



INTERNATIONAL TELECOMMUNICATION UNION

ITU-T

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

Q.1248.4

(07/2001)

SERIES Q: SWITCHING AND SIGNALLING
Intelligent Network

Interface Recommendation for Intelligent Network Capability Set 4: SCF-SDF interface

ITU-T Recommendation Q.1248.4

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 SYSTEMS No. 4 AND No. 5	Q.120–Q.249
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
INTELLIGENT NETWORK	Q.1200–Q.1699
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.1248.4

Interface Recommendation for Intelligent Network Capability Set 4: SCF-SDF interface

Summary

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

Within the Q.124x series of ITU-T Recommendations, the Q.1248.x series describes the protocol realizing the Q.1241 Distributed Functional Plane 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.

This Recommendation belongs to the Q.1248.x series of ITU-T Recommendations for IN Capability Set 4. It covers the SCF-SDF interface including the description of the aspects of the SCF and SDF Functional Entities which are relevant to this interface.

This Recommendation includes an electronic attachment containing Annex A, ASN.1 definitions.

Source

ITU-T Recommendation Q.1248.4 was prepared by ITU-T Study Group 11 (2001-2004) and approved under the WTSA Resolution 1 procedure on 13 July 2001.

Keywords

Intelligent Network (IN), Intelligent Network Application Protocol (INAP), IN Capability Set 4 (IN CS-4), Service Control Function (SCF), Service Data Function (SDF).

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 or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, without permission in writing from ITU.

CONTENTS

Page

1	Scope.....	1
2	References.....	1
3	Abbreviations and acronyms	1
4	Introduction.....	1
4.1	SCF-SDF relationship.....	1
4.2	Introduction to the reuse of X.500 for SDF interfaces	1
4.2.1	Alignment between the X.500 concepts and the IN	1
4.2.2	Use of a limited subset of X.500	2
4.3	Versions and the rules for extensibility	3
4.3.1	Version negotiation.....	3
4.3.2	SCF side.....	3
4.3.3	Request processing at the SDF side.....	4
5	Service Control Function (SCF) Model.....	4
6	Service Data Function (SDF) Model	4
6.1	General.....	4
6.2	SDF components.....	5
6.2.1	General.....	5
6.2.2	SDF data manager	5
6.2.3	Functional entity access manager	5
6.2.4	Security manager	6
6.3	Data types handled by SDF	6
7	FSM for SCF	6
7.1	Overview.....	6
7.2	SCME-FSM	7
7.2.1	State 1: "Idle".....	8
7.2.2	State 2: "Bind Pending"	8
7.2.3	State 3: "Active"	8
7.3	SDF-related states (SCSM-SDF).....	9
7.3.1	State 1: "Idle".....	9
7.3.2	State 2: "Wait for subsequent Requests"	9
7.3.3	State 3: "Wait for Bind Result"	10
7.3.4	State 4: "SDF Bound"	10
8	FSM for SDF	10
8.1	Overview.....	10
8.2	SDME-FSM	12



Documents

Document Sharing Hub

ITU-T Rec. Q.1248.4 (07/2001)

iii

	Page
8.2.1 State 1: "Idle"	12
8.2.2 State 2: "Wait for TFC"	12
8.2.3 State 3: "Wait for BindResult"	13
8.2.4 State 4: "Active"	13
8.3 SDF state transition model for SCF-related states	13
8.3.1 State 1: "Idle"	14
8.3.2 State 2: "Bind Pending"	14
8.3.3 State 3: "SCF Bound"	14
9 Detailed operations procedure	15
9.1 IN-Execute procedure	15
9.1.1 General description	15
9.1.2 Parameters	15
9.1.3 Invoking entity (SCF)	16
9.1.4 Responding entity (SDF)	17
9.2 IN-AddEntry procedure	18
9.2.1 General description	18
9.2.2 Parameters	18
9.2.3 Invoking entity (SCF)	19
9.2.4 Responding entity (SDF)	19
9.3 IN-DirectoryBind procedure	20
9.3.1 General description	20
9.3.2 Parameters	20
9.3.3 Invoking Entity (SCF)	20
9.3.4 Responding Entity (SDF)	21
9.4 IN-ModifyEntry procedure	21
9.4.1 General description	21
9.4.2 Parameters	21
9.4.3 Invoking entity (SCF)	22
9.4.4 Responding entity (SDF)	22
9.5 IN-RemoveEntry procedure	23
9.5.1 General description	23
9.5.2 Parameters	23
9.5.3 Invoking entity (SCF)	23
9.5.4 Responding entity (SDF)	24
9.6 IN-Search procedure	24
9.6.1 General description	24
9.6.2 Parameters	24
9.6.3 Invoking entity (SCF)	25
9.6.4 Responding entity (SDF)	26

	Page
9.7 IN-Unbind procedure	26
9.7.1 General description	26
9.7.2 Invoking entity (SCF in Directory Access)	26
9.7.3 Responding entity (SDF in Directory Access)	27
9.7.4 Invoking entity (SDF in traffic flow control)	27
9.7.5 Responding Entity (SCF in traffic flow control)	27
9.8 TFCBind procedure	28
9.8.1 General description	28
9.8.2 Invoking entity (SDF)	28
9.8.3 Responding entity (SCF)	28
9.9 TrafficFlowControl procedure	29
9.9.1 General description	29
9.9.2 Parameters	29
9.9.3 Invoking entity (SDF)	30
9.9.4 Responding entity (SCF)	30
9.10 Operation timer values	30
10 Parameters	31
10.1 BaseObject	31
10.2 Changes	31
10.3 CheckOverspecified	31
10.4 CommonResults	31
10.5 ControlType	31
10.6 Credentials	31
10.7 Duration	32
10.8 Entry	32
10.9 Filter	32
10.10 IN-CommonArguments	32
10.11 Information	33
10.12 Input-assertions	33
10.13 MatchedValuesOnly	34
10.14 Method-id	34
10.15 Null	34
10.16 Object	34
10.17 Output-assertions	34
10.18 SearchAliases	34
10.19 SearchInfo	34
10.20 Security	35

	Page
10.21 Selection.....	35
10.22 Specific-input.....	36
10.23 Specific-output.....	36
10.24 Subset.....	36
10.25 TargetSystem	36
10.26 TfcCriteria.....	36
10.27 UncorrelatedSearchInfo	37
10.28 Versions	37
10.29 WasChained	37
11 Errors	37
11.1 Operation-related error procedures	37
11.1.1 AttributeError	38
11.1.2 ExecutionError.....	38
11.1.3 IN-DirectoryBindError	39
11.1.4 IN-ServiceError	40
11.1.5 NameError	40
11.1.6 Referral	41
11.1.7 SecurityError	41
11.1.8 TFCBindError.....	42
11.1.9 UpdateError	42
12 Service assumed from TCAP.....	43
12.1 Normal procedures.....	43
12.1.1 SCSM/SDSM-related messages	43
12.1.2 SCME/SDME-related messages.....	43
12.2 Abnormal procedures.....	44
12.3 Dialogue handling.....	44
12.3.1 Dialogue establishment.....	44
12.3.2 Dialogue continuation.....	44
12.3.3 Dialogue termination	44
12.3.4 User abort	44
12.3.5 Provider abort	44
12.3.6 Mapping to TC dialogue primitives.....	44
12.4 Component handling.....	47
12.4.1 Procedures for INAP operations.....	47
12.4.2 Mapping to TC component parameters	47
Annex A – ASN.1 definitions (electronic attachment).....	47

ITU-T Recommendation Q.1248.4

Interface Recommendation for Intelligent Network Capability Set 4: SCF-SDF interface¹

1 Scope

This Recommendation belongs to the Q.1248.x series of ITU-T Recommendations for IN Capability Set 4. It specifies the protocol on the SCF-SDF interface and provides a description of the aspects of the SCF and SDF Functional Entities which are involved in the realization of this interface.

2 References

All ITU-T Recommendations and other references are identified in ITU-T Q.1248.1.

3 Abbreviations and acronyms

All abbreviations and acronyms used in this text are defined in ITU-T Q.1248.1.

4 Introduction

4.1 SCF-SDF relationship

This SCF-SDF relationship is used for messages between an SCF and an SDF in the public network. The relationship supports access to service data, which can be for intra-network or inter-network purposes.

The SCF-SDF relationship supports messages 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 inter-networking. 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.

The SCF-SDF relationship is used for various purposes. Each of the purposes can apply to the intra-network or inter-network cases. Specifically, the relationship is used for the Secured Data Acquisition from an SDF by an SCF. In this case, an SCF requires data to complete its processing of service logic. This relationship is the same as the SCF-SDF relationship as defined in IN CS-1 and CS-2 and uses the same operations. It differs from the IN CS-1 and CS-2 relationship in the provision of additional methods for supporting verification of user/terminal credential.

4.2 Introduction to the reuse of X.500 for SDF interfaces

4.2.1 Alignment between the X.500 concepts and the IN

The X.500 series of ITU-T Recommendations are used to specify the SCF-SDF interface and the contents of the SDF. Most of the concepts of the X.500 series are directly used in the IN environment; however, some alignments need to be done at the terminology level to ensure that the

¹ This Recommendation includes an electronic attachment containing Annex A, ASN.1 definitions

concepts introduced in the Directory are correctly understood. The purpose of this subclause is to provide this alignment. It therefore only concentrates on the terms that are ambiguous in the IN environment.

When looking at the structure of the SCF, the Service Data Management is the part of the SCF responsible for the interactions with the SDF. It can be mapped onto the concept of Directory User Agent (DUA). When an SCF on behalf of a user wants to set up an association with an SDF, an instance of a DUA is created in the SLPI. It is killed when the association is ended.

The SDF is the entity responsible for answering the database requests. This functional entity can be mapped onto the Directory System Agent (DSA). When an association is set up between an SCF and an SDF, an instance of a DSA is created for the length of the association.

The Directory is a collection of DSAs/SDFs. This set can be used for a specific service or for a variety of services. The notion of Directory is equivalent to the concept of database systems in IN.

The Directory can also be seen as a repository of data. IN services provide various kinds of data access to users. The information is organized into entries. An entry is a collection of information that can be identified (or named). When it represents an object (i.e. contains primary information about an object), it is called an object entry.

Objects are anything which are identifiable (can be named) and which are of interest to hold information on in the database. A typical example of an object is a user. Objects can be described by several entries. Each individual information that is used to describe an object is an attribute. They are associated to entries.

In the IN environment, the service provider is responsible for the management and the administration of the data contained in a DSA. Therefore, the service provider plays the administrator role. He is the administrative authority in X.500 terminology. The service provider enforces the security procedure (authentication and access control).

4.2.2 Use of a limited subset of X.500

The primary purpose of the X.500 series of ITU-T Recommendations is to provide a directory service and not the description of the SCF-SDF interface as SG 11 wants to use them. The X.500 functionality covers more than the functionality needed for this capability set. The Operation Procedure, Parameters, and ASN.1 definitions try to indicate which aspects of the Directory Abstract Service should be considered and supported by implementors within the scope of this IN capability set. It also mentions the attitude to adopt when a non-supported parameter is received.

Profiling is used as a means to present the status of the different parameters. For convenience and clarity, this profile is defined using ASN.1 subtyping facilities; however, these definitions do not form a protocol specification. The ASN.1 module is described in subclause 12.5 for information. This simply indicates which parameters should not be sent for an implementation. It does not change the behaviour of the receiving entity which shall still be capable of decoding values which conform to the original definition of the Directory Abstract Service. Nevertheless, elements that are excluded by subtyping should be ignored.

It is important to mention that the number of parameters carried in a message should be minimized, because each of them is associated to a load in the signalling traffic and to some processing time. This is the reason why the parameters are removed unless they are absolutely necessary when they are sent. On reception, removed parameters should not be treated but should be understood by the receiving entity. This allows the extensions of the profile in the future according to its actual description in the third edition of the Directory.

Several assumptions were used to design the Directory Abstract Service profile. They are as follows:

Assumption 1: The version of the Directory Abstract Service used for this capability set is the third edition. The parameters only used for the 1988 version shall be ignored. Functionalities that might be needed in future Capabilities Sets should be at least considered if not supported.

Assumption 2: The alias entries in IN are just a means to provide an alternative name for an object and therefore should be dereferenced when needed.

Assumption 3: An SCF-SDF operation cannot be abandoned. If an operation takes too much time, its timer expires and there is no need to abandon it.

4.3 Versions and the rules for extensibility

The Directory may be distributed and more than two Directory Application Entities may interoperate to service a request. The Directory AEs may be implemented conforming to different editions of the Directory specification of the Directory service which may or may not be represented by different protocol version numbers. The version number is negotiated to the highest common version number between two directly binding Directory AEs.

The rules in ITU-T Q.1248.1 are replaced with the rules in this clause.

4.3.1 Version negotiation

When accepting an association, i.e. binding, utilizing the DAP, the version negotiated shall only affect the point-to-point aspects of the protocol exchanged between the SCF and the SDF to which it is connected. Subsequent requests or responses on the dialogue shall be constrained by the version negotiated.

4.3.2 SCF side

4.3.2.1 Request and response processing at the SCF side

The SCF may initiate requests using the highest edition of the specification of that request it supports. If one or more elements of the request are critical, it shall indicate the extension number(s) in the critical Extensions parameter.

NOTE 1 – If the information the extension replaced in a CHOICE, ENUMERATED or INTEGER (used as ENUMERATED) type would be essential for proper operation in an SDF implemented according to an earlier edition of the Specification, it is recommended that the extension be marked critical.

When processing a response, an SCF shall:

- a) ignore all unknown bit name assignments within a bit string; and
- b) ignore all unknown named numbers in an ENUMERATED type or INTEGER type that is being used in the enumerated style, provided the number occurs as an optional element of a SET or SEQUENCE; and
- c) ignore all unknown elements in SETs, at the end of SEQUENCES, or in CHOICES where the CHOICE is itself an optional element of a SET or SEQUENCE.

NOTE 2 – Implementations may as a local option ignore certain additional elements in a Directory PDU. In particular, some unknown named numbers and unknown CHOICES in mandatory elements of SETs and SEQUENCES can be ignored without invalidating the operation. The identification of such elements is for further study.

- d) not consider the receipt of unknown attribute types and attribute values as a protocol violation; and
- e) optionally report the unknown attribute types and attribute values to the user.

4.3.2.2 Extensibility rules for error handling at the SCF side

When processing a known error type with unknown indicated problems and parameters, an SCF shall:

- a) not consider the receipt of unknown indicated problems and parameters as a protocol violation (i.e. it shall not issue a TC-U-REJECT or abort the dialogue); and
- b) optionally report the additional error information to the user.

When processing an unknown error type, an SCF shall:

- a) not consider the receipt of unknown error type as a protocol violation (i.e. it shall not issue a TC-U-REJECT or abort the application association); and
- b) optionally report the error to the user.

4.3.3 Request processing at the SDF side

If any SDF performing an operation detects an element **criticalExtensions** whose semantic is unknown, it shall return an **unavailableCriticalExtension** indication as an **IN-ServiceError**.

NOTE 1 – If a **criticalExtensions** string with one or more zero values is received, this indicates either that the extensions corresponding to the values are not present or are not critical. The presence of a zero value in a **criticalExtensions** string shall not be inferred as either the presence or absence of the corresponding extension in the APDU.

Otherwise, when processing a request from an SCF, an SDF shall:

- a) ignore all unknown bit name assignments within a bit string; and
- b) ignore all unknown named numbers in an ENUMERATED type or INTEGER type that is being used in the enumerated style, provided the number occurs as an optional element of a SET or SEQUENCE; and
- c) ignore all unknown elements in SETs, at the end of SEQUENCES, or in CHOICES where the CHOICE is itself an optional element of a SET or SEQUENCE.

NOTE 2 – Implementations may as a local option ignore certain additional elements in a Directory PDU. In particular, some unknown named numbers and unknown CHOICES in mandatory elements of SETs and SEQUENCES can be ignored without invalidating the operation. The identification of such elements is not specified in this capability set.

5 Service Control Function (SCF) Model

Generic SCF model is described in ITU-T Q.1248.1.

6 Service Data Function (SDF) Model

6.1 General

An FE model of the SDF is shown in Figure 1. The purpose of this model is to provide a framework for service data functionality subjects with respect to the SDF. Clause 6.2 describes the detailed SDF architecture, and clause 6.3 clarifies and classifies data types which are handled by the SDF.

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

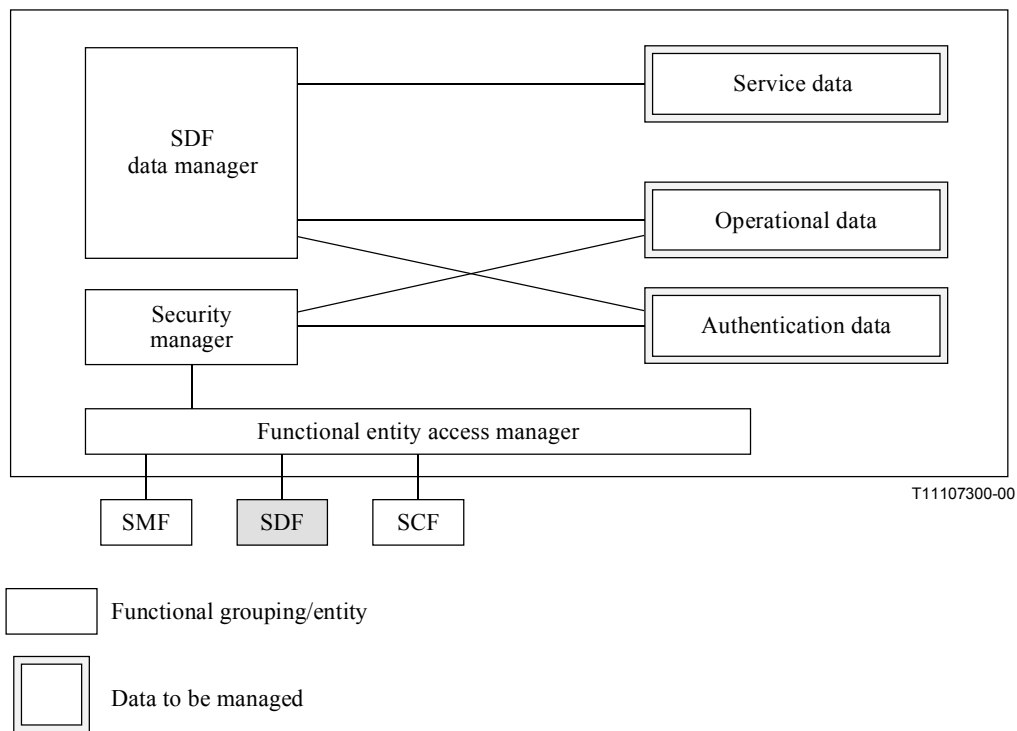


Figure 1/Q.1248.4 – SDF FE model

6.2 SDF components

6.2.1 General

To realize the above defined functionality, the recommended SDF model is shown in Figure 1. Each of the functional entities shown in Figure 1 will be described in the successive clauses. This is not meant to imply any specific implementation.

6.2.2 SDF data manager

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

6.2.3 Functional entity access manager

The functional entity access manager provides the functionality needed by the SDF data manager to exchange information with other functional entities, i.e. SCF, SDF and SMF, via messages. This message handling functionality should:

- provide reliable message transfer;
- ensure sequential message delivery;
- allow message request/response pairs to be correlated;
- allow multiple messages to be associated with each other; and
- comply with OSI structures and principles.

Here, the functional entity access manager may access other SDFs, because the data distribution in the network can completely be transparent to the SCF.

6.2.4 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:

- check the access rights of the SCF;
- authenticate users with provided information;
- 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 in this capability set);
- block data access;
- assign user's access rights;
- memorize user's access rights during his request;
- control user's right to access specific data.

The security manager also provides the functionality needed 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 parameter features are active. The selection of active features is performed through interaction with the SMF.

6.3 Data types handled by SDF

The data which are handled by the SDF can be classified into the following types:

- 1) Authentication data – These data are 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.
- 2) Operational data – These data are not needed by the SLPIs, but are used by the SDF itself for operational and administrative purposes, e.g. references to an object class, access control data.
- 3) Service data – These data are used for the provision of a service, e.g. a subscriber profile, service provider agreements. These data can be used by several services if necessary.

7 FSM for SCF

7.1 Overview

The general SCF FSM structure is depicted in Figure 2. The function of Service Control Function (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, security management, and SLP provisioning management. One of the FSM managed by SCSM is the SDF FSM which handles interaction with SDF. SCME-FSM includes the TFC (Traffic Control) FSM with SDF.

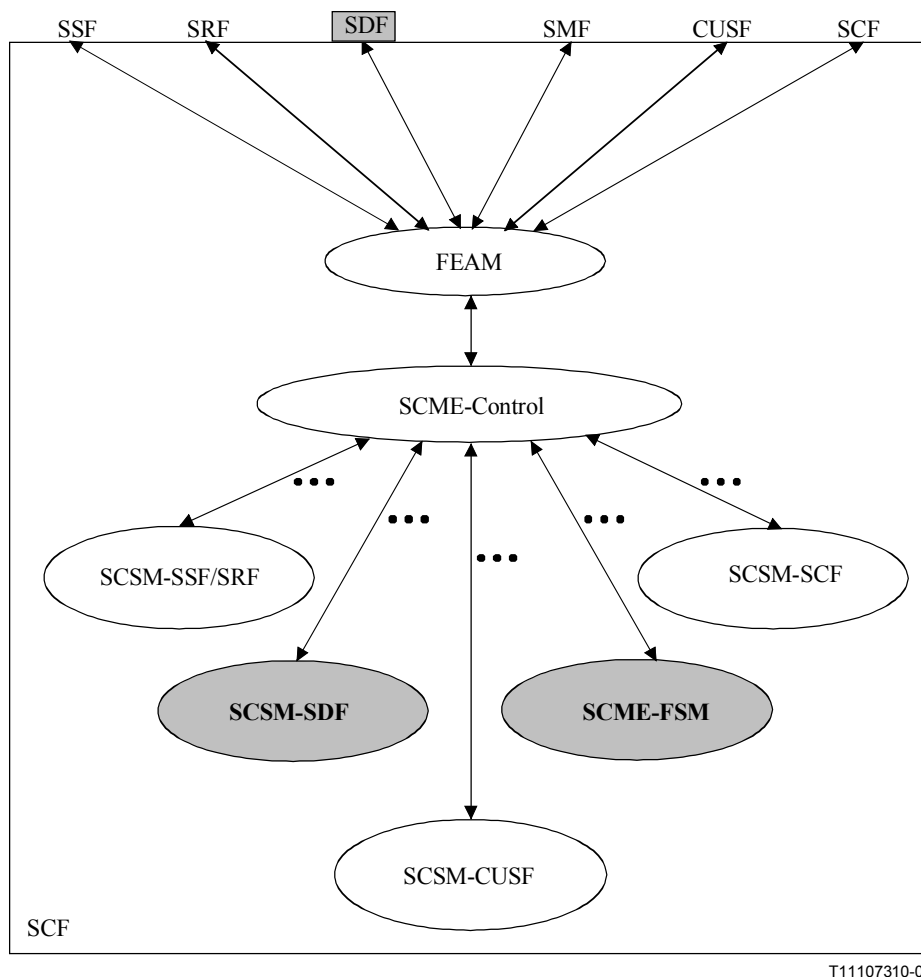


Figure 2/Q.1248.4 – SCF interfaces

7.2 SCME-FSM

The SCME handles a Traffic Flow Control (TFC) related operations in SCF-SDF relationship.

The TFC Responder finite state model (FSM) which is located in SCME-FSM is depicted in Figure 3.

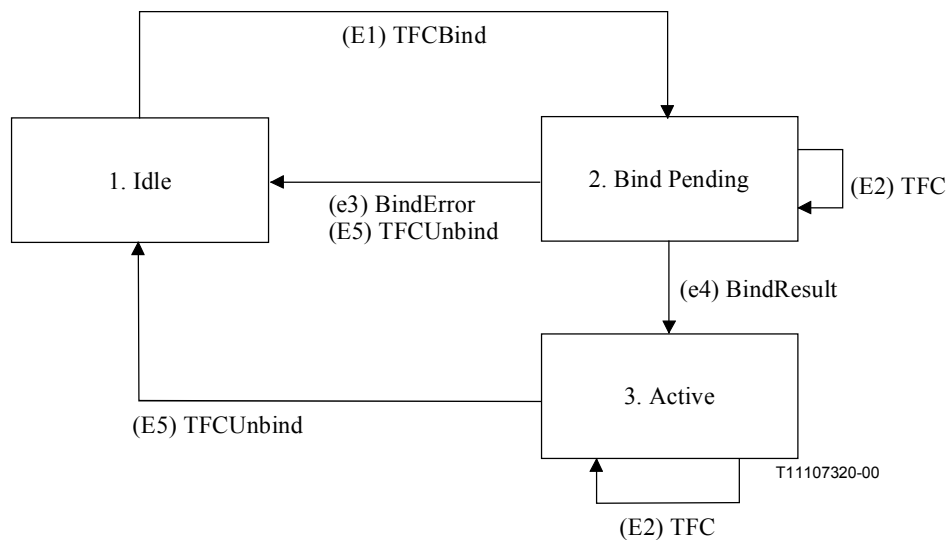


Figure 3/Q.1248.4 – State model for TFC responder

Each state is discussed in one of the following subclauses.

General rules applicable to more than one state are as follows:

In any state, if the dialogue with the Responder is terminated, then the TFC Responder FSM returns to **Idle** state after ensuring that any resources allocated to the dialogue have been deallocated.

7.2.1 State 1: "Idle"

This state is the initial state for the FSM after it has been created by the FE message management. The only event accepted in this state is:

- (E1) TFCBind: This event causes the FSM to notify the Application that a TFC association has been requested by a remote TFC Initiator FSM and causes a transition out of this state to State 2, **Bind Pending**.

7.2.2 State 2: "Bind Pending"

The following events are considered in this state:

- (E2) TFC: This event causes the FSM to save the event to be processed after a transition to another state occurs. The FSM remains in the same state;
- (e3) BindError: This event causes the FSM to construct a **tfcBindError** error message and sends it to the remote TFC Initiator FSM and causes a transition out of this state to State 1, **Idle**;
- (e4) BindResult: This event causes the FSM to construct a **tfcBindResult** message and sends it to the remote TFC Initiator FSM and causes a transition out of this state to State 3, **Active**;
- (E5) TFCUnbind: This event causes the FSM to notify the Application that the association has been terminated by an **IN-Unbind** operation from the remote TFC Initiator FSM and causes a transition out of this state to State 1, **Idle**.

7.2.3 State 3: "Active"

The following events are considered in this state:

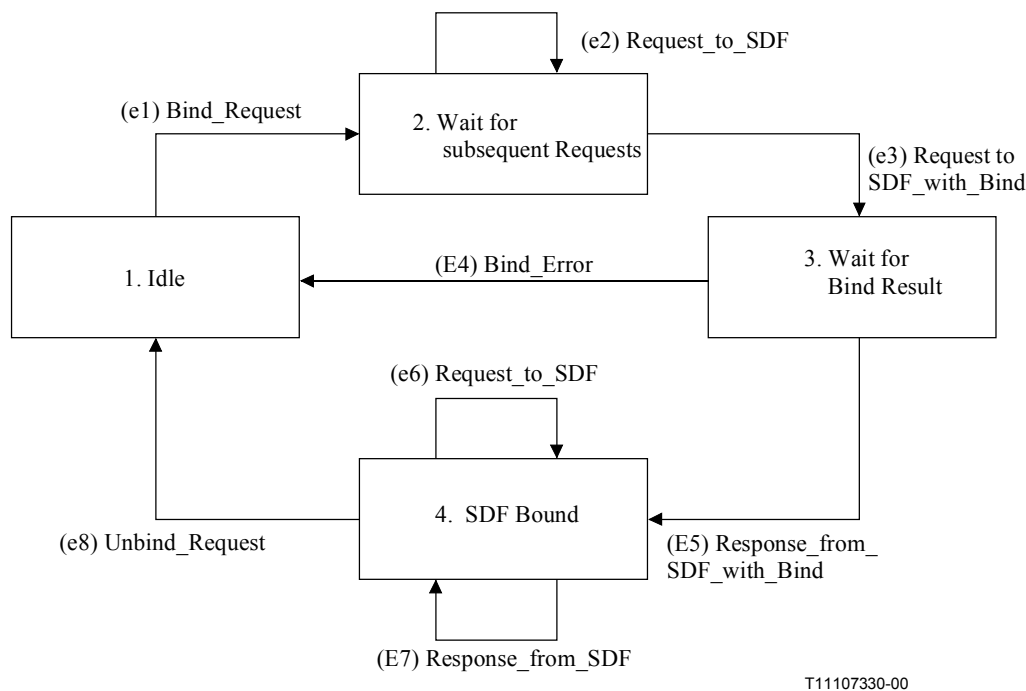
- (E2) TFC: This event causes the FSM to notify the Application of the need to process a new traffic flow control request from the remote TFC Initiator FSM. The FSM remains in the same state; and

- (E5) TFCUnbind: This event causes the FSM to notify the Application that the association has been terminated by an **IN-Unbind** operation from the remote TFC Initiator FSM and causes a transition out of this state to State 1, **Idle**.

7.3 SDF-related states (SCSM-SDF)

The interaction with the SDF is possible from any state of the SCF. In the following subclauses, the SDF-related states are specified. The model describes the relationship of one SCF with one SDF. If an SCF needs to open the new dialogue, a new FSM should be instantiated.

In what follows, the states and events are enumerated independent of the rest of the SCSM; the discussion is accompanied by Figure 4.



T11107330-00

Figure 4/Q.1248.4 – The SDF-related states

7.3.1 State 1: "Idle"

The following event is considered in this state:

- (e1) Bind_Request: This is an internal event, caused by the need of the service logic to create an association with an SDF in order to begin accessing data. This event causes a transition to State 2, **Wait for subsequent Requests**.

7.3.2 State 2: "Wait for subsequent Requests"

In this state, subsequent operations to be sent with the **IN-DirectoryBind** operation (in the same message) to the SDF are expected. The following two events are considered in this state:

- (e2) Request_to_SDF: This is an internal event caused by the reception of an operation. The operation is buffered until the reception of a delimiter (or a timer expiration). The SCSM remains in the same state; and

- (e3) Request_to_SDF_with_Bind: This is an internal event caused by the reception of a delimiter, that indicates the reception of the last operation to be sent. Once the delimiter is received, a message containing the argument of the **IN-DirectoryBind** operation and other operations' arguments, if any, is sent to the SDF. This event causes a transition out of this state to State 3, **Wait for Bind Result**.

7.3.3 State 3: "Wait for Bind Result"

In this state, the SCF is waiting for the response from the SDF. Two events are considered in this state:

- (E4) Bind_Error: This is an external event, caused by the reception of an error to the **IN-DirectoryBind** operation previously issued to the SDF. This event causes a transition out of this state to State 1, **Idle**.
- (E5) Response_from_SDF_with_Bind: This is an external event, caused by the reception of the **IN-DirectoryBind** result combined with the responses to other operations previously issued to the SDF (if any). This event causes a transition to State 4, **SDF Bound**.

7.3.4 State 4: "SDF Bound"

In this state, the SCF has established an access to the SDF, and is waiting for requests to the SDF from the service logic or is waiting for responses to the operations previously issued to the SDF. Three events are considered in this state:

- (e6) Request_to_SDF: This is an internal event, caused by the need of the service logic to access data in the SDF. The SCSM remains in the same state;
- (E7) Response_from_SDF: This is an external event, caused by the reception of responses to the operations previously issued to the SDF. The SCSM remains in the same state; and
- (e8) Unbind_Request: This is an internal event, caused by the need of the service logic to terminate the access to the SDF and by sending an **IN-Unbind** operation. This event causes a transition to State 1, **Idle**.

8 FSM for SDF

8.1 Overview

The structure of the SDF FSMs is illustrated in Figure 5.