChargingInformation {bound}}
        ID                          id-package-billing}

chargingPackage {PARAMETERS-BOUND : bound} OPERATION-PACKAGE ::= {
        CONSUMER INVOKES            {applyCharging {bound}}
        SUPPLIER INVOKES            {applyChargingReport {bound}}
        ID                          id-package-charging}

trafficManagementPackage {PARAMETERS-BOUND : bound} OPERATION-PACKAGE ::= {
        CONSUMER INVOKES            {callGap {bound}}
        ID                          id-package-trafficManagement}

serviceManagementActivatePackage {PARAMETERS-BOUND : bound} OPERATION-PACKAGE ::= {
        CONSUMER INVOKES            {activateServiceFiltering {bound}}
        ID                          id-package-serviceManagementActivate}

serviceManagementResponsePackage {PARAMETERS-BOUND : bound} OPERATION-PACKAGE ::= {
        CONSUMER INVOKES            {serviceFilteringResponse {bound}}
        ID                          id-package-serviceManagementResponse}

callReportPackage {PARAMETERS-BOUND : bound} OPERATION-PACKAGE ::= {
        CONSUMER INVOKES            {callInformationRequest {bound}}
        SUPPLIER INVOKES            {callInformationReport {bound}}
        ID                          id-package-callReport}

signallingControlPackage {PARAMETERS-BOUND : bound} OPERATION-PACKAGE ::= {
        CONSUMER INVOKES            {sendChargingInformation {bound}}
        ID                          id-package-signallingControl}

activityTestPackage  OPERATION-PACKAGE ::= {
        CONSUMER INVOKES            {activityTest}
        ID                          id-package-activityTest}

statusReportingPackage {PARAMETERS-BOUND : bound} OPERATION-PACKAGE ::= {
        CONSUMER INVOKES            {requestCurrentStatusReport  {bound}|
                                    requestEveryStatusChangeReport {bound}|
                                    requestFirstStatusMatchReport {bound}}
        SUPPLIER INVOKES            {statusReport {bound}}
        ID                          id-package-statusReporting}
```

```
cancelPackage {PARAMETERS-BOUND : bound} OPERATION-PACKAGE ::= {
        CONSUMER INVOKES            {cancel {bound}| cancelStatusReportRequest {bound}}
        ID                          id-package-cancel}

cphResponsePackage  {PARAMETERS-BOUND : bound} OPERATION-PACKAGE ::= {
        CONSUMER INVOKES            {
                                    continueWithArgument {bound}| disconnectLeg  {bound}|
                                    mergeCallSegments  {bound}|
                                    moveCallSegments  {bound}|
                                    moveLeg  {bound}|
                                    createCallSegmentAssociation  {bound} |
                                    reconnect  {bound}|
                                    splitLeg {bound}
                                    }
        ID                          id-package-cphResponse}

exceptionInformPackage   {PARAMETERS-BOUND : bound} OPERATION-PACKAGE ::= {
        CONSUMER INVOKES            {entityReleased {bound}}
        ID                          id-package-entityReleased}

triggerManagementPackage   {PARAMETERS-BOUND : bound} OPERATION-PACKAGE ::= {
        CONSUMER INVOKES            {manageTriggerData {bound}}
        ID                          id-package-triggerManagement}

uSIHandlingPackage   {PARAMETERS-BOUND : bound} OPERATION-PACKAGE ::= {
        CONSUMER INVOKES            {requestReportUTSI {bound}| sendSTUI {bound}}
        SUPPLIER INVOKES            {reportUTSI {bound}}
        ID                          id-package-uSIHandling
        }

facilityIEHandlingPackage   {PARAMETERS-BOUND : bound} OPERATION-PACKAGE ::= {
        CONSUMER INVOKES            {requestReportFacilityEvent {bound}|
                                    sendFacilityInformation {bound}}
        SUPPLIER INVOKES            {eventReportFacility {bound}}
        ID                          id-package-facilityIEHandling
        }


-- Abstract Syntaxes

ssf-scfGenericAbstractSyntax  ABSTRACT-SYNTAX ::= {
        GenericSSF-SCF-PDUs
        IDENTIFIED BY               id-as-ssf-scfGenericAS}

GenericSSF-SCF-PDUs ::= TCMessage {{SsfToScfGenericInvokable},
                                    {SsfToScfGenericReturnable}}

SsfScfGenericInvokable  OPERATION ::= {
                                    activateServiceFiltering  {networkSpecificBoundSet} |
                                    activityTest |
                                    applyCharging  {networkSpecificBoundSet} |
                                    applyChargingReport  {networkSpecificBoundSet} |
                                    callInformationReport  {networkSpecificBoundSet} |
                                    callInformationRequest  {networkSpecificBoundSet} |
                                    cancel  {networkSpecificBoundSet} |
                                    cancelStatusReportRequest  {networkSpecificBoundSet} |
                                    collectInformation  {networkSpecificBoundSet} |
                                    connect  {networkSpecificBoundSet} | connectToResource
                                    {networkSpecificBoundSet} |disconnectForwardConnection |
                                    disconnectForwardConnectionWithArgument
                                    {networkSpecificBoundSet}|
                                    disconnectLeg {networkSpecificBoundSet} |
                                    entityReleased {networkSpecificBoundSet} |
                                    establishTemporaryConnection {networkSpecificBoundSet} |
                                    eventNotificationCharging  {networkSpecificBoundSet} |
                                    eventReportBCSM  {networkSpecificBoundSet} |
                                    eventReportFacility {networkSpecificBoundSet} |
                                    furnishChargingInformation  {networkSpecificBoundSet} |
                                    holdCallInNetwork |
                                    initialDP  {networkSpecificBoundSet} |
                                    mergeCallSegments {networkSpecificBoundSet} |
                                    moveCallSegments {networkSpecificBoundSet} |
                                    moveLeg {networkSpecificBoundSet} |
                                    createCallSegmentAssociation {networkSpecificBoundSet} |
                                    reconnect {networkSpecificBoundSet} |
                                    releaseCall {networkSpecificBoundSet} |
                                    reportUTSI {networkSpecificBoundSet} |
                                    requestCurrentStatusReport  {networkSpecificBoundSet} |
                                    requestEveryStatusChangeReport {networkSpecificBoundSet} |
                                    requestFirstStatusMatchReport  {networkSpecificBoundSet} |
                                    requestNotificationChargingEvent  {networkSpecificBoundSet} |
                                    requestReportBCSMEvent  {networkSpecificBoundSet} |
                                    requestReportFacilityEvent {networkSpecificBoundSet} |
                                    requestReportUTSI {networkSpecificBoundSet} |
                                    resetTimer  {networkSpecificBoundSet} |
                                    sendChargingInformation  {networkSpecificBoundSet} |
                                    sendFacilityInformation {networkSpecificBoundSet} |
                                    sendSTUI {networkSpecificBoundSet} |
                                    serviceFilteringResponse  {networkSpecificBoundSet} |
                                    splitLeg {networkSpecificBoundSet} |
                                    statusReport  {networkSpecificBoundSet} |
                                    playAnnouncement  {networkSpecificBoundSet} |
                                    promptAndCollectUserInformation  {networkSpecificBoundSet} |
                                    scriptClose  {networkSpecificBoundSet} |
                                    scriptEvent  {networkSpecificBoundSet} |
                                    scriptInformation  {networkSpecificBoundSet} |
                                    scriptRun  {networkSpecificBoundSet} |
                                    specializedResourceReport |
                                    promptAndReceiveMessage  {networkSpecificBoundSet}
                                    }

SsfScfGenericReturnable  OPERATION ::= {
                                    activateServiceFiltering  {networkSpecificBoundSet} |
                                    activityTest |
                                    applyCharging  {networkSpecificBoundSet} |
                                    applyChargingReport  {networkSpecificBoundSet} |
                                    callGap  {networkSpecificBoundSet} |
                                    callInformationRequest  {networkSpecificBoundSet} |
                                    cancel  {networkSpecificBoundSet} |
                                    cancelStatusReportRequest  {networkSpecificBoundSet} |
                                    collectInformation  {networkSpecificBoundSet} |
                                    connect  {networkSpecificBoundSet} |
                                    connectToResource  {networkSpecificBoundSet} |
                                    continue |
                                    continueWithArgument  {networkSpecificBoundSet}|
                                    disconnectForwardConnection |
                                    disconnectForwardConnectionWithArgument
                                    {networkSpecificBoundSet}|
                                    disconnectLeg {networkSpecificBoundSet}|
                                    establishTemporaryConnection {networkSpecificBoundSet}|
                                    furnishChargingInformation  {networkSpecificBoundSet}|
                                    holdCallInNetwork |
                                    initialDP  {networkSpecificBoundSet}|
                                    mergeCallSegments {networkSpecificBoundSet}|
                                    moveCallSegments {networkSpecificBoundSet}|
                                    moveLeg {networkSpecificBoundSet}|
                                    createCallSegmentAssociation {networkSpecificBoundSet}|
                                    reconnect {networkSpecificBoundSet}|
                                    releaseCall {networkSpecificBoundSet}|
                                    requestCurrentStatusReport  {networkSpecificBoundSet}|
                                    requestEveryStatusChangeReport {networkSpecificBoundSet}|
                                    requestFirstStatusMatchReport  {networkSpecificBoundSet}|
                                    requestNotificationChargingEvent  {networkSpecificBoundSet}|
                                    requestReportBCSMEvent  {networkSpecificBoundSet}|
                                    requestReportFacilityEvent {networkSpecificBoundSet}|
                                    requestReportUTSI {networkSpecificBoundSet}|
                                    resetTimer  {networkSpecificBoundSet}|
                                    sendChargingInformation {networkSpecificBoundSet}|
                                    sendFacilityInformation {networkSpecificBoundSet}|
                                    sendSTUI {networkSpecificBoundSet}|
                                    splitLeg {networkSpecificBoundSet}|
                                    playAnnouncement  {networkSpecificBoundSet}|
                                    promptAndCollectUserInformation  {networkSpecificBoundSet}|
                                    scriptClose  {networkSpecificBoundSet}|
                                    scriptInformation  {networkSpecificBoundSet}|
                                    scriptRun  {networkSpecificBoundSet}|
                                    promptAndReceiveMessage  {networkSpecificBound Set}
                                    }

ssf-scfDpSpecificAbstractSyntax  ABSTRACT-SYNTAX ::= {
     DpSpecificSSF-SCF-PDUs
     IDENTIFIED BY                      id-as-ssf-scfDpSpecificAS}

DpSpecificSSF-SCF-PDUs ::= TCMessage {{SsfToScfDpSpecificInvokable},
                                    {SsfToScfDpSpecificReturnable}}

SsfToScfDpSpecificInvokable  OPERATION ::= {
                                    activateServiceFiltering  {networkSpecificBoundSet}|
                                    activityTest |
                                    analyseInformation  {networkSpecificBoundSet}|
                                    analysedInformation  {networkSpecificBoundSet}|
                                    applyCharging  {networkSpecificBoundSet}|
                                    applyChargingReport  {networkSpecificBoundSet}|
                                    assistRequestInstructions  {networkSpecificBoundSet}|
                                    callInformationReport  {networkSpecificBoundSet}|
                                    callInformationRequest  {networkSpecificBoundSet}|
                                    cancel  {networkSpecificBoundSet}|
                                    cancelStatusReportRequest  {networkSpecificBoundSet}|
                                    collectedInformation  {networkSpecificBoundSet}|
                                    collectInformation  {networkSpecificBoundSet}|
                                    connect  {networkSpecificBoundSet}|
                                    connectToResource  {networkSpecificBoundSet}|
                                    disconnectForwardConnection |
                                    disconnectForwardConnectionWithArgument
                                    {networkSpecificBoundSet}|
                                    disconnectLeg {networkSpecificBoundSet}|
                                    entityReleased {networkSpecificBoundSet}|
                                    establishTemporaryConnection {networkSpecificBoundSet}|
                                    eventNotificationCharging  {networkSpecificBoundSet}|
                                    eventReportFacility {networkSpecificBoundSet}|
                                    furnishChargingInformation  {networkSpecificBoundSet}|
                                    holdCallInNetwork |
                                    initiateCallAttempt {networkSpecificBoundSet}|
                                    mergeCallSegments {networkSpecificBoundSet}|
                                    moveCallSegments {networkSpecificBoundSet}|
                                    moveLeg {networkSpecificBoundSet}|
                                    oAbandon  {networkSpecificBoundSet}|
                                    oAnswer  {networkSpecificBoundSet}|
                                    oCalledPartyBusy  {networkSpecificBoundSet}|
                                    oDisconnect  {networkSpecificBoundSet}|
                                    oMidCall  {networkSpecificBoundSet}|
                                    oNoAnswer  {networkSpecificBoundSet}|
                                    createCallSegmentAssociation {networkSpecificBoundSet}|
                                    originationAttemptAuthorized  {networkSpecificBoundSet}|
                                    reconnect {networkSpecificBoundSet}|
                                    releaseCall {networkSpecificBoundSet} |
                                    reportUTSI {networkSpecificBoundSet}|
                                    requestCurrentStatusReport  {networkSpecificBoundSet}|
                                    requestEveryStatusChangeReport {networkSpecificBoundSet}|
                                    requestFirstStatusMatchReport  {networkSpecificBoundSet}|
                                    requestNotificationChargingEvent  {networkSpecificBoundSet}|
                                    requestReportBCSMEvent  {networkSpecificBoundSet}|
                                    requestReportFacilityEvent {networkSpecificBoundSet}|
                                    requestReportUTSI {networkSpecificBoundSet}|
                                    resetTimer  {networkSpecificBoundSet}|
                                    routeSelectFailure  {networkSpecificBoundSet}|
                                    selectFacility  {networkSpecificBoundSet}|
                                    selectRoute  {networkSpecificBoundSet}|
                                    sendChargingInformation  {networkSpecificBoundSet}|
                                    sendFacilityInformation {networkSpecificBoundSet}|
                                    sendSTUI {networkSpecificBoundSet}|
                                    serviceFilteringResponse  {networkSpecificBoundSet}|
                                    splitLeg {networkSpecificBoundSet}|
                                    statusReport  {networkSpecificBoundSet}|
                                    tAnswer  {networkSpecificBoundSet}|
                                    tBusy  {networkSpecificBoundSet}|
                                    tDisconnect  {networkSpecificBoundSet} |
                                    termAttemptAuthorized  {networkSpecificBoundSet}|
                                    tMidCall  {networkSpecificBoundSet}|
                                    tNoAnswer  {networkSpecificBoundSet} |
                                    playAnnouncement  {networkSpecificBoundSet}|
                                    promptAndCollectUserInformation  {networkSpecificBoundSet}|
                                    scriptClose  {networkSpecificBoundSet}|
                                    scriptEvent  {networkSpecificBoundSet}|
                                    scriptInformation  {networkSpecificBoundSet}|
                                    scriptRun  {networkSpecificBoundSet}|
                                    specializedResourceReport |
                                    promptAndReceiveMessage  {networkSpecificBoundSet}
                                    }

SsfToScfDpSpecificReturnable  OPERATION ::= {
                                    activateServiceFiltering  {networkSpecificBoundSet}|
                                    activityTest |
                                    analyseInformation  {networkSpecificBoundSet}|
                                    analysedInformation  {networkSpecificBoundSet}|
                                    applyCharging  {networkSpecificBoundSet}|
                                    applyChargingReport  {networkSpecificBoundSet}|
                                    assistRequestInstructions  {networkSpecificBoundSet}|
                                    callGap  {networkSpecificBoundSet}|
                                    callInformationRequest  {networkSpecificBoundSet} |
                                    cancel  {networkSpecificBoundSet} |
                                    cancelStatusReportRequest {networkSpecificBoundSet} |
                                    collectedInformation  {networkSpecificBoundSet} |
                                    collectInformation  {networkSpecificBoundSet} |
                                    connect  {networkSpecificBoundSet} |
                                    connectToResource  {networkSpecificBoundSet} |
                                    continue |
                                    continueWithArgument {networkSpecificBoundSet} |
                                    disconnectForwardConnection |
                                    disconnectForwardConnectionWithArgument
                                    {networkSpecificBoundSet} |
                                    disconnectLeg {networkSpecificBoundSet} |
                                    establishTemporaryConnection {networkSpecificBoundSet} |
                                    furnishChargingInformation {networkSpecificBoundSet} |
                                    holdCallInNetwork |
                                    initiateCallAttempt  {networkSpecificBoundSet}|
                                    mergeCallSegments {networkSpecificBoundSet}|
                                    moveCallSegments {networkSpecificBoundSet}|
                                    moveLeg {networkSpecificBoundSet}|
                                    oAbandon  {networkSpecificBoundSet}|
                                    oAnswer  {networkSpecificBoundSet}|
                                    oCalledPartyBusy  {networkSpecificBoundSet}|
                                    oDisconnect  {networkSpecificBoundSet}|
                                    oMidCall  {networkSpecificBoundSet}|
                                    oNoAnswer  {networkSpecificBoundSet}|
                                    createCallSegmentAssociation  {networkSpecificBoundSet}|
                                    originationAttemptAuthorized  {networkSpecificBoundSet}|
                                    reconnect {networkSpecificBoundSet}|
                                    releaseCall {networkSpecificBoundSet}|
                                    requestCurrentStatusReport  {networkSpecificBoundSet}|
                                    requestEveryStatusChangeReport {networkSpecificBoundSet}|
                                    requestFirstStatusMatchReport  {networkSpecificBoundSet}|
                                    requestNotificationChargingEvent  {networkSpecificBoundSet}|
                                    requestReportBCSMEvent  {networkSpecificBoundSet}|
                                    requestReportFacilityEvent {networkSpecificBoundSet}|
                                    requestReportUTSI {networkSpecificBoundSet}|
                                    resetTimer  {networkSpecificBoundSet}|
                                    routeSelectFailure  {networkSpecificBoundSet}|
                                    selectFacility  {networkSpecificBoundSet}|
                                    selectRoute  {networkSpecificBoundSet}|
                                    sendChargingInformation {networkSpecificBoundSet}|
                                    sendFacilityInformation {networkSpecificBoundSet}|
                                    sendSTUI {networkSpecificBoundSet}|
                                    splitLeg {networkSpecificBoundSet}|
                                    tAnswer  {networkSpecificBoundSet}|
                                    tBusy  {networkSpecificBoundSet}|
                                    tDisconnect  {networkSpecificBoundSet}|
                                    termAttemptAuthorized  {networkSpecificBoundSet}|
                                    tMidCall  {networkSpecificBoundSet}|
                                    tNoAnswer  {networkSpecificBoundSet}|
                                    playAnnouncement  {networkSpecificBoundSet}|
                                    promptAndCollectUserInformation  {networkSpecificBoundSet}|
                                    scriptClose  {networkSpecificBoundSet}|
                                    scriptInformation  {networkSpecificBoundSet}|
                                    scriptRun  {networkSpecificBoundSet}|
                                    promptAndReceiveMessage {networkSpecificBoundSet}
                                    }

assistHandoff-ssf-scfAbstractSyntax  ABSTRACT-SYNTAX ::= {
     AssistHandoffSSF-SCF-PDUs
     IDENTIFIED BY                      id-as-assistHandoff-ssf-scfAS}

AssistHandoffSSF-SCF-PDUs ::= TCMessage {{AssistHandoffSsfToScfInvokable},
                                    {AssistHandoffSsfToScfReturnable}}
```

```
AssistHandoffSsfToScfInvokable OPERATION ::= {
                                activityTest |
                                applyCharging  {networkSpecificBoundSet}|
                                applyChargingReport  {networkSpecificBoundSet}|
                                assistRequestInstructions  {networkSpecificBoundSet}|
                                cancel  {networkSpecificBoundSet}|
                                cancelStatusReportRequest  {networkSpecificBoundSet}|
                                connectToResource  {networkSpecificBoundSet}|
                                disconnectForwardConnection |
                                disconnectForwardConnectionWithArgument
                                {networkSpecificBoundSet}|
                                furnishChargingInformation  {networkSpecificBoundSet}|
                                holdCallInNetwork |
                                playAnnouncement  {networkSpecificBoundSet}|
                                promptAndCollectUserInformation  {networkSpecificBoundSet}|
                                requestCurrentStatusReport  {networkSpecificBoundSet}|
                                requestEveryStatusChangeReport {networkSpecificBoundSet}|
                                requestFirstStatusMatchReport  {networkSpecificBoundSet}|
                                resetTimer  {networkSpecificBoundSet}|
                                statusReport  {networkSpecificBoundSet}|
                                scriptClose  {networkSpecificBoundSet}|
                                scriptEvent  {networkSpecificBoundSet}|
                                scriptInformation  {networkSpecificBoundSet}|
                                scriptRun  {networkSpecificBoundSet}|
                                specializedResourceReport |
                                promptAndReceiveMessage {networkSpecificBoundSet}
                                }

AssistHandoffSsfToScfReturnable OPERATION ::= {
                                activityTest |
                                applyCharging {networkSpecificBoundSet}|
                                applyChargingReport {networkSpecificBoundSet}|
                                assistRequestInstructions {networkSpecificBoundSet}|
                                cancel {networkSpecificBoundSet}|
                                cancelStatusReportRequest {networkSpecificBoundSet}|
                                connectToResource {networkSpecificBoundSet}|
                                disconnectForwardConnection |
                                disconnectForwardConnectionWithArgument
                                {networkSpecificBoundSet}|
                                furnishChargingInformation {networkSpecificBoundSet}|
                                holdCallInNetwork |
                                playAnnouncement {networkSpecificBoundSet}|
                                promptAndCollectUserInformation {networkSpecificBoundSet}|
                                requestCurrentStatusReport {networkSpecificBoundSet}|
                                requestEveryStatusChangeReport {networkSpecificBoundSet}|
                                requestFirstStatusMatchReport {networkSpecificBoundSet}|
                                resetTimer {networkSpecificBoundSet}|
                                scriptClose {networkSpecificBoundSet}|
                                scriptInformation {networkSpecificBoundSet}|
                                scriptRun {networkSpecificBoundSet}|
                                promptAndReceiveMessage {networkSpecificBoundSet}
                                }


scf-ssfGenericAbstractSyntax  ABSTRACT-SYNTAX ::= {
     GenericSCF-SSF-PDUs
     IDENTIFIED BY              id-as-scf-ssfGenericAS}

GenericSCF-SSF-PDUs ::= TCMessage {{ScfToSsfGenericInvokable}, {ScfToSsfGenericReturnable}}
```

```
ScfSsfGenericInvokable  OPERATION ::= {
                        activateServiceFiltering {networkSpecificBoundSet}|
                        activityTest |
                        applyCharging {networkSpecificBoundSet}|
                        applyChargingReport {networkSpecificBoundSet}|
                        callInformationRequest {networkSpecificBoundSet}|
                        cancel {networkSpecificBoundSet}|
                        cancelStatusReportRequest {networkSpecificBoundSet}|
                        collectInformation {networkSpecificBoundSet}|
                        connect {networkSpecificBoundSet}|
                        connectToResource {networkSpecificBoundSet}|
                        continue |
                        continueWithArgument{networkSpecificBoundSet}|
                        disconnectForwardConnection |
                        disconnectForwardConnectionWithArgument
                        {networkSpecificBoundSet}|
                        disconnectLeg {networkSpecificBoundSet}|
                        establishTemporaryConnection {networkSpecificBoundSet}|
                        furnishChargingInformation  {networkSpecificBoundSet}|
                        holdCallInNetwork |
                        initiateCallAttempt  {networkSpecificBoundSet}|
                        mergeCallSegments {networkSpecificBoundSet}|
                        moveCallSegments {networkSpecificBoundSet}|
                        moveLeg {networkSpecificBoundSet}|
                        createCallSegmentAssociation {networkSpecificBoundSet}|
                        releaseCall {networkSpecificBoundSet}|
                        reconnect {networkSpecificBoundSet}|
                        requestCurrentStatusReport  {networkSpecificBoundSet}|
                        requestEveryStatusChangeReport {networkSpecificBoundSet}|
                        requestFirstStatusMatchReport  {networkSpecificBoundSet}|
                        requestNotificationChargingEvent  {networkSpecificBoundSet}|
                        requestReportBCSMEvent  {networkSpecificBoundSet}|
                        requestReportFacilityEvent {networkSpecificBoundSet}|
                        requestReportUTSI {networkSpecificBoundSet}|
                        resetTimer  {networkSpecificBoundSet}|
                        sendChargingInformation  {networkSpecificBoundSet}|
                        sendFacilityInformation {networkSpecificBoundSet}|
                        sendSTUI {networkSpecificBoundSet}|
                        splitLeg  {networkSpecificBoundSet}|
                        playAnnouncement  {networkSpecificBoundSet}|
                        promptAndCollectUserInformation  {networkSpecificBoundSet}|
                        scriptClose  {networkSpecificBoundSet}|
                        scriptInformation  {networkSpecificBoundSet}|
                        scriptRun  {networkSpecificBoundSet}|
                        promptAndReceiveMessage  {networkSpecificBoundSet}
                        }

ScfSsfGenericReturnable  OPERATION ::= {
                        activateServiceFiltering  {networkSpecificBoundSet}|
                        activityTest |
                        applyCharging  {networkSpecificBoundSet}|
                        applyChargingReport  {networkSpecificBoundSet}|
                        callInformationReport  {networkSpecificBoundSet}|
                        callInformationRequest  {networkSpecificBoundSet}|
                        cancel  {networkSpecificBoundSet}|
                        cancelStatusReportRequest  {networkSpecificBoundSet}|
                        collectInformation  {networkSpecificBoundSet}|
                        connect  {networkSpecificBoundSet}|
                        connectToResource  {networkSpecificBoundSet}|
                        disconnectForwardConnection |
                        disconnectForwardConnectionWithArgument
```

```
                                                    {networkSpecificBoundSet}|
                                                    disconnectLeg {networkSpecificBoundSet}|
                                                    entityReleased {networkSpecificBoundSet}|
                                                    establishTemporaryConnection {networkSpecificBoundSet}|
                                                    eventNotificationCharging {networkSpecificBoundSet} |
                                                    resetTimer {networkSpecificBoundSet}|
                                                    eventReportBCSM  {networkSpecificBoundSet}|
                                                    eventReportFacility {networkSpecificBoundSet}|
                                                    furnishChargingInformation  {networkSpecificBoundSet}|
                                                    holdCallInNetwork |
                                                    initiateCallAttempt  {networkSpecificBoundSet}|
                                                    mergeCallSegments {networkSpecificBoundSet}|
                                                    moveCallSegments {networkSpecificBoundSet}|
                                                    moveLeg {networkSpecificBoundSet}|
                                                    createCallSegmentAssociation {networkSpecificBoundSet}|
                                                    reconnect {networkSpecificBoundSet}|
                                                    reportUTSI {networkSpecificBoundSet}|
                                                    requestCurrentStatusReport  {networkSpecificBoundSet}|
                                                    requestEveryStatusChangeReport {networkSpecificBoundSet}|
                                                    requestFirstStatusMatchReport  {networkSpecificBoundSet}|
                                                    requestNotificationChargingEvent  {networkSpecificBoundSet}|
                                                    requestReportBCSMEvent  {networkSpecificBoundSet}|
                                                    requestReportFacilityEvent {networkSpecificBoundSet}|
                                                    requestReportUTSI {networkSpecificBoundSet} |
                                                    sendChargingInformation  {networkSpecificBoundSet}|
                                                    sendFacilityInformation {networkSpecificBoundSet}|
                                                    sendSTUI {networkSpecificBoundSet}|
                                                    serviceFilteringResponse  {networkSpecificBoundSet}|
                                                    splitLeg {networkSpecificBoundSet}|
                                                    statusReport  {networkSpecificBoundSet}|
                                                    playAnnouncement  {networkSpecificBoundSet}|
                                                    promptAndCollectUserInformation  {networkSpecificBoundSet}|
                                                    scriptClose  {networkSpecificBoundSet}|
                                                    scriptEvent  {networkSpecificBoundSet}|
                                                    scriptInformation  {networkSpecificBoundSet}|
                                                    scriptRun  {networkSpecificBoundSet}|
                                                    specializedResourceReport |
                                                    promptAndReceiveMessage  {networkSpecificBoundSet}
                                                    }


scf-ssfDpSpecificAbstractSyntax  ABSTRACT-SYNTAX ::= {
        DpSpecificSCF-SSF-PDUs
        IDENTIFIED BY                               id-as-scf-ssfDpSpecificAS}

DpSpecificSCF-SCF-PDUs ::= TCMessage {{ScfToSsfDpSpecificInvokable},
                                                    {ScfToSsfDpSpecificReturnable}}

ScfSsfDpSpecificInvokable  OPERATION ::= {
                                                    activateServiceFiltering  {networkSpecificBoundSet}|
                                                    activityTest |
                                                    analyseInformation {networkSpecificBoundSet} |
                                                    analysedInformation {networkSpecificBoundSet}|
                                                    applyCharging {networkSpecificBoundSet} |
                                                    applyChargingReport {networkSpecificBoundSet} |
                                                    callInformationRequest {networkSpecificBoundSet} |
                                                    cancel {networkSpecificBoundSet} |
                                                    cancelStatusReportRequest {networkSpecificBoundSet} |
                                                    collectedInformation {networkSpecificBoundSet} |
                                                    collectInformation {networkSpecificBoundSet} |
                                                    connect {networkSpecificBoundSet}|
                                                    connectToResource {networkSpecificBoundSet} |
                                                    continue |
                                                    continueWithArgument {networkSpecificBoundSet}|
                                                    disconnectForwardConnection |
                                                    disconnectForwardConnectionWithArgument
                                                    {networkSpecificBoundSet}|
                                                    disconnectLeg {networkSpecificBoundSet}|
                                                    establishTemporaryConnection {networkSpecificBoundSet}|
                                                    furnishChargingInformation  {networkSpecificBoundSet}|
                                                    holdCallInNetwork |
                                                    initiateCallAttempt {networkSpecificBoundSet}|
                                                    mergeCallSegments  {networkSpecificBoundSet}|
                                                    moveCallSegments {networkSpecificBoundSet}|
                                                    moveLeg {networkSpecificBoundSet}|
                                                    oAbandon {networkSpecificBoundSet}|
                                                    oAnswer {networkSpecificBoundSet}|
                                                    oCalledPartyBusy {networkSpecificBoundSet}|
                                                    oDisconnect {networkSpecificBoundSet}|
                                                    oMidCall {networkSpecificBoundSet}|
                                                    oNoAnswer {networkSpecificBoundSet}|
                                                    createCallSegmentAssociation {networkSpecificBoundSet}|
                                                    originationAttemptAuthorized  {networkSpecificBoundSet}|
                                                    reconnect {networkSpecificBoundSet}|
                                                    releaseCall {networkSpecificBoundSet}|
                                                    requestCurrentStatusReport  {networkSpecificBoundSet}|
                                                    requestEveryStatusChangeReport {networkSpecificBoundSet}|
                                                    requestFirstStatusMatchReport  {networkSpecificBoundSet}|
                                                    requestNotificationChargingEvent  {networkSpecificBoundSet}|
                                                    requestReportBCSMEvent  {networkSpecificBoundSet}|
                                                    requestReportFacilityEvent {networkSpecificBoundSet}|
                                                    requestReportUTSI {networkSpecificBoundSet}|
                                                    resetTimer  {networkSpecificBoundSet}|
                                                    routeSelectFailure  {networkSpecificBoundSet}|
                                                    selectFacility  {networkSpecificBoundSet}|
                                                    selectRoute  {networkSpecificBoundSet}|
                                                    sendChargingInformation  {networkSpecificBoundSet}|
                                                    sendFacilityInformation {networkSpecificBoundSet}|
                                                    sendSTUI {networkSpecificBoundSet}|
                                                    splitLeg {networkSpecificBoundSet}|
                                                    tAnswer  {networkSpecificBoundSet}|
                                                    tBusy  {networkSpecificBoundSet}|
                                                    tDisconnect  {networkSpecificBoundSet}|
                                                    termAttemptAuthorized  {networkSpecificBoundSet}|
                                                    tMidCall  {networkSpecificBoundSet}|
                                                    tNoAnswer  {networkSpecificBoundSet}|
                                                    playAnnouncement  {networkSpecificBoundSet}|
                                                    promptAndCollectUserInformation  {networkSpecificBoundSet}|
                                                    scriptClose  {networkSpecificBoundSet}|
                                                    scriptInformation  {networkSpecificBoundSet}|
                                                    scriptRun  {networkSpecificBoundSet}|
                                                    promptAndReceiveMessage  {networkSpecificBoundSet}
                                                    }

ScfSsfDpSpecificReturnable  OPERATION ::= {
                                                    activateServiceFiltering  {networkSpecificBoundSet}|
                                                    activityTest |
                                                    analyseInformation  {networkSpecificBoundSet}|
                                                    analysedInformation  {networkSpecificBoundSet}|
                                                    applyCharging  {networkSpecificBoundSet}|
                                                    applyChargingReport  {networkSpecificBoundSet}|
                                                    callInformationReport  {networkSpecificBoundSet}|
callInformationRequest  {networkSpecificBoundSet}|
cancel  {networkSpecificBoundSet}|
cancelStatusReportRequest  {networkSpecificBoundSet}|
collectedInformation  {networkSpecificBoundSet}|
collectInformation  {networkSpecificBoundSet}|
connect  {networkSpecificBoundSet}|
connectToResource  {networkSpecificBoundSet}|
disconnectForwardConnection |
disconnectForwardConnectionWithArgument
{networkSpecificBoundSet}|
disconnectLeg {networkSpecificBoundSet}|
entityReleased {networkSpecificBoundSet}|
establishTemporaryConnection {networkSpecificBoundSet}|
eventNotificationCharging  {networkSpecificBoundSet}|
eventReportFacility {networkSpecificBoundSet}|
furnishChargingInformation  {networkSpecificBoundSet}|
holdCallInNetwork |
initiateCallAttempt  {networkSpecificBoundSet}|
initiateCallAttempt {networkSpecificBoundSet}|
mergeCallSegments {networkSpecificBoundSet}|
moveCallSegments {networkSpecificBoundSet}|
moveLeg {networkSpecificBoundSet}|
oAnswer  {networkSpecificBoundSet}|
oCalledPartyBusy  {networkSpecificBoundSet}|
oDisconnect  {networkSpecificBoundSet}|
oMidCall  {networkSpecificBoundSet}|
oAbandon {networkSpecificBoundSet}|
oNoAnswer  {networkSpecificBoundSet}|
createCallSegmentAssociation {networkSpecificBoundSet}|
originationAttemptAuthorized  {networkSpecificBoundSet}|
reconnect {networkSpecificBoundSet}|
reportUTSI {networkSpecificBoundSet}|
requestCurrentStatusReport  {networkSpecificBoundSet}|
requestEveryStatusChangeReport {networkSpecificBoundSet}|
requestFirstStatusMatchReport  {networkSpecificBoundSet}|
requestNotificationChargingEvent  {networkSpecificBoundSet}|
requestReportBCSMEvent  {networkSpecificBoundSet}|
requestReportFacilityEvent {networkSpecificBoundSet}|
requestReportUTSI {networkSpecificBoundSet}|
resetTimer  {networkSpecificBoundSet}|
routeSelectFailure  {networkSpecificBoundSet}|
selectFacility  {networkSpecificBoundSet}|
selectRoute  {networkSpecificBoundSet}|
sendChargingInformation  {networkSpecificBoundSet}|
sendFacilityInformation {networkSpecificBoundSet}|
sendSTUI {networkSpecificBoundSet}|
serviceFilteringResponse  {networkSpecificBoundSet}|
splitLeg {networkSpecificBoundSet}|
statusReport  {networkSpecificBoundSet}|
tAnswer  {networkSpecificBoundSet}|
tBusy  {networkSpecificBoundSet}|
tDisconnect  {networkSpecificBoundSet}|
termAttemptAuthorized  {networkSpecificBoundSet}|
tMidCall  {networkSpecificBoundSet}|
tNoAnswer  {networkSpecificBoundSet}|
playAnnouncement  {networkSpecificBoundSet}|
promptAndCollectUserInformation  {networkSpecificBoundSet}|
scriptClose  {networkSpecificBoundSet}|
scriptEvent  {networkSpecificBoundSet}|
                                        scriptInformation  {networkSpecificBoundSet}|
                                        scriptRun  {networkSpecificBoundSet}|
                                        specializedResourceReport |
                                        promptAndReceiveMessage  {networkSpecificBoundSet}
                                        }


scf-ssfTrafficManagementAbstractSyntax  ABSTRACT-SYNTAX ::= {
        TrafficManagementSCF-SSF-PDUs
        IDENTIFIED BY                   id-as-scf-ssfTrafficManagementAS}

TrafficManagementSCF-SSF-PDUs ::= TCMessage {{ScfToSsfTrafficManagementInvokable}}

ScfToSsfTrafficManagementInvokable OPERATION ::= {
        callGap {networkSpecificBoundSet}
        }


scf-ssfServiceManagementAbstractSyntax  ABSTRACT-SYNTAX ::= {
        ServiceManagementSCF-SSF-PDUs
        IDENTIFIED BY                   id-as-scf-ssfServiceManagementAS}

ServiceManagementSCF-SSF-PDUs ::= TCMessage {{ScfToSsfServiceManagementInvokable},
                                        {ScfToSsfServiceManagementReturnable}}

ScfToSsfServiceManagementInvokable OPERATION ::= {
        activateServiceFiltering {networkSpecificBoundSet}
        }

ScfToSsfServiceManagementReturnable OPERATION ::= {
        activateServiceFiltering {networkSpecificBoundSet}
        }

ssf-scfServiceManagementAbstractSyntax  ABSTRACT-SYNTAX ::= {
        ServiceManagementSSF-SCF-PDUs
        IDENTIFIED BY                   id-as-ssf-scfServiceManagementAS}

ServiceManagementSSF-SCF-PDUs ::= TCMessage {{SsfToScfServiceManagementInvokable}}

SsfToScfServiceManagementInvokable OPERATION ::= {
        serviceFilteringResponse {networkSpecificBoundSet}
        }


scf-ssfStatusReportingAbstractSyntax  ABSTRACT-SYNTAX ::= {
        StatusReportingSCF-SSF-PDUs
        IDENTIFIED BY                   id-as-scf-ssfStatusReportingAS}

StatusReportingSCF-SSF-PDUs ::= TCMessage {{ScfToSsfStatusReportingInvokable},
                                    {ScfToSsfStatusReportingReturnable}}

ScfToSsfStatusReportingInvokable OPERATION  ::= {
                                    cancel  {networkSpecificBoundSet}|
                                    cancelStatusReportRequest  {networkSpecificBoundSet}|
                                    requestCurrentStatusReport  {networkSpecificBoundSet}|
                                    requestEveryStatusChangeReport {networkSpecificBoundSet}|
                                    requestFirstStatusMatchReport {networkSpecificBoundSet}
                                    }
```

```
ScfToSsfStatusReportingReturnable OPERATION  ::= {
                                   cancel  {networkSpecificBoundSet}|
                                   cancelStatusReportRequest  {networkSpecificBoundSet}|
                                   requestCurrentStatusReport  {networkSpecificBoundSet}|
                                   requestEveryStatusChangeReport {networkSpecificBoundSet}|
                                   requestFirstStatusMatchReport  {networkSpecificBoundSet}|
                                   statusReport {networkSpecificBoundSet}
                                   }

scf-ssfTriggerManagementAbstractSyntax  ABSTRACT-SYNTAX ::= {
       TriggerManagementSCF-SSF-PDUs
       IDENTIFIED BY               id-as-scf-ssfTriggerManagementAS}

TriggerManagementSCF-SSF-PDUs ::= TCMessage {{ScfToSsfTriggerManagementInvokable},
                                   {ScfToSsfTriggerManagementReturnable}}

ScfToSsfTriggerManagementInvokable OPERATION ::= {
       manageTriggerData {networkSpecificBoundSet}
       }

ScfToSsfTriggerManagementReturnable OPERATION ::= {
       manageTriggerData {networkSpecificBoundSet}
       }

END
```

6 SCF/SRF interface

6.1 SCF/SRF operations and arguments

```
IN-CS2-SCF-SRF-ops-args {itu-t recommendation q 1228 modules(0) in-cs2-scf-srf-ops-args (7) version1(0)}

DEFINITIONS IMPLICIT TAGS ::=

BEGIN

IMPORTS

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

       opcode-playAnnouncement,
       opcode-promptAndCollectUserInformation,
       opcode-promptAndReceiveMessage,
       opcode-scriptClose,
       opcode-scriptEvent,
       opcode-scriptInformation,
       opcode-scriptRun,
       opcode-specializedResourceReport

FROM IN-CS2-operationcodes operationcodes

       CallSegmentID {},
       CollectedInfo,
       Digits {},
       ExtensionField {},
       InformationToRecord {},
       InformationToSend {},
       LegID,
       MailBoxID {},
       Media,
       GenericNumber {},
       ReceivedStatus,
       RecordedMessageID

FROM IN-CS2-datatypes datatypes

       cancelled,
       improperCallerResponse,
       missingParameter,
       parameterOutOfRange,
       systemFailure,
       taskRefused,
       unavailableResource,
       unexpectedComponentSequence,
       unexpectedDataValue,
       unexpectedParameter

FROM IN-CS2-errortypes errortypes

       UISCRIPT,
       SupportedUIScripts {},
       PARAMETERS-BOUND

FROM IN-CS2-classes classes

       ros-InformationObjects, operationcodes, datatypes, errortypes, classes

FROM IN-CS2-object-identifiers
{itu-t recommendation q 1228 modules(0) in-cs2-object-identifiers(17) version1(0)}
;

playAnnouncement {PARAMETERS-BOUND : bound} OPERATION ::= {
       ARGUMENT            PlayAnnouncementArg { bound}
       RETURN RESULT       FALSE
       ERRORS              {cancelled |
                           missingParameter |
                           parameterOutOfRange |
                           systemFailure |
                           taskRefused |
                           unexpectedComponentSequence |
                           unexpectedDataValue |
                           unexpectedParameter |
                           unavailableResource
                           }
       LINKED              {specializedResourceReport}
       CODE                opcode-playAnnouncement
       }
-- Direction: SCF → SRF, Timer: Tpa
-- This operation is to be used after Establish Temporary Connection (assist procedure with a second SSP)
-- or a Connect to Resource (no assist) operation. It may be used for in-band interaction with an analogue user,
-- or for interaction with an ISDN user. In the former case, the SRF is usually collocated with the SSF for
-- standard tones (congestion tone...) or standard announcements. In the latter case, the SRF is always
-- collocated with the SSF in the switch. Any error is returned to the SCF. The timer associated with this
-- operation must be of a sufficient duration to allow its linked operation to be correctly correlated.
```

```
PlayAnnouncementArg {PARAMETERS-BOUND : bound} ::= SEQUENCE {
        informationToSend [0] InformationToSend {bound},
        disconnectFromIPForbidden     [1] BOOLEAN                          DEFAULT TRUE,
        requestAnnouncementComplete   [2] BOOLEAN                          DEFAULT TRUE,
        extensions                    [3] SEQUENCE SIZE(1..bound.&numOfExtensions) OF
                                          ExtensionField  {bound}          OPTIONAL,
        connectedParty                CHOICE {
                                      legID                                [4] LegID,
                                      callSegmentID                        [5] CallSegmentID {bound}
                                      }                                    OPTIONAL,
        ...
        }

promptAndCollectUserInformation {PARAMETERS-BOUND : bound} OPERATION ::= {
        ARGUMENT                      PromptAndCollectUserInformationArg { bound}
        RESULT                        ReceivedInformationArg { bound}
        ERRORS                        {cancelled |
                                      improperCallerResponse |
                                      missingParameter |
                                      parameterOutOfRange |
                                      systemFailure |
                                      taskRefused |
                                      unexpectedComponentSequence |
                                      unavailableResource |
                                      unexpectedDataValue |
                                      unexpectedParameter
                                      }
        CODE                          opcode-promptAndCollectUserInformation
        }
```
-- Direction: SCF $\rightarrow$ SRF, Timer: T_{pc}
-- This operation is used to interact with a user to collect information.

```
PromptAndCollectUserInformationArg {PARAMETERS-BOUND : bound} ::= SEQUENCE {
        collectedInfo                 [0] CollectedInfo,
        disconnectFromIPForbidden     [1] BOOLEAN                          DEFAULT TRUE,

        informationToSend             [2] InformationToSend {bound}        OPTIONAL,
        extensions                    [3] SEQUENCE SIZE(1..bound.&numOfExtensions) OF
                                          ExtensionField  {bound}          OPTIONAL,
        callSegmentID                 [4] CallSegmentID {bound}            OPTIONAL,
        ...
        }

ReceivedInformationArg {PARAMETERS-BOUND : bound} ::= CHOICE {
        digitsResponse                [0] Digits {bound},
        iA5Response                   [1] IA5String
        }

promptAndReceiveMessage {PARAMETERS-BOUND : bound} OPERATION ::= {
        ARGUMENT                      PromptAndReceiveMessageArg { bound}
        RESULT                        MessageReceivedArg { bound}
        ERRORS                        {cancelled |
                                      improperCallerResponse |
                                      missingParameter |
                                      parameterOutOfRange |
                                      taskRefused |
                                      systemFailure |
                                      unavailableResource |
                                        unexpectedComponentSequence |
                                        unexpectedDataValue |
                                        unexpectedParameter
                                        }
     CODE                               opcode-promptAndReceiveMessage
     }
-- Direction: SCF → SRF, Timer: Tprm
-- Used to prompt a user to store a message

PromptAndReceiveMessageArg {PARAMETERS-BOUND : bound} ::= SEQUENCE {
     disconnectFromIPForbidden          [0] BOOLEAN                            DEFAULT TRUE,
     informationToSend                  [1] InformationToSend {bound}          OPTIONAL,
     extensions                         [3] SEQUENCE SIZE(0..bound.&numOfExtensions) OF
                                            ExtensionField {bound}             OPTIONAL,
     subscriberID                       [4] GenericNumber {bound}              OPTIONAL,
     mailBoxID                          [5] MailBoxID {bound}                  OPTIONAL,
     informationToRecord                [6] InformationToRecord {bound},
     media                              [7] Media                              DEFAULT voiceMail,
     callSegmentID                      [8] CallSegmentID {bound}              OPTIONAL,
     ...
     }

MessageReceivedArg {PARAMETERS-BOUND : bound}    ::= SEQUENCE {
     receivedStatus                     [0] ReceivedStatus,
     recordedMessageID                  [1] RecordedMessageID                  OPTIONAL,
     recordedMessageUnits               [2] INTEGER(1..bound.&maxRecordedMessageUnits) OPTIONAL,
     extensions                         [3] SEQUENCE SIZE(1..bound.&numOfExtensions) OF
                                            ExtensionField {bound}             OPTIONAL,
     ...
     }

scriptClose {PARAMETERS-BOUND : bound} OPERATION ::= {
     ARGUMENT                           ScriptCloseArg { bound}
     RETURN RESULT                      FALSE
     ERRORS                             {
                                        systemFailure |
                                        missingParameter |
                                        taskRefused |
                                        unavailableResource |
                                        unexpectedComponentSequence |
                                        unexpectedDataValue |
                                        unexpectedParameter
                                        }
     CODE                               opcode-scriptClose
     }
-- Direction: SCF → SRF, Timer :Tcl
-- This operation is issued by the SCF to deallocate the resources used to perform the
-- instance of the "User Interaction" script : the context is released.

ScriptCloseArg {PARAMETERS-BOUND : bound} ::= SEQUENCE {
     uIScriptId                         UISCRIPT.&id({SupportedUIScripts { bound}}),
     uIScriptSpecificInfo               [0] UISCRIPT.&SpecificInfo({SupportedUIScripts
                                            bound}}{@uIScriptId})                      OPTIONAL,
     extensions                         [1] SEQUENCE SIZE (1..bound.&numOfExtensions)
                                            OF ExtensionField {bound}                  OPTIONAL,
     callSegmentID                      [2] CallSegmentID {bound}                      OPTIONAL,
     ...
     }
```

```
scriptEvent {PARAMETERS-BOUND : bound} OPERATION ::= {
        ARGUMENT                    ScriptEventArg { bound}
        RETURN RESULT               FALSE
        ALWAYS RESPONDS             FALSE
        CODE                        opcode-scriptEvent
        }
-- Direction: SRF → SCF, Timer :Tre
-- This operation is issued by the SRF to return information to the SCF on the results of the
-- execution of the instance of User Interaction script.

ScriptEventArg {PARAMETERS-BOUND : bound} ::= SEQUENCE {
        uIScriptId                  UISCRIPT.&id({SupportedUIScripts { bound}}),
        uIScriptResult              [0] UISCRIPT.&Result({SupportedUIScripts {bound}}{@uIScriptId})
                                                                            OPTIONAL,
        extensions                  [1] SEQUENCE SIZE (1..bound.&numOfExtensions) OF
                                        ExtensionField {bound}              OPTIONAL,
        callSegmentID               [2] CallSegmentID {bound}               OPTIONAL,
        lastEventIndicator          [3] BOOLEAN                             DEFAULT  FALSE,
        ...
        }

scriptInformation {PARAMETERS-BOUND : bound} OPERATION ::={
        ARGUMENT                    ScriptInformationArg { bound}
        RETURN RESULT               FALSE
        ERRORS                      {
                                    systemFailure |
                                    missingParameter |
                                    taskRefused |
                                    unavailableResource |
                                    unexpectedComponentSequence |
                                    unexpectedDataValue |
                                    unexpectedParameter
                                    }
        CODE                        opcode-scriptInformation
        }
-- Direction: SCF → SRF, Timer :Tinf

ScriptInformationArg {PARAMETERS-BOUND : bound }::= SEQUENCE {
        uIScriptId              UISCRIPT.&id({SupportedUIScripts { bound}}),
        uIScriptSpecificInfo    [0] UISCRIPT.&SpecificInfo({SupportedUIScripts { bound}}{@uIScriptId})
                                                                            OPTIONAL,
        extensions              [1] SEQUENCE SIZE(0..bound.&numOfExtensions)        OF
                                    ExtensionField {bound}                  OPTIONAL,
        callSegmentID           [2] CallSegmentID {bound}                   OPTIONAL,
        ...
        }

scriptRun {PARAMETERS-BOUND : bound} OPERATION ::={
        ARGUMENT                ScriptRunArg { bound}
        RETURN RESULT           FALSE
        ERRORS                  {
                                systemFailure |
                                missingParameter |
                                taskRefused |
                                unavailableResource |
                                unexpectedComponentSequence |
                                unexpectedDataValue |
                                unexpectedParameter
                                }
        CODE                    opcode-scriptRun
        }
-- Direction: SCF → SRF, Timer: Tru
-- This operation is issued by the SCF to allocate the necessary resources to perform the
-- instance of the "User Interaction" script and then to activate this "User Interaction" script
-- instance. A context is partially defined for it if necessary.

ScriptRunArg {PARAMETERS-BOUND : bound} ::= SEQUENCE {
        uIScriptId                  UIScRIPT.&id({SupportedUIScripts { bound}}),
        uIScriptSpecificInfo        [0] UISCRIPT.&SpecificInfo({SupportedUIScripts { bound}}{@uIScriptId})
                                                                            OPTIONAL,
        extensions                  [1] SEQUENCE SIZE (1..bound.&numOfExtensions) OF
                                        ExtensionField {bound}              OPTIONAL,
        disconnectFromIPForbidden   [2] BOOLEAN                             DEFAULT TRUE,
        callSegmentID               [3] CallSegmentID {bound}               OPTIONAL,
        ...
        }

specializedResourceReport OPERATION ::= {
        ARGUMENT                    SpecializedResourceReportArg
        RETURN RESULT               FALSE
        ALWAYS RESPONDS             FALSE
        CODE                        opcode-specializedResourceReport
        }
-- Direction: SRF → SCF, Timer: Tsrr
-- This operation is used as  the response to a PlayAnnouncement operation when the announcement completed
-- report indication is set.

SpecializedResourceReportArg ::=   NULL

END
```

Table 6-1 below lists all operation timers and the value range for each timer. The definitive value for each operation timer may be network-specific and has to be defined by the network operator.

NOTE – The following value ranges do apply for operation specific timers in INAP:

short:	1-10 seconds.
medium:	1-60 seconds.
long:	1 second-30 minutes.
ffs:	For Further Study.

Table 6-1/Q.1228 – Operation timers and their value range

Operation name	Timer	Value range
ScriptClose	T_{cl}	Short
ScriptInformation	T_{inf}	Short
PlayAnnouncement	T_{pa}	Long
PromptAndCollectUserInformation	T_{pc}	Long
PromptAndReceiveMessage	T_{prm}	Long
ScriptEvent	T_{re}	Short
ScriptRun	T_{ru}	Long
SpecializedResourceReport	T_{srr}	Short

6.2 SRF/SCF contracts, packages and Application Contexts

6.2.1 Protocol overview

The **srf-scfContract** expresses the form of the service in which the SRF, a ROS-object of class **srf-scf**, initiates the contract. A ROS-object of class **scf-srf** responds in this contract.

The **srf-scfContract** is composed of a connection package, **emptyConnectionPackage** and operation packages: **specializedResourceControlPackage, srf-scfCancelPackage, srf-scfActivationOfAssistPackage scriptControlPackage,** and **messageControlPackage**. The connection package, **emptyConnectionPackage**, is defined as an information object of class CONNECTION-PACKAGE in 4.5.

When an SCF and an SRF are located in different IN physical entities, these association contracts shall be realized as an SS7 application layer protocol. The definition of this protocol in terms of an SS7 application context is provided in 6.2.2.

The operation package **specializedResourceControlPackage** is defined as an information object of class OPERATION-PACKAGE. The operations of this package is defined in 6.1.

The operation package **srf-scfrCancelPackage** is defined as information object of class OPERATION-PACKAGE. The operation of this package is defined in 5.1.

The operation package **scriptControlPackage** is defined as information object of class OPERATION-PACKAGE. The operations of this package are defined in 6.1.

The operation package **messageControlPackage** is defined as an information object of class OPERATION-PACKAGE. The operations of this package are defined in 6.1

```
messageControlPackage OPERATION-PACKAGE ::= {
        CONSUMER INVOKES  {promptAndReceiveMessage}
        ID                id-package-messageControl
        }
```

Abstract Syntax

This version of the INAP requires the support of two types of abstract syntaxes:

a) the abstract syntax of TC dialogue control protocol data units, **dialogue-abstract-syntax,** which is needed to establish the dialogue between FEs and specified in Recommendation Q.773;

b) the abstract syntax for conveying the protocol data units for invoking the operations involved in the operation packages specified as above and reporting their outcome.

The ASN.1 type from which the values of the last abstract syntax are derived is specified using the parameterized types **TCMessage{}** defined in Recommendation Q.773.

All these abstract syntaxes shall (as a minimum) be encoded according to the Basic ASN.1 encoding rules with the restrictions listed in Recommendation Q.773.

The SRF-SCF INAP ASEs that realize the operation packages specified as above and the empty connection package specified in 4.5 share a single abstract syntax, srf-scf-abstract-syntax. This is specified as an information object of the class ABSTRACT-SYNTAX.

SCF-SRF Application Contexts

The srf-scfContract is realized by an application contexts, srf-scf-ac. These application contexts are specified as information objects of the class APPLICATION-CONTEXT.

6.2.2 SRF/SCF ASN.1 modules

```
IN-CS2-SCF-SRF-pkgs-contracts-acs {itu-t recommendation q 1228 modules(0) in-cs2-scf-srf-pkgs-contracts-
acs(8) version1(0)}

DEFINITIONS ::=

BEGIN

-- This module describes the operation-packages, contracts and application-contexts used
--  over the SCF-SRF interface.

IMPORTS

        PARAMETERS-BOUND,
        networkSpecificBoundSet,
        emptyConnectionPackage

FROM IN-CS2-classes classes

        ROS-OBJECT-CLASS, CONTRACT, OPERATION-PACKAGE, OPERATION

FROM Remote-Operations-Information-Objects ros-InformationObjects

        TCMessage {}
                FROM TCAPMessages tc-Messages

        APPLICATION-CONTEXT, dialogue-abstract-syntax
                FROM TC-Notation-Extensions tc-NotationExtensions

        playAnnouncement {},
        promptAndReceiveMessage {},
        promptAndCollectUserInformation {},
        scriptClose {},
        scriptEvent {},
        scriptInformation {},
        scriptRun {},
        specializedResourceReport
FROM IN-CS2-SCF-SRF-ops-args scf-srf-Operations

        cancel {},
        assistRequestInstructions {}
FROM IN-CS2-SSF-SCF-ops-args ssf-scf-Operations

        srf-scfActivationOfAssistPackage {}

FROM IN-CS2-SSF-SCF-pkgs-contracts-acs ssf-scf-Protocol

        id-package-specializedResourceControl,
        id-ac-srf-scf,
        id-contract-srf-scf,
        id-package-srf-scfCancel,
        id-package-scriptControl,
        id-package-messageControl,
        id-as-basic-srf-scf,
        classes, ros-InformationObjects, tc-Messages, tc-NotationExtensions,
        scf-srf-Operations, ssf-scf-Operations, ssf-scf-Protocol
```

```
FROM IN-CS2-object-identifiers {ccitt recommendation q 1228 modules(0) in-cs2-object-identifiers (17)
version1(0)}
;

-- Application Contexts --

srf-scf-ac APPLICATION-CONTEXT ::= {
        CONTRACT                        srf-scf-contract
        DIALOGUE MODE                   structured
        TERMINATION                     basic
        ABSTRACT SYNTAXES               {dialogue-abstract-syntax |
                                        srf-scf-abstract-syntax }
        APPLICATION CONTEXT NAME        id-ac-srf-scf }

-- Contracts --

srf-scf-contract CONTRACT ::= {
        CONNECTION      emptyConnectionPackage
        INITIATOR CONSUMER OF   {srf-scfActivationOfAssistPackage {networkSpecificBoundSet}  }
        RESPONDER CONSUMER OF {specializedResourceControlPackage {networkSpecificBoundSet}|
                                srf-scfCancelPackage {networkSpecificBoundSet}|
                                scriptControlPackage {networkSpecificBoundSet}|
                                messageControlPackage {networkSpecificBoundSet}}
        ID                      id-contract-srf-scf }

-- specializedResourceControl package --

specializedResourceControlPackage  {PARAMETERS-BOUND : bound}  OPERATION-PACKAGE ::= {
        CONSUMER INVOKES        {playAnnouncement {bound} |
                                promptAndCollectUserInformation {bound}
                                }
        SUPPLIER INVOKES        {specializedResourceReport}
        ID                      id-package-specializedResourceControl}

-- srf-scfCancel package --

srf-scfCancelPackage  {PARAMETERS-BOUND : bound}   OPERATION-PACKAGE ::= {
        CONSUMER INVOKES        {cancel {bound}}
        ID                      id-package-srf-scfCancel}

-- scriptControl package --

scriptControlPackage  {PARAMETERS-BOUND : bound}  OPERATION-PACKAGE ::= {
        CONSUMER INVOKES        { scriptClose {bound}| scriptRun {bound} |
                                scriptInformation {bound}}
        SUPPLIER INVOKES        { scriptEvent {bound}}
        ID                      id-package-scriptControl}

-- messageControl package

messageControlPackage  {PARAMETERS-BOUND : bound} OPERATION-PACKAGE ::= {
        CONSUMER INVOKES        {promptAndReceiveMessage {bound}}
        ID                      id-package-messageControl
        }

-- Abstract Syntaxes --

srf-scf-abstract-syntax ABSTRACT-SYNTAX ::= {
        BASIC-SRF-SCF-PDUs
        IDENTIFIED BY           id-as-basic-srf-scf}
```

```
BASIC-SRF-SCF-PDUs ::= TCMessage {{SRF-SCF-Invokable},{SRF-SCF-Returnable} }

SRF-SCF-Invokable OPERATION ::= {
            assistRequestInstructions  {networkSpecificBoundSet}|
            cancel  {networkSpecificBoundSet}|
            playAnnouncement  {networkSpecificBoundSet}|
            promptAndCollectUserInformation  {networkSpecificBoundSet}|
            scriptClose  {networkSpecificBoundSet}|
            scriptEvent  {networkSpecificBoundSet}|
            scriptInformation  {networkSpecificBoundSet}|
            scriptRun  {networkSpecificBoundSet}|
            specializedResourceReport |
            promptAndReceiveMessage  {networkSpecificBoundSet}
            }
SRF-SCF-Returnable OPERATION ::= {
            assistRequestInstructions  {networkSpecificBoundSet}|
            cancel  {networkSpecificBoundSet}|
            playAnnouncement  {networkSpecificBoundSet}|
            promptAndCollectUserInformation  {networkSpecificBoundSet}|
            scriptClose  {networkSpecificBoundSet}|
            scriptInformation  {networkSpecificBoundSet}|
            scriptRun  {networkSpecificBoundSet}|
            promptAndReceiveMessage  {networkSpecificBoundSet}
            }

END
```

7 SCF-SDF interface

7.1 Introduction to the reuse of X.500 for SDF interfaces

7.1.1 Alignment between the X.500 concepts and the IN

The X.500-series Recommendations are used to specify the SCF-SDF interface and the contents of the SDF. Most of the concepts of the X.500 series are directly used in the IN environment; however, some alignments need to be done at the terminology level to ensure that the concepts introduced in the Directory are correctly understood. The purpose of this clause is to provide this alignment. It therefore only concentrates on the terms that are ambiguous in the IN environment.

When looking at the structure of the SCF, the Service Data Management is the part of the SCF responsible for the interactions with the SDF. It can be mapped onto the concept of Directory User Agent (DUA). When an SCF on behalf of a user wants to setup an association with an SDF, an instance of a DUA is created in the SLPI. It is killed when the association is ended.

The SDF is the entity responsible for answering the database requests. This functional entity can be mapped onto the Directory System Agent (DSA). When an association is setup between an SCF and an SDF, an instance of a DSA is created for the length of the association.

The Directory is a collection of DSAs/SDFs. This set can be used for a specific service or for a variety of services. The notion of Directory is equivalent to the concept of database systems in IN.

The Directory can also be seen as a repository of data. IN services provide various kinds of data access to users. The information is organised into entries. An entry is a collection of information that can be identified (or named). When it represents an object (i.e. contains primary information about an object), it is called an object entry.

Objects are anything which are identifiable (can be named) and which are of interest to hold information on in the database. A typical example of an object is a user. Objects can be described by several entries. Each individual information that is used to describe an object is an attribute. They are associated to entries.

In the IN environment, the service provider is responsible for the management and the administration of the data contained in a DSA. Therefore the service provider plays the administrator role. He is the administrative authority in X.500 terminology. The service provider enforces the security procedure (authentication and access control).

7.1.2 Use of a limited subset of X.500

The primary purpose of the X.500-series Recommendations is to provide a directory service and not the description of the SCF-SDF interface as Study Group (SG) 11 wants to use them. The X.500 functionalities cover more than the functionalities needed for IN CS-2. This subclause tries to indicate which aspects of the Directory Abstract Service should be considered and supported by implementors within the scope of CS-2. It also mentions the attitude to adopt when a non-supported parameter is received. Profiling is used as a means to present the status of the different parameters.

It is important to mention that the number of parameters carried in a message should be minimised, because each of them is associated to a load in the signalling traffic and to some processing time. This is the reason why the parameters are removed unless they are absolutely necessary when they are sent. On reception, removed parameters should not be treated but should be understood by the receiving entity. This allows the extensions of the profile in the future according to its actual description in the third edition of the Directory.

For convenience and clarity, this profile is defined using ASN.1 subtyping facilities; however, these definitions do not form a protocol specification. This simply indicates which parameters an implementation should not send. It does not change the behaviour of the receiving entity which shall still be capable of decoding values which conform to the original definition of the Directory Abstract Service. Nevertheless elements that are excluded by subtyping should be ignored.

7.1.3 Working assumptions

Several assumptions were used to design the Directory Abstract Service profile for IN CS-2. References to the assumptions used are made in 7.3 They are as follows:

Assumption 1: The version of the Directory Abstract Service used for CS-2 is the third edition. The parameters only used for the 1988 version shall be ignored. Functionalities that might be needed in future Capabilities Sets should be at least considered if not supported.

Assumption 2: The alias entries in IN are just a means to provide an alternative name for an object and therefore should be dereferenced when needed.

Assumption 3: An SCF-SDF operation cannot be abandoned. If an operation takes too much time, its timer expires and there is no need to abandon it.

7.2 The SDF Information Model

Recommendation X.501 provides a generic information model that is needed to support the service provided by the Directory. In the context of IN, the generic information model should conform to clauses 1 to 7 of Recommendation X.501. However, certain aspects of Recommendation X.501 need not be supported. This includes the DIT content rules whose use is a local matter.

Some other points are outside the scope of this Recommendation. This concerns the items associated with capabilities not covered by IN CS-2. Therefore the following parts of Recommendation X.501 are not applicable:

– paragraphs f), h) and i) in 16.2.3/X.501;

– paragraph a) in 16.2.4/X.501. The compare operation is not used, the search operation is used instead. Therefore the FilterMatch permission replaces the Compare permission.

7.2.1 Information framework

The IN defines a number of extensions to the X.501 information framework in order to meet IN service requirements. Only the enhancements are defined in these clauses. Unless stated, the definition of other elements are the same as for X.501 version 3.

7.2.1.1 METHOD

Each method represents a sequence of DAP operations which are performed under the control of the DSA. The DUA is responsible for providing all necessary information in order for the DSA to complete the method. The DSA is responsible for collecting all information to be returned to the DUA.

For documentation purposes, it is suggested to add a description field to the class definition.

The **&InputAttributes** field identifies the attributes which may be submitted as input to the method execution.

The **&OutputAttributes** field identifies the attributes which may be returned as output of the method execution.

The **&SpecificInput** field provides that syntax of additional information which may be used as input to the method execution.

The **&SpecificOutput** field provides that syntax of additional information which may be used as output to the method execution.

The **&id** field uniquely identifies the method.

```
METHOD ::= CLASS {
        &InputAttributes                ATTRIBUTE OPTIONAL,
        &SpecificInput                  OPTIONAL,
        &OutputAttributes               ATTRIBUTE OPTIONAL,
        &SpecificOutput                 OPTIONAL,
        &description                    PrintableString OPTIONAL,
        &id                             OBJECT IDENTIFIER UNIQUE}
WITH SYNTAX {
        [ INPUT ATTRIBUTES              &InputAttributes ]
        [SPECIFIC-INPUT                 &SpecificInput ]
        [OUTPUT ATTRIBUTES              &OutputAttributes ]
        [SPECIFIC-OUTPUT                &SpecificOutput ]
        [BEHAVIOUR                      &description]
        ID                              &id}
```

7.2.1.2 DIT METHOD Use

7.2.1.2.1 Overview

A DIT METHOD Use is a specification provided by the subschema administrative authority to specify the METHOD types that may be used on entries of a particular object-class.

A DIT METHOD Use definition includes:

a) an indication of the object-class type to which it applies;

b) an indication of the METHOD types that shall be associated with the object-class whenever entries of that object-class are stored.

The DIT Method Use definition for a particular object-class also applies to any subclass which may be subsequently defined.

7.2.1.2.2 DIT METHOD Use specification

The abstract syntax of a DIT METHOD Use is expressed by the following ASN.1 type:

```
DITMethodUse      ::=     SEQUENCE {
        objectClass                 OBJECT-CLASS.&id,
        methods           [1]       SET OF METHOD.&id }
```

The correspondence between the parts of the definition, as listed in 7.2.1.2.1, and the various components of the ASN.1 type defined above, is as follows:

a) the **objectClass** component identifies the object-class to which the DIT METHOD Use applies;

b) the **methods** component specifies types that shall be associated with the object-class whenever entries of that object-class are stored.

The **DITMethodUse** definition for a particular object-class also applies to any subclass which may be subsequently defined.

The METHOD-USE-RULE information object class is provided to facilitate the documentation of the DIT METHOD Use rules:

```
METHOD-USE-RULE ::= CLASS {
        &objectClassType            OBJECT-CLASS.&id        UNIQUE,
        &Mandatory                  METHOD }
WITH SYNTAX {
        OBJECT-CLASS TYPE           &objectClassType
        METHODS                     &Mandatory }
```

The METHOD-USE-RULE definition for a particular object-class also applies to any subclass which may be subsequently defined.

7.2.2 Basic Access Control

The following enhancements to the third edition X.500 specification of Access Control Information (ACI) are required to support IN CS-2 requirements on the SCF/SDF interface. Only the enhancements are described here. The remaining elements apply as described in the third edition X.500-series of Recommendations.

7.2.2.1 ProtectedItems

The definitions of **ProtectedItems** is extended as follows:

```
ProtectedItems      ::=     SEQUENCE {
        entry                               [0]     NULL OPTIONAL,
        allUserAttributeTypes               [1]     NULL OPTIONAL,
        attributeType                       [2]     SET OF AttributeType OPTIONAL,
        allAttributeValues                  [3]     SET OF AttributeType OPTIONAL,
        allUserAttributeTypesAndValues      [4]     NULL OPTIONAL,
        attributeValue                      [5]     SET OF AttributeTypeAndValue OPTIONAL,
        selfValue                           [6]     SET OF AttributeType OPTIONAL,
        rangeOfValues                       [7]     Filter OPTIONAL,
        maxValueCount                       [8]     SET OF MaxValueCount OPTIONAL,
        maxImmSub                           [9]     INTEGER OPTIONAL,
        restrictedBy                        [10]    SET OF RestrictedValue OPTIONAL,
        contexts                            [11]    SET OF ContextAssertion OPTIONAL,
        entryMethods                        [30]    SET OF MethodIDs OPTIONAL}
```

entryMethods identifies the specified Methods for which the level of protection is to be applied.

```
MethodIDs ::=       METHOD.&id
```

7.2.2.2 GrantsAndDenials

The definition of **GrantsAndDenials** is extended as follows:

```
GrantsAndDenials                          ::=    BIT STRING {
        -- permissions that may be used in conjunction
        -- with any component of ProtectedItems
        grantAdd                      (0),
        denyAdd                       (1),
        grantDiscloseOnError          (2),
        denyDiscloseOnError           (3),
        grantRead                     (4),
        denyRead                      (5),
        grantRemove                   (6),
        denyRemove                    (7),
        -- permissions that may be used only in conjunction
        -- with the entry component
        grantBrowse                   (8),
        denyBrowse                    (9),
        grantExport                   (10),
        denyExport                    (11),
        grantImport                   (12),
        denyImport                    (13),
        grantModify                   (14),
        denyModify                    (15),
        grantRename                   (16),
        denyRename                    (17),
        grantReturnDN                 (18),
        denyReturnDN                  (19),
        -- permissions that may be used in conjunction
        -- with any component, except entry, of ProtectedItems
        grantCompare                  (20),
        denyCompare                   (21),
        grantFilterMatch              (22),
        denyFilterMatch               (23),
        -- permissions that may be used in conjunction
        -- with entryMethod component of ProtectedItems
        grantExecuteMethod            (30),
        denyExecuteMethod             (31) }
```

grantExecuteMethod means that the user can perform the specific Methods for the Entry.

NOTE – It is a matter for network operators as to whether the grantExecuteMethod permission bypasses the normal access control mechanisms for Entries and Attributes.

denyExecuteMethod means that the user cannot perform the specific Methods for the Entry

7.2.3 Attribute contexts

7.2.3.1 Basic Service context

This Basic Service context associates an attribute value with a basic service for which the attribute value is semantically valid. For example, the Basic Service context will be associated with an ISDN address to indicate the type of basic service that could be used with it. In the UPT case, this context allows the definition of registration addresses for different basic services.

```
basicServiceContext          CONTEXT ::= {
        WITH SYNTAX          BasicService
        ID                   id-avc-basicService}

BasicService ::= INTEGER {
        telephony (1),
        faxGroup2-3 (2),
        faxGroup4 (3),
        teletexBasicAndMixed (4),
        teletexBasicAndProcessable (5),
        teletexBasic (6),
        syntaxBasedVideotex (7),
        internationalVideotex (8),
        telex (9),
        messageHandlingSystems (10),
        osiApplication (11),
        audioVisual (12)}
```

A presented value is considered to match a stored value if the context value (i.e. a basic service value) in the presented value is identical to that in the stored value.

7.2.3.2 Line Identity context

The line identity context associates an attribute value with the identity of a line for which the attribute value is semantically valid. For example, this Line Identity context will be associated with a routing number to provide calling-line dependent routing.

```
lineIdentityContext CONTEXT ::= {
        WITH SYNTAX     IsdnAddress
        ID              id-avc-lineIdentity}

IsdnAddress ::= AddressString {ub-international-isdn-number}
```

7.2.3.3 Assignment context

The assignment context associates an attribute value with a Distinguished name (e.g. customer's number or customer's name) for which the attribute value is assigned. For example, assuming that a set of available resources is modelled as a multivalued attribute and the customer has been designated by a distinguished name, this Assignment context will be associated with the used resource to provide the state of the resource (reserved) and the name of the current customer using it.

```
assignmentContext  CONTEXT ::= {
        WITH SYNTAX     DistinguishedName
        ID              id-avc-assignment }
```

7.2.4 Attribute definitions

7.2.4.1 DIT Method Use operational attribute

The **methodUse** operational attribute is used to indicate the methods which shall be used with an object-class and all of its subclasses:

```
methodUse ATTRIBUTE ::= {
        WITH SYNTAX                 MethodUseDescription
        EQUALITY MATCHING RULE      objectIdentifierFirstComponentMatch
        USAGE                       directoryOperation
        ID                          id-soa-methodRuleUse }
```

```
MethodUseDescription     ::=    SEQUENCE {
        identifier                OBJECT-CLASS.&id,
        name                      SET OF DirectoryString { ub-schema } OPTIONAL,
        description               DirectoryString { ub-schema } OPTIONAL,
        obsolete                  BOOLEAN DEFAULT FALSE,
        information        [0]    SET OF METHOD.&id }
```

The **identifier** component of a value of the **methodUse** operational attribute is the object identifier of the object-class type to which it applies. The value **id-oa-allObject-classTypes** indicates that it applies to all object-class types.

The **information** component of a value identifies the method types associated with the object-class identified by **identifier.**

Every entry in the DIT is governed by at most one methodUse operational attribute. In addition, the entry is also governed by all the methodUse operation attribute defined for the superclasses of its structural object class.

NOTE – This means that before processing an execute operation, the SDF shall check the methodUse attributes associated with the structural object classes which belong to the inheritance chain of the entry's structural object class.

As a methodRule attribute is associated with a structural object class, it follows that all of the entries on the same structural object class will have the same Method Use Rule regardless of the DIT structure rule governing their location in the DIT and of the DIT content rule governing their contents.

7.3 The SCF-SDF Interface Protocol

7.3.1 Information types and common procedures

7.3.1.1 CommonArguments

```
IN-CommonArguments ::= CommonArguments (
        WITH COMPONENTS {
                ...,
                serviceControls           (IN-ServiceControls),
                aliasedRDNs               ABSENT })
```

The **serviceControls** component is described in 7.3.1.2.

The **aliasedRDNs** component is present in the third edition only for compatibility reasons. It should always be omitted in the third edition implementations of the Directory (assumption 1).

7.3.1.2 ServiceControls

```
IN-ServiceControls ::= ServiceControls
        (WITH COMPONENTS {
                ...,
                timeLimit            ABSENT,
                sizeLimit            ABSENT,
                scopeOfReferral      ABSENT,
                attributeSizeLimit   ABSENT})
```

The **timeLimit** component indicates the maximum elapsed time to fulfil a request. It is redundant with the operation timers of TCAP and therefore is not needed.

The **sizeLimit** and the **attributeSizeLimit** set some size limits on the results either in terms of objects or in terms of attributes. This is useful when requests are expected to be general (the requestor does not know the structure of the DSA), but in the case of IN, this type of limitation does not seem applicable.

7.3.1.3 Entry Information Selection

```
IN-EntryInformationSelection ::= EntryInformationSelection
        (WITH COMPONENTS {
                ...,
                infoTypes               (attributeTypesAndValues) })
```

The **attributes** component specifies the attributes that should be returned in a retrieval service. The **allUserAttributes** option is kept even though it is advised to service specifiers to avoid its use which generates more traffic than needed. Instead the **select** option which precisely names the requested attributes should be used.

The **infoTypes** component specifies whether the attribute types and values should be returned or only the types. IN services are mainly interested in the attribute values that are relevant to the processing of the service. This component should be absent given its default value.

7.3.1.4 EntryInformation

```
IN-EntryInformation ::= EntryInformation
        (WITH COMPONENTS {
                ...,
                fromEntry               (TRUE),
                information             (WITH COMPONENTS {
                        ...,
                        attributeType                   ABSENT}) OPTIONAL})
```

The **fromEntry** component indicates if a copy or the entry itself is returned. Since IN CS-1 does not use copy mechanisms (assumption 3), only the default value of this component should be used.

The **information** parameter contains the relevant information which is returned. Given the choice made for the **infoTypes** component (see 7.3.1.3), only the **attribute** option should be used.

7.3.1.5 SPKM Token profile

The X.511 third edition **Bind** operation allows the use of SPKM security procedures to be specified. This subclause profiles the SPKM token which can be used for IN CS-2 operations.

```
IN-Context-Data ::= Context-Data
        (WITH COMPONENTS {
                ...,
                channelId               ABSENT,
                seq-number              ABSENT})
```

Context-Data specifies options and the confidentiality, integrity, and one-way authentication function algorithm identifiers.

The channel identifier (**channelId**) is not required for IN CS-2, as only one channel is used.

The **seq-number** parameter indicates the sequence number of the token. This is not required for IN CS-2 because the sequencing of messages is assumed via the lower protocol layers.

```
IN-Mic-Header ::= Mic-Header
        (WITH COMPONENTS {
                ...,
                snd-seq         ABSENT})
```

The **snd-seq** parameter indicates the sequence number of the token. This is not required for IN CS-2 because the sequencing of messages is assumed via the lower protocol layers.

Mic-Header contains the token identifier, the context identifier, and the integrity algorithm identifier.

```
IN-Wrap-Header ::= Wrap-Header
        (WITH COMPONENTS {
                ...,
                snd-seq         ABSENT})
```

Wrap-Header specifies header information containing the token identifier, the context identifier, the integrity algorithm identifier, and the confidentiality algorithm identifier.

```
IN-Del-Header ::= Del-Header
        (WITH COMPONENTS {
                ...,
                snd-seq         ABSENT})
```

Del-Header contains the token identifier, the context identifier, and the integrity algorithm identifier.

7.3.2 Operations

7.3.2.1 Bind operation

```
in-DirectoryBind  OPERATION ::= {
        ARGUMENT        DirectoryBindArgument
        RESULT          DirectoryBindResult
        ERRORS          {in-DirectoryBindError}
        CODE            in-opcode-in-bind}
```

7.3.2.2 Search operation

```
in-Search  OPERATION ::= {
        ARGUMENT        IN-SearchArgument
        RESULT          IN-SearchResult
        ERRORS          {nameError | in-ServiceError | securityError | attributeError | referral}
        CODE            id-opcode-in-search}
```

The **search** operation is used to search a portion of the DIT for entries of interest.

```
IN-SearchArgument ::= SearchArgument(
        WITH COMPONENTS {
                ...,
                searchAliases           (TRUE),
                selection               (IN-EntryInformationSelection),
                pagedResults            ABSENT,
                extendedFilter          ABSENT,
                COMPONENTS OF           IN-CommonArguments } )
```

The **filter** parameter is used to eliminate entries from the search space. However, the **extendedFilter** parameter was added in the 1993 version of the Directory for compatibility reasons and should therefore not be sent. Only the **filter** parameter should be sent.

The **searchAliases** parameter indicates whether the aliases encountered in the search space (except the base object) should be considered. Since in IN aliases are always dereferenced when searching, this parameter should be used only with its default value.

The **selection** parameter indicates what information from the entries, e.g. types and values, is requested (see 7.3.1.3).

The **pagedResults** parameter is used to request a page-by-page result. The **pagedResults** parameter is used to present the results of a search operation in a page format. This type of information is not needed in IN CS-2 since the SCF treats the results.

The **abandoned** error is not supported.

```
IN-SearchResult ::= SearchResult
        (WITH COMPONENTS {
                ...,
                searchInfo              (WITH COMPONENTS {
                        ...,
                        entries         (WITH COMPONENT (IN-EntryInformation)),
                        partialOutcomeQualifier   (PartialOutcomeQualifier
                                        (WITH COMPONENTS {
                                        ...,
                                        queryReference          ABSENT}))OPTIONAL})})
```

The **entries** parameter contains the entries that satisfy the filter.

The **partialOutcomeQualifier** parameter is present when the search operation was not fully completed. It contains information on the reasons why the search operation was not finished and on where the operation was stopped.

The **queryReference** parameter is used when paged results were requested and therefore is not needed.

7.3.2.3 AddEntry operation

```
in-AddEntry  OPERATION ::= {
        ARGUMENT        AddEntryArgument
        RESULT          AddEntryResult
        ERRORS          {nameError | in-ServiceError | securityError | attributeError | updateError |
                         referral}
        CODE            id-opcode-in-addEntry}
```

7.3.2.4 RemoveEntry operation

```
in-RemoveEntry  OPERATION ::= {
        ARGUMENT        RemoveEntryArgument
        RESULT          RemoveEntryResult
        ERRORS          {nameError | in-ServiceError | securityError | updateError | referral}
        CODE            id-opcode-in-removeEntry}
```

7.3.2.5 ModifyEntry operation

```
in-ModifyEntry  OPERATION ::= {
        ARGUMENT        IN-ModifyEntryArgument
        RESULT          IN-ModifyEntryResult
        ERRORS          {nameError | in-ServiceError | securityError | attributeError | updateError |
                         referral}
        CODE            id-opcode-in-modifyEntry}

IN-ModifyEntryArgument ::= ModifyEntryArgument
        (WITH COMPONENTS {
                ...,
                selection       (IN-EntryInformationSelection)})
```

The **selection** parameter specifies some attributes and values to be returned (See 7.3.1.3).

```
IN-ModifyEntryResult ::= ModifyEntryResult
        (WITH COMPONENTS {
                ...,
                null           ,
                information  Information
                        (WITH COMPONENTS {
                                ...,
                                entry  (IN-EntryInformation)})}
```

If no information was to be retrieved with the modifyEntry operation, the **null** result is returned. Otherwise the information is to be returned in the **entry** component of the **information** result. For IN CS-2, this component is specified in 7.3.1.4.

7.3.2.6 Execute operation

The execute operation performs a sequence of execution steps, according to a predefined method, using input information and returns result information. Each step is either a DAP operation (that could be an execute operation), the execution of an algorithm or a decision test.

The parameters of the individual DAP operations are taken from the input parameters and the results of previous operations and/or the output of the algorithms associated with the method. The output parameters are taken from the results of the individual operations. The execute operation is considered to be an atomic operation.

```
execute OPERATION ::= {
        ARGUMENT        ExecuteArgument
        RESULT          ExecuteResult
        ERRORS          { attributeError | nameError |
                          serviceError | referral |
                         securityError |
                          updateError | executionError }
        CODE            id-opcode-execute }
ExecuteArgument ::=  OPTIONALLY-PROTECTED {
        SET {
                object              [0] Name,
                method-id           [1] METHOD.&id({SupportedMethods}),
                input-assertions    [2] SEQUENCE OF SEQUENCE {
                                    type METHOD.&InputAttributes.&id({SupportedMethods}{@method-id}),
                                    values SET OF
METHOD.&InputAttributes.&id({SupportedMethods}{@method-id}) OPTIONAL,
                                    valuesWithContext [0] SET OF SEQUENCE {
                                            value  [0]
METHOD.&InputAttributes.&id({SupportedMethods}{@method-id})
OPTIONAL,

                                            contextList   [1] SET OF Context
                                            } OPTIONAL

                                    } OPTIONAL,
                specific-input       [3] METHOD.&SpecificInput({SupportedMethods}{@method-id})
        OPTIONAL,
                COMPONENTS OF        CommonArguments },
        DIRQOP.&dapModifyEntryArg-QOP{@qop} }
```

The **object** field identifies the entry in the DIT from/on which the method is to be executed.

The **execute-id** field identifies the method which is to be executed within the SDF.

The **input-assertions** field provides a set of attribute values which are used as an input to the method execution.

The **specific-input** field identifies the additional information which is required by the SDF in order to perform the method.

```
ExecuteResult ::= OPTIONALLY-PROTECTED {
        SET {
                method-id               [1] METHOD.&id({SupportedMethods}),
                output-assertions       [2] SEQUENCE OF SEQUENCE {
                                        type METHOD.&OutputAttributes.&id({SupportedMethods}{@method-id}),
                                        values SET OF
METHOD.&OutputAttributes.&Type({SupportedMethods}{@method-id,@.type}) OPTIONAL,
                                        valuesWithContext [0] SET OF SEQUENCE {
                                                value  [0]
        METHOD.&OutputAttributes.&Type({SupportedMethods}{@method-id,@.type})     OPTIONAL,
                                                contextList    [1] SET OF Context
                                                } OPTIONAL
                                        } OPTIONAL,
                specific-output         [3] METHOD.&SpecificOutput({SupportedMethods}{@method-id})
                                                                                        OPTIONAL,
                COMPONENTS OF CommonResults },
        DIRQOP.&dapModifyEntryRes-QOP{@qop} }
```

The **specific-output** field contains information returned as a result of the method execution.

The **output-assertions** contains attributes values returned as a result of the method execution.

The **SupportedMethods** set contains all of the defined methods for the interface. Its exact contents are a matter for local determination as it will depend on the service and network provider agreements being supported.

7.3.2.7 in-directoryUnbind operation

The **in-directoryUnbind** operation replaces the X.511 **directoryUnbind** operation to provide class 4 operation behaviour for unbind procedures.

```
in-directoryUnbind  OPERATION ::= inEmptyUnbind
```

7.3.3 Errors

The precedence rule defined in the Directory should apply.

The **abandoned** and **abandonFailed** errors are not considered because they are not supported by CS-2 (assumption 1 and assumption 4)

7.3.3.1 Bind error

```
        IN-DirectoryBindError ::= DirectoryBindError
                (WITH COMPONENTS {
                        ...,
                        error  (WITH COMPONENTS {
                                                securityError (SecurityProblem (1|2|7|10)),
                                                serviceError (ServiceProblem (2))})})
```

SecurityProblem 10 indicates that the supplied SPKM token was found to be invalid.

In reception, all the possible errors should be supported to understand a Bind error.

7.3.3.2 Service error

```
in-ServiceError ERROR ::= {
        PARAMETER       IN-ServiceErrorParameter
        CODE            id-errcode-in-serviceError}

IN-ServiceErrorParameter::=ServiceErrorParameter
        (WITH COMPONENTS {
                problem         (ServiceProblem ( 1 | 2 | 3 | 4 |5 | 6 | 8 | 9 | 10 | 11 | 12))})
```

invalidQueryReference should not be sent because it is linked to the use of paged results.

7.3.3.4 execution Error

The **executionError** is returned by an Execute operation in the case of the operation not completing.

```
executionError ERROR ::= {
        PARAMETER       OPTIONALLY-PROTECTED {
                        SET {
                                problem         [0]     ExecutionProblem,
                                COMPONENTS OF CommonResults },
                        DIRQOP.&dirErrors-QOP{@dirqop} }
        CODE                    id-errcode-executionError }

ExecutionProblem  ::=  INTEGER {
        missingInputValues (1),
        executionFailure(2) }
```

The executeProblem identifies the cause of the execute operation failure:

– **missingInputValues** is returned in the input-values field contains the wrong input information for the method being executed.

– **executionFailure** is returned when the method fails to complete correctly. This is caused by the failure of one of the DAP operations contained within the method.

7.4 Protocol overview

7.4.1 Remote Operations

ITU-T Rec. X.880 | ISO/IEC 13712-1 defines several information object classes that are useful in the specification of ROS-based application protocols such as the various Directory protocols defined in this Directory Specification. A number of these classes are used in subsequent clauses. The specification techniques provided in ITU-T Rec. X.880 | ISO/IEC 13712-1 are used to define a generic protocol between objects. When realised as an SS7 application layer protocol, the concepts of ITU-T Rec. X.880 | ISO/IEC 13712-1 are mapped to SS7 concepts in Recommendation Q.775.

7.4.2 Directory ROS – Objects and Contracts

ITU-T Rec. X.519 | ISO/IEC 9594-5 defines the abstract service between a DUA and the Directory which provides an access point to support a user accessing Directory services. Subclause 7.5 defines the subset of this abstract service used in the context of Intelligent Networks.

The **dua** class of ROS-object describes a DUA, being an instance of this class, as the initiator of the contract **dapContract** or the **dapExecuteContract**. These contracts are referred to in these Directory Specifications as the Directory Abstract Service. It is specified as a ROS-based information object in this subclause.

```
dua ROS-OBJECT-CLASS ::= {
        INITIATES  {dapContract| dapExecuteContract}
        ID              id-rosObject-dua}
```

The **directory** class of ROS-object describes the provider of the Directory Abstract Service. This provider is the responder of the **dapContract/dapExecuteContract.**

```
directory ROS-OBJECT-CLASS ::= {
        RESPONDS {dapContract| dapExecuteContract}
        ID              id-rosObject-directory}
```

The Directory is further modelled as being represented to a DUA by a DSA which supports the particular access point concerned. In the context of Intelligent Networks, each DSA is potentially an access point to the Directory.

The **directory** object is manifested as a set of DSAs *(each of which resides in an SDF).* Each DSA comprising the **directory** is an instance of the **dap-dsa** class. A **dap-dsa** object assumes the role of responder in the **dapContract/dapExecuteContract**.

```
dap-dsa ROS-OBJECT-CLASS ::= {
        RESPONDS {dapContract| dapExecuteContract}
        ID              id-rosObject-dapDSA}
```

Future versions of this Recommendation will enable DSAs to interact with one another to achieve various objectives.

7.4.3 DAP Contract and Packages

The **dapContract** is defined as an information object of class CONTRACT.

```
dapContract CONTRACT ::= {
        CONNECTION                      dapConnectionPackage
        INITIATOR CONSUMER OF   {searchPackage | modifyPackage}
        ID                              id-contract-dap}
```

When a DUA and a DSA are located in different IN physical entities, this association contract shall be realised as an SS7 application layer protocol, referred to as the IN Directory Access Protocol (DAP). The definition of this protocol in terms of an SS7 application context is provided in 7.5.2.1.

The **dapContract** is composed of a connection package, **dapConnectionPackage**, and two operation packages, **searchPackage** and **modifyPackage**.

The **dapExecuteContract** is defined as an information object of class CONTRACT.

```
dapExecuteContract CONTRACT ::= {
        CONNECTION                      dapConnectionPackage
        INITIATOR CONSUMER OF   {searchPackage | modifyPackage | executePackage }
        ID                              id-contract-dapExecute}
```

The **dapExecuteContract** is composed of a connection package, **dapConnectionPackage**, and three operation packages, **searchPackage**, **modifyPackage**, and **executePackage**.

The connection package, **dapConnectionPackage**, is defined as an information object of class CONNECTION-PACKAGE. The bind operation of this connection package, **directoryBind**, is defined in ITU-T Rec. X.511 | ISO/IEC 9594-3. The unbind operation of this connection package, **in-directoryUnbind** is defined in 7.3.2.7.

```
dapConnectionPackage CONNECTION-PACKAGE ::= {
        BIND                in-DirectoryBind
        UNBIND            in-directoryUnbind
        ID                    id-package-dapConnection}
```

The operation packages, **searchPackage** and **modifyPackage**, are defined as information objects of class OPERATION-PACKAGE. The operations of these operation packages are defined in ITU-T Rec. X.511 | ISO/IEC 9594-3. ITU-T Rec. X.511 | ISO/IEC 9594-3 defines additional operations for supporting access to the Directory. Such operations are not used in the context of Intelligent Networks.

```
searchPackage OPERATION-PACKAGE ::= {
        CONSUMER INVOKES            {search}
        ID                          id-package-search}

modifyPackage OPERATION-PACKAGE ::= {
        CONSUMER INVOKES            {addEntry | removeEntry | modifyEntry}
        ID                          id-package-modify}
```

NOTE – These packages, when realised as ASEs, are used for the construction of application contexts defined in this specification. They are not intended to allow for claims of conformance to individual, or other combinations of, ASEs.

The operation package, **executePackage**, is defined as an information object of class OPERATION-PACKAGE. The operation of this operation package is defined in 7.3.2.6.

```
executePackage OPERATION-PACKAGE ::= {
        CONSUMER INVOKES  {execute}
        ID                id-package-execute}
```

Since the DUA is the initiator of the **dapContract**/**dapExecuteContract**, it assumes the role of consumer of the operation packages of the contract. This means that only the DUA can invoke operations in these contracts and their SS7 realisations.

7.5 Directory protocol abstract syntax

7.5.1 Abstract syntaxes

This version of the Directory Access Protocol requires the support of three abstract syntaxes:

a) the abstract-syntax of TC dialogue-control protocol data units, **dialogue-abstract-syntax**, which is needed to establish the dialogues between the SCFs and the SDFs and is specified in Recommendation Q.773;

b) the abstract-syntax for conveying the protocol data units for invoking **directoryBind** and **directoryUnbind** operations and reporting their outcome;

c) the abstract-syntax for conveying the protocol data units for invoking the operations involved in the operation packages specified in 7.3.3 and reporting their outcome.

The ASN.1 type from which the values of the second abstract syntax are derived is specified using the parameterized types, **Bind {}** and **Unbind {}** which are defined in Recommendation X.880.

The ASN.1 type from which the values of the last abstract syntax are derived is specified using the parameterized types **TCMessage {}** defined in Recommendation Q.773.

All these abstract syntaxes shall (as a minimum) be encoded according to the Basic ASN.1 encoding rules with the restrictions listed in Recommendation Q.773.

7.5.1.1 DAP Abstract Syntax

The Directory ASEs that realise the operation packages specified in 7.4.3, excluding the **executePackage**, share a single abstract syntax, **directoryOperationsAbstractSyntax**. This is specified as an information object of the class ABSTRACT-SYNTAX.

```
inDirectoryOperationsAbstractSyntax ABSTRACT-SYNTAX ::= {
        BasicDAP-PDUs
        IDENTIFIED BY    id-as-indirectoryOperationsAS}

BasicDAP-PDUs ::= TCMessage {{DAP-Invokable},{DAP-Returnable}}

DAP-Invokable  OPERATION ::= {search | addEntry | removeEntry | modifyEntry}

DAP-Returnable OPERATION ::= {search | addEntry | removeEntry | modifyEntry}
```

7.5.1.2 Extended DAP Abstract Syntax

The Directory ASEs that realise the operation packages specified in 7.4.3, including the **executePackage**, share a single abstract syntax, **inExtendedDirectoryOperationsAbstractSyntax**. This is specified as an information object of the class ABSTRACT-SYNTAX.

```
inExtendedDirectoryOperationsAbstractSyntax ABSTRACT-SYNTAX ::= {
        Extended-BasicDAP-PDUs
        IDENTIFIED BY    id-as-inExtendedDirectoryOperationsAS}

Extended-BasicDAP-PDUs ::= TCMessage {{Extended-DAP-Invokable},{Extended-DAP-Returnable}}

Extended-DAP-Invokable  OPERATION ::= {search | addEntry | removeEntry | modifyEntry | execute}

Extended-DAP-Returnable OPERATION ::= {search | addEntry | removeEntry | modifyEntry | execute}
```

7.5.1.3 DAP Binding Abstract Syntax

The realisation of the connection package specified in 7.4.3 uses a separate abstract syntax, **directoryBindingAbstractSyntax**. This is specified as an information object of the class ABSTRACT-SYNTAX

```
inDirectoryBindingAbstractSyntax        ABSTRACT-SYNTAX ::= {
        DAPBinding-PDUs
        IDENTIFIED BY    id-as-indirectoryBindingAS}

DAPBinding-PDUs ::= CHOICE {
        bind    Bind {directoryBind},
        unbind          Unbind {in-directoryUnbind}}
```

7.5.1.4 SESE Abstract Syntax

An additional abstract syntax, **inSESEAbstractSyntax,** is used in the **iNdirectoryAccessWith3seAC** defined in 7.5.2.1. This is specified as an information object of the class ABSTRACT-SYNTAX.

```
inSESEAbstractSyntax      ABSTRACT-SYNTAX ::= {
        SESEapdus {{spkmThreeWay},NoInvocationId}
        IDENTIFIED BY    {id-as-inSESEAS}}
```

SESEapdus is imported from Recommendation X.832 and **spkmThreeWay** is imported from Recommendation X.519.

7.5.2 Directory application contexts

7.5.2.1 Directory Access Application Context

The **dapContract** is realised as the **iNdirectoryAccessAC**. This application context is specified as an information object of the class APPLICATION-CONTEXT.

```
iNdirectoryAccessAC APPLICATION-CONTEXT ::= {
    CONTRACT                    dapContract
    DIALOGUE MODE               structured
    TERMINATION                     basic
    ABSTRACT SYNTAXES           {dialogue-abstract-syntax |
                                    inDirectoryOperationsAbstractSyntax |
                                    inDirectoryBindingAbstractSyntax}
    APPLICATION CONTEXT NAME    id-ac-indirectoryAccessAC}
```

If 3-way authentication is required then the **dapContract** is realised as the **iNdirectoryAccessWith3seAC**. This application context is specified as an information object of the class APPLICATION-CONTEXT.

```
iNdirectoryAccessWith3seAC APPLICATION-CONTEXT ::= {
    CONTRACT                    dapContract
    DIALOGUE MODE               structured
    TERMINATION                 basic
    ADDITIONAL ASE              {id-se-threewayse}
    ABSTRACT SYNTAXES           {dialogue-abstract-syntax |
                                inDirectoryOperationsAbstractSyntax |
                                inDirectoryBindingAbstractSyntax |
                                inSESEAbstractSyntax }
    APPLICATION CONTEXT NAME    id-ac-indirectoryAccessWith3seAC}
```

7.5.2.2 Extended Directory Access Application Context

The **dapExecuteContract** is realised as the **inExtendedDirectoryAccessAC**. This application context is specified as an information object of the class APPLICATION-CONTEXT.

```
inExtendedDirectoryAccessAC APPLICATION-CONTEXT ::= {
    CONTRACT                    dapExecuteContract
    DIALOGUE MODE               structured
    TERMINATION                     basic
    ABSTRACT SYNTAXES           {dialogue-abstract-syntax |
                                    inExtendedDirectoryOperationsAbstractSyntax |
                                    inDirectoryBindingAbstractSyntax}
    APPLICATION CONTEXT NAME    id-ac-inExtendedDirectoryAccessAC}
```

If 3-way authentication is required then the **dapExecuteContract** is realised as the **inExtendedDirectoryAccessAC**. This application context is specified as an information object of the class APPLICATION-CONTEXT.

```
inExtendedDirectoryAccessWith3seAC APPLICATION-CONTEXT ::= {
    CONTRACT                    dapExecuteContract
    DIALOGUE MODE               structured
    TERMINATION                 basic
    ADDITIONAL ASE              {id-se-threewayse}
    ABSTRACT SYNTAXES           {dialogue-abstract-syntax |
                                inExtendedDirectoryOperationsAbstractSyntax |
                                inExtendedDirectoryBindingAbstractSyntax |
                                inSESEAbstractSyntax }
    APPLICATION CONTEXT NAME    id-ac-inExtendedDirectoryAccessWith3seAC}
```

7.5.3 Operation codes

The operations involved in the packages defined in this Recommendation are specified in Recommendation X.519 where the assigned operation codes are imported from Recommendation X.519.

7.5.4 Error codes

The errors involved in the packages defined in this Recommendation are specified in Recommendation X.519 where the assigned error codes are imported from Recommendation X.519.

7.5.5 Versions and the rules for extensibility

The Directory may be distributed and more than two Directory Application Entities may interoperate to service a request. The Directory AEs may be implemented conforming to different editions of the Directory specification of the Directory service which may or may not be represented by different protocol version numbers. The version number is negotiated to the highest common version number between two directly binding Directory AEs.

7.5.5.1 Version negotiation

When accepting an association, i.e. binding, utilising the DAP, the version negotiated shall only affect the point-to-point aspects of the protocol exchanged between the DUA and the DSA to which it is connected. Subsequent requests or responses on the dialogue shall be constrained by the version negotiated.

NOTE – There are no point-to-point aspects of the DAP that are currently indicated by different protocol versions.

7.5.5.2 DUA side

7.5.5.2.1 Request and response processing at the DUA side

The DUA may initiate requests using the highest edition of the specification of that request it supports. If one or more elements of the request are critical, it shall indicate the extension number(s) in the critical Extensions parameter.

NOTE 1 – If the information the extension replaced in a CHOICE, ENUMERATED or INTEGER (used as ENUMERATED) type would be essential for proper operation in a DSA implemented according to an earlier edition of the specification, it is recommended that the extension be marked critical.

When processing a response, a DUA shall:

a) ignore all unknown bit name assignments within a bit string; and

b) ignore all unknown named numbers in an ENUMERATED type or INTEGER type that is being used in the enumerated style, provided the number occurs as an optional element of a SET or SEQUENCE; and

c) ignore all unknown elements in SETs, at the end of SEQUENCEs, or in CHOICEs where the CHOICE is itself an optional element of a SET or SEQUENCE;

   NOTE 2 – Implementations may as a local option ignore certain additional elements in a Directory PDU. In particular, some unknown named numbers and unknown CHOICEs in mandatory elements of SETs and SEQUENCEs can be ignored without invalidating the operation. The identification of such elements is for further study.

d) not consider the receipt of unknown attribute types and attribute values as a protocol violation; and

e) optionally report the unknown attribute types and attribute values to the user.

7.5.5.2.2 Extensibility rules for error handling at the DUA side

When processing a known error type with unknown indicated problems and parameters, a DUA shall:

a) not consider the receipt of unknown indicated problems and parameters as a protocol violation (i.e. it shall not issue a TC-U-REJECT or abort the dialogue); and

b) optionally report the additional error information to the user.

When processing an unknown error type, a DUA shall:

a) not consider the receipt of unknown error type as a protocol violation (i.e. it shall not issue a TC-U-REJECT or abort the application association); and

b) optionally report the error to the user.

7.5.5.3 Request processing at the DSA side

If any DSA performing an operation detects an element **criticalExtensions** whose semantic is unkown, it shall return an **unavailableCriticalExtension** indication as a **serviceError**.

NOTE 1 – If a **criticalExtensions** string with one or more zero values is received, this indicates either that the extensions corresponding to the values are not present or are not critical. The presence of a zero value in a **criticalExtensions** string shall not be inferred as either the presence or absence of the corresponding extension in the APDU.

Otherwise, when processing a request from a DUA, a DSA shall:

a) ignore all unknown bit name assignments within a bit string; and

b) ignore all unknown named numbers in an ENUMERATED type or INTEGER type that is being used in the enumerated style, provided the number occurs as an optional element of a SET or SEQUENCE; and

c) ignore all unknown elements in SETs, at the end of SEQUENCEs, or in CHOICEs where the CHOICE is itself an optional element of a SET or SEQUENCE.

NOTE 2 – Implementations may as a local option ignore certain additional elements in a Directory PDU. In particular, some unknown named numbers and unknown CHOICEs in mandatory elements of SETs and SEQUENCEs can be ignored without invalidating the operation. The identification of such elements is not specified in IN CS-2.

7.6 Conformance

This subclause defines the requirements for conformance to this specification.

7.6.1 Conformance by SCFs

An SCF implementation claiming conformance to this specification shall satisfy the requirements specified in 7.6.1.1 through 7.6.1.3.

7.6.1.1 Statement requirements

The following shall be stated:

a) the operations of the **iNdirectoryAccessAC** application-context that the SCF is capable of invoking for which conformance is claimed;

b) the security-level(s) for which conformance is claimed (none, simple, strong);

c) the extensions listed in the Table of 7.3.1 of ITU-T Rec. X.511 | ISO/IEC 9594-3, that the SCF is capable of initiating for which conformance is claimed.

7.6.1.2 Static requirements

An SCF shall:

a) have the capability of supporting the **iNdirectoryAccessAC** application-context as defined by its abstract syntax in 7.5.2.1;

b) conform to the extensions for which conformance was claimed in 7.6.1.1 c).

7.6.1.3 Dynamic requirements

An SCF shall:

a) conform to the mapping onto used services defined in 18.1.6;

b) shall conform to the rules of extensibility procedures defined in 7.5.5.2.

7.6.2 Conformance by SDFs

An SDF implementation claiming conformance to this specification shall satisfy the requirements specified in 7.6.2.1 through 7.6.2.3.

7.6.2.1 Statement requirements

The following shall be stated:

a) the application-context for which conformance is claimed. The present version of this Recommendation only requires conformance to the **iNdirectoryAccessAC** application-context;

NOTE – An application context shall not be divided except as stated herein; in particular, conformance shall not be claimed to particular operations.

b) the security-level(s) for which conformance is claimed (none, simple, strong);

c) the attribute types for which conformance is claimed and whether for attributes based on the syntax **DirectoryString,** conformance is claimed for the **UNIVERSAL STRING** choice;

d) the object classes, for which conformance is claimed;

e) the extensions listed in the Table of 7.3.1 of ITU-T Rec. X.511 | ISO/IEC 9594-3, that the SDF is capable of responding to for which conformance is claimed;

f) whether conformance is claimed for collective attributes as defined in 8.8 of ITU-T Rec. X.501 | ISO/IEC 9594-2 and 7.6, 7.8.2 and 9.2.2 of ITU-T Rec. X.511 | ISO/IEC 9594-3;

g) whether conformance is claimed for hierarchical attributes as defined in 7.6, 7.8.2 and 9.2.2 of ITU-T Rec. X.511 | ISO/IEC 9594-3;

h) the operational attribute types defined in ITU-T Rec. X.501 | ISO/IEC 9594-2 and any other operational attribute types for which conformance is claimed;

i) whether conformance is claimed for return of alias names as described in 7.7.1 of ITU-T Rec. X.511 | ISO/IEC 9594-3;

j) whether conformance is claimed for indicating that returned entry information is complete, as described in 7.7.6 of ITU-T Rec. X.511 | ISO/IEC 9594-3;

k) whether conformance is claimed for modifying the object class attribute to add and/or remove values identifying auxiliary object classes, as described in 11.3.2 of ITU-T Rec. X.511 | ISO/IEC 9594-3;

l) whether conformance is claimed to Basic Access Control;

m) whether conformance is claimed to Simplified Access Control;

n) the name bindings for which conformance is claimed;

o) whether the SDF is capable of administering collective attributes, as defined in ITU-T Rec. X.501 | ISO/IEC 9594-2;

p) whether conformance is claimed for contexts.

7.6.2.2 Static requirements

An SDF shall:

a) have the capability of supporting the application-contexts for which conformance is claimed as defined by their abstract syntax in 7.5.2.1;

b) have the capability of supporting the information framework defined by its abstract syntax in ITU-T Rec. X.501 | ISO/IEC 9594-2;

c) have the capability of supporting the attribute types for which conformance is claimed; as defined by their abstract syntaxes;

d) have the capability of supporting the object classes for which conformance is claimed, as defined by their abstract syntaxes;

e) conform to the extensions for which conformance was claimed in 7.6.2.1;

f) if conformance is claimed for collective attributes, have the capability of performing the related procedures defined in 7.6, 7.8.2 and 9.2.2 of ITU-T Rec. X.511 | ISO/IEC 9594-3;

g) if conformance is claimed for hierarchical attributes, have the capability of performing the related procedures defined in 7.6, 7.8.2 and 9.2.2 of ITU-T Rec. X.511 | ISO/IEC 9594-3;

h) have the capability of supporting the operational attribute types for which conformance is claimed;

i) if conformance is claimed to Basic Access Control, have the capability of holding ACI items that conform to the definitions of Basic Access Control;

j) if conformance is claimed to Simplified Access Control, have the capability of holding ACI items that conform to the definitions of Simplified Access Control.

7.6.2.3 Dynamic requirements

An SDF shall:

a) conform to the mapping onto used services defined in 18.1.6;

b) conform to the rules of extensibility procedures defined in 7.5.5.3;

c) if conformance is claimed to Basic Access Control, have the capability of protecting information within the SDF in accordance with the procedures of Basic Access Control;

d) if conformance is claimed to Simplified Access Control, have the capability of protecting information within the SDF in accordance with the procedures of Simplified Access Control.

7.7 ASN.1 modules for the SCF-SDF interface

The following set of ASN.1 modules define the SCF-SDF interface for IN CS-2. They contain all the modifications to the Directory specifications as required for the support of Intelligent Networks.

The modules also contain the definitions which are impacted by these modifications because they make use of a modified type.

7.7.1 IN-CS2-SDF-InformationFramework module

This module contains the enhancements made to the X.501 Recommendation (InformationFramework module) to meet the IN CS-2 needs.

```
IN-CS2-SDF-InformationFramework
        {itu-t recommendation q 1228 module(0) sdfInformationFramework(9) version1(0) }
DEFINITIONS ::=
BEGIN
-- EXPORTS ALL--
-- types and values are exported for use in the ASN.1 modules which define the IN profile of the Directory
-- Abstract Service,  the Directory Access Protocol and the Directory Information Shadowing Protocol.
-- The types and values defined in this module are exported for use in the other ASN.1 modules contained
-- within the Directory Specifications, and for the use of other applications which will use them to access
-- Directory services. Other applications may use them for their own purposes, but this will not constrain
-- extensions and modifications needed to maintain or improve the Directory service.
IMPORTS
    informationFramework, upperBounds, selectedAttributeTypes
        FROM UsefulDefinitions {joint-iso-ccitt ds(5) module(1) usefulDefinitions(0) 3}
    ATTRIBUTE, OBJECT-CLASS, objectClass, aliasedEntryName
        FROM InformationFramework  informationFramework
    DirectoryString{}, objectIdentifierFirstComponentMatch
        FROM SelectedAttributeTypes   selectedAttributeTypes
    ub-schema
        FROM UpperBounds                upperBounds
    id-soa-methodRuleUse
        FROM IN-CS2-object-identifiers
            { itu-t recommendation q 1228 modules(0) in-cs2-object-identifiers(17) version1(0) }
    ;
-- attribute data types --
-- Definition of the following information object set is deferred, perhaps to standardised
-- profiles or to protocol implementation conformance statements. The set is required to
-- specify a table constraint on the values component of Attribute, the value component
-- of AttributeTypeAndValue, and the assertion component of AttributeValueAssertion.
SupportedAttributes ATTRIBUTE ::= { objectClass | aliasedEntryName , ...}
-- METHOD information object class specification --
METHOD ::= CLASS {
    &InputAttributes        ATTRIBUTE OPTIONAL,
    &SpecificInput          OPTIONAL,
    &OutputAttributes       ATTRIBUTE OPTIONAL,
    &SpecificOutput         OPTIONAL,
    &description            PrintableString OPTIONAL,
    &id                     OBJECT IDENTIFIER UNIQUE}
WITH SYNTAX {
    [ INPUT ATTRIBUTES          &InputAttributes ]
    [SPECIFIC-INPUT             &SpecificInput ]
    [OUTPUT ATTRIBUTES          &OutputAttributes ]
    [SPECIFIC-OUTPUT            &SpecificOutput ]
    [BEHAVIOUR                  &description]
    ID                          &id}

DITMethodUse    ::=     SEQUENCE {
    objectClass         OBJECT-CLASS.&id,
    methods             [1]     SET OF METHOD.&id }

METHOD-USE-RULE ::= CLASS {
    &objectClassType            OBJECT-CLASS.&id        UNIQUE,
    &Mandatory                  METHOD }
WITH SYNTAX {
    OBJECT-CLASS TYPE           &objectClassType
    METHODS                     &Mandatory }
-- attributes --
```

```
methodUse ATTRIBUTE ::= {
        WITH SYNTAX                          MethodUseDescription
        EQUALITY MATCHING RULE               objectIdentifierFirstComponentMatch
        USAGE                                directoryOperation
        ID                                   id-soa-methodRuleUse }

MethodUseDescription     ::=     SEQUENCE {
        identifier               OBJECT-CLASS.&id,
        name                     SET OF DirectoryString { ub-schema } OPTIONAL,
        description              DirectoryString { ub-schema } OPTIONAL,
        obsolete                 BOOLEAN DEFAULT FALSE,
        information      [0]     SET OF METHOD.&id }
END
```

7.7.2 IN-CS2-SDF-BasicAccessControl Module

This module contains the enhancements made to the X.501 Recommendation (InformationFramework module) to meet the IN needs.

```
IN-CS2-SDF-BasicAccessControl
                { itu-t recommendation q 1228 module(0) sdfBasicAccessControl(10) version1(0) }
DEFINITIONS ::=
BEGIN
-- EXPORTS All --
-- The types and values defined in this module are exported for use in the other ASN.1 modules contained
-- within the Directory Specifications, and for the use of other applications which will use them to access
-- Directory services. Other applications may use them for their own purposes, but this will not constrain
-- extensions and modifications needed to maintain or improve the Directory service.
IMPORTS
        informationFramework, upperBounds, selectedAttributeTypes, basicAccessControl,
        directoryAbstractService
                FROM UsefulDefinitions {joint-iso-ccitt ds(5) module(1) usefulDefinitions(0) 3}
        ATTRIBUTE, AttributeType, AttributeTypeAndValue, SubtreeSpecification, ContextAssertion
                FROM InformationFramework informationFramework
        id-aca-prescriptiveACI, id-aca-entryACI, id-aca-subentryACI,
        sdf-InformationFramework
                FROM IN-CS2-object-identifiers
                        { itu-t recommendation q 1228 modules(0) in-cs2-object-identifiers(17) version1(0) }
        ub-tag
                FROM UpperBounds upperBounds
        METHOD
                FROM IN-CS2-SDF-InformationFramework
        sdf-InformationFramework
        Filter
                FROM DirectoryAbstractService  directoryAbstractService
        NameAndOptionalUID, directoryStringFirstComponentMatch, DirectoryString{}
                FROM SelectedAttributeTypes selectedAttributeTypes

-- types --
ACIItem              ::=          SEQUENCE {
        identificationTag          DirectoryString { ub-tag },
        precedence                 Precedence,
        authenticationLevel        AuthenticationLevel,
        itemOrUserFirst            CHOICE {
                itemFirst          [0]     SEQUENCE {
                        protectedItems     ProtectedItems,
                        itemPermissions    SET OF ItemPermission },
                userFirst          [1]     SEQUENCE {
                        userClasses        UserClasses,
                        userPermissions    SET OF UserPermission }}}
```

```
ProtectedItems       ::=     SEQUENCE {
          entry                                  [0]      NULL OPTIONAL,
          allUserAttributeTypes                  [1]      NULL OPTIONAL,
          attributeType                          [2]      SET OF AttributeType OPTIONAL,
          allAttributeValues                     [3]      SET OF AttributeType OPTIONAL,
          allUserAttributeTypesAndValues         [4]      NULL OPTIONAL,
          attributeValue                         [5]      SET OF AttributeTypeAndValue OPTIONAL,
          selfValue                              [6]      SET OF AttributeType OPTIONAL,
          rangeOfValues                          [7]      Filter OPTIONAL,
          maxValueCount                          [8]      SET OF MaxValueCount OPTIONAL,
          maxImmSub                              [9]      INTEGER OPTIONAL,
          restrictedBy                           [10]     SET OF RestrictedValue OPTIONAL,
          contexts                               [11]     SET OF ContextAssertion OPTIONAL,
          entryMethods                           [30]     SET OF MethodIDs OPTIONAL}
MethodIDs ::=        METHOD.&id
UserClasses  ::=     SEQUENCE {
          allUsers        [0]     NULL OPTIONAL,
          thisEntry       [1]     NULL OPTIONAL,
          name            [2]     SET OF NameAndOptionalUID OPTIONAL,
          userGroup       [3]     SET OF NameAndOptionalUID OPTIONAL,
                                          -- dn component must be the name of an
                                          -- entry of GroupOfUniqueNames
          subtree         [4]     SET OF SubtreeSpecification OPTIONAL}
ItemPermission                            ::=     SEQUENCE {
          precedence              Precedence OPTIONAL,
                                          -- defaults to precedence in ACIItem --
          userClasses             UserClasses,
          grantsAndDenials        GrantsAndDenials }
UserPermission                            ::=     SEQUENCE {
          precedence              Precedence OPTIONAL,
                                          -- defaults to precedence in ACIItem
          protectedItems          ProtectedItems,
          grantsAndDenials        GrantsAndDenials }
GrantsAndDenials                          ::=     BIT STRING {
          -- permissions that may be used in conjunction
          -- with any component of ProtectedItems
          grantAdd                        (0),
          denyAdd                         (1),
          grantDiscloseOnError            (2),
          denyDiscloseOnError             (3),
          grantRead                       (4),
          denyRead                        (5),
          grantRemove                     (6),
          denyRemove                      (7),
          -- permissions that may be used only in conjunction
          -- with the entry component
          grantBrowse                     (8),
          denyBrowse                      (9),
          grantExport                     (10),
          denyExport                      (11),
          grantImport                     (12),
          denyImport                      (13),
          grantModify                     (14),
          denyModify                      (15),
          grantRename                     (16),
          denyRename                      (17),
          grantReturnDN                   (18),
          denyReturnDN                    (19),
          -- permissions that may be used in conjunction
          -- with any component, except entry, of ProtectedItems
          grantCompare                    (20),
          denyCompare              (21),
          grantFilterMatch         (22),
          denyFilterMatch          (23),
          -- permissions that may be used in conjunction
          -- with entryMethod component of ProtectedItems
          grantExecuteMethod       (30),
          denyExecuteMethod        (31) }
-- attributes --
prescriptiveACI          ATTRIBUTE ::=    {
          WITH SYNTAX                       ACIItem
          EQUALITY MATCHING RULE            directoryStringFirstComponentMatch
          USAGE                             directoryOperation
          ID                                id-aca-prescriptiveACI }
entryACI                 ATTRIBUTE ::=    {
          WITH SYNTAX                       ACIItem
          EQUALITY MATCHING RULE            directoryStringFirstComponentMatch
          USAGE                             directoryOperation
          ID                                id-aca-entryACI }
subentryACI              ATTRIBUTE ::=    {
          WITH SYNTAX                       ACIItem
          EQUALITY MATCHING RULE            directoryStringFirstComponentMatch
          USAGE                             directoryOperation
          ID                                id-aca-subentryACI }
END
```

7.7.3 IN-CS2-SCF-SDF-Operations Module

```
IN-CS2-SCF-SDF-Operations
                {itu-t recommendation q 1228 module(0) scf-sdf-operations(11) version1(0) }
DEFINITIONS ::=
BEGIN
-- EXPORTS All --
-- The types and values defined in this module are exported for use in the other ASN.1 modules contained
-- within the IN Directory Specifications, and for the use of other applications which will use them to access
-- IN Directory services. Other applications may use them for their own purposes, but this will not constrain
-- extensions and modifications needed to maintain or improve the Directory service.
IMPORTS
        informationFramework, distributedOperations, authenticationFramework, upperBounds,
        directoryAbstractService, enhancedSecurity
                FROM UsefulDefinitions {joint-iso-ccitt ds(5) module(1) usefulDefinitions(0) 3}
        CONTEXT, Context, DistinguishedName, Name
                FROM InformationFramework informationFramework
        OperationProgress, ReferenceType, Exclusions, AccessPoint, ContinuationReference
                FROM  DistributedOperations  distributedOperations
        CertificationPath, SIGNED {}, SIGNATURE {}, AlgorithmIdentifier
                FROM  AuthenticationFramework authenticationFramework
        id-avc-assignment,
        contexts, ros-InformationObjects, sdf-InformationFramework
                FROM IN-CS2-object-identifiers
                        { ccitt recommendation q 1228 modules(0) in-cs2-object-identifiers(17) version1(0) }
        basicServiceContext, lineIdentityContext
                FROM IN-Contexts    contexts
        Code, OPERATION, ERROR
                FROM Remote-Operations-Information-Objects ros-InformationObjects
        inEmptyUnbind
                FROM IN-CS2-classes {ccitt recommendation q 1228 modules(0) in-cs2-classes(4) version1(0)}
        METHOD
                FROM  IN-CS2-SDF-InformationFramework
        sdf-InformationFramework
        OPTIONALLY-PROTECTED{}, DIRQOP
                FROM EnhancedSecurity enhancedSecurity
        CommonArguments, CommonResults, attributeError, nameError, serviceError, securityError, referral,
        updateError
                FROM DirectoryAbstractService            directoryAbstractService
        ;
execute OPERATION ::= {
        ARGUMENT        ExecuteArgument
        RESULT          ExecuteResult
        ERRORS          { attributeError | nameError |
                          serviceError | referral |
                         securityError |
                          updateError | executionError }
        CODE            id-opcode-execute }
ExecuteArgument ::=  OPTIONALLY-PROTECTED {
        SET {
                object                  [0] Name,
                method-id               [1] METHOD.&id({SupportedMethods}),
                input-assertions        [2] SEQUENCE OF SEQUENCE {
                                                type
        METHOD.&InputAttributes.&id({SupportedMethods}{@method-id}),
                                                values SET OF
METHOD.&InputAttributes.&Type({SupportedMethods}{@method-id,@.type}) OPTIONAL,
                                                valuesWithContext [0] SET OF SEQUENCE {
                                                        value  [0]
        METHOD.&InputAttributes.&Type({SupportedMethods}{@method-id,@.type})        OPTIONAL,
                                                contextList [1]         SET OF Context
                                                } OPTIONAL
                                        } OPTIONAL,
        specific-input          [3] METHOD.&SpecificInput({SupportedMethods}{@method-id}) OPTIONAL,
                COMPONENTS OF CommonArguments },
        DIRQOP.&dapModifyEntryArg-QOP{@qop} }
ExecuteResult ::= OPTIONALLY-PROTECTED {
        SET {
                method-id               [1] METHOD.&id({SupportedMethods}),
                output-assertions       [2] SEQUENCE OF SEQUENCE {
                                                type
        METHOD.&OutputAttributes.&id({SupportedMethods}{@method-id}),
                                        values SET OF
METHOD.&OutputAttributes.&Type({SupportedMethods}{@method-id,@.type}) OPTIONAL,
                                        valuesWithContext [0] SET OF SEQUENCE {
                                                value  [0]
        METHOD.&OutputAttributes.&Type({SupportedMethods}{@method-id,@.type})      OPTIONAL,
                                        contextList    [1] SET OF Context
                COMPONENTS OF CommonResults },
        DIRQOP.&dapModifyEntryRes-QOP{@qop} }
SupportedMethods METHOD ::= { ... }
in-directoryUnbind  OPERATION ::= inEmptyUnbind
assignmentContext  CONTEXT ::= {
        WITH SYNTAX                     DistinguishedName
        ID                              id-avc-assignment }

executionError ERROR ::= {
        PARAMETER       OPTIONALLY-PROTECTED {
                                        SET {
                                                problem         [0]     ExecutionProblem ,
                                                COMPONENTS OF CommonResults },
                                        DIRQOP.&dirErrors-QOP{@dirqop} }
        CODE                            id-errcode-executionError }
```

```
ExecutionProblem  ::=  INTEGER {
        missingInputValues (1),
        executionFailure(2) }
-- object identifier assignment
-- error codes
id-errcode-executionError          Code ::= local:10
-- operation codes
id-opcode-execute                  Code ::=local:10

END
```

7.7.4 IN-CS2-SCF-SDF-Protocol Module

This subclause includes all of the ASN.1 type and value definitions contained in this Directory Specification, in the form of the ASN.1 module, "IN-CS2-SCF-SDF-Protocol ".

```
IN-CS2-SCF-SDF-Protocol {itu-t recommendation q 1218 modules(0) in-scf-sdf-protocol(12) version1(0)}
DEFINITIONS ::=
BEGIN
-- EXPORTS All --
-- The types and values defined in this module are exported for use in the other ASN.1 modules contained
-- within the Directory Specifications, and for the use of other applications which will use them to access
-- Directory services. Other applications may use them for their own purposes, but this will not constrain
-- extensions and modifications needed to maintain or improve the Directory service.
IMPORTS
        directoryAbstractService , directorySecurityExchanges, protocolObjectIdentifiers
                FROM  UsefulDefinitions  ds-UsefulDefinitions
        ROS-OBJECT-CLASS, CONTRACT, OPERATION-PACKAGE, CONNECTION-PACKAGE,
        OPERATION
                FROM Remote-Operations-Information-Objects ros-InformationObjects

        Bind{}, Unbind{}
                FROM Remote-Operations-Generic-ROS-PDUs ros-genericPDUs

        TCMessage {}
                FROM  TCAPMessages tc-Messages

        id-ac-indirectoryAccessAC, id-ac-inExtendedDirectoryAccessAC, id-rosObject-dua, id-rosObject-
        directory,
        id-rosObject-dapDSA,
        id-contract-dap, id-contract-dapExecute, id-package-dapConnection, id-package-search, id-package-
        modify,
        id-package-execute,
        id-as-indirectoryOperationsAS, id-as-inExtendedDirectoryOperationsAS, id-as-indirectoryBindingAS,
        id-as-inSESEAS,
        id-ac-inExtendedDirectoryAccessWith3seAC, id-ac-indirectoryAccessWith3seAC,
        ros-InformationObjects, ros-genericPDUs, tc-Messages, tc-NotationExtensions, sese-APDUs,
        ds-UsefulDefinitions, scf-sdf-Operations
                FROM IN-CS2-object-identifiers
                        {itu-t recommendation q 1228 modules(0) in-cs2-object-identifiers (17) version1 (0)}
        directoryBind, search, addEntry, removeEntry, modifyEntry
                FROM DirectoryAbstractService directoryAbstractService
        SESEapdus{}, NoInvocationId
                FROM SeseAPDUs sese-APDUs
        spkmThreeWay
                FROM DirectorySecurityExchanges directorySecurityExchanges
        id-se-threewayse
                FROM ProtocolObjectIdentifiers protocolObjectIdentifiers
        execute, in-directoryUnbind
                FROM IN-CS2-SCF-SDF-Operations
        scf-sdf-Operations
        ;
-- application contexts --
iNdirectoryAccessAC APPLICATION-CONTEXT ::= {
        CONTRACT                        dapContract
        DIALOGUE MODE                   structured
        TERMINATION                             basic
        ABSTRACT SYNTAXES               {dialogue-abstract-syntax |
                                                inDirectoryOperationsAbstractSyntax |
                                                inDirectoryBindingAbstractSyntax}
        APPLICATION CONTEXT NAME        id-ac-indirectoryAccessAC}

iNdirectoryAccessWith3seAC APPLICATION-CONTEXT ::= {
        CONTRACT                        dapContract
        DIALOGUE MODE                   structured
        TERMINATION                             basic
        ADDITIONAL ASE                  {id-se-threewayse}
        ABSTRACT SYNTAXES                       {dialogue-abstract-syntax |
                                        inDirectoryOperationsAbstractSyntax |
                                        inDirectoryBindingAbstractSyntax |
                                        inSESEAbstractSyntax }
        APPLICATION CONTEXT NAME        id-ac-indirectoryAccessWith3seAC}

inExtendedDirectoryAccessAC APPLICATION-CONTEXT ::= {
        CONTRACT                        dapExecuteContract
        DIALOGUE MODE                   structured
        TERMINATION                             basic
        ABSTRACT SYNTAXES               {dialogue-abstract-syntax |
                                                inExtendedDirectoryOperationsAbstractSyntax |
                                                inDirectoryBindingAbstractSyntax}
        APPLICATION CONTEXT NAME        id-ac-inExtendedDirectoryAccessAC}

inExtendedDirectoryAccessWith3seAC APPLICATION-CONTEXT ::= {
        CONTRACT                        dapExecuteContract
        DIALOGUE MODE                   structured
        TERMINATION                     basic
        ADDITIONAL ASE                  {id-se-threewayse}
        ABSTRACT SYNTAXES               {dialogue-abstract-syntax |
                                        inExtendedDirectoryOperationsAbstractSyntax |
                                        inDirectoryBindingAbstractSyntax |
                                        inSESEAbstractSyntax }
        APPLICATION CONTEXT NAME        id-ac-inExtendedDirectoryAccessWith3seAC}
-- ROS-objects --
dua ROS-OBJECT-CLASS ::= {
        INITIATES  {dapContract| dapExecuteContract}
        ID              id-rosObject-dua}

directory ROS-OBJECT-CLASS ::= {
        RESPONDS  {dapContract| dapExecuteContract}
        ID              id-rosObject-directory}

dap-dsa ROS-OBJECT-CLASS ::= {
        RESPONDS  {dapContract| dapExecuteContract}
        ID              id-rosObject-dapDSA}
-- contracts --
dapContract CONTRACT ::= {
        CONNECTION                      dapConnectionPackage
        INITIATOR CONSUMER OF   {searchPackage | modifyPackage}
        ID                              id-contract-dap}
```

```
dapExecuteContract CONTRACT ::= {
        CONNECTION                    dapConnectionPackage
        INITIATOR CONSUMER OF    {searchPackage | modifyPackage | executePackage}
        ID                            id-contract-dapExecute}
-- connection package --
dapConnectionPackage CONNECTION-PACKAGE ::= {
        BIND          directoryBind
        UNBIND        in-directoryUnbind
        ID            id-package-dapConnection}
--  search and modify packages
searchPackage OPERATION-PACKAGE ::= {
        CONSUMER INVOKES              {search}
        ID                            id-package-search}

modifyPackage OPERATION-PACKAGE ::= {
        CONSUMER INVOKES              {addEntry | removeEntry | modifyEntry}
        ID                            id-package-modify}

executePackage OPERATION-PACKAGE ::= {
        CONSUMER INVOKES   {execute}
        ID                            id-package-execute}
-- abstract-syntaxes --
inDirectoryOperationsAbstractSyntax ABSTRACT-SYNTAX ::= {
        BasicDAP-PDUs
        IDENTIFIED BY    id-as-indirectoryOperationsAS}

BasicDAP-PDUs ::= TCMessage {{DAP-Invokable},{DAP-Returnable}}

DAP-Invokable  OPERATION ::= {search | addEntry | removeEntry | modifyEntry}

DAP-Returnable OPERATION ::= {search | addEntry | removeEntry | modifyEntry}

inExtendedDirectoryOperationsAbstractSyntax ABSTRACT-SYNTAX ::= {
        Extended-BasicDAP-PDUs
        IDENTIFIED BY    id-as-inExtendedDirectoryOperationsAS}

Extended-BasicDAP-PDUs ::= TCMessage {{Extended-DAP-Invokable},{Extended-DAP-Returnable}}

Extended-DAP-Invokable  OPERATION ::= {search | addEntry | removeEntry | modifyEntry | execute}

Extended-DAP-Returnable OPERATION ::= {search | addEntry | removeEntry | modifyEntry | execute}

inDirectoryBindingAbstractSyntax        ABSTRACT-SYNTAX ::= {
        DAPBinding-PDUs
        IDENTIFIED BY    id-as-indirectoryBindingAS}

DAPBinding-PDUs ::= CHOICE {
        bind    Bind {directoryBind},
        unbind        Unbind {in-directoryUnbind}}

inSESEAbstractSyntax     ABSTRACT-SYNTAX ::= {
        SESEapdus {{spkmThreeWay},NoInvocationId}
        IDENTIFIED BY    {id-as-inSESEAS}}
END
```

8 SDF/SDF interface

8.1 Introduction to the IN X.500 DSP and DISP Subset

The purpose of the SDF-SDF interface is to allow the transfer of copies of service profiles from one SDF to another and to manage the copies within the database network. The X.500 functionalities cover more than the functionalities needed to fulfil the CS-2 requirements. This clause tries to indicate which aspects of the DSP and DISP should be considered and supported and which should be left out or ignored. Profiling is used as a means to present the status of the different parameters.

It is important to mention that the number of parameters carried in a message should be minimised, to reduce the load on the signalling traffic and processing time. This is the reason why the parameters are removed unless they are absolutely necessary when they are sent. On reception, removed parameters should not be treated but should be understood by the receiving entity. This allows the extension of the profile in the future according to its actual description in the 1993 edition of the Directory.

For convenience and clarity, this profile is defined using ASN.1 subtyping facilities; however, these definitions do not form a protocol specification. This simply indicates which parameters an implementation should not send. It does not change the behaviour of the receiving entity which shall still be capable of decoding values which conform to the original definition of the DSP and DISP. Nevertheless elements that are excluded by subtyping should be understood but not treated.

8.2 Working assumptions

Several assumptions were used to design the DSP and DISP for IN CS-2. They are as follows:

Assumption 1: The agreements between network operators concerning the transfer of data are defined off-line (e.g. management operations). The establishOperationalBinding operation is only used to activate an agreement.

Assumption 2: The agreements cannot be modified by an on-line operation.

Assumption 3: The terminateOperationalBinding operation is used to end an agreement between two network operators. This means that the copy held by the shadow-consumer is no longer maintained. It should not be used and should be deleted. However the agreement could be required for future associations between the two networks, therefore this information should be retained.

Assumption 4: The shadow updates are initiated by the shadow supplier who holds the master copy. Therefore modifications of the copies are not performed on the shadowed copies but only on the master copy. The modification requests are passed to the master copy by using a chained operation. Copies are updated on changes.

Assumption 5: Only direct references are used in DSAs. Operations can only be chained once. If the operation cannot be fulfilled after one chaining, a referral should be sent back.

Assumption 6: It is not possible to make a copy of a copy. One should refer to the master copy to get a copy.

Assumption 7: The shadowing mechanism is initiated by a specific DAP operation or by a management operation. The management operation is for further study.

Assumption 8: The time when a shadowing agreement is terminated depends on the type of service. In most cases it will be based on the number of copies. Once the maximum number of copies is reached for a part of a DIT, then the oldest copy has to be deleted and its agreement de-activated. The maximum number of copies can be equal to one.

Assumption 9: An SDF-SDF operation cannot be abandoned. If an operation takes too much time, its timer expires and there is no need to abandon it.

8.3 The IN X.500 DISP Subset

8.3.1 Shadowing agreement specification

The Shadowing agreement is specified as:

```
IN-ShadowingAgreementInfo ::= ShadowingAgreementInfo (
        WITH COMPONENTS {
                ...,
                master                  ABSENT,
                secondaryShadows        ABSENT})
```

shadowSubject specifies the subtree, entries and attributes to shadow. The components of **UnitOfReplication** are defined in 9.2/X.525.

updateMode specifies when updates of a shadowed area are scheduled to occur. The components of **updateMode** are defined in 9.3/X.252.

master contains the access point of the DSA containing the mastered area. "As this information is already known by the DSA it is not required for IN."

secondaryShadows permits secondary shadow information to be subsequently supplied to the shadow supplier. The secondary shadows are ignored in the IN context (assumption 5), then this component should not be included.

8.3.2 DSA Shadow Bind

A **dSAShadowBind** operation is used at the beginning of a period of providing shadows.

```
        in-dSAShadowBind OPERATION ::= in-DirectoryBind
```

IN CS-2 uses the in-DirectoryBind operation as specified in 7.3.2.1.

8.3.3 IN-DSA Shadow Unbind

The **in-DSAShadowUnbind** operation replaces the X.525 **dSAShadowUnbind** operation to provide class 4 operation behaviour for unbind procedures.

```
        in-DSAShadowUnbind  OPERATION ::= inEmptyUnbind
```

8.3.4 Coordinate Shadow Update

The **inCoordinateShadowUpdate** operation is used by the shadow supplier to indicate the shadowing agreement for which it intends to send updates.

```
        inCoordinateShadowUpdate  OPERATION ::= {
                ARGUMENT        IN-CoordinateShadowUpdateArgument
                RESULT          IN-CoordinateShadowUpdateResult
                ERRORS          {shadowError}
                CODE            id-opcode-coordinateShadowUpdate}

        IN-CoordinateShadowUpdateArgument ::= CoordinateShadowUpdateArgument (
                WITH COMPONENTS {
                        ...,
                        updateStrategy      (standard:{total | incremental})})
```

```
IN-CoordinateShadowUpdateResult     ::=     CoordinateShadowUpdateResult(
        WITH COMPONENTS {
                ...,
                null            PRESENT})
```

The various parameters have the meanings defined below:

a) The **agreementID** argument identifies the shadowing agreement.

b) The **lastUpdate** argument indicates the shadow supplier's understanding of the time at which the last update for this agreement was sent and is the time as provided by the shadow supplier DSA. This argument may only be omitted in the first instance of either a **inCoordinateShadowUpdate** or **inRequestShadowUpdate** operation for a particular shadowing agreement

c) The **updateStrategy** argument identifies the update strategy the shadow supplier intends to use for this update. For IN CS-2, a total or incremental replacement strategy should be used. The "NoChanges" option will not be used.

d) The **securityParameters** argument is defined in 7.10 of ITU-T Rec. X.511 | ISO/IEC 9594-3.

8.3.5 Update Shadow

An **inUpdateShadow** operation is invoked by the shadow supplier to send updates to the shadow consumer for a unit of replication. Prior to this operation being initiated, a **inCoordinateShadowUpdate** or **inRequestShadowUpdate** operation must have been successfully completed for the identified shadowing agreement.

```
inUpdateShadow  OPERATION ::= {
        ARGUMENT        IN-UpdateShadowArgument
        RESULT          IN-UpdateShadowResult
        ERRORS          {shadowError}
        CODE            id-opcode-updateShadow}

IN-UpdateShadowArgument ::= UpdateShadowArgument (
        WITH COMPONENTS {
                ...,
                updatedInfo          (IN-RefreshInformation)})

IN-UpdateShadowResult  ::=      UpdateShadowUpdateResult(
        WITH COMPONENTS {
                ...,
                null            PRESENT})
```

The various parameters have the meanings as defined below:

a) The **agreementID** identifies the shadowing agreement that has been established.

b) The **updateTime** argument is supplied by the shadow supplier. This time is used during the next **inCoordinateShadowUpdate** or **inRequestShadowUpdate** to ensure that the shadow supplier and shadow consumer have a common view of the shadowed information.

c) The **updateWindow** argument, when present, indicates the next window during which the shadow supplier expects to send an update.

d) The **updatedInfo** argument provides the information required by the shadow consumer to update its shadowed information. The semantics of the information conveyed in this parameter shall result in the shadow consumer reflecting the changes supplied.

e) The **securityParameters** argument is defined in 7.10 of ITU-T Rec. X.511 | ISO/IEC 9594-3.

```
IN-RefreshInformation ::= RefreshInformation (
        WITH COMPONENTS {
                ...,
                otherStrategy                   ABSENT})
```

The various parameters have the meanings as defined below:

a) **noRefresh** indicates that there have been no changes to the shadowed information from the previous instance to the present. This may be used where an **updateShadow** operation must be supplied at a certain interval defined in the shadowing agreement (**updateMode**), but no modification has actually occurred.

b) **total** provides a new instance of the shadowed information. The incremental strategy should be preferably used because it saves signalling.

c) **incremental** provides, instead of a complete replacement of the shadowed information, only the changes which have occurred to that shadowed information between **lastUpdate** in the most recent **inCoordinateShadowUpdate** (or **inRequestShadowUpdate**) request and **updateTime** in the current **inUpdateShadow** request (or **inRequestShadowUpdate** response).

d) **otherStrategy** provides the ability to send updates by mechanisms outside the scope of the Directory Specification. For IN CS-2, either a total or incremental strategy should be used.

Should the request succeed, a result will be returned, although no information will be conveyed with it.

Should the request fail, a **shadowError** shall be reported. Circumstances under which the particular shadow problems will be returned are defined in 11.3.3/X.525.

8.3.6 Request Shadow Update

An **inRequestShadowUpdate** operation is used by the shadow consumer to request updates from the shadow supplier.

```
inRequestShadowUpdate  OPERATION ::= {
        ARGUMENT        IN-RequestShadowUpdateArgument
        RESULT          IN-RequestShadowUpdateResult
        ERRORS          {shadowError}
        CODE            id-opcode-RequestShadowUpdate}

IN-RequestShadowUpdateArgument ::= RequestShadowUpdateArgument (
        WITH COMPONENTS {
                ...,
                requestedStrategy   (standard:{incremental | total})})

IN-RequestShadowUpdateResult ::= RequestShadowUpdateResult(
        WITH COMPONENTS {
                ...,
                null            PRESENT})
```

The various parameters have the meanings as defined below:

a) The **agreementID** identifies the shadowing agreement.

b) The **lastUpdate** argument is the time provided by the shadow supplier in the most recent successful update. This argument may only be omitted in the first instance of either a **inCoordinateShadowUpdate** or **inRequestShadowUpdate** operation for a particular shadowing agreement.

c) The **requestedStrategy** argument identifies the type of update being requested by the shadow consumer. The shadow consumer may request either an **incremental** or a **total** update from the shadow supplier.

d) The **securityParameters** argument is defined in 7.10 of ITU-T Rec. X.511 | ISO/IEC 9594-3.

8.4 The IN X.500 DSP Subset

8.4.1 Information types and common procedures

8.4.1.1 Chaining Arguments

The **ChainingArguments** are present in each chained operation, to convey to a DSA the information needed to successfully perform its part of the overall task:

```
IN-ChainingArguments ::= ChainingArguments (
    WITH COMPONENTS {
        ...,
        aliasDereferenced       ABSENT,
        aliasedRDNs             ABSENT,
        returnCrossRefs         ABSENT,
        info                    ABSENT,
        timeLimit               ABSENT,
        excludeShadows          ABSENT,
        nameResolveOnMaster     ABSENT})
```

The various components have the meanings as defined below:

a) The **originator** component conveys the name of the originator of the request unless already specified in the security parameters. If **requester** is present in **CommonArguments**, this argument may be omitted.

b) The **targetObject** component conveys the name of the object whose directory entry is being routed to. The role of this object depends on the particular operation concerned: it may be the object whose entry is to be operated on, or which is to be the base object for a request or sub-request involving multiple objects (e.g. **ChainedModify**). This component can be omitted only if it has the same value as the object or base object parameter in the chained operation, in which case its implied value is that value.

c) The **operationProgress** component is used to inform the DSA of the progress of the operation, and hence of the role which it is expected to play in its overall performance. Even though direct knowledge references are assumed, this parameter is deemed applicable for IN CS-2 since an SDF to which an operation is chained can still respond with a continuation reference in the chained operation dsaReferral error.

d) The **traceInformation** component is used to prevent looping among DSAs when chaining is in operation. A DSA adds a new element to trace information prior to chaining an operation to another DSA. On being requested to perform an operation, a DSA checks, by examination of the trace information, that the operation has not formed a loop.

e) The **aliasDereferenced** component is a boolean value which is used to indicate whether or not one or more alias entries have so far been encountered and dereferenced during the course of distributed name resolution. Since alias entries in IN are just a means to provide an alternative name for an object and therefore should be dereferenced when needed, there is no need for this indicator.

f) The **aliasedRDNs** component indicates how many of the RDNs in the **targetObject** name have been generated from the **aliasedEntryName** attributes of one (or more) alias entries. The integer value is set whenever an alias entry is encountered and dereferenced. Since alias entries in IN are just a means to provide an alternative name for an object and therefore should be dereferenced when needed, there is no need for this indicator.

g) The **returnCrossRefs** component is a Boolean value which indicates whether or not knowledge references, used during the course of performing a distributed operation, are requested to be passed back to the initial DSA as cross-references, along with a result or referral. Since direct knowledge references are assumed, this parameter is deemed not applicable for IN CS-2.

h) The **referenceType** component indicates, to the DSA being asked to perform the operation, what type of knowledge was used to route the request to it. The DSA may therefore be able to detect errors in the knowledge held by the invoker. If such an error is detected, it shall be indicated by a **ServiceError** with the **invalidReference** problem. **ReferenceType** is described fully in 8.4.1.3.

i) The **info** component is used to convey DMD-specific (Directory Management Domain) information among DSAs which are involved in the processing of a common request. As the management protocols are not addressed in CS-2, this parameter is deemed to be not applicable.

j) The **timeLimit** component, if present, indicates the time by which the operation is to be completed. It is redundant with operation timers of TCAP and is therefore not needed.

k) The **SecurityParameters** component is specified in ITU-T Rec. X.511 | ISO/IEC 9594-3.

l) The **entryOnly** component is set to **TRUE** if the original operation was a search, with the subset argument set to **oneLevel** and an alias entry was encountered as an immediate subordinate of the **baseObject**. The DSA which successfully performs name resolution on the **targetObject** name shall perform object evaluation on only the named entry.

m) **AuthenticationLevel** is optionally supplied when it is required to indicate the manner in which authentication has been carried out between the SDFs. The **AuthenticationLevel** element is described in ITU-T Rec. X.501 | ISO/IEC 9594-2.

n) **UniqueIdentifier** is optionally supplied when it is required to confirm the originator name (the originator is the SDF forwarding the request). The **UniqueIdentifier** element is described in ITU-T Rec. X.501 | ISO/IEC 9594-2.

o) The **exclusions** component has significance only for Search operations; it indicates, if present, which subtrees of entries subordinate to the **targetObject** shall be excluded from the result of the Search operation.

p) The **excludeShadows** component has significance only for Search and List operations; it indicates that the search shall be applied to entries and not to entry copies. This optional component may be used by a DSA as one way to avoid the receipt of duplicate results. Since direct knowledge references are assumed, this parameter is deemed not applicable for CS-2.

q) The **nameResolveOnMaster** component only has significance during name resolution, and is only set if NSSRs (non-specific knowledge references) have been encountered. If set to **TRUE**, it signals that subsequent name resolution, i.e. matching the remaining RDNs from **nextRDNToBeResolved**, shall not employ entry copy information; subsequent resolution of each remaining RDN shall be done in the master DSA for the entry identified by that RDN. Since direct knowledge references are assumed, this parameter is deemed not applicable for IN CS-2.

8.4.1.2 Chaining Results

The **ChainingResults** are present in the result of each operation and provide feedback to the DSA which invoked the operation.

```
IN-ChainingResults ::= ChainingResults (
        WITH COMPONENTS {
                ...,
                info                    ABSENT,
                crossReferences         ABSENT })
```

The various components have the meanings as defined below:

a) The **info** component is used to convey DMD-specific information among DSAs which are involved in the processing of a common request. As the management protocols are not addressed in CS-2, this parameter is deemed to be not applicable.

b) The **crossReferences** component is not present in the **ChainingResults** unless the **returnCrossRefs** component of the corresponding request had the value **TRUE**. Since direct knowledge references are assumed, this parameter is deemed not applicable for IN CS-2.

c) The **SecurityParameters** component is specified in ITU-T Rec. X.511 | ISO/IEC 9594-3. Its absence is deemed equivalent to there being an empty set of security parameters.

d) The **alreadySearched** component, if present, indicates which subordinate RDNs immediately subordinate to the **targetObject** have been processed as a part of a chained Search operation and therefore shall be excluded in a subsequent sub-request.

8.4.1.3 Reference Type

A **ReferenceType** value indicates one of the various kinds of reference defined in ITU-T Rec. X.501 | ISO/IEC 9594-2.

```
IN-ReferenceType ::= ReferenceType (1|2|4|5|6|7|8)
```

Value (3)(cross-reference) is not applicable for IN CS-2 as direct references are assumed.

8.4.1.4 Access Point Information

There are three types of access points:

a) An **AccessPoint** value identifies a particular point at which access to the Directory, specifically to a DSA, can occur. The access point has a **Name**, that of the DSA concerned, and a **PresentationAddress**, to be used in SS7 signalling to that DSA.

```
IN-AccessPoint ::= AccessPoint (
        WITH COMPONENTS {
                ..,
                protocolInformation     ABSENT})
```

The **address** contains the network address of the DSA in the SS7.

b) A **MasterOrShadowAccessPoint** value identifies an access point to the Directory. The category, either **master** or **shadow**, of the access point is dependent upon whether it points to a naming context or commonly-useable replicated area.

```
IN-MasterOrShadowAccessPoint ::= MasterOrShadowAccessPoint (
        WITH COMPONENTS {
                ...,
                COMPONENTS OF IN-AccessPoint})
```

c) A **MasterAndShadowAccessPoints** value identifies a set of access points to the Directory, i.e. a set of related DSAs. These access points share the property that each refers to a DSA holding entry information from a common naming context (or a common set of naming contexts mastered in one DSA) when the value is a value of the **nonSpecificKnowledge** attribute. A **MasterAndShadowAccessPoints** value indicates the **category** of each **AccessPoint** value it contains. The access point of the master DSA of the naming context need not be included in the set.

```
IN-MasterAndShadowAccessPoints      ::= MasterOrShadowAccessPoint
```

An **AccessPointInformation** value identifies one or more access points to the Directory.

```
IN-AccessPointInformation ::= AccessPointInformation (
        WITH COMPONENTS {
                ...,
                COMPONENTS OF IN-MasterOrShadowAccessPoint })
```

8.4.1.5 Continuation Reference

A **ContinuationReference** describes how the performance of all or part of an operation can be continued at a different DSA or DSAs. It is typically returned as a referral when the DSA involved is unable or unwilling to propagate the request itself.

```
IN-ContinuationReference        ::=     ContinuationReference (
        WITH COMPONENTS {
                ...,
                aliasedRDNs             ABSENT,
                rdnsResolved            ABSENT,
                referenceType           (IN-ReferenceType),
                accessPoints            SET OF (IN-AccessPoint))
```

The various components have the meanings as defined below:

a) The **targetObject** name indicates the name which is proposed to be used in continuing the operation. This might be different from the **targetObject** name received on the incoming request if, for example, an alias has been dereferenced, or the base object in a search has been located.

b) The **aliasedRDNs** component indicates how many (if any) of the RDNs in the target object name have been produced by dereferencing an alias. Since alias entries in IN are just a means to provide an alternative name for an object and therefore should be dereferenced when needed, there is no need for this indicator.

c) The **operationProgress** indicates the amount of name resolution which has been achieved, and which will govern the further performance of the operation by the DSAs named, should the DSA or DUA receiving the **ContinuationReference** wish to follow it up.

d) The **rdnsResolved** component value (which need only be present if some of the RDNs in the name have not been the subject of full name resolution, but have been assumed to be correct from a cross-reference) indicates how many RDNs have actually been resolved, using internal references only. Since direct knowledge references are assumed, this parameter is deemed not applicable for IN CS-2.

e) The **referenceType** component indicates what type of knowledge was used in generating this continuation.

f) The **accessPoints** component indicates the access points which are to be contacted to achieve this continuation. Only where non-specific subordinate references are involved can there be more than one **AccessPointInformation** item.

g) The **entryOnly** component is set to **TRUE** if the original operation was a search, with the **subset** argument set to **oneLevel**, and an alias entry was encountered as an immediate subordinate of the **baseObject**. The DSA which successfully performs name resolution on the **targetObject** name, shall perform object evaluation on only the named entry. Since alias entries in IN are just a means to provide an alternative name for an object and therefore should be dereferenced when needed, there is no need for this indicator.

h) The **exclusions** component identifies a set of subordinate naming contexts that should not be explored by the receiving DSA.

i) The **returnToDUA** element is optionally supplied when the DSA creating the continuation reference wishes to indicate that it is unwilling to return information via an intermediate DSA (e.g. for security reasons), and wishes to indicate that information may be directly available via an operation over DAP between the originating DUA and the DSA. When **returnToDUA** is set to **TRUE**, **referenceType** may be set to **self.** This element may be used in IN for support of the shadowing agreement established between network operators (e.g. SDF_v to SDF_h Modify may fail based upon access control restrictions).

j) The **nameResolveOnMaster** element is optionally supplied when the DSA creating the continuation reference has encountered NSSRs. Since direct knowledge references are assumed, this parameter is deemed not applicable for IN CS-2.

8.4.2 DSA Bind

A **DSABind** operation is used to begin of a period of cooperation between two DSAs providing the Directory service.

```
dSABind OPERATION ::= in-DirectoryBind
```

IN CS-2 uses the in-DirectoryBind operation as specified in 7.3.2.1.

8.4.3 IN DSA Unbind

The **in-DSAUnbind** operation replaces the X.518 **dSAUnbind** operation to provide class 4 operation behaviour for unbind procedures.

```
in-DSAUnbind  OPERATION ::= inEmptyUnbind
```

8.4.4 Chained Operations

A DSA, having received an operation from a DUA, may elect to construct a chained form of that operation to propagate to another DSA. For IN CS-2 a DSA, having received a chained form of an operation, must either process the operation or if the originating DSA is in another network, chain it to another DSA within the same network as the receiving DSA.

The DSA invoking a chained form of an operation may optionally sign the argument of the operation; the DSA performing the operation, if so requested, may sign the result of the operation.

The chained form of an operation is specified using the parameterized type **IN-chained {}**.

```
IN-chained { OPERATION : operation } OPERATION       ::= {
        ARGUMENT         OPTIONALLY-PROTECTED { SET {
                chainedArgument             (IN-ChainingArguments),
                argument              [0]      operation.&ArgumentType },
                DIRQOP.&dspChainedOp-QOP@dirqop}
        RESULT      OPTIONALLY-PROTECTED { SET {
                IN-chainedResult            ABSENT,
                result              [0]     operation.&ResultType },
                DIRQOP.&dspChainedOp-QOP@dirqop }
        ERRORS      { operation.&Errors EXCEPT (referral | dsaReferral) }
        CODE        operation.&code }
```

a) **IN-chainedArgument**. This is a value of **ChainingArguments** which contains that information, over and above the original DUA-supplied argument, which is needed in order for the performing DSA to carry out the operation.

b) **argument**. This is a value **operation.&Argument** and consists of the original DUA-supplied argument.

Should the request succeed, the result of the derived operation has the components:

a) **IN-chainedResult**. This is a value of **IN-ChainingResults** which contains that information, over and above that to be supplied to the originating DUA, which may be needed by the previous DSAs in a chain. For IN CS-2, it is assumed that chains are not greater than length one, therefore the need of this parameter is not needed.

b) **result**. This is a value **operation.&Result** and consists of the result which is being returned by the performer of this operation, and which is intended to be passed back in the result to the originating DUA. This information is as specified in the appropriate clause of ITU-T Rec. X.511 | ISO/IEC 9594-3.

Should the request fail, one of the errors of the set **operation.&Errors** will be returned, except that **dsaReferral** is returned instead of **referral**.

8.4.5 Chained Errors

The **dsaReferral** error is generated by a DSA when, for whatever reason, it does not wish to continue performing an operation by chaining the operation to another DSA. For IN CS-2, DSAs may not chain operations incoming from another DSA unless the DSA is in another network.

```
IN-dsaReferral  ERROR ::= dsaReferral (
        WITH COMPONENTS {
                ...,
                reference           (IN-ContinuationReference),
                contextPrefix               ABSENT})
```

The various parameters have the meanings as described below:

a) The **IN-ContinuationReference** contains the information needed by the invoker to propagate an appropriate further request, perhaps to another DSA.

b) If the **returnCrossRefs** component of the ChainingArguments for this operation had the value **TRUE**, and the referral is being based upon a subordinate or cross-reference, then the **contextPrefix** parameter may optionally be included. The administrative authority of any DSA will decide which knowledge references, if any, can be returned in this manner (the others, for example, may be confidential to that DSA). Since direct knowledge references are assumed for IN CS-2, this parameter is not applicable.

8.5 Protocol overview

8.5.1 ROS-Objects and contracts

The interactions between DSAs generally required to provide the Directory Abstract Service in the presence of a distributed DIB are defined as a **indspContract**. A DSA that participates in this contract is defined as a ROS-object of class **dsp-dsa**. The contract is referred in this specification as the DSA Abstract Service.

```
dsp-dsa  ROS-OBJECT-CLASS ::= {
        BOTH        { indspContract}
        ID          id-rosObject-dspDSA}
```

The Shadow Abstract Service specifies the shadowing of information between a shadow supplier and a shadow consumer DSA. This service is manifested in two forms and therefore is defined as two distinct contracts. They are specified as a ROS-based information objects in 8.5.2.

The **shadowConsumerContract** expresses the form of the service in which the shadow consumer, a ROS-object of class **initiating-consumer-dsa**, initiates the contract. A ROS-object of class **responding-supplier-dsa** responds in this contract.

```
initiating-consumer-dsa  ROS-OBJECT-CLASS ::= {
        INITIATES  {shadowConsumerContract}
        ID          id-rosObject-initiatingConsumerDSA }

responding-supplier-dsa  ROS-OBJECT-CLASS ::= {
        RESPONDS {shadowConsumerContract}
        ID          id-rosObject-respondingSupplierDSA }
```

The **shadowSupplierContract** expresses the form of the service in which the shadow supplier, a ROS-object of class **initiating-supplier-dsa**, initiates the contract. A ROS-object of class **responding-consumer-dsa**, responds in this contract.

```
initiating-supplier-dsa  ROS-OBJECT-CLASS ::= {
        INITIATES  {shadowSupplierContract}
        ID          id-rosObject-initiatingSupplierDSA }

responding-consumer-dsa  ROS-OBJECT-CLASS ::= {
        RESPONDS {shadowSupplierContract}
        ID          id-rosObject-respondingConsumerDSA }
```

8.5.2 DSP contract and packages

The **indspContract** is defined as an information object of class CONTRACT.

```
indspContract  CONTRACT ::= {
        CONNECTION                  dspConnectionPackage
        INITIATOR CONSUMER OF       { inchainedModifyPackage | inchainedSearchPackage |
                                       chainedExecutePackage }
        ID                          id-contract-indsp}
```

When a pair of DSAs from different open systems interact, this association contract is realised as an SS7 application layer protocol, referred to as the IN Directory System Protocol (DSP). The definition of this protocol in terms of an SS7 application context is provided in 8.6.

The **indspContract** is composed of a connection package, **indspConnectionPackage** and three operation packages, **inchainedModifyPackage**, **inchainedSearchPackage** and **chainedExecutePackage**.

The connection package, **indspConnectionPackage**, is defined as an information object of class CONNECTION-PACKAGE. It is identical to the connection package, **indapConnectionPackage**.

```
dspConnectionPackage  CONNECTION-PACKAGE ::= {
        BIND        dSABind
        UNBIND      in-DSAUnbind
        ID          id-package-dspConnection}
```

The operation packages **inchainedModifyPackage** and **inchainedSearchPackage** are defined as information objects of class OPERATION-PACKAGE. The operations of these packages are defined in Recommendation X.518.

```
inchainedModifyPackage  OPERATION-PACKAGE ::= {
        CONSUMER INVOKES   {chainedAddEntry | chainedRemoveEntry | chainedModifyEntry}
        ID                 id-package-inchainedModify}

inchainedSearchPackage OPERATION-PACKAGE ::= {
        CONSUMER INVOKES   {chainedSearch}
        ID                 id-package-inchainedSearch}
```

The operation packages **chainedExecutePackage** is defined an information objects of class OPERATION-PACKAGE.

```
chainedExecutePackage  OPERATION-PACKAGE ::= {
        CONSUMER INVOKES   { chainedExecute }
        ID                 id-package-inchainedExecute}
```

In the **indspContract** either DSA may assume the role of initiator and invoke the operations of the contract.

8.5.3 DISP contract and packages

The **shadowConsumerContract** and **shadowSupplierContract** are defined as information objects of class CONTRACT.

```
shadowConsumerContract  CONTRACT ::= {
        CONNECTION                   dispConnectionPackage
        INITIATOR CONSUMER OF        {shadowConsumerPackage}
        ID                           id-contract-shadowConsumer}

shadowSupplierContract  CONTRACT ::= {
        CONNECTION                   dispConnectionPackage
        RESPONDER CONSUMER OF        {shadowSupplierPackage}
        ID                           id-contract-shadowSupplier}
```

The SS7 realisation of the two forms of Shadow Abstract Service, referred to as the IN Directory Information Shadowing Protocol (DISP) are defined in terms of several SS7 application contexts provided in 8.6.

The **shadowConsumerContract** and **shadowSupplierContract** are composed of a common connection package, **dispConnectionPackage** and one operation package, either **ShadowConsumerPackage** in the first case or **shadowSupplierPackage** in the second.

The connection package, **dispConnectionPackage**, is defined as an information object of class CONNECTION-PACKAGE. It is identical to the connection package, **dapConnectionPackage**.

```
dispConnectionPackage  CONNECTION-PACKAGE ::= {
        BIND        dSAShadowBind
        UNBIND      in-DSAShadowUnbind
        ID          id-package-dispConnection}
```

The operation packages **shadowConsumerPackage** and **shadowSupplierPackage** are defined as information objects of class OPERATION-PACKAGE. The operations of these packages are defined in Recommendation X.525.

```
shadowConsumerPackage  OPERATION-PACKAGE ::= {
        CONSUMER INVOKES    {requestShadowUpdate}
        SUPPLIER INVOKES    {updateShadow}
        ID                  id-package-shadowConsumer}

shadowSupplierPackage  OPERATION-PACKAGE ::= {
        SUPPLIER INVOKES    {coordinateShadowUpdate | updateShadow}
        ID                  id-package-shadowSupplier}
```

Since the shadow consumer is the initiator of the **ShadowConsumerContract**, it assumes the role of consumer of the **shadowConsumerPackage**. This means that the shadow consumer invokes the **requestShadowUpdate** operation and that the shadow supplier invokes the **updateShadow** operation.

Since the shadow supplier is the initiator of the **shadowSupplierContract**, it assumes the role of supplier of the **shadowSupplierPackage**. This means that the shadow supplier invokes the operations of the contract.

8.6 Protocol abstract syntax

8.6.1 DSP abstract syntax

The Directory ASEs that realise the operation packages specified in 8.5.2 share a single abstract syntax, **indirectorySystemAbstractSyntax**. This is specified as an information object of the class ABSTRACT-SYNTAX.

```
inDirectorySystemAbstractSyntax  ABSTRACT-SYNTAX ::= {
        BasicDSP-PDUs
        IDENTIFIED BY       id-as-indirectorySystemAS}

BasicDSP-PDUs ::= TCMessage {{DSP-Invokable},{DSP-Returnable}}

DSP-Invokable OPERATION ::= {chainedAddEntry | chainedRemoveEntry | chainedModifyEntry |
                              chainedSearch | chainedExecute }

DSP-Returnable OPERATION ::= {chainedAddEntry | chainedRemoveEntry | chainedModifyEntry |
                              chainedSearch | chainedExecute }
```

The realisation of the connection package specified in 8.5.2 uses a separate abstract syntax, **indirectoryDSABindingAbstractSyntax**. This is specified as an information object of the class ABSTRACT-SYNTAX.

```
inDirectoryDSABindingAbstractSyntax  ABSTRACT-SYNTAX ::= {
        DSABinding-PDUs
        IDENTIFIED BY   id-as-indirectoryDSABindingAS}

DSABinding-PDUs ::= CHOICE {
        bind            Bind {dSABind},
        unbind          Unbind {in-DSAUnbind}}
```

8.6.2 DISP Abstract Syntax

The Directory ASEs that realise the operation packages specified in 8.5.3 share the abstract syntax **inDirectoryShadowAbstractSyntax**. This abstract syntax is specified as an information object of the class ABSTRACT-SYNTAX.

```
inDirectoryShadowAbstractSyntax ABSTRACT-SYNTAX ::= {
        BasicDISP-PDUs
        IDENTIFIED BY       id-as-indirectoryShadowAS}
```

```
BasicDISP-PDUs ::= TCMessage {{DISP-Invokable},{DISP-Returnable}}

DISP-Invokable OPERATION ::={requestShadowUpdate | updateShadow | coordinateShadowUpdate}

DISP-Returnable OPERATION ::={requestShadowUpdate | updateShadow | coordinateShadowUpdate}
```

The realisation of the connection package specified above uses a separate abstract syntax, **inDirectoryDSAShadowBindingAbstractSyntax**. This is specified as an information object of class ABSTRACT-SYNTAX.

```
inDirectoryDSAShadowBindingAbstractSyntax  ABSTRACT-SYNTAX ::= {
        DISPBinding-PDUs
        IDENTIFIED BY   id-as-indsaShadowBindingAS}

DISPBinding-PDUs ::= CHOICE {
        bind                    Bind {dSAShadowBind},
        unbind                  Unbind {in-DSAShadowUnbind}}
```

8.6.3 Directory System Application Context

The **indspContract** is realised as the **inDirectorySystemAC**. This application context is specified as an information object of the class APPLICATION-CONTEXT.

```
inDirectorySystemAC APPLICATION-CONTEXT ::=        {
        CONTRACT                        dspContract
        DIALOGUE MODE                   structured
        TERMINATION                     basic
        ABSTRACT SYNTAXES               {dialogue-abstract-syntax |
                                                inDirectorySystemAbstractSyntax |
                                                inDirectoryDSABindingAbstractSyntax}
        APPLICATION CONTEXT NAME id-ac-indirectorySystemAC}
```

If 3-way authentication is required, then the **indspContract** is realised as the **inDirectorySystemWith3seAC**. This application context is specified as an information object of the class APPLICATION-CONTEXT.

```
inDirectorySystemWith3seAC APPLICATION-CONTEXT ::=     {
        CONTRACT                        dspContract
        DIALOGUE MODE                   structured
        TERMINATION                             basic
        ADDITIONAL ASE                  {id-se-threewayse}
        ABSTRACT SYNTAXES               {dialogue-abstract-syntax |
                                                inDirectorySystemAbstractSyntax |
                                        inDirectoryDSABindingAbstractSyntax |
                                        inSESEAbstractSyntax }
        APPLICATION CONTEXT NAME        id-ac-indirectorySystemWith3seAC}
```

8.6.4 Directory Shadow Application Context

The **inshadowSupplierContract** is realised as the **inshadowSupplierInitiatedAC**. This application context is specified as an information object of the class APPLICATION-CONTEXT.

```
inshadowSupplierInitiatedAC APPLICATION-CONTEXT ::= {
        CONTRACT                        shadowSupplierContract
        DIALOGUE MODE                   structured
        TERMINATION                     basic
        ABSTRACT SYNTAXES               {dialogue-abstract-syntax |
                                                inDirectoryShadowAbstractSyntax |
                                                inDirectoryDSAShadowBindingAbstractSyntax}
        APPLICATION CONTEXT NAME                        id-ac-inShadowSupplierInitiatedAC}
```

If 3-way authentication is required, then the **inshadowSupplierContract** is realised as the **inshadowSupplierInitiatedWith3seAC**. This application context is specified as an information object of the class APPLICATION-CONTEXT.

```
inshadowSupplierInitiatedWith3seAC APPLICATION-CONTEXT ::= {
        CONTRACT                        shadowSupplierContract
        DIALOGUE MODE                   structured
        TERMINATION                     basic
        ADDITIONAL ASE          {id-se-threewayse}
        ABSTRACT SYNTAXES               {dialogue-abstract-syntax |
                                        inDirectoryShadowAbstractSyntax |
                                        inDirectoryDSAShadowBindingAbstractSyntax |
                                        inSESEAbstractSyntax }
APPLICATION CONTEXT NAME        id-ac-inShadowSupplierInitiatedWith3seAC}
```

The **inshadowConsumerContract** is realised as the **inshadowConsumerInitiatedAC**. This application context is specified as an information object of the class APPLICATION-CONTEXT.

```
inshadowConsumerInitiatedAC APPLICATION-CONTEXT ::= {
        CONTRACT                        shadowConsumerContract
        DIALOGUE MODE                   structured
        TERMINATION                     basic
        ABSTRACT SYNTAXES               {dialogue-abstract-syntax |
                                                inDirectoryShadowAbstractSyntax |
                                                inDirectoryDSAShadowBindingAbstractSyntax}
APPLICATION CONTEXT NAME                        id-ac-inShadowConsumerInitiatedAC}
```

If 3-way authentication is required, then the **inshadowConsumerContract** is realised as the **inshadowConsumerInitiatedWith3seAC**. This application context is specified as an information object of the class APPLICATION-CONTEXT.

```
inshadowConsumerInitiatedWith3seAC APPLICATION-CONTEXT ::= {
        CONTRACT                        shadowConsumerContract
        DIALOGUE MODE                   structured
        TERMINATION                     basic
        ADDITIONAL ASE          {id-se-threewayse}
        ABSTRACT SYNTAXES               {dialogue-abstract-syntax |
                                        inDirectoryShadowAbstractSyntax |
                                        inDirectoryDSAShadowBindingAbstractSyntax |
                                        inSESEAbstractSyntax }
APPLICATION CONTEXT NAME        id-ac-inShadowConsumerInitiatedWith3seAC}
```

8.6.5 Versions and the rules for extensibility

The Directory may be distributed and more than two Directory Application Entities may interoperate to service a request. The Directory AEs may be implemented conforming to different editions of the Directory specification of the Directory service which may or may not be represented by different protocol version numbers. The version number is negotiated to the highest common version number between two directly binding Directory AEs.

8.6.5.1 Version negotiation

When accepting an association, i.e. binding, utilizing the DSP or DISP, the version negotiated shall only affect the point-to-point aspects of the protocol exchanged between the initiating DSA and the responding DSA to which it is connected. Subsequent requests or responses on the dialogue shall be constrained by the version negotiated.

NOTE – There are no point-to-point aspects of the DSP or DISP that are currently indicated by different protocol versions.

8.6.5.2 Initiating DSA side

8.6.5.2.1 Request and response processing at the initiating DSA side

The initiating DSA may initiate requests using the highest edition of the specification of that request it supports. If one or more elements of the request are critical, it shall indicate the extension number(s) in the critical Extensions parameter.

NOTE 1 – If the information the extension replaced in a CHOICE, ENUMERATED or INTEGER (used as ENUMERATED) type would be essential for proper operation in a responding DSA implemented according to an earlier edition of the specification, it is recommended that the extension be marked critical.

When processing a response, a initiating DSA shall:

a) ignore all unknown bit name assignments within a bit string; and

b) ignore all unknown named numbers in an ENUMERATED type or INTEGER type that is being used in the enumerated style, provided the number occurs as an optional element of a SET or SEQUENCE; and

c) ignore all unknown elements in SETs, at the end of SEQUENCEs, or in CHOICEs where the CHOICE is itself an optional element of a SET or SEQUENCE;

NOTE 2 – Implementations may as a local option ignore certain additional elements in a Directory PDU. In particular, some unknown named numbers and unknown CHOICEs in mandatory elements of SETs and SEQUENCEs can be ignored without invalidating the operation. The identification of such elements is for further study.

d) not consider the receipt of unknown attribute types and attribute values as a protocol violation; and

e) optionally report the unknown attribute types and attribute values to the user.

8.6.5.2.2 Extensibility rules for error handling at the initiating DSA side

When processing a known error type with unknown indicated problems and parameters, a initiating DSA shall:

a) not consider the receipt of unknown indicated problems and parameters as a protocol violation (i.e. it shall not issue a TC-U-REJECT or abort the dialogue); and

b) optionally report the additional error information to the user.

When processing an unknown error type, a initiating DSA shall:

a) not consider the receipt of unknown error type as a protocol violation (i.e. it shall not issue a TC-U-REJECT or abort the application association); and

b) optionally report the error to the user.

8.6.5.3 Request processing at the responding DSA side

If any responding DSA performing an operation detects an element **criticalExtensions** whose semantic is unkown, it shall return an **unavailableCriticalExtension** indication as a **serviceError**.

NOTE 1 – If a **criticalExtensions** string with one or more zero values is received, this indicates either that the extensions corresponding to the values are not present or are not critical. The presence of a zero value in a **criticalExtensions** string shall not be inferred as either the presence or absence of the corresponding extension in the APDU.

Otherwise, when processing a request from a initiating DSA, a responding DSA shall:

a) ignore all unknown bit name assignments within a bit string; and

b) ignore all unknown named numbers in an ENUMERATED type or INTEGER type that is being used in the enumerated style, provided the number occurs as an optional element of a SET or SEQUENCE; and

c) ignore all unknown elements in SETs, at the end of SEQUENCEs, or in CHOICEs where the CHOICE is itself an optional element of a SET or SEQUENCE.

NOTE 2 – Implementations may as a local option ignore certain additional elements in a Directory PDU. In particular, some unknown named numbers and unknown CHOICEs in mandatory elements of SETs and SEQUENCEs can be ignored without invalidating the operation. The identification of such elements is for further study.

8.7 Conformance

For the conformance of SDFs, the following statements should be added to the list of already existing statements.

8.7.1 Conformance by SDFs

8.7.1.1 Statement requirements

The following shall be stated:

a) the application-context for which conformance is claimed. The present version of this Recommendation requires conformance to the **inDirectorySytemAC** application-context;

   NOTE – An application-context shall not be divided except as stated herein; in particular, conformance shall not be claimed to particular operations.

b) if conformance is claimed to the **inDirectorySystemAC** application-context, whether or not the chained mode of operation is supported, as defined in Recommendation X.518;

c) the security-level(s) for which conformance is claimed (none, simple, strong);

d) the attribute types for which conformance is claimed and whether for attributes based on the syntax **DirectoryString,** conformance is claimed for the **UNIVERSAL STRING** choice;

e) the object classes, for which conformance is claimed;

f) whether conformance is claimed for collective attributes as defined in 8.8/X.501 and 7.6, 7.8.2 and 9.2.2 of Recommendation X.511;

g) whether conformance is claimed for hierarchical attributes as defined in 7.6, 7.8.2 and 9.2.2 of Recommendation X.511;

h) the operational attribute types defined in Recommendation X.501 and any other operational attribute types for which conformance is claimed;

i) whether conformance is claimed for return of alias names as described in 7.7.1/X.511;

j) whether conformance is claimed for indicating that returned entry information is complete, as described in section 7.7.6/X.511;

k) whether conformance is claimed for modifying the object class attribute to add and/or remove values identifying auxiliary object classes, as described in 11.3.2/X.511;

l) whether conformance is claimed to Basic Access Control;

m) whether conformance is claimed to Simplified Access Control;

n) the name bindings for which conformance is claimed;

o) whether the SDF is capable of administering collective attributes, as defined in Recommendation X.501;

p) whether conformance is claimed for attribute contexts.

8.7.1.2 Static requirements

An SDF shall:

a) have the capability of supporting the application-contexts for which conformance is claimed as defined by their abstract syntax in 8.6;

b) have the capability of supporting the information framework defined by its abstract syntax in Recommendation X.501;

c) conform to the minimal knowledge requirements defined in Recommendation X.518;

d) have the capability of supporting the attribute types for which conformance is claimed; as defined by their abstract syntaxes;

e) have the capability of supporting the object classes for which conformance is claimed, as defined by their abstract syntaxes;

f) conform to the extensions for which conformance was claimed in 8.7.1.1;

g) if conformance is claimed for collective attributes, have the capability of performing the related procedures defined in 7.6, 7.8.2 and 9.2.2 of Recommendation X.511;

h) if conformance is claimed for hierarchical attributes, have the capability of performing the related procedures defined in 7.6, 7.8.2 and 9.2.2 of Recommendation X.511;

i) have the capability of supporting the operational attribute types for which conformance is claimed;

j) if conformance is claimed to Basic Access Control, have the capability of holding ACI items that conform to the definitions of Basic Access Control;

k) if conformance is claimed to Simplified Access Control, have the capability of holding ACI items that conform to the definitions of Simplified Access Control.

8.7.1.3 Dynamic requirements

An SDF shall:

a) conform to the mapping onto used services defined in 18.1.7;

b) conform to the procedures for distributed operations of the Directory related to referrals, as defined in Recommendation X.518;

c) if conformance to the **directorySystemAC** application-context, conform to the referral mode of interaction as defined in Recommendation X.518;

d) if conformance is claimed for the chained mode of interaction, conformance to the chained mode of interaction as defined in Recommendation X.518;

NOTE – Only in this case, it is necessary for a DSA to be capable of invoking operations of the **directorySystemAC**.

e) conform to the rules of extensibility procedures defined in 7.5.5;

f) if conformance is claimed to Basic Access Control, have the capability of protecting information within the SDF in accordance with the procedures of Basic Access Control;

g) if conformance is claimed to Simplified Access Control, have the capability of protecting information within the SDF in accordance with the procedures of Simplified Access Control.

8.7.2 Conformance by a shadow supplier

A SDF implementation claiming conformance to this Directory Specification in the role of shadow supplier shall satisfy the requirements specified below.

8.7.2.1 Statement requirements

The following shall be stated:

a) the application-context(s) for which conformance is claimed as a shadow supplier: **inShadowSupplierInitiatedAC** and **inShadowConsumerInitiatedAC**;

b) the security-level(s) for which conformance is claimed (none, simple, strong);

c) to which degree the **UnitOfReplication** is supported. Specifically, which (if any) of the following optional features are supported:

   – entry filtering on **ObjectClass**;

   – selection/Exclusion of attributes via **AttributeSelection**;

   – the inclusion of subordinate knowledge in the replicated area;

   – the inclusion of extended knowledge in addition to subordinate knowledge.

8.7.2.2 Static requirements

A SDF shall:

a) have the capability of supporting the application-context(s) for which conformance is claimed as defined in their abstract syntax above;

b) provide support for modifyTimestamp and createTimestamp operational attributes.

8.7.2.3 Dynamic requirements

A SDF shall:

a) conform to the mapping onto used services defined above;

b) conform to the procedures of ITU-T Rec. X.525 | ISO/IEC 9594-9 as they relate to the DISP.

8.7.3 Conformance by a shadow consumer

A SDF implementation claiming conformance to this Directory Specification as a shadow consumer shall satisfy the requirements specified below:

8.7.3.1 Statement requirements

The following shall be stated:

a) the application-context(s) for which conformance is claimed as a shadow supplier: **inShadowSupplierInitiatedAC** and **shadowConsumerInitiatedAC**;

b) the security-level(s) for which conformance is claimed (none, simple, strong);

c) whether the SDF supports shadowing of overlapping units of replication.

8.7.3.2 Static requirements

A SDF shall:

a) have the capability of supporting the application-context(s) for which conformance is claimed as defined in their abstract syntax in 8.6;

b) provide support for modifyTimestamp and createTimestamp operational attributes if overlapping units of replication is supported;

c) provide support for the copyShallDo service control.

8.7.3.3 Dynamic requirements

A SDF shall:

a) conform to the mapping onto used services defined in 18.1.7;

b) conform to the procedures of Recommendation X.525 as they relate to the DISP.

8.8 ASN.1 modules for the SDF-SDF interface

The following set of ASN.1 modules define the SDF-SDF interface for IN CS-2. They contain all the modifications to the Directory specifications as required for the support of Intelligent Networks.

The modules also contain the definitions which are impacted by these modifications because they make use of a modified type.

8.8.1 IN-CS2-SDF-SDF-Protocol Module

This subclause includes all of the ASN.1 type and value definitions contained in this Directory Specification, in the form of the ASN.1 module, "IN-CS2-SDF-SDF-Protocol ".

```
IN-CS2-SDF-SDF-Protocol
      {  ccitt recommendation q 1228 module(0) in-cs2-sdf-sdf-Protocol(18) version1(0) }

DEFINITIONS ::=

BEGIN
-- EXPORTS All --
-- The types and values defined in this module are exported for use in the other ASN.1 modules contained
-- within the Directory Specifications, and for the use of other applications which will use them to access
-- Directory services. Other applications may use them for their own purposes, but this will not constrain
-- extensions and modifications needed to maintain or improve the Directory service.

IMPORTS
      distributedOperations, directoryShadowAbstractService, dsp , protocolObjectIdentifiers
            FROM  UsefulDefinitions ds-UsefulDefinitions

      ROS-OBJECT-CLASS, CONTRACT, OPERATION-PACKAGE, CONNECTION-PACKAGE,
      Code, OPERATION
            FROM Remote-Operations-Information-Objects ros-InformationObjects

      Bind{}, Unbind{}
            FROM Remote-Operations-Generic-ROS-PDUs ros-genericPDUs

      TCMessage {}
            FROM TCAPMessages tc-Messages

      APPLICATION-CONTEXT, dialogue-abstract-syntax
            FROM TC-Notation-Extensions tc-NotationExtensions

      dSABind,
      chainedSearch, chainedAddEntry, chainedRemoveEntry, chainedModifyEntry, chained{}
            FROM DistributedOperations distributedOperations

      dSAShadowBind,
      coordinateShadowUpdate, updateShadow, requestShadowUpdate
            FROM DirectoryShadowAbstractService directoryShadowAbstractService

      execute
            FROM IN-CS2-SCF-SDF-Operations scf-sdf-Operations
```

```
inEmptyUnbind
        FROM IN-CS2-classes {itu-t recommendation q 1228 modules(0) in-cs2-classes(4) version1(0)}

id-rosObject-dspDSA, id-rosObject-initiatingConsumerDSA, id-rosObject-respondingSupplierDSA,
id-rosObject-respondingConsumerDSA, id-rosObject-initiatingSupplierDSA,
id-contract-indsp, id-contract-shadowConsumer, id-contract-shadowSupplier,
id-package-dspConnection, id-package-inchainedModify, id-package-inchainedSearch, id-package-
chainedExecute,
id-package-dispConnection, id-package-shadowConsumer, id-package-shadowSupplier,
id-as-indirectorySystemAS, id-as-indirectoryDSABindingAS, id-as-indirectoryShadowAS,
id-as-indsaShadowBindingAS,
id-ac-indirectorySystemAC, id-ac-inShadowSupplierInitiatedAC, id-ac-inShadowConsumerInitiatedAC,
id-ac-inShadowSupplierInitiatedWith3seAC,id-ac-inShadowConsumerInitiatedWith3seAC,
id-ac-indirectorySystemWith3seAC,
ds-UsefulDefinitions, ros-InformationObjects, ros-genericPDUs, tc-Messages,
tc-NotationExtensions, scf-sdf-Operations, scf-sdf-Protocol
        FROM IN-CS2-object-identifiers
                { itu-t recommendation q 1228 module(0) in-cs2-object-identifiers(17) version1(0) }
inSESEAbstractSyntax
        FROM IN-CS2-SCF-SDF-Protocol scf-sdf-Protocol
id-se-threewayse
        FROM ProtocolObjectIdentifiers protocolObjectIdentifiers

dspContract
        FROM DirectorySystemProtocol dsp

;
dsp-dsa  ROS-OBJECT-CLASS ::= {
        BOTH                {indspContract}
        ID                  id-rosObject-dspDSA}

initiating-consumer-dsa  ROS-OBJECT-CLASS ::= {
        INITIATES           {shadowConsumerContract}
        ID                  id-rosObject-initiatingConsumerDSA }

responding-supplier-dsa  ROS-OBJECT-CLASS ::= {
        RESPONDS            {shadowConsumerContract}
        ID                  id-rosObject-respondingSupplierDSA }

initiating-supplier-dsa  ROS-OBJECT-CLASS ::= {
        INITIATES           {shadowSupplierContract}
        ID                  id-rosObject-initiatingSupplierDSA }

responding-consumer-dsa  ROS-OBJECT-CLASS ::= {
        RESPONDS            {shadowSupplierContract}
        ID                  id-rosObject-respondingConsumerDSA }

indspContract  CONTRACT ::= {
        CONNECTION                  dspConnectionPackage
        INITIATOR CONSUMER OF       { inchainedModifyPackage | inchainedSearchPackage |
                                      chainedExecutePackage }
        ID                          id-contract-indsp}

dspConnectionPackage  CONNECTION-PACKAGE ::= {
        BIND        dSABind
        UNBIND      in-DSAUnbind
        ID          id-package-dspConnection}

in-DSAUnbind        OPERATION  ::=  inEmptyUnbind
```

```
inchainedModifyPackage  OPERATION-PACKAGE ::= {
        CONSUMER INVOKES   {chainedAddEntry | chainedRemoveEntry | chainedModifyEntry}
        ID                 id-package-inchainedModify}

inchainedSearchPackage OPERATION-PACKAGE ::= {
        CONSUMER INVOKES   {chainedSearch}
        ID                 id-package-inchainedSearch}

chainedExecutePackage OPERATION-PACKAGE ::= {
        CONSUMER INVOKES   { chainedExecute }
        ID                 id-package-chainedExecute}

chainedExecute      OPERATION  ::= chained { execute }

shadowConsumerContract  CONTRACT ::= {
        CONNECTION                 dispConnectionPackage
        INITIATOR CONSUMER OF      {shadowConsumerPackage}
        ID                         id-contract-shadowConsumer}

shadowSupplierContract  CONTRACT ::= {
        CONNECTION                 dispConnectionPackage
        RESPONDER CONSUMER OF {shadowSupplierPackage}
        ID                         id-contract-shadowSupplier}

dispConnectionPackage  CONNECTION-PACKAGE ::= {
        BIND        dSAShadowBind
        UNBIND      in-DSAShadowUnbind
        ID          id-package-dispConnection}

in-DSAShadowUnbind      OPERATION  ::=   inEmptyUnbind

shadowConsumerPackage  OPERATION-PACKAGE ::= {
        CONSUMER INVOKES   {requestShadowUpdate}
        SUPPLIER INVOKES   {updateShadow}
        ID                 id-package-shadowConsumer}

shadowSupplierPackage  OPERATION-PACKAGE ::= {
        SUPPLIER INVOKES   {coordinateShadowUpdate | updateShadow}
        ID                 id-package-shadowSupplier}

inDirectorySystemAbstractSyntax  ABSTRACT-SYNTAX ::= {
        BasicDSP-PDUs
        IDENTIFIED BY      id-as-indirectorySystemAS}

BasicDSP-PDUs ::= TCMessage {{DSP-Invokable},{DSP-Returnable}}

DSP-Invokable OPERATION ::= {chainedAddEntry | chainedRemoveEntry | chainedModifyEntry |
                           chainedSearch | chainedExecute }

DSP-Returnable OPERATION ::= {chainedAddEntry | chainedRemoveEntry | chainedModifyEntry |
                           chainedSearch | chainedExecute }

inDirectoryDSABindingAbstractSyntax  ABSTRACT-SYNTAX ::= {
        DSABinding-PDUs
        IDENTIFIED BY      id-as-indirectoryDSABindingAS}

DSABinding-PDUs ::= CHOICE {
        bind        Bind {dSABind},
        unbind      Unbind {in-DSAUnbind}}
```

```
inDirectoryShadowAbstractSyntax ABSTRACT-SYNTAX ::= {
        BasicDISP-PDUs
        IDENTIFIED BY           id-as-indirectoryShadowAS}

BasicDISP-PDUs ::= TCMessage {{DISP-Invokable},{DISP-Returnable}}

DISP-Invokable OPERATION ::={requestShadowUpdate | updateShadow | coordinateShadowUpdate}

DISP-Returnable OPERATION ::={requestShadowUpdate | updateShadow | coordinateShadowUpdate}

inDirectoryDSAShadowBindingAbstractSyntax  ABSTRACT-SYNTAX ::= {
        DISPBinding-PDUs
        IDENTIFIED BY           id-as-indsaShadowBindingAS}

DISPBinding-PDUs ::= CHOICE {
        bind                    Bind {dSAShadowBind},
        unbind                  Unbind {in-DSAShadowUnbind}}

inDirectorySystemAC APPLICATION-CONTEXT ::=       {
        CONTRACT                indspContract
        DIALOGUE MODE           structured
        TERMINATION             basic
        ABSTRACT SYNTAXES {dialogue-abstract-syntax |
                                        inDirectorySystemAbstractSyntax |
                                        inDirectoryDSABindingAbstractSyntax}
APPLICATION CONTEXT NAME id-ac-indirectorySystemAC}

inDirectorySystemWith3seAC APPLICATION-CONTEXT ::=        {
        CONTRACT                dspContract
        DIALOGUE MODE           structured
        TERMINATION                     basic
        ADDITIONAL ASE          {id-se-threewayse}
        ABSTRACT SYNTAXES {dialogue-abstract-syntax |
                                        inDirectorySystemAbstractSyntax |
                                inDirectoryDSABindingAbstractSyntax |
                                inSESEAbstractSyntax }
APPLICATION CONTEXT NAME id-ac-indirectorySystemWith3seAC}

inshadowSupplierInitiatedAC APPLICATION-CONTEXT ::= {
        CONTRACT                        shadowSupplierContract
        DIALOGUE MODE                   structured
        TERMINATION                     basic
        ABSTRACT SYNTAXES               {dialogue-abstract-syntax |
                                                inDirectoryShadowAbstractSyntax |
                                                inDirectoryDSAShadowBindingAbstractSyntax}
        APPLICATION CONTEXT NAME                id-ac-inShadowSupplierInitiatedAC}

inshadowSupplierInitiatedWith3seAC APPLICATION-CONTEXT ::= {
        CONTRACT                        shadowSupplierContract
        DIALOGUE MODE                   structured
        TERMINATION                     basic
        ADDITIONAL ASE                  {id-se-threewayse}
        ABSTRACT SYNTAXES               {dialogue-abstract-syntax |
                                        inDirectoryShadowAbstractSyntax |
                                        inDirectoryDSAShadowBindingAbstractSyntax |
                                        inSESEAbstractSyntax }
        APPLICATION CONTEXT NAME        id-ac-inShadowSupplierInitiatedWith3seAC}
        inshadowConsumerInitiatedAC APPLICATION-CONTEXT ::= {
                CONTRACT                shadowConsumerContract
                DIALOGUE MODE           structured
                TERMINATION             basic
                ABSTRACT SYNTAXES {dialogue-abstract-syntax |
                                                inDirectoryShadowAbstractSyntax |
                                                inDirectoryDSAShadowBindingAbstractSyntax}
                APPLICATION CONTEXT NAME id-ac-inShadowConsumerInitiatedAC}

        inshadowConsumerInitiatedWith3seAC APPLICATION-CONTEXT ::= {
                CONTRACT                shadowConsumerContract
                DIALOGUE MODE           structured
                TERMINATION             basic
                ADDITIONAL ASE          {id-se-threewayse}
                ABSTRACT SYNTAXES {dialogue-abstract-syntax |
                                                inDirectoryShadowAbstractSyntax |
                                                inDirectoryDSAShadowBindingAbstractSyntax |
                                        inSESEAbstractSyntax }
                APPLICATION CONTEXT NAME id-ac-inShadowConsumerInitiatedWith3seAC}
END
```

9 SCF/SCF interface

9.1 SCF/SCF operations and arguments

```
IN-CS2-SCF-SCF-ops-args {itu-t recommendation q 1228 modules(0) in-cs2-scf-scf-ops-args (13) version1(0)}

-- The profiling of Directory Operations Parameters for the SCF-SCF relationship is outside the scope of
-- IN CS-2. Optional parameters received but not used in the SCF-SCF case are ignored.
-- Appropriate parameters to be used should be established via agreement ahead of time.
DEFINITIONS IMPLICIT TAGS ::=
BEGIN
IMPORTS
        OPERATION, Code, ERROR
FROM Remote-Operations-Information-Objects ros-InformationObjects

        SecurityParameters,
        Credentials,
        SecurityProblem,
        securityError
FROM DirectoryAbstractService directoryAbstractService

        OPTIONALLY-PROTECTED{}
FROM EnhancedSecurity enhancedSecurity

        PROTECTION-MAPPING
FROM Notation guls-Notation

        AccessPointInformation
FROM DistributedOperations distributedOperations

        opcode-establishChargingRecord,
        opcode-handlingInformationRequest,
        opcode-handlingInformationResult,
        opcode-networkCapability,
        opcode-notificationProvided,
        opcode-confirmedNotificationProvided,
        opcode-provideUserInformation,
        opcode-confirmedReportChargingInformation,
        opcode-reportChargingInformation,
        opcode-requestNotification
FROM IN-CS2-operationcodes operationcodes

        EXTENSION,
        PARAMETERS-BOUND,
        SupportedExtensions {}
FROM IN-CS2-classes

        AccountNumber,
        ActivableServices,
        BearerCapabilities,
        BearerCapability {},
        CallConditions {},
        CalledPartyNumber {},
        CallingPartyNumber {},
        CallingPartysCategory,
        CallRecord {},
        Carrier,
        Cause {},
        ChargingParameters {},
        Digits {},
        DisplayInformation {},
        ErrorTreatment,
        ExtensionField {},
        HighLayerCompatibilities,
        HighLayerCompatibility,
        InfoToSend {},
        InfoType,
        Integer4,
        InteractionStrategy,
        InvokableService,
        Language,
        LocationNumber {},
        Notification,
        NotificationInformation {},
        NumberMatch {},
        OriginalCalledPartyID {},
        ReceivedInformation {},
        RedirectingPartyID {},
        RedirectionInformation,
        RequestedNotifications {},
        RequestedType,
        RoutingAddress {},
        ScfAddress {},
        ScfID {},
        SubscriberId {},
        SupplementaryServices,
        ToneId,
        TraceInformation{},
        TraceItem{},
        UnavailableNetworkResource,
        UserCredit {},
        UserInfo {},
        UserInformation {},
        UserInteractionModes
```

```
FROM IN-CS2-datatypes datatypes

        improperCallerResponse,
        missingCustomerRecord,
        missingParameter,
        parameterOutOfRange,
        systemFailure,
        unexpectedComponentSequence,
        unexpectedDataValue,
        unexpectedParameter,
        chainingRefused
FROM IN-CS2-errortypes errortypes

        errcode-scfReferral,
        errcode-scfTaskRefused
FROM IN-CS2-errorcodes errorcodes

        AuthenticationLevel
FROM BasicAccessControl basicAccessControl

        SPKM-ERROR
FROM SpkmGssTokens spkmGssTokens

        activityTest
FROM IN-CS2-SSF-SCF-ops-args ssf-scf-Operations

        ros-InformationObjects, ds-UsefulDefinitions,  operationcodes,
        classes, guls-Notation, guls-SecurityTransformations, errortypes, errorcodes,
        scf-scf-Protocol, ssf-scf-Operations, datatypes, spkmGssTokens
FROM IN-CS2-object-identifiers {itu-t recommendation q 1228 modules(0) in-cs2-object-identifiers(17)
version1(0)}


        directoryAbstractService, enhancedSecurity, distributedOperations, basicAccessControl
FROM UsefulDefinitions ds-UsefulDefinitions
;


establishChargingRecord {PARAMETERS-BOUND : bound}  OPERATION ::= {
        ARGUMENT        EstablishChargingRecordArg {bound}
        RETURN RESULT           FALSE
        ERRORS          {missingCustomerRecord |
                                missingParameter |
                                systemFailure |
                                scfTaskRefused |
                                unexpectedComponentSequence |
                                unexpectedDataValue |
                                unexpectedParameter |
                                parameterOutOfRange|
                                securityError
                                }
        CODE            opcode-establishChargingRecord
        }
```

-- Direction: supporting SCF $\rightarrow$ controlling SCF, Timer: T_{ecr}
-- This operation is used by the supporting SCF to give charging information to the controlling
-- SCF so that it can charge the user (on-line charging included).

```
EstablishChargingRecordArg  {PARAMETERS-BOUND : bound} ::= OPTIONALLY-PROTECTED {
SEQUENCE {

        userCredit                  [0] UserCredit {bound}              OPTIONAL,
        chargingParameters          [1] ChargingParameters {bound}      OPTIONAL,
        reportExpected              [2] BOOLEAN                         DEFAULT TRUE,
        securityParameters          [3] SecurityParameters              OPTIONAL,
        extensions                  [4] SEQUENCE SIZE (1..bound.&numOfExtensions)
                                  OF
        ExtensionField {bound}   OPTIONAL,
        ...
        },
        SCFQOP.&scfArgumentQOP{@scfqop}
        }


handlingInformationRequest {PARAMETERS-BOUND : bound}  OPERATION ::= {
        ARGUMENT             HandlingInformationRequestArg {bound}
        RETURN RESULT        FALSE
        ERRORS {missingCustomerRecord |
                             missingParameter |
                             parameterOutOfRange |
                             systemFailure |
                             scfTaskRefused |
                             unexpectedComponentSequence |
                             unexpectedDataValue |
                             unexpectedParameter |
                             securityError |
                             scfReferral
                             }
        LINKED      {handlingInformationResult {bound}}
        CODE        opcode-handlingInformationRequest
        }
```

-- Direction: controlling SCF → supporting SCF (or IAF), Timer: T_{hi}
-- This operation may be used to request the execution of an SLP
-- in the assisting SCF and to provide to the assisting
-- SCF the context of the call so that it can help the controlling SCF in the processing of the call.

```
HandlingInformationRequestArg {PARAMETERS-BOUND : bound} ::= OPTIONALLY-PROTECTED
{SEQUENCE {
        requestedType               [0] RequestedType                       OPTIONAL,
        callingPartyNumber          [1] CallingPartyNumber {bound}          OPTIONAL,
        locationNumber              [2] LocationNumber {bound}              OPTIONAL,
        calledPartyNumber           [3] CalledPartyNumber {bound}           OPTIONAL,
        dialledDigits               [4] Digits {bound}                      OPTIONAL,
        redirectingPartyID          [5] RedirectingPartyID {bound}          OPTIONAL,
        redirectionInformation      [6] RedirectionInformation              OPTIONAL,
        originalCalledPartyID       [7] OriginalCalledPartyID {bound}       OPTIONAL,
        numberOfCallAttempts        [8] INTEGER (1..bound.&ub-nbCall)       OPTIONAL,
        highLayerCompatibility      [9] HighLayerCompatibility              OPTIONAL,
        bearerCapability            [10] BearerCapability {bound}           OPTIONAL,
        invokedSupplementaryService [11] InvokableService                   OPTIONAL,
        activeSupplementaryServices [12] ActivableServices                  OPTIONAL,
        causeOfLastCallFailure      [13] Cause {bound}                      OPTIONAL,
        userInteractionModes        [14] UserInteractionModes               OPTIONAL,
        callingPartysCategory       [15] CallingPartysCategory              OPTIONAL,
        callingPartyBusinessGroupID [16] OCTET STRING                       OPTIONAL,
        securityParameters          [17] SecurityParameters                 OPTIONAL,
        extensions                     [18] SEQUENCE SIZE (1..bound.&numOfExtensions)
                                              OF ExtensionField {bound}              OPTIONAL,
        ...
        },
        SCFQOP.&scfArgumentQOP{@scfqop}
        }


handlingInformationResult {PARAMETERS-BOUND : bound}  OPERATION ::= {
        ARGUMENT          HandlingInformationResultArg {bound}
        RETURN RESULT           FALSE
        ERRORS            { missingParameter |
                                systemFailure |
                                parameterOutOfRange |
                                unexpectedComponentSequence |
                                unexpectedDataValue |
                                unexpectedParameter |
                                securityError
                                }
        CODE              opcode-handlingInformationResult
        }
```

-- Direction: supporting SCF(or IAF) $\rightarrow$ controlling SCF, Timer: T_{hir}
-- This operation is used by the assisting SCF to send information to the controlling SCF on how
-- to process the call and to give conditions under which it should be involved in the call
-- processing.

```
HandlingInformationResultArg {PARAMETERS-BOUND : bound} ::= OPTIONALLY-PROTECTED
{SEQUENCE {
          routingAddress              [0] RoutingAddress {bound}                    OPTIONAL,
          highLayerCompatibility      [1] HighLayerCompatibility                    OPTIONAL,
          supplementaryServices       [2] SupplementaryServices                     OPTIONAL,
          preferredLanguage           [3] Language                                  OPTIONAL,
          carrier                     [4] Carrier                                   OPTIONAL,
          callingPartyNumber          [5] CallingPartyNumber {bound}                OPTIONAL,
          originalCalledPartyID       [6] OriginalCalledPartyID  {bound}            OPTIONAL,
          redirectingPartyID          [7] RedirectingPartyID  {bound}               OPTIONAL,
          redirectionInformation      [8] RedirectionInformation                    OPTIONAL,
          callingPartysCategory       [9] CallingPartysCategory                     OPTIONAL,
          securityParameters          [10] SecurityParameters                       OPTIONAL,
          extensions                  [11] SEQUENCE SIZE (1..bound.&numOfExtensions)
                                            OF ExtensionField {bound}               OPTIONAL,
          ...
          },
          SCFQOP.&scfArgumentQOP{@scfqop}
          }


networkCapability  {PARAMETERS-BOUND : bound}  OPERATION ::= {
          ARGUMENT          NetworkCapabilityArg {bound}
          RESULT            NetworkCapabilityResultArg {bound}
          ERRORS            {missingCustomerRecord |
                                  missingParameter |
                                  systemFailure |
                                  scfTaskRefused |
                                  unexpectedComponentSequence |
                                  unexpectedDataValue |
                                  unexpectedParameter |
                                  securityError
                                  }
```

```
        CODE                opcode-networkCapability
        }

-- Direction: supporting SCF → controlling  SCF, Timer: Tnc
-- This operation is used by the supporting SCF to request from the controlling SCF which type of
-- service it supports.

NetworkCapabilityArg  {PARAMETERS-BOUND : bound} ::= OPTIONALLY-PROTECTED { SEQUENCE {
        bearerCapabilities          [0] BearerCapabilities                      OPTIONAL,
        highLayerCompatibilities    [1] HighLayerCompatibilities                OPTIONAL,
        supplementaryServices       [2] SupplementaryServices                   OPTIONAL,
        securityParameters          [3] SecurityParameters                      OPTIONAL,
        extensions                  [4] SEQUENCE SIZE (1..bound.&numOfExtensions)
                                        OF ExtensionField {bound}               OPTIONAL,
        ...
        },
        SCFQOP.&scfArgumentQOP{@scfqop}
        }

NetworkCapabilityResultArg  {PARAMETERS-BOUND : bound} ::= OPTIONALLY-PROTECTED {
SEQUENCE {
        bearerCapabilities          [0] BearerCapabilities                      OPTIONAL,
        highLayerCompatibilities    [1] HighLayerCompatibilities                OPTIONAL,
        supplementaryServices       [2] SupplementaryServices                   OPTIONAL,
        securityParameters          [3] SecurityParameters                      OPTIONAL,
        extensions                  [4] SEQUENCE SIZE (1..bound.&numOfExtensions)
                                        OF ExtensionField {bound}               OPTIONAL,
        ...
        },
        SCFQOP.&scfArgumentQOP{@scfqop}
        }

notificationProvided  {PARAMETERS-BOUND : bound}   OPERATION ::= {
        ARGUMENT                    NotificationProvidedArg {bound}
        RETURN RESULT               FALSE
        ERRORS      {missingParameter |
                                    systemFailure |
                                    scfTaskRefused |
                                    unexpectedComponentSequence |
                                    unexpectedDataValue |
                                    unexpectedParameter|
                                    missingCustomerRecord |
                                    parameterOutOfRange |
                                    securityError
                                    }
                CODE                opcode-notificationProvided
        }

-- Direction: controlling SCF → supporting SCF(or IAF), Timer: Tnp
-- This operation is used by the controlling SCF to request assistance from the assisting SCF
-- under specific call conditions specified prior to the sending of the operation or to notify the
-- outcome of a previous intervention of the assisting SCF.
```

```
NotificationProvidedArg {PARAMETERS-BOUND : bound} ::= OPTIONALLY-PROTECTED { SEQUENCE {
        notification               [0] Notification,
        notificationInformation    [1] NotificationInformation {bound}              OPTIONAL,
        securityParameters         [2] SecurityParameters                           OPTIONAL,
        extensions                 [3] SEQUENCE SIZE (1..bound.&numOfExtensions)
                                      OF ExtensionField {bound}                     OPTIONAL,
        ...
        },
        SCFQOP.&scfArgumentQOP{@scfqop}
        }

confirmedNotificationProvided {PARAMETERS-BOUND : bound} OPERATION ::= makeConfirm {
        notificationProvided{bound},
        opcode-confirmedNotificationProvided}
```

--Direction: controlling SCF → supporting SCF , Timer: T_{cnp}

```
provideUserInformation  {PARAMETERS-BOUND : bound}   OPERATION ::= {
        ARGUMENT        ProvideUserInformationArg {bound}
        RESULT          ProvideUserInformationResultArg {bound}
        ERRORS          {missingCustomerRecord |
                                missingParameter |
                                systemFailure |
                                scfTaskRefused |
                                unexpectedComponentSequence |
                                unexpectedDataValue |
                                unexpectedParameter |
                                improperCallerResponse |
                                parameterOutOfRange |
                                securityError
        CODE            opcode-provideUserInformation
        }
```

-- Direction: supporting SCF → controlling SCF, Timer: T_{pui}
-- This operation is used by the supporting SCF to request information from the user that can be
-- interrogated by the controlling SCF.

```
ProvideUserInformationArg  {PARAMETERS-BOUND : bound} ::= OPTIONALLY-PROTECTED {
SEQUENCE {
        constraints                [0] CollectedInfo,
        infoToSend                 [1] InformationToSend {bound},
        errorInfo                  [2] InformationToSend {bound}                    OPTIONAL,
        typeOfRequestedInfo        [3] InfoType             DEFAULT numericString,
        numberOfAllowedRetries     [4] INTEGER (0.. 127)    DEFAULT        0,
        actions                    [5] Actions                                      OPTIONAL,
        preferredLanguage          [6] Language                                     OPTIONAL,
        securityParameters         [7] SecurityParameters                           OPTIONAL,
        extensions                 [8] SEQUENCE SIZE (1.. bound.&numOfExtensions)
                                      OF ExtensionField {bound}                     OPTIONAL,
        ...
        },
        SCFQOP.&scfArgumentQOP{@scfqop}
        }



CollectedInfo ::= CHOICE {
        collectedDigits            [0] CollectedDigits,
        iA5Information             [1] BOOLEAN
}
```

```
CollectedDigits ::= SEQUENCE {
        minimumNbOfDigits          [0] INTEGER (1.. 127)     DEFAULT 1,
        maximumNbOfDigits          [1] INTEGER (1.. 127) ,
        endOfReplyDigit            [2] IA5String (SIZE (1) )            OPTIONAL,
        cancelDigit                [3] IA5String (SIZE (1) )            OPTIONAL,
        startDigit                 [4] IA5String (SIZE (1) )            OPTIONAL,
        firstDigitTimeOut          [5] INTEGER (1.. 127)                OPTIONAL,
        interDigitTimeOut          [6] INTEGER (1.. 127)                OPTIONAL,
        errorTreatment             [7] ErrorTreatment        DEFAULT reportErrorToScf,
        interruptableAnnInd        [8] BOOLEAN                          DEFAULT TRUE,
        voiceInformation           [9] BOOLEAN                          DEFAULT FALSE,
        voiceBack                  [10] BOOLEAN                         DEFAULT FALSE
        }


InformationToSend {PARAMETERS-BOUND} ::= CHOICE {
        inbandInfo                 [0] InbandInfo,
        tone                       [1] Tone,
        displayInformation         [2] DisplayInformation{bound}
        }


InbandInfo ::= SEQUENCE {
        messageId                  [0] MessageID,
        numberOfRepetitions        [1] INTEGER (1..127)                         OPTIONAL,
        duration                   [2] INTEGER (1..32767)                       OPTIONAL,
        interval                   [3] INTEGER (1..32767)                       OPTIONAL
        }


Tone ::= SEQUENCE {
        toneId                     [0] Integer4,
        duration                   [1] Integer4                                 OPTIONAL
        }


Actions ::= ENUMERATED {
        play (0) ,
        playandcollect (1)
        }



MessageID ::= OBJECT IDENTIFIER


ProvideUserInformationResultArg   {PARAMETERS-BOUND : bound}
::= OPTIONALLY-PROTECTED { SEQUENCE {
        userInformation            [0] ReceivedInformation {bound},
        securityParameters         [1] SecurityParameters                       OPTIONAL,
        extensions                 [1] SEQUENCE SIZE (1..bound.&numOfExtensions)
                                      OF ExtensionField {bound}                 OPTIONAL
        },
        SCFQOP.&scfArgumentQOP{@scfqop}
        }
```

```
reportChargingInformation   {PARAMETERS-BOUND : bound} OPERATION ::= {
        ARGUMENT        ReportChargingInformationArg {bound}
        RETURN RESULT                   FALSE
        ERRORS          {missingCustomerRecord |
                                        missingParameter |
                                        systemFailure |
                                        scfTaskRefused |
                                        unexpectedComponentSequence |
                                        unexpectedDataValue |
                                        unexpectedParameter |
                                        parameterOutOfRange |
                                        securityError
                                        }
        CODE            opcode-reportChargingInformation
        }
```

-- Direction: controlling SCF $\rightarrow$ *supporting SCF, Timer*: T_{rci}
-- This operation is used to give to the assisting network charging information collected by the
-- controlling network.

```
ReportChargingInformationArg   {PARAMETERS-BOUND : bound} ::= OPTIONALLY-PROTECTED {
SEQUENCE {
        callRecord              [0] CallRecord {bound}                          OPTIONAL,
        remainingUserCredit     [1] UserCredit {bound}                          OPTIONAL,
        uniqueCallID            [2] CallIdentifier                              OPTIONAL,
        accountNumber           [3] AccountNumber                               OPTIONAL,
        securityParameters      [4] SecurityParameters                          OPTIONAL
        },
        SCFQOP.&scfArgumentQOP{@scfqop}
        }

CallIdentifier ::= Integer4


confirmedReportChargingInformation {PARAMETERS-BOUND : bound} OPERATION ::= makeConfirm {
        reportChargingInformation{bound},
        opcode-confirmedReportChargingInformation
        }
```

-- Direction: controlling SCF $\rightarrow$ *supporting SCF , Timer*: T_{crci}

```
requestNotification    {PARAMETERS-BOUND : bound}  OPERATION ::= {
        ARGUMENT        RequestNotificationArg {bound}
        RETURN RESULT           FALSE
        ERRORS      {missingParameter|
                        systemFailure|
                        scfTaskRefused|
                        unexpectedComponentSequence|
                        unexpectedDataValue|
                        unexpectedParameter|
                        parameterOutOfRange|
                        missingCustomerRecord|
                        securityError
                        }
        CODE            opcode-requestNotification
        }
```

-- Direction: supporting SCF (or IAF) $\rightarrow$ *controlling SCF, Timer:* T_{rn}
-- This operation is used by the assisting SCF to request notification from the controlling SCF
-- under specific call conditions specified by this operation.

```
RequestNotificationArg    {PARAMETERS-BOUND : bound} ::= OPTIONALLY-PROTECTED {SEQUENCE {
        requestedNotifications      [0] RequestedNotifications {bound},
        securityParameters          [1] SecurityParameters                              OPTIONAL
        },
        SCFQOP.&scfArgumentQOP{@scfqop}
        }


scfBind {PARAMETERS-BOUND : bound} OPERATION ::= {
        ARGUMENT                    SCFBindArgument{bound}
        RESULT                      SCFBindResult {bound}
        ERRORS              { scfBindFailure}
                }

-- Direction: controlling SCF → assisting SCF (or IAF), Timer: T_bi
-- This operation is used to establish a relationship between two SCFs. It is sent by the controlling SCF each time it
-- needs to initiate communications with another (supporting) SCF.

SCFBindArgument  {PARAMETERS-BOUND : bound} ::= SEQUENCE {
        agreementID                 [0] AgreementID,
        originatingScfAddress       [1] ScfAddress {bound}                      OPTIONAL,
-- absent in a chained operation request which crosses an international internetworking boundary
        credentials                 [2] Credentials                             OPTIONAL
        }


SCFBindResult  {PARAMETERS-BOUND : bound} ::= SEQUENCE {
        respondingScfAddress        [0] ScfAddress {bound}                      OPTIONAL,
-- absent in a chained operation request which crosses an international internetworking boundary
        returnedCredentials         [1] Credentials                             OPTIONAL
        }


AgreementID ::= OBJECT IDENTIFIER


scfUnbind OPERATION ::=  {
RETURN RESULT                               FALSE

ALWAYS RESPONDS                             FALSE
}
-- Direction: controlling SCF → assisting SCF (or IAF)
-- The SCF Unbind operation is used by the controlling SCF to close the relationship with the supporting SCF.

scfChained  {OPERATION : operation, PARAMETERS-BOUND : bound}  OPERATION ::= {
        ARGUMENT            OPTIONALLY-PROTECTED {SEQUENCE {
                chainedArgument     ChainingArgument {bound },
                argument            [0] operation.&ArgumentType
                                                            OPTIONAL
                },
                SCFQOP.&scfArgumentQOP{@scfqop}
                }
        RESULT      OPTIONALLY-PROTECTED {SEQUENCE  {
                chainedResult         ChainingResult {bound},
                result                [0]operation.&ResultType
                                                                  OPTIONAL
                },
                SCFQOP.&scfArgumentQOP{@scfqop}
                }
        ERRORS                 {operation.&Errors |
                               chainingRefused |
                               securityError |
                               scfReferral
                               }
        CODE                   operation.&operationCode
        }

ChainingArgument    {PARAMETERS-BOUND : bound} ::= SEQUENCE  {
        originatingSCF              [0] ScfID {bound},
        target                      [1] SubscriberId {bound}                          OPTIONAL,
        traceInformation            [2] TraceInformation{bound},
        scfAuthenticationLevel      [3] AuthenticationLevel          DEFAULT  basicLevels : {level none},
        timeLimit                   [4] UTCTime                                        OPTIONAL,
        securityParameters          [5] SecurityParameters                             OPTIONAL,
        extensions                  [6] SEQUENCE SIZE (1..bound.&numOfExtensions)
                                        OF ExtensionField {bound}                      OPTIONAL,
        ...
        }


ChainingResult    {PARAMETERS-BOUND : bound} ::= SEQUENCE  {
        ultimateResponder           [0] ScfAddress {bound}                             OPTIONAL,
        traceInformation            [1] TraceInformation{bound},
        securityParameters          [2] SecurityParameters                             OPTIONAL,
        extensions                  [3] SEQUENCE SIZE (1..bound.&numOfExtensions)
                                        OF ExtensionField {bound}                      OPTIONAL,
        ...
        }



makeConfirm {OPERATION:operation, Code:code} OPERATION ::= {
        &ArgumentType               operation.&ArgumentType                            OPTIONAL,
        &argumentTypeOptional       operation.&argumentTypeOptional OPTIONAL,
        &ResultType                 NULL,
        &Errors                     operation.&Errors                                  OPTIONAL,
        &alwaysReturns              BOOLEAN TRUE,
        &operationCode              code}


chainedEstablishChargingRecord   {PARAMETERS-BOUND : bound}   OPERATION ::=
scfChained{establishChargingRecord{bound},bound}

chainedHandlingInformationRequest {PARAMETERS-BOUND : bound} OPERATION ::= scfChained
{handlingInformationRequest{bound},bound}

chainedHandlingInformationResult {PARAMETERS-BOUND : bound} OPERATION ::=
scfChained{handlingInformationResult{bound},bound}

chainedNetworkCapability {PARAMETERS-BOUND : bound}    OPERATION ::= scfChained
{networkCapability{bound},bound}
```

```
chainedNotificationProvided {PARAMETERS-BOUND : bound} OPERATION ::= scfChained
{notificationProvided{bound},bound}

chainedConfirmedNotificationProvided {PARAMETERS-BOUND : bound} OPERATION ::= scfChained
{confirmedNotificationProvided {bound},bound}

chainedProvideUserInformation {PARAMETERS-BOUND : bound} OPERATION ::= scfChained
{provideUserInformation{bound},bound}

chainedReportChargingInformation {PARAMETERS-BOUND : bound} OPERATION ::= scfChained
{reportChargingInformation{bound},bound}

chainedConfirmedReportChargingInformation {PARAMETERS-BOUND : bound} OPERATION ::=
scfChained{confirmedReportChargingInformation{bound}, bound}

chainedRequestNotification {PARAMETERS-BOUND : bound} OPERATION ::=
scfChained{requestNotification{bound}, bound}


SCFQOP ::= CLASS {
        &scfqop-id                  OBJECT IDENTIFIER UNIQUE,
        &scfBindErrorQOP            PROTECTION-MAPPING,
        &scfErrorsQOP               PROTECTION-MAPPING,
        &scfArgumentQOP             PROTECTION-MAPPING,
        &scfResultQOP               PROTECTION-MAPPING
        }
WITH SYNTAX {
        SCFQOP-ID                   &scfqop-id,
        SCFBINDERROR-QOP            &scfBindErrorQOP,
        SCFERRORS-QOP               &scfErrorsQOP,
        SCFOPARG-QOP                &scfArgumentQOP,
        SCFOPRES-QOP                &scfResultQOP
        }


scfBindFailure ERROR ::= {
        PARAMETER                   FailureReason
        }


FailureReason ::= CHOICE {
        systemFailure               [0] UnavailableNetworkResource,
        scfTaskRefused              [1] ScfTaskRefusedParameter,
        securityError               [2] SET {
                problem             [0] SecurityProblem,
                spkmInfo            [1] SPKM-ERROR
                }
        }


scfTaskRefused ERROR ::= {
        PARAMETER                   ScfTaskRefusedParameter
        CODE                        errcode-scfTaskRefused
        }
```

```
ScfTaskRefusedParameter ::= OPTIONALLY-PROTECTED { SEQUENCE {
        reason          ENUMERATED {
                generic(0),
                unobtainable (1),
                congestion(2)
                -- other values FFS
                },
        securityParameters      [1] SecurityParameters                          OPTIONAL
    },
    SCFQOP.&scfErrorsQOP{@scfqop}
    }


scfReferral ERROR ::= {
    PARAMETER                   ReferralParameter
    CODE                        errcode-scfReferral
    }


ReferralParameter ::= OPTIONALLY-PROTECTED {
    SEQUENCE {
        tryhere                 [0] AccessPointInformation,
        securityParameters      [1] SecurityParameters                          OPTIONAL
        },
    SCFQOP.&scfErrorsQOP{@scfqop}
    }

END
```

Table 9-1 below lists all operation timers and the value range for each timer. The definitive value for each operation timer may be network-specific and has to be defined by the network operator.

NOTE – The following value ranges do apply for operation specific timers in INAP:

short:	1-10 seconds.
medium:	1-60 seconds.
long:	1 second-30 minutes.
ffs:	For Further Study.

Table 9-1/Q.1228 – Operation timers and their value range

Operation name	Timer	Value range
EstablishChargingRecord	T_{ecr}	Short
HandlingInformationRequest	T_{hi}	Short
HandlingInformationResult	T_{hir}	Short
NetworkCapability	T_{nc}	Short
NotificationProvided	T_{np}	Short
ConfirmedNotificationProvided	T_{cnp}	Short
ProvideUserInformation	T_{pui}	Long
ReportChargingInformation	T_{rci}	Short
ConfirmedReportChargingInformation	T_{crci}	Short
RequestNotification	T_{rn}	Short
ScfBind	T_{bi}	Medium

9.2 SCF/SCF contracts, packages and Application Contexts

9.2.1 Protocol overview

The **scf-scfContract** expresses the form of the service in which the SCF, a ROS-object of class **scf-scf**, initiates the contract. A ROS-object of class **scf-scf** responds in this contract.

```
scf-scfContract  CONTRACT ::= {
        CONNECTION                    scf-scfConnectionPackage{networkSpecificBoundSet}
        INITIATOR CONSUMER OF    {
                                             activityTestPackage |
                                             handlingInformationPackage
{networkSpecificBoundSet}
                                             }
        RESPONDER CONSUMER OF {
                                             activityTestPackage |
                                             chargingInformationPackage

{networkSpecificBoundSet}|                   networkCapabilityPackage

{networkSpecificBoundSet}|                   notificationPackage

{networkSpecificBoundSet}|                   userInformationPackage

{networkSpecificBoundSet}                    }
        ID                                   id-contract-scf-scf
        }
```

When two SCFs are located in different IN physical entities, this association contract shall be realized as an SS7 application layer protocol. The definition of this protocol in terms of an SS7 application context is provided in 9.2.2.

The **scf-scfContract** is composed of a connection package, **scf-scfConnectionPackage** and six operation packages, **handlingInformationPackage, notificationPackage, chargingInformationPackage, activityTestPackage, userInformationPackage** and **networkCapabilityPackage**.

The **dsspContract** is defined as an information object of class CONTRACT.

```
dsspContract  CONTRACT ::= {
        CONNECTION                    dsspConnectionPackage {networkSpecificBoundSet}
        INITIATOR CONSUMER OF         {chainedSCFOperationPackage{networkSpecificBoundSet}}
        ID                                                         id-contract-dssp
        }
```

When a pair of SCFs from different open systems interact, this association contract is realized as an SS7 application layer protocol, referred to as the IN Distributed SCF System Protocol (DSSP). The definition of this protocol in terms of an SS7 application context is provided in 9.2.2.

The **dsspContract** is composed of a connection package, **dsspConnectionPackage** and one operation package, **chainedSCFOperationPackage**.

The connection package, **scf-scfConnectionPackage**, is defined as an information object of class CONNECTION-PACKAGE defined below.

```
scf-scfConnectionPackage {PARAMETERS-BOUND : bound} CONNECTION-PACKAGE ::= {
        BIND                       scfBind{bound}
        UNBIND                     scfUnbind
        RESPONDER UNBIND           FALSE
        ID                                             id-package-scf-scfConnection
        }
```

The connection package, **dsspConnectionPackage**, is defined as an information object of class CONNECTION-PACKAGE.

```
dsspConnectionPackage {PARAMETERS-BOUND : bound} CONNECTION-PACKAGE ::= {
        BIND                        scfBind{bound}
        UNBIND                      scfUnbind
        RESPONDER UNBIND            FALSE
        ID                                                  id-package-dsspConnection
        }
```

The operation packages, **handlingInformationPackage, notificationPackage, chargingInformationPackage, activityTestPackage, userInformationPackage** and **networkCapabilityPackage**, are defined as information objects of class OPERATION-PACKAGE. The operations of these packages are defined in 9.1.

```
handlingInformationPackage  {PARAMETERS-BOUND : bound}   OPERATION-PACKAGE ::= {
        CONSUMER INVOKES   {handlingInformationRequest {bound}}
        SUPPLIER INVOKES   {handlingInformationResult {bound}}
        ID                                                id-package-handlingInformation
        }

notificationPackage  {PARAMETERS-BOUND : bound}   OPERATION-PACKAGE ::= {
        CONSUMER INVOKES   { requestNotification {bound}}
        SUPPLIER INVOKES   { notificationProvided {bound}| confirmedNotificationProvided }
        ID                                                 id-package-notification
        }

chargingInformationPackage  {PARAMETERS-BOUND : bound}   OPERATION-PACKAGE ::= {
        CONSUMER INVOKES   { establishChargingRecord {bound} }
        SUPPLIER INVOKES   {

        confirmedReportChargingInformation{bound} |
                                                          reportChargingInformation  {bound}
                                                          }
        ID                                                id-package-chargingInformation}

userInformationPackage  {PARAMETERS-BOUND : bound} OPERATION-PACKAGE ::= {
        CONSUMER INVOKES   {provideUserInformation  {bound}  }
        ID                                                id-package-userInformation
        }

networkCapabilityPackage  {PARAMETERS-BOUND : bound} OPERATION-PACKAGE ::= {
        CONSUMER INVOKES   { networkCapability  {bound}}
        ID                                                id-package-networkCapability
        }
```

The operation package, **chainedSCFOperationPackage** is defined as information objects of class OPERATION-PACKAGE. The operations of this packages are defined in 9.1.

```
chainedSCFOperationPackage {PARAMETERS-BOUND : bound} OPERATION-PACKAGE ::= {
        CONSUMER INVOKES          {
                                           chainedHandlingInformationRequest  {bound} |
                                           chainedNotificationProvided { bound}|
                                           chainedConfirmedNotificationProvided {bound}|
                                           chainedReportChargingInformation { bound}|
        chainedConfirmedReportChargingInformation{bound}                  }
        SUPPLIER INVOKES   {
        chainedEstablishChargingRecord { bound}|
        chainedHandlingInformationResult { bound}|
        chainedNetworkCapability { bound}|
        chainedProvideUserInformation { bound}|
        chainedRequestNotification { bound}
        }
        ID                                  id-package-chainedSCFOperations
        }
```

Abstract syntax

This version of the INAP requires the support of two abstract syntaxes:

a) the abstract syntax of TC dialogue control protocol data units, **dialogue-abstract-syntax**, which is needed to establish the dialogues between FEs and specified in Recommendation Q.773;

b) the abstract syntax for conveying the protocol data units for invoking the operations involved in the operation packages specified in 9.2.2 and reporting their outcome.

The ASN.1 type from which the values of the last abstract syntax are derived is specified using the parameterized types **TCMessage {}** defined in Recommendation Q.773.

All these abstract syntaxes shall (as a minimum) be encoded according to the Basic ASN.1 encoding rules with the restrictions listed in Recommendation Q.773.

The SCF-SCF INAP ASEs that realize the operation packages and the connection package specified in 9.2.2 share a single abstract syntax, **scf-scfOperationsAbstractSyntax**. This is specified as an information object of the class ABSTRACT-SYNTAX.

```
scf-scfOperationsAbstractSyntax  ABSTRACT-SYNTAX ::= {
        BasicSCF-SCF-PDUs
        IDENTIFIED BY   id-as-scf-scfOperationsAS}
BasicSCF-SCF-PDUs ::= TCMessage {{SCF-SCF-Invokable}, {SCF-SCF-Returnable}}
SCF-SCF-Invokable {PARAMETERS-BOUND} OPERATION ::= {
                                activityTest |
                                establishChargingRecord {bound}|
                                confirmedNotificationProvided {bound}|
                                confirmedReportChargingInformation {bound} |
                                handlingInformationRequest {bound}|
                                handlingInformationResult {bound}|
                                networkCapability {bound}|
                                notificationProvided {bound}|
                                provideUserInformation {bound}|
                                reportChargingInformation {bound}|
                                requestNotification {bound}
        }
SCF-SCF-Returnable {PARAMETERS-BOUND}  OPERATION ::= {
                                activityTest |
                                establishChargingRecord {bound}|
                                confirmedNotificationProvided {bound}|
                                confirmedReportChargingInformation {bound}|
                                handlingInformationRequest {bound}|
                                handlingInformationResult {bound}|
                                networkCapability {bound}|
                                provideUserInformation {bound}|
                                requestNotification {bound}
        }
```

The Distributed SCF ASEs that realize the operation specified in 9.1 share a single abstract syntax, **distributedSCFSystemAbstractSyntax**. This is specified as an information object of the class ABSTRACT-SYNTAX.

```
distributedSCFSystemAbstractSyntax  ABSTRACT-SYNTAX ::= {
        BasicDSSP-PDUs
        IDENTIFIED BY   id-as-distributedSCFSystemAS}
BasicDSSP-PDUs ::= TCMessage {{DSSP-Invokable}, {DSSP-Returnable}}
DSSP-Invokable {PARAMETERS-BOUND : bound} OPERATION ::= {
                                chainedHandlingInformationRequest {bound}|
                                chainedNotificationProvided {bound}|
                                chainedConfirmedNotificationProvided{bound} |
                                chainedReportChargingInformation {bound}|
                                chainedConfirmedReportChargingInformation {bound}

        }
DSSP-Returnable {PARAMETERS-BOUND : bound} OPERATION ::= {
                                chainedHandlingInformationRequest {bound}|
                                chainedConfirmedNotificationProvided{bound} |
                                chainedConfirmedReportChargingInformation {bound}
        }
```

The realization of the connection package specified in 9.2.2 uses a separate abstract syntax, **distributedSCFBindingAbstractSyntax**. This is specified as an information object of the class ABSTRACT-SYNTAX.

```
distributedSCFBindingAbstractSyntax  ABSTRACT-SYNTAX  ::= {
        SCF-SCFBinding-PDUs{networkSpecificBoundSet}
        IDENTIFIED BY   id-as-scf-scfBindingAS}
SCF-SCFBinding-PDUs{PARAMETERS-BOUND:bound} ::= CHOICE {
        bind                    Bind {scfBind{bound}},
        unbind                  Unbind {scfUnbind}
        }
```

SCF-SCF Application Contexts

The **scf-scfContract** is realized by four application contexts, **scf-scfOperationsAC**, **distributedSCFSystemAC**, **scf-scfOperationWith3seAC**, and **distributedSCFSystemWith3seAC**. These application contexts are specified as information objects of the class APPLICATION-CONTEXT.

```
scf-scfOperationsAC  APPLICATION-CONTEXT ::= {
        CONTRACT                        scf-scfContract
        DIALOGUE MODE                   structured
        TERMINATION                     basic
        ABSTRACT SYNTAXES               {dialogue-abstract-syntax |
                                                distributedSCFBindingAbstractSyntax |
                                                scf-scfOperationsAbstractSyntax }

        APPLICATION CONTEXT NAME        id-ac-scf-scfOperationsAC
        }


distributedSCFSystemAC  APPLICATION-CONTEXT ::= {
        CONTRACT                        dsspContract
        DIALOGUE MODE                   structured
        TERMINATION                     basic
        ABSTRACT SYNTAXES               {dialogue-abstract-syntax |
                                            distributedSCFSystemAbstractSyntax |
                                            distributedSCFBindingAbstractSyntax}

        APPLICATION CONTEXT NAME        id-ac-distributedSCFSystemAC
        }
```

```
scf-scfOperationsWith3seAC  APPLICATION-CONTEXT ::= {
        CONTRACT                        scf-scfContract
        DIALOGUE MODE                   structured
        TERMINATION                     basic
        ADDITIONAL ASE                  {id-se-threewayse}
        ABSTRACT SYNTAXES               {dialogue-abstract-syntax |
                                                distributedSCFBindingAbstractSyntax |
                                                scf-scfOperationsAbstractSyntax  |
                                                inSESEAbstractSyntax }
        APPLICATION CONTEXT NAME        id-ac-scf-scfOperationsWith3seAC
        }

distributedSCFSystemWith3seAC  APPLICATION-CONTEXT ::= {
        CONTRACT                        dsspContract
        DIALOGUE MODE                   structured
        TERMINATION                     basic
        ADDITIONAL ASE                  {id-se-threewayse}
        ABSTRACT SYNTAXES               {dialogue-abstract-syntax |
                                                distributedSCFSystemAbstractSyntax |
                                                distributedSCFBindingAbstractSyntax |
                                                inSESEAbstractSyntax }
        APPLICATION CONTEXT NAME        id-ac-distributedSCFSystemWith3seAC
        }
```

9.2.2 ASN.1 modules

```
-- This subclause includes all of the ASN.1 type and value definitions contained in this SCF/SCF Specification, in the
-- form of the ASN.1 module, " IN-CS2-SCF-SCF-pkgs-contracts-acs ".
IN-CS2-SCF-SCF-pkgs-contracts-acs {itu-t recommendation q 1228 modules(0) in-cs2-scf-scf-pkgs-contracts-acs
(14) version1(0)}

DEFINITIONS ::=
BEGIN

-- This module describes the operation-packages, contracts and application-contexts used
-- over the SCF-SCF interface.

IMPORTS

        PARAMETERS-BOUND,
        networkSpecificBoundSet
FROM IN-CS2-classes classes


        ROS-OBJECT-CLASS, CONTRACT, OPERATION-PACKAGE, CONNECTION-PACKAGE,
          OPERATION
FROM Remote-Operations-Information-Objects ros-InformationObjects


        Bind{}, Unbind{}
FROM Remote-Operations-Generic-ROS-PDUs ros-genericPDUs

        TCMessage {}
FROM  TCAPMessages tc-Messages

        APPLICATION-CONTEXT, dialogue-abstract-syntax
FROM TC-Notation-Extensions tc-NotationExtensions
        establishChargingRecord {},
        confirmedReportChargingInformation{},
        confirmedNotificationProvided {},
        handlingInformationRequest {},
        handlingInformationResult {},
        networkCapability {},
        notificationProvided {},
        provideUserInformation {},
        reportChargingInformation {},
        requestNotification {},
        chainedHandlingInformationRequest {},
        chainedNotificationProvided {},
        chainedConfirmedNotificationProvided {},
        chainedReportChargingInformation {},
        chainedConfirmedReportChargingInformation{},
        chainedEstablishChargingRecord {},
        chainedHandlingInformationResult {},
        chainedNetworkCapability {},
        chainedProvideUserInformation {},
        chainedRequestNotification {},
        scfBind{},
        scfUnbind
FROM IN-CS2-SCF-SCF-ops-args scf-scf-Operations

        id-ac,
        id-rosObject,
        id-contract,
        id-package,
        id-as,
        id-ac-scf-scfOperationsAC,
        id-ac-distributedSCFSystemAC,
        id-ac-scf-scfOperationsWith3seAC,
        id-ac-distributedSCFSystemWith3seAC,
        id-contract-scf-scf,
        id-contract-dssp,
        id-package-dsspConnection,
        id-package-scf-scfConnection,
        id-package-handlingInformation,
        id-package-notification,
        id-package-chargingInformation,
        id-package-userInformation,
        id-package-networkCapability,
        id-package-chainedSCFOperations,
        id-as-scf-scfOperationsAS,
        id-as-distributedSCFSystemAS,
        id-as-scf-scfBindingAS,
        ds-UsefulDefinitions,
        classes,
        tc-Messages, tc-NotationExtensions,
        ros-InformationObjects, ros-genericPDUs,
        scf-scf-Operations, scf-sdf-Protocol,
        ssf-scf-Operations, ssf-scf-Protocol
FROM IN-CS2-object-identifiers {itu-t recommendation q 1228 modules(0) in-cs2-object-identifiers (17)
version1(0)}

        activityTest
FROM IN-CS2-SSF-SCF-ops-args ssf-scf-Operations

        activityTestPackage
FROM IN-CS2-SSF-SCF-pkgs-contracts-acs ssf-scf-Protocol
```

```
        inSESEAbstractSyntax
FROM IN-CS2-SCF-SDF-Protocol scf-sdf-Protocol

        id-se-threewayse
FROM ProtocolObjectIdentifiers protocolObjectIdentifiers

        protocolObjectIdentifiers
FROM UsefulDefinitions ds-UsefulDefinitions
;

-- Application Contexts --

scf-scfOperationsAC  APPLICATION-CONTEXT ::= {
        CONTRACT                        scf-scfContract
        DIALOGUE MODE                   structured
        TERMINATION                     basic
        ABSTRACT SYNTAXES               {dialogue-abstract-syntax |
                                                distributedSCFBindingAbstractSyntax |
                                                scf-scfOperationsAbstractSyntax }

        APPLICATION CONTEXT NAME        id-ac-scf-scfOperationsAC
        }

distributedSCFSystemAC  APPLICATION-CONTEXT ::= {
        CONTRACT                        dsspContract
        DIALOGUE MODE                   structured
        TERMINATION                     basic
        ABSTRACT SYNTAXES               {dialogue-abstract-syntax |
                                                distributedSCFSystemAbstractSyntax |
                                                distributedSCFBindingAbstractSyntax}

        APPLICATION CONTEXT NAME        id-ac-distributedSCFSystemAC
        }

scf-scfOperationsWith3seAC  APPLICATION-CONTEXT ::= {
        CONTRACT                        scf-scfContract
        DIALOGUE MODE                   structured
        TERMINATION                     basic
        ADDITIONAL ASE                  {id-se-threewayse}
        ABSTRACT SYNTAXES               {dialogue-abstract-syntax |
                                                distributedSCFBindingAbstractSyntax |
                                                scf-scfOperationsAbstractSyntax |
                                                inSESEAbstractSyntax}

        APPLICATION CONTEXT NAME        id-ac-scf-scfOperationsWith3seAC
        }

distributedSCFSystemWith3seAC  APPLICATION-CONTEXT ::= {
        CONTRACT                        dsspContract
        DIALOGUE MODE                   structured
        TERMINATION                     basic
        ADDITIONAL ASE                  {id-se-threewayse}
        ABSTRACT SYNTAXES               {dialogue-abstract-syntax |
                                        distributedSCFSystemAbstractSyntax |
                                        distributedSCFBindingAbstractSyntax |
                                        inSESEAbstractSyntax }

        APPLICATION CONTEXT NAME        id-ac-distributedSCFSystemWith3seAC
        }
```

```
-- Contracts --

scf-scfContract  CONTRACT ::= {
        CONNECTION                    scf-scfConnectionPackage{networkSpecificBoundSet}
        INITIATOR CONSUMER OF         {
                                              activityTestPackage |
                                              handlingInformationPackage
{networkSpecificBoundSet}
                                              }
        RESPONDER CONSUMER OF         {
                                              activityTestPackage |
                                              chargingInformationPackage
{networkSpecificBoundSet}|
                                              networkCapabilityPackage
{networkSpecificBoundSet}|
                                              notificationPackage
{networkSpecificBoundSet}|
                                              userInformationPackage
{networkSpecificBoundSet}
                                              }
        ID                                    id-contract-scf-scf
        }

dsspContract  CONTRACT ::= {
        CONNECTION                    dsspConnectionPackage {networkSpecificBoundSet}
        INITIATOR CONSUMER OF         {chainedSCFOperationPackage{networkSpecificBoundSet}}
        ID                                    id-contract-dssp
        }

--  Connection Package --

scf-scfConnectionPackage {PARAMETERS-BOUND : bound} CONNECTION-PACKAGE ::= {
        BIND                          scfBind{bound}
        UNBIND                        scfUnbind
        RESPONDER UNBIND              FALSE
        ID                                    id-package-scf-scfConnection
        }

dsspConnectionPackage {PARAMETERS-BOUND : bound} CONNECTION-PACKAGE ::= {
        BIND                          scfBind{bound}
        UNBIND                        scfUnbind
        RESPONDER UNBIND              FALSE
        ID                                    id-package-dsspConnection
        }

--  handlingInformation package --

handlingInformationPackage  {PARAMETERS-BOUND : bound}   OPERATION-PACKAGE ::= {
        CONSUMER INVOKES              {handlingInformationRequest {bound}}
        SUPPLIER INVOKES              {handlingInformationResult {bound}}
        ID                                    id-package-handlingInformation
        }

--  notification package --

notificationPackage  {PARAMETERS-BOUND : bound}   OPERATION-PACKAGE ::= {
        CONSUMER INVOKES              { requestNotification {bound}}
        SUPPLIER INVOKES              { notificationProvided {bound}| confirmedNotificationProvided }
        ID                                    id-package-notification
        }
```

```
--  chargingInformation package --

chargingInformationPackage  {PARAMETERS-BOUND : bound}   OPERATION-PACKAGE ::= {
        CONSUMER INVOKES                    { establishChargingRecord {bound} }
        SUPPLIER INVOKES                    {
        confirmedReportChargingInformation{bound} |
                                                    reportChargingInformation  {bound}
                                                    }
        ID                                          id-package-chargingInformation}

--  userInformation package --

userInformationPackage  {PARAMETERS-BOUND : bound} OPERATION-PACKAGE ::= {
        CONSUMER INVOKES                    {provideUserInformation  {bound}  }
        ID                                          id-package-userInformation
        }

--  networkCapability package --

networkCapabilityPackage  {PARAMETERS-BOUND : bound} OPERATION-PACKAGE ::= {
        CONSUMER INVOKES                    { networkCapability  {bound}}
        ID                                          id-package-networkCapability
        }

--  chainedSCFOperation package --
chainedSCFOperationPackage {PARAMETERS-BOUND : bound} OPERATION-PACKAGE ::= {
        CONSUMER INVOKES              {
                                      chainedHandlingInformationRequest  {bound} |
                                      chainedNotificationProvided { bound}|
                                      chainedConfirmedNotificationProvided {bound}|
                                      chainedReportChargingInformation { bound}|
        chainedConfirmedReportChargingInformation{bound}                          }
        SUPPLIER INVOKES      {
        chainedEstablishChargingRecord { bound}|
        chainedHandlingInformationResult { bound}|
        chainedNetworkCapability { bound}|
        chainedProvideUserInformation { bound}|
        chainedRequestNotification { bound}
        }
        ID                            id-package-chainedSCFOperations
        }

-- abstract syntaxes --
scf-scfOperationsAbstractSyntax  ABSTRACT-SYNTAX ::= {
        BasicSCF-SCF-PDUs
        IDENTIFIED BY                 id-as-scf-scfOperationsAS}

BasicSCF-SCF-PDUs ::= TCMessage {{SCF-SCF-Invokable}, {SCF-SCF-Returnable}}

SCF-SCF-Invokable {PARAMETERS-BOUND} OPERATION ::= {
                                      activityTest |
                                      establishChargingRecord {bound}|
                                      confirmedNotificationProvided {bound}|
                                      confirmedReportChargingInformation {bound} |
                                      handlingInformationRequest {bound}|
                                      handlingInformationResult {bound}|
                                      networkCapability {bound}|
                                      notificationProvided {bound}|
                                      provideUserInformation {bound}|
                                      reportChargingInformation {bound}|
                                      requestNotification {bound}
        }
```

```
SCF-SCF-Returnable {PARAMETERS-BOUND}  OPERATION ::= {
                                        activityTest |
                                        establishChargingRecord {bound}|
                                        confirmedNotificationProvided {bound}|
                                        confirmedReportChargingInformation {bound}|
                                        handlingInformationRequest {bound}|
                                        handlingInformationResult {bound}|
                                        networkCapability {bound}|
                                        provideUserInformation {bound}|
                                        requestNotification {bound}
       }

distributedSCFSystemAbstractSyntax  ABSTRACT-SYNTAX ::= {
       BasicDSSP-PDUs
       IDENTIFIED BY                    id-as-distributedSCFSystemAS}

BasicDSSP-PDUs ::= TCMessage {{DSSP-Invokable}, {DSSP-Returnable}}

DSSP-Invokable {PARAMETERS-BOUND : bound} OPERATION ::= {
                                        chainedHandlingInformationRequest {bound}|
                                        chainedNotificationProvided {bound}|
                                        chainedConfirmedNotificationProvided{bound} |
                                        chainedReportChargingInformation {bound}|
                                        chainedConfirmedReportChargingInformation {bound}

       }

DSSP-Returnable {PARAMETERS-BOUND : bound} OPERATION ::= {
                                        chainedHandlingInformationRequest {bound}|
                                        chainedConfirmedNotificationProvided{bound} |
                                        chainedConfirmedReportChargingInformation {bound}
       }

distributedSCFBindingAbstractSyntax  ABSTRACT-SYNTAX  ::= {
       SCF-SCFBinding-PDUs{networkSpecificBoundSet}
       IDENTIFIED BY                    id-as-scf-scfBindingAS}

SCF-SCFBinding-PDUs{PARAMETERS-BOUND:bound} ::= CHOICE {
       bind                             Bind {scfBind{bound}},
       unbind                           Unbind {scfUnbind}
       }

END
```

10 SCF/CUSF interface

10.1 Operations and arguments

```
IN-CS2-SCF-CUSF-ops-args {itu-t recommendation q 1228 modules(0) in-cs2-scf-cusf-ops-args (15) version1(0)}

DEFINITIONS IMPLICIT TAGS ::=

BEGIN

IMPORTS
```

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

        EXTENSION,
                PARAMETERS-BOUND,
                SupportedExtensions { }
                        FROM IN-CS2-classes classes



        opcode-activationReceivedAndAuthorized,
        opcode-associationReleaseRequested,
        opcode-componentReceived,
        opcode-initiateAssociation,
        opcode-releaseAssociation,
        opcode-requestReportBCUSMEvent,
        opcode-sendComponent
                FROM IN-CS2-operationcodes operationcodes

        BCUSMEvent,
        CalledPartyNumber {},
        CallUnrelatedDpSpecificCommonParameters {},
        Cause {},
        Component,
        ComponentType,
        ComponentCorrelationID,
        Duration,
        ExtensionField {},
        Message,
        OperationCode
                FROM IN-CS2-datatypes datatypes

        missingCustomerRecord,
        missingParameter,
        parameterOutOfRange,
        systemFailure,
        taskRefused,
        unexpectedComponentSequence,
        unexpectedDataValue,
        unexpectedParameter
                FROM IN-CS2-errortypes errortypes

        activityTest
FROM IN-CS2-SSF-SCF-ops-args
{ccitt recommendation q 1228 modules(0) in-cs2-ssf-scf-ops-args (5) version1(0)}


        classes, operationcodes, ros-InformationObjects, datatypes,errortypes
                FROM IN-CS2-object-identifiers
{ccitt recommendation q 1228 modules(0) in-cs2-object-identifiers(17) version1(0)}

;
-- Direction: SCF → CUSF, Timer: Tat
-- This operation is used to check for the continued existence of a relationship between the SCF
-- and CUSF. If the relationship is still in existence, then the CUSF will respond. If no reply is
-- received, then the SCF will assume that the CUSF has failed in some way and will take the
-- appropriate action.
```

```
activationReceivedAndAuthorized  {PARAMETERS-BOUND : bound} OPERATION ::= {
        ARGUMENT                     ActivationReceivedAndAuthorizedArg {bound}
        RETURN RESULT                FALSE
        ERRORS                       {missingCustomerRecord |
                                     missingParameter |
                                     parameterOutOfRange |
                                     systemFailure |
                                     taskRefused |
                                     unexpectedComponentSequence |
                                     unexpectedDataValue |
                                     unexpectedParameter
                                     }
        CODE                         opcode-activationReceivedAndAuthorized
        }

-- Direction: CUSF → SCF, Timer: Tara
-- This operation is used to indicate the desire from an end user to establish an association between the end user
-- and a network (e.g. Q.932 REGISTER message), and the authority/ability to establish the association is
-- verified (BCUSM DP – Activation Received And Authorized). As the association request can have a request to
-- invoke an operation between the user and the network, this operation optionally indicates the component of
-- the operation to the SCF.

ActivationReceivedAndAuthorizedArg  {PARAMETERS-BOUND : bound} ::= SEQUENCE{
        callUnrelatedDpSpecificCommonParameters [0] CallUnrelatedDpSpecificCommonParameters {bound},
        componentType                [1] ComponentType                         OPTIONAL,
        componentCorrelationID       [3] ComponentCorrelationID                OPTIONAL,
        extensions                   [4] SEQUENCE SIZE(1..bound.&numOfExtensions) OF
                                         ExtensionField {bound}                OPTIONAL,
        component                    [5] Component                             OPTIONAL,
        ...
        }

associationReleaseRequested {PARAMETERS-BOUND : bound} OPERATION ::= {
        ARGUMENT                     AssociationReleaseRequestedArg {bound}
        RETURN RESULT                FALSE
        ERRORS                       {missingCustomerRecord |
                                     missingParameter |
                                     parameterOutOfRange |
                                     systemFailure |
                                     taskRefused |
                                     unexpectedComponentSequence |
                                     unexpectedDataValue |
                                     unexpectedParameter
                                     }
        CODE                         opcode-associationReleaseRequested
        }

-- Direction: CUSF → SCF, Timer: Tarr
-- This operation is issued by the CUSF for reporting the TDP/EDP event to the SCF that a
-- request of association release
-- with optionally an operation invocation request or an response/error has been received, and criteria for the
-- AssociationReleasedRequested DP were met.
```

```
AssociationReleaseRequestedArg {PARAMETERS-BOUND : bound} ::= SEQUENCE {
        callUnrelatedDpSpecificCommonParameters  [0] CallUnrelatedDpSpecificCommonParameters {bound},
        componentType                    [1] ComponentType                         OPTIONAL,
        componentCorrelationID           [3] ComponentCorrelationID                OPTIONAL,
        extensions                       [4] SEQUENCE SIZE(1..bound.&numOfExtensions)
                                             OF ExtensionField {bound}             OPTIONAL,
        component                        [5] Component                             OPTIONAL,
        ...
        }

componentReceived  {PARAMETERS-BOUND : bound}  OPERATION ::= {
        ARGUMENT                         ComponentReceivedArg {bound}
        RETURN RESULT                    FALSE
        ERRORS                           {missingCustomerRecord |
                                         missingParameter |
                                         parameterOutOfRange |
                                         systemFailure |
                                         taskRefused |
                                         unexpectedComponentSequence |
                                         unexpectedDataValue |
                                         unexpectedParameter
                                         }
        CODE                             opcode-componentReceived
        }
-- Direction: CUSF → SCF, Timer: Tcre
-- This operation is used to indicate the reception of invocation of an operation or return result/return error/reject
-- from an end user to the network.  This event is the previously requested EDP with RequestReportBCUSMEvent
-- operation for all cases or the TDP if the new invocation meets the criteria for the ComponentReceived DP.
-- The received result may be correlated with previously delivered invocation/result to the user with
-- the RequestReportBCUSMEvent and SendComponent operation.
-- Note that the multiple points of control is not allowed for the bearer unrelated interaction, and TDP is allowed
-- if there is no control relationship between the SCF and the CUSF.  This is the same as the SCF-SSF case.

ComponentReceivedArg  {PARAMETERS-BOUND : bound} ::= SEQUENCE {
        callUnrelatedDpSpecificCommonParameters  [0] CallUnrelatedDpSpecificCommonParameters {bound},
        componentType                    [1] ComponentType                         OPTIONAL,
        componentCorrelationID           [3] ComponentCorrelationID                OPTIONAL,
        extensions                       [4] SEQUENCE SIZE(1..bound.&numOfExtensions)
                                             OF ExtensionField {bound}             OPTIONAL,
        component                        [5] Component                             OPTIONAL,
        ...
        }

initiateAssociation  {PARAMETERS-BOUND : bound}  OPERATION        ::= {
        ARGUMENT                         InitiateAssociationArg {bound}
        RETURN RESULT                    FALSE
        ERRORS                           {missingParameter |
                                         parameterOutOfRange |
                                         systemFailure |
                                         taskRefused |
                                         unexpectedComponentSequence |
                                         unexpectedDataValue |
                                         unexpectedParameter
                                         }
        CODE                             opcode-initiateAssociation
        }
```

```
-- Direction: SCF → CUSF, Timer: Tia
-- This operation is used for allowing the SCF to initiate a call unrelated association with the user.
-- The subsequent operations can be sent in the same TCAP message in the following order:
--        – the RequestReportBCUSMEvent operation if an answer from the CUSF is expected
--        – the SendComponent operation

InitiateAssociationArg  {PARAMETERS-BOUND : bound} ::= SEQUENCE {
        calledPartyNumber         [0]     CalledPartyNumber {bound},
        extensions                [1]     SEQUENCE SIZE(1..bound.&numOfExtensions)     OF
                                          ExtensionField {bound}                  OPTIONAL,
        ...
        }

releaseAssociation  {PARAMETERS-BOUND : bound}  OPERATION ::= {
        ARGUMENT                  ReleaseAssociationArg {bound}
        RETURN RESULT             FALSE
        ALWAYS RESPONDS           FALSE
        CODE                      opcode-releaseAssociation
        }

--  Direction: SCF → CUSF, Timer: Trel
--  This operation is used to indicate the CUSF to release the existing association between the user and the
--  network, during the BCUSM suspended at a DP.

ReleaseAssociationArg  {PARAMETERS-BOUND : bound} ::= Cause {bound}

requestReportBCUSMEvent  {PARAMETERS-BOUND : bound}  OPERATION ::= {
        ARGUMENT                  RequestReportBCUSMEventArg {bound}
        RETURN RESULT             FALSE
        ERRORS                    {missingParameter |
                                  parameterOutOfRange |
                                  systemFailure |
                                  taskRefused |
                                  unexpectedComponentSequence |
                                  unexpectedDataValue |
                                  unexpectedParameter
                                  }
        CODE                      opcode-requestReportBCUSMEvent
        }

-- Direction: SCF → CUSF, Timer: Trrbce
-- This operation is used to request the CUSF to report the reception of invocation of an operation or return result/reject
-- from the end user to the SCF. The requesting event can be either the result, return error/reject from the end user as the
-- response for the SCF specified invocation/result with the SendComponent operation
-- or the independent invocation/result error from the end user.

RequestReportBCUSMEventArg  {PARAMETERS-BOUND : bound} ::= SEQUENCE{
        bcusmEvents               [0] SEQUENCE SIZE(1..bound.&numOfBCUSMEvents) OF BCUSMEvent,
        componentTypes            [1] SEQUENCE SIZE(1..3) OF ComponentType DEFAULT {any},
        componentCorrelationID    [2] ComponentCorrelationID                OPTIONAL,
        monitorDuration           [3] Duration                              OPTIONAL,
        extensions                [4] SEQUENCE SIZE(1..bound.&numOfExtensions)
                                      OF ExtensionField {bound}             OPTIONAL,
        ...
        }
```

```
sendComponent  {PARAMETERS-BOUND : bound}  OPERATION ::= {
        ARGUMENT                SendComponentArg {bound}
        RETURN RESULT           FALSE
        ERRORS                  {missingParameter |
                                parameterOutOfRange |
                                systemFailure |
                                taskRefused |
                                unexpectedComponentSequence |
                                unexpectedDataValue |
                                unexpectedParameter
                                }
        CODE                    opcode-sendComponent
        }

-- Direction: SCF → CUSF, Timer: Tsdc
-- This operation is used to send a component to the user during the BCUSM suspended at a DP.

SendComponentArg  {PARAMETERS-BOUND : bound} ::= SEQUENCE{
        componentType           [0] ComponentType,
        componentCorrelationID  [2] ComponentCorrelationID          OPTIONAL,
        message                 [3] Message                         DEFAULT   rELeaseCOMPlete,
        monitorDuration         [4] Duration                        OPTIONAL,
        extensions              [5] SEQUENCE SIZE(1..bound.&numOfExtensions)         OF
                                    ExtensionField  {bound}         OPTIONAL,
        component               [6] Component                       OPTIONAL,
        ...
        }

END
```

Table 10-1 below lists all operation timers and the value range for each timer. The definitive value for each operation timer may be network-specific and has to be defined by the network operator.

NOTE – The following value ranges do apply for operation specific timers in INAP:

short:	1-10 seconds.
medium:	1-60 seconds.
long:	1 second-30 minutes.
ffs:	For Further Study.

Table 10-1/Q.1228 – Operation timers and their value range

Operation name	Timer	Value range
activationReceivedAndAuthorized	T_{ara}	Short
associationReleaseRequested	T_{arr}	Short
componentReceived	T_{cre}	Short
initiateAssociation	T_{ia}	Short
releaseAssociation	T_{rel}	Short
requestReportBCUSMEvent	T_{rrbce}	Short
sendComponent	T_{sdc}	Short

10.2 SCF/CUSF Contracts, Operation Packages, and Application Contexts

10.2.1 Protocol overview

The **cusf-scf-contract** expresses the form of the service in which the CUSF, a ROS-object of class **cusf**, initiates the contract. A ROS-object of class **scf** responds in this contract.

```
cusf-scf-contract  CONTRACT ::= {
        CONNECTION              emptyConnectionPackage
                INITIATOR CONSUMER OF    {basic-cusf-scf-package {networkSpecificBoundSet}}
                RESPONDER CONSUMER OF           {activityTestPackage}
                ID                              id-contract-cusf-scf}
```

The **scf-cusf-contract** expresses the form of the service in which the SCF, a ROS-object of class **scf**, initiates the contract. A ROS-object of class **cusf** responds in this contract.

```
scf-cusf-contract  CONTRACT ::= {
        CONNECTION              emptyConnectionPackage
                INITIATOR CONSUMER OF    {basic-scf-cusf-package {networkSpecificBoundSet}|
activityTestPackage}
                ID                              id-contract-scf-cusf}
```

The **cusf-scf-contract** is composed of an operation package, **basic-cusf-scf-package**.

The **scf-cusf-contract** is composed of an operation package, **basic-scf-cusf-package**.

These operation packages are defined as information objects of class OPERATION-PACKAGE. The operations of these packages are defined in 10.1.

```
basic-cusf-scf-package  OPERATION-PACKAGE ::= {
CONSUMER INVOKES   {activationReceivedAndAuthorized |
                        componentReceived | associationReleaseRequested}

SUPPLIER INVOKES        {sendComponent | releaseAssociation |
                        requestReportBCUSMEvent}

ID                      id-package-basic-cusf-scf}

basic-scf-cusf-package  OPERATION-PACKAGE ::= {

CONSUMER INVOKES   {initiateAssociation | sendComponent |
                        releaseAssociation | requestReportBCUSMEvent}

SUPPLIER INVOKES        {componentReceived | associationReleaseRequested}
ID                      id-package-basic-scf-cusf}
```

Abstract syntax

This version of the INAP requires the support of two types of abstract syntaxes:

a) the abstract syntax of TC dialogue control protocol data units, **dialogue-abstract-syntax,** which is needed to establish the dialogue between FEs and specified in Recommendation Q.773.;

b) the abstract syntax for conveying the protocol data units for invoking the operations involved in the operation packages specified as above and reporting their outcome.

The ASN.1 type from which the values of the last abstract syntax are derived is specified using the parameterized types **TCMessage{}** defined in Recommendation Q.773.

All these abstract syntaxes shall (as a minimum) be encoded according to the Basic ASN.1 encoding rules with the restrictions listed in Recommendation Q.773.

The CUSF-SCF INAP ASEs that realize the operation packages specified as above and the emptyConnectionPackage specified in 4.5 share the following two abstract syntaxes. They are specified as information objects of the class ABSTRACT-SYNTAX.

```
cusf-scf-abstract-syntax  ABSTRACT-SYNTAX ::= {
        BASIC-CUSF-SCF-PDUs
        IDENTIFIED BY    id-as-basic-cusf-scf}

BASIC-CUSF-SCF-PDUs ::= TCMessage {{CUSF-SCF-Invokable}, {CUSF-SCF-Returna ble}}

CUSF-SCF-Invokable  OPERATION ::=     {activationReceivedAndAuthorized
                                      {networkSpecificBoundSet} | activityTest|
                                      componentReceived {networkSpecificBoundSet} |
                                      releaseAssocation {networkSpecificBoundSet} |
                                      requestReportBCUSMEvent
                                      {networkSpecificBoundSet} |
                                      sendComponent {networkSpecificBoundSet} |
                                      associationReleaseRequested {networkSpecificBoundSet}}

CUSF-SCF-Returnable OPERATION ::=     {activationReceivedAndAuthorized
                                      {networkSpecificBoundSet} | activityTest|
                                      componentReceived {networkSpecificBoundSet} |
                                      requestReportBCUSMEvent
                                      {networkSpecificBoundSet} |
                                      sendComponent {networkSpecificBoundSet}
                                      | associationReleaseRequested
                                      {networkSpecificBoundSet}}

scf-cusf-abstract-syntax  ABSTRACT-SYNTAX ::= {
        BASIC-SCF-CUSF-PDUs
        IDENTIFIED BY                 id-as-basic-scf-cusf}

BASIC-SCF-CUSF-PDUs ::= TCMessage {{SCF-CUSF-Invokable}, {SCF-CUSF-Returnable}}

SCF-CUSF-Invokable  OPERATION ::=     {activationReceivedAndAuthorized
                                      {networkSpecificBoundSet} | activityTest|
                                      componentReceived  {networkSpecificBoundSet}|
                                      releaseAssociation {networkSpecificBoundSet} |
                                      requestReportBCUSMEvent
                                      {networkSpecificBoundSet} | sendComponent
                                      {networkSpecificBoundSet} | initiateAssociation
                                      {networkSpecificBoundSet} |
                                      associationReleaseRequested
                                      {networkSpecificBoundSet}}

SCF-CUSF-Returnable OPERATION ::=     {activationReceivedAndAuthorized
                                      {networkSpecificBoundSet} | activityTest|
                                      componentReceived {networkSpecificBoundSet} |
                                      requestReportBCUSMEvent
                                      {networkSpecificBoundSet} | sendComponent
                                      {networkSpecificBoundSet}
                                      | initiateAssociation {networkSpecificBoundSet} |
                                      associationReleaseRequested
                                      {networkSpecificBoundSet}}
```

Application Contexts

The **cusf-scf-contract** is realized by an application context, **cusf-scf-ac**, and the **scf-cusf-contract** is realized by an application context, **scf-cusf-ac**. These application contexts are specified as information objects of the class APPLICATION-CONTEXT.

```
        cusf-scf-ac  APPLICATION-CONTEXT ::= {
                CONTRACT                        cusf-scf-contract
                DIALOGUE MODE                   structured
                TERMINATION                     basic
                ABSTRACT SYNTAXES               {dialogue-abstract-syntax | cusf-scf-abstract-syntax}
                APPLICATION CONTEXT NAME  id-ac-cusf-scf}

        scf-cusf-ac  APPLICATION-CONTEXT ::= {
                CONTRACT                        scf-cusf-contract
                DIALOGUE MODE                   structured
                TERMINATION                     basic
                ABSTRACT SYNTAXES               {dialogue-abstract-syntax | scf-cusf-abstract-syntax}
                APPLICATION CONTEXT NAME  id-ac-scf-cusf}
```

10.2.2 ASN.1 module

```
IN-CS2-SCF-CUSF-pkgs-contracts-acs  {itu-t recommendation q 1228 modules(0) in-cs2-scf-cusf-pkgs-contracts-
acs (16) version1(0)}

DEFINITIONS ::=

BEGIN

-- This module describes the operation-packages, contracts and application-contexts used
--  over the SCF-CUSF interface.

IMPORTS

                emptyConnectionPackage,
        PARAMETERS-BOUND,
        networkSpecificBoundSet
FROM IN-CS2-classes classes


        CONTRACT, OPERATION-PACKAGE, OPERATION
FROM Remote-Operations-Information-Objects ros-InformationObjects

        TCMessage {}
                FROM TCAPMessages tc-Messages

        APPLICATION-CONTEXT, dialogue-abstract-syntax
FROM TC-Notation-Extensions tc-NotationExtensions


        activationReceivedAndAuthorized {},
        associationReleaseRequested {},
        componentReceived {},
        releaseAssociation {},
        requestReportBCUSMEvent {},
        sendComponent {},
        initiateAssociation {}
```

```
FROM IN-CS2-SCF-CUSF-ops-args scf-cusf-Operations

        id-ac-cusf-scf,
        id-ac-scf-cusf,
        id-contract-scf-cusf,
        id-contract-cusf-scf,
        id-package-basic-cusf-scf,
        id-package-basic-scf-cusf,
        id-as-basic-cusf-scf,
        id-as-basic-scf-cusf,
        classes, ros-InformationObjects, tc-Messages, scf-cusf-Operations, tc-NotationExtensions,
        ssf-scf-Protocol, ssf-scf-Operations

FROM IN-CS2-object-identifiers {itu-t recommendation q 1228 modules(0) in-cs2-object-identifiers (17)
version1(0)}

        activityTestPackage

FROM IN-CS2-SSF-SCF-pkgs-contracts-acs ssf-scf-Protocol

        activityTest

FROM IN-CS2-SSF-SCF-ops-args ssf-scf-Operations
;

-- application contexts --

cusf-scf-ac                             APPLICATION-CONTEXT ::=        {
        CONTRACT                        cusf-scf-contract
        DIALOGUE MODE                   structured
        TERMINATION                     basic
        ABSTRACT SYNTAXES               {dialogue-abstract-syntax |
                                        cusf-scf-abstract-syntax }
        APPLICATION CONTEXT NAME        id-ac-cusf-scf }


scf-cusf-ac                             APPLICATION-CONTEXT ::=        {
        CONTRACT                        scf-cusf-contract
        DIALOGUE MODE                   structured
        TERMINATION                     basic
        ABSTRACT SYNTAXES               {dialogue-abstract-syntax |
                                        scf-cusf-abstract-syntax }
        APPLICATION CONTEXT NAME        id-ac-scf-cusf}

-- contracts --

cusf-scf-contract                       CONTRACT ::=
        {CONNECTION                     emptyConnectionPackage
        INITIATOR CONSUMER OF           {basic-cusf-scf-package {networkSpecificBoundSet}}
        RESPONDER CONSUMER OF           {activityTestPackage}
        ID                              id-contract-scf-cusf }


scf-cusf-contract                       CONTRACT ::=
        {CONNECTION                     emptyConnectionPackage
        INITIATOR CONSUMER OF           {basic-scf-cusf-package {networkSpecificBoundSet}|
                                        activityTestPackage}
        ID                              id-contract-cusf-scf}
```

```
-- basic cusf-scf package --

basic-cusf-scf-package          {PARAMETERS-BOUND : bound} OPERATION-PACKAGE        ::=
        {CONSUMER INVOKES                 {activationReceivedAndAuthorized  {bound} |
                                          componentReceived {bound}|
                                          associationReleaseRequested {bound}}
        SUPPLIER INVOKES                  {sendComponent {bound}|
                                          releaseAssociation {bound}|
                                          requestReportBCUSMEvent {bound}}
        ID                                id-package-basic-cusf-scf }

--  basic scf-cusf package --

basic-scf-cusf-package          {PARAMETERS-BOUND : bound} OPERATION-PACKAGE        ::=
        {CONSUMER INVOKES                 {initiateAssociation {bound}|
                                          sendComponent {bound}|
                                          releaseAssociation {bound}|
                                          requestReportBCUSMEvent {bound}}
        SUPPLIER INVOKES                  {componentReceived {bound} |
                                          associationReleaseRequested {bound}}
        ID                                id-package-basic-scf-cusf }


-- abstract syntaxes --

cusf-scf-abstract-syntax                  ABSTRACT-SYNTAX ::= {
        BASIC-CUSF-SCF-PDUs
        IDENTIFIED BY                     id-as-basic-cusf-scf}

BASIC-CUSF-SCF-PDUs ::= TCMessage {{CUSF-SCF-Invokable},{CUSF-SCF-Returnable} }

CUSF-SCF-Invokable          OPERATION ::= {activationReceivedAndAuthorized {networkSpecificBoundSet}|
                                          activityTest|
                                          componentReceived  {networkSpecificBoundSet}|
                                          releaseAssociation  {networkSpecificBoundSet}|
                                          requestReportBCUSMEvent {networkSpecificBoundSet}  |
                                          sendComponent  {networkSpecificBoundSet} |
                                          associationReleaseRequested {networkSpecificBoundSet}
                                          }

CUSF-SCF-Returnable         OPERATION ::= {activationReceivedAndAuthorized  {networkSpecificBoundSet}
                                          | activityTest|
                                          componentReceived  {networkSpecificBoundSet}|
                                          requestReportBCUSMEvent {networkSpecificBoundSet} |
                                          sendComponent  {networkSpecificBoundSet}|
                                          associationReleaseRequested {networkSpecificBoundSet}
                                          }


scf-cusf-abstract-syntax    ABSTRACT-SYNTAX::=
        {BASIC-SCF-CUSF-PDUs
        IDENTIFIED BY                     id-as-basic-scf-cusf}

BASIC-SCF-CUSF-PDUs ::= TCMessage {{SCF-CUSF-Invokable},{SCF-CUSF-Returnable} }
```

```
SCF-CUSF-Invokable       OPERATION   ::= {activationReceivedAndAuthorized {networkSpecificBoundSet}|
                                      activityTest|
                                      componentReceived  {networkSpecificBoundSet}|
                                      releaseAssociation  {networkSpecificBoundSet}|
                                      requestReportBCUSMEvent {networkSpecificBoundSet} |
                                      sendComponent  {networkSpecificBoundSet}|
                                      initiateAssociation  {networkSpecificBoundSet}|
                                      associationReleaseRequested {networkSpecificBoundSet}}

SCF-CUSF-Returnable      OPERATION   ::= {activationReceivedAndAuthorized {networkSpecificBoundSet}|
                                      activityTest|
                                      componentReceived  {networkSpecificBoundSet}|
                                      requestReportBCUSMEvent {networkSpecificBoundSet} |
                                      sendComponent  {networkSpecificBoundSet}|
                                      initiateAssociation  {networkSpecificBoundSet}|
                                      associationReleaseRequested {networkSpecificBoundSet}}


END
```

ITU-T RECOMMENDATIONS SERIES

Series A	Organization of the work of the ITU-T
Series B	Means of expression: definitions, symbols, classification
Series C	General telecommunication statistics
Series D	General tariff principles
Series E	Overall network operation, telephone service, service operation and human factors
Series F	Non-telephone telecommunication services
Series G	Transmission systems and media, digital systems and networks
Series H	Audiovisual and multimedia systems
Series I	Integrated services digital network
Series J	Transmission of television, sound programme and other multimedia signals
Series K	Protection against interference
Series L	Construction, installation and protection of cables and other elements of outside plant
Series M	TMN and network maintenance: international transmission systems, telephone circuits, telegraphy, facsimile and leased circuits
Series N	Maintenance: international sound programme and television transmission circuits
Series O	Specifications of measuring equipment
Series P	Telephone transmission quality, telephone installations, local line networks
Series Q	**Switching and signalling**
Series R	Telegraph transmission
Series S	Telegraph services terminal equipment
Series T	Terminals for telematic services
Series U	Telegraph switching
Series V	Data communication over the telephone network
Series X	Data networks and open system communications
Series Y	Global information infrastructure
Series Z	Programming languages

Printed in Switzerland

Geneva, 1998

Printed in Switzerland

Geneva, 1998

Printed in Switzerland
Geneva, 1998