

Final draft **ETSI EN 301 931-3** V1.1.2 (2001-07)

---

*European Standard (Telecommunications series)*

**Intelligent Network (IN);  
Intelligent Network Capability Set 3 (CS3);  
Intelligent Network Application Protocol (INAP);  
Protocol specification;  
Part 3: SCF-SRF interface**

---



---

Reference

DEN/SPAN-03063/1-3

---

Keywords

CS3, CTM, IN, INAP, protocol, UPT

**ETSI**

650 Route des Lucioles  
F-06921 Sophia Antipolis Cedex - FRANCE

Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Siret N° 348 623 562 00017 - NAF 742 C  
Association à but non lucratif enregistrée à la  
Sous-Préfecture de Grasse (06) N° 7803/88

---

**Important notice**

Individual copies of the present document can be downloaded from:

<http://www.etsi.org>

The present document may be made available in more than one electronic version or in print. In any case of existing or perceived difference in contents between such versions, the reference version is the Portable Document Format (PDF). In case of dispute, the reference shall be the printing on ETSI printers of the PDF version kept on a specific network drive within ETSI Secretariat.

Users of the present document should be aware that the document may be subject to revision or change of status. Information on the current status of this and other ETSI documents is available at <http://www.etsi.org/tb/status/>

If you find errors in the present document, send your comment to:  
editor@etsi.fr

---

**Copyright Notification**

No part may be reproduced except as authorized by written permission.  
The copyright and the foregoing restriction extend to reproduction in all media.

© European Telecommunications Standards Institute 2001.  
All rights reserved.

# Contents

Intellectual Property Rights .....	9
Foreword .....	9
1 Scope.....	10
2 References .....	10
3 Definitions and abbreviations.....	10
3.1 Definitions .....	10
3.2 Abbreviations.....	10
4 Relationships .....	10
4.1 SRF-CCF relationship .....	10
4.2 SCF-SRF relationship.....	10
5 SCF FE Model.....	10
6 SRF FE Model.....	11
6.1 Resource Control Part (RCP).....	13
6.1.1 SRF Resource Manager.....	13
6.1.2 Transaction Module .....	13
6.1.3 User Interaction-scripts (UI-Script) .....	13
6.1.4 Resource Logic Library.....	13
6.1.5 Resource Logic Instances .....	13
6.2 Resource Function Part (RFP).....	13
6.3 Data Part (DP).....	13
7 SRF application entity procedures.....	14
7.1 The SRF management finite state model (SRME FSM).....	14
7.2 The SRF call State Model: SRSM.....	15
7.2.1 State 1: "Idle".....	17
7.2.2 State 2: "Connected" .....	17
7.2.3 State 3: "User interaction" .....	18
7.3 Example SRF control procedures.....	20
7.3.1 SRF connect procedures.....	20
7.3.1.1 SRF connect physical procedures.....	20
7.3.1.1.1 SSF relay .....	23
7.3.1.1.2 Direct Path SCP to IP .....	23
7.3.1.1.3 Assist with relay.....	24
7.3.1.1.4 Assist without relay .....	24
7.3.1.1.5 Hand-off .....	25
7.3.2 SRF end user interaction procedures.....	25
7.3.2.1 (PA/P&C/P&R).....	27
7.3.2.2 User Interaction Script .....	28
7.3.3 SRF disconnection procedures.....	28
7.3.3.1 SRF initiated disconnect .....	30
7.3.3.2 SCF Initiated Disconnect .....	30
7.3.4 Call-Unrelated Interaction .....	31
7.3.5 Examples Illustrating Complete User Interaction Sequences .....	32
7.3.5.1 SSP with integrated SRF.....	32
7.3.5.2 SSP relays messages between SCP and IP.....	33
7.3.5.3 Direct SCP-IP information transfer .....	35
7.3.5.4 SSP assist(relay SSP).....	36
7.3.5.5 Message sequences for service assist.....	37
7.3.5.6 Message sequences for hand-off.....	38
7.3.6 Example illustrating the use of SDSS .....	39
8 SCF Application Entity procedure.....	40
8.1 The SCF Management state model for SRF.....	40

8.1.1	The Activity Test FSM.....	41
8.1.1.1	State M1: "Activity test idle" .....	41
8.1.1.2	State M2: "Waiting for activity test response" .....	41
8.1.2	The Call GAP FSM.....	41
8.1.2.1	State M1: "idle" .....	42
8.1.2.2	State M3: "Call GAP Active" .....	42
8.2	The SCF Call State Model (SCSM): FSM for specialized resource .....	42
8.2.1	State R1: "SRF Control Idle" .....	43
8.2.2	State R2: "Controlling SRF" .....	44
9	Detailed Operation procedure .....	45
9.1	ActivityTest procedure .....	45
9.1.1	General description .....	45
9.1.2	Parameters .....	45
9.1.2.1	Argument Parameters .....	45
9.1.3	Invoking entity (SCF) .....	45
9.1.3.1	Normal procedure.....	45
9.1.3.2	Error handling .....	45
9.1.4	Responding entity (SRF).....	45
9.1.4.1	Normal procedure.....	45
9.1.4.2	Error handling .....	45
9.2	AssistRequestInstructions procedure.....	46
9.2.1	General description .....	46
9.2.2	Parameters .....	46
9.2.2.1	Argument Parameters .....	46
9.2.3	Invoking entity (SRF) .....	46
9.2.3.1	Normal procedure.....	46
9.2.3.2	Error handling .....	46
9.2.4	Responding entity (SCF).....	46
9.2.4.1	Normal procedure.....	46
9.2.4.2	Error handling .....	47
9.3	Cancel procedure.....	47
9.3.1	General description .....	47
9.3.2	Parameters .....	47
9.3.2.1	Argument Parameters .....	47
9.3.3	Invoking entity (SCF) .....	48
9.3.3.1	Normal procedure.....	48
9.3.3.2	Error handling .....	48
9.3.4	Responding entity (SRF).....	48
9.3.4.1	Normal procedure.....	48
9.3.4.2	Error handling .....	48
9.3.5	Responding entity (SSF) .....	48
9.4	ConnectToResource procedure .....	48
9.5	DisconnectForwardConnection procedure.....	48
9.6	DisconnectForwardConnectionWithArgument procedure.....	49
9.7	EstablishTemporaryConnection procedure .....	49
9.8	PlayAnnouncement procedure .....	49
9.8.1	General description .....	49
9.8.2	Parameters .....	49
9.8.2.1	Argument Parameters .....	49
9.8.3	Invoking entity (SCF) .....	50
9.8.3.1	Normal procedure.....	50
9.8.3.2	Error handling .....	50
9.8.4	Responding entity (SRF).....	50
9.8.4.1	Normal procedure.....	50
9.8.4.2	Error handling .....	51
9.9	PromptAndCollectUserInformation procedure .....	51
9.9.1	General description .....	51
9.9.2	Parameters .....	51
9.9.2.1	Argument Parameters .....	51
9.9.2.2	Result Parameters .....	52
9.9.3	Invoking entity (SCF) .....	52

9.9.3.1	Normal procedure.....	52
9.9.3.2	Error handling .....	52
9.9.4	Responding entity (SRF) .....	53
9.9.4.1	Normal procedure.....	53
9.9.4.2	Error handling .....	54
9.10	PromptAndReceiveMessage procedure .....	54
9.10.1	General description .....	54
9.10.2	Parameters .....	54
9.10.2.1	Argument Parameters .....	54
9.10.2.2	Result Parameters .....	55
9.10.3	Invoking entity (SCF) .....	55
9.10.3.1	Normal procedure.....	55
9.10.3.2	Error handling .....	55
9.10.4	Responding entity (SRF) .....	56
9.10.4.1	Normal procedure.....	56
9.10.4.2	Error handling .....	56
9.11	ReportUTSI procedure .....	56
9.12	RequestReportUTSI procedure .....	56
9.13	SendSTUI procedure .....	57
9.14	ScriptClose procedure.....	57
9.14.1	General description .....	57
9.14.2	Parameters .....	57
9.14.2.1	Argument Parameters .....	57
9.14.3	Invoking entity (SCF) .....	57
9.14.3.1	Normal procedure.....	57
9.14.3.2	Error handling .....	57
9.14.4	Responding entity (SRF) .....	58
9.14.4.1	Normal procedure.....	58
9.14.4.2	Error handling .....	58
9.15	ScriptEvent procedure .....	58
9.15.1	General Description .....	58
9.15.2	Parameters .....	58
9.15.2.1	Argument Parameters .....	58
9.15.3	Invoking entity (SRF) .....	59
9.15.3.1	Normal procedure.....	59
9.15.3.2	Error handling .....	59
9.15.4	Responding entity (SCF) .....	60
9.15.4.1	Normal procedure.....	60
9.15.4.2	Error handling .....	60
9.16	ScriptInformation procedure .....	60
9.16.1	General description .....	60
9.16.2	Parameters .....	60
9.16.2.1	Argument Parameters .....	60
9.16.3	Invoking entity (SCF) .....	61
9.16.3.1	Normal procedure.....	61
9.16.3.2	Error handling .....	61
9.16.4	Responding entity (controlling SRF) .....	61
9.16.4.1	Normal procedure.....	61
9.16.4.2	Error Handling.....	61
9.17	ScriptRun procedure.....	61
9.17.1	General description .....	61
9.17.2	Parameters .....	62
9.17.2.1	Argument Parameters .....	62
9.17.3	Invoking entity (SCF) .....	62
9.17.3.1	Normal procedure.....	62
9.17.3.2	Error handling .....	62
9.17.4	Responding entity (SRF) .....	62
9.17.4.1	Normal procedure.....	62
9.17.4.2	Error handling .....	63
9.18	SpecializedResourceReport procedure .....	63
9.18.1	General description .....	63
9.18.2	Parameters .....	63

9.18.2.1	Argument Parameters .....	63
9.18.3	Invoking entity (SRF) .....	63
9.18.3.1	Normal procedure.....	63
9.18.3.2	Error handling .....	63
9.18.4	Responding entity (SCF).....	64
9.18.4.1	Normal procedure.....	64
9.18.4.2	Error handling .....	64
9.19	SRFCallGap procedure.....	64
9.19.1	General description .....	64
9.19.2	Parameters .....	64
9.19.2.1	Argument Parameters .....	64
9.19.3	Invoking entity (SRF) .....	65
9.19.3.1	Normal procedure.....	65
9.19.3.2	Error handling .....	65
9.19.4	Responding entity (SCF).....	65
9.19.4.1	Normal procedure.....	65
9.19.4.2	Error handling .....	66
10	Parameters .....	66
10.1	CallSegmentID.....	66
10.2	CallSegmentToCancel .....	66
10.3	CollectedInfo .....	66
10.4	ControlType .....	68
10.5	Digits .....	68
10.6	CorrelationID .....	69
10.7	DisconnectFromIPForbidden .....	69
10.8	GapIndicators.....	69
10.9	InformationToRecord .....	69
10.10	InformationToSend .....	70
10.11	LastEventIndicator .....	71
10.12	LegID .....	71
10.13	Media.....	71
10.14	MailBoxID.....	71
10.15	ReceivedStatus .....	71
10.16	RecordedMessageID.....	71
10.17	RecordedMessageUnits .....	71
10.18	RequestAnnouncementComplete .....	72
10.19	SRFgapCriteria .....	72
10.20	SubscriberID .....	72
10.21	UIScriptId .....	72
10.22	UIScriptResultInfo .....	72
10.23	UIScriptSpecificInfo.....	72
11	Error procedures .....	72
11.1	Operation related error procedures .....	72
11.1.1	Cancelled.....	73
11.1.1.1	Operations SCF->SRF .....	73
11.1.1.1.1	Procedures at invoking entity (SCF) .....	73
11.1.1.1.2	Procedures at responding entity (SRF) .....	74
11.1.2	ImproperCallerResponse .....	74
11.1.2.1	Operations SCF->SRF .....	74
11.1.2.1.1	Procedures at invoking entity (SCF) .....	74
11.1.2.1.2	Procedures at responding entity (SRF) .....	74
11.1.3	MissingCustomerRecord .....	75
11.1.3.1	Operations SCF->SRF .....	75
11.1.3.2	Operations SRF->SCF .....	75
11.1.3.2.1	Procedures at invoking entity (SRF) .....	75
11.1.3.2.2	Procedures at responding entity (SCF) .....	75
11.1.4	MissingParameter .....	75
11.1.4.1	Operations SCF->SRF .....	75
11.1.4.1.1	Procedures at invoking entity (SCF) .....	75
11.1.4.1.2	Procedures at responding entity (SRF) .....	76

11.1.4.2	Operations SRF->SCF .....	76
11.1.4.2.1	Procedures at invoking entity (SRF) .....	76
11.1.4.2.2	Procedures at responding entity (SCF) .....	76
11.1.5	ParameterOutOfRange .....	76
11.1.5.1	Operations SCF->SRF .....	76
11.1.6	SystemFailure .....	76
11.1.6.1	Operations SCF->SRF .....	76
11.1.6.2	Operations SRF->SCF .....	77
11.1.7	TaskRefused .....	77
11.1.7.1	Operations SCF->SRF .....	77
11.1.7.2	Operations SCF->SRF .....	77
11.1.7.3	Operations SRF->SCF .....	77
11.1.8	UnavailableResource .....	77
11.1.8.1	Operations SCF->SRF .....	77
11.1.8.2	Operations SCF->SRF .....	77
11.1.8.2.1	Procedures at invoking entity (SCF) .....	77
11.1.8.2.2	Procedures at responding entity (SRF) .....	78
11.1.9	UnexpectedComponentSequence .....	78
11.1.9.1	Operations SCF->SRF .....	78
11.1.9.2	Operations SRF->SCF .....	78
11.1.10	UnexpectedDataValue .....	78
11.1.10.1	Operations SCF->SRF .....	78
11.1.10.2	Operations SCF->SRF .....	78
11.1.10.3	Operations SRF->SCF .....	79
11.1.11	UnexpectedParameter .....	79
11.1.11.1	Operations SCF->SRF .....	79
11.1.11.2	Operations SRF->SCF .....	79
11.1.12	UnknownLegID .....	79
11.1.12.1	Operations SCF->SRF .....	79
11.1.13	UnknownSubscriber .....	79
11.1.13.1	Operations SCF->SRF .....	79
11.1.14	Expiration of TSRF .....	79
11.1.14.1	General Description .....	79
11.1.14.1.1	Error description .....	79
11.1.14.2	Procedures SRF->SCF .....	80
11.1.14.2.1	Procedures at the invoking entity (SRF) .....	80
11.1.14.2.2	Procedures at the responding entity (SCF) .....	80
12	ASN1 Definition .....	80
12.1	IN CS-3 Types .....	80
12.1.1	Data Types .....	80
12.1.2	Classes .....	83
12.2	Operations and Arguments .....	85
12.3	Package, contracts and Application Contexts .....	91
12.3.1	ASN.1 modules .....	91
13	Services assumed from TCAP .....	94
13.1	Normal Procedures .....	94
13.1.1	SCF-to/from-SRF messages .....	94
13.2	Abnormal Procedures .....	94
13.2.1	SCF-to-SRF messages .....	94
13.2.2	SRF/-to-SCF messages .....	95
13.3	Dialogue Handling .....	95
13.3.1	Dialogue Establishment .....	95
13.3.2	Dialogue Continuation .....	95
13.3.3	Dialogue Termination .....	95
13.3.4	User Abort .....	95
13.3.5	Provider Abort .....	95
13.3.6	Mapping to TC Dialogue Primitives .....	95
13.3.7	Component Handling .....	96
13.3.7.1	Procedures for INAP Operations .....	96
13.3.7.2	Mapping to TC Component Parameters .....	96

History .....97



---

## Intellectual Property Rights

IPRs essential or potentially essential to the present document may have been declared to ETSI. The information pertaining to these essential IPRs, if any, is publicly available for **ETSI members and non-members**, and can be found in ETSI SR 000 314: *"Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards"*, which is available from the ETSI Secretariat. Latest updates are available on the ETSI Web server (<http://www.etsi.org/ipr>).

Pursuant to the ETSI IPR Policy, no investigation, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

---

## Foreword

This European Standard (Telecommunications series) has been produced by ETSI Technical Committee Services and Protocols for Advanced Networks (SPAN), and is now submitted for the Vote phase of the ETSI standards Two-step Approval Procedure.

The present document is part 3 of a multi-part deliverable covering Intelligent Network (IN); Intelligent Network Application Protocol (INAP); Capability Set 3 (CS3); Protocol specification, as identified below:

- Part 1: "Common aspects";
- Part 2: "SCF-SSF interface";
- Part 3: "SCF-SRF interface";**
- Part 4: "SDLs for SCF-SSF interface".

The present document and parts 1, 2 and 4 define the Intelligent Network (IN) Application Protocol (INAP) for IN Capability Set 3 (IN CS-3). The present document and parts 1,2 and 4 define the INAP for IN CS-3 based upon ETSI Core INAP CS-2 (EN 301 140-1) and ITU-T IN CS3 Recommendation Q.1238 (1999).

The structure of the present document and parts 1, 2 and 4 follows the ITU-T Recommendation Q.1238 rather than that usual for an ETSI deliverable.

<b>Proposed national transposition dates</b>	
Date of latest announcement of this EN (doa):	3 months after ETSI publication
Date of latest publication of new National Standard or endorsement of this EN (dop/e):	6 months after doa
Date of withdrawal of any conflicting National Standard (dow):	6 months after doa

---

## 1 Scope

The present document is part 3 of a multi-part deliverable for IN CS-3. The part 3 covers the SCF-SRF interface including the description of the aspects of the Functional Entities SRF and SCF which are relevant to this interface.

---

## 2 References

All documents referred to in the present document are identified in EN 301 931-1.

---

## 3 Definitions and abbreviations

### 3.1 Definitions

For the purposes of the present document, the terms and definitions given in EN 301 931-1 apply.

### 3.2 Abbreviations

For the purposes of the present document, the abbreviations and acronyms given in EN 301 931-1 apply.

---

## 4 Relationships

### 4.1 SRF-CCF relationship

At present, it is considered that the CCF is under control of the SSF: a SRF-SSF/CCF relationship exists for connection control to specialized resources.

The relationship between the SRF and the CCF is not defined (e.g. in the case of releasing a connection on which a resource is used). The SRF may contain functionality similar to the CCF to manage bearer connections to specialized resources, but no call model is specified.

### 4.2 SCF-SRF relationship

This SCF-SRF relationship is used when the SCF sends instructions to the SRF.

The SCF-SRF relationship could be a direct link or established via a relay through the SSF.

In some cases, this SCF-SRF relationship is used when the SCF delegates some of the service logic to the SRF, which executes a specialized type of service logic, known as *User Interaction-scripts*. This can avoid long response times, which are unavoidable if functions are physically distributed over two networks nodes, the SCP and the IP.

In assisting scenarios the SRF establishes the relationship towards the SCF.

---

## 5 SCF FE Model

The prime function of the Service Control Function (SCF) is the execution of service logic. Service logics interfaces and interacts with the Service Switching and Call Control Function for establishing End User Interaction to send and receive information. Specialized resources used in the context of End user interaction are managed by the Specialized Resource Function (SRF) and controlled by the Service Control functionality.

---

## 6 SRF FE Model

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.

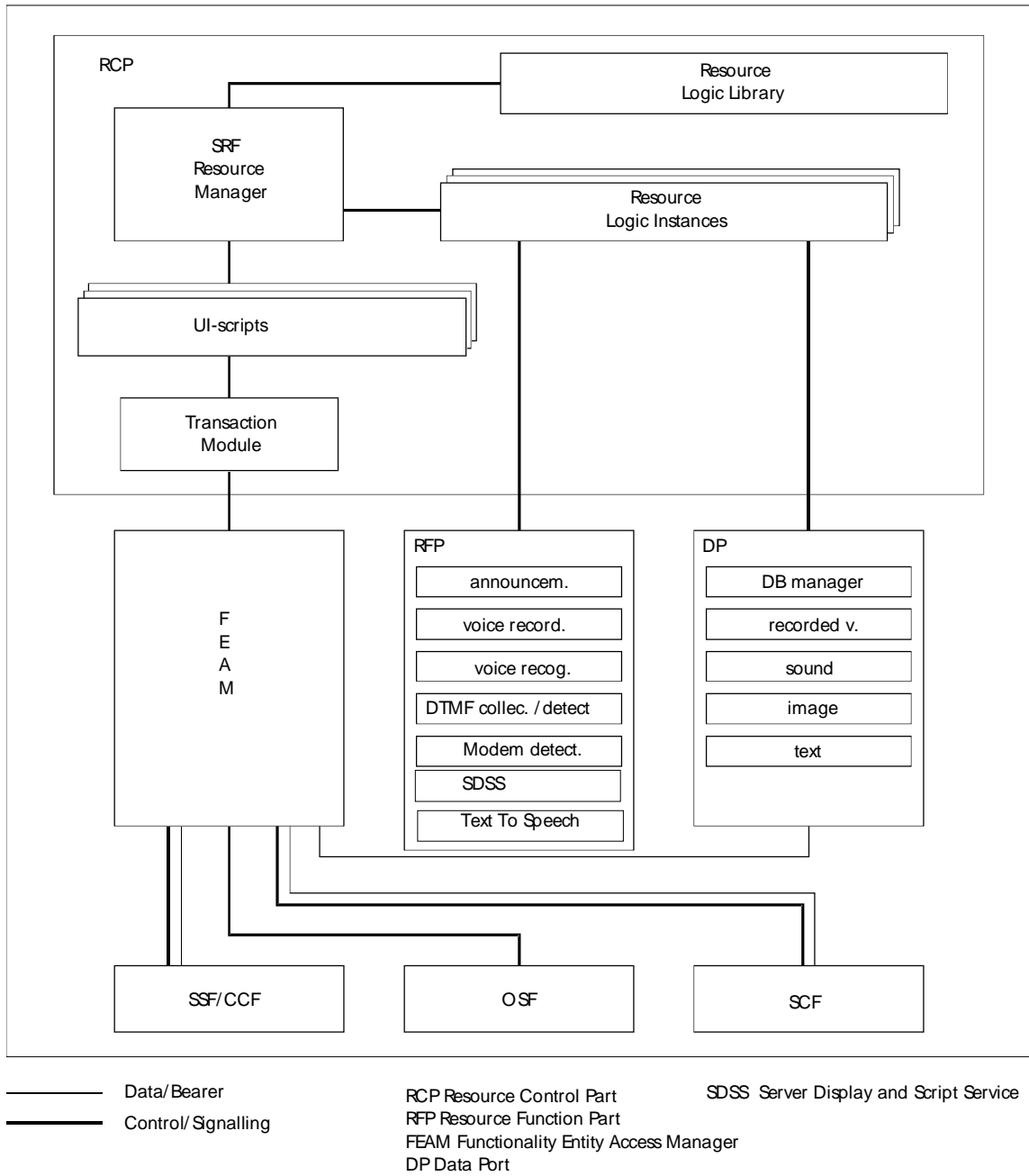
Main SRF components are:

- a) Functional Entity Access Manager (FEAM);
- b) SRF Resource Manager (RM);
- c) Resources.

The SRF Resource Manager is contained in a block called Resource Control Part along with the Resource Logic Library and the Resource Logic Instances; it is possible to split the Resources in the following blocks:

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

Enhancements of the SRF components are described in the following clauses and in figure 1.



**Figure 1: SRF FE Model**

## 6.1 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 DP.

Whenever RCP receives a call requesting a specialized resource-function pair, it invokes the internal resource controller, which manages the first needed resource function pair to make a decision of admission or rejection of the call. The SRF sends ACK or NACK messages according to the decision by the resource controller.

There are as many controllers as there are special resource function types. The controllers accept or reject calls requesting a resource-function pair on the basis of characteristic parameters. A controller consists of an interface unit and a decision unit. First one encodes and decodes messages from/to the FEAM, and makes the input patterns for the decision unit; the characteristic parameter permitting the acceptance or rejection of the call controlled by an algorithm within the decision unit, and is based on the parameters from the interface unit.

### 6.1.1 SRF Resource Manager

The RM provides the functionality which is necessary for the SRF to manage the resources contained in it. The RM contains the capabilities to search for a resource, to allocate or de-allocate it, to manage the status of a resource, and to control its actions.

### 6.1.2 Transaction Module

The Transaction Module provides the functionality necessary for:

- detection of transactions from the communication links;
- routing of transactions to the right applications scripts.

### 6.1.3 User Interaction-scripts (UI-Script)

It provides to the SCF a vision of the different specialized resources functions that the SRF can perform. A User Interaction-script is an aggregation of Resource Function.

### 6.1.4 Resource Logic Library

It indicates the SRF Logic and Physic Resources that are necessary for a given User Interaction-script.

### 6.1.5 Resource Logic Instances

It instances the SRF resources that are necessary for the correct execution of the invoked specialized resource.

## 6.2 Resource Function Part (RFP)

The RFP is a collection of resource-function pairs or functional elements of resources. Resources in a resource-function pair for a service procedure, are allocated and released together.

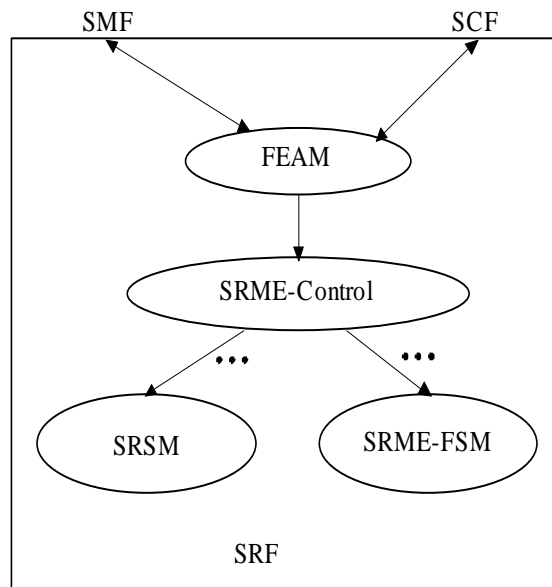
## 6.3 Data Part (DP)

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

## 7 SRF application entity procedures

As described in EN 301 931-1, the SRF-FSM handles interactions with the SSF FSM and the SCF FSM entity. The SRME-control interfaces to the various SRF call state models (SRSM) and the functional entity access manager (FEAM). The SRF-FSM structure is described in figure 2. The FEAM is described in EN 301 931-1.

General tasks of the SRME control is defined in EN 301 931-1. In addition to the general tasks, the SRME-control checks the existence of a SCF-SRF relationship by receiving an Activitytest operation from the SCF and returns the result to the SCF.



**Figure 2: SRF FSM Structure**

In the following clauses the SRSM and SRME-FSM are described.

### 7.1 The SRF management finite state model (SRME FSM)

The SRME handles the following operation:

- sRFCallGap.

The sRFCallGap operation is issued within a context of an existing relationship and doesn't cause state transitions in the SRME.

All other operations have no effect on the SRME-FSMs; the operations are passed by the SRME-Control to the relevant FSM.

## 7.2 The SRF call State Model: SRSM

The SRSM is presented in figure 3. In what follows, each state is described in a separate clause together with the events that cause a transition out of this state. Finally, the outputs are presented within smaller rectangles than the states are; unlike the states and events, the outputs are not enumerated.

Each state is discussed in the following clauses. General rules applicable to more than one state are addressed here.

Received messages may include a single operation or multiple operations, and it is processed as follows:

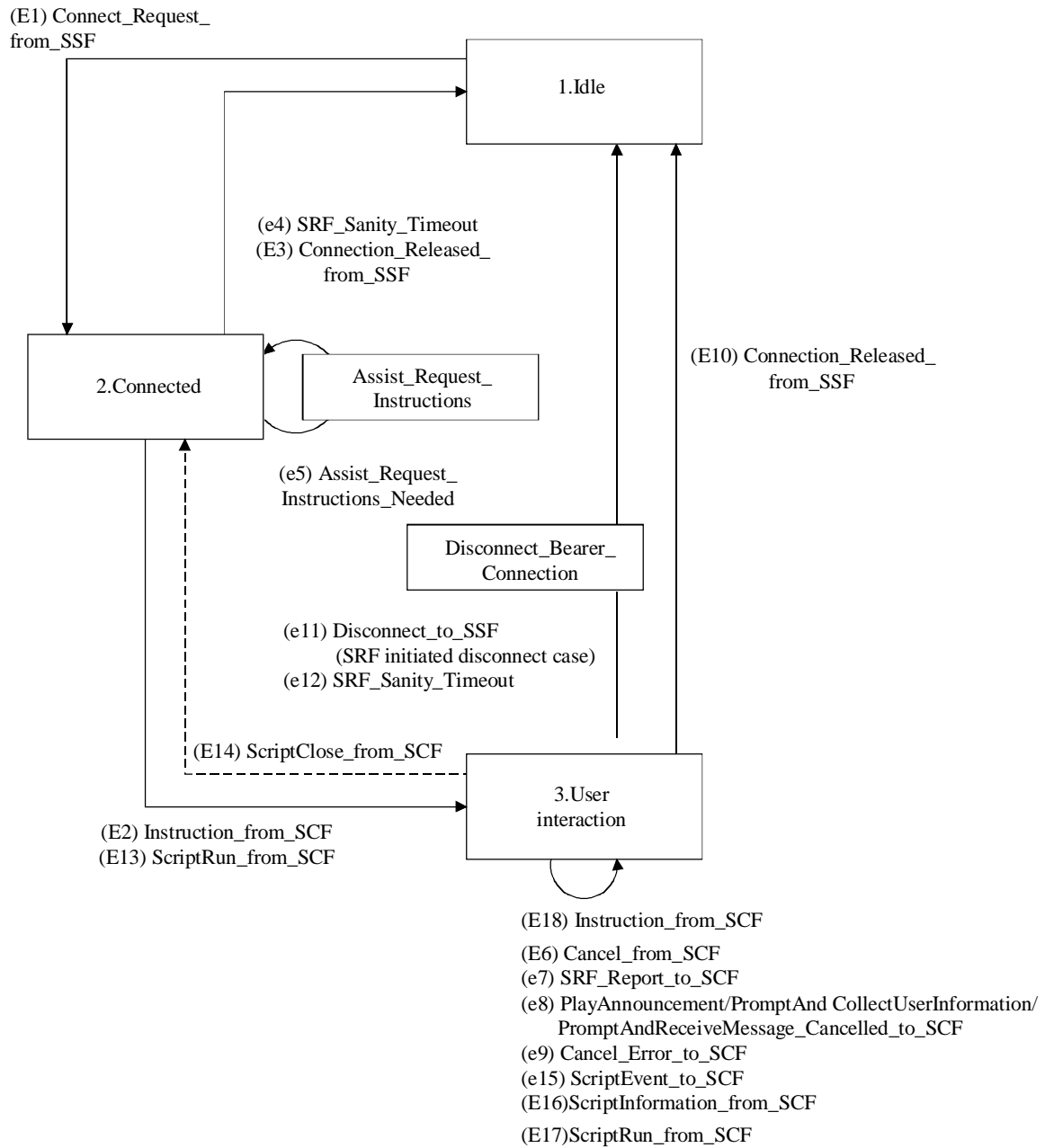
- The SRSM processes the operations in the order in which they are received.
- The SRSM examines subsequent operations in the sequence. When a Cancel (for PlayAnnouncement, PromptAndCollectUserInformation or PromptAndReceiveMessage) operation is encountered in the sequence in state "user interaction", it executes it immediately. In all other cases, the SRSM queues the operations and awaits an event (such an event would be the completion of the operation being executed, or reception of an external event).
- If there is an error in processing one of the operations in the sequence, the SRF FSM processes the error (see below) and discards all remaining operations in the sequence.
- If an operation is not understood or is out of context (i.e. it violates the SACF rules defined by the SRSM) as described above, the SRF FSM processes the error according to the rules given in EN 301 931-1 (using TC-U-REJECT or the operation error UnexpectedComponentSequence).

In any state, if there is an error in a received operation, the maintenance functions are informed. Generally, the SRSM remains in the same state in which it received the erroneous operations, however different error treatment is possible in specific cases as described in EN 301 931-1; depending on the class of the operation, the error could be reported by the SRF to the SCF using the appropriate component (see ETS 300 287-1).

In any state, if the dialogue with the SCF (direct SCF-SRF case) is terminated, then the SRSM returns to idle state after ensuring that all resources allocated to the dialogue have been de-allocated. The SRF shall remain connected to the SSF as long as it has PlayAnnouncement operations active or buffered. Depending on the type of script, the SRF may also remain connected when a script is active or when a SriptRun is buffered. The resources allocated to the call will be de-allocated when all announcements are completed or when the SSF disconnects the bearer connection (i.e. call party release).

In any state (except "idle"), if the SSF disconnects the bearer connection to the SRF before the SRF completes the user interaction, then the SRSM clears the call and ensures that all SRF resources allocated to the call have been de-allocated. Then it transits to the "idle" state.

The SRSM has an application timer,  $T_{\text{SRF}}$ , whose purpose is to prevent excessive unnecessary resource allocation. This timer is set when the SRF sends Setup Response bearer message to the SSF (SSF relay case) or the AssistRequestInstructions operation (Direct SCF-SRF case). This timer is stopped when a request is received from the SCF. The SRF may reset  $T_{\text{SRF}}$  on transmission of the SpecializedResourceReport, ScriptEvent and ScriptClose operation, the return result for the PromptAndCollectUserInformation operation or the return result for the PromptAndReceiveMessage operation when there is no queued user interaction operation. On the expiration of  $T_{\text{SRF}}$ , the SRSM transits to the "idle" state ensuring that all SRF resources allocated to the call have been de-allocated.



**Figure 3: The SRS**



## 7.2.1 State 1: "Idle"

The "idle" state represents the condition prior to, or at the completion of, an instance of user interaction. This state is entered as a result of events E3, e4, E10, e11 and e12:

- (E3) `Connection_Released_from_SSF`: This event takes place when the SRSM receives a release message from the SSF in connected state. The SRSM goes to state "Idle".
- (e4) `SRF_Sanity_Timeout`: This event occurs when the SRSM has been in connected state for a network-operator-defined period of time (timer  $T_{SRF}$ ) without having an SCF initiated operation to execute. The SRF initiates a bearer channel disconnect sequence to the SSF using the applicable bearer channel signalling system. The SRSM goes to state "Idle".
- (E10) `Connection_Released_from_SSF`: This event takes place when the SRSM receives a release message from the SSF in "user interaction" state. The SRSM goes to state "Idle".
- (e11) `Disconnect_to_SSF`: This event occurs when the SCF has enabled SRF initiated disconnect by:
  - the last `PlayAnnouncement/PromptAndCollectUserInformation/PromptAndReceiveMessage/ScriptRun` from SCF (E18) or (E2) with the parameter `disconnectFromIPForbidden` set to false. The SRSM initiates a bearer channel disconnect sequence to the SSF using the applicable bearer channel signalling system after sending the last `SpecializedResourceReport` operation, a return result for `PromptAndCollectUserInformation` or a return result for `PromptAndReceiveMessage` or last `ScriptEvent` operation to the SCF. The SRSM goes to state "Idle".
- (e12) `SRF_Sanity_Timeout`: This event occurs when the SRSM has been in User interaction state for a network-operator-defined period of time (timer  $T_{SRF}$ ) without having an SCF initiated operation to execute. The SRF initiates a bearer channel disconnect sequence to the SSF using the applicable bearer channel signalling system. The SRSM goes to state "Idle".

The following event may be received and leads to exit the current state:

- (E1) `Connect_Request_from_SSF`: This event corresponds to a bearer signalling connection request message from the SSF. The details of the bearer signalling state machine related to establishing the connection are not of interest to the FSM. The SRSM goes to state "Connected".

## 7.2.2 State 2: "Connected"

This state represents the condition of the SRSM when a bearer channel has been established between a user and the SRF but the initial `PlayAnnouncement/PromptAndCollectUserInformation/PromptAndReceiveMessage/ScriptRun` has not yet been received (e.g. when `EstablishTemporaryConnection` procedures are used). The method used to provide this bearer channel is not of interest in the FSM.

This state is entered as result of:

- (E1) `Connect_Request_from_SSF`: This event corresponds to a bearer signalling connection request message from the SSF in the "Idle" state.
- (E14) `ScriptClose_from_SCF`: This event takes place when the `ScriptClose` operation from the SCF is received.
- (e15) `Final_implicit_ScriptEvent_to_SCF`: This event takes place when the final Result of the User Interaction script execution is sent from the SRF to the SCF. It covers the implicit termination case when automatic disconnection is either not allowed or not specified.

In this state, the following events may be received and do not lead to FSM state change state:

- (e5) Assist\_Request\_Instructions\_Needed: This event occurs when the AssistRequestInstructions operation is sent from the SRSM to the SCF in the absence of an operation concatenated with the Setup request from SSF (E1) (Direct SCF-SRF case). No state change occurs as a result of this event.
- (E14) ScriptClose\_from\_SCF: This event takes place when the ScriptClose operation from the SCF is received. The SRSM remains in the state "Connected".

NOTE: This state transition is permitted only if the SRSM received ScriptRun operation from SCF previously.

This state is exited as result of one of the following events:

- (E2) First\_Instruction\_from\_SCF: This event takes place when the first PlayAnnouncement, PromptAndCollectUserInformation or PromptAndReceiveMessage or ScriptRun operation from the SCF is received. The SRSM goes to state "User interaction".
- (E3) Connection\_Released\_from\_SSF: This event takes place when the SRF receives a release message from the SSF. The SRSM goes to state "Idle".
- (e4) SRF\_Sanity\_Timeout: This event occurs when the SRSM has been connected for a network-operator-defined period of time (timer T<sub>SRF</sub>) without having an SCF initiated operation to execute. The SRSM initiates a bearer channel disconnect sequence to the SSF using the applicable bearer channel signalling system. The SRSM goes to state "Idle".
- (E13) ScriptRun\_from\_SCF: This event takes place when the ScriptRun operation from the SCF is received. The SRSM goes to state "User Interaction".

### 7.2.3 State 3: "User interaction"

The "User interaction" state indicates that communication is occurring between the user and the SRF via the bearer channel established at the "Connected" state.

This state is entered as a result of event E2 and E13:

- (E2) First\_Instruction\_from\_SCF: This event takes place when the first PlayAnnouncement, PromptAndCollectUserInformation or PromptAndReceiveMessage or ScriptRun operation from the SCF is received.
- (E13) ScriptRun\_from\_SCF: This event takes place when the ScriptRun operation from the SCF is received.

In this state, the following events may be received and do not lead to FSM state change:

- (E18) Subsequent\_Instruction\_from\_SCF: This event takes place when subsequent PlayAnnouncement, PromptAndCollectUserInformation, or PromptAndReceiveMessage operation(s) from the SCF are received. The SRSM remains in state "User interaction". Event E18 also represents additional PlayAnnouncement/PromptAndCollectUser Information/PromptAndReceiveMessage operations which are buffered as discussed in the procedures.
- (E6) Cancel\_from\_SCF (for PlayAnnouncement/PromptAndCollectUserInformation /PromptAndReceiveMessage): This event takes place when the corresponding PlayAnnouncement, PromptAndCollectUserInformation or PromptAndReceiveMessage operation is received from the SCF. The indicated interaction is terminated if it is presently running, otherwise it is deleted from the buffer. The SRSM remains in state F "User interaction".
- (e7) SRF\_Report\_to\_SCF: This event takes place when a SpecializedResourceReport, a return result for PromptAndCollectUserInformation or a return result for PromptAndReceiveMessage operation is sent to the SCF. The SRSM remains in state "User interaction".

- (e8) *PlayAnnouncement/PromptAndCollectUserInformation/PromptAndReceiveMessage\_Cancelled\_to\_SCF*: This event takes place when the *PlayAnnouncement/PromptAndCollectUserInformation/PromptAndReceiveMessage* error caused by the *Cancel* (for *PlayAnnouncement*, *PromptAndCollectUserInformation* or *PromptAndReceiveMessage*) operation is sent to the SCF. This event represents the successful cancellation of an active or buffered *PlayAnnouncement/PromptAndCollectUserInformation/PromptAndReceiveMessage* operation. The SRSM remains in state "User interaction".
- (e9) *Cancel\_Error\_to\_SCF*: This event takes place when the cancel error (for *PlayAnnouncement*, *PromptAndCollectUserInformation* or *PromptAndReceiveMessage*) is sent to the SCF. This event represents the unsuccessful cancellation of a *PlayAnnouncement/PromptAndCollectUserInformation/PromptAndReceiveMessage* operation. The SRSM remains in state "User interaction".
- (e19) *Intermediate\_ScriptEvent\_to\_SCF*: This event takes place when a partial result is sent from a SRF to the SCF in case the SRF needs additional information. The SRSM remains in the state "User Interaction".
- (e20) *Final\_explicit\_ScriptEvent\_to\_SCF*: This event takes place when the final result of the User Interaction script is sent from the SRF to the SCF. It covers the explicit termination case. The SRSM remains in the state "User Interaction".
- (E16) *ScriptInformation\_from\_SCF*: This event takes place when the *ScriptInformation* operation from the SCF is received. The SRSM remains in the state "User Interaction".
- (E17) *ScriptRun\_from\_SCF*: This event takes place when the *ScriptRun* operation from the SCF is received. The SRSM remains in state "User Interaction". It follows a *PlayAnnouncement*, *PromptAndCollectUserInformation* or *PromptAndReceiveMessage* which is terminated. No other User Interaction Script shall be already active for the call.

NOTE 1: A subsequent *ScriptRun* operation from the SCF is not permitted in this state.

This state is exited as result of one of the following events:

- (E10) *Connection\_Released\_from\_SSF*: This event takes place when the SRSM receives a release message from the SSF. The SRSM goes to state "Idle".
- (e11) *Disconnect\_to\_SSF*: This event occurs when the SCF has enabled SRF initiated disconnect by:
  - the last *PlayAnnouncement/PromptAndCollectUserInformation/PromptAndReceiveMessage/ScriptRun* from SCF (E18) or (E2) with the parameter *DisconnectFromIPForbidden* set to false. The SRSM initiates a bearer channel disconnect sequence to the SSF using the applicable bearer channel signalling system after sending the last *SpecializedResourceReport* operation, a return result for *PromptAndCollectUserInformation* or a return result for *PromptAndReceiveMessage* or last *ScriptEvent* operation to the SCF. The SRSM goes to state "Idle".
- (e12) *SRF\_Sanity\_Timeout*: This event occurs when the SRSM has been in user interaction state for a network-operator-defined-period of time (timer  $T_{SRF}$ ) without having a *PlayAnnouncement/PromptAndCollectUserInformation/PromptAndReceiveMessage* operation to execute. The SRSM initiates a bearer channel disconnect sequence to the SSF using the applicable bearer channel signalling system. The SRSM goes to state "Idle".
- (E14) *ScriptClose\_from\_SCF*: This event takes place when the *ScriptClose* operation from the SCF is received. The SRSM goes to state "Connected".

NOTE 2: This state transition is permitted only if the SRSM received *ScriptRun* operation from SCF previously.

- (e15) *Final\_implicit\_ScriptEvent\_to\_SCF*: This event takes place when the final Result of the User Interaction script execution is sent from the SRF to the SCF. It covers the implicit termination case when automatic disconnection is either not allowed or not specified. The SRSM moves to state "Connected".

In addition to these explicitly marked transitions, failure of a user-SRF bearer connection will cause the SRSM to transit to "Idle" from any state. These transitions are not shown on figure 3 for the purpose of visual clarity.

## 7.3 Example SRF control procedures

This clause provides a detailed description of the SRF procedures. Arrow diagrams are used for the description of the connect, interaction with the end user, and disconnect stages.

The SRF control procedures are based on various physical allocation patterns of SRF.

The service assist and hand-off procedures are also described in this clause as examples.

Note that, through this clause, bearer connection control signalling messages are used for explanatory purpose, and are not subject for standardization in the present document. The terms used for bearer connection control signalling messages only represent the functional meaning.

### 7.3.1 SRF connect procedures

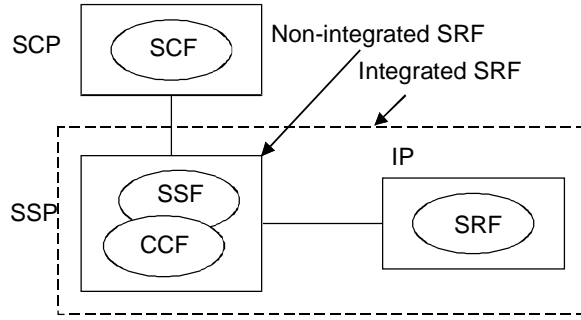
#### 7.3.1.1 SRF connect physical procedures

Several procedures are required for different physical scenarios. The cases to be covered are described below and illustrated in figure 4.

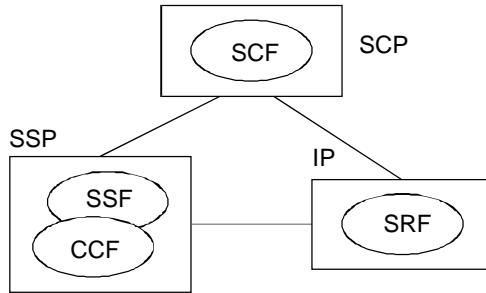
The SCF-SRF relationship could be done through a direct link as depicted in scenario ii) and iv) or may be established through a relay SSF as depicted in scenario i), iii) and v):

- i) the IP is integrated into the SSP, or attached to the SSP, possibly via a local exchange, that is interacting with the SCP but the SCP's operations to the IP are relayed via the SSP which performs any needed protocol conversion;
- ii) the IP is directly attached to the SSP that is interacting with the SCP but the SCP's operations to the IP are sent directly to the IP without SSP relaying involved;
- iii) the IP is integrated into another SSP, or directly attached to another SSP, than the one that is interacting with the SCP but the SCP's operations to the IP are relayed via the second SSP (called the "Assist" method), and on completion of the user interaction, control is returned to the first SSP;
- iv) the IP is directly attached to a node other than the SSP that is interacting with the SCP but the SCP's operations to the IP are sent directly to the IP without SSP relaying involved (called the "Assist" method, but with a variation on the physical connectivity of the entities involved), and on completion of the user interaction, control is returned to the first SSP; and
- v) the IP is attached to another SSP and on completion of the user interaction, control of the call is retained at that SSP (called the "Hand-off" approach).

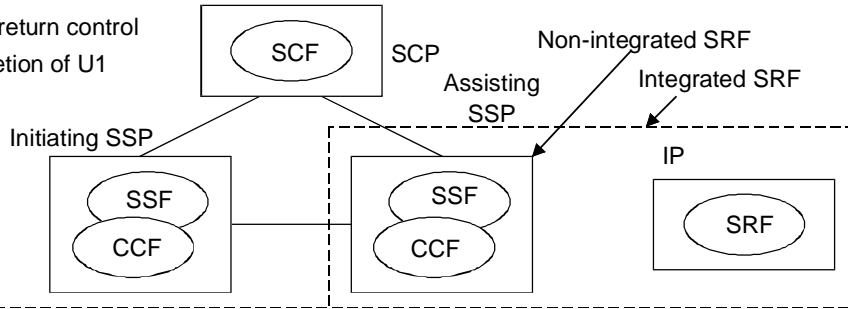
Case i) SSF relay



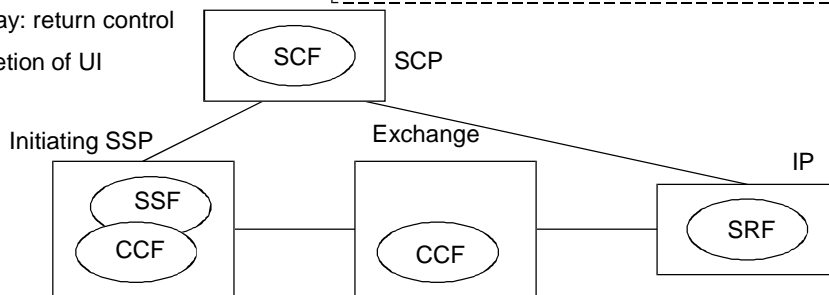
Case ii) Direct path SCP to IP



Case iii) Assist with relay: return control to initiating SSP on completion of U1



Case iv) Assist without relay: return control to initiating SSP on completion of U1



Case v) Hand-off: retain control at assisting SSP after UI completed

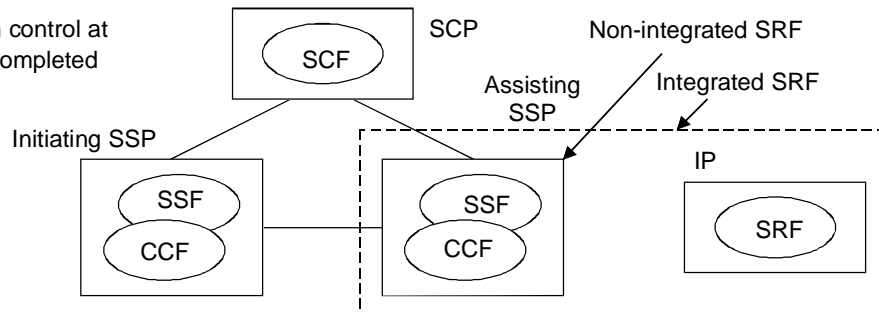


Figure 4: Physical scenarios

As within a Call Segment Association several users may be present at the same time, it shall be possible to interact with one of the user independently from the others. For each independent User Interaction, any of one of above scenario may be used.

Per Call Segment only one User Interaction is allowed; the resource can be either connected to one leg or to the connection point of the Call Segment (that means to all the legs of the Call Segment).

It is recommended that before performing the User Interaction towards a user, the SCF isolates the user (the leg) in a new CallSegment. PromptAndCollectInformation, PromptAndReceiveMessage ScriptRun, ScriptInformation and ScriptClose operations are only accepted for Call Segment where only one "joined" leg exists.

Note that this control is not applicable in case of "direct" scenario, this is left to the SCF responsibility.

In case several legs are connected within a Call Segment, it's possible to perform a User Interaction towards one of the legs of the Call Segment. However, only a PlayAnnouncement is allowed in that configuration and will be applied to the dedicated leg (user) only. It has to be noted this control could not be performed in case of "direct" scenario.

In order to be able to connect the resource to the targeted user(s), the LegId or Csid parameter is used in the ConnectToResource/EstablishTemporaryConnection. In order to be able to direct the operation to the right resource, the LegId or CSid parameters is also used in the subsequent PlayAnnouncement, PromptAndCollectUserInformation, PromptAndReceiveMessage ScriptRun,ScriptInformation, ScriptClose operation and OCCRUI operations.

NOTE: In case of direct scenario case ii) and iv), established via EstablishTemporaryConnection, as there is one independent direct SCF-SRF relationship per established User Interaction, the LegId,CSid parameters in subsequent operation are ignored by the IP.

In case of relay scenario, subsequent operation shall address the Leg/Callsegment in the same way as it was done by the ConnectToResource operation.

In case of Assist or Handed-off SSF, the legID and CSID parameters are not to be used with the PlayAnnouncement, the PromptAndCollectUserInformation and the PromptAndReceiveMessage operations but if received they are ignored.

If the PlayAnnouncement operation is received after a ConnectToResource operation only the following combinations are allowed, depending on the 'resourceAddress' parameter of the related ConnectToResource operation.

<b>ConnectToResource ('resourceAddress')</b>	<b>PlayAnnouncement ('connectedParty')</b>
'none' or 'iPRoutingAddress'	<not included>
'none' or 'iPRoutingAddress'	'legID' (note)
'none' or 'iPRoutingAddress'	'callSegmentIdentifier'
'legID' or 'iPAddressAndLegID'	'legID'
'callSegmentID' or 'iPAddressAndCallSegment'	'callSegmentIdentifier'
'callSegmentID' or 'iPAddressAndCallSegment'	'legID' (note)
NOTE: Applicable if 'informationToSend' parameter contains outband information (i.e. 'displayInformation'). It is not possible for 'inbandInfo' and 'tone'. This way individual display information can be sent to each party of a call segment without each time connecting and disconnecting the SRF.	

If the PromptAndCollectUserInformation, PromptAndReceiveMessage ScriptRun,ScriptInformation, ScriptClose operation is received after a ConnectToResource operation only the following combinations are allowed, depending on the 'resourceAddress' parameter of the related ConnectToResource operation.

<b>ConnectToResource ('resourceAddress')</b>	<b>PromptAndCollectUserInformation, PromptAndReceiveMessage ScriptRun,ScriptInformation, ScriptClose ('callSegmentIdentifier')</b>
'none' or 'iPRoutingAddress'	<not included>
'none' or 'iPRoutingAddress'	'callSegmentIdentifier'
'callSegmentID' or 'iPAddressAndCallSegment'	'callSegmentIdentifier'

If the RequestreportUTSI, SendSTUI, operation is received after a ConnectToResource operation only the following combinations are allowed, depending on the 'resourceAddress' parameter of the related ConnectToResource operation.

ConnectToResource ('resourceAddress')	RequestreportUTSI, SendSTUI ('legId')
'iPRoutingAddress'	<not included>
'legID' or 'iPAddressAndLegID'	'LegID'

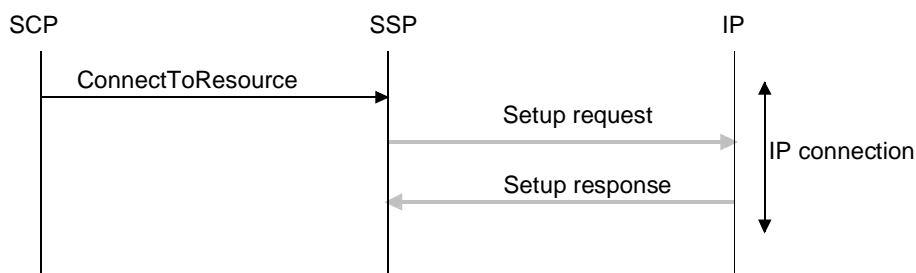
If the use of the subsequent operation LegID/CSid parameter is not consistent with the 'resourceAddress' of a related ConnectToResource operation (i.e. violates the allowed combinations) then the error "UnexpectedDataValue" is reported to the invoking entity.

Each of the scenarios will now be examined using arrow diagrams.

In the following figures up to figure 16, black lines indicate INAP operations, grey line indicate DSS 1 operations.

#### 7.3.1.1.1 SSF relay

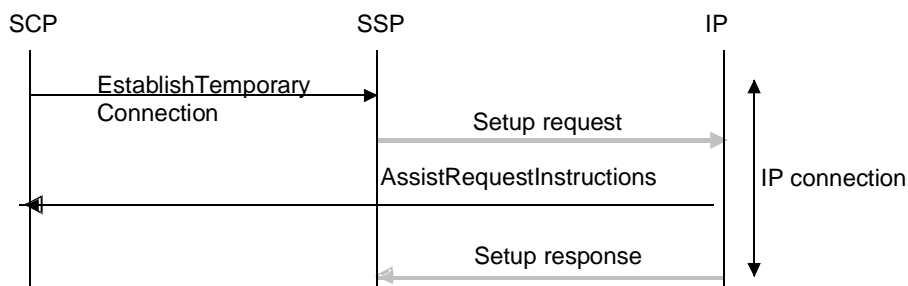
Case i) is illustrated in figure 5. Note that for the integrated IP/SSP, the internal activities of the node can still be modelled in this way, but the details of how this is achieved are left to the implementor. This approach makes it unnecessary for the SCP to distinguish between integrated and external but directly connected IPs. See also a note on the possibility of concatenating the first user interaction operation with the ConnectToResource operation discussed in the clause on user interaction below. The establishment of the SCF-SRF relationship in this case is implicit.



**Figure 5: Connection to integrated or external IP with SSP relay of IP operations**

#### 7.3.1.1.2 Direct Path SCP to IP

Case ii) requires that the IP indicates to the SCP that it is ready to receive operations (see figure 6). The establishment of the SCF-SRF relationship is explicit. Note that it is necessary to convey a correlation ID to ensure that the transaction established between the SCP and the IP can be correlated to the bearer connection setup as a result of the SCP's preceding operation to the SSP.

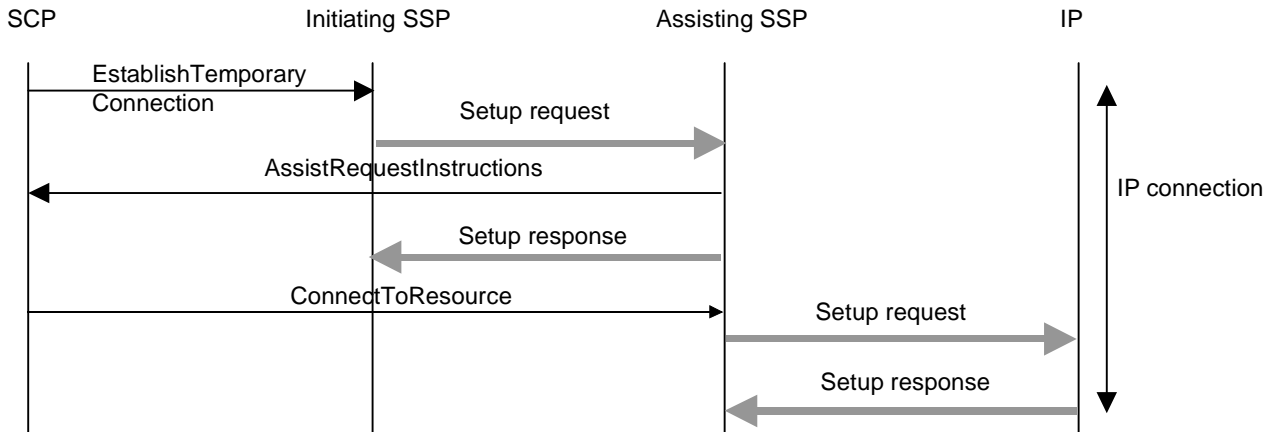


NOTE: Black lines indicate INAP operations, grey lines indicate DSS1 operations.

**Figure 6: Connection to IP with direct link to SCP, IP initiates interaction with SCP**

### 7.3.1.1.3 Assist with relay

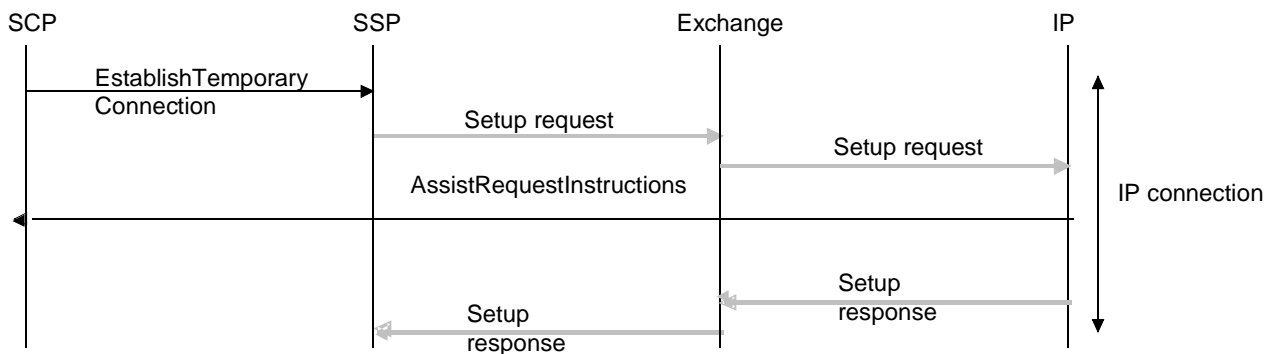
Case iii) requires that a transaction be opened with the assisting SSP so that it may relay operations from the SCP to the IP (integrated or external). Once the bearer control signalling has reached the assisting SSP, it triggers on the identity of the called facility, and initiates an interaction with the SCP that has requested the assistance. It would also be possible to trigger on other IEs such as the incoming address. The bearer control signalling must contain information to identify the SCP requesting the assistance, and a correlation ID. This information may be hidden in the address information in such a way that non-message based signalling systems may also be used to establish the bearer connection to the assisting SSP. After the AssistRequestInstructions are received by the SCP, the procedures are the same as case i). Figure 7 illustrates the preamble involved.



**Figure 7: Preamble for assist case with integrated IP or external IP and SSP relay of SCP-IP messages**

### 7.3.1.1.4 Assist without relay

Case iv) does not require the establishment of a second transaction from the assisting exchange, hence it doesn't need to be an SSP. This then becomes a preamble to the procedure shown in figure 6 as shown in figure 8.



**Figure 8: Preamble for assist case with external IP and direct SCP-IP messaging**



### 7.3.1.1.5 Hand-off

Case v) merely requires the sending of an operation to the first SSP to route the call to the handed-off SSP, and then figure 5 applies at handed-off SSP. This is shown in figure 9. Note that the activity at handed-off SSP represents a new interaction with the SCP and "AssistRequestInstructions" is used. Once the bearer control signalling has reached the assisting SSP, it triggers on the identity of the called facility, and initiates an interaction with the SCP that has requested the assistance. It would also be possible to trigger on other IEs such as the incoming address. The bearer control signalling must contain information to identify the SCP requesting the assistance, and a correlation ID. This information may be hidden in the address information in such a way that non-message based signalling systems may also be used to establish the bearer connection to the assisting SSP.

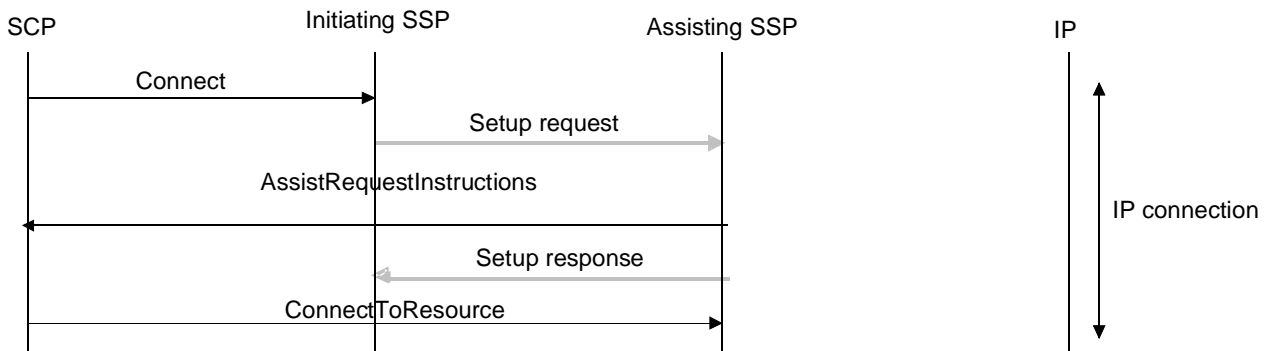


Figure 9: Preamble for hand-off case

## 7.3.2 SRF end user interaction procedures

The end user interaction procedures allow:

- the sending of one or multiple messages to the end user by using the **PlayAnnouncement** operations;
- a dialogue with the end user by using one or a sequence of **PromptAndCollectUserInformation** operations;
- a dialogue with the end user by using one or a sequence of **PromptAndReceiveMessage** operations;
- cancellation of a **PlayAnnouncement**, **PromptAndCollectUserInformation** or **PromptAndReceiveMessage** operations by using a generic cancel operation;
- the execution of script in the SRF;
- a combination of the above.

There are only two physical scenarios for user interaction:

- the SSP relays the operations from the SCP to the IP and the responses from the IP to the SCP (SSF relay case); and
- the operations from the SCP to the IP and the responses from the IP are sent directly between the SCP and the IP without involving the SSP (direct SCF-SRF case).

In case of "direct" scenario, the messaging between the SCP and the IP may be SS No. 7 TCAP based; and bearer control signalling may be any signalling system.

In case of "relay" scenarios, the operations between the SCP and the SSP may be SS No. 7 TCAP-based; the messaging between the SSP and the IP may be any signalling system (e.g. DSS1, ISUP) and the way the operation are transported depends on the signalling system capabilities. The SSP would have to do protocol conversion from the transport protocol used at the SSP-SCP interface to the network one used for the SSP-IP for the operations and responses it relays between the SCP and the IP (e.g. SS No. 7 TCAP to DSS1 facility IE).

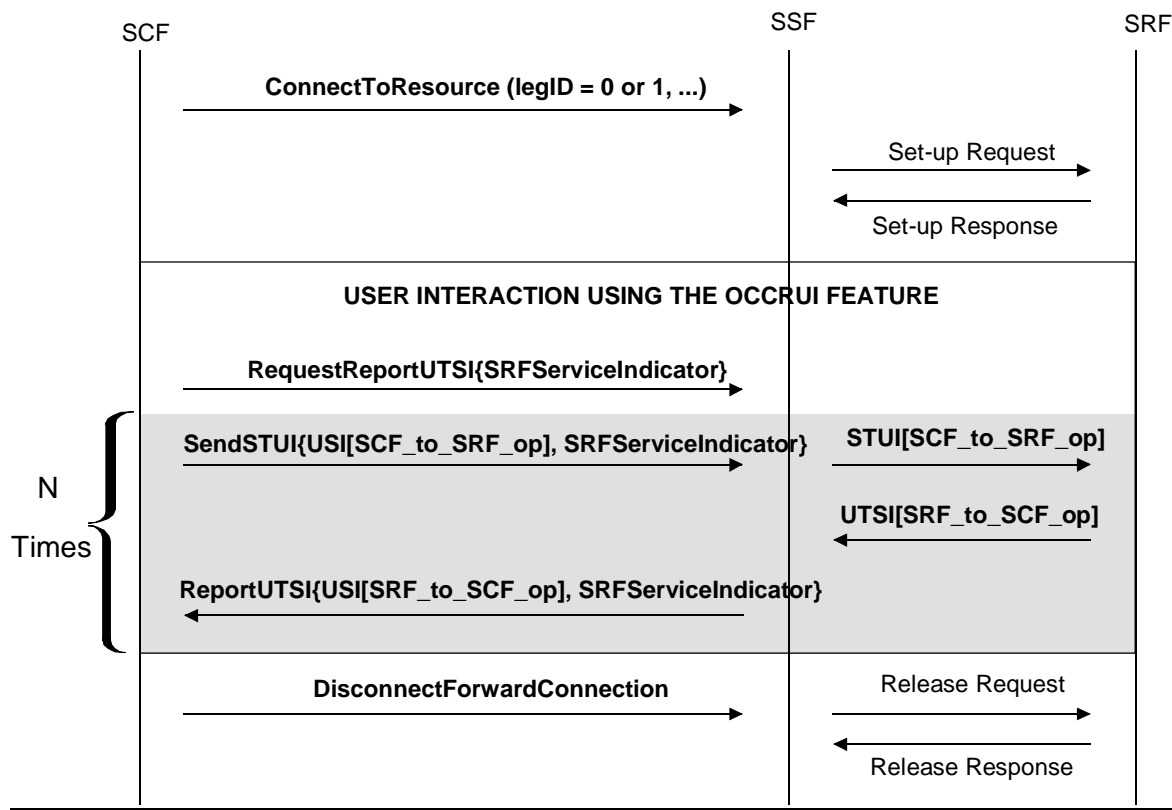
In case SS No. 7 TCAP is used, it is also necessary to consider the capability of SS No. 7 TCAP to concatenate several Invoke PDUs in one message. This capability allows to concatenate any SCF-SRF subsequent UserInteraction operation PlayAnnouncement/PromptAndCollectUserInformation/PromptAndReceiveMessage/ScriptRun/ScriptInformation/ScriptClose to be carried in one message. This has some advantages in this physical scenario, such as reduced numbers of messages, and possibly better end-user perceived performance.

In the Relay case, when the SCF uses the *ConnectToResource* operation to connect to an External SRF, as an alternative the SCF and the SRF may embed the "User Interaction" operations exchanged with each other using the "Out-Channel Call Related User Interaction" operations: *SendSTUI*, *ReportUTSI* and *RequestReportUTSI*. The signalling system should support this transport mechanism.

A network specific value of the *serviceIndicator* parameter is allocated for the "External SRF connection": *SRF\_Connection*. Once receiving the *SendSTUI* (resp. *RequestReportUTSI*) operation from the SCF with a *serviceIndicator* parameter value set to *SRF\_connection*, the SSF checks only this parameter to decide that this operation is related to the "SCF-External SRF communication". The SSF uses then the *legId* to address the right IP (e.g. in case within the CSA several User Interaction are running in parallel on legs). The same processing for *serviceIndicator* applies for the *reportUTSI* operation in the "SRF to SCF" direction.

The mapping of "User Interaction" operations on to USI mechanism is not subject to standardization. E.g. the value of the *USIInformation* could be the encoding of the ROSE component invoking the "User Interaction" operation.

The message sequence chart in figure 10 illustrates the UserInteraction in relay case using the OCCRUI operations.



NOTE: For the SSF there is no correlation between a *SendSTUI* and a *ReportUTSI*.

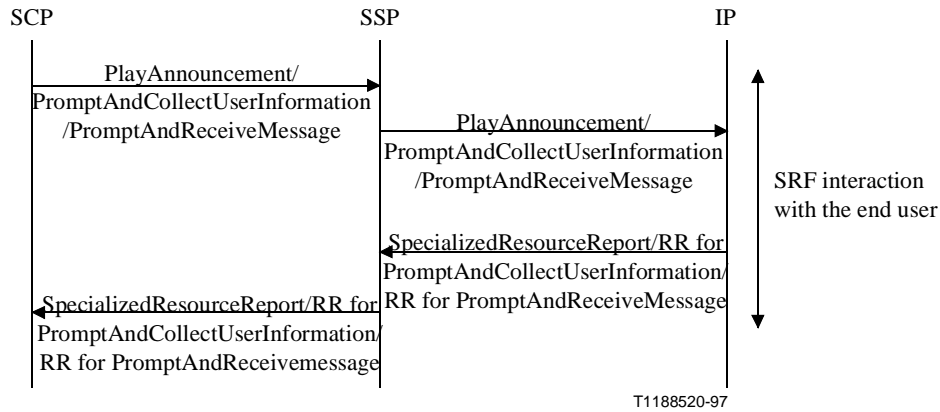
**Figure 10: User Interaction in the SRF relay case: OCCRUI mechanism**

### 7.3.2.1 (PA/P&C/P&R)

This clause illustrates User Interaction "relay" or "direct" scenario for PA/P&C/P&R.

Case i) is illustrated in figure 11.

The corresponding connect procedure is described in figure 5.

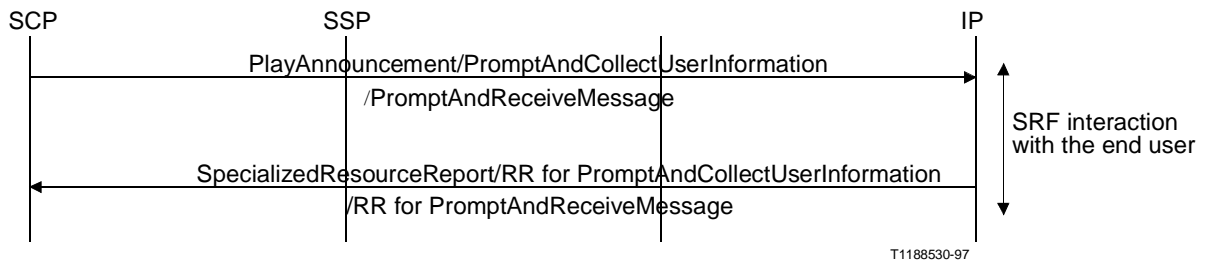


**Figure 11: SSP relay of user interaction operations and responses**

It is also necessary to consider the capability of SS No. 7 TCAP to concatenate several Invoke PDUs in one message. This capability allows, for the scenario in figure 5, the `ConnectToResource` and the first `PlayAnnouncement/PromptAndCollectUserInformation/PromptAndReceiveMessage/PromptAndReceiveMessage/ScriptRun`, to be carried in one message. This has some advantages in this physical scenario, such as reduced numbers of messages, and possibly better end-user perceived performance.

Case ii) is illustrated in figure 12.

The corresponding connect procedure is described in figure 8.



**Figure 12: Direct SCF-SRF of user interaction operations and responses**

### 7.3.2.2 User Interaction Script

Usually, the user-interaction is defined as a chain of "Questions and Answers", which result in the SCF sending several commands to the SRF in sequence. This process is optimized by making the SRF perform an indivisible block of "Questions and Answers" called a "User Interaction-script". Refer to figure 13.

User Interaction-scripts allow the grouping of the user interaction parts of the service into functional blocks which use SRF resources in the most efficient way. The transition from one user interaction part to the other is triggered by internal results (e.g. error condition) or external decision (e.g. user choice, or result from a database interrogation). The SRF then communicates the results back to the SCF. Depending on the User Interaction script to be run, the SRF may request additional information from the SCF, and receive it during the User Interaction script execution.

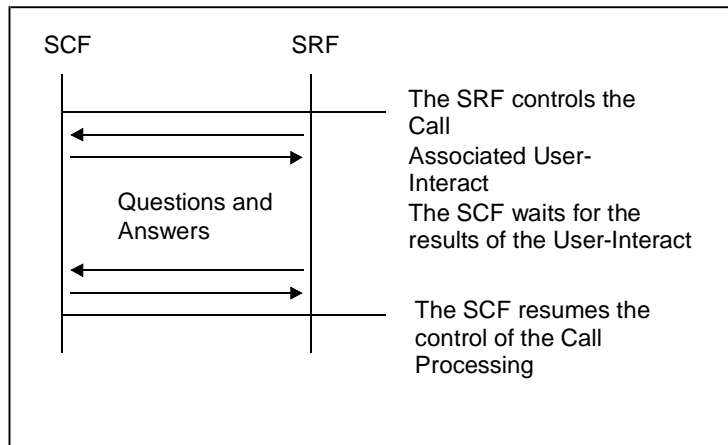


Figure 13: User Interaction –script execution

### 7.3.3 SRF disconnection procedures

The disconnection procedures are controlled by the SCF and the procedure used is selected based on the needs of the service being executed. The bearer disconnection procedure selected by the SCF is to either allow the SRF to disconnect on completion of user interaction, or to have the SCF explicitly order the SSF to disconnect.

SRF disconnect does not cause disconnection by the SSF/CCF back to the end user terminal. The SSF/CCF recognizes that a connection to an SRF is involved because the operations from the SCF for this purpose are distinct from the operations that would be used to route the call towards a destination. There is no impact on bearer signalling state machines as a result of this since incoming and outgoing bearer signalling events are not simply transferred to each other, but rather are absorbed in call processing, and regenerated as needed by call processing. Therefore, to achieve the desired functionality, call processing needs simply choose not to regenerate the disconnect in the backward direction. Figure 14 illustrates this concept.

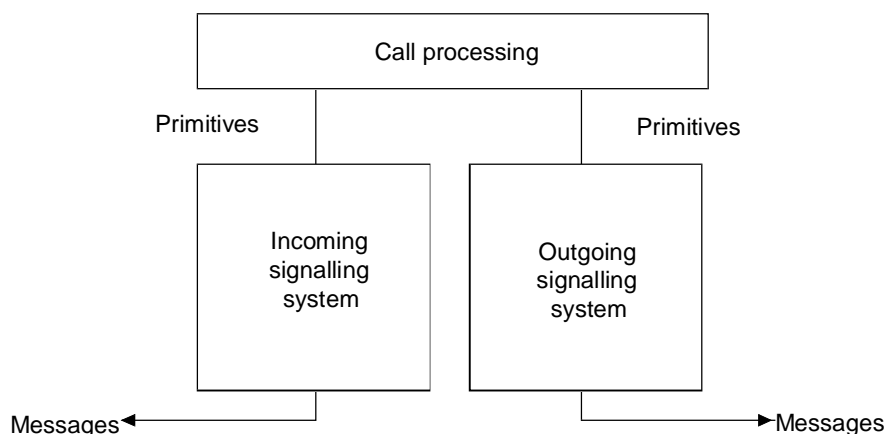


Figure 14: Relationship of incoming and outgoing signalling systems to call processing

As for the SRF connection procedures, the SRF disconnection is affected by the physical network configuration.

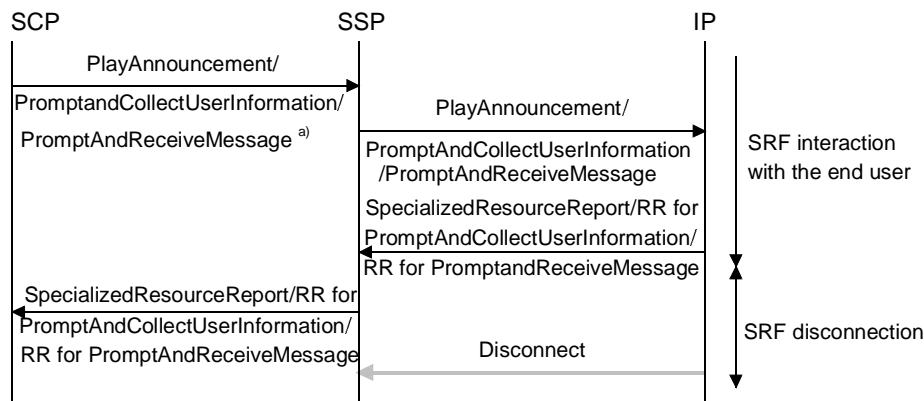
In order to simplify the interface between the SCF and the SRF, a number of assumptions are made. The assumptions, and the resulting rules, result in unambiguous procedures from both the SCF and the SRF points of view. The rules, presented below, refer to the SRF originated disconnect, or "SRF Initiated Disconnect", and to the SCF originated disconnect, or "SCF Initiated Disconnect". While other scenarios are possible, they are not included because they either duplicate the functionality presented below or they otherwise do not add value from a service perspective.

As within a CSA several User Interaction may be activated in parallel on different legs, a procedure for disconnection only applies for one User Interaction active on one leg or CS and towards one IP resource:

- 1) If a series of PlayAnnouncement/PromptAndCollectUserInformation/PromptAndReceiveMessage/ScriptRun operations are to be executed by the same SRF-FSM instance, then SRF disconnect is inhibited for all but the last and is inhibited on the last PlayAnnouncement/PromptAndCollectUserInformation/PromptAndReceiveMessage/ScriptRun provided the SRF disconnect is forbidden. When a subsequent PlayAnnouncement/PromptAndCollectUserInformation/PromptAndReceiveMessage/ScriptRun is received, it is buffered until the completion of any preceding PlayAnnouncement/PromptAndCollectUserInformation/PromptAndReceiveMessage/ScriptRun.
- 2) A generic cancel operation terminates the indicated PlayAnnouncement/PromptAndCollectUserInformation/PromptAndReceiveMessage if it is being executed by the SRF, but does not disconnect the SRF. If the cancel operation is for a buffered PlayAnnouncement/PromptAndCollectUserInformation/PromptAndReceiveMessage, that PlayAnnouncement/PromptAndCollectUserInformation/PromptAndReceiveMessage is discarded, but the current and any buffered PlayAnnouncement/PromptAndCollectUserInformation/PromptAndReceiveMessage are executed. An SRF interacts with one user (legid) or several users (CSid) and therefore cancelling a PlayAnnouncement/PromptAndCollectUserInformation/PromptAndReceiveMessage only affects the user(s) to which the SRF is connected.
- 3) The SCF must either explicitly order "Disconnect" or enable SRF initiated disconnect at the end of the PlayAnnouncement/PromptAndCollectUserInformation/PromptAndReceiveMessage/ScriptRun. An SRF left connected without an operation to execute may autonomously disconnect if it has not received any operations within a defined time limit. This could occur, for example, after an EstablishTemporaryConnection which is not followed within a reasonable time period with a PlayAnnouncement/PromptAndCollectUserInformation/PromptAndReceiveMessage/ScriptRun operation. This sanity timing value will depend on the nature of the interaction the SRF supports and should be selected by the network operator accordingly.
- 4) When SRF initiated disconnect is enabled in a PlayAnnouncement/PromptAndCollectUserInformation/PromptAndReceiveMessage/ScriptRun, then the SRF must disconnect on completion of the user interaction.
- 5) When SRF initiated disconnect is not enabled, the SCF should ask the SRF to inform it of the completion of the user interaction using the SpecializedResourceReport operation for "announcement complete", using the return result for the PromptAndCollectUserInformation operation or using the return result for the PromptAndReceiveMessage operation or the last scriptEvent. Therefore no check on consistency between those parameters is performed by the SRF.
- 6) If the user disconnects, the SRF is disconnected and the SSF releases resources and handles the transaction between the SSF and the SCF. The SRF discards any buffered operations and returns its resources to "idle". The relationship with the SCF is terminated.
- 7) When the SCF explicitly orders the SSF to disconnect by "DisconnectForwardConnection/DisconnectForwardConnectionWithArgument" operation, the SSF releases the bearer connection to the SRF. No operation reporting SRF disconnect from the SSF to the SCF is required.

### 7.3.3.1 SRF initiated disconnect

The SRF disconnect procedure is illustrated in figure 15. Bearer messages are shown in grey. The figure shows local or embedded SRF and hand-off disconnection procedures. The direct SCF-SRF case, the assist case is not shown in figure 15. The SRF disconnect is enabled by the SCF within a PlayAnnouncement/PromptAndCollectUserInformation/PromptAndReceiveMessage/ScriptRun operation. When the SRF receives a PlayAnnouncement/PromptAndCollectUserInformation/PromptAndReceiveMessage/ScriptRun enabling disconnection, it completes the dialogue as instructed by the PlayAnnouncement/PromptAndCollectUserInformation/PromptAndReceiveMessage/ScriptRun, and then initiates the SRF initiated disconnection using the applicable bearer control signalling. The SSF/CCF knows that it is an SRF disconnecting and does not continue clearing the call toward the end user. The SSF-FSM for CS exits state reflecting that a User Interaction is running and executes any buffered operations. In the hand-off case, the SSP shown in figure 15 is the "handed-off" SSP.



<sup>a)</sup> Disconnect from SRF is not forbidden.

**Figure 15: SCF disconnect for local, embedded and hand-off scenarios**

For the direct SCF-SRF case, the procedures also work in the same manner. The SRF disconnect is enabled by the SCF within a PlayAnnouncement/PromptAndCollectUserInformation/PromptAndReceiveMessage/ScriptRun operation. When the SRF receives a PlayAnnouncement/PromptAndCollectUserInformation/PromptAndReceiveMessage/ScriptRun enabling disconnection, it completes the dialogue as instructed by the PlayAnnouncement/PromptAndCollectUserInformation/PromptAndReceiveMessage/ScriptRun, and then initiates the SRF initiated disconnection using the applicable bearer control signalling. The Initiating SSF/CCF knows that it is an SRF disconnecting and does not continue clearing the call toward the end user. The Initiating SSF-FSM for CS exits state reflecting that a User Interaction is running and executes any buffered operations for that CS. Meanwhile the SCF/SRF dialogue is closed by means of a TC-END primitive with basic end (cf. clause 13.1.1).

For the assisting SSF case, the SRF initiated disconnect procedures are not used because the assisting SSF does not propagate the disconnection of the bearer connection to the Initiating SSF. The SCF initiated disconnect procedures described in the following clauses are used for the assisting SSF case.

### 7.3.3.2 SCF Initiated Disconnect

The SCF initiated disconnect procedure is illustrated in figure 16. Bearer messages are shown in gray. The figure shows only the assisting SSF case, and the direct SCF-SRF case is not shown. To initiate the SCF initiated disconnection of the SRF for a specific User Interaction, the SCF must request and receive a reply to the last PlayAnnouncement/PromptAndCollectUserInformation/PromptAndReceiveMessage/ScriptRun operation requested sent to the corresponding IP. The SpecializedResourceReport operation contains an "announcement complete", the return result for PromptAndCollectUserInformation contains "collected information", the return result for PromptAndReceiveMessage contains "recorded information", and the ScriptEvent contains the "last event".

For the local/embedded cases, the SCF initiated disconnect uses an operation called DisconnectForwardConnection/DisconnectForwardConnectionWithArgument. Once the DisconnectForwardConnection/DisconnectForwardConnectionWithArgument is received by the SSF, it will initiate a "release of bearer channel connection" between the physical entities containing the SSF and the corresponding SRF, using applicable bearer control signalling.

For the direct case, the SCF sends a DisconnectForwardConnection/DisconnectForwardConnectionWithArgument to the SSF. The SSF initiates the disconnection of the SRF using the applicable bearer control signalling. The initiating SSP, knowing that the forward connection was initiated as the result of an EstablishTemporaryConnection, does not disconnect back to the user and the SSF-FSM for CS exits state reflecting a User Interaction is running. Upon the receipt of the disconnection request, the SRF closes its dialogue toward the SCF, it may use the prearranged end or the abort procedures for that purpose (cf. clause 13.1.1).

For the assisting SSF case, the disconnection procedures are shown in figure 16. On receipt of the DisconnectForwardConnection/DisconnectForwardConnectionWithArgument from the SCP, the initiating SSF initiates the disconnection of the Assisting SSF using the applicable bearer control signalling. The initiating SSP, knowing that the forward connection was initiated as the result of an EstablishTemporaryConnection, does not disconnect back to the user and the SSF-FSM for CS exits state reflecting a User Interaction is running.

The Assisting SSF in turn disconnects the SRF using the applicable bearer control signalling. Meanwhile the Assisting SSF closes its dialogue toward the SCF, it may use either the prearranged end or the abort procedures for that purpose (cf. clause 13.1.1).

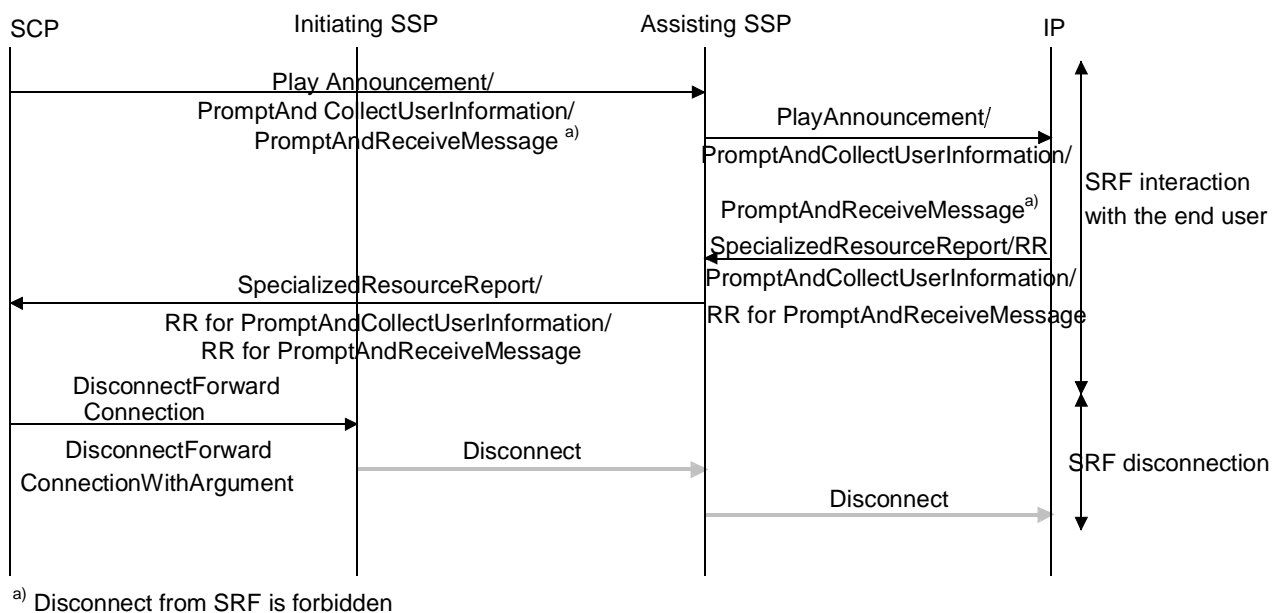


Figure 16: SCF initiated disconnect for assist scenario

### 7.3.4 Call-Unrelated Interaction

The establishment of this relationship does not need to be preceded by the establishment of a relationship between the SCF and the SSF when a call unrelated interaction is concerned. Call-unrelated interaction is based on a set of procedures to exchange information with SCF when SRF is not processing a call. These procedures are generic in that they may be used for many services or operations, but are typically used for management capabilities implementing in the logics. Some examples of call-unrelated interaction are:

- a) the SCF monitoring the availability of resources at the SRF;
- b) the SCF requesting control of some SRF resources outside the context of a call, e.g. for automatic deletion of messages stored at the SRF (e.g. based on data/time parameters rather than commands from the user).

Signalling capabilities to support such a feature are not subject to standardization.

### 7.3.5 Examples Illustrating Complete User Interaction Sequences

The following figures and their accompanying tables provide examples of complete sequences of user interaction operations covering the three stages:

- Connect the SRF and the end user (bearer connection) and establish the SCF-SRF relationship.
- Interact with the end user.
- Disconnect the SRF and the end user (bearer connection) and terminate the SCF-SRF relationship.

Examples are:

- SSP with integrated SRF, clause 7.3.5.1.
- SSP relays message between SCP and IP, clause 7.3.5.2.
- Direct SCP-IP information transfer, clause 7.3.5.3.
- SSP assist (relay SSP), clause 7.3.5.4.
- Message sequences for service assist, clause 7.3.5.5.
- Message sequence for Hand Off, clause 7.3.5.6.

#### 7.3.5.1 SSP with integrated SRF

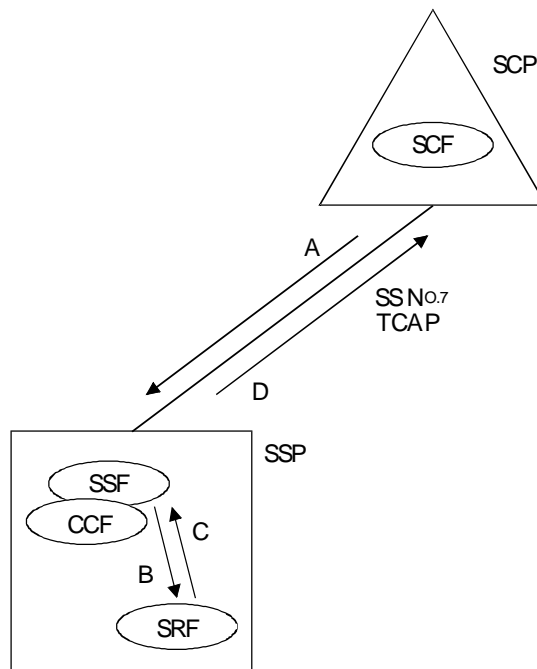


Figure 17: SSP with integrated SRF



In figure 17, the SSP with an integrated (or embedded) SRF, the procedural scenarios can be mapped as follows:

Procedure name	Operations	Protocol flows
Connect to resource and first P&P&C/P&R	ConnectToResource; PlayAnnouncement/ PromptAndCollectUserInformation /PromptAndReceiveMessage Setup; PlayAnnouncement/ PromptAndCollectUserInformation/Prompt	A B
User interaction	AndReceiveMessage PlayAnnouncement/ PromptAndCollectUserInformation/Prompt AndReceiveMessage	A then B C then D
SRF initiated disconnect	SpecializedResourceReport/RR for PromptAndCollectUserInformation/RR for PromptAndReceiveMessage	C then D
SCF initiated disconnect	SpecializedResourceReport/RR for PromptAndCollectUserInformation/RR for PromptAndReceiveMessage Disconnect	C (intra-SSP bearer control) C then D
	DisconnectForwardConnection/Disconnect ForwardConnectionWithArgument Disconnect	A B (intra-SSP bearer control)

A simple extension to this integrated case is the configuration where the SRF is located in an intelligent peripheral locally attached to the SSP. The SCP-IP operations are relayed via the SSF in the SSP. This is depicted in figure 18.

As an example, in case the User Interaction is to be established towards one user only, the ConnectToResource, PlayAnnouncement/PromptAndCollectUserInformation/Prompt AndReceiveMessage and DisconnectForwardConnectionWithArgument should all contain the LegId parameter indicating the targeted user/resource within the Call Segment Association.

7.3.5.2 SSP relays messages between SCP and IP

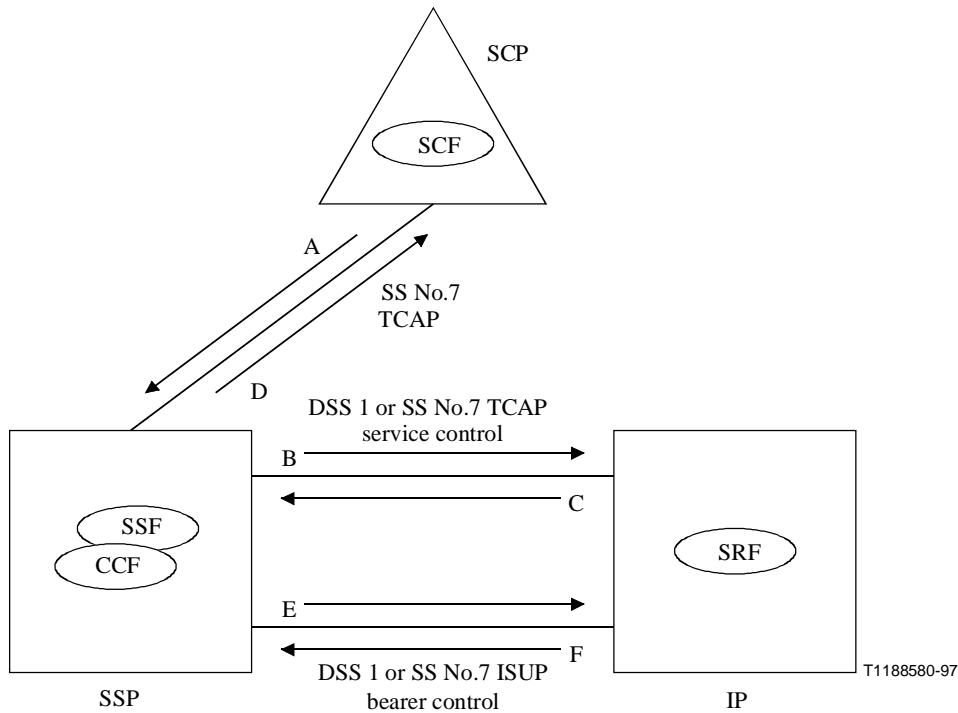


Figure 18: SSP relays messages between SCP and IP

The procedural scenarios for this relay SSF with an IP (figure 18) can be mapped as follows:

Procedure name	Operations	Protocol flows
Connect to resource and first P&P/C&P&R	ConnectToResource; PlayAnnouncement/PromptAndCollectUserInformation/PromptAndReceiveMessage <i>If DSS 1 used:</i> Setup; PlayAnnouncement/ PromptAndCollectUserInformation/PromptAndReceiveMessage <i>If SS No. 7 used:</i> IAM	A  E and B (Facility IE)  E
User interaction	PlayAnnouncement/PromptAndCollectUserInformation/PromptAndReceiveMessage PlayAnnouncement/PromptAndCollectUserInformation/PromptAndReceiveMessage SpecializedResourceReport/RR for PromptAndCollectUserInformation/RR for PromptAndReceiveMessage	B A then B C then D
SRF initiated disconnect	SpecializedResourceReport/RR for PromptAndCollectUserInformation/RR for PromptAndReceiveMessage <i>If DSS 1 used:</i> Disconnect <i>If SS No. 7 used:</i> Release	C then D C then D F
SCF initiated disconnect	SpecializedResourceReport/RR for PromptAndCollectUserInformation/RR for PromptAndReceiveMessage DisconnectForwardConnection/DFCWithArgument <i>If DSS 1 used:</i> Disconnect <i>If SS No. 7 used:</i> Release	F C then D  A E E

In some cases, the IP may have an SS No. 7 or other interface to the controlling SCP. This case is shown in figure 19. Note that the SCP must correlate two transactions to coordinate the activities.

As an example, in case the User Interaction is to be established towards one user only, the ConnectToResource, PlayAnnouncement/PromptAndCollectUserInformation/PromptAndReceiveMessage and DisconnectForwardConnectionWithArgument should all contain the LegId parameter indicating the targeted user/resource within the Call Segment Association.

## 7.3.5.3 Direct SCP-IP information transfer

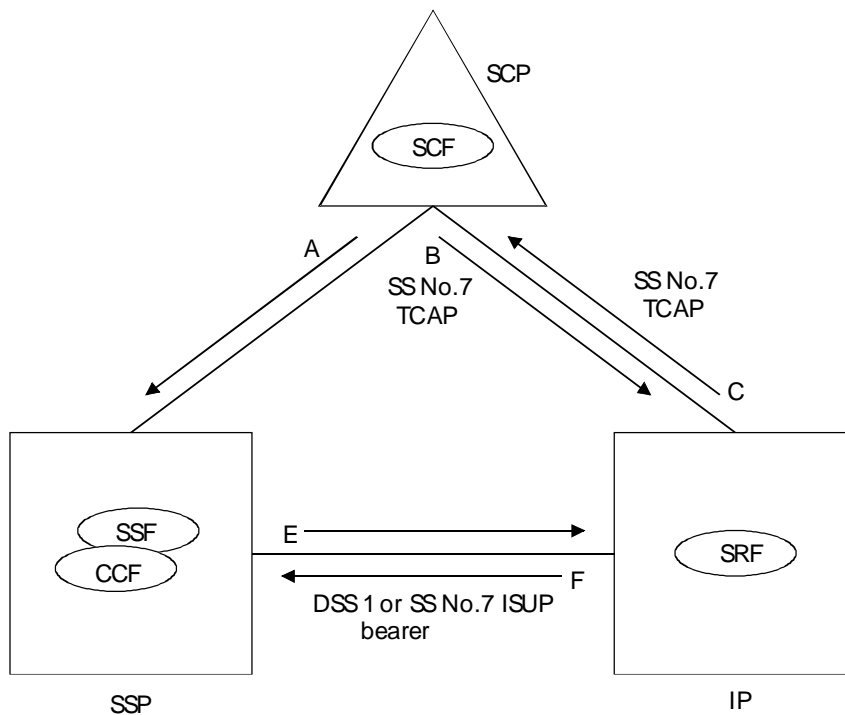


Figure 19: Direct SCP-IP information transfer

In figure 19, the procedural scenarios can be mapped as follows:

Procedure name	Operations	Protocol flows
Connect to resource	EstablishTemporaryConnection <i>If DSS 1 used:</i> Setup AssistRequestInstructions <i>If SS No. 7 used:</i> IAM AssistRequestInstructions	A E C E C
User interaction	PlayAnnouncement/ PromptAndCollectUserInfo/Prompt AndReceiveMessage SpecializedResourceReport/RR for PromptAndCollectUserInfo/RR for PromptAndReceiveMessage	B C
SRF initiated disconnect	SpecializedResourceReport/RR for PromptAndCollectUserInfo/RR for PromptAndReceiveMessage <i>If DSS 1 used:</i> Disconnect <i>If SS No. 7 used:</i> Release	C F F
SCF initiated disconnect	SpecializedResourceReport/RR for PromptAndCollectUserInfo/RR for PromptAndReceiveMessage DisconnectForwardConnection/DFCWith Argument <i>If DSS 1 used:</i> Disconnect <i>If SS No. 7 used:</i> Release	C A E E

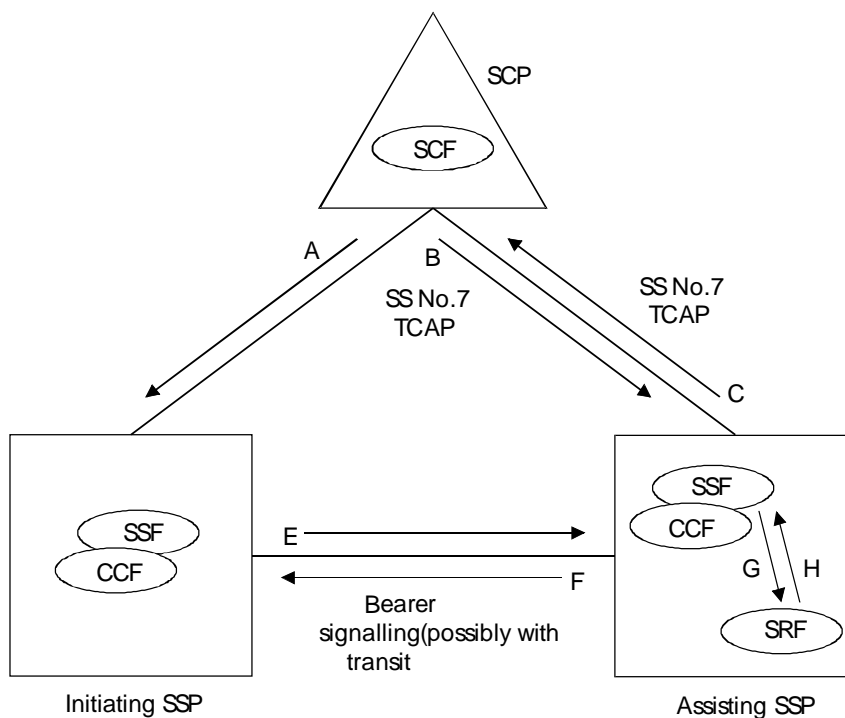
The assisting SSF scenario involves straightforward procedural extensions to the basic cases shown above. One mapping of the assisting SSF case is shown in figure 20. In this case, SRF initiated disconnect cannot be used. Other physical mappings can be derived as described in the text following the figure and its accompanying table.

Note that the integrated SRF and SSF relay case requires a transaction between the SCP and the assisting SSP (figure 20) but the SCP direct case does not since the transaction is directly between the SCP and the IP connected to the remote exchange. In the latter case, any transit exchanges, including the one the IP (SRF) is connected to, are transparent to the procedures.

Note also that the SCP must again correlate two transactions.

As an example, in case the User Interaction is to be established towards one user only, the EstablishTemporaryConnection and DisconnectForwardConnectionWithArgument should both contain the LegId parameter indicating the targeted user/resource within the Call Segment Association. For operations sent on the SCF-SRF direct link, LegId parameter is useless.

#### 7.3.5.4 SSP assist(relay SSP)



**Figure 20: SSP assist(relay SSP)**

In figure 20, the procedural scenarios can be mapped as follows:

Procedure name	Operations	Protocol flows
Assist preamble	EstablishTemporaryConnection	A
	<i>If DSS 1 used:</i>	
	Setup	E
	AssistRequestInstructions	C
	ConnectToResource	B
	Setup	G
	ResetTimer	A
	<i>If SS No. 7 used:</i>	
	IAM AssistRequestInstructions	E
	ConnectToResource	C
	Setup	B
	ResetTimer	G
		A
		B then G
User interaction	PlayAnnouncement/ PromptAndCollectUserInformation/Prompt AndReceiveMessage	
	SpecializedResourceReport/RR for PromptAndCollectUserInformation/RR for PromptAndReceiveMessage	H then C
	SpecializedResourceReport/RR for PromptAndCollectUserInformation/RR for PromptAndReceiveMessage	H then C
	DisconnectForwardConnection/DFCWith Argument	A
SCF initiated disconnect	<i>If DSS 1 used:</i>	E and G (intra-SSP bearer ctrl)
	Disconnect	
	<i>If SS No. 7 used:</i>	E and G (intra-SSP bearer ctrl)
	Release	

Note that the assisting SSP case shown in figure 20 can be generalized to cover both the case where the SRF is embedded in assisting SSP (as shown), and the case where the SRF is locally connected to assisting SSP. In this latter case, the SRF communication (protocol flows B, C, G and H) would conform to the physical scenario shown in figure 18.

As an example, in case the User Interaction is to be established towards one user only, the EstablishTemporaryConnection and DisconnectForwardConnectionWithArgument in the Initiating should both contain the LegId parameter indicating the targeted user/resource within the Call Segment Association. In the assisting since the CSA is simply reduced to one CS, ConnectToResource, PlayAnnouncement/PromptAndCollectUserInformation/PromptAndReceiveMessage might use default LegID or CSid positioning.

The service hand-off scenario can similarly be viewed as a sequence consisting of an IN service to route a call from one SSP to another, followed by any one of the previously described physical user interaction scenarios. For describing this scenario figure 20 can be used also.

### 7.3.5.5 Message sequences for service assist

The following clause provides additional details on the message sequences for the service assist procedure in figure 20:

#### 1) Protocol Flow A:

- The SCP, during the processing of a request for instruction, determines that resources remote from the initiating SSP are required and that call processing will continue from the initiating SSP after the remote resources have been used (e.g., the call will be completed to a destination address after information is collected from the calling party). An EstablishTemporaryConnection operation containing the address of the assisting SSP (for routing the call), the ScfID, the CorrelationID (both used for the assisting SSP to establish communication back to the SCP) and the legid or Csid to indicate for which user the link should be established, is sent to the initiating SSP. The EstablishTemporaryConnection is used instead of a regular Connect operation because of the nature of the connection to the assisting SSP. The initiating SSP must be aware that the SCP will ask it to continue in the processing of the call at some point in the future.

NOTE 1: The ScfID and CorrelationID may be included in the routing address of the assisting SSP.

## 2) Protocol Flow E:

- The initiating SSP routes the call to the assisting SSP. The ScfID and CorrelationID are sent to the assisting SSP. Existing in-band signalling and SS7 information elements (e.g. routing number) could be used to transport this information. The transport mechanism used to send this information between SSPs is independent of the service assist control procedures between the SCF and SSF.

## 3) Protocol Flow C:

- The assisting SSP uses an AssistRequestInstructions operation to establish communication with the SCP. The CorrelationID is sent in the AssistRequestInstructions to allow the SCP to correlate two transactions.

## 4) Protocol Flow B:

- The SCP sends instructions to the assisting SSP based on service logic control.

## 5) Protocol Flow A:

- The SCP may need to generate reset timer events to the initiating SSP so that it does not time out the call.

NOTE 2: The usage of ResetTimer operation is optional.

## 6) Protocol Flow A:

- When resource functions have been completed, a DisconnectForwardConnection/DisconnectForwardConnectionWithArgument operation is sent to the initiating SSP) with the same legid or Csid as indicated in the request. This indicates, that the temporary connection to the assisting SSP has to be disconnected.

NOTE 3: A DisconnectForwardConnection/DisconnectForwardConnectionWithArgument operation followed by a ConnectToResource may be sent to the assisting SSP to access several resources in the assisting case.

## 7) Protocol Flow E:

- The initiating SSP sends a message via bearer control signalling to the assisting SSP to close the "assist" transaction.

## 8) The call control returns to the initiating SSP.

### 7.3.5.6 Message sequences for hand-off

The following clause outlines message sequences for the hand-off procedure using the protocol flows shown in figure 20:

## 1) Protocol Flow A:

The SCP, during the processing of a request for instruction, determines that resources remote from the initiating SSP are required and that call processing need not continue from the initiating SSP after the remote resources have been used (e.g., a terminating announcement will be played). A Connect operation containing the address of the assisting SSP (for routing the call), the ScfID and the CorrelationID (both used for the assisting SSP to establish communication back to the SCP) is sent to the initiating SSP.

NOTE: The ScfID and CorrelationID may be included in the routing address of the assisting SSP.

## 2) Protocol Flow E:

The initiating SSP routes the call to the assisting SSP. The ScfID and CorrelationID are sent to the assisting SSP. Existing in-band signalling and SS7 information elements (e.g. routing number) could be used to transport this information. The transport mechanism used to send this information between SSPs is independent of the service assist control procedures between the SCF and SSF.

## 3) Protocol Flow C:

The assisting SSP uses an AssistRequestInstructions operation to establish communication with the SCP. The CorrelationID is sent in the AssistRequestInstructions to allow the SCP to correlate two transactions. The AssistRequestInstructions is used instead of a regular request instruction (InitialDP or DP-specific operation) because the SCP must associate the AssistRequestInstructions from the assisting SSP/IP with an already active dialogue the SCP has with another SSP.

## 4) Protocol Flow B:

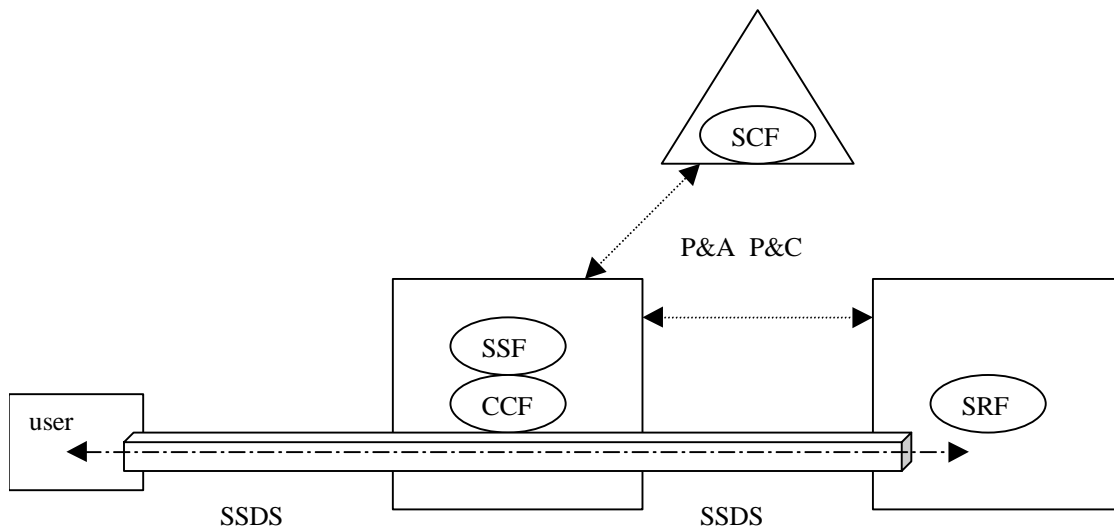
The SCP sends instructions to the assisting SSP based on service logic control.

The call control remains at the assisting SSP.

The same service assist and hand-off procedures can be reused for a direct link to an IP in this and future capability sets.

### 7.3.6 Example illustrating the use of SDSS

The SDSS protocol could be used for the dialog with the user. The figure 21 shows an example of an SSF relay scenario:



**Figure 21: SSP relays messages between SCP and IP; SDSS protocol is used towards the user**

The SRF is responsible for mapping the contents of the SDSS information parameters onto the SDSS "lower" layers protocol (e.g. conversion to analog signals, handling of checksum, acknowledgements, ...).

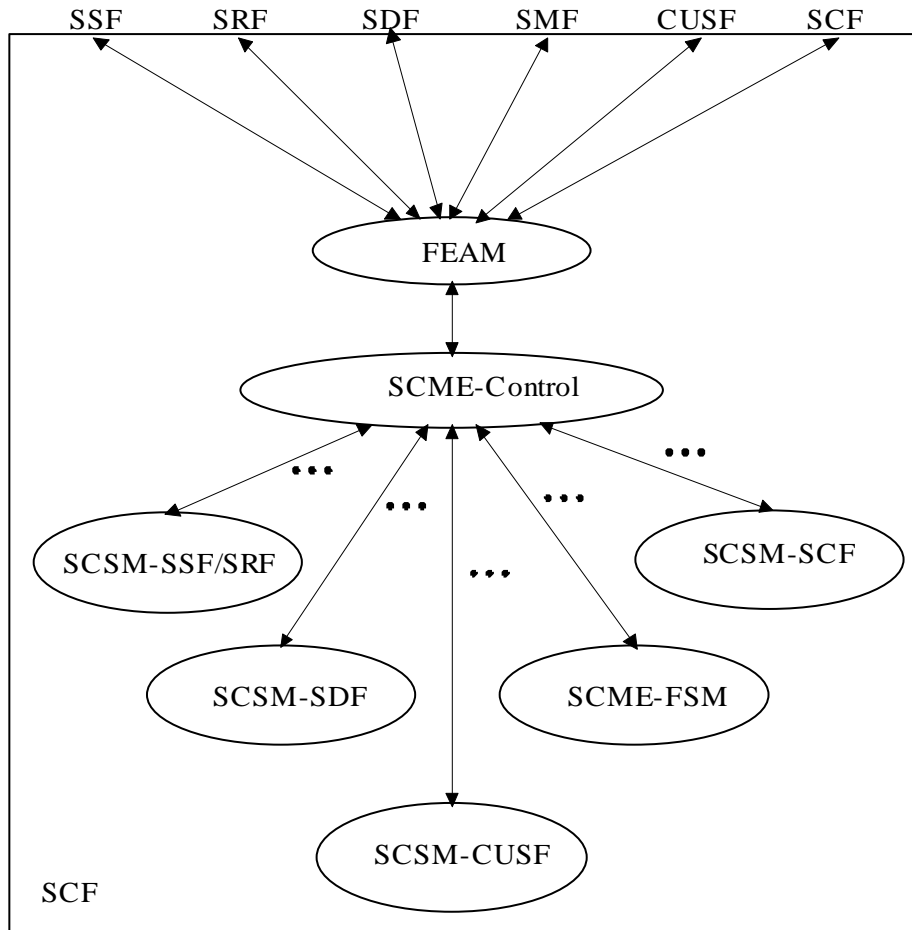
The SCF interacts with the analogue user in the following way:

- In the "SCF to User" direction, the SRF encapsulates the information (display information, menus, ...) it receives over the INAP interface onto the SDSS information over the analog interface.
- In the "User to SCF" direction, the SRF receives DTMF information from the User and relays them to the SCF onto the INAP interface.
- Before the SDSS interaction takes place, the SRF is already connected to the User; there is a "voice path" established between the User and the SCF.

## 8 SCF Application Entity procedure

The general SCF FSM structure is depicted in figure 22.

One of the FSM managed by the SCSM is the SSF/SRF FSM which handles interaction with the SSF/SRF.



**Figure 22: SCF FSM structure**

The SSF/SRF FSM is composed by smaller FSMs, amongst them, one is dedicated to Specialized resource supervision.

The SCME FSM is solicited for SRF management operation.

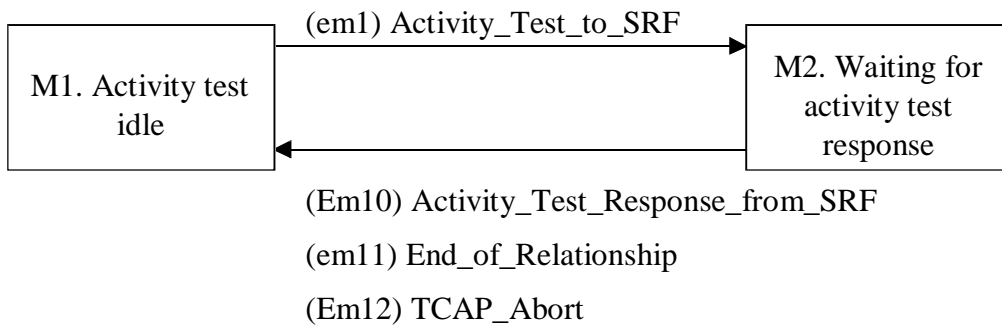
### 8.1 The SCF Management state model for SRF

The SCME handles the following operations:

- sRFCallGap;
- ActivityTest.



### 8.1.1 The Activity Test FSM



**Figure 23: Activitytest FSM**

Figure 23 provides an overview of the ActivityTest FSM for SRF.

#### 8.1.1.1 State M1: "Activity test idle"

The following event is considered in this state:

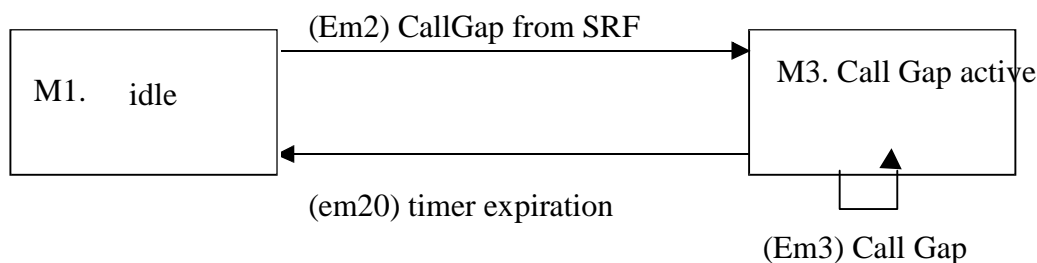
- (em1) Activity\_Test\_to\_SRF: This is an internal event, caused by the expiration of activity test timer in the SCF, and by transmission of the ActivityTest operation. This event causes a transition to state M2, "waiting for activity test response".

#### 8.1.1.2 State M2: "Waiting for activity test response"

In this state, the SCF is waiting for the activity test response from the SRF. The following events are considered in this state:

- (Em10) Activity\_Test\_Response\_from\_SRF: This is an external event, caused by reception of the response to the activity test previously issued to the SRF. This event causes a transition out of this state to state M1, "activity test idle".
- (em11) End\_of\_Relationship: This is an internal event, caused by the expiration of ActivityTest operation timer in the SCF. This event causes a transition to state M1, "activity test idle".
- (Em12) TCAP\_Abort: This is an external event, caused by reception of a P-Abort from TCAP in response to the ActivityTest operation previously issued to the SRF. This event causes a transition to state M1, "activity test idle".

### 8.1.2 The Call GAP FSM



**Figure 24: CallGap FSM**

Figure 24 provides an overview of the CallGap FSM for SRF.

The sRFCallGap operation is received inside a call context transaction.

### 8.1.2.1 State M1: "idle"

The following event is considered in this state:

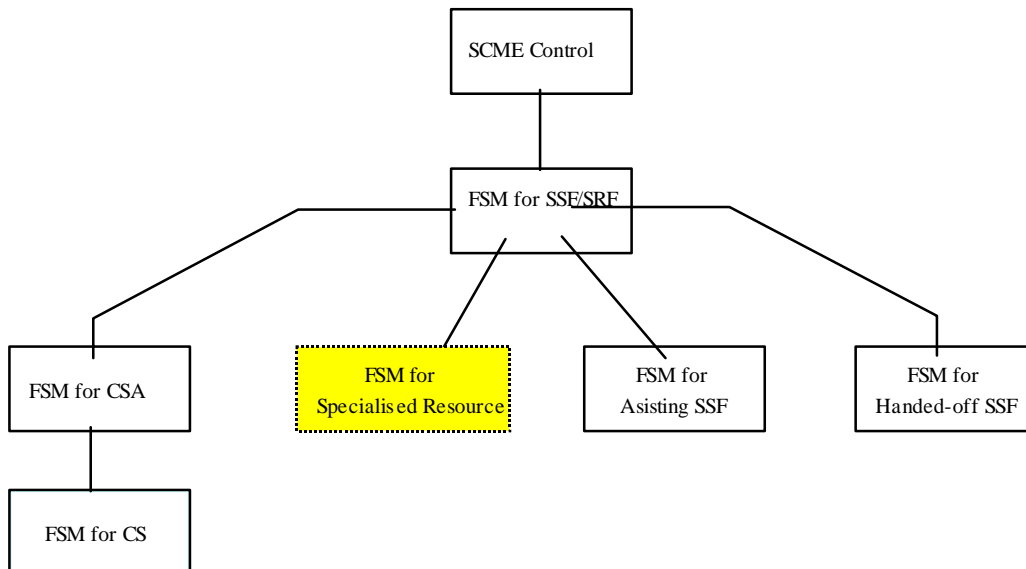
- (Em2) Call Gap: This is an external event received from the SRF, this event causes a transition to state M3, "Call Gap Active".

### 8.1.2.2 State M3: "Call GAP Active"

During this state the following events can occur:

- if call gap related duration timer expires, the SCME FSM moves to the "Idle" Management state (transition em20);
- given that call gap is active, another sRFCallGap operation having the same gapping criteria can be received from the SRF: the second "gap" replaces the first one (transition Em3) unless the duration timer value is equal to zero, in which case the SSME FSM moves to the "Idle" Management state (transition em20).

## 8.2 The SCF Call State Model (SCSM): FSM for specialized resource



NOTE: FSM for Specialised Resource is described within the present document.

**Figure 25: FSM interactions for SCSM-SSF/SRF**

Figure 25 provides an overview of the FSM interactions for SCSM-SSF/SRF.

FSM for specialized resource is created by the SSF/SRF FSM.

This FSM is used in the context of service assist where the SRF directly addresses the SCF.

SCF FSM for other SCF-SRF scenarios are described in EN 301 931-2.

The following rule is applicable to more than one state:

In every state, if there is an error in a received operation, the SLP and the maintenance functions are informed.

Figure 26 shows the general State Diagram of the FSM for Specialized Resource as relevant to the procedures concerning the SCF FSM part of the SCP/AD/SN during the processing of an IN call. Each state is discussed in one of the following clauses.

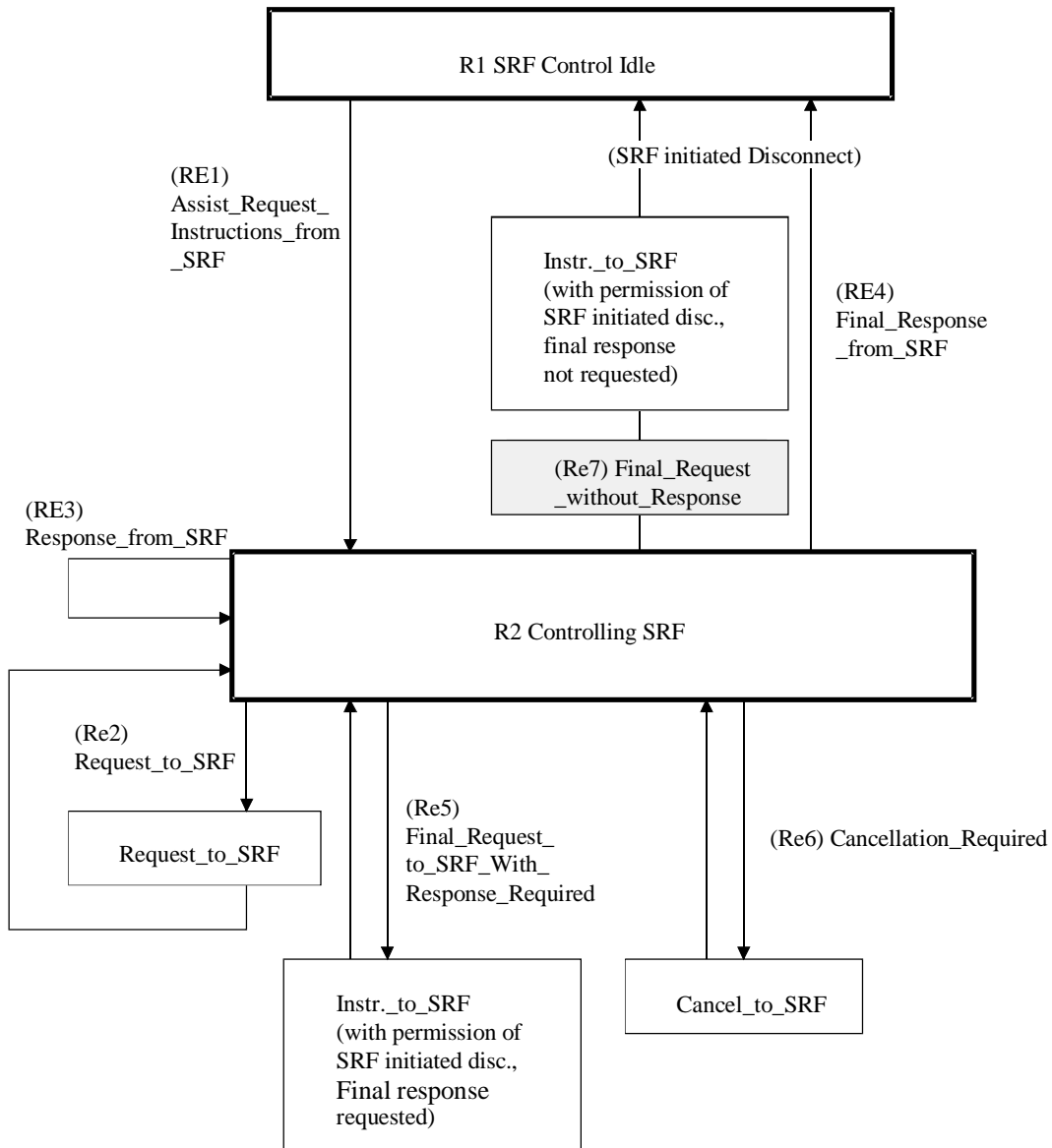


Figure 26: FSM for specialized resource

### 8.2.1 State R1: "SRF Control Idle"

The following events are considered in this state:

- **(RE1) Assist\_Request\_Instructions\_from\_SRF**: This is an external event caused by a reception of an **AssistRequestInstructions** operations from the SRF. This event causes a transition to the state R2, "Controlling SRF".

## 8.2.2 State R2: "Controlling SRF"

The following events are considered in this state:

- (Re2) Request\_to\_SRF: This is an internal event caused by the SLPI when there is a need to send one or more of the following operations without permission of SRF-initiated disconnect, to be issued to the SRF:
  - PlayAnnouncement; and
  - PromptAndCollectUserInformation;
  - ScriptRun;
  - ScriptInfo;
  - ScriptClose;
  - PromptAndReceiveMessage.

This event causes a transition back to the same state.

- (RE3) Response\_from\_SRF: This is an external event caused by a reception of one of the followings:
  - SpecializedResourceReport; and
  - Return Result for PromptAndCollectUserInformation;
  - Return Result for PromptAndReceiveMessage;
  - ScriptEvent. **This event causes a transition back to the same state.**
- (RE4) Final\_Response\_from\_SRF: This is an external event caused by a reception of one of the followings after SRF-initiated disconnect:
  - SpecializedResourceReport; and
  - Return Result for PromptAndCollectUserInformation;
  - Return Result for PromptAndReceiveMessage;
  - ScriptEvent.

This event causes a transition to the state R1, "**SRF Control Idle**".

- (Re5) Final\_Request\_to\_SRF\_with\_Response\_Required: This is an internal event caused by the SLPI when there is a need to send one of the following operations with permission of SRF-initiated disconnect to be issued to the SRF:
  - PlayAnnouncement;
  - PromptAndCollectUserInformation; and
  - PromptAndReceiveMessage.

This event causes a transition back to the same state.

- (Re6) Cancellation\_Required: This is an internal event that takes place when the SLPI cancels the previous **Play Announcement** or **PromptAndCollectUserInformation** or **PromptAndReceiveMessage** operation. In this case, the SCSSM sends the **Cancel** operation to the SRF, and transits back to the same state.
- (Re7) Final\_Request\_without\_Response: This is an internal event that takes place when the SCSSM finishes the user interaction and requests the disconnection of bearer connection between the initiating SSF and the SRF by means of SRF Initiated disconnect, while no **SpecializedResourceReport** operation is requested to be returned to the SCF in case an announcement is completed. In this case, the SCF sends the **PlayAnnouncement** (not containing a request for returning a **SpecializedResourceReport** operation as an indication of completion of the operation) and with permission of SRF-initiated disconnect to the SRF. **ScriptRun** operation is also valid at this event. This event causes a transition to the state R1, "SRF Control Idle".

---

## 9 Detailed Operation procedure

### 9.1 ActivityTest procedure

#### 9.1.1 General description

This operation is used to check for the continued existence of a relationship between the SCF and the SRF. If the relationship is still in existence, then the SRF will respond. If no reply is detected, then the SCF will assume that the SRF has failed in some way and will take the appropriate action.

#### 9.1.2 Parameters

##### 9.1.2.1 Argument Parameters

None.

#### 9.1.3 Invoking entity (SCF)

##### 9.1.3.1 Normal procedure

SCF Preconditions:

- (1) A relationship exists between the SCF and the SRF.
- (2) The activity test timer expires, TActTest, after which the ActivityTest operation is sent to the SRF.

SCF Postcondition:

- (1) If a return result "ActivityTest" is received, the SCME resets the activity test timer and takes no further action.

##### 9.1.3.2 Error handling

If a time-out is received, SCF aborts the dialogue.

The SLPI that was the user of this dialogue will be informed, the corresponding SCSM FSM will move to the state "Idle".

#### 9.1.4 Responding entity (SRF)

##### 9.1.4.1 Normal procedure

SRF Precondition:

- (1) A relationship exists between the SCF and the SRF.

SRF Postconditions:

- (1) If the dialogue ID is active and if there is a SRSM using the dialogue, the SRSM sends a return result "ActivityTest" to the SCF. If there are no other management activities, the SRSM returns to the state "Idle Management"; or
- (2) If the dialogue ID is not active, the SRSM will in that case never receive the ActivityTest operation and thus will not be able to reply.

##### 9.1.4.2 Error handling

Not applicable.

## 9.2 AssistRequestInstructions procedure

### 9.2.1 General description

This operation is sent to the SCF by an SRF in case of assisting procedure. This operation could also be sent by an SSF in case of assisting or Hand-Off procedures but these cases are described in EN 301 931-2.

### 9.2.2 Parameters

#### 9.2.2.1 Argument Parameters

The operation argument consists of the following parameters. These parameters are defined in clause 10:

- correlationID;
- iPAavailable;
- iPSSPCapabilities.

### 9.2.3 Invoking entity (SRF)

#### 9.2.3.1 Normal procedure

SRF Precondition:

- (1) An assist indication is detected by the assisting SRF.

SRF Postcondition:

- (1) The assisting SRF waits for instructions.

On receipt of an assist indication from the initiating SSF, the SRF shall assure that the required resources are available to invoke an "AssistRequestInstructions" operation in the SRF and indicate to the initiating SSF that the call is accepted. The "AssistRequestInstructions" operation is invoked by the SRF after the call, which initiated the assist indication, is accepted. The SRSM transits to state "Connected".

#### 9.2.3.2 Error handling

Generic error handling for the operation related errors are described in clause 11 and the TCAP services which are used for reporting operation errors are described in clause 13 and in EN 301 931-1.

### 9.2.4 Responding entity (SCF)

#### 9.2.4.1 Normal procedure

SCF Preconditions:

- (1) A relationship exists between the SCF and the initiating SSF in case of assist procedure.
- (2) The SCF waits for "AssistRequestInstructions".

SCF Postcondition:

- (1) An SRF instruction is being prepared.

On receipt of this operation in the FSM for SSF/SRF state "Waiting for Assist Request Instructions", the SCF has to perform the following actions.

The SCF determines SRF by means of "correlationID" or network knowledge.

If resource is available, the SCF-FSM for SSF/SRF creates a FSM for Specialized Resource and transmits the event to the new FSM. It prepares the "PlayAnnouncement" or "PromptAndCollectUserInformation" or "PromptAndReceiveMessage" or "ScriptRun" to be sent to the SRF.

#### 9.2.4.2 Error handling

Generic error handling for the operation related errors are described in clause 11 and the TCAP services which are used for reporting operation errors are described in clause 13 and in EN 301 931-1.

### 9.3 Cancel procedure

#### 9.3.1 General description

This operation could be used to request the SRF/SSF to cancel a correlated previous operation:

- PlayAnnouncement;
- PromptAndCollectUserInformation;
- PromptAnd ReceivedMessage.

The Cancel "All request" and "All request for call segment" procedures are described in EN 301 931-2 and are not applicable for SRF.

The cancellation of an operation is indicated via a respective error indication, "Cancelled", to the invoking entity of the cancelled "PlayAnnouncement" or "PromptAndCollectUserInformation" or promptAndReceivedMessage operation.

#### 9.3.2 Parameters

##### 9.3.2.1 Argument Parameters

The operation argument consists of the following parameters. These parameters are defined in clause 10.

The operation argument consists of the following alternatives:

- invokeID; or

This parameter specifies which operation invocation is to be cancelled, i.e. PromptAndCollectUserInformation, PromptAndReceiveMessage or PlayAnnouncement.

- allRequests: (not applicable for SRF); or
- callSegmentToCancel; or

This parameter specifies to which call segment the cancellation of a user interaction operation shall apply as well as the InvokeID to be cancelled.

- allRequestsForCallSegment (not applicable).

### 9.3.3 Invoking entity (SCF)

#### 9.3.3.1 Normal procedure

The SCF may either invoke this operation to the SSF or to the SRF, different conditions will prevail in each case.

SCF Preconditions:

- (1) A relationship exists between the SCF and the SSF/SRF.
- (2) An SLPI in the states has determined that a previously requested operation is to be cancelled; or
- (3) An SLPI has determined that it is no longer interested in any reports or notifications from the SSF and that the relationship should be ended.

SCF Postcondition:

- (1) The SLPI remains in the same state.

#### 9.3.3.2 Error handling

Generic error handling for the operation related errors are described in clause 11 and the TCAP services which are used for reporting operation errors are described in clause 13 and in EN 301 931-1.

### 9.3.4 Responding entity (SRF)

#### 9.3.4.1 Normal procedure

In case of Cancel(invokedID) or Cancel(callSegmentToCancel) the SRF is the responding entity.

SRF Precondition:

- (1) A PlayAnnouncement or PromptAndCollectUserInformation or PromptAndReceivedMessage operation has been received and the SRF is in the "User Interaction" state.

SRF Postcondition:

- (1) The execution of the PlayAnnouncement or PromptAndCollectUserInformation or PromptAndReceivedmessage operation has been aborted and the SRF remains in the "User Interaction" state.

#### 9.3.4.2 Error handling

Generic error handling for the operation related errors are described in clause 11 and the TCAP services which are used for reporting operation errors are described in clause 13 and in EN 301 931-1.

### 9.3.5 Responding entity (SSF)

In case of Cancel(for all requests) the SSF only is the responding entity. Refer to EN 301 931-2.

## 9.4 ConnectToResource procedure

The operation procedure is described in EN 301 931-2.

## 9.5 DisconnectForwardConnection procedure

The operation procedure is described in EN 301 931-2.



## 9.6 DisconnectForwardConnectionWithArgument procedure

The operation procedure is described in EN 301 931-2.

## 9.7 EstablishTemporaryConnection procedure

The operation procedure is described in EN 301 931-2.

## 9.8 PlayAnnouncement procedure

### 9.8.1 General description

This operation is used for in-band interaction with an analogue user or for interaction with an ISDN user. When used to apply user treatment to the indicated leg, the user treatment can be audible (e.g. inband tone) or visual (e.g. text display).

### 9.8.2 Parameters

#### 9.8.2.1 Argument Parameters

The operation argument consists of the following parameters. These parameters are defined in clause 10.

- informationToSend:

This parameter indicates which type of information to be sent. It comprises the following alternative parameters:

- inbandInfo; or
  - tone; or
  - displayInformation; or
  - sDSSInformation.
- disconnectFromIPForbidden;
  - requestAnnouncementComplete;
  - connectedParty.

This parameter indicates to the SSF which party the announcement shall be played. Ignored for Assist/Handed-off SSF. It comprises the following alternative parameters:

- legID; or
- callSegmentID.

In case this parameter is omitted, operation is directed to initial CS or the Assist/Handed-off SSF.

## 9.8.3 Invoking entity (SCF)

### 9.8.3.1 Normal procedure

SCF Preconditions:

- (1) The SLPI detects that information should be sent to the user.
- (2) A connection between the user and a SRF has been established.

SCF Postconditions:

- (1) If "RequestAnnouncementComplete" was set TRUE, the SCSM will stay in the same and wait for the "SpecializedResourceReport".
- (2) If "RequestAnnouncementComplete" was set FALSE and more information needs to be sent ("DisconnectFromIPForbidden" was set to TRUE), the SCSM will stay in the same state.
- (3) If "RequestAnnouncementComplete" was set FALSE and no more information needs to be sent ("DisconnectFromIPForbidden" was set to FALSE), the SCSM will move to the state "Preparing SSF Instructions".

### 9.8.3.2 Error handling

Generic error handling for the operation related errors are described in clause 11 and the TCAP services which are used for reporting operation errors are described in clause 13 and in EN 301 931-1.

## 9.8.4 Responding entity (SRF)

### 9.8.4.1 Normal procedure

SRF Precondition:

- (1) The SRSM-FSM is in the state "Connected"; or in the state "User Interaction" if the SRF received previously an operation from the SCF.

SRF Postconditions:

- (1) The SRF sends the information to the user as indicated by "informationToSend".
- (2) The SRSM-FSM moves to the state "User Interaction"; or remains in the same state.
- (3) If all information has been sent and "RequestAnnouncementComplete" was set TRUE, the SRSM sends a "SpecializedResourceReport" operation to the SCF.
- (4) If all information has been sent and "disconnectFromIPForbidden" was set FALSE, the SRSM disconnects the SRF from the user and the FSM moves to Idle.

The announcement send to the end-user is ended in the following conditions:

- if neither "duration" nor "numberOfRepetitions" is specified, then the network specific announcement ending conditions shall apply; or
- if "numberOfRepetitions" is specified, when all repetitions have been sent; or
- if duration is specified, when the duration has expired. The announcement is repeated until this condition is met; or
- if "duration" and "numberOfRepetitions" is specified, when one of both conditions is satisfied (whatever comes first).

### 9.8.4.2 Error handling

If a Cancel operation is received before or during the processing of the operation then the operation is immediately cancelled and the error "Cancelled" is reported to the invoking entity.

Generic error handling for the operation related errors are described in clause 11 and the TCAP services which are used for reporting operation errors are described in clause 13 and in EN 301 931-1.

## 9.9 PromptAndCollectUserInformation procedure

### 9.9.1 General description

This operation is used to interact with a call party in order to collect information.

### 9.9.2 Parameters

#### 9.9.2.1 Argument Parameters

The operation argument consists of the following parameters. These parameters are defined in clause 10.

- collectedInfo:

This parameter indicates which type of information to be collected. It comprises the following alternative parameters:

- collectedIDigits; or
- IA5Informartion; or
- detectModem.
- disconnectFromIPForbidden;
- informationToSend:

This parameter indicates which type of information to be sent. It comprises the following alternative parameters:

- inbandInfo; or
- tone; or
- displayInformation; or
- sDSSInformation.
- callSegmentID:

This parameter indicates in which CS the user (leg) is. Note that the resource is connected to the only one "joined" leg. Ignored for Assist/Handed-off SSF.

In case this parameter is omitted, operation is directed to initial CS or the Assist/Handed-off SSF.

### 9.9.2.2 Result Parameters

The operation result consists of the following parameter. This parameter is defined in clause 10.

This parameter indicates the information that have been collected. It can be one of the following field:

- digitsResponse:

This alternative indicates that digits have been collected. The field contains the following sub-field:

- digits; or
- iA5Response; or

This alternative indicates that IA5 characters have been collected. The field contains the IA5 string.

- modemdetected.

This alternative indicates that a modem has been detected.

## 9.9.3 Invoking entity (SCF)

### 9.9.3.1 Normal procedure

SCF Preconditions:

- (1) The SLPI detects that information should be collected from the end-user.
- (2) A connection between the end-user and a SRF has been established.
- (3) Only one call party is connected to the call segment to which the user interaction is to be applied.

SCF Postconditions:

- (1) The collected information is received from the SRF as response to the "PromptAndCollectUserInformation" operation.
- (2) If the "disconnectFromIPForbidden" was set to FALSE, the SCSM FSM will move to the state "Preparing SSF Instructions".
- (3) Otherwise the SCSM FSM remains in the same state.

The SLPI may continue execution before the response is received from the "PromptAndCollectUserInformation" operation, more than one operation may be sent to the SRF before the response is received. The "disconnectFromIPForbidden" parameter may only be set to FALSE if the "PromptAndCollectUserInformation" operation is the last operation sent to the SRF.

### 9.9.3.2 Error handling

Generic error handling for the operation related errors are described in clause 11 and the TCAP services which are used for reporting operation errors are described in clause 13 and in EN 301 931-1.

## 9.9.4 Responding entity (SRF)

### 9.9.4.1 Normal procedure

SRF Precondition:

- (1) The SRSF-FSM is in the state "Connected", or in state "User Interaction" if the SRF received previously an operation from the SCF.

SRF Postconditions:

- (1) The SRF has sent the information to the end-user as indicated by "informationToSend".
- (2) The collected information from the end-user is sent to the SCF as RETURN RESULT of the "PromptAndCollectUserInformation".
- (3) If the "disconnectFromIPForbidden" was set to FALSE, the SRF initiates a bearer channel disconnect to the SSF and the SRSF FSM moves to the state "Idle".
- (4) Otherwise the SRSF FSM moves to the state "User Interaction"; or remains in the same state.

The announcement send to the end-user is ended in the following conditions:

- if neither "duration" nor "numberOfRepetitions" is specified, then the network specific announcement ending conditions shall apply; or
- if "numberOfRepetitions" is specified, when all repetitions have been sent; or
- if duration is specified, when the duration has expired. The announcement is repeated until this condition is met; or
- if "duration" and "numberOfRepetitions" is specified, when one of both conditions is satisfied (whatever comes first).

The above conditions are overruled if the parameter "interruptableAnnInd" is not set to FALSE and the end-user has responded with a digit during the sending of the announcement. In this case the announcement is ended immediately. The above procedures apply only to in-band information and tones send to the end-user, for "displayInformation" the end conditions are met upon sending, i.e. no interruption can occur.

The parameter "errorTreatment" specifies how the SRF shall treat the error. The default value "reportErrorToSCF" means that the error shall be reported to SCF by means of Return Error with "ImproperCallerResponse". The value "help" indicates that no error shall be reported to SCF but assistance shall be given to the end-user in form of a network dependent default announcement (which may dependent on the context, i.e. the send message). The value "repeatPrompt" indicates that no error shall be reported to the SCF but the prompt shall be repeated to the end-user. The last two procedures shall only be done once per "PromptAndCollectUserInformation" operation.

#### **Note on processing "endOfInput"**

The receipt of any "endOfInput" condition (e.g. endOfReplyDigit, cancelDigit, firstDigitTimeout, interDigitTimeout) terminates immediately the ongoing input. In other words when e.g. an endOfReplyDigit is received, the receipt of a subsequent cancelDigit will not be processed anymore.

#### **Note on processing "modemDetection"**

- Only modem detection case:
  - When the operation is used only to detect the modem, then the choice "detectModem" has to be used in collectedInfo parameter.
  - If a modem is detected, the operation is completed and a result with an indication "modem detected" is sent to the SCF.
  - If a modem is not detected, a return Error with "ImproperCalledResponse" is reported to the SCF.

- Modem detection or digit collection case:

In this case, the choice "collectedDigits" has to be used in collectedInfo parameter and the parameter "detectModem" of "collectedDigits" parameter shall be set to "TRUE". The SRF activates simultaneously the modem detection and the digit collection process.

If a modem is detected, the operation is completed and a result with an indication "modem detected" is sent to the SCF.

If the first digit timer expires and the digit collection processing ends as an error before the modem detection expires, the operation is not completed until the expiration of the modem detection timer. In this case, after the digit collection processing was erroneous, the modem detection processing is same as the case "only modem detection" described above.

If the modem detection timer expires before the first digit is input, the operation is not completed until the end of the digit collection processing. In this case, the digit collection processing continues as the case where only digit collection was required.

### 9.9.4.2 Error handling

If a Cancel operation is received before or during the processing of the operation then the operation is immediately cancelled and the error "Cancelled" is reported to the invoking entity.

Generic error handling for the operation related errors are described in clause 11 and the TCAP services which are used for reporting operation errors are described in clause 13 and in EN 301 931-1.

If any of the parameter restrictions are violated (e.g. minimumNbOfDigits > maximumNbOfDigits) then an operation error has occurred.

## 9.10 PromptAndReceiveMessage procedure

### 9.10.1 General description

This operation is used to interact with a call party in order to record information.

### 9.10.2 Parameters

#### 9.10.2.1 Argument Parameters

The operation argument consists of the following parameters. These parameters are defined in clause 10.

- disconnectFromIPForbidden;
- informationToSend:

This parameter indicates which type of information to be sent. It comprises the following alternative parameters:

- inbandInfo; or
  - tone; or
  - displayInformation; or
  - sDSSInformation.
- subscriberID:

This parameter identifies the subscriber for whom information shall be recorded;

- mailBoxID;
- informationToRecord;
- media;
- callSegmentID.

This parameter indicates in which CS the user (leg) is. Note that the resource is connected to the only one "joined" leg. Ignored for Assist/Handed-off SSF.

In case this parameter is omitted, operation is directed to initial CS or the Assist/Handed-off SSF.

### 9.10.2.2 Result Parameters

The operation result consists of the following parameters. These parameters are defined in clause 10.

- receivedStatus;
- recordedMessageID;
- recordedMessageUnits.

## 9.10.3 Invoking entity (SCF)

### 9.10.3.1 Normal procedure

SCF Preconditions:

- (1) The SLPI detects that information should be recorded from the end-user.
- (2) A connection between the end-user and a SRF has been established.
- (3) Only one call party is connected to the call segment to which the user interaction is to be applied.

SCF Postconditions:

- (1) The identifier of the recorded message is received from the SRF as response to the "PromptAndReceiveMessage" operation.
- (2) If the "disconnectFromIPForbidden" was set to FALSE, the SCSM FSM will move to the state "Preparing SSF Instructions".
- (3) Otherwise the SCSM FSM remains in the same state.

The SLPI may continue execution before the response is received from the "PromptAndReceiveMessage" operation, more than one operation may be sent to the SRF before the response is received. The "disconnectFromIPForbidden" parameter may only be set to FALSE if the "PromptAndReceiveMessage" operation is the last operation sent to the SRF.

### 9.10.3.2 Error handling

Generic error handling for the operation related errors are described in clause 11 and the TCAP services which are used for reporting operation errors are described in clause 13 and in EN 301 931-1.

## 9.10.4 Responding entity (SRF)

### 9.10.4.1 Normal procedure

SRF Precondition:

- (1) The SRSM-FSM is in the state "Connected", or in state "User Interaction" if the SRF received previously an operation from the SCF.

SRF Postconditions:

- (1) The SRF has sent the information to the end-user as indicated by "informationToSend".
- (2) After having recorded the message from the end-user, the identifier assigned to the recorded message is sent to the SCF as RETURN RESULT of the "PromptAndReceiveMessage".
- (3) If the "disconnectFromIPForbidden" was set to FALSE, the SRF initiates a bearer channel disconnect to the SSF and the SRSM FSM moves to the state "Idle".
- (4) Otherwise the SRSM FSM moves to the state "User Interaction", or remains in the same state.

The announcement send to the end-user is ended in the following conditions:

- if neither "duration" nor "numberOfRepetitions" is specified, then the network specific announcement ending conditions shall apply; or
- if "numberOfRepetitions" is specified, when all repetitions have been sent; or
- if duration is specified, when the duration has expired. The announcement is repeated until this condition is met; or
- if "duration" and "numberOfRepetitions" is specified, when one of both conditions is satisfied (whatever comes first).

The above procedures apply only to inband information and tones send to the end-user, for "displayInformation" the end conditions are met upon sending, i.e. no interruption can occur.

#### **Note on processing "endOfInput"**

The receipt of any "endOfInput" condition (e.g. endOfRecordingDigit timeToRecord terminates immediately the ongoing input. In other words when e.g. an endOfReplyDigit is received, the receipt of a subsequent cancelDigit will not be processed anymore.

### 9.10.4.2 Error handling

If a Cancel operation is received before or during the processing of the operation then the operation is immediately cancelled and the error "Cancelled" is reported to the invoking entity.

Generic error handling for the operation related errors are described in clause 11 and the TCAP services which are used for reporting operation errors are described in clause 13 and in EN 301 931-1.

## 9.11 ReportUTSI procedure

The operation procedure is described in EN 301 931-2.

## 9.12 RequestReportUTSI procedure

The operation procedure is described in EN 301 931-2.



## 9.13 SendSTUI procedure

The operation procedure is described in EN 301 931-2.

## 9.14 ScriptClose procedure

### 9.14.1 General description

This operation is used to de-allocate the resources used to perform the instance of the "User Interaction" script: the context is released.

### 9.14.2 Parameters

#### 9.14.2.1 Argument Parameters

The operation argument consists of the following parameters. These parameters are defined in clause 10.

- uScriptId;
- uScriptSpecificInfo;
- callSegmentID.

In case the CSA contains more than one CS, this parameter indicates in which CS the user (leg) is.

In case this parameter is omitted, operation is directed to initial CS.

### 9.14.3 Invoking entity (SCF)

#### 9.14.3.1 Normal procedure

SCF Preconditions:

- (1) The SLPI has decided that it is necessary to close an instance of a given User Interaction script.
- (2) A connection between the user and the SRF has been established.

SCF Postconditions:

- (1) SLPI execution continues.
- (2) The SCSM FSM remains in the same state.

#### 9.14.3.2 Error handling

Generic error handling for the operation related errors are described in clause 11 and the TCAP services which are used for reporting operation errors are described in clause 13 and in EN 301 931-1.

## 9.14.4 Responding entity (SRF)

### 9.14.4.1 Normal procedure

SRF Preconditions:

- (1) The SRF can receive order from the SCF.
- (2) The SRF FSM is in the state "User Interaction". A User Interaction script is executing.
- (3) The SRF FSM is in the state "Connected". Last event of a script execution has been sent to the SCF and the SRF is not authorized to disconnect.

SRF Postconditions:

- (1) The SRF FSM is in the state "Connected".
- (2) The User Interaction script has been terminated and resources used to perform the script execution have been released.
- (3) Tsrp is started.

### 9.14.4.2 Error handling

Errors specific to the "SCF to SRF relationship based on the User Interaction script concept" should be added to the current list of errors.

Generic error handling for the operation related errors are described in clause 11 and the TCAP services which are used for reporting operation errors are described in clause 13 and in EN 301 931-1.

## 9.15 ScriptEvent procedure

### 9.15.1 General Description

This operation is used to return information to the SCF on the results of the execution of the instance of User Interaction script (yes/no/cancel, identifier+PIN, dialled number, etc.).

### 9.15.2 Parameters

#### 9.15.2.1 Argument Parameters

The operation argument consists of the following parameters. These parameters are defined in clause 10.

- uIScriptId;
- uIScriptResult;
- callSegmentID.

In case the CSA contains more than one CS, this parameter indicates in which CS the user (leg) is.

In case this parameter is omitted, operation is directed to initial CS.

- lastEventIndicator.

## 9.15.3 Invoking entity (SRF)

### 9.15.3.1 Normal procedure

SRF Preconditions:

- (1) The SRF can send information to the SCF.
- (2) The SRF FSM is in the state "User Interaction". A User Interaction script has been or is being executed. A condition is reached to send an intermediate or, if execution is completed, the final result.

SRF Postcondition (intermediate result):

- (1) The SRF FSM remains in the same state.

SRF Postconditions (final result, implicit termination):

- (1) Possible data about automatic disconnection of the bearer channel is checked.
- (2) If no such data was present, the SRF FSM transits back to state "Connected". Tsrfs is started.
- (3) If such data was present and indicates "disconnection not allowed", the SRF FSM transits back to state "Connected". Tsrfs is started.
- (4) If such data was present and indicates "disconnection allowed", the SRF initiates bearer channel disconnection. The SRF FSM transits to state "idle".

SRF Postconditions (final result, explicit termination):

- (1) The SRF FSM remains in state "User Interaction".
- (2) Tsrfs is started.
- (3) A ScriptClose operation is awaited.

### 9.15.3.2 Error handling

Generic error handling for the operation related errors are described in clause 11 and the TCAP services which are used for reporting operation errors are described in clause 13 and in EN 301 931-1.

## 9.15.4 Responding entity (SCF)

### 9.15.4.1 Normal procedure

SCF Preconditions:

- (1) The SCF can receive information from the SRF.
- (2) A connection between the user and the SRF has been established.

SCF Postconditions (intermediate result):

- (1) SLPI execution continues.
- (2) The SCSM FSM remains in the same state.
- (3) Result data has been passed to the SLPI (which may e.g. decide to send more information via ScriptInformation).

SCF Postconditions (final result, no other instruction to send):

- (1) A ScriptClose may have been sent to explicitly terminate the script dialogue.
- (2) Possible data about automatic SRF disconnection of the bearer channel is checked.
- (3) If no such data was present, the SCSM FSM remains in the same state. SLPI execution continues.
- (4) If such data was present and indicates "disconnection not allowed", the SCSM FSM remains in the same state.
- (5) If such data was present and indicates "disconnection allowed", the SCSM FSM transits back to state C2 Preparing CS Instructions, R1 SRF Control Idle, A1 Assisting SSF Idle or H2 Preparing SSF Instructions.

SCF Postconditions (final result, other instruction to send):

- (1) A ScriptClose may have been sent to explicitly terminate the script dialogue.
- (2) The SCSM FSM remains in the same state, SLPI execution continues.

### 9.15.4.2 Error handling

Errors specific to the "SCF to SRF relationship based on the User Interaction script concept" should be added to the current list of errors.

Generic error handling for the operation related errors are described in clause 11 and the TCAP services which are used for reporting operation errors are described in clause 13 and in EN 301 931-1.

## 9.16 ScriptInformation procedure

### 9.16.1 General description

This operation is used to send to the SRF additional information during the User Interaction script execution.

### 9.16.2 Parameters

#### 9.16.2.1 Argument Parameters

The operation argument consists of the following parameters. These parameters are defined in clause 10.

- uUIScriptId;
- uIScriptSpecificInfo;
- callSegmentID.

In case the CSA contains more than one CS, this parameter indicates in which CS the user (leg) is.

In case this parameter is omitted, operation is directed to initial CS.

## 9.16.3 Invoking entity (SCF)

### 9.16.3.1 Normal procedure

SCF Preconditions:

- (1) The SLPI has decided that it is necessary to send to the SRF additional information during the User Interaction script execution.
- (2) A connection between the user and the SRF has been established.

SCF Postconditions:

- (1) SLPI execution continues.
- (2) The SCSM remains in the same state.
- (3) Further results are awaited.

### 9.16.3.2 Error handling

Generic error handling for the operation related errors are described in clause 11 and the TCAP services which are used for reporting operation errors are described in clause 13 and in EN 301 931-1.

## 9.16.4 Responding entity (controlling SRF)

### 9.16.4.1 Normal procedure

SRF Preconditions:

- (1) The SRF can receive order from the SCF.
- (2) The SRF FSM is in the state "User Interaction".

SRF Postcondition:

- (1) The SRF FSM remains in the same state.

### 9.16.4.2 Error Handling

Generic error handling for the operation related errors are described in clause 11 and the TCAP services which are used for reporting operation errors are described in clause 13 and in EN 301 931-1.

## 9.17 ScriptRun procedure

### 9.17.1 General description

This operation is used to allocate the resources necessary to perform the instance of the "User Interaction" script (a context is partially defined for it) if necessary, then to activate this "User Interaction" script instance.

## 9.17.2 Parameters

### 9.17.2.1 Argument Parameters

The operation argument consists of the following parameters. These parameters are defined in clause 10.

- uIScriptId;
- uIScriptSpecificInfo;
- callSegmentID.

In case the CSA contains more than one CS, this parameter indicates in which CS the user (leg) is.

In case this parameter is omitted, operation is directed to initial CS.

- disconnectFromIPForbidden.

## 9.17.3 Invoking entity (SCF)

### 9.17.3.1 Normal procedure

SCF Preconditions:

- (1) The SLPI has decided that it is necessary to run an instance of a given User Interaction script.
- (2) A connection between the user and a SRF has been established.

SCF Postconditions:

- (1) SLPI execution continues.
- (2) The SCSM FSM remains in the same state.
- (3) A result is awaited.

### 9.17.3.2 Error handling

Generic error handling for the operation related errors are described in clause 11 and the TCAP services which are used for reporting operation errors are described in clause 13 and in EN 301 931-1.

## 9.17.4 Responding entity (SRF)

### 9.17.4.1 Normal procedure

SRF Preconditions:

- (1) A relationship between the SCF and SRF has been established.
- (2) The SRF FSM is in the state "Connected", or in the state "UserInteraction" if no ScriptRun operation has not been previously sent.

SRF Postconditions (state Connected):

- (1) The SRF FSM is in the state "User Interaction".
- (2) Tsrfs is stopped.
- (3) The User Interaction script is executing.

SRF Postconditions (state User Interaction):

- (1) The SRF FSM is in the state "User Interaction".
- (2) The ScriptRun operation is buffered.

#### 9.17.4.2 Error handling

Generic error handling for the operation related errors are described in clause 11 and the TCAP services which are used for reporting operation errors are described in clause 13 and in EN 301 931-1.

Errors specific to the "SCF to SRF relationship based on the User Interaction script concept" should be added to the current list of errors.

## 9.18 SpecializedResourceReport procedure

### 9.18.1 General description

This operation is used as the response to a "PlayAnnouncement" operation when the announcement completed indication is set.

### 9.18.2 Parameters

#### 9.18.2.1 Argument Parameters

None.

### 9.18.3 Invoking entity (SRF)

#### 9.18.3.1 Normal procedure

SRF Preconditions:

- (1) The SRSM FSM is in the state "User Interaction".
- (2) A "PlayAnnouncement" operation is being executed for which the parameter "RequestAnnouncementComplete" was set TRUE.
- (3) All information has been sent to the user.

SRF Postconditions:

- (1) The SRSM FSM remains in the same state.
- (2) If the "DisconnectFromIPForbidden" parameter was set FALSE, the SRSM initiates a bearer channel disconnect sequence to the SSF using the applicable bearer channel signalling system after sending the "SpecializedResourceReport" operation to the SCF. The SRSM FSM moves to the state "Idle".

#### 9.18.3.2 Error handling

Operation related error handling is not applicable, due to class 4 operation.

## 9.18.4 Responding entity (SCF)

### 9.18.4.1 Normal procedure

SCF Precondition:

- (1) A connection between the user and a SRF has been established.

SCF Postconditions:

- (1) The SCSM FSM remains in the same state.
- (2) If the "SpecializedResourceReport" relates to a "PlayAnnouncement" operation with permission of SRF initiated disconnection, the SCSM FSM moves to the state "Preparing SSF Instructions".

### 9.18.4.2 Error handling

Operation related error handling is not applicable, due to class 4 operation.

## 9.19 SRFCallGap procedure

### 9.19.1 General description

This operation is used to request the SCF to reduce the rate at which specific User Interaction requests are sent to the SRF.

### 9.19.2 Parameters

#### 9.19.2.1 Argument Parameters

The operation argument consists of the following parameters. These parameters are defined in clause 10.

- sRFgapCriteria.

This parameter indicates the Gap criteria to be applied. It comprises the following alternative parameters:

- iPAddressValue; or
- gapOnResource; or
- iPAddressAndresource;
- gapIndicators;
- controlType.

In case of automatic overload detection, the SRF uses the "destination overloaded" value.



## 9.19.3 Invoking entity (SRF)

### 9.19.3.1 Normal procedure

SRF Preconditions:

- (1) The SRF detects an overload condition and decides that call gapping has to be initiated at the SCF; or
- (2) The SRF receives a manually initiated call gapping request from the SMF.

SRF Postcondition:

- (1) The SRFSM remains in the same state upon issuing the sRFCallGap operation.

A congestion detection and control algorithm monitors the load of IP resources. After detection of a congestion situation the parameters for the sRFCallGap operation are provided.

If the congestion level changes new sRFCallGap operations may be sent for active gap criteria but with new gap interval. If no congestion is detected gapping may be removed.

A manual initiated call gap will take prevail over an automatic initiated call gap.

In the IN architecture, the SRF is not allowed to initiate a call with an SCF. As a consequence, the SRME functions are "call related"; the SRF should receive an IP connection attempt via the SSF (acknowledged or not) before indicating to the SCF its load.

### 9.19.3.2 Error handling

Operation related error handling is not applicable, due to class 4 operation.

## 9.19.4 Responding entity (SCF)

### 9.19.4.1 Normal procedure

SCF Preconditions:

- (1) Call gapping for gapCriteria is not active; or
- (2) Call gapping for gapCriteria is active.

SCF Postconditions:

- (1) Call gapping for gapCriteria is activated; or
- (2) Call gapping for gapCriteria is renewed; or
- (3) Call gapping for gapCriteria is removed.

If there is no already existing SCME FSM for the gap criteria provided then a new SCME FSM is created. This SCME FSM initializes call gapping for the specified IP connections. The parameters "gapIndicators" and "controlType" for the indicated gap criteria will be set as provided by the sRFCallGap operation.

In general, the manuallyInitiated call gapping will prevail over automatically initiated ("destination overloaded"). More specifically, the following rules will be applied in the SCF to manage the priority of different control Types associated with the same "gapCriteria".

If an SCME FSM already exists for the "gapCriteria" provided, then:

- 1) if the (new) "controlType" equals an existing "controlType", then the new parameters (i.e., "gapIndicators") will overwrite the existing parameter values;
- 2) if the (new) "controlType" is different from the existing "controlType", then the new parameters (i.e., "controlType" and "gapIndicators") will be appended to the appropriate SCME FSM (in addition to the existing parameters). The SCME FSM remains in the same state upon receiving the "sRFCallGap operation".

If the SCF detects that a User Interaction is necessary, it will check if call gapping was initiated either for the "resourceIndicator" (e.g. idScript, opcode-PA, opcode-PC,...) either for the "iPAddressValue" assigned to this User Interaction. If not, a ConnectToResource (or EstablishTemporaryConnection) operation can be sent.

In case call gapping was initiated for "iPAddressAndResource" and the "resourceIndicator" (e.g. idScript, opcode-PA, opcode-PC,...) matches, a check on the "iPAddressValue" for active call gapping is performed. If not, a ConnectToResource (or EstablishTemporaryConnection) operation can be sent.

If an IP connection to a controlled number matches only one "gapCriteria", then the corresponding control is applied. If both "manuallyInitiated" and "destination overloaded" controls are active, then only the manually initiated control will be applied.

#### 9.19.4.2 Error handling

Operation related error handling is not applicable, due to class 4 operation.

---

## 10 Parameters

This clause defines the parameters used in the operations procedures as specified in clause 9.

### 10.1 CallSegmentID

This parameter is defined in EN 301 931-2.

### 10.2 CallSegmentToCancel

This parameter indicates the Call Segment to which the SRF is connected.

### 10.3 CollectedInfo

This parameter indicates the characteristic of the information to be collected. It comprises the following alternative sub-parameters:

- **CollectedDigits**

This parameter provides guidance on how information shall be collected. It comprises the following fields:

- minimumNbOfDigits:
  - If this parameter is missing, the default value is defined to be 1. The "minimumNbOfDigits" specifies the minimum number of valid digits to be collected.
- maximumNbOfDigits:
  - This parameter should always be present and specifies the maximum number of valid digits to be collected. The following applies: "maximumNbOfDigits" >= "minimumNbOfDigits".
- endOfReplyDigit:

This parameter indicates the digit used to signal the end of input.

In case the "maximumNbOfDigits" = "minimumNbOfDigits", the "endOfReplyDigit" (could be present but) has no further meaning. This parameter can be one or two digits.

- In case the "maximumNbOfDigits" > "minimumNbOfDigits" the following applies:
  - If "endOfReplyDigit" is not present, the end of input is indicated:
    - When the inter-digit timer expires; or
    - When the number of valid digits received equals the "maximumNbOfDigits".
  - If "endOfReplyDigit" is present, the end of input is indicated:
    - When the inter-digit timer expires; or
    - When the end of reply digit is received; or
    - When the number of valid digits received equals the "maximumNbOfDigits".
    - When the end of input is attained, the collected digits are send from SRF to the SCF, including the "endOfReplyDigit" if received by the SRF.
    - In the case the number of valid digits received is less than the "minimumNbOfDigits" when the inter-digit timer expires or when the end of reply digit is received, the input is specified as being erroneous.
- cancelDigit:
  - If this parameter is present, the cancel digit can be entered by the user to request a possible retry. All digits already received by the SRF are discarded and the same "PromptAndCollectInformation" procedure is performed again, thus e.g. the same announcement to request user information is given to the user and information is collected. This parameter can be one or two digits.
  - If this parameter is not present, the user is not able to request a possible retry.
- startDigit:
  - If this parameter is present, the start digit indicates the start of the valid digits to be collected. The digits that are received by the SRF before this start digit is received, are discarded and are not considered to be valid. This parameter can be one or two digits.
  - If this parameter is not present, all received digits are considered to be valid.
- firstDigitTimeout:
  - If this parameter is present, the first digit should be received by the SRF before the first-digit timer expiration. In case the first digit is not received before first-digit timer expiration, the input is regarded to be erroneous. After receipt of the first valid or non-valid input digit, the corresponding first-digit timer is stopped.
  - If this parameter is not present, then the SRF uses a default value (network operator specific) for the first-digit timer in which the first valid or non-valid input digit is received.
  - If "startDigit" is present, the first-digit timer is stopped after the start digit is received.
- interDigitTimeOut:
  - If this parameter is present any subsequent valid or non-valid digit, should be received by the SRF before the inter-digit timer expires. As result the inter-digit timer is reset and restarted.
  - In case a subsequent valid or non-valid digit is not received before the inter-digit timer expires and the number of received valid digits is less than the "minimumNbOfDigits", the input is regarded to be unsuccessful.
  - In case a subsequent valid or non-valid digit is not received before the inter-digit timer expires and the number of received valid digits is greater than the "minimumNbOfDigits", and less than or equal to the "maximumNbOfDigits", the input is regarded to be successful.
  - If the "interDigitTimeOut" is not present, then the SRF uses a default value for the inter-digit time.

- **errortreatment:**
  - This optional parameter defines what specific action should be taken by the SRF in the event of error conditions occurring. The default value is reportErrorToSCF.
- **interruptableAnnInd:**
  - This parameter is optional, where the default value is specified being TRUE.
  - If this parameter is TRUE, the announcement is interrupted after the first valid or non-valid digit is received by the SRF. If the announcement is interrupted, a possible start-digit timer will not apply anymore. However, if the announcement has not been interrupted, a possible start-digit timer is started after the announcement has been finished.
  - If this parameter is present and explicitly set to FALSE, the announcement will not be interrupted after the first digit is received by the SRF. The received digits during the announcement are discarded and considered to be non-valid. All other specified parameters ("minimumNbOfDigits", "maximumNbOfDigits", "endOfReplyDigit", etc.) do not apply before the announcement has been finished. The possible start-digit timer is started after the announcement has been finished.
- **voiceInformation:**
  - This parameter is optional, where the default value is specified being FALSE. If the "voiceInformation" parameter is FALSE, all valid or non-valid digits are entered by DTMF.
  - If this parameter is present and explicitly set to TRUE, calling user is required to provide all valid or non-valid information by speech. The SRF will perform voice recognition and translation of the provided information into digits. A possible end of reply digit will also have to be provided by speech.
- **voiceBack:**
  - This parameter is optional, where the default value is specified being FALSE. If the "voiceBack" parameter is FALSE, no voice back information is given by the SRF. If this parameter is present and explicitly set to TRUE, the valid input digits received by the SRF will be announced back to the calling user immediately after the end of input is received. The non-valid input digits will not be announced back to the calling user.
  - A possible end of reply digit is not voiced back.
- **detectmodem:**
  - This parameter indicates that the SRF activates the Modem detection process in parallel to the digit collection process; or
- **IA5Information**

This parameter indicates that IA5 information is expected; or

- **detectModem**

This parameter indicates that the SRF activates the Modem detection process without any digit collection.

## 10.4 ControlType

This parameter is defined in EN 301 931-2.

## 10.5 Digits

This parameter is defined in EN 301 931-2.

## 10.6 CorrelationID

This parameter is defined in EN 301 931-2.

## 10.7 DisconnectFromIPForbidden

This parameter indicates whether or not the SRF should be disconnected from the user when the operation is completed.

## 10.8 GapIndicators

This parameter is defined in EN 301 931-2.

## 10.9 InformationToRecord

This parameter provides guidance on how information shall be recorded. It comprises the following sub-parameters:

- messageID:

This parameter indicates the identifier that shall be assigned to the recorded message. This option is used when the recording is not intended for a mailbox belonging to a specific subscriber, but a temporary recording, e.g. within the context of 1 call.

- messageDeletionTimeOut:

This parameter indicates the maximum time duration a message recording shall be stored in the SRF.

- timeToRecord:

This parameter indicates the maximum allowed time for the recording.

- controlDigits:

This sub-parameter indicates information related to the record process. It comprises the following fields:

- endOfRecordingDigit:
  - If this parameter is present, the end of recording digit can be entered by the user in order to mark the end of the recording. This parameter can be one or two digits.
- cancelDigit:
  - If this parameter is present, the cancel digit can be entered by the user in order to cancel the ongoing recording. Any information received so far shall be erased. This parameter can be one or two digits.
- replayDigit:
  - If this parameter is present, the replay digit can be entered by the user to have the recorded message replayed. This parameter can be one or two digits.
- restartRecordingDigit:
  - If this parameter is present, the restart digit can be entered by the user to request a possible retry. All information already received by the SRF is erased and the same "PromptAndReceiveMessage" procedure is performed again, thus e.g. the same announcement to request user information is given to the user and information is collected. This parameter can be one or two digits.
- restartAllowed:
  - This parameter indicates whether or not a possible restart recording is allowed.

- replayAllowed:
  - This parameter indicates whether or not a possible replay of the recording is allowed.

## 10.10 InformationToSend

This parameter indicates an announcement, a tone or display information to be sent to the end user by the SRF. It comprises the following alternative sub-parameters:

- **inbandInfo:**

This parameter specifies the inband information to be sent, It comprises the following fields:

- messageID:

This parameter indicates the message(s) to be sent, this can be one of the following fields:

- elementaryMessageID:

This parameter indicates a single announcement.

- text:

This parameter indicates a text to be sent. The text shall be transformed to inband information (speech) by the SRF. This parameter consist of two subparameters, messageContent and attributes. The attributes of text may consist of items such as language.

- elementaryMessageIDs:

This parameter specifies a sequence of announcements.

- variableMessage:

This specifies an announcement with one or more variable parts.

- numberOfRepetitions:

This parameter indicates the maximum number of times the message shall be sent to the end-user.

- duration:

This parameter indicates the maximum time duration in seconds that the message shall be played/repeated. ZERO indicates endless repetition.

- interval:

This parameter indicates the time interval in seconds between repetitions, i.e. the time between the end of the announcement and the start of the next repetition. This parameter can only be used when the number of repetitions is > 1.

- preferred Language:

This parameter indicates the language that should preferably be used for the UserInteractions. If the preferred language is not available, the default language shall be used; or

- **tone:**

This parameter specifies a tone to be sent to the end-user. It comprises the following fields:

- toneID:

This parameter indicates the tone to be sent.

- duration:

This parameter indicates the time duration in seconds of the tone to be sent. ZERO indicates infinite duration; or

- **displayInformation:**

This parameter indicates a text string to be sent to the end-user. This information can not be received by a PSTN end-user; or

NOTE: As the current signalling systems (DSS 1/ISUP) don't provide an indication whether or not information can be displayed by the user's terminal, in case of user interaction with an ISDN user two consecutive "PlayAnnouncement" operations are sent. The first contains the display information, the second contains the inband information to be sent to the user. Since the execution of the display information by the SRF should take a limited amount of time, the inband information will be immediately sent by the SRF to the user, in sequence with the display information.

- **sDSSInformation:**

This parameter contains the "application" layer SDSS protocol. It is used for the SCF to interact with an analogue user by mean of the SDSS (ServerDisplay and Script Service) protocol on the analogue interface (V23).

## 10.11 LastEventIndicator

This parameter means that the ScriptEvent operation contains the final result of the script execution. If the event is the final one for the UI Script, "TRUE" must be set.

## 10.12 LegID

This parameter is defined in EN 301 931-2.

## 10.13 Media

This parameter indicates the type of media for the recording e.g. a voice mail, a faxgroup.

## 10.14 MailBoxID

This parameter identifies the mailbox for which information shall be recorded, in case the subscriber owns multiple mailboxes.

## 10.15 ReceivedStatus

This parameter indicates the result of the recording. Three values are distinguished:

- MessageComplete - the recording was successfully completed. End of recording may be indicated, e.g. by an end of recording digit, voice activity monitoring or by onhook.
- MessageInterrupted - the user has abandoned the recording, e.g. by going onhook or by pushing the cancel digit.
- MessageTimeOut - the maximum recording time has been exceeded.

## 10.16 RecordedMessageID

This parameter reports to the SCF the identifier assigned to the recorded message. It is only used, if the MessageID was not assigned by the SCF. This option can be used when the recording is intended for a mailbox belonging to a specific subscriber.

## 10.17 RecordedMessageUnits

This parameter indicates the amount of resources occupied by the recorded message. This can be expressed in units of time, memory usage, etc.

## 10.18 RequestAnnouncementComplete

This parameter indicates whether or not a "SpecializedResourceReport" shall be sent to the SCF when all information has been sent.

## 10.19 SRFgapCriteria

This parameter identifies the criteria for an IP connection to be subject to call gapping. It comprises the following alternative sub-parameter:

- `iPAddressValue`:

This parameter indicates that call gapping will be applied when the leading digits of the dialled number of an IP connection attempt match those specified in "gapCriteria"; or

- `gapOnResource`:

This parameter indicates that call gapping will be applied when the "resourceIndicator" (e.g. `idScript`, `opcode-PA`, `opcode-PC`,...) of an IP connection attempt match those specified in "gapCriteria"; or

- `iPAddressAndResource`:

This parameter indicates that call gapping will be applied when the "resourceIndicator" and the leading digits of the dialled number of an IP connection attempt match those specified in "gapCriteria".

## 10.20 SubscriberID

This parameter identifies the user identity.

## 10.21 UIScriptId

This parameter is used to address the User Interaction script.

## 10.22 UIScriptResultInfo

This parameter is used to give to the SCF the result of the User Interaction Script.

## 10.23 UIScriptSpecificInfo

This parameter is used to give to the SRF information dependent on the User Interaction script invoked.

---

# 11 Error procedures

This clause defines the error procedures for the SRF-SCF interface. Error descriptions are provided in EN 301 931-1 and the following clauses provide operation related error procedures and when relevant, error procedures related to error conditions which are not directly related to the failure of an operation.

## 11.1 Operation related error procedures

The following clauses define the generic error handling for the operation related error procedures on the SRF-SCF interface. The errors are defined as operation errors in the ASN.1 operations related description. The TC services which are used for reporting operations errors are described in clause 9.

The following table provides the list of operations which may return each of the errors used on the SRF-SCF interface.



**Table 1: Available errors for each operation**

Errors	ARI	Ca	ScC	ScI	PA	PCUI	PRM	ScR
Cancelled					X	X	X	
ImproperCallerResponse						X	X	
MissingCustomerRecord	X1							
MissingParameter	X1	X	X	X	X	X	X	X
ParameterOutOfRange					X	X	X	
SystemFailure	X1	X	X	X	X	X	X	X
TaskRefused	X1	X	X	X	X	X	X	X
UnavailableResource		X	X	X	X	X	X	X
UnexpectedComponentSequence	X1	X1	X1	X1	X1	X1	X1	X1
UnexpectedDataValue	X1	X	X	X	X	X	X	X
UnexpectedParameter	X1	X	X	X	X	X	X	X
UnknownLegID					X			
UnknownSubscriber							X	
NOTE: X1 Only applicable for direct SCF-SRF case.								
<b>ARI</b> AssistingRoutingInformation								
<b>Ca</b> Cancel								
<b>ScC</b> ScriptClose								
<b>ScI</b> ScriptInformation								
<b>ScR</b> ScriptRun								
<b>PA</b> PlayAnnouncement								
<b>PCUI</b> PromptAndCollectUserInformation								
<b>PRM</b> PromptAndReceiveMessage								

### 11.1.1 Cancelled

The Error "Cancelled" is defined in EN 301 931-1.

#### 11.1.1.1 Operations SCF->SRF

This clause describes the procedure when the error for an operation invoked from the SCF occurs in the SRF.

Relevant operations are described in table 1.

##### 11.1.1.1.1 Procedures at invoking entity (SCF)

A) SCF sends the operation to SRF

Precondition: refer to the relevant "Operation procedures – SCF - precondition" clause.

Postcondition: refer to the relevant "Operation procedures – SCF - postcondition" clause.

The SCF sends a Cancel after a Play Announcement or PromptAndCollectUserInformation or PromptAndReceiveMessage has been sent. The SCF remains in the same state.

B) Receiving Cancelled Error

Precondition: refer to the relevant "Operation procedures – SCF - postcondition" clause.

Postcondition: SCF state: remains in the same state.

After sending a Cancel operation the Service Logic may continue (e.g., sending more PlayAnnouncement or PromptAndCollectUserInformation or PromptAndReceiveMessage or a DisconnectForwardConnection/DisconnectForwardConnectionWithArgument). The Cancelled Error can therefore be received in any state. The treatment is Service Logic dependent.

### 11.1.1.1.2 Procedures at responding entity (SRF)

#### A) Receiving Cancel

Precondition: refer to the relevant "Operation procedures SRF - precondition" clause.

Postcondition: refer to the relevant "Operation procedures SRF - postcondition" clause.

The indicated PlayAnnouncement or PromptAndCollectUserInformation or PromptAndReceiveMessage is terminated if it is presently executing or deleted from the buffer. If the indicated PlayAnnouncement or PromptAndCollectUserInformation or PromptAndReceiveMessage is already executed this causes a failure ("CancelFailed") to the Cancel operation.

#### B) Sending Cancel Error

Precondition: refer to the relevant "Operation procedures – SRF- postcondition" clause.

Postcondition: remain in the same state.

After returning the "Cancelled" Error the SRF stays in the same state. The execution of the indicated PlayAnnouncement or PromptAndCollectUserInformation or PromptAndReceiveMessage is aborted, i.e. the SRF remains connected and the next operation e.g. PlayAnnouncement or PromptAndCollectUserInformation is executed if available.

## 11.1.2 ImproperCallerResponse

The Error "ImproperCallerResponse" is defined in EN 301 931-1.

### 11.1.2.1 Operations SCF->SRF

This clause describes the procedure when the error for an operation invoked from the SCF occurs in the SRF.

Relevant operations are described in table 1.

#### 11.1.2.1.1 Procedures at invoking entity (SCF)

##### A) SCF sends the operation to SRF

Precondition: refer to the relevant "Operation procedures – SCF - precondition" clause.

Postcondition: refer to the relevant "Operation procedures – SCF - postcondition" clause.

##### B) SCF receives ImproperCallerResponse Error from SRF

Precondition: refer to the relevant "Operation procedures – SCF - postcondition" clause.

Postcondition: SCSM state: same state.

Error treatment depends on Service Logic. A SCF can initiate new User Interaction or force a Disconnect (to SSF).

#### 11.1.2.1.2 Procedures at responding entity (SRF)

##### A) SRF receives the operation

Precondition: refer to the relevant "Operation procedures SRF - precondition" clause.

Postcondition: refer to the relevant "Operation procedures SRF - postcondition" clause.

##### B) response from caller is not correct, SRF returns ImproperCallerResponse to SCF

Precondition: refer to the relevant "Operation procedures – SRF- postcondition" clause.

Postcondition: SRSM state 3: User Interaction.

SRF waits for a new Operation from SCF. This may be a new PromptAndCollectUserInformation or PlayAnnouncement.

### 11.1.3 MissingCustomerRecord

The Error "MissingCustomerRecord" is defined in EN 301 931-1.

#### 11.1.3.1 Operations SCF->SRF

None.

#### 11.1.3.2 Operations SRF->SCF

This clause describes the procedure when the error for an operation invoked from the SRF occurs in the SCF.

Relevant operations are described in table 1.

##### 11.1.3.2.1 Procedures at invoking entity (SRF)

###### A) Sending Operation

Precondition: refer to the relevant "Operation procedures SRF - precondition" clause.

Postcondition: refer to the relevant "Operation procedures SRF - postcondition" clause.

###### B) Receiving Error

Precondition: refer to the relevant "Operation procedures – SRF - postcondition" clause.

Postcondition: SRSMS state 1: Idle.

##### 11.1.3.2.2 Procedures at responding entity (SCF)

Precondition: refer to the relevant "Operation procedures SCF - precondition" clause.

Postcondition: refer to the relevant "Operation procedures SCF - postcondition" clause.

The SCSM detects the error in the received operation. The Error parameter is used to inform the SRF of this situation. The Service Logic and maintenance functions are informed. The SCF might try another SRF, route the call or release the call (Service Logic dependent).

### 11.1.4 MissingParameter

The Error "MissingParameter" is defined in EN 301 931-1.

#### 11.1.4.1 Operations SCF->SRF

This clause describes the procedure when the error for an operation invoked from the SCF occurs in the SRF.

Relevant operations are described in table 1.

##### 11.1.4.1.1 Procedures at invoking entity (SCF)

###### A) Sending Operation

Precondition: refer to the relevant "Operation procedures – SCF - precondition" clause.

Postcondition: refer to the relevant "Operation procedures – SCF - postcondition" clause.

###### B) Receiving Error

Precondition: refer to the relevant "Operation procedures – SCF - postcondition" clause.

Postcondition: SCSM state: remain in the same state.

Error treatment depends on Service logic. SCF can initiate new User Interaction or force Disconnect (to SSF).

#### 11.1.4.1.2 Procedures at responding entity (SRF)

Precondition: refer to the relevant "Operation procedures SRF - precondition" clause.

Postcondition: refer to the relevant "Operation procedures SRF - postcondition" clause.

The SRSM detects that a required parameter is not present in the Operation argument. The Error parameter MissingParameter is used to inform the SCF of this situation. The SCF should take the appropriate actions to treat this error.

#### 11.1.4.2 Operations SRF->SCF

This clause describes the procedure when the error for an operation invoked from the SRF occurs in the SCF.

Relevant operations are described in table 1.

##### 11.1.4.2.1 Procedures at invoking entity (SRF)

###### A) Sending Operation

Precondition: refer to the relevant "Operation procedures SRF - precondition" clause.

Postcondition: refer to the relevant "Operation procedures SRF - postcondition" clause.

###### B) Receiving Error

Precondition: refer to the relevant "Operation procedures - SRF - postcondition" clause.

Postcondition: SRSM state 1: Idle.

##### 11.1.4.2.2 Procedures at responding entity (SCF)

Precondition: refer to the relevant "Operation procedures SCF - precondition" clause.

Postcondition: refer to the relevant "Operation procedures SCF - postcondition" clause.

The SCSM detects the error in the received operation. The Error parameter is used to inform the SRF of this situation. The Service Logic and maintenance functions are informed. The SCF might try another SRF, route the call or release the call (Service Logic dependent).

#### 11.1.5 ParameterOutOfRange

The Error "ParameterOutOfRange" is defined in EN 301 931-1.

##### 11.1.5.1 Operations SCF->SRF

This clause describes the procedure when the error for an operation invoked from the SCF occurs in the SRF.

Relevant operations are described in table 1.

Refer to clause related to MissingParameter for the appropriate error procedures.

#### 11.1.6 SystemFailure

The Error "SystemFailure" is defined in EN 301 931-1.

##### 11.1.6.1 Operations SCF->SRF

This clause describes the procedure when the error for an operation invoked from the SCF occurs in the SRF.

Relevant operations are described in table 1.

Refer to clause 16.1.9 MissingParameter for the appropriate error procedures.

### 11.1.6.2 Operations SRF->SCF

This clause describes the procedure when the error for an operation invoked from the SRF occurs in the SCF.

Relevant operations are described in table 1.

Refer to clause related to MissingParameter for the appropriate error procedures.

### 11.1.7 TaskRefused

The Error "TaskRefused" is defined in EN 301 931-1.

#### 11.1.7.1 Operations SCF->SRF

This clause describes the procedure when the error for an operation invoked from the SCF occurs in the SRF.

Relevant operations are described in table 1.

#### 11.1.7.2 Operations SCF->SRF

Refer to clause related to MissingParameter for the appropriate error procedures.

#### 11.1.7.3 Operations SRF->SCF

This clause describes the procedure when the error for an operation invoked from the SRF occurs in the SCF.

Refer to clause related to MissingParameter for the appropriate error procedures.

### 11.1.8 UnavailableResource

The Error "UnavailableResource" is defined in EN 301 931-1.

#### 11.1.8.1 Operations SCF->SRF

This clause describes the procedure when the error for an operation invoked from the SCF occurs in the SRF.

Relevant operations are described in table 1.

#### 11.1.8.2 Operations SCF->SRF

##### 11.1.8.2.1 Procedures at invoking entity (SCF)

A) SCF sends PlayAnnouncement or PromptAndCollectUserInformation to SRF

Precondition: refer to the relevant "Operation procedures SCF - precondition" clause.

Postcondition: refer to the relevant "Operation procedures SCF - postcondition" clause.

B) SCF receives UnavailableResource Error from SRF

Precondition: refer to the relevant "Operation procedures SCF - postcondition" clause.

Postcondition: SRSM state 3: remain in the same state.

If the chosen resource cannot perform its function the further treatment is service dependent.

EXAMPLES: request SSF to connect to alternative SRF;  
service processing without PlayAnnouncement; or  
PromptAndCollectUserInformation (if possible);  
terminate service processing.

### 11.1.8.2.2 Procedures at responding entity (SRF)

A) SRF receiving an operation

Precondition: refer to the relevant "Operation procedures SRF - precondition" clause.

Postcondition: refer to the relevant "Operation procedures SRF - postcondition" clause.

B) SRF is not able to perform its function (and cannot be replaced). SRF sends UnavailableResource

Precondition: refer to the relevant "Operation procedures SRF - postcondition" clause.

Postcondition: SRSM remains in the state.

## 11.1.9 UnexpectedComponentSequence

The Error "UnexpectedComponentSequence" is defined in EN 301 931-1.

### 11.1.9.1 Operations SCF->SRF

This clause describes the procedure when the error for an operation invoked from the SCF occurs in the SRF.

Relevant operations are described in table 1.

In this case the SRF detects the erroneous situation, sends the UnexpectedComponentSequence error and remains in the same state. In the SCF, the Service Logic and maintenance functions are informed and the Service Logic decides about error treatment. Possible error treatment is to send the

DisconnectForwardConnection/DisconnectForwardConnectionArgument operation to the SSF.

### 11.1.9.2 Operations SRF->SCF

This clause describes the procedure when the error for an operation invoked from the SRF occurs in the SCF.

Relevant operations are described in table 1.

In this case, an error occurs if the SRF has already an established relationship with the SCF and sends an AssistRequestInstructions. The SCF detects the erroneous situation, informs Service Logic and maintenance functions and returns the error parameter. On receiving the parameter the SRF moves to idle and releases the temporary connection.

## 11.1.10 UnexpectedDataValue

The Error "UnexpectedDataValue" is defined in EN 301 931-1.

### 11.1.10.1 Operations SCF->SRF

This clause describes the procedure when the error for an operation invoked from the SCF occurs in the SRF.

Relevant operations are described in table 1.

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".

### 11.1.10.2 Operations SCF->SRF

Refer to clause related to MissingParameter for the appropriate error procedures.

### 11.1.10.3 Operations SRF->SCF

This clause describes the procedure when the error for an operation invoked from the SRF occurs in the SCF.

Relevant operations are described in table 1.

Refer to clause related to MissingParameter for the appropriate error procedures.

### 11.1.11 UnexpectedParameter

The Error "UnexpectedParameter" is defined in EN 301 931-1.

#### 11.1.11.1 Operations SCF->SRF

This clause describes the procedure when the error for an operation invoked from the SCF occurs in the SRF.

Relevant operations are described table 1.

Refer to clause related to MissingParameter for the appropriate error procedures.

#### 11.1.11.2 Operations SRF->SCF

This clause describes the procedure when the error for an operation invoked from the SRF occurs in the SCF.

Relevant operations are described table 1.

Refer to clause related to MissingParameter for the appropriate error procedures.

### 11.1.12 UnknownLegID

The Error "UnknownLegID" is defined in EN 301 931-1.

#### 11.1.12.1 Operations SCF->SRF

This clause describes the procedure when the error for an operation invoked from the SCF occurs in the SRF.

Relevant operations are described table 1.

Refer to clause related to MissingParameter for the appropriate error procedures.

### 11.1.13 UnknownSubscriber

The Error "UnknownSubscriber" is defined in EN 301 931-1.

#### 11.1.13.1 Operations SCF->SRF

This clause describes the procedure when the error for an operation invoked from the SCF occurs in the SRF.

Relevant operations are described table 1.

Refer to clause related to MissingParameter for the appropriate error procedures.

### 11.1.14 Expiration of TSRF

#### 11.1.14.1 General Description

##### 11.1.14.1.1 Error description

A timeout occurred in the SRF on the response from the SCF. This procedure concerns only the direct SCF-SRF case.

## 11.1.14.2 Procedures SRF->SCF

### 11.1.14.2.1 Procedures at the invoking entity (SRF)

Timeout occurs in SRF on TSRF.

Precondition: SRSMS state 2: Connected or SRSMS state 3: User Interaction.

Postcondition: SRSMS state 1: Idle.

The SRF aborts the dialogue and moves to the Idle state, all allocated resources are de-allocated. The abort is reported to the maintenance functions.

### 11.1.14.2.2 Procedures at the responding entity (SCF)

SCF receives a dialogue abort.

Precondition: SCSMS state: A connection between the user and the SRF has been established.

Postcondition: SCSMS state 2: Preparing SSF instructions.

The SCF releases all resources related to the dialogue, reports the abort to the maintenance functions and returns to state preparing SSF instructions.

## 12 ASN1 Definition

### 12.1 IN CS-3 Types

#### 12.1.1 Data Types

```
IN-CS3-scf-srf-datatypes {itu-t(0) identified-organization(4) etsi(0) inDomain(1) in-network(1)
cs3(30) modules(1) in-cs3-scf-srf-datatypes(10) version1(0)}
```

```
DEFINITIONS IMPLICIT TAGS ::=
```

```
BEGIN
```

```
IMPORTS
```

```
    tc-Messages,
    common-classes,
    common-datatypes,
    ssf-scf-classes,
    scf-srf-classes,
    ssf-scf-datatypes,
    ros-InformationObjects
```

```
FROM IN-CS3-object-identifiers
    {itu-t(0) identified-organization(4) etsi(0) inDomain(1) in-network(1) cs3(30) modules(1)
in-cs3-object-identifiers(0) version1(0) }
```

```
    EXTENSION,
    SupportedExtensions { },
    COMMON-BOUNDS
```

```
FROM IN-CS3-common-classes common-classes
```

```
    Integer4
```

```
FROM IN-CS3-common-datatypes common-datatypes
```

```
    Code
```

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

```
    Digits {},
    DisplayInformation {},
    SDSSinformation {}
```



```

FROM IN-CS3-SSF-SCF-datatypes ssf-scf-datatypes

    SCF-SSF-BOUNDS
FROM IN-CS3-SSF-SCF-Classes ssf-scf-classes

    SCF-SRF-BOUNDS

FROM IN-CS3-scf-srf-classes scf-srf-classes
;

CollectedDigits ::= SEQUENCE {
    minimumNbOfDigits [0] INTEGER (1..127)    DEFAULT 1,
    maximumNbOfDigits [1] INTEGER (1..127),
    endOfReplyDigit [2] OCTET STRING (SIZE (1..2))    OPTIONAL,
    cancelDigit [3] OCTET STRING (SIZE (1..2))    OPTIONAL,
    startDigit [4] OCTET STRING (SIZE (1..2))    OPTIONAL,
    firstDigitTimeout [5] INTEGER (1..127)    OPTIONAL,
    interDigitTimeout [6] INTEGER (1..127)    OPTIONAL,
    errorTreatment [7] ErrorTreatment DEFAULT reportErrorToScf,
    interruptableAnnInd [8] BOOLEAN DEFAULT TRUE,
    voiceInformation [9] BOOLEAN DEFAULT FALSE,
    voiceBack [10] BOOLEAN    DEFAULT FALSE,
    detectModem [11] BOOLEAN    DEFAULT FALSE,
    ...
}
-- The use of voiceBack is network operator specific.
-- The endOfReplyDigit, cancelDigit, and startDigit parameters have been designated as OCTET STRING,
-- and are to be encoded as BCD, one digit per octet only, contained
-- in the four least significant bits of each OCTET. The usage is service dependent.
-- firstDigitTimeout and interDigitTimeout are measured in seconds.

CollectedInfo ::= CHOICE {
    collectedDigits [0] CollectedDigits,
    ia5Information [1] BOOLEAN,
    detectModem [2] BOOLEAN
}

ElementaryMessageID ::= Integer4

ErrorTreatment ::= ENUMERATED {
    reportErrorToScf(0),
    help(1),
    repeatPrompt(2)
}
-- reportErrorToScf means returning the "ImproperCallerResponse" error in the event of an error

-- condition during collection of user info.

GapOnResource ::= Code

InbandInfo {SCF-SSF-BOUNDS: bound2, SCF-SRF-BOUNDS: bound3} ::= SEQUENCE {
    messageID [0] MessageID {bound2, bound3},
    numberOfRepetitions [1] INTEGER (1..127)    OPTIONAL,
    duration [2] INTEGER (0..32767)    OPTIONAL,
    interval [3] INTEGER (0.. 32767)    OPTIONAL,
    preferredLanguage [4] Language    OPTIONAL,
    ...
}
-- Interval is the time in seconds between each repeated announcement. Duration is the total
-- amount of time in seconds, including repetitions and intervals.
-- The end of announcement is either the end of duration or numberOfRepetitions, whatever comes
-- first.
-- duration with value 0 indicates infinite duration

InformationToRecord { SCF-SRF-BOUNDS: bound3} ::= SEQUENCE {
    messageID [0] ElementaryMessageID OPTIONAL,
    messageDeletionTimeout [1] INTEGER (1..3600)    OPTIONAL,
    --Time units = hours
    timeToRecord [3] INTEGER (0..bound3.&maxRecordingTime)    OPTIONAL,
    --Time units = seconds
    controlDigits [4] SEQUENCE {
        endOfRecordingDigit [0] OCTET STRING (SIZE(1..2))    OPTIONAL,
        cancelDigit [1] OCTET STRING (SIZE(1..2))    OPTIONAL,
        replayDigit [2] OCTET STRING (SIZE(1..2))    OPTIONAL,
        restartRecordingDigit [3] OCTET STRING (SIZE(1..2))    OPTIONAL,
        restartAllowed [4] BOOLEAN    DEFAULT FALSE,
    }
}

```

```

        replayAllowed [5] BOOLEAN    DEFAULT FALSE,
        ...
    },
    ...
}

InformationToSend { COMMON-BOUNDS: bound1, SCF-SSF-BOUNDS: bound2, SCF-SRF-BOUNDS: bound3} ::= CHOICE
{
    inbandInfo [0] InbandInfo {bound1, bound3},
    tone [1] Tone,
    displayInformation [2] DisplayInformation {bound3},
    sDSSinformation [3] SDSSinformation {bound2}
}

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

MailBoxID { SCF-SRF-BOUNDS: bound3} ::= OCTET STRING (SIZE(
    bound3.&minMailBoxIDLength..bound3.&maxMailBoxIDLength))

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

MessageID { SCF-SSF-BOUNDS: bound2, SCF-SRF-BOUNDS: bound3} ::= CHOICE {
    elementaryMessageID [0] Integer4,
    text [1] SEQUENCE {
        messageContent [0] IA5String (SIZE (
            bound3.&minMessageContentLength..
            bound3.&maxMessageContentLength)),
        attributes [1] OCTET STRING (SIZE (
            bound3.&minAttributesLength..
            bound3.&maxAttributesLength))
            OPTIONAL,
        ...
    },
    elementaryMessageIDs [29] SEQUENCE SIZE (1.. bound3.&numOfMessageIDs) OF Integer4,
    variableMessage [30] SEQUENCE {
        elementaryMessageID[0] Integer4,
        variableParts [1] SEQUENCE SIZE (1.. bound3.&maxVariableParts) OF VariablePart
    }
}

-- OPTIONAL denotes network operator specific use.

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

RecordedMessageID ::= Integer4

SRFGapCriteria { SCF-SSF-BOUNDS: bound2} ::= CHOICE {
    ipAddressValue [1] Digits {bound2},
    gapOnResource [2] GapOnResource,
    ipAddressAndresource [3] SEQUENCE {
        ipAddressValue [1] Digits {bound2},
        gapOnResource [2] GapOnResource,
        ...
    }
}

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

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

```

```

VariablePart { SCF-SSF-BOUNDS: bound2} ::= CHOICE {
    integer [0] Integer4,
    number  [1] Digits { bound2},    -- Generic digits
    time    [2] OCTET STRING (SIZE(2)), -- HH:MM, BCD coded
    date    [3] OCTET STRING (SIZE(3..4)),
    -- size 3: YYMMDD, BCD coded.
    -- size 4: YYYYMMDD, BCD coded
    price   [4] OCTET STRING (SIZE(4))
}
-- Indicates the variable part of the message.
-- BCD coded variable parts are encoded as described in the examples below.
-- The most significant hours digit occupies bits 0-3 of the first octet, and the least
-- significant digit occupies bits 4-7 of the first octet. The most significant minutes digit
-- occupies bits 0-3 of the second octet, and the least significant digit occupies bits 4-7
-- of the second octet.
-- For example, time = 12:15 would be encoded as:
-- Bits          HGFE          DCBA
-- leading octet  2    1
--                5    1
-- OPTION 1 (date: Size 3):
-- date = 1993 September 30th would be encoded as:
-- Bits          HGFE          DCBA
-- leading octet  3    9
--                9    0
--                0    3
-- For a system operating when or after this Recommendation is released, the 2 digit value
-- representing a Year shall be interpreted as follows:
-- - If the two-digits value is 00 through 49 inclusive, it shall be interpreted as
representing
--     year 2000 through 2049.
-- - If the two-digits value is 50 through 99 inclusive, it shall be interpreted as
representing
--     year 1950 through 1999.
-- OPTION 2 (date: Size 4):
-- Date is BCD encoded. The year digit indicating millenium occupies bits 0-3 of the first octet,
-- and the year digit indicating century occupies bits 4-7 of the first octet. The year digit
-- indicating decade occupies bits 0-3 of the second octet, whilst the digit indicating the year
-- within the decade occupies bits 4-7 of the second octet.
-- The most significant month digit occupies bits 0-3 of the third octet, and the least
-- significant month digit occupies bits 4-7 of the third octet. The most significant day digit
-- occupies bits 0-3 of the fourth octet, and the least significant day digit occupies bits 4-7
-- of the fourth octet.
-- Price is BCD encoded. The digit indicating hundreds of thousands occupies bits 0-3 of the
-- first octet, and the digit indicating tens of thousands occupies bits 4-7 of the first octet.
-- The digit indicating thousands occupies bits 0-3 of the second octet, whilst the digit
-- indicating hundreds occupies bits 4-7 of the second octet. The digit indicating tens occupies
-- bits 0-3 of the third octet, and the digit indicating 0 to 9 occupies bits 4-7 of the third
-- octet. The tenths digit occupies bits 0-3 of the fourth octet, and the hundredths digit
-- occupies bits 4-7 of the fourth octet.
--
-- For the encoding of digits in an octet, refer to the timeAndtimezone parameter

```

END

## 12.1.2 Classes

```

IN-CS3-scf-srf-classes {itu-t(0) identified-organization(4) etsi(0) inDomain(1) in-network(1)
cs3(30) modules(1) in-cs3-scf-srf-classes (11) version1(0)}

```

DEFINITIONS::=

BEGIN

IMPORTS

```

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

```

```

FROM IN-CS3-object-identifiers {itu-t(0) identified-organization(4) etsi(0) inDomain(1) in-
network(1) cs3(30) modules(1) in-cs3-object-identifiers(0) version1(0) }

```

```

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

```

```
;
```

```

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

```

```

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

```

```

-- SpecificInfo is used for ScriptRun and ScriptInformation and ScriptClose
-- Result is used for ScritEvent

```

```

firstScript UISCRIPT ::=

```

```

{
    IDENTIFIED BY local:1
}

```

```

-- firstScript is just an example.

```

```

SupportedUIScripts UISCRIPT ::= {firstScript,...

```

```

--full set of User Interaction script
}

```

```

-- SupportedUIScripts is the full set of User Interaction scripts.

```

```

SCF-SRF-BOUNDS ::= CLASS

```

```

{
    &minAttributesLength INTEGER OPTIONAL,
    &maxAttributesLength INTEGER OPTIONAL,
    &minMailBoxIDLength INTEGER OPTIONAL,
    &maxMailBoxIDLength INTEGER OPTIONAL,
    &minMessageContentLength INTEGER OPTIONAL,
    &maxMessageContentLength INTEGER OPTIONAL,
    &minReceivedInformationLength INTEGER OPTIONAL,
    &maxReceivedInformationLength INTEGER OPTIONAL,
    &maxRecordingTime INTEGER OPTIONAL,
    &numOfMessageIDs INTEGER OPTIONAL,
    &maxRecordedMessageUnits INTEGER OPTIONAL,
    &maxVariableParts INTEGER OPTIONAL
}

```

```

WITH SYNTAX

```

```

{
    MINIMUM-FOR-ATTRIBUTES &minAttributesLength
    MAXIMUM-FOR-ATTRIBUTES &maxAttributesLength
    MINIMUM-FOR-MAIL-BOX-ID &minMailBoxIDLength
    MAXIMUM-FOR-MAIL-BOX-ID &maxMailBoxIDLength
    MINIMUM-FOR-MESSAGE-CONTENT &minMessageContentLength
    MAXIMUM-FOR-MESSAGE-CONTENT &maxMessageContentLength
    MINIMUM-FOR-RECEIVED-INFORMATION &minReceivedInformationLength
    MAXIMUM-FOR-RECEIVED-INFORMATION &maxReceivedInformationLength
    MAXIMUM-FOR-RECORDING-TIME &maxRecordingTime
    NUM-OF-MESSAGE-IDS &numOfMessageIDs
    MAXIMUM-FOR-RECORDED-MESSAGE-UNITS &maxRecordedMessageUnits
    NUM-OF-VARIABLE-PARTS &maxVariableParts
}

```

```

-- The following instance of the parameter bound is just an example

```

```

networkSpecificSCFSRFBoundSet SCF-SRF-BOUNDS ::=

```

```

{

```

```

MINIMUM-FOR-ATTRIBUTES      1
MAXIMUM-FOR-ATTRIBUTES      5
MINIMUM-FOR-MAIL-BOX-ID     1
MAXIMUM-FOR-MAIL-BOX-ID     5
MINIMUM-FOR-MESSAGE-CONTENT 1
MAXIMUM-FOR-MESSAGE-CONTENT 5
MINIMUM-FOR-RECEIVED-INFORMATION 1
MAXIMUM-FOR-RECEIVED-INFORMATION 5
MAXIMUM-FOR-RECORDING-TIME  5
NUM-OF-MESSAGE-IDS          2
MAXIMUM-FOR-RECORDED-MESSAGE-UNITS 5
NUM-OF-VARIABLE-PARTS      5 -- must be 5 or
-- greater.
}
END

```

## 12.2 Operations and Arguments

```

IN-CS3-scf-srf-ops-args {itu-t(0) identified-organization(4) etsi(0) inDomain(1) in-network(1)
cs3(30) modules(1) in-cs3-scf-srf-ops-args (12) version1(0)}

```

```

DEFINITIONS IMPLICIT TAGS ::=

```

```

BEGIN

```

```

IMPORTS

```

```

    ros-InformationObjects,
    operationcodes,
    common-datatypes,
    errortypes,
    common-classes,
    scf-srf-classes,
    ssf-scf-classes,
    ssf-scf-datatypes,
    scf-srf-datatypes

```

```

FROM IN-CS3-object-identifiers
{ itu-t(0) identified-organization(4) etsi(0) inDomain(1) in-network(1) cs3(30) modules(1) in-cs3-
object-identifiers(0) version1(0) }

```

```

OPERATION

```

```

FROM Remote-Operations-Information-Objects ros-InformationObjects

```

```

    opcode-playAnnouncement,
    opcode-promptAndCollectUserInformation,
    opcode-promptAndReceiveMessage,
    opcode-scriptClose,
    opcode-scriptEvent,
    opcode-scriptInformation,
    opcode-scriptRun,
    opcode-specializedResourceReport,
    opcode-activityTest,
    opcode-srfCallGap

```

```

FROM IN-CS3-scf-srf-operationcodes operationcodes

```

```

SCF-SSF-BOUNDS

```

```

FROM IN-CS3-SSF-SCF-classes ssf-scf-classes

```

```

    CallSegmentID {},
    Digits {},
    GenericNumber {},
    LegID,
    ControlType,
    GapIndicators

```

```

FROM IN-CS3-SSF-SCF-datatypes ssf-scf-datatypes

    InformationToSend {},
    CollectedInfo,
    MailBoxID {},
    InformationToRecord {},
    Media,
    ReceivedStatus,
    RecordedMessageID,
    SRFGapCriteria {}

FROM IN-CS3-scf-srf-datatypes scf-srf-datatypes

    canceled,
    improperCallerResponse,
    missingParameter,
    parameterOutOfRange,
    systemFailure,
    taskRefused,
    unavailableResource,
    unexpectedComponentSequence,
    unexpectedDataValue,
    unexpectedParameter,
    unknownLegID,
    unknownSubscriber

FROM IN-CS3-erroratypes erroratypes

    UISCRIPT,
    SupportedUIScripts,
    SCF-SRF-BOUNDS

FROM IN-CS3-scf-srf-classes scf-srf-classes

    Extensions

FROM IN-CS3-common-datatypes common-datatypes

    COMMON-BOUNDS

FROM IN-CS3-common-classes common-classes
;
B1::=COMMON-BOUNDS
B2::=SCF-SSF-BOUNDS
B3::=SCF-SRF-BOUNDS

-- activityTest OPERATION
-- This operation is described in EN 301 931-2.

-- assistRequestInstruction OPERATION
-- This operation is described in EN 301 931-2.

--cancel OPERATION
-- This operation is described in EN 301 931-2.

--connectToResource OPERATION
-- This operation is described in EN 301 931-2.

--disconnectForwardConnection OPERATION
-- This operation is described in EN 301 931-2.

-- disconnectForwardConnectionWithArgument OPERATION
-- This operation is described in EN 301 931-2.

-- establishTemporaryConnection OPERATION
-- This operation is described in EN 301 931-2.

```

```

playAnnouncement { B1: bound1, B2: bound2, B3: bound3 } OPERATION ::= {
  ARGUMENT    PlayAnnouncementArg { bound1,bound2,bound3}
  RETURN RESULT  FALSE
  ERRORS      {canceled |
               missingParameter |
               parameterOutOfRange |
               systemFailure |
               taskRefused |
               unexpectedComponentSequence |
               unexpectedDataValue |
               unexpectedParameter |
               unavailableResource |
               unknownLegID
              }
  LINKED      {specializedResourceReport}
  ALWAYS RESPONDS  FALSE
  CODE        opcode-playAnnouncement
}
-- Direction: SCF -> SRF, Timer: Tpa
-- This operation is to be used after Establish Temporary Connection (assist procedure with a second
SSP)
-- or a Connect to Resource (no assist) operation. It may be used for inband interaction with an
analog user,
-- or for interaction with an ISDN user. In the former case, the SRF is usually collocated with the
SSF for
-- standard tones (congestion tone...) or standard announcements. In the latter case, the SRF is
always
-- collocated with the SSF in the switch. Any error is returned to the SCF. The timer associated
with this
-- operation must be of a sufficient duration to allow its linked operation to be correctly
correlated.

PlayAnnouncementArg {B1: bound1, B2: bound2, B3: bound3} ::= SEQUENCE {
  informationToSend [0] InformationToSend { bound1, bound2, bound3},
  disconnectFromIPForbidden [1] BOOLEAN DEFAULT TRUE,
  requestAnnouncementComplete [2] BOOLEAN DEFAULT TRUE,
  extensions [3] Extensions {bound1} OPTIONAL,
  connectedParty CHOICE {
    legID [4] LegID,
    callSegmentID [5] CallSegmentID {bound2}
  } OPTIONAL,
  ...
}

promptAndCollectUserInformation { B1: bound1, B2: bound2, B3: bound3 } OPERATION ::= {
  ARGUMENT    PromptAndCollectUserInformationArg {bound1,bound2,bound3}
  RESULT      ReceivedInformationArg {bound2}
  ERRORS      {canceled |
               improperCallerResponse |
               missingParameter |
               parameterOutOfRange |
               systemFailure |
               taskRefused |
               unexpectedComponentSequence |
               unavailableResource |
               unexpectedDataValue |
               unexpectedParameter
              }
  CODE        opcode-promptAndCollectUserInformation
}
-- Direction: SCF -> SRF, Timer: Tpc
-- This operation is used to interact with a user to collect information.

PromptAndCollectUserInformationArg { B1: bound1, B2: bound2, B3: bound3 } ::= SEQUENCE {
  collectedInfo [0] CollectedInfo,
  disconnectFromIPForbidden [1] BOOLEAN DEFAULT TRUE,
  informationToSend [2] InformationToSend { bound1, bound2, bound3} OPTIONAL,
  extensions [3] Extensions {bound1} OPTIONAL,
  callSegmentID [4] CallSegmentID {bound2} OPTIONAL,
  ...
}

ReceivedInformationArg {B2: bound2} ::= CHOICE {
  digitsResponse [0] Digits {bound2},
  ia5Response [1] IA5String,
  modemdetected [2] BOOLEAN
}

```

```

promptAndReceiveMessage { B1: bound1, B2: bound2, B3: bound3 } OPERATION ::= {
  ARGUMENT      PromptAndReceiveMessageArg{bound1,bound2,bound3}
  RESULT        MessageReceivedArg { bound1,bound3}
  ERRORS        {canceled |
                 improperCallerResponse |
                 missingParameter |
                 parameterOutOfRange |
                 taskRefused |
                 systemFailure |
                 unavailableResource |
                 unexpectedComponentSequence |
                 unexpectedDataValue |
                 unexpectedParameter |
                 unknownSubscriber
                }
  CODE          opcode-promptAndReceiveMessage
}
-- Direction: SCF ->SRF, Timer: Tprm
-- Used to prompt a user to store a message

PromptAndReceiveMessageArg { B1: bound1, B2: bound2, B3: bound3 } ::= SEQUENCE {
  disconnectFromIPForbidden [0] BOOLEAN DEFAULT TRUE,
  informationToSend [1] InformationToSend { bound1, bound2, bound3} OPTIONAL,
  extensions [3] Extensions {bound1} OPTIONAL,
  subscriberID [4] GenericNumber {bound2} OPTIONAL,
  mailBoxID [5] MailBoxID {bound3} OPTIONAL,
  informationToRecord [6] InformationToRecord {bound3},
  media [7] Media DEFAULT voiceMail,
  callSegmentID [8] CallSegmentID {bound2} OPTIONAL,
  ...
}

MessageReceivedArg { B1: bound1, B3: bound3 } ::= SEQUENCE {
  receivedStatus [0] ReceivedStatus,
  recordedMessageID [1] RecordedMessageID OPTIONAL,
  recordedMessageUnits [2] INTEGER(1..bound3.&maxRecordedMessageUnits) OPTIONAL,
  extensions [3] Extensions {bound1} OPTIONAL,
  ...
}

-- reportUTSI OPERATION
-- This operation is described in EN 301 931-2.

-- requestReportUTSI OPERATION
-- This operation is described in EN 301 931-2.

-- sendSTUI OPERATION
-- This operation is described in EN 301 931-2.

scriptClose { B1: bound1, B2: bound2} OPERATION ::= {
  ARGUMENT      ScriptCloseArg { bound1,bound2 }
  RETURN RESULT FALSE
  ERRORS        {
                 systemFailure |
                 missingParameter |
                 taskRefused |
                 unavailableResource |
                 unexpectedComponentSequence |
                 unexpectedDataValue |
                 unexpectedParameter
                }
  ALWAYS RESPONDS FALSE
  CODE          opcode-scriptClose
}
-- Direction:SCF-> SRF. Timer:TC1
-- This operation is issued by the SCF to deallocate the resources used to perform the
-- instance of the "User Interaction" script: the context is released.

ScriptCloseArg { B1: bound1, B2: bound2 } ::= SEQUENCE {
  uIScriptId UISCRIPT.&id({SupportedUIScripts}),
  uIScriptSpecificInfo [0] UISCRIPT.&SpecificInfo({SupportedUIScripts }{@uIScriptId})
  OPTIONAL,
  extensions [1] Extensions {bound1} OPTIONAL,
  callSegmentID [2] CallSegmentID {bound2} OPTIONAL,
  ...
}

```



```

}

scriptEvent { B1: bound1, B2: bound2 } OPERATION::= {
  ARGUMENT    ScriptEventArg { bound1, bound2 }
  RETURN RESULT  FALSE
  ALWAYS RESPONDS FALSE
  CODE    opcode-scriptEvent
}
-- Direction:SRF-> SCF. Timer:TRe
-- This operation is issued by the SRF to return information to the SCF on the results of the
-- execution of the instance of User Interaction script.

ScriptEventArg { B1: bound1, B2: bound2 } ::= SEQUENCE {
  uIScriptId    UISCRIPT.&id({SupportedUIScripts}),
  uIScriptResult [0] UISCRIPT.&Result({SupportedUIScripts }{@uIScriptId})
  OPTIONAL,
  extensions [1] Extensions {bound1} OPTIONAL,
  callSegmentID [2] CallSegmentID {bound2} OPTIONAL,
  lastEventIndicator [3] BOOLEAN    DEFAULT FALSE,
  ...
}

scriptInformation { B1: bound1, B2: bound2 } OPERATION::={
  ARGUMENT    ScriptInformationArg { bound1, bound2 }
  RETURN RESULT  FALSE
  ERRORS {
    systemFailure |
    missingParameter |
    taskRefused |
    unavailableResource |
    unexpectedComponentSequence |
    unexpectedDataValue |
    unexpectedParameter
  }
  ALWAYS RESPONDS    FALSE
  CODE    opcode-scriptInformation
}
-- Direction:SCF-> SRF. Timer:Tinf

ScriptInformationArg { B1: bound1, B2: bound2 } ::= SEQUENCE {
  uIScriptId    UISCRIPT.&id({SupportedUIScripts}),
  uIScriptSpecificInfo [0] UISCRIPT.&SpecificInfo({SupportedUIScripts }{@uIScriptId})
  OPTIONAL,
  extensions [1] Extensions {bound1} OPTIONAL,
  callSegmentID [2] CallSegmentID {bound2} OPTIONAL,
  ...
}

scriptRun { B1: bound1, B2: bound2 } OPERATION::={
  ARGUMENT    ScriptRunArg { bound1, bound2 }
  RETURN RESULT  FALSE
  ERRORS {
    systemFailure |
    missingParameter |
    taskRefused |
    unavailableResource |
    unexpectedComponentSequence |
    unexpectedDataValue |
    unexpectedParameter
  }
  ALWAYS RESPONDS    FALSE
  CODE    opcode-scriptRun
}
-- Direction:SCF-> SRF. Timer:Tru
-- This operation is issued by the SCF to allocate the necessary resources to perform the
-- instance of the "User Interaction" script and then to activate this "User Interaction" script
-- instance. A context is partially defined for it if necessary.

ScriptRunArg { B1: bound1, B2: bound2 } ::= SEQUENCE {
  uIScriptId    UISCRIPT.&id({SupportedUIScripts}),
  uIScriptSpecificInfo [0] UISCRIPT.&SpecificInfo({SupportedUIScripts }{@uIScriptId})
  OPTIONAL,
  extensions [1] Extensions {bound1} OPTIONAL,
  disconnectFromIPForbidden [2] BOOLEAN DEFAULT TRUE,
  callSegmentID [3] CallSegmentID {bound2} OPTIONAL,
  ...
}

```

```

}

specializedResourceReport OPERATION ::= {
  ARGUMENT    SpecializedResourceReportArg
  RETURN RESULT  FALSE
  ALWAYS RESPONDS FALSE
  CODE        opcode-specializedResourceReport
}
-- Direction: SRF -> SCF, Timer: Tsrr
-- This operation is used as the response to a PlayAnnouncement operation when the announcement
-- completed
-- report indication is set.

SpecializedResourceReportArg ::= NULL

srfCallGap { B1: bound1 } OPERATION ::= {
  ARGUMENT    srfCallGapArg {bound1}
  RETURN RESULT  FALSE
  ALWAYS RESPONDS  FALSE
  CODE        opcode-srfCallGap
}
-- Direction: SRF -> SCF, Timer: Tcg
-- This operation is used to request the SCF to reduce the rate at which specific service requests
-- are sent to
-- the SRF.

srfCallGapArg { B1: bound1 } ::= SEQUENCE {
  srfGapCriteria [0] srfGapCriteria {bound1},
  gapIndicators [1] GapIndicators,
  controlType [2] ControlType OPTIONAL,
  . . . ,
  . . . ,
  extensions [30] Extensions {bound1} OPTIONAL
}

END

```

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

- short: 1 to 10 seconds;
- medium: 1 to 60 seconds;
- long: 1 second to 30 minutes.

Table 2 lists all operation timers and the value range for each timer. The definitive value for each operation timer may be network specific and has to be defined by the network operator.

**Table 2: Operation timers and their value range**

Operation Name	Timer	value range
ScriptClose	T <sub>cl</sub>	short
ScriptInformation	T <sub>inf</sub>	Short
PlayAnnouncement	T <sub>pa</sub>	Long
PromptAndCollectUserInformation	T <sub>pc</sub>	Long
PromptAndReceiveMessage	T <sub>prm</sub>	Long
ScriptEvent	T <sub>re</sub>	Short
ScriptRun	T <sub>ru</sub>	Long
SpecializedResourceReport	T <sub>srr</sub>	Short
SRFActivitytest	T <sub>at</sub>	Short
SRFCallGap	T <sub>cg</sub>	Short

## 12.3 Package, contracts and Application Contexts

### 12.3.1 ASN.1 modules

```
IN-CS3-SCF-SRF-pkgs-contracts-acsc { itu-t(0) identified-organization(4) etsi(0) inDomain(1) in-
network(1) cs3(30) modules(1) in-cs3-scf-srf-pkgs-contracts-acsc(13) version1(0)}
```

```
DEFINITIONS ::=
```

```
BEGIN
```

```
-- This module describes the operation-packages, contracts and application-contexts used
-- over the SCF-SRF interface.
```

```
IMPORTS
```

```
id-package-specializedResourceControl,
id-ac-srf-scfAC,
id-contract-srf-scf,
id-package-srf-scfCancel,
id-package-scriptControl,
id-package-messageControl,
id-package-srfManagement,
id-package-activityTest,
id-as-basic-srf-scf,
common-classes, ros-InformationObjects, tc-Messages, tc-NotationExtensions,
scf-srf-Operations, ssf-scf-Operations, ssf-scf-classes, scf-srf-classes, ssf-scf-Protocol
```

```
FROM IN-CS3-object-identifiers { itu-t(0) identified-organization(4) etsi(0) inDomain(1) in-
network(1) cs3(30) modules(1) in-cs3-object-identifiers (0) version1(0)}
```

```
SCF-SSF-BOUNDS
```

```
FROM IN-CS3-SSF-SCF-Classes ssf-scf-classes
```

```
COMMON-BOUNDS
```

```
FROM IN-CS3-common-classes common-classes
```

```
SCF-SRF-BOUNDS,
networkSpecificSCFSRFBoundSet
```

```
FROM IN-CS3-scf-srf-classes scf-srf-classes
```

```
ROS-OBJECT-CLASS, CONTRACT, OPERATION-PACKAGE, OPERATION
```

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

```
TCMessage {}
FROM TCAPMessages tc-Messages
```

```
APPLICATION-CONTEXT, dialogue-abstract-syntax
FROM TC-Notation-Extensions tc-NotationExtensions
```

```
playAnnouncement {},
promptAndReceiveMessage {},
promptAndCollectUserInformation {},
scriptClose {},
scriptEvent {},
scriptInformation {},
scriptRun {},
specializedResourceReport,
sRFCallGap {}
```

```
FROM IN-CS3-scf-srf-ops-args scf-srf-Operations
```

```
cancel {},
assistRequestInstructions {},
activityTest
```

```
FROM IN-CS3-SSF-SCF-ops-args ssf-scf-Operations
```

```
srf-scfActivationOfAssistPackage {},
activityTestPackage
```

```

FROM IN-CS3-SSF-SCF-pkgs-contracts-ac srf-scf-Protocol
;

B1::=COMMON-BOUNDS
B2::=SCF-SSF-BOUNDS
B3::=SCF-SRF-BOUNDS

networkSpecificBoundSet COMMON-BOUNDS::=
{
  NUM-OF-EXTENSIONS 1
}

networkSpecificSSFSCFBoundSet SCF-SSF-BOUNDS::=
{
  MINIMUM-FOR-DIGITS 1
  MAXIMUM-FOR-DIGITS 5
  MINIMUM-FOR-DISPLAY 1
  MAXIMUM-FOR-DISPLAY 5
  MINIMUM-FOR-GENERIC-NUMBER 1
  MAXIMUM-FOR-GENERIC-NUMBER 5
  MINIMUM-FOR-SCF-ID 1
  MAXIMUM-FOR-SCF-ID 5
  MINIMUM-FOR-SDSS-INFORMATION 1
  MAXIMUM-FOR-SDSS-INFORMATION 5
  NUM-OF-GENERIC-NUMBERS 2
}

-- Application Contexts --

srf-scf-ac APPLICATION-CONTEXT::= {
  CONTRACT srf-scf-contract
  DIALOGUE MODE structured
  TERMINATION basic
  ABSTRACT SYNTAXES {dialogue-abstract-syntax |
    srf-scf-abstract-syntax }
  APPLICATION CONTEXT NAME id-ac-srf-scfAC}

-- Contracts --

srf-scf-contract CONTRACT::= {
  INITIATOR CONSUMER OF {srf-scfActivationOfAssistPackage {networkSpecificBoundSet,
networkSpecificSSFSCFBoundSet }}
    srfManagementPackage {networkSpecificBoundSet}
}
  RESPONDER CONSUMER OF {specializedResourceControlPackage {networkSpecificBoundSet,
networkSpecificSSFSCFBoundSet, networkSpecificSCFSRFBoundSet }}|
    srf-scfCancelPackage {networkSpecificSSFSCFBoundSet}|
    scriptControlPackage {networkSpecificBoundSet, networkSpecificSSFSCFBoundSet }|
    activityTestPackage |
    messageControlPackage {networkSpecificBoundSet, networkSpecificSSFSCFBoundSet,
networkSpecificSCFSRFBoundSet }}
  ID id-contract-srf-scf }

-- specializedResourceControl package --

specializedResourceControlPackage { B1: bound1, B2: bound2, B3: bound3 } OPERATION-PACKAGE::= {
  CONSUMER INVOKES {playAnnouncement { bound1,bound2,bound3} |
    promptAndCollectUserInformation { bound1,bound2,bound3}
}
  SUPPLIER INVOKES {specializedResourceReport}
  ID id-package-specializedResourceControl}

-- srf-scfCancel package --

srf-scfCancelPackage {B2: bound2} OPERATION-PACKAGE::= {
  CONSUMER INVOKES {cancel {bound2 }}
  ID id-package-srf-scfCancel}

```

```

-- scriptControl package --

scriptControlPackage { B1: bound1, B2: bound2 } OPERATION-PACKAGE ::= {
    CONSUMER INVOKES { scriptClose {bound1, bound2}| scriptRun {bound1, bound2} |
        scriptInformation { bound1,bound2 }}
    SUPPLIER INVOKES { scriptEvent { bound1,bound2 }}
    ID id-package-scriptControl}

-- messageControl package

messageControlPackage { B1: bound1, B2: bound2, B3: bound3 } OPERATION-PACKAGE ::= {
    CONSUMER INVOKES {promptAndReceiveMessage { bound1,bound2,bound3}}
    ID id-package-messageControl
}

-- SRF management package

srfManagementPackage { B1: bound1} OPERATION-PACKAGE ::= {
    CONSUMER INVOKES {sRFCallGap { bound1}}
    ID id-package-srfManagement
}

-- Abstract Syntaxes --

srf-scf-abstract-syntax ABSTRACT-SYNTAX ::= {
    BASIC-SRF-SCF-PDUs
    IDENTIFIED BY id-as-basic-srf-scf}

BASIC-SRF-SCF-PDUs ::= TCMMessage {{SRF-SCF-Invokable},{SRF-SCF-Returnable} }

SRF-SCF-Invokable OPERATION ::= {
    assistRequestInstructions {networkSpecificBoundSet, networkSpecificSSFSCFBoundSet }|
    cancel {networkSpecificSSFSCFBoundSet}|
    playAnnouncement {networkSpecificBoundSet, networkSpecificSSFSCFBoundSet,
networkSpecificSCFSRFBoundSet }|
    promptAndCollectUserInformation {networkSpecificBoundSet, networkSpecificSSFSCFBoundSet,
networkSpecificSCFSRFBoundSet }|
    scriptClose {networkSpecificBoundSet, networkSpecificSSFSCFBoundSet }|
    scriptEvent {networkSpecificBoundSet, networkSpecificSSFSCFBoundSet }|
    scriptInformation {networkSpecificBoundSet, networkSpecificSSFSCFBoundSet }|
    scriptRun {networkSpecificBoundSet, networkSpecificSSFSCFBoundSet }|
    specializedResourceReport |
    promptAndReceiveMessage {networkSpecificBoundSet, networkSpecificSSFSCFBoundSet,
networkSpecificSCFSRFBoundSet }|
    sRFCallGap{networkSpecificBoundSet}|
    activityTest
}

SRF-SCF-Returnable OPERATION ::= {
    assistRequestInstructions {networkSpecificBoundSet, networkSpecificSSFSCFBoundSet }|
    cancel {networkSpecificSSFSCFBoundSet}|
    playAnnouncement {networkSpecificBoundSet, networkSpecificSSFSCFBoundSet,
networkSpecificSCFSRFBoundSet }|
    promptAndCollectUserInformation {networkSpecificBoundSet, networkSpecificSSFSCFBoundSet,
networkSpecificSCFSRFBoundSet }|
    scriptClose {networkSpecificBoundSet, networkSpecificSSFSCFBoundSet }|
    scriptInformation {networkSpecificBoundSet, networkSpecificSSFSCFBoundSet }|
    scriptRun {networkSpecificBoundSet, networkSpecificSSFSCFBoundSet }|
    promptAndReceiveMessage {networkSpecificBoundSet, networkSpecificSSFSCFBoundSet,
networkSpecificSCFSRFBoundSet }|
    activityTest
}

END

```

## 13 Services assumed from TCAP

Common procedures are defined in EN 301 931-1.

### 13.1 Normal Procedures

#### 13.1.1 SCF-to/from-SRF messages

A dialogue is established when the SRF sends an AssistRequestInstructions operation to the SCF.

In the case that there is no pending operation and TCAP dialogue is established, TCAP dialogue can be terminated by TC-END primitive with zero component. When the SCSM makes a non-error case state transition to end user interaction and there is no operation to be sent, the dialogue is ended by means of a TC-END request primitive (basic) with zero components.

The dialogue shall no longer be maintained when sending the SRReport operation for PlayAnnouncement with disconnection from the SRF set to true or Return Result of the PromptAndCollectUserInformation with disconnection from the SRF set to true. The dialogue is ended by means of a TC-END request primitive with basic end, and the one of above operations is transmitted with the same request.

Regardless of whether pending operation exists or not, when the SRSM-FSM is informed of the disconnection of bearer connection (in the case of SCF initiated disconnection or call abandon from call party) and dialogue is established, the dialogue is ended by means of a TC-END request primitive (basic) with zero components or TC-END request primitive (prearranged end).

The dialogue shall no longer be maintained when the prearranged end condition is met in the SRF. When the SRSM-FSM is informed the disconnection of bearer connection and TCAP dialogue is not yet established (waiting for e.g. a TC\_CONTINUE), TCAP dialogue is locally terminated by TC-END primitive with prearranged end.

When the SCF does not expect any messages other than possibly REJECT or ERROR messages for the operations sent and when the last associated operation timer expires, the dialogue is locally ended by means of a TC-END request primitive with prearranged end. Alternatively, the sending of operations, leading to the termination of the relationship, by means of a TC-END request primitive (basic end) is possible.

In the relay case, the SRF-SCF relationship uses the SSF-SCF TCAP dialogue. This is possible, because begin and end of the SRF-SCF relationship are embedded in the SSF-SCF relationship. SRF-SCF information shall be exchanged with TC\_CONTINUE request primitives.

In the case of the SSF relay, it is outside the scope of this capability set how to map messages to ROSE capability of bearer signalling system between the SSF and the SRF, and what services are assumed from ROSE.

The dialog shall be maintained by the SCF when a ActivityTest operation is sent and by the SRF when a ActivitestTestResult is sent.

A dialog shall not be established by the SRF for sending a sRFCallGap operation, the SRF shall use an existing relationship and maintain the relationship after sending the operation. The SCF shall maintain the dialog when receiving a sRFCallGap operation.

### 13.2 Abnormal Procedures

#### 13.2.1 SCF-to-SRF messages

Considering that SRF, does not have the logic to recover from error cases detected on the SCF-SRF interface, the following shall apply:

- Operation errors and rejection of TCAP components shall be transmitted to the SRF with a TC-END request primitive, basic end.

If, in violation of the above procedure, an ERROR or REJECT component is received with a TC\_CONTINUE indication primitive, the SRF shall abort the dialogue with a TC-U-ABORT request primitive.

## 13.2.2 SRF/-to-SCF messages

Operation errors and rejection of TCAP components shall be transmitted to the SCF according to the following rules:

- The dialogue shall be maintained when the preceding message, which contained the erroneous component, indicated that the dialogue shall be maintained, i.e. the error or reject shall be transmitted with a TC-CONTINUE request primitive if the erroneous component was received with a TC-CONTINUE indication primitive.

On receipt of an ERROR or REJECT component the SCF decides on further processing. It may either continue, explicitly end or abort the dialogue.

- In all other situations the dialogue shall no longer be maintained, i.e. the error or reject shall be transmitted with a TC-END request primitive, basic end, if the erroneous component was received with a TC-BEGIN indication primitive.
- On expiration of application timer T<sub>SRF</sub> dialogue shall be terminated by means of by TC-U-ABORT primitive with an Abort reason, regardless of TCAP dialogue is established or not.

If the error processing in the SRF leads to the case where the SRF is not able to process further SCF operations while the dialogue is to be maintained, the SRF aborts the dialogue with a TC-END request primitive with basic end or a TC-U-ABORT request primitive, depending on whether any pending ERROR or REJECT component is to be sent or not.

## 13.3 Dialogue Handling

### 13.3.1 Dialogue Establishment

Common procedures as described in EN 301 931-1 are applicable.

### 13.3.2 Dialogue Continuation

Common procedures as described in EN 301 931-1 are applicable.

### 13.3.3 Dialogue Termination

Common procedures as described in EN 301 931-1 are applicable.

### 13.3.4 User Abort

Common procedures as described in EN 301 931-1 are applicable.

### 13.3.5 Provider Abort

Common procedures as described in EN 301 931-1 are applicable.

### 13.3.6 Mapping to TC Dialogue Primitives

The SCF-SRF IN services can be mapped onto TC services. This clause defines the mapping of the SCF-SRF IN services onto the services of the TC dialogue handling services defined in ETS 300 287.

The mapping of parameters onto the TC Dialogue services is as defined in EN 301 931-1 with the following qualifications.

The mapping of the parameters onto the TC-BEGIN primitive is defined in EN 301 931-1 with the following qualifications.

- The Application Context Name parameter shall take the value of the application-context-name field of the **srf-scf-ac** object.

## 13.3.7 Component Handling

### 13.3.7.1 Procedures for INAP Operations

Common procedures as described in EN 301 931-1 are applicable.

### 13.3.7.2 Mapping to TC Component Parameters

The mapping of parameters for the TC component services is defined in EN 301 931-1 with the following qualifications:

The Timeout Parameter of the TC-INVOKE service is set according to the requirements set out in the SCF-SRF interface clause.



---

## History

<b>Document history</b>			
V1.1.1	August 2000	Public Enquiry	PE 20001215: 2000-08-16 to 2000-12-15
V1.1.2	July 2001	Vote	V 20010831: 2001-07-02 to 2001-08-31