

























	System file	rw	C:\tau34\Q1228\Inap\inap.sdt
	Source directory	rw	C:\tau34\Q1228\Inap\
	Text readme	--	TCAP\readme.txt

Chapter ASN.1 Files

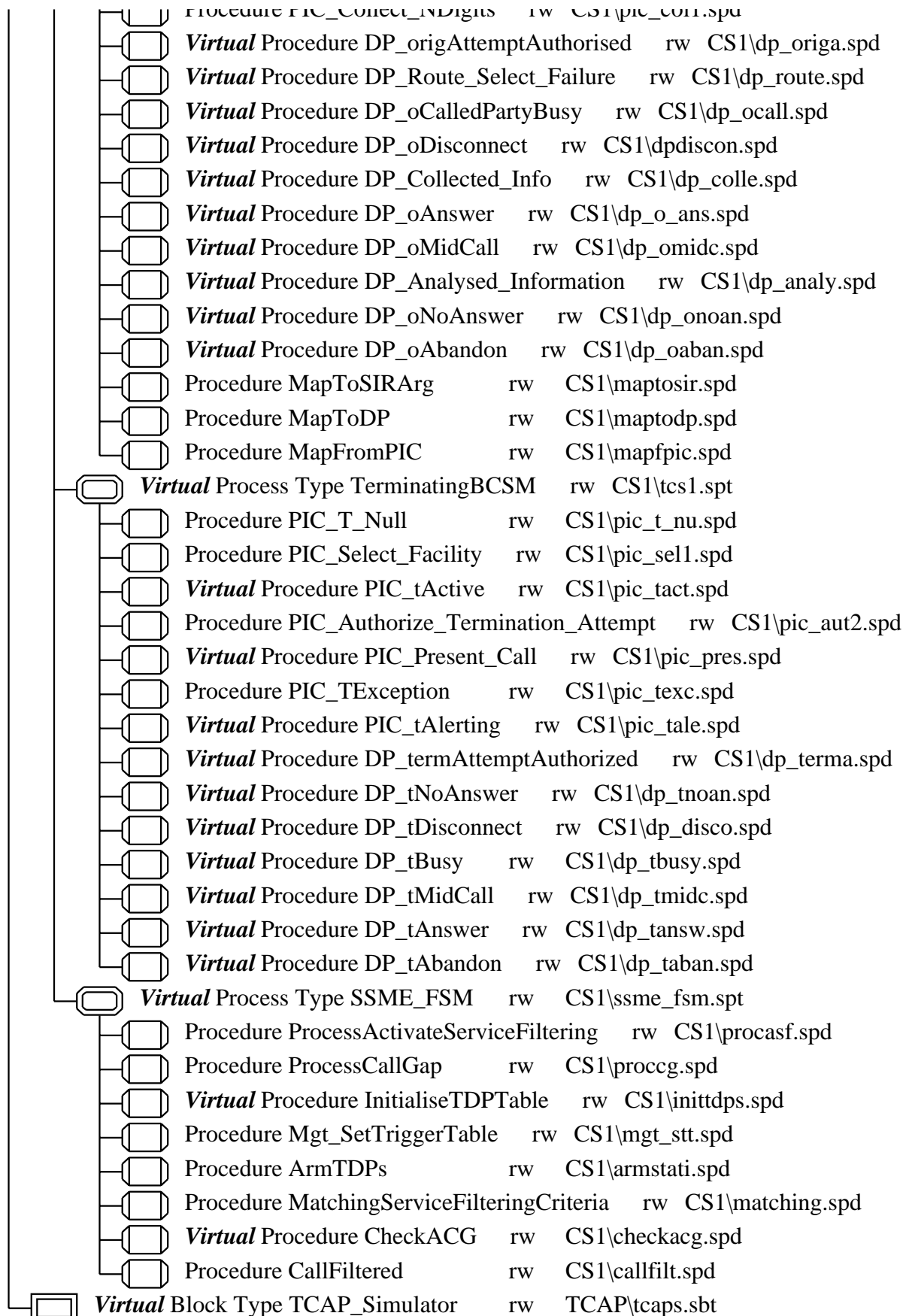
	ASN.1 Text INCS2datatypes	rw	ASN1\incs2-dt.asn
	ASN.1 Text INCS2SSFSCFopsargs	rw	ASN1\ssf-scf.asn
	ASN.1 Text INCS2BundleArg	rw	ASN1\incs2b.asn
	ASN.1 Text INCS2Internals	rw	ASN1\cs2i.asn

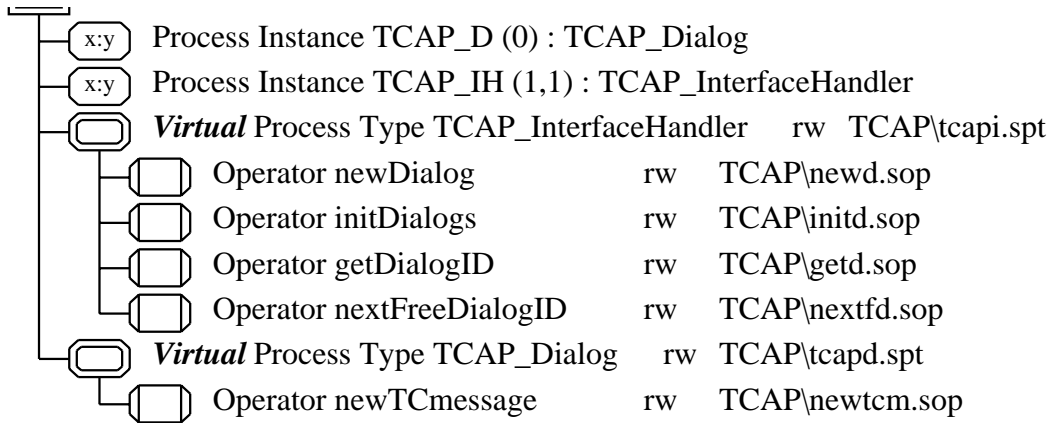
Chapter IN CS-1 Specification

	Package CS1_INAP	rw	CS1\cs1_inap.sun
	Depending on ASN.1 Text INCS2BundleArg		
	Depending on ASN.1 Text INCS2datatypes		
	Depending on ASN.1 Text INCS2Internals		
	Depending on ASN.1 Text INCS2SSFSCFopsargs		
	System Type CS1_INAP	rw	CS1\cs1_inap.sst
	Block Instance SSF_CCF_A : SSF_CCF		
	Block Instance SSF_CCF_B : SSF_CCF		
	Block Instance TCAP_Adapter : TCAP_Simulator		
	Virtual Block Type SSF_CCF	rw	CS1\ssf1.sbt
	Process Instance CS (0) : CallSegment		
	Process Instance CSA (0) : CallSegmentAssociation		
	Process Instance IH (1,1) : InterfaceHandler		
	Process Instance O_BCSM (0) : OriginatingBCSM		
	Process Instance SSF (0) : SSF_FSM		
	Process Instance SSME (1,1) : SSME_FSM		
	Process Instance T_BCSM (0) : TerminatingBCSM		
	Virtual Process Type InterfaceHandler	rw	CS1\ih.spt
	Procedure AllocateCSAId	rw	CS1\alldid.spd
	Procedure AddCSAId	rw	CS1\adddid.spd
	Procedure GetCSAIdfromCSA	rw	CS1\getdid.spd
	Procedure GetCSAfromCSAId	rw	CS1\getcvd.spd
	Procedure GetCSAfromSigConId	rw	CS1\getcvu.spd

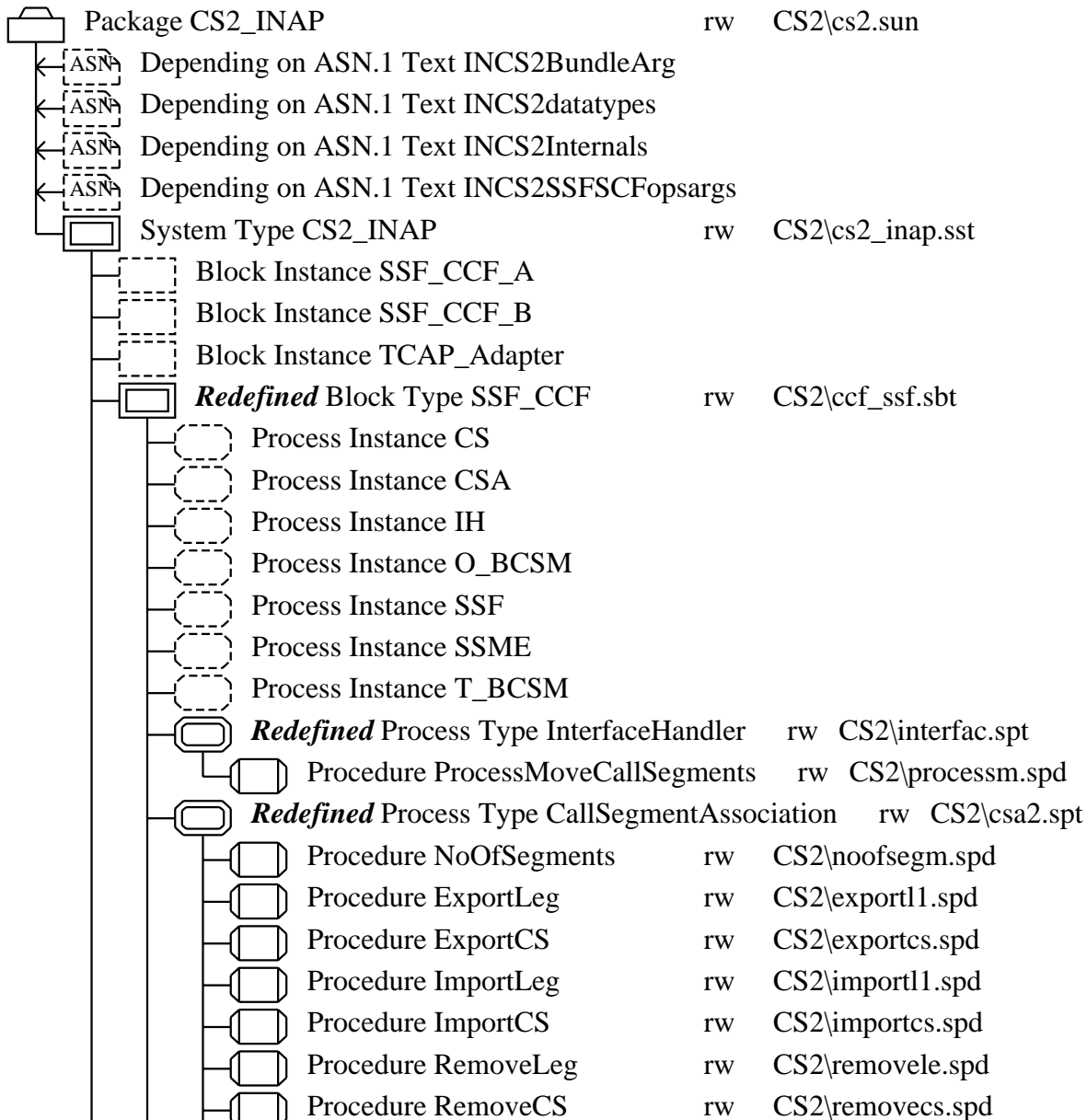
	Procedure SetSigConAssoc	rw	CS1\setsca.spd
	Procedure SetRemoteAssoc	rw	CS1\setrema.spd
	Procedure GetCSAfromRemoteId	rw	CS1\getlcsa.spd
	Procedure IsCSA	rw	CS1\iscsa.spd
	Virtual Process Type CallSegmentAssociation	rw	CS1\csa.spt
	Procedure AddCS	rw	CS1\addcs.spd
	Procedure SetLegLocation	rw	CS1\setlegl.spd
	Procedure GetCSPtr	rw	CS1\getcsptr.spd
	Procedure GetLegLocation	rw	CS1\getlegl.spd
	Procedure ExistCS	rw	CS1\existcs.spd
	Procedure SetLegAssoc	rw	CS1\setlassc.spd
	Procedure IsCS	rw	CS1\iscs.spd
	Procedure GetLegIdfromRemoteLegId	rw	CS1\getlegrc.spd
	Procedure ExistLeg	rw	CS1\existleg.spd
	Virtual Process Type CallSegment	rw	CS1\cv1.spt
	Procedure AddLeg	rw	CS1\addleg.spd
	Procedure RemoveLeg	rw	CS1\remleg.spd
	Procedure SetLegStatus	rw	CS1\setlstat.spd
	Procedure ReleaseAllLegs	rw	CS1\releasea.spd
	Procedure GetLegStatus	rw	CS1\getlstat.spd
	Procedure SetLegAssoc	rw	CS1\setlass.spd
	Procedure SetLegPtr	rw	CS1\setlptr.spd
	Procedure GetLegIdfromRemoteLegId	rw	CS1\getlegrc.spd
	Procedure GetLegPtr	rw	CS1\getlptr.spd
	Procedure IsBCSM	rw	CS1\isbcsm.spd
	Procedure GetPassiveLegId	rw	CS1\getpl.spd
	Procedure MapConnectToBCSM	rw	CS1\mpcobcsm.spd
	Procedure MapSIToBCSM	rw	CS1\mpsibcsm.spd
	Virtual Process Type SSF_FSM	rw	CS1\ssf_fsm.spt
	Virtual Procedure InitialiseDPTable	rw	CS1\initavl.spd
	Procedure ExistLeg	rw	CS1\exleg.spd
	Procedure AnyDPArmed	rw	CS1\anyevarm.spd
	Procedure IsDPArmed	rw	CS1\evarm.spd
	Procedure DisarmDPs	rw	CS1\disarm.spd
	Procedure DPArmed	rw	CS1\dparmed.spd
	Procedure CallInformationReportPending	rw	CS1\callinfo.spd
	Procedure ApplyChargingReportPending	rw	CS1\applycha.spd
	Procedure ArmTDPs	rw	CS1\armtdps.spd

	Procedure MatchingServiceFilteringCriteria	rw	CS1\msfc.spd
	Procedure CheckACG	rw	CS1\cacg.spd
	Procedure CallFiltered	rw	CS1\cf.spd
	Procedure ProcessApplyCharging	rw	CS1\process6.spd
	Procedure ProcessContinue	rw	CS1\proces12.spd
	Procedure ProcessRequestReportBCSMEvent	rw	CS1\prreqrep.spd
	Procedure ProcessAnalyseInformation	rw	CS2\procai.spd
	Procedure ProcessCallInformationRequest	rw	CS1\process7.spd
	Procedure ProcessDisconnectForwardConnection	rw	CS1\processd.spd
	Procedure ProcessSendChargingInformation	rw	CS1\procsci.spd
	Procedure ProcessSelectRoute	rw	CS2\procsr.spd
	Procedure ProcessCancel	rw	CS1\process8.spd
	Procedure ProcessEstablishTemporaryConnection	rw	CS1\processe.spd
	Procedure ProcessEventReportBCSM	rw	CS1\prevrep.spd
	Procedure ProcessSelectFacility	rw	CS2\procsf.spd
	Procedure ProcessCollectInformation	rw	CS1\process9.spd
	Procedure ProcessFurnishChargingInformation	rw	CS1\proces13.spd
	Procedure ProcessForwardConnectionReleased	rw	CS1\processf.spd
	Procedure ProcessConnect	rw	CS1\proces10.spd
	Procedure ProcessInitiateCallAttempt	rw	CS1\processi.spd
	Procedure ProcessInitialDP	rw	CS1\pridp.spd
	Procedure ProcessConnectToResource	rw	CS1\proces11.spd
	Procedure ProcessRequestNotificationChargingEvent	rw	CS1\proces16.spd
	Virtual Procedure ProcessDPSpecific	rw	CS1\ProcDPS.spd
	Procedure ConnnectAnalysis	rw	CS1\ERROR\conpa.spd
	Virtual Process Type OriginatingBCSM	rw	CS1\ocs1.spt
	Procedure PIC_O_Null	rw	CS1\pic_o_nu.spd
	Procedure PIC_Analyse_Information	rw	CS1\pic_anal.spd
	Virtual Procedure PIC_Send_Call	rw	CS1\pic_send.spd
	Virtual Procedure PIC_O_Active	rw	CS1\pic_o_ac.spd
	Procedure PIC_Authorise_Origination_Attempt	rw	CS1\pic_auth.spd
	Procedure PIC_Select_Route	rw	CS1\pic_sele.spd
	Virtual Procedure PIC_O_Alerting	rw	CS1\pic_o_al.spd
	Procedure PIC_O_Abandon	rw	CS1\pic_o_ab.spd
	Procedure PIC_Collect_Information	rw	CS1\pic_coll.spd
	Procedure PIC_Authorise_Call_Setup	rw	CS1\pic_aut1.spd
	Procedure PIC_O_Answer	rw	CS1\pic_o_an.spd
	Procedure PIC_OException	rw	CS1\pic_oexc.spd
	Procedure PIC_Collect_NDigits	rw	CS1\pic_coll.spd

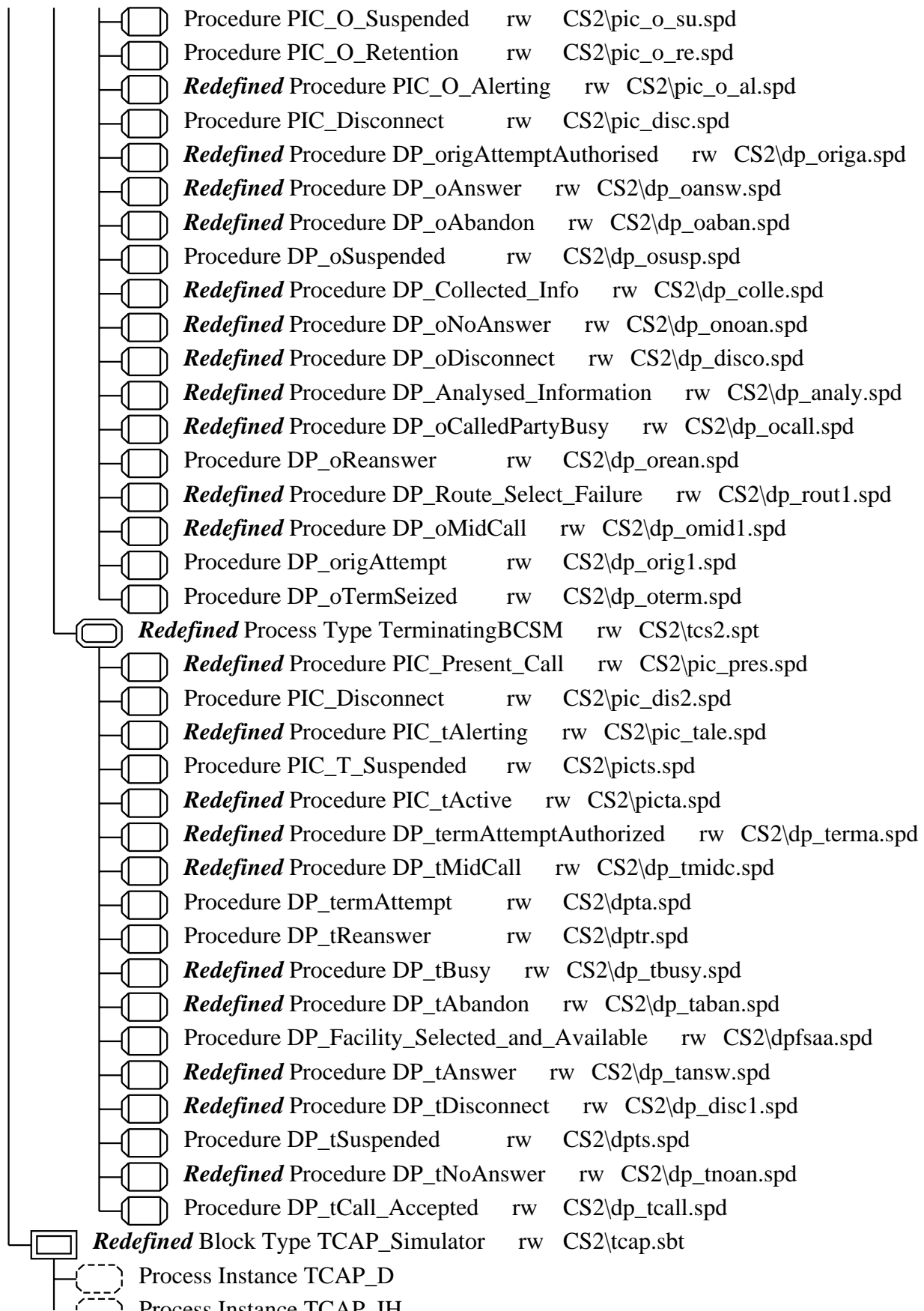


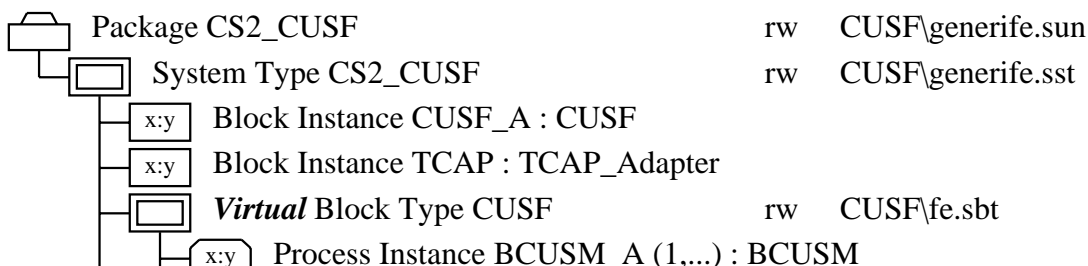
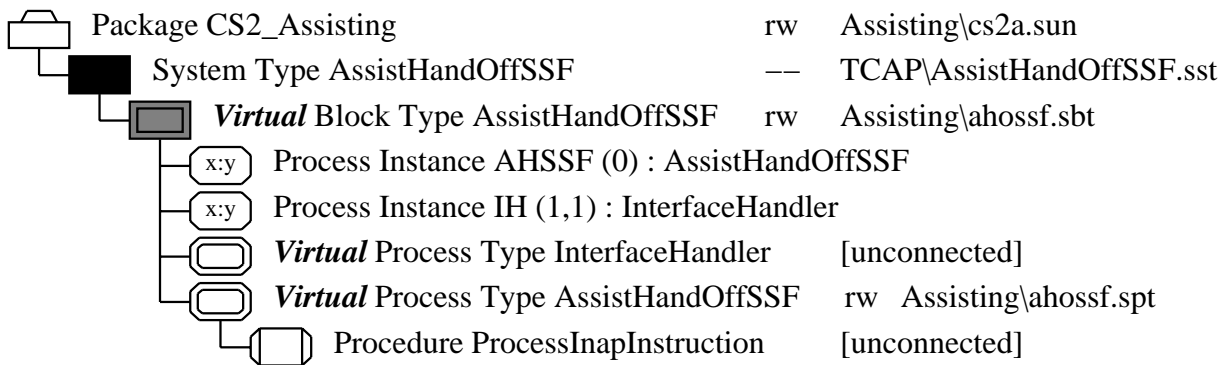
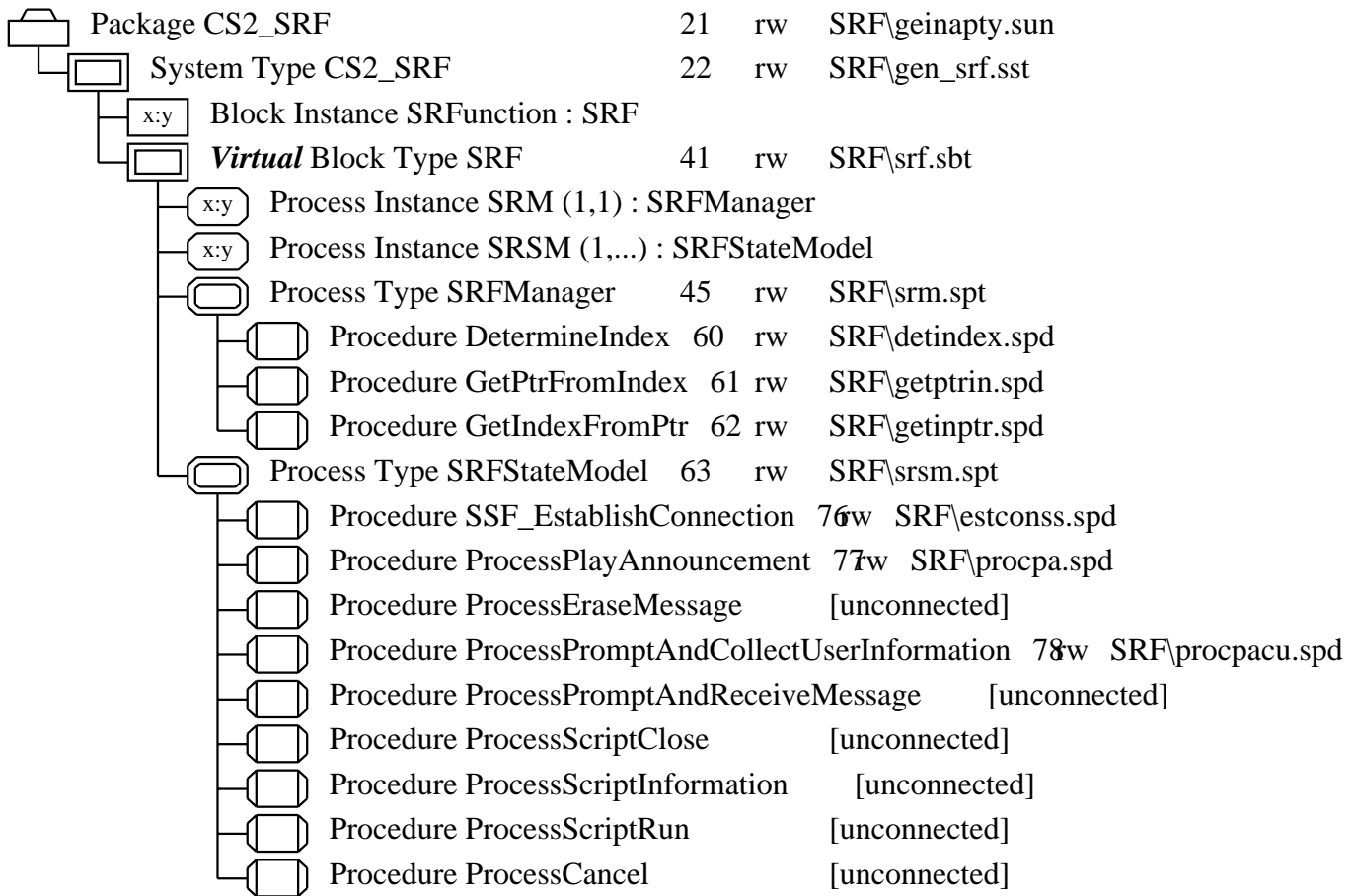
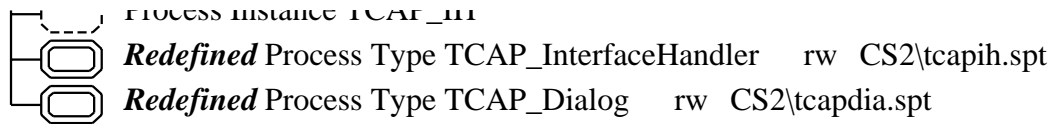


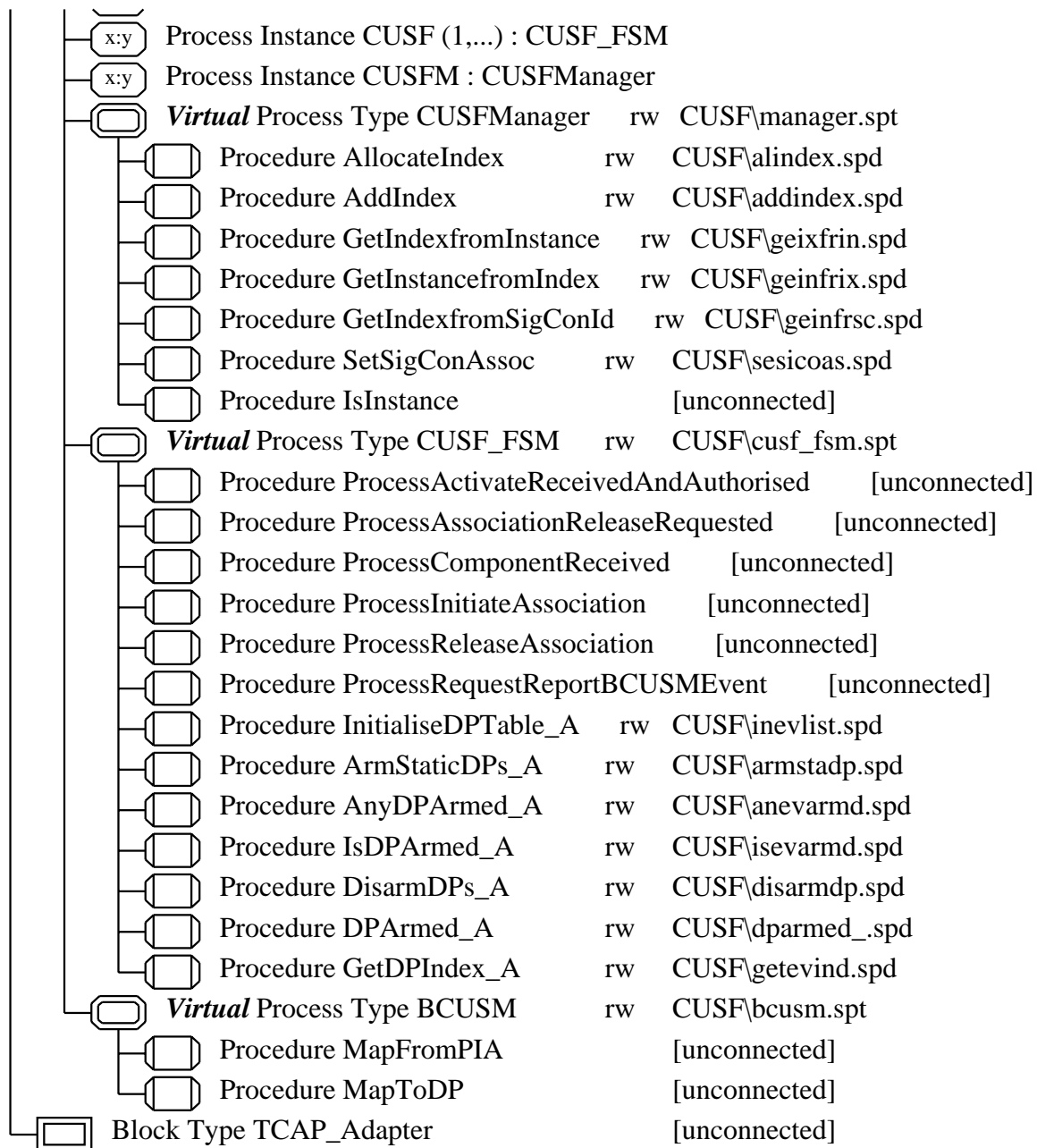
— Chapter IN CS-2 Specification



	Procedure MoveLegs	rw	CS2\movelegs.spd
	Procedure ExportControllingLeg	rw	CS2\expcl.spd
	Procedure ProcessRRBE	rw	CS2\processr.spd
	Procedure DisconnectControllingLeg	rw	CS2\dlc.spd
	Procedure ProcessQueuedRRBE	rw	CS2\processq.spd
	Procedure ReleaseCall	rw	CS2\release1.spd
	Redefined Process Type CallSegment	rw	CS2\cv2.spt
	Procedure ImportLeg	rw	CS2\importle.spd
	Procedure BroadcastToLegs	rw	CS2\broadcas.spd
	Procedure ExportLeg	rw	CS2\exportle.spd
	Procedure NoOfLegs	rw	CS2\nooflegs.spd
	Procedure DPFiltering	rw	CS2\dpfil.spd
	Procedure MapSFTToBCSM	rw	CS2\msftbcsm.spd
	Procedure DPFilteringUTSI	rw	CS2\dpfilter.spd
	Procedure MapAIToBCSM	rw	CS2\maitbcsm.spd
	Procedure MapSRToBCSM	rw	CS2\msrtbcsm.spd
	Redefined Process Type SSF_FSM	rw	CS2\ssf_fsm2.spt
	Procedure ExportEventRecord	rw	CS2\expevl.spd
	Procedure ProcessContinueWithArgument	rw	CS2\pcwa.spd
	Procedure ProcessRequestReportUTSI	rw	CS2\process3.spd
	Procedure ImportEventRecord	rw	CS2\impevl.spd
	Procedure ProcessDisconnectLeg	rw	CS2\processd.spd
	Procedure ProcessSendSTUI	rw	CS2\process4.spd
	Procedure ProcessReportUTSI	rw	CS2\process5.spd
	Redefined Procedure ProcessDPSpecific	rw	CS2\ProcDPS.spd
	Procedure IsUTSIArmed	rw	CS2\isutsiar.spd
	Procedure ProcessReportFacility	rw	CS2\prf.spd
	Procedure ProcessRequestReportFacilityEvent	rw	CS2\prrfe.spd
	Procedure IsFacilityArmed	rw	CS2\isfa.spd
	Procedure ProcessSendFacility	rw	CS2\psf.spd
	Redefined Procedure InitialiseDPTable	rw	CS2\initiali.spd
	Redefined Process Type SSME_FSM	rw	CS2\ssmefsm.spt
	Procedure ProcessManageTriggerData	rw	CS2\procmtd.spd
	Redefined Procedure CheckACG	rw	CS2\checkac1.spd
	Redefined Procedure InitialiseTDPTTable	rw	CS2\inittdp2.spd
	Redefined Process Type OriginatingBCSM	rw	CS2\ocs2.spt
	Redefined Procedure PIC_O_Active	rw	CS2\pic_o_ac.spd
	Redefined Procedure PIC_Send_Call	rw	CS2\pic_send.spd







Chapter System Instantiations

	System CS1_INAP	rw	CS1\cs1.ssy
	System CS2_INAP	rw	CS2\cs2_inap.ssy

Chapter CS-1 Examples

	MSC CI_Overview	rw	EXAMPLES\cirov.msc
	MSC CI_Detailed	rw	EXAMPLES\cid.msc



MSC CO_Overview

rw EXAMPLES\coov.msc



MSC CO_Detailed

rw EXAMPLES\cod.msc

—— Chapter CPH Examples



MSC 3Pty_Overview

rw EXAMPLES\3ptyov.msc



MSC 3Pty_Detailed

rw EXAMPLES\3ptyd.msc



MSC CF_Overview

rw EXAMPLES\cfov.msc



MSC CF_Detailed

rw EXAMPLES\cfd.msc

—— Chapter CS–1 Test Purposes

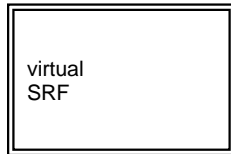


Package CS2_SRF

1(1)



system
CS2_SRF



System Type CS2_SRF

2(11)



↑ [CS2_INAP_Prim_To_SCF]

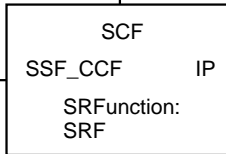
SCF_Channel

↓ [CS2_INAP_Prim_From_SCF]

SSF_CCF_Channel

← [srfTossf]

→ [ssfTosrf]



IP_Channel

← [SigCon_In]

→ [SigCon_Out]



```
/* Maximum Number of call segments supported */
SYNONYM MaxNumOfCSs NATURAL = 10;

SYNTYPE CalledPartyNumberType = Integer
ENDSYNTYPE;

SYNTYPE CallingPartyNumberType = Integer
ENDSYNTYPE;

SYNTYPE CalledPartyNumberType
= IPRoutingAddressType
ENDSYNTYPE;

SYNTYPE CorrelationType = Integer/*DigitsType*/
ENDSYNTYPE;

SYNTYPE ScfIDType = Integer
ENDSYNTYPE;

SYNTYPE CSIdType = NATURAL
DEFAULT 1
CONSTANTS 1:MaxNumOfCSs
ENDSYNTYPE;
```

```
NEWTYPE CallFlagType
LITERALS
  FromAddress, ToAddress;
ENDNEWTYPE;

NEWTYPE CallRefType STRUCT
  callFlag callFlagType;
  callId CallIdType;
ENDNEWTYPE;
```

```
NEWTYPE TableType
  ARRAY(IndexType,RecordType)
ENDNEWTYPE;

SYNTYPE IndexType = NATURAL
DEFAULT 1
CONSTANTS 1:MaxNumRecords
ENDSYNTYPE;

/* Maximum Number of Records supported */
SYNONYM MaxNumOfRecords NATURAL = 20;

NEWTYPE RecordType STRUCT
  used Boolean;
  instance PId;
ENDNEWTYPE;

SYNTYPE TransactionIdType,CallIdType,
  InvokeldType, LinkedIdType =IndexType
ENDSYNTYPE;
```




```
/* Call Progress Primitive Definition */
NEWTYPE CallProgressType
STRUCT
  callRef CallRefType;
  CS1ProgressIndicator CS1ProgressIndicatorType;
ENDNEWTYPE;
```

```
/* Setup Primitive Definition*/
NEWTYPE Ssf_SetupType
STRUCT
  callRef CallRefType;
  callingPartyNumber CallingPartyNumberType;
  calledPartyNumber CalledPartyNumberType;
  ipRoutingAddress IPRoutingAddressType;
  scfID ScfIDType;
  correlationID CorrelationIDType;
ENDNEWTYPE;
/*
These are the types taken from the EstablishTemporary Connection
and the ConnectToResource operations.
*/
```

```
NEWTYPE Srf_SetupType
STRUCT
  callRef CallRefType;
  callingPartyNumber CallingPartyNumberType;
  calledPartyNumber CalledPartyNumberType;
ENDNEWTYPE;
```

```
NEWTYPE SubsequentAddressType
STRUCT
  callRef CallRefType;
  iA5Information CharString;
ENDNEWTYPE;
```

```
/* Data Primitive Definition */
NEWTYPE DataType
STRUCT
  callRef CallRefType;
  serviceIndicator ServiceIndicatorType;
  data CharString ;/* OctetString*/
ENDNEWTYPE;
```

```
/* Facility Primitive Definition */
NEWTYPE FacilityType
STRUCT
  callRef CallRefType;
  invokeld InvokeldType;
  linkedId LinkedIdType;
  operationCode OperationCodeType;
  parameter Charstring;
ENDNEWTYPE;
```

```
STRUCT FacilityResultType
  callRef CallRefType;
  result ResultType;
ENDNEWTYPE;
```

```
STRUCT ResultType
  operationCode OperationCodeType;
  parameter ParameterType;
ENDNEWTYPE;
```

```
STRUCT FacilityError
  invokeld InvokeldType;
  errorcode ErrorCodeType;
  parameter ParameterType;
ENDNEWTYPE;
```

```
/* Call Release Primitive Definition */
NEWTYPE ReleaseType
STRUCT
  callRef CallRefType;
  cause CauseType;
  location LocationType;
/*diagnostics DiagnosticsType;*/
ENDNEWTYPE;
```



SIGNAL

/*
It should be noted that a connection can only be set up from the SSF-CCF towards the SRF/IP.
*/

```
Ssf_Setup.req(Ssf_SetupType),
Srf_Setup.req(Srf_SetupType),
Ssf_CallProgress.ind(CallProgressType),
Srf_CallProgress.ind(CallProgressType),
Ssf_Facility.req(FacilityType),
Srf_Facility.req(FacilityType),
Ssf_Facility.ind(FacilityType),
Srf_Facility.ind(FacilityType),
Ssf_FacilityResult.req(FacilityType),
Srf_FacilityResult.req(FacilityType),
Ssf_FacilityResult.ind(FacilityType),
Srf_FacilityResult.ind(FacilityType),
Ssf_FacilityError.req(FacilityType),
Srf_FacilityError.req(FacilityType),
Ssf_FacilityError.ind(FacilityType),
Srf_FacilityError.ind(FacilityType),
Ssf_Data.req(DataType),
Srf_Data.req(DataType),
Ssf_Data.ind(DataType),
Srf_Data.ind(DataType),
Ssf_Setup.conf(Ssf_SetupType),
Srf_Setup.conf(Srf_SetupType),
Ssf_Release.req(ReleaseType),
Srf_Release.req(ReleaseType),
Ssf_Release.ind(ReleaseType),
Srf_Release.ind(ReleaseType),
Ssf_SubsequentAddress.req(SubsequentAddressType);
```

/* Additional Signals required when the connection is established from the IP, this is not the normal case. */

```
SIGNAL
Srf_Setup.ind(Srf_SetupType),
Ssf_Setup.ind(Ssf_SetupType),
Ssf_Setup.resp(Ssf_SetupType),
Ssf_Setup.resp(Ssf_SetupType);
```

NEWTYPE OperationCode

LITERALS

```
eraseMessage,
playAnnouncement,
assistRequestInstructions,
promptAndCollectUserInfo,
promptAndReceiveMessage,
scriptClose,
scriptEvent,
scriptInformation,
scriptRun,
specialisedResourceReport,
cancel,
assistRequestInstructions;
ENDNEWTYPE;
```

/* Signals from SRF to SSF-CCF*/

```
SIGNALLIST srfTossf=
Ssf_CallProgress.ind,
Ssf_Facility.ind,
Ssf_FacilityResult.ind,
Ssf_FacilityError.ind,
Ssf_Data.ind,
Ssf_Setup.conf,
Ssf_Release.ind,
/* Additional signal for IP setup*/
Ssf_Setup.ind;
```

/* Signals from SSF-CCF to SRF */

```
SIGNALLIST ssfToSrf=
Ssf_Setup.req,
Ssf_Facility.req,
Ssf_FacilityResult.req,
Ssf_FacilityError.req,
Ssf_Data.req,
Ssf_Release.req,
Ssf_SubsequentAddress.req,
/* Additional signal for IP setup*/
Ssf_Setup.resp;
```

/* Signals from SRF to IP*/

```
SIGNALLIST SigCon_Out=
Srf_Setup.req,
Srf_Facility.req,
Srf_FacilityResult.req,
Srf_FacilityError.req,
Srf_Data.req,
Srf_Release.req,
/* Additional signal for IP setup*/
Srf_Setup.resp;
```

/* Signals from IP to SRF*/

```
SIGNALLIST SigCon_In=
Srf_CallProgress.ind,
Srf_Facility.ind,
Srf_FacilityResult.ind,
Srf_FacilityError.ind,
Srf_Data.ind,
Srf_Setup.conf,
Srf_Release.ind,
/* Additional signal for IP setup*/
Srf_Setup.ind;
```



NEWTYPE CauseTy
LITERALS
unallocatedNumbe
noRouteToSpecifie
noRouteToDestin

/*Cause definitions
1. Normal class
Cause No: 1 unalloc
This cause indicates
valid format, it is

System Type CS2_SRF

NEWTYPE CauseType

LITERALS

unallocatedNumber,
noRouteToSpecifiedTransitNetwork,
noRouteToDestination,
sendSpecialInformationTone,
misdialledTrunkPrefix,
channelUnacceptable,
callAwardedAndBeingDeliveredInAnEstablishedChannel,
preemption,
preemptionCircuitReservedForReuse,
normalCallClearing,
userBusy,
noUserResponding,
noAnswerFromUser,
subscriberAbsent,
callRejected,
numberChanged,
reRouteToNewDestination,
nonSelectedUserClearing,
destinationOutOfOrder,
invalidNumberFormat,
facilityRejected,
responseToStatusEnquiry,
normalUnspecified,
noCircuitChannelAvailable,
networkOutOfOrder,
permanentFrameModeConnectionOutOfService,
permanentFrameModeConnectionOperational,
temporaryFailure,
switchingEquipmentCongestion,
accessInformationDiscarded,
requestedCircuitChannelNotAvailable,
precedenceCallBlocked,
resourceUnavailableUnspecified,
qualityOfServiceUnavailable,
requestedFacilityNotSubscribed,
outgoingCallsBarredWithinCUG,
incomingCallsBarredWithinCUG,
bearerCapabilityNotAuthorised,
bearerCapabilityNotPresentlyAvailable,
inconsistencyInDesignatedOutgoingAccessInformationAndSubscriberClass,
serviceOrOptionNotAvailableUnspecified,
bearerCapabilityNotImplemented,
channelTypeNotImplemented,
requestedFacilityNotImplemented,
onlyRestrictedDigitalInformationBearerCapabilityIsAvailable,
serviceOrOptionNotImplementedUnspecified,
invalidCallReferenceValue,
identifiedChannelDoesNotExist,
aSuspendedCallExistButThisCallIdentityDoesNot,
callIdentityInUse,
noCallSuspended,
callHavingTheRequestCallIdentityHasBeenCleared,
userNotMemberOfCUG,
incompatibleDestination,
nonExistCUG,
invalidTransitNetworkSelection,
invalidMessageUnspecified,
mandatoryInformationElementIsMissing,
messageTypeNonExistentOrNotImplemented,
messageNotCompatibleWithCallStateOrMessageTypeNonExistentOrNotImplemented,

informationElementParameterNonExistentOrNotImplemented,
invalidInformationElementContents,
messageNotCompatibleWithCallState,
recoveryOnTimerExpiry,
parameterNonExistentOrNotImplementedPassedOn,
messageWithUnrecognisedParameterDiscarded,
protocolErrorUnspecified,
interworkingUnspecified;
ENDNEWTYPE;

NEWTYPE LocationType
LITERALS
notPresent,
user,
privateNetworkServingTheLocalUser,
publicNetworkServingTheLocalUser,
transitNetwork,
internationalTransitNetwork,
privateNetworkServingTheRemoteUser,
publicNetworkServingTheRemoteUser;
DEFAULT notPresent
ENDNEWTYPE;

System Type CS2_SRF

/*Cause definitions

1. Normal class

Cause No: 1 unallocatedNumber

This cause indicates that the called party cannot be reached because, although the called party number is in a valid format, it is not currently allocated (assigned).

Cause No: 2 noRouteToSpecifiedTransitNetwork

This cause indicates that the equipment sending this cause has received a request to route the call through a particular transit network which it does not recognise. The equipment sending this cause does not recognise the transit network either because the transit network does not exist or because that particular transit network, while it does exist, does not serve the equipment which is sending this cause. This cause is supported on a network-dependent basis.

Cause No: 3 noRouteToDestination

This cause indicates that the called party cannot be reached because the network through which the call has been routed does not serve destination desired.

This cause is supported on a network-dependent basis.

Cause No: 4 sendSpecialInformationTone

This cause indicates that the called party cannot be reached for reasons that are of a long term nature and that the special information tone should be returned to the calling party.

Cause No: 5 misdialledTrunk prefix

This cause indicates the erroneous inclusion of a trunk prefix in the called party number.

Cause No: channelUnacceptable

This cause indicates that the channel most recently identified is not acceptable to the sending entity for use in this call.

Cause No: 7 callAwardedAndBeingDeliveredInAnEstablishedChannel

This cause indicates that the user has been awarded the incoming call, and that the incoming call is being connected to a channel already established to that user for similar calls (e.g., packet-mode X.25 virtual calls).

Cause No: 8 preemption

This cause indicates that the call is being preempted.

Cause No: 9 preemptionCircuitReservedForReuse

This cause indicates that the call is being preempted and that the circuit is reserved for reuse by the preempting exchange.

Cause No: 16 normalCallClearing

This cause indicates that the call is being cleared because one of the users involved in the call has requested that the call be cleared.

Under normal situations, the source of this cause is not the network.

Cause No: 17 userBusy

This cause is used to indicate that called party is unable to accept another call because the user busy condition has been encountered. This cause may be generated by called user or by the network. In the case of user determined user busy it is noted that the user equipment is compatible with the call.

Cause No: 18 noUserResponding

This cause is used when the called party does not respond to a call establishment message with either an alerting or connect indication within the prescribed period of time allocated.

Cause No: 19 noAnswerFromUser

This cause is used when the called party has been alerted but does not respond with a connect indication.

within a prescribed period of time.

Note – This cause is not necessarily generated by Q.931 procedures but may be generated by internal network timers.

Cause No: 20 subscriberAbsent

This cause value is used when a mobile station has logged off, radio contact is not obtained with a mobile station or if a personal telecommunication user is temporarily not addressable at any user network interface

Cause No: 21 calledRejected

This cause indicates that the equipment sending this cause does not wish to accept this call, although it could have accepted the call because the equipment sending this cause is either busy nor incompatible.

This cause may also be generated by the network, indicating that the call was cleared due to a supplementary service constraint. The diagnostic field may contain additional information about the supplementary service and reason rejected.

Cause No: 22 numberChanged

This cause is returned to a calling party when the called party number indicated by the calling party is no longer assigned. The new called party number may optionally be included in the diagnostic field. If a network does not support this cause value, cause no: 1 unallocated number, shall be used.

Cause No: 23 reRouteToNewDestination

This cause is used by a general ISUP protocol mechanism that can be invoked by an exchange that decides

that the call should be set-up to a different called number. Such an exchange can invoke a re route mechanism, by use of this of this cause value, to request a preceding exchange involved in the call to route the call to be new number.

Cause No: 26 nonSelectedUserClearing

This cause indicates that the user has not been awarded the incoming call.

Cause No: 27 destinationOutOfOrder

This cause indicates that the destination indicated by the user cannot be reached because the interface to the destination is not functioning correctly. The term not functioning correctly indicates that a signalling message was unable to be delivered to the remote party, e.g. a physical layer or data link layer failure at the remote party, or user equipment off-line.

Cause No: 28 invalidNumberFormat

This cause indicates that the called party cannot be reached because the called party number is not a valid format or is not complete.

Note – This condition may be determined:

- immediately after reception of an ST signal, or
- on time-out after the last received digit.

Cause No: 29 facilityRejected

This cause is returned when a supplementary service requested by the user cannot be provided by the network.

Cause No: 30 responseToStatusReport

This cause is included in the STATUS message when the reason for generating the STATUS message was the prior receipt of a STATUS ENQUIRY message.

Cause No: 31 normalUnspecified

This cause used to report a normal event only when no other cause in the normal class applies.

Cause No: 34 noCircuitChannelAvailable

This cause indicates that there is no appropriate circuit / channel presently available to handle the call.

Cause No: 38 networkOutOfOrder

This cause indicates that the network is not functioning correctly and that the condition is likely to last a relatively long period of time, e.g. immediately re-attempting the call is not likely to be successful.

Cause No: 39 permanentFrameModeConnectionOut-Of-Service

This cause is included in a STATUS message to indicate that a permanently established frame mode connection

is out-of-service (e.g. due to go equipment or section failure) (see Annex A/Q.933).

Cause No: 40 permanentFrameModeConnectionOperational

This cause is included in a STATUS message to indicate that a permanently established frame mode connection

is operational and capable of carrying user information (see Annex A/Q.933).

Cause No: 41 temporaryFailure

This cause indicates that the network is not functioning correctly and that the condition is not likely to last a long period of time, e.g. the user may wish to try another call attempt almost immediately.

Cause No: 42 switchingEquipmentCongestion

This cause indicates that the switching equipment generating this cause is experiencing a period of high traffic.

Cause No: 43 accessInformationDiscarded

This cause indicates that the network could not deliver access information to the remote user as requested, i.e.

user-to-user information, low layer compatibility, high layer compatibility or sub address, as indicated in the diagnostic.

It is noted that the particular type of access information discarded is optionally included in the diagnostic.

Cause No: 44 requestedCircuit/ChannelNotAvailable

The cause is returned when the circuit or channel indicated by the requesting entity cannot be provided by the other side of the interface.

Cause No: 46 precedenceCallBlocked

This cause indicates that there are no preemptable circuits or that the called user is busy with a call of equal or higher preemptable level.

Cause No: 47 resourceUnavailable,Unspecified

This cause is used to report a resource unavailable event only when no other cause in the resource unavailable class applies.

Cause No: 49 qualityOfServiceNotAvailable

This cause is used to report that the requested Quality of Service, as defined in Recommendation X.213, cannot be provided (e.g. throughput or transit delay cannot be supported).

Cause No: 50 requestedFacilityNotSubscribed

This cause indicates that the user has requested a supplementary service which is implemented by the equipment which generated this cause, but the user is not authorised to use.

Cause No: 53 outgoingCallsBarredWithinCUG

This cause indicates that although the calling party is a member of the CUG for the outgoing CUG call, outgoing calls are not allowed for this member of the CUG.

Cause No: 55 incomingCallsBarredWithinCUG

This cause indicates that although the called party is a member of the CUG for the incoming CUG call, incoming calls are not allowed to this member of the CUG.

Cause No: 57BearerCapabilityNotAuthorised

This cause indicates that the user has requested a bearer capability which is implemented by the equipment which generated this cause but the user is not authorised to use.

Cause No: 58 bearerCapabilityNotPresentlyAvailable

This cause indicates that the user has requested a bearer capability which is implemented by the equipment which generated this cause but which is not available at this time.

Cause No: 62 inconsistencyInDesignatedOutgoingAccessInformationAndSubscriberClasses

This cause indicates that there is an inconsistency in the designated outgoing access information and subscriber class.

Cause No: 63 serviceOrOptionNotAvailableUnspecified

This cause is used to report a service or option not available event only when no other cause in the service or option not available class applies.

Cause No: 65 bearerCapabilityNotImplemented

This cause indicates that the equipment sending this cause does not support the bearer capability requested.

Cause No: 66 channelTypeNotImplemented

This cause indicates that the equipment sending this cause does not support the channel type requested.

Cause No: 69 requestedFacilityNotImplemented

This cause indicates that the equipment sending this cause does not support the requested supplementary service.

Cause No: 70 only restricted digital information bearer capability is available

This cause indicates that the calling party has requested an unrestricted bearer service but that the equipment sending this cause only supports the restricted version of the requested bearer capability.

Cause No: 79 serviceOrOptionNotImplemented,Unspecified

This cause is used to report a service or option not implemented event only when no other cause in the service or option not implemented class applies.

Cause No: 81 invalidCallReferenceValue

This cause indicates that the equipment sending this cause has received a message with a call reference which is not currently in use on the user-network interface.

Cause No: 82 identifiedChannelDoesNotExist

This cause indicates that the equipment sending this cause has received a request to use a channel not activated on the interface for a call. For example, if a user has subscribed to those channels on a primary rate interface numbered from 1 to 12 and the user equipment or the network attempts to use channels 13 through 23, this cause is generated.

Cause No: 83 aSuspendedCallExists,ButThisCallIdentityDoesNot

This cause indicates that a call resume has been attempted with a call identity which differs from that in use for any presently suspended call(s).

Cause No: 84 callIdentityInUse

This cause indicates that the network has received a call suspended request containing a call identity (including the null call identity) which is already in use for a suspended call within the domain of interfaces over which the call might be resumed.

Cause No: 85 noCallSuspended

This cause indicates that the network has received a call resume request containing a Call identity information element which presently does not indicate any suspended call within the domain of interfaces over which calls may be resumed.

Cause No: 86 callHavingTheRequestedCallIdentityHasBeenCleared

This cause indicates that the network has received a call resume request containing a Call identity information element indicating a suspended call that has in the meantime been cleared while suspended (either by network timeout or by the remote user).

Cause No: 87 callHavingTheRequestedCallIdentityHasBeenCleared

This cause indicates that the called user for the incoming CUG call is not a member of the specified CUG or that the calling user is an ordinary subscriber calling a CUG subscriber.

Cause No: 88 incompatibleDestination

This cause indicates that the equipment sending this cause has received a request to establish a call which has low layer compatibility, high layer compatibility or other compatibility attributes (e.g. data rate) which cannot be accommodated.

Cause No: 90 nonExistentCUG

This cause indicates that the specified CUG does not exist.

Cause No: 91 invalidTransitNetworkSelection

This cause indicates that a transit network identification was received which is of an incorrect format as defined in Annex C/Q.931.

Cause No: 95 invalidMessageUnspecified

This cause is used to report an invalid message event only when no other cause in the valid message class applies.

Cause No: 96 mandatoryInformationElementMissing

This cause indicates that the equipment sending this cause has received a message which is missing an information element which must be present in the message before that message can be processed.

Cause No: 97 messageTypeNonExistentOrNotImplemented

This cause indicates that the equipment sending this cause has received a message with a message type it does not recognise either because this is a message not defined or defined but not implemented by the equipment sending this cause.

Cause No: 98 messageNotCompatibleWithCallStateOrMessageTypeNonExistentOrNotImplemented

This cause indicates that the equipment sending this cause has received a message such that the procedures do not indicate that this is a permissible message to receive while in the call state, or a STATUS message was received indicating an incompatible call state.

Cause No: 99 informationElementParameterNonExistentOrNotImplemented

This cause indicates that the equipment sending this cause has received a message which includes information element(s) / parameter(s) not recognised because the information element identifier(s) / parameter name(s) are not defined or are defined but not implemented by the equipment sending this cause.

This cause indicates that the information element(s) / parameter(s) were discarded. However, the information element is not required to be present in the message in order for the equipment sending the cause to process the message.

Cause No: 100 invalidInformationElementContents

This cause indicates that the equipment sending this cause has received an information element which it has implemented, however, one or more fields in the information element are coded in such a way which has not been implemented by the equipment sending this cause.

Cause No: 101 messageNotCompatibleWithCallState

This cause indicates that a message has been received which is incompatible with the call state.

Cause No: 102 recoveryOrTimerExpiry

This cause indicates that a procedure has been initiated by the expiry of a timer in association with error handling procedures.

Cause No: 103 ParameterNonExistentOrNotImplementedPassedOn

This cause indicates that the equipment sending this cause has received a message which includes parameters not recognised because the parameters are not defined or are defined but not implemented by sending the cause.

This cause indicates that the parameter(s) were ignored. In addition, if the equipment sending this cause is an intermediate point, then this cause indicates that the parameter(s) were passed

on unchanged.

Cause No: 110 messageWithUnrecognisedParameterDiscarded

This indicates that the equipment sending this cause has discarded a received message which includes a parameter that is not recognised.

Cause No: 111 protocolErrorUnspecified

This cause is used to report a protocol error event only when no other cause in the protocol error class applies.

Cause No: 127 interworkingUnspecified

This cause indicates that there has been interworking with a network which does not provide causes for actions it takes. Thus, the precise cause for a message which is being sent cannot be ascertained.

*/



```
NEWTYPE SrfErrorCodeType
LITERALS
  canceled,
  improperCallerResponse,
  parameterOutOfRange,
  systemfailure,
  taskRefused,
  unexpectedComponentSequence,
  unavailableResource,
  unexpectedDataValue,
  unexpectedParameter;
ENDNEWTYPE;
/*
SYNTYPE DigitsType = Octet_String
CONSTANTS SIZE (minDigitsLength : maxDigitsLength)
ENDSYNTYPE;
/*
```



/* PlayAnnouncement INAP Definitions*/

```
NEWTYPE PlayAnnouncementType
STRUCT
transactionId TransactionIdType;
invokeld Invokeldtype;
cSID CSIdType;
disconnectFromIPForbidden Boolean;
informationToSend InformationToSendType;
requestAnnouncementComplete Boolean;
ENDNEWTYPE;
```

```
NEWTYPE InformationToSendType
LITERALS
messageld,
tone,
displayInformation;
ENDNEWTYPE;
```

/* SpecialResourceReport INAP Definitions*/

```
NEWTYPE SpecialResourceReportType
STRUCT
transactionId TransactionIdType;
invokeld Invokeldtype;
cSID CSIdType;
ENDNEWTYPE;
```

/* AssistRequestInstructions INAP Definitions*/

```
NEWTYPE AssistRequestInstructionsType
STRUCT
transactionId TransactionIdType;
invokeld Invokeldtype;
correlationId CorrelationIdType;
/* ipAvailable IpAvailableType=Octet_String*/;
ENDNEWTYPE;
```

/* PlayAnnouncementError INAP Definitions*/

```
NEWTYPE PlayAnnouncementErrorType
STRUCT
transactionId TransactionIdType;
invokeld InvokeldType;
errorCode ErrorCodeType;
ENDNEWTYPE;
```

/* EraseMessage INAP Definitions*/

```
NEWTYPE EraseMessageType
STRUCT
transactionId TransactionIdType;
invokeld Invokeldtype;
cSID CSIdType;
disconnectFromIPForbidden DisconnectFromIPForbiddenType;
/* recodedMessageIDsToErase RecodedMessageIDsToEraseType*/;
ENDNEWTYPE;
```



```
/* PromptAndCollectUserInfoError INAP Definitions*/  
NEWTYPE PromptAndCollectUserInfoErrorType  
STRUCT  
    transactionId TransactionIdType;  
    invokeId InvokeIdType;  
    errorCode ErrorCodeType;  
ENDNEWTYPE;
```

```
NEWTYPE CollectedInfoType STRUCT  
/*collectedDigits CollecteddigitsType*/  
iA5Information Boolean;  
ENDNEWTYPE;
```

```
/* PromptAndCollectUserInfo INAP Definitions*/  
NEWTYPE PromptAndCollectUserInfoType  
STRUCT  
    transactionId TransactionIdType;  
    invokeId InvokeIdType;  
    cSId CSIdType;  
    collectedInfo CollectedInfoType;  
    disconnectFromIPForbidden Boolean;  
/* informationToSend InformationToSendType*/;  
ENDNEWTYPE;
```

```
/* PromptAndReceiveMessageResult INAP Definitions*/  
NEWTYPE PromptAndReceiveMessageResultType  
STRUCT  
    transactionId TransactionIdType;  
    invokeId InvokeIdType;  
/*receivedStatus receivedStatusType*/  
recordedMessageID Charstring;  
ENDNEWTYPE;
```

```
/* PromptAndCollectUserInfo Result INAP Definitions*/  
NEWTYPE PromptAndCollectUserInfoResultType  
STRUCT  
    transactionId TransactionIdType;  
    invokeId InvokeIdType;  
/*digitsResponse DigitsType*/  
iA5Response CharString;  
ENDNEWTYPE;
```

```
/* PromptAndReceiveMessage INAP Definitions*/  
NEWTYPE PromptAndReceiveMessageType  
STRUCT  
    transactionId TransactionIdType;  
    invokeId InvokeIdType;  
    cSID CSIdType;  
    disconnectFromIPForbidden Boolean;  
/* informationToSend InformationToSendType*/;  
ENDNEWTYPE;
```

```
/* PromptAndReceiveMessageError INAP Definitions*/  
NEWTYPE PromptAndReceiveMessageErrorType  
STRUCT  
    transactionId TransactionIdType;  
    invokeId InvokeIdType;  
    errorCode ErrorCodeType;  
ENDNEWTYPE;
```



```
/* ScriptClose INAP Definitions*/  
NEWTYPE ScriptCloseType  
STRUCT  
  transactionId TransactionIdType;  
  invokeId InvokeIdType;  
  cSID CSIdType;  
  /*ulScriptId UIScriptIdType;  
  ulScriptSpecificInfo UIScriptSpecificInfoType;*/  
ENDNEWTYPE;
```

```
/* ScriptCloseError INAP Definitions*/  
NEWTYPE ScriptCloseErrorType  
STRUCT  
  transactionId TransactionIdType;  
  invokeId InvokeIdType;  
  errorCode ErrorCodeType;  
ENDNEWTYPE;
```

```
/* ScriptInformation INAP Definitions*/  
NEWTYPE ScriptInformationType  
STRUCT  
  transactionId TransactionIdType;  
  invokeId InvokeIdType;  
  cSID CSIdType;  
  /*ulScriptId UIScriptIdType;  
  ulScriptSpecificInfo UIScriptSpecificInfoType;*/  
ENDNEWTYPE;
```

```
/* ScriptInformationError INAP Definitions*/  
NEWTYPE ScriptInformationErrorType  
STRUCT  
  transactionId TransactionIdType;  
  invokeId InvokeIdType;  
  errorCode ErrorCodeType;  
ENDNEWTYPE;
```

```
/* ScriptRun INAP Definitions*/  
NEWTYPE ScriptRunType  
STRUCT  
  transactionId TransactionIdType;  
  invokeId InvokeIdType;  
  cSID CSIdType;  
  /*ulScriptId UIScriptIdType;  
  ulScriptSpecificInfo UIScriptSpecificInfoType;*/  
ENDNEWTYPE;
```

```
/* ScriptRunError INAP Definitions*/  
NEWTYPE ScriptRunErrorType  
STRUCT  
  transactionId TransactionIdType;  
  invokeId InvokeIdType;  
  errorCode ErrorCodeType;  
ENDNEWTYPE;
```

```
/* ScriptEvent INAP Definitions*/  
NEWTYPE ScriptEventType  
STRUCT  
  transactionId TransactionIdType;  
  invokeId InvokeIdType;  
  cSID CSIdType;  
  /*ulScriptId UIScriptIdType;  
  ulScriptResult UIScriptResultType;*/  
ENDNEWTYPE;
```

```
/* Cancel INAP Definitions*/  
NEWTYPE CancelType  
STRUCT  
  transactionId TransactionIdType;  
  invokeId InvokeIdType;  
  cSID CSIdType;  
  cancelInvokeId InvokeIdType;  
  allRequests AllRequestsType;  
ENDNEWTYPE;
```



```

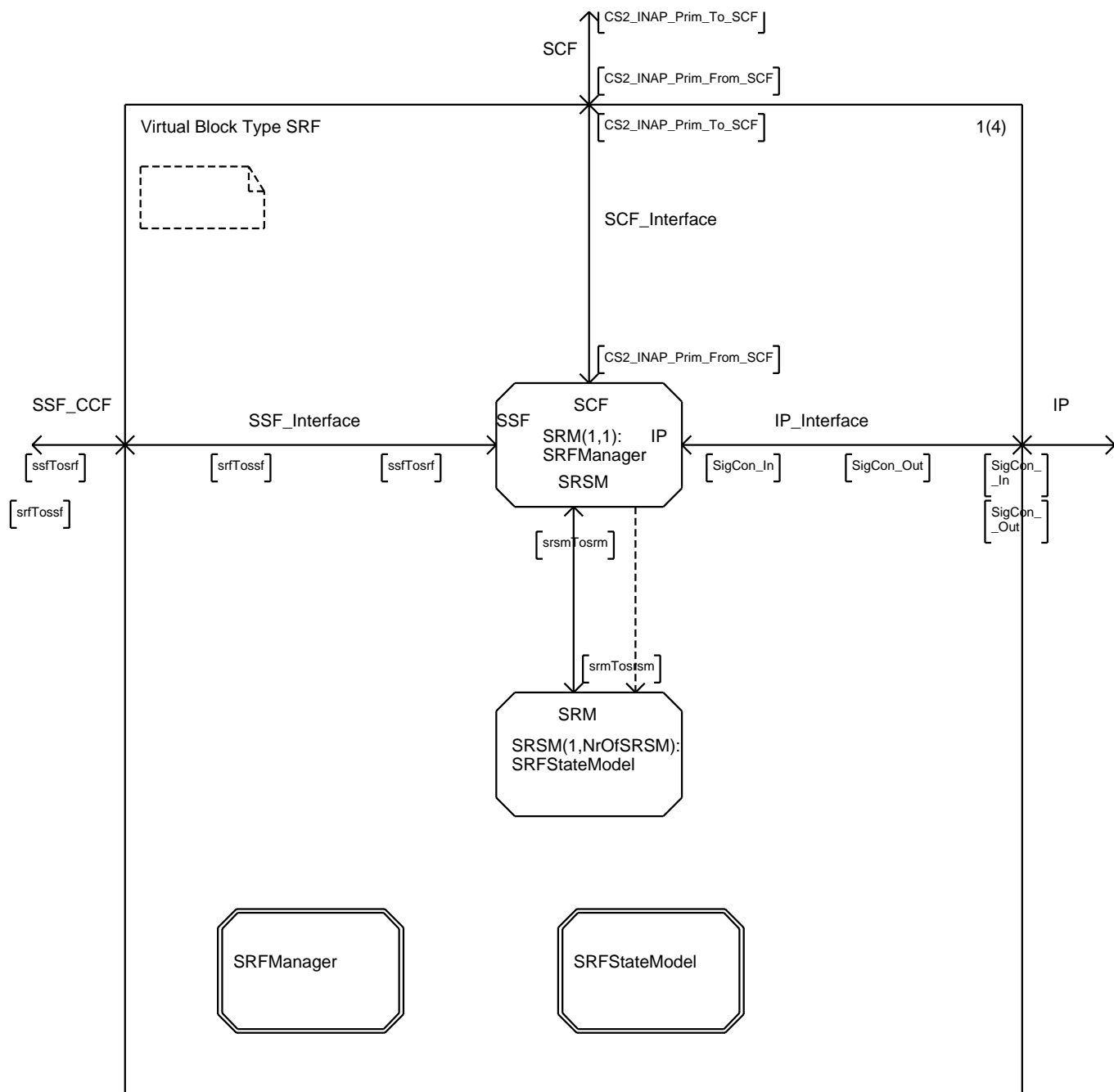
SIGNAL
EraseMessagePrim(EraseMessageType),
PlayAnnouncementPrim(PlayAnnouncementType),
PlayAnnouncementErrorPrim(PlayAnnouncementErrorType),
PromptAndCollectUserInformationPrim(PromptAndCollectUserInformationType),
PromptAndCollectUserInformationResultPrim(PromptAndCollectUserInformationResultType),
PromptAndCollectUserInformationErrorPrim(PromptAndCollectUserInformationErrorType),
PromptAndReceiveMessagePrim(PromptAndReceiveMessageType),
PromptAndReceiveMessageResultPrim(PromptAndReceiveMessageResultType),
PromptAndReceiveMessageErrorPrim(PromptAndReceiveMessageErrorType),
ScriptClosePrim(ScriptCloseType),
ScriptCloseErrorPrim(ScriptCloseErrorType),
ScriptEventPrim(ScriptEventType),
ScriptInformationPrim(ScriptInformationType),
ScriptInformationErrorPrim(ScriptInformationErrorType),
ScriptRunPrim(ScriptRunType),
ScriptRunErrorPrim(ScriptRunErrorType),
SpecialisedResourceReportPrim(SpecialisedResourceReportType),
CancelPrim(CancelType),
CancelErrorPrim(CancelErrorType),
AssistRequestInstructionsPrim(AssistRequestInstructionsType),
ApplicationBeginPrim(TransactionIdType),
ApplicationAbortPrim(TransactionIdType),
ApplicationEndPrim(TransactionIdType);

```

```

/*Signals from SRF to SCF*/
SIGNALLIST CS2_INAP_Prim_To_SCF=
PlayAnnouncementErrorPrim,
PromptAndCollectUserInformationResultPrim,
PromptAndCollectUserInformationErrorPrim,
PromptAndReceiveMessageResultPrim,
PromptAndReceiveMessageErrorPrim,
ScriptCloseErrorPrim,
ScriptEventPrim,
ScriptInformationErrorPrim,
ScriptRunErrorPrim,
SpecialisedResourceReportPrim,
CancelErrorPrim,
AssistRequestInstructionsPrim,
ApplicationBeginPrim,
ApplicationAbortPrim,
ApplicationEndPrim;
/*Signals from SCF to SRF*/
SIGNALLIST CS2_INAP_Prim_From_SCF=
EraseMessagePrim,
PlayAnnouncementPrim,
PromptAndCollectUserInformationPrim,
PromptAndReceiveMessagePrim,
ScriptClosePrim,
ScriptInformationPrim,
ScriptRunPrim,
CancelPrim,
ApplicationBeginPrim,
ApplicationAbortPrim,
ApplicationEndPrim;

```



SIGNAL

/*

It should be noted that a connection can only be set up from the SSF-CCF towards the SRF/IP.

*/

```
Ssf_Setup.req(Ssf_SetupType),
Srf_Setup.req(Srf_SetupType),
Ssf_CallProgress.ind(CallProgressType),
Srf_CallProgress.ind(CallProgressType),
Ssf_Facility.req(FacilityType),
Srf_Facility.req(FacilityType),
Ssf_Facility.ind(FacilityType),
Srf_Facility.ind(FacilityType),
Ssf_FacilityResult.req(FacilityType),
Srf_FacilityResult.req(FacilityType),
Ssf_FacilityResult.ind(FacilityType),
Srf_FacilityResult.ind(FacilityType),
Ssf_FacilityError.req(FacilityType),
Srf_FacilityError.req(FacilityType),
Ssf_FacilityError.ind(FacilityType),
Srf_FacilityError.ind(FacilityType),
Ssf_Data.req(DataType),
Srf_Data.req(DataType),
Ssf_Data.ind(DataType),
Srf_Data.ind(DataType),
Ssf_Setup.conf(Ssf_SetupType),
Srf_Setup.conf(Srf_SetupType),
Ssf_Release.req(ReleaseType),
Srf_Release.req(ReleaseType),
Ssf_Release.ind(ReleaseType),
Srf_Release.ind(ReleaseType),
Ssf_SubsequentAddress.req(SubsequentAddressType);
```

/* Signals from SRF to SSF-CCF */

```
SIGNALLIST srfToSsf=
Ssf_CallProgress.ind,
Ssf_Facility.ind,
Ssf_FacilityResult.ind,
Ssf_FacilityError.ind,
Ssf_Data.ind,
Ssf_Setup.conf,
Ssf_Release.ind,
/*Additional signal for IP setup*/
Ssf_Setup.ind;
```

/* Signals from SSF-CCF to SRF */

```
SIGNALLIST ssfToSrf=
Ssf_Setup.req,
Ssf_Facility.req,
Ssf_FacilityResult.req,
Ssf_FacilityError.req,
Ssf_Data.req,
Ssf_Release.req,
Ssf_SubsequentAddress.req,
/*Additional signal for IP setup*/
Ssf_Setup.resp;
```

/* Signals from SRF to IP */

```
SIGNALLIST SigCon_Out=
Srf_Setup.req,
Srf_Facility.req,
Srf_FacilityResult.req,
Srf_FacilityError.req,
Srf_Data.req,
Srf_Release.req,
/*Additional signal for IP setup*/
Srf_Setup.resp;
```

/* Signals from IP to SRF */

```
SIGNALLIST SigCon_In=
Srf_CallProgress.ind,
Srf_Facility.ind,
Srf_FacilityResult.ind,
Srf_FacilityError.ind,
Srf_Data.ind,
Srf_Setup.conf,
Srf_Release.ind,
/*Additional signal for IP setup*/
Srf_Setup.ind;
```

/* Additional Signals required when the connection is established from the IP, this is not the normal case. */

```
SIGNAL
Srf_Setup.ind(Srf_SetupType),
Ssf_Setup.ind(Ssf_SetupType),
Ssf_Setup.resp(Ssf_SetupType),
Ssf_Setup.resp(Ssf_SetupType);
```



```

SIGNAL
EraseMessagePrim(EraseMessageType),
PlayAnnouncementPrim(PlayAnnouncementType),
PlayAnnouncementErrorPrim(PlayAnnouncementErrorType),
PromptAndCollectUserInformationPrim(PromptAndCollectUserInformationType),
PromptAndCollectUserInformationResultPrim(PromptAndCollectUserInformationResultType),
PromptAndCollectUserInformationErrorPrim(PromptAndCollectUserInformationErrorType),
PromptAndReceiveMessagePrim(PromptAndReceiveMessageType),
PromptAndReceiveMessageResultPrim(PromptAndReceiveMessageResultType),
PromptAndReceiveMessageErrorPrim(PromptAndReceiveMessageErrorType),
ScriptClosePrim(ScriptCloseType),
ScriptCloseErrorPrim(ScriptCloseErrorType),
ScriptEventPrim(ScriptEventType),
ScriptInformationPrim(ScriptInformationType),
ScriptInformationErrorPrim(ScriptInformationErrorType),
ScriptRunPrim(ScriptRunType),
ScriptRunErrorPrim(ScriptRunErrorType),
SpecialisedResourceReportPrim(SpecialisedResourceReportType),
CancelPrim(CancelType),
CancelErrorPrim(CancelErrorType),
AssistRequestInstructionsPrim(AssistRequestInstructionsType),
ApplicationBeginPrim(TransactionIdType),
ApplicationAbortPrim(TransactionIdType),
ApplicationEndPrim(TransactionIdType);

```

```

/*Signals from SRF to SCF*/
SIGNALLIST CS2_INAP_Prim_To_SCF=
PlayAnnouncementErrorPrim,
PromptAndCollectUserInformationResultPrim,
PromptAndCollectUserInformationErrorPrim,
PromptAndReceiveMessageResultPrim,
PromptAndReceiveMessageErrorPrim,
ScriptCloseErrorPrim,
ScriptEventPrim,
ScriptInformationErrorPrim,
ScriptRunErrorPrim,
SpecialisedResourceReportPrim,
CancelErrorPrim,
AssistRequestInstructionsPrim,
ApplicationBeginPrim,
ApplicationAbortPrim,
ApplicationEndPrim;
/*Signals from SCF to SRF*/
SIGNALLIST CS2_INAP_Prim_From_SCF=
EraseMessagePrim,
PlayAnnouncementPrim,
PromptAndCollectUserInformationPrim,
PromptAndReceiveMessagePrim,
ScriptClosePrim,
ScriptInformationPrim,
ScriptRunPrim,
CancelPrim,
ApplicationBeginPrim,
ApplicationAbortPrim,
ApplicationEndPrim;

```

```

SIGNALLIST srmTosrm=
(CS2_INAP_Prim_From_SCF),
(SigCon_In),
(ssfTosrf);

SIGNALLIST srsrmTosrm
(CS2_INAP_Prim_To_SCF),
(SigCon_Out),
(srfTossf);

```



/*
Notes to be resolved at the ITU-T Q.6 1996 Interim Meeting

NOTE 1:

It is proposed that all SCF/SRF in the ASN1 Module IN-CS2-SCF-SRF-ops-args {ccitt recommendation q 1228 modules(0) in-cs2-scf-srf-ops-args (7) version1(0)} contain a CSId instead of a LegId (or no coding at all), in order to address the right call segment where the operation should relay the message in case of multiple CS's in a CSA.

NOTE 2:

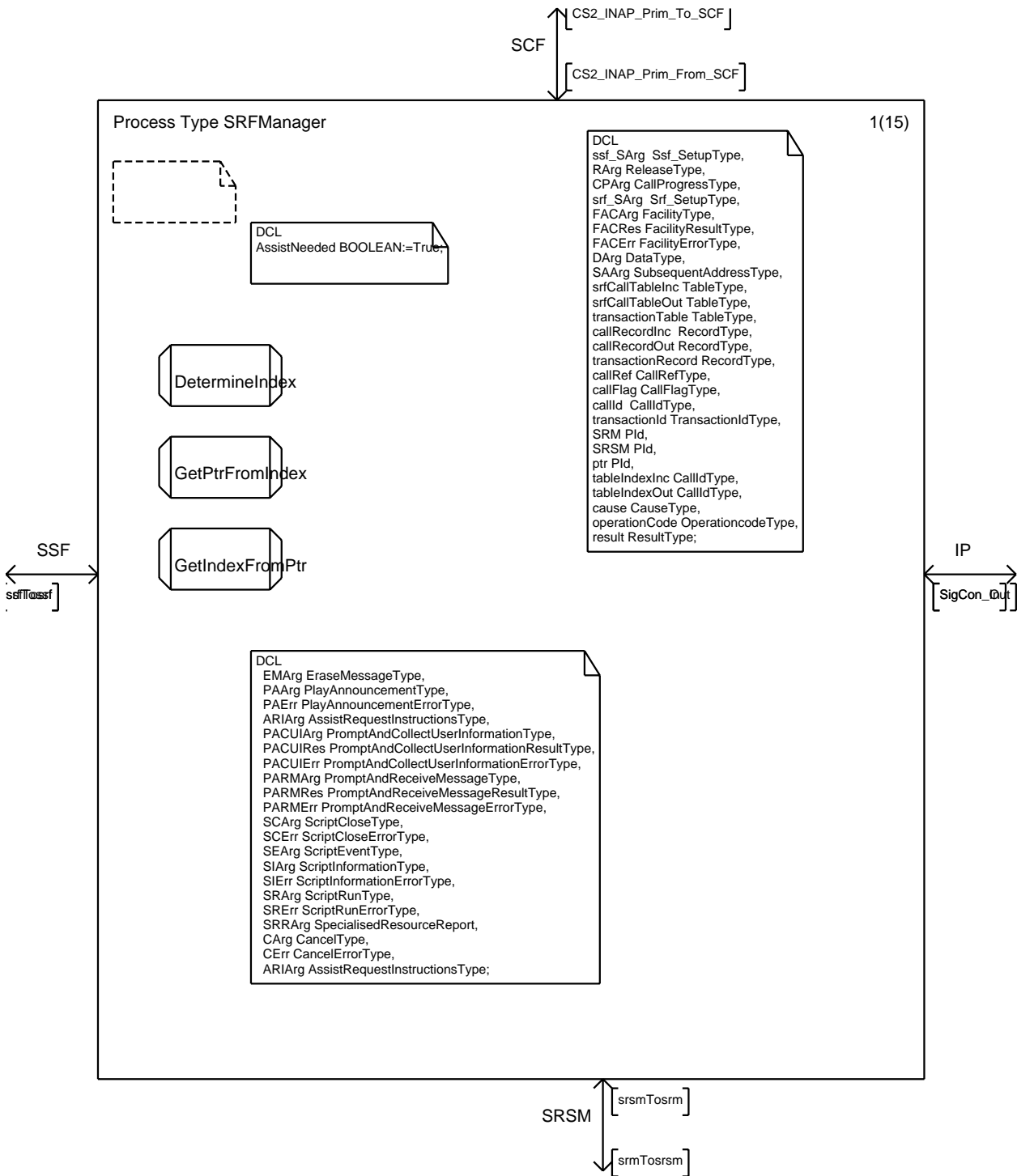
The disconnectFromIPForbidden parameter is missing in the following SCF-SSF-SRF messages ScriptRun.

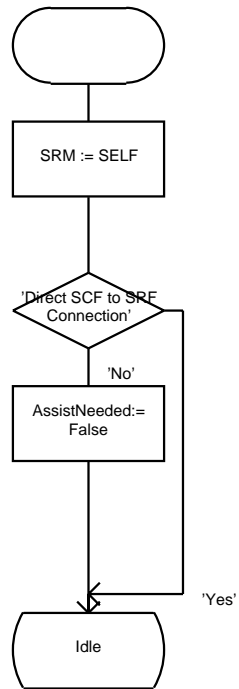
A parameter should be included into the ScriptRun operation to enable from the SCF an SRF initiated disconnect

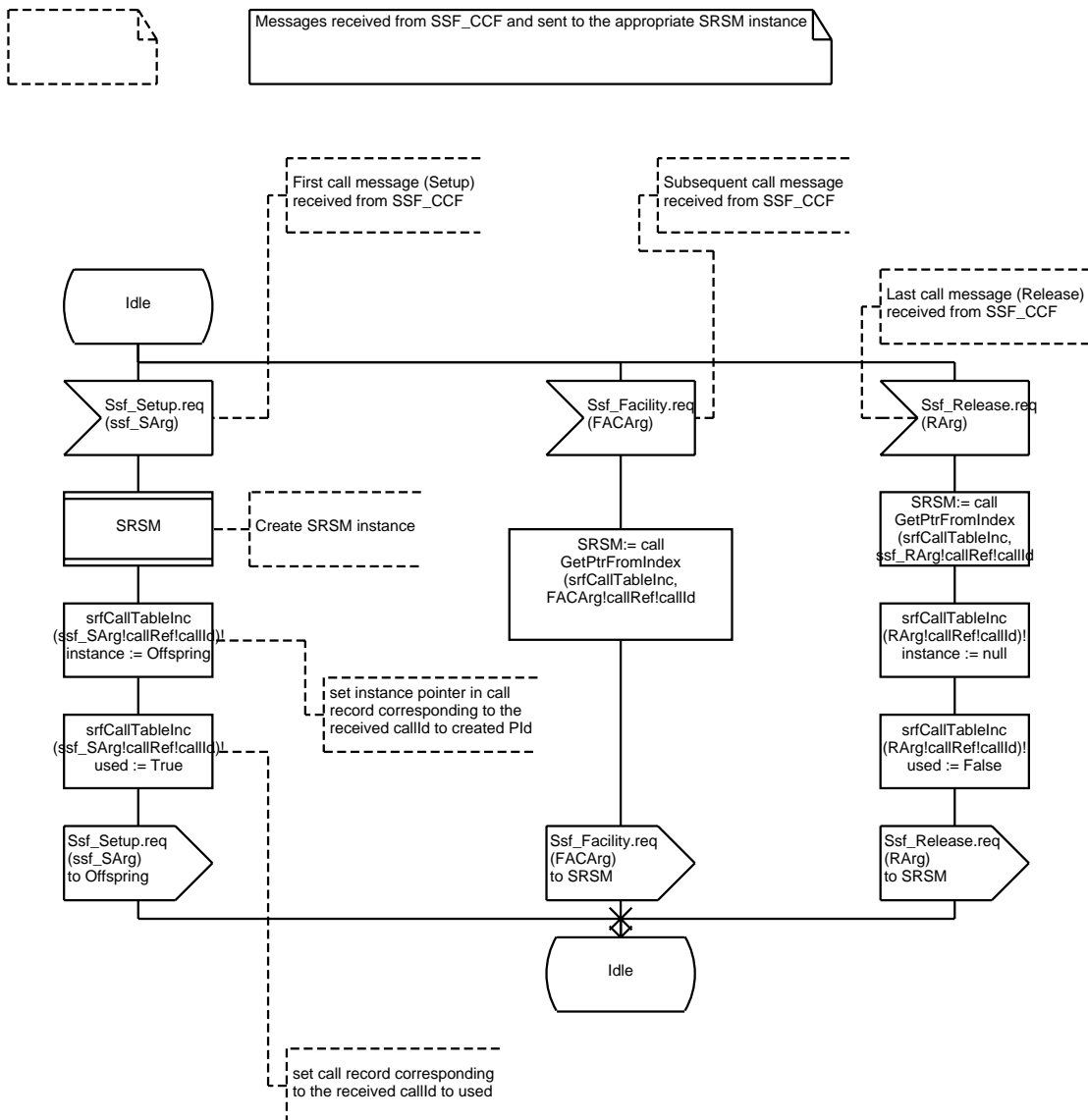
NOTE 3:

The Cancel operation must have an CSId, in order to address the right call segment where the operation should relay the message in case of multiple CS's in a CSA.

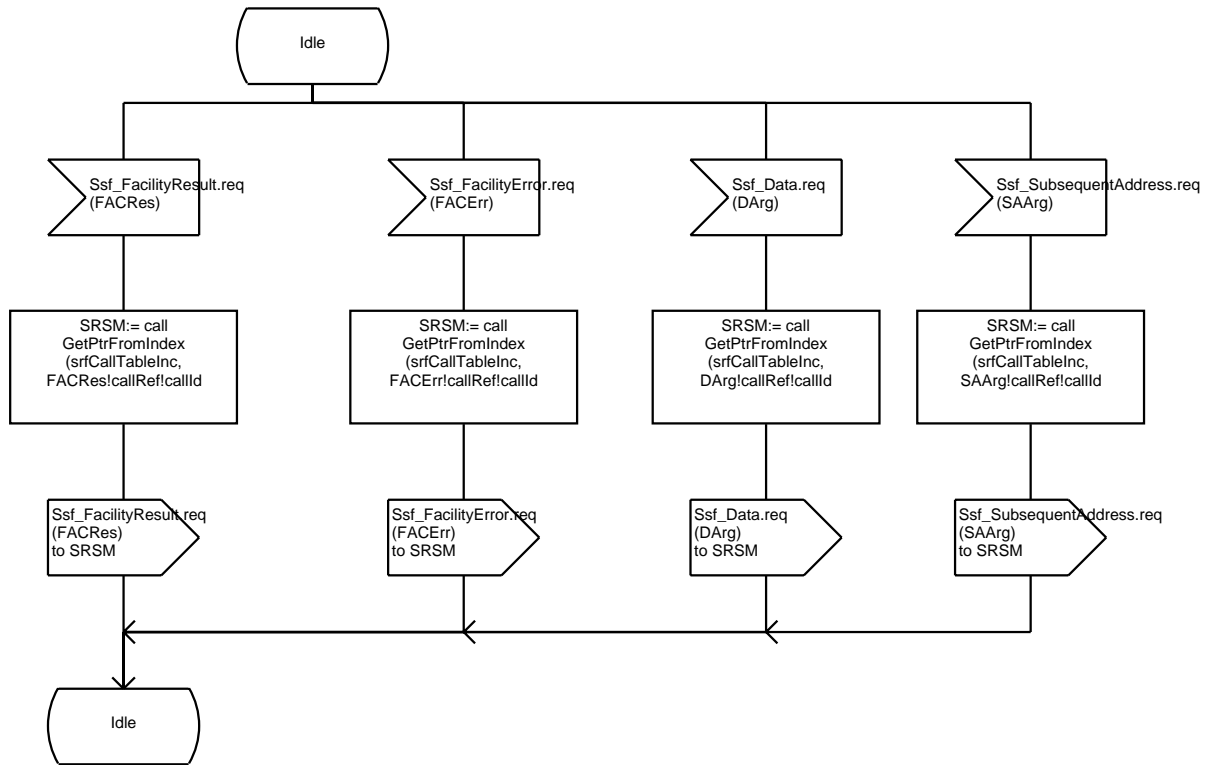
*/

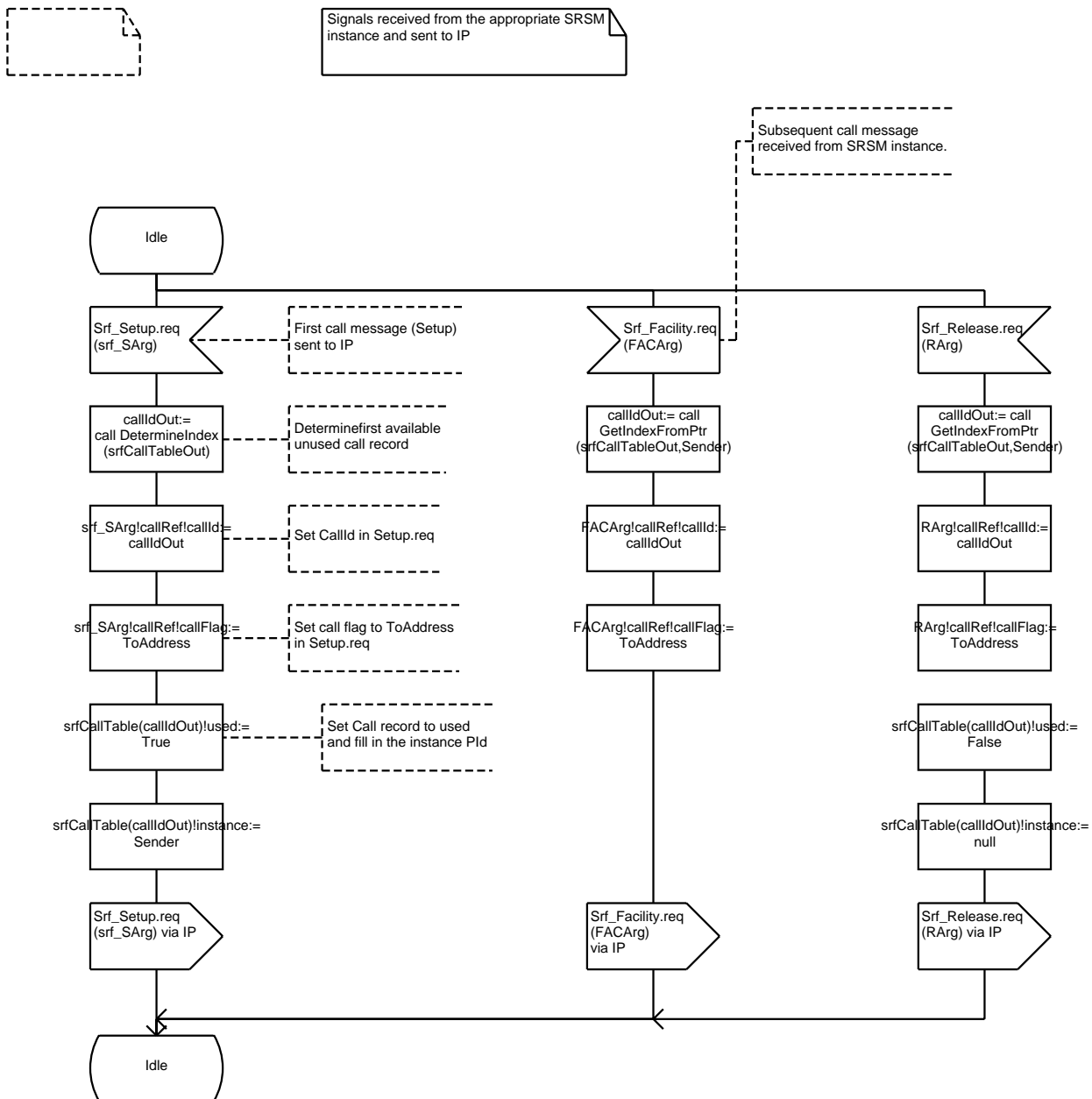


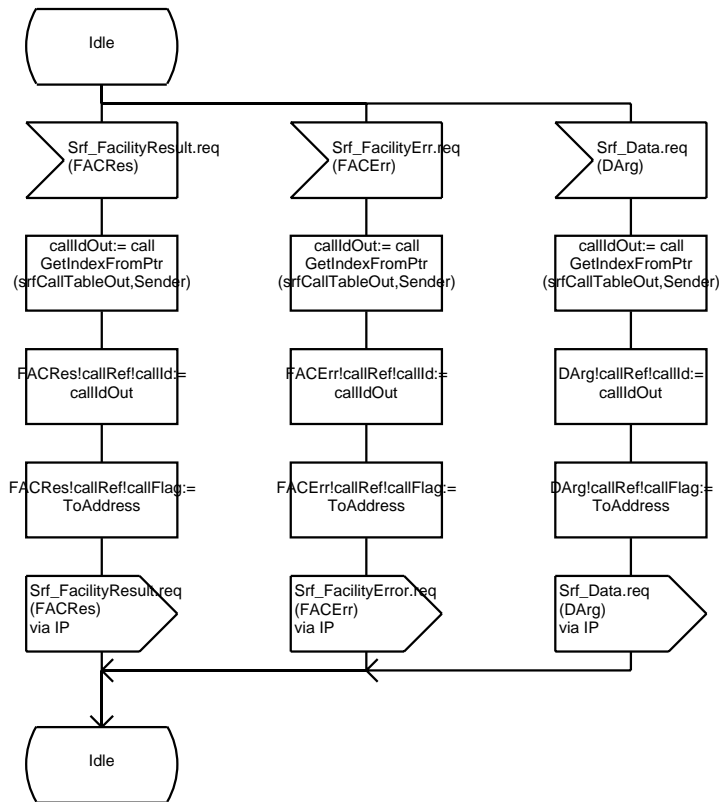
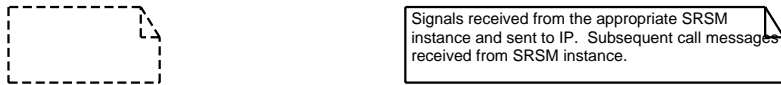


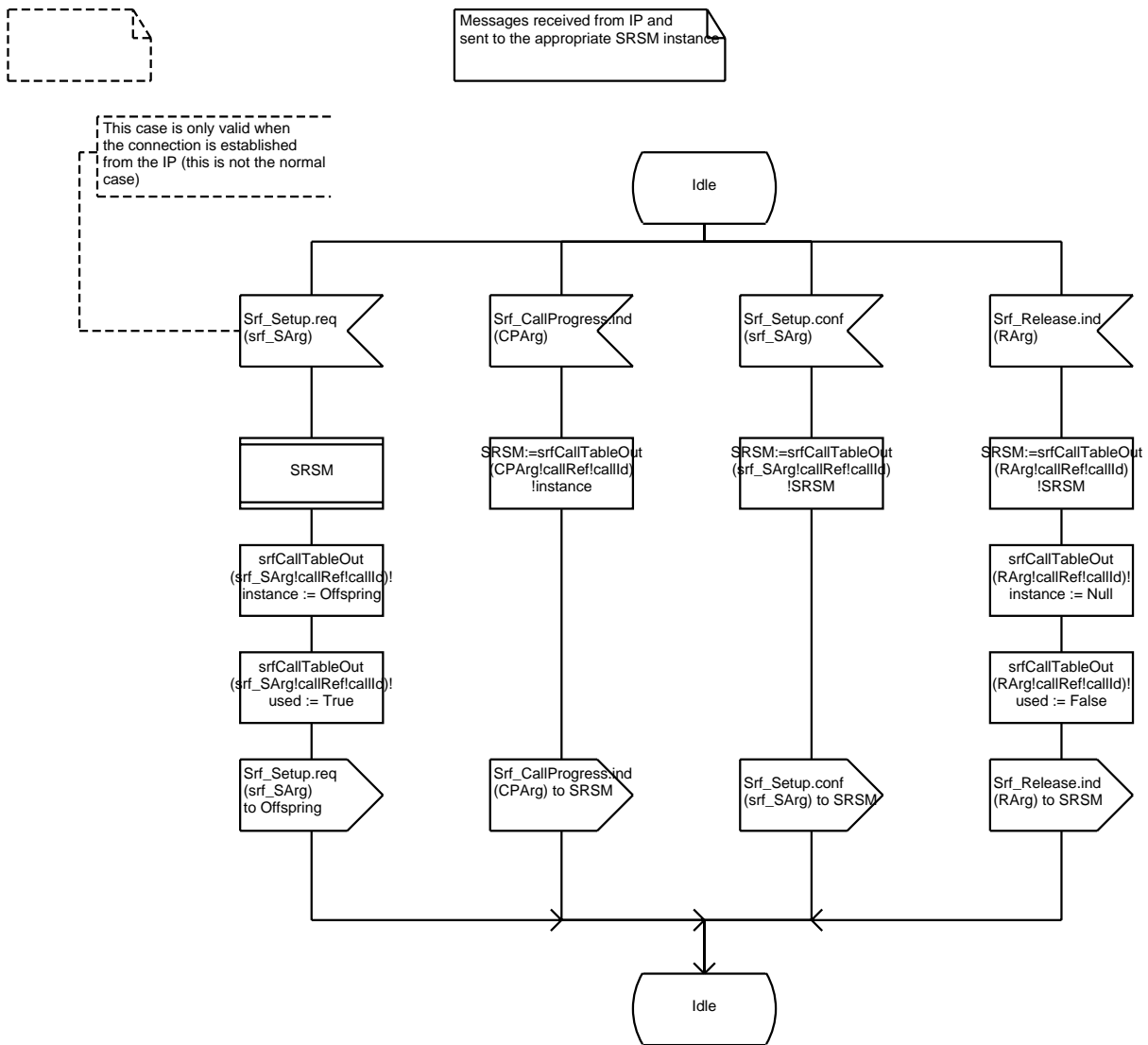


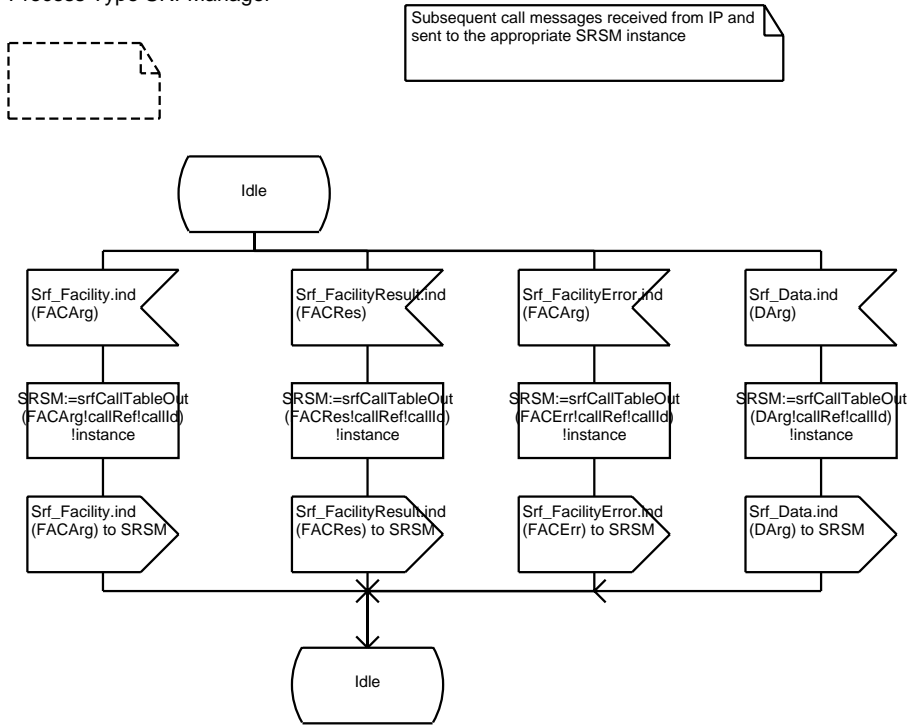
Messages received from SSF_CCF and sent to the appropriate SRSM instance.
Subsequent call messages received from SSF_CCF.

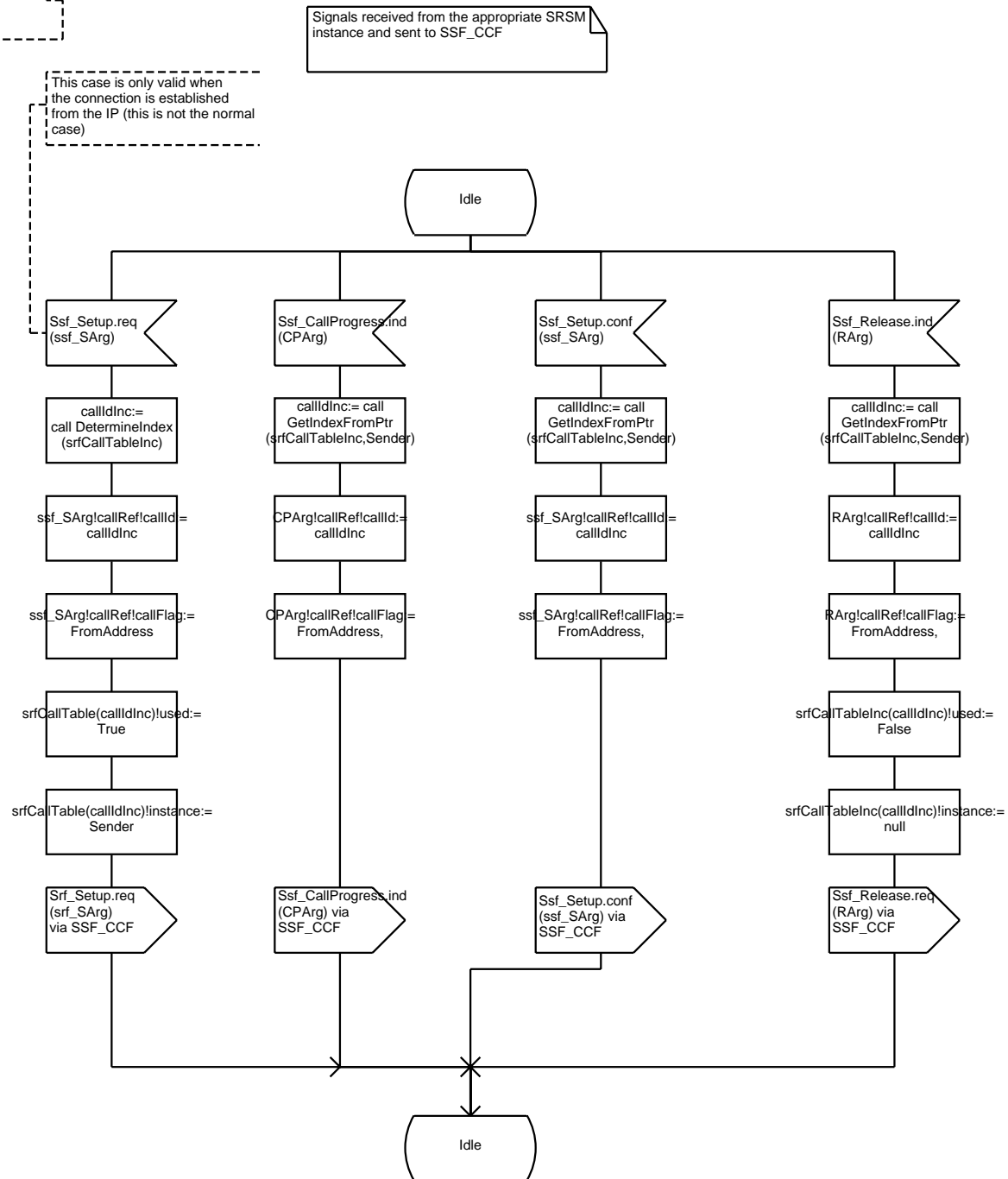


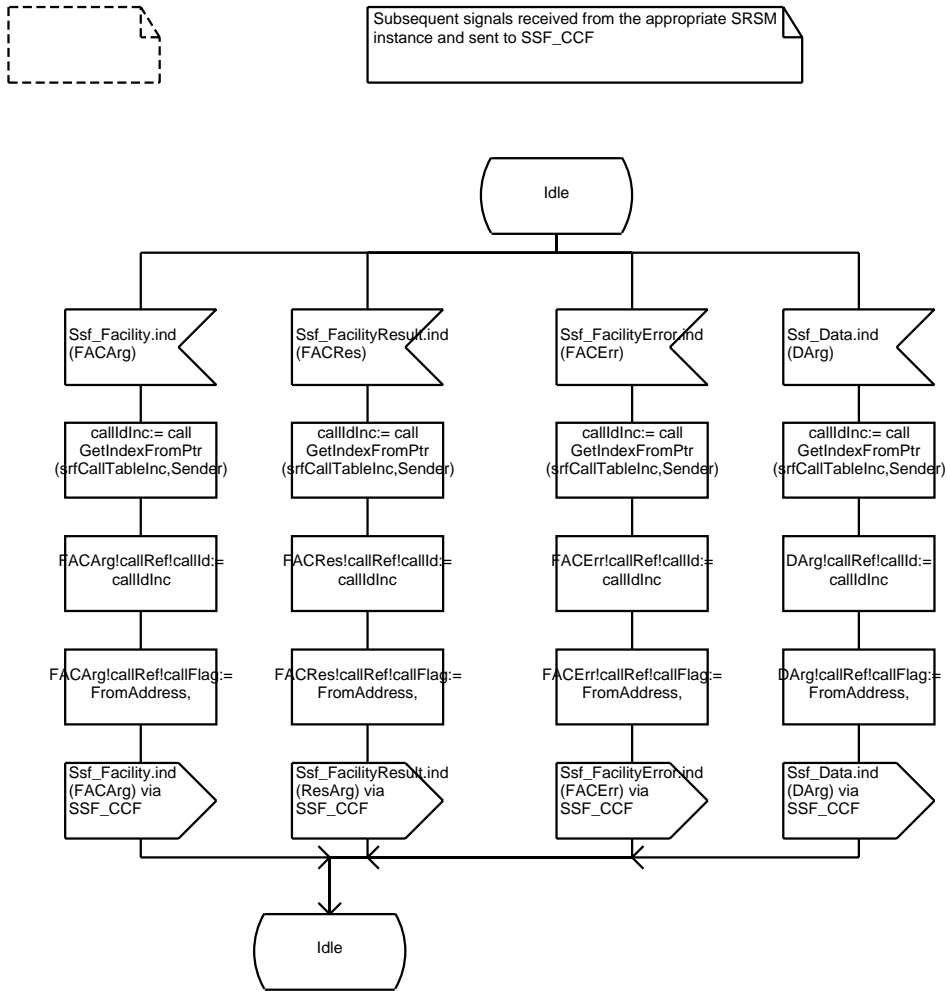






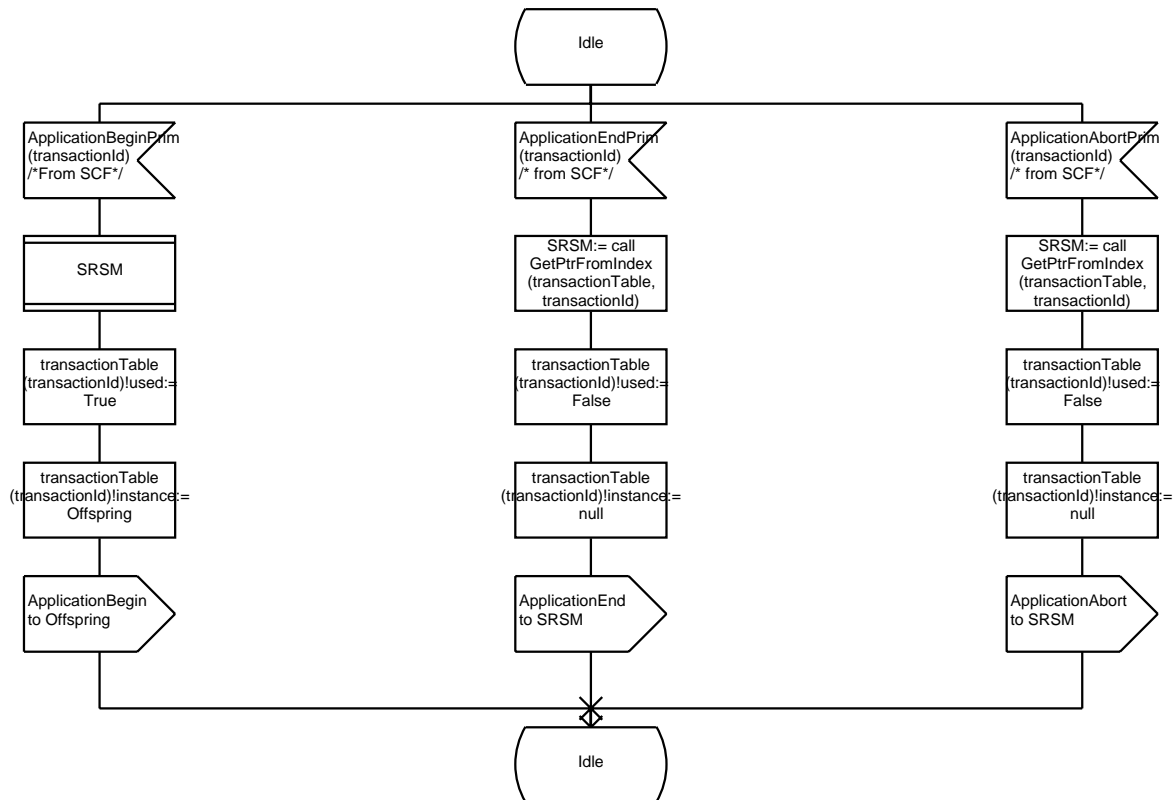


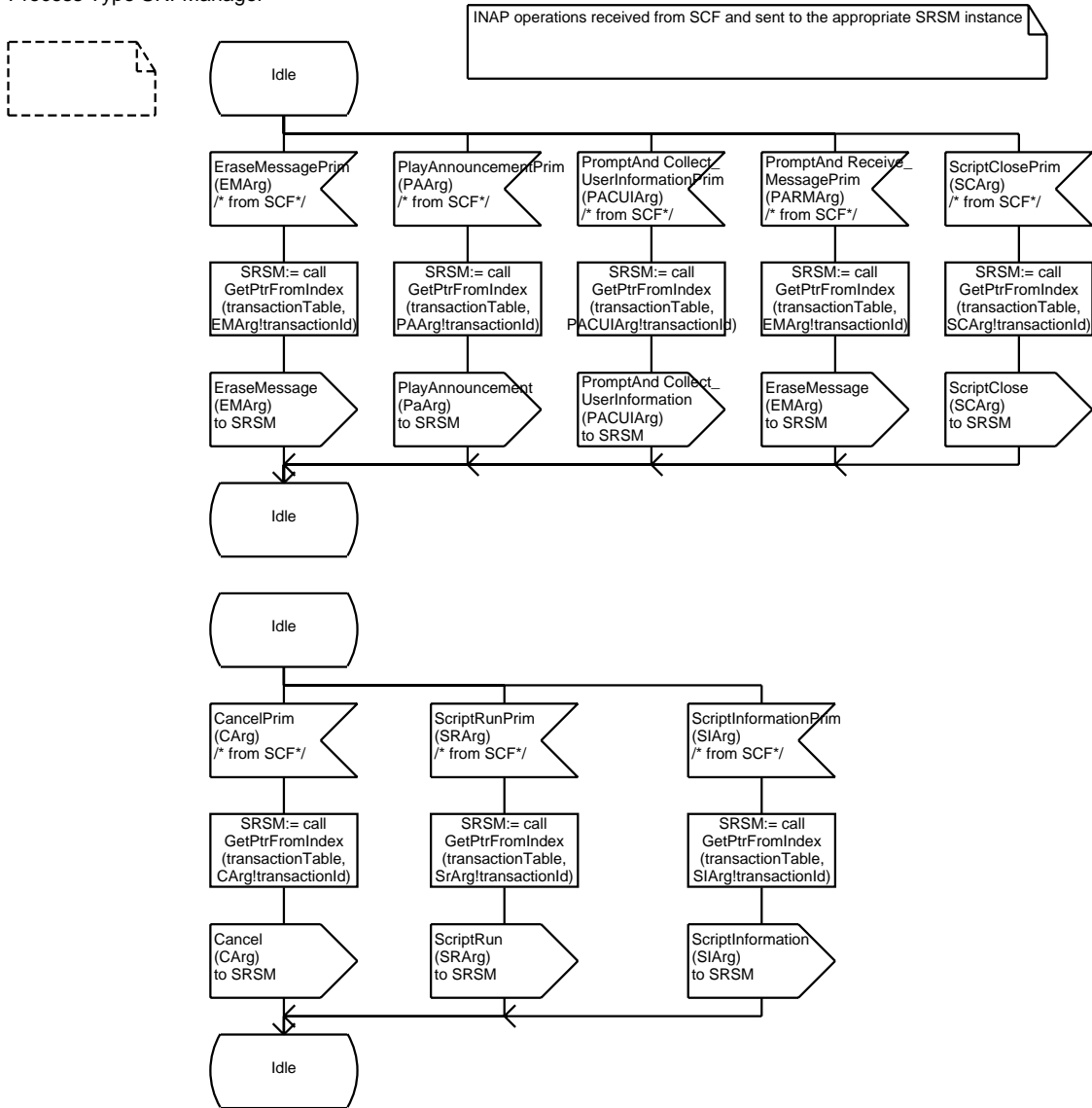






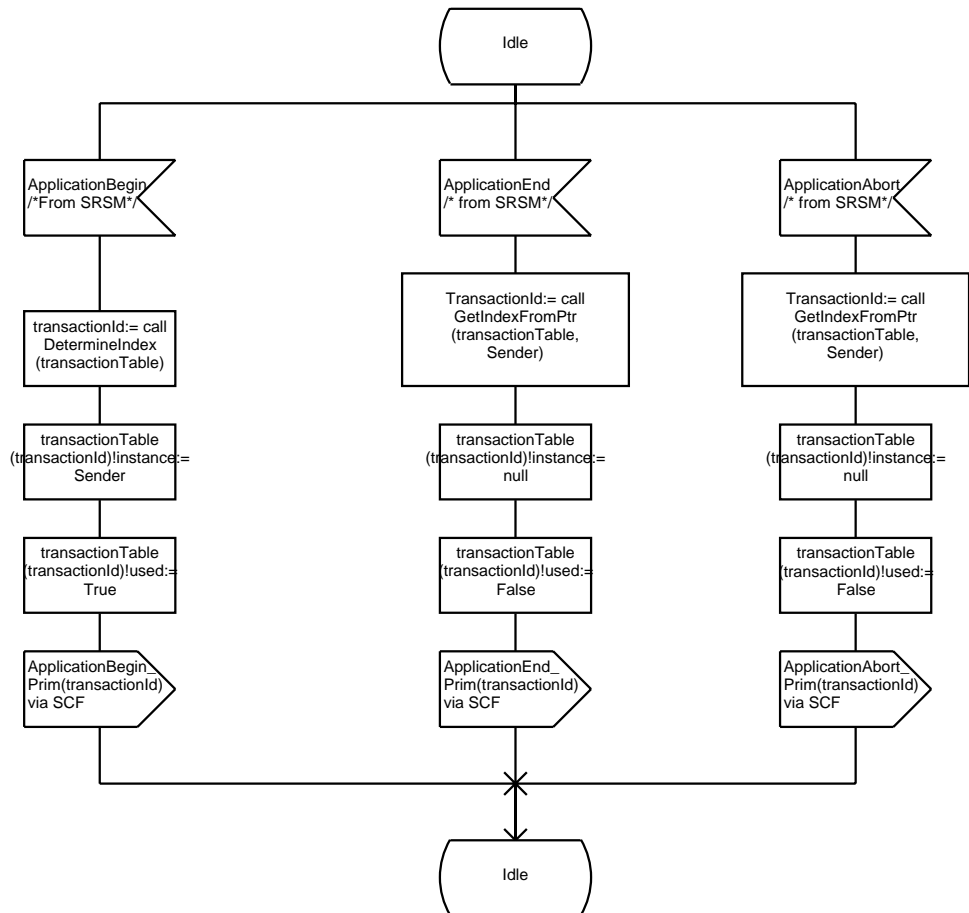
Signals received from SCF and sent to the appropriate SRSM instance

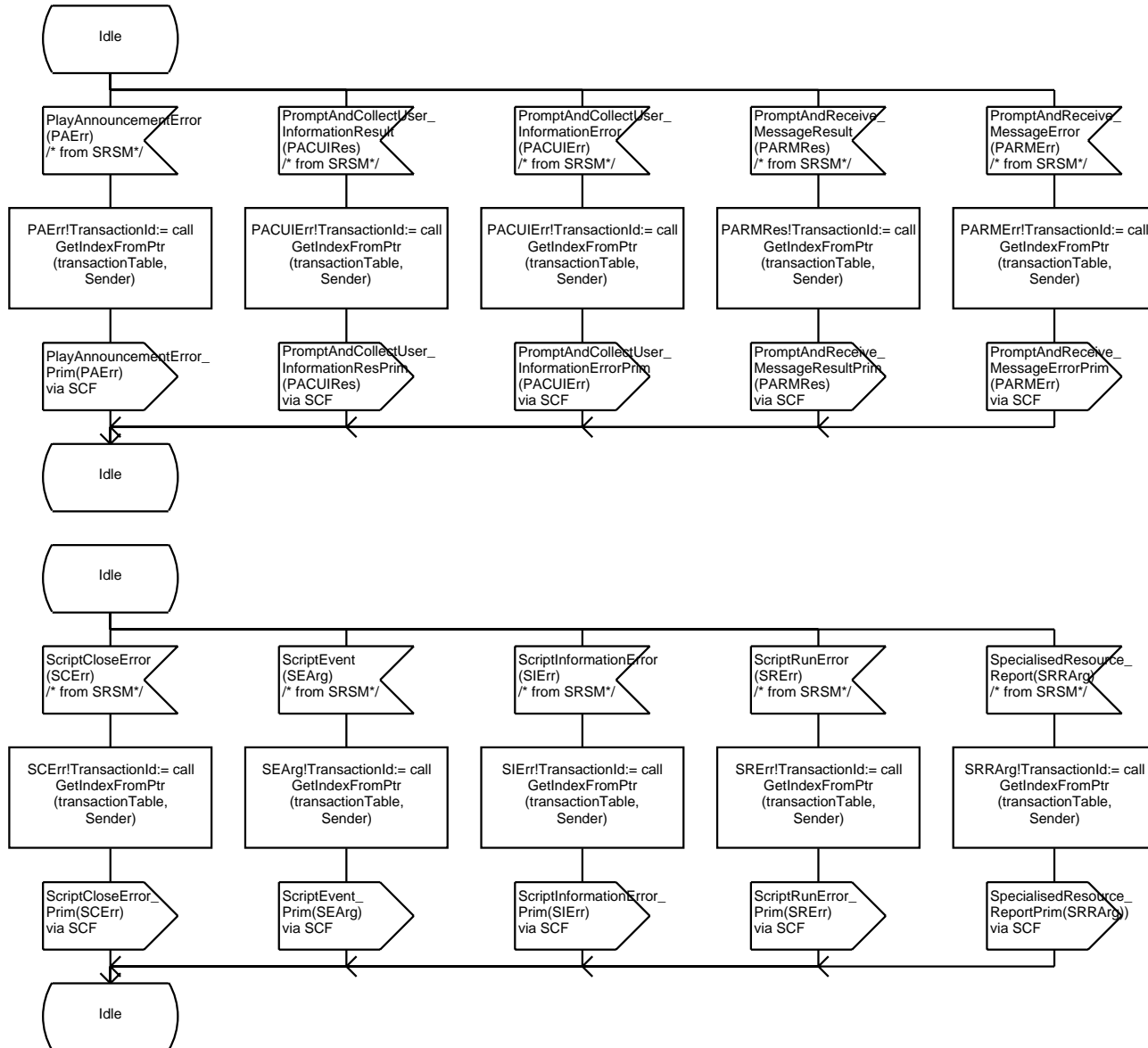


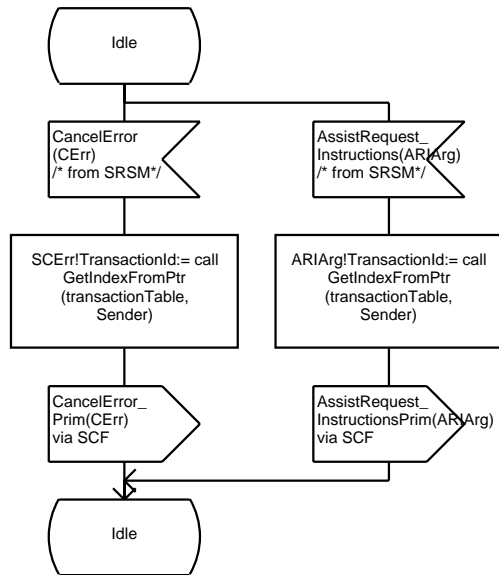




Signals received from SRSM instance and sent to the SCF





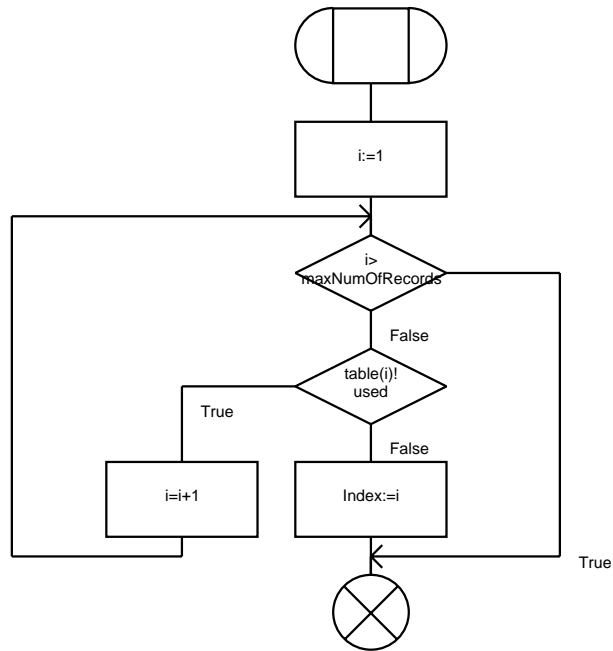


Procedure DetermineIndex

1(1)

FPAR
IN table TableType;
RETURNS index IndexType;

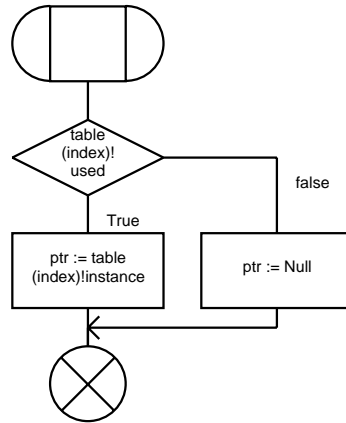
DCL
i Integer;



Procedure GetPtrFromIndex

1(1)

```
;FPAR  
;IN table TableType;  
;IN index IndexType;  
;RETURNS ptr;
```

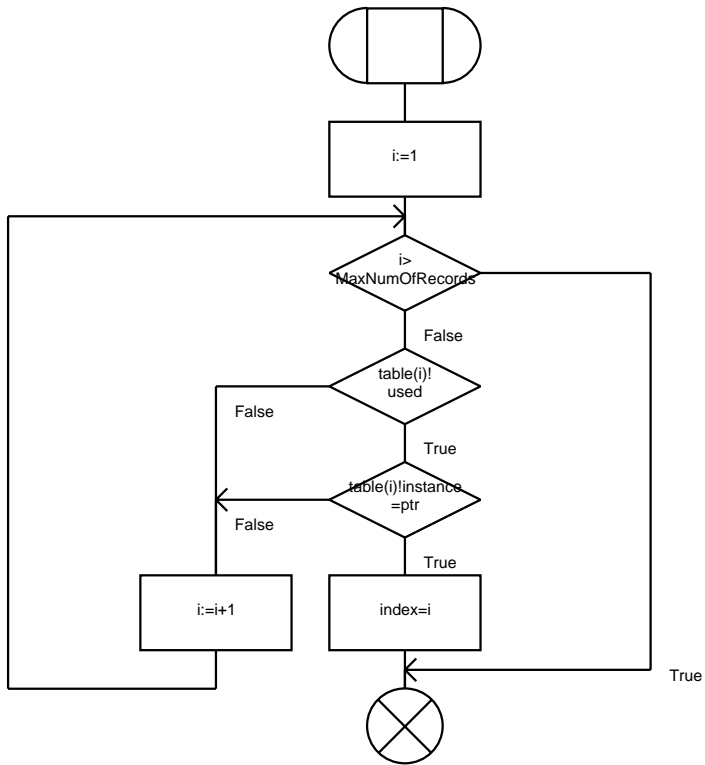


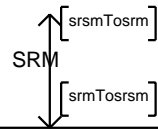
Procedure GetIndexFromPtr

1(1)

```
;FPAR
;IN table TableType;
;IN ptr PId;
;RETURN index IndexType;
```

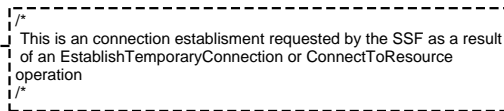
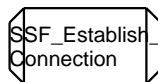
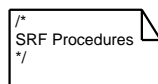
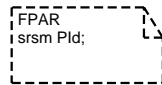
```
DCL
i Integer;
```





Process Type SRFStateModel

1(13)



FPAR
srsm PId;

/*
INAP Operation processing procedures
*/

Process_
PlayAnnouncement

Process_
EraseMessage

Process_
PromptAndCollect_
UserInformation

Process_
PromptAndReceive_
Message

Process_
ScriptClose

Process_
ScriptInformation

Process_
ScriptRun

Process_
Cancel

FPAR
srm PId;

/* Variable Declarations */

DCL

ApplicationActive BOOLEAN := FALSE;
AssistNeeded BOOLEAN := TRUE;
disconnectFromIPForbiddenFlag BOOLEAN := FALSE;
requestAnnouncementCompleteFlag BOOLEAN := FALSE;

/* Timer Definitions */

TIMER Tsrf;

DCL

TsrfDuration1 DURATION := 3600000;
TsrfDuration DURATION;

DCL

ssf_SArg Ssf_SetupType,
RArg ReleaseType,
CPArg CallProgressType,
srf_SArg Srf_SetupType,
FACArg FacilityType,
FACRes FacilityResultType,
FACErr FacilityErrorType,
DArg DataType,
SAArg SubsequentAddressType,
srfCallTableInc TableType,
srfCallTableOut TableType,
transactionTable TableType,
callRecordInc RecordType,
callRecordOut RecordType,
transactionRecord RecordType,
callRef CallRefType,
callFlag CallFlagType,
callId CallIdType,
transactionId TransactionIdType,
SRM PId,
SRSM PId,
ptr PId,
tableIndexInc CallIdType,
tableIndexOut CallIdType,
cause CauseType,
operationCode OperationcodeType,
result ResultType;

SIGNAL

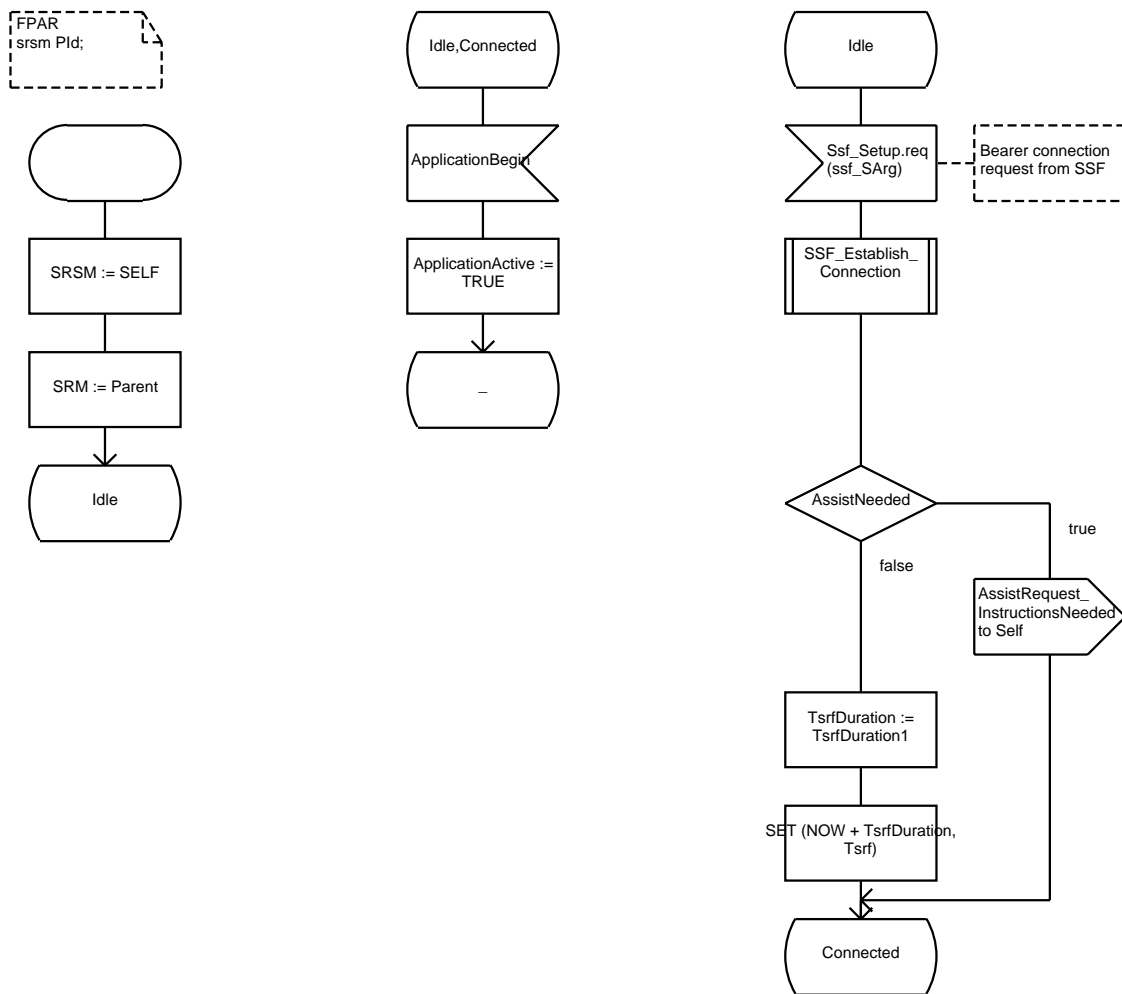
AssistRequestInstructionsNeeded;
ssf_FacPAREs.req;

FPAR
srsrn PId;

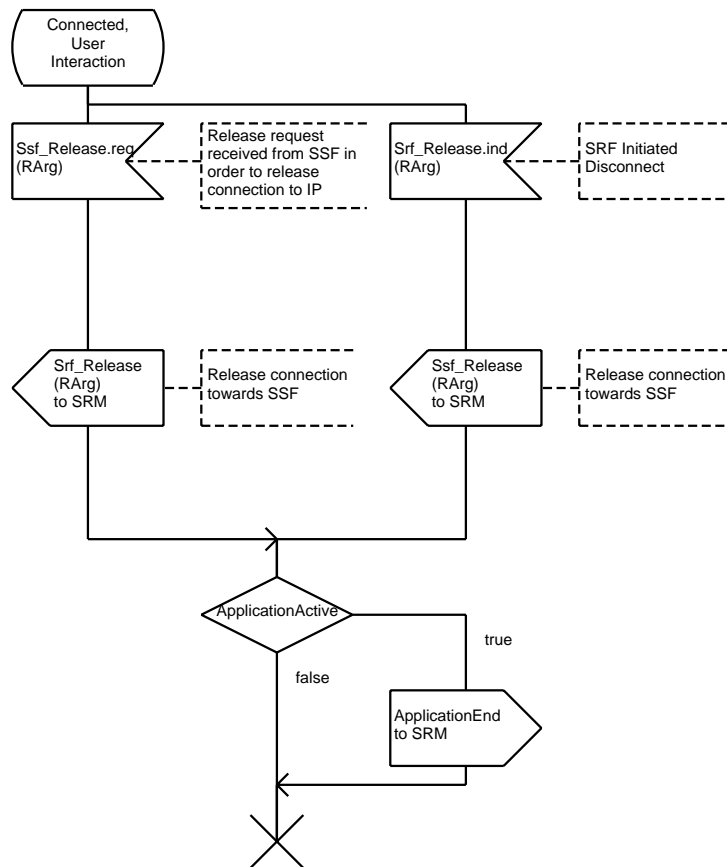
DCL
EMArg EraseMessageType,
PAAArg PlayAnnouncementType,
PAErr PlayAnnouncementErrorType,
ARIAArg AssistRequestInstructionsType,
PACUIArg PromptAndCollectUserInformationType,
PACUIRes PromptAndCollectUserInformationResultType,
PACUIErr PromptAndCollectUserInformationErrorType,
PARMArg PromptAndReceiveMessageType,
PARMRes PromptAndReceiveMessageResultType,
PARMErr PromptAndReceiveMessageErrorType,
SCArg ScriptCloseType,
SEArg ScriptCloseErrorType,
SEArg ScriptEventType,
SIArg ScriptInformationType,
SIErr ScriptInformationErrorType,
SRArg ScriptRunType,
SRArg ScriptRunErrorType,
SRRArg SpecialisedResourceReport,
CArg CancelType,
CErr CancelErrorType,
ARIAArg AssistRequestInstructionsType;

FPAR
srsm PId;

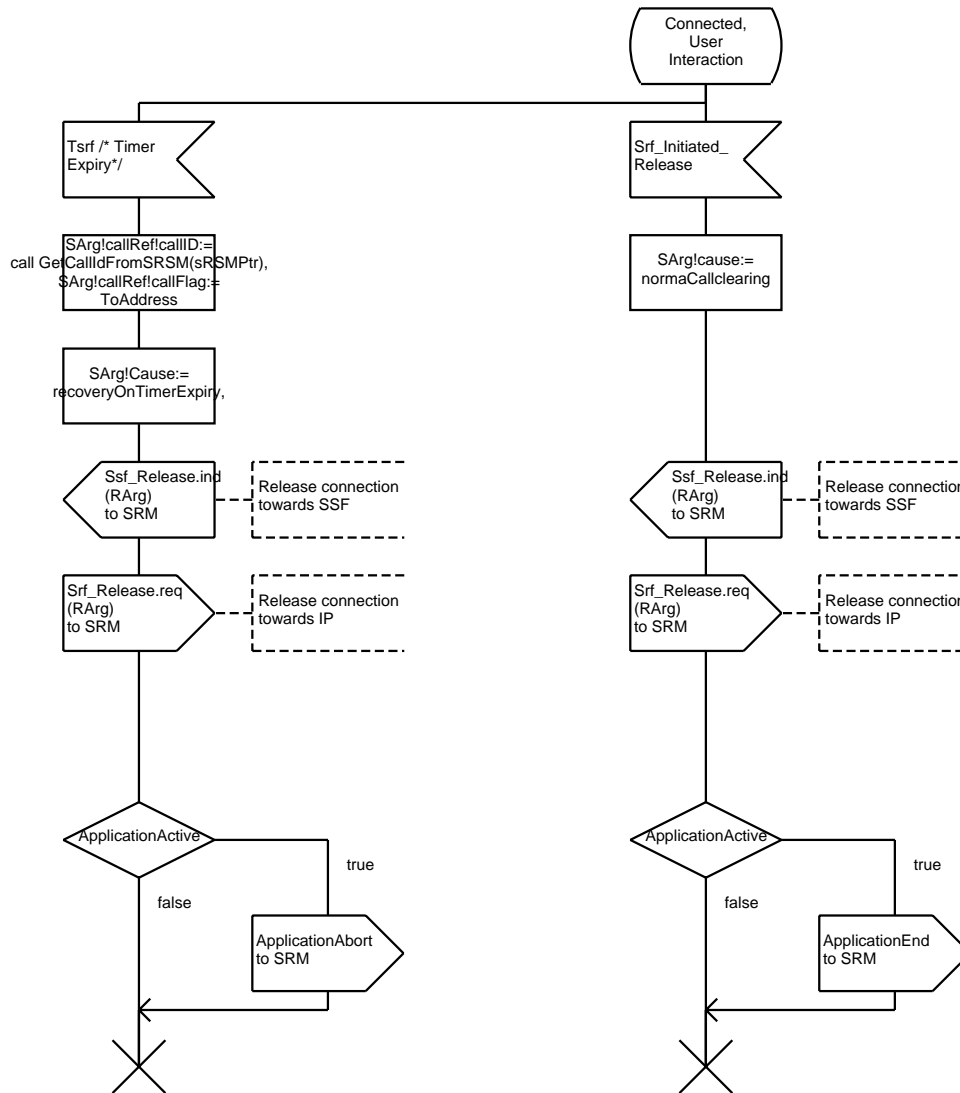
/*
INAP Operation processing procedures
*/

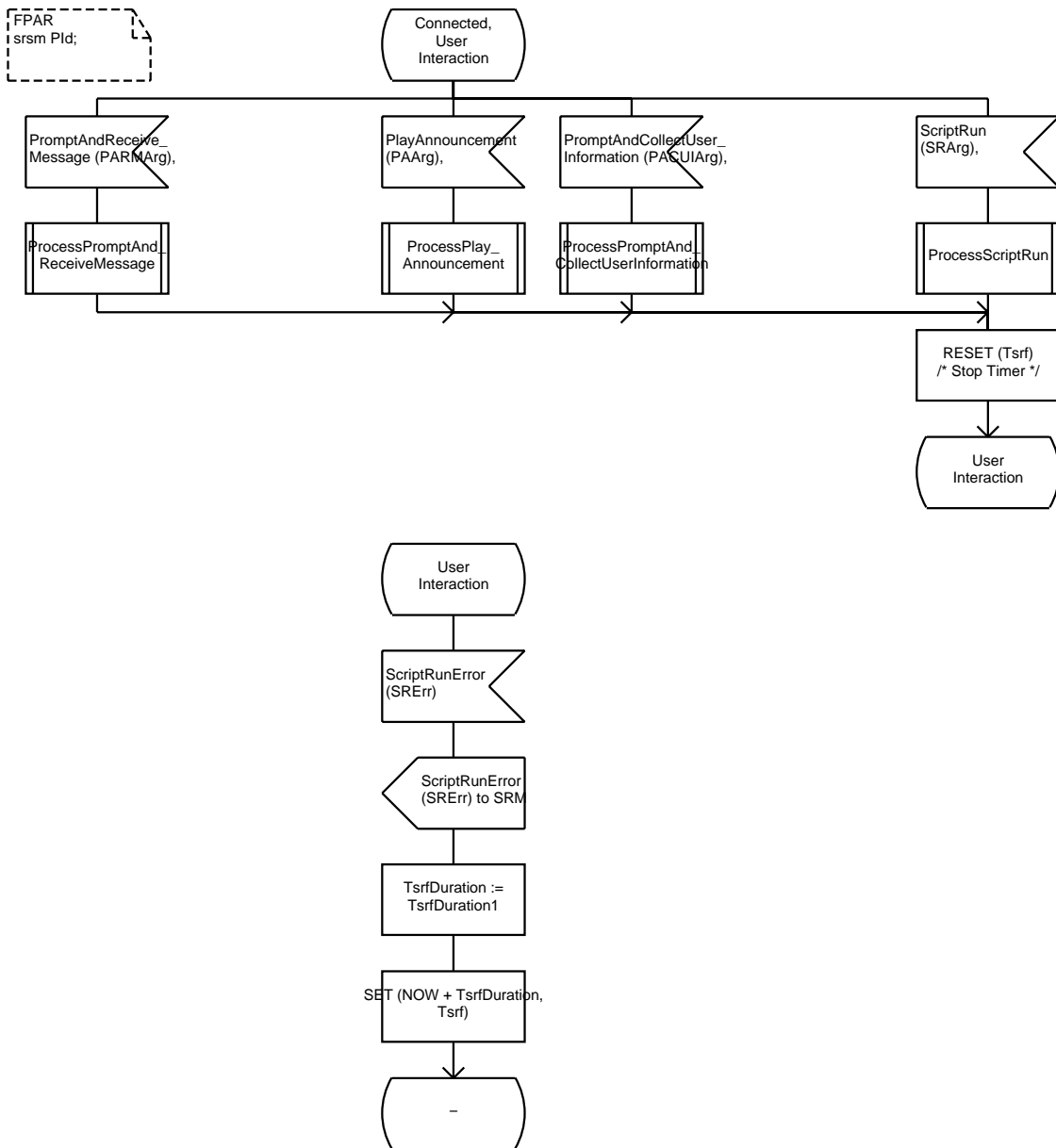


FPAR
srm PId;

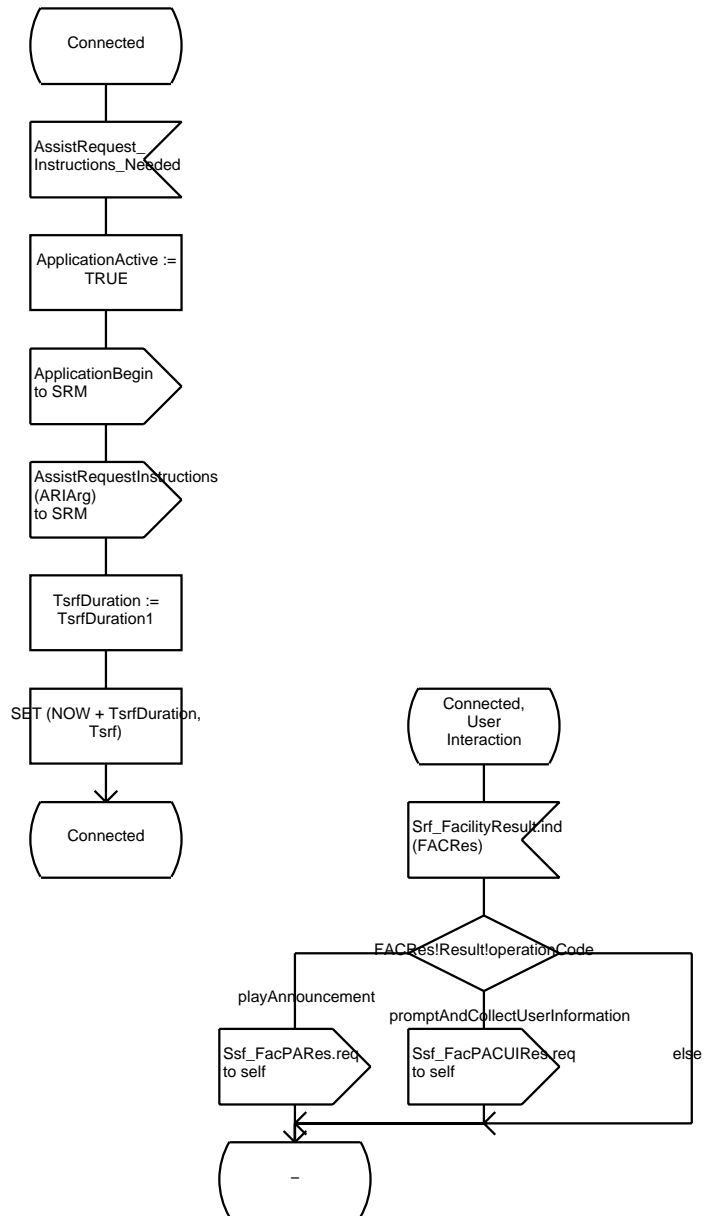


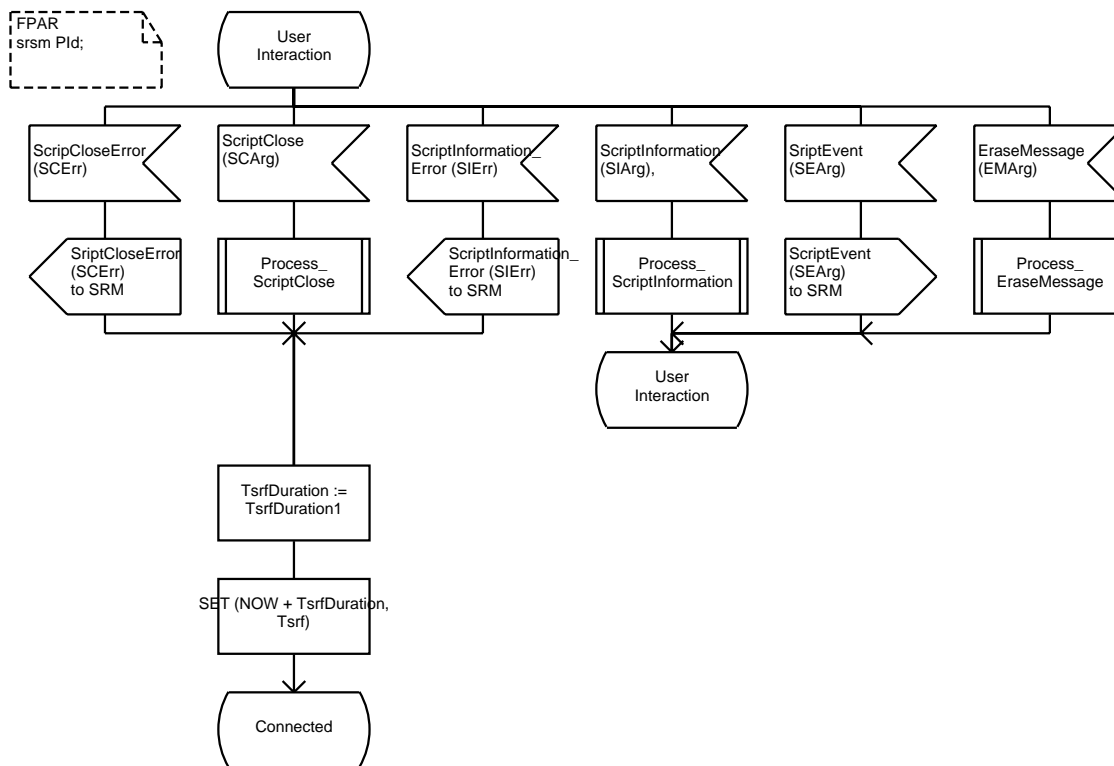
FPAR
srm PId;

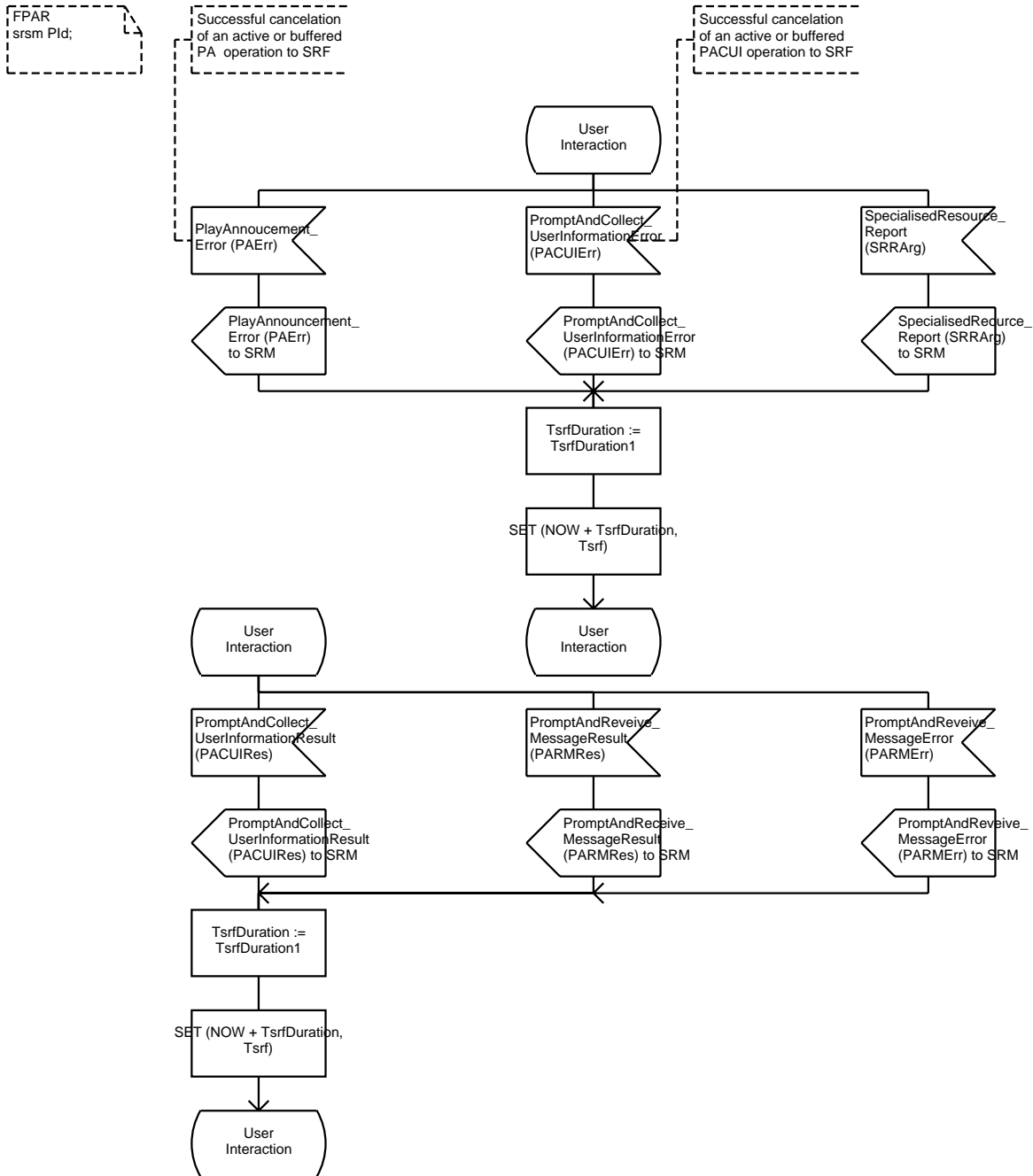


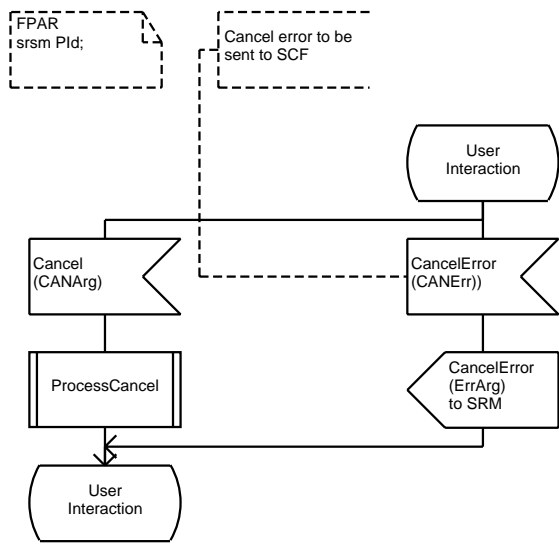


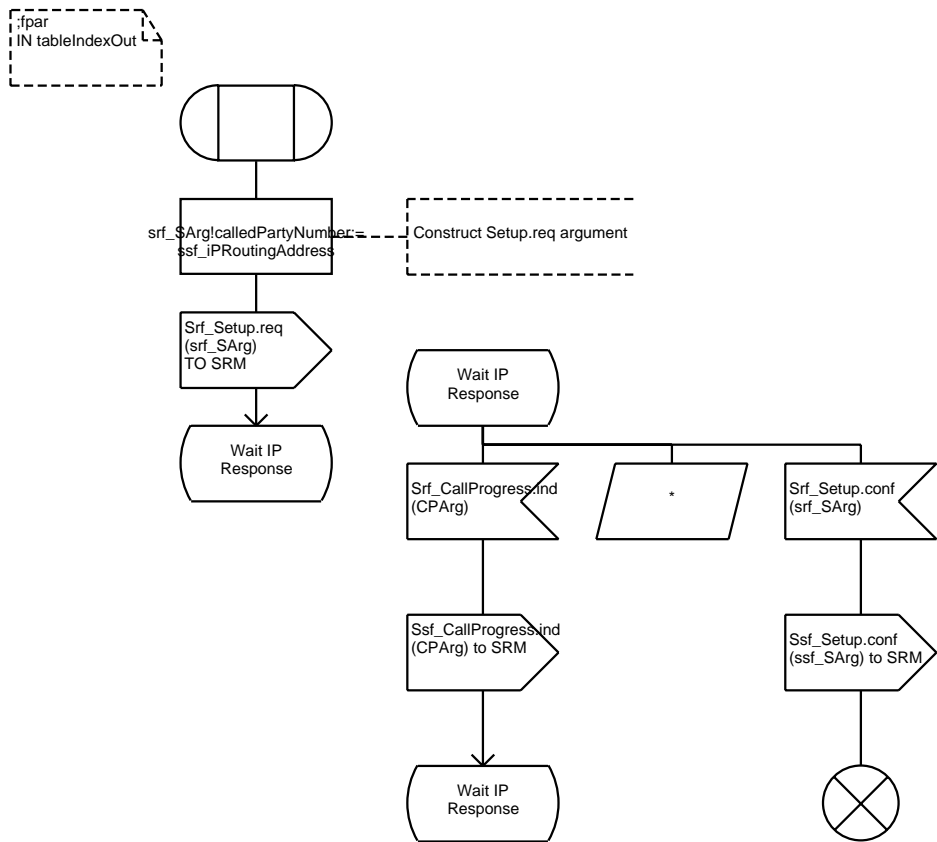
FPAR
srm PId;







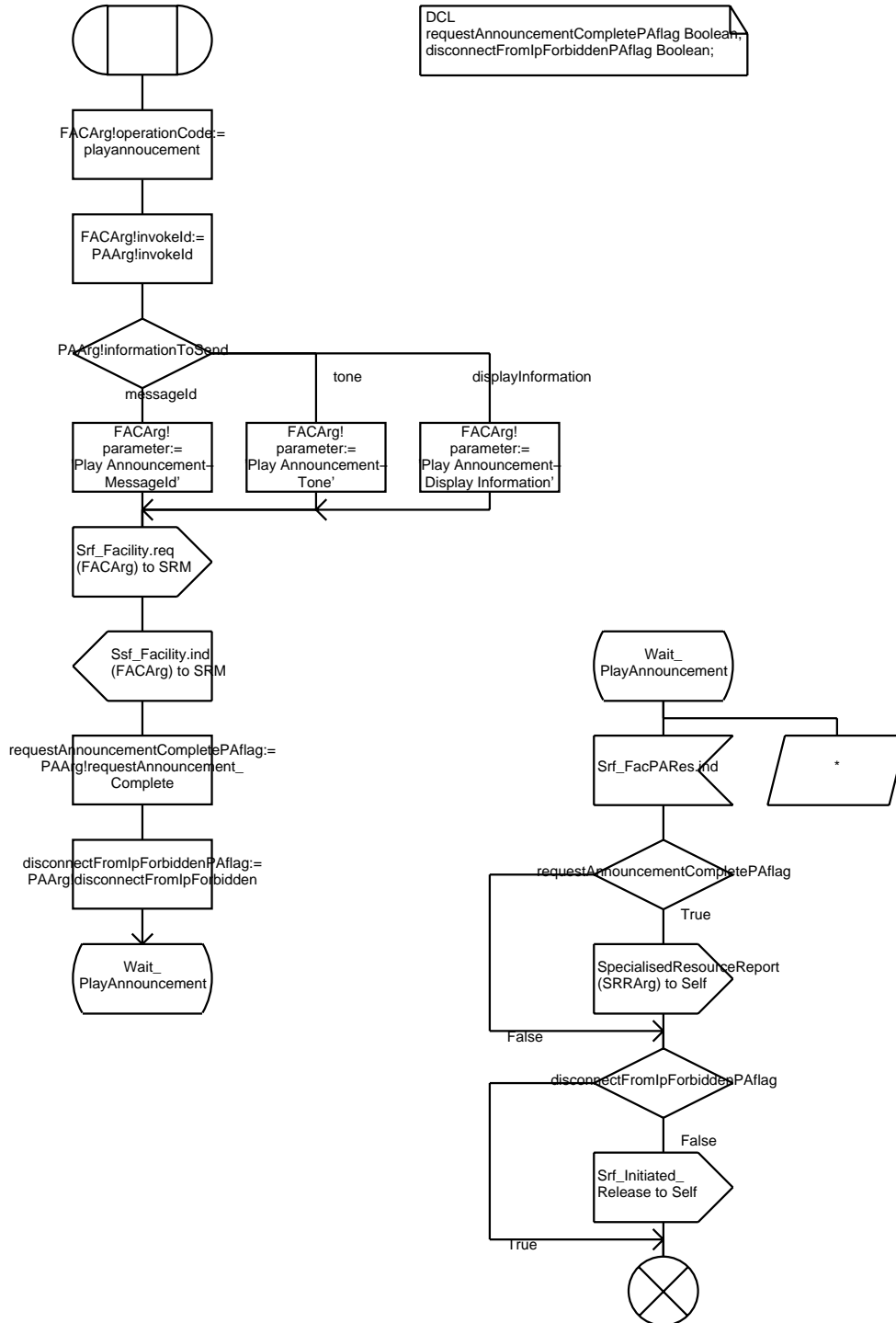






/* This procedure is given for simulation and validation purposes only */

DCL
requestAnnouncementCompletePAflag Boolean;
disconnectFromIpForbiddenPAflag Boolean;





DCL
disconnectFromIpForbiddenPACUIflag Boolean

