



INTERNATIONAL TELECOMMUNICATION UNION

**ITU-T**

TELECOMMUNICATION  
STANDARDIZATION SECTOR  
OF ITU

**Q.1248.1**

(07/2001)

SERIES Q: SWITCHING AND SIGNALLING

Intelligent Network

---

**Interface Recommendation for Intelligent  
Network Capability Set 4: Common aspects**

ITU-T Recommendation Q.1248.1

---

# ITU-T Q-SERIES RECOMMENDATIONS

## SWITCHING AND SIGNALLING

SIGNALLING IN THE INTERNATIONAL MANUAL SERVICE	Q.1–Q.3
INTERNATIONAL AUTOMATIC AND SEMI-AUTOMATIC WORKING	Q.4–Q.59
FUNCTIONS AND INFORMATION FLOWS FOR SERVICES IN THE ISDN	Q.60–Q.99
CLAUSES APPLICABLE TO ITU-T STANDARD SYSTEMS	Q.100–Q.119
SPECIFICATIONS OF SIGNALLING SYSTEM No. 4	Q.120–Q.139
SPECIFICATIONS OF SIGNALLING SYSTEM No. 5	Q.140–Q.199
SPECIFICATIONS OF SIGNALLING SYSTEM No. 6	Q.250–Q.309
SPECIFICATIONS OF SIGNALLING SYSTEM R1	Q.310–Q.399
SPECIFICATIONS OF SIGNALLING SYSTEM R2	Q.400–Q.499
DIGITAL EXCHANGES	Q.500–Q.599
INTERWORKING OF SIGNALLING SYSTEMS	Q.600–Q.699
SPECIFICATIONS OF SIGNALLING SYSTEM No. 7	Q.700–Q.799
Q3 INTERFACE	Q.800–Q.849
DIGITAL SUBSCRIBER SIGNALLING SYSTEM No. 1	Q.850–Q.999
PUBLIC LAND MOBILE NETWORK	Q.1000–Q.1099
INTERWORKING WITH SATELLITE MOBILE SYSTEMS	Q.1100–Q.1199
<b>INTELLIGENT NETWORK</b>	<b>Q.1200–Q.1699</b>
SIGNALLING REQUIREMENTS AND PROTOCOLS FOR IMT-2000	Q.1700–Q.1799
SPECIFICATIONS OF SIGNALLING RELATED TO BEARER INDEPENDENT CALL CONTROL (BICC)	Q.1900–Q.1999
BROADBAND ISDN	Q.2000–Q.2999

*For further details, please refer to the list of ITU-T Recommendations.*

# ITU-T Recommendation Q.1248.1

## Interface Recommendation for Intelligent Network Capability Set 4: Common aspects

### Summary

The Q.1248.x family of ITU-T Recommendations defines the Intelligent Network (IN) Application Protocol (INAP) for IN Capability Set 4 (IN CS-4). This definition is based upon IN CS-3 Q.1238.x (2000) family of ITU-T Recommendations and upon the general rules for INAP provided in ITU-T Rec. Q.1208, and is consistent with the scope of IN CS-4 as defined in ITU-T Rec. Q.1241.

In addition to the features supporting IN CS-3, the Q.1248.x family Recommendations provides general extensions to the CS-3 INAP in support of IN CS-4 target services;

Within the Q.124x series of ITU-T Recommendations, the Q.1248.x family of ITU-T Recommendations describes the protocol realizing the Q.1241 Distributed Functional Plane (DFP) in a service and vendor implementation independent manner, as constrained by the capabilities of the embedded base of network technology. This provides the flexibility to allocate distributed functionality into multiple physical network configurations and to evolve IN from IN CS-4 to some future CS-N.

The Q.1248.x family of ITU-T Recommendations describes the Intelligent Network modelling of each Functional Entity (FE) and their relationships as defined as part of the DFP. It also defines the INAP (Intelligent Network Application Protocol) required for the communication of functional entities when they are located in different physical entities.

Intelligent Network Capability Set 4 (IN CS-4) supports the following functional entities:

- Service Switching Function (SSF);
- Service Control Function (SCF);
- Specialized Resource Function (SRF);
- Service Data Function (SDF);
- Call-Unrelated Service Function (CUSF);
- Service Management Function (SMF).

Each of the interfaces defined between the functional entities is defined in a separate Recommendation in the Q.1248.x series:

- ITU-T Rec. Q.1248.1 Describes the common aspects of IN including an overview of each FE. It also describes aspects of the protocol that are common to all interfaces, including services assumed from lower layers, IN security information and the common data types used on all the interfaces.

The interfaces and the description of the functional entities in relation to those interfaces are defined in:

- ITU-T Rec. Q.1248.2 SSF-SCF interface.
- ITU-T Rec. Q.1248.3 SCF-SRF interface.
- ITU-T Rec. Q.1248.4 SCF-SDF interface.
- ITU-T Rec. Q.1248.5 SDF-SDF interface.

- ITU-T Rec. Q.1248.6 SCF-SCF interface.
- ITU-T Rec. Q.1248.7 SCF-CUSF interface.

This Recommendation includes an electronic attachment containing the ASN.1 definitions for the CS-4 common aspects.

### **Source**

ITU-T Recommendation Q.1248.1 was prepared by ITU-T Study Group 11 (2001-2004) and approved under the WTSA Resolution 1 procedure on 13 July 2001.

### **Keywords**

IN CS-4, INAP.

## FOREWORD

The International Telecommunication Union (ITU) is the United Nations specialized agency in the field of telecommunications. The ITU Telecommunication Standardization Sector (ITU-T) is a permanent organ of ITU. ITU-T is responsible for studying technical, operating and tariff questions and issuing Recommendations on them with a view to standardizing telecommunications on a worldwide basis.

The World Telecommunication Standardization Assembly (WTSA), which meets every four years, establishes the topics for study by the ITU-T study groups which, in turn, produce Recommendations on these topics.

The approval of ITU-T Recommendations is covered by the procedure laid down in WTSA Resolution 1.

In some areas of information technology which fall within ITU-T's purview, the necessary standards are prepared on a collaborative basis with ISO and IEC.

## NOTE

In this Recommendation, the expression "Administration" is used for conciseness to indicate both a telecommunication administration and a recognized operating agency.

## INTELLECTUAL PROPERTY RIGHTS

ITU draws attention to the possibility that the practice or implementation of this Recommendation may involve the use of a claimed Intellectual Property Right. ITU takes no position concerning the evidence, validity or applicability of claimed Intellectual Property Rights, whether asserted by ITU members or others outside of the Recommendation development process.

As of the date of approval of this Recommendation, ITU had not received notice of intellectual property, protected by patents, which may be required to implement this Recommendation. However, implementors are cautioned that this may not represent the latest information and are therefore strongly urged to consult the TSB patent database.

© ITU 2002

All rights reserved. No part of this publication may be reproduced, by any means whatsoever, without the prior written permission of ITU.

# CONTENTS

	<b>Page</b>
1 Scope.....	1
2 References.....	1
2.1 Normative references.....	1
2.2 Bibliography .....	4
3 Abbreviations and acronyms .....	5
4 Scope of IN Distributed Functional Plane for Capability Set 4.....	10
4.1 End-user access.....	11
4.2 Call-related service invocation and control .....	11
4.3 End-user interaction.....	12
4.4 IN service management functionality.....	12
4.5 Call Party Handling .....	12
4.6 Internetworking .....	13
4.7 Security .....	13
4.8 Out-Channel User Interaction.....	13
4.9 Call-unrelated service invocation and control .....	13
4.10 Feature Interaction .....	13
4.11 Distributed Functional Model.....	13
4.12 Communication model .....	14
5 FE models .....	15
5.1 Call Control Function/Service Switching Function (CCF/SSF) models.....	15
5.1.1 General.....	15
5.1.2 CCF/SSF components .....	18
5.1.3 CCF/SSF trigger information objects.....	19
5.1.4 SSF FSM structure .....	20
5.2 Specialized Resource Function (SRF) models .....	22
5.2.1 General.....	22
5.2.2 SRF components.....	23
5.2.3 SRF functional resources.....	23
5.2.4 SRF FSM structure .....	24
5.3 Service Control Function (SCF) models .....	25
5.3.1 General.....	25
5.3.2 SCF components.....	26
5.3.3 The SCF FSM structure.....	29
5.4 Service Data Function (SDF) models.....	31
5.4.1 General.....	31
5.4.2 SDF components .....	32
5.4.3 Data types handled by SDF .....	33

	<b>Page</b>
5.4.4 SDF FSM structure .....	33
5.5 Call Control Function/Call-Unrelated Service Function (CCF/CUSF) models .....	34
5.5.1 General .....	34
5.5.2 CCF/CUSF components .....	35
5.5.3 Relationship of CCF/CUSF model components .....	36
5.5.4 CUSF trigger information objects .....	36
5.5.5 CUSF FSM structure .....	37
5.6 Service Management Function (SMF) models .....	38
5.6.1 General .....	38
5.6.2 SMF components .....	39
6 Use of FE relationships .....	40
6.1 SCF-SSF relationship .....	40
6.2 SCF-SCF relationship .....	40
6.3 SCF-IAF relationship .....	40
6.4 SRF-CCF relationship .....	40
6.5 SRF-SCF relationship .....	40
6.6 SCF-SDF relationship .....	40
6.7 SDF-SDF relationship .....	41
6.8 CUSF-SCF relationship .....	41
6.9 CUSF-SSF relationship .....	41
6.10 SMF-SCF relationship .....	41
6.11 SMF-SDF relationship .....	41
6.12 SMF-CCF/SSF relationship .....	41
6.13 SMF-SRF relationship .....	42
6.14 SMF-SMAF relationship .....	42
6.15 SMF-SCEF relationship .....	42
6.16 SMF-SMF relationship .....	42
6.17 SMF-CCF/CUSF relationship .....	43
7 Protocol realization .....	43
7.1 Overview .....	43
7.2 Application contexts .....	43
7.3 Abstract syntax and transfer syntax .....	44
7.4 SACF/MACF rules .....	44
7.4.1 Reflection of TCAP AC .....	44
7.4.2 Sequential/Parallel execution of operations .....	44
8 Congestion control .....	45
8.1 Lower layer flow control in a SS7 environment .....	45
8.2 Application level flow control .....	46

	<b>Page</b>
9 Protocol mechanisms .....	46
9.1 Compatibility mechanisms and extensibility rules .....	46
9.1.1 Introduction .....	46
9.1.2 Definition of INAP compatibility mechanisms .....	47
9.2 IN generic interface security.....	47
9.2.1 Interface security requirements .....	47
9.2.2 INAP screening requirements.....	48
9.2.3 Security procedures and algorithms .....	49
9.2.4 Mapping of Security Information Flow Definitions to Tokens.....	50
9.2.5 Security FSM definitions.....	50
10 Services assumed from lower layers.....	50
10.1 Services assumed from TCAP.....	50
10.1.1 Common procedures.....	51
10.2 Services assumed from SCCP .....	61
10.2.1 Normal procedures .....	61
10.2.2 Service functions from SCCP.....	61
11 Error definitions.....	64
11.1 AttributeError .....	64
11.1.1 Error description .....	64
11.1.2 Parameter description .....	64
11.1.3 Relevant interfaces .....	64
11.2 Cancelled .....	64
11.2.1 Error description .....	64
11.2.2 Parameter description .....	64
11.2.3 Relevant interface .....	64
11.3 CancelFailed .....	64
11.3.1 Error description .....	64
11.3.2 Parameter description .....	64
11.3.3 Relevant interfaces .....	65
11.4 ChainingRefused .....	65
11.4.1 Error description .....	65
11.4.2 Parameter description .....	65
11.4.3 Relevant interface .....	65
11.5 DirectoryBindError.....	65
11.5.1 Error description .....	65
11.5.2 Parameter description .....	65
11.5.3 Relevant interfaces .....	65
11.6 DSAReferral .....	65
11.6.1 Error description .....	65



	<b>Page</b>
11.6.2 Parameter description .....	65
11.6.3 Relevant interface .....	65
11.7 ETCFailed.....	65
11.7.1 Error description .....	65
11.7.2 Parameter description .....	65
11.7.3 Relevant interface .....	66
11.8 ExecutionError.....	66
11.8.1 Error description .....	66
11.8.2 Parameter description .....	66
11.8.3 Relevant interfaces .....	66
11.9 ImproperCallerResponse .....	66
11.9.1 Error description .....	66
11.9.2 Parameter description .....	66
11.9.3 Relevant interfaces .....	66
11.10 MissingCustomerRecord .....	66
11.10.1 Error description .....	66
11.10.2 Parameter description .....	67
11.10.3 Relevant interfaces .....	67
11.11 MissingParameter .....	67
11.11.1 Error description .....	67
11.11.2 Parameter description .....	67
11.11.3 Relevant interfaces .....	67
11.12 NameError .....	67
11.12.1 Error description .....	67
11.12.2 Parameter description .....	67
11.12.3 Relevant interfaces .....	67
11.13 ParameterOutOfRange.....	67
11.13.1 Error description .....	67
11.13.2 Parameter description .....	67
11.13.3 Relevant interfaces .....	68
11.14 Referral .....	68
11.14.1 Error description .....	68
11.14.2 Parameter description .....	68
11.14.3 Relevant interface .....	68
11.15 RequestedInfoError .....	68
11.15.1 Error description .....	68
11.15.2 Parameter description .....	68
11.15.3 Relevant interface .....	68
11.16 ScfBindFailure.....	68
11.16.1 Error description .....	68

	<b>Page</b>
11.16.2 Parameter description .....	68
11.16.3 Relevant interface .....	68
11.17 ScfReferral .....	69
11.17.1 Error description .....	69
11.17.2 Parameter description .....	69
11.17.3 Relevant interface .....	69
11.18 ScfTaskRefused .....	69
11.18.1 Error description .....	69
11.18.2 Parameter description .....	69
11.18.3 Relevant interfaces .....	69
11.19 SecurityError .....	69
11.19.1 Error description .....	69
11.19.2 Parameter description .....	69
11.19.3 Relevant interfaces .....	69
11.20 ServiceError .....	69
11.20.1 Error description .....	69
11.20.2 Parameter description .....	70
11.20.3 Relevant interfaces .....	70
11.21 ShadowError .....	70
11.21.1 Error description .....	70
11.21.2 Parameter description .....	70
11.21.3 Relevant interface .....	70
11.22 SystemFailure .....	70
11.22.1 Error description .....	70
11.22.2 Parameter description .....	70
11.22.3 Relevant interfaces .....	70
11.23 TaskRefused .....	70
11.23.1 Error description .....	70
11.23.2 Parameter description .....	70
11.23.3 Relevant interfaces .....	71
11.24 TfcBindError .....	71
11.24.1 Error description .....	71
11.24.2 Parameter description .....	71
11.24.3 Relevant interfaces .....	71
11.25 UnavailableResource .....	71
11.25.1 Error description .....	71
11.25.2 Parameter description .....	71
11.25.3 Relevant interface .....	71
11.26 UnexpectedComponentSequence .....	71
11.26.1 Error description .....	71

	<b>Page</b>
11.26.2 Parameter description .....	71
11.26.3 Relevant interfaces .....	71
11.27 UnexpectedDataValue .....	72
11.27.1 Error description .....	72
11.27.2 Parameter description .....	72
11.27.3 Relevant interfaces .....	72
11.28 UnexpectedParameter .....	72
11.28.1 Error description .....	72
11.28.2 Parameter description .....	72
11.28.3 Relevant interfaces .....	72
11.29 UnknownLegID .....	72
11.29.1 Error description .....	72
11.29.2 Parameter description .....	72
11.29.3 Relevant interfaces .....	72
11.30 UnknownResource .....	72
11.30.1 Error description .....	72
11.30.2 Parameter description .....	73
11.30.3 Relevant interface .....	73
11.31 UpdateError .....	73
11.31.1 Error description .....	73
11.31.2 Parameter description .....	73
11.31.3 Relevant interfaces .....	73
Annex A – ASN.1 – Common definitions (electronic attachment) .....	73
Annex B – Overview of IN-CS-4 additions .....	73



# ITU-T Recommendation Q.1248.1

## Interface Recommendation for Intelligent Network Capability Set 4: Common aspects<sup>1</sup>

### 1 Scope

This Recommendation provides a brief introduction to the modelling of IN CS-4 Functional Entities (FE). It also describes aspects of the protocol that are common to all interfaces. This includes physical examples of interface interconnection, services assumed from lower layers (e.g. TCAP), IN security information and the common data types used on all the interfaces.

### 2 References

#### 2.1 Normative references

The following ITU-T Recommendations and other references contain provisions which, through reference in this text, constitute provisions of this Recommendation. At the time of publication, the editions indicated were valid. All Recommendations and other references are subject to revision; all users of this Recommendation are therefore encouraged to investigate the possibility of applying the most recent edition of the Recommendations and other references listed below. A list of the currently valid ITU-T Recommendations is regularly published.

- ITU-T Recommendation E.164/I.331 (1997), *The international public telecommunication numbering plan*.
- ITU-T Recommendation E.410 (1998), *International network management – General information*.
- ITU-T Recommendation F.811 (1996), *Broadband connection-oriented bearer service*.
- ITU-T Recommendation H.225.0 (2000), *Call signalling protocols and media stream packetization for packet-based multimedia communication systems*.
- ITU-T Recommendation H.245 (2001), *Control protocol for multimedia communication*.
- ITU-T Recommendation H.323 (2000), *Packet-based multimedia communications systems*.
- ITU-T Recommendation H.450.3 (1998), *Call diversion supplementary service for H.323*.
- ITU-T Recommendation M.3010 (2000), *Principles for a Telecommunications management network*.
- ITU-T Recommendation M.3320 (1997), *Management requirements framework for the TMN X-Interface*.
- ITU-T Recommendation M.3400 (2000), *TMN management functions*.
- ITU-T Recommendation Q.29 (1988), *Causes of noise and ways of reducing noise in telephone exchanges*.
- ITU-T Recommendation Q.700 (1993), *Introduction to CCITT Signalling System No. 7*.
- ITU-T Recommendation Q.710 (1988), *Simplified MTP version for small systems*.

---

<sup>1</sup> This Recommendation includes an electronic attachment containing the ASN.1 definitions for the CS-4 common aspects.

- ITU-T Recommendation Q.711 (2001), *Functional description of the signalling connection control part.*
- ITU-T Recommendation Q.713 (2001), *Signalling connection control part formats and codes.*
- ITU-T Recommendation Q.714 (2001), *Signalling connection control part procedures.*
- ITU-T Recommendation Q.715 (1996), *Signalling connection control part user guide.*
- ITU-T Recommendations Q.735.x series, *Stage 3 description for community of interest supplementary services using SS No. 7.*
- ITU-T Recommendation Q.762 (1999), *Signalling System No. 7 – ISDN User Part general function of messages and signals.*
- ITU-T Recommendation Q.763 (1999), *Signalling System No. 7 – User Part formats and codes.*
- ITU-T Recommendation Q.765 (2000), *Signalling System No. 7 – Application transport mechanism.*
- ITU-T Recommendation Q.767 (1991), *Application of the ISDN User Part of CCITT Signalling System No. 7 for international ISDN interconnections.*
- ITU-T Recommendation Q.771 (1997), *Functional description of transaction capabilities.*
- ITU-T Recommendation Q.772 (1997), *Transaction capabilities information element definitions.*
- ITU-T Recommendation Q.773 (1997), *Transaction capabilities format and encodings.*
- ITU-T Recommendation Q.774 (1997), *Transaction capabilities procedures.*
- ITU-T Recommendation Q.775 (1997), *Guidelines for using transaction capabilities.*
- ITU-T Recommendation Q.822 (1994), *Stage 1, Stage 2 and Stage 3 description for the Q3 interface – Performance management.*
- ITU-T Recommendation Q.850 (1998), *Usage of cause and location in the Digital Subscriber Signalling System No. 1 and the Signalling System No. 7 ISDN User Part.*
- ITU-T Recommendation Q.931 (1998), *ISDN user-network interface layer 3 specification for basic call control.*
- ITU-T Recommendation Q.932 (1998), *Digital Subscriber Signalling System No. 1 – Generic procedures for the control of ISDN supplementary services.*
- ITU-T Recommendation Q.1208 (1997), *General aspects of the Intelligent Network Application protocol.*
- ITU-T Recommendation Q.1218 (1995), *Interface Recommendation for Intelligent Network CS-1.*
- ITU-T Recommendation Q.1224 (1997), *Distributed functional plane for Intelligent Network Capability Set 2.*
- ITU-T Recommendation Q.1228 (1997), *Interface Recommendation for Intelligent Network Capability Set 2.*
- ITU-T Recommendation Q.1231 (1999), *Introduction to Intelligent Network Capability Set 3.*
- ITU-T Recommendation Q.1236 (1999), *Intelligent Network Capability Set 3 – Management information model requirements and methodology.*

- ITU-T Recommendation Q.1237 (2000), *Extensions to Intelligent Network Capability Set 3 in support of B-ISDN*.
- ITU-T Recommendation Q.1238.1 (2000), *Interface Recommendation for Intelligent Network Capability Set 3: Common aspects*.
- ITU-T Recommendation Q.1238.2 (2000), *Interface Recommendation for Intelligent Network Capability Set 3: SCF-SSF interface*.
- ITU-T Recommendation Q.1238.3 (2000), *Interface Recommendation for Intelligent Network Capability Set 3: SCF-SRF interface*.
- ITU-T Recommendation Q.1238.4 (2000), *Interface Recommendation for Intelligent Network Capability Set 3: SCF-SDF interface*.
- ITU-T Recommendation Q.1238.5 (2000), *Interface Recommendation for Intelligent Network Capability Set 3: SDF-SDF interface*.
- ITU-T Recommendation Q.1238.6 (2000), *Interface Recommendation for Intelligent Network Capability Set 3: SCF-SCF interface*.
- ITU-T Recommendation Q.1238.7 (2000), *Interface Recommendation for Intelligent Network Capability Set 3: SCF-CUSF interface*.
- ITU-T Recommendation Q.1290 (1998), *Glossary of terms used in the definition of Intelligent Networks*.
- ITU-T Recommendation Q.1400 (1993), *Architecture framework for the development of signalling and OA&M protocols using OSI concepts*.
- ITU-T Recommendation Q.1601 (1999), *Signalling System No. 7 – Interaction between N-ISDN and INAP CS-2*.
- ITU-T Recommendation Q.1902.3 (2001), *Bearer Independent Call Control Protocol (CS2) and Signalling System No. 7 – ISDN User Part formats and codes*.
- ITU-T Recommendation Q.2762 (1999), *General functions of messages and signals of the B-ISDN user part (B-ISUP) of Signalling System No. 7*.
- ITU-T Recommendation Q.2763 (1999), *Signalling System No. 7 B-ISDN User Part (B-ISUP) – Formats and codes*.
- ITU-T Recommendations Q.2961 series, *Digital Subscriber Signalling System No. 2 (DSS2) – Additional traffic parameters*.
- ITU-T Recommendation X.500 (2001) | ISO/IEC 9594-1:2001, *Information technology – Open Systems Interconnection – The Directory: Overview of concepts, models and services*.
- ITU-T Recommendation X.501 (2001) | ISO/IEC 9594-2:2001, *Information technology – Open Systems Interconnection – The Directory: Models*.
- ITU-T Recommendation X.509 (2000) | ISO/IEC 9594-8:2001, *Information technology – Open Systems Interconnection – The directory: Public-key and attribute certificate frameworks*.
- ITU-T Recommendation X.511 (2001) | ISO/IEC 9594-3:2001, *Information technology – Open Systems Interconnection – The directory: Abstract service definition*.
- ITU-T Recommendation X.518 (2001) | ISO/IEC 9594-4:2001, *Information technology – Open Systems Interconnection – The directory: Procedures for distributed operations*.
- ITU-T Recommendation X.519 (2001) | ISO/IEC 9594-5:2001, *Information technology – Open Systems Interconnection – The directory: Protocol specifications*.

- ITU-T Recommendation X.525 (2001) | ISO/IEC 9594-9:2001, *Information technology – Open Systems Interconnection – The directory: Replication.*
- ITU-T Recommendation X.680 (1997) | ISO/IEC 8824-1:1998, *Information technology – Abstract Syntax Notation One (ASN.1): Specification of basic notation.*
- ITU-T Recommendation X.681 (1997) | ISO/IEC 8824-2:1998, *Information technology – Abstract Syntax Notation One (ASN.1): Information object specification.*
- ITU-T Recommendation X.682 (1997) | ISO/IEC 8824-3:1998, *Information technology – Abstract Syntax Notation One (ASN.1): Constraint specification.*
- ITU-T Recommendation X.683 (1997) | ISO/IEC 8824-4:1998, *Information technology – Abstract Syntax Notation One (ASN.1): Parameterization of ASN.1 specifications.*
- ITU-T Recommendation X.690 (1997) | ISO/IEC 8825-1:1998, *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.733 (1992) | ISO/IEC 10164-4:1992, *Information technology – Open Systems Interconnection – Systems management: Alarm reporting function.*
- ITU-T Recommendation X.734 (1992) | ISO/IEC 10164-5:1993, *Information technology – Open Systems Interconnection – Systems management: Event report management function.*
- ITU-T Recommendation X.738 (1993) | ISO/IEC 10164-13:1995, *Information technology – Open Systems Interconnection – Systems management: Summarization function.*
- ITU-T Recommendation X.739 (1993) | ISO/IEC 10164-11:1994, *Information technology – Open Systems Interconnection – Systems management: Metric objects and attributes.*
- 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.*
- ITU-T Recommendation Z.100 (1999), *Specification and description language (SDL).*
- ISO 639:1988, *Code for representation of names of languages.*

## 2.2 Bibliography

The documents listed in this clause provide informative background information for the reader and are not normative within this Recommendation.

- IETF RFC 1738 (1994), *Uniform Resource Locators (URL).*
- IETF RFC 2543 (1999), *SIP: Session Initiation Protocol.*



### **3 Abbreviations and acronyms**

This Recommendation uses the following abbreviations:

AAL	ATM Adaptation Layer
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
AESA	ATM End System Address
AOC	Advice Of Charge
APCI	Application Protocol Control Information
APDU	Application Protocol Data Unit
API	Application Process Instance
API	Application Programming Interface
ASE	Application Service Element
ASR	Automatic Speech Recognition
ATM	Asynchronous Transfer Mode
BCC	Basic Call Control
BCM	Basic Call Manager
BCP	Basic Call Process
BCSM	Basic Call State Model
BCUP	Basic Call Unrelated Process
BCUSM	Basic Call Unrelated State Model
BGID	Business Group Identity
BICC	Bearer Independent Call Control
BNCM	Basic Non-Call Manager
BRC	Bearer Controller
BRI	Basic Rate Interface
CAC	Carrier Access Code
CCAF	Call Control Agent Function
CCF	Call Control Function
CDP	Customized Dialling Plan
CDVT	Cell Delay Variation Tolerance
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
CSCV	Call Segment Connection View
CSM	Call Segment Model
CUG	Closed User Group
CUSF	Call-Unrelated Service Function
CUSP	Call-Unrelated Service Point
CVS	Connection View State
DAP	Directory Access Protocol
DET	Determination (charging)
DFP	Distributed Functional Plane
DHA	Dialogue Handler
DLE	Destination Local Exchange
DN	Directory Number
DN	Distinguished Name
DP	Data Part
DP	Detection Point
DPC	Destination Point Code
DSA	Directory System Agent
DSL	Distributed Service Logic
DSP	Directory System Protocol
DSS1	Digital Subscriber Signalling No. 1 Protocol
DSS2	Digital Subscriber Signalling No. 2 Protocol (Broadband)
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
FE	Functional Entity
FEA	Functional Entity Action
FEAM	Functional Entity Access Manager
FIM	Feature Interactions Manager

FPLMTS	Future Public Land Mobile Telecommunications Systems
FRL	Facility Restriction Level
FSM	Finite State Machine
GEN	Generation (charging)
GFP	Global Functional Plane
GSL	Global Service Logic
GSS	Generic Security Service
GT	Global Title
GULS	Generic Upper Layer Security
GVNS	Global Virtual Network Service
HLSIB	High-Level Service Independent Block
IAF	Intelligent Access Function
IBI	Inter BCSM Interface
IEC	International Electrotechnical Commission
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
ISO	International Organization for Standardization
ISUP	Integrated Services Digital Network-User Part
ITU-T	International Telecommunication Union – Telecommunication Standardization Sector
LE	Local Exchange
MACF	Multiple Association Control Function
MPC	Multiple Point of Control
MSR	Message Storage and Retrieval
MTP	Message Transfer Part
NAP	Network Access Point
NCM	Non-Call Manager
NEF	Network Element Function
NFA	Network Functional Architecture
NM	Network Manager

NSAP	Network Service Access Point
OCCRUI	Out-Channel Call-Related User Interaction
OCCUUI	Out-Channel Call-Unrelated User Interaction
OFC	Off-line Charging (billing/accounting information)
OLE	Originating Local Exchange
OLI	Originating Line Information
ONC	On-line Charging (user access information)
OSF	Operations System Function
OSI	Open Systems Interconnection
OUT	Output
PE	Physical Entity
PIC	Point In Call
PM	Personal Mobility
POC	Point Of Control
POI	Point Of Initiation
POR	Point Of Return
POS	Point Of Synchronization
PRI	Primary Rate Interface
PSTN	Public Switched Telephony Network
PTNX	Private Telecommunications Network Exchange
QoS	Quality of Service
RCP	Resource Control Part
RDN	Relative Distinguished Name
REG	Registration (charging)
RFP	Resource Function Part
RLF	Radio Link Function
ROS	Remote Operations
ROSE	Remote Operations Service Element
ROA	Recognized Operating Agency
SACF	Single Association Control Function
SAO	Single Association Object
SAP	Service Access Point
SCCP	Signalling Connection Control Part
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
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	Service Data Function
SDF FSM	Service Data Function Finite State Machine
SDL	Specification and Description Language
SDME	Service Data Function Management Entity
SDP	Service Data Point
SDSM	Service Data Function Call State Model
SDSS	Server Display and Script Services
SESE	Security Exchange Service Element
SF	Service Feature
SIB	Service-Independent Building Block
SIP	Session Initiation Protocol
SL	Service Logic
SLCP	Service Logic Control Program
SLEE	Service Logic Execution Environment
SLEM	Service Logic Execution Manager
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
SPC	Single Point of Control
SRF	Specialized Resource Function
SRF FSM	Specialized Resource Function Finite State Machine
SRME	Specialized Resource Function Management Entity
SRSF	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
SSN	SCCP Subsystem Number
SSP	Service Switching Point
ST	Signalling Termination
STI	Service Trigger Information
STUI	Service-To-User 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
UNI	User-Network Interface
UPT	Universal Personal Telecommunication
URL	Uniform Resource Locator
USI	User Service Information
UTSI	User-To-Service Information
VC	Virtual Channel
VP	Virtual Path
VPN	Virtual Private Network

#### **4 Scope of IN Distributed Functional Plane for Capability Set 4**

The scope of the IN Distributed Functional Plane (DFP) architecture and supporting protocol for IN Capability Set 4 (IN CS-4) is driven by the requirements of the services desired for IN CS-4, and constrained by the capabilities of the embedded base of network technology. The functionality required to support IN CS-4 services includes functionality to provide:

- end-user access to call/service processing;
- call-related service invocation and control;
- end-user interaction with service control;
- service management;
- Call Party Handling;

- internetworking;
- security;
- Out-Channel User Interaction;
- call-unrelated service invocation and control; and
- feature interactions.

Each of these aspects is addressed below.

#### **4.1 End-user access**

End-user access to call/service processing is provided via the following access arrangements<sup>2</sup>:

- analogue line interfaces;
- ISDN BRI and PRI;
- traditional trunk and SS7 interfaces; and
- any interface using call signalling procedures that are compatible with the BCSM (e.g. SIP-based interfaces).

#### **4.2 Call-related service invocation and control**

Call/service processing builds upon the current call processing infrastructure of existing digital exchanges. It does so by using a generic model of existing call control functions to process basic two-party calls, then adding service switching functions to invoke and manage IN service logic. This generic model is known as the Basic Call State Model (BCSM). Once invoked, IN service logic is executed under the control of service control functions, in conjunction with service data functions. With this distributed approach to call/service processing, the existing call control function retains ultimate responsibility for the integrity of calls, as well as for the control of call processing resources. The following call/service processing constraints apply:

- a) Call control and service switching functionality are tightly coupled; thus, the relationship between SSF and CCF is not standardized.
- b) A call is either between two or more end users that are external to the network and addressable via a directory number or combination of directory number and bearer capability, or a call is between one or more end users and the network itself.
- c) A call may be initiated by an end user, or by a SCF within the network on behalf of an end user. To supplement a call, IN service logic may either be invoked by an end user served by an IN exchange, or by the network on behalf of an end user.
- d) A call may span multiple exchanges. As such, each exchange only controls the portion of the call in that exchange – call processing is functionally separated between exchanges. IN service logic invoked on IN exchanges in such inter-exchange calls are managed independently by each IN exchange.
- e) Existing exchanges can be viewed as having two functionally separate sets of call processing logic that coordinate call processing activities to create and maintain a basic two-party call. This functional separation is provided between the originating portion of the call and the terminating portion of the call. This functional separation should be maintained in an IN exchange to allow IN service logic invoked on the originating portion of the call (i.e. on behalf of the calling party) to be managed independently of IN service logic invoked on the terminating portion of the call (i.e. on behalf of the called party).

---

<sup>2</sup> This does not preclude the use of these interfaces to support access from private or mobile networks.

- f) It is desirable to allow multiple IN-supported service logic instances to be simultaneously active for a given end user. It is also recognized that non-IN service logic will continue to exist in the network. As such, service feature logic instances mechanisms should:
  - determine which service logic to invoke for a given service request. This mechanism should select the appropriate IN-supported service logic or non-IN-supported service logic, and block the invocation of any other service logic for that particular service request;
  - ensure that simultaneously active IN-supported service logic instances adhere to the single-ended restriction on service processing.
- g) The distributed approach and added complexity of call/service processing requires mechanisms for fault detection and recovery, allowing graceful termination of calls and appropriate treatments for end users.

### **4.3 End-user interaction**

End-user interaction with the network to send and receive information is provided by service switching and call control resources, augmented by specialized resources. These specialized resources are controlled by service control functionality, and are connected to end users via call control and service switching functionality.

### **4.4 IN service management functionality**

IN service management functionality is used to provision and manage the service control functionality, service data functionality, specialized resource functionality and the combined service switching/call control functionality in the network, outside of the context of call/service processing. The IN management functions are modelled according to the TMN functional architecture.

ITU-T Rec. M.3010 specifies the functional information and physical architectures that support the TMN management services. A TMN management function is the smallest part of the TMN management service as perceived by the user of the service. It will generally consist of a sequence of actions on a defined managed object or objects. The TMN management functions are described in ITU-T Rec. M.3400.

Many of the IN management functions are not IN-specific but can be mapped onto TMN management functions and reused for IN purposes.

Furthermore, the TMN uses OSI system management functions. These functions provide generic management controls and capabilities which can be used by specific TMN management services and functions. As a consequence of what is described above, a set of managed objects which are defined in the different system management functions, can be identified and reused for IN purposes.

In particular one can point out: "Summarization Function" (ITU-T Rec. X.738), "Workload Monitoring Function" (ITU-T Rec. X.739), "Event Management Function" (ITU-T Rec. X.734), "Alarm Reporting Function" (ITU-T Rec. X.733) and "Performance Management Function" (ITU-T Rec. Q.822).

### **4.5 Call Party Handling**

Call Party Handling (CPH) is the ability to manage various parties' involvement in a call. CPH capabilities rely on an abstract model of call and connection processing activities, maintained by the SSF, in terms of call and connection states. They provide IN services with the ability to add, delete, join and/or separate parties bearer channels from the other parties involved in a call.



## **4.6 Internetworking**

This capability set identifies the SCF-SDF, SCF-SCF, SMF-SMF and SDF-SDF relationships for internetworking purposes. Distributed service logic, but not distributed service control, is supported. Additionally, internetwork management interactions and distributed data handling processes are supported.

## **4.7 Security**

Security is supported through the provision of a number of security-assisting functions. A security-assisting function requires the cooperation of two or more functional entities. The security-assisting functions support both internal network operations and interworking between two or more networks. Security features do not automatically guarantee network integrity, but rather they provide some of the tools in order to allow secured systems to be built.

## **4.8 Out-Channel User Interaction**

The "Out-Channel Call-Related User Interaction" (OCCRUI) and the "Out-Channel Call-Unrelated User Interaction" (OCCUUI) network aspects are used to provide IN services with the ability to request the transparent transfer of information between a User and a Service Logic.

The term "user" refers to any application process which may reside in physical entity connected to the network (e.g. an ISDN terminal, a SCP or any equipment which may be connected to the network through a PRI or trunk line).

## **4.9 Call-unrelated service invocation and control**

This capability enables IN-supported service logic instances to be invoked outside the context of a call. It may be required for services/service features such as message waiting indication and registration of the user location.

Call-unrelated service processing builds upon the current call-unrelated processing infrastructure of existing digital exchanges. It does so by using a generic model of existing call-unrelated activities, then adding call-unrelated service functions to invoke and manage IN service logic. This generic model is known as the Basic Call-Unrelated State Model (BCUSM).

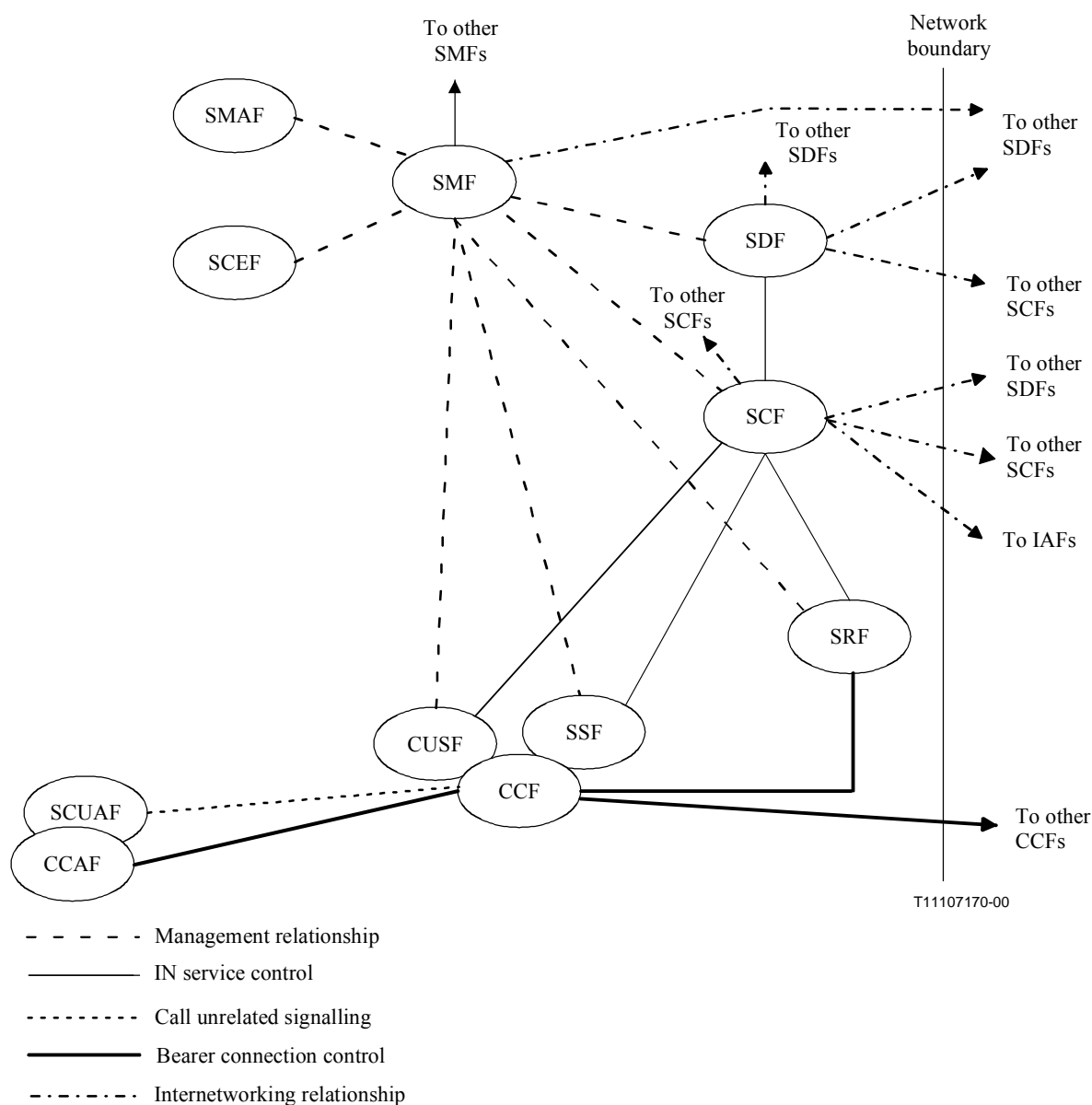
Call-unrelated processing and call-unrelated service functionality are tightly coupled; thus, the relationship between CUSF and CCF is not standardized.

## **4.10 Feature Interaction**

Feature Interaction mechanisms are means to manage feature/service interference situations which have been recognized prior to service deployment. Use of these mechanisms is a network operator option.

## **4.11 Distributed Functional Model**

Figure 1 identifies the IN DFP model for IN CS-4. This figure is provided here as a reference, and is used to show the relevant interfaces on which the protocol in ITU-T Rec. Q.1248 is provided. For detailed descriptions of the functional entities, one should refer to ITU-T Rec. Q.1241.



**Figure 1/Q.1248.1 – IN CS-4 DFP architecture**

#### 4.12 Communication model

For the purpose of communication, each Functional Entity (FE) includes a Functional Entity Access Manager (FEAM) which provides the necessary functionality to support the exchange of information with other FEs. This includes:

- establishing and maintaining the interfaces to other FEs using the appropriate underlying communication mechanism;
- passing and queuing (when necessary) the messages received from other FEs to other components of the FE; and
- formatting, queuing (when necessary), and sending to other FEs the messages received from other components of the FE.

On the following IN interfaces, interactions between a pair of communicating FEs are defined and specified using the Remote Operations (ROS) paradigm defined in ITU-T Rec. X.880:

- SSF-SCF interface;
- SCF-SRF interface;

- SCF-SDF interface;
- SDF-SDF interface;
- SCF-SCF interface;
- SCF-CUSF interface.

In such cases, the FEAM is modelled as a *ROS-Object* whose class definition includes the list of *contracts* used over each of the IN interfaces it supports. Each *contract* is in turn defined as a set of operation *packages*, each of which includes one or more *operations* supporting the elementary interactions between the involved FEs.

The internal behaviour of each FE is modelled using one or more Finite State Machines (FSMs) which are interfaced with the FEAM.

This communication model does not assume any particular physical realization. It remains valid when two FEs are co-located in the same equipment or communicate through a network. At the Physical Plane, when two communicating FEs are located in different physical entities (PE), each contract is realized by an application-context using the Transaction Capabilities (TC) or using the ROSE capabilities. In addition to TC, the application-context includes a set of Application Service Element (ASE) each of which embodies the knowledge of an operation package. This is further explained in clause 7.

The communication model used on interfaces to the SMF is not subject to standardization.

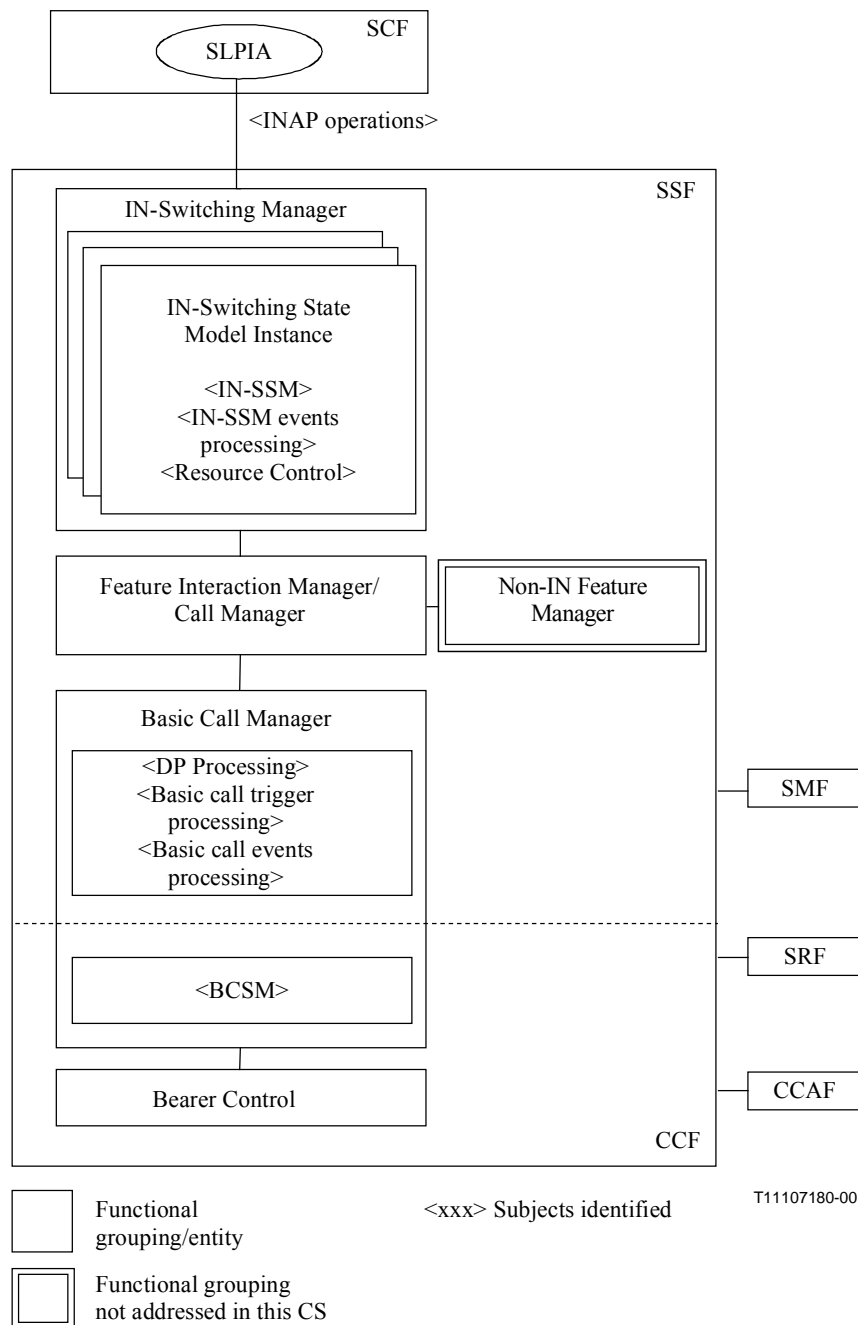
## 5 FE models

This clause provides a general description of each of the functional entities identified in the Distributed Functional Plane. Detailed descriptions of the functional entities are provided in the Q.1248.x family of ITU-T Recommendations. In case of discrepancy between this clause and these Recommendations, the latter take precedence.

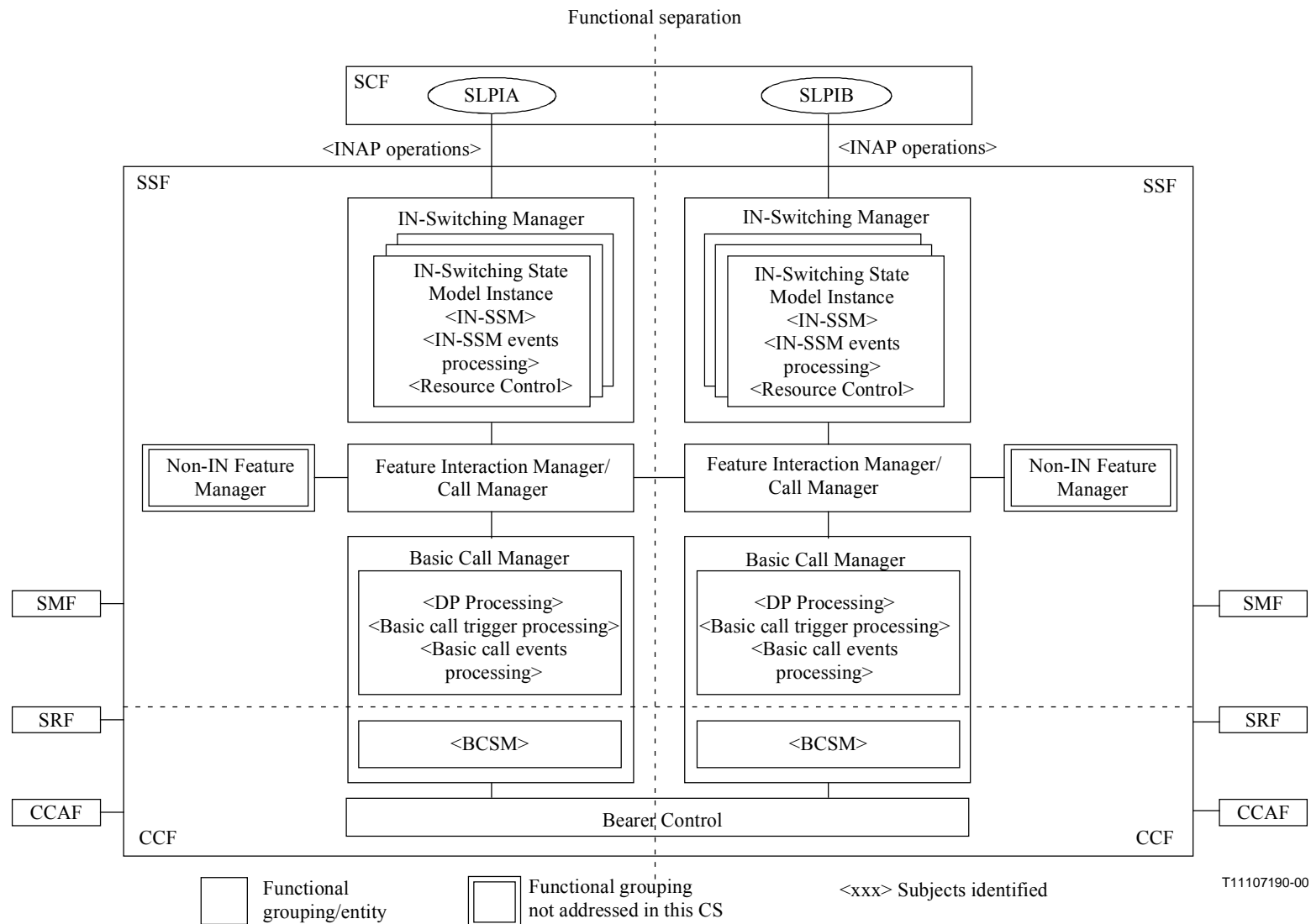
### 5.1 Call Control Function/Service Switching Function (CCF/SSF) models

#### 5.1.1 General

A model of the CCF/SSF is shown in Figures 2 and 3. Figure 2 shows the CCF/SSF model for a single-ended service logic instance related to a calling or called party. Figure 3 shows the CCF/SSF model for separate single-ended service logic instances related to the calling and called parties on the same call. The functional separation serves to isolate single-ended service logic instances related to the calling party from single-ended service logic instances related to the called party for the same call. The purpose of the models is to provide a framework for the understanding of call modelling with respect to the CCF/SSF.



**Figure 2/Q.1248.1 – CCF/SSF model – Single-ended SLPI related to calling or called party**



**Figure 3/Q.1248.1 – CCF/SSF model – Separated single-ended SLPIs related to calling and called parties**

### 5.1.2 CCF/SSF components

The CCF/SSF model consists of several entities. Each of the entities are briefly described below, with additional details in ITU-T Rec. Q.1248.2. It is noted that this shows a conceptual model of CCF/SSF and is not intended to imply an actual implementation of the CCF/SSF.

#### 5.1.2.1 Basic Call Manager (BCM)

The BCM is not a functional entity. It provides an abstraction of the part of a switch which implements basic call and connection control to establish communication paths for users, and to interconnect such communication paths. It detects basic call and connection control events that can lead to the invocation of IN service logic instances or should be reported to active IN service logic instances, and manages CCF/SSF resources required to support basic call and connection control.

The BCM also implements the BCSM and the DP processing logic.

The DP processing logic is the entity of the BCM that interacts with the FIM/CM as described in the FIM/CM description below.

#### 5.1.2.2 Feature Interaction Manager/Call Manager (FIM/CM)

This entity in the SSF provides mechanisms to support multiple concurrent instances of IN and non-IN service logic instances on a single call. In particular, the FIM/CM can prevent multiple instances of IN and non-IN service logic instances from being invoked. As well, there is no functionality in the SSF for handling service feature interactions between the separate SSF calling party processes and SSF called party processes in this capability set. The FIM/CM integrates these interaction mechanisms with the BCM and IN-SM to provide the SSF with a unified view of call/service processing internal to the SSF for a single call.

In this capability set, the FIM/CM supports:

- prevention of non-IN service logic instance invocation according to the service interactions indicators received from an IN service logic instance. This IN-based – switch-based feature/service interaction provides to an IN service logic the ability to allow/deny or modify switch-based service logic execution via call-related signalling. This mechanism is extensible, such that new service interactions may be covered as required. New service interactions may arise due to the introduction of new basic network services or new IN services.
- prevention of multiple IN service logic invocations according to the service compatibility indicators which are assigned, via administration, to each IN service by the network operator. This IN-based – IN-based feature/service interaction consists of two different approaches, which may be applied either singly or in combination.

##### a) *SSF and/or single SCF-oriented approach*

The triggering mechanisms are used and enhanced to avoid feature interference situations. Trigger precedence rules and service compatibility checks are already in place, which may be used to avoid feature interference. During triggering, the FIM performs a generic compatibility check procedure using an administered exclusion matrix for these indicators.

For each triggering of an IN service during a call, the according indicator is derived and shall be propagated through the network. For the case where a service compatibility indicator was already received from the network, the new indicator is added to the received one. The call signalling (e.g. ISDN User Part (ISUP)) is required to convey this indication.

When received by the IN service logic, the ability to overwrite it is a network-operator-specific option.

b) *Two SCF-oriented approach*

This approach is based on the exchange of information between the two involved SCFs (bidirectional), using the OCCRUI mechanism.

In case such mechanisms are used across network boundaries, the relevant signalling and the relevant procedures need to be standardized or bilaterally agreed.

Global management of feature interactions also requires conveying these indicators via basic network signalling to the originating and terminating instances.

**5.1.2.3 IN – Switching Manager (IN-SM)**

This entity in the SSF interacts with the SCF in the course of providing IN service features to users. It provides the SCF with an observable view of CCF/SSF call/connection processing activities, and provides the SCF with access to control CCF/SSF capabilities and resources. It manages SSF resources required to support IN service logic instances. The IN-SM interacts with the FIM/CM as described above. It also includes the necessary functionality to exchange information with other functional entities (i.e. the FEAM).

Other aspects shown in Figure 2 are not addressed in this capability set, but are assumed to exist.

**5.1.3 CCF/SSF trigger information objects**

This clause provides an overview of the trigger information objects required for the CCF/SSF. Detailed descriptions of the trigger information can be found in ITU-T Rec. Q.1248.2.

The entities/components that are involved in triggering include:

- a) the DP processing entity in the SSF;
- b) the Feature Interactions Manager/Call Manager in the SSF; and
- c) the IN-Switching Manager entity in the SSF.

The entities/components shall collectively:

- a) perform the DP processing actions specified in ITU-T Rec. Q.1248.2;
- b) check if traffic mechanisms are active;
- c) check for SCF accessibility;
- d) handle service feature interactions.

They shall hand back control to the CCF for at least the following cases:

- a) if call gapping is in effect: the SSF logic instructs the CCF to continue the call processing with the appropriate treatment;
- b) if service filtering is in effect: the call is counted (if required) and the SSF logic instructs the CCF to continue the call processing with the appropriate treatment;
- c) if a trigger criteria match is not found (e.g. insufficient information to proceed): the SSF logic returns call control to the CCF;
- d) if the call is abandoned: the SSF logic returns call control to the CCF and continues processing;
- e) if the destination SCF is not accessible: the SSF logic instructs the CCF to route the call if possible (e.g. default routing to a terminating announcement);
- f) if there is an existing control relationship for the call and a DP is encountered which is armed as an TDP-R: the SSF returns call control to the CCF in case Multiple Point of Control (MPC) is not supported.

Triggering relies on information stored in the trigger tables in the form of a set of trigger items. For each item, the trigger information model includes:

- a) Trigger mode (i.e. an indication on whether triggering should cause call processing suspension).
- b) Detection Point in BCSM.
- c) Trigger criteria.
- d) Information required for SCF addressing.
- e) Service key.
- f) Application Context Name.
- g) Fault handling (to indicate call handling on an error being encountered):
  - complete call as dialled;
  - release call;
  - play specified announcement and release call;
  - route call to specified alternate destination.
- h) Association to CCF/SSF user/network resource.
- i) Service Compatibility Indicators.
- j) Trigger precedence.

The following Configuration Management Functions are possible:

a) *Trigger Table Configuration*

The following capabilities should be possible on the CCF/SSF:

- 1) Introduction of new trigger data along with the corresponding service key and relevant information.
- 2) Remove a trigger data identified by its service key.
- 3) Activate or deactivate existing trigger data within the CCF/SSF.

The SSF should be able to predict at service deployment time unwanted service interactions. If this occurs, new trigger data should be disallowed and an error handling mechanism invoked with the SMF.

b) *Configuration verification*

The SMF should be able to determine the current configuration of the CCF/SSF at any time. This may be accomplished by having the SMF request the CCF/SSF to provide to the SMF a copy of the trigger information for an CCF/SSF user resource.

#### **5.1.4 SSF FSM structure**

The SSF FSM provides the SCF an observable view of the current state of the SSF involved in a SSF/SCF relationship. The SSF FSM is a finite state machine that is part of the IN SSM, instantiated by the IN-SM. The state of the SSF FSM depends on information exchanged between the SSF FSM and the SCF via the FEAM.

Figure 4 shows the SSF FSM structure. The interfaces shown in this figure are internal, and are not subject to standardization.





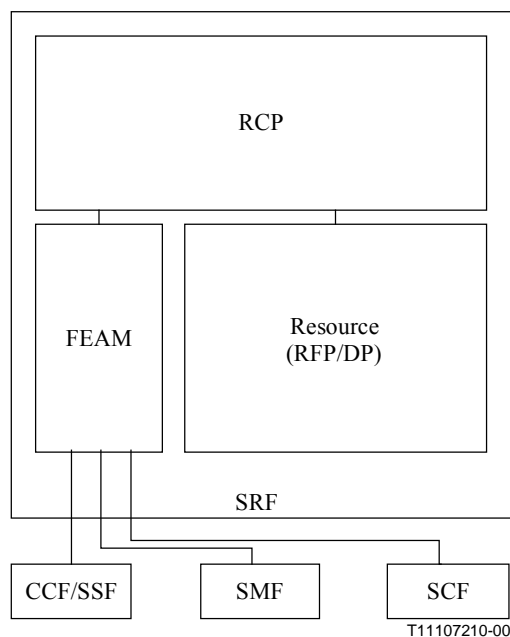
The SSME-control maintains the dialogues with the SCF, and SRF on behalf of all instances of the SSF Finite State Model (FSM). These instances of the SSF FSM occur concurrently and asynchronously as calls occur, which is why there is only a single entity that performs the task of creation, invocation, and maintenance of the SSF FSMs. In particular the SSME-control performs the following tasks:

- a) Interprets the input messages from other FEs and translates them into corresponding SSF FSM events.
- b) Translates the SSF FSM outputs into corresponding messages to other FEs.
- c) Captures asynchronous (with call processing) activities related to management or supervisory functions in the SSF and creates an instance of a SSME FSM. For example, the SSME provides non-call-associated treatment due to changes in Service Filtering or Call Gapping. Therefore, the SSME-control separates the SSF FSM from the Call Gapping and Service Filtering functions by creating instances of SSME FSMs for each context of management-related operations.

## 5.2 Specialized Resource Function (SRF) models

### 5.2.1 General

A model of the SRF is shown in Figure 5. The purpose of this model is to provide a framework for the understanding specialized resource functions with respect to the SRF.



**Figure 5/Q.1248.1 – SRF model**

Management of the SRF is required for placing resources in or out of service, e.g. for provisioning, administration and maintenance purposes. The SRF management is activated by a request from another functional entity, and never takes action by itself.

For call/service processing, the SRF has a logical relationship with the CCF/SSF and the SCF. The SCF controls the connection between the CCF/SSF and the SRF, and sends instructions to the SRF.

As part of the process of formulating a response to the SSF, the SCF may need to enter into a dialogue with a calling or a called party. This could, for example, take the form of a prompt and collect digits sequence. In this case the SCF will instruct the SRF to start a dialogue with a user after setting up a bearer path between the CCF/SSF and the SRF. The dialogue between the SRF

and the user allows the SRF to play an announcement, and if appropriate, collect digits. If digits have been collected, the SRF will pass the digit information to the SCF. When the service logic in the SCF does have further need the resources, the SCF requests the CCF/SSF to release the connection to the SRF, and the SRF resource will be released.

### **5.2.2 SRF components**

This clause describes the various components found within an SRF. It is noted that this shows a conceptual model of SRF and is not intended to imply an actual implementation of the SRF.

The main SRF components are:

- a) Functional Entity Access Manager (FEAM) as described in clause 4;
- b) Resource Control Part (RCP);
- c) Resources.

The RCP mainly comprises a block called the SRF Resource Manager along with the Resource Logic Library and the Resource Logic Instances, each of which controls two types of resources:

- a) Resource Function Part (RFP);
- b) Data Part (DP).

Details of the components are described in the following subclauses:

#### **5.2.2.1 Functional Entity Access Manager (FEAM)**

The FEAM provides the necessary functionality to exchange information with other functional entities as described in clause 4.

#### **5.2.2.2 Resource Control Part (RCP)**

The RCP contains SRF service logic, and controls the service procedure using the capabilities of other blocks. To offer a specialized resource, RCP uses resource-function pair in the RFP and data in the Data Part (DP).

The RCP contains the SRF Resource Manager, the Transaction Module, the User-Interaction script (UI script), the Resource Logic Library and the Resource Logic Instances. Detailed descriptions can be found in ITU-T Rec. Q.1248.3.

#### **5.2.2.3 Resource Function Part (RFP)**

The RFP is a collection of resource-function pairs or functional resource elements. For a service procedure, resources in a resource-function pair are allocated and released together. It should be noted that new resources could be identified as new services/service features are considered.

#### **5.2.2.4 Data Part (DP)**

The DP is composed of a database manager and a database containing recorded voice, sound, image, text, etc.

### **5.2.3 SRF functional resources**

This clause provides an overview of the SRF functional resources. Further information can be found in ITU-T Rec. Q.1248.3.

The following functional resources are supported in this capability set and are defined as follows:

- a) *DTMF receiver*  
This resource receives dual tone multi-frequency (DTMF) from a linked resource.
- b) *Tone generator/announcements*  
This resource provides in-channel information to the specified virtual resource.

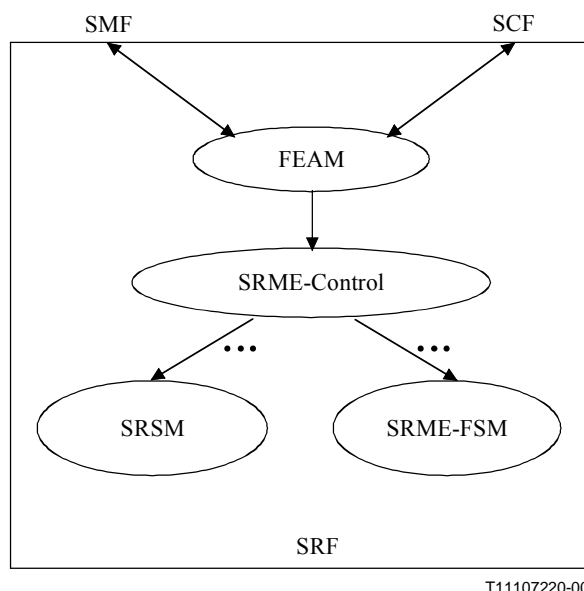
- c) *Message sender/receiver*  
This resource sends or receives messages, such as electronic messages, voice messages, etc., to/from users.
- d) *Synthesized voice/speech recognition device with interactive prompting facilities*  
This resource receives in-channel speech information from a linked virtual resource. When the information is input from a user, it is recognized by this resource and this resource converts it to IN operations. When this resource receives an instruction to send a voice message with source-information, it is converted to voice message. Usually, such action is performed with interactive prompting.
- e) *Text-to-speech synthesis*  
This resource provides the ability to convert a text on an inband information.
- f) *Modem detection*  
This resource provides the ability to detect a modem.
- g) *SDSS capability*  
This resource provides the ability to interact with an analogue user by means of the SDSS (Server Display and Script Service) protocol on the analogue interface (ITU-T Rec. V.23).

Examples of other kind of functional resources not supported in this capability set are provided below:

- a) *Audio conference bridge*  
Receiving in-channel audio information from any other linked virtual resources, this resource mixes this information and sends the mixed information to all the linked virtual resources. Another new virtual resource can be joined to or any virtual resources linked to it can be split from this connection resource. It is used as an audio conference bridge.
- b) *Information distribution bridge*  
Receiving in-channel information from a linked virtual resource, this resource distributes the information to all the other linked virtual resources. New virtual resources can be joined to existing connection resource, or existing virtual resources can be split from the connection resource. It is used as a broadcasting device.
- c) *Protocol converters*

#### **5.2.4 SRF FSM structure**

Figure 6 shows the SRF FSM structure. The interfaces shown in this figure are internal, and are not for standardization.



**Figure 6/Q.1248.1 – SRF FSM structure**

When a call attempt is initiated by the SSF, an instance of an SRF FSM is created. The SRF FSM handles the interaction with the SCF FSM and the SSF FSM.

The management functions related to the execution of operations received from the SCF are executed by the SRF Management Entity (SRME). The SRME interfaces to the various SRF call State Models (SRSMS) and the functional entity access manager (FEAM).

The model associates a Finite State Machine (FSM) with each initial interaction request from the SCF. Thus, multiple initial requests may be executed concurrently and asynchronously by the SRF, which is why there is only a single entity (SRME) that performs the tasks of creation, invocation, and maintenance of the SRSMS objects. In addition to the above tasks, the SRME maintains the dialogues with the SCF and SSF on behalf of all instances of the SCSM. In particular, the SRME:

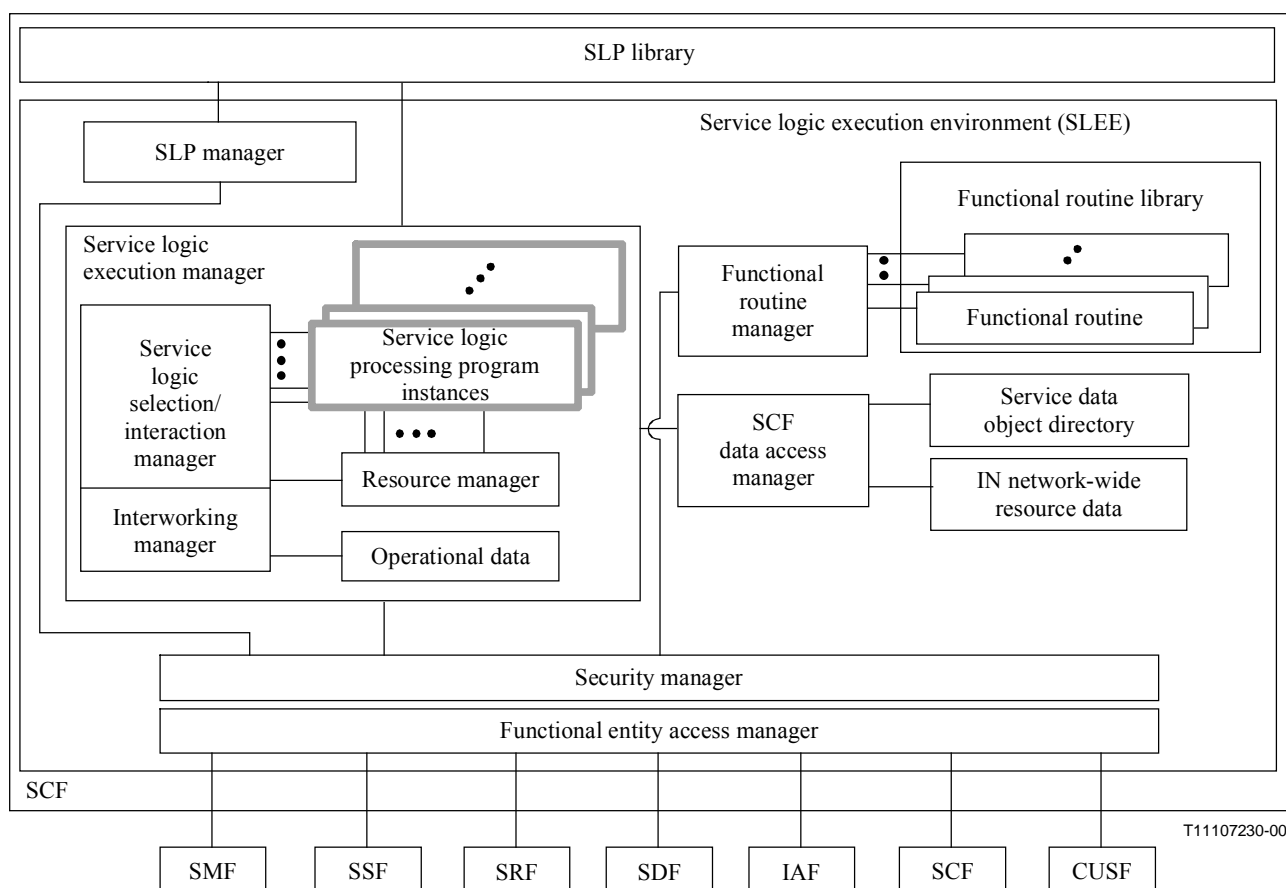
- interprets the input messages from other FEs and translates them into corresponding SRSMS events; and
- translates the SRSMS outputs into corresponding messages to other FEs.

Finally, the FEAM provides the lower level maintenance functions on the interface to SSF and SCF, as described in clause 4.

### 5.3 Service Control Function (SCF) models

#### 5.3.1 General

A model of the SCF is shown in Figure 7. The purpose of this model is to provide a framework for the understanding of service logic processing with respect to the SCF.



**Figure 7/Q.1248.1 – SCF model**

The prime function of the SCF is the execution of service logic, which is provided in the form of Service Logic Processing programs (SLPs). The SCF also includes the SLP execution supporting functions, such as service logic selection/interaction management, functional entity access management, and SLP provisioning management.

### 5.3.2 SCF components

The SCF model providing the functionality defined above is shown in Figure 7. It is noted that this shows a conceptual model of the SCF and is not intended to imply an actual implementation of the SCF.

The SCF platform provides a Service Logic Execution Environment (SLEE) on which a Service Logic Processing program (SLP) runs to provide service processing. An SLP is a service application program invoked by the SLEE and is used to realize service processing under the control of the SLEE. The simultaneous invocation and execution of multiple SLPs are also managed by the SLEE.

Each of the entities shown in Figure 7 will be described in the following subclauses.

#### 5.3.2.1 Service Logic Execution Manager (SLEM)

##### 5.3.2.1.1 General

The SLEM contains the functions to handle and control the complete service logic execution action. The SLEM also contains service logic processing program instances (SLPIs), service logic selection/interaction manager, and resource manager. It also interacts with SCF data access manager and FEAM to support SLPI execution. In addition to these aspects, the SLEM needs functionality to:

- a) execute SLPIs and maintain transient data associated with SLPIs (i.e. information that only persists during the lifetime of the SLPI, such as SLPI state information);
- b) execute functional routines in support of SLPI execution;
- c) manage SLPI access to SCF and SDF data via the SCF data access manager;
- d) manage the exchange of information between SLPIs and entities in other functional entities via the FEAM.

#### **5.3.2.1.2 Service Logic Selection/Interaction Manager (SLSIM)**

The SLSIM is the entity that selects an SLP for execution and controls the simultaneous execution and/or execution order of multiple SLPs in the same SCF. This capability set does not define whether or not the SLSIM is explicitly divided into two different entities, i.e. the service logic selection manager and the service logic interaction manager.

As part of the functionality, the SLSIM provides a means to manage service interactions by managing interactions among multiple SLPIs in the same SCF that are simultaneously active on a single call. The relationship between SLSIM and the feature interaction manager/call manager in the CCF/SSF is not defined.

SLP selection is performed via the SLSIM in response to:

- a) an external event from another functional entity;
- b) the occurrence of internally recognized conditions (e.g. time of day or other internal events); and
- c) the execution of a functional routine via an SLPI that requests the execution of another SLP.

In addition, the SLSIM should invoke the execution of the selected SLP and provide for mutual exclusion and precedence during this SLP selection and invocation:

- a) mutual exclusion prevents the invocation of an SLP whose execution would be incompatible with a currently executing SLPI;
- b) precedence provides a scheme to select a particular SLP from a set of SLPs which meet the same selection criteria.

#### **5.3.2.1.3 Service Logic Processing program Instance (SLPI)**

A Service Logic Processing program (SLP) is a service application program invoked by the SLEE and used to realize service processing. It contains logical constructs which, when executed, control the flow of service execution, and invoke functional routines in the SCF to access network data resources needed for service execution. When an SLP is selected and invoked, it is referred to as a Service Logic Processing program Instance (SLPI). In contrast to an SLP, a corresponding SLPI is a dynamic entity that actively controls the flow of service execution and invokes SCF functional routines.

Functional routines are programs in the SCF that can be invoked by SLPIs to cause a sequence of actions to be performed in the network in support of service execution. This sequence of actions provide the functionality defined for a service-independent building block (SIB) on the Global Functional Plane. Therefore, functional routines are considered to be service independent.

#### **5.3.2.1.4 Resource manager**

The resource manager provides functionality to control the allocation of local SCF resources and provides access to network resources in support of SLPI execution. The resource manager contains functionality to:

- a) identify and locate local SCF resources;
- b) identify and locate network resources via the SCF data access manager and IN network-wide resource data;
- c) identify one or more local SCF resources requested by a particular SLPI;
- d) release one or more local SCF resources no longer needed by a particular SLPI; and
- e) interact with other functional entities via the FEAM to provide for the reservation and release of network resources to be used by SLPIs.

It should be noted that the selection of an SRF is not always performed by the SLEM resource manager. In some cases selection is performed by an SSF, for example, where assist/hand-off procedures are being used.

#### **5.3.2.1.5 Internetworking manager**

Internetworking manager provides the functionality required for internetworking:

- a) access control based on the originating entity and the associated service/feature;
- b) secure access to the service logic and monitoring of the illegal access;
- c) accounting of the interworking activity (if needed).

#### **5.3.2.2 SCF data access manager**

##### **5.3.2.2.1 General**

The SCF data access manager provides the functionality required to provide for the storage, management, and access of shared and persistent information in the SCF (i.e. information that persists beyond the lifetime of a SLPI). The SCF data access manager also provides the functionality required to access remote information in SDFs. The SCF data access manager interacts with the SLEM to provide these capabilities to SLPIs.

Figure 7 identifies two data structures that contain SCF data. These include:

- a) the service data object directory; and
- b) the IN network-wide resource data.

These are described in the following clauses.

##### **5.3.2.2.2 Service data object directory**

Figure 7 identifies a service data object directory. The service data object directory provides a means to address the appropriate SCF for access to a specific data object.

The SLEM interacts with the SCF data access manager to access service data objects in SDFs. The SCF data access manager uses the service data object directory to locate service data objects in the network in a manner transparent to the SLEM (and its SLPI). As such, the SLEM (and its SLPIs) has a global and uniform view of service data objects in the network.

##### **5.3.2.2.3 IN network-wide resource data**

This is a data structure, accessible to SLPIs, in which information resides about the location and capabilities of resources in the network. It provides a means to address the appropriate functional entity (e.g. SRF) for access to specific resources with the appropriate capabilities.



The SLEM resource manager interacts with the SCF data access manager to access network resource data. The SLEM resource manager provides SLPs with access to network resources in a manner transparent to SLPs. As such, SLPs have a global and uniform view of resources in the network.

#### **5.3.2.3 Functional routine manager**

The Functional Routine Manager is used for receiving and distributing of functional routines to the functional routine library via the FEAM. This entity also manages the addition, deletion and suspension of a particular functional routine. The management of these functional routines by the SMF is not defined.

Functional routine library is an entity where the actual functional routines reside.

#### **5.3.2.4 Functional Entity Access Manager (FEAM)**

The FEAM provides the functionality required by the SLEM to exchange information with other functional entities via messages, as described in clause 4.

#### **5.3.2.5 SLP manager**

The SLP manager manages the reception and distribution function of SLPs from other entities. The SLP manager, therefore, interworks with the FEAM. This entity also manages addition, deletion and suspension of a particular SLP. The management of the SLP is via the SMF.

#### **5.3.2.6 Security manager**

The security manager provides the functionality required for providing secure access to the SCF by other Functional Entities (SCF, SDF). The security manager should:

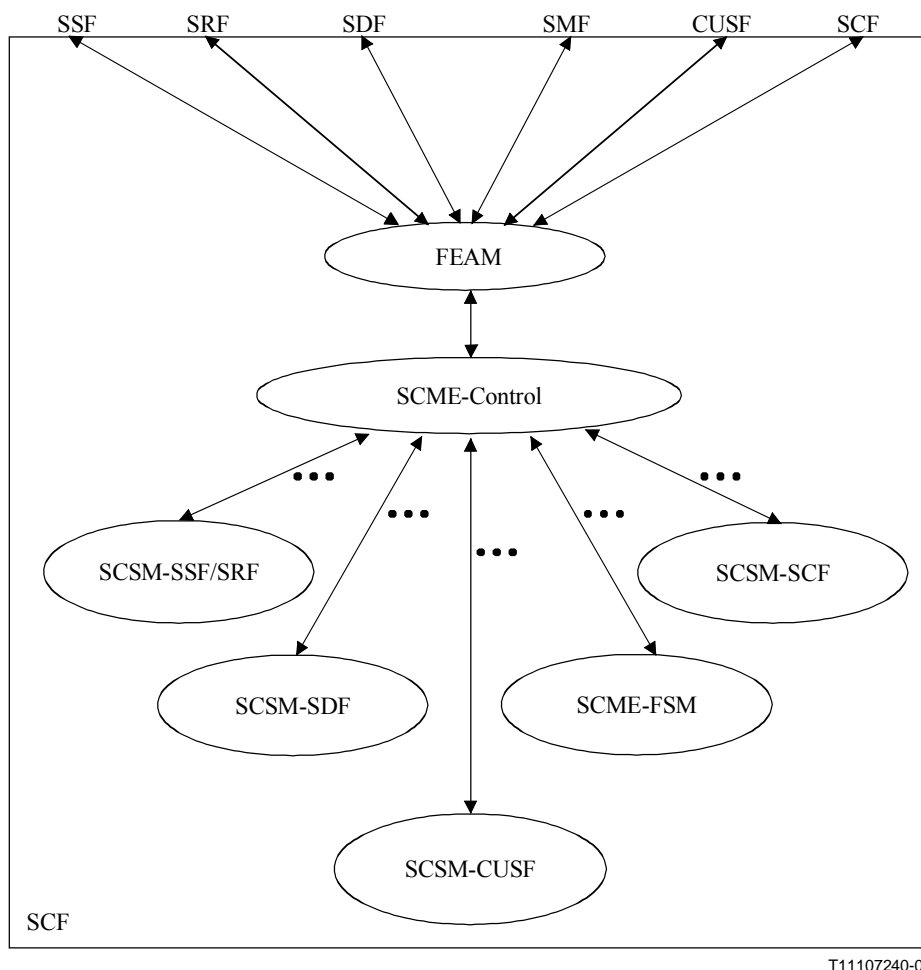
- a) generate security parameters for outgoing messages to provide data confidentiality, data origin authentication, data integrity for a specific interface;
- b) verify security parameters for incoming messages to provide data confidentiality, data origin authentication, data integrity for a specific interface. If an incoming message has an invalid security parameter then the message should be rejected and the invalid access attempt logged.

The level of security applied to an interface is determined by which of the security parameter features are active. The selection of active features is performed through interaction with the SMF.

### **5.3.3 The SCF FSM structure**

Figure 8 shows the SCF FSM structure. The interfaces shown in this figure are internal, and are not for standardization. In the SCF Call State Model (SCSM), there are four kinds of FSMs: SSF/SRF, SDF, SCF and CUSF. They consist of:

- a) the SSF/SRF FSM (SCSM-SSF/SRF), which handles the interaction with the SSF/SRF.
- b) the SDF FSM (SCSM-SDF), which handles the interaction with the SDF FSM.
- c) the CUSF FSM (SCSM-CUSF), which handles the interaction with the CUSF FSM.
- d) the inter-SCF FSM (SCSM-SCF), which handles interactions with other SCFs.



**Figure 8/Q.1248.1 – SCF FSM structure**

The SCF also includes the SCME, which handles the interaction between the SCF and the SCF management functions.

The following text provides an overview of the procedural aspects of the interfaces between the SCF and other functional entities.

The relationship between the SLP and the SCF FSM may be described as follows:

- a) If a request for IN call processing is received from the SSF, an instance of an SCF Call State Model (SCSM) is created, and the relevant SLP is invoked.
- b) When initiation of a call is requested from service logic, an instance of the SCSM is created.

In either case, the SCF FSM handles the interaction with the SSF FSM, SRF FSM, SCF FSM, CUSF FSM, and SDF FSM, and notifies the SLP of events as required.

The management functions related to the execution of operations received from the SSF, SRF, SDF, CUSF, or cooperating SCF are executed by the SCF Management Entity (SCME).

The SCME is comprised of the SCME-Control and multiple instances of SCME FSMs. The SCME-Control interfaces the different SCF Call State Models (SCSMs) and the Functional Entity Access Manager (FEAM). These instances of the SCF FSMs occur concurrently and asynchronously as SCF related events occur, which is why there is a need for a single entity that performs the task of creation, invocation, and maintenance of the SCF FSMs. In particular, the SCME-Control performs the following tasks:

- a) interprets the input messages from other FEs and translates them into corresponding SCSM events;
- b) translates the SCSM outputs into corresponding messages to other FEs;
- c) performs some asynchronous (with call processing) activities (one such activity is flow control) related to management and supervisory functions in the SCF and creates an instance of a SCME FSM. It is the responsibility of the SCME-Control to detect nodal overload and send the Overload Indication (e.g. Automatic Call Gap) to the SSF to place flow control on queries. Other such activities include non-call associated treatment due to changes in Service Filtering, Call Gapping, or Resource Monitoring status and also provision of resource status messages to the SSF. Therefore, the SCME-Control separates the SCSM from these activities by creating instances of SCME FSMs for each context of related operations;
- d) supports persistent interactions between the SCF and other FEs.

The different contexts of the SCME FSMs may be distinguished based on the address information provided in the initiating operations. In the case of service filtering, this address information is given by Filtering Criteria, i.e. all ActivateServiceFiltering operations using the same address will access the same SCME FSM handling this specific service filtering instance. ActivateServiceFiltering operations providing different Filtering Criteria cause the invocation of different SCME FSMs.

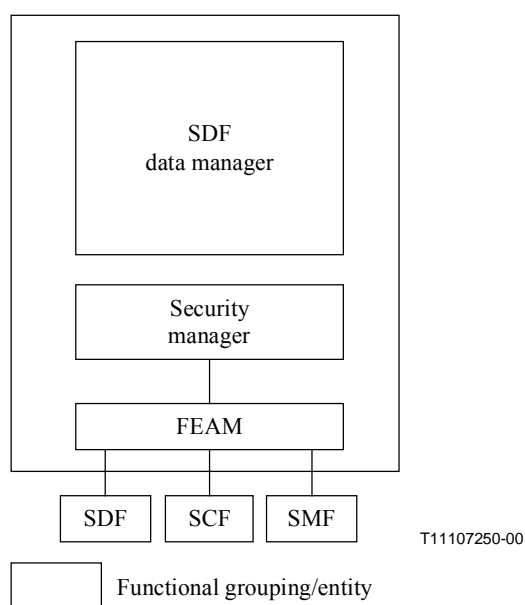
Finally, the FEAM provides the lower-level maintenance functions on the interface as described in clause 4.

Note that although the SCSM includes a state and procedures concerning queue management, this type of resource management only represents one way of managing network call queues. Another alternative is to let the CCF/SSF manage call queues; however, the technical details of how the CCF/SSF performs queue management is beyond the scope of IN.

## 5.4 Service Data Function (SDF) models

### 5.4.1 General

A model of the SDF is shown in Figure 9. The purpose of this model is to provide a framework for the understanding of service data functions with respect to the SDF.



**Figure 9/Q.1248.1 – SDF model**

Clause 5.4.2 describes the detailed SDF architecture, and classifies the data types which are handled by the SDF.

The SDF contains and manages data associated with the service logic processing programs (SLPs), and accessed during the execution of the SLP instances (SLPIs). Therefore, data such as SLP selection data and SCF directory, which are accessed before the execution of an SLPI, are not included in the SDF data handling.

#### **5.4.2 SDF components**

The SDF model to provide the functionality defined above is shown in Figure 9. Each of the functional entities shown in Figure 9 will be described in the following subclauses. It is noted that this shows a conceptual model of SDF and is not intended to imply an actual implementation of the SDF.

##### **5.4.2.1 SDF data manager**

The SDF data manager provides the functionality required for storing, managing and accessing information in the SDF. If, for example, the data is physically structured as a database, the SDF data manager may also handle the database accessing language such as SQL.

##### **5.4.2.2 Functional Entity Access Manager (FEAM)**

The FEAM provides the functionality required by the SDF data manager to exchange information with other functional entities (i.e. SCF, SDF and SMF), as described in clause 4.

##### **5.4.2.3 Security manager**

The security manager provides secure access to the different types of data held in the SDF, for example, denied access to the data for unauthenticated users. This functionality should:

- a) check the access rights of the SCF;
- b) authenticate users with provided information;
- c) count the failed authentication attempts for a given user (whether the implementation of this function can be realized in the SDF has not been investigated yet);
- d) block data access;
- e) assign user's access rights;
- f) memorize user's access rights during his request;
- g) control user's right to access specific data.

The security manager also provides the functionality required for providing secure access to the functions of the SDF by other Functional Entities (SCF, SDF). This functionality should:

- generate security parameters for outgoing messages to provide data confidentiality, data origin authentication, data integrity for a specific interface;
- verify security parameters for incoming messages to provide data confidentiality, data origin authentication, data integrity for a specific interface. If an incoming message has an invalid security parameter, then the message should be rejected and the invalid access attempt logged.

The level of security applied to an interface is determined by which of the security features are active. The selection of active features is performed through interaction with the SMF.

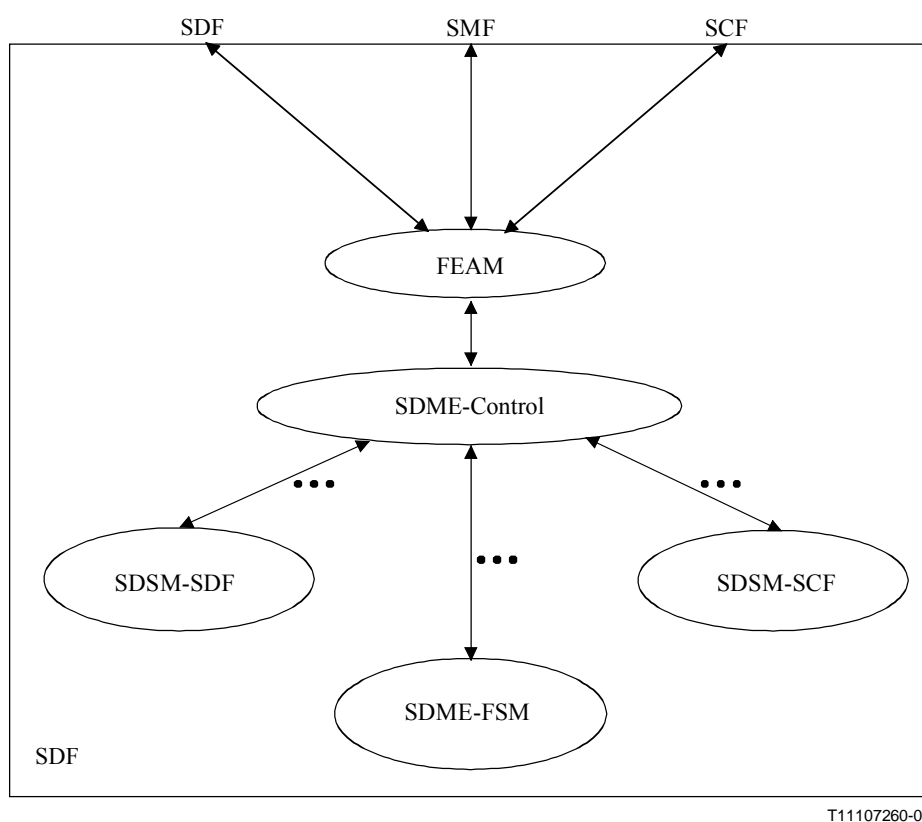
### 5.4.3 Data types handled by SDF

Data handled by the SDF can be classified into the following types:

- a) Authentication data – This data is used to authenticate a user that accesses the database through an SCF, e.g. a PIN code, the value of a counter for failed authentication. The set of authentication data used is associated to a level of access rights.
- b) Operational data – This data is not required by the SLPIs, but is used by the SDF itself for operational and administrative purposes, e.g. references to an object class, access control data.
- c) Service data – This data is used for the provision of a service, e.g. a subscriber profile, service provider agreements. If necessary, this data can be used by several services.

### 5.4.4 SDF FSM structure

Figure 10 shows the SDF FSM structure. The interfaces shown in this figure are internal, and are not for standardization.



**Figure 10/Q.1248.1 – SDF FSM structure**

In the SDF Call State Model (SDSM), there are two kinds of FSMs. They consist of:

- a) the SCF FSM (SDSM-SCF), which handles the interaction with the SCF.
- b) the SDF FSM (SDSM-SDF), which handles interactions with other SDF.

The SDF also includes the SDME, which handles the interaction between the SDF and the SDF management functions.

The SDME is comprised of an SDME-Control and several instances of SDME FSMs. The SDME control interfaces the different SDF FSMs (e.g. SDSM-SCF) and SDME-FSMs respectively as well as the Functional Entity Access Manager (FEAM).

The SDME-Control maintains the associations with the SCF and cooperating SDFs on behalf of all instances of the SDF FSMs (e.g. SDSM-SCF, SDSM-SDF). These instances of the SDF FSMs occur concurrently and asynchronously as SDF related events occur, which explains the need for a single entity that performs the task of creation, invocation and maintenance of the SDF FSMs. In particular, the SDME-Control performs the following tasks:

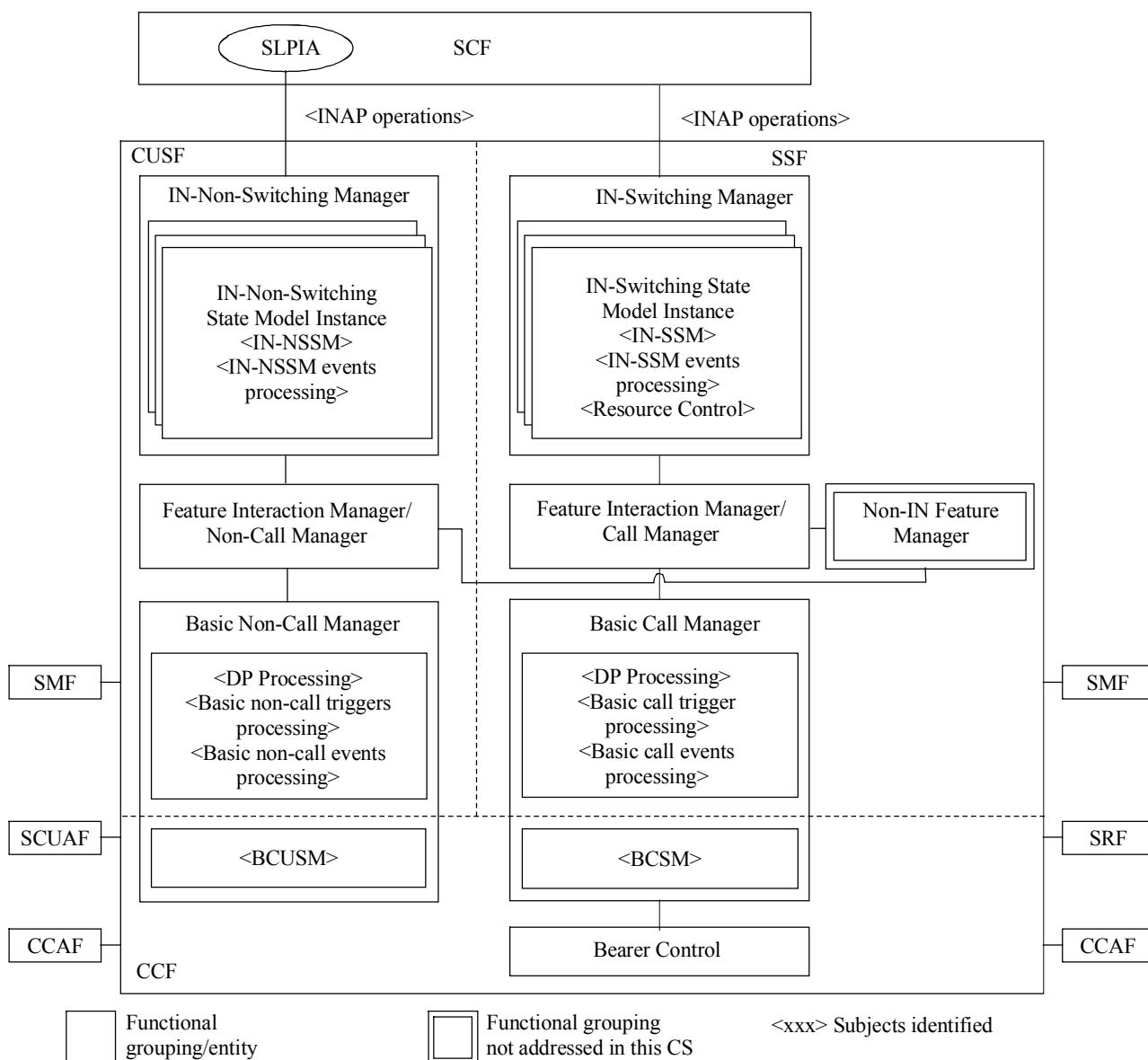
- a) Interprets the input messages from other FEs and translates them into corresponding SDF FSM events.
- b) Translates the SDF FSM outputs into corresponding messages to other FEs.
- c) Captures asynchronous activities related to management or supervisory functions in the SDF and creates an instance of an SDME-FSM as for example, management invocation of a shadowing procedure between network operators. In this case, the SDME-Control will create an instance of the SDME-FSM to handle this management-related operation.

Finally, the Functional Entity Access Manager (FEAM) provides the lower-level maintenance functions on the interface as described in clause 4.

## **5.5 Call Control Function/Call-Unrelated Service Function (CCF/CUSF) models**

### **5.5.1 General**

The CUSF model shown in Figure 11 is for a single-ended service logic instance related to a user. The purpose of this model is to provide a framework for the understanding of call-unrelated modelling with respect to the CUSF.



**Figure 11/Q.1248.1 – Internal CUSF structure and the relationship between FEs**

The SCF-CCF/CUSF relationship is used for message between a SCF and a CUSF in the public network. This relationship provides service or service features which need call-unrelated user interaction. This provides, for example, user location registration, user authentication, supplementary service activation or de-activation.

The relationship is used for three cases, and respective cases are exclusive for the others.

Case 1: for sending/receiving ROSE APDUs to/from a user.

Case 2: for sending/receiving USI information to/from a user.

Case 3: for relaying some information for call-unrelated association establishment from a supplementary service in CCF (BCUSM).

### 5.5.2 CCF/CUSF components

The CCF/CUSF model consists of several entities. Each of the entities are briefly described below, with additional details in ITU-T Rec. Q.1248.7. It is noted that this shows a conceptual model of CCF/CUSF and is not intended to imply an actual implementation of the CCF/CUSF.

### **5.5.2.1 Basic Non-Call Manager (BNCM)**

This entity provides an abstraction of an association and proceedings for call-unrelated interactions between a user and a network. It detects basic call-unrelated events that can lead to the invocation of IN service logic instances or those that should be reported to active IN service logic instances. It also manages resources required to support basic call-unrelated control. The BNCM comprises the BCUSM and the DP processing logic. The DP processing logic is the entity of the BNCM that interacts with the FIM/NCM as described in the FIM/NCM description below.

### **5.5.2.2 IN – Non-Switching Manager (IN-NSM)**

The entity which interacts with the SCF via Functional Entity Access Manager in the course of providing IN services/features to users. It provides the SCF an observable view of a CUSF call-unrelated processing activities, and provides the SCF with access to CUSF capabilities, such as managing of an association and invoked operations. It also detects non-call processing events that should be reported to active IN service logic instances or can lead to the invocation of IN service logic instances. It manages CUSF resources required to support IN service logic instances, if any.

### **5.5.2.3 Feature Interaction Manager/Non-Call Manager (FIM/NCM)**

The entity which provides mechanisms to prevent invocation of multiple instances of IN and non-IN service logic instances on a single association. The ability of the FIM/NCM to arbitrate IN/non-IN call associated/call-unrelated associated services/features and IN call-unrelated associated services/features is outside the scope of this capability set. The FIM/NCM integrates these interactions mechanisms with the BNCM and IN-NSM to provide the CUSF with a unified view of call-unrelated processing internal to the CUSF for an association.

## **5.5.3 Relationship of CCF/CUSF model components**

### **5.5.3.1 BNCM relationship to IN-NSM**

This is the relationship that encompassed the interaction between the BNCM and the IN-NSM, through the FIM/NCM. The information flow related to this interaction is not externally visible and is not for standardization. However, an understanding of this subject is required to identify how basic call-unrelated processing and IN call-unrelated processing interact.

### **5.5.3.2 BNCM and IN-NSM relationships to FIM/NCM**

This is the relationships that encompass the interaction between the BNCM and the FIM/NCM, and the IN-NSM and the FIM/NCM. The information flows related to these interactions are not externally visible and are not for standardization. However, an understanding of this subject is required in order to unify the BNCM, IN-NSM and FIM/NCM.

## **5.5.4 CUSF trigger information objects**

This clause provides an overview of the trigger information required for the CUSF.

Detailed description of these objects can be found in ITU-T Rec. Q.1248.7.

The entities/components in the CUSF that are involved in the triggering mechanism are:

- a) Basic Non-Call Manager (BNCM);
- b) IN Non-Switching Manager (IN-NSM);
- c) Feature Interaction Manager/Non-Call Manager (FIM/NCM).

The entities/components shall collectively:

- a) perform the DP processing actions in ITU-T Rec. Q.1248.7;
- b) check for SCF accessibility;
- c) handle service feature interactions.



They shall hand back control to the CCF for at least the following cases:

- a) if a trigger (TDP) criteria match is not found (e.g. insufficient information to proceed): the CUSF logic returns supplementary service control to the CCF;
- b) if the association is abandoned: the CUSF logic returns supplementary service control to the CCF and continues processing;
- c) if the destination SCF is not accessible: the CUSF logic will apply final treatment to the end user for the TDP-R case, or return supplementary service control to the CCF for the TDP-N case.

The CUSF FSM passes component handling instructions to the related instances of the BCUSM as needed. DPs may be dynamically armed as EDPs, requiring the CUSF FSM to remain active. At some point, further interaction with the SCF is not needed, and the CUSF FSM may be terminated while the BCUSM continues to handle the association as needed.

A Configuration Management Function shall be available for managing the trigger tables. The following minimal capabilities shall be provided:

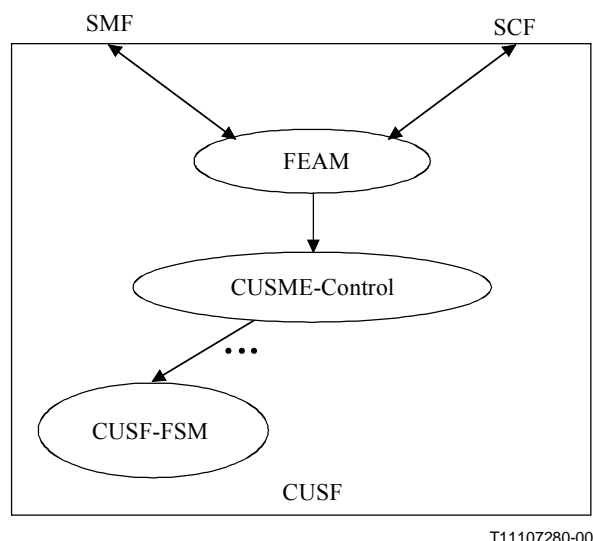
- a) Introduction of new trigger data along with the corresponding service key and revision number;
- b) Remove a trigger data identified by its service key;
- c) Activate or Deactivate existing trigger data;
- d) Verification of the current configuration.

The CUSF trigger information model includes:

- a) Trigger mode (i.e. an indication on whether triggering should cause call-unrelated processing suspension).
- b) Detection Point in the BCUSM.
- c) Trigger Criteria.
- d) Information required for SCF addressing.
- e) Service Key.
- f) Application context name.
- g) Association to a CCF/CUSF user resource (e.g. trunk, line, ...).

### **5.5.5 CUSF FSM structure**

Figure 12 shows the CUSF FSM structure. The interfaces shown in this figure are internal, and are not for standardization.



**Figure 12/Q.1248.1 – CUSF FSM structure**

The relationship between the BCUSM and the CUSF FSM may be described as follows:

- a) When a call-unrelated associated association/operation attempt is initiated by an end user and is processed at an exchange, an instance of a BCUSM is created. As the BCUSM proceeds, it encounters detection points. If a DP is armed as a Trigger DP (TDP), an instance of a CUSF FSM is created.
- b) If an InitiateAssociation is received from the SCF, an instance of a BCUSM is created, as well as an instance of a CUSF FSM.

The management functions related to the execution of operations received from the SCF are executed by the CUSF Management Entity (CUSME)-Control. The CUSME-Control interfaces the different various CUSF FSMs and the Functional Entity Access Manager (FEAM). The Functional Entity Access Manager (FEAM) provides the lower-level maintenance functions on the interface as described in clause 4.

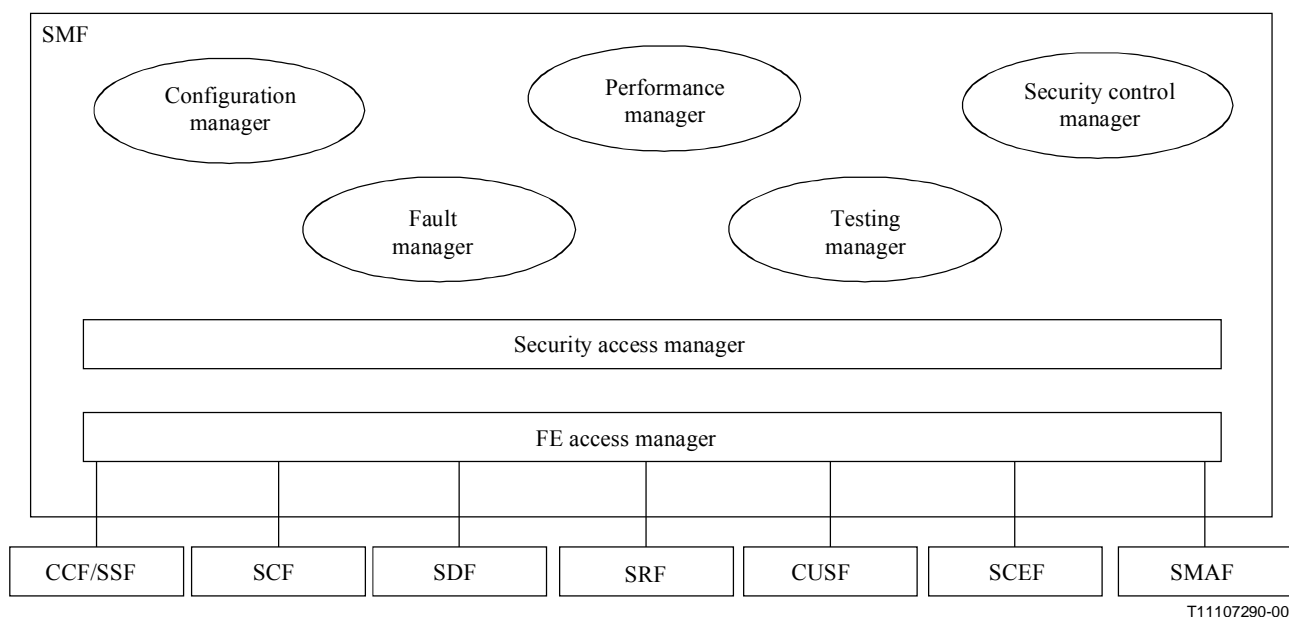
The CUSME-Control maintains the dialogues with the SCF on behalf of all instances of the CUSF Finite State Model (FSM). These instances of the CUSF FSM occur concurrently and asynchronously as associations occur, which explains the need for a single entity that performs the task of creation, invocation, and maintenance of the CUSF FSMs. In particular the CUSME-Control performs the following tasks:

- 1) Interprets the input messages from other FEs and translates them into corresponding CUSF FSM events.
- 2) Translates the CUSF FSM outputs into corresponding messages to other FEs.
- 3) Captures asynchronous (with processing association and/or operation request from the end user) activities related to management or supervisory functions in the CUSF.

## **5.6 Service Management Function (SMF) models**

### **5.6.1 General**

A model of the SMF is shown in Figure 13. The purpose of this model is to provide a framework for the understanding of service deployment functions, service provisioning functions, service operation control functions, and service monitoring functions. Detailed description of the SMF can be found in ITU-T Rec. Q.1246.



**Figure 13/Q.1248.1 – SMF model**

## **5.6.2 SMF components**

The SMF model to provide the required functionality is shown in Figure 13. It is noted that this shows a conceptual model of the SMF and is not intended to imply an actual implementation of the SMF. The SMF platform provides a configuration manager, performance manager, fault manager, security control manager, and testing manager.

### **5.6.2.1 Configuration Manager**

The Configuration Manager is responsible for changing configurations for service deployment and service operation control, or to correct faults.

### **5.6.2.2 Fault Manager**

The Fault Manager analyses and correlates alarms and performs appropriate tests to determine root cause of fault conditions.

### **5.6.2.3 Performance manager**

The performance manager is responsible for gathering the appropriate performance information necessary to gauge the overall performance of service and network resources.

### **5.6.2.4 Testing manager**

The testing manager is responsible for the invocation and management of testing capabilities. The Testing Manager may be used by the Fault Manager in the fault resolution process.

### **5.6.2.5 Security control manager**

The security control manager provides for logging of invalid access attempts and generation of alarms, activation and deactivation of appropriate security features in each FE to enable correct interface operation, and distribution of security-related information.

### **5.6.2.6 Security access manager**

The security access manager provides the functionality needed for secure access to the functions of the IN FEs. This functionality applies security procedures to ensure data confidentiality, origin authentication and data integrity.

### **5.6.2.7 Functional Entity Access Manager (FEAM)**

The FEAM provides the lower level maintenance functions on the interface as described in clause 4.

## **6 Use of FE relationships**

This clause provides a summary of the FE relationships as defined in this Recommendation. A more detailed description can be found in the appropriate interface section of ITU-T Recs. Q.1248.2 to Q.1248.7, and in ITU-T Rec. Q.1236.

Most of the relationships are designed only for use within a trusted network since they are not secure. However, a few internetworking interfaces have been designed to allow data transfer among networks. Where this capability is allowed, it is mentioned in the relationship description below.

### **6.1 SCF-SSF relationship**

The SCF-SSF relationship is used for the transfer of information between an SCF and an SSF in the public network. This relationship, with the aid of the SCF-SDF/SRF, and SDF-SDF relationship, provides a variety of service features like Freephone, UPT, VPN, etc.

### **6.2 SCF-SCF relationship**

While several types of information can be transferred across the SCF-SCF interface, the SCF-SCF relationship is used mainly for the distribution of service logic between two SCFs in the public network. The relationship is used to support call-related operations.

The SCF-SCF relationship is also one of a limited set of relationships to support internetworking. As such, it provides a point of interconnection to the network, effectively hiding the specific structure of the network and providing access security to the network from other public networks.

### **6.3 SCF-IAF relationship**

For INAP purposes, the IAF is treated as an SCF, and therefore has the same attributes as the SCF-SCF relationship.

### **6.4 SRF-CCF relationship**

Current modelling techniques place the CCF under control of the SSF, and therefore any relationship is also with the SSF. An SRF-CCF/SSF relationship exists to manage and convey information such that the bearer channel of a call can be connected to an SRF.

### **6.5 SRF-SCF relationship**

This clause describes the relationship between the SRF and the SCF for connection to and control of specialized resources and User Interaction-scripts.

For all cases in which there is to be call-related interaction, the establishment of this relationship must be preceded by the establishment of a relationship between the SCF and the SSF.

### **6.6 SCF-SDF relationship**

This SCF-SDF relationship is used for information flows between a SCF and a SDF in the public network. The relationship is used mainly for Secured Data Acquisition for service data.

The SCF-SDF relationship supports information flows for services when no call is actually taking place (call-unrelated) as well as during call processing. For the most part, the call-unrelated actions are to support the registration, authentication, encryption and handover procedures for terminal and personal mobility.

The SCF-SDF relationship is also one of a limited set of relationships to support internetworking. As such, it provides a point of interconnection to the network, effectively hiding the specific structure of the network and providing access security to the network from other public networks.

### **6.7 SDF-SDF relationship**

This SDF-SDF relationship is used for supporting data duplication or data replication between two or more SDFs.

The SDF-SDF relationship is also one of a limited set of relationships to support internetworking. As such, it provides a point of interconnection to the network, effectively hiding the specific structure of the network and providing access security to the network from other public networks.

### **6.8 CUSF-SCF relationship**

The SCF-CUSF relationship is used for messages between a SCF and a CUSF in the public network. This relationship provides service or service features which need call-unrelated user interaction. Major features of this relationship are: user location registration, user authentication, supplementary service activation or de-activation.

### **6.9 CUSF-SSF relationship**

The CUSF and the SSF may have a relationship, but the relationship is not specified, and is not mandatory. This relationship may be used to influence the basic call processing in the CCF/SSF (e.g. activation or de-activation of Call Forwarding) via call-unrelated interaction.

### **6.10 SMF-SCF relationship**

This relationship manages the entities/components and data related to the SCF.

This relationship performs:

- a) Configuration Management functions;
- b) Fault Management functions;
- c) Performance Management functions;
- d) Testing functions;
- e) Security Management functions.

### **6.11 SMF-SDF relationship**

This relationship manages the entities/components and data related to the SDF.

This relationship performs:

- a) Configuration Management functions;
- b) Fault Management functions;
- c) Security Management functions.

### **6.12 SMF-CCF/SSF relationship**

This relationship manages the entities/components and data related to the CCF/SSF.

This relationship performs:

- a) Configuration Management functions;
- b) Fault Management functions;
- c) Performance Management functions;

- d) Testing functions;
- e) Security Management functions.

### **6.13 SMF-SRF relationship**

The relationship between the SRF and the SMF deals with the management of the specialized resources.

This relationship performs:

- a) Configuration Management functions;
- b) Fault Management functions;
- c) Performance Management functions;
- d) Testing functions;
- e) Security Management functions.

### **6.14 SMF-SMAF relationship**

There are two kinds of SMAF users, service administrator and service subscriber. Service Administrator consists of both the Service Provider and Network Provider. The relationship manages the entities/components and data related to SMAF.

This relationship provides access to the following functions:

- a) Fault Management functions;
- b) Testing functions;
- c) Performance Management functions;
- d) Security Management functions;
- e) Configuration Management functions.

### **6.15 SMF-SCEF relationship**

The SCEF-SMF relationship conveys the output of the SCEF to the SMF. Once a new service has been installed on the SMF from the SCEF, the task of deploying the service is the role of the SMF and initiated from within the SMF.

This relationship performs:

- a) Configuration Management functions;
- b) Testing functions;
- c) Generic Management Scripting interface;
- d) Security Management functions.

### **6.16 SMF-SMF relationship**

This relationship manages the interworking between entities/components and data residing in different networks. This relationship is the TMN X-interface as defined in ITU-T Rec. M.3320.

The SMF-SMF relationship performs:

- a) Fault Management functions;
- b) Testing functions;
- c) Performance Management functions;
- d) Security Management functions.

## **6.17 SMF-CCF/CUSF relationship**

This relationship manages the entities/components and data related to the CCF/CUSF.

This relationship performs:

- a) Configuration Management functions;
- b) Fault Management functions;
- c) Performance Management functions;
- d) Testing functions;
- e) Security Management functions.

## **7 Protocol realization**

### **7.1 Overview**

For the purpose of communicating with functional entities located in different physical entities, each FE uses one or more Application Entity (AE).

On the following physical interfaces, the AE includes TC (Transaction Capabilities) and one Application Service Element (ASE) for supporting each of the operation packages which may be used during an instance of communication:

- SSP-SCP interface;
- SCP-IP interface;
- SCP-SDP interface;
- SDP-SDP interface;
- SCP-SCP interface;
- SCP-CUSP interface.

Other protocol entity may also be included in an AE for supporting communication on other interfaces (e.g. ISUP).

During an instance of communication, an FE creates an application process instance (API) which has either single interactions or multiple coordinated interactions with application process invocations located in other physical entities. In the first case, a Single Association Control Function (SACF) coordinates the ASEs and their use of the TC services. The SACF together with the ASEs and TC resources involved in one instance of communication form a Single Association Object (SAO). In the second case, a Multiple Association Control Function (MACF) coordinates several SAOs, each of which interact with an SAO in a remote PE.

The MACF (when relevant) together with the SACF(s) and ASEs provide the same functionality than the FEAM, described in clause 4.

AE-types are not intended to be standardized. The organization of a PE in terms of AE is a local matter. Any combination of ASE/AE is acceptable provided that it supports all the application-contexts which realize the contracts defined for the contained ROS-objects (i.e. FEs).

### **7.2 Application contexts**

Operation contracts are realized using the services of the Transaction Capabilities (TC) as defined in ITU-T Rec. Q.771.

The particular set of ASEs involved in the realization of a contract, together with TC and any rules for their coordinated working, constitutes an application-context.

The static aspects of an application context are specified using the APPLICATION-CONTEXT Information Object Class notation as defined in ITU-T Rec. Q.775. This includes the identification of the abstract-syntaxes which are required for representing the information conveyed between the initiator and the responder of the contract.

This Recommendation defines a particular realization of the operation contracts using TCAP; however, other signalling systems may equally be used (e.g. DSS1 Layer 3) providing that all aspects of these operation contracts can be implemented.

### 7.3 Abstract syntax and transfer 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 ITU-T Rec. Q.773;
- b) the abstract syntax for conveying the protocol data units for invoking the operations involved in the operation packages 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 **TCMessages{}** defined in ITU-T Rec. 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 ITU-T Rec. Q.773.

This Recommendation includes a number of network-specific parameters which are defined as being of type OCTET STRING. The internal structure of such parameters may in turn be defined using ASN.1 and use the related Basic Encoding Rules (BER). In such a case the value of this parameter (after the first tag and length information) is the BER encoding of the defined ASN.1 internal structure. The tag of this parameter as defined by ITU-T shall never be replaced.

### 7.4 SACF/MACF rules

#### 7.4.1 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.77X-series Recommendation (*Transaction capabilities application part*) for a more detailed description of the TCAP AC negotiation mechanism.

#### 7.4.2 Sequential/Parallel execution of operations

In some cases it may be necessary to distinguish whether operations should be performed sequentially or in parallel (synchronized). Operations which may be synchronized with any other operation are:

- charging operations.

The method of indicating that operations are to be synchronized 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 synchronized.



In case of inconsistency between the above-mentioned generic rules and the FE-specific rules, as specified in other parts of the Q.1248.x family of ITU-T Recommendations, the FE-specific rules take precedence over the generic rules.

## 8 Congestion control

### 8.1 Lower layer flow control in a SS7 environment

SCCP flow control allows the SCCP User to toggle access to SCCP Subsystems (identified by SSN). This allows the user to turn on, or off, all of the traffic to a particular subsystem within the node. 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 ITU-T Rec. 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, AnalysedInformation, or InitiateCallAttempt) 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.

In addition to this subsystem management, SCCP provides the capability to restrict traffic to the SCCP based on a specific congestion level. This congestion level applies to the total SCCP traffic for the node and is determined locally within the SCCP. Changes to the congestion level are broadcast to SCCP User to enable traffic limitation. Traffic limitation is achieved by comparing the importance parameter passed with SCCP data messages to the current congestion level for the destination node and discarding excess traffic. Judicious choice of the Importance values for TCAP messages would allow restriction of new service requests while allowing existing requests to proceed. This form of traffic limitation would apply to all IN services using the resources of the node under congestion.

In the absence of network operator-defined values, the Importance parameter should be assigned from the default values defined in Table 1.

**Table 1/Q.1248.1 – Proposed Importance assignment for INAP messages**

Traffic type	SCCP Importance setting		
	BEGIN	CONTINUE	END
INAP (SSF→SCF)	0	4	4
INAP (CUSF→SCF)	0	4	4
INAP (SCF→SSF)	4	6	6
INAP (SCF→CUSF)	4	6	6
INAP (SCF-SCF)	4	6	6
INAP (SCF-SDF)	4	6	6
INAP (SCF-SRF)	4	6	6
INAP (SDF-SDF)	4	6	6
→ indicates restricted direction of initial BEGIN message.			
- indicates initial BEGIN can be in either direction.			

## 8.2 Application level flow control

The INAP protocol includes procedures (e.g. call gapping on the SCP-SSP interface) for requesting other entities to reduce the number of new incoming dialogues according to a set of interface-specific criteria. Such procedures can be initiated by a local management function which detects congestion or by an SMF.

For all interfaces, other than the SSF-SCF or CUSF-SCF, the application of flow control mechanisms can potentially interrupt the operation of a Service Logic Program (SLP) executing in the initial SCF. The flow control logic must ensure that the running SLP receives a meaningful error message (e.g. network congestion) as the result of a request being denied due to flow control limitations.

NOTE – This type of handling may not be acceptable if the service is one where the user has already performed a large degree of interaction with the service (e.g. collected authentication information).

The flow control logic can detect network congestion either by analysis of error return and/or operation request timeout patterns or by receipt of an explicit flow control message from the remote node.

The flow control logic should initiate SSF/CUSF level flow control in response to analysis of remote FE flow control to reduce network message flow closer to the source of the traffic.

For each interface, a detailed description is provided in ITU-T Recs. Q.1248.2 to Q.1248.7.

## 9 Protocol mechanisms

### 9.1 Compatibility mechanisms and extensibility rules

#### 9.1.1 Introduction

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

There are three levels of changes:

a) *Minor changes to INAP in future standardized versions*

A minor change can be defined as a change in 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.

b) *Major changes to INAP in future standardized versions*

A major change can be defined as a change in 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.

c) *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. Such extensions may be defined by individual network operators, vendors or any other body (e.g. regional standardization organizations).

## **9.1.2 Definition of INAP compatibility mechanisms**

### **9.1.2.1 Procedures for major additions to INAP**

In order to support the introduction of major functional changes, the protocol allows a synchronization between the two applications with regard to which functionality is to be performed. This synchronization 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 synchronization is to use the AC negotiation procedures provided in ITU-T Rec. Q.773.

### **9.1.2.2 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 unrecognized parameters that occur after the marker, they are ignored (see ITU-T Rec. X.680). These procedures are not applicable to SCF-SDF and SDF-SDF interfaces which have their own rules, inherited from ITU-T Rec. X.519 and defined in ITU-T Rec. Q.1248.4.

### **9.1.2.3 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 open type 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 unrecognized, or reject the message, similar to the comprehension required mechanism described in the previous clause. Refer to ITU-T Rec. Q.1400 for a definition of this mechanism.

In addition to this mechanism, it is also possible to include network specific additions as components of an operation argument: the range of 50 to 59 (both inclusive) tag values is reserved for this usage.

## **9.2 IN generic interface security**

Any interface within the IN functional architecture may have the need to apply security functions to the information flows passing across the interface. This clause defines a generic set of security mechanisms and procedures based on the ITU-T Rec. X.509 Authentication Framework and the Simple Public Key GSS API Mechanism (SPKM) to enable any interfaces to provide suitable secured communications. This clause defines a sub-profile of ITU-T Rec. X.509 and SPKM for use in this capability set. This clause also defines screening requirements for internetworking interfaces.

For this capability set the provision for security functions is required for the SCF-SDF, SDF-SDF, and SCF-SCF interfaces.

### **9.2.1 Interface security requirements**

The security requirements on an IN system could be divided in the two following main families:

a) *Requirements to offer security features (Network access security requirements)*

This covers the various aspects relating to the protection of user access to services and of terminals to networks against attacks at the access interface (for instance user impersonation), by means of security features such as: user/terminal authentication (i.e. the result of a process by which a service user proves his or her identity to an IN system), user profile verification (i.e. the verification that a user is authorized to use a functionality), etc.

b) *Requirements on the internetworking interfaces (Internetworking security requirements)*

This covers the protection of interactions between the various entities and agents involved in the provision of a telecommunication service against attacks at the internal interfaces of the system or even against data stored inside equipment, by means of security features such as: peer entity authentication (i.e. a process which allows a communicating entity to prove its identity to another entity in the network), signalling data or TMN data integrity, non-repudiation, confidentiality, entity profile verification (i.e. the verification that an entity is authorized to use a functionality), INAP protocol screening (i.e. the filtering of INAP operations, parameter and parameter contents), etc.

For each of the above problems, the interfaces have to provide processes to enable data confidentiality, data origin authentication, data integrity and key management. These capabilities are generally described in ITU-T Rec. X.500.

For the internetworking interfaces, the entities involved have to provide the functionality to enable screening of the messages sent and received across the interfaces.

### **9.2.2 INAP screening requirements**

Screening refers to the ability to filter the messages and the contents of the messages sent/received across internetworking interfaces. With each internetwork traffic relation, an INAP screening profile performing the screening actions shall be associated. This screening functionality is part of the FE's Management Entity.

It shall be possible to attach another INAP screening profile to a specific internetwork relation without severe service interruption (i.e. only outstanding transactions may be affected during the replacement of the screening profile and normal traffic handling shall be resumed immediately following the attachment of the new screening profile to the specific traffic relation).

#### **9.2.2.1 INAP Application Context screening requirements**

This Recommendation defines a set of Application Contexts for INAP. INAP Application Context screening ensures the control per internetwork traffic relation of the Application Context of the INAP capability set. Therefore, INAP operations not within the Application Context negotiated shall be rejected. Also INAP parameters per specific operation, not within the Application Context negotiated shall be rejected or ignored.

The following signalling information shall pass the INAP Application Context screening functions unaffected:

- 1) Application Context negotiation information;
- 2) SCCP-related address information elements.

#### **9.2.2.2 INAP protocol screening requirements**

INAP protocol screening ensures the control per internetwork traffic relation of the signalling capabilities and handling of INAP operations, parameters and parameter contents. The INAP screening functions imply manipulations of received and/or sent signalling information per internetwork traffic relation. The screening actions may be different for INAP signalling information sent or received.

For INAP operations, the protocol screening functions shall be able to let the operation pass unchanged or discard the operation depending on the screening profile associated with the traffic relation.

For INAP parameters, the protocol screening functions shall be able to let the parameter pass unchanged or discard the parameter depending on the screening profile associated with the traffic relation.

For INAP parameter contents, the protocol screening functions shall be able to let the parameter content pass unchanged or change the parameter independent on the received/sent content or change the parameter content dependent on the received/sent content.

The following signalling information shall pass the INAP protocol screening functions unaffected:

- 1) information related to the basic operation or parameter structures;
- 2) information related to maintenance functions;
- 3) SCCP-related address information elements.

### 9.2.3 Security procedures and algorithms

Several assumptions are provided to the current interface security procedures to focus the work, but which also provides acceptable levels of security:

- Generic Security Interface shall be based on SPKM as much as possible.
- The number of parameters in messages and procedures should be minimized, because each of them adds to the signalling traffic load and to some processing time.
- Only one type of security shall be applied to the interface. The order of selection will be none, User, Peer (see Table 2):

**Table 2/Q.1248.1 – Categories of security levels**

	<b>Signature</b>	<b>Encryption</b>
None	None	None
User	None	User
	User	None
	User	User
Peer	None	Peer
	Peer	None
	Peer	Peer

- If both user and peer signature/encryption are requested, then user signature/encryption should be used.
- Only the mechanisms such as Encryption and Digital Signatures (including Message Authentication Code (MAC) techniques) shall be used.

#### 9.2.3.1 Authentication procedures

The authentication procedure defines the process of establishing authentication and shall perform the following functions:

- Authenticate the Entity (User/FE) as being allowed to access an FE.
- Establish the level of security for all messages between the Entity and the FE.
- If required, establish a session key for encryption of messages between the Entity and the FE.

The current authentication uses possible procedures defined in ITU-T Rec. X.509, together with a new procedure, SPKM. The primary purpose of ITU-T Rec. X.509 is to provide authentication for a directory service, which includes mechanisms on Simple Authentication, Strong Authentication and SPKM. This clause indicates which aspects of the Directory Authentication should be considered and supported by implementers within the scope of this capability set.

The current authentication procedures are:

- 1) Simple Authentication (as defined in ITU-T Rec. X.509), using Bind operations.
- 2) Strong Authentication (as defined in ITU-T Rec. X.509), using Bind operations with strong credentials, or using SPKM.
- 3) External Procedures, using Bind operations.
- 4) SPKM (Simple Public Key GSS-API Mechanism) (as defined in Rec. ITU-T X.519):
  - The SPKM allows both unilateral and mutual authentication to be accomplished without the use of secure timestamps. This enables environments which do not have access to secure time to nevertheless have access to secure authentication.
  - The SPKM uses Algorithm Identifiers to specify various algorithms (for data confidentiality, data integrity, etc.) to be used by the communicating peers. This allows maximum flexibility for a variety of environments, for future enhancements, and for alternative algorithms.

#### **9.2.3.2 Three-way mutual authentication**

The use of three-way mutual authentication procedures (including SPKM) in the establishment of a security association between two Functional Entities (FEs) implies the use of the Generic Upper Layers Security (GULS) Security Exchange Service Element (SESE) protocol (defined in ITU-T Recs. X.831 and X.832) as described in ITU-T Rec. X.519.

#### **9.2.3.3 Assignment of credentials**

Procedures for assigning credentials to users or FEs are generally defined in ITU-T Rec. X.500.

#### **9.2.4 Mapping of Security Information Flow Definitions to Tokens**

In the case of the SCF-SDF interface, the Context Establishment Tokens will be conveyed in Bind procedure and the Per-Message Tokens will be used in DAP operations.

In the case of the SDF-SDF interface, the Context Establishment Tokens will be conveyed in Bind procedure and the Per-Message Tokens will be used in DSP and DISP operations.

In the case of the SCF-SCF interface, the Context Establishment Tokens will be conveyed in the SCF Bind procedure and the Per-Message Tokens will be used in SCF-SCF operations and Distributed SCF System operations.

#### **9.2.5 Security FSM definitions**

The Security FSMs for two-way and three-way mutual authentication are described in ITU-T Rec. Q.1228. They illustrate how the security procedures are handled across the interface between two FEs.

### **10 Services assumed from lower layers**

This clause provides a general description of services assumed from lower layers. Further descriptions in the context of functional entities are provided in ITU-T Recs. Q.1248.2 to Q.1248.7.

#### **10.1 Services assumed from TCAP**

The application layer protocol defined in this Recommendation is a protocol to provide communication between a pair of application processes. It is represented as communication between a pair of application entities (AEs) using the Transaction Capabilities. The function of an AE is provided by a set of application service elements (ASEs). The interaction between AEs is described in terms of their use of the services provided by the ASEs.

If Application Contexts (AC) are to be used for FE differentiation within a physical node, then the version of TC used must support the dialogue portion of TC (i.e. *White Book* TC).

This requirement applies to all interfaces, not just those used for internetworking.

Table 3 defines which versions of TC are the minimum versions required to support the defined IN interfaces:

**Table 3/Q.1248.1 – Minimum TC Recommendations requirements for INAP interfaces**

Interface	IN CS-3
SSF-SCF	<i>Blue Book</i> (Note)
SCF-SRF	<i>Blue Book</i> (Note)
SCF-SDF	<i>White Book</i> (1993)
SCF-SCF	<i>White Book</i> (1993)
SDF-SDF	<i>White Book</i> (1993)
CUSF-SCF	<i>Blue Book</i> (Note)
NOTE – If AC negotiation is required, then <i>White Book</i> (1993) is the minimum version required.	

### 10.1.1 Common procedures

This clause defines the procedures and mapping which apply between INAP and TC to be used in the absence of specific procedures and mapping instructions for the specific INAP interfaces as defined in subsequent subclauses.

#### 10.1.1.1 Normal procedures

This clause describes the procedures and TCAP primitives that shall be used for transmitting messages between AEs under normal operation.

The INAP, as TC-user, uses only the structured dialogue facility provided by TCAP. The following situations can occur when a message is sent between two physical entities:

- a dialogue shall be established: the TC-user issues a TC-BEGIN request primitive;
- a dialogue shall be maintained: the TC-user issues a TC-CONTINUE request primitive;
- a dialogue shall no longer be maintained: the TC-user issues a TC-END request primitive with either basic end or with pre-arranged end depending on the following conditions:
  - *Basic end*
    - In the case the dialogue is established, operations, leading to a termination of the relationship, can be transmitted by the FE with a TC-END request primitive (basic) in case the FE is not interested in the reception of any ERROR or REJECT components for these sent operations. Once the FE dialogue resources have been released, any ERROR or REJECT components received for these operations will be discarded by TC as described in ITU-T Rec. Q.774.
    - In case the dialogue is established and the FE has received an operation leading to the termination of the relationship, and in case it does not interest the FE to continue dialogue and there is no operation to be sent, a TC-END request primitive (basic) with zero components can be sent from the FE.

- *Pre-arranged end*

In the case where an entity is interested in possible ERROR or REJECT messages on response to sent operations leading to a termination of the relationship, the dialogue is ended with a TC-END request primitive (pre-arranged end) after the last associated operation timer expires. The receiving entity can end the dialogue with a TC-END request primitive (pre-arranged end) after successful processing of these operations (i.e. the relationship is terminated).

In general, the use of pre-arranged end shall be limited to the case for both communicating entities clearly recognizable that peer entity applies pre-arranged end. In all other cases, basic end shall be used;

- a dialogue shall not be established: for class 2 or 4 operations only the sending TC-user issues a TC-BEGIN request primitive and ends the dialogue locally after operation timeout by means of a pre-arranged end. Upon reception of the TC-BEGIN indication primitive, the receiving TC-user shall end the dialogue locally.

### 10.1.1.2 Abnormal procedures

This clause describes the procedures and TCAP primitives that shall be used for reporting abnormal situations between AEs. The error cases are defined in the different interface parts of this Recommendation.

The following primitives shall be used to report abnormal situations:

- operation errors, as defined in the INAP, are reported with TC-U-ERROR request primitive;
- rejection of a TCAP component by the TC-user shall be reported with TC-U-REJECT request primitive;
- when the FE detecting error or rejecting operation decides the termination of TC dialogue, TC-END request primitive (basic) with error or reject can be used for the termination of TC dialogue;
- when the SSF, the SRF, or the CUSF detecting error or rejecting operation recognizes the possibility to continue dialogue, TC-CONTINUE request primitive with error or reject can be used for the continuation of TC dialogue;
- a dialogue shall be aborted by the TC-user with a TC-U-ABORT request primitive;
- on expiration of application timer  $T_{SRF}$  or  $T_{CUSF}$  or  $T_{SSF}$  (in case of the last CS in the CSA), dialogue shall be terminated by means of a TC-U-ABORT primitive with an Abort reason, regardless of TCAP dialogue is established or not.

For abnormal situations detected by TCAP, the same rules shall apply for reception of TC-R-REJECT indication as for transmission of TC-U-REJECT request and for transmission of TC-P-ABORT indication as for transmission of TC-U-ABORT request primitive.

The following rules shall be applied to terminate the TCAP dialogue under abnormal situations:

- in the case that abort condition is detected and TCAP dialogue is established, TCAP dialogue is terminated by TC-U-ABORT primitive with an Abort reason;
- in the case that abort condition is detected and TCAP dialogue is not established, TCAP dialogue is locally terminated by TC-U-ABORT primitive (in the case application time out).

In error situations, pre-arranged end shall not be used to terminate the TCAP dialogue. In case any application entity encounters an error situation, the peer entity shall be explicitly notified of the error, if possible. If from any entity's point of view the error encountered requires the relationship to be ended, it shall close the dialogue via a TC-END request primitive with basic end or via a



TC-U-ABORT request primitive, depending on whether any pending ERROR or REJECT component is to be sent or not.

In case an entity receives a TC-END indication primitive and after all components have been considered, the FSM is not in a state to terminate the relationship, an appropriate internal error should be provided.

In cases a dialogue needs to be closed by the initiating entity before its establishment has been completed (before the first TC indication primitive to the TC-BEGIN request primitive has been received from the responding entity), the TC-user shall issue a TC-END request primitive with pre-arranged end or a TC-U-ABORT request primitive. The result of these primitives will be only local; any subsequent TC indication received for this dialogue will be handled according to the abnormal procedures as specified in ITU-T Rec. Q.774.

### **10.1.1.3 Dialogue handling**

#### **10.1.1.3.1 Dialogue establishment**

The establishment of an INAP dialogue involves two application processes, one that is the dialogue-initiator and one that is the dialogue-responder.

Application context negotiation may not be supported in all physical entities and/or all networks.

This procedure is driven by the following signals:

- a TC-BEGIN request primitive from the dialogue-initiator;
- a TC-BEGIN indication primitive occurring at the responding side;
- the first TC-CONTINUE indication primitive occurring at the initiating side or under specific conditions:
  - a TC-END indication primitive occurring at the initiating side;
  - a TC-U-ABORT indication primitive occurring at the initiating side;
  - a TC-P-ABORT indication primitive occurring at the initiating side.

##### **10.1.1.3.1.1 Sending of a TC-BEGIN request**

Before issuing a TC-BEGIN request primitive, SACF shall store the AC-name and, if present, the user-information parameter.

SACF shall request the invocation of the associated operations using the TC-INVOKE service.

After processing of the last invocation request, SACF shall issue a TC-BEGIN request primitive.

The initiator SACF then waits for a TC indication primitive and will not issue any other requests, except a TC-U-ABORT request or a TC-END request with the release method parameter set to "pre-arranged release".

If no TC indication primitive is expected because no dialogue is to be established according to the rules as stated in detail in the relevant clause of ITU-T Recs. Q.1248.2 to Q.1248.7, SACF will wait for the last associated TCAP operation timer to expire and issue a TC-END request with the release method parameter set to "pre-arranged release".

##### **10.1.1.3.1.2 Receipt of a TC-BEGIN indication**

On receipt of a TC-BEGIN indication primitive, responder SACF shall:

- analyse the application-context-name if included in the primitive. If it is supported, process any other indication primitives received from TC.
- if no dialogue is to be established according to the rules as stated in detail in the relevant clause of ITU-T Recs. Q.1248.2 to Q.1248.7, wait for the last indication primitive from TC

and issue a TC-END request with the release method parameter set to "pre-arranged release".

- if the application-context-name included in the primitive is not supported, issue a TC-U-ABORT request primitive. If an alternative application-context can be offered its name is included in the TC-U-ABORT request primitive.

It is for further study whether or not the application-context-negotiation is limited only for using the TC-U-ABORT primitive.

#### **10.1.1.3.1.3 Receipt of the first TC-CONTINUE indication**

On receipt of the first TC-CONTINUE indication primitive for a dialogue, SACF shall check the value of the application-context-name parameter. If this value matches the one used in the TC-BEGIN request primitive, SACF shall process the following TC component handling indication primitives; otherwise, it shall issue a TC-U-ABORT request primitive.

It is for further study whether or not the application-context-negotiation is limited only for using the TC-U-ABORT primitive.

#### **10.1.1.3.1.4 Receipt of a TC-END indication**

On receipt of a TC-END indication primitive in the dialogue initiated state, SACF shall check the value of the application-context-name parameter. If this value matches the one used in the TC-BEGIN request primitive, then the SACF shall process the following TC component handling indication primitives.

#### **10.1.1.3.1.5 Receipt of a TC-U-ABORT indication**

Receipt of a TC-U-ABORT indication primitive is described as part of user abort procedure (see 10.1.1.3.4). If the abort reason is application-context-name not supported, the responding side may propose an alternative application-context-name in the TC-U-ABORT indication. If an alternative application-context is proposed, the receiving entity shall check this name and if it can be supported a new dialogue may be established.

#### **10.1.1.3.1.6 Receipt of a TC-P-ABORT indication**

Receipt of a TC-P-ABORT indication primitive is described as part of provider abort procedure.

#### **10.1.1.3.2 Dialogue continuation**

Once established, the dialogue is said to be in a continuation phase.

Both application processes can request the transfer of INAP APDUs until one of them requests the termination of the dialogue.

##### **10.1.1.3.2.1 Sending entity**

SACF shall process any component handling request primitives.

After processing the last component handling request primitive, SACF shall issue a TC-CONTINUE request primitive.

##### **10.1.1.3.2.2 Receiving entity**

On receipt of a TC-CONTINUE indication primitive, SACF shall accept zero, one, or several TC component handling indication primitives and process them.

#### **10.1.1.3.3 Dialogue termination**

Both the dialogue-initiator and the dialogue-responder have the ability to request the termination of a dialogue after it has been established when no dialogue is to be established or when a dialogue is

no longer to be maintained according to the rules as stated in detail in the relevant clause of ITU-T Recs. Q.1248.2 to Q.1248.7.

The dialogue termination procedure is driven by the following events:

- a TC-END request primitive;
- a TC-END indication primitive.

#### **10.1.1.3.3.1 Sending of TC-END request**

When the dialogue shall no longer be maintained, SACF shall process any component handling request primitives.

After processing the last component handling request primitive (if any), SACF shall issue a TC-END request primitive with the release method parameter set to "basic end" or "pre-arranged release", according to the rules as stated in detail in the relevant clause of ITU-T Recs. Q.1248.2 to Q.1248.7.

#### **10.1.1.3.3.2 Receipt of a TC-END indication**

On receipt of a TC-END indication primitive, the SACF shall accept any component handling indication primitives and process them.

After processing the last component handling primitive, all dialogue related resources are released.

#### **10.1.1.3.4 User Abort**

Both the dialogue-initiator and the dialogue-responder have the ability to abort a dialogue at any time.

The User Abort procedure is driven by one of the following events:

- a TC-U-ABORT request primitive;
- a TC-U-ABORT indication primitive.

##### **10.1.1.3.4.1 Sending of TC-U-ABORT request**

After issuing a TC-U-ABORT request primitive, all dialogue-related resources are released.

##### **10.1.1.3.4.2 Receipt of a TC-U-ABORT indication**

On receipt of a TC-U-ABORT indication, all dialogue-related resources are released.

#### **10.1.1.3.5 Provider Abort**

TC has the ability to abort a dialogue at both the dialogue-initiator side and the dialogue-responder side.

The Provider Abort procedure is driven by the following event:

- a TC-P-ABORT indication primitive.

##### **10.1.1.3.5.1 Receipt of a TC-P-ABORT indication**

On receipt of a TC-P-ABORT indication, all dialogue-related resources are released.

#### **10.1.1.3.6 Mapping to TC Dialogue primitives**

The TC-UNI service is not used by INAP.

The mapping of parameters onto the TC Dialogue services is as follows:

The use of parameters of the TC-BEGIN service is with the following qualifications:

- The Destination Address parameter of the TC-BEGIN service shall be set to the INAP address of the AE which is to respond to the TC-BEGIN service.

NOTE 1 – The address used in this parameter may be mapped by SCCP address translation to one of a number of alternative AEs.

- The Application Context Name parameter of the TC-BEGIN service shall be set according to the specific interface being used between the initiating AE and the responding AE.
- The Originating Address parameter of the TC-BEGIN service shall be set to the unambiguous INAP address of the AE initiating the TC-BEGIN service.

The use of parameters of the TC-CONTINUE service is with the following qualifications:

- The Application Context Name parameter of the TC-CONTINUE service shall be set to the value of the Application Context Name parameter of the TC-BEGIN service for the same Dialogue ID parameter value.
- If present, the Originating Address parameter of the TC-CONTINUE service shall be set to the unambiguous INAP address of the AE initiating the TC-CONTINUE service. This parameter is only present in the first TC-CONTINUE service after a TC-BEGIN service with the same Dialogue ID parameter value.

The use of parameters of the TC-END service is with the following qualifications:

- The Application Context Name parameter of the TC-END service shall be set to the value of the Application Context Name parameter of the TC-BEGIN service for the same Dialogue ID parameter value. This parameter is only present if the TC-END service is used immediately after the TC-BEGIN service.

The use of parameters of the TC-U-ABORT service is with the following qualifications:

- The Abort Reason parameter of the TC-U-ABORT service shall be used as specified in ITU-T Rec. Q.771.
- The Application Context Name parameter of the TC-U-ABORT service shall be set to either the value used in the TC-BEGIN service or an alternative value which can be used to establish the dialogue between the initiating AE and the responding AE.

NOTE 2 – This parameter is only present if the TC-U-ABORT is the immediate response to a TC-BEGIN indication.

The use of parameters of the TC-P-ABORT service is with the following qualifications:

- The P-Abort parameter of the TC-P-ABORT service is set by TC to indicate the reason why TC aborted the dialogue. It shall take the values as defined in ITU-T Rec. Q.771.

#### **10.1.1.3.7 Default mapping to TC Dialogue parameters**

##### **10.1.1.3.7.1 Dialogue Id**

The value of this parameter is associated with the INAP invocation in an implementation dependent manner. This parameter uniquely identifies a specific TC dialogue to a remote INAP AE for an INAP AE.

##### **10.1.1.3.7.2 Application-context-name**

The Application Context Name parameter is set according to the set of operations which need to be supported by the TC dialogue. The defined Application Context Names can be found in Annex A.

##### **10.1.1.3.7.3 User information**

This parameter may be used by both initiating and responding application process in a network-operator-specific manner.

##### **10.1.1.3.7.4 Component present**

This parameter is used by SACF as described in ITU-T Rec. Q.771.

#### **10.1.1.3.7.5 Termination**

The value of the release method parameter of the TC-END request primitive is set by SACF according to the rules as stated in detail in the relevant clause of ITU-T Recs. Q.1248.2 to Q.1248.7.

#### **10.1.1.3.7.6 Quality of service**

The quality of service of TC request primitives is set by the SACF to the following value:

- sequencing requested;
- return option, this parameter is set by SACF in an implementation-dependent manner.

#### **10.1.1.4 Component handling**

##### **10.1.1.4.1 Procedures for INAP operations**

This clause describes the procedures for INAP operations.

##### **10.1.1.4.1.1 Operation invocation**

SACF shall build an operation argument from the parameters received and request the invocation of the associated operation using the TC-INVOKE procedure. If a linked ID parameter is inserted in the primitive, this indicates a child operation and implies that the operation is linked to a parent operation.

##### **10.1.1.4.1.2 Operation invocation receipt**

On receipt of a TC-INVOKE indication primitive, SACF shall:

- if the operation code does not correspond to an operation supported by the application-context, request the transfer of a reject component using the TC-U-REJECT request primitive, with the appropriate problem code (unrecognized operation);
- if a linked ID is included, perform the following checks: If the operation referred to by the linked ID does not allow linked operations or if the operation code does not correspond to a permitted linked operation, or if the parent operation invocation is not active, issue a TC-U-REJECT request primitive with the appropriate problem code (linked response unexpected or unexpected linked operation);
- if the type of the argument is not the one defined for the operation, request the transfer of a reject component using the TC-U-REJECT request primitive, with the appropriate problem code (mistyped parameter);
- if the operation cannot be invoked because the INAP-related dialogue is about to be released, request the transfer of the reject component using the TC-U-REJECT request primitive with the problem code (Initiating Release);
- if sufficient INAP-related resources are not available to perform the requested operation, request the transfer of a reject component using the TC-U-REJECT request primitive with the problem code (Resource Limitation);
- otherwise, accept the TC-INVOKE indication primitive. If the operation is to be user confirmed, SACF waits for the corresponding response.

##### **10.1.1.4.1.3 Operation response**

For user confirmed operations, SACF shall:

- if no error indication is included in the response to a class 1 or 3 operation, construct a result information element from the parameters received and request its transfer using the TC-RESULT-L service;

- if an error indication is included in the response to a class 1 or 2 operation, construct an error parameter from the parameters received and request its transfer using the TC-U-ERROR request primitive.

#### **10.1.1.4.1.4 Receipt of a response**

On receipt of a TC-RESULT-NL indication, SACF shall:

- request the transfer of a reject component using the TC-U-REJECT request primitive, with the appropriate problem code (mistyped parameter).

On receipt of a TC-RESULT-L indication, SACF shall:

- if the type of the result parameter is not the one defined for the result of this operation, request the transfer of a reject component using the TC-U-REJECT request primitive, with the appropriate problem code (mistyped parameter);
- otherwise, accept the TC-RESULT-L indication primitive.

On receipt of a TC-U-ERROR indication, SACF shall:

- if the error code is not defined for the SACF or is not one associated with the operation referred to by the invoke identifier, request the transfer of a reject component using the TC-U-REJECT request primitive, with the appropriate problem code (unrecognized error or unexpected error);
- if the type of the error parameter is not the one defined for this error, request the transfer of a reject component using the TC-U-REJECT request primitive, with the appropriate problem code (mistyped parameter);
- otherwise, accept the TC-U-ERROR indication primitive.

On receipt of a TC-U-REJECT indication primitive which affects a pending operation, SACF shall:

- accept the TC-U-REJECT indication primitive.

On receipt of a TC-L-REJECT indicating "return result problem, return error unexpected", SACF shall inform the application process.

This event occurs when the local TC detects a protocol error in an incoming component which affects an operation.

When the problem code indicates a general problem, it is considered that the event cannot be related to an active operation even if the invoke Id is provided by TC. This is because it is unclear whether the invoke Id refers to a local or remote invocation.

On receipt of a TC-L-CANCEL indication, the SACF shall:

- if the associated operation is a class 1 operation, inform the application process;
- if the associated operation is a class 2 operation and no linked operations are defined for this operation, ignore the primitive;
- if the associated operation is a class 2 operation and has linked operations but none of them has been invoked, inform the application process;
- if the associated operation is a class 2 operation and a linked operation invocation has already been received in response to this operation, ignore the primitive;
- if the associated operation is a class 3 operation, inform the application process;
- if the associated operation is a class 4 operation, ignore the primitive.

#### 10.1.1.4.1.5 Other events

This clause describes the behaviour of SACF on receipt of a component handling indication primitive which cannot be related to any operation or which does not affect a pending one.

On receipt of a TC-U-REJECT indication primitive which does not affect an active operation (i.e. indicating a return result or return error problem), it is up to the application process to abort, continue or terminate the dialogue, if not already terminated by the sending application process according to the rules as stated in detail in the relevant clause of ITU-T Recs. Q.1248.2 to Q.1248.7. This is also applicable for invoke problems related to a class 4 linked operation.

On receipt of a TC-R-REJECT indication (i.e. when a protocol error has been detected by the peer TC entity) which does not affect an active operation, it is up to the application process to abort, continue or terminate the dialogue, if not already terminated by the sending application process according to the rules as stated in detail in the relevant clause of ITU-T Recs. Q.1248.2 to Q.1248.7.

On receipt of a TC-L-REJECT indication primitive (i.e. when a protocol error has been detected by the local TC entity) which cannot be related to an active operation, it is up to the application process to continue, or to terminate the dialogue and implicitly trigger the transmission of the reject component or to abort the dialogue.

On receipt of a TC-NOTICE indication primitive, which informs the SACF that a message cannot be delivered by the Network Layer, it is for the application process to decide whether to terminate the dialogue or retry.

This primitive can only occur if the Return Option has been set.

#### 10.1.1.4.2 Mapping to TC Component primitives

The mapping of parameters onto the TC Component services is as follows:

The TC-U-CANCEL service is not used.

The TC-RESULT-NL service is not used.

The use of parameters of the TC-INVOKE service is as defined in the relevant subclause of clause 10 with the following qualifications:

- The Operation parameter of the TC-INVOKE service shall contain the *operation.&operationCode* value of the INAP operation to be invoked. The operation must be one of the valid operations supported by the negotiated Application Context for the TC dialogue and must be invocable by the local AE.
- The Parameters parameter of the TC-INVOKE service shall contain a value of the *operation.&ArgumentType* value for the operation being invoked, as specified by the Operation parameter.

The use of parameters of the TC-RESULT-L service is with the following qualifications:

- The Invoke Id parameter of the TC-RESULT-L service shall be set to the value of the Invoke Id parameter of the TC-INVOKE service from the remote AE to which a result is being sent.
- The Operation parameter of the TC-RESULT-L service shall be set to the value of the Operation parameter of the TC-INVOKE service from the remote AE which contains the same Invoke Id Parameter value.
- The Parameters parameter of the TC-RESULT-L service shall contain the *operation.&ResultType* value for the operation result, as specified by the Operation parameter.

The use of parameters of the TC-U-ERROR service is with the following qualifications:

- The Invoke Id parameter of the TC-U-ERROR service shall be set to the value of the Invoke Id parameter of the TC-INVOKE service from the remote AE to which an error is being sent.
- The Error parameter of the TC-U-ERROR service shall be set to the value of the *error.&errorCode* of the error to be sent. It must be one of the errors which is expected for the invoked operation as defined in the *operation.&Errors* specification.
- The Parameters parameter of the TC-U-ERROR service shall be set to the value of the *error.&ParameterType* of the error to be sent, as identified by the Error parameter.

The use of parameters of the TC-U-REJECT service is with the following qualifications:

- The Invoke Id parameter of the TC-U-REJECT service shall be set to the Invoke Id Parameter of the TC component service from the remote AE which is being rejected.

#### **10.1.1.4.3 Default mapping to TC Component parameters**

##### **10.1.1.4.3.1 Invoke Id**

This parameter is set by the sending application process. It represents the unique identity of an instance of an operation which is invoked by an AE within a specific TC dialogue. The TC dialogue is identified by the Dialogue Id parameter.

##### **10.1.1.4.3.2 Linked Id**

This parameter is set by the sending application process. It represents the Invoke Id of an operation which was received from the remote AE for a specific TC dialogue to which the operation being invoked by the local AE is to be linked. This parameter is only present if the original operation invoked by the remote AE is defined as having linked operations. The type of local operation invoked must be the same type as one of the operations defined as being linked.

##### **10.1.1.4.3.3 Dialogue Id**

The value of this parameter is associated with the INAP invocation in an implementation-dependent manner. It represents the identity of the established TC dialogue which will carry the component services between the local AE and the remote AE.

##### **10.1.1.4.3.4 Class**

The value of this parameter is set according to the type of the operation to be invoked according to the operation definitions in the various parts of this Recommendation containing the interface specifications.

##### **10.1.1.4.3.5 Time out**

The value of this parameter is set according to the type of operation invoked.

##### **10.1.1.4.3.6 Last component**

This parameter is used as described in ITU-T Rec. Q.771.

##### **10.1.1.4.3.7 Problem code**

This parameter is used as described in the relevant subclause of clause 10.

##### **10.1.1.4.3.8 Abort reason**

This parameter is used by SACF, and attributes and coding are specified by network operator.



## **10.2 Services assumed from SCCP**

This clause describes the services required from the SCCP that may be used by the IN applications for the IN Application Protocol used between the SSF, SCF and SRF, SDF and CUSF.

The services described are those given in the SCCP post *White Book* Recommendations (1993) and ITU-T Rec. Q.715 (SCCP User Guide) should be consulted to identify possible interworking and compatibility issues between the different SCCP versions.

### **10.2.1 Normal procedures**

The SCCP forms the link between the TC and the transport layer and provides (in conjunction with the transport layer) the network services for the IN applications. The network services provided allow the signalling messages sent by the application to the lower layers to be successfully delivered to the peer application.

### **10.2.2 Service functions from SCCP**

#### **10.2.2.1 SCCP Connectionless services**

The following Connectionless services are expected from the SCCP:

- a) Network Addressing to enable signalling connections between SCCP users;
- b) Sequence Control to enable the SCCP users to invoke "sequence guaranteed" or "sequence not guaranteed" options for a given stream of messages to the same destination;
- c) Segmentation/reassembly of large user messages;
- d) Return Option to enable the SCCP users to invoke "discard message on error" or "return message on error" for a given message not able to be delivered by the SCCP to the destination SCCP user, due to routing or segmentation/reassembly failure;
- e) Congestion control.

The primitives used for the above services are given below.

The N-UNITDATA request and N-UNITDATA indication primitives are used to send and receive data. The parameters of these primitives include the Called and Calling Addresses, Sequence Control, Return Option and User Data with the addressing parameters always mandatory.

The N-NOTICE indication primitive is used to return undelivered data if return option is set and a routing/segmentation error occurs.

#### **10.2.2.1.1 INAP addressing**

The INAP addressing elements consist of information contained within the Calling and the Called Addresses which are sent by the application to TC for use by SCCP.

The application expects the SCCP to route messages by either:

- a) the use of the Destination Point Code (DPC), the Subsystem Number (SSN) and in a SS7 environment the MTP SAP (Service Access Point) instance; or
- b) the use of the Global Title (GT) plus optionally the SSN, DPC and in a SS7 environment the MTP SAP.

The application also specifies to SCCP whether to use Route on SSN or Route on GT for both the Called and Calling Addresses.

If INAP requires additional addressing information, it must be carried in the GT portion of the address specification regardless of which form of routing is specified.

Method a) above may be used when the application is aware of the destination point code and the destination SSN located at that point code to which the message is to be delivered. Within a national

network, different SSNs, according to ITU-T Rec. Q.713, may be allocated for the different network-specific applications, e.g. a SSN may be allocated for a SCF functionality.

Method b) above may be used when a message is to be delivered to a SCCP user which can be identified by the combination of the elements within the GT. An example of the use of this method is when messages have to be delivered between different networks. This method may be used since the originating network is unaware of the point code and SSNs allocations within the destination network. The network that determines the end-node to which the message is to be delivered has to perform a Global Title Translation to derive the destination Point Code and the SSN. If optionally the original address contained the SSN, then this may be used as the destination SSN, or the translation may, if required, provide an appropriate new SSN. Where the destination node is in another network (and is not the gateway node) then the application populates the SSN field with either the SSN in use at the destination or zero.

When GT is used for addressing, the IN application expects that the SCCP supports the following elements as defined in ITU-T Rec. Q.713:

#### **10.2.2.1.1.1 Global Title Indicator**

This indicator specifies the method employed for the formatting of the address information. The format with the indicator value 4 is always used for internetwork connections.

#### **10.2.2.1.1.2 Translation Type**

The Translation Types are defined within ITU-T Rec. Q.713.

#### **10.2.2.1.1.3 Numbering Plan**

- 1) For Addresses used on internetwork interfaces the Numbering Plan must be either Generic Numbering, or E.164 as defined in Annex B/Q.713.
- 2) For Addresses used on all other interfaces, any of the Numbering plans defined in ITU-T Rec. Q.713 may be used if deemed suitable.

#### **10.2.2.1.1.4 Global Title Address information**

This is the actual INAP address information supplied by the application and is encoded as indicated by the encoding scheme.

#### **10.2.2.1.1.5 Encoding scheme**

The application should set the value of the encoding scheme according to the format of the GTAI. The allowed values are defined in ITU-T Rec. Q.713.

The network provider must ensure that any change of GT value during translation preserves any INAP-specific information contained in the initial GT value. The GT translation data in the network must not delete the GT information, if present, from the Address.

This requirement applies to all interfaces, not just those used for internetworking.

If Route on SSN is to be supported from the originating node, then a non-zero internationally standardized SSN is required for international internetworking.

In the absence of a standardized non-zero SSN for INAP services, the use of Route on GT is mandatory from the origin node to the network containing the destination node.

The version of SCCP used to support INAP operations must be at least *White Book* 1992.

#### **10.2.2.1.2 Sequence control**

The application will specify whether SCCP protocol class 0 or 1 is required.

Class 0 is used when the in-sequence delivery of messages to a specific called address is not required.

Class 1 is used when the in-sequence delivery of messages to a specific called address is required.

#### **10.2.2.1.3 Return on Error**

The use of Return on Error mechanism may be required by the IN applications so that the application is aware of messages that have not been delivered to the destination by the SCCP. The return option allows the return of the message that was not delivered due to routing or segmentation/reassembly failure back to the issuing user.

If the return option is invoked by the application and the message is not delivered, then the SCCP specifies the "return reason" as specified in ITU-T Rec. Q.711. The N-NOTICE primitive is used to return the undelivered message to the originating user.

#### **10.2.2.1.4 Segmentation/reassembly**

The application expects that since the SCCP can send up to 260 octets of user data (including the address information and TC-message) in a UDT message (248 octets in a XUDT message performing segmentation and congestion control), segmentation is available for long user data.

Also, the SCCP is expected to perform the reassembly function on received segmented messages and deliver the reassembled user data to the user.

However, it should be noted that even though the theoretical maximum size of SCCP-user data and addresses that can be segmented by the SCCP is 3968 octets, the SCCP user would limit the length to about 2560 octets to allow for the largest known addresses. Note that the application must also allow for the octets used for the TC-message in the 2560 octets.

The IN application does not expect the SCCP to segment the user data into more than 16 segments.

#### **10.2.2.1.5 Congestion Control**

To help control of possible congestion that might occur in the lower layers, the application may assign a value to indicate the importance of the message. The use of this parameter requires the use of SCCP (1996) Recommendations.

Also, there exist other congestion control mechanisms as indicated below in SCCP Management.

#### **10.2.2.2 SCCP connection-oriented services**

The use by IN applications for the connection-oriented services is outside the scope of this capability set.

#### **10.2.2.3 SCCP management in a SS7 environment**

The subsystems used within the IN scenario expect the SCCP to provide management procedures to maintain network performance by re-routing in the event of failure of a subsystem, and in case of network congestion by use of the congestion handling procedure. These procedures have appropriate interactions with the SCCP user as described in ITU-T Rec. Q.711.

To achieve the above, the SCCP is expected to perform the following procedures:

- Signalling point status management (N-PCSTATE, which includes the signalling point prohibited, signalling point allowed, signalling point congested, and local MTP availability subprocedures);
- Subsystem status management (N-STATE, which includes the subsystem prohibited, subsystem allowed, and subsystem status test subprocedures);
- Coordinated state change (N-COORD, which is a procedure which allows a duplicated subsystem to be withdrawn from service without affecting the performance of the network).

## 11 Error definitions

This clause provides a general description of errors related to INAP operations. Detailed error procedure descriptions are defined in ITU-T Recs. Q.1248.2 to Q.1248.7.

### 11.1 AttributeError

#### 11.1.1 Error description

The AttributeError error is sent by the SDF to the SCF or another SDF to report an attribute-related problem. The conditions under which an AttributeError error is to be issued are defined in ITU-T Rec. X.511.

#### 11.1.2 Parameter description

The AttributeError parameters and problem codes are specified in ITU-T Rec. X.511.

#### 11.1.3 Relevant interfaces

The AttributeError error may be reported for operations on the SCF-SDF interface and SDF-SDF interface.

### 11.2 Cancelled

#### 11.2.1 Error description

The Cancelled error gives an indication to the SCF that the cancellation, as it was requested by the SCF, of a specific Operation, has been successful. The SCF is only able to cancel certain predefined SCF→SRF Operations.

#### 11.2.2 Parameter description

None.

#### 11.2.3 Relevant interface

The Cancelled error may be reported for operations on the SCF-SRF interface.

### 11.3 CancelFailed

#### 11.3.1 Error description

The CancelFailed error is returned by Cancel if the cancelling of an Operation, as requested by the SCF, was not successful.

#### 11.3.2 Parameter description

The following parameters may be returned when this error is reported:

- **problem**: This parameter identifies failure reasons:
  - 1) unknownOperation, when the InvokeID of the operation to cancel is not known to SRF (this may also happen in case the operation has already been completed);
  - 2) tooLate, when the invokeID is known but the execution of the operation is in a state that it cannot be Cancelled anymore. For instance the announcement is finished but the SpecializedResourceReport has not been sent to the SCF yet. The conditions for the occurrence of failure reason "tooLate" may be implementation dependent;
  - 3) operationNotCancellable, when the invokeID points to an Operation that the SCF is not allowed to cancel.
- **operation**: This parameter contains the information that the operation failed to be cancelled.

### **11.3.3 Relevant interfaces**

The CancelFailed error may be reported for operations on the SCF-SCF interface and SSF-SRF interface.

## **11.4 ChainingRefused**

### **11.4.1 Error description**

The ChainingRefused error is sent by the chaining terminator (or initiator) supporting SCF to the initiator (or terminator) supporting SCF to reject a chained operation.

### **11.4.2 Parameter description**

None.

### **11.4.3 Relevant interface**

The ChainingRefused error may be reported for operations on the SCF-SCF interface.

## **11.5 DirectoryBindError**

### **11.5.1 Error description**

The DirectoryBindError error is sent by the SDF to the SCF/SDF to report a problem in establishing an authorized relationship. The conditions under which a directoryBindError is to be issued are defined in ITU-T Rec. X.511.

### **11.5.2 Parameter description**

The DirectoryBindError parameters and problem codes are specified in ITU-T Rec. X.511.

### **11.5.3 Relevant interfaces**

The DirectoryBindError may be reported for operations on the SCF-SDF interface and SDF-SDF interface.

## **11.6 DSAReferral**

### **11.6.1 Error description**

The DSAReferral error is sent by the SDF to another SDF to report a problem related to chaining to a chained operation. The conditions under which a DSAReferral error is to be issued are defined in ITU-T Rec. X.518.

### **11.6.2 Parameter description**

The DSAReferral error parameters and problem codes are specified in ITU-T Rec. X.518.

### **11.6.3 Relevant interface**

The DSAReferral error may be reported for operations on the SDF-SDF interface.

## **11.7 ETCFailed**

### **11.7.1 Error description**

The ETCFailed is an error from SSF to SCF, indicating the fact that the establishment of a temporary connection to an assisting SSF or SRF was not successful (e.g. receiving a "Backwards Release" after sending an IAM).

### **11.7.2 Parameter description**

None.

### 11.7.3 Relevant interface

The ETCFailed error may be reported for operations on the SSF-SCF interface.

## 11.8 ExecutionError

### 11.8.1 Error description

The ExecutionError is an error sent from the SDF to the SCF, indicating that the request to execute an entry method has failed. The failure can be due to an incorrect input value or the failure of an internal operation or data access logic associated with the execution of the entry method.

### 11.8.2 Parameter description

The following parameters may be returned when this error is reported:

- **problem**: This parameter identifies the cause of the execute operation failure:
  - missingInputValues is returned if 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.
- **common results**: This parameter contains information which may be present to qualify the outcome of any operation used on the SCF-SDF or SDF-SDF interface.

### 11.8.3 Relevant interfaces

The ExecutionError error may be reported for operations used on the SCF-SDF interface and SDF-SDF interface.

## 11.9 ImproperCallerResponse

### 11.9.1 Error description

The ImproperCallerResponse error is reported when the format of the user input has been checked by the SRF and does not correspond to the required format as it was defined in the initiating Operation.

### 11.9.2 Parameter description

None.

### 11.9.3 Relevant interfaces

The ImproperCallerResponse error may be reported for operations on the SCF-SRF interface and SCF-SCF interface.

## 11.10 MissingCustomerRecord

### 11.10.1 Error description

The MissingCustomerRecord error is reported when the Service Logic Program could not be found in the SCF, because the required customer record does not exist, or the requested Service Logic Program Instance, indicated by the correlationID in "AssistRequestInstructions" does not exist anymore. These two cases should be distinguished as two different error situations, because the error procedure shows that the occurrence of the MissingCustomerRecord error is reported to the maintenance function, but the report to the maintenance function for the occurrence of the former case should be optional because it occurs not only in extraordinary situations but in ordinary situations. For example, the former may occur when the end user dials a missing freephone number.

### **11.10.2 Parameter description**

None.

### **11.10.3 Relevant interfaces**

The MissingCustomerRecord error may be reported for operations on the SSF-SCF interface, SCF-SRF interface, SCF-SCF interface and SCF-CUSF interface.

## **11.11 MissingParameter**

### **11.11.1 Error description**

The MissingParameter error is reported when there is an Error in the received Operation argument. The responding entity cannot start to process the requested Operation because the argument is incorrect: a mandatory parameter (the application shall always return this error in case it is not detected by the ASN.1 decoder) or an expected optional parameter which is essential for the application is not included in the Operation argument.

### **11.11.2 Parameter description**

None.

### **11.11.3 Relevant interfaces**

The MissingParameter error may be reported for operations on the SSF-SCF interface, SCF-SRF interface, SCF-SCF interface and SCF-CUSF interface.

## **11.12 NameError**

### **11.12.1 Error description**

The NameError error is sent by the SDF to the SCF or another SDF to report a problem related to the name of the object. The conditions under which a name error is to be issued are defined in ITU-T Rec. X.511.

### **11.12.2 Parameter description**

The name error parameter and problem codes are specified in ITU-T Rec. X.511.

### **11.12.3 Relevant interfaces**

The NameError may be reported for operations on the SCF-SDF interface and SDF-SDF interface.

## **11.13 ParameterOutOfRange**

### **11.13.1 Error description**

The ParameterOutOfRange error is reported when the responding entity cannot start the processing of the requested Operation because an Error in a parameter of the Operation argument is detected: a parameter value is out of range. This error is applied for the following two cases (when the error is determined by the application):

- 1) For the parameter the type of which is defined with the range of its size, such as INTEGER(x..y), SEQUENCE SIZE(x..y) OF Type. This error is applied when the parameter value is z or the parameter size is z where  $z < x$  or  $z > y$ .
- 2) For the parameter the type of which is defined as list of ENUMERATED value, the ParameterOutOfRange error is applied when the parameter value is not equal to any of the ENUMERATED values in the list.

### **11.13.2 Parameter description**

None.

### 11.13.3 Relevant interfaces

The ParameterOutOfRange error may be reported for operations on the SSF-SCF interface, SCF-SRF interface, SCF-SCF interface and SCF-CUSF interface.

## 11.14 Referral

### 11.14.1 Error description

The Referral error is sent by the SDF to the SCF to report a problem related to service data location. The conditions under which a referral error is to be issued are defined in ITU-T Rec. X.511.

### 11.14.2 Parameter description

The Referral error parameter and problem codes are specified in ITU-T Rec. X.511.

### 11.14.3 Relevant interface

The Referral error may be reported for operations on the SCF-SDF interface.

## 11.15 RequestedInfoError

### 11.15.1 Error description

The RequestedInfoError is an immediate response to the CallInformationRequest operation, indicating that the requested information is not known to the SSF or is not available. RequestedInfoError is used when a specific CCF/SSF cannot offer the information specified with RequestedInformationType but there exists other CCF/SSF that can offer the information.

### 11.15.2 Parameter description

A parameter with the following values may be returned when this error is reported:

- unknownRequestInfo when the requested information is not known to the SSF;
- requestedInfoNotAvailable when the requested information is not available in the SSF.

### 11.15.3 Relevant interface

The RequestedInfoError error may be reported for operation on the SSF-SCF interface.

## 11.16 ScfBindFailure

### 11.16.1 Error description

The ScfBindFailure error is sent by the supporting SCF (or terminator supporting in the chaining case) to the controlling SCF (or initiator supporting SCF) to report a problem in establishing an authorized relationship.

### 11.16.2 Parameter description

The following parameter may be returned when the ScfBindFailure error is reported.

- **FailureReason:** This parameter provides the reason why a bind operation has been rejected. It may convey one of the following kinds of indications:
  - 1) system failure;
  - 2) an scfTaskRefused; or
  - 3) a securityError.

In all cases, a more precise diagnostic is also provided.

### 11.16.3 Relevant interface

The ScfBindFailure error may be reported for operations on the SCF-SCF interface.



## **11.17 ScfReferral**

### **11.17.1 Error description**

The ScfReferral error is sent by the supporting SCF to the controlling SCF or by the chaining terminator to the chaining initiator, in order to report that it is not able to provide the assistance and to provide the address of another SCF.

### **11.17.2 Parameter description**

The following parameters may be returned when this error is reported:

- **tryhere**: This parameter provides the Access Point Information as specified in ITU-T Rec. X.511;
- **securityParameters**: This parameter provides the security parameters as specified in ITU-T Rec. X.511.

### **11.17.3 Relevant interface**

The ScfReferral error may be reported for operations on the SCF-SCF interface.

## **11.18 ScfTaskRefused**

### **11.18.1 Error description**

The ScfTaskRefused error is returned by a physical entity if it was not able to fulfil a specific task as requested by an operation. It is identical to taskRefused when security protection is not activated.

### **11.18.2 Parameter description**

The following parameters may be returned when the ScfTaskRefused error is reported:

- **reason**: This parameter provides the same indications as the taskRefused Error parameter;
- **securityParameter**: This parameter contains information for security protection.

### **11.18.3 Relevant interfaces**

The ScfTaskRefused error may be reported for operations on the SCF-SCF interface.

## **11.19 SecurityError**

### **11.19.1 Error description**

The SecurityError error is sent by the SCF/SDF to the SCF/SDF to report a problem in carrying out an operation for security reasons. The conditions under which a security error is to be issued are defined in ITU-T Rec. X.511 (1997).

### **11.19.2 Parameter description**

The security error parameters and problem codes are specified in ITU-T Rec. X.511.

### **11.19.3 Relevant interfaces**

The SecurityError error may be reported for operations on the SCF-SDF interface, SDF-SDF interface and SCF-SCF interface.

## **11.20 ServiceError**

### **11.20.1 Error description**

The ServiceError error is sent by the SDF to the SCF or another SDF to report a problem related to the provision of the service. The conditions under which a service error is to be issued are defined in ITU-T Rec. X.511.

### **11.20.2 Parameter description**

The ServiceError error parameters and problem codes are specified in ITU-T Rec. X.511.

### **11.20.3 Relevant interfaces**

The ServiceError may be reported for operations on SCF-SDF interface and SDF-SDF interface.

## **11.21 ShadowError**

### **11.21.1 Error description**

The ShadowError error is sent by the SDF to another SDF to report a problem related to shadowing. The conditions under which a shadow error is to be issued are defined in ITU-T Rec. X.525.

### **11.21.2 Parameter description**

The ShadowError error parameters and problem codes are specified in ITU-T Rec. X.525.

### **11.21.3 Relevant interface**

The ShadowError error may be reported for operations on the SDF-SDF interface.

## **11.22 SystemFailure**

### **11.22.1 Error description**

The SystemFailure error is returned by a physical entity if it was not able to fulfil a specific task as requested by an operation, and recovery is not expected to be completed within the current call instance.

### **11.22.2 Parameter description**

A parameter with the following values may be returned when this error is reported:

- unavailable resource;
- a component failure;
- a basic call processing exception;
- a resource status failure;
- an end user failure;
- a screening failure.

### **11.22.3 Relevant interfaces**

The systemFailure error may be reported for operations on the SSF-SCF interface, SCF-SRF interface, SCF-SCF interface and SCF-CUSF interface.

## **11.23 TaskRefused**

### **11.23.1 Error description**

The TaskRefused Error is returned by a physical entity if it was not able to fulfil a specific task as requested by an operation, and recovery is expected to be completed within the current call instance.

### **11.23.2 Parameter description**

A parameter with the following values may be returned when this error is reported:

- generic;
- unobtainable when e.g. the address used in the Connect operation is not obtainable;

- congestion.

### **11.23.3 Relevant interfaces**

The TaskRefused error may be reported for operations on the SSF-SCF interface, SCF-SRF interface and SCF-CUSF interface.

## **11.24 TfcBindError**

### **11.24.1 Error description**

The TfcBindError error is sent by the SCF to the SCF/SDF, and by the SDF to the SDF to report a problem in establishing an authorized relationship. The conditions under which this error is to be issued are similar to those for the directoryBindError defined in ITU-T Rec. X.511.

### **11.24.2 Parameter description**

The following parameters may be returned when this error is reported:

- **versions**: This parameter identifies the protocol version supported by the entity which sent this error;
- **error**: This parameter is either a Service Problem or a Security Problem defined in ITU-T Rec. X.511.

### **11.24.3 Relevant interfaces**

The TfcBindError may be reported for operations on the SCF-SCF interface, SCF-SDF interface and SDF-SDF interface.

## **11.25 UnavailableResource**

### **11.25.1 Error description**

The UnavailableResource error is reported when the SRF is not able to perform its function (i.e. play a certain announcement and/or collect specific user information), and cannot be replaced. A reattempt is not possible.

### **11.25.2 Parameter description**

None.

### **11.25.3 Relevant interface**

The UnavailableResource error may be reported for operations on the SCF-SRF interface.

## **11.26 UnexpectedComponentSequence**

### **11.26.1 Error description**

The UnexpectedComponentSequence error is reported when the responding entity cannot start the processing of the requested operation because a SACF or MACF rule is violated, or the operation could not be processed in the current state of the FSM.

### **11.26.2 Parameter description**

None.

### **11.26.3 Relevant interfaces**

The UnexpectedComponentSequence error may be reported for operations on the SSF-SCF interface, SCF-SRF interface, SCF-SCF interface and SCF-CUSF interface.

## **11.27 UnexpectedDataValue**

### **11.27.1 Error description**

The UnexpectedDataValue is reported when the responding entity cannot complete the processing of the requested operation because a parameter has an unexpected data value.

Note that this error does not overlap with "ParameterOutOfRange".

Example: `startTime DateAndTime ::= -- value indicating January 32 1993, 12:15:01`

The responding entity does not expect this value and responds with "UnexpectedDataValue".

### **11.27.2 Parameter description**

None.

### **11.27.3 Relevant interfaces**

The UnexpectedDataValue error may be reported for operations on the SSF-SCF interface, SCF-SRF interface, SCF-SCF interface and SCF-CUSF interface.

## **11.28 UnexpectedParameter**

### **11.28.1 Error description**

The UnexpectedParameter error is reported when there is an error in the received Operation argument. A valid but unexpected parameter was present in the Operation argument. The presence of this parameter is not consistent with the presence of the other parameters. The responding entity cannot start to process the Operation.

### **11.28.2 Parameter description**

None.

### **11.28.3 Relevant interfaces**

The UnexpectedParameter error may be reported for operations on the SSF-SCF interface, SCF-SRF interface, SCF-SCF interface and SCF-CUSF interface.

## **11.29 UnknownLegID**

### **11.29.1 Error description**

The UnknownLegID error is used to indicate to the SCF that a specific leg, indicated by the LegID parameter value in the operation, is unknown to the SSF.

### **11.29.2 Parameter description**

None.

### **11.29.3 Relevant interfaces**

The UnknownLegID error may be reported for operations on SSF-SCF interface and SCF-CUSF interface.

## **11.30 UnknownResource**

### **11.30.1 Error description**

The UnknownResource error is used to indicate to the SCF that a specific physical resource which is indicated by the ResourceID parameter, is not known to the SSF.

### **11.30.2 Parameter description**

None.

### **11.30.3 Relevant interface**

The UnknownResource error may be reported for operations on the SSF-SCF interface.

## **11.31 UpdateError**

### **11.31.1 Error description**

The UpdateError error is sent by the SDF to the SCF or another SDF to report a problem related to attempts to add, delete, or modify information in the SDF. The conditions under which an update error is to be issued are defined in ITU-T Rec. X.511.

### **11.31.2 Parameter description**

The UpdateError error parameters and problem codes are specified in ITU-T Rec. X.511.

### **11.31.3 Relevant interfaces**

The UpdateError may be reported for operations on the SCF-SDF interface and SDF-SDF interface.

## **Annex A**

### **ASN.1 – Common definitions**

This annex contains the ASN.1 description and is available in electronic form only.

## **Annex B**

### **Overview of IN-CS-4 additions**

This annex summarises the list of additions to the INAP CS-3 protocol that have been included in Q.1248.x series of ITU-T Recommendations.

**B.1 Support of IP-based addressing schemes.** Addition of new parameters for transporting RFC 1738 URLs in the argument of the InitialDP and DP Specific operations, and in the argument of the Connect, ContinueWithArgument, InitiateCallAttempt, SelectRoute and SelectFacility operations.

**B.2 Support of broadband specific parameters** as per ITU-T Rec. Q.1237.

**B.3 Support of explicit control of the impact of CPH operations on signalling relationships**

- Addition of new optional indicators to the SplitLeg, MergeCallSegments and MoveLeg operations to enable service logic programs to explicitly indicate whether the execution of a CPH operation should impact both bearer connections and signalling relationships, or bearer connections only.
- Addition of a new optional indicator to the InitiateCallAttempt operation to enable service logic program to explicitly request that the setup.ind primitive generated by the CCF – as a result of executing the operation – be populated with the information available in the operation argument and in the setup.req primitive received from the Signalling Termination associated with the controlling leg (if any).

**B.4 Call Filtering:** Addition of a Call Filtering operation to enable IN service logic programs to interact with switch-based call gapping, based on E.410 destinations.

**B.5 Route monitoring:** Addition of a MonitorRoute operation to enable IN service logic programs to monitor the number of failures encountered for a particular route.

**B.6 Miscellaneous improvements**

- Support of global call references (as defined in ISUP) in the InitialDP and InitiateCallAttempt operations.
- Addition of CUG-related parameters in the InitialDP, Connect, ContinueWithArgument and InitiateCallAttempt operations.
- Addition of a set of parameters to the argument of the ContinueWithArgument and InitiateCallAttempt operations, in order to ensure a alignment with the Connect operation.
- Extension of the BackwardServiceInteractionInd and ForwardServiceInteractionInd parameters to enable service logic programs to control the execution if the Call Hold and Explicit Call Transfer ISDN supplementary services.
- Addition of the Cause parameter in the oNoAnswerSpecificInfo and tNoAnswerSpecificInfo parameters of the EventReportBCSM operation.
- Evolution of the GapCriteria to allow the combination of existing gap criteria with the SCF Identity.



## SERIES OF ITU-T RECOMMENDATIONS

Series A	Organization of the work of 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	Cable networks and 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
<b>Series Q</b>	<b>Switching and signalling</b>
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 and Internet protocol aspects
Series Z	Languages and general software aspects for telecommunication systems