Vorbemerkungen

Die Richtlinie 163TR78 spezifiziert das im Netz der Telekom anzuwendende "Intelligent Network Application Protocol", im folgenden als TelekomINAP bezeichnet. Angewendet wird das TelekomINAP für folgende Schnittstellen:

- SCF und SSF
- SCF und SRF.

Das Protokoll zwischen SCF und SDF ist zur Zeit noch nicht spezifiziert und daher nicht Gegenstand dieser Richtlinie.

Teil 1 dieser Richtlinie beinhaltet die Spezifikation des TelekomINAP, die sich am ETSI-Standard ETS 300 374-1: "Core Intelligent Network Application Protocol (INAP), Part 1: Protocol Specification (1994)" - im folgenden als CoreINAP bezeichnet - orientiert. Dieses Kapitel ist daher im wesentlichen wie der CoreINAP strukturiert.

Teil 2 legt Besonderheiten zur Nutzung des SCCP fest.

Teil 3 beinhaltet die Festlegungen zu netzspezifischen Parametern und Werten.

Teil 1

1 Introduction

This document defines the INAP (Intelligent Network Application Protocol) required for support of Capability Set 1 used within the PSTN/ISDN of Deutsche Telekom and the PLMN of DeTeMobil. It supports interactions between the following functional entities (FE's), as defined in the IN Functional Model:

- *Service Switching Function (SSF)*

- *Service Control Function (SCF)*

2 Normative references

[1] For fixed network (Telekom):163 TR 72: Nachrichtentransferteil (MTP)
For mobile network (T-Mobil): The used MTP-Version shall be determined in separate negotiations.

[2] For fixed network (Telekom):163 TR 73: Steuerteil für Zeichengabeverbindungen (SCCP)
For mobile network (T-Mobil): The used SCCP-Version shall be determined in separate negotiations.

[3] ETS 300 121 (1992): "Integrated Services Digital Network (ISDN); Application of the ISDN User Part (ISUP) of CCITT Signalling System No.7 for international ISDN interconnections (ISUP version 1)".

[4] ETS 300 196-1 (1993): "Integrated Services Digital Network (ISDN); Generic functional protocol for the support of supplementary services; Digital Subscriber Signalling System No. one (DSS1) protocol; Part 1: Protocol specification".

NOTE: ETS 300 196-1 (1993) was initially published as ETS 300 196 (1993).

[5] For fixed network (Telekom):163 TR 74: Transaction Capabilities Application Part (TCAP)
For mobile network (T-Mobil): The used TCAP-Version shall be determined in separate negotiations.

[6] ETS 300 348: "Physical Plane for Intelligent Network (IN); Capability Set 1 (CS-1)".

[7] For fixed network (Telekom):163 TR 75 (ISUP) resp. 163 TR 80 (Interworkings).
For mobile network (T-Mobil): The used ISUP-Version shall be determined in separate negotiations.

[8] For fixed network (Telekom):1 TR 67: (DSS1)
For mobile network (T-Mobil): The used DSS1-Version shall be determined in separate negotiations.

[9] ITU-T Recommendation Q.700 (1993): "Introduction to CCITT Signalling System No.7".

[10] ITU-T Recommendation Q.773 (1993): "Specifications of Signalling System No.7; Transaction Capabilities formats and encoding".

[11] ITU-T Recommendation Q.1214 (1993): "Distributed functional plane for intelligent network CS1".

[12] ITU-T Recommendation Q.1218 (1993): "Interface Recommendation for intelligent network CS1".

[13] ITU-T Recommendation Q.1400 (1993): "Architecture framework for the development of signalling and organization, administration and maintenance protocols using OSI principles".

[14] CCITT Recommendation X.208 (1988): "Specification of Abstract Syntax Notation One (ASN.1)".

[15] CCITT Recommendation X.209 (1988): "Specification of basic encoding rules for Abstract Syntax Notation One (ASN.1)".

[16] CCITT Recommendation X.219 (1988): "Remote operations: Model, notation and service definition".

[17] CCITT Recommendation X.229 (1988): "Remote operations: Protocol specification".

[18] ISO 9545 (1989): "Information technology - Open Systems Interconnection - Application Layer structure".

[19] ETS 300 599: ”Mobile Application Part (MAP) specification (GSM 09.02)”.

[20] ETS 300 564: ”Mobile radio interface layer 3, Supplementary services specification, Formats and coding (GSM 04.80)”

3 Definitions and abbreviations

Refer to 163 TR 71.

4 General

4.1 Definition methodology

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

- the definition of the SACF/MACF Rules for the protocol (Section 1).
- the definition of the operations transferred between entities (Section 2).
- the definition of the actions taken at each entity (Section 3).

The SACF/MACF rules are defined in prose. The operation definitions are in Abstract Syntax Notation 1 (ASN.1, see Recommendations X.208), and the actions are defined in terms of state transition diagrams. Further guidance on the actions to be performed on receipt of an operation can be gained from section 2 and from the relevant detailed procedures in section 3 .

The INAP is a ROSE user protocol (see X.219/229). The ROSE protocol is contained within the Component Sublayer of TCAP (see Q.771-775) and DSS1 (Q.932). At present the ROSE APDU's (Application Protocol Data Units) are conveyed in Transaction sublayer messages in SS No. 7 and in the Q.931 , FACILITY and Call Control messages in DSS1. Other supporting protocols may be added at a later date.

Note that the TCAP Component Sub Layer has an additional APDU Return Result Not Last which is not present in ROSE. This may be used only if Blue Book (1988) SCCP is used to support INAP.

The INAP (as a ROSE user) and the ROSE protocol have been specified using ASN.1. At present, the only standardised way to encode the resulting PDU's is the Basic Encoding Rules (see X.209).

4.2 Example physical scenarios

The figure 4 depicted in this section shows how INAP will be supported in Telekom SS No. 7 network environment.

The SRF control procedures to be applied are shown in section 3.1.3.5.

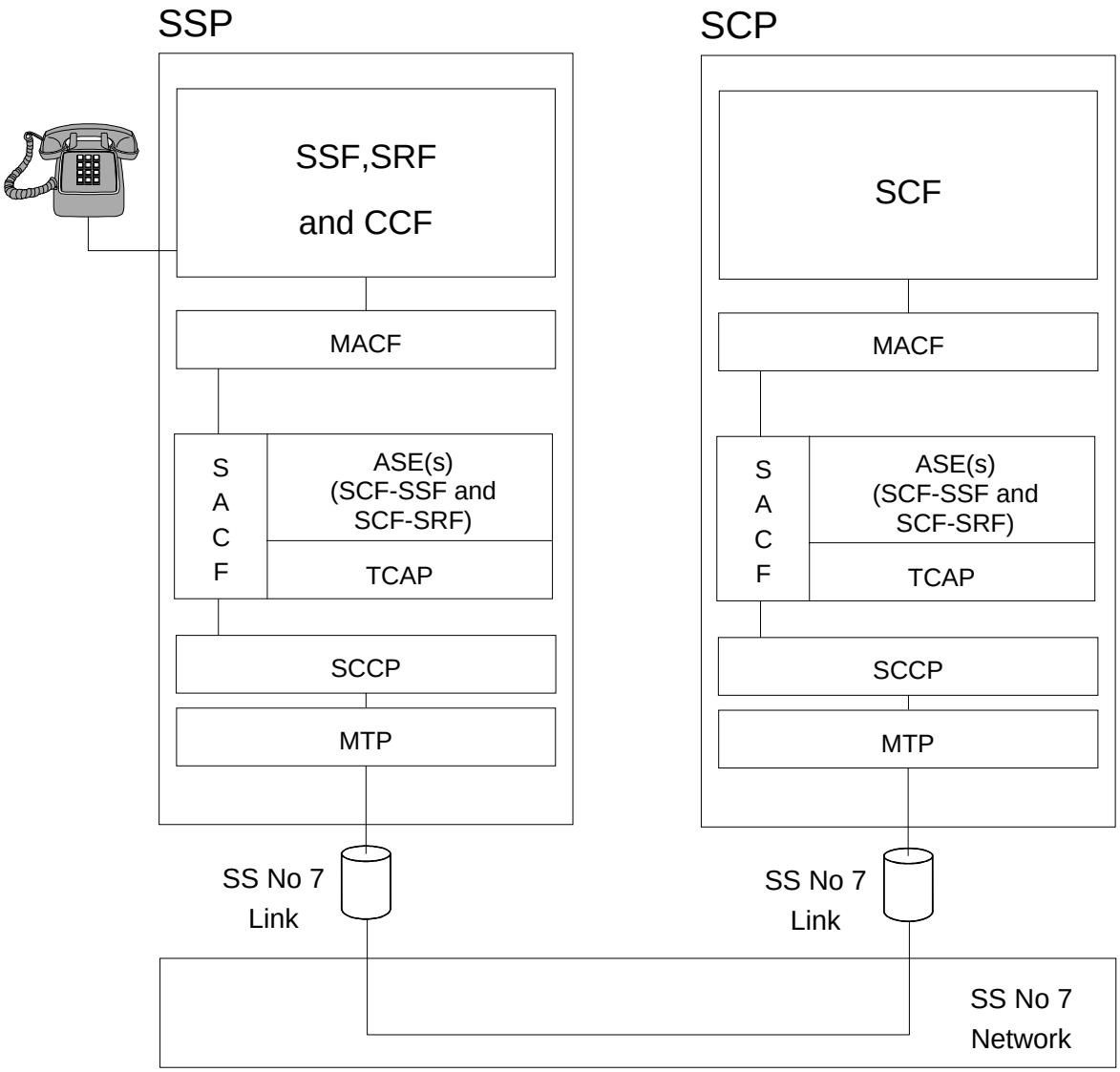


Figure 4: Example Architecture for supporting SRF,
case 3 (SRF in SSP and accessed via AP of SSP)

4.3 INAP protocol architecture

Many of the terms used in this section are based on the OSI Application Layer Structure as defined in ISO IS-9545.

The INAP protocol architecture can be illustrated as shown in figure 7.

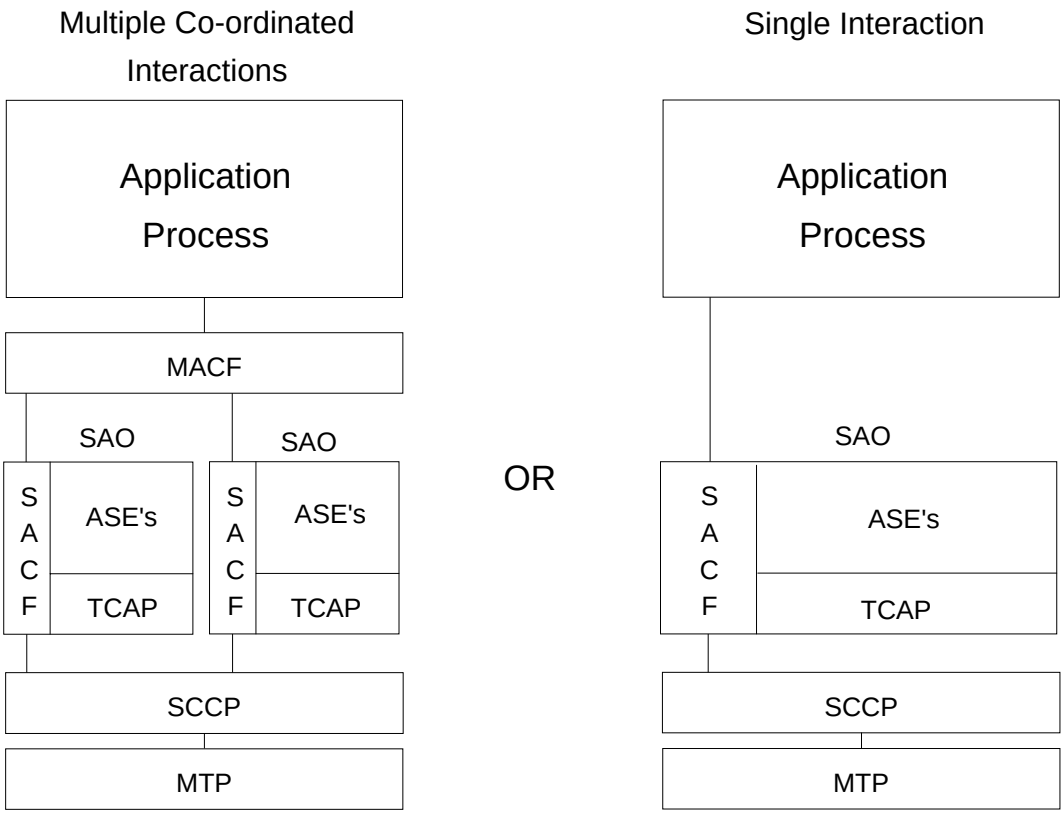


Figure 7: INAP protocol architecture

Legend to figure 7:

SACF: Single Association Control Function

MACF: Multiple Association Control Function

SAO: Single Association Object

ASE: Application Service Element

INAP: Intelligent Network Application Protocol (Note: INAP is the collection of specifications of all IN ASE's).

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

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

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

Each ASE supports one or more operations. Description of each operation is tied with the action of corresponding FE modelling (see Q.1214 and Section 3/Q.1218). Each operation is specified using the operation macro described in figure 8.

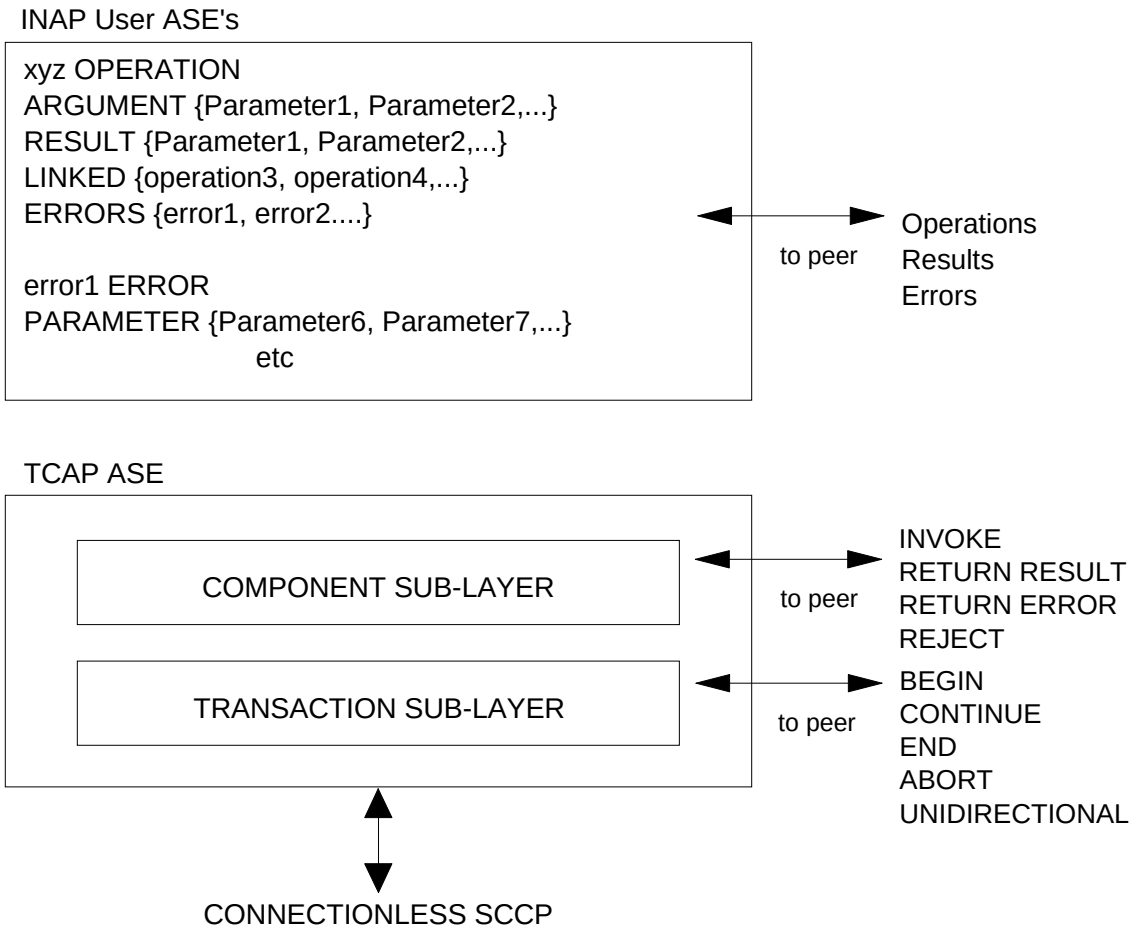


Figure 8: Operation Description

The use of the Application Context negotiation mechanism (as defined in the Q.77x series Recommendations) allows the two communicating entities to identify exactly what their capabilities are and also what the capabilities required on the interface should be. This should be used to allow evolution through capability sets.

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

4.3.1 INAP signalling congestion control for Signalling System No.7

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

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

4.4 INAP addressing

See Kapitel 2 "Use of SCCP".

4.5 Relationship between Q.1214 and Q.1218

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

Reference	Q.1214 Information Flow	Operation
6.4.2.1	Activate Service Filtering	Same
6.4.2.2	Activity Test	Same
6.4.2.3	Activity Test Response	Return Result from ActivityTest
6.4.2.4	Analyzed Information	InitialDP
6.4.2.5	Analyze Information	*not applicable*
6.4.2.6	Apply Charging	Same
6.4.2.7	Apply Charging Report	Same
6.4.2.8	Assist Request Instructions	Same
6.4.2.9	Call Gap	Same
6.4.2.10	Call Information Report	Same
6.4.2.11	Call Information Request	Same
6.4.2.14	Collected Information	InitialDP, EventReportBCSM
6.4.2.15	Collect Information	Same
6.4.2.16	Connect	Same
6.4.2.17	Connect to Resource	Same
6.4.2.18	Continue	Same
6.4.3.19	Disconnect Forward Connection	Same
6.4.2.20	Establish Temporary Connection	Same
6.4.2.21	Event Notification Charging	*not applicable*
6.4.2.22	Event Report BCSM	Same
6.4.2.23	Furnish Charging Information	Same
6.4.2.24	Hold call in network	*not applicable*
6.4.2.25	Initial DP	Same
6.4.2.26	Initiate Call Attempt	*not applicable*
6.4.2.27	OAnswer	EventReportBCSM
6.4.2.28	OCalledPartyBusy	EventReportBCSM
6.4.2.29	ODisconnect	EventReportBCSM
6.4.2.30	O_MidCall	*not applicable*
6.4.2.31	O_No_Answer	EventReportBCSM
6.4.2.32	Origination Attempt Authorized	InitialDP
6.4.2.33	Release Call	Same
6.4.2.34	Request Notification Charging Event	*not applicable*
6.4.2.35	Request Report BCSM Event	Same
6.4.2.37	Reset Timer	Same
6.4.2.38	Route Select Failure	InitialDP, EventReportBCSM
6.4.2.40	Select Route	*not applicable*
6.4.2.41	Send Charging Information	Same
6.4.2.42	Service Filtering Response	Same
6.4.2.44	TAnswer	*EventReportBCSM*
6.4.2.45	TCalled Party Busy	*InitialDP, EventReportBCSM*
6.4.2.46	TDisconnect	*EventReportBCSM*
6.4.2.47	Term Attempt Authorized	*InitialDP*

6.4.2.48	T_MidCall	*not applicable*
6.4.2.49	TNoAnswer	*InitialDP, EventReportBCSM*
6.5.2.1	AssistRequestInstructions from SRF	*not applicable*
6.5.2.2	Cancel Announcement	*not applicable*
6.5.2.3	CollectedUserInformation	Return Result from PromptAndCollectUserInformation
6.5.2.4	Play Announcement	Same
6.5.2.5	Prompt and collect user information	Same
6.5.2.6	Specialized Resource Report	Same

4.6 Compatibility Mechanisms used for INAP

4.6.1 Introduction

This section describes the compatibility mechanisms that shall be used for the INAP according to 163 TR 78 also refered as TelekomINAP:

- Minor changes to the ETSI INAP in future standardised versions.
  A minor change can be defined as a change of a functionality which is not essential for the requested IN service. In case it is a modification of an existing function, it is acceptable that the addressed function is executed in either the older or the modified variant. If the change is purely additional, it is acceptable that it is not executed at all and that the peer application entity need not know about the effects of the change.
- Major changes to the ETSI INAP in future standardised versions.
  A major change can be defined as a change of a functionality which is essential for the requested IN service. In case it is a modification of an existing function, both application entities must have a shared knowledge about the adressed functional variant. If the change is purely additional, the requested IN service will not be provided if one of the application entities does not support the additional functionality.

  NOTE: The argument ExtensionParameter, which is defined for operations within the ETSI CoreINAP will not be used.

4.6.2 Definition of ETSI INAP Compatibility Mechanisms

4.6.2.1 Compatibility mechanism for interworking *163TR78 and* ETSI INAP / CCITT Q.1218 INAP

On receipt of a Q.1218 *or ETSI-INAP* operation which is not part of the *163TR78* or is part of the *163TR78* but which contains parameters which are not part of the *163TR78* ,

- The SSF shall apply the normal error handling for unknown operations or parameters, i.e.the normal error handling procedures as specified in section 3.4 shall be followed.
- The SCF shall apply the normal error handling for unknown operations or parameters except for parameters in the InitialDP operation. All parameters specified in Q.1218 *or ETSI-INAP* for InitialDP shall be known by the SCF, those not included in the *163TR78* shall be ignored.

4.6.2.2 Procedures for Major additions to INAP *according to 163 TR 78*

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

4.6.2.3 Procedures for Minor additions to INAP *according to 163 TR 78*

The extension mechanism marker shall be used for future standardised minor additions to INAP. This mechanism implements extensions differently by including an 'extensions marker' in the type definition. The extensions are expressed by optional fields that are placed after the marker. When an entity receives an unrecognised parameter that occurs after the marker, they are ignored (see ISO/IEC PDAM 8824-1).

Because this version of ASN.1 has not yet been ratified by ETSI, the marker is placed within a comment for now so that it can easily be uncommented when the '92 version of ASN.1 is ratified by ETSI. As in the ISO case, the extensions are expressed by optional fields that are placed after the (commented out) marker, in which case they can be ignored.

5. SACF/MACF rules

5.1 Reflection of TCAP AC

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

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

TCAP Application Context negotiation applies only to the SCF interfaces.

Refer to Q.77x for a more detailed description of the TCAP AC negotiation mechanism.

5.2 Sequential/Parallel execution of operations

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

- charging operations may be synchronised with any other operation.

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

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

In case of inconsistency between the above mentioned generic rules and the functional entity specific rules, as specified in section 3, the functional entity specific rules take precedence over the generic rules.

6. Abstract Syntax of the IN CS 1 Application Protocol

This section specifies the abstract syntax for the IN CS 1 Application Protocol using Abstract Syntax Notation One (ASN.1), defined in Recommendation CCITT X.208.

The encoding rules which are applicable to the defined abstract syntax are the Basic Encoding Rules for ASN.1, defined in Recommendation CCITT X.209 with the restrictions as described in Section 4.1.1/Q.773. Additional encodings are cited for parameters used in existing ISUP (Q.763) and DSS1 (Q.931) and MAP (GSM 09.02) Recommendations.
For the ISUP and DSS1 and MAP parameters used in the INAP, only the coding of the parameter value will be coded as defined in ISUP or DSS1 or MAP. The DSS1/ISUP/MAP defined parameter identifiers are removed and replaced by the INAP defined parameter identifiers.

The mapping of OPERATION and ERROR to TCAP components is defined in Recommendation CCITT Q.773. The class of an operation is not stated explicitly but is specified in the ASN.1 OPERATION MACRO, as follows:

Class 1: both RESULT and ERRORS appear in the ASN.1 OPERATION MACRO definition,

Class 2: only ERRORS appears in the ASN.1 OPERATION MACRO definition,

Class 3: only RESULT appears in the ASN.1 OPERATION MACRO definition,

Class 4: neither RESULT nor ERRORS appears in the ASN.1 OPERATION MACRO definition.

These map to the classes 2 through 5, respectively, specified in Recommendations CCITT X.219 and Q.932.

The abstract syntax for INAP is composed of several ASN.1 modules describing operations, errors, and associated data types. The values (operation codes and error codes) are defined in a separate module.

The module containing all the type definitions for INAP operations is **IN-CS1-Operations** and is described in Section *6.1*.

The module containing all the type definitions for INAP errors is **IN-CS1-Errors** and is described in Section *6.2*.

The module containing all the type definitions for INAP data types is **IN-CS1-DataTypes** and is described in Section *6.3*.

The module containing the operation codes and error codes for INAP is **IN-CS1-Codes** and is described in Section *6.4*.

All the application context definitions for *Telekom* INAP are described in Section *6.5*.

6.1 IN CS1 Operation Types

Telekom-INAP-CS1-Operations {ccitt(0) *administration(2) bmpt(262) telekom(1) zgs_nr7(5) inap(0)* modules(0) cs1-operations(0) version5(4)}

-- This module contains the type definitions for the IN CS1 operations.

```
DEFINITIONS IMPLICIT TAGS ::=
BEGIN
IMPORTS
        OPERATION
FROM TCAPMessages {ccitt recommendation q 773 modules(2) messages(1) version2(2)}
-- error types
        ETCFailed,
        ImproperCallerResponse,
        MissingCustomerRecord,
        MissingParameter,
        ParameterOutOfRange,
        SystemFailure,
        TaskRefused,
        UnavailableResource,
        UnexpectedComponentSequence,
        UnexpectedDataValue,
        UnexpectedParameter
```

```
FROM Telekom-INAP-CS1-Errors {ccitt(0) administration(2) bmpt(262) telekom(1) zgs_nr7(5) inap(0)
modules(0) cs1-errors(1) version3(2)}
-- argument types
        ActivateServiceFilteringArg,
        ApplyChargingArg,
        ApplyChargingReportArg,
        AssistRequestInstructionsArg,
        CallGapArg,
        CallInformationReportArg,
        CallInformationRequestArg,
        CollectInformationArg,
        ConnectArg,
        ConnectToResourceArg,
        EstablishTemporaryConnectionArg,
        EventReportBCSMArg,
        FurnishChargingInformationArg,
        InitialDPArg,
        PlayAnnouncementArg,
        PromptAndCollectUserInformationArg,
        ReceivedInformationArg,
        ReleaseCallArg,
        RequestReportBCSMEventArg,
        ResetTimerArg,
        SendChargingInformationArg,
        ServiceFilteringResponseArg,
        SpecializedResourceReportArg

FROM Telekom-INAP-CS1-DataTypes {ccitt(0) administration(2) bmpt(262) telekom(1) zgs_nr7(5)
inap(0) modules(0) cs1-datatypes(2) version5(4)};

-- TYPE DEFINITIONS FOR IN CS1 OPERATIONS FOLLOWS

-- SCF - SSF operations

ActivateServiceFiltering ::= OPERATION
        ARGUMENT
                ActivateServiceFilteringArg
        RESULT
        ERRORS {
                MissingParameter,
                ParameterOutOfRange,
                SystemFailure,
                TaskRefused,
                UnexpectedComponentSequence,
                UnexpectedParameter
                }

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

ActivityTest ::= OPERATION
        RESULT
-- Direction: SCF -> SSF, Timer: T_at
-- This operation is used to check for the continued existence of a relationship between the SCF
-- and SSF. If the relationship is still in existence, then the SSF will respond. If no reply is received,
```

```
-- then the SCF will assume that the SSF has failed in some way and will take the appropriate
-- action.

ApplyCharging           ::= OPERATION
        ARGUMENT
                ApplyChargingArg
        ERRORS {
                MissingParameter,
                UnexpectedComponentSequence,
                UnexpectedParameter,
                UnexpectedDataValue,
                ParameterOutOfRange,
                SystemFailure,
                TaskRefused
                }

-- Direction: SCF -> SSF, Timer: Tac
-- This operation is used for interacting from the SCF with the SSF charging mechanisms. The
-- ApplyChargingReport operation provides the feedback from the SSF to the SCF.

ApplyChargingReport                     ::= OPERATION
        ARGUMENT
                ApplyChargingReportArg
        ERRORS {
                MissingParameter,
                UnexpectedComponentSequence,
                UnexpectedParameter,
                UnexpectedDataValue,
                ParameterOutOfRange,
                SystemFailure,
                TaskRefused
                }

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


AssistRequestInstructions     ::= OPERATION
      ARGUMENT
            AssistRequestInstructionsArg
      ERRORS {
            MissingCustomerRecord,
            MissingParameter,
            TaskRefused,
            UnexpectedComponentSequence,
            UnexpectedDataValue,
            UnexpectedParameter
            }

-- Direction: SSF -> SCF, Timer: Tari
-- This operation is used when there is an assist procedure and may be sent by
the
-- SSF  to the SCF.  This operation is sent by the assisting SSF  to the SCF,
-- when the initiating SSF has set up a connection to the assisting SSF
-- as a result of receiving an  EstablishTemporaryConnection operation from
the SCF.

CallGap                         ::= OPERATION
        ARGUMENT
```

```
                CallGapArg

--  Direction: SCF -> SSF, Timer: Tcg
--  This operation is used to request the SSF to reduce the rate at which specific service requests
--  are sent to the SSF.

CallInformationReport           ::= OPERATION
        ARGUMENT
                CallInformationReportArg

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

CallInformationRequest          ::= OPERATION
        ARGUMENT
                CallInformationRequestArg
        ERRORS {
                MissingParameter,
                ParameterOutOfRange,
                SystemFailure,
                TaskRefused,
                UnexpectedComponentSequence,
                UnexpectedParameter
                }

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



CollectInformation      ::= OPERATION
        ARGUMENT
                CollectInformationArg
        ERRORS {
                SystemFailure,
                TaskRefused,
                UnexpectedComponentSequence
                }

--  Direction: SCF -> SSF, Timer: Tci
--  This operation is used to request the SSF to perform the originating basic call processing
--  actions to collect destination information
--  according to a specified numbering plan (e.g., for virtual private networks).

Connect                 ::= OPERATION
        ARGUMENT
                ConnectArg
        ERRORS {
                MissingParameter,
                SystemFailure,
                TaskRefused,
                UnexpectedComponentSequence,
                UnexpectedDataValue,
                UnexpectedParameter
                }
```

```
-- Direction: SCF -> SSF, Timer: Tcon
-- This operation is used to request the SSF to perform the call processing actions to route or
-- forward a call to a specified destination.  To do so, the SSF may or may not use destination
-- information from the calling party (e.g., dialed digits) and existing call setup information,
--  depending on the information provided by the SCF.

ConnectToResource      ::= OPERATION
        ARGUMENT
                ConnectToResourceArg
        ERRORS {
                MissingParameter,
                SystemFailure,
                TaskRefused,
                UnexpectedComponentSequence,
                UnexpectedDataValue,
                UnexpectedParameter
                }

-- Direction: SCF -> SSF, Timer: Tctr
-- This operation is used to connect a call from the SSP to the physical entity containing the SRF.

Continue                                        ::= OPERATION

-- Direction: SCF -> SSF, Timer: Tcue
-- This operation is used to request the SSF to proceed with call processing at the DP at which it
-- previously suspended call processing to await SCF instructions (i.e., proceed to the next point
-- in call in the BCSM). The SSF continues call processing without substituting new data from SCF.


DisconnectForwardConnection   ::= OPERATION
        ERRORS {
                SystemFailure,
                TaskRefused,
                UnexpectedComponentSequence
                }

-- Direction: SCF -> SSF, Timer: Tdfc
-- This operation is used to disconnect a forward temporary connection  or a connection to a
-- resource.

EstablishTemporaryConnection  ::= OPERATION
       ARGUMENT
              EstablishTemporaryConnectionArg
       ERRORS {
              ETCFailed,
              MissingParameter,
              SystemFailure,
              TaskRefused,
              UnexpectedComponentSequence,
              UnexpectedDataValue,
              UnexpectedParameter
              }

-- Direction: SCF -> SSF, Timer: Tetc
-- This operation is used to create a connection to a resource for a limited
period of time (e.g. to
-- play an announcement, to collect user information); it implies the use of
the assist procedure.
```

```
EventReportBCSM        ::= OPERATION
        ARGUMENT
                EventReportBCSMArg

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

FurnishChargingInformation      ::= OPERATION
        ARGUMENT
                FurnishChargingInformationArg
        ERRORS {
                MissingParameter,
                TaskRefused,
                UnexpectedComponentSequence,
                UnexpectedDataValue,
                UnexpectedParameter
                }

-- Direction: SCF -> SSF, Timer: Tfci
-- This operation is used to request the SSF to generate, register a call record.
--  The registered call record is intended for off line charging
-- of the call.

InitialDP          ::= OPERATION
        ARGUMENT
                InitialDPArg
        ERRORS {
                MissingCustomerRecord,
                MissingParameter,
                SystemFailure,
                TaskRefused,
                UnexpectedComponentSequence,
                UnexpectedDataValue,
                UnexpectedParameter
                }

-- Direction: SSF -> SCF, Timer: Tidp
-- This operation is used after a TDP to indicate request for service.


ReleaseCall                ::= OPERATION
        ARGUMENT
                ReleaseCallArg

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

```
RequestReportBCSMEvent      ::= OPERATION
        ARGUMENT
                RequestReportBCSMEventArg
        ERRORS {
                MissingParameter,
                SystemFailure,
                TaskRefused,
                UnexpectedComponentSequence,
                UnexpectedDataValue,
                UnexpectedParameter
                }

-- Direction: SCF -> SSF, Timer: Trrb
-- This operation is used to request the SSF to monitor for a call-related event (e.g., BCSM events
-- such as busy or no answer), then send a notification back to the SCF when the event is
-- detected.

ResetTimer          ::= OPERATION
        ARGUMENT
                ResetTimerArg
        ERRORS {
                MissingParameter,
                TaskRefused,
                UnexpectedComponentSequence,
                UnexpectedDataValue,
                UnexpectedParameter
                }

-- Direction: SCF -> SSF, Timer: Trt
-- This operation is used to request the SSF to refresh an application timer
in the SSF.


SendChargingInformation             ::= OPERATION
        ARGUMENT
                SendChargingInformationArg
        ERRORS {
                MissingParameter,
                UnexpectedComponentSequence,
                UnexpectedParameter,
                ParameterOutOfRange,
                SystemFailure,
                TaskRefused
                }

-- Direction: SCF -> SSF, Timer: Tsci
-- This operation is used to  instruct the
-- SSF on the charging information to be sent by the SSF.
-- The charging information can either be sent back by means of signalling or internal if the SSF is
-- located in the local exchange. In the local exchange this information may be used to update the
-- charge meter or to create a standard call record.
-- The charging scenario supported by this operation is: 3.2 (refer to Annex B where these are
-- defined).


ServiceFilteringResponse        ::= OPERATION
        ARGUMENT
                ServiceFilteringResponseArg
```

-- Direction: SSF -> SCF, Timer: T_{sfr}
-- This operation is used to send back to the SCF the values of counters specified in a previous
-- ActivateServiceFiltering operation.

-- SCF - SRF operations

```
--AssistRequestInstructions
--SRF -> SCF
--Refer to previous description of this operation in the SCF - SSF operations
section.
```

```
PlayAnnouncement        ::= OPERATION
        ARGUMENT
                PlayAnnouncementArg
        ERRORS {
                MissingParameter,
                SystemFailure,
                UnavailableResource,
                UnexpectedComponentSequence,
                UnexpectedDataValue,
                UnexpectedParameter
                }
        LINKED {
                SpecializedResourceReport
                }
```

-- Direction: SCF -> SRF, Timer: T_{pa}
-- This operation is to be used after Establish Temporary Connection (assist procedure with a
-- second SSP) or a Connect to Resource operation. It may be used for inband
-- interaction with an analog user, or for interaction with an ISDN user.
-- 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.

```
PromptAndCollectUserInformation         ::= OPERATION
        ARGUMENT
                PromptAndCollectUserInformationArg
        RESULT
                ReceivedInformationArg
        ERRORS {
                ImproperCallerResponse,
                MissingParameter,
                SystemFailure,
                TaskRefused,
                UnavailableResource,
                UnexpectedComponentSequence,
                UnexpectedDataValue,
                UnexpectedParameter
                }
```

-- Direction: SCF -> SRF, Timer: T_{pc}
-- This operation is used to interact with a user to collect information.

```
SpecializedResourceReport       ::= OPERATION
```

```
        ARGUMENT
                SpecializedResourceReportArg

-- Direction: SRF -> SCF, Timer: Tsrr
-- This operation is used as
-- the response to a PlayAnnouncement operation when the announcement completed report
-- indication is set.

END
```

Operation Timers

The following value ranges do apply for operation specific timers in INAP:
short: *10* seconds
medium: *30* seconds
long: *2* minutes
The table below lists all operation timers and the value range for each timer.

Operation Name	Timer	value range
ActivateServiceFiltering	T_{asf}	short
ActivityTest	T_{at}	short
ApplyCharging	T_{ac}	short
ApplyChargingReport	T_{acr}	short
AssistRequestInstructions	T_{ari}	short
CallGap	T_{cg}	short
CallInformationReport	T_{cirp}	short
CallInformationRequest	T_{cirq}	short
CollectInformation	T_{ci}	medium
Connect	T_{con}	short
ConnectToResource	T_{ctr}	short
Continue	T_{cue}	short
DisconnectForwardConnection	T_{dfc}	short
EstablishTemporaryConnection	Tetc	medium
EventNotificationCharging	T_{enc}	short
EventReportBCSM	T_{erb}	short
FurnishChargingInformation	T_{fci}	short
InitialDP	T_{idp}	short
ReleaseCall	T_{rc}	short
RequestNotificationChargingEvent	T_{rnc}	short
RequestReportBCSMEvent	T_{rrb}	short
ResetTimer	T_{rt}	short
SendChargingInformation	T_{sci}	short
ServiceFilteringResponse	T_{sfr}	short
PlayAnnouncement	T_{pa}	long
PromptAndCollectUserInformation	T_{pc}	long
SpecializedResourceReport	Tsrr	short

6.2 IN CS1 Error Types

Telekom-INAP-CS1-Errors { ccitt(0) *administration(2) bmpt(262) telekom(1) zgs_nr7(5) inap(0)* modules(0) cs1-errors(1) version3(2)}
-- This module contains the type definitions for the IN CS1 errors.

DEFINITIONS IMPLICIT TAGS ::=
BEGIN

IMPORTS
ERROR
FROM TCAPMessages {ccitt recommendation q 773 modules(2) messages(1) version2(2)}

UnavailableNetworkResource
FROM *Telekom*-INAP-CS1-DataTypes {ccitt(0) *administration(2) bmpt(262) telekom(1) zgs_nr7(5) inap(0)* modules(0) cs1-datatypes(2) version5(4)};

-- TYPE DEFINITION FOR **IN CS1** ERRORS FOLLOWS

ETCFailed ::= ERROR

-- The establish temporary connection failed.

ImproperCallerResponse ::= ERROR

-- The caller response was not as expected.

MissingCustomerRecord ::= ERROR

-- The Service Logic Program could not be found in the SCF.

MissingParameter ::= ERROR

-- An expected optional parameter was not received.

ParameterOutOfRange ::= ERROR

-- The parameter was not as expected (e.g., missing or out of range).

SystemFailure ::= ERROR
PARAMETER
UnavailableNetworkResource

-- The operation could not be completed due to a system failure at the serving physical entity.

TaskRefused ::= ERROR
PARAMETER ENUMERATED {
generic(0),
unobtainable (1),
congestion(2)

}

-- An entity normally capable of the task requested cannot or chooses not to perform the task at
-- this time. (This includes error situations like congestion and unobtainable address as used in
-- e.g., the connect operation).

UnavailableResource ::= ERROR

-- A requested resource is not available at the serving entity.

UnexpectedComponentSequence ::= ERROR

```
-- An incorrect sequence of Components was received (e.g., "DisconnectForwardConnection"
-- followed by "PlayAnnouncement").

UnexpectedDataValue             ::= ERROR

-- The data value was not as expected
UnexpectedParameter             ::= ERROR

-- A parameter received was not expected.



END
```

6.3 IN CS1 Data Types

Telekom-INAP-CS1-DataTypes {ccitt(0) *administration(2) bmpt(262) telekom(1) zgs_nr7(5) inap(0)* modules(0) cs1-datatypes(2) version5(4)}

```
-- This module contains the type definitions for the IN CS1 data types.
-- The following parameters map onto bearer protocol (i.e [8] (DSS1), [7] (ISUP)
-- and [19] (MAP)) parameters:
-- CalledPartyNumber,
-- BearerCapability, CallingPartyNumber,
-- HighLayerCompatibility,
-- DestinationRoutingAddress, OriginalCalledPartyID, RedirectingPartyID, RedirectionInformation,
-- CallingPartysCategory, LocationNumber,
-- ReleaseCause (and other Cause parameters), AdditionalCallingPartyNumber,
-- natCallingPartysCategory, locationNumberB, userDialogInfo.

-- The following SSF parameters do not map onto bearer protocol (i.e., [8] (DSS1) and [7] (ISUP))
-- parameters and therefore are assumed to be local to the switching system:

-- LegID, MiscCallInfo, and ServiceKey.


DEFINITIONS IMPLICIT TAGS ::=

BEGIN



-- TYPE DEFINITIONS FOR   IN CS1   DATA TYPES FOLLOWS

-- Argument Data Types

ActivateServiceFilteringArg             ::= SEQUENCE {
        filteredCallTreatment           [0] FilteredCallTreatment,
        filteringCharacteristics        [1] FilteringCharacteristics,
        filteringTimeOut                [2] FilteringTimeOut,
        filteringCriteria               [3] FilteringCriteria,
        startTime                       [4] DateAndTime         OPTIONAL
--      ...
        }

ApplyChargingArg                        ::= SEQUENCE {
```

```
        aChBillingChargingCharacteristics   [0] AChBillingChargingCharacteristics,
        partyToCharge                       [2] LegID                   OPTIONAL
--  ...
        }

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

ApplyChargingReportArg                      ::= CallResult


AssistRequestInstructionsArg    ::= SEQUENCE {
        correlationID               [0] CorrelationID
--      ...
        }
-- The correlationID contains   the same value received in ETC from the SCF.

CallGapArg                                  ::= SEQUENCE {
        gapCriteria                         [0] GapCriteria,
        gapIndicators                       [1] GapIndicators,
        gapTreatment                        [3] GapTreatment        OPTIONAL
--      ...
        }

--  If gap treatment is not present, then all gapped calls will be released using
--  the default cause value #42.

CallInformationReportArg                    ::= SEQUENCE {
        requestedInformationList            [0] RequestedInformationList
--      ...
        }


CallInformationRequestArg                   ::= SEQUENCE {
        requestedInformationTypeList        [0] RequestedInformationTypeList
--      ...
        }




CollectInformationArg                       ::= SEQUENCE {
--      ...
        }

ConnectArg                                      ::= SEQUENCE {
        destinationRoutingAddress           [0] DestinationRoutingAddress,
        cutAndPaste                         [3] CutAndPaste                         OPTIONAL,
        originalCalledPartyID               [6] OriginalCalledPartyID               OPTIONAL,

        callingPartyNumber                  [27] CallingPartyNumber                 OPTIONAL,
        callingPartysCategory               [28] CallingPartysCategory              OPTIONAL,
        redirectingPartyID                  [29] RedirectingPartyID                 OPTIONAL,
        redirectionInformation              [30] RedirectionInformation             OPTIONAL,
        serviceInteractionIndicatorsTwo     [15] ServiceInteractionIndicatorsTwo    OPTIONAL,
        natServiceInteractionIndicators     [PRIVATE 1] NatServiceInteractionIndicators  OPTIONAL,
        natCallingPartysCategory            [PRIVATE 5] NatCallingPartysCategory    OPTIONAL,
        iNContainer                         [PRIVATE 7] INContainer                 OPTIONAL,
        userUser                            [PRIVATE 8] UserUser                    OPTIONAL
```

```
--      ...
        }

ConnectToResourceArg                ::= SEQUENCE {
        resourceAddress             CHOICE {
                ipRoutingAddress    [0] IPRoutingAddress,
                none                [3] NULL
                },
        serviceInteractionIndicatorsTwo  [7] ServiceInteractionIndicatorsTwo        OPTIONAL,
        natServiceInteractionIndicators  [PRIVATE 1] NatServiceInteractionIndicators  OPTIONAL
--      ...
        }

EstablishTemporaryConnectionArg ::= SEQUENCE {
        assistingSSPIPRoutingAddress  [0] AssistingSSPIPRoutingAddress,
        correlationID                 [1] CorrelationID,
        scfID                         [3] scfID
        serviceInteractionIndicatorsTwo  [6] ServiceInteractionIndicatorsTwo        OPTIONAL
--      ...
        }

EventReportBCSMArg                  ::= SEQUENCE {
        eventTypeBCSM               [0] EventTypeBCSM,
        eventSpecificInformationBCSM [2] EventSpecificInformationBCSM  OPTIONAL,
        legID                       [3] LegID                          OPTIONAL,
        miscCallInfo                [4] MiscCallInfo  DEFAULT  {messageType request}
--      ...
        }

FurnishChargingInformationArg       ::= FCIBillingChargingCharacteristics


InitialDPArg                        ::= SEQUENCE {
        serviceKey                  [0] ServiceKey,
--      dialledDigits               [1] CalledPartyNumber                 OPTIONAL,
        calledPartyNumber           [2] CalledPartyNumber                 OPTIONAL,
        callingPartyNumber          [3] CallingPartyNumber                OPTIONAL,
--      callingPartyBusinessGroupID [4] CallingPartyBusinessGroupID       OPTIONAL,
        callingPartysCategory       [5] CallingPartysCategory             OPTIONAL,
--      callingPartySubaddress      [6] CallingPartySubaddress            OPTIONAL,
--      cGEncountered               [7] CGEncountered                     OPTIONAL,
--      iPSSPCapabilities           [8] IPSSPCapabilities                 OPTIONAL,
--      iPAvailable                 [9] IPAvailable                       OPTIONAL,
        locationNumber              [10] LocationNumber                   OPTIONAL,
--      miscCallInfo                [11] MiscCallInfo                     OPTIONAL,
        originalCalledPartyID       [12] OriginalCalledPartyID            OPTIONAL,
--      serviceProfileIdentifier    [13] ServiceProfileIdentifier         OPTIONAL,
--      terminalType                [14] TerminalType                     OPTIONAL,
--      extensions                  [15] SEQUENCE SIZE(1..numOfExtensions) OF
                                                 ExtensionField         OPTIONAL,
        highLayerCompatibility      [23] HighLayerCompatibility           OPTIONAL,
--      serviceInteractionIndicators [24] ServiceInteractionIndicators
                                                                          OPTIONAL,
        additionalCallingPartyNumber [25] AdditionalCallingPartyNumber    OPTIONAL,
--      forwardCallIndicators       [26] ForwardCallIndicators            OPTIONAL,
        bearerCapability            [27] BearerCapability                 OPTIONAL,
        eventTypeBCSM               [28] EventTypeBCSM                    OPTIONAL,
        redirectingPartyID          [29] RedirectingPartyID               OPTIONAL,
        redirectionInformation      [30] RedirectionInformation           OPTIONAL,
        sFEncountered               [PRIVATE 1] BOOLEAN                   DEFAULT FALSE,
```

```
        gapInterval                 [PRIVATE 2] Interval                DEFAULT 0,
        locationNumberB             [PRIVATE 3] LocationNumberB         OPTIONAL,
        userDialogInfo              [PRIVATE 4] UserDialogInfo          OPTIONAL,
        iMSI                        [PRIVATE 5] IMSI                    OPTIONAL,
        natCallingPartysCategory    [PRIVATE 6] NatCallingPartysCategory OPTIONAL,
        iNContainer                 [PRIVATE 7] INContainer             OPTIONAL
--      ...
        }

-- OPTIONAL for  callingPartyNumber, and callingPartysCategory refer to 9.19.2.1 for the
-- trigger detection point processing rules to specify
-- when these parameters are included in the message.
-- The  parameters included as comments shall be recognised by the SCF
-- upon reception of InitialDP. These
-- parameters shall be ignored by the SCF and not lead to any error procedures.
-- These parameters shall not be sent by a SSF following this standard.


PlayAnnouncementArg                 ::= SEQUENCE {
        informationToSend           [0] InformationToSend,
        disconnectFromIPForbidden   [1] BOOLEAN                 DEFAULT TRUE,
        requestAnnouncementComplete [2] BOOLEAN                 DEFAULT TRUE
--      ...
        }

PromptAndCollectUserInformationArg::= SEQUENCE {
        collectedInfo               [0] CollectedInfo,
        disconnectFromIPForbidden   [1] BOOLEAN                 DEFAULT TRUE,
        informationToSend           [2] InformationToSend       OPTIONAL
--      ...
        }

ReceivedInformationArg              ::= CHOICE {
        digitsResponse              [0]  Digits
        }

ReleaseCallArg                      ::= Cause

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

RequestReportBCSMEventArg           ::= SEQUENCE {
        bcsmEvents                  [0] SEQUENCE SIZE (1..numOfBCSMEvents) OF
                                            BCSMEvent
--      ...
        }

-- Indicates the BCSM related events for notification.


ResetTimerArg                       ::= SEQUENCE {
        timerID                     [0] TimerID                 DEFAULT tssf,
        timervalue                  [1] TimerValue
--      ...
        }


SendChargingInformationArg          ::= SEQUENCE {
        sCIBillingChargingCharacteristics [0] SCIBillingChargingCharacteristics
```

```
--      ...
        }

ServiceFilteringResponseArg             ::= SEQUENCE {
        countersValue                   [0] CountersValue,
        filteringCriteria               [1] FilteringCriteria
--      ...
        }

SpecializedResourceReportArg            ::= NULL
```

-- The Definition of Common Data Types Follows

```
AChBillingChargingCharacteristics       ::= OCTET STRING (SIZE(minAChBillingChargingLength ..
                                                maxAChBillingChargingLength))

--  The structure of the AChBillingChargingCharacteristics is defined in Teil 3.

AdditionalCallingPartyNumber ::= Digits

--  Indicates the Additional Calling Party Number.

ApplicationTimer    ::= INTEGER(1..60)

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

AssistingSSPIPRoutingAddress ::= Digits

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

BackwardServiceInteractionInd           ::= SEQUENCE {
        conferenceTreatmentIndicator    [1] OCTET STRING (SIZE(1))          OPTIONAL
        --  acceptConferenceRequest     'xxxx xx01'B
        --  rejectConferenceRequest     'xxxx xx10'B
        --  network default is accept conference request
        }

BCSMEvent                       ::= SEQUENCE {
        eventTypeBCSM           [0] EventTypeBCSM,
        monitorMode             [1] MonitorMode,
        legID                   [2] LegID                   OPTIONAL,
        dPSpecificCriteria      [30] DPSpecificCriteria     OPTIONAL
        }

--  Indicates the BCSM Event information for monitoring.

BearerCapability                ::= CHOICE {
        bearerCap               [0] OCTET STRING (SIZE (2..maxBearerCapabilityLength))
        }
```

```
-- Indicates the type of bearer capability connection to the user. For bearerCap,
--  the value as described in
-- DSS1([8])/ISUP([7], User Service Information) shall be used.

CalledPartyNumber                ::= OCTET STRING (SIZE (minCalledPartyNumberLength ..
                                                         maxCalledPartyNumberLength ))

-- Indicates the Called Party Number.  Refer to [7] for encoding.
-- Further encodings in addition to [7]
--                      Address Signal
--                          1 1 1 0          code 14 (Hex E)

CallingPartyNumber               ::= OCTET STRING (SIZE (minCallingPartyNumberLength ..
                                                         maxCallingPartyNumberLength ))

-- Indicates the Calling Party Number.  Refer to [7] for encoding.
-- This argument may also contain an  incomplete Calling Party Number.

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

-- Indicates the type of calling party (e.g., operator, payphone, ordinary subscriber).  Refer to
-- [7] for encoding.

CallResult                       ::= OCTET STRING (SIZE (minCallResultLength ..
                                                         maxCallResultLength))

-- The structure of the CallResult is defined in Teil 3


Cause                    ::= OCTET STRING (SIZE (causeLength ))

-- Indicates the cause for interface related information. Refer to the [7] Cause parameter for
-- encoding. For the use of Cause and Location values refer to Q.850.

CollectedDigits  ::= SEQUENCE {
        minimumNbOfDigits       [0] INTEGER (1.. 50)     DEFAULT 1,
        maximumNbOfDigits       [1] INTEGER (1.. 50),
        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 stdErrorAndInfo,
        interruptableAnnInd     [8]  BOOLEAN   DEFAULT TRUE
        }

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

CollectedInfo                    ::= CHOICE {
      collectedDigits            [0] CollectedDigits
      }

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

```
CorrelationID      ::= Digits

--used by SCF for correlation with a previous operation. Refer to section 7
for a
--description of the procedures associated with this parameter.
```

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

CounterID ::= INTEGER (0..99)

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

CountersValue ::= SEQUENCE SIZE(*1*..numOfCounters) OF CounterAndValue

CutAndPaste ::= INTEGER (0..22)

-- Indicates the number of digits to be deleted. Refer to Q.1214, Section 6.4.2.16 for additional
-- information.

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

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

	Bits	HGFE	DCBA
--	leading octet	3	9
--		9	0
--		0	3
--		2	1
--		5	1
--		1	0

DestinationRoutingAddress ::= SEQUENCE SIZE (1) OF CalledPartyNumber

-- Indicates the Called Party Number.

Digits ::= OCTET STRING (SIZE (minDigitsLength .. maxDigitsLength))

-- Indicates the address signalling digits. Refer to the *[7]* Generic Number and Generic Digits
-- parameters for encoding. The coding of the subfields 'NumberQualifier' in Generic Number and
-- 'Type Of Digits' in Generic Digits are irrelevant to the INAP, the ASN.1 tags are sufficient to
-- identify the parameter. The ISUP format does not allow to exclude these subfields, therefor the
-- value *should be taken from ISUP if available.*
-- The following parameter should use Generic Number :
-- AdditionalCallingPartyNumber for InitialDP,
-- AssistingSSPIPRoutingAddress for EstablishTemporaryConnection,
-- calledAddressValue for all occurances,
--
-- The following parameters should use Generic Digits *(The "Type Of Digits" should be B´11110´)*:
-- CorrelationID
-- number VariablePart,
-- digitsResponse ReceivedInformationArg,

```
DPSpecificCriteria              ::= CHOICE {
        numberOfDigits          [0] NumberOfDigits,
        applicationTimer        [1] ApplicationTimer
        }

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

Duration                        ::= INTEGER ( 0..2047)

-- Values are seconds.


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

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

EventSpecificInformationBCSM    ::=  CHOICE {
        collectedInfoSpecificInfo       [0] SEQUENCE {
                calledPartyNumber               [0] CalledPartyNumber
                },
        routeSelectFailureSpecificInfo  [2] SEQUENCE {
                failureCause                    [0] Cause                       OPTIONAL
                },
        oCalledPartyBusySpecificInfo    [3] SEQUENCE {
                busyCause                       [0] Cause                       OPTIONAL
                },
        oNoAnswerSpecificInfo           [4] SEQUENCE {
                },
        oAnswerSpecificInfo             [5] SEQUENCE {
                },
        oDisconnectSpecificInfo         [7] SEQUENCE {
                releaseCause                    [0] Cause                       OPTIONAL,
                userUser                        [PRIVATE 8] UserUser            OPTIONAL
                },
        tCalledPartyBusySpecificInfo    [8] SEQUENCE {
                busyCause                       [0] Cause                       OPTIONAL
                },
        tNoAnswerSpecificInfo           [9] SEQUENCE {
                },
        tAnswerSpecificInfo             [10] SEQUENCE {
                },
        tDisconnectSpecificInfo         [12] SEQUENCE {
                releaseCause                    [0] Cause                       OPTIONAL
                }
        }

-- Indicates the call related information specific to the event.


EventTypeBCSM                   ::= ENUMERATED {
        origAttemptAuthorized(1),
        collectedInfo(2),
```

```
        analyzedInformation(3),
        routeSelectFailure(4),
        oCalledPartyBusy(5),
        oNoAnswer(6),
        oAnswer(7),
        oAbandon(10),
        oDisconnect(9),
        termAttemptAuthorized(12),
        tCalledPartyBusy(13),
        tNoAnswer(14),
        tAnswer(15),
        tDisconnect(17),
        tAbandon(18)
        }

-- Indicates the BCSM detection point event.  Refer to Q.1214 [11] § 4.2.2.2 for additional information
-- on the events.
-- 163 TR 78:
--        Possible as TDP-R: origAttemptAuthorized, collectedInfo, analyzedInformation,
routeSelectFailure,
--                        termAttemptAuthorized, tCalledPartyBusy, tNoAnswer
--        Possible as EDP-R: collectedInfo, routeSelectFailure, oCalledPartyBusy, oNoAnswer,
--                        oDisconnect (for B-party only), tCalledPartyBusy, tNoAnswer,
--                        tDisconnect (for B-party only)
--        Possible as EDP-N: routeSelectFailure, oCalledPartyBusy, oNoAnswer, oAnswer, oDisconnect,
--                        oAbandon, tCalledPartyBusy, tNoAnswer, tAnswer, tDisconnect, tAbandon

FCIBillingChargingCharacteristics       ::= OCTET STRING (SIZE (minFCIBillingChargingLength ..
                                                        maxFCIBillingChargingLength ))

-- The internal structure of the FCIBIllingChargingCharacteristics is defined in Teil 3
-- using an ASN.1 definition. Applying the Basic Encoding Rules (BER) to this ASN.1 structure
-- provides the content of the OctetString, this means the length of the OctetString
-- is equal to the length of the ASN.1 structure.

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

-- If releaseCause is not present, the default value  is  the same as the ISUP cause value decimal
-- 31.
-- If informationToSend is present, the call will be released after the end of the announcement with
-- the indicated or default releaseCause.

FilteringCharacteristics                ::= CHOICE {
        interval                        [0] INTEGER (1..3600),
        numberOfCalls                   [1]  INTEGER(1..10000)
        }

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

```
FilteringCriteria                ::= CHOICE {

        addressAndService        [30] SEQUENCE {
                                 calledAddressValue      [0] Digits,
                                 serviceKey              [1] ServiceKey          }
        }

--  In case calledAddressValue is specified, the numbers to be filtered are from
--  calledAddressValue upto and including calledAddressValue +maximumNumberOfCounters-1.
--  The total value of: last two digits of calledAddressValue + NumberOfCounters - 1 cannot exceed 99.

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

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


ForwardServiceInteractionInd     ::= SEQUENCE {
                conferenceTreatmentIndicator     [1] OCTET STRING (SIZE(1))    OPTIONAL,
                -- acceptConferenceRequest   'xxxx xx01'B
                -- rejectConferenceRequest    'xxxx xx10'B
                -- network default is accept conference request
                callDiversionTreatmentIndicator   [2] OCTET STRING (SIZE(1))    OPTIONAL,
                -- callDiversionAllowed          'xxxx xx01'B
                -- callDiversionNotAllowed       'xxxx xx10'B
                -- network default is call diversion allowed
                callOfferingTreatmentIndicator    [3] OCTET STRING (SIZE(1))    OPTIONAL
                -- callOfferingNotAllowed        'xxxx xx01'B
                -- callOfferingAllowed           'xxxx xx10'B
                -- network default is call offering not allowed
                -- this parameter is not used by T-Mobil
        }

GapCriteria                      ::= CHOICE {
        gapOnService             [2] GapOnService,
        calledAddressAndService  [29] SEQUENCE {
                                 calledAddressValue      [0] Digits,
                                 serviceKey              [1] ServiceKey
                                 }
        }

--  CalledAddressValue  can be incomplete numbers, in the sense
--  that a limited amount of digits can be given.
--  For the handling of numbers starting with the same digit string refer to the detailed procedure
--  of the CallGap operation in section 9

GapOnService                     ::= SEQUENCE {
        serviceKey               [0] ServiceKey
        }

GapIndicators                    ::= SEQUENCE {
        duration                 [0] Duration,
        gapInterval              [1] Interval
        }

--  Indicates the gapping characteristics. No gapping when gapInterval equals 0, and gap all
--  calls when gapInterval equals-1. For further information regarding the meaning of specific
```

```
-- values of duration and gapinterval refer to the detailed procedure of the CallGap operation
-- in section 9.

GapTreatment                ::= CHOICE {
        informationToSend   [0] InformationToSend,
        releaseCause        [1] Cause,
        both                [2] SEQUENCE {
                                informationToSend   [0] InformationToSend,
                                releaseCause        [1] Cause
                                }
        }

-- The default value for Cause is 42 (switching equipment congestion).

HighLayerCompatibility ::= OCTET STRING (SIZE(highLayerCompatibilityLength))

-- Indicates the teleservice. For encoding, DSS1 ([8]) is used.

IMSI ::= OCTET STRING (SIZE(3..8))
-- This type contains the IMSI.
-- For encoding refer to GSM 03.03 specification.
-- This Parameter is only included in case of mobile originated call treatment.

InbandInfo                  ::= SEQUENCE {
        messageID           [0] MessageID,
        numberOfRepetitions [1] INTEGER (1..127)        OPTIONAL,
        duration            [2] INTEGER ( 0..4095)      OPTIONAL,
        interval            [3] INTEGER ( 1..4095)      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.


INContainer                 ::= OCTET STRING (SIZE (1..4))

-- This parameter may be used by a service logic to provide a message in forward direction
-- to another service logic which is invoked later for the same call.

InformationToSend           ::= CHOICE {
        inbandinfo          [0] InbandInfo,
        tone                [1] Tone,
        userDialogInfo      [PRIVATE 0] UserDialogInfo
        }

-- The userDialogInfo parameter in the InformationToSend shall only be sent in the
-- PlayAnnouncement operation, in other operations using the InformationToSend parameter
-- the choice userDialogInfo is not allowed to be sent.


Integer4        ::= INTEGER (0..2147483647)

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

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

IPRoutingAddress        ::= CalledPartyNumber
```

```
-- Indicates the routing address for the IP.


LegID ::= CHOICE{
        sendingSideID                   [0] LegType,
--        used in operations sent from SCF to SSF
        receivingSideID                 [1] LegType
--        used in operations sent from SSF to SCF
        }

-- Indicates a reference to a specific party in a call.  OPTIONAL denotes network operator
-- specific use with  unilateral ID assignment .
-- OPTIONAL for LegID also denotes the following:
--   - when only one party exists in the call, this parameter is not needed (as no ambiguity exists).
--   - when more than one party exists in the call, one of the following alternatives applies:
--        1. LegID is present and indicates which party is concerned.
--        2. LegID is not present and a default value is assumed

LegType                         ::= OCTET STRING (SIZE(1))
        leg1 LegType            ::= '01'H
        leg2 LegType            ::= '02'H

LocationNumber              ::= OCTET STRING (SIZE (minLocationNumberLength ..
                                                    maxLocationNumberLength ))

-- Indicates the Location Number for the calling party. Refer to [7] for
encoding.

LocationNumberB         ::= OCTET STRING (SIZE (1 .. 9 ))

-- Indicates the location number of the called mobile subscriber. The locationNumberB will be
-- represented by the MSRN. For encoding refer to MAP [19] Address String parameter.
-- The parameter is only included in the case of IN-Mobile terminating Call treatment.


MaximumNumberOfCounters  ::= INTEGER (1.. numOfCounters)

MessageID                       ::= CHOICE {
      elementaryMessageID       [0] INTEGER(0..32767),
      elementaryMessageIDs      [29] SEQUENCE SIZE (1..numOfMessageIDs) OF
                                                INTEGER(0..32767),
      variableMessage           [30] SEQUENCE {
                                elementaryMessageID         [0] INTEGER(0..32767),
                                variableParts               [1] SEQUENCE SIZE(1..5) OF
                                                                VariablePart
                                }
      }

MiscCallInfo                        ::= SEQUENCE {
        messageType                 [0] ENUMERATED {
                                        request(0),
                                        notification(1)
                                        }
        }

-- Indicates detection point related information.

MonitorMode                         ::= ENUMERATED {
        interrupted(0),
```

notifyAndContinue(1)}

-- Indicates the event is relayed and/or processed by the SSP.

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

-- Indicates the National Calling Party Category. Refer to [7] for encoding.

NatServiceInteractionIndicators ::= OCTET STRING(SIZE(minNatServiceInteractionIndicatorsLength..
maxNatServiceInteractionIndicatorsLength))

-- Indicators which are exchanged between SSP and SCP to resolve interactions between
-- IN based services and network based services, respectively between different IN based
-- services.
-- The internal structure of the NatServiceInteractionIndicators is defined in Teil 3
-- using an ASN.1 definition. Applying the Basic Encoding Rules (BER) to this ASN.1 structure
-- provides the content of the OctetString, this means the length of the OctetString
-- is equal to the length of the ASN.1 structure.

NumberOfDigits::=INTEGER(1.. *22*)

-- Indicates the number of digits to be collected

OriginalCalledPartyID ::= OCTET STRING (SIZE (minOriginalCalledPartyIDLength ..
maxOriginalCalledPartyIDLength))

-- Indicates the original called number. Refer to the *[7]* Original Called Number for encoding.

RedirectingPartyID ::= OCTET STRING (SIZE (minRedirectingPartyIDLength ..
maxRedirectingPartyIDLength))

-- Indicates redirecting number. Refer to the *[7]* Redirecting number for encoding.

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

-- Indicates redirection information. Refer to the *[7]* Redirection Information for encoding.

RequestedInformationList ::= SEQUENCE SIZE(1..numOfInfoItems) OF RequestedInformation

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

```
RequestedInformation                ::= SEQUENCE {
        requestedInformationType    [0] RequestedInformationType,
        requestedInformationValue   [1] RequestedInformationValue
        }

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

RequestedInformationValue           ::= CHOICE {
        callAttemptElapsedTimeValue [0] INTEGER (0..255),
        callStopTimeValue           [1] DateAndTime,
        callConnectedElapsedTimeValue [2] Integer4,
```

releaseCauseValue [30] Cause
}

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

```
ScfID   ::=  OCTET STRING (SIZE (minScfIDLength .. maxScfIDLength ))

--Indicates the SCF identifier.
```
-- *To be encoded like SCCP Called Party Address [2].*

SCIBillingChargingCharacteristics ::= OCTET STRING (SIZE (minSCIBillingChargingLength .. maxSCIBillingChargingLength))

-- *The internal structure of the SCIBIllingChargingCharacteristics is defined in Teil 3*
-- *using an ASN.1 definition. Applying the Basic Encoding Rules (BER) to this ASN.1 structure*
-- *provides the content of the OctetString, this means the length of the OctetString*
-- *is equal to the length of the ASN.1 structure.*

ServiceInteractionIndicatorsTwo ::= SEQUENCE {
forwardServiceInteractionInd [0] ForwardServiceInteractionInd OPTIONAL,
-- applicable to operation CON
backwardServiceInteractionInd [1] BackwardServiceInteractionInd OPTIONAL,
-- applicable to operations CON, CTR, ETC.
connectedNumberTreatmentInd [4] ConnectedNumberTreatmentInd OPTIONAL,
-- applicable to operation CON.
-- This parameter is not used by T-Mobil
allowCdINNoPresentationInd [7] BOOLEAN OPTIONAL,
-- applicable to operation CON
-- Indicates whether the Number Presentation not allowed indicator
-- of the ISUP ”called IN number” shall be set to presentation allowed (TRUE)
-- or presentation not allowed (FALSE).
-- This parameter is not used by T-Mobil
userDialogueDurationInd [8] BOOLEAN OPTIONAL
-- applicable when interaction with the user is required
-- TRUE means the user interaction may last longer than 90 seconds
-- Otherwise the indicator should be set to FALSE
-- ...
}
-- Indicators which are exchanged between SSP and SCP to resolve interactions between IN based
-- services and network based services.
--The ServiceInteractionIndicatorsTwo parameter and the NatServiceInteractionIndicators can
-- be received in the same INAP operation. In case, the interaction indicators provided
-- in both parameters control the same interworking procedures, the interaction indicators
-- provided by ServiceInteractionIndicatorsTwo have precedence.
-- For detailed interworking procedures refer to [7].

ServiceKey ::= *INTEGER(0..32767)*

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

SFBillingChargingCharacteristics ::= OCTET STRING (SIZE (minSFBillingChargingLength .. maxSFBillingChargingLength))

-- *The internal structure of the SFBIllingChargingCharacteristics is defined in Teil 3*
-- *using an ASN.1 definition. Applying the Basic Encoding Rules (BER) to this ASN.1 structure*
-- *provides the content of the OctetString, this means the length of the OctetString*
-- *is equal to the length of the ASN.1 structure.*

-- This parameter indicates the billing and/or charging characteristics for filtered calls.

```
TimerID                         ::= ENUMERATED {
      tssf(0)
      }

-- Indicates the timer to be reset.

TimerValue   ::= INTEGER(1..32767)

-- Indicates the timer  value (in seconds).
```

```
Tone  ::= SEQUENCE {
      toneID                    [0]  INTEGER(1..32),
      duration                  [1]  INTEGER(1..4095)          OPTIONAL
      }
```

-- The duration specifies the length of the tone in seconds.

```
UnavailableNetworkResource          ::= ENUMERATED {
      unavailableResources(0),
      componentFailure(1),
      basicCallProcessingException(2)                                  }
```

-- Indicates the network resource that failed.

```
UserDialogInfo  ::=       OCTET STRING (SIZE(1..100))

--  The userDialogInfo parameter indicates the USSD information from/to a Mobile Station, coded as
--  component of the Facility-Information-Element as specified in GSM 04.80 [20]..
--  The parameter can only be used in the InitialDP and the PlayAnnouncement operation.

UserUser        ::= OCTET STRING (SIZE(minUserUserLength..maxUserUserLength))

-- Refer to [8] User-user information element for encoding
```

```
VariablePart                ::= CHOICE {
      integer               [0]  INTEGER(0..999999),
      number                [1] Digits,                        -- Generic digits
      time                  [2] OCTET STRING (SIZE(2)),        -- HH:MM, BCD coded
      date                  [3] OCTET STRING (SIZE(3)),        -- YYMMDD, BCD coded
      price                 [4] OCTET STRING (SIZE(4))         -- DDDDDD.DD, BCD coded
      }
```

```
-- Indicates the variable part of the message
--
-- BCD coded variable parts are encoded as described in the examples below.
-- For example, time = 12:15 would be encoded as:
--      Bits            HGFE    DCBA
--      leading octet      2       1
--                         5       1
-- date = 1993 September 30th would be encoded as:
--      Bits            HGFE    DCBA
--      leading octet      3       9
--                         9       0
--                         0       3
-- price = ECU 249.50 would be encoded as:
--      Bits            HGFE    DCBA
--      leading octet      0       0
--                         0       0
--                         2       0
--                         9       4
--                         0       5


-- Definition of range constants

highLayerCompatibilityLength                INTEGER ::= 2
minAChBillingChargingLength                 INTEGER ::= 2
maxAChBillingChargingLength                 INTEGER ::= 134
maxBearerCapabilityLength                   INTEGER ::= 11
minCalledPartyNumberLength                  INTEGER ::= 2
maxCalledPartyNumberLength                  INTEGER ::= 13
minCallingPartyNumberLength                 INTEGER ::= 2
maxCallingPartyNumberLength                 INTEGER ::= 13
minCallResultLength                         INTEGER ::= 5
maxCallResultLength                         INTEGER ::= 63
causeLength                                 INTEGER ::= 2
minDigitsLength                             INTEGER ::= 2
maxDigitsLength                             INTEGER ::= 26
minFCIBillingChargingLength                 INTEGER ::= 12
maxFCIBillingChargingLength                 INTEGER ::= 56
minLocationNumberLength                     INTEGER ::= 2
maxLocationNumberLength                     INTEGER ::= 12
minNatServiceInteractionIndicatorsLength    INTEGER ::= 2
maxNatServiceInteractionIndicatorsLength    INTEGER ::= 56
minOriginalCalledPartyIDLength              INTEGER ::= 3
maxOriginalCalledPartyIDLength              INTEGER ::= 13
minRedirectingPartyIDLength                 INTEGER ::= 3
maxRedirectingPartyIDLength                 INTEGER ::= 13
minScfIDLength                              INTEGER ::= 2
maxScfIDLength                              INTEGER ::= 13
minSCIBillingChargingLength                 INTEGER ::= 2
maxSCIBillingChargingLength                 INTEGER ::= 92
minSFBillingChargingLength                  INTEGER ::= 2
maxSFBillingChargingLength                  INTEGER ::= 817
minUserUserLength                           INTEGER ::= 1
maxUserUserLength                           INTEGER ::= 32
numOfBCSMEvents                             INTEGER ::= 6
numOfCounters                               INTEGER ::= 30
numOfInfoItems                              INTEGER ::= 4
numOfMessageIDs                             INTEGER ::= 16
```

END

6.4 IN CS1 Application Protocol (Operation and Error Codes)

-- This module contain the operation and error code assignments for the IN CS1 application
-- protocol.

Telekom-INAP-CS1-Codes {ccitt(0) *administration(2) bmpt(262) telekom(1) zgs_nr7(5) inap(0)* modules(0) cs1-codes(3) version5(4)}

DEFINITIONS IMPLICIT TAGS ::=

BEGIN

-- OPERATION AND ERROR CODE ASSIGNMENTS FOR THE **IN CS1** PROTOCOL
-- FOLLOWS

IMPORTS

-- macros

APPLICATION-SERVICE-ELEMENT
FROM Remote-Operations-Notation-Extension
{joint-iso-ccitt remote-operations(4) notation-extension(2)}
-- operation types
ActivateServiceFiltering,
ActivityTest,
ApplyCharging,
ApplyChargingReport,
AssistRequestInstructions,
CallGap,
CallInformationReport,
CallInformationRequest,
CollectInformation,
Connect,
ConnectToResource,
Continue,
DisconnectForwardConnection,
EstablishTemporaryConnection,
EventReportBCSM,
FurnishChargingInformation,
InitialDP,
PlayAnnouncement,
PromptAndCollectUserInformation,
ReleaseCall,
RequestReportBCSMEvent,
ResetTimer,
SendChargingInformation,
ServiceFilteringResponse,
SpecializedResourceReport

FROM *Telekom*-INAP-CS1-Operations { ccitt(0) *administration(2) bmpt(262) telekom(1) zgs_nr7(5) inap(0)* modules(0) cs1-operations(0) version5(4)}

-- error types
ETCFailed,
ImproperCallerResponse,
MissingCustomerRecord,
MissingParameter,
ParameterOutOfRange,
SystemFailure,
TaskRefused,
UnavailableResource,
UnexpectedComponentSequence,
UnexpectedDataValue,
UnexpectedParameter

FROM *Telekom*-INAP-CS1-Errors {ccitt(0) *administration(2) bmpt(262) telekom(1) zgs_nr7(5) inap(0)* modules(0) cs1-errors(1) version3(2)} ;

-- the operations are grouped by the identified ASEs.

-- SCF activation ASE

initialDP InitialDP ::= localValue 0

-- SCF/SRF activation of assist ASE

assistRequestInstructions AssistRequestInstructions ::= localValue 16

-- Assist connection establishment ASE

establishTemporaryConnection EstablishTemporaryConnection ::= localValue 17

-- Generic disconnect resource ASE

disconnectForwardConnection DisconnectForwardConnection ::= localValue 18

-- Non-assisted connection establishment ASE

connectToResource ConnectToResource ::= localValue 19

-- Connect ASE (elementary SSF function)

connect Connect ::= localValue 20

-- Call handling ASE (elementary SSF function)

releaseCall ReleaseCall ::= localValue 22

-- BCSM Event handling ASE

requestReportBCSMEvent RequestReportBCSMEvent ::= localValue 23
eventReportBCSM EventReportBCSM ::= localValue 24

-- SSF call processing ASE

collectInformation CollectInformation ::= localValue 27
continue Continue ::= localValue 31

-- Timer ASE

```
resetTimer                      ResetTimer                      ::= localValue 33
```

-- Billing ASE

furnishChargingInformation	FurnishChargingInformation	::= localValue 34

-- Charging ASE

applyCharging	ApplyCharging	::= localValue 35
applyChargingReport	ApplyChargingReport	::= localValue 36

-- Traffic management ASE

callGap	CallGap	::= localValue 41

-- Service management ASE

activateServiceFiltering	ActivateServiceFiltering	::= localValue 42
serviceFilteringResponse	ServiceFilteringResponse	::= localValue 43

-- Call report ASE

callInformationReport	CallInformationReport	::= localValue 44
callInformationRequest	CallInformationRequest	::= localValue 45

-- Signalling control ASE

sendChargingInformation	SendChargingInformation	::= localValue 46

-- Specialized resource control ASE

playAnnouncement	PlayAnnouncement	::= localValue 47
promptAndCollectUserInformation	PromptAndCollectUserInformation	::= localValue 48
specializedResourceReport	SpecializedResourceReport	::= localValue 49

-- Activity Test ASE

activityTest	ActivityTest	::= localValue 55

-- ERROR codes

```
eTCFailed                       ETCFailed                       ::= localValue 3
```

improperCallerResponse	ImproperCallerResponse	::= localValue 4
missingCustomerRecord	MissingCustomerRecord	::= localValue 6
missingParameter	MissingParameter	::= localValue 7
parameterOutOfRange	ParameterOutOfRange	::= localValue 8
systemFailure	SystemFailure	::= localValue 11
taskRefused	TaskRefused	::= localValue 12
unavailableResource	UnavailableResource	::= localValue 13
unexpectedComponentSequence	UnexpectedComponentSequence	::= localValue 14
unexpectedDataValue	UnexpectedDataValue	::= localValue 15
unexpectedParameter	UnexpectedParameter	::= localValue 16

-- APPLICATION SERVICE ELEMENTS

```
SCF-Activation-ASE      ::= APPLICATION-SERVICE-ELEMENT
        -- consumer is SSF
        CONSUMER INVOKES {
                initialDP
                }
        -- supplier is SCF
        SUPPLIER INVOKES

SCF-SRF-activation-of-assist-ASE     ::= APPLICATION-SERVICE-ELEMENT
       -- consumer is SSF
       CONSUMER INVOKES {
              assistRequestInstructions
              }
       -- supplier is SCF
       SUPPLIER INVOKES

Assist-connection-establishment-ASE ::= APPLICATION-SERVICE-ELEMENT
       -- consumer is SSF
       CONSUMER INVOKES
       -- supplier is SCF
       SUPPLIER INVOKES {
              establishTemporaryConnection
              }

Generic-disconnect-resource-ASE         ::= APPLICATION-SERVICE-ELEMENT
        -- consumer is SSF
        CONSUMER INVOKES
        -- supplier is SCF
        SUPPLIER INVOKES {
                disconnectForwardConnection
                }

Non-assisted-connection-establishment-ASE      ::= APPLICATION-SERVICE-ELEMENT
        -- consumer is SSF
        CONSUMER INVOKES
        -- supplier is SCF
        SUPPLIER INVOKES {
                connectToResource
                }

Connect-ASE    ::= APPLICATION-SERVICE-ELEMENT
        -- consumer is SSF
        CONSUMER INVOKES
        -- supplier is SCF
        SUPPLIER INVOKES {
                connect
                }

Call-handling-ASE        ::= APPLICATION-SERVICE-ELEMENT
        -- consumer is SSF
        CONSUMER INVOKES
        -- supplier is SCF
        SUPPLIER INVOKES {
                releaseCall
                }

BCSM-event-handling-ASE        ::= APPLICATION-SERVICE-ELEMENT
        -- consumer is SSF
        CONSUMER INVOKES {
                eventReportBCSM
```

```
                }
        -- supplier is SCF
        SUPPLIER INVOKES {
                requestReportBCSMEvent
                }

SSF-call-processing-ASE         ::= APPLICATION-SERVICE-ELEMENT
        -- supplier is SCF
        SUPPLIER INVOKES {
                collectInformation,
                continue
                }


Timer-ASE    ::= APPLICATION-SERVICE-ELEMENT
       -- consumer is SSF/SRF
       CONSUMER INVOKES
       -- supplier is SCF
       SUPPLIER INVOKES {
              resetTimer
              }

Billing-ASE       ::= APPLICATION-SERVICE-ELEMENT
        -- consumer is SSF
        CONSUMER INVOKES
        -- supplier is SCF
        SUPPLIER INVOKES {
                furnishChargingInformation
                }

Charging-ASE   ::= APPLICATION-SERVICE-ELEMENT
        -- consumer is SSF
        CONSUMER INVOKES {
                applyChargingReport
                }
        -- supplier is SCF
        SUPPLIER INVOKES {
                applyCharging
                }


Traffic-management-ASE          ::= APPLICATION-SERVICE-ELEMENT
        --consumer is SSF
        CONSUMER INVOKES
        -- supplier is SCF
        SUPPLIER INVOKES {
                callGap
                }

Service-management-ASE        ::= APPLICATION-SERVICE-ELEMENT
        -- consumer is SSF
        CONSUMER INVOKES {
                serviceFilteringResponse
                }
        -- supplier is SCF
        SUPPLIER INVOKES {
                activateServiceFiltering
                }
```

```
Call-report-ASE ::= APPLICATION-SERVICE-ELEMENT
        -- consumer is SSF
        CONSUMER INVOKES {
                callInformationReport
                }
        -- supplier is SCF
        SUPPLIER INVOKES {
                callInformationRequest
                }

Signalling-control-ASE   ::= APPLICATION-SERVICE-ELEMENT
        -- consumer is SSF
        CONSUMER INVOKES
        -- supplier is SCF
        SUPPLIER INVOKES {
                sendChargingInformation
                }

Specialized-resource-control-ASE        ::= APPLICATION-SERVICE-ELEMENT
        -- consumer is SSF/SRF
        CONSUMER INVOKES {
                specializedResourceReport
                }
        -- supplier is SCF
        SUPPLIER INVOKES {
                playAnnouncement,
                promptAndCollectUserInformation
                }


Activity-test-ASE       ::= APPLICATION-SERVICE-ELEMENT
        -- consumer is SSF/SRF
        CONSUMER INVOKES
        -- supplier is SCF
        SUPPLIER INVOKES {
                activityTest
                }


END
```

6.5 IN CS1 Application Contexts

```
APPLICATION-CONTEXT MACRO::=
BEGIN
TYPE NOTATION       ::= Symmetric | InitiatorConsumerOf ResponderConsumerOf | empty
VALUE NOTATION      ::= value(VALUE OBJECT IDENTIFIER)
Symmetric           ::= "OPERATIONS OF" "{" ASEList "}"
InitiatorConsumerOf     ::= "INITIATOR CONSUMER OF" "{" ASEList "}" | empty
ResponderConsumerOf ::= "RESPONDER CONSUMER OF" "{" ASEList "}" | empty
ASEList         ::= ASE | ASEList "," ASE
ASE                 ::= type -- shall reference an APPLICATION-SERVICE-ELEMENT type.
END

Telekom-INAP-CS1-SSP-to-SCP-AC      APPLICATION-CONTEXT
-- dialogue initiated by SSP with InitialDP
INITIATOR CONSUMER OF {
        SCF-activation-ASE,
        Assist-connection-establishment-ASE,
```

```
        Generic-disconnect-resource-ASE,
        Non-assisted-connection-establishment-ASE,
        Connect-ASE
        Call-handling-ASE,
        BCSM-event-handling-ASE,
        SSF-call-processing-ASE,
        Timer-ASE,
        Billing-ASE,
        Charging-ASE,
        Traffic-management-ASE,
        Call-report-ASE,
        Signalling-control-ASE,
        Specialized-resource-control-ASE,
        Activity-test-ASE
        }
RESPONDER CONSUMER OF
::= {ccitt(0)  administration(2)  bmpt(262)  telekom(1)  zgs_nr7(5)  inap(0)  ac(1)  cs1-ssp-to-scp(0)
version5(4)} ;

Telekom-INAP-CS1-assist-SSP-to-SCP-AC      APPLICATION-CONTEXT
-- dialogue initiated by SSP with AssistRequestInstructions
INITIATOR CONSUMER OF {
      SCF-SRF-activation-of-assist-ASE,
      Generic-disconnect-resource-ASE,
      Non-assisted-connection-establishment-ASE,
      Timer-ASE,
      Specialized-resource-control-ASE,
      Activity-test-ASE
      }
RESPONDER CONSUMER OF
::= {
{ccitt(0) administration(2) bmpt(262) telekom(1) zgs_nr7(5) inap(0) ac(1) cs1-assist-ssp-to-
scp(1) version2(1)} ;
::= {ccitt(0) administration(2) bmpt(262) telekom(1) zgs_nr7(5) inap(0) ac(1) cs1-assist-
ssp-to-scp(1) version3(2)}
};

-- Both versions, i.e. "version2(1)" and "version3(2)", are valid for 'Telekom-INAP-CS1-assist-SSP-
to-
-- SCP-AC' of T-INAP version 163TR78.97.
--  Within  " version2(1)"  the  ASEs  of  T-INAP  version  163TR78.96  are
applicable.

-- Note: This is because there are different SSP implementations.
-- In the Siemens implementation the "version2(1)" includes the ASEs of 163TR78.97 and
-- 163TR78.96.
-- In the Alcatel implementation the "version2(1)" includes the ASEs of 163TR78.96; ASEs of
-- 163TR78.97 are part of "version3(2).
-- In a new version of T-INAP the differences in the implementation should be removed.


Telekom-INAP-CS1-SCP-to-SSP-traffic-management-AC          APPLICATION-CONTEXT
-- dialogue initiated by SCP with CallGap
INITIATOR CONSUMER OF
RESPONDER CONSUMER OF {
        traffic-management-ASE
        }
::= {ccitt(0) administration(2) bmpt(262) telekom(1) zgs_nr7(5) inap(0) ac(1) cs1-scp-to-ssp-traffic-
management(4) version1(0)} ;
```

Telekom-INAP-CS1-SCP-to-SSP-service-management-AC APPLICATION-CONTEXT
-- dialogue initiated by SCP with ActivateServiceFiltering
INITIATOR CONSUMER OF
RESPONDER CONSUMER OF {
service-management-ASE
}
::= {ccitt(0) *administration(2) bmpt(262) telekom(1) zgs_nr7(5) inap(0)* ac(1) cs1-scp-to-ssp-service-management(5) version2(1)3(2)} ;

Telekom-INAP-CS1-SSP-to-SCP-service-management-AC APPLICATION-CONTEXT
-- dialogue initiated by SSP with ServiceFilteringResponse
INITIATOR CONSUMER OF {
service-management-ASE
}
RESPONDER CONSUMER OF
::= {ccitt(0) *administration(2) bmpt(262) telekom(1) zgs_nr7(5) inap(0)* ac(1) cs1-ssp-to-scp-service-management(6) version1(0)} ;

7 Application entity procedures

7.1 SSF application entity procedures

7.1.1 General

This section provides the definition of the SSF Application Entity (AE) procedures related to the SSP - SCP interface. The procedures are based on the use of Common Channel Signalling System N. 7 (CCS #7).

Capabilities not explicitly covered by these procedures may be supported in an implementation dependent manner in the SSP, while remaining in line with Q.1218 section 2.

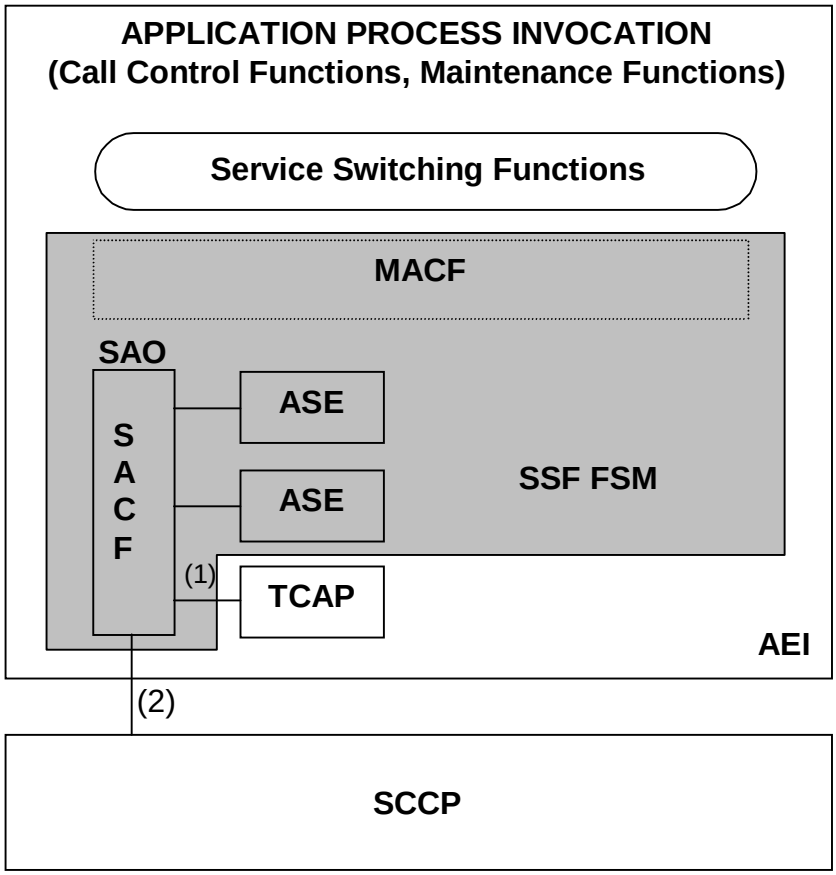
The AE, following the architecture defined in the CCITT Recommendations Q.700, Q.771 and Q.1400, includes TCAP (Transaction Capabilities Application Part) and one or more ASEs called TC-users. The following sections define the TC-user ASE which interfaces with TCAP using the primitives specified in CCITT Recommendation Q.771.

The procedure may equally be used with other signalling message transport systems supporting the Application Layer structures defined.

In case interpretations for the application entity procedures defined in the following differ from detailed procedures and the rules for using of TCAP service, the statements and rules contained in the detailed clause 9 and 10 shall be followed.

7.1.2 Model and interfaces

The functional model of the AE-SSF is shown in figure 9; the ASEs interface to TCAP to communicate with the SCF, and interface to the Call Control Function (CCF) and the maintenance functions already defined for switching systems. The scope of this Recommendation is limited to the shaded area in figure 9.



(1) TC-Primitives

(2) N-Primitives

AEI: Application Entity Invocation
SSF: Service Switching Functions
FSM: Finite State Model
MACF: Multiple Association Control Function
SACF: Single Association Control Function
SAO: Single Association Object

NOTE: Note that SSF FSM includes several Finite State Machines.

Figure 9: Functional Model of SSF AE

The interfaces shown in figure 9 use the TC-user ASE primitives specified in CCITT Recommendation Q.771 (interface (1)) and N-Primitives specified in CCITT Recommendation Q.711 (interface (2)). The operations and parameters of Intelligent Network Application Protocol (INAP) are defined in section *6* of this Recommendation.

7.1.3 Relations between SSF FSM and the CCF and maintenance functions

The primitive interface between the SSF FSM and the CCF/maintenance functions is an internal interface and is not subject to standardization in CS1. Nevertheless this interface should be in line with the BCSM defined in section 4.2.1.2/Q.1214.

The relationship between the BCSM and the SSF FSM may be described as follows for the case of a call/attempt initiated by an end user, and the case of a call/attempt initiated by IN service logic:

- When a call/attempt is initiated by an end user and processed at an exchange, an instance of a BCSM is created. As the BCSM proceeds, it encounters Detection Points (DPs, see Q.1214, section 4.2). If a DP is armed as a Trigger DP (TDP) an instance of an SSF FSM is created.

The management functions related to the execution of operations received from the SCF are executed by the SSF Management Entity (SSME). The SSME comprises a SSME-Control and several instances of SSME FSMs. The SSME-control interfaces the different SSF FSMs and SSME FSMs respectively and the Functional Entity Access Manager (FEAM). figure 10 shows the SSF Interfaces.

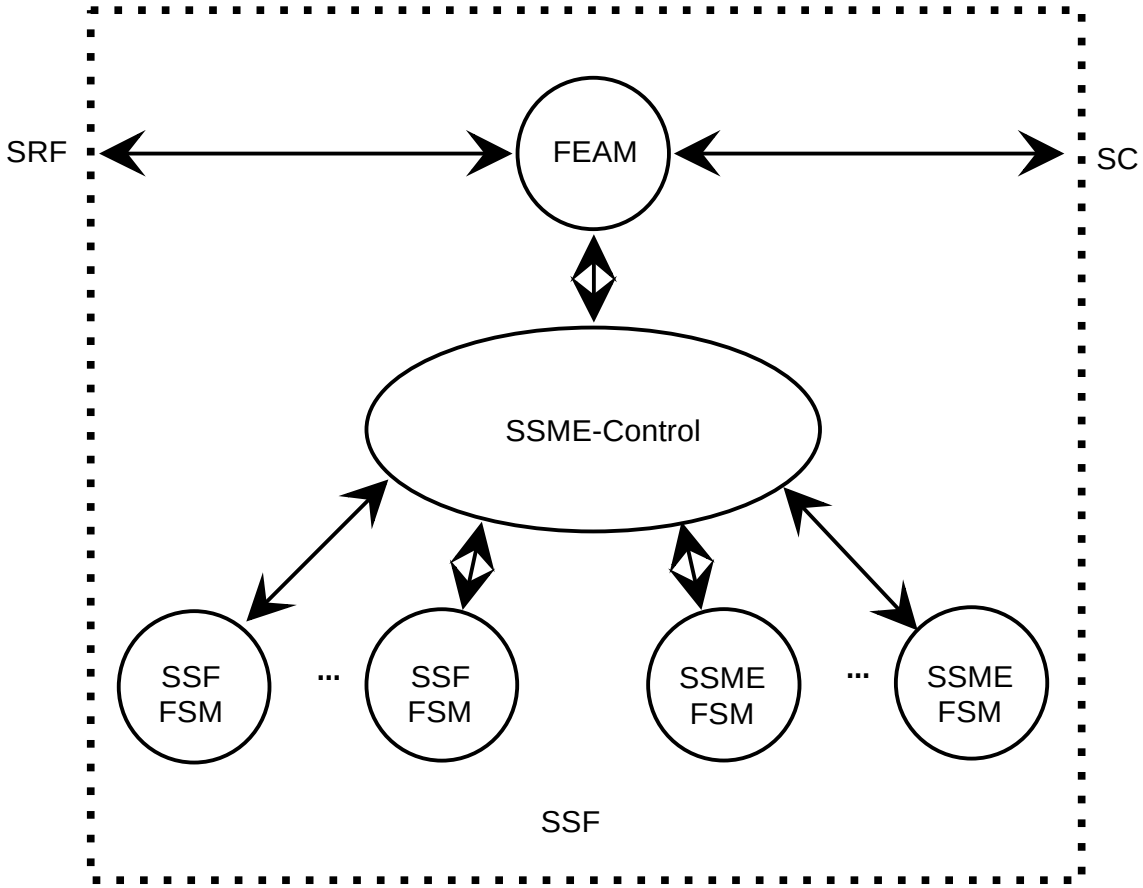


Figure 10: SSF Interfaces

The Functional Entity Access Manager (FEAM) provides the low level interface maintenance functions including the following:

1. Establishing and maintaining the interfaces to the SCF and SRF;

2. Passing and queueing (when necessary) the messages received from the SCF and SRF to the SSME-Control ;

3. Formatting, queueing (when necessary), and sending the messages received from the SSME-Control to the SCF and SRF.

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

1. Interprets the input messages from other FEs and translates them into corresponding SSF FSM events;

2. Translates the SSF FSM outputs into corresponding messages to other FEs.

3. Captures asynchronous (with call processing) activities related to management or supervisory functions in the SSF and creates an instance of a SSME FSM. For example, the SSME provides non-call associated treatment due to changes in Service Filtering or Call Gapping. Therefore, the SSME-control separates the SSF FSM from the Call Gapping and Service Filtering functions by creating instances of SSME FSMs for each context of management related operations.

The different contexts of the SSME FSMs may be distinguished based on the address information provided in the initiating operations. In the case of service filtering this address information is given by filteringCriteria, i.e. all ActivateServiceFiltering operations using the same address, address the same SSME-FSM handling this specific service filtering instance. For example ActivateServiceFiltering operations providing different filtering Criteria cause the invocation of new SSME-FSM's.

The SSF FSM passes call handling instructions to the related instances of the BCSM as needed. DPs may be dynamically armed as Event DPs, requiring the SSF FSM to remain active. At some point, further interaction with the SCF is not needed, and the SSF FSM may be terminated while the BCSM continues to handle the call as needed. A later TDP in the BCSM may result in a new instance of the SSF FSM for the same call.

Consistent with the single-ended control characteristic of IN service features for CS1, the SSF FSM only applies to a functionally separate call portion (e.g., the originating BCSM or the terminating BCSM in a two-party call, but not both).

7.1.4 SSF Management Finite State Model (SSME FSM)

The SSME FSM State Diagram is described in figure 11.

The SSME FSM is independent of the individual SSF FSMs.

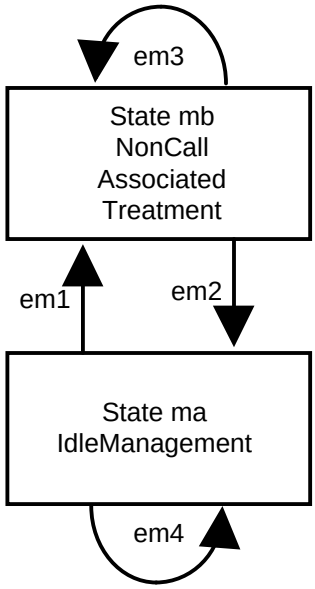


Figure 11: SSME FSM State Diagram

The **Non-Call Associated Treatment** state is entered from the **IdleManagement** State when one of the following non-call associated operations is received (transition em1):

ActivateServiceFiltering
CallGap
Activity Test

The *CallGap operation* may be received inside as well as outside a call context transaction. *ActivateServiceFiltering can not be received inside a call context transaction.*

During this State the following events can occur:

- given that Service Filtering is active, the SSF needs to send a Service Filtering Response to the SCF: the SSME-FSM remains in this state (transition em3);

- given that Service Filtering is active and the Service Filtering duration expires: the SSME-FSM should move to the **IdleManagement** state (transition em2) and send a ServiceFilteringResponse operation to the SCF

- If Call Gap related duration timer expires, the SSME-FSM should move to the **IdleManagement** state (transition em2);

- given that Call Gap/Service Filtering is active, another Call Gap/ActivateServiceFiltering operation could be received by the SSF, which has the same gapping/filtering criteria: the second "filter" or "gap" replace the first one (transition em3) unless the duration timer value is equal to zero, in which case the SSF should move to the Idle Management state (transition em2).

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

7.1.5 SSF State Transition Diagram

Figure 12 shows the State diagram of the SSF part of the SSP during the processing of an IN call/attempt.

Each State is discussed in the following sub-sections. General rules applicable to more than one state are addressed here.

One or a sequence of components received in one or more TCAP messages may include a single operation or multiple operations, and is processed as follows:

- Process the operations in the order in which they are received.

- Each operation causes a state transition independent of whether or not a single operation or multiple operations are received in a message.

- The SSF examines subsequent operations in the sequence. As long as sequential execution of these operations would leave the FSM in the same state, it will execute them (e.g., RequestReportBCSMEvent). If a subsequent operation causes a transition out of the state then the following operations should be buffered until the current operation has been executed. In all other cases, await an event that would cause a transition out of the current state (such an event would be the completion of operation being executed, or reception of an external event. An example of this is as follows:

  The SSF receives the operations FurnishChargingInformation, ConnectTo-Resource, and PlayAnnouncement in a component sequence inside a single TCAP message. Upon receipt of this message, these operations are executed up to and including ConnectToResource while the SSF is in the **Waiting for Instruction** state. As the ConnectToResource operation is executed (and when, or after the FurnishChargingInformation operation has been completed), the SSF FSM will transition to the **Waiting for End of User Interaction** state. The PlayAnnouncement operation is relayed to the SRF while the SSF is in **Waiting for End of User Interaction** state.

- If there is an error in processing one of the operations in the sequence, the SSF 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. violates the SACF rules defined by the SSF FSM) as described above, the SSF FSM processes the error according to the rules given in section 10 (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 SSF FSM remains in the same State as when it received the erroneous operation, however different error treatment is possible in specific cases as described in section 8/3.2; depending on the class of the operation, the error could be reported by the SSF to the SCF using the appropriate component (Recommendation Q.774).

In any State (except **Idle**), if the calling party abandons the call before it is answered (i.e., before the Active PIC in the BCSM), then the SSF FSM should instruct the CCF to clear the call and ensure that any CCF resources allocated to the call have been de-allocated, then continue processing as follows:~~*sends a U-ABORT with UserAbortInformation "callerAbandon"*~~.

If the Abandon DP is not armed, the SSF FSM sends a TC-U-ABORT with UserAbortInformation 'callerAbandon'. Two cases can be distinguished:

- if the Abandon DP is not armed and there is no CallInformationRequest pending, then transition to the **Idle** state;

- if the Abandon DP is not armed and there is a CallInformationRequest pending, send a CallInformationReport, then transition to the **Idle** state;

In case of 'callerAbandon' has been detected in the state Waiting for Instructions the TC-U-ABORT (with the appropriate UserAbortInformation) will only be sent after receipt of any operation from the SCF or after expiration of the Tssf timer. In every case the SSF transits into the Idle state.'

If the Abandon DP is armed, the following cases can be distinguished:

- if the Abandon DP is armed as an EDP-N and there is no CallInformationRequest pending, send an EventReportBCSM, then transistion to the **Idle** state

- if the Abandon DP is armed as an EDP-N and there is a CallInformationRequest pending, send an EventReportBCSM and a CallInformationReport, then transition to the **Idle** state

Other pending requests that are treated in the same way as the CallInformationRequest operation in the above list is the ApplyCharging operation.

~~*In case 'caller abandon' has been detected in the state Waiting for Instructions , the TC-U-ABORT (with the appropriate User Abort Information) will only be sent after receipt of any operation from the SCF or after expiration of the TSSF timer. In every case the SSF transits into the Idle state.*~~

In any State (except **Idle)**, if a call party disconnects from a stable call (i.e., from the Active PIC in the BCSM), then the SSF FSM should process this event as follows:

- If the Disconnect DP is not armed for that specific leg and neither Call Information Request nor a request to send ApplyChargingReport is pending, transition to the **Idle** state . *In case the disconnect DP has been armed for the other leg the TC-user-abort (with User-Abort-Information ´caller DisconnectNo CIR req´) is sent to the SCF before transition to idle.*

- If the Disconnect DP is not armed and there is a Call Information Request or a request to send ApplyChargingReport pending, send a CallInformationReport and/or an ApplyChargingReport and transition to the **Idle** state

- *if the disconnect DP is armed as EDP_R for the B-Side and actually the B-Side disconnects from the call, send an EventReportBCSM, then transition to the 'Waiting for Instruction' state. If there is a Call Information Request or a request to send ApplyCharging pending, a CallInformationReport and/or a ApplyChargingReport will be sent immediately before the EventReportBCSM operation.*

- If the Disconnect DP is armed as an EDP-N and neither Call Information Request nor a request to send ApplyChargingReport pending, send an EventReportBCSM , then transition to the **Idle** state

- If the Disconnect DP is armed as an EDP-N and there is a Call Information Request or a request to send ApplyChargingReport pending, send all outstanding reports, then transition to the **Idle** state

The SSF has an application timer, T_{SSF}, whose purpose is to prevent excessive call suspension time and to guard the association between the SSF and the SCF.

Timer T_{SSF} is set in the following cases:

- when the SSF sends an InitialDP (Ref. section 7.1.5.3 State c: "Waiting for Instructions") or AssistRequestInstructions operation (Ref. section 7.1.6.2 State b:"Waiting for Instructions" in assisting case). While waiting for the first response from the SCF, the timer T_{SSF} can be reset only once by a Reset Timer operation. Subsequent to the first response, the timer can be reset any number of times.

- when the SSF enters the "Waiting for instructions" state (Ref. section 7.1.5.3) under any other condition as the ones listed in the previous case. In this case the SCF may reset the T_{SSF} timer using the Reset operation any number of times.

- when the SSF enters the "Waiting for End of User Interaction" state or the "Waiting for End of Temporary Connection" state (Ref. section 7.1.5.4 and section 7.1.5.5). In these cases the SCF may reset T_{SSF} using the Reset Timer operation any number of times.

In the three above cases T_{SSF} may respectively have three different values as defined by the application.

The SSF should cancel TSSF after entering the 'Waiting for Instruction' state, when a correct operation sequence has been received from the SCF.

Beeing in the 'Wait for End of UI' state, the SSF should cancel TSSF after leaving this state either due to receipt of a DisconnectForwardConnection operation or due to SRF initiated disconnection (Ref. section 7.1.5.4).

Beeing in the 'Waiting for End of Temporary Connection' state, the SSF should cancel TSSF after leaving this state either due to receipt of a DisconnectForwardConnection operation or due to an disconnection initiated by the assisting SSF (Ref. section 7.1.5.5).

On expiration of T_{SSF} the SSF FSM transitions to the Idle state, aborts the interaction with the SCF and the CCF progresses the BCSM if possible.

The SSF can receive an ActivityTest operation in any state sent by the SCF to check the continued existence of a relationship. If there is still a call associated relationship between CCF/SSF/SCF a RETURN_RESULT will be sent back to the SCF, otherwise a TC_U_ABORT or TC_P_ABORT (Ref. section 7.2).

The SSF State diagram contains the following transitions (events):

e1. TDP encountered
e2. Trigger fail
e4. Trigger detected
e5. User Interaction requested
e6. User Interaction ended
e7. Temporary connection created
e8. Temporary connection ended

e9. Idle return from Wait for Instruction
e9'. Idle return from Waiting for End of User Interaction
e9". Idle return from Waiting for End of Temporary Connection
e10. EDP_R encountered
e11. Routing instruction received
e12. EDP_N last encountered
e13. Waiting for End of User Interaction state no change
e14. Waiting for Instruction state no change
e16. Monitoring state no change
e17. Abandon (from any state) (not shown in the SSF state diagram)
e18. Disconnect (from any state) (not shown in the SSF state diagram)
e19. Non call associated treatment from any state (not shown in the SSF state diagram)

The SSF State diagram contains the following states:

State a. Idle

State b. Trigger Processing

State c. Waiting for Instructions

State d. Waiting for End of User Interaction

State e. Waiting for End of Temporary Connection

State f. Monitoring

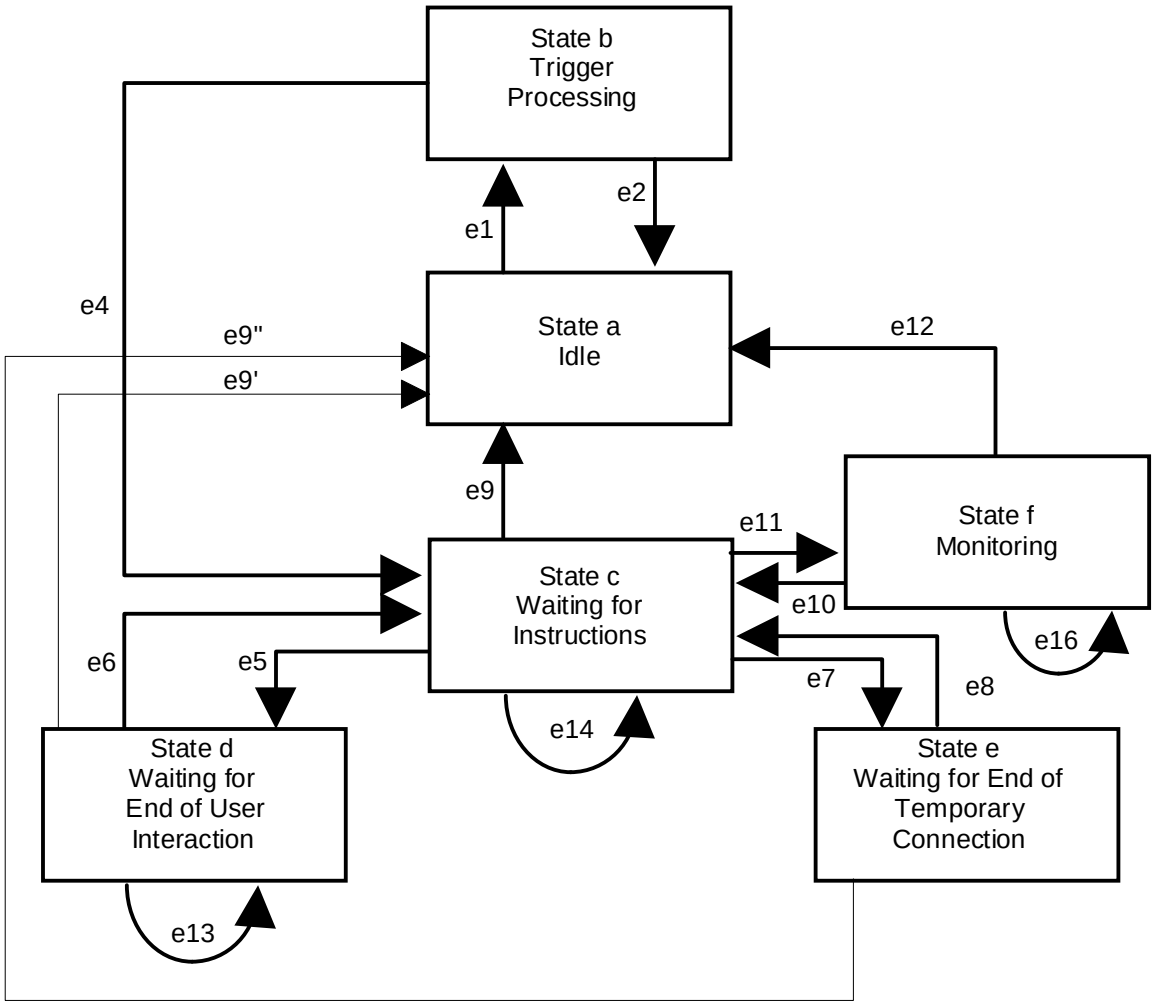


Figure 12: SSF Finite State Model

7.1.5.1 State a:"Idle"

The SSF FSM enters the **Idle** state under a variety of conditions, as described below.

The SSF FSM enters the **Idle** State when sending or receiving an ABORT TCAP primitive in any state.

The SSF FSM enters the **Idle** State when DP processing fails in the **Trigger Processing** state (transition e2).

The SSF FSM enters the **Idle** State when one of the following occurs :

- when the call is abandoned or one or more call parties disconnect in any other state under the conditions identified in 7.1.5 .;
- when a Connect, or Proceed Call Processing operation is processed in the **Waiting for Instructions** state, and no EDPs are armed and there are no outstanding report requests (transition e9);
- when the application timer T_{SSF} expires in the state: **Waiting for Instructions** (transition e9);
- *when the application timer TSSF expires in the state: Waiting for End of User Interaction (e9');*
- *when the application timer TSSF expires in the state: Waiting for End of Temporary Connection (e9'');*
- when a ReleaseCall operation is processed in **Waiting for Instructions** (transition e9) or **Monitoring** (transition e12);
- *when a ReleaseCall operation has been received in 'Waiting for End of UI' this operation will be saved by the SSF until SRF initiated disconnection.*
- when the last EDP-N is encountered in the **Monitoring** State, and there are no EDP-Rs armed (transition e12);

In certain situations when transitioning to the **Idle** state, if there is a Call Information Request pending (see Note 1 section 7.1.5), the SSF sends a CallInformationReport operation to the SCF before returning to **Idle** *(Refer section 9.7)*.

During this State the following call-associated events can occur:

- indication from the CCF that an armed TDP is encountered related to a possible IN call/service attempt: in this case the SSF FSM moves to the State **Trigger Processing** (transition e1).

Any other operation received from the SCF while the SSF is in Idle State should be treated as an error. The event should be reported to the maintenance functions and the transaction should be aborted according to the procedure specified in TCAP (Recommendation Q.774).

7.1.5.2 State b: "Trigger Processing"

Following a trigger detection related to an armed TDP in the BCSM, the SSF FSM is activated and moves from the **Idle** State to the **Trigger Processing** State (transition e1).

In this State, the SSF/CCF should:

- perform the DP processing actions specified in Q.1214, section 4.2.2.5
  - * check if call gapping or service filtering mechanisms are active;
  - * determine if DP criteria are met;
  - * handle service feature interactions;
- collect and verify necessary parameters for sending a InitialDetectionPoint to the SCF:
  - * if successful and the DP is a TDP-R, send a generic InitialDetectionPoint to the SCF, as determined from DP processing, and transition to the **Waiting for Instructions** state (transition e4);
  - * if DP processing fails, return to the **Idle** state (transition e2). DP processing fails in the following cases:
    - if CallGapping is in effect: the SSF FSM will instruct the CCF to terminate the call with the appropriate treatment;
    - if Service Filtering is in effect: the call is counted (if required) and the SSF FSM instructs the CCF to terminate the call with the appropriate treatment;
    - if a trigger criteria match is not found: the SSF FSM returns call control to the CCF;
    - if the Call is abandoned: the SSF returns call control to the CCF and continues processing as described in section 7.1.5;

7.1.5.3 State c: "Waiting for Instructions"

This State is entered from either the **Trigger Processing** State, as indicated above (transition e4), or from the state **Monitoring** on detection of an EDP-R (transition e10).

In this State the SSF FSM is waiting for an instruction from the SCF; call handling is suspended and an application timer (T_{SSF}) should be set on entering this State.

During this State the following events can occur:

- The user dials additional digits (applies for open-ended numbering plans): the CCF should store the additional digits dialled by the user .
- The user abandons or disconnects. This should be processed in accordance with the general rules in section 7.1.5.
- The application Timer T_{SSF} expires: the SSF FSM moves to the **Idle** State, the CCF routes the call if possible (e.g. default routing to a terminating announcement), the T_{SSF} expiration is reported to the maintenance functions and the transaction is aborted.
- An operation is received from the SCF: The SSF FSM acts according to the operation received as described below.

The following operations may be received from the SCF and processed by the SSF with no resulting transition to a different state (transition e14):

RequestReportBCSMEvent;
ResetTimer;
FurnishChargingInformation;
ApplyCharging;
CallInformationRequest;
SendChargingInformation

Before leaving the state 'Waiting for Instruction' due to receipt of Connect, Continue, ConnectToResource or EstablishTemporaryConnection (see below) only one FCI and one SCI charging operation will be accepted.

Before leaving the state 'Waiting for Instruction' due to receipt of ConnectToResource or EstablishTemporaryConnection (see below) no CallInformationRequest and no RequestReportBCSMEvent operation will be accepted.

Before leaving the state 'Waiting for Instruction' due to receipt of CollectInformation operation (see below) only a RequestReportBCSMEvent requiring the arming of DP2 as EDP_R will be accepted.

The following operations may be received from the SCF and processed by the SSF, causing a state transition to **Waiting for End of User Interaction** state (transition e5):

ConnectToResource

The following operations may be received from the SCF and processed by the SSF, causing a state transition to **Waiting for End of Temporary Connection** state (transition e7):

EstablishTemporaryConnection

The following operations may be received from the SCF and processed by the SSF, causing a state transition to either **Monitoring** state (if any EDPs were armed or any reports were requested) (transition e11) or **Idle** State (transition e9):

Connect;
CollectInformation;
Continue.

The CollectInformation operation will be processed by the SSF one or more times only before a Connect, Continue, ConnectToResource or an EstablishTemporaryConnection operation has been received.

The **ReleaseCall** operation may be received from SCF and processed by the SSF, causing a transition to **Idle** state (transition e9). *Before transition into the 'idle' state after receipt of a ReleaseCall operation only a FurnishChargingInformation operation will be accepted by the SSF.*

When processing the above operations, any necessary call handling information is provided to the Call Control Function (CCF).

Any other operation received in this state should be processed in accordance with the general rules in section 7.1.5.

7.1.5.4 State d: "Waiting for End of User Interaction"

The SSF enters this State from the **Waiting for Instructions** State (transition e5) on the reception of one of the following operations:

ConnectToResource

During this State the following events can occur:

- A valid SCF-SRF operation (i.e.., Play Announcement *and* Prompt & Collect User Information for relaying is received and is correct, the operation is transferred to the SRF for execution. The SSF FSM remains in the **Waiting for End of User Interaction** State (transition e13).

- A valid SRF-SCF operation (i.e., Specialized Resource Report and Return Result from Prompt and Collect user Information) for relaying is received and is correct, the operation is transferred to the SCF. The SSF FSM remains in the Waiting for End of User Interaction State (transition e13).

- The application timer T_{SSF} expires: the SSF FSM moves to the **Idle** State, the CCF routes the call if possible (e.g. default routing to a terminating announcement), the T_{SSF} expiration is reported to the maintenance functions and the transaction is aborted.

- An operation is received from the SCF: The SSF FSM acts according to the operation received as described below.

- The user abandons. This should be processed in accordance with the general rules in section 7.1.5.

The following operations may be received from the SCF and processed by the SSF with no resulting transition to a different state (transition e13):

ResetTimer;
ApplyCharging;
Send Charging Information.

The **DisconnectForwardConnection** operation may be received from the SCF and processed by the SSF in this state. Call disconnect can also be received from the SRF. In both cases this causes the release of the connection to the SRF and the transition to the **Waiting for Instructions** state. The disconnection is not transferred to the other party.(transition e6).

The ReleaseCall operation may be received from the SCF and will be saved by the SSF until end of user interaction initiated by the SRF.

Any other operation received in this state should be processed in accordance with the general rules in section 7.1.5.

7.1.5.5 State e: "Waiting for End of Temporary Connection"

The SSF enters this State from the **Waiting for Instructions** State (transition e7) upon receiving an EstablishTemporaryConnection operation.

The call is routed to the assisting SSF and call handling is suspended while waiting for the end of the assisting procedure. The timer T_{SSF} is active in this State.

During this State the following events can occur:

- The application Timer T_{SSF} expires: the SSF FSM moves to the **Idle** State, the CCF routes the call if possible (e.g. default routing to a terminating announcement), the T_{SSF} expiration is reported to the maintenance functions and the transaction is aborted.

- The receipt of an indication of disconnection of forward connection from the CCF. In this case, the SSF moves to the **Waiting for Instructions** state (transition e8). The disconnection is not transferred to the Calling party.

- The user abandons. This should be processed in accordance with the general rules in section 7.1.5

- An operation is received from the SCF; The SSF acts according to the operation received as described below.

The following operations can be received from the SCF and processed by the SSF with no resulting transition to a different state (transition e15):

ResetTimer,
ApplyCharging,
SendChargingInformation.

The **DisconnectForwardConnection** operation may be received from the SCF and processed by the SSF in this state. *This* causes the release of the connection to the *assisting SSF* and the transition to the **Waiting for Instructions** state. The disconnection is not transferred to the other party.(transition e8).

Any other operation received in this state should be processed in accordance with the general rules in section 7.1.5/3.1.1.5.

7.1.5.6 State f: "Monitoring"

The SSF enters this State from the **Wait for instructions** State (transition e11) upon receiving a Connect, Continue *or* CollectInformation operation when one or more EDPs are armed or/and there is Call Information Request pending and/or there is a call supervision due to a received ApplyCharging.

In this state the timer T_{SSF} is not used; i.e., the expiration of T_{SSF} does not have any impact on the SSF FSM.

During this State the following events can occur:

- An EDP-N should be reported to the SCF by sending an Event ReportBCSM operation ; the SSF FSM should remain in the **Monitoring** State (transition e16) if one or more EDPs are armed or there is a Call Information Request pending. The SSF FSM should move to the **Idle** state (transition e12) if there are no remaining EDPs armed or there is no Call Information Request pending. *A possibly outstanding CallInformationReport operation will be sent immediately after EventReportBCSM for all detected DPs, which result in termination of the call.*

- An EDP-R should be reported to the SCF by sending an Event Report BCSM operation ; the SSF FSM should move to the Wait for instructions State (transition e10). *A possibly outstanding CallInformationReport operation will be sent immediately before the EventReportBCSM operation.*

- The receipt of an END or ABORT primitive from TCAP has no effect on *a stable* call; . In this case, the SSF FSM transitions to the **Idle** state (transition e12), disassociating the SSF FSM from the call.

- The user abandons or disconnects. This should be processed in accordance with the general rules in section 7.1.5.

- *A charging event that was requested by the SCF via a ApplyCharging operation has been detected. This event should be reported to the SCF by sending an ApplyChargingReport operation. It causes no FSM state transition.*

The following operations can be received from the SCF and processed by the SSF with no resulting transition to a different state (transition e16):

- **ApplyCharging,**
- **SendChargingInformation.**

The ReleaseCall operation may be received from the SCF and processed by the SSF, causing a state transition to the **Idle** State (transition e12). If there is a Call Information Request pending and/or call supervision is active due to a received ApplyCharging, the SSF sends the corresponding outstanding report(s) the SCF.

Any other operation received in this state should be processed in accordance with the general rules in section 7.1.5.

7.1.6 Assisting SSF FSM

The present subclause describes the SSF FSM related to the Assisting SSF. The Assisting SSF is structured as defined in subclauses 7.1.1 through 7.1.5 of this ETS.

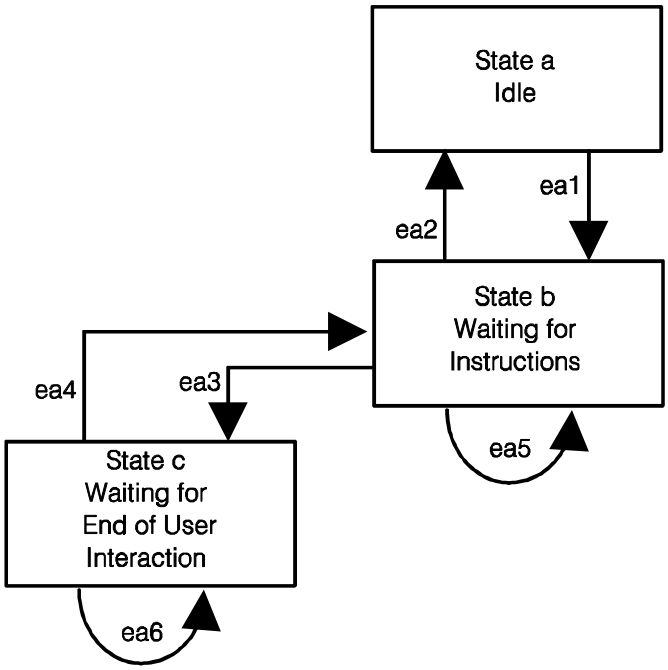


Figure 13: Assisting SSF FSM state diagram

The Assisting-off SSF state diagram contains the following transitions (events):

ea1: Assist detected
ea2: Assist ended (fail or success)
ea3: User Interaction requested
ea4: User Interaction ended
ea5: Waiting for Instruction state no change
ea6: Waiting for End of User Interaction state no change

The Assisting SSF state diagram contains the following states:

State a: Idle
State b: Waiting for Instructions
State c: Waiting for End of User Interaction

7.1.6.1 State a: "Idle"

The SSF FSM enters the **Idle** state when one of the following occurs:

- when sending or receiving an ABORT TCAP primitive due to abnormal conditions in any state;

- given a temporary connection between an upstream SSF and the Assisting SSF, when a bearer channel disconnect is received from the upstream SSF; (transition ea2).

Once in the **Idle** state, if there are any outstanding responses to send to the SCF, they are discarded by the Assisting SSF.

The Assisting SSF FSM transitions from the **Idle** state to the **Waiting for Instructions** state on receipt of an assist indication at the assisting SSF from another SSF (transition ea1).

Any operation received from the SCF while the Assisting SSF is in Idle state should be treated as an error. The event should be reported to the maintenance functions and the transaction should be aborted according to the procedure specified in TCAP, see ETS 300 287 (ITU-T Recommendation Q.774).

7.1.6.2 State b: "Waiting for Instructions"

This state is entered from the **Idle** state on receipt of a connect at an SSF from another SSF indicating that an assist is required, based on an implementation dependent detection mechanism(transition ea1).

In this state the SSF sends an AssistRequestInstructions operation to the SCF and the Assisting SSF FSM is waiting for an instruction from the SCF; call handling is suspended and an application timer (T_{SSF}) should be set on entering this state.

During this state the following events can occur:

- the application Timer T_{SSF} expires: the Assisting SSF FSM moves to the **Idle** state (transition ea2) and the expiration is reported to the maintenance functions and the transaction is aborted;

- an operation is received from the SCF: The SSF FSM acts according to the operation received as described below;

- a bearer channel disconnect is received and the FSM moves to the **Idle** state (transition ea2).

The following operations may be received from the SCF and processed by the Assisting SSF with no resulting transition to a different state (transition ea5):

- **ResetTimer;**

The following operations can be received from the SCF and processed by the Assisting SSF, causing a state transition to **Waiting for End of User Interaction** state (transition ea3):

- **ConnectToResource.**

Any other operation received in this state should be processed in accordance with the general rules in subclause 7.1.5.

7.1.6.3 State c: "Waiting for End of User Interaction"

The Assisting SSF enters this state from the **Waiting for Instructions** state (transition ea3) on the reception of one of the following operations:

- **ConnectToResource.**

During this state the following events can occur:

- a valid SCF-SRF operation (i.e., PlayAnnouncement, PromptAndCollectUserInformation for relaying is received and is correct, the operation is transferred to the SRF for execution. The SSF FSM remains in the **Waiting for End of User Interaction** state (transition ea6);

- a valid SRF-SCF operation (i.e., SpecializedResourceReport and return result from PromptAndCollectUserInformation) for relaying is received and is correct, the operation is transferred to the SCF. The SSF FSM remains in the **Waiting for End of User Interaction** state (transition ea6);

- the application Timer T_{SSF} expires: the SSF FSM moves to the **Idle** state, the CCF routes the call if possible (e.g., default routing to a terminating announcement), the T_{SSF} expiration is reported to the maintenance functions and the transaction is aborted;

- an operation is received from the SCF: The SSF FSM acts according to the operation received as described below;

- a bearer channel disconnect is received from the initiating SSF. The assisting SSF FSM moves to the **Idle** state, the connection to the SRF is released and the transaction is aborted.

The following operations can be received from the SCF and processed by the SSF with no resulting transition to a different state (transition ea6):

- **ResetTimer.**

The **DisconnectForwardConnection** operation may be received from the SCF and processed by the Assisting SSF in this state, causing a transition to the **Waiting for Instructions** state (transition ea4). This procedure is only valid if a ConnectToResource was previously processed to cause a transition into the **Waiting for End of User Interaction** state.

Any other operation received in this state should be processed in accordance with the general rules in subclause 7.1.5.

7.2 SCF Application Entity Procedures

7.2.1 General

This section provides the definition of the SCF Application Entity (AE) procedures related to the SCF-SSF/SRF/SDF interface. The procedures are based on the use of Signalling System No.7 (SS #7).

In addition, other capabilities may be supported in an implementation-dependent manner in the SCP, AD or SN.

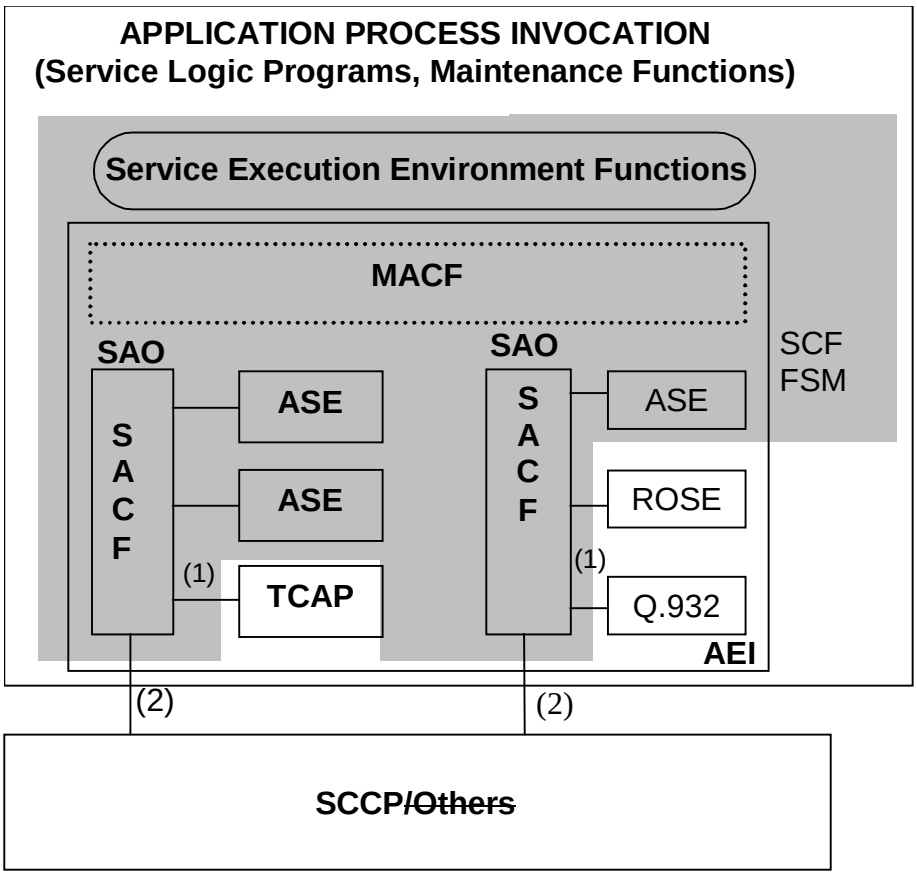
The AE, following the architecture defined in CCITT Recommendations Q.700, Q.771 and Q.1400, includes TCAP (Transaction Capabilities Application Part) and one or more ASEs called TC-users. The following sections define the TC-user ASE and SACF & MACF rules, which interface with TCAP using the primitives specified in CCITT Recommendation Q.771.

The procedure may equally be used with other message-based signalling systems supporting the Application Layer structures defined. By no means is this text intended to dictate any limitations to service logic programs.

In case interpretations for the application entity procedures defined in the following differ from detailed procedures and the rules for using of TCAP service, the statements and rules contained in the detailed clause 9 and 10 shall be followed.

7.2.2 Model and Interfaces

The functional model of the AE-SCF is shown in figure 14; the ASEs interface with supporting protocol layers to communicate with the SSF, SRF and SDF, and interface to the Service Logic Programs and maintenance functions. The scope of this Recommendation is limited to the shaded area in figure 14.



(1) TC-Primitives

(2) N-Primitives

NOTE: The SRF FSM includes several Finite State Machines.

Abschnitt 7.2.2

AEI: Application Entity Invocation
SCF: Service Control Functions
FSM: Finite State Model
MACF: Multiple Association Control Function
SACF: Single Association Control Function
SAO: Single Association Object

Note that SCF FSM includes several Finite State Machines.

Figure 14: Functional Model of SCF AE

The interfaces shown in figure 14 use the TC-user ASE primitives specified in CCITT Recommendation Q.771 (interface (1)) and N-primitives specified in CCITT Recommendation Q.711 (interface (2)). The operations and parameters of Intelligent Network Application Protocol (INAP) are defined in section 6 of this Recommendation.

7.2.3 Relationship between the SCF FSM and SLPs/Maintenance Functions

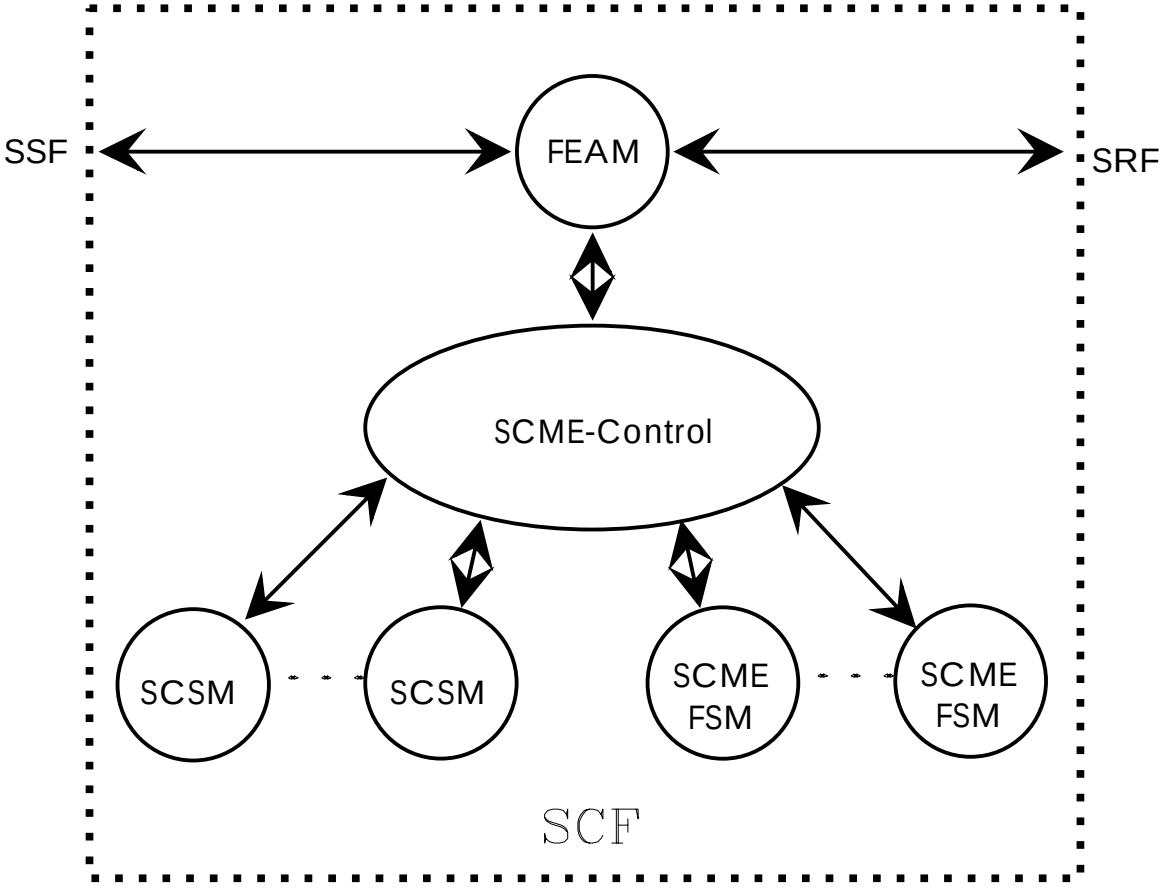
The primitive interface between the SCF FSM and the Service Logic Programs/maintenance functions is an internal interface and is not a subject for standardization in CS1.

The relationship between the Service Logic Program and the SCF FSM may be described as follows :

- If a request for IN call processing is received from the SSF, an instance of an SCF State Model (SCSM) is created, and the relevant Service Logic Program is invoked.

In *this* case, the SCF FSM handles the interaction with the SSF FSM (and the SRF FSM and SDF FSM) as required, and notifies the Service Logic Program of events as needed.

The management functions related to the execution of operations received from the SCF are executed by the SCF Management Entity (SCME). The SCME is comprised of the SCME-Control and multiple instances of SCME FSMs. The SCME-Control interfaces different SCF Call State Models (SCSMs) and the Functional Entity Access Manager (FEAM). figure 15 shows the SCF FSM structure.



SCME: SCF Management Entity
SCSM: SCF Call state Model
FEAM: Functional Entity Access Manager

Figure 15: SCF FSM Structure

The following text systematically describes the procedural aspects of the interface between the SCF and other functional entities, with the main goal of specifying the proper order of operations rather than entities' functional capabilities. Consequently, this text describes only a subset of the SCF functional capabilities.

The procedural model associates an SCSM with each query from the SSF. The SCSM maintains dialogues with the SSF, SRF, and SDF on behalf of service logic.

Multiple requests may be executed concurrently and asynchronously by the SCF, which explains the need for a single entity that performs the tasks of creation, invocation, and maintenance of the SCF FSM objects. This entity is called the SCF Management Entity-Control (SCME-Control). In addition to the above tasks, the SCME maintains the dialogues with the SSF, SDF, and SRF on behalf of all instances of the SCF FSMs. In particular, the SCME-Control:

1. Interprets the input messages from other FEs and translates them into corresponding SCSM events;

2. Translates the SCSM outputs into corresponding messages to other FEs;

3. Performs some asynchronous (with call processing) activities (One such activity is flow control). It is the SCME-control's responsibility to detect nodal overload and send the Overload Indication (e.g., Automatic Call Gap) to the SSF to place flow control on queries. Other such activities include non-call associated treatment due to changes in Service Filtering *or* Call Gapping; and

4. Supports persistent interactions between the SCF and other FEs.

5. Captures asynchronous (with call processing) activities related to management and supervisory functions in the SCF and creates an instance of a SCME FSM. For example, the SCME provides the non-call associated treatment due to changes in Service Filtering. Therefore, the SCME-Control separates the SCSM from the Service Filtering by creating instances of SCME FSMs for each context of related operations.

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

Finally, the Functional Entity Access Manager (FEAM) relieves the SCME of low-level interface functions. The FEAM's functions include:

1. Establishing and maintaining the interfaces to the SSF, SRF and SDF;

2. Passing (and queueing when necessary) the messages received from the SSF, SRF, and SDF to the SCME; and

3. Formatting, queueing (when necessary), and sending messages received from the SCME to the SSF, SRF, and SDF.

7.2.4 Partial SCF Management Entity (SCME) State Transition Diagram

The key part of SCF Management Entity (SCME) State Diagram is described in figure 16.

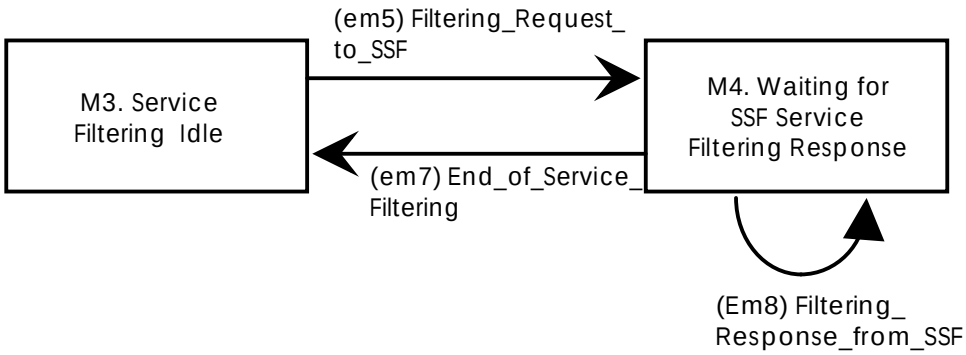


Figure 16: The Service Filtering FSM in the SCME

The SCME handles the following operations:

- **Activate Service Filtering;**
- **Service Filtering Report;**
- **Call Gap; and**
- **Activity Test.**

The issuing of the **Call Gap** and **Activity Test** operations *are processed* in the SCME. The procedures for the rest of the above operations are described below.

The ReturnResult for a successful ActivityTest or the timeout or a received P/U-ABORT for this operation is processed in the SCME.

The operations that are not listed above do not affect the state of the SCME; these operations are passed to the relevant SCSM.

7.2.4.3 State M3: "Service Filtering Idle"

The following event[1] is considered in this state:

- (em5) Filtering_Request_to_SSF: This is an internal event, caused by service logic's need to filter service requests to the SSF, and by transmission of the **Activate Service Filtering** operation. This event causes a transition to State M4, **Waiting for SSF Service Filtering Response**.

7.2.4.4 State M4: "Waiting for SSF Service Filtering Response"

In this state, the SCF is waiting for the service filtering response from the SSF. The following events are considered in this state:

- (em7) End_of_Service_Filtering: This is an internal event, caused by the expiration of service filtering duration timer in the SCF. This event causes a transition to state M3, **Service Filtering Idle** *when the last ServiceFilteringResponse is received*.

- (Em8) Filtering_Response_from_SSF: This is an external event, caused by reception of the response to the **Request Service Filtering** operation previously issued to the SSF. *The ReturnResult for sent ActivateServiceFiltering operations related to the same service filtering relationship is also received*

[1] All events are enumerated, and the number of an event is prefixed with either the letter "E" (for external events) or "e" (for internal ones) and included in parentheses in the beginning of the event name.

in this state. This event does not cause a transition out of this state, and the SCSM remains in state M4, **Waiting for SSF Service Filtering Response**.

When Service Filtering is active, another **Service Filtering** operation could be sent to the SSF that has the same filtering criteria; this second "filter" replaces the first one.

In case of a received error indication for a ActivateServiceFiltering a retransmission is initiated.

If further ActivateServiceFiltering operations whith the same filtering criteria are not successful, the last activated filtering data remain valid.

7.2.5 The SCSM

Figure 17 shows the general State Diagram of the SCSM 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 sub-sections. Each state, except **Idle**, **SDF Request Idle** and **Waiting for SDF Response**, has internal sub-FSMs composed of the sub-states.

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

In every state, if there is an error in a received operation, the service logic program and the maintenance functions are informed. *Different* error treatment is possible in specific cases as described in section 8. Depending on the class of the operation, the error can be reported to the SSF, SRF, or SDF (see Recommendation Q.774).

It also holds that, in every state, if the SCSM is informed that the dialogue with the SSF is terminated, then it informs the service logic program and returns to the Idle state. In this case, all resources allocated to that call, including those required for relevant dialogues with the other functions, must be de-allocated. To simplify the diagram, such transitions are not demonstrated in the figures.

When the Service Logic Program requests call information, the SCSM transmits the **Call Information Request** operation to the SSF, and then **Call Information Report** is outstanding.

In *the* State*s ´Waiting for Notification or Request ´ and ´User Interaction´* , the SCSM may receive the **Call Information Report** operation from the SSF, when the **Call Information Report** is outstanding.

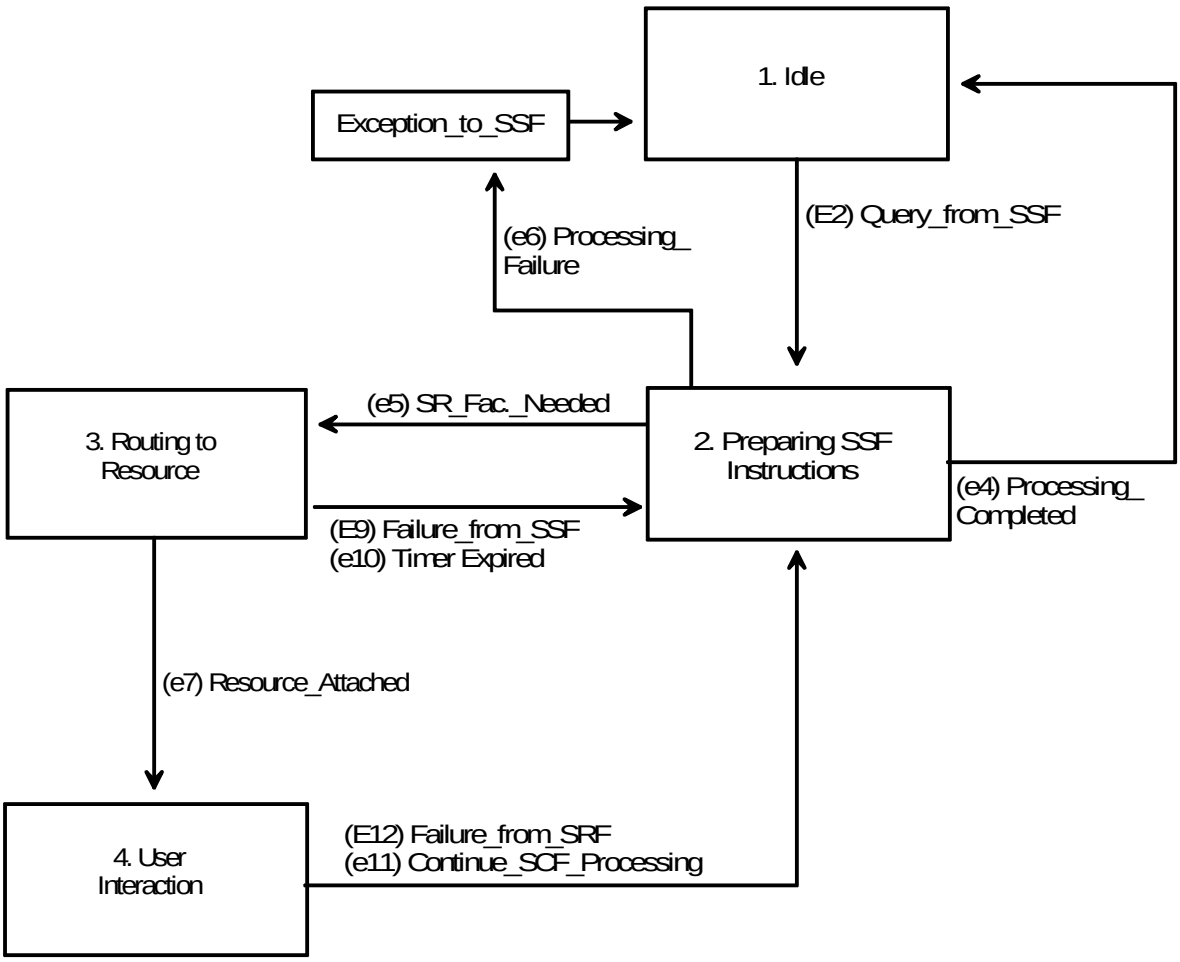


Figure 17: The SCSM Finite State Model

From any State (except **Idle**), if **Call Information Report** is outstanding and the Service Logic Program indicates that the processing has been completed, the SCSM remains in the same state until it receives the **Call Information Report** operation.

The general rules for one or a sequence of components sent in one or more TCAP messages, which may include a single operation or multiple operations, are specified in section 7.1.5 SSF State Transition Diagram(they are not described here).

In each state, where an external event is received, the SLPI is informed. Whether the state is to be changed or not is controlled by the SCSM itself.

The SCSM has an application timer, $T_{SCF-SSF}$, whose purpose is to reset the timer T_{SSF}, which is used to prevent excessive call suspension time and to guard the association between the SSF and the SCF.

Timer $T_{SCF-SSF}$ is set in the following cases:

- when the SCF receives an **Initial DP** or **Assist Request Instructions** operation.(See section 7.2.5.2.1 State 2.1: "**Preparing SSF Instructions**", and section 7.2.5.2.2.1 State 2.2.1: "**Preparing SSF Instructions**"). In this case, this timer is reset when the first request, other than **Reset Timer** operation, is sent to the SSF. On the expiration of $T_{SCF-SSF}$, the SCF may reset T_{SSF} once, using the **Reset Timer** operation, and also reset $T_{SCF-SSF}$. On second expiration of $T_{SCF-SSF}$, the SCSM informs Service Logic Program and the maintenance functions, and the SCSM transits to the **Idle** state;

- when the SCF enters the "**Waiting for Assist Request Instructions**" state or the "**User Interaction**" state (See section 7.2.5.3.2 and section 7.2.5.4). In these cases, on the expiration of $T_{SCF-SSF}$, the SCF may reset T_{SSF} using the **Reset Timer** operation any number of times.

In *these two* cases, $T_{SCF-SSF}$ may respectively have two different values as defined by the application. The value of $T_{SCF-SSF}$ are smaller than the respective value of T_{SSF}.

When receiving or sending any other operation, the SCF should reset $T_{SCF-SSF}$. In the "**Waiting for Notification or Request**" state (See _section 7.2.5.2.3), $T_{SCF-SSF}$ is not used.

The SCSM also has an application timer, $T_{ASSIST/HAND-OFF}$, whose purpose is to prevent excessive assist/hand-off suspension time. The SCSM sets the timer $T_{ASSIST/HAND-OFF}$ when the SCSM sends the **Establish Temporary Connection** with a correlation ID operation. This timer is stopped when the SCSM receives the **Assist Request Instructions** operation from the assisting/handed-off SSF. On expiration of $T_{ASSIST/HAND-OFF}$, the SCSM informs Service Logic and the maintenance functions, and the SCSM transits to the **Preparing SSF instructions** state.

The call-control-related operations relevant to the SCF-SSFinterface (except the SCME related operations) are categorized into

Call-processing-related operations, and

Non-call-processing-related operations.

Call-processing-related operations are grouped into the following two sets

- CollectInformation
- Connect
- ReleaseCall

and

- ConnectToResource
- DisconnectForwardConnection
- EstablishTemporaryConnection.

For the first set of call-processing-related operations, the SCF may not send two operations of the same set in a series of TCAP messages or in a component sequence to the SSF, but send them only one at a time. Two operations of the first set shall be separated by at least one EDP-R message received by the SCSM.

Only the DisconnectForwardConnection of the second set may be followed by either a Connect or a ReleaseCall of the first set or a ConnectToResource or EstablishTemporaryConnection of the second set.

The non-call-processing operations include the rest of the operations at the SCF-SSF interface (but not the SCME related operations). When the service logic needs to send operations in parallel, they are sent in the component sequence.

Call-processing-related operations lead to a state transition in the SCSM (and the SSF-FSM and SRSM).

Non-call-processing-related operations do not lead to a state transition (neither in the SCF nor in the SSF or SRF).

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

7.2.5.1 State 1: "Idle"

The following events are considered in this state:

- (E2) Query_from_SSF: This is an external event, caused by a reception of one of the following operations:
  - **Initial DP;**
  - **Assist Request Instructions.**

Both events causea transition to State 2, **Preparing SSF Instructions**.

7.2.5.2 State 2: "Preparing SSF Instructions"

In this state, the SCF determines how to further process.

The following events are considered in this state:

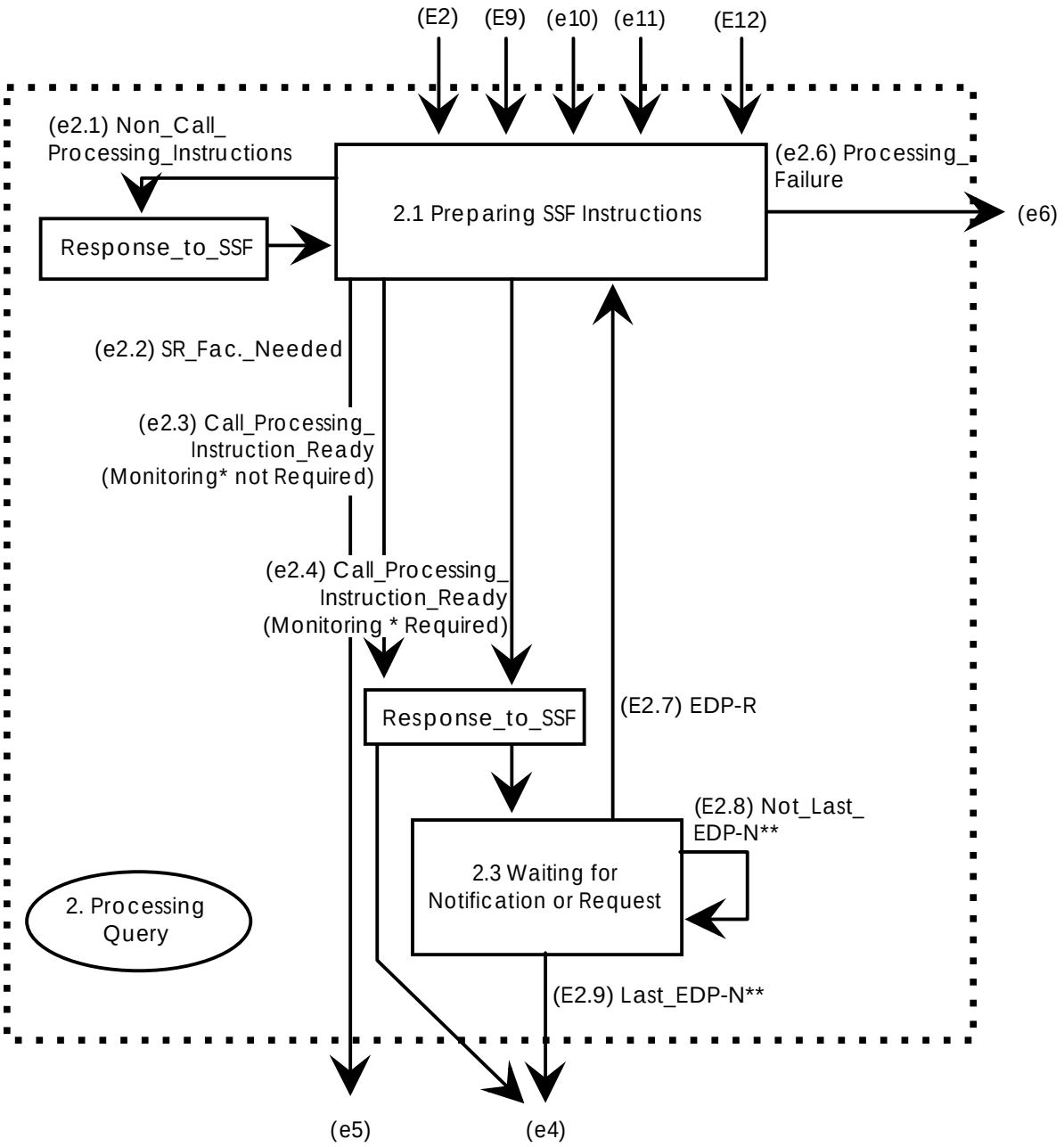
- (e4) Processing_Completed: This is an internal event. In this case, the SCF has completed the processing of the instructions to the SSF. This event causes a response to be sent to the SSF and a transition to State 1, **Idle**;

- (e5) SR_Facilities_Needed: This is an (internal) event caused by the service logic's need for additional information from the call party; hence is the necessity to set up a connection between the call party and the SRF. This event causes a transition to State 3, **Routing to Resource**. *In this case the SCF may have prepared already some non-call-processing instructions (e2.1).*;

- (e6) Processing_Failure: This (internal) event causes an appropriate exception processing[2] and a transition back to State 1, **Idle**.

To further describe the procedures relevant to this state, the state is divided into three sub-states, which are described in the following three subsections. (This subdivision is illustrated in figure 18)

Note [2]: Here and further in this document, the exception processing is not defined. It is assumed, however, that it must include releasing all the involved resources and sending an appropriate response message to the SSF.



*Note 1: Including Call Information Request

**Note 2: Including Call Information Report

Figure 18: Partial Expansion of the State 2.FSM

7.2.5.2.1 State 2.1: "Preparing SSF Instructions"

The State 2.1, **Preparing SSF Instructions**, is where the initial decision is made on whether the SDF information or a Specialized Resource is needed etc. In addition, the EDP-R-related processing is also performed in this state.

The following events are considered in this state:

- (e2.1) Non-Call_Processing_Instructions: This is an internal event caused by the service logic when there is a need to send such an operation to the SSF. It causes one or more of the following operations to be issued to the SSF:

  - **ApplyCharging;**
  - **Call Information Request;**
  - **Furnish Charging Information;**
  - **Request Report BCSM Event;**
  - **Send Charging Information.**

  This event causes a transition back to State 2.1, **Preparing SSF Instructions**.

- (e2.2) SR_Facilities_Needed: This is an internal event, caused by the service logic when there is a need to use the SRF. This event maps into the SCSM event (e5).
  Before establishing a connection to the SRF with a ConnectToResource:
  - *No EDP will be armed for a user interaction-phase, this means that no RequestReportBCSMEvent will be sent.*
  - *No CallInformationReport will be requested for a user interaction-phase, this means that no CallInformationRequest will be sent.*
  - *Only one charging operation causing a tarif change may be sent. This charging operation applies to the whole user interaction phase.*

- (e2.3) Call_Processing_Instruction_Ready (Monitoring[3] not required): This is an internal event caused by the service logic when the final call-processing-related operation is ready and there is no armed EDP and no outstanding **Call Information Report** or **ApplyChargingReport** operation. It causes one of the following operations to be issued to the SSF:

  - **Connect;**
  - **Continue;**
  - **Release Call.**

  Only one FCI and one SCI charging operation may be sent before Connect.
  Only one FCI operation my be sent before ReleaseCall.

  This event maps into the SCSM event (e4).

- (e2.4) Call_Processing_Instruction_Ready (Monitoring5 required): This is an internal event caused by the service logic when a call-processing-related operation is ready and the monitoring of the call is required (e.g., an EDP is set, or there is an outstanding **Call Information Report** or **ApplyChargingReport,** or there is a need to issue such a request). It causes one of the following operations to be issued to the SSF:

  - **Collect Information;**
  - **Connect;**
  - **Continue;**
  - **Release Call.**

  Before leaving the state 'Preparing SSF Instructions' in case a CollectInformation operation is to be issued, a RequestReportBCSMEvent requiring the arming of DP2 as EDP_R will be sent.

  Note: Detection points (EDP-R/N) or requests for notification (CallInformationReport) are armed only in the 'Preparing SSF Instructions' state.

Note [3]: Including Call Information Report

This event causes a transition into State 2.3, **Waiting for Notification or Request**.

- (e2.6) Processing_Failure: This is an internal event, and it maps into the SCSM event (e6) Processing_Failure.

7.2.5.2.3 State 2.3: "Waiting for Notification or Request"

In this state, the SCSM waits for a notification or a request from the SSF.

The following events are considered in this state:

- (E2.7) EDP-R: This is an external event, caused by a reception of the following operation:
  - **Event Report BCSM** (for EDP_R).

  This event causes a transition to State 2.1 **Preparing SSF Instructions**.

- (E2.8) Not_Last_EDP-N: This is an external event, caused by a reception of one of the following operations:

  - **ApplyChargingReport;**
  - **Call Information Report;**
  - **Event Report BCSM** (for EDP_N).

  In case a CallInformationReport is received as a Not_Last EDP-N, there is still an outstanding armed EDP-R.

  In case an EventReportBCSM is received as a Not_Last EDP-N, there is still an outstanding CallInformationReport.

  This event causes a transition to back to State 2.3 **Waiting for Notification or Request**.

- (E2.9) Last_EDP-N: This is an external event, caused by a reception of one of the following operations:

  - **ApplyChargingReport;**
  - **Call Information Report;**
  - **Event Report BCSM (for EDP_N).**

  In this case, there is no outstanding armed EDP[4]. This event maps into the SCSM event (e4).

This concludes the description of State 2 **Preparing SSF Instructions**.

7.2.5.3 State 3: "Routing to Resource"

The resource is any SRF facility (e.g., Intelligent Peripheral).

In this state, interactions with the SSF are necessary. Accordingly, the following events cause transitions out of this state:

- (e7) Resource_Attached: The SRF is available. This event causes a transition to State 4, **User Interaction**;

Note [4]: Including Call Information Report

- (E9) Failure_from_SSF: The inability of the SSF to connect to requested resources causes a transition to State 2, **Preparing SSF Instructions**; and

- (e10) Timer_Expired: This event takes place when TASSIST/HAND-OFF expires. This event causes a transition to State 2, **Preparing SSF Instructions**.

To further describe the procedures relevant to this state, the state is divided into two sub-state, which are described in the following two subsections. (This subdivision is illustrated in figure 20)

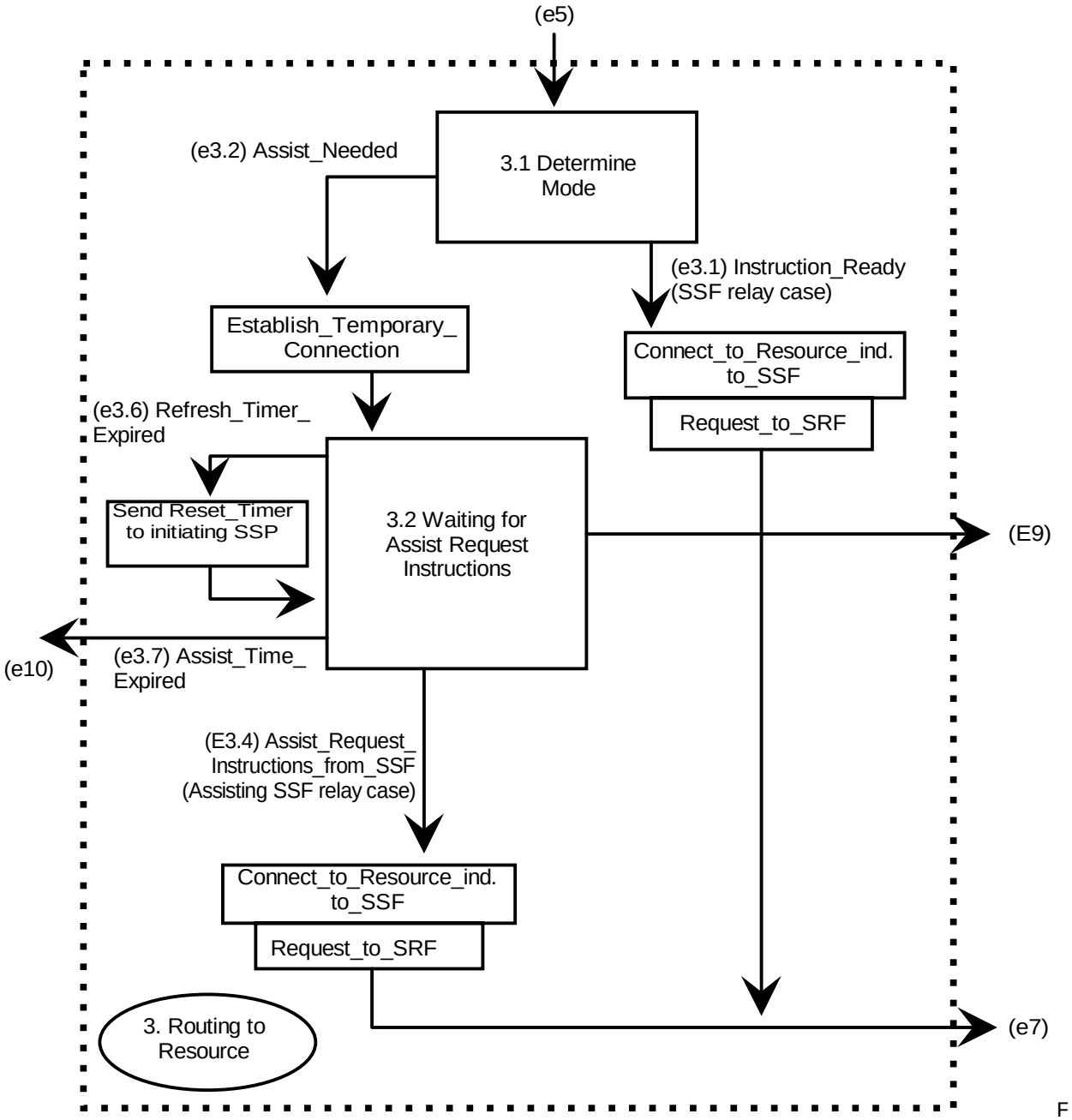


Figure 20: The State 3 FSM

7.2.5.3.1 State 3.1: "Determine Mode"

In this state, the SCSM determines the User Interaction mode to connect the call to SRF. The following events are considered in this state:

- (e3.1) Instruction_Ready: This is an internal event that takes place only when the Initiating SSF relay case. In this case, the SCSM sends the **Connect to Resource** operation accompanied by **Play Announcement** or **Prompt & Collect User Information** operation to the Initiating SSF, and transits out to the State 4 **User Interaction**. *For the first PlayAnnouncement or Prompt&CollectUserInformation SRF-initiated disconnect may be already allowed.*

  Note: For a PlayAnnouncement a SpecializedResourceReport will always be requested (Exception: Final Announcement).

  This transition maps into the event (e7);

- (e3.2) Assist_Needed: This event is an internal event that takes place when the Assisting SSF is needed . In this case, the SCSM sends the **Establish Temporary Connection** operation to the Initiating SSF with the Assisting SSF address or SRF address, and transits to the State 3.2 **Waiting for Assist Request Instructions**;

7.2.5.3.2 State 3.2: "Waiting for Assist Request Instructions"

In this state, the SCSM waits for the **AssistRequestInstructions** operation from the Assisting SSF (SSF relay case). On entering this state the SCSM starts the Timer $T_{ASSIST/HAND-OFF}$, and resets the timer $T_{SCF-SSF}$. The following events are considered in this state:

- (E3.4) Assist_Request_Instructions_from_SSF (Assisting SSF relay case): This is an external event caused by the receipt of **AssistRequestInstructions** from the Assisting SSF. In this case, the SCSM transmits the **ConnectToResource** operation accompanied by **PlayAnnouncement** or **PromptAndCollectUserInformation** operation to the Assisting SSF, and transits out to the State 4, **User Interaction**. This transition maps into the event (e7);

- (e3.6) Refresh_Timer_Expired: This is an internal event that takes place on the expiration of $T_{SCF-SSF}$. In this case, the SCSM transmits the **ResetTimer** operation to the Initiating SSF, and transits back to the same state;

- (e3.7) Assist_Timer_Expired: This is an internal event that takes place on the expiration of $T_{ASSIST/HAND-OFF}$. In this case, the SCSM informs the SCME and SLP, and transits to the State 2 **Preparing SSF Instructions**. This event maps into the event (e10); and

- (E3.8) Initial_SSF_Failure: This is an external event caused by the reception of SSF failure. This event causes a transition that maps into the SCSM event (E9).

7.2.5.4 State 4: "User Interaction"

In this state, the SCF requests the SRF to provide user interaction (e.g., collect additional information and/or play announcements). When an interaction is finished the SCF can instruct the SSF to disconnect the bearer between SSF and SRF. Alternatively, it can send a user interaction operation to the SRF containing an indication that allows the SRF initiated disconnect.

On entering this state the SCSM resets the timer $T_{SCF-SSF}$.

The following events cause transitions out of this state:

- (e11) Continue_SCF_Processing: In this case, the SCF has obtained all the information from the SRF, that is needed to instruct the SSF to complete the call. This event causes a transition to State 2, **Preparing SSF Instructions**.

- (E12) Failure_from_SRF: In this case, the SRF has detected that
  - The chosen resource cannot perform its function.

Accordingly, this event causes a transition to State 2, **Preparing SSF Instructions**.

To consider the processing of this state in more detail, it is expanded into a separate FSM depicted in figure 22.

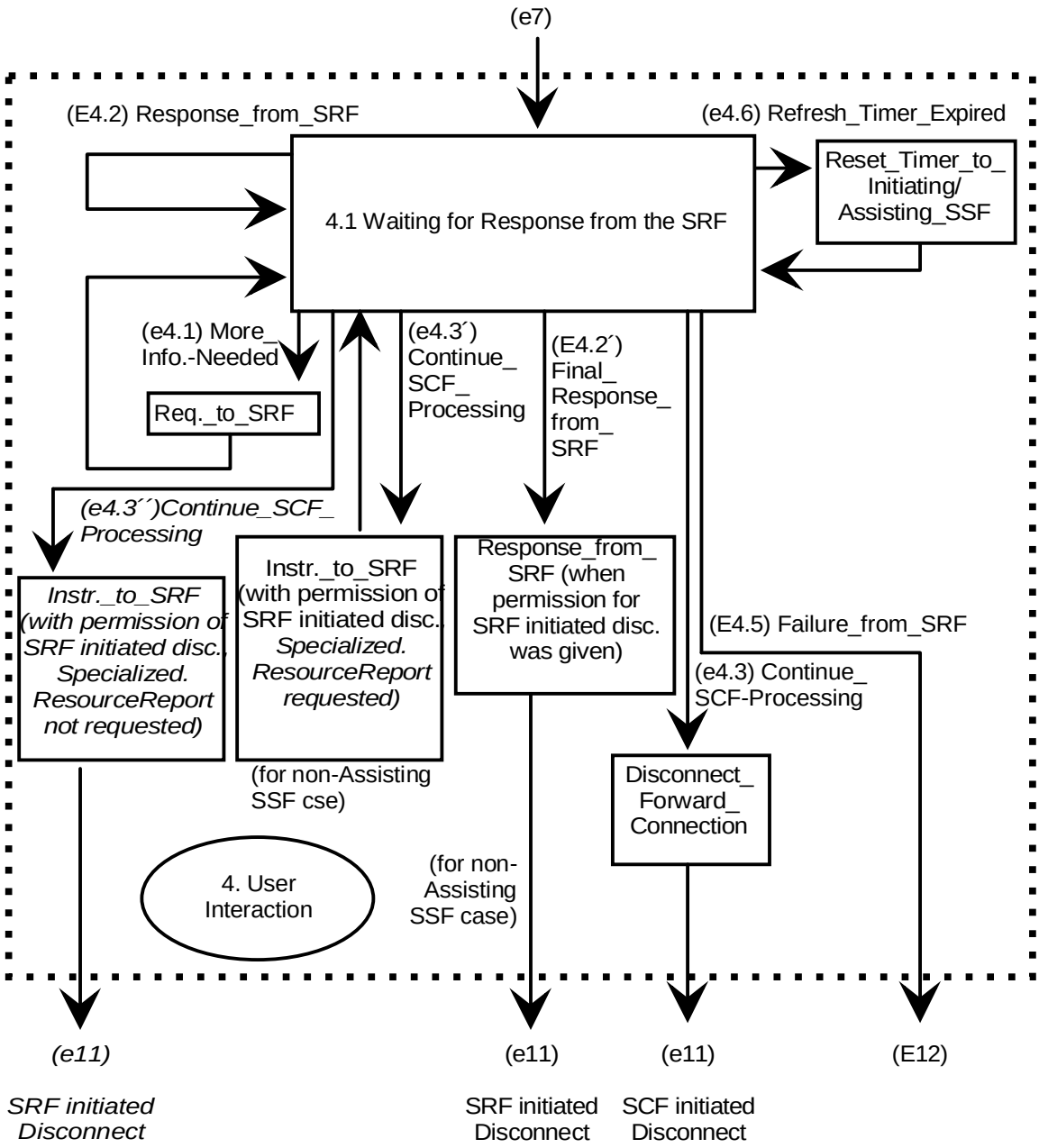


Figure 21: The State 4 FSM

7.2.5.4.1 State 4.1 "Waiting for Response from the SRF"

In this state, the SCF waits for the response to the previously sent operation and evaluates this response. The following events are considered in this state:

- (e4.1) More_Information_Needed results in issuing yet another operation to the SRF; it causes a transition back to State 4.1;

- (E4.2) Response_from_SRF: This is an external event caused by the reception of **Specialized Resource Report** or Return Result from **Prompt and Collect User Information** operation. On the receipt of this operation, the SCSM transits back to the same state;

- (E4.2') Final_Response_from_SRF: This is an external event caused by the reception of **Specialized Resource Report** operation, as requested by the SCF, in response to the previous **Play Announcement** or Return Result from **Prompt & Collect User Information** operation with permission of SRF-initiated disconnect. In the Initiating SSF relay case and the Direct SCF-SRF case, on the receipt of this event, the SCSM transits to the State 2, **Preparing SSF Instructions**. This event maps into the event (e11);

- (e4.3) Continue_SCF_Processing: This is an internal event that takes place when the SCSM finishes the User Interaction and requests the disconnection of bearer connection between the Initiating SSF and SRF by means of SCF initiated disconnect. In this case, the SCSM sends the **Disconnect Forward Connection** operation to the Initiating SSF and transits to the State 2, **Preparing SSF Instructions**. This event maps into the event (e11);
  This internal event also takes place when the SCSM finishes the User Interaction in case an error is occured for the operations PlayAnnouncement or Prompt&CollectUserInformation. This could be either a received error indication or the timeout for the appropriate operation timers.
  In this case the SLPI has indicated not to release the call due to the occured error and the state is changed back to 'Preparing SSF Instructions' to continue with an alternative processing, e.g. route to an alternative IP or continue without IP-functions.
  A DisconnectForwardConnection operation is to be sent, to release the established relay from SSF to SRF.

- (e4.3') Continue_SCF_Processing: This is an internal event that takes place when the SCSM finishes the User Interaction and requests the disconnection of bearer connection between the Initiating SSF and SRF by means of SRF-initiated disconnect, while an SpecializedResourceReport operation is requested to be returned to the SCF in case an announcement is completed.. In this case, the SCSM sends the **Play Announcement** (containing a request for returning a **SpecializedResourceReport** operation as an indication of completion of the operation) or **Prompt and Collect** operation with permission of SRF-initiated disconnect to the SRF;

- (e4.3") Continue_SCF_Processing: This is an internal event that takes place when the SCSM finishes the User Interaction and requests the disconnection of bearer connection between the Initiating SSF and SRF by means of SRF initiated disconnect, while no SpecializedResourceReport operation is requested to be returned to the SCF in case the announcement is completed. In this case, the SCSM sends the **PlayAnnouncement** operation (**not** containing a request for returning a **SpecializedResourceReport** operation as an indication of completion of the operation) with permission of SRF-initiated disconnect to the SRF. In the case of Assisting SSF, the SRF-initiated disconnect cannot be used. In this case, the SCSM transits to the state 2, **Preparing SSF Instructions**. This event maps into the event (e11);

- *The event Failure_from_SRF (E4.5) is mapped into the event (E12) Failure_from_SRF and the SCSM transits to State 2,Preparing SSF Instructions.*
  This event occurs when an error is received for the sent ConnectToResource operation. In this case the SLPI has indicated not to release the call due to this error and the state is changed back to 'Preparing SSFInstructions' to continue with an alternative processing, e.g. route to an alternative IP or continue without IP-functions.
  No DisconnectForwardConnection operation is prepared, because the relay from SSF to SRF has not been established.

- (e4.6) Refresh_Timer_Expired: This is an internal event that takes place on the expiration of $T_{SCF-SSF}$. In this case, the SCSM transmits the **Reset Timer** operation to the Initiating/Assisting SSF, and transits back to the same state

It should be noted that the bearer connection between the SSF and the SRF is disconnected when the SCSM exits from this state.

7.3 SRF Application Entity Procedures

7.3.1 General

This section provides the definition of the SRF Application Entity (AE) procedures related to the SRF-SCF interface. The procedures are based on the use of Signalling System No.7 (SS #7).

Other capabilities may be supported in an implementation-dependent manner in the IP, SSP or SN.

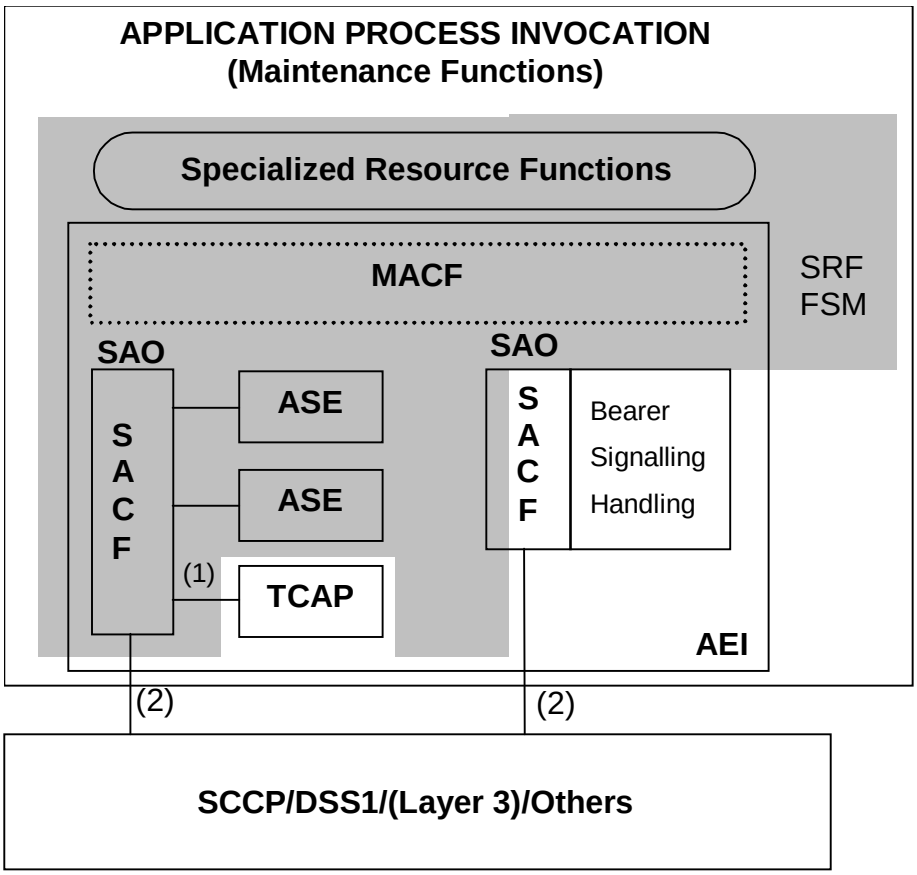
The AE, following the architecture defined in CCITT Recommendations Q.700, Q.771 and Q.1400, includes TCAP (Transaction Capabilities Application Part) and one or more ASEs called TC-users. The following sections define the TC-user ASE and SACF & MACF rules, which interface with TCAP using the primitives specified in CCITT Recommendation Q.771.

The procedure may equally be used with other message based signalling systems supporting the Application Layer structures defined.

In case interpretations for the application entity procedures defined in the following differ from detailed procedures and the rules for using of TCAP service, the statements and rules contained in the detailed clause 9 and 10 shall be followed.

7.3.2 Model and Interfaces

The functional model of the AE-SRF is shown in figure 22; the ASEs interface to TCAP (to communicate with the SCF) as well as interface to the maintenance functions. The scope of this Recommendation is limited to the shaded area in figure 22.



(1) TC-Primitives

(2) N-Primitives

NOTE: The SRF FSM includes several Finite State Machines.

AEI: Application Entity Invocation
SRF: Specialized Resource Functions
FSM: Finite State Model
MACF: Multiple Association Control Function
SACF: Single Association Control Function
SAO: Single Association Object

Note that SRF FSM includes several Finite State Machines.

Figure 22: Functional Model of SRF AE

The interfaces shown in figure 22 use the TC-user ASE primitives specified in CCITT Recommendation Q.771 (interface (1)) and N-primitives specified in CCITT Recommendation Q.711 (interface (2)). The operations and parameters of Intelligent Network Application Protocol (INAP) are defined in section 6 of this Recommendation.

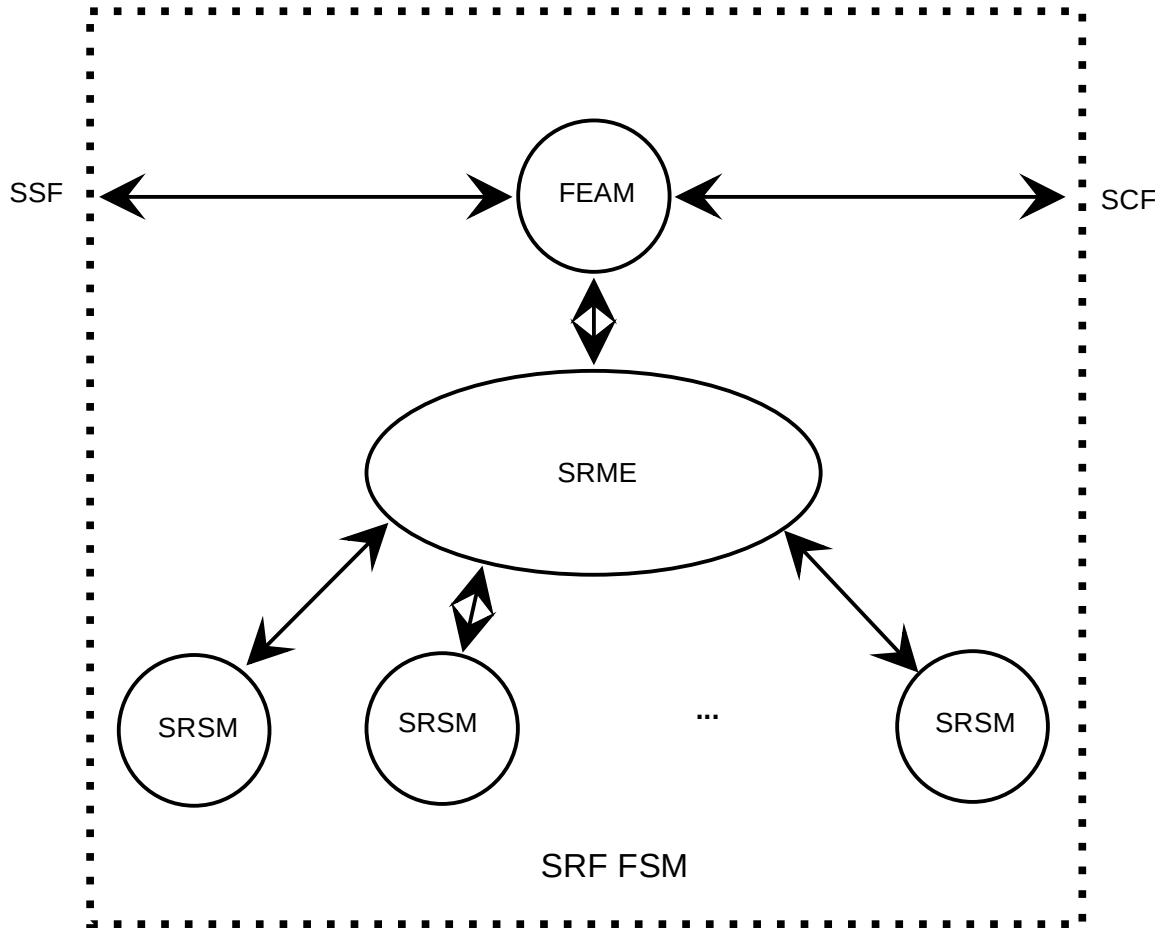
7.3.3 Relationship between the SRF FSM and Maintenance Functions/Bearer Connection Handling

The primitive interface between the SRF FSM and the maintenance functions is an internal interface and is not subject for Standardization in CS1.

The relationship between the Bearer Connection Handling and the SRF FSM may be described as follows for the case of a call initiated by the SSF: When a call attempt is initiated by the SSF, an instance of an SRF FSM is created.

The SRF FSM handles the interaction with the SCF FSM and the SSF FSM.

The management functions related to the execution of operation received from the SCF are executed by the SRF Management Entity (SRME). The SRME interface the different SRF Call State Models (SRSM) and the Functional Entity Access Manager (FEAM). figure 23 shows the SRF FSM structure.



SRME: SRF Management Entity
SRSM: SRF Call state Model
FEAM: Functional Entity Access Manager

Figure 23: SRF FSM Structure

The model associates a Finite State Model (FSM) with each initial interaction request from the SCF[5]. Thus, multiple initial requests may be executed concurrently and asynchronously by the SRF, which explains the need for a single entity that performs the tasks of creation, invocation, and maintenance of the SRSM objects.

Note [5]: Such a request is executed by the SCSM when it is in its State 4.

This entity is called the SRF Managing Entity (SRME). In addition to the above tasks, the SRME maintains the dialogues with the SCF and SSF on behalf of all instances of the SCSM. In particular, the SRME

1. Interprets the input messages from other FEs and translates them into corresponding SRSM events;

2. Translates the SRSM outputs into corresponding messages to other FEs; and

3. Handles the activity test functionality for the SCF-SRF relationship.

Finally, the Functional Entity Access Manager (FEAM) relieves the SRME of low-level interface functions. The FEAM functions include:

1. Establishing and maintaining the interfaces to the SSF and SCF;

2. Passing (and queueing when necessary) the messages received from the SSF and SCF to the SRME; and

3. Formatting, queueing (when necessary), and sending messages received from the SRME to the SSF and SCF.

7.3.4 The SRSM

The SRSM is presented in figure 24. In what follows, each state is described in a separate subsection 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 sub-sections. General rules applicable to more than one state are addressed here.

One component received in one TCAP message includes a single operation , 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.

- If there is an error in processing one of the operations in the sequence, the SRF FSM processes the error (see below).

- 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 section 10 (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 section 8; depending on the class of the operation, the error could be reported by the SRF to the SCF using the appropriate component (see Recommendation Q.774).

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 *cancels the user interaction* 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_{SRF}, whose purpose is to prevent excessive call suspension time. This timer is set when the SRF sends Setup Response bearer message to the SSF (SSF relay case) . This timer is stopped when a request is received from the SCF. The SRF may reset T_{SRF} on transmission of the **Specialized Resource Report** operation or the Return Result for the **Prompt and Collect User Information** operation. On the expiration of T_{SRF}, the SRSM transits to the **Idle** state ensuring that all SRF resources allocated to the call have been de-allocated.

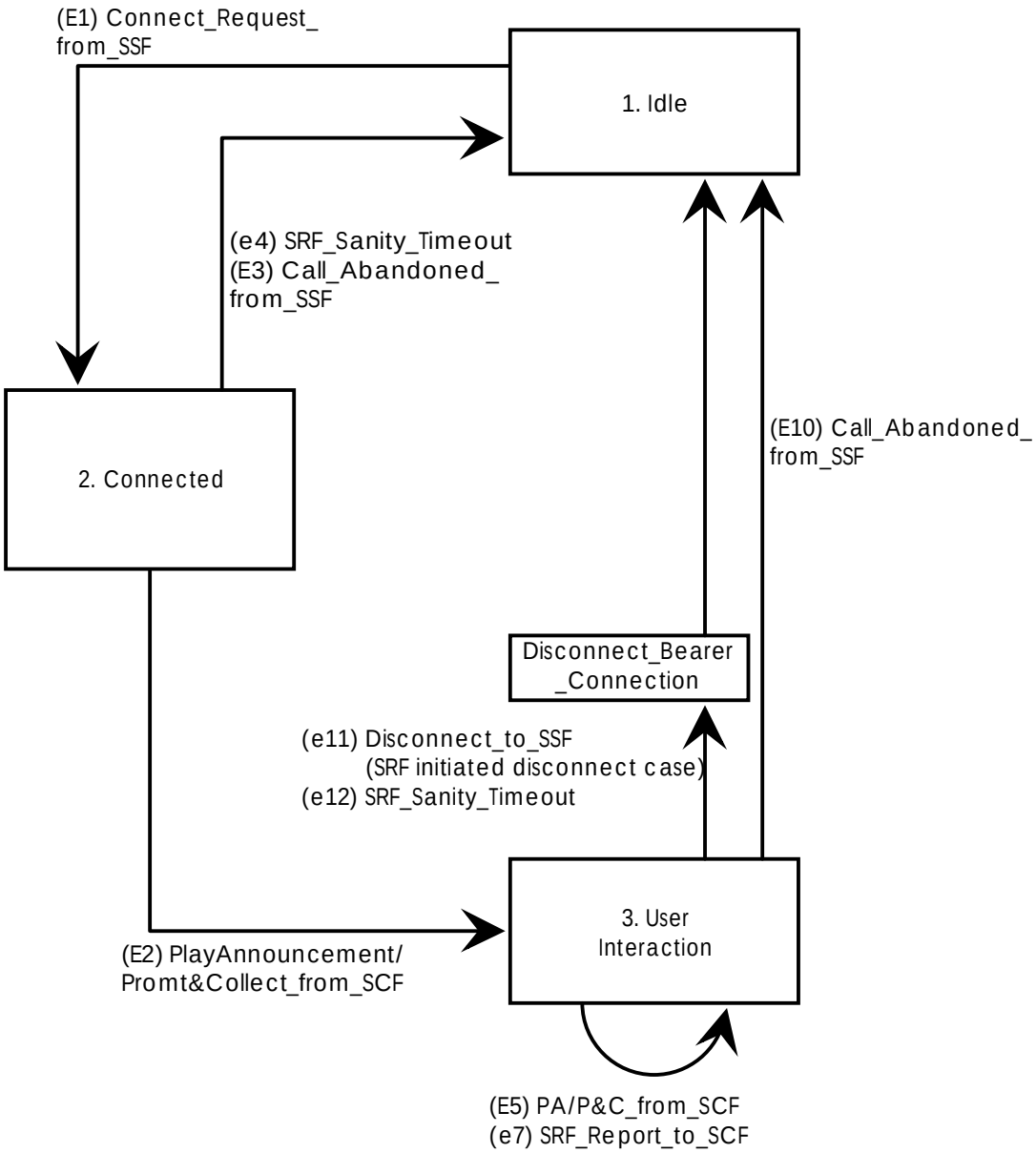


Figure 24: The SRSM

7.3.4.1 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. It is exited as a result of event E1.

- (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**";

- (E3) Call_Abandoned_from_SSF: this event takes place when the SRSM receives a Release message from the SSF in **Connected** state, indicating that the call party has disconnected. 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 a **PA/P&C** 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) Call_Abandoned_from_SSF: this event takes place when the SRSM receives a Release message from the SSF in **User Interaction** state, indicating that the call party has disconnected. The SRSM goes to state "**Idle**";

- (e11) Disconnect_to_SSF: this event occurs when the SCF has enabled SRF initiated disconnect by the last PA/P&C from SCF (E2) or (E5) with the parameter. The SRSM initiates a bearer channel disconnect sequence to the SSF using the applicable bearer channel signalling system after sending last **Specialized Resource Report** operation to the SCF (e7). The SRSM goes to state "**Idle**"; and

- (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 **PA/P&C** 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**".

7.3.4.2 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 **PA/P&C** has not yet been received . The method used to provide this bearer channel is not of interest in the FSM.

- (E1) Connect_request_from_SSF: this event corresponds to a bearer signalling connection request message from the SSF in the **Idle** state. The details of the bearer signalling state machine related to establishing the connection are not of interest in the SRF FSM. The SRSM goes to state "**Connected**".

- (E2) PA/P&C_from_SCF: this event takes place when the first **Play Announcement** or **Prompt & Collect User Information** operation from the SCF is received. The SRSM goes to state "**User Interaction**".

- (E3) Call_Abandoned_from_SSF: this event takes place when the SRF receives a release message from the SSF, indicating that the call party has disconnected. 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_{SRF}) without having a **PA/P&C** 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**".

7.3.4.3 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. It is exited as a result of events E10, e11 and e12. Events E5, e7 do not cause a state change.

- (E2) and (E5) PA/P&C_from_SCF: this event takes place when an initial or subsequent **Play Announcement** or **Prompt & Collect User Information** operation from the SCF is received. The SRSM goes to state "**User Interaction**" on the first (E2). The SRSM remains in state "**User Interaction**" for subsequent (E5)s.

- (e7) SRF_Report_to_SCF: this event takes place when an **Specialized Resource Report** operation is sent to the SCF. The SRSM remains in state "**User Interaction**".

- (E10) Call_disconnect_from_SSF: this event takes place when the SRSM receives a release message from the SSF, indicating that the call party has disconnected. The SRSM goes to state "**Idle**".

- (e11) Disconnect_to_SSF: this event occurs when the SCF has enabled SRF initiated disconnect with the last **PA/P&C** from SCF (E2) or (E5). The SRSM initiates a bearer channel disconnect sequence to the SSF using the applicable bearer channel signalling system after sending last **Specialized Resource Report** operation to the SCF. The SRSM goes to state "**Idle**".

- (e12) SRF_sanity_timeout: this event occurs when the SRSM has been connected for a network operator defined period of time (timer T_{SRF}) without having a **PA/P&C** 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**".

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 24 for the purpose of visual clarity.

7.3.5 Examples SRF Control Procedures

This section 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 various control procedures are described in this section in accordance with the example physical scenarios of protocol architecture in Q.1218 Section 4.2.

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

7.3.5.1 SRF Connect Procedures

7.3.5.1.1 SRF Connect Physical Procedures

Several procedures are required for different physical scenarios. The cases to be covered are described below and illustrated in figure 25.

i. the IP is integrated into the SSP, or directly attached to the SSP, 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;

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;

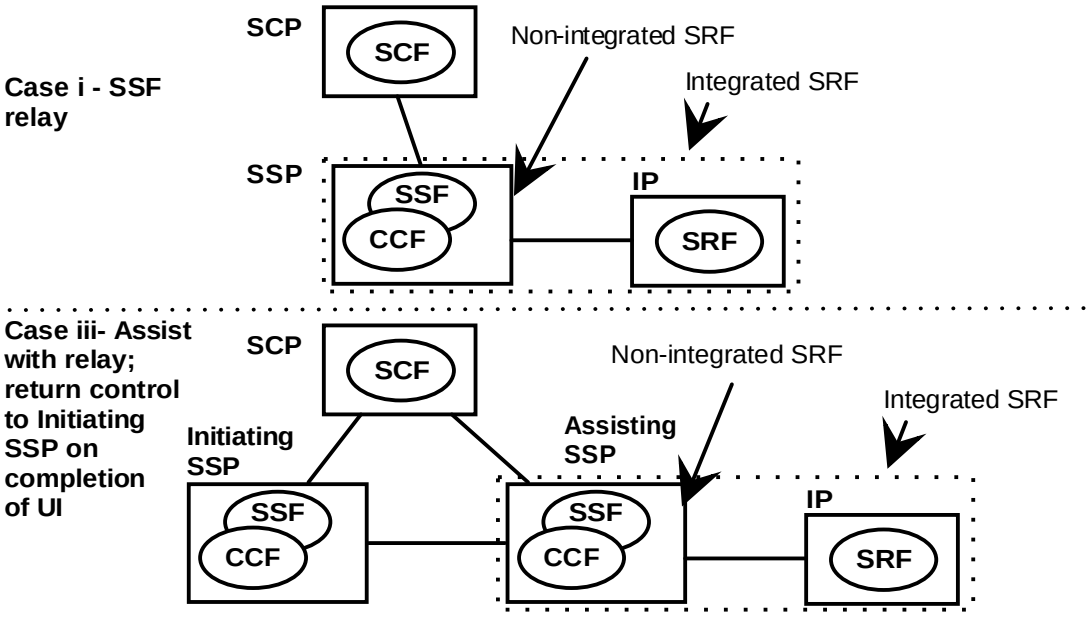


Figure 25: Physical Scenarios

In the above case, the operations between the SCP and the SSP *are* SS No.7 TCAP-based; the messaging between the SSP and the IP when the SSP does relaying *is an internal interface (Non-integrated SRF and Integrated SRF)* ; and bearer control signalling *is according to [7]*.

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

Case i is illustrated in figure 26. 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 **Connect to Resource** operation discussed in the section on user interaction below. The establishment of the SCF-SRF relationship in this case is implicit.

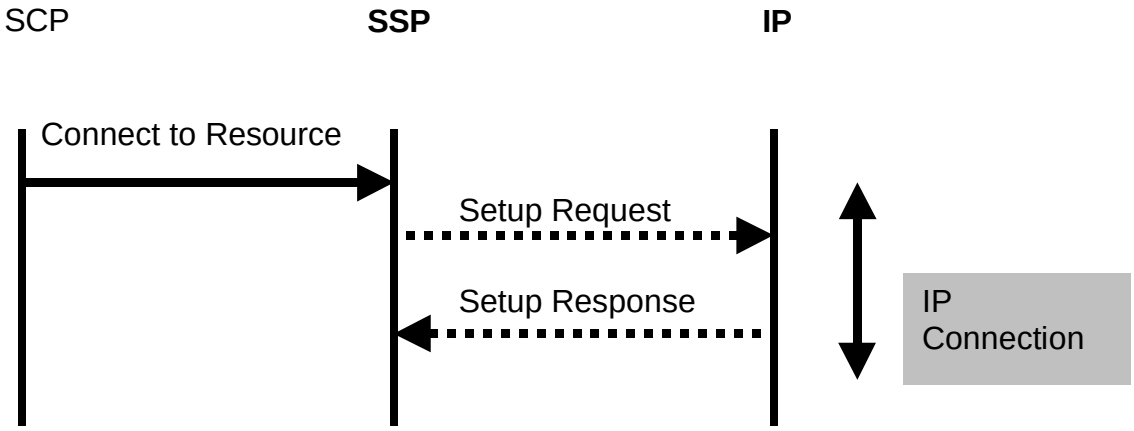


Figure 26: Connection to Integrated or External IP with SSP Relay of IP operations

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.
17.03.989

After the **Assist Request Instructions** is received by the SCP, the procedures are the same as case i. figure 28 illustrates the preamble involved.

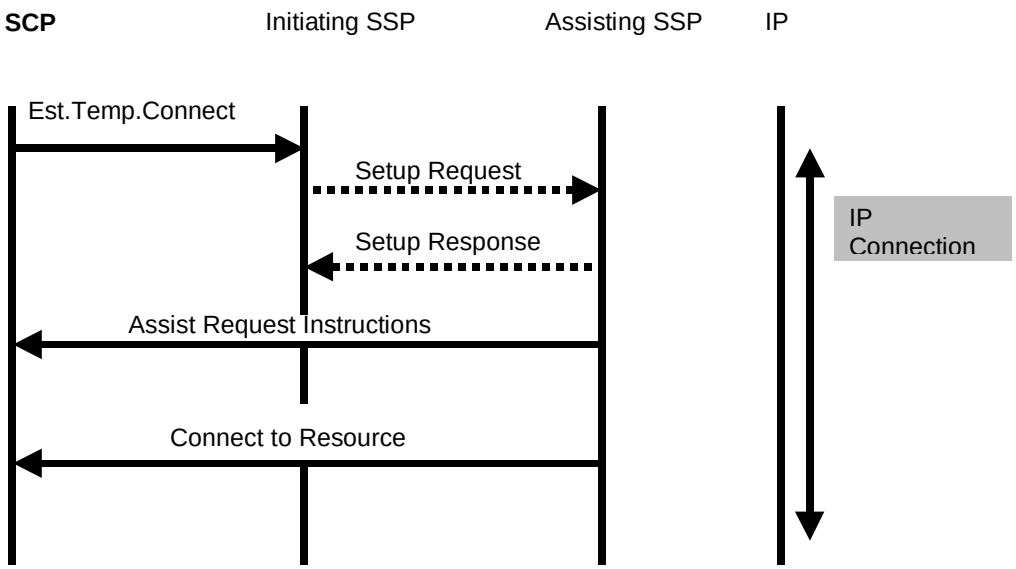


Figure 28: Preamble for Assist Case with Integrated IP or External IP and SSP Relay of SCP - IP Messages

7.3.5.2 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 **Play Announcement** operations;
- a dialogue with the end user by using one or a sequence of **Prompt & Collect User Information** operations;
- a combination of the above;

7.3.5.2.1 Play Announcement / Prompt and Collect (PA/P&C)There ***is only one*** physical scenario for user interaction:

i. the SSP relays the operations from the SCP to the IP and the responses from the IP to the SCP (SSF relay case); and

Case i is illustrated in figure 31 below.

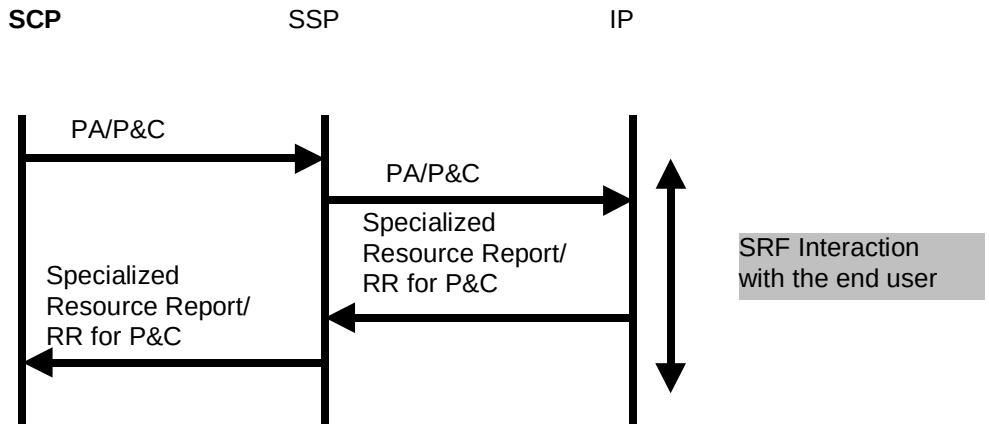


Figure 31: SSP Relay of User Interaction Operations and Responses

7.3.5.3 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 unless the transaction with the SCF has been terminated, indicating the user interaction completed the call. 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 need simply choose not to regenerate the disconnect in the backward direction. figure 33 illustrates this concept.

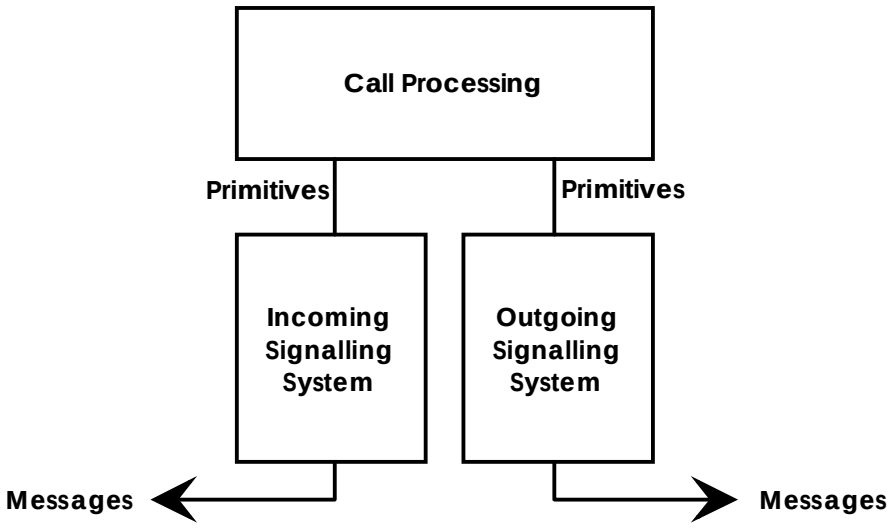


Figure 33: Relationship of Incoming and Outgoing Signalling Systems to Call Processing

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.

1. If a series of **PA/P&C** operations are to be executed by the same SRF, then SRF disconnect is inhibited for all but the last and may be inhibited on the last **PA/P&C**.

3. The SCF must either explicitly order "Disconnect" or enable SRF initiated disconnect at the end of the **PA/P&C**. An SRF left connected without a **PA/P&C** to execute may autonomously disconnect if it has not received any **PA/P&C** operations within a defined time limit. (This could occur, for example, after an **Establish Temporary Connection** which is not followed within a reasonable time period with a **PA/P&C** 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 **PA/P&C**, then the SRF must disconnect on completion of the user interaction.

5. When SRF initiated disconnect is not enabled, the SCF must ask the SRF to inform it of the completion of the User Interaction using the **Specialized Resource Report** operation for "announcement complete" or using the Return Result for the **Prompt and Collect User Information** operation.

6. If the user disconnects, the SRF is disconnected and the SSF releases resources and handles the transaction between the SSF and the SCF as specified in Q.1214 and Q.1218. The SRF returns its resources to idle. The relationship with the SCF is terminated.

7. When the SCF explicitly orders the SSF to disconnect by "**Disconnect Forward Connection**" operation, the SSF releases the bearer connection to the SRF, and returns to the "**Waiting for Instructions**" state. No operation reporting SRF disconnect from the SSF to the SCF is required.

7.3.5.3.1 SRF Initiated Disconnect

The SRF disconnect procedure is illustrated in figure 34. The SRF disconnect is enabled by the SCF within a **PA/P&C** operation. When the SRF receives a **PA/P&C** enabling disconnection, it completes the dialog as instructed by the **PA/P&C**, 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 returns to the "**Waiting for Instructions**" state and executes any buffered operations.

For the Assisting SSF case, the SRF initiated disconnect procedures are not used because the Assisting SSF remains in the "**Waiting for Instructions**" state and does not propagate the disconnection of the bearer connection to the Initiating SSF. The SCF initiated disconnect procedures described in the following sub-section are used for the Assisting SSF case.

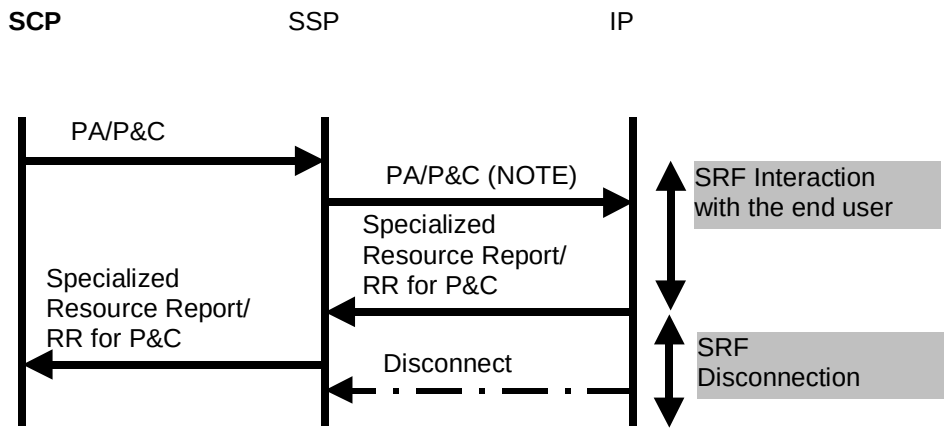


Figure 34: *SRF* Disconnect

7.3.5.3.2 SCF Initiated Disconnect

The SCF initiated disconnect procedure is illustrated in figure 35. (Bearer messages are shown in gray). The figure shows only the Assisting SSR case. To initiate the SCF initiated disconnection of the SRF, the SCF must request and receive a reply to the last **PA/P&C** operation requested. The **Specialized Resource Report** operation contains an "announcement complete" and Return Result for **P&C** contains "collected information."

The SCF initiated disconnect uses an operation called **Disconnect Forward Connection**. Once the **Disconnect Forward Connection** operation is received by the SSF, it will initiate a "release of bearer channel connection" between the physical entities containing the SSF and SRF, using applicable bearer control signalling. Since the SCF (which initiates the disconnect), the SSF (which instructs bearer signalling to disconnect) and the SRF (which receives disconnect notification via bearer signalling) are aware that disconnect is occurring, they are synchronized. Therefore, a "pre-arranged" end may be used to close the transaction. This does not preclude the use of explicit end messages for this purpose.

For Assisting SSF case, the initiating SSP, on receipt of the Disconnect Forward Connection from the SCP, disconnects forward to the assisting SSP, and this disconnection is propagated to the IP. The initiating SSP, knowing that the forward connection was initiated as the result of an Establish Temporary Connection, does not disconnect back to the user but returns to the "Waiting for Instructions" state.

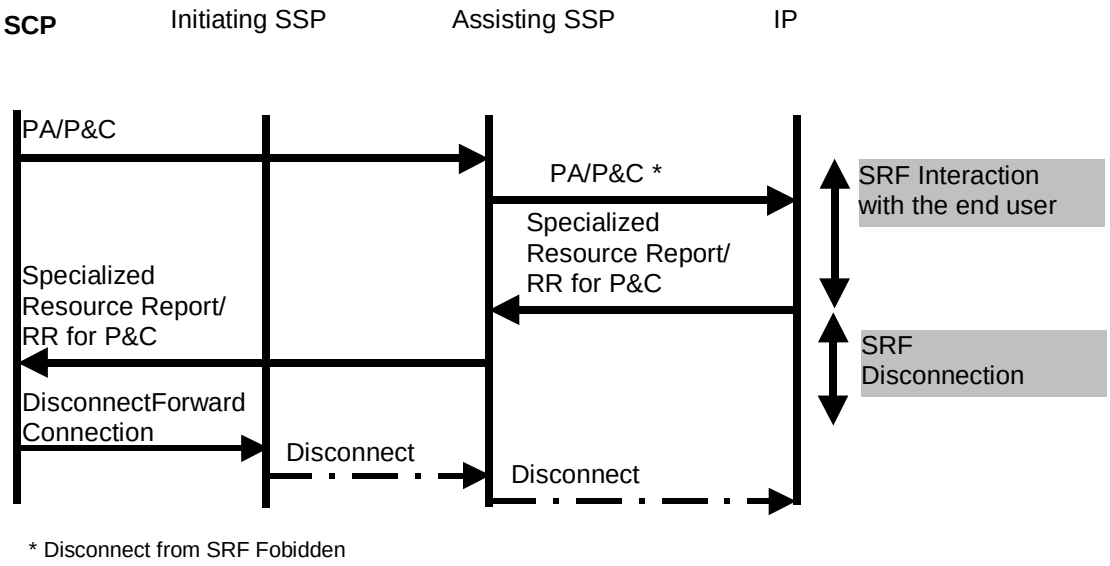


Figure 35: SCF Initiated Disconnect for Assist Scenario

7.3.5.4 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

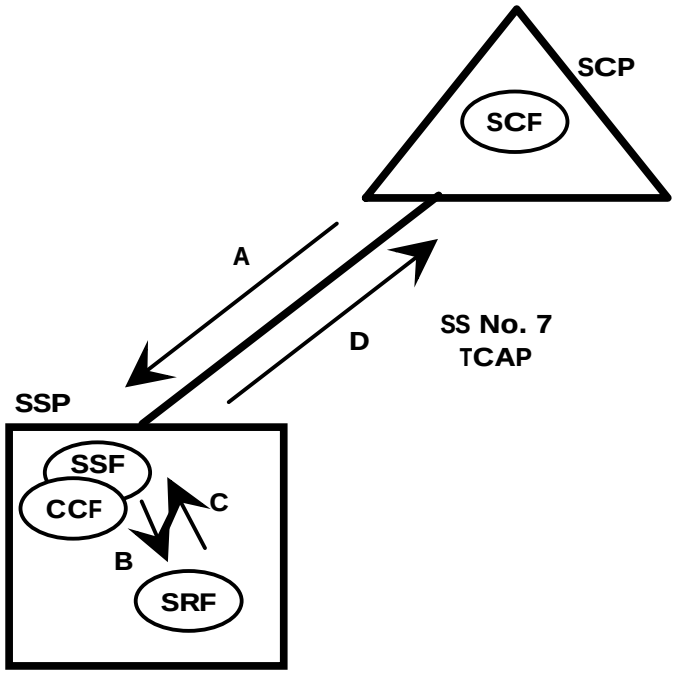


Figure 36: SSP with integrated SRF

In figure 36 above, the SSP with an integrated (or embedded) SRF, the procedural scenarios can be mapped as given in table 4:

Table 4

Procedure Name	**Operations**	Protocol Flows
Connect to Resource and first PA/P&C	Connect to Resource; PA/P&C	A
	Setup; PA/P&C	B
User Interaction	PA/P&C	A then B
	Specialized Resource Report/RR for P&C	C then D
SRF Initiated Disconnect	Specialized Resource Report/RR for P&C	C then D
	Disconnect	C (intra-SSP bearer control)
SCF Initiated Disconnect	Specialized Resource Report/RR for P&C	C then D
	Disconnect Forward Connection	A
	Disconnect	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 37.

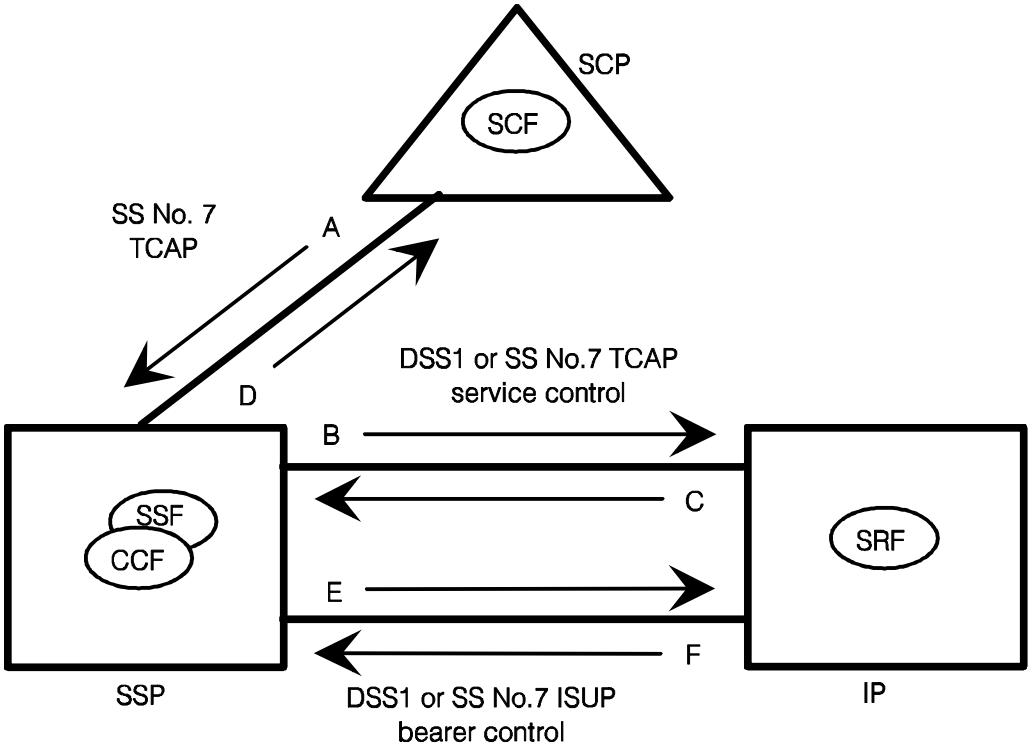


Figure 37: SSP relays messages between SCP and IP

NOTE: The information flows B, C, E and F are only examples and not specified by this document. The interface between SSP and IP is an internal interface.

The procedural scenarios for this Relay SSF with an IP (figure 37) can be mapped as shown in table 5.

Table 5

Procedure Name	Operations	Protocol Flows
Connect to Resource and first PA/P&C	ConnectToResource; PA/P&C	A
	If DSS1 used:	
	Setup; PA/P&C	E and B (Facility IE)
	If Signalling System No.7 used:	
	Setup	E
	PA/P&C	B
User Interaction	PA/P&C	A then B
	SpecializedResourceReport/ RR for P&C	C then D
SRF Initiated Disconnect	SpecializedResourceReport/ RR for P&C	C then D
	Disconnect	F
SCF Initiated Disconnect	SpecializedResourceReport/ RR for P&C	C then D
	DisconnectForwardConnection	A
	Disconnect	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 39. 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.

The integrated SRF and SSF relay case requires a transaction between the SCP and the assisting SSP (figure 39).

The SCP shall again correlate two transactions.

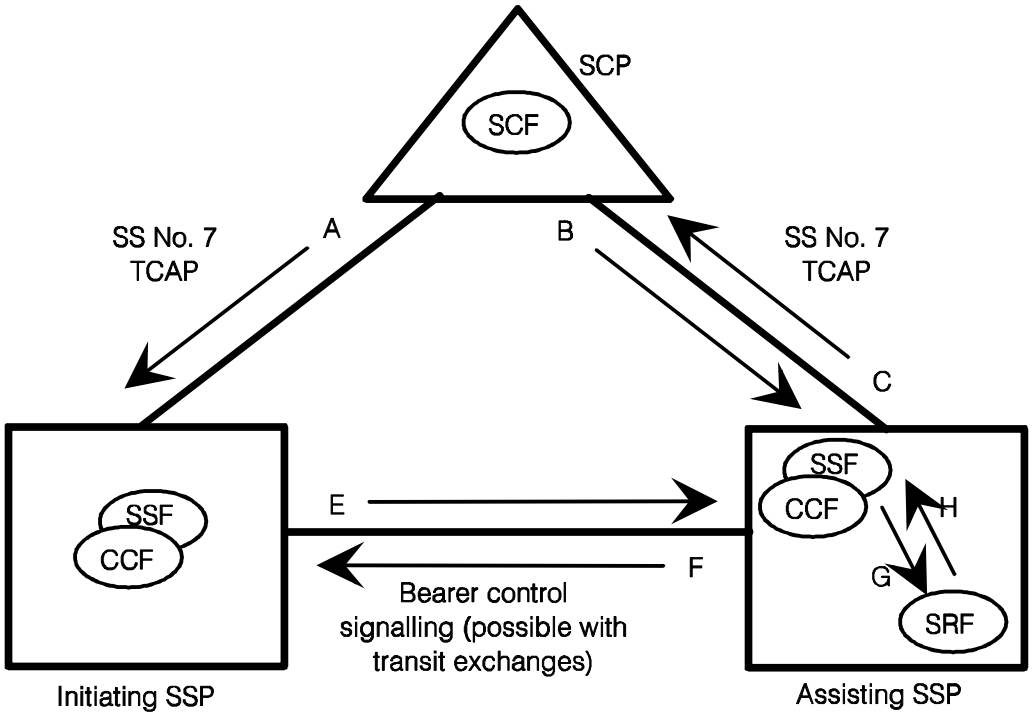


Figure 39: SSP Assist (Relay SSP)

In figure 39, the procedural scenarios can be mapped as shown in table 7.

Table 7

Procedure Name	Operations	Protocol Flows
Assist preamble	EstablishTemporaryConnection	A
	Setup	E
	AssistRequestInstructions	C
	ConnectToResource	B
	Setup	G
	ResetTimer	A
User Interaction	PA/P&C	B then G
	SpecializedResourceReport/ RR for P&C	H then C
SCF Initiated Disconnect	SpecializedResourceReport/ RR for P&C	H then C
	DisconnectForwardConnection	A
	Disconnect	E and G (intra-SSP bearer ctrl)

The Assisting SSP case shown in figure 39 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 37.

7.3.5.4.1 Message sequences for service assist

The following subclause provides additional details on the message sequences for the service assist procedure in figure 39:

1) 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 and the CorrelationID (both used for the assisting SSP to establish communication back to the SCP) is sent to the initiating SSP. EstablishTemporaryConnection is used instead of a regular Connect operation because of the nature of the connection to the assisting SSP. The initiating SSP shall be aware that the SCP will ask it to continue in the processing of the call at some point in the future.

   Protocol Flow A

2) The initiating SSP routes the call to the assisting SSP. The ScfID and CorrelationID are sent to the assisting SSP. The transport mechanism used to send this information between SSPs is independent of the service assist control procedures between the SCF and SSF.

   Protocol Flow E

3) 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.

   Protocol Flow C

4) The SCP sends instructions to the assisting SSP based on SL control.

   Protocol Flow B

5) The SCP may need to generate ResetTimer events to the initiating SSP so that it does not time out the call.

   Protocol Flow A

6) When resource functions have been completed, a DisconnectForwardConnection operation is sent to the initiating SSP. This indicates, that the temporary connection to the assisting SSP has to be disconnected.

   Protocol Flow A

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

   Protocol Flow E

8) The call control returns to the initiating SSP.

8 Error procedures

8.1 Classification of error cases

In this section the different interface-related error cases for errors detected by the TC-user are defined. The mapping onto TC-U-REJECT and TC-U-ERROR is indicated in section 8.4 . The application specific TC-user exception handling is described in the following way:

- *Section 8.2 listing the errors related to INAP operations*
- *Section 8.3 listing the errors related to error conditions in the different FE's which are not directly related to the INAP operations.*
- *Section 8.4 describes the call handling in error cases*
- *In section 9 operation specific SCF and SSF actions are described, if there are any*
- *In section 10 the TCAP dialogue handling in error cases is described*

8.1.1 Errors related to a received operation

The following errors may occure:

- *Operation not understood:*
  Means that the operation code is unknown at the receiving application.

- *Operation out of context:*
  Means that the operation is known at the receiving application but is not allowed in this particular transaction or in the present state of the receiving application entity.

- *Error in an operation:*
  Means that the operation has been understood and is in context but an error occured during its further processing. The following errors can occur:

  - *General errors:*
    - *Protocol error*
    - *Lack of resources*
    - *Dialogue is about to be released*

  - *Application specific user errors:*
    These errors are specified in the operation definition and sent in an TC-U-ERROR component.

8.1.2 Errors related to a sent operation

The following errors can occur related to a sent message:

- *Operation timer expiry for class 1 and class 3 operations*
- *Receipt of ERROR or REJECT for a component sent*
- *Receipt of a TC-NOTICE, indicating that the message could not be delivered to the peer entity*

8.1.3 General interface related errors

In addition to the above defined errors general interface related errors such as application timer expiry or unexpected closing of the transaction may occur.

8.1.4 Errors at application level

If a requested class 1 or class 2 operation cannot be executed succesfully by the receiving side the application specific errors are sent to the invoker. In section 8.2 the procedures for the application specific errors are described.

8.2 Operation related error procedures

The following sections define the generic error handling for the operation related errors. The errors are defined as operation errors in section 6. The TCAP services which are used for reporting operation errors are described in section 10.

Errors which have a specific procedure for an operation are described in section 9 with the detailed procedure of the related operation.

8.2.4 ETCFailed

8.2.4.1 General description

8.2.4.1.1 Error description

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

8.2.4.2 Operations SCF->SSF

EstablishTemporaryConnection

Procedures at invoking entity (SCF)

A) SCF sends ETC to SSF
precondition: SCSM state 3.1 Determine Mode
postcondition: SCSM state 3.2 Waiting for AssistRequestInstructions

B) SCF receives ETCFailed error from SSF
precondition: SCSM state 3.2 Waiting for AssistRequestInstructions
postcondition: SCSM state 2.1 Preparing SSF Instructions
Error handling depends on the SL, e.g. selecting another SRF or continue the processing of the call.

Procedures at responding entity (SSF**)**
SSF receives ETC from SCF but the establishment of the connection fails, resulting in the returning of the ETCFailed error to the SCF
precondition: SSF FSM state c Waiting for Instructions
postcondition: SSF FSM state c Waiting for Instructions
No further error treatment.

8.2.5 ImproperCallerResponse

8.2.5.1 General description

8.2.5.1.1 Error description

The format of the user input has been checked by the SRF and doesn't correspond with the required format as it was defined in the initiating P&C Operation.

8.2.5.2 Operations SCF-->SRF

PromptAndCollectUserInformation

Procedures at invoking entity (SCF)
A) SCF sends P&C to SRF
precondition: SCSM state 3.1 Determine Mode; P&C will accompany the ConnectToResource

	or SCSM state 3.2	Waiting for AssistRequestInstructions; after EstablishTemporaryConnection
	or SCSM state 4.1	Waiting for Response from the SRF; if more PAs or P&Cs are active.
postcondition:	SCSM state 4.1	Waiting for Response from the SRF

B) SCF receives ImproperCallerResponse Error from SRF

precondition:	SCSM state 4.1	Waiting for Response from the SRF
postcondition:	SCSM state 4.1	Waiting for Response from the SRF

Error treatment depends on SL. SCF can initiate new User Interaction or forcing a Disconnect (to SSF).

Procedures at responding entity (SRF)

A) SRF receives P&C

Precondition:	SRSM state 2	Connected
	or SRSM state 3	User Interaction
Postcondition:	SRSM state 3	User Interaction

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

Precondition:	SRSM state 3	User Interaction
Postcondition:	SRSM state 3	User Interaction

SRF waits for a new Operation from SCF. This may be a new P&C or PA.

8.2.6 MissingCustomerRecord

8.2.6.1 General description

8.2.6.1.1 Error description

The Service Logic Program could not be found in the SCF, because the required customer record does not exist, or the requested Service Logic Program instance, indicated by correlationID in 'AssistRequestInstructions' does not exist anymore.

8.2.6.2 Operations SSF-->SCF

AssistRequestInstructions
InitialDP

Procedures at invoking entity (SSF)

A) Sending Operation

precondition:	SSF FSM state b	Trigger processing
	or SSF FSM state b'	Waiting for Instructions; in case of assist/handoff
postcondition:	SSF FSM state c	Waiting for Instructions
	or SSF FSM state b'	Waiting for Instructions; in case of assist/handoff

B) SSF receives Error 'MissingCustomerRecord'

precondition:	SSF FSM state c	Waiting for Instructions
	or SSF FSM state b'	Waiting for Instructions; in case of assist/handoff
postcondition:	SSF FSM state a	Idle
	or SSF FSM state a'	Idle; in case of assist/handoff

The CCF routes the call if necessary (e.g. default routing to a terminating announcement).

Procedures at responding entity (SCF)

The SCSM detects that the required Service Logic Program does not exist *(i.e. the customer record in the SCF does not exist, e.g. in case of TDPs a SLP is attempted to be invoked)*. The SLPI may not exist anymore (e.g. in case of the operation AssistRequestInstructions). The Error parameter MissingCustomerRecord is used to inform the invoking entity of this situation. The maintenance functions are informed.

8.2.7 MissingParameter

8.2.7.1 General description

8.2.7.1.1 Error description

There is an Error in the received Operation argument. The responding entity cannot start to process the requested Operation because the argument is incorrect: an expected optional parameter which is essential for the application is not included in the Operation argument.

8.2.7.2 Operations SCF-->SSF

Non Call Associated
ActivateServiceFiltering

Call Associated/Non Call Processing

ApplyCharging	CallInformationRequest
FurnishChargingInformation	
RequestReportBCSMEvent	ResetTimer
SendChargingInformation	

Call Associated/Call Processing

Connect	ConnectToResource
EstablishTemporaryConnection	CollectInformation

Procedures at invoking entity (SCF)
A) Sending Operation

precondition:	SCSM	any state in which the above operations can be transferred.
postcondition:	SCSM	any state as result of the transfer of any of the above operations

B) SCF receives Error 'MissingParameter'

precondition:	SCSM	any state as result of the transfer of any of the above operations.
postcondition:	SCSM	transition to the initial state (i.e. before sending the erroneous operation).

The SL and maintenance functions are informed. Further treatment of the call is dependent on Service Logic.

Procedures at responding entity (SSF)

precondition:	(1) SSF-FSM	appropriate state.
	(2) SSF-FSM	Operation received, appropriate event occurred.
postcondition:	(1) SSF-FSM	transition to the same state.

The SSF-FSM detects the error in the received operation. The Error parameter is returned to inform the SCF of this situation.

8.2.7.3 Operations SSF-->SCF

AssistRequestInstructions
InitialDP
ApplyChargingReport

Procedures at invoking entity (SSF)
A) Sending Operation

precondition:	SSF FSM	any state in which the above operations can be transferred.
postcondition:	SSF FSM	any state as result of the transfer of any of the above operations

B) SSF receives Error 'MissingParameter'

precondition:	SSF FSM	any state as result of the transfer of any of the above operations.
postcondition:	SSF FSM state a	Idle

After receiving this Error, the SSF-FSM returns to the state Idle, the CCF routes the call if necessary (default routing to a terminating announcement). If the call is already established (i.e. ApplyChargingReport), the CCF disconnects the call. In case of an assisting SSF the temporary connection is released by the assisting SSF.

Procedures at responding entity (SCF)

precondition:	(1) SCSM	appropriate state.
	(2) SCSM	Operation received, appropriate event occurred.
postcondition:	(1) SCSM state 1	Idle; in case of InitialDP or ApplyChargingReport.
	or	(2) SCSM state 2.1 Preparing SSF Instructions; in case of AssistRequestInstructions.

The SCSM detects the erroneous situation. The Error parameter is used to inform the SSF of this situation. The SL and maintenance functions are informed.

8.2.7.4 Operations SCF-->SRF

PlayAnnouncement
PromptAndCollectUserInformation

Procedures at invoking entity (SCF)

A) Sending Operation

precondition:	SCSM state 3.1	Determine Mode; P&C or PA will accompany the ConnectToResource
	or SCSM state 3.2	Waiting for AssistRequestInstructions; after EstablishTemporaryConnection
	or SCSM state 4.1	Waiting for Response from the SRF; if more PAs or P&Cs are outstanding.
postcondition:	SCSM state 4.1	Waiting for Response from the SRF

B) Receiving Error

precondition:	SCSM state 4.1	Waiting for Response from the SRF
postcondition:	SCSM state 4.1	Waiting for Response from the SRF

Error treatment depends on SL. SCF can initiate new User Interaction or forcing a Disconnect (to SSF).

Procedures at responding entity (SRF)

precondition:	SRSM state 2	Connected
	or SRSM state 3	User Interaction
postcondition:	SRSM state 3	User Interaction

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.

8.2.9 ParameterOutOfRange

8.2.9.1 General description

8.2.9.1.1 Error description

The responding entity cannot start the processing of the requested Operation because an Error in a parameter of the Operation argument is detected: a parameter value is out of range.

All errors, which can be detected by the ASN.1 decoder, already may be detected during the decoding of the TCAP message and indicated by the TC error indication "MistypedParameter".

8.2.9.2 Operations SCF-->SSF

Non Call Associated
ActivateServiceFiltering

Call Associated/Non Call Processing
ApplyCharging
CallInformationRequest
SendChargingInformation

Refer to section 8.2.7 MissingParameter for the appropriate error procedures.

8.1.8.3 Operations SSF->SCF

ApplyChargingReport

Refer to section 8.2.7 MissingParameter for the appropriate error procedures.

8.2.14 SystemFailure

8.2.14.1 General description

8.2.14.1.1 Error description

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

8.2.14.1.2 Argument description

PARAMETER
UnavailableNetworkResource

UnavailableNetworkResource ::= ENUMERATED {
unavailableResources(0),
componentFailure(1),
basicCallProcessingException(2)
}

8.2.14.2 Operations SCF-->SSF

Non Call Associated
ActivateServiceFiltering

Call Associated/Non Call Processing
ApplyCharging CallInformationRequest
RequestReportBCSMEvent
SendChargingInformation

Call Associated/Call Processing
CollectInformation Connect
ConnectToResource DisconnectForwardConnection
`EstablishTemporaryConnection`

Refer to section 8.2.7 MissingParameter for the appropriate error procedures.

8.2.14.3 Operations SSF-->SCF

InitialDP
ApplyChargingReport

Refer to section 8.2.7 MissingParameter for the appropriate error procedures.

8.2.14.4 Operations SCF-->SRF

PlayAnnouncement
PromptAndCollectUserInformation

Refer to section 8.2.7 MissingParameter for the appropriate error procedures.

8.2.15 TaskRefused

8.2.15.1 General introduction

8.2.15.1.1 Error description

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

8.2.15.1.2 Argument description

```
PARAMETER    ENUMERATED {
        generic(0),
        unobtainable (1),
        congestion(2)
        }
```

8.2.15.2 Operations SCF-->SSF

Non Call Associated
ActivateServiceFiltering

Call Associated/Non Call Processing

ApplyCharging	CallInformationRequest
FurnishChargingInformation	
RequestReportBCSMEvent	`ResetTimer`
SendChargingInformation	

Call Associated/Call Processing

CollectInformation	Connect
ConnectToResource	DisconnectForwardConnection
`EstablishTemporaryConnection`	

Refer to section 8.2.7 MissingParameter for the appropriate error procedures.

8.2.15.3 Operations SSF-->SCF

AssistRequestInstructions
InitialDP
ApplyChargingReport

Refer to section 8.2.7 MissingParameter for the appropriate error procedures.

8.2.15.4 Operations SCF-->SRF

PromptAndCollectUserInformation

Refer to section 8.2.7 MissingParameter for the appropriate error procedures.

8.2.16 UnavailableResource

8.2.16.1 General description

8.2.16.1.1 Error description

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

8.2.16.2 Operations SCF-->SRF

PlayAnnouncement
PromptAndCollectUserInformation

Procedures at invoking entity (SCF)
A) SCF sends PA or P&C to SRF

precondition:	SCSM state 3.1	Determine Mode; PA or P&C will accompany the ConnectToResource
	or SCSM state 3.2	Waiting for AssistRequestInstructions; after EstablishTemporaryConnection
	or SCSM state 4.1	Waiting for Response from the SRF; if more PAs or P&Cs are outstanding.
Postcondition:	SCSM state 4.1	Waiting for Response from the SRF

B) SCF receives UnavailableResource Error from SRF

precondition:	SCSM state 4.1	Waiting for Response from the SRF
postcondition:	SCSM state 4.1	Waiting for Response from the SRF

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 PA or P&C (if possible)
- terminate service processing

Procedures at responding entity (SRF)
A) SRF receiving PA or P&C

precondition:	SRSM state 2	Connected; if initial PA or P&C
	or SRSM state 3	User Interaction; if not initial PA or P&C

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

Precondition:	SRSM state 3	User Interaction
Postcondition:	SRSM state 3	User Interaction

8.2.17 UnexpectedComponentsequence

8.2.17.1 General description

8.2.17.1.1 Error description

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

8.2.17.2 Operations SCF-->SSF

Non-call Associated
ActivateServiceFiltering

Call Associated/Non-call Processing

ApplyCharging	CallInformationRequest
FurnishChargingInformation	
RequestReportBCSMEvent	ResetTimer
SendChargingInformation	

Call Associated/Call Processing

CollectInformation	Connect
ConnectToResource	Continue
DisconnectForwardConnection	EstablishTemporaryConnection

In this case the SSF detects the erroneous situation, sends the UnexpectedComponentSequence error and remains in the same state. In the SCF the SL and maintenance functions are informed and the SL decides about error treatment.

8.2.17.3 Operations SSF-->SCF

ApplyChargingReport
AssistRequestInstructions
InitialDP

In case of assisting SSF an error occurs in case an AssistRequestInstructions is sent while a relationship between SCF and assisting SSF has already been established. In that case the SCF returns the error parameter. SL and maintenance are informed. On receiving the error the assisting SSF moves to Idle and the temporary connection is released.

In case the operation is sent by an 'initiating' SSF in the context of an existing relationship, the SCF returns the error parameter. SL and maintenance are informed. On receiving the error the SSF moves to Idle.

8.2.17.4 Operations SCF-->SRF (only applicable for direct SCF-SRF case)

PlayAnnouncement
PromptAndCollectUserInformation

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

8.2.18 UnexpectedDataValue

8.2.18.1 General description

8.2.18.1.1 Error description

The responding entity cannot complete the processing of the requested Operation because a parameter has an unexpected data value.

Note that this error does not overlap with 'ParameterOutOfRange'

Example:

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

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

8.2.18.2 Operations SCF-->SSF

Call Associated/Non Call Processing

ApplyCharging FurnishChargingInformation
RequestReportBCSMEvent
ResetTimer

Call Associated/Call Processing
CollectInformation Connect
ConnectToResource EstablishTemporaryConnection

Refer to section 8.2.7 MissingParameter for the appropriate error procedures.

8.2.18.3 Operations SSF-->SCF

AssistRequestInstructions
InitialDP
ApplyChargingReport

Refer to section 8.2.7 MissingParameter for the appropriate error procedures.

8.2.18.4 Operations SCF-->SRF

PlayAnnouncement
PromptAndCollectUserInformation

Refer to section 8.2.7 MissingParameter for the appropriate error procedures.

8.2.19 UnexpectedParameter

8.2.19.1 General description

8.2.19.1.1 Error description

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

8.2.19.2 Operations SCF-->SSF

Non Call Associated
ActivateServiceFiltering

Call Associated/Non Call Processing
ApplyCharging CallInformationRequest
FurnishChargingInformation
RequestReportBCSMEvent ResetTimer
SendChargingInformation

Call Associated/Call Processing
CollectInformation Connect
ConnectToResource DisconnectForwardConnection
EstablishTemporaryConnection

Refer to section 8.2.7 MissingParameter for the appropriate error procedures.

8.2.19.3 Operations SSF-->SCF

AssistRequestInstructions
InitialDP
ApplyChargingReport

Refer to section 8.2.7 MissingParameter for the appropriate error procedures.

8.3 Entity related error procedures

The following sections define the error handling for the entity related errors. Since the error situations are not originated by the reception of an operation, the invoking entity is denoted here as the entity at which the error situation is detected. The responding entity is the entity which receives the error report.

Which TCAP services are used for reporting errors are described in section 10.

8.3.1 Expiration of T_{SSF}

8.3.1.1 General description

8.3.1.1.1 Error description

A timeout occurred in the SSF on the response from the SCF.

8.3.1.2 Procedures SSF->SCF

Procedure at the invoking entity (SSF)
Timeout occurs in SSF on Tssf

Precondition:	SSF FSM state c	Waiting for instructions
	or SSF FSM state d	Waiting for end of User Interaction
	or SSF FSM state e	Waiting for end of Temporary connection
Postcondition:	SSF FSM state a	Idle

The SSF-FSM aborts the dialogue and moves to the Idle state, the CCF routes the call if necessary (e.g. default routing to a terminating announcement). The abort is reported to the maintenance functions.

Procedure at the responding entity (SCF)
SCF receives a dialogue abort

Precondition:	Any state	
Postcondition:	SCSM state 1	Idle; if the abort is related to a SSF dialogue
	or SCSM state 2	Preparing SSF instructions; if the abort is related to an assisting SSF dialogue

The SCF releases all allocated resources and reports the abort to the maintenance functions, if the abort is received on a SSF dialogue.
The SCF releases all resources related to the dialogue, reports the abort to the maintenance functions and returns to state preparing SSF instructions, if the abort is received on an assisting SSF dialogue.-

8.3.2 Expiration of T_{SRF}

8.3.2.1 General Description

8.3.2.1.1 Error description

A timeout occurred in the SRF on the response from the SCF.

8.3.2.2 Procedures SRF->SCF

Procedure at the invoking entity (SRF)
Timeout occurs in SRF on Tsrf

Precondition:	SRSM state 2	Connected
	or SRSM state 3	User Interaction
Postcondition:	SRSM state 1	Idle

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

Procedure at the relaying entity (SSF)
SSF receives a bearer disconnection indication.
Precondition: SSF-FSM state d Waiting for End of User Interaction
Postcondition: SSF-FSM state c Waiting for Instructions
The SSF-FSM moves to state Waiting for Instructions. The timer Tssf is set. On expiration of Tssf refer to 8.3.1.

8.4 Call handling in error cases

The reception of a Reject component by the SSF shall result in aborting the TCAP dialogue. The call handling treatment shall be:

- *If the call is in the stable phase, that is the processing to select a circuit has already been successfully completed (the Connect operation has been completed and in addition no RouteSelectFailure event occured), the SSF shall continue autonomously with the call processing. The event treatment shall be as for a non IN call.*
- *If the call is not yet in the stable phase, the SSF shall release the call. The release cause #38 (Network out of order) shall be provided to the calling user.*

When an U-ABORT or P-ABORT primitive is received the SSF shall:

- *Release the call if it is not in the stable phase.*
- *If the call is in the stable phase, the SSF shall continue autonomously with the call processing. In the worst case the call is released e.g. if the SSF cannot continue autonomously with the call processing.*

9. Detailed procedures

The detailed procedures in this section contain a subsection "error handling". The text of this subsection is however restricted to error handling procedures specific for the operation which is described in this particular detailed procedure . In many cases , the error handling procedures are not operation specific. The generic error handling for the operation related errors are described in section 8 and the TCAP services which are used for reporting errors are described in section 10 Which of the errors apply to a particular operation is specified by the ASN.1 specification of every operation provided in section 6.1.

9.1. ActivateServiceFiltering procedure

9.1.1. General description

When receiving this operation, the SSF handles calls to destinations in a specified manner without request for instructions to the SCF. In the case of service filtering the SSF executes a specific service filtering algorithm. For the transfer of service filtering results refer to the operation 'ServiceFilteringResponse'.

9.1.1.1. Parameters

- filteredCallTreatment:
  This parameter specifies how filtered calls are treated. It includes information about the announcement to be played, the charging approach, the number of counters used and the release cause to be applied to filtered calls.

  - sFBillingChargingCharacteristics:
    This parameter determines the charging to be applied for service filtering. .
    It is possible to provide an sCPChargeNumber from the SCP which can have the values 0 or 2-255.
    In case the value 0 has been sent, all calls to be filtered are free of charge.
    In case the SCF has not sent any sCPChargeNumber, the calling line is not influenced by IN (no IN charge control).

  - informationToSend:
    This parameter indicates an announcement *or* a tone to be sent to the calling party. At the end of information sending, the call shall be released.

    - inbandInfo:
      This parameter specifies the inband information to be sent.

      - messageID:
        This parameter indicates the message(s) to be sent, it can be one of the following:

        - elementaryMessageID:
          This parameter indicates a single announcement. *Only values 0..126 are allowed. For details with respect to the handling of values outside this range refer to the error handling description below.*

        - elementaryMessageIDs:
          This parameter *is not supported for this operation. For details with respect to the receipt of this parameter refer to the error handling description below.*

        - variableMessage:
          This parameter *is not supported for this operation*. *For details with respect to the receipt of this parameter refer to the error handling description below.*

- 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. *Only values 1..255 are allowed*. *For details with respect to the handling of values outside this range refer to the error handling description below.* In case neither the 'numberOfRepetitions' nor 'duration' parameters are included, the SSP uses a default value 'numberOfRepetitions' = 1.

- interval:
  This parameter *is not supported for this operation. If received, it will be ignored.*

- tone:
  This parameter specifies a tone to be sent to the end-user.

  - toneID:
    This parameter indicates the tone to be sent.

  - duration:
    This parameter indicates the time duration in seconds of the tone to be sent. *Only values 1..255 are allowed*. *For details with respect to the handling of values outside this range refer to the error handling description below.*

- *userDialogInfo:*
  This parameter is not allowed to be sent for this operation. If received, it will be ignored.

- maximumNumberOfCounters:
  This parameter provides the number of counters to be allocated as well as the number of destinations included in the service filtering, i.e. 'maximumNumberOfCounters' subsequent destination addresses beginning with the destination address provided in 'filteringCriteria' are used for service filtering. One counter is assigned to each of these destination addresses.
  The number of counters may only be >1 if the 'filteringCriteria' are of the type 'addressAndService'.
  If the SCP specifies no 'maximumNumOfCounters' the SSP takes the default value 1. The maximal number of counters for one service filtering entity is 30.

- releaseCause:
  This parameter provides the cause value used for call release after the 'informationToSend' (for example announcement) has been sent to the calling party. If 'releaseCause' is not present, the default value is the same as the ISUP value decimal 31 (normal unspecified).

- filteringCharacteristics:
  This parameter indicates the severity of the filtering and the point in time when the 'ServiceFilteringResponse' shall be sent. It determines whether the 'interval' or the 'numberOfCalls' are used.

  - interval:
    After expiration of the interval timer the next call to arrive causes following actions:
    - sending of an 'InitialDP',
    - sending of an 'ServiceFilteringResponse',
    - starting again the interval timer.

    When filtering is started the first interval is started.

  - numberOfCalls:

The n'th call causes an 'InitialDP' and an 'ServiceFilteringResponse' operation sent to the SCF. This threshold value is met if the sum of all counters assigned to one service filtering entity is equal to 'numberOfCalls'.

- filteringTimeOut:
  This parameter indicates the duration of the filtering. When the time expires, a 'ServiceFilteringResponse' is sent to the SCF and service filtering is stopped. Two approaches are supported (duration or stopTime):

  - duration:
    If the duration time expires, then service filtering is stopped and the final report is sent to the SCF.
    A duration of 0 indicates that service filtering is to be removed.
    Other values indicate duration in seconds.

  - stopTime:
    When the 'stopTime' is met then service filtering is stopped and the final report is sent to the SCF. If 'stopTime' was already met, i.e. the value of the stopTime is less than the value of the actual time but the difference does not exceed the value equivalent to 50 years, then service filtering is immediately stopped and the actual counter values are reported to the SCF. This occurs in cases where the SCF wishes to explicitly stop a running service filtering.

- filteringCriteria:
  This parameter specifies which calls are filtered based on 'serviceKey' *and* 'calledAddressValue'

  - addressAndService:
    This parameter identifies the IN service and dialled number for which filtering should be applied.

    - serviceKey:

    - calledAddressValue:
      This parameter contains the dialled number towards which filtering shall be applied. The complete called party number shall be specified. *A maximum length of 15 digits is allowed. For details with respect to the handling of a longer digit string refer to the error handling description below.*

- startTime:
  This parameter defines when filtering is started. If 'startTime' is not provided or was already met, the SSF starts filtering immediately.

9.1.2. Invoking entity (SCF)

9.1.2.1. Normal procedure

SCF Precondition:
(1) SLPI detects that service filtering has to be initiated at the SSF.

SCF Postconditions:
(1) SLPI starts an application timer to monitor the expected end of service filtering.
(2) The SCME is in the state "Waiting For ServiceFilteringResponse".

Sending the 'ActivateServiceFiltering' operation causes a transition of the SCME from the state "Service Filtering Idle" to the state "Waiting For SSF Service Filtering Response". The SCME remains in this state until the application timer in the SLPI expires. The SCME is informed by the SLPI about timer expiration. Then it moves to the state "Service Filtering Idle".

If no errors occurred after receiving an 'ActivateServiceFiltering' at the SSF an empty Return Result is sent to the SCF. That causes no state transition in the SCME.

To change the parameters of an existing service filtering entity the SCF has to send an 'ActivateServiceFiltering' operation with the same 'filtering Criteria'. The second parameter set replaces the first one.

9.1.2.2. Error handling

If sending the ActivateServiceFiltering operation is not successful the SCME initiates one retransmission. If the sending fails a second time the SCME informs the SLPI about the error.

NOTE: Since the service filtering is not activated, all calls which should be filtered are handled as normal calls in the SSP. The flag 'SF encountered' is not set in the InitialDP for these calls.

9.1.3. Responding entity (SSF)

9.1.3.1. Normal procedure

SSF Precondition:
None.

SSF Postcondition:
(1) The SSME-FSM is in the state "Non Call Associated Treatment".

If there is no already existing SSME-FSM for the 'filteringCriteria' provided then a new SSME-FSM is created. This SSME-FSM enters the state "Non-Call Associated Treatment" and initialises the service filtering for the specified IN calls. The parameters 'filteredCallTreatment', 'filteringCharacteristics', 'filteringCriteria', 'filteringTimeOut' and 'startTime' are set as provided in the operation. A number of counters will be allocated and reset. In the case of the 'startTime' that has not been met yet, the service filtering will be started at the specified point in time.

If the operation 'ActivateServiceFiltering' addresses an already existing service filtering entity the parameters 'filteredCallTreatment', 'filteringCharacteristics' 'filteringTimeOut' and 'startTime' are modified as provided in the operation. In the case that the addressed service filtering entity is active the SSF reports the counter values to the SCF via the operation 'ServiceFilteringResponse'. The service filtering process is stopped if an already expired 'stopTime' or 'duration' equal to ZERO or a new not yet met 'startTime' is provided. The SSF then proceeds as described for 'ServiceFilteringResponse'. In the case of the 'startTime' that has not been met yet, the service filtering will be continued at the specified point in time.

If the service filtering proceeds then the SSME-FSM remains in the state "Non-Call Associated Treatment". Otherwise the SSME-FSM moves to state "Idle Management".

The maximal number of service filtering entities which are entered in the SSP at the same time is 100. One service filtering entity specifies the handling for a group of dialed numbers described by:

- *one filteringCriteria and*
- *the maximumNumberOfCounters.*

If no errors occurred after receiving an 'ActivateServiceFiltering' on the SSF an empty Return Result is sent to the SCF. That causes no state transition in the SSME-FSM.

Following application timers are used:
- detect moment to start service filtering (start time)
- duration time for service filtering
- interval time for service filtering (for timer controlled approach)

9.1.3.2. Error handling

If the SSF detects an error with any of the defined error values then this error is reported to the SCF.
The event is recorded in the SSF and an error condition indicated.

In case a new SSME-FSM should be created, the relationship is ended and all concerned resources are released. The SSME-FSM remains in / moves to the state "Idle Management".

In case there is already an existing SSME-FSM, the service filtering data remain unchanged. The SSME-FSM remains in the state "Non-Call Associated Treatment".

The error 'UnexpectedParameter' is sent by the SSP in the case of:

- *'messageID' contains any other choice than 'elementaryMessageID'.*

The error 'ParameterOutOfRange' is sent by the SSP in the case of:

- *'elementaryMessageID' is out of the restricted range 0...126 which is allowed for service filtering.*
- *'InbandInfo.duration' is out of the restricted range 1...255 which is allowed for service filtering.*
- *'Tone.duration' is out of the restricted range 1...255 which is allowed for service filtering.*
- *'calledAddressValue' contains more than 15 digits.*
- *coding of the octet string 'sFBillingChargingCharacteristics' is not correct.*

The error 'TaskRefused' is sent by the SSP if the SSP is not able to perform service filtering for the moment, e.g. in the case that it is not possible to update the service filtering data in the SSP or if the service filtering database in the SSP has reached its maximal extension.

9.2. ActivityTest procedure

9.2.1. General description

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

9.2.1.1. Parameters

None.

9.2.2. Invoking entity (SCF)

9.2.2.1. Normal procedure

SCF Preconditions:
(1) A *call associated* relationship exists between the SCF and the SSF.
(2) The activity test timer expires, after which the 'ActivityTest' operation is sent to the SSF.

SCF Postcondition:
(1) If a Return Result 'ActivityTest' is received, the SCME resets the activity test timer and takes no further action.

9.2.2.2. Error handling

If a time out on the 'ActivityTest' operation, a P-Abort *or an U-Abort (activityTestFailed)* is received from TCAP, this is an indication that the relationship with the SSF was somehow lost. If a time-out is received, SCF aborts the dialogue.

The SLPI that was the user of this dialogue *and the maintenance function* will be informed, the corresponding SCSM-FSM will move to the state "idle".

9.2.3. Responding entity (SSF)

9.2.3.1. Normal procedure

SSF Precondition:
(1) A *call associated* relationship exists between the SCF and the SSF.

SSF Postconditions:
(1) The SSME-FSM stays in, or moves to the state "Non Call Associated Treatment".
(2) If the Dialogue ID is active and if there is a SSF-FSM using the dialogue, the SSME sends a Return Result 'ActivityTest' to the SCF. If there are no other management activities, the SSME-FSM returns to the state "Idle Management", or
If the Dialogue ID is not active, the TCAP in the SSP will issue a P-Abort, the SSME will in that case never receive the 'ActivityTest' req.ind and thus will not be able to reply.

9.2.3.2. Error handling

If a corresponding relationship exists in the SSF, but there is no related call, the SSF will sent U-ABORT (activityTestFailed) to the SCF.

In case ActivityTest is received when the dialogue is about to be closed via prearranged end (e.g. after sending EventRportBCSM (EDP-N) or CallInformationReport), the SSF aborts the dialogue via U-ABORT (initiatingRelease).

If a SSF internal error is detected when processing the ActivityTest, the dialogue is aborted by the SSF. In the case of a call associated relationship the call will remain in the stable state.

9.3. ApplyCharging procedure

9.3.1. General Description

This operation is used for interacting from the SCF with the SSF charging mechanism. The operation ApplyChargingReport provides the feedback from the SSF to the SCF.

9.3.1.1. Parameters

For the ASN.1 representation of the parameters see Teil 3.

- aChBillingChargingCharacteristics:
  This parameter specifies the charging related information to be provided by the SSF and the conditions on which this information has to be reported back to the SCF via the operation ACR. For the ASN.1 representation of the parameters see Teil 3.

- partyToCharge
  This parameter indicates the party in the call to which the AC operation shall be applied. If it is not present, then AC applies to the A-party.

  Even in case the ‘partyToCharge’ is the A-party, the AC-operation is related to the connection SSP<->B-party (i.e. no supervision of connection A-party<->SSP). Therefore, in case of a release from the B-party, an ACR-operation is setup with a ‘sequenceInfo’ set to ‘final-callReleased’.

9.3.2. Invoking Entity (SCF)

9.3.2.1. Normal Procedure

SCF Preconditions:
(1) 'PreparingSSFInstructions',
or
(2) 'WaitingForNotificationOrRequest'

SCF Postconditions:
(1) No SCF-FSM state transition

If there is a control or monitor relationship between the SCF and the SSF, the SLPI can initiate the sending of the AC operation for requesting the SSF to support the charging of the subscriber. If the call duration is supervised by the SCF, a duration limit is granted to the SSF. If the limit is reached, the call will have been either already released by the SSF (forced release handling requested) or further instructions will be invoked by the SSF.

The SCF updates the limit according to the information received in the appropriate ACR operations.

The operation AC is sent:
- at the beginning of the accounting process,
- the SLPI has received ACR and the call being still alive shall be continued .

9.3.2.2. Error Handling

9.3.3. Responding Entity (SSF)

9.3.3.1. Normal Procedure

SSF Preconditions:

(1) The SSF FSM is in one of the following states:
"Waiting for Instructions" (state c); or
"Waiting for End of User Interaction" (state d); or
"Waiting for End of Temporary Connection" (state e); or
"Monitoring" (state f).

Note: The Precondition 'Monitoring' is not specified in ETSI Core INAP.

SSF Postconditions:
(1) No FSM state transition.

On receipt of this operation, the SSF sets the charging data using the information elements included in the operation. For tariff determination at SCF this comprises:

- A 'heartbeat' timer has to be started, if the SSF receives the answer from the B-party / IP.
  Note: The heartbeat timer is mentioned for backward compatibility reasons of the service logic /A.14/. In case that the account of the subscriber is devided into several partitions this parameter shall not be used any longer. Nevertheless, if the whole account is considered as one partition, the heart beat technique can be applied.

- Concerning the supervision of a time limit two cases have to be considered:

  a) AC operation during setting up a new call configuration

  Time measurement intended to supervise the time granted has to be started, if the SSF receives the answer from the B-party / IP.
  If supplied, the adviceOfCharge parameter shall be delivered to the CCF.

  b) AC operation during a stable connection

  The time supervision has to be restarted. I.e. the provided time limit (parameter timeGranted) reduced by time elapsed since the ACR operation (accuracy 1 second) has been sent will be supervised.

- If the changedCallSupervision parameter is present (i.e. a tariff switch is scheduled within the interval to be supervised), a timer task has to be immediately started according to the given tariff switch time. Upon expiration of this timer the actual elapsed call duration has to be stored in the parameter usedTime and the time measurement has to be started again. Upon expiration of the next supervision interval or upon call release, the call duration elapsed after the switch time is provided in the parameter usedTimeAfterSwitch.
  If the adviceOfCharge parameter is included, it will be applied at the point of tariff switch.

- If the remaining supervision time equals the timeBeforeRelease, the SSF requests the CCF to play the warning tone to the A-party.

- If the forcedRelease is provided, the call shall be released by the SSF immediately after the time limit granted by the SCF has been exhausted. The final values of the SSF based time measurement are reported to the SCF via ACR operation (final-callReleased).
  Otherwise (no forcedRelease provided), the current snap-shot of the time measurement is reported via ACR operation (final-callActive). Instead of releasing the call further instructions are requested from the SCF.

- If the timeBeforeRelease is bigger than timeGranted, the SSF shall request the CCF to play the warning tone to the A-party at once.

- The instructions of the AC operation are only valid for just one connection phase. As soon as the connection set-up is breaked-off, an ACR with final-callReleased indication is sent to the SCF.

9.3.3.2 Error Handling

If a coding error for a sub-parameter of the aChBillingChargingCharacteristics has been detected, and a default value is specified for this sub-parameter, and there is no special error handling specified (see below), then the default value shall apply. This procedure also applies for parameters, which are specified as OPTIONAL, but a network operator specific default value is foreseen.

- missingParameter:
  This error code is used in the following cases:

  - the operation does not contain an argument; this error may be sent

  - the parameter initTimeSupervision in TimeMeasurement is not present

  - the parameter callSupervision in AChBillingChargingCharacteristics is not present.

- unexpectedDataValue:
  This error code is used in the following cases:

  - the coding of the parameter callSupervision at all does not allow further decoding (tag errors, length errors, etc.)

  - timeMeasurement : coding error in the sub-parameters: timeGranted, switchTime, e-parameters of the CAI-GSM0224 type.

    the parameter timeGranted received equals ‘0’.

    the parameter switchTime equals ‘0’ is received before the connection is set up.

- unexpectedParameter
  This error code is used in the following case:

  - adviceOfCharge is provided in the parameter initTimeSupervision during a stable connection

9.4. ApplyChargingReport procedure

9.4.1. General Description

This operation is used by the SSF to report charging related information to the SCF as requested by the SCF using the operation ApplyCharging.

9.4.1.1. Parameters

For the ASN.1 representation of the parameters see Teil 3.

- sequenceInfo
  This parameter indicates the situation in the SSF which has caused the sending of the ACR operation:

  1. The indication ”intermediate” is sent in case the ACR was sent after expiration of the heartBeat timer.
  2. The indication ”final-callReleased” is sent in all situations in which the actual connection configuration will be released.
  3. The indication ”final-callActive” is sent in case the given time limit was reached and the SSF requests the next instruction from the SCF (the parameter forcedRelease has not been sent by the SCF).

- partyToCharge
  The partyToCharge has to be set to the same value as received in the related AC operation. If partyToCharge is not present in AC, then this parameter is not sent.

- supervisionResult
  The supervisionResult parameter provides the charging related information previously requested via the AC operation.

  - supervisionMethod
    This parameter contains at present only one choice:

    - timeMeasurement

      - usedTime
        In case of no tariff switch in the current supervision interval, the usedTime would contain the time elapsed either since the B-party / IP has answered or since the last tariff switch that occured before the current supervision interval.
        If there was a tariff switch in the current supervision interval, the usedTime would contain just the time elapsed until tariff switch.
        In case of unsuccessful connection setup, the value ‘0’ will be reorted to the SCF.

      - usedTimeAfterSwitch
        The parameter usedTimeAfterSwitch would only be sent if there was a tariff switch during the actual supervision interval. It contains the time elapsed since this tariff switch.

  - cause
    The parameter cause is only present, if the ACR is the final one for the regarded connection configuration (final-callReleased). It indicates the release cause.

9.4.2. Invoking Entity (SSF)

9.4.2.1. Normal Procedure

SSF Preconditions:

17.03.9899

The SSF is in one of the following states:
(1) 'WaitingForEndOfUserInteraction',
or
(2) 'Monitoring'
or
(3) 'WaitingForEndOfTemporaryConnection'.

Note: The Precondition 'Monitoring' is not specified in ETSI-Core INAP.

SSF Postconditions:
(1) 'Idle', if there are no outstanding reports to be sent,
or
(2) no state transition.

This operation is invoked, if a charging event has been detected that was requested by the SCF.

If the connection configuration does not change then no SSF-FSM state transition shall be performed. If the connection configuration changes, then the SSF-FSM

- shall move to state 'idle', if there is no other EDP armed and no report requests are pending, or otherwise
- shall remain in the same state.

If the event causing the 'ApplyChargingReport' is related to an armed EDP-R, then it shall be sent before the corresponding 'EventReportBCSM'.

The events relevant to be handled by the SSF are:

- Heartbeat time out: the call connection time is determined and provided in the usedTime and conditionally usedTimeAfterSwitch parameter, the heartbeat timer is restarted and the operation is invoked with 'intermediate' indication.
  Note: The heartbeat time out is mentioned for backward compatibility reasons of the service logic /A.14/. In case that the account of the subscriber is devided into several partitions this parameter shall not be used any longer. Nevertheless, if the whole account is considered as one partition, the heart beat technique can be applied.

- Time limit exhausted (supervision of timeGranted expired): the elapsed time measured is sent to the SCF in the usedTime and conditionally usedTimeAfterSwitch parameter;
  If the SSF requests the CCF to release the call (forced release handling required), then the SSF invokes the ACR operation with 'final-callReleased' indication. Otherwise, the call will not be released and an ACR operation with 'final-callActive' indication will be sent.
  The SSF shall set a guard timer (10 s) waiting for further instructions from the SCF.

- Expiration of the tariff switch timer: the actual snap-shot of the time measurement (i.e. the elapsed call duration) has to be stored in the parameter usedTime and the time measurement has to be restarted. Upon expiration of the next supervision interval or upon call release, the call duration elapsed after the switch time is provided in the parameter usedTimeAfterSwitch.

- Release indication from the A- or B-party (normal call release): the elapsed call connection time is determined and provided in the usedTime or usedTimeAfterSwitch parameter, the ACR operation is invoked with 'final-callReleased' indication.

- Release indication from the network (unsuccessful connection e.g. A-party abandons, B-party busy or no answer): the usedTime is set to 0 and the ACR operation is invoked with 'final-callReleased' indication.

- Internal failure situations: Any internal failure situation is handled in the same way as the normal call release.
  Note: In order to indicate the error situation to the SCF an U-Abort is sent immediately after the ACR operation.

- Release of a connection by the SCF (-> receipt of DFC or RC operation): The SSF reacts by sending an ACR operation with ‘final-callReleased’ indication.

The parameter usedTimeAfterSwitch would be always provided (in addition to usedTime) if a tariff switch occured during the current supervision interval (i.e. the time measurement is splitted into two separate intervals).The cause used to release the call shall be attached to the ACR operation only if the sequenceInfo indicates ‘final-callReleased’.

9.4.2.2. Error Handling

The following error situation can occur at IN application level:

U-Abort with UserInfo internalEventsSSF, general

- The usedTime or usedTimeAfterSwitch cannot be determined. In this case the call shall be released without warning tone. As release cause the value 41 'temporary failure' will be used.

U-Abort with UserInfo sSF-TimerExpired

- If the guard timer expires before receipt of a new AC operation, this will be handled as an internal failure situation (refer to paragraph ‘Normal Procedure’).

9.4.3. Responding Entity (SCF)

9.4.3.1 Normal Procedure

SCF Preconditions:
(1) State 'WaitingForNotificationOrRequest'.
or
(2) 'UserInteraction'

SCF Postconditions:
(1) No FSM state transition, if further reports are expected
or
(2) Transition to 'Idle', if the report is the last one.

On receipt of this operation, the SLPI which is expecting this operation will continue (e.g., - depending on the parameter sequenceInfo - the SLPI may either request the sending of the AC operation to inform the SSF about new call supervision data to be applied during the next supervision interval or complete its task and terminate).

With respect to the handling of the operation the parameter sequenceInfo has to be evaluated. Following cases have to be distinguished:

a) ACR - final-callActive

Another time limit to be granted via the next AC operation is expected by the SSF.

b) ACR - final-callReleased

The call has been already released either due to subscriber handling (on-hook) or by the SSF on request of the SCF (time limit exceeded). There is no response expected by the SSF.

c) ACR - intermediate

While the call is still alive intermediate data of the time measurement are provided in the course of long lasting calls for safety reasons. There is no response expected by the SSF.

Note: This parameter is mentioned for backward compatibility reasons of the service logic /A.14/. In case that the account of the subscriber is devided into several partitions this parameter shall not be used any longer. Nevertheless, if the whole account is considered as one partition, the heart beat technique can be applied.

9.4.3.2 Error Handling

- missingParameter
  This error code is used in the following cases:

  - the operation does not contain an argument; this error may be sent
  - the callSupervision parameter is missing .

- unexpectedComponentSequence
  This error code is used in the following case:

  - the operation is received in a state other than 'Waiting for Notification or Request'.

- unexpectedParameter
  This error code is used in the following cases:

  -
  - the cause parameter is present, but the ACR operation being of type 'intermediate' or ‘final-callActive’.

- unexpectedDataValue
  This error code is used in the following cases:

  - the coding of the callResult parameter at all does not allow further decoding (tag errors, length errors, etc.)
  - coding error in the sequenceInfo parameter
  - Accounting: coding error in the cause parameter
  - Accounting: coding error in the parameters of type MaximumCallDuration.

9.5 AssistRequestInstructions procedure

9.5.1 General description

This operation is sent to the SCF by an SSF, which is acting as the assisting SSF in an assist procedure. The operation is sent when the assisting SSF receives an indication from an initiating SSF containing information indicating an assist procedure.

9.5.1.1 Parameters

- correlationID:
  This parameter is used by the SCF to associate the AssistRequestInstructions from the assisting SSF with the InitialDP from the initiating SSF.

9.5.2 Invoking entity (SSF/SRF)

9.5.2.1 Normal procedure

SSF precondition:
(1) An assist indication is detected by the assisting SSF.

SSF postcondition:
(1) The assisting SSF waits for instructions.

On receipt of an assist indication from the initiating SSF shall assure that the required resources are available to invoke an AssistRequestInstructions operation in the SSF and indicate to the initiating SSF that the call is accepted. The AssistRequestInstructions operation is invoked by the SSF after the call, which initiated the assist indication, is accepted. *The call is accepted only if the parameters scfID and correlationID were received from the initiating SSF. The respective SCF where the AssistRequestInstruction operation is to sent to, is determined by the parameter scfID.* The assisting SSF FSM transitions to state "Waiting For Instructions". *Additionally the application timer TSSF is set.*

9.5.2.2 Error handling

9.5.3 Responding entity (SCF)

9.5.3.1 Normal procedure

SCF preconditions:
(1) A control relationship exists between the SCF and the initiating SSF.
(2) The SCF waits for AssistRequestInstructions.

SCF postcondition:
(1) An SSF instruction *followed by a SRF instruction* is being prepared.

On receipt of this operation in the SCSM state "Waiting for Assist Request Instructions", the SCP has to perform the following actions:

– If the AssistRequestInstructions operation was received from an assisting SSF, and the resource is available, the SCSM prepares the ConnectToResource and PlayAnnouncement or PromptAndCollect-UserInformation to be sent to the assisting SSF.

– The SCF determines SSF by means of "correlationID", "destinationNumber" or network knowledge.

– *The application timer TASSIST/HANDOFF is stopped.*

9.5.3.2 Error handling

On expiration of TASSIST/HANDOFF , the SCF FSM informs the SLPI and maintenance functions. The SCF FSM transitions to the state "Preparing SSF Instructions".

9.6. CallGap procedure

9.6.1. General description

This operation is used to request the SSF to reduce the rate at which specific service requests are sent to the SCF.

The operation is used for IN traffic managementpurposes. The operation has got an one way information flow from SCF to SSF. The operation invokes an automatic call gapping mechanism applicable to all new TCAP dialogues. Established TCAP dialogues are not affected by the requested call rate reduction method.

9.6.1.1. Parameters

- gapCriteria:
  This parameter identifies the criteria for a call to be subject to call gapping.

  - gapOnService:
    This parameter indicates that call gapping will be applied when the 'servicekey' of a call attempt match those specified in 'gapCriteria'.

- calledAddressAndService:
  This parameter indicates that call gapping will be applied when the 'servicekey' of a call attempt and the leading digits of the dialled number of a call attempt match those specified in ”gapCriteria”.
  It can be an incomplete number, in the sense that a limited amount of digits can be given. In that case the gapping will apply to all numbers which start with the same string of digits.

- gapIndicators:
  This parameter indicates the gapping characteristics.

  - duration:
    Duration specifies the total time interval during which call gapping for the specified gap criteria will be active.
    A duration of 0 indicates that gapping is to be removed.
    Other values indicate duration in seconds.

  - gapInterval:
    This parameter specifies the minimum time between calls being allowed through.
    An interval of 0 indicates that calls meeting the gap criteria are not to be rejected.
    An interval of -1 indicates that all calls meeting the gap criteria are to be rejected.
    Other values indicate interval in milliseconds.

- gapTreatment:
  This parameter indicates how calls that were stopped by the call gapping mechanism shall be treated.

  - informationToSend:
    This parameter indicates an announcement *or* a tone to be sent to the calling party. At the end of information sending, the call shall be released.

    - inbandInfo:
      This parameter specifies the inband information to be sent.

      - messageID:
        This parameter indicates the message(s) to be sent, it can be one of the following:

        - elementaryMessageID:
          This parameter indicates a single announcement. *Only values 0..126 are allowed to be sent. Others will be ignored, if they are received.*

        - elementaryMessageIDs:
          This parameter *is not supported for this operation*. *If received, it will be ignored.*

        - variableMessage:
          This parameter *is not supported for this operation*. *If received, it will be ignored.*

      - 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. *Only values 1..255 are allowed to be sent.* In

case neither the ‘numberOfRepetitions’ nor ‘duration’ parameters are included, the SSP uses a default value 'numberOfRepetitions' = 1.

- interval:
This parameter *is not supported for this operation*. *If received, it will be ignored.*

- tone:
This parameter specifies a tone to be sent to the end-user.

  - toneID:
  This parameter indicates the tone to be sent.

  - duration:
  This parameter indicates the time duration in seconds of the tone to be sent. *Only values 1..255 are allowed to be sent.*

- *userDialogInfo:*
This parameter is not allowed to be sent for this operation. If received, it will be ignored.

- releaseCause:
This parameter indicates that the call shall be released using the given release cause.

- both:
This parameter indicates inband info, a tone or display information to be sent to the calling party. At the end of information sending, the call shall be released, using the given release cause.

9.6.2. Invoking entity (SCF)

9.6.2.1. Normal procedure

SCF Preconditions:
(1) The SCF detects an overload condition persists and call gapping has to be initiated at the SSF.

SCF Postcondition:
(1) The SCME FSM remains in the same state upon issuing the 'CallGap' operation.

A congestion detection and control algorithm monitors the load of SCP resources. After detection of a congestion situation the parameters for the 'CallGap' operation are provided *taking into account former CallGap requests*.

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

9.6.2.2. Error handling

9.6.3. Responding entity (SSF)

9.6.3.1. Normal procedure

SSF Preconditions:
(1) Call gapping for gapCriteria is not active, or
Call gapping for gapCriteria is active.

SSF Postconditions:
(1) The SSME-FSM is in the state '"Non call associated treatment".
(2) Call gapping for gapCriteria is activated, or
Call gapping for gapCriteria is renewed, or
Call gapping for gapCriteria is removed.

If there is no already existing SSME-FSM for the gap criteria provided the a new SSME-FSM is created. This SSME-FSM enters the state 'Non call associated treatment' and initialises call gapping for the specified IN calls. The parameters 'gapIndicators' and 'gapTreatment' for the indicated gap criteria will be set as provided by the 'CallGap' operation.

- If an SSME-FSM already exists for the 'gapCriteria' provided, then

  the new parameters (i.e., 'gapIndicators' and 'gapTreatment') will overwrite the existing parameter values.

If the SSF meets a TDP, it will check if call gapping was initiated either for the 'serviceKey' or for the 'calledAddressValue' assigned to this TDP. If not, an 'InitialDP' operation can be sent.

If a call to a controlled called number matches several active 'gapCriteria', then only the 'gapCriteria' associated with the longest called party number should be used. For example, the codes 1234 and 12345 are under control. Then the call with 123456 is subject to the control on 12345. *If a call to a controlled called number arrives when the gap interval time expires the call will be allowed to pass and a new gap interval will be opened. The actual value of the gap interval will be included into the InitialDP operation issued for this call.*

If a gap interval is active, no 'InitialDP' is sent and the call is treated as indicated by 'gapTreatment'.

The call gap process is stopped if the indicated duration equals ZERO.

If the duration of a ´gap Criteria´ expires, no gapping applies any more.

If call gapping proceeds then the SSME-FSM remains in the state "Non call associated treatment". Otherwise, the SSME-FSM moves to state 'idle management'.

9.6.3.2. Error handling

9.7. CallInformationReport procedure

9.7.1. General description

This operation is used to send specific call information for a single call to the SCF as requested by the SCF in a previous 'CallInformationRequest' operation.

9.7.1.1. Parameters

- requestedInformationList:
  According to the requested information the SSF sends the appropriate types and values to the SCF.

9.7.2. Invoking entity (SSF)

9.7.2.1. Normal procedure

SSF Preconditions:
(1) At least one party disconnects from a call or the connection to the SRF is released.
(2) Requested call information has been collected.
(3) 'CallInformationReport' is pending due to a previously received 'CallInformationRequest' operation.
(4) A control relationship exists between the SCF and the SSF.

SSF Postcondition:
(1) SSF is in the state "Wait for Instructions" or "Idle".

If the SSF FSM executes a state transition caused by one of the following events:

- A party release
- A party abandon
- B party release
- B party busy
- SSF no answer timer expiration
- route select failure indicated by the network
- release call initiated by the SCF,

and a 'CallInformationRequest' is pending then the 'CallInformationReport' operation is sent to the SCF.

If a 'CallInformationReport' has been sent to the SCF then no 'CallInformationReport' is pending, i.e. a further 'CallInformationReport', for example in the case of follow-on, has to be explicitly requested by the SCF.

If the event causing the 'CallInformationReport' is related to an armed EDP-R then it shall be sent before the corresponding 'EventReportBCSM'.

9.7.2.2. Error handling

9.7.3. Responding entity (SCF)

9.7.3.1. Normal procedure

SCF Preconditions:
(1) An SLPI is expecting 'CallInformationReport'.
(2) A control relationship exists between the SCF and the SSF.

SCF Postcondition:
(1) The SLPI may be further executed.

In any state (except 'Idle') the SCSM may receive 'CallInformationReport' from the SSF, when the 'CallInformationReport' is outstanding.

If 'CallInformationReport' is outstanding and the service logic program indicates that the processing has been completed, the SCSM remains in the same state until it receives the 'CallInformationReport' operation.

When the SCF receives the 'CallInformationReport' operation and the service logic processing has been completed, then the SCSM moves to the "Idle" state.

When the SCF receives the 'CallInformationReport' operation and the service logic processing has not been completed yet, then the SCSM moves to the "Preparing SSF Instructions" state (no EventReportBCSM pending) or remains in the "Waiting for Notification or Report" state (EventReportBCSM pending).

9.7.3.2. Error handling

9.8. CallInformationRequest procedure

9.8.1. General description

This operation is used to request the SSF to record specific information about a single call and report it to the SCF using the 'CallInformationReport' operation.

9.8.1.1. Parameters

- requestedInformationTypeList:
  This parameter specifies a list of specific items of information which is requested.
  The list may contain:

- callAttemptElapsedTime:
  This parameter indicates the duration between the end of INAP processing of operations initiating call setup ('Connect', 'ConnectToResource') and the received answer indication from the called party side.
  In case of unsuccessful call setup the network event indicating the unsuccessful call setup stops the measurement of 'callAttemptElapsedTime'.

- callStopTime:
  This parameter indicates the time stamp when the connection is released.

- callConnectedElapsedTime:
  This parameter indicates the duration between the received answer indication from the called party side and the release of the connection.

- releaseCause:
  This parameter indicates the release cause for the call.

Any set of these values can be requested.

9.8.2. Invoking entity (SCF)

9.8.2.1. Normal procedure

SCF Preconditions:
(1) A control relationship exists between the SCF and the SSF.
(2) The SLPI has determined that a 'CallInformationRequest' operation has to be sent by the SCF.

SCF Postcondition:
(1) The SLPI is expecting a 'CallInformationReport' from SSF.

When the service logic program requests call information, the SCF sends the 'CallInformationRequest' operation to the SSF to request the SSF to provide call related information.

The 'CallInformationRequest' operation specifies the information items to be provided by the SSF.

9.8.2.2. Error handling

9.8.3. Responding entity (SSF)

9.8.3.1. Normal procedure

SSF Preconditions:
(1) Call origination attempt has been initiated.
(2) A control relationship exists between SSF and SCF.

SSF Postconditions:
(1) Requested call information is retained by the SSF.
(2) The SSF is waiting for further instructions.

The SSF may receive the 'CallInformationRequest' operation within an existing call associated (CA) dialogue only.

The 'CallInformationRequest' operation is accepted by the SSF Finite State Machine (SSF-FSM) only in the state 'Waiting for Instructions'. The operation does not lead to any transition to another state.

The SSF allocates a record and stores the requested information if already available and prepares the recording of information items, that will become available later like for example 'callStopTimeValue'.

9.8.3.2. Error handling

In any other than the 'Waiting for Instruction' state the 'CallInformationRequest' operation will be handled as 'out of context'.

9.10. CollectInformation procedure

9.10.1. General description

This is a class 2 operation which is used to requested the SSF to perform the basic originating call processing actions which will collect destination information from a calling party. (It is normally associated with a 'RequestReportBCSMEvent' operation to arm DP2 and to specify the number of digits to be collected.)

This operation uses only the resources of the SSF/CCF to collect the information, unlike 'PromptAndCollectUserInformation', which uses the capabilities of the SRF. It follows that the use of this operation is only appropriate for a call which has not yet left the setup phase.

9.10.1.1. Parameters

- None

9.10.2. Invoking entity (SCF)

9.10.2.1. Normal procedure

SCF Precondition:
(1) An SLPI has determined that more information from the calling party is required to enable processing to proceed.

SCF Postcondition:
(1) SLPI execution is suspended pending receipt of dialled digits.

This operation is invoked in the SCSM FSM state "Preparing SSF Instructions" if the SLP requires additional information to progress the call. It causes a transition of the FSM to the state "Waiting for Notification or Report".

9.10.2.2. Error handling

9.10.3. Responding entity (SSF)

9.10.3.1. Normal procedure

SSF Precondition:
(1) An 'InitialDP' operation has been invoked.

SSF Postconditions:
(1) The SSF has executed a transition to the state "Monitoring".
(2) The SSF performs the call processing actions to collect destination information from the calling party.

The operation is only valid in the state "Waiting for Instruction" and after having received an operation 'RequestReportBCSMEvent' for DP2.. The SSP has to perform the following actions:

- The SSF cancels T_{SSF}.

- When the requisite number of digits (specified when DP2 was armed) has been received, DP2 will be encountered, an 'EventReportBCSM' operation will be invoked, and the SSF FSM will return to the state "Waiting for Instruction".

9.10.3.2. Error handling

If the operation is received in an invalid SSF FSM state (., not in state 'Waiting for Instruction' or the SSF FSM has already received a Connect, ConnectToResource or EstablishTemporaryConnection operation), the operation will be rejected with 'UnexpectedComponentSequence'.

If no RequestReportBCSM operation was received for DP2 (requestmode), the operation will be rejected with 'UnexpectedComponentSequence'.

If any of the following operation has been received before CollectInformation, the operation will be rejected with 'UnexpectedComponentSequence': CallInformationReport, FurnishChargingInformation, SendChargingInformation.

If there are problems to allocate/access needed resources 'TaskRefused' is sent to the SCP.

9.11. Connect procedure

9.11.1. General description

This operation is used to request the SSF to perform the call processing actions to route a call to a specific destination. To do so, the SSF may use destination information from the calling party (e.g. dialled digits) and existing call set-up information depending on the information provided by the SCF.

9.11.1.1. Parameters

- destinationRoutingAddress:
  This parameter contains the called party number towards which the call is to be routed. The encoding of the parameter is defined in Q.763.
  If this parameter contains no destination digits, the SSF will use the dialed digits to route the call. In this case, 'cutAndPaste' is mandatory.

- cutAndPaste:
  This parameter is used by the SCF to instruct the SSF to delete (cut) a specified number of leading digits that it has received from the calling party and to paste the remaining digits on to the end of the digits supplied by the SCF in the 'destinationRoutingAddress'.

- callingPartyNumber:
  This parameter is used to provide an alternative to the "callingPartyNumber" supplied by the network. It may be used for applications such as UPT, where only the SCF can verify the identity of the calling party. See [7] regarding the mapping on ISUP.

- callingPartysCategory:
  This parameter indicates the type of calling party (e.g., operator, pay phone, ordinary subscriber). See [7] regarding the mapping on ISUP.

- originalCalledPartyID:
  This parameter carries the dialled digits if the call has met call forwarding on route to the SSP or is forwarded by the SCP. See [7] regarding the mapping on ISUP.

- serviceInteractionIndicatorsTwo:
  Indicators which are exchanged between SSP and SCP to resolve interactions between IN based services and network based services, respectively between different IN based services.

  Note: The ServiceInteractionIndicatorsTwo parameter and the Nat ServiceInteractionIndicators can be received in the same INAP operation. In case, the interaction indicators provided in both parameters control the same interworking procedures, the interaction indicators provided by ServiceInteractionIndicatorsTwo have precedence. For detailed interworking procedures refer to [7].

- BackwardServiceInteractionInd

  -- conferenceTreatmentIndicator
  This parameter controls the acceptance of conference requests.

- ForwardServiceInteractionInd:

  -- conferenceTreatmentIndicator:
  This parameter controls the acceptance of conference requests.

  -- callDiversionTreatmentIndicator:
  This parameter controls the acceptance of call diversion requests.

  -- callOfferingTreatmentIndicator:
  This parameter controls the call offering procedures at the terminating local exchange. This parameter is not used by T-Mobil.

- connectedNumberTreatmentInd
  This parameter controls the behavior of the ISUP supplementary service COLP/COLR. This parameter is not used by T-Mobil.

- allowCdINNoPresentationInd:
  This parameter controls the Number Presentation not allowed indicator of ISUP messages. This parameter is not used by T-Mobil.

- redirectingPartyID:
  This parameter indicates the directory number the call was redirected from. See [7] regarding the mapping on ISUP.

- redirectionInformation:
  This parameter contains forwarding related information, such as redirecting counter. See [7] regarding the mapping on ISUP.

- *natServiceInteractionIndicators:*
  This parameter contains national indicators sent from the SCP to the SSP for control of the network based services at the originating exchange and the destination exchange. For the details with respect to coding and procedural aspects refer to Teil 3.

- *natCallingPartysCategory:*
  This parameter indicates the type of calling party. It may contain Telekom specific types in addition to the parameter CallingPartysCategory. See [7] regarding the mapping on ISUP.

- *INContainer:*
  This parameter may be used by a service logic to provide a message in forward direction to another service logic which is invoked later for the same call. This message may be sent using the Connect operation. It will be transported transparently through the basic network an will be presented to the next service logic within the operation InitialDP.

- *userUser:*
  This parameter contains a user-to-user information provided by the service logic. It will be forwarded by the SSP as a user-to-user information in the appropriate ISUP message. See [7] regarding the mapping on ISUP.

9.11.2. Invoking entity (SCF)

9.11.2.1. Normal procedure

SCF Preconditions:
17.03.989

(1) A control relationship exists between the SCF and the SSF.
(2) An SLPI has determined that a 'Connect' has to be sent by the SCF.

SCF Postcondition:
(1) SLPI execution may continue.

In the SCSM FSM state "Preparing SSF Instructions", this operation is invoked by the SCF if the service logic results in the request to the SSF to route a call to a specific destination. If no event monitoring has been requested in a previously sent operation, a SCSM FSM transition to state "Idle" occurs. Otherwise, if event monitoring has been requested, the SCSM FSM transitions to state "Waiting for Notification or Report".

9.11.2.2. Error handling

If reject or error messages are received, then the SCSM informs the SLPI and remains in the state "Preparing SSF Instructions".

9.11.3. Responding entity (SSF)

9.11.3.1. Normal procedure

SSF Preconditions:
(1) Call origination attempt has been initiated.
(2) Basic call processing has been suspended at a DP.
(3) The SSF waits for instructions.

SSF Postcondition:
(1) The SSF performs the call processing actions to route the call to the specified destination.
(2) In the O-BCSM, call processing resumes at PIC 3.
(3) In the T-BCSM, when the Connect operation is received with a DestinationRoutingAddress, then a new O-BCSM shall be created and chained to the T-BCSM. The T-BCSM shall pass the information available (e.g. new number to which the call is to be routed) to the O-BCSM.

On receipt of this operation in the SSF FSM state "Waiting for Instructions", the SSP performs the following actions:

- The SSF cancels T_{SSF}.
- If 'cutAndPaste' is present, then the SSF deletes ('cut') from the address signal field in the called party number available in the SSF/CCF the indicated number of digits and pastes the remaining dialled digits at the end of the 'destinationRoutingAddress' parameter delivered by the SCF. The resulting directory number is used for routing to complete the related call.
- If 'cutAndPaste' is not present, then the 'destinationRoutingAddress' parameter delivered by the SCF is used for routing to complete the related call.
- If no EDPs have been armed, the FSM goes to state "Idle" (e9). Otherwise, the FSM goes to state "Monitoring" (e11).

No implicit activation or deactivation of DPs occurs.

Statistic counter(s) are not affected.

Connect completes when the INAP processing of the operation is complete and before the SSP starts the processing necessary to select a circuit.

Therefore in order to detect route select failure after a 'Connect' it is necessary to explicitly arm the 'Route Select Failure' EDP before sending the 'Connect' (although they may be in the same message).

9.11.3.2. Error handling

MissingParameter:

The operation contains no parameters at all. Missing CutAndPaste parameter if no destination digits are included.

UnexpectedDataValue:

The parameter destinationRoutingAddress contains digits but cutAndPaste = 0. The parameter destinationRoutingAddress contains any other unexpected value.

If the requested connection cannot be established by the CCF due to either an error of the CCF in the SSP or due to an error in the network, the SCF will be informed of this failure in different ways. In any case, the SSP closes the TCAP-dialogue. If a CallInformationReport has been requested, the SSP closes the TCAP-dialogue by prearranged end after sending the CallInformationReport with the release cause indication. If no CallInformationReport has been requested, but an EDP has been armed, which cannot be encountered due to call release, the SSP will abort the TCAP-dialogue and include the release cause into the UserAbortInformation.

9.12. ConnectToResource procedure

9.12.1. General description

This operation is used to connect a call from the SSF to a specialised resource. After successful connection to the SRF, the interaction with the caller can take place. The SSF relays all operations for the SRF and all responses from the SRF.

9.12.1.1. Parameters

- resourceAddress:
  This parameter identifies the physical location of the SRF. *Two approaches are supported:*

  - iPRoutingAddress:
    This parameter indicates the routing address to set up a connection towards the SRF.

  - none:
    This parameter indicates that the call party is to be connected to a predefined SRF.

- *natServiceInteractionIndicators:*
  This parameter contains nationally defined interaction indicators as e.g. a 'longUIDInd'. For the details with respect to coding and procedural aspects refer to Teil 3.

- serviceInteractionIndicatorsTwo:
  Indicators which are exchanged between SSP and SCP to resolve interactions between IN based services and network based services, respectively between different IN based services.

  Note: The ServiceInteractionIndicatorsTwo parameter and the Nat ServiceInteractionIndicators can be received in the same INAP operation. In case, the interaction indicators provided in both parameters control the same interworking procedures, the interaction indicators provided by ServiceInteractionIndicatorsTwo have precedence. For detailed interworking procedures refer to [7].

  - BackwardServiceInteractionInd:

    -- conferenceTreatmentIndicator:
       This parameter controls the acceptance of conference requests.

- UserDialogueDurationInd
  This parameter indicates whether the user dialogue may last long (e.g. longer than 90 seconds).

9.12.2. Invoking entity (SCF)

9.12.2.1. Normal procedure

SCF Preconditions:
(1) A control relationship exists between the SCF and the SSF.
(2) The SLPI has determined that additional information from the call party is needed.
(3) The SCSM-FSM is in the state "Routing to Resource", substate "Determine Mode".

SCF Postconditions:
(1) The SCSM sends out a 'PlayAnnouncement' or 'PromptAndCollectUserInformation' operation accompanying the 'ConnectToResource'.
(2) The SCSM-FSM moves to the state "User Interaction".

ConnectToResource operation is generated in the state 'Routing to Resource' on reaction to the event 'SR Facility Needed' . This indicates the SSP relay case, i.e. the SSF transmits the UI operations to the SRF.

In case the ConnectToResource operation is sent to an assisting SSF, the parameter "longUIDInd" is always set to the default value TRUE.

9.12.2.2. Error handling

9.12.3. Responding entity (SSF)

9.12.3.1. Normal procedure

SSF Preconditions:
(1) Basic call processing has been suspended at a DP and a control relationship has been established.
(2) The SSF-FSM is in the state "Waiting for Instructions".

SSF Postconditions:
(1) The call is switched to the SRF.
(2) A control relationship to the SRF is established.
(3) The SSF-FSM moves to the state "Waiting for End of User Interaction". T_{SSF} is set.

The receipt ConnectToResource operation results in switching the concerned call to the physical entity containing the SRF. This operation applies only to the SSP relay case, i.e. the SSF transmits the UI operations to the SRF. The relationship SCF-SRF is implicitly established.

NOTE: The successful connection to the SRF causes a state transition in the SRF FSM from "Idle" to "Connected".

If the parameter 'longUIDialogue'/'userDialogueDurationInd' is set to TRUE always a premature Answer signal is sent by the SSF before starting the UI-dialogue.

9.12.3.2. Error handling

TaskRefused:

The connection to the IP can not be esthablished (detected by SSF).

MissingParameter:

The operation contains no parameters.

If the requested connection to the SRF cannot be established by the CCF due to an error of the CCF in the SSP, the SSP releases the call and aborts the TCAP-dialogue.

9.13 Continue procedure

9.13.1 General description

This operation is used to request the SSF to proceed with call processing at the DP at which it previously suspended call processing to await SCF instructions. The SSF continues call processing without substituting new data from the SCF.

9.13.1.1 Parameters

None.

9.13.2 Invoking entity (SCF)

9.13.2.1 Normal procedure

SCF precondition:
(1) SCSM is in the state "Preparing SSF instructions".

SCF postcondition:
(1) SCSM is in the state "Waiting for Notification of Report", in case monitoring was required, or in the state "Idle", in case no monitoring was required.

The SCSM is in state "Preparing SSF instructions". The Continue operation is invoked by a SLPI. This causes a SCSM transition to state "Idle" if no subsequent monitoring is required. However, if monitoring is required, like in the case of armed EDPs or outstanding report requests, the SCSM transitions to state "Waiting for Notification of Report".

9.13.2.2 Error handling

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

9.13.3 Responding entity (SSF)

9.13.3.1 Normal procedure

SSF preconditions
(1) BCSM: Basic call processing has been suspended at any DP.
(2) SSF FSM is in the state "Waiting for Instructions".

SSF postconditions
(1) BCSM: Basic call processing continues.
(2) SSF FSM is in the state "Monitoring", because at least one EDP was armed, or a CallInformationReport or ApplyChargingReport was requested; or
SSF FSM is in the state "Idle", because no EDPs were armed and neither the CallInformationReport nor the ApplyChargingReport was requested.

The SSF FSM is in state "Waiting for instructions". The SSME receives the Continue operation and relays it to the appropriate SSF FSM. The SSF FSM transitions to state "Idle" in case no EDPs are armed and no outstanding report requests are present. The SSF FSM transits to state "Monitoring" if at least one EDP is armed, or if there is at least one outstanding report request. Basic call processing is resumed.

9.13.3.2 Error handling

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

9.14. DisconnectForwardConnection procedure

9.14.1. General description

This operation is used in the following two cases:

1) To clear a connection to a SRF

This operation is used to explicitly disconnect a connection to a resource (SRF) established previously with a 'ConnectToResource' operation. It is used for a forward disconnection from the SSF. An alternative solution is the backward disconnect from the SRF, controlled by the 'DisconnectFromIPForbidden' parameter in the 'PlayAnnouncement' and 'PromptAndCollectUserInformation' operations.

2) To clear a connection to an assisting SSF

This operation is sent to the non-assisting SSF of a pair of SSFs involved in an assist procedure. It is used to disconnect the temporary connection between the initiating SSF and the assisting SSF, and the assisting SSF and its associated SRF.

9.14.1.1. Parameters

- none

9.14.2. Invoking entity (SCF)

9.14.2.1. Normal procedure

SCF Preconditions:
(1) A control relationship exists between the SCF and the SSF.
(2) An assist- or a relay procedure is in progress.
(3) An SLPI has determined that a 'DisconnectForwardConnection' operation has to be sent by the SCF.

SCF Postcondition:
(1) SLPI execution may continue.

The 'DisconnectForwardConnection' operation is used to instruct the SSF to disconnect the concerned forward connection to the assisting SSF or the physical entity containing the SRF.

In the SCSM FSM state "User Interaction", substate "Waiting for Response from the SRF", this operation is invoked by the SCF when the service logic determines that user interaction is finished and requests the SSF to disconnect the temporary connection to the assisting SSF or the SRF. The SCSM FSM then transitions to state "Preparing SSF Instructions".

The 'DisconnectForwardConnection' operation contains no parameter since there may be only one SRF connection to one call.

9.14.2.2. Error handling

9.14.3. Responding entity (SSF)

9.14.3.1. Normal procedure

SSF Preconditions:
(1) Call origination attempt has been initiated.
(2) Basic call processing has been suspended at a DP.
(3) The initiating SSF is in the state "Waiting for End of User Interaction" or "Waiting for End of Temporary Connection".

SSF Postconditions:
(1) The connection to the SRF or assisting SSF is released.
(2) The SSF is waiting for instructions.

The receipt of 'DisconnectForwardConnection' results in disconnecting the assisting SSF orthe physical entity containing the SRF from the concerned call. It does not release the connection from the SSF back to the end user.

This operation is accepted in the SSF FSM states "Waiting for End of Temporary Connection" or "Waiting for End of User Interaction". On receipt of this operation in these states , the SSP must perform the following actions:

- The initiating SSF releases the connection to the assisting SSF or the relay SRF.

- The SSF resets T_{SSF}.

- The SSF FSM goes to state "Waiting for Instructions" (e8).

The 'DisconnectForwardConnection' operation contains no parameters.

NOTE: The successful disconnection to the SRF causes a state transition in the SRF FSM to "Idle". A current order ('PlayAnnouncement' or 'PromptAndCollectUserInformation') is cancelled.

9.14.3.2. Error handling

9.15 EstablishTemporaryConnection procedure

9.15.1 General description

This operation is used to create a connection between an initiating SSF and an assisting SSF as part of a service assist procedure.

9.15.1.1 Parameters

- assistingSSPIPRoutingAddress:
  This parameter indicates the destination address of the *SSF* for assist procedure.

- correlationID:
  This parameter is used by the SCF to associate the AssistRequestInstructions from the assisting SSF (or the SRF) with the InitialDP from the initiating SSF.

- scfID:
  This parameter indicates the SCF identifier and enables the assisting SSF to identify which SCF the AssistRequestInstructions should be sent to.

- serviceInteractionIndicatorsTwo:
  Indicators which are exchanged between SSP and SCP to resolve interactions between IN based services and network based services, respectively between different IN based services.

  Note: The ServiceInteractionIndicatorsTwo parameter and the Nat ServiceInteractionIndicators can be received in the same INAP operation. In case, the interaction indicators provided in both parameters control the same interworking procedures, the interaction indicators provided by ServiceInteractionIndicatorsTwo have precedence. For detailed interworking procedures refer to [7].

  - BackwardServiceInteractionInd

-- conferenceTreatmentIndicator
This parameter controls the acceptance of conference requests.

- UserDialogueDurationInd
This parameter indicates whether the user dialogue may last long (e.g. longer than 90 seconds).

9.15.2 Invoking entity (SCF)

9.15.2.1 Normal procedure

SCF preconditions:
(1) A control relationship exists between the SCF and the SSF.
(2) The SL has determined that a connection is needed between the SSF and an assisting SSF.
(3) The call party is not connected to any other party.

SCF postcondition:
(1) The SCF is "Waiting for Assist Request Instructions".

In the SCSM FSM state "Routing to Resource", this operation is invoked by the SCF when the SL determines that an assisting SSF is needed. The SCSM FSM then transitions to state "Waiting for Assisting Requested Instructions".

Additionally the SCSM sets the application timer TASSIST/HANDOFF to prevent excessive assist suspension time.

9.15.2.2 Error handling

9.15.3 Responding entity (SSF)

9.15.3.1 Normal procedure

SSF preconditions:
(1) Call origination attempt has been initiated.
(2) Basic call processing has been suspended at a DP.
(3) The SSF waits for instructions.
(4) The SSF is not an assisting SSF.

SSF postconditions:
(1) The SSF performs the call processing actions to route the call to the assisting SSF.
(2) The SSF waits for end of temporary connection.

On receipt of this operation in the SSF FSM state "Waiting for Instructions", the SSP has to perform the following actions:

- reset the T_{SSF};

- route the call to assisting SSF using "assistingSSPIPRoutingAddress";

- the SSF FSM goes to state "Waiting for End of Temporary Connection" (e7).

9.15.3.2 Error handling

Until the connection setup has been accepted by the assisting SSF, all received failure indications from the network on the ETC establishment shall be reported to the SCF as ETC error ETCFailed (e.g., busy, congestion). The operation timer for ETC shall be longer than the maximum allowed time for the signalling procedures to accept the connection.

9.17. EventReportBCSM procedure

9.17.1. General description

This operation is used to notify the SCF of a call related event previously requested by the SCF in an 'RequestReportBCSMEvent' operation. The monitoring of more than one event could be requested with a 'RequestReportBCSMEvent' operation, but each of these requested events is reported in a separate 'EventReportBCSM' operation.

9.17.1.1. Parameters

- eventTypeBCSM:
  This parameter specifies the type of event that is reported.

- eventSpecificInformationBCSM:
  This parameter indicates the call related information specific to the event.
  For 'CollectedInfo' it will contain the 'calledPartyNumber'.
  NOTE: All received digits are included in this parameter
  For RouteSelectFailure it will contain the "FailureCause", if available.
  For O- or T-CalledPartyBusy it will contain the 'BusyCause', if available.
  For O- or T-NoAnswer it will be empty.
  For T-Answer it will be empty.
  For O-Disconnect it will contain the 'releaseCause' and/or the ‘userUser’, if available.
  For T-Disconnect it will contain the 'releaseCause', if available.

- legID:
  This parameters indicates the party in the call for which the event is reported. SSF will use the option 'ReceivingSideID' only.

  - receivingSideID:
    The following values for 'legID' are assumed:
    'legID' = 1 indicates the party that was present at the moment of the 'InitialDP' *(A-party)*.
    'legID' = 2 indicates the party that was created with a 'Connect' operation *(B-party)*.

    If not included, the following defaults are assumed:
    legID = 1 for the event CollectInfo, O- and T- Abandon
    legID = 2 for the events RouteSelectFailure, O-CalledPartyBusy, O-NoAnswer, O-Answer, T-CalledPartyBusy, T-NoAnswer, T-Answer.
    The 'legID' parameter shall always be included for the events O-Disconnect and T-Disconnect.
    Only legID = 2 is allowed in case O- or T-Disconnect is armed as an EDP-R.

- miscCallInfo:
  This parameter indicates detection point related information.

  - messageType:
    This parameter indicates whether the message is a request, i.e. resulting from a 'RequestReportBCSMEvent' with monitorMode = interrupted, or a notification, i.e. resulting from a 'RequestReportBCSMEvent' with 'monitorMode' = 'notifyAndContinue'. *If omitted request type is assumed.*

9.17.2. Invoking entity (SSF)

9.17.2.1. Normal procedure

SSF Preconditions:
(1) The SSF-FSM is in the state "Monitoring".
(2) The BCSM proceeds to an EDP that is armed.

SSF Postconditions:

(1) The SSF-FSM stays in the state "Monitoring" if the message type was notification and there are still EDPs armed or a 'CallInformationReport' or 'ApplyChargingReport' requested.
(2) The SSF-FSM moves to the state 'idle' if the message type was notification and there are no more EDPs armed, no 'CallInformationReport' or 'ApplyChargingReport' are requested.
(3) The SSF-FSM moves to the state "Waiting for Instructions" if the message type was request. Call processing is interrupted. *The application timer Tssf is set.* All other outstanding reports ('CallInformationReport' and/or 'ApplyChargingReport') have to be sent before the 'EventReportBCSM' operation.

For certain service features it is necessary that the same O-BCSM instance is reused. Examples are follow on calls.

The decision to reuse the same O-BCSM instance can only be taken by the SCF after certain armed EDP-R's are reported.

The considered EDP-R's are:

- RouteSelectFailure
- O_CalledPartyBusy
- O_NoAnswer
- O_Disconnect *(only for B-party)*

If a DP armed as EDP-R or EDP-N has been detected which does not cause the release of the related leg, only all the alternative DP's related to that leg will be disarmed

If a*n* EDP-R *or an EDP-N* is met that causes the release of the related leg all EDPs related to that leg are disarmed and the event is reported via 'EventReportBCSM'. To allow the reuse of the same O-BCSM instance the BCSM has to store all call related signalling parameters (e.g. 'callingPartyNumber', 'callingPartysCategory') until the BCSM instance is released.

In the case the respective EDP-R is met (see list above with 'O-Disconnect' only for legID = 2), the A-leg will be held, the B-leg will be released and the SCF is informed via 'EventReportBCSM'.

If the A-party disconnects the call is released whether or not a DP was armed.

The following tables provide the rules followed when an armed EDP has to be disarmed because of the occurence of another DP.

Disarming rules of DPs in the O_BCSM:

	Occurred DP							
(Implicitely) disarmed DP	*DP2*	*DP4*	*DP5*	*DP6*	*DP7*	*DP9 Leg A*	*DP9 Leg B*	*DP10*
DP2	*X*					*X*		*X*
DP4		*X*	*X*	*X*	*X*	*X*	*X*	*X*
DP5		*X*	*X*	*X*	*X*	*X*	*X*	*X*
DP6		*X*	*X*	*X*	*X*	*X*	*X*	*X*
DP7		*X*	*X*	*X*	*X*	*X*	*X*	*X*
DP9 Leg A						*X*		*X*
DP9 Leg B		*X*	*X*	*X*		*X*	*X*	*X*
DP10					*X*	*X*		*X*

Disarming rules of DPs in the T_BCSM:

	Occurred DP					
(Implicitely) disarmed DP	*DP13*	*DP14*	*DP15*	*DP17 Leg A*	*DP17 Leg B*	*DP18*
DP13	*X*	*X*	*X*	*X*	*X*	*X*
DP14	*X*	*X*	*X*	*X*	*X*	*X*
DP15	*X*	*X*	*X*	*X*	*X*	*X*
DP17 Leg A				*X*		*X*
DP17 Leg B	*X*	*X*		*X*	*X*	*X*
DP18			*X*	*X*		*X*

9.17.2.2. Error handling

9.17.3. Responding entity (SCF)

9.17.3.1. Normal procedure

SCF Preconditions:
(1) A control relationship exists between the SSF and the SCF.
(2) The SCSM-FSM is in the state "Preparing SSF Instructions", substate "Waiting for Notification or Request".

SCF Postconditions:
(1) The SCSM-FSM stays in the substate "Waiting for Notification or Request" if the message type was notification and there are still EDPs armed or a 'CallInformationReport' requested, or
The SCSM-FSM moves to the state "Idle" if the message type was notification and there are no more EDPs armed, no 'CallInformationReport' *is* requested, or
The SCSM-FSM moves to the state "Preparing SSF Instructions" if the message type was request.
(2) The event is reported to a SLPI, based on the dialogue ID. The SCF will prepare SSF or SRF instructions in accordance with the SLPI.

9.17.3.2. Error handling

9.18. FurnishChargingInformation procedure

9.18.1. General description

This operation is used to request the SSF to generate a call record. The registered call record is intended for off-line charging of the call. A possibility exist for the FCI operation to be invoked on multiple occasions. *For each connection configuration FCI may be used only once.* The charging scenario supported by this operation *is* 2.3 (refer to Annex 'Charging scenarios supported by Core INAP').

Only if a FurnishChargingInformation operation from the SCP is received, an IN-specific ticket is generated at the SSP (exception: subsequent tickets for long duration calls).

9.18.1.1. Parameters

- FCIBillingChargingCharacteristics:
This parameter indicates billing and/or charging characteristics. Its content is network operator specific. Depending on the applied charging scenario the following information elements can be included (refer to Annex 'Charging scenarios supported by Core INAP'):

- charge party (scenario 2.3)
- charge items (scenario 2.3)

9.18.2. Invoking entity (SCF)

9.18.2.1. Normal procedure

SCF Preconditions:
(1) A control relationship exist between the SCF and the SSF.
(2) An SLPI has determined that a 'FurnishChargingInformation' has to be sent by the SCF.

SCF Postconditions:
(1) No FSM state transition.
(2) SLPI execution may continue.

The SCSM FSM is in state "Preparing SSF instruction" . This operation is invoked by the SCF if a SLPI results in the request of creating a call record to the SSF or to include some billing or charging information into the default call record. This causes no SCSM FSM state transition.

The SCF shall limit the number of FCIs during one SSP/SCP- dialog to 6.

9.18.2.2. Error handling

9.18.3. Responding entity (SSF)

9.18.3.1. Normal procedure

SSF Preconditions:
(1) SSF-FSM: State c, "Waiting for Instructions"' or

SSF Postcondition:
(1) No FSM state transition.

On receipt of this operation the SSF performs actions to create the call record according the off line charging scenario which is applicable using the information elements included in the operation:

generates and registers a call record according the information (charge party, charge items),

The SSF records charge related data like for example the call duration, begin time stamp or end time stamp. Additionally the SSF records further data if required.

The FurnishChargingInformation operation can be received before

- *a ConnectToResource or*
- *a Connect operation or*
- *a Continue operation or*
- *an EstablishTemporaryConnection or*
- *a ReleaseCall operation.*

At most one FurnishChargingInformation operation corresponds to one of the above listed operations.

The owner of the call ticket is indicated in the parameter Charged_Party_Identification. The SSP generates an IN-ticket for at most one charged party per connection.

With the receipt of each FCI operation, the SSP writes a time stamp into the call record. This time stamp will be overwritten when ANSWER is received at the SSP and time measurement starts to evaluate the duration of conversation. Time measurement stops at the end of the connection. The evaluated duration of conversation and -according to the SSP settings- either the connection begin time or the connection end time shall be included in the call record. In case the SSP receives no ANSWER the connection shall be

regarded as unsuccessful. If ticket generation is also required for unsuccessful connections, the SSP includes the time stamp that was taken when receiving the FCI operation in the call record. The duration of conversation shall be set to zero.

9.18.3.2. Error handling

By receiving the operation the SSP evaluates the parameter and performs plausibility checks.

The following error situations may occur:

- *More than one FurnishChargingInformation operation are received before a ConnectToResource, a ReleaseCall, an EstablishTemporaryConnection, a continue or a Connect operation. An error report shall be sent to the SCP with error = UnexpectedComponentSequence.*
- *The allowed parameter values are defined in Teil 3, section 1. If there are other values than allowed an error report shall be sent to the SCP with error = UnexpectedDataValue.*

9.19. InitialDP procedure

9.19.1. General description

This operation is sent by the SSF after detection of a TDP-R in the BCSM, to request the SCF for instructions to complete the call.

9.19.1.1. Parameters

- serviceKey:
  This parameter identifies for the SCF unambiguously the requested IN service. It is used to address the correct application/SLP within the SCF (not for SCP addressing).

- calledPartyNumber:
  This parameter contains the dialled number and identifies the called party.

- callingPartyNumber:
  This parameter carries the calling party number to identify the calling party or the origin of the call. The encoding of the parameter is defined in Q.763.
  If the CLI can not be determined completely, the origin (area code) of the A-party is used. If there is no information about the A-party, the parameter is not sent.

- callingPartysCategory:
  Indicates the type of calling party (e.g., operator, pay phone, ordinary subscriber). *Refer to Q.763 for encoding).*

- originalCalledPartyID:
  This parameter carries the dialled digits if the call has met call forwarding on the route to the SSP.

- `locationNumber:`
  `This parameter is used to convey the geographical area address for mobility services. It is used when 'callingPartyNumber' does not contain any information about the geographical location of the calling party (e.g., origin dependent routing when the calling party is a mobile subscriber).`

- bearerCapability:
  This parameter indicates the type of the bearer capability connection to the user:

  - bearerCap:
    This parameter contains the value of the ISUP User Service Information parameter in case the SSF is at transit exchange level. *Refer to Q.763 for encoding.*

The parameter 'bearerCapability' shall only be included in the 'InitialDP' operation in case the ISUP User Service Information parameter is available at the SSP.

If User Service Information and User Service Information Prime are available at the SSF the 'bearerCap' shall contain the value of the User Service Information Prime parameter.

- eventTypeBCSM:
  This parameter indicates the armed BCSM DP event, resulting in the InitialDP operation.

- redirectingPartyID:
  This parameter indicates the directory number the call was redirected from.

- redirectionInformation:
  It contains forwarding related information, such as redirecting counter.

- additionalCallingPartyNumber:
  The calling party number provided by the access signalling system of the calling user.

- highlayerCompatibility:
  This parameter indicates the type of the high layer compatibility, which will be used *together with the bearer capability* to determine the ISDN - teleservice of a connected ISDN terminal. For encoding DSS1 (Q.931) is used.

- *sFEncountered:*
  It indicates that service filtering is encountered and that this call is allowed to pass through, because it is the n-th call.

- *gapInterval:*
  This parameter indicates that the related call has passed ACG. It indicates the actual ACG interval time between calls being allowed to pass ACG. The default value '0' indicates that no calls are affected by call gapping.

- *locationNumberB:*
  This parameter indicates the location number B of the called mobile subscriber. The location number B will be represented by the MSRN as specified in MAP [19] .

- *userDialogInfo:*
  This parameter contains the USSD information from the Mobile Station, coded as component of the facility-Information-Element as specified in GSM 04.80 [20]. In case the USSD information exceeds the maximum length of userDialogInfo only the first Octets (until the maximum length of userDialogInfo) will be used.

- *iMSI:*
  This parameter contains the IMSI information from the Mobile Station. It is only used in mobile networks.

- *natCallingPartysCategory:*
  This parameter indicates the type of calling party. It may contain Telekom specific types in addition to the parameter CallingPartysCategory. See [7] regarding the mapping on ISUP.

- *INContainer: This parameter may be used by a service logic to provide a message in forward direction to another service logic which is invoked later for the same call. This message may be sent using the Connect operation. It will be transported transparently through the basic network an will be presented to the next service logic within the operation InitialDP.*

9.19.2. Invoking entity (SSF)

9.19.2.1. Normal procedure

SSF Preconditions:
17.03.9818.01.20009

(1) Call origination attempt has been initiated.
(2) An event has been detected at a DP.
(3) Call gapping and SS7 overload are not in effect for the call, and the call is not to be filtered.

SSF Postcondition:
(1) A control relationship has been established and the SSF waits for instructions from the SCF.

Following a trigger detection related to an armed TDP-R in the BCSM caused by a call origination attempt, the SSF checks if call gapping, SS7 overload or service filtering are not in effect for the related call segment.

If these conditions are met, then the 'InitialDP' operation is invoked by the SSF. The address of the SCF the 'InitialDP' operation has to be sent to is determined on the base of trigger related data. The SSF provide as many parameters as available.

In some service-specific cases, some parameters shall be available (such as "callingPartyNumber" or "callingPartysCategory"). This shall be handled appropriately by the SSF in its trigger table (to know that such parameter are necessary for some triggering conditions).

If call gapping or service filtering are active for the related call, but the call is allowed to be indicated to the SCF, the SSF includes the appropriate parameters 'SFEncountered' and 'GapInterval' in the InitialDP operation.

A control relationship is established to the SCF. The SSF application timer T_{SSF} is set when the SSF sends 'InitialDP' for requesting instructions from the SCF. It is used to prevent from excessive call suspension time.

9.19.2.2. Error handling

If the destination SCF is not accessible, the SSF FSM will usually close the dialogue, locally, and instruct the CCF to play a tone to the A-party. However depending on trigger-specific information, the SSF also may instruct the CCF to complete the call if possible.

On expiration of TSSF before receiving any operation, the SSF will usually abort the interaction with the SCF and instruct the CCF to play a tone to the A-Party. However depending on trigger-specific information, the SSF also may instruct the CCF to complete the call if possible.

If the calling party abandons after the sending of 'InitialDP', then the SSF aborts the control relationship after the first answer message from the SCF has been received.

9.19.3. Responding entity (SCF)

9.19.3.1. Normal procedure

SCF Precondition:
None.

SCF Postcondition:
(1) An SLPI has been invoked.

On receipt of 'InitialDP' operation the SCSM moves from "Idle" to the state "Preparing SSF Instructions", a control relationship to the related SSF is created. A Service Logic Program Instance (SLPI) is invoked for processing the 'InitialDP' operation based on the 'serviceKey' parameter. By means of this control relationship, the SCF may influence the Basic Call Processing in accordance with the service logic invoked.

The actions to be performed in the SLPI depend on the parameters conveyed via this operation and the SLPI, i.e. the requested IN service, itself.

9.19.3.2. Error handling

If the 'InitialDP' operation is rejected then the SCSM remains in "Idle". The maintenance function is informed and no SLPI is invoked.

The following error cases are indicated to the SSF:

- MissingCustomerRecord:
  The SCF has not found the appropriate service logic corresponding to the provided service key or the appropriate service subscriber related service logic.

9.21. PlayAnnouncement procedure

9.21.1. General description

This operation is used for inband interaction with an analogue user or an ISDN user.

9.21.1.1. Parameters

- informationToSend:
  This parameter indicates an announcement *or* a tone to be sent to the end user by the SRF.
  - inbandInfo:
    This parameter specifies the inband information to be sent.
    - messageID:

      This parameter indicates the message(s) to be sent, this can be one of the following:
      - elementaryMessageID:
        This parameter indicates a single announcement.
      - elementaryMessageIDs:
        This parameter specifies a sequence of announcements.
      - variableMessages:
        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. In case neither the ‘numberOfRepetitions’ nor ‘duration’ parameters are included, the SSP uses a default value 'numberOfRepetitions' = 1.
    - 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.
  - tone:
    This parameter specifies a tone to be sent to the end-user.
    - toneID:
      This parameter indicates the tone to be sent.

- duration:
  This parameter indicates the time duration in seconds of the tone to be sent.

- *userDialogInfo:*
  This parameter contains the USSD information, that shall be delivered to a Mobile Station, coded as component of the facility-Information-Element as specified in GSM 04.80 [20].
  The parameter can be used if the A-party is connect at the GSM A-interface directly to the SSF. If received in other cases (e.g. DSS1 interface of GSM MSC or ISDN LE) the parameter is ignored.

- disconnectFromIPForbidden:
  This parameter indicates whether or not the SRF should be disconnected from the user when all information has been sent.

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

9.21.2. Invoking entity (SCF)

9.21.2.1. 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.
(3) The SCSM-FSM is in the state "User interaction", substate "Waiting for response from the SRF".

SCF Postconditions:
(1) If 'RequestAnnouncementComplete' was set TRUE, the SCSM will stay in substate "Waiting for Response from the SRF" and wait for the 'SpecializedResourceReport'.
(2) 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".

This operation may be sent with the first SRF-request for user interaction (together with a ConnectToResource) or as a further SRF-request.

The flags 'disconnectFromIPForbidden' or 'requestAnnouncementComplete' may be set or not, controlled by the SLPI.

When SRF-initiated disconnect has been allowed, the user interaction phase is ended in the SCSM after reception of the SpecializedResourceReport and the state is changed to 'Preparing_SSF_Instructions'. When SRF-initiated disconnect has been not allowed, the SLPI indicates the end of the user interaction phase. The connection to the SRF is released by sending a DisconnectForwardConnection and the state is changed to 'Preparing_SSF_Instructions'.

9.21.2.2. Error handling

9.21.3. Responding entity (SRF)

9.21.3.1. 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 state *"User Interaction"*.

(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.

The PlayAnnouncement operation is accepted in the state 'Connected'. The SRF-FSM changes to the state 'User Interaction' and the Timer TSRF is stopped. In the state 'User Interaction' this operation is accepted as a subsequent UI operation. No state transition is made. The Timer TSRF is stopped.

In cases where 'requestAnnouncementComplete' was set to TRUE and 'disconnectFromIPForbidden' was set to FALSE the sending of SpecializedResourceReport has to be done first.

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

- when the complete announcement is sent, 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).

9.21.3.2. Error handling

If there are any internal error situation to provide the requested operation (e.g. the operation can not be relayed from the SSF to the SRF) the Reject: 'ResourceLimitation' is indicated to the SCF.

If the timer TSRF expires, the bearer connection is disconnected.

If any error is generated and sent to the SCF, no bearer disconnection is initiated by the SRF, independent of whether a SRF-initiated disconnect has been allowed by the SCF or not.

MissingParameter:

The operation contains no parameters.
The check of the AnnouncementId detects that variable parts are needed and these parts are missing.

UnexpectedDataValue:

Parameter MessageId has an unexpected value.

UnavailableResource:

The requested tone or announcement is not available on the IP (indicated by the SRF).

SystemFailure:

After the successful seizure of the resource the SRF detects an error.

9.22. PromptAndCollectUserInformation procedure

9.22.1. General description

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

9.22.1.1. Parameters

- collectedInfo

- collectedDigits

  - 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.

    Alcatel: If endOfReplyDigit, cancelDigit or startDigit parameters are defined, the indicated digits are not considered as" valid digits" (i.e. do not count for the minimumNbOfDigits condition).

  Siemens: If cancelDigit or startDigit parameters are defined, the indicated digits are not considered as" valid digits" (i.e. do not count for the minimumNbOfDigits condition). If endOfReplyDigit parameter is defined, the indicated digits are considered as" valid digits"

  - maximumNbOfDigits
    This parameter should always be present and specifies the maximum number of valid digits to be collected. The following applies: 'maximumNbOfDigits' >= 'minimumNbOfDigits'.

    Alcatel: If endOfReplyDigit, cancelDigit or startDigit parameters are defined, the indicated digits are not considered as" valid digits" (i.e. do not count for the maximumNbOfDigits condition).
    Siemens: If cancelDigit or startDigit parameters are defined, the indicated digits are not considered as" valid digits" (i.e. do not count for the maximumNbOfDigits condition).If endOfReplyDigit parameter is defined, the indicated digits are considered as" valid digits".

  - 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. *It shall not use digits 0-9.*

    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 PromptAndCollectUserInformation 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. *It shall not use digits 0-9.* 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. *It shall not use digits 0-9.* If this parameter is not present, all received digits are considered to be valid.

  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 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 stdErrorAndInfo.

- 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 *not supported for this operation*.

- voiceBack
  This parameter is *not supported for this operation*. *If received, it will be ignored.*

- disconnectFromIPForbidden:
  This parameter indicates whether the SRF should initiate disconnection to the SSF/CCF after the interaction has been completed. If the parameter is not present or set to TRUE, the SRF shall not initiate disconnection.

- informationToSend:
  This parameter indicates an announcement *or* a tone to be sent to the end user by the SRF.

  - inbandInfo:
    This parameter specifies the inband information to be sent.

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

      - elementaryMessageID:
        This parameter indicates a single announcement.

      - elementaryMessageIDs:
        This parameter specifies a sequence of announcements.

      - variableMessage:
        This parameter 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. In case neither the ‘numberOfRepetitions’ nor ‘duration’ parameters are included, the SSP uses a default value 'numberOfRepetitions' = 1.

    - 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.

  - tone:
    This parameter specifies a tone to be sent to the end-user.

    - toneID:
      This parameter indicates the tone to be sent.

    - duration:
      This parameter indicates the time duration in seconds of the tone to be sent.

  - *userDialogInfo:*
    This parameter is not allowed to be sent for this operation. If received, it will be ignored.

- digitsResponse:
  This parameter contains the information collected from the end-user.

9.22.2. Invoking entity (SCF)

9.22.2.1. 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) The SCSM FSM is in state "User Interaction", substate "Waiting for Response from the SRF".

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.

This operation may be sent with the first SRF-request for user interaction (together with a ConnectToResource) or as a further SRF-request.

The flag 'disconnectFromIPForbidden' may be set or not, controlled by the SLPI.

If no errors occur the PromptAndCollectUserInformation operation causes a Return Result from the SRF.

If SRF-initated disconnect has been allowed, the SCSM changes the state to 'Preparing_SSF_Instructions' on receipt of the Return Result containing the collected digits.

If SRF-initated disconnect has been not allowed, no state transition is made. The connection to the SRF is released on indication of the SLPI by sending a DisconnectForwardConnection and the state is changed to 'Preparing_SSF_Instructions'.

The 'disconnectFromIPForbidden' parameter may only be set to FALSE if the 'PromptAndCollectUserInformation' operation is the last operation sent to the SRF.

9.22.2.2. Error handling

9.22.3. Responding entity (SRF)

9.22.3.1. 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) 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 SRSM FSM moves to the state 'Idle'.
(4) Otherwise the SRSM FSM moves to the state "User Interaction", or
remains in the state *"User Interaction"*.

The PromptAndCollectUserInformation operation is accepted in the state 'Connected'. The SRF-FSM changes to the state 'User Interaction' and the Timer TSRF is stopped.

In the state 'User Interaction' this operation is accepted as a subsequent UI operation. No state transition is made. The Timer TSRF is stopped.

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

- when the complete announcement is sent, 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 parameter 'errorTreatment' specifies how the SRF shall treat the error 'ImproperCallerResponse'. The default value 'stdErrorAndInfo' means that the error shall be reported to SCF as specified in section 8. 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.

9.22.3.2. Error handling

If there are any internal error situation to provide the requested operation (e.g. the operation can not be relayed from the SSF to the SRF) the Reject: 'ResourceLimitation' is indicated to the SCF.

If the timer TSRF expires, the bearer connection is disconnected.

If any error is generated and sent to the SCF, no bearer disconnection is initiated by the SRF, independent of whether a SRF-initiated disconnect has been allowed by the SCF or not.

MissingParameter:

> *The operation contains no parameters. The check of the AnnouncementId detects that variable parts are needed and these parts are missing.*

UnexpectedDataValue:

> *The Parameter messageID, minimumNbOfDigits, maximumNbOfDigits, or errorTreatment have an unexpected value.*
> *The value of minimumNbOfDigits is greater than the value of maximumNbOfDigits.*
> *The value of the parameters endOf ReplyDigit, and cancelDigit are equal.*

UnavailableResource:

> *The requested tone, announcement, or resource for digit collection is not available on the IP (indicated by the SRF).*

SystemFailure:

> *After the successful seizure of the resource the SRF detects an error.*

9.23. ReleaseCall procedure

9.23.1. General description

This operation is used to tear down by the SCF an existing call at any phase of the call for all parties involved in the call.

9.23.1.1. Parameters

- Cause
  A number giving an indication to the SSF about the reason of releasing this specific call. This may be used by SSF for generating specific tones to the different parties in the call or to fill in the 'cause' in the release message.

9.23.2. Invoking entity (SCF)

9.23.2.1. Normal procedure

SCF Precondition:
(1) State 2.1, "Preparing SSF instructions".

SCF Postconditions:
(1) State 1, "Idle", if no 'CallInformationReport' has to be received from the SSF. All resources related to the call are released by the SCF, or
State 2.3, "Waiting for Notification or Request" if a 'CallInformationReport' still has to be received from the SSF.

9.23.2.2. Error handling

9.23.3. Responding entity (SSF)

9.23.3.1. Normal procedure

SSF Preconditions:
(1) State C," Waiting for Instructions", or
State F, "Monitoring"

SSF Postcondition:
(1) "Idle", state a, after sending any outstanding 'CallInformationReport'. Possible armed EDPs are ignored. All connections and resources related to the call are released.

The ReleaseCall operation is accepted in the SSF FSM states 'Waiting for Instructions' and 'Monitoring'.

In any case, the SSF instructs the CCF to tear down the related call. The parameter 'Cause' provided by ReleaseCall is sent to the call parties.

NOTE: If ReleaseCall is received in the state 'Wait for End of User Interaction', then the operation is saved, i.e. the operation will be executed after a transition to the 'Waiting for Instructions' state.

9.23.3.2. Error handling

9.25. RequestReportBCSMEvent procedure

9.25.1. General description

This operation is used to request the SSF to monitor for a call-related event (e.g., BCSM events such as busy or no answer), then send a notification back to the SCF when the event is detected.

The requested events are armed dynamically as EDPs. Encountering of such an EDP the SSF will send a report back to the SCF.

9.25.1.1. Parameters

- bcsmEvents:
  This parameter specifies the event or events of which a report is requested.

  - eventTypeBCSM:
    This parameter specifies the type of event of which a report is requested.

  - monitorMode:
    This parameter indicates how the event should be reported. When the 'monitorMode' is 'interrupted', the event shall be reported as a request, if the 'monitorMode' is 'notifyAndContinue', the event shall be reported as a notification.

  - legID:
    This parameter indicates the party in the call for which the event shall be reported. SCF will use the option 'sendingSideID' only.

    - sendingSideID:
      The following values for 'legID' are assumed:
      legID = 1 indicates the party that was present at the moment of the 'InitialDP' *(A-party)*.
      legID = 2 indicates the party that was created with a 'Connect' operation *(B-party)*.
      If not included, the following defaults are assumed:
      legID = 1 for the event CollectInfo, O- and T- Abandon
      legID = 2 for the events RouteSelectFailure, O-CalledPartyBusy, O-NoAnswer, O-Answer, T-CalledPartyBusy, T-NoAnswer, T-Answer.
      The 'legID' parameter shall always be included for the event O-Disconnect and T-Disconnect.
      Only legID = 2 is allowed in case O- or T-Disconnect is armed as an EDP-R.

  - dPSpecificCriteria:
    This parameter indicates information specific to the EDP to be armed. *Two approaches are supported:*

    - numberOfDigits:
      This parameter indicates the number of digits to be collected by the SSF for the CollectedInfo event. If the indicated number of digits is collected, SSF reports the event to the SCF.

    - applicationTimer:
      This parameter indicates the application timer for the NoAnswer event. If the user does not answer the call within the allotted time, the SSF reports the event to the SCF. This timer shall be shorter than the network no-answer timer.
      If this timer expires and the NoAnswer event was armed as EDP-R, the SSF automatically tears down the forward connection to the B-party to avoid synchronization problems, then informs the SCF.

9.25.2. Invoking entity (SCF)

9.25.2.1. Normal procedure

SCF Preconditions:
(1) A control relationship exists between the SCF and the SSF.
(2) The SLPI has decided that a request for an event report BCSM is needed.
(3) The SCSM FSM is in the appropriate state to send 'RequestReportBCSMEvent'.

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

If the SCF uses the RequestReportBCSMEvent operation to arm Event Detection Points (EDPs) for a terminating IN call which are related to a new leg which is to be setup by the SSP (-> SCP based call forwarding: SLPI requests the sending of a Connect operation after receipt of an InitialDP operation indicating the TDP-12), these EDPs shall be armed as EDPs of the O_BCSM ('originating EDPs'). This, because the new 'forwarded leg' shall be monitored by a new O_BCSM (refer to the detailled description of the Connect procedure). This means the SCF can request the following EDPs: DP4, DP5, DP6, DP7 as well as DP9 related to the B-party ('legID' = 2). The SCF must not arm EDPs related to the A-party ('legID' = 1), this means neither EDP 9 related to the A-party nor EDP17 related to the A-party must be armed.

9.25.2.2. Error handling

9.25.3. Responding entity (SSF)

9.25.3.1. Normal procedure

SSF Precondition:
(1) The SSF FSM is in the state "Waiting for Instructions" .

SSF Postconditions:
(1) The requested EDPs have been armed as indicated.
(2) The SSF FSM remains in the same state.

9.25.3.2. Error handling

MissingParameter:

- *DP-2 shall be armed but ´numberOfDigits´is not contained or*
- *DP-6 respectively DP-14 shall be armed but ´applicationTimer´is not contained or*
- *DP-9 respectively DP-17 shall be armed but ´legID´is not contained.*
- *This error may also be sent, if the operation contains no parameter at all.*

UnexpectedDataValue:

- a*rming DP-7 respectively DP-15 as EDP-R or*
- *'legID' = 1 (A-Party) in case of arming DP9 as EDP-R respectively arming DP17 as EDP-R/-N or*
- *error in parameter 'legID': OCTET STRING typed parameters 'sendingSideID' or 'receivingSideID' have an unexpected value.*

UnexpectedParameter:

- *DP-2 will not be armed but 'numberOfDigits' is contained or*
- *DP-6 respectively DP-14 will not be armed but 'applicationTimer' is contained.*

9.26 ResetTimer procedure

9.26.1 General description

This class 2 operation is used by the SCF to refresh the T_{SSF} application timer, in order to avoid the T_{SSF} time-out at the SSF.

9.26.1.1 Parameters

- timerValue:
  This parameter specified the value to which the T_{SSF} is to be set.

- timerID:
  This parameter has a default value identifying the T_{SSF} timer.

9.26.2 Invoking entity (SCF)

9.26.2.1 Normal procedure

SCF preconditions:
(1) A control relationship exists between the SCF and the SSF.
(2) An SLPI has determined by the $T_{SCF-SSF}$ guard timer expiration, that the ResetTimer operation has to be sent in order to avoid T_{SSF} time-out at the SSF.

SCF postcondition:
(1) The SLPI reset the $T_{SCF-SSF}$ guard timer.

9.26.2.2 Error handling

Generic error handling for the operation related errors is described in Clause 8 and the TCAP services which are used for reporting operation errors are described in Clause 10.

9.26.3 Responding entity (SSF)

9.26.3.1 Normal procedure

SSF preconditions:
(1) Call origination attempt has been initiated.
(2) Basic call processing has been suspended at a DP.
(3) The SSF FSM is in the "Waiting for Instruction" state or in the "Waiting for End of User Interaction" state or in the "Waiting for End of Temporary Connection" state.

SSF postconditions:
(1) The T_{SSF} timer has been reset.
(2) The SSF FSM remains in the same state.

9.26.3.2 Error handling

9.27. SendChargingInformation procedure

9.27.1. General description

This operation is used to instruct the SSF on the charging information to be sent by the SSF. The sending of charging information *is done by* signalling . In the LE, either charge meter can be updated or a standard call record created. A possibility exists for the SCI operation to be invoked on multiple occasions.

9.27.1.1. Parameters

sCIBillingChargingCharacteristics:
This parameter indicates billing and/or charging characteristics. Its content is network operator specific. Depending on the applied charging scenario the following information elements can be included (refer to Annex 'Charging scenarios supported by Core INAP' *and Teil 3*):

charg*ing* level (scenario 3.2),

- *adviceOfCharge (for the details with respect to coding and procedural aspects refer to Teil 3).*

The sub-parameter 'adviceOfCharge' is for mobile applications only and therefore to be ignored when received by the fixed network.

9.27.2. Invoking entity (SCF)

9.27.2.1. Normal procedure

SCF Preconditions:
(1) A control relationship exist between the SCF and the SSF.
(2) An SLPI has determined that a 'SendChargingInformation' has to be sent by the SCF.

SCF Postconditions:
(1) No FSM state transition,
(2) SLPI execution may continue.

The SCSM FSM is in state "Preparing SSF Instruction" - *substate "Preparing SSF Instructions" or "Waiting for Notification or Request" - or is in state "User Interaction" - substate "Waiting for Response from the SRF". The SendChargingInformation procedure shall be invoked by the SCF in accordance with the demands of the SLPI for relevant charging information. If appropriate this information shall be sent back down the call path.*

The SCP indicates either the appropriate chargeband number to be used by the parameter SCPchargenumber or requires that the charging level is still to be determined on the base of the given destination (parameter 'zoningRequired'). The latter case is only applicable for connections to a B-subscriber.

In case, new charging instructions are necessary before setup of a new connection configuration (setup to a B-party, to an IP reached via SSF relay or via an assisting SSF), the corresponding SCI operation is to be sent before the CON, CTR, CUE or ETC operation.

In case, the SLPI requires a tariff change during a connection to a B-party, the SCF shall monitor the answer status of the B-Party by sending an RequestReportBCSMEvent before the corresponding Connect operation to arm the DP7 'OAnswer' in 'notifyAndContinue' mode. Only after receipt of the related EventReportBCSM operation from the SSF a tariff change may be initiated by the SLPI. The SLPI determines at which point in time the corresponding SCI operation is to be sent, i.e. an application timer shall be started after receipt of the answer indication and the expiration of this timer triggers the sending of the SCI.

In case, a tariff change is required during a connection to an IP (that means during an active UI phase), the SCF shall monitor the result of the first UI operation by waiting for the end of an announcement indicated by a SpecializedResourceReport operation or for the result on a requested prompt. Only after receipt of this first result a corresponding SCI operation may be sent.

In case, a tariff change is required during a connection to a centralised IP using the assist procedure, the SCF shall monitor the result of the first UI operation by waiting for the end of an announcement indicated by a SpecialisedResourceReport operation or for the result on a requested prompt. After receipt of this first result a corresponding SCI operation shall be sent to the initiating SSP.

The following is valid only for the Telekom ISDN/PSTN:
In any case the SCF intends to send an SCI related to a certain connection (either before setup or during a stable connection to a B-party or to an integrated or attached IP), an FCI must be sent before.

NOTE: The FCI is needed for the inclusion of the charge band number into the ticket of the Telekom SSP.

9.27.2.2. Error handling

9.27.3. Responding entity (SSF)

9.27.3.1. Normal procedure

SSF Preconditions:
(1) SSF-FSM: State c: "Waiting for Instructions", or
SSF-FSM State d: "Waiting for End of User Interaction", or
SSF-FSM State e: "Waiting for End of Temporary Connection", or
SSF-FSM State f: "Monitoring".

SSF Postcondition:
(1) No FSM state transition

On receipt of this operation the SSF performs actions to send the charging information. The sending of charging information *is done by* signalling. In the LE, either charge meter can be updated or a standard call record created. The interworking between SSF and PSTN is *defined in [7].*

For instance, by sending an operation 'SendChargingInformation' the SCF instructs the SSF to initiate the PSTN/ISDN charging functions according to the given information about the charging level to use.

The charging level can be determined either by one of the following functions

(a) the SCF, or
(b) the charging function in a succeeding exchange.

In case of the SCF has determined the charging level the 'SendChargingInformation' operation contains the charging level *(determining the chargeBand number)* to be applied.

If the charging level was determined by the IN (SCF) the SSF provides the charging level to be applied to the charging functions (case a).

The SCF may deliver additional AoC Information. At the moment this parameter is used only in conjunction with a GSM A-interface. When received for an other access type this information shall be ignored by the SSF. The information shall be used as input for the AoC procedure defined for the access.

In case *b* the charging level is determined in a succeeding exchange. *The* SSF detects during trying to determine the charging level based on SCF provided parameters that the charging level shall be determined in a succeeding exchange. Based on already existing PSTN/ISDN capabilities the SSF provides the PSTN/ISDN charging functions with the necessary information and backward charge messages shall be transferred down the call path when allowed by the SCF (generated by a succeeding exchange for example an international gateway).

If the SCP provides a 'ScpChargeNumber' , any charging messages received from the B-side are not passed to the calling line.

In the scenario described above charging/billing is performed by means of existing mechanisms of the PSTN/ISDN initiated and controlled by the IN.

That means the determination of the charging method - on-line or off-line - shall be done in the basic network, just like the charge generation and the charge registration.

A SendChargingInformation operation may be sent before a

- *ConnectToResource operation (only in case CTR is not sent to an assisting SSP) or a*
- *Connect operation or a*

- *Continue operation or an*
- *EstablishTemporaryConnection operation.*

Just one SendChargingInformation operation is allowed before the operations listed above. Otherwise the error UnexpectedComponentSequence shall be sent to the SCP (see error case 1).

Additionally, a SendChargingInformation operation is allowed during a stable connection

- *to a B-party or*
- *to an integrated or attached IP or*
- *to a centralized IP using the assist procedure.*

The SSP accepts this charging request only in case, an Answer message has already been sent to the A-party, otherwise the call is to be released, refer to the error handling described below.

During already established connections (to a B-party or to an integrated or attached IP) the SSP accepts only one SCI containing an 'scpChargeNumber'.If the SCI is used to send only the parameter 'adviceOfCharge', more than one SCI operation, received during the established connection, will be possible.

NOTE: In case, an SCI has been received during a connection configuration, the SSP provides a subsequent ticket.After receipt of an SCI the SSP performs the appropriate interworking with the charging functions (see [7]) and assures, that any charging instructions from the SCP for the calling line are appropriately signalled to the originating exchange and registered in the call record.

9.27.3.2. Error handling

The following error situations may occur:

- *More than one SendChargingInformation operation is received before a ConnectToResource, a Connect, a Continue or an EstablishTemporaryConnection operation. An error report shall be sent to the SCP with error = UnexpectedComponentSequence.*

- *An SCI has been received during a stable connection to a B-party or to an integrated or attached IP with the parameter 'zoningRequired':*
  The error 'ParameterOutOfRange' shall be sent to the SCF in this case.

- *In case, the SSP has received more than one SCI operation containing the parameter 'scpChargeNumber' and thus causing a tariff change during a stable connection to a B-party or to an integrated or attached IP the IN call has to be released. The SCF gets an U-Abort message with the UserAbort-Information 'interfaceEventsSSF, chargingError'.*

- *In case, the SCP has sent an SCI operation during a connection configuration and the SSP has not yet sent the Answer message to the A-party the IN call has to be released. The SCF gets an U-Abort message with the UserAbort-Information 'interfaceEventsSSF, chargingError'.*

- *The coding if the parameter 'sCIBillingChargingCharacteristics' is defined in Teil 3, section 2 using the ASN.1 notation. If a received chargeBand number has a value different from 0 or 2..255 an error report shall be sent to the SCP with error = ParameterOutOfRange.*

- *The following is valid only for the Telekom ISDN/PSTN:*
  If the SSP receives SCI and no FCI has initiated a callrecord generation before, an error report shall be sent to the SCP with error = UnexpectedComponentSequence. This is also valid for the case, the SCI operation has been received during a stable connection.

- *In case the SSP receives a SCI and an acknowledgement for a chargeband message which has been sent due to an earlier SCI is still outstanding, the SSP aborts the TCAP-dialogue with the SCP and releases the call.*

- *If the assist procedure is active and the assisting SSP receives an SCI operation this operation is rejected by a REJECT message with error code 'unrecognizedOperation'.*

9.28. ServiceFilteringResponse procedure

9.28.1. General description

This operation is used to report the values of counters specified in a previous sent 'ActivateServiceFiltering' operation to the SCF.

9.28.1.1. Parameters

- countersValue:
  The parameter contains the count of calls filtered during the filtering period. It is a list of counter identifications and the related values.

- filteringCriteria:
  This parameter is used to address the concerned service logic at the SCF.

9.28.2. Invoking entity (SSF)

9.28.2.1. Normal procedure

SSF Preconditions:
(1) Service filtering is running and the interval time is expired and a call is received, or
(2) Service filtering is running and the threshold value is reached, or
(3) Service filtering has been finished (duration time expired or stop time met), or
(4) The operation 'ActivateServiceFiltering' is received and encounters an active service filtering entity.

SSF Postcondition:
(1) Service filtering proceeds or is ended depending on the duration time.

The SSF sends the 'ServiceFilteringResponse' operation to the SCF. The 'filteringCriteria' parameter is provided to enable the addressing of the concerned service logic at the SCF.

Before 'ServiceFilteringResponse' is sent, it is checked whether call gapping criteria are met. If so, the 'ServiceFilteringResponse' is not sent and the counting continues without resetting the counters. The last 'ServiceFilteringResponse' (stop time is met or duration time expired) is sent without checking any call gap criteria.

After sending 'ServiceFilteringResponse' the service filtering counters are reset.

If service filtering proceeds after sending 'ServiceFilteringResponse' (e.g. interval time expired) the SSME-FSM remains in the state "Non-Call Associated Treatment".

If service filtering is stopped after sending 'ServiceFilteringResponse' (duration time expired or stop time is met) then the SSME-FSM moves to the "Idle Management" state. All concerned resources are released, i.e. the SSME-FSM is removed as well.

9.28.2.2. Error handling

9.28.3. Responding entity (SCF)

9.28.3.1. Normal procedure

SCF Preconditions:
(1) Service filtering is running.
(2) The SCME is in the state "Waiting for Service Filtering Response".

SCF Postcondition:
(1) The SCME forwards the received counter values to the SLPI.

The operation is handled by the Service Filtering FSM part of the SCF Management Entity (SCME). The SCME passes the received counter values to the SLPI where they are added to previously received counter values.

The 'filteringCriteria' parameter as provided in 'ServiceFilteringResponse' is used to address the SCME and the concerned service logic instance.

The Service Filtering FSM of the SCME remains in the state "Waiting For SSF Service Filtering Response" until the internal service filtering duration time in the SLPI expires. Then the SLPI informs the SCME about timer expiration. Now the SCME moves to the state "Service Filtering Idle".

9.28.3.2. Error handling

If the SCME is in the state "Service Filtering Idle" an incoming 'ServiceFilteringResponse' operation is ignored.

9.29. SpecializedResourceReport procedure

9.29.1. General description

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

9.29.1.1. Parameters

- None

9.29.2. Invoking entity (SRF)

9.29.2.1. 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 'RequestAnnouncementCompelete' 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".

9.29.2.2. Error handling

9.29.3. Responding entity (SCF)

9.29.3.1. Normal procedure

SCF Precondition:
(1) The SCSM FSM is in the state "User Interaction", substate "Waiting for response from the SRF".

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".

9.29.3.2. Error handling

10 Services assumed from TCAP

10.1 Normal procedures

This section describes the procedures and TCAP primitives that shall be used for transmitting messages between SSF, SCF, *and* SRF under normal operation.

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

- a dialogue shall be established: the TC-user issues a TC-BEGIN request primitive.
- a dialogue shall be maintained: the TC-user issues a TC-CONTINUE request primitive.
- a dialogue shall no longer be maintained: the TC-user issues a TC-END request primitive with either basic end or with pre-arranged end depending on the following conditions:
  - Basic End
    - Operations, leading to a termination of the control relationship, can be transmitted by the SCF with a TC-END request primitive (basic) in case the SCF is not interested in the reception of any ERROR or REJECT components for these sent operations.
    - In case the SCF entity has received a*n* operation, leading to the termination of the control relationship, a TC-END request primitive (basic) with zero components can be sent from the SCF.
  - Pre-arranged End
    In case, an entity is interested in possible ERROR or REJECT messages on response to sent operations leading to a termination of the control relationship, the dialogue is ended with a TC-END request primitive (pre-arranged end) after the last associated operation timer expires. The receiving entity can end the dialogue with a TC-END request primitive (pre-arranged end) after successful processing of these operations (i.e. the control relationship is terminated).
- a dialogue shall not be established: the sending TC-user issues a TC-BEGIN request primitive and ends the dialogue locally after operation timeout by means of a prearranged end. Upon reception of the TC-BEGIN indication primitive the receiving TC-user shall end the dialogue locally.

10.1.1 SSF-to-SCF messages

10.1.1.1 SSF-FSM related messages

A dialogue shall be established when the SSF-FSM moves from the state **Trigger Processing** to the state **Waiting for Instructions**. The relevant INAP operation, which can only be the InitialDP operation, shall be transmitted in the same message.

For all other operations sent from the SSF-FSM, the dialogue shall be maintained.

The dialogue shall no longer be maintained when the prearranged end condition is met in the SSF. When the SSF-FSM makes a regular state transition to the state **Idle**, the dialogue is locally ended by means of a TC-END request primitive with prearranged end.

When the SSF has sent the last EventReportBCSM or CallInformationReport the dialogue may be ended from the SCF by a TC-END request primitive with basic end.

10.1.1.2 Assisting SSF FSM related messages

A dialogue shall be established when the Assisting SSF FSM moves from the state **Idle** to the state **Waiting for Instructions**. The AssistRequestInstructions operation shall be transmitted with a TC-BEGIN request primitive.

For all other operations sent from the Assisting SSF FSM, the dialogue shall be maintained.

The dialogue shall no longer be maintained when the prearranged end condition is met in the SSF. When the SSF FSM makes a state transition to the state **Idle**, the dialogue is locally ended by means of a TC-END request primitive with prearranged end.

10.1.1.3 SSME-FSM related messages

The following procedures shall be followed:

- The dialogue shall be maintained when the ActivityTest Return Result is sent
- No dialogue shall be established when the ServiceFilteringResponse operation is sent. The operation is sent with a TC-BEGIN request primitive and the dialogue is ended by means of a TC-END request primitive with prearranged end.
- A dialogue shall no longer be maintained when the Return Result of the ActivateServiceFiltering operation is sent. The dialogue is ended by means of a TC-END request primitive with basic end, the Return Result is transmitted with the same request.
- The dialogue is locally terminated upon reception of a TC-BEGIN indication primitive with a CallGap operation.

10.1.2 SCF-to-SSF messages

10.1.2.1 SCSM-FSM related messages

For subsequent operations sent from the SCSM-FSM, the dialogue shall be maintained, i.e. all other operations are sent after a dialogue was established from the SSF (the SCF has previously received a TC-BEGIN indication primitive with `either an InitialDP or an AssistRequestInstructions operation).`

The dialogue shall no longer be maintained when the prearranged end condition is met in the SCF. 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 control relationship, by means of a TC-END request primitive (basic end) is possible.

10.1.2.2 SCME-FSM related messages

The operations sent from the SCME-FSM shall be issued according to the following procedures:

- The dialogue shall be maintained when the ActivityTest operation is sent
- A dialogue shall not be established when a CallGap operation is sent without using a SCSM associated dialogue. The operation is sent using a TC-BEGIN request primitive and the dialogue is terminated with a prearranged end.
- For sending one or more CallGap operations, the SCME FSM may use an existing SCSM FSM associated dialogue which was initiated by a SSF-FSM (i.e. established for the transmission of the InitialDP operation). The dialogue shall be maintained and the CallGap operation(s) shall be sent *in a separare TC-CONTINUE as* the first response of the SCSM FSM to the InitialDP operation *or EventReportBCSM operation (in case it is requested via a RequestReportBCSM operation and CollectInformation before). The normal response of the SCSM-FSM is then send in a further TC-CONTINUE*.
- A dialogue shall be established when an ActivateServiceFiltering operation is sent. The operation shall be transmitted with a TC-BEGIN request primitive.
- The dialogue is locally terminated upon reception of a ServiceFilteringResponse operation using a TC-END request primitive with prearranged end.

10.1.3 SCF-to/from-SRF messages

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.

10.2 Abnormal procedures

This section describes the procedures and TCAP primitives that shall be used for reporting abnormal situations between SSF, SCF, *and* SRF. The error cases are defined in § 3.2.

The following primitives shall be used to report abnormal situations:

- operation errors, as defined in the *Telekom* INAP, are reported with TC-U-ERROR request primitive.
- rejection of a TCAP component by the TC-user shall be reported with TC-U-REJECT request primitive.
- a dialogue shall be aborted by the TC-user with a TC-U-ABORT request primitive

For abnormal situations detected by TCAP the same rules shall apply for transmission of TC-R-REJECT indication as for transmission of TC-U-REJECT request.

In error situations prearranged end shall not be used. In case any application entity encounters an error situation the peer entity shall be explicitly notified of the error, if possible. If from any entity's point of view the error encountered requires the relationship to be ended, it shall close the dialogue via a TC-END request primitive with basic end or via a TC-U-ABORT request primitive , depending on whether any pending ERROR or REJECT component is to be sent or not.

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

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

10.2.1 SCF-to-SSF/SRF messages

Considering that both SSF and SRF do not have the logic to recover from error cases detected on the SCF-SSF/SRF interface, the following shall apply:

- Operation errors and rejection of TCAP components shall be transmitted to the SSF respectively 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 SSF respectively the SRF shall abort the dialogue with a TC-U-ABORT request primitive.

The SCF aborts a dialogue with a TC-U-ABORT request primitive in case of no reply to an ActivityTest operation sent to the SSF.

10.2.2 SSF/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.

If the error processing in the SSF/SRF leads to the case where the SSF/SRF is not able to process further SCF operations while the dialogue is to be maintained, the SSF/SRF aborts the dialogue with a TC-U-ABORT request primitive.

The SSF aborts a dialogue with a TC-U-ABORT request primitive in case call release is initiated by any other entity then the SCF and the SSF has no pending call information requests (or pending requests which should be treated in the same way, see Note 1 section 7.1.5) nor any armed EDP to notify the SCF of the call release.

The SSF also aborts a dialogue with a TC-U-ABORT request primitive in case of no resources after a TC-Begin indication or on expiry of the application timer tSSF.

10.3 Dialogue establishment

The establishment of an INAP dialogue involves two application processes as described in section 4.3, one that is the dialogue-initiator and one that is the dialogue-responder.

This procedure is driven by the following signals:

- A TC-BEGIN request primitive from the dialogue-initiator.

- A TC-BEGIN indication primitive occurring at the responding side

- The first TC-CONTINUE indication primitive occurring at the initiating side or under specific conditions:
  - A TC-END indication primitive occurring at the initiating side
  - A TC-U-ABORT indication primitive occurring at the initiating side
  - A TC-P-ABORT indication primitive occurring at the initiating side

10.3.1 Sending of a TC-BEGIN request primitive

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

SACF shall request the invocation of the associated operations using the TC-INVOKE service. See section 10.8 for a description of the invocation procedure.

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

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

If no TC indication primitive is expected because no dialogue is to be established according to the rules as stated in section 10.1 and 10.2, SACF will wait for the last associated TCAP operation timer to expire and issue a TC-END request with the release method parameter set to "pre-arranged release".

10.3.2 Receipt of a TC-BEGIN indication

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

- Analyse the application-context-name included in the primitive and if it is supported, process any other indication primitives received from TC as described in section 10.8.

Once all the received primitives have been processed, SACF does not accept any primitive from TC, except a TC-P-ABORT indication.

- If no dialogue is to be established according to the rules as stated in section 10.1 and 10.2, SACF will wait for the last indication primitive from TC and issue a TC-END request with the release method parameter set to "pre-arranged release".

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

10.3.3 Receipt of the first TC-CONTINUE ind

On receipt of the first TC-CONTINUE indication primitive for a dialogue, SACF shall check the value of the application-context-name parameter. If this value matches the one used in the TC-BEGIN request primitive, SACF shall process the following TC component handling indication primitives as described in section 10.8, otherwise it shall issue a TC-U-ABORT request primitive.

10.3.4 Receipt of a TC-END ind

On receipt of a TC-END indication primitive in the dialogue initiated state, SACF shall check the value of the application-context-name parameter. If this value does not match the one used in the TC-BEGIN request primitive, SACF shall issue a TC-U-ABORT request primitive, otherwise it shall process the following TC component handling indication primitives as described in section 10.8.

10.3.5 Receipt of a TC-U-ABORT ind

Receipt of a TC-U-ABORT indication primitive is described as part of user abort procedure (See 10.6.2).

10.3.6 Receipt of a TC-P-ABORT ind

Receipt of a TC-P-ABORT indication primitive is described as part of provider abort procedure (See 10.7.1).

10.4 Dialogue continuation

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

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

10.4.1 Sending entity

SACF shall process any component handling request primitives as described in section 10.8

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

10.4.2 Receiving entity

On receipt of a TC-CONTINUE indication primitive SACF shall accept zero, one or several TC component handling indication primitives and process them as described in section 10.8.

10.5 Dialogue termination

Both the dialogue-initiator and the dialogue-responder have the ability to request the termination of a dialogue after it has been established when no dialogue is to be established or when a dialogue is no longer to be maintained according to the rules as stated in section 10.1 and 10.2..

The dialogue termination procedure is driven by the following events:

- A TC-END request primitive

- A TC-END indication primitive

10.5.1 Sending of TC-END request

When the dialogue shall no longer be maintained, SACF shall process any component handling request primitives as described in section 10.8

After processing the last component handling request primitive (if any), SACF shall issue a TC-END request primitive with the release method parameter set to "basic end" or "prearranged release', according to the rules as stated in section 10.1 and 10.2.

When no dialogue is to be established, refer to sections 10.3.1 and 10.3.2.

10.5.2 Receipt of a TC-END indication

On receipt of a TC-END indication primitive, the SACF shall accept any component handling indication primitives and process them as described in section 10.8.

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

10.6 User Abort

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

The user abort procedure is driven by one of the following events:

- A TC-U-ABORT request primitive

- A TC-U-ABORT indication primitive

10.6.1 Sending of TC-U-ABORT request

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

10.6.2 Receipt of a TC-U-ABORT indication

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

10.7 Provider Abort

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

The provider abort procedure is driven by the following event:

- A TC-P-ABORT indication primitive

10.7.1 Receipt of a TC-P-ABORT indication

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

10.8 Procedures for INAP operations

This section describes the procedures for INAP operations.

10.8.1 Operation invocation

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

10.8.2 Operation invocation receipt

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

- If the invoke ID is already in use by an active operation, request the transfer of a reject component using the TC-U-REJECT request primitive with the appropriate problem code (duplicated invokeID)

- If the operation code does not correspond to an operation supported by the application-context, request the transfer of a reject component using the TC-U-REJECT request primitive, with the appropriate problem code (unrecognized operation)

- If a linked ID is included, perform the following checks: If the operation referred to by the linked ID does not allow linked operations or if the operation code does not correspond to a permitted linked operation, issue a TC-U-REJECT request primitive with the appropriate problem code (linked response unexpected or unexpected linked operation)

- If the type of the argument is not the one defined for the operation, request the transfer of a reject component using the TC-U-REJECT request primitive, with the appropriate problem code (mistyped parameter)

- Otherwise, accept the TC-INVOKE indication primitive. If the operation is to be user confirmed, SACF waits for the corresponding response.

10.8.3 Operation Response

For user confirmed operations, SACF shall:

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

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

10.8.4 Receipt of a response

10.8.4.1 Receipt of TC-RESULT-NL indication

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

- Request the transfer of a reject component using the TC-U-REJECT request primitive, with the appropriate problem code (mistyped parameter). SACF shall also issue a TC-U-CANCEL request primitive so that all subsequent result components for this operation are discarded by TC.

10.8.4.2 Receipt of TC-RESULT-L indication

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

- If the type of the result parameter is not the one defined for the result of this operation, request the transfer of a reject component using the TC-U-REJECT request primitive, with the appropriate problem code (mistyped parameter)

- Otherwise, accept the TC-RESULT-L indication primitive

10.8.4.3 Receipt of TC-U-ERROR indication

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

- If the error code is not defined for the SACF or is not one associated with the operation referred to by the invoke identifier, request the transfer of a reject component using the TC-U-REJECT request primitive, with the appropriate problem code (unrecognized error or unexpected error)

- If the type of the error parameter is not the one defined for this error, request the transfer of a reject component using the TC-U-REJECT request primitive, with the appropriate problem code (mistyped parameter)

- Otherwise, accept the TC-U-ERROR indication primitive.

10.8.4.4 Receipt of TC-U-REJECT indication

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

10.8.4.5 Receipt of a TC-L-REJECT indication

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

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

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

Note that when the problem code indicates a general problem, it is considered that the event cannot be related to an active operation even if the invoke Id is provided by TC. This is because it is unclear whether the invoke Id refers to a local or remote invocation. The behaviour of SACF in such a case is described in section 10.8.5.3.

10.8.4.6 Receipt of a TC-L-CANCEL indication

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

- If the associated operation is a class 1 operation, inform the application process.

- If the associated operation is a class 2 operation and no linked operations are defined for this operation, ignore the primitive.

- If the associated operation is a class 2 operation and has linked operations but none of them has been invoked, inform the application process.

- If the associated operation is a class 2 operation and a linked operation invocation has already been received in response to this operation, ignore the primitive.

- If the associated operation is a class 3 operation, inform the application process.

- If the associated operation is a class 4 operation, ignore the primitive.

10.8.5 Other events

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

10.8.5.1 Receipt of a TC-U-REJECT

On receipt of a TC-U-REJECT indication primitive which does not affect an active operation (i.e. indicating a return result or return error problem), it is up to the application process to abort, continue or terminate the dialogue, if not already terminated by the sending application process according to the rules as stated in section 10.2. This is also applicable for invoke problems related to a class 4 linked operation.

10.8.5.2 Receipt of a TC-R-REJECT indication

On receipt of a TC-R-REJECT indication (i.e. when a protocol error has been detected by the peer TC entity) which does not affect an active operation, it is up to the application process to abort, continue or terminate the dialogue, if not already terminated by the sending application process according to the rules as stated in section 10.2.

10.8.5.3 Receipt of a TC-L-REJECT indication

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

10.9 Mapping on to TC services

10.9.1 Dialogue control

The TC-UNI service is not used by INAP.

10.9.1.1 Destination address

This parameter is set by the dialogue initiating application process

10.9.1.2 Originating address

This parameter is set by the dialogue initiating application process

10.9.1.3 Dialogue Id

The value of this parameter is associated with the INAP invocation in an implementation dependent manner.

10.9.1.4 Application-context-name

The application-context-name parameter is set by SACF as defined in section 6.4.

10.9.1.5 User information

This parameter is set by the dialogue initiating application process

10.9.1.6 Component present

This parameter is used by SACF as described in recommendation Q.771.

10.9.1.7 Termination

The value of the release method parameter of the TC-END request primitive is set by SACF according to the rules as stated in sections 10.1 and 10.2.

10.9.1.8 Quality of service

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

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

10.9.2 Operation procedures

10.9.2.1 Invoke Id

This parameter is set by the sending application process.

10.9.2.2 Linked Id

This parameter is set by the sending application process.

10.9.2.3 Dialogue Id

The value of this parameter is associated with the INAP invocation in an implementation dependent manner.

10.9.2.4 Class

The value of this parameter is set by SACF according to the type of the operation to be invoked according to section 6.1.

10.9.2.5 Operation

The operation code of a TC-INVOKE request primitive is set by the sending application process as defined in section 6.4

SACF shall set the operation code of the TC-RESULT-L primitive (if required) to the same value as the one received at invocation time.

10.9.2.6 Error

The error parameter of the TC-U-ERROR request primitive is set by the sending application process as defined in section 6.4.

10.9.2.7. Parameters

The argument parameter of TC-INVOKE primitives is set by the sending application process as defined in sections 6.1 and 6.3.

The result parameter of TC-RESULT-L primitives is set by the sending application process as defined in sections 6.1 and 6.3.

The parameter of TC-U-ERROR primitives are set by the sending application process as defined in sections 6.2 and 6.3

10.9.2.8 Time out

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

10.9.2.9 Last component

This parameter is used by SACF as described in recommendation Q.771.

10.9.2.10 Problem code

This parameter is used by SACF as described in section 10.8.

Teil 2: Use of SCCP

1 SCCP Class

Only the sequenced connectionless service of the SCCP is requested (SCCP class 1).

2 SCCP addressing

2.1 General

As the SSP and the SCP are part of different MTP networks only global title addressing is used for the calling and the called address and no signalling point code is included in the SCCP messages.

The format and coding of address parameters carried by the SCCP for that purpose should comply with recommendation CCITT Q.713 with the following restrictions:

a) Called Address

For the encoding of the parameters refer to the national SCCP specification 163 TR 73, Anhang 2.

b) Calling address

see called address

2.2 Addressing of the Service Switching Point

There are two cases where it is necessary to address the SSP.

2.2.1 Call associated dialogues

In this case the SSP will have initiated the dialogue and will have given its E.164 address in the calling address field of the initiating message.

2.2.2 Service filtering/Call Gapping

In case of activation of service filtering or call gapping the SCP will send the activation message to each SSP known at the SCP. The SCP uses the E.164 addresses of these SSPs to address them.

2.3 Addressing of the Service Control Point

There are three cases where it is necessary to address the SCP.

2.3.1 Call associated dialogues

When the dialogue is initiated the SSP will use the dialled IN number as Global Title for office based trigger for TDP3. For all other cases the Global Title is derived from administrable tables. The SCP will give its E.164 address in the calling address field of the response message. This address is used for all subsequent messages to the SCP within this dialogue.

2.3.2 Service filtering

In this case the SCP will have given its E.164 address in the calling address field of the activation message.

2.3.3 Call associated dialogues for the Assist case

When the dialogue is initiated, the assisting SSP will use the address information in the SCF id field received from the initiating SSP as Global Title. The SCP will give its E.164 address in the calling address field of the response message. This address is used for all subsequent messages to the SCP within this dialogue.

3 Return option

Another SCCP support is the possibility to use a "return option" if the message cannot be delivered. By this it is possible to return the message in case the SCCP cannot deliver the message to the addressed TCAP application.

The "return option" is used in the IN context either for the set up of each TCAP dialogue only or for all TCAP messages. Whether the "return option" is used only for TC-BEGIN messages or for all TCAP messages is implementation dependent.

If the TC-BEGIN message is returned, the SSF application shall release the corresponding call. The release cause shall be in accordance with the received return cause.

Each new call, resulting in a new dialogue, is as such used to test the recovery of the CCSS-N7 network. This way no timings are to be introduced to test if the CCSS-N7 network is recovered.

Teil 3: Festlegungen zu netzspezifischen Parametern und Werten

1 Specification of FCIBillingChargingCharacteristics

```
Telekom-INAP-CS1-FCIBCC

DEFINITIONS IMPLICIT TAGS ::= BEGIN

FCIBCC ::= SEQUENCE {
        chargedPartyId                  [0] ChargedPartyId,
        chargingInformation             [1] ChargingInformation
        }

ChargedPartyId ::= OCTET STRING (SIZE(minChargedPartyIdLen..maxChargedPartyIdLen))

-- This parameter represents the party to be charged. It contains a subscriber identification
-- and possibly additional information, e.g. the 'Kostenstelle' for VPN.

-- From the SSF point of view the content of this parameter is transparent,
-- i.e. each value (ISO7 bit coded) will be accepted and stored into the ticket.

ChargingInformation ::= CHOICE {
        initChargingInformation         [0] InitChargingInformation    }

-- For the parameter 'chargingInformation', which contains all relevant data
-- to generate a subscriber related ticket for a certain 'chargedPartyId',
-- at present only one choice is possible:

-- 'initChargingInformation':
--      It is used in case the SCF sends data for a new connection configuration

--      or the release of the call is requested.
--      This means, the FCI is sent before a CON, CTR or ETC operation
--      or before a RC operation, respectively.


InitChargingInformation ::= SEQUENCE {
        callTypeAndStatus               [1] CallTypeAndStatus,
        billedItemList                  [2] BilledItemList          OPTIONAL,
        transparentData                 [3] TransparentData         OPTIONAL
        }

CallTypeAndStatus ::= INTEGER (0..999)

-- This parameter contains an IN specific cause.

BilledItemList ::= SEQUENCE SIZE(minBilledItemNum..maxBilledItemNum)
                                        OF BilledItem

BilledItem ::= OCTET STRING (SIZE(fixBilledItemLen))

-- The parameter 'billedItemList' consists of a number of single 'billedItems'.
-- Each 'billedItem' is used to generate a specific amount of charge.

-- Currently only one 'billedItem' is used, which contains the 'destinationType' (1 Byte).
-- In case, the SCF has not sent this parameter, the SSF provides the default
-- value H'00 for the ticket. Otherwise the received value is included into
-- the ticket in a transparent manner.
```

```
TransparentData ::= OCTET STRING (SIZE(minTransparentDataLen..
                                      maxTransparentDataLen))

-- This parameter comprises an integral number of octets.
-- It can be used to transfer transparent service specific additional billing information
-- from the service logic to the ticket postprocessing application.
-- The internal structure of the transparent data has to be agreed
-- between the service logic and the postprocessing centre.
-- From the SSF point of view these data are included into the ticket in a
-- transparent manner.
-- The specifications for the ticket generation in the fixed or mobile network have to
-- define a handling for the case that the parameter is missing or the length of the
-- parameter is longer than the length which is defined for the ticket (there may be
-- different handlings for the different networks).

-- The following Size Constraints are valid:

fixBilledItemLen        INTEGER ::= 1
minBilledItemNum        INTEGER ::= 1
maxBilledItemNum        INTEGER ::= 1
minChargedPartyIdLen    INTEGER ::= 1
maxChargedPartyIdLen    INTEGER ::= 16
minTransparentDataLen   INTEGER ::= 2
maxTransparentDataLen   INTEGER ::= 20

END
```

2 Specification of SCIBillingChargingCharacteristics

```
Telekom-INAP-CS1-SCIBCC

DEFINITIONS IMPLICIT TAGS ::= BEGIN

IMPORTS
        DateAndTime
FROM Telekom-INAP-CS1-DataTypes {ccitt(0)  administration(2) bmpt(262) telekom(1) zgs_nr7(5) inap(0)
modules(0) cs1-datatypes(2) ~~version4(3)~~version5(4)}

        AocParameters,
        Integer2
FROM Telekom-INAP-CS1-AC
;

SCIBCC ::= SEQUENCE {
        chargingLevel                   [0] ChargingLevelControl        OPTIONAL,
        adviceOfCharge                  [1] AocInformation              OPTIONAL
        }

-- The parameter ‘adviceOfCharge’ contains the GSM AoC information.

ChargingLevelControl ::= CHOICE {
        scpChargeNumber                 [0] ScpChargeNumber,
        zoningRequired                  [1] SSPZoningInformation
        }

ScpChargeNumber ::= OCTET STRING (SIZE(fixScpChargeNumberLen))

fixScpChargeNumberLen INTEGER ::= 2

-- The parameter 'scpChargeNumber' contains a charge number with
-- the possible values 0 or 2-255 (binary coded as for realization 96).

SSPZoningInformation ::= CHOICE {
        aParty-BParty                   [0] NULL
        }

-- This parameter gives instructions to be used by the SSP zoning function.
-- More precisely the origin as well as the destination is given, which are
-- necessary to derive the appropriate zoning information.

-- At present only one choice is possible, i.e. the charging/billing of the calling
-- line depends on the distance from the A- to the B-Party.
-- In case the SSP is not the tariff determination point this means, that
-- backward charging information is required before connection establishment.

AocInformation ::= SEQUENCE {
        aocParameters                   [0] AocParameters,
        changedAocParameters            [1] AocParameters               OPTIONAL,
        switchTime                      SwitchTime                      OPTIONAL
        }
```

```
-- The 'aocParameters' provide the Charge Advice Information (CAI) according
-- GSM 02.24 to be applied after the answer indication.
-- The 'changedAocParameters' provide the Charge Advice Information (CAI) according
-- GSM 02.24 to be applied after the tariff switch.
-- The 'switchTime' fixes the point in time the tariff has to be switched.

-- For the definition of the 'AocParameters' see specification 'of
-- AChBillingChargingCharacteristics.

SwitchTime ::= CHOICE {
        timeTillSwitch                  [2] Integer2,
        dateAndTime                     [4] DateAndTime
        }

-- The 'timeTillSwitch' indicates the time interval (in seconds) until the tariff switch. The time
-- measurement has to be started after the receipt of the operation.
-- The 'dateAndTime' specifies the point of tariff switch in an absolute manner.

END
```

3 Specification of SFBillingChargingCharacteristics

```
Telekom-INAP-CS1-SFBCC

DEFINITIONS IMPLICIT TAGS ::= BEGIN

IMPORTS
        ScpChargeNumber
FROM Telekom-INAP-CS1-SCIBCC

        NatCallingPartysCategory
FROM Telekom-INAP-CS1- Data Types {ccitt(0)  administration(2) bmpt(262) telekom(1) zgs  nr7(5) inap(0)
modules(0) cs1-datatypes(2) version5(4)}

;

SFBCC ::= SEQUENCE {
        scpChargeNumber                 [0] ScpChargeNumber         OPTIONAL
        natCpcCharge                    [1] NatCpcCharge OPTIONAL }

natCpcCharge ::= SEQUENCE {
        scpChargeNumberNatCPC       [0] ScpChargeNumber,
        natCallingPartysCategory[1] NatCallingPartysCategory }

--
--1.    If 'natCpcCharge' has not been sent by the SCF, the following behaviour shall be applied:
--      In case, an 'scpChargeNumber' has been sent by the SCF, all filtered calls are charged according
--      to the 'scpChargeNumber' indicated. The 'scpChargeNumber' is used as already
--      defined for 'sCIBillingChargingCharacteristics', this means the values 0 and 2-255 are applicable.}

-- For this argument the same parameter 'scpChargeNumber' is used as already
-- defined for 'sCIBillingChargingCharacteristics', this means the values 0 and 2-255
-- are applicable.

-- 2.   In case, an 'scpChargeNumber' has not been sent by the SCF, this will be interpreted
--      by the SSF in the sense of 'no IN charge control'. This means, all filtered calls will be
--      charged according to the normal charging mechanism of the underlaying basic network.


--      If 'natCpcCharge' has been sent by the SCF, the following behaviour shall be applied:

--      1.  Filtered calls of which the 'natCallingPartysCategory' parameter matches the
--      natCallingPartysCategory in 'natCpcCharge' are charged as indicated in the
--      'scpChargeNumberNatCPC'.

--      2. Filtered calls with other values of 'natCallingPartyCategory' are charged as indicated above for the
--      case when no 'natCpcCharge' has been sent by SCF.


END
```

4 Specification of User Abort Information

```
Telekom-UserAbortInformation {
        ccitt                   (0)
        administration          (2)
        bmpt                    (262)
        telekom                 (1)
        zgs_nr7                 (5)
        inap                    (0)
        modules                 (0)
        userAbortInfo           (4)
        version1                (0)}

DEFINITIONS IMPLICIT TAGS ::= BEGIN

EXPORTS
        telekom-UserAbortInfo-as-id,
        Telekom-UserAbortInfo;

IMPORTS
        Cause
FROM Telekom-INAP-CS1-DataTypes {
        ccitt                   (0)
        administration          (2)
        bmpt                    (262)
        telekom                 (1)
        zgs_nr7                 (5)
        inap                    (0)
        modules                 (0)
        cs1-datatypes           (2)
        version2                (1)};

--================================================================--
-- Specification of User Abort Information Object Identifier
--================================================================--
telekom-UserAbortInfo-as-id OBJECT IDENTIFIER ::= {
        ccitt                   (0)
        administration          (2)
        bmpt                    (262)
        telekom                 (1)
        zgs_nr7                 (5)
        inap                    (0)
        userAbortInfoASId       (2)
        version1                (0)}

--================================================================--
-- Specification of User Abort Information Type
--================================================================--
Telekom-UserAbortInfo ::= SEQUENCE {
        userAbortCause          [0] UserAbortCause ,
        releaseCause            [1] Cause OPTIONAL }

--================================================================--
-- Specification of User Abort Cause
--================================================================--
UserAbortCause ::= CHOICE {
        abortedBySSP            [0] AbortedBySSP ,
```

```
        abortedBySCP                    [1] AbortedBySCP }

--================================================================--
-- Specification of Aborted-By-SSP Type
--================================================================--
AbortedBySSP ::= CHOICE {
        networkEvent                    [0] NetworkEvents ,
        interfaceEvent                  [1] InterfaceEventsSSF ,
        internalEvent                   [2] InternalEventsSSF ,
        srfEvent                        [3] SrfEventsSSF ,
        undetermined                    [4] NULL }

NetworkEvents ::= ENUMERATED {
        general                         (0) ,
        callerAbandon                   (1) ,
        callerDisconnect-NoCIRreq       (2) ,
        callReleasedByNetwork           (3) }

InterfaceEventsSSF ::= ENUMERATED {
        general                         (0) ,
        sSF-TimerExpired                (1) ,
        errRejWithContinue              (2) ,
        in-In-Interaction               (3) ,
        chargingError                   (4) }
-- errRejWithContinue: ReturnError or Reject Component in a Continue-Message
-- received.

InternalEventsSSF ::= ENUMERATED {
        general                         (0) ,
        routeSelectFailure              (1) ,
        activityTestFailed              (2) ,
        initiatingRelease               (3) }

SrfEventsSSF ::= ENUMERATED {
        general                         (0) ,
        srfError                        (1) }

--================================================================--
-- Specification of Aborted-By-SCP Type
--================================================================--
AbortedBySCP ::= CHOICE {
        interfaceEvent                  [1] InterfaceEventsSCF ,
        internalEvent                   [2] InternalEventsSCF ,
        undetermined                    [3] NULL }

InterfaceEventsSCF ::= ENUMERATED {
        general                         (0) ,
        at-OpTimerExpired               (1) }

InternalEventsSCF ::= ENUMERATED {
        general                         (0) }

END
```

5 Specification of NatServiceInteractionIndicator for Connect and ConnectToResource

```
Telekom-INAP-CS1-NatSII

DEFINITIONS IMPLICIT TAGS ::= BEGIN

IMPORTS
        LocationNumber
FROM Telekom-INAP-CS1-DataTypes {ccitt(0)  administration(2) bmpt(262) telekom(1) zgs_nr7(5) inap(0)
modules(0) cs1-datatypes(2) ~~version4(3)};~~version5(4)};

NSII   ::= SEQUENCE {
        inToNetworkBitInd               [0] InToNetworkBitInd           OPTIONAL,
        inNumber                        [1] InNumber                    OPTIONAL,
        locationNumber                  [2] LocationNumber              OPTIONAL,
        nPSSP                           [3] NPSSP                       OPTIONAL,
        cUGInterlockCode                [4] CUGInterlockCode            OPTIONAL}

InToNetworkBitInd ::= SEQUENCE {
        inToNetworkBitMask              [0] InToNetworkBitMask,
        inToNetworkBitValues            [1] InToNetworkBitValues}

-- This parameter contains a set of flags, which are used by the SSF to derive different network
-- relevant instructions.
-- The 'inToNetworkBitMask' is a mask, which indicates whether the according information
-- in the 'inNetworkBitValues' is valid or not. For all bits, for which the transmitted value is
-- not valid the SSF uses default values.
-- In case, this parameter has not been sent by the SCF, the SSF uses corresponding
-- default values.

InToNetworkBitMask      ::= BIT STRING {
        freePhoneInd                    (0),
        longUIDInd                      (1)} (SIZE(1..8))

InToNetworkBitValues    ::= BIT STRING {
        freePhoneInd                    (0),
        longUIDInd                      (1)} (SIZE(1..8))

-- The default value for 'FreePhoneInd' is equal to 'FALSE'.
-- The default value for 'LongUIDInd' is equal to 'TRUE'.
-- For further information refer to ~~Teil 4.~~[7].

InNumber    ::= OCTET STRING (SIZE(minInNumberLen .. maxInNumberLen))

-- The parameter inNumber has the purpose to display the dialled number (by calling party)
-- at the called party side.
--
-- Refer to the 163 TR 75 parameter NP.INCdPNo for encoding.
--
-- The parameter inNumber will only be used by the SSP within a Connect operation.
-- The parameter inNumber will be ignored by the SSP within a ConnectToResource
-- operation.
--
-- If the parameter inNumber is present it is used by the SSP to generate the ISUP
-- parameter NP.INCdPNo.
```

```
--
-- If the parameter inNumber is not present the ISUP parameter NP.INCdPNo will be generated
-- using the CdPNo received from the calling party. In this case the restriction indicator within
-- the NP.INCdPNo will always be set to 'presentation not allowed'.
--
-- For further information refer to Teil 4.[7].

-- LocationNumber ::= OCTET STRING (SIZE(minLocationNumberLen..maxLocationNumberLen))
--
-- The definition of the data type ‘LocationNumber’ is included in comments because it is imported.
--
-- With respect to encoding of the octet string refer to Q.763 / 'Location number'.
--
-- The parameter ‘locationNumber’ has the purpose to display the location of the calling party at the called
-- party side.
--
-- It will only be handled by the SSP, if it is received within a Connect operation.
-- It will be ignored by the SSP, if it is received within a ConnectToResource operation.
--
-- If it is present, it will be used by the SSP to generate the ISUP parameter 'Location number' or to overwrite
-- an already existing - received with an IAM - parameter 'Location number'.
--
-- For further information refer to Teil 4.[7].

NPSSP       ::= OCTET STRING (SIZE (minNPSSPLen .. maxNPSSPLen))

-- Refer to 163 TR 75 for encoding.
--
-- For further information refer to Teil 4.[7].

CUGInterlockCode::= OCTET STRING (SIZE (minCUGInterlockLen .. maxCUGInterlockLen))

-- Refer to 163 TR 75 for encoding.
-- If present, the SSP shall include this parameter in the outgoing Initial Address Message (IAM)
-- according the rules given in 163 TR 75.
--
-- For further information refer to Teil 4.[7].

minInNumberLen          INTEGER ::= 2
maxInNumberLen          INTEGER ::= 10
minNPSSPLen             INTEGER ::= 2
maxNPSSPLen             INTEGER ::= 2
minCUGInterlockLen      INTEGER ::= 4
maxCUGInterlockLen      INTEGER ::= 4

END
```

6 Other Parameters and Values

6.1 Value "0" for ServiceKey

The value "0" for ServiceKey is reserved for special purposes in connection with the operation CallGap. Therefore this value should not be used in the operation InitialDP.

6.2 Coding of Characters * and # in PromptAndCollectUserInformation

The characters * and # which may be used in the operation PromptAndCollectUserInformation are specified as follows:

PromptAndCollectUserInformationArg, sub-parameter collectedDigits

The four least significant bits of each octet in the sub-parameters 'endOfReplyDigit', 'cancelDigit' and 'startDigit' may contain the characters * and #. The following coding must be used:

Character	**Coding**
*	B '1011' (H 'B')
#	B '1100' (H 'C')

ReceivedInformationArg, parameter digitsResponse

The parameter digitsResponse may contain the characters * and #.. The following coding must be used:

Character	**Coding**
*	B '1011' (code 11)
#	B '1100' (code 12)

6.3 Timer Values for T_{SSF} and T_{SRF}

To be specified. At present the following values are implemented:

Siemens AG: T_{SSF} = 10s (in state Waiting For Instructions),

T_{SSF} = 30 min (in all user interaction states),

T_{SRF} = 5s.

Alcatel-SEL AG: T_{SSF} = 60s,

T_{SRF} = 5s.

7 Specification of AChBillingChargingCharacteristics

```
Telekom-INAP-CS1-AC

DEFINITIONS IMPLICIT TAGS ::= BEGIN

IMPORTS
        Cause,
        minAChBillingChargingLength,
        maxAChBillingChargingLength
FROM Telekom-INAP-CS1-DataTypes {ccitt(0)  administration(2) bmpt(262) telekom(1) zgs_nr7(5) inap(0)
modules(0) cs1-datatypes(2) ~~version4(3)};~~version5(4)};

AChBCC  ::=  SEQUENCE {
        callSupervision                 [0] CallSupervision     OPTIONAL
        }

CallSupervision  ::= CHOICE {
        timeMeasurement                 [1] TimeMeasurement
        }

--  This parameter specifies the information necessary for supervising a call in the
--  SSF on behalf of the SCF.

TimeMeasurement  ::= SEQUENCE {
        initTimeSupervision             [0] CallSupervisionTimeBased            OPTIONAL,
        changedTimeSupervision          [1] ChangedCallSupervisionTimeBased     OPTIONAL
        }

--  This parameter contains the  data for call supervision/AoC to be applied by the SSF.
--
--      initTimeSupervision
--      - This parameter specifies the needed information to start a new supervision of the call based on a
--      - given time limit. It can be sent before the setup of a new connection configuration but also during a
--      - stable connection.
--      changedTimeSupervision
--      - If during the next supervision interval a tariff switch - determined by the SCF - occurs, this parameter
--      - will provide data relating to this tariff switch (see below).
--
--  The sub-parameters are sent in the following situations:
--
--      The parameter initTimeSupervision must be always present. If during the time interval
--      to be supervised  a tariff
--      switch is foreseen, additionally the parameter changedTimeSupervision is sent in
--      advance to the SSF.

CallSupervisionTimeBased  ::= SEQUENCE {
        timeGranted                     [0]  MaximumCallDuration~~-----OPTIONAL~~,
        forcedRelease                   [1]  ForcedRelease              OPTIONAL,
        heartBeat                       [2]  INTEGER (1..30)            OPTIONAL,
        adviceOfCharge                  [3]  AocParameters              OPTIONAL
        }
```

-- The timeGranted parameter represents the next time limit (in seconds) to be supervised by the SSF. In SSF.

-- case that the AC operation is received in the context of a new connection setup, the time measurement
-- shall be started after the B-party / IP has answered. If the AC operation is received during a stable
-- connection, the time measurement has to be started immediately.
--
-- The parameter 'forcedRelease' specifies the information for an enhanced call release
-- procedure, if the call is tomust be released immediately after reaching the given time limit.
-- If this parameter is not present, the SSFshall not release
-- themust not release the call itself after reaching the
-- given time limit. Instead of forced release handling, an ACR operation (with indication
-- 'final-callActive') will be sent to the SCF in order to invoke further instructions. If this
-- parameter once is received in the SSP, it must be valid only as long as the SCP does not
-- provide a next AC operation. I.e.: if the SSP receives a next AC operation containing this
-- parameter (new one), the SSP must modify an existing forced release handling acc. to the
-- new characteristics. If a next AC operation does not contain this parameter, the SSP must
-- handle the call as if this parameter was never present. If the SCP did not yet provide this
-- parameter but sends it in a next AC, the SSP must initiate the appropriate handling.
--
-- The parameter 'heartBeat' indicates the time interval (in minutes) after which the SSF must
-- send an ACR operation (with indication 'intermediate'). If the parameter is not provided, no
-- intermediate reports will be requested by the SCF. If this parameter once is received in the
-- SSP, it must be valid only as long as the SCP does not provide a next AC operation. I.e.: if
-- the SSP receives a next AC operation containing this parameter (new one), the SSP must
-- modify an existing Heart-Beat handling acc. to the new characteristics. I.e. the period e.g.
-- may remain the same as before but the timer itself be started again immediately. If a next
-- AC operation does not contain this parameter, the SSP must handle the call as if this
-- parameter was never present. If the SCP did not yet provide this parameter but sends it in
-- a next AC, the SSP must initiate the appropriate handling.
-- **Note: This parameter is contained for backward compatibility reasons of the service**
-- **logic /A.14/.**
-- **In case the account of the subscriber is devided into several partitions this parameter shall**
-- **must not be used any longer.**
--
-- The parameter 'adviceOfCharge' contains the initial set of e-parameters necessary to support
-- the GSM supplementary service Advice of Charge (AoC). It is tomust be applied at the beginning
-- of the connection configuration. If the AOC feature does not apply for the subscriber, this
-- information will be ignored by the CCF. This parameter is for mobile network (T-Mobil) use
-- only and therefore must be ignored, if the fixed network (Telekom) SSP receives it.

```
ChangedCallSupervisionTimeBased ::= SEQUENCE {
        adviceOfCharge                  [0] AocParameters
        OPTIONAL,
        switchTime                      [1] Integer2
        }

Integer2 ::= INTEGER (0..65535)
```

-- The switchTime defines the duration (in seconds) from the receipt of the AC operation until the next tariff
-- switch. In the SSF this event restarts the time measurement in a way that the SSF is able to report two
-- separate time durations (duration until tariff switch and time elapsed since tariff switch) in the next ACR
-- operation.
-- The parameter adviceOfCharge in the ChangedCallSupervisionTimeBased type has to be applied when
-- switchTime is reached.

```
MaximumCallDuration::= INTEGER (0..86400)

--  This type specifies the call duration given in seconds.

ForcedRelease ::= SEQUENCE {
        cause                       [1]  Cause                  DEFAULT normalCause,
        warningBeforeRelease        [2]  WarningTimeBased       OPTIONAL
        }

normalCause Cause ::='829F'H        -- Value 31, normal unspecified --

-- If the maximum call duration is expired, the call will be released immediately with the given cause.
-- Before the call reaches the granted call duration, a warning tone shall be provided to the
-- A- party, specified in the warningBeforeRelease parameter. If this parameter is not
-- present, the call will be released without any warning before.

WarningTimeBased    ::=  SEQUENCE {
        timeBeforeRelease           [0]  INTEGER (1..120) DEFAULT 15    --sec--,
        tone                        [1]  ToneShort              OPTIONAL
        }

--  If the remaining time reaches the value of timeBeforeRelease, the SSF shall play a warning
--  tone to the A- party.
--  If tone is not present, the following default values shall be taken:
--      toneId = 1,
--      duration = 500 msec.

ToneShort  ::= SEQUENCE  {
        toneId                      [0]  INTEGER (1..32),
        duration                    [1]  INTEGER (1..4095)      OPTIONAL}

-- The parameter toneId denotes a specific tone to be played.
-- The parameter duration denotes the tone duration given in milliseconds. The granularity to be applied for
-- this duration is 100 ms.
-- If the duration is not present, the SSP will use a  default value of
-- 500 msec.

AocParameters::=CHOICE {
        chargeAdviceInformation [0]         CAI-GSM0224
        }

-- Depending on the underlying network and access, different information is needed to control the advice
-- of charge (AOC) feature.The Charge Advice Information is supported with respect to GSM 02.24.
-- In the future this structure may be enhanced by AOC Information
-- according to ETS 300 182 to support AoC for the DSS1 interface.

CAI-GSM0224     ::=  SEQUENCE {
        e1                          [1] E1          OPTIONAL,
        e2                          [2] E2          OPTIONAL,
        e3                          [3] E3          OPTIONAL,
        e4                          [4] E4          OPTIONAL,
        e5                          [5] E5          OPTIONAL,
        e6                          [6] E6          OPTIONAL,
        e7                          [7] E7          OPTIONAL
        }

--  This parameter contains a set of 7 e-parameters as specified in GSM 02.24.
```

```
-- All e-parameters relevant for the connection must be contained.
-- The missing parameters are set to 0 in the M-SSP.

-- The e1 parameter defines the number of units incremented per interval. It is set in terms
-- of LPLMN units per interval to a resolution of 0.1.
-- The e2 parameter defines the time interval for unitization, and is specified in seconds, to
-- a resolution of 0.1.
-- The e3 parameter defines the scaling factor to convert from LPLMN units to HPLMN
-- units. It is a dimensionless multiplier given to a resolution of 0.01.
-- The e4 parameter defines the number of units to be incremented on receipt of the
-- message containing the CAI elements. It is specified in units of LPLMN to a resolution of
-- 0.1.
-- The e5 parameter defines the number of units incremented per data interval. It is set in
-- terms of LPLMN units per interval to a resolution of 0.1.
-- The e6 parameter defines the data usage interval for unitization, and is specified in
-- segments (SEG), to a resolution of 1, for Dedicated Access to the PSPDN (whether
-- directly or via Dedicated PAD). It does not apply to circuit switched access to moderns of
-- PADs, (except Dedicated PAD´s) or MS to MS calls.
-- The e7 parameter defines the initial time interval for unitization, and is specified in
-- seconds, to a resolution of 0.1.

E1 ::= INTEGER (0..max10TimesUnitsPerTime)
max10TimesUnitsPerTime INTEGER ::= 8191

E2 ::= INTEGER (0..max10TimesTimeInterval)
max10TimesTimeInterval INTEGER ::= 8191

E3 ::= INTEGER (0..max100TimesScalingFactor)
max100TimesScalingFactor INTEGER ::= 8191

E4 ::= INTEGER (0..max10TimesIncrement)
max10TimesIncrement INTEGER ::= 8191

E5 ::= INTEGER (0..max10TimesIncrementPerDataInterval)
max10TimesIncrementPerDataInterval INTEGER ::= 8191

E6 ::= INTEGER (0..maxNumberOfSegmentsPerDataInterval)
maxNumberOfSegmentsPerDataInterval INTEGER ::= 8191

E7 ::= INTEGER (0..max10TimesInitialTime)
max10TimesInitialTime INTEGER ::= 8191

END
```

8 ~~8~~ Specification of CallResult

```
Telekom-INAP-CS1-ACR

DEFINITIONS IMPLICIT TAGS ::= BEGIN

IMPORTS
        minCallResultLength,
        maxCallResultLength,
        LegID,
        Cause

FROM Telekom-INAP-CS1-DataTypes {ccitt(0)  administration(2) bmpt(262) telekom(1) zgs_nr7(5) inap(0)
modules(0) cs1-datatypes(2) ~~version4(3)~~version5(4)}

MaximumCallDuration

FROM Telekom-INAP-CS1-AC
; -- End of IMPORTS

CallResult ::= OCTET STRING (SIZE (minCallResultLength..maxCallResultLength))

CaRt ::=  SEQUENCE {
        sequenceInfo                [1]  SequenceInfo,
        partyToCharge               [2]  LegID                OPTIONAL,
        supervisionResult           [3]  SupervisionResult    OPTIONAL
        }

--  The sequenceInfo parameter indicates the context the ACR operation is sent in
--  .
--  The partyToCharge parameter is used to correlate the ACR operation to the related AC operation.
--  The partyToCharge has to be set to the same value as received in the related AC operation. If
--  partyToCharge is not present in AC, then this parameter is not sent.
--  The supervisionResult parameter provides the charging related information previously
--  requested via the AC operation.

SequenceInfo ::= ENUMERATED {
        intermediate                (0),
        final-callReleased          (1),
        final-callActive            (2)
        }

--  The type SequenceInfo indicates the situation in the SSF which has caused the sending of the ACR
--  operation.
--  intermediate
--       This indication is sent  in  case  the  ACR  was  sent  after expiration of the heartBeat timer.
--  Note:  This parameter is contained for backward compatibility reasons of the service logic /A.14/.
--   In case that the account of the subscriber is devided into several partitions this parameter shall
--   not be used any longer.
--
--  final-callReleased
--  This indication is sent in all situations in which the actual connection configuration will be released.
--
```

```
-- final-callActive
--      This indication is sent in case the given time limit was reached and the SSF requests the next
--      instruction from the SCF (the parameter forcedRelease has not been sent by the SCF).

SupervisionResult ::= SEQUENCE {
        supervisionMethod               [0]  SupervisionMethod,
        cause                           [1]  Cause                      OPTIONAL
        }

-- The parameter cause is only present, if the ACR is the final one for the regarded connection
-- configuration (final-callReleased). It indicates the release cause.

SupervisionMethod ::= CHOICE {
        timeMeasurement                 [1]  TimeMeasurementResult
        }

TimeMeasurementResult ::= SEQUENCE {
        usedTime                        [0]  MaximumCallDuration,
        usedTimeAfterSwitch             [1]  MaximumCallDuration        OPTIONAL
        }

-- With respect to the parameter usedTime two cases have to be distinguished. In case of no tariff switch
-- in the current supervision interval, the usedTime would contain the time elapsed
-- either since the B-party / IP
-- has answered or since the last tariff switch that occured before the current supervision
-- interval. If there
--  was a tariff switch in the current supervision interval, the usedTime would contain just the time elapsed
-- until tariff switch.
-- In case of unsuccessful connection setup, the value ‘0’ will be reorted to the SCF.
-- The parameter usedTimeAfterSwitch would be only sent if there was a tariff switch during the actual
-- supervision interval. It contains the time elapsed since this tariff switch.

END
```