Table of Contents

Organizer View 1
ASN.1 Text DataModule 3
Package VB5_2 23
Block Type AN 51
Block AN_IMPLICITOR 52

Process AN_Router 53
Block AN_BCCMGT 61
Process AN_CTR 63
Process AN_PRESYNC 68
Block AN_BCCPROT 69

Process AN_PROT_MGR 72
Process AN_PROT_FSM 94
Block AN_CONNMGT 109
Process AN_CONN_MGR 112
Process AN_CONN_FSM 125

Process AN_BRANCH_FSM 138
Block AN_GLUER 142
Process AN_GLUE 143
Block Type SN 159
Block SN_IMPLICITOR 160

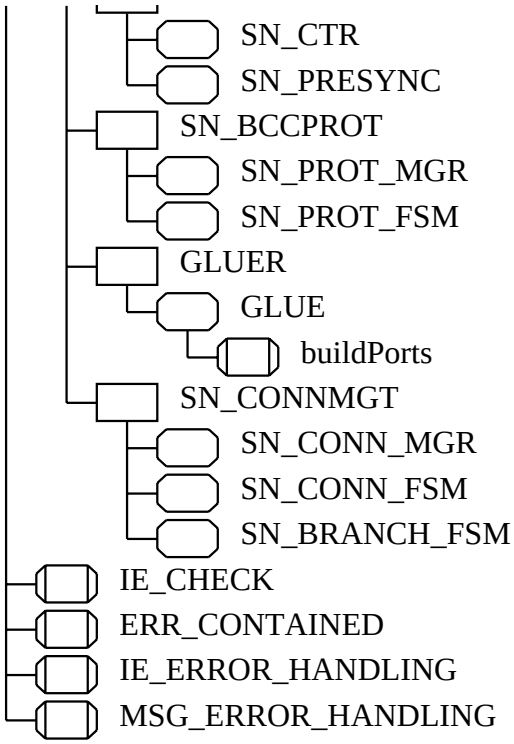
Process SN_Router 161
Block SN_BCCMGT 170
Process SN_CTR 172
Process SN_PRESYNC 180
Block SN_BCCPROT 181

Process SN_PROT_MGR 184
Process SN_PROT_FSM 198
Block GLUER 217
Process GLUE 218
Procedure buildPorts 240

Block SN_CONNMGT 242
Process SN_CONN_MGR 246
Process SN_CONN_FSM 259
Process SN_BRANCH_FSM 269
Procedure IE_CHECK 271

Procedure ERR_CONTAINED 272
Procedure IE_ERROR_HANDLING 273
Procedure MSG_ERROR_HANDLING 274

SDT		rw	/export/home/stf/stf151/Final_SDL_TTCN/System/ANorSN.sdt
		rw	/export/home/stf/stf151/Final_SDL_TTCN/System/
AN_VR2		rw	AN_VR2.ssy
x:y AN_VB5_2 : AN			
SN_VR2		rw	SN_VR2.ssy
x:y SN_VB5_2 : SN			
Analysis Model	3		
ASN DataModule	3	rw	vb52_rev15.asn
Used Files	23		
SDL System Structure	23		
TTCN Test Specification	23		
VB5_2	23	rw	VB5_2.sun
ASN DataModule			
AN	51	rw	AN.sbt
AN_IMPLICITOR	52	rw	AN_IMPLICITOR.sbk
AN_Router	53	rw	AN_Router.spr
AN_BCCMGT	61	rw	AN_BCCMGT.sbk
AN_CTR	63	rw	AN_CTR.spr
AN_PRESYNC	68	rw	AN_PRESYNC.spr
AN_BCCPROT	69	rw	AN_BCCPROT.sbk
AN_PROT_MGR	72	rw	AN_PROT_MGR.spr
AN_PROT_FSM	94	rw	AN_PROT_FSM.spr
AN_CONNMGT	109	rw	AN_CONNMGT.sbk
AN_CONN_MGR	112	rw	AN_CONN_MGR.spr
AN_CONN_FSM	125	rw	AN_CONN_FSM.spr
AN_BRANCH_FSM	138	rw	AN_BRANCH_FSM.spr
AN_GLUER	142	rw	AN_GLUER.sbk
AN_GLUE	143	rw	AN_GLUE.spr
SN	159	rw	SN.sbt
SN_IMPLICITOR	160	rw	SN_IMPLICITOR.sbk
SN_Router	161	rw	SN_Router.spr
SN_BCCMGT	170	rw	SN_BCCMGT.sbk



SN_CTR	172	rw	SN_CTR.spr
SN_PRESYNC	180	rw	SN_PRESYNC.spr
SN_BCCPROT	181	rw	SN_BCCPROT.sbk
SN_PROT_MGR	184	rw	SN_PROT_MGR.spr
SN_PROT_FSM	198	rw	SN_PROT_FSM.spr
GLUER	217	rw	GLUER.sbk
GLUE	218	rw	GLUE.spr
buildPorts	240	rw	buildPorts.spd
SN_CONNMGT	242	rw	SN_CONNMGT.sbk
SN_CONN_MGR	246	rw	SN_CONN_MGR.spr
SN_CONN_FSM	259	rw	SN_CONN_FSM.spr
SN_BRANCH_FSM	269	rw	SN_BRANCH_FSM.spr
IE_CHECK	271	rw	IE_CHECK.spd
ERR_CONTAINED	272	rw	ERR_CONTAINED.spd
IE_ERROR_HANDLING	273	rw	IE_ERROR_HANDLING.spd
MSG_ERROR_HANDLING	274	rw	MSG_ERROR_HANDLING.spd

Other Documents

ASN.1 Text DataModule

```
DataModule DEFINITIONS ::=
-- vb52_rev4.asn
-- This revision backtracks from version 3 to reintroduce specific typing for
-- PDUs rather than OPTIONAL IE fields.  These cannot be implemented in SDL.
-- Instead, one has to define a specific type for each IE combination one
-- wishes to pass to and from the environment.  22 Nov 99
-- This ASN.1 contains declarations which align with the SDL model data types.


BEGIN

ProtDiscr ::= OCTET STRING (SIZE(1))  --Always defaulted to '049'H, EN 301 005-4
, Annex E

TransId ::= SEQUENCE      -- Transaction Identifier, EN 301 005-4, Annex E
 {  transIdLength      [1] TransIdLength,
    transIdFlag         [2] TransIdFlag,
    transIdVal          [3] TransIdVal
 }
-- {  sparebits4          [1] SpareBits4, : DEFAULT '0000'B,
--    lengthOfTransIdVal  [2] BIT STRING (SIZE (4)), : DEFAULT '0011'B,

MsgType ::= OCTET STRING (SIZE (1))       --EN 301 005-4, Annex E

MsgCompatInd ::= OCTET STRING (SIZE (1))
-- SEQUENCE  EN 301 005-4, Annex E
-- {  extBitLast          [1] ExtBitLast,  DEFAULT '1'B,
--    spareBits2          [2] SpareBits2,  DEFAULT '00'B,
--    compatFlag          [3] CompatFlag,
--    spareBits21         [4] SpareBits2,  DEFAULT '00'B,
--    msgActInd           [5] MsgActInd
-- }

MsgLength ::= OCTET STRING (SIZE (2))      --EN 301 005-4, Annex E

TransIdFlag ::= BIT STRING (SIZE(1))       --EN 301 005-4, Annex E

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

TransIdVal ::= INTEGER --BIT STRING (SIZE (23))
    --EN 301 005-4, Annex E

MsgActInd ::= BIT STRING (SIZE(2))      --Table 35, EN 301 005-1
  -- msgActIndReject           MsgActInd ::= '00'B reject by PROTO_ERROR msg
  -- msgActIndDiscardAndIgnore MsgActInd ::= '01'B discard  and ignore
  -- msgActIndDiscardAndReport MsgActInd ::= '10'B not used by this VB5.1 versio
n
  -- All other values are reserved

SpareBits4 ::= BIT STRING (SIZE(4))    --Always defaulted to '0000'B, EN 301 005
-4, Annex E
SpareBits3 ::= BIT STRING (SIZE(3))    --Always defaulted to '000'B, EN 301 005-
4, Annex E
SpareBits2 ::= BIT STRING (SIZE(2))    --Always defaulted to '00'B, EN 301 005-4
, Annex E
SpareBits1 ::= BIT STRING (SIZE(1))    --Always defaulted to '0'B, EN 301 005-4,
 Annex E

ConnRefNoIE ::= SEQUENCE  -- EN 301 217-1, 14.3.6.2
 {       commonIEInfo     CommonIEInfo,
     connRefNoVal     ConnRefNoVal
 }
```

```
ConnRefNoVal ::= INTEGER -- OCTET STRING (SIZE (3))
     --EN 301 217-1, Figure 33

ConnRefNos ::= INTEGER
--SET SIZE (1..MAX) OF ConnRefNoVal SAM 15Sep

ConnRefNoListIE ::= SEQUENCE -- EN 301 217-1, 14.3.6.3
 {       commonIEInfo     CommonIEInfo,
     connRefNos     ConnRefNos  --SET SIZE (1..MAX) OF ConnRefNoVal, but now Integer
 }
     -- length of IE not to shall not exceed the maximum message length

UsrPortConnIdIE ::= SEQUENCE  -- EN 301 217-1, 14.3.6.4
 {       commonIEInfo           CommonIEInfo,
     usrPortConnIdIEOctet5     UsrPortConnIdIEOctet5,
     lgclUsrPortId          LogId,
     vpci               [1]     VPCI OPTIONAL,
     vci               [2]     VCI  OPTIONAL
 }


UsrPortConnIdIEOctet5 ::= OCTET STRING (SIZE (1))
--SEQUENCE EN 301 217-1, Table 54
-- {     extBitLast       [1]      ExtBitLast,   DEFAULT '1'B,
--     usrRscInd       [2]     UsrRscInd,
--     dirFlg          [3]     DirFlg,
--     connIdFlg     [4]     ConnIdFlg
-- }

UsrRscInd ::= BIT STRING (SIZE (3))  -- EN 301 217-1, Table 54
--     completeLUP          UsrRscInd ::= '000'B  Complete LUP
--     vPCatLUP          UsrRscInd ::= '001'B  VPC at the LUP
--     vCCatLUP          UsrRscInd ::= '010'B  VCC at the LUP
     -- all other values are reserved

DirFlg ::= BIT STRING (SIZE (2))  -- EN 301 217-1, Table 54
--     usrOrigin     DirFlg ::= '00'B  User port is bearer connect origin
--     usrTerminates     DirFlg ::= '01'B  User port is bearer connection termination
     -- all other values are reserved

ConnIdFlg ::= BIT STRING (SIZE (2)) -- EN 301 217-1, Table 54
--     exclCombo     ConnIdFlg ::= '00'B  exclusive VPCI/VCI combination
--     prefCombo     ConnIdFlg ::= '01'B  preferred VPCI/VCI combination
     -- all other values are reserved

LogId ::= OCTET STRING (SIZE (3))     --logical port identifier
  --LogicalServicePortId  (0 .. 16777215)
  --LogicalUserPortId     (0 .. 16777215)

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

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

SrvcPortConnIdIE ::= SEQUENCE  -- EN 301 217-1, 14.3.6.5
 {       commonIEInfo           CommonIEInfo,
     srvcPortConnIdIEOctet5     SrvcPortConnIdIEOctet5,
     vpci               [1]     VPCI OPTIONAL,
     vci               [2]     VCI OPTIONAL
 }


SrvcPortConnIdIEOctet5 ::= OCTET STRING (SIZE (1))
-- SEQUENCE  EN 301 217-1, Table 55
-- {     extBitLast       [1]      ExtBitLast,  DEFAULT '1'B,
```

```
--     srvcRscInd       [2]     SrvcRscInd,
--     spare          [3]     SpareBits2,
--     connIdFlg     [4]     ConnIdFlg
-- }

SrvcRscInd ::= BIT STRING (SIZE (3))  -- EN 301 217-1, Table 55
--     completeLSP          SrvcRscInd ::= '000'B   Complete LSP
--     vPCatLSP          SrvcRscInd ::= '001'B   VPC at the LSP
--     vCCatLSP          SrvcRscInd ::= '010'B   VCC at the LSP
     -- all other values are reserved

AltUsrPortVPCIIE ::= SEQUENCE  -- EN 301 217-1, 14.3.6.6
 {       commonIEInfo           CommonIEInfo,
     vpcis               VPCI -- SET SIZE (1..MAX) OF VPCI
 }
     -- length of IE shall not exceed the maximum message length

AltSrvcPortVPCIIE ::= SEQUENCE -- EN 301 217-1, 14.3.6.7
 {       commonIEInfo           CommonIEInfo,
     vpcis               VPCI -- SET SIZE (1..MAX) OF VPCI
 }
     -- length of IE shall not exceed the maximum message length


AutoCongLvlIE ::= SEQUENCE -- EN 301 217-1, 14.3.6.8
 {       commonIEInfo           CommonIEInfo,
     congLvl               CongLvl
 }

CongLvl ::= OCTET STRING (SIZE (8)) --Table 56, EN 301 217-1
--      congLvl1               CongLvl ::= '00000001'B
--      congLvl2               CongLvl ::= '00000010'B
     -- all other values are reserved


RejCauseIE ::= SEQUENCE -- EN 301 217-1, 14.3.6.9
 {       commonIEInfo           CommonIEInfo,
     rejCauseOctet          INTEGER -- RejCauseOctet
 }

RejCauseOctet ::= SEQUENCE  -- EN 301 217-1, Figure 38
 {         extBitLast     [1]      ExtBitLast DEFAULT '1'B,
     rejCause     [2]     RejCause
 }

RejCause ::= BIT STRING (SIZE (7)) -- Table 57, EN 301 217-1
--      unspcfdErr          RejCause ::= '0000000'B
--     connStateIncmptblty     RejCause ::= '0000001'B
--     branchStateIncmptblty     RejCause ::= '0000010'B
--     branchDeallocExpctd     RejCause ::= '0000011'B
--     aNFault               RejCause ::= '0000100'B
--     insuffANRsrces          RejCause ::= '0000101'B
--     insuffUsrPortRsrces     RejCause ::= '0000110'B
--     insuffSrvcPortRsrces     RejCause ::= '0000111'B
--     unsupTrfcParamsCombo     RejCause ::= '0001000'B
--     unsupATMTrnsfrCap     RejCause ::= '0001001'B
--     usedUsrPortConnID     RejCause ::= '0001010'B
--     usedSrvcPortConnID     RejCause ::= '0001011'B
--     invalLUPId          RejCause ::= '0001100'B
--     invalUsrPortVPCI     RejCause ::= '0001101'B
--     invalUsrPortVCI          RejCause ::= '0001110'B
--     invalSrvcPortVPCI     RejCause ::= '0001111'B
--     invalSrvcPortVCI     RejCause ::= '0010000'B
     --all other values are reserved
```

```
BranchIdIE::= SEQUENCE  -- EN 301 217-1, 14.3.6.10
 {       commonIEInfo           CommonIEInfo,
     branchIdVal          BranchIdVal
 }

BranchIdVal ::= INTEGER -- OCTET STRING (SIZE (2))
     -- EN 301 217-1, Figure 39

BranchIdListIE  ::= SEQUENCE  -- EN 301 217-1, 14.3.6.11
 {       commonIEInfo           CommonIEInfo,
     branchIds          BranchIds -- SAM, 15sep99 SET SIZE (1..MAX) OF BranchIdVal
 }

BranchIds ::= INTEGER
--SET SIZE (1..MAX) OF BranchIdVal  SAM, 15sep99

ProtErrCauseIE ::= SEQUENCE -- EN 301 217-1, 14.3.7.10,  source: RTMC Protocol E
rror Cause
 {  commonIEInfo         CommonIEInfo,
    protErrCause         ProtErrCauseContent
 }

ProtErrCauseContent ::= INTEGER

-- ProtErrCauseContent ::= SEQUENCE
-- {  extBitLast          [1] ExtBitLast DEFAULT '1'B,
--    protErrCauseVal     [2] ProtErrCauseVal,
--    protErrCauseDiagn   [3] ProtErrCauseDiagn OPTIONAL
--  }

ProtErrCauseVal ::= BIT STRING (SIZE (7))
  unkMsgType                 ProtErrCauseVal ::= '0000001'B
  mandIEMiss                 ProtErrCauseVal ::= '0000011'B
  unrecogIE                  ProtErrCauseVal ::= '0000100'B
  iEContErr                  ProtErrCauseVal ::= '0000101'B
  iENotAllowed               ProtErrCauseVal ::= '0000110'B
  msgNotCompatWithProtState  ProtErrCauseVal ::= '0000111'B

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

QoSParamsIE  ::= SEQUENCE -- EN 301 217-1, 14.3.7.4
 {  commonIEInfo         CommonIEInfo,
    contents          QoSContents
 }

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

FwdQoSClass ::= BIT STRING (SIZE (8))          -- Table 4-18/Q.2931
--  unspecFwdQoSClass     FwdQoSClass ::= '00000000'B
--  futureFwdReserve     FwdQoSClass ::= '11111111'B

-- all other values are reserved

BkwdQoSClass ::= BIT STRING (SIZE (8)) -- Table 4-18/Q.2931
--  unspecBkwdQoSClass     BkwdQoSClass ::= '00000000'B
--  futureBkwdReserve     BkwdQoSClass ::= '11111111'B

-- all other values are reserved

ATMTrfcDscrptrIE ::= CHOICE -- EN 301 217-1, 14.3.7.1
  {  send SndATMDscrptrIE,
     receive RcvATMDscrptrIE
  }

SndATMDscrptrIE ::=  SEQUENCE
```

```
  {  commonIEInfo     CommonIEInfo,
     contents     ATM_Contents
  }

RcvATMDscrptrIE ::= SEQUENCE
  {       commonIEInfo          CommonIEInfo,
     contents SEQUENCE
       {  rates               SET OF Rates
--          trafficMgmtOptions      TrafficMgmtOptions OPTIONAL
       }
  }

Rates ::= SEQUENCE
  {     identifier Rate_Id,
           contents   BIT STRING(SIZE(24))
  }

TrafficMgmtOptions ::= SEQUENCE
  {     trafficManagementOptionsId      BIT STRING (SIZE (8)), -- ('10111111'B),
     spare_345678                    BIT STRING(SIZE(6)), -- Spare bits, normally set to
 '000000'B
     tb                              BIT STRING(SIZE(1)),
     tf                              BIT STRING(SIZE(1))
  }

ATM_Contents ::= OCTET STRING(SIZE(0..50))  -- EN 300 443-1 subclause 4.5.6
                                     -- ATM Traffic Descriptor contents

BrdbndBcapIE ::= SEQUENCE     -- EN 300 443-1 subclause 4.5.7
  {     commonIEInfo          CommonIEInfo,
     contents           BBC_contents
  }

BBC_contents ::=      OCTET STRING(SIZE(3..4))  -- EN 300 443-1 subclause 4.5.7
     -- Broadband Bearer Capability contents


OAMTrfcDscrptrIE ::= SEQUENCE     -- EN 300 443-1 subclause 4.5.24
  {       commonIEInfo          CommonIEInfo,
     contents          OAM_contents
  }

OAM_contents ::= OCTET STRING (SIZE (2))
--     extension_o5            BIT STRING('1'B),     Extension bit, set to '1'B
--     shaping_indicator       BIT STRING(SIZE(2)),  Type of subaddress
--     complete_indicator      BIT STRING(SIZE(1)),
--     spare_4_o5              BIT STRING(SIZE(1)),  Spare bit, normally set to '0'B
--     fault_mngt_indicator      BIT STRING(SIZE(4)),  (1)
--     extension_o6            BIT STRING('1'B),     Extension bit, set to '1'B
--     fwd_flow_indicator      BIT STRING(SIZE(3)),  (2)
--     spare_4_o6              BIT STRING(SIZE(1)),  Spare bit, normally set to '0'B
--     bwd_flow_indicator      BIT STRING(SIZE(4))   (3)
--  }     (1) User-network fault management indicator
     -- (2) Forward end-to-end OAM F5 flow indicator
     -- (3) Backward end-to-end OAM F5 flow indicator

ABRSetupParamsIE ::= SEQUENCE        -- Q.2961.3, 8.2.2
  {       commonIEInfo          CommonIEInfo,
     contents          ABR_contents
  }

ABR_contents ::= OCTET STRING (SIZE (2..28)) -- Q.2961.3, 8.2.2

EndEndTrnstDlayIE ::= SEQUENCE -- EN 300 443-1 subclause 4.5.17
  {     commonIEInfo          CommonIEInfo,
```

```
     contents          EETD_contents
  }

EETD_contents ::= OCTET STRING (SIZE (3..6))
--     cumulative_td_id      BIT STRING('00000001'B), : Cumulative transit delay identif
ier
--     cumulative_td         BIT STRING(SIZE(16))    : Cumulative transit delay value
--     maximum_td            Maximum_td OPTIONAL      : Maximum transit delay
--  }

Maximum_td ::= SEQUENCE
  {     maximum_td_id      BIT STRING (SIZE (8)), -- ('00000011'B), Maximum transit dela
y id
     maximum_td         BIT STRING(SIZE(16))     -- Maximum transit delay value
  }

CDVTDscrptrIE ::=  SEQUENCE  -- Q.2961.5, 8.2
  {       commonIEInfo          CommonIEInfo,
     contents          CDVT_contents
  }

CDVT_contents ::= OCTET STRING (SIZE (3..18)) -- Q.2961.5, 8.2

AltATMTrfcDscrptrIE ::= CHOICE  -- EN 301 443-1 subclause 4.5.6, EN 301 067-1 su
bclause 8.2.1, EN 301 068
  {      send           SndATMDscrptrIE,
     receive      RcvATMDscrptrIE
  }

MinATMTrfcDscrptorIE ::= CHOICE  -- EN 301 443-1 subclause 4.5.6, EN 301 067-1 s
ubclause 8.2.1, EN 301 068
  {      send SndATMDscrptrIE,
     receive RcvMinATMDscrptrIE
  }

RcvMinATMDscrptrIE ::= SEQUENCE
  {     commonIEInfo     CommonIEInfo,
     contents      SET OF Rates
  }

CommonIEInfo ::= SEQUENCE  --EN 301 005-4, Annex E
 {  iEType               IEType,
    iECompatInd          IECompatInd,
    iELength             IELength
 }

IEType ::= OCTET STRING (SIZE (1)) -- Table 53, EN 301 217-1
--  iEConnRefNo           IEType ::= '00000000'B
--  iEConnRefNoList     IEType ::= '00000001'B
iEUsrPortConnId           IEType ::= '02'H
--  iESrvcPortConnId     IEType ::= '00000011'B
--  iEAltUsrPortVPCI     IEType ::= '00000100'B
--  iEAltSrvcPortVPCI     IEType ::= '00000101'B
--  iEAutoCongLvl          IEType ::= '00000110'B
--  iERejCause          IEType ::= '00000111'B
--  iEBranchId          IEType ::= '00001000'B
 iEBranchIdList     IEType ::= '00001001'B
--  iEATMTrfcDscrptr     IEType ::= '00001010'B
--  iEBrdbndBcap          IEType ::= '00001011'B
--  iEOAMTrfcDscrptr     IEType ::= '00001100'B
--  iEQoSParams          IEType ::= '00001101'B
--  iEABRSetupParams     IEType ::= '00001110'B
--  iEEndEndTrnstDlay     IEType ::= '00001111'B
--  iECDVTDscrptr          IEType ::= '00010000'B
--  iEAltATMTrfcDscrptr     IEType ::= '00010001'B
```

```
--  iEMinATMTrfcDscrptor     IEType ::= '00010010'B
--  iEProtErrCause     IEType ::= '00010011'B
  -- All other values are reserved

IECompatInd ::= OCTET STRING (SIZE (1))
--SEQUENCE   EN 301 005-4, Annex E
-- {  extBitLast          [1] ExtBitLast, DEFAULT '1'B,
--    spareBits2          [2] SpareBits2,  DEFAULT '00'B,
--    compatFlag          [3] CompatFlag,
--    spareBits1          [4] SpareBits1,  DEFAULT '0'B,
--    iEActInd            [5] IEActInd
-- }

IELength ::= OCTET STRING (SIZE (2))  -- EN 301 005-4, Annex E

ExtBitLast ::= BIT STRING (SIZE (1)) --Always defaulted to '1'B (last octet in g
roup), EN 301 005-4, Annex E

CompatFlag ::= BIT STRING (SIZE(1)) --Table 35, EN 301 005-1
--  compatFlagNotSignif     CompatFlag ::= '0'B
--  compatFlagExplicit      CompatFlag ::= '1'B

IEActInd ::= BIT STRING (SIZE (3)) --Table 36, EN 301 005-1
--  iEActIndReject                  IEActInd ::= '000'B
--  iEActIndDiscardAndProceed       IEActInd ::= '001'B
--  iEActIndDiscardProceedAndReport IEActInd ::= '010'B
--  iEActIndDiscardMsgAndIgnore     IEActInd ::= '101'B
--  iEActIndDiscardMsgAndReport     IEActInd ::= '110'B
  --All other values are reserved

 -- definition of BBCC messages

CommonMsgInfo ::= SEQUENCE     --EN 301 005-4, Annex E
  {     protDiscr          ProtDiscr, --DEFAULT '49'H,
        transId               TransId,
        msgType               MsgType,
        msgCompatInd          MsgCompatInd,
        msgLength          MsgLength
  }

Alloc ::= SEQUENCE          -- En 301 217-1, Table 23
  {     commonMsgInfo          CommonMsgInfo,
     connRefNoIE          ConnRefNoIE,
     aTMTrfcDscrptrIE     ATMTrfcDscrptrIE,
     brdbndBcapIE          BrdbndBcapIE,
     qoSParamsIE          QoSParamsIE,
     usrPortConnIdIE          UsrPortConnIdIE,
     srvcPortConnIdIE     SrvcPortConnIdIE,
     altUsrPortVPCIIE     [1]  AltUsrPortVPCIIE     OPTIONAL,
     altSrvcPortVPCIIE     [2]  AltSrvcPortVPCIIE     OPTIONAL,
     branchIdIE          [3]  BranchIdIE          OPTIONAL,
     aBRSetupParamsIE     [4]  ABRSetupParamsIE     OPTIONAL,
     endEndTrnstDlayIE     [5]  EndEndTrnstDlayIE     OPTIONAL,
     cDVTDscrptrIE          [6]  CDVTDscrptrIE          OPTIONAL,
     oAMTrfcDscrptrIE     [7]  OAMTrfcDscrptrIE     OPTIONAL,
     altATMTrfcDscrptrIE     [8]  AltATMTrfcDscrptrIE     OPTIONAL,
     minATMTrfcDscrptorIE     [9]  MinATMTrfcDscrptorIE     OPTIONAL
  }



AllocAcc ::= SEQUENCE          -- En 301 217-1, Table 24
  {     commonMsgInfo          CommonMsgInfo,
     aTMTrfcDscrptrIE     ATMTrfcDscrptrIE     OPTIONAL,
```

```
--      usrPortConnIdIE          UsrPortConnIdIE          OPTIONAL,
--      srvcPortConnIdIE     SrvcPortConnIdIE     OPTIONAL,
     branchIdIE          BranchIdIE          OPTIONAL,
--      aBRSetupParamsIE     ABRSetupParamsIE     OPTIONAL,
--      endEndTrnstDlayIE     EndEndTrnstDlayIE     OPTIONAL,
--      cDVTDscrptrIE          CDVTDscrptrIE          OPTIONAL,
--      oAMTrfcDscrptrIE     OAMTrfcDscrptrIE     OPTIONAL,
     altATMTrfcDscrptrIE     AltATMTrfcDscrptrIE     OPTIONAL,
     minATMTrfcDscrptorIE     MinATMTrfcDscrptorIE     OPTIONAL--,
--      autoCongLvlIE          AutoCongLvlIE          OPTIONAL
  }

AllocRej ::= SEQUENCE          -- En 301 217-1, Table 25
  {     commonMsgInfo          CommonMsgInfo,
     rejCauseIE          RejCauseIE
--      autoCongLvlIE          AutoCongLvlIE     OPTIONAL
  }

AllocComp ::= SEQUENCE          -- En 301 217-1, Table 26
  {     commonMsgInfo          CommonMsgInfo,
     connRefNoIE          ConnRefNoIE,
     aTMTrfcDscrptrIE     [1]  ATMTrfcDscrptrIE     OPTIONAL,
     usrPortConnIdIE          [2]  UsrPortConnIdIE          OPTIONAL,
     aBRSetupParamsIE     [3]  ABRSetupParamsIE     OPTIONAL,
     cDVTDscrptrIE          [4]  CDVTDscrptrIE          OPTIONAL
  }

AllocCompAcc ::= SEQUENCE     -- EN 301 217-1, 14.3.2.5
  {     commonMsgInfo        CommonMsgInfo
  }

AllocCompRej ::= SEQUENCE      -- En 301 217-1, Table 27
  {     commonMsgInfo          CommonMsgInfo,
     rejCauseIE          RejCauseIE
  }

Dealloc ::= SEQUENCE          -- En 301 217-1, Table 28
  {     commonMsgInfo          CommonMsgInfo,
     connRefNoListIE          ConnRefNoListIE
  }

DeallocAcc ::= SEQUENCE          -- En 301 217-1, Table 29
  {     commonMsgInfo          CommonMsgInfo,
     autoCongLvlIE          AutoCongLvlIE     OPTIONAL
  }

BbccReset ::= SEQUENCE          -- En 301 217-1, Table 30
  {     commonMsgInfo          CommonMsgInfo,
     usrPortConnIdIE          [1] UsrPortConnIdIE          OPTIONAL,
     srvcPortConnIdIE     [2] SrvcPortConnIdIE     OPTIONAL
  }

BbccResetAcc ::= SEQUENCE     -- EN 301 217-1, 14.3.3.2
  {     commonMsgInfo           CommonMsgInfo
  }

BbccResetRej ::= SEQUENCE      -- En 301 217-1, Table 31
  {     commonMsgInfo          CommonMsgInfo,
     rejCauseIE          RejCauseIE
  }

BbccPresync ::= SEQUENCE     -- En 301 217-1, Table 32
  {     commonMsgInfo          CommonMsgInfo,
     usrPortConnIdIE          [1] UsrPortConnIdIE     OPTIONAL,
     srvcPortConnIdIE     [2] SrvcPortConnIdIE     OPTIONAL
```

```
  }

BbccPresyncAcc ::= SEQUENCE     -- EN 301 217-1, 14.3.3.5
  {     commonMsgInfo           CommonMsgInfo
  }

BbccPresyncRej ::= SEQUENCE      -- En 301 217-1, Table 33
  {     commonMsgInfo          CommonMsgInfo,
     rejCauseIE          RejCauseIE
  }

AnFault ::= SEQUENCE          -- En 301 217-1, Table 34
  {     commonMsgInfo          CommonMsgInfo,
     connRefNoIE          ConnRefNoIE          OPTIONAL,
     branchIdIE          BranchIdIE          OPTIONAL,
     usrPortConnIdIE          UsrPortConnIdIE          OPTIONAL,
     srvcPortConnIdIE     SrvcPortConnIdIE     OPTIONAL
  }

AnFaultAcc ::= SEQUENCE          -- EN 301 217-1, 14.3.3.8
  {     commonMsgInfo           CommonMsgInfo
  }

ProtocolError ::= SEQUENCE     -- En 301 217-1, Table 35
  {     commonMsgInfo          CommonMsgInfo,
     protErrCauseIE          ProtErrCauseIE
  }

Modify ::= SEQUENCE          -- En 301 217-1, Table 36
  {     commonMsgInfo          CommonMsgInfo,
     connRefNoIE          ConnRefNoIE,
     aTMTrfcDscrptrIE     ATMTrfcDscrptrIE,
     altATMTrfcDscrptrIE     [1]  AltATMTrfcDscrptrIE     OPTIONAL
--     minATMTrfcDscrptorIE     MinATMTrfcDscrptorIE     OPTIONAL
  }

ModifyAcc ::= SEQUENCE           -- En 301 217-1, Table 37
  {     commonMsgInfo          CommonMsgInfo,
     aTMTrfcDscrptrIE     ATMTrfcDscrptrIE     OPTIONAL,
     altATMTrfcDscrptrIE     AltATMTrfcDscrptrIE     OPTIONAL,
     minATMTrfcDscrptorIE     MinATMTrfcDscrptorIE     OPTIONAL --,
--     autoCongLvlIE          AutoCongLvlIE          OPTIONAL
  }

ModifyRej ::= SEQUENCE           -- En 301 217-1, Table 38
  {     commonMsgInfo          CommonMsgInfo,
     rejCauseIE          RejCauseIE --,
--     autoCongLvlIE          AutoCongLvlIE          OPTIONAL
  }

ModifyComp ::= SEQUENCE          -- En 301 217-1, Table 39
  {     commonMsgInfo          CommonMsgInfo,
        connRefNoIE             ConnRefNoIE,
     aTMTrfcDscrptrIE     [1]  ATMTrfcDscrptrIE     OPTIONAL
  }

ModifyCompAcc ::= SEQUENCE     -- EN 301 217-1, 14.3.4.5
  {     commonMsgInfo           CommonMsgInfo
  }

ModifyCompRej ::= SEQUENCE      -- En 301 217-1, Table 40
  {     commonMsgInfo          CommonMsgInfo,
     rejCauseIE          RejCauseIE
  }
```

```
ModifyAbort ::= SEQUENCE     -- En 301 217-1, Table 41
  {     commonMsgInfo          CommonMsgInfo,
     connRefNoIE          ConnRefNoIE
  }

ModifyAbortAcc ::= SEQUENCE     -- EN 301 217-1, 14.3.4.8
  {     commonMsgInfo           CommonMsgInfo
  }

ModifyAbortRej ::= SEQUENCE      -- En 301 217-1, Table 41
  {     commonMsgInfo          CommonMsgInfo,
     rejCauseIE          RejCauseIE
  }

AddBranch ::= SEQUENCE          -- En 301 217-1, Table 43
  {     commonMsgInfo          CommonMsgInfo,
     connRefNoIE          ConnRefNoIE,
     branchIdIE          BranchIdIE,
     usrPortConnIdIE          UsrPortConnIdIE
--     altUsrPortVPCIIE     AltUsrPortVPCIIE          OPTIONAL
  }

AddBranchAcc  ::= SEQUENCE      -- En 301 217-1, Table 44
  {     commonMsgInfo          CommonMsgInfo
--     usrPortConnIdIE          UsrPortConnIdIE     OPTIONAL,
--     autoCongLvlIE          AutoCongLvlIE     OPTIONAL
  }

AddBranchRej  ::= SEQUENCE      -- En 301 217-1, Table 45
  {     commonMsgInfo          CommonMsgInfo,
     rejCauseIE          RejCauseIE
--     autoCongLvlIE          AutoCongLvlIE     OPTIONAL
  }

UpdateBranch ::= SEQUENCE      -- En 301 217-1, Table 46
  {     commonMsgInfo          CommonMsgInfo,
     connRefNoIE          ConnRefNoIE,
     branchIdIE          BranchIdIE,
     usrPortConnIdIE          UsrPortConnIdIE
  }

UpdateBranchAcc ::= SEQUENCE     -- EN 301 217-1, 14.3.5.5
  {     commonMsgInfo           CommonMsgInfo
  }

UpdateBranchRej ::= SEQUENCE      -- En 301 217-1, Table 47
  {     commonMsgInfo          CommonMsgInfo,
     rejCauseIE          RejCauseIE
  }

DropBranch ::= SEQUENCE          -- En 301 217-1, Table 48
  {     commonMsgInfo          CommonMsgInfo,
     connRefNoIE          ConnRefNoIE,
     branchIdListIE          BranchIdListIE
  }

DropBranchAcc ::= SEQUENCE      -- En 301 217-1, Table 49
  {     commonMsgInfo          CommonMsgInfo
--     autoCongLvlIE          AutoCongLvlIE     OPTIONAL
  }

DropBranchRej ::= SEQUENCE      -- En 301 217-1, Table 50
  {     commonMsgInfo          CommonMsgInfo,
     rejCauseIE          RejCauseIE
--     autoCongLvlIE          AutoCongLvlIE     OPTIONAL
```

```
  }

UnknownPDU ::= SEQUENCE
  {     commonMsgInfo          CommonMsgInfo,
     usrPortConnIdIE          UsrPortConnIdIE      OPTIONAL,
     srvcPortConnIdIE     SrvcPortConnIdIE     OPTIONAL
  }

TruncInfoBbccReset ::= SEQUENCE
  {     truncCommonMsgInfo     TruncCommonMsgInfo,
     usrPortConnIdIE          UsrPortConnIdIE          OPTIONAL,
     srvcPortConnIdIE     SrvcPortConnIdIE     OPTIONAL
  }

TruncInfoAnFault ::= SEQUENCE
  {     truncCommonMsgInfo     TruncCommonMsgInfo,
     connRefNoIE          ConnRefNoIE          OPTIONAL,
     branchIdIE          BranchIdIE          OPTIONAL,
     usrPortConnIdIE          UsrPortConnIdIE          OPTIONAL,
     srvcPortConnIdIE     SrvcPortConnIdIE     OPTIONAL
  }

TruncCommonMsgInfo ::= SEQUENCE
  {     protDiscr           ProtDiscr, --DEFAULT '49'H,
        transId               TransId,
        msgType               MsgType,
        msgCompatInd          MsgCompatInd,
        truncMsgLength          OCTET STRING (SIZE (1))
  }

BbccResetMissingMndtryIE ::= SEQUENCE
  {     commonMsgInfo          CommonMsgInfo
  }

BbccResetXtraUnkIE ::= SEQUENCE
  {     commonMsgInfo          CommonMsgInfo,
     srvcPortConnIdIE     SrvcPortConnIdIE,
     unknownIE          UnknownIE
  }


BbccResetXtraUnexpIE ::= SEQUENCE
  {     commonMsgInfo          CommonMsgInfo,
     connRefNoIE          ConnRefNoIE,
     srvcPortConnIdIE     SrvcPortConnIdIE
  }

AnFaultMissingMndtryIE ::= SEQUENCE
  {     commonMsgInfo          CommonMsgInfo
  }

AnFaultXtraUnkIE ::= SEQUENCE
  {     commonMsgInfo          CommonMsgInfo,
     usrPortConnIdIE          UsrPortConnIdIE,
     unknownIE          UnknownIE
  }

AnFaultXtraUnexpIE ::= SEQUENCE
  {     commonMsgInfo          CommonMsgInfo,
     usrPortConnIdIE          UsrPortConnIdIE,
     rejCauseIE          RejCauseIE
  }

UnknownIE ::= SEQUENCE
  {       commonIEInfo          CommonIEInfo,
```

```
    contents          OCTET STRING (SIZE (1))
  }


Rate_Id ::=  -- EN 301 443-1 subclause 4.5.6, EN 301 067-1 subclause 8.2.1, EN 3
01 068
BIT STRING ( '10000010'B | -- forward peak cell rate (CLP = 0)
             '10000011'B | -- backward peak cell rate (CLP = 0)
             '10000100'B | -- forward peak cell rate (CLP = 0 + 1)
             '10000101'B | -- backward peak cell rate (CLP = 0 + 1)
             '10001000'B | -- forward sustainable cell rate (CLP = 0)
             '10001001'B | -- backward sustainable cell rate (CLP = 0)
             '10010000'B | -- forward sustainable cell rate (CLP = 0 + 1)
             '10010001'B | -- backward sustainable cell rate (CLP = 0 + 1)
             '10100000'B | -- forward maximum burst size (CLP = 0)
             '10100001'B | -- backward maximum burst size (CLP = 0)
             '10110000'B | -- forward maximum burst size (CLP = 0 + 1)
             '10110001'B | -- backward maximum burst size (CLP = 0 + 1)
             '10010010'B | -- forward ABR minimum cell rate (CLP = 0 + 1)
             '10010011'B | -- backward ABR minmum cell rate (CLP = 0 + 1)
             '11000000'B | -- forward RM peak cell rate
             '11000001'B   -- backward RM peak cell rate
           )     -- Possible identifiers for the traffic rates of the ATM traffic de
scriptor information

BbccMsg ::= CHOICE      -- EN 201 217-1, Table 21
  {     alloc          [1]     Alloc,
    allocAcc     [2]     AllocAcc,
    allocRej     [3]     AllocRej,
    allocComp     [4]     AllocComp,
    allocCompAcc     [5]     AllocCompAcc,
    allocCompRej     [6]     AllocCompRej,
    dealloc          [7]     Dealloc,
    deallocAcc     [8]     DeallocAcc,
    bbccReset     [9]     BbccReset,
    bbccResetAcc     [10]     BbccResetAcc,
    bbccResetRej     [11]     BbccResetRej,
    bbccPresync     [12]     BbccPresync,
    bbccPresyncAcc     [13]     BbccPresyncAcc,
    bbccPresyncRej     [14]     BbccPresyncRej,
    anFault          [15]     AnFault,
    anFaultAcc     [16]     AnFaultAcc,
    protErr          [17]     ProtocolError,
    modify          [18]     Modify,
    modifyAcc     [19]     ModifyAcc,
    modifyRej     [20]     ModifyRej,
    modifyComp     [21]     ModifyComp,
    modifyCompAcc     [22]     ModifyCompAcc,
    modifyCompRej     [23]     ModifyCompRej,
    modifyAbort     [24]     ModifyAbort,
    modifyAbortAcc     [25]     ModifyAbortAcc,
    modifyAbortRej     [26]     ModifyAbortRej,
    addBranch     [27]     AddBranch,
    addBranchAcc     [28]     AddBranchAcc,
    addBranchRej     [29]     AddBranchRej,
    updateBranch     [30]     UpdateBranch,
    updateBranchAcc     [31]     UpdateBranchAcc,
    updateBranchRej     [32]     UpdateBranchRej,
    dropBranch     [33]     DropBranch,
    dropBranchAcc     [34]     DropBranchAcc,
    dropBranchRej     [35]     DropBranchRej
  }


SN_ConnEstPars  ::= SEQUENCE {
```

```
  connRefNo     INTEGER,
  branchId     INTEGER
  }


dummyCMI CommonMsgInfo ::= {
  protDiscr ’49’H,
  transId {
    transIdLength ’03’H,
    transIdFlag ’0’B,
    transIdVal 0
    },
  msgType ’00’H,
  msgCompatInd ’80’H, -- {
 --   extBitLast ’1’B,
 --   spareBits2 ’00’B,
 --   compatFlag ’0’B,
 --   spareBits21 ’00’B,
 --   msgActInd ’01’B
 --   },
  msgLength ’0000’H
  }

dummyUPCIIE UsrPortConnIdIE ::= {
  commonIEInfo {
    iEType ’02’H,
    iECompatInd ’80’H, --{
--      extBitLast ’1’B,
--      spareBits2 ’00’B,
 --     compatFlag ’0’B,
--      spareBits1 ’0’B,
--      iEActInd ’001’B
--      },
    iELength ’0000’H
    },
  usrPortConnIdIEOctet5 ’A0’H, -- {
--    extBitLast ’1’B,
--    usrRscInd ’000’B,
--    dirFlg ’00’B,
--    connIdFlg ’00’B
--    },
  lgclUsrPortId ’000100’H
  }

dummy2UPCIIE UsrPortConnIdIE ::= {
  commonIEInfo {
    iEType ’02’H,
    iECompatInd ’80’H, --{
--      extBitLast ’1’B,
--      spareBits2 ’00’B,
 --     compatFlag ’0’B,
--      spareBits1 ’0’B,
--      iEActInd ’001’B
--      },
    iELength ’0000’H
    },
  usrPortConnIdIEOctet5 ’A0’H, -- {
--    extBitLast ’1’B,
--    usrRscInd ’000’B,
--    dirFlg ’00’B,
--    connIdFlg ’00’B
--    },
  lgclUsrPortId ’A00101’H
  }
```

```
dummyCRNIE ConnRefNoIE ::= {
commonIEInfo {
    iEType '00'H,
    iECompatInd '80'H, -- {
--      extBitLast '1'B,
--      spareBits2 '00'B,
--      compatFlag '0'B,
--      spareBits1 '0'B,
--      iEActInd '001'B
--      },
    iELength '0000'H
    },
  connRefNoVal 100
  }

dummyATMTDIE ATMTrfcDscrptrIE ::= send : -- '0A110002FFFF'H
  { commonIEInfo {
    iEType '0A'H,
    iECompatInd '80'H, -- {
--        extBitLast '1'B,
--        spareBits2 '00'B,
--        compatFlag '0'B,
--        spareBits1 '0'B,
--        iEActInd '001'B
--        },
      iELength '0000'H
      },
    contents '7100'H
    }

dummy2ATMTDIE ATMTrfcDscrptrIE ::= send : -- '0A110002FFFF'H
  { commonIEInfo {
    iEType '0A'H,
    iECompatInd '80'H, -- {
--        extBitLast '1'B,
--        spareBits2 '00'B,
--        compatFlag '0'B,
--        spareBits1 '0'B,
--        iEActInd '001'B
--        },
      iELength '0000'H
      },
    contents '7201'H
    }

dummy3ATMTDIE ATMTrfcDscrptrIE ::= send : -- '0A110002FFFF'H
  { commonIEInfo {
    iEType '0A'H,
    iECompatInd '80'H, -- {
--        extBitLast '1'B,
--        spareBits2 '00'B,
--        compatFlag '0'B,
--        spareBits1 '0'B,
--        iEActInd '001'B
--        },
      iELength '0000'H
      },
    contents '7100'H
    }

dummyBBIE BrdbndBcapIE ::= -- '0B110003FFFFFF'H
  {
  commonIEInfo {
    iEType '0B'H,
```

```
    iECompatInd '80'H, --{
--     extBitLast '1'B,
--      spareBits2 '00'B,
--      compatFlag '0'B,
--      spareBits1 '0'B,
--      iEActInd '001'B
--      },
    iELength '0000'H
    },
  contents '751000'H
  }

dummyQSPIE QoSParamsIE ::= --'0D11000400000000'H
{
  commonIEInfo {
    iEType '0D'H,
    iECompatInd '80'H, --{
--      extBitLast '1'B,
--      spareBits2 '00'B,
--      compatFlag '0'B,
--      spareBits1 '0'B,
--      iEActInd '001'B
--      },
    iELength '0000'H
    },
   contents '7700'H
  }

dummySPCIIE SrvcPortConnIdIE ::= {
  commonIEInfo {
    iEType '03'H,
    iECompatInd '80'H, -- {
--      extBitLast '1'B,
--      spareBits2 '00'B,
--      compatFlag '0'B,
--      spareBits1 '0'B,
--      iEActInd '001'B
--      },
    iELength '0000'H
    },
  srvcPortConnIdIEOctet5 '80'H --, {
--    extBitLast '1'B,
--    srvcRscInd '000'B,
--    spare '00'B,
--    connIdFlg '00'B
--    },
  }


dummyAATMTDIE AltATMTrfcDscrptrIE ::= -- '11110002FFFF'H
  send : {
    commonIEInfo {
     iEType '11'H,
      iECompatInd '80'H, --{
--        extBitLast '1'B,
--        spareBits2 '00'B,
--       compatFlag '0'B,
--        spareBits1 '0'B,
--        iEActInd '001'B
--        },
     iELength '0000'H
      },
    contents '7200'H
    }
```

```
dummy2AATMTDIE AltATMTrfcDscrptrIE ::= -- '11110002FFFF'H
  send : {
    commonIEInfo {
     iEType '11'H,
      iECompatInd '80'H, --{
--        extBitLast '1'B,
--        spareBits2 '00'B,
--       compatFlag '0'B,
--        spareBits1 '0'B,
--        iEActInd '001'B
--        },
     iELength '0000'H
      },
    contents '7200'H
    }

dummyAUPVPCIIE AltUsrPortVPCIIE ::= {
  commonIEInfo {
    iEType '04'H,
    iECompatInd '80'H, -- {
--      extBitLast '1'B,
--      spareBits2 '00'B,
--      compatFlag '0'B,
--      spareBits1 '0'B,
--      iEActInd '001'B
--      },
    iELength '0000'H
    },
  vpcis 'A00C'H
  }

dummyASPVPCIIE AltSrvcPortVPCIIE ::= {
  commonIEInfo {
    iEType '05'H,
    iECompatInd '80'H, -- {
--      extBitLast '1'B,
--      spareBits2 '00'B,
--      compatFlag '0'B,
--      spareBits1 '0'B,
--      iEActInd '001'B
--      },
    iELength '0000'H
    },
  vpcis 'A08C'H
  }

dummyRCIE RejCauseIE ::= {
  commonIEInfo {
    iEType '00000111'B,
    iECompatInd '80'H, -- {
--      extBitLast '1'B,
--      spareBits2 '00'B,
--      compatFlag '0'B,
--      spareBits1 '0'B,
--      iEActInd '001'B
--      },
    iELength '0000'H
    },
  rejCauseOctet 0
  }

dummyBIIE BranchIdIE ::= {
  commonIEInfo {
    iEType '08'H,
    iECompatInd '80'H, -- {
```

```
--      extBitLast '1'B,
--      spareBits2 '00'B,
--      compatFlag '0'B,
--      spareBits1 '0'B,
--      iEActInd '001'B
--      },
    iELength '0000'H
    },
  branchIdVal 0
  }

dummyCRNLIE ConnRefNoListIE ::= {
  commonIEInfo {
    iEType '00000001'B,
    iECompatInd '80'H, --{
--      extBitLast '1'B,
--      spareBits2 '00'B,
--      compatFlag '0'B,
--      spareBits1 '0'B,
--      iEActInd '001'B
--      },
    iELength '0000'H
    },
  connRefNos 0
  }

dummyBILIE BranchIdListIE ::= {
  commonIEInfo {
    iEType '00001001'B,
    iECompatInd '80'H, -- {
--      extBitLast '1'B,
--      spareBits2 '00'B,
--      compatFlag '0'B,
--      spareBits1 '0'B,
--      iEActInd '001'B
--     },
    iELength '0000'H
    },
  branchIds 0
  }

dummyABRSPIE ABRSetupParamsIE ::= {
  commonIEInfo {
    iEType '0E'H,
    iECompatInd '80'H, -- {
--      extBitLast '1'B,
--      spareBits2 '00'B,
--      compatFlag '0'B,
--      spareBits1 '0'B,
--      iEActInd '001'B
--     },
    iELength '0000'H
    },
  contents  '7600'H
  }

dummy2ABRSPIE ABRSetupParamsIE ::= {
  commonIEInfo {
    iEType '0E'H,
    iECompatInd '80'H, -- {
--      extBitLast '1'B,
--      spareBits2 '00'B,
--      compatFlag '0'B,
--      spareBits1 '0'B,
--      iEActInd '001'B
```

```
--      },
    iELength '0000'H
    },
  contents  '7601'H
  }


dummyCDVTDIE CDVTDscrptrIE ::= {
  commonIEInfo {
    iEType '10'H,
    iECompatInd '80'H, -- {
--      extBitLast '1'B,
--      spareBits2 '00'B,
--      compatFlag '0'B,
--      spareBits1 '0'B,
--      iEActInd '001'B
--     },
    iELength '0000'H
    },
  contents  '781000'H
  }

dummy2CDVTDIE CDVTDscrptrIE ::= {
  commonIEInfo {
    iEType '10'H,
    iECompatInd '80'H, -- {
--      extBitLast '1'B,
--      spareBits2 '00'B,
--      compatFlag '0'B,
--      spareBits1 '0'B,
--      iEActInd '001'B
--     },
    iELength '0000'H
    },
  contents  '781001'H
  }


dummyOAMTDIE OAMTrfcDscrptrIE  ::= {
  commonIEInfo {
    iEType '0C'H,
    iECompatInd '80'H, -- {
--      extBitLast '1'B,
--      spareBits2 '00'B,
--      compatFlag '0'B,
--      spareBits1 '0'B,
--      iEActInd '001'B
--     },
    iELength '0000'H
    },
  contents  '7400'H
  }



allocMsgType            MsgType ::= '40'H
allocAccMsgType       MsgType ::= '41'H
allocRejMsgType          MsgType ::= '42'H
allocCompMsgType     MsgType ::= '43'H
allocCompAccMsgType     MsgType ::= '44'H
allocCompRejMsgType     MsgType ::= '45'H
deallocMsgType           MsgType ::= '46'H
deallocAccMsgType     MsgType ::= '47'H
bbccResetMsgType      MsgType ::= '48'H
bbccResetAccMsgType     MsgType ::= '49'H
```

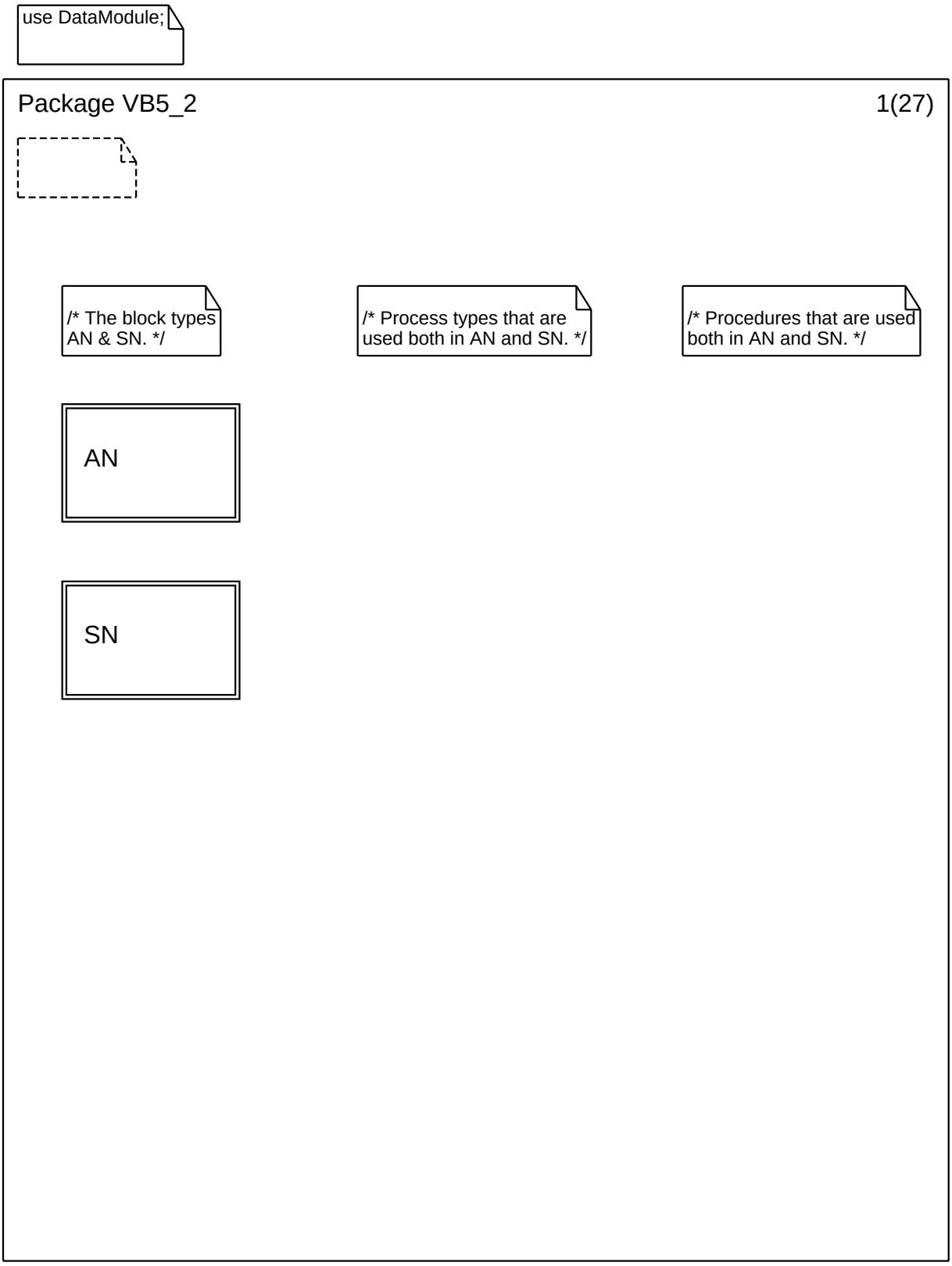
```
bbccResetRejMsgType     MsgType ::= ’4A’H
bbccPresyncMsgType     MsgType ::= ’4B’H
bbccPresyncAccMsgType     MsgType ::= ’4C’H
bbccPresyncRejMsgType     MsgType ::= ’4D’H
anFaultMsgType          MsgType ::= ’4E’H
anFaultAccMsgType     MsgType ::= ’4F’H
protocolErrorMsgType     MsgType ::= ’50’H
modifyMsgType          MsgType ::= ’51’H
modifyAccMsgType     MsgType ::= ’52’H
modifyRejMsgType     MsgType ::= ’53’H
modifyCompMsgType     MsgType ::= ’54’H
modifyCompAccMsgType     MsgType ::= ’55’H
modifyCompRejMsgType     MsgType ::= ’56’H
modifyAbortMsgType     MsgType ::= ’57’H
modifyAbortAccMsgType     MsgType ::= ’58’H
modifyAbortRejMsgType     MsgType ::= ’59’H
addBranchMsgType      MsgType ::= ’5A’H
addBranchAccMsgType     MsgType ::= ’5B’H
addBranchRejMsgType     MsgType ::= ’5C’H
updateBranchMsgType     MsgType ::= ’5D’H
updateBranchAccMsgType     MsgType ::= ’5E’H
updateBranchRejMsgType     MsgType ::= ’5F’H
dropBranchMsgType     MsgType ::= ’60’H
dropBranchAccMsgType     MsgType ::= ’61’H
dropBranchRejMsgType     MsgType ::= ’62’H

flag1 TransIdFlag ::= ’1’B

hex80           OCTET STRING (SIZE (1)) ::= ’80’H
hex0006          OCTET STRING (SIZE (2)) ::= ’0006’H
hexA0          OCTET STRING (SIZE (1)) ::= ’A0’H
hexA00101     OCTET STRING (SIZE (3)) ::= ’A00101’H
hexA001          OCTET STRING (SIZE (2)) ::= ’A001’H
hexA002          OCTET STRING (SIZE (2)) ::= ’A002’H
hexA003          OCTET STRING (SIZE (2)) ::= ’A003’H
hexA004          OCTET STRING (SIZE (2)) ::= ’A004’H
hexA005          OCTET STRING (SIZE (2)) ::= ’A005’H
hexA006          OCTET STRING (SIZE (2)) ::= ’A006’H
hexA007          OCTET STRING (SIZE (2)) ::= ’A007’H
hexA008          OCTET STRING (SIZE (2)) ::= ’A008’H
hexA009          OCTET STRING (SIZE (2)) ::= ’A009’H
hexA00A          OCTET STRING (SIZE (2)) ::= ’A00A’H
hexA00B          OCTET STRING (SIZE (2)) ::= ’A00B’H
hexA101          OCTET STRING (SIZE (2))     ::= ’A101’H
hexA102          OCTET STRING (SIZE (2))     ::= ’A102’H
hexA103          OCTET STRING (SIZE (2))     ::= ’A103’H
hexA104          OCTET STRING (SIZE (2))     ::= ’A104’H
hexA105          OCTET STRING (SIZE (2))     ::= ’A105’H
hexA106          OCTET STRING (SIZE (2))     ::= ’A106’H
hexA107          OCTET STRING (SIZE (2))     ::= ’A107’H
hexA108          OCTET STRING (SIZE (2))     ::= ’A108’H
hexA109          OCTET STRING (SIZE (2))     ::= ’A109’H
hexA10A          OCTET STRING (SIZE (2))     ::= ’A10A’H
hexA10B          OCTET STRING (SIZE (2))     ::= ’A10B’H
hexA081          OCTET STRING (SIZE (2))     ::= ’A081’H
hexA082          OCTET STRING (SIZE (2))     ::= ’A082’H
hexA083          OCTET STRING (SIZE (2))     ::= ’A083’H
hexA084          OCTET STRING (SIZE (2))     ::= ’A084’H
hexA085          OCTET STRING (SIZE (2))     ::= ’A085’H
hexA086          OCTET STRING (SIZE (2))     ::= ’A086’H
hexA201          OCTET STRING (SIZE (2))     ::= ’A201’H
hexA202          OCTET STRING (SIZE (2))     ::= ’A202’H
hexA203          OCTET STRING (SIZE (2))     ::= ’A203’H
hexA204          OCTET STRING (SIZE (2))     ::= ’A204’H
hexA205          OCTET STRING (SIZE (2))     ::= ’A205’H
```

```
hexA206          OCTET STRING (SIZE (2))     ::= ’A206’H
lup1          OCTET STRING (SIZE (3)) ::= ’A00101’H
lup2          OCTET STRING (SIZE (3)) ::= ’A00102’H
lup3          OCTET STRING (SIZE (3)) ::= ’A00103’H

END
```


use DataModule;
Package VB5_2
1(27)
/* The block types AN & SN. */
/* Process types that are used both in AN and SN. */
/* Procedures that are used both in AN and SN. */
AN
SN

use DataModule;

Package VB5_2 2(27)

/*Procedures, data and comments for enabling/facilitating tool supported simulation*/

/* For handling of protocol syntax errors, two types of errors are distinguished (see subclauses 13.6.1.5 and 14.1 on general error handling procedures)
type 1: errors in the common message part, e.g. protocol discriminator, transaction id, message type
type 2: errors within message parameters (IE errors)
The checking for and handling of errrors is shown in the SDLs in a different way.
For type 1 errors it is assumed that the common message part of RTMC messages is checked prior to processing in the SDL processes. These errors are represented in the SDLs by pseudo events. The SDLs show the subsequent error handling (MSG_ERROR_HANDLING).
For type 2 errors the SDLs show the possible results of IE syntax checking (procedure IE_CHECK) only and the subseq error handling (procedure IE_ERROR_HANDLING)
*/

IE_CHECK

Procedure IE_CHECK is used to check the syntax of the IEs of a B−BCC−message. The common octects of each message (first 9 octets) are not considered for error handling. The syntax check is described in an informal manner and shall be used for simulation purposes only. Subclause 13.3.1.5 'General error handling' contains the normative part of the error handling procedures which shall be applied for real implementations.

ERR_CONTAINED

Procedure ERR_CONTAINED is used to control the error handling during simulation runs. It allows switching on and off of the error handling via an external synonym C_ErrContained. If C_ErrContained is set to FALSE, no error handling is activated. If C_ErrConatined is set to TRUE, then you have to indicate the type of error during simulation.

IE_ERROR_
_HANDLING

Procedure IE_ERROR_HANDLING is used for handling of an IE syntax error

MSG_ERROR_
_HANDLING

Procedure MSG_ERROR_HANDLING is used for handling of syntax errors in the common octets of each message (first 9 octets, see table 22)

Package VB5_2

```
/* For handling of protocol syntax errors, two types of errors are distinguished
 (see subclauses 13.6.1.5 and 14.1 on
  general error handling procedures)
        type 1: errors in the common message part, e.g. protocol discriminator,
 transaction id, message type
        type 2: errors within message parameters (IE errors)
  The checking for and handling of errrors is shown in the SDLs in a different w
ay.
  For type 1 errors it is assumed that the common message part of RTMC messages
is checked prior to processing in the
  SDL processes. These errors are represented in the SDLs by pseudo events. The
SDLs show the subsequent error
  handling (MSG_ERROR_HANDLING).
  For type 2 errors the SDLs show the possible results of IE syntax checking (pr
ocedure IE_CHECK) only and the subsequent
  error handling (procedure IE_ERROR_HANDLING)
*/
```

```
use DataModule;
```



```
/* Shall error checking be done */
SYNONYM C_ErrContained BOOLEAN = EXTERNAL;
```

```
SYNONYM
C_PDU INTEGER = EXTERNAL;
/* Used to determine which optional Information Elements are added by the process Glue to */
/* the base PDU. */

SYNONYM
C_PrimSeq INTEGER = EXTERNAL;
/* Used to determine which CCF Service Primitive responds to the VB5.2 primitive sent to the CCF. */
```

```
use DataModule;
```



```
/*Procedures, data and comments for
enabling/facilitating
tool supported simulation*/
```

```
/* Declaration of primitives for establishing the communication
   between the protocol management processes at the AN and SN side. */
```

```
signal
simSnStarted, /* primitive to announce the PId of the SN_PROT_MGR process to the AN_PROT_MGR process */
simAnStarted, /* primitive to announce the PId of the AN_PROT_MGR process to the SN_PROT_MGR process */
AnStarted,
SnStarted;
```

```
/* Example for command file for the startup of the AN and SN VB5 SDL system. After
   system startup further commands may be directed to the simulator
   depending on the specific use case you want to perform.

−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−− begin of file
Go
Output−Via meeBbccStartupReq CH_SNVB5_MF
Proceed−Until +0.1
Output−Via aalBbccEstablishConf CH_SNVB5_SAAL
Output−Via aalBbccEstablishInd CH_ANVB5_SAAL
Proceed−Until +0.1
Proceed−Until +0.1
Output−Via ceeBbccResetAccRes (: (others: (. 0,0 .)) :) CH_ANVB5_CCF
go
−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−− end of file
*/
```

```
/*
Before you can start simulation, the values for the external synonym C_ErrContained must
be set. To avoid direct input during simulation start, you should use an input−file instead.
Example of an input file:
−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−− begin of input file
C_ErrContained   FALSE;
−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−− end of input−file
*/
```

```
/* The VB5 messages DEALLOC, DROP_BRANCH and the corresponding responses and primitives may
   contain a list of connection reference numbers or branch identifiers e.g. several connections may be
   deallocated simultaneously. For simplification of the SDL specification, only one connection or
   branch identifier is used but no list of parameters. Furthermore no segmentation of the messages is
   shown in the SDLs. The segmentation of invoke messages and the collection of  response messages
   has to be taken into account in a real implementation. */
```

```
use DataModule;
```

Package VB5_2


```
/* Naming conventions */
```

```
/*Type definitions for signal parameters*/
/**/
/*In the following the SDL data types of signal parameters are defined. This is done in order
to allow for tool supported simulation and validation of the SDLs. This SDL data modelling
shall not supersede the message and parameter formats and codes as defined in clause 14.
Only those data are modelled which are needed for simulation and validation*/
/**/
/*Prefix Naming Conventions for Data Type Definitions
AT_xxx   ARRAY Data Type
IT_xxx    INTEGER Data Type
PT_xxx   PID Data Type
ST_xxx   STRUCT Data Type*/
/**/
/*Prefix Naming Conventions for Variables and Constants
A_xxx    ARRAY Variable
B_xxx    BOOLEAN Variable
C_xxx    Constant of any type
P_xxx    PID Variable
S_xxx    STRUCT Variable
V_xxx    INTEGER or NATURAL Variable
*/
```

```
/*Prefixes for primitives*/
/**/
/*B−BCC primitives represent in an abstract way the exchange of information and control
between functional entities. They do not specify or constrain implementation. The primitives
are classifed as follows*/
/*
aal        primitives between the B−BCC management and the ATM adaptation layer
mdu        primitives between B−BCC management or connection management and B−BCC protocol
mee        primitives between B−BCC management or B−BCC protocol and the
           B−BCC management application functions
mie        primitives internal to a SDL block
mph        primitives between B−BCC management and connection management
cee        primitives between B−BCC connection management and the connection control
           application functions
*/
```

```
/* Drawing conventions */
```

```
/*In order to distinguish internal and external communication, two different types of input and output
  symbols are used. External signals use input symbols coming from the left and output symbols
  pointing to the left. Internal signals use input symbols coming from the right and output symbols
  pointing to the right. External signals are signals coming from or going to the environment,
  i.e. management entity, SAAL entity and remote side.
*/
```

```
use DataModule;
```



```
/*Data Type Definitions for
  B−BCC messages*/
```

```
/* Information Elements */
/**/
/* Protocol error cause */
SYNTYPE IT_ProtErrCause =INTEGER
ENDSYNTYPE IT_ProtErrCause;
/*Possible Protocol error cause values*/
SYNONYM C_UnkMsgType                    IT_ProtErrCause = 1;
SYNONYM C_RepIE                         IT_ProtErrCause = 2;
SYNONYM C_MandIEMiss                    IT_ProtErrCause = 3;
SYNONYM C_UnrecogIE                     IT_ProtErrCause = 4;
SYNONYM C_IEContErr                     IT_ProtErrCause = 5;
SYNONYM C_IENotAllowed                  IT_ProtErrCause = 6;
/**/
/* Transaction identifier */
SYNTYPE IT_TransId = INTEGER
ENDSYNTYPE IT_TransId;
/**/
/* Connection reference number */
SYNTYPE IT_ConnRefNo = INTEGER
ENDSYNTYPE IT_ConnRefNo;
/**/
/* Branch identifier */
SYNTYPE IT_BranchId = INTEGER
ENDSYNTYPE IT_BranchId;
/* For simulation only, the value ’0’ is used to indicate a not present
   branch identifier. Although this value may be used in a real
   implementation as branch identifier. This SDL data modelling
   does not supersede the message and parameter formats and codes
   as defined in clause 14.*/
SYNONYM  C_BranchIdNotPresent      IT_BranchId = 0;
/**/
/* Reject Cause, changed 7Dec99 to be aligned with 301 217−1, v1.1.3 */
SYNTYPE IT_RejCause = INTEGER
ENDSYNTYPE IT_RejCause;
SYNONYM C_Success                   IT_RejCause  = 255;
SYNONYM C_TransErr                  IT_RejCause = 101;
SYNONYM C_IncompConnState       IT_RejCause = 1;
SYNONYM C_IncompBranchState    IT_RejCause = 2;
SYNONYM C_DeallocExpected         IT_RejCause = 3;
SYNONYM C_OpRejLoc                  IT_RejCause = 0;
SYNONYM C_Saal                      IT_RejCause = 103;
```

```
/* Information Elements */
/* All other IEs are transported transparently trough the vb5−system.
   They are neither set nor evaluated by the vb5−system. */
```

```
use DataModule;
```



```
/*Data type definitions for
  primitives */
```

```
/* Struct necessary to identify a single branch */
NEWTYPE ST_ConnBranchId STRUCT
  V_ConnRefNo   IT_ConnRefNo;
  V_BranchId    IT_BranchId;
ENDNEWTYPE;
SYNONYM  C_EmptyElement ST_ConnBranchId = (. 0, 0 .);
/* Default value of an empty struct.
   For simulation only, the values '0' are used to indicate an empty element.
   Although these values may be used in a real implementation as connection
   reference number or branch identifier. This SDL data modelling does not
   supersede the message and parameter formats and codes as defined in
   clause 14.*/
/**/
/* List of connections, used in reset procedure */
NEWTYPE AT_ConnList ARRAY
   (INTEGER,ST_ConnBranchId)
ENDNEWTYPE;
/**/
/* Update needed in case of addBranch */
SYNTYPE  IT_UpdateNeeded = INTEGER
   CONSTANTS 0:1
ENDSYNTYPE IT_UpdateNeeded;
/* Possible values for IT_UpdateNeeded */
SYNONYM  B_UpdateNotNeeded IT_UpdateNeeded = 0;
SYNONYM  B_UpdateNeeded    IT_UpdateNeeded = 1;
```

```
/*Data type definitions for
  common used data types */
```

```
/* Data type used to indicate the result of the syntax check of the information elements */
SYNTYPE IT_SynResult = INTEGER
ENDSYNTYPE IT_SynResult;
/*Possible values for IT_SynResult */
SYNONYM C_Proceed INTEGER =1;     /*possible result of syntax check: proceed with message processing*/
SYNONYM C_Error INTEGER =2;       /*possible result of syntax check: error in message*/
/**/
/* Data type used to identify the own protocol process in a transaction */
SYNTYPE PT_ProtId = PID
ENDSYNTYPE PT_ProtId;
```

```
use DataModule;
```

Package VB5_2


```
/*Declaration of B−BCC messages*/
```

```
signal
ALLOC            (IT_TransId, IT_ConnRefNo, IT_BranchId),
ALLOC_ACC        (IT_TransId, IT_BranchId),
ALLOC_REJ        (IT_TransId, IT_RejCause),
ALLOC_COMP       (IT_TransId, IT_ConnRefNo),
ALLOC_COMP_ACC   (IT_TransId),
ALLOC_COMP_REJ   (IT_TransId, IT_RejCause),
AN_FAULT         (IT_TransId, IT_ConnRefNo, IT_BranchId),
AN_FAULT_ACC     (IT_TransId),
BBCC_RESET       (IT_TransId),
BBCC_RESET_ACC   (IT_TransId),
BBCC_RESET_REJ   (IT_TransId, IT_RejCause),
BBCC_PRESYNC     (IT_TransId),
BBCC_PRESYNC_ACC (IT_TransId),
BBCC_PRESYNC_REJ (IT_TransId),
DEALLOC          (IT_TransId, IT_ConnRefNo),
DEALLOC_ACC      (IT_TransId),
MODIFY           (IT_TransId, IT_ConnRefNo),
MODIFY_ACC       (IT_TransId),
MODIFY_REJ       (IT_TransId, IT_RejCause),
MODIFY_COMP      (IT_TransId, IT_ConnRefNo),
MODIFY_COMP_ACC  (IT_TransId),
MODIFY_COMP_REJ  (IT_TransId, IT_RejCause),
MODIFY_ABORT     (IT_TransId, IT_ConnRefNo),
MODIFY_ABORT_ACC (IT_TransId),
MODIFY_ABORT_REJ (IT_TransId, IT_RejCause),
ADD_BRANCH       (IT_TransId, IT_ConnRefNo, IT_BranchId),
ADD_BRANCH_ACC   (IT_TransId),
ADD_BRANCH_REJ   (IT_TransId, IT_RejCause),
UPDATE_BRANCH    (IT_TransId, IT_ConnRefNo, IT_BranchId),
UPDATE_BRANCH_ACC (IT_TransId),
UPDATE_BRANCH_REJ (IT_TransId, IT_RejCause),
DROP_BRANCH      (IT_TransId, IT_ConnRefNo, IT_BranchId),
DROP_BRANCH_ACC  (IT_TransId),
DROP_BRANCH_REJ  (IT_TransId, IT_RejCause),
PROTOCOL_ERROR   (IT_TransId, IT_ProtErrCause);
```

```
/*Primitives to the B−BCC protocol entity representing erroneous VB5 messages */
signal
unknown_message(IT_TransId),
/*Representation of a message containing an unspecified message type information element */
msg_too_short,
/*Representation of a message that is too short to contain a complete message length information element */
trans_id_err,
/*Representation of a message containing an invalid transaction identifier format or a transaction identifier
procedural error */
unknown_protdisc;
/*Representation of a message containing an unspecified protocol discriminator information element*/
```

```
use DataModule;
```

Package VB5_2


```
signal
BbccRESET (BbccReset),
BbccRESET_ACC (BbccResetAcc),
BbccRESET_REJ (BbccResetRej),
BbccPRESYNC (BbccPresync),
BbccPRESYNC_ACC (BbccPresyncAcc),
BbccPRESYNC_REJ (BbccPresyncRej),
pduALLOC (Alloc),
pduALLOC_ACC (AllocAcc),
pduALLOC_COMP (AllocComp),
pduALLOC_COMP_ACC (AllocCompAcc),
pduALLOC_REJ (AllocRej),
pduALLOC_COMP_REJ (AllocCompRej),
pduDEALLOC( Dealloc),
pduDEALLOC_ACC( DeallocAcc),
pduADD_BRANCH (AddBranch),
pduADD_BRANCH_ACC (AddBranchAcc),
pduADD_BRANCH_REJ (AddBranchRej),
pduUPDATE_BRANCH (UpdateBranch),
pduUPDATE_BRANCH_ACC (UpdateBranchAcc),
pduUPDATE_BRANCH_REJ (UpdateBranchRej),
pduDROP_BRANCH (DropBranch),
pduDROP_BRANCH_ACC (DropBranchAcc),
pduDROP_BRANCH_REJ (DropBranchRej),
pduMODIFY (Modify),
pduMODIFY_ACC (ModifyAcc),
pduMODIFY_REJ (ModifyRej),
pduMODIFY_COMP (ModifyComp),
pduMODIFY_COMP_ACC (ModifyCompAcc),
pduMODIFY_COMP_REJ (ModifyCompRej),
pduMODIFY_ABORT (ModifyAbort),
pduMODIFY_ABORT_ACC (ModifyAbortAcc),
pduMODIFY_ABORT_REJ (ModifyAbortRej) ,
pduAN_FAULT  (AnFault),
pduAN_FAULT_ACC (AnFaultAcc),
pduAN_unk (UnknownPDU),
pduSN_unk (UnknownPDU),
pduTruncReset (TruncInfoBbccReset),
pduMndtryReset (BbccResetMissingMndtryIE),
pduUnkReset (BbccResetXtraUnkIE),
pduUnexpReset (BbccResetXtraUnexpIE),
pduTruncFault (TruncInfoANFault),
pduMndtryFault (ANFaultMissingMndtryIE),
pduUnkFault (ANFaultXtraUnkIE),
pduUnexpFault (ANFaultXtraUnexpIE),
pduProtoErr (ProtocolError);
```

```
signallist SnPDU_IN =
BbccRESET_ACC,
BbccRESET_REJ,
BbccPRESYNC_ACC,
BbccPRESYNC_REJ,
pduMODIFY_ABORT_REJ,
pduALLOC_ACC,
pduALLOC_REJ,
pduALLOC_COMP_ACC,
pduALLOC_COMP_REJ,
pduDEALLOC_ACC,
pduADD_BRANCH_ACC,
pduADD_BRANCH_REJ,
pduUPDATE_BRANCH_ACC,
pduUPDATE_BRANCH_REJ,
pduDROP_BRANCH_ACC,
pduDROP_BRANCH_REJ,
pduMODIFY_ACC,
pduMODIFY_REJ,
pduMODIFY_COMP_ACC,
pduMODIFY_COMP_REJ,
pduMODIFY_ABORT_ACC,
pduMODIFY_ABORT_REJ,
pduAN_FAULT,
pduSN_unk,
pduTruncFault,
pduMndtryFault,
pduUnkFault,
pduUnexpFault;
```

```
signallist AnPDU_OUT =
BbccRESET_ACC,
BbccRESET_REJ,
BbccPRESYNC_ACC,
BbccPRESYNC_REJ,
pduMODIFY_ABORT_REJ,
pduALLOC_ACC,
pduALLOC_REJ,
pduALLOC_COMP_ACC,
pduALLOC_COMP_REJ,
pduDEALLOC_ACC,
pduADD_BRANCH_ACC,
pduADD_BRANCH_REJ,
pduUPDATE_BRANCH_ACC,
pduUPDATE_BRANCH_REJ,
pduDROP_BRANCH_ACC,
pduDROP_BRANCH_REJ,
pduMODIFY_ACC,
pduMODIFY_REJ,
pduMODIFY_COMP_ACC,
pduMODIFY_COMP_REJ,
pduMODIFY_ABORT_ACC,
pduMODIFY_ABORT_REJ,
pduAN_FAULT,
pduProtoErr ;
```

```
signallist SnPDU_OUT =
BbccRESET,
BbccPRESYNC,
pduALLOC,
pduALLOC_COMP,
pduDEALLOC,
pduADD_BRANCH,
pduUPDATE_BRANCH,
pduDROP_BRANCH,
pduMODIFY,
pduMODIFY_COMP,
pduMODIFY_ABORT,
pduAN_FAULT_ACC,
pduProtoErr ;
```

```
signallist AnPDU_IN =
BbccRESET,
BbccPRESYNC,
pduALLOC,
pduALLOC_COMP,
pduDEALLOC,
pduADD_BRANCH,
pduUPDATE_BRANCH,
pduDROP_BRANCH,
pduMODIFY,
pduMODIFY_COMP,
pduMODIFY_ABORT,
pduAN_FAULT_ACC,
pduAN_unk,
pduTruncReset,
pduMndtryReset,
pduUnkReset,
pduUnexpReset;
```

```
use DataModule;
```

Package VB5_2


```
signal  /* for implicits */
SN_Restart,
SN_ConnEst (IT_ConnRefNo, IT_BranchId),
SN_Dealloc (IT_ConnRefNo),
SN_AddBranch (IT_ConnRefNo, IT_BranchId),
SN_UpdateBranch(IT_ConnRefNo, IT_BranchId),
SN_DropBranch (IT_ConnRefNo, IT_BranchId),
SN_Presync,
SN_Modify (IT_ConnRefNo),
SN_ModifyComp (IT_ConnRefNo),
SN_ModifyAbort(IT_ConnRefNo),
SN_Reset (AT_ConnList) ,
AN_Faulty (IT_ConnRefNo, IT_BranchId);
```

```
signallist  IMPLICITS =
SN_Restart,
SN_ConnEst,
SN_Dealloc,
SN_AddBranch,
SN_UpdateBranch,
SN_DropBranch,
SN_Modify,
SN_ModifyComp,
SN_ModifyAbort,
SN_Presync,
SN_Reset,
AN_Faulty;
```

use DataModule;

Package VB5_2

/* Declaration of primitives
from/to the SAAL */

```
signal
aalBbccEstablishReq,
/*Request from VB5 system management to SAAL to establish SAAL connection to the peer side
*/
aalBbccEstablishConf,
/*Confirmation from SAAL to VB5 system management of successful SAAL establishment
*/
aalBbccEstablishInd,
/*Indication from SAAL to VB5 system management that peer side requested SAAL establishment
*/
aalBbccReleaseReq,
/*Request from VB5 system management to SAAL to release SAAL connection to the peer side
*/
aalBbccReleaseConf,
/*Confirmation from SAAL to VB5 system management of successful SAAL release
*/
aalBbccReleaseInd;
/*Indication from SAAL to VB5 system management that peer side requested SAAL release*/
```

```
use DataModule;
```



```
/*Declaration of common primitives
  from/to the management application function*/
```

```
signal
meeBbccStartTrafficInd,
meeBbccStopTrafficReq,
meeBbccStopTrafficConf,
meeBbccStopTrafficInd,
meeBbccErrorInd;
```

```
/*Declaration of AN specific primitives
from/to the management application function*/
```

```
signal
meeBbccStartTrafficReq,
meeBbccStartTrafficConf,
meeBbccPresyncInd   ,
meeBbccPresyncAccRes,
meeBbccPresyncRejRes;
```

```
use DataModule;
```



```
/*Declaration of AN specific primitives
  from/to the connection and control application function*/
```

```
signal
ceeAllocInd           (IT_ConnRefNo, IT_BranchId),
ceeAllocAccRes        (IT_ConnRefNo, IT_BranchId),
ceeAllocRejRes        (IT_ConnRefNo, IT_BranchId, IT_RejCause),
ceeAllocCompInd       (IT_ConnRefNo),
ceeAllocCompAccRes    (IT_ConnRefNo),
ceeAllocCompRejRes    (IT_ConnRefNo, IT_RejCause),
ceeDeallocInd         (IT_ConnRefNo),
ceeDeallocAccRes      (IT_ConnRefNo),
ceeModifyInd          (IT_ConnRefNo),
ceeModifyAccRes       (IT_ConnRefNo),
ceeModifyRejRes       (IT_ConnRefNo, IT_RejCause),
ceeModifyCompInd      (IT_ConnRefNo),
ceeModifyCompAccRes   (IT_ConnRefNo),
ceeModifyCompRejRes   (IT_ConnRefNo, IT_RejCause),
ceeModifyAbortInd     (IT_ConnRefNo),
ceeModifyAbortAccRes  (IT_ConnRefNo),
ceeModifyAbortRejRes  (IT_ConnRefNo, IT_RejCause),
ceeAddBranchInd       (IT_ConnRefNo, IT_BranchId),
ceeAddBranchAccRes    (IT_ConnRefNo, IT_BranchId, IT_UpdateNeeded),
ceeAddBranchRejRes    (IT_ConnRefNo, IT_BranchId, IT_RejCause),
ceeUpdateBranchInd    (IT_ConnRefNo, IT_BranchId),
ceeUpdateBranchAccRes (IT_ConnRefNo, IT_BranchId),
ceeUpdateBranchRejRes (IT_ConnRefNo, IT_BranchId, IT_RejCause),
ceeDropBranchInd      (IT_ConnRefNo, IT_BranchId),
ceeDropBranchAccRes   (IT_ConnRefNo, IT_BranchId),
ceeDropBranchRejRes (IT_ConnRefNo, IT_BranchId, IT_RejCause),
ceeBbccResetInd       ,
ceeBbccResetAccRes    (AT_ConnList),
ceeBbccResetRejRes    (IT_RejCause),
ceeAnFaultReq         (IT_ConnRefNo, IT_BranchId),
ceeAnFaultAccConf     (IT_ConnRefNo, IT_BranchId),
ceeAnFaultRejConf     (IT_ConnRefNo, IT_BranchId, IT_RejCause);
```

```
/*
It is assumed that the ceeAnFaultReq primitive
at the management interface always contains at
least the connection refererence number and
optionally a branch identifier as parameters.
This does not mean that they are included as
parameters in the AN_FAULT message at the
VB5.2 reference point. They may be omitted if the
user port connection identifer or the service port
connection identifier are present.
*/
```

```
use DataModule;
```

Package VB5_2


```
/*Declaration of SN specific primitives
from/to the connection and control application function*/
```

```
signal
ceeAllocReq            (IT_ConnRefNo, IT_BranchId),
ceeAllocAccConf        (IT_ConnRefNo, IT_BranchId),
ceeAllocRejConf        (IT_ConnRefNo, IT_BranchId, IT_RejCause),
ceeAllocCompReq        (IT_ConnRefNo),
ceeAllocCompAccConf    (IT_ConnRefNo),
ceeAllocCompRejConf    (IT_ConnRefNo, IT_RejCause),
ceeDeallocReq          (IT_ConnRefNo),
ceeDeallocAccConf      (IT_ConnRefNo),
ceeDeallocRejConf      (IT_ConnRefNo, IT_RejCause),
ceeModifyReq           (IT_ConnRefNo),
ceeModifyAccConf       (IT_ConnRefNo),
ceeModifyRejConf       (IT_ConnRefNo, IT_RejCause),
ceeModifyCompReq       (IT_ConnRefNo),
ceeModifyCompAccConf   (IT_ConnRefNo),
ceeModifyCompRejConf   (IT_ConnRefNo, IT_RejCause),
ceeModifyAbortReq      (IT_ConnRefNo),
ceeModifyAbortAccConf  (IT_ConnRefNo),
ceeModifyAbortRejConf  (IT_ConnRefNo, IT_RejCause),
ceeAddBranchReq        (IT_ConnRefNo, IT_BranchId),
ceeAddBranchAccConf    (IT_ConnRefNo, IT_BranchId),
ceeAddBranchRejConf    (IT_ConnRefNo, IT_BranchId, IT_RejCause),
ceeUpdateBranchReq     (IT_ConnRefNo, IT_BranchId),
ceeUpdateBranchAccConf (IT_ConnRefNo, IT_BranchId),
ceeUpdateBranchRejConf (IT_ConnRefNo, IT_BranchId, IT_RejCause),
ceeDropBranchReq       (IT_ConnRefNo, IT_BranchId),
ceeDropBranchAccConf   (IT_ConnRefNo, IT_BranchId),
ceeDropBranchRejConf   (IT_ConnRefNo, IT_BranchId, IT_RejCause),
ceeBbccResetReq        (AT_ConnList),
ceeBbccResetAccConf    ,
ceeBbccResetRejConf    (IT_RejCause),
ceeAnFaultInd          (IT_ConnRefNo, IT_BranchId);
```

```
/*Declaration of SN specific primitives
from/to the management application function*/
```

```
signal
meeBbccStartupReq      ,
meeBbccStartupConf     (IT_RejCause),
meeBbccRestartReq      ,
meeBbccRestartConf     (IT_RejCause),
meeBbccStartResetInd;
```

```
use DataModule;
```

Package VB5_2


```
/*Declaration of common primitives
from/to BCC protocol entitiy*/
```

```
signal
mduStartTraffic,
mduStopTraffic,
mduErrorInd;
```

```
/*Declaration of AN specific primitives
  from/to BCC protocol entity*/
```

```
signal
mduAllocInd            (PT_ProtId, IT_ConnRefNo, IT_BranchId),
mduAllocAccRes         (PT_ProtId, IT_ConnRefNo, IT_BranchId),
mduAllocRejRes         (PT_ProtId, IT_ConnRefNo, IT_BranchId, IT_RejCause),
mduAllocCompInd        (PT_ProtId, IT_ConnRefNo),
mduAllocCompAccRes     (PT_ProtId, IT_ConnRefNo),
mduAllocCompRejRes     (PT_ProtId, IT_ConnRefNo, IT_RejCause),
mduDeallocInd          (PT_ProtId, IT_ConnRefNo),
mduDeallocAccRes       (PT_ProtId, IT_ConnRefNo),
mduModifyInd           (PT_ProtId, IT_ConnRefNo),
mduModifyAccRes        (PT_ProtId, IT_ConnRefNo),
mduModifyRejRes        (PT_ProtId, IT_ConnRefNo, IT_RejCause),
mduModifyCompInd       (PT_ProtId, IT_ConnRefNo),
mduModifyCompAccRes    (PT_ProtId, IT_ConnRefNo),
mduModifyCompRejRes    (PT_ProtId, IT_ConnRefNo, IT_RejCause),
mduModifyAbortInd      (PT_ProtId, IT_ConnRefNo),
mduModifyAbortAccRes   (PT_ProtId, IT_ConnRefNo),
mduModifyAbortRejRes   (PT_ProtId, IT_ConnRefNo, IT_RejCause),
mduAddBranchInd        (PT_ProtId, IT_ConnRefNo, IT_BranchId),
mduAddBranchAccRes     (PT_ProtId, IT_ConnRefNo, IT_BranchId),
mduAddBranchRejRes     (PT_ProtId, IT_ConnRefNo, IT_BranchId, IT_RejCause),
mduUpdateBranchInd     (PT_ProtId, IT_ConnRefNo, IT_BranchId),
mduUpdateBranchAccRes  (PT_ProtId, IT_ConnRefNo, IT_BranchId),
mduUpdateBranchRejRes  (PT_ProtId, IT_ConnRefNo, IT_BranchId, IT_RejCause),
mduDropBranchInd       (PT_ProtId, IT_ConnRefNo, IT_BranchId),
mduDropBranchAccRes    (PT_ProtId, IT_ConnRefNo, IT_BranchId),
mduDropBranchRejRes    (PT_ProtId, IT_ConnRefNo, IT_BranchId, IT_RejCause),
mduAnFaultReq          (IT_ConnRefNo, IT_BranchId),
mduAnFaultAccConf      (IT_ConnRefNo, IT_BranchId),
mduAnFaultRejConf      (IT_ConnRefNo, IT_BranchId, IT_RejCause),
mduBbccResetInd        (PT_ProtId),
mduBbccResetAccRes     (PT_ProtId),
mduBbccResetRejRes     (PT_ProtId, IT_RejCause),
mduBbccPresyncInd      (PT_ProtId),
mduBbccPresyncAccRes   (PT_ProtId),
mduBbccPresyncRejRes   (PT_ProtId);
```

use DataModule;

/*Declaration of SN specific primitives
from/to BCC protocol entity*/

```
signal
mduAllocReq            (IT_ConnRefNo, IT_BranchId),
mduAllocAccConf        (IT_ConnRefNo, IT_BranchId),
mduAllocRejConf        (IT_ConnRefNo, IT_BranchId, IT_RejCause),
mduAllocCompReq        (IT_ConnRefNo),
mduAllocCompAccConf    (IT_ConnRefNo),
mduAllocCompRejConf    (IT_ConnRefNo, IT_RejCause),
mduDeallocReq          (IT_ConnRefNo),
mduDeallocAccConf      (IT_ConnRefNo),
mduDeallocRejConf      (IT_ConnRefNo, IT_RejCause),
mduModifyReq           (IT_ConnRefNo),
mduModifyAccConf       (IT_ConnRefNo),
mduModifyRejConf       (IT_ConnRefNo, IT_RejCause),
mduModifyCompReq       (IT_ConnRefNo),
mduModifyCompAccConf   (IT_ConnRefNo),
mduModifyCompRejConf   (IT_ConnRefNo, IT_RejCause),
mduModifyAbortReq      (IT_ConnRefNo),
mduModifyAbortAccConf  (IT_ConnRefNo),
mduModifyAbortRejConf  (IT_ConnRefNo, IT_RejCause),
mduAddBranchReq        (IT_ConnRefNo, IT_BranchId),
mduAddBranchAccConf    (IT_ConnRefNo, IT_BranchId),
mduAddBranchRejConf    (IT_ConnRefNo, IT_BranchId, IT_RejCause),
mduUpdateBranchReq     (IT_ConnRefNo, IT_BranchId),
mduUpdateBranchAccConf (IT_ConnRefNo, IT_BranchId),
mduUpdateBranchRejConf (IT_ConnRefNo, IT_BranchId, IT_RejCause),
mduDropBranchReq       (IT_ConnRefNo, IT_BranchId),
mduDropBranchAccConf   (IT_ConnRefNo, IT_BranchId),
mduDropBranchRejConf   (IT_ConnRefNo, IT_BranchId, IT_RejCause),
mduAnFaultInd          (IT_ConnRefNo, IT_BranchId),
mduBbccResetReq        ,
mduBbccResetAccConf    ,
mduBbccResetRejConf    (IT_RejCause),
mduBbccPresyncReq      ,
mduBbccPresyncAccConf  ,
mduBbccPresyncRejConf  (IT_RejCause);
```

```
use DataModule;
```

Package VB5_2


```
/*Declaration of common primitives
  between BCC management and connection management */
```

```
signal
mphStopTraffic,
mphStartTraffic;
```

```
/*Declaration of  SN specific primitives
  between BCC management and connection management */
```

```
signal
mphBbccResetReq,
mphBbccResetAccConf,
mphBbccResetRejConf (IT_RejCause);
```

use DataModule;

Package VB5_2

/*Declaration of signallists
common for AN and SN,
from/to block BCCMGT */

signallist SL_CTR_SAAL =
aalBbccEstablishReq,
aalBbccReleaseReq;

signallist SL_SAAL_CTR =
aalBbccEstablishConf,
aalBbccEstablishInd,
aalBbccReleaseConf,
aalBbccReleaseInd;

signallist SL_CTR_PROT =
mduStartTraffic,
mduStopTraffic;

signallist SL_PROT_CTR =
mduErrorInd;

```
use DataModule;
```

Package VB5_2


```
/*Declaration of signallists
  from/to block AN_BCCMGT */
```

```
signallist SL_MF_ANCTR =
meeBbccStartTrafficReq,
meeBbccStopTrafficReq;
```

```
signallist SL_ANCTR_MF =
meeBbccStartTrafficConf,
meeBbccStartTrafficInd,
meeBbccStopTrafficConf,
meeBbccStopTrafficInd,
meeBbccErrorInd;
```

```
signallist SL_ANPRESYNC_PROT =
mduBbccPresyncAccRes,
mduBbccPresyncRejRes;
```

```
signallist SL_PROT_ANPRESYNC =
mduBbccPresyncInd;
```

```
signallist SL_ANPRESYNC_MF =
meeBbccPresyncInd;
```

```
signallist SL_MF_ANPRESYNC =
meeBbccPresyncAccRes,
meeBbccPresyncRejRes;
```

```
signallist SL_CTR_ANCONN =
mphStartTraffic,
mphStopTraffic;
```

```
signallist SL_ANMGT_PROT =
(SL_CTR_PROT),
(SL_ANPRESYNC_PROT);
```

```
signallist SL_PROT_ANMGT =
(SL_PROT_ANPRESYNC),
(SL_PROT_CTR);
```

```
signallist SL_ANMGT_CONN =
(SL_CTR_ANCONN);
```

```
use DataModule;
```

Package VB5_2


```
/*Declaration of signallists
  from/to block AN_CONNMGT */
```

```
signallist SL_ANCMGR_CCF =
ceeAllocInd,
ceeAllocCompInd,
ceeDeallocInd,
ceeModifyInd,
ceeModifyCompInd,
ceeModifyAbortInd,
ceeAddBranchInd,
ceeUpdateBranchInd,
ceeDropBranchInd,
ceeBbccResetInd,
ceeAnFaultAccConf,
ceeAnFaultRejConf;
```

```
signallist SL_CCF_ANCMGR =
ceeAllocAccRes,
ceeAllocRejRes,
ceeAllocCompAccRes,
ceeAllocCompRejRes,
ceeDeallocAccRes,
ceeModifyAccRes,
ceeModifyRejRes,
ceeModifyCompAccRes,
ceeModifyCompRejRes,
ceeModifyAbortAccRes,
ceeModifyAbortRejRes,
ceeAddBranchAccRes,
ceeAddBranchRejRes,
ceeUpdateBranchAccRes,
ceeUpdateBranchRejRes,
ceeDropBranchAccRes,
ceeDropBranchRejRes,
ceeBbccResetAccRes,
ceeBbccResetRejRes,
ceeAnFaultReq;
```

```
signallist SL_ANCONN_CCF =
(SL_ANCMGR_CCF);
```

```
signallist SL_CCF_ANCONN =
(SL_CCF_ANCMGR);
```

```
signallist SL_ANPROT_CMGR =
mduAllocInd,
mduAllocCompInd,
mduDeallocInd,
mduModifyInd,
mduModifyCompInd,
mduModifyAbortInd,
mduAddBranchInd,
mduUpdateBranchInd,
mduDropBranchInd,
mduBbccResetInd,
mduAnFaultAccConf,
mduAnFaultRejConf;
```

```
signallist SL_CMGR_ANPROT =
mduAllocAccRes,
mduAllocRejRes,
mduAllocCompAccRes,
mduAllocCompRejRes,
mduDeallocAccRes,
mduModifyAccRes,
mduModifyRejRes,
mduModifyCompAccRes,
mduModifyCompRejRes,
mduModifyAbortAccRes,
mduModifyAbortRejRes,
mduAddBranchAccRes,
mduAddBranchRejRes,
mduUpdateBranchAccRes,
mduUpdateBranchRejRes,
mduDropBranchAccRes,
mduDropBranchRejRes,
mduBbccResetAccRes,
mduBbccResetRejRes,
mduAnFaultReq;
```

```
signallist SL_ANPROT_CONN =
(SL_ANPROT_CMGR);
```

```
signallist SL_CONN_ANPROT =
(SL_CMGR_ANPROT);
```

```
signallist SL_ANMGT_CMGR =
(SL_ANMGT_CONN);
```

use DataModule;

Package VB5_2

/*Declaration of signallists
from/to block AN_BCCPROT */

```
signallist SL_VB5_ANPROT =
ALLOC,
ALLOC_COMP,
AN_FAULT_ACC,
BBCC_RESET,
BBCC_PRESYNC,
DEALLOC,
MODIFY,
MODIFY_COMP,
MODIFY_ABORT,
ADD_BRANCH,
UPDATE_BRANCH,
DROP_BRANCH,
PROTOCOL_ERROR,
unknown_protdisc,
unknown_message,
msg_too_short,
trans_id_err,
simSnStarted;
```

```
signallist SL_ANPROT_VB5 =
ALLOC_ACC,
ALLOC_REJ,
ALLOC_COMP_ACC,
ALLOC_COMP_REJ,
AN_FAULT,
BBCC_RESET_ACC,
BBCC_RESET_REJ,
BBCC_PRESYNC_ACC,
BBCC_PRESYNC_REJ,
DEALLOC_ACC,
MODIFY_ACC,
MODIFY_REJ,
MODIFY_COMP_ACC,
MODIFY_COMP_REJ,
MODIFY_ABORT_ACC,
MODIFY_ABORT_REJ,
ADD_BRANCH_ACC,
ADD_BRANCH_REJ,
UPDATE_BRANCH_ACC,
UPDATE_BRANCH_REJ,
DROP_BRANCH_ACC,
DROP_BRANCH_REJ,
PROTOCOL_ERROR,
unknown_protdisc,
unknown_message,
msg_too_short,
trans_id_err,
simAnStarted;
```

```
use DataModule;
```

Package VB5_2


```
/*Declaration of signallists
  from/to block SN_BCCMGT */
```

```
signallist SL_MF_SNCTR =
meeBbccStopTrafficReq,
meeBbccStartupReq,
meeBbccRestartReq;
```

```
signallist SL_SNCTR_MF =
meeBbccStartTrafficInd,
meeBbccStopTrafficConf,
meeBbccStopTrafficInd,
meeBbccStartupConf,
meeBbccRestartConf,
meeBbccStartResetInd,
meeBbccErrorInd;
```

```
signallist SL_SNPRESYNC_PROT =
mduBbccPresyncReq;
```

```
signallist SL_PROT_SNPRESYNC =
mduBbccPresyncAccConf,
mduBbccPresyncRejConf;
```

```
signallist SL_CTR_SNCONN =
mphStartTraffic,
mphStopTraffic,
mphBbccResetReq;
```

```
signallist SL_SNCONN_CTR =
mphBbccResetRejConf,
mphBbccResetAccConf;
```

```
signallist SL_SNMGT_PROT =
(SL_CTR_PROT),
(SL_SNPRESYNC_PROT);
```

```
signallist SL_PROT_SNMGT =
(SL_PROT_SNPRESYNC),
(SL_PROT_CTR);
```

```
signallist SL_SNMGT_CONN =
(SL_CTR_SNCONN);
```

```
signallist SL_CONN_SNMGT =
(SL_SNCONN_CTR);
```

```
use DataModule;
```

Package VB5_2


```
/*Declaration of signallists
  from/to block SN_CONNMGT */
```

```
signallist SL_SNCMGR_CCF =
ceeAllocAccConf,
ceeAllocRejConf,
ceeAllocCompAccConf,
ceeAllocCompRejConf,
ceeDeallocAccConf,
ceeDeallocRejConf,
ceeModifyAccConf,
ceeModifyRejConf,
ceeModifyCompAccConf,
ceeModifyCompRejConf,
ceeModifyAbortAccConf,
ceeModifyAbortRejConf,
ceeAddBranchAccConf,
ceeAddBranchRejConf,
ceeUpdateBranchAccConf,
ceeUpdateBranchRejConf,
ceeDropBranchAccConf,
ceeDropBranchRejConf,
ceeBbccResetAccConf,
ceeBbccResetRejConf,
ceeAnFaultInd;
```

```
signallist SL_CCF_SNCMGR =
ceeAllocReq,
ceeAllocCompReq,
ceeDeallocReq,
ceeModifyReq,
ceeModifyCompReq,
ceeModifyAbortReq,
ceeAddBranchReq,
ceeUpdateBranchReq,
ceeDropBranchReq,
ceeBbccResetReq;
```

```
signallist SL_SNCONN_CCF =
(SL_SNCMGR_CCF);
```

```
signallist SL_CCF_SNCONN =
(SL_CCF_SNCMGR);
```

```
signallist SL_CMGR_SNPROT =
mduAllocReq,
mduAllocCompReq,
mduDeallocReq,
mduModifyReq,
mduModifyCompReq,
mduModifyAbortReq,
mduAddBranchReq,
mduUpdateBranchReq,
mduDropBranchReq,
mduBbccResetReq;
```

```
signallist SL_SNPROT_CMGR =
mduAllocAccConf,
mduAllocRejConf,
mduAllocCompAccConf,
mduAllocCompRejConf,
mduDeallocAccConf,
mduDeallocRejConf,
mduModifyAccConf,
mduModifyRejConf,
mduModifyCompAccConf,
mduModifyCompRejConf,
mduModifyAbortAccConf,
mduModifyAbortRejConf,
mduAddBranchAccConf,
mduAddBranchRejConf,
mduUpdateBranchAccConf,
mduUpdateBranchRejConf,
mduDropBranchAccConf,
mduDropBranchRejConf,
mduBbccResetAccConf,
mduBbccResetRejConf,
mduAnFaultInd;
```

```
signallist SL_CONN_SNPROT =
(SL_CMGR_SNPROT);
```

```
signallist SL_SNPROT_CONN =
(SL_SNPROT_CMGR);
```

```
signallist SL_SNMGT_CMGR =
(SL_SNMGT_CONN);
```

```
signallist SL_CMGR_SNMGT =
(SL_CONN_SNMGT);
```

use DataModule;

Package VB5_2

/*Declaration of signallists
from/to block SN_BCCPROT */

```
signallist SL_SNPROT_VB5 =
ALLOC,
ALLOC_COMP,
AN_FAULT_ACC,
BBCC_RESET,
BBCC_PRESYNC,
DEALLOC,
MODIFY,
MODIFY_COMP,
MODIFY_ABORT,
ADD_BRANCH,
UPDATE_BRANCH,
DROP_BRANCH,
PROTOCOL_ERROR,
unknown_protdisc,
unknown_message,
msg_too_short,
trans_id_err,
simSnStarted;
```

```
signallist SL_VB5_SNPROT =
ALLOC_ACC,
ALLOC_REJ,
ALLOC_COMP_ACC,
ALLOC_COMP_REJ,
AN_FAULT,
BBCC_RESET_ACC,
BBCC_RESET_REJ,
BBCC_PRESYNC_ACC,
BBCC_PRESYNC_REJ,
DEALLOC_ACC,
MODIFY_ACC,
MODIFY_REJ,
MODIFY_COMP_ACC,
MODIFY_COMP_REJ,
MODIFY_ABORT_ACC,
MODIFY_ABORT_REJ,
ADD_BRANCH_ACC,
ADD_BRANCH_REJ,
UPDATE_BRANCH_ACC,
UPDATE_BRANCH_REJ,
DROP_BRANCH_ACC,
DROP_BRANCH_REJ,
PROTOCOL_ERROR,
unknown_protdisc,
unknown_message,
msg_too_short,
trans_id_err,
simAnStarted;
```

use DataModule;

Package VB5_2

/*Declaration of signallists,
common for block type AN and SN */

signallist SL_SAAL_MGT =
(SL_SAAL_CTR);

signallist SL_SAAL_SYS =
(SL_SAAL_MGT);

signallist SL_MGT_SAAL =
(SL_CTR_SAAL);

signallist SL_SYS_SAAL =
(SL_MGT_SAAL);

/*Declaration of signallists,
for block type AN*/

signallist SL_MF_ANMGT =
(SL_MF_ANCTR),
(SL_MF_ANPRESYNC);

signallist SL_MF_ANSYS =
(SL_MF_ANMGT);

signallist SL_VB5_ANSYS =
(SL_VB5_ANPROT);

signallist SL_ANMGT_MF =
(SL_ANCTR_MF),
(SL_ANPRESYNC_MF);

signallist SL_ANSYS_MF =
(SL_ANMGT_MF);

signallist SL_ANSYS_VB5 =
(SL_ANPROT_VB5);

signallist SL_ANSYS_CCF =
(SL_ANCONN_CCF);

signallist SL_CCF_ANSYS =
(SL_CCF_ANCONN);

signallist SL_ANSYS_ENV =
(SL_ANSYS_MF),
(SL_ANSYS_CCF);

signallist SL_ENV_ANSYS =
(SL_MF_ANSYS),
(SL_CCF_ANSYS);

use DataModule;

Package VB5_2

/*Declaration of signallists,
for block type SN*/

signallist SL_MF_SNMGT =
(SL_MF_SNCTR);

signallist SL_MF_SNSYS =
(SL_MF_SNMGT);

signallist SL_VB5_SNSYS =
(SL_VB5_SNPROT);

signallist SL_SNMGT_MF =
(SL_SNCTR_MF);

signallist SL_SNSYS_MF =
(SL_SNMGT_MF);

signallist SL_SNSYS_VB5 =
(SL_SNPROT_VB5);

signallist SL_SNSYS_CCF =
(SL_SNCONN_CCF);

signallist SL_CCF_SNSYS =
(SL_CCF_SNCONN);

signallist SL_SNSYS_ENV =
(SL_SNSYS_CCF),
(SL_SNSYS_MF);

signallist SL_ENV_SNSYS =
(SL_CCF_SNSYS),
(SL_MF_SNSYS);

```
use DataModule;
```

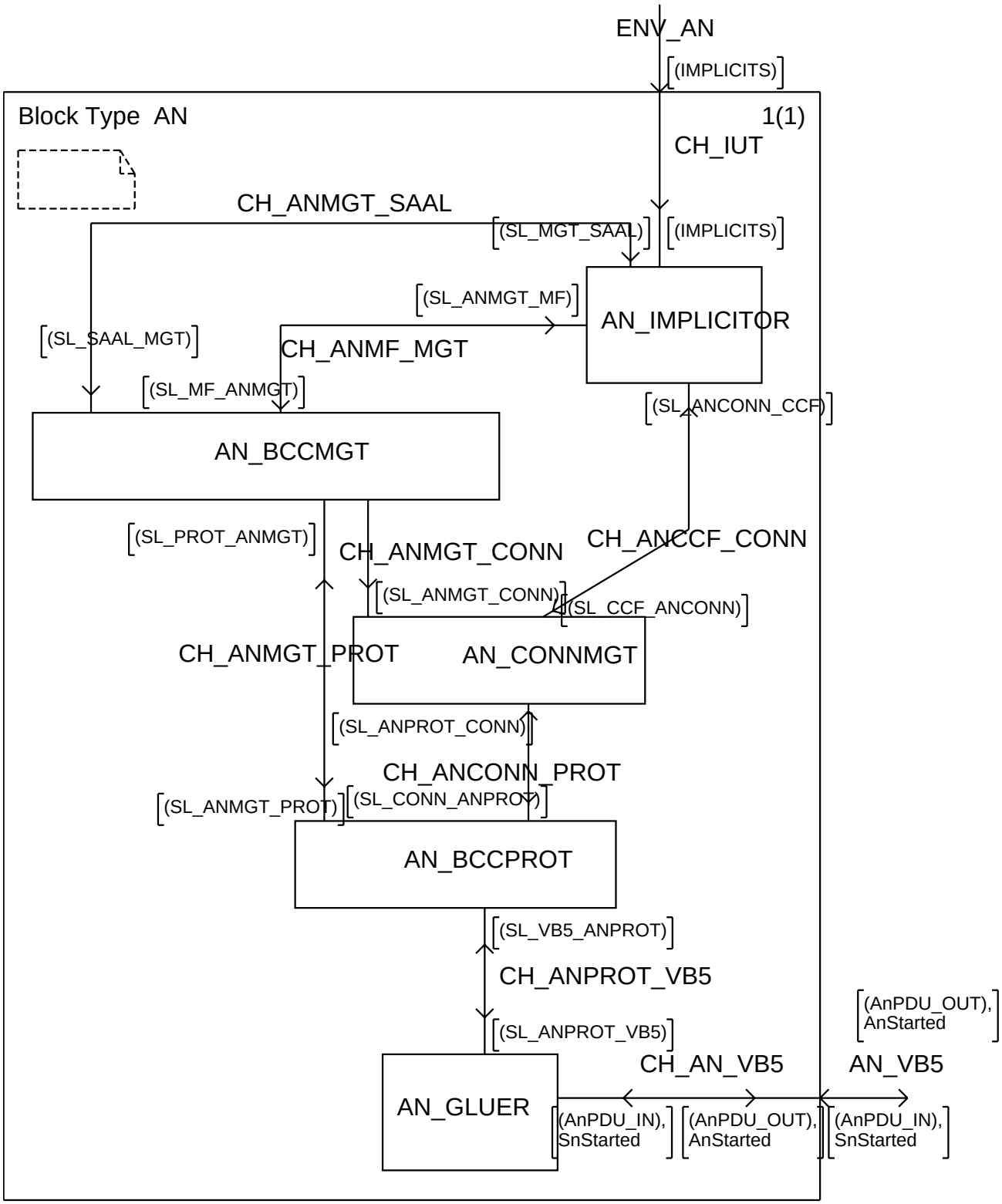
Package VB5_2

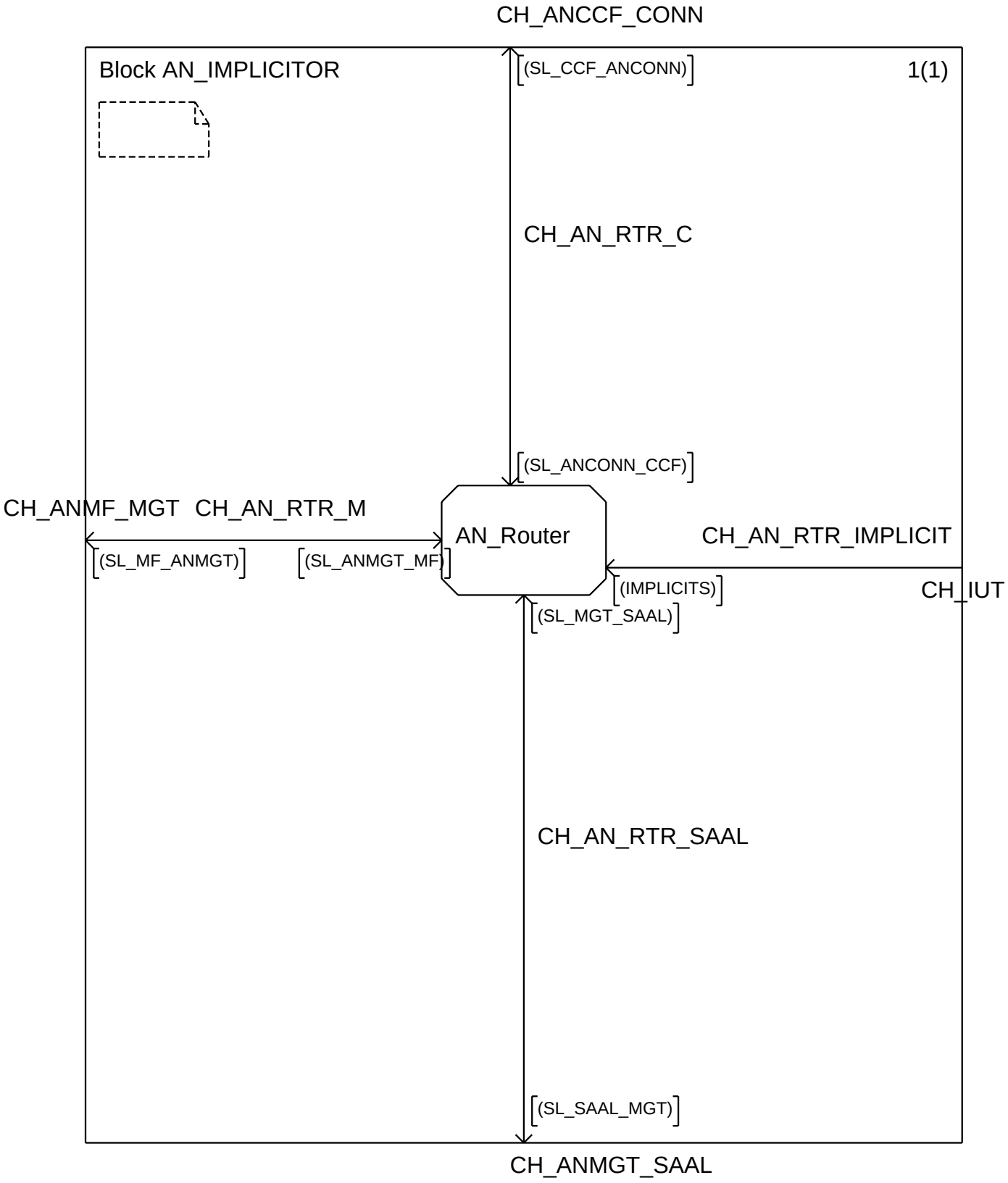

```
/* Declaration of common primitives internal
   to AN_BCCPROT and SN_BCCPROT */
```

```
signal
bccAlloc            (IT_TransId, IT_ConnRefNo, IT_BranchId),
bccAllocAcc         (IT_TransId, IT_BranchId),
bccAllocRej         (IT_TransId, IT_RejCause),
bccAllocComp        (IT_TransId, IT_ConnRefNo),
bccAllocCompAcc     (IT_TransId),
bccAllocCompRej     (IT_TransId, IT_RejCause),
bccBbccReset        (IT_TransId),
bccBbccResetAcc     (IT_TransId),
bccBbccResetRej     (IT_TransId, IT_RejCause),
bccBbccPresync      (IT_TransId),
bccBbccPresyncAcc   (IT_TransId),
bccBbccPresyncRej   (IT_TransId),
bccDealloc          (IT_TransId, IT_ConnRefNo),
bccDeallocAcc       (IT_TransId),
bccModify           (IT_TransId, IT_ConnRefNo),
bccModifyAcc        (IT_TransId),
bccModifyRej        (IT_TransId, IT_RejCause),
bccModifyComp       (IT_TransId, IT_ConnRefNo),
bccModifyCompAcc    (IT_TransId),
bccModifyCompRej    (IT_TransId, IT_RejCause),
bccModifyAbort      (IT_TransId, IT_ConnRefNo),
bccModifyAbortAcc   (IT_TransId),
bccModifyAbortRej   (IT_TransId, IT_RejCause),
bccAddBranch        (IT_TransId, IT_ConnRefNo, IT_BranchId),
bccAddBranchAcc     (IT_TransId),
bccAddBranchRej     (IT_TransId, IT_RejCause),
bccUpdateBranch     (IT_TransId, IT_ConnRefNo, IT_BranchId),
bccUpdateBranchAcc  (IT_TransId),
bccUpdateBranchRej  (IT_TransId, IT_RejCause),
bccDropBranch       (IT_TransId, IT_ConnRefNo, IT_BranchId),
bccDropBranchAcc    (IT_TransId),
bccDropBranchRej    (IT_TransId, IT_RejCause),
bccAnFault          (IT_TransId, IT_ConnRefNo, IT_BranchId),
bccAnFaultAcc       (IT_TransId),
bccProtocolError    (IT_TransId, IT_ProtErrCause),
bccUnknownMessage   (IT_TransId),
bccUnknownProtdisc,
bccMsgTooShort,
bccTransIdErr,
mieProtocolError (IT_TransId, IT_ProtErrCause),
mieAbortReq,
mieErrorInd;
```

```
signal
mieStartTraffic,
mieStopTraffic;
```

```
/* Declaration of common primitives internal
   to AN_BCCMGT and SN_BCCMGT */
```

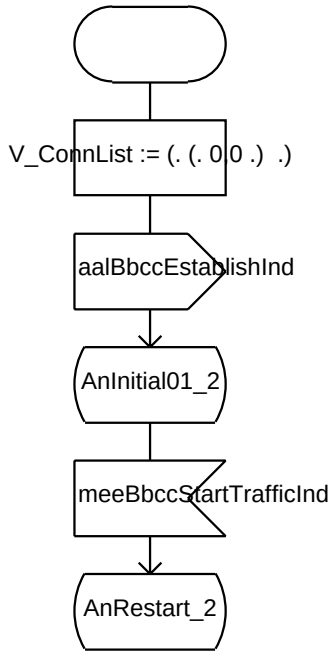

ENV_AN
(IMPLICITS)
Block Type AN
1(1)
CH_IUT
CH_ANMGT_SAAL
(SL_MGT_SAAL)
(IMPLICITS)
(SL_ANMGT_MF)
AN_IMPLICITOR
(SL_SAAL_MGT)
CH_ANMF_MGT
(SL_MF_ANMGT)
(SL_ANCONN_CCF)
AN_BCCMGT
(SL_PROT_ANMGT)
CH_ANMGT_CONN
CH_ANCCF_CONN
(SL_ANMGT_CONN)
(SL_CCF_ANCONN)
CH_ANMGT_PROT
AN_CONNMGT
(SL_ANPROT_CONN)
CH_ANCONN_PROT
(SL_ANMGT_PROT)
(SL_CONN_ANPROT)
AN_BCCPROT
(SL_VB5_ANPROT)
CH_ANPROT_VB5
(AnPDU_OUT),
AnStarted
(SL_ANPROT_VB5)
CH_AN_VB5
AN_VB5
AN_GLUER
(AnPDU_IN),
SnStarted
(AnPDU_OUT),
AnStarted
(AnPDU_IN),
SnStarted

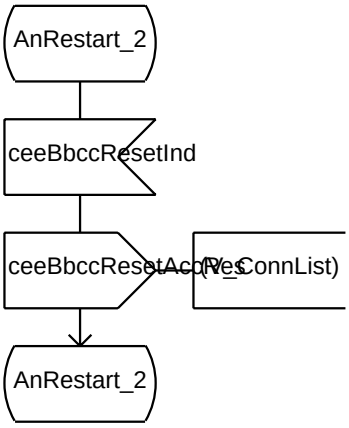

CH_ANCCF_CONN
Block AN_IMPLICITOR
1(1)
(SL_CCF_ANCONN)
CH_AN_RTR_C
(SL_ANCONN_CCF)
CH_ANMF_MGT
CH_AN_RTR_M
AN_Router
CH_AN_RTR_IMPLICIT
(SL_MF_ANMGT)
(SL_ANMGT_MF)
(IMPLICITS)
CH_IUT
(SL_MGT_SAAL)
CH_AN_RTR_SAAL
(SL_SAAL_MGT)
CH_ANMGT_SAAL

/* AN_Router is responsible for receiving and returning service primitives which normally should be sent to the surrounding system. */

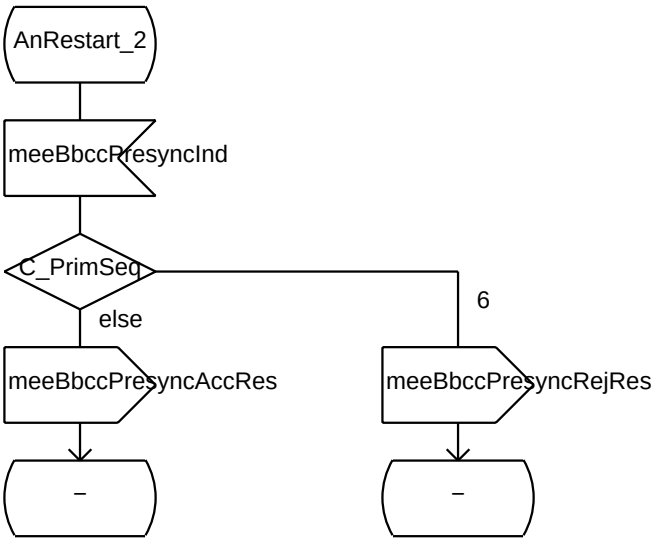
```
DCL
V_RejCause IT_RejCause,
V_ConnRefNo IT_ConnRefNo,
V_BranchId  IT_BranchId,
V_ConnList  AT_ConnList := (. (. 0, 0 .) .);
```

/* AN_Router is responsible for receiving and returning service primitives which normally should be sent to the surrounding system. */

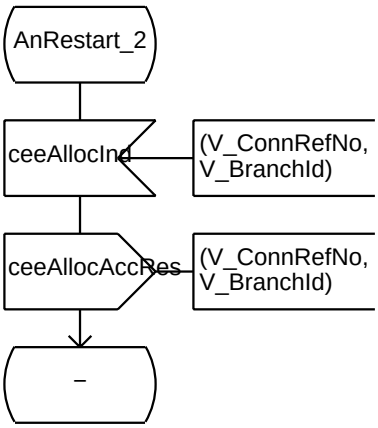


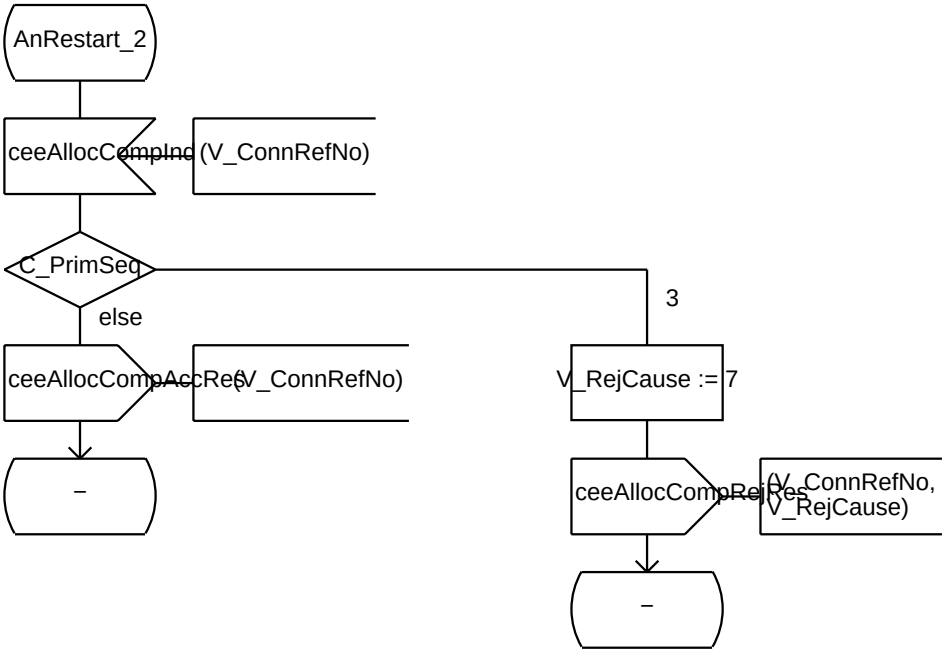


/* AN_Router is responsible for receiving and returning service primitives which normally should be sent to the surrounding system. */

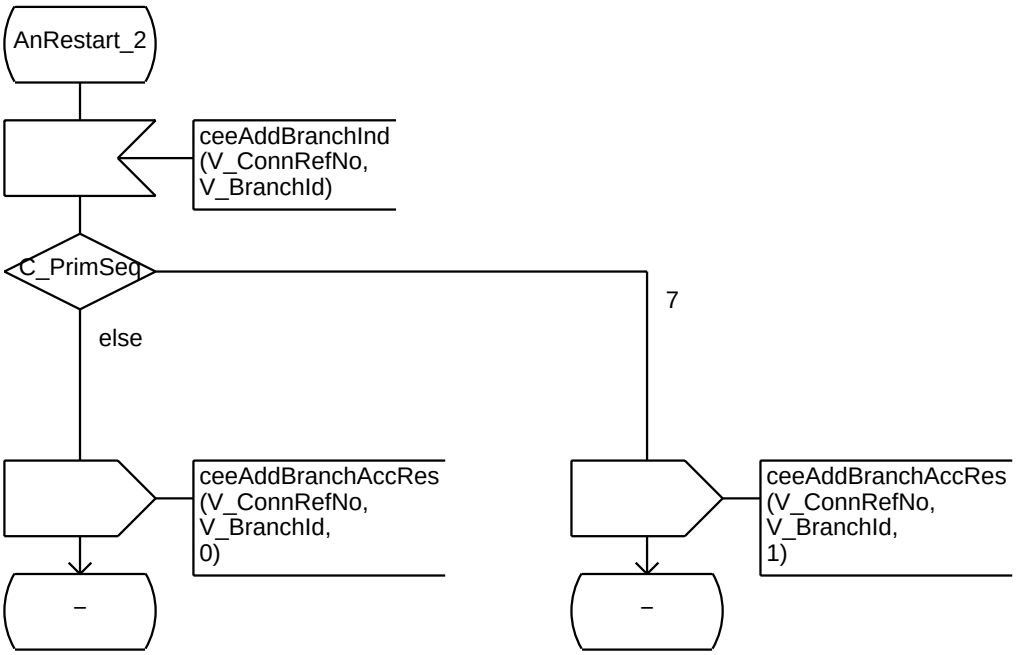


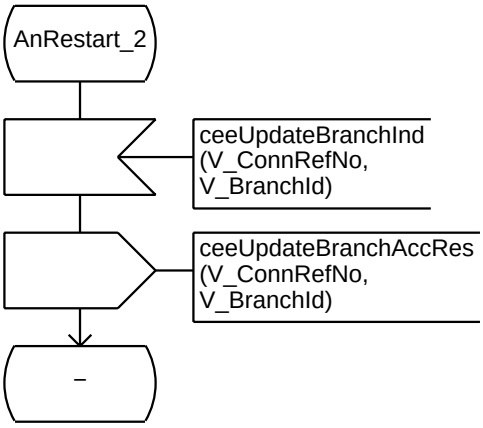
/* AN_Router is responsible for receiving and returning service primitives which normally should be sent to the surrounding system. */



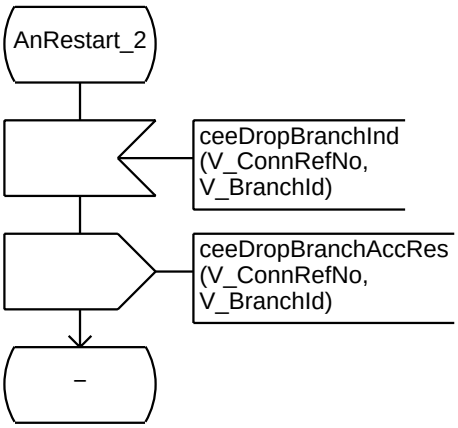


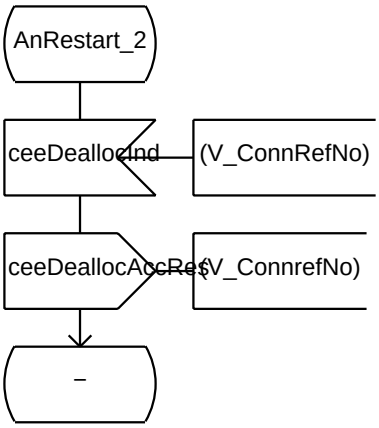
/* AN_Router is responsible for receiving and returning service primitives which normally should be sent to the surrounding system. */



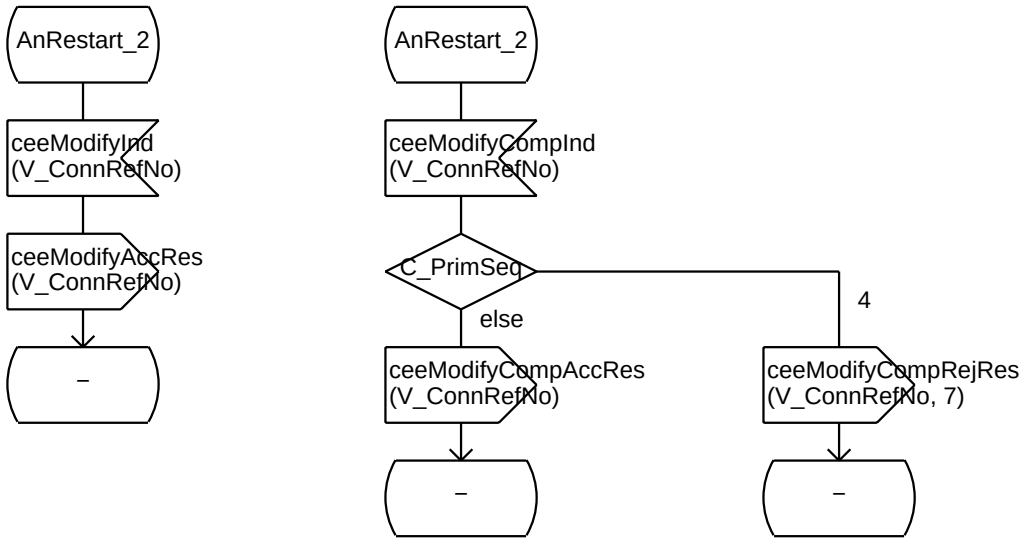


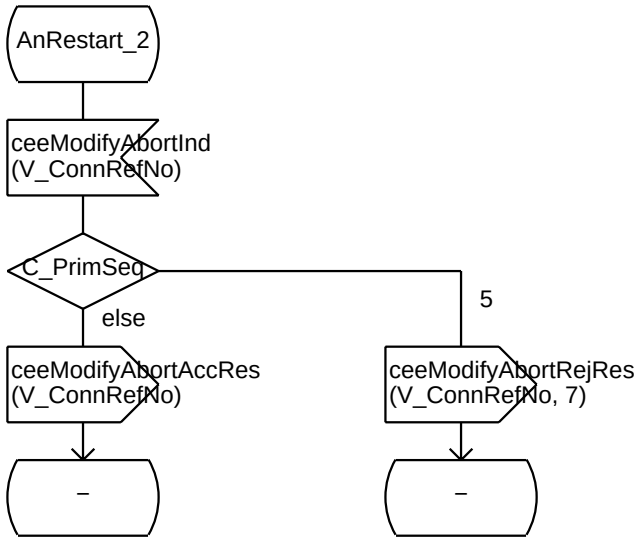
/* AN_Router is responsible for receiving and returning service primitives which normally should be sent to the surrounding system. */



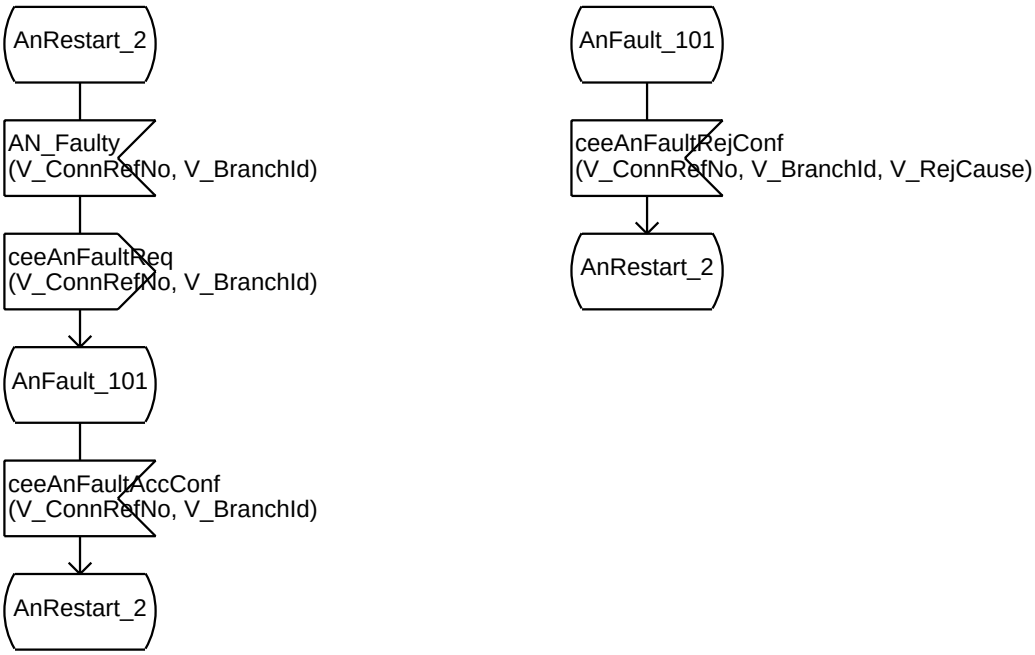


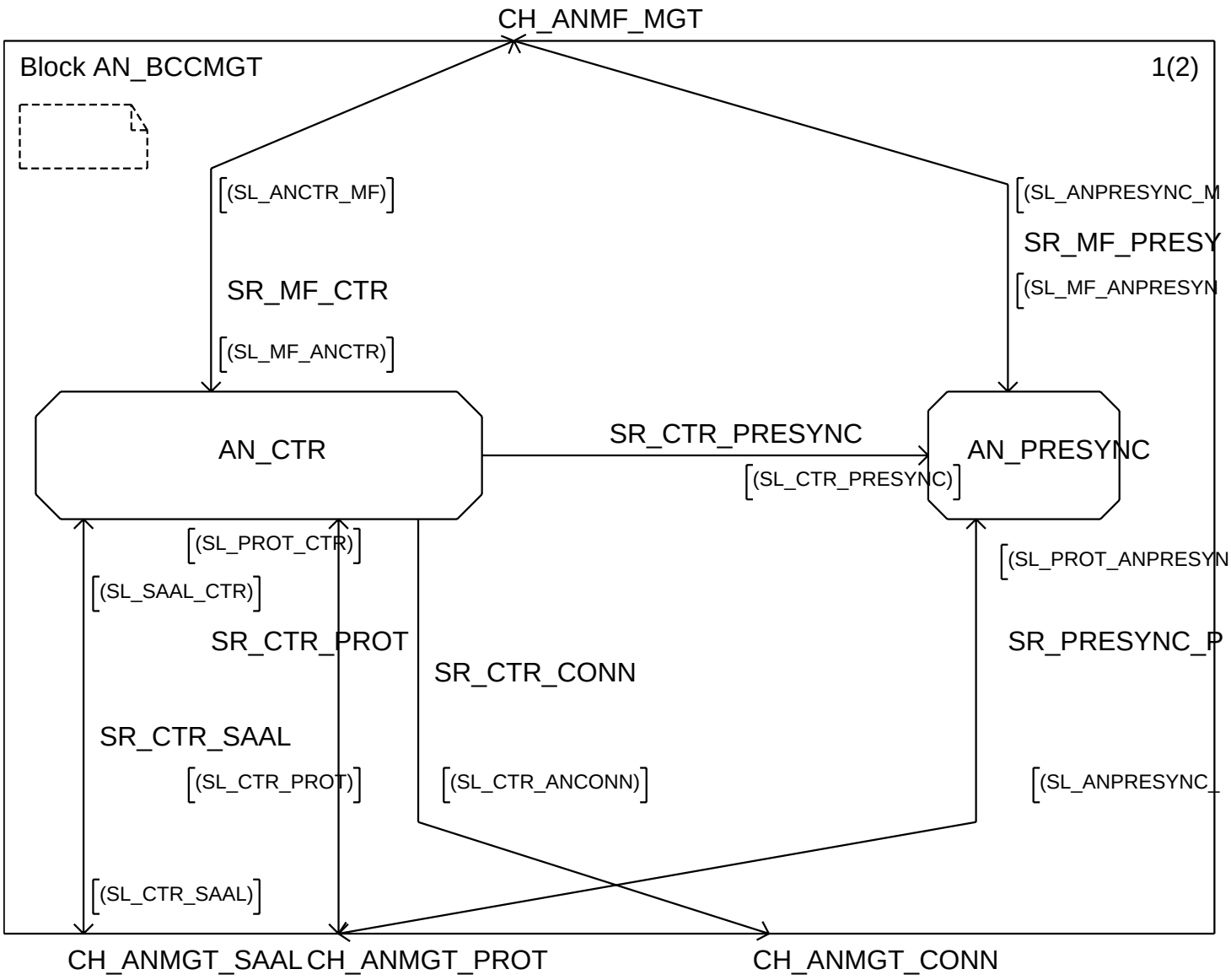
/* AN_Router is responsible for receiving and returning service primitives which normally should be sent to the surrounding system. */





/* AN_Router is responsible for receiving and returning service primitives which normally should be sent to the surrounding system. */



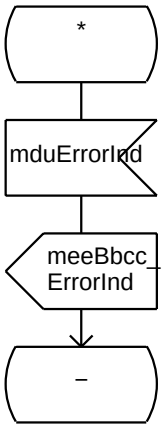

CH_ANMF_MGT
Block AN_BCCMGT
1(2)
(SL_ANCTR_MF)
SR_MF_CTR
(SL_MF_ANCTR)
(SL_ANPRESYNC_M
SR_MF_PRESY
(SL_MF_ANPRESYN
AN_CTR
SR_CTR_PRESYNC
(SL_CTR_PRESYNC)
AN_PRESYNC
(SL_PROT_CTR)
(SL_SAAL_CTR)
SR_CTR_PROT
SR_CTR_CONN
SR_CTR_SAAL
(SL_CTR_PROT)
(SL_CTR_ANCONN)
(SL_CTR_SAAL)
(SL_PROT_ANPRESYN
SR_PRESYNC_P
(SL_ANPRESYNC_
CH_ANMGT_SAAL
CH_ANMGT_PROT
CH_ANMGT_CONN

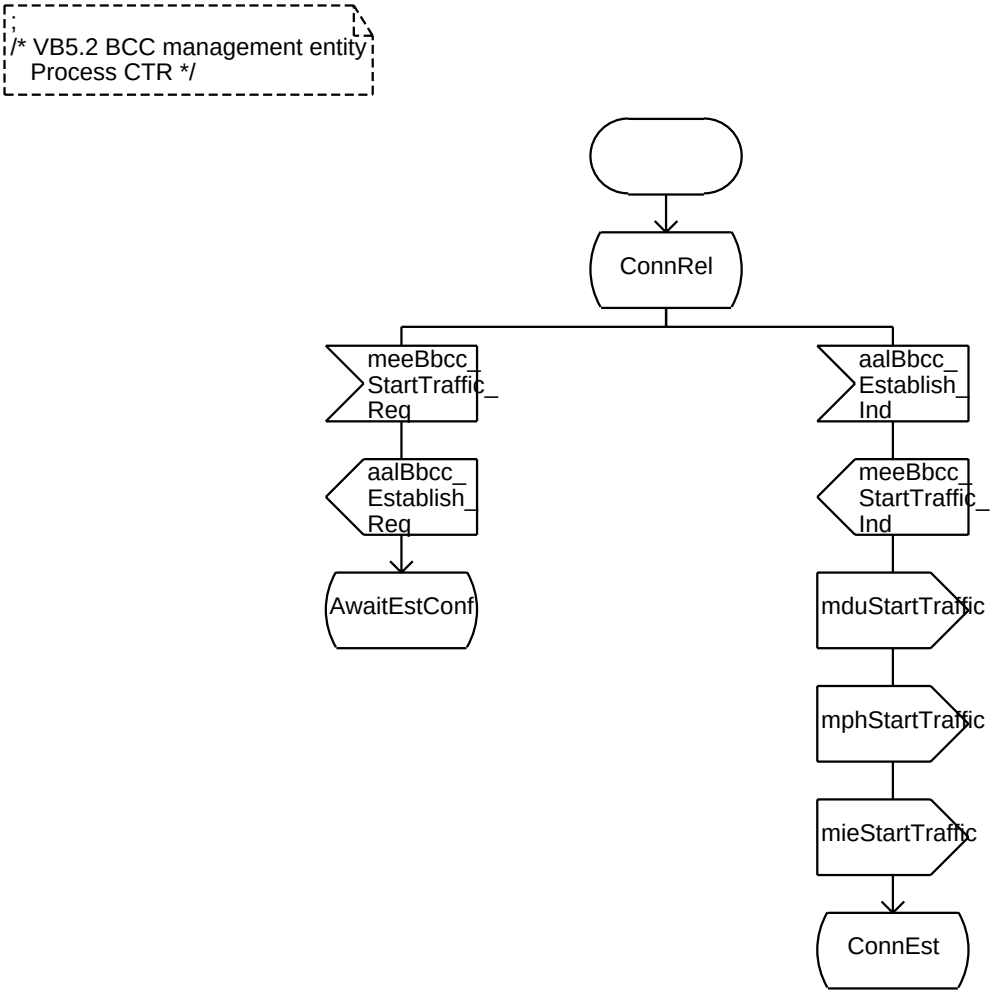
```
/* Declaration of signallists
   internal to block SN_CONNMGT */
```

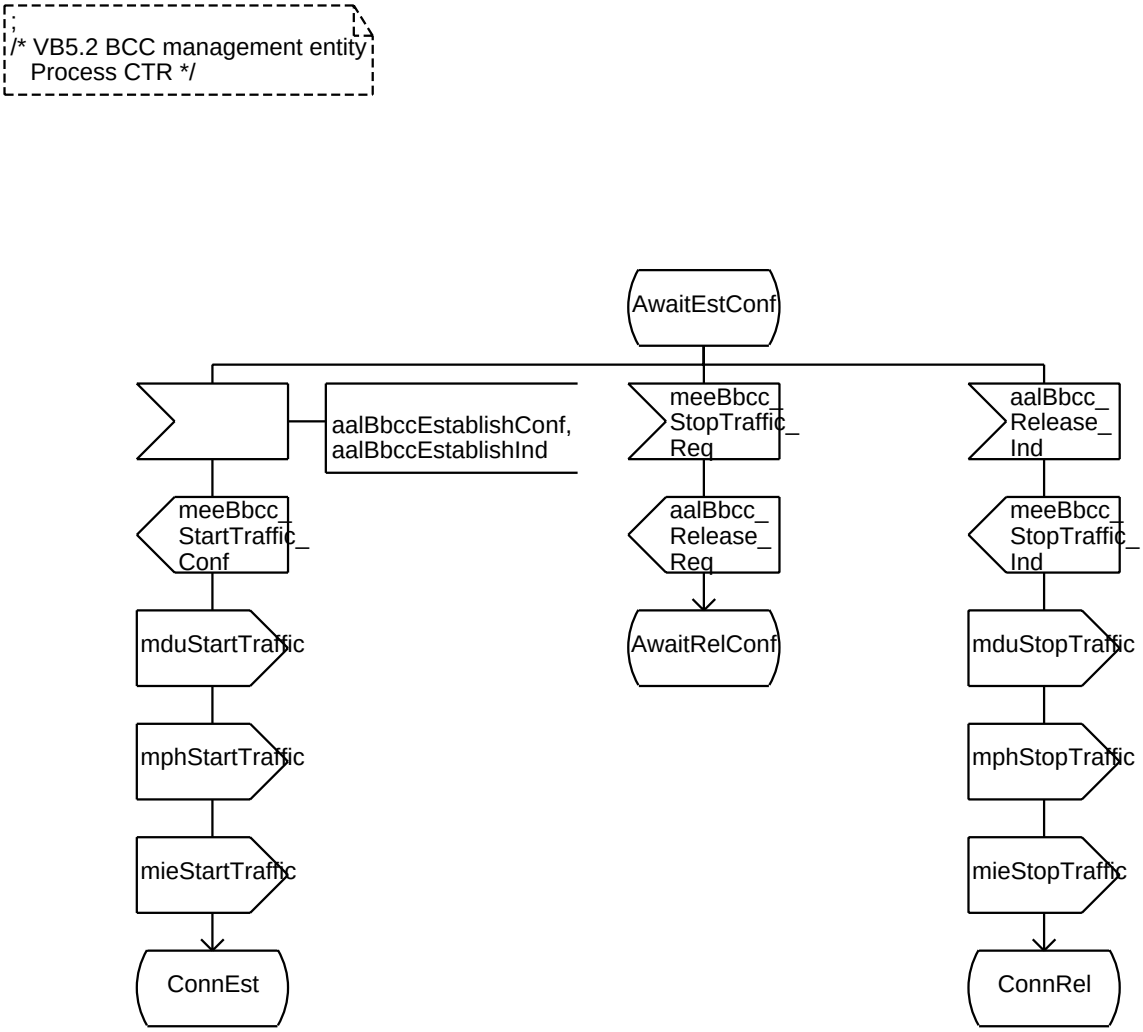
```
signallist SL_CTR_PRESYNC =
mieStartTraffic,
mieStopTraffic;
```

/* VB5.2 BCC management entity
Process CTR */

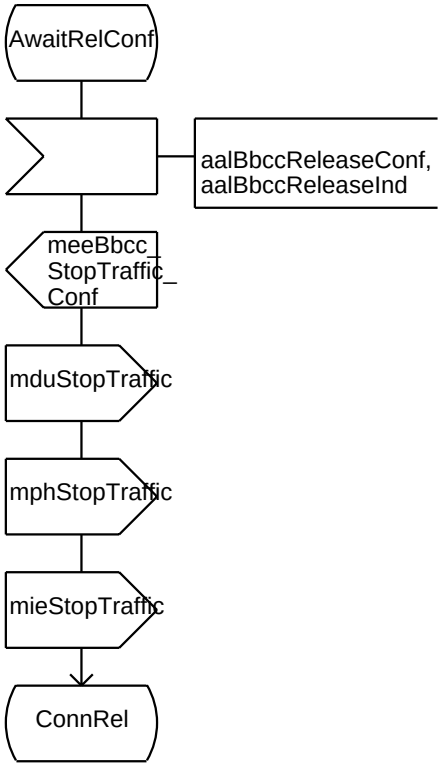
/* State descriptions
ConnRel: initial state, no SAAL connection available.
AwaitEstConf: process is waiting for establish confirmation of the peer entity.
AwaitRelConf: process is waiting for release confirmation of the peer entity.
ConnEst: SAAL in operation. */

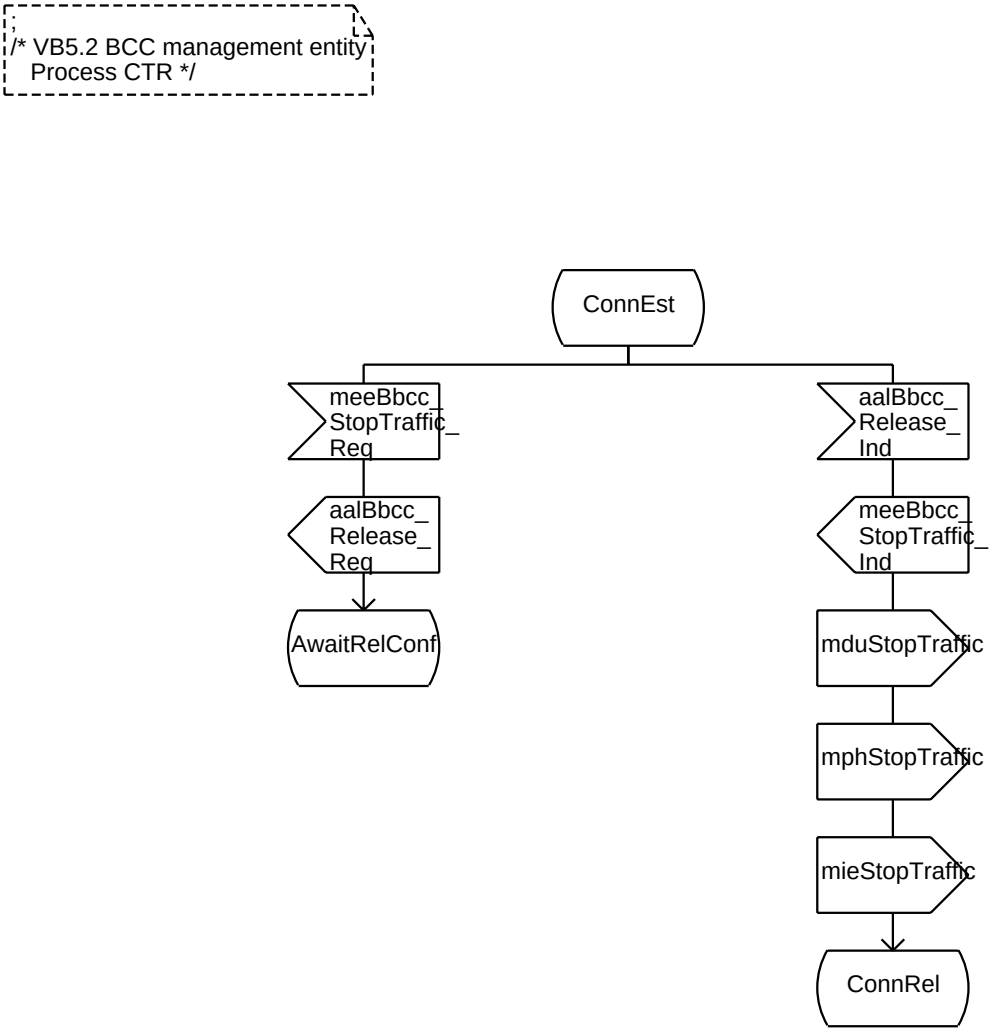



/* VB5.2 BCC management entity
Process CTR */
ConnRel
meeBbcc_StartTraffic_Req
aalBbcc_Establish_Ind
aalBbcc_Establish_Req
meeBbcc_StartTraffic_Ind
AwaitEstConf
mduStartTraffic
mphStartTraffic
mieStartTraffic
ConnEst

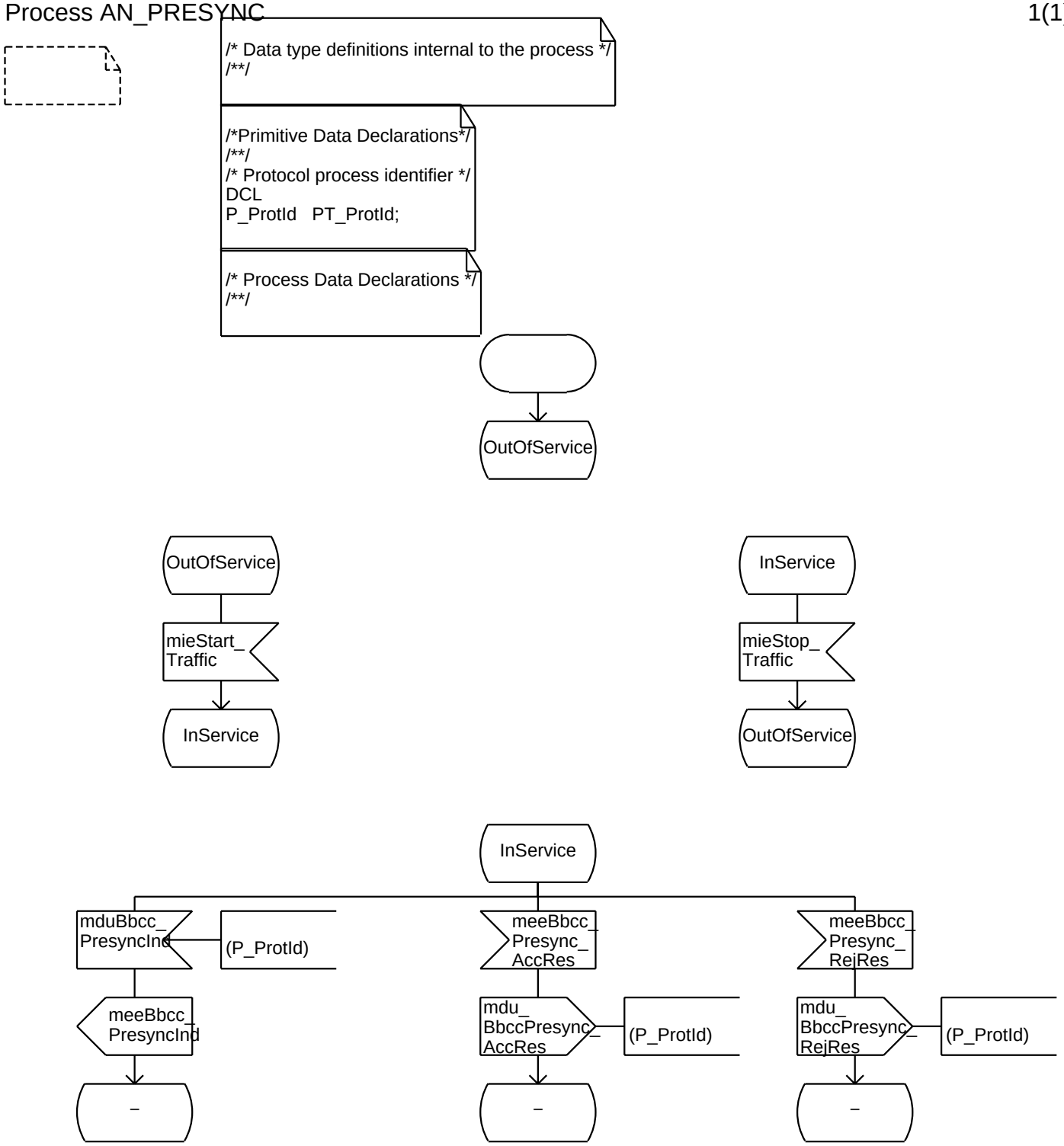

/* VB5.2 BCC management entity
Process CTR */
AwaitEstConf
aalBbccEstablishConf,
aalBbccEstablishInd
meeBbcc_
StopTraffic_
Req
aalBbcc_
Release_
Ind
meeBbcc_
StartTraffic_
Conf
aalBbcc_
Release_
Req
meeBbcc_
StopTraffic_
Ind
mduStartTraffic
AwaitRelConf
mduStopTraffic
mphStartTraffic
mphStopTraffic
mieStartTraffic
mieStopTraffic
ConnEst
ConnRel

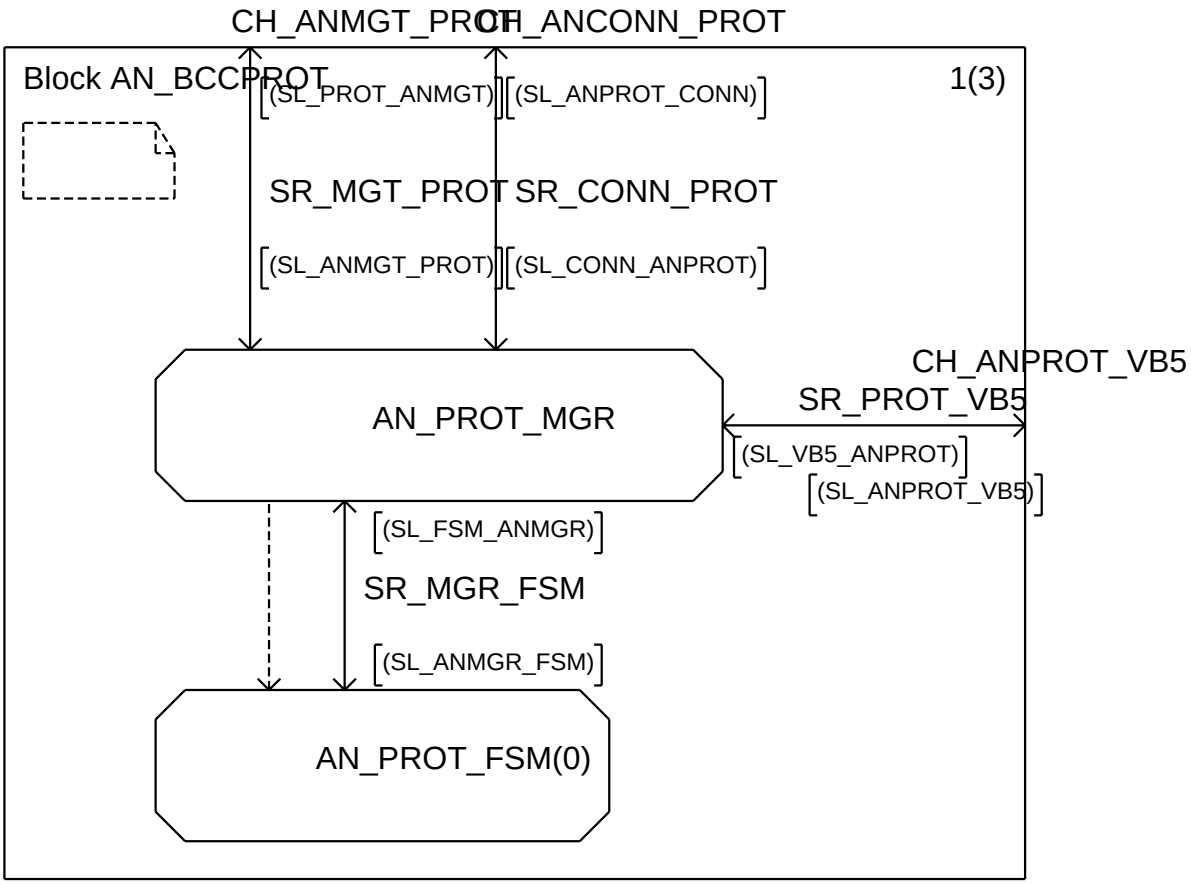
/* VB5.2 BCC management entity
Process CTR */




/* VB5.2 BCC management entity
Process CTR */
ConnEst
meeBbcc_
StopTraffic_
Req
aalBbcc_
Release_
Ind
aalBbcc_
Release_
Req
meeBbcc_
StopTraffic_
Ind
AwaitRelConf
mduStopTraffic
mphStopTraffic
mieStopTraffic
ConnRel

Process AN_PRESYNC




CH_ANMGT_PROT
CH_ANCONN_PROT
Block AN_BCCPROT
1(3)
(SL_PROT_ANMGT)
(SL_ANPROT_CONN)
SR_MGT_PROT
SR_CONN_PROT
(SL_ANMGT_PROT)
(SL_CONN_ANPROT)
CH_ANPROT_VB5
SR_PROT_VB5
AN_PROT_MGR
(SL_VB5_ANPROT)
(SL_ANPROT_VB5)
(SL_FSM_ANMGR)
SR_MGR_FSM
(SL_ANMGR_FSM)
AN_PROT_FSM(0)

/* Declaration of primitives internal
to block AN_BCCPROT */

```
signal
mieAllocInd            (IT_ConnRefNo, IT_BranchId),
mieAllocAccRes         (IT_ConnRefNo, IT_BranchId),
mieAllocRejRes         (IT_ConnRefNo, IT_BranchId, IT_RejCause),
mieAllocCompInd        (IT_ConnRefNo),
mieAllocCompAccRes     (IT_ConnRefNo),
mieAllocCompRejRes     (IT_ConnRefNo, IT_RejCause),
mieBbccResetInd        ,
mieBbccResetAccRes     ,
mieBbccResetRejRes     (IT_RejCause),
mieBbccPresyncInd      ,
mieBbccPresyncAccRes   ,
mieBbccPresyncRejRes   ,
mieDeallocInd          (IT_ConnRefNo),
mieDeallocAccRes       (IT_ConnRefNo),
mieModifyInd           (IT_ConnRefNo),
mieModifyAccRes        (IT_ConnRefNo),
mieModifyRejRes        (IT_ConnRefNo, IT_RejCause),
mieModifyCompInd       (IT_ConnRefNo),
mieModifyCompAccRes    (IT_ConnRefNo),
mieModifyCompRejRes    (IT_ConnRefNo, IT_RejCause),
mieModifyAbortInd      (IT_ConnRefNo),
mieModifyAbortAccRes   (IT_ConnRefNo),
mieModifyAbortRejRes   (IT_ConnRefNo, IT_RejCause),
mieAddBranchInd        (IT_ConnRefNo, IT_BranchId),
mieAddBranchAccRes     (IT_ConnRefNo, IT_BranchId),
mieAddBranchRejRes     (IT_ConnRefNo, IT_BranchId, IT_RejCause),
mieUpdateBranchInd     (IT_ConnRefNo, IT_BranchId),
mieUpdateBranchAccRes  (IT_ConnRefNo, IT_BranchId),
mieUpdateBranchRejRes  (IT_ConnRefNo, IT_BranchId, IT_RejCause),
mieDropBranchInd       (IT_ConnRefNo, IT_BranchId),
mieDropBranchAccRes    (IT_ConnRefNo, IT_BranchId),
mieDropBranchRejRes    (IT_ConnRefNo, IT_BranchId, IT_RejCause),
mieAnFaultReq          (IT_TransId, IT_ConnRefNo, IT_BranchId),
mieAnFaultAccConf      (IT_ConnRefNo, IT_BranchId),
mieAnFaultRejConf      (IT_ConnRefNo, IT_BranchId, IT_RejCause);
```

/* Declaration of signallists
internal to block AN_BCCPROT */

```
signallist SL_FSM_ANMGR =
mieAllocInd,
mieAllocCompInd,
mieBbccResetInd,
mieBbccPresyncInd,
mieDeallocInd,
mieModifyInd,
mieModifyCompInd,
mieModifyAbortInd,
mieAddBranchInd,
mieUpdateBranchInd,
mieDropBranchInd,
bccAllocAcc,
bccAllocRej,
bccAllocCompAcc,
bccAllocCompRej,
bccBbccResetAcc,
bccBbccResetRej,
bccBbccPresyncAcc,
bccBbccPresyncRej,
bccDeallocAcc,
bccModifyAcc,
bccModifyRej,
bccModifyCompAcc,
bccModifyCompRej,
bccModifyAbortAcc,
bccModifyAbortRej,
bccAddBranchAcc,
bccAddBranchRej,
bccUpdateBranchAcc,
bccUpdateBranchRej,
bccDropBranchAcc,
bccDropBranchRej,
bccProtocolError,
bccAnFault,
mieProtocolError,
mieAnFaultAccConf,
mieAnFaultRejConf,
mieErrorInd;
```

```
signallist SL_ANMGR_FSM =
bccAlloc,
bccAllocComp,
bccBbccReset,
bccBbccPresync,
bccDealloc,
bccModify,
bccModifyComp,
bccModifyAbort,
bccAddBranch,
bccUpdateBranch,
bccDropBranch,
bccUnknownMessage,
bccUnknownProtdisc,
bccMsgTooShort,
bccTransIdErr,
bccAnFaultAcc,
mieAllocAccRes,
mieAllocRejRes,
mieAllocCompAccRes,
mieAllocCompRejRes,
mieBbccResetAccRes,
mieBbccResetRejRes,
mieBbccPresyncAccRes,
mieBbccPresyncRejRes,
mieDeallocAccRes,
mieModifyAccRes,
mieModifyRejRes,
mieModifyCompAccRes,
mieModifyCompRejRes,
mieModifyAbortAccRes,
mieModifyAbortRejRes,
mieAddBranchAccRes,
mieAddBranchRejRes,
mieUpdateBranchAccRes,
mieUpdateBranchRejRes,
mieDropBranchAccRes,
mieDropBranchRejRes,
mieAnFaultReq,
mieAbortReq;
```

```
;
/*AN BCC protocol management entity
  controls instantiation of
  protocol FSMs (process AN_PROT_FSM) */
```

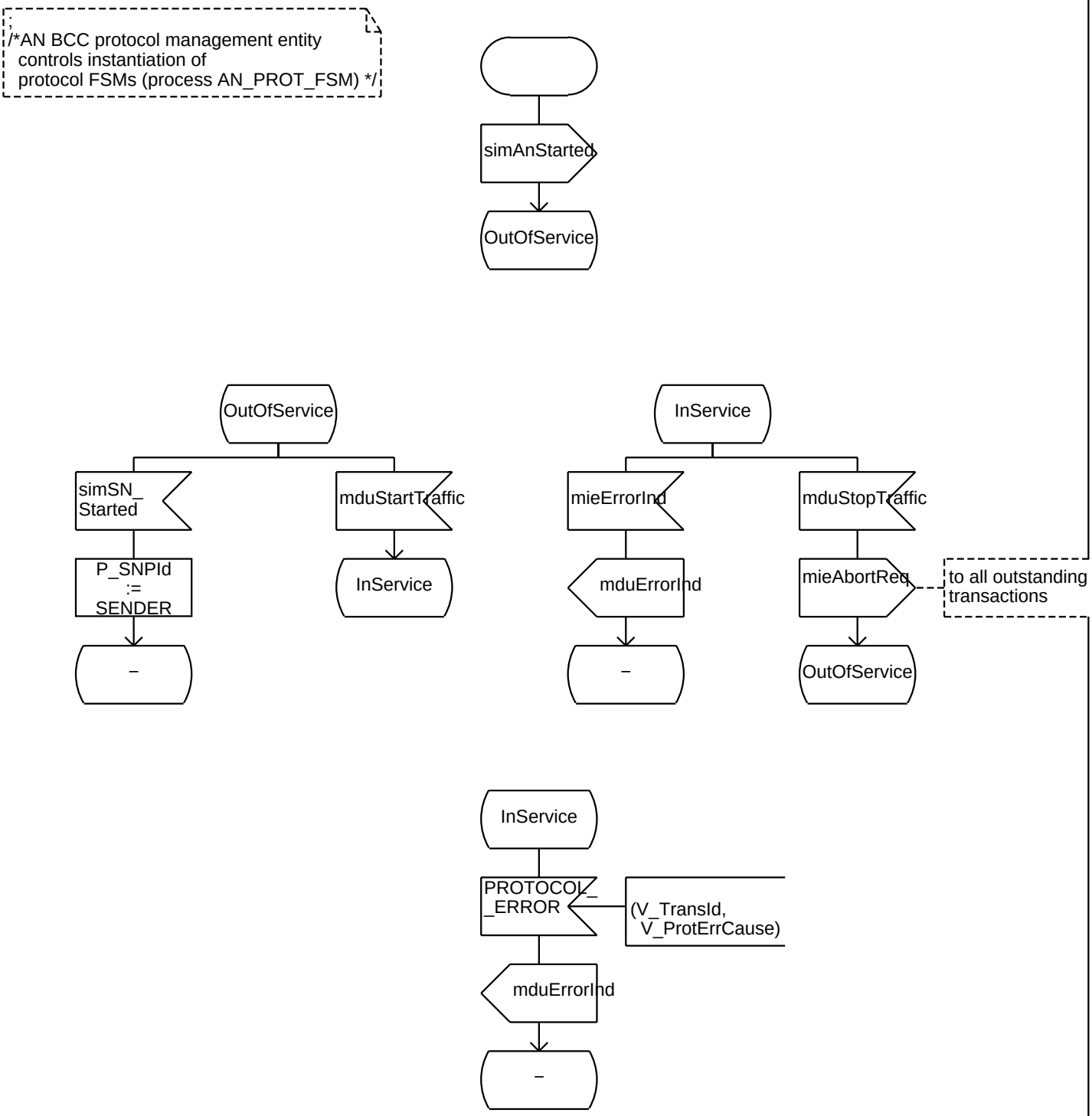
```
/* If a message is repeated by the remote side and an acknowledgement is
already sent, i.e. the corresponding PROT_FSM is deleted, the process
PROT_MGR has to take care that the same acknowledgement as before is
sent to the remote side. A repeated message by the remote side can be
detected by a transaction identifier recently used*/
```

```
/* The generation of the transaction identifier, i.e. the increment by one
   for each invoke message, is for simulation purposes only. In an
   implementation, any other method which is appropriate may be applied */
```

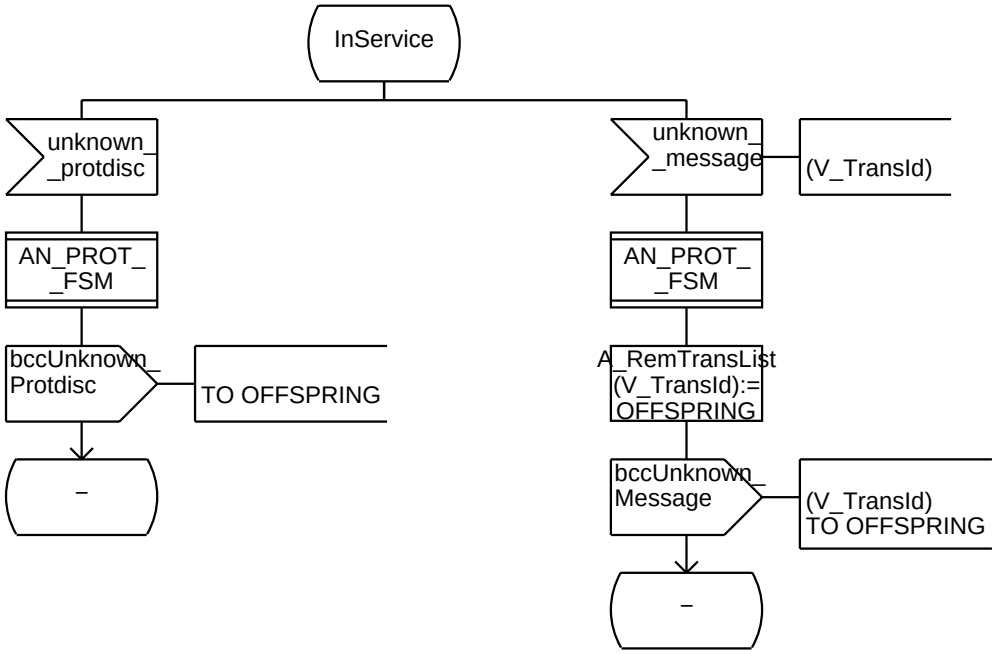
```
/* Data type definitions internal to the process */
/**/
/* This data type is used for translation of the remote transaction identifier to the process identifier (PID)
   of the process handling the corresponding transaction in the AN */
NEWTYPE AT_RemTransList ARRAY
  (IT_TransId, PID)
ENDNEWTYPE AT_RemTransList;
/**/
/* This data type is used for translation of the local transaction identifier to the process identifier (PID)
   of the process handling the corresponding transaction in the AN */
NEWTYPE AT_LocTransList ARRAY
  (IT_TransId, PID)
ENDNEWTYPE AT_LocTransList;
```

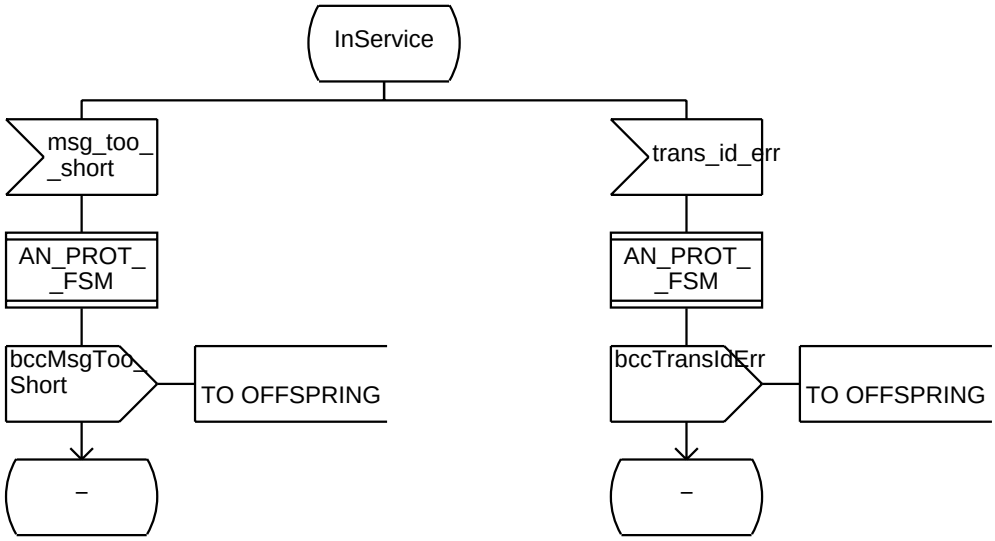
```
/*Primitive Data Declarations*/
/**/
/* Protocol error cause */
DCL
V_ProtErrCause IT_ProtErrCause;
/**/
/* Transaction identifier */
DCL
V_TransId IT_TransId;
/**/
/* Protocol process identifier */
DCL
P_ProtId PT_ProtId;
/**/
/* Connection reference number */
DCL
V_ConnRefNo  IT_ConnRefNo;
/**/
/* Branch identifier */
DCL
V_BranchId  IT_BranchId;
/**/
/* Reject Cause */
DCL
V_RejCause   IT_RejCause;
```

```
/* Process Data Declarations */
/**/
/* List of AN_PROT_FSMs */
DCL
A_RemTransList AT_RemTransList,
A_LocTransList AT_LocTransList;
/**/
/* PID of the receiver process in the SN */
DCL
P_SNPId   PID;
/**/
/* Counter for transaction identifier */
DCL
V_NewTransId IT_TransId := 1;
```

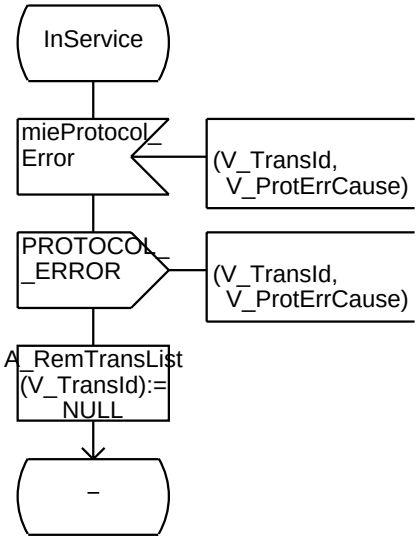

;
/*AN BCC protocol management entity
controls instantiation of
protocol FSMs (process AN_PROT_FSM) */
simAnStarted
OutOfService
OutOfService
simSN_
Started
mduStartTraffic
P_SNPId
:=
SENDER
InService
–
InService
mieErrorInd
mduStopTraffic
mduErrorInd
mieAbortReq
to all outstanding
transactions
–
OutOfService
InService
PROTOCOL_
_ERROR
(V_TransId,
V_ProtErrCause)
mduErrorInd
–

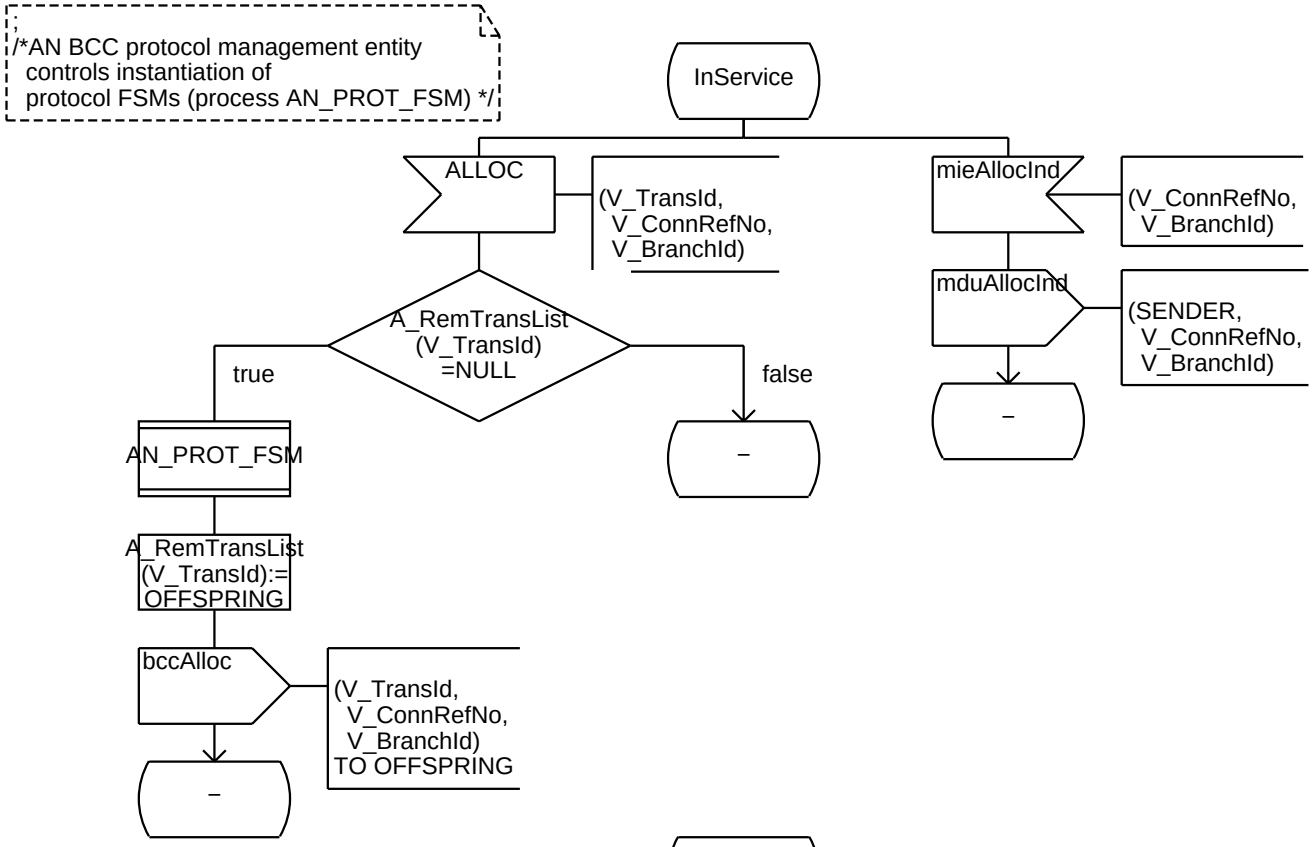
;
/*AN BCC protocol management entity
controls instantiation of
protocol FSMs (process AN_PROT_FSM) */

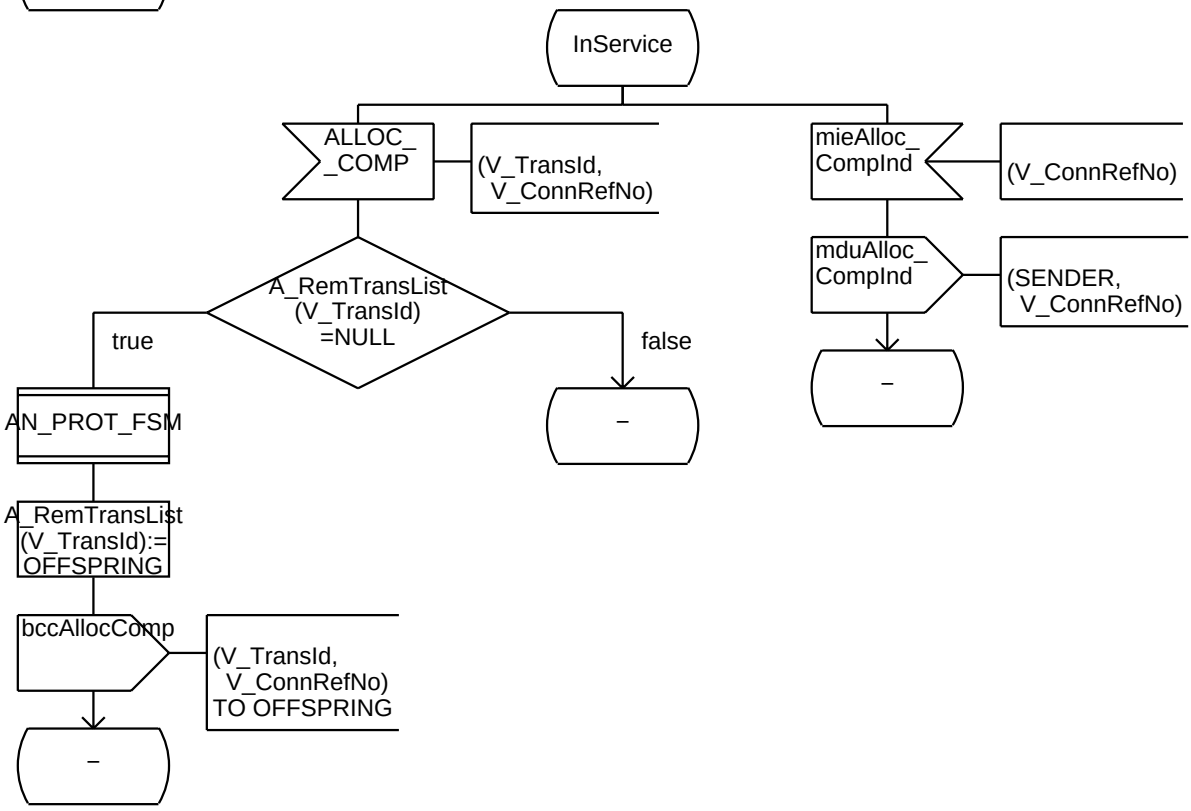


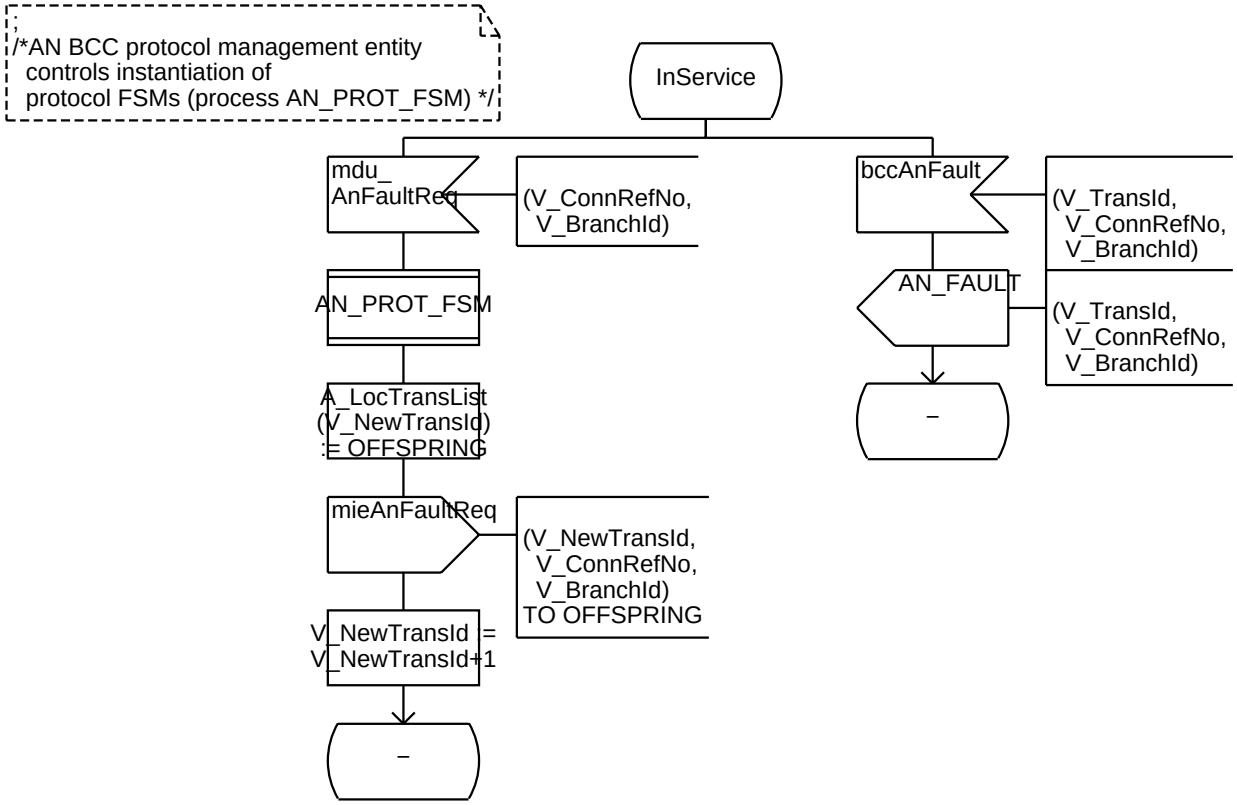


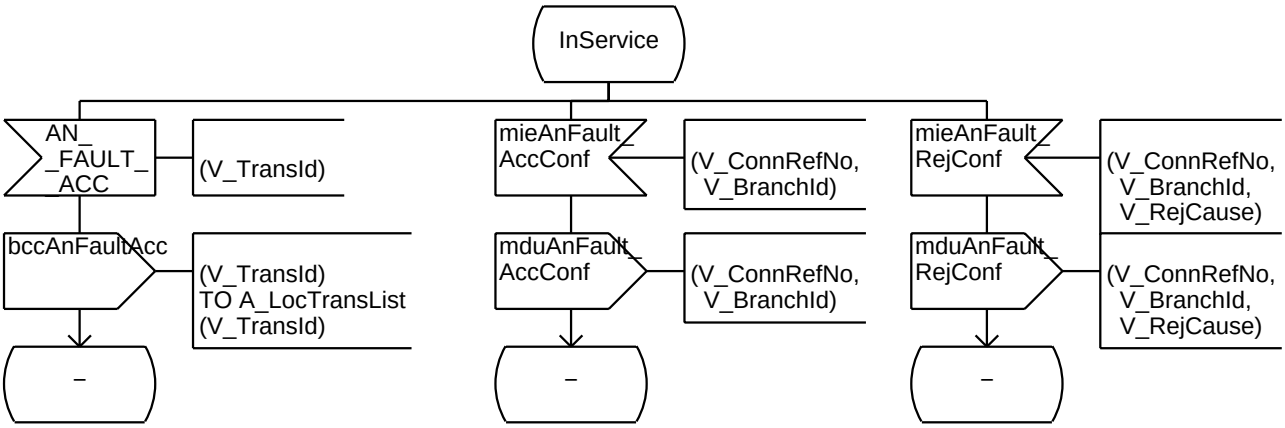
;
/*AN BCC protocol management entity
controls instantiation of
protocol FSMs (process AN_PROT_FSM) */

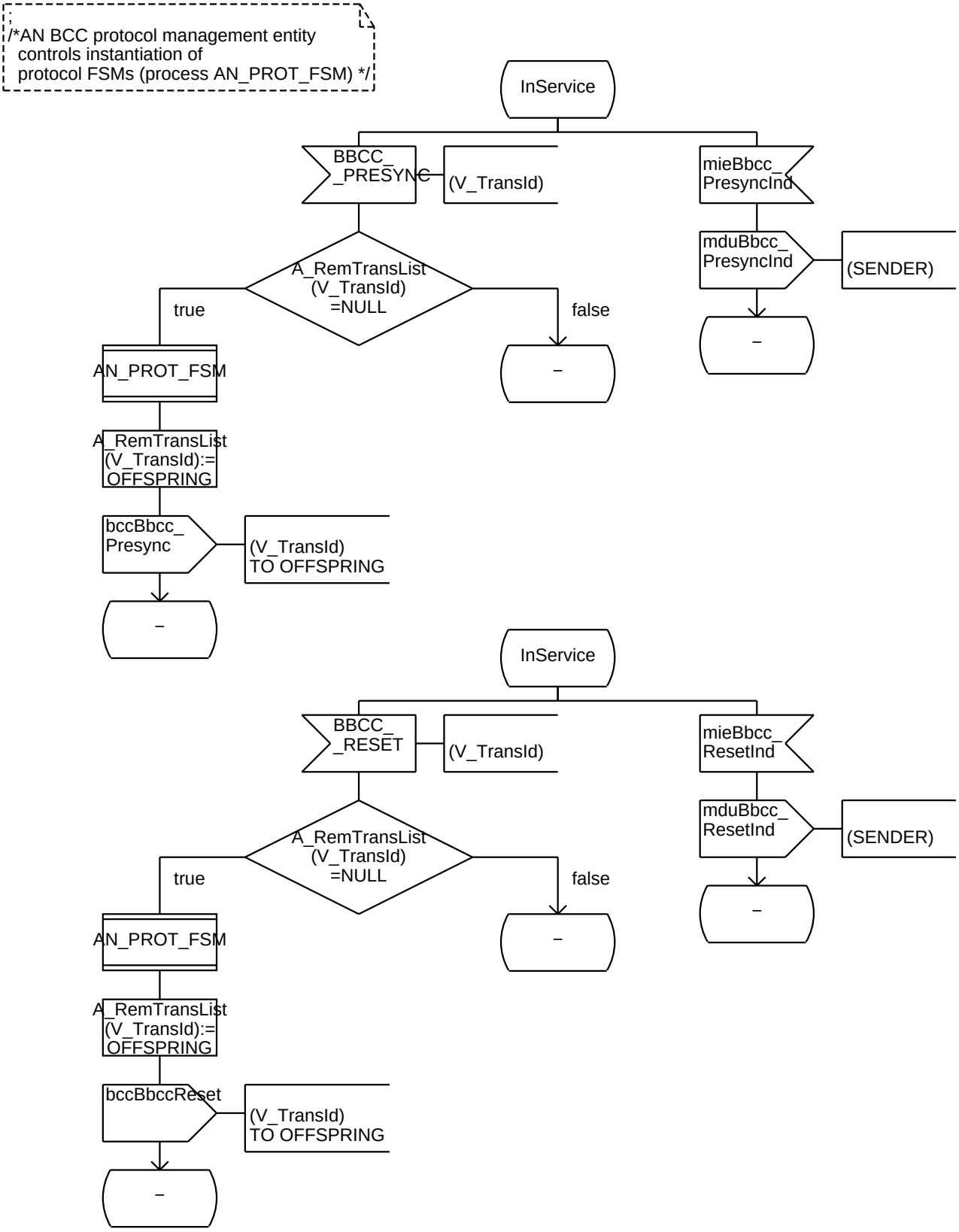


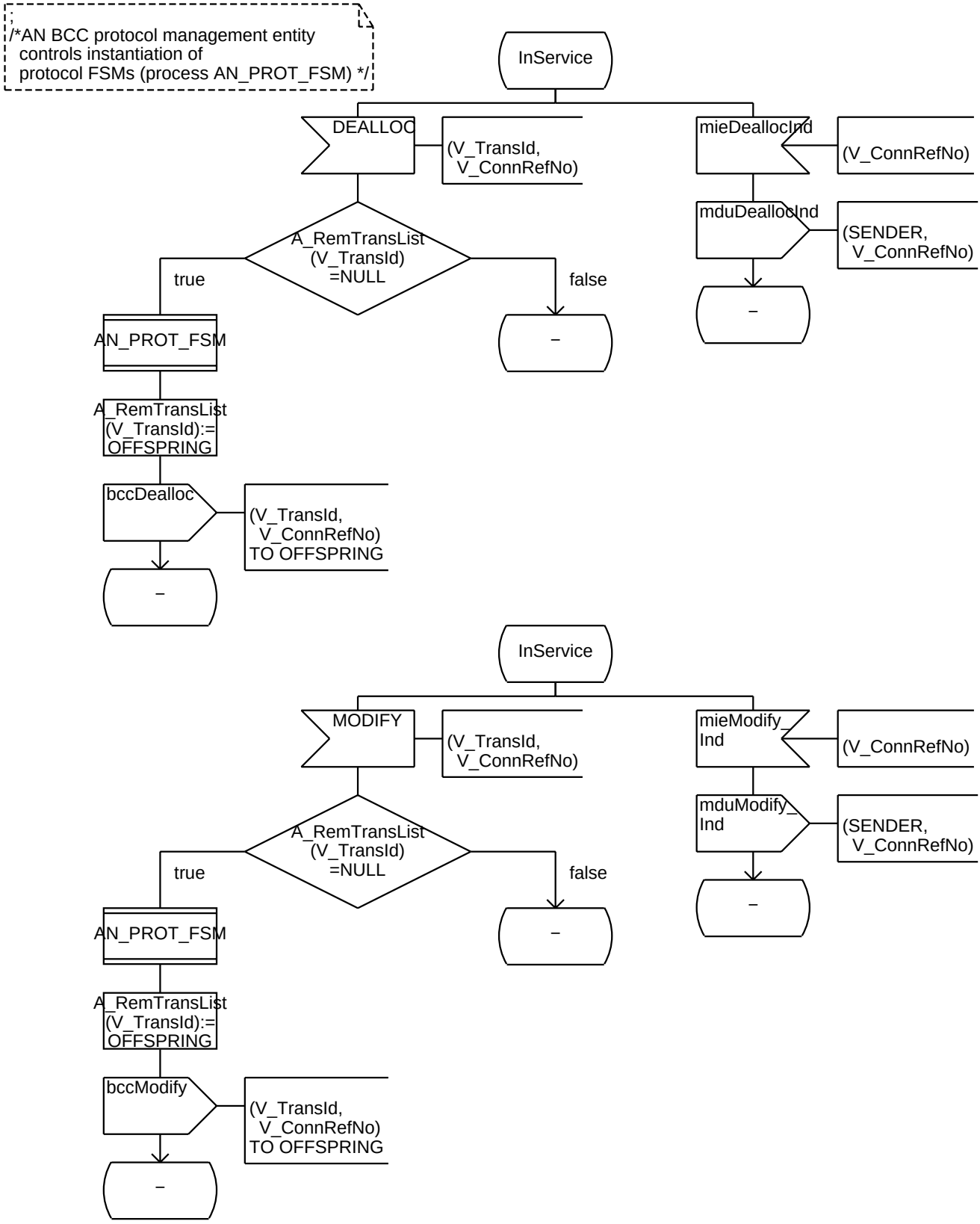

/*AN BCC protocol management entity
controls instantiation of
protocol FSMs (process AN_PROT_FSM) */
InService
ALLOC
(V_TransId,
V_ConnRefNo,
V_BranchId)
mieAllocInd
(V_ConnRefNo,
V_BranchId)
mduAllocInd
(SENDER,
V_ConnRefNo,
V_BranchId)
A_RemTransList
(V_TransId)
=NULL
true
false
AN_PROT_FSM
A_RemTransList
(V_TransId):=
OFFSPRING
bccAlloc
(V_TransId,
V_ConnRefNo,
V_BranchId)
TO OFFSPRING

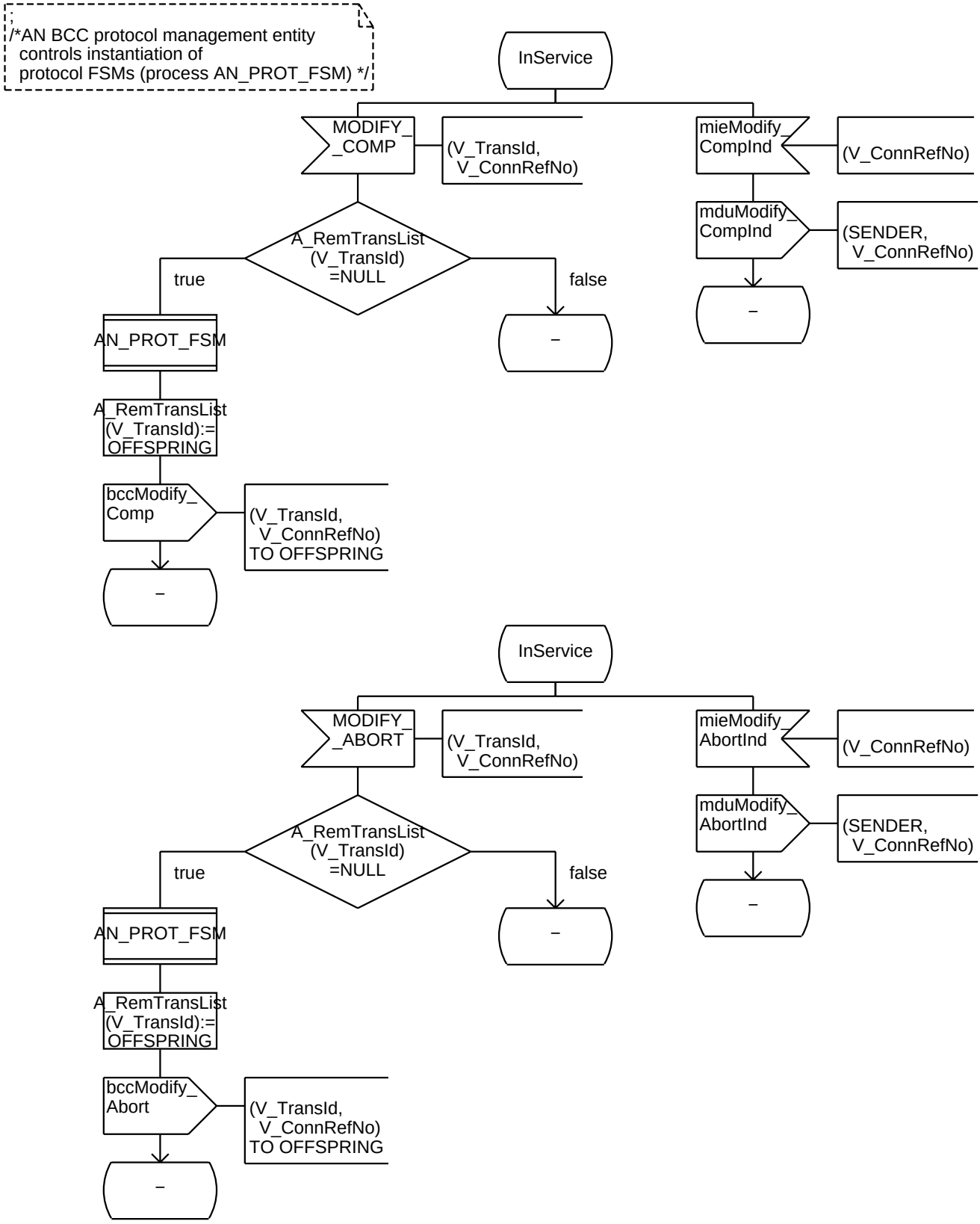

InService
ALLOC_
_COMP
(V_TransId,
V_ConnRefNo)
mieAlloc_
CompInd
(V_ConnRefNo)
mduAlloc_
CompInd
(SENDER,
V_ConnRefNo)
A_RemTransList
(V_TransId)
=NULL
true
false
AN_PROT_FSM
A_RemTransList
(V_TransId):=
OFFSPRING
bccAllocComp
(V_TransId,
V_ConnRefNo)
TO OFFSPRING

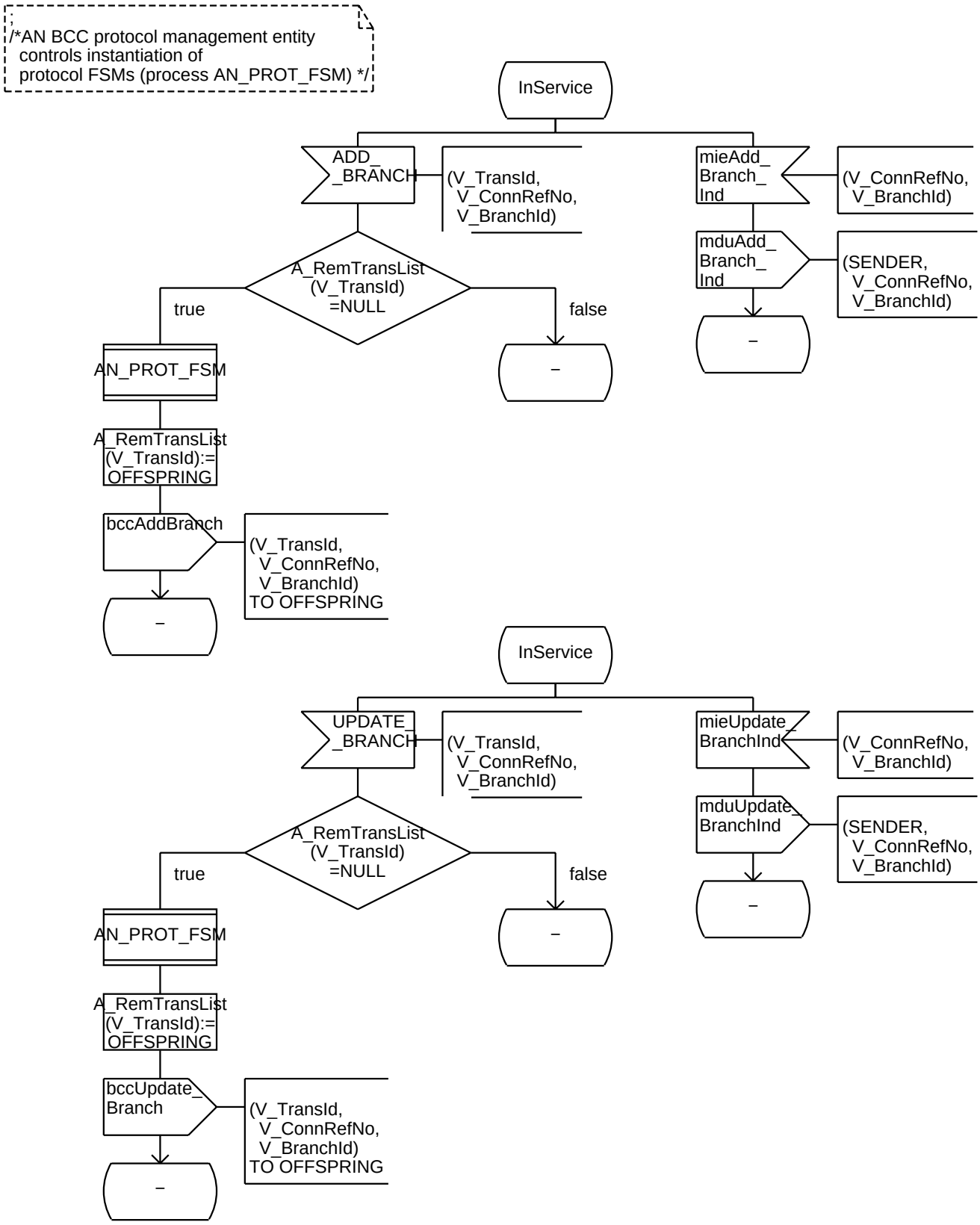

;
/*AN BCC protocol management entity
controls instantiation of
protocol FSMs (process AN_PROT_FSM) */
InService
mdu_
AnFaultReq
(V_ConnRefNo,
V_BranchId)
bccAnFault
(V_TransId,
V_ConnRefNo,
V_BranchId)
AN_PROT_FSM
AN_FAULT
(V_TransId,
V_ConnRefNo,
V_BranchId)
A_LocTransList
(V_NewTransId)
:= OFFSPRING
–
mieAnFaultReq
(V_NewTransId,
V_ConnRefNo,
V_BranchId)
TO OFFSPRING
V_NewTransId :=
V_NewTransId+1
–

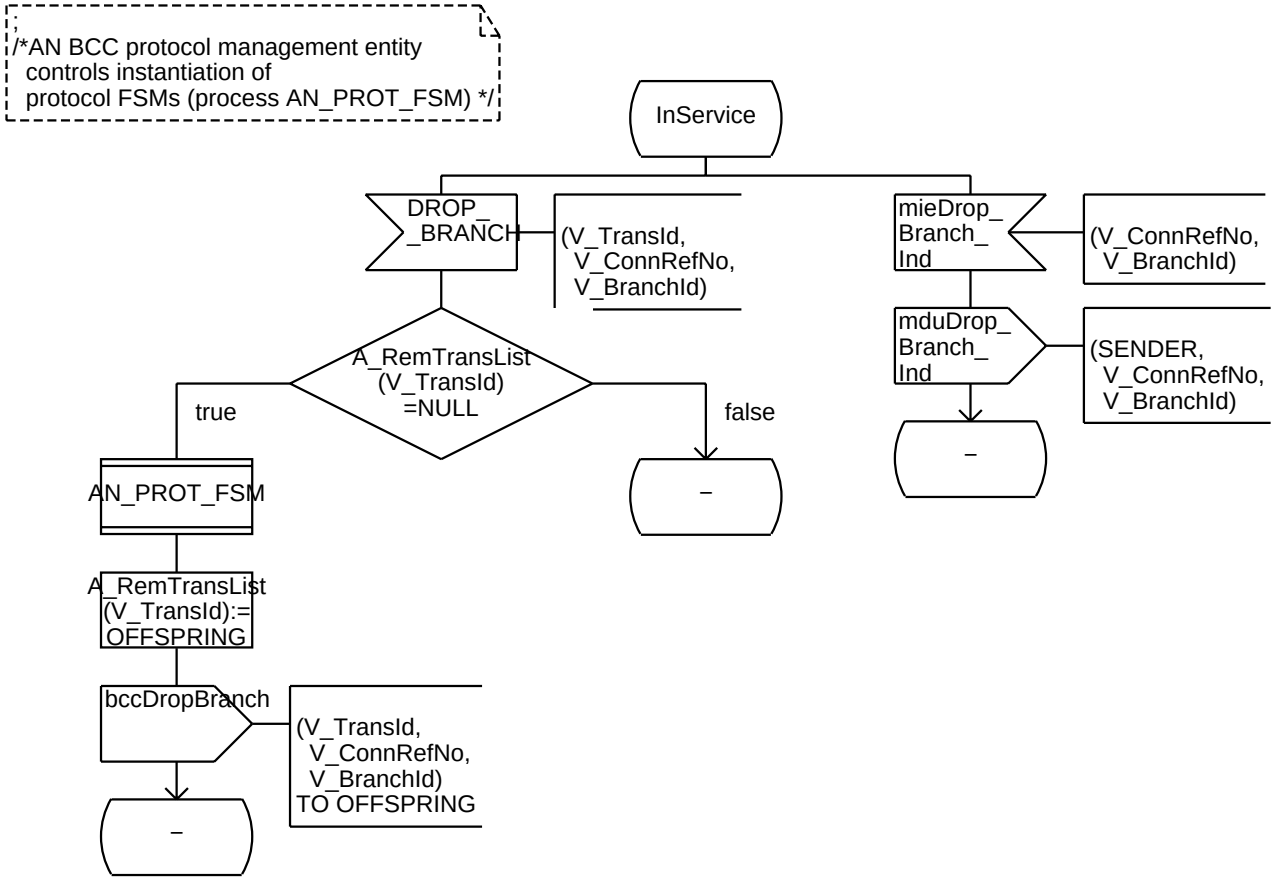

InService
AN_
FAULT
_ACC
(V_TransId)
mieAnFault_
AccConf
(V_ConnRefNo,
V_BranchId)
mieAnFault_
RejConf
(V_ConnRefNo,
V_BranchId,
V_RejCause)
bccAnFaultAcc
(V_TransId)
TO A_LocTransList
(V_TransId)
mduAnFault_
AccConf
(V_ConnRefNo,
V_BranchId)
mduAnFault_
RejConf
(V_ConnRefNo,
V_BranchId,
V_RejCause)
–
–
–


/*AN BCC protocol management entity
controls instantiation of
protocol FSMs (process AN_PROT_FSM) */
InService
BBCC_
_PRESYNC
(V_TransId)
mieBbcc_
PresyncInd
mduBbcc_
PresyncInd
(SENDER)
A_RemTransList
(V_TransId)
=NULL
true
false
AN_PROT_FSM
A_RemTransList
(V_TransId):=
OFFSPRING
bccBbcc_
Presync
(V_TransId)
TO OFFSPRING
InService
BBCC_
_RESET
(V_TransId)
mieBbcc_
ResetInd
mduBbcc_
ResetInd
(SENDER)
A_RemTransList
(V_TransId)
=NULL
true
false
AN_PROT_FSM
A_RemTransList
(V_TransId):=
OFFSPRING
bccBbccReset
(V_TransId)
TO OFFSPRING

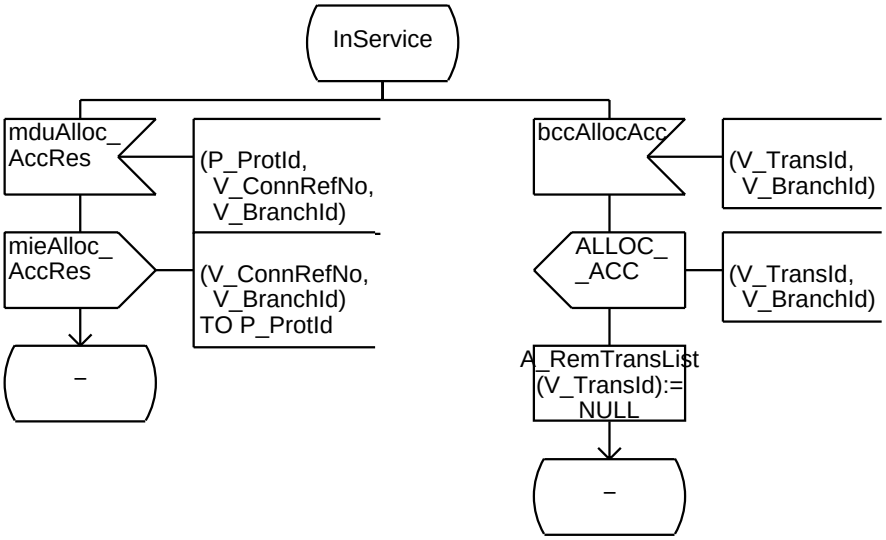

/*AN BCC protocol management entity
controls instantiation of
protocol FSMs (process AN_PROT_FSM) */
InService
DEALLOC
(V_TransId,
V_ConnRefNo)
mieDeallocInd
(V_ConnRefNo)
mduDeallocInd
(SENDER,
V_ConnRefNo)
A_RemTransList
(V_TransId)
=NULL
true
false
AN_PROT_FSM
A_RemTransList
(V_TransId):=
OFFSPRING
bccDealloc
(V_TransId,
V_ConnRefNo)
TO OFFSPRING
InService
MODIFY
(V_TransId,
V_ConnRefNo)
mieModify_
Ind
(V_ConnRefNo)
mduModify_
Ind
(SENDER,
V_ConnRefNo)
A_RemTransList
(V_TransId)
=NULL
true
false
AN_PROT_FSM
A_RemTransList
(V_TransId):=
OFFSPRING
bccModify
(V_TransId,
V_ConnRefNo)
TO OFFSPRING

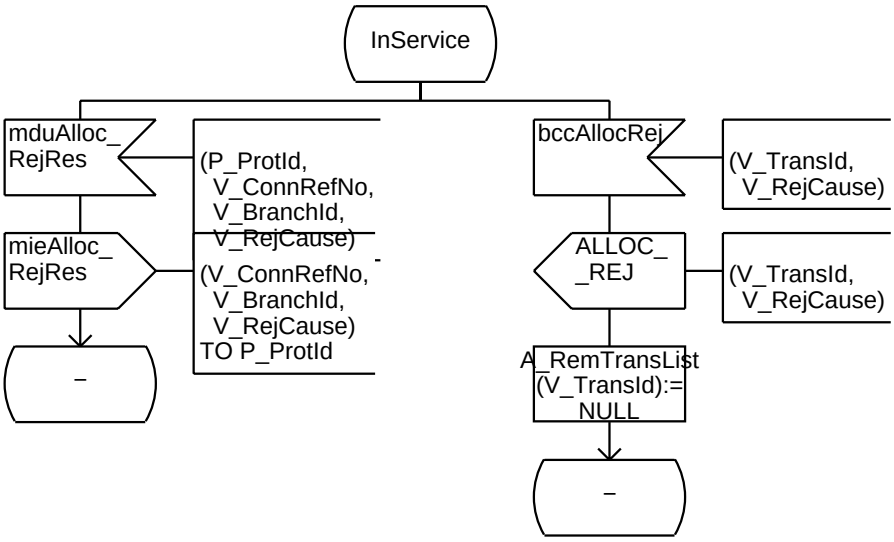

/*AN BCC protocol management entity
controls instantiation of
protocol FSMs (process AN_PROT_FSM) */
InService
MODIFY_
_COMP
(V_TransId,
V_ConnRefNo)
mieModify_
CompInd
(V_ConnRefNo)
mduModify_
CompInd
(SENDER,
V_ConnRefNo)
A_RemTransList
(V_TransId)
=NULL
true
false
AN_PROT_FSM
A_RemTransList
(V_TransId):=
OFFSPRING
bccModify_
Comp
(V_TransId,
V_ConnRefNo)
TO OFFSPRING
InService
MODIFY_
_ABORT
(V_TransId,
V_ConnRefNo)
mieModify_
AbortInd
(V_ConnRefNo)
mduModify_
AbortInd
(SENDER,
V_ConnRefNo)
A_RemTransList
(V_TransId)
=NULL
true
false
AN_PROT_FSM
A_RemTransList
(V_TransId):=
OFFSPRING
bccModify_
Abort
(V_TransId,
V_ConnRefNo)
TO OFFSPRING


/*AN BCC protocol management entity
controls instantiation of
protocol FSMs (process AN_PROT_FSM) */
InService
ADD_
_BRANCH
(V_TransId,
V_ConnRefNo,
V_BranchId)
mieAdd_
Branch_
Ind
(V_ConnRefNo,
V_BranchId)
mduAdd_
Branch_
Ind
(SENDER,
V_ConnRefNo,
V_BranchId)
A_RemTransList
(V_TransId)
=NULL
true
false
AN_PROT_FSM
A_RemTransList
(V_TransId):=
OFFSPRING
bccAddBranch
(V_TransId,
V_ConnRefNo,
V_BranchId)
TO OFFSPRING
InService
UPDATE_
_BRANCH
(V_TransId,
V_ConnRefNo,
V_BranchId)
mieUpdate_
BranchInd
(V_ConnRefNo,
V_BranchId)
mduUpdate_
BranchInd
(SENDER,
V_ConnRefNo,
V_BranchId)
A_RemTransList
(V_TransId)
=NULL
true
false
AN_PROT_FSM
A_RemTransList
(V_TransId):=
OFFSPRING
bccUpdate_
Branch
(V_TransId,
V_ConnRefNo,
V_BranchId)
TO OFFSPRING

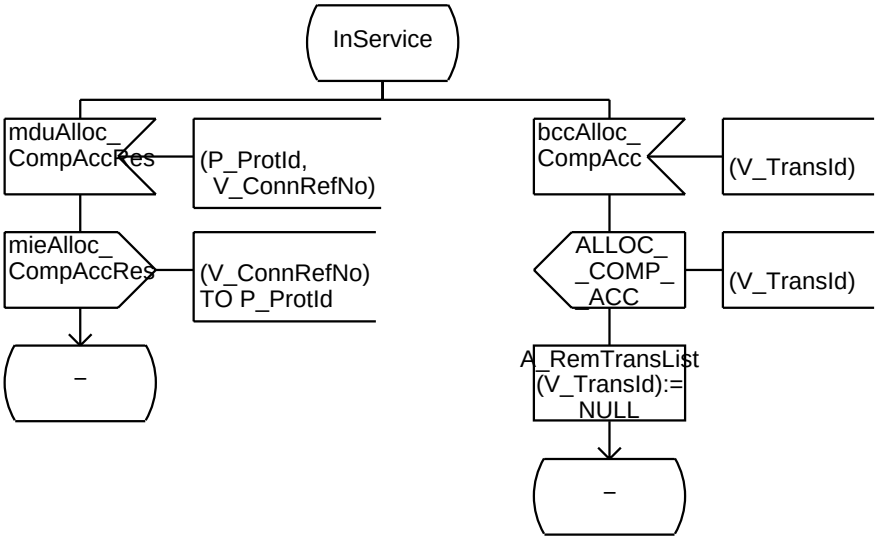

;
/*AN BCC protocol management entity
controls instantiation of
protocol FSMs (process AN_PROT_FSM) */
InService
DROP_
_BRANCH
(V_TransId,
V_ConnRefNo,
V_BranchId)
mieDrop_
Branch_
Ind
(V_ConnRefNo,
V_BranchId)
mduDrop_
Branch_
Ind
(SENDER,
V_ConnRefNo,
V_BranchId)
A_RemTransList
(V_TransId)
=NULL
true
false
AN_PROT_FSM
A_RemTransList
(V_TransId):=
OFFSPRING
bccDropBranch
(V_TransId,
V_ConnRefNo,
V_BranchId)
TO OFFSPRING

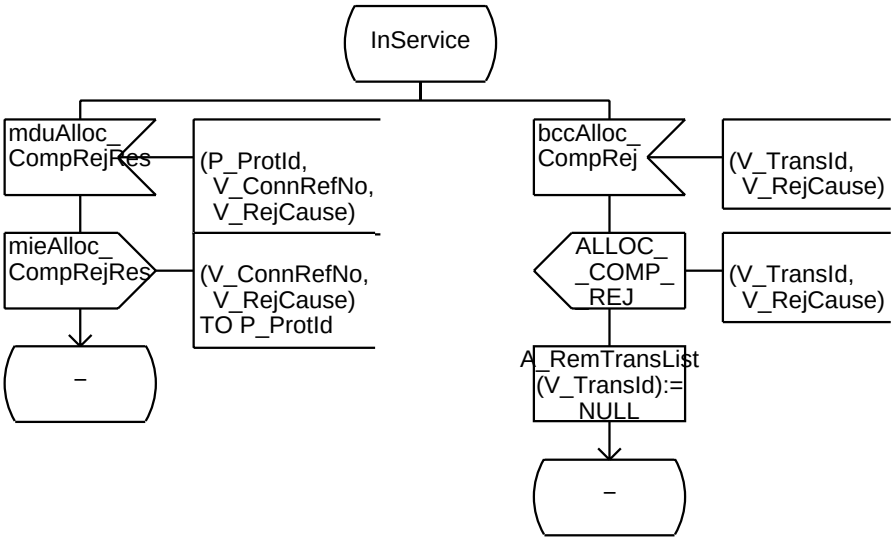
;
/*AN BCC protocol management entity
controls instantiation of
protocol FSMs (process AN_PROT_FSM) */



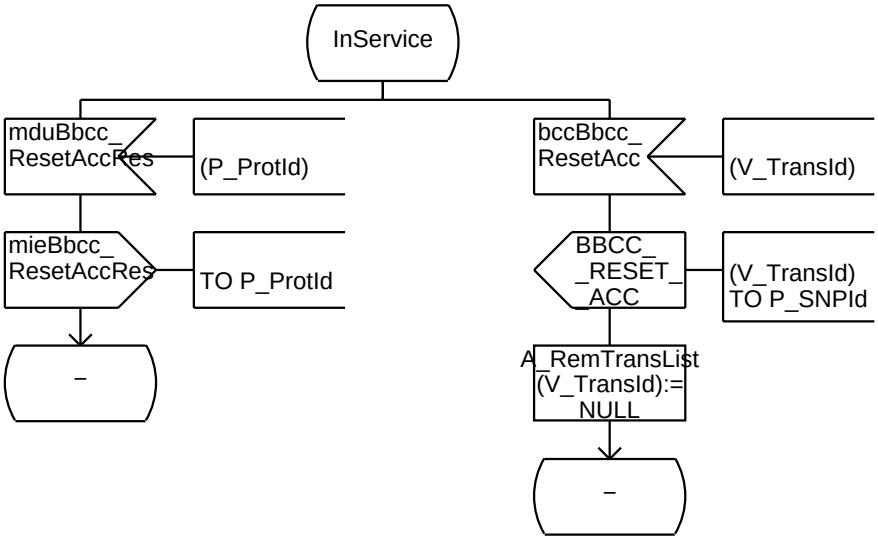


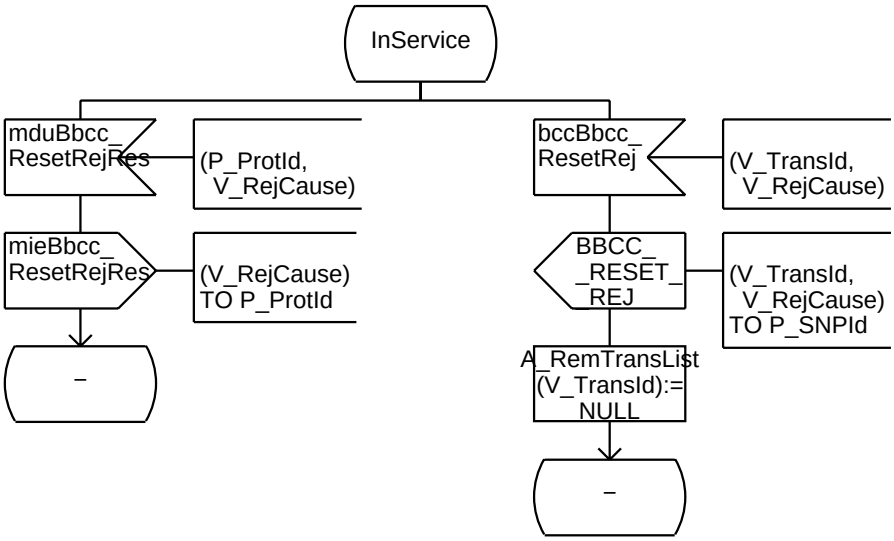
;
/*AN BCC protocol management entity
controls instantiation of
protocol FSMs (process AN_PROT_FSM) */



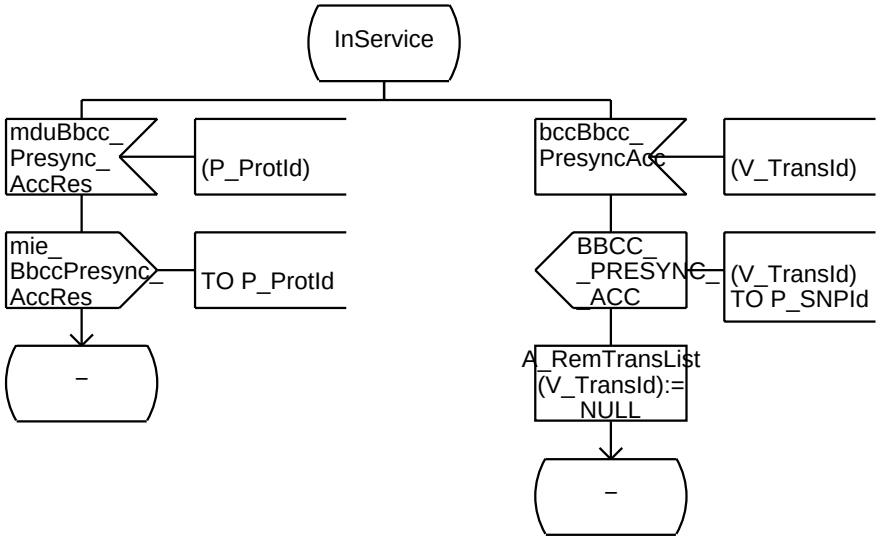


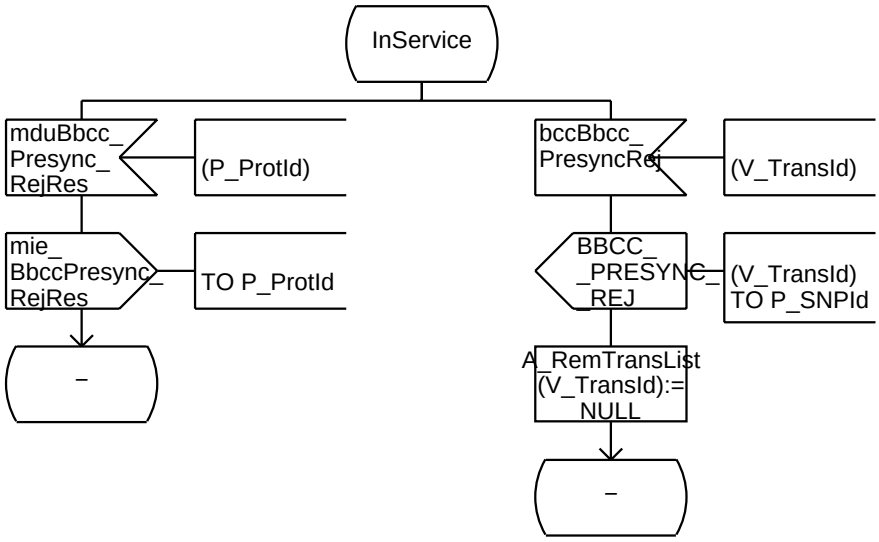
;
/*AN BCC protocol management entity
controls instantiation of
protocol FSMs (process AN_PROT_FSM) */



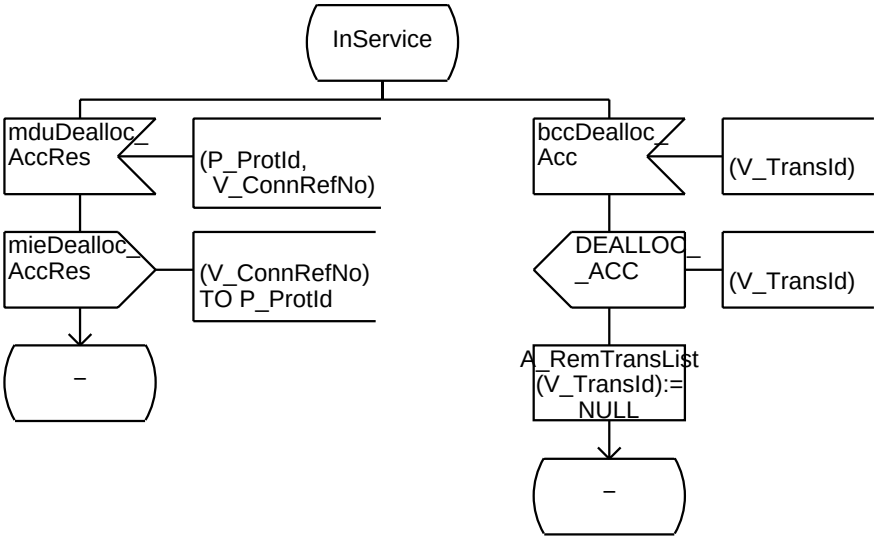


;
/*AN BCC protocol management entity
controls instantiation of
protocol FSMs (process AN_PROT_FSM) */

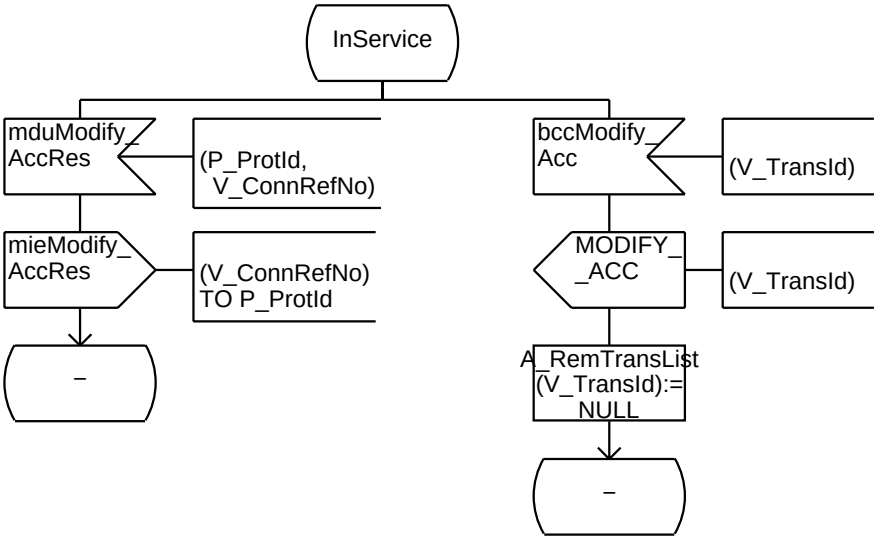


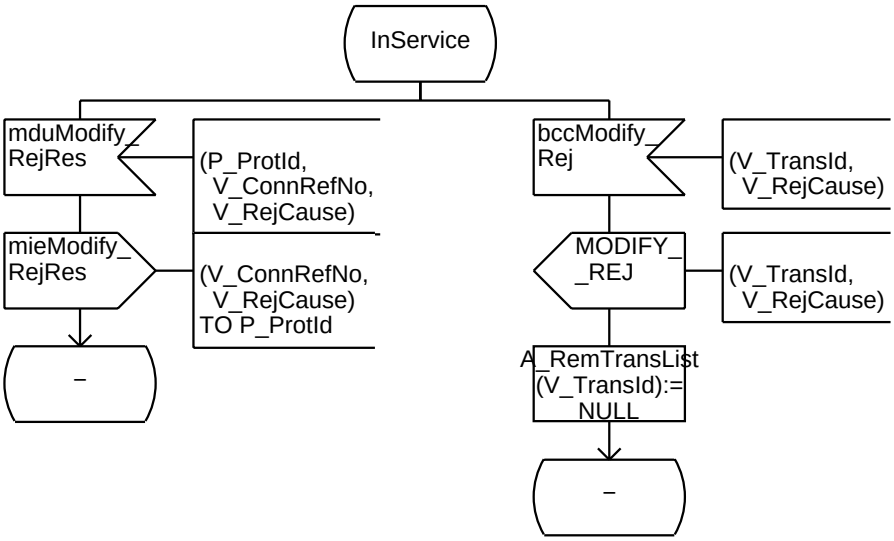


;
/*AN BCC protocol management entity
controls instantiation of
protocol FSMs (process AN_PROT_FSM) */

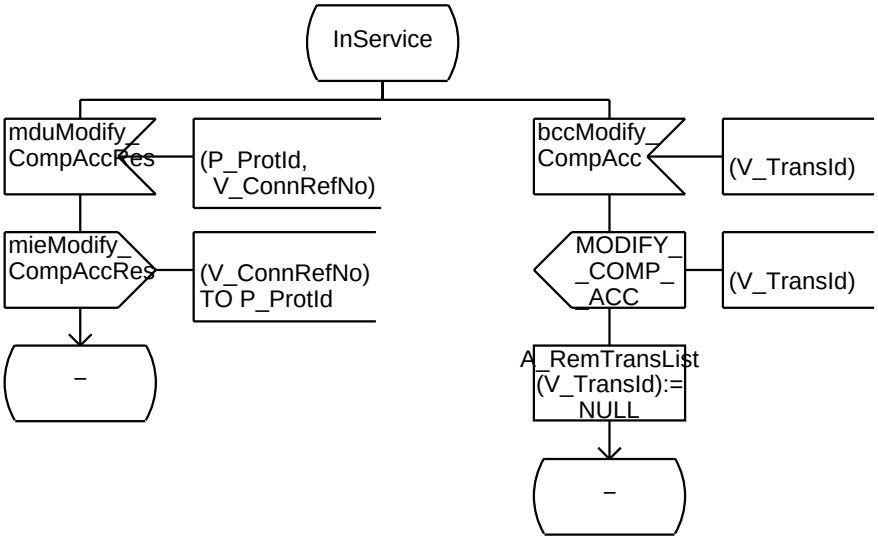


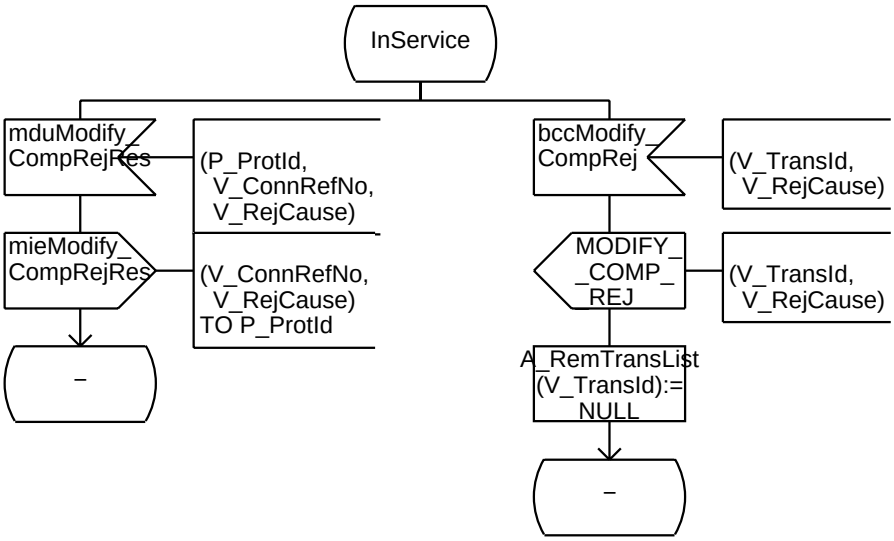
;
/*AN BCC protocol management entity
controls instantiation of
protocol FSMs (process AN_PROT_FSM) */



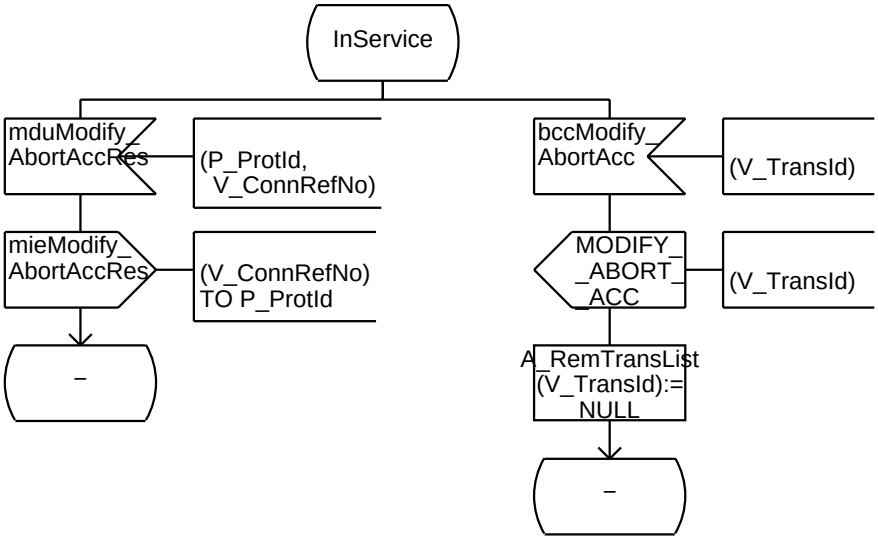


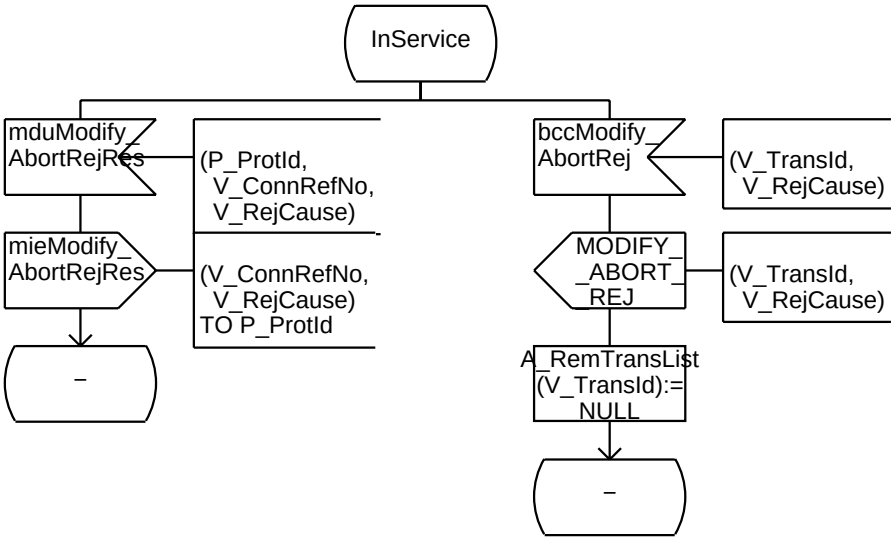
;
/*AN BCC protocol management entity
controls instantiation of
protocol FSMs (process AN_PROT_FSM) */



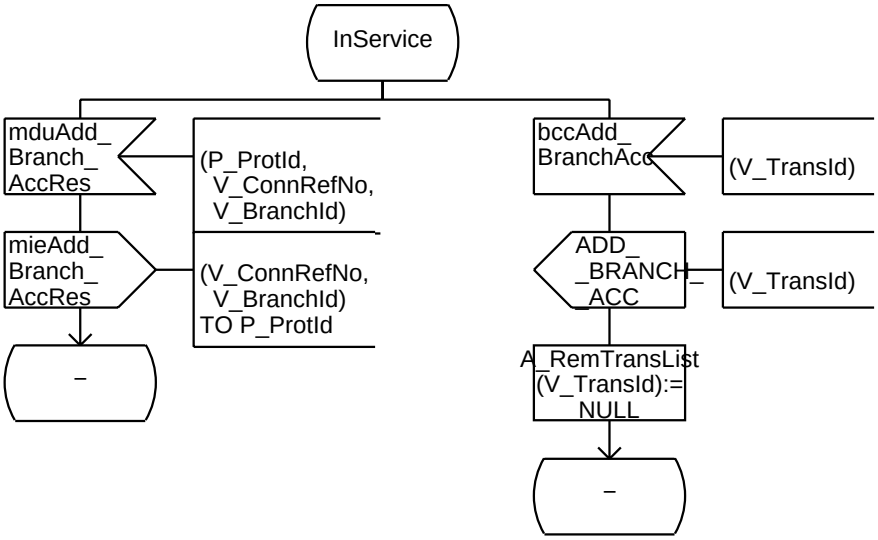


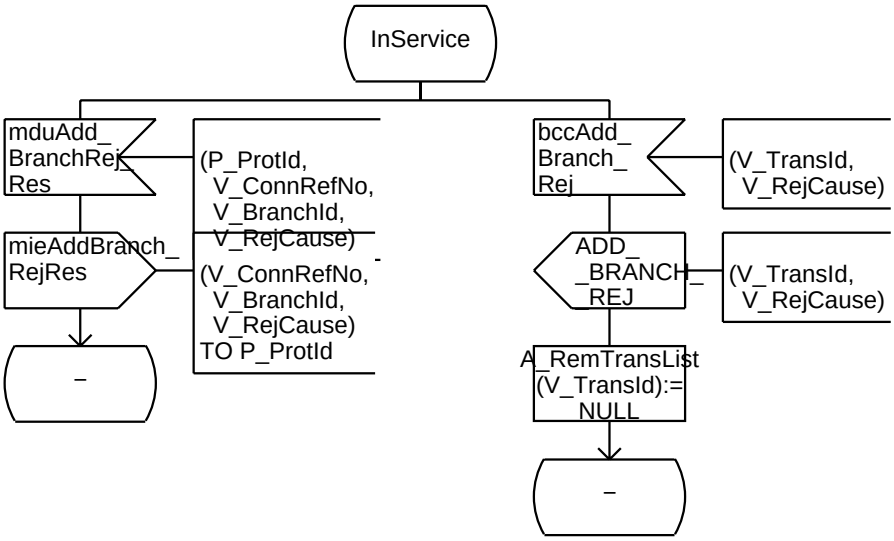
;
/*AN BCC protocol management entity
controls instantiation of
protocol FSMs (process AN_PROT_FSM) */



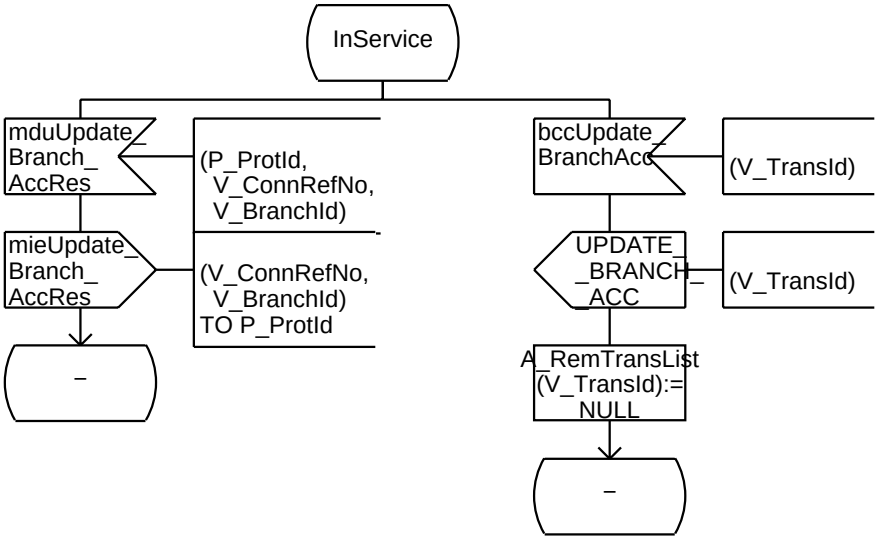


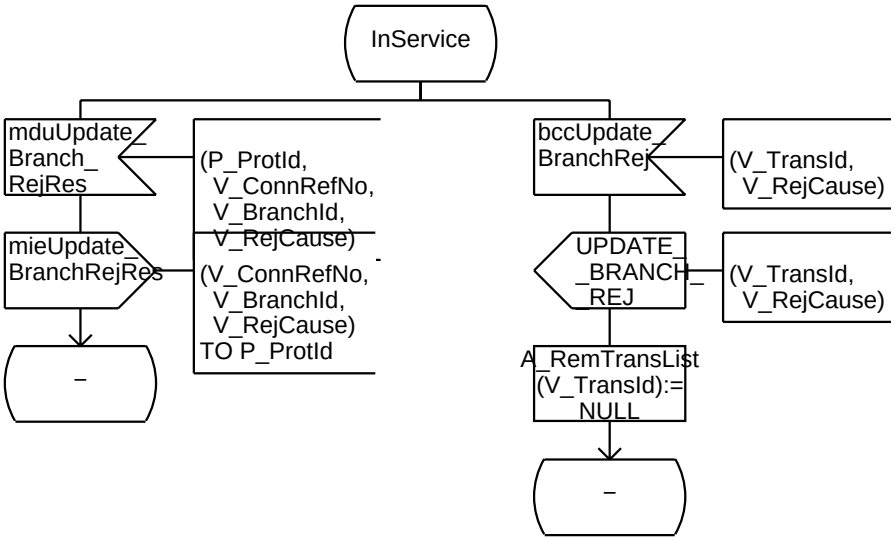
;
/*AN BCC protocol management entity
controls instantiation of
protocol FSMs (process AN_PROT_FSM) */



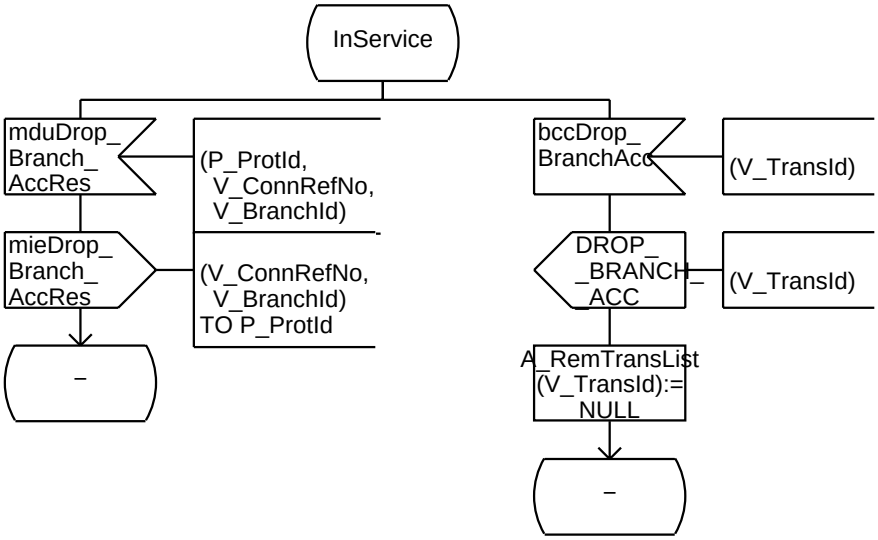


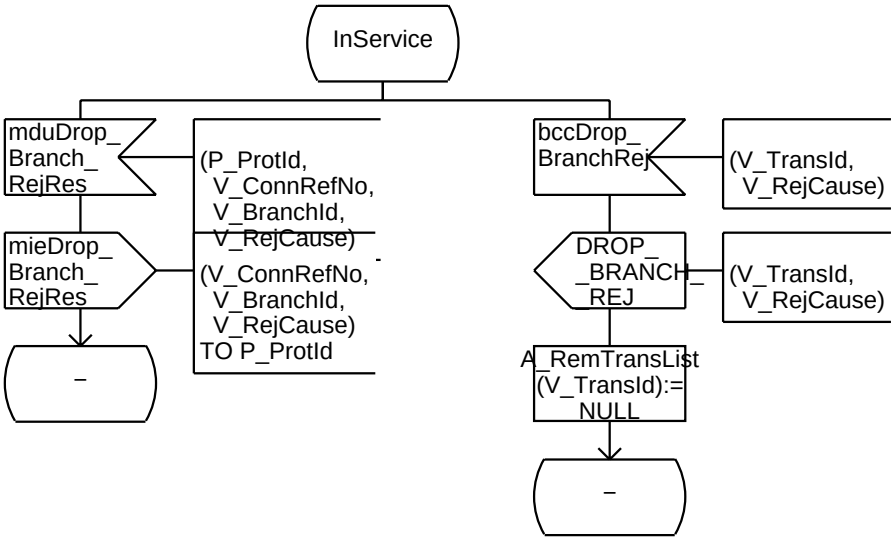
;
/*AN BCC protocol management entity
controls instantiation of
protocol FSMs (process AN_PROT_FSM) */





;
/*AN BCC protocol management entity
controls instantiation of
protocol FSMs (process AN_PROT_FSM) */





```
;
/*AN_PROT_FSM is responsible for sending/
receiving BCC messages*/
```

```
/* Definintions and declarations for
AN_PROT_FSM */
```

```
/* State descriptions

InService: Init State of AN_PROT_FSM, process is waiting for initial AN_SYSMGT primitives or SN messages
AwaitLocAck: AN_PROT_FSM has passed a SN request to AN_SYSMGT and is now awaiting the AN_SYSMGT response
AwaitRemAck: AN_PROT_FSM has sent a VB5 message to the SN and is now awaiting the SN response
*/
```

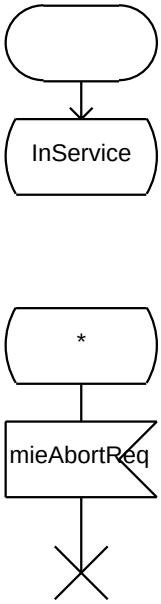
```
/* Timerdefinitions */
TIMER
T_AnFault := 15;   /* Default value for AN_FAULT supervision timer is 10 sec, Tolerance +/− 10%  */
```

```
/* AN_PROT_FSM internal variables and constants*/
/**/
/* timer handling */
DCL V_ExpCount NATURAL;  /*number of actual message repetitions*/
/*possible values*/
SYNONYM C_ExpMax INTEGER = 1;        /*maximum number of message repetitions is 1*/
/**/
/*syntax check handling*/
DCL V_SynResult IT_SynResult;         /*syntax check result, provided by procedure IE_CHECK*/
```

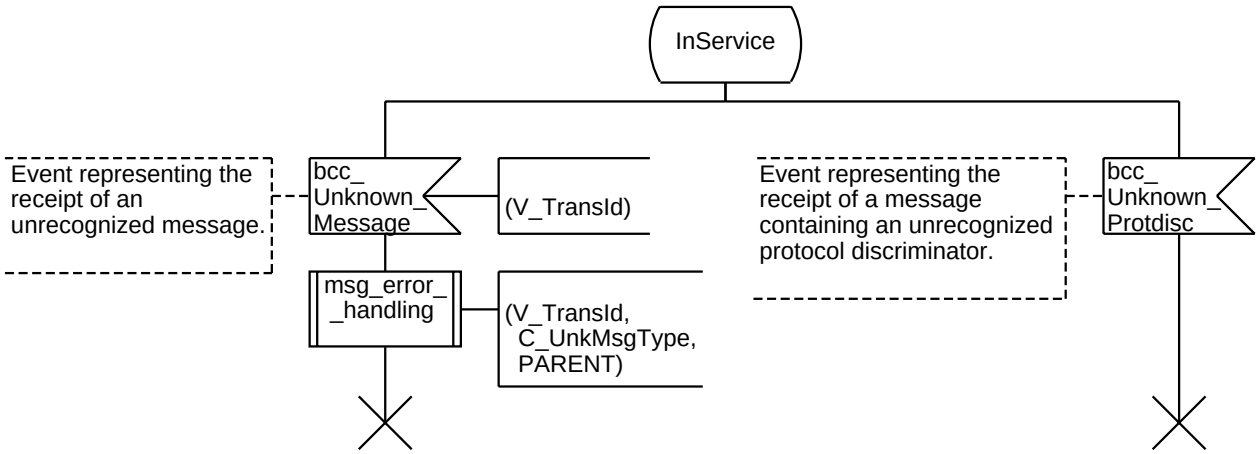
```
/* Signal data declarations*/
/**/
/* Transaction identifier from/to SN*/
DCL
V_TransId   IT_TransId;
/**/
/* Connection reference number */
DCL
V_ConnRefNo IT_ConnRefNo;
/**/
/* Branch identifier */
DCL
V_BranchId  IT_BranchId;
/**/
/* Reject cause */
DCL
V_RejCause  IT_RejCause;
/**/
/* Protocol error cause */
DCL
V_ProtErrCause   IT_ProtErrCause;
```

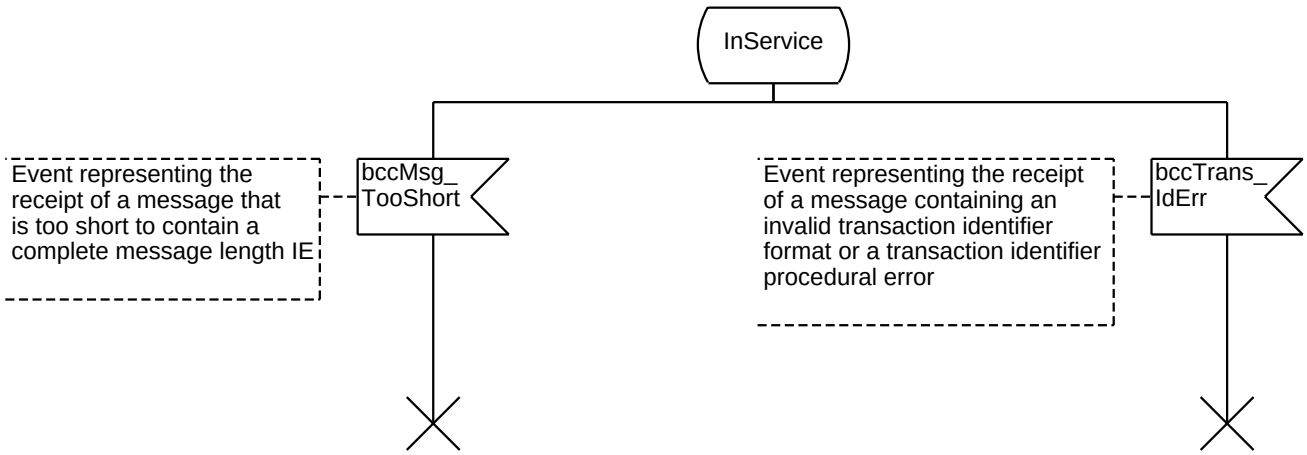
```
/* Internal data declarations */
```

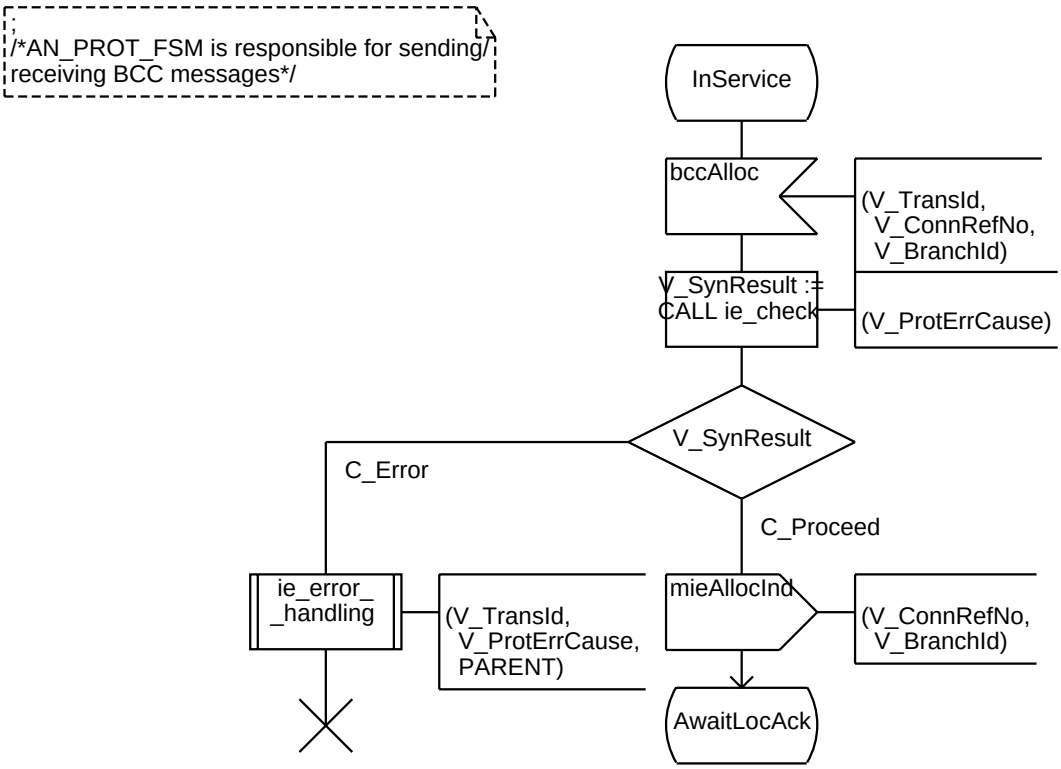
;
/*AN_PROT_FSM is responsible for sending/
receiving BCC messages*/

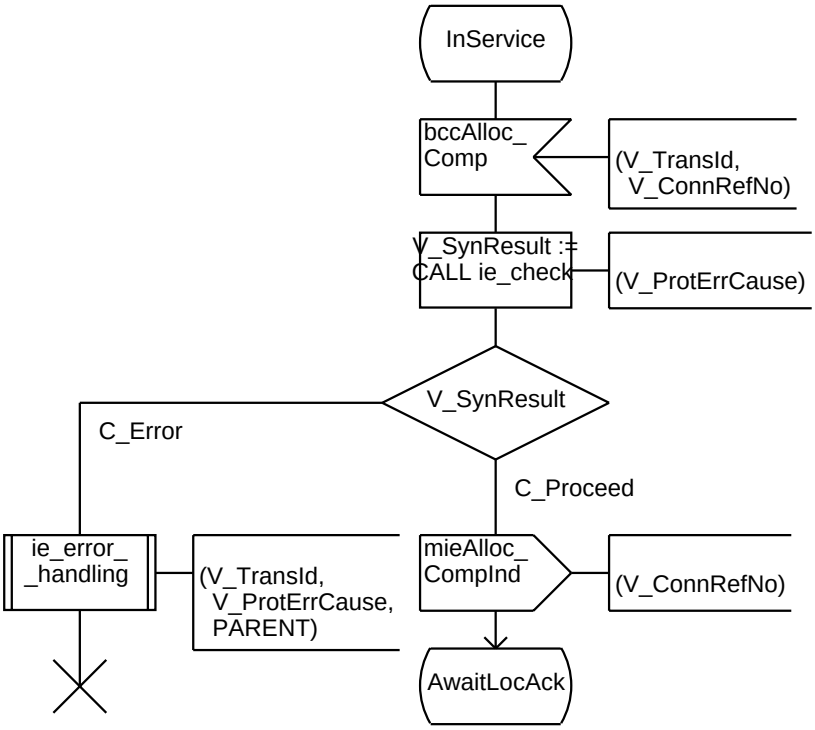


/*AN_PROT_FSM is responsible for sending/ receiving BCC messages*/

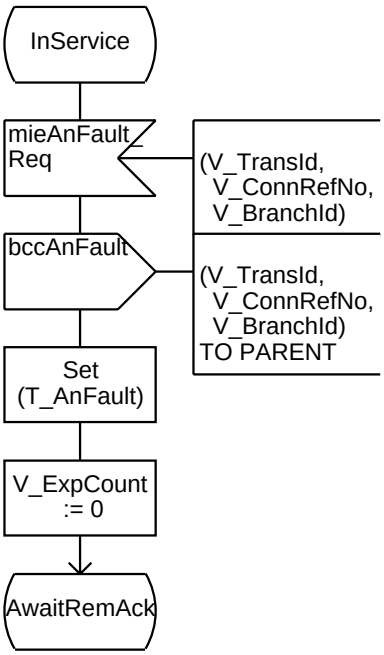


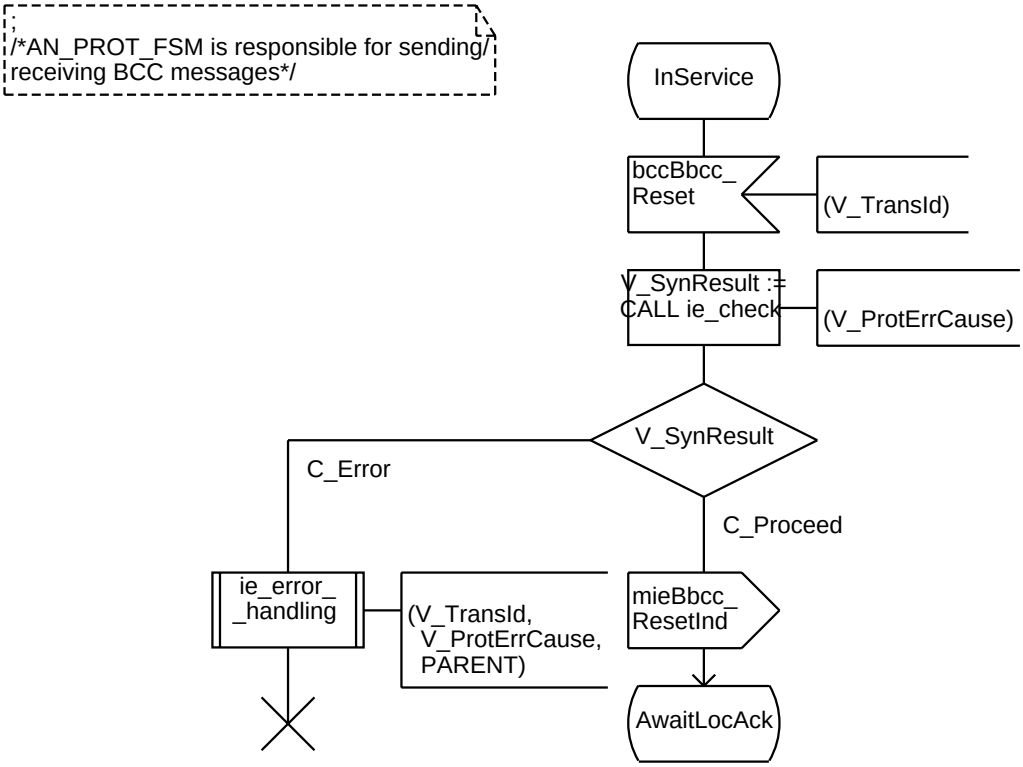


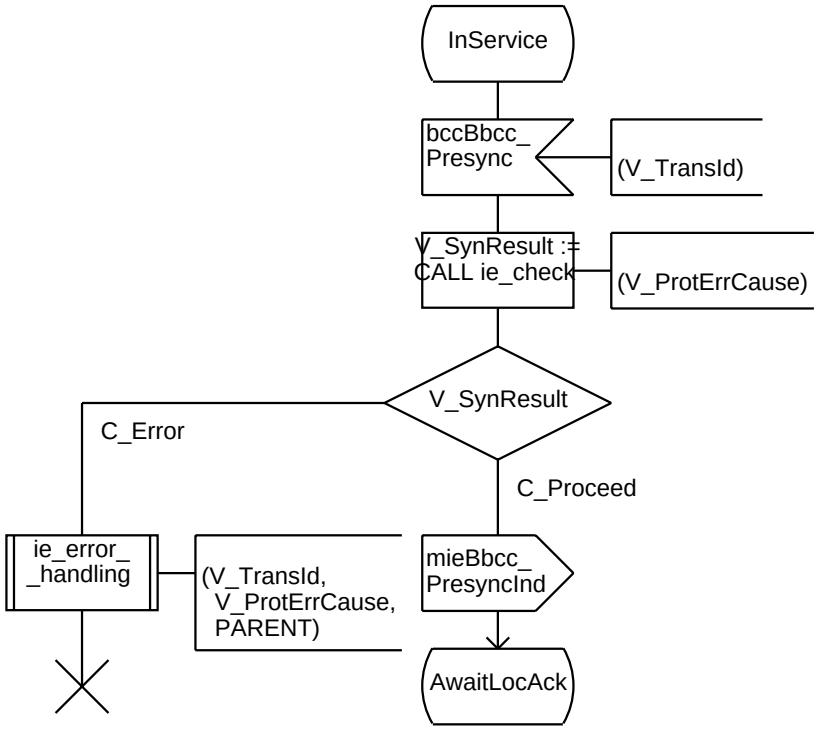

;
/*AN_PROT_FSM is responsible for sending/
receiving BCC messages*/
InService
bccAlloc
(V_TransId,
V_ConnRefNo,
V_BranchId)
V_SynResult :=
CALL ie_check
(V_ProtErrCause)
V_SynResult
C_Error
C_Proceed
ie_error_
_handling
(V_TransId,
V_ProtErrCause,
PARENT)
mieAllocInd
(V_ConnRefNo,
V_BranchId)
AwaitLocAck

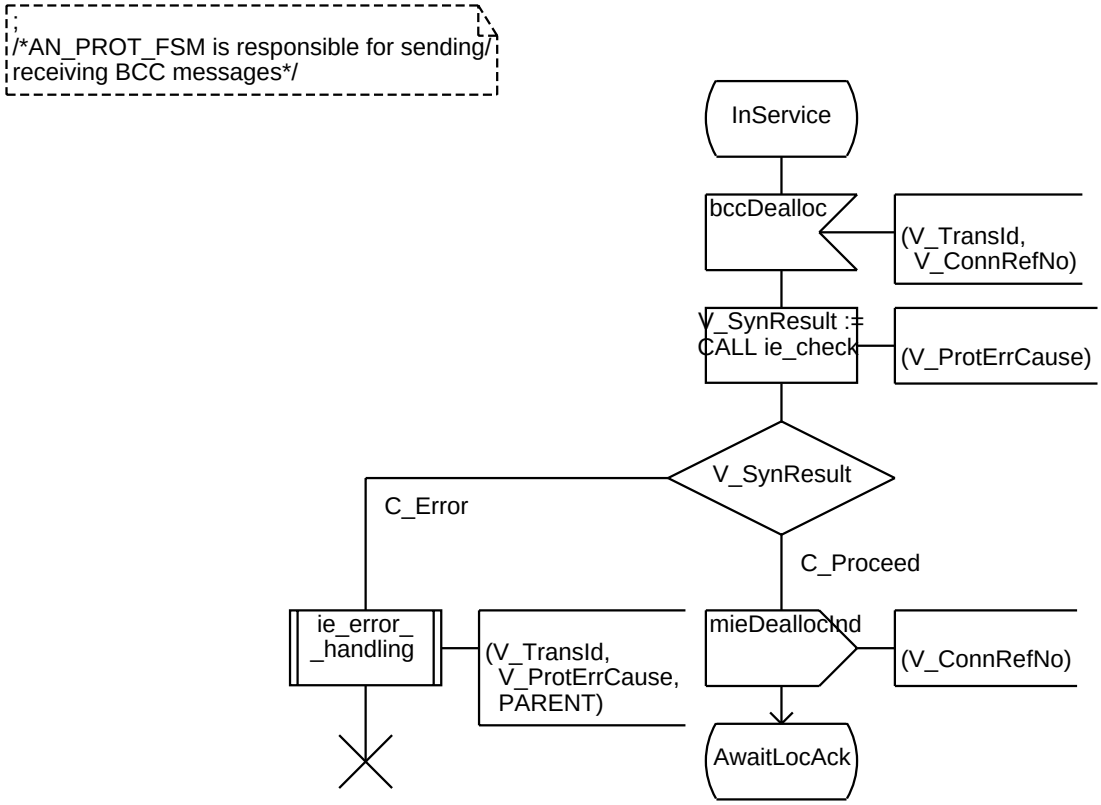

InService
bccAlloc_
Comp
(V_TransId,
V_ConnRefNo)
V_SynResult :=
CALL ie_check
(V_ProtErrCause)
V_SynResult
C_Error
C_Proceed
ie_error_
_handling
(V_TransId,
V_ProtErrCause,
PARENT)
mieAlloc_
CompInd
(V_ConnRefNo)
AwaitLocAck

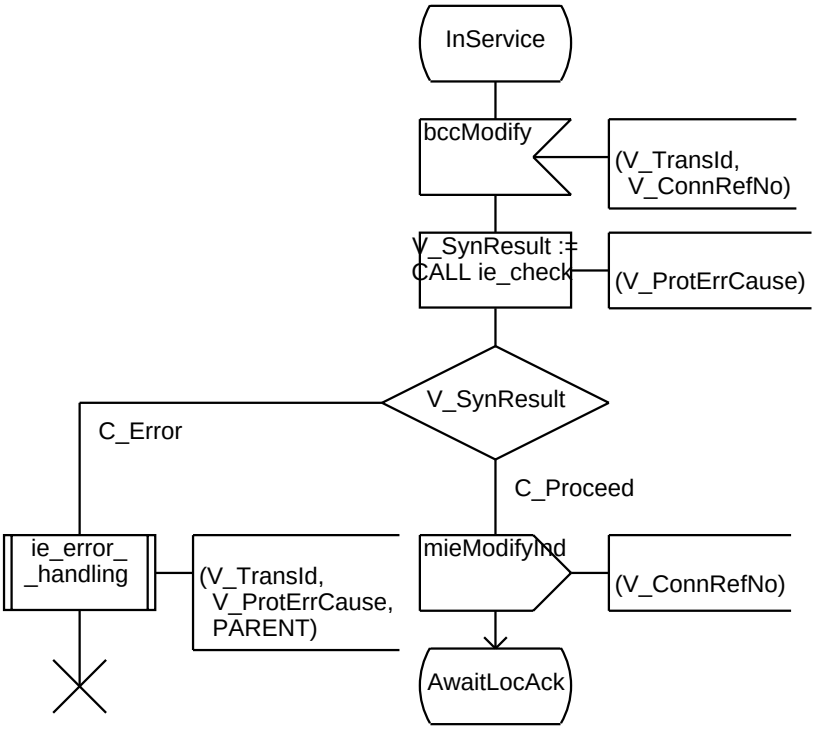
;
/*AN_PROT_FSM is responsible for sending/
receiving BCC messages*/

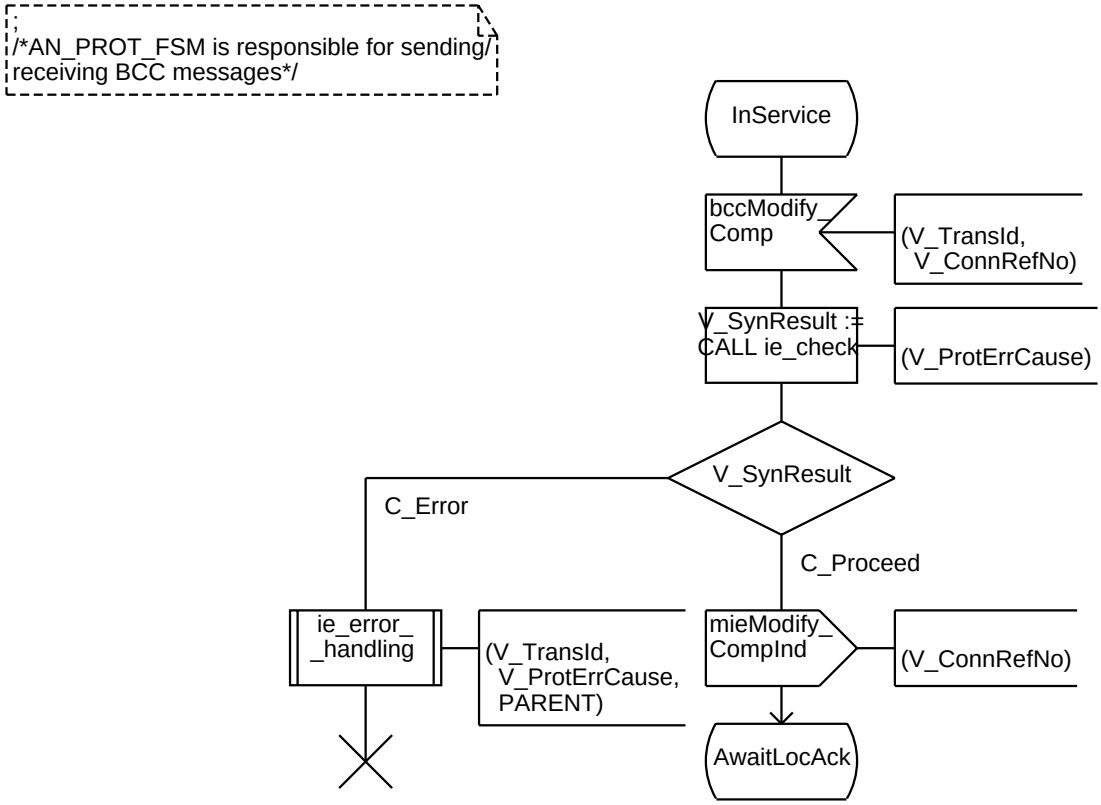


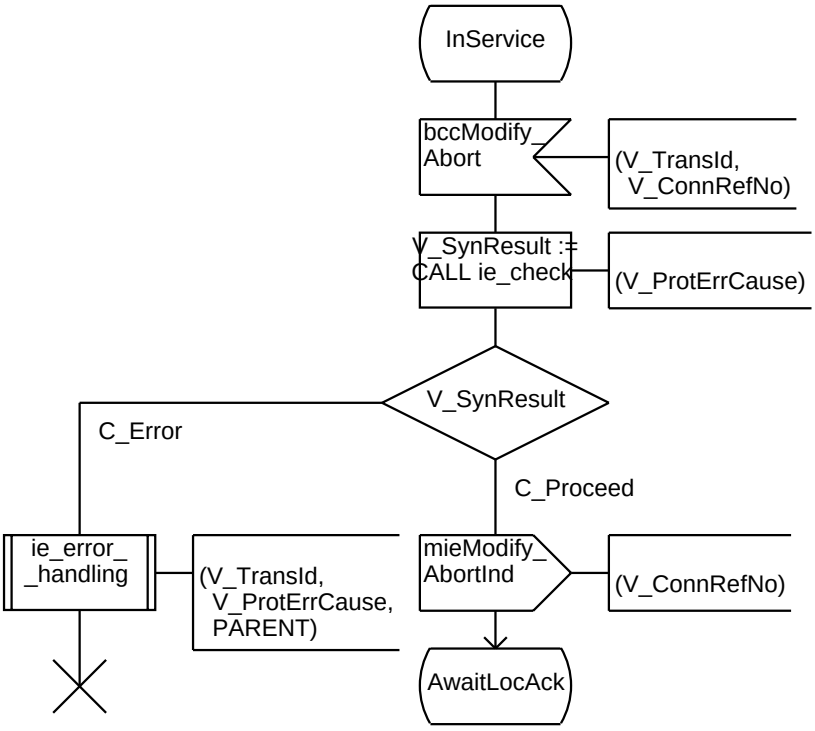

/*AN_PROT_FSM is responsible for sending/
receiving BCC messages*/
InService
bccBbcc_
Reset
(V_TransId)
V_SynResult :=
CALL ie_check
(V_ProtErrCause)
V_SynResult
C_Error
C_Proceed
ie_error_
_handling
(V_TransId,
V_ProtErrCause,
PARENT)
mieBbcc_
ResetInd
AwaitLocAck

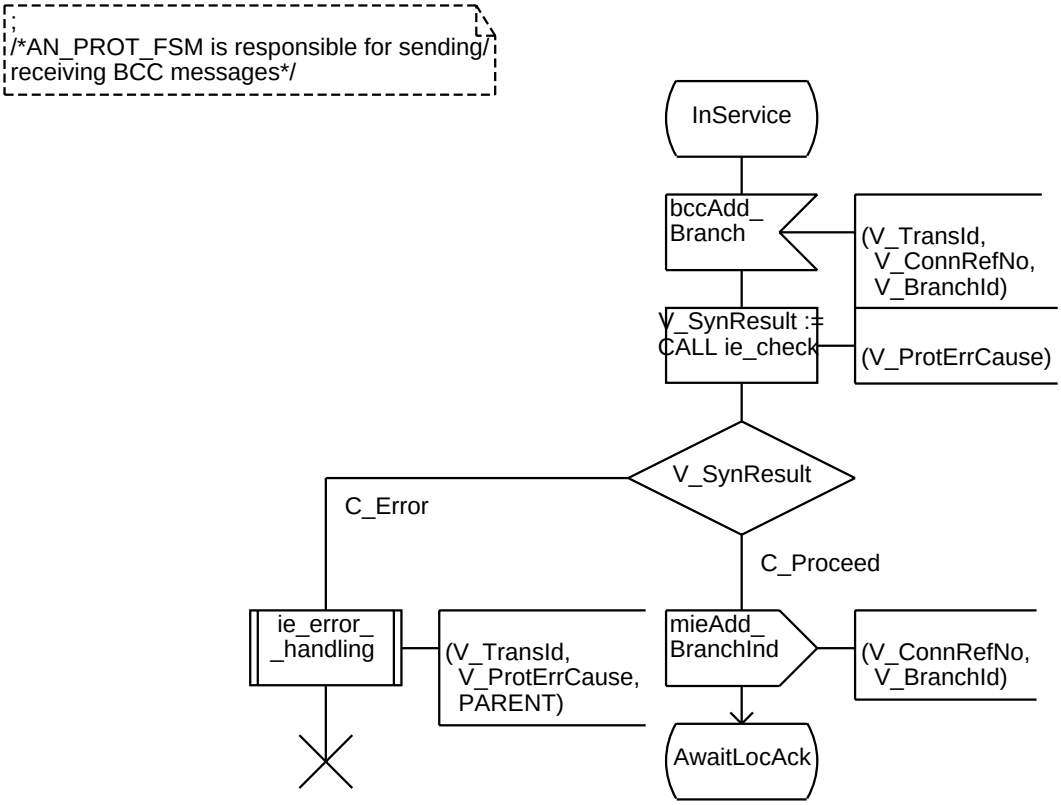

InService
bccBbcc_
Presync
(V_TransId)
V_SynResult :=
CALL ie_check
(V_ProtErrCause)
V_SynResult
C_Error
C_Proceed
ie_error_
_handling
(V_TransId,
V_ProtErrCause,
PARENT)
mieBbcc_
PresyncInd
AwaitLocAck

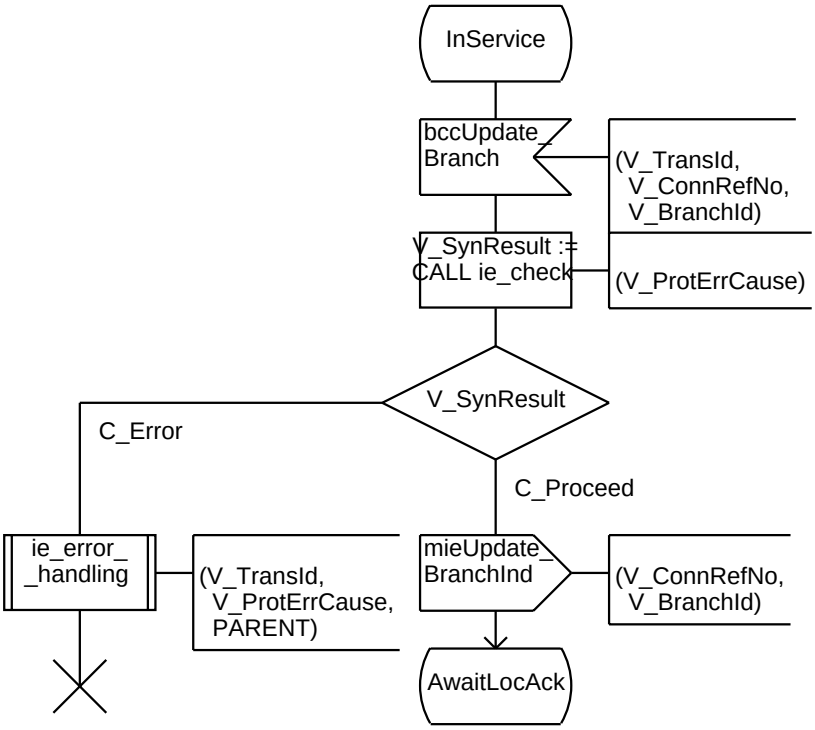

/*AN_PROT_FSM is responsible for sending/
receiving BCC messages*/
InService
bccDealloc
(V_TransId,
V_ConnRefNo)
V_SynResult :=
CALL ie_check
(V_ProtErrCause)
V_SynResult
C_Error
C_Proceed
ie_error_
_handling
(V_TransId,
V_ProtErrCause,
PARENT)
mieDeallocInd
(V_ConnRefNo)
AwaitLocAck

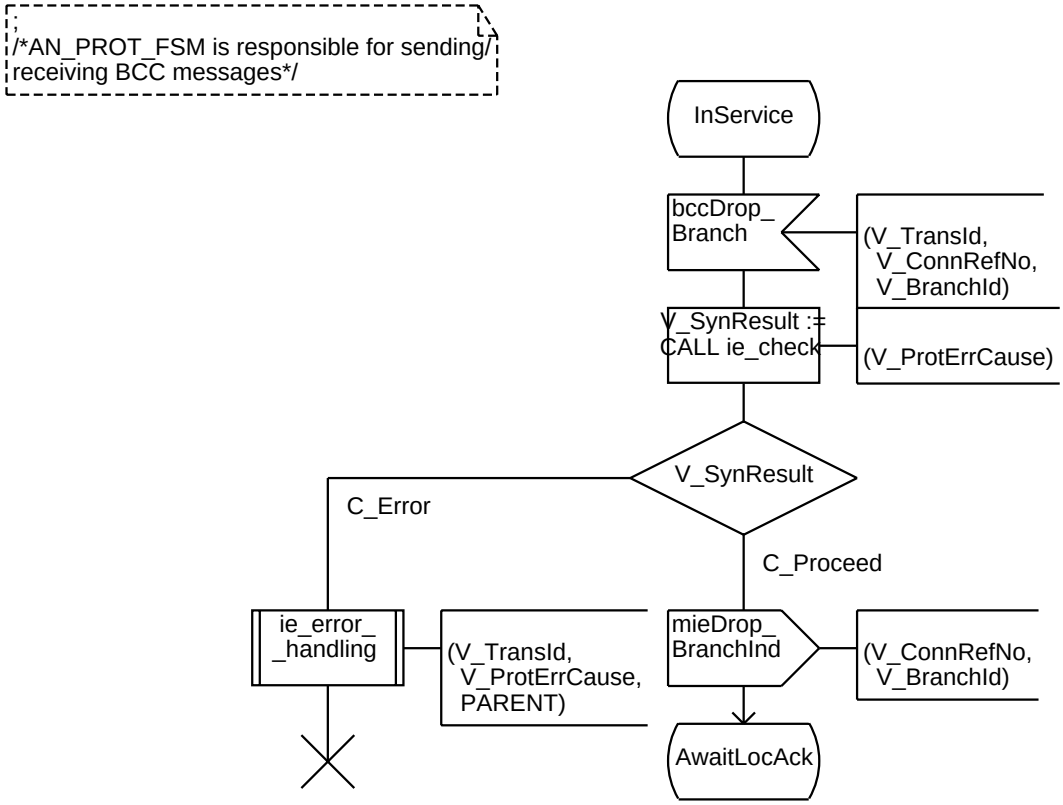

InService
bccModify
(V_TransId,
V_ConnRefNo)
V_SynResult :=
CALL ie_check
(V_ProtErrCause)
V_SynResult
C_Error
C_Proceed
ie_error_
_handling
(V_TransId,
V_ProtErrCause,
PARENT)
mieModifyInd
(V_ConnRefNo)
AwaitLocAck


/*AN_PROT_FSM is responsible for sending/
receiving BCC messages*/
InService
bccModify_
Comp
(V_TransId,
V_ConnRefNo)
V_SynResult :=
CALL ie_check
(V_ProtErrCause)
V_SynResult
C_Error
C_Proceed
ie_error_
_handling
(V_TransId,
V_ProtErrCause,
PARENT)
mieModify_
CompInd
(V_ConnRefNo)
AwaitLocAck

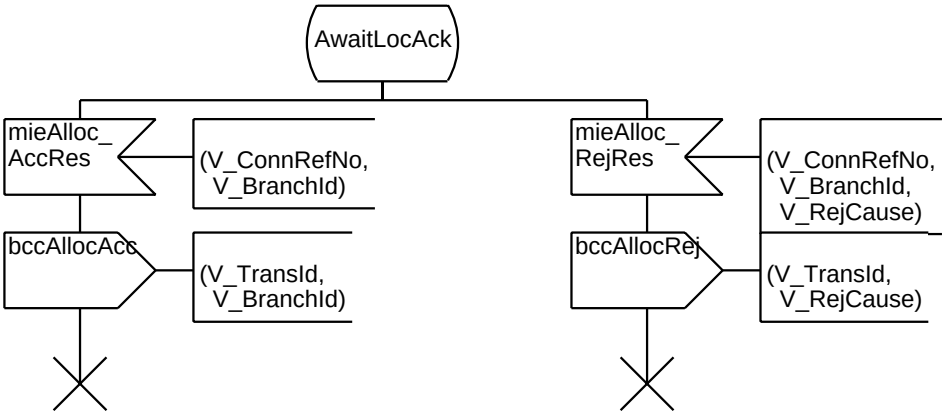

InService
bccModify_
Abort
(V_TransId,
V_ConnRefNo)
V_SynResult :=
CALL ie_check
(V_ProtErrCause)
V_SynResult
C_Error
C_Proceed
ie_error_
_handling
(V_TransId,
V_ProtErrCause,
PARENT)
mieModify_
AbortInd
(V_ConnRefNo)
AwaitLocAck

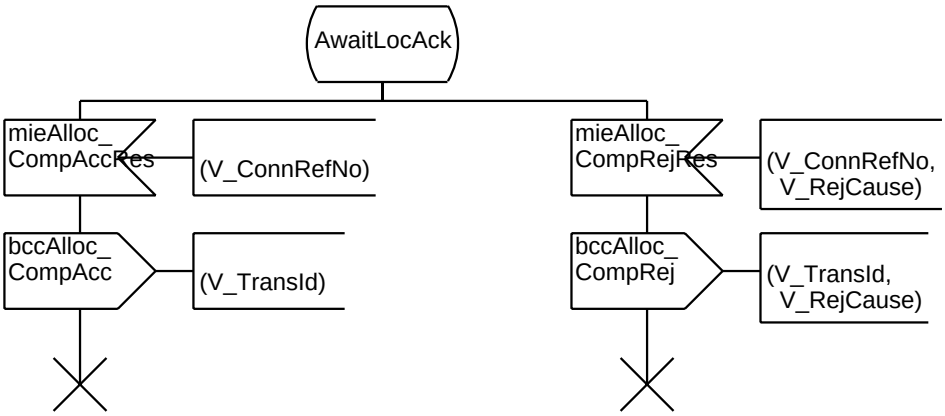

/*AN_PROT_FSM is responsible for sending/
receiving BCC messages*/
InService
bccAdd_
Branch
(V_TransId,
V_ConnRefNo,
V_BranchId)
V_SynResult :=
CALL ie_check
(V_ProtErrCause)
V_SynResult
C_Error
C_Proceed
ie_error_
_handling
(V_TransId,
V_ProtErrCause,
PARENT)
mieAdd_
BranchInd
(V_ConnRefNo,
V_BranchId)
AwaitLocAck

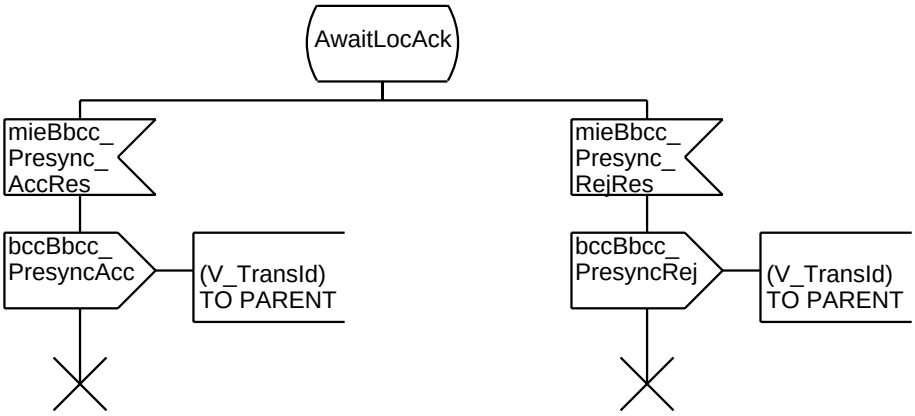

InService
bccUpdate_
Branch
(V_TransId,
V_ConnRefNo,
V_BranchId)
V_SynResult :=
CALL ie_check
(V_ProtErrCause)
V_SynResult
C_Error
C_Proceed
ie_error_
_handling
(V_TransId,
V_ProtErrCause,
PARENT)
mieUpdate_
BranchInd
(V_ConnRefNo,
V_BranchId)
AwaitLocAck


/*AN_PROT_FSM is responsible for sending/
receiving BCC messages*/
InService
bccDrop_
Branch
(V_TransId,
V_ConnRefNo,
V_BranchId)
V_SynResult :=
CALL ie_check
(V_ProtErrCause)
V_SynResult
C_Error
C_Proceed
ie_error_
_handling
(V_TransId,
V_ProtErrCause,
PARENT)
mieDrop_
BranchInd
(V_ConnRefNo,
V_BranchId)
AwaitLocAck

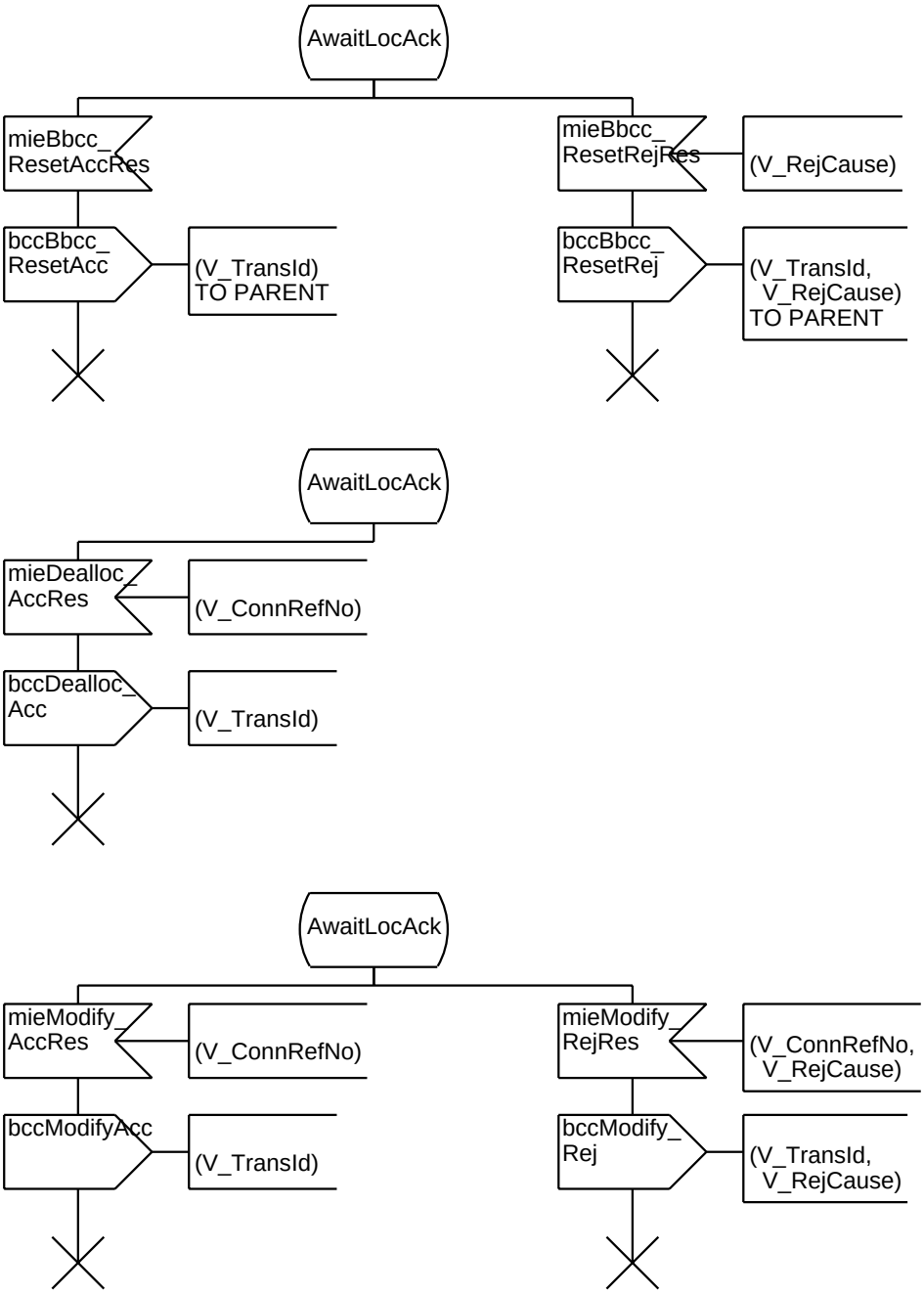
/*AN_PROT_FSM is responsible for sending/
receiving BCC messages*/



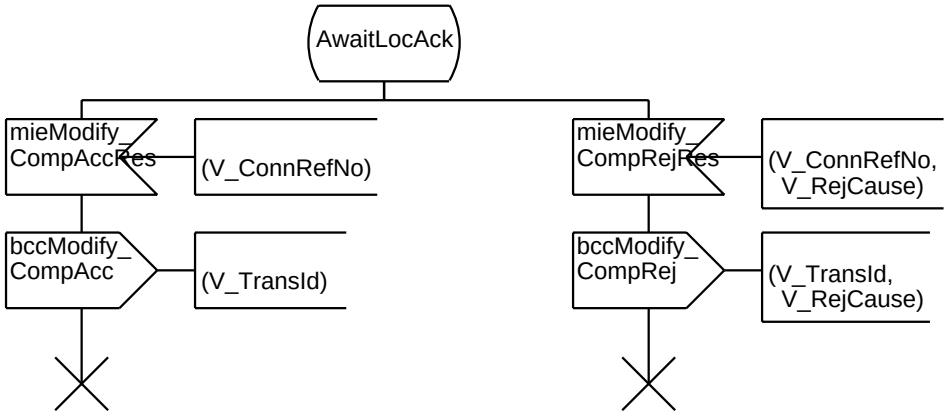


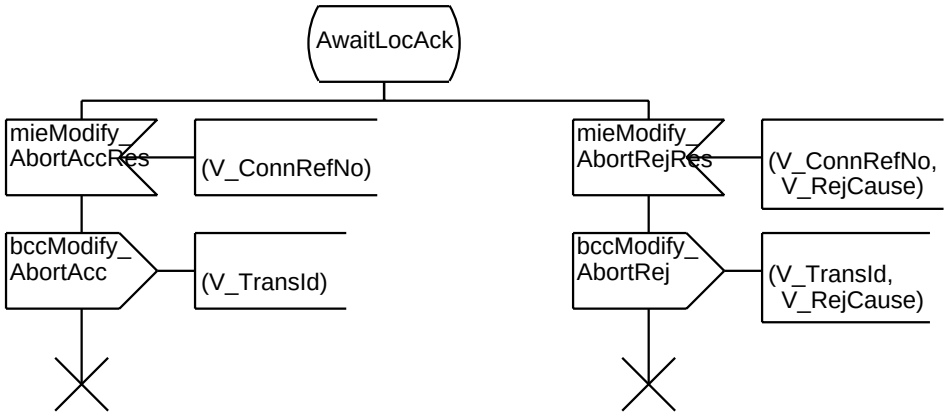


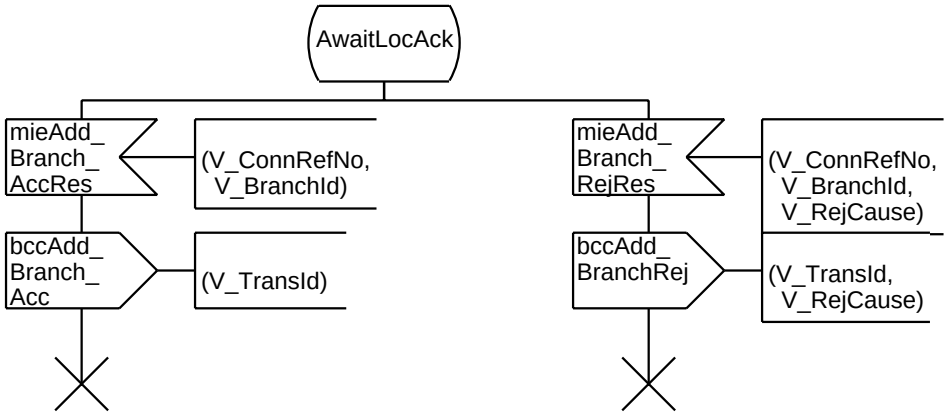
/*AN_PROT_FSM is responsible for sending/
receiving BCC messages*/



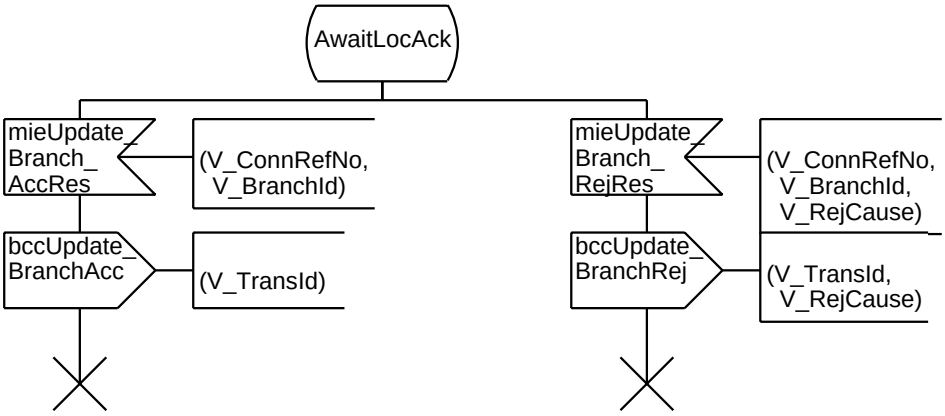
/*AN_PROT_FSM is responsible for sending/
receiving BCC messages*/

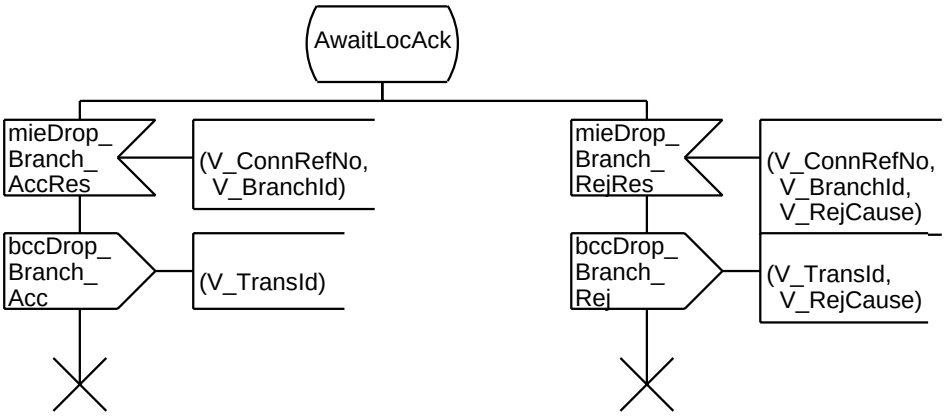


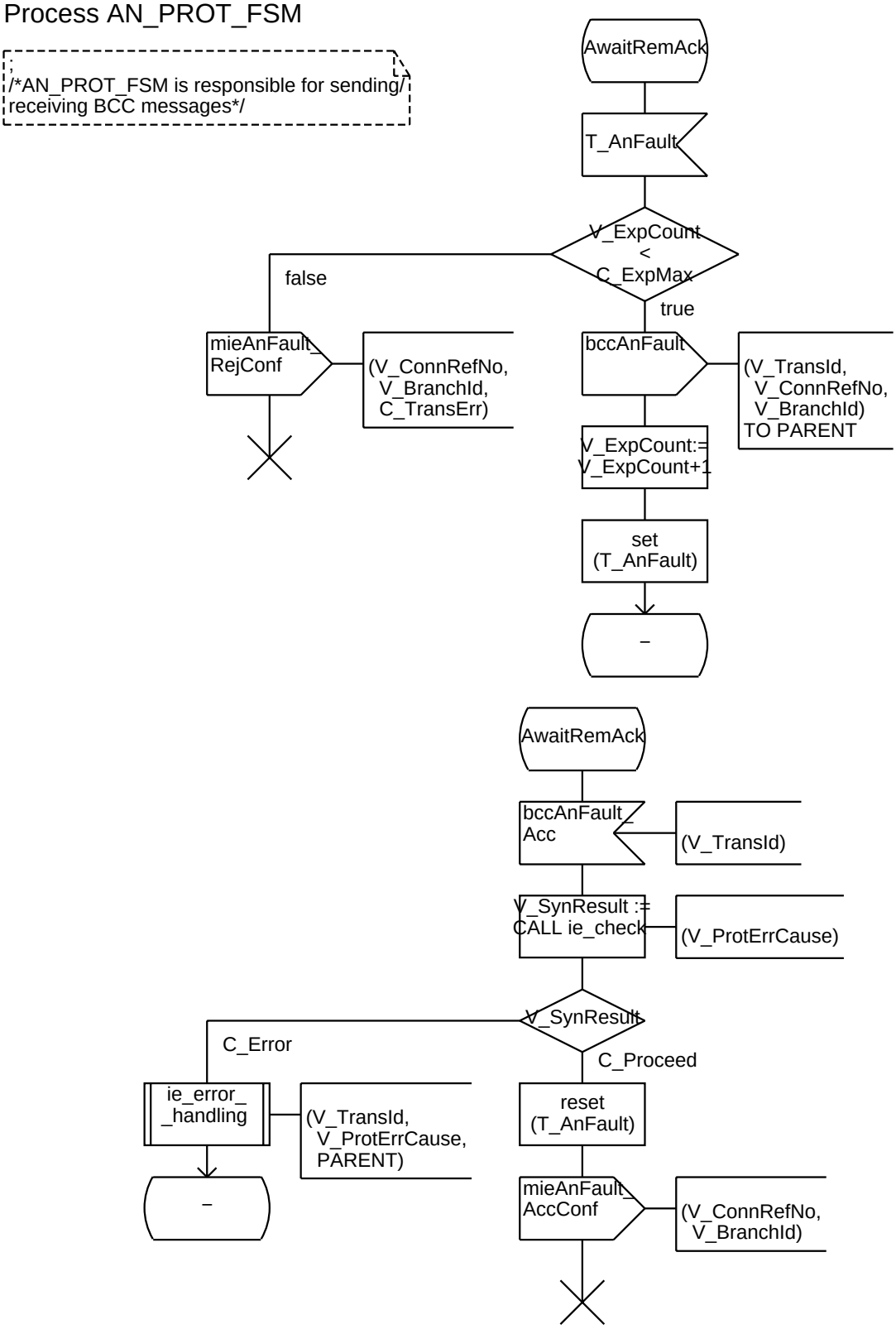


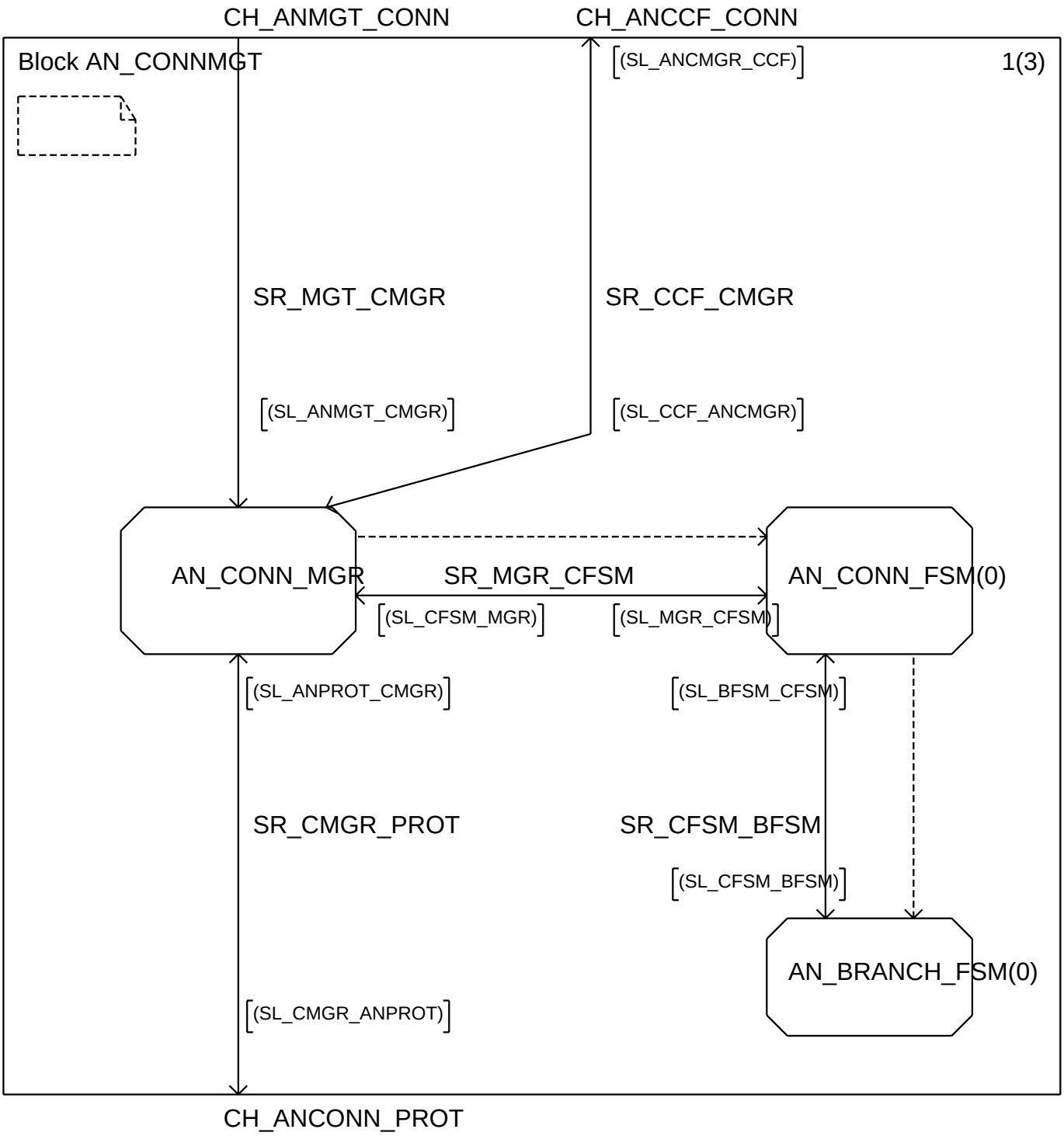


/*AN_PROT_FSM is responsible for sending/
receiving BCC messages*/






/*AN_PROT_FSM is responsible for sending/
receiving BCC messages*/
AwaitRemAck
T_AnFault
V_ExpCount
<
C_ExpMax
false
true
mieAnFault_
RejConf
(V_ConnRefNo,
V_BranchId,
C_TransErr)
bccAnFault
(V_TransId,
V_ConnRefNo,
V_BranchId)
TO PARENT
V_ExpCount:=
V_ExpCount+1
set
(T_AnFault)
–
AwaitRemAck
bccAnFault_
Acc
(V_TransId)
V_SynResult :=
CALL ie_check
(V_ProtErrCause)
V_SynResult
C_Error
C_Proceed
ie_error_
_handling
(V_TransId,
V_ProtErrCause,
PARENT)
reset
(T_AnFault)
–
mieAnFault_
AccConf
(V_ConnRefNo,
V_BranchId)


CH_ANMGT_CONN
CH_ANCCF_CONN
Block AN_CONNMGT
(SL_ANCMGR_CCF)
1(3)
SR_MGT_CMGR
SR_CCF_CMGR
(SL_ANMGT_CMGR)
(SL_CCF_ANCMGR)
AN_CONN_MGR
SR_MGR_CFSM
AN_CONN_FSM(0)
(SL_CFSM_MGR)
(SL_MGR_CFSM)
(SL_ANPROT_CMGR)
(SL_BFSM_CFSM)
SR_CMGR_PROT
SR_CFSM_BFSM
(SL_CFSM_BFSM)
AN_BRANCH_FSM(0)
(SL_CMGR_ANPROT)
CH_ANCONN_PROT

Block AN_CONNMGT

```
/* Declaration of primitives internal
   to block AN_CONNMGT, between  AN_CONN_MGR
   and  AN_CONN_FSM*/
```

```
signal
mieAllocCompInd   (IT_ConnRefNo),
mieAllocCompRes   (IT_ConnRefNo, IT_RejCause),
mieModifyInd      (IT_ConnRefNo),
mieModifyRes      (IT_ConnRefNo, IT_RejCause),
mieModifyCompInd  (IT_ConnRefNo),
mieModifyCompRes  (IT_ConnRefNo, IT_RejCause),
mieModifyAbortInd (IT_ConnRefNo),
mieModifyAbortRes (IT_ConnRefNo, IT_RejCause),
mieAddBranchInd   (IT_ConnRefNo, IT_BranchId),
mieAddBranchRes   (IT_ConnRefNo, IT_BranchId, IT_RejCause),
mieUpdateBranchInd(IT_ConnRefNo, IT_BranchId),
mieUpdateBranchRes(IT_ConnRefNo, IT_BranchId, IT_RejCause),
mieDropBranchInd  (IT_ConnRefNo, IT_BranchId),
mieDropBranchRes  (IT_ConnRefNo, IT_BranchId, IT_RejCause),
mieDeallocInd     (IT_ConnRefNo),
mieDeallocRes     (IT_ConnRefNo);
```

```
signal
mieAllocAccInd       (IT_ConnRefNo, IT_BranchId),
mieAllocRejInd       (IT_BranchId),
mieAllocCompAccInd,
mieAllocCompRejInd,
mieDeallocAccInd     (IT_ConnRefNo),
mieDeallocAccRes     (IT_ConnRefNo),
mieModifyAccInd,
mieModifyRejInd,
mieModifyCompAccInd,
mieModifyCompRejInd,
mieModifyAbortAccInd,
mieModifyAbortRejInd,
mieAddBranchAccInd   (IT_BranchId, IT_UpdateNeeded
mieAddBranchRejInd   (IT_BranchId),
mieUpdateBranchAccInd (IT_BranchId),
mieUpdateBranchRejInd (IT_BranchId),
mieDropBranchAccInd  (IT_ConnRefNo, IT_BranchId),
mieDropBranchAccRes  (IT_ConnRefNo, IT_BranchId),
mieDropBranchRejInd (IT_BranchId),
mieResetInd          (IT_BranchId),
mieAnFaultInd        (IT_BranchId);
```

```
/* Declaration of primitives internal
   to block AN_CONNMGT, between AN_CONN_FSM
   and AN_BRANCH_FSM */
```

```
signal
mieUpdateBranchFSMInd (IT_BranchId),
mieUpdateBranchFSMRes (IT_BranchId, IT_RejCause),
mieDropBranchFSMInd (IT_BranchId),
mieDropBranchFSMRes (IT_BranchId, IT_RejCause),
mieAddBranchFSMAccInd (IT_BranchId, IT_UpdateNeeded),
mieAddBranchFSMRejInd,
mieUpdateBranchFSMAccInd,
mieUpdateBranchFSMRejInd,
mieDropBranchFSMAccInd (IT_BranchId),
mieDropBranchFSMAccRes (IT_BranchId),
mieDropBranchFSMRejInd,
mieDelBranchFSMInd,
mieAnFaultFSMInd;
```

/* Declaration of signallists
internal to block AN_CONNMGT */

```
signallist SL_MGR_CFSM =
mieAllocCompInd,
mieModifyInd,
mieModifyCompInd,
mieModifyAbortInd,
mieAddBranchInd,
mieUpdateBranchInd,
mieDropBranchInd,
mieDeallocInd,
mieAllocAccInd,
mieAllocRejInd,
mieAllocCompAccInd,
mieAllocCompRejInd,
mieDeallocAccInd,
mieModifyAccInd,
mieModifyRejInd,
mieModifyCompAccInd,
mieModifyCompRejInd,
mieModifyAbortAccInd,
mieModifyAbortRejInd,
mieAddBranchAccInd,
mieAddBranchRejInd,
mieUpdateBranchAccInd,
mieUpdateBranchRejInd,
mieDropBranchAccInd,
mieDropBranchRejInd,
mieResetInd,
mieAnFaultInd;
```

```
signallist SL_CFSM_MGR =
mieAllocCompRes,
mieModifyRes,
mieModifyCompRes,
mieModifyAbortRes,
mieAddBranchRes,
mieUpdateBranchRes,
mieDropBranchRes,
mieDropBranchAccRes,
mieDeallocRes,
mieDeallocAccRes;
```

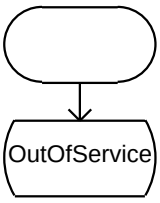
```
signallist SL_CFSM_BFSM =
mieUpdateBranchFSMInd,
mieDropBranchFSMInd,
mieAddBranchFSMAccInd,
mieAddBranchFSMRejInd,
mieUpdateBranchFSMAccInd,
mieUpdateBranchFSMRejInd,
mieDropBranchFSMAccInd,
mieDropBranchFSMRejInd,
mieDelBranchFSMInd,
mieAnFaultFSMInd;
```

```
signallist SL_BFSM_CFSM =
mieUpdateBranchFSMRes,
mieDropBranchFSMRes,
mieDropBranchFSMAccRes;
```

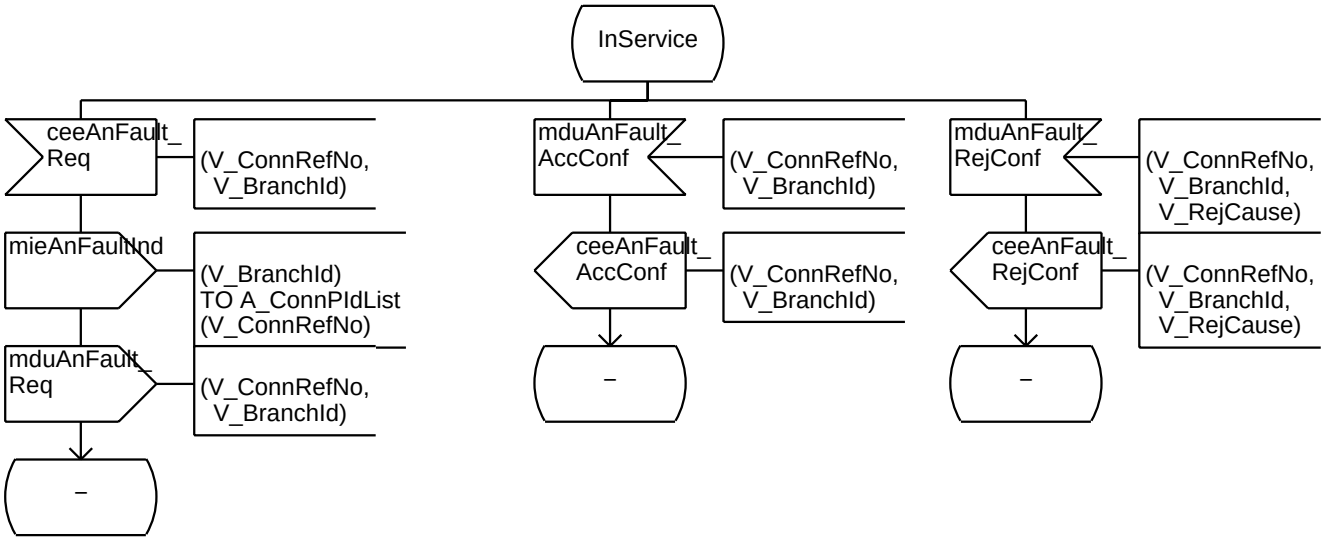
```
/* Data type definitions internal to the process */
/**/
/* This data type is used for translation of the connection reference number to the process identifier (PID)
   of the connection FSM */
NEWTYPE AT_ConnPIdList ARRAY
(IT_ConnRefNo,PID)
ENDNEWTYPE AT_ConnPIdList;
```

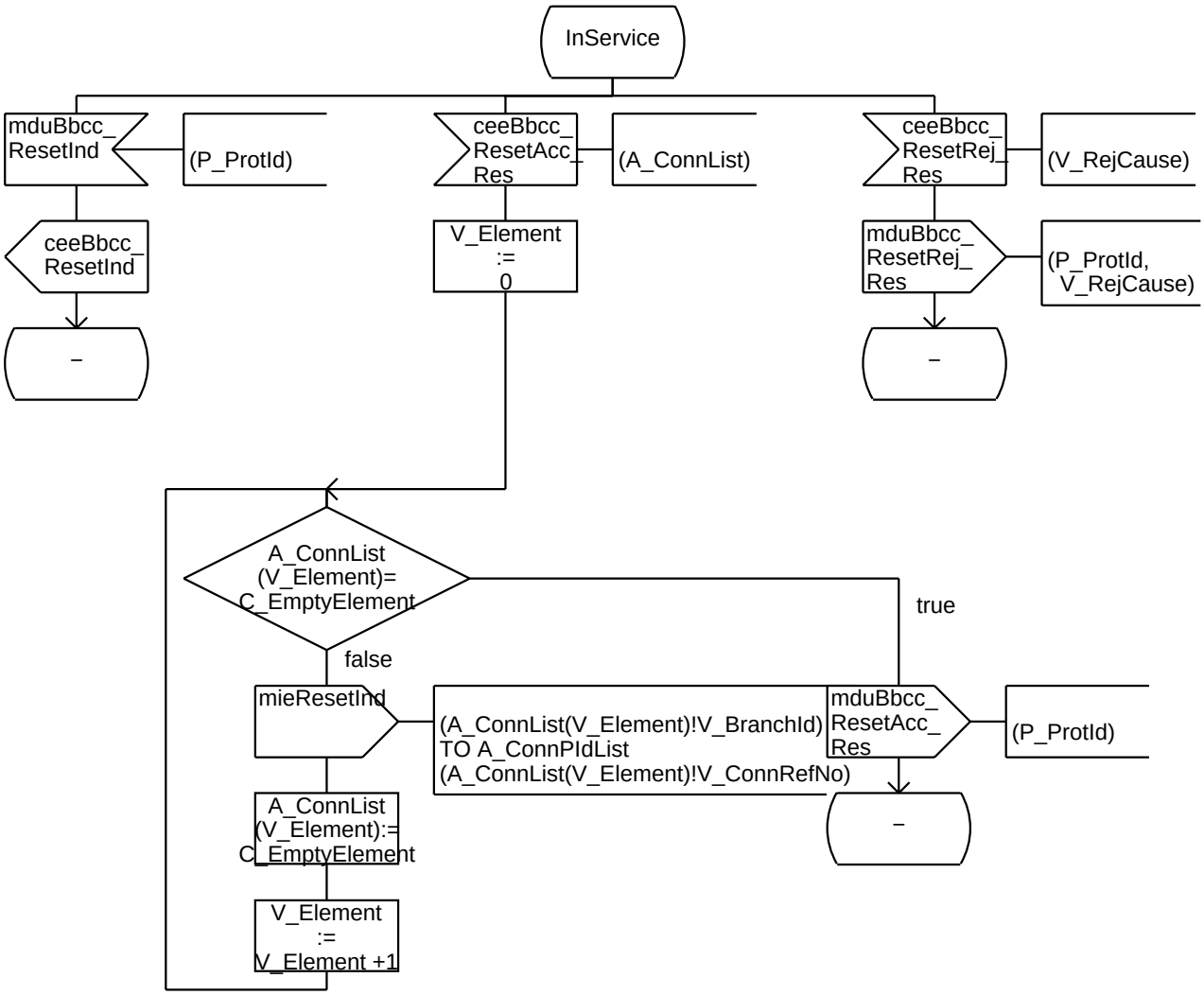
```
/*Primitive Data Declarations*/
/**/
/* Protocol process identifier */
DCL
P_ProtId   PT_ProtId;
/**/
/* List of connections */
DCL
A_ConnList  AT_ConnList;
/**/
/* Connection reference number */
DCL
V_ConnRefNo  IT_ConnRefNo;
/**/
/* Branch identifier */
DCL
V_BranchId  IT_BranchId;
/**/
/* Reject Cause */
DCL
V_RejCause  IT_RejCause;
/**/
/* Update needed for addBranch */
DCL
V_UpdateNeeded  IT_UpdateNeeded;
```

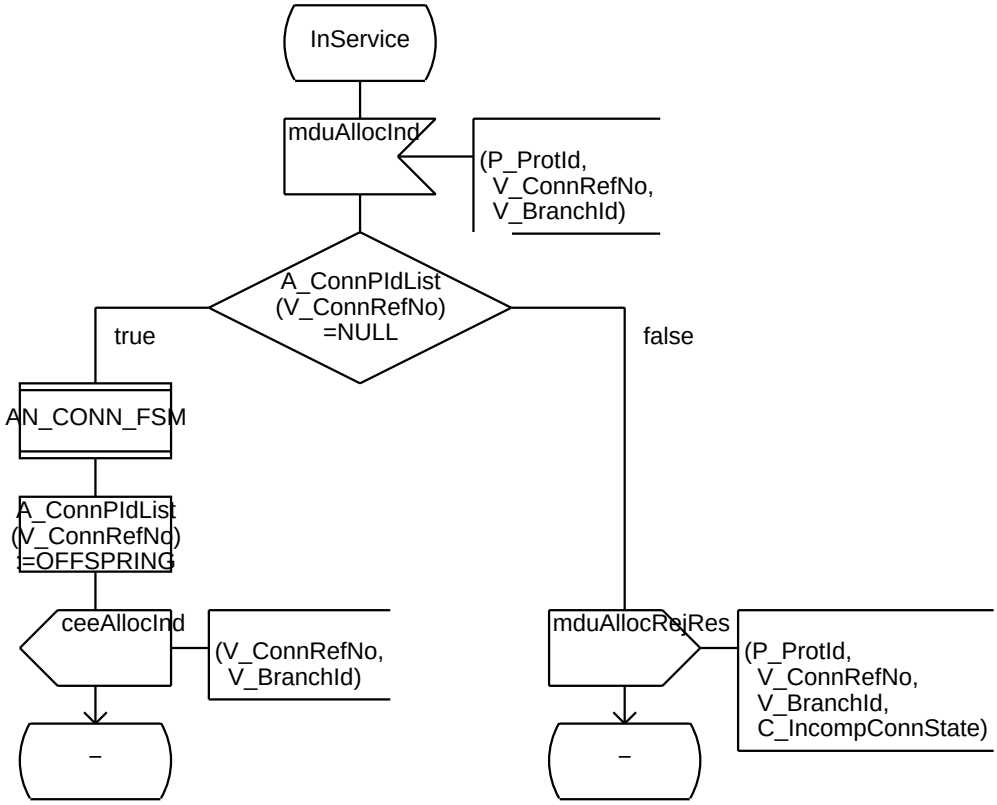
```
/* Process Data Declarations */
/**/
/* List of connection FSMs */
DCL
A_ConnPIdList   AT_ConnPIdList;
/**/
/* Index of A_ConnList */
DCL
V_Element     INTEGER;
```

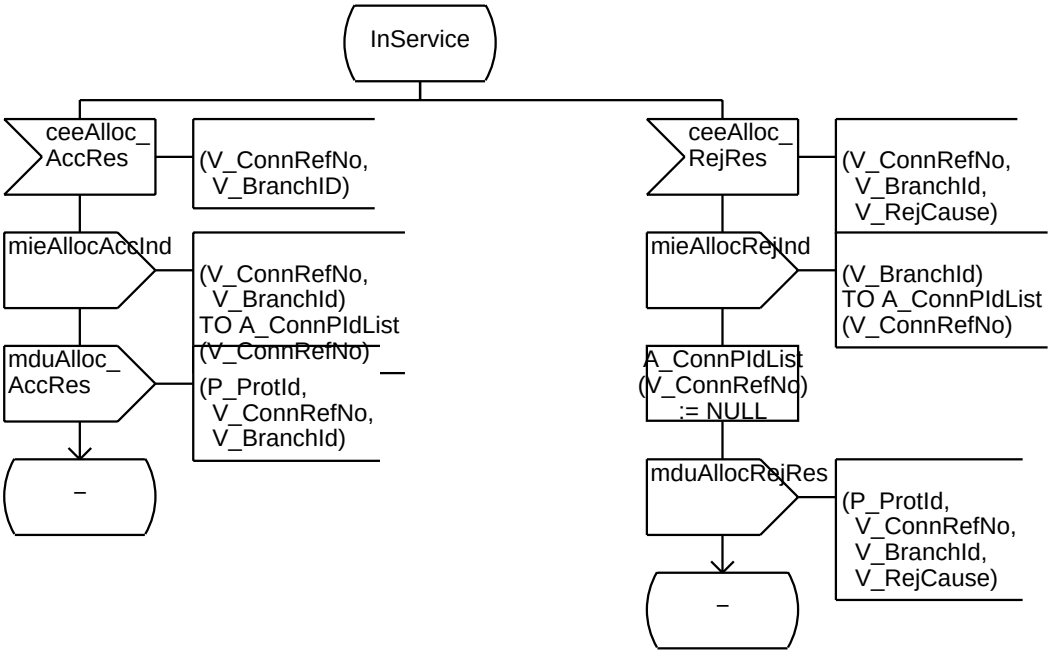


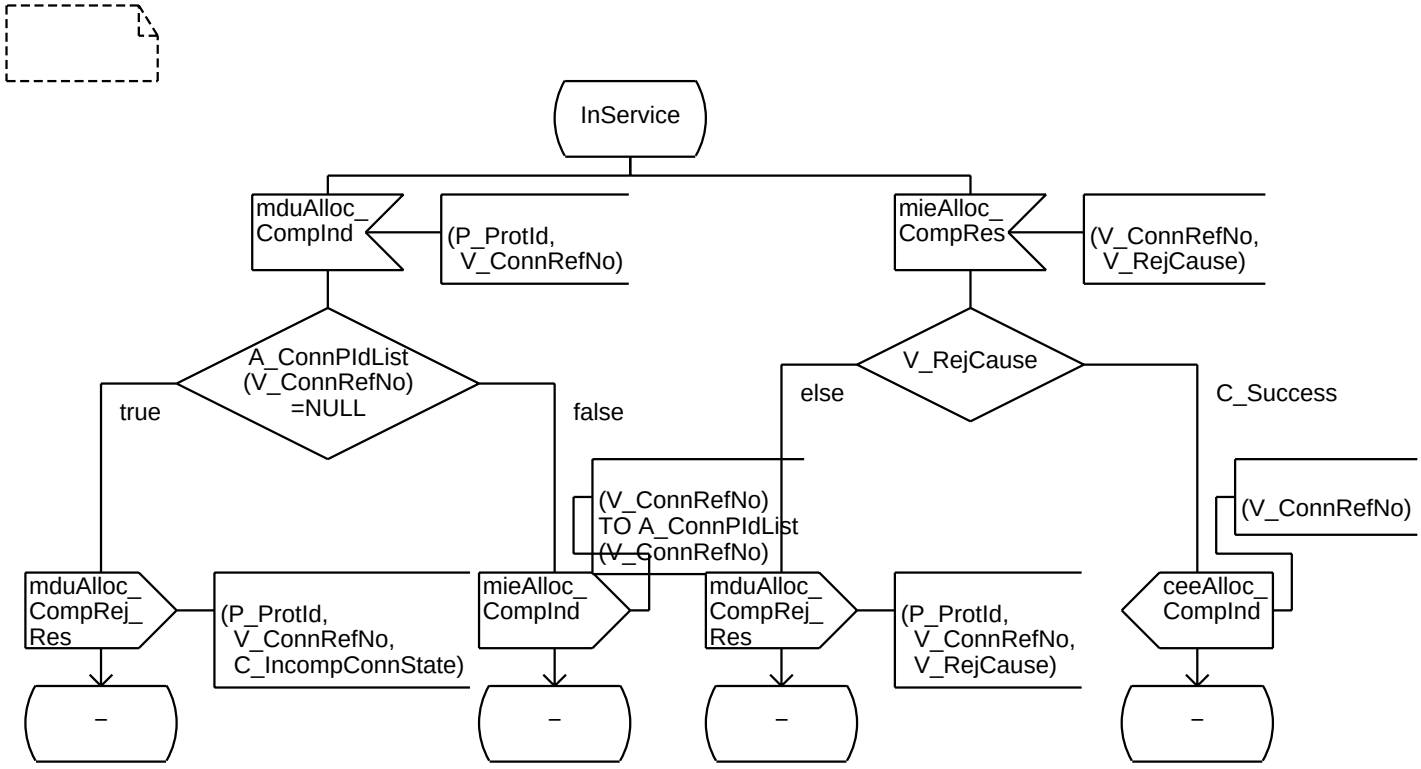
OutOfService
mphStart_
Traffic
InService
InService
mphStop_
Traffic
OutOfService

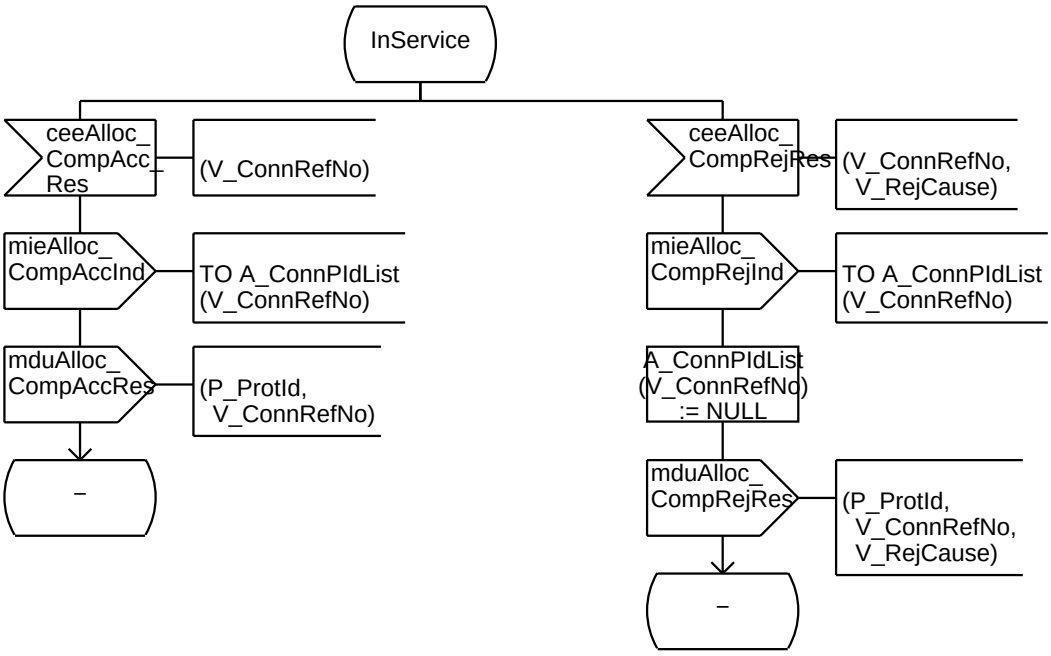

InService
ceeAnFault_
Req
(V_ConnRefNo,
V_BranchId)
mieAnFaultInd
(V_BranchId)
TO A_ConnPIdList
(V_ConnRefNo)
mduAnFault_
Req
(V_ConnRefNo,
V_BranchId)
–
mduAnFault_
AccConf
(V_ConnRefNo,
V_BranchId)
ceeAnFault_
AccConf
(V_ConnRefNo,
V_BranchId)
–
mduAnFault_
RejConf
(V_ConnRefNo,
V_BranchId,
V_RejCause)
ceeAnFault_
RejConf
(V_ConnRefNo,
V_BranchId,
V_RejCause)
–

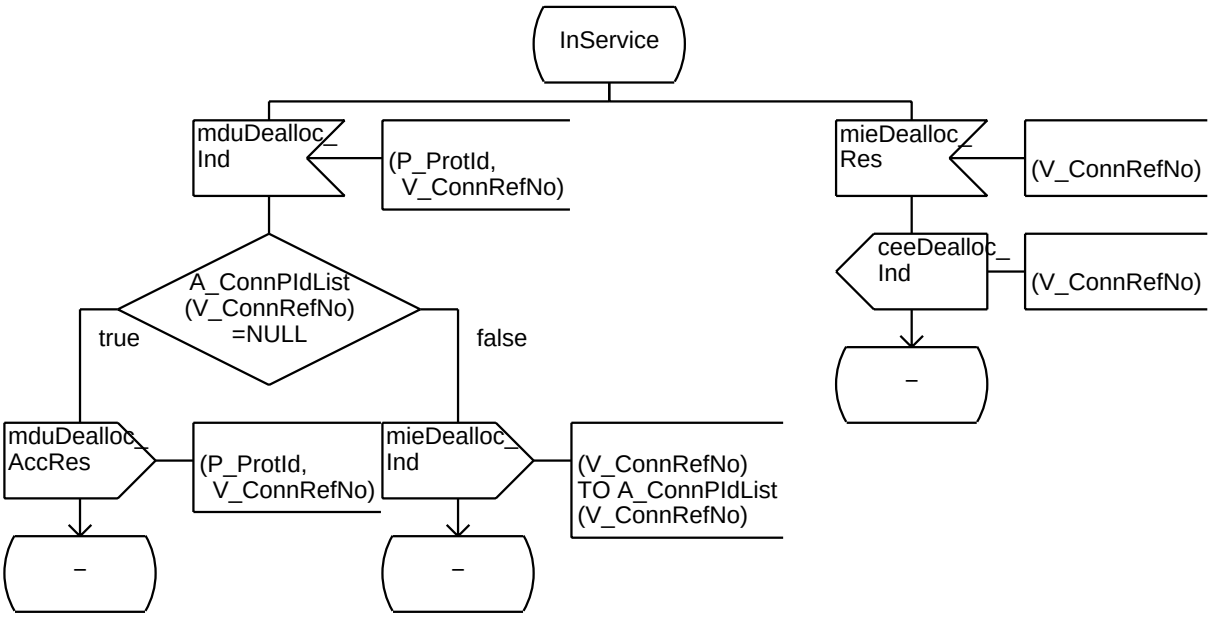

InService
mduBbcc_ ResetInd
(P_ProtId)
ceeBbcc_ ResetAcc_ Res
(A_ConnList)
ceeBbcc_ ResetRej_ Res
(V_RejCause)
ceeBbcc_ ResetInd
V_Element := 0
mduBbcc_ ResetRej_ Res
(P_ProtId, V_RejCause)
–
–
A_ConnList (V_Element)= C_EmptyElement
true
false
mieResetInd
(A_ConnList(V_Element)!V_BranchId) TO A_ConnPIdList (A_ConnList(V_Element)!V_ConnRefNo)
mduBbcc_ ResetAcc_ Res
(P_ProtId)
A_ConnList (V_Element):= C_EmptyElement
–
V_Element := V_Element +1

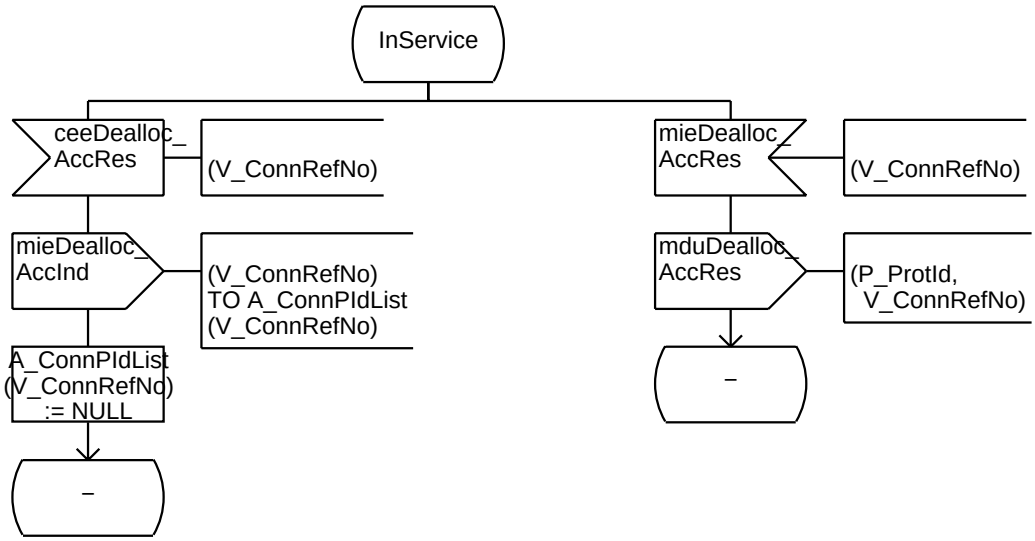

InService
mduAllocInd
(P_ProtId,
V_ConnRefNo,
V_BranchId)
A_ConnPIdList
(V_ConnRefNo)
=NULL
true
false
AN_CONN_FSM
A_ConnPIdList
(V_ConnRefNo)
:=OFFSPRING
ceeAllocInd
(V_ConnRefNo,
V_BranchId)
mduAllocRejRes
(P_ProtId,
V_ConnRefNo,
V_BranchId,
C_IncompConnState)
–
–

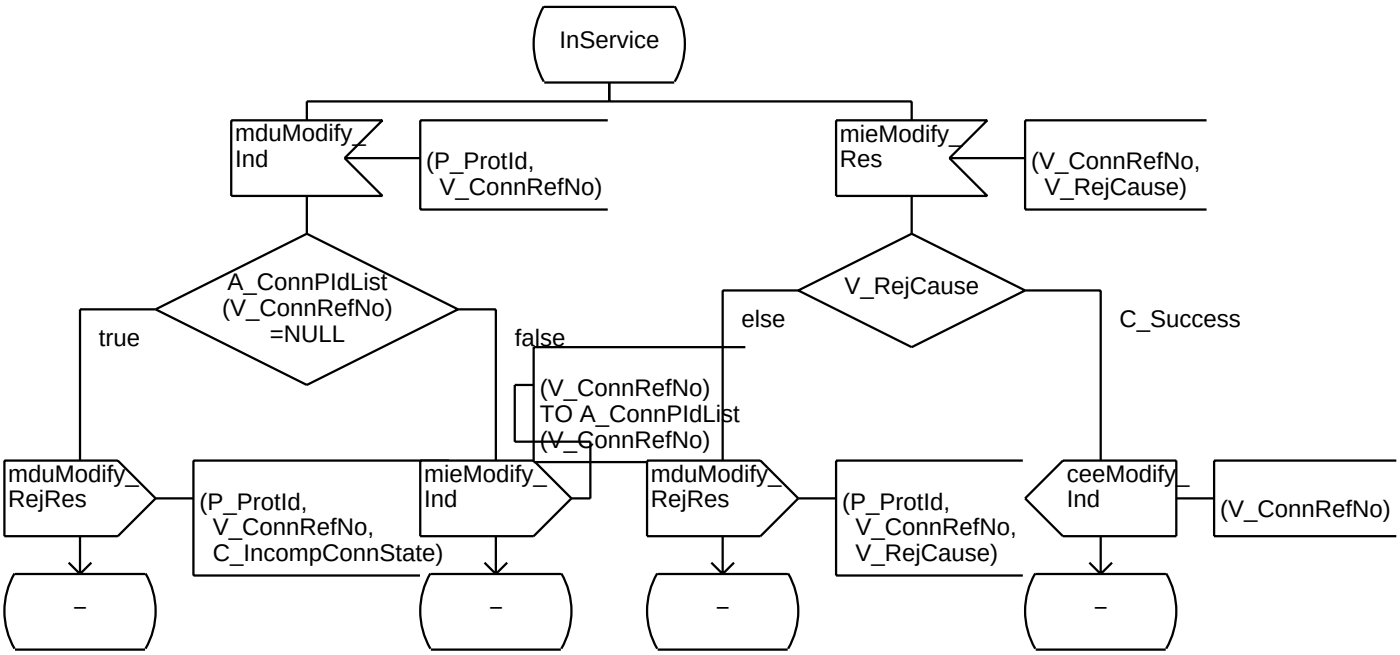

InService
ceeAlloc_
AccRes
(V_ConnRefNo,
V_BranchID)
mieAllocAccInd
(V_ConnRefNo,
V_BranchId)
TO A_ConnPIdList
(V_ConnRefNo)
mduAlloc_
AccRes
(P_ProtId,
V_ConnRefNo,
V_BranchId)
–
ceeAlloc_
RejRes
(V_ConnRefNo,
V_BranchId,
V_RejCause)
mieAllocRejInd
(V_BranchId)
TO A_ConnPIdList
(V_ConnRefNo)
A_ConnPIdList
(V_ConnRefNo)
:= NULL
mduAllocRejRes
(P_ProtId,
V_ConnRefNo,
V_BranchId,
V_RejCause)
–

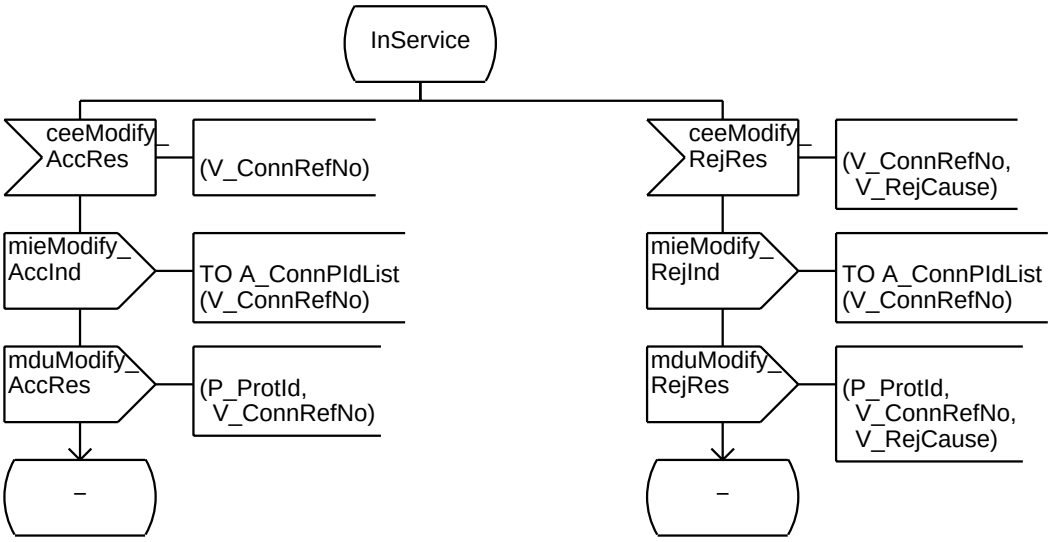

InService
mduAlloc_
CompInd
(P_ProtId,
V_ConnRefNo)
mieAlloc_
CompRes
(V_ConnRefNo,
V_RejCause)
A_ConnPIdList
(V_ConnRefNo)
=NULL
true
false
V_RejCause
else
C_Success
(V_ConnRefNo)
TO A_ConnPIdList
(V_ConnRefNo)
(V_ConnRefNo)
mduAlloc_
CompRej_
Res
(P_ProtId,
V_ConnRefNo,
C_IncompConnState)
mieAlloc_
CompInd
mduAlloc_
CompRej_
Res
(P_ProtId,
V_ConnRefNo,
V_RejCause)
ceeAlloc_
CompInd
–
–
–
–

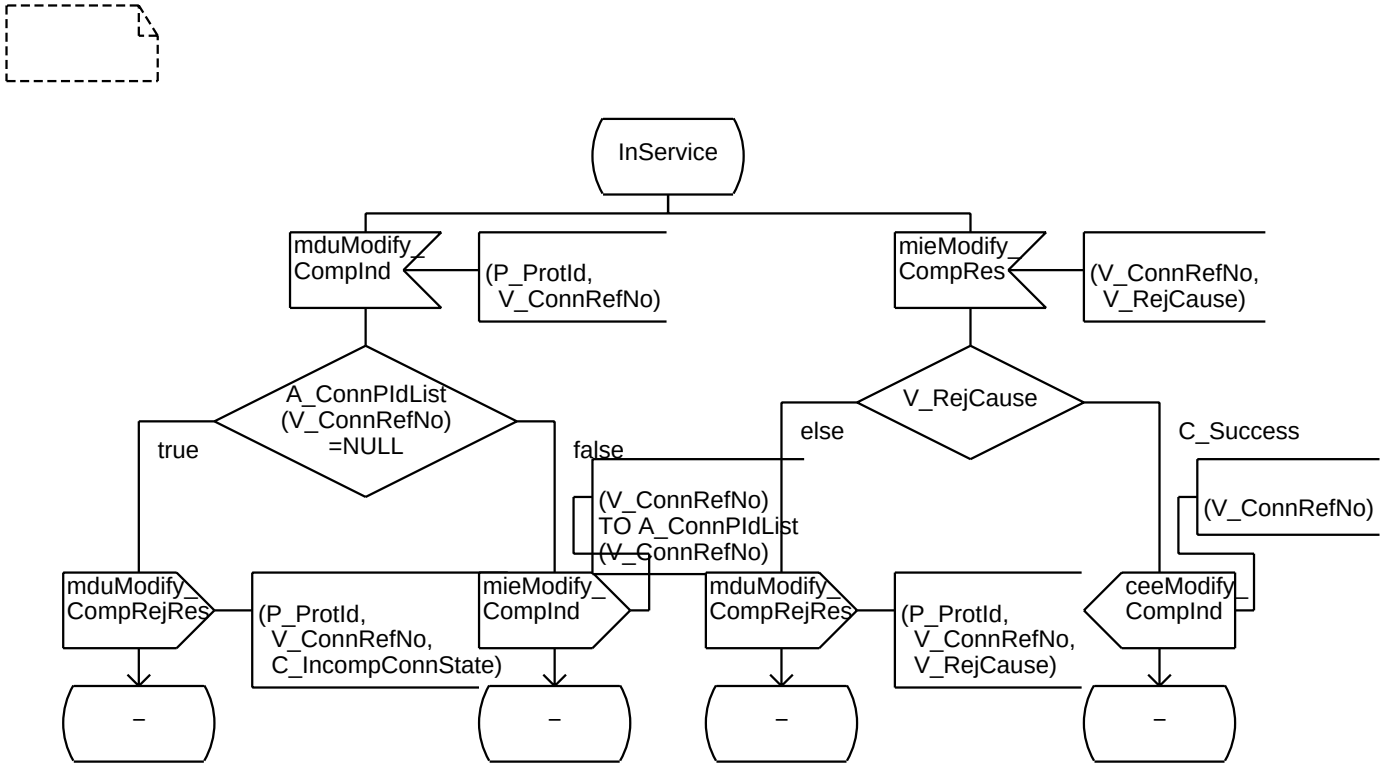

InService
ceeAlloc_
CompAcc_
Res
(V_ConnRefNo)
ceeAlloc_
CompRejRes
(V_ConnRefNo,
V_RejCause)
mieAlloc_
CompAccInd
TO A_ConnPIdList
(V_ConnRefNo)
mieAlloc_
CompRejInd
TO A_ConnPIdList
(V_ConnRefNo)
mduAlloc_
CompAccRes
(P_ProtId,
V_ConnRefNo)
A_ConnPIdList
(V_ConnRefNo)
:= NULL
mduAlloc_
CompRejRes
(P_ProtId,
V_ConnRefNo,
V_RejCause)
–
–
–

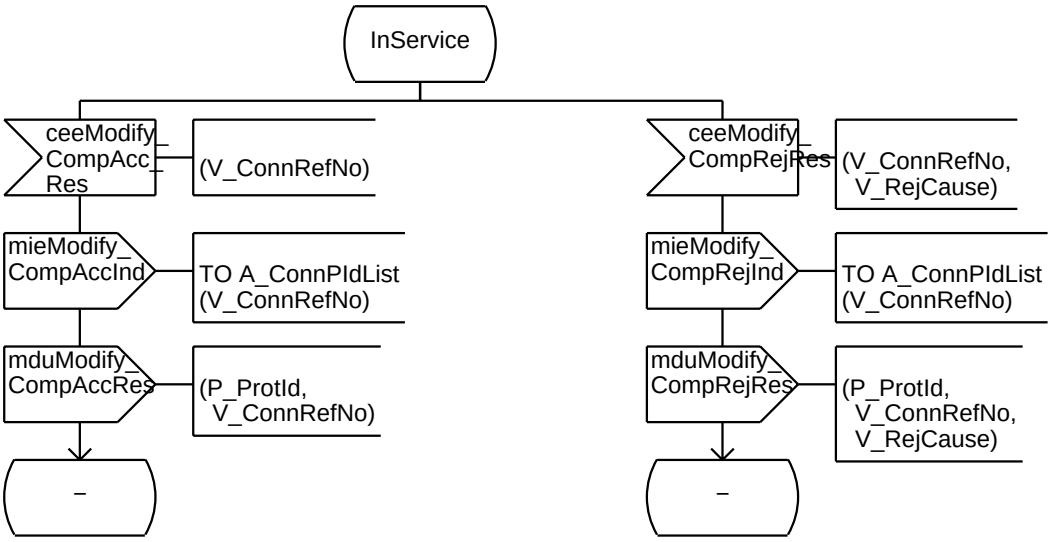

InService
mduDealloc_
Ind
(P_ProtId,
V_ConnRefNo)
mieDealloc_
Res
(V_ConnRefNo)
ceeDealloc_
Ind
(V_ConnRefNo)
–
A_ConnPIdList
(V_ConnRefNo)
=NULL
true
false
mduDealloc_
AccRes
(P_ProtId,
V_ConnRefNo)
mieDealloc_
Ind
(V_ConnRefNo)
TO A_ConnPIdList
(V_ConnRefNo)
–
–

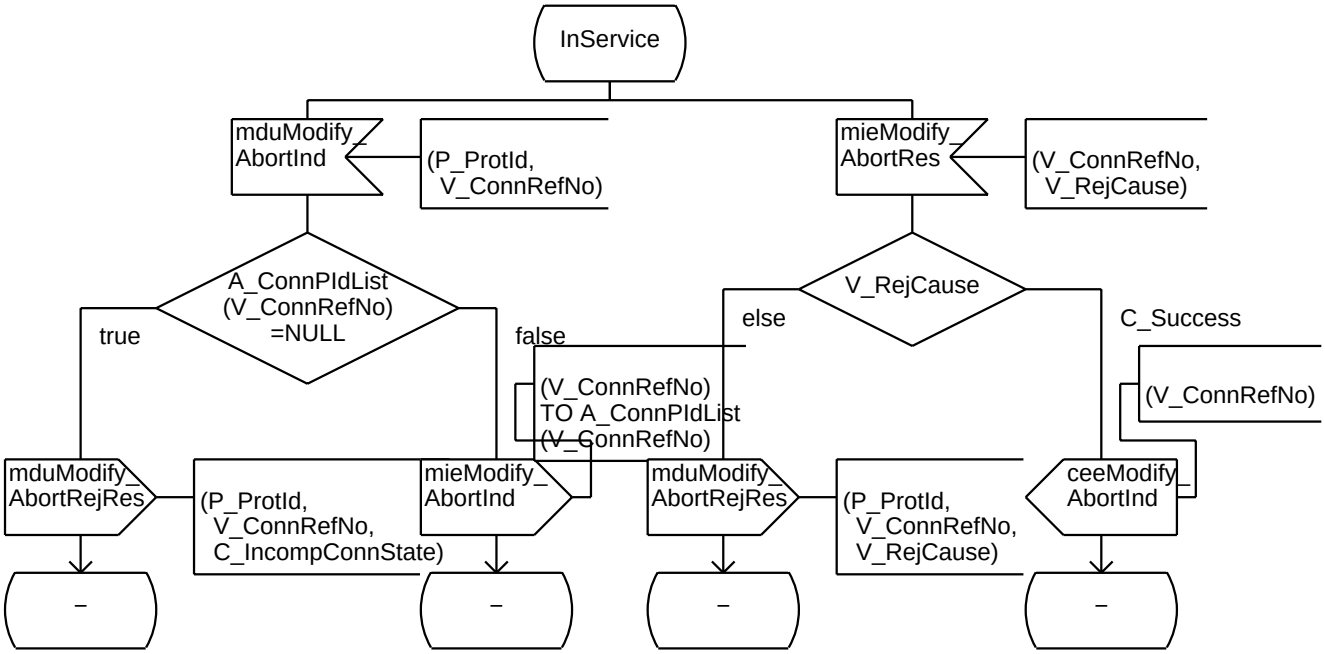

InService
ceeDealloc_
AccRes
(V_ConnRefNo)
mieDealloc_
AccRes
(V_ConnRefNo)
mieDealloc_
AccInd
(V_ConnRefNo)
TO A_ConnPIdList
(V_ConnRefNo)
mduDealloc_
AccRes
(P_ProtId,
V_ConnRefNo)
A_ConnPIdList
(V_ConnRefNo)
:= NULL
–
–

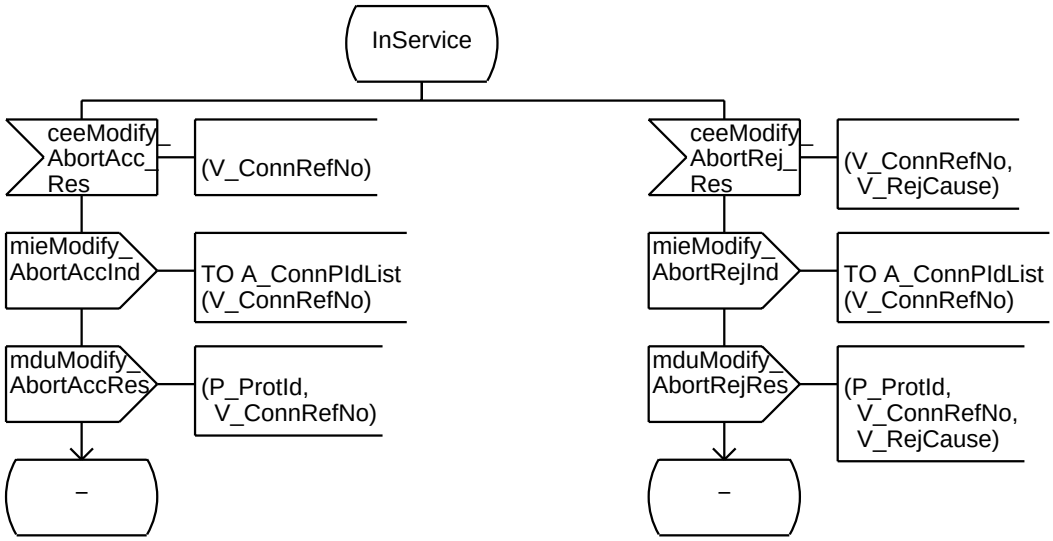

InService
mduModify_
Ind
(P_ProtId,
V_ConnRefNo)
mieModify_
Res
(V_ConnRefNo,
V_RejCause)
A_ConnPIdList
(V_ConnRefNo)
=NULL
true
false
V_RejCause
else
C_Success
(V_ConnRefNo)
TO A_ConnPIdList
(V_ConnRefNo)
mduModify_
RejRes
(P_ProtId,
V_ConnRefNo,
C_IncompConnState)
mieModify_
Ind
mduModify_
RejRes
(P_ProtId,
V_ConnRefNo,
V_RejCause)
ceeModify_
Ind
(V_ConnRefNo)
–
–
–
–

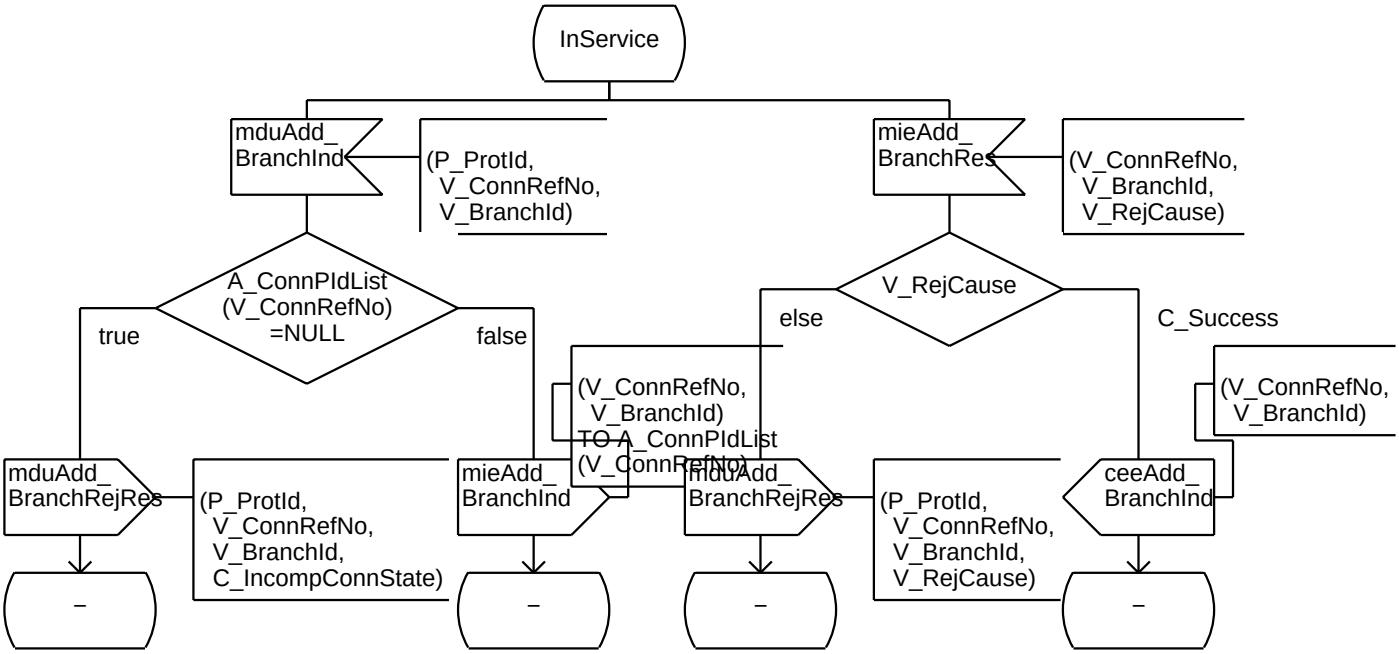

InService
ceeModify_
AccRes
(V_ConnRefNo)
ceeModify_
RejRes
(V_ConnRefNo,
V_RejCause)
mieModify_
AccInd
TO A_ConnPIdList
(V_ConnRefNo)
mieModify_
RejInd
TO A_ConnPIdList
(V_ConnRefNo)
mduModify_
AccRes
(P_ProtId,
V_ConnRefNo)
mduModify_
RejRes
(P_ProtId,
V_ConnRefNo,
V_RejCause)
–
–

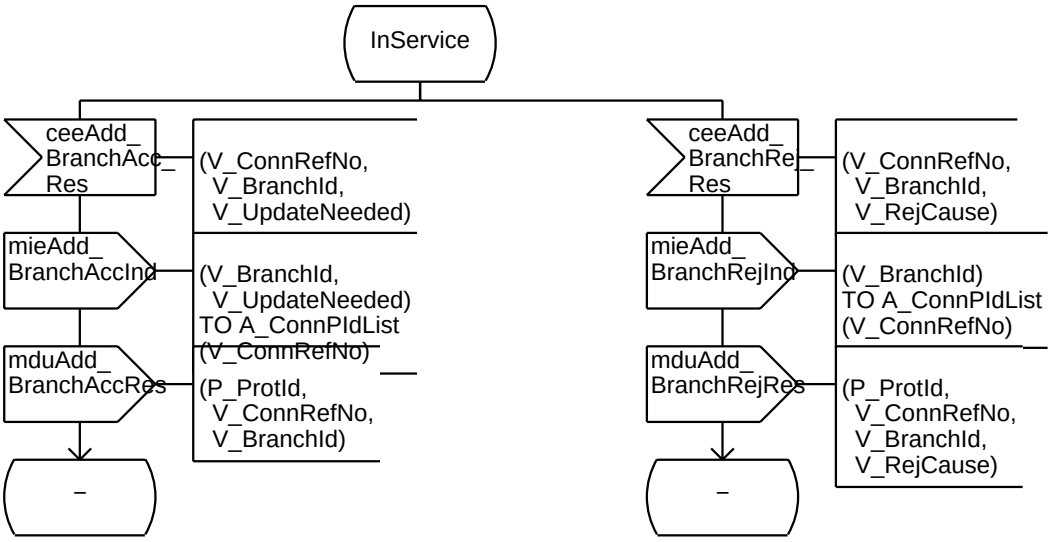

InService
mduModify_
CompInd
(P_ProtId,
V_ConnRefNo)
mieModify_
CompRes
(V_ConnRefNo,
V_RejCause)
A_ConnPIdList
(V_ConnRefNo)
=NULL
true
false
V_RejCause
else
C_Success
(V_ConnRefNo)
TO A_ConnPIdList
(V_ConnRefNo)
(V_ConnRefNo)
mduModify_
CompRejRes
(P_ProtId,
V_ConnRefNo,
C_IncompConnState)
mieModify_
CompInd
mduModify_
CompRejRes
(P_ProtId,
V_ConnRefNo,
V_RejCause)
ceeModify_
CompInd
–
–
–
–

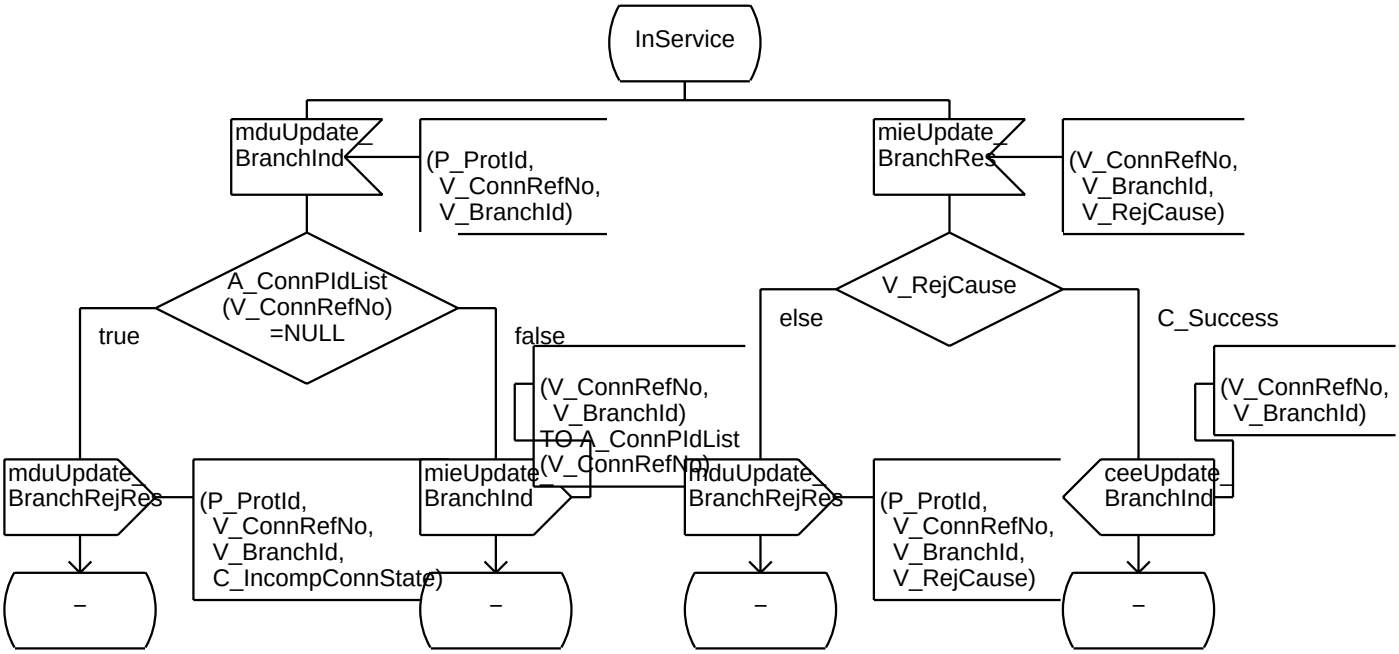

InService
ceeModify_
CompAcc_
Res
(V_ConnRefNo)
ceeModify_
CompRejRes
(V_ConnRefNo,
V_RejCause)
mieModify_
CompAccInd
TO A_ConnPIdList
(V_ConnRefNo)
mieModify_
CompRejInd
TO A_ConnPIdList
(V_ConnRefNo)
mduModify_
CompAccRes
(P_ProtId,
V_ConnRefNo)
mduModify_
CompRejRes
(P_ProtId,
V_ConnRefNo,
V_RejCause)
–
–

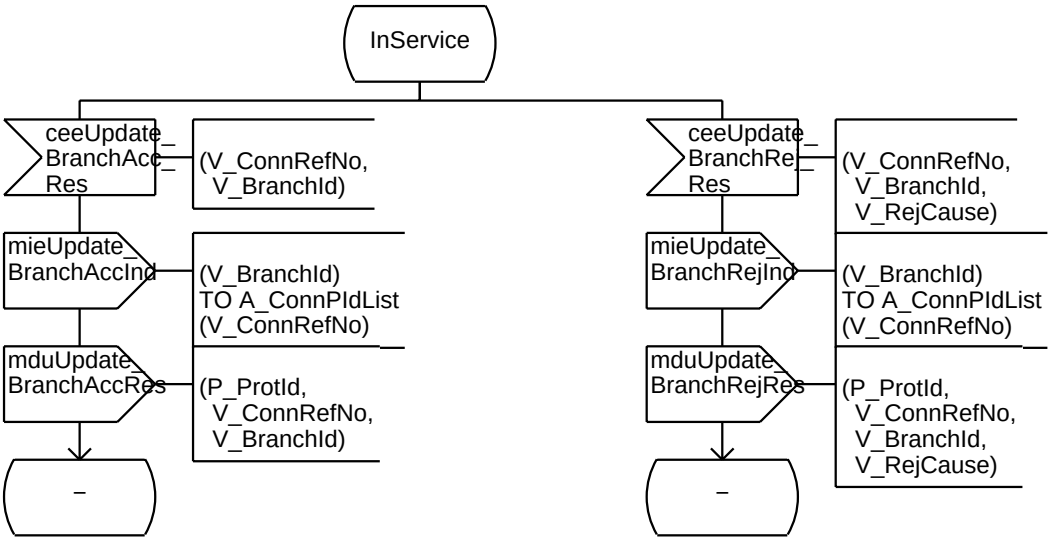

InService
mduModify_
AbortInd
(P_ProtId,
V_ConnRefNo)
mieModify_
AbortRes
(V_ConnRefNo,
V_RejCause)
A_ConnPIdList
(V_ConnRefNo)
=NULL
true
false
V_RejCause
else
C_Success
(V_ConnRefNo)
TO A_ConnPIdList
(V_ConnRefNo)
(V_ConnRefNo)
mduModify_
AbortRejRes
(P_ProtId,
V_ConnRefNo,
C_IncompConnState)
mieModify_
AbortInd
mduModify_
AbortRejRes
(P_ProtId,
V_ConnRefNo,
V_RejCause)
ceeModify_
AbortInd
–
–
–
–

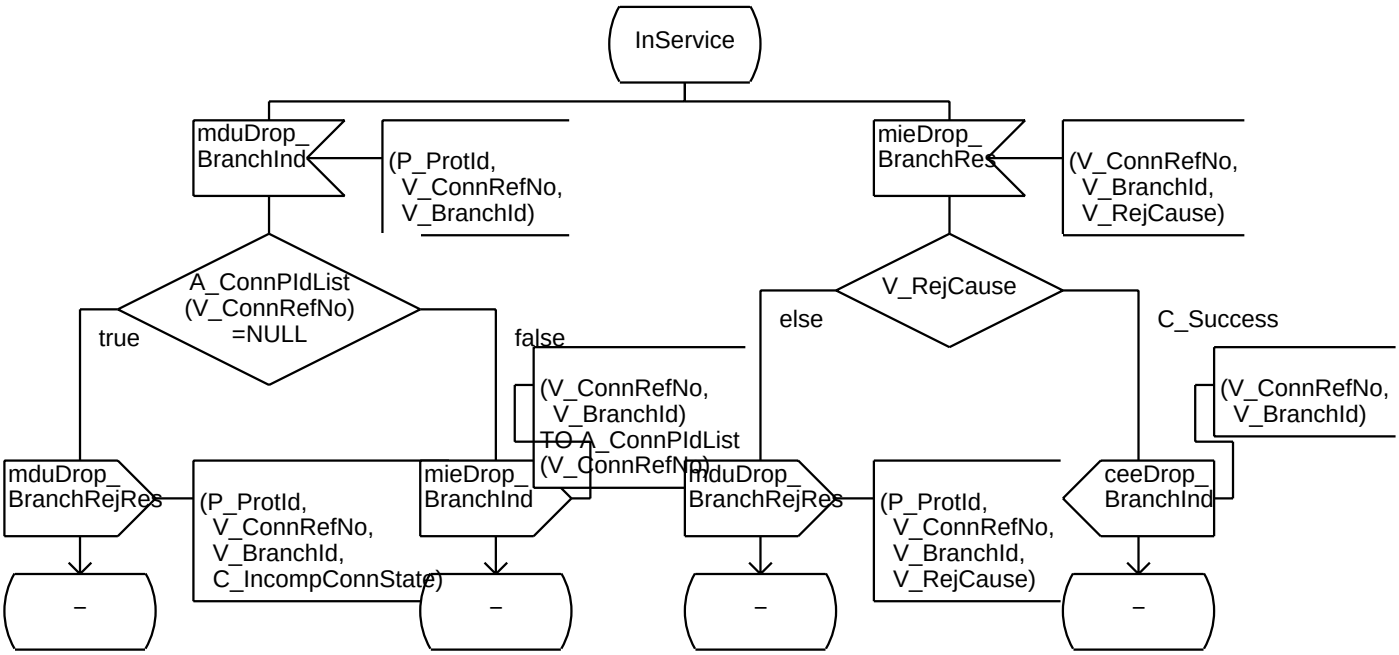

InService
ceeModify_
AbortAcc_
Res
(V_ConnRefNo)
ceeModify_
AbortRej_
Res
(V_ConnRefNo,
V_RejCause)
mieModify_
AbortAccInd
TO A_ConnPIdList
(V_ConnRefNo)
mieModify_
AbortRejInd
TO A_ConnPIdList
(V_ConnRefNo)
mduModify_
AbortAccRes
(P_ProtId,
V_ConnRefNo)
mduModify_
AbortRejRes
(P_ProtId,
V_ConnRefNo,
V_RejCause)
–
–

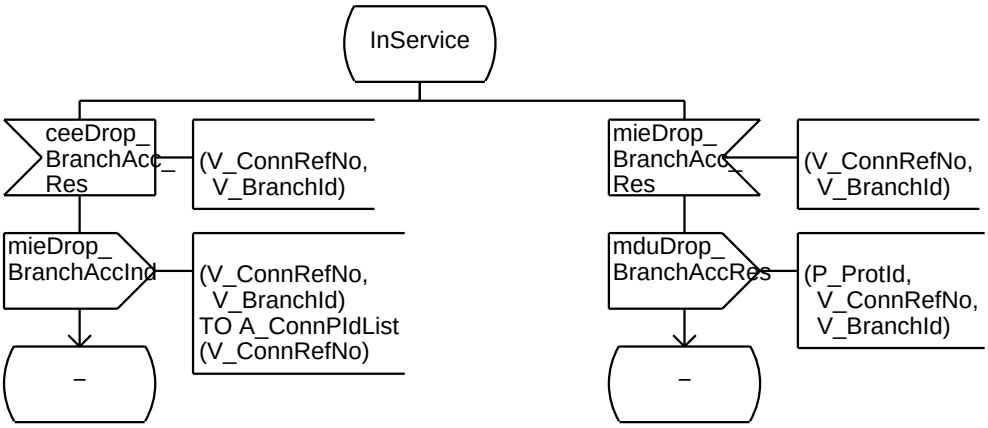

InService
mduAdd_
BranchInd
(P_ProtId,
V_ConnRefNo,
V_BranchId)
mieAdd_
BranchRes
(V_ConnRefNo,
V_BranchId,
V_RejCause)
A_ConnPIdList
(V_ConnRefNo)
=NULL
true
false
V_RejCause
else
C_Success
(V_ConnRefNo,
V_BranchId)
TO A_ConnPIdList
(V_ConnRefNo)
(V_ConnRefNo,
V_BranchId)
mduAdd_
BranchRejRes
(P_ProtId,
V_ConnRefNo,
V_BranchId,
C_IncompConnState)
mieAdd_
BranchInd
mduAdd_
BranchRejRes
(P_ProtId,
V_ConnRefNo,
V_BranchId,
V_RejCause)
ceeAdd_
BranchInd
–
–
–
–

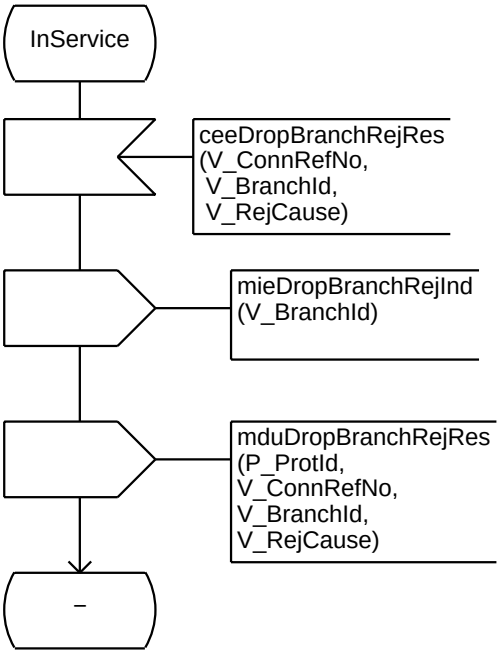

InService
ceeAdd_
BranchAcc_
Res
(V_ConnRefNo,
V_BranchId,
V_UpdateNeeded)
mieAdd_
BranchAccInd
(V_BranchId,
V_UpdateNeeded)
TO A_ConnPIdList
(V_ConnRefNo)
mduAdd_
BranchAccRes
(P_ProtId,
V_ConnRefNo,
V_BranchId)
–
ceeAdd_
BranchRej_
Res
(V_ConnRefNo,
V_BranchId,
V_RejCause)
mieAdd_
BranchRejInd
(V_BranchId)
TO A_ConnPIdList
(V_ConnRefNo)
mduAdd_
BranchRejRes
(P_ProtId,
V_ConnRefNo,
V_BranchId,
V_RejCause)
–


InService
mduUpdate_
BranchInd
(P_ProtId,
V_ConnRefNo,
V_BranchId)
mieUpdate_
BranchRes
(V_ConnRefNo,
V_BranchId,
V_RejCause)
A_ConnPIdList
(V_ConnRefNo)
=NULL
true
false
V_RejCause
else
C_Success
(V_ConnRefNo,
V_BranchId)
TO A_ConnPIdList
(V_ConnRefNo)
(V_ConnRefNo,
V_BranchId)
mduUpdate_
BranchRejRes
(P_ProtId,
V_ConnRefNo,
V_BranchId,
C_IncompConnState)
mieUpdate_
BranchInd
mduUpdate_
BranchRejRes
(P_ProtId,
V_ConnRefNo,
V_BranchId,
V_RejCause)
ceeUpdate_
BranchInd
–
–
–
–


InService
ceeUpdate_
BranchAcc_
Res
(V_ConnRefNo,
V_BranchId)
ceeUpdate_
BranchRej_
Res
(V_ConnRefNo,
V_BranchId,
V_RejCause)
mieUpdate_
BranchAccInd
(V_BranchId)
TO A_ConnPIdList
(V_ConnRefNo)
mieUpdate_
BranchRejInd
(V_BranchId)
TO A_ConnPIdList
(V_ConnRefNo)
mduUpdate_
BranchAccRes
(P_ProtId,
V_ConnRefNo,
V_BranchId)
mduUpdate_
BranchRejRes
(P_ProtId,
V_ConnRefNo,
V_BranchId,
V_RejCause)
–
–


InService
mduDrop_
BranchInd
(P_ProtId,
V_ConnRefNo,
V_BranchId)
mieDrop_
BranchRes
(V_ConnRefNo,
V_BranchId,
V_RejCause)
A_ConnPIdList
(V_ConnRefNo)
=NULL
true
false
V_RejCause
else
C_Success
(V_ConnRefNo,
V_BranchId)
TO A_ConnPIdList
(V_ConnRefNo)
(V_ConnRefNo,
V_BranchId)
mduDrop_
BranchRejRes
(P_ProtId,
V_ConnRefNo,
V_BranchId,
C_IncompConnState)
mieDrop_
BranchInd
mduDrop_
BranchRejRes
(P_ProtId,
V_ConnRefNo,
V_BranchId,
V_RejCause)
ceeDrop_
BranchInd
–
–
–
–


InService
ceeDrop_
BranchAcc_
Res
(V_ConnRefNo,
V_BranchId)
mieDrop_
BranchAcc_
Res
(V_ConnRefNo,
V_BranchId)
mieDrop_
BranchAccInd
(V_ConnRefNo,
V_BranchId)
TO A_ConnPIdList
(V_ConnRefNo)
mduDrop_
BranchAccRes
(P_ProtId,
V_ConnRefNo,
V_BranchId)
–
–


InService
ceeDropBranchRejRes
(V_ConnRefNo,
V_BranchId,
V_RejCause)
mieDropBranchRejInd
(V_BranchId)
mduDropBranchRejRes
(P_ProtId,
V_ConnRefNo,
V_BranchId,
V_RejCause)
–

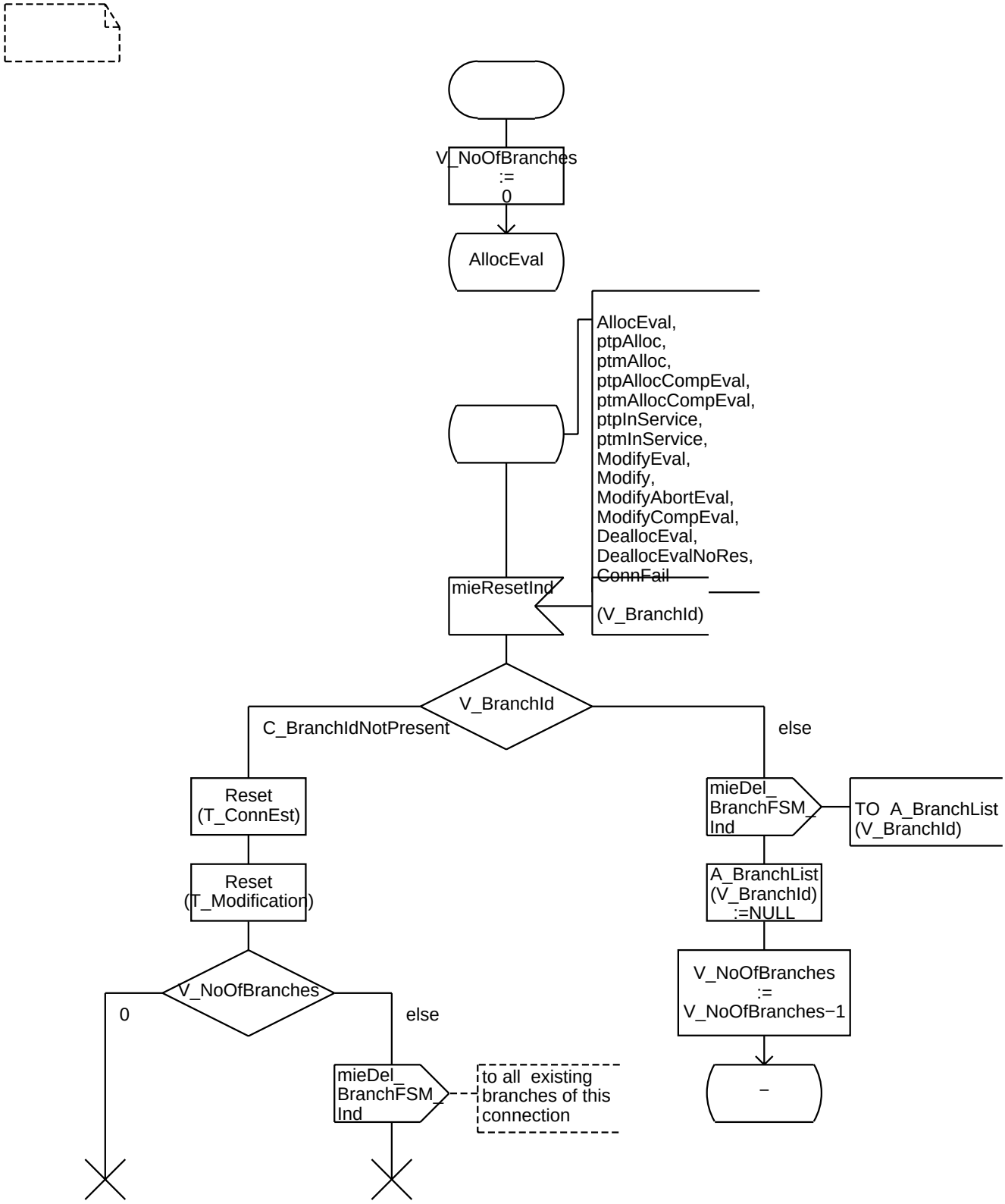
```
/* State descriptions
AllocEval:        An AllocReq is received but not evaluated yet
ptpAlloc:         Resources for a ptp connection are reserved (allocated) but the connection is not established yet
ptmAlloc:         Resources for a ptm connection are reserved (allocated) but the connection is not established yet
ptpAllocCompEval: An AllocCompReq for a ptp connection is received but not evaluated yet
ptmAllocCompEval: An AllocCompReq for a ptm connection is received but not evaluated yet
ptpInService:     A ptp connection is established
ptmInService:     A ptm connection is established
ModifyEval:       A ModifyReq for a ptp connection is received but not evaluated yet
Modify:           A ptp connection is prepared for modification, but the modification is not completed yet
ModifyAbortEval:  A ModifyAbortReq for a ptp connection is received but not evaluated yet
ModifyCompEval:   A ModifyCompReq for a ptp connection is received but not evaluated yet
DeallocEval:      A DeallocReq for a connection is received but not evaluated yet
DeallocEvalNoRes: A deallocation of the connection is requested, but no response shall be given in case of indication
ConnFail:         Failure state, only dealloc or reset is accepted
*/
```

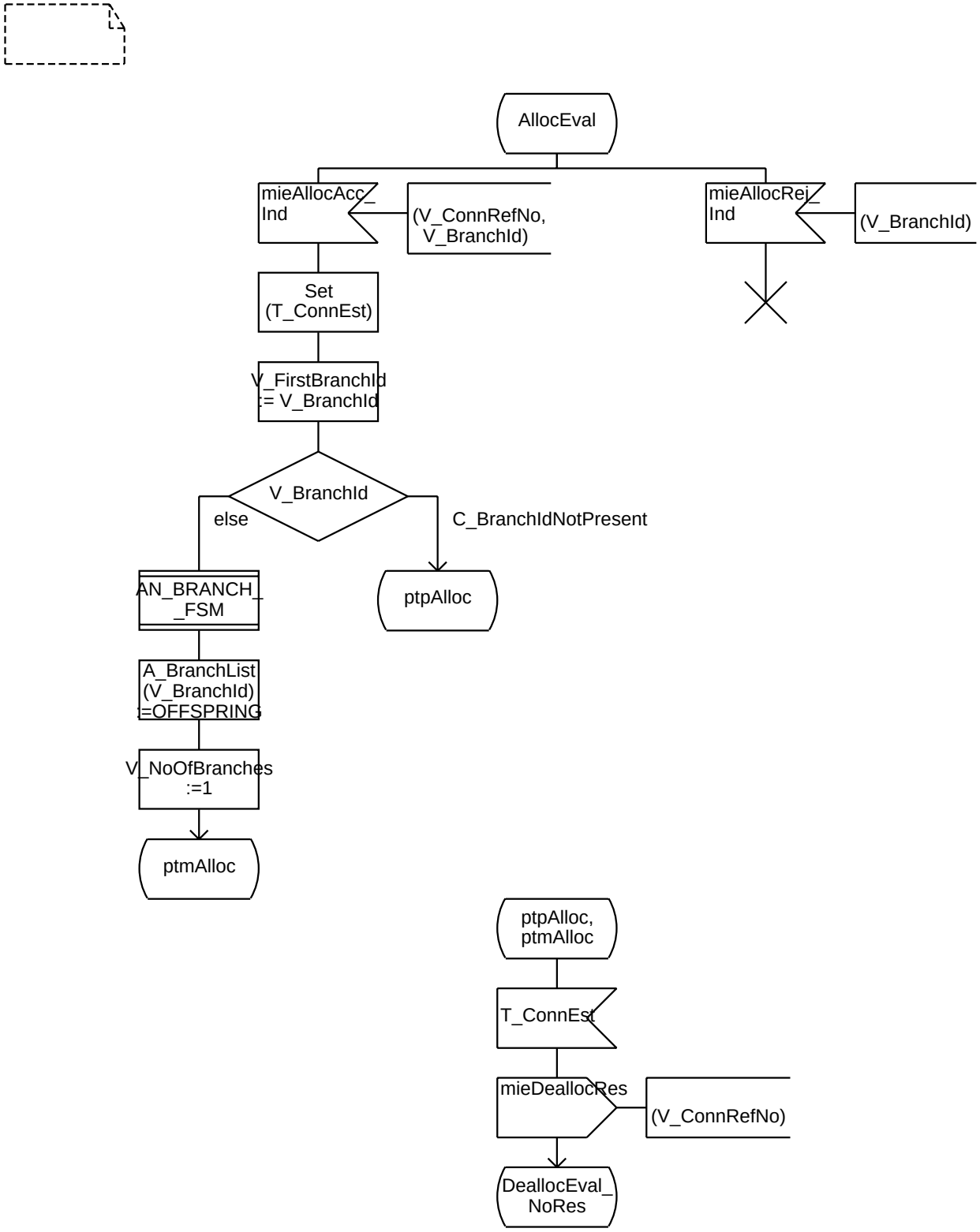
```
/* Timerdefinitions for supervision of applications */
TIMER
T_Modification  := 180,   /* Default value for modification supervision timer is 180 sec, Tolerance +/− 10%  */
T_ConnEst       := 180;   /* Default value for allocation supervision timer is 180 sec, Tolerance +/− 10%  */
```

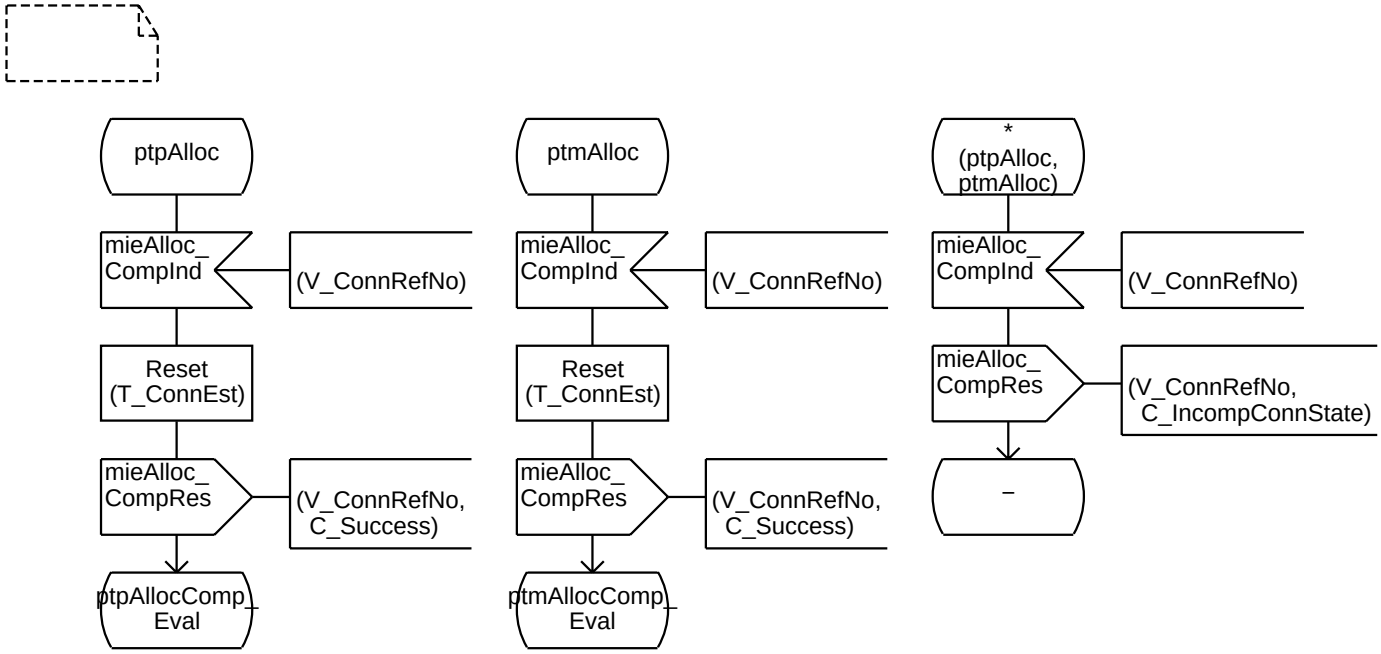
```
/* Data type definitions internal to the process */
/**/
/* This data type is used for translation of the branch identifier to the process identifier (PID)
   of the branch FSM */
NEWTYPE AT_BranchList ARRAY
(IT_BranchId,PID)
ENDNEWTYPE AT_BranchList;
```

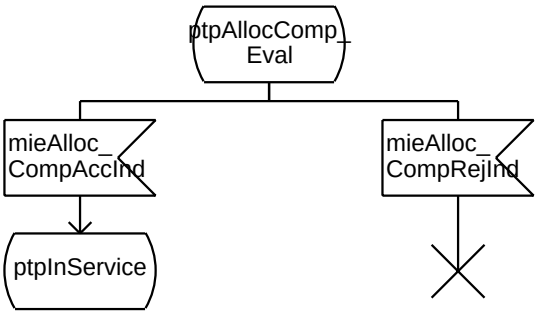
```
/*Primitive Data Declarations*/
/**/
/* Connection identifier */
DCL
V_ConnRefNo  IT_ConnRefNo;
/**/
/* Branch identifier */
DCL
V_BranchId  IT_BranchId;
/**/
/* Reject Cause */
DCL
V_RejCause   IT_RejCause;
/**/
/* Update needed for addBranch */
DCL
V_UpdateNeeded  IT_UpdateNeeded;
```

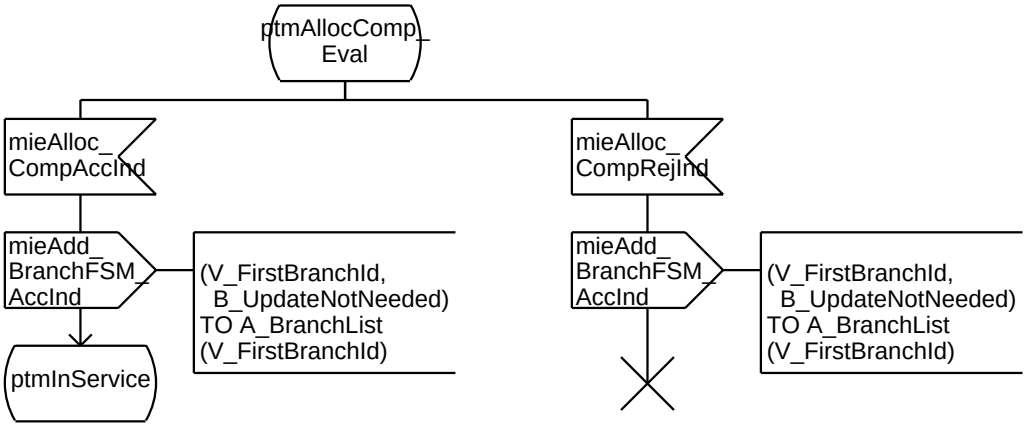
```
/* Process Data Declarations */
/**/
/* List of connection FSMs */
DCL
A_BranchList   AT_BranchList;
/**/
/* Number of branches */
DCL
V_NoOfBranches   INTEGER;
/**/
/* First branch */
DCL
V_FirstBranchId IT_BranchId;
```

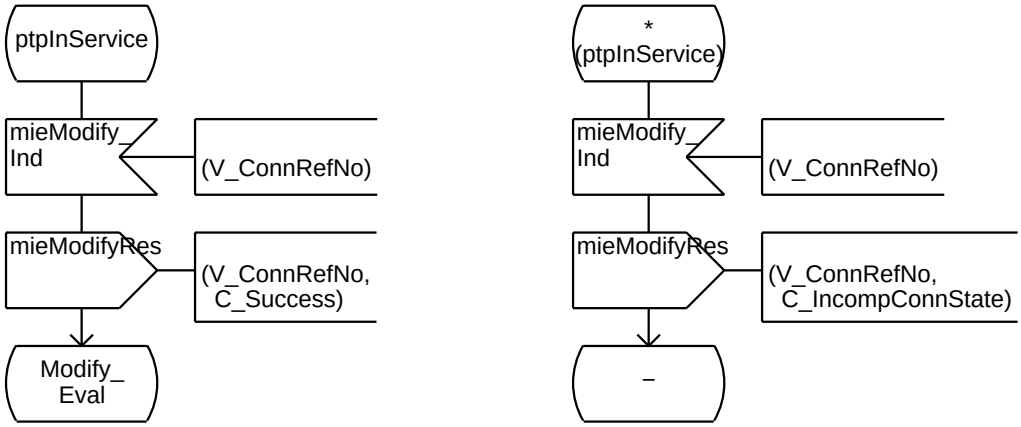

V_NoOfBranches
:=
0
AllocEval
AllocEval,
ptpAlloc,
ptmAlloc,
ptpAllocCompEval,
ptmAllocCompEval,
ptpInService,
ptmInService,
ModifyEval,
Modify,
ModifyAbortEval,
ModifyCompEval,
DeallocEval,
DeallocEvalNoRes,
ConnFail
mieResetInd
(V_BranchId)
V_BranchId
C_BranchIdNotPresent
else
Reset
(T_ConnEst)
Reset
(T_Modification)
V_NoOfBranches
0
else
mieDel_
BranchFSM_
Ind
to all existing
branches of this
connection
mieDel_
BranchFSM_
Ind
TO A_BranchList
(V_BranchId)
A_BranchList
(V_BranchId)
:=NULL
V_NoOfBranches
:=
V_NoOfBranches−1
–

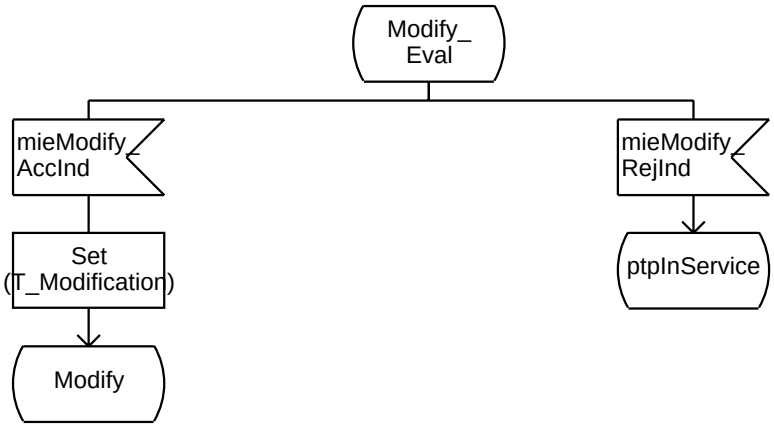

AllocEval
mieAllocAcc_
Ind
(V_ConnRefNo,
V_BranchId)
mieAllocRej_
Ind
(V_BranchId)
Set
(T_ConnEst)
V_FirstBranchId
:= V_BranchId
V_BranchId
else
C_BranchIdNotPresent
AN_BRANCH_
_FSM
ptpAlloc
A_BranchList
(V_BranchId)
!=OFFSPRING
V_NoOfBranches
:=1
ptmAlloc
ptpAlloc,
ptmAlloc
T_ConnEst
mieDeallocRes
(V_ConnRefNo)
DeallocEval_
NoRes

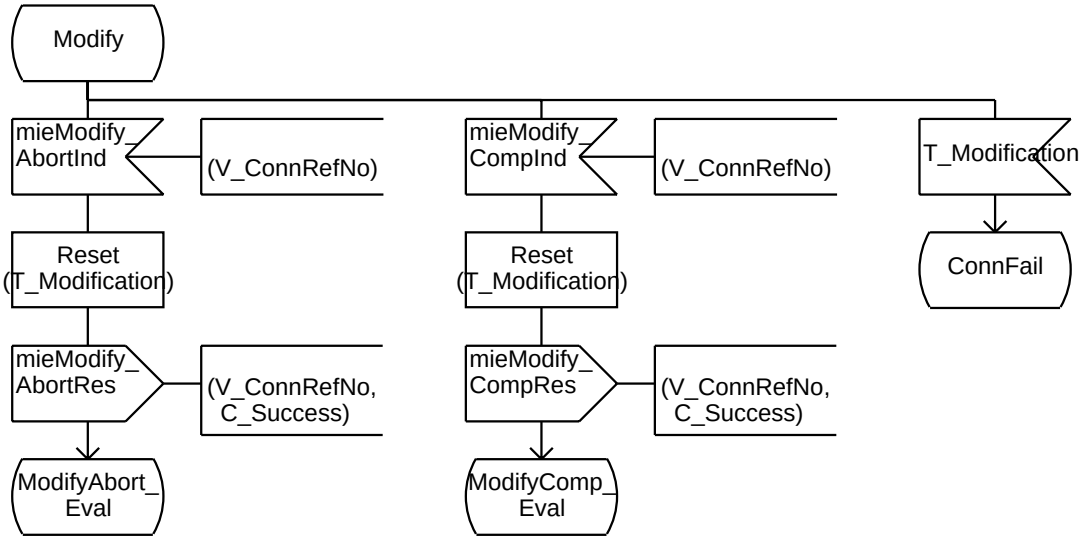

ptpAlloc
mieAlloc_
CompInd
(V_ConnRefNo)
Reset
(T_ConnEst)
mieAlloc_
CompRes
(V_ConnRefNo,
C_Success)
ptpAllocComp_
Eval
ptmAlloc
mieAlloc_
CompInd
(V_ConnRefNo)
Reset
(T_ConnEst)
mieAlloc_
CompRes
(V_ConnRefNo,
C_Success)
ptmAllocComp_
Eval
*
(ptpAlloc,
ptmAlloc)
mieAlloc_
CompInd
(V_ConnRefNo)
mieAlloc_
CompRes
(V_ConnRefNo,
C_IncompConnState)
–

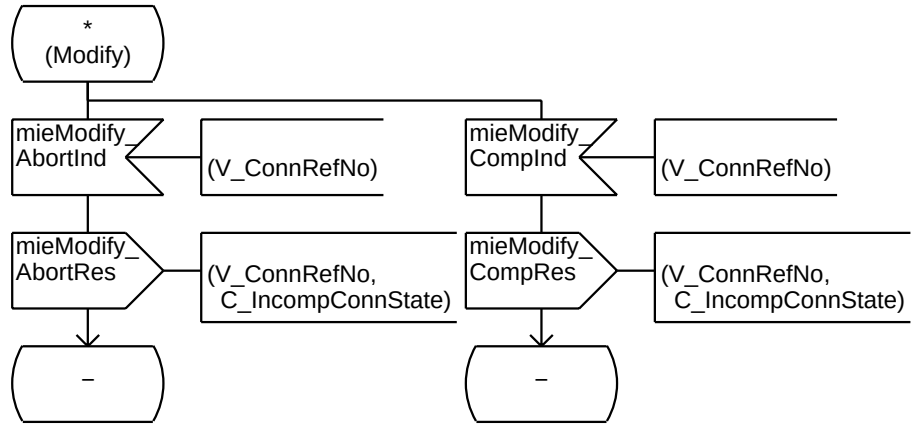

ptpAllocComp_
Eval
mieAlloc_
CompAccInd
mieAlloc_
CompRejInd
ptpInService

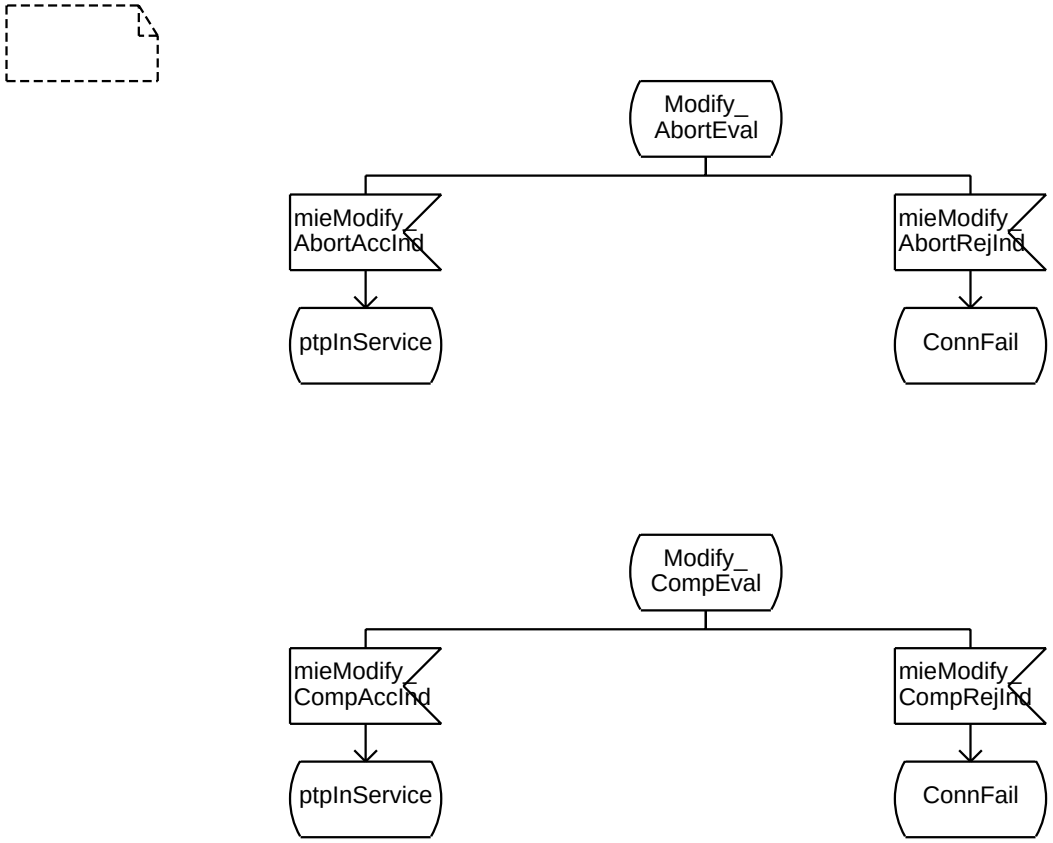

ptmAllocComp_
Eval
mieAlloc_
CompAccInd
mieAlloc_
CompRejInd
mieAdd_
BranchFSM_
AccInd
(V_FirstBranchId,
B_UpdateNotNeeded)
TO A_BranchList
(V_FirstBranchId)
ptmInService
mieAdd_
BranchFSM_
AccInd
(V_FirstBranchId,
B_UpdateNotNeeded)
TO A_BranchList
(V_FirstBranchId)

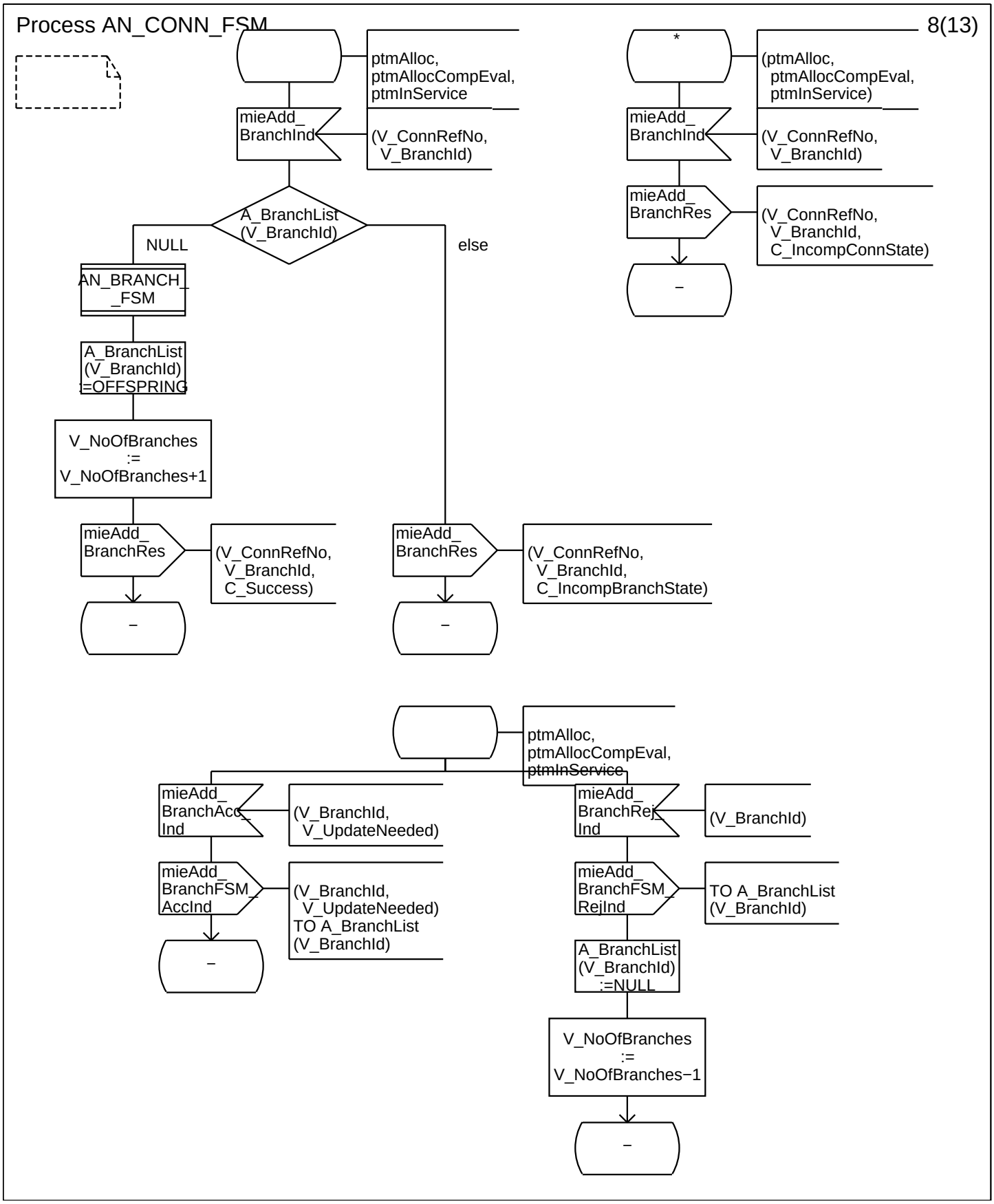

ptpInService
mieModify_
Ind
(V_ConnRefNo)
mieModifyRes
(V_ConnRefNo,
C_Success)
Modify_
Eval
*
(ptpInService)
mieModify_
Ind
(V_ConnRefNo)
mieModifyRes
(V_ConnRefNo,
C_IncompConnState)
_

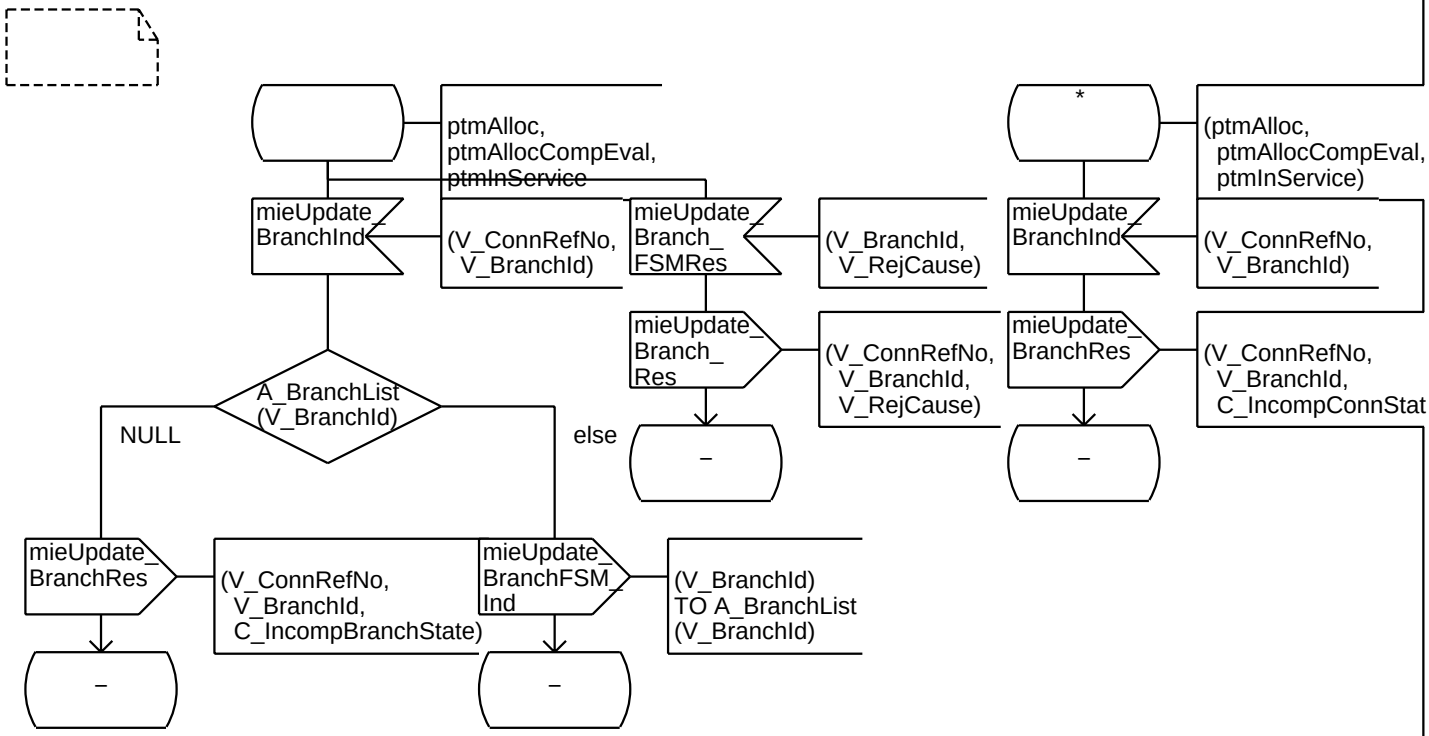

Modify_
Eval
mieModify_
AccInd
mieModify_
RejInd
Set
(T_Modification)
ptpInService
Modify

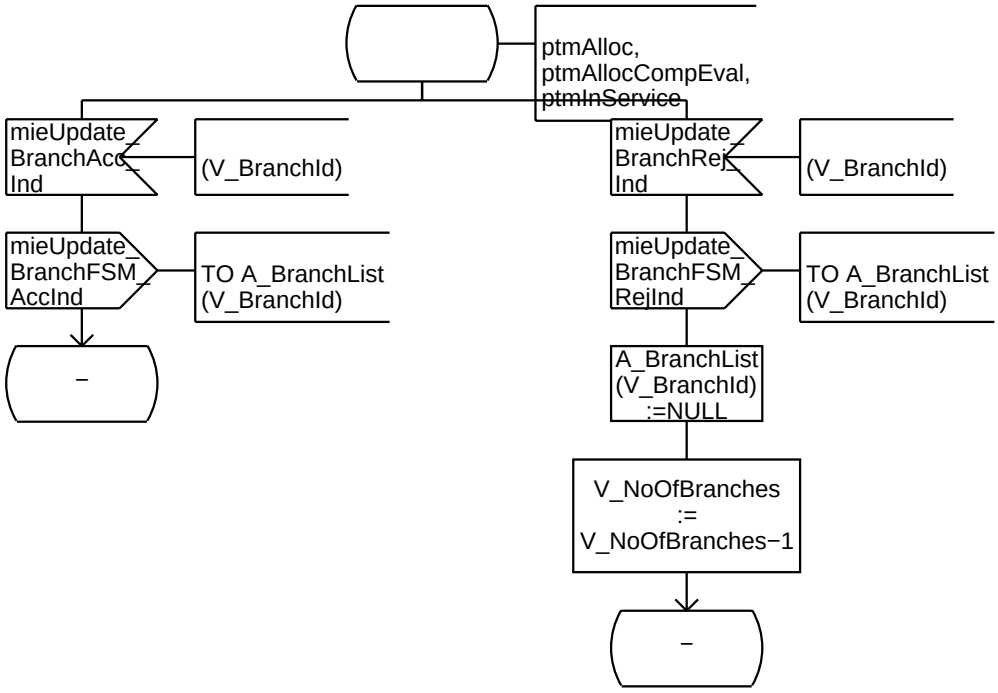

Modify
mieModify_
AbortInd
(V_ConnRefNo)
mieModify_
CompInd
(V_ConnRefNo)
T_Modification
Reset
(T_Modification)
Reset
(T_Modification)
ConnFail
mieModify_
AbortRes
(V_ConnRefNo,
C_Success)
mieModify_
CompRes
(V_ConnRefNo,
C_Success)
ModifyAbort_
Eval
ModifyComp_
Eval

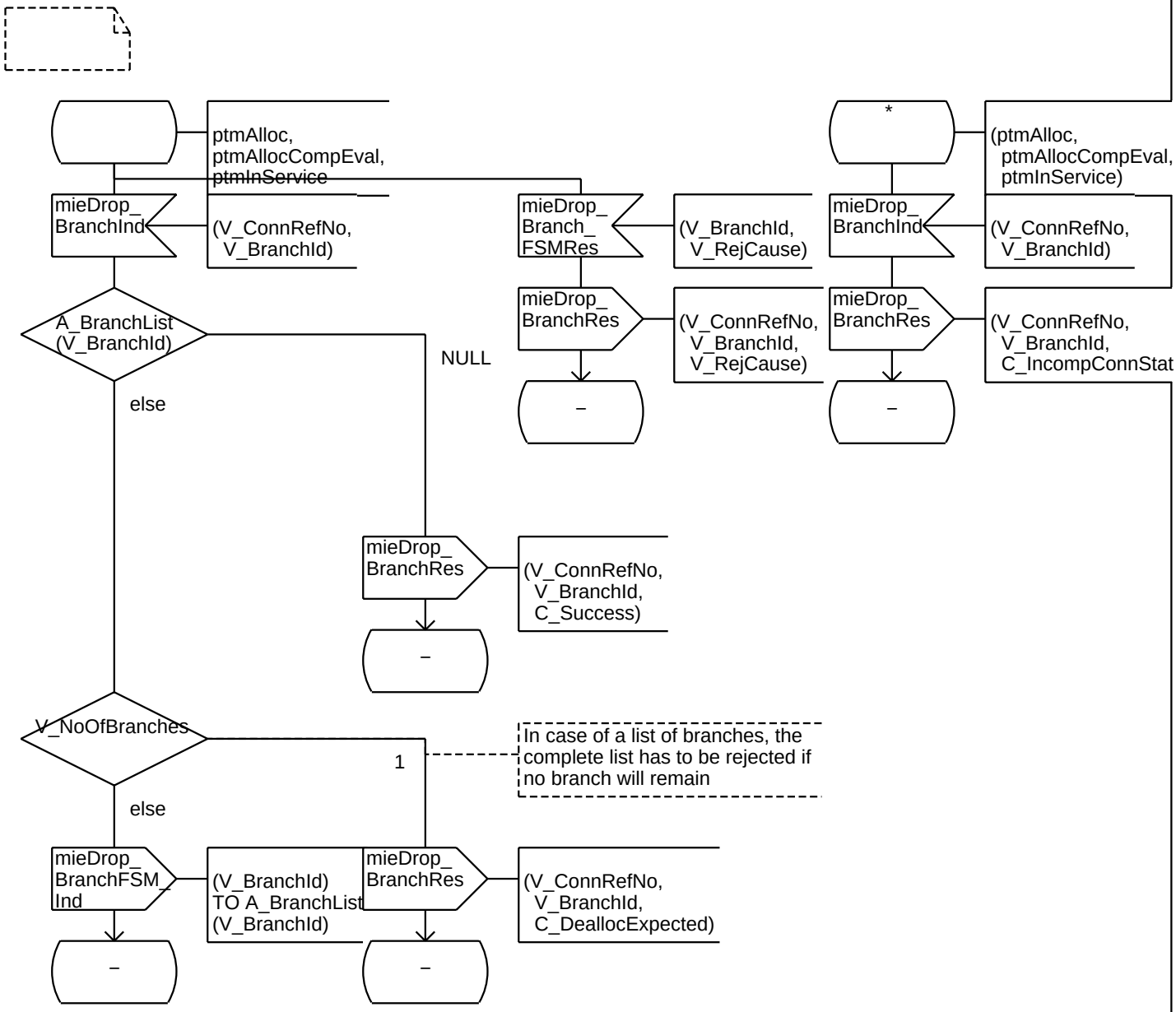

*
(Modify)
mieModify_
AbortInd
(V_ConnRefNo)
mieModify_
CompInd
(V_ConnRefNo)
mieModify_
AbortRes
(V_ConnRefNo,
C_IncompConnState)
mieModify_
CompRes
(V_ConnRefNo,
C_IncompConnState)
–
–

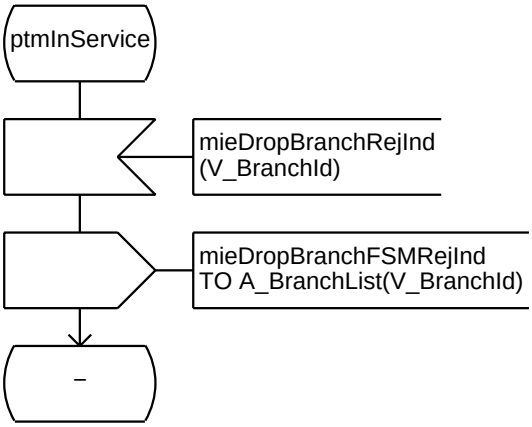

Modify_
AbortEval
mieModify_
AbortAccInd
mieModify_
AbortRejInd
ptpInService
ConnFail
Modify_
CompEval
mieModify_
CompAccInd
mieModify_
CompRejInd
ptpInService
ConnFail

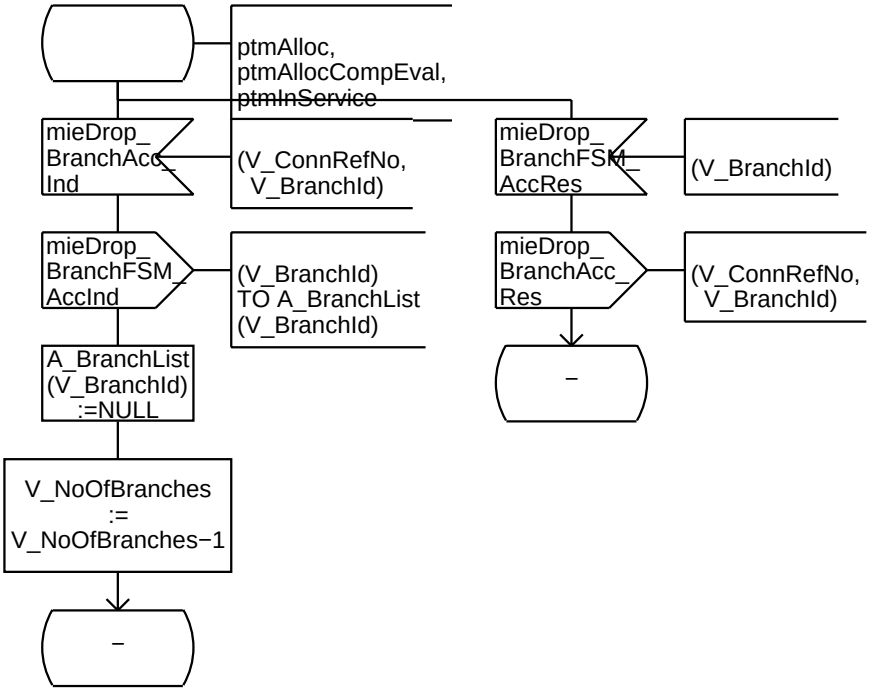

ptmAlloc,
ptmAllocCompEval,
ptmInService
mieAdd_
BranchInd
(V_ConnRefNo,
V_BranchId)
A_BranchList
(V_BranchId)
NULL
else
AN_BRANCH_
_FSM
A_BranchList
(V_BranchId)
:=OFFSPRING
V_NoOfBranches
:=
V_NoOfBranches+1
mieAdd_
BranchRes
(V_ConnRefNo,
V_BranchId,
C_Success)
–
mieAdd_
BranchRes
(V_ConnRefNo,
V_BranchId,
C_IncompBranchState)
–
*
(ptmAlloc,
ptmAllocCompEval,
ptmInService)
mieAdd_
BranchInd
(V_ConnRefNo,
V_BranchId)
mieAdd_
BranchRes
(V_ConnRefNo,
V_BranchId,
C_IncompConnState)
–
ptmAlloc,
ptmAllocCompEval,
ptmInService
mieAdd_
BranchAcc_
Ind
(V_BranchId,
V_UpdateNeeded)
mieAdd_
BranchFSM_
AccInd
(V_BranchId,
V_UpdateNeeded)
TO A_BranchList
(V_BranchId)
–
mieAdd_
BranchRej_
Ind
(V_BranchId)
mieAdd_
BranchFSM_
RejInd
TO A_BranchList
(V_BranchId)
A_BranchList
(V_BranchId)
:=NULL
V_NoOfBranches
:=
V_NoOfBranches−1
–

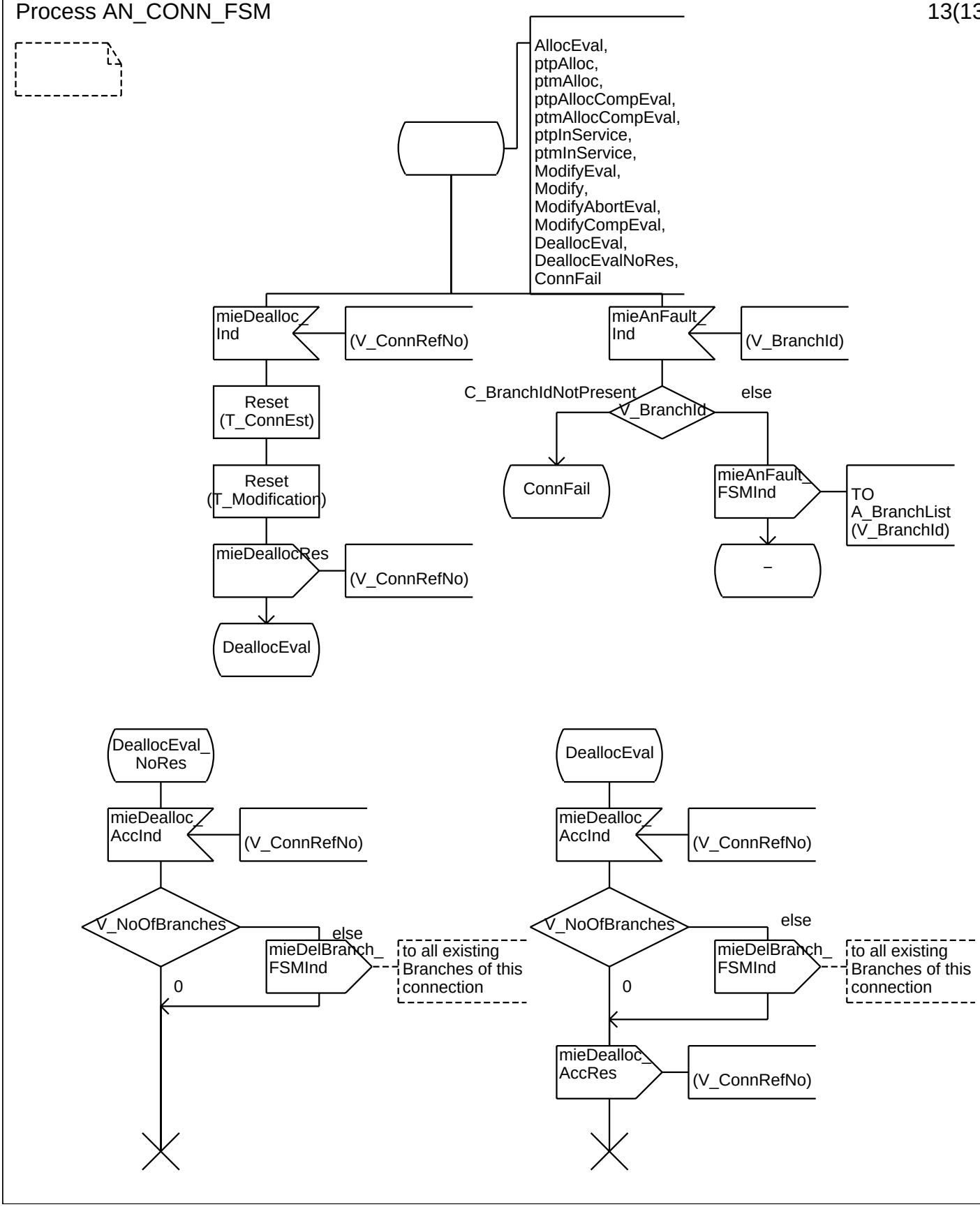

ptmAlloc,
ptmAllocCompEval,
ptmInService
mieUpdate_
BranchInd
(V_ConnRefNo,
V_BranchId)
mieUpdate_
Branch_
FSMRes
(V_BranchId,
V_RejCause)
mieUpdate_
Branch_
Res
(V_ConnRefNo,
V_BranchId,
V_RejCause)
–
*
(ptmAlloc,
ptmAllocCompEval,
ptmInService)
mieUpdate_
BranchInd
(V_ConnRefNo,
V_BranchId)
mieUpdate_
BranchRes
(V_ConnRefNo,
V_BranchId,
C_IncompConnStat
–
A_BranchList
(V_BranchId)
NULL
else
mieUpdate_
BranchRes
(V_ConnRefNo,
V_BranchId,
C_IncompBranchState)
mieUpdate_
BranchFSM_
Ind
(V_BranchId)
TO A_BranchList
(V_BranchId)
–
–


ptmAlloc,
ptmAllocCompEval,
ptmInService
mieUpdate_
BranchAcc_
Ind
(V_BranchId)
mieUpdate_
BranchFSM_
AccInd
TO A_BranchList
(V_BranchId)
–
mieUpdate_
BranchRej_
Ind
(V_BranchId)
mieUpdate_
BranchFSM_
RejInd
TO A_BranchList
(V_BranchId)
A_BranchList
(V_BranchId)
:=NULL
V_NoOfBranches
:=
V_NoOfBranches−1
–


ptmAlloc,
ptmAllocCompEval,
ptmInService
mieDrop_
BranchInd
(V_ConnRefNo,
V_BranchId)
A_BranchList
(V_BranchId)
NULL
else
mieDrop_
Branch_
FSMRes
(V_BranchId,
V_RejCause)
mieDrop_
BranchRes
(V_ConnRefNo,
V_BranchId,
V_RejCause)
–
*
(ptmAlloc,
ptmAllocCompEval,
ptmInService)
mieDrop_
BranchInd
(V_ConnRefNo,
V_BranchId)
mieDrop_
BranchRes
(V_ConnRefNo,
V_BranchId,
C_IncompConnStat
–
mieDrop_
BranchRes
(V_ConnRefNo,
V_BranchId,
C_Success)
–
V_NoOfBranches
1
In case of a list of branches, the
complete list has to be rejected if
no branch will remain
else
mieDrop_
BranchFSM_
Ind
(V_BranchId)
TO A_BranchList
(V_BranchId)
mieDrop_
BranchRes
(V_ConnRefNo,
V_BranchId,
C_DeallocExpected)
–
–


ptmInService
mieDropBranchRejInd
(V_BranchId)
mieDropBranchFSMRejInd
TO A_BranchList(V_BranchId)
–

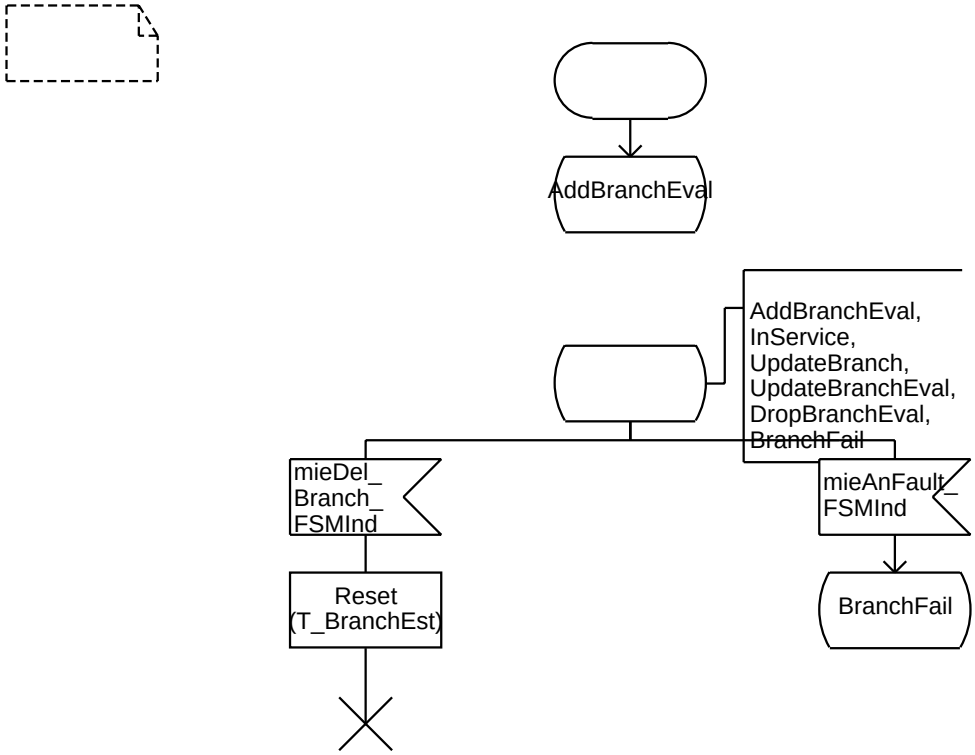

ptmAlloc,
ptmAllocCompEval,
ptmInService
mieDrop_
BranchAcc_
Ind
(V_ConnRefNo,
V_BranchId)
mieDrop_
BranchFSM_
AccInd
(V_BranchId)
TO A_BranchList
(V_BranchId)
A_BranchList
(V_BranchId)
:=NULL
V_NoOfBranches
:=
V_NoOfBranches−1
–
mieDrop_
BranchFSM_
AccRes
(V_BranchId)
mieDrop_
BranchAcc_
Res
(V_ConnRefNo,
V_BranchId)
–

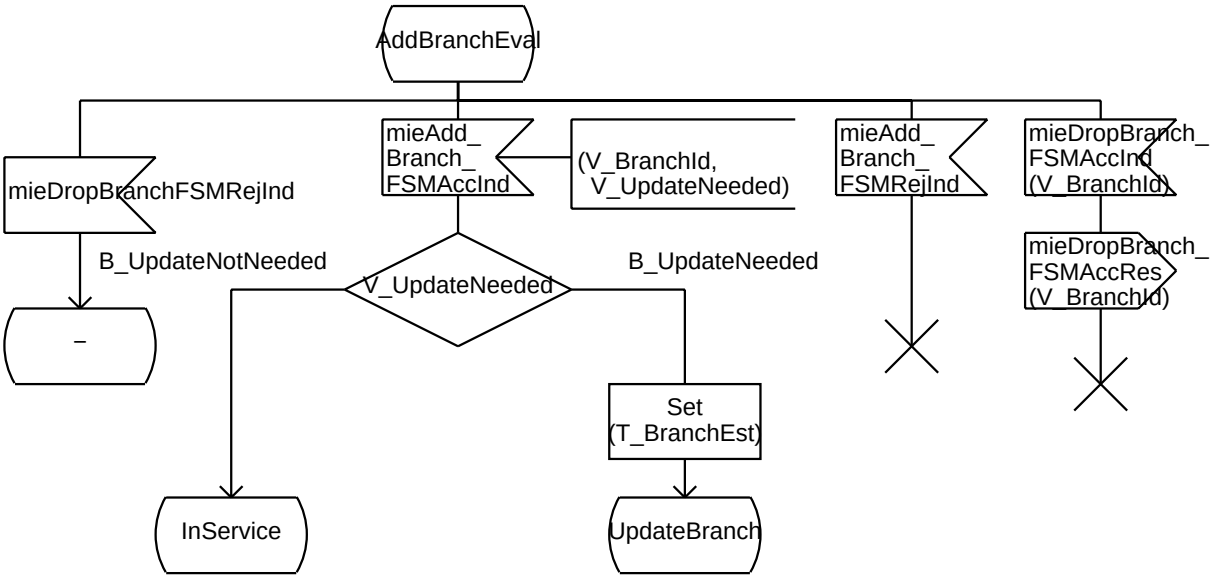

AllocEval,
ptpAlloc,
ptmAlloc,
ptpAllocCompEval,
ptmAllocCompEval,
ptpInService,
ptmInService,
ModifyEval,
Modify,
ModifyAbortEval,
ModifyCompEval,
DeallocEval,
DeallocEvalNoRes,
ConnFail
mieDealloc_Ind
(V_ConnRefNo)
mieAnFault_Ind
(V_BranchId)
Reset (T_ConnEst)
Reset (T_Modification)
mieDeallocRes
(V_ConnRefNo)
DeallocEval
C_BranchIdNotPresent
V_BranchId
else
ConnFail
mieAnFault_FSMInd
TO A_BranchList (V_BranchId)
–
DeallocEval_NoRes
mieDealloc_AccInd
(V_ConnRefNo)
V_NoOfBranches
else
mieDelBranch_FSMInd
to all existing Branches of this connection
0
DeallocEval
mieDealloc_AccInd
(V_ConnRefNo)
V_NoOfBranches
else
mieDelBranch_FSMInd
to all existing Branches of this connection
0
mieDealloc_AccRes
(V_ConnRefNo)

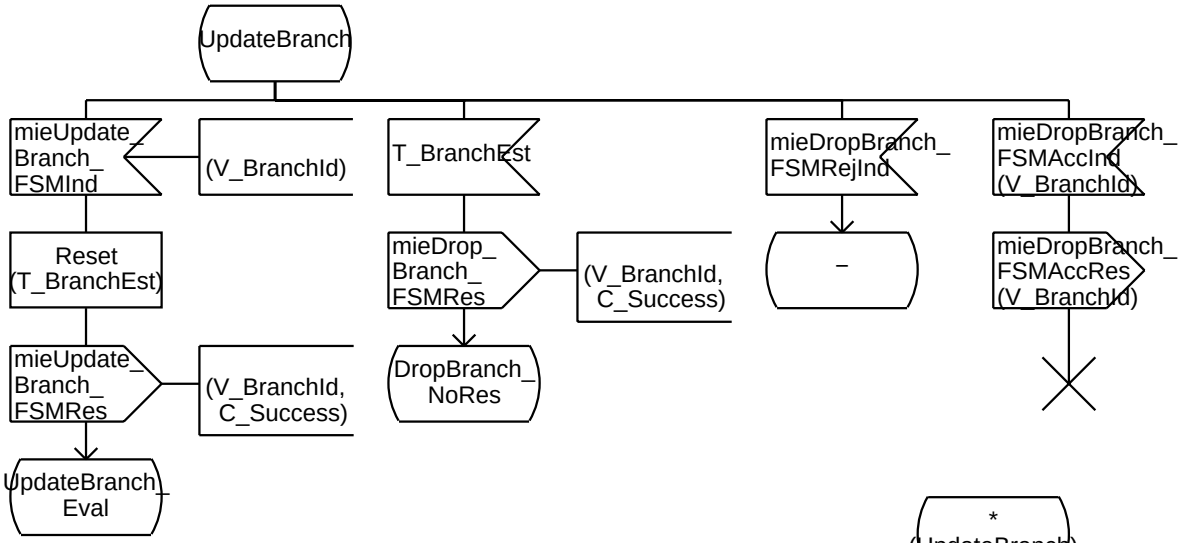
```
/* State descriptions
AddBranchEval:    Init State, this branch of the connection is allocated
InService:        The branch is established, i.e the allocation is completed
UpdateBranch:     The branch has to be updated
UpdateBranchEval: The update branch message is received but not evaluated yet
DropBranchEval:   The drop branch message is received but not evaluated yet
BranchFail:       Failure state, only dealloc message is accepted
*/
```

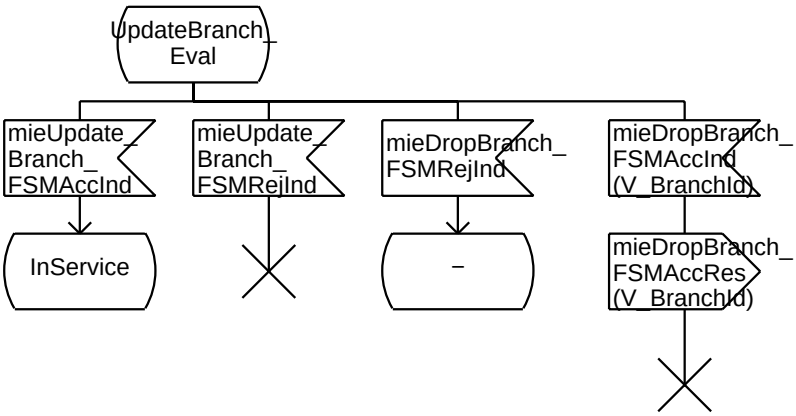
```
/* Timerdefinitions for supervision of application */
TIMER
T_BranchEst       := 180;   /* Default value for update branch supervision timer is 180 sec, Tolerance +/− 10%  */
```

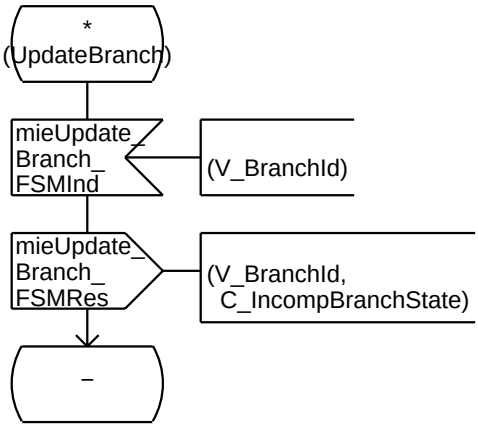
```
/*Primitive Data Declarations*/
/**/
/* Branch identifier */
DCL
V_BranchId  IT_BranchId;
/**/
/* Update needed for addBranch */
DCL
V_UpdateNeeded  IT_UpdateNeeded;
```

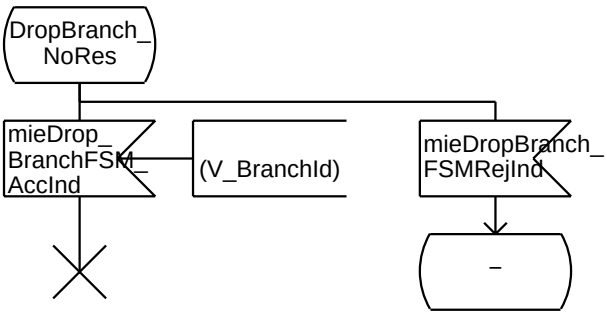

AddBranchEval
AddBranchEval,
InService,
UpdateBranch,
UpdateBranchEval,
DropBranchEval,
BranchFail
mieDel_
Branch_
FSMInd
mieAnFault_
FSMInd
Reset
(T_BranchEst)
BranchFail

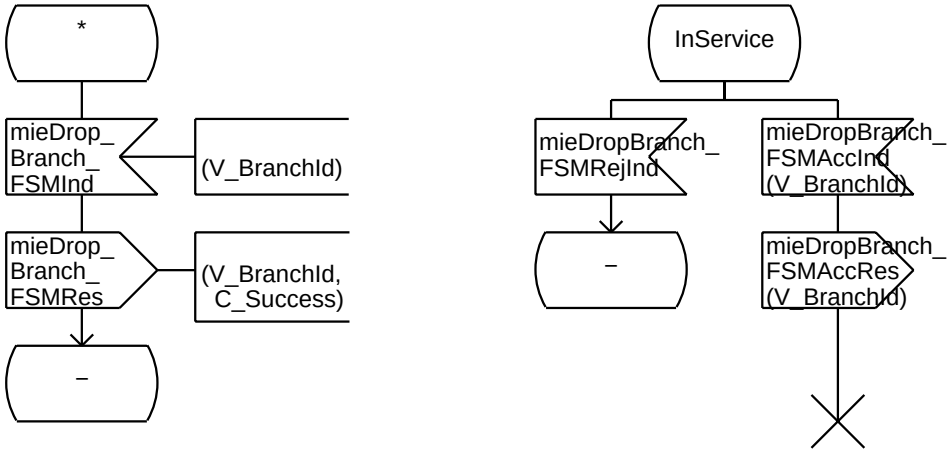

AddBranchEval
mieDropBranchFSMRejInd
mieAdd_
Branch_
FSMAccInd
(V_BranchId,
V_UpdateNeeded)
mieAdd_
Branch_
FSMRejInd
mieDropBranch_
FSMAccInd
(V_BranchId)
B_UpdateNotneeded
V_UpdateNeeded
B_UpdateNeeded
mieDropBranch_
FSMAccRes
(V_BranchId)
–
Set
(T_BranchEst)
InService
UpdateBranch

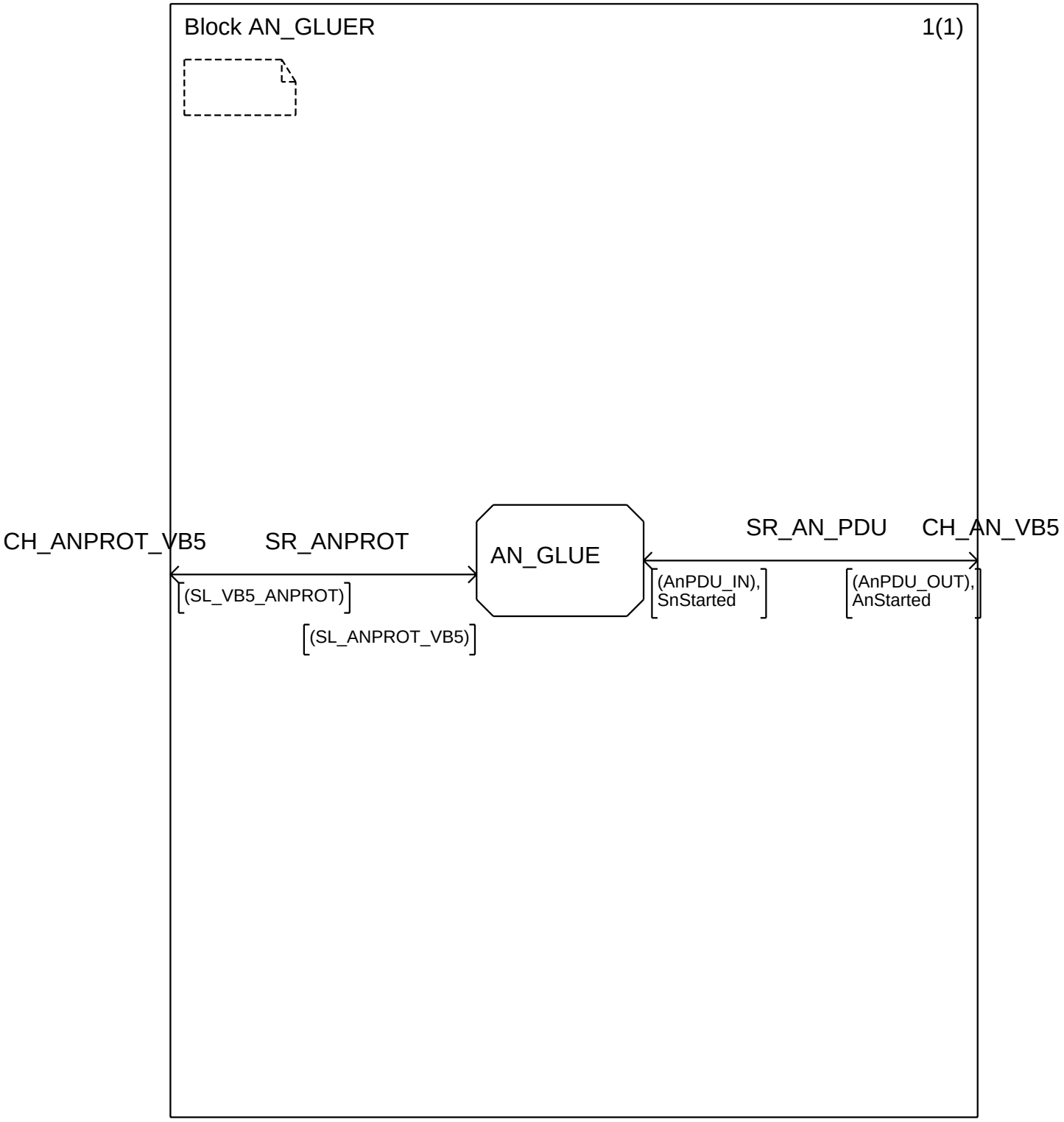

UpdateBranch
mieUpdate_ Branch_ FSMInd
(V_BranchId)
T_BranchEst
mieDropBranch_ FSMRejInd
mieDropBranch_ FSMAccInd (V_BranchId)
Reset (T_BranchEst)
mieDrop_ Branch_ FSMRes
(V_BranchId, C_Success)
–
mieDropBranch_ FSMAccRes (V_BranchId)
mieUpdate_ Branch_ FSMRes
(V_BranchId, C_Success)
DropBranch_ NoRes
UpdateBranch_ Eval


UpdateBranch_ Eval
mieUpdate_ Branch_ FSMAccInd
mieUpdate_ Branch_ FSMRejInd
mieDropBranch_ FSMRejInd
mieDropBranch_ FSMAccInd (V_BranchId)
InService
–
mieDropBranch_ FSMAccRes (V_BranchId)


* (UpdateBranch)
mieUpdate_ Branch_ FSMInd
(V_BranchId)
mieUpdate_ Branch_ FSMRes
(V_BranchId, C_IncompBranchState)
–


DropBranch_ NoRes
mieDrop_ BranchFSM_ AccInd
(V_BranchId)
mieDropBranch_ FSMRejInd
–


*
mieDrop_
Branch_
FSMInd
(V_BranchId)
mieDrop_
Branch_
FSMRes
(V_BranchId,
C_Success)
–
InService
mieDropBranch_
FSMRejInd
mieDropBranch_
FSMAccInd
(V_BranchId)
–
mieDropBranch_
FSMAccRes
(V_BranchId)


Block AN_GLUER
1(1)
CH_ANPROT_VB5
SR_ANPROT
(SL_VB5_ANPROT)
(SL_ANPROT_VB5)
AN_GLUE
SR_AN_PDU
CH_AN_VB5
(AnPDU_IN),
SnStarted
(AnPDU_OUT),
AnStarted

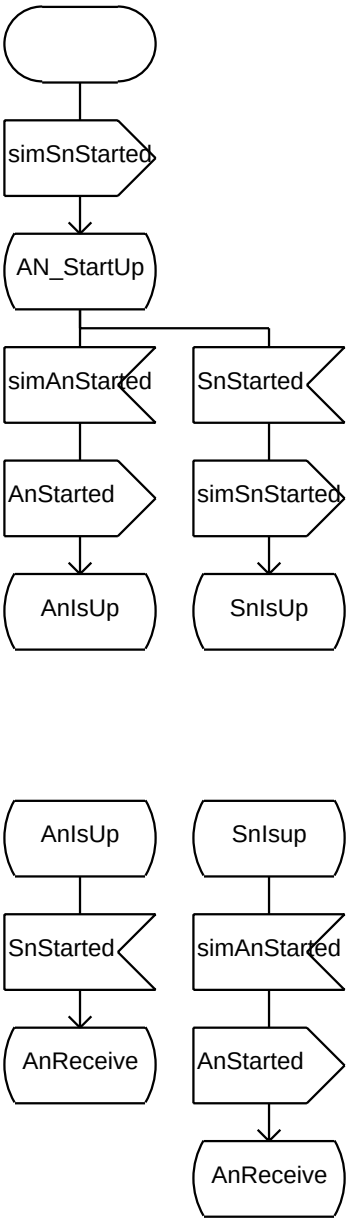
/* AN_GLUE is responsible for forming the PDUs into ASN1 and adding optional Information Elements. */

```
DCL
V_TransId IT_TransId,
V_ConnRefNo IT_ConnRefNo,
V_BranchId IT_BranchId,
V_RejCause IT_RejCause;
```

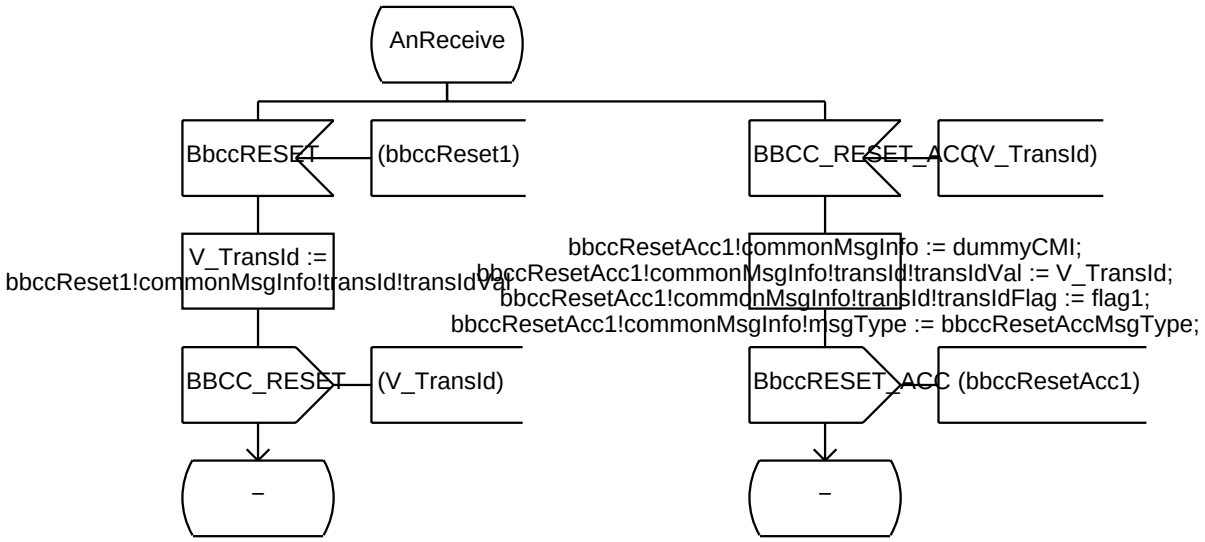
```
DCL
bbccReset1       BbccReset,
bbccResetAcc1 BbccResetAcc,
bbccResetRej1  BbccResetRej,
bbccPresync1  BbccPresync,
bbccPresyncAcc1  BbccPresyncAcc,
bbccPresyncRej1  BbccPresyncRej,
anFault1  AnFault,
anFaultAcc1 AnFaultAcc;
```

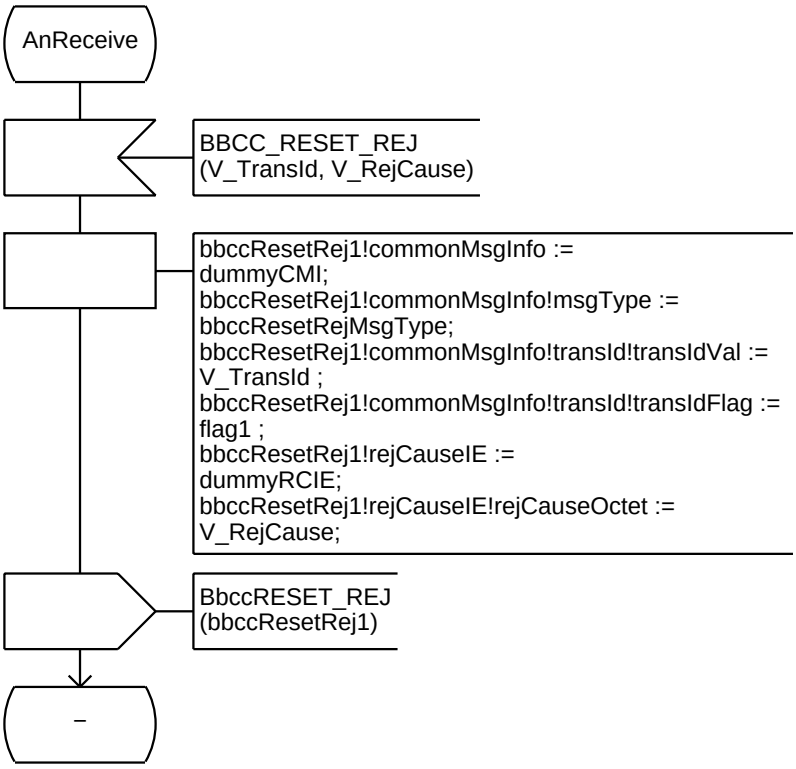
```
DCL
alloc1 Alloc,
allocAcc1 AllocAcc,
allocComp1 AllocComp,
allocCompAcc1 AllocCompAcc,
allocCompRej1 AllocCompRej,
allocRej1 AllocRej,
dealloc1 Dealloc,
deallocAcc1 DeallocAcc,
addBranch1 AddBranch,
addBranchAcc1 AddBranchAcc,
addBranchRej1 AddBranchRej,
updateBranch1 UpdateBranch,
updateBranchAcc1 UpdateBranchAcc,
updateBranchRej1 UpdateBranchRej,
dropBranch1 DropBranch,
dropBranchAcc1 DropBranchAcc,
dropBranchRej1 DropBranchRej,
modify1 Modify,
modifyAcc1 ModifyAcc,
modifyRej1 ModifyRej,
modifyComp1 ModifyComp,
modifyCompAcc1 ModifyCompAcc,
modifyCompRej1 ModifyCompRej,
modifyAbort1  ModifyAbort,
modifyAbortAcc1 ModifyAbortAcc,
modifyAbortRej1 ModifyAbortRej,
unknownPDU1 UnknownPDU,
truncInfoBbccReset1 TruncInfoBbccReset,
bbccResetMissingMndtryIE1 BbccResetMissingMndtryIE,
bbccResetXtraUnkIE1 BbccResetXtraUnkIE,
bbccResetXtraUnexpIE1 BbccResetXtraUnexpIE,
protocolError1 ProtocolError ;
```

/* AN_GLUE is responsible for forming the PDUs into ASN1 and adding optional Information Elements. */



/* AN_GLUE is responsible for forming the PDUs into ASN1 and adding optional Information Elements. */





/* AN_GLUE is responsible for forming the PDUs into ASN1 and adding optional Information Elements. */

AnReceive

BbccPRESYNC (bbccPresync1)

V_TransId :=
bbccPresync1!commonMsgInfo!transId!transIdVal

BBCC_PRESYNC(V_TransId)

–

AnReceive

BBCC_PRESYNC_REJ
(V_TransId)

bbccPresyncRej1!commonMsgInfo :=
dummyCMI;
bbccPresyncRej1!commonMsgInfo!msgTy
bbccPresyncRejMsgType;
bbccPresyncRej1!commonMsgInfo!transId
V_TransId ;
bbccPresyncRej1!commonMsgInfo!transId
flag1 ;
bbccPresyncRej1!rejCauseIE :=
dummyRCIE;
bbccPresyncRej1!rejCauseIE!rejCauseOct
C_OpRejLoc;
/* C_OpRejLoc
SAM add 7Dec99.
SDL (BBCC_PRESYNC_REJ does not co
PDU contains it. */

BbccPRESYNC_REJ
(bbccPresyncRej1)

–

AnReceive

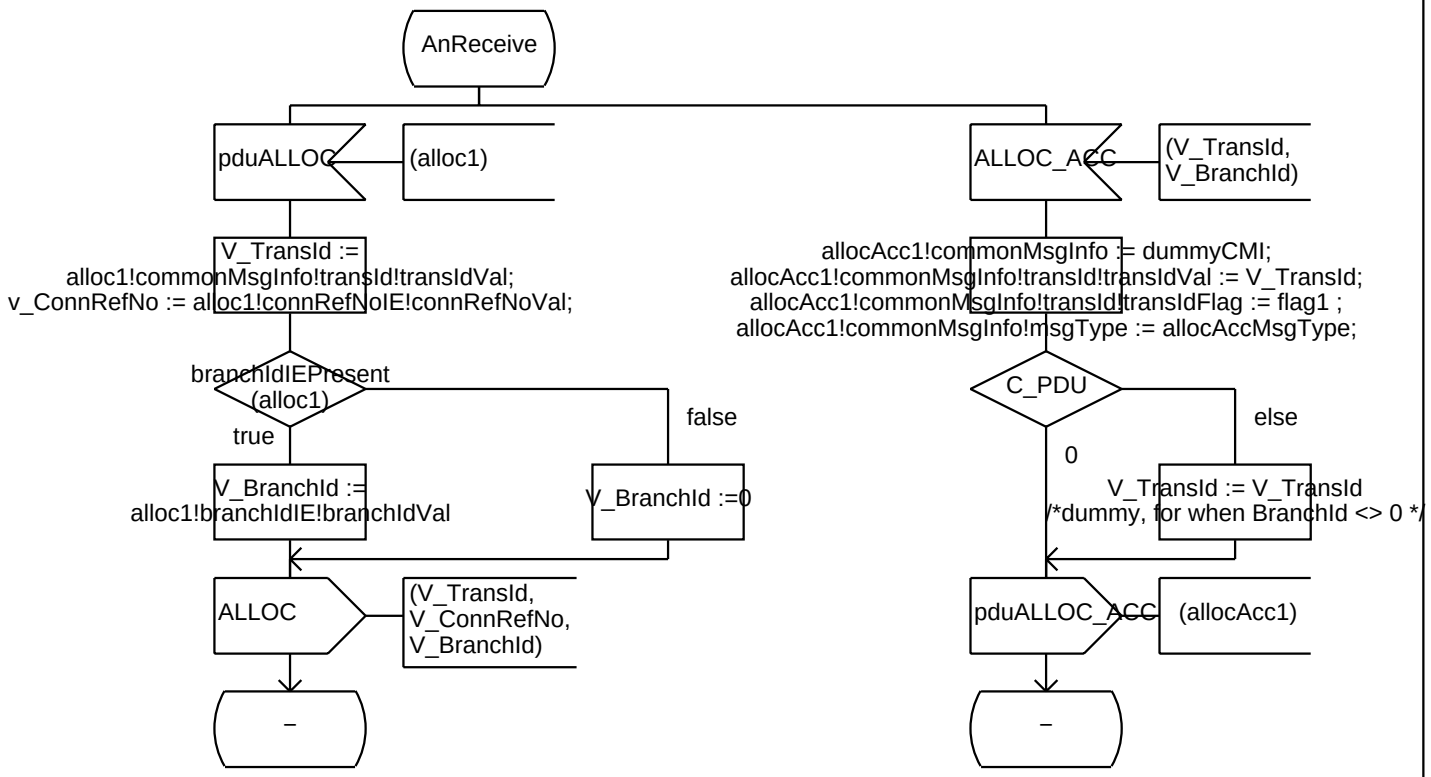
BBCC_PRESYNC_ACC
(V_TransId)

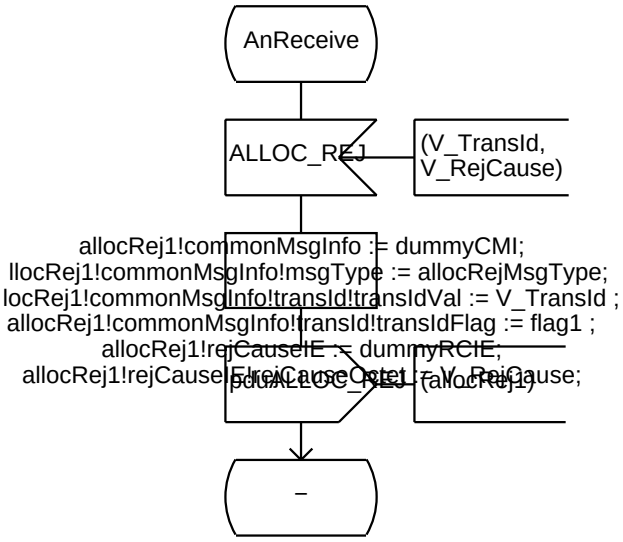
bbccPresyncAcc1!commonMsgInfo := dummyCMI;
bbccPresyncAcc1!commonMsgInfo!transId!transIdVal := V_TransId;
bbccPresyncAcc1!commonMsgInfo!transId!transIdFlag := flag1;
bbccPresyncAcc1!commonMsgInfo!msgType := bbccPresyncAccMsgType;

BbccPRESYNC_ACC
(bbccPresyncAcc1)

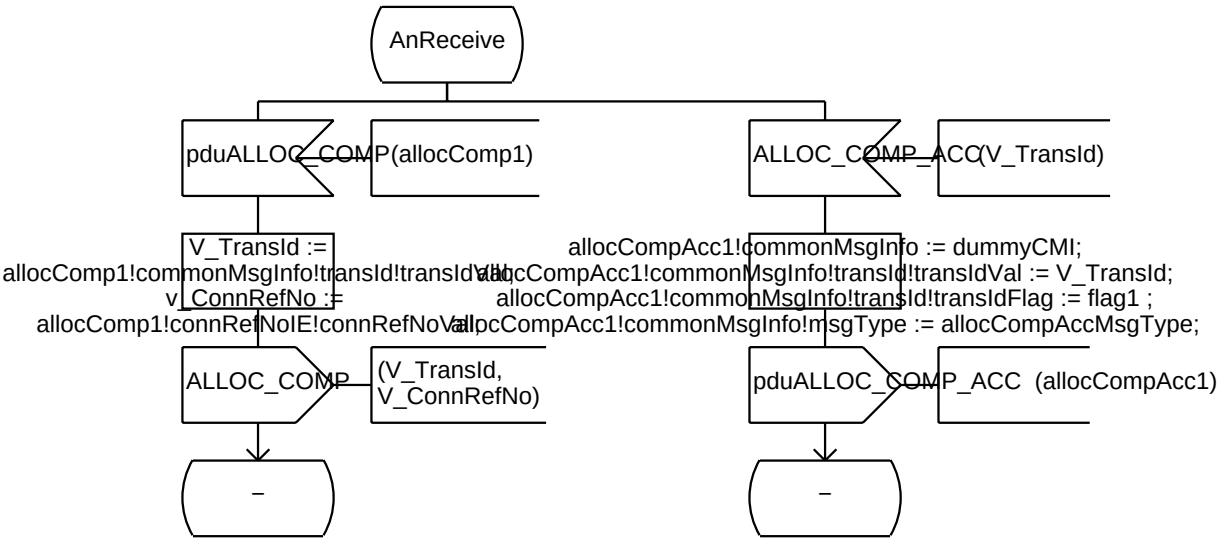
–

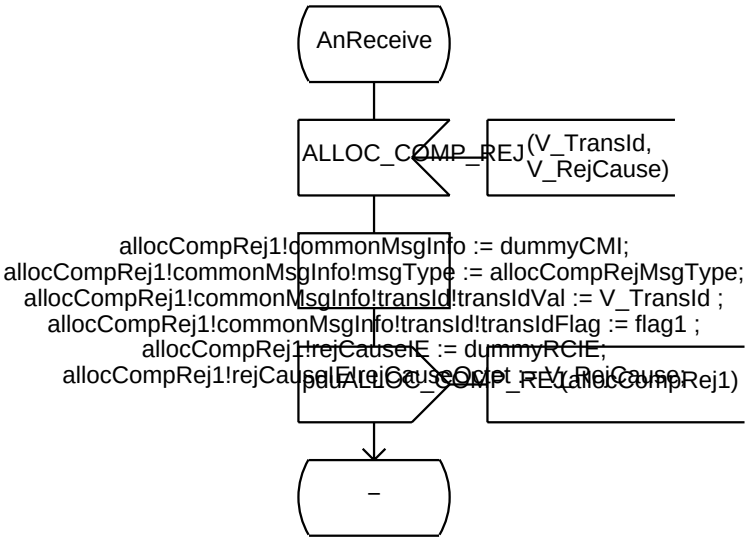
/* AN_GLUE is responsible for forming the PDUs into ASN1 and adding optional Information Elements. */



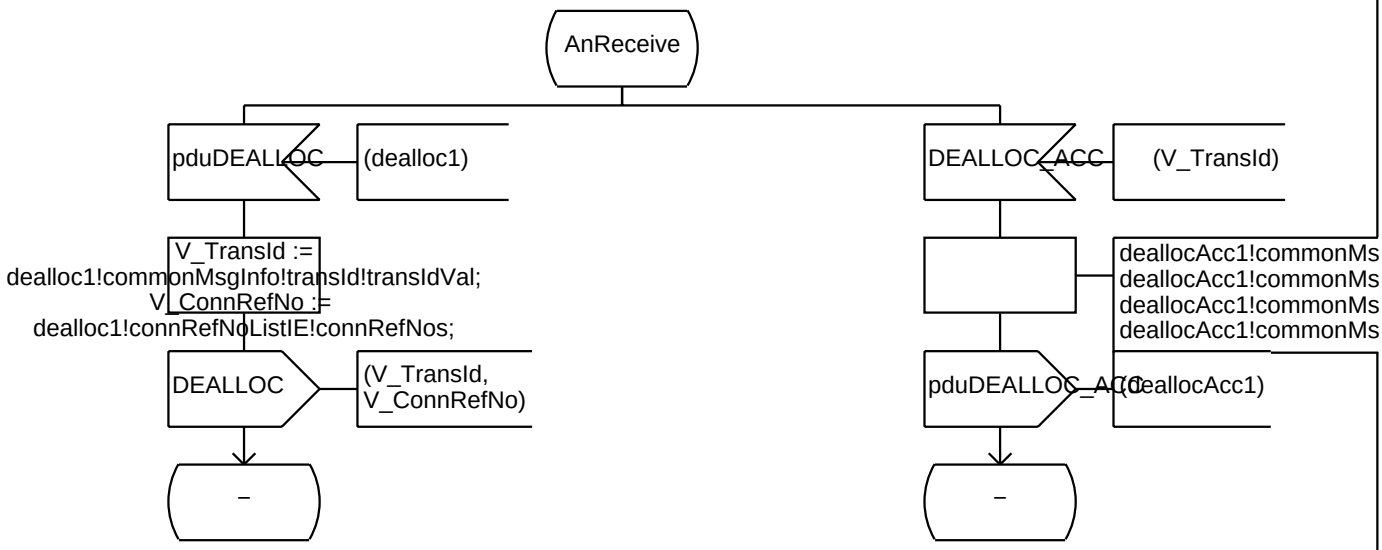


/* AN_GLUE is responsible for forming the PDUs into ASN1 and adding optional Information Elements. */

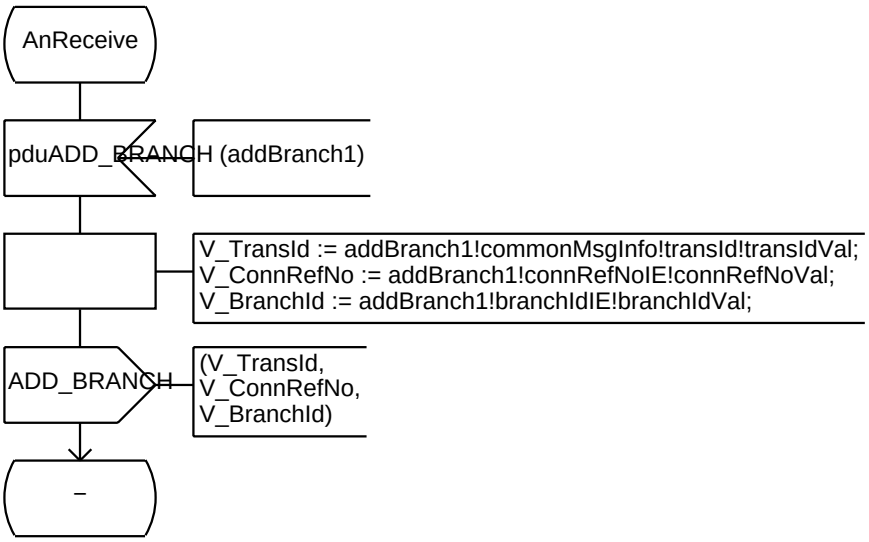


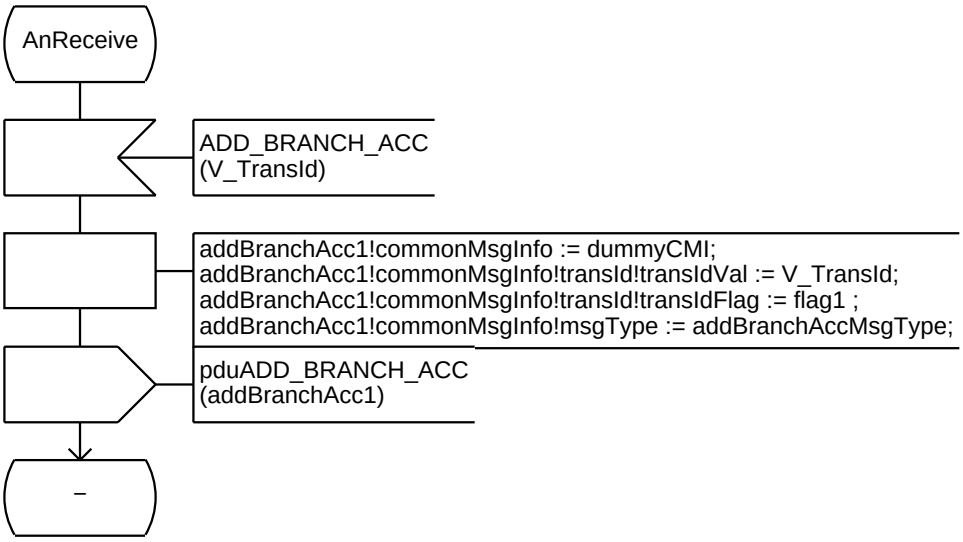


/* AN_GLUE is responsible for forming the PDUs into ASN1 and adding optional Information Elements. */

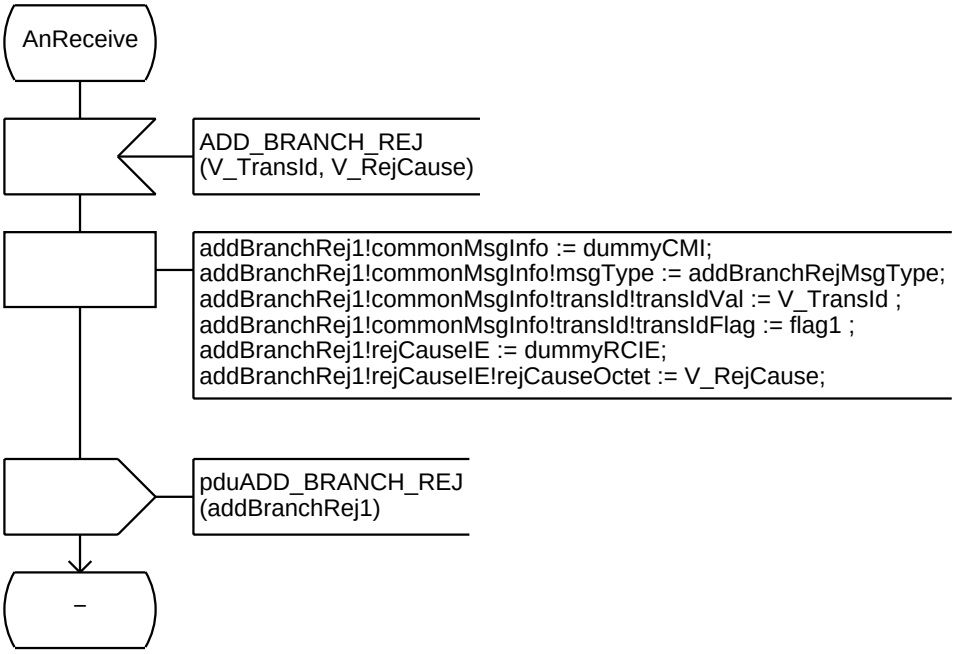


/* AN_GLUE is responsible for forming the PDUs into ASN1 and adding optional Information Elements. */

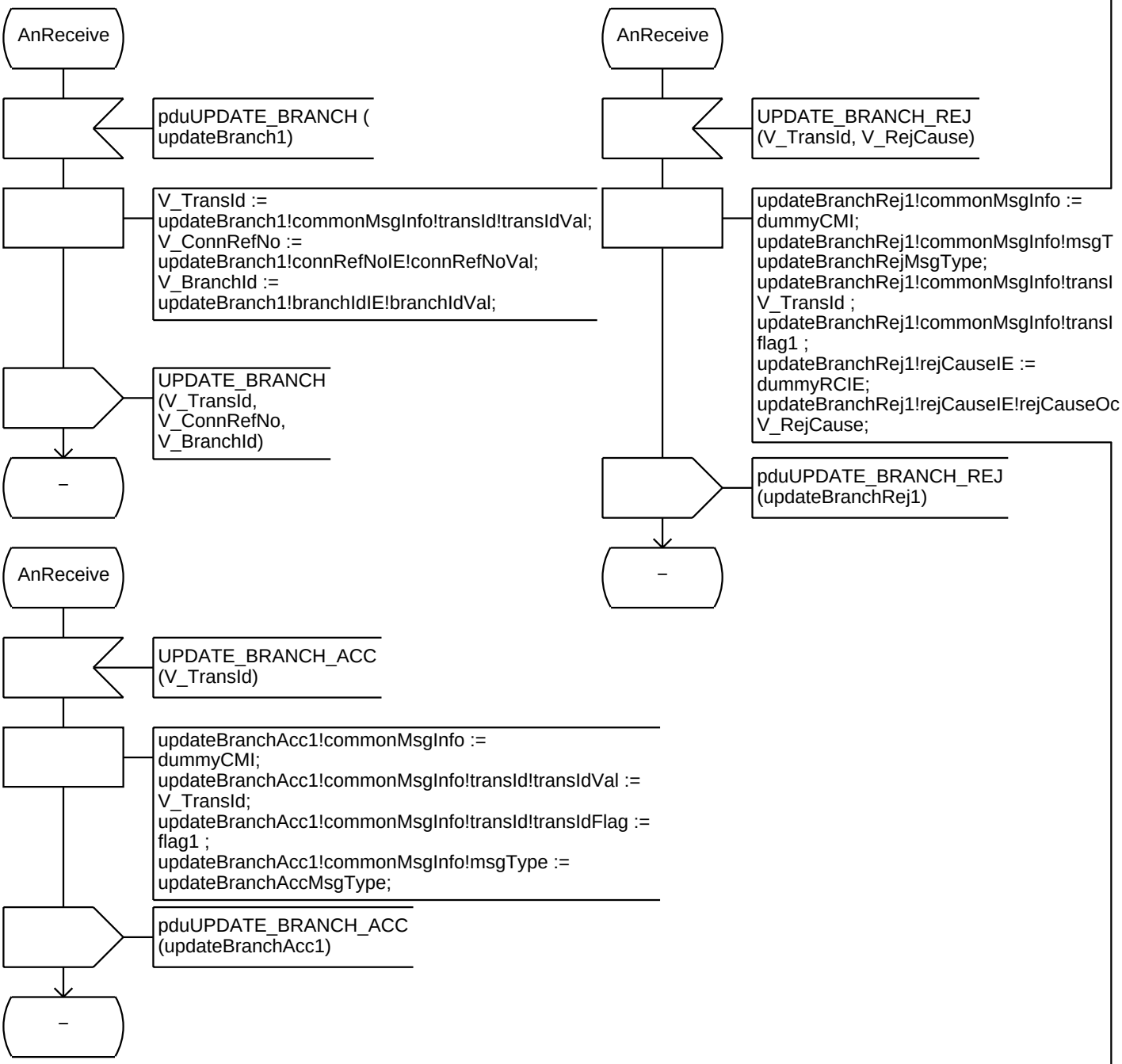




/* AN_GLUE is responsible for forming the PDUs into ASN1 and adding optional Information Elements. */



/* AN_GLUE is responsible for forming the PDUs into ASN1 and adding optional Information Elements. */



/* AN_GLUE is responsible for forming the PDUs into ASN1 and adding optional Information Elements. */

AnReceive

pduDROP_BRANCH(dropBranch1)

V_TransId :=
dropBranch1!commonMsgInfo!transId!transIdVal;
V_ConnRefNo :=
dropBranch1!connRefNoIE!connRefNoVal;
V_BranchId :=
dropBranch1!branchIdListIE!branchIds;

DROP_BRANCH
(V_TransId,
V_ConnRefNo,
V_BranchId)

–

AnReceive

DROP_BRANCH_REJ
(V_TransId, V_RejCause)

dropBranchRej1!commonMsgInfo :=
dummyCMI;
dropBranchRej1!commonMsgInfo!msgTyp
dropBranchRejMsgType;
dropBranchRej1!commonMsgInfo!transId!t
V_TransId ;
dropBranchRej1!commonMsgInfo!transId!t
flag1 ;
dropBranchRej1!rejCauseIE :=
dummyRCIE;
dropBranchRej1!rejCauseIE!rejCauseOcte
V_RejCause;

pduDROP_BRANCH_REJ
(dropBranchRej1)

–

AnReceive

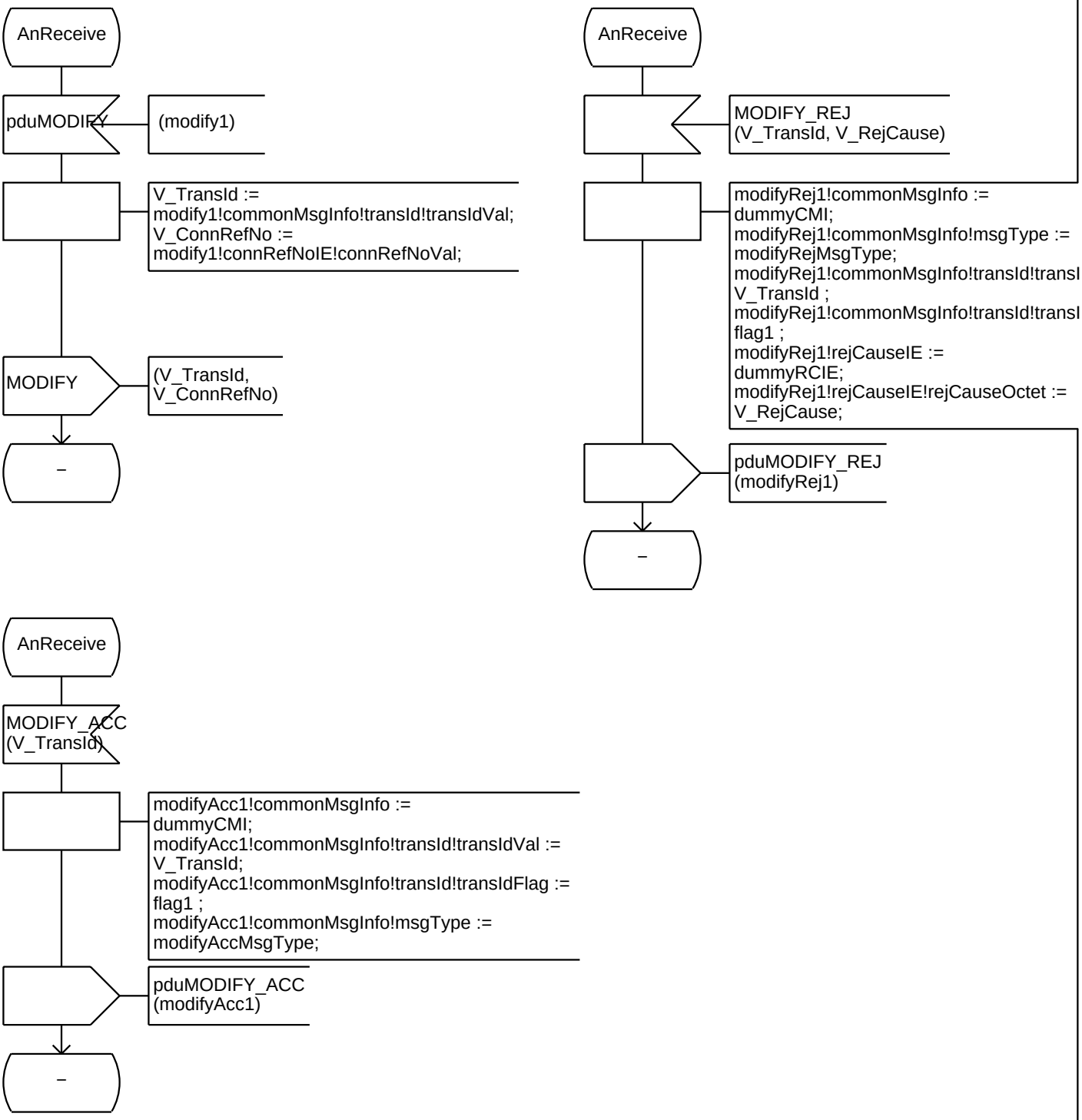
DROP_BRANCH_ACC
(V_TransId)

dropBranchAcc1!commonMsgInfo :=
dummyCMI;
dropBranchAcc1!commonMsgInfo!transId!transIdVal :=
V_TransId;
dropBranchAcc1!commonMsgInfo!transId!transIdFlag :=
flag1 ;
dropBranchAcc1!commonMsgInfo!msgType :=
dropBranchAccMsgType;

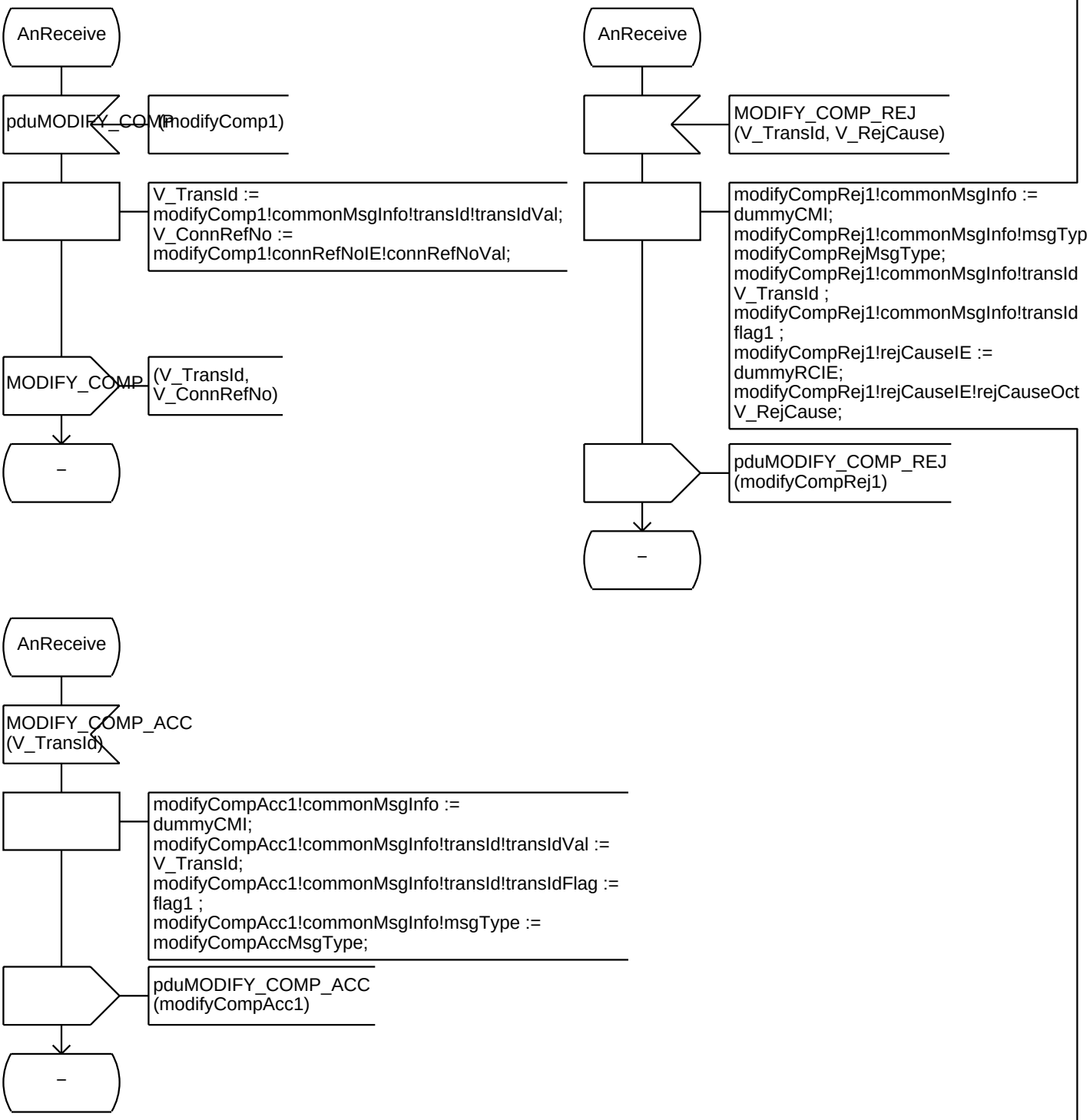
pduDROP_BRANCH_ACC
(dropBranchAcc1)

–

/* AN_GLUE is responsible for forming the PDUs into ASN1 and adding optional Information Elements. */



/* AN_GLUE is responsible for forming the PDUs into ASN1 and adding optional Information Elements. */



/* AN_GLUE is responsible for forming the PDUs into ASN1 and adding optional Information Elements. */

AnReceive

AN_FAULT
(V_TransId, V_ConnRefNo, V_BranchId)

```
anFault1!commonMsgInfo := dummyCMI;
anFault1!commonMsgInfo!transId!transIdVal := V_TransId;
anFault1!commonMsgInfo!msgType := anFaultMsgType;
anFault1!usrPortConnIdIE!commonIEInfo!iEType := iEUsrPortConnId;
anFault1!usrPortConnIdIE!commonIEInfo!iECompatInd := hex80;
anFault1!usrPortConnIdIE!commonIEInfo!iELength := hex0006;
anFault1!usrPortConnIdIE!usrPortConnIdIEOctet5 := hexA0;
anFault1!usrPortConnIdIE!lgclUsrPortId := hexA00101;
anFault1!usrPortConnIdIE!vpci := hexA001;
anFault1!usrPortConnIdIE!vci := hexA101;
```

pduAN_FAULT
(anFault1)

–

AnReceive

pduAN_FAULT_ACC
(anFaultAcc1)

```
V_TransId := anFaultAcc1!commonMsgInfo!transId!transIdVal
```

AN_FAULT_ACC
(V_TransId)

–

/* AN_GLUE is responsible for forming the PDUs into ASN1 and adding optional Information Elements. */

AnReceive

pduMODIFY_ABORT (modifyAbort1)

V_TransId :=
modifyAbort1!commonMsgInfo!transId!transIdVal;
V_ConnRefNo :=
modifyAbort1!connRefNoIE!connRefNoVal;

MODIFY_ABORT (V_TransId, V_ConnRefNo)

–

AnReceive

MODIFY_ABORT_REJ
(V_TransId, V_RejCause)

modifyAbortRej1!commonMsgInfo :=
dummyCMI;
modifyAbortRej1!commonMsgInfo!msgTyp
modifyAbortRejMsgType;
modifyAbortRej1!commonMsgInfo!transId!
V_TransId ;
modifyAbortRej1!commonMsgInfo!transId!
flag1 ;
modifyAbortRej1!rejCauseIE :=
dummyRCIE;
modifyAbortRej1!rejCauseIE!rejCauseOcte
V_RejCause;

pduMODIFY_ABORT_REJ
(modifyAbortRej1)

–

AnReceive

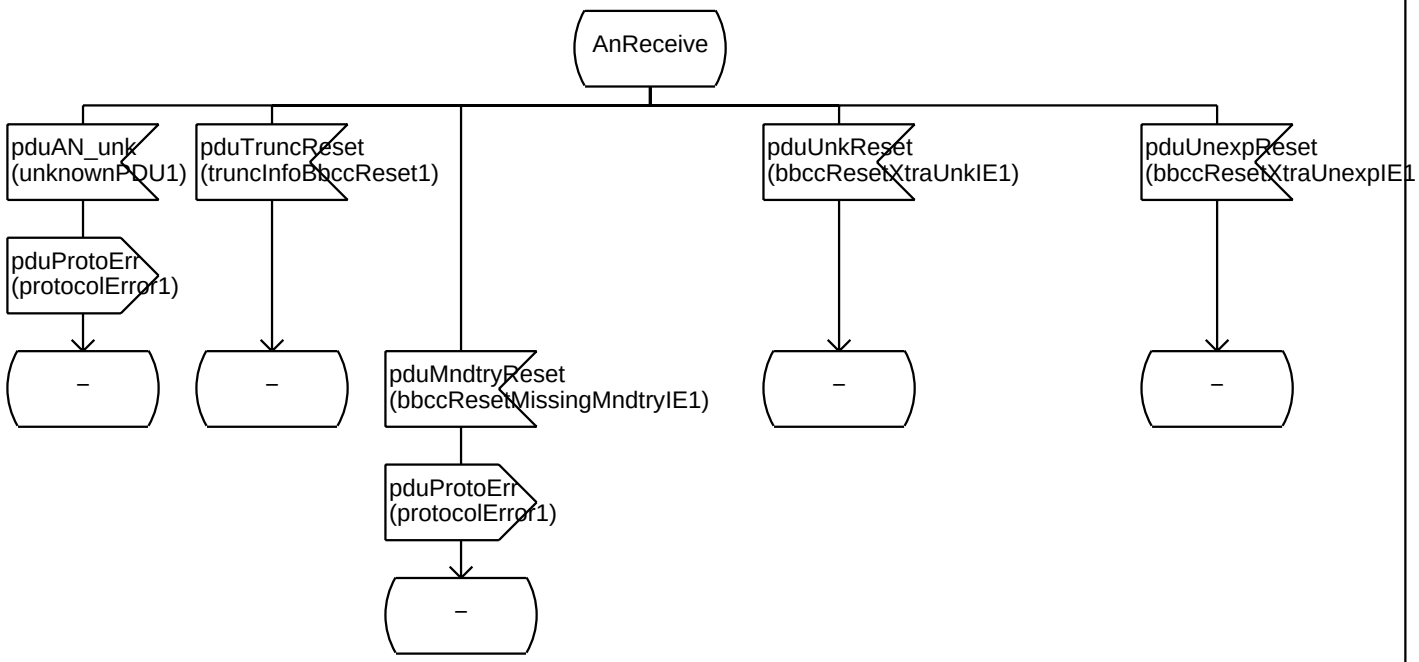
MODIFY_ABORT_ACC
(V_TransId)

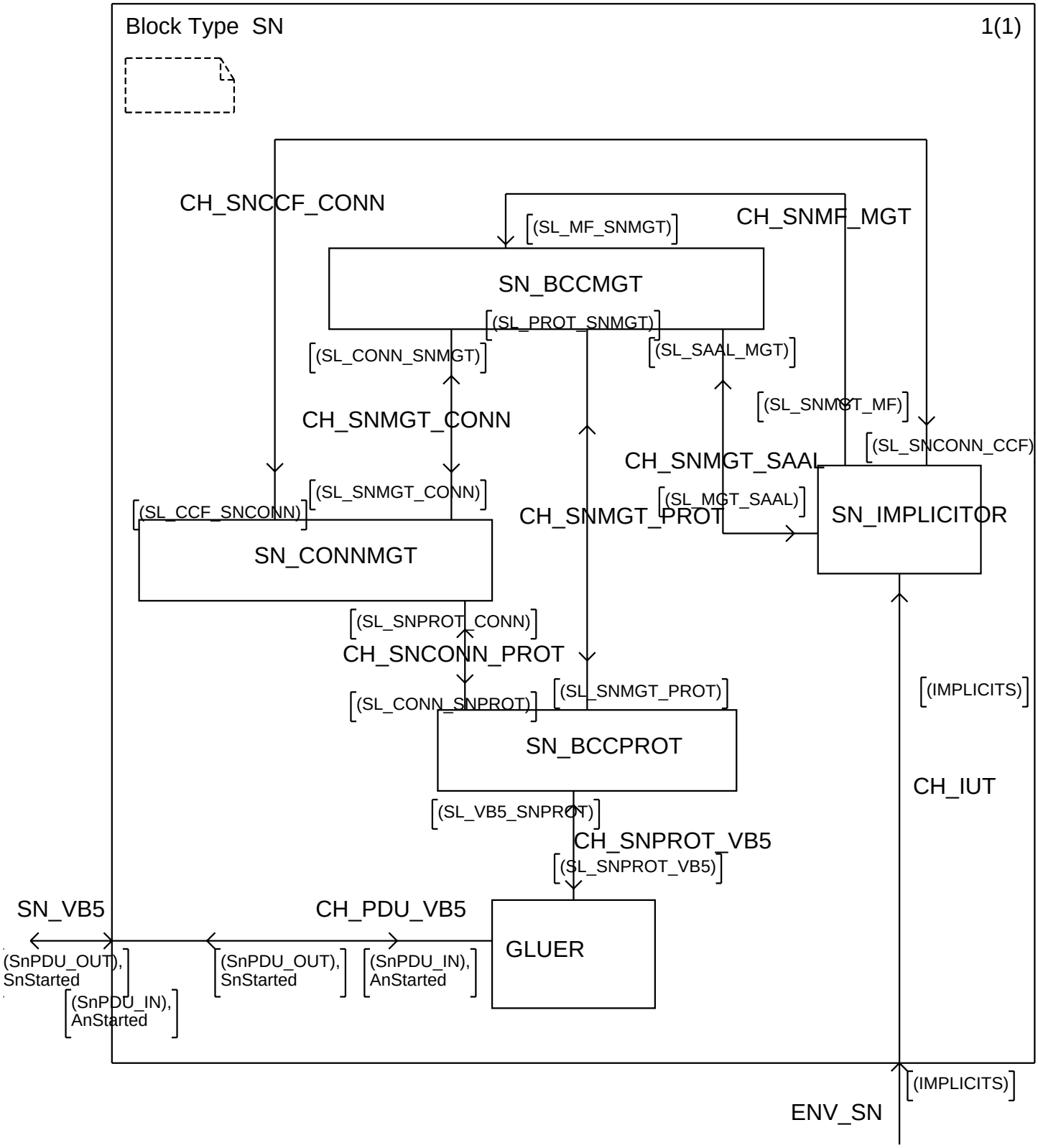
modifyAbortAcc1!commonMsgInfo :=
dummyCMI;
modifyAbortAcc1!commonMsgInfo!transId!transIdVal :=
V_TransId;
modifyAbortAcc1!commonMsgInfo!transId!transIdFlag :=
flag1 ;
modifyAbortAcc1!commonMsgInfo!msgType :=
modifyAbortAccMsgType;

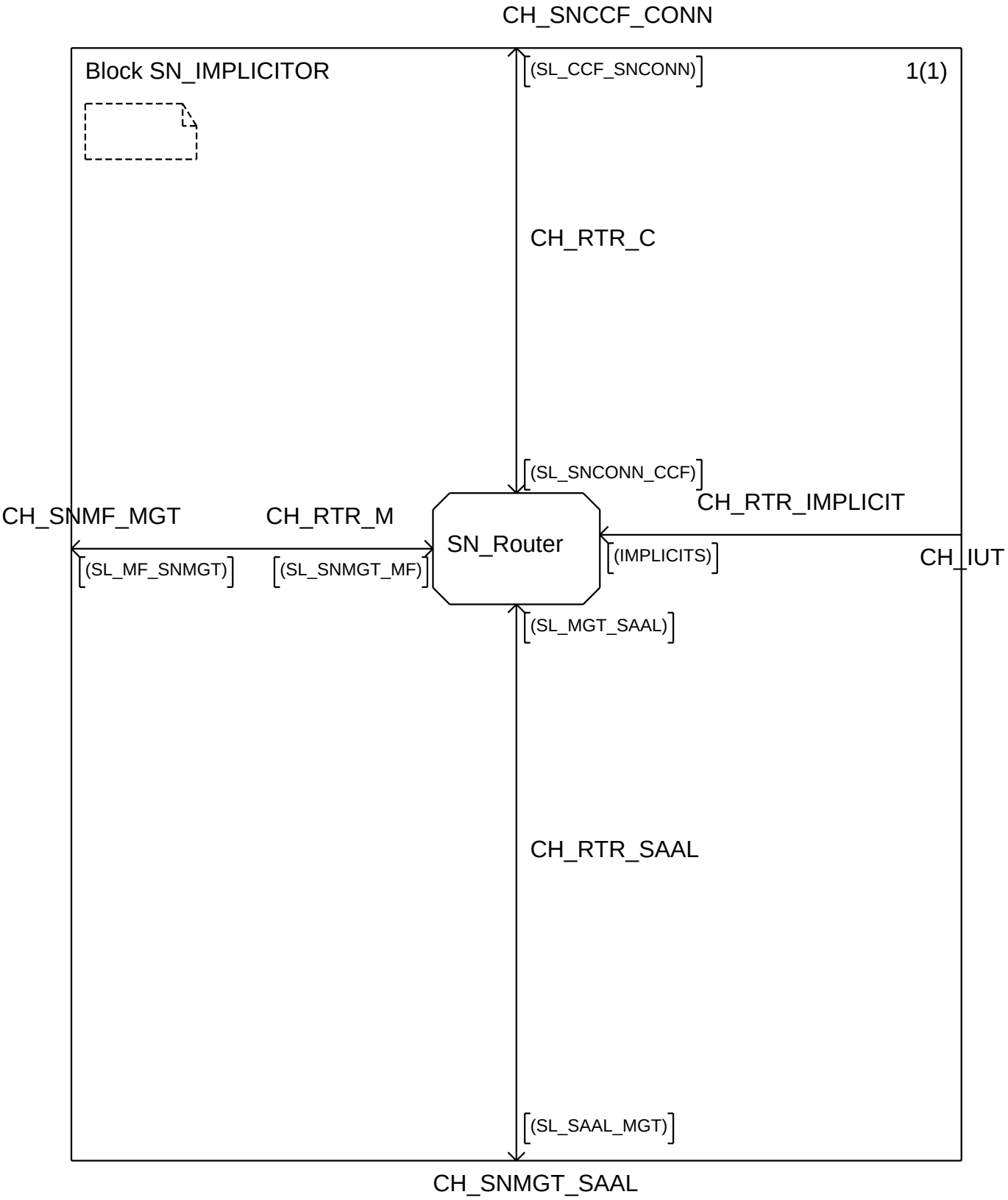
pduMODIFY_ABORT_ACC
(modifyAbortAcc1)

–

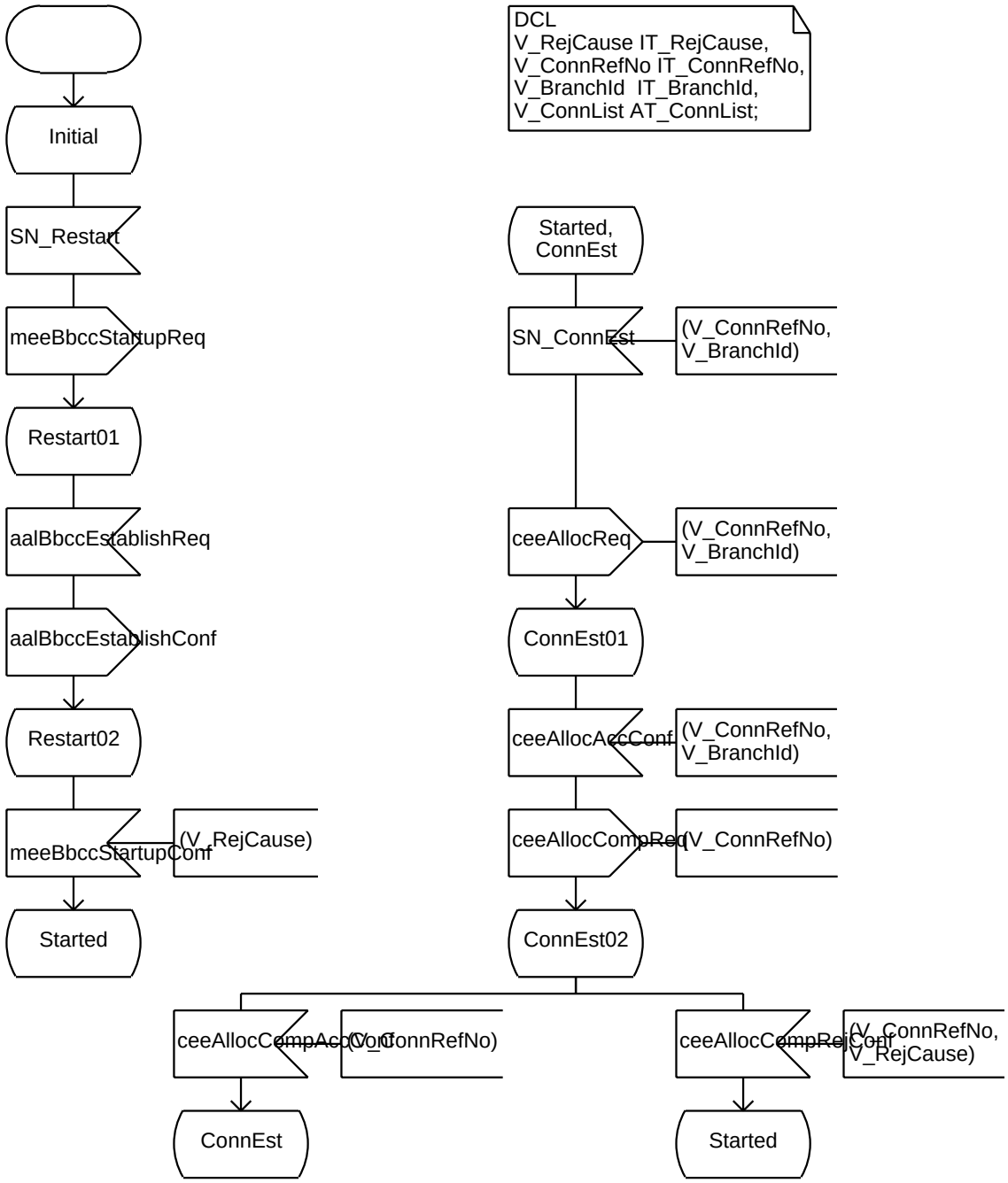
/* AN_GLUE is responsible for forming the PDUs into ASN1 and adding optional Information Elements. */



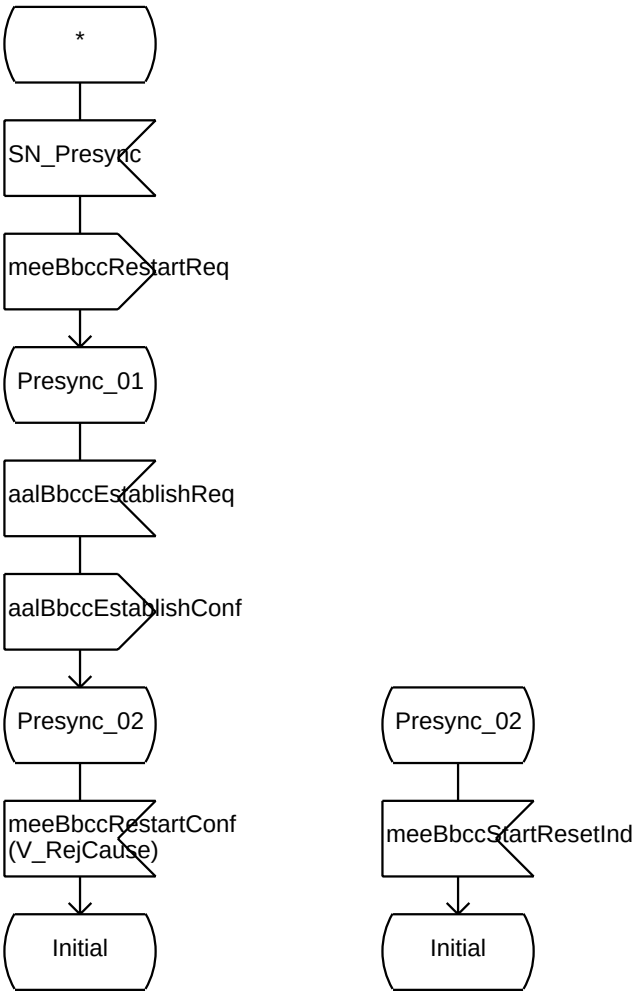

Block Type SN
1(1)
CH_SNCCF_CONN
(SL_MF_SNMGT)
CH_SNMF_MGT
SN_BCCMGT
(SL_PROT_SNMGT)
(SL_CONN_SNMGT)
(SL_SAAL_MGT)
(SL_SNMGT_MF)
CH_SNMGT_CONN
(SL_SNCONN_CCF)
CH_SNMGT_SAAL
(SL_SNMGT_CONN)
(SL_MGT_SAAL)
(SL_CCF_SNCONN)
CH_SNMGT_PROT
SN_IMPLICITOR
SN_CONNMGT
(SL_SNPROT_CONN)
CH_SNCONN_PROT
(IMPLICITS)
(SL_SNMGT_PROT)
(SL_CONN_SNPROT)
SN_BCCPROT
CH_IUT
(SL_VB5_SNPROT)
CH_SNPROT_VB5
(SL_SNPROT_VB5)
SN_VB5
CH_PDU_VB5
GLUER
(SnPDU_OUT), SnStarted
(SnPDU_OUT), SnStarted
(SnPDU_IN), AnStarted
(SnPDU_IN), AnStarted
(IMPLICITS)
ENV_SN


CH_SNCCF_CONN
Block SN_IMPLICITOR
1(1)
(SL_CCF_SNCONN)
CH_RTR_C
(SL_SNCONN_CCF)
CH_RTR_IMPLICIT
CH_SNMF_MGT
CH_RTR_M
SN_Router
(IMPLICITS)
CH_IUT
(SL_MF_SNMGT)
(SL_SNMGT_MF)
(SL_MGT_SAAL)
CH_RTR_SAAL
(SL_SAAL_MGT)
CH_SNMGT_SAAL

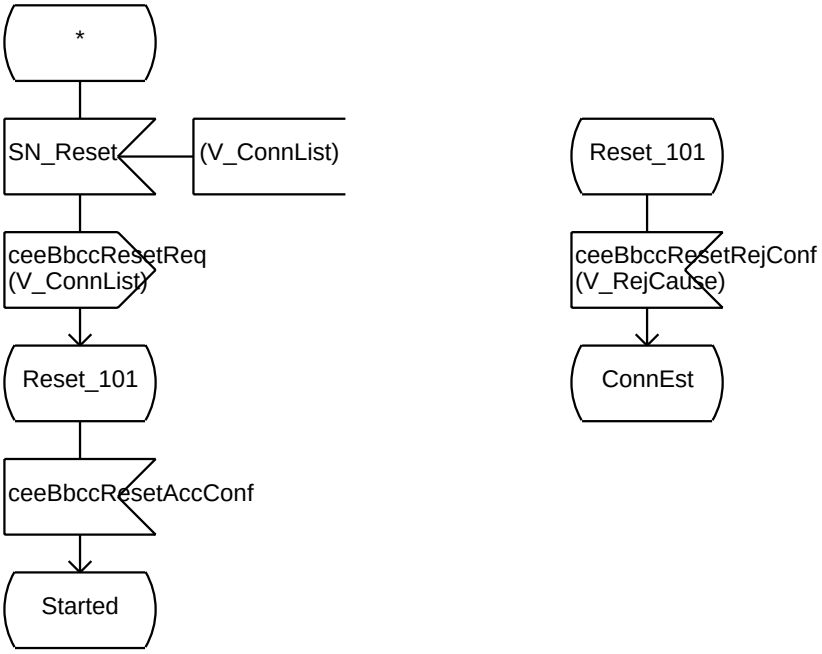
/* SN_Router is responsible for receiving and returning service primitives which normally should be sent to the surrounding system. */



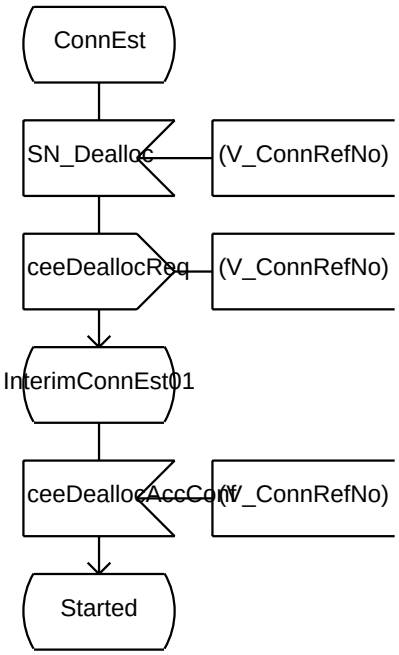
/* SN_Router is responsible for receiving and returning service primitives which normally should be sent to the surrounding system. */



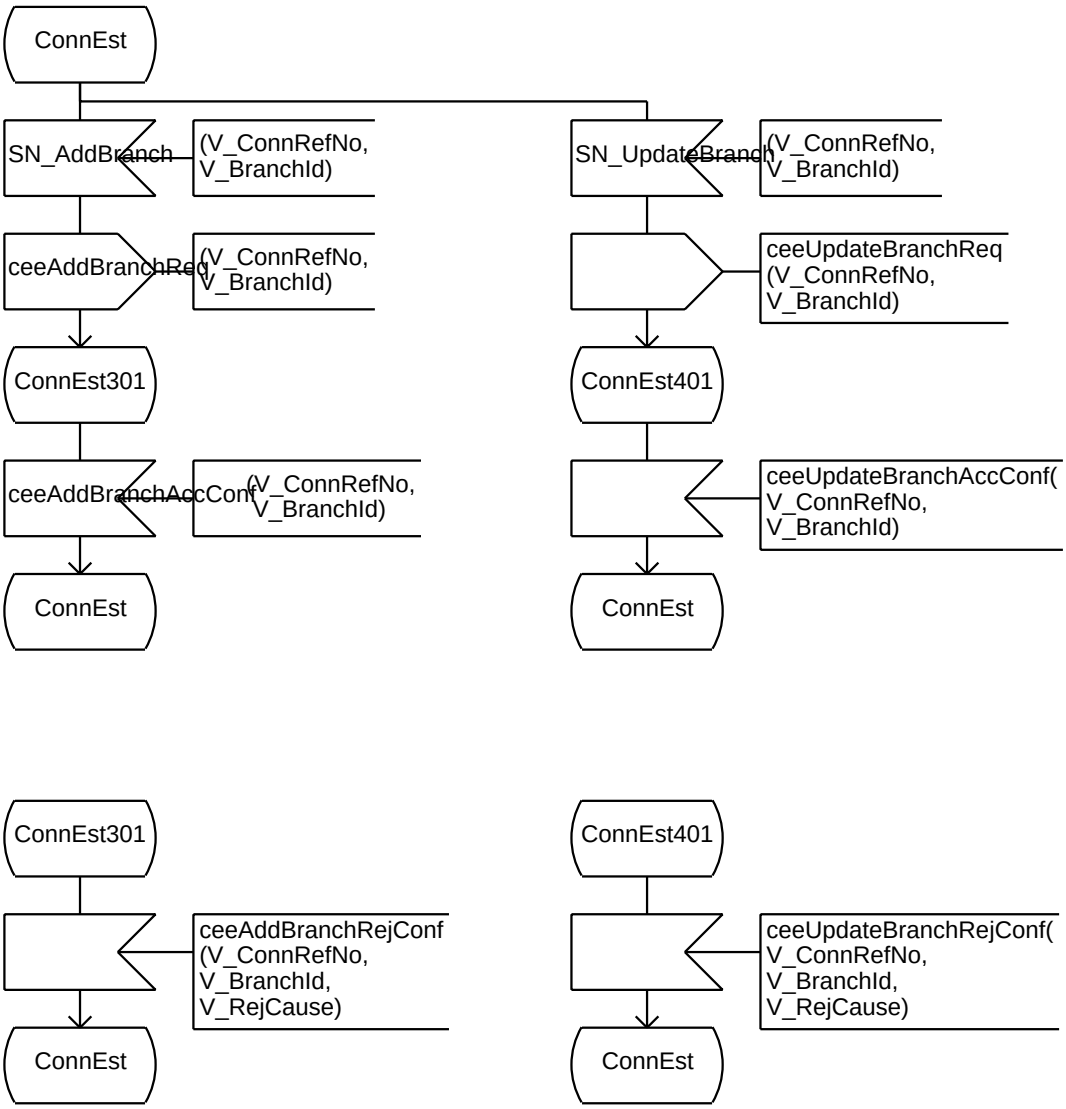
/* SN_Router is responsible for receiving and returning service primitives which normally should be sent to the surrounding system. */



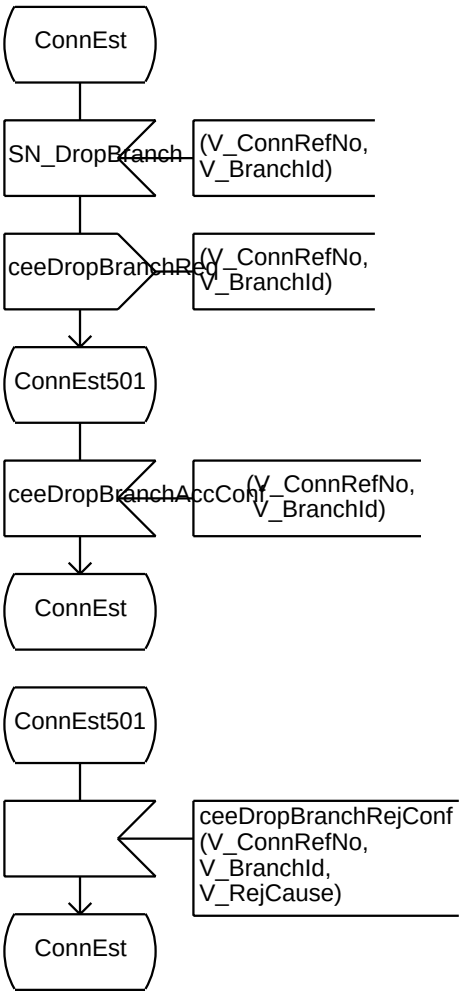
/* SN_Router is responsible for receiving and returning service primitives which normally should be sent to the surrounding system. */



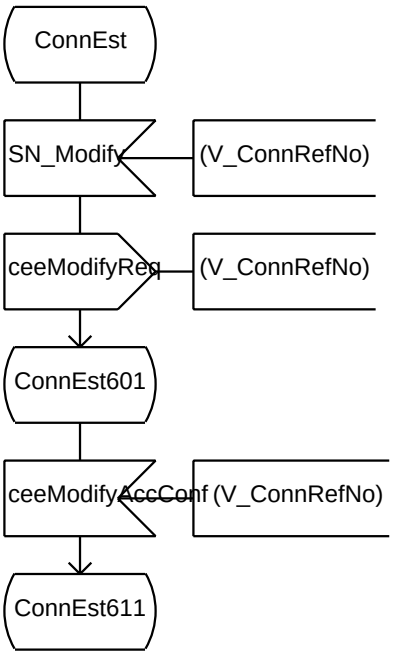
/* SN_Router is responsible for receiving and returning service primitives which normally should be sent to the surrounding system. */

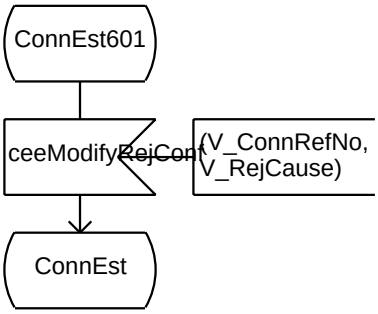


/* SN_Router is responsible for receiving and returning service primitives which normally should be sent to the surrounding system. */

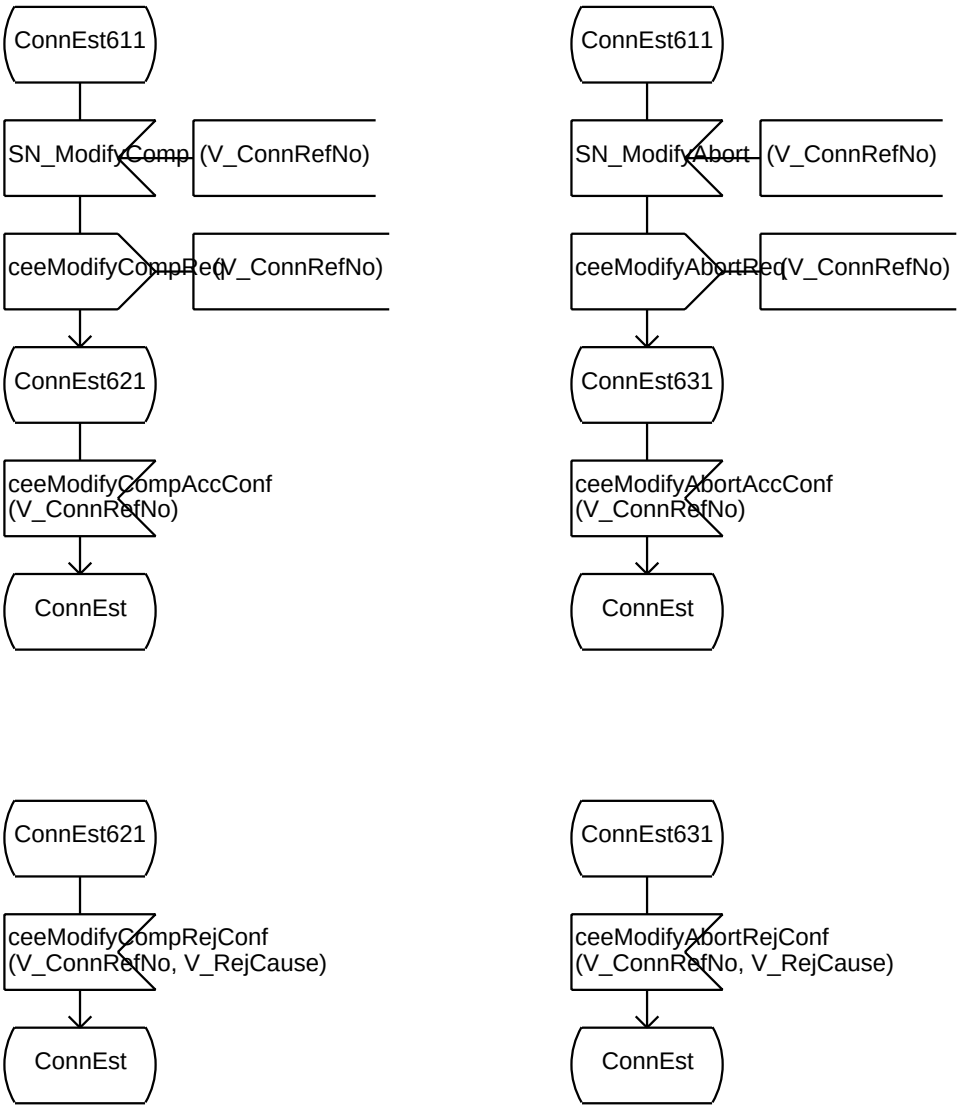


/* SN_Router is responsible for receiving and returning service primitives which normally should be sent to the surrounding system. */

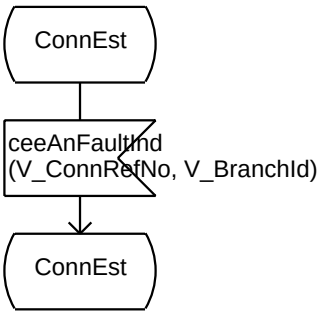


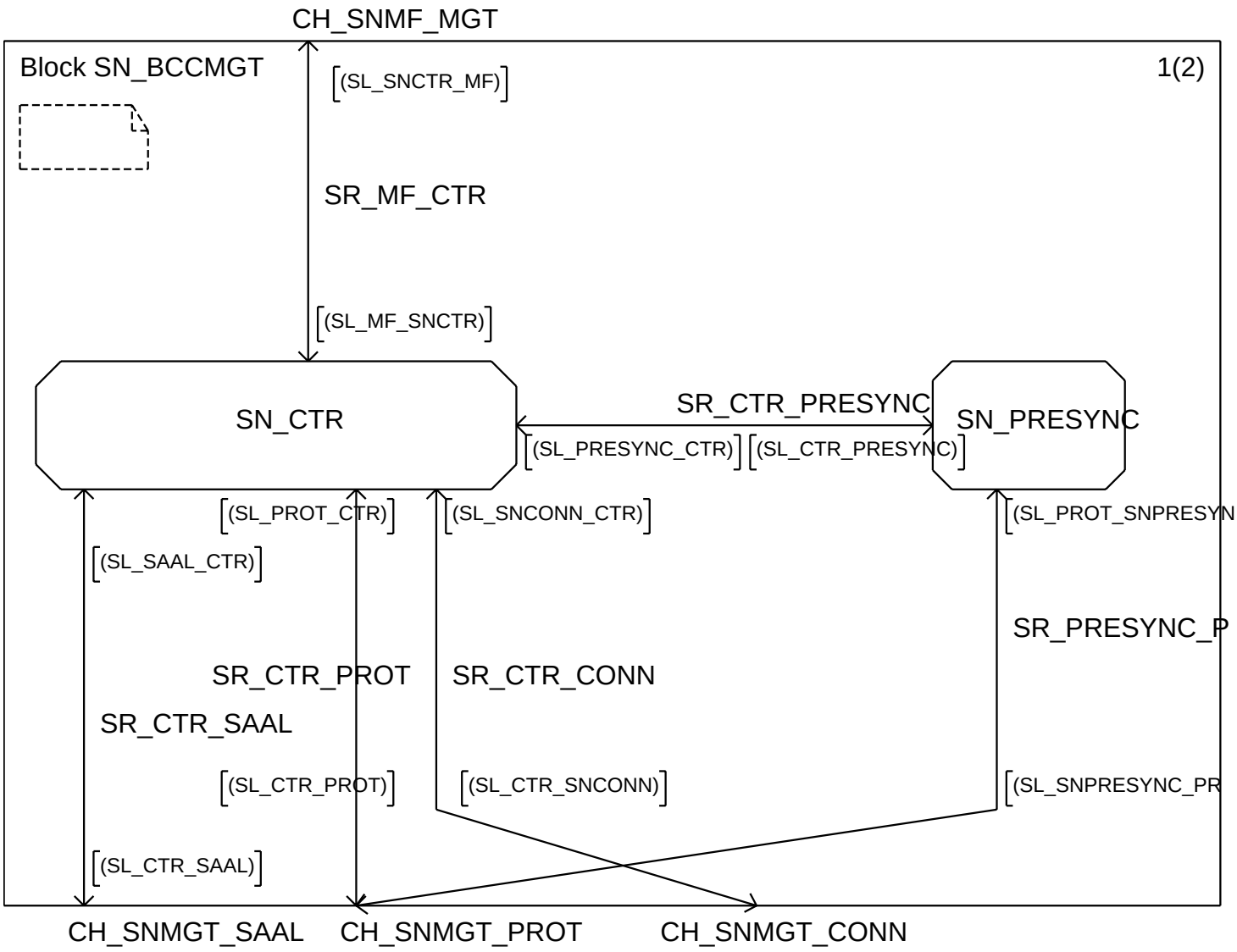


/* SN_Router is responsible for receiving and returning service primitives which normally should be sent to the surrounding system. */



/* SN_Router is responsible for receiving and returning service primitives which normally should be sent to the surrounding system. */




CH_SNMF_MGT
Block SN_BCCMGT
1(2)
(SL_SNCTR_MF)
SR_MF_CTR
(SL_MF_SNCTR)
SN_CTR
SR_CTR_PRESYNC
SN_PRESYNC
(SL_PRESYNC_CTR)
(SL_CTR_PRESYNC)
(SL_PROT_CTR)
(SL_SNCONN_CTR)
(SL_PROT_SNPRESYN
(SL_SAAL_CTR)
SR_PRESYNC_P
SR_CTR_PROT
SR_CTR_CONN
SR_CTR_SAAL
(SL_CTR_PROT)
(SL_CTR_SNCONN)
(SL_SNPRESYNC_PR
(SL_CTR_SAAL)
CH_SNMGT_SAAL
CH_SNMGT_PROT
CH_SNMGT_CONN

/* Declaration of primitives internal
to block SN_BCCMGT */

```
signal
mieBbccPresyncReq,
mieBbccPresyncAccConf,
mieBbccPresyncRejConf (IT_RejCause);
```

/* Declaration of signallists
internal to block SN_CONNMGT */

```
signallist SL_PRESYNC_CTR =
mieBbccPresyncAccConf,
mieBbccPresyncRejConf;
```

```
signallist SL_CTR_PRESYNC =
mieBbccPresyncReq,
mieStartTraffic,
mieStopTraffic;
```

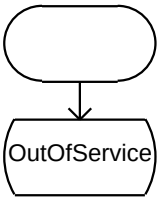
```
;
/* VB5.2 BCC management entity
   Process CTR */
```

```
/* State descriptions
OutOfService:          Initial state, no SAAL connection available
RstrtAwaitEstConf: Process is waiting for establish confirmation of the peer entity, restart procedure
StrtpAwaitEstConf: Process is waiting for establish confirmation of the peer entity, startup procedure
SaalEst:               SAAL in operation
Presync:               Process is waiting for presync confirmation, restart procedure
RstrtReset:            Process is waiting for reset confirmation, restart procedure
StrtpReset:            Process is waiting for reset confirmation, startup procedure
InService:             Interface is in operational state
AwaitRelConf:          Process is waiting for release confirmation of the peer entity
*/
```

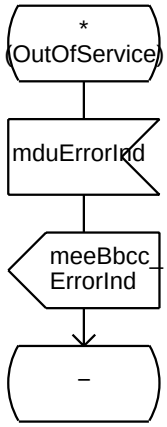
```
/* Timerdefinitions */
TIMER
T_BbccStartup := 180,   /* Default value for startup supervision timer is 180 sec, Tolerance +/− 10%  */
T_BbccRestart := 180;   /* Default value for restart supervision timer is 180 sec, Tolerance +/− 10%  */
```

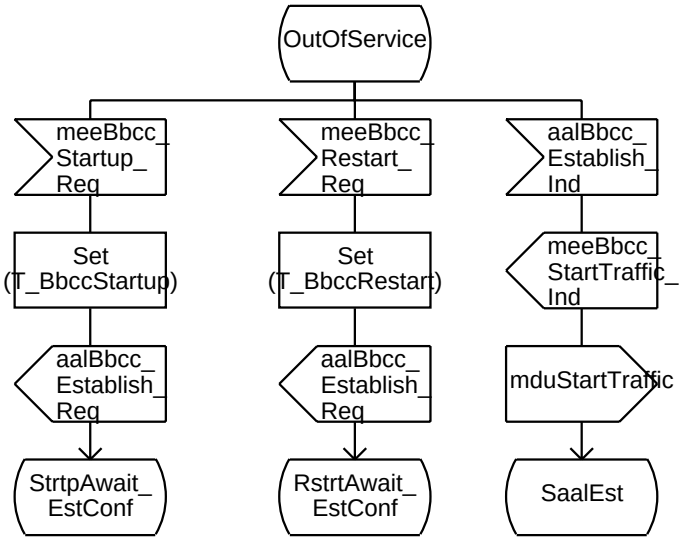
```
/* Data type definitions internal to the process */
/**/
```

```
/*Primitive Data Declarations*/
/**/
/* Reject Cause */
DCL
V_RejCause  IT_RejCause;
```

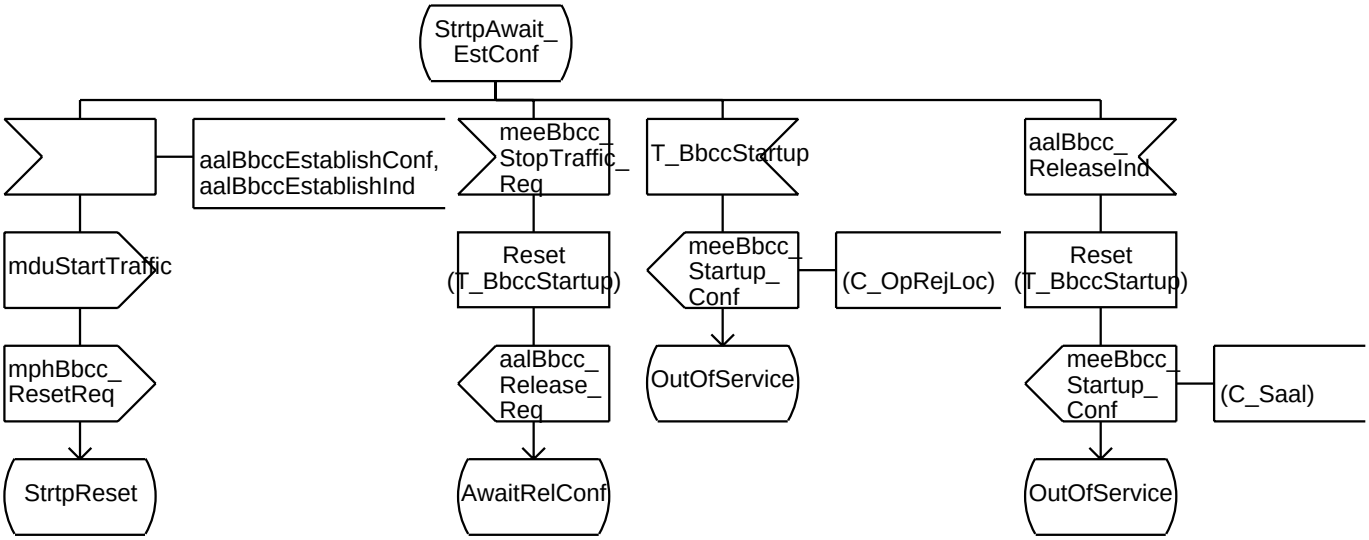


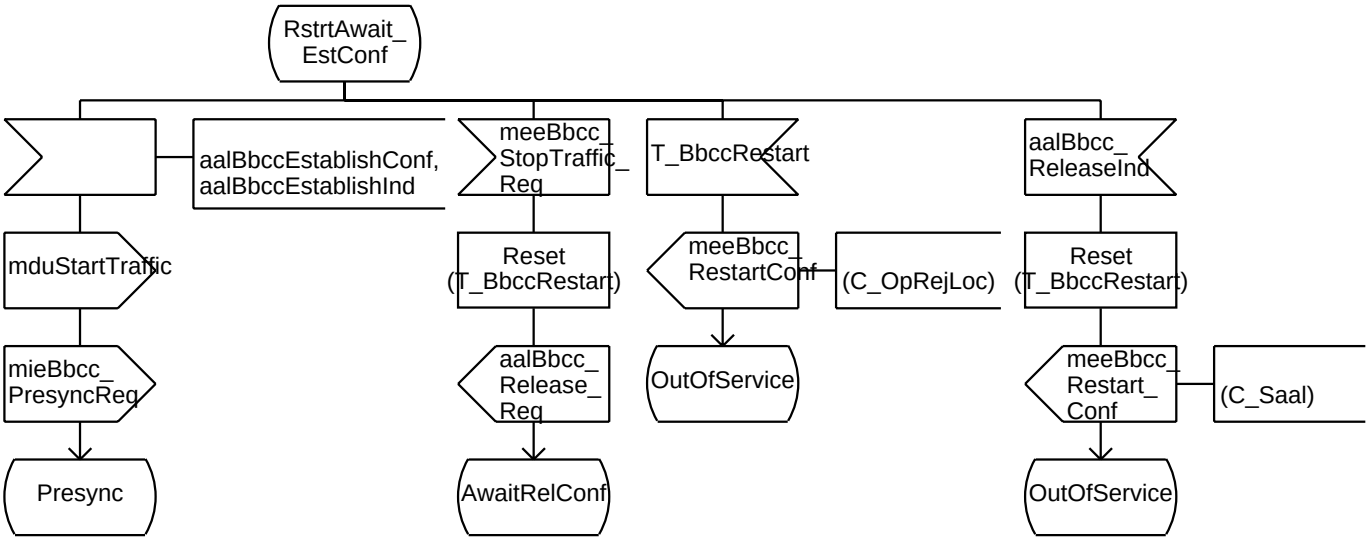
/* VB5.2 BCC management entity
Process CTR */



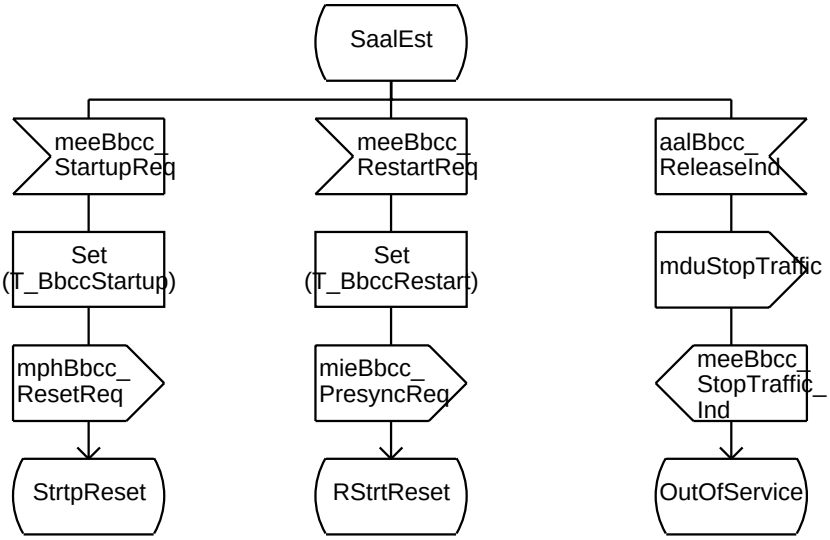


/* VB5.2 BCC management entity
Process CTR */

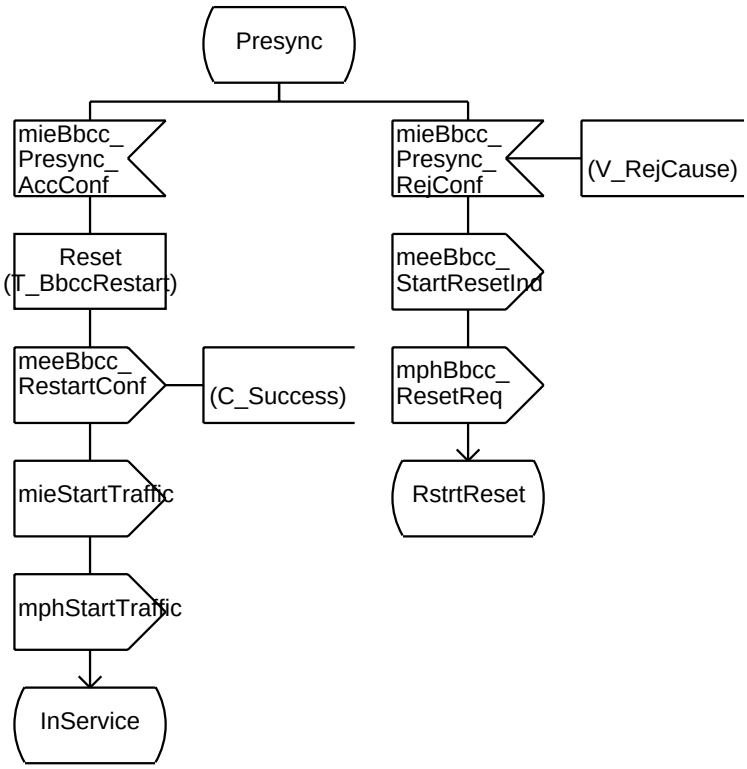


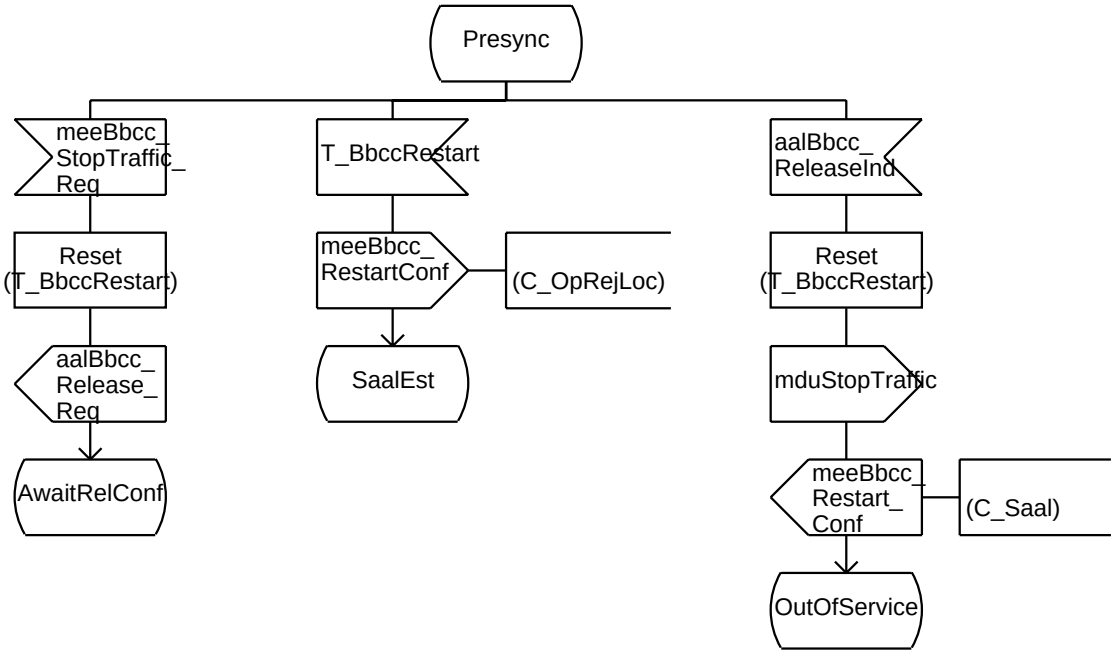


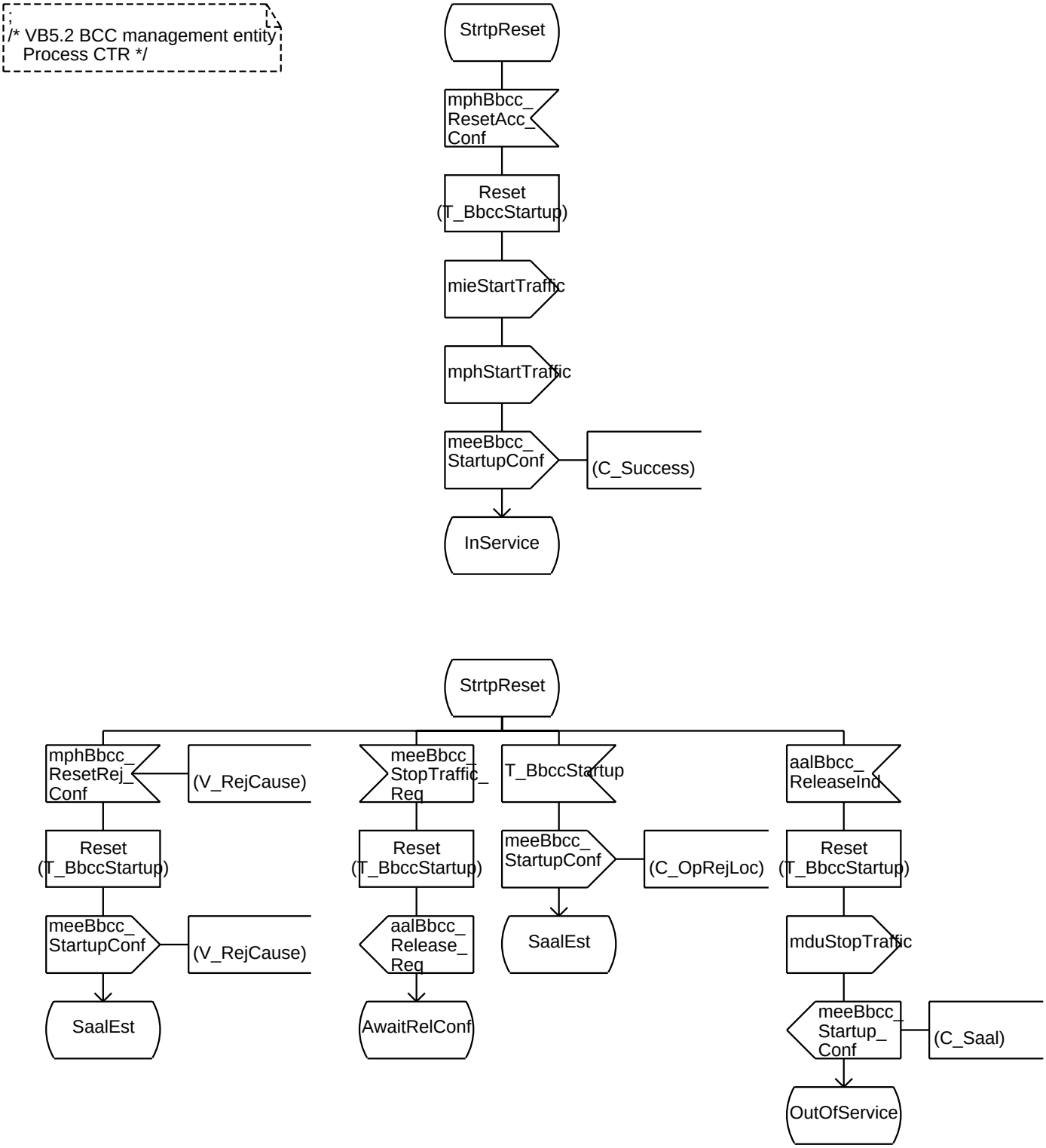
/* VB5.2 BCC management entity
Process CTR */

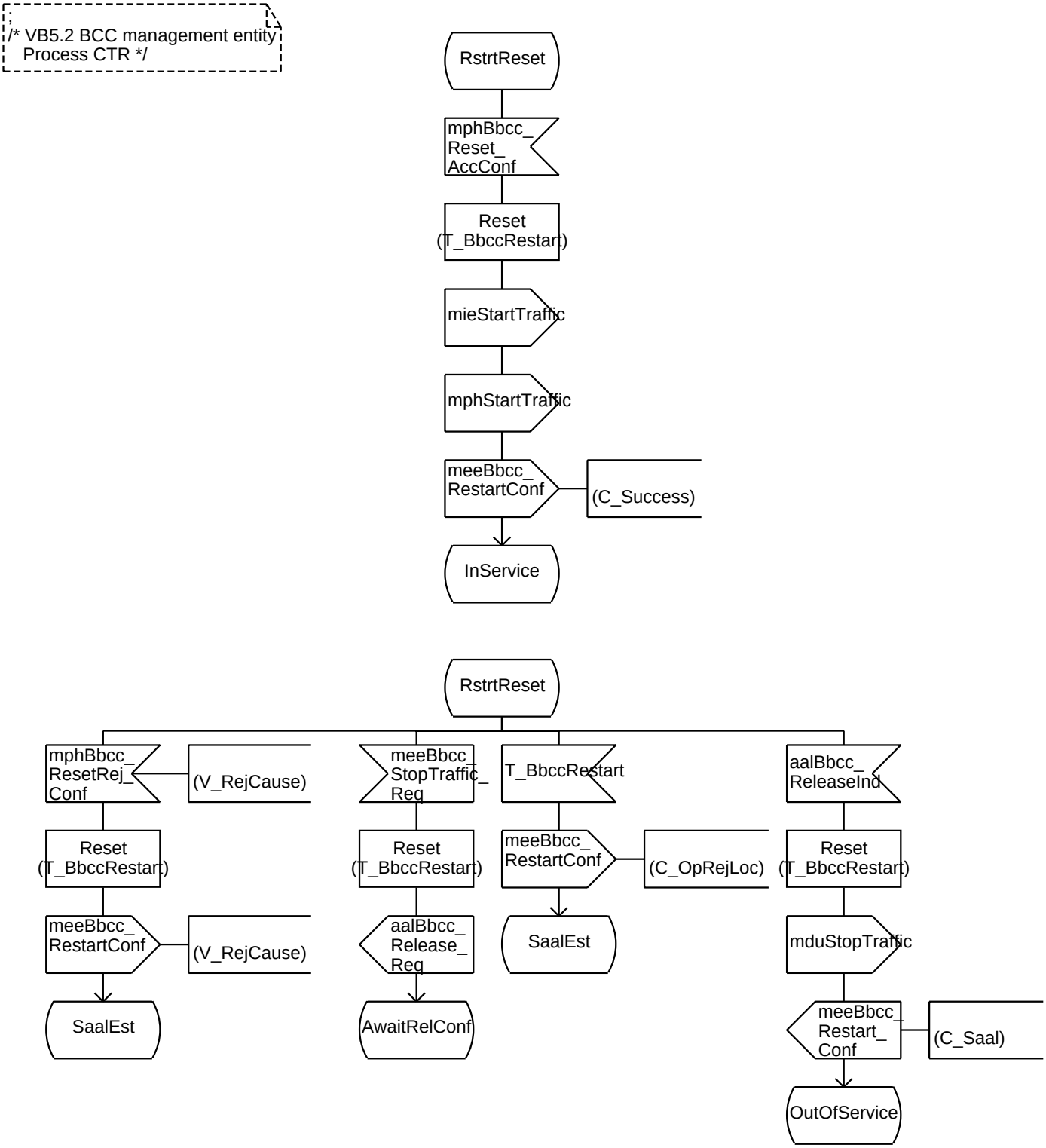


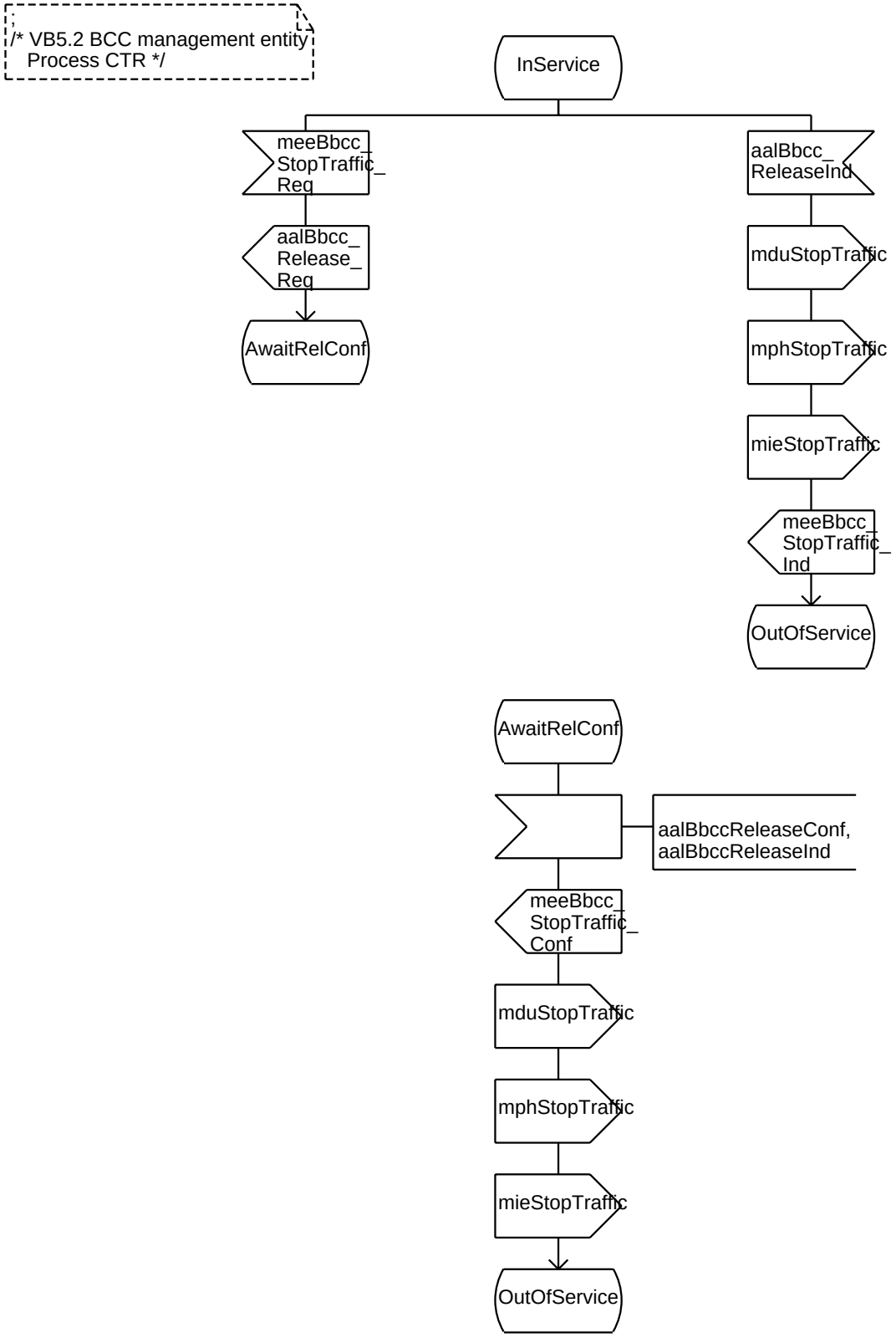
/* VB5.2 BCC management entity
Process CTR */



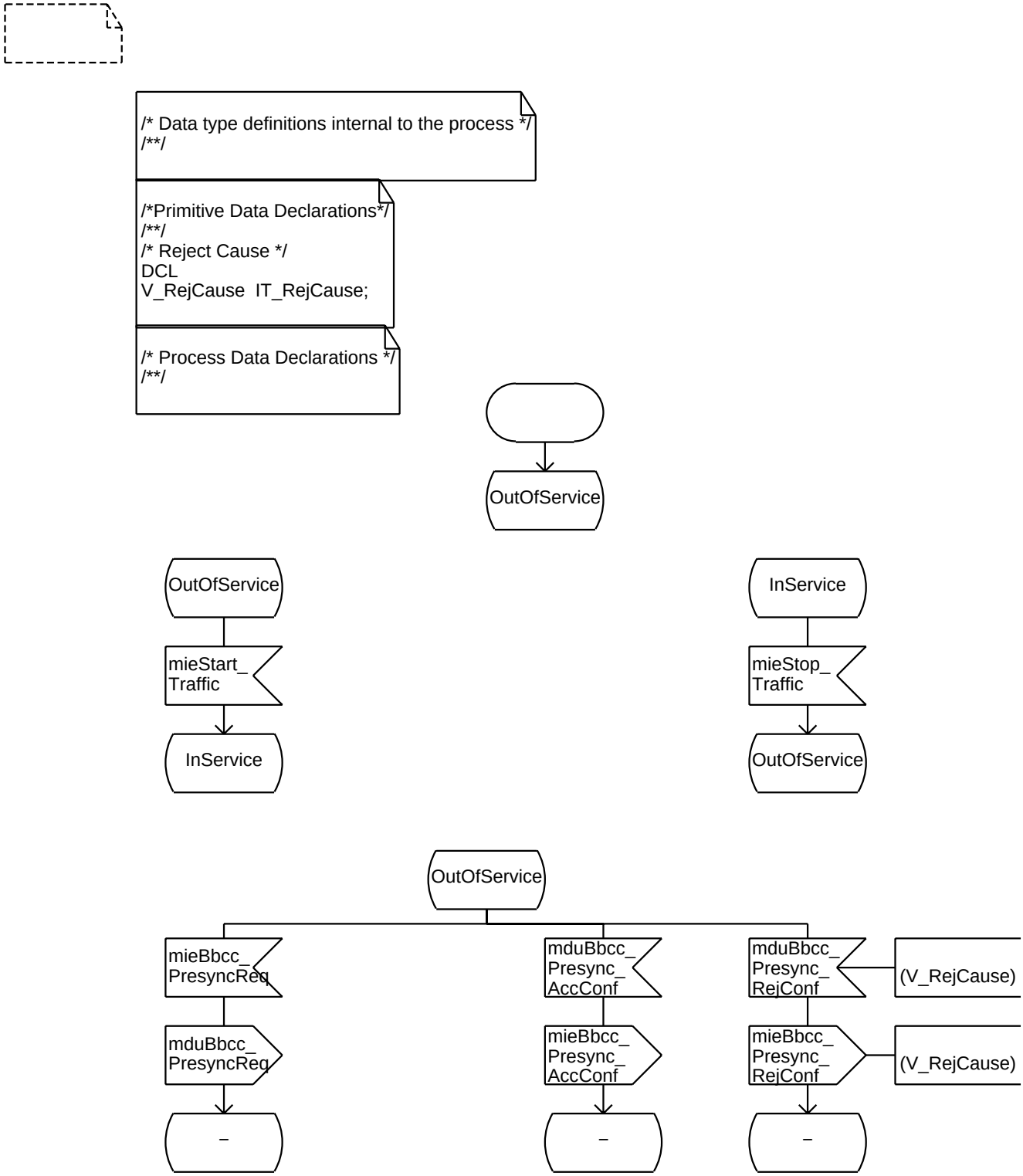


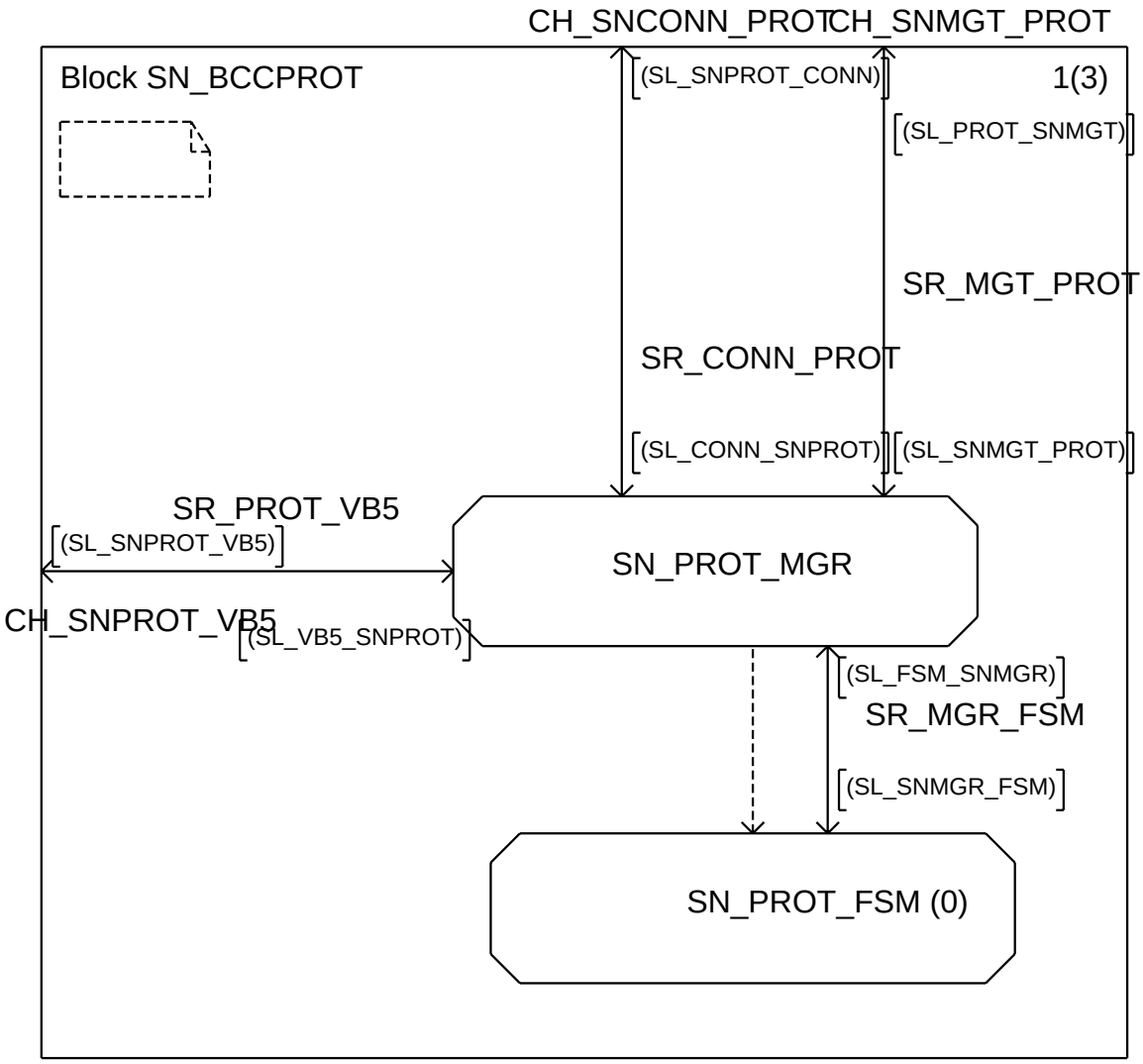

/* VB5.2 BCC management entity
Process CTR */
StrtpReset
mphBbcc_ ResetAcc_ Conf
Reset (T_BbccStartup)
mieStartTraffic
mphStartTraffic
meeBbcc_ StartupConf
(C_Success)
InService
StrtpReset
mphBbcc_ ResetRej_ Conf
(V_RejCause)
Reset (T_BbccStartup)
meeBbcc_ StartupConf
(V_RejCause)
SaalEst
meeBbcc_ StopTraffic_ Req
Reset (T_BbccStartup)
aalBbcc_ Release_ Req
AwaitRelConf
T_BbccStartup
meeBbcc_ StartupConf
(C_OpRejLoc)
SaalEst
aalBbcc_ ReleaseInd
Reset (T_BbccStartup)
mduStopTraffic
meeBbcc_ Startup_ Conf
(C_Saal)
OutOfService


/* VB5.2 BCC management entity
Process CTR */
RstrtReset
mphBbcc_
Reset_
AccConf
Reset
(T_BbccRestart)
mieStartTraffic
mphStartTraffic
meeBbcc_
RestartConf
(C_Success)
InService
RstrtReset
mphBbcc_
ResetRej_
Conf
(V_RejCause)
meeBbcc_
StopTraffic_
Req
T_BbccRestart
aalBbcc_
ReleaseInd
Reset
(T_BbccRestart)
Reset
(T_BbccRestart)
meeBbcc_
RestartConf
(C_OpRejLoc)
Reset
(T_BbccRestart)
meeBbcc_
RestartConf
(V_RejCause)
aalBbcc_
Release_
Req
SaalEst
mduStopTraffic
SaalEst
AwaitRelConf
meeBbcc_
Restart_
Conf
(C_Saal)
OutOfService


/* VB5.2 BCC management entity
Process CTR */
InService
meeBbcc_StopTraffic_Req
aalBbcc_Release_Req
AwaitRelConf
aalBbcc_ReleaseInd
mduStopTraffic
mphStopTraffic
mieStopTraffic
meeBbcc_StopTraffic_Ind
OutOfService
AwaitRelConf
aalBbccReleaseConf,
aalBbccReleaseInd
meeBbcc_StopTraffic_Conf
mduStopTraffic
mphStopTraffic
mieStopTraffic
OutOfService

Process SN_PRESYNC 1(1)




CH_SNCONN_PROT
CH_SNMGT_PROT
Block SN_BCCPROT
1(3)
(SL_SNPROT_CONN)
(SL_PROT_SNMGT)
SR_MGT_PROT
SR_CONN_PROT
(SL_CONN_SNPROT)
(SL_SNMGT_PROT)
SR_PROT_VB5
(SL_SNPROT_VB5)
SN_PROT_MGR
CH_SNPROT_VB5
(SL_VB5_SNPROT)
(SL_FSM_SNMGR)
SR_MGR_FSM
(SL_SNMGR_FSM)
SN_PROT_FSM (0)

/* Declaration of primitives internal
to block SN_BCCPROT */

```
signal
mieAllocReq            (IT_TransId, IT_ConnRefNo, IT_BranchId),
mieAllocAccConf        (IT_ConnRefNo, IT_BranchId),
mieAllocRejConf        (IT_ConnRefNo, IT_BranchId, IT_RejCause),
mieAllocCompReq        (IT_TransId, IT_ConnRefNo),
mieAllocCompAccConf    (IT_ConnRefNo),
mieAllocCompRejConf    (IT_ConnRefNo, IT_RejCause),
mieBbccResetReq        (IT_TransId),
mieBbccResetAccConf    ,
mieBbccResetRejConf    (IT_RejCause),
mieBbccPresyncReq      (IT_TransId),
mieBbccPresyncAccConf  ,
mieBbccPresyncRejConf  (IT_RejCause),
mieDeallocReq          (IT_TransId, IT_ConnRefNo),
mieDeallocAccConf      (IT_ConnRefNo),
mieDeallocRejConf      (IT_ConnRefNo, IT_RejCause),
mieModifyReq           (IT_TransId, IT_ConnRefNo),
mieModifyAccConf       (IT_ConnRefNo),
mieModifyRejConf       (IT_ConnRefNo, IT_RejCause),
mieModifyCompReq       (IT_TransId, IT_ConnRefNo),
mieModifyCompAccConf   (IT_ConnRefNo),
mieModifyCompRejConf   (IT_ConnRefNo, IT_RejCause),
mieModifyAbortReq      (IT_TransId, IT_ConnRefNo),
mieModifyAbortAccConf  (IT_ConnRefNo),
mieModifyAbortRejConf  (IT_ConnRefNo, IT_RejCause),
mieAddBranchReq        (IT_TransId, IT_ConnRefNo, IT_BranchId),
mieAddBranchAccConf    (IT_ConnRefNo, IT_BranchId),
mieAddBranchRejConf    (IT_ConnRefNo, IT_BranchId, IT_RejCause),
mieUpdateBranchReq     (IT_TransId, IT_ConnRefNo, IT_BranchId),
mieUpdateBranchAccConf (IT_ConnRefNo, IT_BranchId),
mieUpdateBranchRejConf (IT_ConnRefNo, IT_BranchId, IT_RejCause),
mieDropBranchReq       (IT_TransId, IT_ConnRefNo, IT_BranchId),
mieDropBranchAccConf   (IT_ConnRefNo, IT_BranchId),
mieDropBranchRejConf   (IT_ConnRefNo, IT_BranchId, IT_RejCause),
mieAnFaultInd          (IT_ConnRefNo, IT_BranchId);
```

```
/* Declaration of signallists
   internal to block SN_BCCPROT */
```

```
signallist SL_FSM_SNMGR =
bccAlloc,
bccAllocComp,
bccBbccReset,
bccBbccPresync,
bccDealloc,
bccModify,
bccModifyComp,
bccModifyAbort,
bccAddBranch,
bccUpdateBranch,
bccDropBranch,
bccAnFaultAcc,
mieAllocAccConf,
mieAllocRejConf,
mieAllocCompAccConf,
mieAllocCompRejConf,
mieBbccResetAccConf,
mieBbccResetRejConf,
mieBbccPresyncAccConf,
mieBbccPresyncRejConf,
mieDeallocAccConf,
mieDeallocRejConf,
mieModifyAccConf,
mieModifyRejConf,
mieModifyCompAccConf,
mieModifyCompRejConf,
mieModifyAbortAccConf,
mieModifyAbortRejConf,
mieAddBranchAccConf,
mieAddBranchRejConf,
mieUpdateBranchAccConf,
mieUpdateBranchRejConf,
mieDropBranchAccConf,
mieDropBranchRejConf,
mieAnFaultInd,
mieProtocolError,
mieErrorInd;
```

```
signallist SL_SNMGR_FSM =
mieAllocReq,
mieAllocCompReq,
mieBbccResetReq,
mieBbccPresyncReq,
mieDeallocReq,
mieModifyReq,
mieModifyCompReq,
mieModifyAbortReq,
mieAddBranchReq,
mieUpdateBranchReq,
mieDropBranchReq,
bccAllocAcc,
bccAllocRej,
bccAllocCompAcc,
bccAllocCompRej,
bccBbccResetAcc,
bccBbccResetRej,
bccBbccPresyncAcc,
bccBbccPresyncRej,
bccDeallocAcc,
bccModifyAcc,
bccModifyRej,
bccModifyCompAcc,
bccModifyCompRej,
bccModifyAbortAcc,
bccModifyAbortRej,
bccAddBranchAcc,
bccAddBranchRej,
bccUpdateBranchAcc,
bccUpdateBranchRej,
bccDropBranchAcc,
bccDropBranchRej,
bccAnFault,
bccProtocolError,
mieAbortReq;
```

```
;
/*SN BCC protocol management entity
  controls instantiation of
  protocol FSMs (process SN_PROT_FSM) */
```

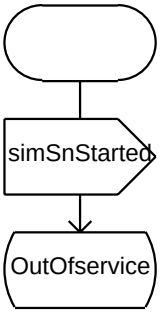
```
/* The generation of the transaction identifier, i.e. the increment by one
   for each invoke message, is for simulation purposes only. In an
   implementation, any other method which is appropriate may be applied */
```

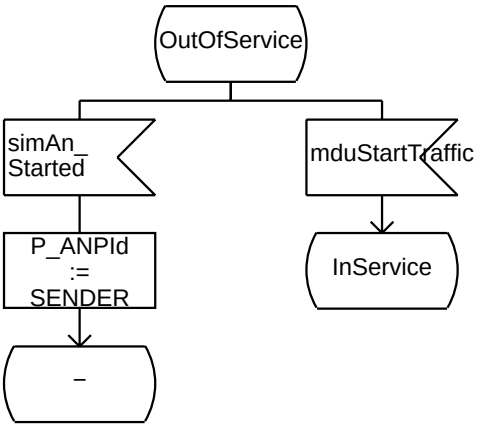
```
/* Data type definitions internal to the process */
/**/
/* This data type is used for translation of the remote transaction identifier to the process identifier (PID)
   of the process handling the corresponding transaction in the SN */
NEWTYPE AT_RemTransList ARRAY
  (IT_TransId, PID)
ENDNEWTYPE AT_RemTransList;
/**/
/* This data type is used for translation of the local transaction identifier to the process identifier (PID)
   of the process handling the corresponding transaction in the SN */
NEWTYPE AT_LocTransList ARRAY
  (IT_TransId, PID)
ENDNEWTYPE AT_LocTransList;
```

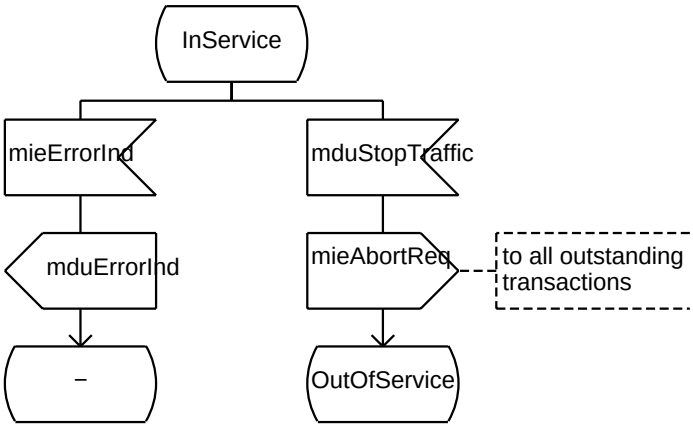
```
/*Primitive Data Declarations*/
/**/
/* Protocol error cause */
DCL
V_ProtErrCause IT_ProtErrCause;
/**/
/* Transaction identifier */
DCL
V_TransId   IT_TransId;
/**/
/* Connection reference number */
DCL
V_ConnRefNo  IT_ConnRefNo;
/**/
/* Branch identifier */
DCL
V_BranchId  IT_BranchId;
/**/
/* Reject cause */
DCL
V_RejCause  IT_RejCause;
```

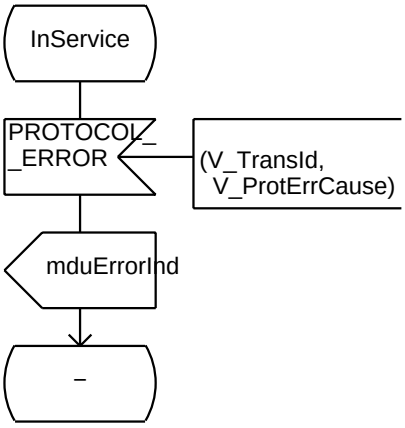
```
/* Process Data Declarations */
/**/
/* List of SN_PROT_FSMs */
DCL
A_RemTransList AT_RemTransList,
A_LocTransList AT_LocTransList;
/**/
/* PID of the receiver process in the AN */
DCL
P_ANPId   PID;
/* Counter for transaction identifier */
DCL
V_NewTransId IT_TransId;
```

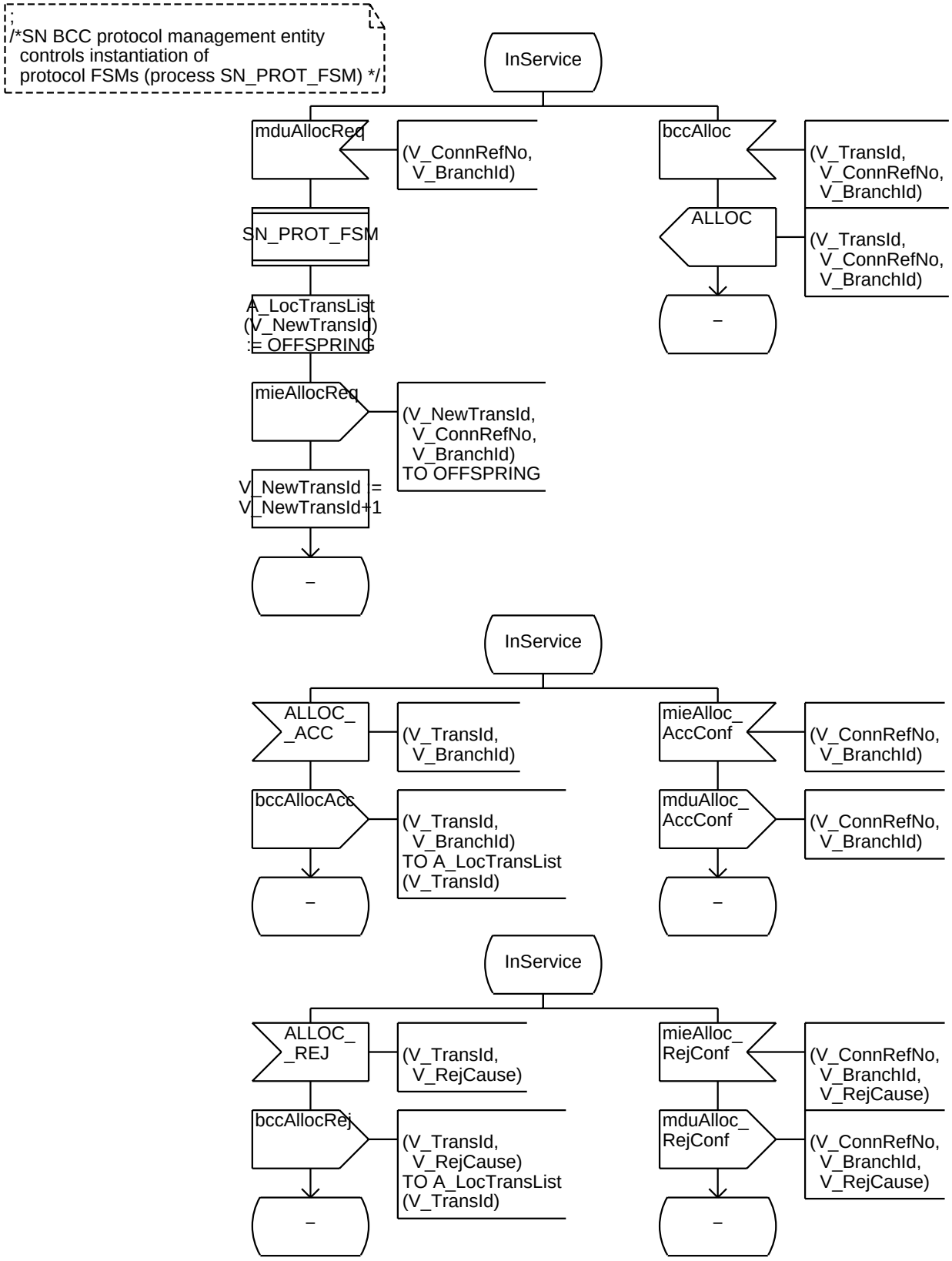
;
/*SN BCC protocol management entity
controls instantiation of
protocol FSMs (process SN_PROT_FSM) */

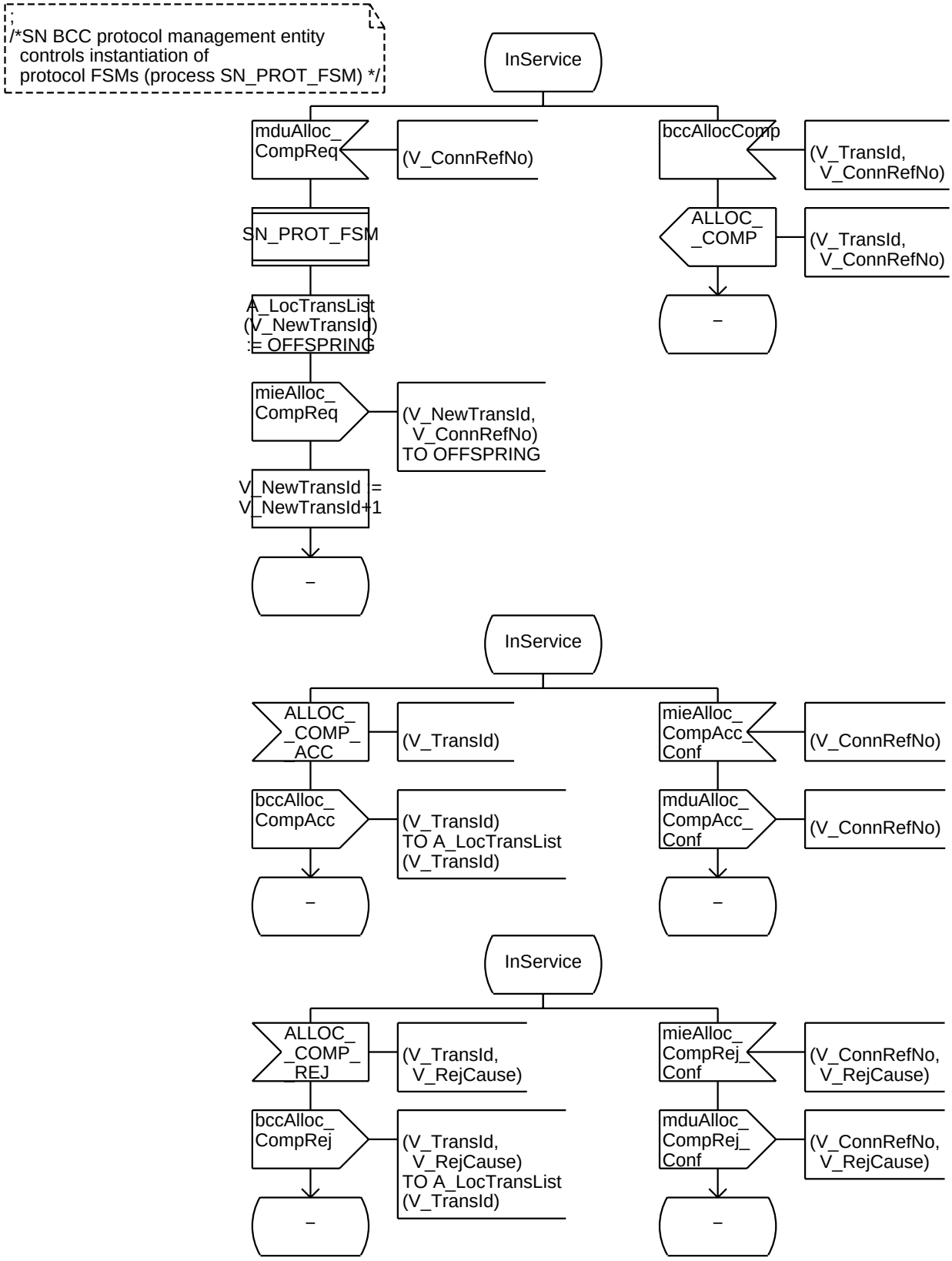


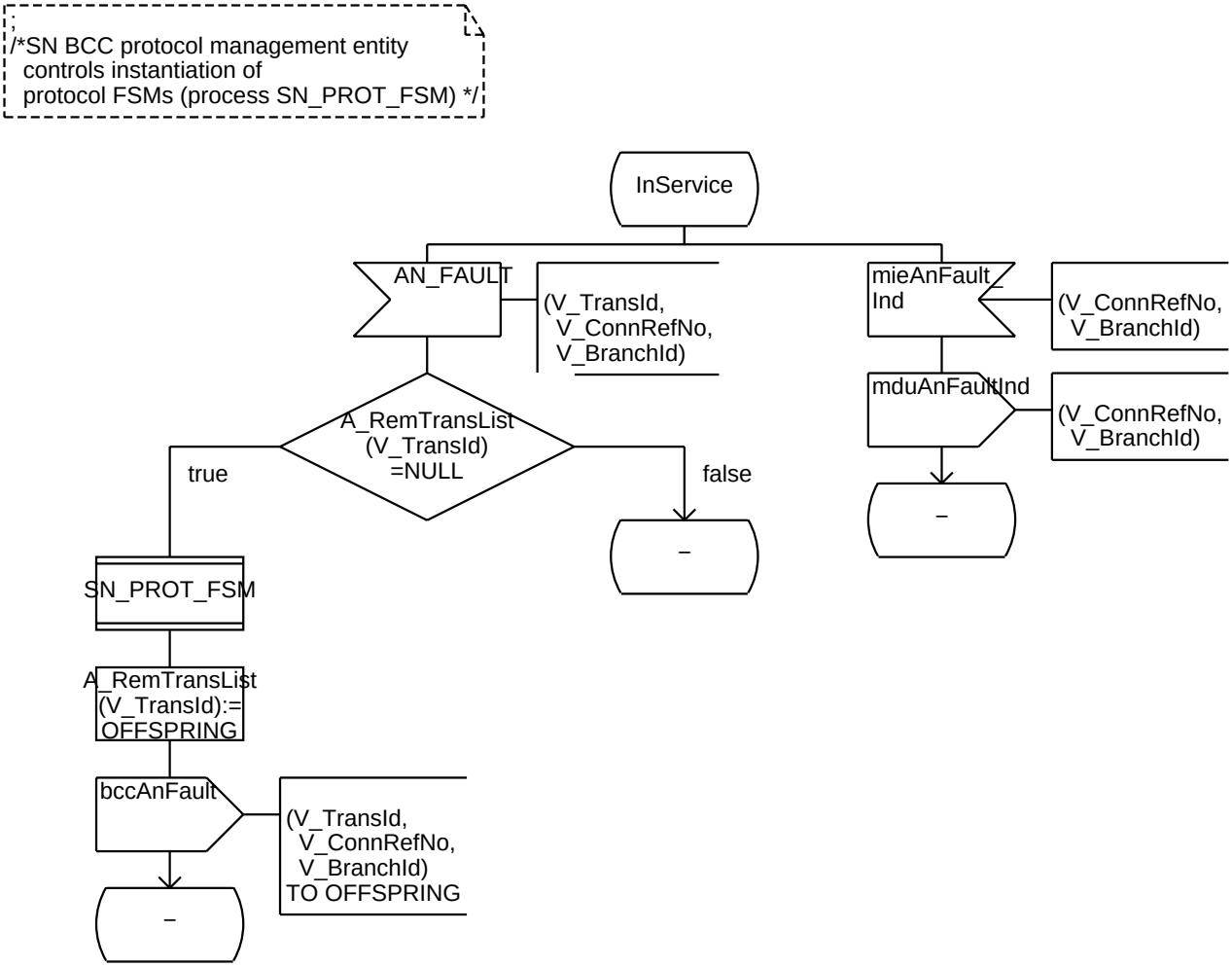


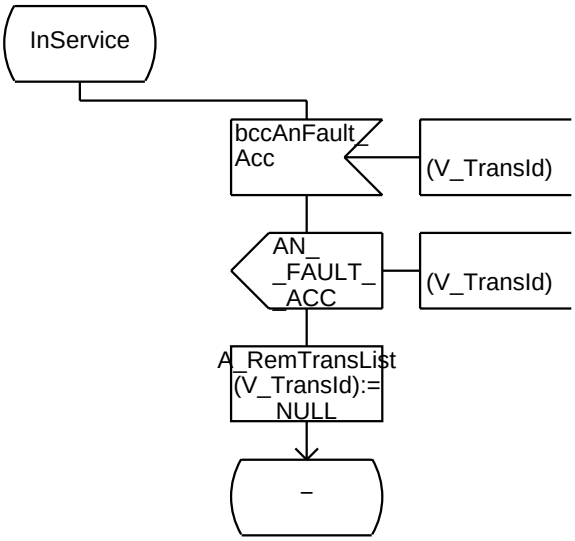


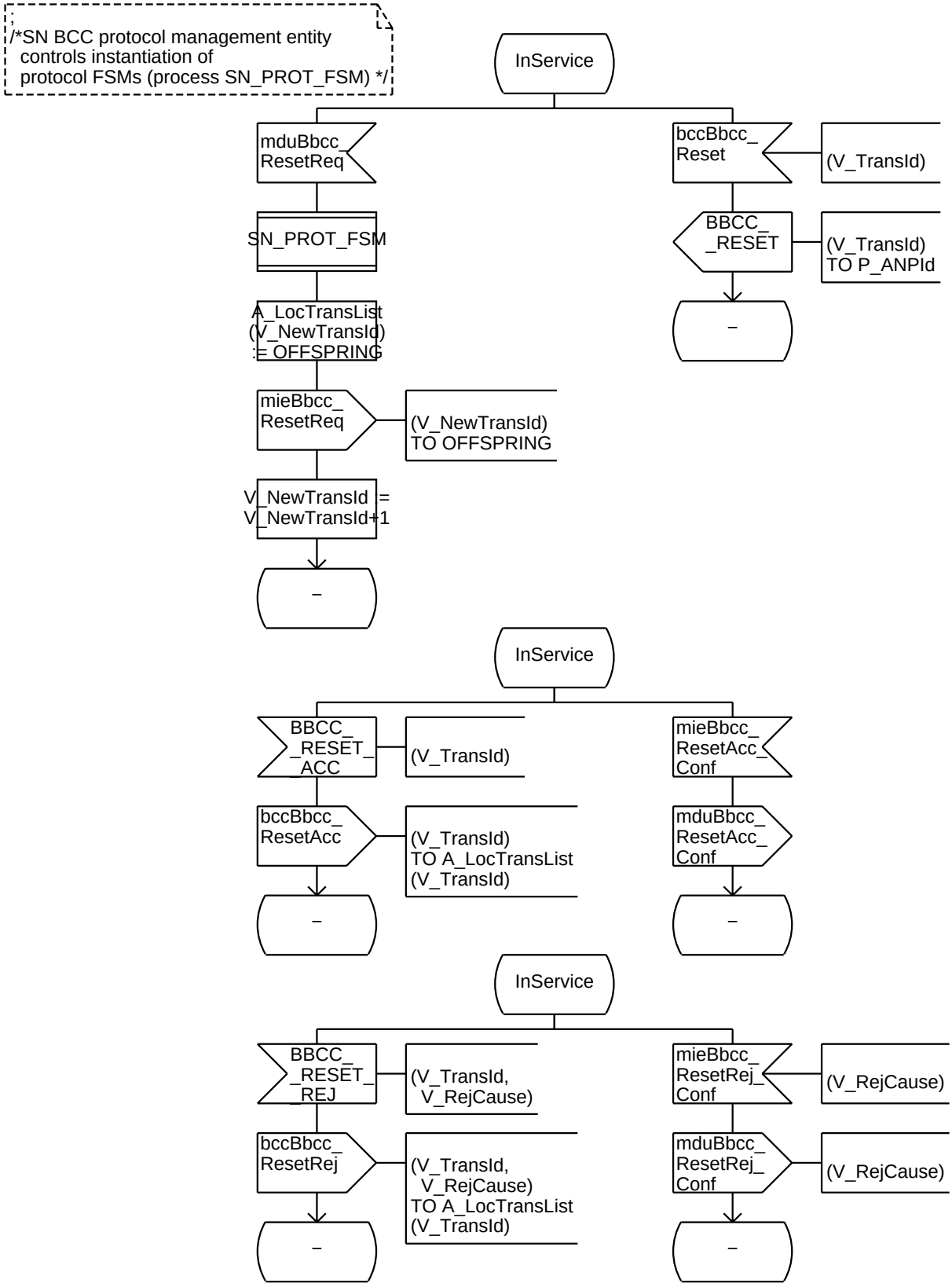


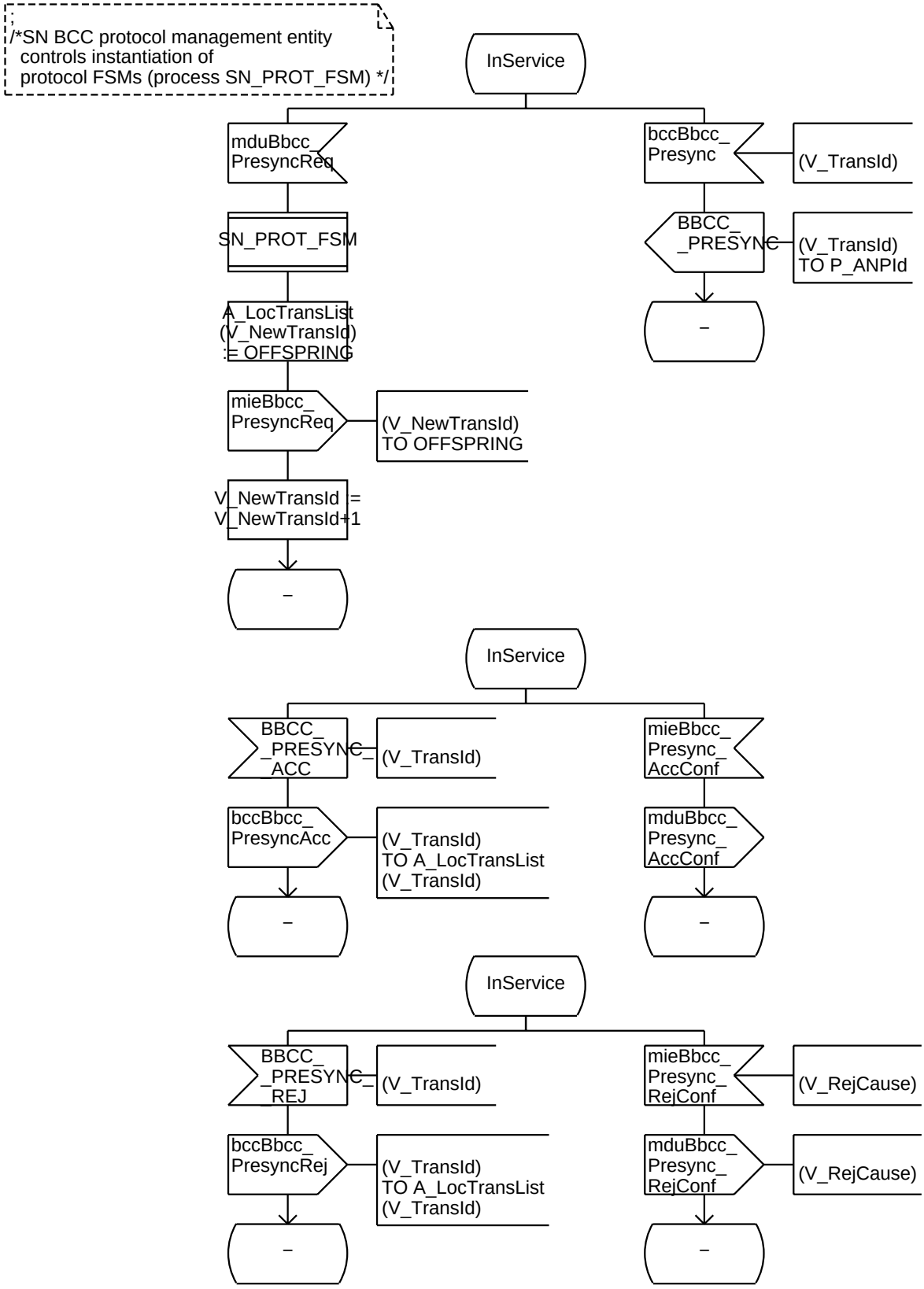

/*SN BCC protocol management entity
controls instantiation of
protocol FSMs (process SN_PROT_FSM) */
InService
mduAllocReq
(V_ConnRefNo,
V_BranchId)
SN_PROT_FSM
A_LocTransList
(V_NewTransId)
:= OFFSPRING
mieAllocReq
(V_NewTransId,
V_ConnRefNo,
V_BranchId)
TO OFFSPRING
V_NewTransId :=
V_NewTransId+1
–
bccAlloc
(V_TransId,
V_ConnRefNo,
V_BranchId)
ALLOC
(V_TransId,
V_ConnRefNo,
V_BranchId)
–
InService
ALLOC_
_ACC
(V_TransId,
V_BranchId)
bccAllocAcc
(V_TransId,
V_BranchId)
TO A_LocTransList
(V_TransId)
–
mieAlloc_
AccConf
(V_ConnRefNo,
V_BranchId)
mduAlloc_
AccConf
(V_ConnRefNo,
V_BranchId)
–
InService
ALLOC_
_REJ
(V_TransId,
V_RejCause)
bccAllocRej
(V_TransId,
V_RejCause)
TO A_LocTransList
(V_TransId)
–
mieAlloc_
RejConf
(V_ConnRefNo,
V_BranchId,
V_RejCause)
mduAlloc_
RejConf
(V_ConnRefNo,
V_BranchId,
V_RejCause)
–

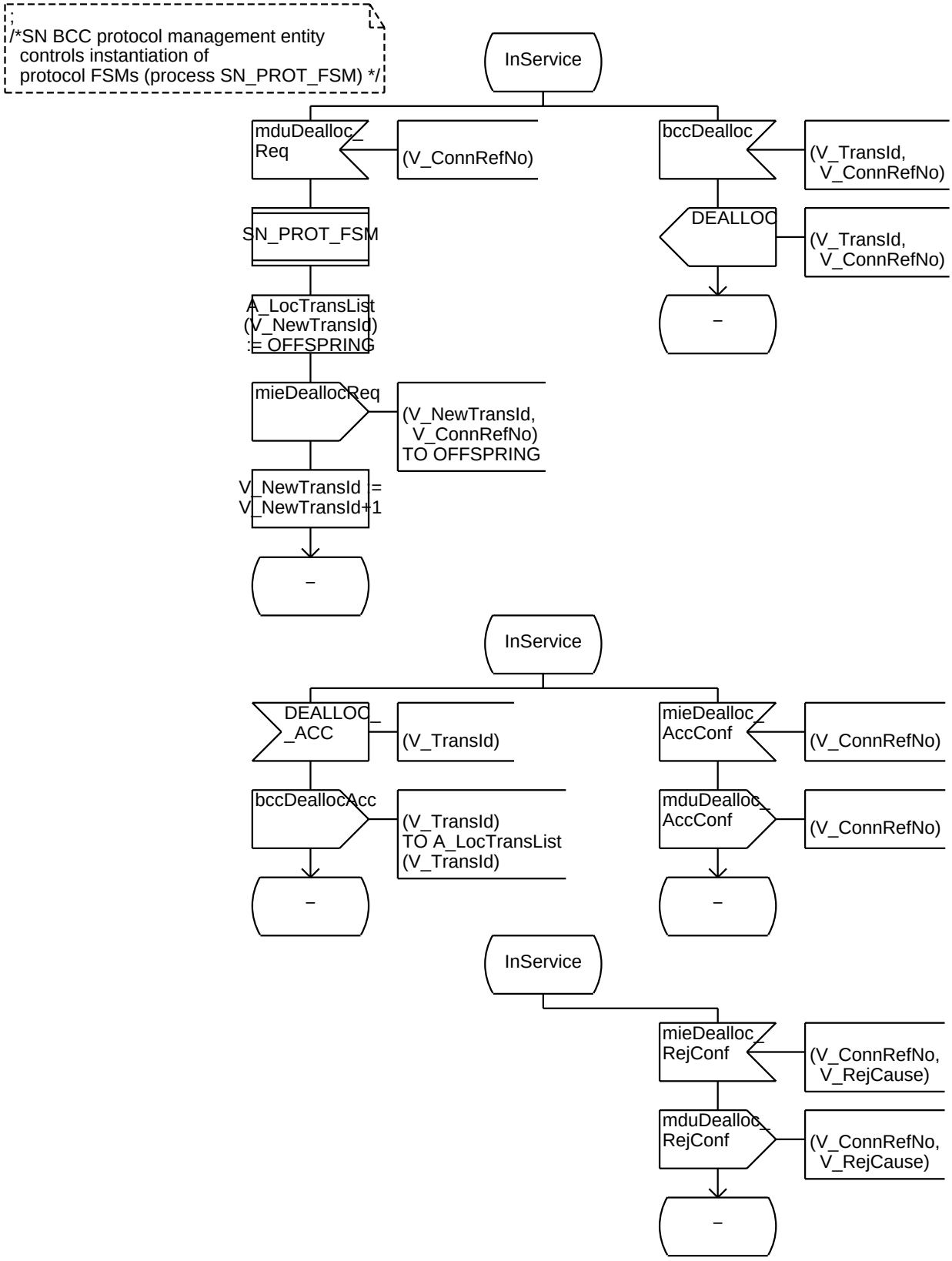

;
/*SN BCC protocol management entity
controls instantiation of
protocol FSMs (process SN_PROT_FSM) */
InService
mduAlloc_
CompReq
(V_ConnRefNo)
SN_PROT_FSM
A_LocTransList
(V_NewTransId)
:= OFFSPRING
mieAlloc_
CompReq
(V_NewTransId,
V_ConnRefNo)
TO OFFSPRING
V_NewTransId :=
V_NewTransId+1
–
bccAllocComp
(V_TransId,
V_ConnRefNo)
ALLOC_
_COMP
(V_TransId,
V_ConnRefNo)
–
InService
ALLOC_
COMP
_ACC
(V_TransId)
bccAlloc_
CompAcc
(V_TransId)
TO A_LocTransList
(V_TransId)
–
mieAlloc_
CompAcc_
Conf
(V_ConnRefNo)
mduAlloc_
CompAcc_
Conf
(V_ConnRefNo)
–
InService
ALLOC_
COMP
_REJ
(V_TransId,
V_RejCause)
bccAlloc_
CompRej
(V_TransId,
V_RejCause)
TO A_LocTransList
(V_TransId)
–
mieAlloc_
CompRej_
Conf
(V_ConnRefNo,
V_RejCause)
mduAlloc_
CompRej_
Conf
(V_ConnRefNo,
V_RejCause)
–

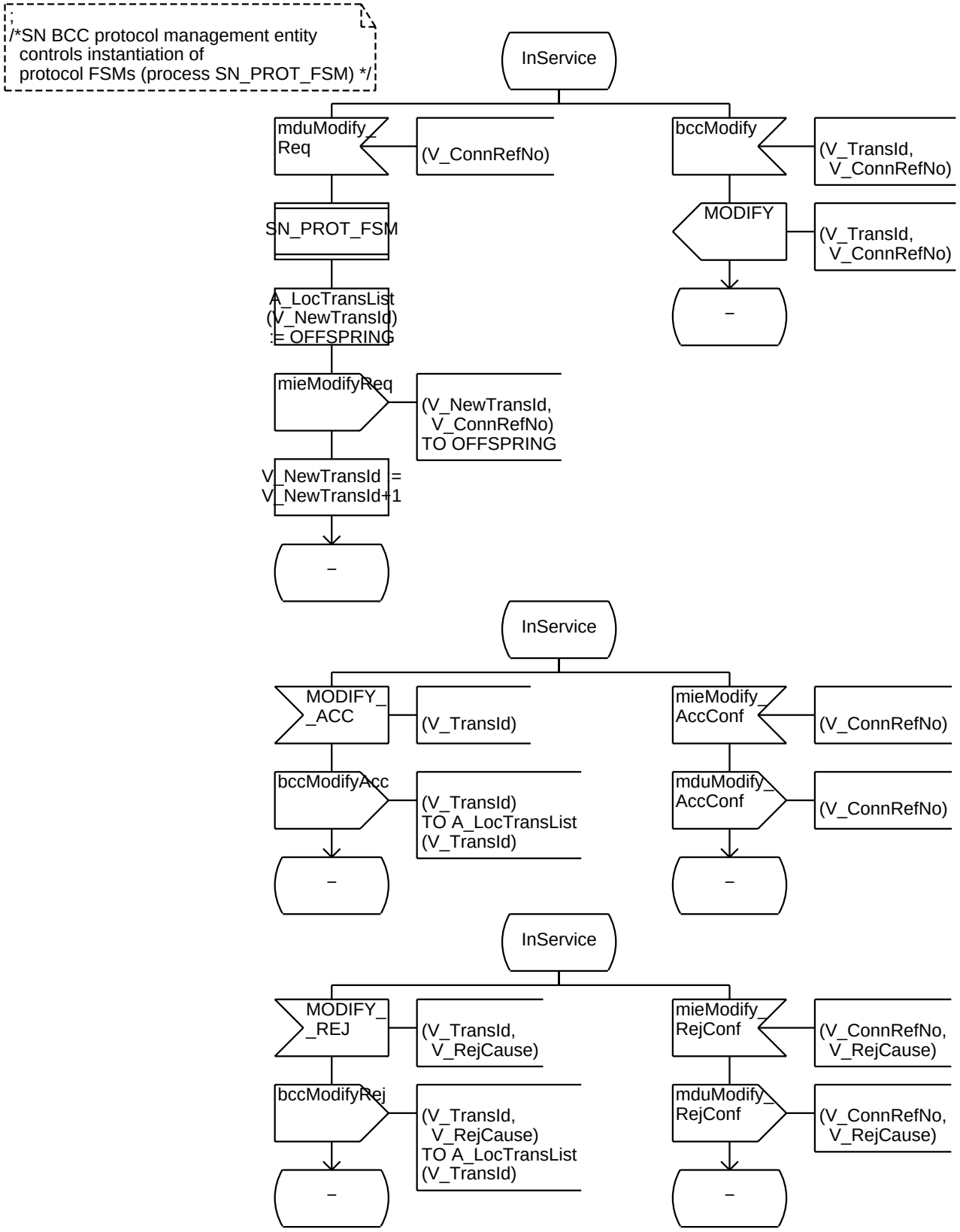

;
/*SN BCC protocol management entity
controls instantiation of
protocol FSMs (process SN_PROT_FSM) */
InService
AN_FAULT
(V_TransId,
V_ConnRefNo,
V_BranchId)
mieAnFault_
Ind
(V_ConnRefNo,
V_BranchId)
mduAnFaultInd
(V_ConnRefNo,
V_BranchId)
A_RemTransList
(V_TransId)
=NULL
true
false
–
–
SN_PROT_FSM
A_RemTransList
(V_TransId):=
OFFSPRING
bccAnFault
(V_TransId,
V_ConnRefNo,
V_BranchId)
TO OFFSPRING
–

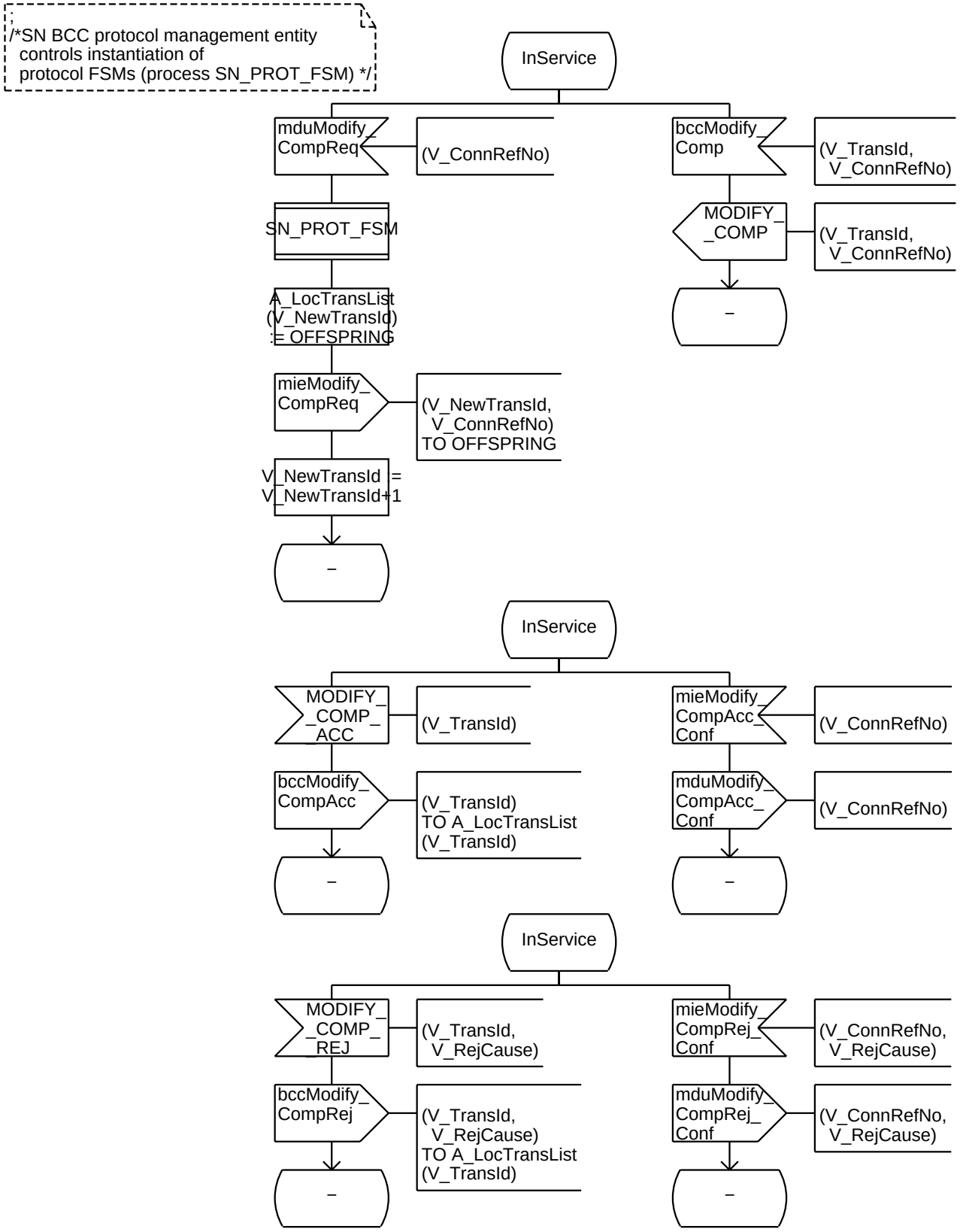

InService
bccAnFault_
Acc
(V_TransId)
AN_
FAULT
_ACC
(V_TransId)
A_RemTransList
(V_TransId):=
NULL
–

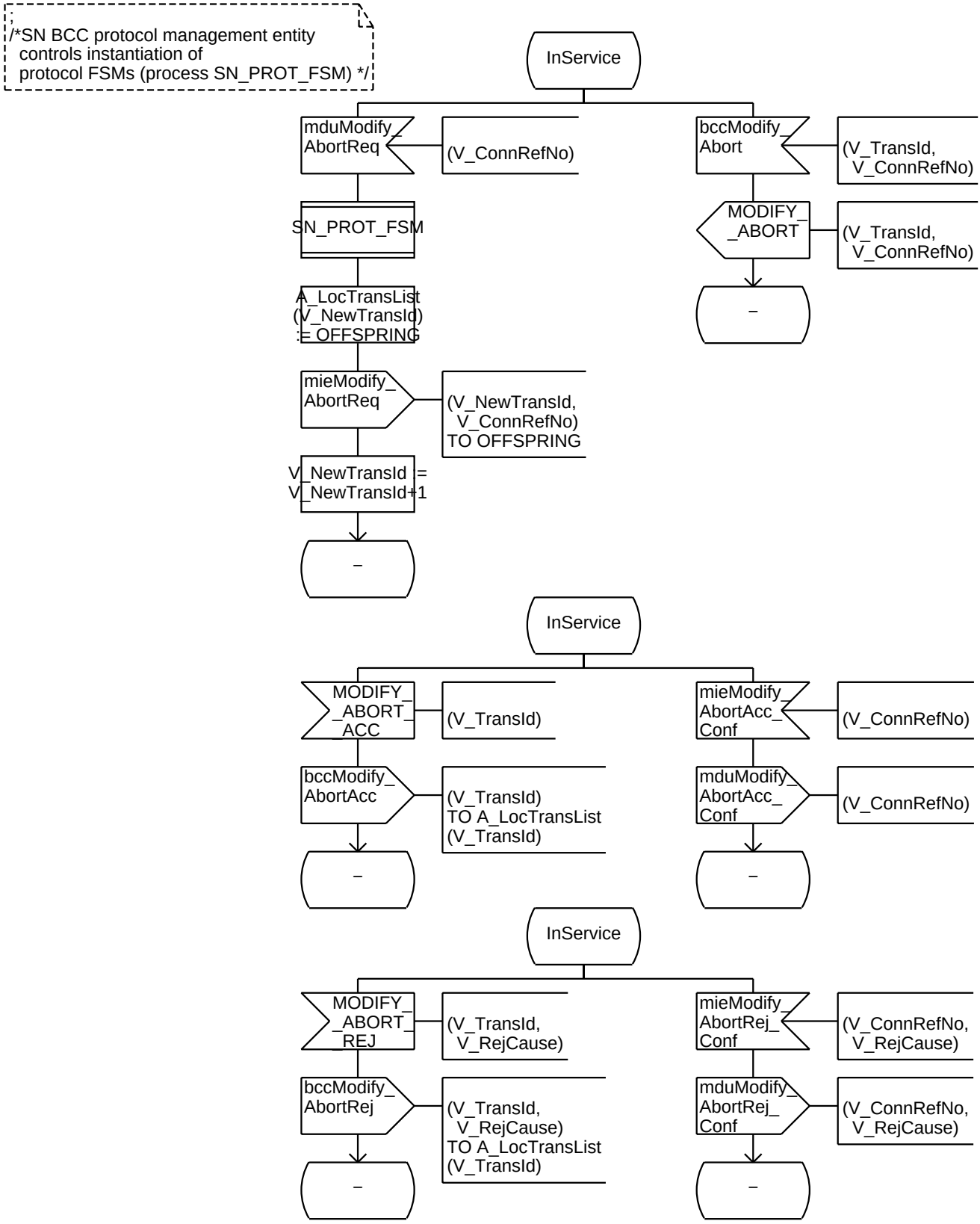

/*SN BCC protocol management entity
controls instantiation of
protocol FSMs (process SN_PROT_FSM) */
InService
mduBbcc_
ResetReq
SN_PROT_FSM
A_LocTransList
(V_NewTransId)
:= OFFSPRING
mieBbcc_
ResetReq
(V_NewTransId)
TO OFFSPRING
V_NewTransId =
V_NewTransId+1
–
bccBbcc_
Reset
(V_TransId)
BBCC_
_RESET
(V_TransId)
TO P_ANPId
–
InService
BBCC_
RESET
_ACC
(V_TransId)
bccBbcc_
ResetAcc
(V_TransId)
TO A_LocTransList
(V_TransId)
–
mieBbcc_
ResetAcc_
Conf
mduBbcc_
ResetAcc_
Conf
–
InService
BBCC_
RESET
_REJ
(V_TransId,
V_RejCause)
bccBbcc_
ResetRej
(V_TransId,
V_RejCause)
TO A_LocTransList
(V_TransId)
–
mieBbcc_
ResetRej_
Conf
(V_RejCause)
mduBbcc_
ResetRej_
Conf
(V_RejCause)
–

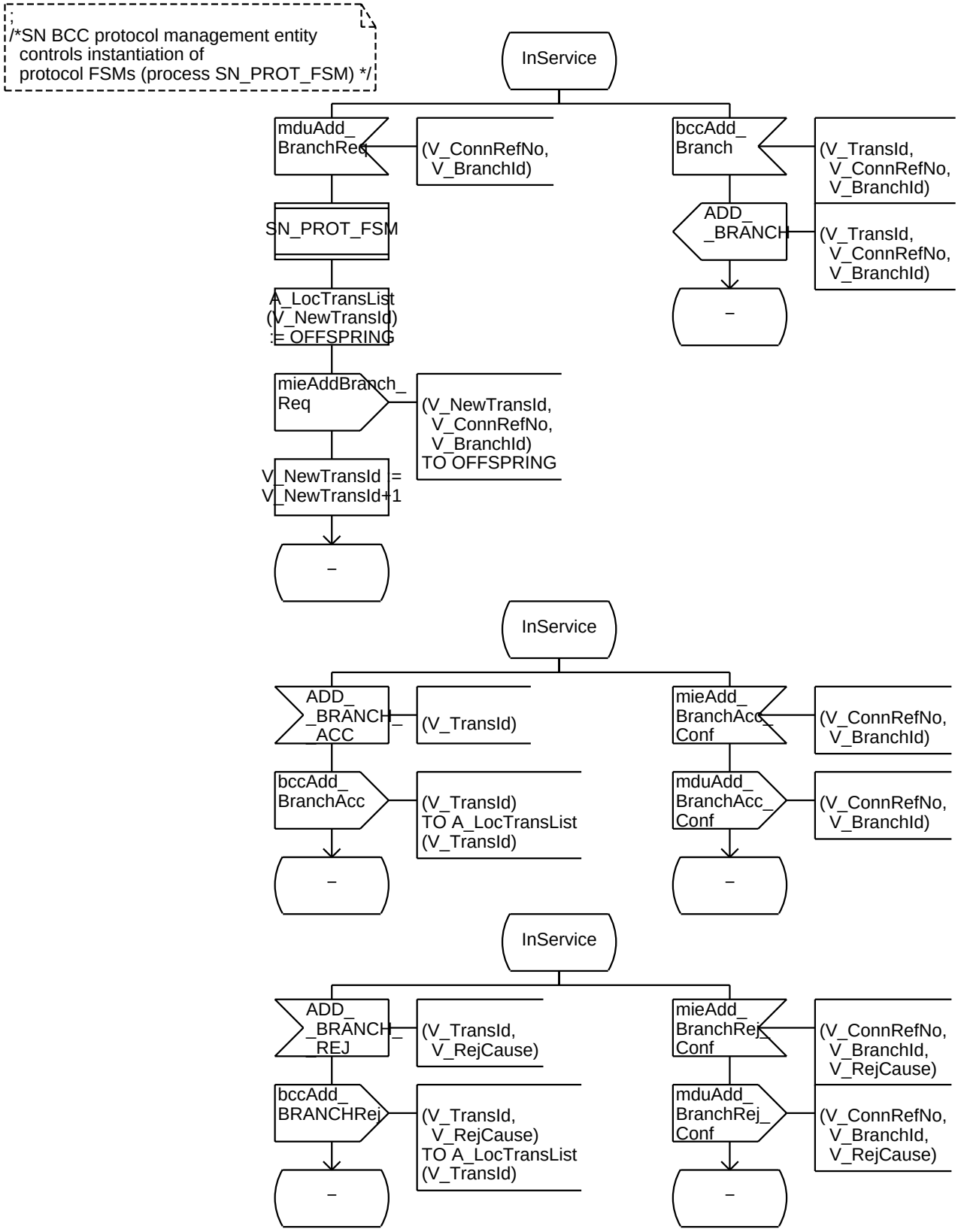

/*SN BCC protocol management entity
controls instantiation of
protocol FSMs (process SN_PROT_FSM) */
InService
mduBbcc_
PresyncReq
SN_PROT_FSM
A_LocTransList
(V_NewTransId)
:= OFFSPRING
mieBbcc_
PresyncReq
(V_NewTransId)
TO OFFSPRING
V_NewTransId =
V_NewTransId+1
–
bccBbcc_
Presync
(V_TransId)
BBCC_
_PRESYNC
(V_TransId)
TO P_ANPId
–
InService
BBCC_
PRESYNC
_ACC
(V_TransId)
bccBbcc_
PresyncAcc
(V_TransId)
TO A_LocTransList
(V_TransId)
–
mieBbcc_
Presync_
AccConf
mduBbcc_
Presync_
AccConf
–
InService
BBCC_
PRESYNC
_REJ
(V_TransId)
bccBbcc_
PresyncRej
(V_TransId)
TO A_LocTransList
(V_TransId)
–
mieBbcc_
Presync_
RejConf
(V_RejCause)
mduBbcc_
Presync_
RejConf
(V_RejCause)
–

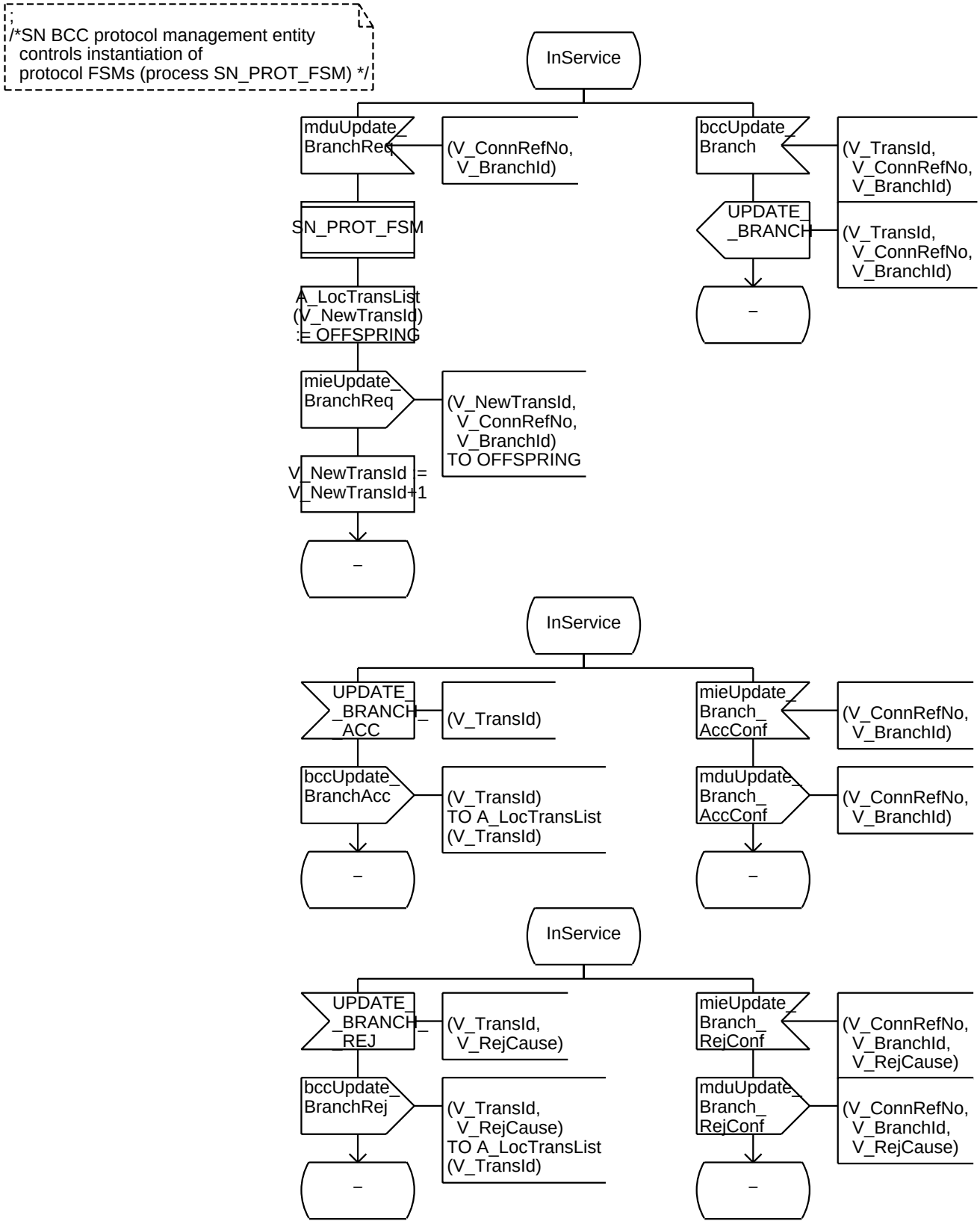

;
/*SN BCC protocol management entity
controls instantiation of
protocol FSMs (process SN_PROT_FSM) */
InService
mduDealloc_
Req
(V_ConnRefNo)
SN_PROT_FSM
A_LocTransList
(V_NewTransId)
:= OFFSPRING
mieDeallocReq
(V_NewTransId,
V_ConnRefNo)
TO OFFSPRING
V_NewTransId :=
V_NewTransId+1
–
bccDealloc
(V_TransId,
V_ConnRefNo)
DEALLOC
(V_TransId,
V_ConnRefNo)
–
InService
DEALLOC_
_ACC
(V_TransId)
bccDeallocAcc
(V_TransId)
TO A_LocTransList
(V_TransId)
–
mieDealloc_
AccConf
(V_ConnRefNo)
mduDealloc_
AccConf
(V_ConnRefNo)
–
InService
mieDealloc_
RejConf
(V_ConnRefNo,
V_RejCause)
mduDealloc_
RejConf
(V_ConnRefNo,
V_RejCause)
–

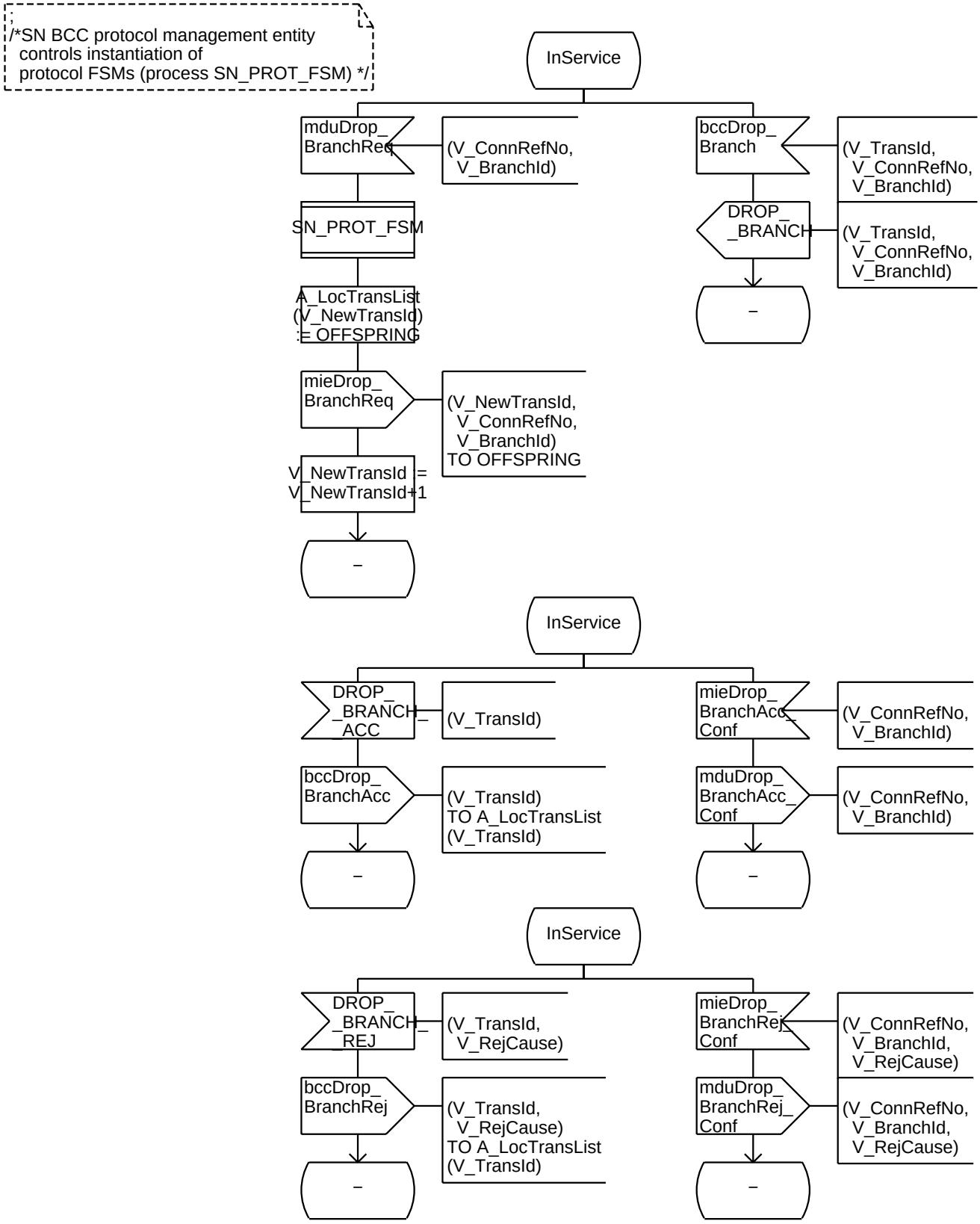

/*SN BCC protocol management entity
controls instantiation of
protocol FSMs (process SN_PROT_FSM) */
InService
mduModify_
Req
(V_ConnRefNo)
bccModify
(V_TransId,
V_ConnRefNo)
SN_PROT_FSM
MODIFY
(V_TransId,
V_ConnRefNo)
A_LocTransList
(V_NewTransId)
:= OFFSPRING
mieModifyReq
(V_NewTransId,
V_ConnRefNo)
TO OFFSPRING
V_NewTransId =
V_NewTransId+1
InService
MODIFY_
_ACC
(V_TransId)
mieModify_
AccConf
(V_ConnRefNo)
bccModifyAcc
(V_TransId)
TO A_LocTransList
(V_TransId)
mduModify_
AccConf
(V_ConnRefNo)
InService
MODIFY_
_REJ
(V_TransId,
V_RejCause)
mieModify_
RejConf
(V_ConnRefNo,
V_RejCause)
bccModifyRej
(V_TransId,
V_RejCause)
TO A_LocTransList
(V_TransId)
mduModify_
RejConf
(V_ConnRefNo,
V_RejCause)


/*SN BCC protocol management entity
controls instantiation of
protocol FSMs (process SN_PROT_FSM) */
InService
mduModify_
CompReq
(V_ConnRefNo)
bccModify_
Comp
(V_TransId,
V_ConnRefNo)
SN_PROT_FSM
MODIFY_
_COMP
(V_TransId,
V_ConnRefNo)
A_LocTransList
(V_NewTransId)
:= OFFSPRING
mieModify_
CompReq
(V_NewTransId,
V_ConnRefNo)
TO OFFSPRING
V_NewTransId =
V_NewTransId+1
InService
MODIFY_
COMP
_ACC
(V_TransId)
mieModify_
CompAcc_
Conf
(V_ConnRefNo)
bccModify_
CompAcc
(V_TransId)
TO A_LocTransList
(V_TransId)
mduModify_
CompAcc_
Conf
(V_ConnRefNo)
InService
MODIFY_
COMP
_REJ
(V_TransId,
V_RejCause)
mieModify_
CompRej_
Conf
(V_ConnRefNo,
V_RejCause)
bccModify_
CompRej
(V_TransId,
V_RejCause)
TO A_LocTransList
(V_TransId)
mduModify_
CompRej_
Conf
(V_ConnRefNo,
V_RejCause)


/*SN BCC protocol management entity
controls instantiation of
protocol FSMs (process SN_PROT_FSM) */
InService
mduModify_
AbortReq
(V_ConnRefNo)
SN_PROT_FSM
A_LocTransList
(V_NewTransId)
:= OFFSPRING
mieModify_
AbortReq
(V_NewTransId,
V_ConnRefNo)
TO OFFSPRING
V_NewTransId =
V_NewTransId+1
–
bccModify_
Abort
(V_TransId,
V_ConnRefNo)
MODIFY_
_ABORT
(V_TransId,
V_ConnRefNo)
–
InService
MODIFY_
ABORT
_ACC
(V_TransId)
bccModify_
AbortAcc
(V_TransId)
TO A_LocTransList
(V_TransId)
–
mieModify_
AbortAcc_
Conf
(V_ConnRefNo)
mduModify_
AbortAcc_
Conf
(V_ConnRefNo)
–
InService
MODIFY_
ABORT
_REJ
(V_TransId,
V_RejCause)
bccModify_
AbortRej
(V_TransId,
V_RejCause)
TO A_LocTransList
(V_TransId)
–
mieModify_
AbortRej_
Conf
(V_ConnRefNo,
V_RejCause)
mduModify_
AbortRej_
Conf
(V_ConnRefNo,
V_RejCause)
–


;
/*SN BCC protocol management entity
controls instantiation of
protocol FSMs (process SN_PROT_FSM) */
InService
mduAdd_
BranchReq
(V_ConnRefNo,
V_BranchId)
bccAdd_
Branch
(V_TransId,
V_ConnRefNo,
V_BranchId)
SN_PROT_FSM
ADD_
_BRANCH
(V_TransId,
V_ConnRefNo,
V_BranchId)
A_LocTransList
(V_NewTransId)
:= OFFSPRING
–
mieAddBranch_
Req
(V_NewTransId,
V_ConnRefNo,
V_BranchId)
TO OFFSPRING
V_NewTransId =
V_NewTransId+1
–
InService
ADD_
BRANCH
_ACC
(V_TransId)
mieAdd_
BranchAcc_
Conf
(V_ConnRefNo,
V_BranchId)
bccAdd_
BranchAcc
(V_TransId)
TO A_LocTransList
(V_TransId)
mduAdd_
BranchAcc_
Conf
(V_ConnRefNo,
V_BranchId)
–
–
InService
ADD_
BRANCH
_REJ
(V_TransId,
V_RejCause)
mieAdd_
BranchRej_
Conf
(V_ConnRefNo,
V_BranchId,
V_RejCause)
bccAdd_
BRANCHRej
(V_TransId,
V_RejCause)
TO A_LocTransList
(V_TransId)
mduAdd_
BranchRej_
Conf
(V_ConnRefNo,
V_BranchId,
V_RejCause)
–
–


/*SN BCC protocol management entity
controls instantiation of
protocol FSMs (process SN_PROT_FSM) */
InService
mduUpdate_
BranchReq
(V_ConnRefNo,
V_BranchId)
SN_PROT_FSM
A_LocTransList
(V_NewTransId)
:= OFFSPRING
mieUpdate_
BranchReq
(V_NewTransId,
V_ConnRefNo,
V_BranchId)
TO OFFSPRING
V_NewTransId :=
V_NewTransId+1
–
bccUpdate_
Branch
(V_TransId,
V_ConnRefNo,
V_BranchId)
UPDATE_
_BRANCH
(V_TransId,
V_ConnRefNo,
V_BranchId)
–
InService
UPDATE_
BRANCH
_ACC
(V_TransId)
bccUpdate_
BranchAcc
(V_TransId)
TO A_LocTransList
(V_TransId)
–
mieUpdate_
Branch_
AccConf
(V_ConnRefNo,
V_BranchId)
mduUpdate_
Branch_
AccConf
(V_ConnRefNo,
V_BranchId)
–
InService
UPDATE_
BRANCH
_REJ
(V_TransId,
V_RejCause)
bccUpdate_
BranchRej
(V_TransId,
V_RejCause)
TO A_LocTransList
(V_TransId)
–
mieUpdate_
Branch_
RejConf
(V_ConnRefNo,
V_BranchId,
V_RejCause)
mduUpdate_
Branch_
RejConf
(V_ConnRefNo,
V_BranchId,
V_RejCause)
–


/*SN BCC protocol management entity
controls instantiation of
protocol FSMs (process SN_PROT_FSM) */
InService
mduDrop_
BranchReq
(V_ConnRefNo,
V_BranchId)
SN_PROT_FSM
A_LocTransList
(V_NewTransId)
:= OFFSPRING
mieDrop_
BranchReq
(V_NewTransId,
V_ConnRefNo,
V_BranchId)
TO OFFSPRING
V_NewTransId :=
V_NewTransId+1
–
bccDrop_
Branch
(V_TransId,
V_ConnRefNo,
V_BranchId)
DROP_
_BRANCH
(V_TransId,
V_ConnRefNo,
V_BranchId)
–
InService
DROP_
BRANCH
_ACC
(V_TransId)
bccDrop_
BranchAcc
(V_TransId)
TO A_LocTransList
(V_TransId)
–
mieDrop_
BranchAcc_
Conf
(V_ConnRefNo,
V_BranchId)
mduDrop_
BranchAcc_
Conf
(V_ConnRefNo,
V_BranchId)
–
InService
DROP_
BRANCH
_REJ
(V_TransId,
V_RejCause)
bccDrop_
BranchRej
(V_TransId,
V_RejCause)
TO A_LocTransList
(V_TransId)
–
mieDrop_
BranchRej_
Conf
(V_ConnRefNo,
V_BranchId,
V_RejCause)
mduDrop_
BranchRej_
Conf
(V_ConnRefNo,
V_BranchId,
V_RejCause)
–

```
;
/*SN_PROT_FSM is responsible for sending/
receiving BCC messages*/
```

```
/* Definintions and declarations for
SN_PROT_FSM */
```

```
/* State descriptions

InService: Init State of SN_PROT_FSM, process is waiting for initial SN_SYSMGT primitives or AN messages
AwaitLocAck: SN_PROT_FSM has passed an AN request to SN_SYSMGT and is now awaiting the SN_SYSMGT response
AwaitRemAck: SN_PROT_FSM has sent a VB5 message to the AN and is now awaiting the AN response
*/
```

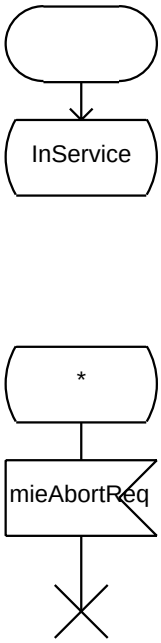
```
/* Timerdefinitions */
TIMER
T_Alloc          := 15,   /* Default value for ALLOC supervision timer is 15 sec, Tolerance +/− 10%  */
T_AllocComp      := 15,   /* Default value for ALLOC_COMP supervision timer is 15 sec, Tolerance +/− 10%  */
T_BbccReset      := 30,   /* Default value for BBCC_RESET supervision timer is 30 sec, Tolerance +/− 10%  */
T_BbccPresync    := 15,   /* Default value for PRESYNC supervision timer is 15 sec, Tolerance +/− 10%  */
T_Dealloc        := 30,   /* Default value for DEALLOC supervision timer is 30 sec, Tolerance +/− 10%  */
T_Modify         := 15,   /* Default value for MODIFY supervision timer is 15 sec, Tolerance +/− 10%  */
T_ModifyComp     := 15,   /* Default value for MODIFY_COMP supervision timer is 15 sec, Tolerance +/− 10%  */
T_ModifyAbort    := 15,   /* Default value for MODIFY_ABORT supervision timer is 15 sec, Tolerance +/− 10%  */
T_AddBranch      := 15,   /* Default value for ADD_BRANCH supervision timer is 15 sec, Tolerance +/− 10%  */
T_UpdateBranch   := 15,   /* Default value for UPDATE_BRANCH supervision timer is 15 sec, Tolerance +/− 10%  */
T_DropBranch     := 30;   /* Default value for DROP_BRANCH supervision timer is 30 sec, Tolerance +/− 10%  */
```

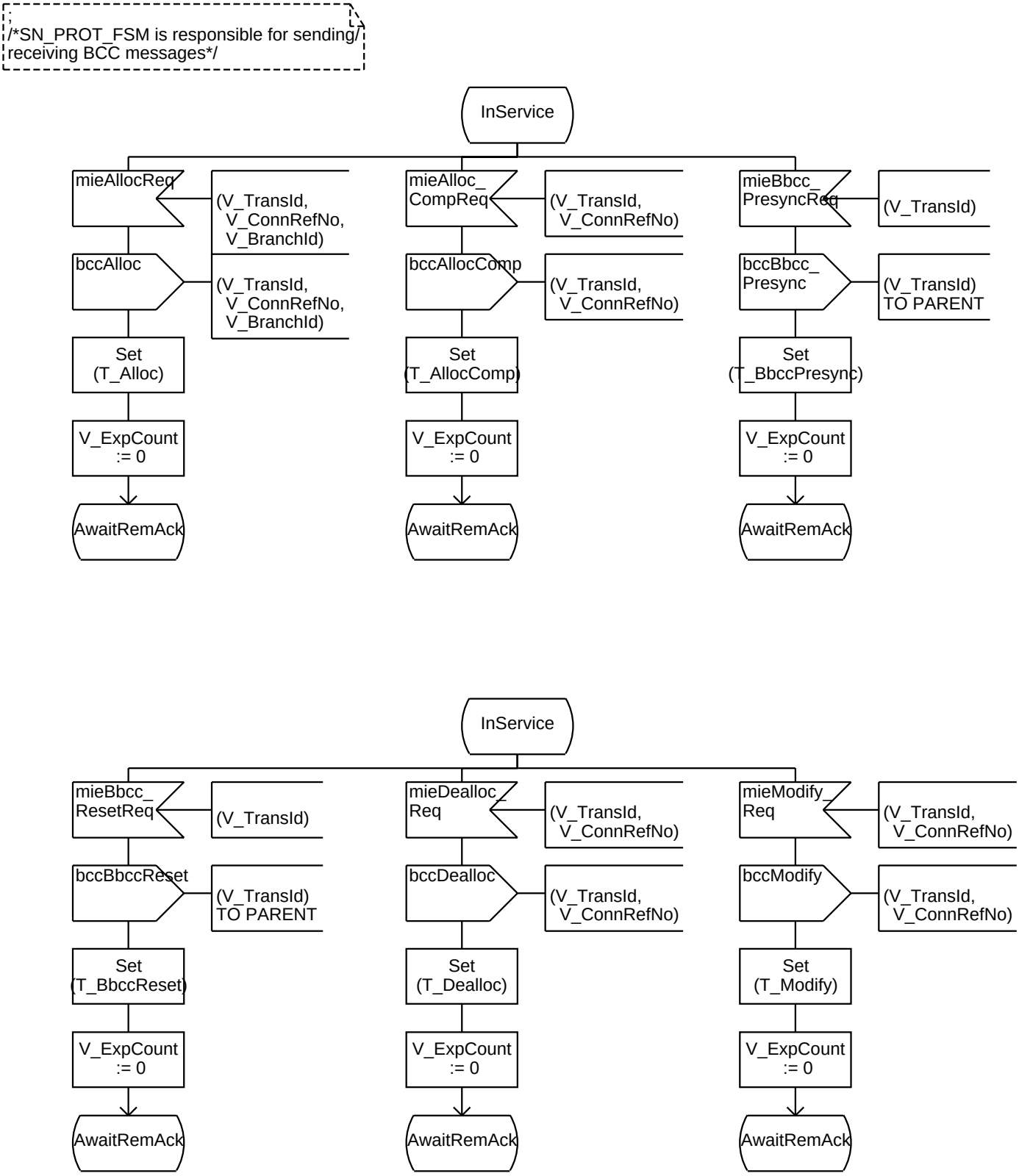
```
/* SN_PROT_FSM internal variables and constants*/
/**/
/* timer handling */
DCL V_ExpCount NATURAL;  /*number of actual message repetitions*/
/*possible values*/
SYNONYM C_ExpMax INTEGER = 1;        /*maximum number of message repetitions is 1*/
/**/
/*syntax check handling*/
DCL V_SynResult IT_SynResult;  /*syntax check result, provided by procedure IE_CHECK*/
```

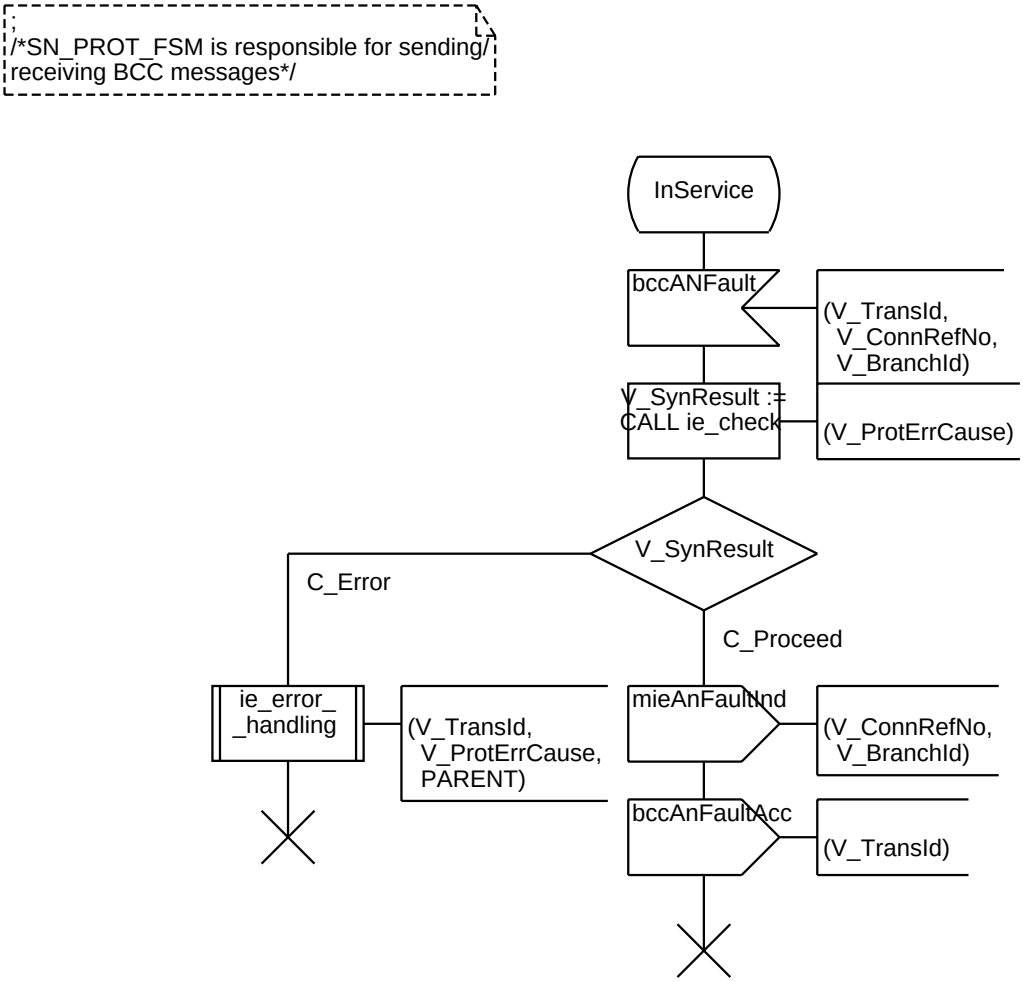
```
/*Signal data declarations*/
/**/
/* Transaction identifier towards AN */
DCL
V_ConnRefNo IT_ConnRefNo;
/**/
/* Transaction identifier */
DCL
V_TransId IT_TransId;
/**/
/* Branch identifier */
DCL
V_BranchId  IT_BranchId;
/**/
/*Error Cause in case of reject*/
DCL
V_ProtErrCause   IT_ProtErrCause;
/**/
/*  Reject cause */
DCL
V_RejCause  IT_RejCause;
```

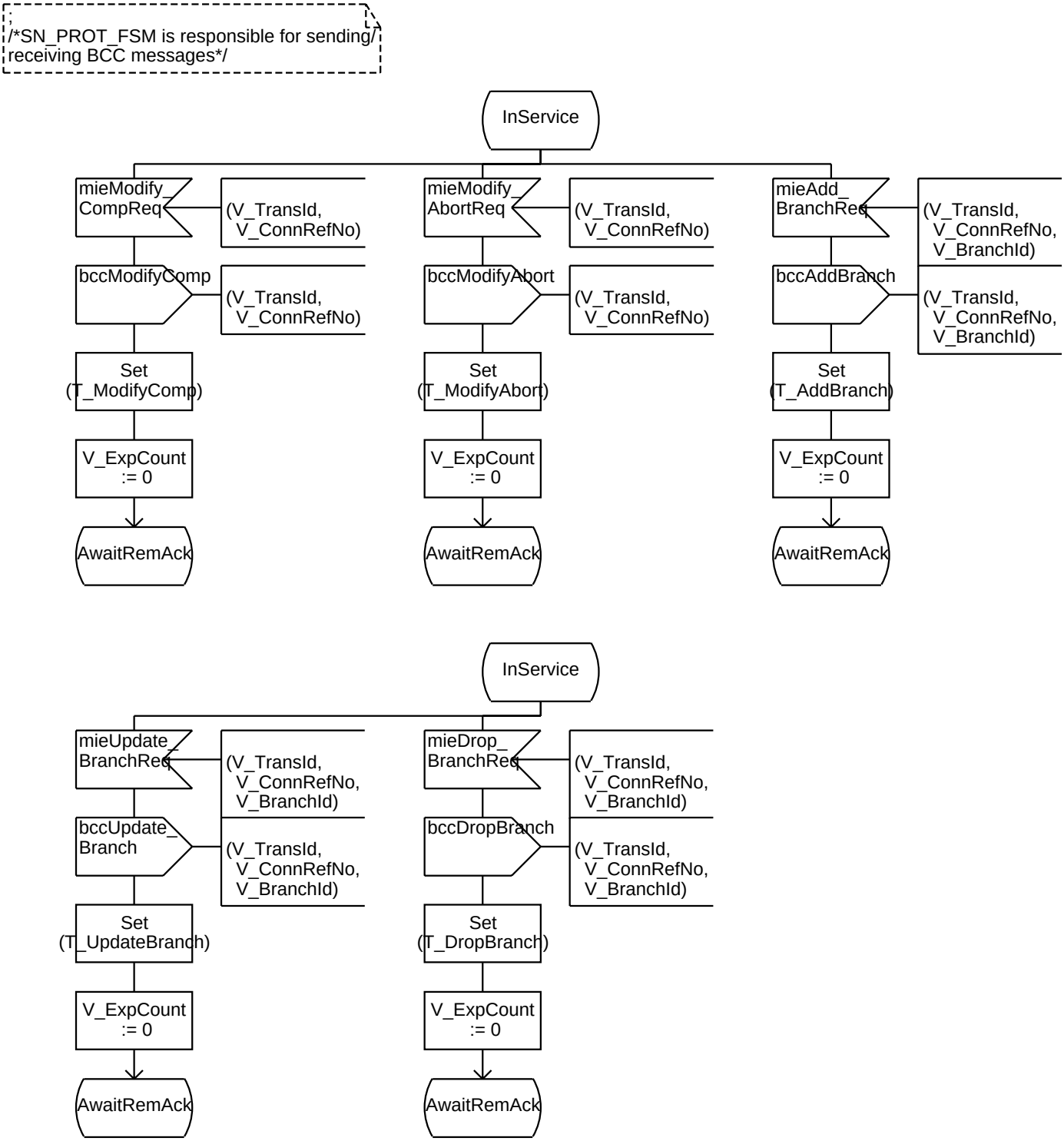
;
/*SN_PROT_FSM is responsible for sending/
receiving BCC messages*/

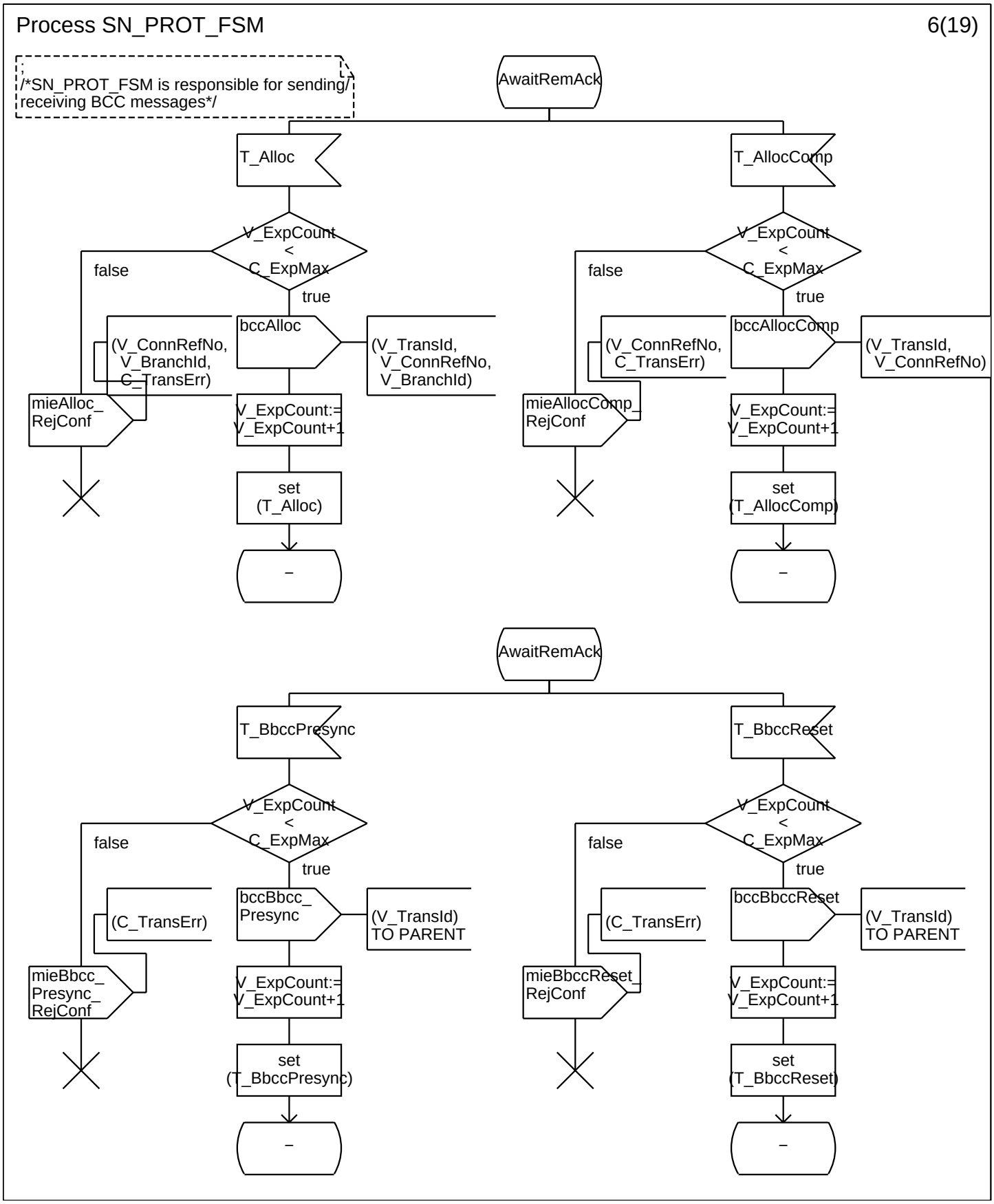
/*Handling of
Requests from
SN System Management */

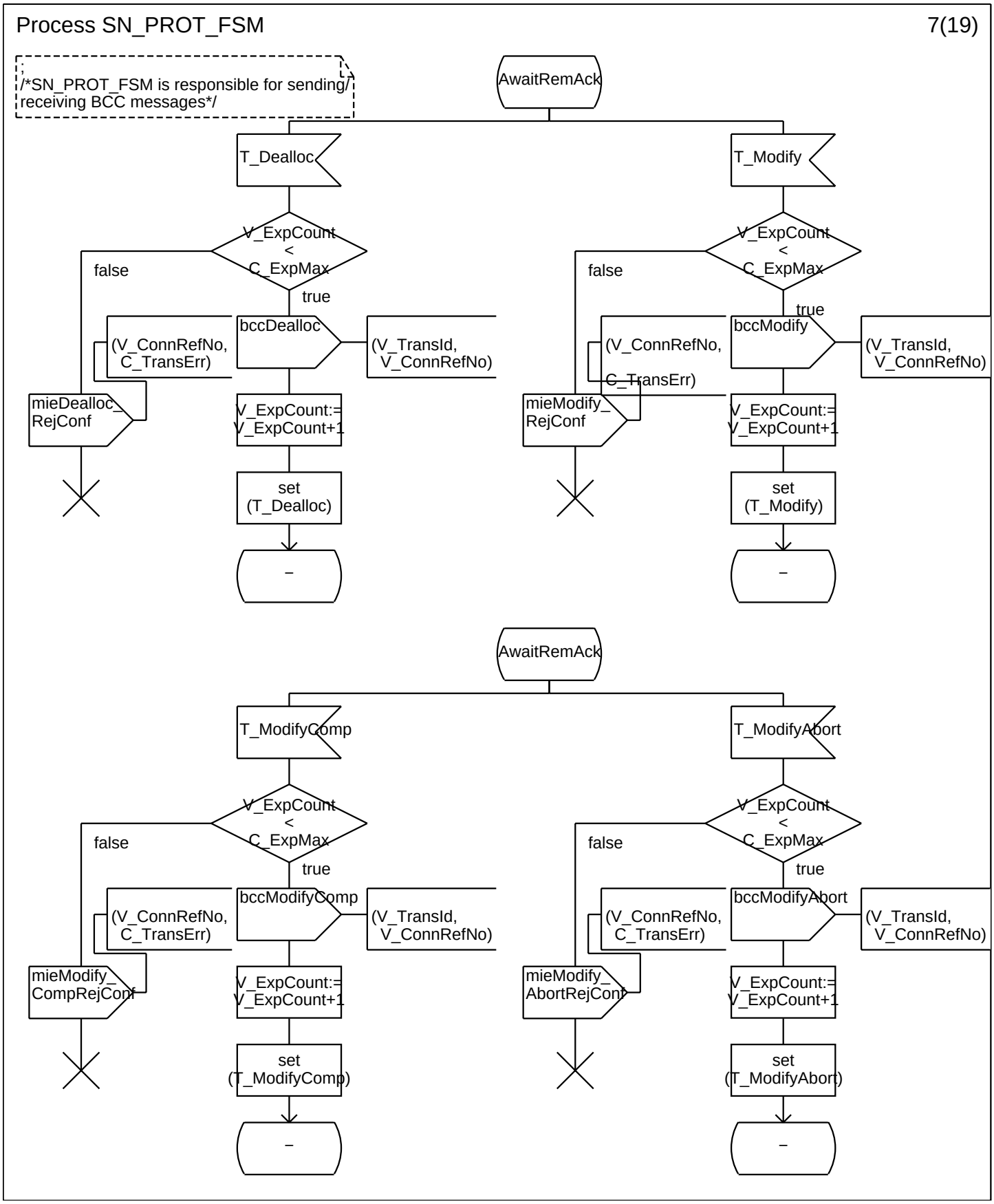


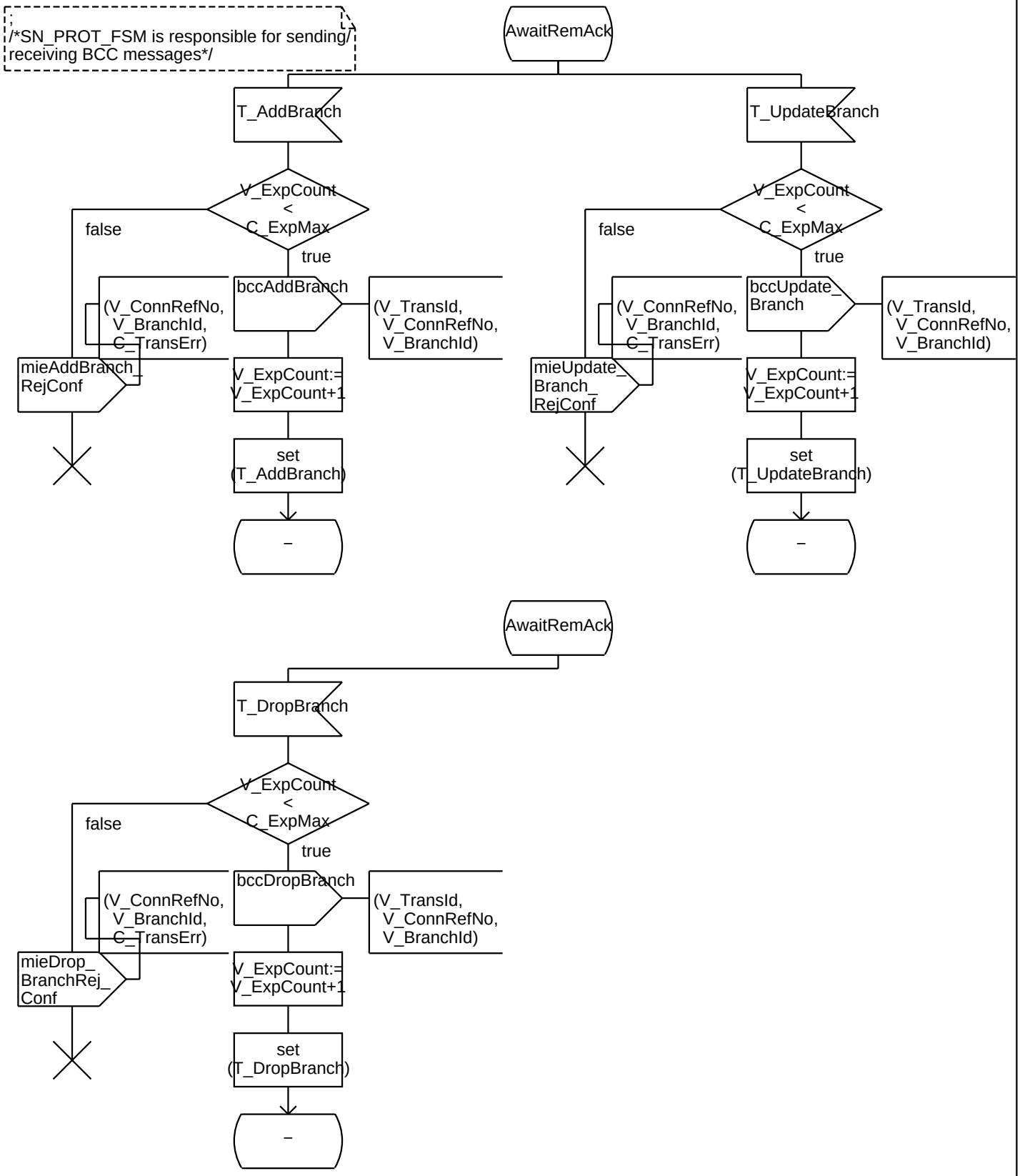

/*SN_PROT_FSM is responsible for sending/
receiving BCC messages*/
InService
mieAllocReq
(V_TransId,
V_ConnRefNo,
V_BranchId)
bccAlloc
(V_TransId,
V_ConnRefNo,
V_BranchId)
Set
(T_Alloc)
V_ExpCount
:= 0
AwaitRemAck
mieAlloc_
CompReq
(V_TransId,
V_ConnRefNo)
bccAllocComp
(V_TransId,
V_ConnRefNo)
Set
(T_AllocComp)
V_ExpCount
:= 0
AwaitRemAck
mieBbcc_
PresyncReq
(V_TransId)
bccBbcc_
Presync
(V_TransId)
TO PARENT
Set
(T_BbccPresync)
V_ExpCount
:= 0
AwaitRemAck
InService
mieBbcc_
ResetReq
(V_TransId)
bccBbccReset
(V_TransId)
TO PARENT
Set
(T_BbccReset)
V_ExpCount
:= 0
AwaitRemAck
mieDealloc_
Req
(V_TransId,
V_ConnRefNo)
bccDealloc
(V_TransId,
V_ConnRefNo)
Set
(T_Dealloc)
V_ExpCount
:= 0
AwaitRemAck
mieModify_
Req
(V_TransId,
V_ConnRefNo)
bccModify
(V_TransId,
V_ConnRefNo)
Set
(T_Modify)
V_ExpCount
:= 0
AwaitRemAck

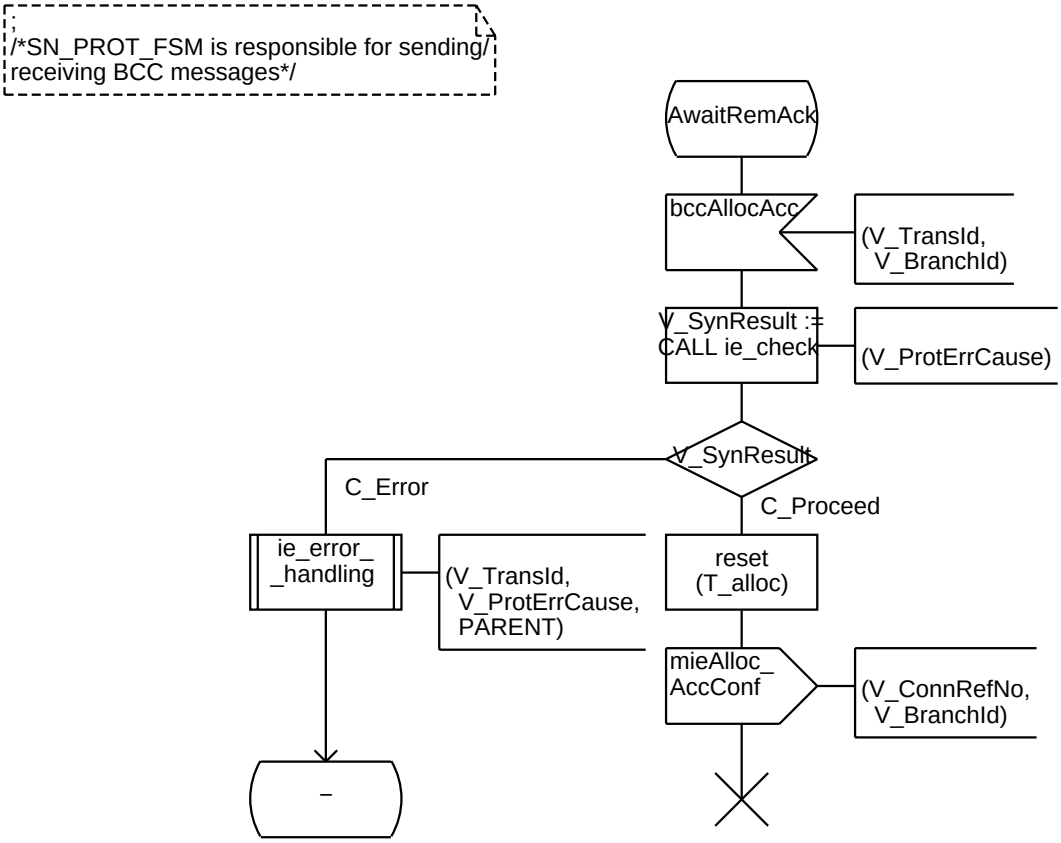

/*SN_PROT_FSM is responsible for sending/
receiving BCC messages*/
InService
bccANFault
(V_TransId,
V_ConnRefNo,
V_BranchId)
V_SynResult :=
CALL ie_check
(V_ProtErrCause)
V_SynResult
C_Error
C_Proceed
ie_error_
_handling
(V_TransId,
V_ProtErrCause,
PARENT)
mieAnFaultInd
(V_ConnRefNo,
V_BranchId)
bccAnFaultAcc
(V_TransId)

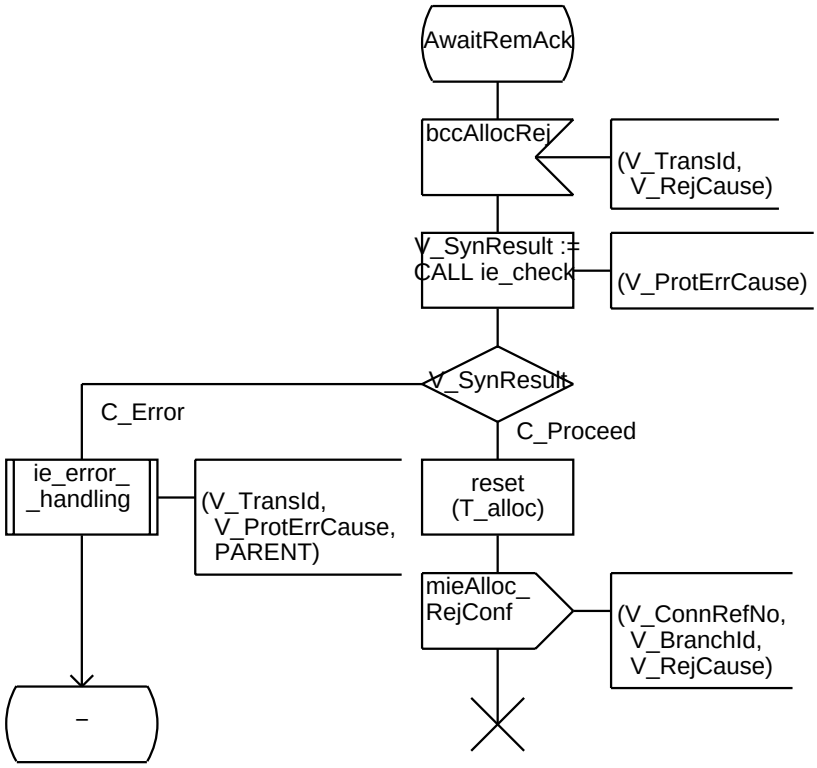

/*SN_PROT_FSM is responsible for sending/
receiving BCC messages*/
InService
mieModify_
CompReq
(V_TransId,
V_ConnRefNo)
bccModifyComp
(V_TransId,
V_ConnRefNo)
Set
(T_ModifyComp)
V_ExpCount
:= 0
AwaitRemAck
mieModify_
AbortReq
(V_TransId,
V_ConnRefNo)
bccModifyAbort
(V_TransId,
V_ConnRefNo)
Set
(T_ModifyAbort)
V_ExpCount
:= 0
AwaitRemAck
mieAdd_
BranchReq
(V_TransId,
V_ConnRefNo,
V_BranchId)
bccAddBranch
(V_TransId,
V_ConnRefNo,
V_BranchId)
Set
(T_AddBranch)
V_ExpCount
:= 0
AwaitRemAck
InService
mieUpdate_
BranchReq
(V_TransId,
V_ConnRefNo,
V_BranchId)
bccUpdate_
Branch
(V_TransId,
V_ConnRefNo,
V_BranchId)
Set
(T_UpdateBranch)
V_ExpCount
:= 0
AwaitRemAck
mieDrop_
BranchReq
(V_TransId,
V_ConnRefNo,
V_BranchId)
bccDropBranch
(V_TransId,
V_ConnRefNo,
V_BranchId)
Set
(T_DropBranch)
V_ExpCount
:= 0
AwaitRemAck

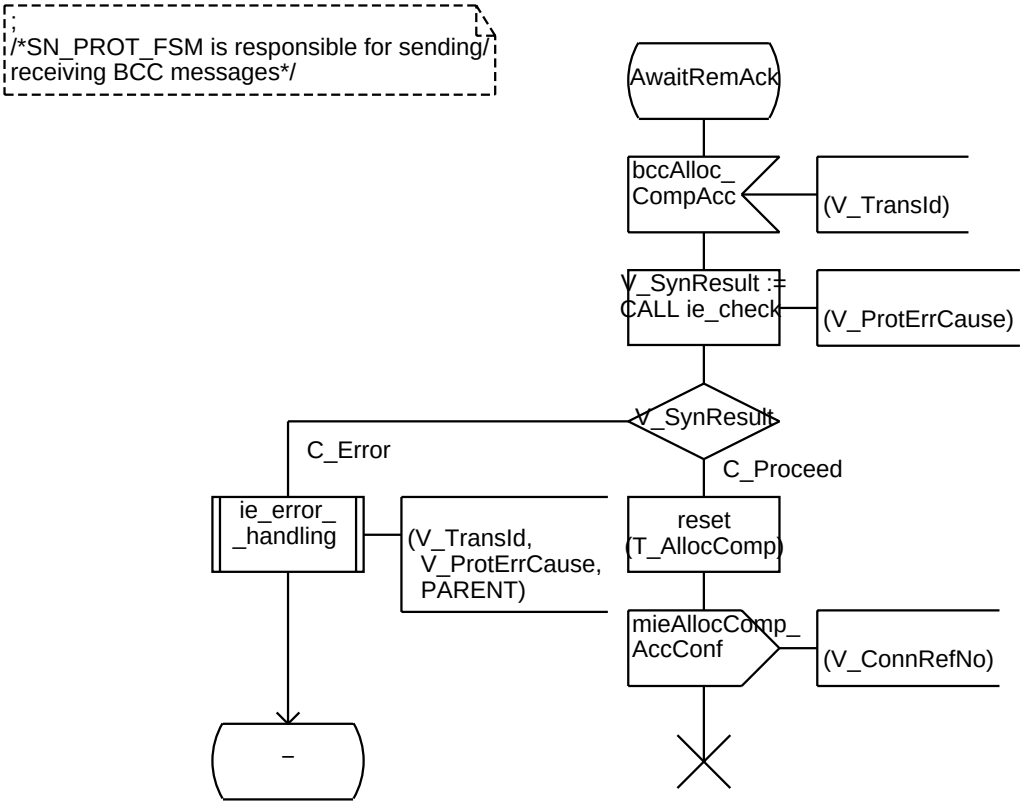

/*SN_PROT_FSM is responsible for sending/ receiving BCC messages*/
AwaitRemAck
T_Alloc
V_ExpCount < C_ExpMax
false
true
bccAlloc
(V_ConnRefNo, V_BranchId, C_TransErr)
(V_TransId, V_ConnRefNo, V_BranchId)
mieAlloc_ RejConf
V_ExpCount:= V_ExpCount+1
set (T_Alloc)
–
T_AllocComp
V_ExpCount < C_ExpMax
false
true
bccAllocComp
(V_ConnRefNo, C_TransErr)
(V_TransId, V_ConnRefNo)
mieAllocComp_ RejConf
V_ExpCount:= V_ExpCount+1
set (T_AllocComp)
–
AwaitRemAck
T_BbccPresync
V_ExpCount < C_ExpMax
false
true
bccBbcc_ Presync
(C_TransErr)
(V_TransId) TO PARENT
mieBbcc_ Presync_ RejConf
V_ExpCount:= V_ExpCount+1
set (T_BbccPresync)
–
T_BbccReset
V_ExpCount < C_ExpMax
false
true
bccBbccReset
(C_TransErr)
(V_TransId) TO PARENT
mieBbccReset_ RejConf
V_ExpCount:= V_ExpCount+1
set (T_BbccReset)
–

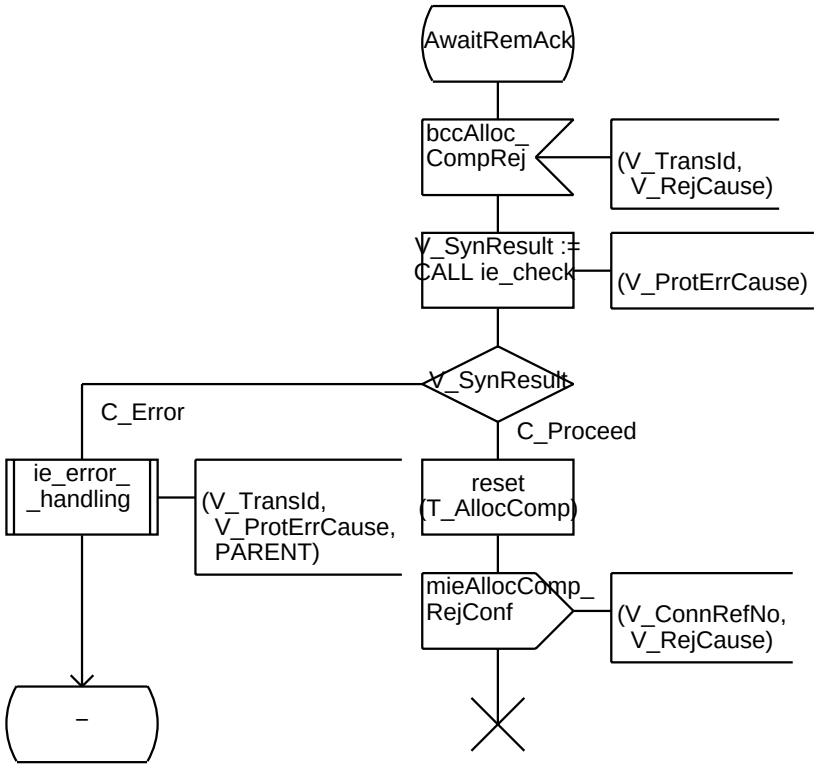

/*SN_PROT_FSM is responsible for sending/
receiving BCC messages*/
AwaitRemAck
T_Dealloc
V_ExpCount
<
C_ExpMax
false
true
bccDealloc
(V_ConnRefNo,
C_TransErr)
(V_TransId,
V_ConnRefNo)
mieDealloc_
RejConf
V_ExpCount:=
V_ExpCount+1
set
(T_Dealloc)
–
T_Modify
V_ExpCount
<
C_ExpMax
false
true
bccModify
(V_ConnRefNo,
C_TransErr)
(V_TransId,
V_ConnRefNo)
mieModify_
RejConf
V_ExpCount:=
V_ExpCount+1
set
(T_Modify)
–
AwaitRemAck
T_ModifyComp
V_ExpCount
<
C_ExpMax
false
true
bccModifyComp
(V_ConnRefNo,
C_TransErr)
(V_TransId,
V_ConnRefNo)
mieModify_
CompRejConf
V_ExpCount:=
V_ExpCount+1
set
(T_ModifyComp)
–
T_ModifyAbort
V_ExpCount
<
C_ExpMax
false
true
bccModifyAbort
(V_ConnRefNo,
C_TransErr)
(V_TransId,
V_ConnRefNo)
mieModify_
AbortRejConf
V_ExpCount:=
V_ExpCount+1
set
(T_ModifyAbort)
–

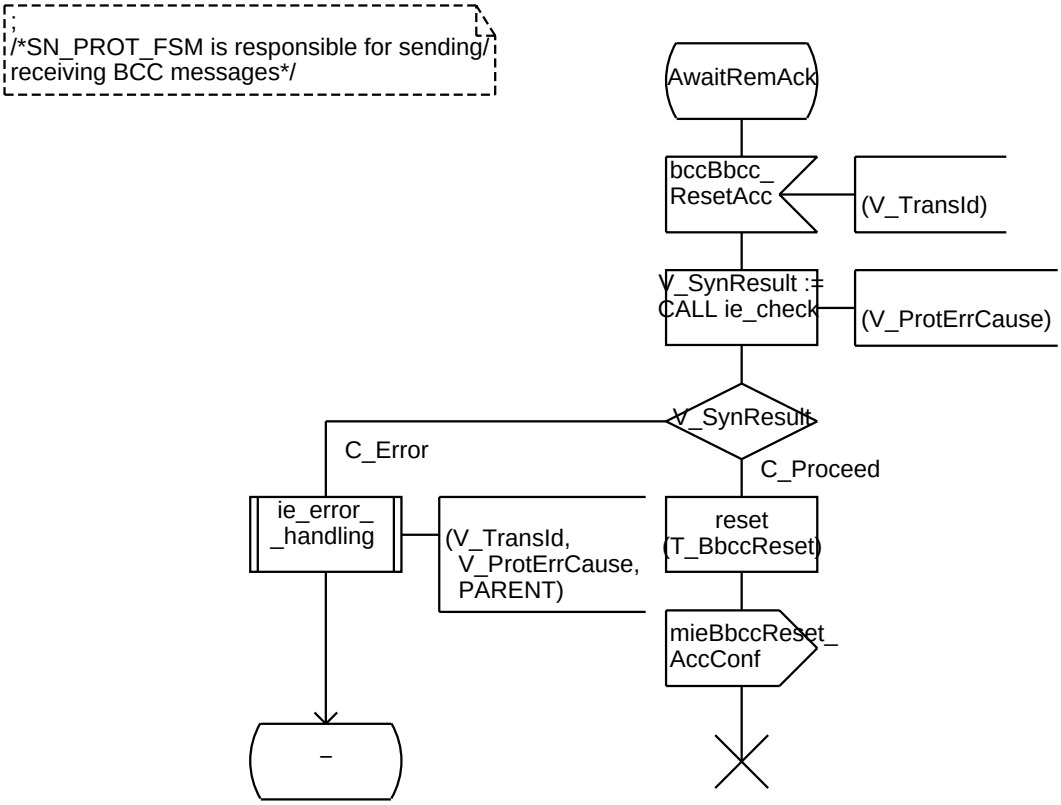

/*SN_PROT_FSM is responsible for sending/
receiving BCC messages*/
AwaitRemAck
T_AddBranch
V_ExpCount
<
C_ExpMax
false
true
bccAddBranch
(V_ConnRefNo,
V_BranchId,
C_TransErr)
(V_TransId,
V_ConnRefNo,
V_BranchId)
mieAddBranch_
RejConf
V_ExpCount:=
V_ExpCount+1
set
(T_AddBranch)
–
T_UpdateBranch
V_ExpCount
<
C_ExpMax
false
true
bccUpdate_
Branch
(V_ConnRefNo,
V_BranchId,
C_TransErr)
(V_TransId,
V_ConnRefNo,
V_BranchId)
mieUpdate_
Branch_
RejConf
V_ExpCount:=
V_ExpCount+1
set
(T_UpdateBranch)
–
AwaitRemAck
T_DropBranch
V_ExpCount
<
C_ExpMax
false
true
bccDropBranch
(V_ConnRefNo,
V_BranchId,
C_TransErr)
(V_TransId,
V_ConnRefNo,
V_BranchId)
mieDrop_
BranchRej_
Conf
V_ExpCount:=
V_ExpCount+1
set
(T_DropBranch)
–

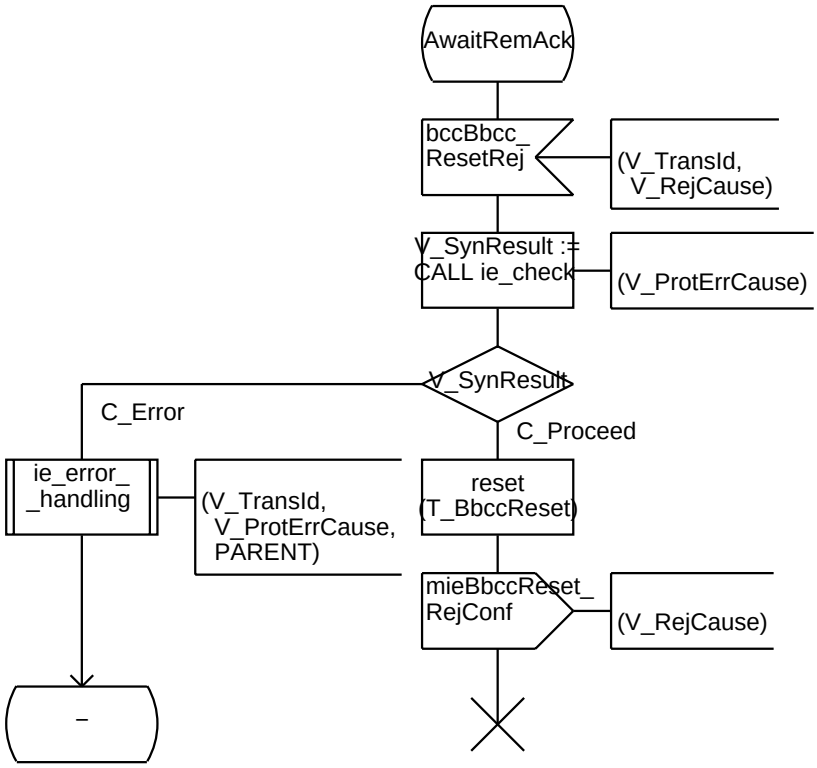

/*SN_PROT_FSM is responsible for sending/
receiving BCC messages*/
AwaitRemAck
bccAllocAcc
(V_TransId,
V_BranchId)
V_SynResult :=
CALL ie_check
(V_ProtErrCause)
V_SynResult
C_Error
C_Proceed
ie_error_
_handling
(V_TransId,
V_ProtErrCause,
PARENT)
reset
(T_alloc)
mieAlloc_
AccConf
(V_ConnRefNo,
V_BranchId)
–

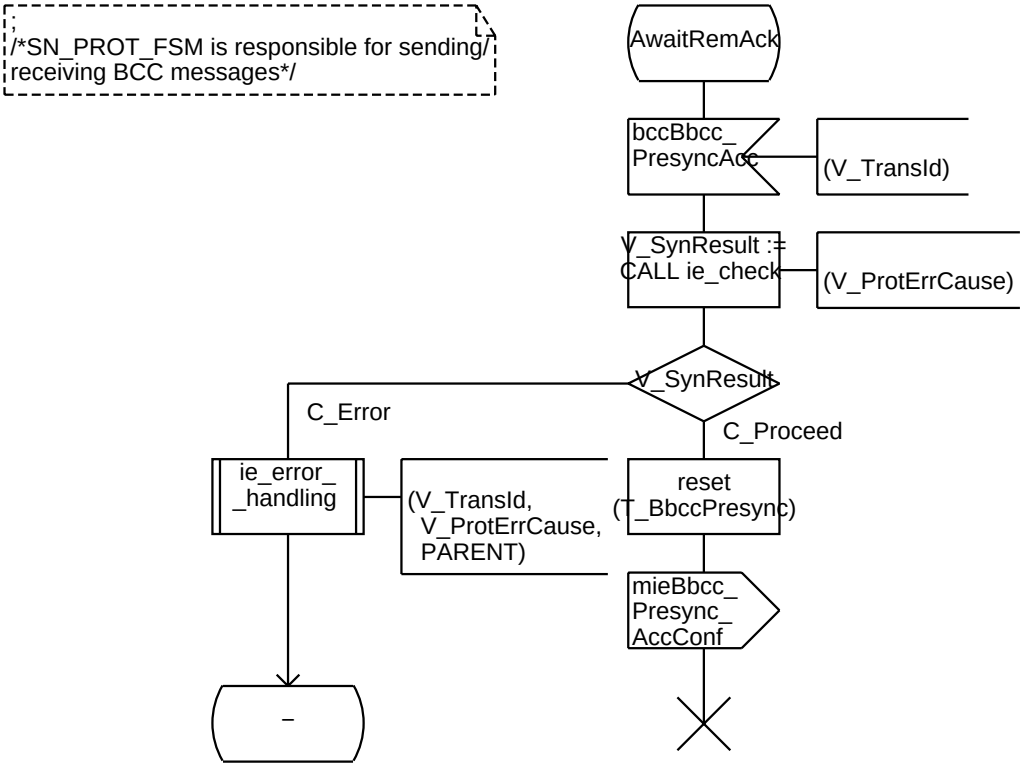

AwaitRemAck
bccAllocRej
(V_TransId,
V_RejCause)
V_SynResult :=
CALL ie_check
(V_ProtErrCause)
V_SynResult
C_Error
C_Proceed
ie_error_
_handling
(V_TransId,
V_ProtErrCause,
PARENT)
reset
(T_alloc)
mieAlloc_
RejConf
(V_ConnRefNo,
V_BranchId,
V_RejCause)
–

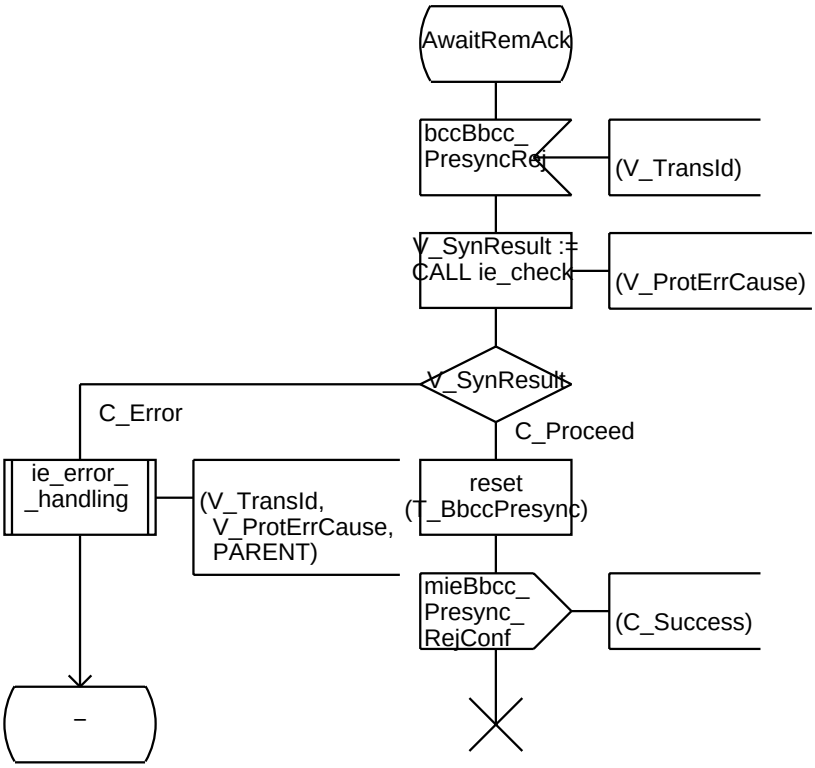

/*SN_PROT_FSM is responsible for sending/
receiving BCC messages*/
AwaitRemAck
bccAlloc_
CompAcc
(V_TransId)
V_SynResult :=
CALL ie_check
(V_ProtErrCause)
V_SynResult
C_Error
C_Proceed
ie_error_
_handling
(V_TransId,
V_ProtErrCause,
PARENT)
reset
(T_AllocComp)
mieAllocComp_
AccConf
(V_ConnRefNo)
–

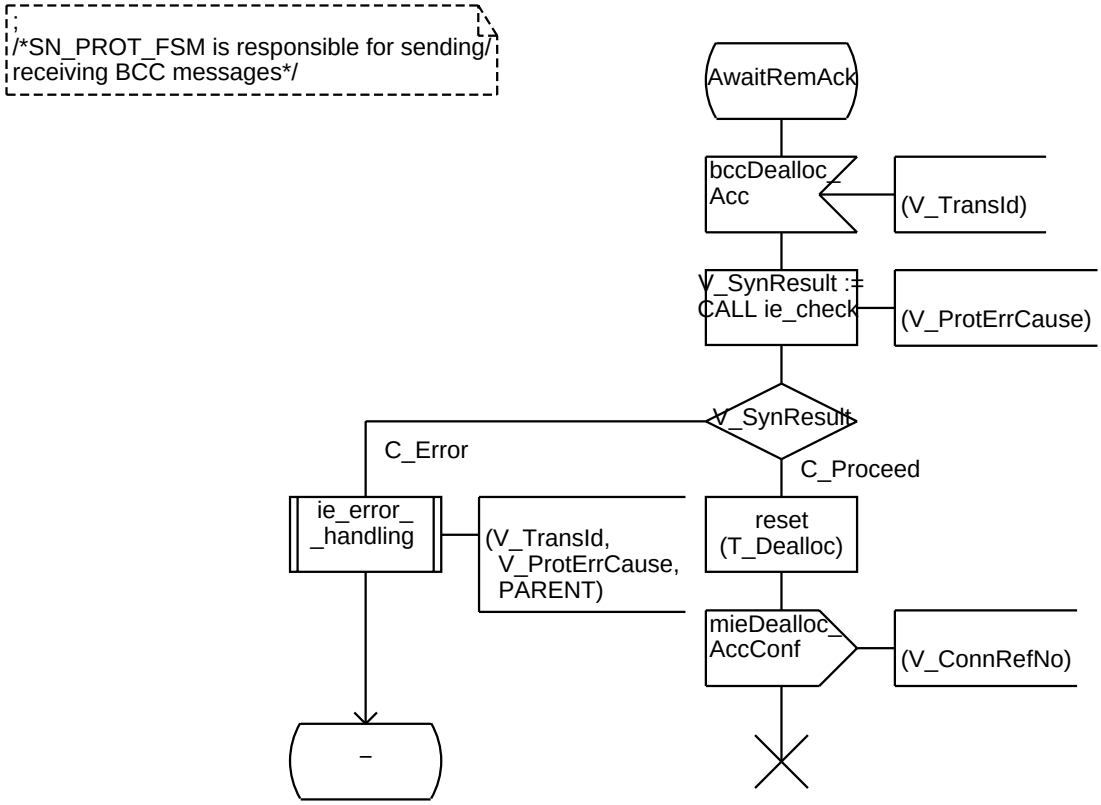

AwaitRemAck
bccAlloc_
CompRej
(V_TransId,
V_RejCause)
V_SynResult :=
CALL ie_check
(V_ProtErrCause)
V_SynResult
C_Error
C_Proceed
ie_error_
_handling
(V_TransId,
V_ProtErrCause,
PARENT)
reset
(T_AllocComp)
mieAllocComp_
RejConf
(V_ConnRefNo,
V_RejCause)
–

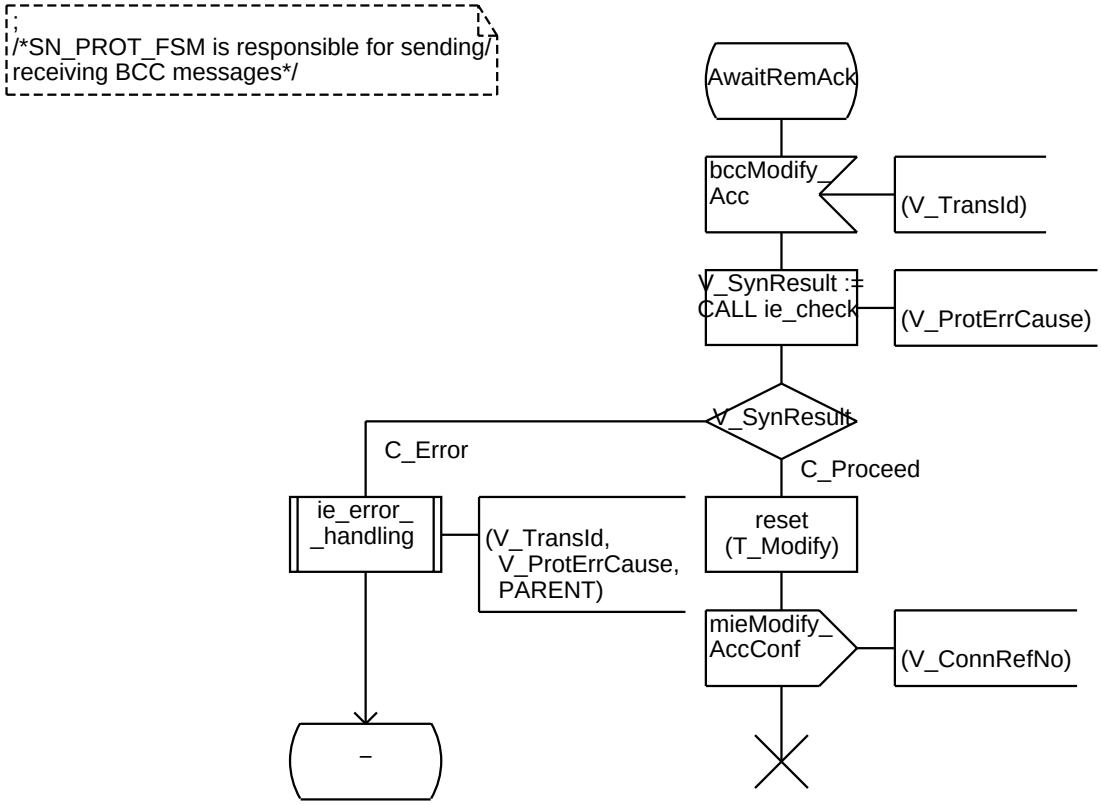

/*SN_PROT_FSM is responsible for sending/
receiving BCC messages*/
AwaitRemAck
bccBbcc_
ResetAcc
(V_TransId)
V_SynResult :=
CALL ie_check
(V_ProtErrCause)
V_SynResult
C_Error
C_Proceed
ie_error_
_handling
(V_TransId,
V_ProtErrCause,
PARENT)
reset
(T_BbccReset)
mieBbccReset_
AccConf
–

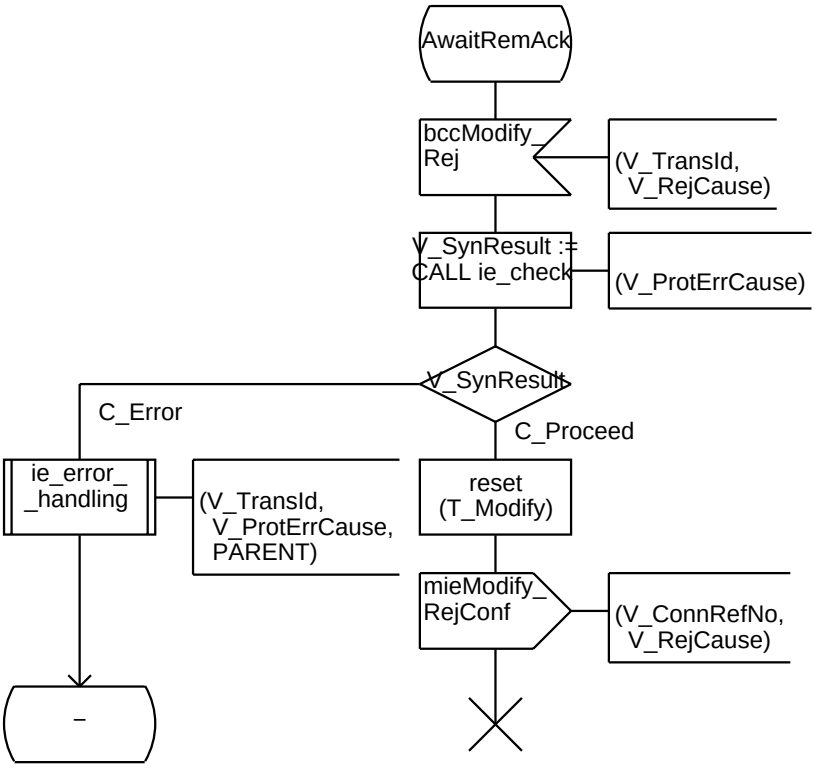

AwaitRemAck
bccBbcc_
ResetRej
(V_TransId,
V_RejCause)
V_SynResult :=
CALL ie_check
(V_ProtErrCause)
V_SynResult
C_Error
C_Proceed
ie_error_
_handling
(V_TransId,
V_ProtErrCause,
PARENT)
reset
(T_BbccReset)
mieBbccReset_
RejConf
(V_RejCause)
–

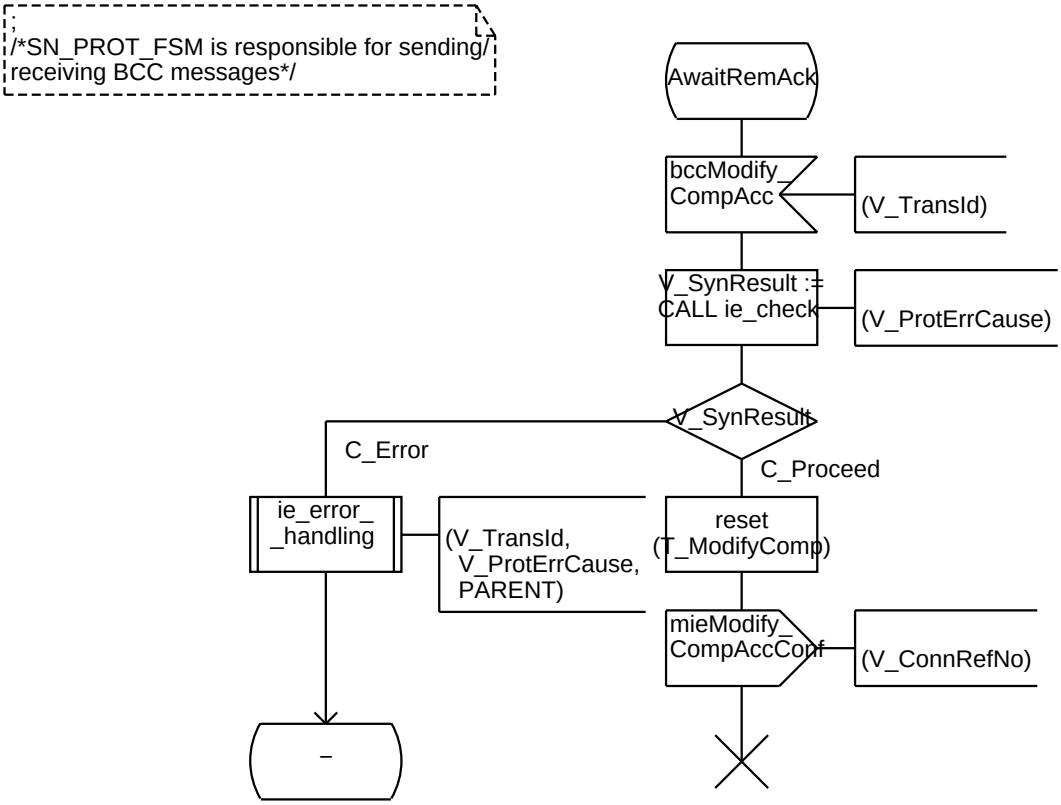

/*SN_PROT_FSM is responsible for sending/
receiving BCC messages*/
AwaitRemAck
bccBbcc_
PresyncAcc
(V_TransId)
V_SynResult :=
CALL ie_check
(V_ProtErrCause)
V_SynResult
C_Error
C_Proceed
ie_error_
_handling
(V_TransId,
V_ProtErrCause,
PARENT)
reset
(T_BbccPresync)
mieBbcc_
Presync_
AccConf
–

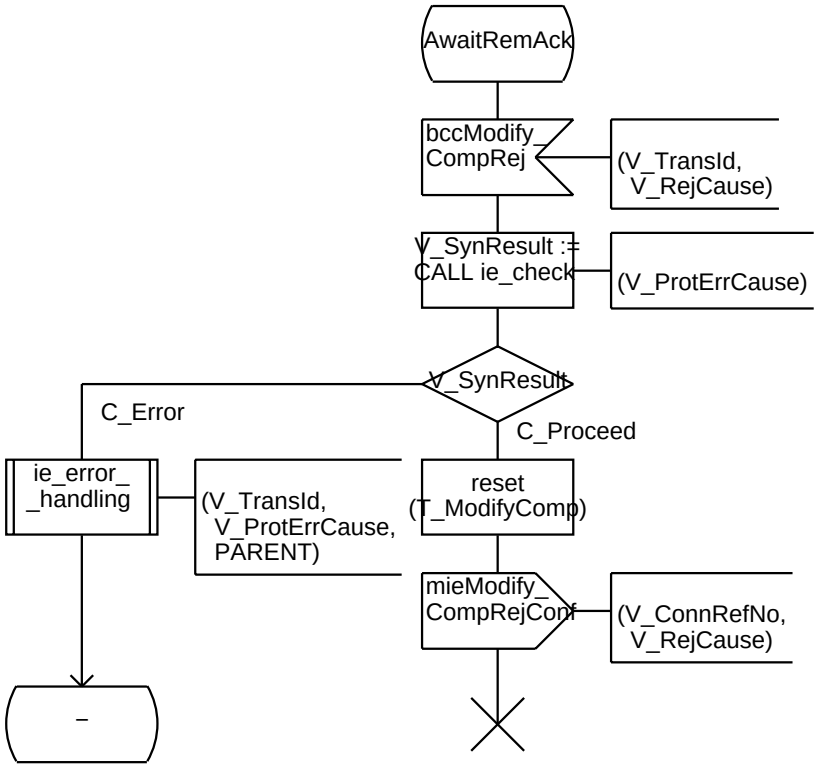

AwaitRemAck
bccBbcc_
PresyncRej
(V_TransId)
V_SynResult :=
CALL ie_check
(V_ProtErrCause)
V_SynResult
C_Error
C_Proceed
ie_error_
_handling
(V_TransId,
V_ProtErrCause,
PARENT)
reset
(T_BbccPresync)
mieBbcc_
Presync_
RejConf
(C_Success)
–

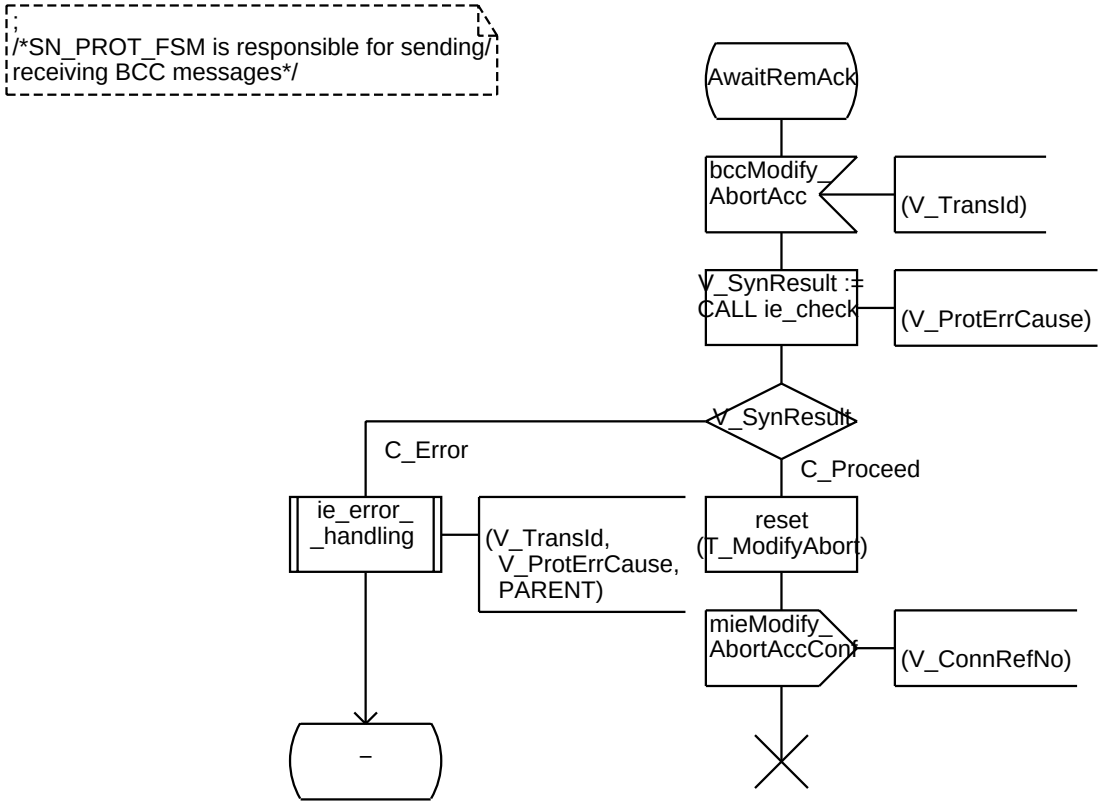

/*SN_PROT_FSM is responsible for sending/
receiving BCC messages*/
AwaitRemAck
bccDealloc_
Acc
(V_TransId)
V_SynResult :=
CALL ie_check
(V_ProtErrCause)
V_SynResult
C_Error
C_Proceed
ie_error_
_handling
(V_TransId,
V_ProtErrCause,
PARENT)
reset
(T_Dealloc)
mieDealloc_
AccConf
(V_ConnRefNo)
–

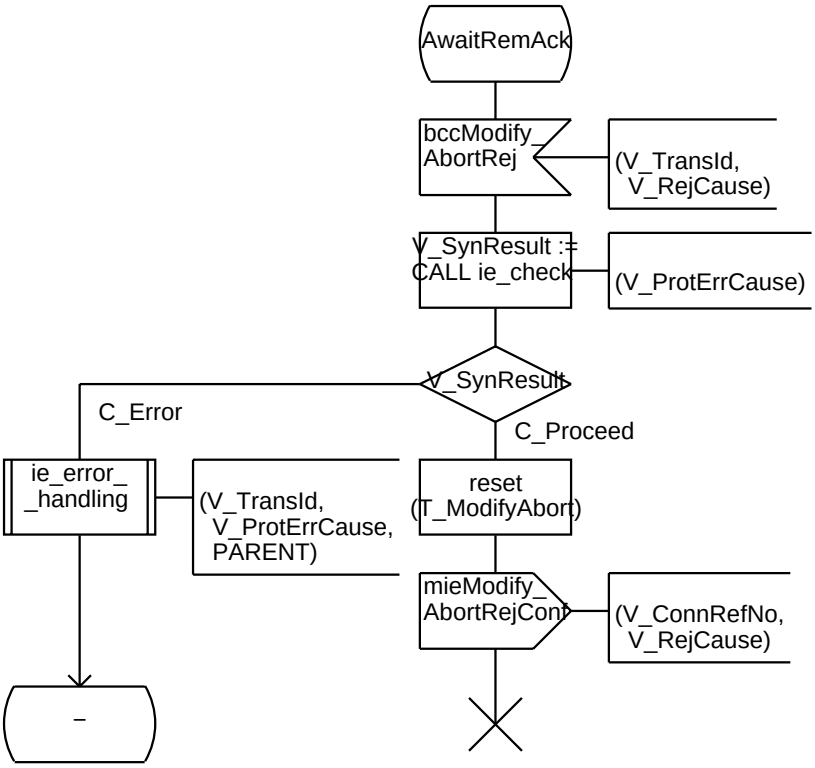

/*SN_PROT_FSM is responsible for sending/
receiving BCC messages*/
AwaitRemAck
bccModify_
Acc
(V_TransId)
V_SynResult :=
CALL ie_check
(V_ProtErrCause)
V_SynResult
C_Error
C_Proceed
ie_error_
_handling
(V_TransId,
V_ProtErrCause,
PARENT)
reset
(T_Modify)
mieModify_
AccConf
(V_ConnRefNo)
–

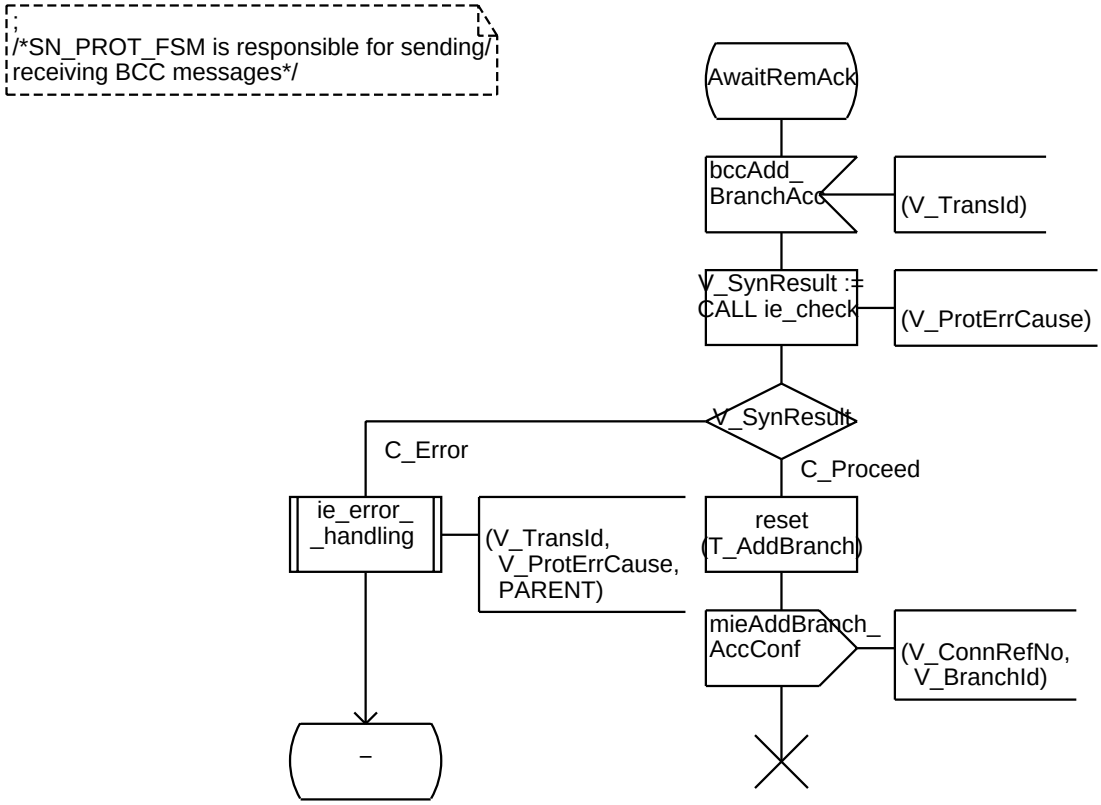

AwaitRemAck
bccModify_
Rej
(V_TransId,
V_RejCause)
V_SynResult :=
CALL ie_check
(V_ProtErrCause)
V_SynResult
C_Error
C_Proceed
ie_error_
_handling
(V_TransId,
V_ProtErrCause,
PARENT)
reset
(T_Modify)
mieModify_
RejConf
(V_ConnRefNo,
V_RejCause)
–

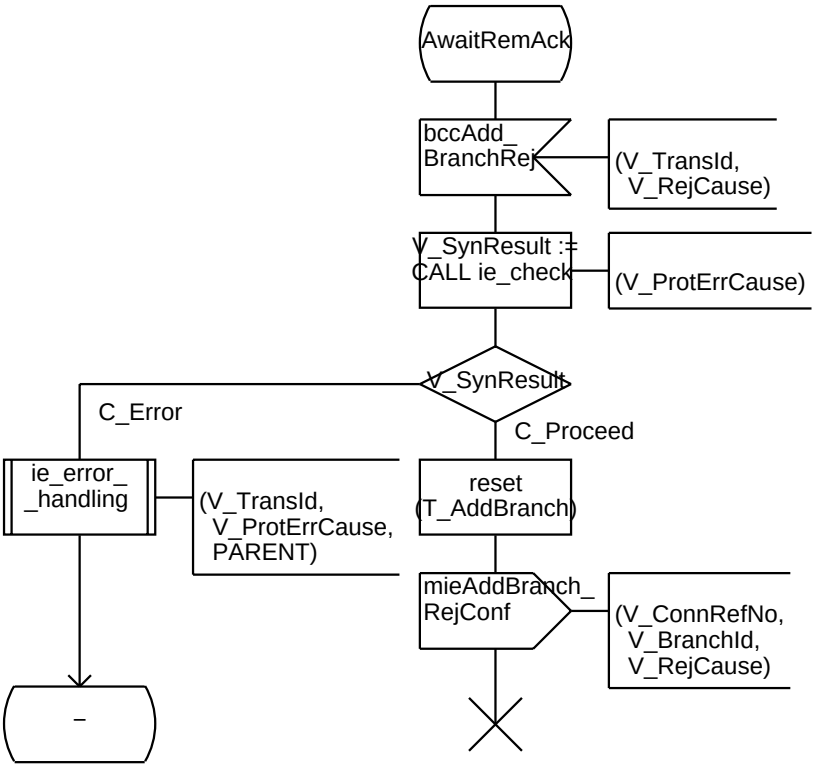

/*SN_PROT_FSM is responsible for sending/
receiving BCC messages*/
AwaitRemAck
bccModify_
CompAcc
(V_TransId)
V_SynResult :=
CALL ie_check
(V_ProtErrCause)
V_SynResult
C_Error
C_Proceed
ie_error_
_handling
(V_TransId,
V_ProtErrCause,
PARENT)
reset
(T_ModifyComp)
mieModify_
CompAccConf
(V_ConnRefNo)
–

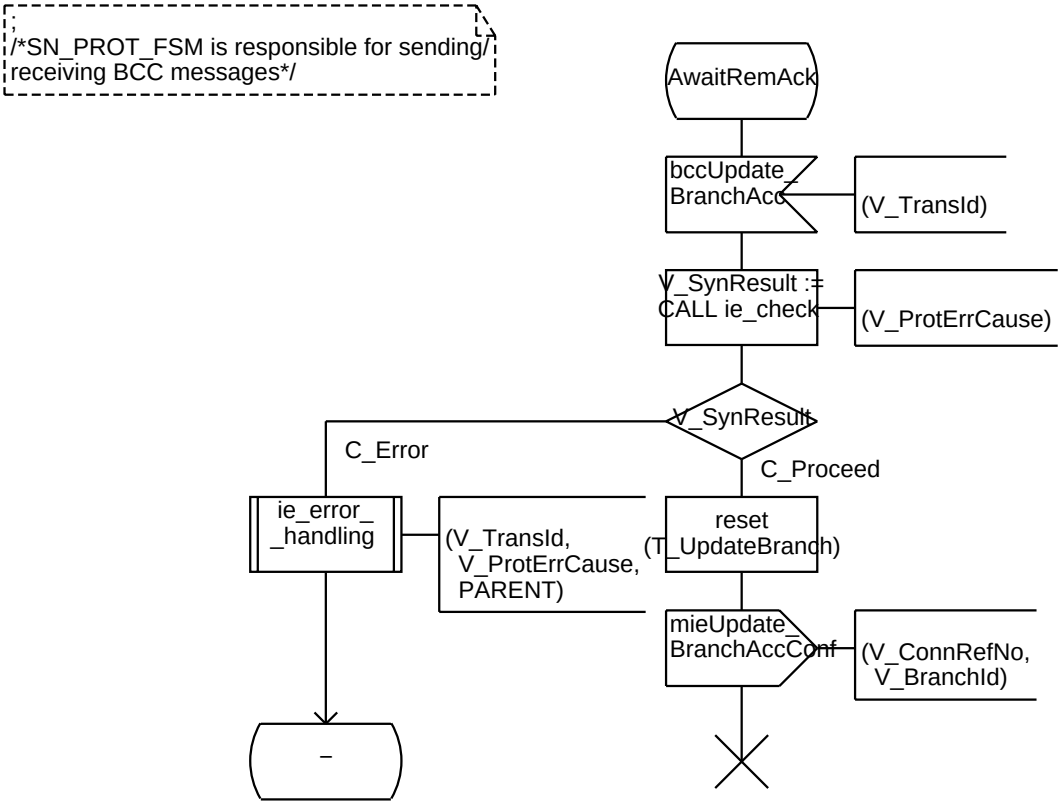

AwaitRemAck
bccModify_
CompRej
(V_TransId,
V_RejCause)
V_SynResult :=
CALL ie_check
(V_ProtErrCause)
V_SynResult
C_Error
C_Proceed
ie_error_
_handling
(V_TransId,
V_ProtErrCause,
PARENT)
reset
(T_ModifyComp)
mieModify_
CompRejConf
(V_ConnRefNo,
V_RejCause)
–

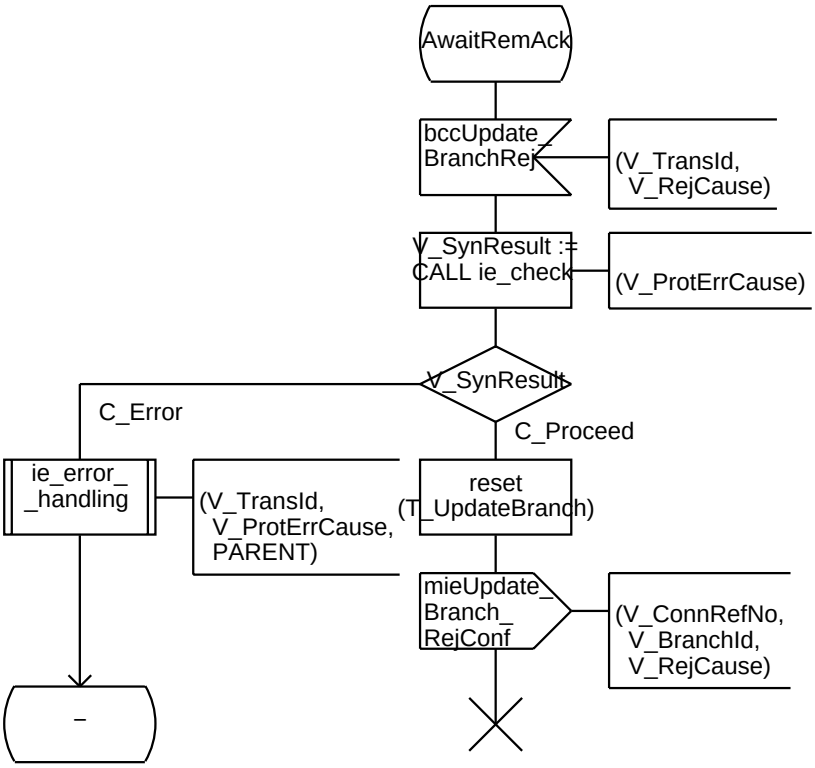

/*SN_PROT_FSM is responsible for sending/
receiving BCC messages*/
AwaitRemAck
bccModify_
AbortAcc
(V_TransId)
V_SynResult :=
CALL ie_check
(V_ProtErrCause)
V_SynResult
C_Error
C_Proceed
ie_error_
_handling
(V_TransId,
V_ProtErrCause,
PARENT)
reset
(T_ModifyAbort)
mieModify_
AbortAccConf
(V_ConnRefNo)
–

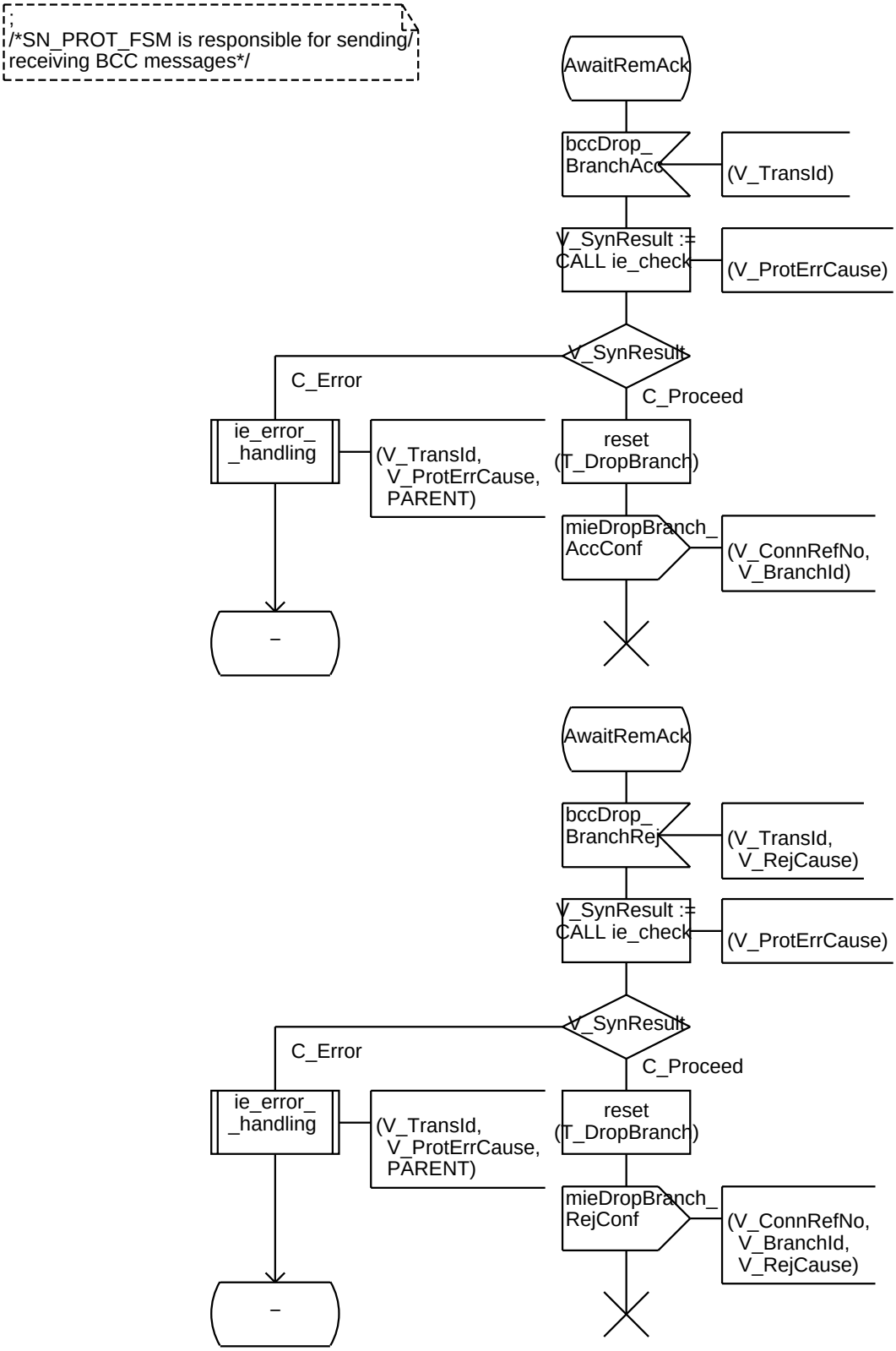

AwaitRemAck
bccModify_
AbortRej
(V_TransId,
V_RejCause)
V_SynResult :=
CALL ie_check
(V_ProtErrCause)
V_SynResult
C_Error
C_Proceed
ie_error_
_handling
(V_TransId,
V_ProtErrCause,
PARENT)
reset
(T_ModifyAbort)
mieModify_
AbortRejConf
(V_ConnRefNo,
V_RejCause)
–

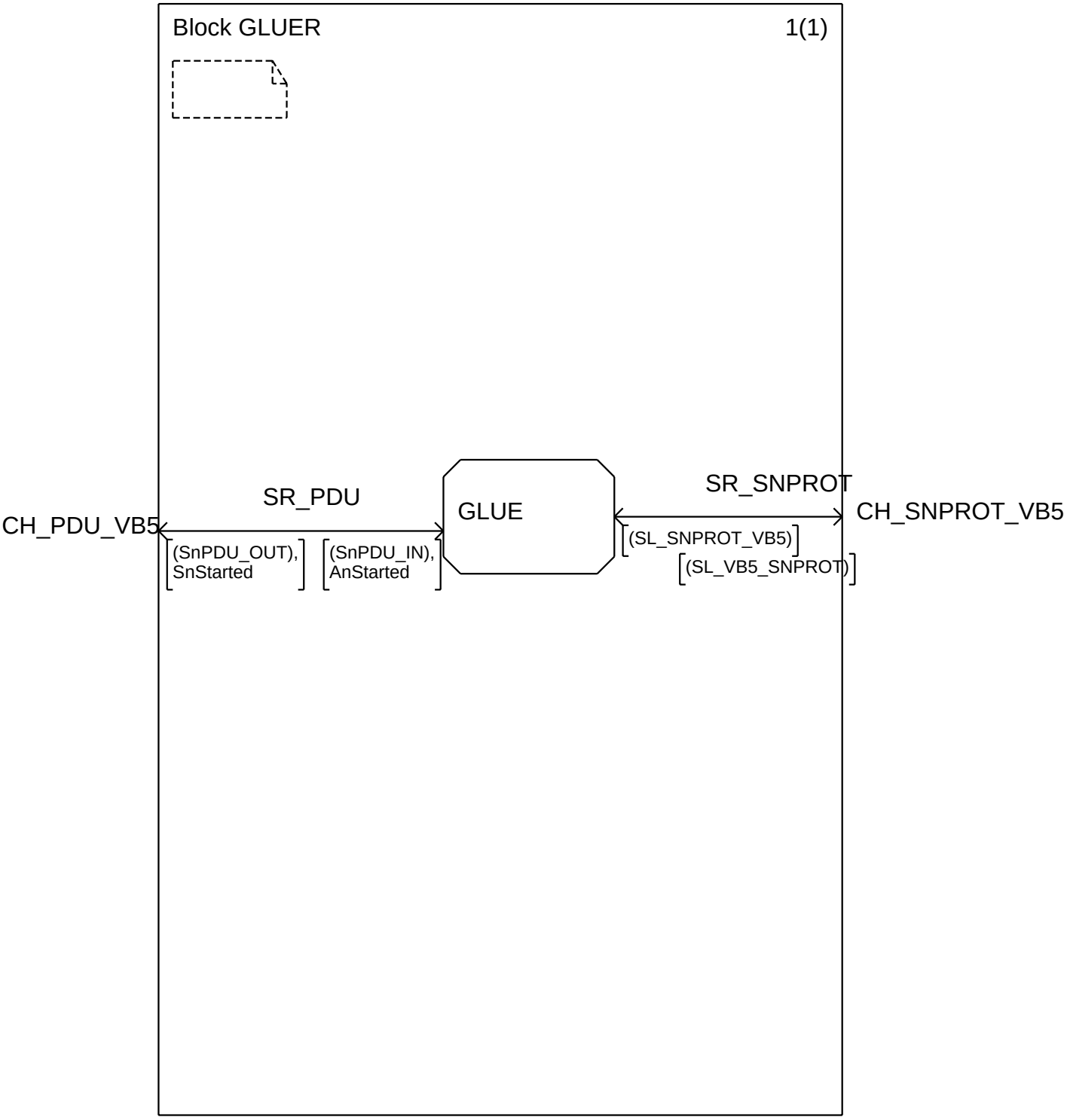

/*SN_PROT_FSM is responsible for sending/
receiving BCC messages*/
AwaitRemAck
bccAdd_
BranchAcc
(V_TransId)
V_SynResult :=
CALL ie_check
(V_ProtErrCause)
V_SynResult
C_Error
C_Proceed
ie_error_
_handling
(V_TransId,
V_ProtErrCause,
PARENT)
reset
(T_AddBranch)
mieAddBranch_
AccConf
(V_ConnRefNo,
V_BranchId)
–


AwaitRemAck
bccAdd_
BranchRej
(V_TransId,
V_RejCause)
V_SynResult :=
CALL ie_check
(V_ProtErrCause)
V_SynResult
C_Error
C_Proceed
ie_error_
_handling
(V_TransId,
V_ProtErrCause,
PARENT)
reset
(T_AddBranch)
mieAddBranch_
RejConf
(V_ConnRefNo,
V_BranchId,
V_RejCause)
–


/*SN_PROT_FSM is responsible for sending/
receiving BCC messages*/
AwaitRemAck
bccUpdate_
BranchAcc
(V_TransId)
V_SynResult :=
CALL ie_check
(V_ProtErrCause)
V_SynResult
C_Error
C_Proceed
ie_error_
_handling
(V_TransId,
V_ProtErrCause,
PARENT)
reset
(T_UpdateBranch)
mieUpdate_
BranchAccConf
(V_ConnRefNo,
V_BranchId)
–


AwaitRemAck
bccUpdate_
BranchRej
(V_TransId,
V_RejCause)
V_SynResult :=
CALL ie_check
(V_ProtErrCause)
V_SynResult
C_Error
C_Proceed
ie_error_
_handling
(V_TransId,
V_ProtErrCause,
PARENT)
reset
(T_UpdateBranch)
mieUpdate_
Branch_
RejConf
(V_ConnRefNo,
V_BranchId,
V_RejCause)
–


/*SN_PROT_FSM is responsible for sending/
receiving BCC messages*/
AwaitRemAck
bccDrop_
BranchAcc
(V_TransId)
V_SynResult :=
CALL ie_check
(V_ProtErrCause)
V_SynResult
C_Error
C_Proceed
ie_error_
_handling
(V_TransId,
V_ProtErrCause,
PARENT)
reset
(T_DropBranch)
mieDropBranch_
AccConf
(V_ConnRefNo,
V_BranchId)
–
AwaitRemAck
bccDrop_
BranchRej
(V_TransId,
V_RejCause)
V_SynResult :=
CALL ie_check
(V_ProtErrCause)
V_SynResult
C_Error
C_Proceed
ie_error_
_handling
(V_TransId,
V_ProtErrCause,
PARENT)
reset
(T_DropBranch)
mieDropBranch_
RejConf
(V_ConnRefNo,
V_BranchId,
V_RejCause)
–


Block GLUER
1(1)
CH_PDU_VB5
SR_PDU
(SnPDU_OUT),
SnStarted
(SnPDU_IN),
AnStarted
GLUE
SR_SNPROT
(SL_SNPROT_VB5)
(SL_VB5_SNPROT)
CH_SNPROT_VB5

/* GLUE is responsible for forming the PDUs into ASN1 and adding optional Information Elements. */

```
DCL
V_TransId IT_TransId,
V_ConnRefNo IT_ConnRefNo,
V_BranchId IT_BranchId,
V_RejCause IT_RejCause;
```

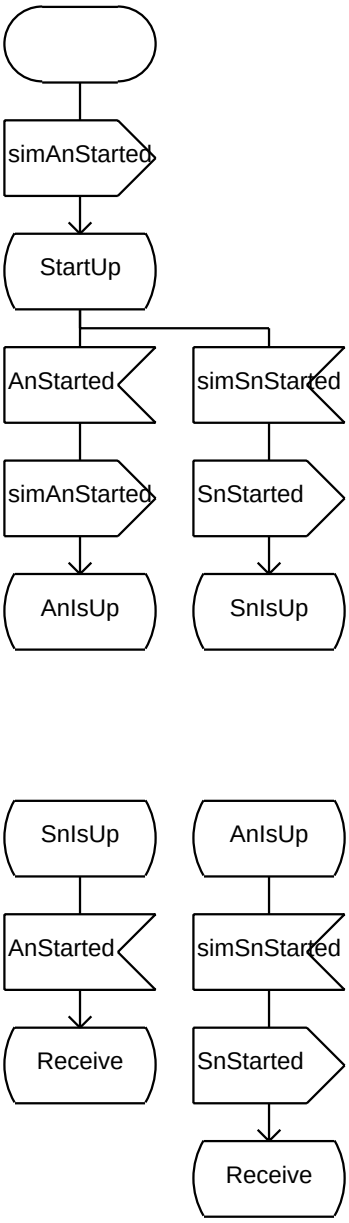
```
DCL
bbccMsg BbccMsg,
bbccReset1 BbccReset,
bbccResetAcc1 BbccResetAcc,
bbccResetRej1 BbccResetRej,
bbccPresync1 BbccPresync,
bbccPresyncAcc1 BbccPresyncAcc,
bbccPresyncRej1 BbccPresyncRej,
anFault1  AnFault,
anFaultAcc1  AnFaultAcc;
```

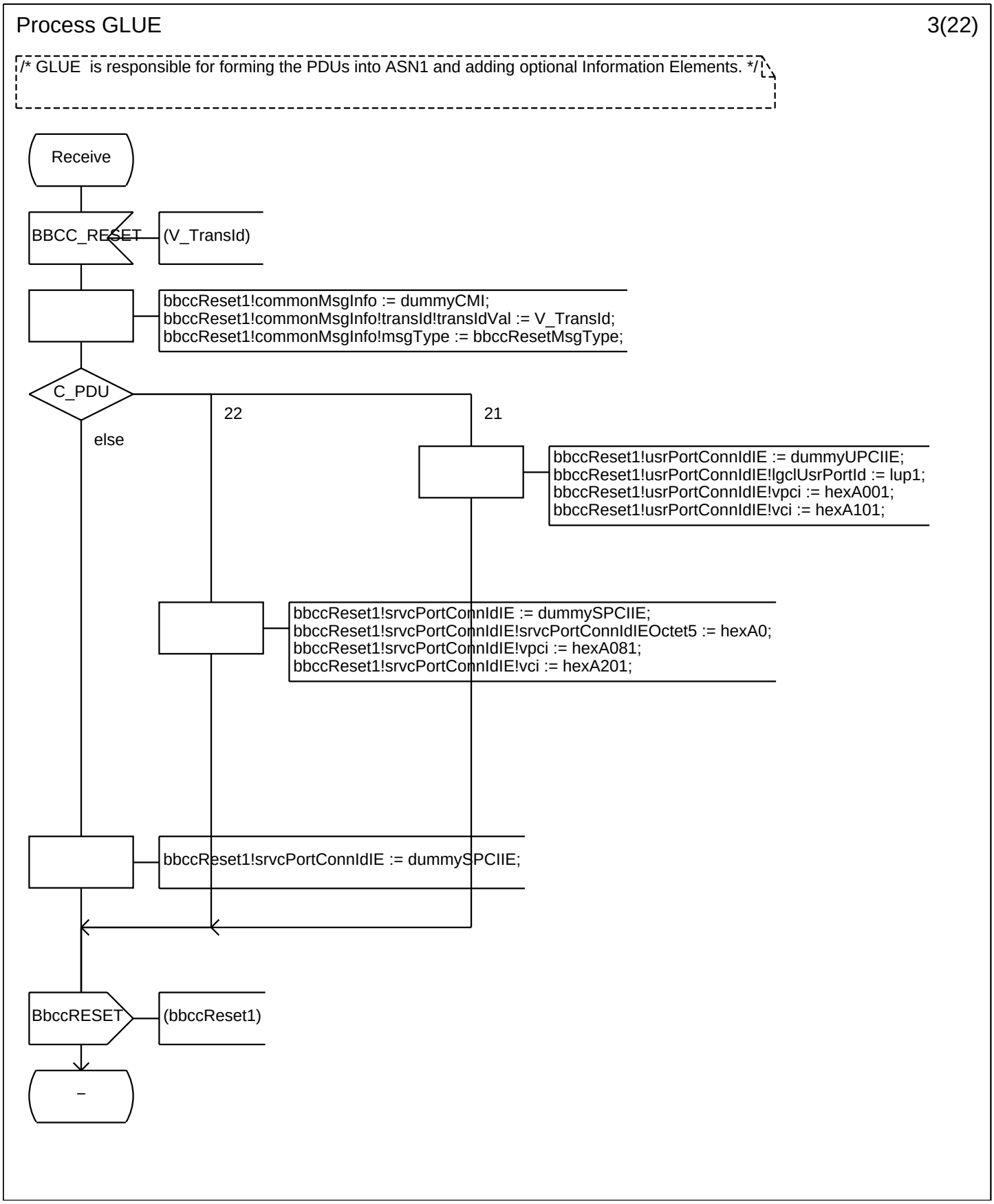
buildPorts

```
DCL
spVCI, upVCI  VCI,
spVPCI, upVPCI VPCI,
lup  LogId;
```

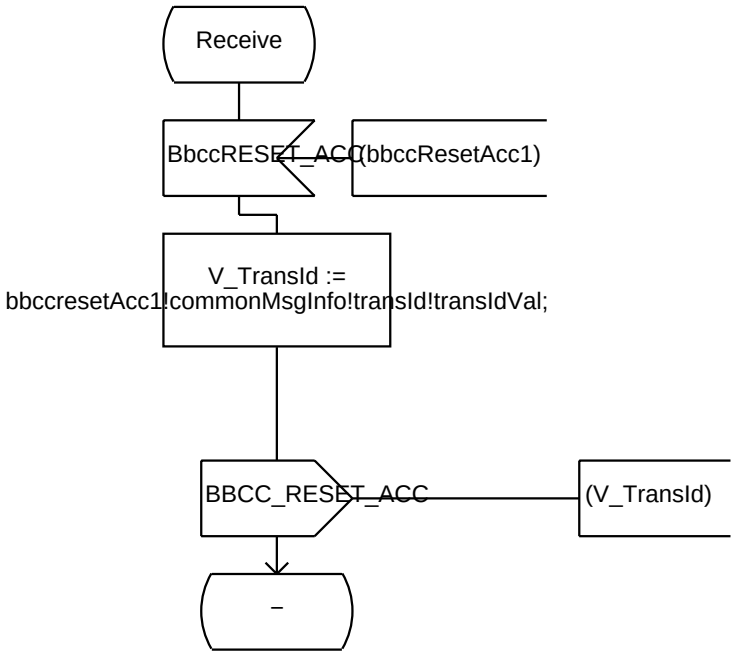
```
DCL
alloc1 Alloc,
allocAcc1 AllocAcc,
allocComp1 AllocComp,
allocCompAcc1 AllocCompAcc,
allocCompRej1 AllocCompRej,
allocRej1 AllocRej,
dealloc1 Dealloc,
deallocAcc1 DeallocAcc,
addBranch1 AddBranch,
addBranchAcc1 AddBranchAcc,
addBranchRej1 AddBranchRej,
updateBranch1 UpdateBranch,
updateBranchAcc1 UpdateBranchAcc,
updateBranchRej1 UpdateBranchRej,
dropBranch1 DropBranch,
dropBranchAcc1 DropBranchAcc,
dropBranchRej1 DropBranchRej,
modify1 Modify,
modifyAcc1 ModifyAcc,
modifyRej1 ModifyRej,
modifyComp1 ModifyComp,
modifyCompAcc1 ModifyCompAcc,
modifyCompRej1 ModifyCompRej,
modifyAbort1  ModifyAbort,
modifyAbortAcc1 ModifyAbortAcc,
modifyAbortRej1 ModifyAbortRej,
truncInfoANFault1 TruncInfoANFault,
aNFaultMissingMndtryIE1 ANFaultMissingMndtryIE,
aNFaultXtraUnkIE1 ANFaultXtraUnkIE,
aNFaultXtraUnexpIE1 ANFaultXtraUnexpIE,
protocolError1 ProtocolError,
unknownPDU1 UnknownPDU;
```

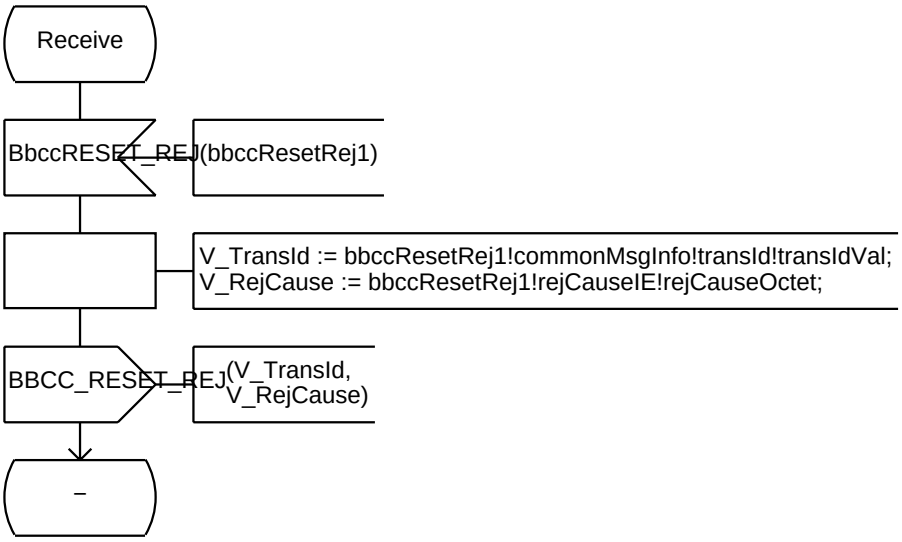
/* GLUE is responsible for forming the PDUs into ASN1 and adding optional Information Elements. */



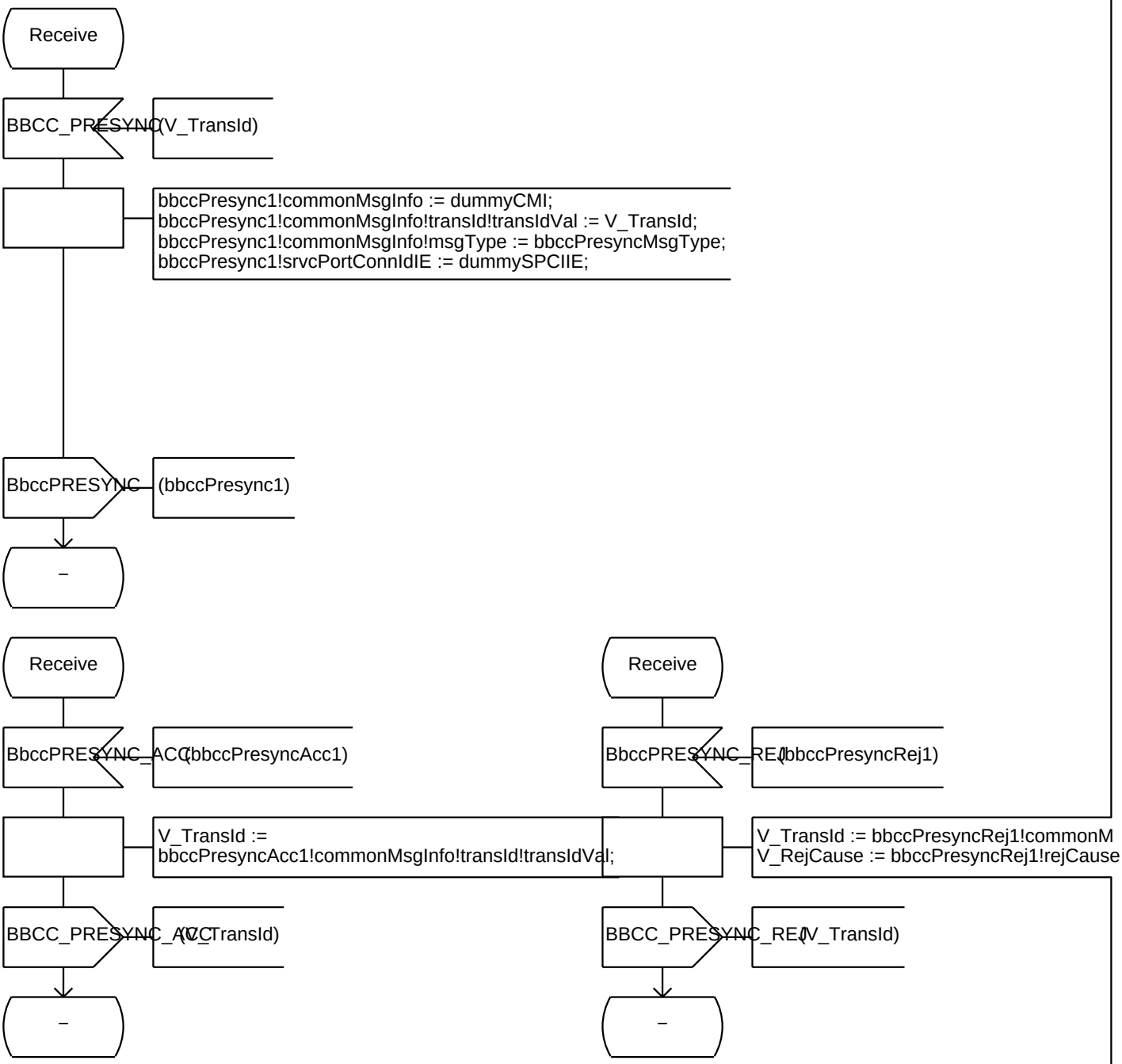

/* GLUE is responsible for forming the PDUs into ASN1 and adding optional Information Elements. */
Receive
BBCC_RESET
(V_TransId)
bbccReset1!commonMsgInfo := dummyCMI;
bbccReset1!commonMsgInfo!transId!transIdVal := V_TransId;
bbccReset1!commonMsgInfo!msgType := bbccResetMsgType;
C_PDU
22
21
else
bbccReset1!usrPortConnIdIE := dummyUPCIIE;
bbccReset1!usrPortConnIdIE!lgclUsrPortId := lup1;
bbccReset1!usrPortConnIdIE!vpci := hexA001;
bbccReset1!usrPortConnIdIE!vci := hexA101;
bbccReset1!srvcPortConnIdIE := dummySPCIIE;
bbccReset1!srvcPortConnIdIE!srvcPortConnIdIEOctet5 := hexA0;
bbccReset1!srvcPortConnIdIE!vpci := hexA081;
bbccReset1!srvcPortConnIdIE!vci := hexA201;
bbccReset1!srvcPortConnIdIE := dummySPCIIE;
BbccRESET
(bbccReset1)
–

/* GLUE is responsible for forming the PDUs into ASN1 and adding optional Information Elements. */

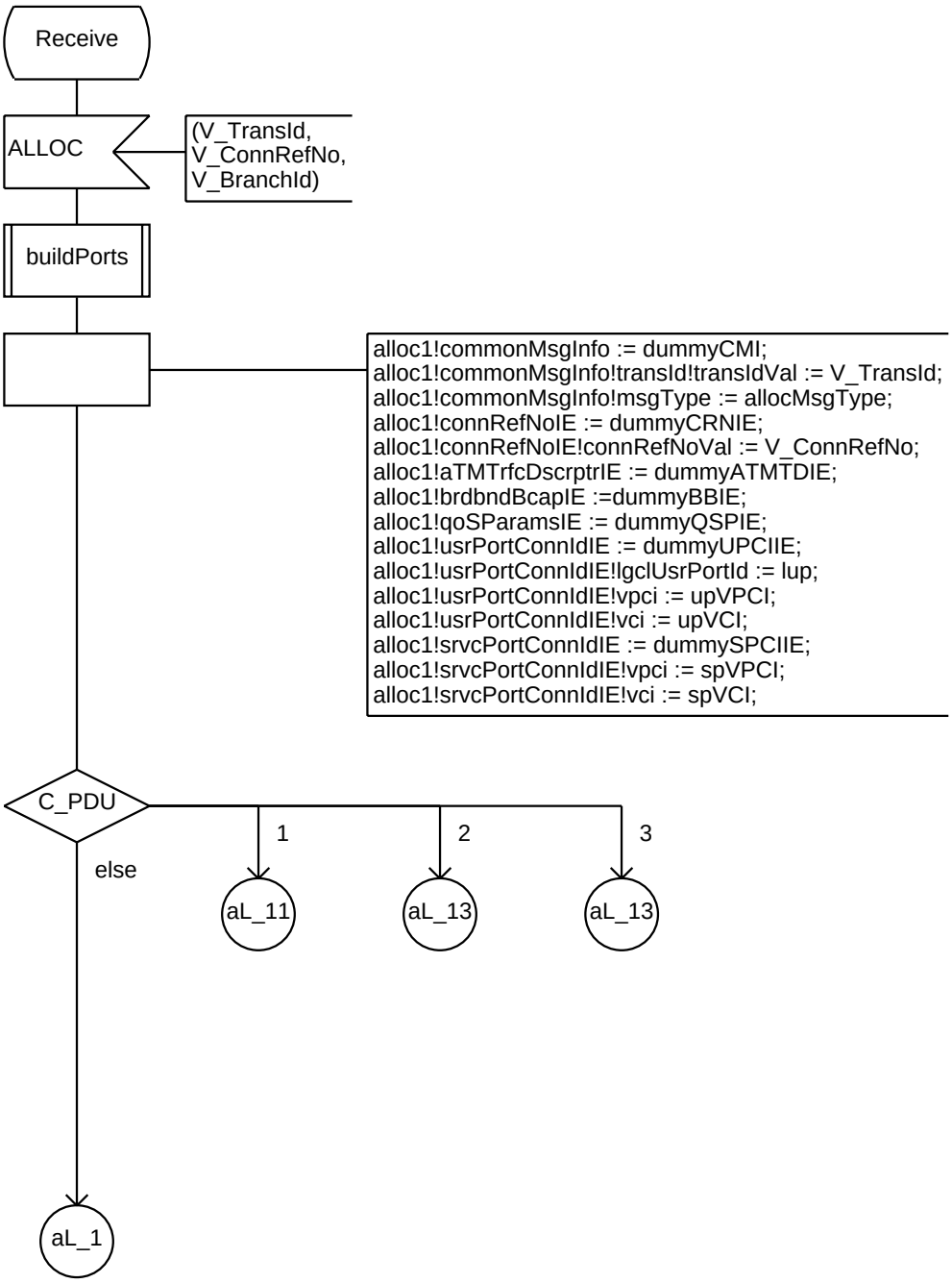




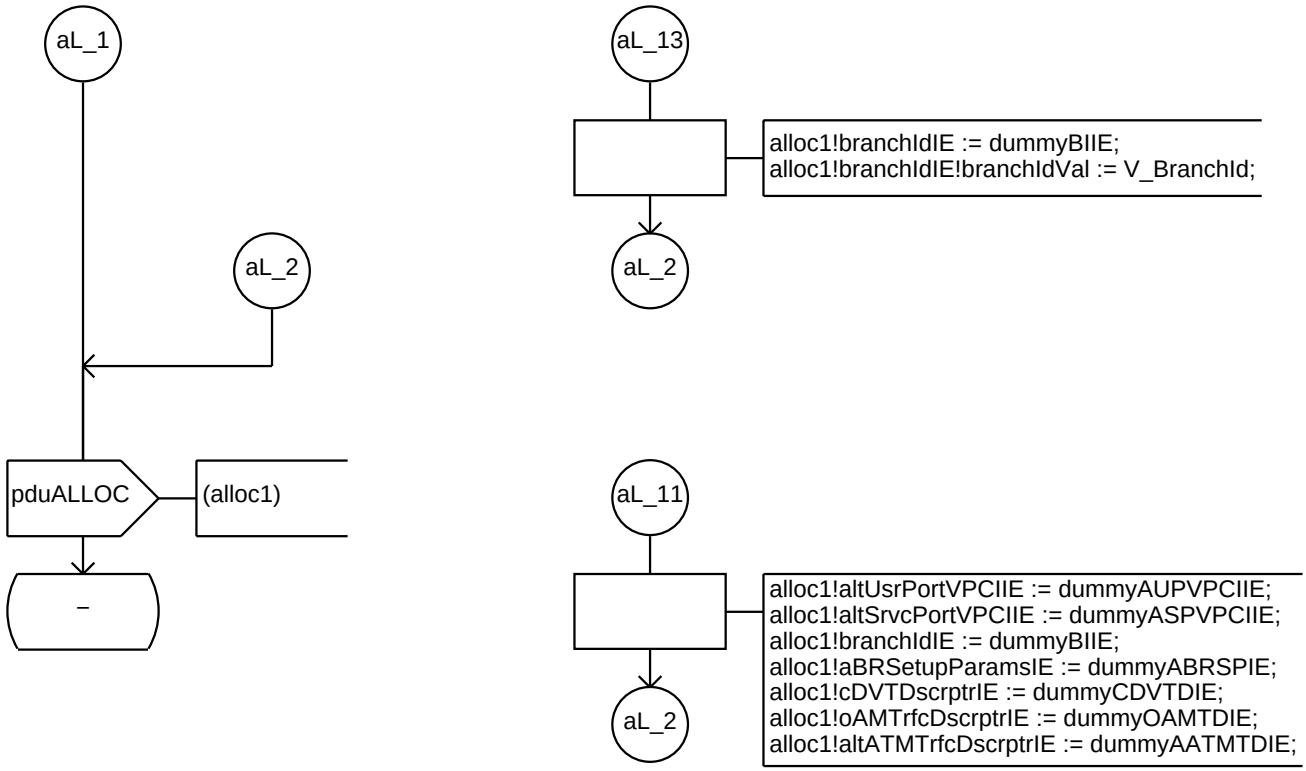
/* GLUE is responsible for forming the PDUs into ASN1 and adding optional Information Elements. */



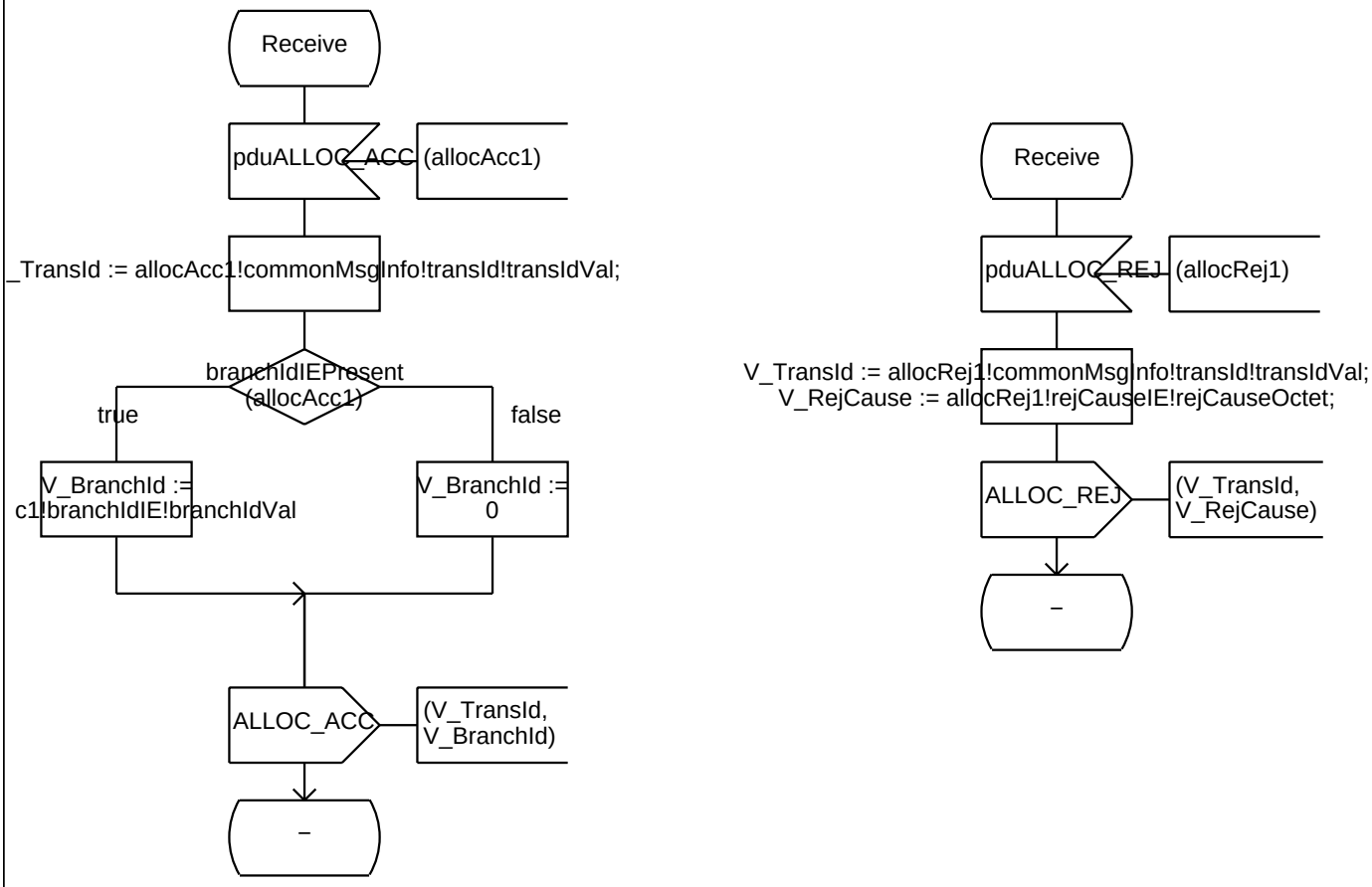
/* GLUE is responsible for forming the PDUs into ASN1 and adding optional Information Elements. */



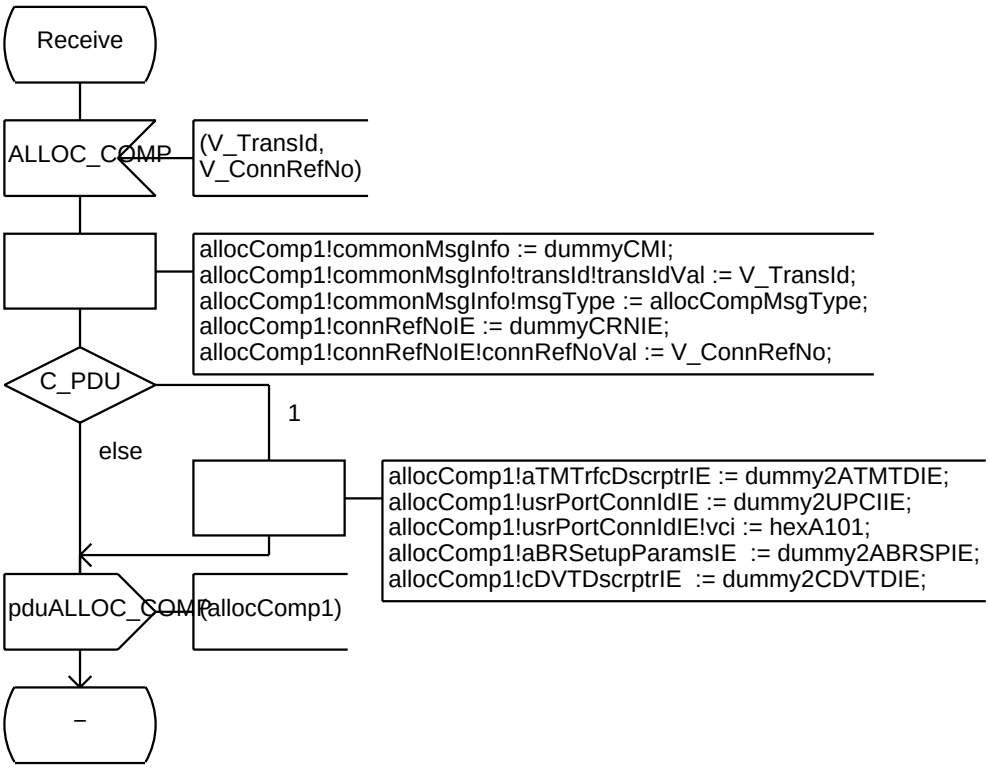
/* GLUE is responsible for forming the PDUs into ASN1 and adding optional Information Elements. */



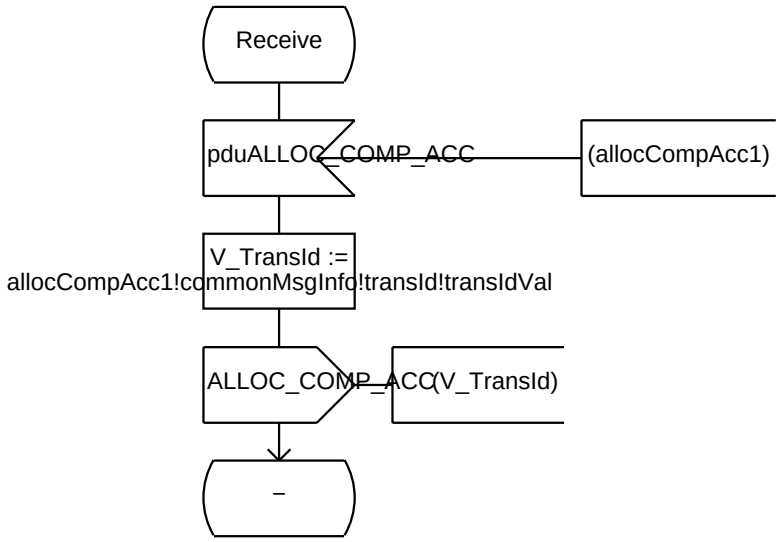
/* GLUE is responsible for forming the PDUs into ASN1 and adding optional Information Elements. */

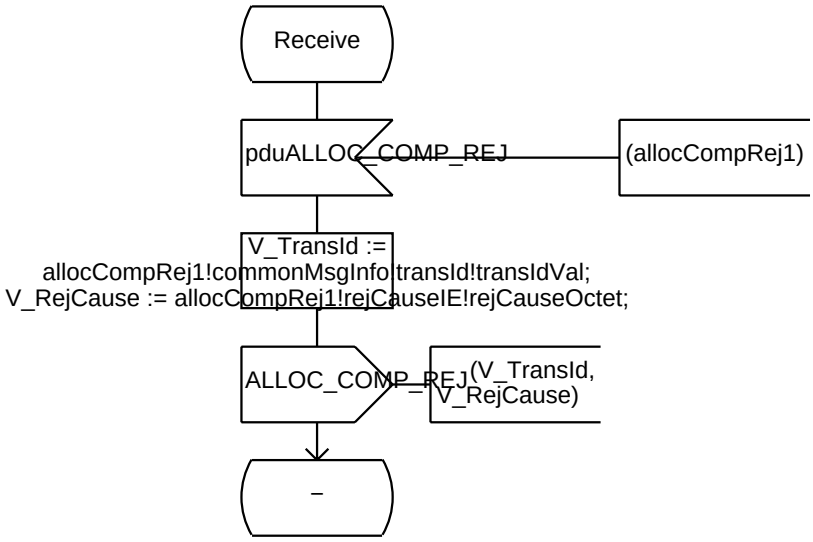


/* GLUE is responsible for forming the PDUs into ASN1 and adding optional Information Elements. */

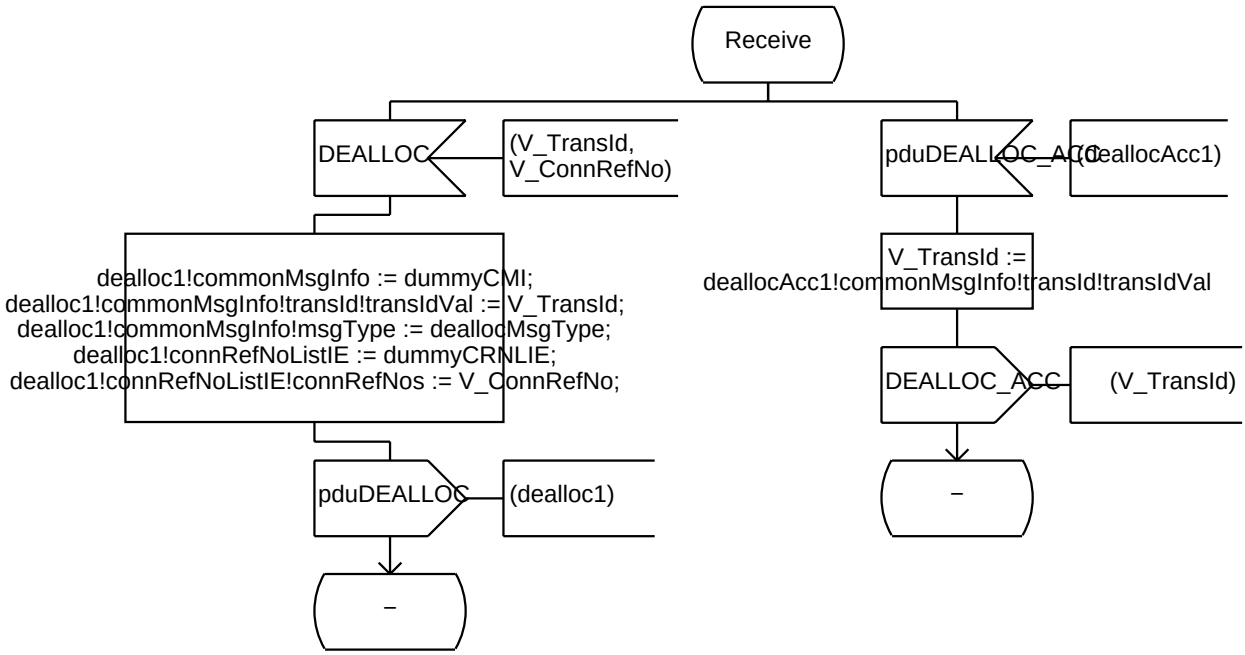


/* GLUE is responsible for forming the PDUs into ASN1 and adding optional Information Elements. */

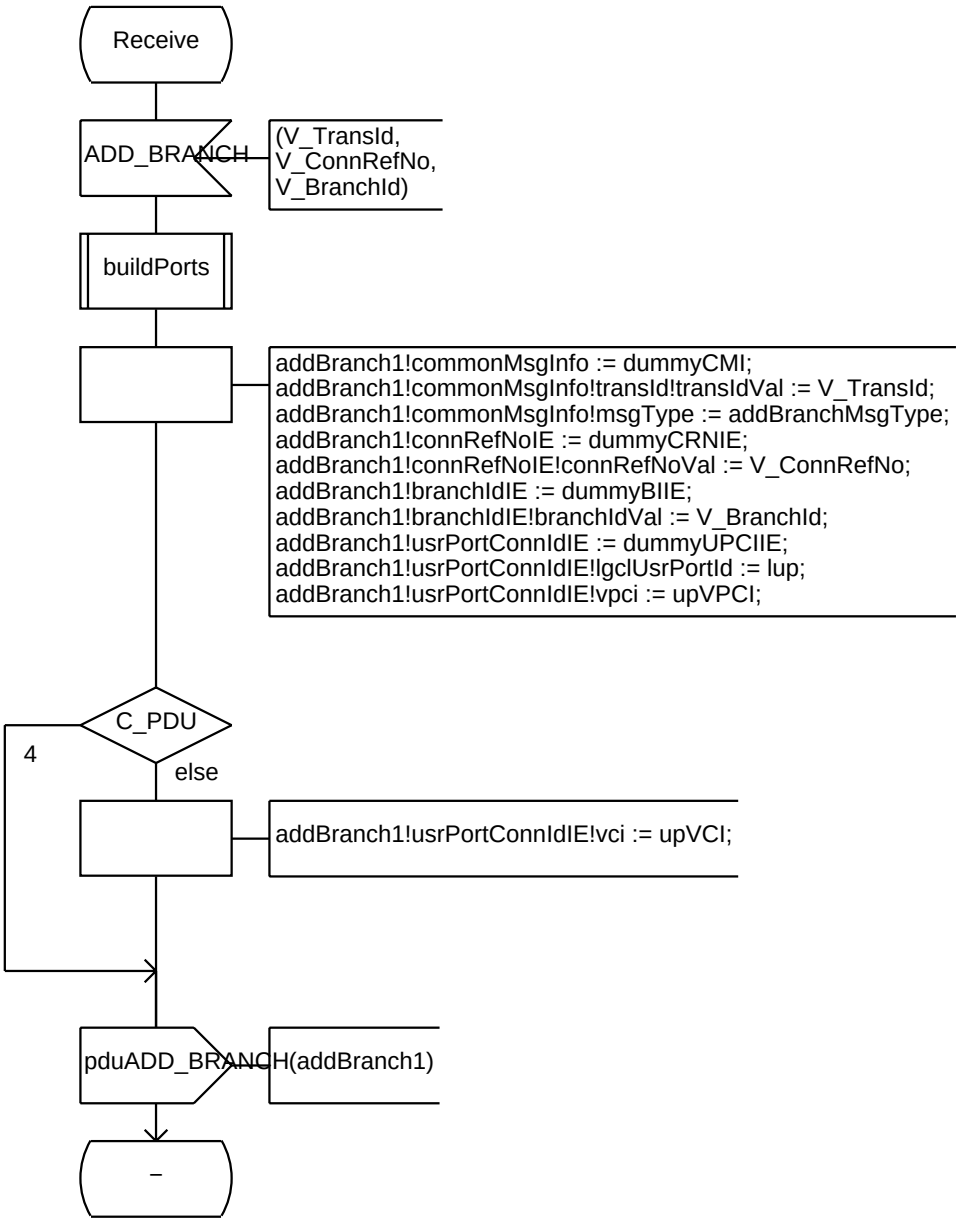




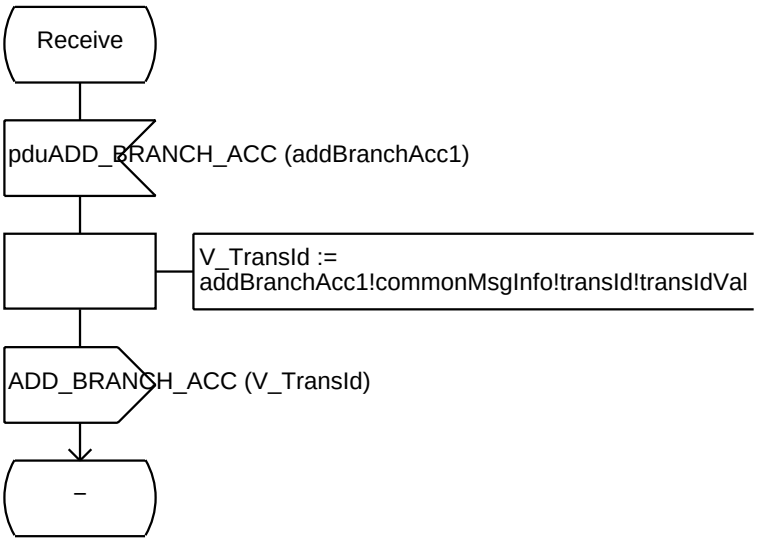
/* GLUE is responsible for forming the PDUs into ASN1 and adding optional Information Elements. */

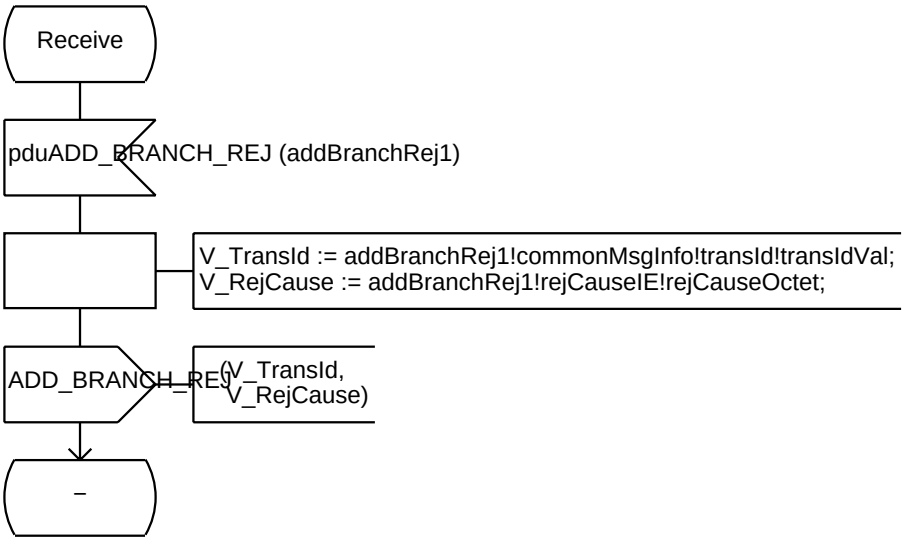


/* GLUE is responsible for forming the PDUs into ASN1 and adding optional Information Elements. */

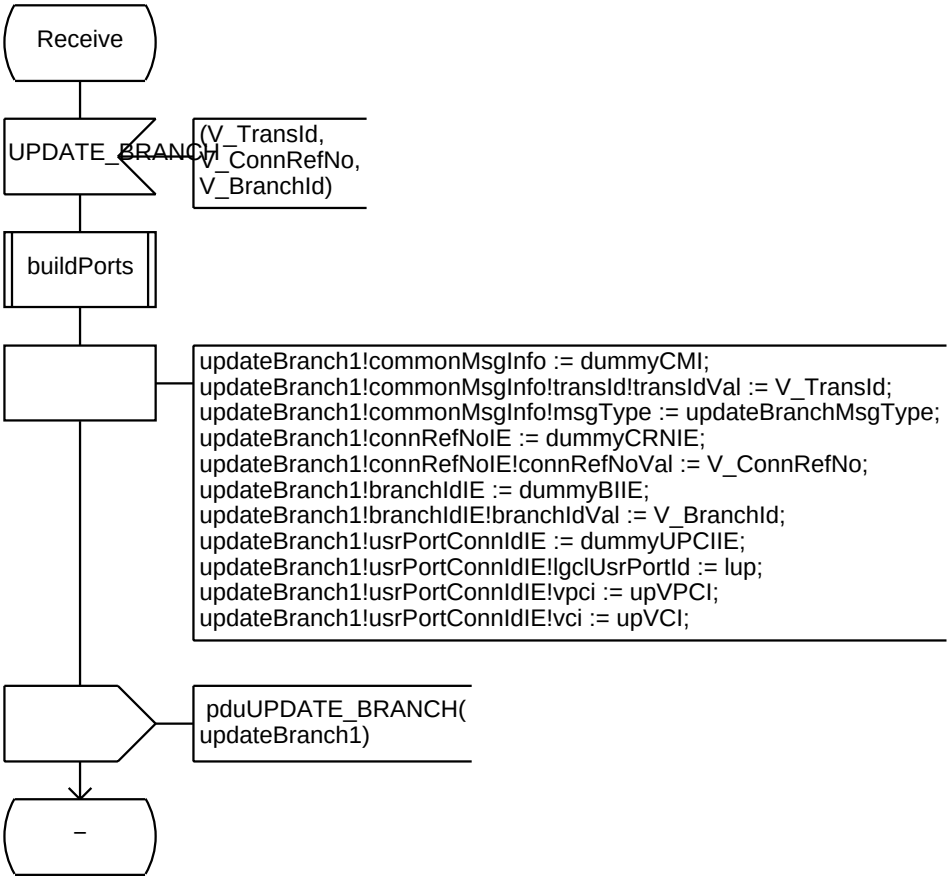


/* GLUE is responsible for forming the PDUs into ASN1 and adding optional Information Elements. */

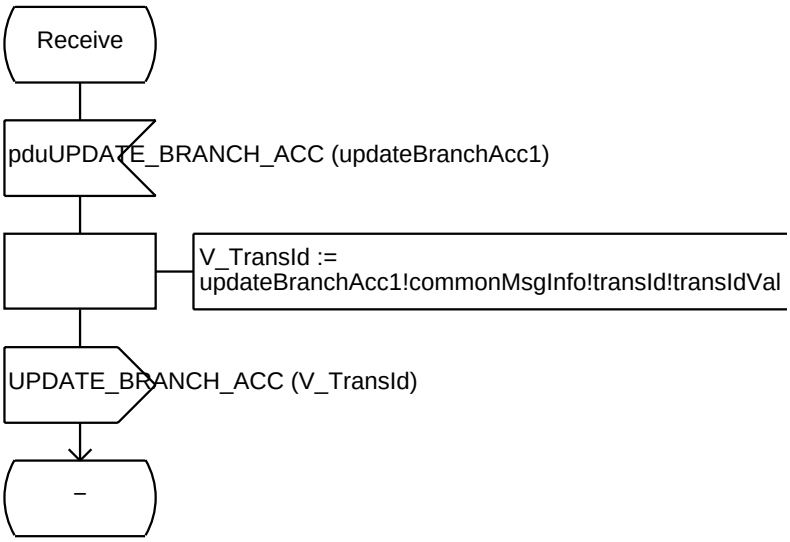


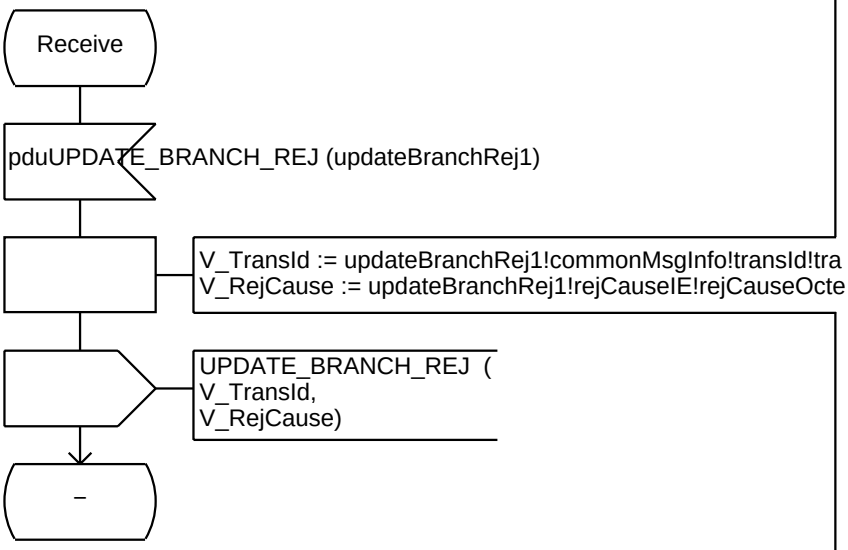


/* GLUE is responsible for forming the PDUs into ASN1 and adding optional Information Elements. */



/* GLUE is responsible for forming the PDUs into ASN1 and adding optional Information Elements. */





/* GLUE is responsible for forming the PDUs into ASN1 and adding optional Information Elements. */

Receive

DROP_BRANCH (V_TransId, V_ConnRefNo, V_BranchId)

dropBranch1!commonMsgInfo := dummyCMI;
dropBranch1!commonMsgInfo!transId!transIdVal := V_TransId;
dropBranch1!commonMsgInfo!msgType := dropBranchMsgType;
dropBranch1!connRefNoIE := dummyCRNIE;
dropBranch1!connRefNoIE!connRefNoVal := V_ConnRefNo;
dropBranch1!branchIdListIE := dummyBILIE;
dropBranch1!branchIdListIE!branchIds := V_BranchId;

pduDROP_BRANCH (dropBranch1)

–

pduDROP_BRANCH_ACC (dropBranchAcc1)

V_TransId := dropBranchAcc1!commonMsgInfo!transId!

DROP_BRANCH_ACC (V_TransId)

–

Receive

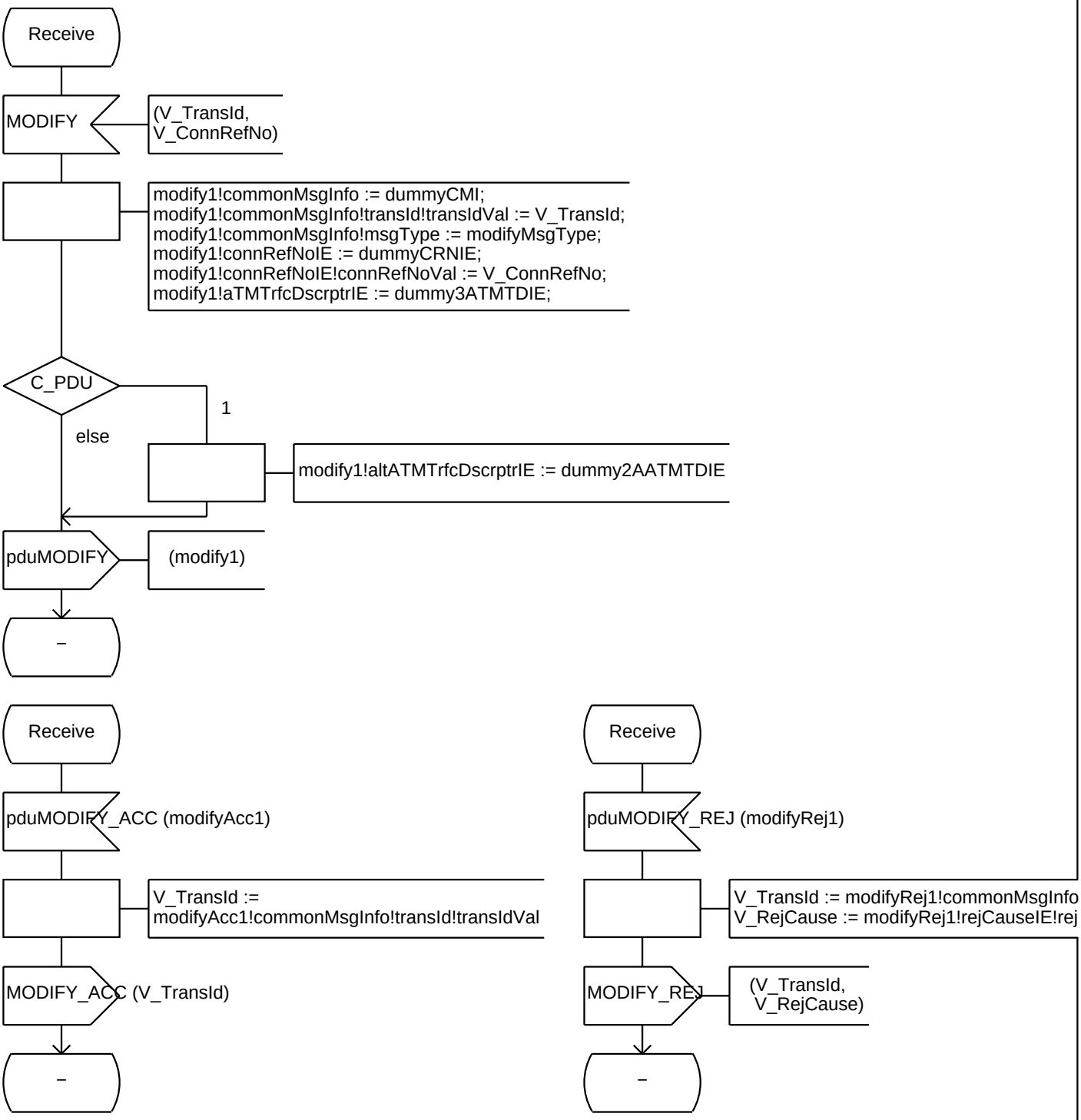
pduDROP_BRANCH_REJ (dropBranchRej1)

V_TransId := dropBranchRej1!commonMsgInfo!transId!transIdVal;
V_RejCause := dropBranchRej1!rejCauseIE!rejCauseOctet;

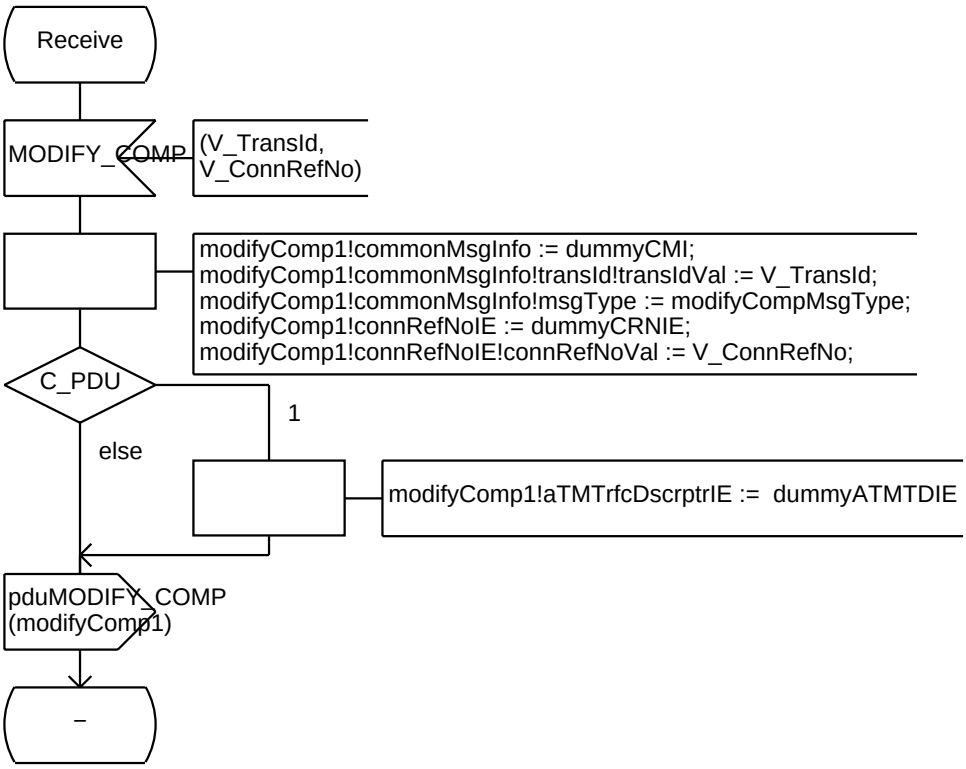
DROP_BRANCH_REJ (V_TransId, V_RejCause)

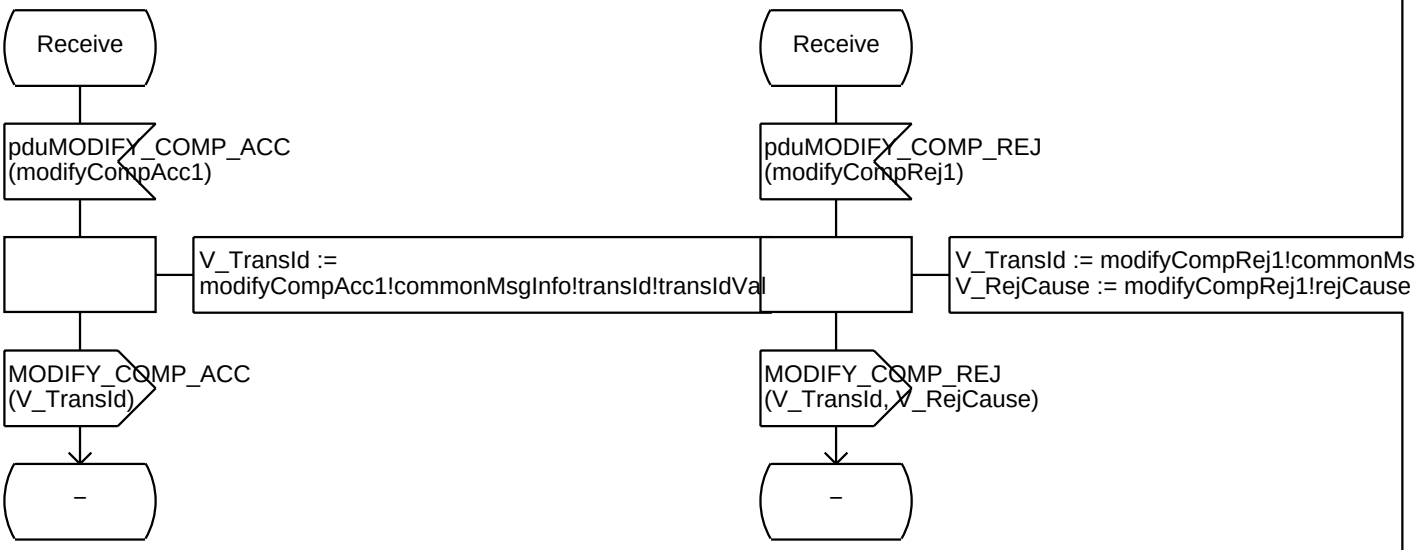
–

/* GLUE is responsible for forming the PDUs into ASN1 and adding optional Information Elements. */

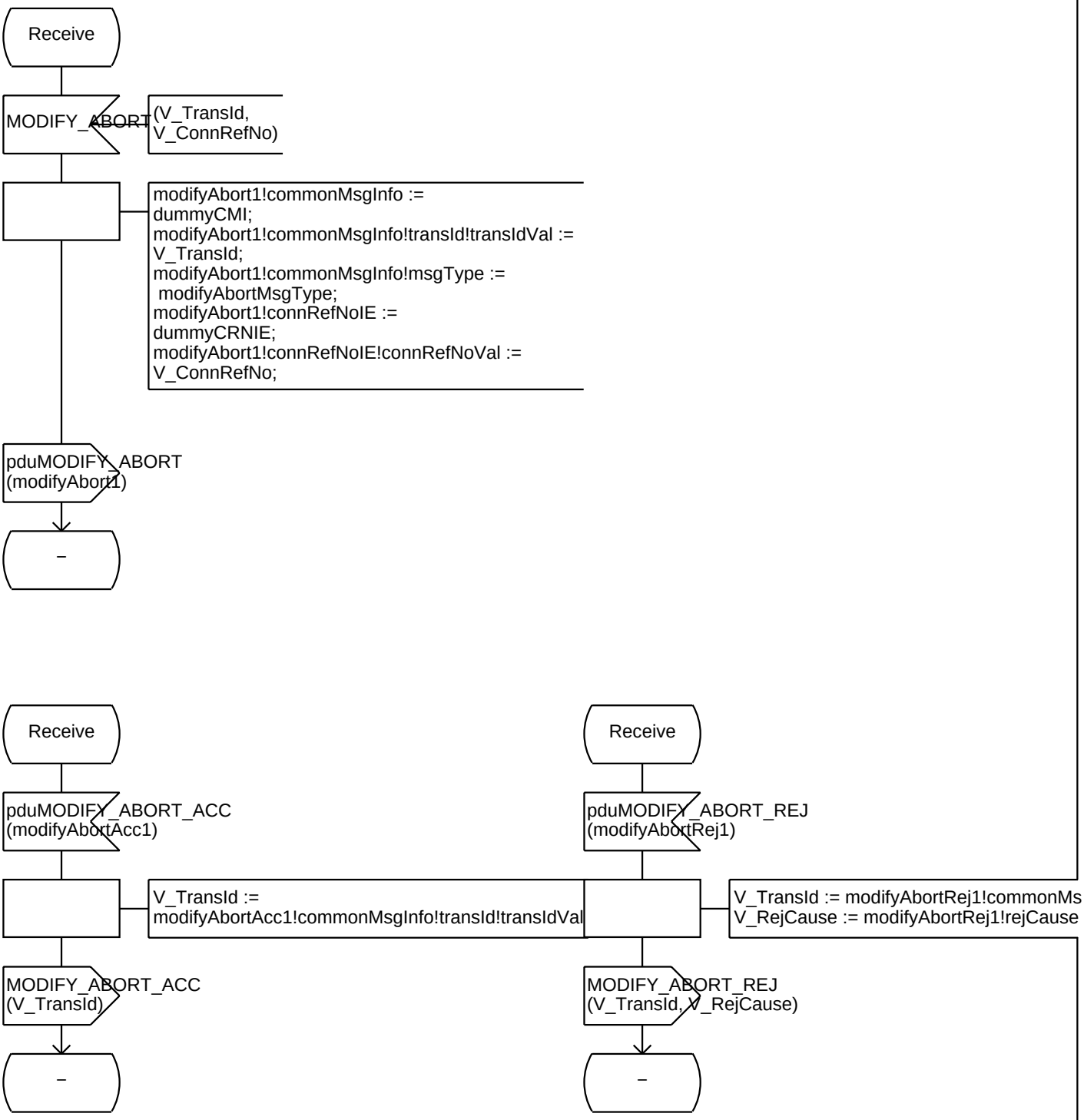


/* GLUE is responsible for forming the PDUs into ASN1 and adding optional Information Elements. */

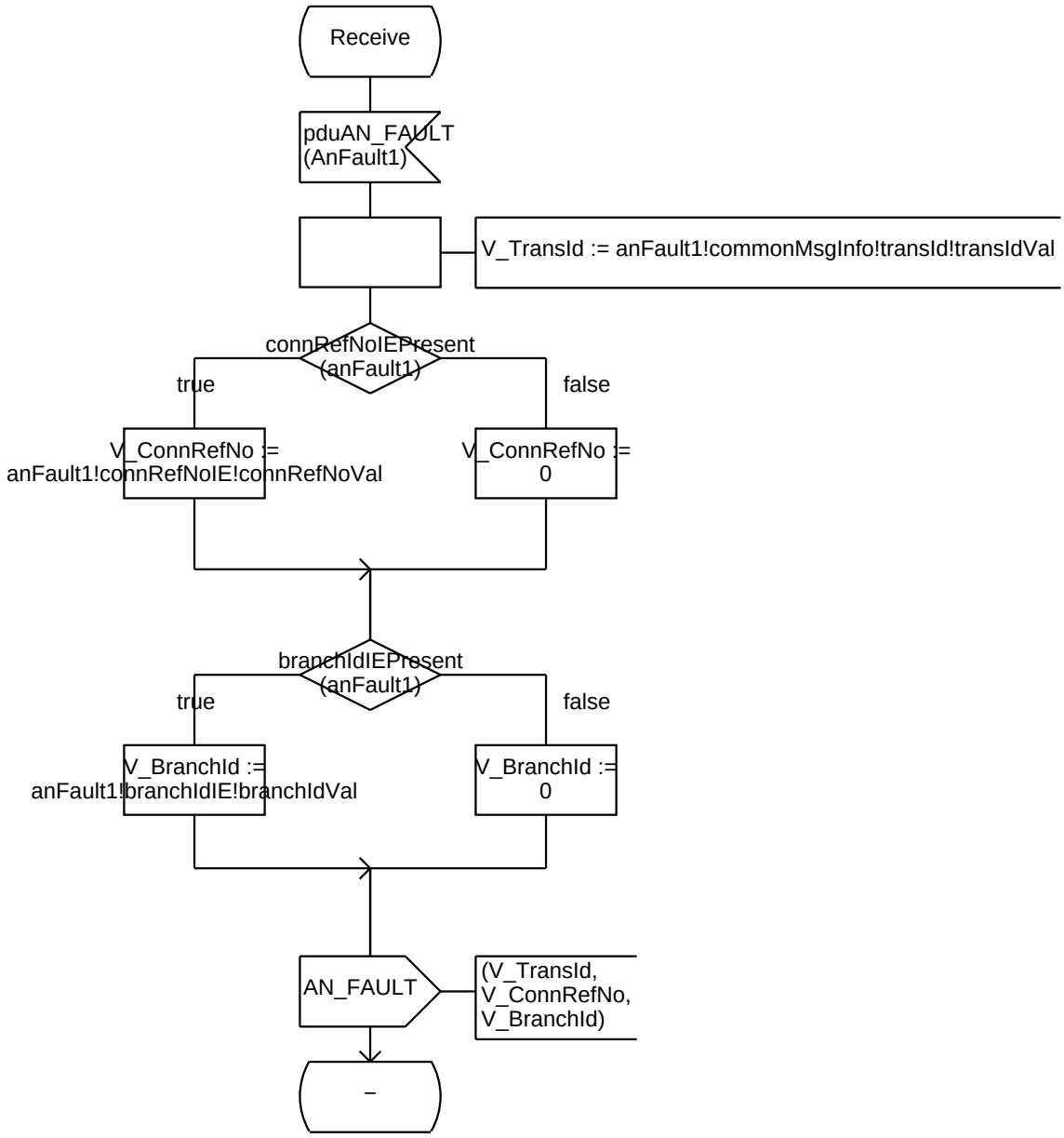




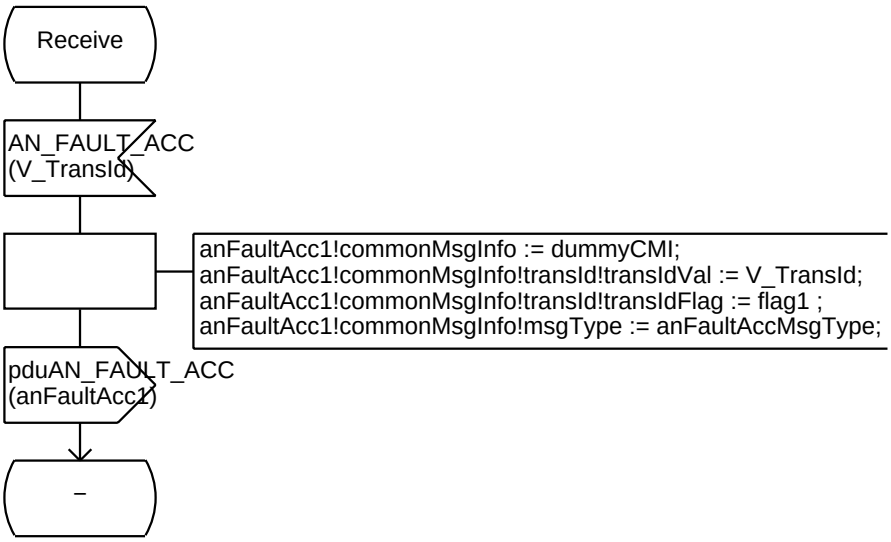
/* GLUE is responsible for forming the PDUs into ASN1 and adding optional Information Elements. */



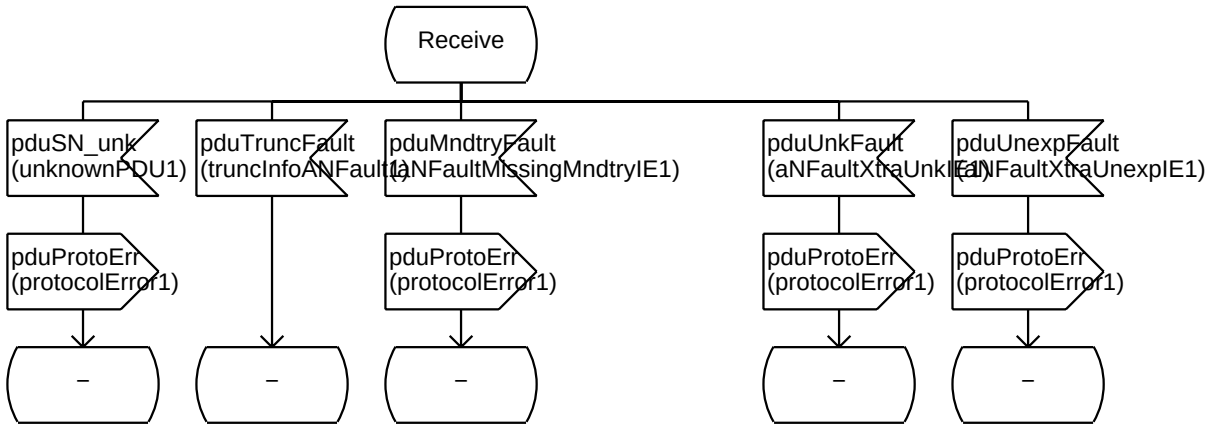
/* GLUE is responsible for forming the PDUs into ASN1 and adding optional Information Elements. */



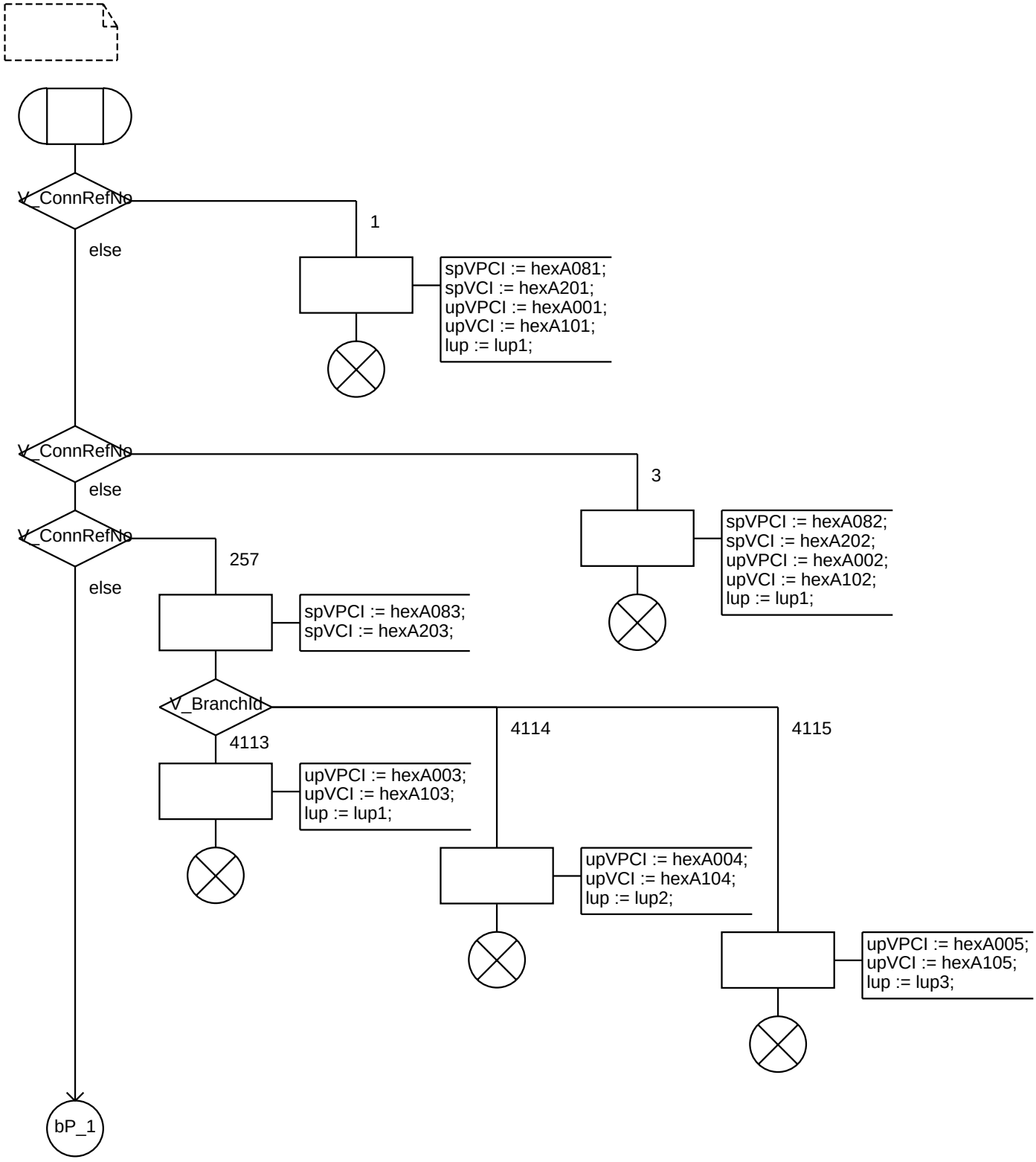
/* GLUE is responsible for forming the PDUs into ASN1 and adding optional Information Elements. */



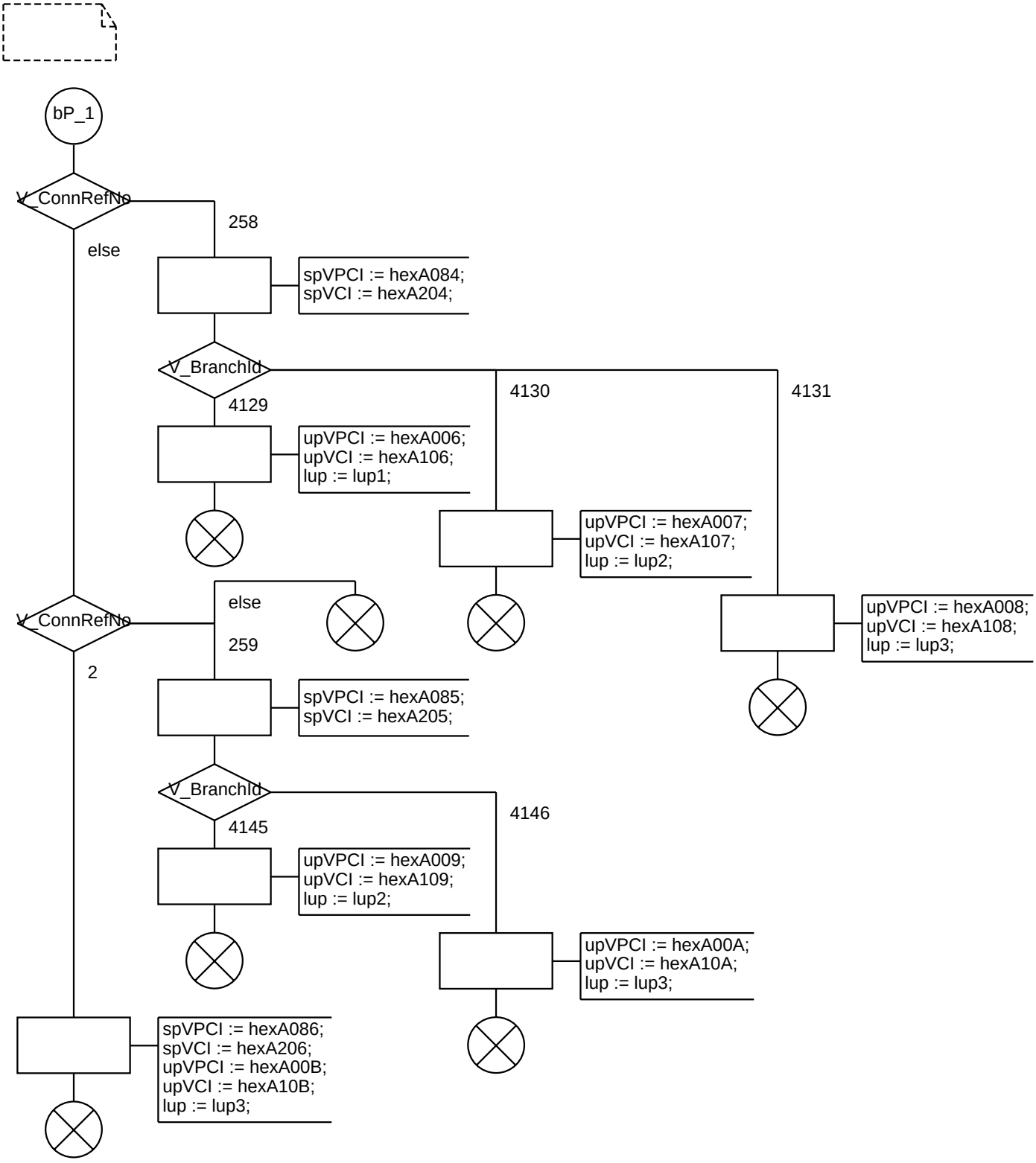
/* GLUE is responsible for forming the PDUs into ASN1 and adding optional Information Elements. */



Procedure buildPorts



Procedure buildPorts



CH_SNMGT_CONN
CH_SNCCF_CONN
Block SN_CONNMGT
1(4)
(SL_CMGR_SNMGT)
(SL_SNCMGR_CCF)
SR_MGT_CMGR
SR_CCF_CMGR
(SL_SNMGT_CMGR)
(SL_CCF_SNCMGR)
SN_CONN_MGR
SR_MGR_CFSM
SN_CONN_FSM(0)
(SL_CFSM_MGR)
(SL_MGR_CFSM)
(SL_SNPROT_CMGR)
(SL_BFSM_CFSM)
SR_CFSM_BFSM
(SL_CFSM_BFSM)
SR_CMGR_PROT
SN_BRANCH_FSM(0)
(SL_CMGR_SNPROT)
CH_SNCONN_PROT

```
/* Declaration of primitives internal
   to block SN_CONNMGT, between  SN_CONN_MGR
   and  SN_CONN_FSM*/
```

```
signal
mieAllocInd        (IT_BranchId),
mieAllocCompInd,
mieModifyInd,
mieModifyCompInd,
mieModifyAbortInd,
mieAddBranchInd    (IT_BranchId),
mieUpdateBranchInd (IT_BranchId),
mieDropBranchInd   (IT_BranchId),
mieDeallocInd      ,
mieResetInd        (IT_BranchId);
```

```
signal
mieAllocAccReq        (IT_ConnRefNo, IT_BranchId),
mieAllocAccConf       (IT_ConnRefNo, IT_BranchId),
mieAllocRejReq        (IT_ConnRefNo, IT_BranchId, IT_RejCause),
mieAllocRejConf       (IT_ConnRefNo, IT_BranchId, IT_RejCause),
mieAllocCompAccReq    (IT_ConnRefNo),
mieAllocCompAccConf   (IT_ConnRefNo),
mieAllocCompRejReq    (IT_ConnRefNo, IT_RejCause),
mieAllocCompRejConf   (IT_ConnRefNo, IT_RejCause),
mieDeallocAccReq      (IT_ConnRefNo),
mieDeallocAccConf     (IT_ConnRefNo),
mieDeallocRejReq      (IT_ConnRefNo, IT_RejCause),
mieDeallocRejConf     (IT_ConnRefNo, IT_RejCause),
mieModifyAccReq       (IT_ConnRefNo),
mieModifyAccConf      (IT_ConnRefNo),
mieModifyRejReq       (IT_ConnRefNo, IT_RejCause),
mieModifyRejConf      (IT_ConnRefNo, IT_RejCause),
mieModifyCompAccReq   (IT_ConnRefNo),
mieModifyCompAccConf  (IT_ConnRefNo),
mieModifyCompRejReq   (IT_ConnRefNo, IT_RejCause),
mieModifyCompRejConf  (IT_ConnRefNo, IT_RejCause),
mieModifyAbortAccReq  (IT_ConnRefNo),
mieModifyAbortAccConf (IT_ConnRefNo),
mieModifyAbortRejReq  (IT_ConnRefNo, IT_RejCause),
mieModifyAbortRejConf (IT_ConnRefNo, IT_RejCause),
mieAddBranchAccReq    (IT_ConnRefNo, IT_BranchId),
mieAddBranchAccConf   (IT_ConnRefNo, IT_BranchId),
mieAddBranchRejReq    (IT_ConnRefNo, IT_BranchId, IT_RejCause),
mieAddBranchRejConf   (IT_ConnRefNo, IT_BranchId, IT_RejCause),
mieUpdateBranchAccReq (IT_ConnRefNo, IT_BranchId),
mieUpdateBranchAccConf (IT_ConnRefNo, IT_BranchId),
mieUpdateBranchRejReq (IT_ConnRefNo, IT_BranchId, IT_RejCause),
mieUpdateBranchRejConf (IT_ConnRefNo, IT_BranchId, IT_RejCause),
mieDropBranchAccReq   (IT_ConnRefNo, IT_BranchId),
mieDropBranchAccConf  (IT_ConnRefNo, IT_BranchId),
mieDropBranchRejReq   (IT_ConnRefNo, IT_BranchId, IT_RejCause),
mieDropBranchRejConf  (IT_ConnRefNo, IT_BranchId, IT_RejCause);
```

```
/* Declaration of primitives internal
   to block SN_CONNMGT, between SN_CONN_FSM
   and SN_BRANCH_FSM */
```

```
signal
mieUpdateBranchFSMInd     ,
mieDropBranchFSMInd       ,
mieAddBranchFSMAccReq     (IT_BranchId),
mieFirstAddBranchFSMAccInd(IT_BranchId),
mieAddBranchFSMAccConf    (IT_BranchId),
mieAddBranchFSMRejReq     (IT_BranchId, IT_RejCause),
mieAddBranchFSMRejConf    (IT_BranchId, IT_RejCause),
mieUpdateBranchFSMAccReq  (IT_BranchId),
mieUpdateBranchFSMAccConf (IT_BranchId),
mieUpdateBranchFSMRejReq  (IT_BranchId, IT_RejCause),
mieUpdateBranchFSMRejConf (IT_BranchId, IT_RejCause),
mieDropBranchFSMAccReq    (IT_BranchId),
mieDropBranchFSMAccConf   (IT_BranchId),
mieDropBranchFSMRejReq    (IT_BranchId, IT_RejCause),
mieDropBranchFSMRejConf   (IT_BranchId, IT_RejCause),
mieDelBranchFSMInd        ;
```

/* Declaration of signallists
internal to block SN_CONNMGT */

```
signallist SL_MGR_CFSM =
mieAllocInd,
mieAllocCompInd,
mieModifyInd,
mieModifyCompInd,
mieModifyAbortInd,
mieAddBranchInd,
mieUpdateBranchInd,
mieDropBranchInd,
mieDeallocInd,
mieAllocAccReq,
mieAllocRejReq,
mieAllocCompAccReq,
mieAllocCompRejReq,
mieDeallocAccReq,
mieDeallocRejReq,
mieModifyAccReq,
mieModifyRejReq,
mieModifyCompAccReq,
mieModifyCompRejReq,
mieModifyAbortAccReq,
mieModifyAbortRejReq,
mieAddBranchAccReq,
mieAddBranchRejReq,
mieUpdateBranchAccReq,
mieUpdateBranchRejReq,
mieDropBranchAccReq,
mieDropBranchRejReq,
mieResetInd;
```

```
signallist SL_CFSM_MGR =
mieAllocAccConf,
mieAllocRejConf,
mieAllocCompAccConf,
mieAllocCompRejConf,
mieDeallocAccConf,
mieDeallocRejConf,
mieModifyAccConf,
mieModifyRejConf,
mieModifyCompAccConf,
mieModifyCompRejConf,
mieModifyAbortAccConf,
mieModifyAbortRejConf,
mieAddBranchAccConf,
mieAddBranchRejConf,
mieUpdateBranchAccConf,
mieUpdateBranchRejConf,
mieDropBranchAccConf,
mieDropBranchRejConf;
```

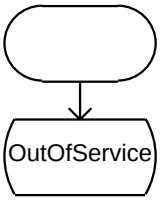
```
signallist SL_CFSM_BFSM =
mieUpdateBranchFSMInd,
mieDropBranchFSMInd,
mieAddBranchFSMAccReq,
mieFirstAddBranchFSMAccInd,
mieAddBranchFSMRejReq,
mieUpdateBranchFSMAccReq,
mieUpdateBranchFSMRejReq,
mieDropBranchFSMAccReq,
mieDropBranchFSMRejReq,
mieDelBranchFSMInd;
```

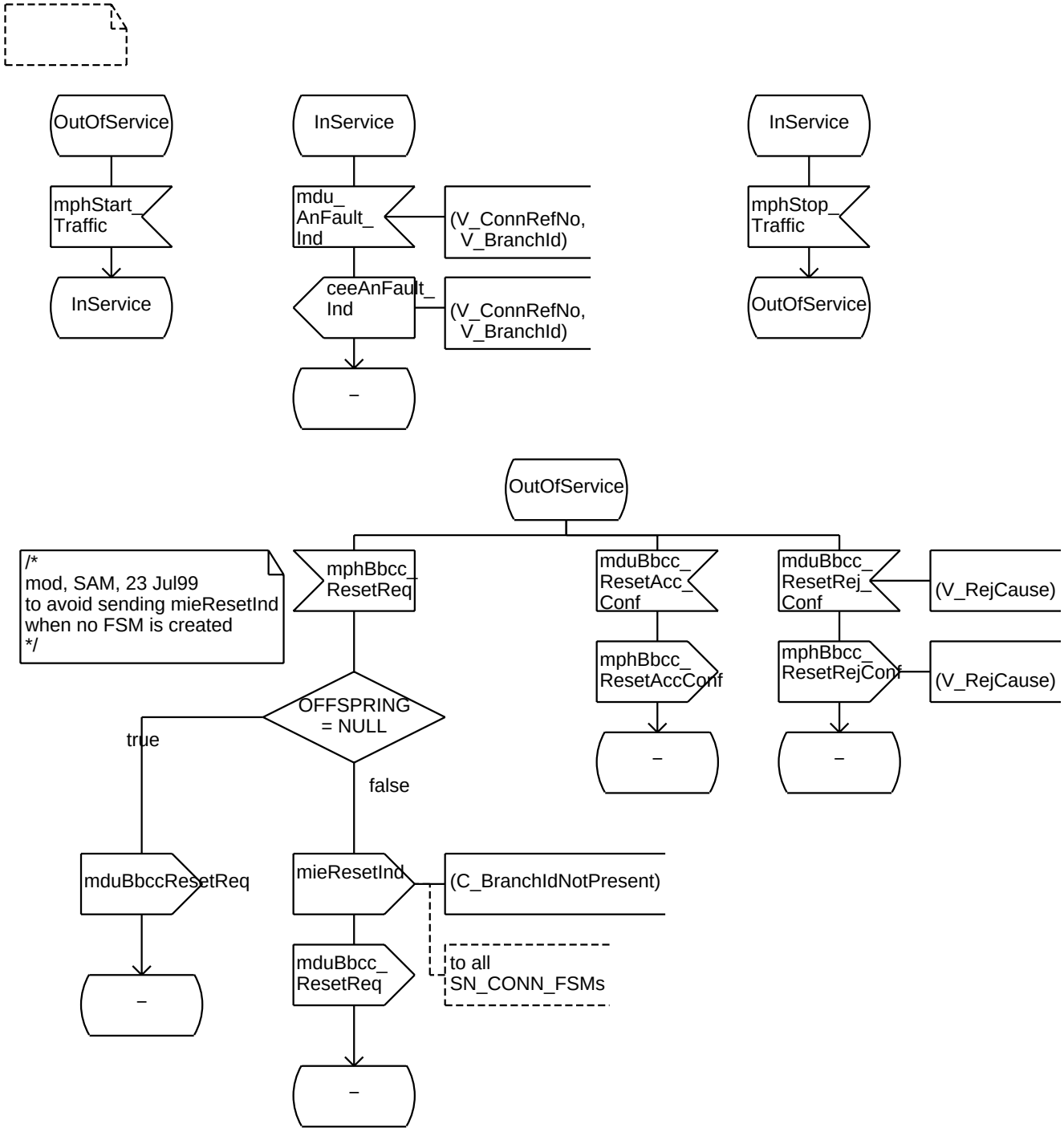
```
signallist SL_BFSM_CFSM =
mieAddBranchFSMAccConf,
mieAddBranchFSMRejConf,
mieUpdateBranchFSMAccConf,
mieUpdateBranchFSMRejConf,
mieDropBranchFSMAccConf,
mieDropBranchFSMRejConf;
```

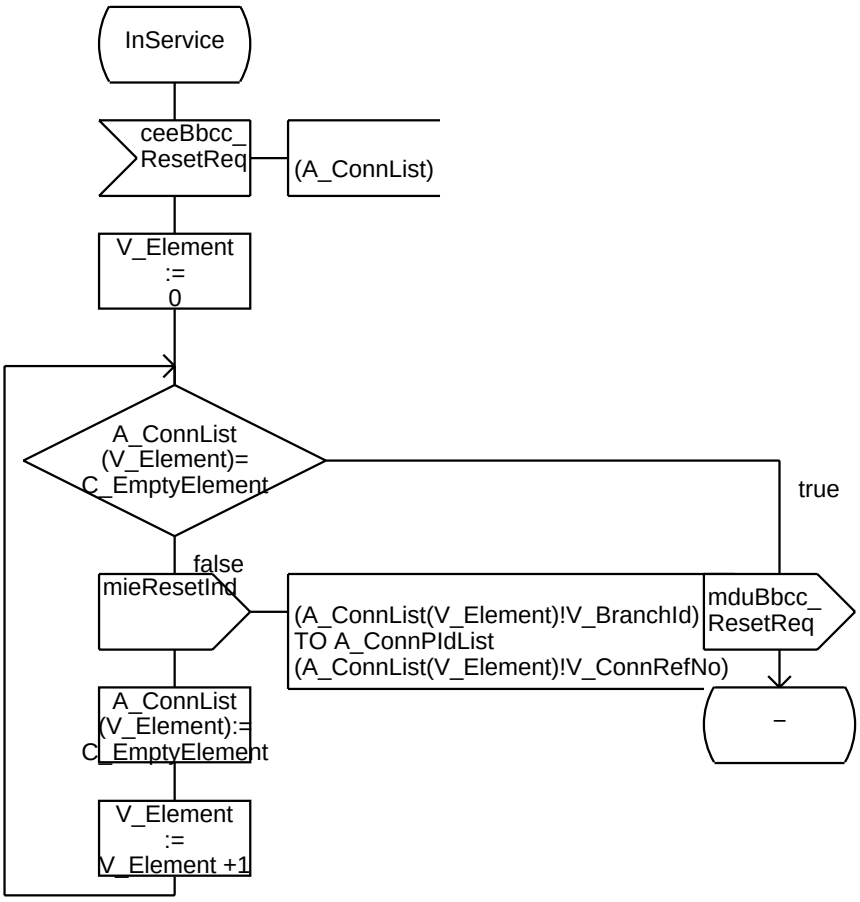
```
/* Data type definitions internal to the process */
/**/
/* This data type is used for translation of the connection reference number to the process identifier (PID)
   of the connection FSM */
NEWTYPE AT_ConnPIdList ARRAY
(IT_ConnRefNo,PID)
ENDNEWTYPE AT_ConnPIdList;
```

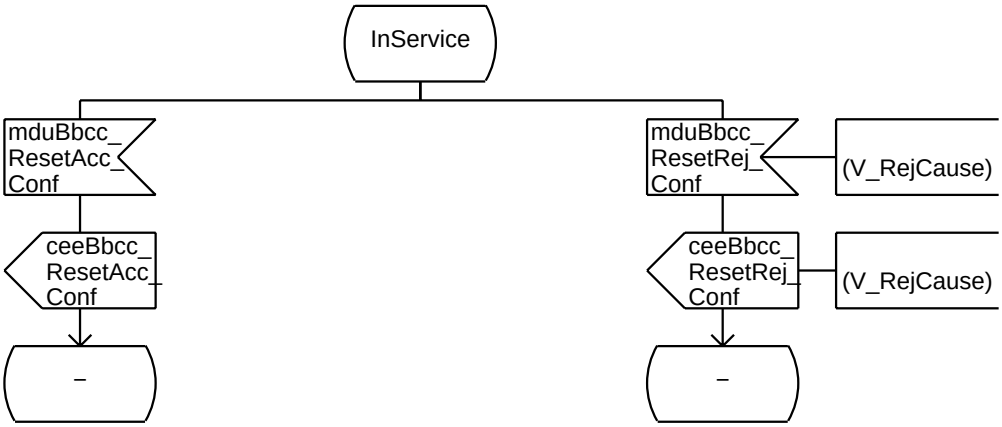
```
/*Primitive Data Declarations*/
/**/
/* List of connections */
DCL
A_ConnList  AT_ConnList;
/**/
/* Connection reference number */
DCL
V_ConnRefNo  IT_ConnRefNo;
/**/
/* Branch identifier */
DCL
V_BranchId  IT_BranchId;
/**/
/* Reject Cause */
DCL
V_RejCause  IT_RejCause;
```

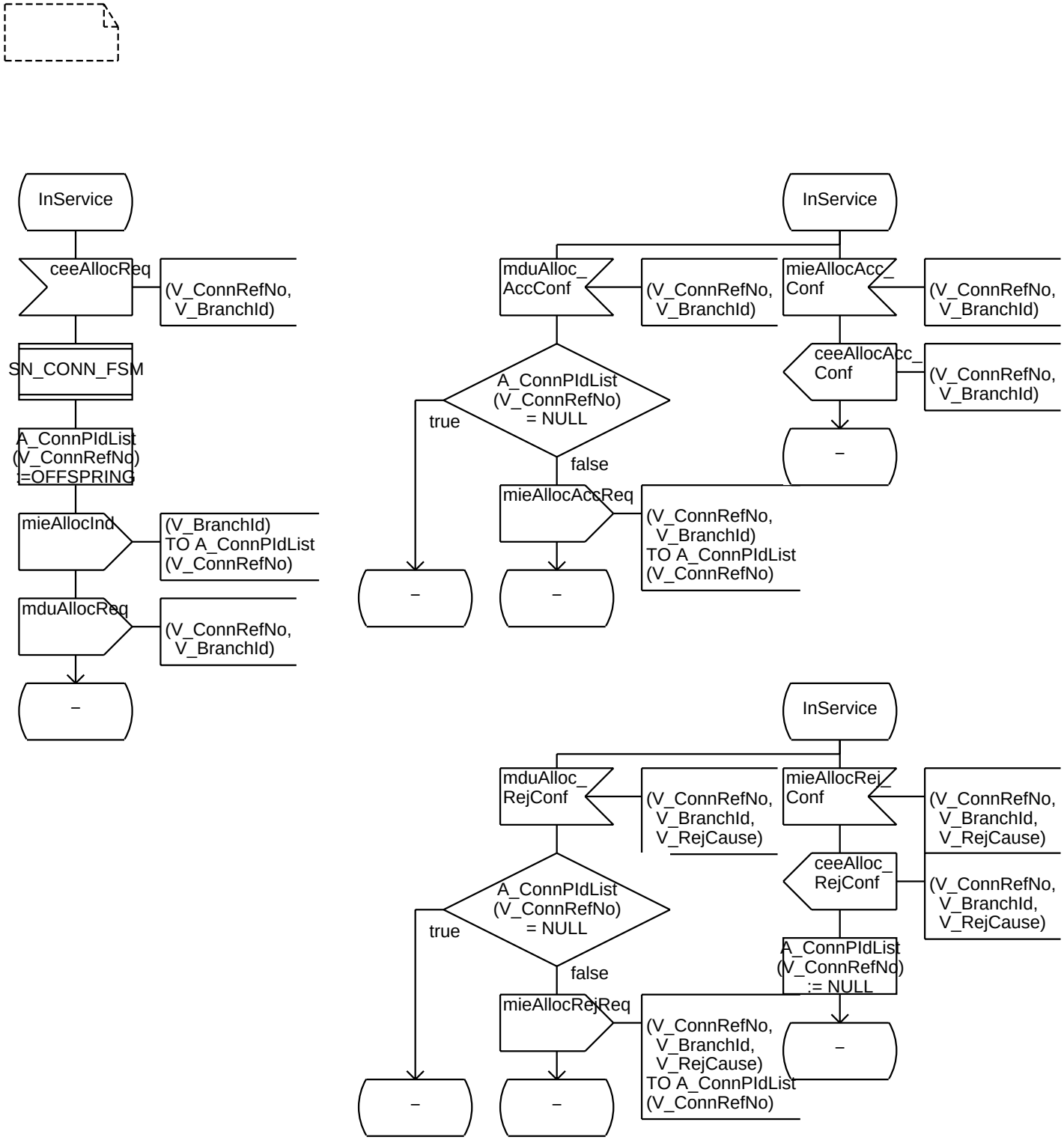
```
/* Process Data Declarations */
/**/
/* List of connection FSMs */
DCL
A_ConnPIdList   AT_ConnPIdList;
/**/
/* Index of A_ConnList */
DCL
V_Element     INTEGER;
```

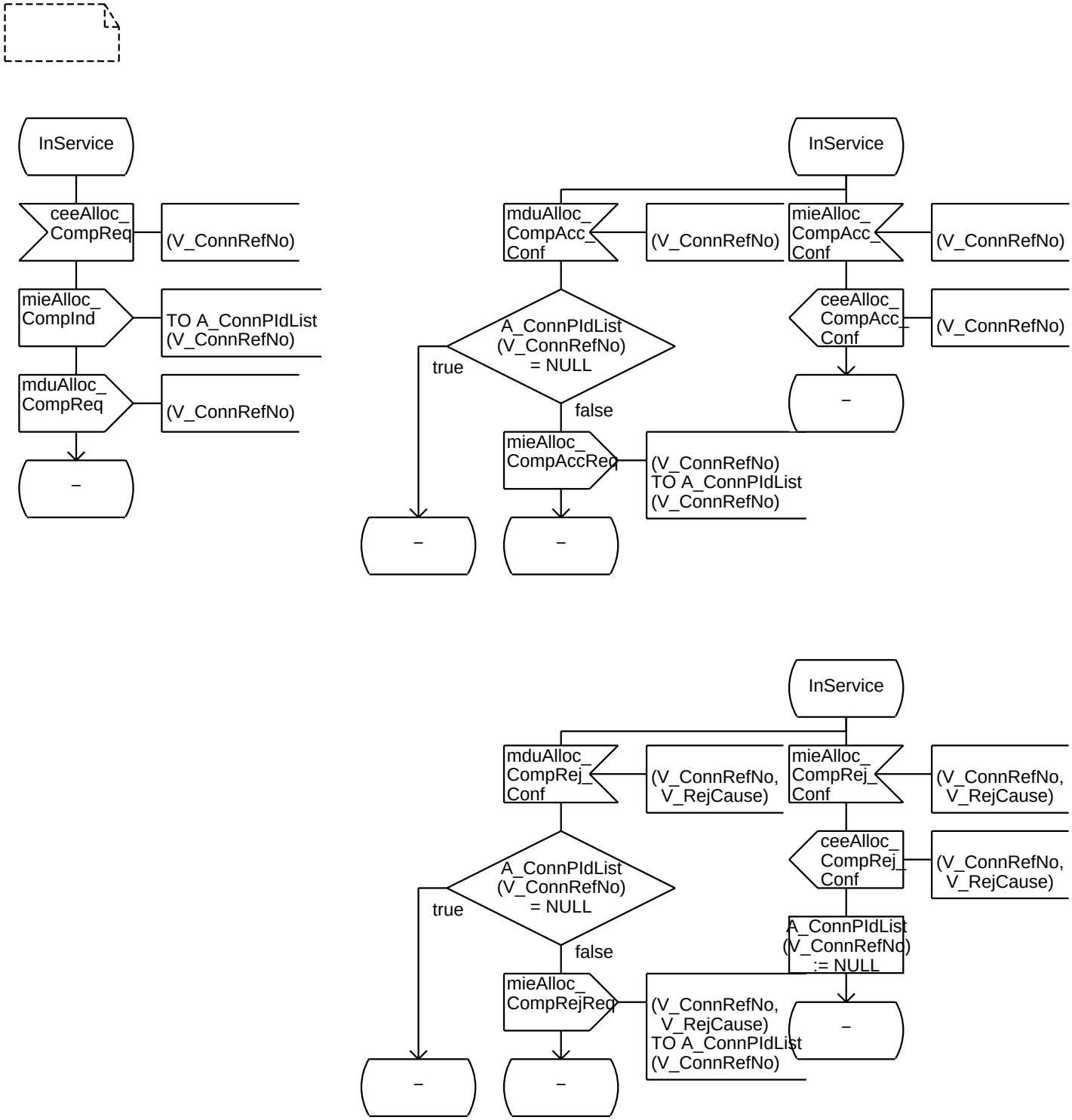


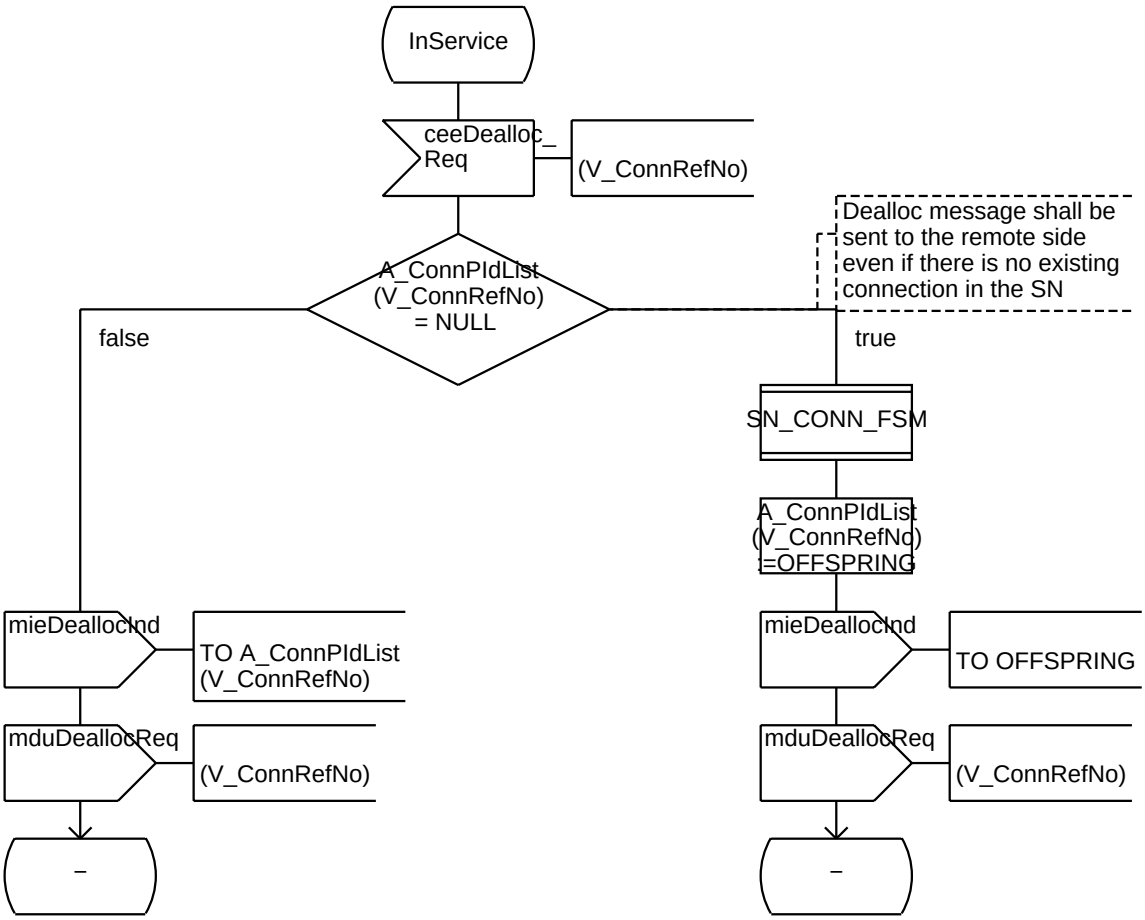

OutOfService
mphStart_
Traffic
InService
InService
mdu_
AnFault_
Ind
(V_ConnRefNo,
V_BranchId)
ceeAnFault_
Ind
(V_ConnRefNo,
V_BranchId)
–
InService
mphStop_
Traffic
OutOfService
OutOfService
/*
mod, SAM, 23 Jul99
to avoid sending mieResetInd
when no FSM is created
*/
mphBbcc_
ResetReq
mduBbcc_
ResetAcc_
Conf
mduBbcc_
ResetRej_
Conf
(V_RejCause)
mphBbcc_
ResetAccConf
mphBbcc_
ResetRejConf
(V_RejCause)
–
–
OFFSPRING
= NULL
true
false
mduBbccResetReq
mieResetInd
(C_BranchIdNotPresent)
mduBbcc_
ResetReq
to all
SN_CONN_FSMs
–
–

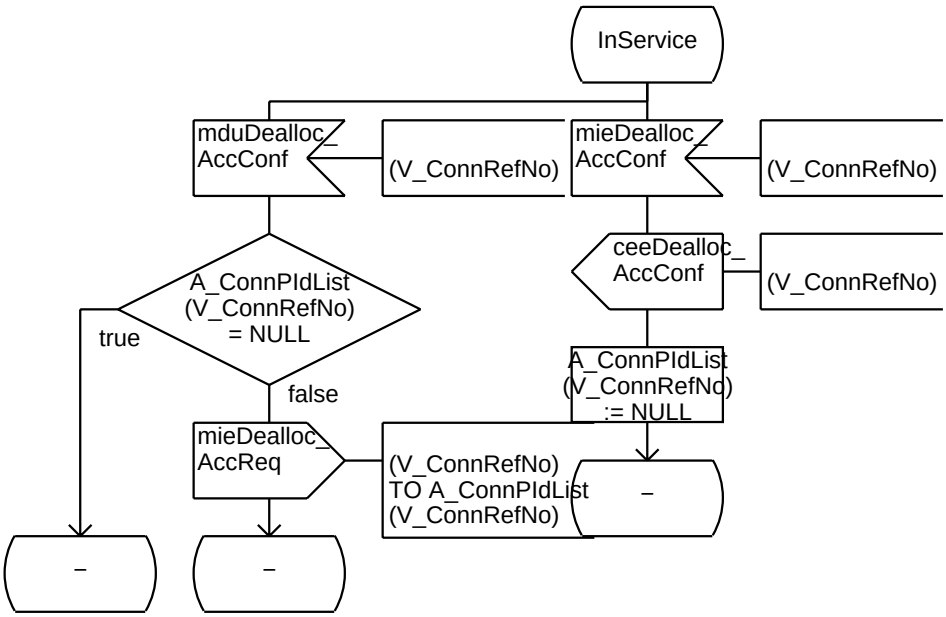

InService
ceeBbcc_
ResetReq
(A_ConnList)
V_Element
:=
0
A_ConnList
(V_Element)=
C_EmptyElement
true
false
mieResetInd
(A_ConnList(V_Element)!V_BranchId)
TO A_ConnPIdList
(A_ConnList(V_Element)!V_ConnRefNo)
mduBbcc_
ResetReq
–
A_ConnList
(V_Element):=
C_EmptyElement
V_Element
:=
V_Element +1

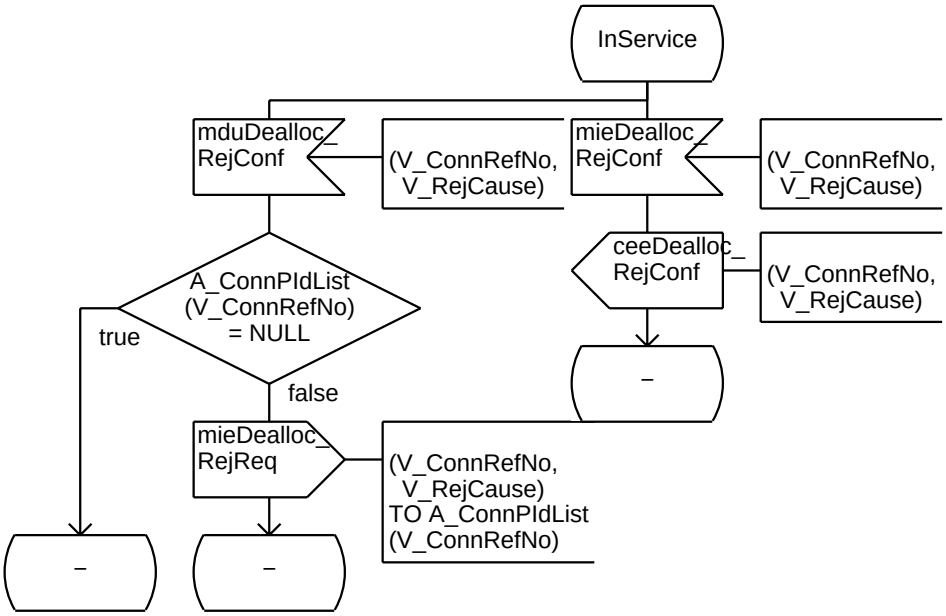

InService
mduBbcc_
ResetAcc_
Conf
mduBbcc_
ResetRej_
Conf
(V_RejCause)
ceeBbcc_
ResetAcc_
Conf
ceeBbcc_
ResetRej_
Conf
(V_RejCause)
–
–

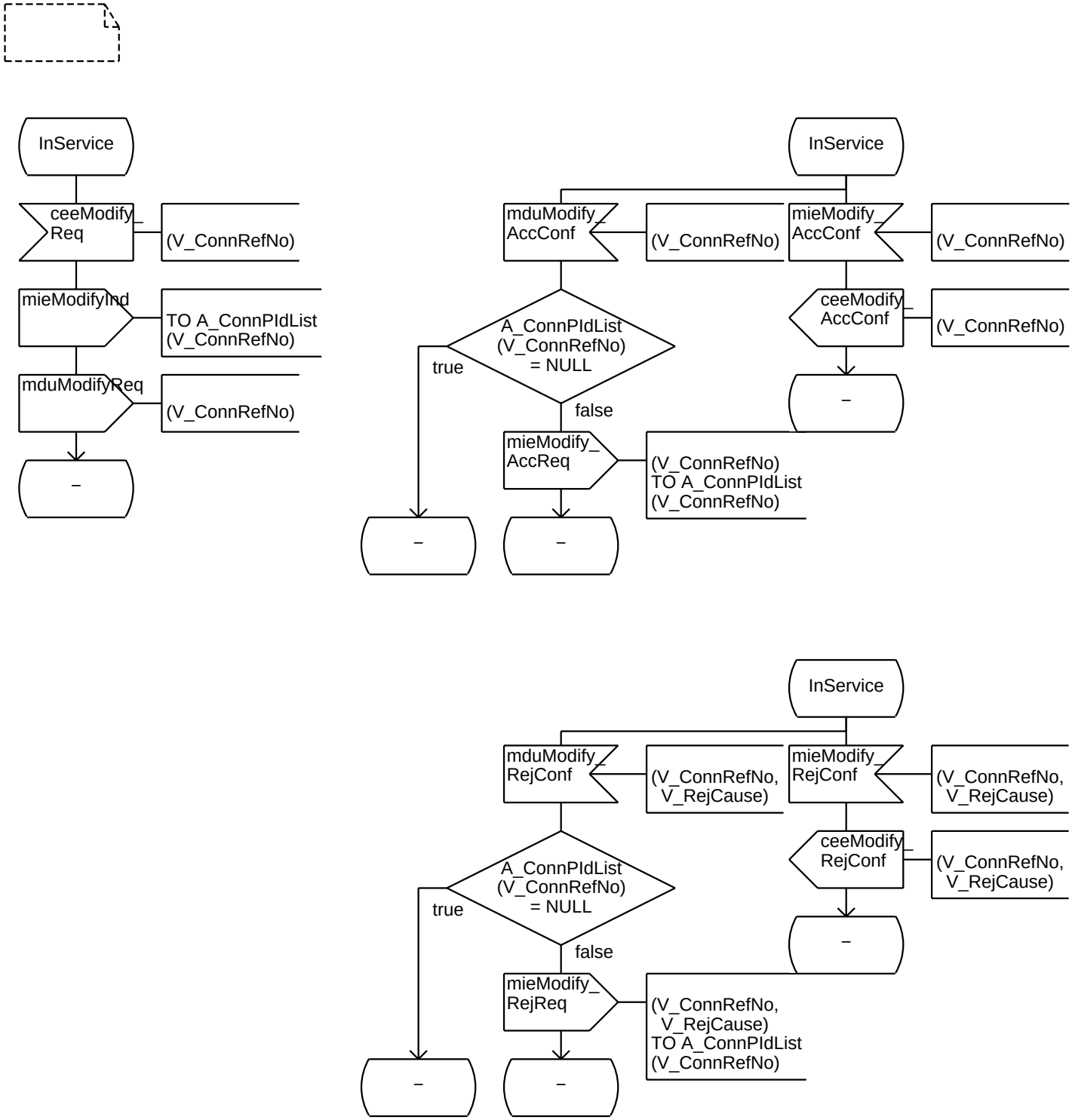

InService
ceeAllocReq
(V_ConnRefNo, V_BranchId)
SN_CONN_FSM
A_ConnPIdList (V_ConnRefNo) !=OFFSPRING
mieAllocInd
(V_BranchId) TO A_ConnPIdList (V_ConnRefNo)
mduAllocReq
(V_ConnRefNo, V_BranchId)
–
InService
mduAlloc_ AccConf
(V_ConnRefNo, V_BranchId)
mieAllocAcc_ Conf
(V_ConnRefNo, V_BranchId)
ceeAllocAcc_ Conf
(V_ConnRefNo, V_BranchId)
A_ConnPIdList (V_ConnRefNo) = NULL
true
false
mieAllocAccReq
(V_ConnRefNo, V_BranchId) TO A_ConnPIdList (V_ConnRefNo)
–
–
–
InService
mduAlloc_ RejConf
(V_ConnRefNo, V_BranchId, V_RejCause)
mieAllocRej_ Conf
(V_ConnRefNo, V_BranchId, V_RejCause)
ceeAlloc_ RejConf
(V_ConnRefNo, V_BranchId, V_RejCause)
A_ConnPIdList (V_ConnRefNo) = NULL
true
false
A_ConnPIdList (V_ConnRefNo) := NULL
mieAllocRejReq
(V_ConnRefNo, V_BranchId, V_RejCause) TO A_ConnPIdList (V_ConnRefNo)
–
–
–

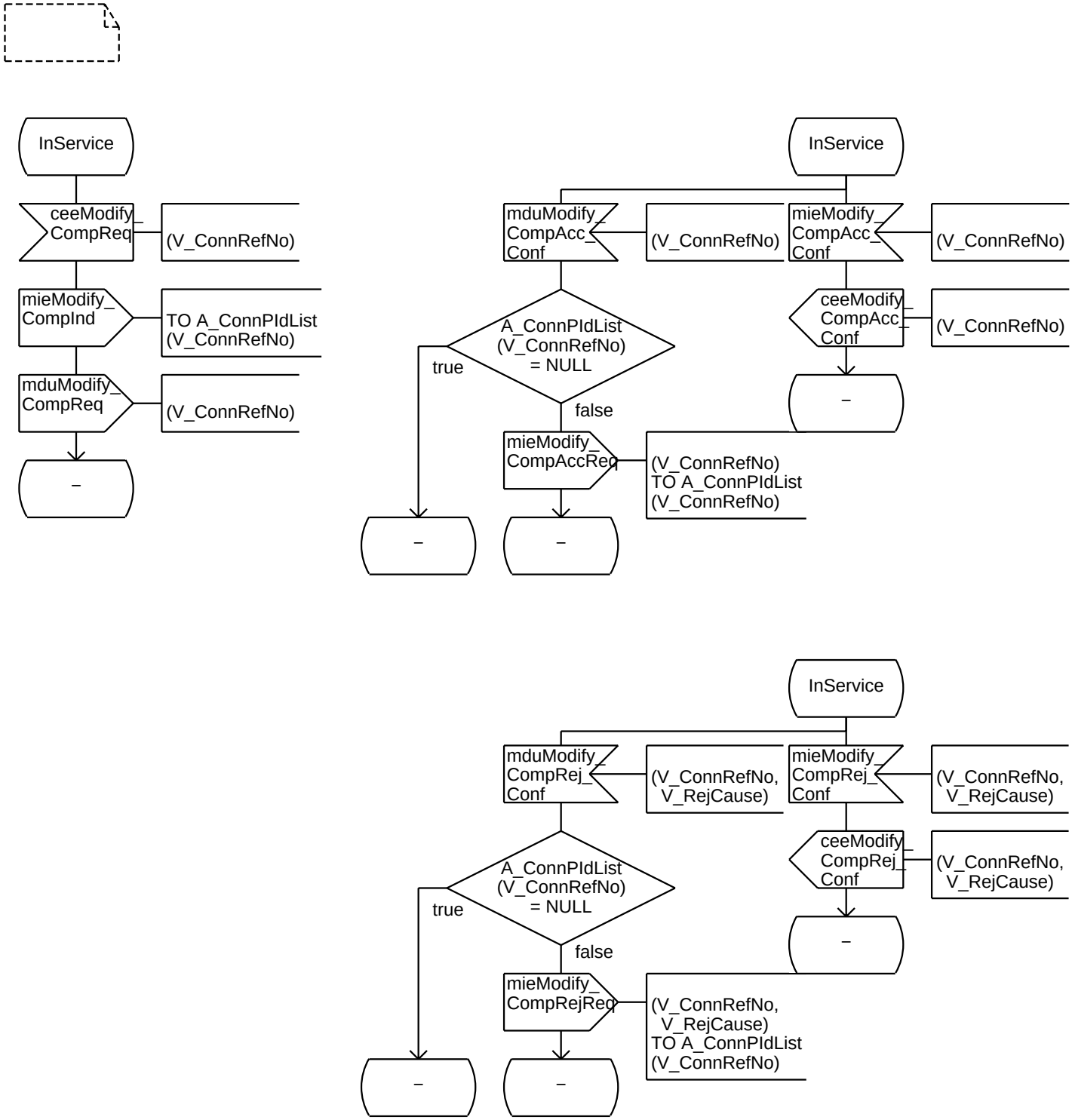

InService
ceeAlloc_
CompReq
(V_ConnRefNo)
mieAlloc_
CompInd
TO A_ConnPIdList
(V_ConnRefNo)
mduAlloc_
CompReq
(V_ConnRefNo)
–
InService
mduAlloc_
CompAcc_
Conf
(V_ConnRefNo)
mieAlloc_
CompAcc_
Conf
(V_ConnRefNo)
A_ConnPIdList
(V_ConnRefNo)
= NULL
true
false
ceeAlloc_
CompAcc_
Conf
(V_ConnRefNo)
–
mieAlloc_
CompAccReq
(V_ConnRefNo)
TO A_ConnPIdList
(V_ConnRefNo)
–
–
InService
mduAlloc_
CompRej_
Conf
(V_ConnRefNo,
V_RejCause)
mieAlloc_
CompRej_
Conf
(V_ConnRefNo,
V_RejCause)
A_ConnPIdList
(V_ConnRefNo)
= NULL
true
false
ceeAlloc_
CompRej_
Conf
(V_ConnRefNo,
V_RejCause)
A_ConnPIdList
(V_ConnRefNo)
:= NULL
mieAlloc_
CompRejReq
(V_ConnRefNo,
V_RejCause)
TO A_ConnPIdList
(V_ConnRefNo)
–
–
–

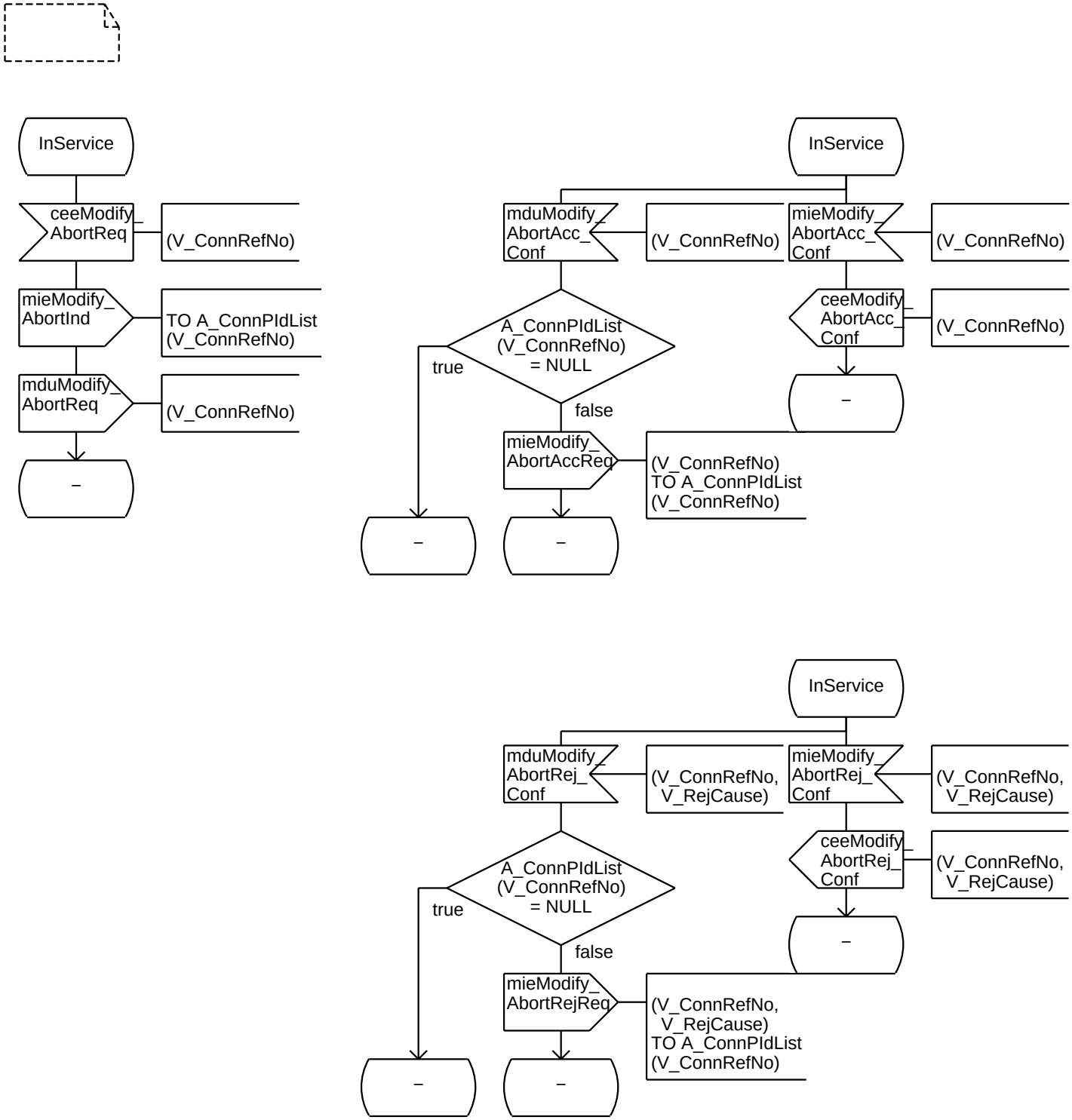

InService
ceeDealloc_
Req
(V_ConnRefNo)
Dealloc message shall be
sent to the remote side
even if there is no existing
connection in the SN
A_ConnPIdList
(V_ConnRefNo)
= NULL
false
true
SN_CONN_FSM
A_ConnPIdList
(V_ConnRefNo)
!=OFFSPRING
mieDeallocInd
TO A_ConnPIdList
(V_ConnRefNo)
mieDeallocInd
TO OFFSPRING
mduDeallocReq
(V_ConnRefNo)
mduDeallocReq
(V_ConnRefNo)
–
–

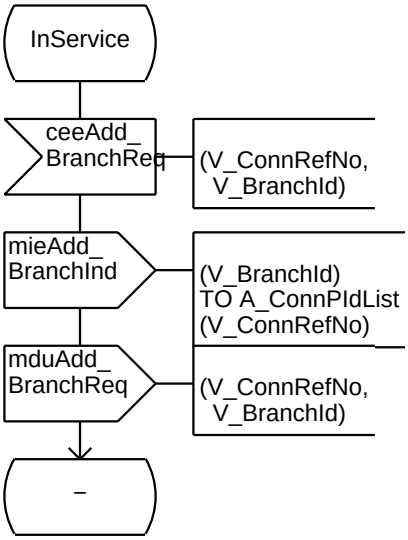

InService
mduDealloc_
AccConf
(V_ConnRefNo)
mieDealloc_
AccConf
(V_ConnRefNo)
ceeDealloc_
AccConf
(V_ConnRefNo)
A_ConnPIdList
(V_ConnRefNo)
= NULL
true
false
A_ConnPIdList
(V_ConnRefNo)
:= NULL
mieDealloc_
AccReq
(V_ConnRefNo)
TO A_ConnPIdList
(V_ConnRefNo)
–
–
–

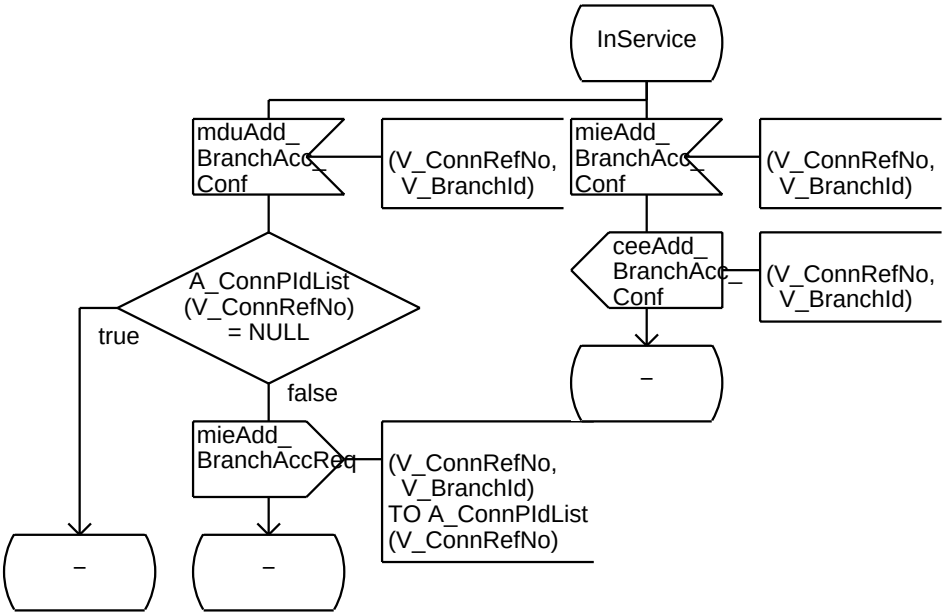

InService
mduDealloc_
RejConf
(V_ConnRefNo,
V_RejCause)
mieDealloc_
RejConf
(V_ConnRefNo,
V_RejCause)
ceeDealloc_
RejConf
(V_ConnRefNo,
V_RejCause)
A_ConnPIdList
(V_ConnRefNo)
= NULL
true
false
–
mieDealloc_
RejReq
(V_ConnRefNo,
V_RejCause)
TO A_ConnPIdList
(V_ConnRefNo)
–
–

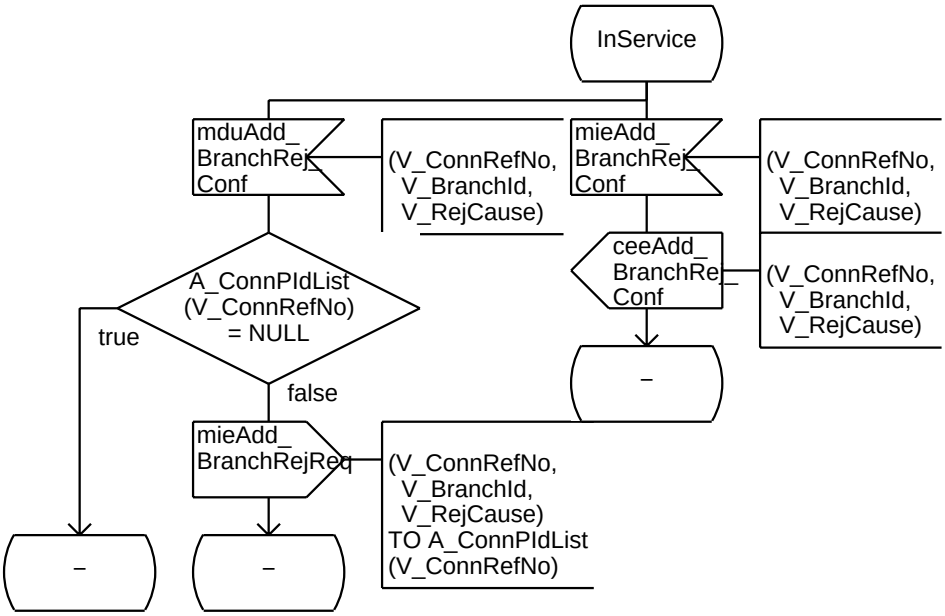

InService
ceeModify_
Req
(V_ConnRefNo)
mieModifyInd
TO A_ConnPIdList
(V_ConnRefNo)
mduModifyReq
(V_ConnRefNo)
–
InService
mduModify_
AccConf
(V_ConnRefNo)
mieModify_
AccConf
(V_ConnRefNo)
A_ConnPIdList
(V_ConnRefNo)
= NULL
true
false
ceeModify_
AccConf
(V_ConnRefNo)
–
mieModify_
AccReq
(V_ConnRefNo)
TO A_ConnPIdList
(V_ConnRefNo)
–
–
InService
mduModify_
RejConf
(V_ConnRefNo,
V_RejCause)
mieModify_
RejConf
(V_ConnRefNo,
V_RejCause)
A_ConnPIdList
(V_ConnRefNo)
= NULL
true
false
ceeModify_
RejConf
(V_ConnRefNo,
V_RejCause)
–
mieModify_
RejReq
(V_ConnRefNo,
V_RejCause)
TO A_ConnPIdList
(V_ConnRefNo)
–
–

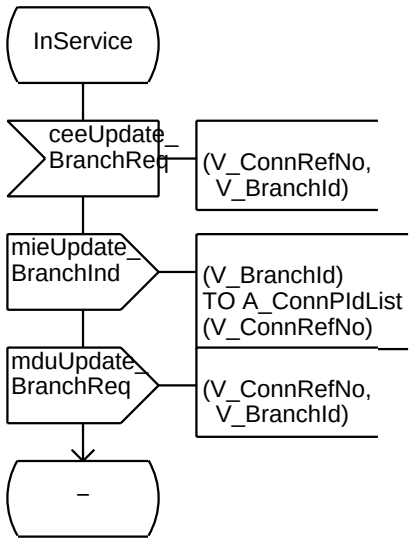

InService
ceeModify_
CompReq
(V_ConnRefNo)
mieModify_
CompInd
TO A_ConnPIdList
(V_ConnRefNo)
mduModify_
CompReq
(V_ConnRefNo)
–
InService
mduModify_
CompAcc_
Conf
(V_ConnRefNo)
mieModify_
CompAcc_
Conf
(V_ConnRefNo)
A_ConnPIdList
(V_ConnRefNo)
= NULL
true
false
ceeModify_
CompAcc_
Conf
(V_ConnRefNo)
–
mieModify_
CompAccReq
(V_ConnRefNo)
TO A_ConnPIdList
(V_ConnRefNo)
–
–
InService
mduModify_
CompRej_
Conf
(V_ConnRefNo,
V_RejCause)
mieModify_
CompRej_
Conf
(V_ConnRefNo,
V_RejCause)
A_ConnPIdList
(V_ConnRefNo)
= NULL
true
false
ceeModify_
CompRej_
Conf
(V_ConnRefNo,
V_RejCause)
–
mieModify_
CompRejReq
(V_ConnRefNo,
V_RejCause)
TO A_ConnPIdList
(V_ConnRefNo)
–
–

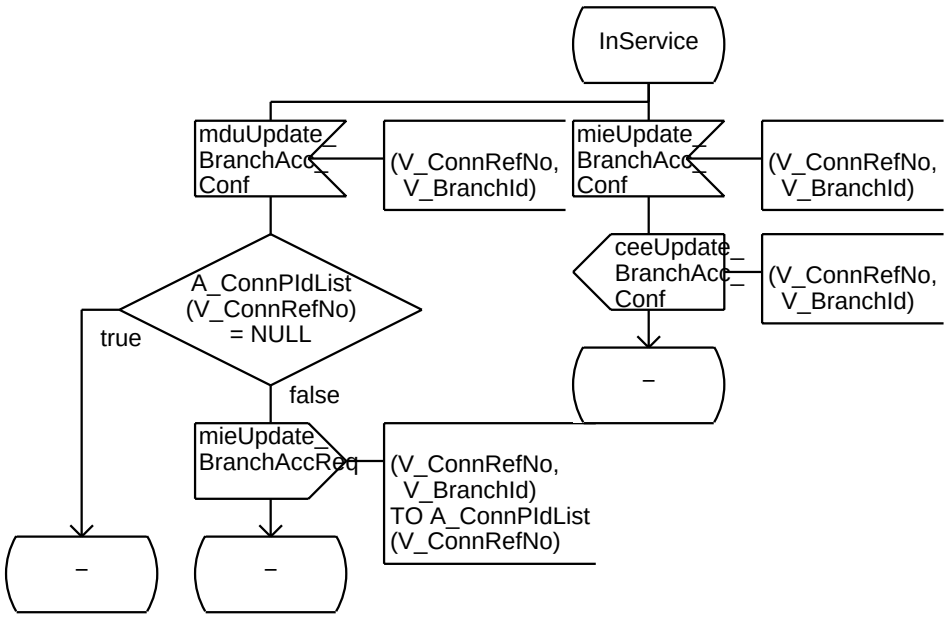

InService
ceeModify_
AbortReq
(V_ConnRefNo)
mieModify_
AbortInd
TO A_ConnPIdList
(V_ConnRefNo)
mduModify_
AbortReq
(V_ConnRefNo)
–
InService
mduModify_
AbortAcc_
Conf
(V_ConnRefNo)
mieModify_
AbortAcc_
Conf
(V_ConnRefNo)
A_ConnPIdList
(V_ConnRefNo)
= NULL
true
false
ceeModify_
AbortAcc_
Conf
(V_ConnRefNo)
–
mieModify_
AbortAccReq
(V_ConnRefNo)
TO A_ConnPIdList
(V_ConnRefNo)
–
–
InService
mduModify_
AbortRej_
Conf
(V_ConnRefNo,
V_RejCause)
mieModify_
AbortRej_
Conf
(V_ConnRefNo,
V_RejCause)
A_ConnPIdList
(V_ConnRefNo)
= NULL
true
false
ceeModify_
AbortRej_
Conf
(V_ConnRefNo,
V_RejCause)
–
mieModify_
AbortRejReq
(V_ConnRefNo,
V_RejCause)
TO A_ConnPIdList
(V_ConnRefNo)
–
–

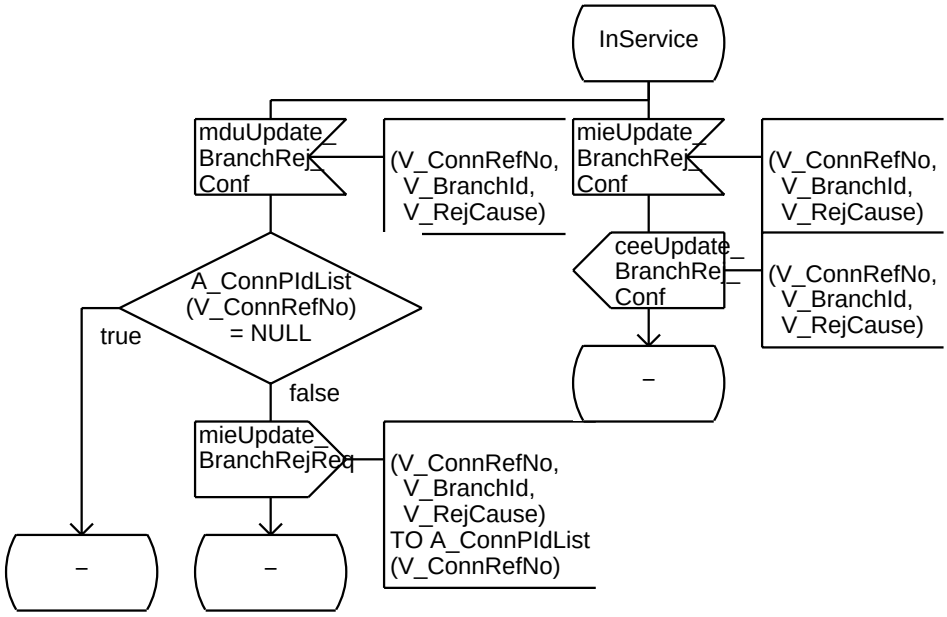

InService
ceeAdd_
BranchReq
(V_ConnRefNo,
V_BranchId)
mieAdd_
BranchInd
(V_BranchId)
TO A_ConnPIdList
(V_ConnRefNo)
mduAdd_
BranchReq
(V_ConnRefNo,
V_BranchId)
–

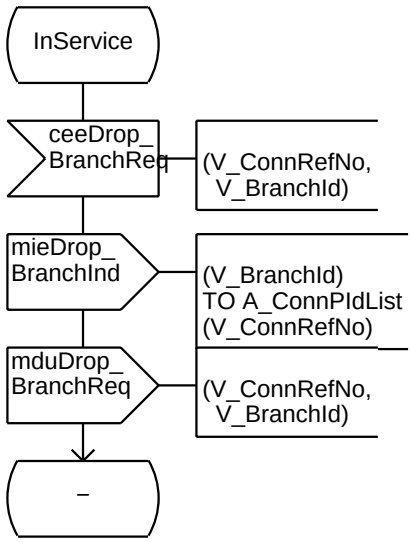

InService
mduAdd_
BranchAcc_
Conf
(V_ConnRefNo,
V_BranchId)
mieAdd_
BranchAcc_
Conf
(V_ConnRefNo,
V_BranchId)
ceeAdd_
BranchAcc_
Conf
(V_ConnRefNo,
V_BranchId)
–
A_ConnPIdList
(V_ConnRefNo)
= NULL
true
false
mieAdd_
BranchAccReq
(V_ConnRefNo,
V_BranchId)
TO A_ConnPIdList
(V_ConnRefNo)
–
–

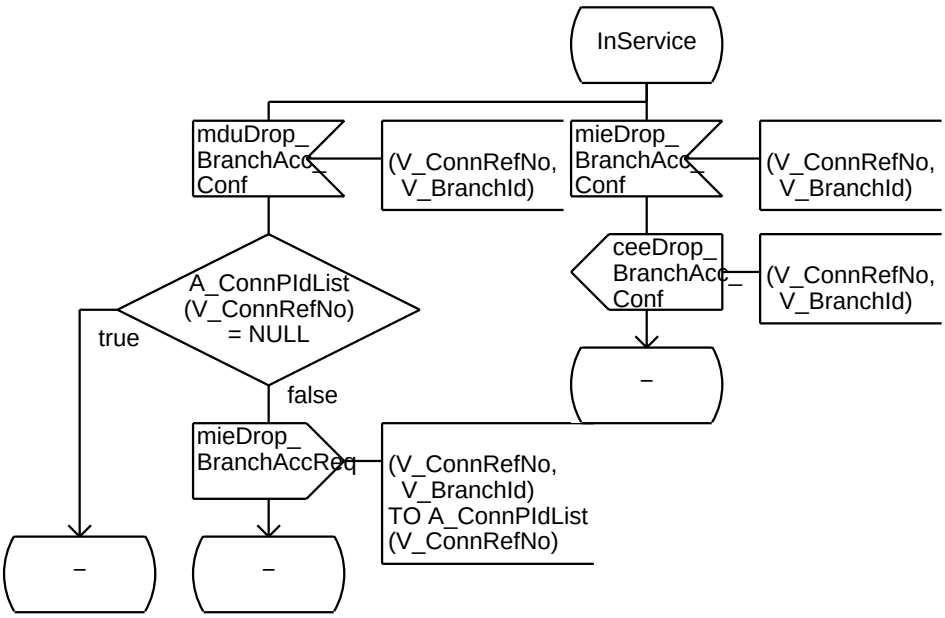

InService
mduAdd_
BranchRej_
Conf
(V_ConnRefNo,
V_BranchId,
V_RejCause)
mieAdd_
BranchRej_
Conf
(V_ConnRefNo,
V_BranchId,
V_RejCause)
ceeAdd_
BranchRej_
Conf
(V_ConnRefNo,
V_BranchId,
V_RejCause)
–
A_ConnPIdList
(V_ConnRefNo)
= NULL
true
false
mieAdd_
BranchRejReq
(V_ConnRefNo,
V_BranchId,
V_RejCause)
TO A_ConnPIdList
(V_ConnRefNo)
–
–

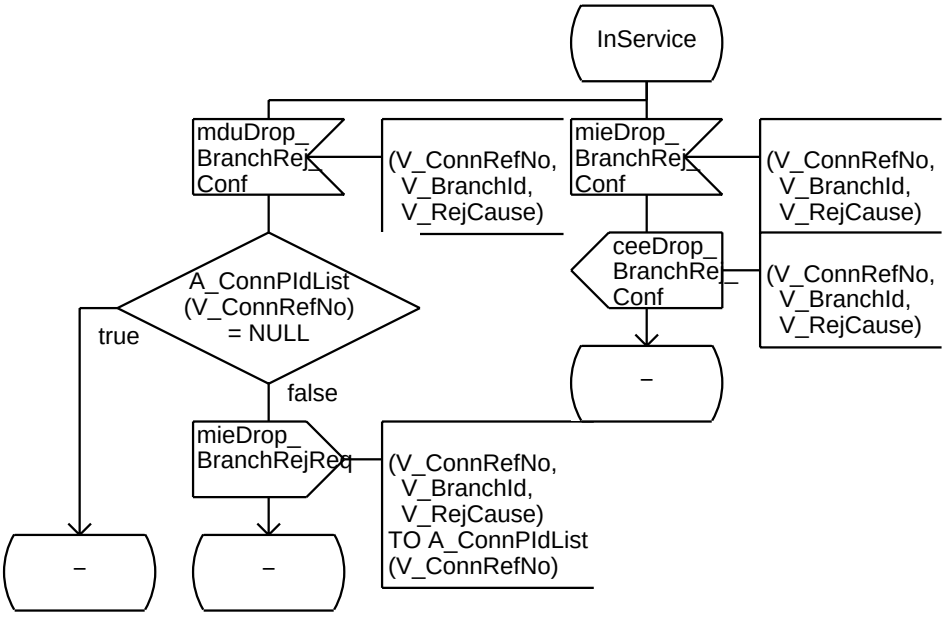

InService
ceeUpdate_
BranchReq
(V_ConnRefNo,
V_BranchId)
mieUpdate_
BranchInd
(V_BranchId)
TO A_ConnPIdList
(V_ConnRefNo)
mduUpdate_
BranchReq
(V_ConnRefNo,
V_BranchId)
–


InService
mduUpdate_
BranchAcc_
Conf
(V_ConnRefNo,
V_BranchId)
mieUpdate_
BranchAcc_
Conf
(V_ConnRefNo,
V_BranchId)
A_ConnPIdList
(V_ConnRefNo)
= NULL
true
false
ceeUpdate_
BranchAcc_
Conf
(V_ConnRefNo,
V_BranchId)
–
mieUpdate_
BranchAccReq
(V_ConnRefNo,
V_BranchId)
TO A_ConnPIdList
(V_ConnRefNo)
–
–


InService
mduUpdate_
BranchRej_
Conf
(V_ConnRefNo,
V_BranchId,
V_RejCause)
mieUpdate_
BranchRej_
Conf
(V_ConnRefNo,
V_BranchId,
V_RejCause)
A_ConnPIdList
(V_ConnRefNo)
= NULL
true
false
ceeUpdate_
BranchRej_
Conf
(V_ConnRefNo,
V_BranchId,
V_RejCause)
–
mieUpdate_
BranchRejReq
(V_ConnRefNo,
V_BranchId,
V_RejCause)
TO A_ConnPIdList
(V_ConnRefNo)
–
–


InService
ceeDrop_
BranchReq
(V_ConnRefNo,
V_BranchId)
mieDrop_
BranchInd
(V_BranchId)
TO A_ConnPIdList
(V_ConnRefNo)
mduDrop_
BranchReq
(V_ConnRefNo,
V_BranchId)
–


InService
mduDrop_
BranchAcc_
Conf
(V_ConnRefNo,
V_BranchId)
mieDrop_
BranchAcc_
Conf
(V_ConnRefNo,
V_BranchId)
A_ConnPIdList
(V_ConnRefNo)
= NULL
true
false
ceeDrop_
BranchAcc_
Conf
(V_ConnRefNo,
V_BranchId)
–
mieDrop_
BranchAccReq
(V_ConnRefNo,
V_BranchId)
TO A_ConnPIdList
(V_ConnRefNo)
–
–

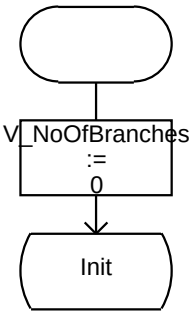

InService
mduDrop_
BranchRej_
Conf
(V_ConnRefNo,
V_BranchId,
V_RejCause)
mieDrop_
BranchRej_
Conf
(V_ConnRefNo,
V_BranchId,
V_RejCause)
A_ConnPIdList
(V_ConnRefNo)
= NULL
true
false
ceeDrop_
BranchRej_
Conf
(V_ConnRefNo,
V_BranchId,
V_RejCause)
–
mieDrop_
BranchRejReq
(V_ConnRefNo,
V_BranchId,
V_RejCause)
TO A_ConnPIdList
(V_ConnRefNo)
–
–

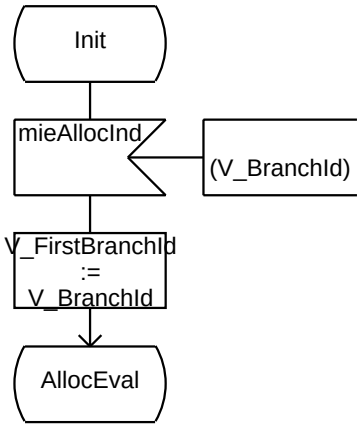
```
/* State descriptions
Init:            The process is waiting for the first message from the manager
                 indicating a ptp or ptm connection
InService:       The process is not waiting for a confirmation of a message
AllocEval:       An alloc is sent but not confirmed yet
AllocCompEval:   An alloc complete is sent but not confirmed yet
ModifyEval:      An modify is sent but not confirmed yet
ModifyCompEval:  An modify complete is sent but not confirmed yet
ModifyAbortEval: An modify abort is sent but not confirmed yet
DeallocEval:     An dealloc is sent but not confirmed yet
*/
```

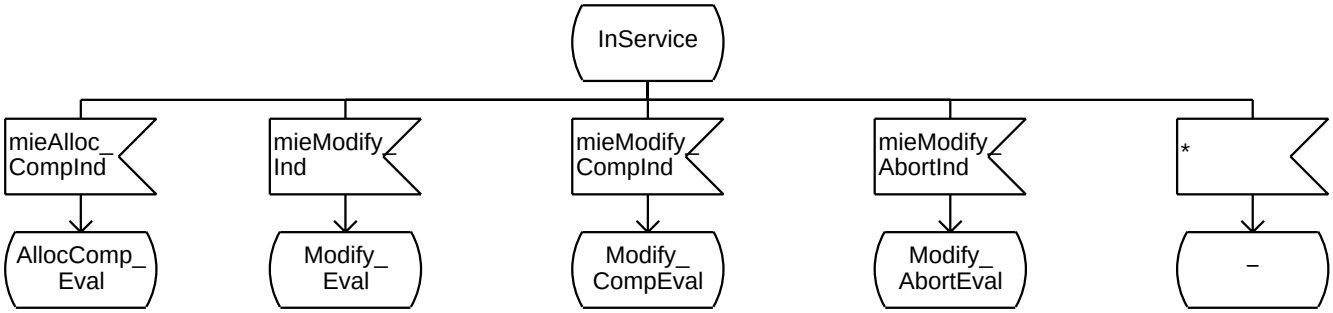
```
/* Data type definitions internal to the process */
/**/
/* This data type is used for translation of the branch identifier to the process identifier (PID)
   of the branch FSM */
NEWTYPE AT_BranchList ARRAY
(IT_BranchId,PID)
ENDNEWTYPE AT_BranchList;
```

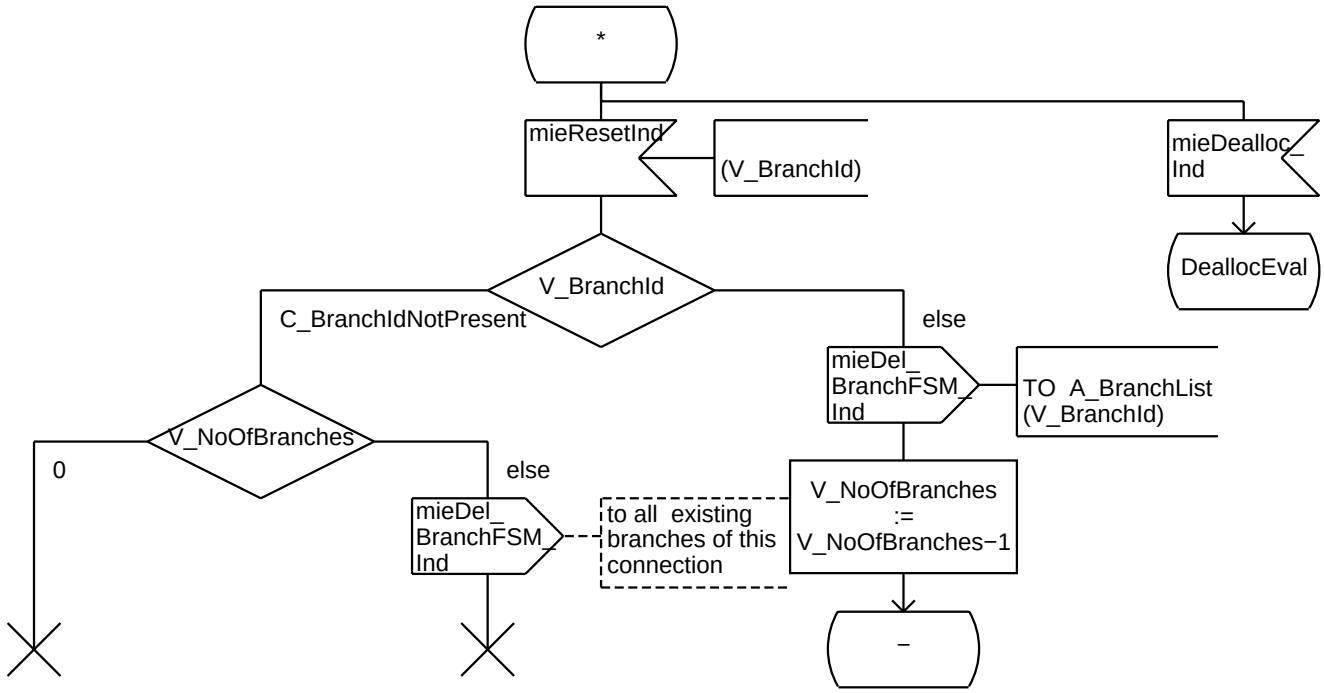
```
/*Primitive Data Declarations*/
/**/
/* Connection identifier */
DCL
V_ConnRefNo  IT_ConnRefNo;
/**/
/* Branch identifier */
DCL
V_BranchId  IT_BranchId;
/**/
/* Reject Cause */
DCL
V_RejCause   IT_RejCause;
```

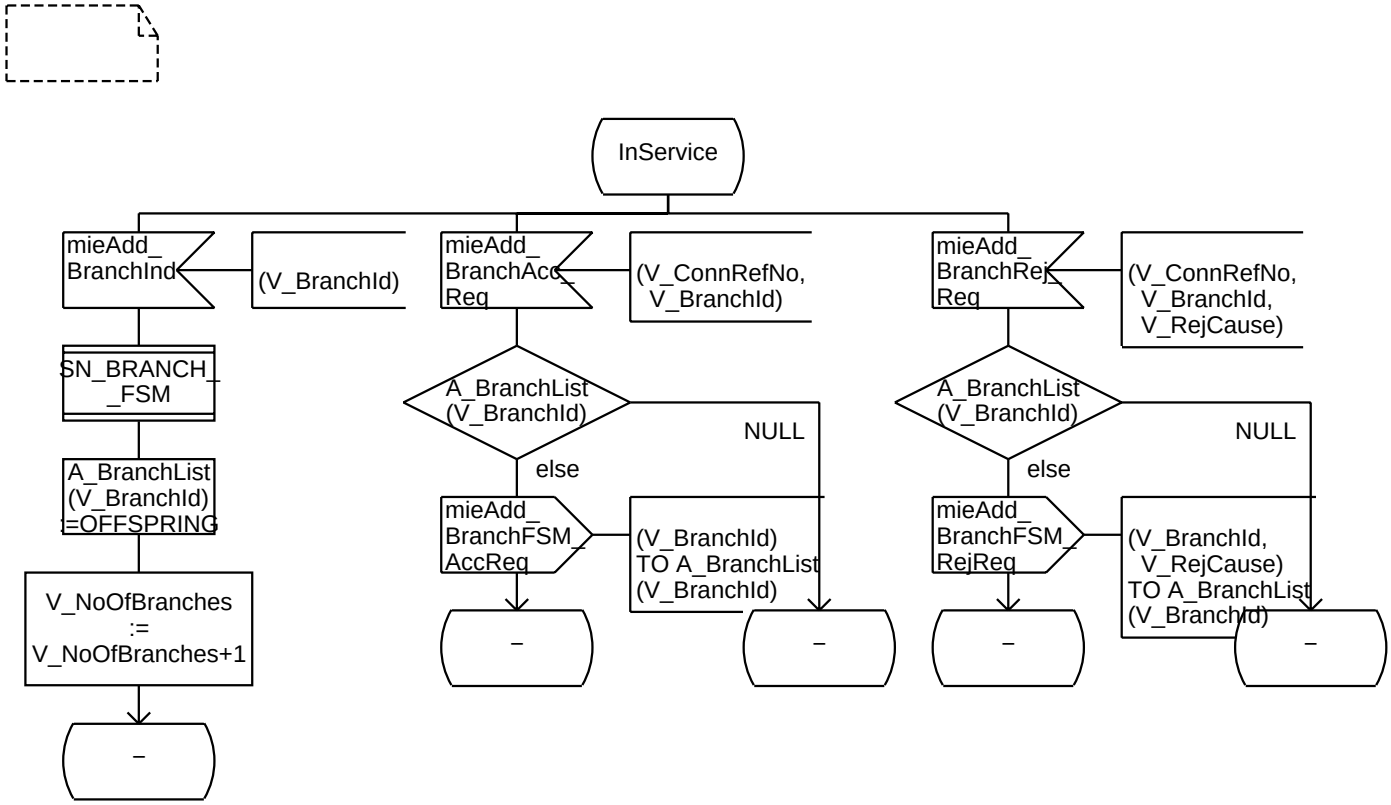
```
/* Process Data Declarations */
/**/
/* List of connection FSMs */
DCL
A_BranchList   AT_BranchList;
/**/
/* Number of branches */
DCL
V_NoOfBranches   INTEGER;
/**/
/* First branch */
DCL
V_FirstBranchId IT_BranchId;
```

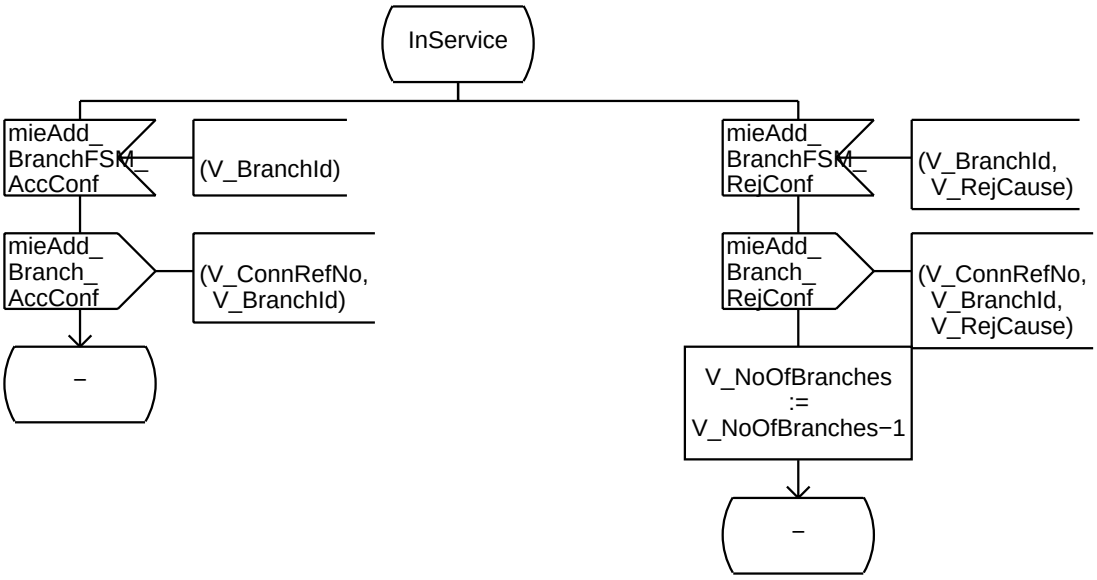

V_NoOfBranches
:=
0
Init

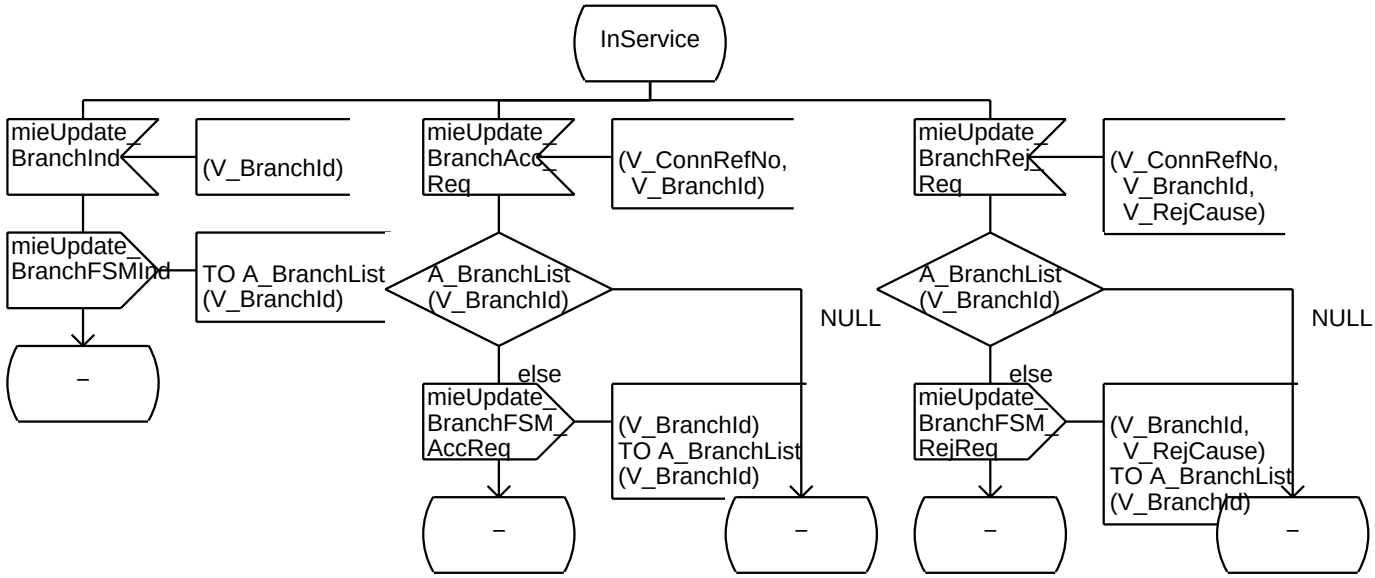

Init
mieAllocInd
(V_BranchId)
V_FirstBranchId
:=
V_BranchId
AllocEval

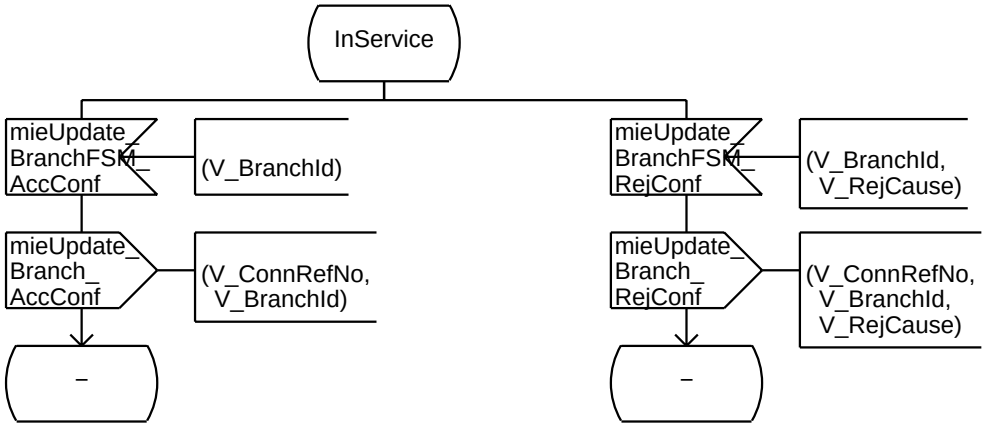

InService
mieAlloc_
CompInd
mieModify_
Ind
mieModify_
CompInd
mieModify_
AbortInd
*
AllocComp_
Eval
Modify_
Eval
Modify_
CompEval
Modify_
AbortEval
–

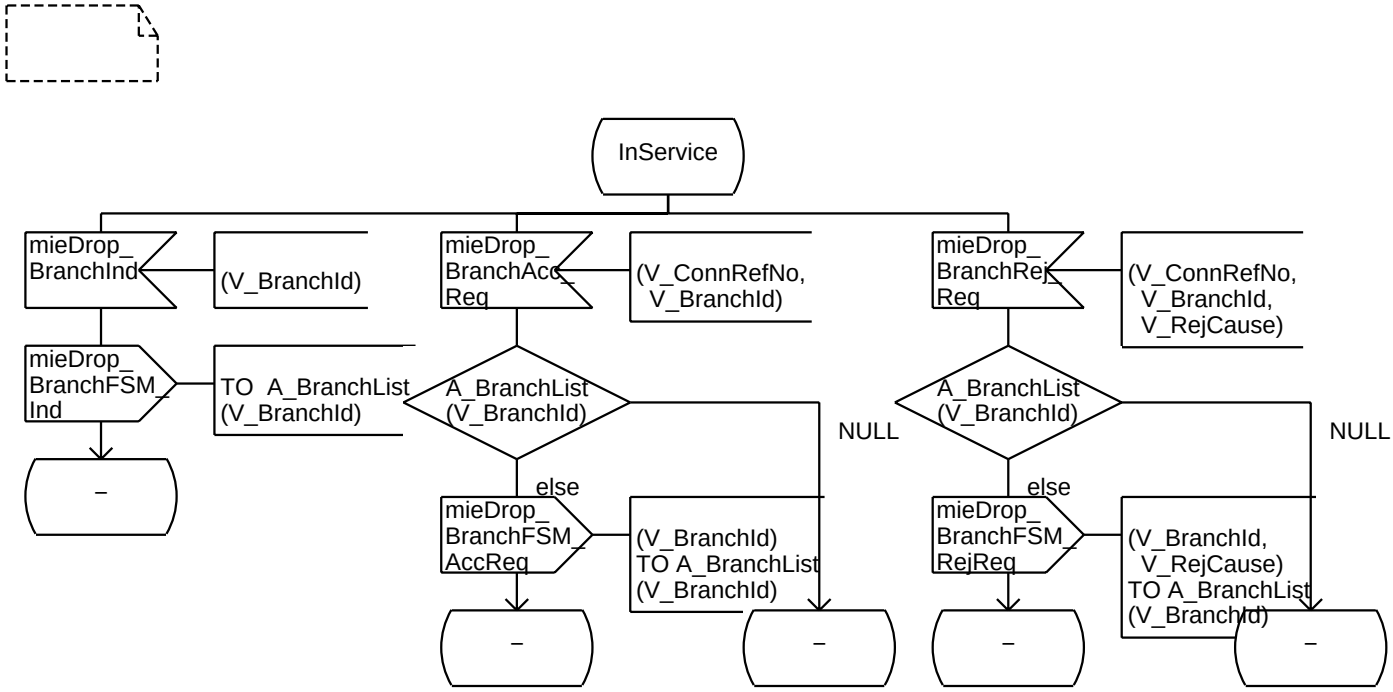

*
mieResetInd
(V_BranchId)
mieDealloc_
Ind
DeallocEval
V_BranchId
C_BranchIdNotPresent
else
mieDel_
BranchFSM_
Ind
TO A_BranchList
(V_BranchId)
V_NoOfBranches
0
else
mieDel_
BranchFSM_
Ind
to all existing
branches of this
connection
V_NoOfBranches
:=
V_NoOfBranches−1
–

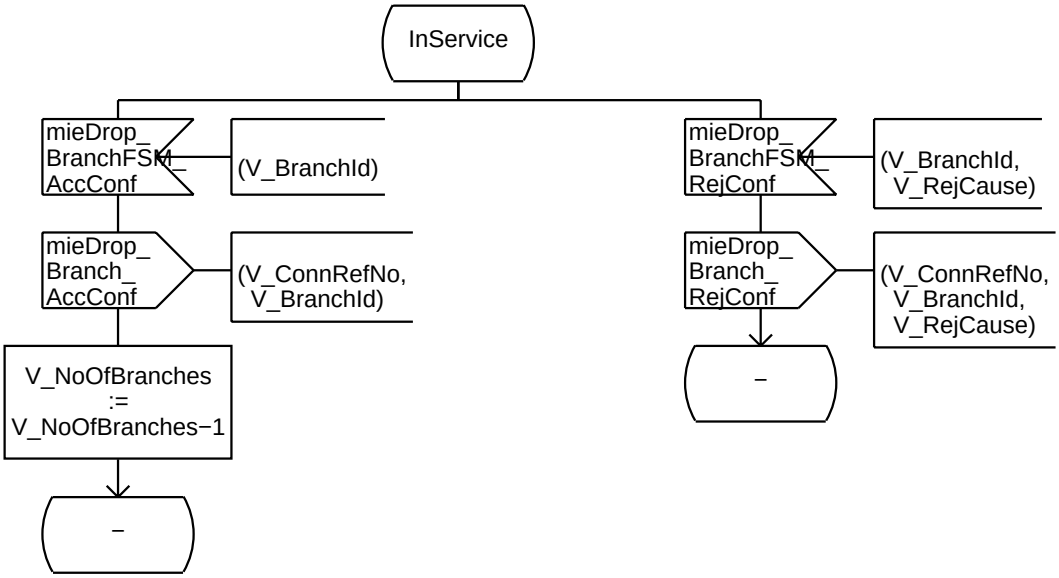

InService
mieAdd_
BranchInd
(V_BranchId)
mieAdd_
BranchAcc_
Req
(V_ConnRefNo,
V_BranchId)
mieAdd_
BranchRej_
Req
(V_ConnRefNo,
V_BranchId,
V_RejCause)
SN_BRANCH_
_FSM
A_BranchList
(V_BranchId)
NULL
else
A_BranchList
(V_BranchId)
NULL
else
A_BranchList
(V_BranchId)
:=OFFSPRING
mieAdd_
BranchFSM_
AccReq
(V_BranchId)
TO A_BranchList
(V_BranchId)
mieAdd_
BranchFSM_
RejReq
(V_BranchId,
V_RejCause)
TO A_BranchList
(V_BranchId)
V_NoOfBranches
:=
V_NoOfBranches+1
–
–
–
–
–

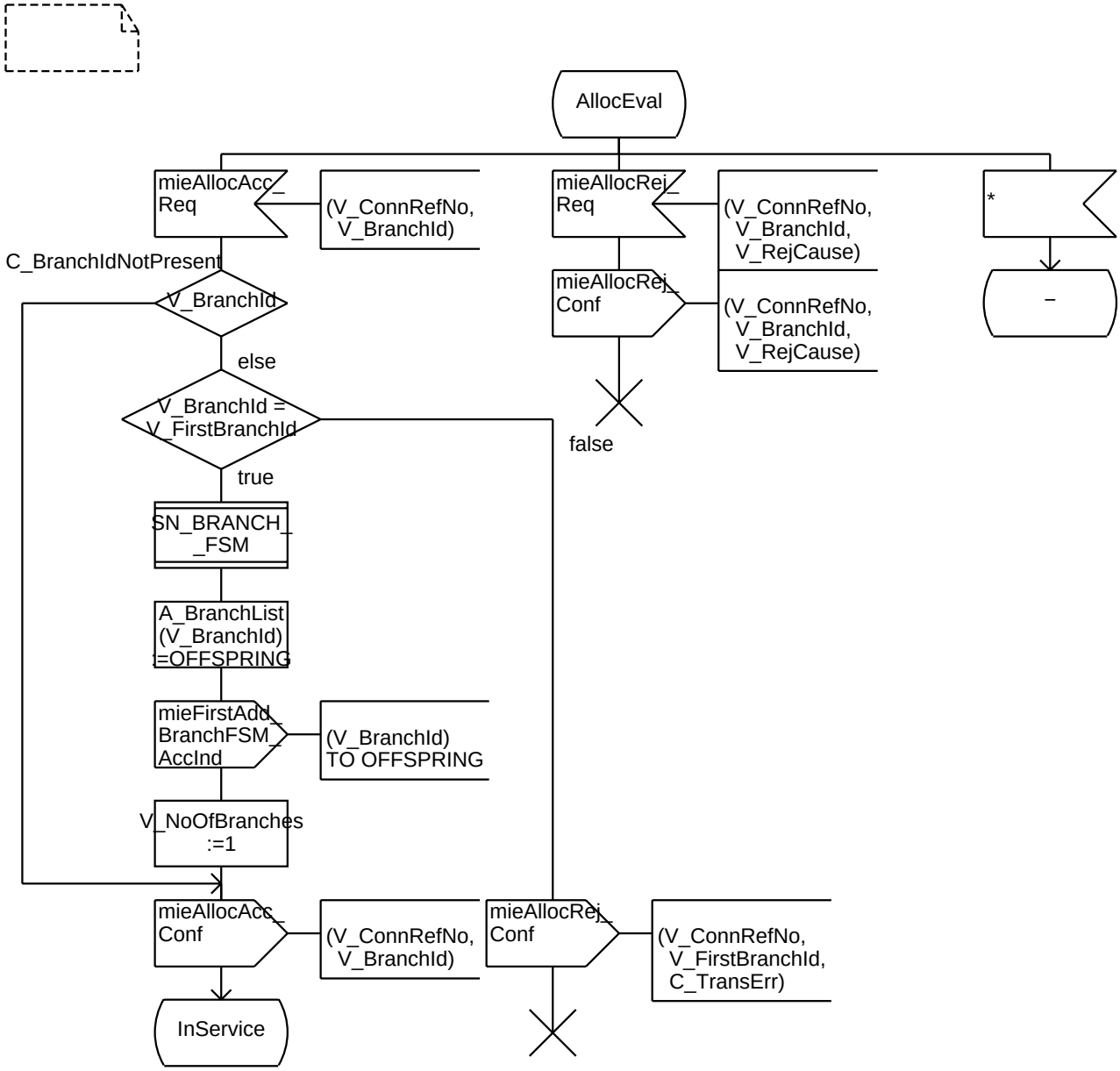

InService
mieAdd_
BranchFSM_
AccConf
(V_BranchId)
mieAdd_
BranchFSM_
RejConf
(V_BranchId,
V_RejCause)
mieAdd_
Branch_
AccConf
(V_ConnRefNo,
V_BranchId)
mieAdd_
Branch_
RejConf
(V_ConnRefNo,
V_BranchId,
V_RejCause)
–
V_NoOfBranches
:=
V_NoOfBranches−1
–

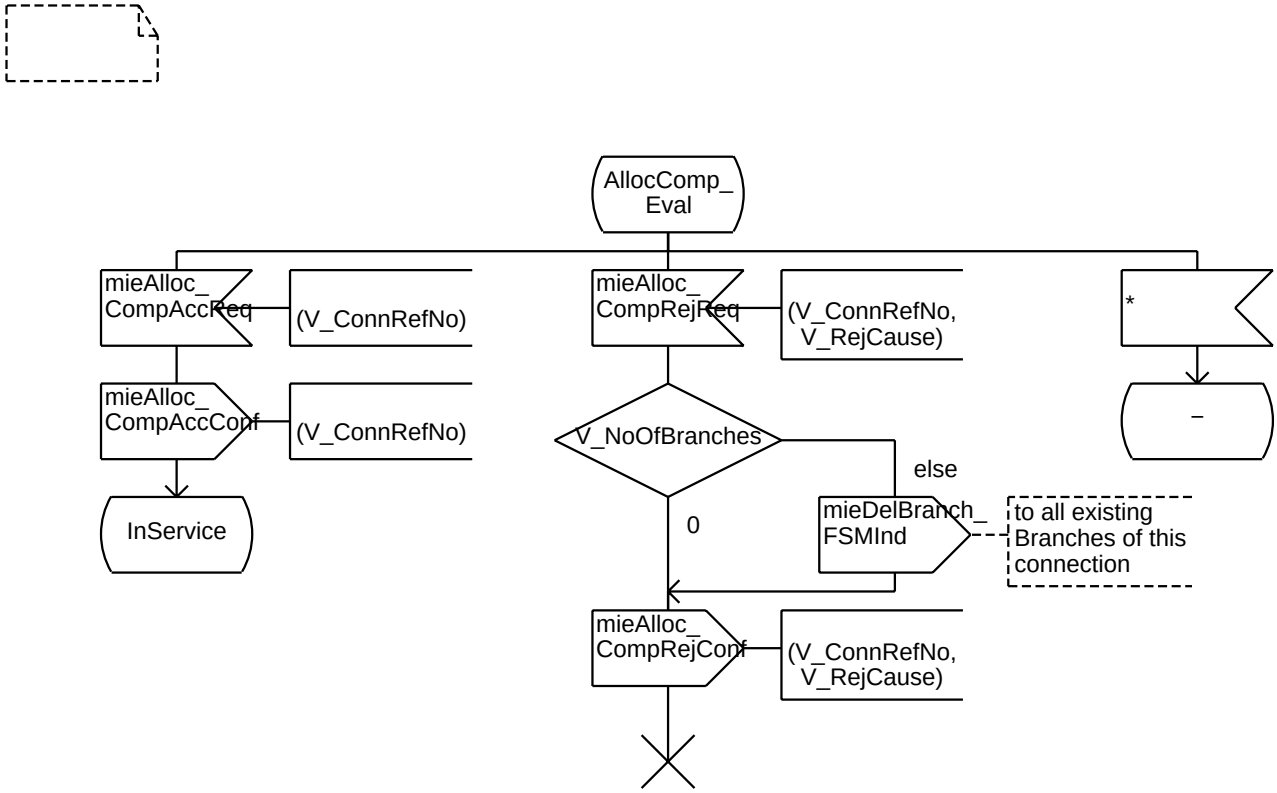

InService
mieUpdate_
BranchInd
(V_BranchId)
mieUpdate_
BranchFSMInd
TO A_BranchList
(V_BranchId)
–
mieUpdate_
BranchAcc_
Req
(V_ConnRefNo,
V_BranchId)
A_BranchList
(V_BranchId)
NULL
else
mieUpdate_
BranchFSM_
AccReq
(V_BranchId)
TO A_BranchList
(V_BranchId)
–
–
mieUpdate_
BranchRej_
Req
(V_ConnRefNo,
V_BranchId,
V_RejCause)
A_BranchList
(V_BranchId)
NULL
else
mieUpdate_
BranchFSM_
RejReq
(V_BranchId,
V_RejCause)
TO A_BranchList
(V_BranchId)
–
–

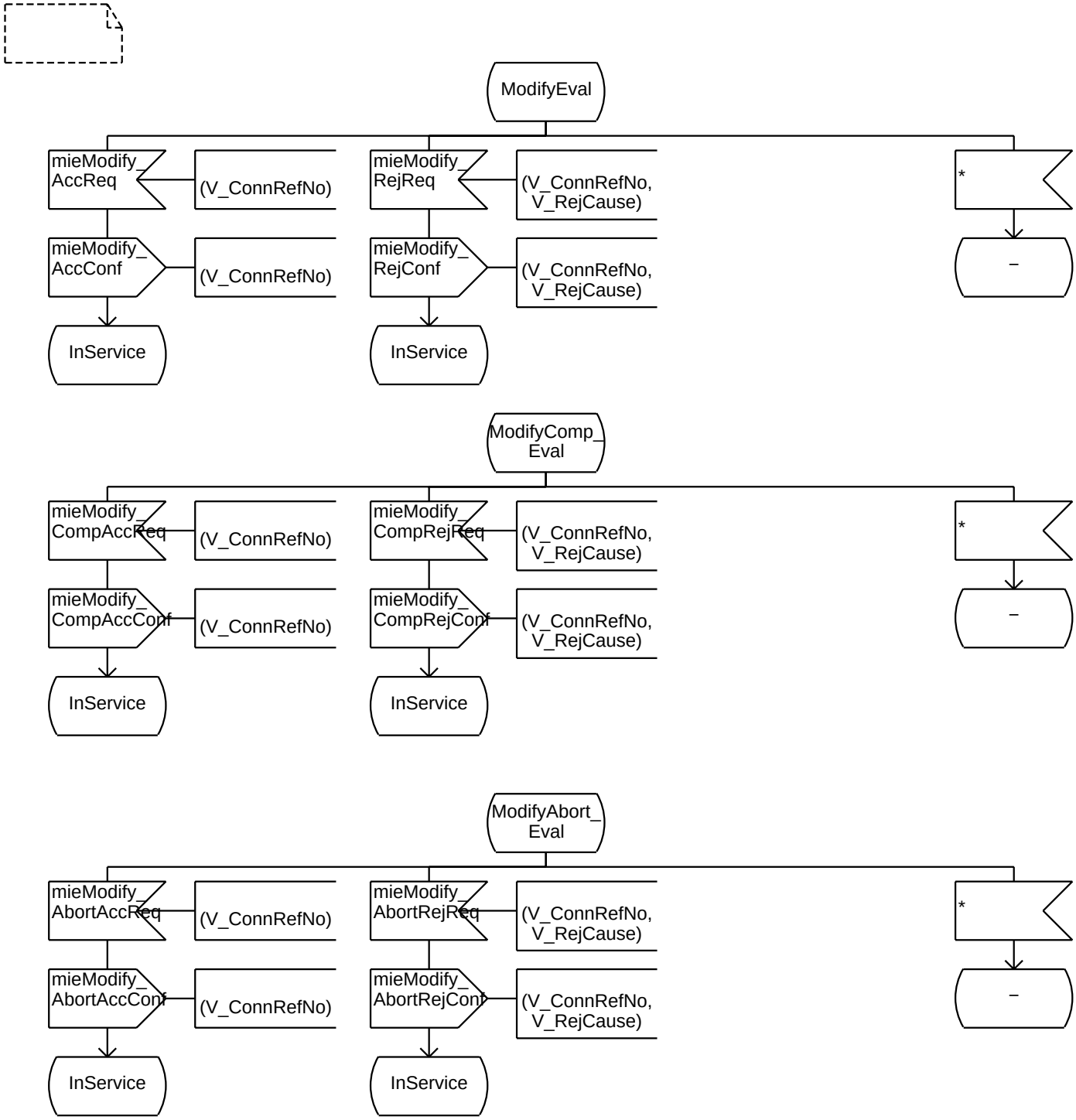

InService
mieUpdate_
BranchFSM_
AccConf
(V_BranchId)
mieUpdate_
Branch_
AccConf
(V_ConnRefNo,
V_BranchId)
–
mieUpdate_
BranchFSM_
RejConf
(V_BranchId,
V_RejCause)
mieUpdate_
Branch_
RejConf
(V_ConnRefNo,
V_BranchId,
V_RejCause)
–

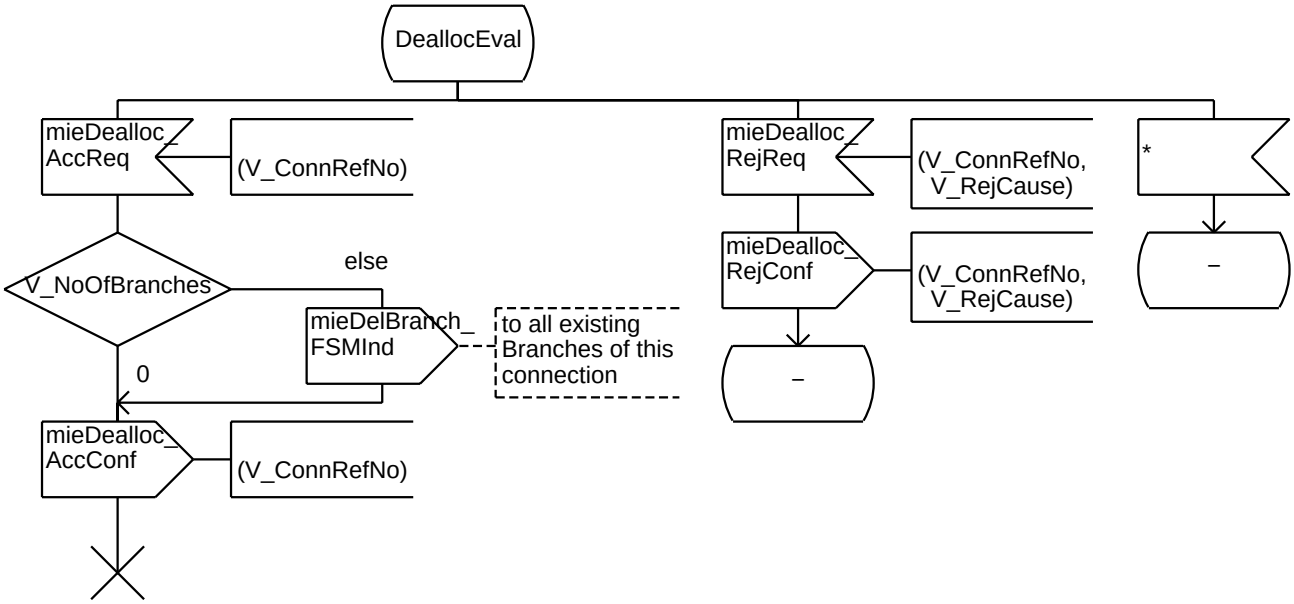

InService
mieDrop_
BranchInd
(V_BranchId)
mieDrop_
BranchFSM_
Ind
TO A_BranchList
(V_BranchId)
–
mieDrop_
BranchAcc_
Req
(V_ConnRefNo,
V_BranchId)
A_BranchList
(V_BranchId)
NULL
else
mieDrop_
BranchFSM_
AccReq
(V_BranchId)
TO A_BranchList
(V_BranchId)
–
–
mieDrop_
BranchRej_
Req
(V_ConnRefNo,
V_BranchId,
V_RejCause)
A_BranchList
(V_BranchId)
NULL
else
mieDrop_
BranchFSM_
RejReq
(V_BranchId,
V_RejCause)
TO A_BranchList
(V_BranchId)
–
–


InService
mieDrop_
BranchFSM_
AccConf
(V_BranchId)
mieDrop_
Branch_
AccConf
(V_ConnRefNo,
V_BranchId)
V_NoOfBranches
:=
V_NoOfBranches−1
–
mieDrop_
BranchFSM_
RejConf
(V_BranchId,
V_RejCause)
mieDrop_
Branch_
RejConf
(V_ConnRefNo,
V_BranchId,
V_RejCause)
–


AllocEval
mieAllocAcc_
Req
(V_ConnRefNo,
V_BranchId)
mieAllocRej_
Req
(V_ConnRefNo,
V_BranchId,
V_RejCause)
*
C_BranchIdNotPresent
V_BranchId
mieAllocRej_
Conf
(V_ConnRefNo,
V_BranchId,
V_RejCause)
–
else
V_BranchId =
V_FirstBranchId
false
true
SN_BRANCH_
_FSM
A_BranchList
(V_BranchId)
:=OFFSPRING
mieFirstAdd_
BranchFSM_
AccInd
(V_BranchId)
TO OFFSPRING
V_NoOfBranches
:=1
mieAllocAcc_
Conf
(V_ConnRefNo,
V_BranchId)
mieAllocRej_
Conf
(V_ConnRefNo,
V_FirstBranchId,
C_TransErr)
InService

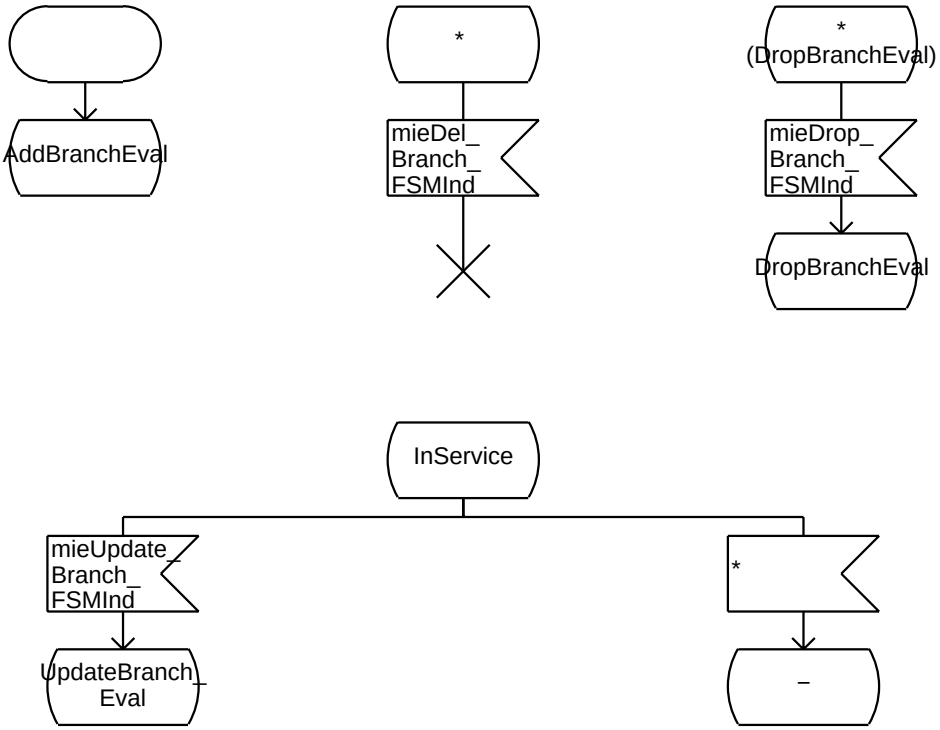

AllocComp_
Eval
mieAlloc_
CompAccReq
(V_ConnRefNo)
mieAlloc_
CompRejReq
(V_ConnRefNo,
V_RejCause)
*
mieAlloc_
CompAccConf
(V_ConnRefNo)
V_NoOfBranches
–
else
InService
0
mieDelBranch_
FSMInd
to all existing
Branches of this
connection
mieAlloc_
CompRejConf
(V_ConnRefNo,
V_RejCause)

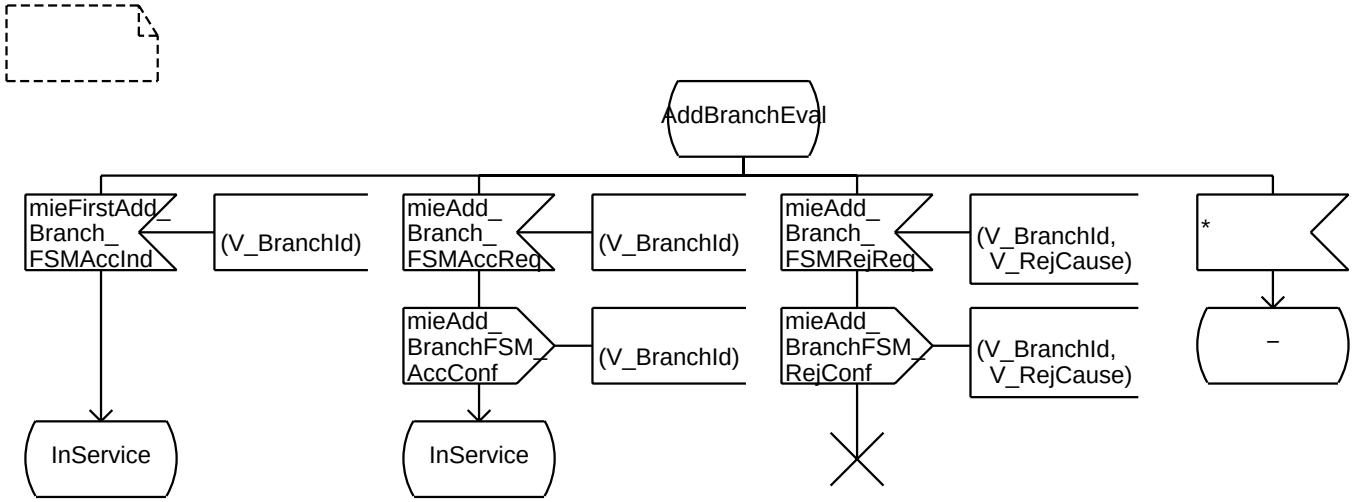

ModifyEval
mieModify_ AccReq
(V_ConnRefNo)
mieModify_ RejReq
(V_ConnRefNo, V_RejCause)
*
mieModify_ AccConf
(V_ConnRefNo)
mieModify_ RejConf
(V_ConnRefNo, V_RejCause)
–
InService
InService
ModifyComp_ Eval
mieModify_ CompAccReq
(V_ConnRefNo)
mieModify_ CompRejReq
(V_ConnRefNo, V_RejCause)
*
mieModify_ CompAccConf
(V_ConnRefNo)
mieModify_ CompRejConf
(V_ConnRefNo, V_RejCause)
–
InService
InService
ModifyAbort_ Eval
mieModify_ AbortAccReq
(V_ConnRefNo)
mieModify_ AbortRejReq
(V_ConnRefNo, V_RejCause)
*
mieModify_ AbortAccConf
(V_ConnRefNo)
mieModify_ AbortRejConf
(V_ConnRefNo, V_RejCause)
–
InService
InService

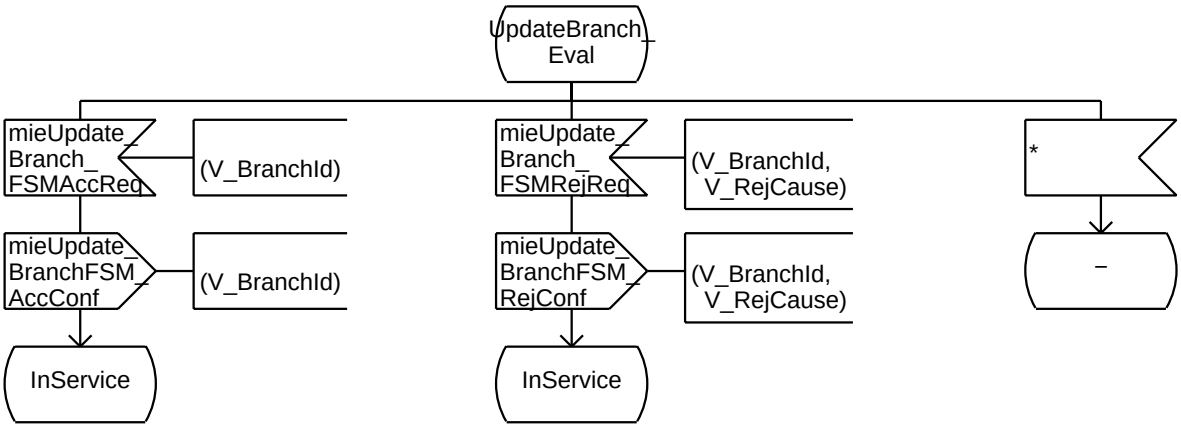

DeallocEval
mieDealloc_
AccReq
(V_ConnRefNo)
mieDealloc_
RejReq
(V_ConnRefNo,
V_RejCause)
*
mieDealloc_
RejConf
(V_ConnRefNo,
V_RejCause)
–
V_NoOfBranches
else
mieDelBranch_
FSMInd
to all existing
Branches of this
connection
0
–
mieDealloc_
AccConf
(V_ConnRefNo)

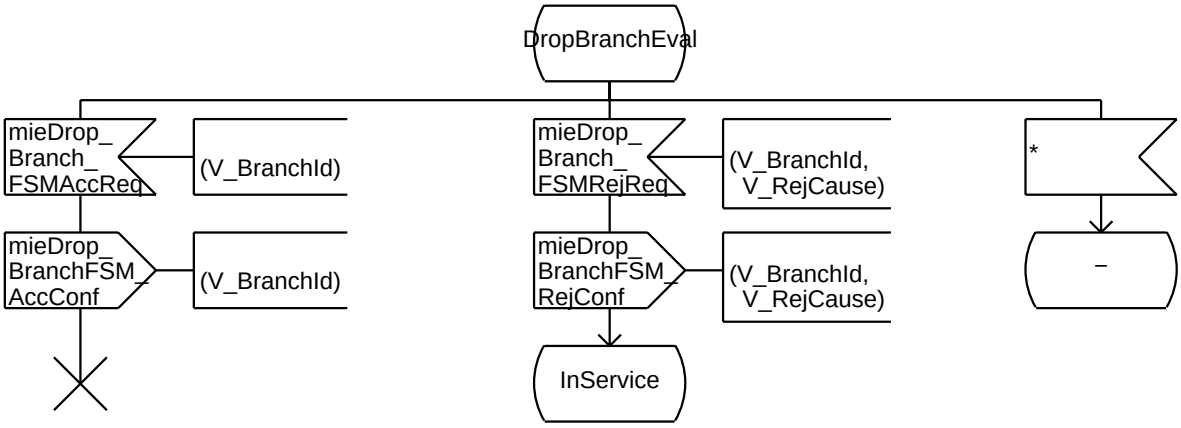
```
/* State descriptions
InService:        The process is not waiting for a confirmation of a message
AddBranchEval:    An add branch is sent but not confirmed yet
UpdateBranchEval: An update branch is sent but not confirmed yet
DropBranchEval:   An drop branch is sent but not confirmed yet
*/
```

```
/*Primitive Data Declarations*/
/**/
/* Branch identifier */
DCL
V_BranchId  IT_BranchId;
/**/
/* Reject Cause */
DCL
V_RejCause  IT_RejCause;
```




AddBranchEval
mieFirstAdd_ Branch_ FSMAccInd
(V_BranchId)
mieAdd_ Branch_ FSMAccReq
(V_BranchId)
mieAdd_ Branch_ FSMRejReq
(V_BranchId, V_RejCause)
*
mieAdd_ BranchFSM_ AccConf
(V_BranchId)
mieAdd_ BranchFSM_ RejConf
(V_BranchId, V_RejCause)
–
InService
InService

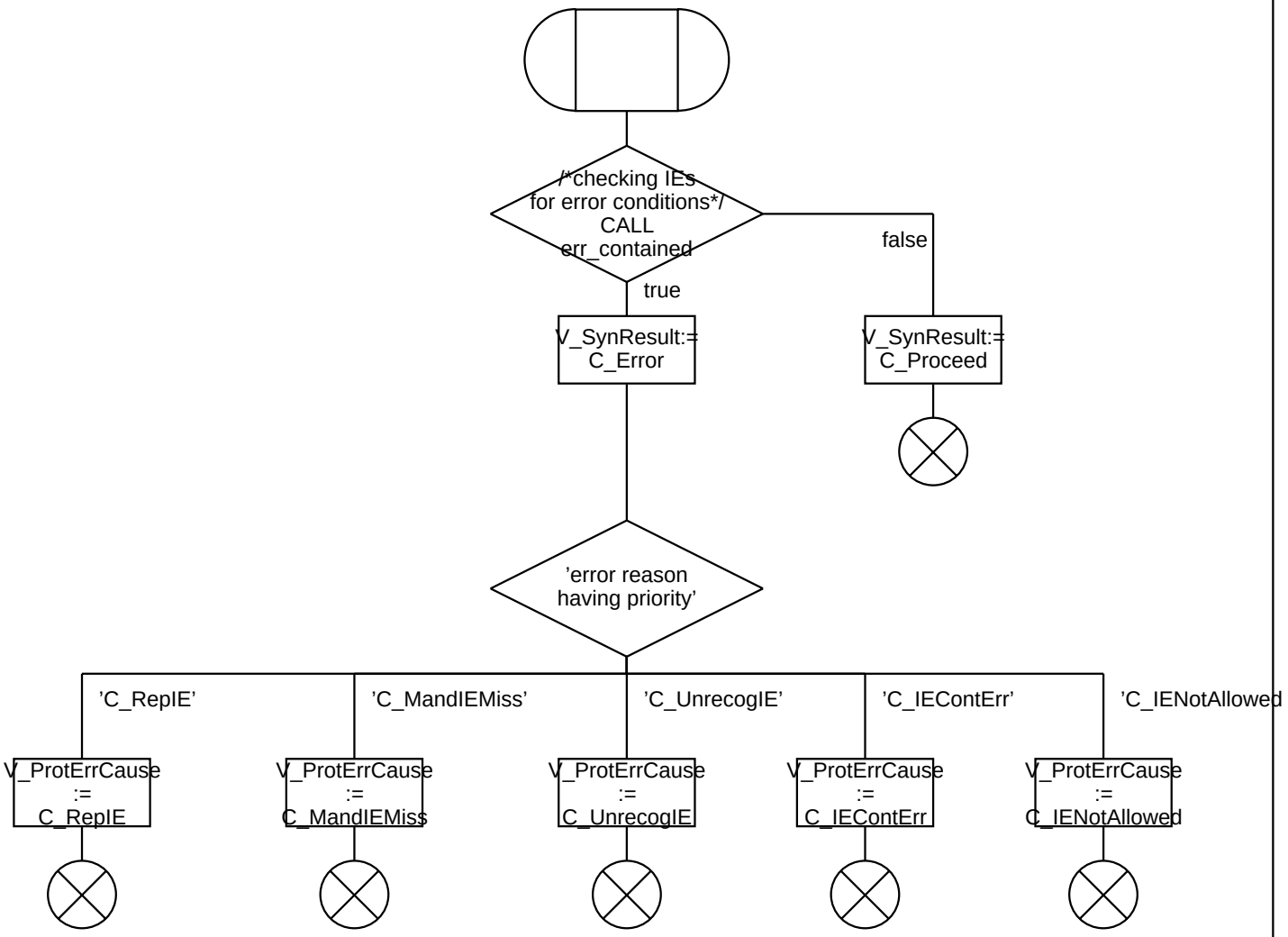

UpdateBranch_ Eval
mieUpdate_ Branch_ FSMAccReq
(V_BranchId)
mieUpdate_ Branch_ FSMRejReq
(V_BranchId, V_RejCause)
*
mieUpdate_ BranchFSM_ AccConf
(V_BranchId)
mieUpdate_ BranchFSM_ RejConf
(V_BranchId, V_RejCause)
–
InService
InService


DropBranchEval
mieDrop_ Branch_ FSMAccReq
(V_BranchId)
mieDrop_ Branch_ FSMRejReq
(V_BranchId, V_RejCause)
*
mieDrop_ BranchFSM_ AccConf
(V_BranchId)
mieDrop_ BranchFSM_ RejConf
(V_BranchId, V_RejCause)
–
InService

Procedure IE_CHECK

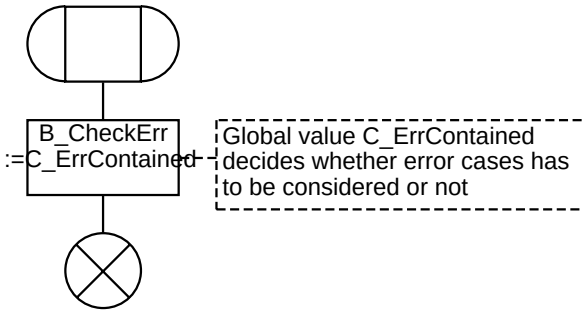

```
;
FPAR
IN/OUT V_ProtErrCause  IT_ProtErrCause;
RETURNS V_SynResult INTEGER;
```

```
/* Procedure used only for simulation purposes. Supports the syntax error checking
   of  IEs. The procedure checks first wether error checking is activated at all via the
   external synonym C_ErrContained. If error checking is activated, then the possible
   protocol error cause (see subclause 14.3.7.10) can be set by direct input during simulation.
   If error checking is de−activated, the normal procedures apply. */
```



Procedure ERR_CONTAINED

;RETURNS
B_CheckErr BOOLEAN;
/*Procedure only for
simulation use. Decides whether
error cases have to be considered
or not*/

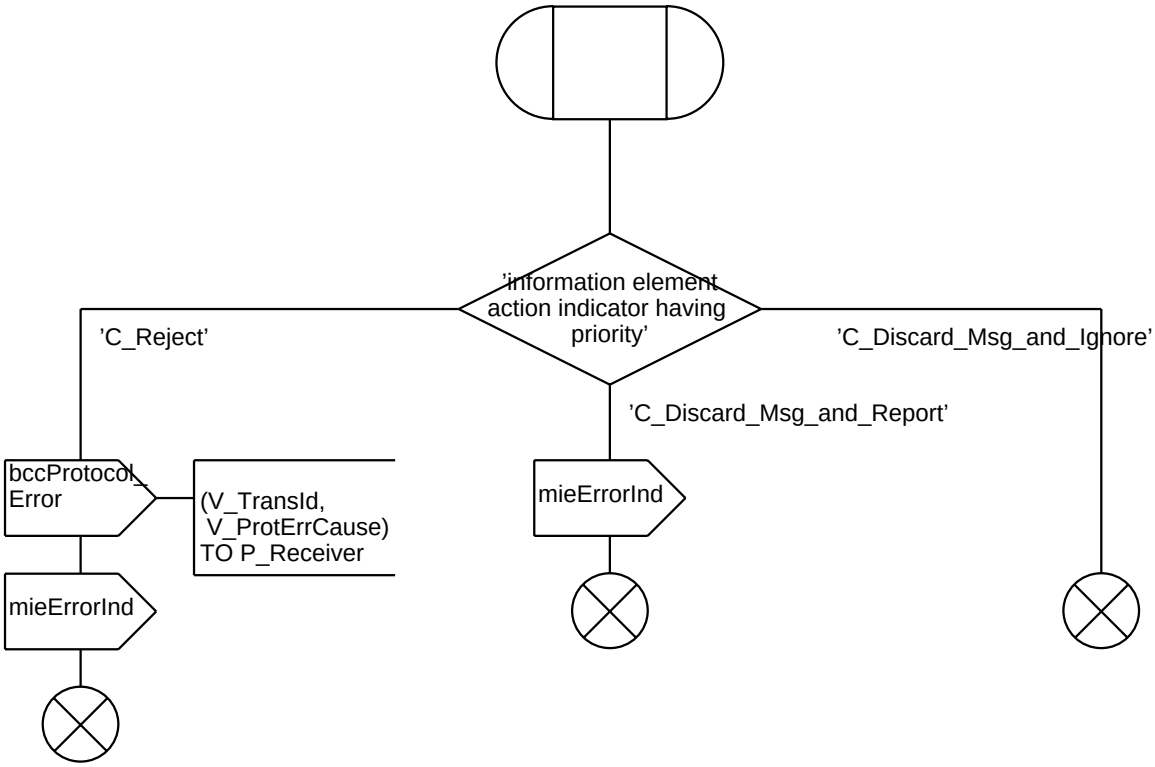


Procedure IE_ERROR_HANDLING 1(1)

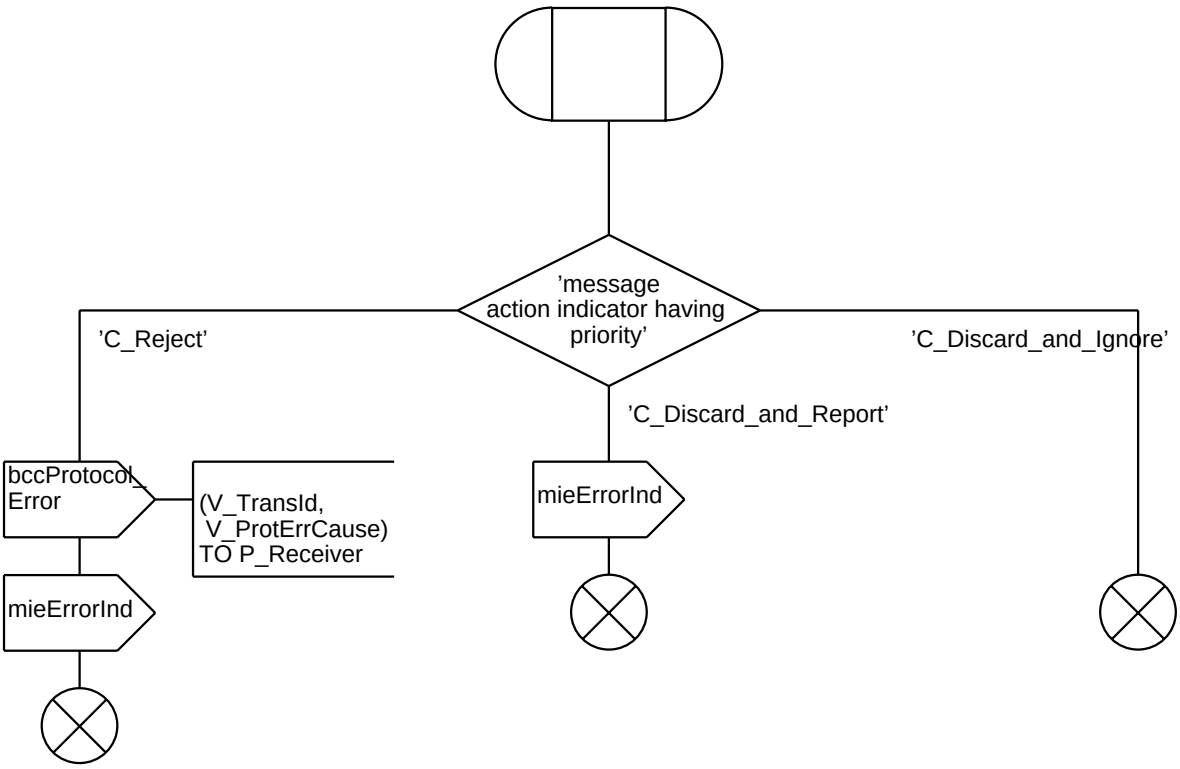
```
;
FPAR
V_TransId          IT_TransId,
V_ProtErrCause     IT_ProtErrCause,
P_Receiver         PID;
```

/* Procedure used only for simulation purposes. The possible actions for handling of an IE syntax error (see table 36) can be set by direct input during simulation. */

/* The possible actions 'Discard IE and proceed' and 'Discard IE, proceed and report' are not included in the simulation for simplification reasons but shall be included in an implementation */



Procedure MSG_ERROR_HANDLING 1(1)

```
;
FPAR
V_TransId          IT_TransId,
V_ProtErrCause  IT_ProtErrCause,
P_Receiver         PID;
```

/* Procedure used only for simulation purposes. The possible actions for handling of syntax errors in the common message information fields (see table 35) can be set by direct input during simulation. */