

Annex B: q2931ss

```
Q2931ss DEFINITIONS ::=  
-- Description: Data for Q.2931 to and from the lower layer (SAAL)  
-- Author: PT87/ETSI, Nils Fischbeck/HUB  
-- Version: 2.1  
-- Last Change: 20.11.97  
BEGIN  
  
AALData ::= OCTET STRING  
AALUnitdata ::= OCTET STRING  
  
-- Who caused the release: 1 - other side, 0 - error condition  
AALRelease ::= INTEGER  
  
END
```

Annex B: appq2931ss

```

AppQ2931ss DEFINITIONS ::=

-- Description: Data for upper interface of Q.2931/Q.2971/Q.2932
-- Author: PT87/ETSI, Nils Fischbeck/HUB
-- Version: 2.1
-- LastChange: 20.11.97
BEGIN

IMPORTS TrunkNumber FROM AdmQ2931ss;

MessageType ::= ENUMERATED {
    alertingMsgId(1),
    callProceedingMsgId(2),
    connectMsgId(7),
    connectAcknowledgeMsgId(15),
    notifyMsgId(110),
    releaseMsgId(77),
    releaseCompleteMsgId(90),
    restartMsgId(70),
    restartAcknowledgeMsgId(78),
    setupMsgId(5),
    statusMsgId(125),
    statusEnquiryMsgId(117),
    addPartyMsgId(128),
    addPartyAcknowledgeMsgId(129),
    partyAlertingMsgId(133),
    addPartyRejectMsgId(130),
    dropPartyMsgId(131),
    dropPartyAcknowledgeMsgId(132),
    facilityMsgId(98),
    unknownMsgId(255)
}

Flag ::= BOOLEAN
-- Used as Call reference or Endpoint reference flag
-- '0'B: Originator, '1'B: Destination
-- ETS 300 443-1 subclause 4.3, ETS 300 771-1 subclause 8.2.1

CRValue ::= INTEGER -- (0..8388607)

-- Call reference value BITSTRING[23]
-- ETS 300 443-1 subclause 4.3

CallReference ::= SEQUENCE {
    -- Call reference
    -- ETS 300 443-1 subclause 4.3
    CRFlag [1] Flag,
    CRValue [2] CRValue
}

-- AALType ::= ENUMERATED {
-- aALTypeVoice(0),
-- aALType1(1),
-- aALType2(2),
-- aALType34(3),
-- aALType5(5),
-- aALTypeUserDefined(16)
-- }

AAL1Subtype ::= ENUMERATED {
    -- ETS 300 443-1 subclause 4.5.5
    aAL1Subtype0(0),
    aAL1Subtype1(1),
    aAL1Subtype2(2),
    aAL1Subtype4(4),
    aAL1Subtype5(5)
}

CBRRate ::= ENUMERATED {
    -- ETS 300 443-1 subclause 4.5.5
    cBR64k(1),
    cBR1544k(4),
    cBR6312k(5),
    cBR32064k(6),
    cBR44736k(7),
    cBR97728k(8),
    cBR2048k(16),
    cBR8448k(17),
    cBR34368k(18),
    cBR139264k(19),
}

```

Annex B: appq2931ss

```

cBRn64k(128),
cBRn8k(129)
}

Multiplier ::= INTEGER -- (0..65535)
-- ETS 300 443-1 subclause 4.5.5

SCFRMethod ::= ENUMERATED {
-- Source Clock Frequency Recovery Method
-- ETS 300 443-1 subclause 4.5.5

    nullSCFRMethod0(0), -- null
    synchronousResidualTimeStamp(1), -- synchronous residual time stamp
    adaptiveClockMethod(2) -- adaptive clock method
}

ECMethod ::= ENUMERATED {
-- Error correction method
-- ETS 300 443-1 subclause 4.5.5
    nullECMethod(0), -- null
    fwdCorrectionLossSensitive(1), -- a forward error correction method for loss sensitive signal transport
    fwdCorrectionDelaySensitive(2) -- a forward error correction method for delay sensitive signal transport
}

SDTBSIZE ::= INTEGER -- (0..65535)
-- Structured data transfer block size
-- ETS 300 443-1 subclause 4.5.5

PFCMethod ::= INTEGER -- (1..47)
-- Partially filled cells method, all values other than 1-47 reserved
-- ETS 300 443-1 subclause 4.5.5

AALType1ParameterValues ::= SEQUENCE {
-- ETS 300 443-1 subclause 4.5.5
    aAL1Subtype [1] AAL1Subtype,
    cBRRate [2] CBRRate,
    multiplier [3] Multiplier OPTIONAL,
    sCFRMethod [4] SCFRMethod OPTIONAL,
    eCMethod [5] ECMethod OPTIONAL,
    sDTBSIZE [6] SDTBSIZE OPTIONAL,
    pFCMethod [7] PFCMethod OPTIONAL
}

FwdMaxCPCSSDUSize ::= INTEGER -- (0..65535)
-- Forward maximum CPCS SDU size
-- ETS 300 443-1 subclause 4.5.5

BwdMaxCPCSSDUSize ::= INTEGER -- (0..65535)
-- Backward maximum CPCS SDU size
-- ETS 300 443-1 subclause 4.5.5

MIDRangeValue ::= INTEGER -- (0..1023)
-- MID value, 0..1023 only
-- ETS 300 443-1 subclause 4.5.5

MIDRange ::= SEQUENCE {
-- MID range
    lowestMIDRange [1] MIDRangeValue,
    highestMIDRange [2] MIDRangeValue
}

SSCSType ::= ENUMERATED {
-- SSCS type
-- ETS 300 443-1 subclause 4.5.5
    nullSSCSType(0), -- null
    assuredSSCSonSSCOP(1), -- data SSCS based on SSCOP (assured)
    nonAssuredSSCSonSSCOP(2), -- data SSCS based on SSCOP (non-assured)
    frameRelaySSCS(4) -- frame relay SSCS(4)
}

AALType34ParameterValues ::= SEQUENCE {
-- ETS 300 443-1 subclause 4.5.5
    fwdMaxCPCSSDUSize [1] FwdMaxCPCSSDUSize OPTIONAL,
    bwdMaxCPCSSDUSize [2] BwdMaxCPCSSDUSize OPTIONAL,
    mIDRange [3] MIDRange OPTIONAL,
    sSCSType [4] SSCSType OPTIONAL
}

AALType5ParameterValues ::= SEQUENCE {
-- ETS 300 443-1 subclause 4.5.5
    fwdMaxCPCSSDUSize [1] FwdMaxCPCSSDUSize OPTIONAL,
}

```

Annex B: appq2931ss

```

bwdMaxCPCSSDUSize [2] BwdMaxCPCSSDUSize OPTIONAL,
sSCSType [3] SSCSType OPTIONAL
}

AALTypeUserDefinedParameterValues ::= OCTET STRING -- max size 4
-- ETS 300 443-1 subclause 4.5.5

AALTypeVoiceParameterValues ::= NULL

AALType2ParameterValues ::= NULL

--AALParameterValues
ATMAdaptionLayerParameters ::= CHOICE {
-- ETS 300 443-1 subclause 4.5.5
aALTypeVoiceParameterValues [1] AALTypeVoiceParameterValues,
aALType1ParameterValues [2] AALType1ParameterValues,
aALType2ParameterValues [3] AALType2ParameterValues,
aALType34ParameterValues [4] AALType34ParameterValues,
aALType5ParameterValues [5] AALType5ParameterValues,
aALTypeUserDefinedParameterValues [6] AALTypeUserDefinedParameterValues
}

--ATMAdaptionLayerParameters ::= SEQUENCE {
-- ETS 300 443-1 subclause 4.5.5
-- aALType AALType,
-- aALParameterValues AALParameterValues
--}

PCRValue ::= INTEGER -- (0..16777215)
-- forward, backward peak cell rate (0..2**24-1)
-- ETS 300 443-1 subclause 4.5.6

ATMTrafficDescriptor ::= SEQUENCE {
-- ETS 300 443-1 subclause 4.5.6
fwdPeakCellRateCLP0 [1] PCRValue OPTIONAL,
bwdPeakCellRateCLP0 [2] PCRValue OPTIONAL,
fwdPeakCellRateCLP01 [3] PCRValue,
bwdPeakCellRateCLP01 [4] PCRValue
}

BearerClass ::= ENUMERATED {
bCOBA(1), -- BCOB-A
bCOBC(3), -- BCOB-C
bCOBX(16) -- BCOB-X
}

TrafficType ::= ENUMERATED {
nullTrafficType(0), -- no indication
constantBitRate(1), -- constant bit rate
variableBitRate(2) -- variable bit rate
}

TimingRequirements ::= ENUMERATED {
noTimingRequirements(0), -- no indication
endToEndTimingRequirements(1), -- end-to-end timing required
noEndToEndTimingRequirements(2) -- end-to-end timing not required
}

SusceptibilityToClipping ::= ENUMERATED {
isSusceptibleToClipping(0),
notSusceptibleToClipping(1)
}

UserPlaneConnectionConfiguration ::= ENUMERATED {
pointToPoint(0),
pointToMultiPoint(1)
}

BroadbandBearerCapability ::= SEQUENCE {
-- ETS 300 443-1 subclause 4.5.7
bearerClass [1] BearerClass,
trafficType [2] TrafficType OPTIONAL,
timingRequirements [3] TimingRequirements OPTIONAL,
susceptibilityToClipping [4] SusceptibilityToClipping,
userPlaneConnectionConfiguration [5] UserPlaneConnectionConfiguration
}

HighLayerInformationType ::= ENUMERATED {
isoIecHighLayerInformation(0),
userSpecificHighLayerInformation(1),
vendorSpecificApplicationIdentifier(3),
}

```

Annex B: appq2931ss

```

teleserviceHighLayerInformation(4)
}

HighLayerInformation ::= OCTET STRING

BroadbandHighLayerInformation ::= SEQUENCE {
-- ETS 300 443-1 subclause 4.5.8
  highLayerInformationType [1] HighLayerInformationType,
  highLayerInformation [2] HighLayerInformation OPTIONAL
}

UIL2ProtocolId ::= ENUMERATED {
  basicModeISO1745(1),
  iTUTQ921(2),
  iTUTX25LinkLayer(6),
  iTUTX25MultiLink(7),
  iTUTT71(8),
  hDLCARM(9),
  hDLCNRM(10),
  hDLCABM(11),
  LANLogicalLinkControl(12),
  iTUTX75SLP(13),
  iTUTQ922(14),
  iSOIEC7776(17)
}

UIL2PMode ::= ENUMERATED {
  normalOperation(1),
  extendedOperation(2)
}

StandardizedUIL2P ::= SEQUENCE {
  uIL2ProtocolId [1] UIL2ProtocolId,
  uIL2PMode [2] UIL2PMode OPTIONAL,
  -- q933 use must be '00'B all other are reserved
  uIL2PWindowSize [3] UIL2PWindowSize OPTIONAL
  -- only present if uIL2PMode is HDLC
}
UIL2PWindowSize ::= INTEGER -- (1..127)

UserSpecifiedUIL2P ::= SEQUENCE {
  value [1] INTEGER OPTIONAL
}

UserInformationLayer2Protocol ::= CHOICE {
  standardizedUIL2P [1] StandardizedUIL2P,
  userSpecifiedUIL2P [2] UserSpecifiedUIL2P
}

UIL3ProtocolId ::= ENUMERATED {
  iTUTX25PacketLayer(6),
  iSOIEC8208(7),
  iSOIEC8878(8),
  iSOIEC8473(9),
  iTUTT70MinimumNetworkLayer(10),
  iSOIECTR9577(16)
}

UIL3PMode ::= ENUMERATED {
  normalPackSequenceNumbering(1),
  extendedPackSequenceNumbering(2)
}

UIL3PDefaultPacketSize ::= ENUMERATED {
  defaultPacketSize16(4),
  defaultPacketSize32(5),
  defaultPacketSize64(6),
  defaultPacketSize128(7),
  defaultPacketSize256(8),
  defaultPacketSize512(9),
  defaultPacketSize1024(10),
  defaultPacketSize2048(11),
  defaultPacketSize4096(12)
}

UIL3PPacketWindowSize ::= INTEGER -- (1..127)

StandardizedUIL3P ::= SEQUENCE {
  uIL3ProtocolId [1] UIL3ProtocolId,
  uIL3PMode [2] UIL3PMode OPTIONAL,
  -- only present if mode is iTUTX25PacketLayer, iSOIEC8208,
}

```

Annex B: appq2931ss

```

-- ISOIEC8878, ISOIEC8473
uIL3PDefaultPacketSize [3] UIL3PDefaultPacketSize OPTIONAL,
-- only present if mode is iTUTX25PacketLayer,ISOIEC8208,
-- ISOIEC8878, ISOIEC8473
uIL3PPacketWindowSize [4] UIL3PPacketWindowSize OPTIONAL,
-- only present if mode is iTUTX25PacketLayer,ISOIEC8208,
-- ISOIEC8878, ISOIEC8473
additionalLayer3ProtocolInformation [5] OCTET STRING OPTIONAL
-- only present if mode is ISOIECTR9577, max length 8 octets
}

UserSpecifiedUIL3P ::= SEQUENCE {
  value [1] INTEGER OPTIONAL -- (SIZE(7))
}

UserInformationLayer3Protocol ::= CHOICE {
  standardizedUIL3P [1] StandardizedUIL3P,
  userSpecifiedUIL3P [2] UserSpecifiedUIL3P
}

BroadbandLowLayerInformation ::= SEQUENCE {
  -- ETS 300 443-1 subclause 4.5.9
  -- all values off user information layer 1 are reserved
  userInformationLayer2Protocol [1] UserInformationLayer2Protocol OPTIONAL,
  userInformationLayer3Protocol [2] UserInformationLayer3Protocol OPTIONAL
}

BroadbandLowLayerInformations ::= SEQUENCE OF BroadbandLowLayerInformation
-- ETS 300 443-1 Annex C, max 3 times present in setup

BroadbandSendingComplete ::= NULL

CallState ::= ENUMERATED {
  -- ETS 300 443-1 subclause 4.5.10
  nullState(0),
  callInitiated(1),
  overlapSending(2),
  outgoingCallProceeding(3),
  callDelivered(4),
  callPresent(6),
  callReceived(7),
  connectRequest(8),
  incomingCallProceeding(9),
  activeState(10),
  releaseRequest(11),
  releaseIndication(12),
  overlapReceiving(25),
  restartRequest(61),
  restartState(62)
}

TypeOfNumber ::= ENUMERATED {
  unknownTypeOfNumber(0),
  internationalNumber(1),
  nationalNumber(2),
  networkSpecificNumber(3),
  subscriberNumber(4),
  abbreviatedNumber(6)
}

NumberingPlanIdentification ::= ENUMERATED {
  unknownNumberingPlan(0),
  iSDNNumberingPlanE164(1),
  -- nSAPAddressing(2), commented according to ETS 300 443-1
  privateNumberingPlan(33)
}

PartyNumberContents ::= CHOICE {
  numberDigits [1] IA5String,
  nSAPAddressOctets [2] OCTET STRING
  -- only if numberingPlanIdentification is nSAPAddressing
}

CalledPartyNumber ::= SEQUENCE {
  -- ETS 300 443-1 subclause 4.5.11
  typeOfNumber [1] TypeOfNumber,
  numberingPlanIdentification [2] NumberingPlanIdentification,
  contents [3] PartyNumberContents
}

```

Annex B: appq2931ss

```

TypeOfSubaddress ::= ENUMERATED {
  nSAPSubAddressing(0),
  userSpecifiedATMEndSystemAddress(1),
  userSpecifiedSubAddressing(2)
}

OddEvenIndicator ::= BOOLEAN

CalledPartySubaddress ::= SEQUENCE {
  -- ETS 300 443-1 subclause 4.5.12
  typeOfSubAddress [1] TypeOfSubaddress,
  oddEvenIndicator [2] OddEvenIndicator,
  subAddressInformation [3] OCTET STRING
}

PresentationIndicator ::= ENUMERATED {
  presentationAllowed(0),
  presentationRestricted(1),
  numberNotAvailable(2)
}

ScreeningIndicator ::= ENUMERATED {
  userProvidedNotScreened(0),
  userProvidedVerifiedAndPassed(1),
  userProvidedVerifiedAndFailed(2),
  networkProvided(3)
}

PresentationScreening ::= SEQUENCE {
  presentationIndicator [1] PresentationIndicator,
  screeningIndicator [2] ScreeningIndicator
}

CallingPartyNumber ::= SEQUENCE {
  -- ETS 300 443-1 subclause 4.5.13
  typeOfNumber [1] TypeOfNumber,
  numberingPlanIdentification [2] NumberingPlanIdentification,
  presentationScreening [3] PresentationScreening OPTIONAL,
  contents [4] PartyNumberContents
}

CallingPartySubaddress ::= SEQUENCE {
  -- ETS 300 443-1 subclause 4.5.14
  typeOfSubAddress [1] TypeOfSubaddress,
  oddEvenIndicator [2] OddEvenIndicator,
  subAddressInformation [3] OCTET STRING
}

CauseValue ::= INTEGER -- (0..127)

CauseLocation ::= INTEGER -- (0..15)

Cause ::= SEQUENCE {
  -- ETS 300 443-1 subclause 4.5.15
  location [1] CauseLocation,
  causeValue [2] CauseValue,
  diagnostics [3] OCTET STRING OPTIONAL
}

defCause Cause ::= { location 0, causeValue 0 }

Causes ::= SEQUENCE OF Cause
-- max. 2 times in a message

VPAssociatedSignalling ::= ENUMERATED {
  vPAssociated(0),
  explicitVPCIIndication(1)
}

VCIChoice ::= ENUMERATED {
  exclusiveVCI(0),
  anyVCI(1)
}

VPCIVValues ::= INTEGER -- (0..65535)
VCIVValues ::= INTEGER -- (32..65535)

ConnectionIdentifier ::= SEQUENCE {
  -- ETS 300 443-1 subclause 4.5.16
  vPAssociatedSignalling [1] VPAssociatedSignalling,
  VCIChoice [2] VCIChoice,
}

```

Annex B: appq2931ss

```

vPCI  [3] VPCIVValues,
vCI   [4] VCIVValues
}

ERValue ::= INTEGER -- (0..32767)

EndpointReference ::= SEQUENCE {
-- ETS 300 771-1 subclause 8.2.1
eRFlag [1] Flag,
eRValue [2] ERValue
}

EndpointState ::= ENUMERATED {
-- ETS 300 771-1 subclause 8.2.2
nullPartyState(0),
addPartyInitiated(1),
partyAlertingDelivered(4),
addPartyReceived(6),
partyAlertingReceived(7),
dropPartyInitiated(11),
dropPartyReceived(12),
activePartyState(10)
}

CumulativeTransitDelay ::= INTEGER -- (0..65535)
MaximumEndToEndTransitDelay ::= INTEGER -- (0..65535)

EndToEndTransitDelay ::= SEQUENCE {
-- ETS 300 443-1 subclause 4.5.17
cumulativeTransitDelay [1] CumulativeTransitDelay,
maximumEndToEndTransitDelay [2] MaximumEndToEndTransitDelay OPTIONAL
}

NotificationIndicator ::= OCTET STRING
-- ETS 300 443-1 subclause 4.5.23
-- further defined in other standards like Q.932

QualityOfServiceParameter ::= NULL
-- ETS 300 443-1 subclause 4.5.18
-- all other values than unspecified are reserved

ShapingIndicator ::= ENUMERATED {
noRequirementOnShaping(0),
aggregateShaping(1)
}

ComplianceIndicator ::= ENUMERATED {
oAMF5FlowOptional(0),
oAMF5FlowMandatory(1)
}

UserNetworkFaultMgmtIndicator ::= ENUMERATED {
noUserOriginatedFaultMgmtIndications(0),
userOriginatedFaultMgmtIndications(1)
}

OAMF5FlowIndicator ::= ENUMERATED {
percent0(0),
percent01(1),
percent1(4)
}

OAMTrafficDescriptor ::= SEQUENCE {
-- ETS 300 443-1 subclause 4.5.24
shapingIndicator [1] ShapingIndicator,
complianceIndicator [2] ComplianceIndicator,
userNetworkFaultMgmtIndicator [3] UserNetworkFaultMgmtIndicator,
fwdOAMF5FlowIndicator [4] OAMF5FlowIndicator,
bwdOAMF5FlowIndicator [5] OAMF5FlowIndicator
}

RestartIndicator ::= ENUMERATED {
-- ETS 300 443-1 subclause 4.5.19
indicatedVC(0),
allVCControlledBySignallingChannel(1),
allVC(2)
}

TypeOfNetwork ::= ENUMERATED {
userSpecifiedTypeOfNetwork(0),

```

Annex B: appq2931ss

```

nationalNetworkIdentification(2),
internationalNetworkIdentification(3)
}

NetworkIdentificationPlan ::= ENUMERATED {
  unknownNetworkIdentificationPlan(0),
  carrierNetworkCode(1),
  dataNetworkIdentificationCode(3)
}

NetworkIdentification ::= IA5String

TransitNetworkSelection ::= SEQUENCE {
-- ETS 300 443-1 subclause 4.5.22
  typeOfNetwork [1] TypeOfNetwork,
  networkIdentificationPlan [2] NetworkIdentificationPlan,
  networkIdentification [3] NetworkIdentification
}

TransitNetworkSelections ::= SEQUENCE OF TransitNetworkSelection
-- max. 4 times in a setup

GFPData ::= OCTET STRING
emptyGFPData GFPData ::= ''H

Facilities ::= SEQUENCE OF GFPData

PlainParameter ::= OCTET STRING
-- not identified parameters as octet string (starting with IEId)

PlainParameters ::= SEQUENCE OF PlainParameter

InformationElements ::= SEQUENCE {
-- ETS 300 443-1 subclause 4.5
  aTMAdaptionLayerParameters [1] ATMAdaptionLayerParameters OPTIONAL,
  aTMTrafficDescriptor [2] ATMTrafficDescriptor OPTIONAL,
  broadbandBearerCapability [3] BroadbandBearerCapability OPTIONAL,
  broadbandHighLayerInformation [4] BroadbandHighLayerInformation OPTIONAL,
  broadbandLowLayerInformations [5] BroadbandLowLayerInformations OPTIONAL,
  broadbandSendingComplete [6] BroadbandSendingComplete OPTIONAL,
  callState [7] CallState OPTIONAL,
  calledPartyNumber [8] CalledPartyNumber OPTIONAL,
  calledPartySubaddress [9] CalledPartySubaddress OPTIONAL,
  callingPartyNumber [10] CallingPartyNumber OPTIONAL,
  callingPartySubaddress [11] CallingPartySubaddress OPTIONAL,
  causes [12] Causes OPTIONAL,
  connectionIdentifier [13] ConnectionIdentifier OPTIONAL,
  endpointReference [14] EndpointReference OPTIONAL,
  endpointState [15] EndpointState OPTIONAL,
  endToEndTransitDelay [16] EndToEndTransitDelay OPTIONAL,
  notificationIndicator [17] NotificationIndicator OPTIONAL,
  oAMTrafficDescriptor [18] OAMTrafficDescriptor OPTIONAL,
  qualityOfServiceParameter [19] QualityOfServiceParameter OPTIONAL,
  restartIndicator [20] RestartIndicator OPTIONAL,
  transitNetworkSelections [21] TransitNetworkSelections OPTIONAL,
  facilities [22] Facilities OPTIONAL,
  plainParameters [23] PlainParameters OPTIONAL -- not used
}

Q2931ssMessage ::= SEQUENCE {
  msgType [1] MessageType,
  callReference [2] CallReference,
  ies [3] InformationElements
}

-- defInformationElements InformationElements ::= {}

-- defQ2931ssMessage Q2931ssMessage ::= {
--   msgType unknownMsgId,
--   callReference { cRFlag TRUE, cRValue dummyCallReferenceValue },
--   informationElements {}
-- }

Location ::= SEQUENCE {
  locside [1] LocationSide,
  trunkNumber [2] TrunkNumber
}

LocationSide ::= ENUMERATED {

```

Annex B: appq2931ss

```
userProtocolIncoming(0),
userProtocolOutgoing(1),
userApplication(2),
networkProtocolIncoming(3),
networkProtocolOutgoing(4),
networkApplication(5),
noLocation(6)
}

--defLocation Location ::= {
-- locside noLocation,
-- trunknr 0
--}

globalCallReferenceInc CallReference ::= {
  cRFlag TRUE,
  cRValue 0
}

globalCallReferenceOut CallReference ::= {
  cRFlag FALSE,
  cRValue 0
}

dummyCallReferenceValue CRValue ::= 8388607

END
```

Annex B: admq2931ss

```
AdmQ2931ss DEFINITIONS ::=  
-- Description: Data for administration of Q.2931/Q.2971/Q.2932  
-- Author: PT87/ETSI, Nils Fischbeck/HUB  
-- Version: 2.1  
-- LastChange: 20.11.97  
BEGIN  
  
TrunkNumber ::= INTEGER  
  
TypeOfReferencePoint ::= ENUMERATED {  
    tbRefPoint(0),  
    sbRefPoint(1)  
}  
  
SupportedProtocol ::= ENUMERATED {  
    q2931Supported(0),  
    q2971Supported(1)  
}  
  
InitTrunkConfig ::= SEQUENCE {  
    trunkNumber [1] TrunkNumber,  
    typeOfRP [2] TypeOfReferencePoint,  
    supportedProtocol [3] SupportedProtocol  
}  
  
END
```

Annex B: intq2931ss

```

IntQ2931ss DEFINITIONS ::=

-- Description: Internal data for Q.2931/Q.2971/Q.2932 protocol specification
-- Author: PT87/ETSI, Nils Fischbeck/HUB
-- Version: 2.1
-- Last Change: 20.11.97
BEGIN

IMPORTS CallReference, EndpointReference, CauseValue FROM AppQ2931ss;

-- What action if message is wrong?
MsgActionIndicator ::= ENUMERATED {
  msgAIAClearCall(0),
  msgAIDiscardAndIgnore(1),
  msgAIDiscardAndReport(2),
  msgAIReserved(3),
  msgAINotSignificant(4)
}

-- What error does the message have?
MsgErrorDescr ::= ENUMERATED {
  noMsgError(0),
  protocolDiscriminatorError(1),
  messageTooShort(2),
  invalidCallReference(3)
}

-- What action if an information element is wrong?
IEActionIndicator ::= ENUMERATED {
  ieAIAClearCall(0),
  ieAIDiscardAndProceed(1),
  ieAIDiscardProceedAndReport(2),
  ieAIDiscardMsgAndIgnore(3),
  ieAIDiscardMsgAndReport(4),
  ieAIReserved(5),
  ieAINotSignificant(8)
}

-- What error does an information element have?
IEErrorDescr ::= ENUMERATED {
  noIEError(0),
  ieContentError(1)
}

-- Pair of action/error description for each information element.
IEErrorStruct ::= SEQUENCE {
  ieAI [1] IEActionIndicator,
  ieErrorDescr [2] IEErrorDescr
}

-- For information elements which can occur at least two times
-- in a message.
IEErrorStructs ::= SEQUENCE OF IEErrorStruct

-- Error description of a message.
MsgError ::= SEQUENCE {
  msgAI [1] MsgActionIndicator,
  msgErrorDescr [2] MsgErrorDescr,
  aTMAdaptionLayerParametersED [3] IEErrorStruct OPTIONAL,
  aTMTrafficDescriptorED [4] IEErrorStruct OPTIONAL,
  broadbandBearerCapabilityED [5] IEErrorStruct OPTIONAL,
  broadbandHighLayerInformationED [6] IEErrorStruct OPTIONAL,
  broadbandLowLayerInformationsEDs [7] IEErrorStructs OPTIONAL,
  broadbandSendingCompleteED [8] IEErrorStruct OPTIONAL,
  callStateED [9] IEErrorStruct OPTIONAL,
  calledPartyNumberED [10] IEErrorStruct OPTIONAL,
  calledPartySubaddressED [11] IEErrorStruct OPTIONAL,
  callingPartyNumberED [12] IEErrorStruct OPTIONAL,
  callingPartySubaddressED [13] IEErrorStruct OPTIONAL,
  causesEDs [14] IEErrorStructs OPTIONAL,
  connectionIdentifierED [15] IEErrorStruct OPTIONAL,
  endpointReferenceED [16] IEErrorStruct OPTIONAL,
  endpointStateED [17] IEErrorStruct OPTIONAL,
  endToEndTransitDelayED [18] IEErrorStruct OPTIONAL,
  notificationIndicatorED [19] IEErrorStruct OPTIONAL,
  oAMTrafficDescriptorED [20] IEErrorStruct OPTIONAL,
  qualityOfServiceParameterED [21] IEErrorStruct OPTIONAL,
  restartIndicatorED [22] IEErrorStruct OPTIONAL,
  transitNetworkSelectionsEDs [23] IEErrorStructs OPTIONAL,
  facilitiesEDs [24] IEErrorStructs OPTIONAL,
  plainParametersEDs [25] IEErrorStructs OPTIONAL
  -- facility
}

```

Annex B: intq2931ss

```
}

-- supporting data
CRSet ::= SET OF CallReference

ERSet ::= SET OF EndpointReference

AALBitData ::= BIT STRING

-- result of message checking
CheckResultType ::= ENUMERATED {
  ok(0),
  clearCall(1),
  statusAndIgnore(2),
  causeSplitAndOk(3),
  ignore(4),
  dropAck(5),
  clearParty(6),
  statusAndOk(7),
  discardIE(8)
}

-- to convey the result of message checking and the cause values
-- to the party control processes from the call control processes
CheckResultTypeStatus ::= SEQUENCE {
  res [1] CheckResultType,
  cv1 [2] CauseValue OPTIONAL,
  cv2 [3] CauseValue OPTIONAL
}

-- For what purpose is the protocol used?
UsageType ::= ENUMERATED {
  netUsage(0), -- network side, implementation
  userUsage(1), -- user side, implementation
  netSimUsage(2), -- network side, implementation
  userSimUsage(3) -- user side implementation
}

END
```

Annex B: intnem

```
IntNEM DEFINITIONS ::=  
BEGIN  
IMPORTS CalledPartyNumber,  
TypeOfNumber,  
NumberingPlanIdentification,  
PartyNumberContents  
FROM  
AppQ2931ss;  
  
--firstCPN CalledPartyNumber ::= {  
-- typeOfNumber internationalNumber,  
-- numberingPlanIdentification iSDNNumberingPlanE164,  
-- contents numberDigits "1"  
--}  
  
END
```

Annex B: saal

```
SAAL DEFINITIONS ::=  
-- Description: Data for lower interface of SAAL.  
-- Author: PT87/ETSI, Nils Fischbeck/HUB  
-- Version: 2.1  
-- Last Change: 20.11.97  
BEGIN  
  
-- name of PCO will be conveyed in SAAL messages for simulation purposes  
-- e.g. 'L0', 'L1'  
PCOName ::= Charstring  
  
-- the contents of AALData messages will be transmitted in BITSTRING  
-- representation as Charstring e.g. '01010001...'  
PDUContents ::= Charstring  
  
-- data to transmit on lower SAAL interface  
PCOCont ::= SEQUENCE {  
  pco [1] PCOName,  
  cont [2] PDUContents  
}  
  
END
```

Annex B: gfpapdu

```

GFPAPDU DEFINITIONS ::=

-- Description: Data for Q.2932.
-- Author: PT87/ETSI, Nils Fischbeck/HUB
-- Version: 2.1
-- Last Change: 20.11.97
BEGIN

IMPORTS CalledPartyNumber, CallingPartyNumber
FROM AppQ2931ss;

-- When to send error because of no reply?
TimeOutValue ::= Duration

-- to whom to send replies
HandlerId ::= PId

-- data definition for messages to and from ORB,
-- see CORBA 2.1 for details

INVOKE-MESSAGE ::= SEQUENCE
{
  request-header [1] OCTET STRING,
  request-body [2] OCTET STRING,
  invocation-id [3] INTEGER,
  is-one-way [4] BOOLEAN,
  time-out-value [5] TimeOutValue,
  boa-identifier [7] INTEGER
}

RESULT-MESSAGE ::= SEQUENCE
{
  result-header [1] OCTET STRING,
  result-body [2] OCTET STRING,
  invocation-id [3] INTEGER
}

ERROR-MESSAGE ::= SEQUENCE
{
  invocation-id [1] INTEGER,
  cause-value [2] INTEGER
}

REJECT-MESSAGE ::= SEQUENCE
{
  invocation-id [1] INTEGER,
  cause-value [2] INTEGER
}

-- message to register BOA

BOA-REGISTERED-MESSAGE ::= SEQUENCE
{
  boa-id [1] INTEGER,
  boa-handler [2] HandlerId
}

PHONENUMBER ::= CalledPartyNumber

TRANSPORT-QOS ::= ENUMERATED {
  notSpecified(0),
  connectionLessBearerIndependent(1),
  connectionOrientedBearerIndependent(2),
  bearerRelated(3)
}

BINDING-QOS ::= SEQUENCE
{
  transport-characteristic [1] TRANSPORT-QOS
}

-- connect to other BOA

CONNECT-MESSAGE ::= SEQUENCE
{
  client-phonenumbers [1] PHONENUMBER,
  server-phonenumbers [2] PHONENUMBER,
  server-boaid [3] INTEGER,
  binding-characteristics [4] BINDING-QOS
}

```

Annex B: gfpapdu

```
-- GFP APDU definitions

gfpipErrorLocalValue INTEGER ::= 8888
gfpipOperationLocalValue INTEGER ::= 8889

InvokeIDType ::= INTEGER -- (-32768..32767)

OperationOp ::= CHOICE {
  localValue INTEGER,
  globalValue BOOLEAN -- OBJECT IDENTIFIER
}

-- gFPIOPOperation Operation ::= localValue: 8888

ErrorOp ::= CHOICE {
  localValue INTEGER,
  globalValue BOOLEAN -- OBJECT IDENTIFIER
}

-- gFPIOPError ErrorOp ::= localValue: 8889

APDU ::= CHOICE {
  invokeAPDU [1] IMPLICIT InvokeAPDU,
  returnResultAPDU [2] IMPLICIT ReturnResultAPDU,
  returnErrorAPDU [3] IMPLICIT ReturnErrorAPDU,
  rejectAPDU [4] IMPLICIT RejectAPDU
}

Argument ::= INVOKE-MESSAGE

InvokeAPDU ::= SEQUENCE {
  invokeID InvokeIDType,
  linkedId [0] IMPLICIT InvokeIDType OPTIONAL,
  operationOp OperationOp,
  argument Argument
}

Result ::= RESULT-MESSAGE

ResultSeq ::= SEQUENCE {
  operationOp OperationOp,
  result Result
}

ReturnResultAPDU ::= SEQUENCE {
  invokeID InvokeIDType,
  resultSeq ResultSeq OPTIONAL
}

Parameter ::= ERROR-MESSAGE

ReturnErrorAPDU ::= SEQUENCE {
  invokeID InvokeIDType,
  errorOp ErrorOp,
  parameter Parameter
}

InvokeChoice ::= CHOICE {
  invokeID InvokeIDType,
  nullID NULL
}

GeneralProblem ::= INTEGER {
  unrecognizedAPDU (0),
  mistypedAPDU (1),
  badlyStructuredAPDU (2)
}

InvokeProblem ::= INTEGER { -- ROSE-user detected
  duplicateInvocation (0),
  unrecognizedOperation (1),
  mistypedArgument (2),
  resourceLimitation (3),
  initiatorReleasing(4),
  unrecognizedLinkedID(5),
  linkedResponseUnexpected(6),
  unexpectedChildOperation(7)
}

ReturnResultProblem ::= INTEGER {
  unrecognizedInvocationResult (0),
  
```

Annex B: gfpapdu

```

resultResponseUnexpected (1),
mistypedResult (2)
}

ReturnErrorProblem ::= INTEGER {
  unrecognizedInvocationError (0),
  errorResponseUnexpected (1),
  unrecognizedError (2),
  unexpectedError (3),
  mistypedParameter (4)
}

ProblemChoice ::= CHOICE {
  general [0] IMPLICIT GeneralProblem,
  invoke [1] IMPLICIT InvokeProblem,
  returnResult [2] IMPLICIT ReturnResultProblem,
  returnError [3] IMPLICIT ReturnErrorProblem
}

RejectAPDU ::= SEQUENCE {
  invokeID InvokeChoice,
  problem ProblemChoice
}

IncompleteAPDUIInfo ::= SEQUENCE {
  invokeID InvokeIDType OPTIONAL
}

IncompleteAPDU ::= CHOICE {
  incompleteInvoke [1] IncompleteAPDUIInfo,
  incompleteReject [2] IncompleteAPDUIInfo,
  incompleteReturnResult [3] IncompleteAPDUIInfo,
  incompleteReturnError [4] IncompleteAPDUIInfo,
  nullAPDU [5] NULL
}

PartialAPDU ::= CHOICE { -- type to define partial APDU decode function
  incompleteAPDU [1] IncompleteAPDU,
  apdu [2] APDU
}

GFPInvokeMessage ::= SEQUENCE {
  invoke [1] InvokeAPDU,
  calledPartyNumber [2] CalledPartyNumber OPTIONAL,
  callingPartyNumber [3] CallingPartyNumber OPTIONAL,
  callReference [4] CallReference OPTIONAL
}

GFPResultMessage ::= SEQUENCE {
  result [1] ReturnResultAPDU,
  calledPartyNumber [2] CalledPartyNumber OPTIONAL,
  callingPartyNumber [3] CallingPartyNumber OPTIONAL,
  callReference [4] CallReference OPTIONAL
}

GFPErrorMessage ::= SEQUENCE {
  errorVal [1] ReturnErrorAPDU,
  calledPartyNumber [2] CalledPartyNumber OPTIONAL,
  callingPartyNumber [3] CallingPartyNumber OPTIONAL,
  callReference [4] CallReference OPTIONAL
}

GFPRejectMessage ::= SEQUENCE {
  reject [1] RejectAPDU,
  calledPartyNumber [2] CalledPartyNumber OPTIONAL,
  callingPartyNumber [3] CallingPartyNumber OPTIONAL,
  callReference [4] CallReference OPTIONAL
}

END

```

Annex B: AdmAppQ2931ssInterface

```
USE AppQ2931ss;
USE AdmAppQ2931ssInterface;
USE AdmQ2931ss;
```

Package AdmAppQ2931ssInterface

1(1)

```
/*
This package defines signals and types for the
administration of the application interface of Q.2931.

Author: PT87/ETSI, Nils Fischbeck/HUB
Version: 2.1
Last Change: 20.11.97
*/
```

```
SIGNAL
/* inform application about protocol PId for trunk */
init_AppQ2931ss_req(InitTrunkConfig,PId),

/* sent as answer to NEM with application PId */
init_AppQ2931ss_cnf(PId),

/* sent to application to announce party available at a trunk */
add_party_to_trunk(CalledPartyNumber,TrunkNumber),

/* sent to application to cause restart to be sent at trunk */
netapp_restart_req(TrunkNumber),

/* clear all informations about calls (but do not release) */
netapp_reset_calls,

/* send to application after all trunk initialisations are done */
init_trunks_done;
```

```
SIGNALLIST AdmAppQ2931ssSLDown =
/* from NEM to application */
init_AppQ2931ss_req,
netapp_restart_req,
add_party_to_trunk,
netapp_reset_calls,
init_trunks_done;

SIGNALLIST AdmAppQ2931ssSLUp =
/* from application to NEM */
init_AppQ2931ss_cnf;
```

Annex B: AppQ2931ssInterface

```
USE AppQ2931ss;
USE AdmQ2931ss;
USE GFPAPDU;
```

Package AppQ2931ssInterface

1(6)

```
/*
This package defines signals and types for the
application interface of Q.2931.

Author: PT87/ETSI, Nils Fischbeck/HUB
Version: 2.1
Last Change: 20.11.97
*/
```

```
SIGNALLIST AppQ2931ssSLDown =
/* from application to protocol */
(APCoord_Down), (APOCC_Down),
(APICC_Down), (APIOR_Down);

SIGNALLIST AppQ2931ssSLUp =
/* from protocol to application */
(APCoord_Up), (APOCC_Up),
(APICC_Up), (APIOR_Up);
```

Annex B: AppQ2931ssInterface

```
USE AppQ2931ss;
USE AdmQ2931ss;
USE GFPAPDU;
```

Package AppQ2931ssInterface

2(6)

```
SIGNALIST UCSLDown =
/* from env to user side application */
SETUP_PARTY1,
SETUP_PARTY1_EREF0 ,
ADD_PARTY2 ,
DROP_LAST_PARTY ,
DROP_PARTY1 ,
DROP_PARTY2 ,
CONNECT_PARTY1 ,
CONNECT_PARTY2 ,
ACCEPT_PARTY2_REQUEST ,
REJECT_PARTY2_REQUEST ,
DROP_ALL_PARTIES ,
CAUSE_NOTIFY,
DISPLAY;
/* in the current test suite only DISPLAY is used
to cause the sending of user side messages */
```

```
SIGNAL
/* coordination messages */
/* cause the sending of user application messages */
/* see EN 300 771-4 */
SETUP_PARTY1,
SETUP_PARTY1_EREF0 ,
ADD_PARTY2 ,
DROP_LAST_PARTY ,
DROP_PARTY1 ,
DROP_PARTY2 ,
CONNECT_PARTY1 ,
CONNECT_PARTY2 ,
ACCEPT_PARTY2_REQUEST ,
REJECT_PARTY2_REQUEST ,
DROP_ALL_PARTIES ,
CAUSE_NOTIFY,
DISPLAY(Charstring);
```

Annex B: AppQ2931ssInterface

```
USE AppQ2931ss;
USE AdmQ2931ss;
USE GFPAPDU;
```

Package AppQ2931ssInterface

3(6)

```
SIGNAL
/* Signals for interface to call control coordinator */
/* for process creation */

/* to application: create Outgoing Call Control */
create_apocc_req(CallReference),
create_apocc_cnf(PId),

/* to call control coordinator: create Incoming Call Control */
create_icc_req(CallReference),
create_icc_cnf(PId),
/* can not create ICC because SAAL connection not established */
create_icc_rej,

/* delete Call Control */
delete_icc_req(CallReference,TrunkNumber),
delete_occ_req(CallReference,TrunkNumber),

/* to call control coordinator: create restart process */
create_restart_req,
create_restart_cnf(PId),

/* to application: first message to setup has no er for this trunk */
first_msg_no_er(TrunkNumber),

/* kill OCC or ICC and dependend PCs without confirm */
kill_call;
```

Annex B: AppQ2931ssInterface

USE AppQ2931ss;
 USE AdmQ2931ss;
 USE GFPAPDU;

Package AppQ2931ssInterface

4(6)

```
SIGNAL
/* from application to protocol processes */
setup_req(Q2931ssMessage),
call_proceeding_req(Q2931ssMessage),
alerting_req(Q2931ssMessage),
connect_req(Q2931ssMessage),
connect_acknowledge_req(Q2931ssMessage),
release_req(Q2931ssMessage),
release_complete_req(Q2931ssMessage),
status_enquiry_req(Q2931ssMessage),
status_req(Q2931ssMessage),
restart_req(Q2931ssMessage),
restart_acknowledge_req(Q2931ssMessage),
/* CS2 mesages */
add_party_req(Q2931ssMessage),
add_party_reject_req(Q2931ssMessage),
add_party_acknowledge_req(Q2931ssMessage),
party_alerting_req(Q2931ssMessage),
drop_party_req(Q2931ssMessage),
drop_party_acknowledge_req(Q2931ssMessage),
notify_req(Q2931ssMessage),
gfp_invoke_req(GFPInvokeMessage),
gfp_result_req(GFPResultMessage),
gfp_error_req(GFPErrorMessage),
gfp_reject_req(GFPRejectMessage),
facility_req(Q2931ssMessage);
```

```
SIGNAL
/* from protocol processes to application */
setup_ind(Location,Q2931ssMessage),
call_proceeding_ind(Location,Q2931ssMessage),
alerting_ind(Location,Q2931ssMessage),
connect_ind(Location,Q2931ssMessage),
connect_acknowledge_ind(Location,Q2931ssMessage),
release_ind(Location,Q2931ssMessage),
release_complete_ind(Location,Q2931ssMessage),
status_enquiry_ind(Location,Q2931ssMessage),
status_ind(Location,Q2931ssMessage),
restart_ind(Location,Q2931ssMessage),
restart_acknowledge_ind(Location,Q2931ssMessage),
/* CS2 mesages */
add_party_ind(Location,Q2931ssMessage),
add_party_reject_ind(Location,Q2931ssMessage),
add_party_acknowledge_ind(Location,Q2931ssMessage),
party_alerting_ind(Location,Q2931ssMessage),
drop_party_ind(Location,Q2931ssMessage),
drop_party_acknowledge_ind(Location,Q2931ssMessage),
notify_ind(Location,Q2931ssMessage),
gfp_invoke_ind(GFPInvokeMessage),
gfp_result_ind(GFPResultMessage),
gfp_error_ind(GFPErrorMessage),
gfp_reject_ind(GFPRejectMessage),
facility_ind(Location,Q2931ssMessage);
```

Annex B: AppQ2931ssInterface

```
USE AppQ2931ss;
USE AdmQ2931ss;
USE GFPAPDU;
```

Package AppQ2931ssInterface

5(6)

```
SIGNALLIST APCoord_Up =
/* from call control coordinator to application */
create_apocc_req,
create_icc_cnf,
create_restart_cnf,
delete_icc_req,
delete_occ_req,
create_icc_rej,
(GFPSLUp);

SIGNALLIST APCoord_Down =
/* from application to call control coordinator */
create_apocc_cnf,
create_restart_req,
create_icc_req,
(GFPSLDown);
```

```
SIGNALLIST GFPSLUp =
/* from call control coordinator to gfp application */
gfp_invoke_ind,
gfp_result_ind,
gfp_error_ind,
gfp_reject_ind,
facility_ind;

SIGNALLIST GFPSLDown =
/* from gfp application to call control coordinator */
gfp_invoke_req,
gfp_result_req,
gfp_error_req,
gfp_reject_req,
facility_req;
```

```
SIGNALLIST APIOR_Up =
/* from restart to call control coordinator */
restart_ind,
restart_acknowledge_ind;

SIGNALLIST APIOR_Down =
/* from call control coordinator to restart */
restart_req,
restart_acknowledge_req;
```

Annex B: AppQ2931ssInterface

```
USE AppQ2931ss;
USE AdmQ2931ss;
USE GFPAPDU;
```

Package AppQ2931ssInterface

6(6)

```
SIGNALIST APICC_Down=
/* from application to incoming call control process */
setup_req,
connect_acknowledge_req,
release_req,
release_complete_req,
add_party_req,
drop_party_req,
drop_party_acknowledge_req,
notify_req,
gfp_invoke_req,
gfp_result_req,
gfp_error_req,
gfp_reject_req,
facility_req,
status_enquiry_req,
kill_call;

SIGNALIST APICC_Up=
/* from incoming call control process to application */
alerting_ind,
connect_ind,
release_ind,
call_proceeding_ind,
release_complete_ind,
party_alerting_ind,
add_party_reject_ind,
add_party_acknowledge_ind,
drop_party_ind,
drop_party_acknowledge_ind,
notify_ind,
gfp_invoke_ind,
gfp_result_ind,
gfp_error_ind,
gfp_reject_ind,
facility_ind,
first_msg_no_er;
```

```
SIGNALIST APOCC_Up=
/* from outgoing call control process to application */
setup_ind,
connect_acknowledge_ind,
release_ind,
release_complete_ind,
add_party_ind,
drop_party_ind,
notify_ind,
drop_party_acknowledge_ind,
gfp_invoke_ind,
gfp_result_ind,
gfp_error_ind,
gfp_reject_ind,
facility_ind;

SIGNALIST APOCC_Down=
/* from application to outgoing call control */
call_proceeding_req,
release_complete_req,
alerting_req,
party_alerting_req,
connect_req,
release_req,
release_complete_req,
add_party_reject_req,
add_party_acknowledge_req,
drop_party_req,
drop_party_acknowledge_req,
notify_req,
gfp_invoke_req,
gfp_result_req,
gfp_error_req,
gfp_reject_req,
facility_req,
status_enquiry_req,
kill_call;
```

Annex B: AdmQ2931ssInterface

```
USE AdmQ2931ss;
USE AppQ2931ss;
```

Package AdmQ2931ssInterface

1(1)

/*
This package defines signals and types for the administration of the lower interface of Q.2931.
Author: PT87/ETSI, Nils Fischbeck/HUB
Version: 2.1
Last Change: 20.11.97
*/

```
SIGNALLIST AdmQ2931ssSLDown =  
/* from NEM to protocol */  
init_Q2931ss_req,  
reset_Q2931ss_req,  
init_AppQ2931ss_notify,  
reset_call_req;  
  
SIGNALLIST AdmQ2931ssSLUp =  
/* from protocol to NEM */  
init_Q2931ss_cnf,  
trunk_understands_no_ptmp,  
reset_Q2931ss_cnf,  
reset_call_cnf,  
no_aal_connection;
```

```
SIGNAL  
/* from NEM to protocol with SAAL PId for trunk */  
init_Q2931ss_req(InitTrunkConfig,PId),  
  
/* back with protocol pid */  
init_Q2931ss_cnf(PId),  
  
/* from NEM with application PId for trunk */  
init_AppQ2931ss_notify(PId,PId), /* appPartner, gfpPartner */  
  
/* to NEM if trunk does not support ER */  
/* currently not used, */  
/* use first_msg_no_er to application instead */  
trunk_understands_no_ptmp(TrunkNumber),  
  
/* reset a trunk (currently without sending RELEASE) */  
/* by sending this to a call control coordination process of */  
/* of a trunk */  
reset_Q2931ss_req,  
reset_Q2931ss_cnf,  
  
/* reset a particular call (send to a call control coordination) */  
/* without sending RELEASE */  
reset_call_req(CallReference),  
reset_call_cnf,  
  
/* send to NEM if SAAL cannot establish connection */  
/* after certain time */  
no_aal_connection(TrunkNumber);
```

Annex B: Q2931ssInterface

```
USE AppQ2931ss;
USE Q2931ss;
```

Package Q2931ssInterface

1(4)

```
/*
This package defines signals and types for the
lower interface of Q.2931.
```

```
Author: PT87/ETSI, Nils Fischbeck/HUB
Version: 2.1
Last Change: 20.11.97
```

```
*/
```

```
SIGNAL /* AAL signals */
AAL_ESTABLISH_IND,
AAL_ESTABLISH_REQ,
AAL_ESTABLISH_CNF,
AAL_RELEASE_IND(AALRelease),
AAL_RELEASE_REQ,
AAL_RELEASE_CNF,
AAL_DATA_IND(AALData),
AAL_DATA_REQ(AALData),
AAL_UNIDATA_REQ(AALUnitdata),
AAL_UNIDATA_IND(AALUnitdata);
```

Annex B: Q2931ssInterface

```
USE AppQ2931ss;
USE Q2931ss;
```

Package Q2931ssInterface

2(4)

```
SIGNAL
/* experimental: messages exposed to env for generating ATS with Autolink */
SETUP_val_req(Q2931ssMessage),
SETUP_val_ind(Q2931ssMessage),
CALL_PROCEEDING_val_req(Q2931ssMessage),
CALL_PROCEEDING_val_ind(Q2931ssMessage),
ALERTING_val_req(Q2931ssMessage),
ALERTING_val_ind(Q2931ssMessage),
CONNECT_val_req(Q2931ssMessage),
CONNECT_val_ind(Q2931ssMessage),
CONNECT_ACKNOWLEDGE_val_req(Q2931ssMessage),
CONNECT_ACKNOWLEDGE_val_ind(Q2931ssMessage),
RELEASE_val_req(Q2931ssMessage),
RELEASE_val_ind(Q2931ssMessage),
RELEASE_COMPLETE_val_req(Q2931ssMessage),
RELEASE_COMPLETE_val_ind(Q2931ssMessage),
STATUS_val_req(Q2931ssMessage),
STATUS_val_ind(Q2931ssMessage),
STATUS_ENQUIRY_val_req(Q2931ssMessage),
STATUS_ENQUIRY_val_ind(Q2931ssMessage),
RESTART_val_req(Q2931ssMessage),
RESTART_val_ind(Q2931ssMessage),
RESTART_ACKNOWLEDGE_val_req(Q2931ssMessage),
RESTART_ACKNOWLEDGE_val_ind(Q2931ssMessage),
NOTIFY_val_req(Q2931ssMessage),
NOTIFY_val_ind(Q2931ssMessage),
UNKNOWN_val_ind(Q2931ssMessage),
FACILITY_val_ind(Q2931ssMessage),
FACILITY_val_req(Q2931ssMessage);
```

```
SIGNAL
/* experimental: messages exposed to env for generating ATS with Autolink */
PARTY_ALERTING_val_ind(Q2931ssMessage),
PARTY_ALERTING_val_req(Q2931ssMessage),
ADD_PARTY_val_req(Q2931ssMessage),
ADD_PARTY_val_ind(Q2931ssMessage),
ADD_PARTY_ACKNOWLEDGE_val_req(Q2931ssMessage),
ADD_PARTY_ACKNOWLEDGE_val_ind(Q2931ssMessage),
ADD_PARTY_REJECT_val_req(Q2931ssMessage),
ADD_PARTY_REJECT_val_ind(Q2931ssMessage),
DROP_PARTY_val_req(Q2931ssMessage),
DROP_PARTY_val_ind(Q2931ssMessage),
DROP_PARTY_ACKNOWLEDGE_val_req(Q2931ssMessage),
DROP_PARTY_ACKNOWLEDGE_val_ind(Q2931ssMessage);
```

Annex B: Q2931ssInterface

```
USE AppQ2931ss;
USE Q2931ss;
```

Package Q2931ssInterface

3(4)

```
SIGNALLIST Q2931ssSLDown =
/* from call control coordinator to env */
AAL_ESTABLISH_REQ,
AAL_RELEASE_REQ,
AAL_DATA_REQ,
AAL_UNIDATA_REQ,
SETUP_val_req,
ADD_PARTY_val_req,
DROP_PARTY_val_req,
DROP_PARTY_ACKNOWLEDGE_val_req,
CONNECT_ACKNOWLEDGE_val_req,
RELEASE_val_req,
RELEASE_COMPLETE_val_req,
STATUS_ENQUIRY_val_req,
STATUS_val_req,
NOTIFY_val_req,
FACILITY_val_req,
PARTY_ALERTING_val_req,
CALL_PROCEEDING_val_req,
ALERTING_val_req,
ADD_PARTY_ACKNOWLEDGE_val_req,
ADD_PARTY_REJECT_val_req,
CONNECT_val_req,
RESTART_val_req,
RESTART_ACKNOWLEDGE_val_req;
```

Annex B: Q2931ssInterface

```
USE AppQ2931ss;
USE Q2931ss;
```

Package Q2931ssInterface

4(4)

```
SIGNALLIST Q2931ssSLUp =
/* from env to call control coordinator */
AAL_ESTABLISH_IND,
AAL_ESTABLISH_CNF,
AAL_RELEASE_IND,
AAL_RELEASE_CNF,
AAL_DATA_IND,
AAL_UNIDATA_IND,
SETUP_val_ind,
ADD_PARTY_val_ind,
ADD_PARTY_REJECT_val_ind,
ADD_PARTY_ACKNOWLEDGE_val_ind,
CONNECT_ACKNOWLEDGE_val_ind,
ALERTING_val_ind,
CALL_PROCEEDING_val_ind,
PARTY_ALERTING_val_ind,
CONNECT_val_ind,
RELEASE_val_ind,
RELEASE_COMPLETE_val_ind,
DROP_PARTY_val_ind,
DROP_PARTY_ACKNOWLEDGE_val_ind,
STATUS_ENQUIRY_val_ind,
STATUS_val_ind,
NOTIFY_val_ind,
RESTART_val_ind,
RESTART_ACKNOWLEDGE_val_ind,
FACILITY_val_ind;
```

Annex B: AdmSAALInterface

```
USE AdmQ2931ss;
```

Package AdmSAALInterface

1(1)

```
/*
This package defines signals and types for the
administration of the SAAL
(abstraction of the signalling ATM abstraction layer for
simulation purposes).
```

```
Author: PT87/ETSI, Nils Fischbeck/HUB
Version: 2.1
Last Change: 20.11.97
```

```
*/
```

```
SIGNAL
/* from NEM to SAAL with ATMCard PId to establish trunk connection */
init_SAAL_req(InitTrunkConfig,PId);

/* back with SAAL PId for trunk */
init_SAAL_cnf(PId);

/* from NEM to SAAL with protocol PId */
init_Q2931ss_notify(PId);
```

```
SIGNALLIST AdmSAALSLDown =
/* from NEM to SAAL */
init_SAAL_req,
init_Q2931ss_notify;

SIGNALLIST AdmSAALSLUp =
/* from SAAL to NEM */
init_SAAL_cnf;
```

Annex B: SAALInterface

```
USE SAAL;
USE Q2931ssInterface;
```

Package SAALInterface

1(2)

```
/*
This package defines signals and types for the
SAAL
(abstraction of the signalling ATM abstraction layer for
simulation purposes).
```

```
Author: PT87/ETSI, Nils Fischbeck/HUB
Version: 2.1
Last Change: 20.11.97
```

```
*/
```

SIGNAL

```
/* message of lower interface of SAAL */
/* equal to PDU/ASPs of EN 300 771-4..8 */
AAL_EST_RQ(PCOName),
AAL_EST_IN(PCOName),
AAL_EST_CO(PCOName),
AAL_REL_RQ(PCOName),
AAL_REL_IN(PCOName),
AAL_REL_CO(PCOName),
DSS2_PDU(PCOCont);
```

Annex B: SAALInterface

```
USE SAAL;
USE Q2931ssInterface;
```

Package SAALInterface

2(2)

```
SIGNALLIST SAALSLUp =
/* from ATMCARD to SAAL */
/* even validation messages for Autolink ATS generation */
AAL_EST_IN, AAL_REL_IN,
DSS2_PDU,
SETUP_val_ind,
ADD_PARTY_val_ind,
ADD_PARTY_REJECT_val_ind,
ADD_PARTY_ACKNOWLEDGE_val_ind,
CONNECT_ACKNOWLEDGE_val_ind,
ALERTING_val_ind,
CALL_PROCEEDING_val_ind,
PARTY_ALERTING_val_ind,
CONNECT_val_ind,
RELEASE_val_ind,
RELEASE_COMPLETE_val_ind,
DROP_PARTY_val_ind,
DROP_PARTY_ACKNOWLEDGE_val_ind,
STATUS_ENQUIRY_val_ind,
STATUS_val_ind,
NOTIFY_val_ind,
RESTART_val_ind,
RESTART_ACKNOWLEDGE_val_ind,
FACILITY_val_ind;

SIGNALLIST SAALSLDown =
/* from SAAL to ATMCARD */
AAL_EST_IN, AAL_REL_IN,
AAL_EST_CO, AAL_REL_CO,
DSS2_PDU, SETUP_val_req,
ADD_PARTY_val_req,
DROP_PARTY_val_req,
DROP_PARTY_ACKNOWLEDGE_val_req,
CONNECT_ACKNOWLEDGE_val_req,
RELEASE_val_req,
RELEASE_COMPLETE_val_req,
STATUS_ENQUIRY_val_req,
STATUS_val_req,
NOTIFY_val_req,
FACILITY_val_req,
PARTY_ALERTING_val_req,
CALL_PROCEEDING_val_req,
ALERTING_val_req,
ADD_PARTY_ACKNOWLEDGE_val_req,
ADD_PARTY_REJECT_val_req,
CONNECT_val_req,
RESTART_val_req,
RESTART_ACKNOWLEDGE_val_req;
```

Annex B: AdmGFPInterface

```
USE AdmQ2931ss;
USE AdmAppQ2931ssInterface;
```

Package AdmGFPInterface

1(1)

```
/*
This package defines signals and types for the
administration of the GFP application (Q.2932).

Author: PT87/ETSI, Nils Fischbeck/HUB
Version: 2.1
Last Change: 20.11.97

*/
```

```
SIGNAL
/* From NEM to GFP application with protocol PId for trunk */
init_gfpiop_req(InitTrunkConfig,PId);

/* back with GFP application PId */
init_gfpiop_cnf(PId);
```

```
SIGNALLIST AdmGFPSLUp =
/* from NEM to GFP application */
init_gfpiop_cnf;

SIGNALLIST AdmGFPSLDown =
/* from GFP application to NEM */
init_gfpiop_req,
netapp_restart_req;
```

Annex B: SDTCoding

```
USE GFPAPDU;
USE AppQ2931ss;
```

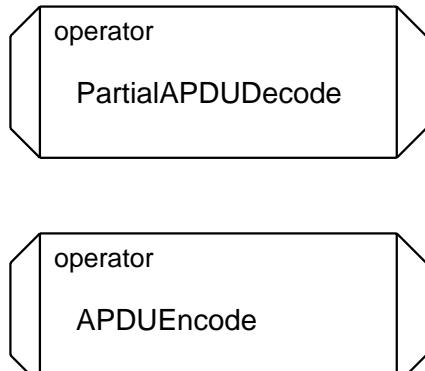
Package SDTCoding

1(1)

```
/*
This package defines operators used for encoding/decoding of GFP
values (incomplete).
These operations are never used and are only to satisfy the
SDT analyzer.
```

```
Author: PT87/ETSI, Nils Fischbeck/HUB
Version: 2.1
Last Change: 20.11.97
```

```
*/
```



```
NEWTYPE APDUCoding
OPERATORS
PartialAPDUDecode : GFPData -> PartialAPDU;
APDUEncode : APDU -> GFPData;
Operator PartialAPDUDecode REFERENCED;
Operator APDUEncode REFERENCED;
ENDNEWTYPE;
```

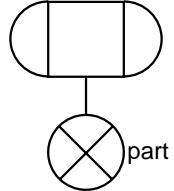
Annex B: PartialAPDUDecode

Operator PartialAPDUDecode

1(1)

```
:FPAR oct GFPData;  
RETURNS PartialAPDU;
```

```
DCL part PartialAPDU;
```



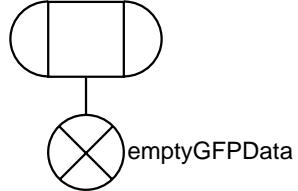
```
/* part!incompleteAPDU!incompleteReject!invokedID!localValue := 0 */  
/* dummy value because, operator is never used from  
this package */
```

Annex B: APDUEncode

Operator APDUEncode

1(1)

```
:FPAR apdu APDU;  
RETURNS GFPData;
```



Annex B: Q2931ssTypes

```
USE GFPAPDU;
USE AdmGFPIInterface;
USE SDTCoding;
```

Package Q2931ssTypes

1(1)

```
/*
This package defines block and process types
for a specification of B-ISDN DSS2 (Q.2931)
with point to multipoint extension (Q.2971)
and GFP (Q.2932).
This version does contain the network
side and user side.

Author: PT87/ETSI, Nils Fischbeck/HUB
Version: 2.1
Last Change: 20.11.97
*/
```

Q2931ssTypes_BT

Annex B: Q2931ssTypes_BT

Block Type Q2931ssTypes_BT

1(5)

/* This Package contains four different kind of block types:
application block types, protocol block types, SAAL block types,
administration block types.

application block types: NetAP_BT, UserAP_BT, GFPAP_BT
for applications at the network side, user side and network side GFP only.

protocol block types: Q2931ss_BT (basic definition of Q.2931/71 processes,
Q2932_BT (Q.2932 signals only).
Derived from Q2931ss_BT are Q2931ssUser_BT, Q2931ssNet_BT, Q2931ssUserSim_BT,
Q2931ssNetSim_BT, Q2931ssNetVal_BT which implement additional behaviour for
implementation of the user side, implementation of the network side,
simulation of the user side, simulation of the network side, automatic
test case generation for the network side.

SAAL block types: SAAL_BT and VALSAAL_BT for adapting protocol messages
to the simulation of the ATS and for adapting protocol messages for automatic
test case generation. Please notice that no original SAAL functionality
is specified here (Q.2921) because it is not the scope of this specification.

administration block types: NEM_BT (Network Element Manager) to build
stacks of application, protocol, SAAL for each trunk. A trunk is like
an ATM network card connected to another ATM network card of the Terminal.
For each such network card in the switch a separate stack of SAAL, protocol, application
must be build. In this specification the application process is shared between
different trunks although provision has been made that each trunk gets
its own application process.

How many trunks are established depends from how often buildStack
in NEM_PT is called (with different trunk numbers).

Examples how to connect these different kind of block types are
in package Q2931ssTest. */

Annex B: Q2931ssTypes_BT

Block Type Q2931ssTypes_BT

2(5)

```
Synonym statusEnqNonActiveParties Boolean = False;  
/* check parties not in active state with status enquiry in  
case of AAL establish confirm */
```

NetAP_BT

GFPAP_BT

UserAP_BT

Q2931ss_BT

Q2932_BT

Q2931ssNet_BT

Q2931ssUser_BT

Q2931ssNetSim_BT

Q2931ssUserSim_BT

Q2931ssNetVal_BT

VALSAAL_BT

SAAL_BT

NEM_BT

Annex B: Q2931ssTypes_BT

Block Type Q2931ssTypes_BT

3(5)

```
/* remote procedure definition such that information can  
be transferred over the boundary of the defined block types. */
```

```
/* what restart state is the restart process in */  
REMOTE PROCEDURE whatRestartState;returns CallState;
```

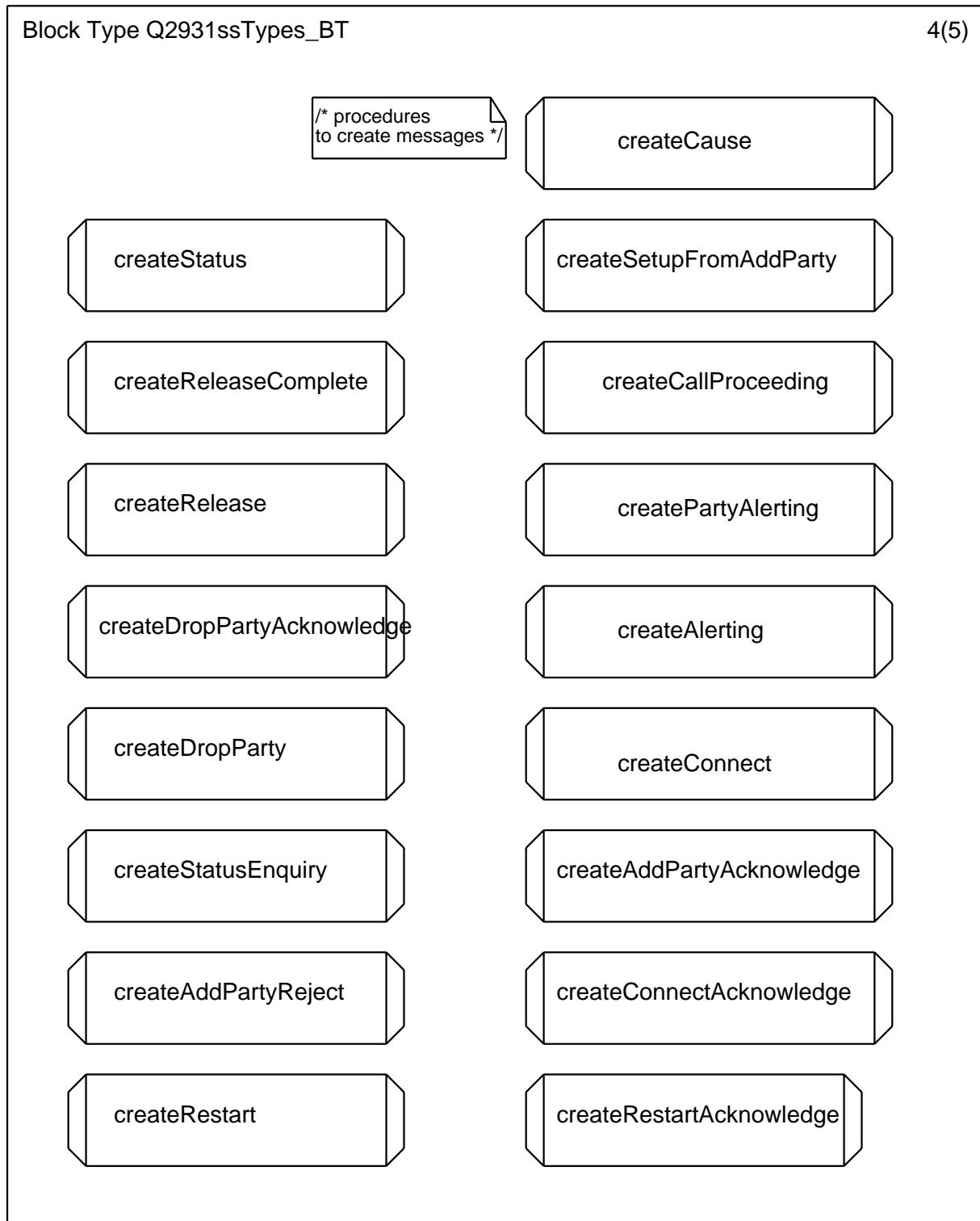
```
/* what state is the call control process in */  
REMOTE PROCEDURE whatLinkState;returns CallState;
```

```
/* is there any other party active in this call (to call control process) */  
REMOTE PROCEDURE otherPartyActive;fpar EndpointReference;returns Boolean;
```

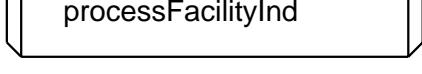
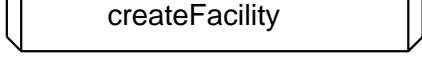
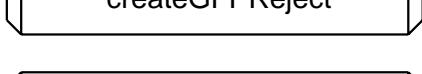
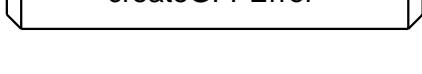
```
/* what party state is the party control process in */  
REMOTE PROCEDURE whatPartyState;returns EndpointState;
```

```
/* what party state is a party in (to party control process) */  
REMOTE PROCEDURE whatPartyStateOfLink;FPAR EndpointReference;RETURNS EndpointState;
```

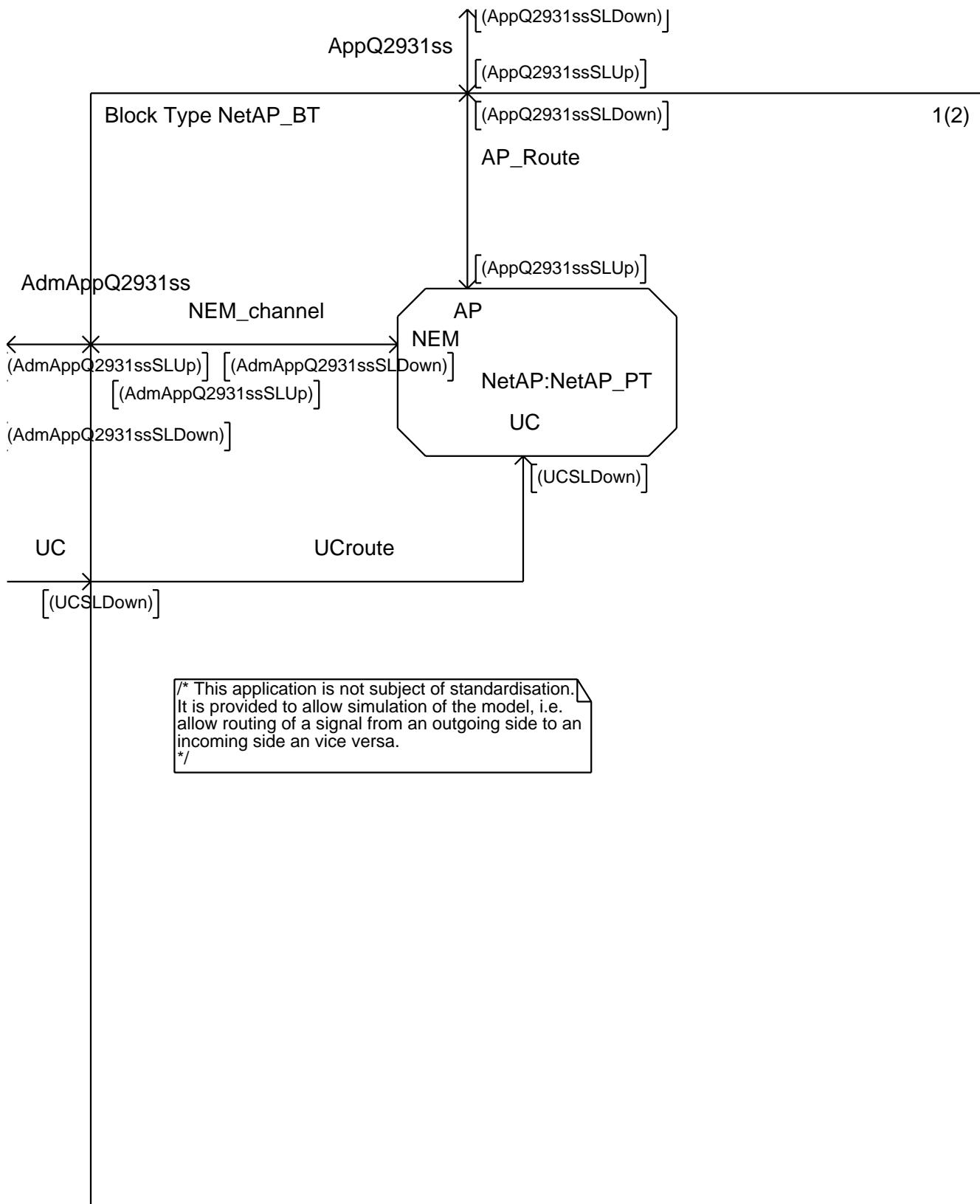
Annex B: Q2931ssTypes_BT



Annex B: Q2931ssTypes_BT

Block Type Q2931ssTypes_BT	5(5)
 processGFPResult	/* procedures to create messages even from other messages in case of "process..."
 processGFPIvoke	
 processGFPReject	
 processGFPError	
 processFacilityInd	
 createFacility	
 createGFPIvoke	
 createGFPResult	
 createGFPReject	
 createGFPError	

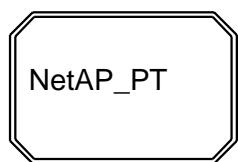
Annex B: NetAP_BT



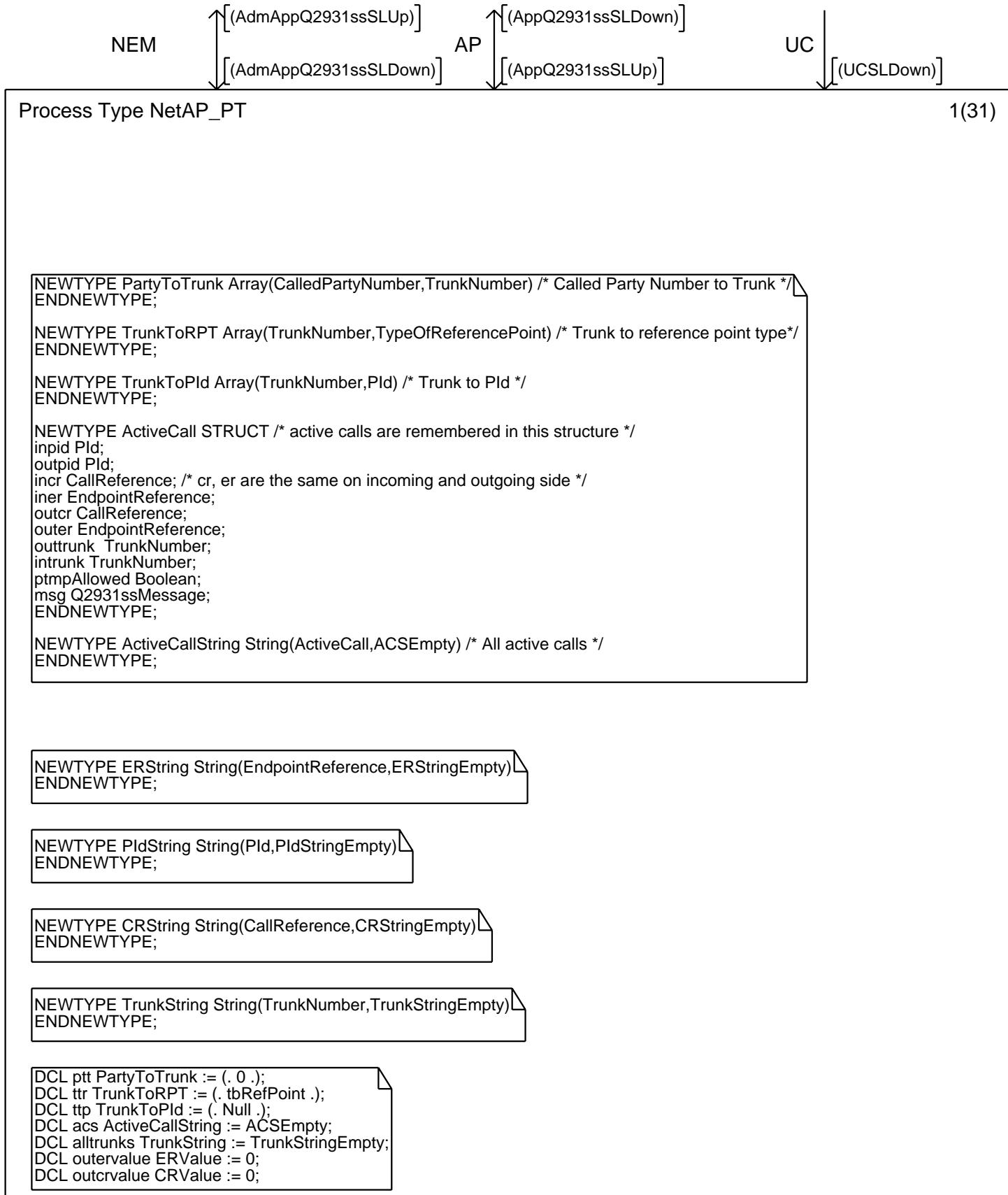
Annex B: NetAP_BT

Block Type NetAP_BT

2(2)



Annex B: NetAP_PT



Annex B: NetAP_PT

Process Type NetAP_PT

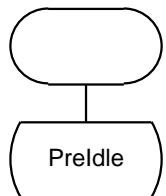
2(31)

```
/* Variable valid only during one transition */
DCL msg Q2931ssMessage;
DCL lastsetup Q2931ssMessage;
DCL tmpmsg Q2931ssMessage;
DCL outmsg Q2931ssMessage;
DCL outmsgClear Q2931ssMessage;
DCL outmsgDropAck Q2931ssMessage;
DCL config InitTrunkConfig;
DCL cr CallReference;
DCL er EndpointReference;
DCL pid PId;
DCL gloc Location;
DCL lastSender PId;
DCL intrunk TrunkNumber;
DCL trunkNumber TrunkNumber;
DCL atrunk TrunkNumber;
DCL party CalledPartyNumber;
DCL pidstr PIdString;
DCL killpidstr PIdString;
DCL erstr ERString;
DCL crstr CRString;
DCL index Integer;
DCL allnum Integer;
DCL outcr CallReference;
DCL outer EndpointReference;
DCL incr CallReference;
DCL iner EndpointReference;
DCL outrunk TrunkNumber;
DCL str Charstring;
```

Annex B: NetAP_PT

Process Type NetAP_PT

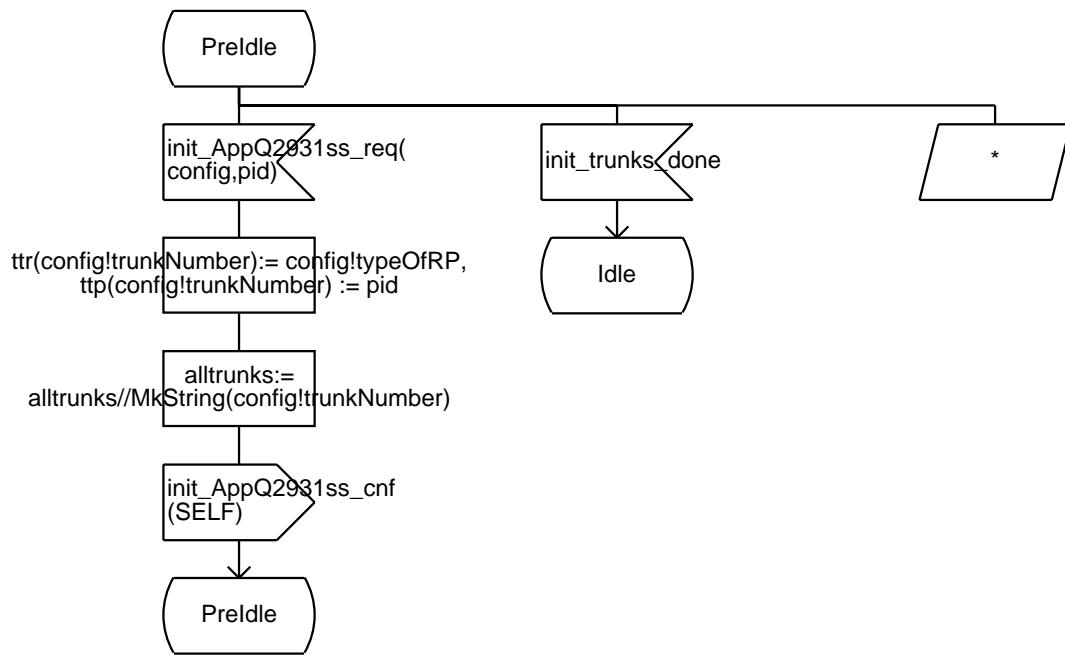
3(31)



Annex B: NetAP_PT

Process Type NetAP_PT

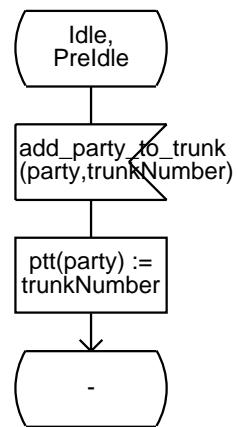
4(31)



Annex B: NetAP_PT

Process Type NetAP_PT

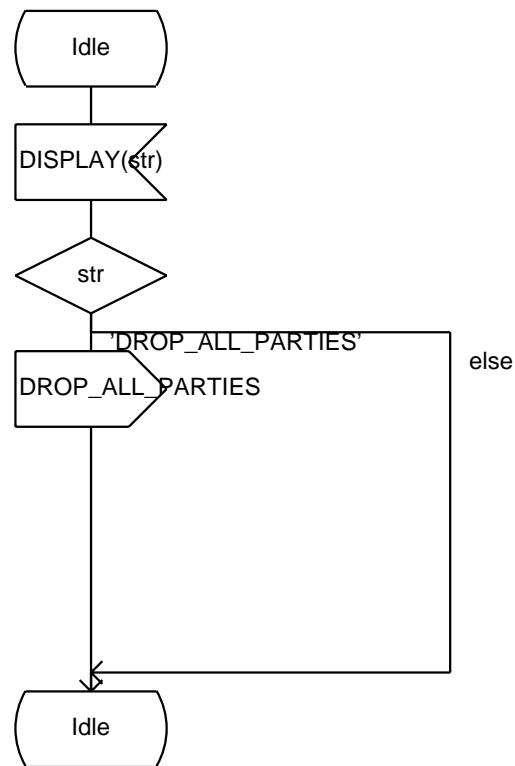
5(31)



Annex B: NetAP_PT

Process Type NetAP_PT

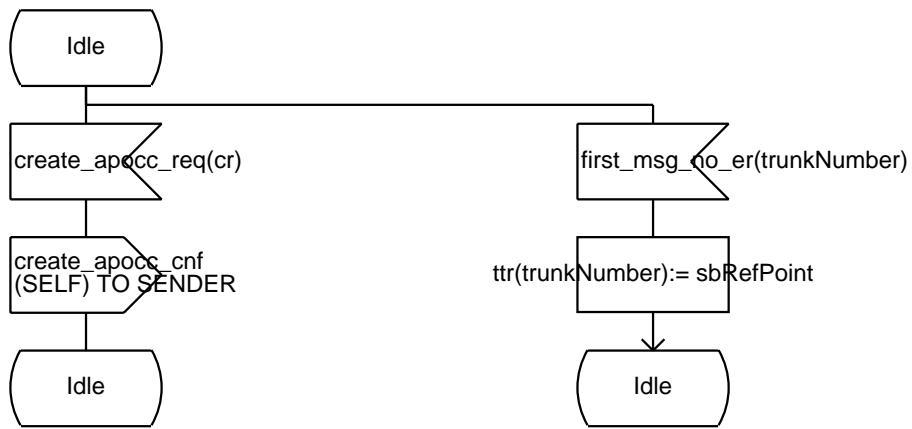
6(31)



Annex B: NetAP_PT

Process Type NetAP_PT

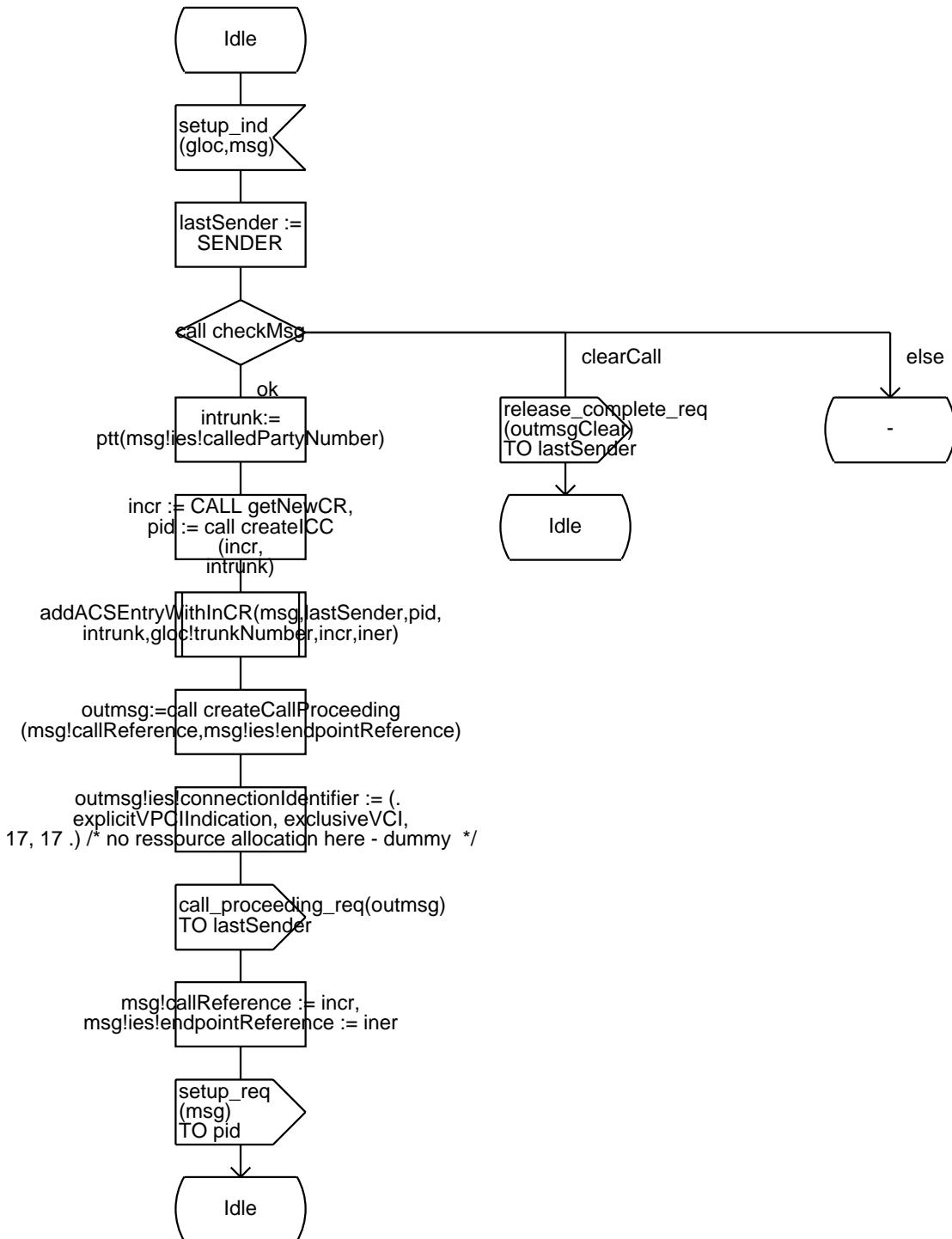
7(31)



Annex B: NetAP_PT

Process Type NetAP_PT

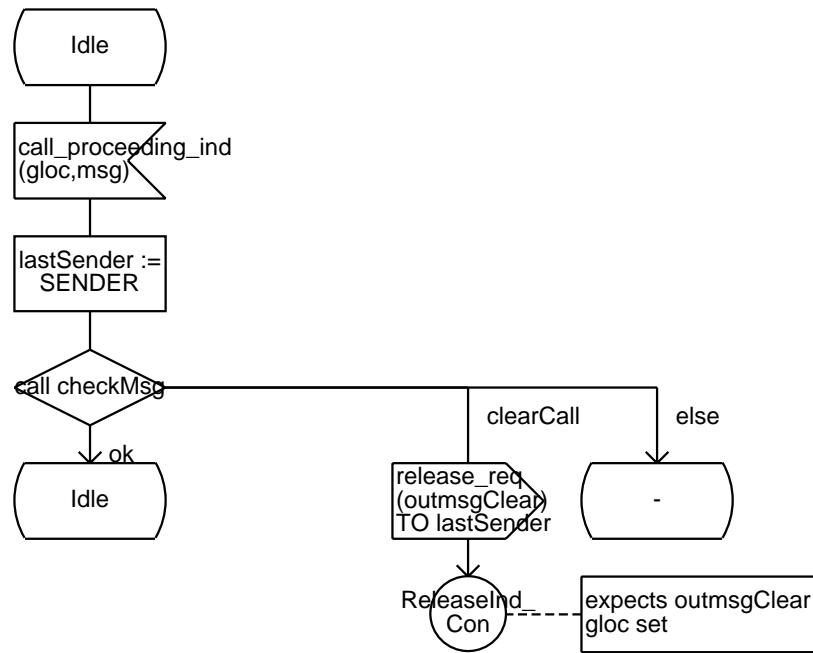
8(31)



Annex B: NetAP_PT

Process Type NetAP_PT

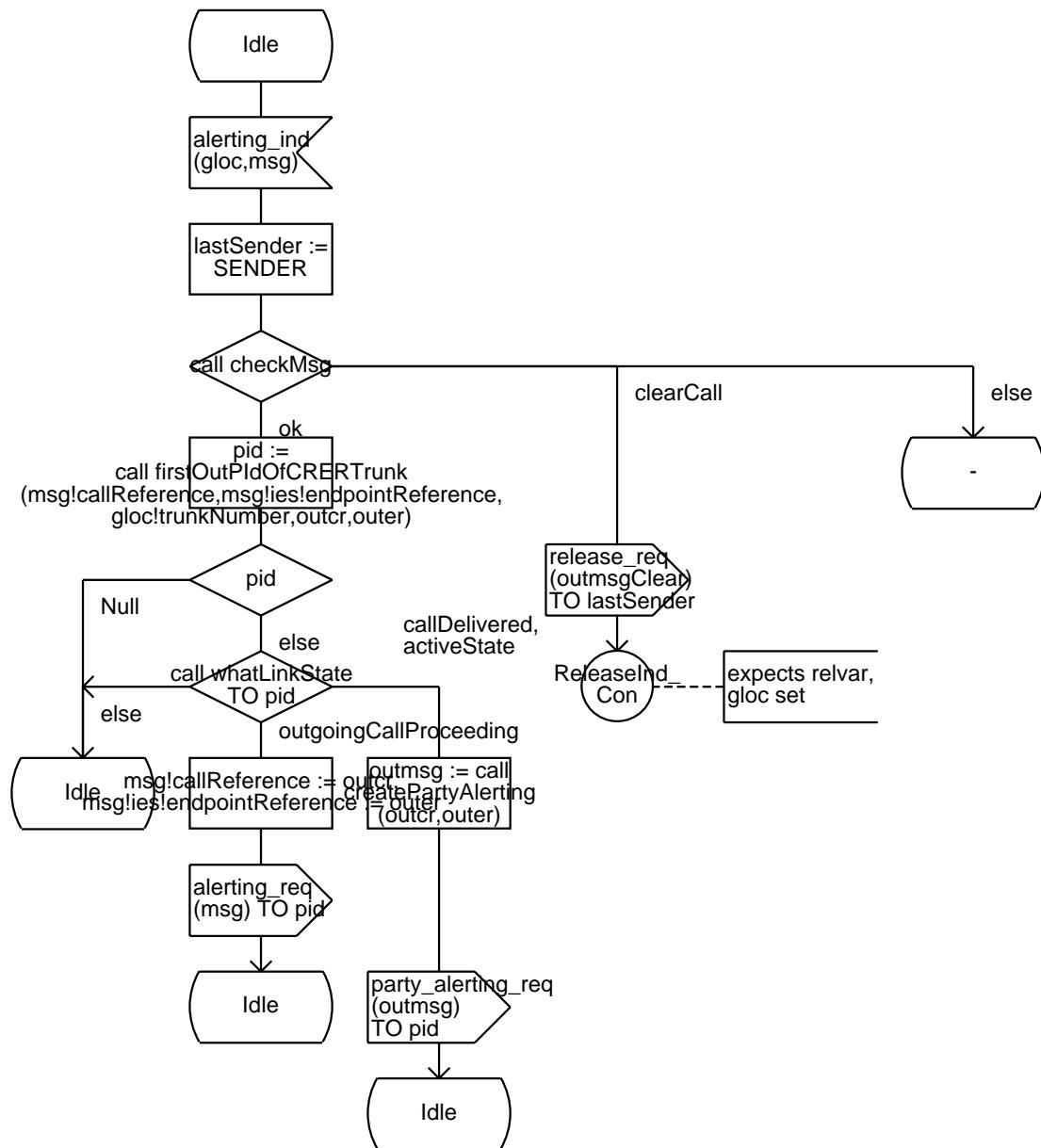
9(31)



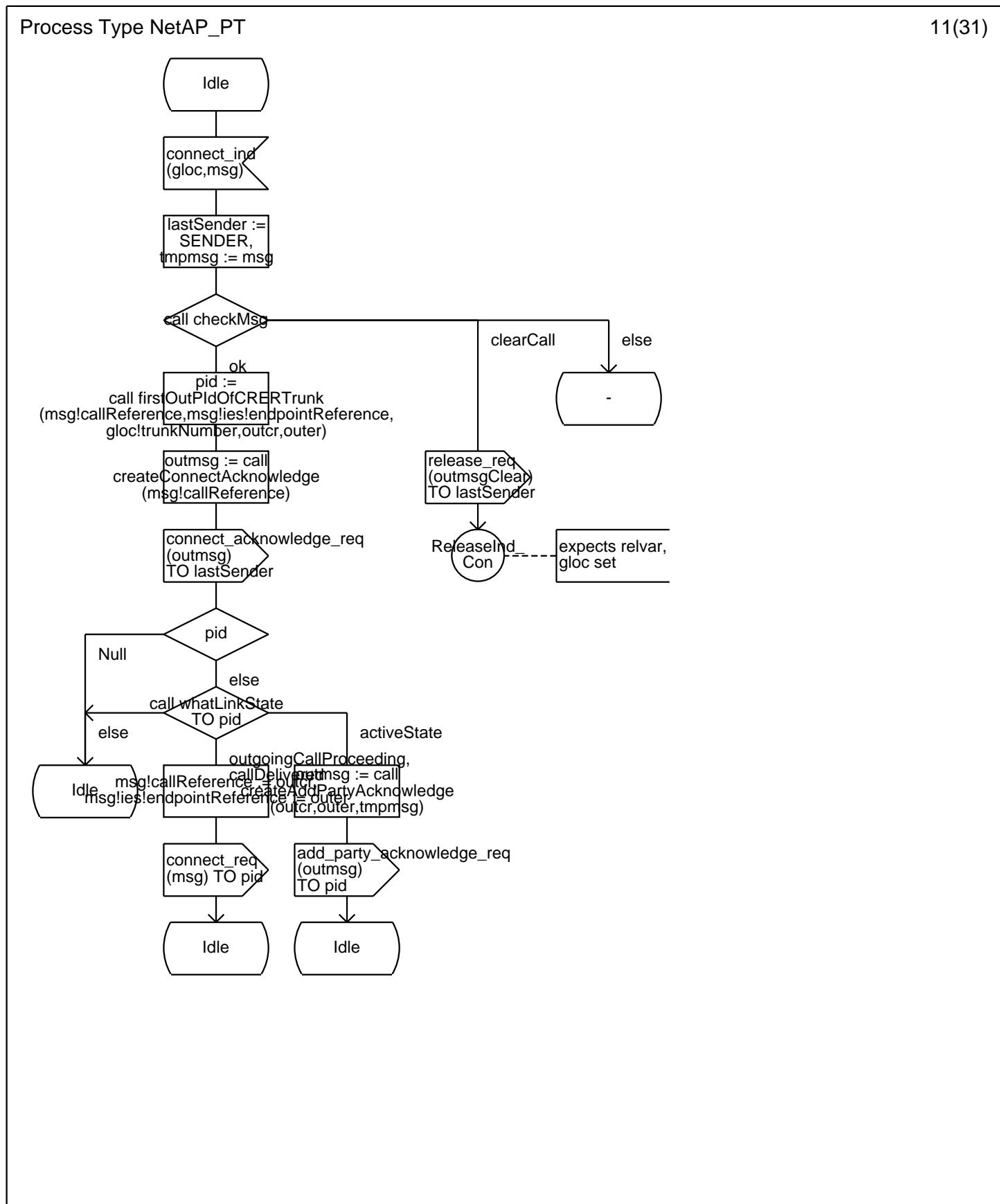
Annex B: NetAP_PT

Process Type NetAP_PT

10(31)



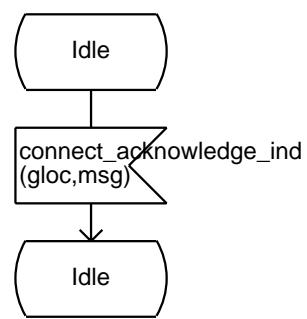
Annex B: NetAP_PT



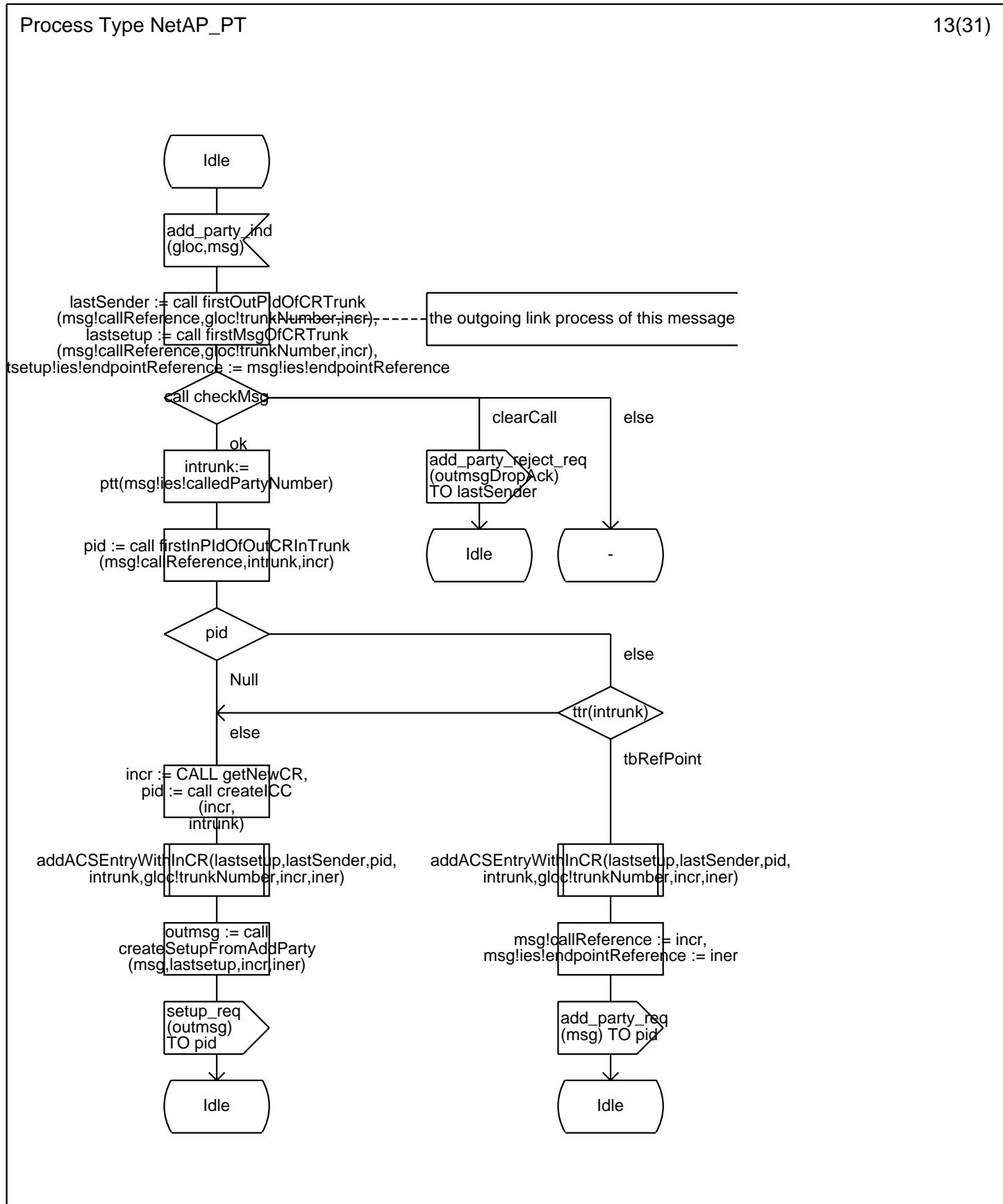
Annex B: NetAP_PT

Process Type NetAP_PT

12(31)



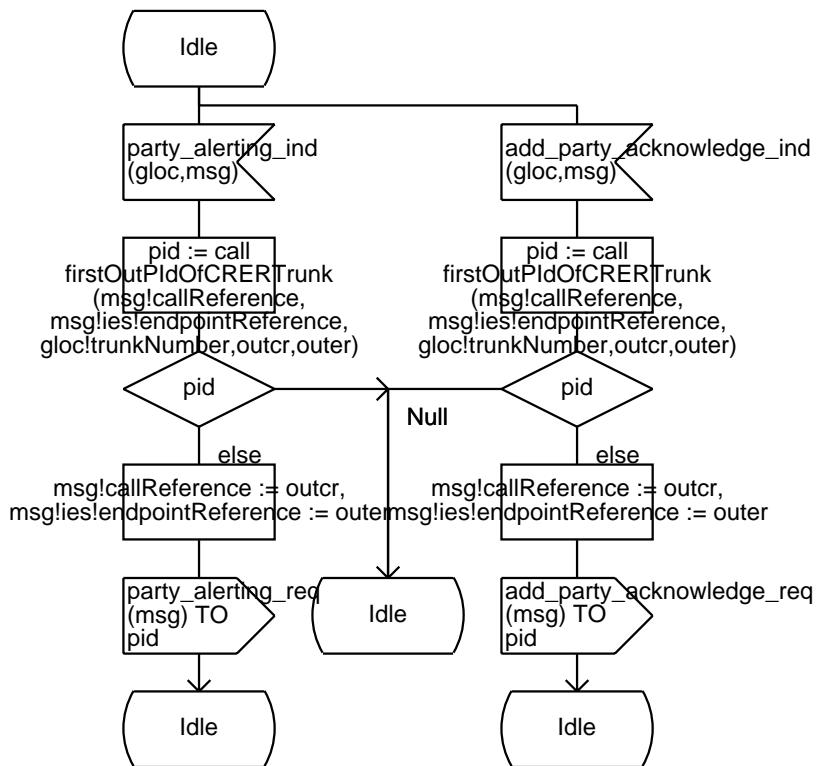
Annex B: NetAP_PT



Annex B: NetAP_PT

Process Type NetAP_PT

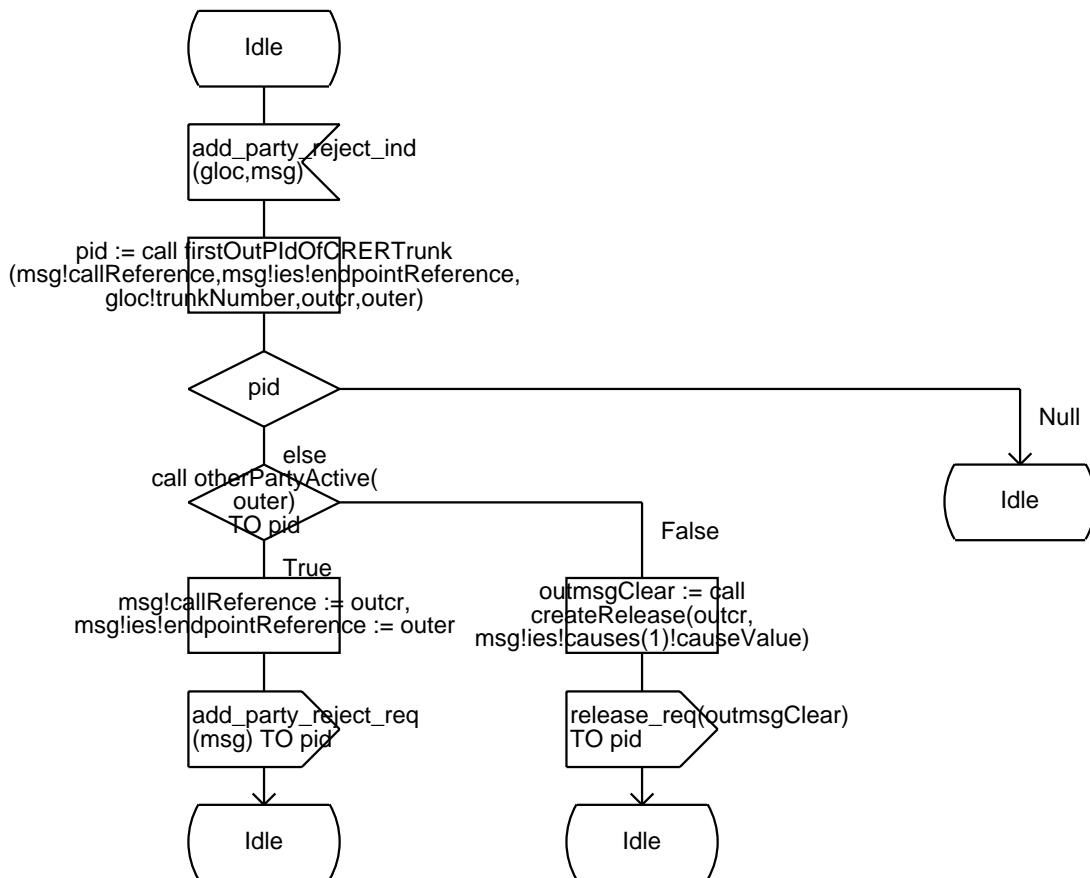
14(31)



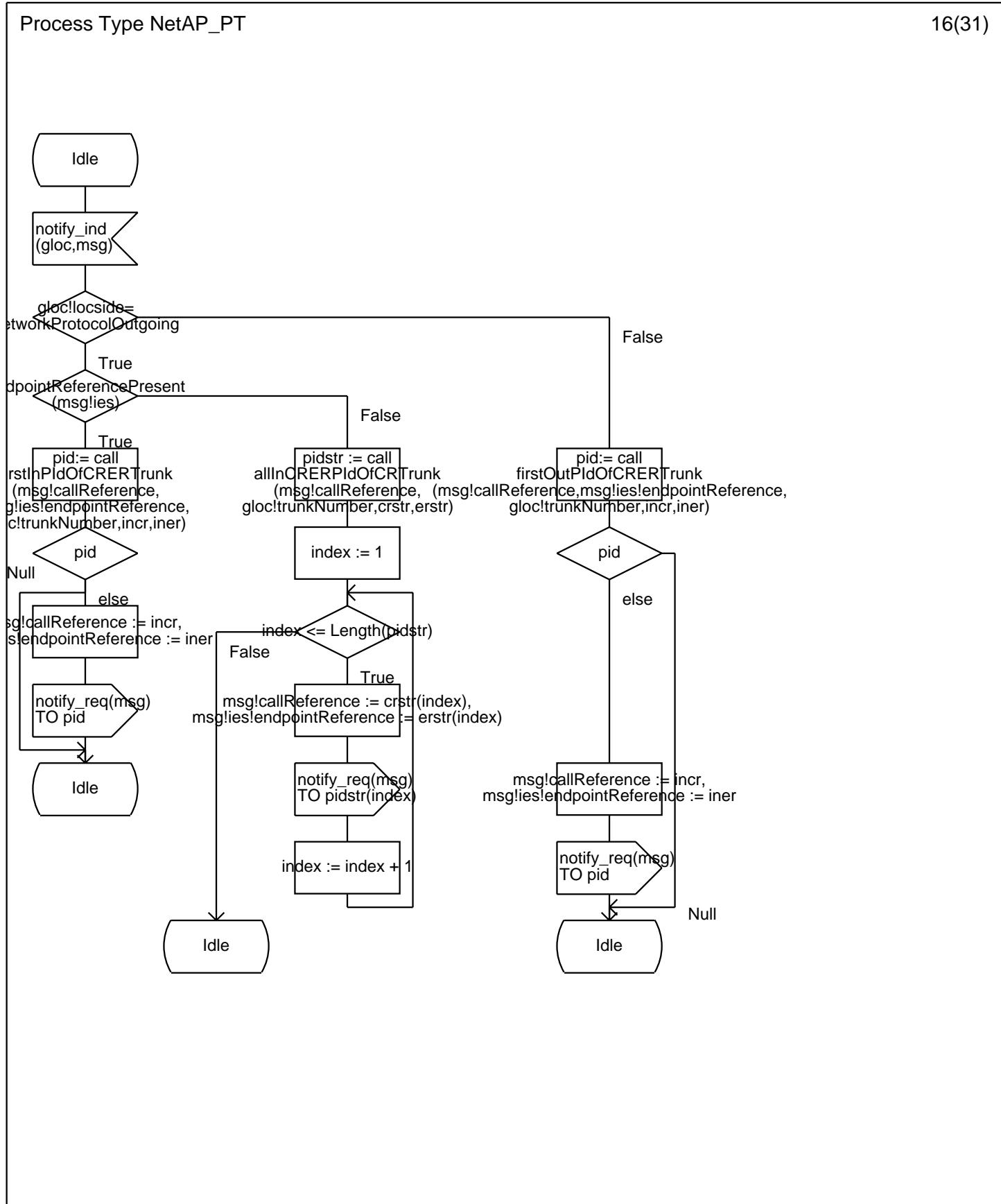
Annex B: NetAP_PT

Process Type NetAP_PT

15(31)



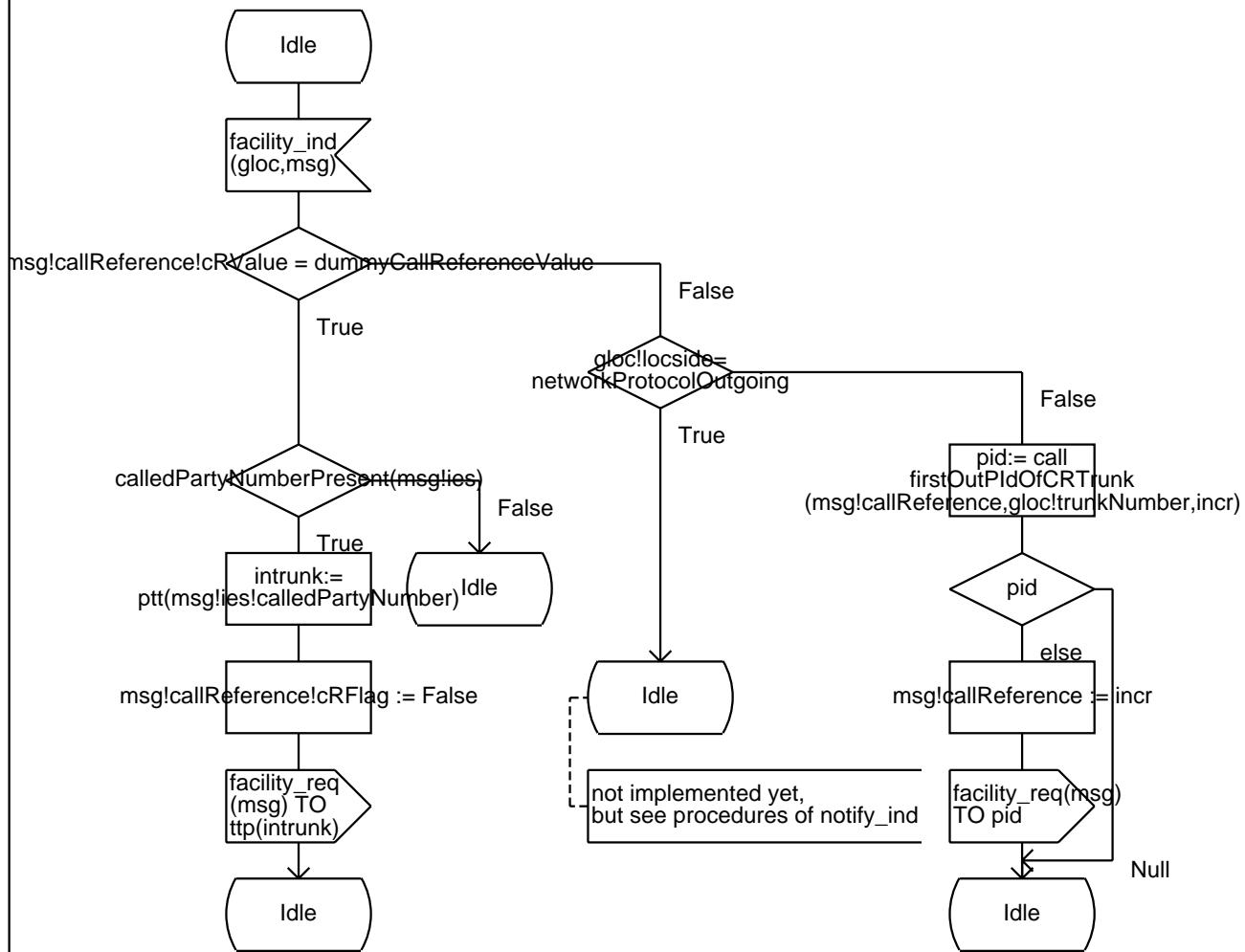
Annex B: NetAP_PT



Annex B: NetAP_PT

Process Type NetAP_PT

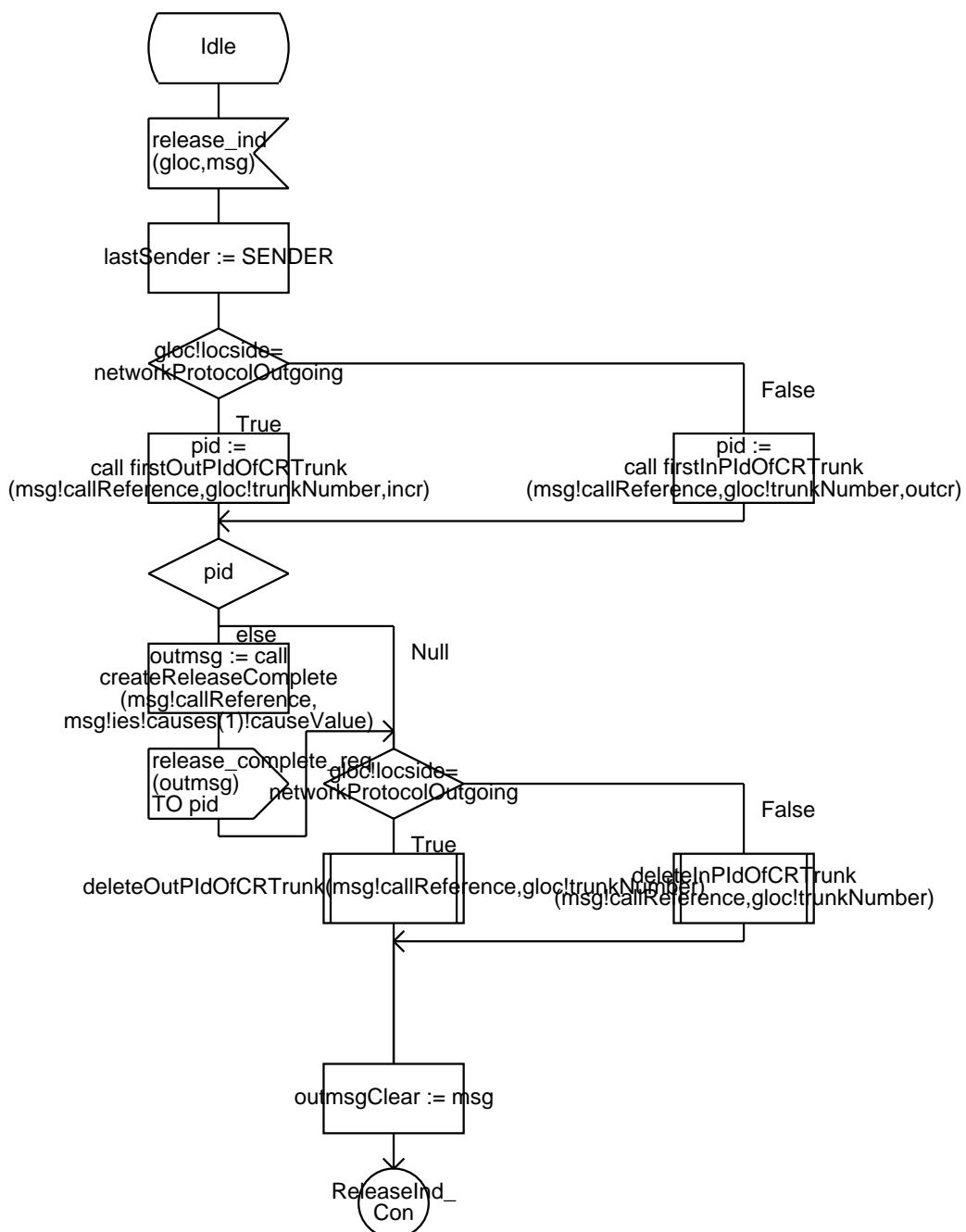
17(31)



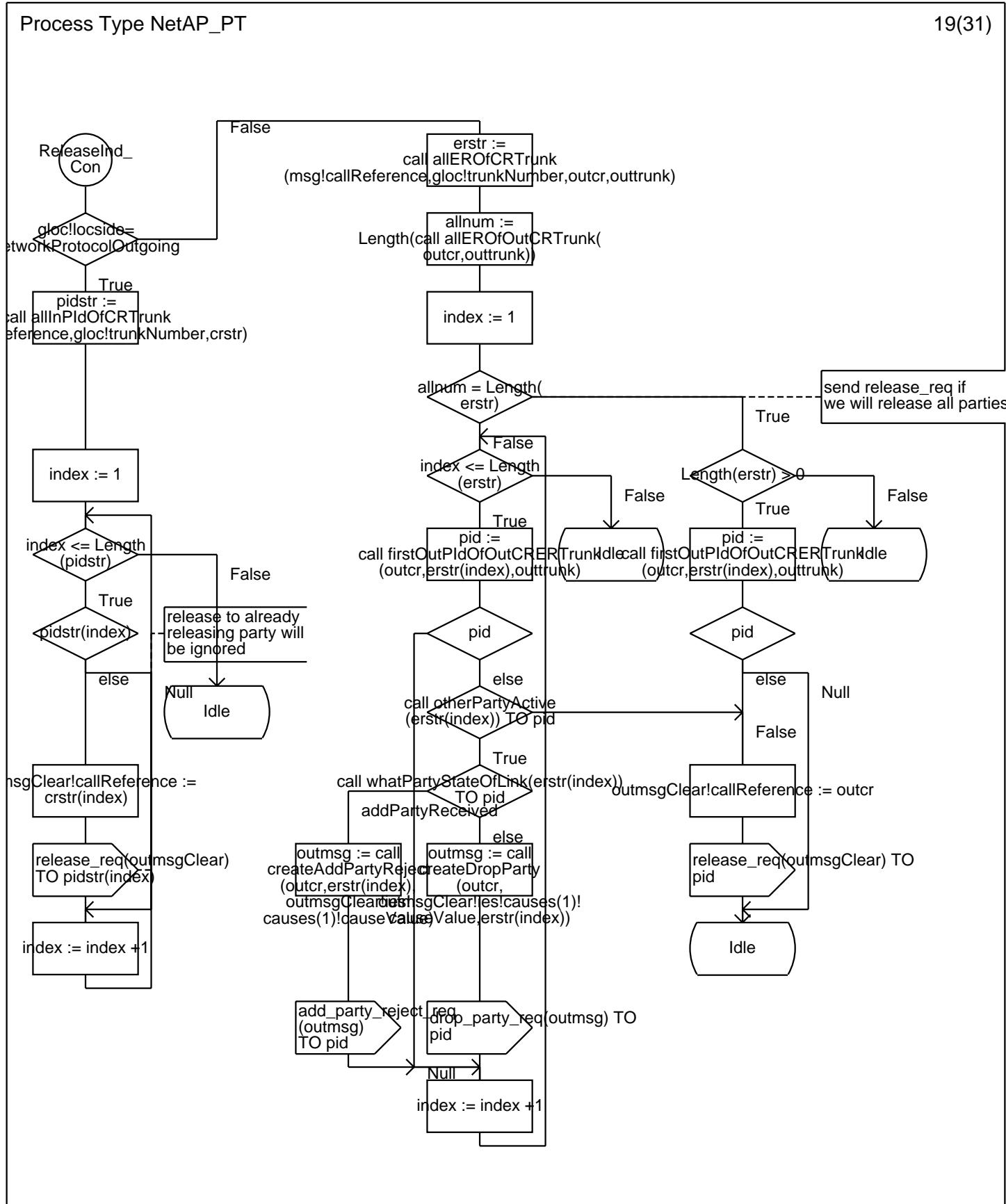
Annex B: NetAP_PT

Process Type NetAP_PT

18(31)



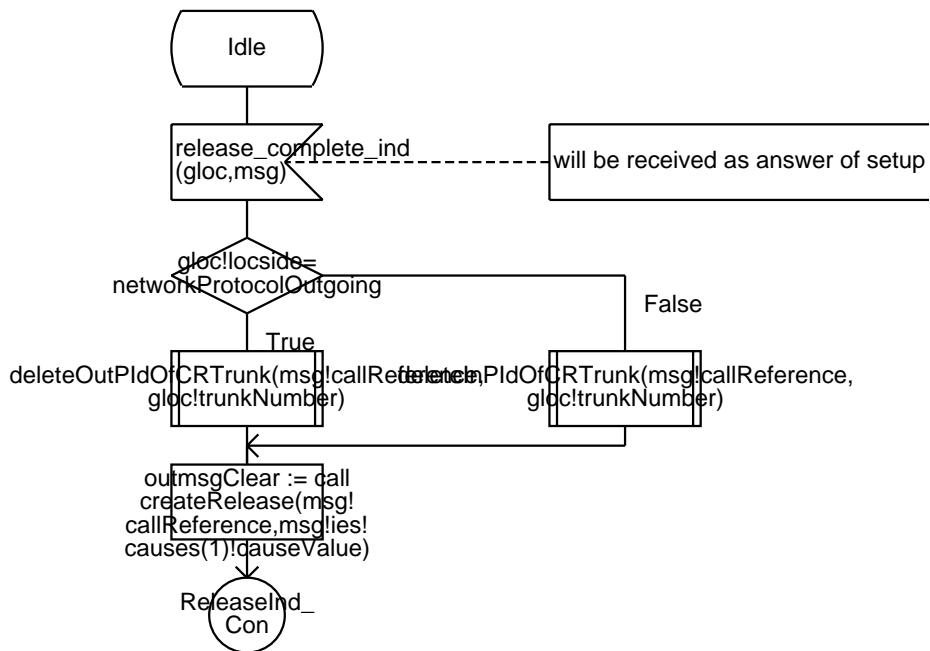
Annex B: NetAP_PT



Annex B: NetAP_PT

Process Type NetAP_PT

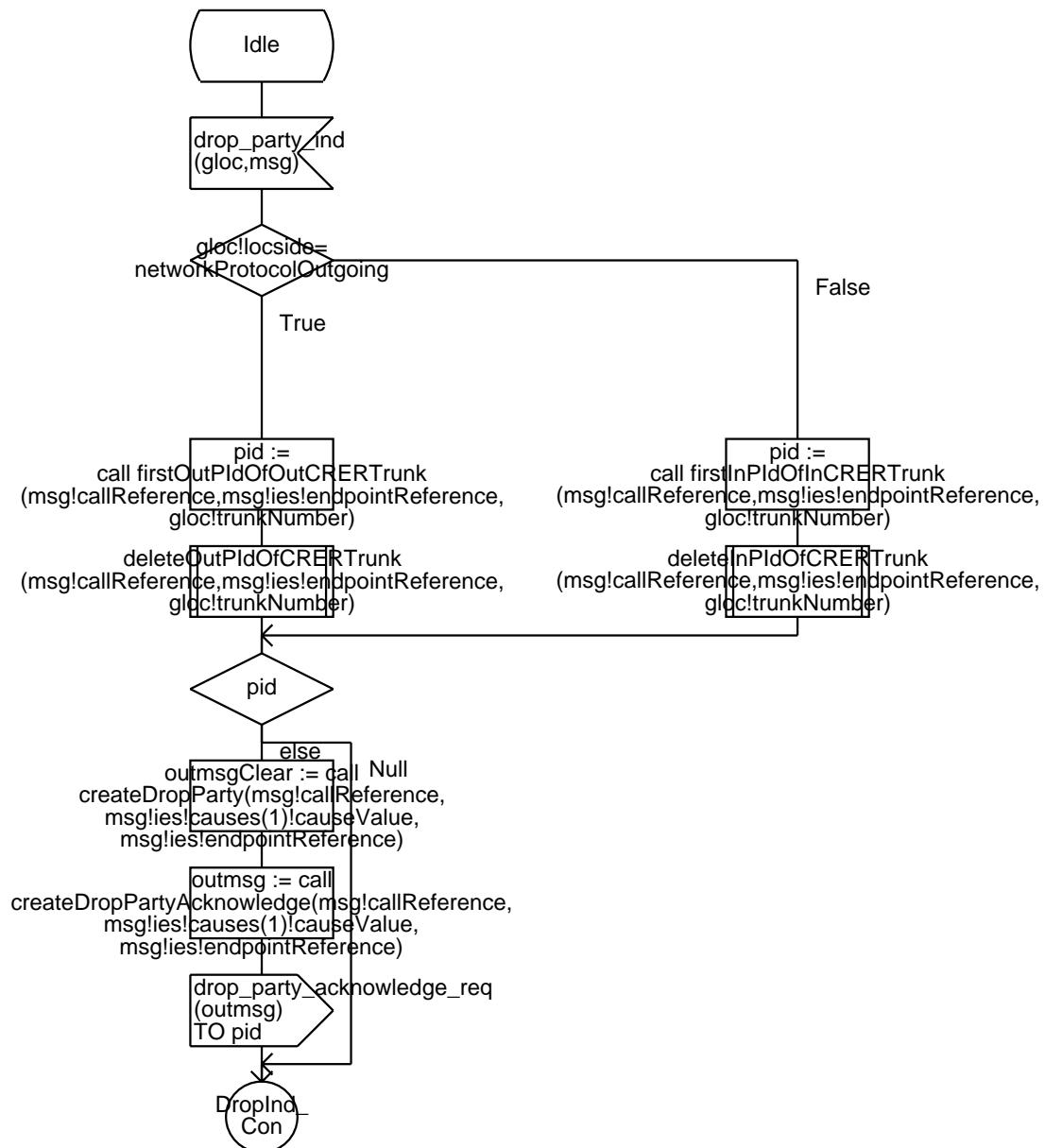
20(31)



Annex B: NetAP_PT

Process Type NetAP_PT

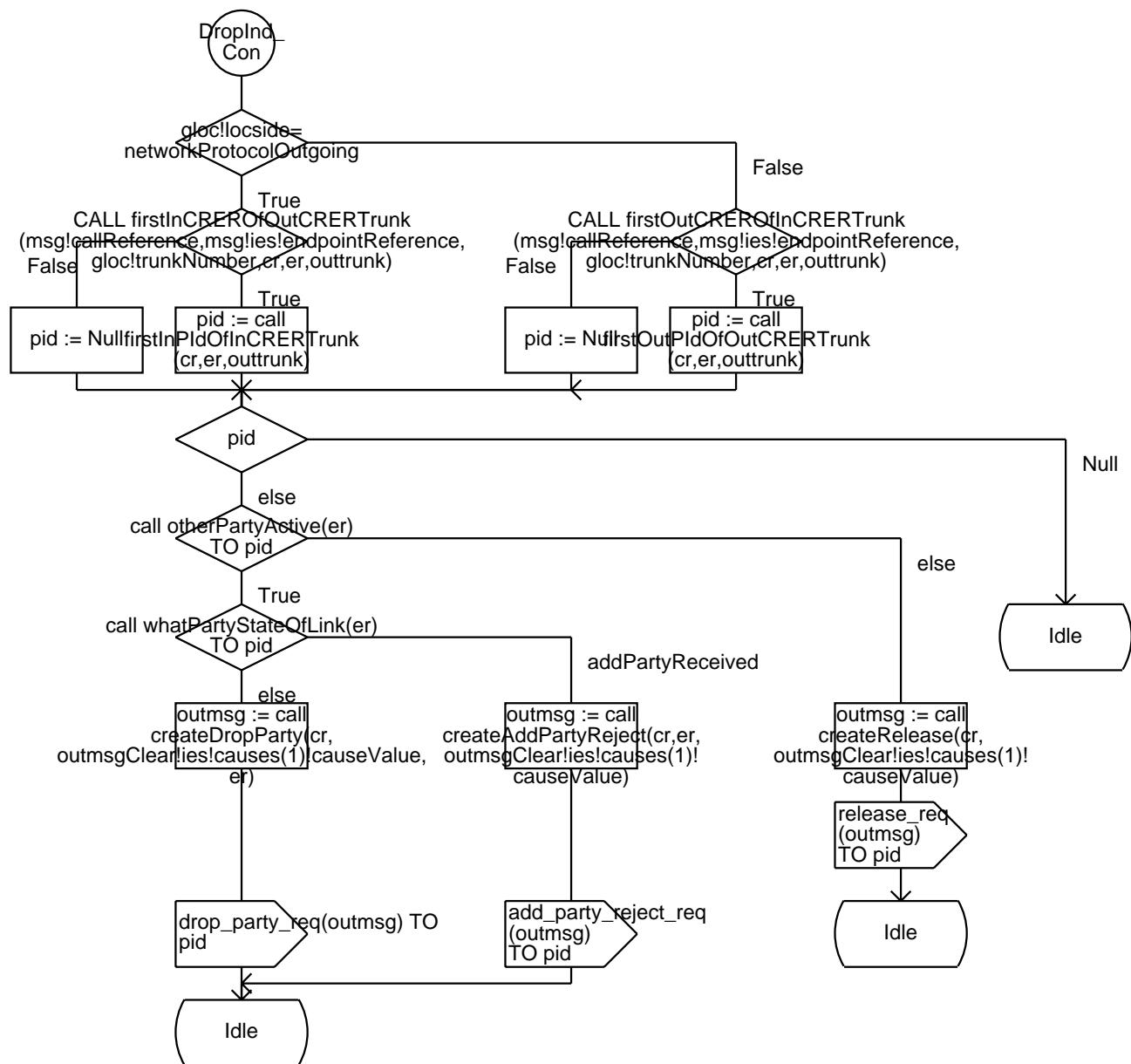
21(31)



Annex B: NetAP_PT

Process Type NetAP_PT

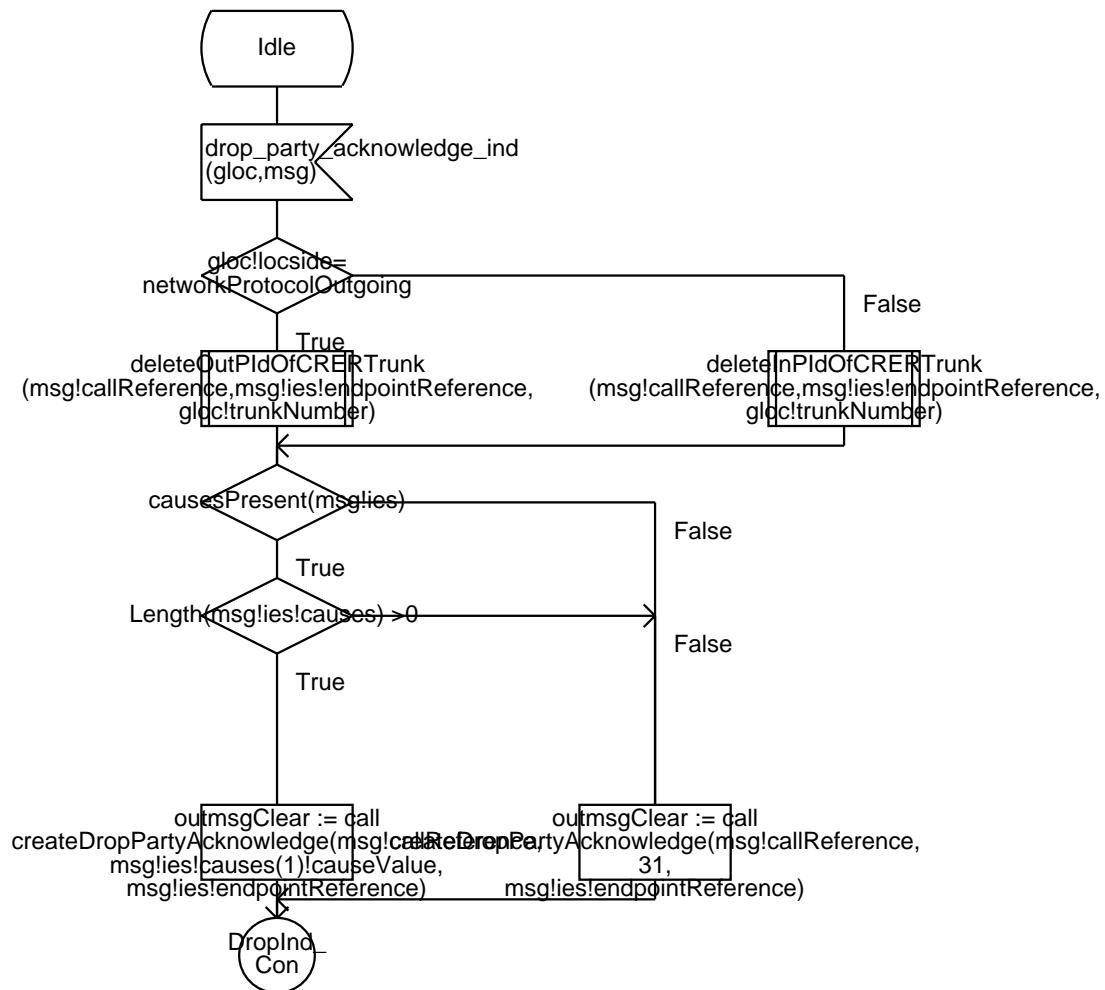
22(31)



Annex B: NetAP_PT

Process Type NetAP_PT

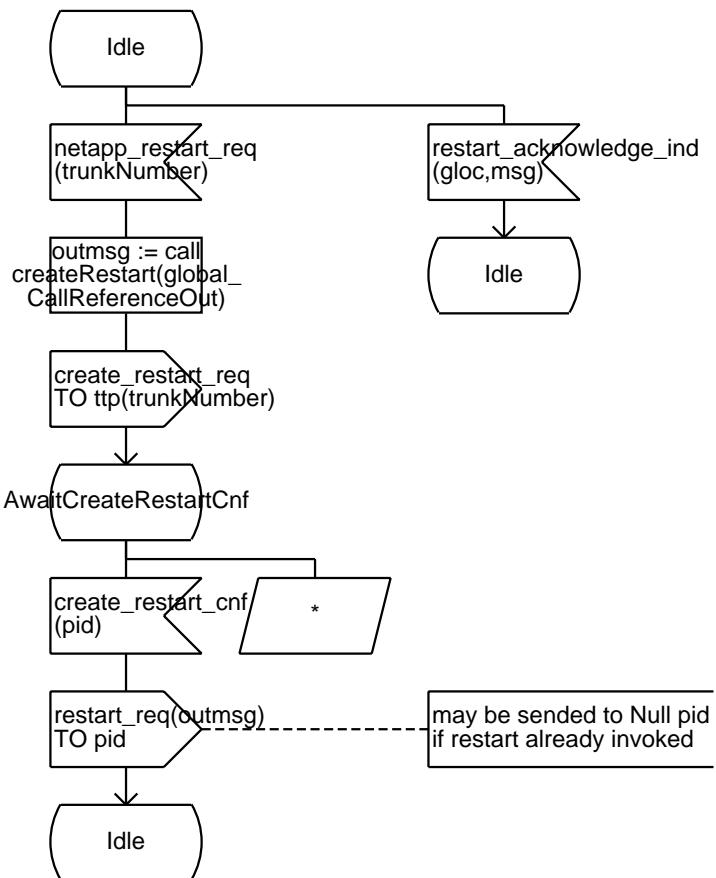
23(31)



Annex B: NetAP_PT

Process Type NetAP_PT

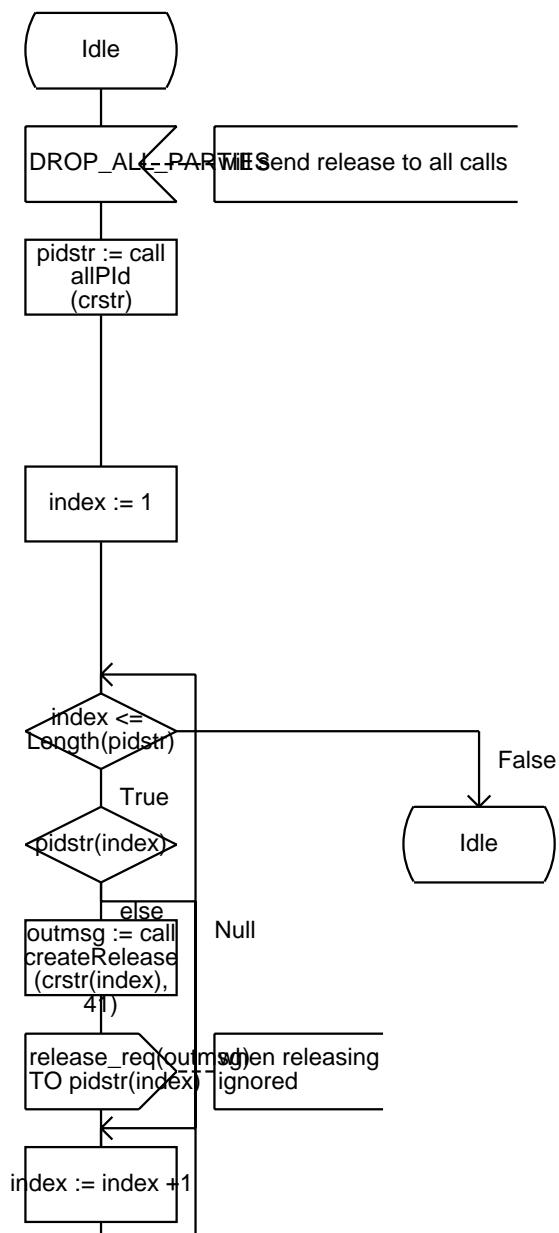
24(31)



Annex B: NetAP_PT

Process Type NetAP_PT

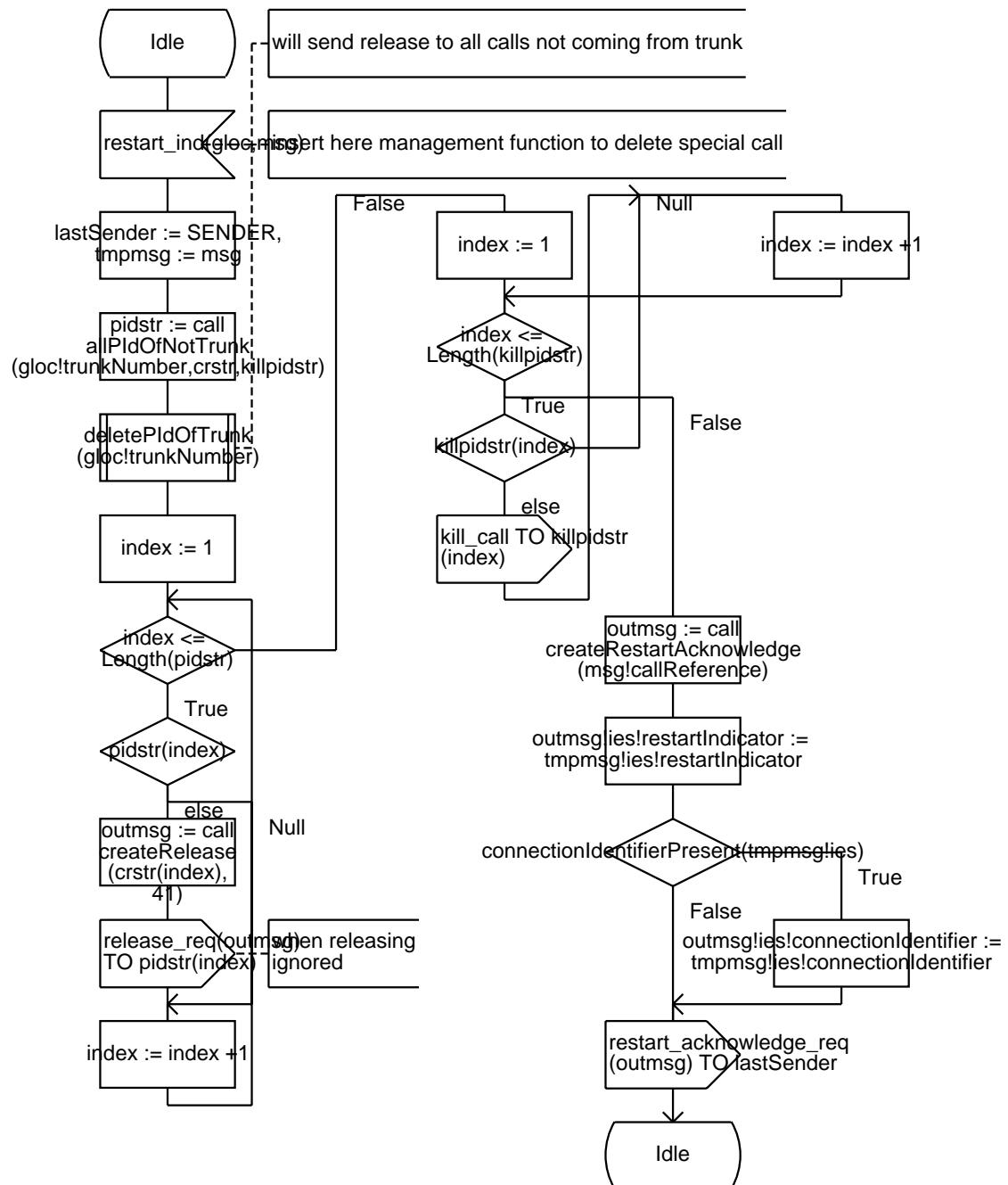
25(31)



Annex B: NetAP_PT

Process Type NetAP_PT

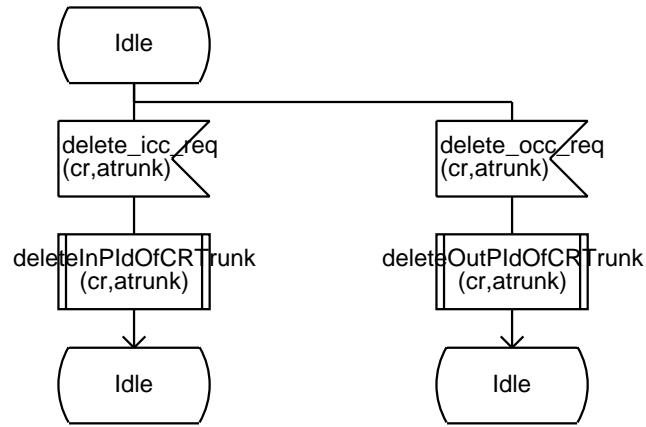
26(31)



Annex B: NetAP_PT

Process Type NetAP_PT

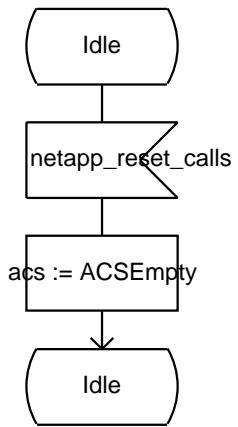
27(31)



Annex B: NetAP_PT

Process Type NetAP_PT

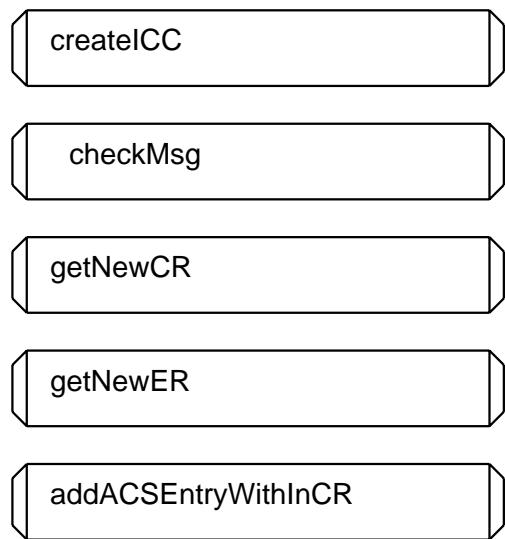
28(31)



Annex B: NetAP_PT

Process Type NetAP_PT

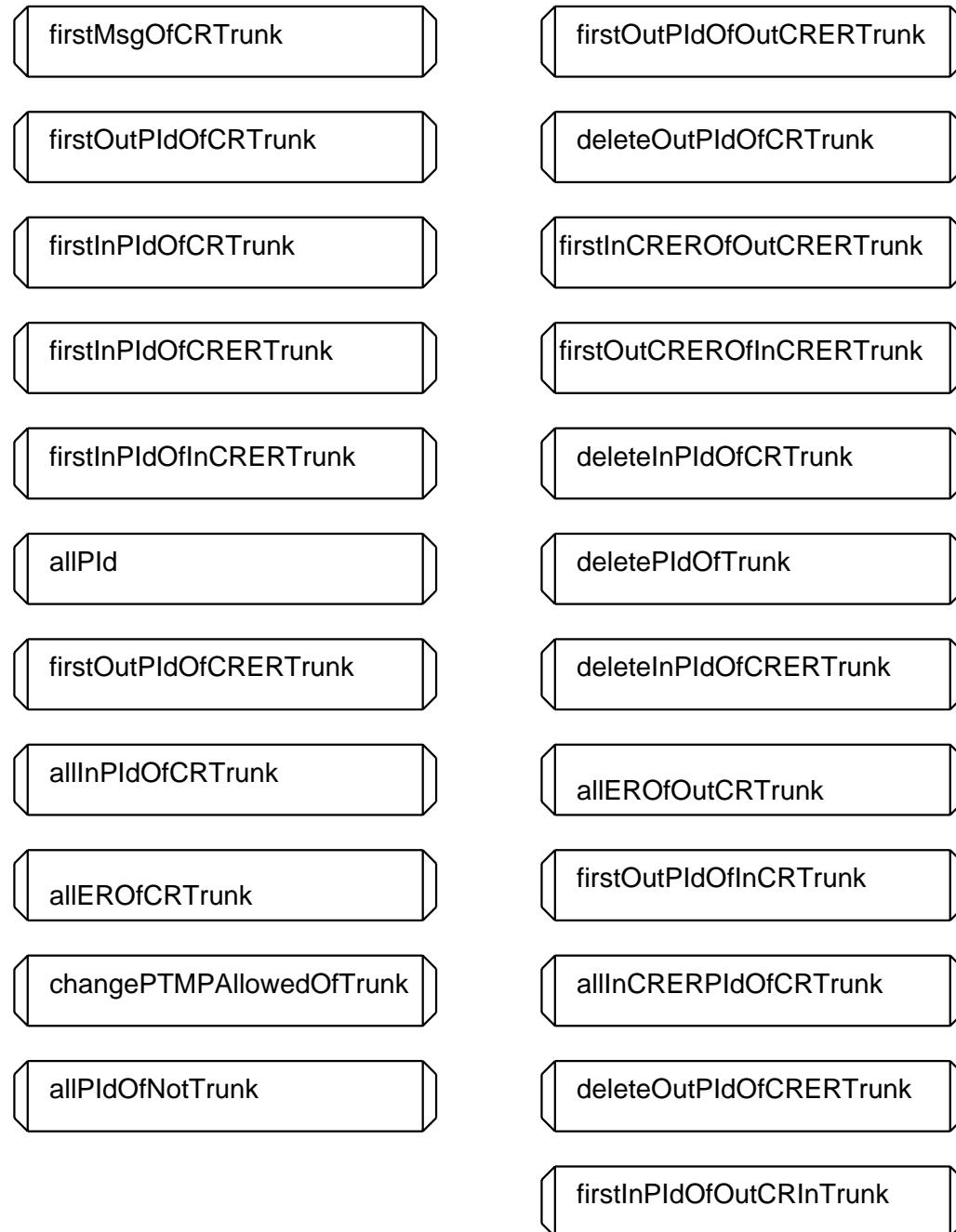
29(31)



Annex B: NetAP_PT

Process Type NetAP_PT

30(31)



Annex B: NetAP_PT

Process Type NetAP_PT

31(31)

```
IMPORTED PROCEDURE whatPartyStateOfLink;  
FPAR EndpointReference;  
RETURNS EndpointState;
```

```
IMPORTED PROCEDURE whatLinkState;  
RETURNS CallState;
```

```
IMPORTED PROCEDURE otherPartyActive;  
FPAR EndpointReference;  
RETURNS Boolean;
```

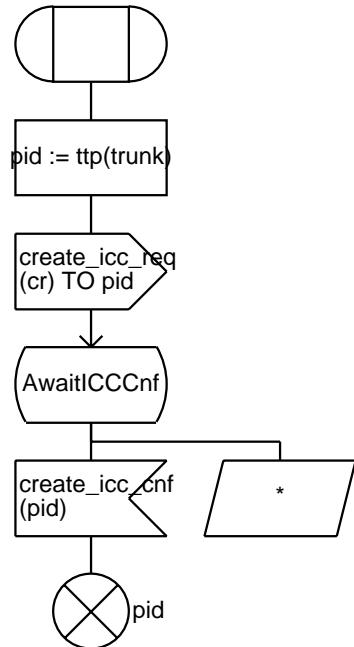
Annex B: createICC

Procedure createICC

1(1)

```
;fpar cr CallReference,trunk TrunkNumber;  
returns PId;
```

```
DCL pid PId;
```

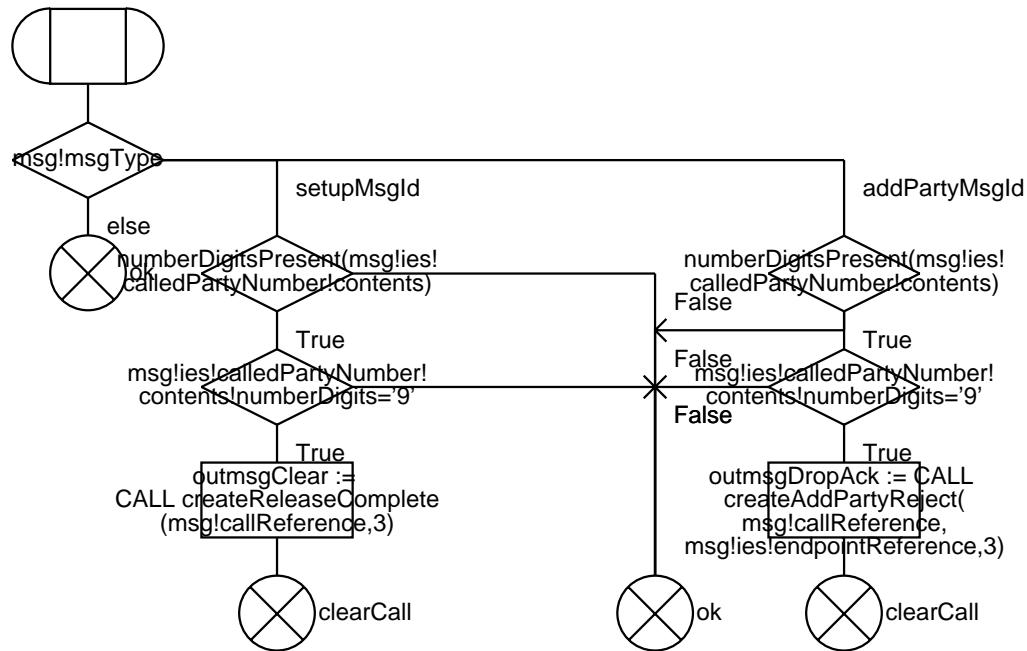


Annex B: checkMsg

Procedure checkMsg

1(1)

;RETURNS CheckResultType;

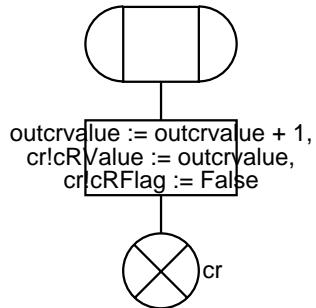


Annex B: getNewCR

Procedure <<Process Type NetAP_PT>> getNewCR

1(1)

:RETURNS CallReference;



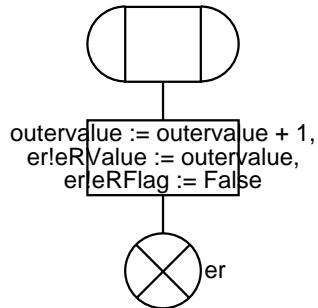
DCL cr CallReference;

Annex B: getNewER

Procedure <<Process Type NetAP_PT>> getNewER

1(1)

:RETURNS EndpointReference;



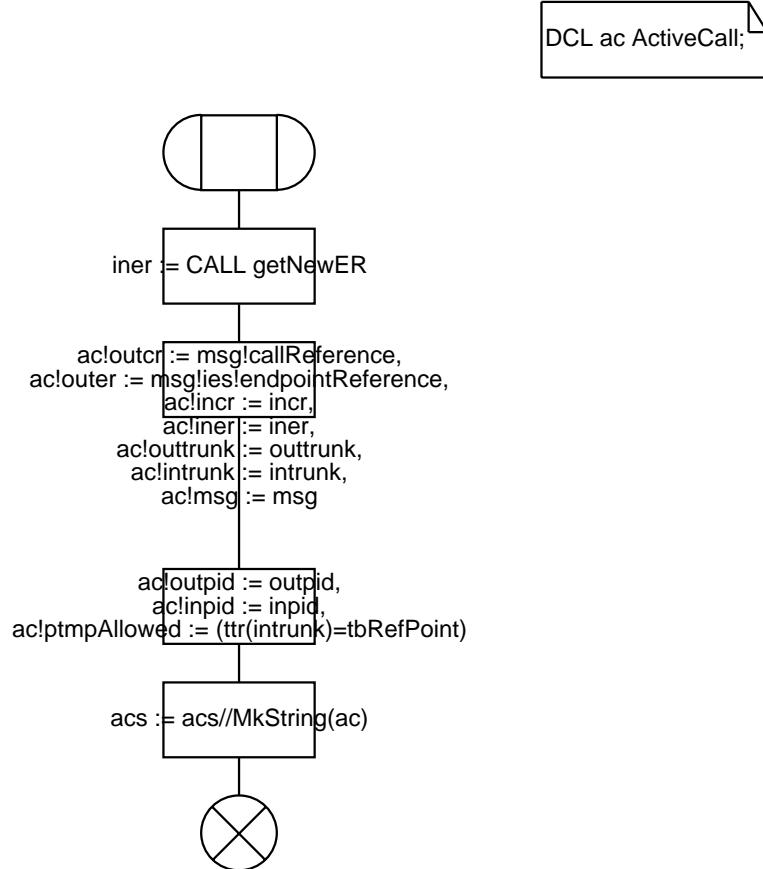
DCL er EndpointReference;

Annex B: addACSEntryWithInCR

Procedure addACSEntryWithInCR

1(1)

;fpar msg Q2931ssMessage, outpid PId, inpid PId, intrunk TrunkNumber,outtrunk TrunkNumber, in incr CallReference, in/out iner End

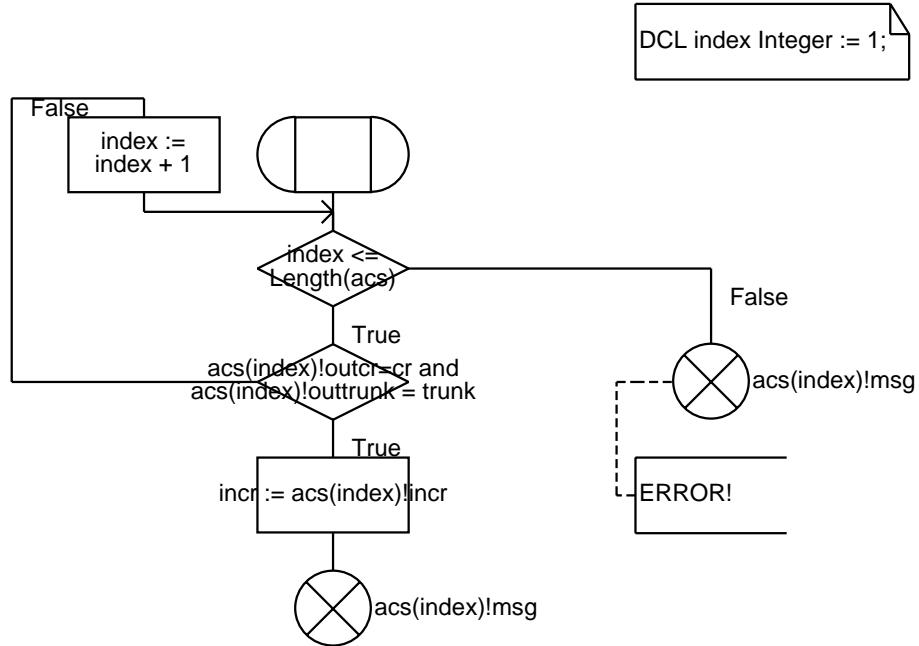


Annex B: firstMsgOfCRTTrunk

Procedure firstMsgOfCRTTrunk

1(1)

;fpar cr CallReference, trunk TrunkNumber ,in/out incr CallReference;
returns Q2931ssMessage;



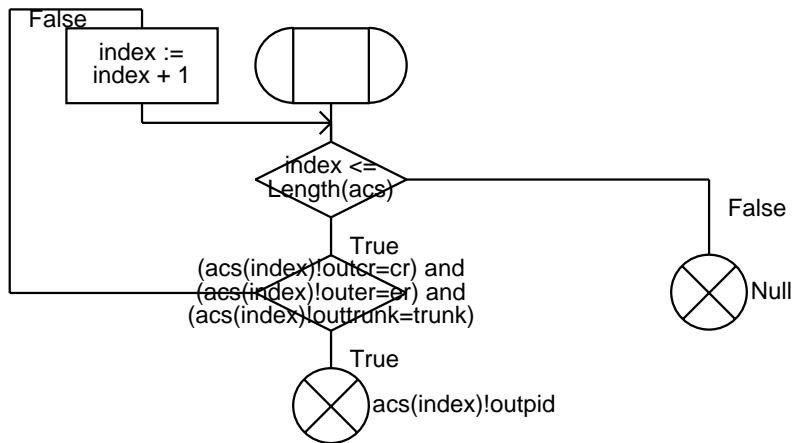
Annex B: firstOutPIdOfOutCRERTrunk

Procedure firstOutPIdOfOutCRERTrunk

1(1)

;fpar cr CallReference,er EndpointReference,trunk TrunkNumber;
returns PId;

DCL index Integer := 1;



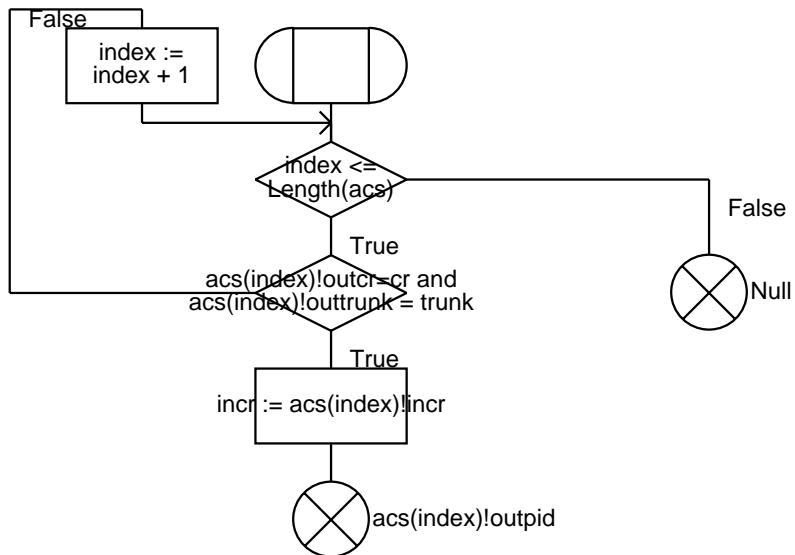
Annex B: firstOutPlidOfCRTTrunk

Procedure firstOutPlidOfCRTTrunk

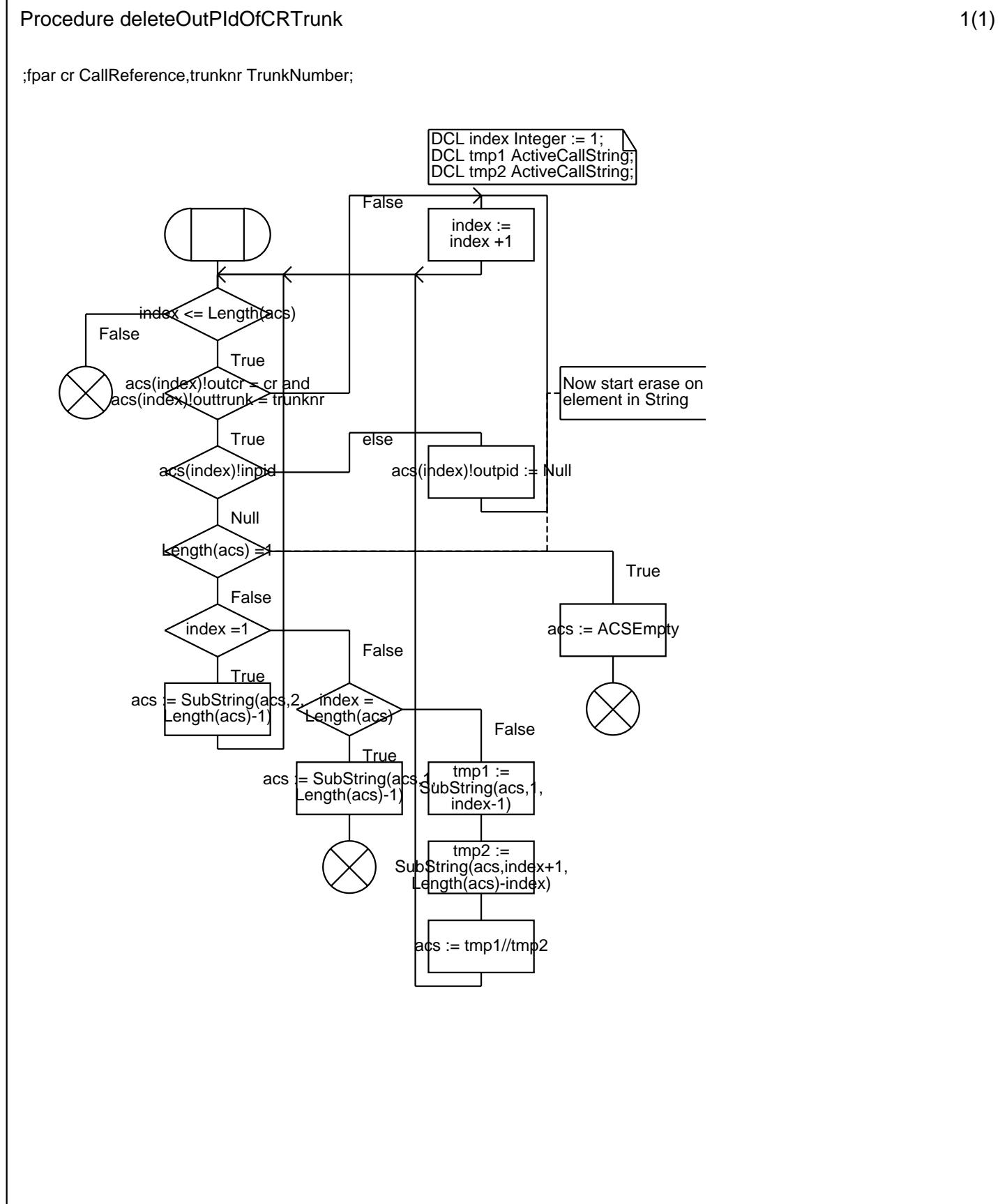
1(1)

;fpar cr CallReference, trunk TrunkNumber ,in/out incr CallReference;
returns Plid;

DCL index Integer := 1;



Annex B: deleteOutPlidOfCRTTrunk



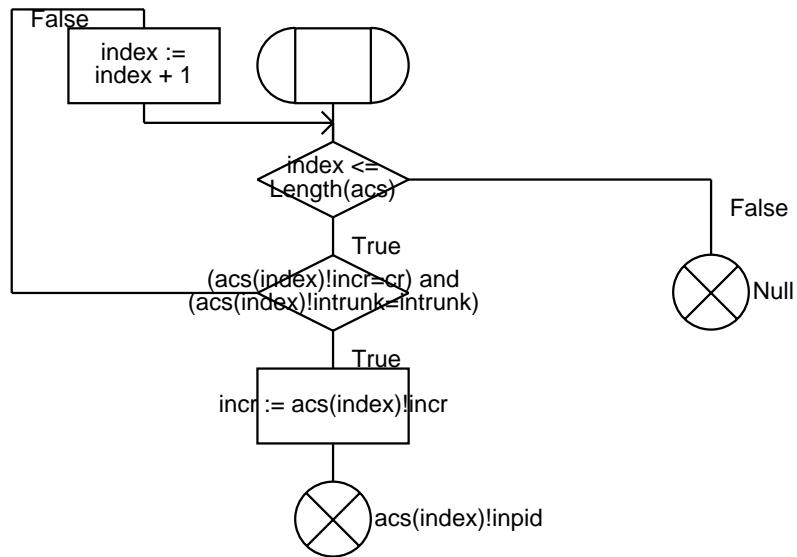
Annex B: firstInPlidOfCRTTrunk

Procedure firstInPlidOfCRTTrunk

1(1)

;fpar cr CallReference, intrunk TrunkNumber,in/out incr CallReference;
returns Plid;

DCL index Integer := 1;

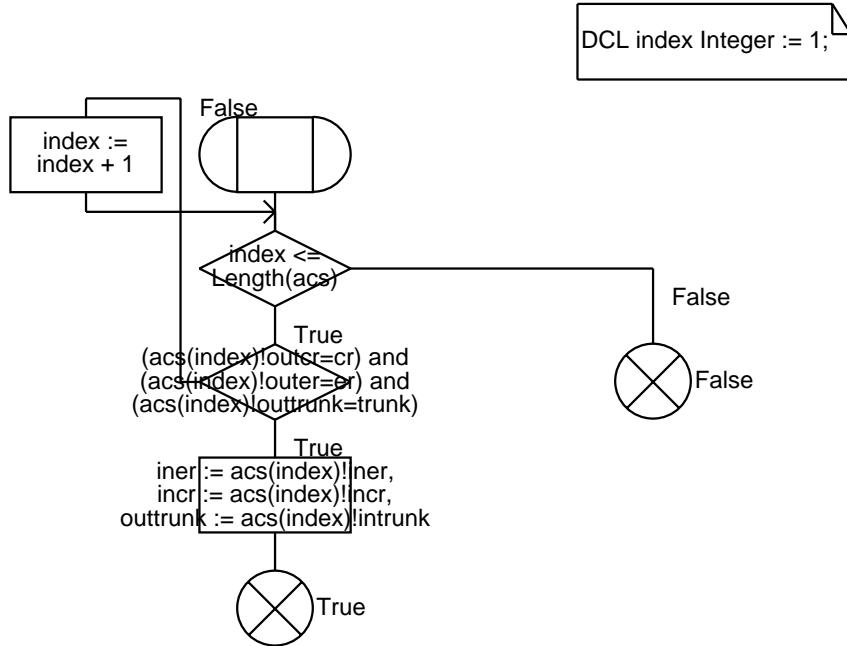


Annex B: firstInCREROfOutCRERTrunk

Procedure firstInCREROfOutCRERTrunk

1(1)

;fpar cr CallReference, er EndpointReference, trunk TrunkNumber, in/out incr CallReference, in/out iner EndpointReference, in/out ou returns Boolean;

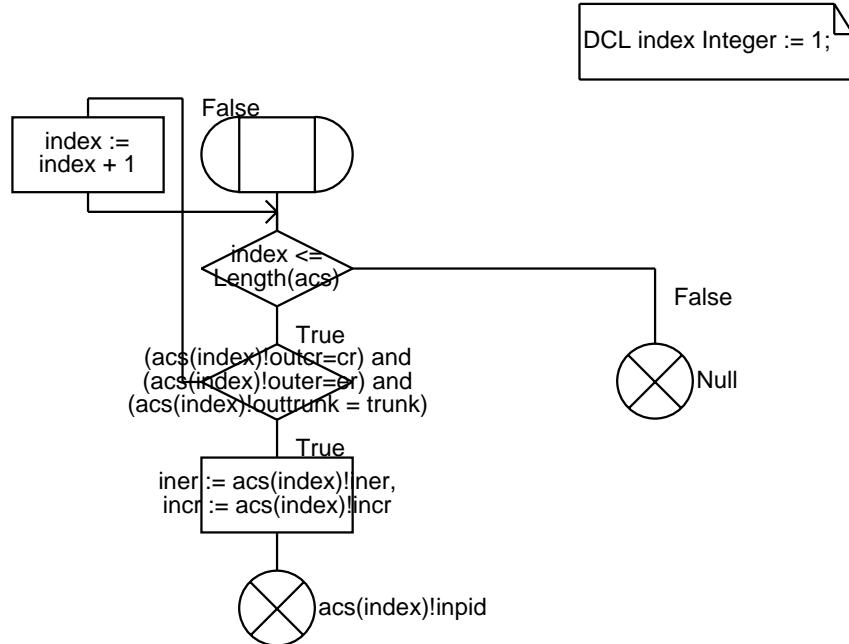


Annex B: firstInPIdOfCRERTrunk

Procedure firstInPIdOfCRERTrunk

1(1)

;fpar cr CallReference, er EndpointReference, trunk TrunkNumber,in/out incr CallReference,in/out iner EndpointReference;
returns PId;

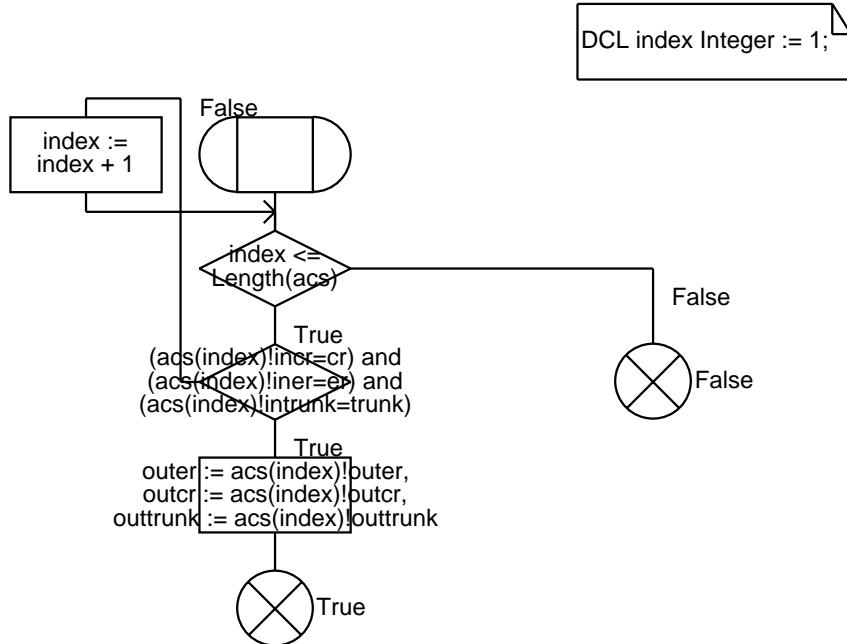


Annex B: firstOutCREROflnCRERTrunk

Procedure firstOutCREROflnCRERTrunk

1(1)

:fpar cr CallReference, er EndpointReference, trunk TrunkNumber, in/out outcr CallReference, in/out outer EndpointReference,in/out
 returns Boolean;



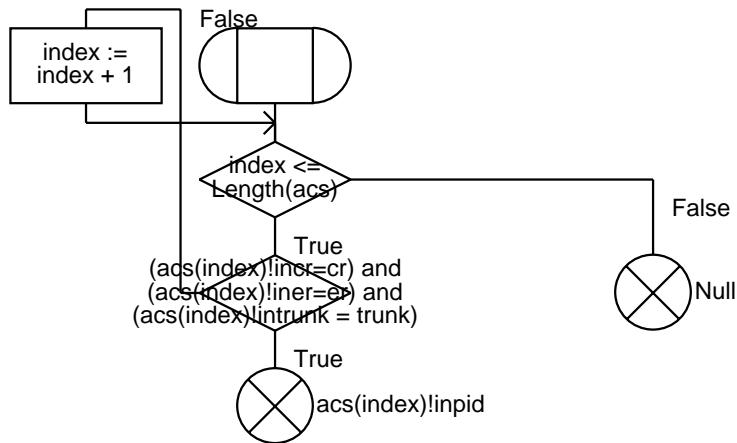
Annex B: firstInPlidOfInCRERTrunk

Procedure firstInPlidOfInCRERTrunk

1(1)

;fpar cr CallReference, er EndpointReference, trunk TrunkNumber;
returns Plid;

DCL index Integer := 1;

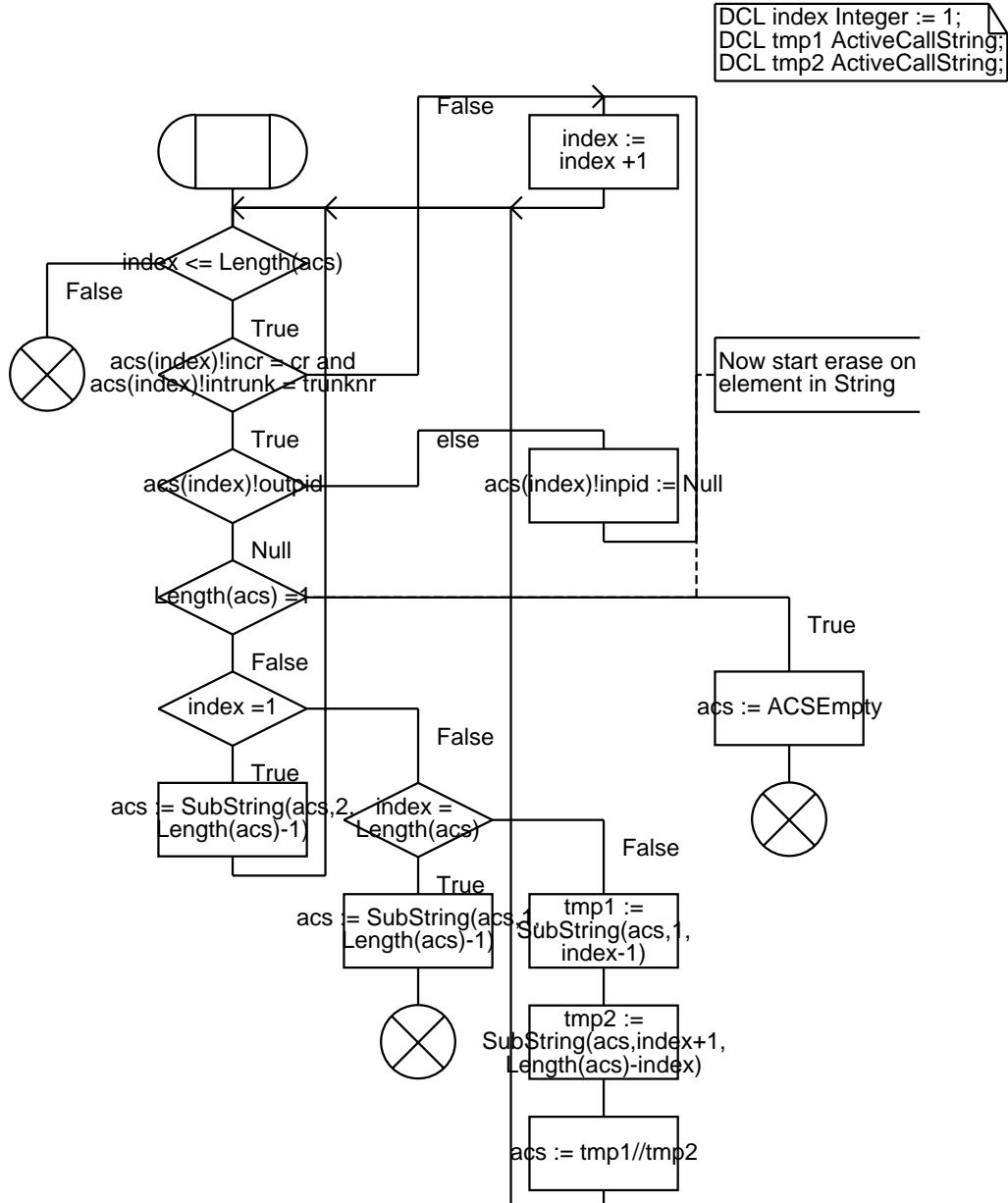


Annex B: deleteInPlidOfCRTTrunk

Procedure deleteInPlidOfCRTTrunk

1(1)

;fpar cr CallReference,trunknr TrunkNumber;

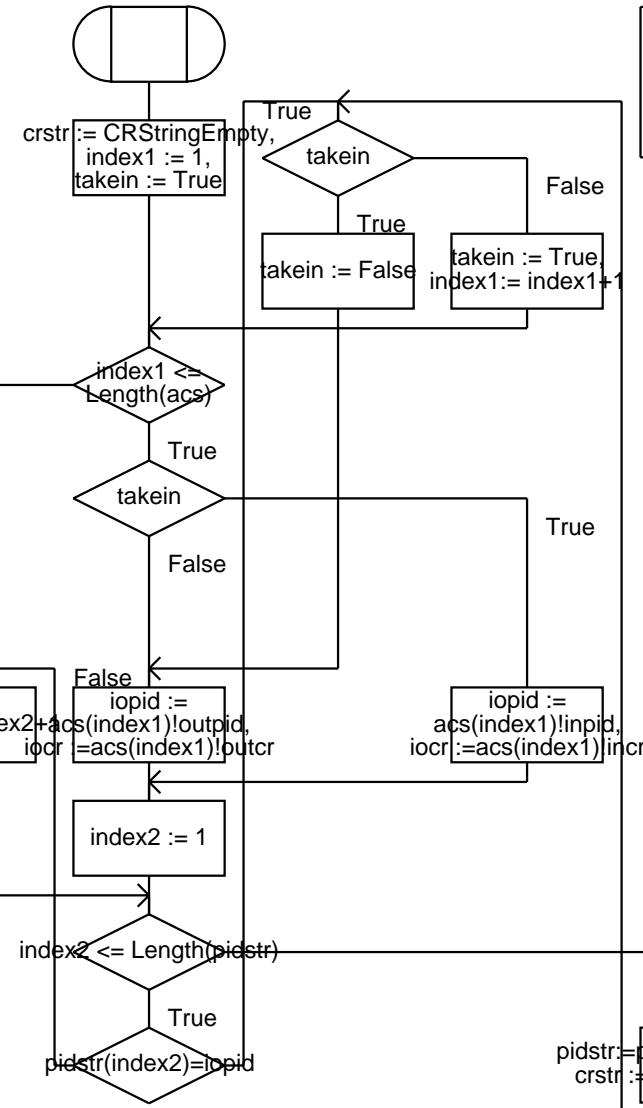


Annex B: allPlId

Procedure allPlId

1(1)

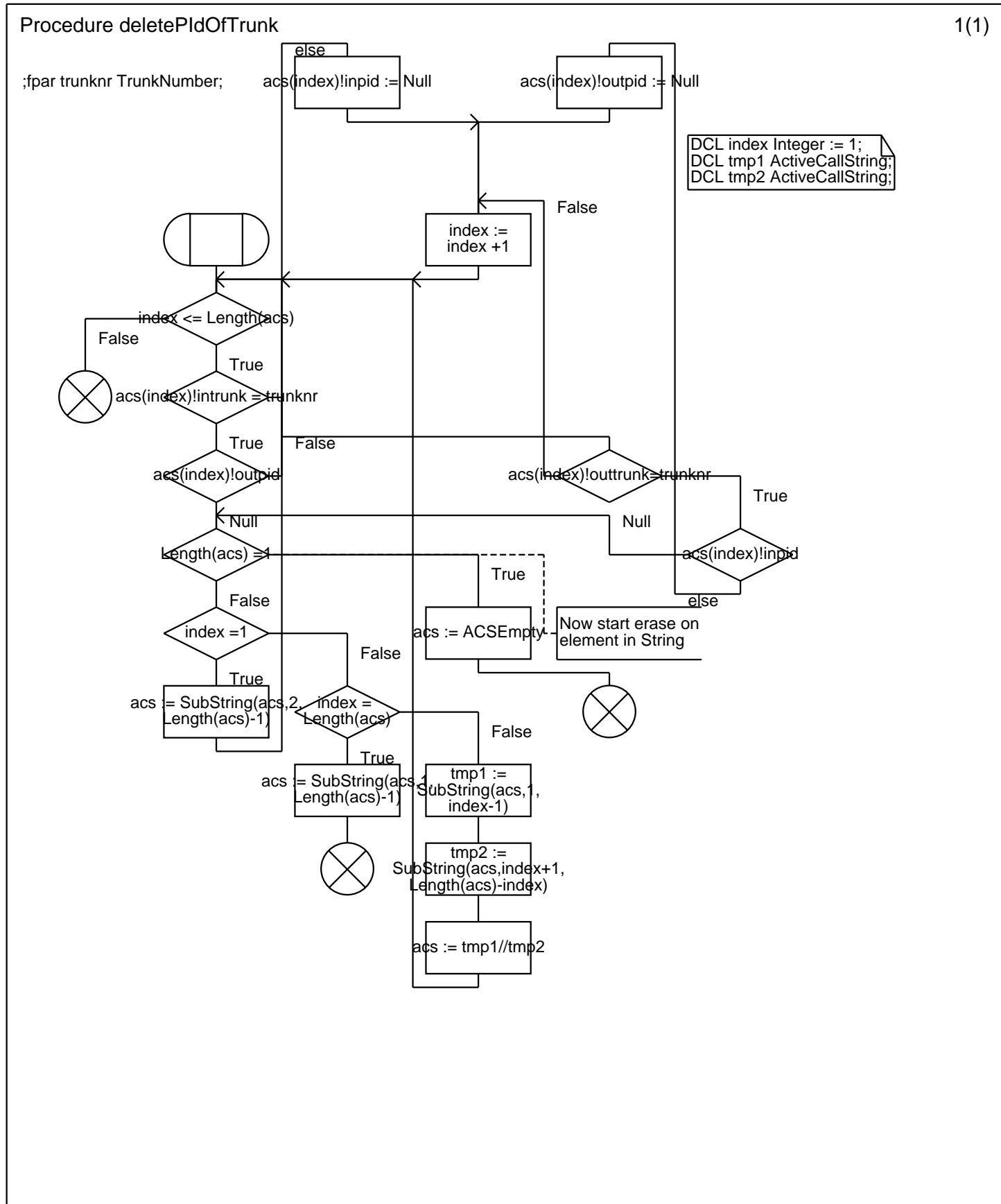
;fpar in/out crstr CRString;
returns PldString;



```

DCL pidstr PldString := PldStringEmpty;
DCL index1 Integer := 0;
DCL index2 Integer := 0;
DCL iopid PlId := Null;
DCL iocr CallReference;
DCL takein Boolean;
    
```

Annex B: deletePlidOfTrunk

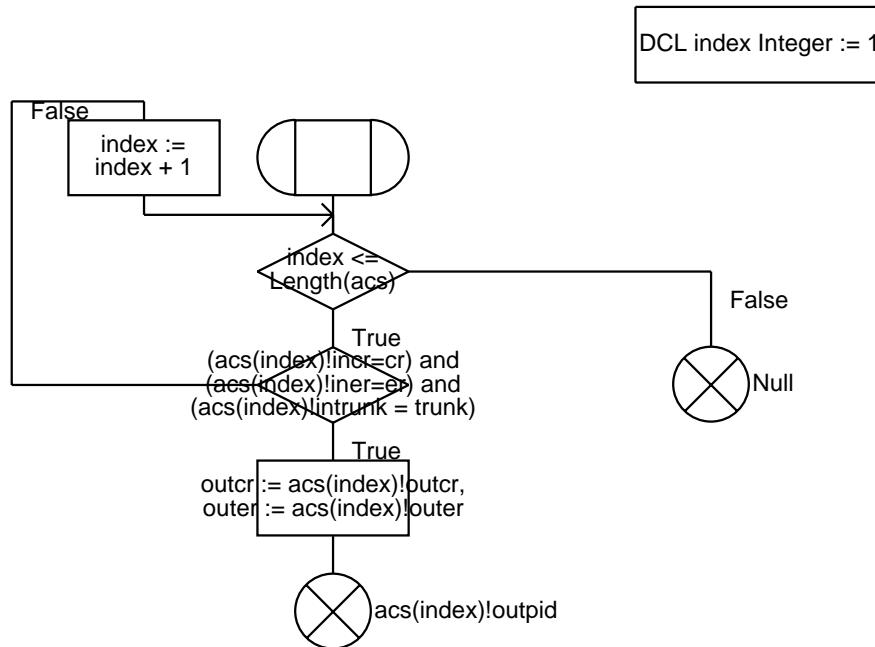


Annex B: firstOutPIdOfCRERTrunk

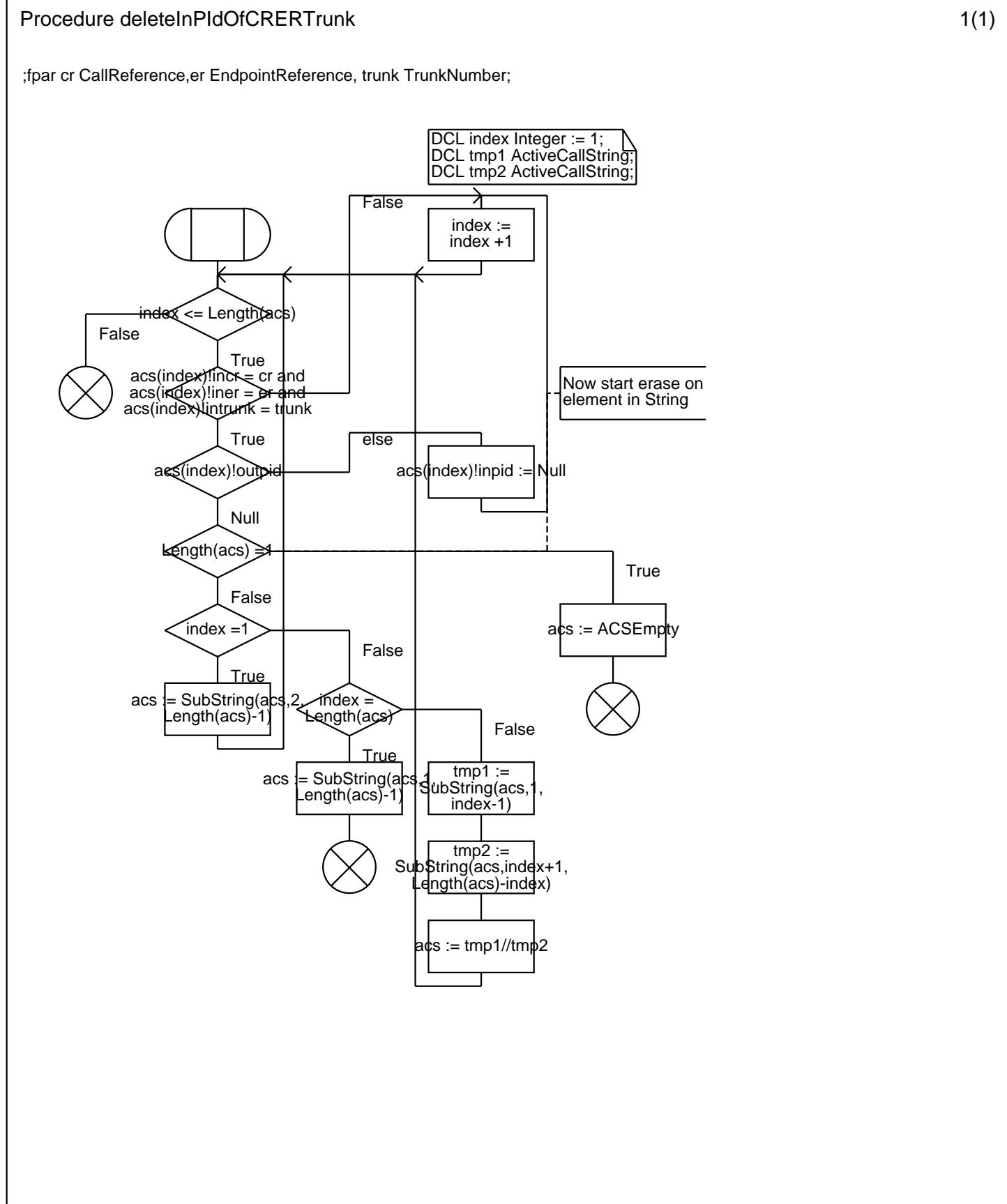
Procedure firstOutPIdOfCRERTrunk

1(1)

;fpar cr CallReference,er EndpointReference,trunk TrunkNumber,in/out outcr CallReference, in/out outer EndpointReference;
returns PId;



Annex B: deleteInPlidOfCRERTrunk



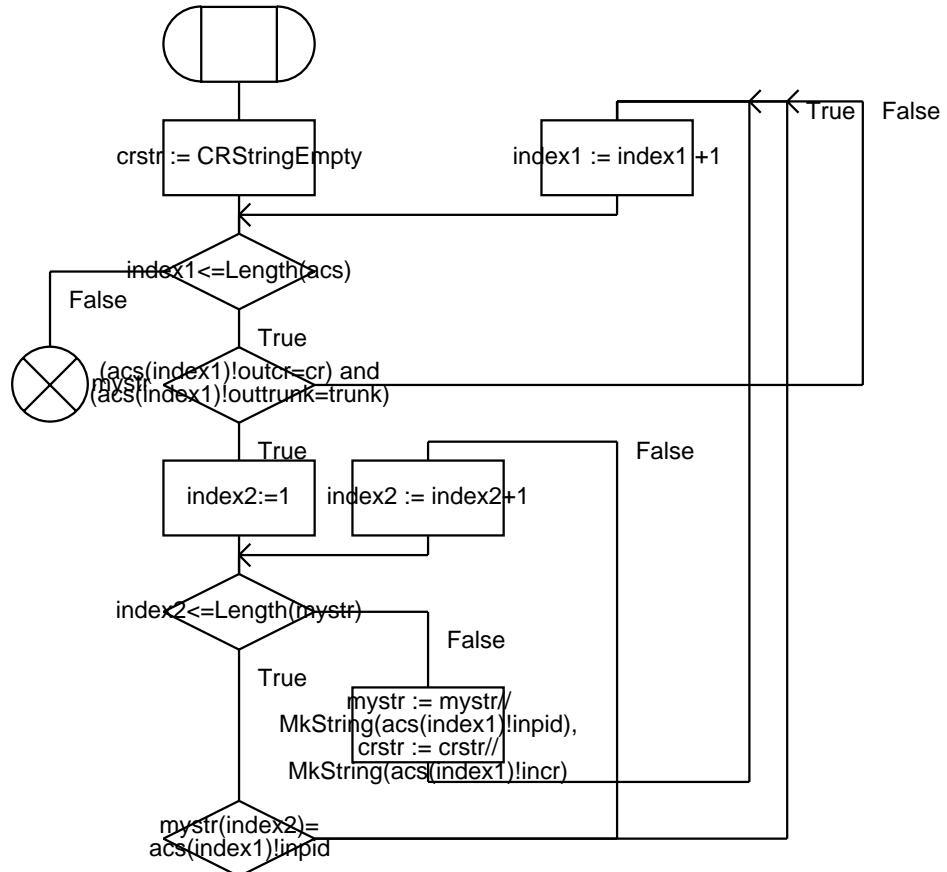
Annex B: allInPIdOfCRTTrunk

Procedure allInPIdOfCRTTrunk

1(1)

;fpar cr CallReference, trunk TrunkNumber, in/out crstr CRString;
returns PIdString;

```
DCL mystr PIdString := PIdStringEmpty;
DCL index1 Integer := 1;
DCL index2 Integer := 2;
```

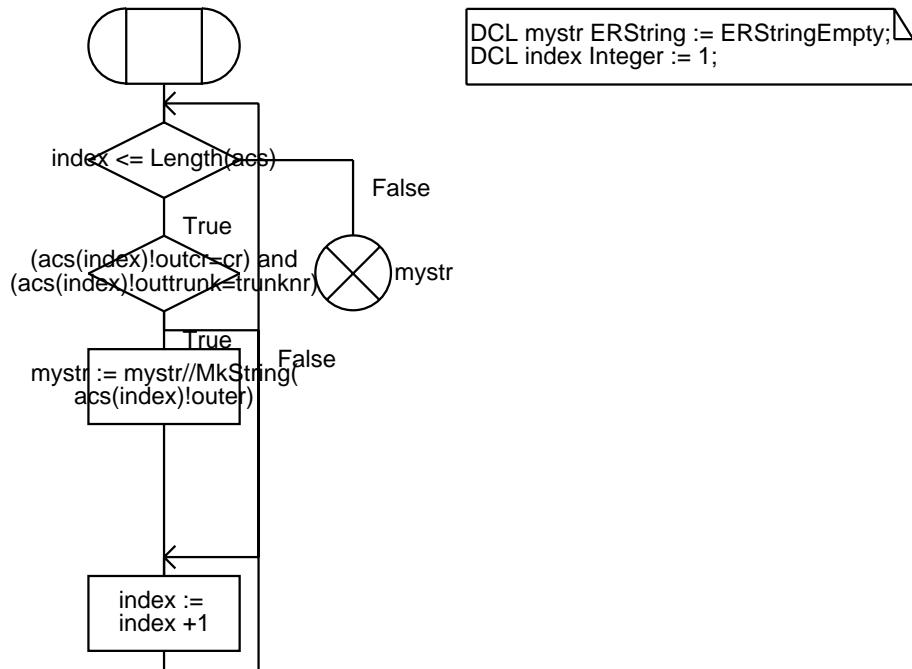


Annex B: allEROfOutCRTTrunk

Procedure allEROfOutCRTTrunk

1(1)

;fpar cr CallReference,trunknr TrunkNumber;
returns ERString;



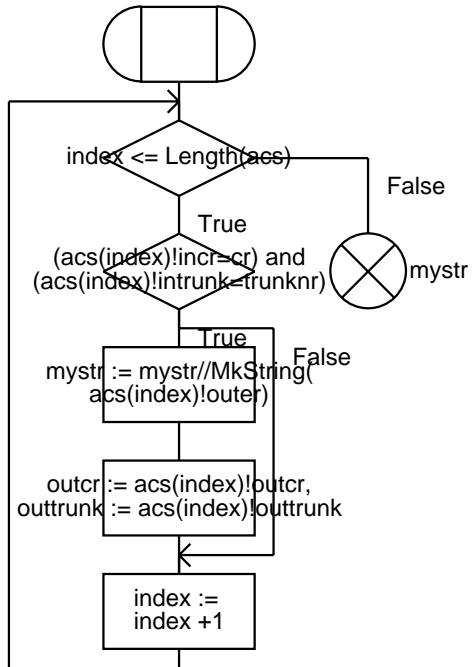
Annex B: allEROfCRTTrunk

Procedure allEROfCRTTrunk

1(1)

```
;fpar cr CallReference,trunknr TrunkNumber,
in/out outcr CallReference,in/out outrunk TrunkNumber;
returns ERString;
```

```
DCL mystr ERString := ERStringEmpty;
DCL index Integer := 1;
```



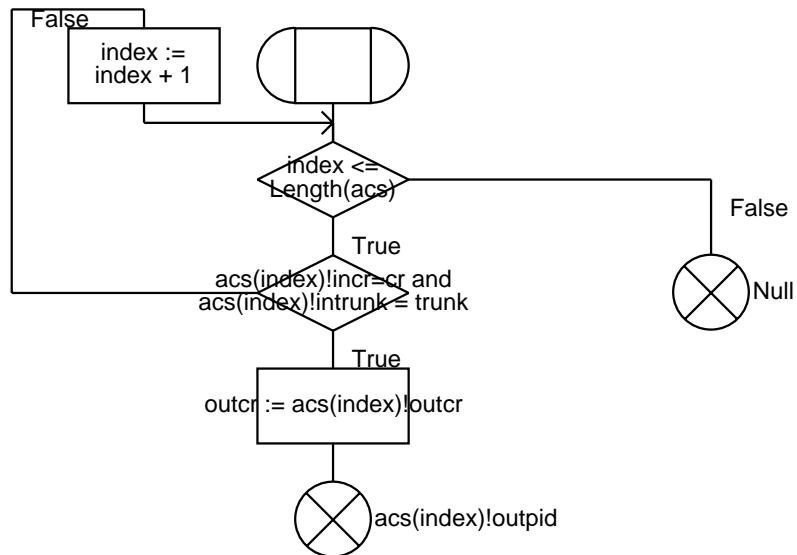
Annex B: firstOutPIdOfInCRTTrunk

Procedure firstOutPIdOfInCRTTrunk

1(1)

;fpar cr CallReference, trunk TrunkNumber ,in/out outcr CallReference;
returns PId;

DCL index Integer := 1;

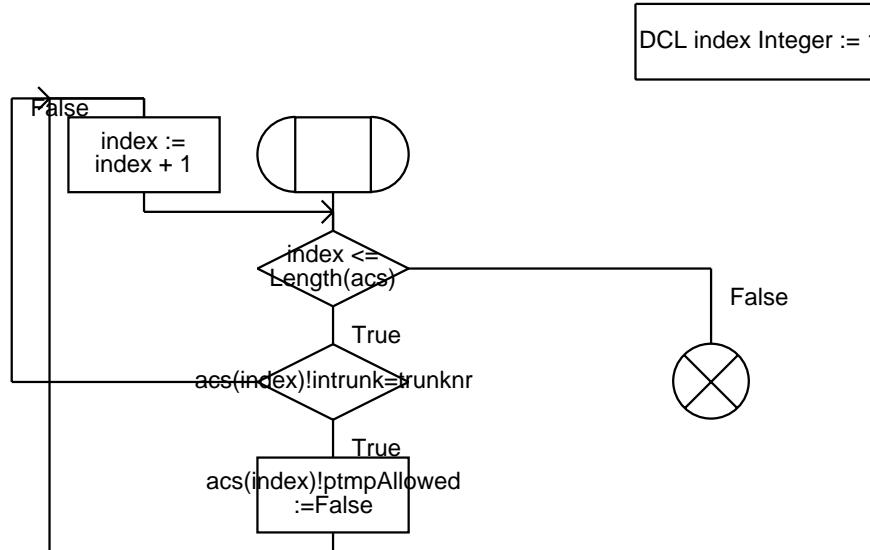


Annex B: changePTMPAllowedOfTrunk

Procedure changePTMPAllowedOfTrunk

1(1)

;fpar trunknr TrunkNumber;



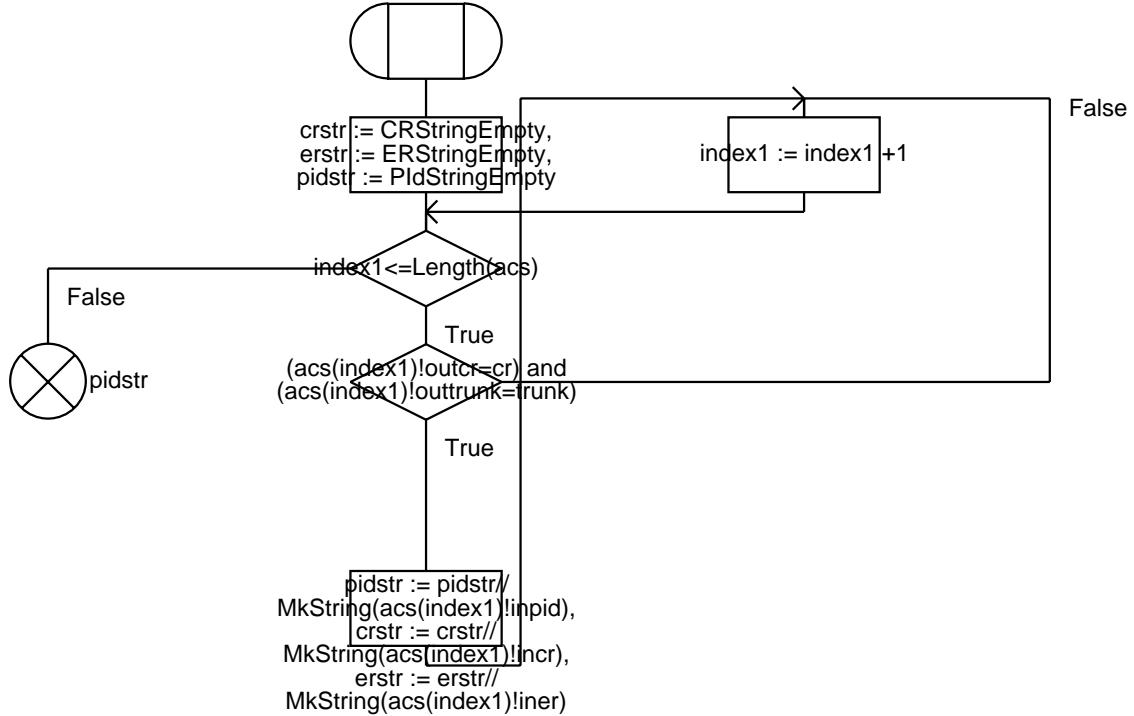
Annex B: allInCRERPidOfCRTTrunk

Procedure allInCRERPidOfCRTTrunk

1(1)

```
;fpar cr CallReference, trunk TrunkNumber,
in/out crstr CRString,in/out erstr ERString;
returns PIdString;
```

```
DCL pidstr PIdString := PIdStringEmpty;
DCL index1 Integer := 1;
```

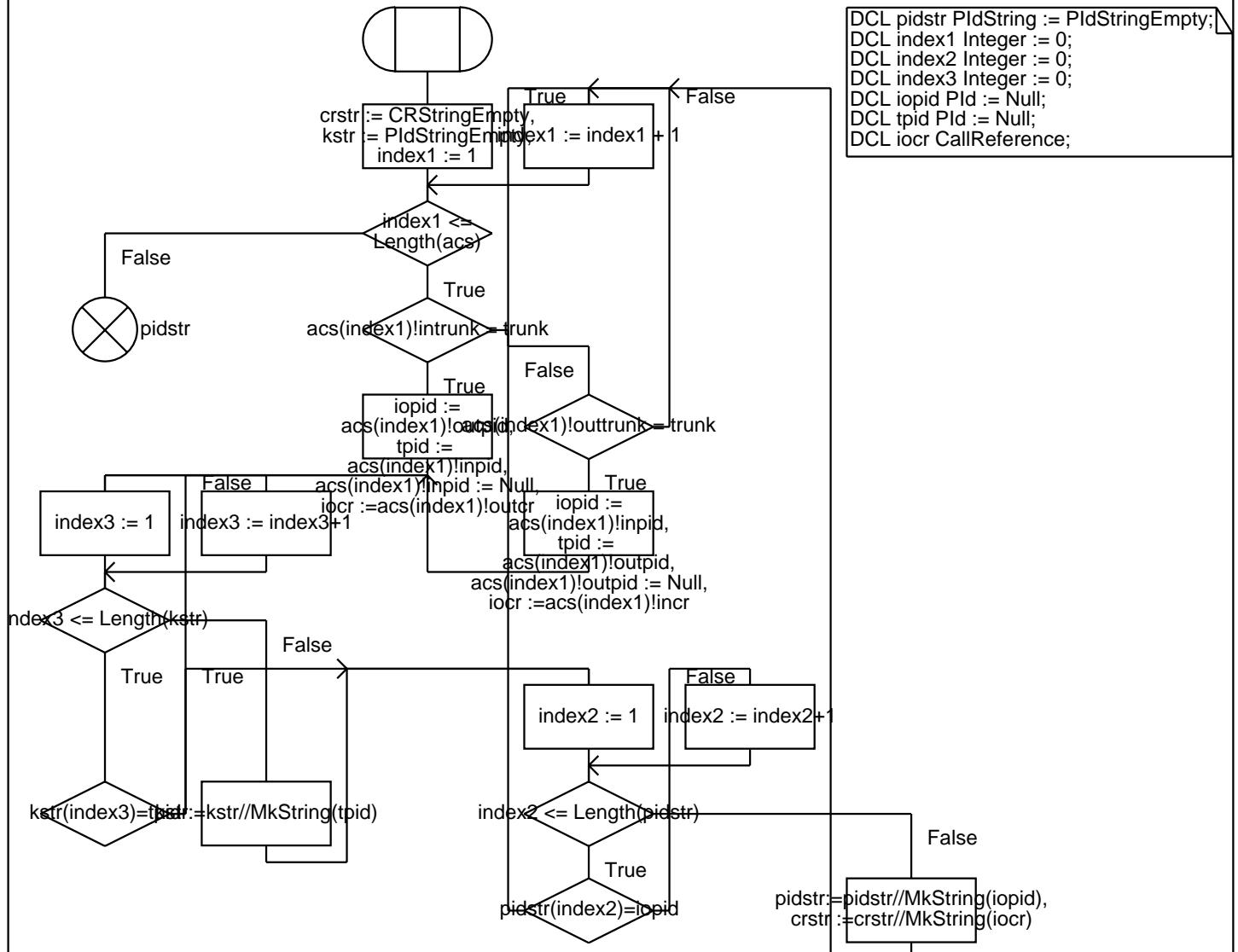


Annex B: allPIdOfNotTrunk

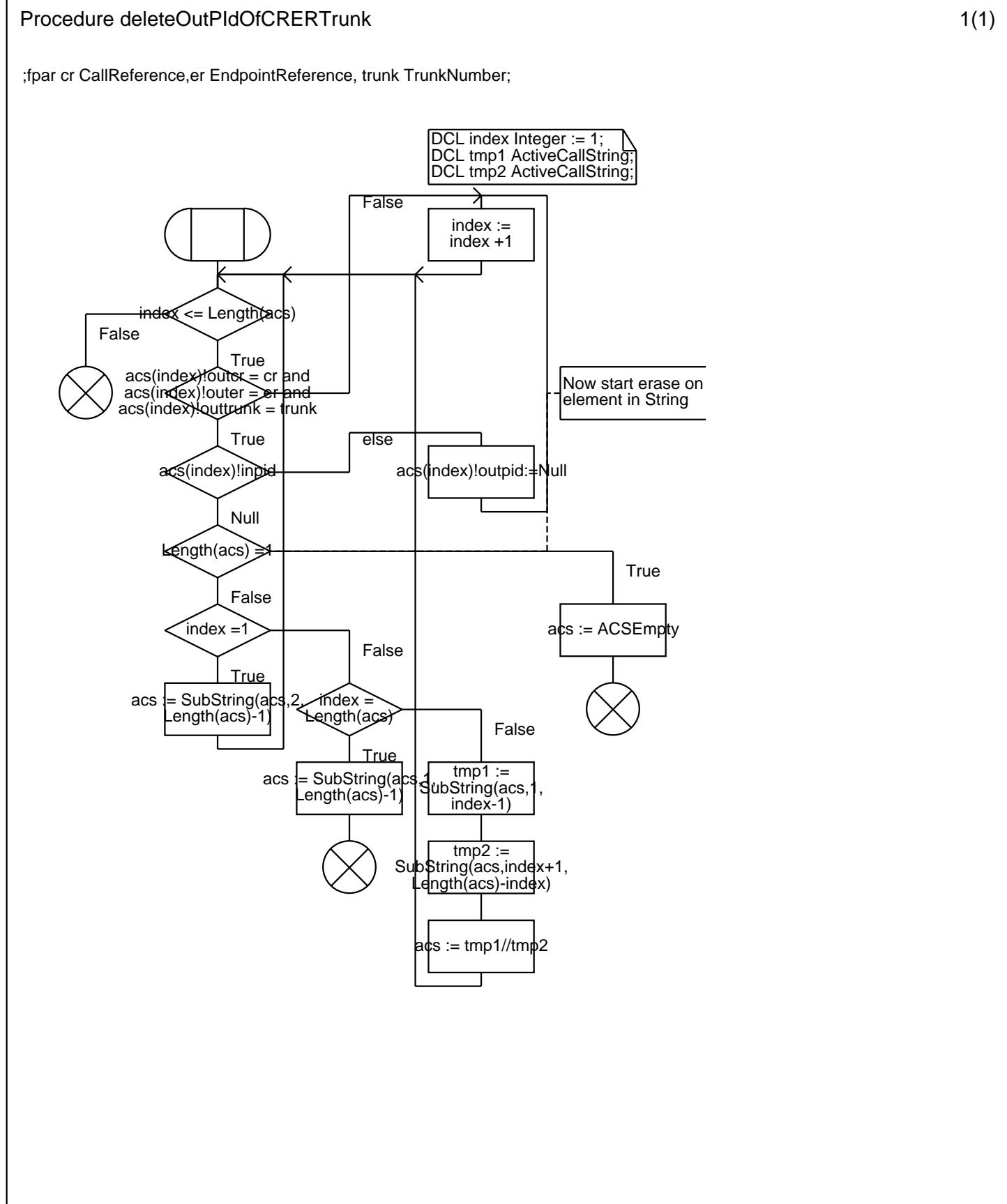
Procedure allPIdOfNotTrunk

1(1)

;fpar trunk TrunkNumber, in/out crstr CRString, in/out kstr PIdString;
returns PIdString;



Annex B: deleteOutPlidOfCRERTrunk



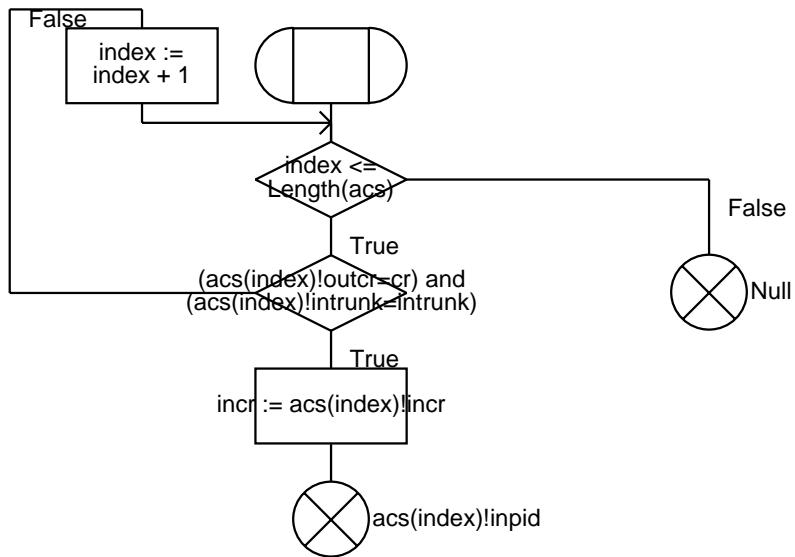
Annex B: firstInPlidOfOutCRInTrunk

Procedure firstInPlidOfOutCRInTrunk

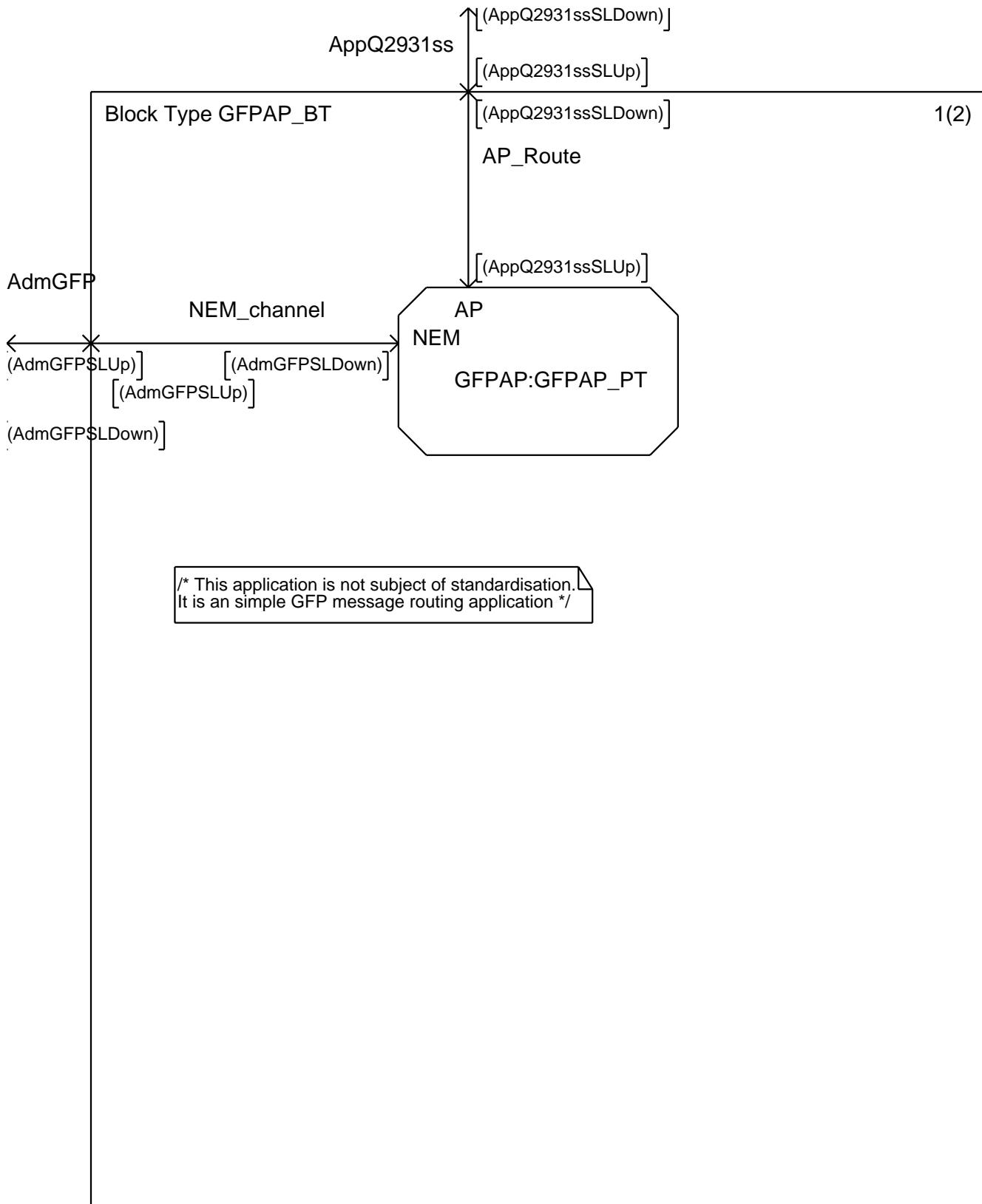
1(1)

;fpar cr CallReference, intrunk TrunkNumber,in/out incr CallReference;
returns Plid;

DCL index Integer := 1;



Annex B: GFPAP_BT



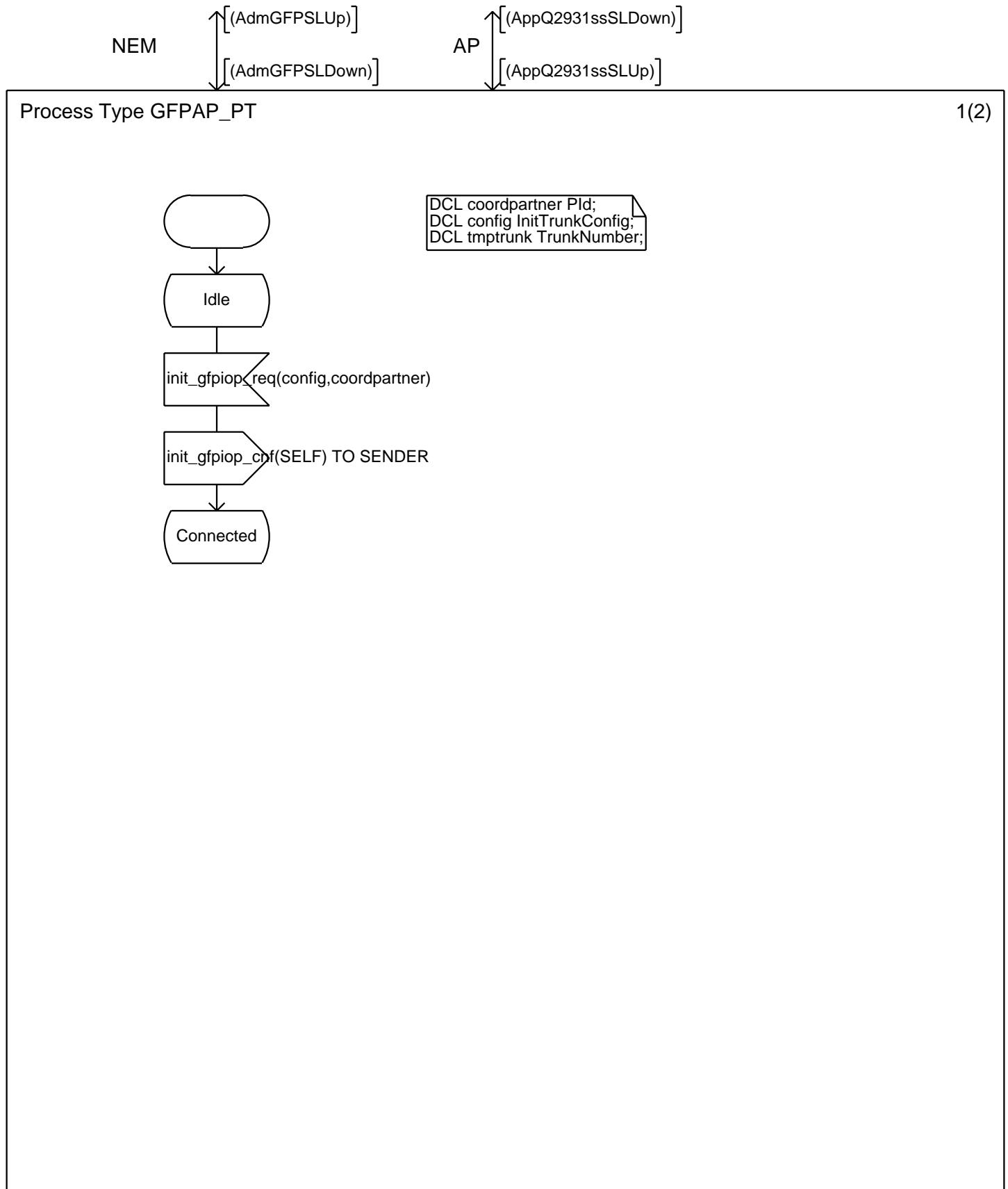
Annex B: GFPAP_BT

Block Type GFPAP_BT

2(2)



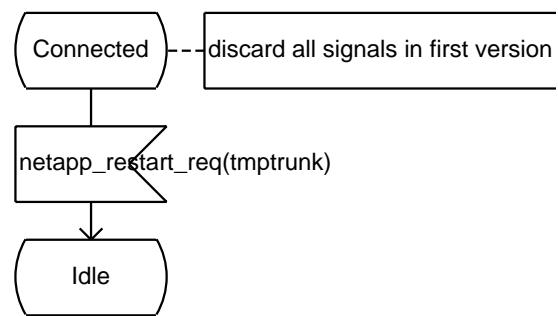
Annex B: GFPAP_PT



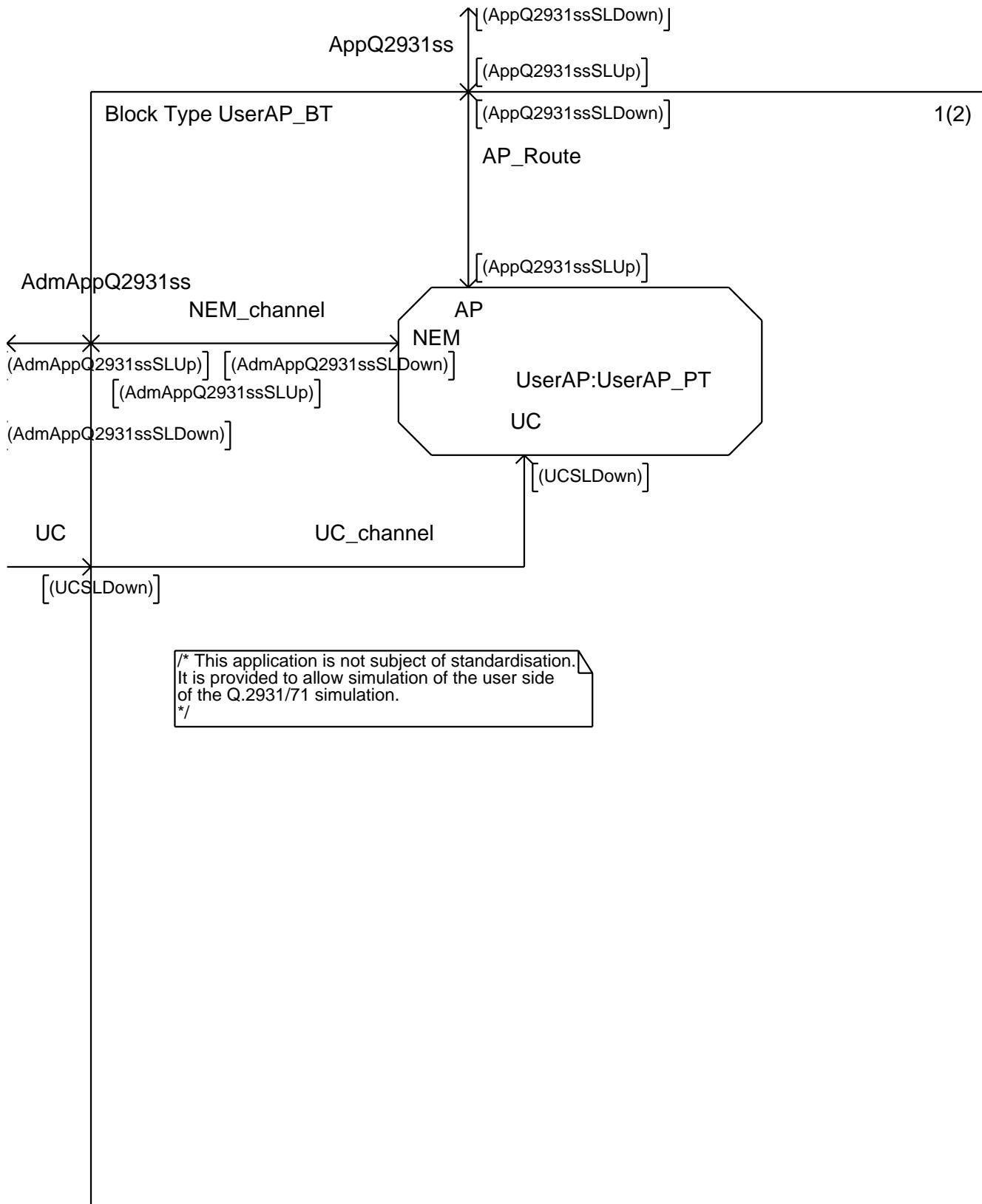
Annex B: GFPAP_PT

Process Type GFPAP_PT

2(2)



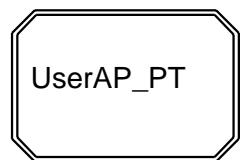
Annex B: UserAP_BT



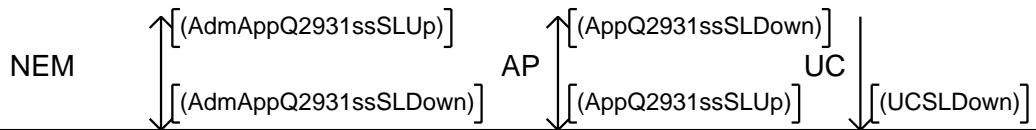
Annex B: UserAP_BT

Block Type UserAP_BT

2(2)



Annex B: UserAP_PT



Process Type UserAP_PT

1(30)

```
DCL coord PId;
DCL uicc PId;
DCL p1cr CallReference;
DCL p2cr CallReference;
DCL p1er EndpointReference;
DCL p2er EndpointReference;
DCL outcrvalue Integer := 0;
DCL outervalue Integer := 0;
DCL AwaitAlertingAndSendAP2 Boolean := False;
```

Annex B: UserAP_PT

Process Type UserAP_PT

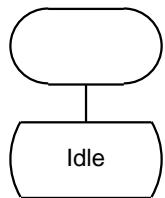
2(30)

```
/* Variable valid only during one transition */
DCL msg Q2931ssMessage;
DCL defmsg Q2931ssMessage;
DCL lastsetup Q2931ssMessage;
DCL outmsg Q2931ssMessage;
DCL outmsgClear Q2931ssMessage;
DCL outmsgDropAck Q2931ssMessage;
DCL config InitTrunkConfig;
DCL cr CallReference;
DCL er EndpointReference;
DCL pid PId;
DCL gloc Location;
DCL lastSender PId;
DCL intrunk TrunkNumber;
DCL trunkNumber TrunkNumber;
DCL atrunk TrunkNumber;
DCL str Charstring;
```

Annex B: UserAP_PT

Process Type UserAP_PT

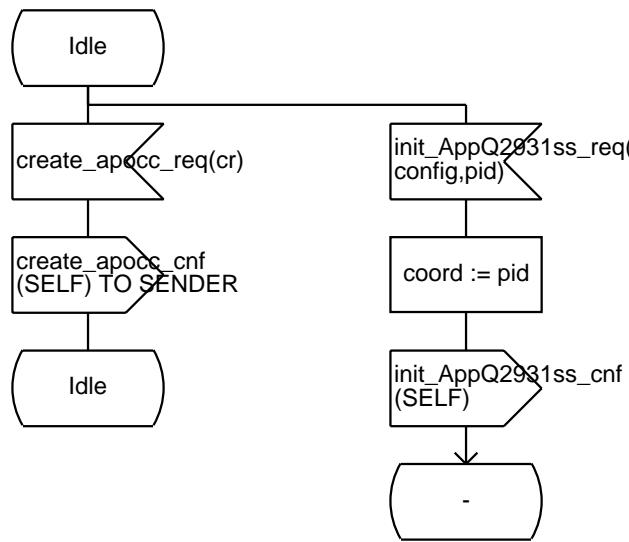
3(30)



Annex B: UserAP_PT

Process Type UserAP_PT

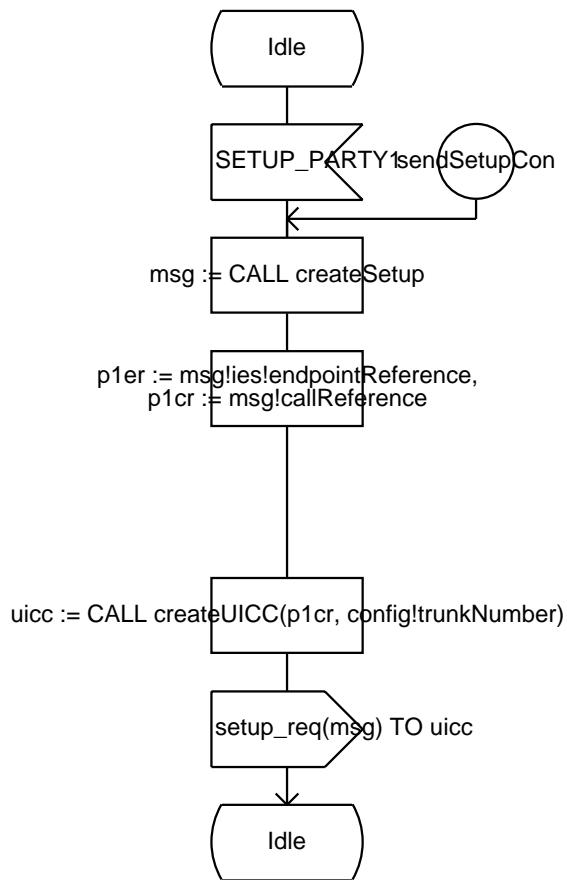
4(30)



Annex B: UserAP_PT

Process Type UserAP_PT

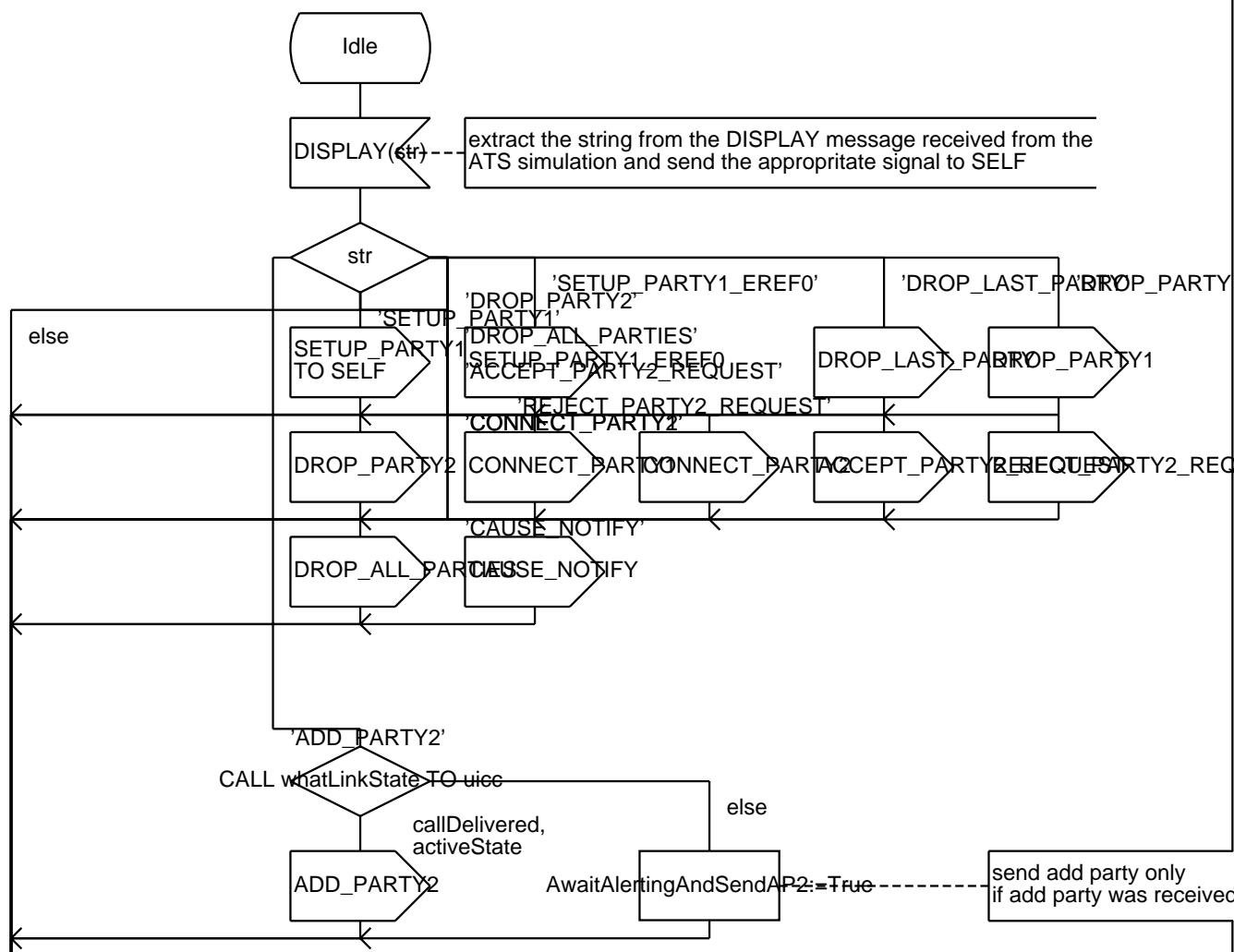
5(30)



Annex B: UserAP_PT

Process Type UserAP_PT

6(30)

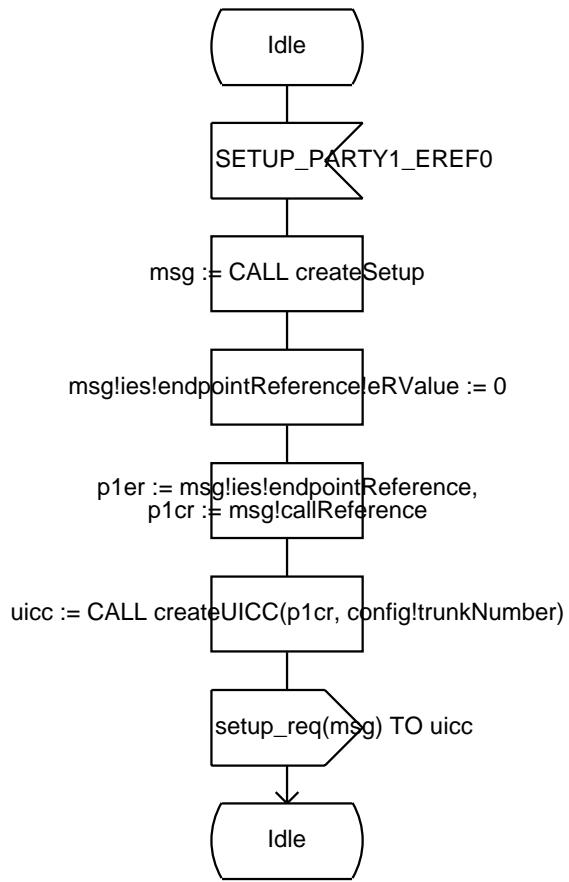


Idle

Annex B: UserAP_PT

Process Type UserAP_PT

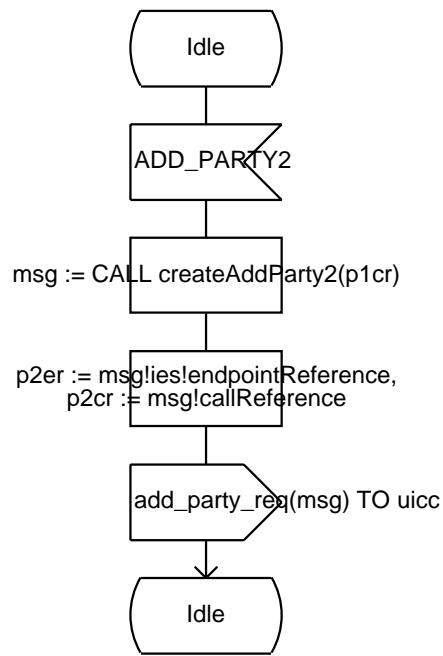
7(30)



Annex B: UserAP_PT

Process Type UserAP_PT

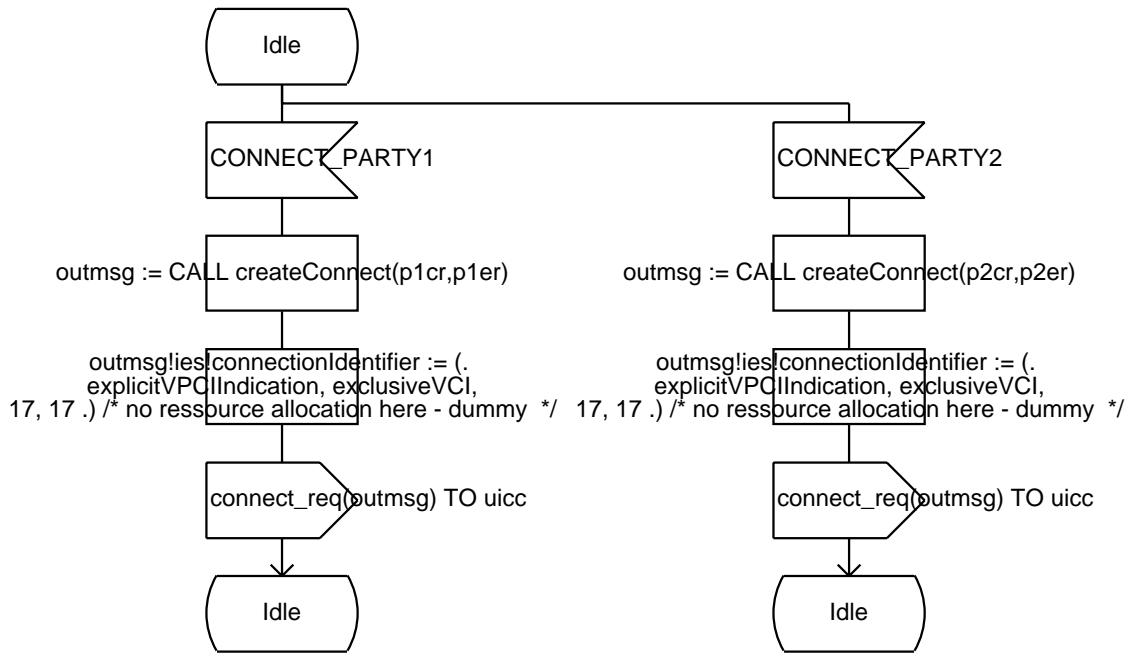
8(30)



Annex B: UserAP_PT

Process Type UserAP_PT

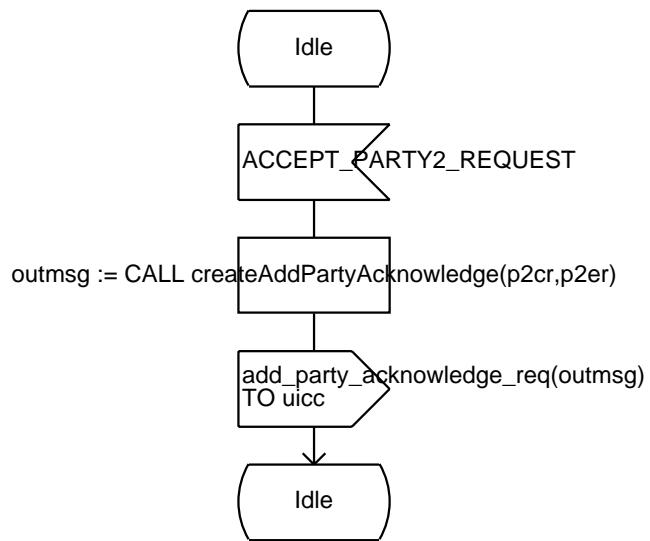
9(30)



Annex B: UserAP_PT

Process Type UserAP_PT

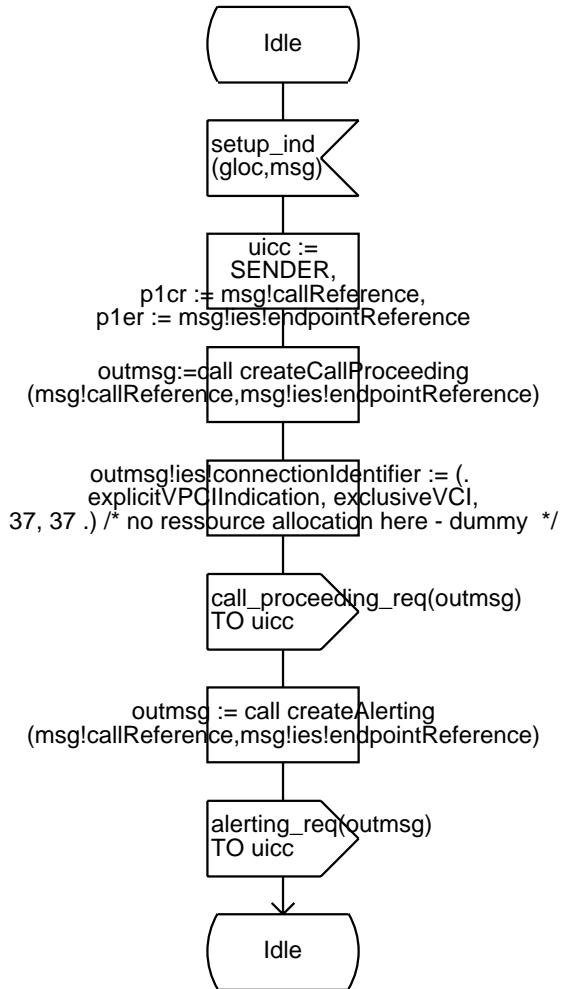
10(30)



Annex B: UserAP_PT

Process Type UserAP_PT

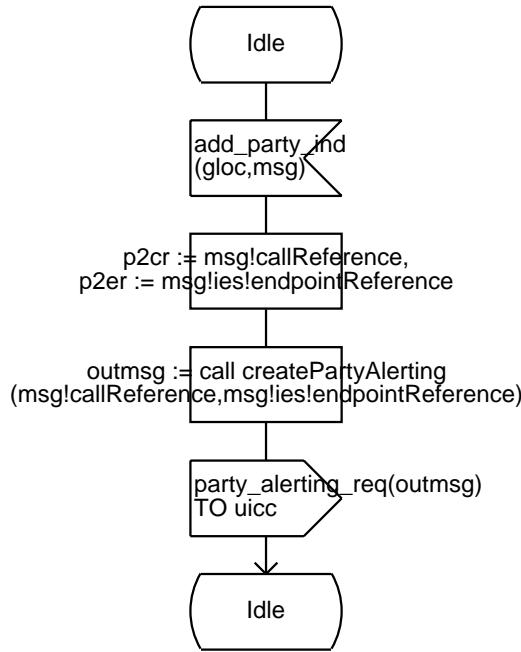
11(30)



Annex B: UserAP_PT

Process Type UserAP_PT

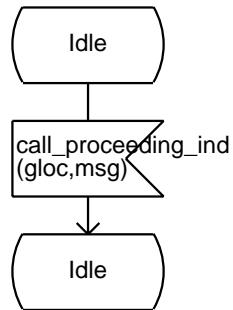
12(30)



Annex B: UserAP_PT

Process Type UserAP_PT

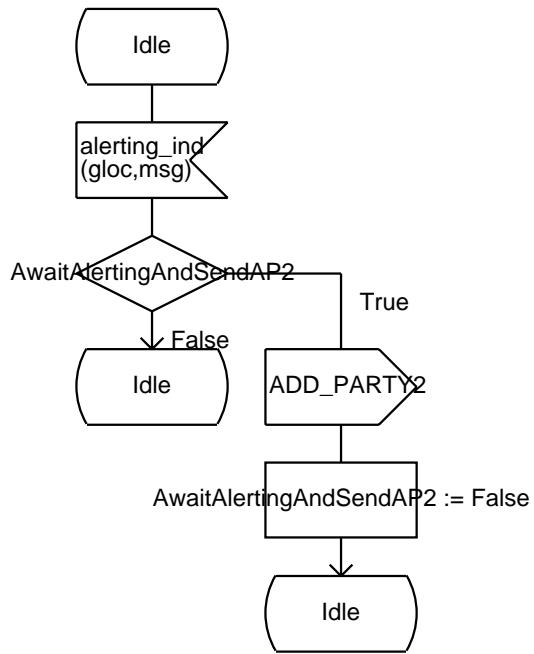
13(30)



Annex B: UserAP_PT

Process Type UserAP_PT

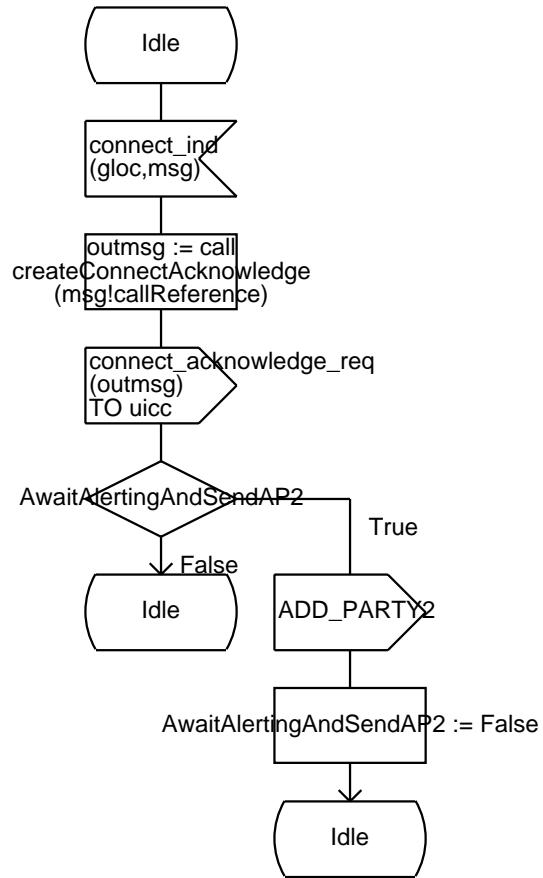
14(30)



Annex B: UserAP_PT

Process Type UserAP_PT

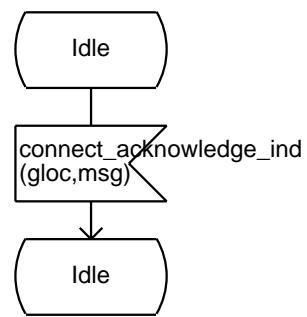
15(30)



Annex B: UserAP_PT

Process Type UserAP_PT

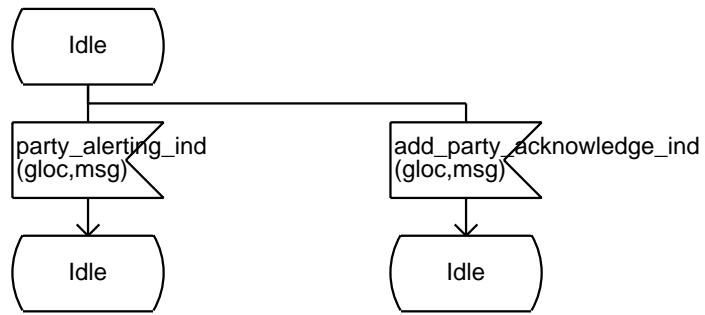
16(30)



Annex B: UserAP_PT

Process Type UserAP_PT

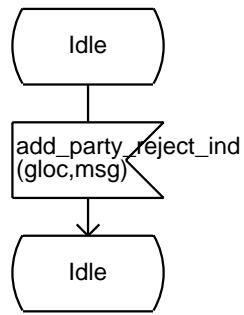
17(30)



Annex B: UserAP_PT

Process Type UserAP_PT

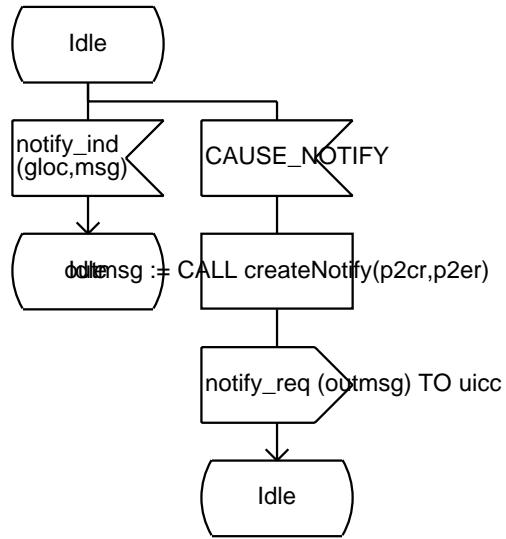
18(30)



Annex B: UserAP_PT

Process Type UserAP_PT

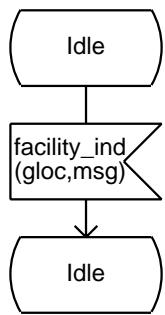
19(30)



Annex B: UserAP_PT

Process Type UserAP_PT

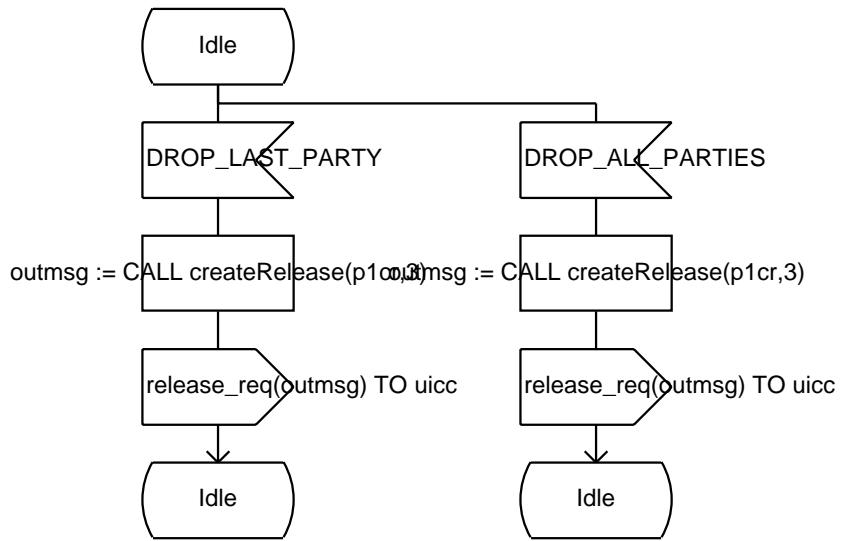
20(30)



Annex B: UserAP_PT

Process Type UserAP_PT

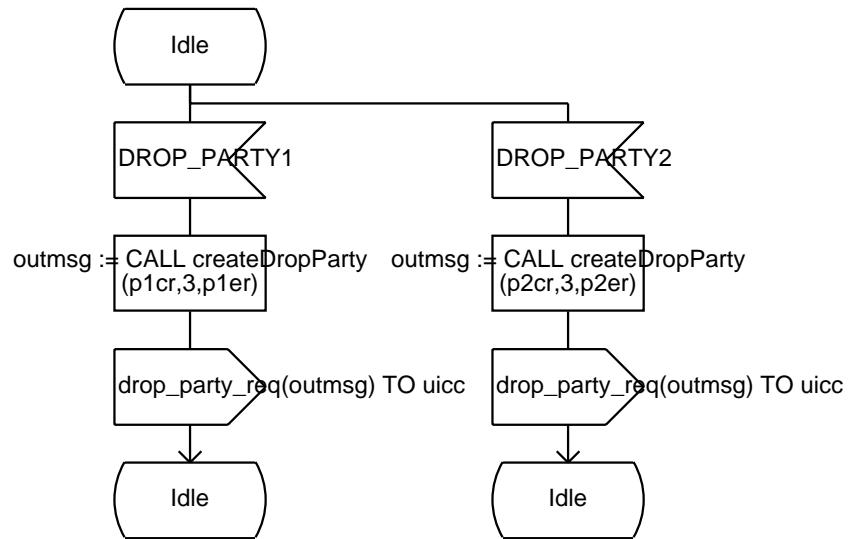
21(30)



Annex B: UserAP_PT

Process Type UserAP_PT

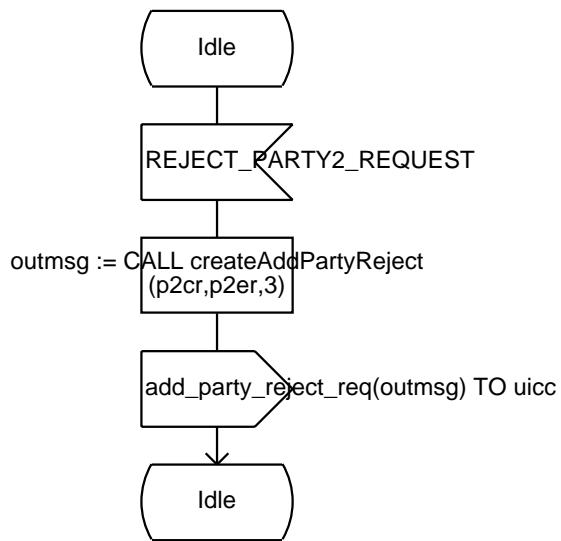
22(30)



Annex B: UserAP_PT

Process Type UserAP_PT

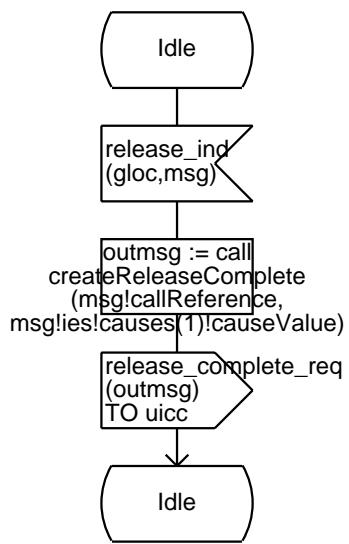
23(30)



Annex B: UserAP_PT

Process Type UserAP_PT

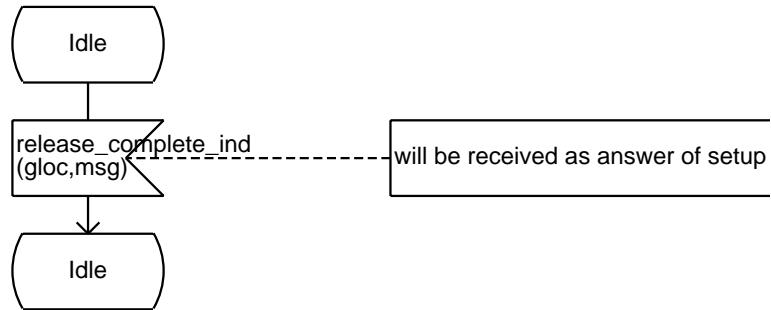
24(30)



Annex B: UserAP_PT

Process Type UserAP_PT

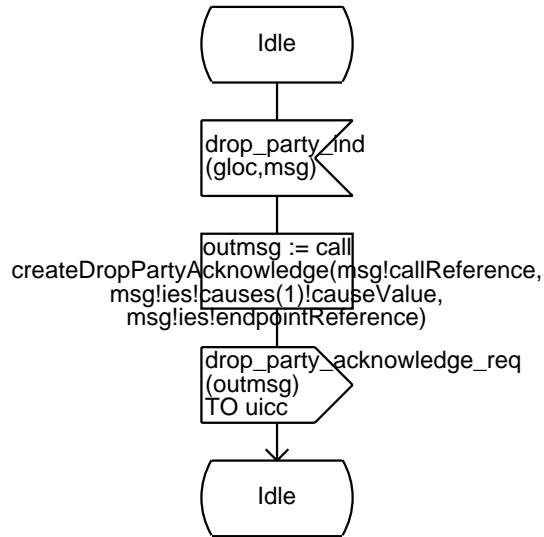
25(30)



Annex B: UserAP_PT

Process Type UserAP_PT

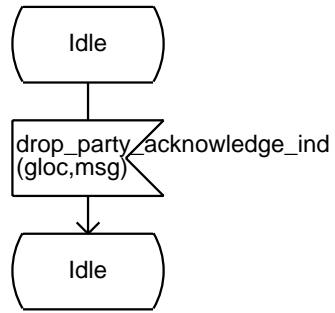
26(30)



Annex B: UserAP_PT

Process Type UserAP_PT

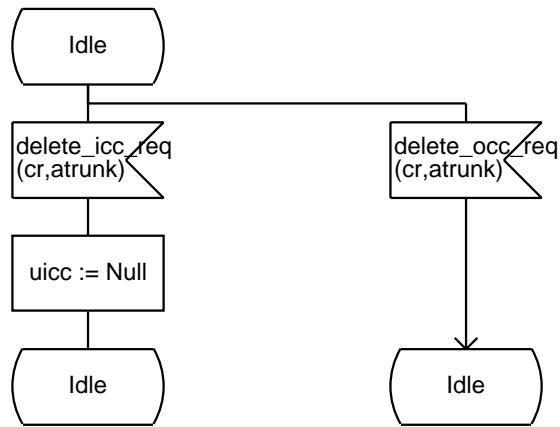
27(30)



Annex B: UserAP_PT

Process Type UserAP_PT

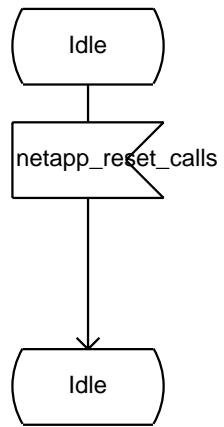
28(30)



Annex B: UserAP_PT

Process Type UserAP_PT

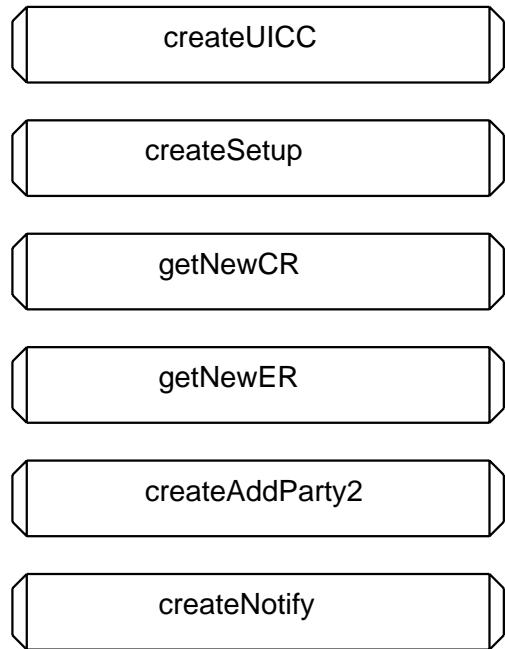
29(30)



Annex B: UserAP_PT

Process Type UserAP_PT

30(30)



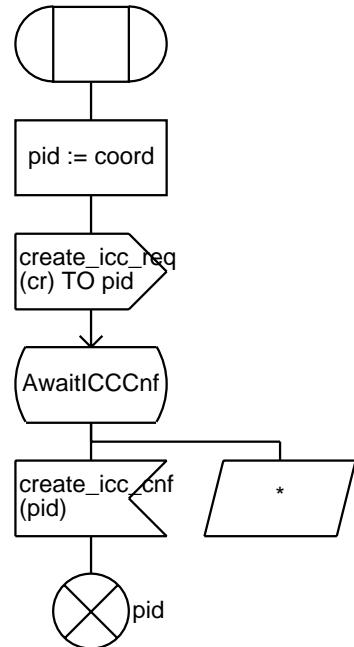
Annex B: createUICC

Procedure createUICC

1(1)

```
;fpar cr CallReference,trunk TrunkNumber;  
returns PId;
```

```
DCL pid PId;
```



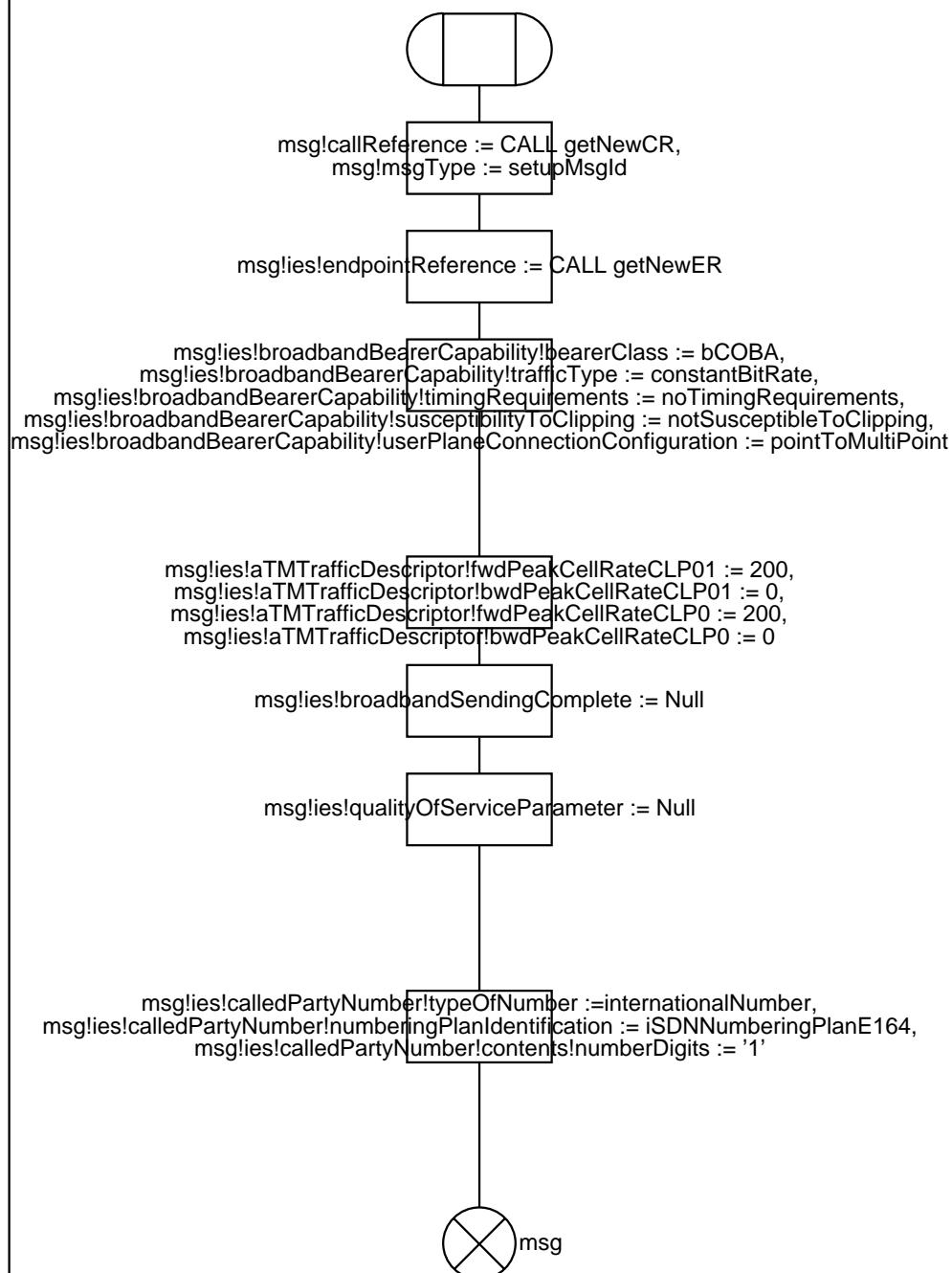
Annex B: createSetup

Procedure createSetup

1(1)

;RETURNS Q2931ssMessage;

DCL msg Q2931ssMessage;

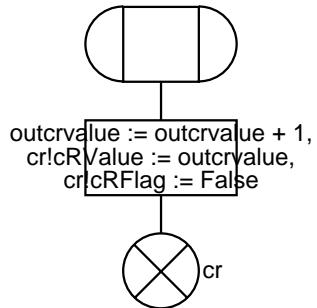


Annex B: getNewCR

Procedure <<Process Type UserAP_PT>> getNewCR

1(1)

:RETURNS CallReference;



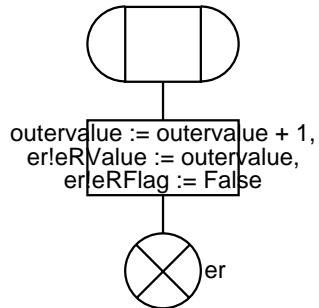
DCL cr CallReference;

Annex B: getNewER

Procedure <<Process Type UserAP_PT>> getNewER

1(1)

;RETURNS EndpointReference;



DCL er EndpointReference;

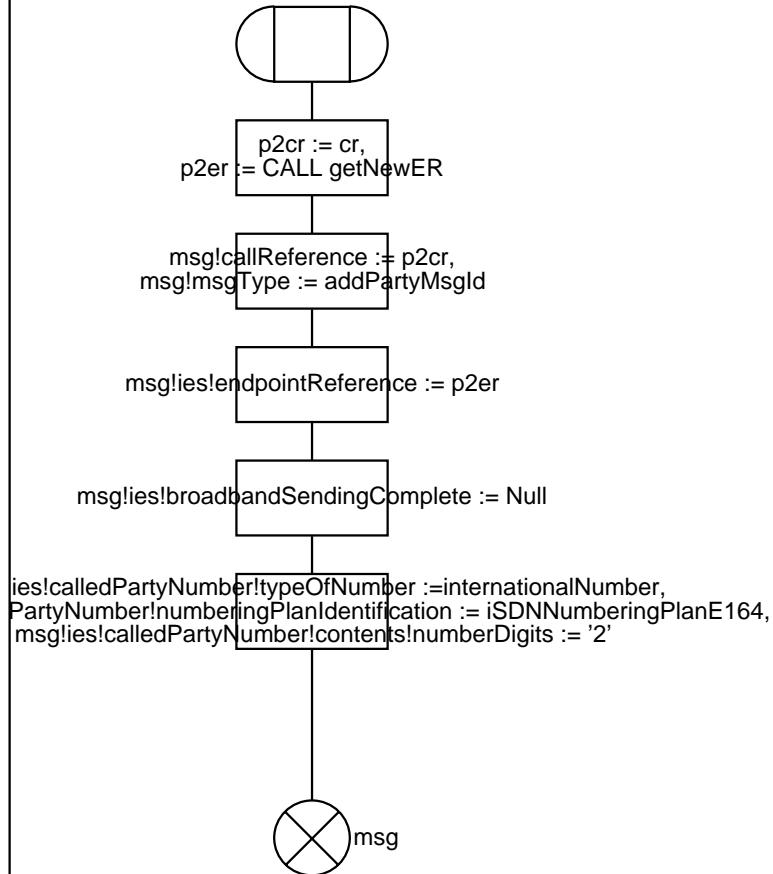
Annex B: createAddParty2

Procedure createAddParty2

1(1)

:FPAR cr CallReference;
 RETURNS Q2931ssMessage;

DCL msg Q2931ssMessage;



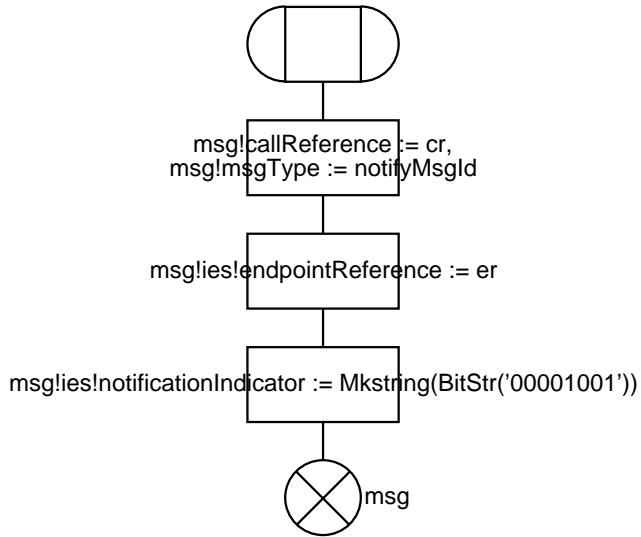
Annex B: createNotify

Procedure createNotify

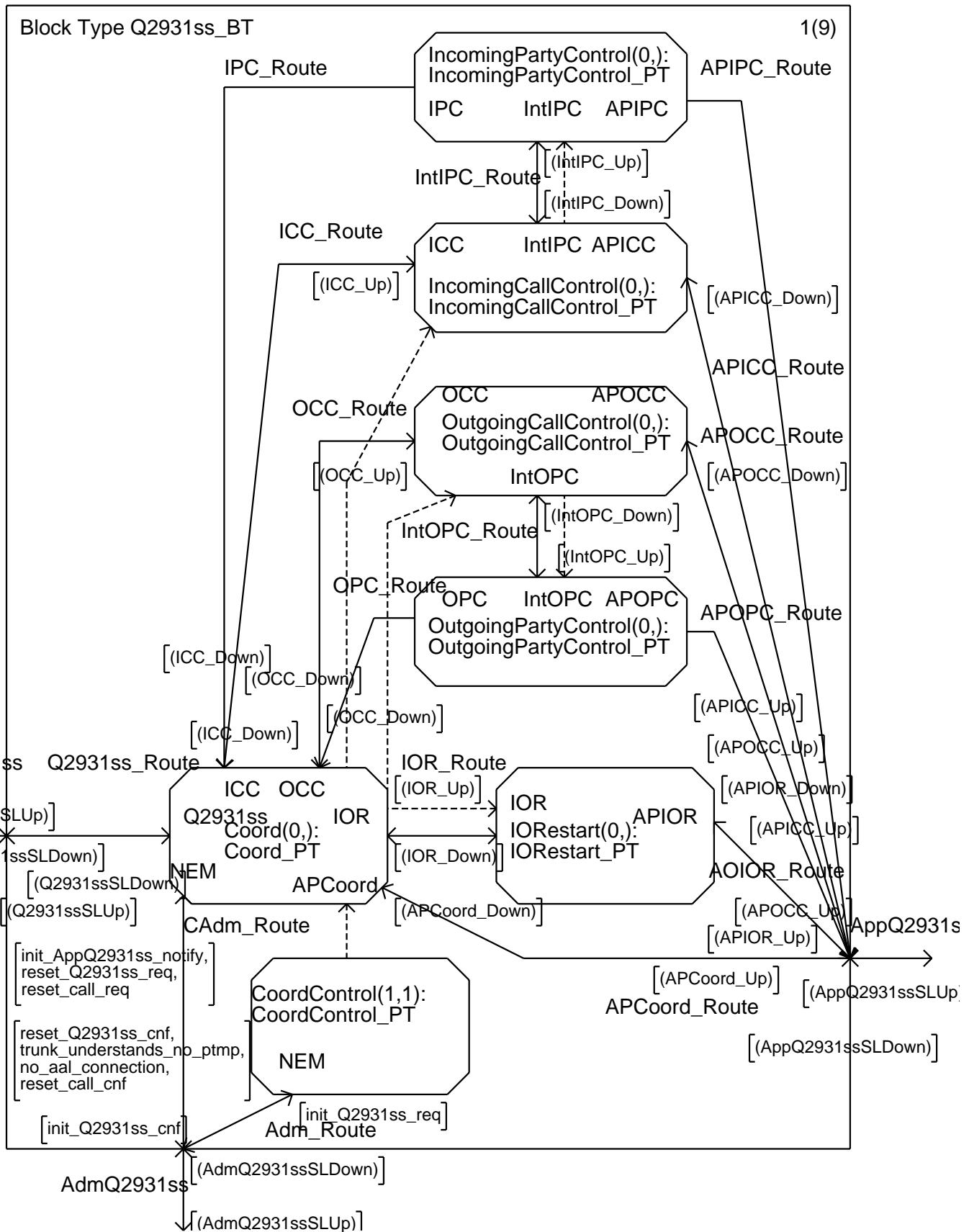
1(1)

:FPAR cr CallReference, er EndpointReference;
RETURNS Q2931ssMessage;

DCL msg Q2931ssMessage;



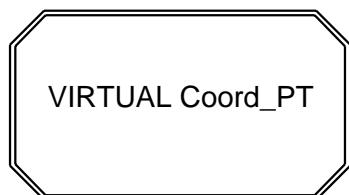
Annex B: Q2931ss_BT



Annex B: Q2931ss_BT

Block Type Q2931ss_BT

2(9)



VIRTUAL Coord_PT



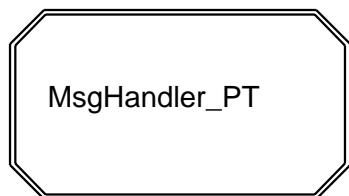
IncomingCallControl_PT



CoordControl_PT



IncomingPartyControl_PT



MsgHandler_PT



OutgoingCallControl_PT



PartyControl_PT



OutgoingPartyControl_PT



CallControl_PT



IORestart_PT

Annex B: Q2931ss_BT

Block Type Q2931ss_BT

3(9)

```
SYNONYM startTime301 Boolean = True;
```

Annex B: Q2931ss_BT

Block Type Q2931ss_BT

4(9)

```

SIGNALLIST ICC_Down=
/* from ICC to Coord */
delete_icc_req,
AAL_ESTABLISHreq,
SETUPreq,
ADD_PARTYreq,
DROP_PARTYreq,
DROP_PARTY_ACKNOWLEDGEreq,
CONNECT_ACKNOWLEDGEreq,
RELEASEreq,
RELEASE_COMPLETEreq,
STATUS_ENQUIRYreq,
STATUSreq,
trunk_understands_no_ptmp,
NOTIFYreq,
FACILITYreq;

SIGNALLIST ICC_Up=
/* from Coord to ICC */
AAL_ESTABLISHreq,
AAL_RELEASEind,
AAL_ESTABLISHcnf,
AAL_ESTABLISHind,
AAL_ESTABLISH_ERROR,
ADD_PARTYind,
SETUPind,
CONNECT_ACKNOWLEDGEind,
ADD_PARTY_ACKNOWLEDGEind,
ADD_PARTY_REJECTind,
ALERTINGind,
CALL_PROCEEDINGind,
PARTY_ALERTINGind,
CONNECTind,
RELEASEind,
RELEASE_COMPLETEind,
DROP_PARTYind,
DROP_PARTY_ACKNOWLEDGEind,
STATUS_ENQUIRYind,
STATUSind,
NOTIFYind,
UNKNOWNind,
FACILITYind,
kill_call;

```

```

SIGNALLIST OCC_Down =
/* from OCC to Coord */
delete_occ_req,
AAL_ESTABLISHreq,
PARTY_ALERTINGreq,
CALL_PROCEEDINGreq,
ALERTINGreq,
ADD_PARTY_ACKNOWLEDGEreq,
ADD_PARTY_REJECTreq,
CONNECTreq,
RELEASEreq,
RELEASE_COMPLETEreq,
DROP_PARTYreq,
DROP_PARTY_ACKNOWLEDGEreq,
STATUS_ENQUIRYreq,
STATUSreq,
trunk_understands_no_ptmp,
NOTIFYreq,
FACILITYreq;

SIGNALLIST OCC_Up =
/* from Coord to OCC */
AAL_ESTABLISHreq,
AAL_RELEASEind,
AAL_ESTABLISHcnf,
AAL_ESTABLISHind,
AAL_ESTABLISH_ERROR,
ADD_PARTYind,
SETUPind,
CONNECT_ACKNOWLEDGEind,
ADD_PARTY_ACKNOWLEDGEind,
ADD_PARTY_REJECTind,
ALERTINGind,
CALL_PROCEEDINGind,
PARTY_ALERTINGind,
CONNECTind,
RELEASEind,
RELEASE_COMPLETEind,
DROP_PARTYind,
DROP_PARTY_ACKNOWLEDGEind,
STATUS_ENQUIRYind,
STATUSind,
NOTIFYind,
UNKNOWNind,
FACILITYind,
kill_call;

```

Annex B: Q2931ss_BT

Block Type Q2931ss_BT

5(9)

```

SIGNALLIST IntOPC_Up =
/* from OCC to OPC */
SETUPind,
RELEASEind,
DROP_PARTYEXTind,
alerting_req,
connect_req,
release_complete_req,
ADD_PARTYEXTind,
party_alerting_req,
add_party_acknowledge_req,
add_party_reject_req,
drop_party_req,
drop_party_acknowledge_req,
release_req,
RELEASEind,
STATUS_ENQUIRYind,
STATUSind,
DROP_PARTY_ACKNOWLEDGEEXTind,
kill_party,
delete_pc_cnf;

SIGNALLIST IntOPC_Down =
delete_pc_req,
release_cnf,
release_req;

```

```

SIGNALLIST IOR_Up =
/* from Coord to restart process */
RESTARTind,
RESTART_ACKNOWLEDGEind,
STATUSind,
kill_call;

SIGNALLIST IOR_Down =
/* from restart process to Coord */
RESTARTreq,
RESTART_ACKNOWLEDGEreq,
STATUSreq;

```

```

SIGNALLIST IntIPC_Up =
/* from ICC to IPC */
setup_req,
RELEASEind,
ALERTINGind,
PARTY_ALERTINGEXTind,
CONNECTind,
DROP_PARTYEXTind,
add_party_req,
ADD_PARTY_ACKNOWLEDGEEXTind,
ADD_PARTY_REJECTEXTind,
drop_party_req,
drop_party_acknowledge_req,
release_req,
RELEASEind,
STATUS_ENQUIRYind,
STATUSind,
DROP_PARTY_ACKNOWLEDGEEXTind,
kill_party,
delete_pc_cnf;

SIGNALLIST IntIPC_Down =
delete_pc_req,
release_cnf,
release_req;

```

Annex B: Q2931ss_BT

Block Type Q2931ss_BT

6(9)

/* signals used internally */

```

SIGNAL
SETUPreq(Q2931ssMessage),
SETUPind(MsgError,Q2931ssMessage),
CALL_PROCEEDINGreq(Q2931ssMessage),
CALL_PROCEEDINGind(MsgError,Q2931ssMessage),
ALERTINGreq(Q2931ssMessage),
ALERTINGind(MsgError,Q2931ssMessage),
CONNECTreq(Q2931ssMessage),
CONNECTind(MsgError,Q2931ssMessage),
CONNECT_ACKNOWLEDGReq(Q2931ssMessage),
CONNECT_ACKNOWLEDGEind(MsgError,Q2931ssMessage),
RELEASEreq(Q2931ssMessage),
RELEASEind(MsgError,Q2931ssMessage),
RELEASE_COMPLETEreq(Q2931ssMessage),
RELEASE_COMPLETEind(MsgError,Q2931ssMessage),
STATUSreq(Q2931ssMessage),
STATUSind(MsgError,Q2931ssMessage),
STATUS_ENQUIRYreq(Q2931ssMessage),
STATUS_ENQUIRYind(MsgError,Q2931ssMessage),
RESTARTreq(Q2931ssMessage),
RESTARTind(MsgError,Q2931ssMessage),
RESTART_ACKNOWLEDGReq(Q2931ssMessage),
RESTART_ACKNOWLEDGEind(MsgError,Q2931ssMessage),
NOTIFYreq(Q2931ssMessage),
NOTIFYind(MsgError,Q2931ssMessage),
UNKNOWNind(MsgError,Q2931ssMessage),
FACILITYind(MsgError,Q2931ssMessage),
FACILITYreq(Q2931ssMessage);

```

```

SIGNAL
PARTY_ALERTINGind(MsgError,Q2931ssMessage),
PARTY_ALERTINGreq(Q2931ssMessage),
ADD_PARTYreq(Q2931ssMessage),
ADD_PARTYind(MsgError,Q2931ssMessage),
ADD_PARTY_ACKNOWLEDGReq(Q2931ssMessage),
ADD_PARTY_ACKNOWLEDGEind(MsgError,Q2931ssMessage),
ADD_PARTY_REJECTreq(Q2931ssMessage),
ADD_PARTY_REJECTind(MsgError,Q2931ssMessage),
DROP_PARTYreq(Q2931ssMessage),
DROP_PARTYind(MsgError,Q2931ssMessage),
DROP_PARTY_ACKNOWLEDGReq(Q2931ssMessage),
DROP_PARTY_ACKNOWLEDGEind(MsgError,Q2931ssMessage),
PARTY_ALERTINGEXTind(MsgError,Q2931ssMessage,CheckResultTypeStatus),
ADD_PARTYEXTind(MsgError,Q2931ssMessage,CheckResultTypeStatus),
ADD_PARTY_ACKNOWLEDGEXTind(MsgError,Q2931ssMessage,CheckResultTypeStatus),
ADD_PARTY_REJECTEXTind(MsgError,Q2931ssMessage,CheckResultTypeStatus),
DROP_PARTYEXTind(MsgError,Q2931ssMessage,CheckResultTypeStatus),
DROP_PARTY_ACKNOWLEDGEXTind(MsgError,Q2931ssMessage,CheckResultTypeStatus);

```

Annex B: Q2931ss_BT

Block Type Q2931ss_BT

7(9)

```
SIGNAL
/* to announce AAL connection state change to OCC, ICC from Coordinator */
AAL_ESTABLISHreq,
AAL_RELEASEind,
AAL_ESTABLISHcnf,
AAL_ESTABLISHind,
AAL_ESTABLISH_ERROR;
```

```
SIGNAL
/* currently not used: announces that trunk does not support Q.2971 */
party_understands_no_ptmp,
release_cnf,
/* for confirmation that
add party reject has been sended in case of
dropping of all parties in add party received state */
delete_pc_req(EndpointReference), /* party process was exited, remove from list */
delete_pc_cnf, /* confirmed operation */
kill_party; /* exit OPC or IPC unconfirmed */
```

Annex B: Q2931ss_BT

Block Type Q2931ss_BT

8(9)

```
/* It is not necessary to define these types here,  
but SDT generates wrong code, when put I put these into  
CallControl_PT */
```

```
/* Dictionary from Endpoint Reference to PId */  
NEWTYPE ERToPId Array(EndpointReference,PId)  
ENDNEWTYPE ERToPId;
```

```
/* String of ER */  
NEWTYPE ERSString String(EndpointReference,ERStringEmpty)  
ENDNEWTYPE;
```

Annex B: Q2931ss_BT

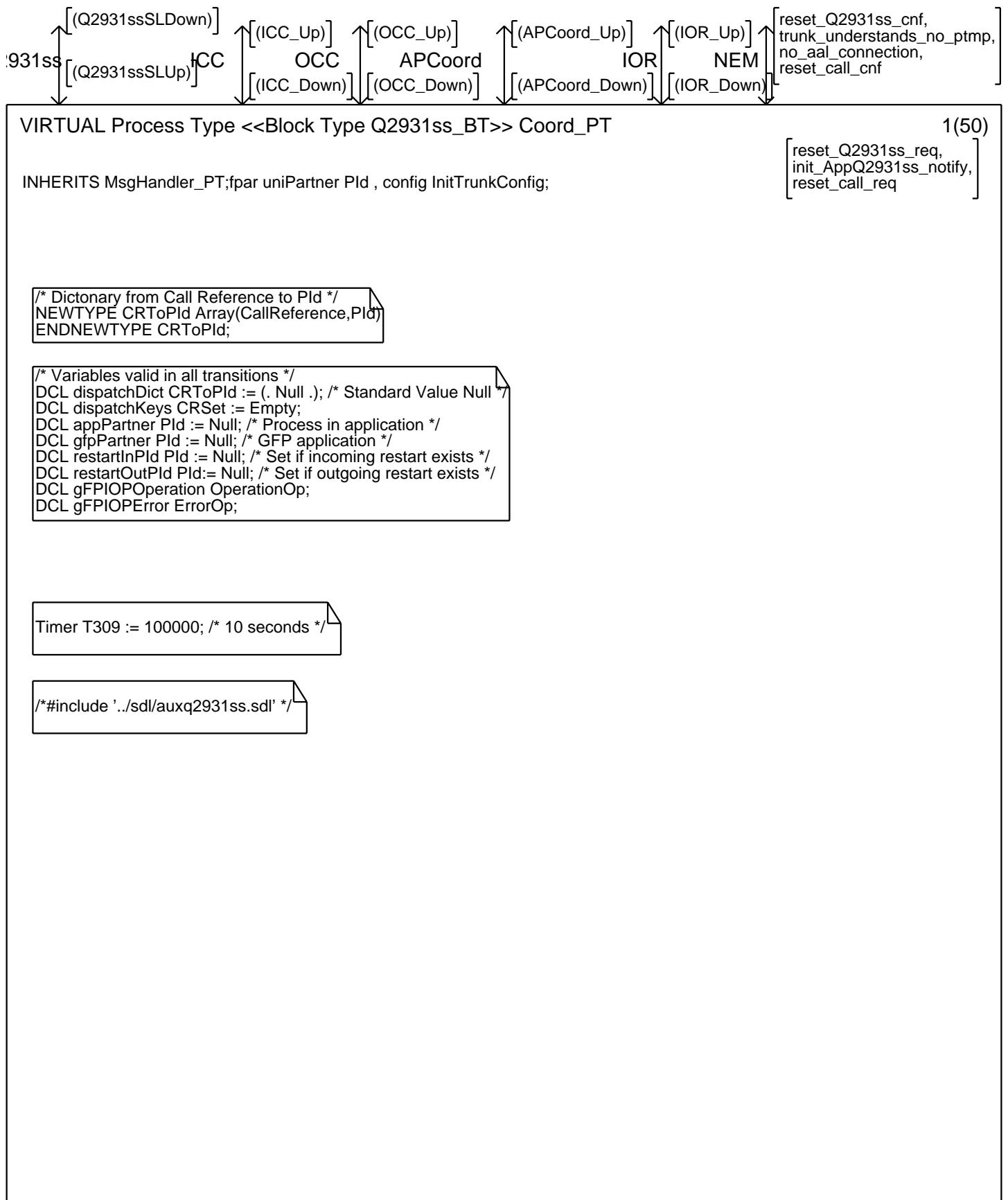
Block Type Q2931ss_BT

9(9)

VIRTUAL whatUsage

In what configuration is this block type used:
user/network, simulation/implementation, validation

Annex B: Coord_PT



Annex B: Coord_PT

VIRTUAL Process Type <<Block Type Q2931ss_BT>> Coord_PT

2(50)

INHERITS MsgHandler_PT;fpar uniPartner PId , config InitTrunkConfig;

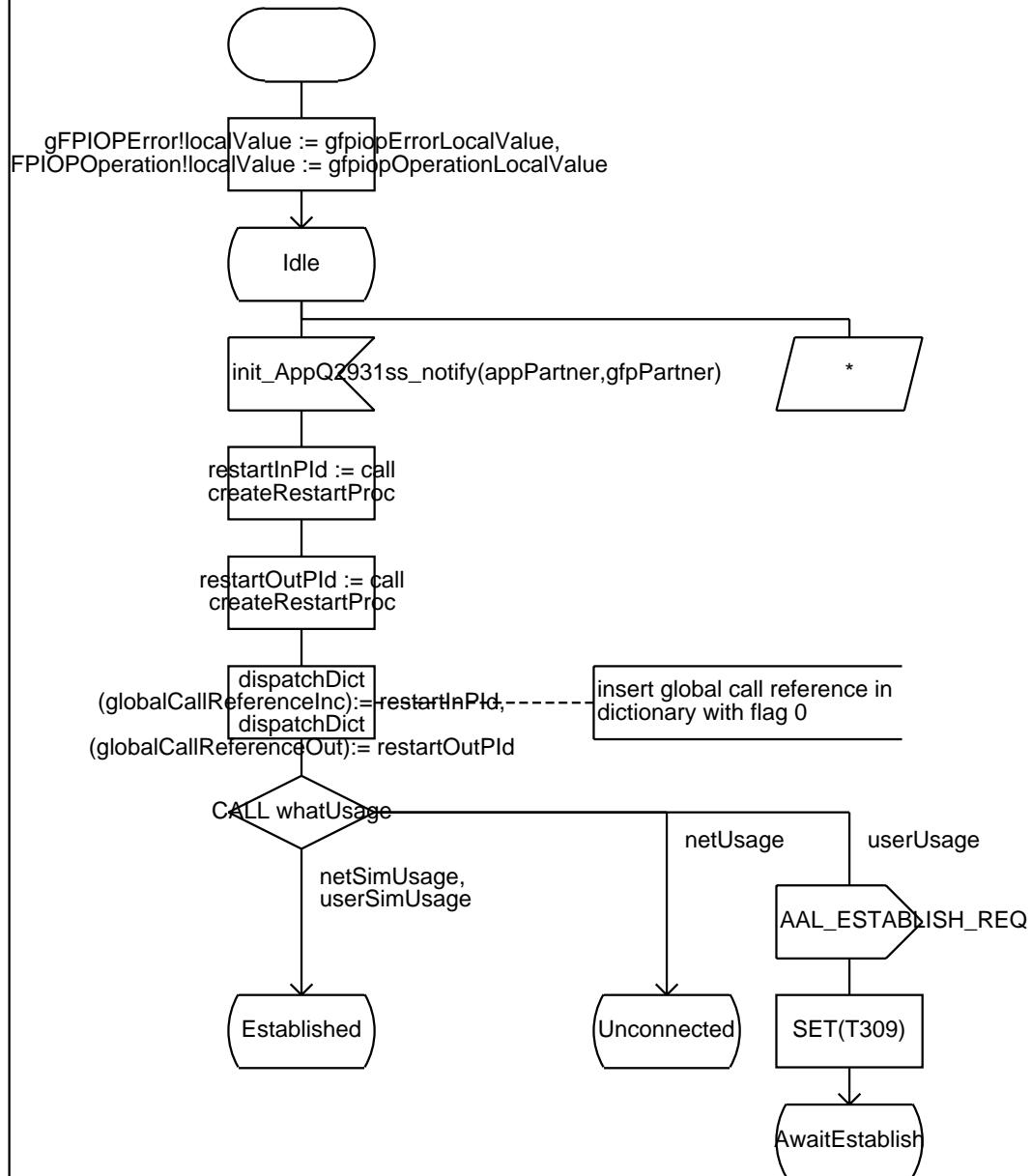
```
/* Variable valid only during one transition */
DCL cr CallReference;
DCL pid PId;
DCL atrunk TrunkNumber;
DCL data AALData;
DCL resetsender PId := Null;
```

Annex B: Coord_PT

VIRTUAL Process Type <<Block Type Q2931ss_BT>> Coord_PT

3(50)

INHERITS MsgHandler_PT;fpar uniPartner PId , config InitTrunkConfig;

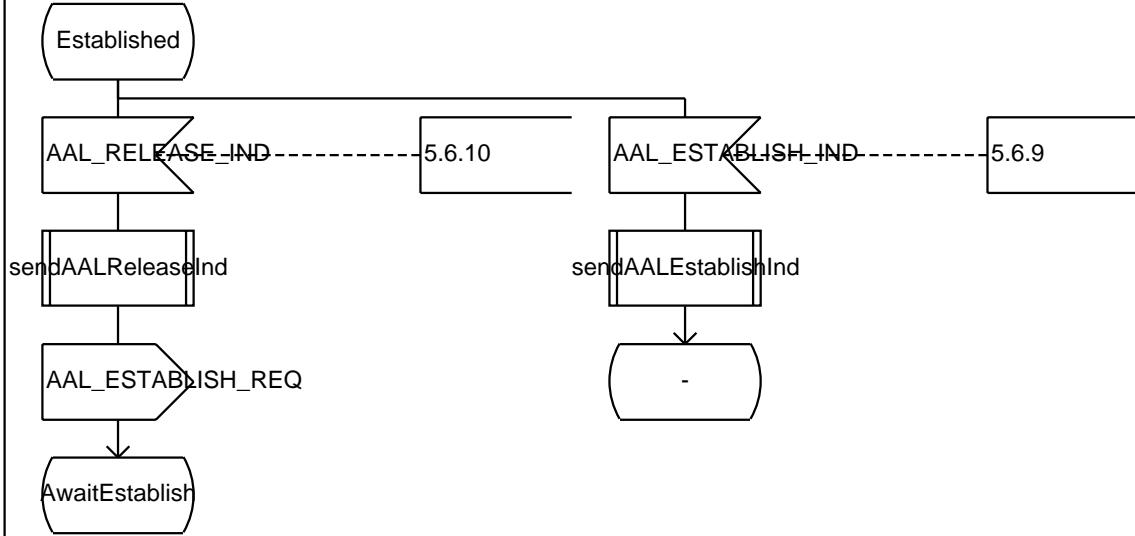


Annex B: Coord_PT

VIRTUAL Process Type <<Block Type Q2931ss_BT>> Coord_PT

4(50)

INHERITS MsgHandler_PT;fpar uniPartner PId , config InitTrunkConfig;

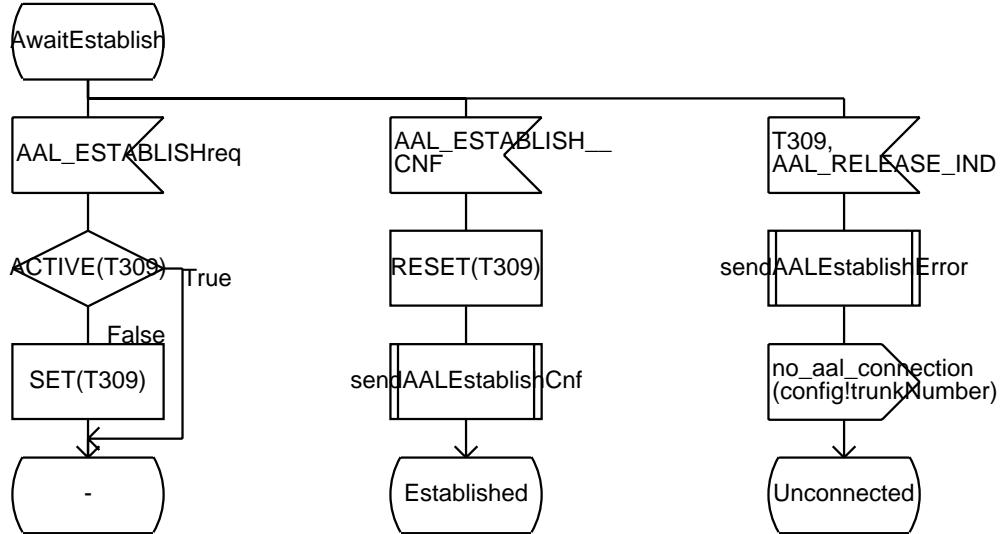


Annex B: Coord_PT

VIRTUAL Process Type <<Block Type Q2931ss_BT>> Coord_PT

5(50)

INHERITS MsgHandler_PT;fpar uniPartner PId , config InitTrunkConfig;

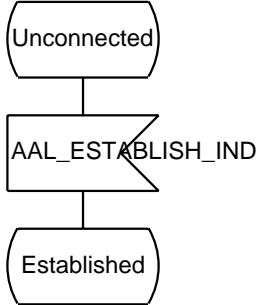


Annex B: Coord_PT

VIRTUAL Process Type <<Block Type Q2931ss_BT>> Coord_PT

6(50)

INHERITS MsgHandler_PT;fpar uniPartner PId , config InitTrunkConfig;

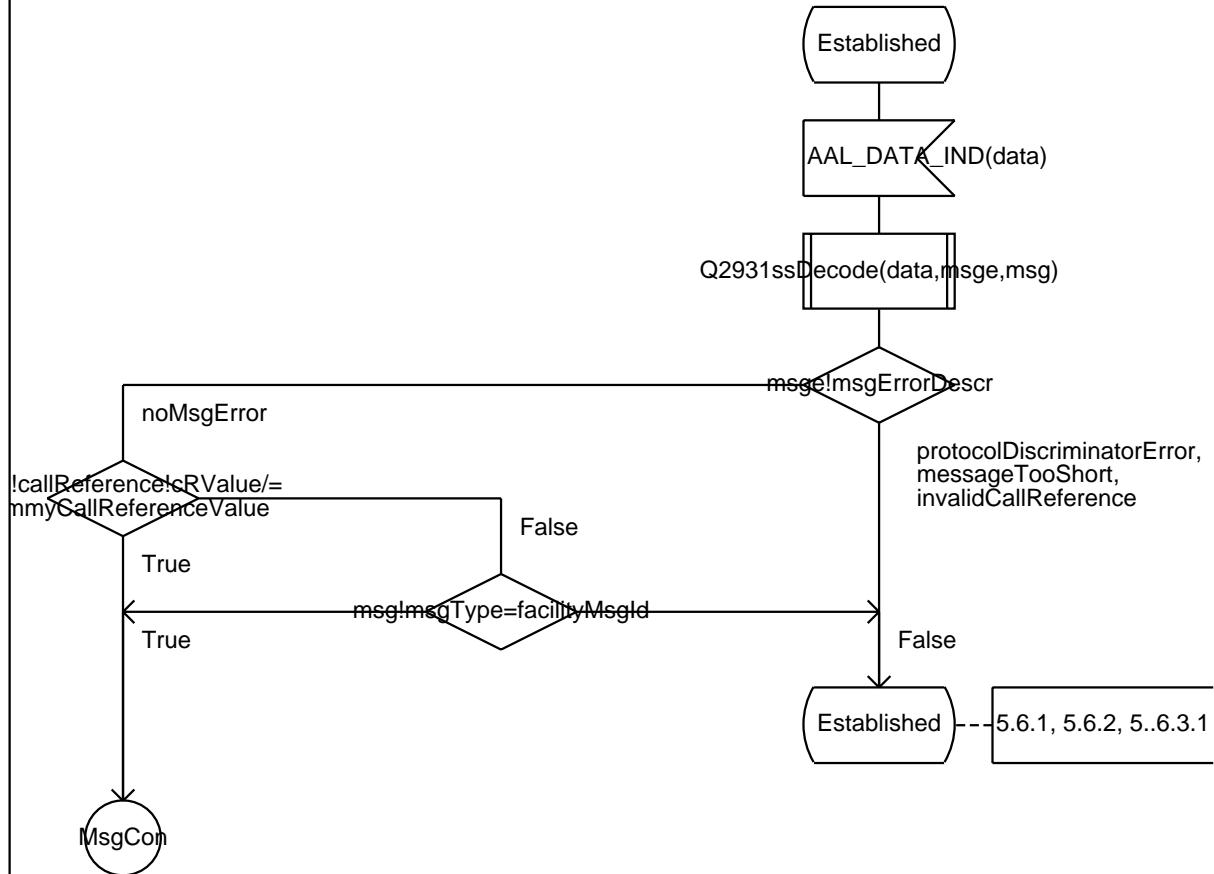


Annex B: Coord_PT

VIRTUAL Process Type <<Block Type Q2931ss_BT>> Coord_PT

7(50)

INHERITS MsgHandler_PT;fpar uniPartner PId , config InitTrunkConfig;



Annex B: Coord_PT

VIRTUAL Process Type <<Block Type Q2931ss_BT>> Coord_PT

8(50)

INHERITS MsgHandler_PT; fpar uniPartner PId , config InitTrunkConfig;

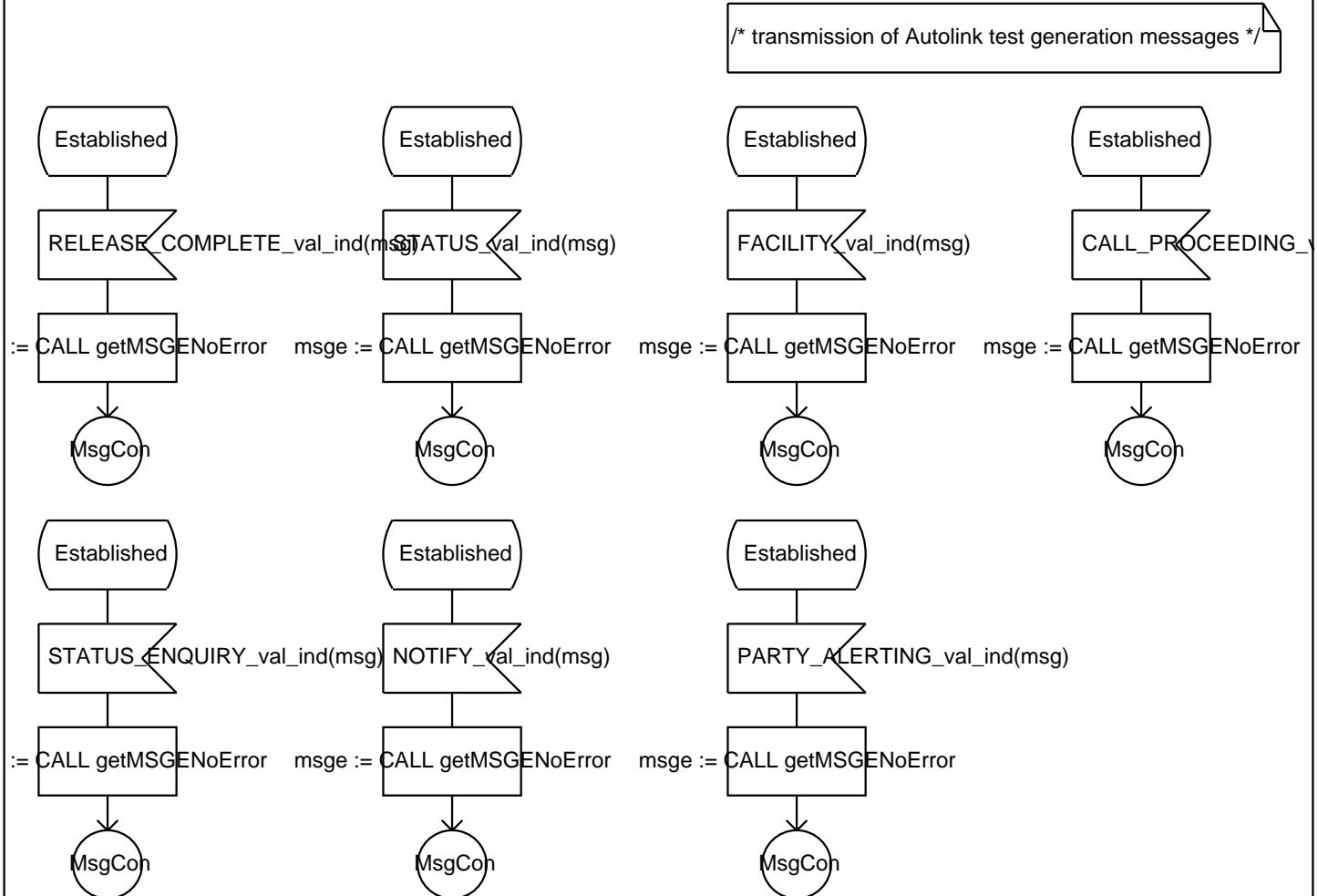
IMPORTED PROCEDURE whatRestartState;
RETURNS CallState;

Annex B: Coord_PT

VIRTUAL Process Type <<Block Type Q2931ss_BT>> Coord_PT

9(50)

INHERITS MsgHandler_PT;fpar uniPartner Pld , config InitTrunkConfig;

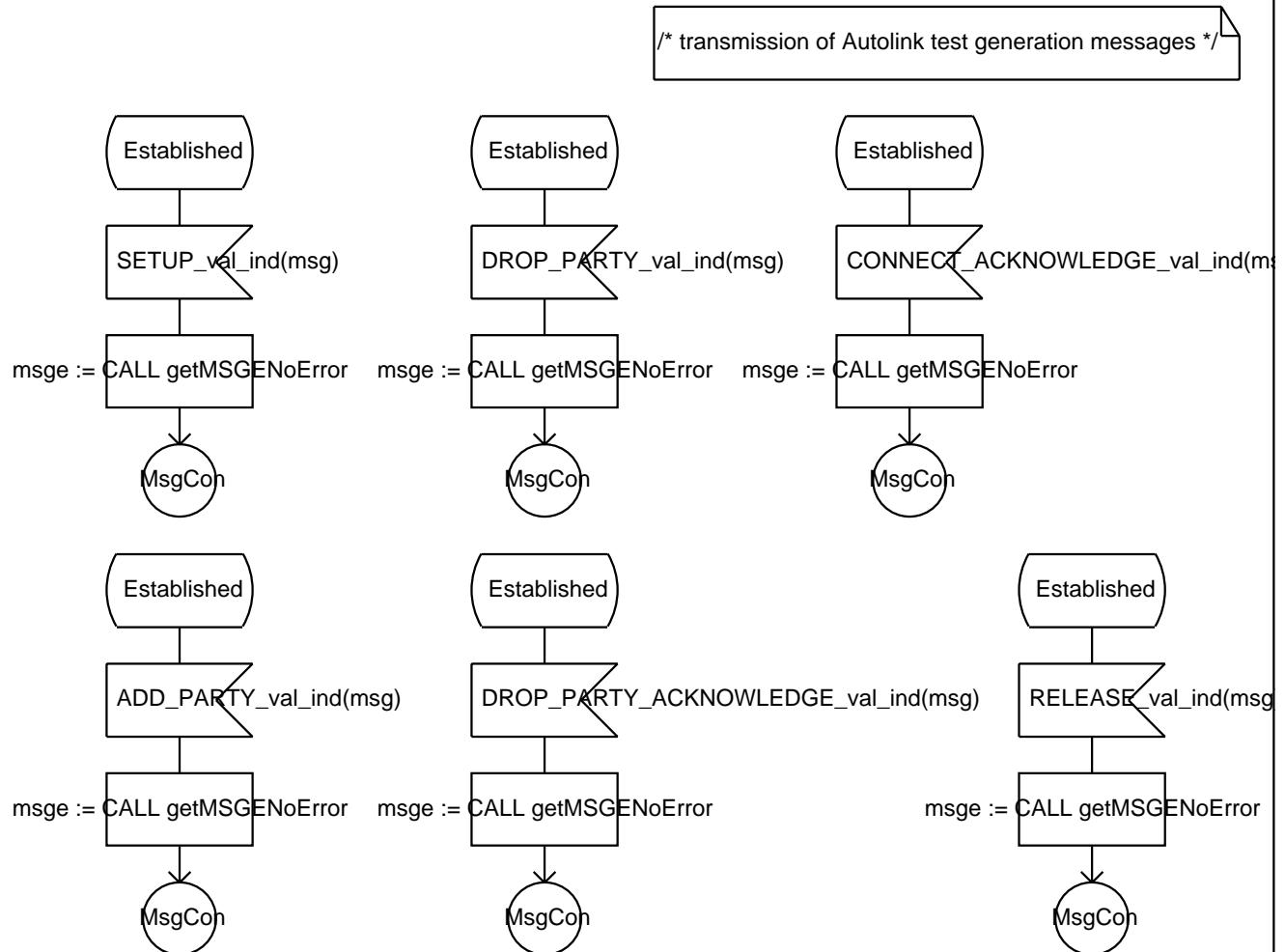


Annex B: Coord_PT

VIRTUAL Process Type <<Block Type Q2931ss_BT>> Coord_PT

10(50)

INHERITS MsgHandler_PT;fpar uniPartner PId , config InitTrunkConfig;



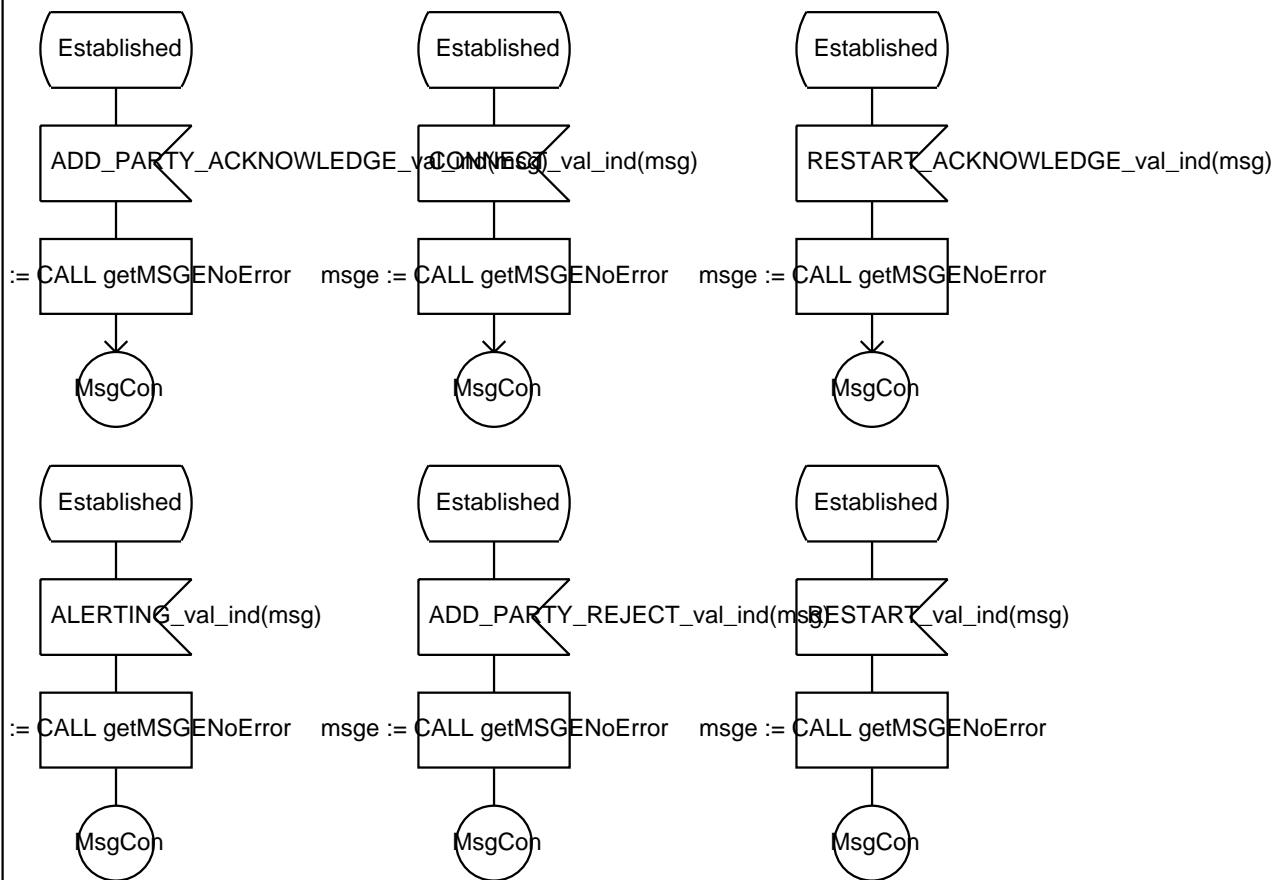
Annex B: Coord_PT

VIRTUAL Process Type <<Block Type Q2931ss_BT>> Coord_PT

11(50)

INHERITS MsgHandler_PT;fpar uniPartner PId , config InitTrunkConfig;

/* transmission of Autolink test generation messages */

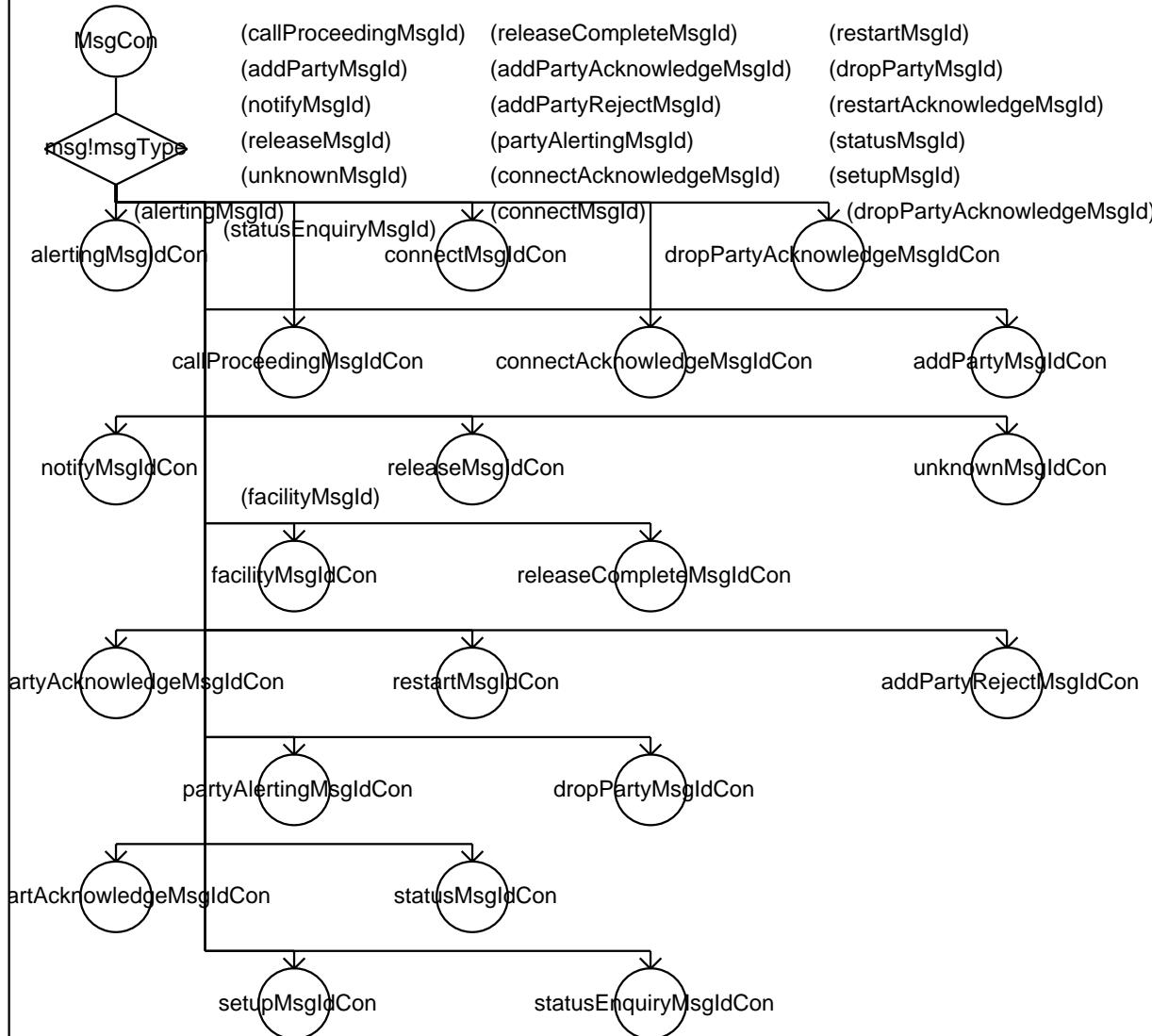


Annex B: Coord_PT

VIRTUAL Process Type <<Block Type Q2931ss_BT>> Coord_PT

12(50)

INHERITS MsgHandler_PT;fpar uniPartner PId , config InitTrunkConfig;

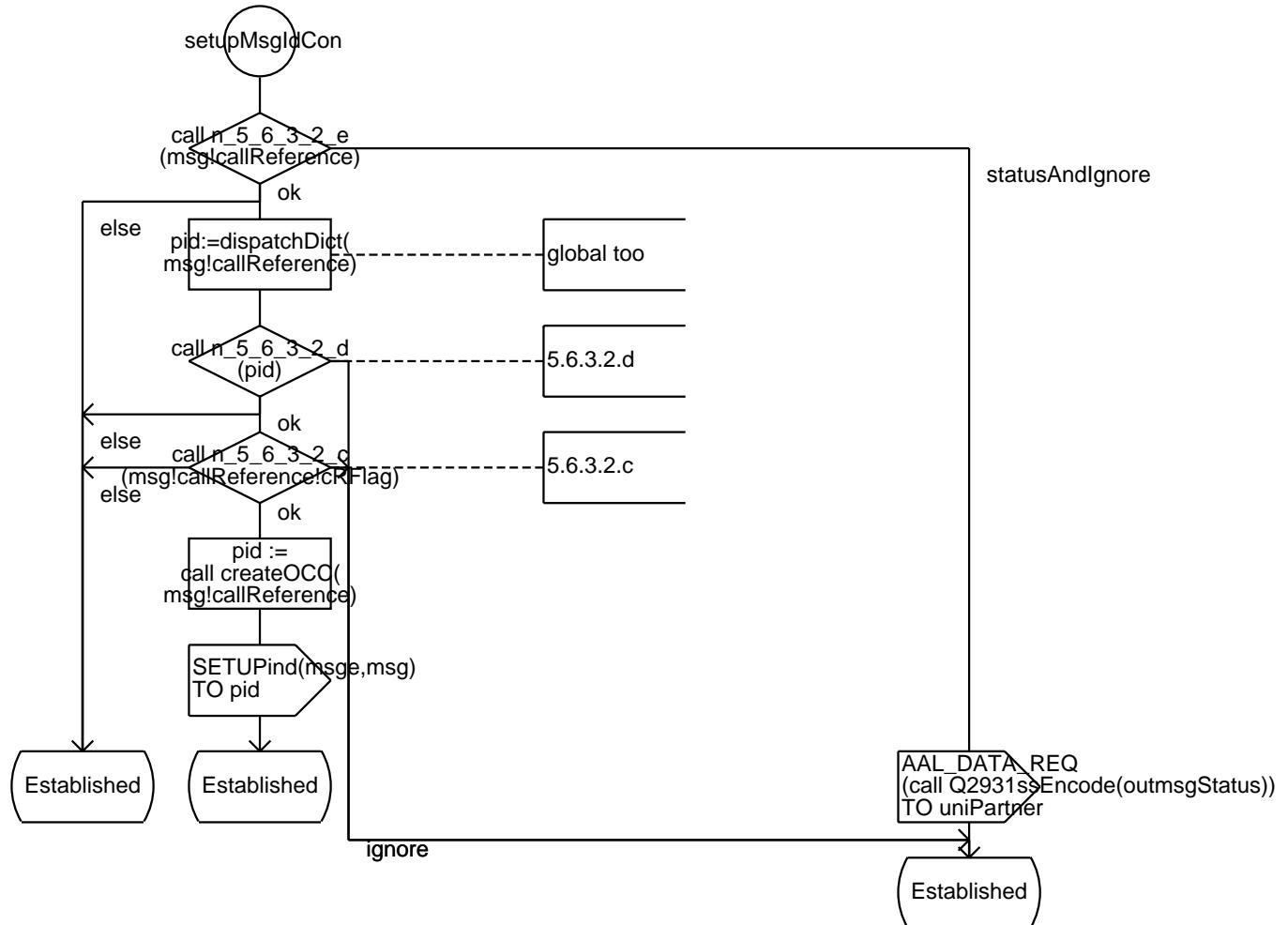


Annex B: Coord_PT

VIRTUAL Process Type <<Block Type Q2931ss_BT>> Coord_PT

13(50)

INHERITS MsgHandler_PT;fpar uniPartner PId , config InitTrunkConfig;

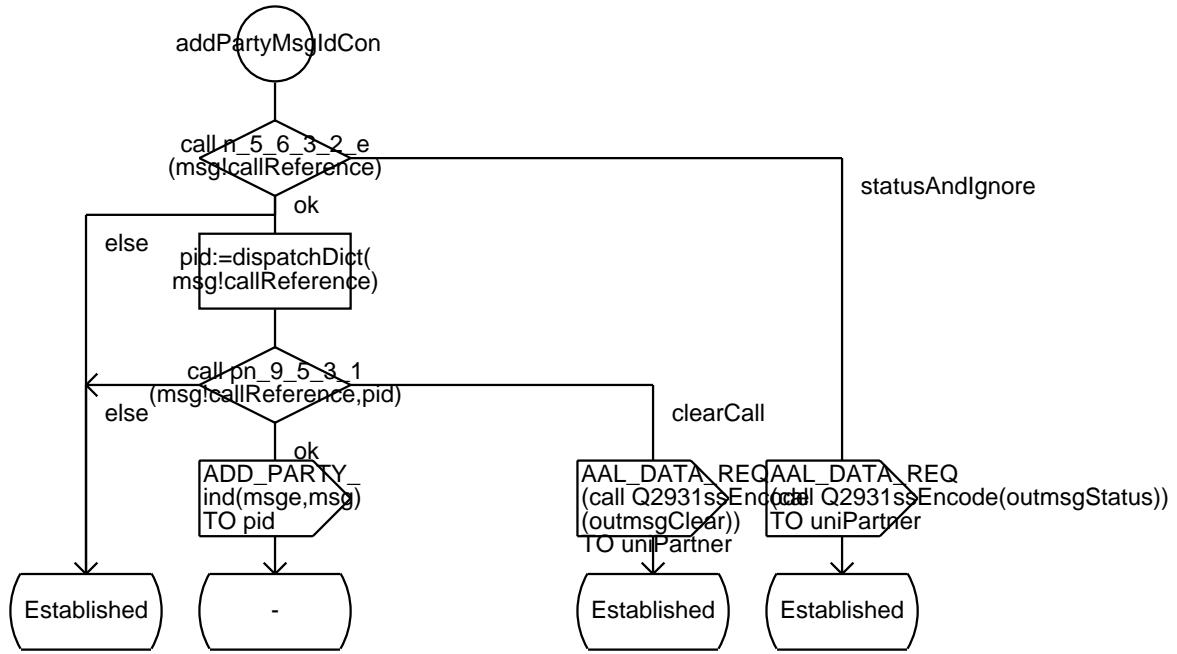


Annex B: Coord_PT

VIRTUAL Process Type <<Block Type Q2931ss_BT>> Coord_PT

14(50)

INHERITS MsgHandler_PT;fpar uniPartner PId , config InitTrunkConfig;

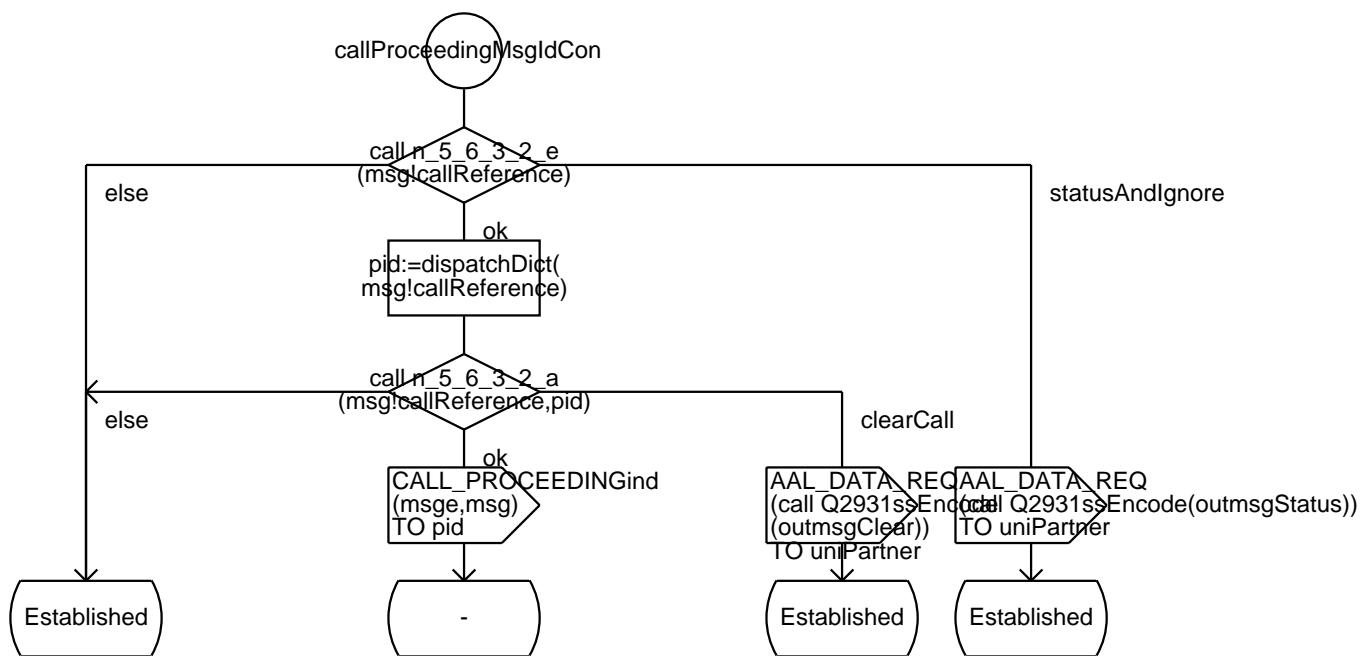


Annex B: Coord_PT

VIRTUAL Process Type <<Block Type Q2931ss_BT>> Coord_PT

15(50)

INHERITS MsgHandler_PT;fpar uniPartner PId , config InitTrunkConfig;

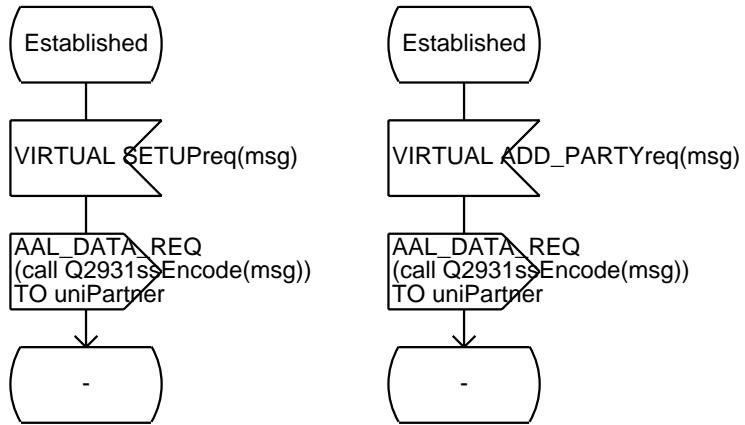


Annex B: Coord_PT

VIRTUAL Process Type <<Block Type Q2931ss_BT>> Coord_PT

16(50)

INHERITS MsgHandler_PT;fpar uniPartner PId , config InitTrunkConfig;

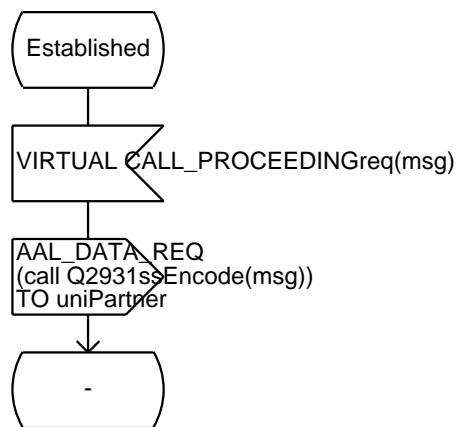


Annex B: Coord_PT

VIRTUAL Process Type <<Block Type Q2931ss_BT>> Coord_PT

17(50)

INHERITS MsgHandler_PT;fpar uniPartner PId , config InitTrunkConfig;

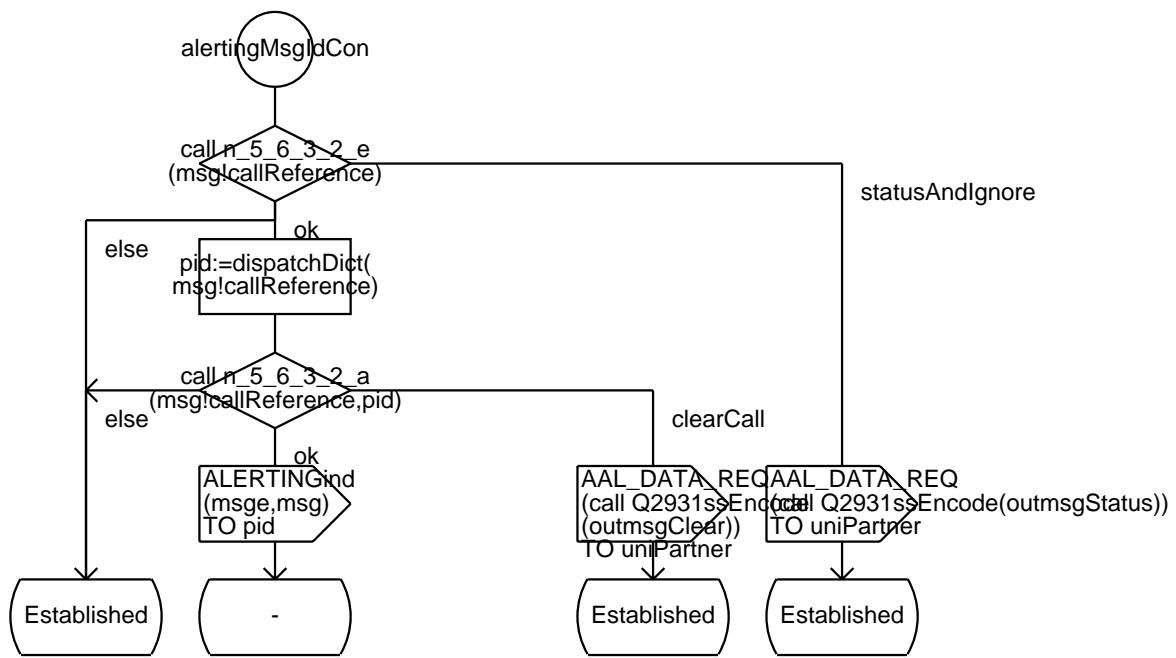


Annex B: Coord_PT

VIRTUAL Process Type <<Block Type Q2931ss_BT>> Coord_PT

18(50)

INHERITS MsgHandler_PT;fpar uniPartner PId , config InitTrunkConfig;

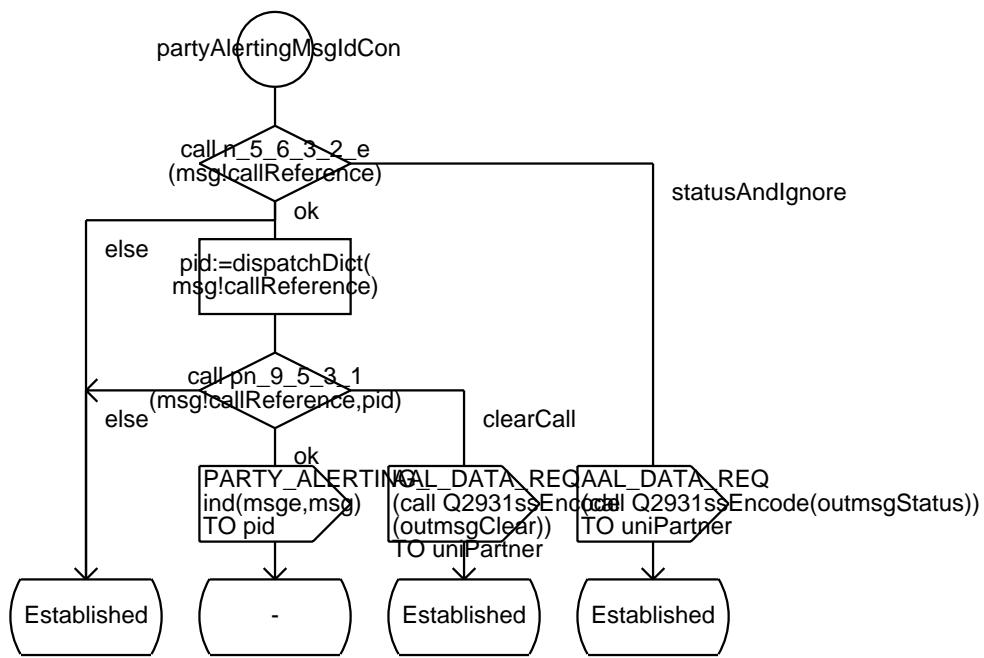


Annex B: Coord_PT

VIRTUAL Process Type <<Block Type Q2931ss_BT>> Coord_PT

19(50)

INHERITS MsgHandler_PT;fpar uniPartner PId , config InitTrunkConfig;

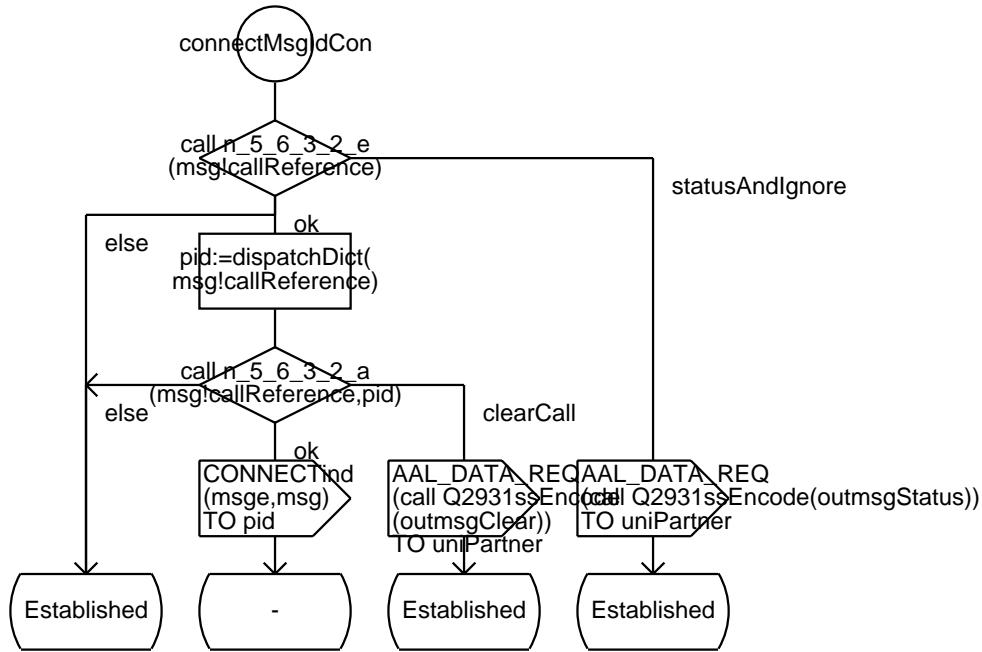


Annex B: Coord_PT

VIRTUAL Process Type <<Block Type Q2931ss_BT>> Coord_PT

20(50)

INHERITS MsgHandler_PT;fpar uniPartner PId , config InitTrunkConfig;

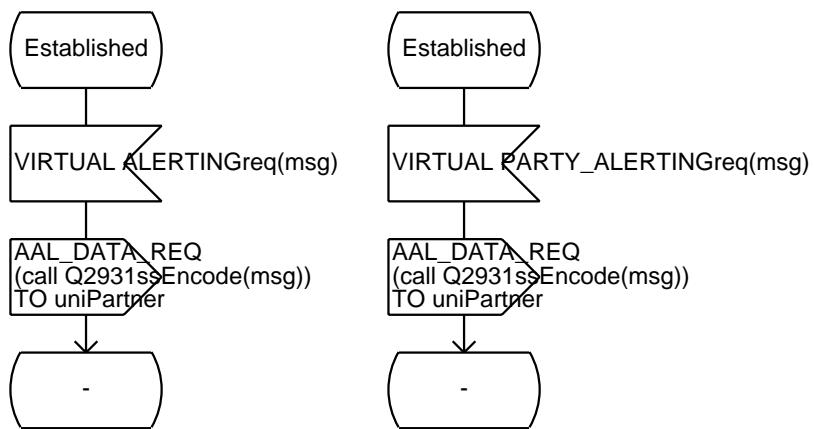


Annex B: Coord_PT

VIRTUAL Process Type <<Block Type Q2931ss_BT>> Coord_PT

21(50)

INHERITS MsgHandler_PT;fpar uniPartner PId , config InitTrunkConfig;

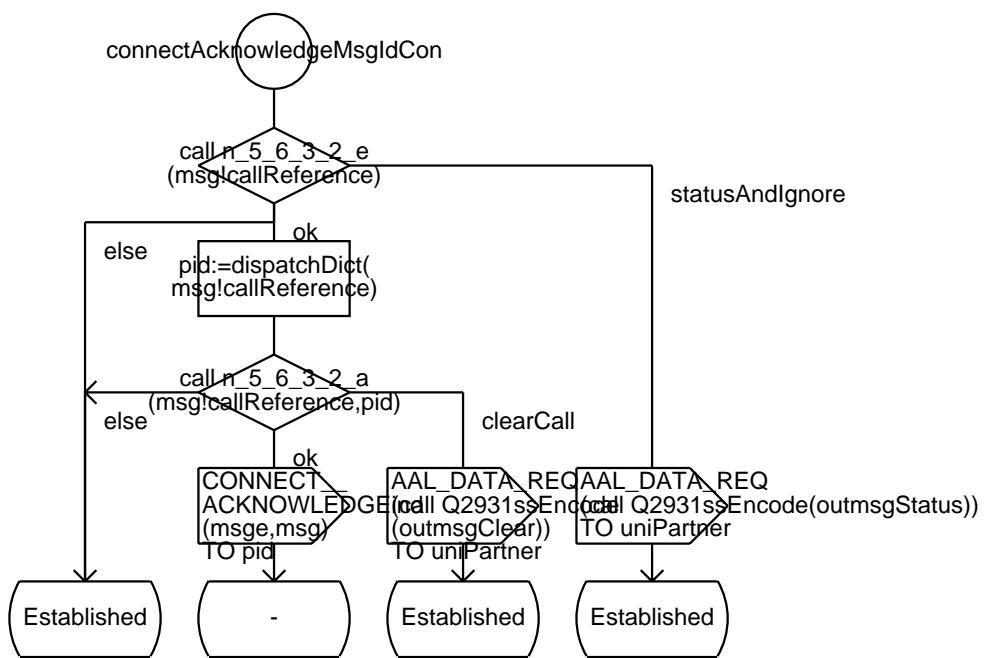


Annex B: Coord_PT

VIRTUAL Process Type <<Block Type Q2931ss_BT>> Coord_PT

22(50)

INHERITS MsgHandler_PT;fpar uniPartner PId , config InitTrunkConfig;

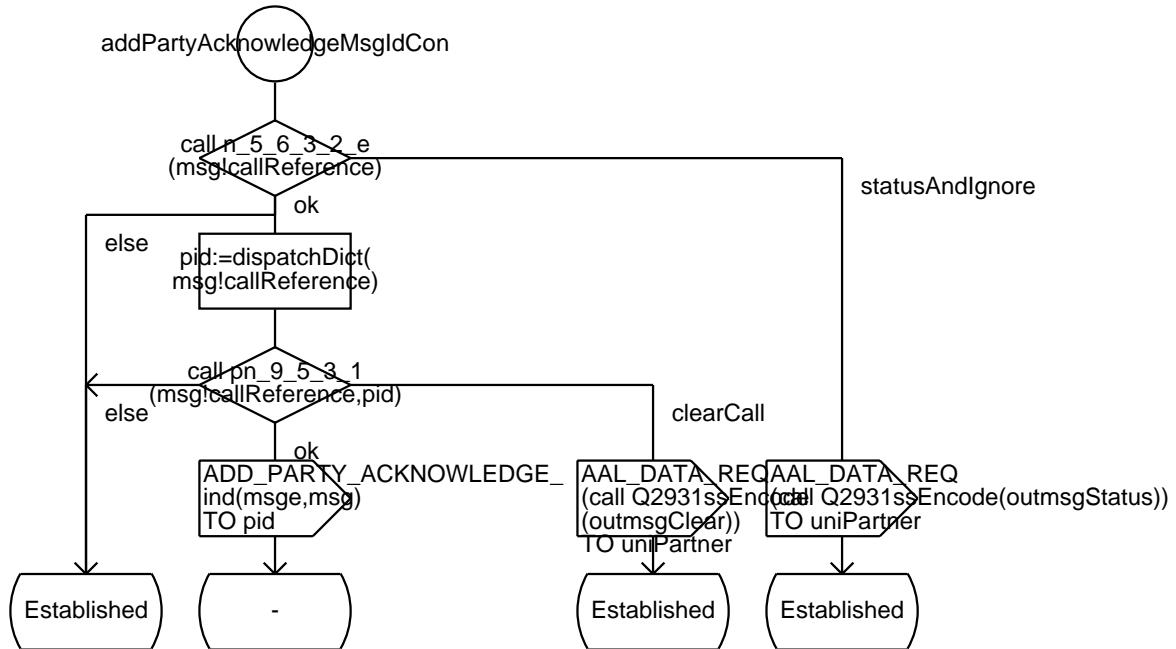


Annex B: Coord_PT

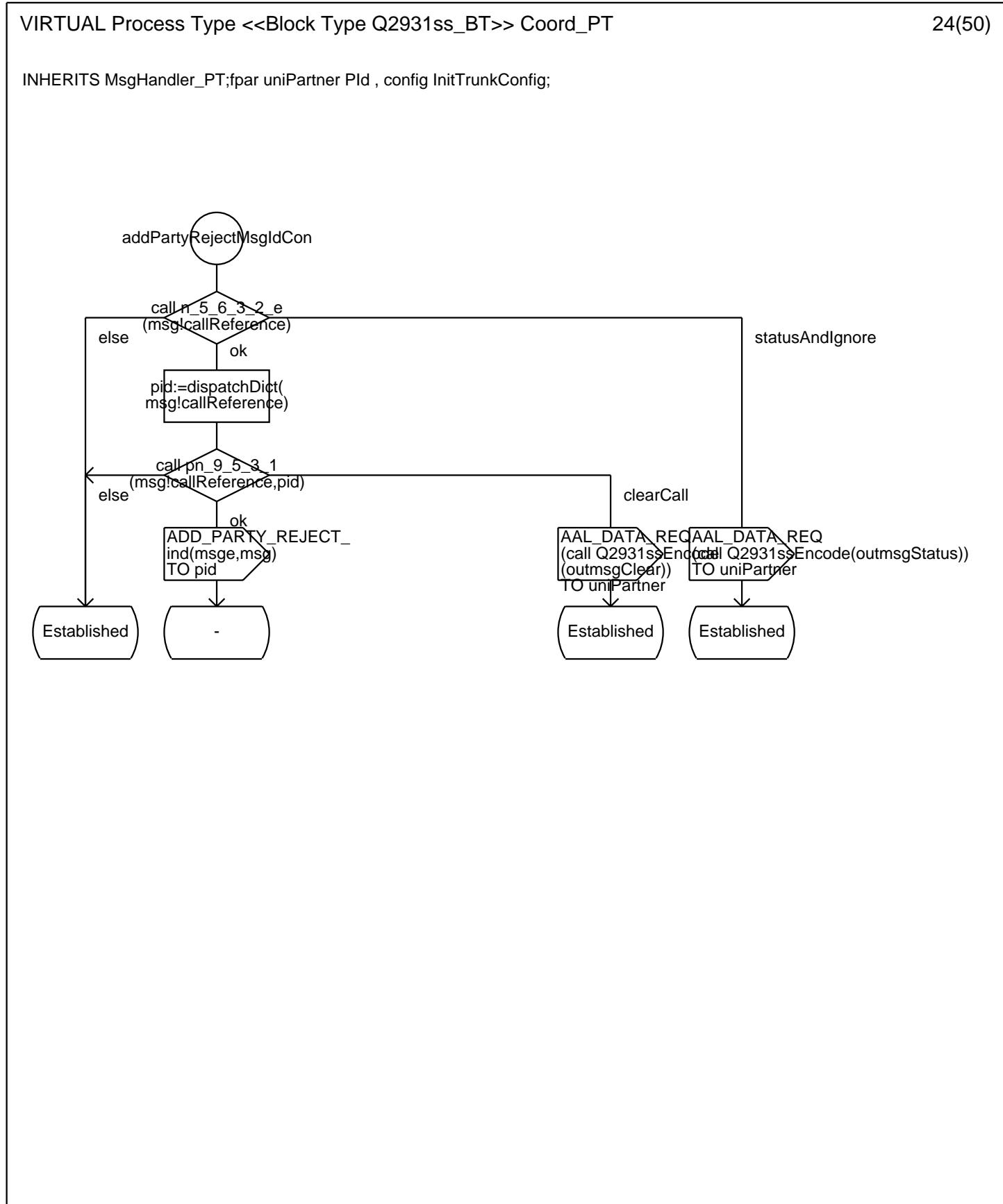
VIRTUAL Process Type <<Block Type Q2931ss_BT>> Coord_PT

23(50)

INHERITS MsgHandler_PT;fpar uniPartner PId , config InitTrunkConfig;



Annex B: Coord_PT

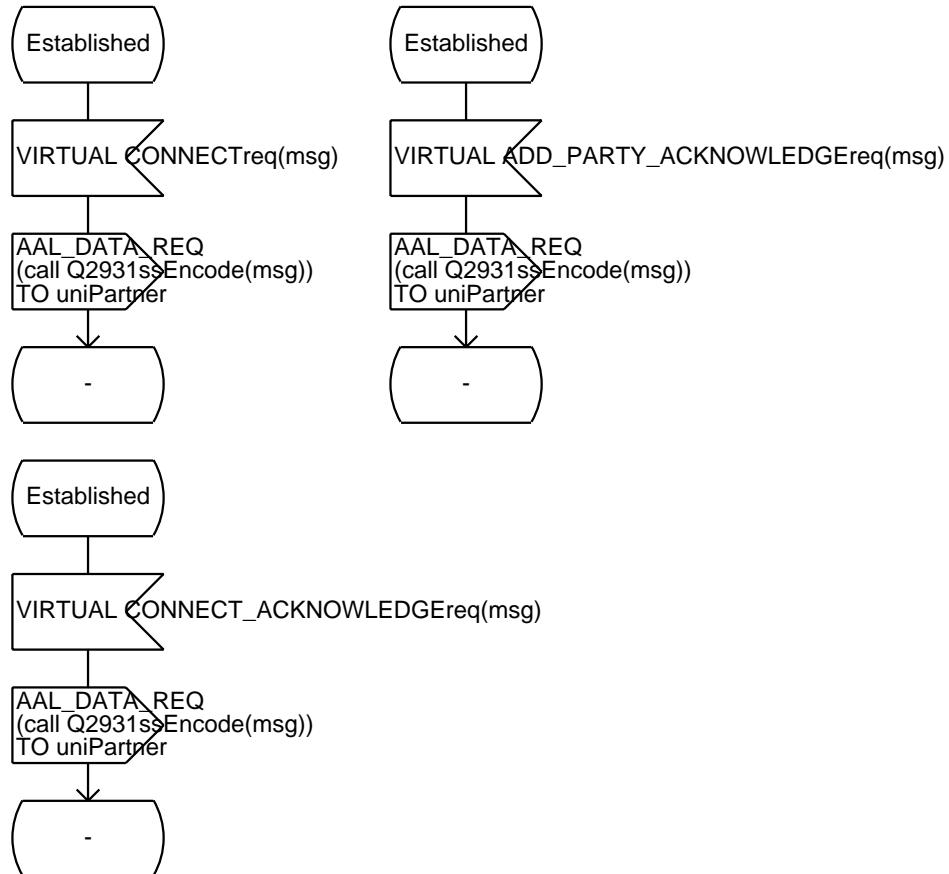


Annex B: Coord_PT

VIRTUAL Process Type <<Block Type Q2931ss_BT>> Coord_PT

25(50)

INHERITS MsgHandler_PT;fpar uniPartner PId , config InitTrunkConfig;

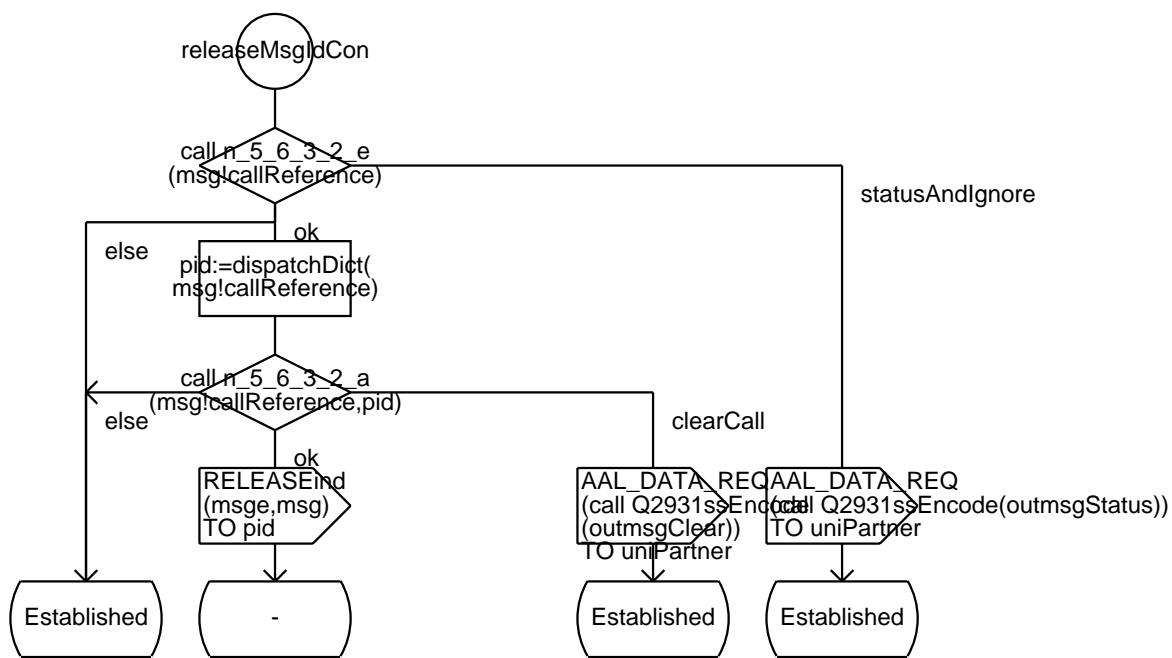


Annex B: Coord_PT

VIRTUAL Process Type <<Block Type Q2931ss_BT>> Coord_PT

26(50)

INHERITS MsgHandler_PT;fpar uniPartner PId , config InitTrunkConfig;

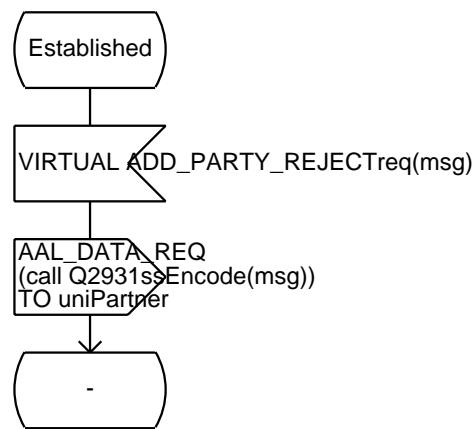


Annex B: Coord_PT

VIRTUAL Process Type <<Block Type Q2931ss_BT>> Coord_PT

27(50)

INHERITS MsgHandler_PT;fpar uniPartner PId , config InitTrunkConfig;

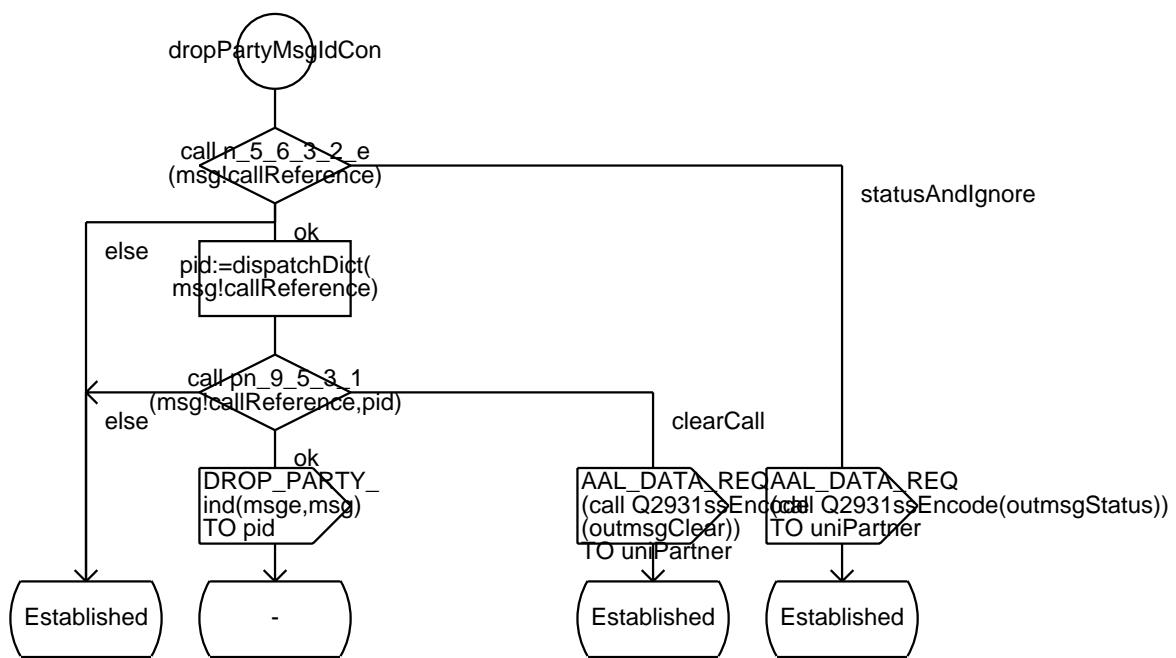


Annex B: Coord_PT

VIRTUAL Process Type <<Block Type Q2931ss_BT>> Coord_PT

28(50)

INHERITS MsgHandler_PT;fpar uniPartner PId , config InitTrunkConfig;

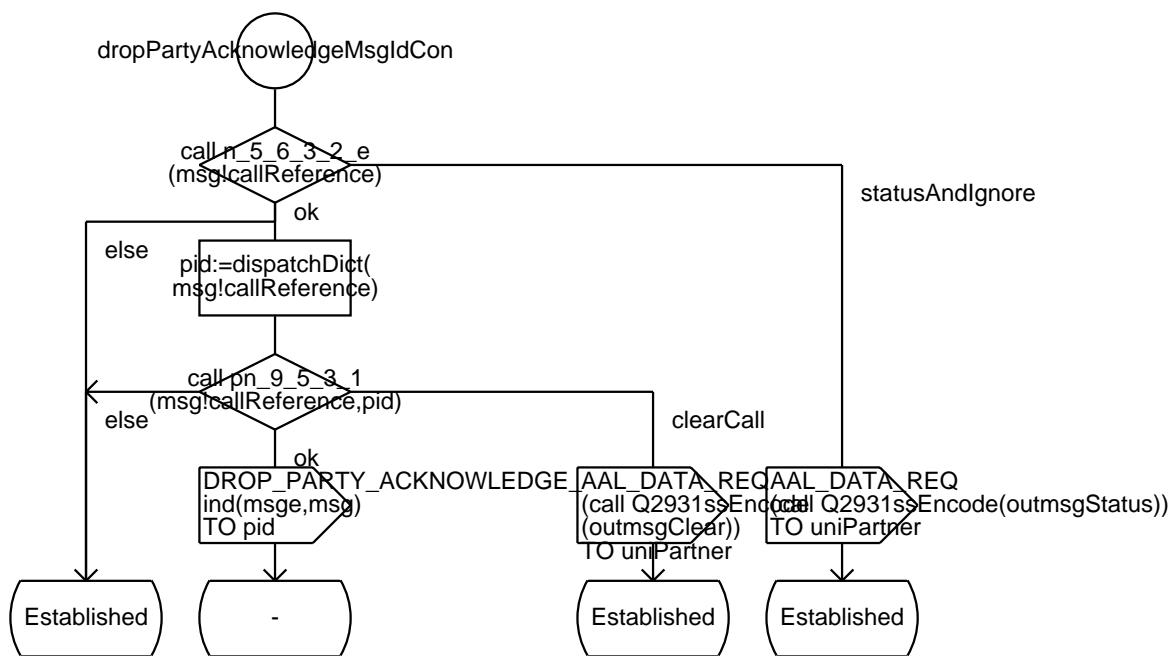


Annex B: Coord_PT

VIRTUAL Process Type <<Block Type Q2931ss_BT>> Coord_PT

29(50)

INHERITS MsgHandler_PT;fpar uniPartner PId , config InitTrunkConfig;

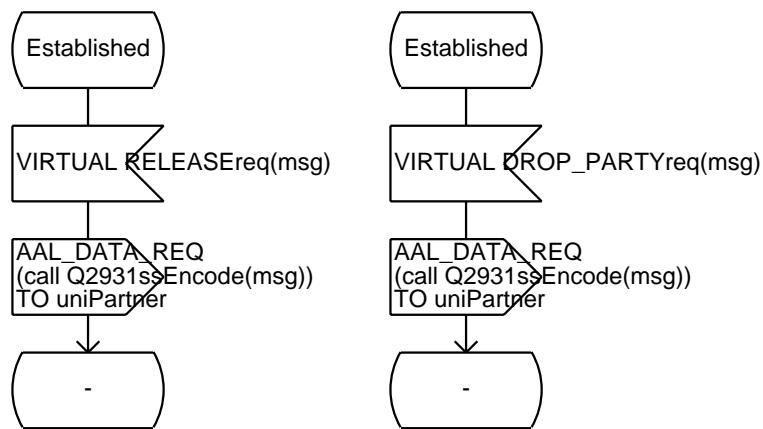


Annex B: Coord_PT

VIRTUAL Process Type <<Block Type Q2931ss_BT>> Coord_PT

30(50)

INHERITS MsgHandler_PT;fpar uniPartner PId , config InitTrunkConfig;

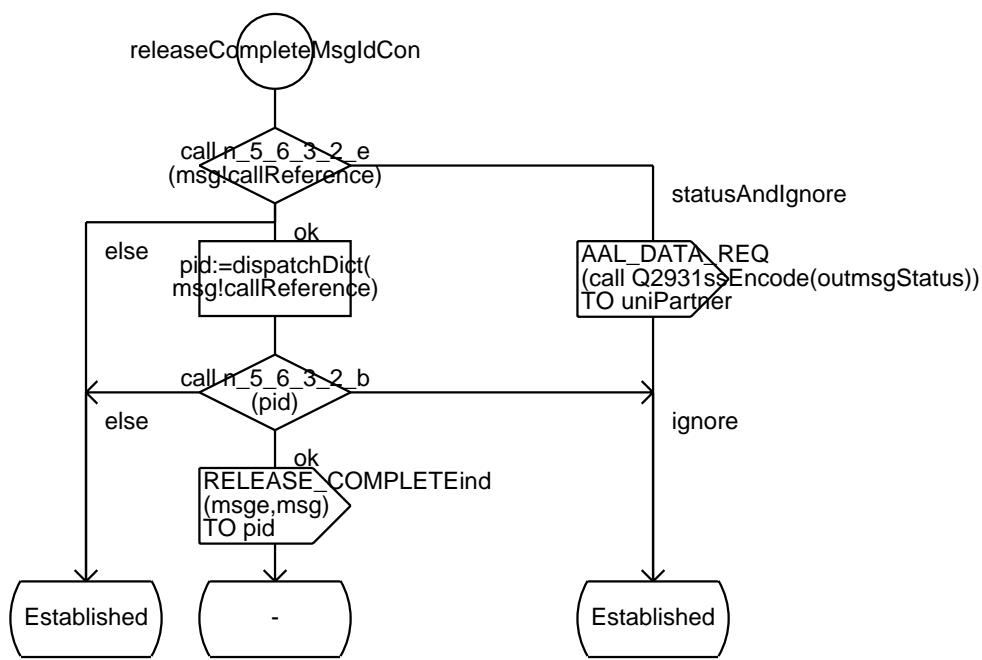


Annex B: Coord_PT

VIRTUAL Process Type <<Block Type Q2931ss_BT>> Coord_PT

31(50)

INHERITS MsgHandler_PT;fpar uniPartner PId , config InitTrunkConfig;

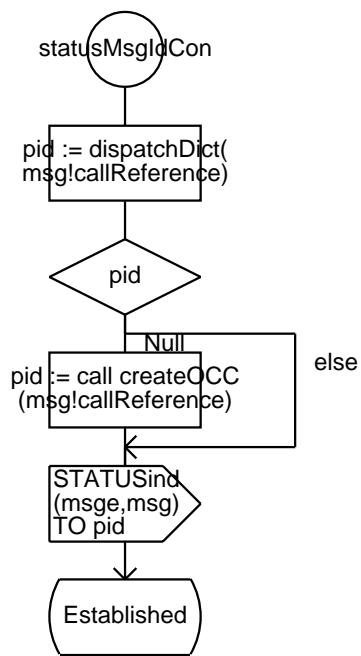


Annex B: Coord_PT

VIRTUAL Process Type <<Block Type Q2931ss_BT>> Coord_PT

32(50)

INHERITS MsgHandler_PT;fpar uniPartner PId , config InitTrunkConfig;

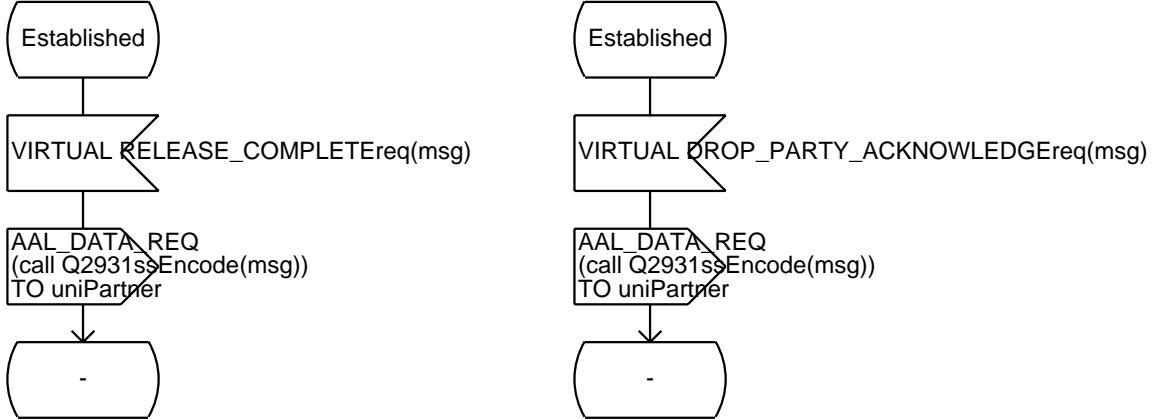


Annex B: Coord_PT

VIRTUAL Process Type <<Block Type Q2931ss_BT>> Coord_PT

33(50)

INHERITS MsgHandler_PT;fpar uniPartner PId , config InitTrunkConfig;

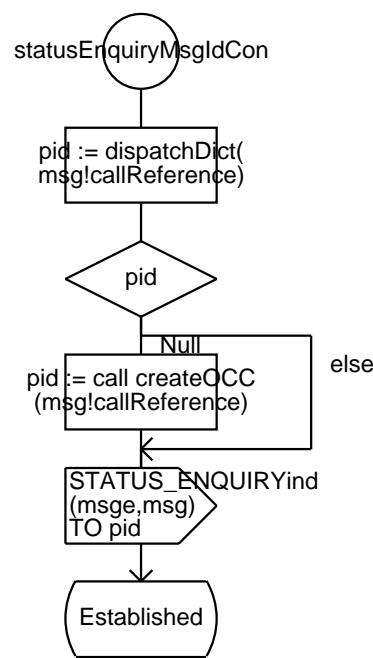


Annex B: Coord_PT

VIRTUAL Process Type <<Block Type Q2931ss_BT>> Coord_PT

34(50)

INHERITS MsgHandler_PT;fpar uniPartner PId , config InitTrunkConfig;

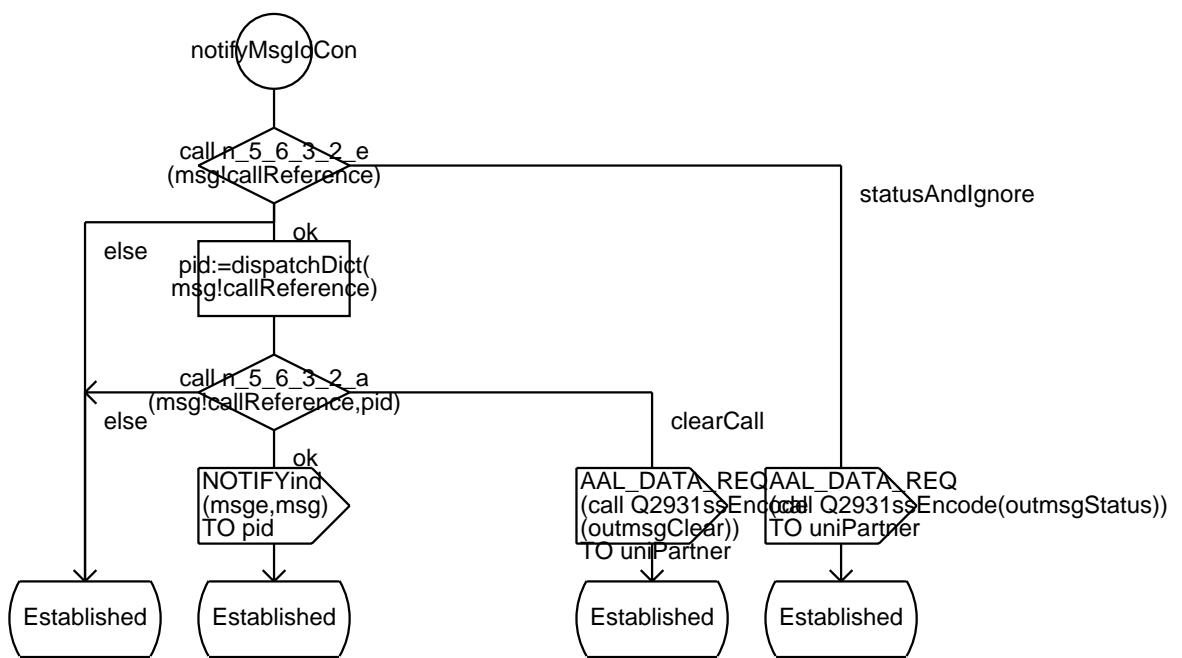


Annex B: Coord_PT

VIRTUAL Process Type <<Block Type Q2931ss_BT>> Coord_PT

35(50)

INHERITS MsgHandler_PT;fpar uniPartner PId , config InitTrunkConfig;

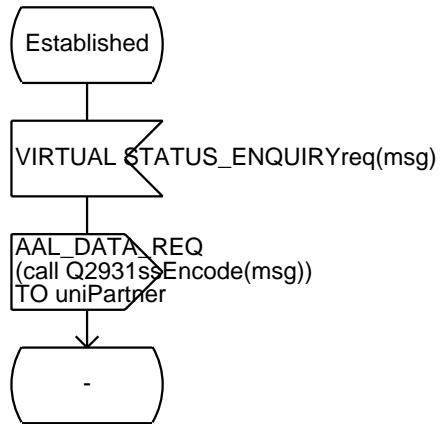
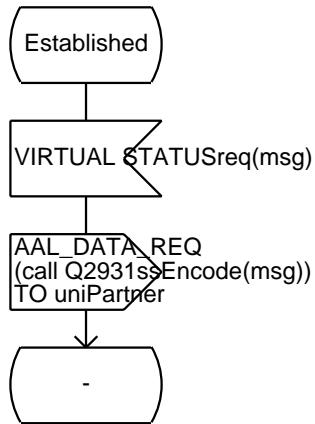


Annex B: Coord_PT

VIRTUAL Process Type <<Block Type Q2931ss_BT>> Coord_PT

36(50)

INHERITS MsgHandler_PT;fpar uniPartner PId , config InitTrunkConfig;

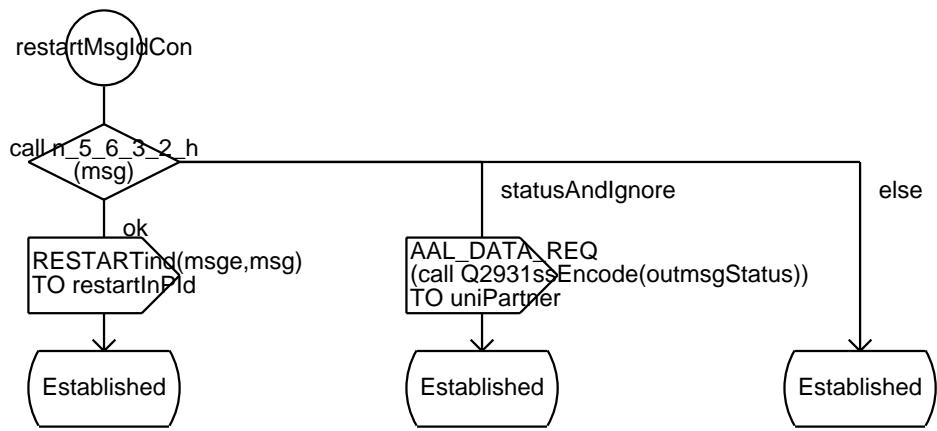


Annex B: Coord_PT

VIRTUAL Process Type <<Block Type Q2931ss_BT>> Coord_PT

37(50)

INHERITS MsgHandler_PT;fpar uniPartner PId , config InitTrunkConfig;

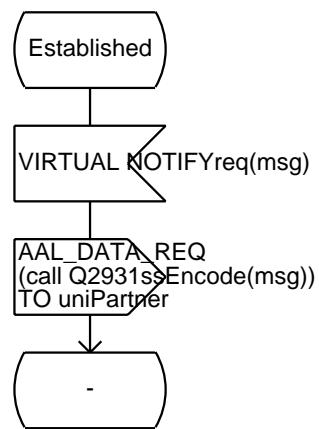


Annex B: Coord_PT

VIRTUAL Process Type <<Block Type Q2931ss_BT>> Coord_PT

38(50)

INHERITS MsgHandler_PT;fpar uniPartner PId , config InitTrunkConfig;

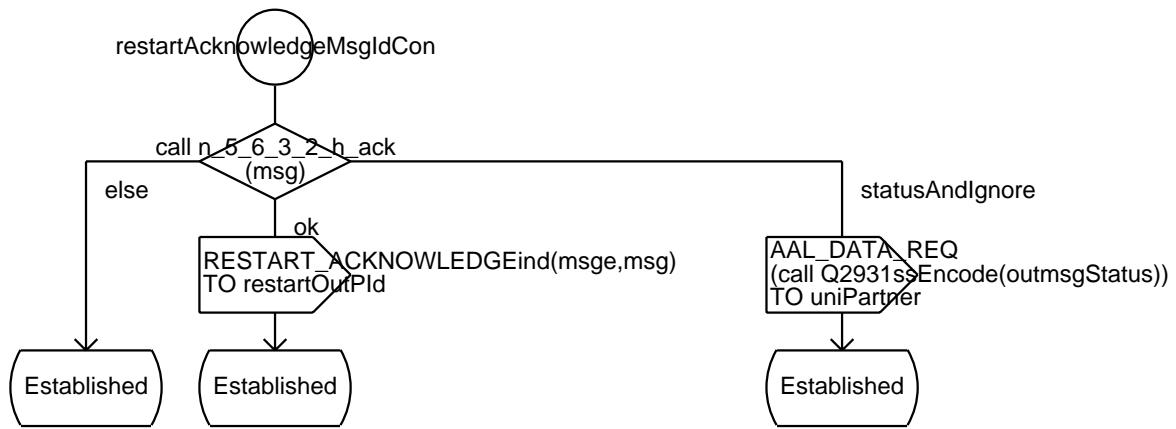


Annex B: Coord_PT

VIRTUAL Process Type <<Block Type Q2931ss_BT>> Coord_PT

39(50)

INHERITS MsgHandler_PT; fpar uniPartner PId , config InitTrunkConfig;

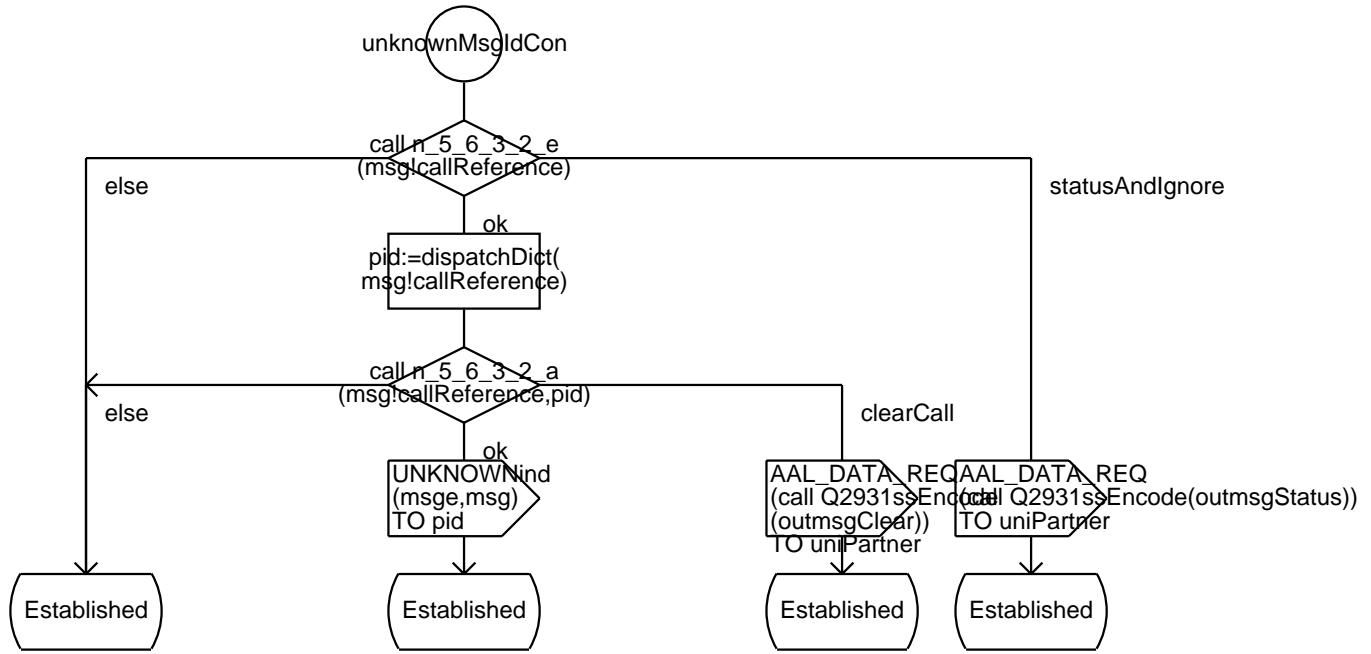


Annex B: Coord_PT

VIRTUAL Process Type <<Block Type Q2931ss_BT>> Coord_PT

40(50)

INHERITS MsgHandler_PT;fpar uniPartner PId , config InitTrunkConfig;

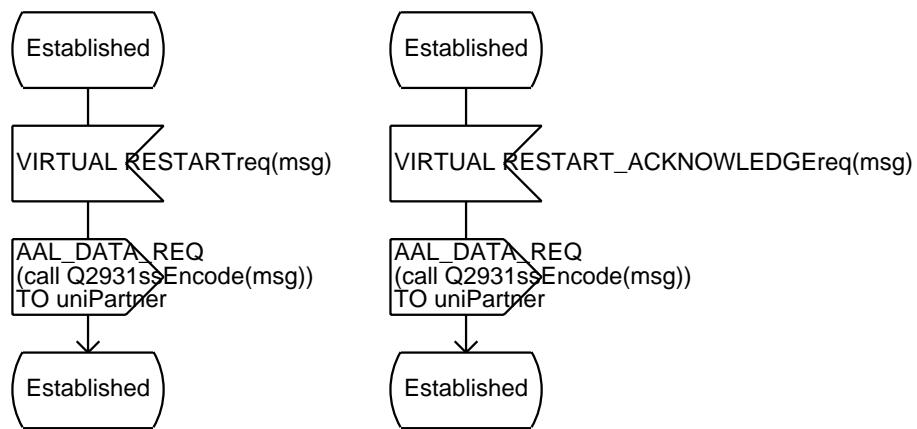


Annex B: Coord_PT

VIRTUAL Process Type <<Block Type Q2931ss_BT>> Coord_PT

41(50)

INHERITS MsgHandler_PT;fpar uniPartner PId , config InitTrunkConfig;

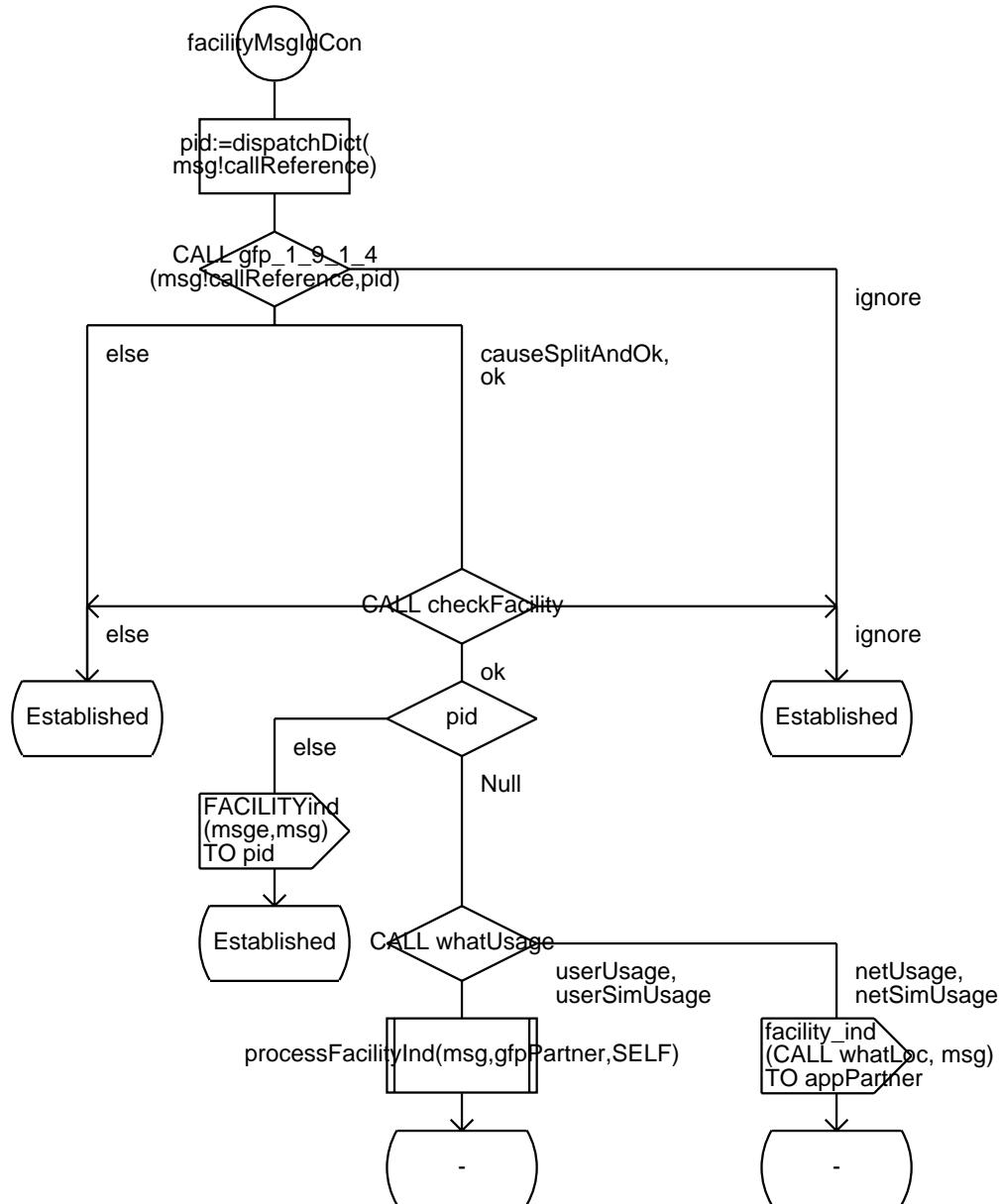


Annex B: Coord_PT

VIRTUAL Process Type <<Block Type Q2931ss_BT>> Coord_PT

42(50)

INHERITS MsgHandler_PT;fpar uniPartner PId , config InitTrunkConfig;

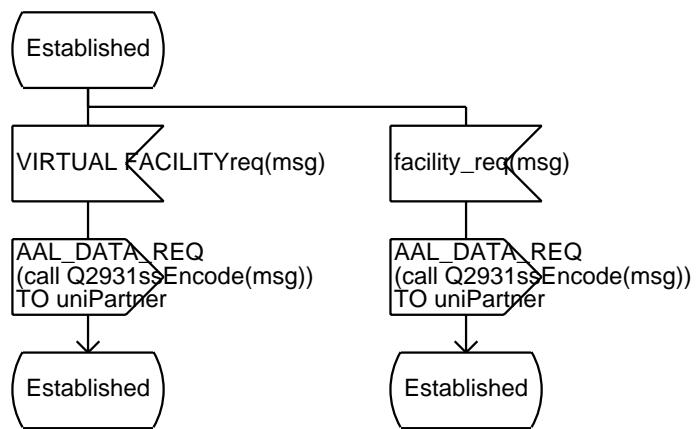


Annex B: Coord_PT

VIRTUAL Process Type <<Block Type Q2931ss_BT>> Coord_PT

43(50)

INHERITS MsgHandler_PT;fpar uniPartner PId , config InitTrunkConfig;

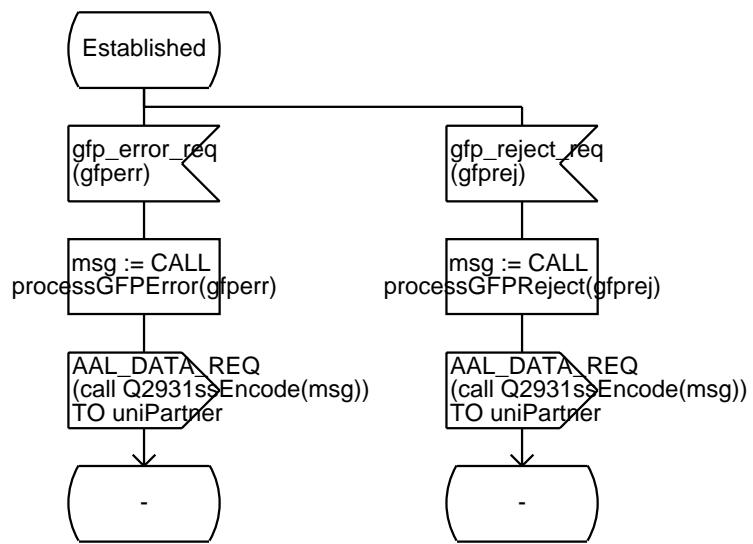


Annex B: Coord_PT

VIRTUAL Process Type <<Block Type Q2931ss_BT>> Coord_PT

44(50)

INHERITS MsgHandler_PT;fpar uniPartner PId , config InitTrunkConfig;

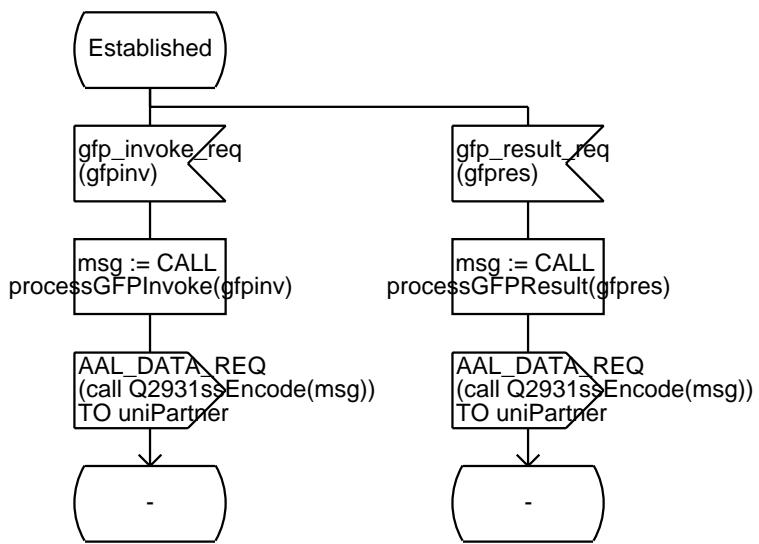


Annex B: Coord_PT

VIRTUAL Process Type <<Block Type Q2931ss_BT>> Coord_PT

45(50)

INHERITS MsgHandler_PT;fpar uniPartner PId , config InitTrunkConfig;

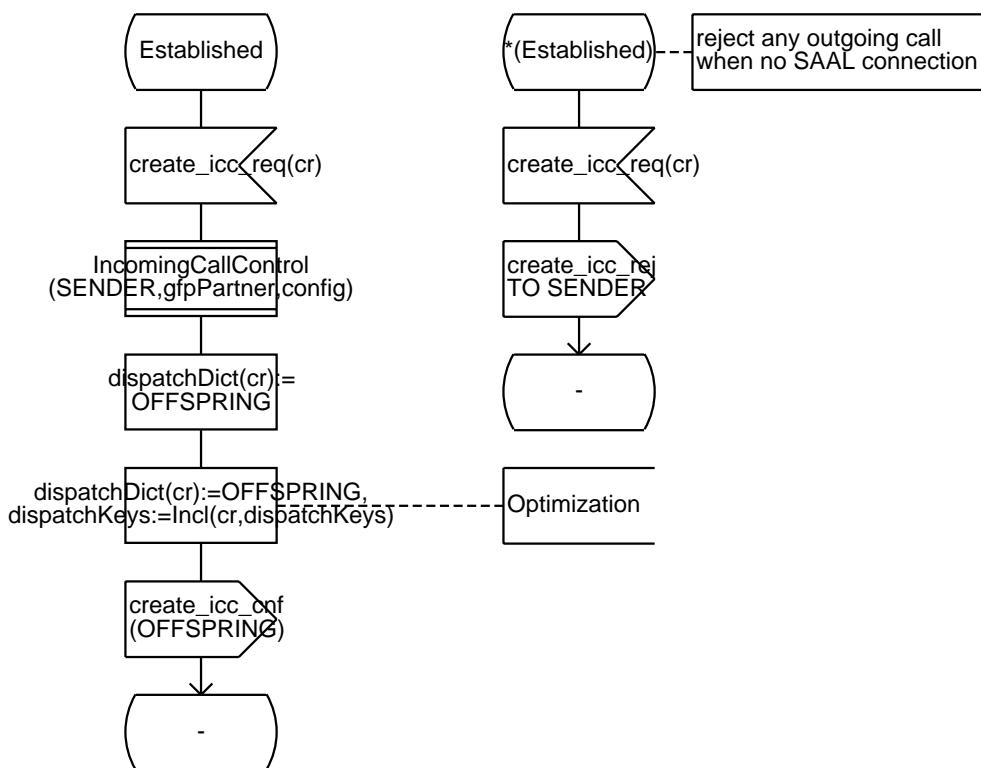


Annex B: Coord_PT

VIRTUAL Process Type <<Block Type Q2931ss_BT>> Coord_PT

46(50)

INHERITS MsgHandler_PT;fpar uniPartner PId , config InitTrunkConfig;

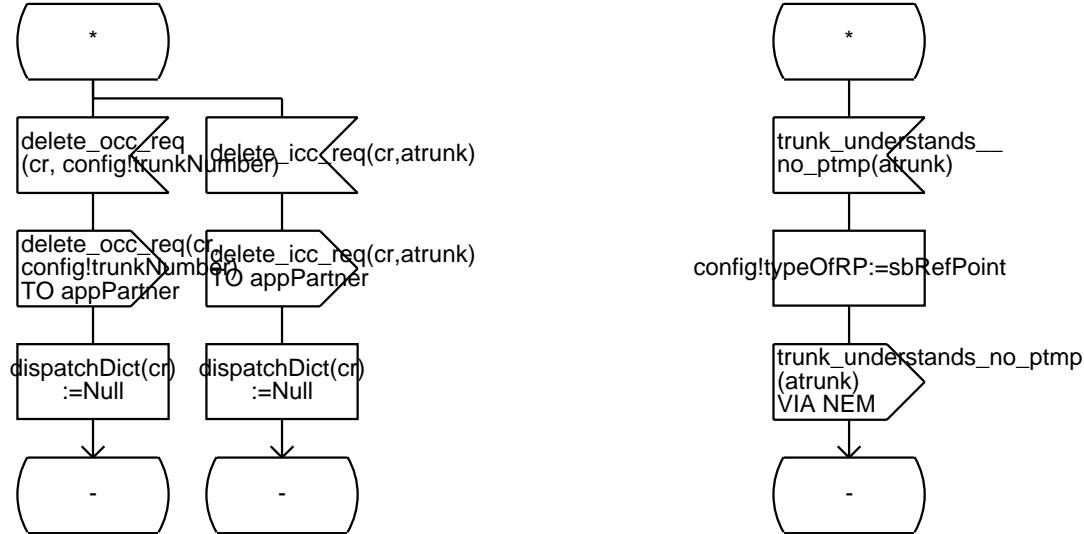


Annex B: Coord_PT

VIRTUAL Process Type <<Block Type Q2931ss_BT>> Coord_PT

47(50)

INHERITS MsgHandler_PT;fpar uniPartner PId , config InitTrunkConfig;

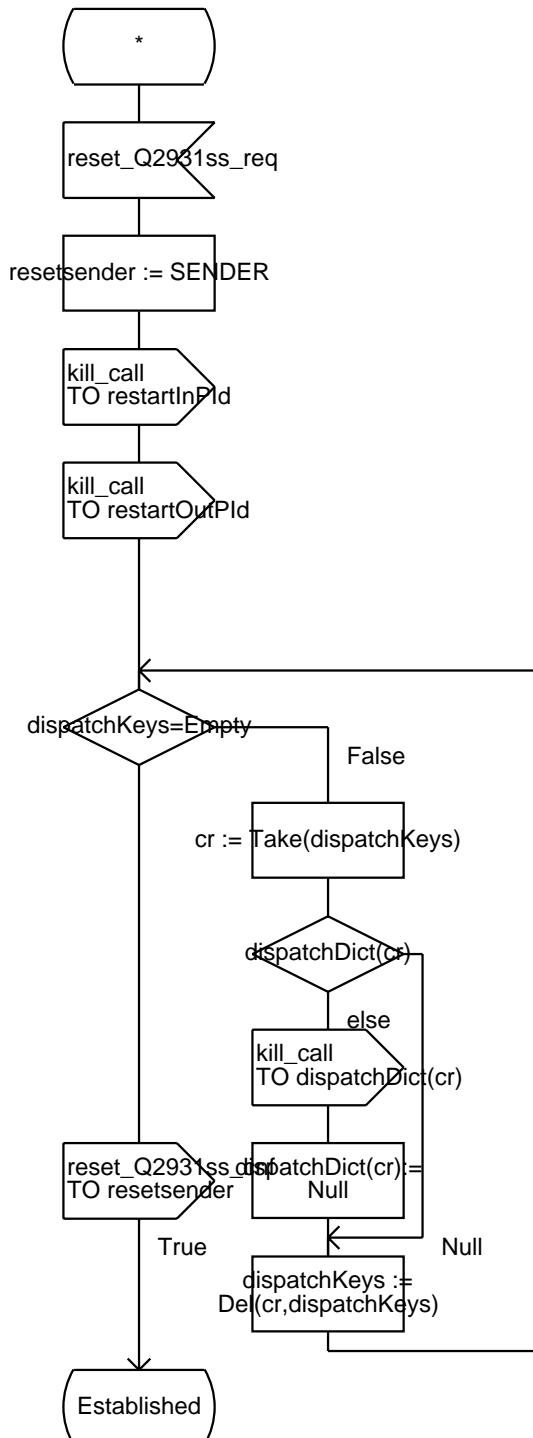


Annex B: Coord_PT

VIRTUAL Process Type <<Block Type Q2931ss_BT>> Coord_PT

48(50)

INHERITS MsgHandler_PT;fpar uniPartner PId , config InitTrunkConfig;



Annex B: Coord_PT

VIRTUAL Process Type <<Block Type Q2931ss_BT>> Coord_PT

49(50)

INHERITS MsgHandler_PT;fpar uniPartner PId , config InitTrunkConfig;

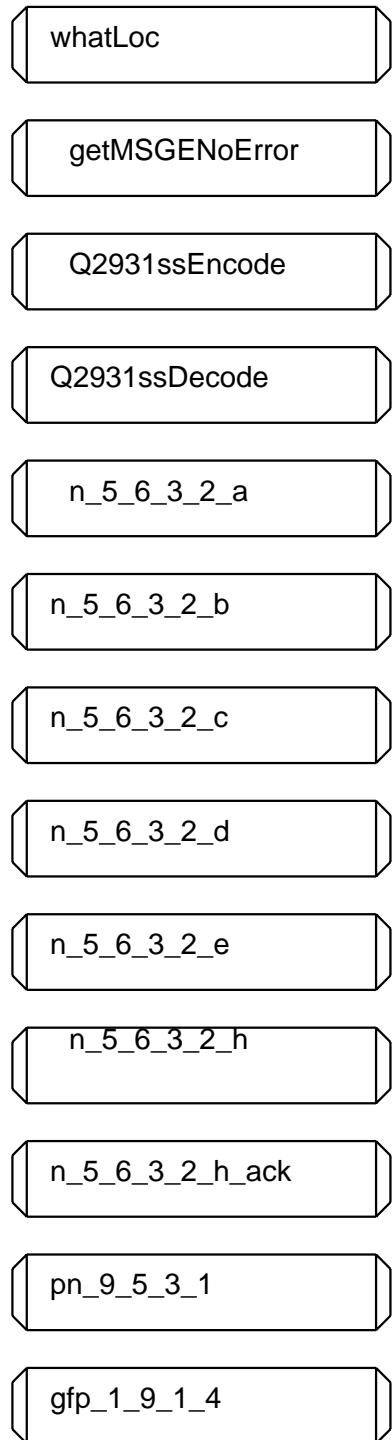


Annex B: Coord_PT

VIRTUAL Process Type <<Block Type Q2931ss_BT>> Coord_PT

50(50)

INHERITS MsgHandler_PT;fpar uniPartner PId , config InitTrunkConfig;

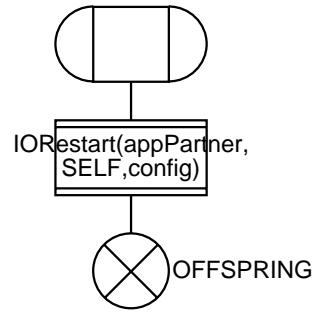


Annex B: createRestartProc

Procedure <<Process Type Coord_PT>> createRestartProc

1(1)

;RETURNS PId;

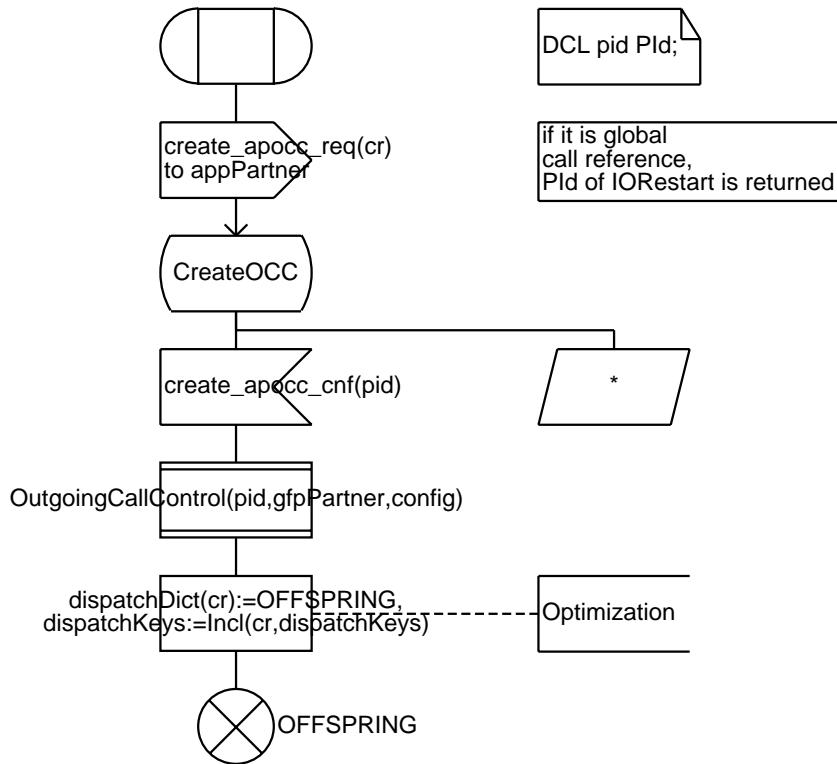


Annex B: createOCC

Procedure createOCC

1(1)

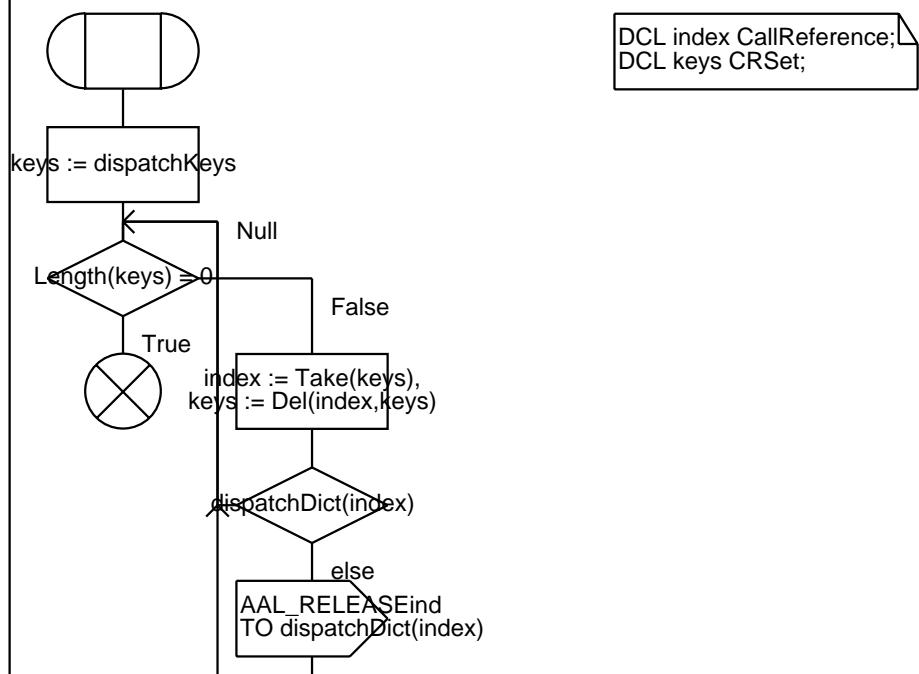
:fpar cr CallReference;
returns Plid;



Annex B: sendAALReleaseInd

Procedure sendAALReleaselnd

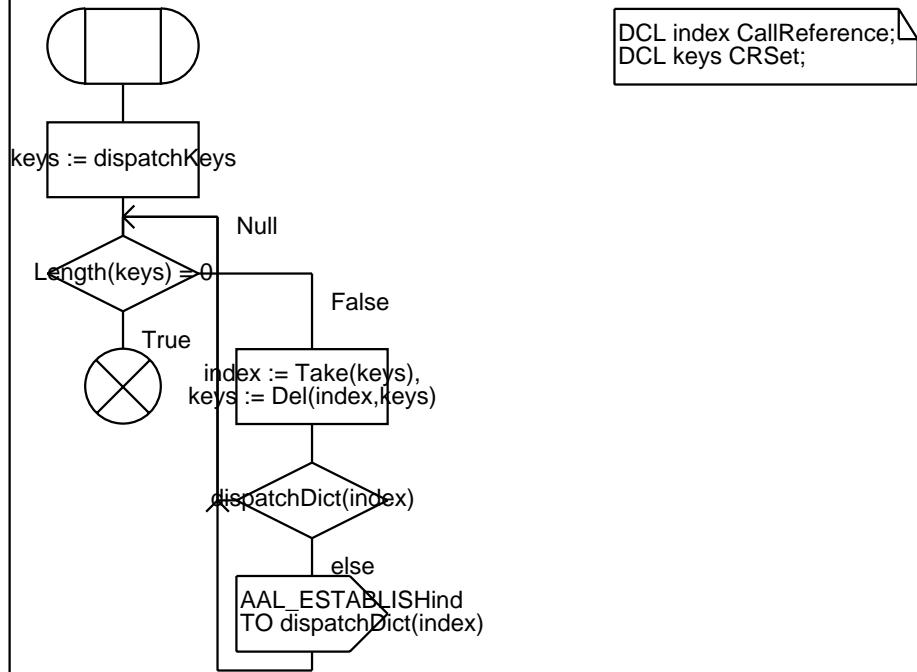
1(1)



Annex B: sendAALEstablishInd

Procedure sendAALEstablishInd

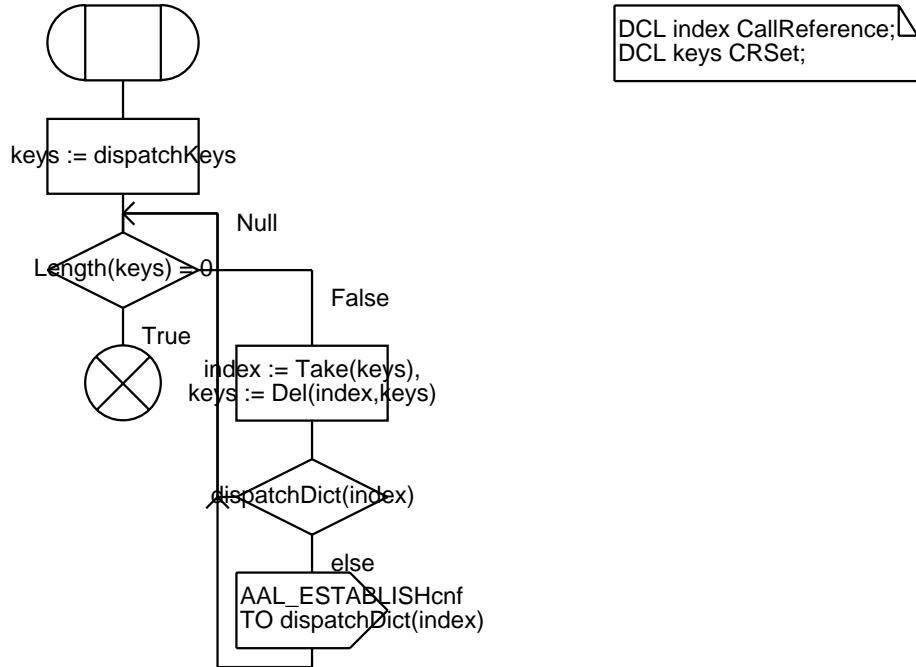
1(1)



Annex B: sendAALEstablishCnf

Procedure sendAALEstablishCnf

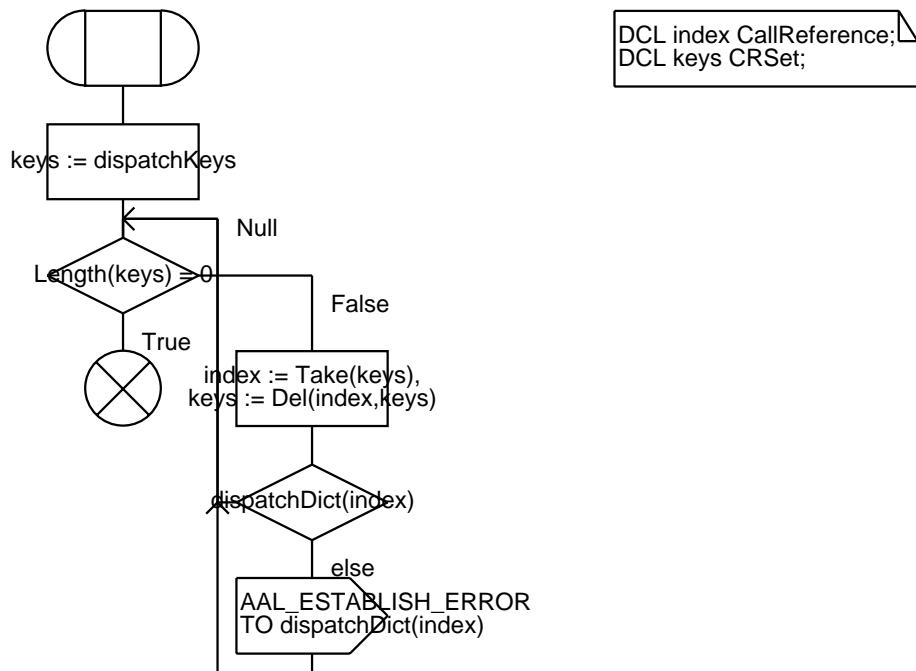
1(1)



Annex B: sendAALEstablishError

Procedure sendAALEstablishError

1(1)

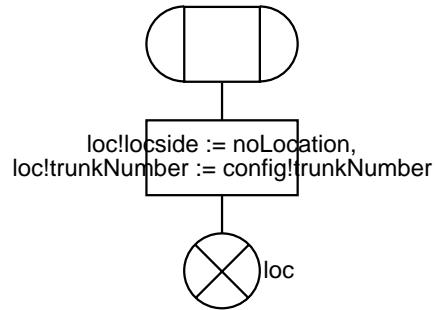


Annex B: whatLoc

Procedure <<Process Type Coord_PT>> whatLoc

1(1)

;RETURNS loc Location;



Annex B: getMSGNoError

PROCEDURE getMSGNoError

DefinitionPage(1)

;RETURNS MsgError;

/*#include '..\sdl\getmsgnoerror.sdl' */

Annex B: Q2931ssEncode

Procedure Q2931ssEncode

1(1)

```
; FPAR msg Q2931ssMessage;
RETURNS AALData;
```

```
/*#include '../sdl/q2931ssencode.sdl' */
```

Annex B: Q2931ssDecode

Procedure Q2931ssDecode

1(1)

; FPAR in data AALData,
in/out msge MsgError, in/out msg Q2931ssMessage;

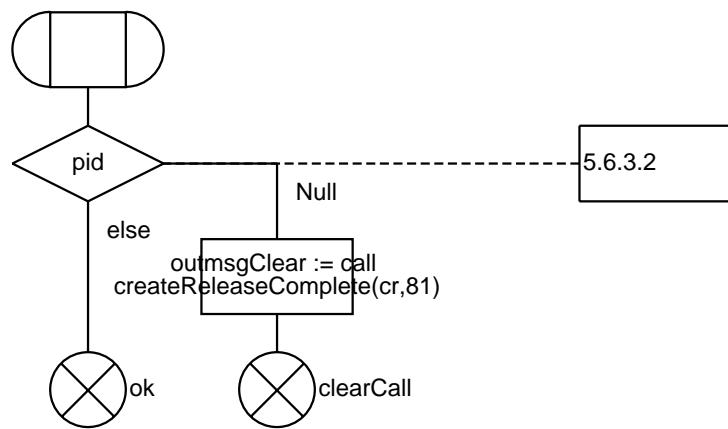
```
/*#include '../sdl/q2931ssdecode.sdl' */
```

Annex B: n_5_6_3_2_a

Procedure <>Process Type Coord_PT>> n_5_6_3_2_a

1(1)

```
;fpar cr CallReference,pid PId;
returns CheckResultType;
```

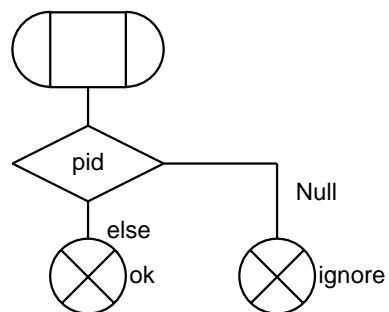


Annex B: n_5_6_3_2_b

Procedure n_5_6_3_2_b

1(1)

```
;fpar pid Pld;
returns CheckResultType;
```

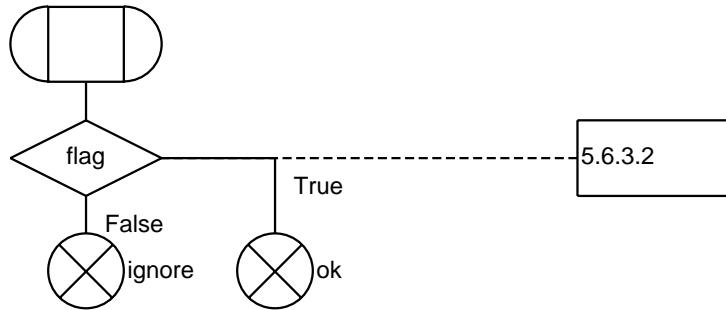


Annex B: n_5_6_3_2_c

Procedure <<Process Type Coord_PT>> n_5_6_3_2_c

1(1)

```
;fpar flag Flag;  
returns CheckResultType;
```

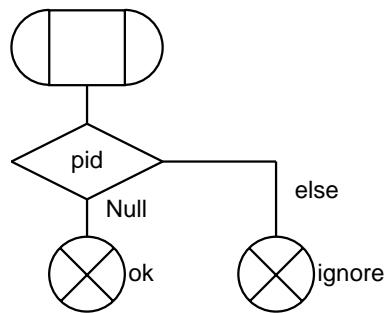


Annex B: n_5_6_3_2_d

Procedure n_5_6_3_2_d

1(1)

```
;fpar pid Pld;
returns CheckResultType;
```

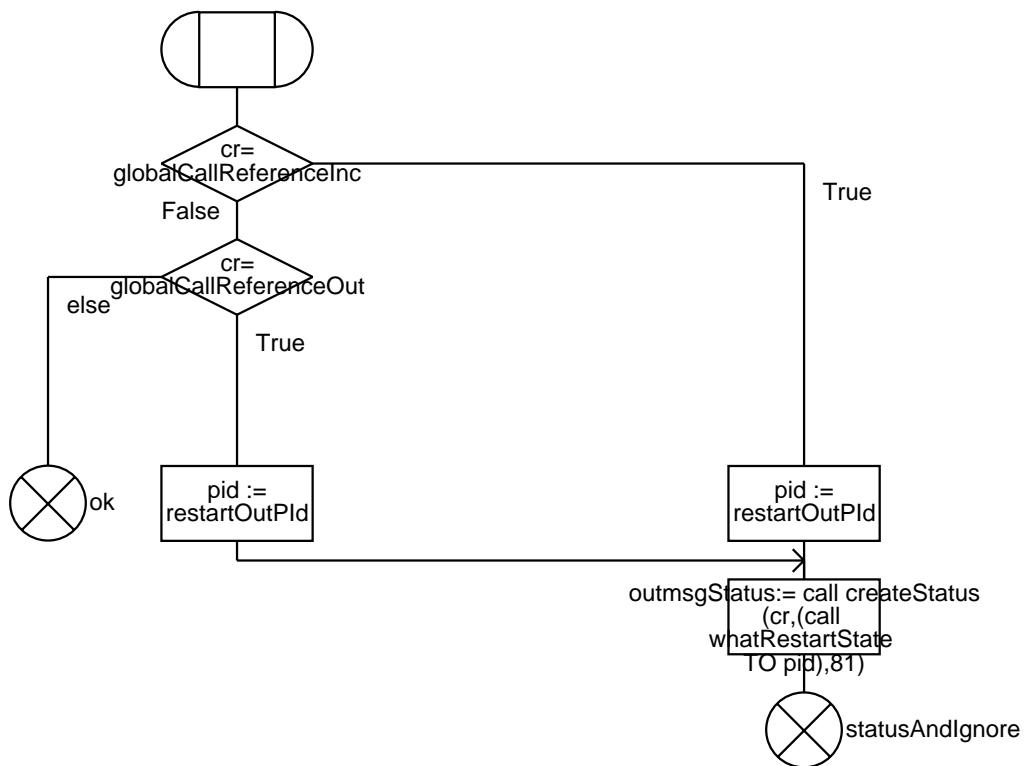


Annex B: n_5_6_3_2_e

Procedure n_5_6_3_2_e

1(1)

```
;fpar cr CallReference;
returns CheckResultType;
```

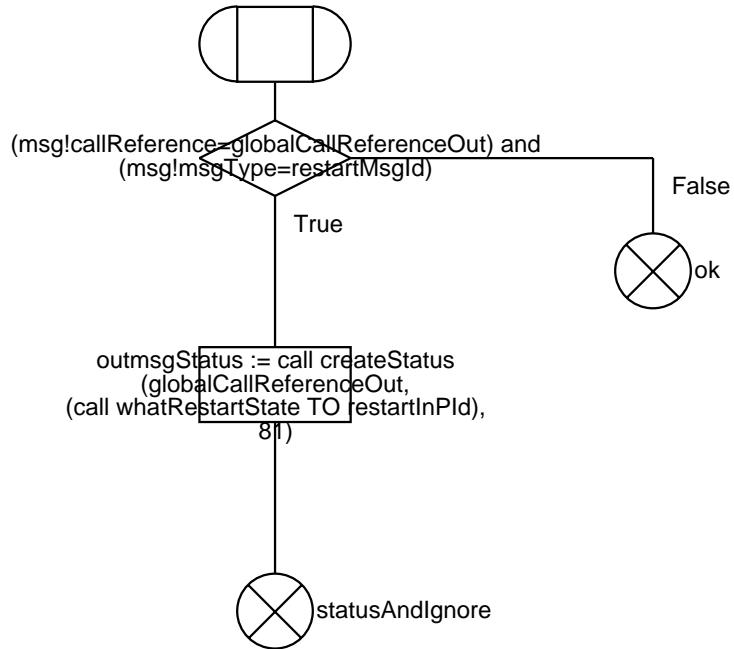


Annex B: n_5_6_3_2_h

Procedure n_5_6_3_2_h

1(1)

```
;fpar msg Q2931ssMessage;
returns CheckResultType;
```

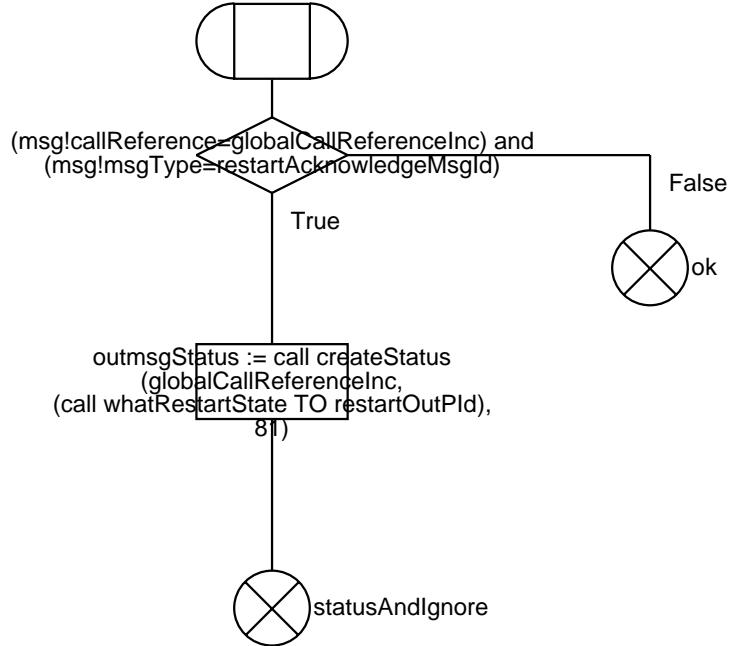


Annex B: n_5_6_3_2_h_ack

Procedure n_5_6_3_2_h_ack

1(1)

```
;fpar msg Q2931ssMessage;
returns CheckResultType;
```

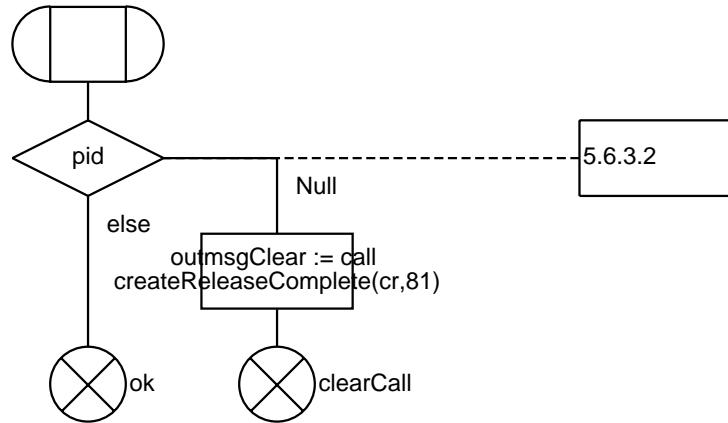


Annex B: pn_9_5_3_1

Procedure <>Process Type Coord_PT>> pn_9_5_3_1

1(1)

```
;fpar cr CallReference,pid PId;  
returns CheckResultType;
```

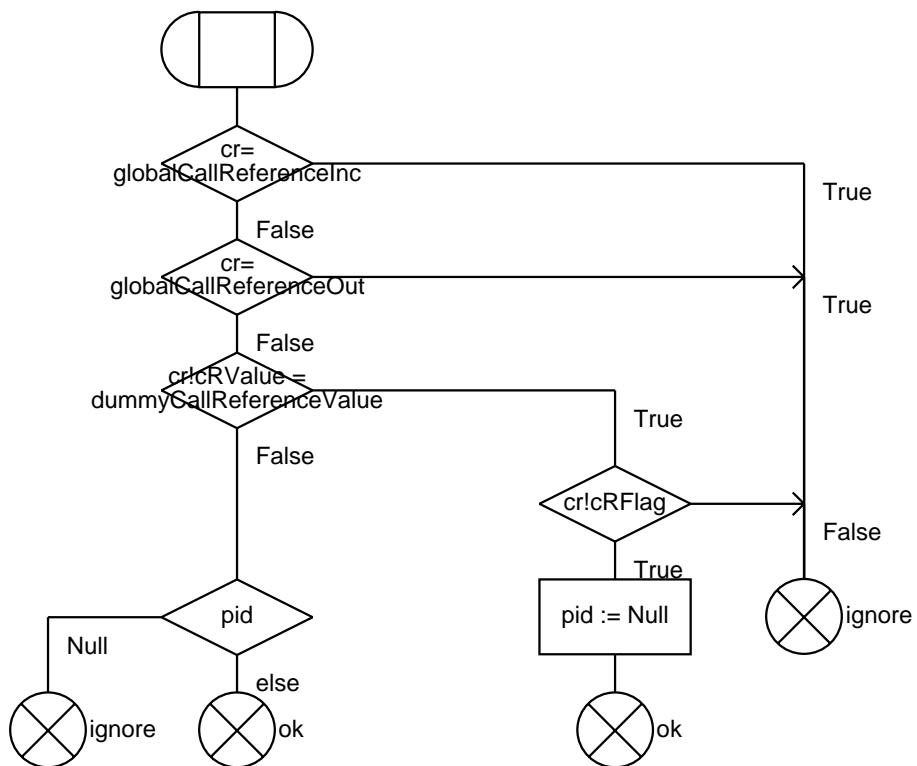


Annex B: gfp_1_9_1_4

Procedure gfp_1_9_1_4

1(1)

:FPAR cr CallReference, in/out pid PId;
 RETURNS CheckResultType;



Annex B: IncomingCallControl_PT

Process Type <<Block Type Q2931ss_BT>> IncomingCallControl_PT

1(21)

INHERITS CallControl_PT;

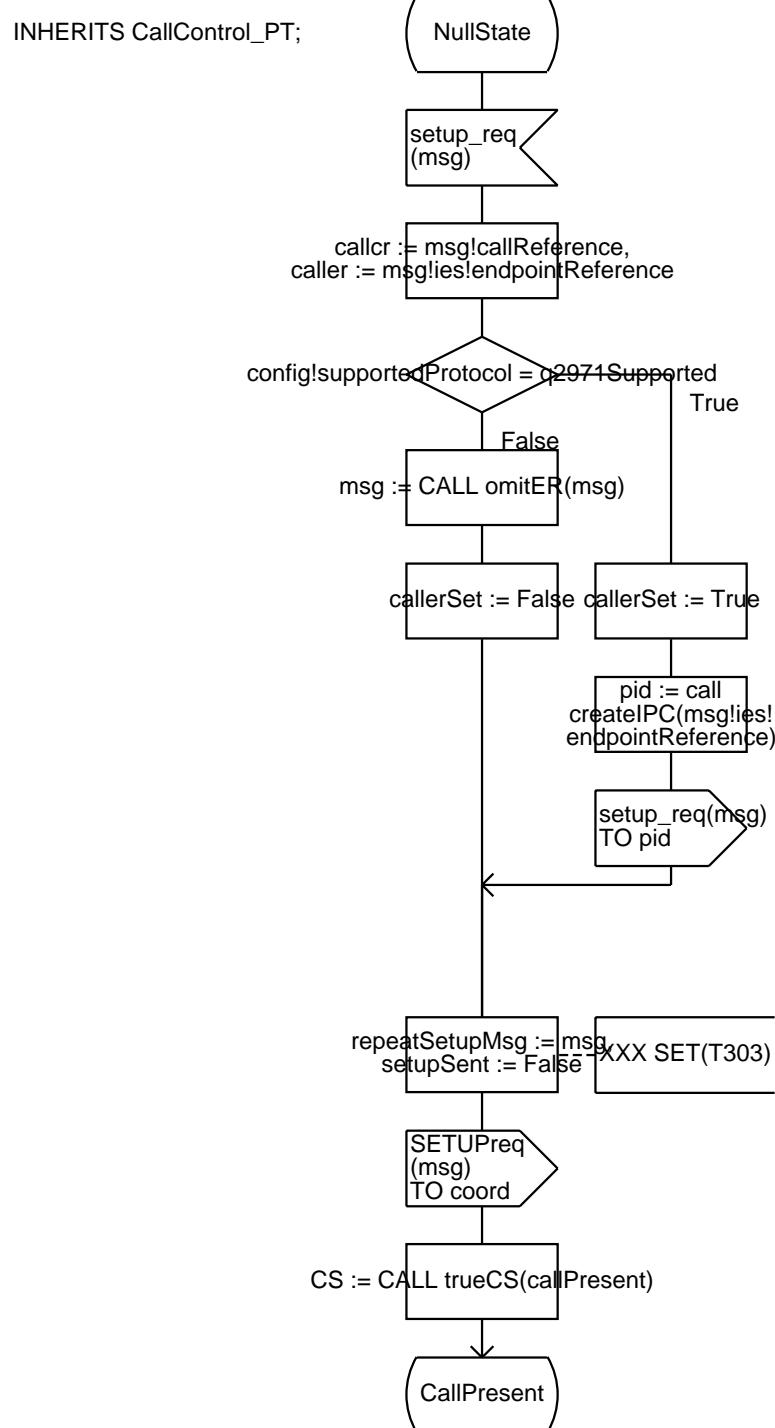
```
DCL repeatSetupMsg Q2931ssMessage;
/* setup to repeat after expiry of t303 */
DCL setupSent Boolean := False;
/* true if setup already repeated */
```

```
Timer T303 := 40; /* 4 seconds */
Timer T310 := 10; /* 10 seconds */
Timer T301 := 180; /* 3 min */
```

Annex B: IncomingCallControl_PT

Process Type <<Block Type Q2931ss_BT>> IncomingCallControl_PT

2(21)

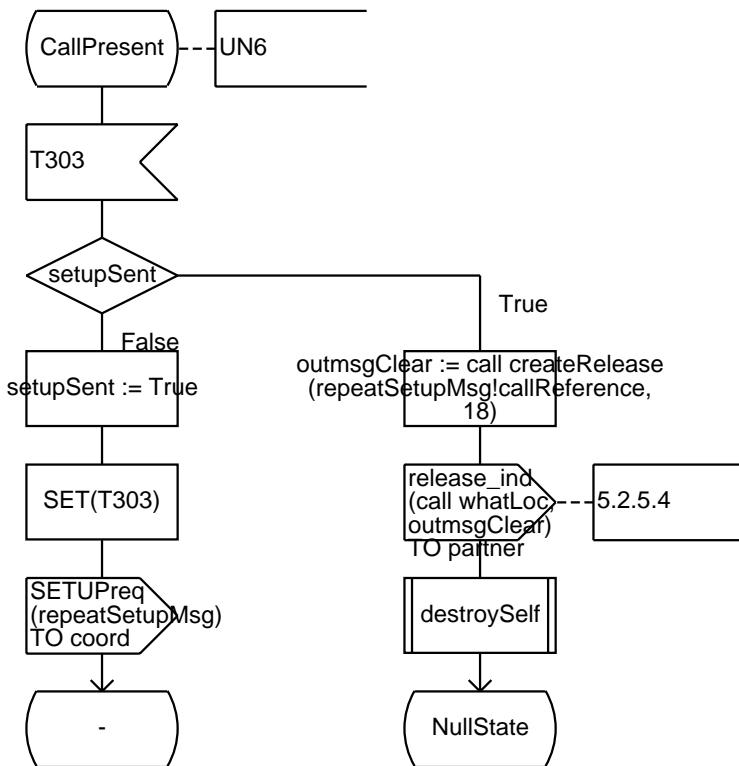


Annex B: IncomingCallControl_PT

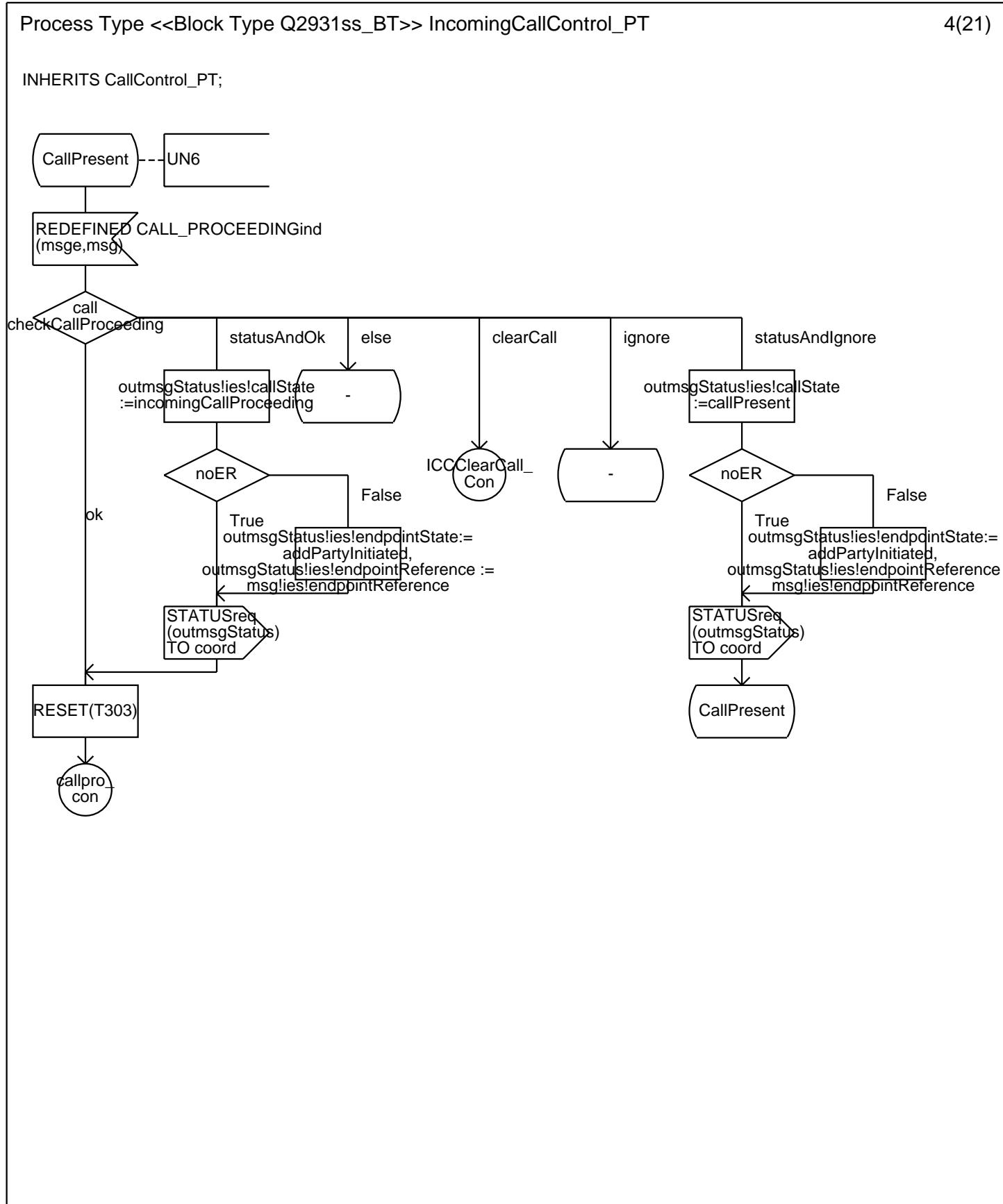
Process Type <<Block Type Q2931ss_BT>> IncomingCallControl_PT

3(21)

INHERITS CallControl_PT;



Annex B: IncomingCallControl_PT

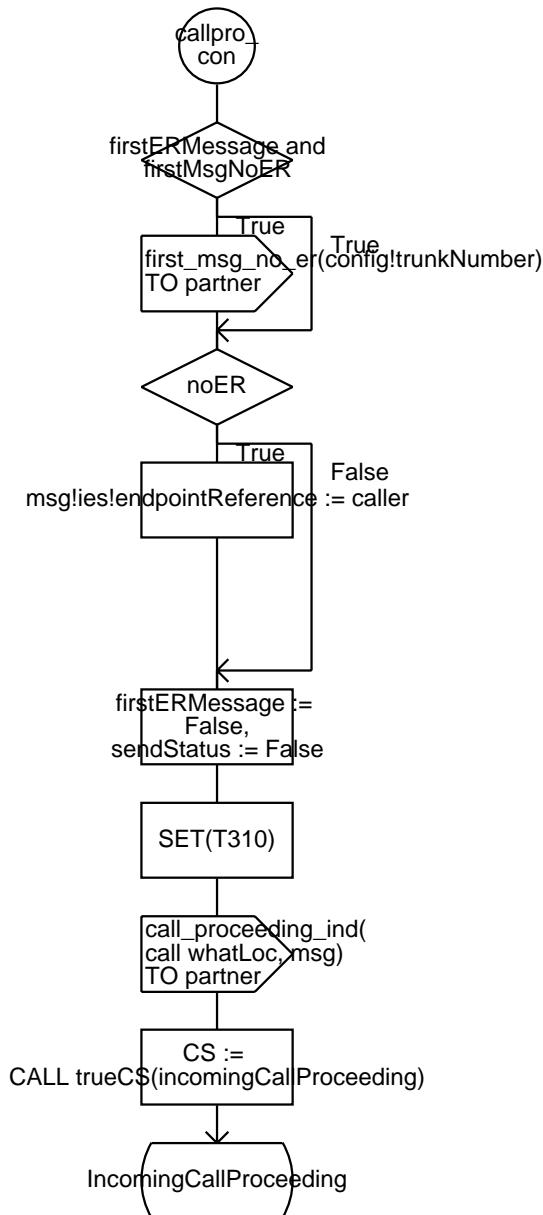


Annex B: IncomingCallControl_PT

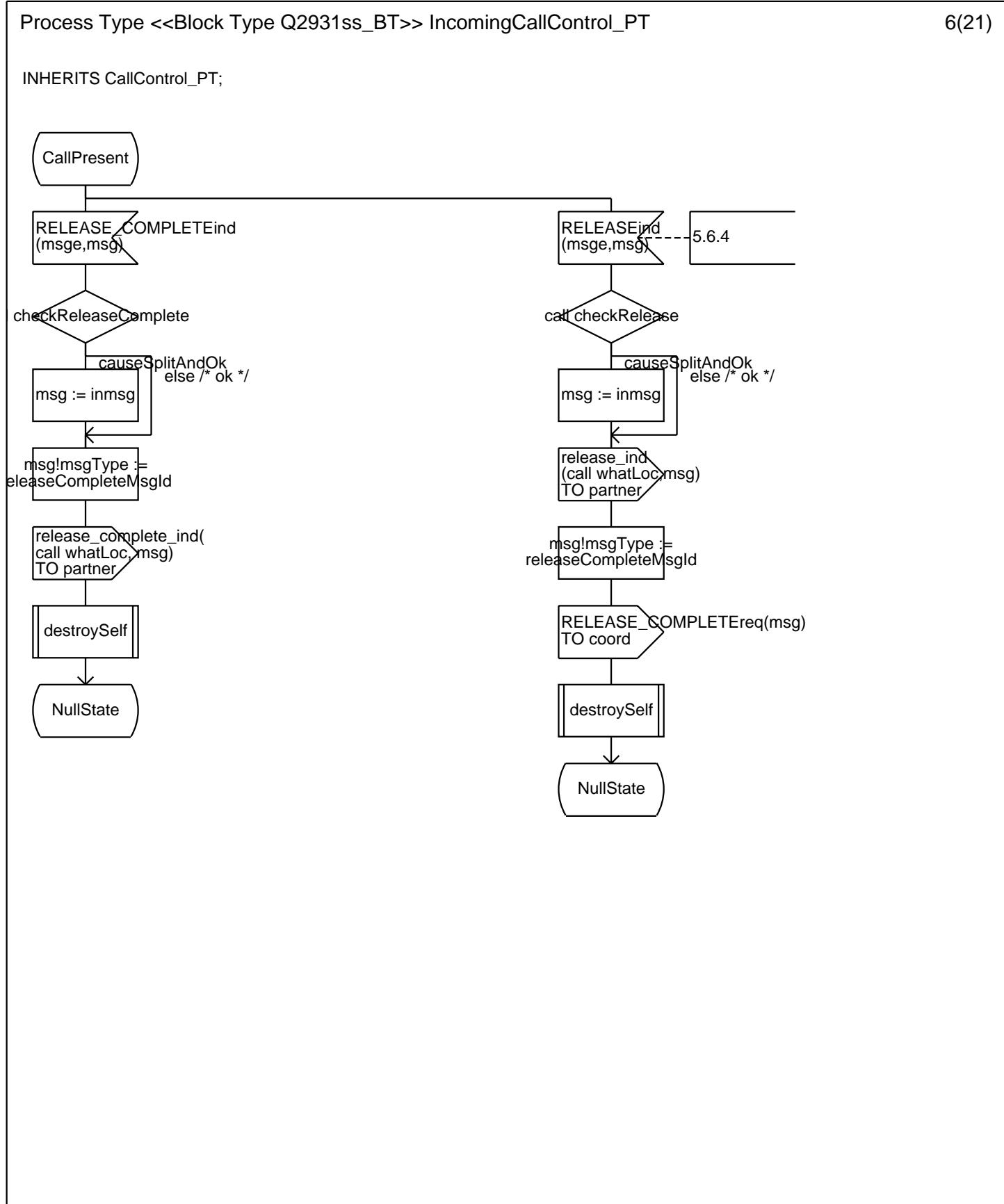
Process Type <<Block Type Q2931ss_BT>> IncomingCallControl_PT

5(21)

INHERITS CallControl_PT;



Annex B: IncomingCallControl_PT

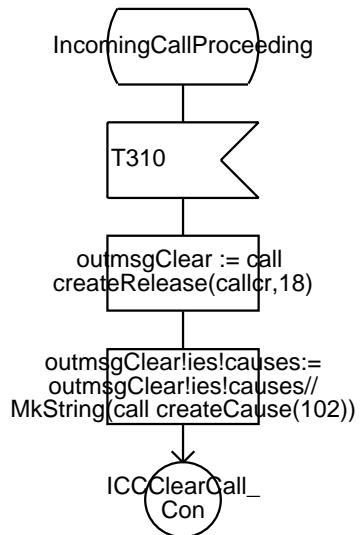


Annex B: IncomingCallControl_PT

Process Type <<Block Type Q2931ss_BT>> IncomingCallControl_PT

7(21)

INHERITS CallControl_PT;

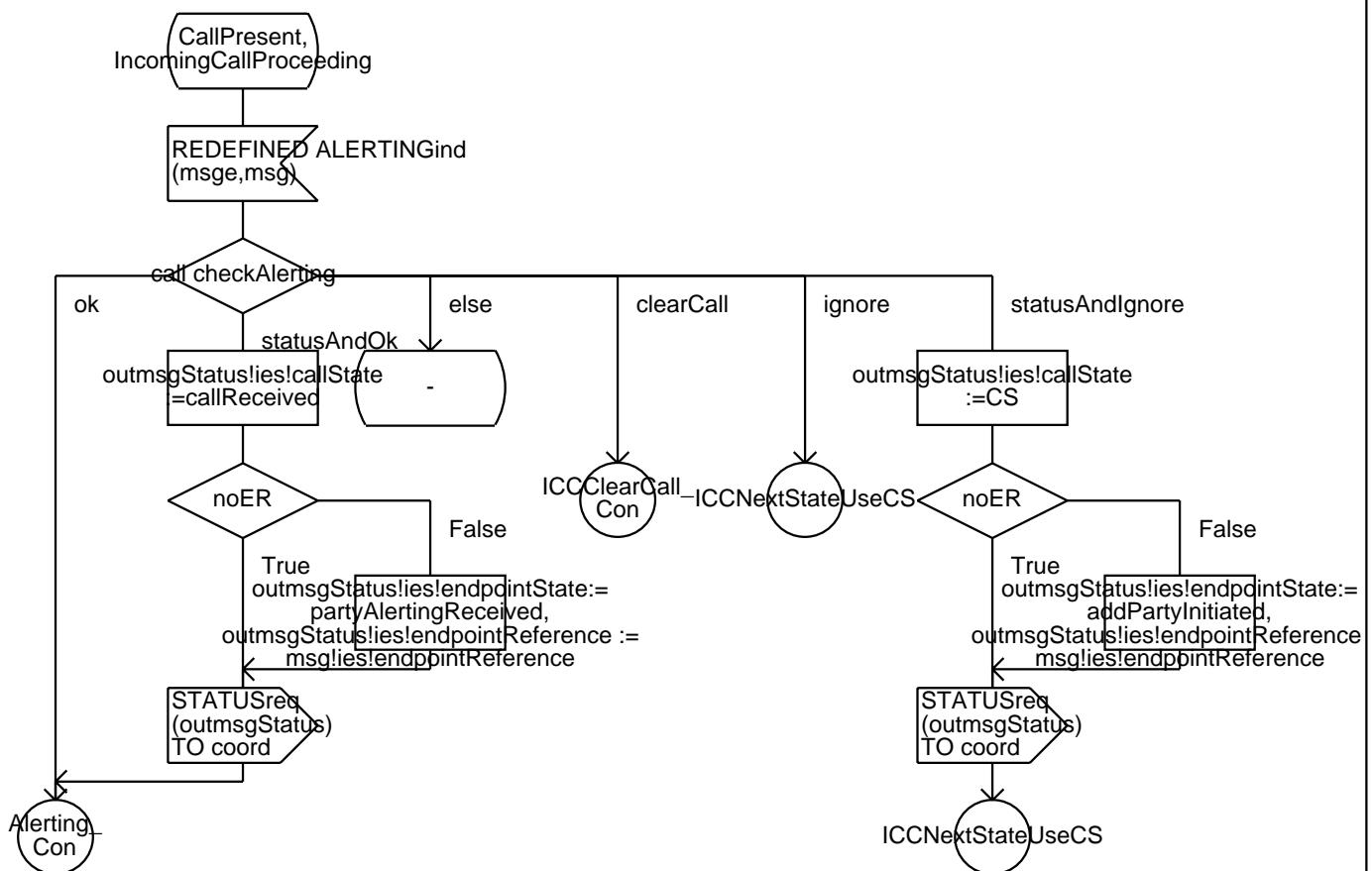


Annex B: IncomingCallControl_PT

Process Type <<Block Type Q2931ss_BT>> IncomingCallControl_PT

8(21)

INHERITS CallControl_PT;

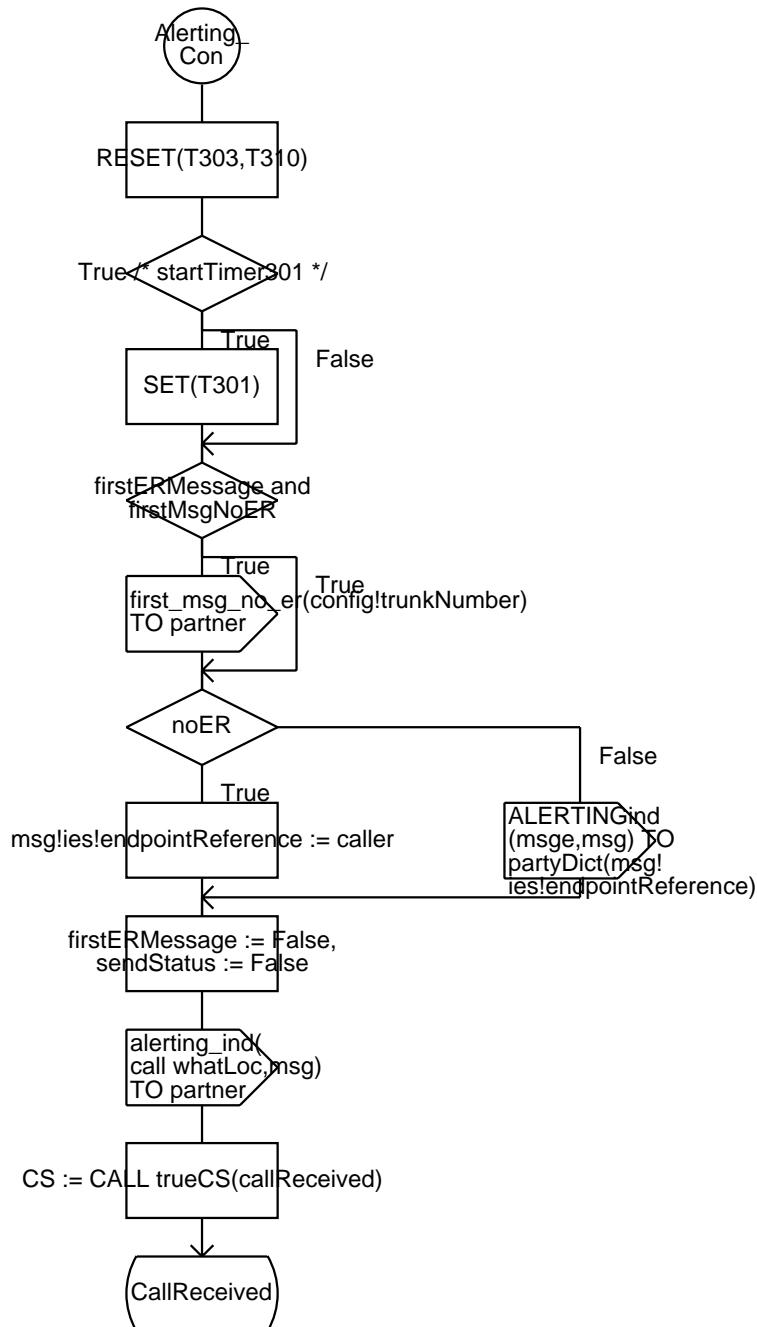


Annex B: IncomingCallControl_PT

Process Type <<Block Type Q2931ss_BT>> IncomingCallControl_PT

9(21)

INHERITS CallControl_PT;

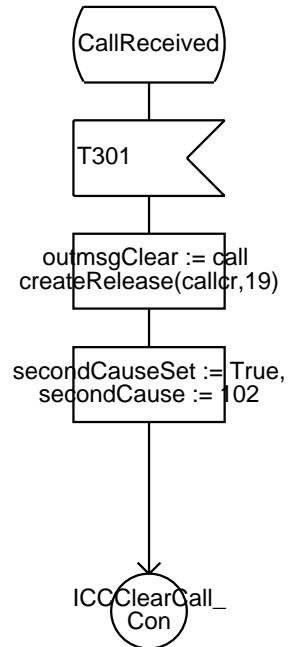


Annex B: IncomingCallControl_PT

Process Type <<Block Type Q2931ss_BT>> IncomingCallControl_PT

10(21)

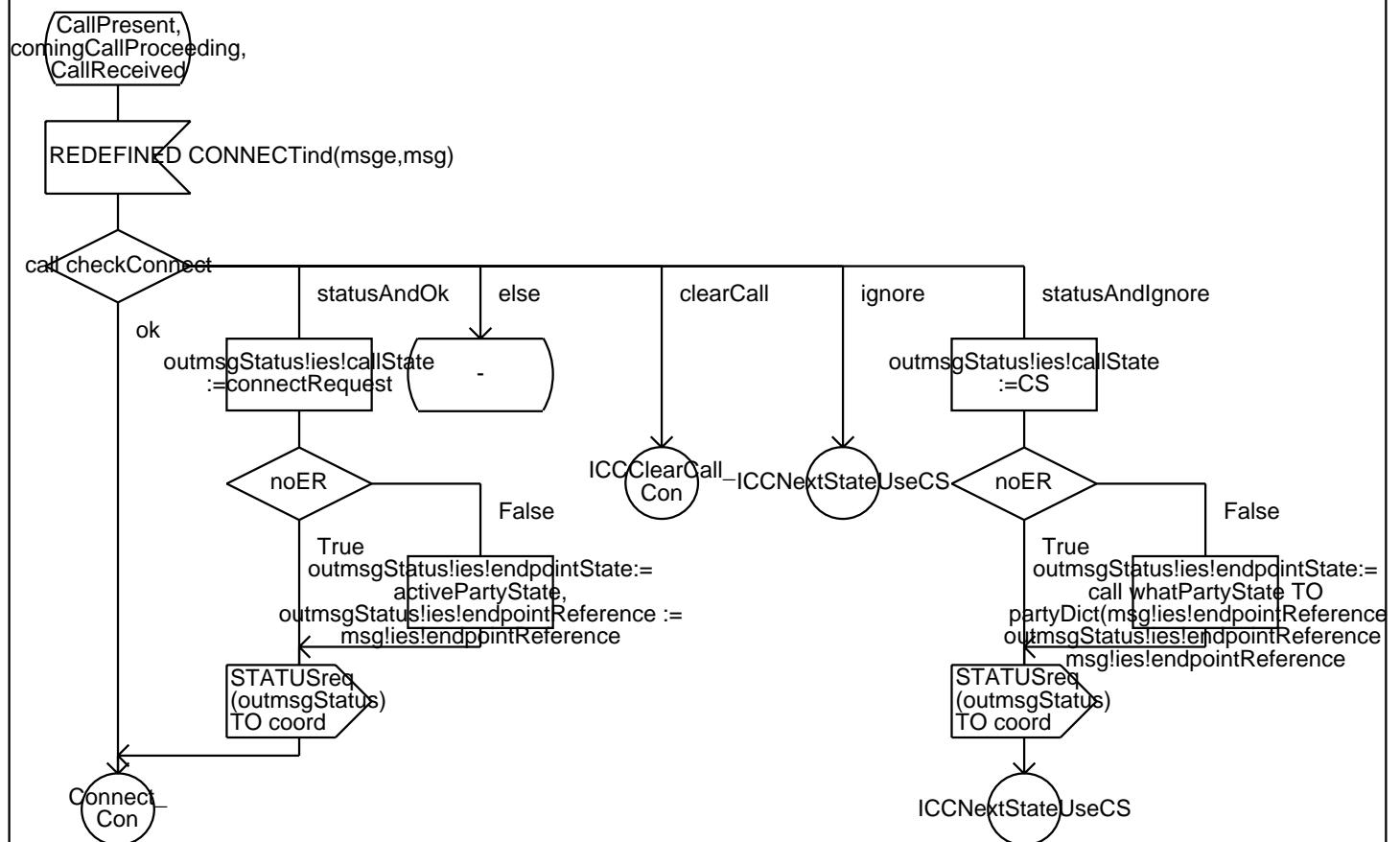
INHERITS CallControl_PT;



Annex B: IncomingCallControl_PT

11(21)

INHERITS CallControl_PT;

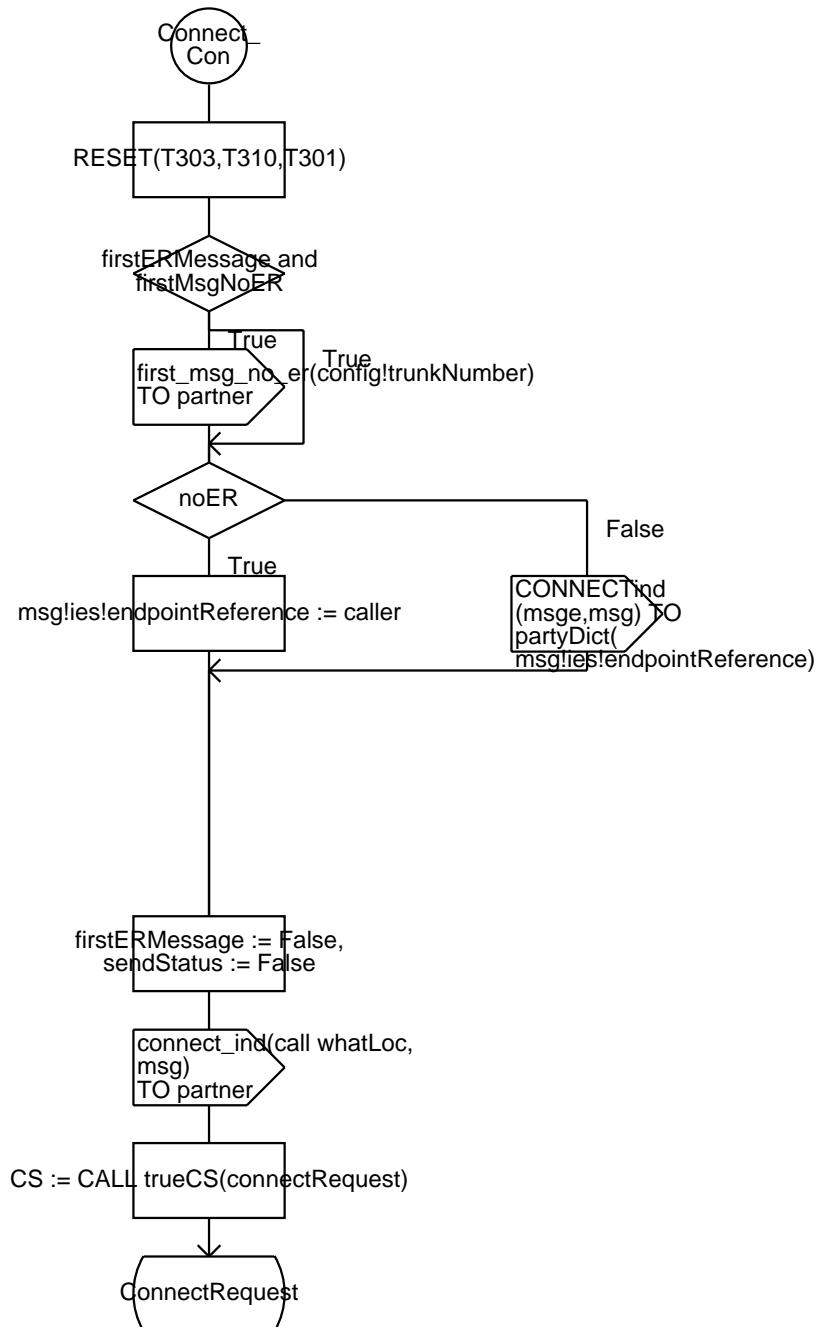


Annex B: IncomingCallControl_PT

Process Type <<Block Type Q2931ss_BT>> IncomingCallControl_PT

12(21)

INHERITS CallControl_PT;

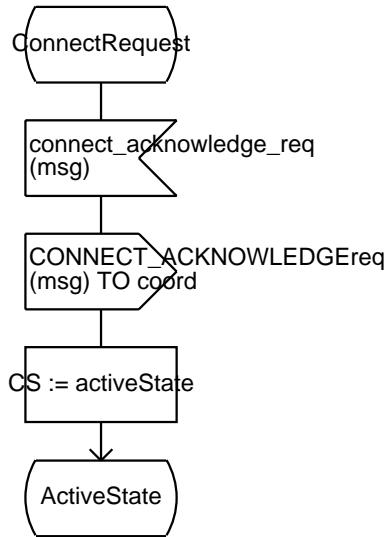


Annex B: IncomingCallControl_PT

Process Type <<Block Type Q2931ss_BT>> IncomingCallControl_PT

13(21)

INHERITS CallControl_PT;

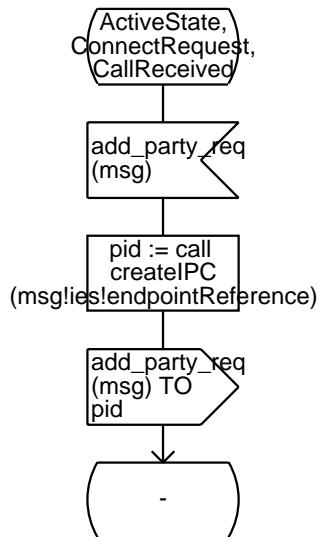


Annex B: IncomingCallControl_PT

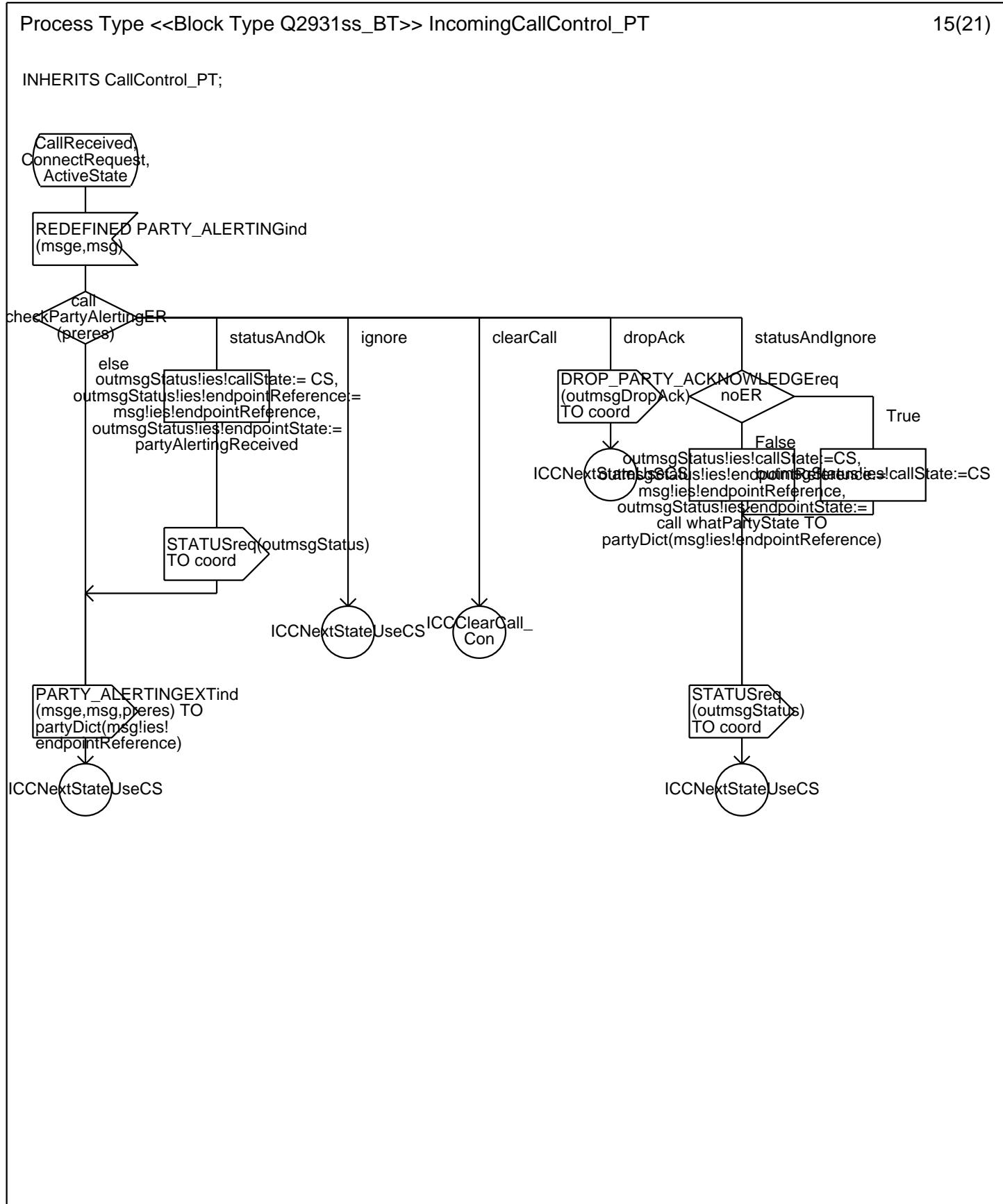
Process Type <<Block Type Q2931ss_BT>> IncomingCallControl_PT

14(21)

INHERITS CallControl_PT;



Annex B: IncomingCallControl_PT

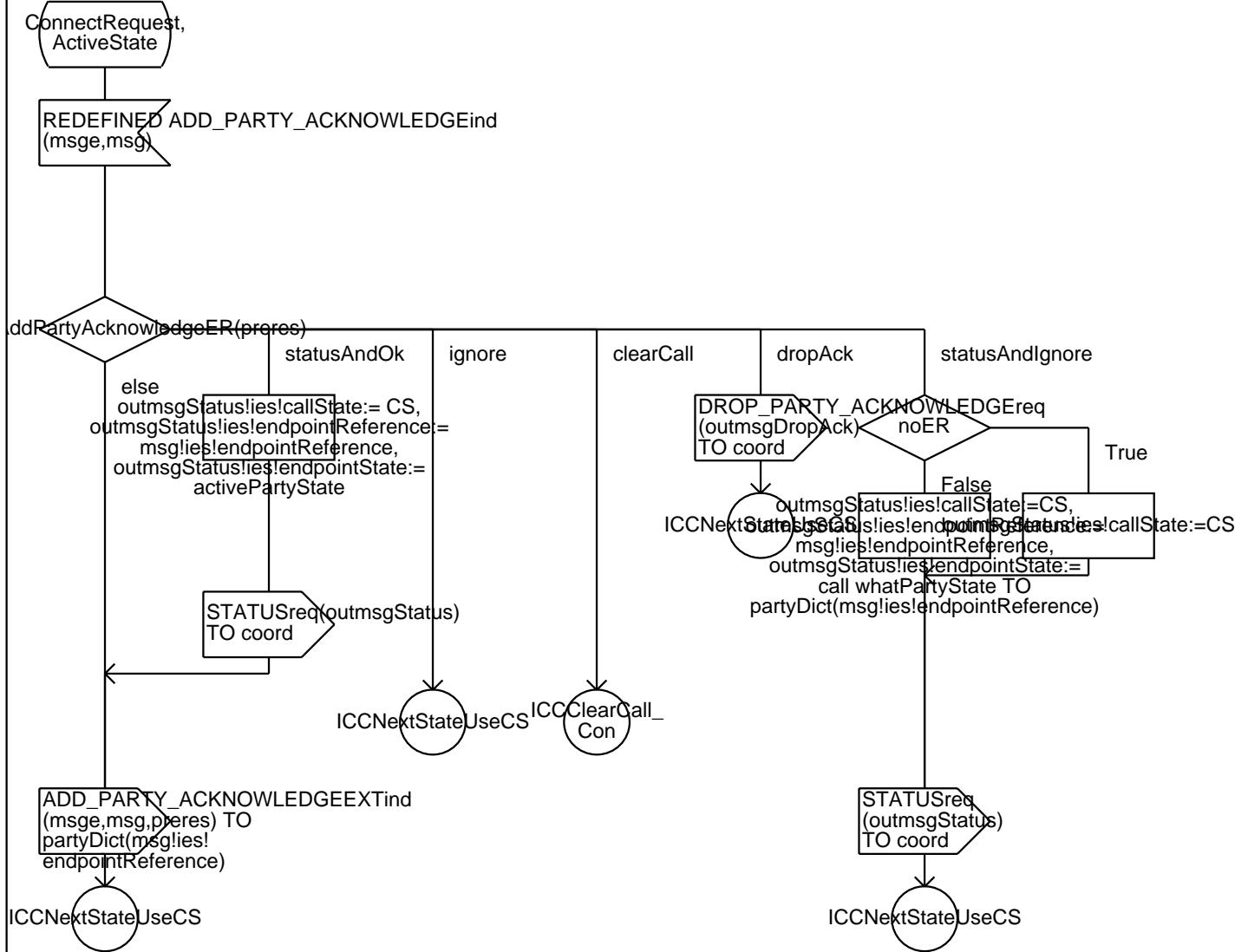


Annex B: IncomingCallControl_PT

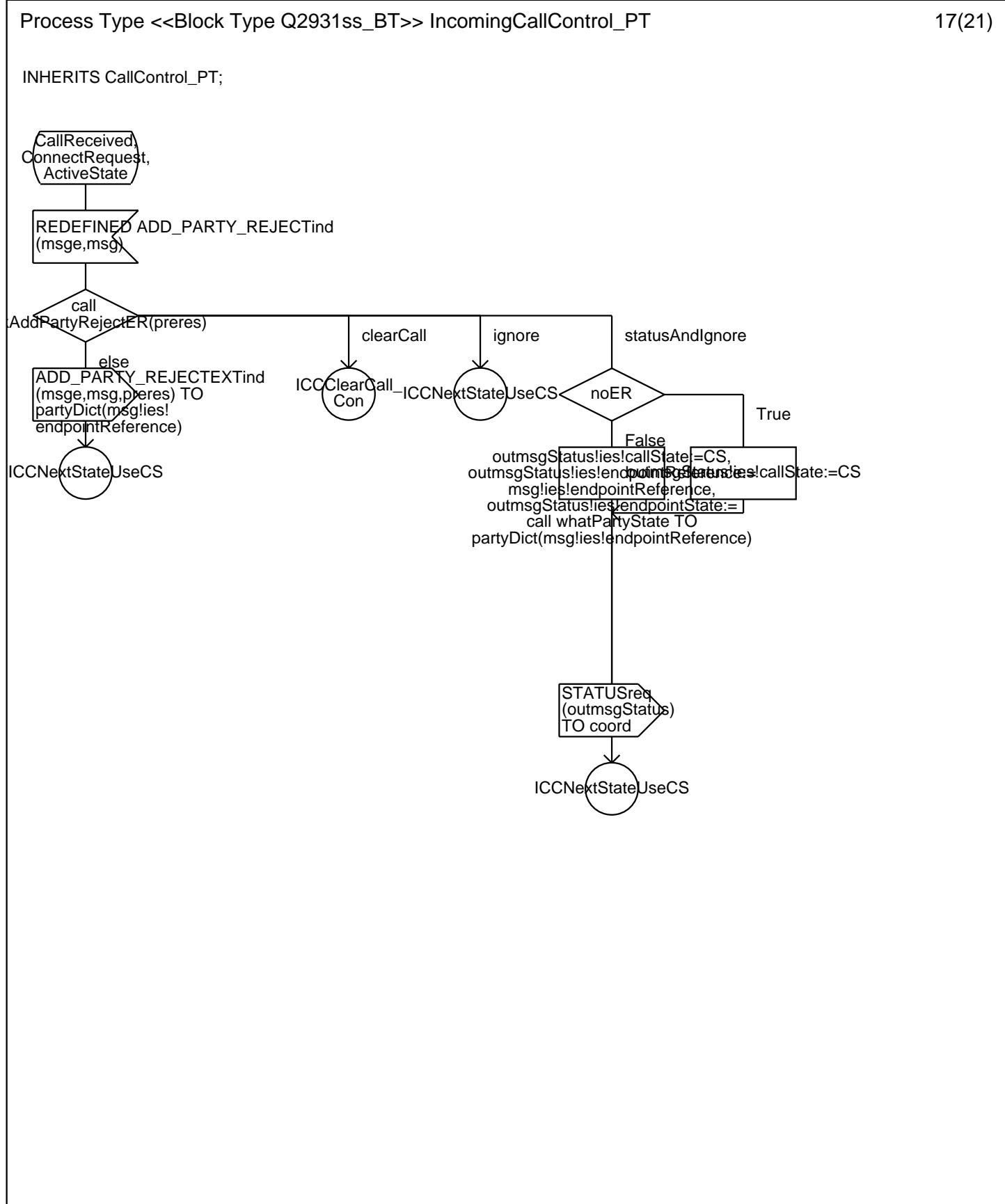
Process Type <<Block Type Q2931ss_BT>> IncomingCallControl_PT

16(21)

INHERITS CallControl_PT;



Annex B: IncomingCallControl_PT



Annex B: IncomingCallControl_PT

Process Type <<Block Type Q2931ss_BT>> IncomingCallControl_PT

18(21)

INHERITS CallControl_PT;



Annex B: IncomingCallControl_PT

Process Type <<Block Type Q2931ss_BT>> IncomingCallControl_PT

19(21)

INHERITS CallControl_PT;

REDEFINED whatLoc

checkCallProceeding

checkAlerting

checkConnect

checkPartyAlertingER

checkAddPartyAcknowledgeER

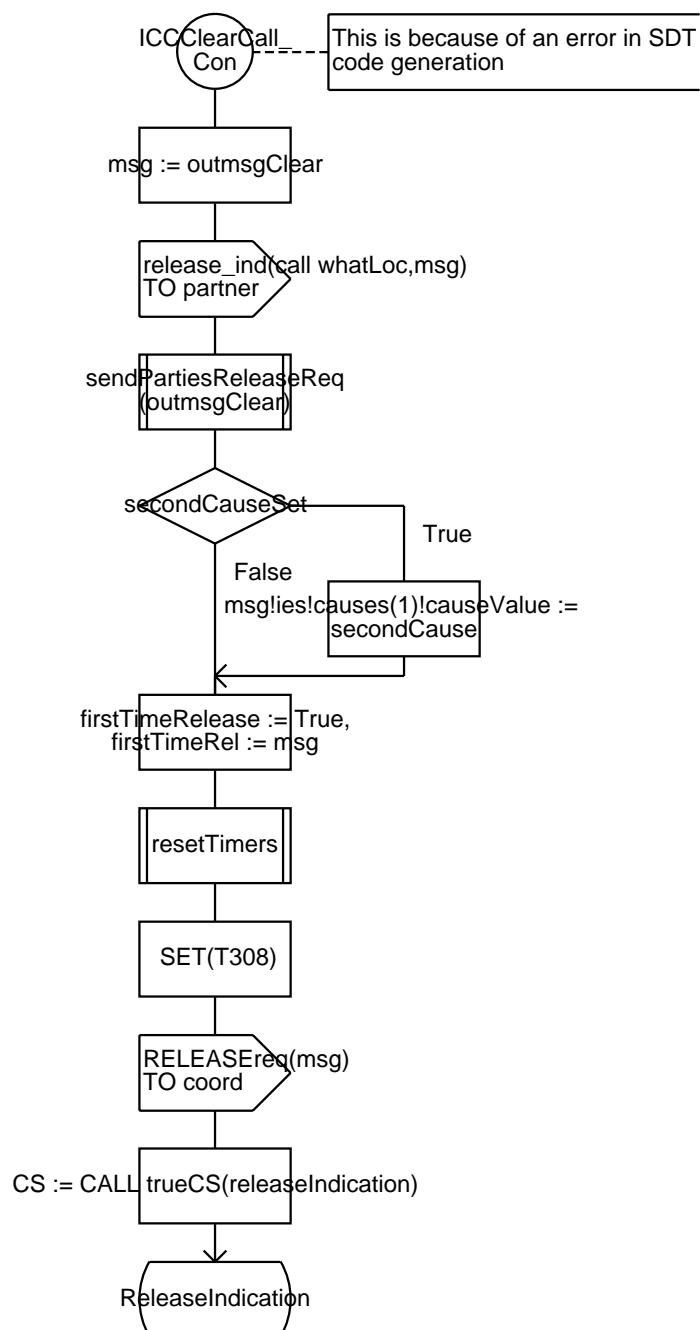
checkAddPartyRejectER

Annex B: IncomingCallControl_PT

Process Type <<Block Type Q2931ss_BT>> IncomingCallControl_PT

20(21)

INHERITS CallControl_PT;

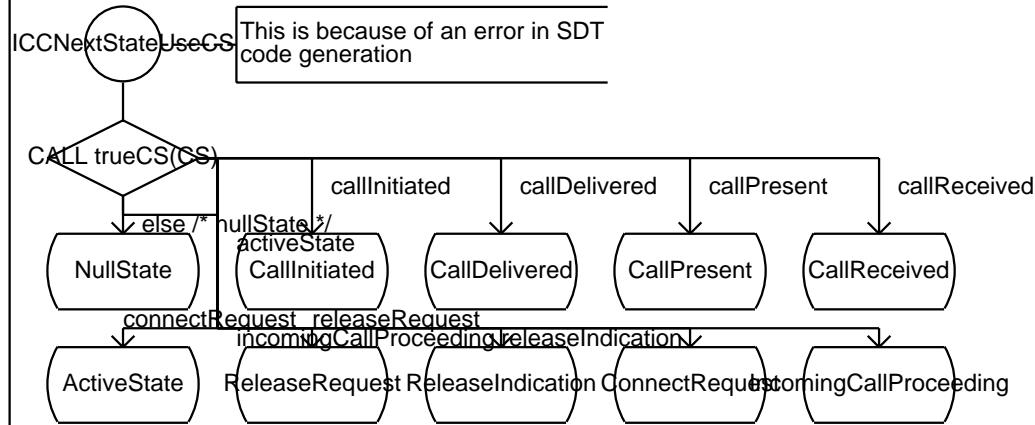


Annex B: IncomingCallControl_PT

Process Type <<Block Type Q2931ss_BT>> IncomingCallControl_PT

21(21)

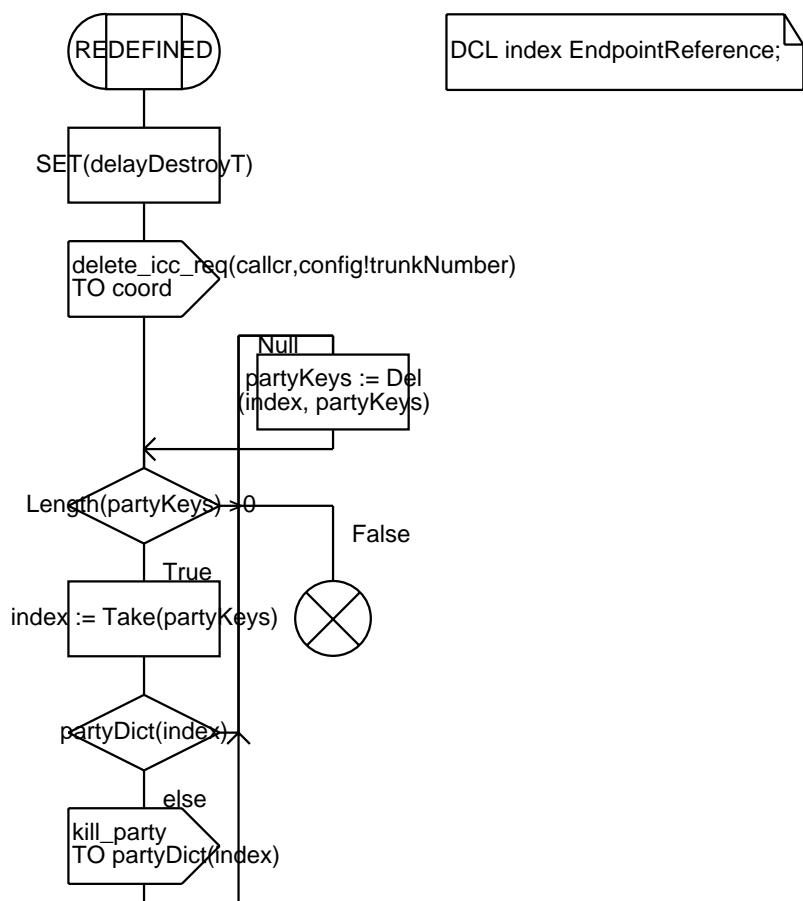
INHERITS CallControl_PT;



Annex B: destroySelf

Redefined Procedure <<Process Type IncomingCallControl_PT>> destroySelf

1(1)

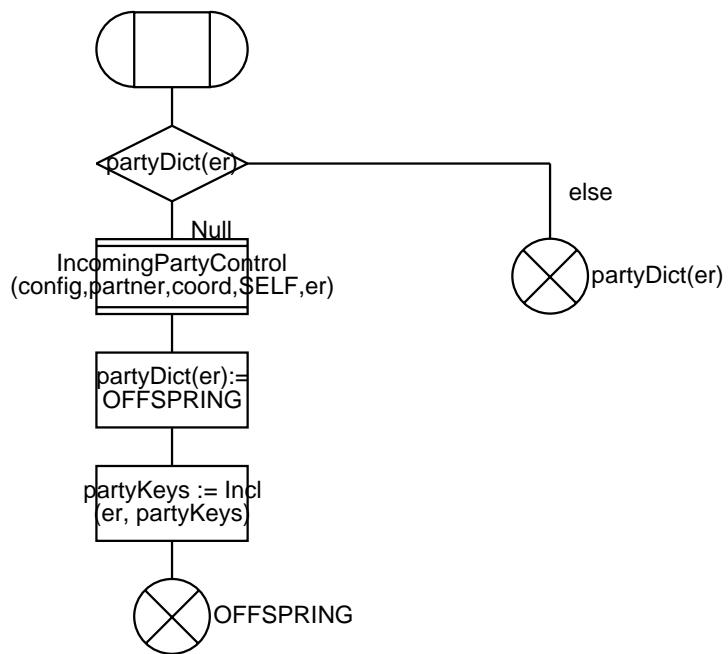


Annex B: createIPC

Procedure createIPC

1(1)

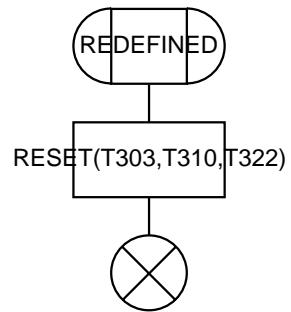
:fpar er EndpointReference;
returns Pld;



Annex B: resetTimers

Redefined Procedure <<Process Type IncomingCallControl_PT>> resetTimers

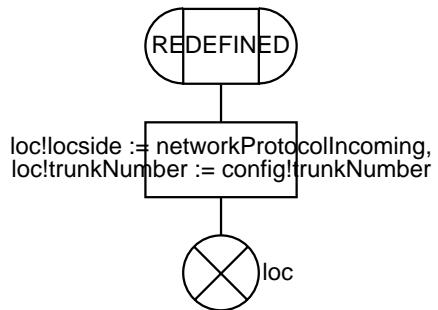
1(1)



Annex B: whatLoc

Redefined Procedure <<Process Type IncomingCallControl_PT>> whatLoc

1(1)



Annex B: checkCallProceeding

Procedure checkCallProceeding

1(1)

;RETURNS res CheckResultType;

```
/*#include '../sdl/CheckcallProceeding.sdl' */
```

Annex B: checkAlerting

Procedure <<Process Type IncomingCallControl_PT>> checkAlerting

1(1)

;RETURNS res CheckResultType;

```
/*#include '../sdl/Checkalerting.sdl' */
```

Annex B: checkConnect

Procedure checkConnect

1(1)

;RETURNS res CheckResultType;

/*#include '../sdl/Checkconnect.sdl' */

Annex B: checkPartyAlertingER

Procedure checkPartyAlertingER

1(1)

```
:FPAR in/out preres CheckResultTypeStatus;RETURNS res CheckResultType;
```

```
/*#include '../sdl/CheckpartyAlertingER.sdl' */
```

Annex B: checkAddPartyAcknowledgeER

Procedure checkAddPartyAcknowledgeER

1(1)

;FPAR in/out preres CheckResultTypeStatus;RETURNS res CheckResultType;

```
/*#include './sdl/CheckaddPartyAcknowledgeER.sdl' */
```

Annex B: checkAddPartyRejectER

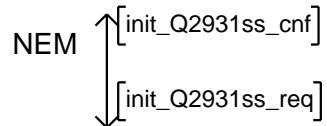
Procedure <<Process Type IncomingCallControl_PT>> checkAddPartyRejectER

1(1)

;FPAR in/out preres CheckResultTypeStatus;RETURNS res CheckResultType;

```
/*#include '../sdl/CheckaddPartyRejectER.sdl' */
```

Annex B: CoordControl_PT



Process Type <<Block Type Q2931ss_BT>> CoordControl_PT

1(2)

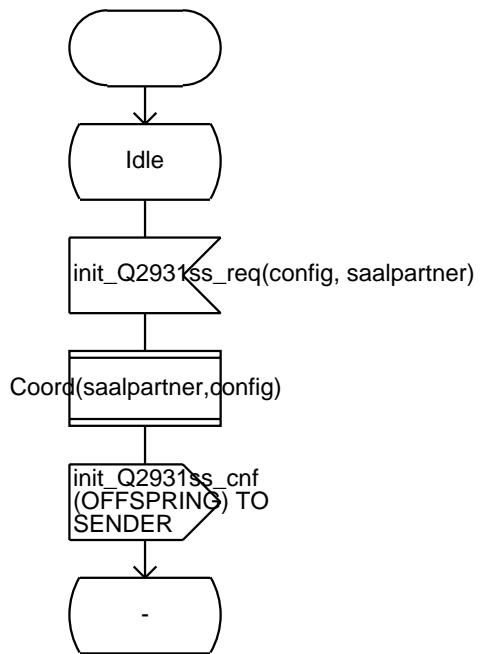
```
DCL saalpartner PId := Null;
```

```
/* local vars */
DCL config InitTrunkConfig;
```

Annex B: CoordControl_PT

Process Type <<Block Type Q2931ss_BT>> CoordControl_PT

2(2)



Annex B: IncomingPartyControl_PT

Process Type <<Block Type Q2931ss_BT>> IncomingPartyControl_PT

1(11)

INHERITS PartyControl_PT;

```
TIMER T399 := 15; /* 15 seconds */
TIMER T397 := 180; /* 3 minutes min */
```

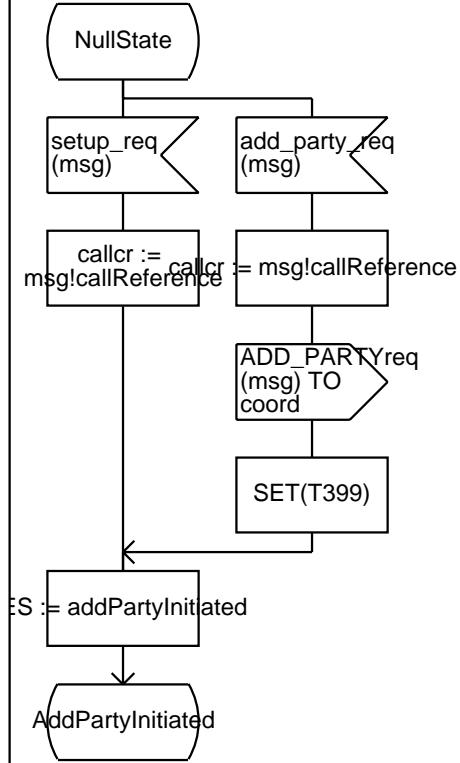
```
DCL tmpcause Cause;
```

Annex B: IncomingPartyControl_PT

Process Type <<Block Type Q2931ss_BT>> IncomingPartyControl_PT

2(11)

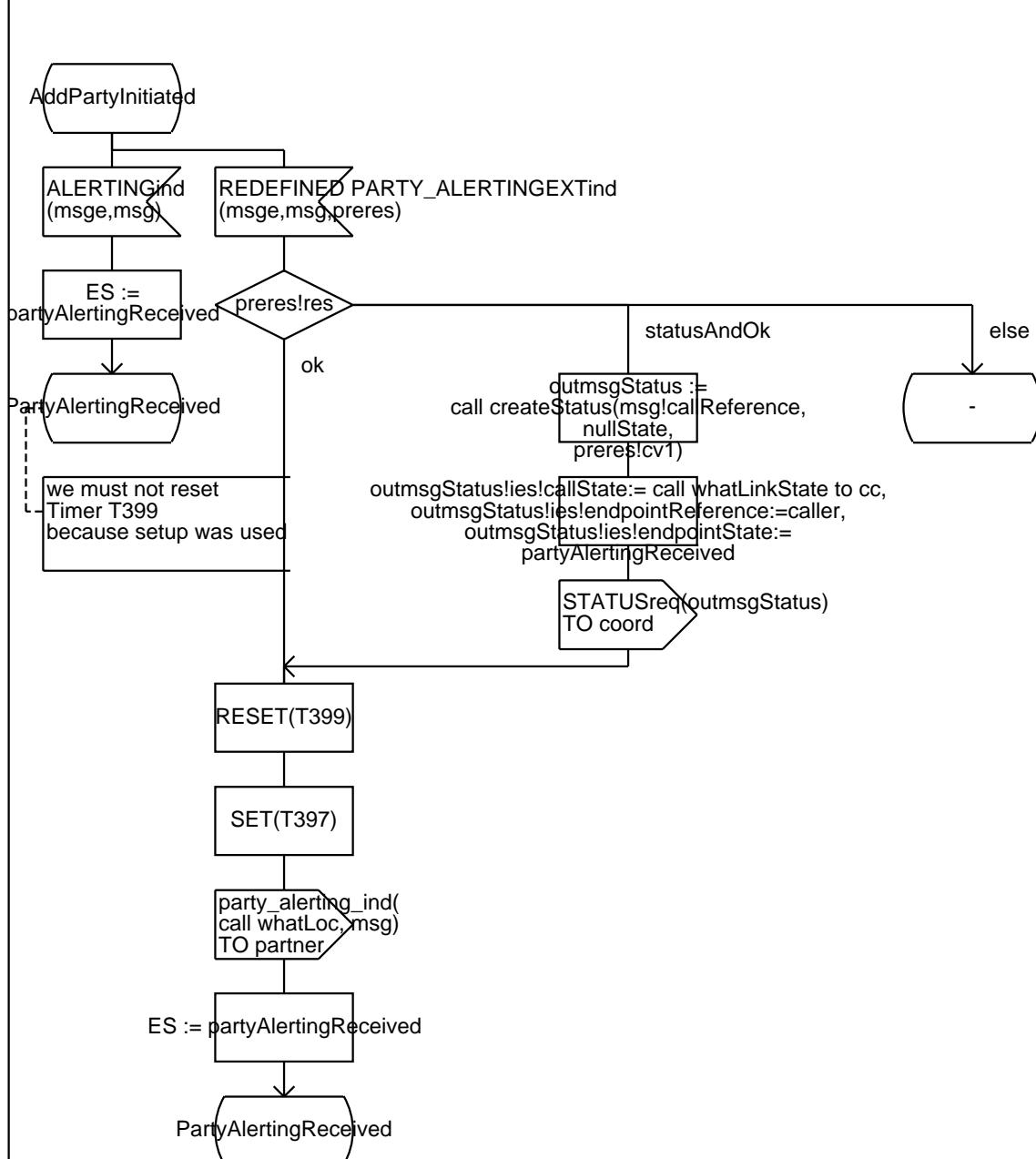
INHERITS PartyControl_PT;



Annex B: IncomingPartyControl_PT

3(11)

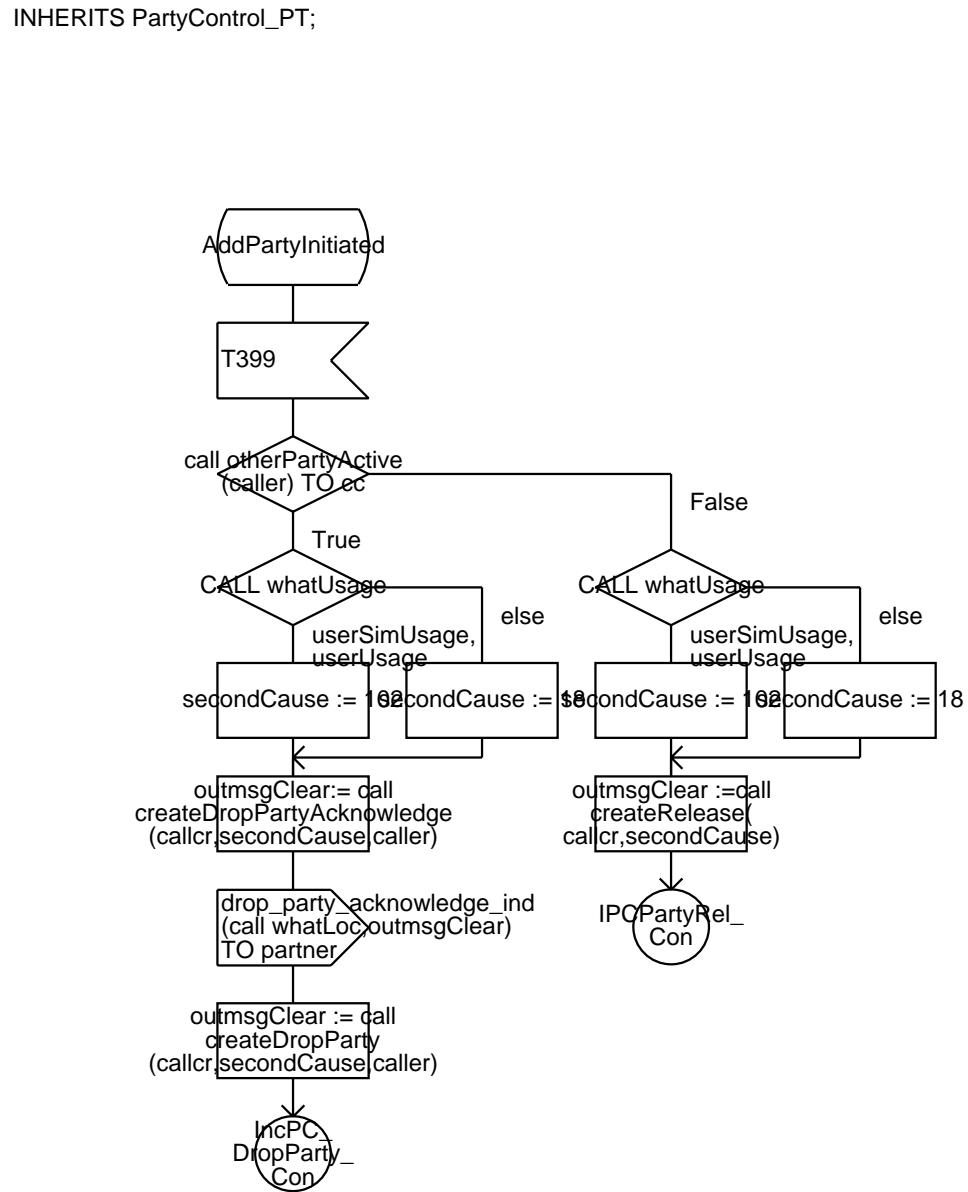
Process Type <<Block Type Q2931ss_BT>> IncomingPartyControl_PT



Annex B: IncomingPartyControl_PT

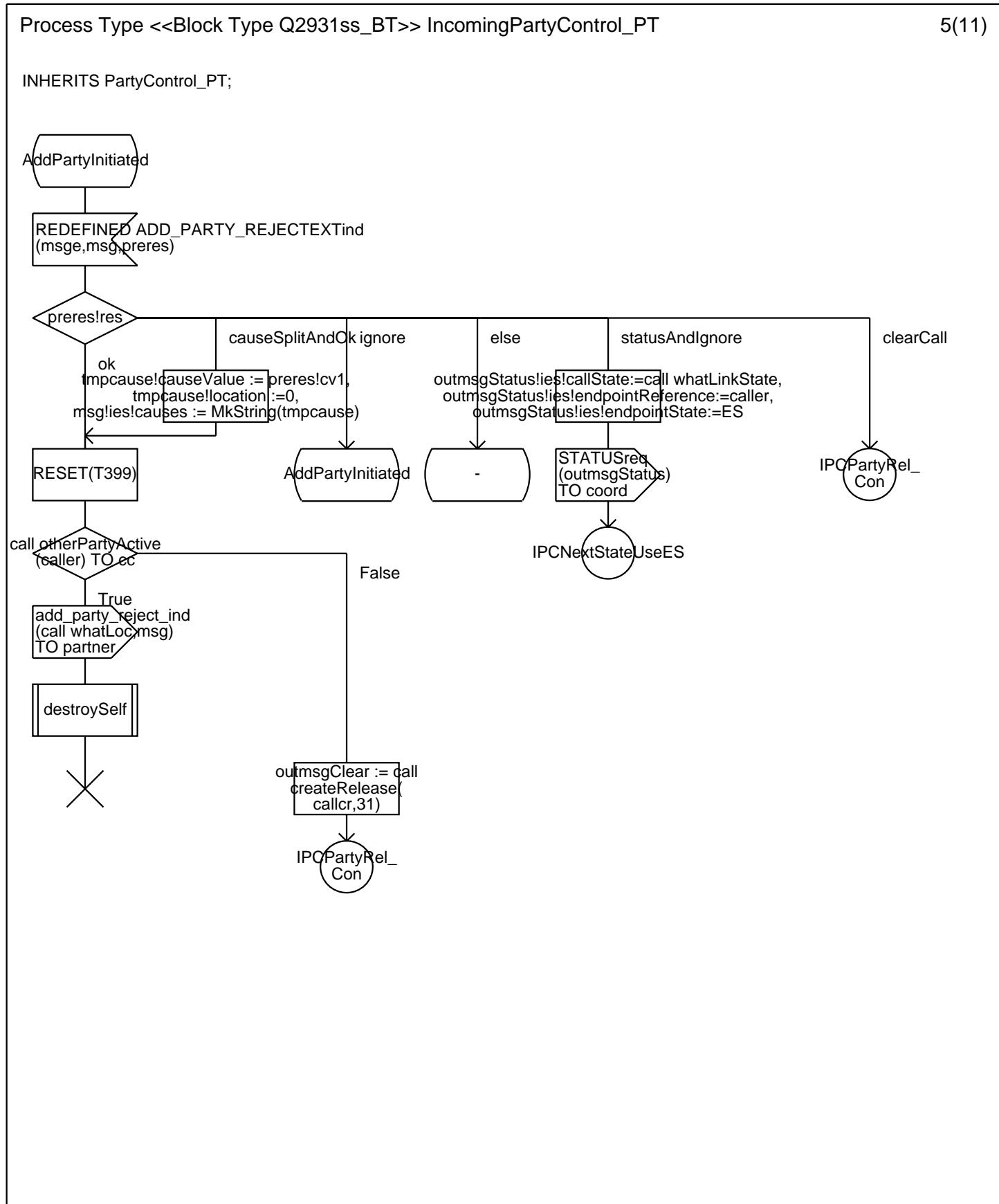
4(11)

Process Type <<Block Type Q2931ss_BT>> IncomingPartyControl_PT



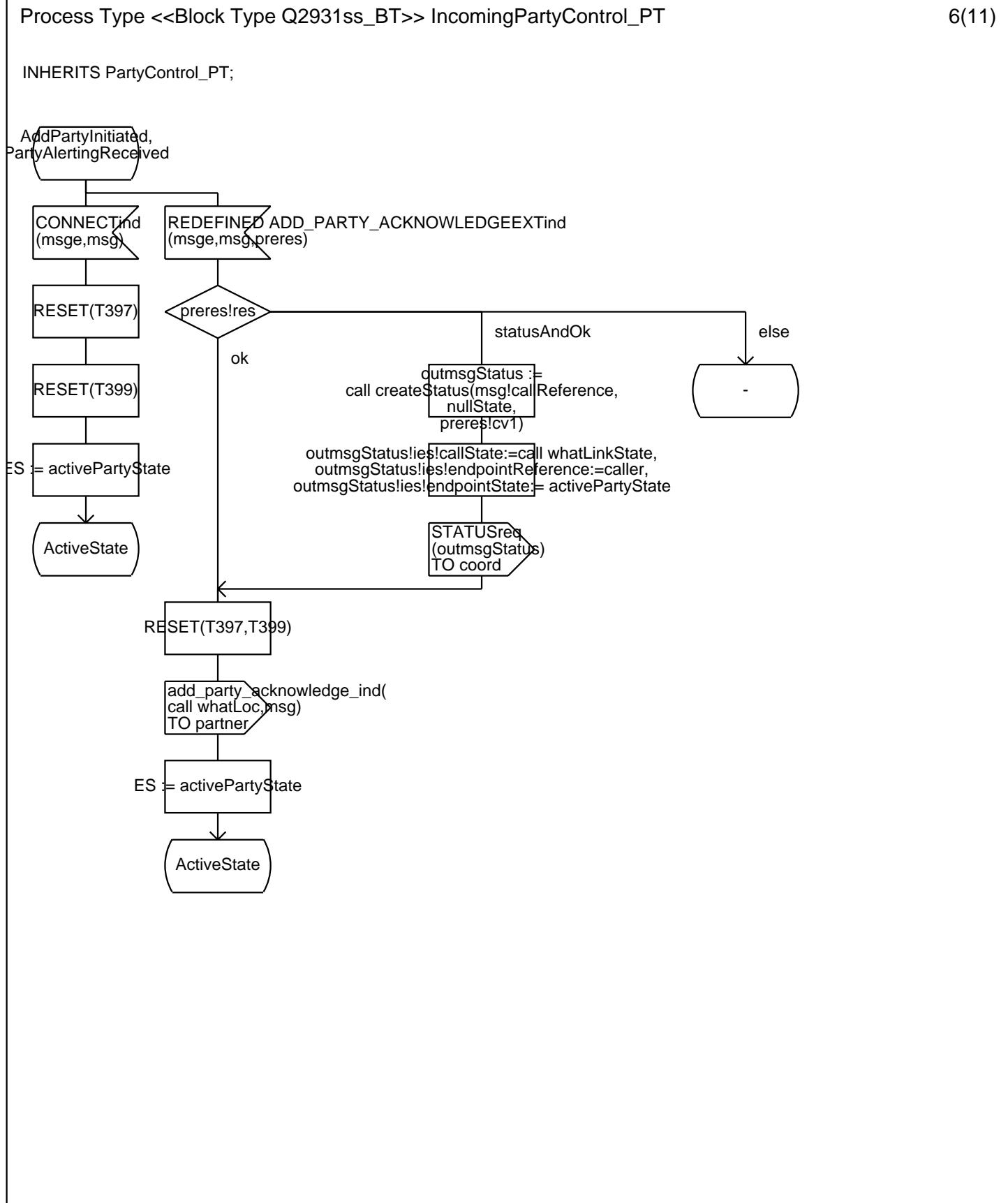
Annex B: IncomingPartyControl_PT

5(11)



Annex B: IncomingPartyControl_PT

6(11)

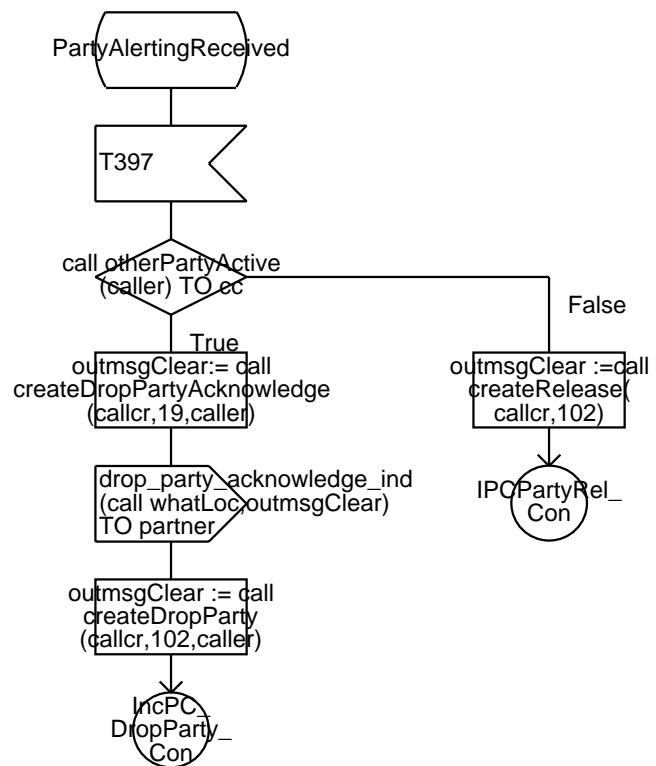


Annex B: IncomingPartyControl_PT

7(11)

Process Type <<Block Type Q2931ss_BT>> IncomingPartyControl_PT

INHERITS PartyControl_PT;

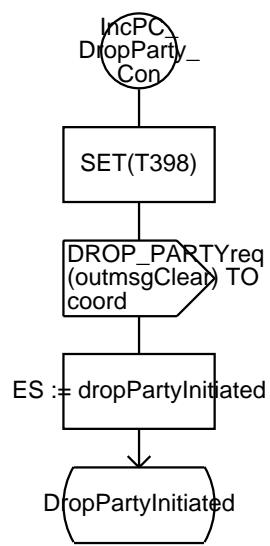


Annex B: IncomingPartyControl_PT

Process Type <<Block Type Q2931ss_BT>> IncomingPartyControl_PT

8(11)

INHERITS PartyControl_PT;

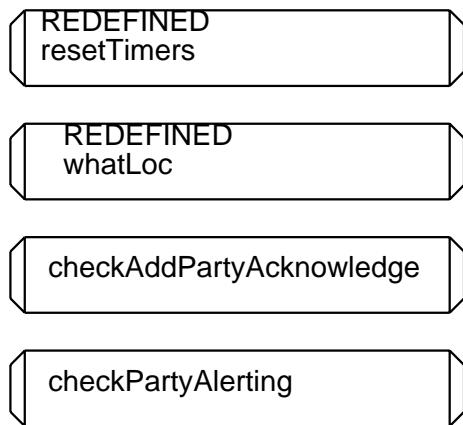


Annex B: IncomingPartyControl_PT

Process Type <<Block Type Q2931ss_BT>> IncomingPartyControl_PT

9(11)

INHERITS PartyControl_PT;

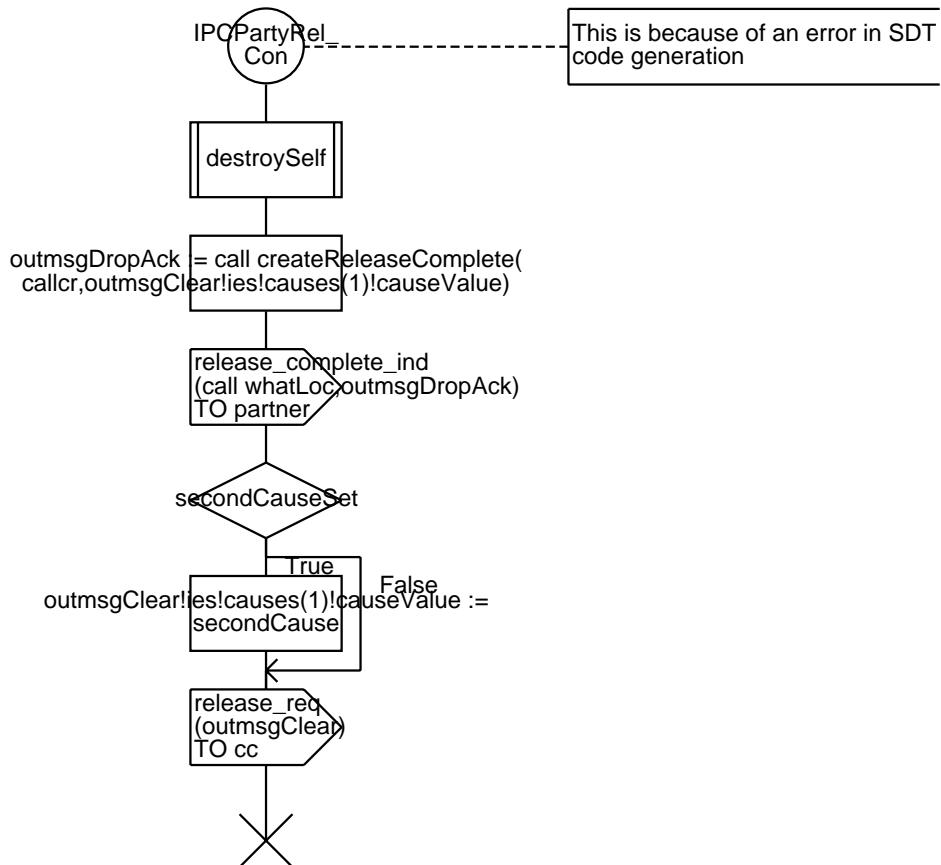


Annex B: IncomingPartyControl_PT

Process Type <<Block Type Q2931ss_BT>> IncomingPartyControl_PT

10(11)

INHERITS PartyControl_PT;

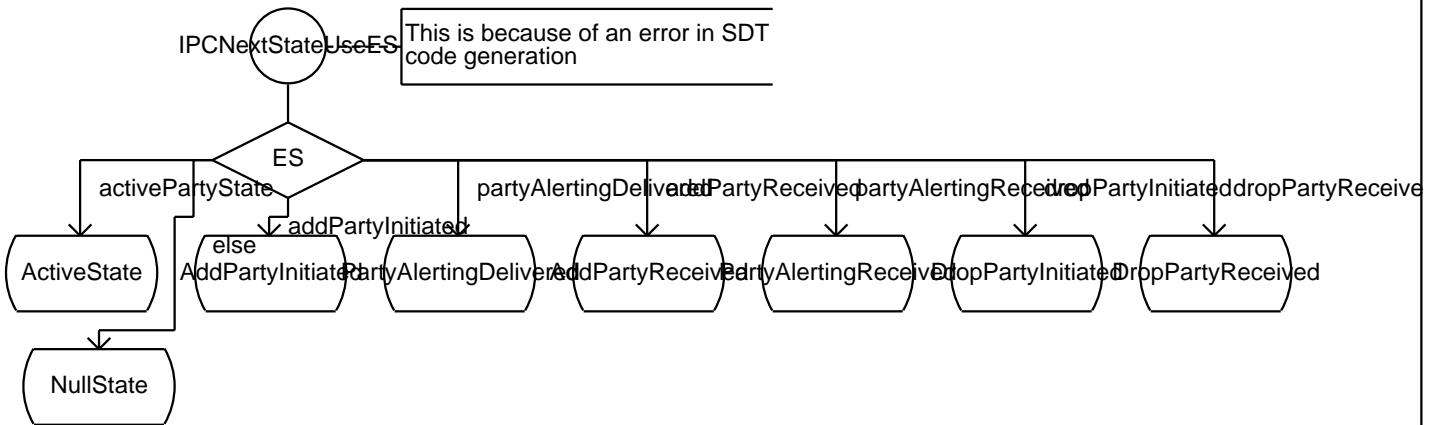


Annex B: IncomingPartyControl_PT

Process Type <<Block Type Q2931ss_BT>> IncomingPartyControl_PT

11(11)

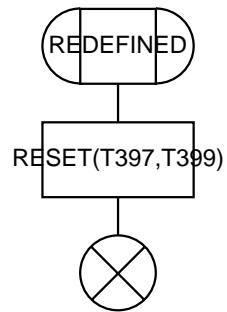
INHERITS PartyControl_PT;



Annex B: resetTimers

Redefined Procedure <<Process Type IncomingPartyControl_PT>> resetTimers

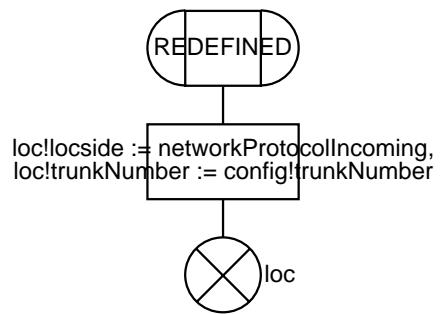
1(1)



Annex B: whatLoc

Redefined Procedure <<Process Type IncomingPartyControl_PT>> whatLoc

1(1)



Annex B: checkAddPartyAcknowledge

Procedure checkAddPartyAcknowledge

1(1)

;RETURNS res CheckResultType;

```
/*#include '../sdl/CheckaddPartyAcknowledge.sdl' */
```

Annex B: checkPartyAlerting

Procedure checkPartyAlerting

1(1)

;RETURNS res CheckResultType;

/*#include '../sdl/CheckpartyAlerting.sdl' */

Annex B: MsgHandler_PT

Process Type <<Block Type Q2931ss_BT>> MsgHandler_PT

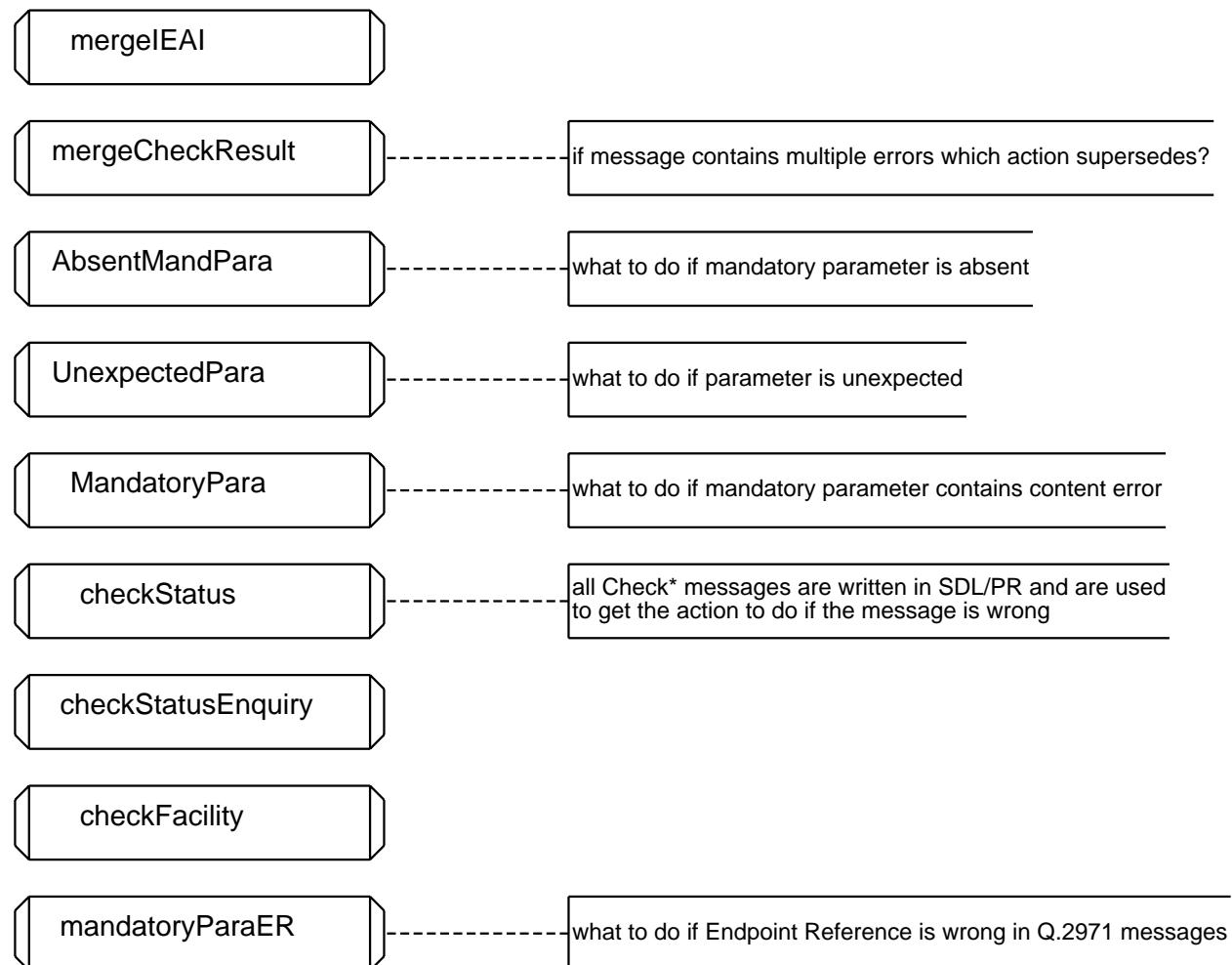
1(2)

```
/* Process type is used to define common procedures.  
The procedures manipulate the variables outmsg* and inmsg and  
secondCauseSet, secondCause.  
Although this is a side effect of the procedures  
the code looks more readable than give the variable  
as return values */  
/* the following variables are set in procedures which check messages.  
outmsgStatus does never contain an endpoint reference when returned - it  
must be provided on CallControl level. */  
  
DCL msg Q2931ssMessage; /* the message received */  
DCL outmsgStatus Q2931ssMessage; /* output of STATUS */  
DCL outmsgClear Q2931ssMessage; /* output of RELEASE or RELEASE_COMPLETE */  
DCL outmsgDropAck Q2931ssMessage;  
/* output of DROP_PARTY_ACKNOWLEDGE, ADD_PARTY_REJECT */  
DCL inmsg Q2931ssMessage; /* corrected msg (fix of cause value) */  
DCL msge MsgError; /* the error of the received message */  
DCL secondCauseSet Boolean := False; /* whether different cause must be transmitted when  
the message is answered at same trunk as the message came in */  
DCL secondCause CauseValue; /* the cause value to set when secondCauseSet is true */  
DCL noER Boolean := True; /* Is EndpointReference contained in Message? */  
DCL firstERMessage Boolean := True; /* is message first message to SETUP */  
DCL firstMsgNoER Boolean := False; /* has first message to SETUP no ER */  
DCL gfpinv GFPInvokeMessage;  
DCL gfprej GFPRejectMessage;  
DCL gperr GFPErrorMessage;  
DCL gfpres GFPResultMessage;
```

Annex B: MsgHandler_PT

Process Type <<Block Type Q2931ss_BT>> MsgHandler_PT

2(2)

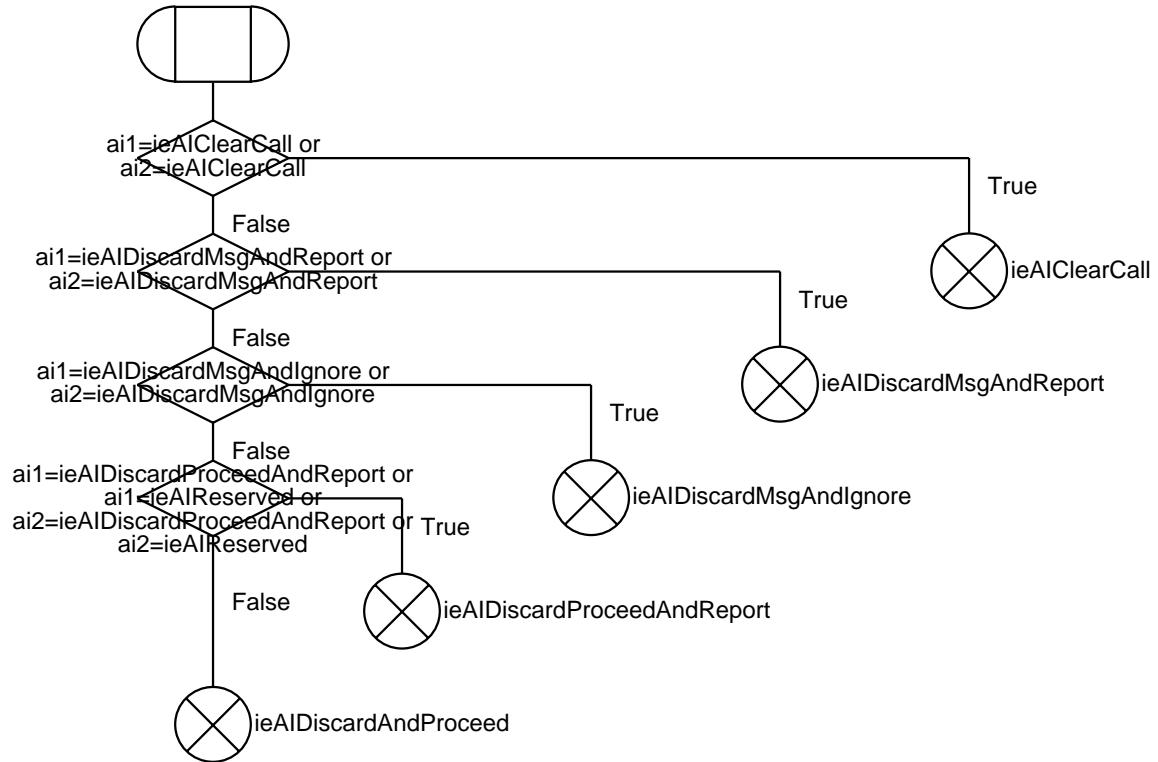


Annex B: mergeIEAI

Procedure mergeIEAI

1(1)

;fpar ai1 IEActionIndicator, ai2 IEActionIndicator;
returns IEActionIndicator;

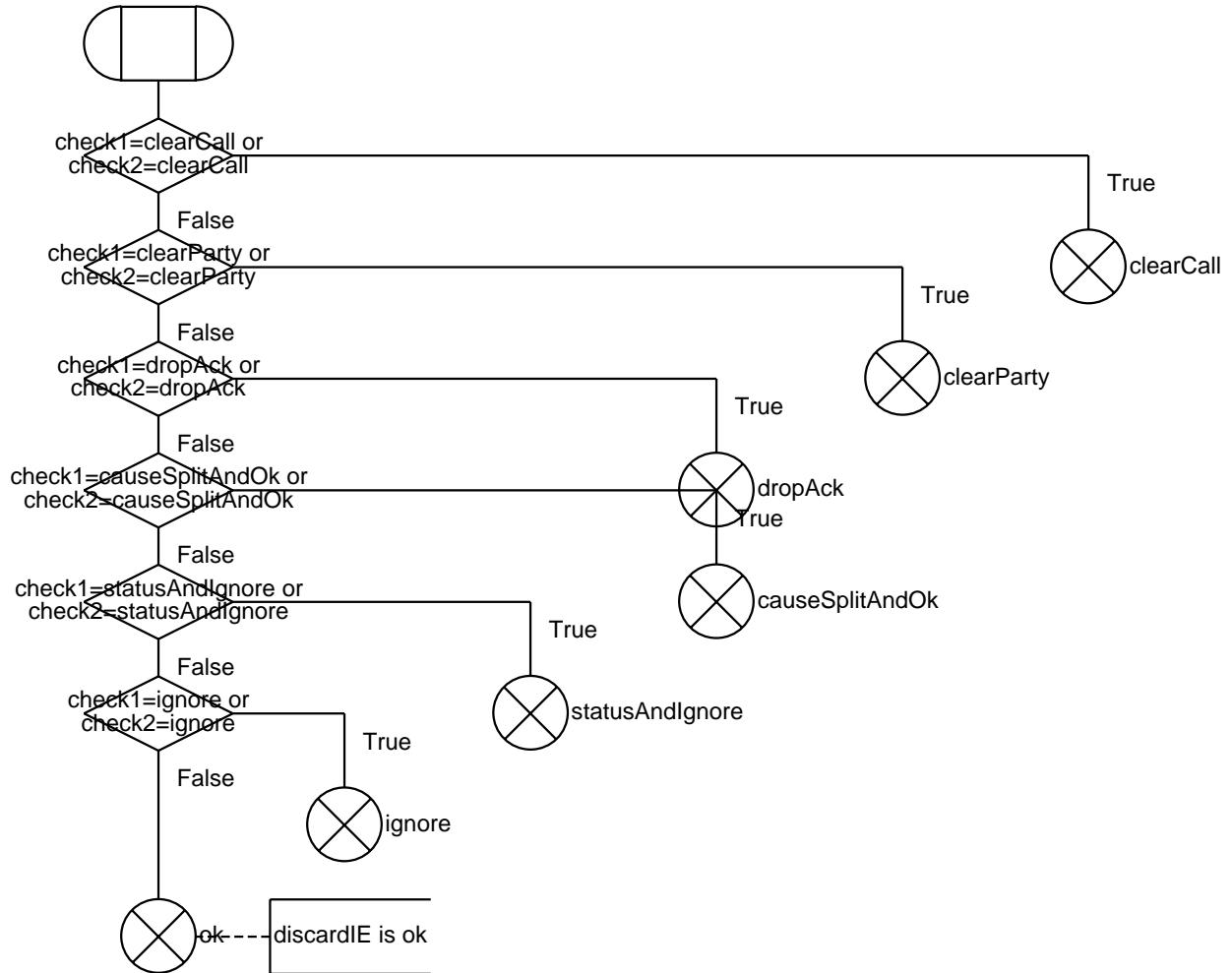


Annex B: mergeCheckResult

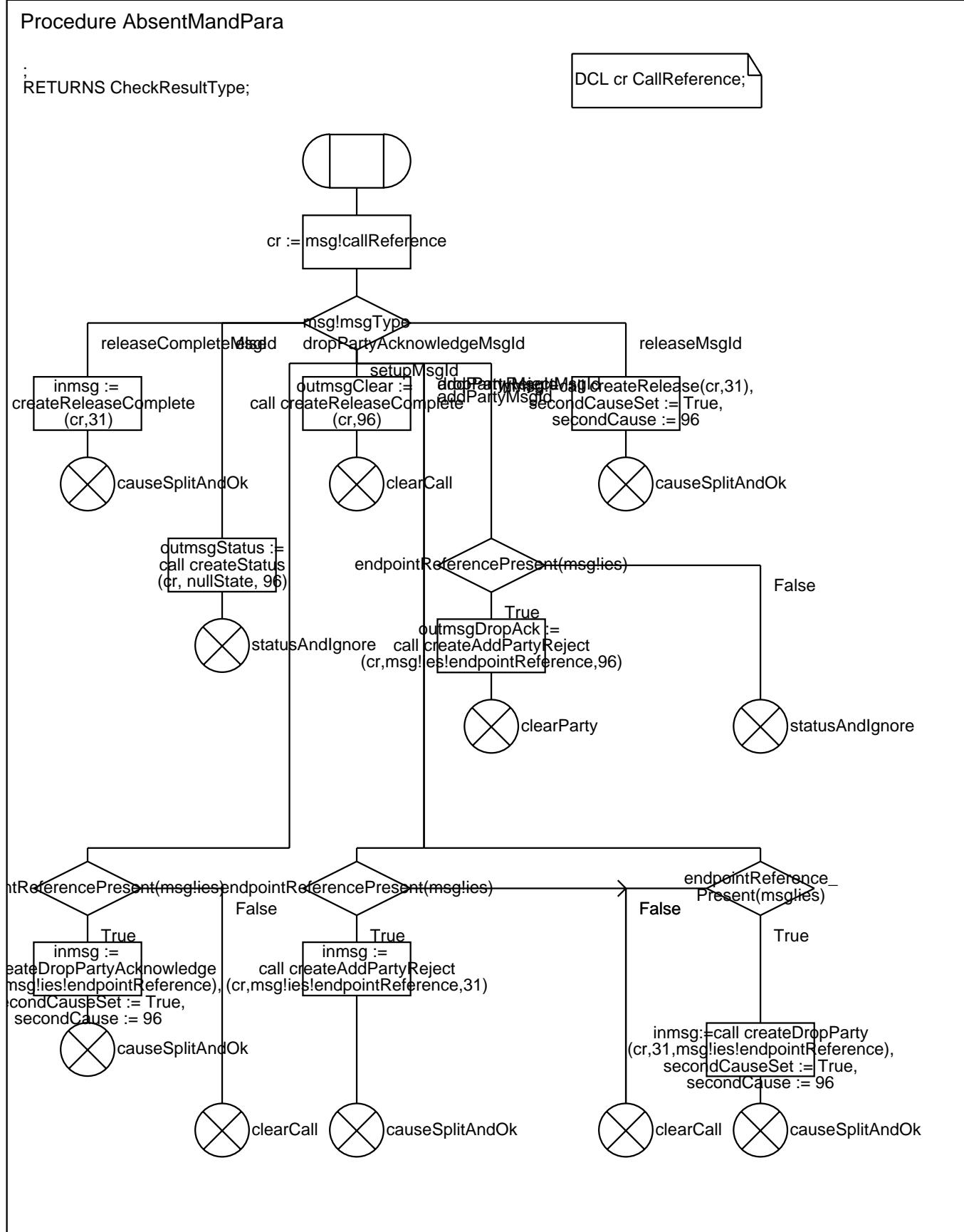
Procedure <<Process Type MsgHandler_PT>> mergeCheckResult

1(1)

```
;fpar check1 CheckResultType, check2 CheckResultType;
returns CheckResultType;
```



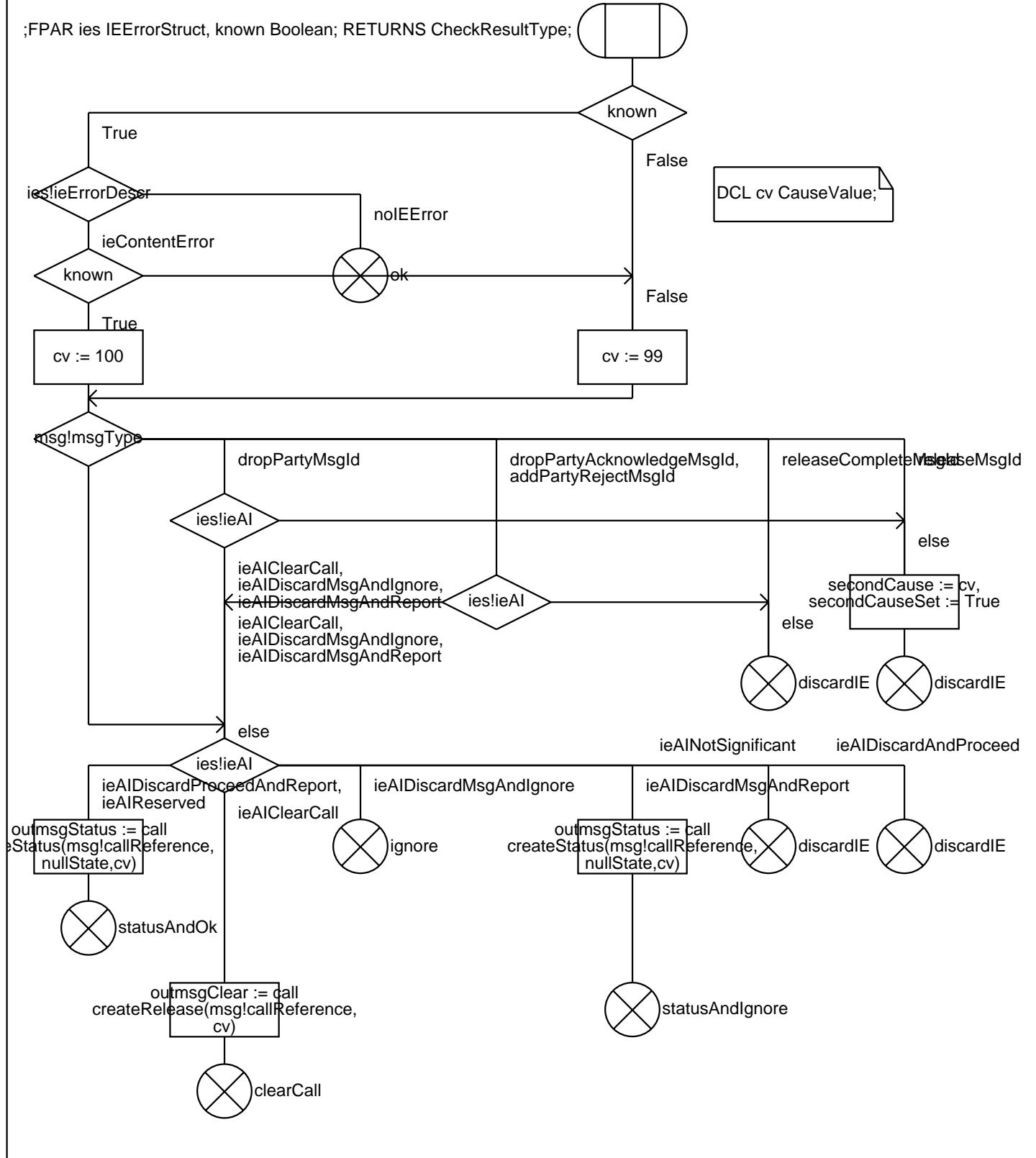
Annex B: AbsentMandPara



Annex B: UnexpectedPara

Procedure <<Process Type MsgHandler_PT>> UnexpectedPara

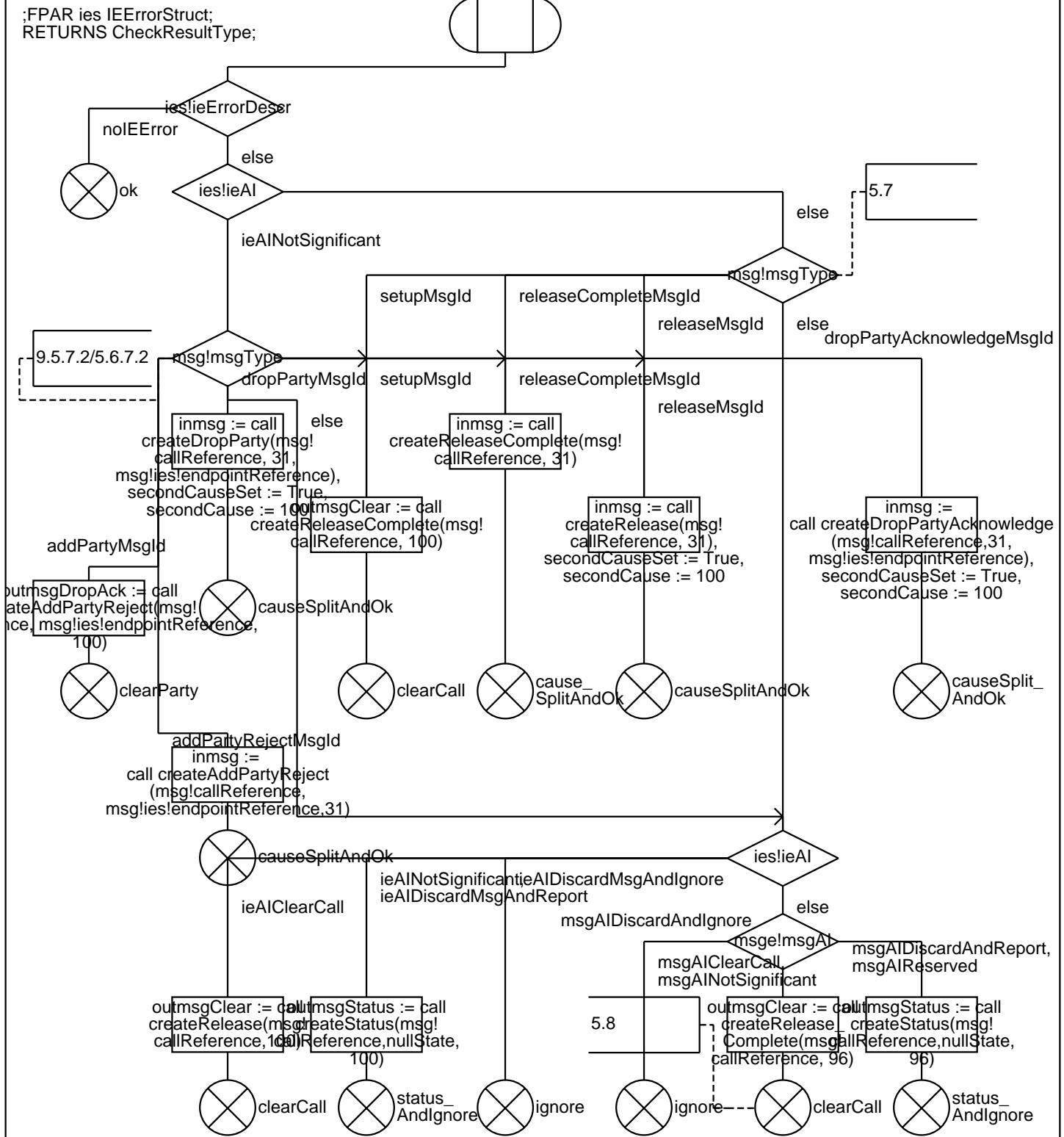
1(1)



Annex B: MandatoryPara

Procedure MandatoryPara

1(1)



Annex B: checkStatus

Procedure checkStatus

1(1)

;RETURNS res CheckResultType;

```
/*#include '../sdl/Checkstatus.sdl' */
```

Annex B: checkStatusEnquiry

Procedure checkStatusEnquiry

1(1)

;RETURNS res CheckResultType;

```
/*#include '../sdl/CheckstatusEnquiry.sdl' */
```

Annex B: checkFacility

Procedure <<Process Type MsgHandler_PT>> checkFacility

1(1)

;RETURNS res CheckResultType;

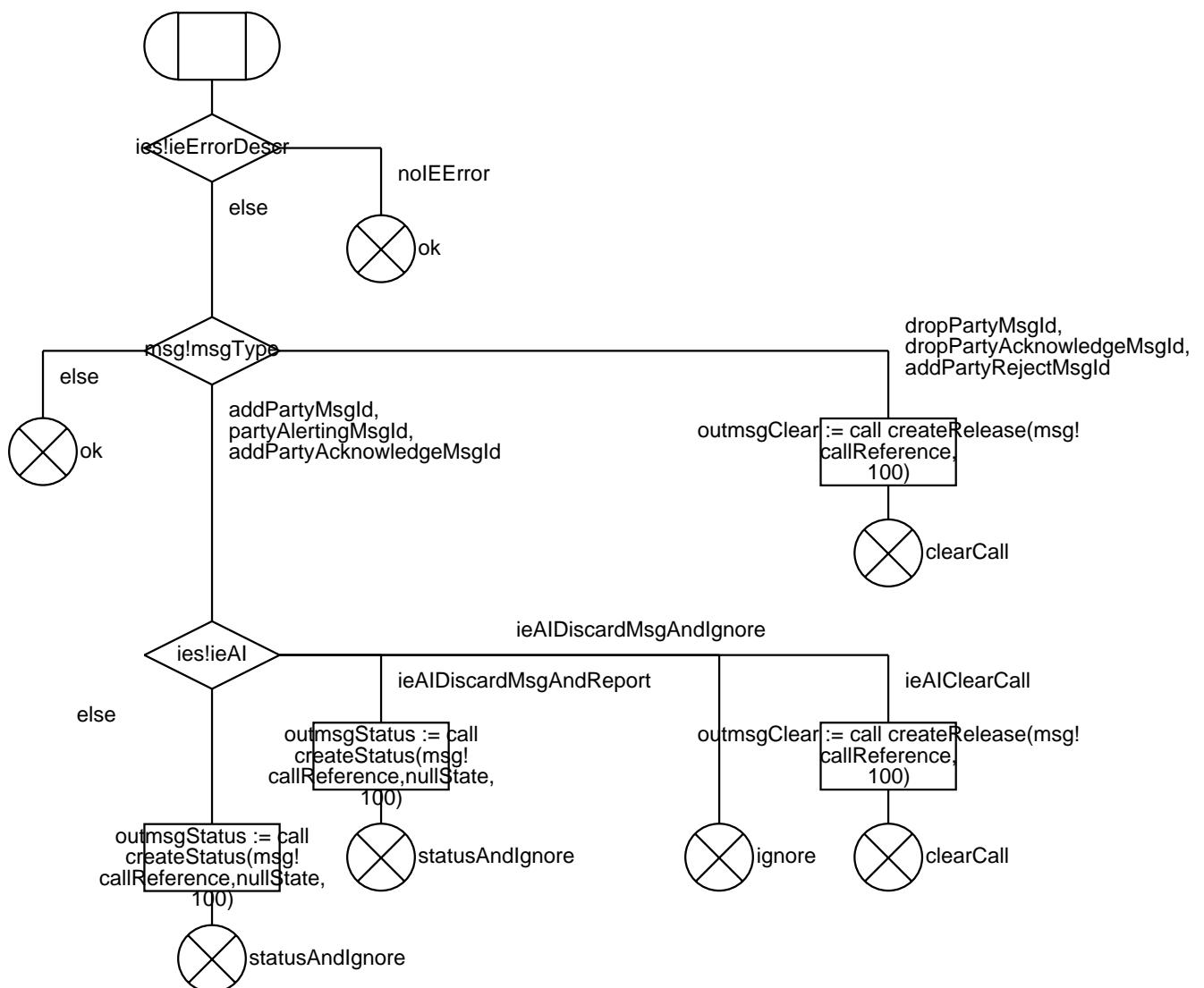
```
/*#include '../sdl/Checkfacility.sdl' */
```

Annex B: MandatoryParaER

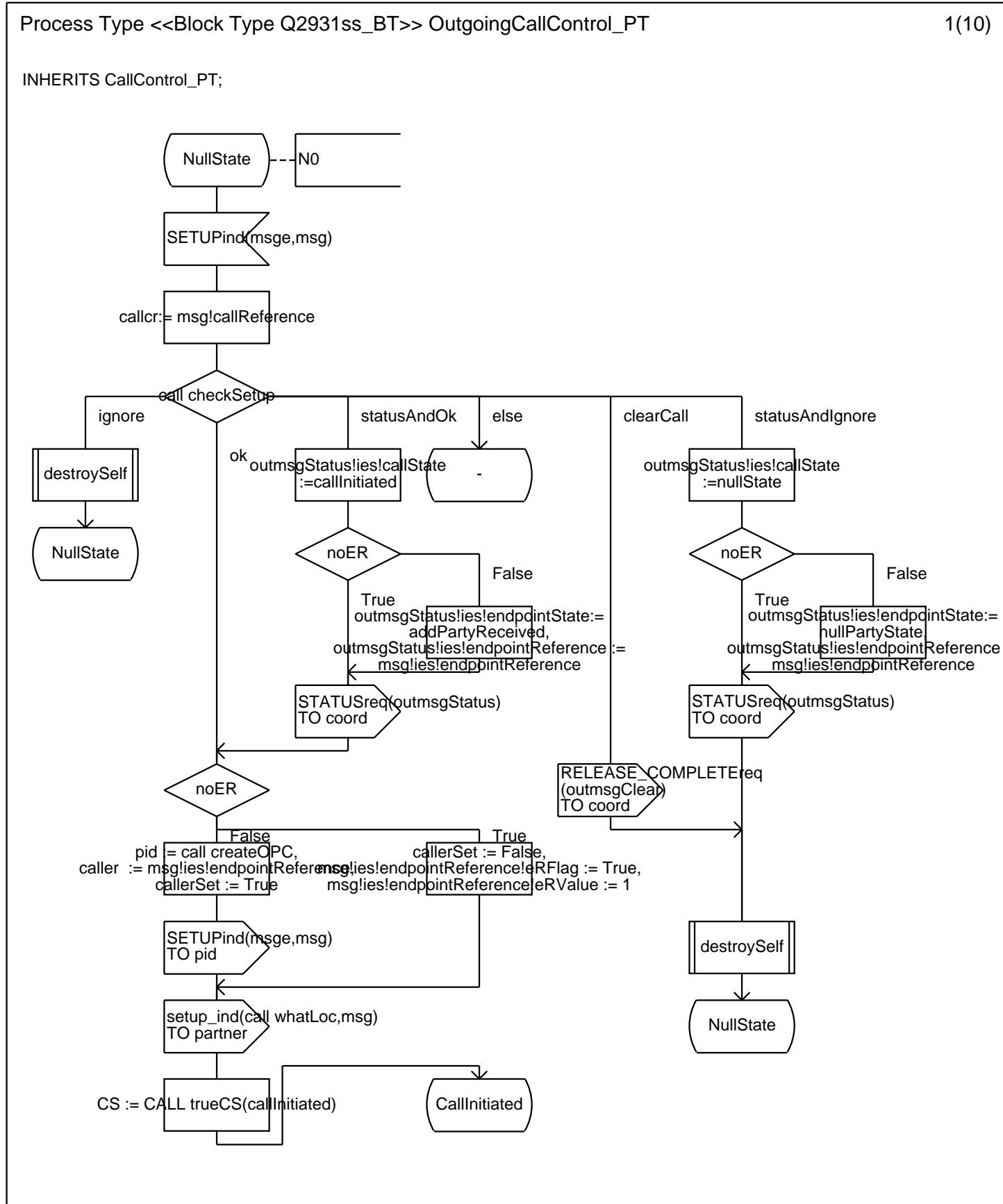
Procedure MandatoryParaER

1(1)

```
:FPAR ies IEEErrorStruct;
RETURNS CheckResultType;
```



Annex B: OutgoingCallControl_PT

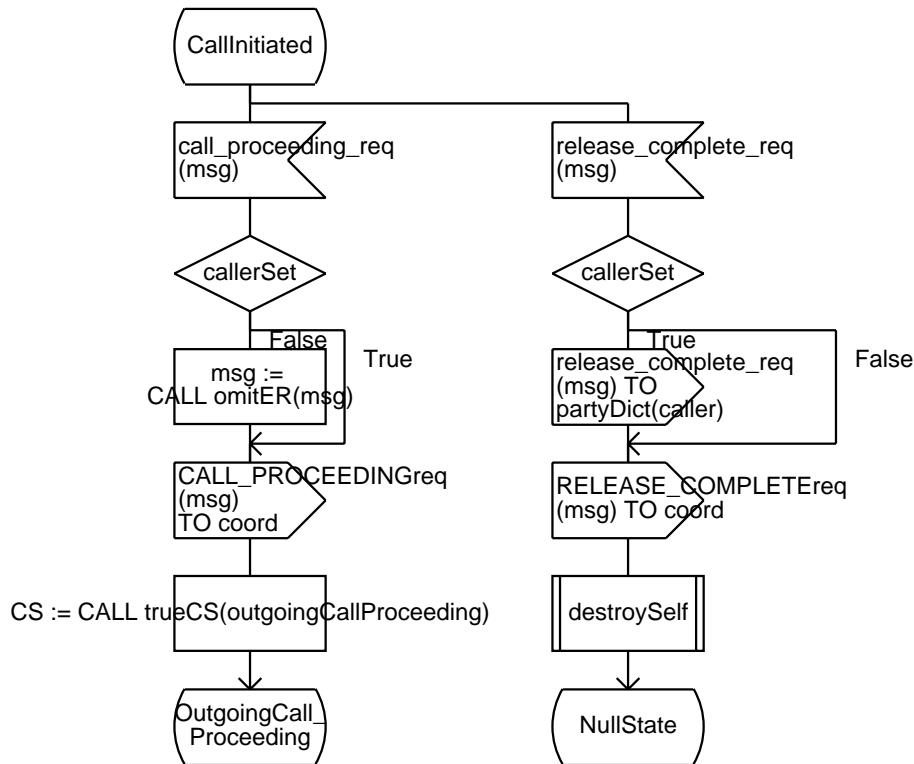


Annex B: OutgoingCallControl_PT

Process Type <<Block Type Q2931ss_BT>> OutgoingCallControl_PT

2(10)

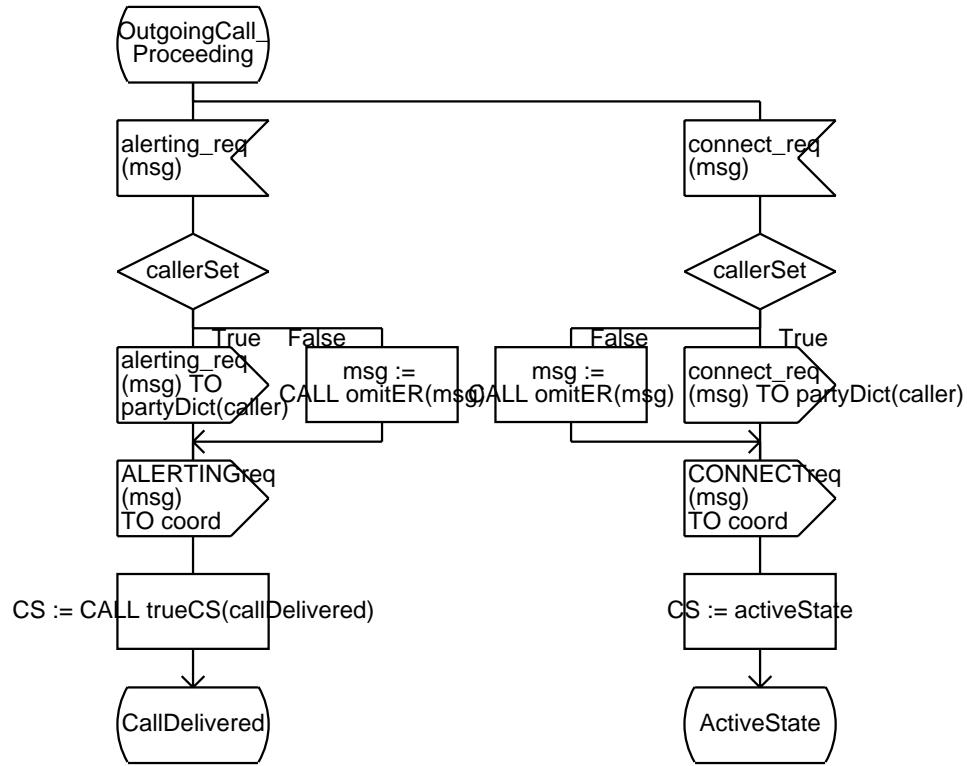
INHERITS CallControl_PT;



Annex B: OutgoingCallControl_PT

3(10)

INHERITS CallControl_PT;

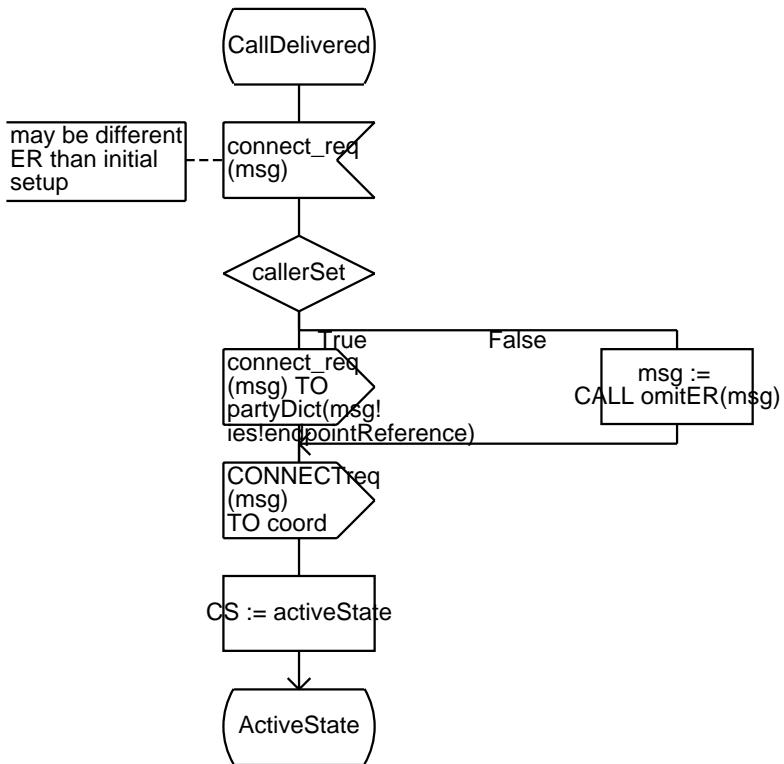


Annex B: OutgoingCallControl_PT

Process Type <<Block Type Q2931ss_BT>> OutgoingCallControl_PT

4(10)

INHERITS CallControl_PT;

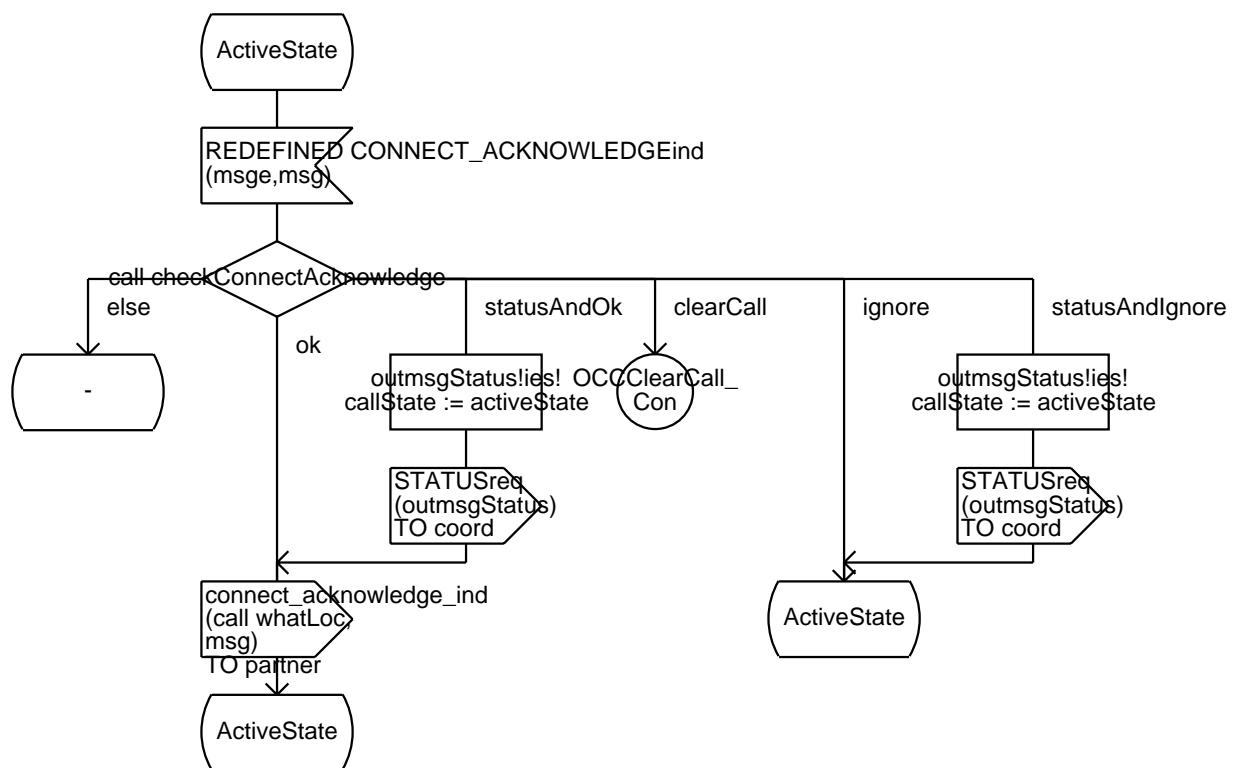


Annex B: OutgoingCallControl_PT

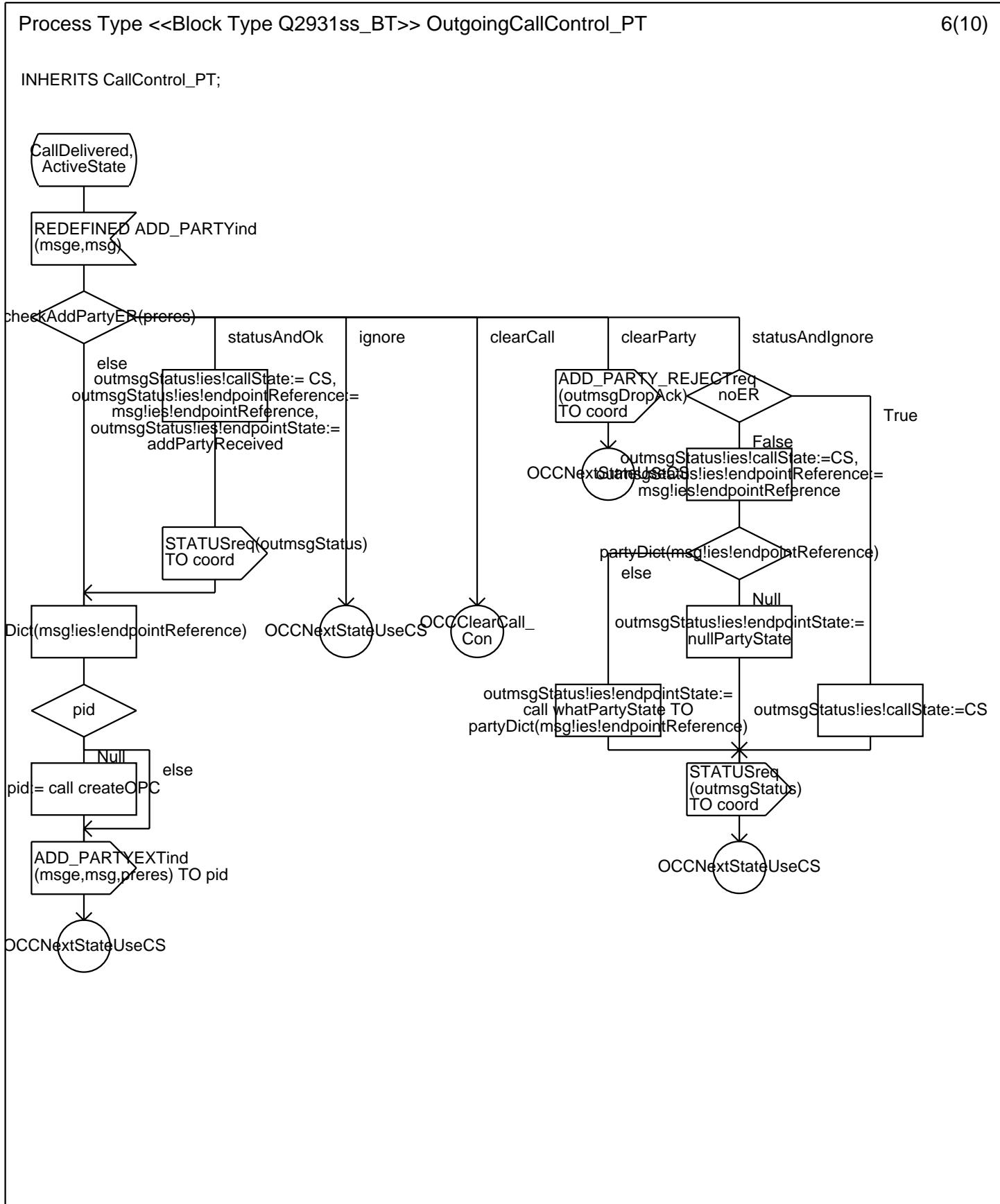
Process Type <<Block Type Q2931ss_BT>> OutgoingCallControl_PT

5(10)

INHERITS CallControl_PT;



Annex B: OutgoingCallControl_PT

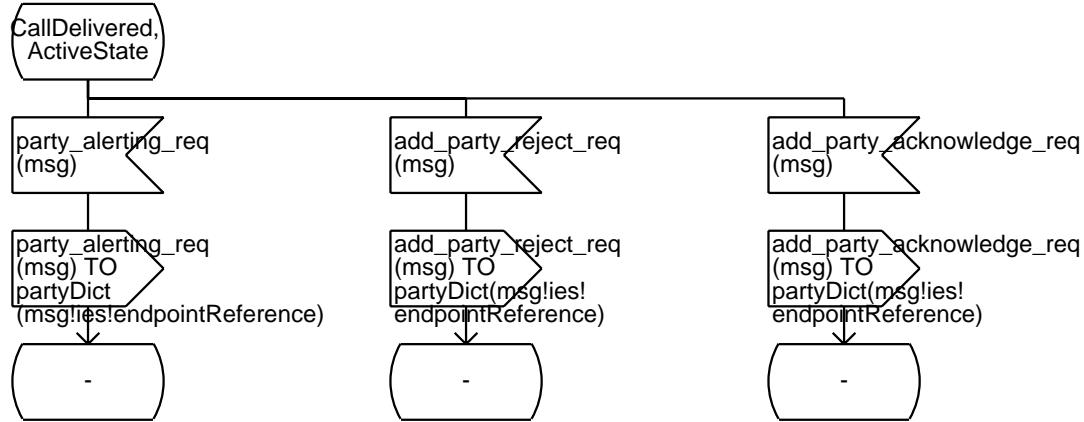


Annex B: OutgoingCallControl_PT

Process Type <<Block Type Q2931ss_BT>> OutgoingCallControl_PT

7(10)

INHERITS CallControl_PT;

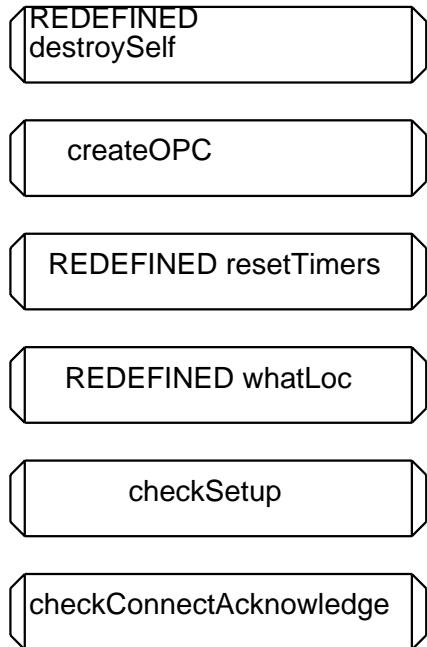


Annex B: OutgoingCallControl_PT

Process Type <<Block Type Q2931ss_BT>> OutgoingCallControl_PT

8(10)

INHERITS CallControl_PT;

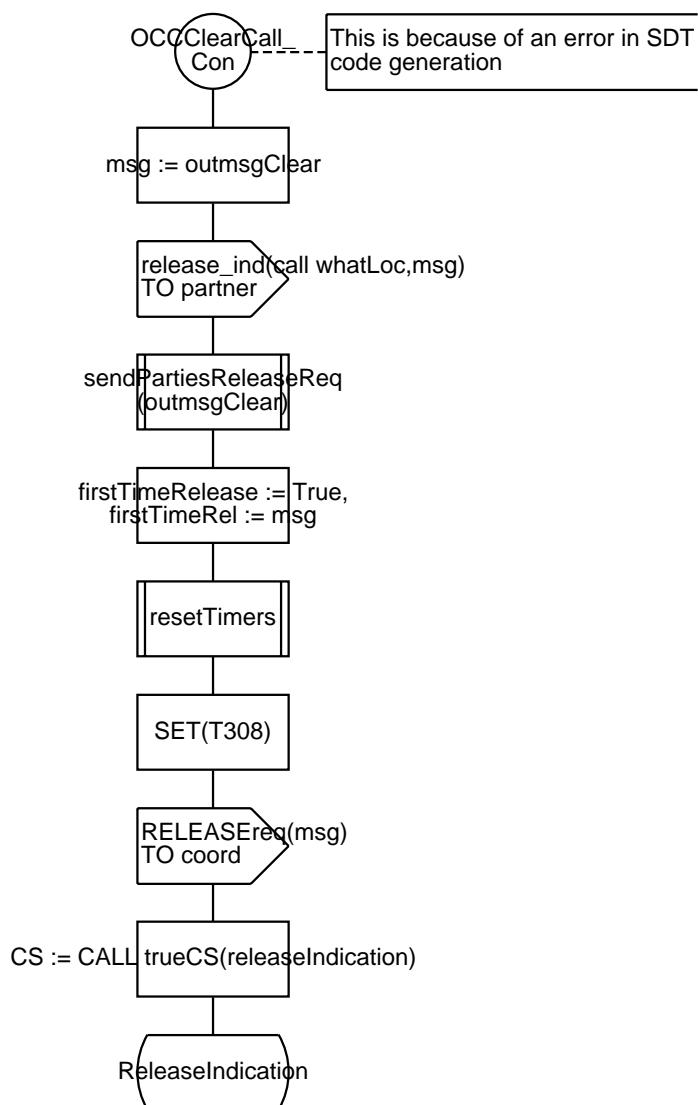


Annex B: OutgoingCallControl_PT

Process Type <<Block Type Q2931ss_BT>> OutgoingCallControl_PT

9(10)

INHERITS CallControl_PT;

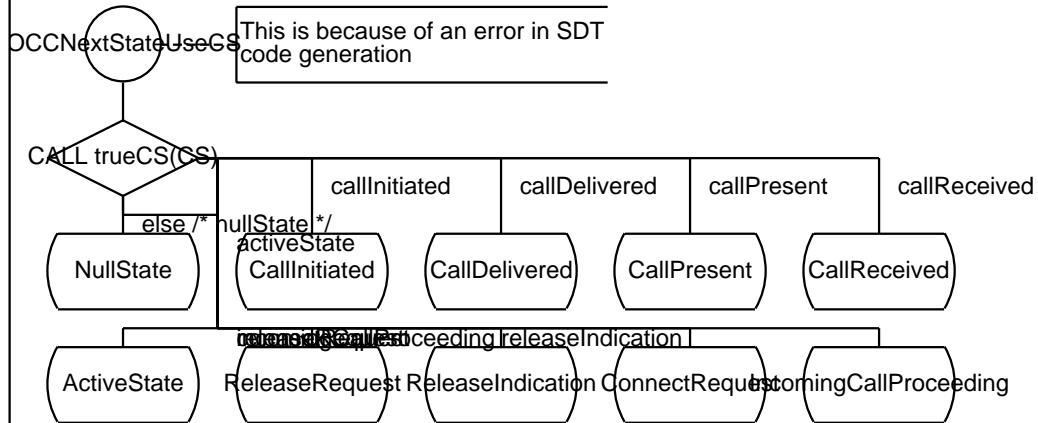


Annex B: OutgoingCallControl_PT

Process Type <<Block Type Q2931ss_BT>> OutgoingCallControl_PT

10(10)

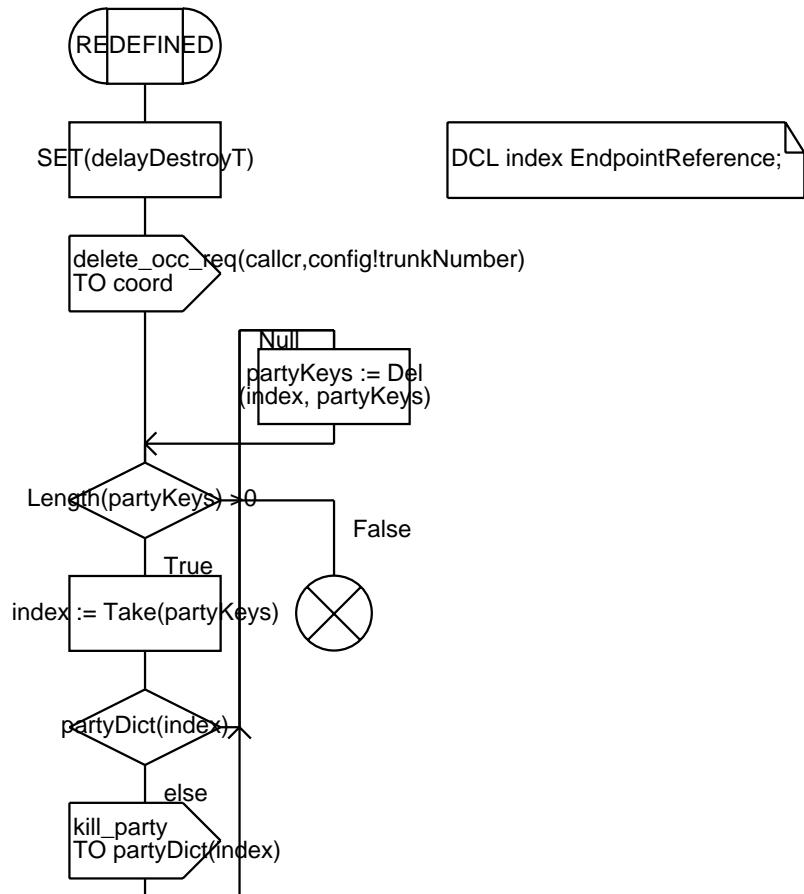
INHERITS CallControl_PT;



Annex B: destroySelf

Redefined Procedure <<Process Type OutgoingCallControl_PT>> destroySelf

1(1)

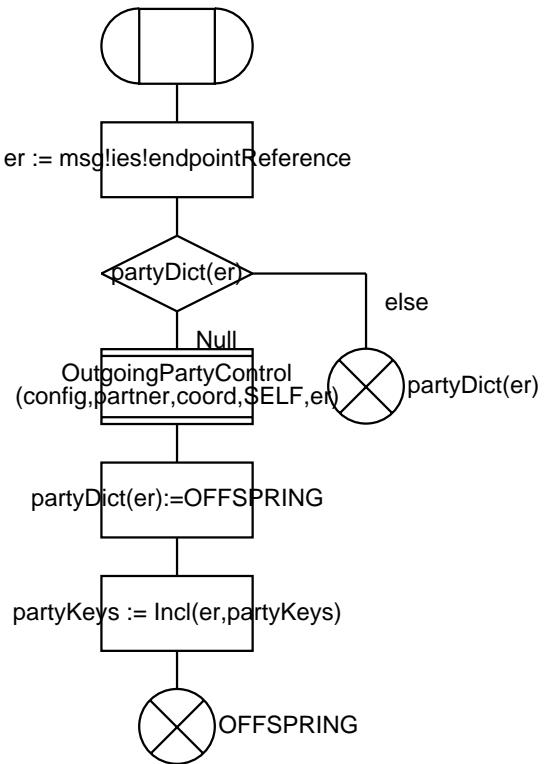


Annex B: createOPC

Procedure createOPC

1(1)

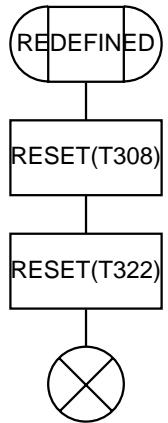
;RETURNS PId;



Annex B: resetTimers

Redefined Procedure <<Process Type OutgoingCallControl_PT>> resetTimers

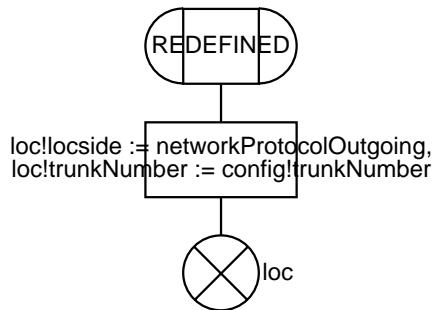
1(1)



Annex B: whatLoc

Redefined Procedure <<Process Type OutgoingCallControl_PT>> whatLoc

1(1)



Annex B: checkSetup

Procedure checkSetup

1(1)

;RETURNS res CheckResultType;

```
/*#include '../sdl/Checksetup.sdl' */
```

Annex B: checkConnectAcknowledge

Procedure checkConnectAcknowledge

1(1)

;RETURNS res CheckResultType;

/*#include '../sdl/CheckconnectAcknowledge.sdl' */

Annex B: PartyControl_PT

Process Type <<Block Type Q2931ss_BT>> PartyControl_PT

1(20)

```
INHERITS MsgHandler_PT;
FPAR config InitTrunkConfig,partner PId, coord PId, cc PId, caller EndpointReference;
```

```
DCL ES EndpointState; /* current party state */
DCL callcr CallReference; /* my call reference */
DCL dropmsg Q2931ssMessage;
```

```
/*
trunknr is my Trunk,
partner is application,
coord is pid of coordination process,
cc is call control process id,
myer is my Endpoint Reference
*/
```

```
TIMER T398 := 4; /* 4 seconds */
```

Annex B: PartyControl_PT

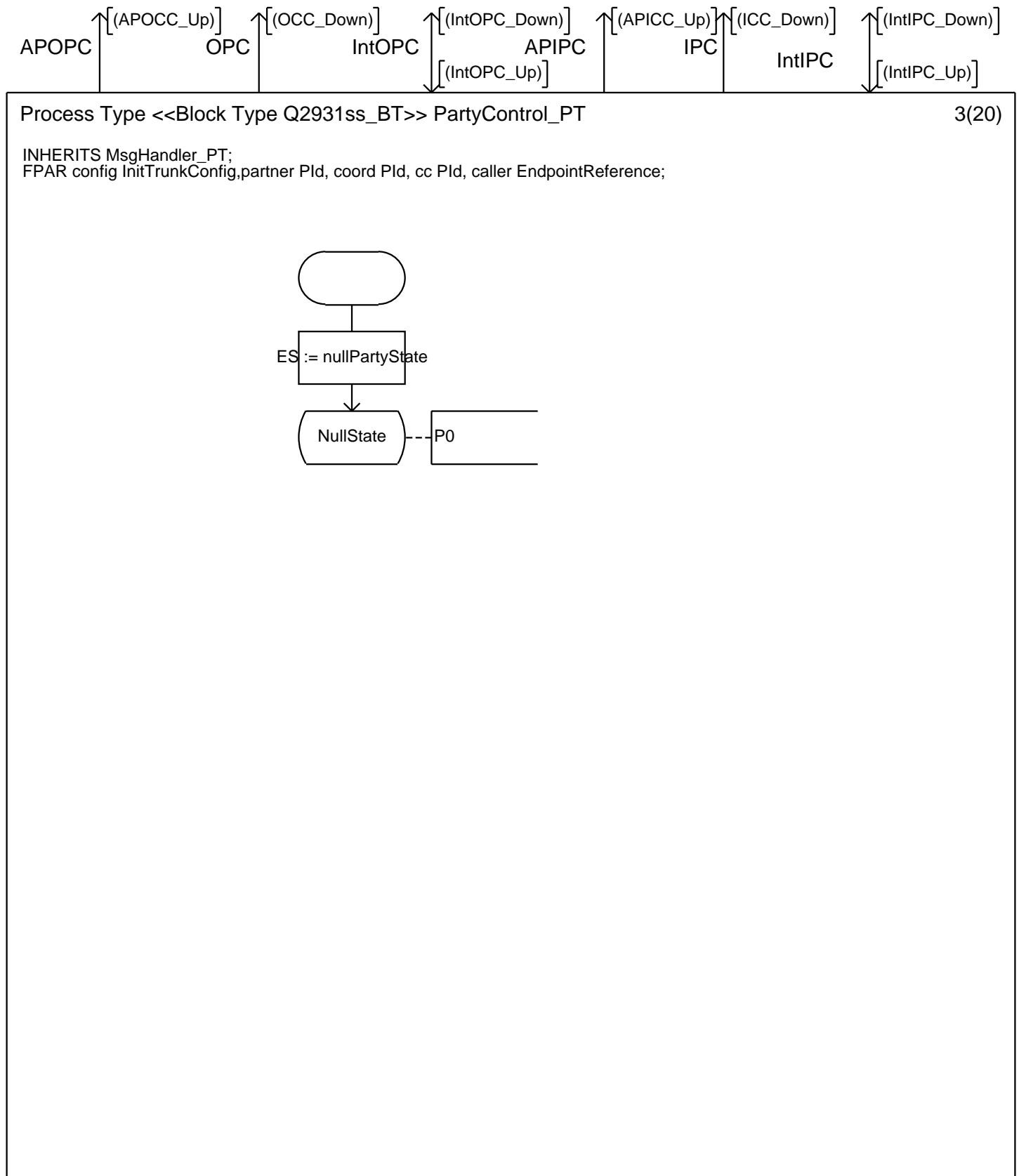
Process Type <>Block Type Q2931ss_BT>> PartyControl_PT

2(20)

```
INHERITS MsgHandler_PT;
FPAR config InitTrunkConfig,partner PId, coord PId, cc PId, caller EndpointReference;
```

```
/* Variable valid only during one transition */
DCL sendStatus Boolean := False;
DCL relesender PId;
DCL preres CheckResultTypeStatus;
DCL tmpcs CallState;
```

Annex B: PartyControl_PT

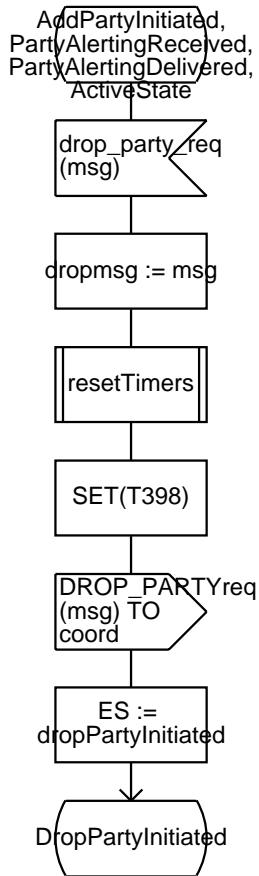


Annex B: PartyControl_PT

Process Type <<Block Type Q2931ss_BT>> PartyControl_PT

4(20)

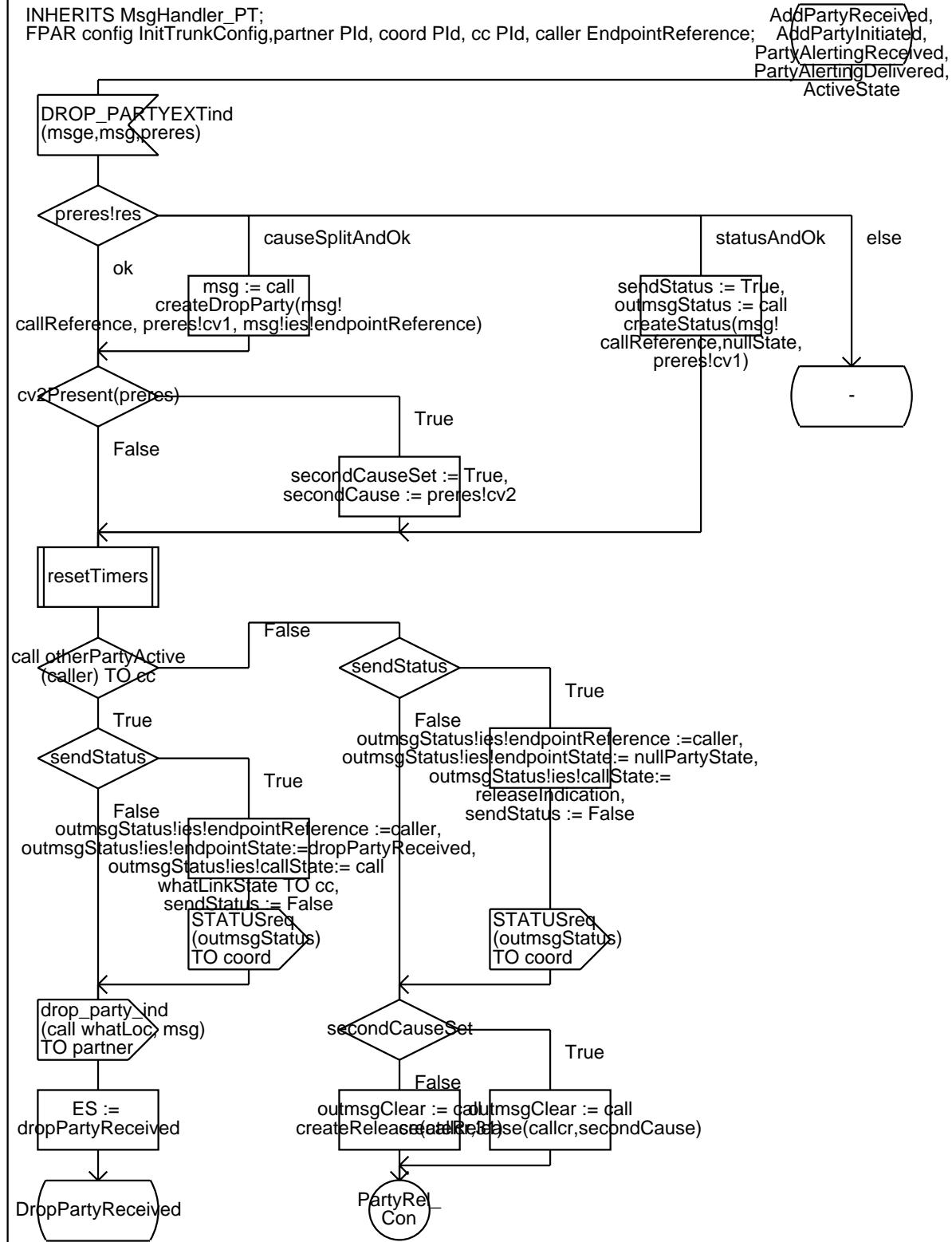
INHERITS MsgHandler_PT;
 FPAR config InitTrunkConfig,partner PId, coord PId, cc PId, caller EndpointReference;



Annex B: PartyControl_PT

Process Type <<Block Type Q2931ss_BT>> PartyControl_PT

5(20)

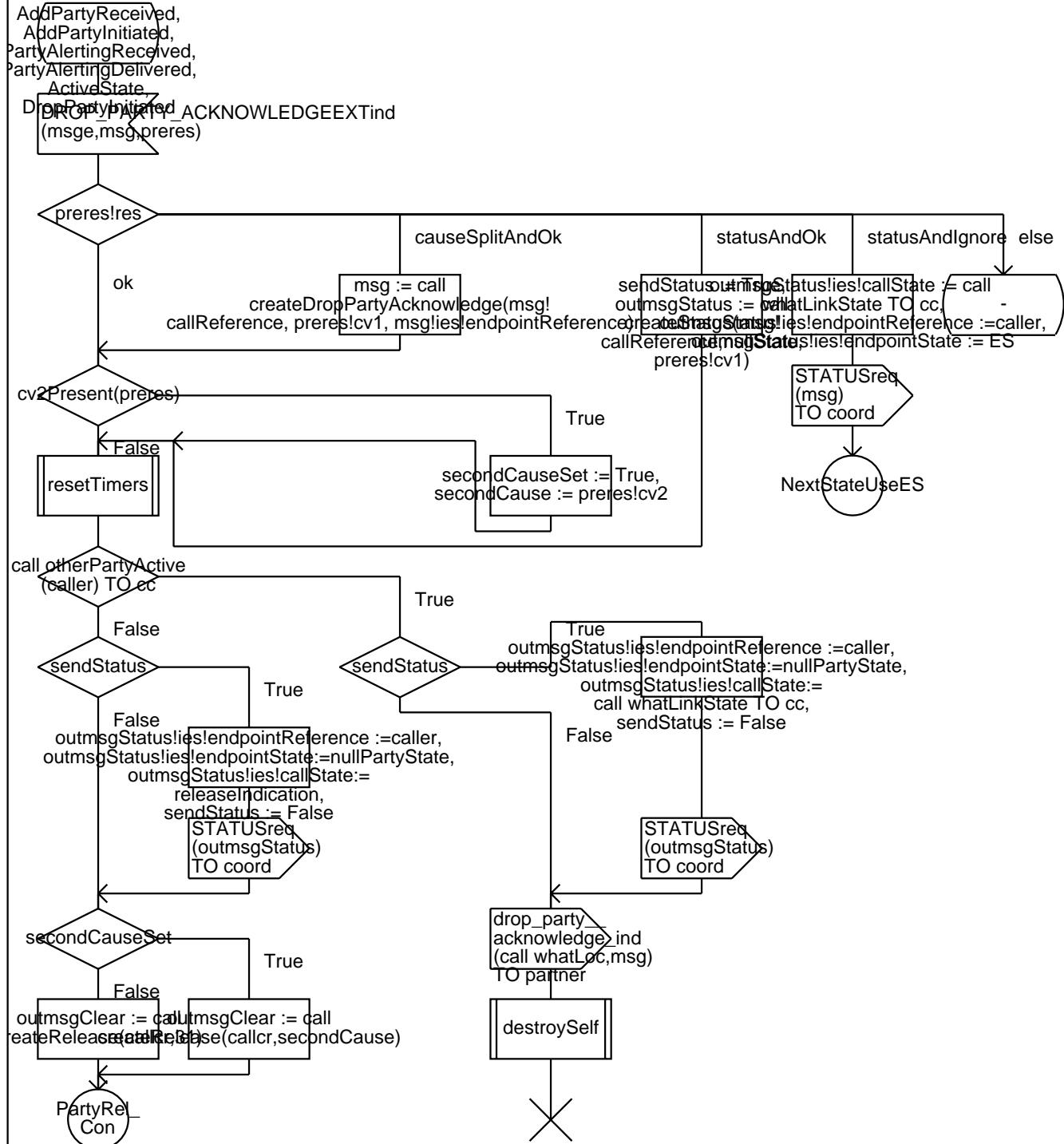


Annex B: PartyControl_PT

Process Type <<Block Type Q2931ss_BT>> PartyControl_PT

6(20)

INHERITS MsgHandler_PT;
 FPAR config InitTrunkConfig,partner PId, coord PId, cc PId, caller EndpointReference;

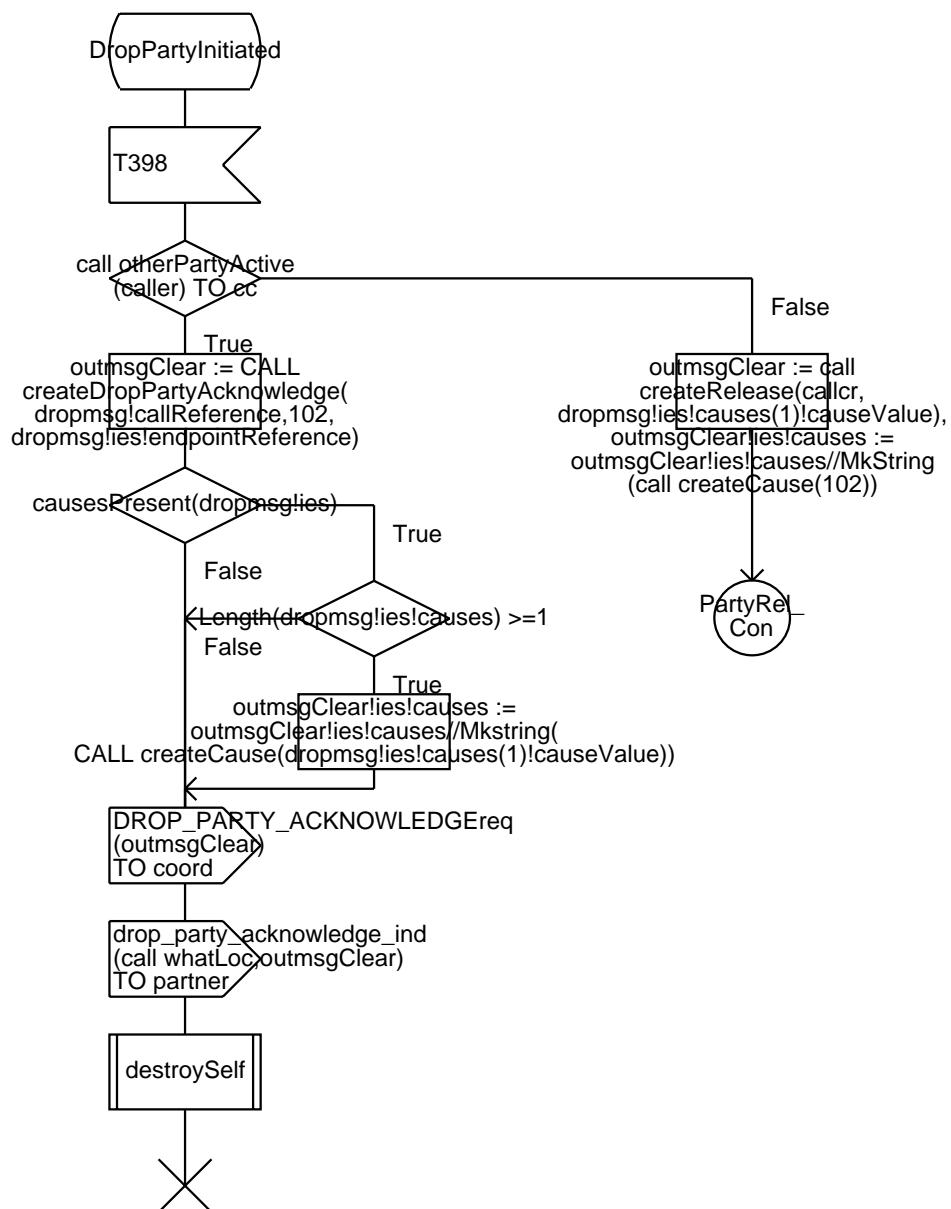


Annex B: PartyControl_PT

Process Type <<Block Type Q2931ss_BT>> PartyControl_PT

7(20)

INHERITS MsgHandler_PT;
 FPAR config InitTrunkConfig,partner PId, coord PId, cc PId, caller EndpointReference;

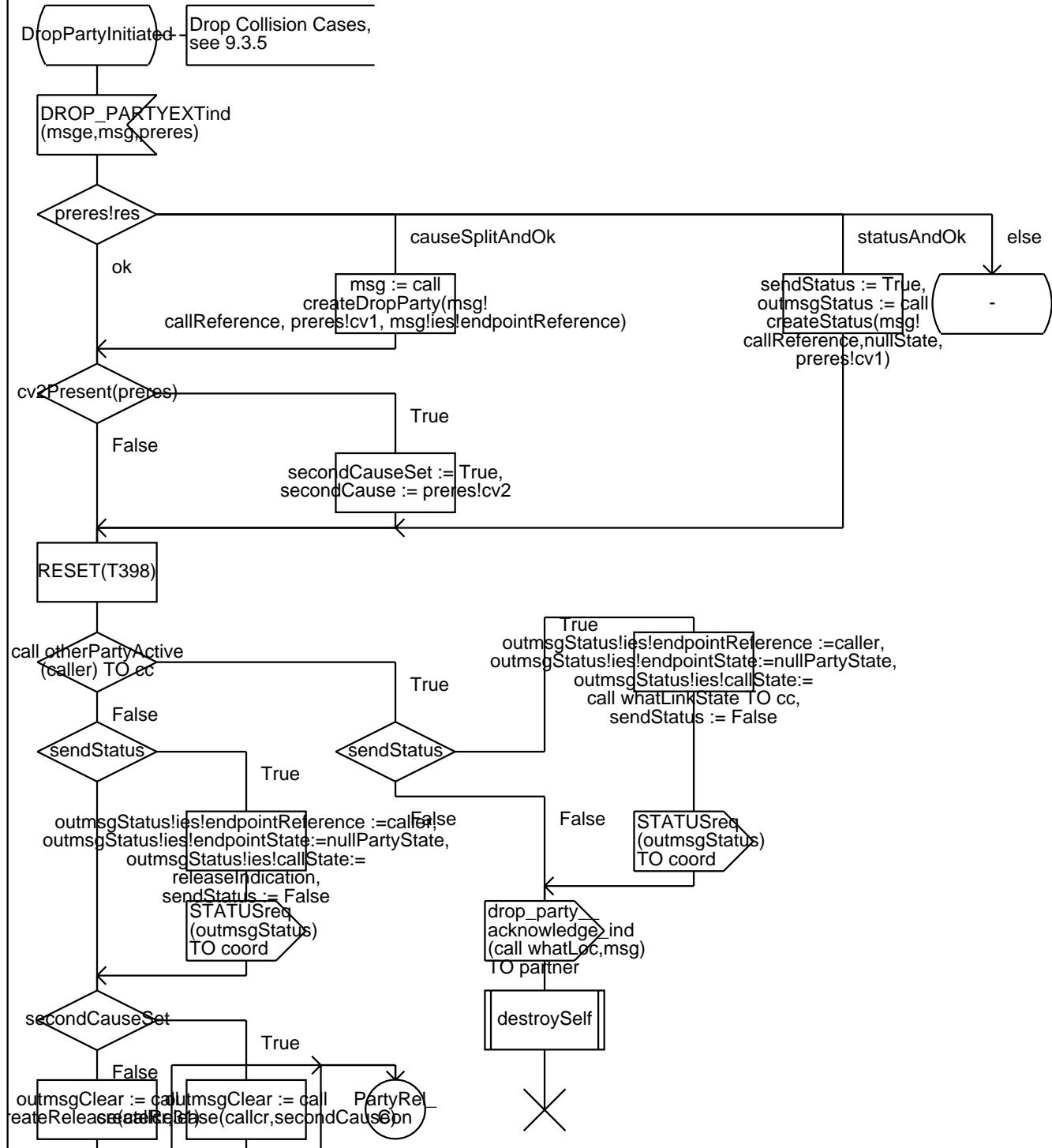


Annex B: PartyControl_PT

Process Type <<Block Type Q2931ss_BT>> PartyControl_PT

8(20)

INHERITS MsgHandler_PT;
 FPAR config InitTrunkConfig,partner PId, coord PId, cc PId, caller EndpointReference;

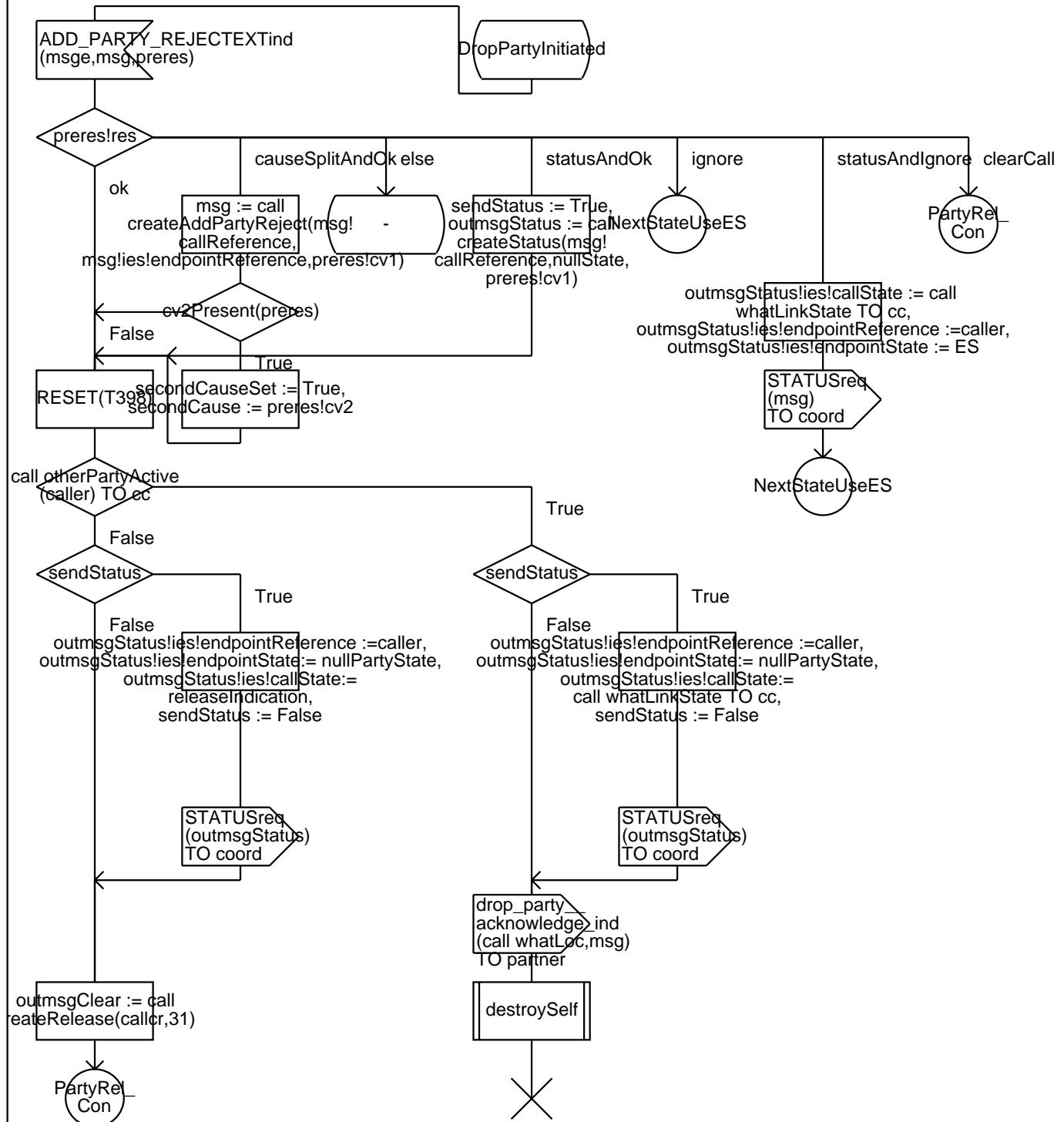


Annex B: PartyControl_PT

Process Type <<Block Type Q2931ss_BT>> PartyControl_PT

9(20)

INHERITS MsgHandler_PT;
 FPAR config InitTrunkConfig,partner PId, coord PId, cc PId, caller EndpointReference;

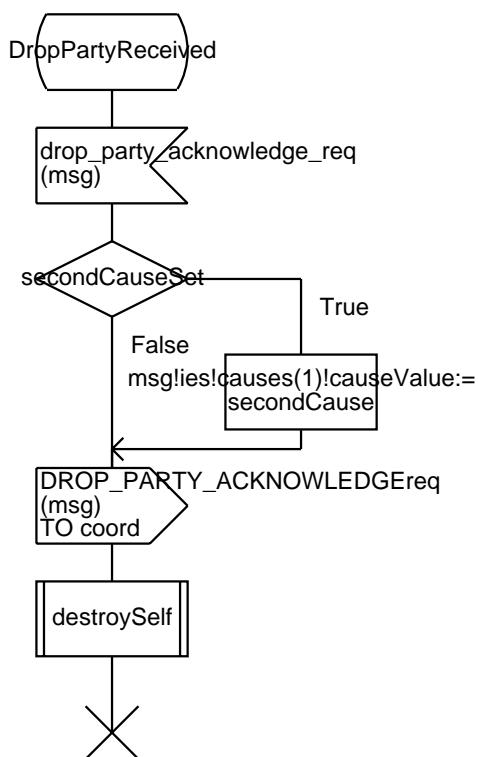


Annex B: PartyControl_PT

Process Type <<Block Type Q2931ss_BT>> PartyControl_PT

10(20)

INHERITS MsgHandler_PT;
 FPAR config InitTrunkConfig,partner PId, coord PId, cc PId, caller EndpointReference;

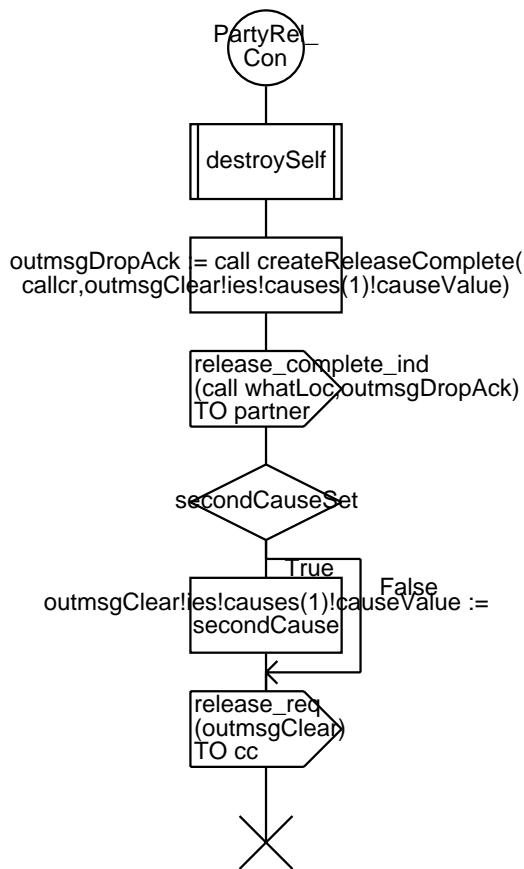


Annex B: PartyControl_PT

Process Type <<Block Type Q2931ss_BT>> PartyControl_PT

11(20)

INHERITS MsgHandler_PT;
 FPAR config InitTrunkConfig,partner PId, coord PId, cc PId, caller EndpointReference;

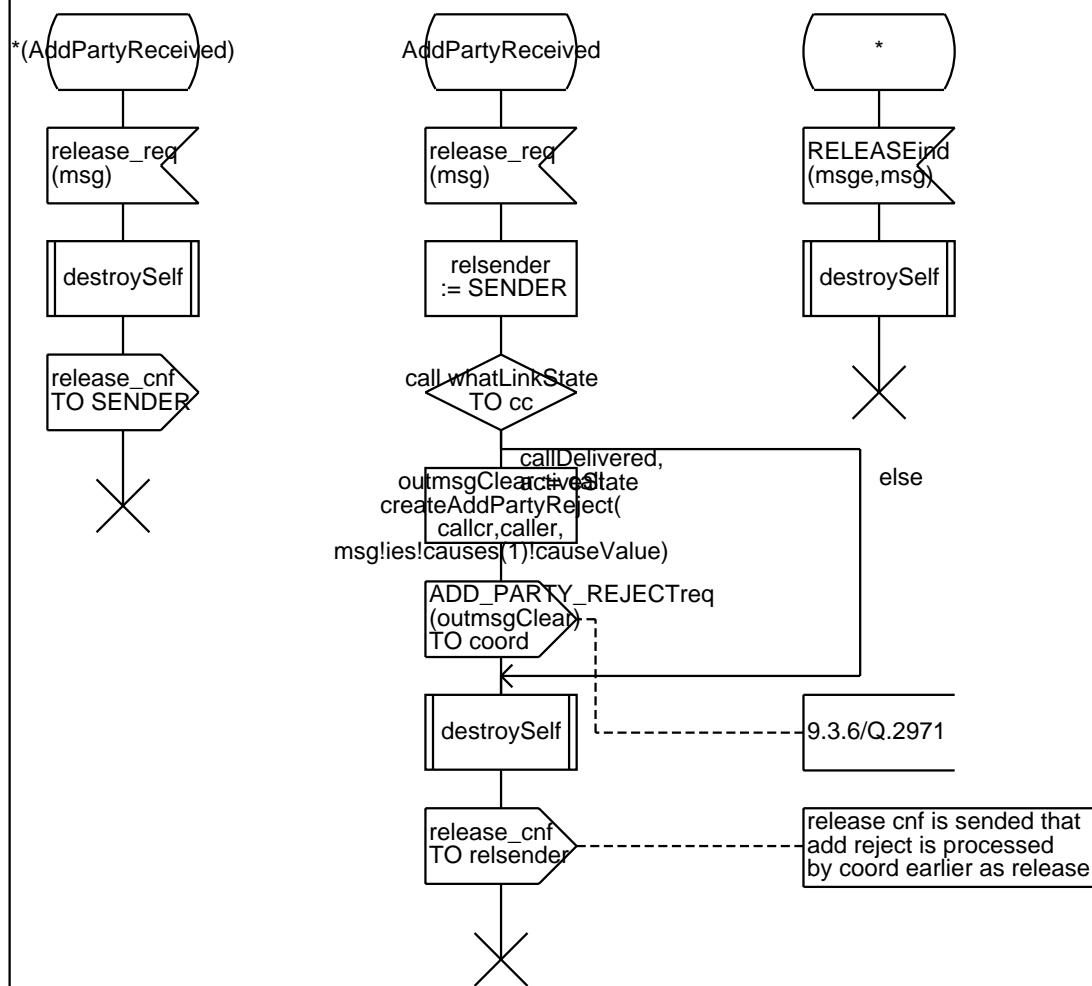


Annex B: PartyControl_PT

Process Type <<Block Type Q2931ss_BT>> PartyControl_PT

12(20)

INHERITS MsgHandler_PT;
 FPAR config InitTrunkConfig,partner PId, coord PId, cc PId, caller EndpointReference;

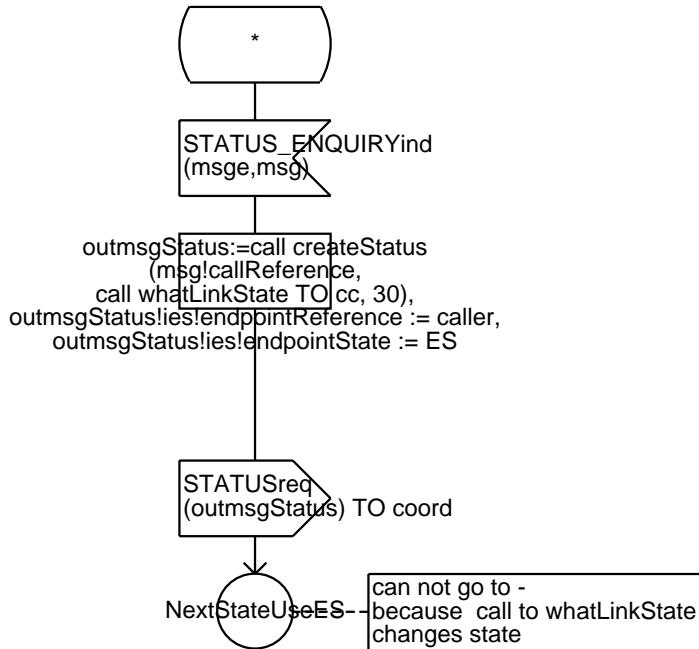


Annex B: PartyControl_PT

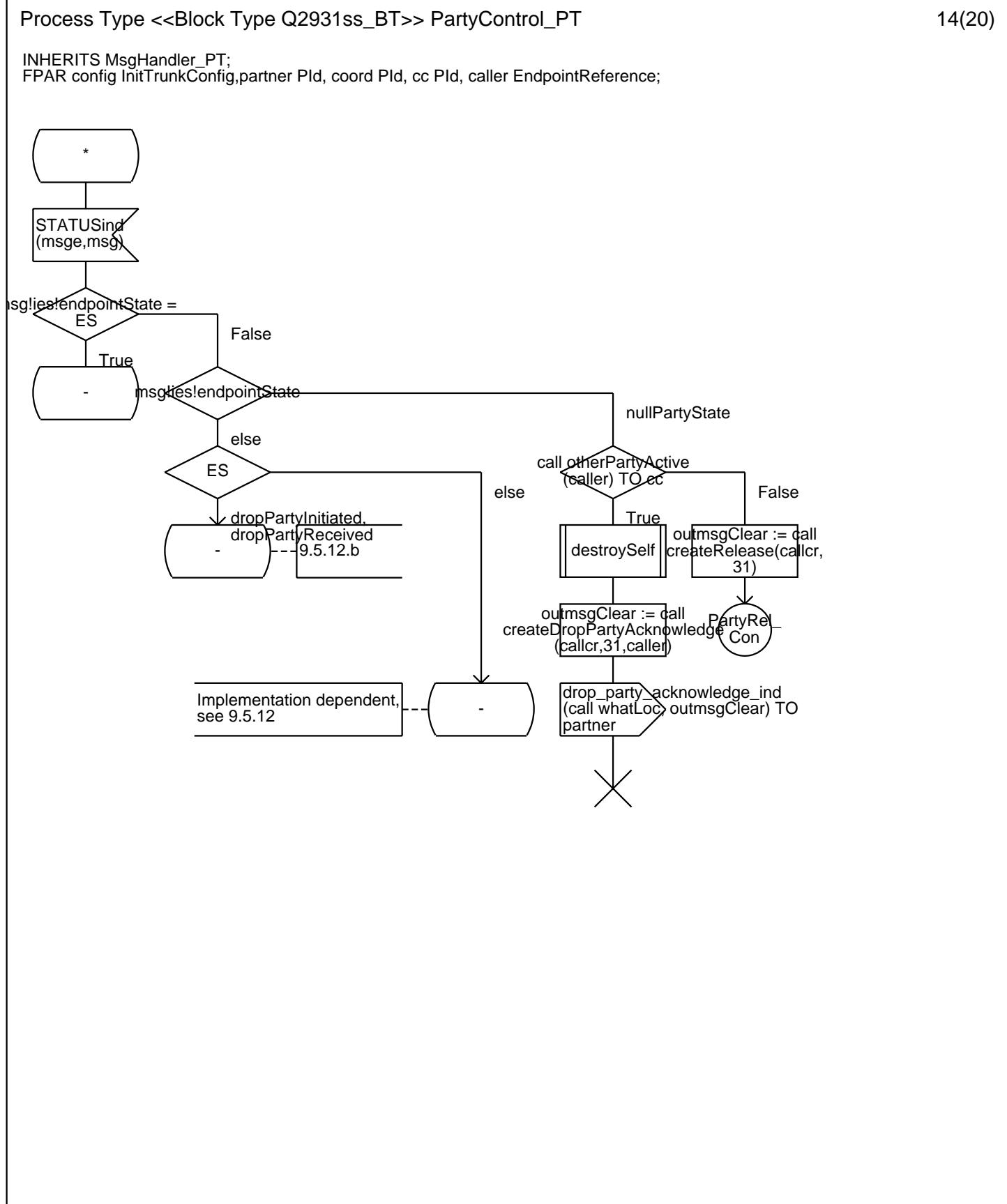
Process Type <<Block Type Q2931ss_BT>> PartyControl_PT

13(20)

INHERITS MsgHandler_PT;
 FPAR config InitTrunkConfig,partner PId, coord PId, cc PId, caller EndpointReference;



Annex B: PartyControl_PT

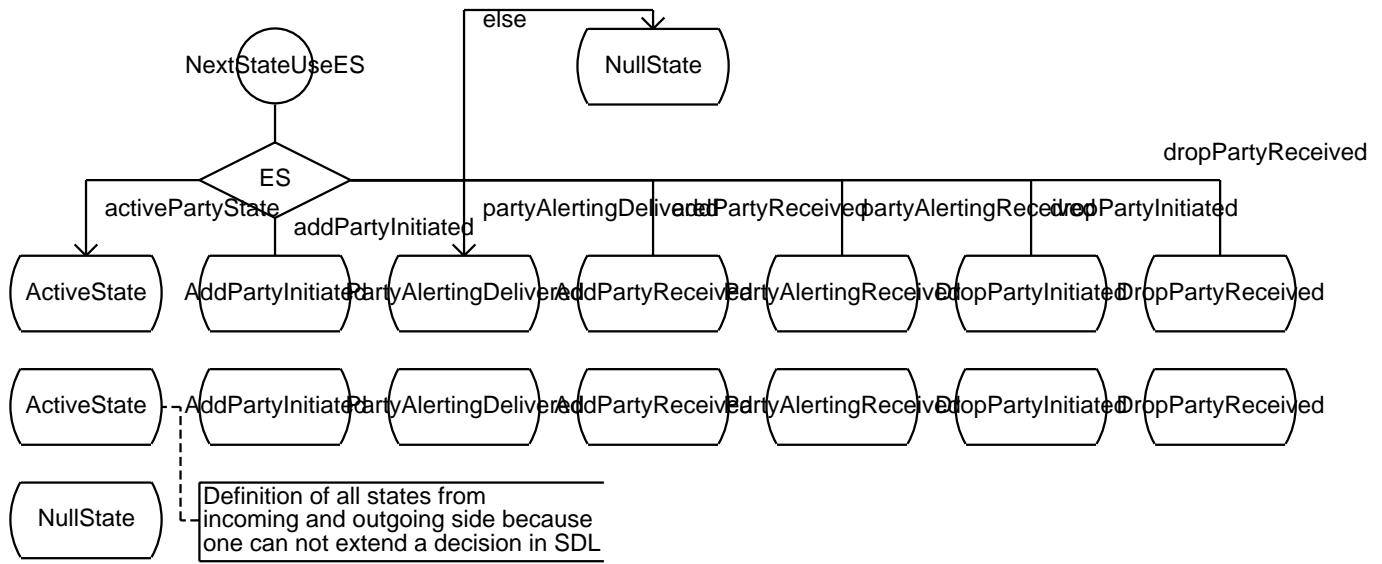


Annex B: PartyControl_PT

Process Type <<Block Type Q2931ss_BT>> PartyControl_PT

15(20)

INHERITS MsgHandler_PT;
 FPAR config InitTrunkConfig,partner PId, coord PId, cc PId, caller EndpointReference;

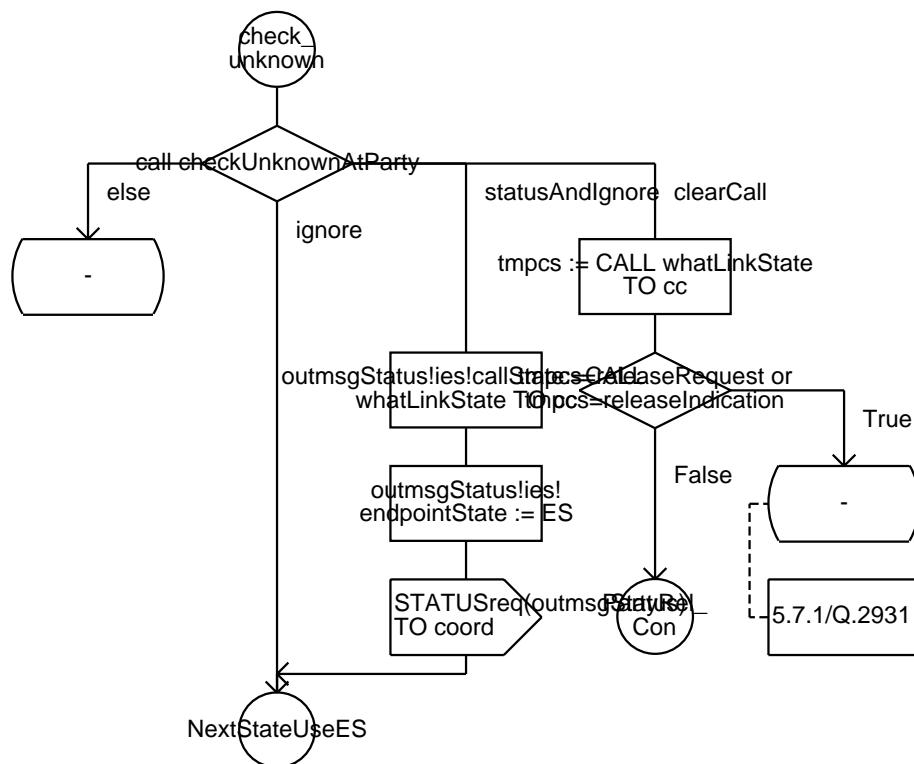


Annex B: PartyControl_PT

Process Type <<Block Type Q2931ss_BT>> PartyControl_PT

16(20)

INHERITS MsgHandler_PT;
 FPAR config InitTrunkConfig,partner PId, coord PId, cc PId, caller EndpointReference;

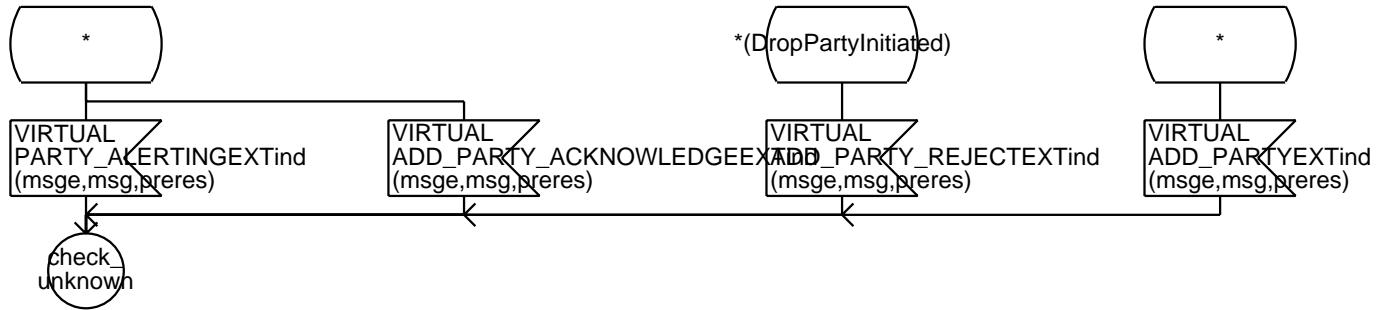


Annex B: PartyControl_PT

Process Type <<Block Type Q2931ss_BT>> PartyControl_PT

17(20)

INHERITS MsgHandler_PT;
 FPAR config InitTrunkConfig,partner PId, coord PId, cc PId, caller EndpointReference;

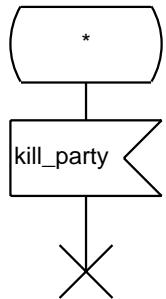


Annex B: PartyControl_PT

Process Type <<Block Type Q2931ss_BT>> PartyControl_PT

18(20)

```
INHERITS MsgHandler_PT;
FPAR config InitTrunkConfig,partner PId, coord PId, cc PId, caller EndpointReference;
```



Annex B: PartyControl_PT

Process Type <>Block Type Q2931ss_BT>> PartyControl_PT

19(20)

INHERITS MsgHandler_PT;
FPAR config InitTrunkConfig,partner PId, coord PId, cc PId, caller EndpointReference;

IMPORTED PROCEDURE whatLinkState;
RETURNS CallState;

IMPORTED PROCEDURE otherPartyActive;
FPAR EndpointReference;RETURNS Boolean;

Annex B: PartyControl_PT

Process Type <<Block Type Q2931ss_BT>> PartyControl_PT

20(20)

INHERITS MsgHandler_PT;
FPAR config InitTrunkConfig,partner PId, coord PId, cc PId, caller EndpointReference;

VIRTUAL whatLoc

EXPORTED whatPartyState

destroySelf

VIRTUAL resetTimers

checkAddPartyReject

checkDropParty

checkDropPartyAcknowledge

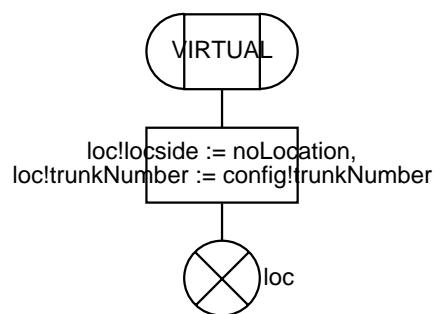
checkUnknownAtParty

Annex B: whatLoc

Virtual Procedure <<Process Type PartyControl_PT>> whatLoc

1(1)

;RETURNS loc Location;

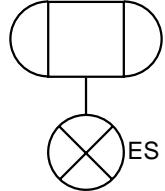


Annex B: whatPartyState

EXPORTED Procedure whatPartyState

1(1)

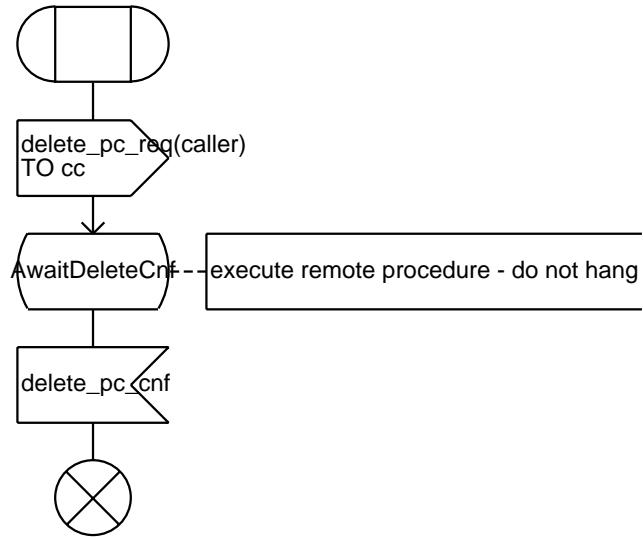
;returns EndpointState;



Annex B: destroySelf

Procedure <<Process Type PartyControl_PT>> destroySelf

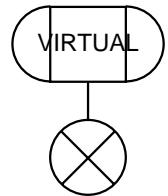
1(1)



Annex B: resetTimers

Virtual Procedure <<Process Type PartyControl_PT>> resetTimers

1(1)



Annex B: checkAddPartyReject

Procedure <<Process Type PartyControl_PT>> checkAddPartyReject

1(1)

;RETURNS res CheckResultType;

```
/*#include '../sdl/CheckaddPartyReject.sdl' */
```

Annex B: checkDropParty

Procedure checkDropParty

1(1)

;RETURNS res CheckResultType;

```
/*#include '../sdl/CheckdropParty.sdl' */
```

Annex B: checkDropPartyAcknowledge

Procedure checkDropPartyAcknowledge

1(1)

;RETURNS res CheckResultType;

```
/*#include '../sdl/CheckdropPartyAcknowledge.sdl' */
```

Annex B: checkUnknownAtParty

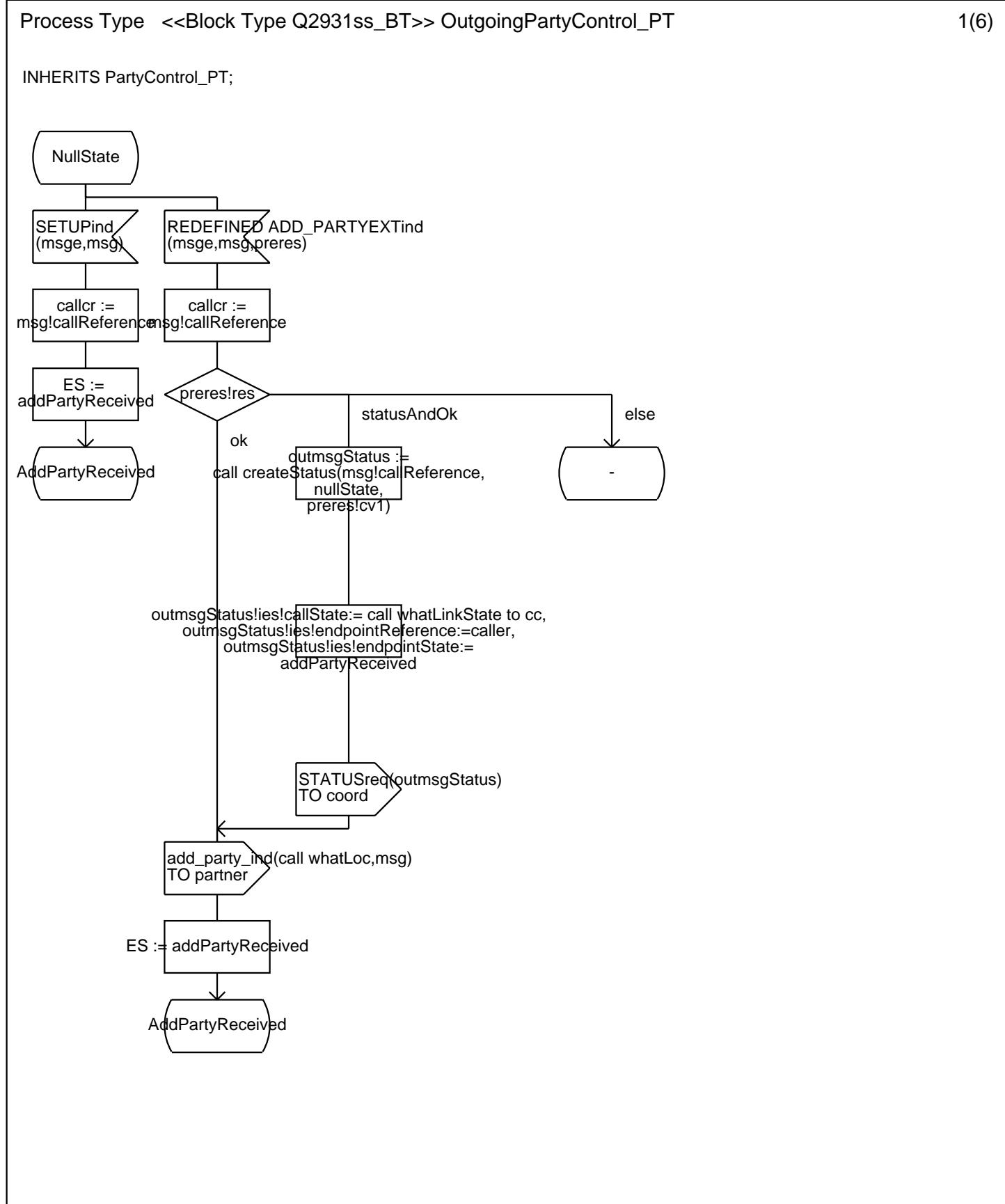
Procedure checkUnknownAtParty

1(1)

;RETURNS res CheckResultType;

```
/*#include '../sdl/CheckunknownAtParty.sdl' */
```

Annex B: OutgoingPartyControl_PT

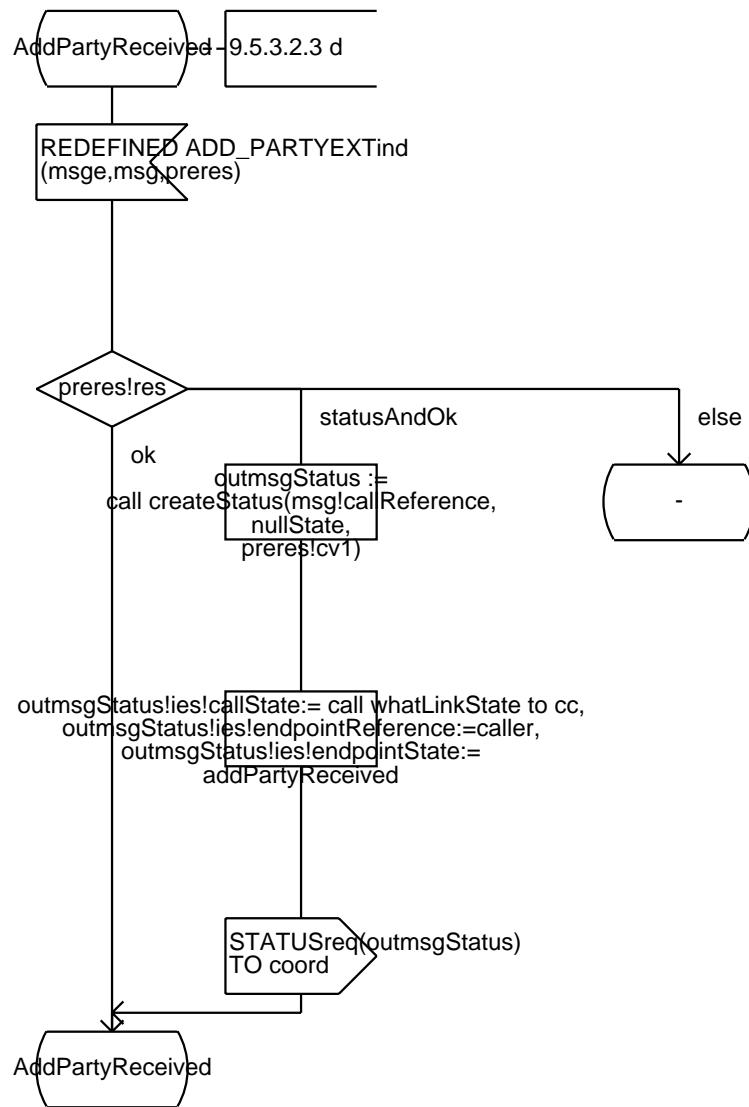


Annex B: OutgoingPartyControl_PT

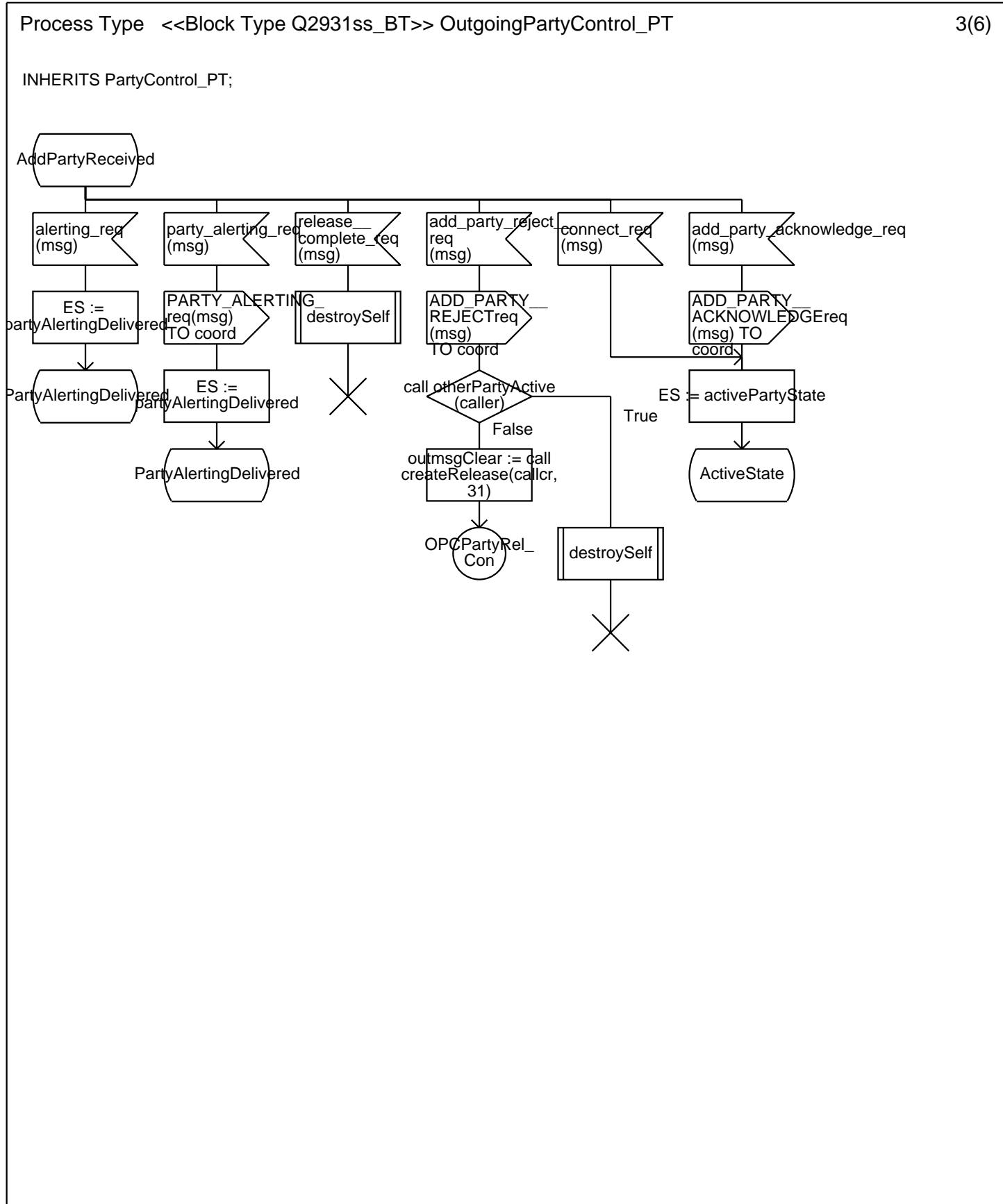
Process Type <<Block Type Q2931ss_BT>> OutgoingPartyControl_PT

2(6)

INHERITS PartyControl_PT;



Annex B: OutgoingPartyControl_PT

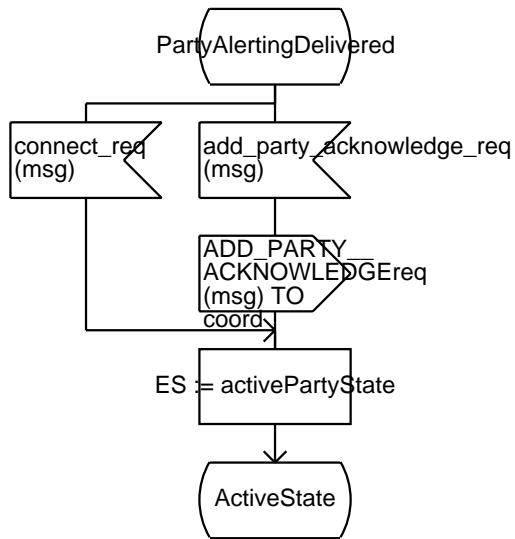


Annex B: OutgoingPartyControl_PT

Process Type <<Block Type Q2931ss_BT>> OutgoingPartyControl_PT

4(6)

INHERITS PartyControl_PT;

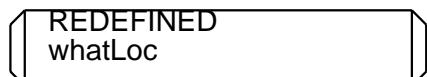


Annex B: OutgoingPartyControl_PT

Process Type <<Block Type Q2931ss_BT>> OutgoingPartyControl_PT

5(6)

INHERITS PartyControl_PT;

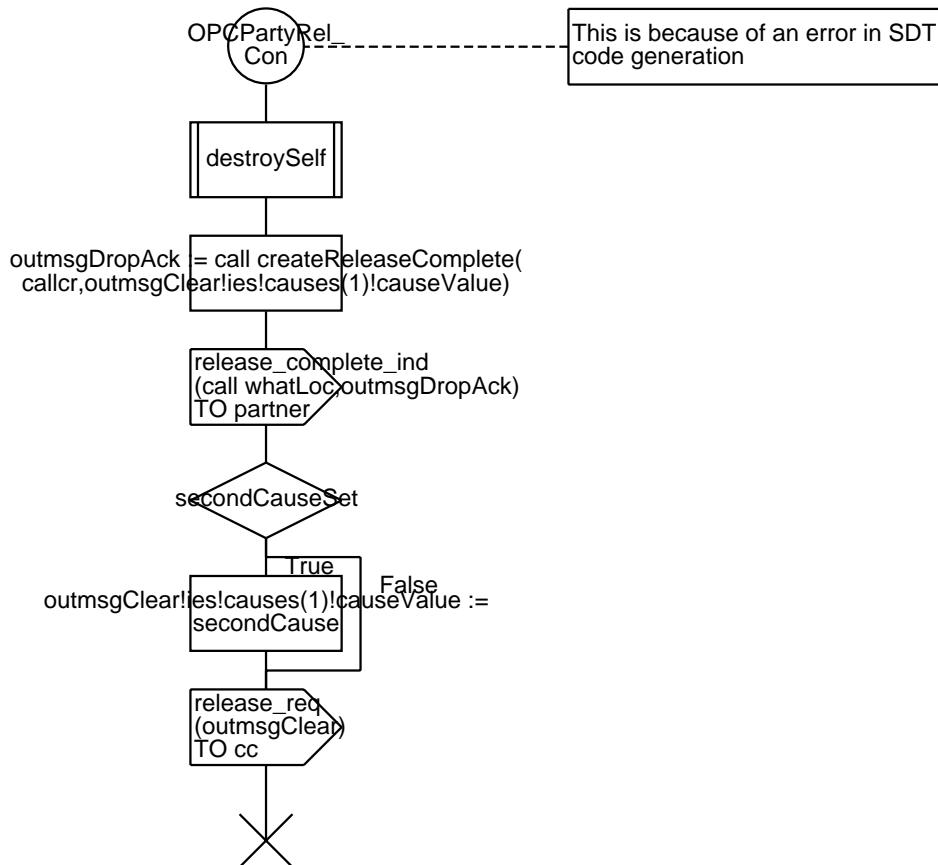
A rounded rectangle containing the text "REDEFINED whatLoc".A rounded rectangle containing the text "checkAddParty".

Annex B: OutgoingPartyControl_PT

Process Type <<Block Type Q2931ss_BT>> OutgoingPartyControl_PT

6(6)

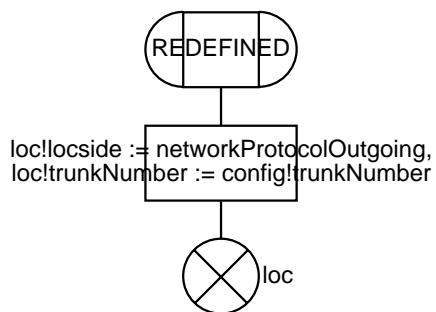
INHERITS PartyControl_PT;



Annex B: whatLoc

Redefined Procedure <<Process Type OutgoingPartyControl_PT>> whatLoc

1(1)



Annex B: checkAddParty

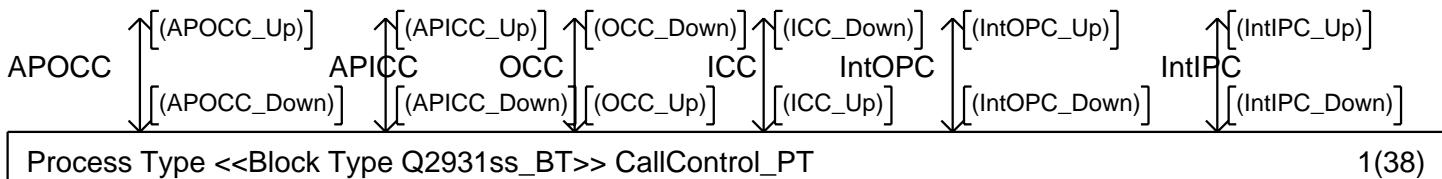
Procedure checkAddParty

1(1)

;RETURNS res CheckResultType;

```
/*#include '../sdl/CheckaddParty.sdl' */
```

Annex B: CallControl_PT



Process Type <<Block Type Q2931ss_BT>> CallControl_PT

1(38)

INHERITS MsgHandler_PT;
 fpar partner PId, gfppartner PId, config InitTrunkConfig;

```

/* Variables valid in all transitions */
DCL partyDict ERToPId := (. Null .); /* Standard Value Null */
DCL partyKeys ERSet := Empty;
DCL coord PId; /* Coordinator */
DCL CS CallState; /* Current Call State */
DCL callcr CallReference; /* Set during Setup */
DCL caller EndpointReference; /* ER of first SETUP */
DCL callerSet Boolean := False; /* does first setup have er */
DCL repeatStatusMaxTimes Integer := 2; /* how many times repeat
status - implementation dependant */
DCL repeatStatusTimes Integer := 0; /* how many times
status enquiry already repeated */
DCL statusQueue ERString := ERStringEmpty;
/* waiting ERs for sending Status Enquiry */
DCL statusQueueIndex Integer := 0;
/* last index checked in wait queue */
DCL statusEnqSended Boolean := False;
/* Status Enquiry sended for one party - do
not send it for link process 9.5.10,11/Q.2971 */
DCL repeatStatusMsg Q2931ssMessage;
DCL apRequestedStatusEnq Boolean := False;
/* did Application request sending status enq? */
DCL firstTimeAALRelease Boolean := True;
/* first Time AAL_RELEASEind received */
  
```

/*
partner is application process id,
rptype is reference point type,
trunknr is trunk number
*/

```

TIMER T308 := 30; /* 30 seconds */
TIMER T322 := 4; /* 4 seconds */
TIMER delayDestroyT := 2000; /* 2 seconds for delayed destroy */
  
```

Annex B: CallControl_PT

Process Type <>Block Type Q2931ss_BT>> CallControl_PT

2(38)

INHERITS MsgHandler_PT;
fpair partner PId, gfppartner PId, config InitTrunkConfig;

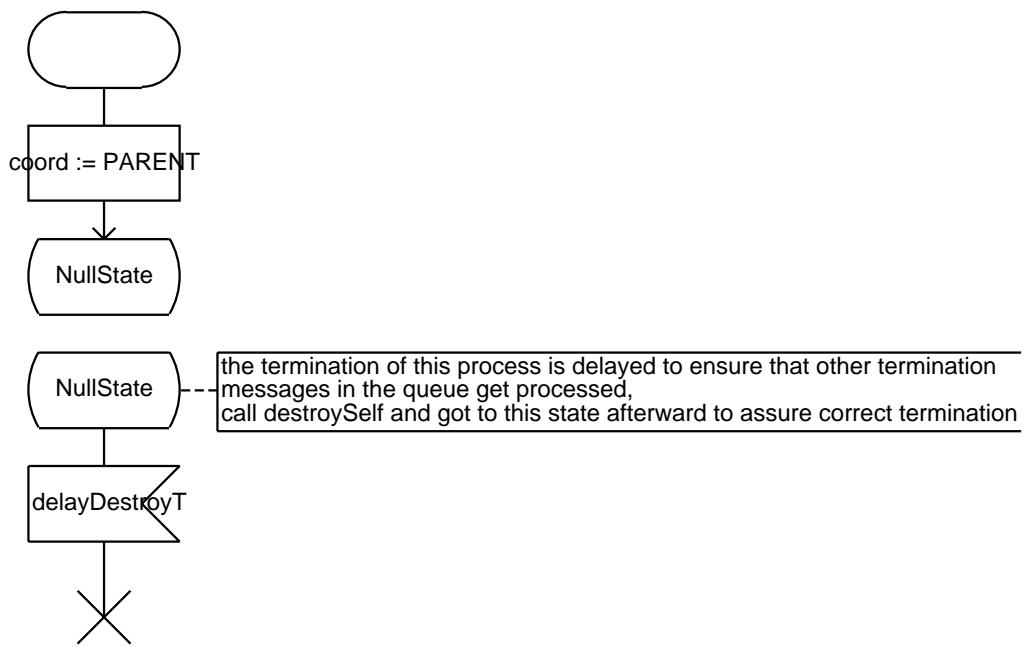
```
/* Variable valid only during one transition */  
DCL er EndpointReference;  
DCL pid PId;  
DCL firstTimeRelease Boolean;  
DCL firstTimeRel Q2931ssMessage;  
DCL statusTimerWasActive Boolean := False;  
DCL sendStatus Boolean := False;  
DCL preres CheckResultTypeStatus;  
DCL genpreres CheckResultTypeStatus;  
DCL tmpres CheckResultType;
```

Annex B: CallControl_PT

Process Type <<Block Type Q2931ss_BT>> CallControl_PT

3(38)

INHERITS MsgHandler_PT;
fpar partner PId, gfppartner PId, config InitTrunkConfig;

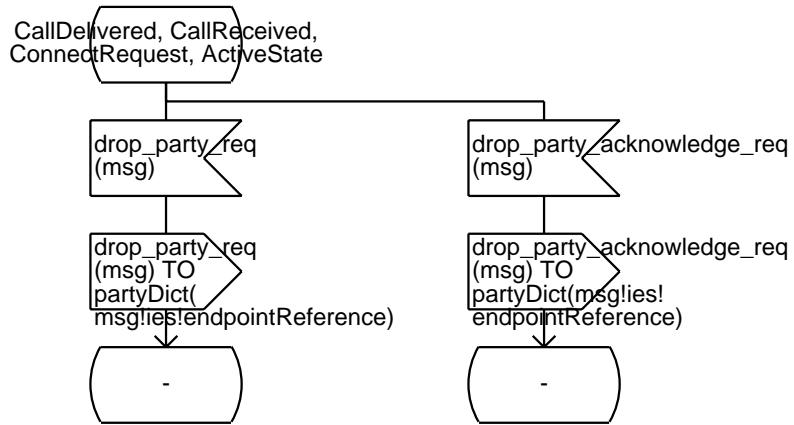


Annex B: CallControl_PT

Process Type <<Block Type Q2931ss_BT>> CallControl_PT

4(38)

INHERITS MsgHandler_PT;
fpar partner PId, gfppartner PId, config InitTrunkConfig;

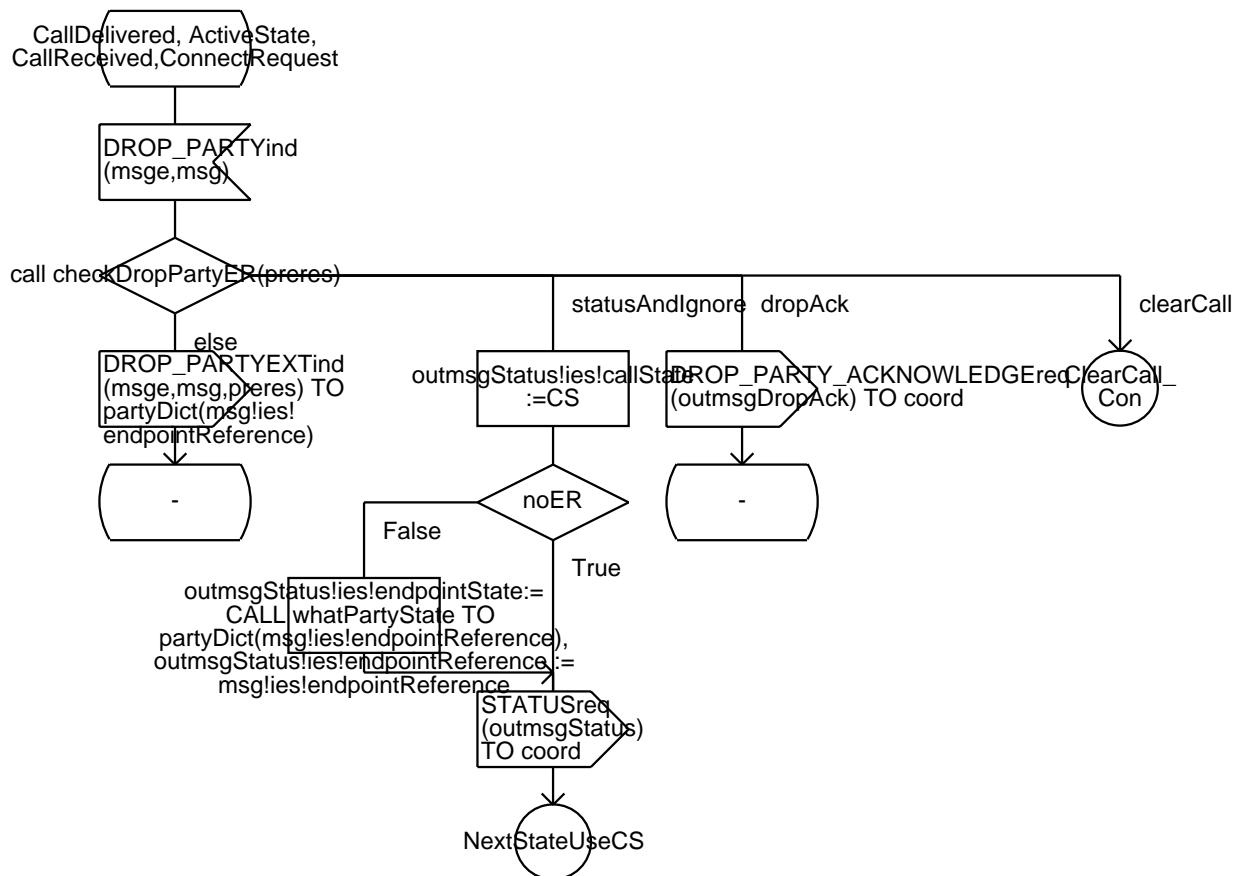


Annex B: CallControl_PT

Process Type <<Block Type Q2931ss_BT>> CallControl_PT

5(38)

INHERITS MsgHandler_PT;
 fpair partner PId, gfppartner PId, config InitTrunkConfig;

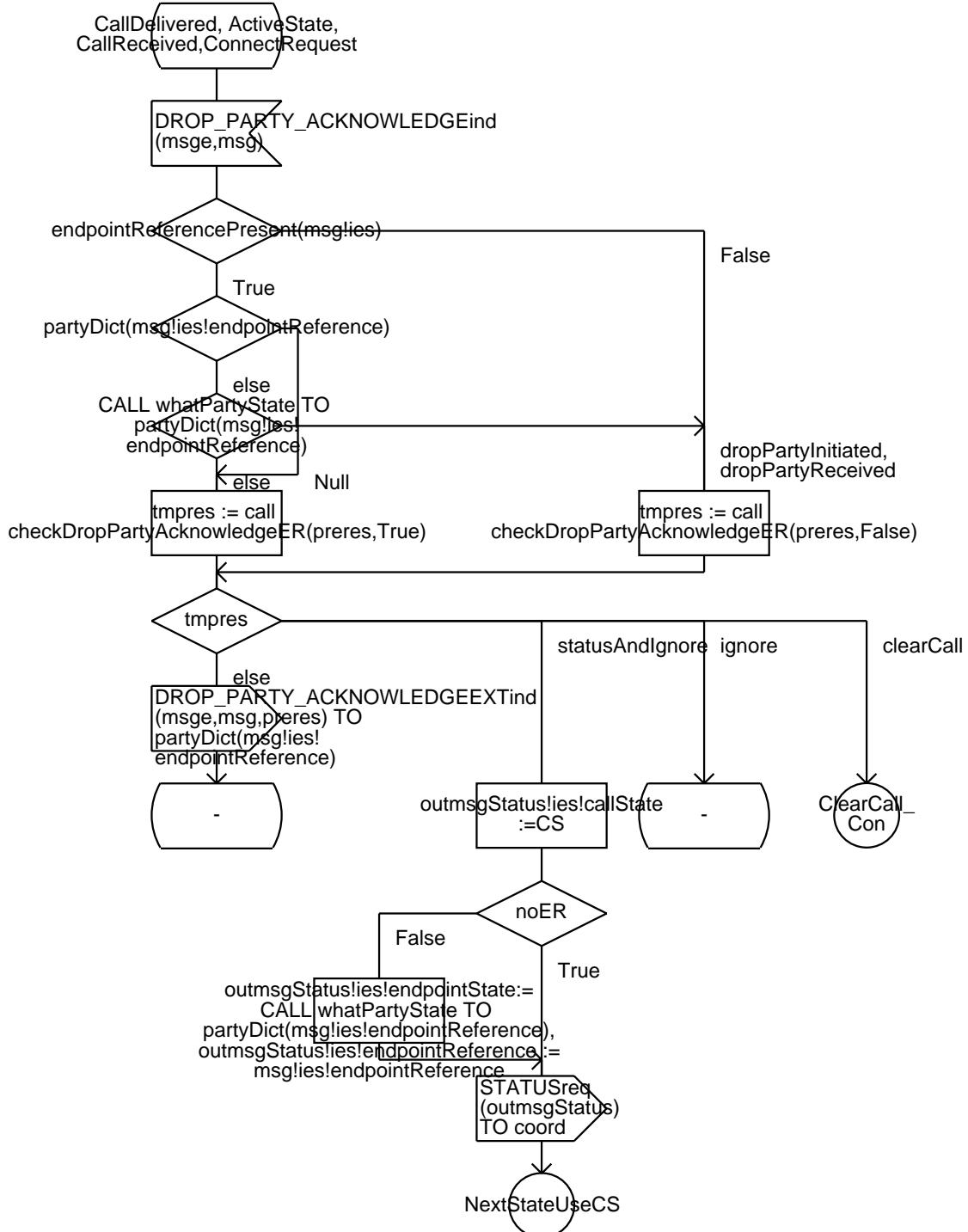


Annex B: CallControl_PT

Process Type <<Block Type Q2931ss_BT>> CallControl_PT

6(38)

INHERITS MsgHandler_PT;
 fpair partner PId, gfppartner PId, config InitTrunkConfig;

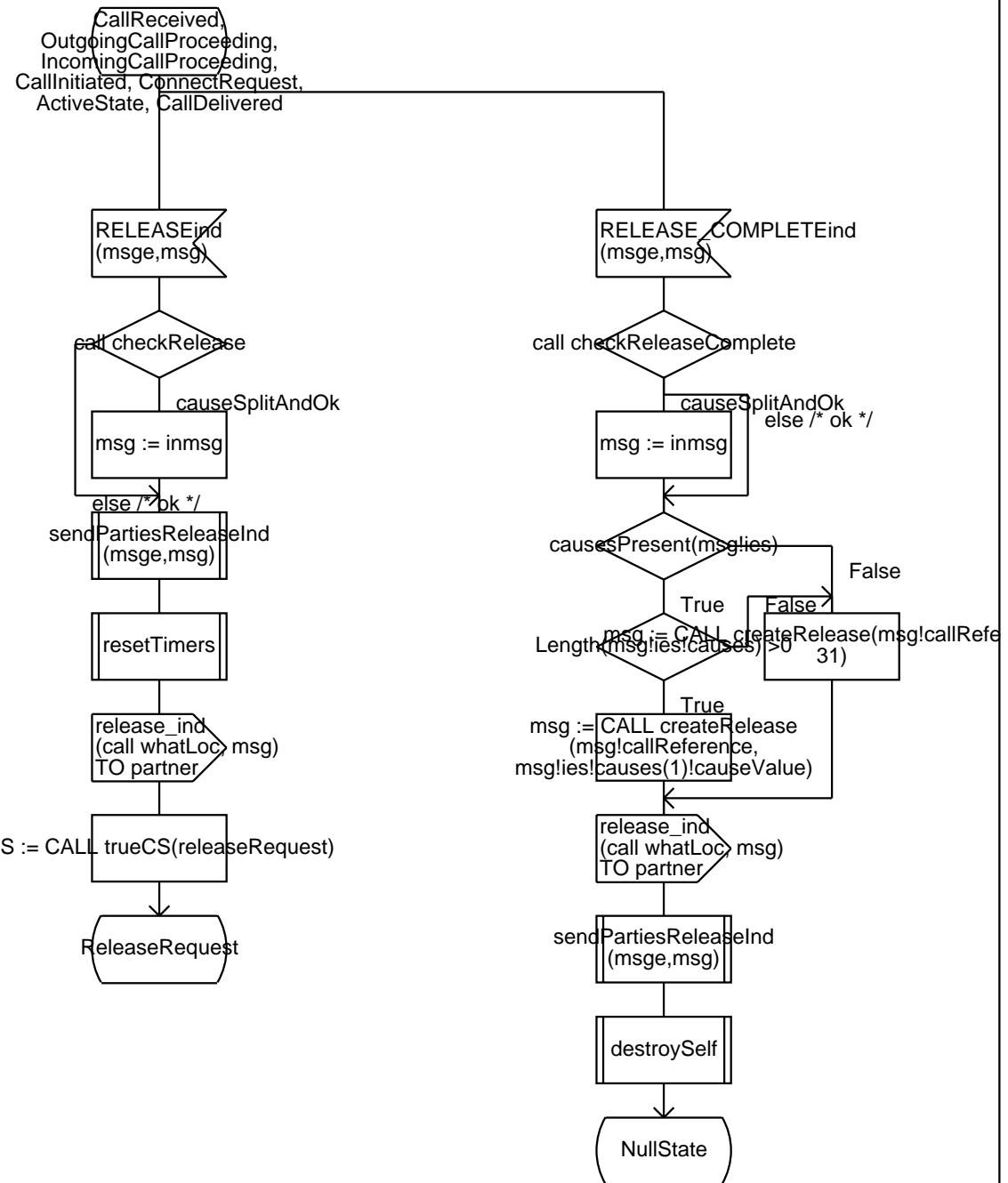


Annex B: CallControl_PT

Process Type <<Block Type Q2931ss_BT>> CallControl_PT

7(38)

INHERITS MsgHandler_PT;
 fpair partner PId, gfppartner PId, config InitTrunkConfig;

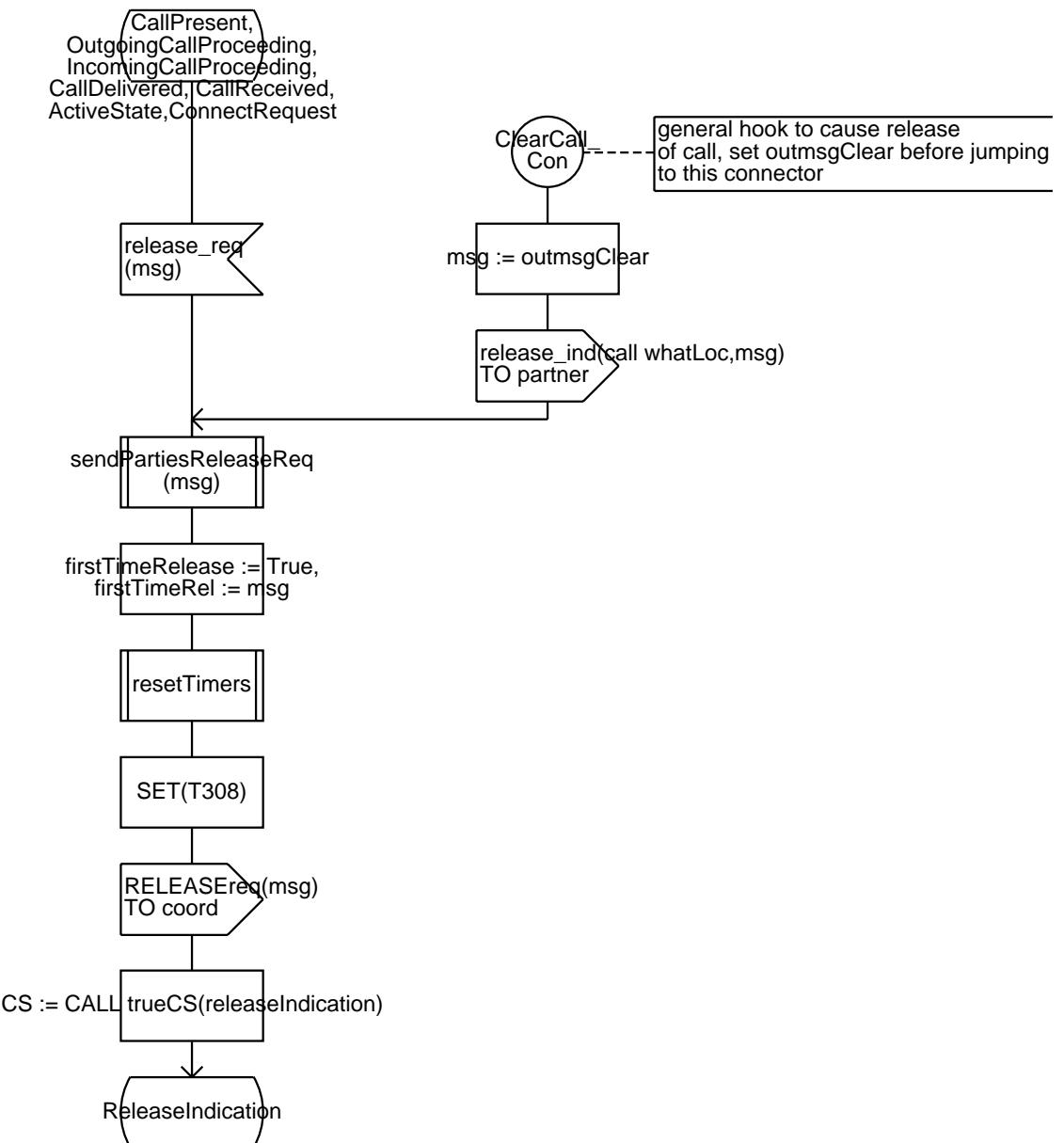


Annex B: CallControl_PT

Process Type <<Block Type Q2931ss_BT>> CallControl_PT

8(38)

INHERITS MsgHandler_PT;
 fpair partner PId, gfppartner PId, config InitTrunkConfig;

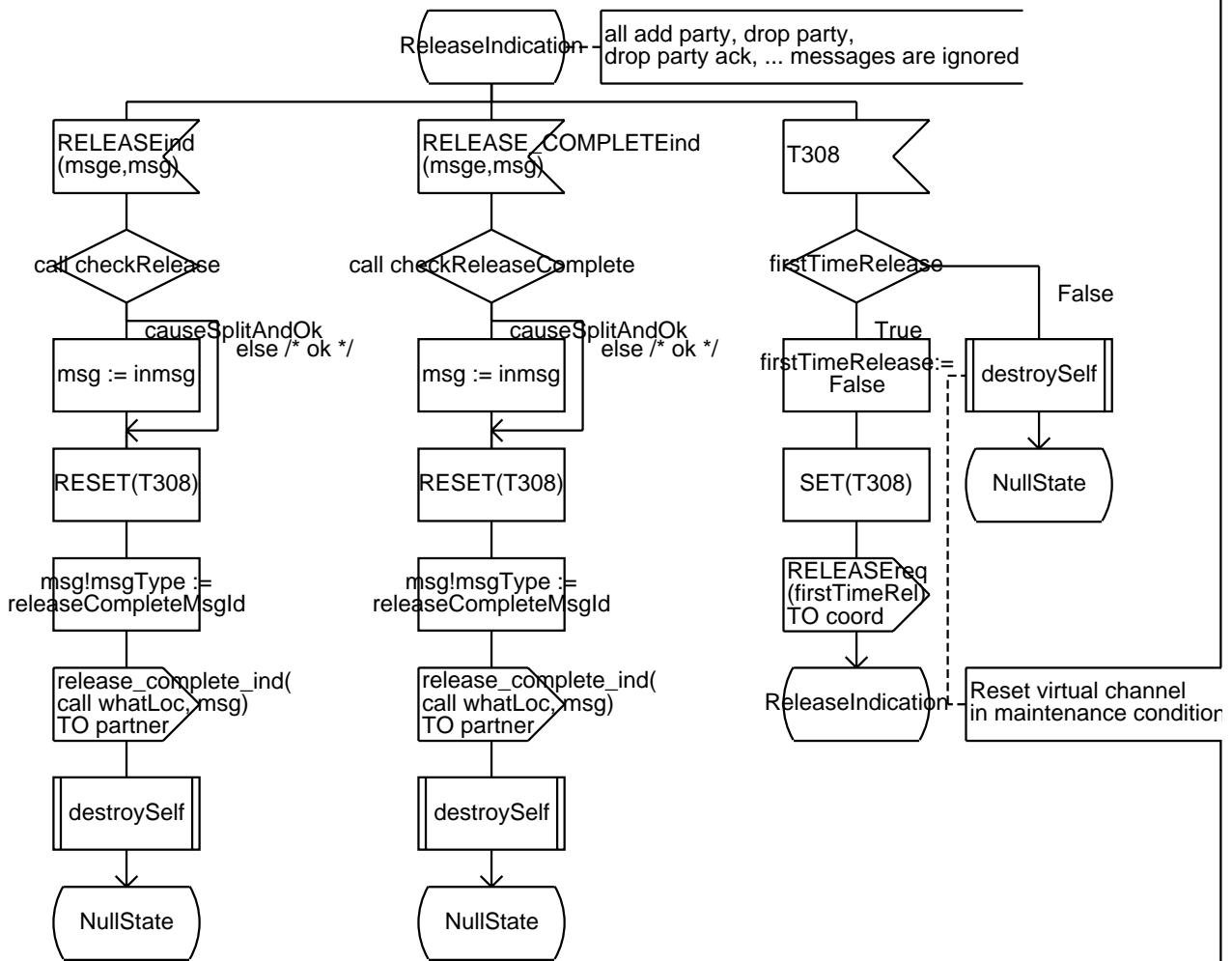


Annex B: CallControl_PT

Process Type <<Block Type Q2931ss_BT>> CallControl_PT

9(38)

INHERITS MsgHandler_PT;
 fpair partner PId, gfppartner PId, config InitTrunkConfig;

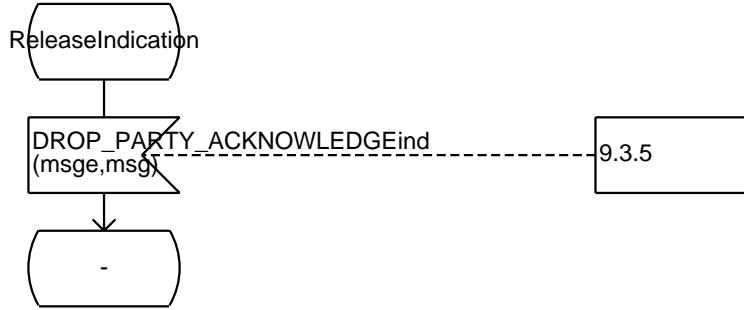


Annex B: CallControl_PT

Process Type <<Block Type Q2931ss_BT>> CallControl_PT

10(38)

INHERITS MsgHandler_PT;
fpar partner PId, gfppartner PId, config InitTrunkConfig;

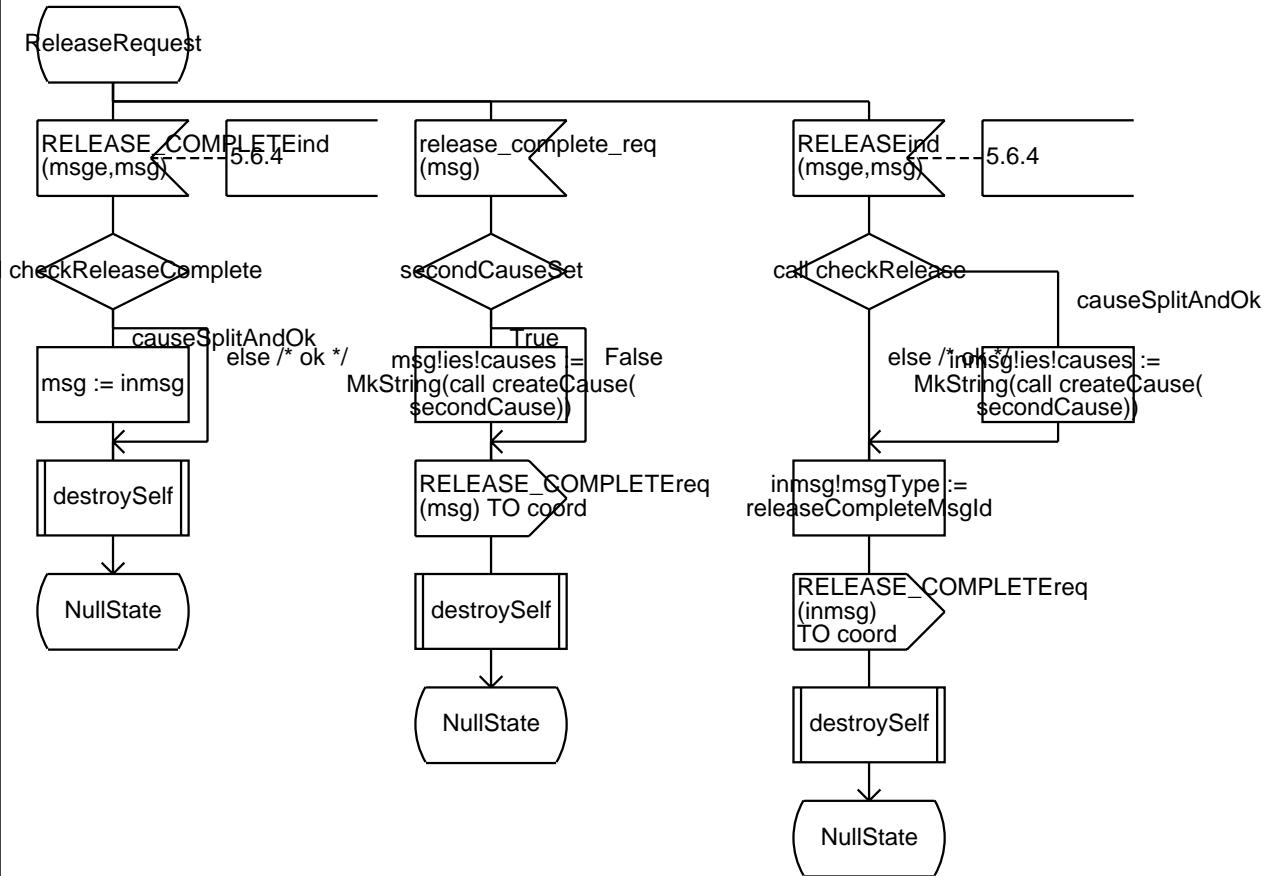


Annex B: CallControl_PT

Process Type <<Block Type Q2931ss_BT>> CallControl_PT

11(38)

INHERITS MsgHandler_PT;
 fpair partner PId, gfppartner PId, config InitTrunkConfig;

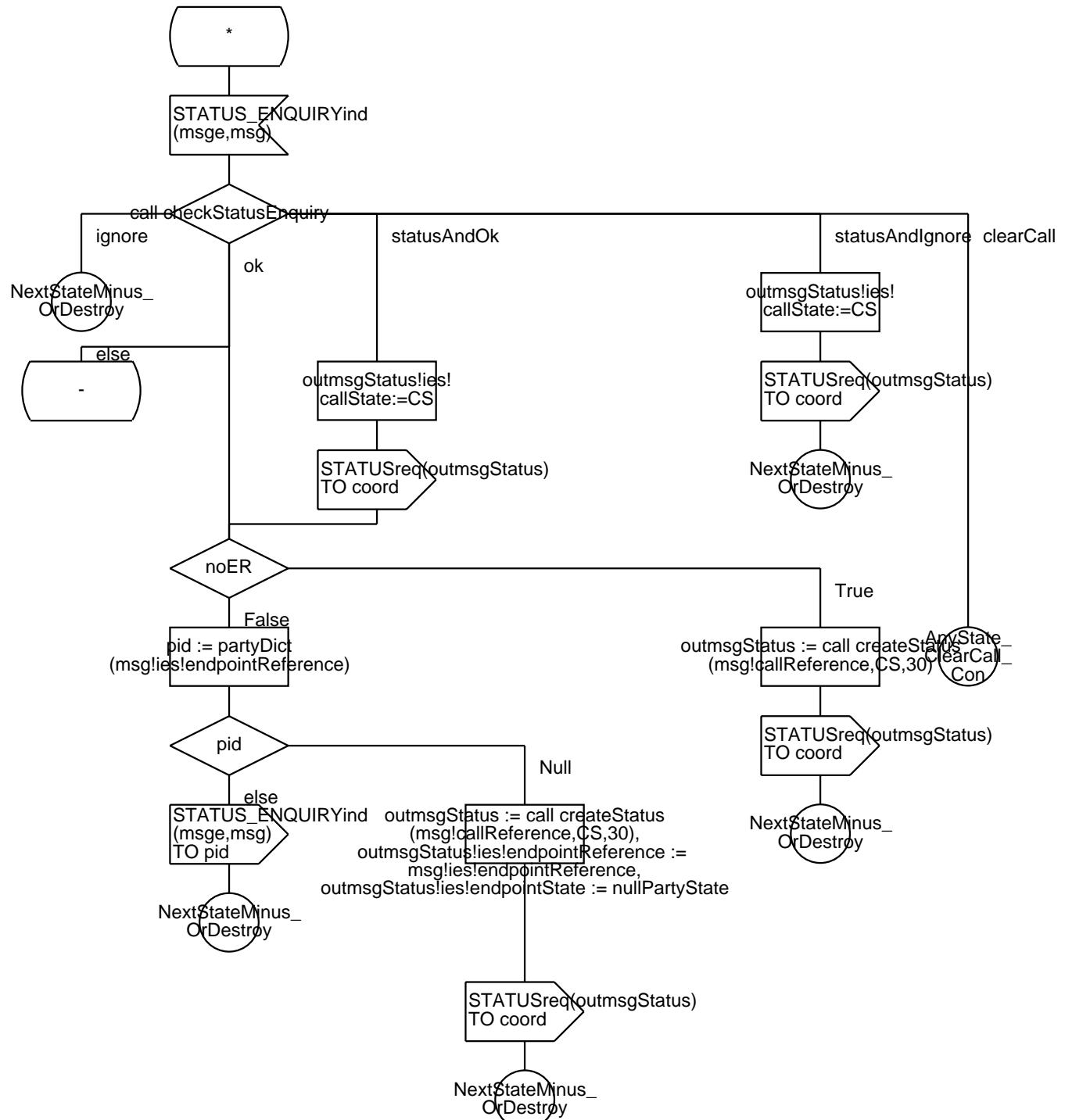


Annex B: CallControl_PT

Process Type <<Block Type Q2931ss_BT>> CallControl_PT

12(38)

INHERITS MsgHandler_PT;
 fpair partner PId, gfppartner PId, config InitTrunkConfig;

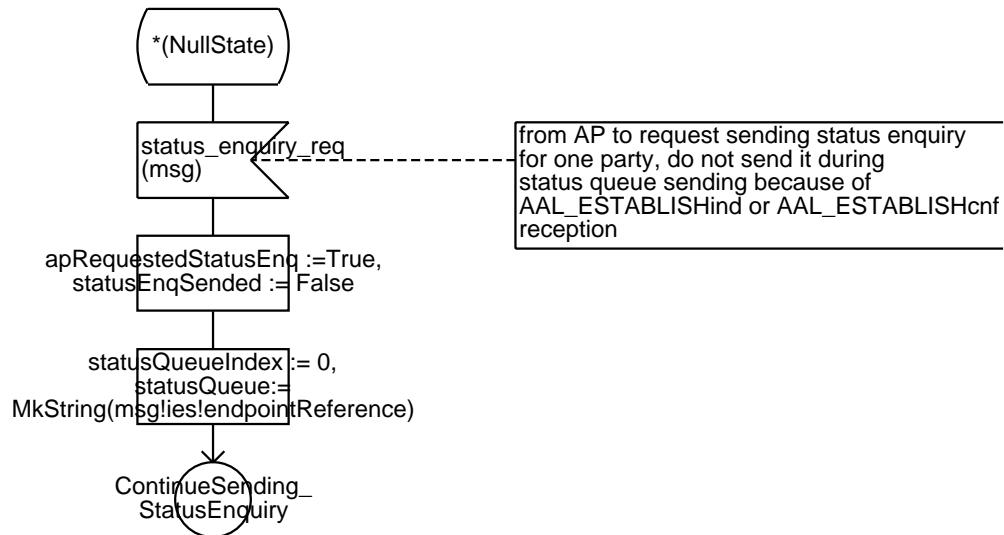


Annex B: CallControl_PT

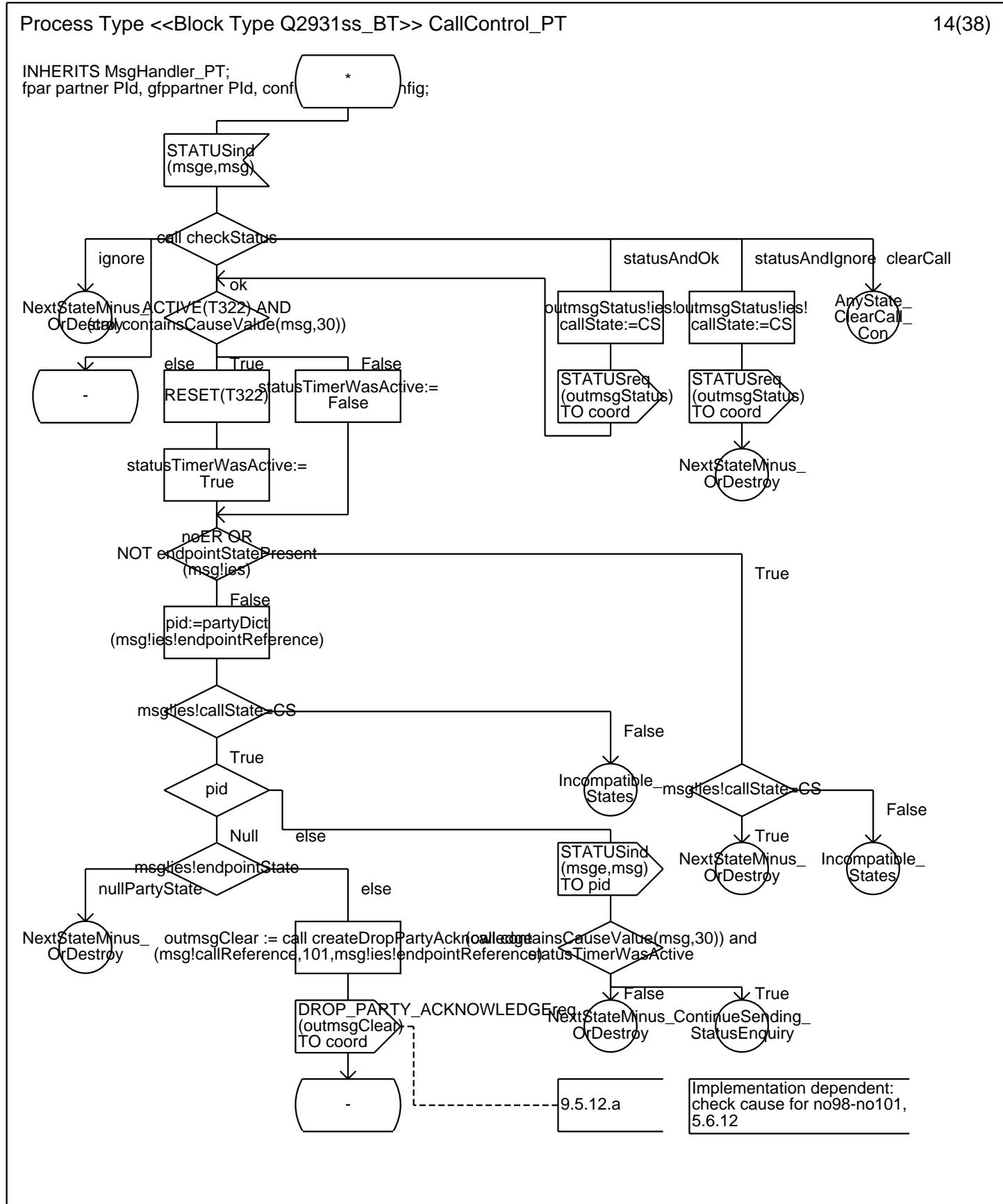
Process Type <>Block Type Q2931ss_BT>> CallControl_PT

13(38)

INHERITS MsgHandler_PT;
 fpair partner PId, gfppartner PId, config InitTrunkConfig;



Annex B: CallControl_PT

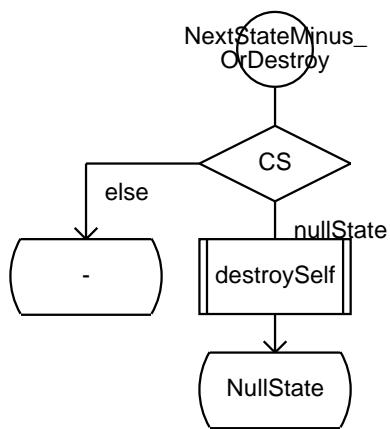


Annex B: CallControl_PT

Process Type <<Block Type Q2931ss_BT>> CallControl_PT

15(38)

INHERITS MsgHandler_PT;
fpar partner PId, gfppartner PId, config InitTrunkConfig;

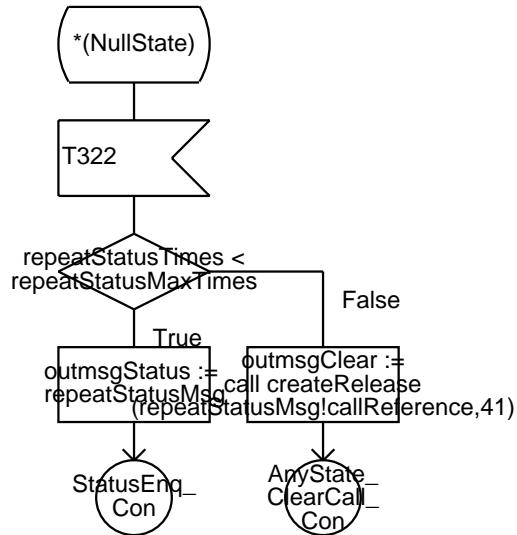


Annex B: CallControl_PT

Process Type <<Block Type Q2931ss_BT>> CallControl_PT

16(38)

INHERITS MsgHandler_PT;
fpar partner PId, gfppartner PId, config InitTrunkConfig;

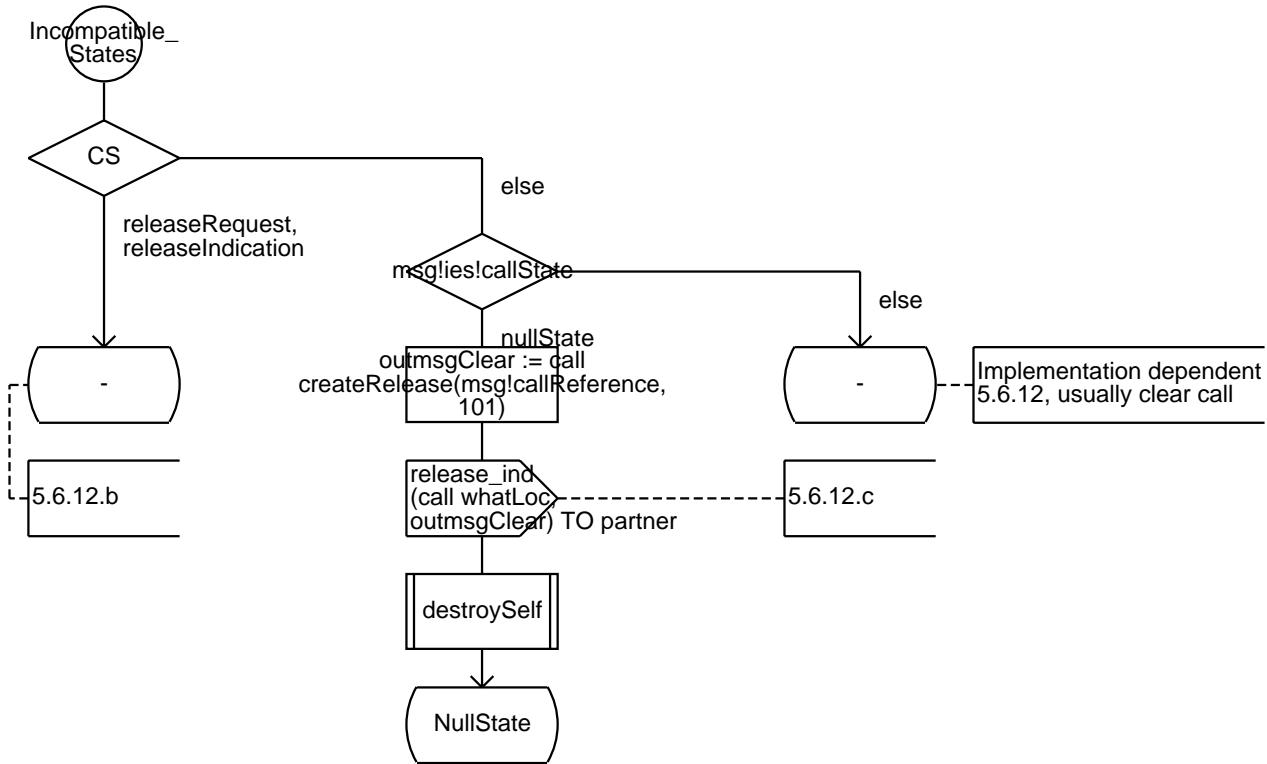


Annex B: CallControl_PT

Process Type <<Block Type Q2931ss_BT>> CallControl_PT

17(38)

INHERITS MsgHandler_PT;
 fpair partner PId, gfppartner PId, config InitTrunkConfig;

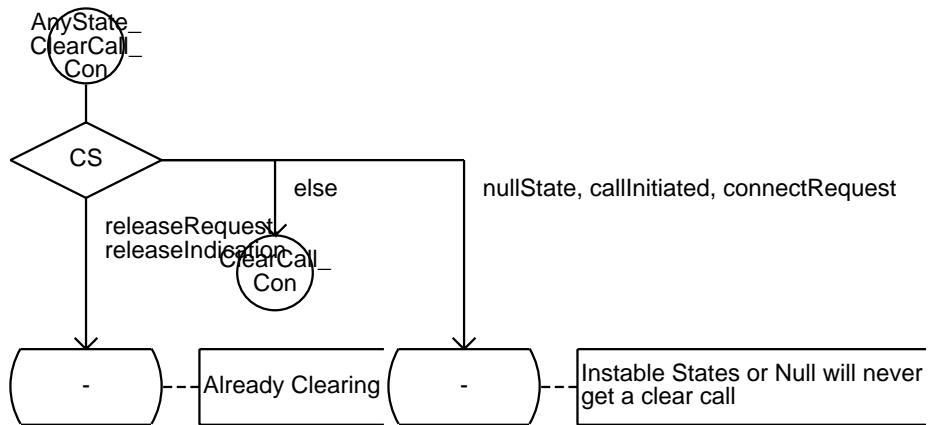


Annex B: CallControl_PT

Process Type <<Block Type Q2931ss_BT>> CallControl_PT

18(38)

INHERITS MsgHandler_PT;
 fpair partner PId, gfppartner PId, config InitTrunkConfig;

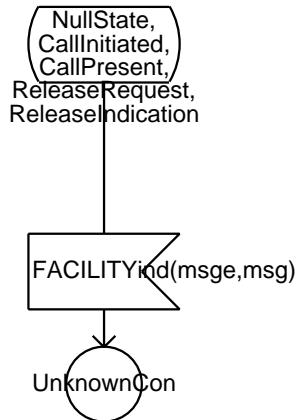


Annex B: CallControl_PT

Process Type <<Block Type Q2931ss_BT>> CallControl_PT

19(38)

INHERITS MsgHandler_PT;
fpar partner PId, gfppartner PId, config InitTrunkConfig;

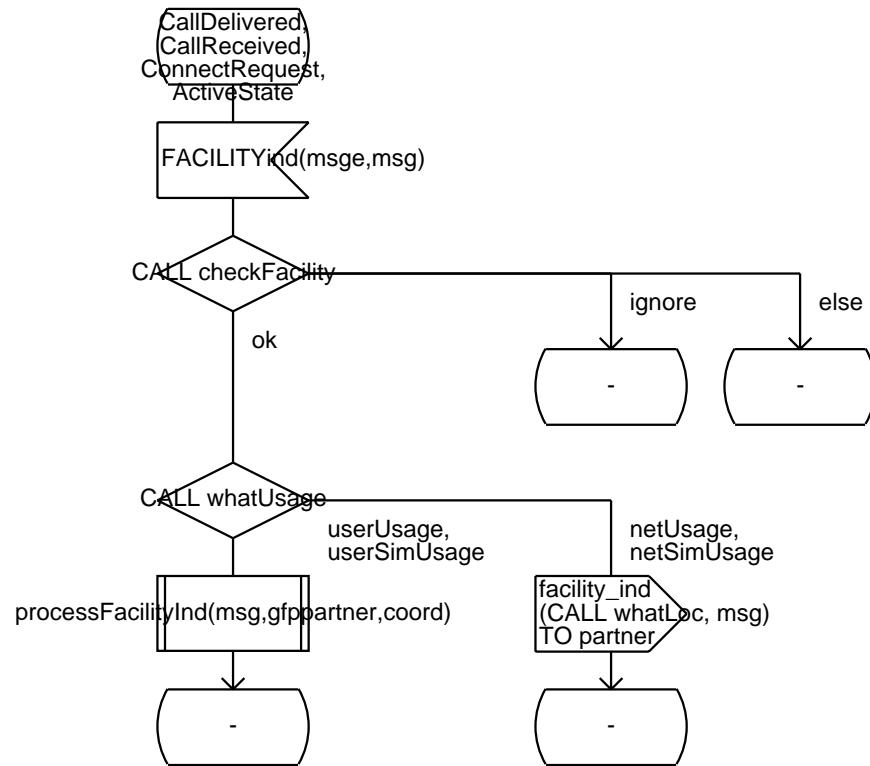


Annex B: CallControl_PT

Process Type <<Block Type Q2931ss_BT>> CallControl_PT

20(38)

INHERITS MsgHandler_PT;
 fpair partner PId, gfppartner PId, config InitTrunkConfig;

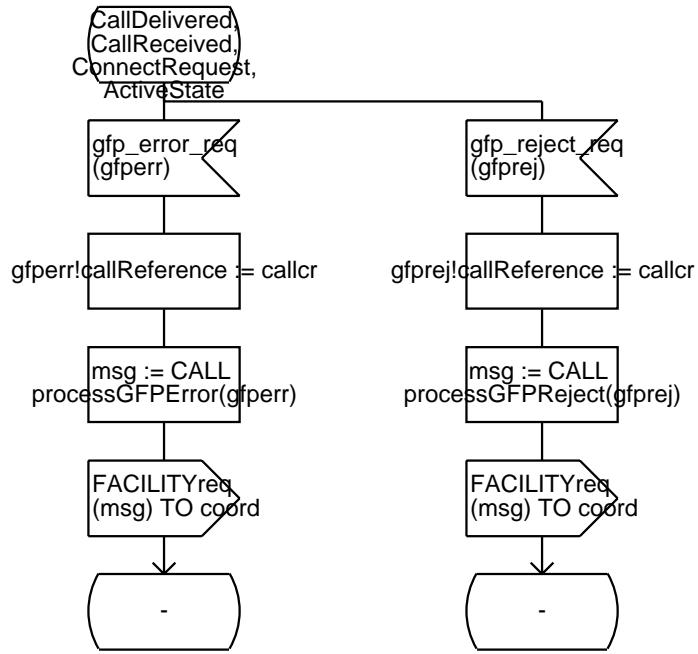


Annex B: CallControl_PT

Process Type <<Block Type Q2931ss_BT>> CallControl_PT

21(38)

INHERITS MsgHandler_PT;
 fpair partner PId, gfppartner PId, config InitTrunkConfig;

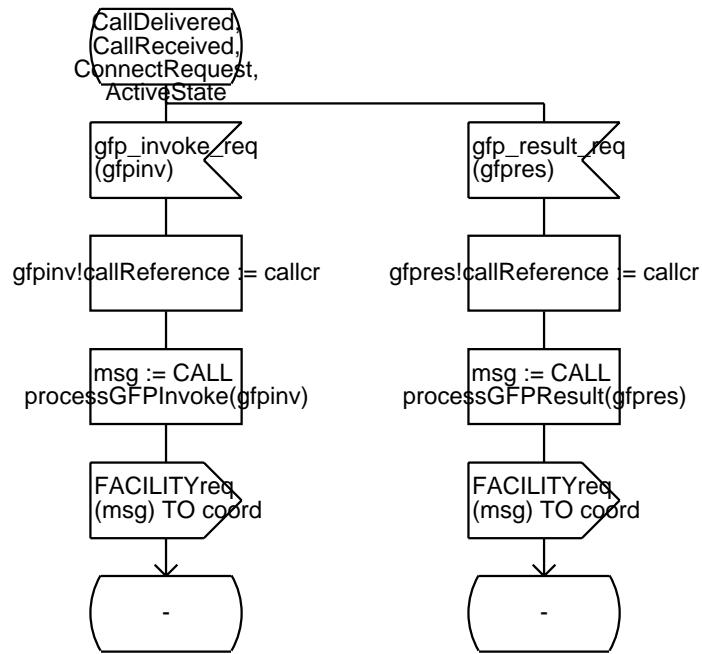


Annex B: CallControl_PT

Process Type <<Block Type Q2931ss_BT>> CallControl_PT

22(38)

INHERITS MsgHandler_PT;
 fpair partner PId, gfppartner PId, config InitTrunkConfig;

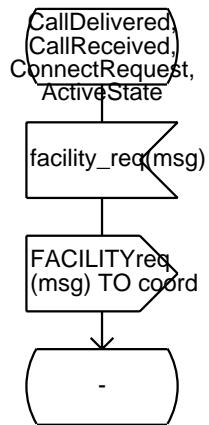


Annex B: CallControl_PT

Process Type <<Block Type Q2931ss_BT>> CallControl_PT

23(38)

INHERITS MsgHandler_PT;
fpar partner PId, gfppartner PId, config InitTrunkConfig;

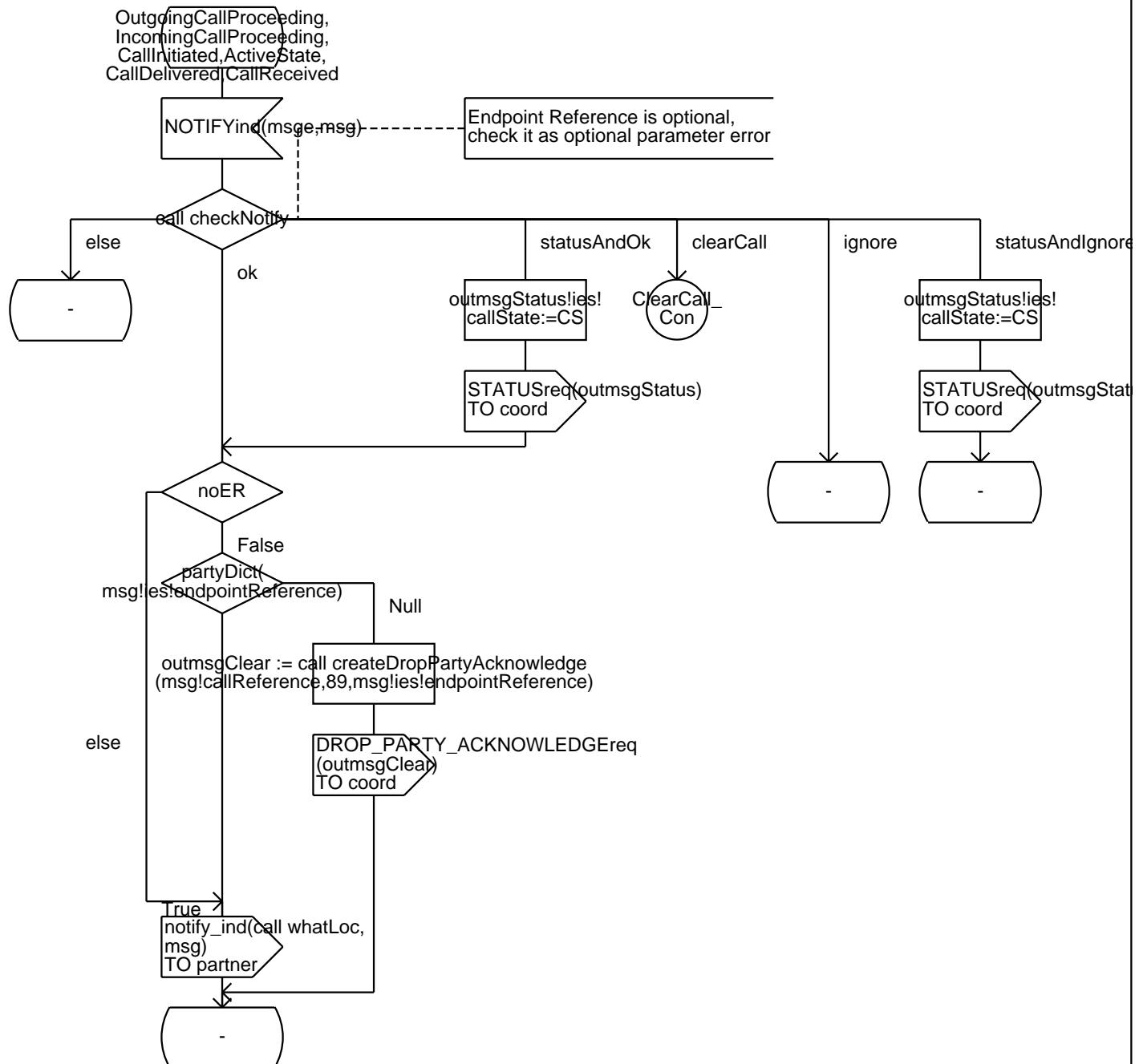


Annex B: CallControl_PT

Process Type <<Block Type Q2931ss_BT>> CallControl_PT

24(38)

INHERITS MsgHandler_PT;
 fpair partner PId, gfppartner PId, config InitTrunkConfig;

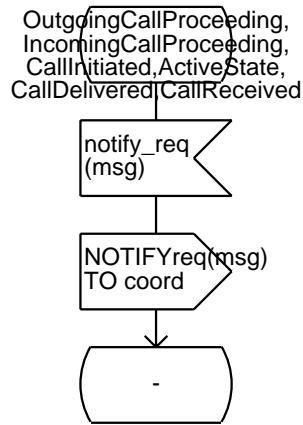


Annex B: CallControl_PT

Process Type <<Block Type Q2931ss_BT>> CallControl_PT

25(38)

INHERITS MsgHandler_PT;
fpar partner PId, gfppartner PId, config InitTrunkConfig;

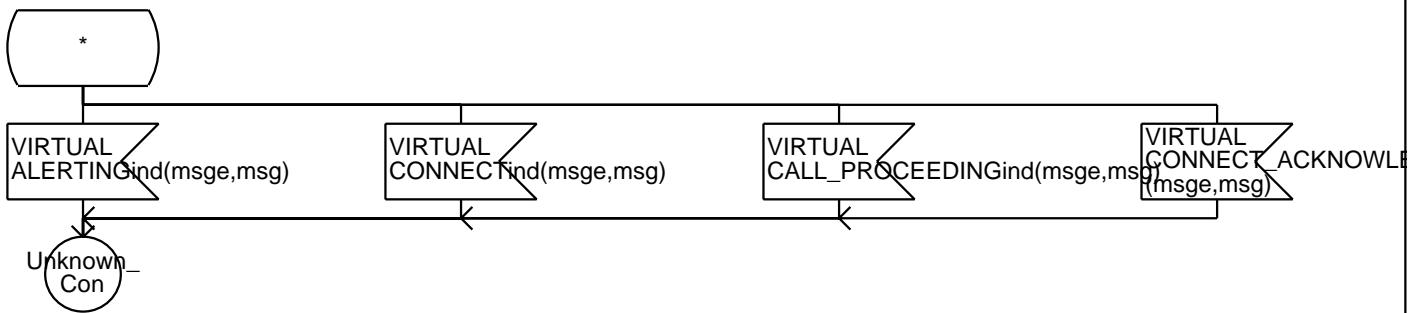


Annex B: CallControl_PT

Process Type <<Block Type Q2931ss_BT>> CallControl_PT

26(38)

INHERITS MsgHandler_PT;
fpar partner PId, gfppartner PId, config InitTrunkConfig;

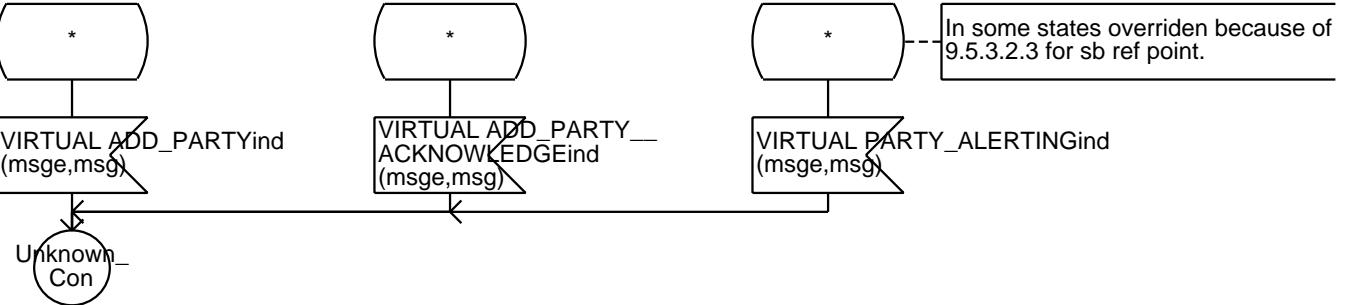


Annex B: CallControl_PT

Process Type <<Block Type Q2931ss_BT>> CallControl_PT

27(38)

INHERITS MsgHandler_PT;
fpar partner PId, gfppartner PId, config InitTrunkConfig;

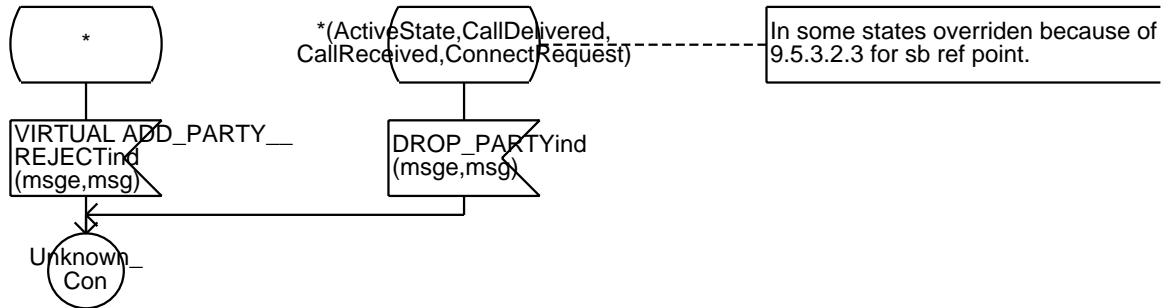


Annex B: CallControl_PT

Process Type <<Block Type Q2931ss_BT>> CallControl_PT

28(38)

INHERITS MsgHandler_PT;
fpair partner PId, gfppartner PId, config InitTrunkConfig;

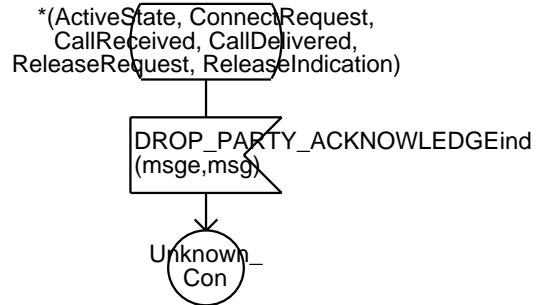


Annex B: CallControl_PT

Process Type <<Block Type Q2931ss_BT>> CallControl_PT

29(38)

INHERITS MsgHandler_PT;
fpar partner PId, gfppartner PId, config InitTrunkConfig;

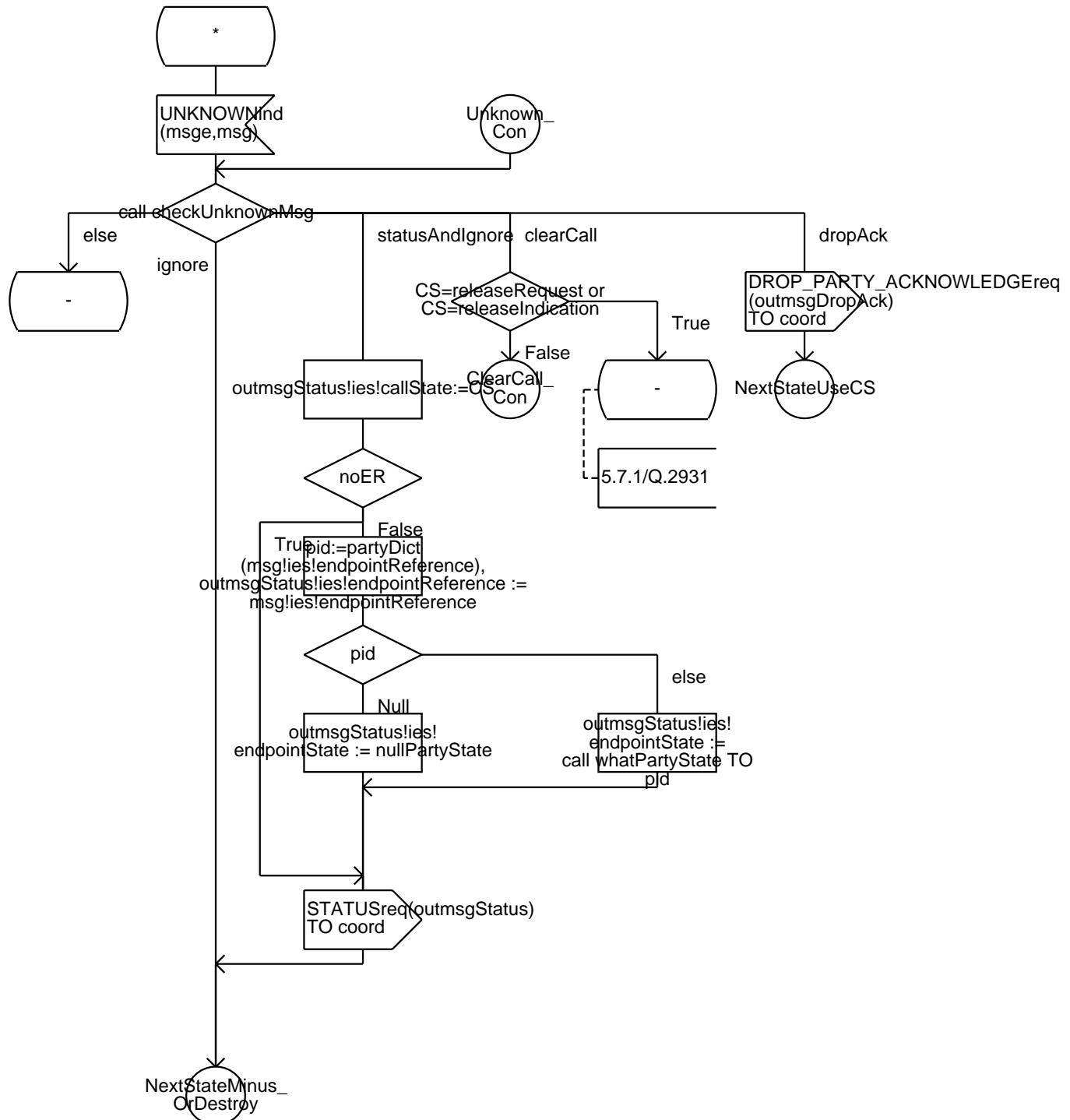


Annex B: CallControl_PT

Process Type <<Block Type Q2931ss_BT>> CallControl_PT

30(38)

INHERITS MsgHandler_PT;
 fpair partner PId, gfppartner PId, config InitTrunkConfig;

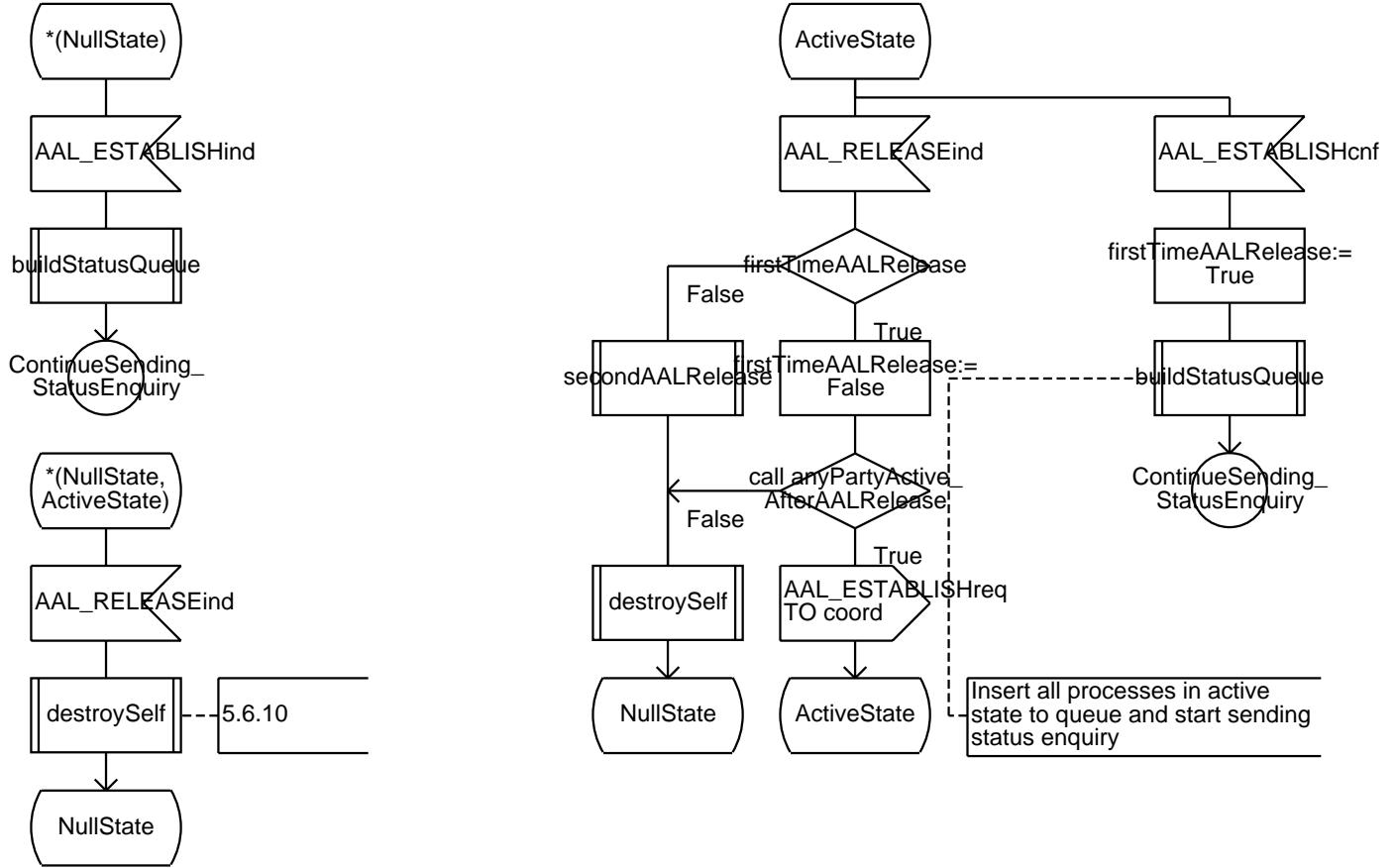


Annex B: CallControl_PT

Process Type <<Block Type Q2931ss_BT>> CallControl_PT

31(38)

INHERITS MsgHandler_PT;
 fpair partner PId, gfppartner PId, config InitTrunkConfig;

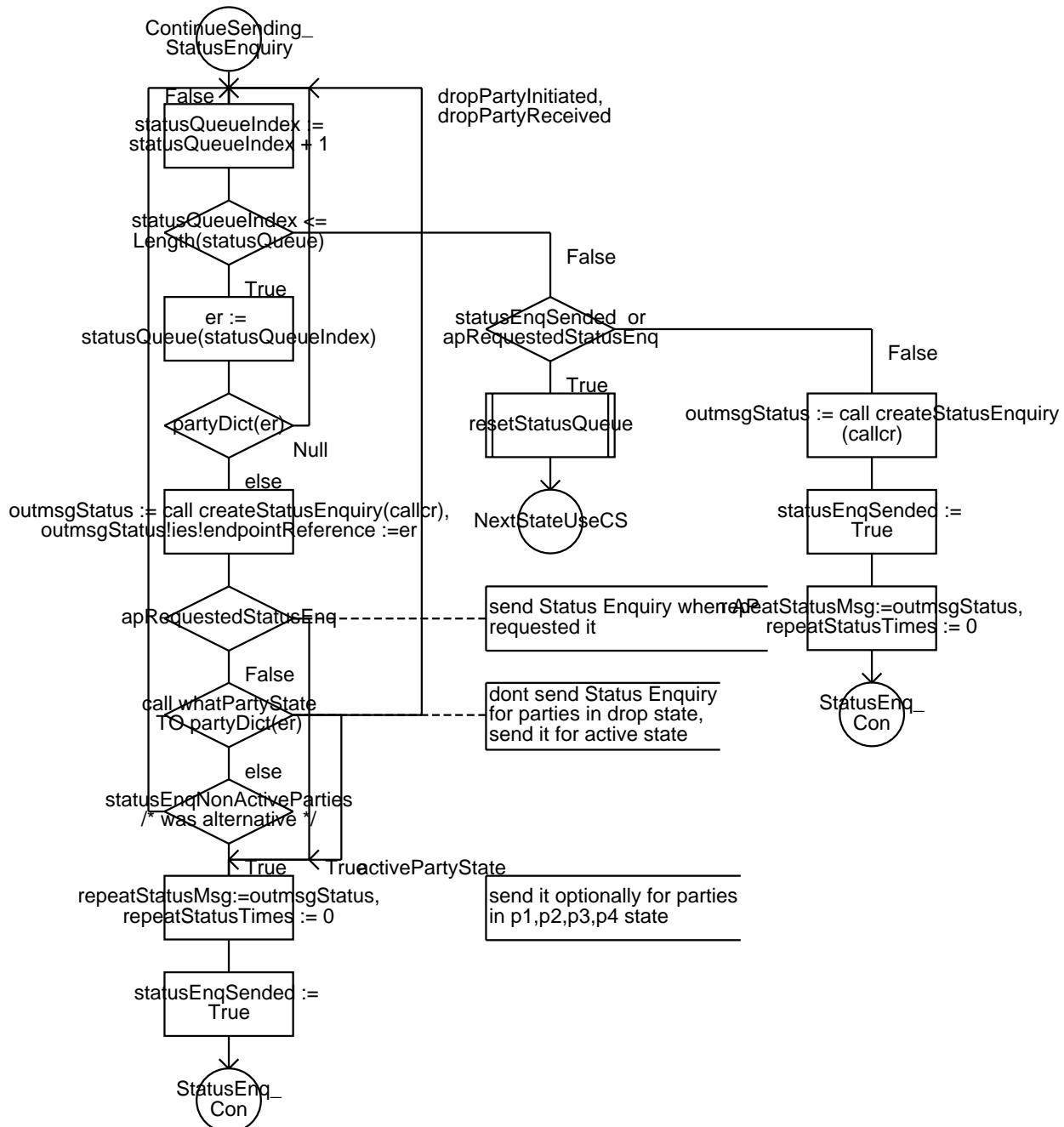


Annex B: CallControl_PT

Process Type <<Block Type Q2931ss_BT>> CallControl_PT

32(38)

INHERITS MsgHandler_PT;
 fpair partner PId, gfppartner PId, config InitTrunkConfig;

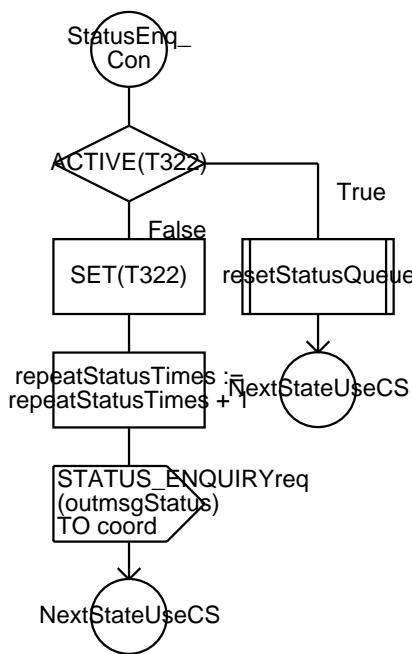


Annex B: CallControl_PT

Process Type <<Block Type Q2931ss_BT>> CallControl_PT

33(38)

INHERITS MsgHandler_PT;
fpar partner PId, gfppartner PId, config InitTrunkConfig;

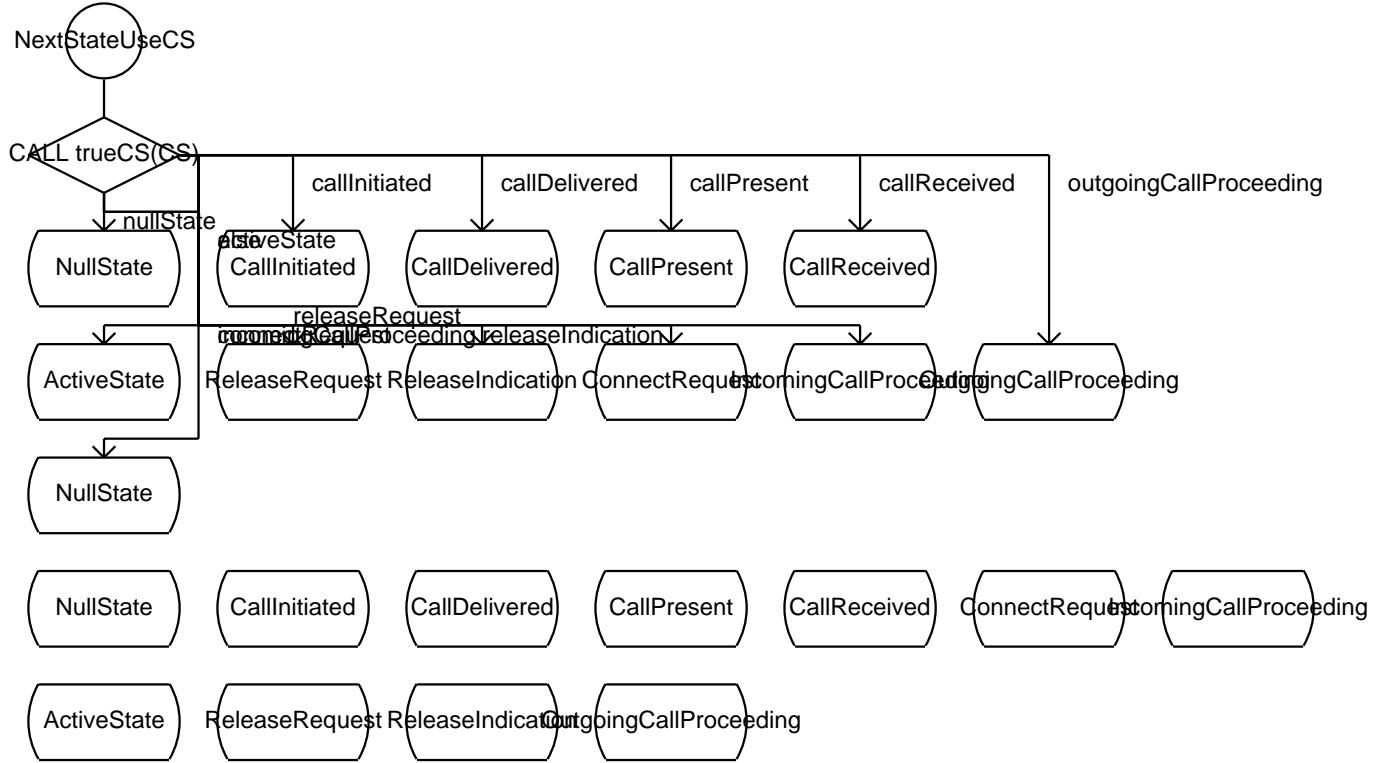


Annex B: CallControl_PT

Process Type <<Block Type Q2931ss_BT>> CallControl_PT

34(38)

INHERITS MsgHandler_PT;
 fpair partner PId, gfppartner PId, config InitTrunkConfig;

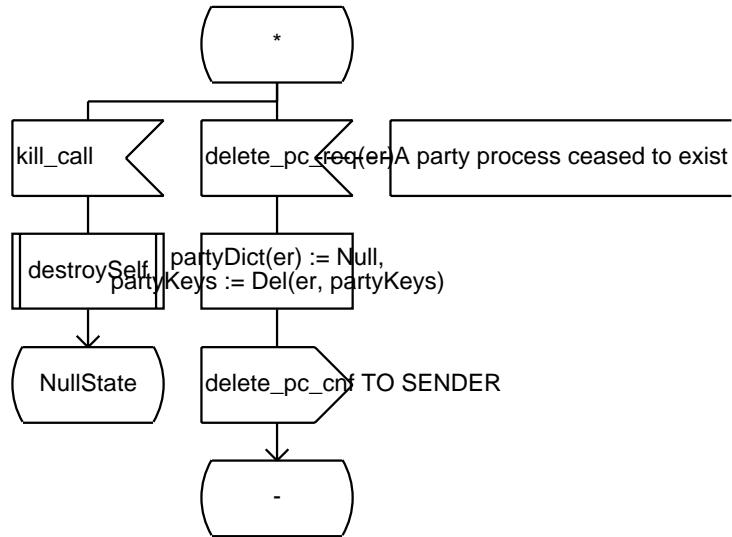


Annex B: CallControl_PT

Process Type <<Block Type Q2931ss_BT>> CallControl_PT

35(38)

INHERITS MsgHandler_PT;
 fpair partner PId, gfppartner PId, config InitTrunkConfig;

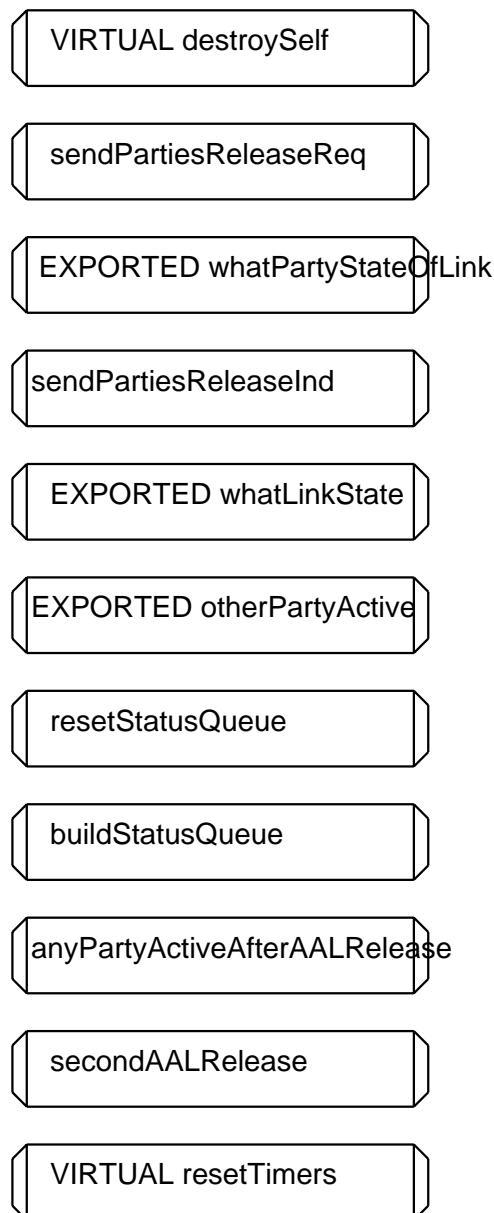


Annex B: CallControl_PT

Process Type <<Block Type Q2931ss_BT>> CallControl_PT

36(38)

INHERITS MsgHandler_PT;
fpar partner PId, gfppartner PId, config InitTrunkConfig;



Annex B: CallControl_PT

Process Type <<Block Type Q2931ss_BT>> CallControl_PT

37(38)

INHERITS MsgHandler_PT;
 fpair partner PId, gfppartner PId, config InitTrunkConfig;

<code>checkDropPartyER</code>	Check*ER checks a Q.2971 message in call control processes and returns other result than ok only if the action is clearCall, ignore, ignoreAndStatus, dropParty. These actions (clearCall,...) shall be executed immedeately in call control process. The original value of CheckResultType (ok, statusAndOk,...) is written to variable preres. This result in preres should than transmitted to party control process and the action (i.e. sending status) executed there.
<code>checkDropPartyAcknowledgeER</code>	
<code>checkRelease</code>	
<code>checkReleaseComplete</code>	
<code>checkNotify</code>	
<code>checkUnknownMsg</code>	
<code>VIRTUAL whatLoc</code>	returns location who generated the current message (trunk, incoming or outgoing), needed by application to determine from which message arrived, redefine in derived process types.
<code>containsCauseValue</code>	checks whether a message contains a specific cause value
<code>checkAddPartyER</code>	
<code>omitER</code>	empties field of endpoint reference in a message
<code>trueCS</code>	User side processes are modelled by the same processes (ICC,OCC) as the network side, but the names of the states are different. In this specification the network side names are used during state change. This procedure converts the call state names of user and network side to each other. This function should be used before assigning a new value to CS and before using the value CS to go into a state. The value CS is used to answer with the proper call state in STATUS messages. User side Incoming Call Control = Network side Incoming Call Control process User side Outgoing Call Control = Network side Outgoing Call Control process

Annex B: CallControl_PT

Process Type <<Block Type Q2931ss_BT>> CallControl_PT

38(38)

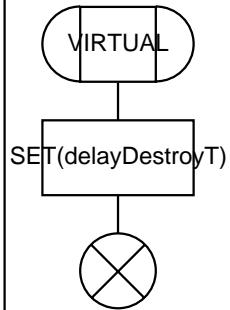
```
INHERITS MsgHandler_PT;
fpar partner PId, gfppartner PId, config InitTrunkConfig;
```

```
IMPORTED PROCEDURE whatPartyState;
returns EndpointState;
```

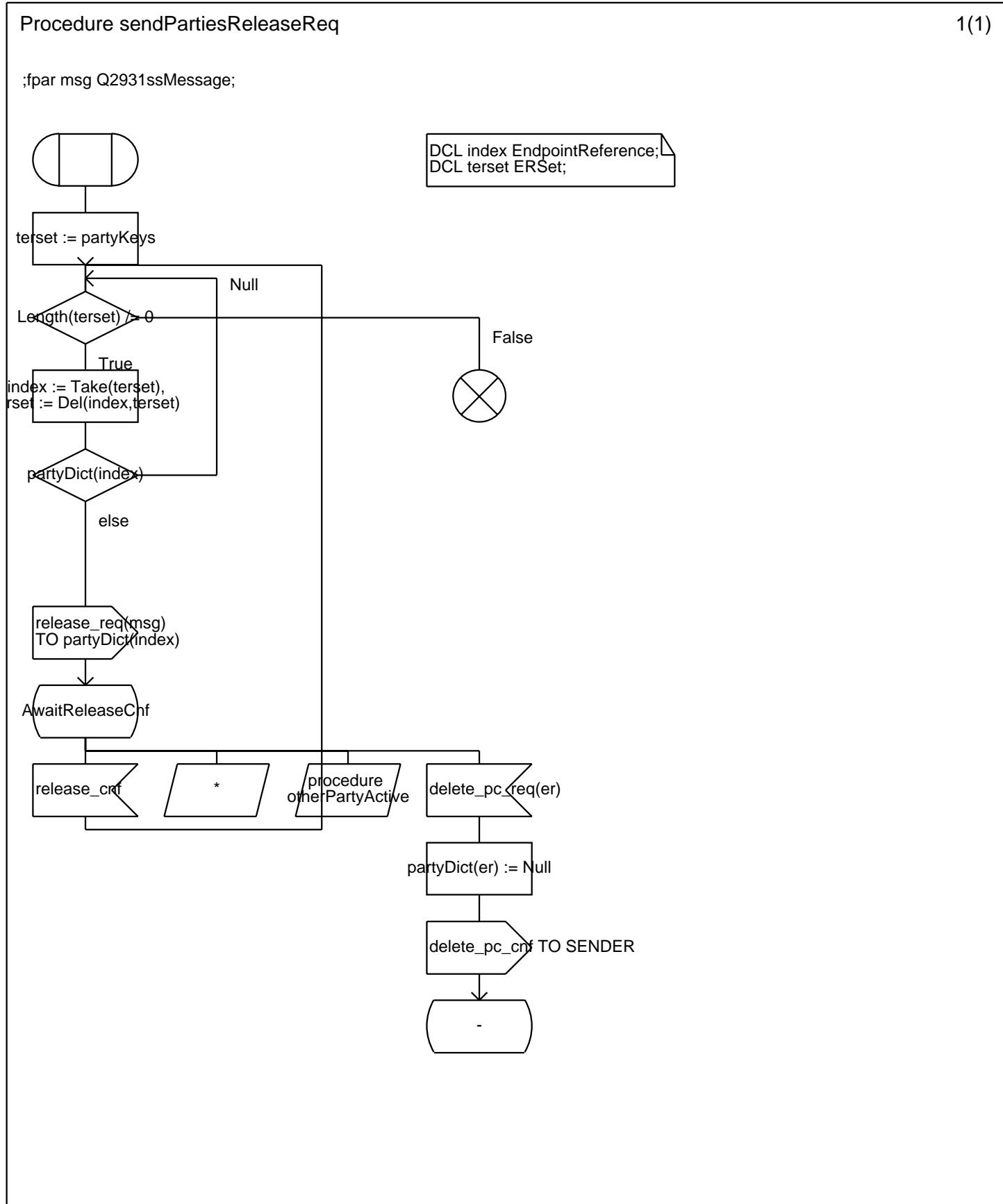
Annex B: destroySelf

VIRTUAL Procedure <<Process Type CallControl_PT>> destroySelf

1(1)



Annex B: sendPartiesReleaseReq

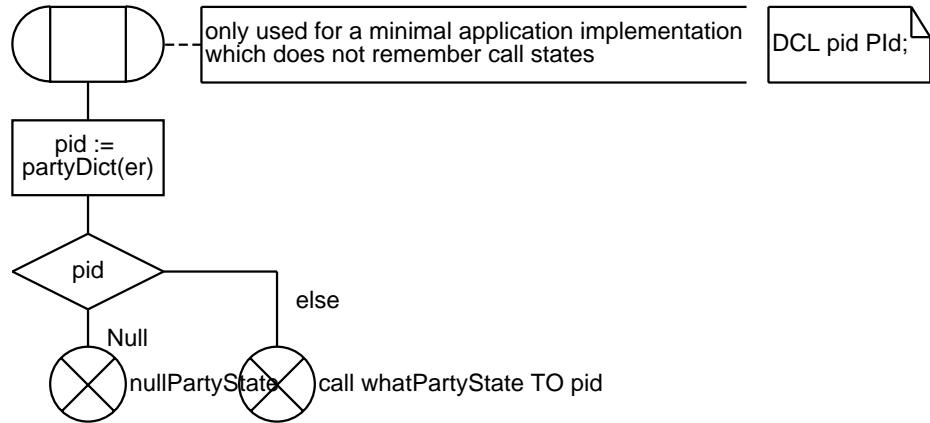


Annex B: whatPartyStateOfLink

EXPORTED Procedure whatPartyStateOfLink

1(1)

```
:fpar er EndpointReference;  
returns EndpointState;
```



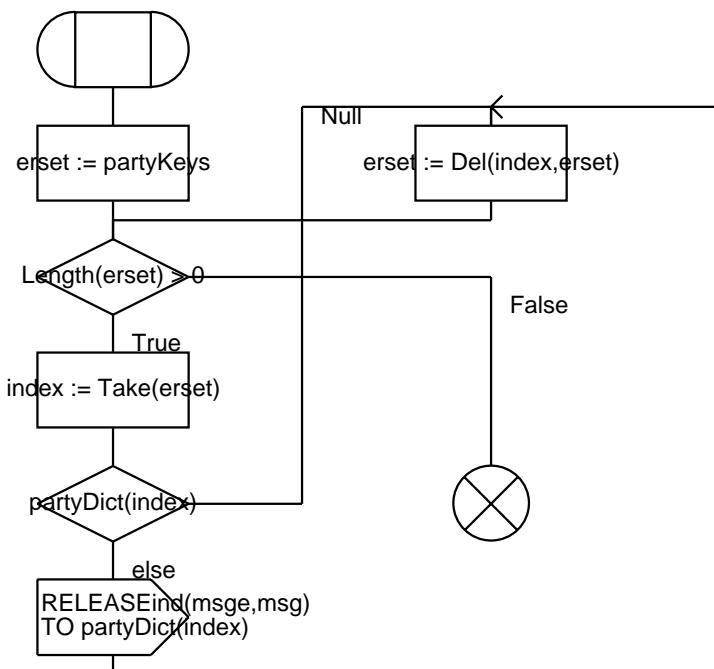
Annex B: sendPartiesReleaseInd

Procedure sendPartiesReleaseInd

1(1)

```
:fpar msge MsgError, msg Q2931ssMessage;
```

DCL index EndpointReference;
 DCL erset ERSet;

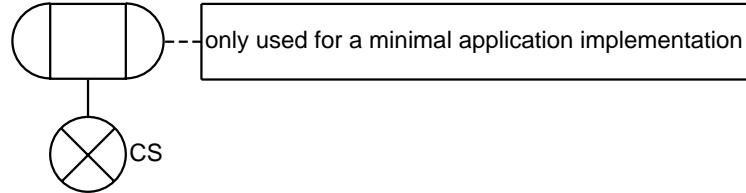


Annex B: whatLinkState

EXPORTED Procedure whatLinkState

1(1)

;returns CallState;



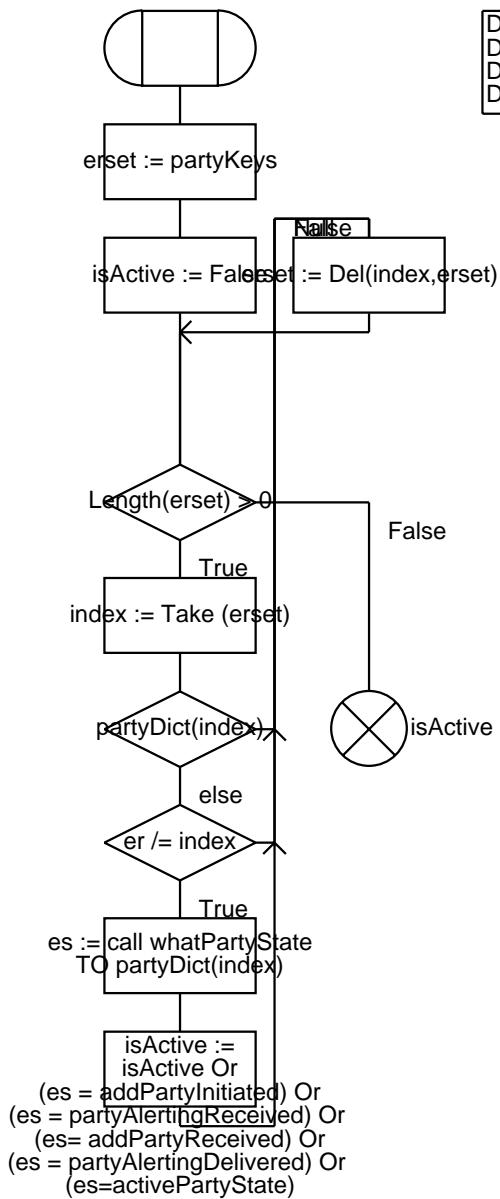
Annex B: otherPartyActive

EXPORTED Procedure otherPartyActive

1(1)

:fpar er EndpointReference;
returns Boolean;

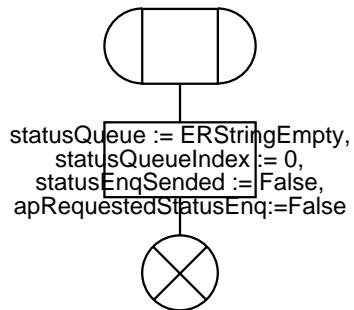
DCL isActive Boolean;
DCL index EndpointReference;
DCL es EndpointState;
DCL erset ERSet ;



Annex B: resetStatusQueue

Procedure resetStatusQueue

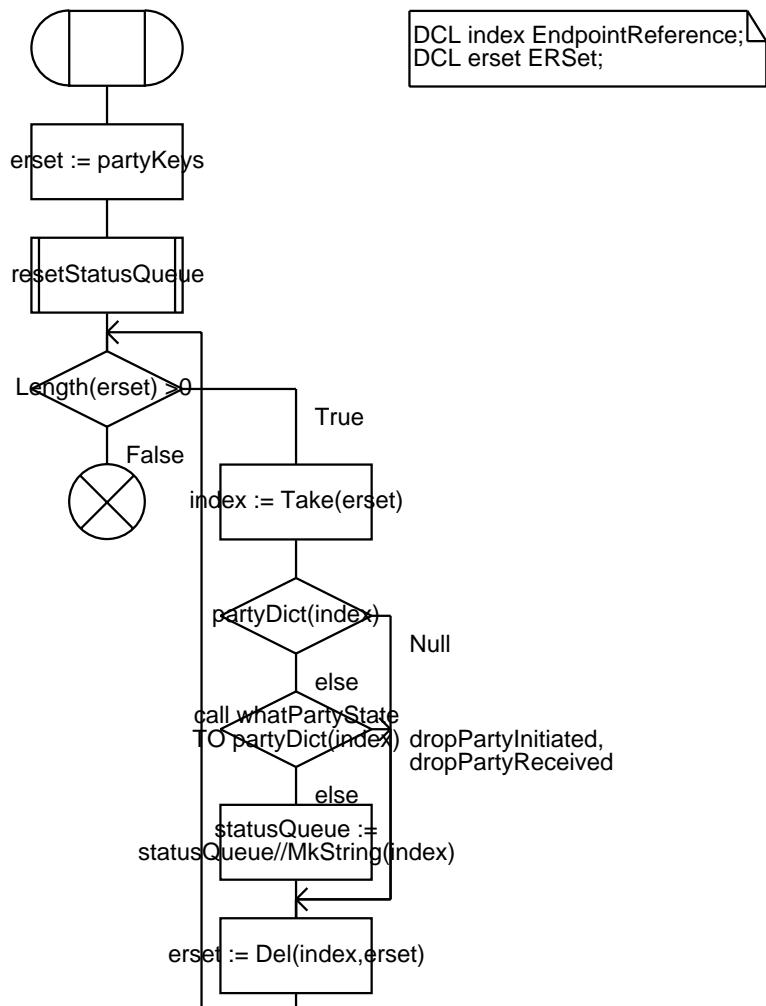
1(1)



Annex B: buildStatusQueue

Procedure buildStatusQueue

1(1)

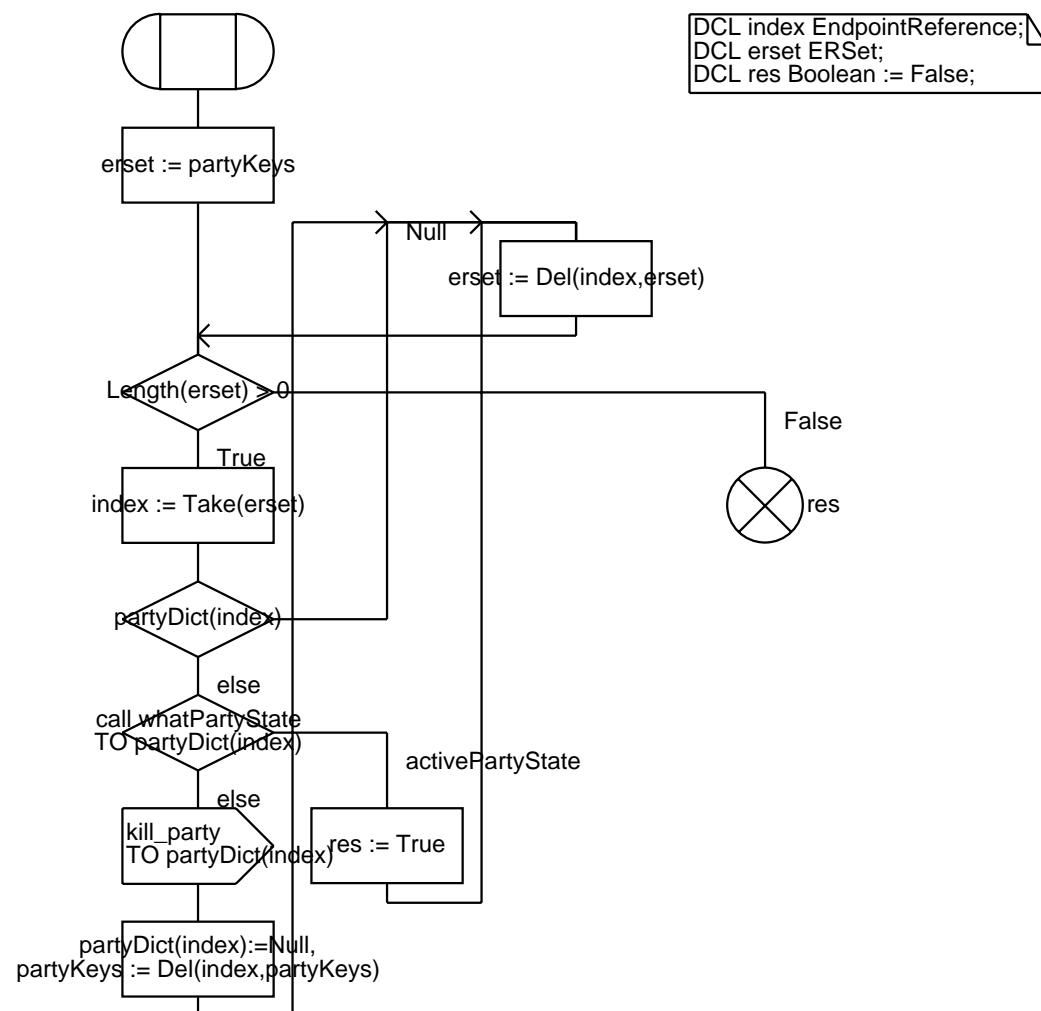


Annex B: anyPartyActiveAfterAALRelease

Procedure anyPartyActiveAfterAALRelease

1(1)

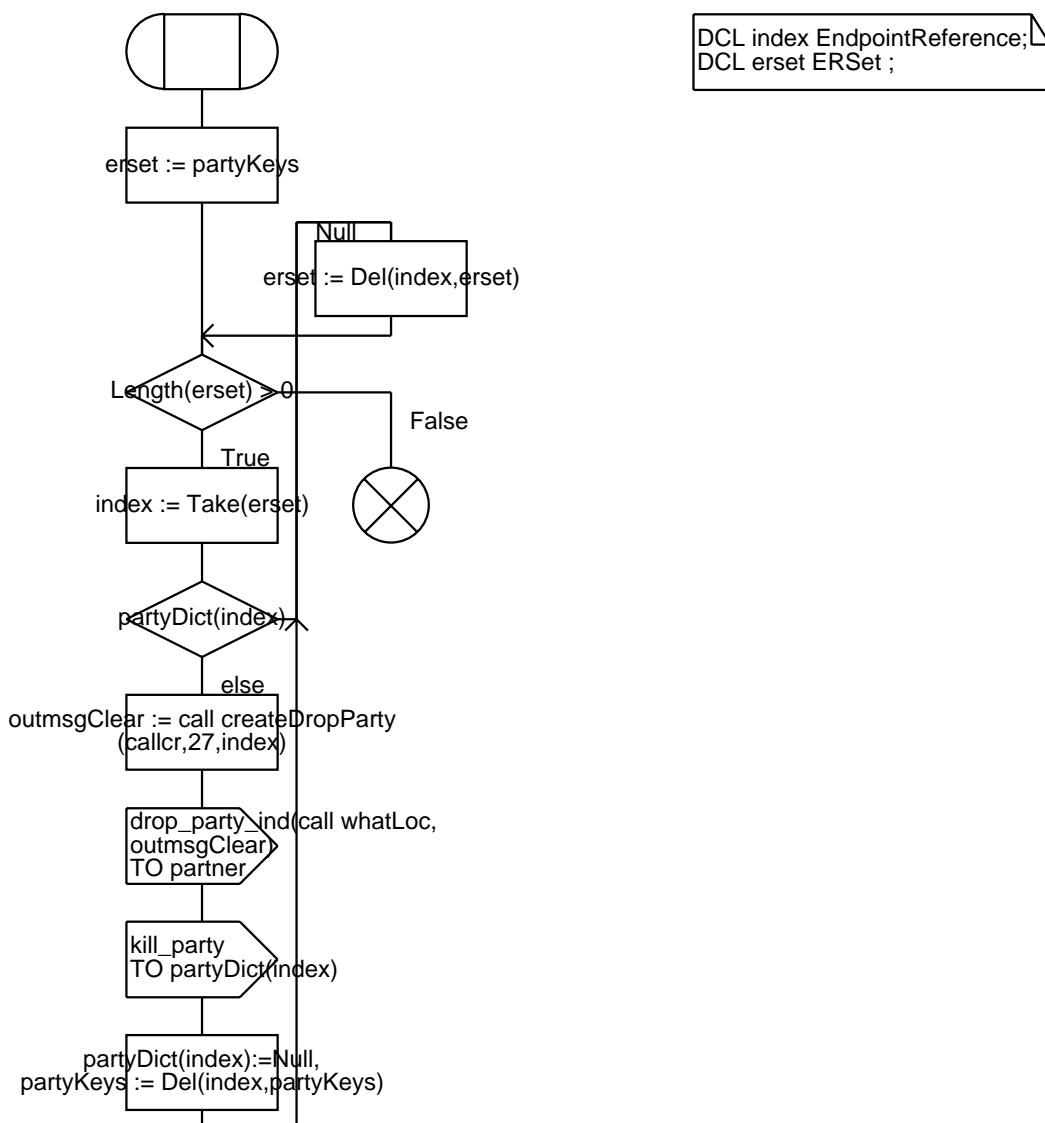
;returns Boolean;



Annex B: secondAALRelease

Procedure secondAALRelease

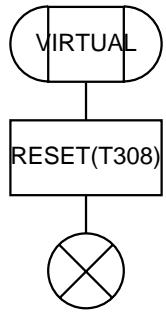
1(1)



Annex B: resetTimers

Virtual Procedure <<Process Type CallControl_PT>> resetTimers

1(1)



Annex B: checkDropPartyER

Procedure checkDropPartyER

1(1)

;FPAR in/out preres CheckResultTypeStatus;RETURNS res CheckResultType;

```
/*#include '../sdl/CheckdropPartyER.sdl' */
```

Annex B: checkDropPartyAcknowledgeER

Procedure checkDropPartyAcknowledgeER

1(1)

;FPAR in/out preres CheckResultTypeStatus,in firstdropping Boolean;RETURNS res CheckResultType;

```
/*#include '../sdl/CheckdropPartyAcknowledgeER.sdl' */
```

Annex B: checkRelease

Procedure checkRelease

1(1)

```
:RETURNS res CheckResultType;
```

```
/*#include '../sdl/Checkrelease.sdl' */
```

Annex B: checkReleaseComplete

Procedure checkReleaseComplete

1(1)

;RETURNS res CheckResultType;

```
/*#include '../sdl/CheckreleaseComplete.sdl' */
```

Annex B: checkNotify

Procedure checkNotify

1(1)

;RETURNS res CheckResultType;

```
/*#include '../sdl/Checknotify.sdl' */
```

Annex B: checkUnknownMsg

Procedure checkUnknownMsg

1(1)

;RETURNS res CheckResultType;

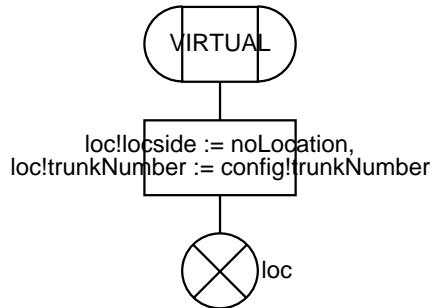
```
/*#include '../sdl/Checkunknown.sdl' */
```

Annex B: whatLoc

Virtual Procedure <<Process Type CallControl_PT>> whatLoc

1(1)

;RETURNS loc Location;



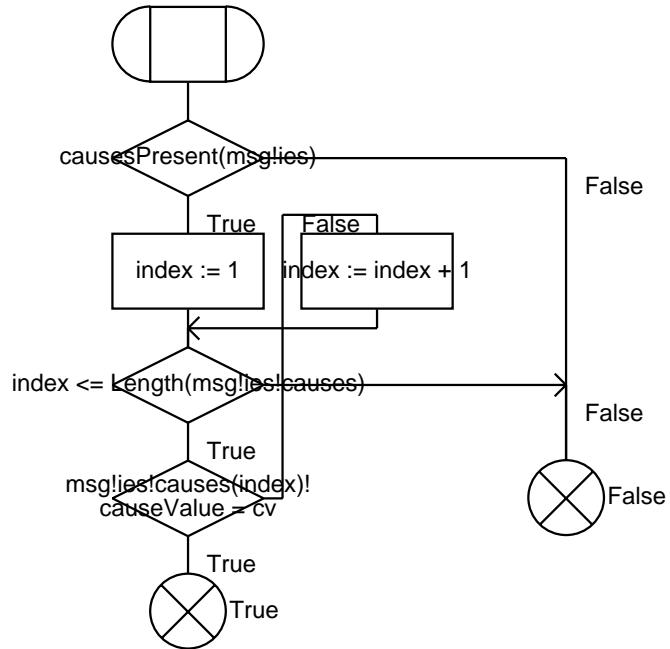
Annex B: containsCauseValue

Procedure containsCauseValue

1(1)

```
:FPAR msg Q2931ssMessage, cv CauseValue;
RETURNS Boolean;
```

DCL index Integer;



Annex B: checkAddPartyER

Procedure checkAddPartyER

1(1)

;FPAR in/out preres CheckResultTypeStatus;RETURNS res CheckResultType;

/*#include '../sdl/CheckaddPartyER.sdl' */

Annex B: omitER

Procedure omitER

1(1)

```
:FPAR msg Q2931ssMessage;
RETURNS Q2931ssMessage;
```

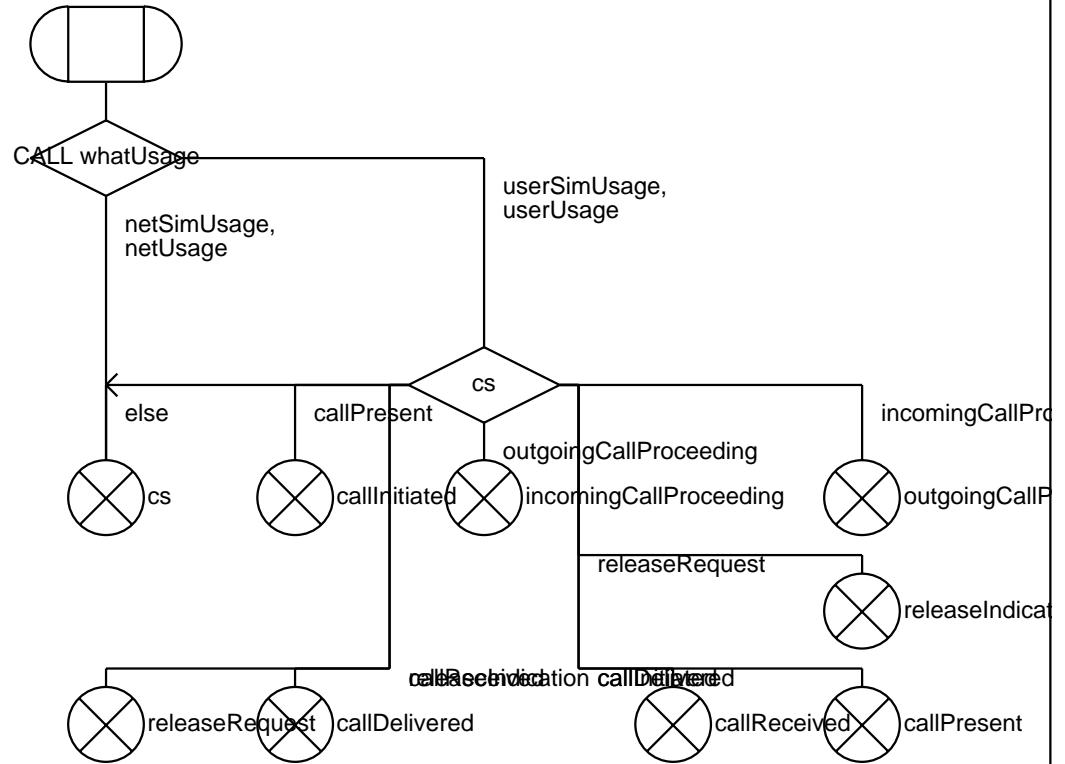
```
/*#include '../sdl/omitER.sdl' */
```

Annex B: trueCS

Procedure trueCS

1(1)

;FPAR cs CallState; RETURNS CallState;



Annex B: IORestart_PT

Process Type <<Block Type Q2931ss_BT>> IORestart_PT

1(9)

INHERITS MsgHandler_PT;
fpar partner PId, coord PId, config InitTrunkConfig;

```
DCL CS CallState; /* Call State */  
DCL restartTimes Integer := 0;  
/* how many restarts already sent */  
DCL restartTimesMax Integer := 2;  
/* maximum of restarts sent */  
DCL restartmsg Q2931ssMessage;
```

```
TIMER T316 := 300000; /* 3 min */  
TIMER T317 := 300000; /* less than 3 min */
```

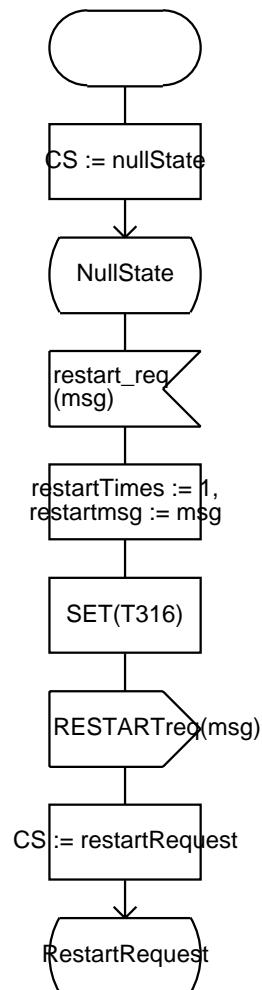
Annex B: IORestart_PT



Process Type <<Block Type Q2931ss_BT>> IORestart_PT

2(9)

```
INHERITS MsgHandler_PT;
fpar partner PId, coord PId, config InitTrunkConfig;
```

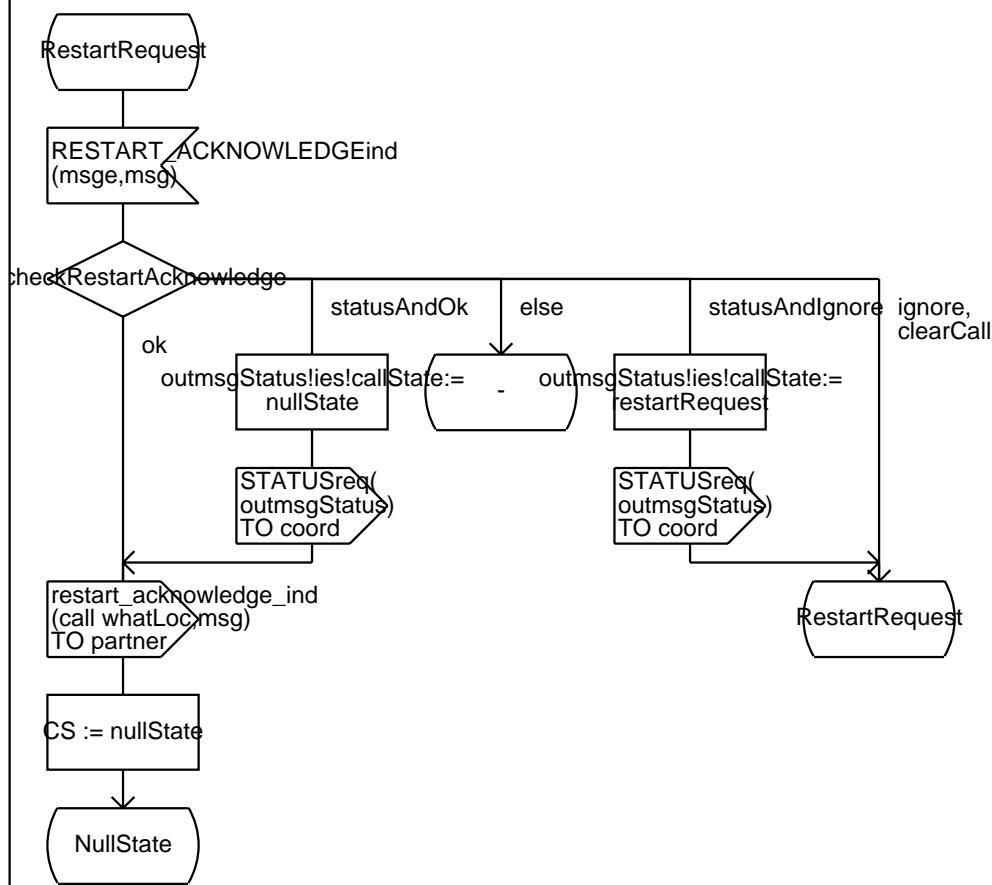


Annex B: IORestart_PT

Process Type <<Block Type Q2931ss_BT>> IORestart_PT

3(9)

INHERITS MsgHandler_PT;
 fpar partner PId, coord PId, config InitTrunkConfig;

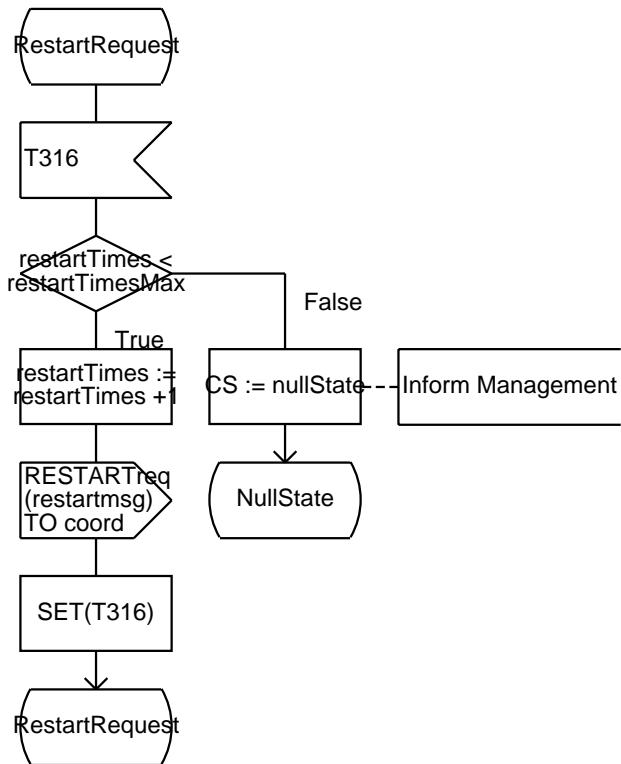


Annex B: IORestart_PT

Process Type <<Block Type Q2931ss_BT>> IORestart_PT

4(9)

INHERITS MsgHandler_PT;
 fpair partner PId, coord PId, config InitTrunkConfig;

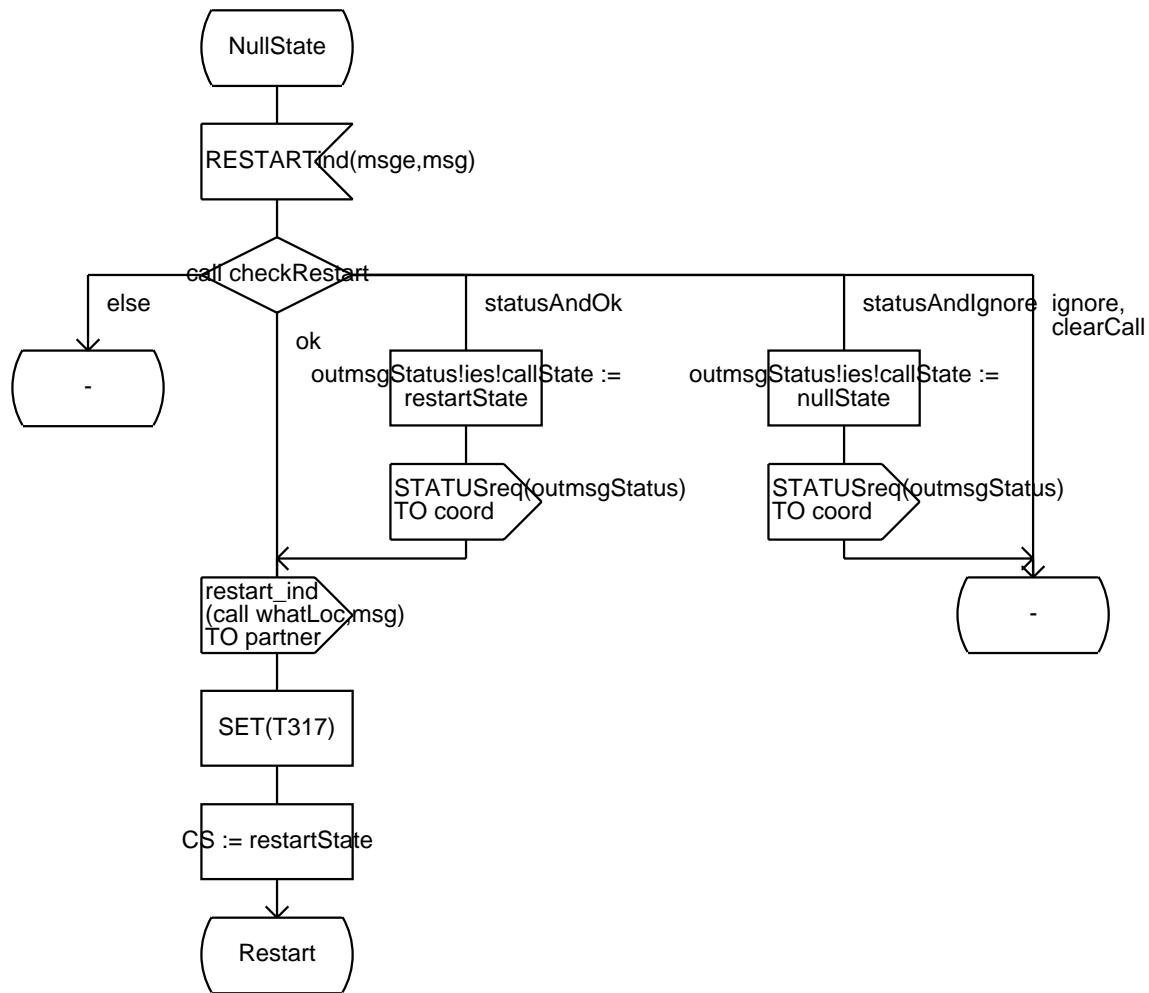


Annex B: IORestart_PT

Process Type <<Block Type Q2931ss_BT>> IORestart_PT

5(9)

INHERITS MsgHandler_PT;
 fpar partner Plid, coord Plid, config InitTrunkConfig;

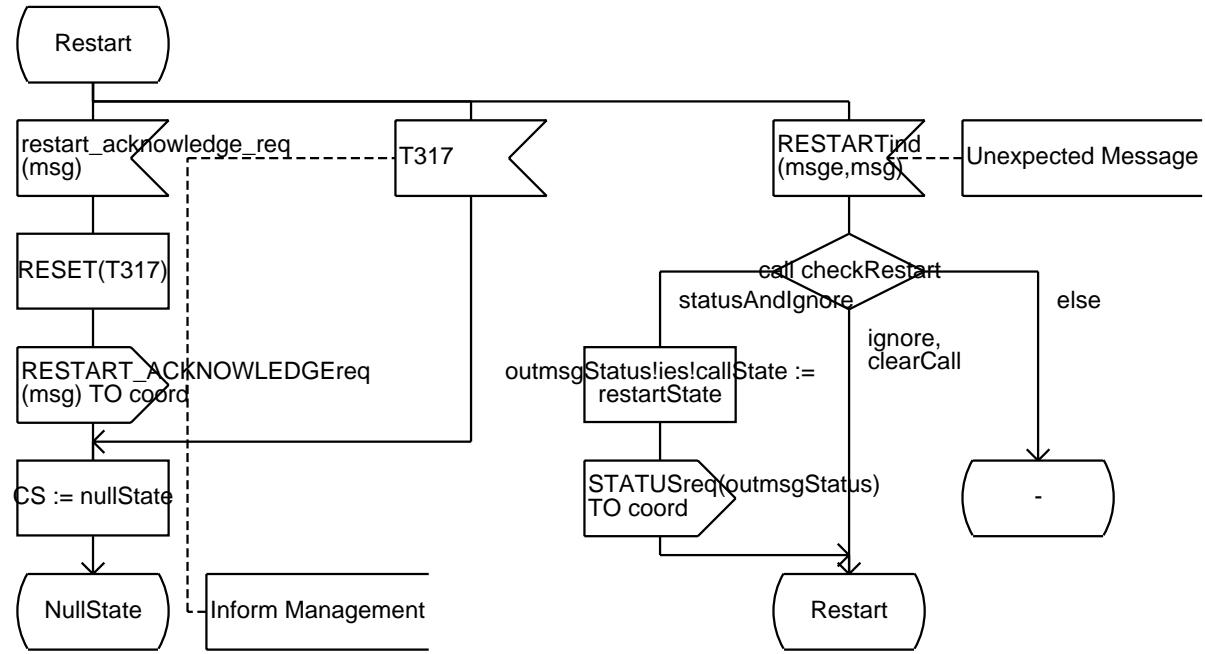


Annex B: IORestart_PT

Process Type <<Block Type Q2931ss_BT>> IORestart_PT

6(9)

INHERITS MsgHandler_PT;
 fpar partner PId, coord PId, config InitTrunkConfig;

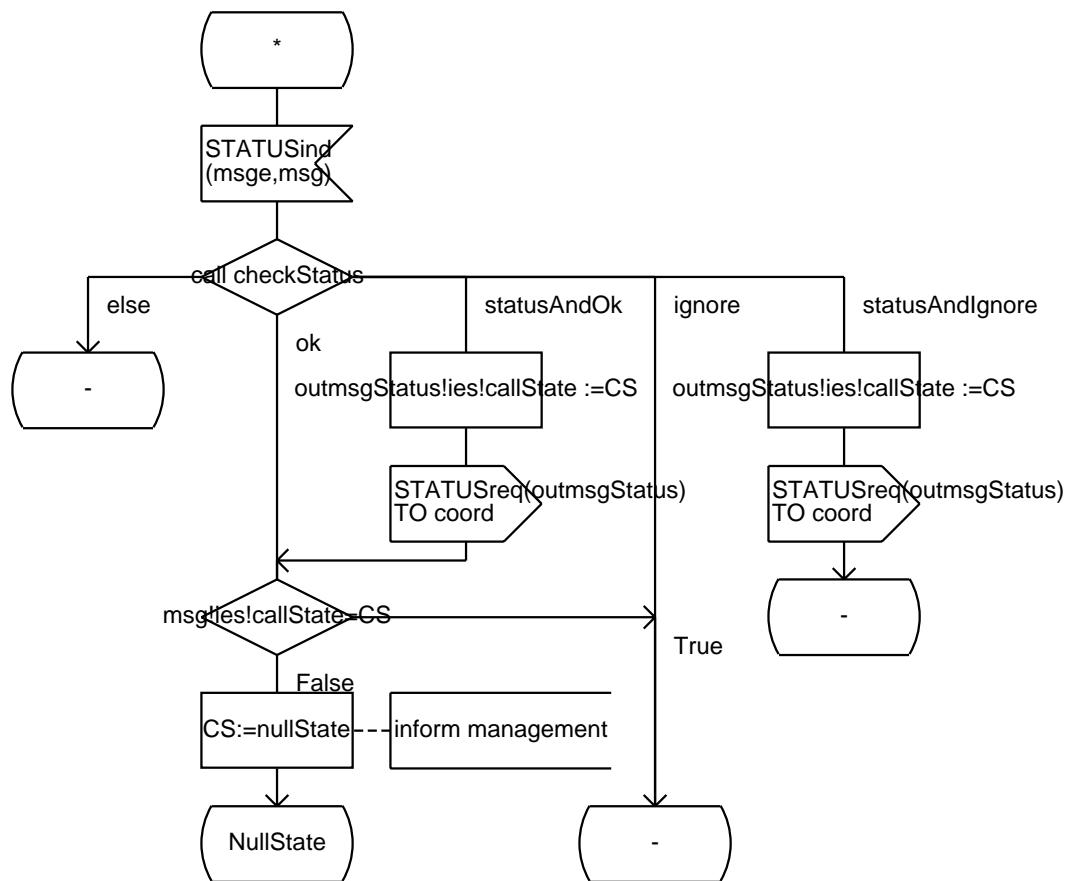


Annex B: IORestart_PT

Process Type <<Block Type Q2931ss_BT>> IORestart_PT

7(9)

INHERITS MsgHandler_PT;
 fpar partner PId, coord PId, config InitTrunkConfig;

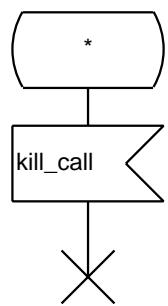


Annex B: IORestart_PT

Process Type <<Block Type Q2931ss_BT>> IORestart_PT

8(9)

INHERITS MsgHandler_PT;
fpar partner PId, coord PId, config InitTrunkConfig;



Annex B: IORestart_PT

Process Type <>Block Type Q2931ss_BT>> IORestart_PT

9(9)

INHERITS MsgHandler_PT;
fpar partner PId, coord PId, config InitTrunkConfig;

EXPORTED whatRestartState

whatLoc

checkRestartAcknowledge

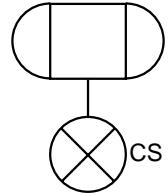
checkRestart

Annex B: whatRestartState

EXPORTED Procedure whatRestartState

1(1)

;returns CallState;

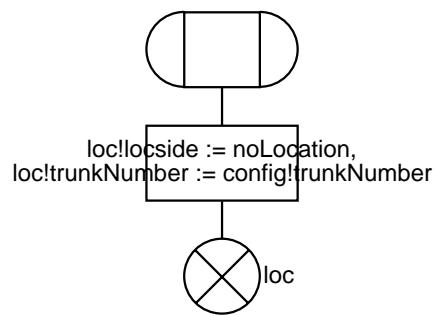


Annex B: whatLoc

Procedure <<Process Type IORestart_PT>> whatLoc

1(1)

;RETURNS loc Location;



Annex B: checkRestartAcknowledge

Procedure checkRestartAcknowledge

1(1)

;RETURNS res CheckResultType;

```
/*#include '../sdl/CheckrestartAcknowledge.sdl' */
```

Annex B: checkRestart

Procedure checkRestart

1(1)

;RETURNS res CheckResultType;

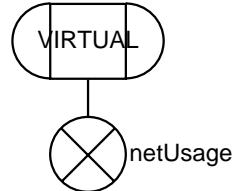
/*#include '../sdl/Checkrestart.sdl' */

Annex B: whatUsage

Virtual Procedure <<Block Type Q2931ss_BT>> whatUsage

1(1)

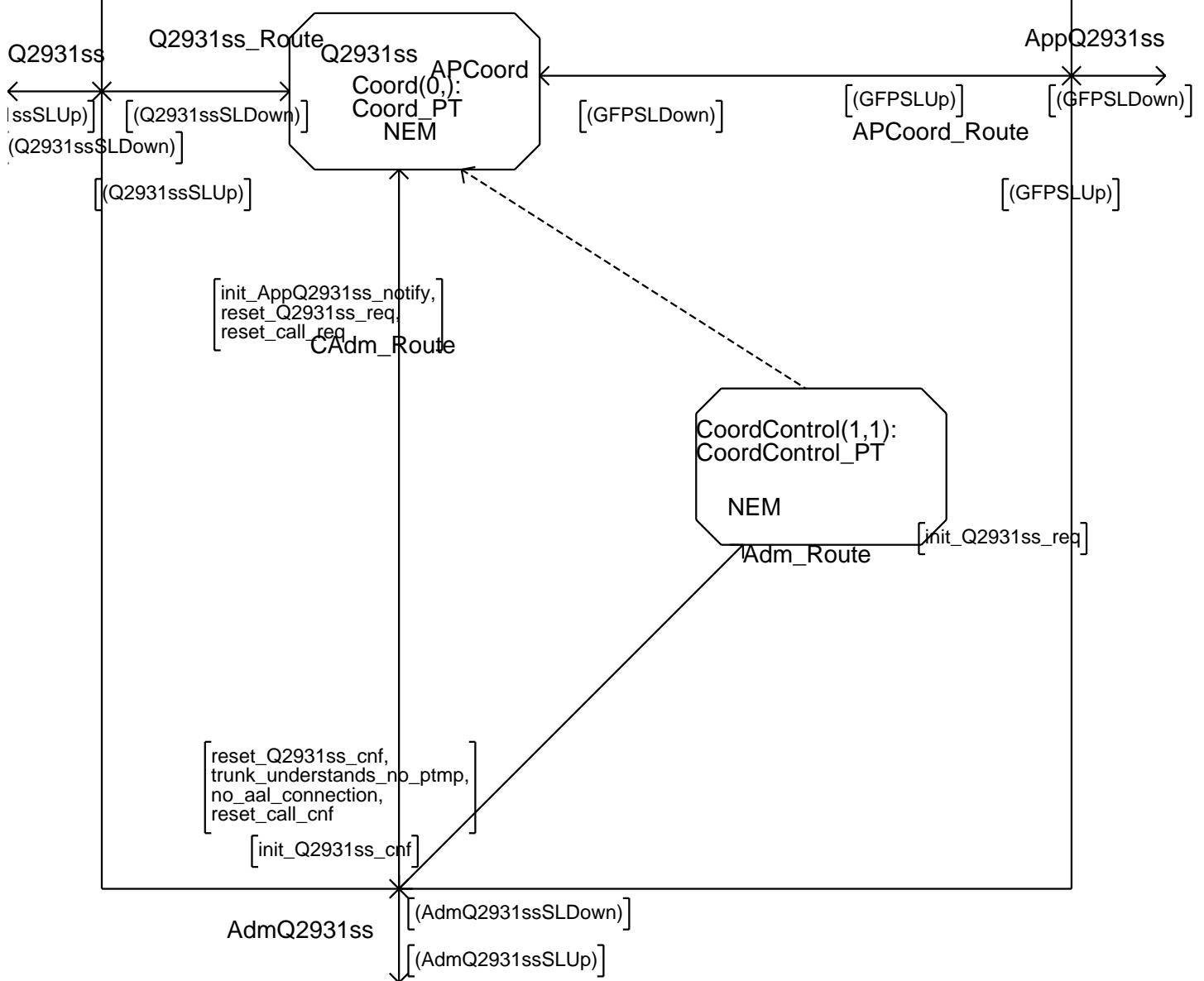
:RETURNS UsageType;



Annex B: Q2932_BT

Block Type Q2932_BT

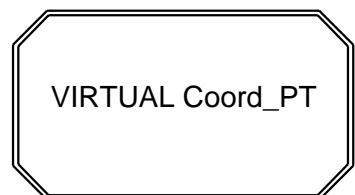
1(2)



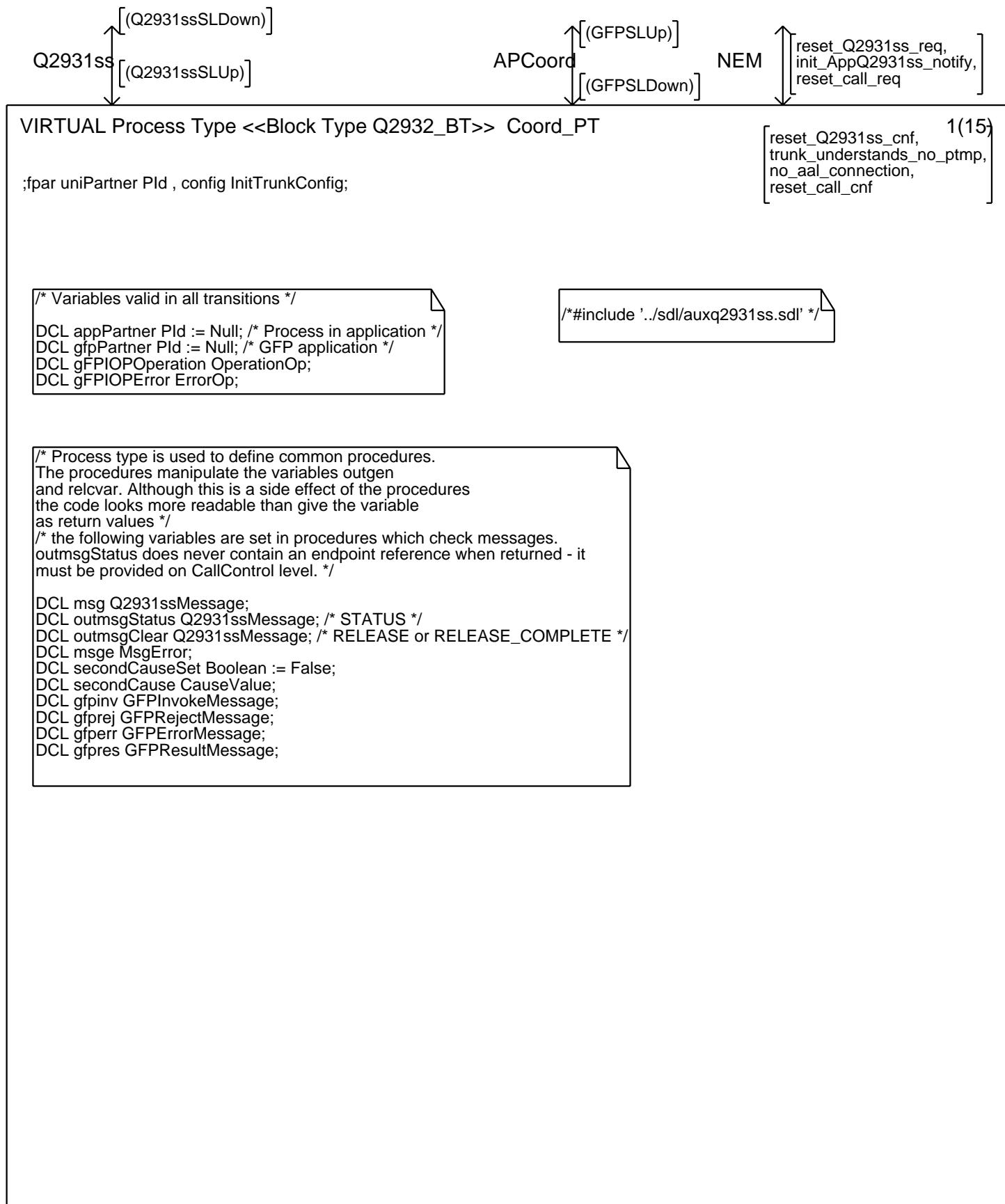
Annex B: Q2932_BT

Block Type Q2932_BT

2(2)



Annex B: Coord_PT



Annex B: Coord_PT

VIRTUAL Process Type <<Block Type Q2932_BT>> Coord_PT

2(15)

```
;fpar uniPartner PId , config InitTrunkConfig;
```

```
/* Variable valid only during one transition */  
DCL pid PId;  
DCL data AALData;  
DCL relcause AALRelease;
```

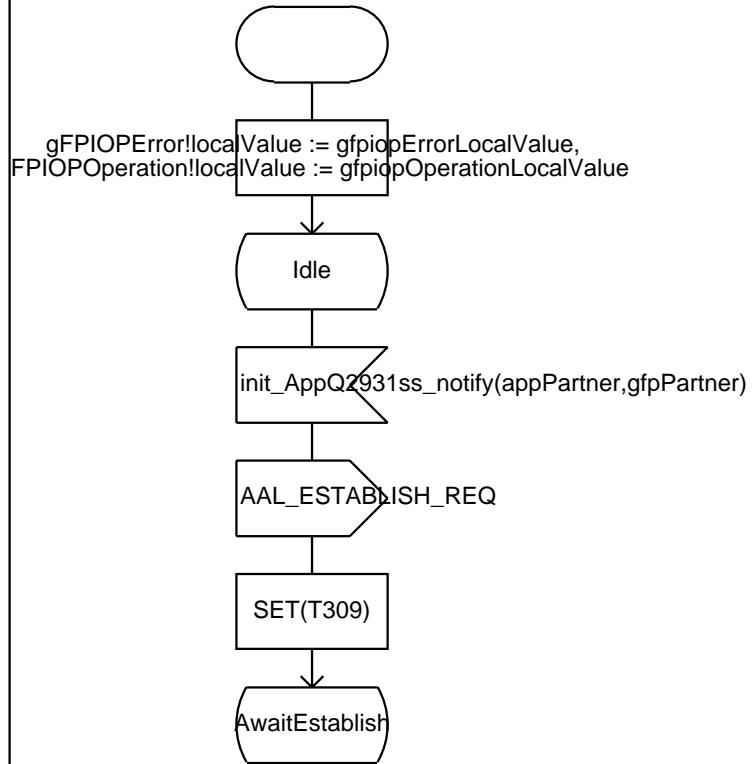
```
TIMER T309 := 60000;
```

Annex B: Coord_PT

VIRTUAL Process Type <<Block Type Q2932_BT>> Coord_PT

3(15)

;fpar uniPartner PId , config InitTrunkConfig;

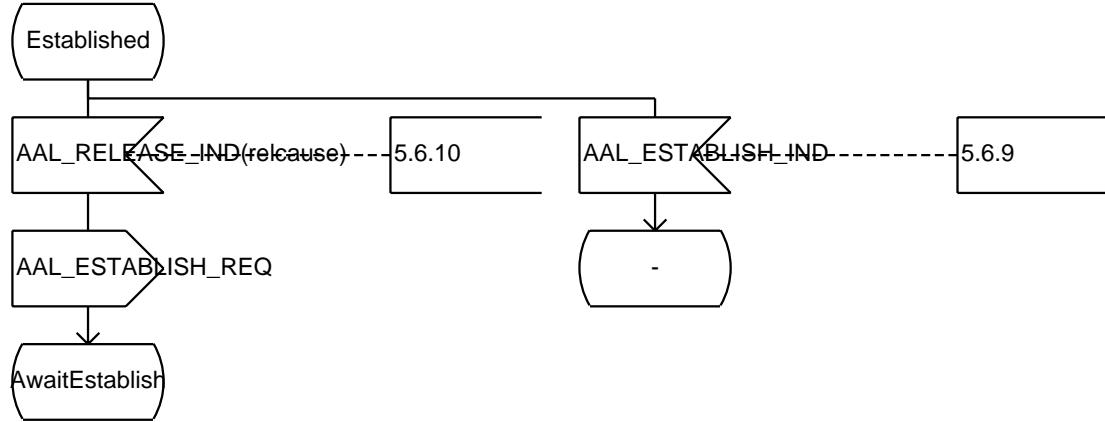


Annex B: Coord_PT

VIRTUAL Process Type <<Block Type Q2932_BT>> Coord_PT

4(15)

:fpar uniPartner PId , config InitTrunkConfig;

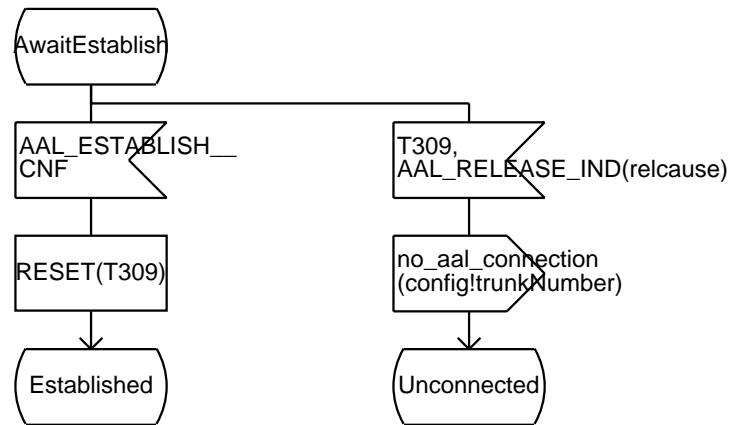


Annex B: Coord_PT

VIRTUAL Process Type <<Block Type Q2932_BT>> Coord_PT

5(15)

;fpar uniPartner PId , config InitTrunkConfig;

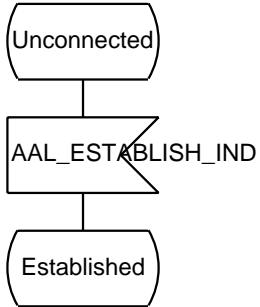


Annex B: Coord_PT

VIRTUAL Process Type <<Block Type Q2932_BT>> Coord_PT

6(15)

```
;fpar uniPartner PId , config InitTrunkConfig;
```

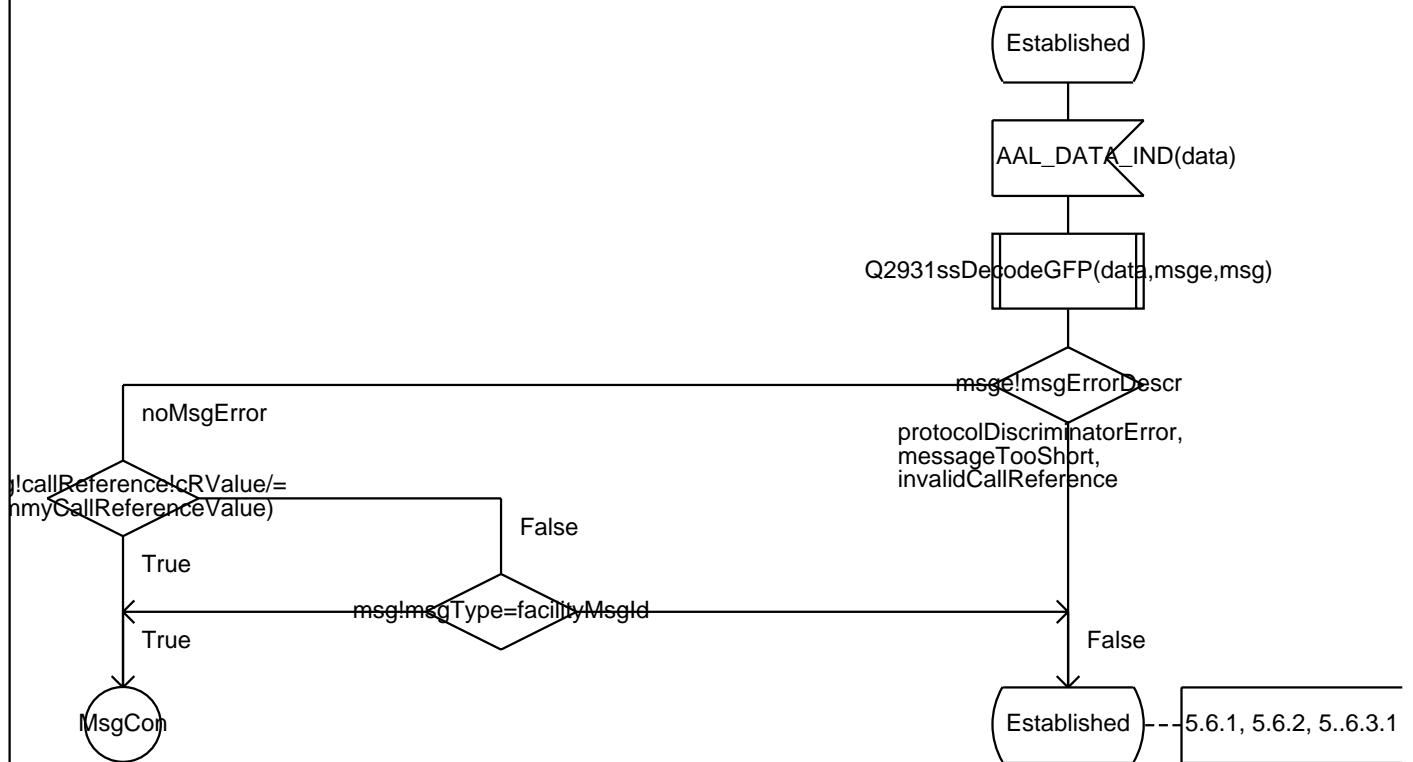


Annex B: Coord_PT

VIRTUAL Process Type <<Block Type Q2932_BT>> Coord_PT

7(15)

;fpar uniPartner PId , config InitTrunkConfig;

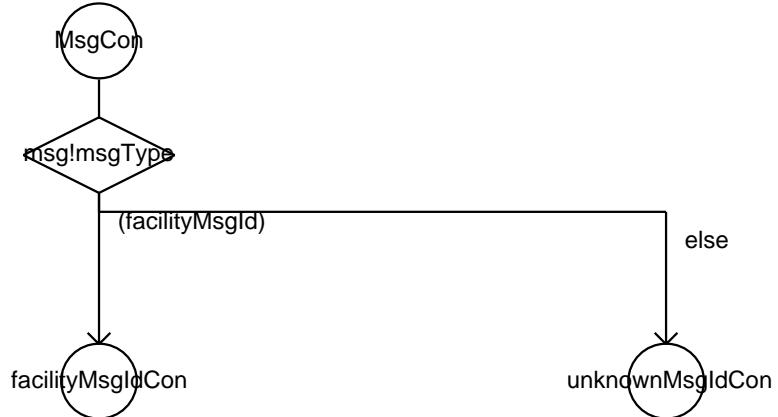


Annex B: Coord_PT

VIRTUAL Process Type <<Block Type Q2932_BT>> Coord_PT

8(15)

:fpar uniPartner PId , config InitTrunkConfig;

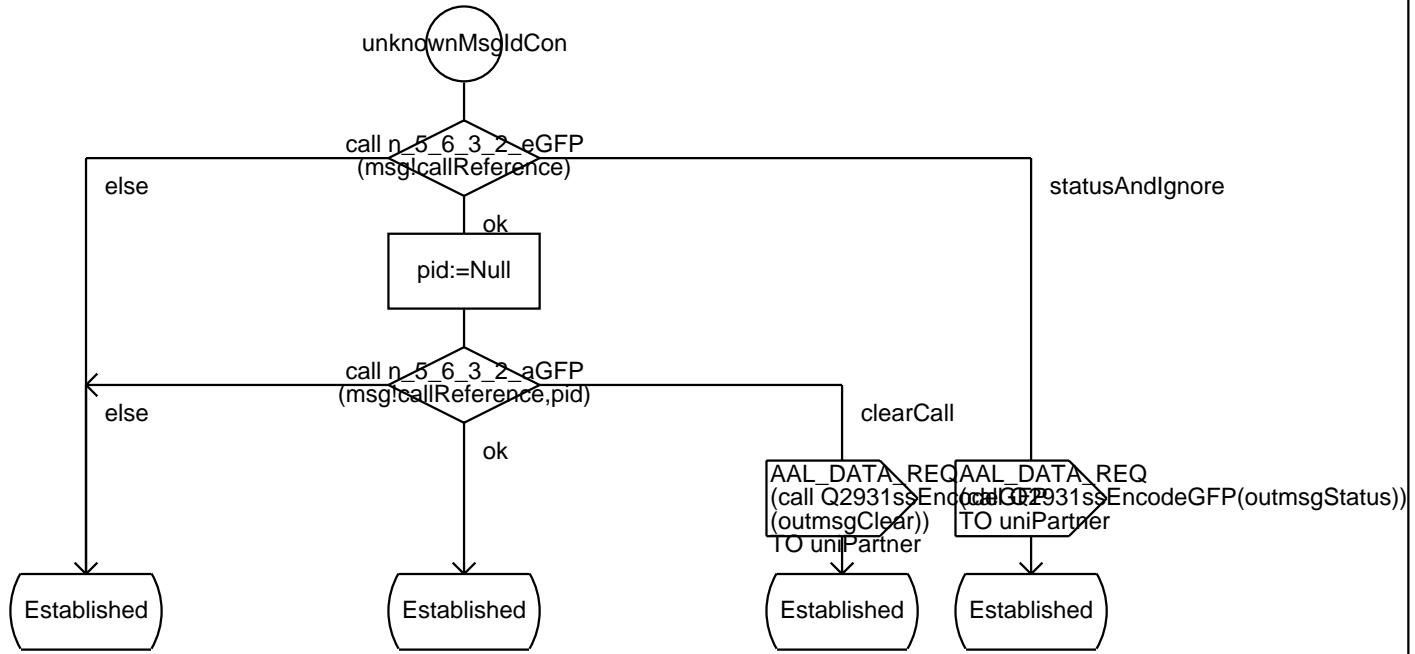


Annex B: Coord_PT

VIRTUAL Process Type <<Block Type Q2932_BT>> Coord_PT

9(15)

;fpar uniPartner PId , config InitTrunkConfig;

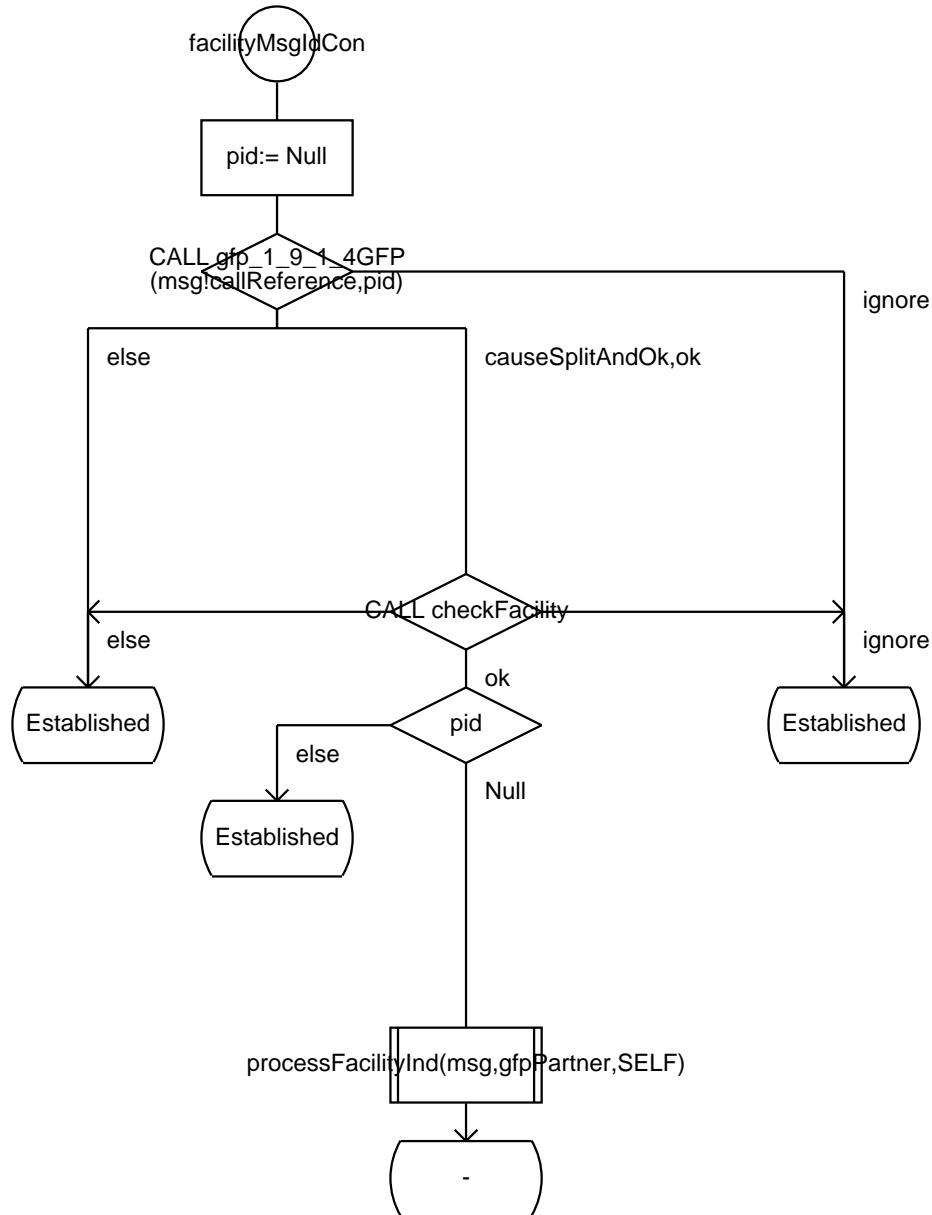


Annex B: Coord_PT

VIRTUAL Process Type <<Block Type Q2932_BT>> Coord_PT

10(15)

;fpar uniPartner PId , config InitTrunkConfig;

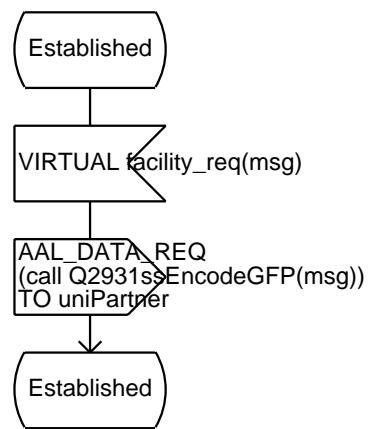


Annex B: Coord_PT

VIRTUAL Process Type <<Block Type Q2932_BT>> Coord_PT

11(15)

;fpar uniPartner PId , config InitTrunkConfig;

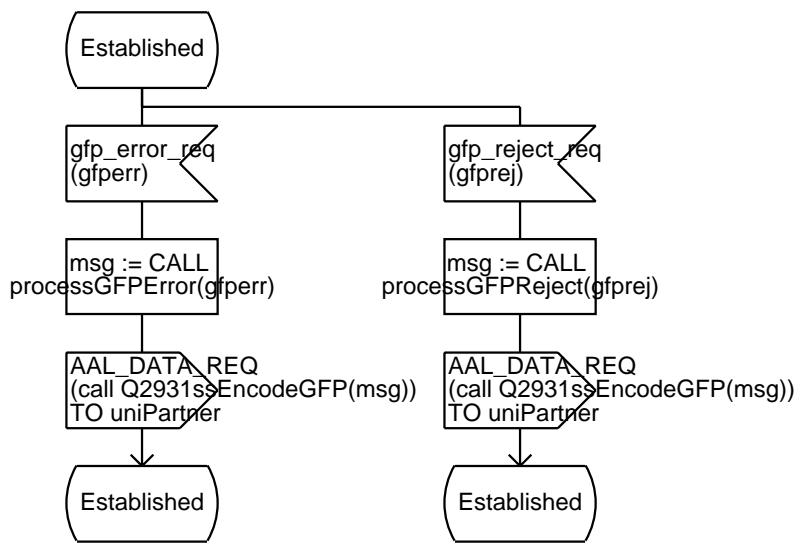


Annex B: Coord_PT

VIRTUAL Process Type <<Block Type Q2932_BT>> Coord_PT

12(15)

;fpar uniPartner PId , config InitTrunkConfig;

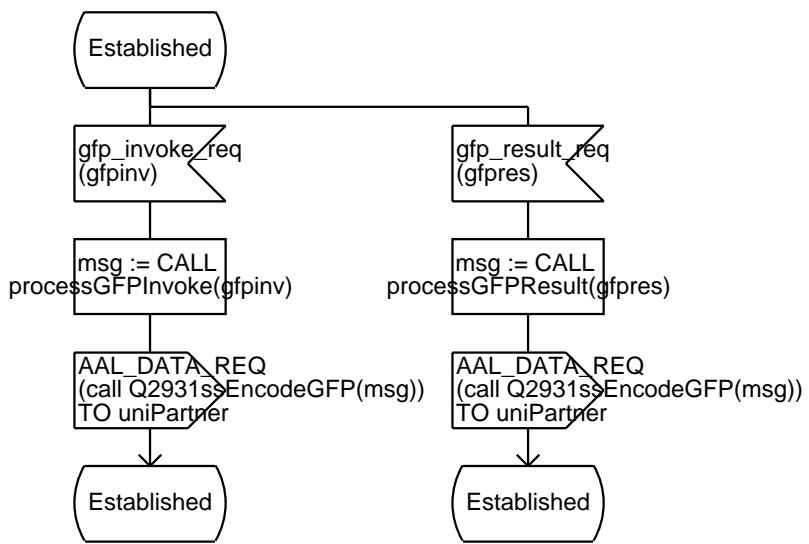


Annex B: Coord_PT

VIRTUAL Process Type <<Block Type Q2932_BT>> Coord_PT

13(15)

;fpar uniPartner PId , config InitTrunkConfig;

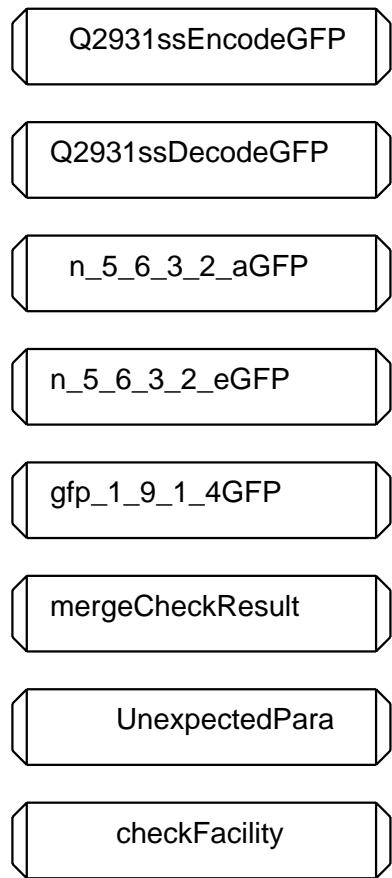


Annex B: Coord_PT

VIRTUAL Process Type <<Block Type Q2932_BT>> Coord_PT

14(15)

```
:fpar uniPartner PId , config InitTrunkConfig;
```



Annex B: Coord_PT

VIRTUAL Process Type <<Block Type Q2932_BT>> Coord_PT

15(15)

```
;fpar uniPartner PId , config InitTrunkConfig;
```

Annex B: Q2931ssEncodeGFP

Procedure Q2931ssEncodeGFP

1(1)

```
; FPAR msg Q2931ssMessage;
RETURNS AALData;
```

```
/*#include '../sdl/q2931ssencode.sdl' */
```

Annex B: Q2931ssDecodeGFP

Procedure Q2931ssDecodeGFP

1(1)

; FPAR in data AALData,
in/out msge MsgError, in/out msg Q2931ssMessage;

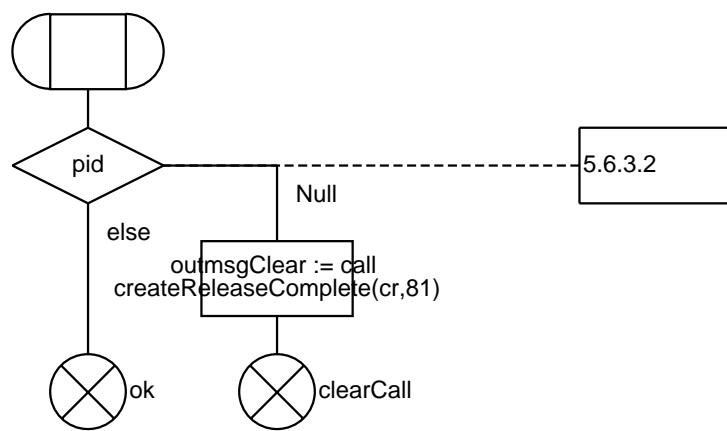
```
/*#include '../sdl/q2931ssdecode.sdl' */
```

Annex B: n_5_6_3_2_aGFP

Procedure <>Process Type Coord_PT>> n_5_6_3_2_aGFP

1(1)

```
;fpar cr CallReference,pid PId;
returns CheckResultType;
```

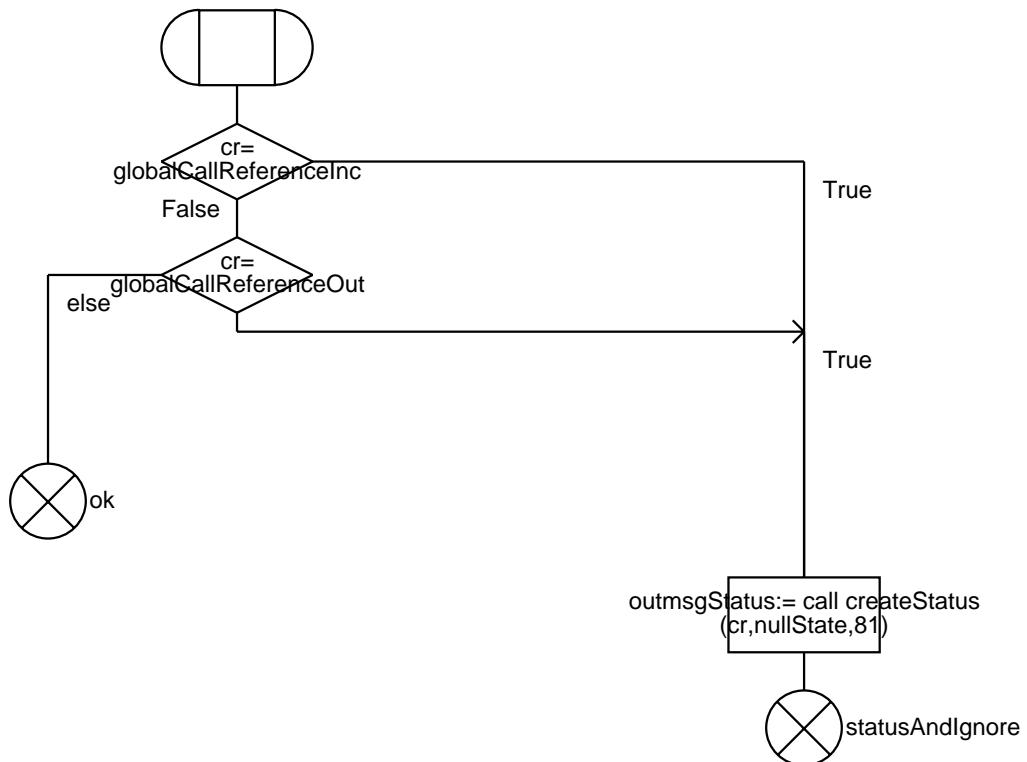


Annex B: n_5_6_3_2_eGFP

Procedure n_5_6_3_2_eGFP

1(1)

```
;fpar cr CallReference;
returns CheckResultType;
```

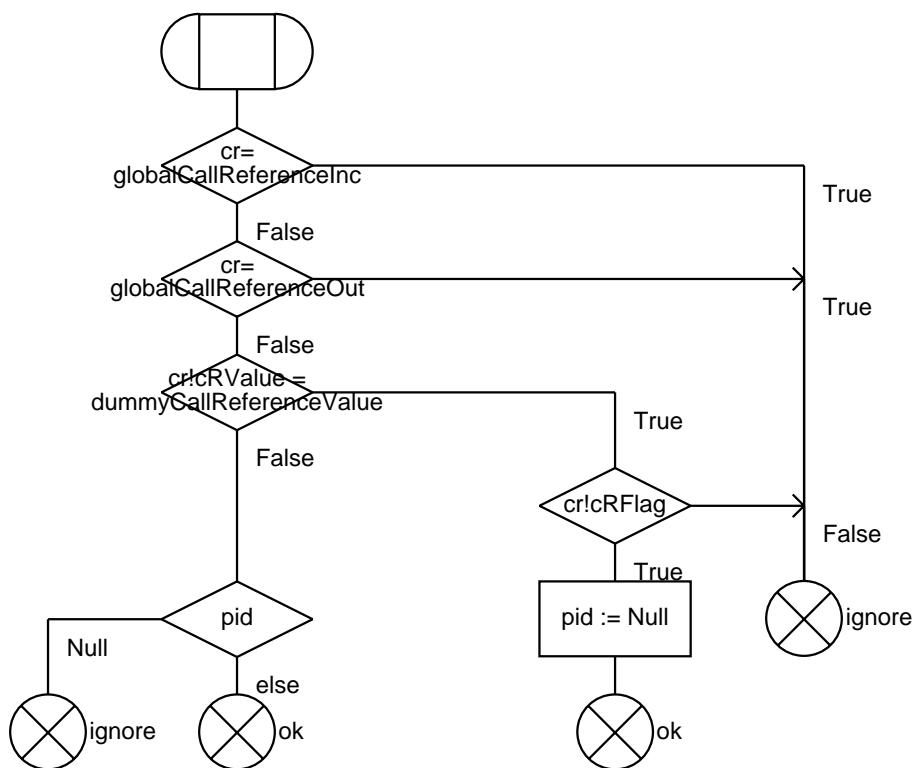


Annex B: gfp_1_9_1_4GFP

Procedure <<Block Type Q2932_BT/Process Type Coord_PT>> gfp_1_9_1_4GFP

1(1)

:FPAR cr CallReference, in/out pid Pld;
 RETURNS CheckResultType;

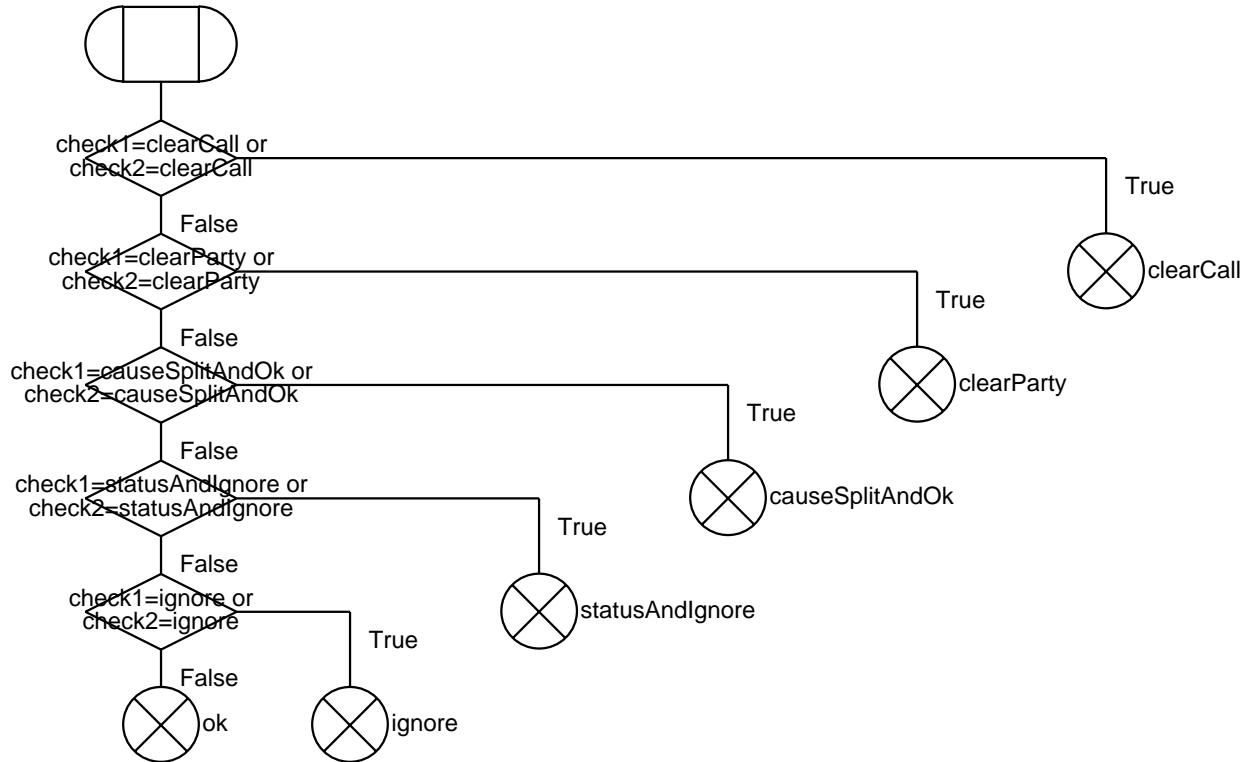


Annex B: mergeCheckResult

Procedure <>Process Type Coord_PT>> mergeCheckResult

1(1)

```
;fpar check1 CheckResultType, check2 CheckResultType;
returns CheckResultType;
```

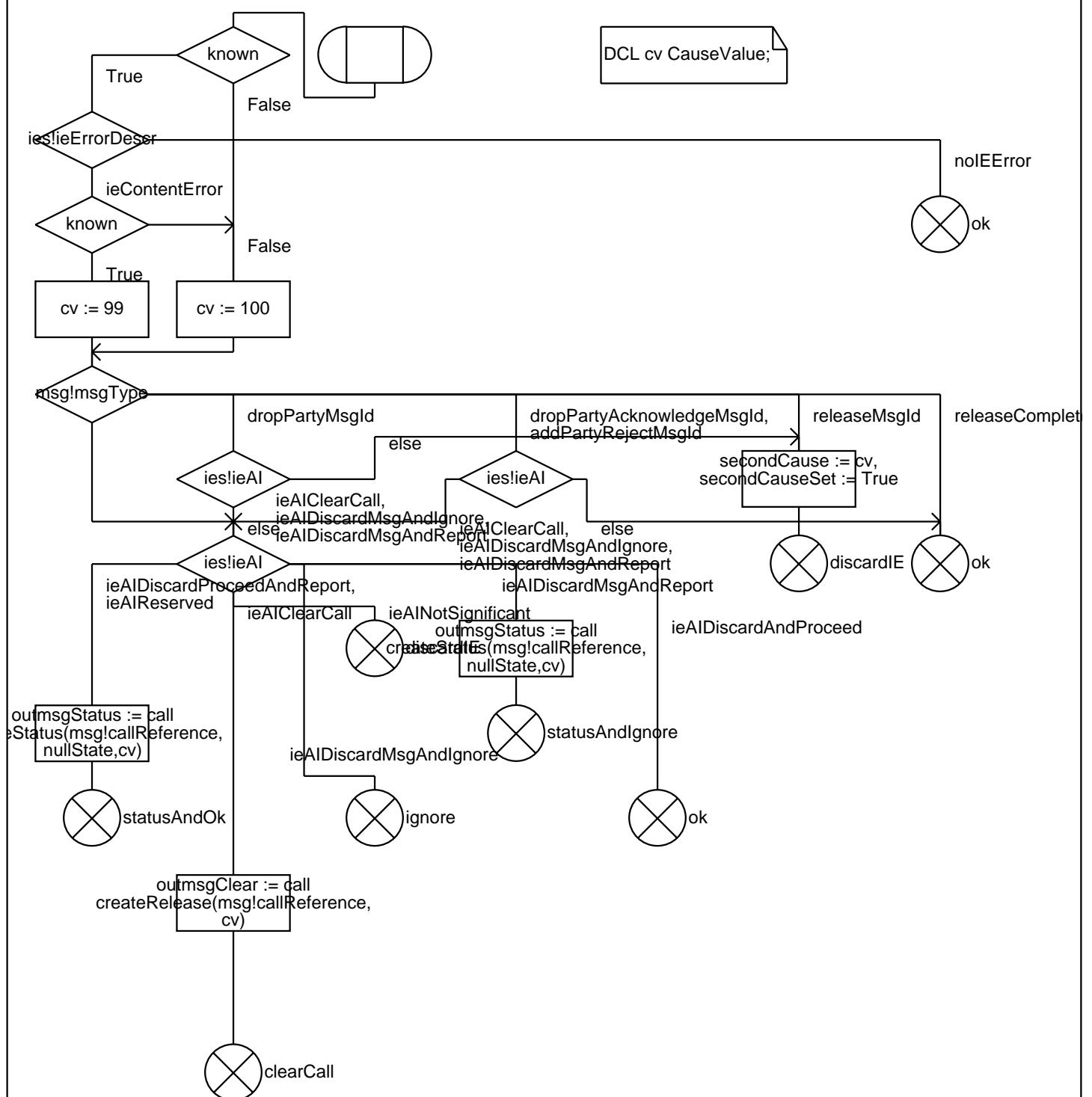


Annex B: UnexpectedPara

Procedure <<Process Type Coord_PT>> UnexpectedPara

1(1)

;FPAR ies IEEErrorStruct, known Boolean; RETURNS CheckResultType;



Annex B: checkFacility

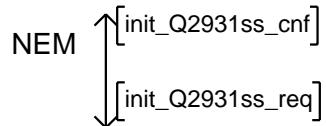
Procedure <<Process Type Coord_PT>> checkFacility

1(1)

;RETURNS res CheckResultType;

```
/*#include '../sdl/Checkfacility.sdl' */
```

Annex B: CoordControl_PT



Process Type <<Block Type Q2932_BT>> CoordControl_PT

1(2)

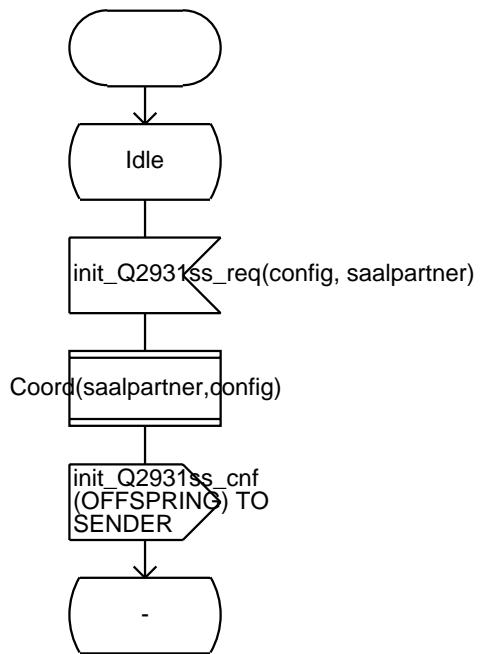
```
DCL saalpartner PId := Null;
```

```
/* local vars */
DCL config InitTrunkConfig;
```

Annex B: CoordControl_PT

Process Type <<Block Type Q2932_BT>> CoordControl_PT

2(2)



Annex B: Q2931ssNet_BT

Block Type Q2931ssNet_BT

1(1)

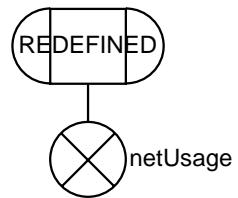
INHERITS Q2931ss_BT;

REDEFINED whatUsage

Annex B: whatUsage

Redefined Procedure <<Block Type Q2931ssNet_BT>> whatUsage

1(1)



Annex B: Q2931ssUser_BT

Block Type Q2931ssUser_BT

1(1)

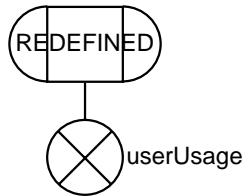
INHERITS Q2931ss_BT;

REDEFINED whatUsage

Annex B: whatUsage

Redefined Procedure <<Block Type Q2931ssUser_BT>> whatUsage

1(1)

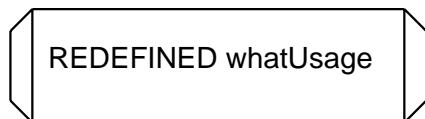


Annex B: Q2931ssNetSim_BT

Block Type Q2931ssNetSim_BT

1(1)

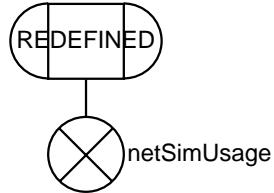
INHERITS Q2931ss_BT;



Annex B: whatUsage

Redefined Procedure <<Block Type Q2931ssNetSim_BT>> whatUsage

1(1)



Annex B: Q2931ssUserSim_BT

Block Type Q2931ssUserSim_BT

1(1)

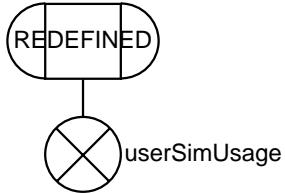
INHERITS Q2931ss_BT;

REDEFINED whatUsage

Annex B: whatUsage

Redefined Procedure <<Block Type Q2931ssUserSim_BT>> whatUsage

1(1)

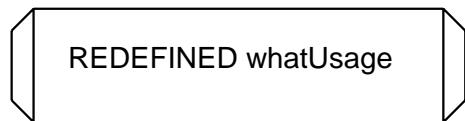


Annex B: Q2931ssNetVal_BT

Block Type Q2931ssNetVal_BT

1(1)

INHERITS Q2931ss_BT;



Annex B: Coord_PT

Redefined Process Type <<Block Type Q2931ssNetVal_BT>> Coord_PT

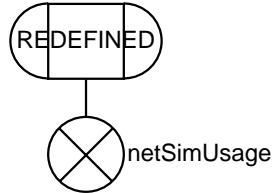
1(1)

```
/*#include '../sdl/coordval.sdl' */
```

Annex B: whatUsage

Redefined Procedure <<Block Type Q2931ssNetVal_BT>> whatUsage

1(1)



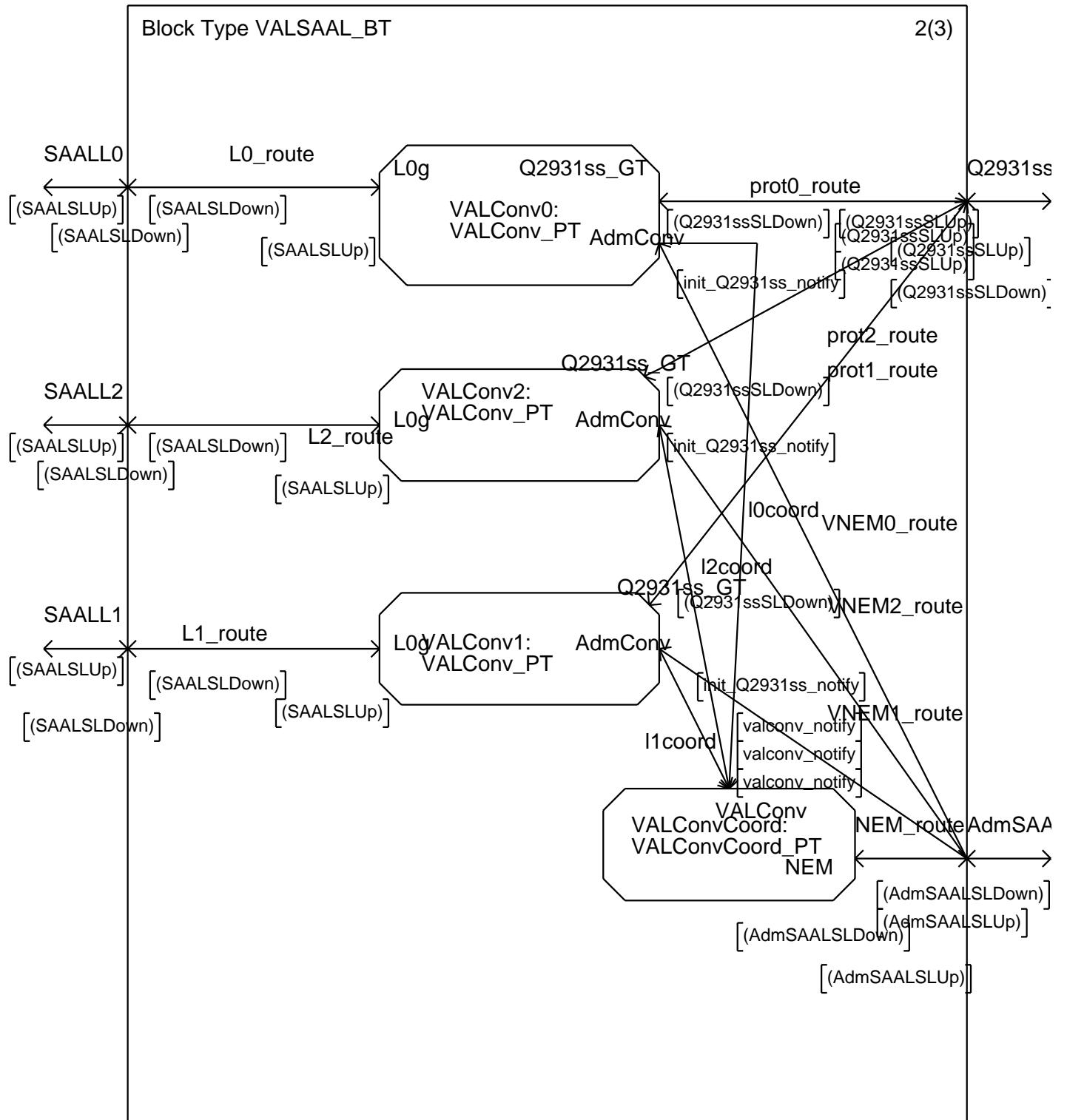
Annex B: VALSAAL_BT

Block Type VALSAAL_BT

1(3)

```
SIGNAL valconv_notify(PId),  
valconv_init_req(PId);
```

Annex B: VALSAAL_BT



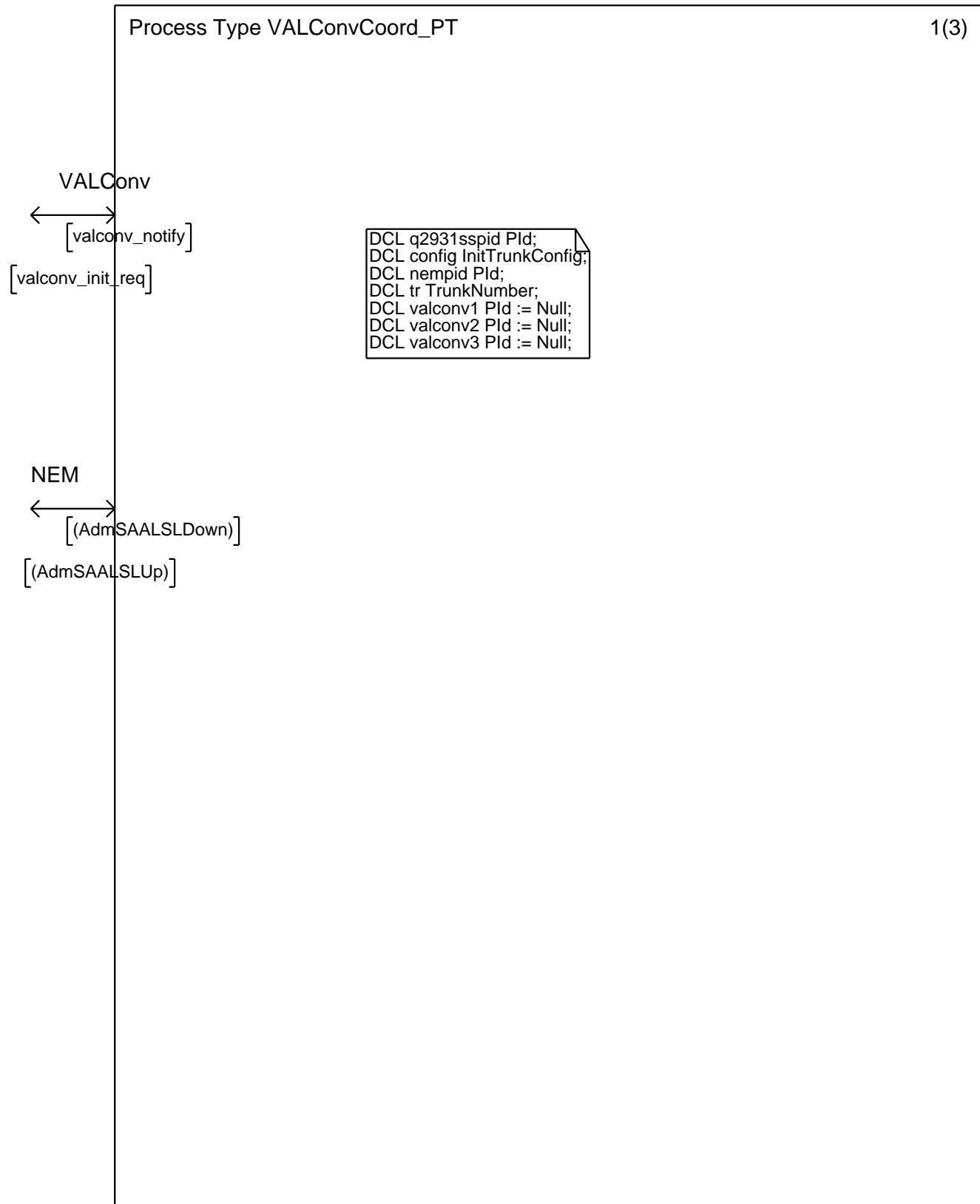
Annex B: VALSAAL_BT

Block Type VALSAAL_BT

3(3)

VALConvCoord_PTVALConv_PT

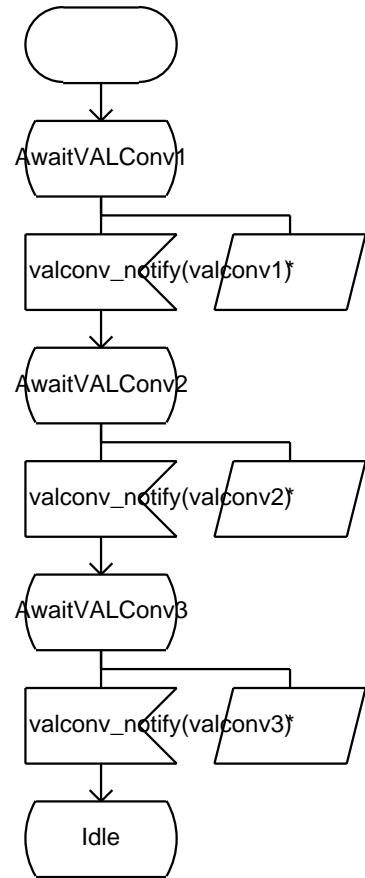
Annex B: VALConvCoord_PT



Annex B: VALConvCoord_PT

Process Type VALConvCoord_PT

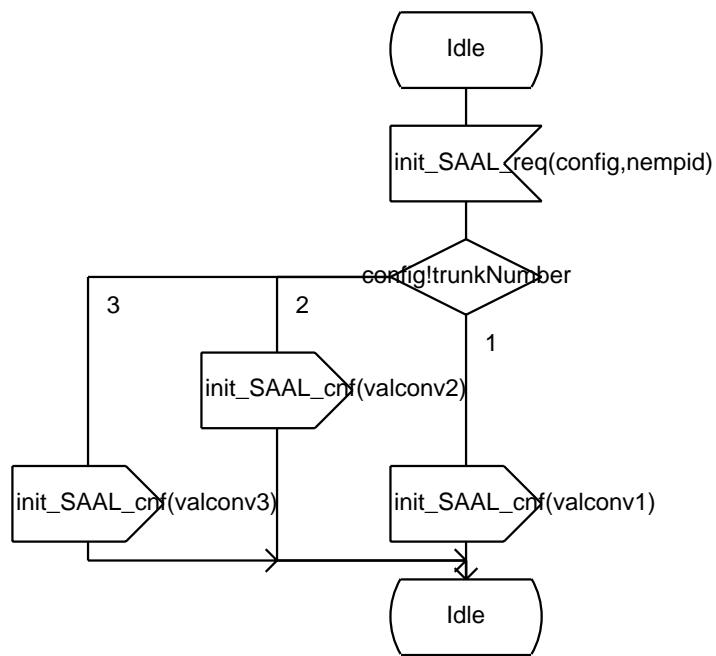
2(3)



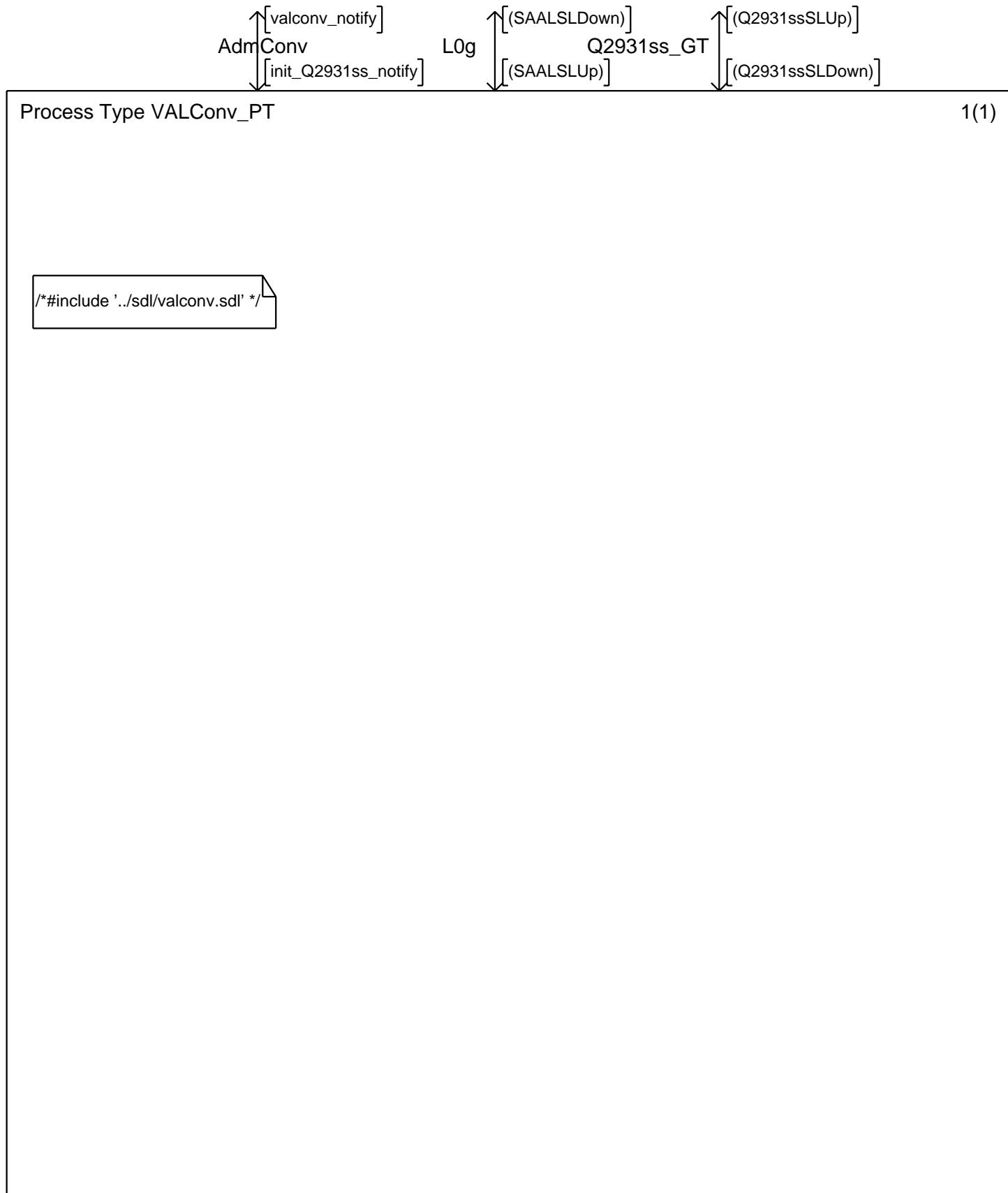
Annex B: VALConvCoord_PT

Process Type VALConvCoord_PT

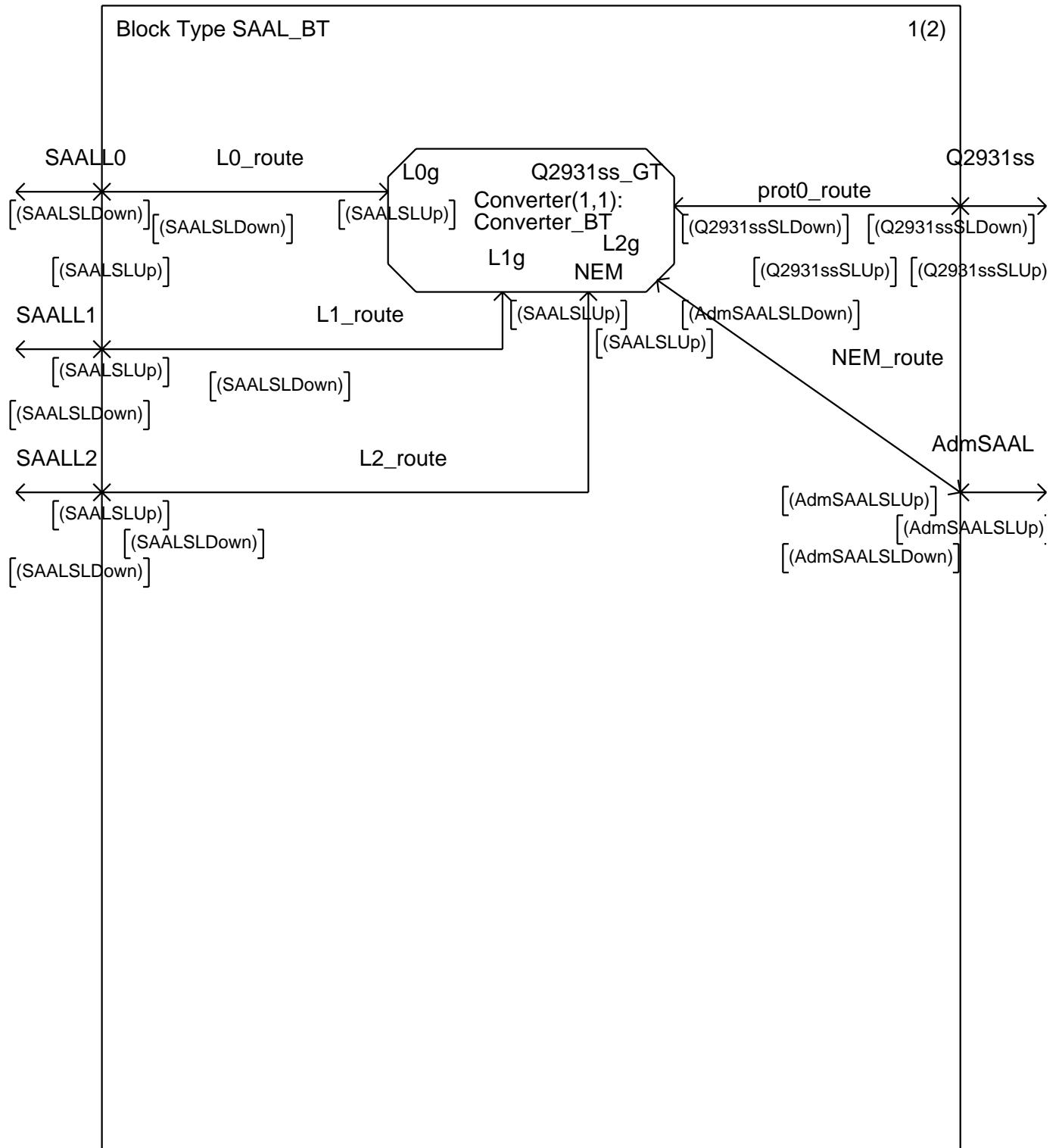
3(3)



Annex B: VALConv_PT



Annex B: SAAL_BT



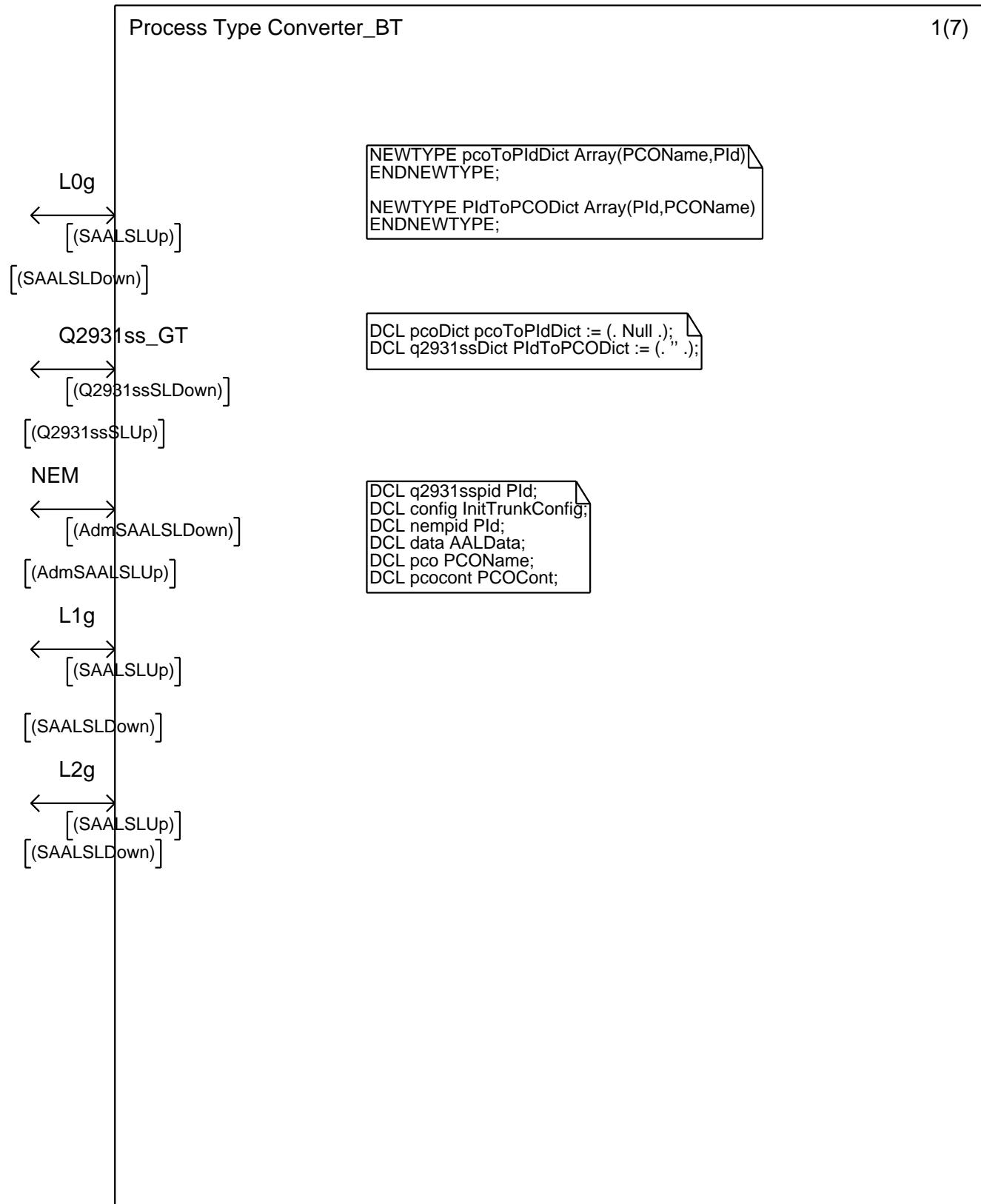
Annex B: SAAL_BT

Block Type SAAL_BT

2(2)



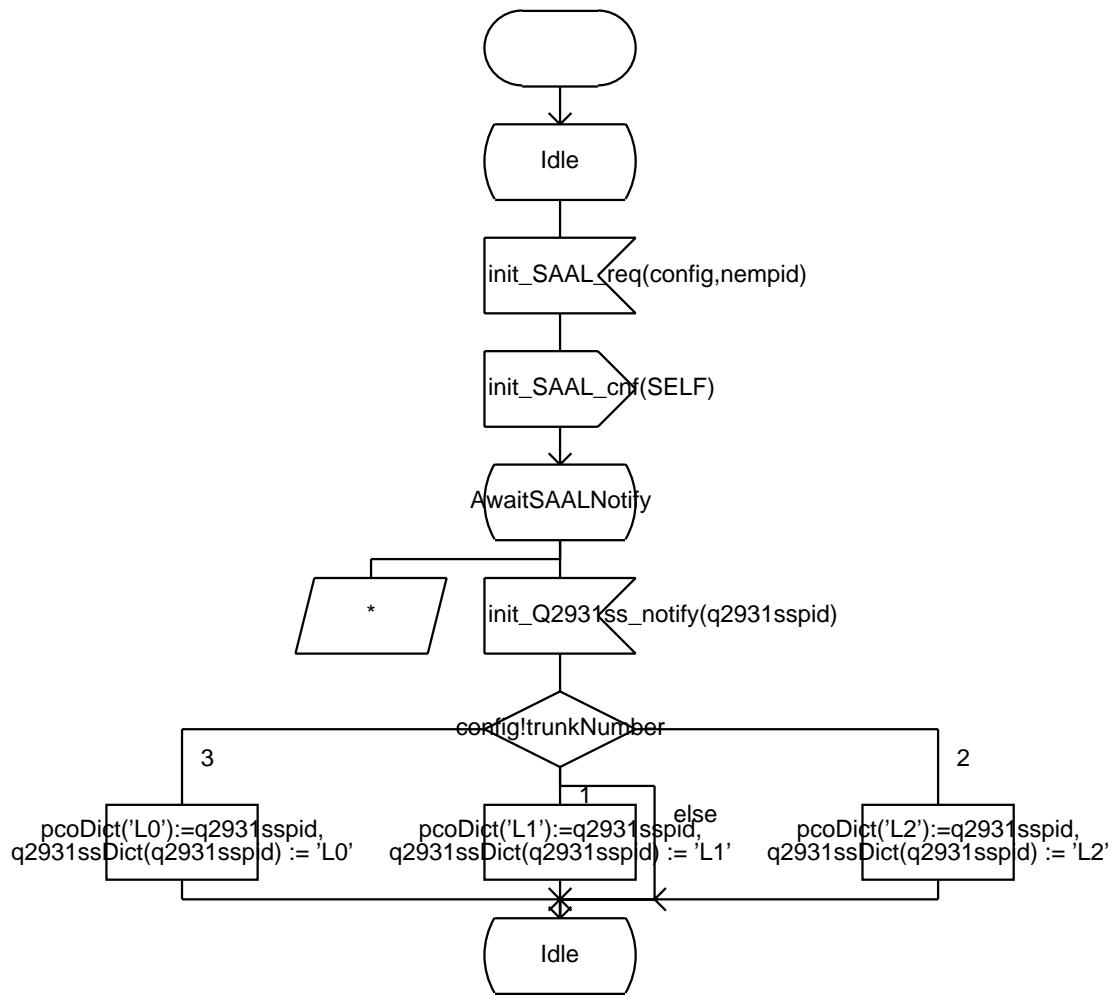
Annex B: Converter_BT



Annex B: Converter_BT

Process Type Converter_BT

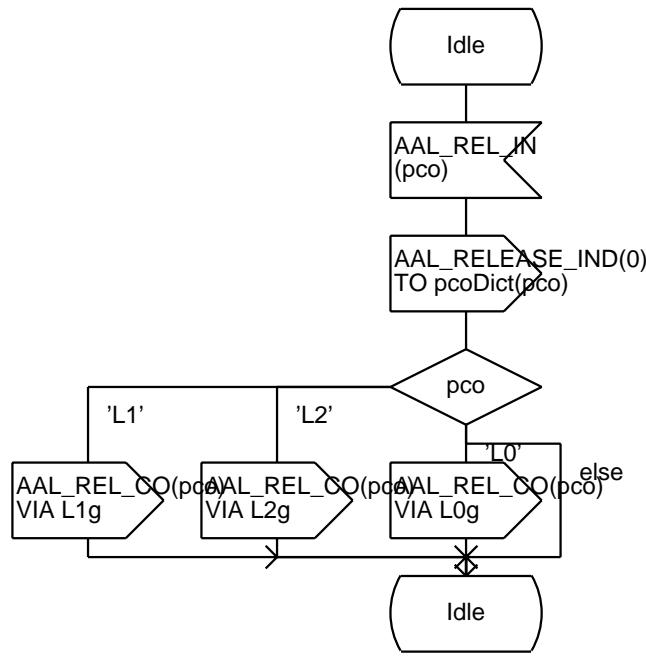
2(7)



Annex B: Converter_BT

Process Type Converter_BT

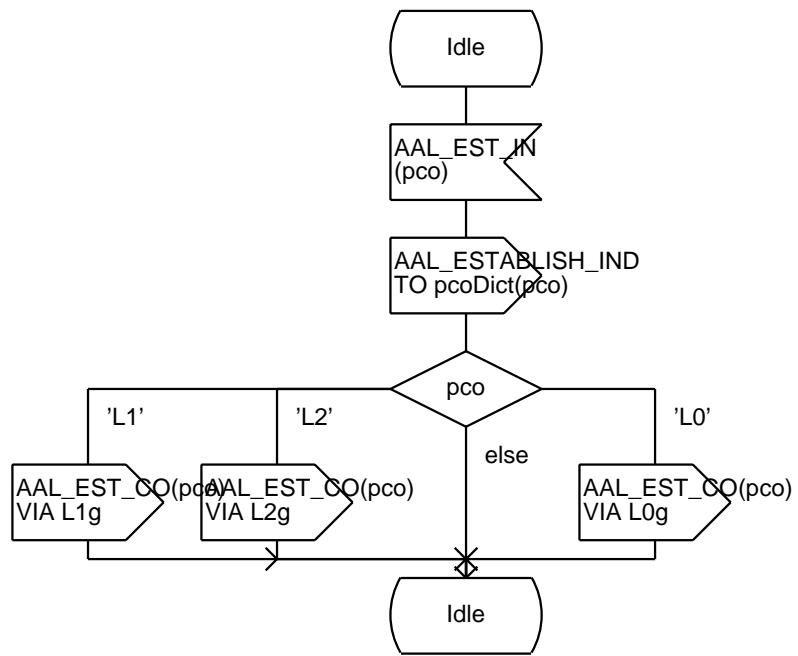
3(7)



Annex B: Converter_BT

Process Type Converter_BT

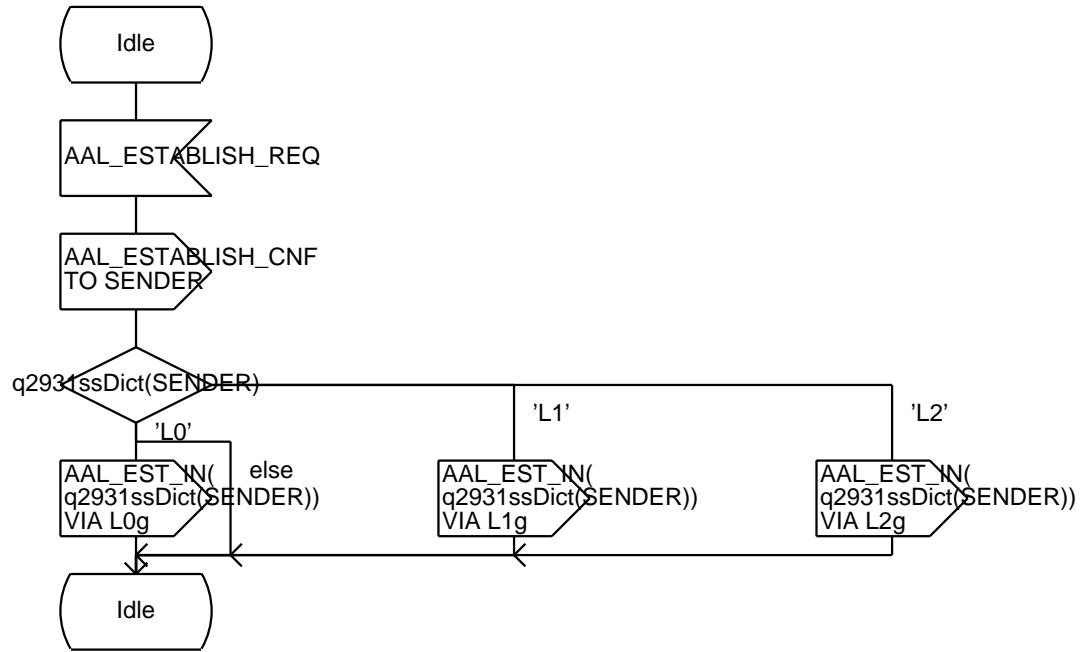
4(7)



Annex B: Converter_BT

Process Type Converter_BT

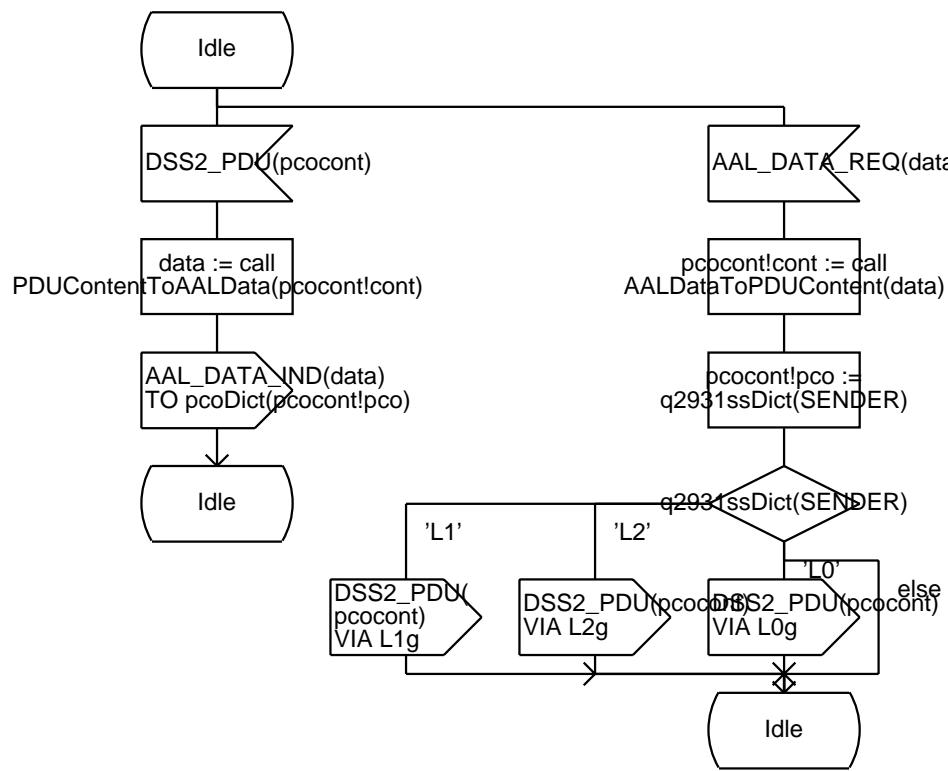
5(7)



Annex B: Converter_BT

Process Type Converter_BT

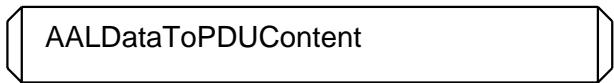
6(7)



Annex B: Converter_BT

Process Type Converter_BT

7(7)

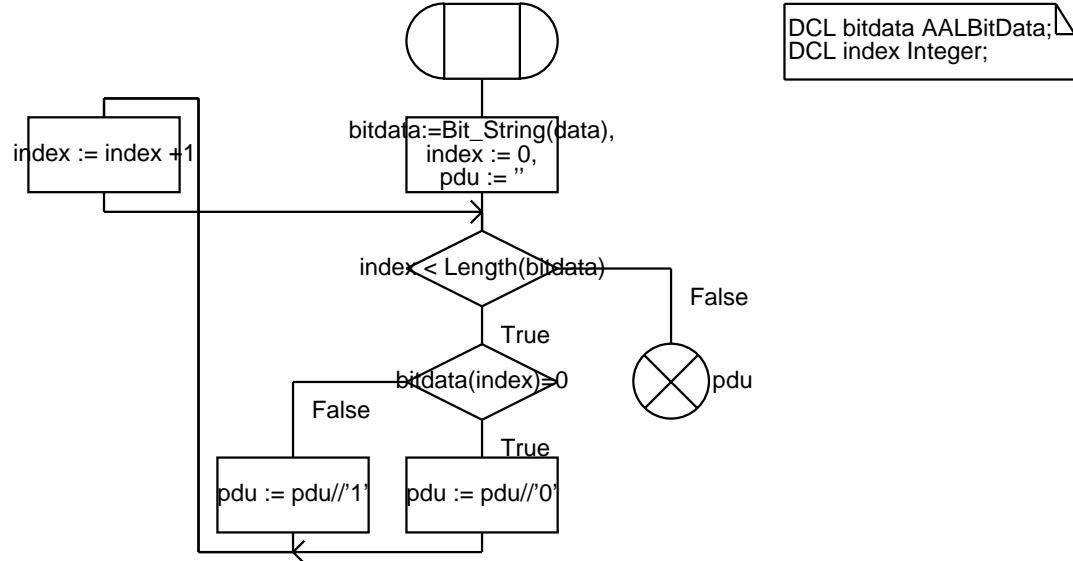


Annex B: AALDataToPDUContent

Procedure AALDataToPDUContent

1(1)

```
:FPAR data AALData;
RETURNS pdu PDUContents;
```

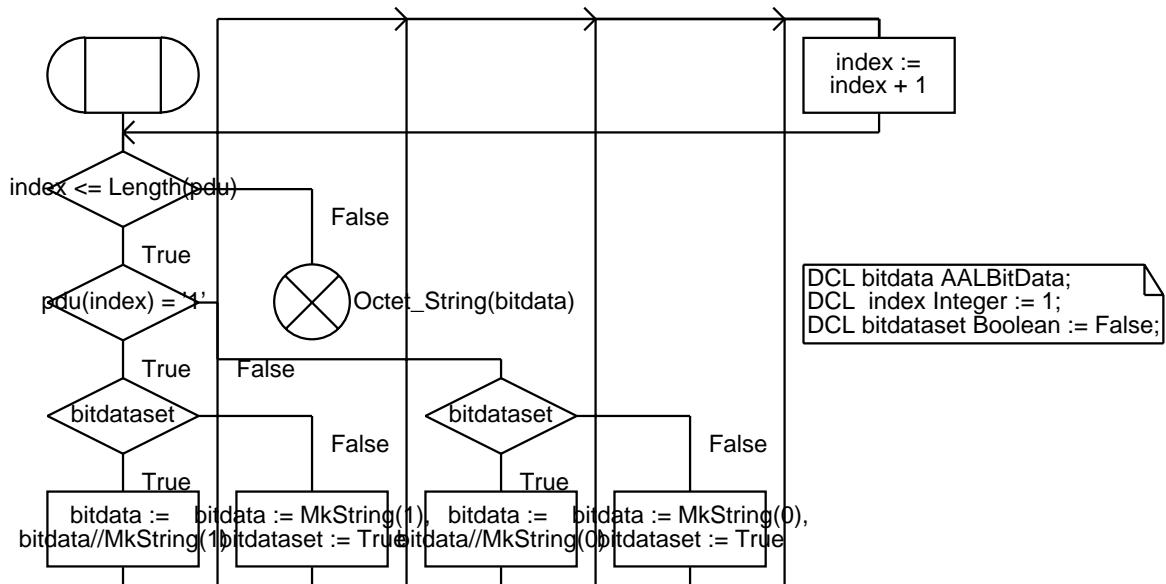


Annex B: PDUContentToAALData

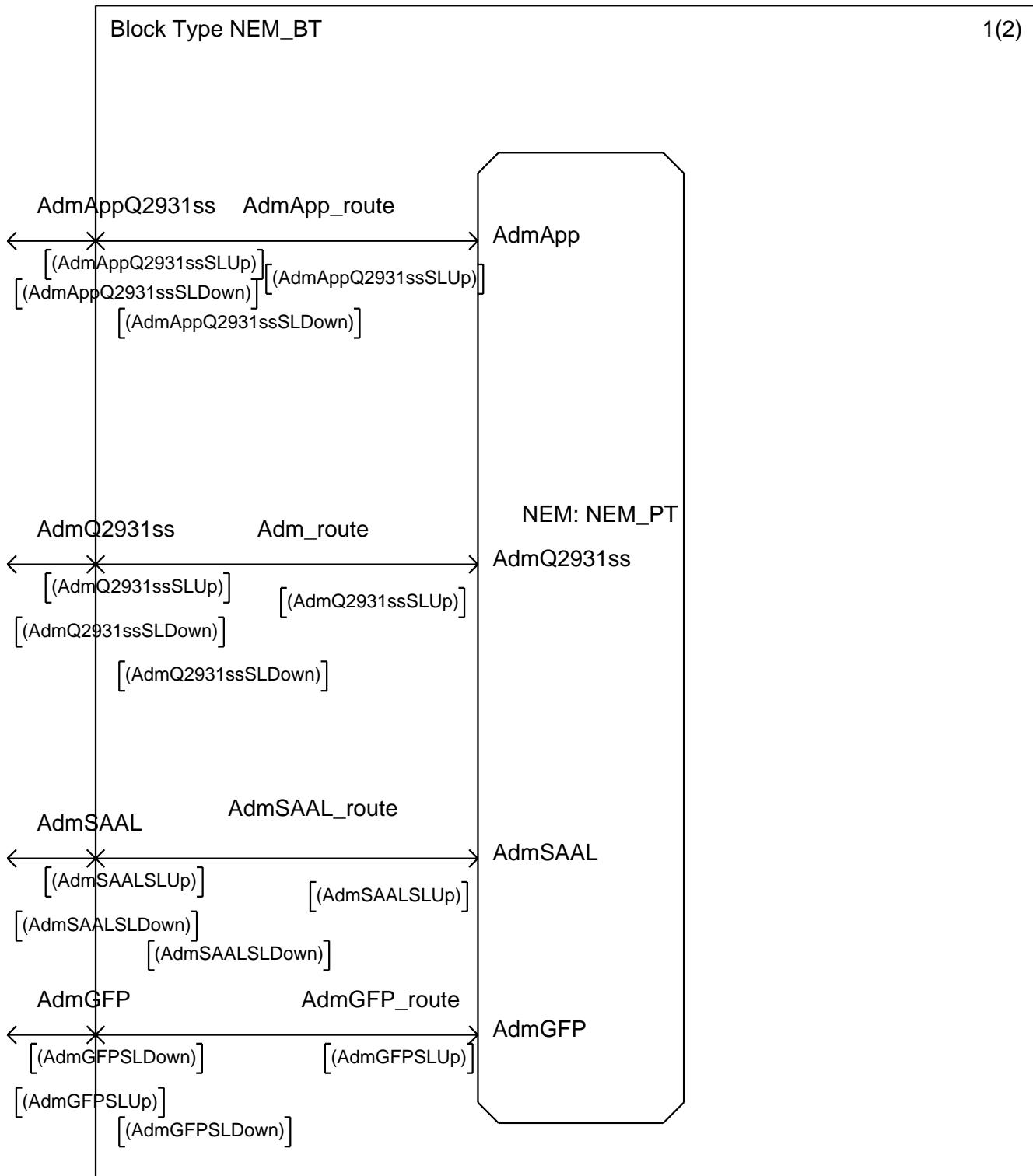
Procedure PDUContentToAALData

1(1)

```
:FPAR pdu PDUContents;
RETURNS AALData;
```



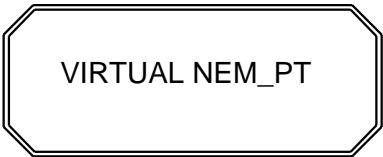
Annex B: NEM_BT



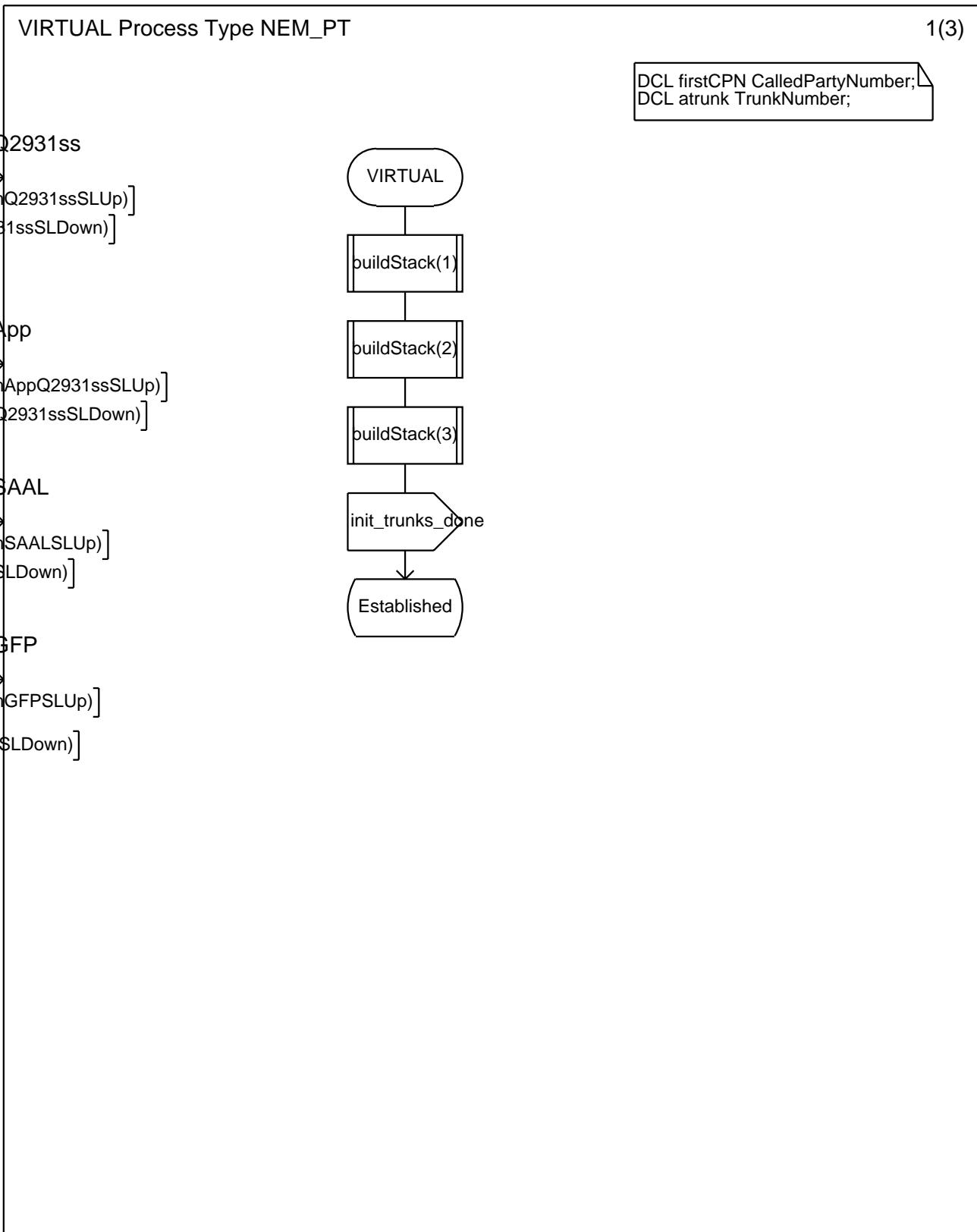
Annex B: NEM_BT

Block Type NEM_BT

2(2)

VIRTUAL NEM_PT

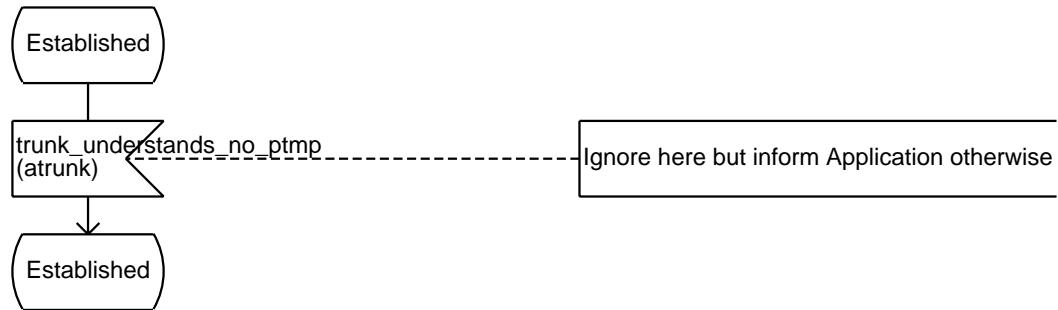
Annex B: NEM_PT



Annex B: NEM_PT

VIRTUAL Process Type NEM_PT

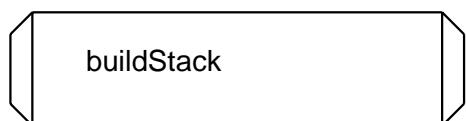
2(3)



Annex B: NEM_PT

VIRTUAL Process Type NEM_PT

3(3)



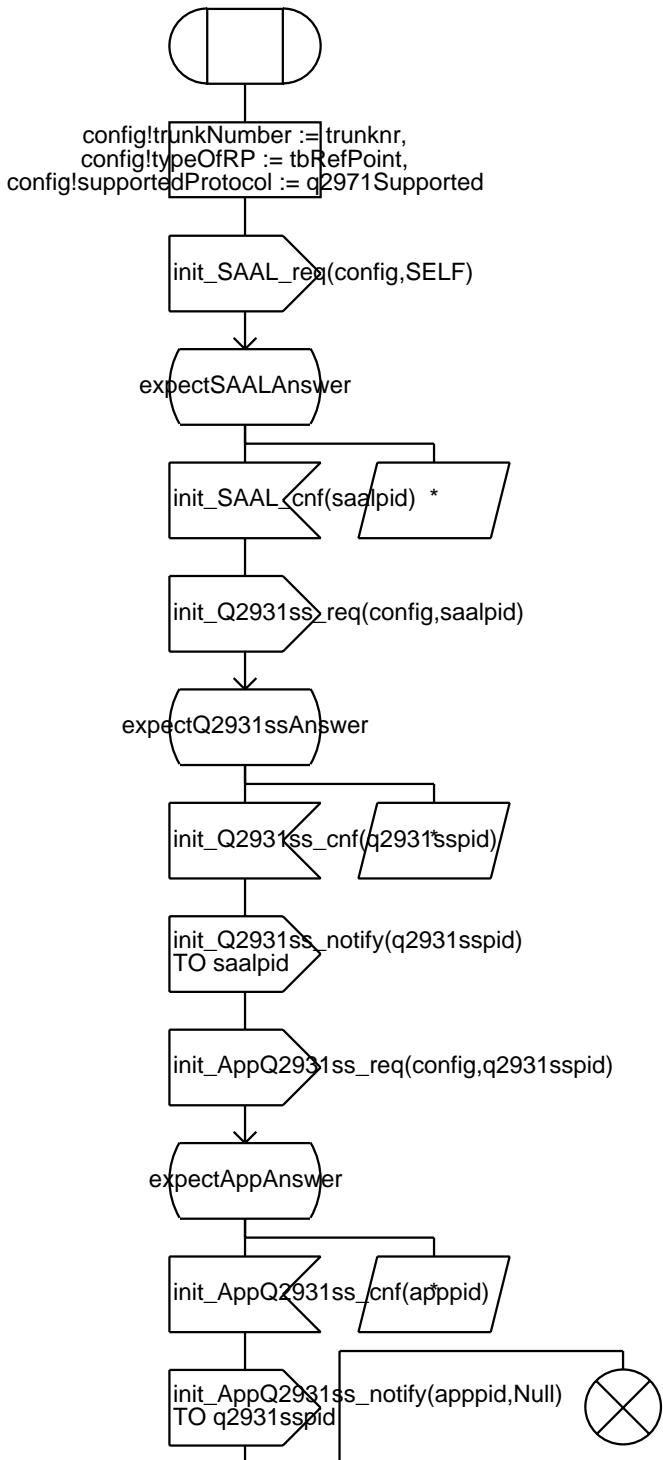
Annex B: buildStack

Procedure buildStack

1(1)

;FPAR trunknr TrunkNumber;

DCL config InitTrunkConfig;
 DCL saalpid Plid;
 DCL q2931sspid Plid;
 DCL apppid Plid;

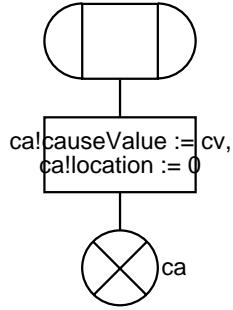


Annex B: createCause

Procedure createCause

1(1)

```
:FPAR cv CauseValue;  
RETURNS ca Cause;
```

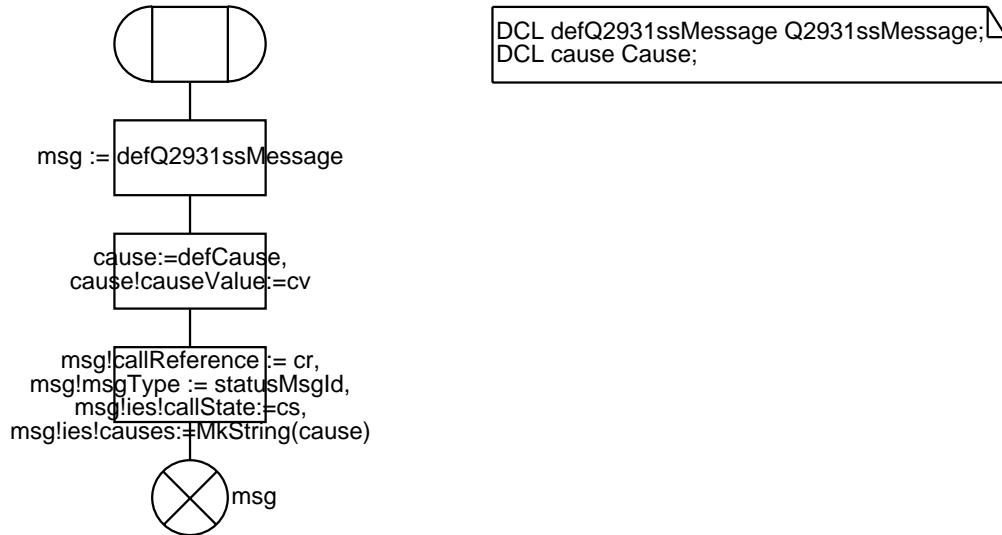


Annex B: createStatus

Procedure createStatus

1(1)

;fpar cr CallReference,cs CallState,cv CauseValue;
 returns msg Q2931ssMessage;

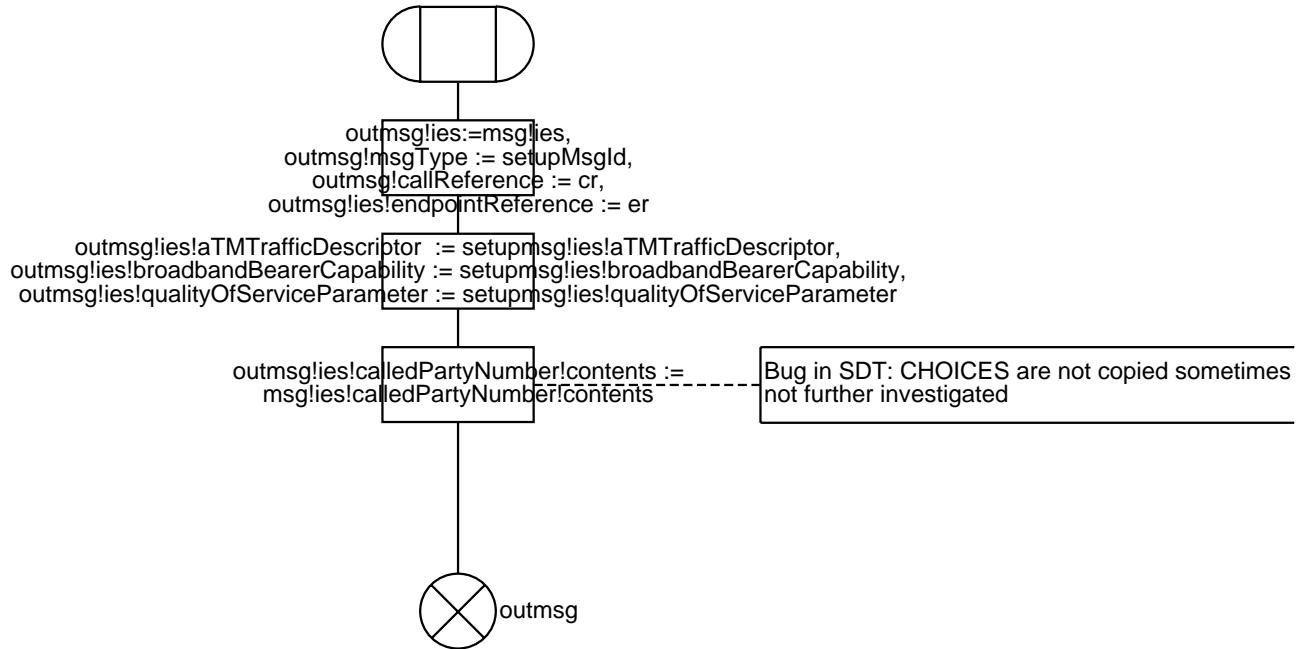


Annex B: createSetupFromAddParty

Procedure createSetupFromAddParty

1(1)

;FPAR msg Q2931ssMessage, setupmsg Q2931ssMessage, cr CallReference, er EndpointReference;
returns outmsg Q2931ssMessage;

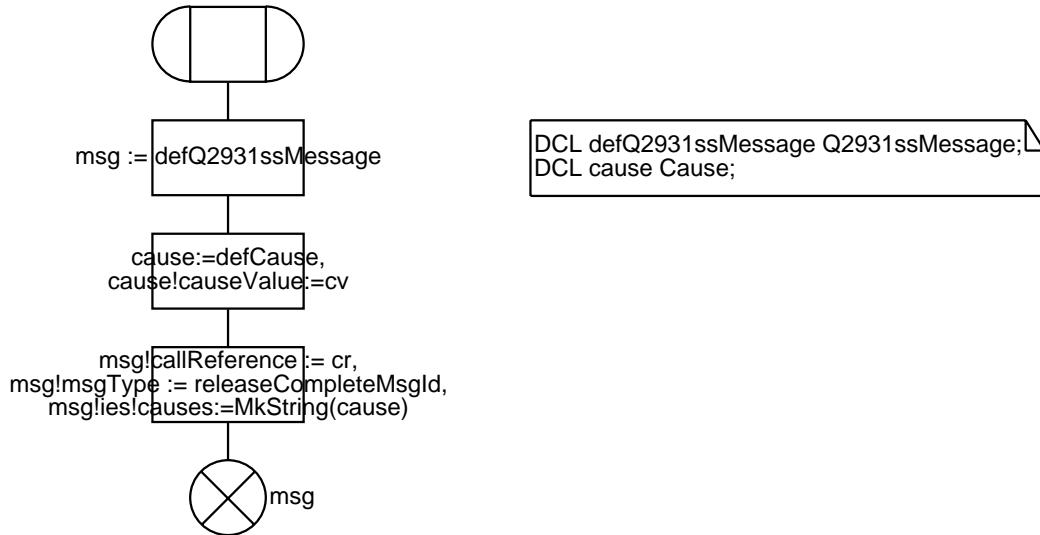


Annex B: createReleaseComplete

Procedure createReleaseComplete

1(1)

;fpar cr CallReference.cv CauseValue;
 returns msg Q2931ssMessage;

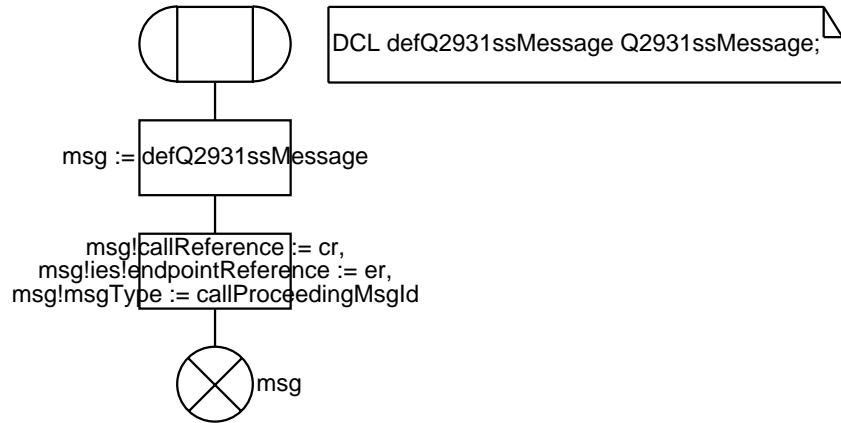


Annex B: createCallProceeding

Procedure createCallProceeding

1(1)

```
:FPAR cr CallReference, er EndpointReference;  
RETURNS msg Q2931ssMessage;
```

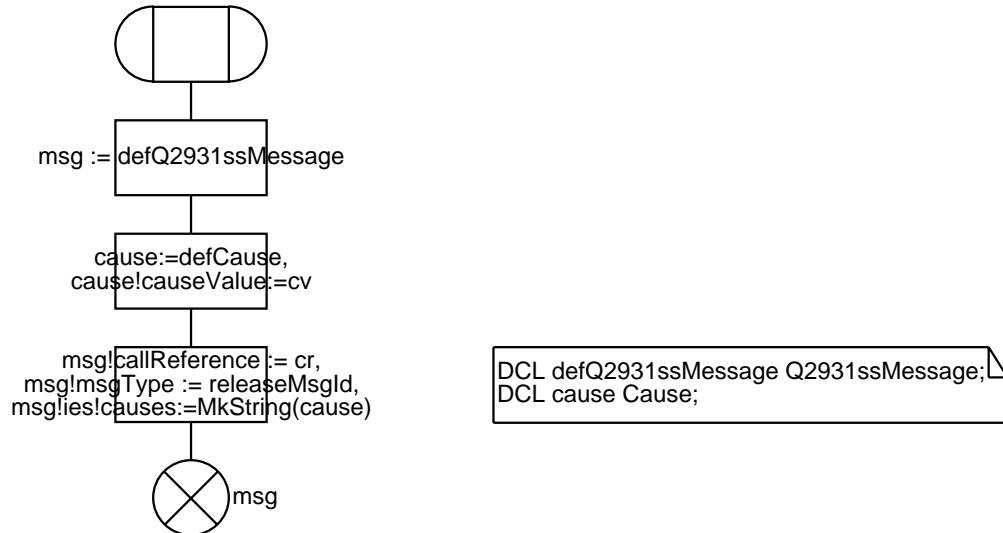


Annex B: createRelease

Procedure createRelease

1(1)

```
:fpar cr CallReference.cv CauseValue;
returns msg Q2931ssMessage;
```

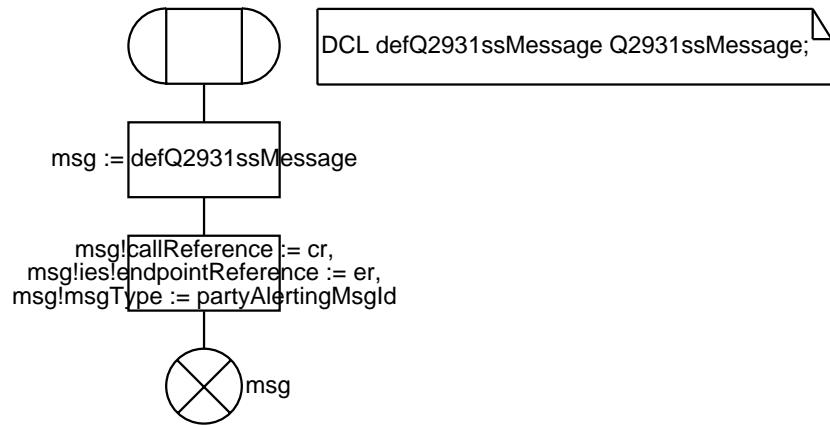


Annex B: createPartyAlerting

Procedure createPartyAlerting

1(1)

```
:FPAR cr CallReference, er EndpointReference;  
RETURNS msg Q2931ssMessage;
```

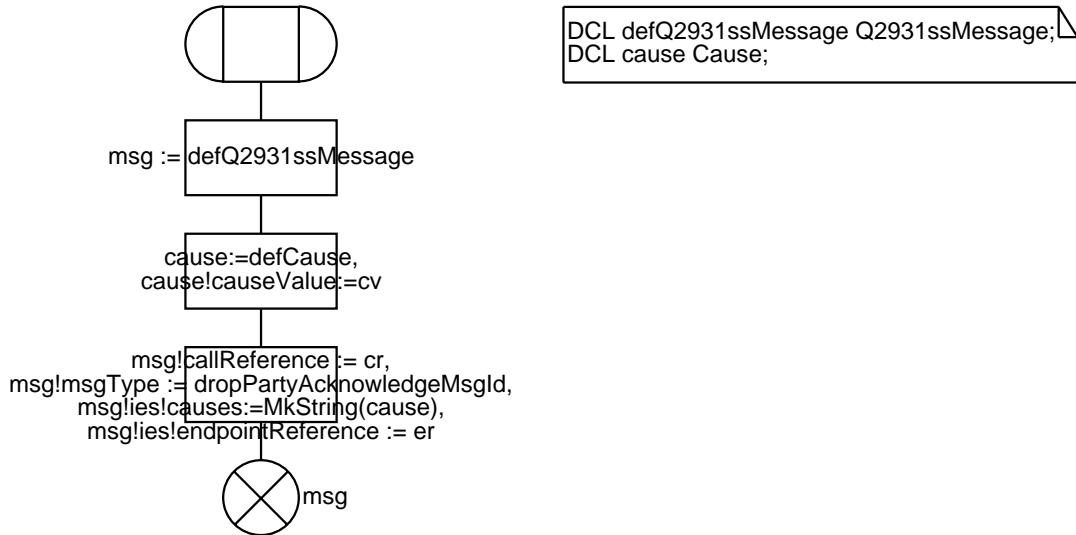


Annex B: createDropPartyAcknowledge

Procedure createDropPartyAcknowledge

1(1)

:fpar cr CallReference, cv CauseValue, er EndpointReference;
 returns msg Q2931ssMessage;

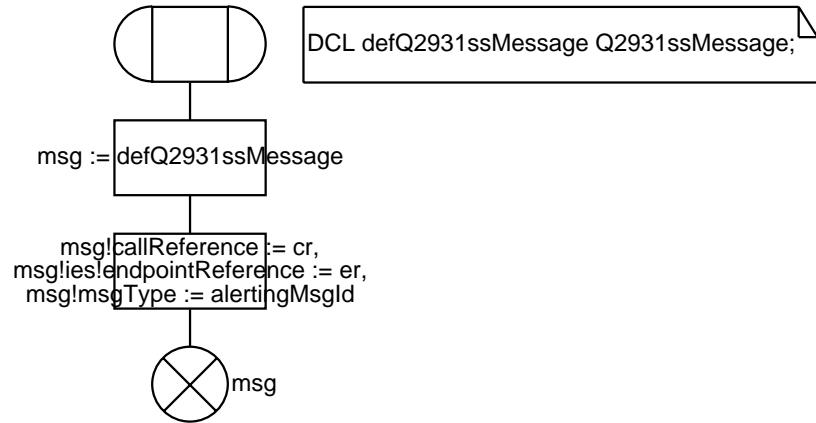


Annex B: createAlerting

Procedure createAlerting

1(1)

```
:FPAR cr CallReference, er EndpointReference;  
RETURNS msg Q2931ssMessage;
```

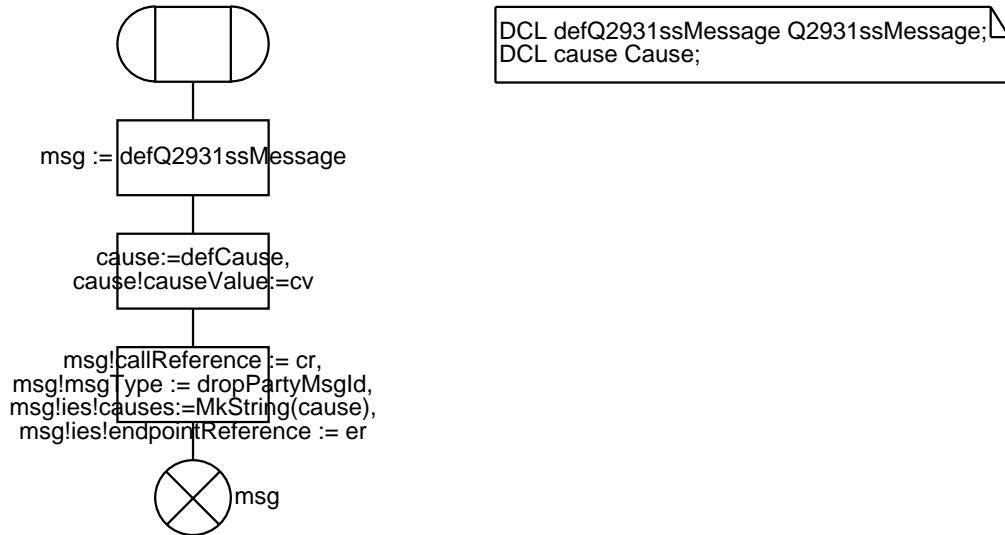


Annex B: createDropParty

Procedure createDropParty

1(1)

:fpar cr CallReference.cv CauseValue, er EndpointReference;
 returns msg Q2931ssMessage;

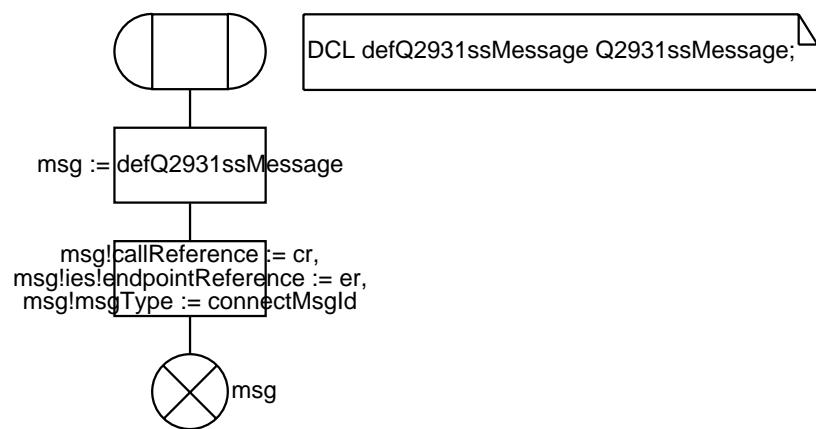


Annex B: createConnect

Procedure createConnect

1(1)

```
:FPAR cr CallReference,er EndpointReference;
RETURNS msg Q2931ssMessage;
```

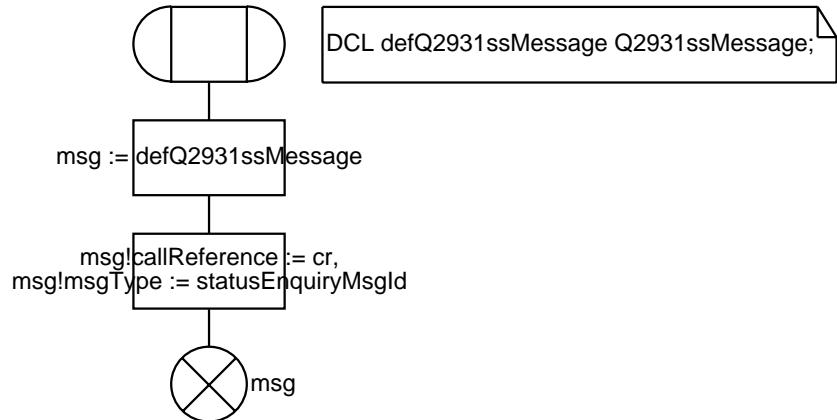


Annex B: createStatusEnquiry

Procedure createStatusEnquiry

1(1)

```
;fpar cr CallReference;
returns msg Q2931ssMessage;
```

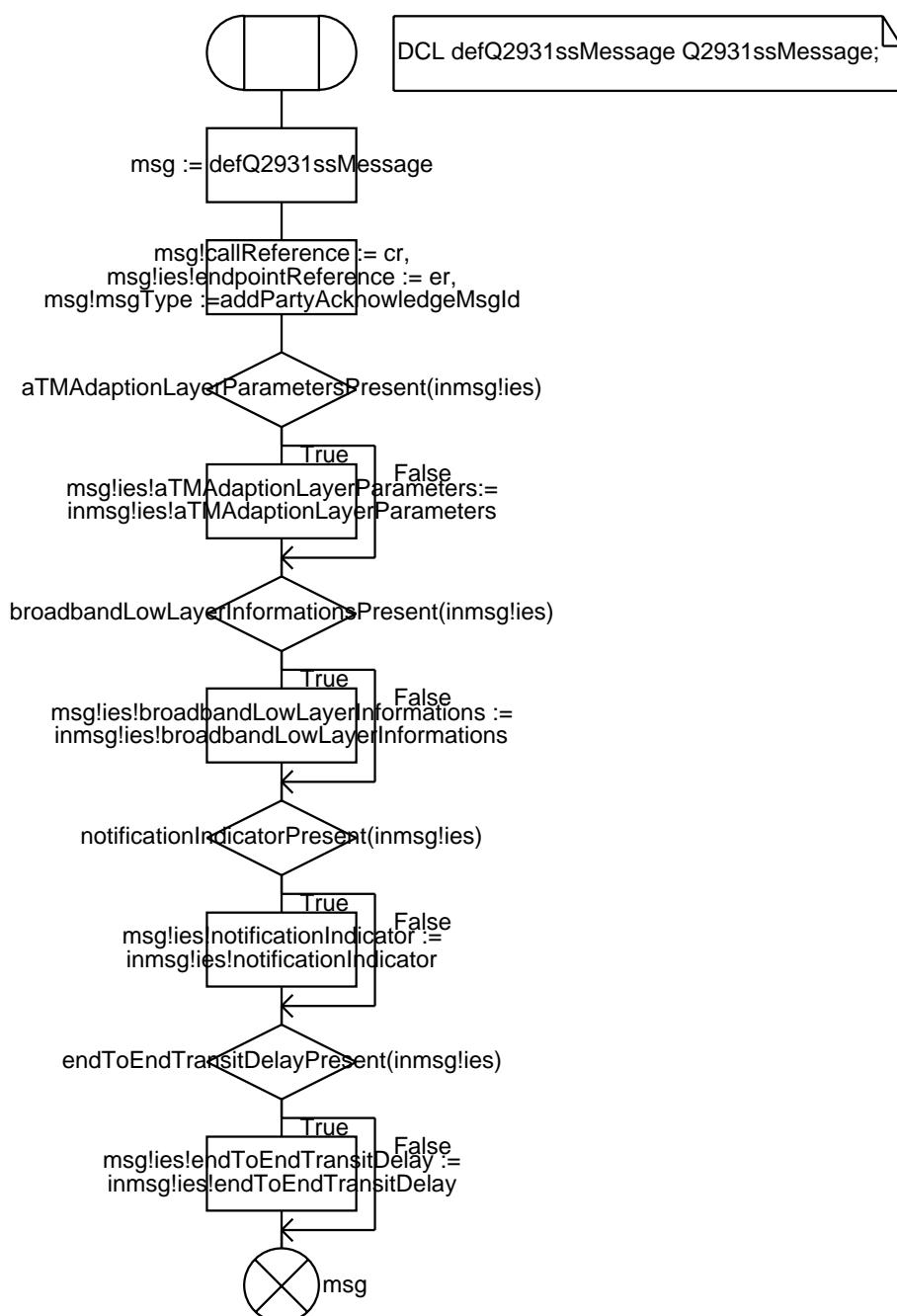


Annex B: createAddPartyAcknowledge

Procedure createAddPartyAcknowledge

1(1)

:FPAR cr CallReference, er EndpointReference, inmsg Q2931ssMessage;
 RETURNS msg Q2931ssMessage;

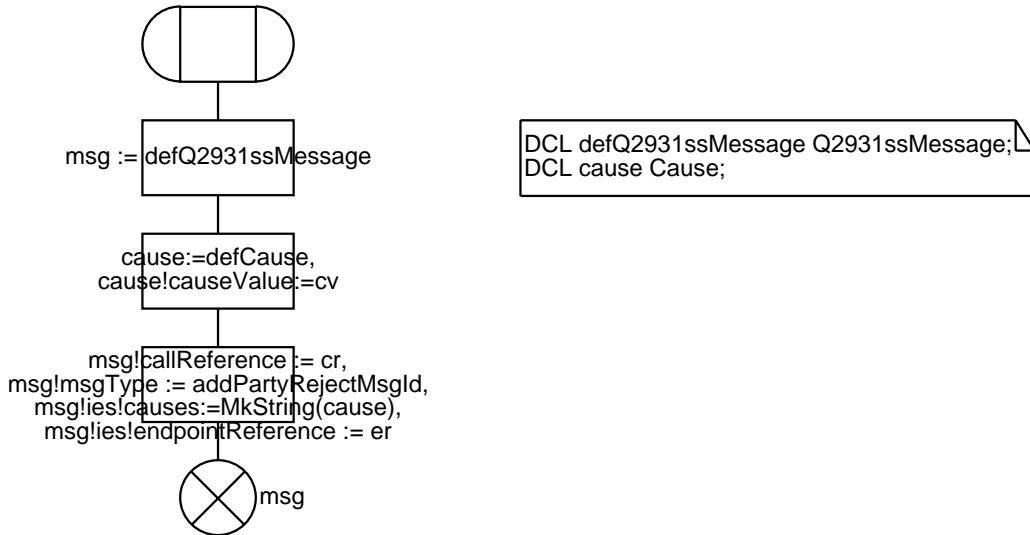


Annex B: createAddPartyReject

Procedure createAddPartyReject

1(1)

;fpar cr CallReference,er EndpointReference,cv CauseValue;
 returns msg Q2931ssMessage;

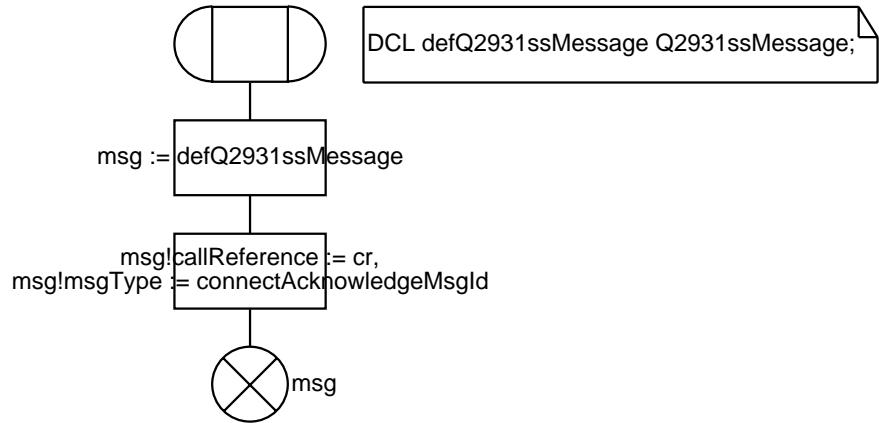


Annex B: createConnectAcknowledge

Procedure createConnectAcknowledge

1(1)

```
:FPAR cr CallReference;  
RETURNS msg Q2931ssMessage;
```

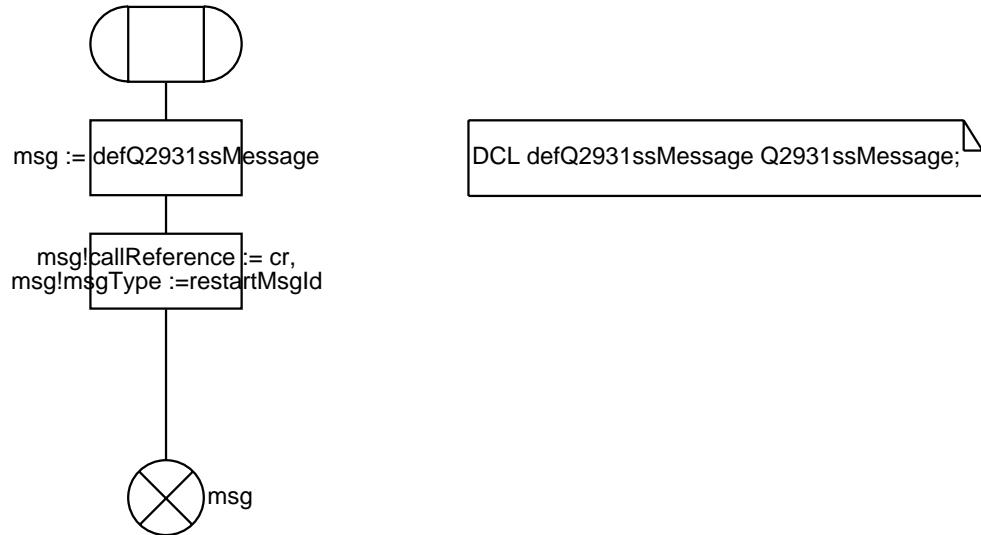


Annex B: createRestart

Procedure <<Block Type Q2931ssTypes_BT>> createRestart

1(1)

```
;fpar cr CallReference;  
returns msg Q2931ssMessage;
```

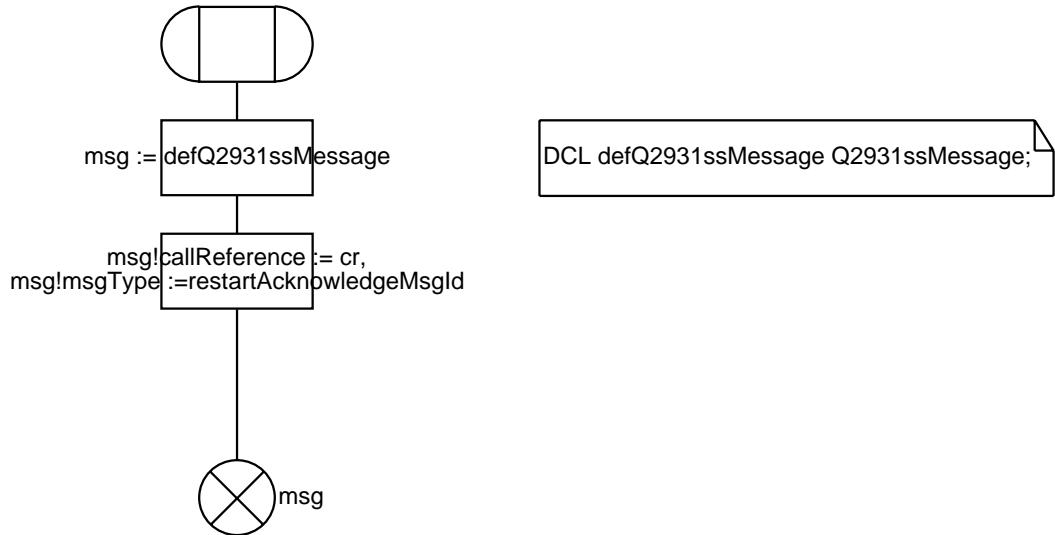


Annex B: createRestartAcknowledge

Procedure createRestartAcknowledge

1(1)

```
:fpar cr CallReference;
returns msg Q2931ssMessage;
```

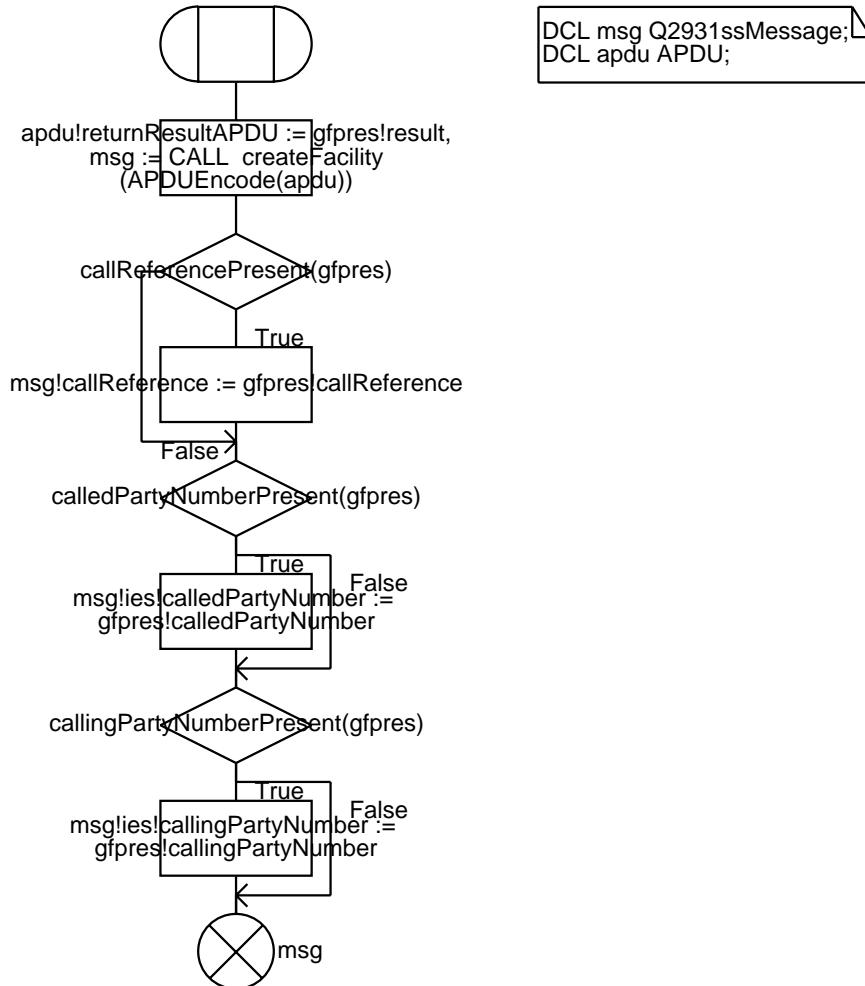


Annex B: processGFPResult

Procedure processGFPResult

1(1)

:FPAR gfpres GFPResultMessage;
 RETURNS Q2931ssMessage;

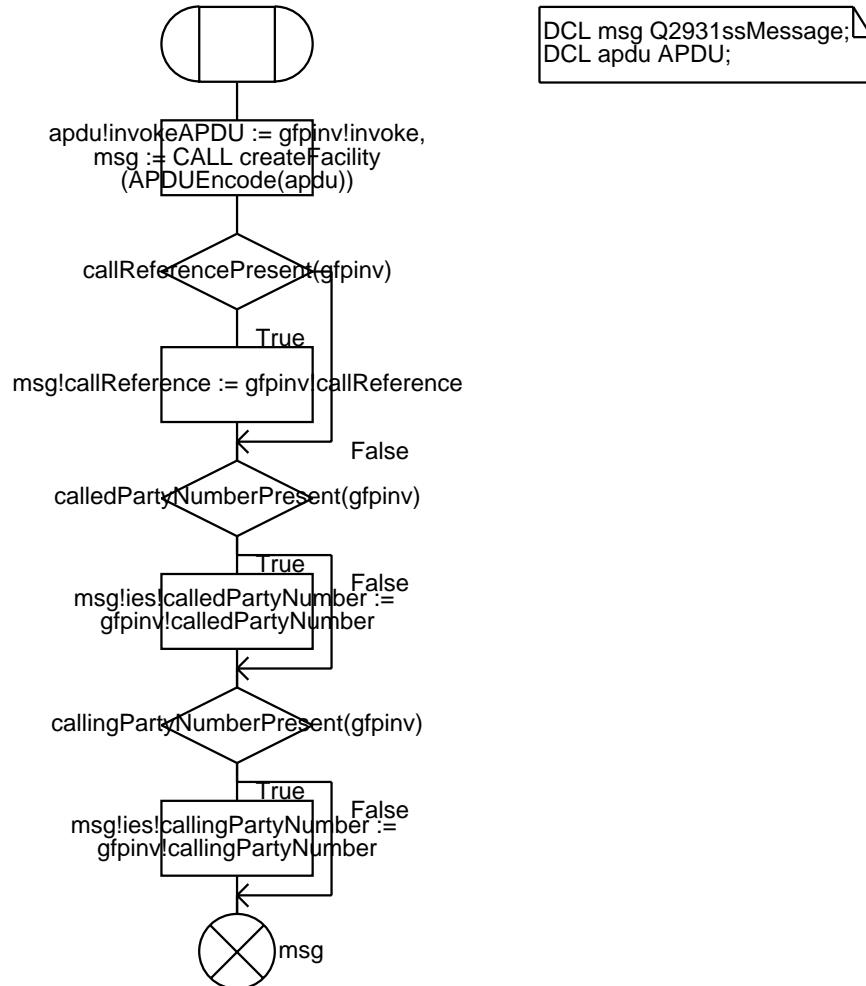


Annex B: processGFPInvoke

Procedure processGFPInvoke

1(1)

:FPAR gfpinv GFPInvokeMessage;
 RETURNS Q2931ssMessage;

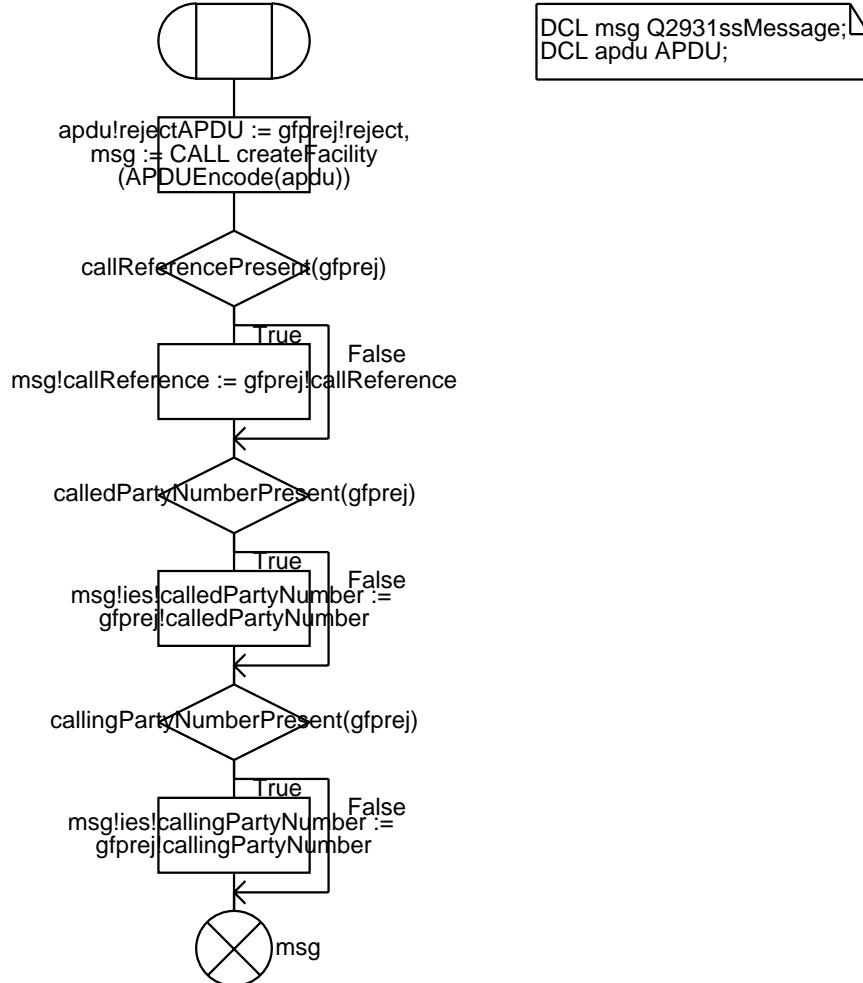


Annex B: processGFPReject

Procedure processGFPReject

1(1)

:FPAR gfprej GFPRejectMessage;
 RETURNS Q2931ssMessage;

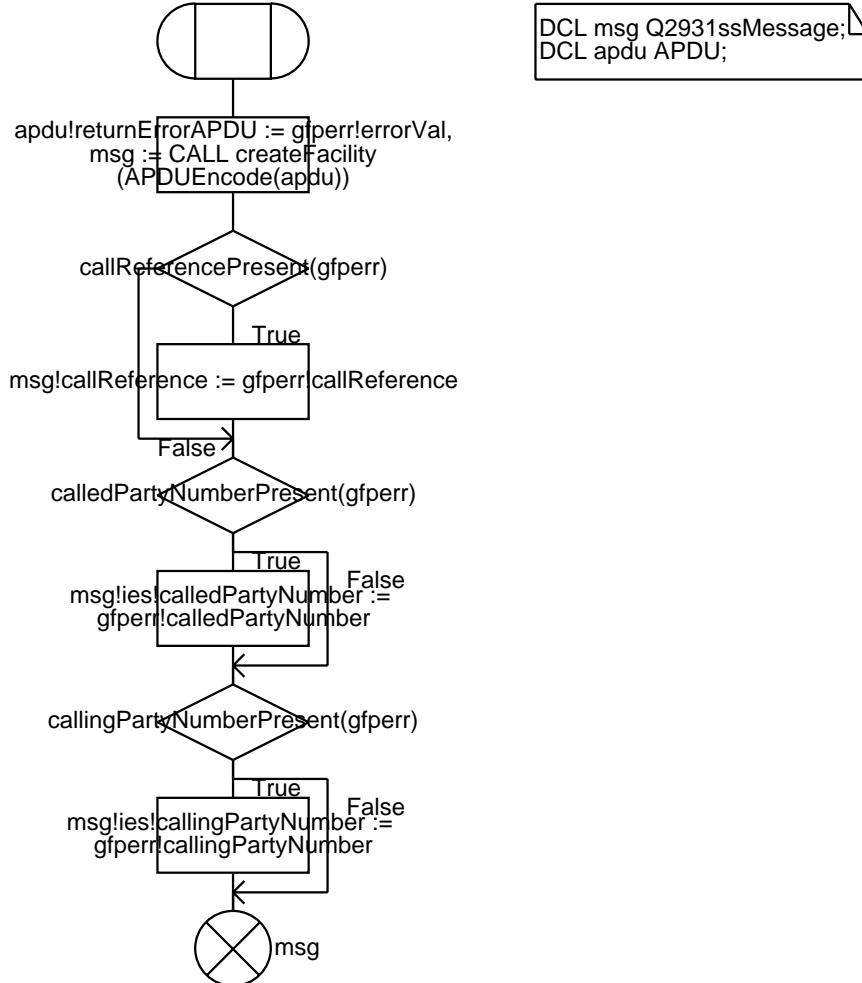


Annex B: processGFPError

Procedure processGFPError

1(1)

```
:FPAR gfperr GFPErrorMessage;
RETURNS Q2931ssMessage;
```



Annex B: processFacilityInd

Procedure processFacilityInd

1(4)

;FPAR in msg Q2931ssMessage, IN gfppartner PId, IN coord PId;

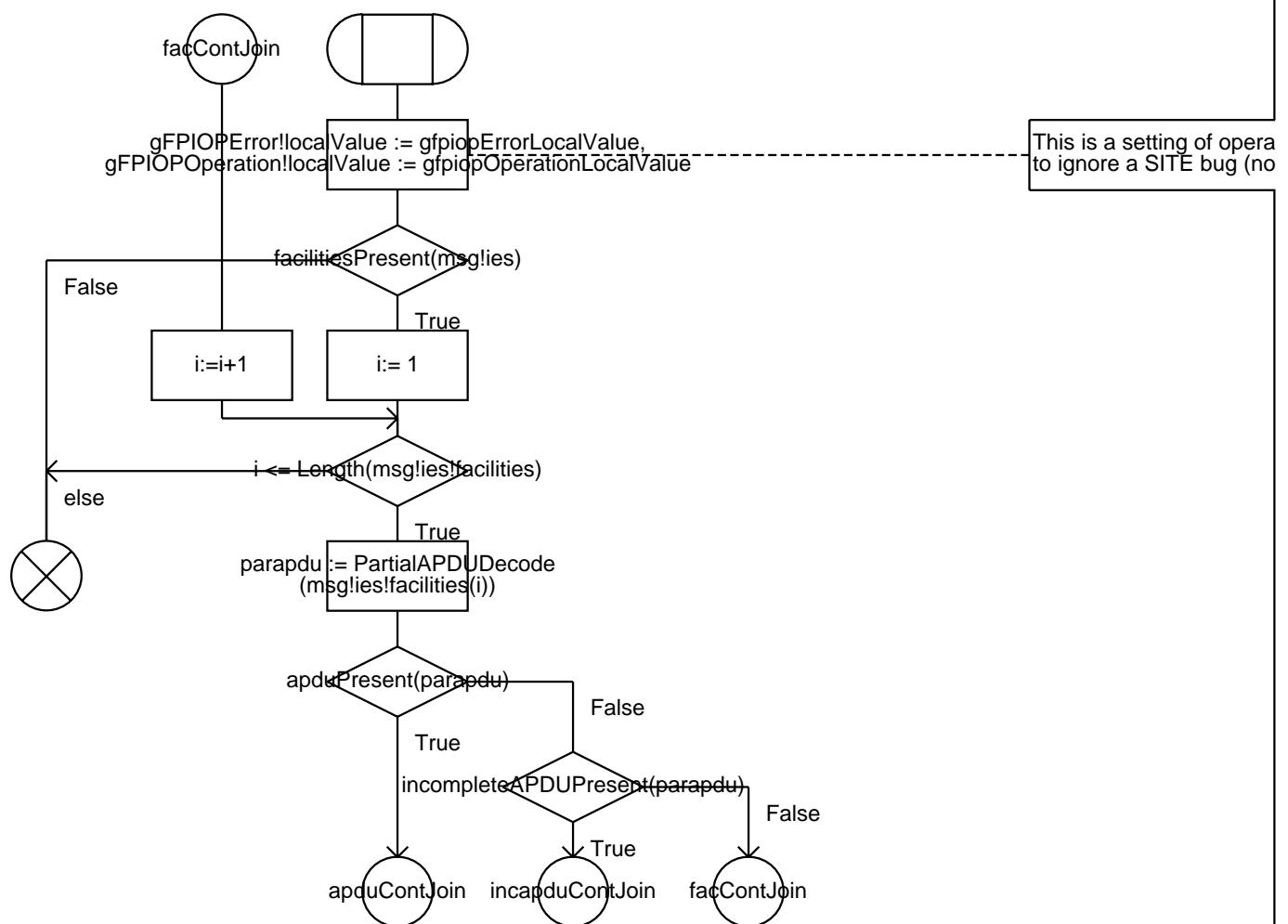
```
DCL i Integer;
DCL parapdu PartialAPDU;
DCL apdu APDU;
DCL gFPIOPOperation OperationOp;
DCL gFPIOPError ErrorOp;
DCL outmsg Q2931ssMessage;
```

Annex B: processFacilityInd

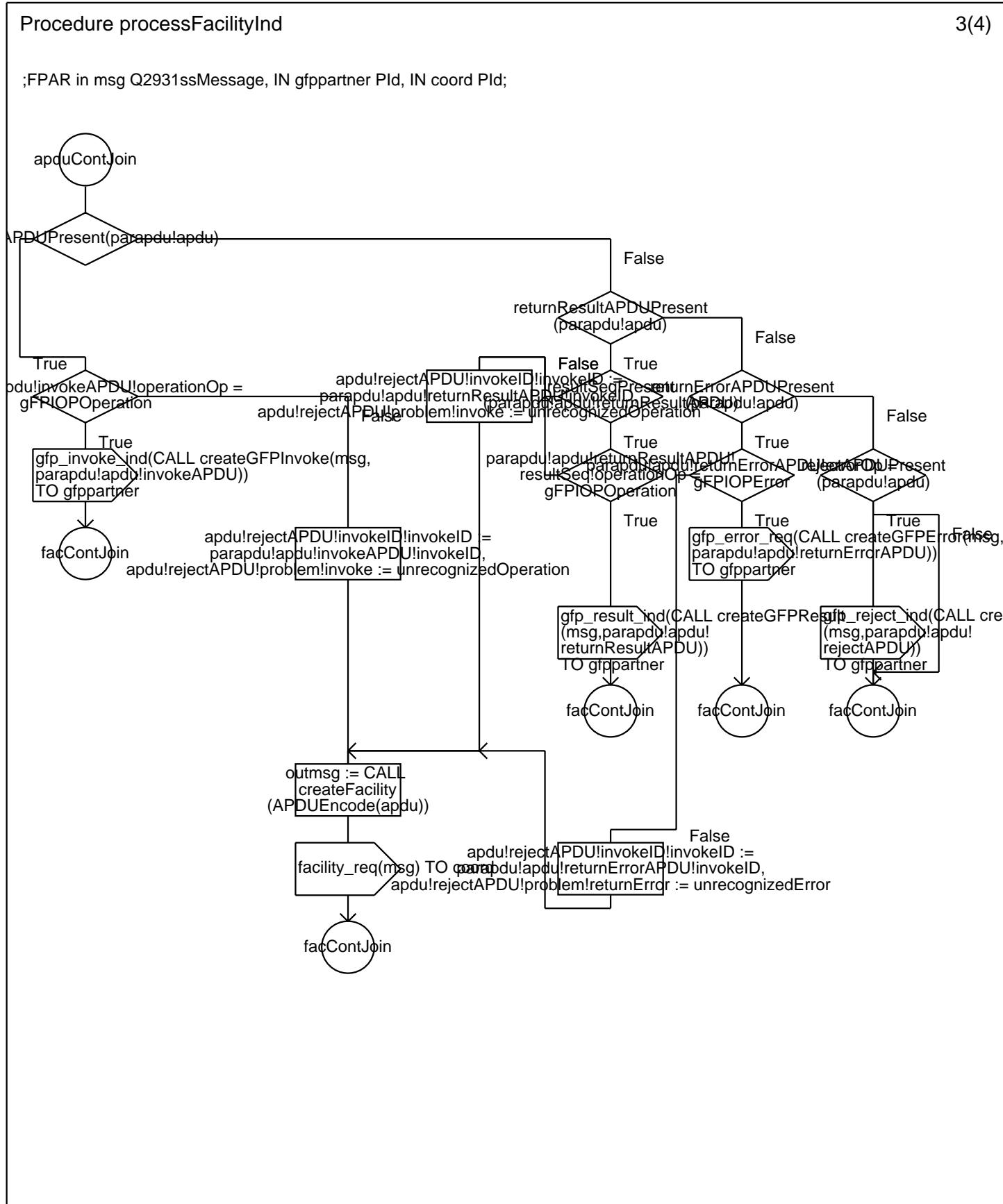
Procedure processFacilityInd

2(4)

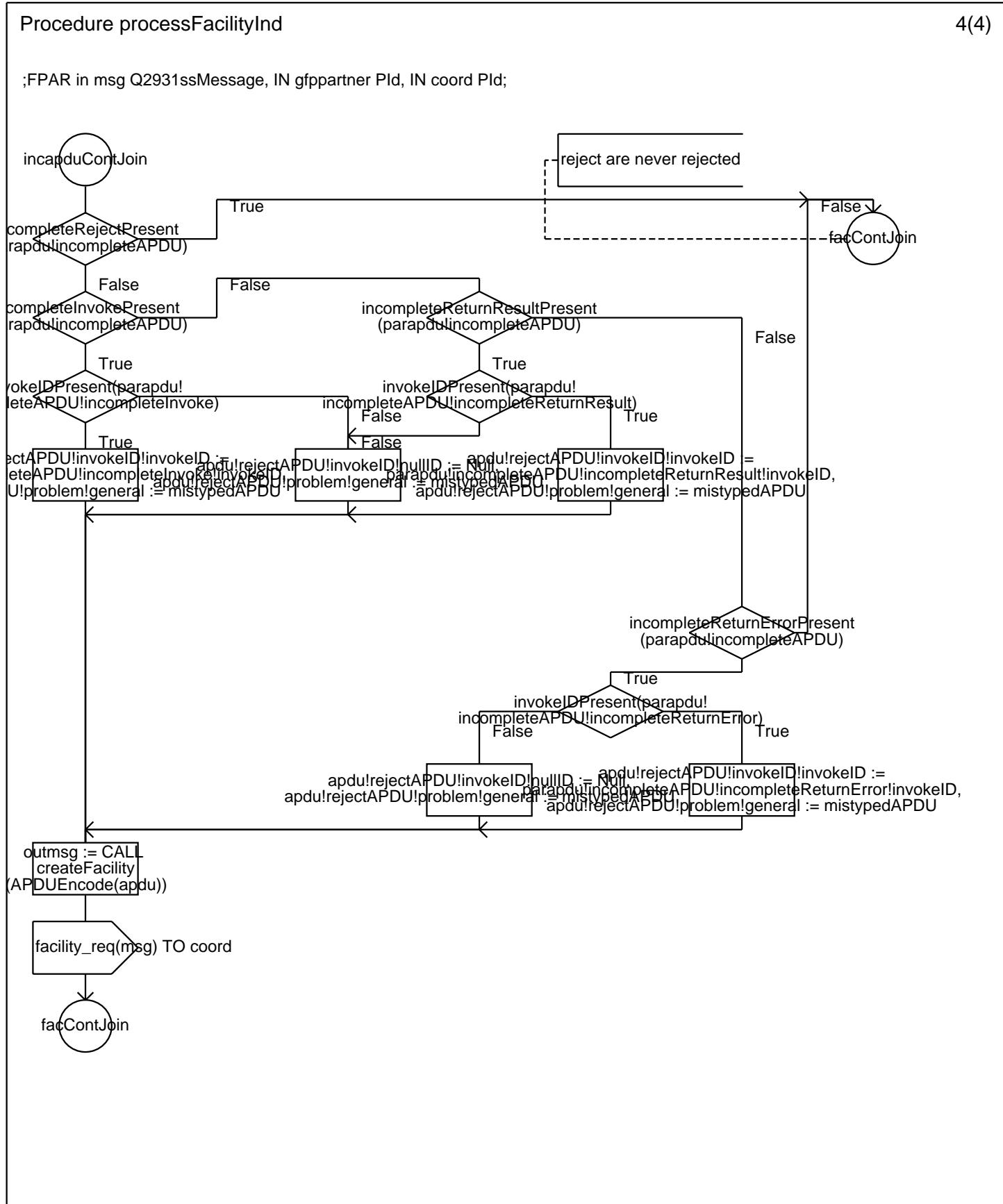
;FPAR in msg Q2931ssMessage, IN gfppartner PId, IN coord PId;



Annex B: processFacilityInd



Annex B: processFacilityInd

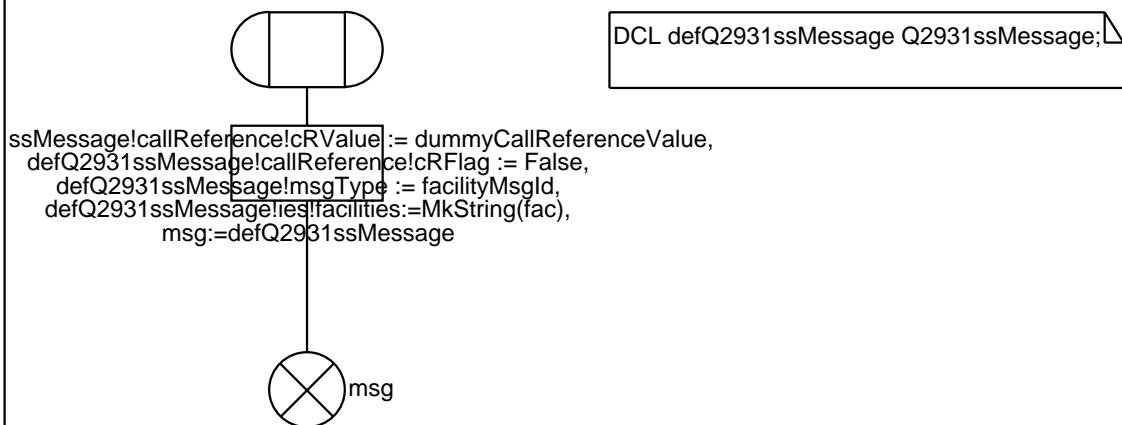


Annex B: createFacility

Procedure createFacility

1(1)

```
;FPAR fac GFPData;  
returns msg Q2931ssMessage;
```



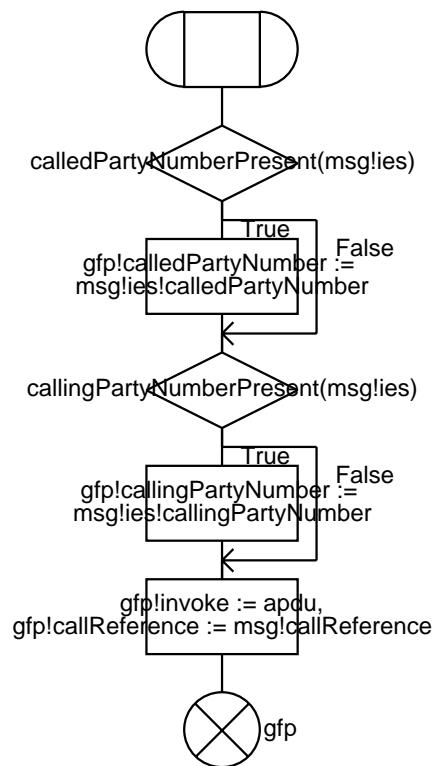
Annex B: createGFPIvoke

Procedure createGFPIvoke

1(1)

:FPAR msg Q2931ssMessage, apdu InvokeAPDU;
 RETURNS GFPIvokeMessage;

DCL gfp GFPIvokeMessage;

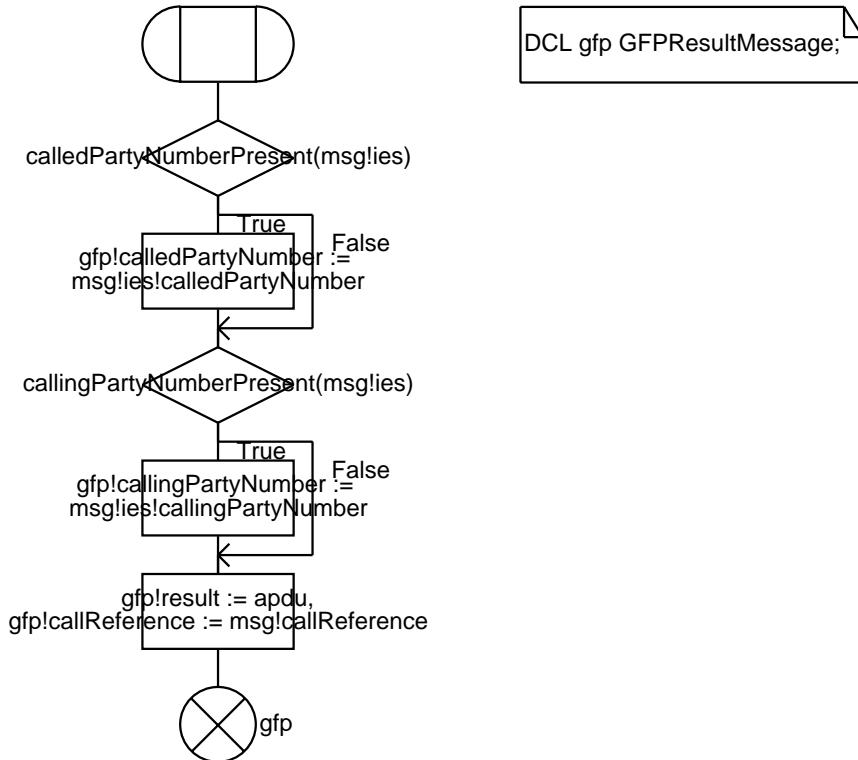


Annex B: createGFPResult

Procedure createGFPResult

1(1)

```
:FPAR msg Q2931ssMessage, apdu  ReturnResultAPDU;  
RETURNS GFPResultMessage;
```

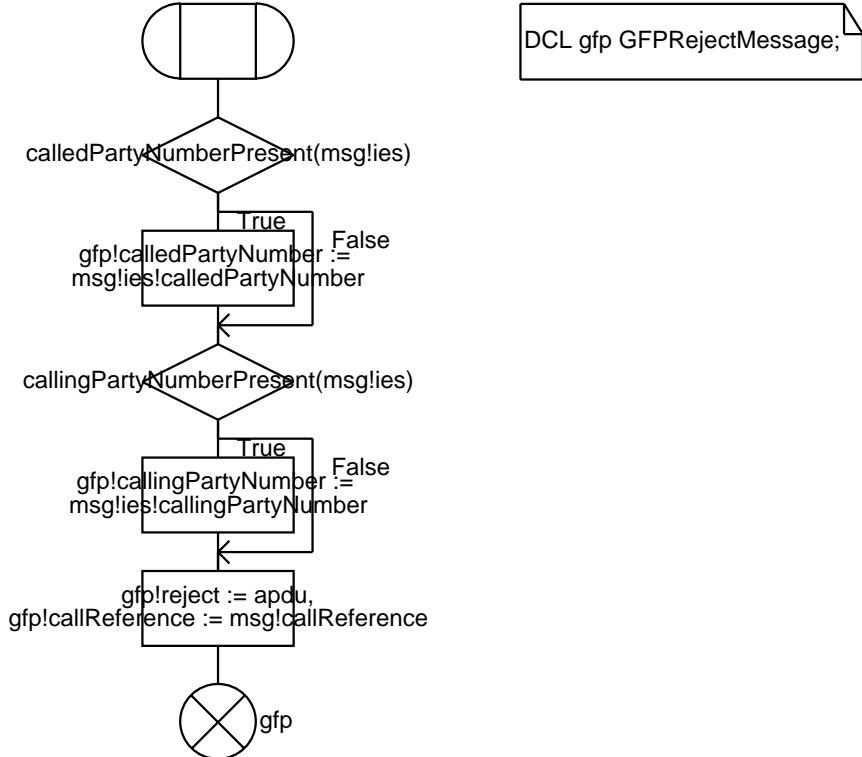


Annex B: createGFPReject

Procedure createGFPReject

1(1)

:FPAR msg Q2931ssMessage, apdu RejectAPDU;
 RETURNS GFPRejectMessage;

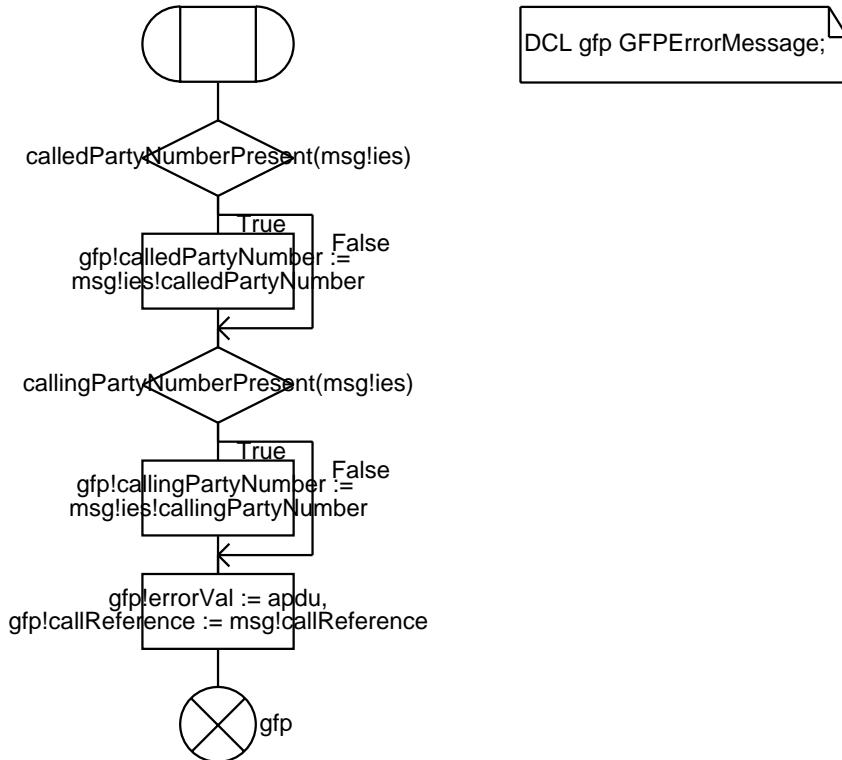


Annex B: createGFPError

Procedure createGFPError

1(1)

```
:FPAR msg Q2931ssMessage, apdu ReturnErrorAPDU;
RETURNS GFPErrorMessage;
```



Annex B: Q2931ssTest

USE SDTCoding;

Package Q2931ssTest

1(1)

```
/*
This package defines system types for
the simulation of
- Q.2931/71 network side (Q2931ssProtocolTest_ST)
- Q.2931/71 user side and (q2931ssUserTest_ST)
the validation/test suite generation of
- Q.2931/71 network side (Q2931ssNetVal_ST).
```

Author: PT87/ETSI, Nils Fischbeck/HUB
Version: 2.1
Last Change: 20.11.97

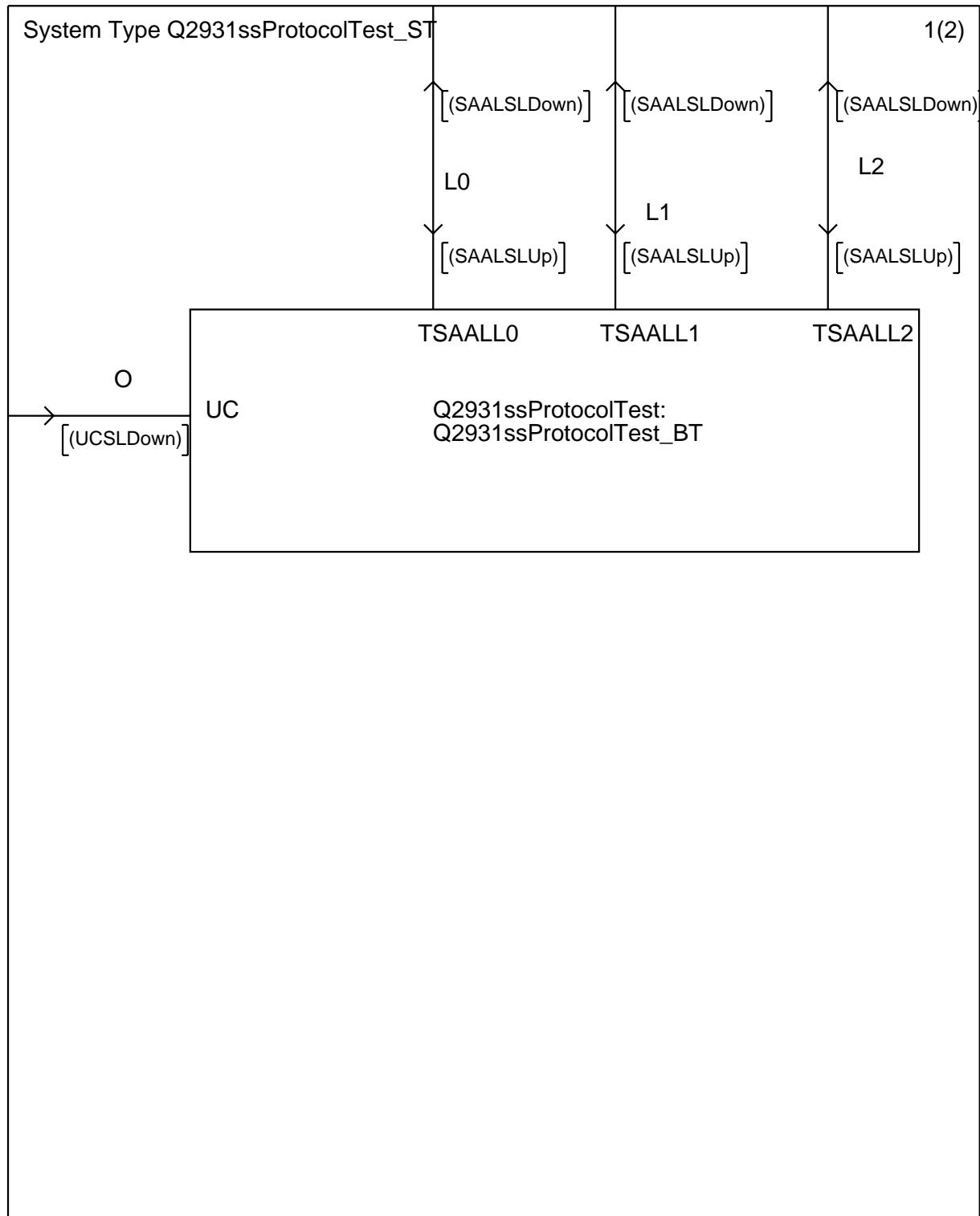
```
*/
```

Q2931ssProtocolTest_ST

Q2931ssUserTest_ST

Q2931ssNetVal_ST

Annex B: Q2931ssProtocolTest_ST



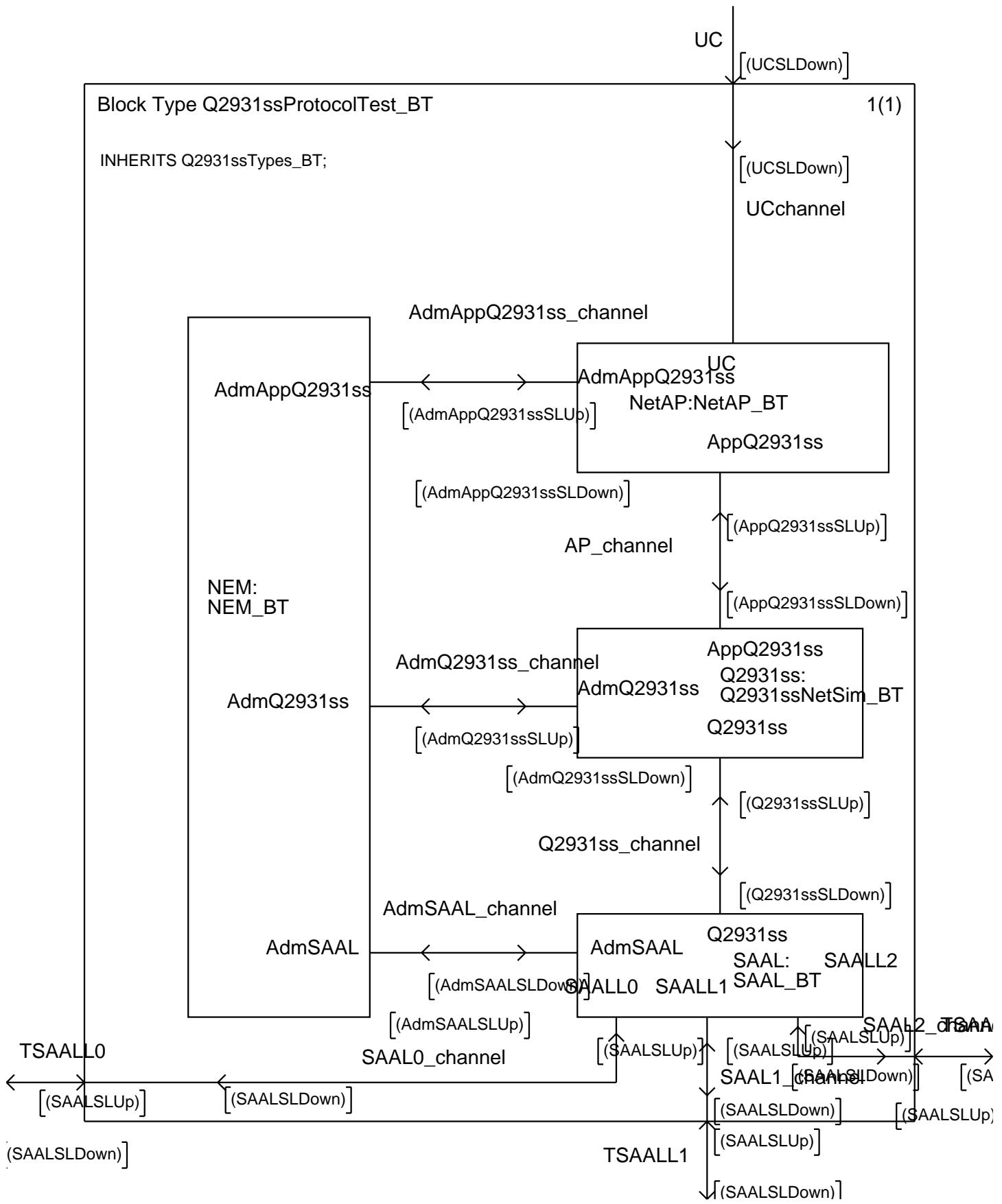
Annex B: Q2931ssProtocolTest_ST

System Type Q2931ssProtocolTest_ST

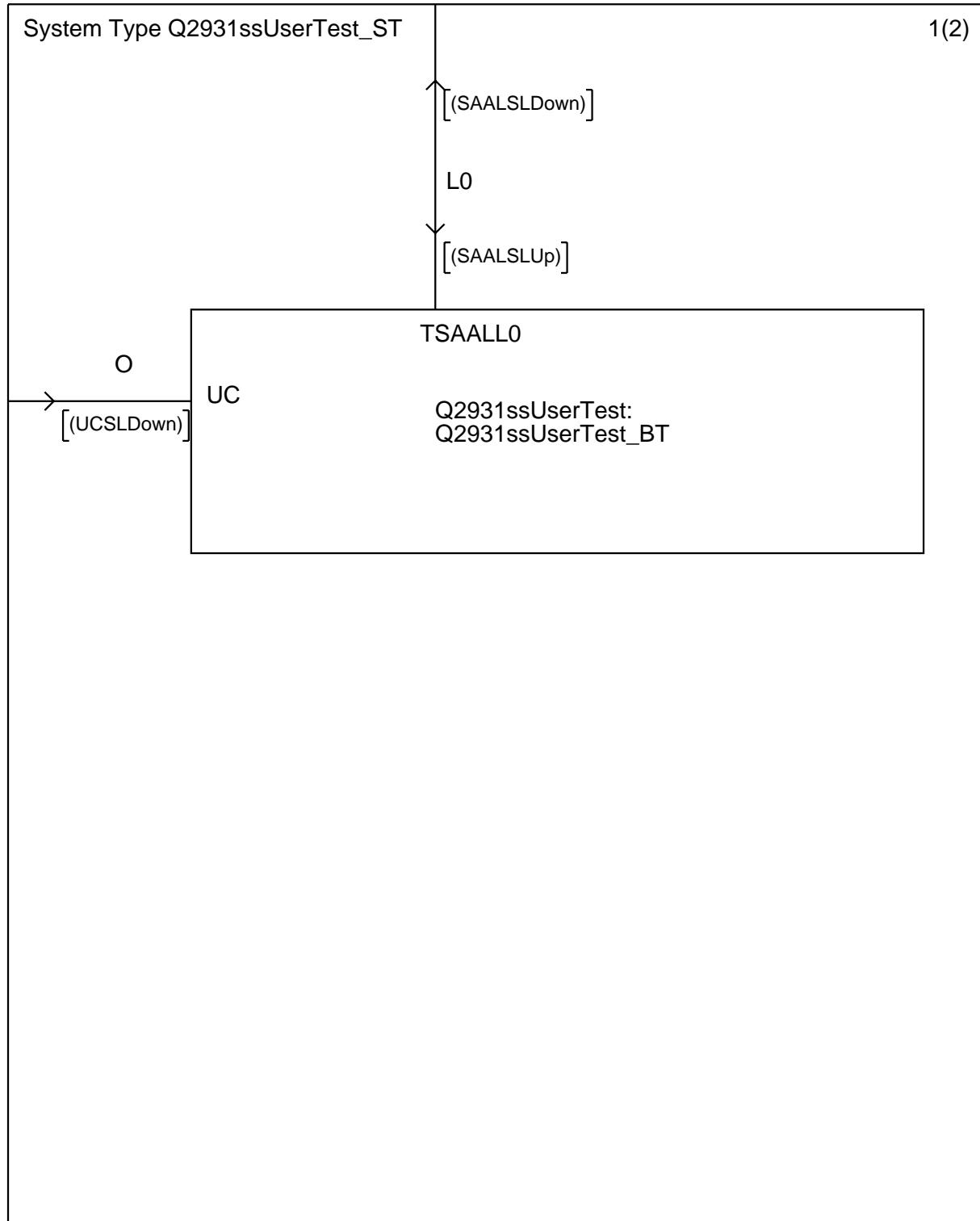
2(2)

Q2931ssProtocolTest_BT

Annex B: Q2931ssProtocolTest_BT



Annex B: Q2931ssUserTest_ST



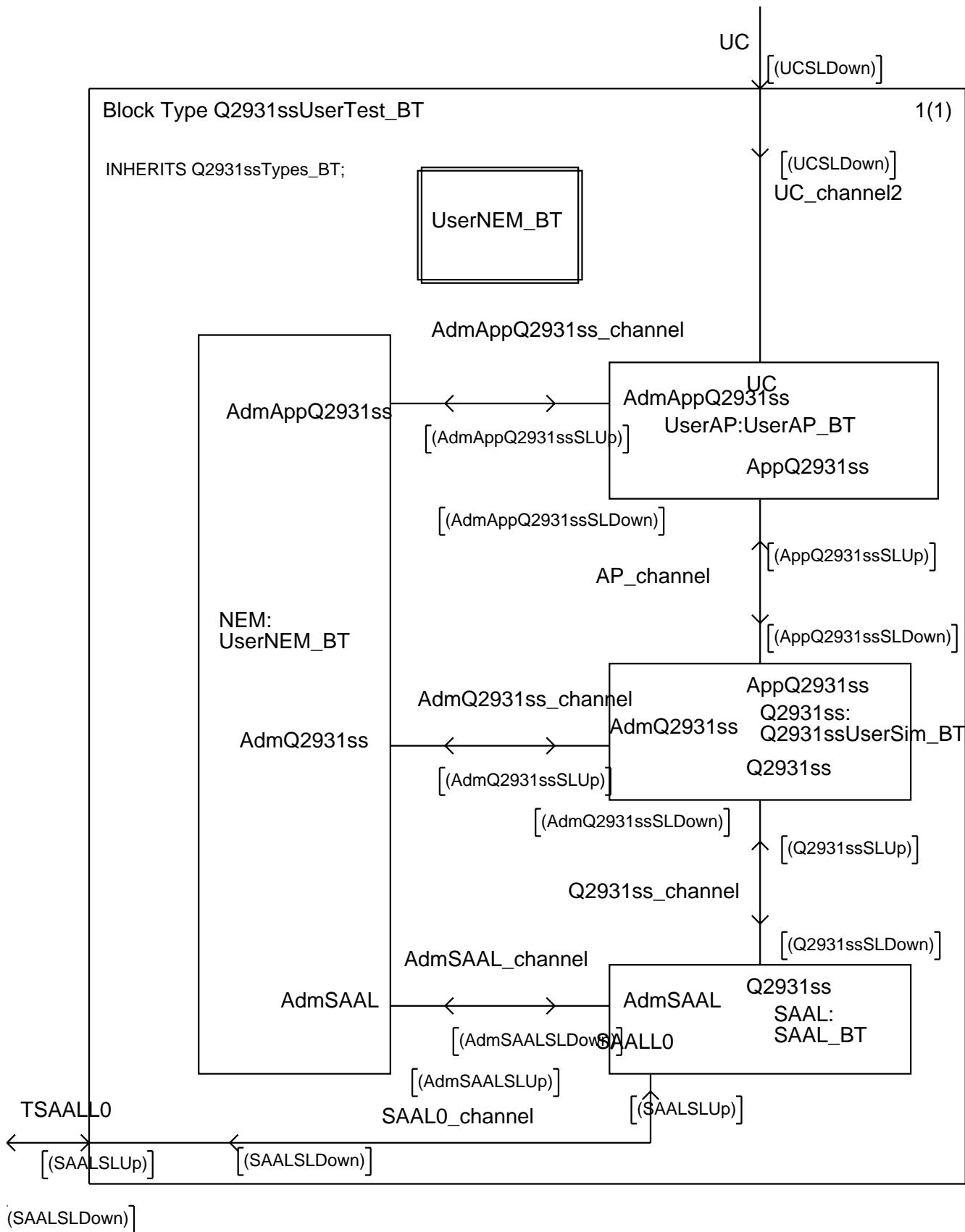
Annex B: Q2931ssUserTest_ST

System Type Q2931ssUserTest_ST

2(2)

Q2931ssUserTest_BT

Annex B: Q2931ssUserTest_BT

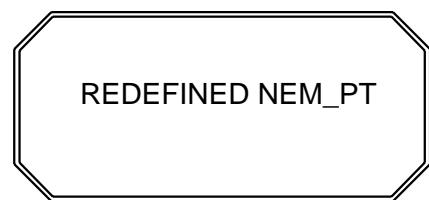


Annex B: UserNEM_BT

Block Type UserNEM_BT

1(1)

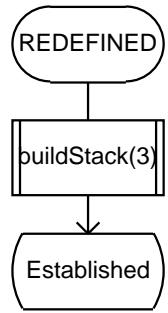
INHERITS NEM_BT;



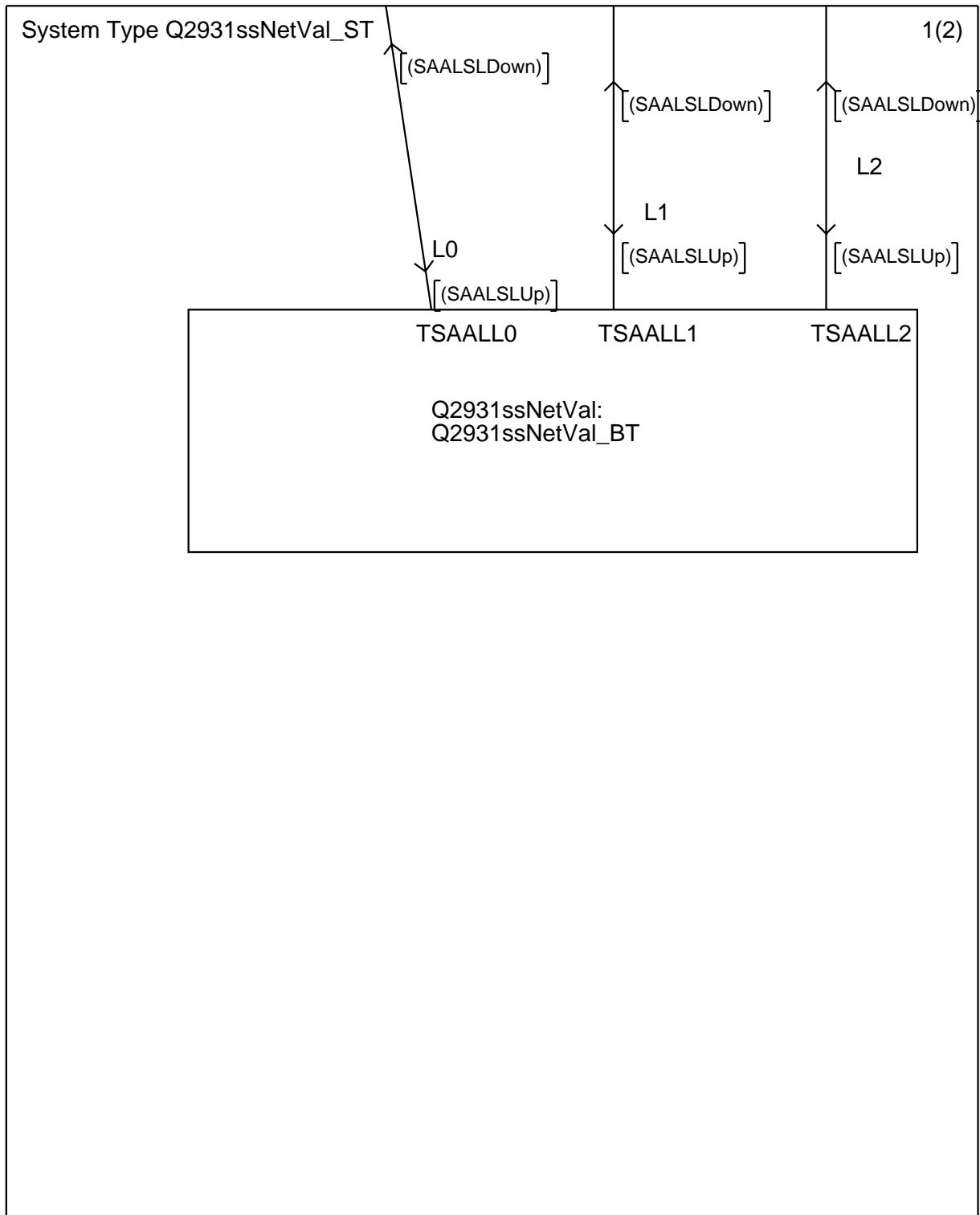
Annex B: NEM_PT

Redefined Process Type <<Block Type UserNEM_BT>> NEM_PT

1(1)



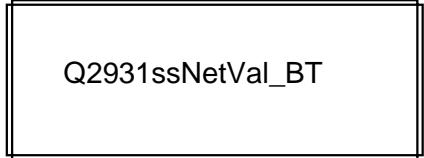
Annex B: Q2931ssNetVal_ST



Annex B: Q2931ssNetVal_ST

System Type Q2931ssNetVal_ST

2(2)



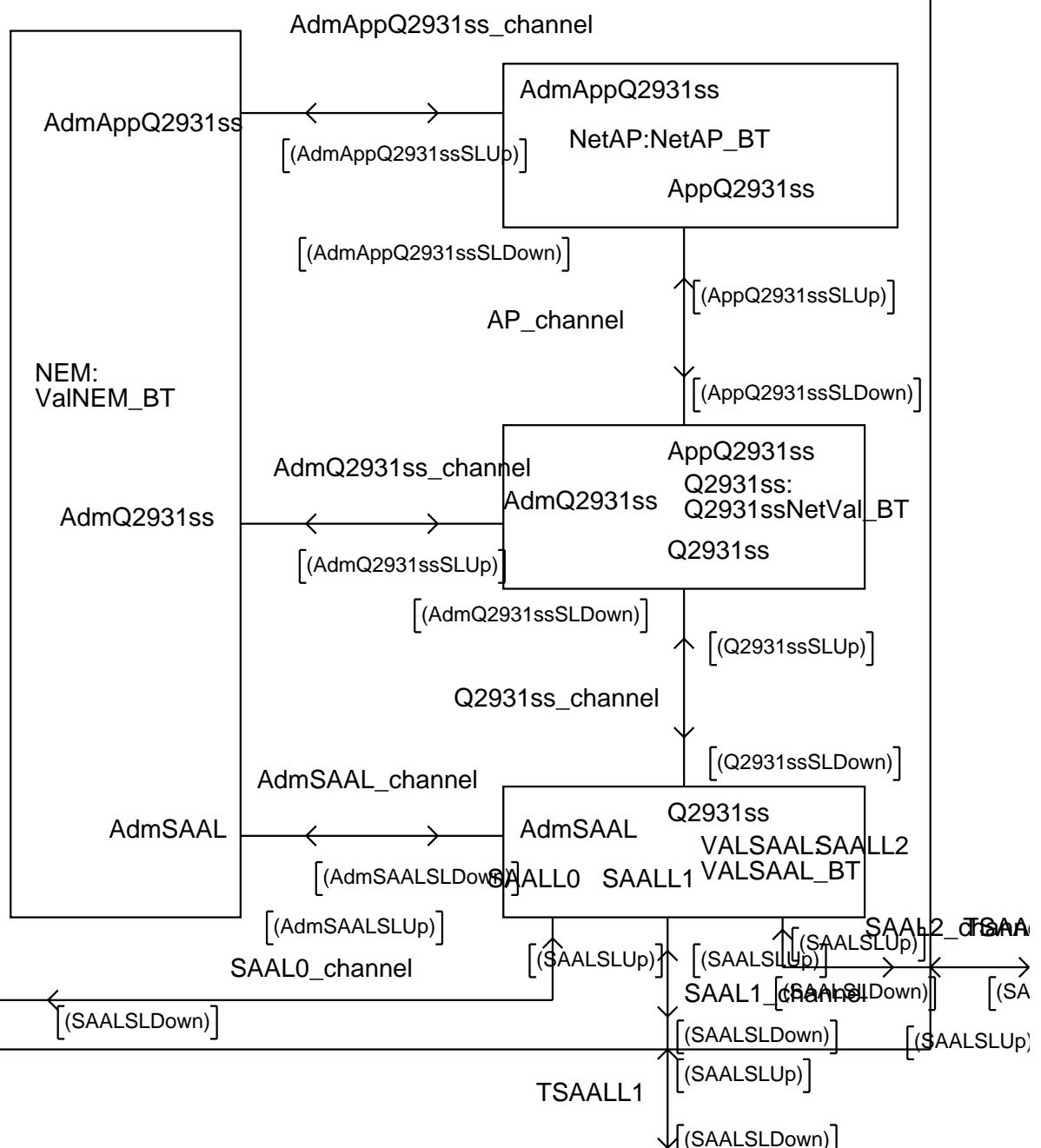
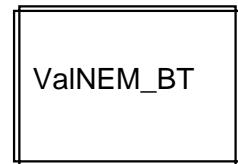
Q2931ssNetVal_BT

Annex B: Q2931ssNetVal_BT

Block Type Q2931ssNetVal_BT

1(1)

INHERITS Q2931ssTypes_BT;

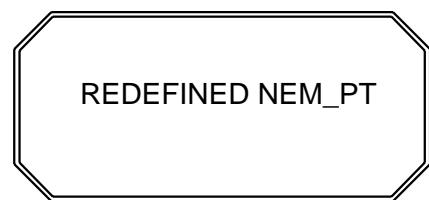


Annex B: ValNEM_BT

Block Type ValNEM_BT

1(1)

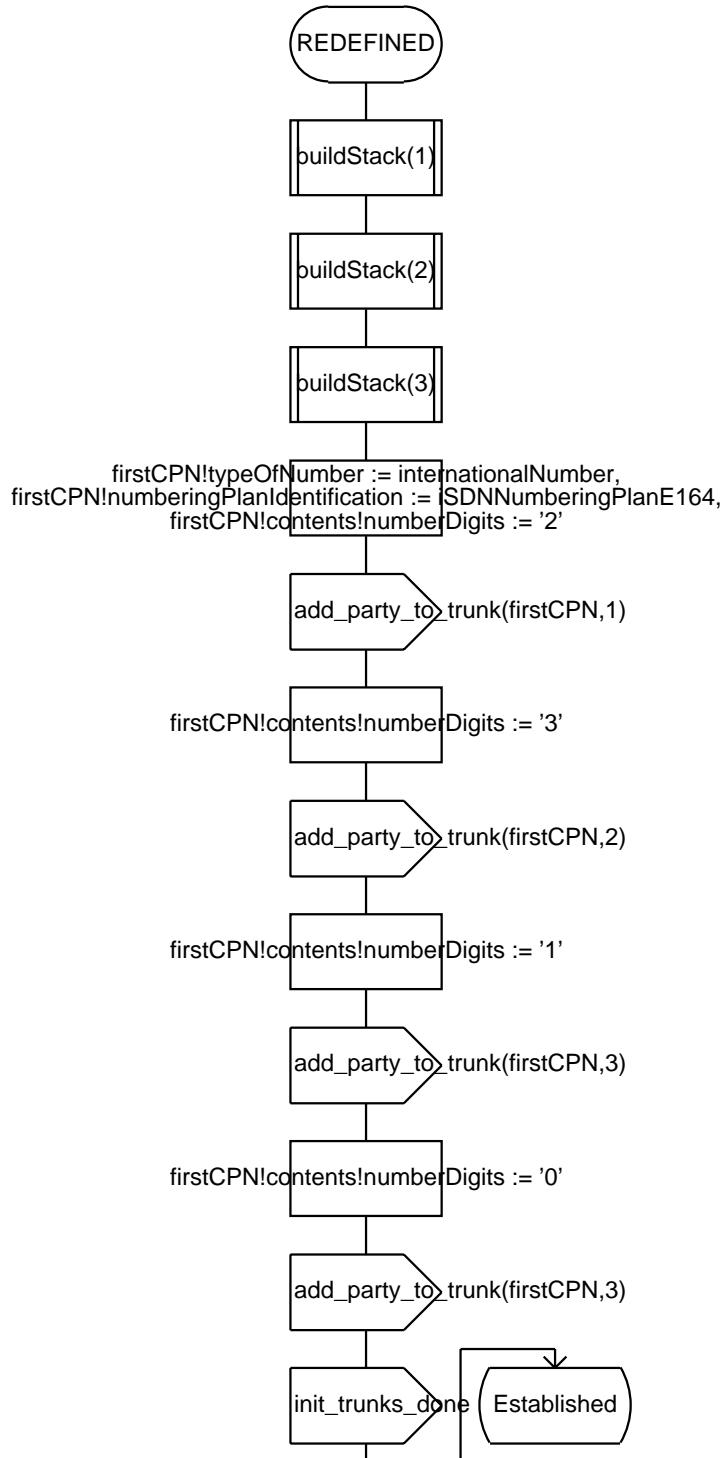
INHERITS NEM_BT;



Annex B: NEM_PT

Redefined Process Type <<Block Type ValNEM_BT>> NEM_PT

1(1)



Annex B: ProtocolTest

USE Q2931ssTest;

System ProtocolTest:Q2931ssProtocolTest_ST

1(1)

```
/*
This system provides a model of B-ISDN DSS2 (Q.2931)
with point to multipoint extension (Q.2971).
This version does contain the network
side only.
```

```
Author: ETSI, PT87
Version: 1.0
Date: 5/12/96
*/
```

Annex B: UserTest

USE Q2931ssTest;

System UserTest:Q2931ssUserTest_ST

1(1)

```
/*
This system provides a model of B-ISDN DSS2 (Q.2931)
with point to multipoint extension (Q.2971).
This version does contain the network
side only.
```

```
Author: ETSI, PT87
Version: 1.0
Date: 5/12/96
*/
```

Annex B: ProtocolVal

USE Q2931ssTest;

System ProtocolVal:Q2931ssNetVal_ST

1(1)