Table of Contents

Organizer View 1
Package VB5_1 3
Procedure ANA_RSC 40
Procedure UNK_RSC 45
Procedure UPDATE_TRANSLIST 46
Procedure ERR_CONTAINED 47
Procedure IE_CHECK 48
Procedure IE_ERROR_HANDLING 49
Procedure MSG_ERROR_HANDLING 50
Procedure CONV_RSCID 51
Procedure IE_CONTAINED 52
Process Type CTR 53
Process Type LSPID_VERIFY 60
Process Type RESET_RSC 65
Process Type SAAL_MGT 82
Block Type AN 87
Block AN_SYSMGT 88
Process AN_VPCI_CC 90
Block AN_RSCMGT 95
Process AN_STATUS_MGT 97
Process AN_VPCSTAT 118
Process AN_LSPSTAT 121
Block AN_CPROT 123
Process AN_MGR_CPROT 124
Process AN_CPROT 134
Block Type SN 150
Block SN_SYSMGT 151
Process SN_VPCI_CC 153
Block SN_RSCMGT 157
Process SN_STATUS_MGT 159
Procedure decode_block_ind 174

Procedure reason_decode 179
Process SN_VPCSTAT 180
Process SN_LSPSTAT 182
Block SN_CPROT 183
Process SN_MGR_CPROT 184
Process SN_CPROT 194
System AN_SN_VB5_1 210
System AN 211
System SN 212

Diagram Structure

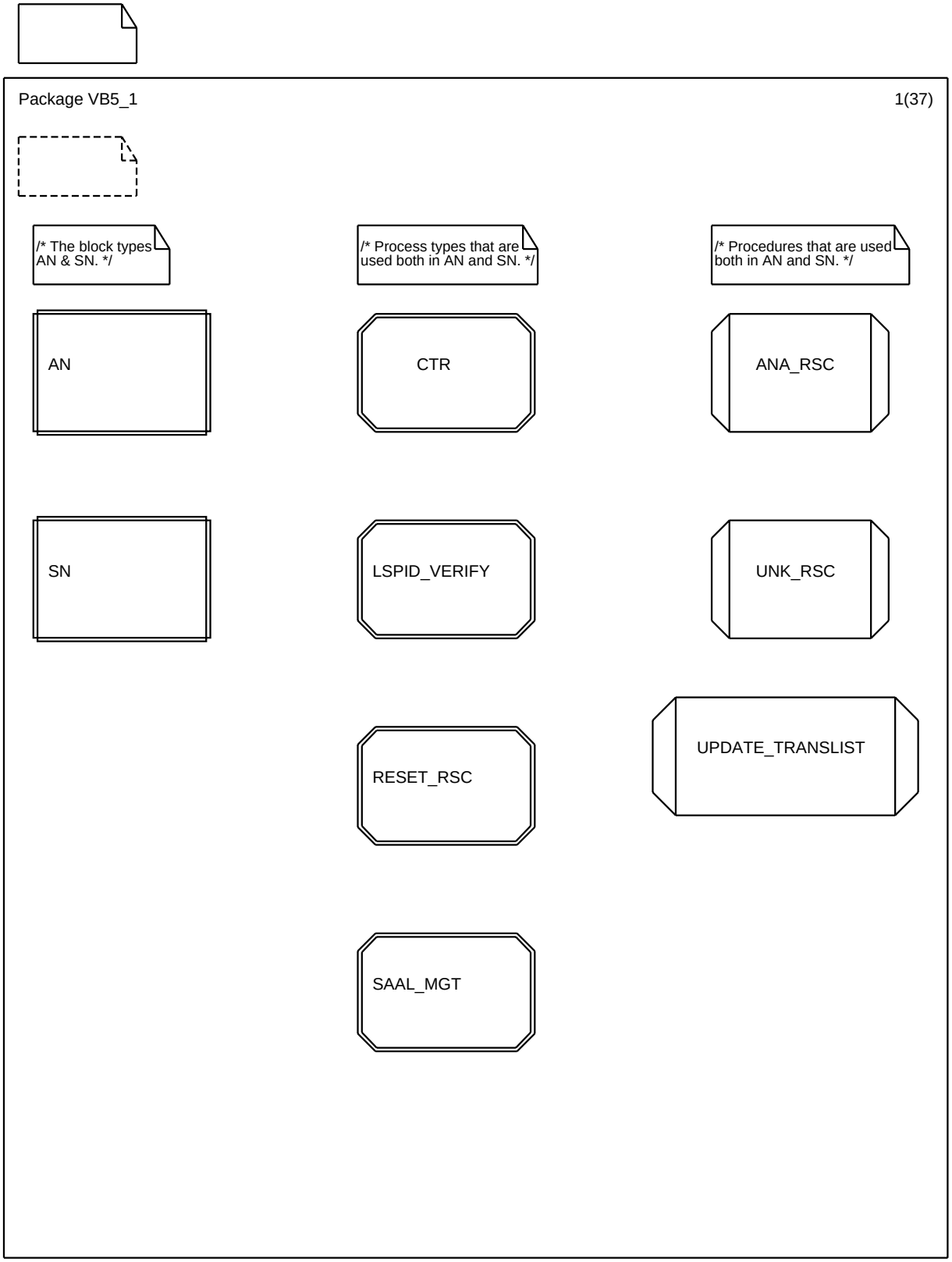
VB5_1	vb5_1.sun
ANA_RSC	ana_res.spd
UNK_RSC	unk_rsc.spd
UPDATE_TRANSLIST	updlist.spd
ERR_CONTAINED	checkerr.spd
IE_CHECK	syntax.spd
IE_ERROR_HANDLING	ie_errh.spd
MSG_ERROR_HANDLING	msg_errh.spd
CONV_RSCID	conrscid.spd
IE_CONTAINED	ie_cont.spd
CTR	ctr.spt
LSPID_VERIFY	lspid.spt
RESET_RSC	resrsc.spt
SAAL_MGT	saal_mgt.spt
AN	an.sbt
AN_SYSMGT	an_sysmg.sbk
x:y AN_CTR : CTR	
x:y AN_LSPID : LSPID_VERIFY	
x:y AN_RESET : RESET_RSC	
x:y AN_SAAL : SAAL_MGT	
AN_VPCI_CC	ancons.spr
AN_RSCMGT	anrscmg.sbk
AN_STATUS_MGT	anstatus.spr
AN_VPCSTAT	anvpcst.spr
AN_LSPSTAT	anlspst.spr
AN_CPROT	an_cprot.sbk
AN_MGR_CPROT	ancpmgr.spr
AN_CPROT	ancprot.spr
SN	sn.sbt
SN_SYSMGT	sn_sysmg.sbk
x:y SN_CTR : CTR	
x:y SN_LSPID : LSPID_VERIFY	
x:y SN_RESET : RESET_RSC	
x:y SN_SAAL : SAAL_MGT	
SN_VPCI_CC	sncons.spr
SN_RSCMGT	snrscmg.sbk

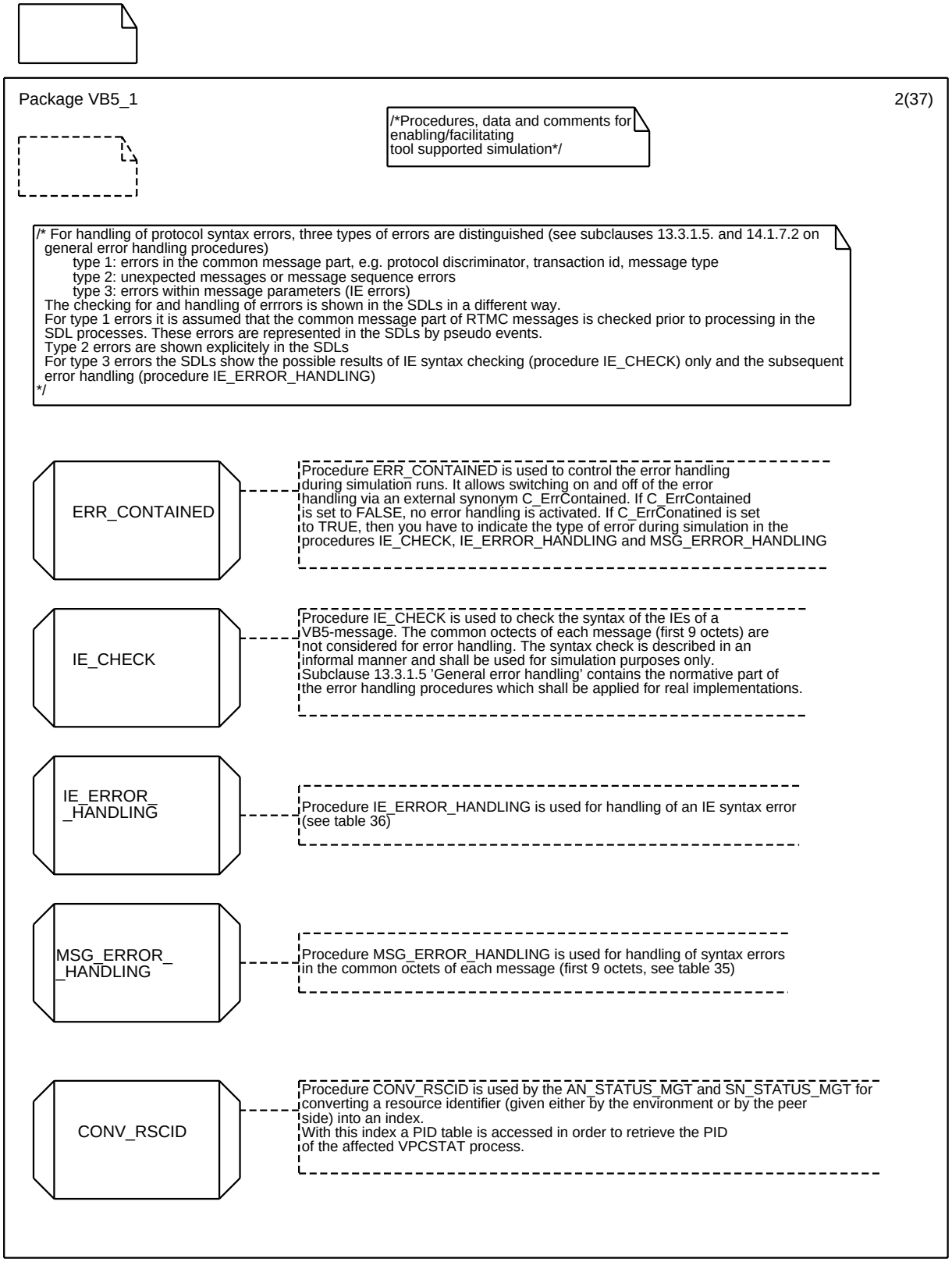
SN_STATUS_MGT	snstatus.spr
decode_block_ind	dec_bloc.spd
reason_decode	dec_reas.spd
SN_VPCSTAT	snvpcst.spr
SN_LSPSTAT	snlspst.spr
SN_CPROT	sn_cprot.sbk
SN_MGR_CPROT	sncpmgr.spr
SN_CPROT	sncprot.spr

Systems

AN_SN_VB5_1	ansn.ssy
x:y AN_VB5_1 : AN	
x:y SN_VB5_1 : SN	
AN	an.ssy
x:y AN_VB5_1 : AN	
SN	sn.ssy
x:y SN_VB5_1 : SN	

Associated Documents


Package VB5_1
1(37)
/* 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
CTR
ANA_RSC
SN
LSPID_VERIFY
UNK_RSC
UPDATE_TRANSLIST
RESET_RSC
SAAL_MGT


Package VB5_1
2(37)
/*Procedures, data and comments for enabling/facilitating tool supported simulation*/
/* For handling of protocol syntax errors, three types of errors are distinguished (see subclauses 13.3.1.5. and 14.1.7.2 on general error handling procedures)
type 1: errors in the common message part, e.g. protocol discriminator, transaction id, message type
type 2: unexpected messages or message sequence errors
type 3: 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.
Type 2 errors are shown explicitely in the SDLs
For type 3 errors the SDLs show the possible results of IE syntax checking (procedure IE_CHECK) only and the subsequent error handling (procedure IE_ERROR_HANDLING)
*/
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 in the procedures IE_CHECK, IE_ERROR_HANDLING and MSG_ERROR_HANDLING
IE_CHECK
Procedure IE_CHECK is used to check the syntax of the IEs of a VB5-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.
IE_ERROR_ _HANDLING
Procedure IE_ERROR_HANDLING is used for handling of an IE syntax error (see table 36)
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 35)
CONV_RSCID
Procedure CONV_RSCID is used by the AN_STATUS_MGT and SN_STATUS_MGT for converting a resource identifier (given either by the environment or by the peer side) into an index.
With this index a PID table is accessed in order to retrieve the PID of the affected VPCSTAT process.

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

IE_CONTAINED

```
Procedure IE_CONTAINED shall be used for analysis of RscIds within a VB5 message
(procedure ANA_RSC).
For simulation purposes this procedure evaluates the number of present RscIds.
The RscIds are stored within an unlimited array A_RTMCRscList. The default value of
each entry is (. 0,0,0,0 .) which is the initial value within the SDT tool. This value
is represented by C_EmptyRscId, i.e. C_EmptyRscId has to be set to (. 0,0,0,0 .)
To check whether a resource ID is present or not, this synonym C_EmptyRscId is used.
```

```
/*Resource identifier IE not present*/
SYNONYM C_EmptyRscId ST_RscId = (. 0, 0, 0, 0 .);
```

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

```
signal
SIM_SN_started, /* primitive to announce the PId of the SN_MGR_CPROT process to the AN_MGR_CPROT process */
SIM_AN_started; /* primitive to announce the PId of the AN_MGR_CPROT process to the SN_MGR_CPROT process */
```

```
/* Provisioning of resource identifiers to VPCSTAT processes*/
/*
AN_VPCSTAT and SN_VPCSTAT processes are instantiated dynamically by the AN/SN_STATUS_MGT process
when a meeVPCCreateReq primitive is received. The RscId of each VPC is given by a parameter of the primitive to the
corresponding STATUS_MGT process, see signal definition.
The RscId is only known by the STATUS_MGT process. This process maps the RscId to the corresponding PID. I.e. the
VPCSTAT process needs not to know his own RscId.
For the indication of RscIds within primitives, see definition of type ST_RscId. The following components are used:
- Resource Indicator
      0 complete LSP
      1 connections at LSP
      2 connections at LUP
- Unknown Resource Indicator, this value is only relevant in _conf primitives
      0 LSP Id or LUP Id unknown
      1 VPCI unknown
- value of LUP Identifier/LSP Identifier
- value of VPCI
  Note:
  For simulation only a single VPCI and not a range of VPCIs per RscId is considered

Example: For creating three VPCI status FSMs for VPCI=1..3 at LUPId=4 you may specify the following primitives
at AN and SN side:
meeVPCCreateReq (. 2, 0, 4, 1 .)
meeVPCCreateReq (. 2, 0, 4, 2 .)
meeVPCCreateReq (. 2, 0, 4, 3 .)

It is assumed that  - the VPCI value of a VPC resource is unique in the VB5.1 system
                     - the VPCI values start with 1
                     - there are no gaps in the provisioned VPCI values
*/
```

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

```
/* 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 meeVPCCreateReq (. 2, 0, 10, 1 .) CH_ANVB5_ENV
go
output-via meeVPCCreateReq (. 2, 0, 20, 2 .) CH_ANVB5_ENV
go
output-via meeVPCCreateReq (. 2, 0, 30, 3 .) CH_ANVB5_ENV
go
output-via meeVPCCreateReq (. 2, 0, 10, 1 .) CH_SNVB5_ENV
go
output-via meeVPCCreateReq (. 2, 0, 20, 2 .) CH_SNVB5_ENV
go
output-via meeVPCCreateReq (. 2, 0, 30, 3 .) CH_SNVB5_ENV
go
output-via meeStartupReq CH_SNVB5_ENV
proceed-until 1
output-via AAL_establish_conf CH_SNMGT_SAAL
output-via AAL_establish_ind CH_ANMGT_SAAL
go
---------------------------------------------------- end of file
*/
```

```
/* Assignment of values to the external synonyms */
```

```
/*Local LSPID*/
SYNONYM C_LocalLSPId INTEGER = EXTERNAL; /*Own LSPID*/
```

```
/*Activation or Deactivation of error handling*/
SYNONYM C_ErrContained BOOLEAN = EXTERNAL;
```

```
/*
Before you can start simulation, the values for the external synonyms C_ErrContained and C_LocalLSPId
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;
C_LocalLSPId     8;
---------------------------------------------- end of input-file
*/
```

/* 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
*/
```

/* For naming conventions for primitives see subclause 13.2.1.2 */

/* 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.
*/
```

/*Data Type Definitions for VB5 messages*/

```
/*RESOURCE IDENTIFIER Information Element*/
/*refer to subclause 14.2.2.6*/
/**/
/*Typedefinition for Resource Indicator Value*/
/*refer to table 58*/
SYNTYPE IT_RTMCRscInd = INTEGER
   CONSTANTS 0:2
ENDSYNTYPE IT_RTMCRscInd;
/*Possible Resource Indicator Values*/
SYNONYM C_CompLSP    IT_RTMCRscInd = 0; /*Complete LSP*/
SYNONYM C_ConnLSP    IT_RTMCRscInd = 1; /*Connections at LSP*/
SYNONYM C_ConnLUP    IT_RTMCRscInd = 2; /*Connections at LUP*/
/**/
/*Typedefinition for Unknown Resource Indicator Value*/
/*refer to table 58*/
SYNTYPE IT_RTMCUnkRscInd = INTEGER
   CONSTANTS 0:1
ENDSYNTYPE IT_RTMCUnkRscInd;
/*Possible Unknown Resource Indicator Values*/
SYNONYM C_LogIdUnk    IT_RTMCUnkRscInd = 0; /*LSP Id or LUP Id unknown*/
SYNONYM C_VPCIUnk     IT_RTMCUnkRscInd = 1; /*VPCI unknown*/
/**/
/*Typedefinition for Resource Identifier Information Element*/
/*refer to figure 74*/
NEWTYPE ST_RscId STRUCT
   V_RscInd          IT_RTMCRscInd;        /*Resource Indicator*/
   V_UnkRscInd       IT_RTMCUnkRscInd;     /*Unknown resource Indicator*/
   V_LogId           NATURAL;              /*LUP/LSP Identifier*/
   V_VPCI            NATURAL;              /*VPCI*/
ENDNEWTYPE;
```

/*Data Type Definitions for VB5 messages*/

```
/*BLOCKED RESOURCE IDENTIFIER Information Element*/
/*refer to subclause 14.2.2.2*/
/**/
/*Typedefinition for Administrative Reason Value*/
/*refer to table 54*/
SYNTYPE IT_RTMCAdmReason = INTEGER
   CONSTANTS 0:2
ENDSYNTYPE IT_RTMCAdmReason;
/*Possible Administrative Reason Values*/
SYNONYM C_RTMCAdmNone    IT_RTMCAdmReason = 0;
SYNONYM C_RTMCAdmPart    IT_RTMCAdmReason = 1;
SYNONYM C_RTMCAdmFull    IT_RTMCAdmReason = 2;
/**/
/*Typedefinition for Error Reason Value*/
/*refer to table 54*/
SYNTYPE IT_RTMCErrReason = INTEGER
   CONSTANTS 0:1
ENDSYNTYPE IT_RTMCErrReason;
/*Possible Error Reasons*/
SYNONYM C_RTMCErrNone IT_RTMCErrReason = 0;
SYNONYM C_RTMCErr     IT_RTMCErrReason = 1;
/**/
/*Typedefinition for Blocking Reason Indicator*/
/*refer to table 54*/
NEWTYPE ST_RTMCReason STRUCT
   V_RTMCAdmReason   IT_RTMCAdmReason;
   V_RTMCErrReason   IT_RTMCErrReason;
ENDNEWTYPE ST_RTMCReason;
/**/
/*Typedefinition for Blocked Resource Identifier Information Element*/
/*refer to figure 70*/
NEWTYPE ST_RTMCBlRscId STRUCT
   S_RTMCRscId       ST_RscId;          /*Resource Identifier IE*/
   S_RTMCReason      ST_RTMCReason;   /*Reason IE*/
ENDNEWTYPE;
```

```
/*Typedefinition for repeated information elements*/
/**/
/*Definition of maximum length of a message*/
SYNONYM C_MaxMsgLength INTEGER =4096;
/**/
/*Typedefinition for repeated Resource Identifier Information Elements in a RTMC message*/
NEWTYPE AT_RscList ARRAY
   (INTEGER,ST_RscId)
ENDNEWTYPE;
/**/
/*Typedefinition for repeated Blocked Resource Identifier Information Elements in a RTMC message*/
NEWTYPE AT_RTMCBlRscList ARRAY
   (INTEGER,ST_RTMCBlRscId)
ENDNEWTYPE;
```

```
/*Data Type Definitions for
VB5 messages*/
```

```
/*PROTOCOL ERROR CAUSE Information Element*/
/*refer to subclause 14.2.2.3*/
/**/
/*Typedefinition for Protocol Error Cause Value*/
/* refer to table 55*/
SYNTYPE IT_RTMCProtErrCause = INTEGER
   CONSTANTS 1:7
ENDSYNTYPE IT_RTMCProtErrCause;
/**/
/*Possible Protocol Error Cause Values*/
SYNONYM C_UnkMsgType                    IT_RTMCProtErrCause = 1;
SYNONYM C_RepIE                         IT_RTMCProtErrCause = 2;
SYNONYM C_MandIEMiss                    IT_RTMCProtErrCause = 3;
SYNONYM C_UnrecogIE                     IT_RTMCProtErrCause = 4;
SYNONYM C_IEContErr                     IT_RTMCProtErrCause = 5;
SYNONYM C_IENotAllowed                  IT_RTMCProtErrCause = 6;
SYNONYM C_MsgNotCompatWithProtState IT_RTMCProtErrCause = 7;
```

```
/*RESULT Information Element
*/
/*refer to subclause 14.2.2.5*/
/**/
/*Typedefinition for Result Indicator Value*/
/* refer to table 51*/
SYNTYPE IT_RTMCResult = INTEGER
CONSTANTS 0:4
ENDSYNTYPE IT_RTMCResult;
/**/
/*Possible Values, see Definition of IT_Result */
```

```
/*Transaction Identifier
*/
/*refer to table 35*/
SYNTYPE PT_TransId = PID
ENDSYNTYPE PT_TransId;
```

```
/*Data Type Definitions for
Primitives*/
```

```
/* Some data type definitions
   for VB5 messages are reused */
```

```
/*STATUS CHANGE REASONS*/
/**/
/*Used in mee block request messages to AN_STATUS_MGT process and MIE_ to
  AN_VPCSTAT/AN_LSPSTAT processes*/
/**/
SYNTYPE IT_Reason = INTEGER
   CONSTANTS 1:5
ENDSYNTYPE IT_Reason;
/**/
/*Possible Values*/
SYNONYM C_AdmPart        IT_Reason = 1;  /*Admin Partial*/
SYNONYM C_AdmFull         IT_Reason = 2;  /*AdminFulll*/
SYNONYM C_Error           IT_Reason = 3;  /*Error*/
SYNONYM C_AdmPartErr   IT_Reason = 4;  /*Admin Partial and Error*/
SYNONYM C_AdmFullErr     IT_Reason = 5;  /*Admin Full and Error*/
```

```
/*LIST OF RESOURCES*/
/**/
/*List of blocked resources used in meeBlockRscInd primitive in the SN*/
/**/
NEWTYPE ST_BlRscId STRUCT
   S_RscId              ST_RscId;
   V_Reason   IT_Reason;
ENDNEWTYPE ST_BlRscId;
/**/
NEWTYPE AT_BlRscList ARRAY
   (INTEGER, ST_BlRscId)
ENDNEWTYPE AT_BlRscList;
```

/*Data Type Definitions for Primitives*/

```
/*OPERATION RESULT*/
/**/
/*used in mee confirmation primitives to the environment*/
/**/
SYNTYPE IT_Result = INTEGER
   CONSTANTS 0:8
ENDSYNTYPE IT_Result;
/**/
/*Possible Values*/
SYNONYM C_Success          IT_Result = 0;
SYNONYM C_UnkRsc           IT_Result = 1;  /* unknown resource(s) */
SYNONYM C_OpRejLoc         IT_Result = 2;  /* operation rejected at local network element */
SYNONYM C_OpRej            IT_Result = 3;  /* operation rejected at peer entity */
/* transparent in FSM  SYNONYM C_OpFail      IT_Result = 4; operation failed */
SYNONYM C_TransErr         IT_Result = 5; /* message exchange transmission error */
SYNONYM C_Mismatch         IT_Result = 6; /* mismatch of identifier at local and peer network element */
/* transparent in FSM  SYNONYM C_NotPerf     IT_Result = 7; not performed */
SYNONYM C_SAAL             IT_Result = 8; /* not recoverable failure of SAAL */
```

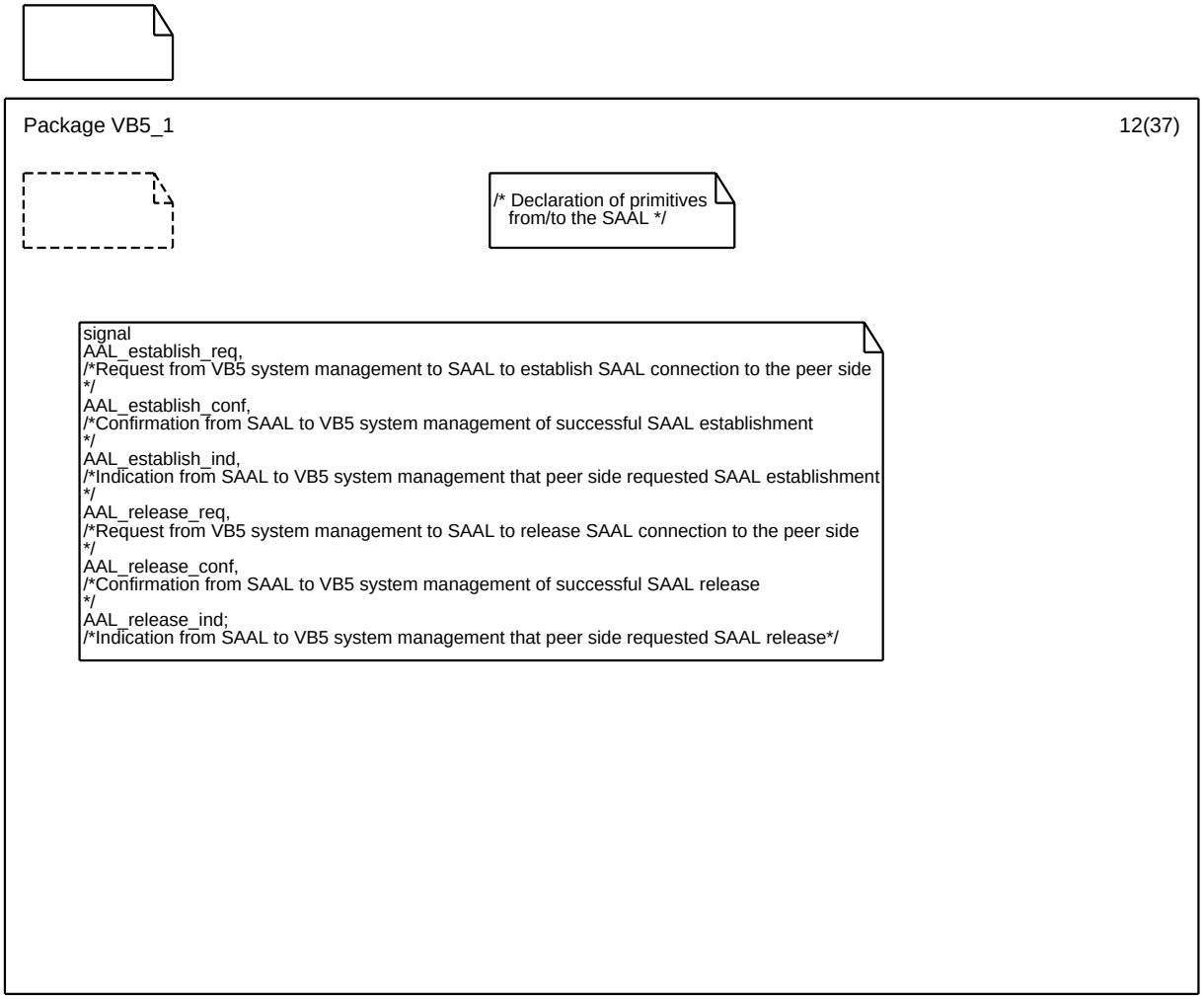
```
/*INTERNAL TRANSACTION IDENTIFIER*/
/**/
/*used in MDU_ primitives from/to AN_STATUS_MGT process in order to identify all RTMC messages which have
   been triggered by a single mee request*/
/**/
SYNTYPE IT_IntTransId = INTEGER
ENDSYNTYPE IT_IntTransId;
/**/
SYNONYM C_UndefIntTransId    IT_IntTransId = 0;
/**/
/*Data Type Definitions for CPROT Process instance Id*/
/**/
SYNTYPE PT_CprotPID = PID
ENDSYNTYPE PT_CprotPID;
```

/* Definitions of common DataTypes */

```
/*Typedefinition for Transaction ID table, used for co-ordinating MDU_ primitives*/
NEWTYPE ST_IntTransInfo STRUCT
   V_NoOfRTMCMsg     INTEGER;
   A_UnkRscList      AT_RscList;
   A_RscList         AT_RscList
ENDNEWTYPE ST_IntTransInfo;
NEWTYPE AT_IntTransList ARRAY
   (INTEGER, ST_IntTransInfo)
ENDNEWTYPE AT_IntTransList;
```

/*Declaration of VB5 messages*/

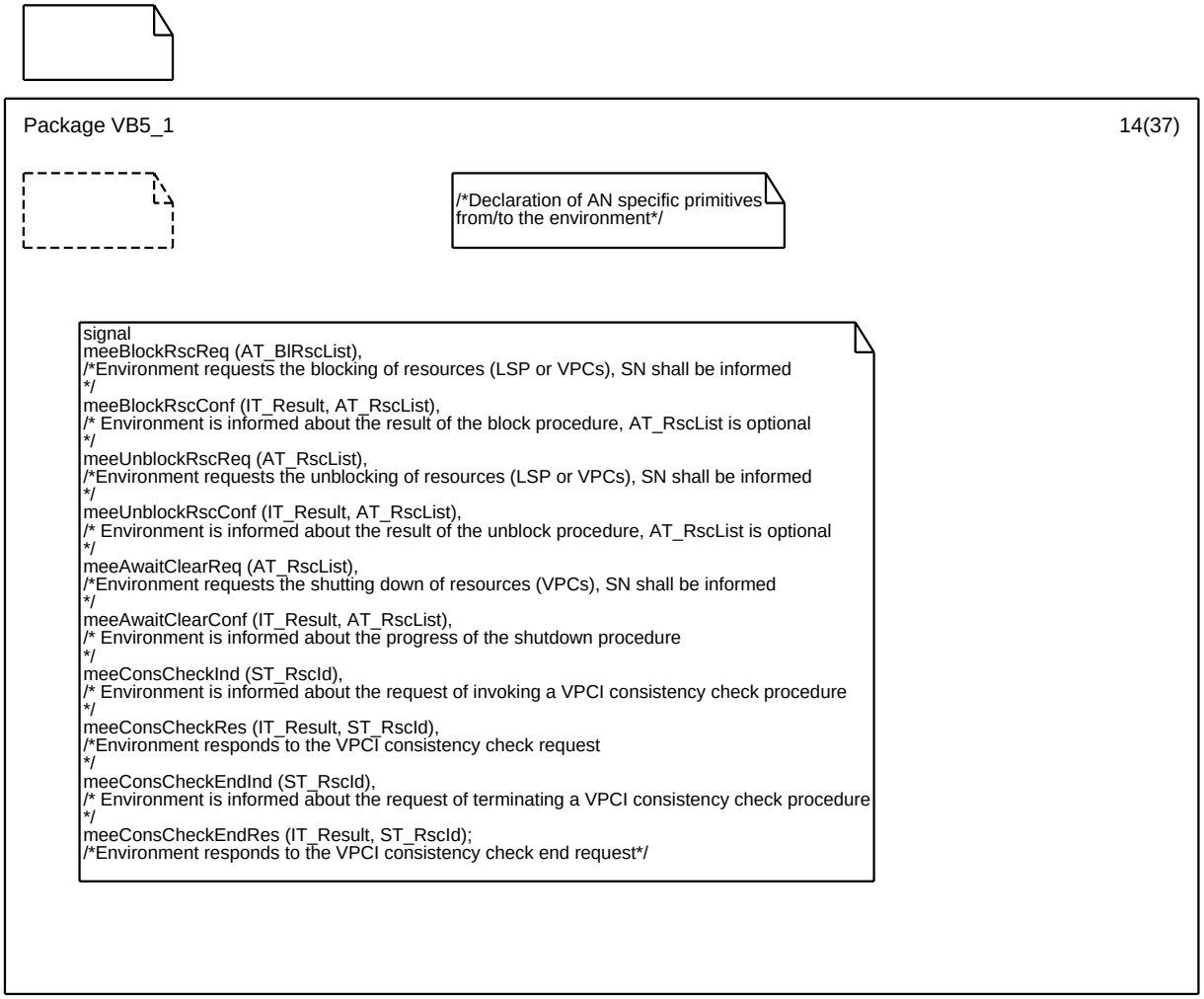
```
signal
BLOCK_RSC (PT_TransId, AT_RTMCBlRscList),
/*Indication from AN to SN that resources (LSP or VPCs) have become unavailable for Call Processing
*/
BLOCK_RSC_ACK (PT_TransId, IT_RTMCResult, AT_RscList),
/*Direct acknowledgement  on BLOCK_RSC
*/
UNBLOCK_RSC (PT_TransId, AT_RscList),
/*Indication from AN to SN that resources (LSP or VPCs) have become available for Call Processing
*/
UNBLOCK_RSC_ACK (PT_TransId, IT_RTMCResult, AT_RscList),
/*Direct acknowledgement  on UNBLOCK_RSC
*/
AWAIT_CLEAR (PT_TransId, AT_RscList),
/*Request from AN to SN to shutdown resources (VPCs)
*/
AWAIT_CLEAR_ACK (PT_TransId, IT_RTMCResult, AT_RscList),
/*Direct acknowledgement  on AWAIT_CLEAR
*/
AWAIT_CLEAR_COMP (PT_TransId, AT_RscList),
/*Indication from SN to AN that no on-demand connections are any longer present on the indicated resources
*/
AWAIT_CLEAR_COMP_ACK (PT_TransId),
/*Direct acknowledgement  on AWAIT_CLEAR_COMP
*/
REQ_LSPID (PT_TransId),
/*Request to the peer side to report its LSP ID
*/
LSPID (PT_TransId, ST_RscId),
/*Direct acknowledgement  on REQ_LSPID
*/
RESET_RSC (PT_TransId, AT_RscList),
/*Request to the peer side to reset resources (VPC/LSP), i.e. to release all on-demand connections
*/
RESET_RSC_ACK (PT_TransId, IT_RTMCResult, AT_RscList),
/*Direct acknowledgement on RESET_RSC
*/
CONS_CHECK_REQ (PT_TransId, ST_RscId),
/*Request from SN to AN to start VPCI consistency check procedure
*/
CONS_CHECK_REQ_ACK (PT_TransId, IT_RTMCResult, ST_RscId),
/*Direct acknowledgement  on CONS_CHECK_REQ
*/
CONS_CHECK_END (PT_TransId, ST_RscId),
/*Request from SN to AN to stop VPCI consistency check procedure
*/
CONS_CHECK_END_ACK (PT_TransId, IT_RTMCResult, ST_RscId),
/*Direct acknowledgement  on CONS_CHECK_END
*/
PROTOCOL_ERROR (PT_TransId, IT_RTMCProtErrCause);
/*Reporting of protocol errors to the peer side
*/
```

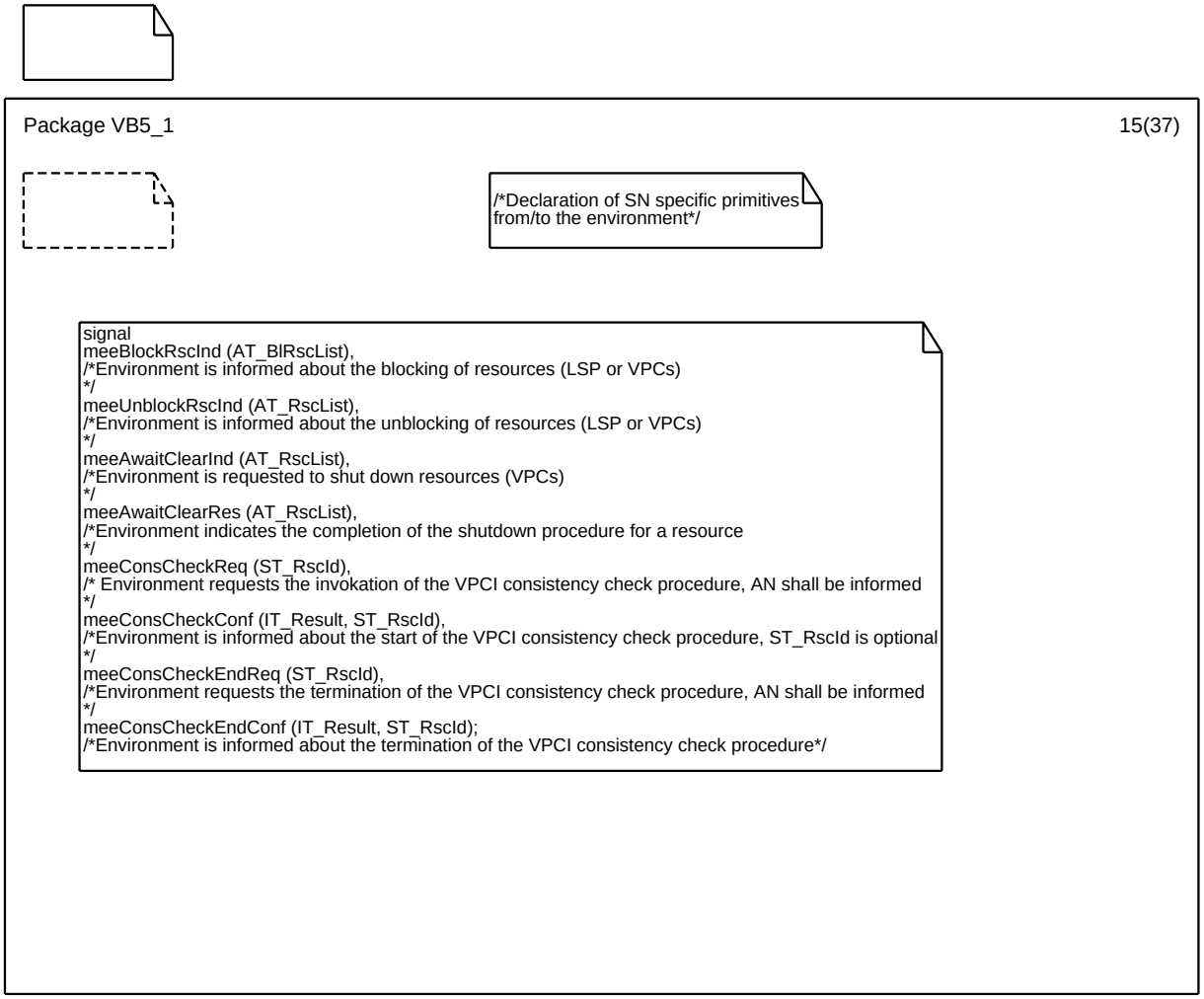

Package VB5_1
12(37)
/* Declaration of primitives from/to the SAAL */
signal
AAL_establish_req,
/*Request from VB5 system management to SAAL to establish SAAL connection to the peer side
*/
AAL_establish_conf,
/*Confirmation from SAAL to VB5 system management of successful SAAL establishment
*/
AAL_establish_ind,
/*Indication from SAAL to VB5 system management that peer side requested SAAL establishment
*/
AAL_release_req,
/*Request from VB5 system management to SAAL to release SAAL connection to the peer side
*/
AAL_release_conf,
/*Confirmation from SAAL to VB5 system management of successful SAAL release
*/
AAL_release_ind;
/*Indication from SAAL to VB5 system management that peer side requested SAAL release*/

```
/*Declaration of common primitives
from/to the environment*/
```

```
signal
meeVerifyLspIdReq,
/*Environment requests the verification of the LSP ID
*/
meeVerifyLspIdConf (IT_Result),
/* Environment is informed about the result of the verify LSP ID procedure
*/
meeResetRscReq (AT_RscList),
/*Environment requests the reset of resources (VPC/LSP), SN shall be informed
*/
meeResetRscConf (IT_Result, AT_RscList),
/* Environment is informed about the result of the reset procedure, AT_RscId is optional
*/
meeResetRscInd (AT_RscList),
/*Environment is informed about the reset of resources (VPC/LSP) by the peer side
*/
meeStartupReq,
/*Environment requests the start-up of the VB5 interface
*/
meeStartupConf (IT_Result),
/* Environment is informed about the result of the startup procedure
*/
meeStartupInd,
/* Environment is informed about the startup of the interface invoked by the other side
*/
meeStopTrafficReq,
/* Environment requests the stopping of all communication to the peer side, SAAL is to be released
*/
meeStopTrafficConf,
/* Environment is informed about the stopping of communication to the peer side
*/
meeLSPFailureInd,
/* Environment is informed about an LSP failure (i.e SAAL failure)
*/
meeErrorInd;
/*Environment is informed about detected protocol errors*/
```

```
/* These primitives are necessary to provision the VPCs of the AN or SN */
signal
meeVPCCreateReq (ST_RscId),
/* Environment requests the creation of a VPC-resource
*/
meeVPCCreateConf,
/* VB5-System reports the result of the creation
*/
meeVPCDeleteReq (ST_RscId),
/* Environment requests the deletion of a VPC-resource
*/
meeVPCDeleteConf;
/* VB5-System reports the result of the deletion
*/
```


Package VB5_1
14(37)
/*Declaration of AN specific primitives from/to the environment*/
signal
meeBlockRscReq (AT_BlRscList),
/*Environment requests the blocking of resources (LSP or VPCs), SN shall be informed
*/
meeBlockRscConf (IT_Result, AT_RscList),
/* Environment is informed about the result of the block procedure, AT_RscList is optional
*/
meeUnblockRscReq (AT_RscList),
/*Environment requests the unblocking of resources (LSP or VPCs), SN shall be informed
*/
meeUnblockRscConf (IT_Result, AT_RscList),
/* Environment is informed about the result of the unblock procedure, AT_RscList is optional
*/
meeAwaitClearReq (AT_RscList),
/*Environment requests the shutting down of resources (VPCs), SN shall be informed
*/
meeAwaitClearConf (IT_Result, AT_RscList),
/* Environment is informed about the progress of the shutdown procedure
*/
meeConsCheckInd (ST_RscId),
/* Environment is informed about the request of invoking a VPCI consistency check procedure
*/
meeConsCheckRes (IT_Result, ST_RscId),
/*Environment responds to the VPCI consistency check request
*/
meeConsCheckEndInd (ST_RscId),
/* Environment is informed about the request of terminating a VPCI consistency check procedure
*/
meeConsCheckEndRes (IT_Result, ST_RscId);
/*Environment responds to the VPCI consistency check end request*/


Package VB5_1
15(37)
/*Declaration of SN specific primitives from/to the environment*/
signal
meeBlockRscInd (AT_BlRscList),
/*Environment is informed about the blocking of resources (LSP or VPCs)
*/
meeUnblockRscInd (AT_RscList),
/*Environment is informed about the unblocking of resources (LSP or VPCs)
*/
meeAwaitClearInd (AT_RscList),
/*Environment is requested to shut down resources (VPCs)
*/
meeAwaitClearRes (AT_RscList),
/*Environment indicates the completion of the shutdown procedure for a resource
*/
meeConsCheckReq (ST_RscId),
/* Environment requests the invokation of the VPCI consistency check procedure, AN shall be informed
*/
meeConsCheckConf (IT_Result, ST_RscId),
/*Environment is informed about the start of the VPCI consistency check procedure, ST_RscId is optional
*/
meeConsCheckEndReq (ST_RscId),
/*Environment requests the termination of the VPCI consistency check procedure, AN shall be informed
*/
meeConsCheckEndConf (IT_Result, ST_RscId);
/*Environment is informed about the termination of the VPCI consistency check procedure*/

```
/*Declaration of common primitives
from/to CPROT entitiy*/
```

```
signal
MDU_reset_req (IT_IntTransId, AT_RscList),
/*CPROT entity is requested to send a RESET_RSC message for the VPCs and/or LSP to the SN
*/
MDU_reset_ind (PT_CprotPID, AT_RscList),
/*CPROT entity indicates the reception of a RESET_RSC message for the VPCs and/or LSP from the SN
*/
MDU_reset_res (PT_CprotPID, IT_RTMCResult, AT_RscList),
/*CPROT entity is requested to send a RESET_RSC_ACK message to the SN
*/
MDU_reset_conf (IT_IntTransId, IT_Result, AT_RscList),
/*CPROT entity confirms the execution of the reset procedure
*/
MDU_LSP_ID_req,
/*CPROT entity is requested to send a REQUEST_LSPID message to the SN
*/
MDU_LSP_ID_ind (PT_CprotPID),
/*CPROT entity indicates the reception of  a REQUEST_LSPID message from the SN
*/
MDU_LSP_ID_res (PT_CprotPID, ST_RscId),
/*CPROT entity is requested to send a LSPID message for the LSP to the SN
*/
MDU_LSP_ID_conf (IT_Result, ST_RscId),
/*CPROT entity confirms the receipt of the LSPID message
*/
MDU_abort_req,
/*CPROT entity is requested to abort the transaction immediately
*/
MDU_resrsc_prc_started,
/*System Management process RESET RESOURCES announces its PID to AN_CPROT entity
*/
MDU_lspid_prc_started,
/*System Management process LSPID CHECK announces its PID to AN_CPROT entity
*/
MDU_cons_prc_started,
/*System Management process VPCI CONSISTENCY CHECK announces its PID to AN_CPROT entity
*/
MDU_status_prc_started,
/*System Management process STATUS MGT announces its PID to AN_CPROT entity
*/
MDU_start_traffic,
/*CTR announces that traffic to the remote side is possible
*/
MDU_stop_traffic;
/*CTR announces that traffic to the remote side is not possible*/
```

```
/*Primitives to the RTMC protocol entity representing erroneous VB5 messages */
signal
unknown_message(PT_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*/
```

/*Declaration of primitives
from/to AN_CPROT entitiy*/

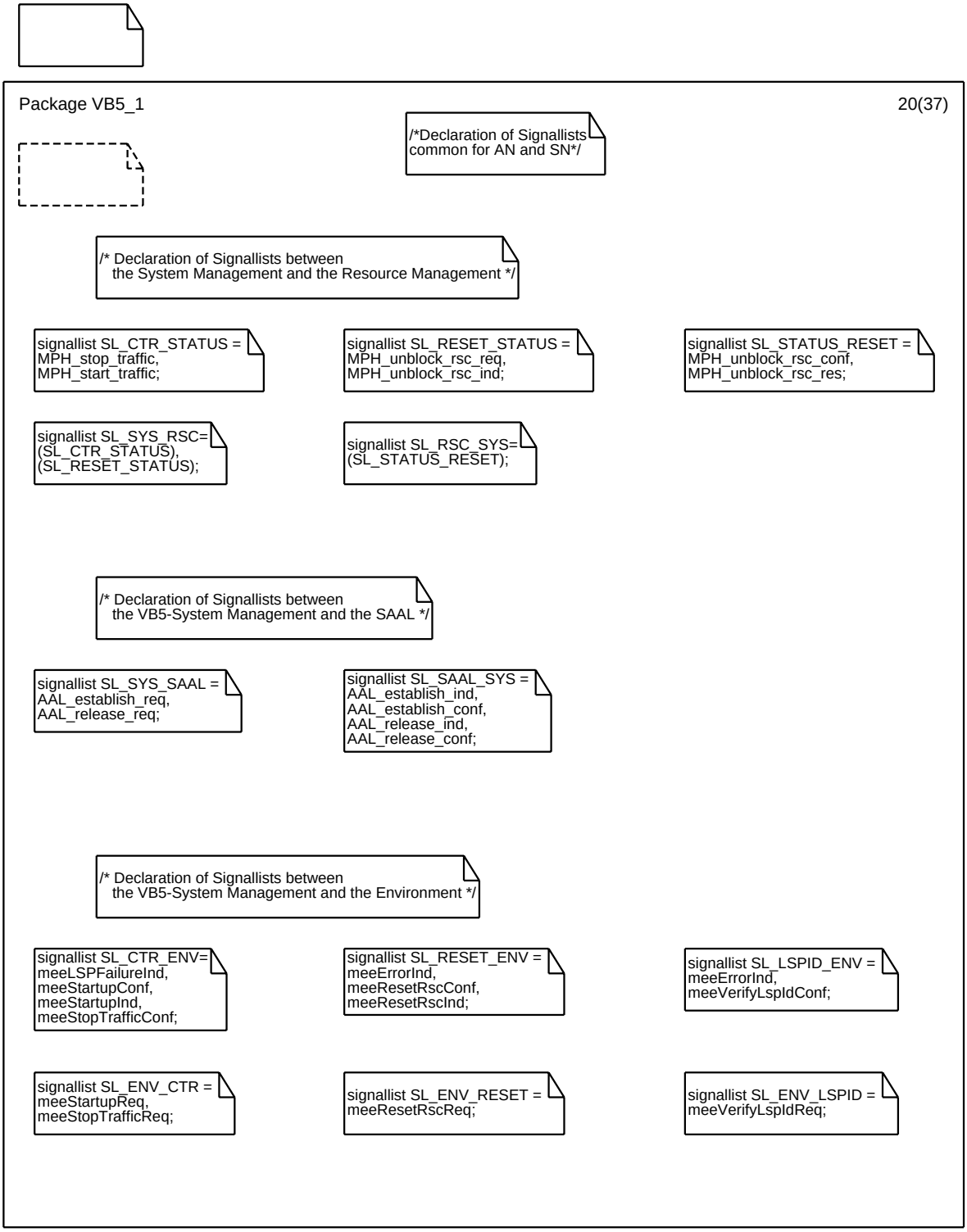
```
signal
MDU_unblock_req (IT_IntTransId, AT_RscList),
/*CPROT entity is requested to send an UNBLOCK_RSC message to the SN
*/
MDU_unblock_conf (IT_IntTransId, IT_Result, AT_RscList),
/*CPROT entity confirms the execution of the unblock procedure
*/
MDU_block_req (IT_IntTransId, AT_RTMCBlRscList),
/*CPROT entity is requested to send a BLOCK_RSC message for VPC(s) to the SN
*/
MDU_block_conf (IT_IntTransId, IT_Result, AT_RscList),
/*CPROT entity confirms the execution of the VPC block procedure
*/
MDU_await_clear_req (IT_IntTransId, AT_RscList),
/*CPROT entity is requested to send a SHUTDOWN message for VPC(s) to the SN
*/
MDU_await_clear_conf (IT_IntTransId, IT_Result, AT_RscList),
/*CPROT entity confirms the execution of the VPC shutdown procedure
*/
MDU_cons_check_ind (PT_CprotPID, ST_RscId),
/*CPROT entity indicates the reception of  a CONSISTECNY_CHECK_REQ message from the SN
*/
MDU_cons_check_res (PT_CprotPID, IT_RTMCResult, ST_RscId),
/*CPROT entity is requested to send  a CONSISTECNY_CHECK_REQ_ACK  message to the SN
*/
MDU_cons_check_end_ind (PT_CprotPID, ST_RscId),
/*CPROT entity indicates the reception of  a CONSISTECNY_CHECK_END message from the SN
*/
MDU_cons_check_end_res (PT_CprotPID, IT_RTMCResult, ST_RscId);
/*CPROT entity is requested to send  a CONSISTECNY_CHECK_END_ACK  message to the SN
*/
```

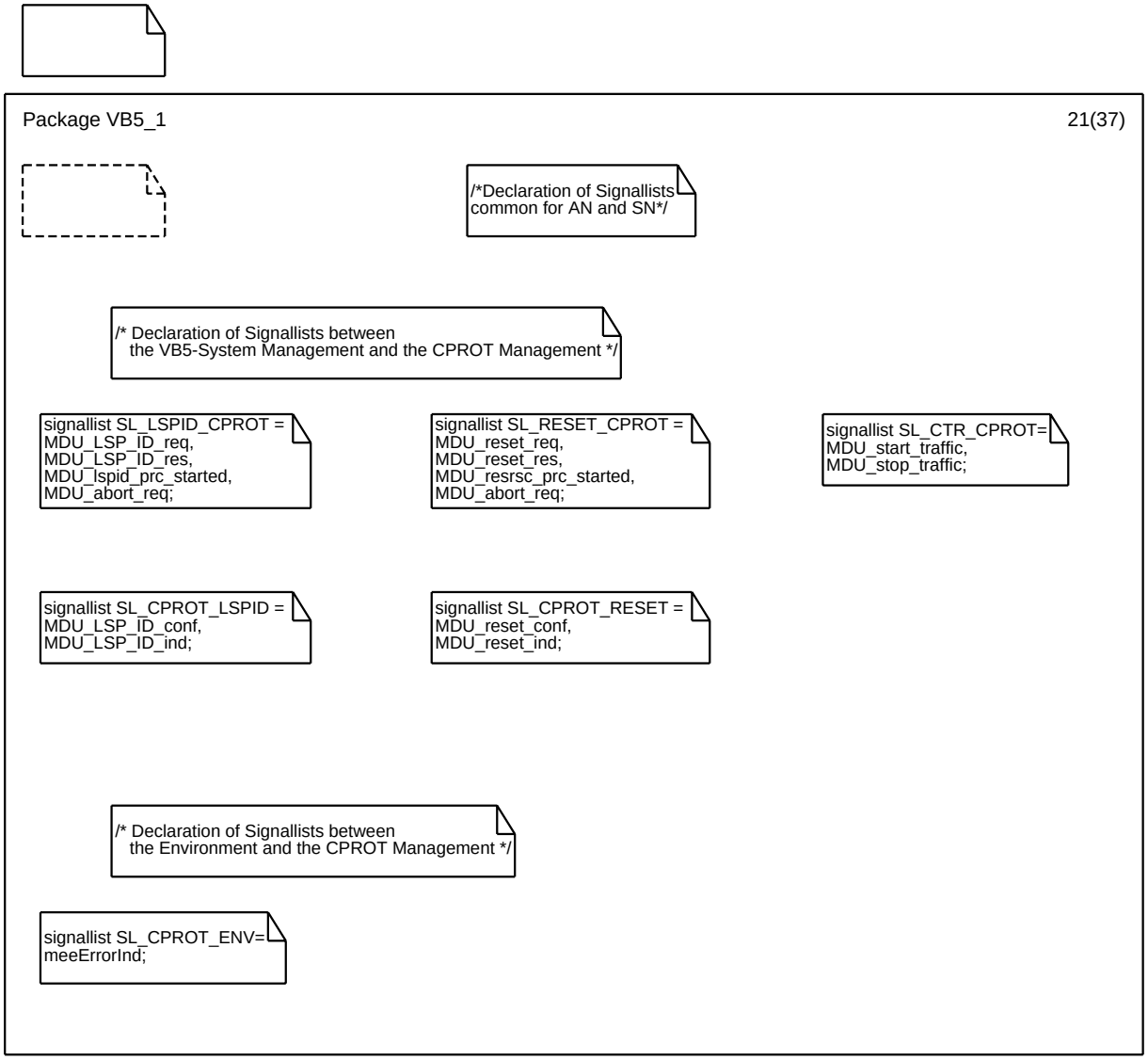
/*Declaration of primitives
from/to SN_CPROT entitiy*/

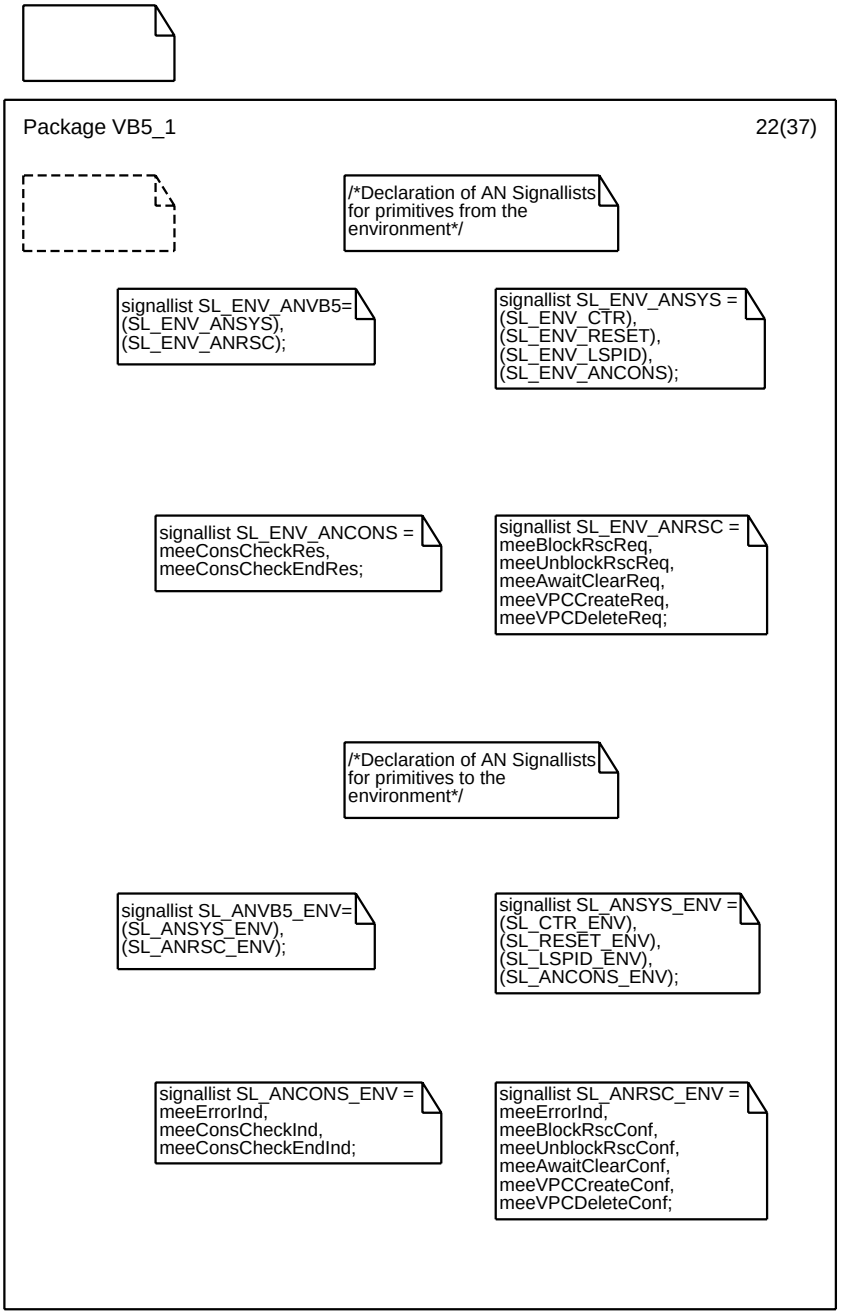
```
signal
MDU_unblock_ind (PT_CprotPID, AT_RscList),
/*CPROT entity indicates the receipt of a UNBLOCK_RSC message
*/
MDU_unblock_res (PT_CprotPID, IT_RTMCResult, AT_RscList),
/*CPROT entity is requested to send a UNBLOCK_RSC_ACK message
*/
MDU_block_ind (PT_CprotPID, AT_RTMCBlRscList),
/*CPROT entity indicates the receipt of a BLOCK_RSC message
*/
MDU_block_res (PT_CprotPID, IT_RTMCResult, AT_RscList),
/* CPROT entity is requested to send a BLOCK_RSC_ACK message
*/
MDU_await_clear_ind (PT_CprotPID, AT_RscList),
/*CPROT entity indicates the receipt of an AWAIT CLEAR message
*/
MDU_await_clear_res (AT_RscList),
/*CPROT entity is requested to send an AWAIT_CLEAR_COMP message to the AN
*/
MDU_await_clear_ack (PT_CprotPID, IT_RTMCResult, AT_RscList),
/*CPROT entity is requested to send an AWAIT_CLEAR_ACK message for VPCs to the AN
*/
MDU_cons_check_req (ST_RscId),
/*CPROT entity is requested to send a CONS_CHECK_REQ message to the AN
*/
MDU_cons_check_conf (IT_Result, ST_RscId),
/*CPROT entity indicates the reception of  a CONS_CHECK_ACK  message from the AN
*/
MDU_cons_check_end_req (ST_RscId),
/*CPROT entity is requested to send a CONS_CHECK_END message to the AN
*/
MDU_cons_check_end_conf (IT_Result, ST_RscId);
/*CPROT entity indicates the reception of  a CONS_CHECK_END_ACK  message from the AN
*/
```

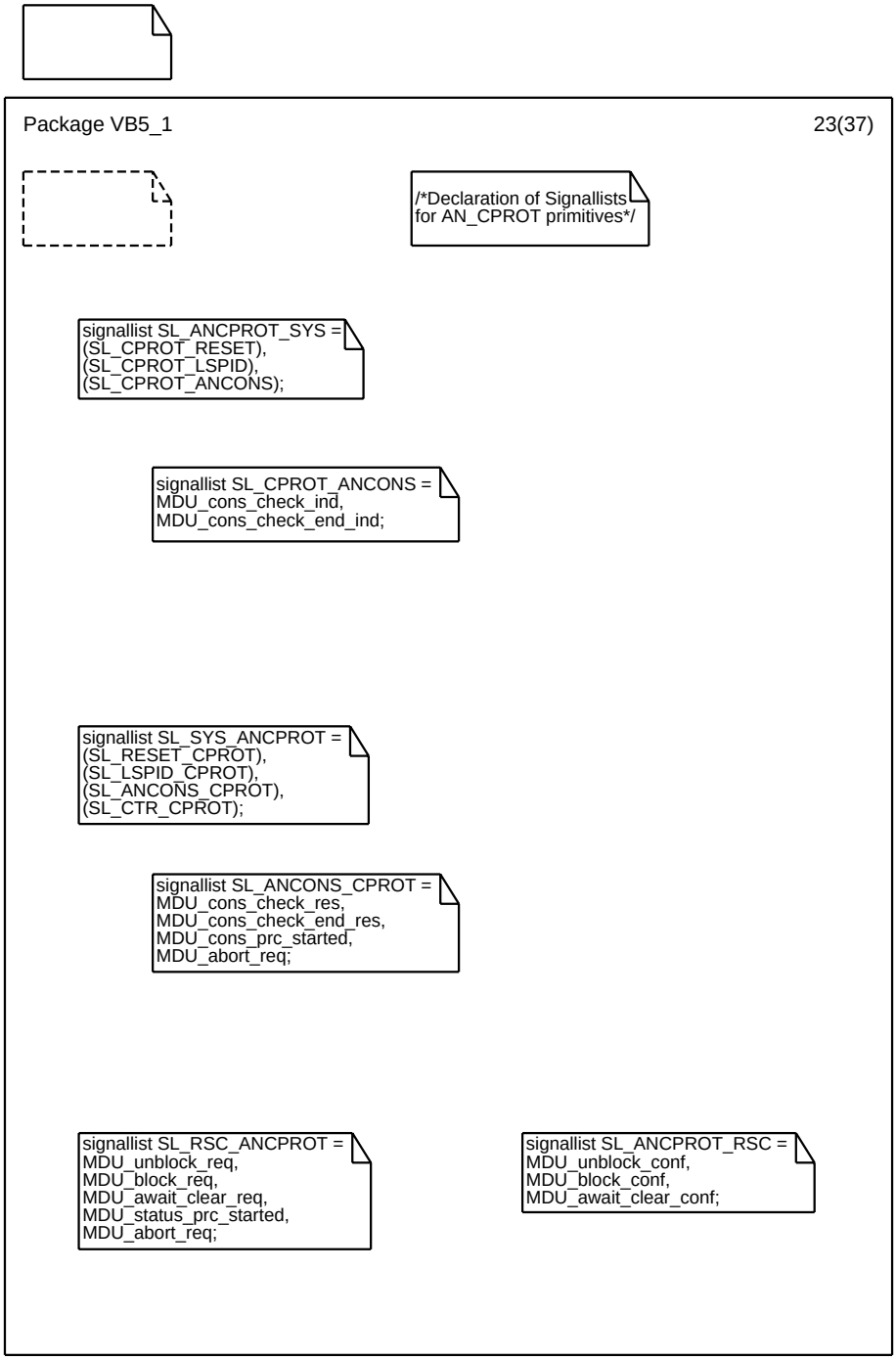
```
/* Declaration of common primitives between
   the System Management and the Resource Management */
```

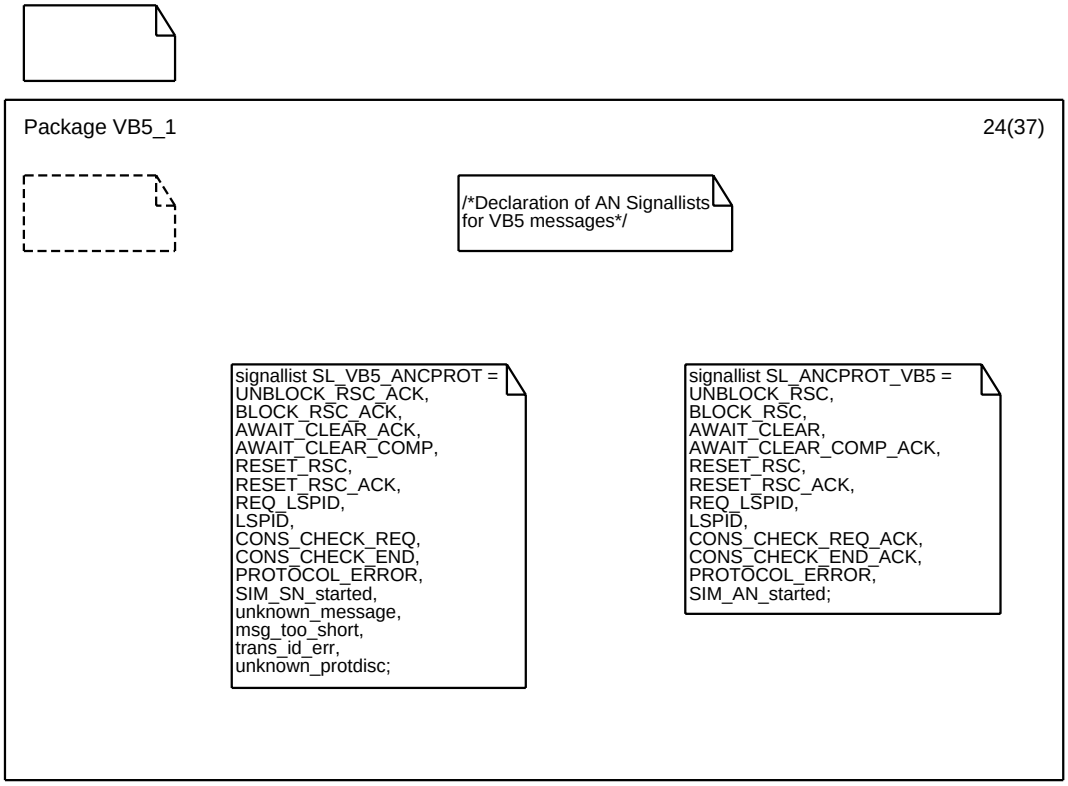
```
signal
MPH_unblock_rsc_req (IT_IntTransId, AT_RscList),
/*RESET-process requests the unblocking of VPCFSMs and/or the LSPFSM
*/
MPH_unblock_rsc_conf (IT_IntTransId),
/*STATUS-process confirms the unblocking of VPCFSMs and/or the LSPFSM
*/
MPH_unblock_rsc_ind (AT_RscList),
/*RESET-process indicates the unblocking of VPCFSMs and/or the LSPFSM
*/
MPH_unblock_rsc_res,
/*STATUS-process confirms the unblocking of VPCFSMs and/or the LSPFSM
*/
MPH_stop_traffic,
/*CTR-process requests the stop of the status management process (e.g. in case of SAAL failure)
*/
MPH_start_traffic;
/*CTR-process requests the start of the status management process (e.g.after startup)
*/
```

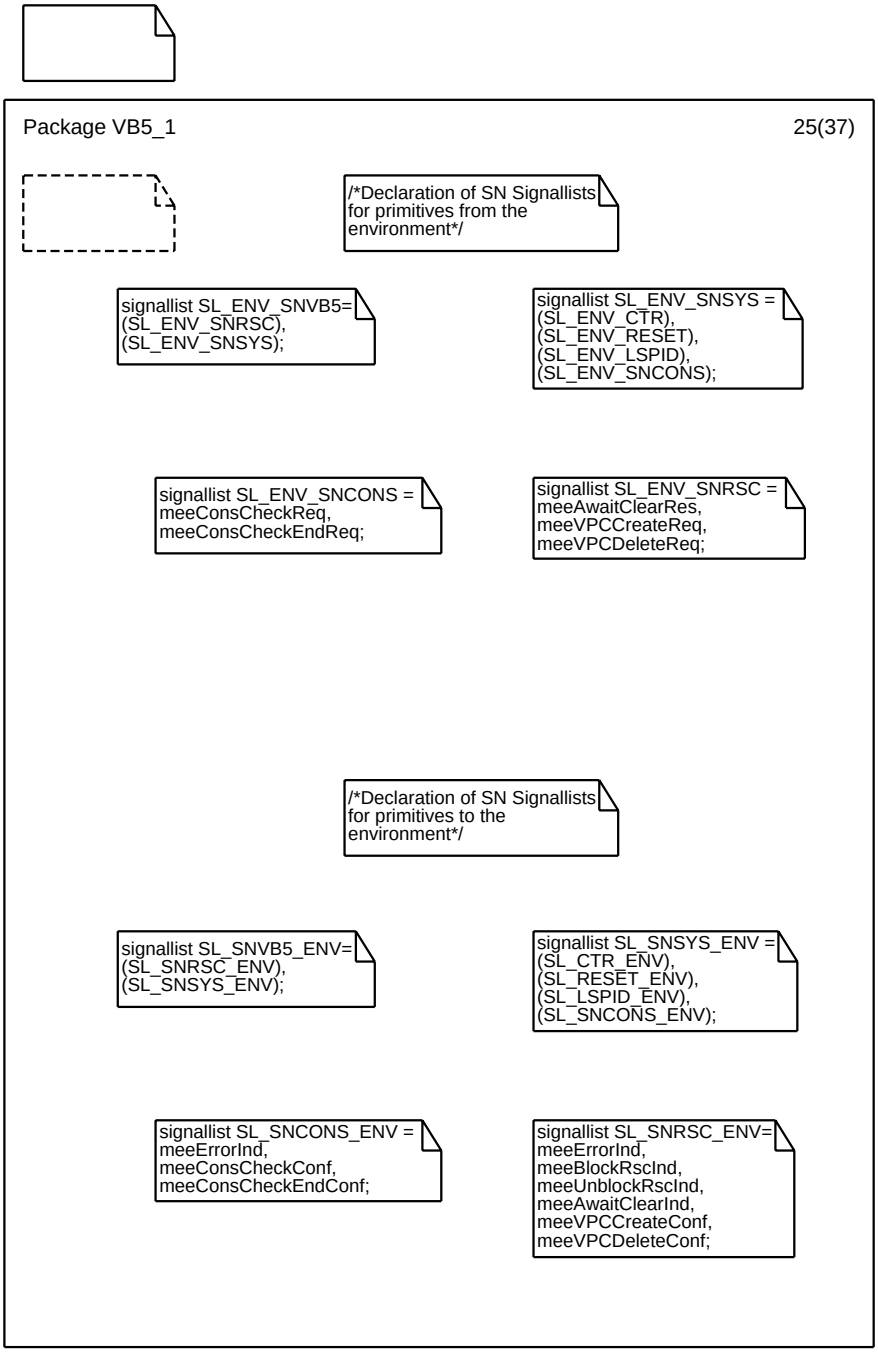

Package VB5_1
20(37)
/*Declaration of Signallists common for AN and SN*/
/* Declaration of Signallists between the System Management and the Resource Management */
signallist SL_CTR_STATUS = MPH_stop_traffic, MPH_start_traffic;
signallist SL_RESET_STATUS = MPH_unblock_rsc_req, MPH_unblock_rsc_ind;
signallist SL_STATUS_RESET = MPH_unblock_rsc_conf, MPH_unblock_rsc_res;
signallist SL_SYS_RSC= (SL_CTR_STATUS), (SL_RESET_STATUS);
signallist SL_RSC_SYS= (SL_STATUS_RESET);
/* Declaration of Signallists between the VB5-System Management and the SAAL */
signallist SL_SYS_SAAL = AAL_establish_req, AAL_release_req;
signallist SL_SAAL_SYS = AAL_establish_ind, AAL_establish_conf, AAL_release_ind, AAL_release_conf;
/* Declaration of Signallists between the VB5-System Management and the Environment */
signallist SL_CTR_ENV= meeLSPFailureInd, meeStartupConf, meeStartupInd, meeStopTrafficConf;
signallist SL_RESET_ENV = meeErrorInd, meeResetRscConf, meeResetRscInd;
signallist SL_LSPID_ENV = meeErrorInd, meeVerifyLspIdConf;
signallist SL_ENV_CTR = meeStartupReq, meeStopTrafficReq;
signallist SL_ENV_RESET = meeResetRscReq;
signallist SL_ENV_LSPID = meeVerifyLspIdReq;

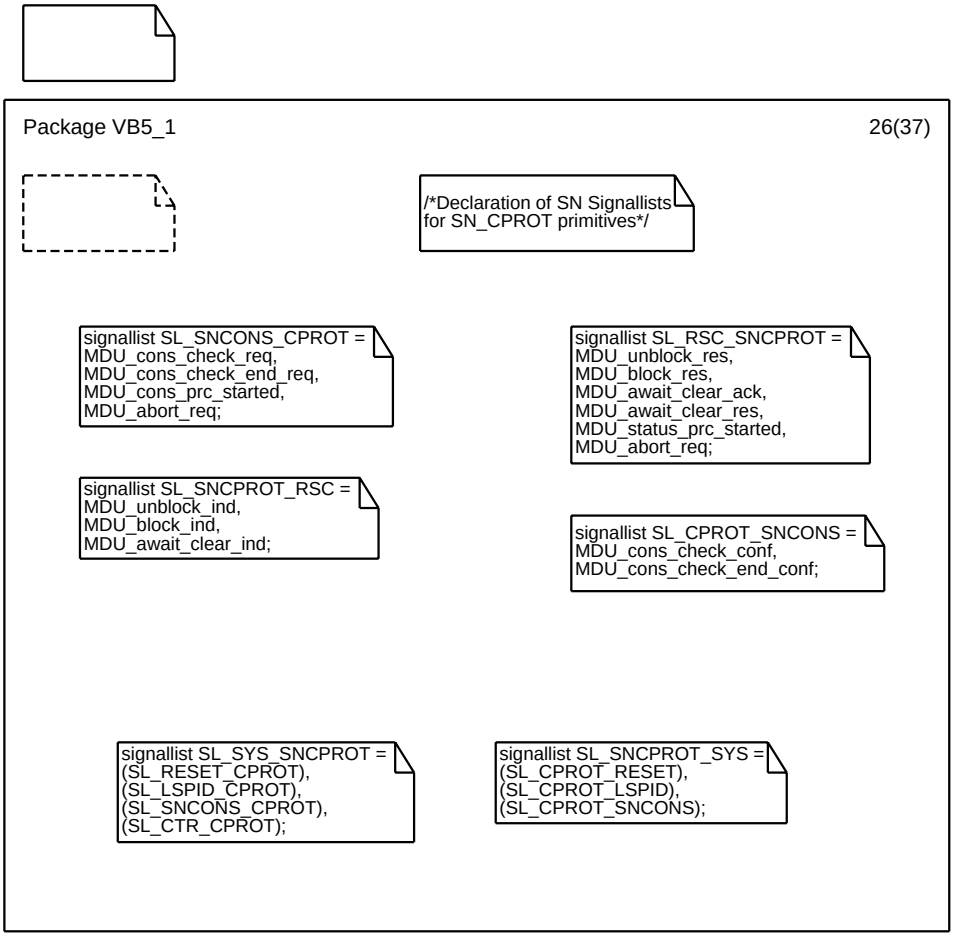

Package VB5_1
21(37)
/*Declaration of Signallists common for AN and SN*/
/* Declaration of Signallists between the VB5-System Management and the CPROT Management */
signallist SL_LSPID_CPROT = MDU_LSP_ID_req, MDU_LSP_ID_res, MDU_lspid_prc_started, MDU_abort_req;
signallist SL_RESET_CPROT = MDU_reset_req, MDU_reset_res, MDU_resrsc_prc_started, MDU_abort_req;
signallist SL_CTR_CPROT= MDU_start_traffic, MDU_stop_traffic;
signallist SL_CPROT_LSPID = MDU_LSP_ID_conf, MDU_LSP_ID_ind;
signallist SL_CPROT_RESET = MDU_reset_conf, MDU_reset_ind;
/* Declaration of Signallists between the Environment and the CPROT Management */
signallist SL_CPROT_ENV= meeErrorInd;

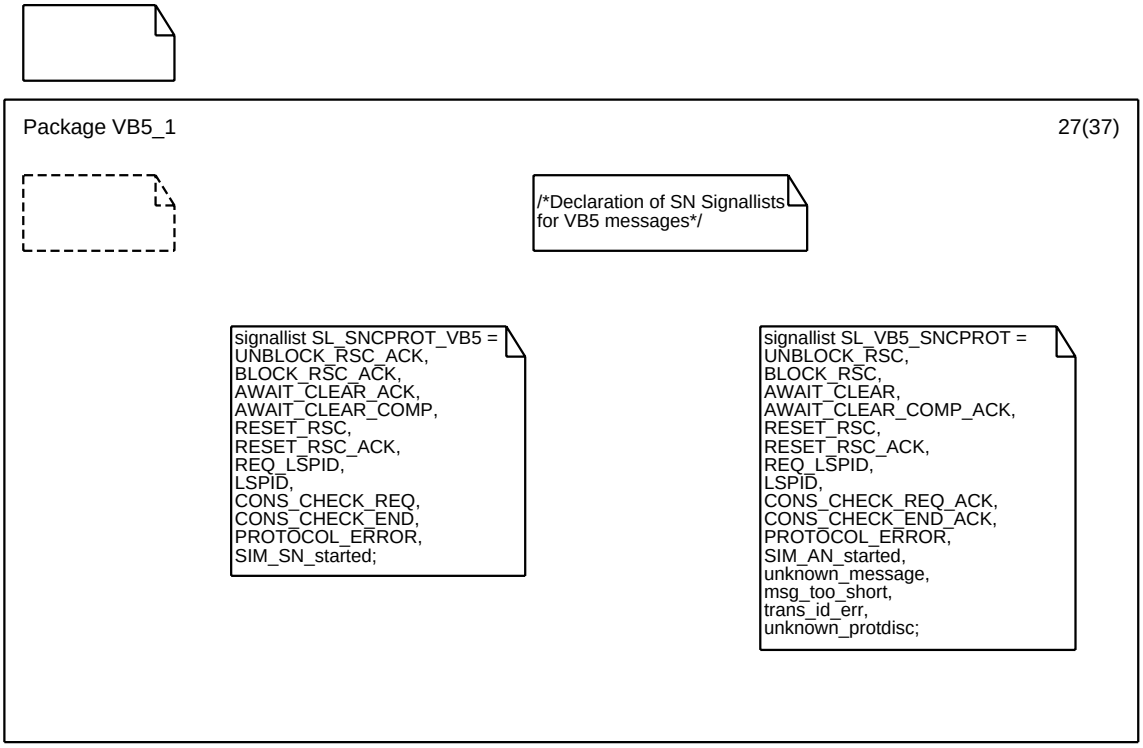

Package VB5_1
22(37)
/*Declaration of AN Signallists for primitives from the environment*/
signallist SL_ENV_ANVB5= (SL_ENV_ANSYS), (SL_ENV_ANRSC);
signallist SL_ENV_ANSYS = (SL_ENV_CTR), (SL_ENV_RESET), (SL_ENV_LSPID), (SL_ENV_ANCONS);
signallist SL_ENV_ANCONS = meeConsCheckRes, meeConsCheckEndRes;
signallist SL_ENV_ANRSC = meeBlockRscReq, meeUnblockRscReq, meeAwaitClearReq, meeVPCCreateReq, meeVPCDeleteReq;
/*Declaration of AN Signallists for primitives to the environment*/
signallist SL_ANVB5_ENV= (SL_ANSYS_ENV), (SL_ANRSC_ENV);
signallist SL_ANSYS_ENV = (SL_CTR_ENV), (SL_RESET_ENV), (SL_LSPID_ENV), (SL_ANCONS_ENV);
signallist SL_ANCONS_ENV = meeErrorInd, meeConsCheckInd, meeConsCheckEndInd;
signallist SL_ANRSC_ENV = meeErrorInd, meeBlockRscConf, meeUnblockRscConf, meeAwaitClearConf, meeVPCCreateConf, meeVPCDeleteConf;


Package VB5_1
23(37)
/*Declaration of Signallists
for AN_CPROT primitives*/
signallist SL_ANCPROT_SYS =
(SL_CPROT_RESET),
(SL_CPROT_LSPID),
(SL_CPROT_ANCONS);
signallist SL_CPROT_ANCONS =
MDU_cons_check_ind,
MDU_cons_check_end_ind;
signallist SL_SYS_ANCPROT =
(SL_RESET_CPROT),
(SL_LSPID_CPROT),
(SL_ANCONS_CPROT),
(SL_CTR_CPROT);
signallist SL_ANCONS_CPROT =
MDU_cons_check_res,
MDU_cons_check_end_res,
MDU_cons_prc_started,
MDU_abort_req;
signallist SL_RSC_ANCPROT =
MDU_unblock_req,
MDU_block_req,
MDU_await_clear_req,
MDU_status_prc_started,
MDU_abort_req;
signallist SL_ANCPROT_RSC =
MDU_unblock_conf,
MDU_block_conf,
MDU_await_clear_conf;


Package VB5_1
24(37)
/*Declaration of AN Signallists
for VB5 messages*/
signallist SL_VB5_ANCPROT =
UNBLOCK_RSC_ACK,
BLOCK_RSC_ACK,
AWAIT_CLEAR_ACK,
AWAIT_CLEAR_COMP,
RESET_RSC,
RESET_RSC_ACK,
REQ_LSPID,
LSPID,
CONS_CHECK_REQ,
CONS_CHECK_END,
PROTOCOL_ERROR,
SIM_SN_started,
unknown_message,
msg_too_short,
trans_id_err,
unknown_protdisc;
signallist SL_ANCPROT_VB5 =
UNBLOCK_RSC,
BLOCK_RSC,
AWAIT_CLEAR,
AWAIT_CLEAR_COMP_ACK,
RESET_RSC,
RESET_RSC_ACK,
REQ_LSPID,
LSPID,
CONS_CHECK_REQ_ACK,
CONS_CHECK_END_ACK,
PROTOCOL_ERROR,
SIM_AN_started;


Package VB5_1
25(37)
/*Declaration of SN Signallists for primitives from the environment*/
signallist SL_ENV_SNVB5= (SL_ENV_SNRSC), (SL_ENV_SNSYS);
signallist SL_ENV_SNSYS = (SL_ENV_CTR), (SL_ENV_RESET), (SL_ENV_LSPID), (SL_ENV_SNCONS);
signallist SL_ENV_SNCONS = meeConsCheckReq, meeConsCheckEndReq;
signallist SL_ENV_SNRSC = meeAwaitClearRes, meeVPCCreateReq, meeVPCDeleteReq;
/*Declaration of SN Signallists for primitives to the environment*/
signallist SL_SNVB5_ENV= (SL_SNRSC_ENV), (SL_SNSYS_ENV);
signallist SL_SNSYS_ENV = (SL_CTR_ENV), (SL_RESET_ENV), (SL_LSPID_ENV), (SL_SNCONS_ENV);
signallist SL_SNCONS_ENV = meeErrorInd, meeConsCheckConf, meeConsCheckEndConf;
signallist SL_SNRSC_ENV= meeErrorInd, meeBlockRscInd, meeUnblockRscInd, meeAwaitClearInd, meeVPCCreateConf, meeVPCDeleteConf;

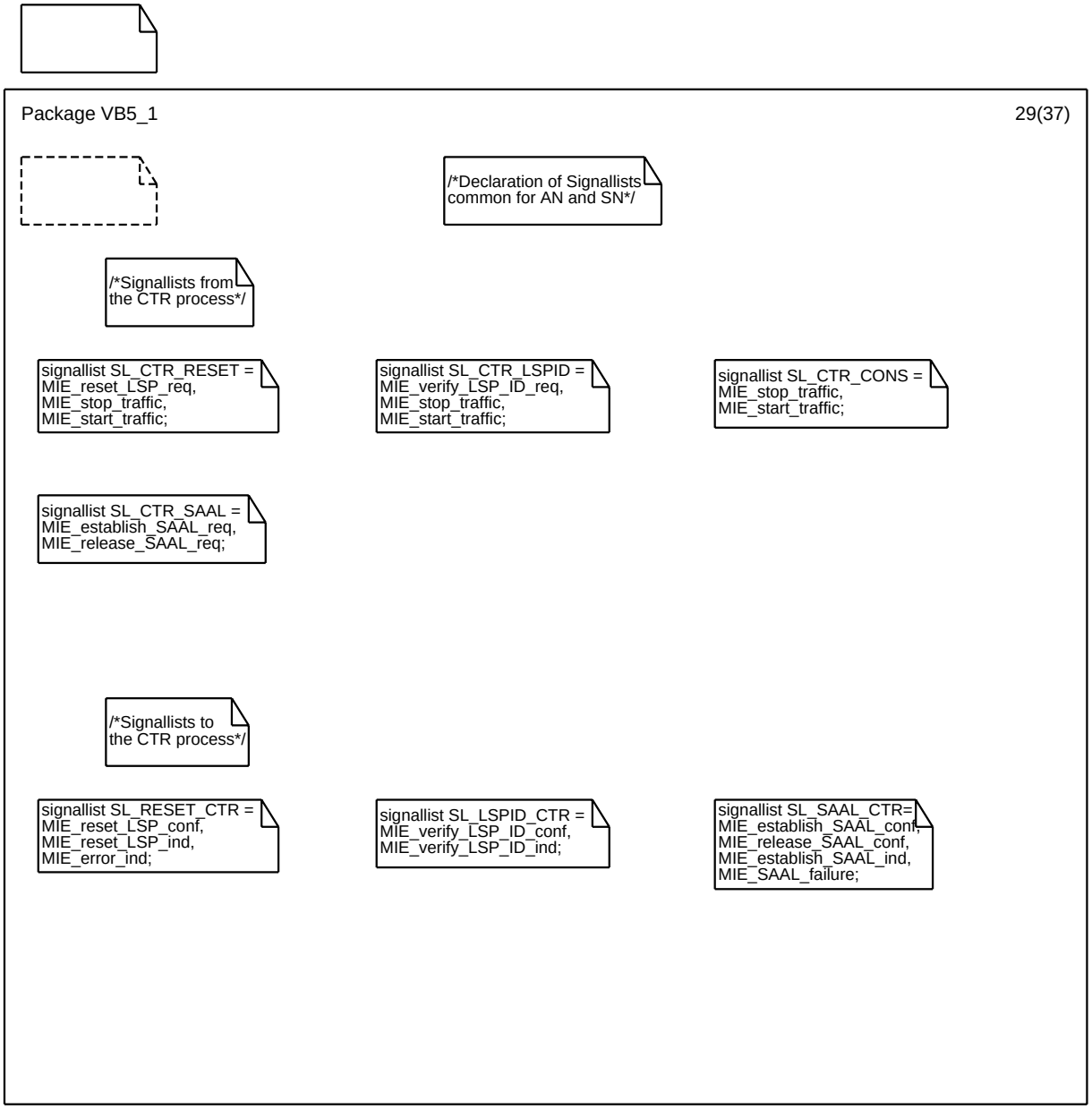

Package VB5_1
26(37)
/*Declaration of SN Signallists
for SN_CPROT primitives*/
signallist SL_SNCONS_CPROT =
MDU_cons_check_req,
MDU_cons_check_end_req,
MDU_cons_prc_started,
MDU_abort_req;
signallist SL_RSC_SNCPROT =
MDU_unblock_res,
MDU_block_res,
MDU_await_clear_ack,
MDU_await_clear_res,
MDU_status_prc_started,
MDU_abort_req;
signallist SL_SNCPROT_RSC =
MDU_unblock_ind,
MDU_block_ind,
MDU_await_clear_ind;
signallist SL_CPROT_SNCONS =
MDU_cons_check_conf,
MDU_cons_check_end_conf;
signallist SL_SYS_SNCPROT =
(SL_RESET_CPROT),
(SL_LSPID_CPROT),
(SL_SNCONS_CPROT),
(SL_CTR_CPROT);
signallist SL_SNCPROT_SYS =
(SL_CPROT_RESET),
(SL_CPROT_LSPID),
(SL_CPROT_SNCONS);


Package VB5_1
27(37)
/*Declaration of SN Signallists
for VB5 messages*/
signallist SL_SNCPROT_VB5 =
UNBLOCK_RSC_ACK,
BLOCK_RSC_ACK,
AWAIT_CLEAR_ACK,
AWAIT_CLEAR_COMP,
RESET_RSC,
RESET_RSC_ACK,
REQ_LSPID,
LSPID,
CONS_CHECK_REQ,
CONS_CHECK_END,
PROTOCOL_ERROR,
SIM_SN_started;
signallist SL_VB5_SNCPROT =
UNBLOCK_RSC,
BLOCK_RSC,
AWAIT_CLEAR,
AWAIT_CLEAR_COMP_ACK,
RESET_RSC,
RESET_RSC_ACK,
REQ_LSPID,
LSPID,
CONS_CHECK_REQ_ACK,
CONS_CHECK_END_ACK,
PROTOCOL_ERROR,
SIM_AN_started,
unknown_message,
msg_too_short,
trans_id_err,
unknown_protdisc;

```
/* Declaration of
Internal System Management Primitives */
```

```
signal
MIE_reset_LSP_req,
/*CTR-process requests the reset of the VPCs and/or LSP (e.g. during startup)
*/
MIE_reset_LSP_conf (IT_Result),
/*RESET-process confirms the reset of the VPCs and/or LSP (e.g. during startup)
*/
MIE_reset_LSP_ind,
/*RESET-process indicates the reset of the VPCs and/or LSP (e.g. during startup)
*/
MIE_verify_LSP_ID_req,
/*CTR-process requests the verification of the LSP ID (e.g. during startup)
*/
MIE_verify_LSP_ID_conf (IT_Result),
/*LSPID-process confirms the verification of the LSP ID (e.g. during startup)
*/
MIE_verify_LSP_ID_ind,
/*LSPID-process indicates the verification of the LSP ID (e.g. during startup)
*/
MIE_error_ind,
/*error indication towards CTR process
*/
MIE_stop_traffic,
/*CTR-process requests the stop of all system management processes (e.g. in case of SAAL failure)
*/
MIE_start_traffic;
/*CTR-process requests the start of all system management processes (e.g.after startup)
*/
```

```
signal
MIE_establish_SAAL_req,
/*CTR-process requests the establishment of the SAAL connection
*/
MIE_establish_SAAL_conf,
/*SAAL-process confirms the establishment of the SAAL connection
*/
MIE_establish_SAAL_ind,
/*SAAL-process indicates the establishment of the SAAL connection
*/
MIE_release_SAAL_req,
/*CTR-process requests the release of the SAAL connection
*/
MIE_release_SAAL_conf,
/*SAAL-process confirms the release of the SAAL connection
*/
MIE_SAAL_failure;
/*CTR-process is informed about detected SAAL failure
*/
```


Package VB5_1
29(37)
/*Declaration of Signallists common for AN and SN*/
/*Signallists from the CTR process*/
signallist SL_CTR_RESET = MIE_reset_LSP_req, MIE_stop_traffic, MIE_start_traffic;
signallist SL_CTR_LSPID = MIE_verify_LSP_ID_req, MIE_stop_traffic, MIE_start_traffic;
signallist SL_CTR_CONS = MIE_stop_traffic, MIE_start_traffic;
signallist SL_CTR_SAAL = MIE_establish_SAAL_req, MIE_release_SAAL_req;
/*Signallists to the CTR process*/
signallist SL_RESET_CTR = MIE_reset_LSP_conf, MIE_reset_LSP_ind, MIE_error_ind;
signallist SL_LSPID_CTR = MIE_verify_LSP_ID_conf, MIE_verify_LSP_ID_ind;
signallist SL_SAAL_CTR= MIE_establish_SAAL_conf, MIE_release_SAAL_conf, MIE_establish_SAAL_ind, MIE_SAAL_failure;

```
/* Declaration of primitives, internal to the Resource Management */
```

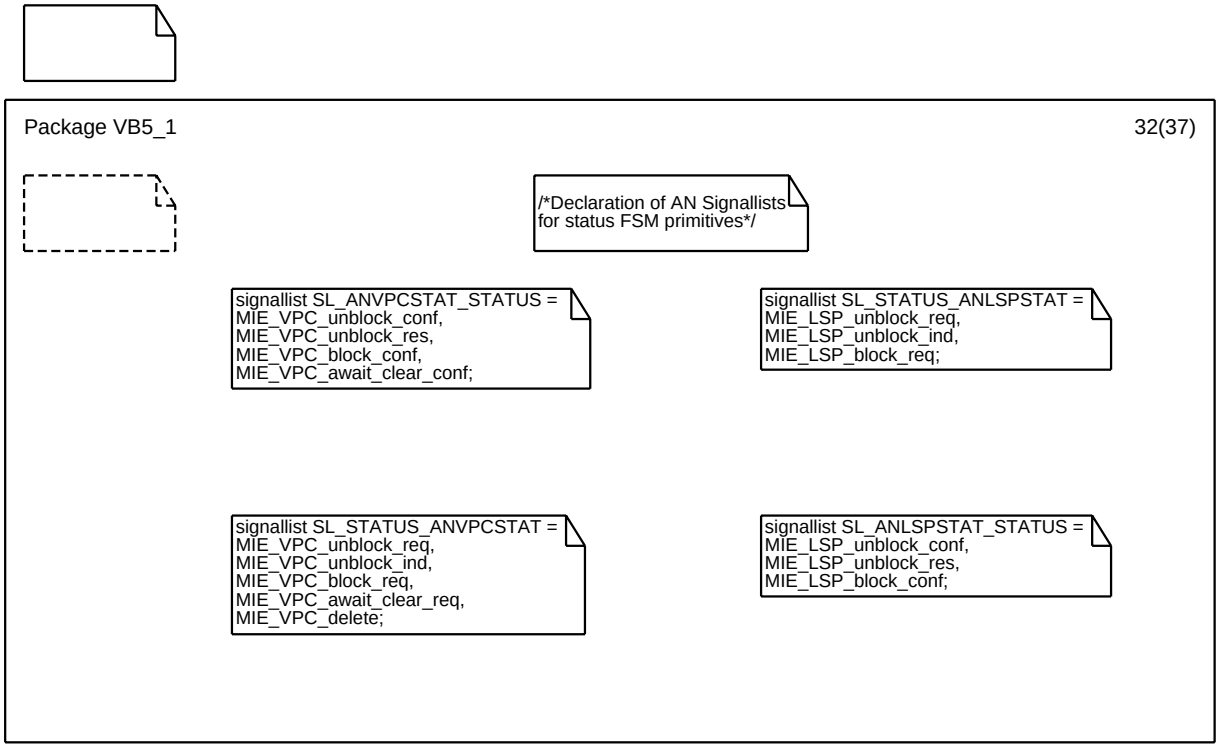
```
/* MIE Primitive, common for AN and SN */
/**/
signal
MIE_VPC_unblock_req (ST_RscId),
/*VPC STATFSM is requested to reset a blocking reason
*/
MIE_VPC_unblock_conf (ST_RscId),
/*VPC STATFSM confirms the resetting of a blocking reason
*/
MIE_VPC_unblock_ind (ST_RscId),
/*Indication to SN_VPCSTAT process that all blocking reason have been reset in the AN
*/
MIE_VPC_unblock_res (ST_RscId),
/* SN_VPCSTAT process confirms the resetting of all blocking reasons
*/
MIE_LSP_unblock_req (ST_RscId),
/*LSP STATFSM is requested to reset a blocking reason
*/
MIE_LSP_unblock_conf (ST_RscId),
/*LSP STATFSM confirms the resetting of a blocking reason
*/
MIE_LSP_unblock_ind (ST_RscId),
/*Indication to SN_LSPSTAT process that all blocking reason have been reset in the AN
*/
MIE_LSP_unblock_res (ST_RscId),
/* SN_VPCSTAT process confirms the resetting of all blocking reasons
*/
MIE_VPC_delete;
/* AN_STATUS_MGT or SN_STATUS_MGT informs VPC_STAT to stop */
```

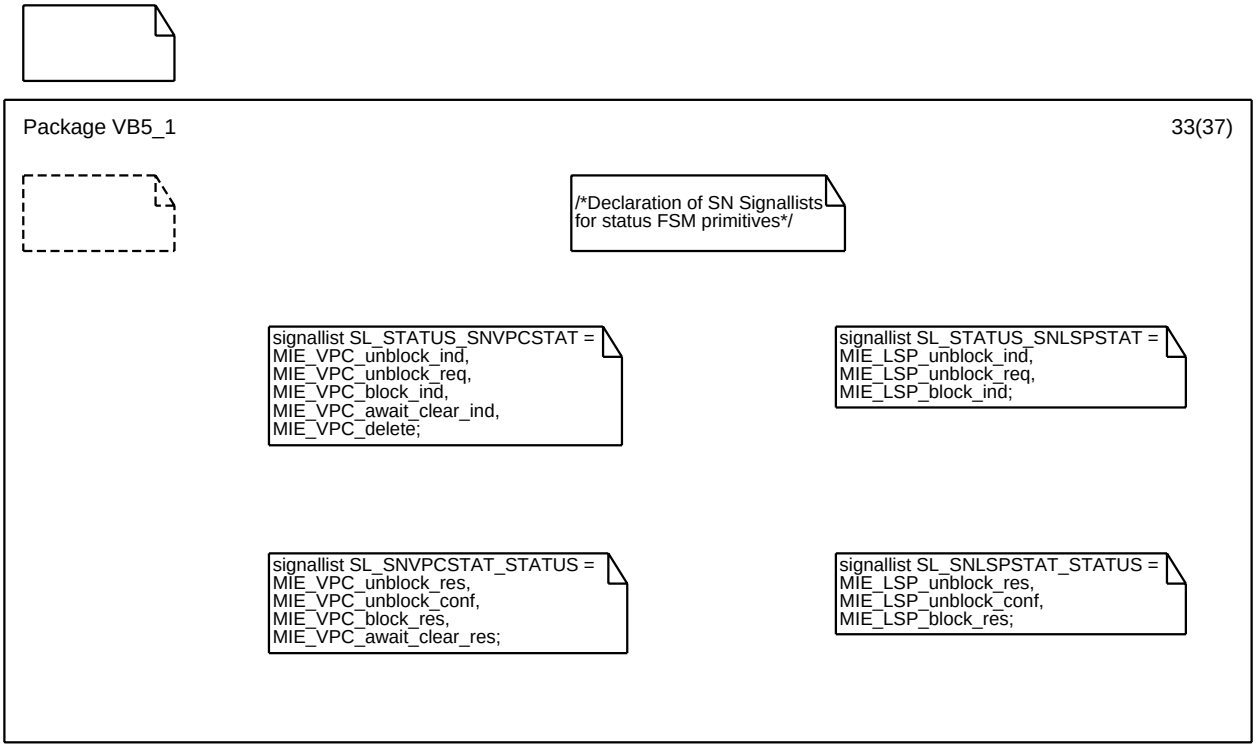
```
/*Declaration of primitives from/to
AN_LSPSTAT and AN_VPCSTAT processes*/
```

```
signal
MIE_VPC_block_req (ST_RscId, IT_Reason),
/*VPC STATFSM is requested to set a blocking reason
*/
MIE_VPC_block_conf (ST_RscId, ST_RTMCReason),
/*VPC STATFSM confirms the setting of a blocking reason
*/
MIE_VPC_await_clear_req  (ST_RscId),
/*VPC STATFSM is requested to enter the shutdown state
*/
MIE_VPC_await_clear_conf (ST_RscId),
/*VPC STATFSM confirms the transition to the shutdown state
*/
MIE_LSP_block_req (ST_RscId, IT_Reason),
/*LSP STATFSM is requested to set a blocking reason
*/
MIE_LSP_block_conf (ST_RscId, ST_RTMCReason);
/*LSP STATFSM confirms the setting of a blocking reason*/
```

```
/*Declaration of primitives from/to
SN_LSPSTAT and SN_VPCSTAT processes*/
```

```
signal
MIE_VPC_block_ind (IT_Reason),
/*Indication to SN_VPCSTAT process that a blocking reason has been set in the AN
*/
MIE_VPC_block_res,
/*VPC STATFSM confirms the setting of a blocking reason
*/
MIE_VPC_await_clear_ind,
/*VPC STATFSM is requested to enter the shutdown state
*/
MIE_VPC_await_clear_res,
/*VPC STATFSM confirms the transition to the shutdown state
*/
MIE_LSP_block_ind (IT_Reason),
/*Indication to SN_VPCSTAT process that a blocking reason has been set in the AN
*/
MIE_LSP_block_res;
/*VPC STATFSM confirms the setting of a blocking reason*/
```


Package VB5_1
32(37)
/*Declaration of AN Signallists
for status FSM primitives*/
signallist SL_ANVPCSTAT_STATUS =
MIE_VPC_unblock_conf,
MIE_VPC_unblock_res,
MIE_VPC_block_conf,
MIE_VPC_await_clear_conf;
signallist SL_STATUS_ANLSPSTAT =
MIE_LSP_unblock_req,
MIE_LSP_unblock_ind,
MIE_LSP_block_req;
signallist SL_STATUS_ANVPCSTAT =
MIE_VPC_unblock_req,
MIE_VPC_unblock_ind,
MIE_VPC_block_req,
MIE_VPC_await_clear_req,
MIE_VPC_delete;
signallist SL_ANLSPSTAT_STATUS =
MIE_LSP_unblock_conf,
MIE_LSP_unblock_res,
MIE_LSP_block_conf;


Package VB5_1
33(37)
/*Declaration of SN Signallists
for status FSM primitives*/
signallist SL_STATUS_SNVPCSTAT =
MIE_VPC_unblock_ind,
MIE_VPC_unblock_req,
MIE_VPC_block_ind,
MIE_VPC_await_clear_ind,
MIE_VPC_delete;
signallist SL_STATUS_SNLSPSTAT =
MIE_LSP_unblock_ind,
MIE_LSP_unblock_req,
MIE_LSP_block_ind;
signallist SL_SNVPCSTAT_STATUS =
MIE_VPC_unblock_res,
MIE_VPC_unblock_conf,
MIE_VPC_block_res,
MIE_VPC_await_clear_res;
signallist SL_SNLSPSTAT_STATUS =
MIE_LSP_unblock_res,
MIE_LSP_unblock_conf,
MIE_LSP_block_res;

/* Declaration of common primitives internal to the CPROT */

```
signal
RTMC_await_clear (PT_TransId, AT_RscList),
RTMC_await_clear_ack (PT_TransId, IT_RTMCResult, AT_RscList),
RTMC_await_clear_comp (PT_TransId, AT_RscList),
RTMC_await_clear_comp_ack (PT_TransId),
RTMC_block_rsc_ack (PT_TransId, IT_RTMCResult, AT_RscList),
RTMC_block_rsc (PT_TransId, AT_RTMCBlRscList),
RTMC_cons_check_end (PT_TransId, ST_RscId),
RTMC_cons_check_end_ack (PT_TransId, IT_RTMCResult, ST_RscId),
RTMC_cons_check_req (PT_TransId, ST_RscId),
RTMC_cons_check_req_ack (PT_TransId, IT_RTMCResult, ST_RscId),
RTMC_LSPID (PT_TransId, ST_RscId),
RTMC_protocol_error (PT_TransId, IT_RTMCProtErrCause),
RTMC_req_LSPID (PT_TransId),
RTMC_reset_rsc_ack (PT_TransId, IT_RTMCResult, AT_RscList),
RTMC_reset_rsc (PT_TransId, AT_RscList),
RTMC_unblock_rsc_ack (PT_TransId, IT_RTMCResult, AT_RscList),
RTMC_unblock_rsc (PT_TransId, AT_RscList),
RTMC_unknown_message(PT_TransId),
RTMC_msg_too_short,
RTMC_trans_id_err,
RTMC_unknown_protdisc,
MIE_LSP_ID_conf (IT_Result, ST_RscId),
MIE_LSP_ID_ind,
MIE_LSP_ID_req,
MIE_LSP_ID_res (ST_RscId),
MIE_abort_req,
/*MIE_error_ind, already defined in the System Management Block*/
MIE_reset_conf (IT_IntTransId, IT_Result, AT_RscList),
MIE_reset_ind (AT_RscList),
MIE_reset_req (IT_IntTransId, AT_RscList),
MIE_reset_res (IT_RTMCResult, AT_RscList);
```

Package VB5_1 35(37)

/* Declaration of primitives internal
to AN_CPROT */

```
signal
MIE_await_clear_conf (IT_IntTransId, IT_Result, AT_RscList),
MIE_await_clear_req (IT_IntTransId, AT_RscList),
MIE_block_conf (IT_IntTransId, IT_Result, AT_RscList),
MIE_block_req (IT_IntTransId, AT_RTMCBlRscList),
MIE_cons_check_end_ind (ST_RscId),
MIE_cons_check_end_res (IT_RTMCResult, ST_RscId),
MIE_cons_check_ind (ST_RscId),
MIE_cons_check_res (IT_RTMCResult, ST_RscId),
MIE_unblock_conf (IT_IntTransId, IT_Result, AT_RscList),
MIE_unblock_req (IT_IntTransId, AT_RscList);
```

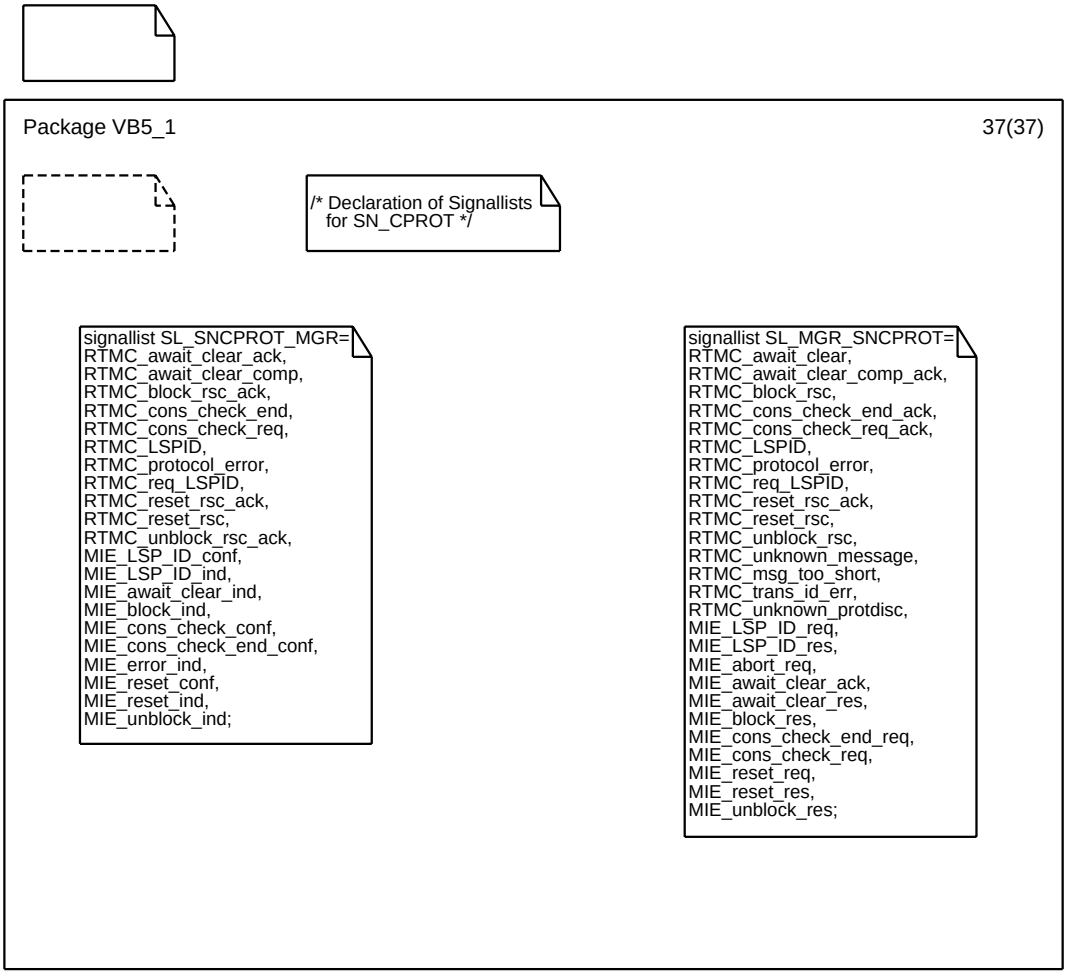
/* Declaration of primitives internal
to SN_CPROT */

```
signal
MIE_await_clear_ack (IT_RTMCResult, AT_RscList),
MIE_await_clear_ind (AT_RscList),
MIE_await_clear_res (AT_RscList),
MIE_block_ind (AT_RTMCBlRscList),
MIE_block_res (IT_RTMCResult, AT_RscList),
MIE_cons_check_conf (IT_Result, ST_RscId),
MIE_cons_check_end_conf (IT_Result, ST_RscId),
MIE_cons_check_end_req (ST_RscId),
MIE_cons_check_req (ST_RscId),
MIE_unblock_ind (AT_RscList),
MIE_unblock_res (IT_RTMCResult, AT_RscList);
```

/* Declaration of Signallists
for AN_CPROT */

```
signallist SL_MGR_ANCPROT=
RTMC_await_clear_ack,
RTMC_await_clear_comp,
RTMC_block_rsc_ack,
RTMC_cons_check_end,
RTMC_cons_check_req,
RTMC_LSPID,
RTMC_protocol_error,
RTMC_req_LSPID,
RTMC_reset_rsc_ack,
RTMC_reset_rsc,
RTMC_unblock_rsc_ack,
RTMC_unknown_message,
RTMC_msg_too_short,
RTMC_trans_id_err,
RTMC_unknown_protdisc,
MIE_LSP_ID_req,
MIE_LSP_ID_res,
MIE_abort_req,
MIE_await_clear_req,
MIE_block_req,
MIE_cons_check_res,
MIE_cons_check_end_res,
MIE_reset_req,
MIE_reset_res,
MIE_unblock_req;
```

```
signallist SL_ANCPROT_MGR=
RTMC_await_clear,
RTMC_await_clear_comp_ack,
RTMC_block_rsc,
RTMC_cons_check_end_ack,
RTMC_cons_check_req_ack,
RTMC_LSPID,
RTMC_protocol_error,
RTMC_req_LSPID,
RTMC_reset_rsc_ack,
RTMC_reset_rsc,
RTMC_unblock_rsc,
MIE_LSP_ID_conf,
MIE_LSP_ID_ind,
MIE_await_clear_conf,
MIE_block_conf,
MIE_cons_check_end_ind,
MIE_cons_check_ind,
MIE_error_ind,
MIE_reset_conf,
MIE_reset_ind,
MIE_unblock_conf;
```


Package VB5_1
37(37)
/* Declaration of Signallists
for SN_CPROT */
signallist SL_SNCPROT_MGR=
RTMC_await_clear_ack,
RTMC_await_clear_comp,
RTMC_block_rsc_ack,
RTMC_cons_check_end,
RTMC_cons_check_req,
RTMC_LSPID,
RTMC_protocol_error,
RTMC_req_LSPID,
RTMC_reset_rsc_ack,
RTMC_reset_rsc,
RTMC_unblock_rsc_ack,
MIE_LSP_ID_conf,
MIE_LSP_ID_ind,
MIE_await_clear_ind,
MIE_block_ind,
MIE_cons_check_conf,
MIE_cons_check_end_conf,
MIE_error_ind,
MIE_reset_conf,
MIE_reset_ind,
MIE_unblock_ind;
signallist SL_MGR_SNCPROT=
RTMC_await_clear,
RTMC_await_clear_comp_ack,
RTMC_block_rsc,
RTMC_cons_check_end_ack,
RTMC_cons_check_req_ack,
RTMC_LSPID,
RTMC_protocol_error,
RTMC_req_LSPID,
RTMC_reset_rsc_ack,
RTMC_reset_rsc,
RTMC_unblock_rsc,
RTMC_unknown_message,
RTMC_msg_too_short,
RTMC_trans_id_err,
RTMC_unknown_protdisc,
MIE_LSP_ID_req,
MIE_LSP_ID_res,
MIE_abort_req,
MIE_await_clear_ack,
MIE_await_clear_res,
MIE_block_res,
MIE_cons_check_end_req,
MIE_cons_check_req,
MIE_reset_req,
MIE_reset_res,
MIE_unblock_res;

Procedure <<Package VB5_1>> ANA_RSC 1(5)

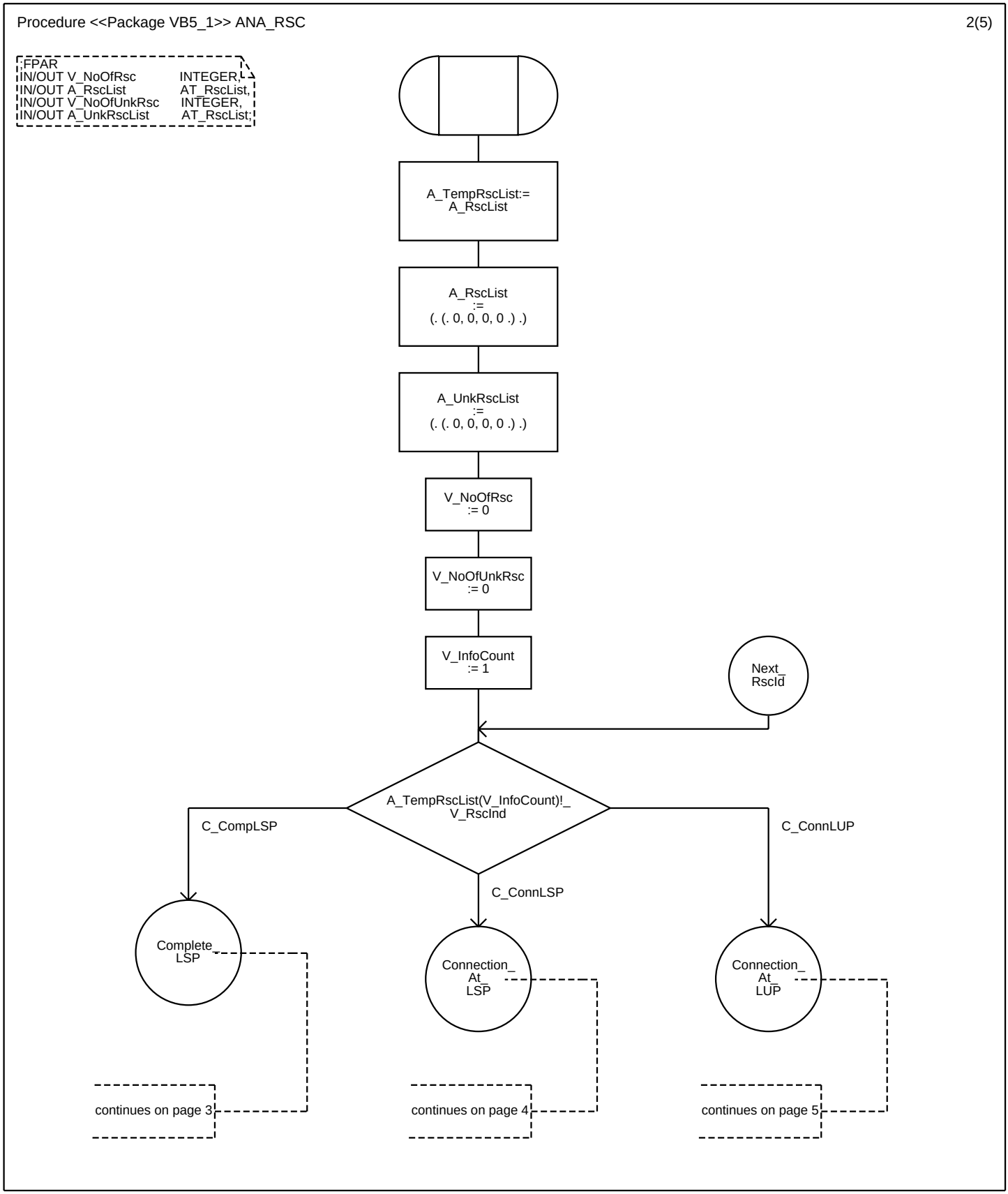
```
;FPAR
IN/OUT V_NoOfRsc          INTEGER,
IN/OUT A_RscList          AT_RscList,
IN/OUT V_NoOfUnkRsc       INTEGER,
IN/OUT A_UnkRscList       AT_RscList;
```

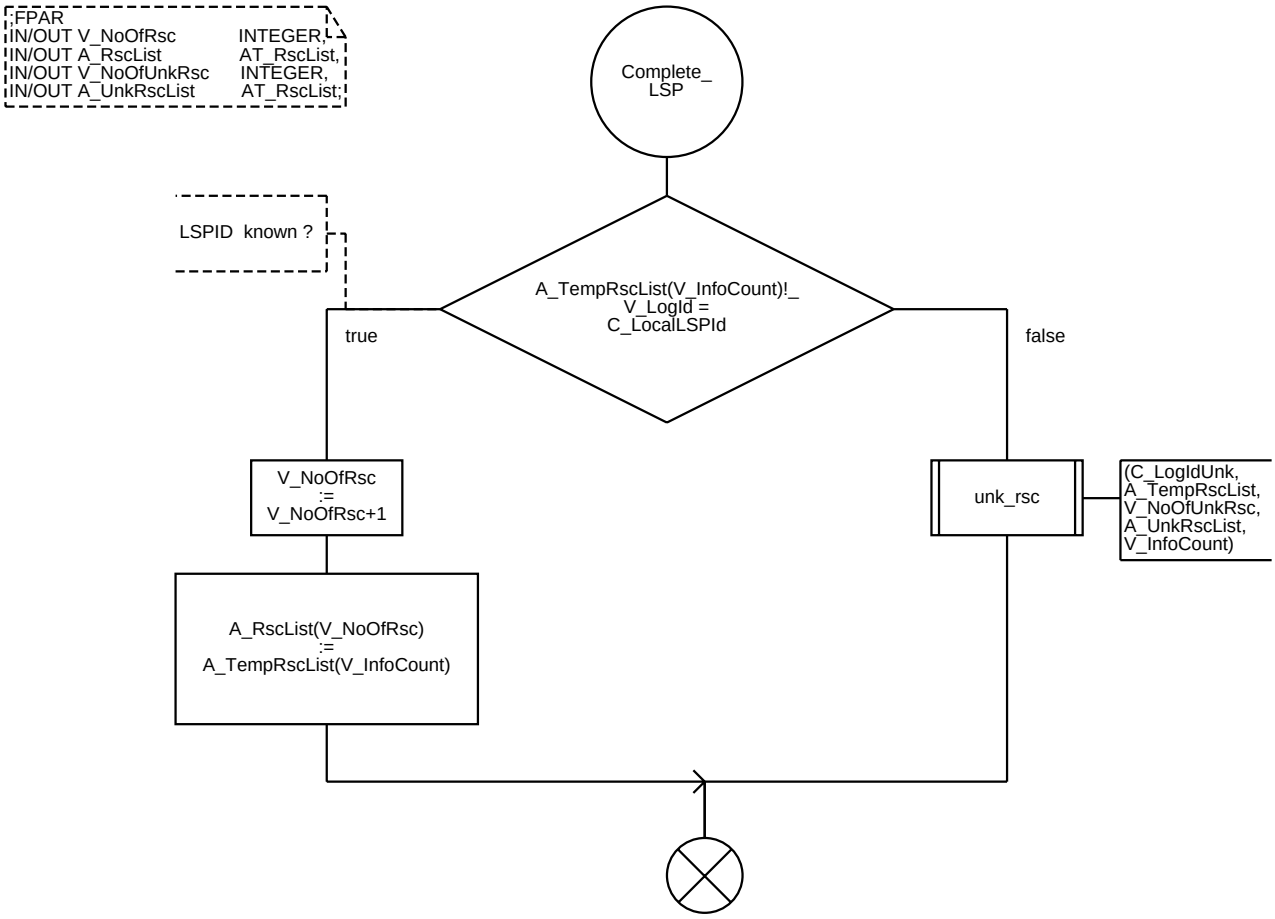
```
/* analyze resource list wrt.
   known or unknown resource */
```

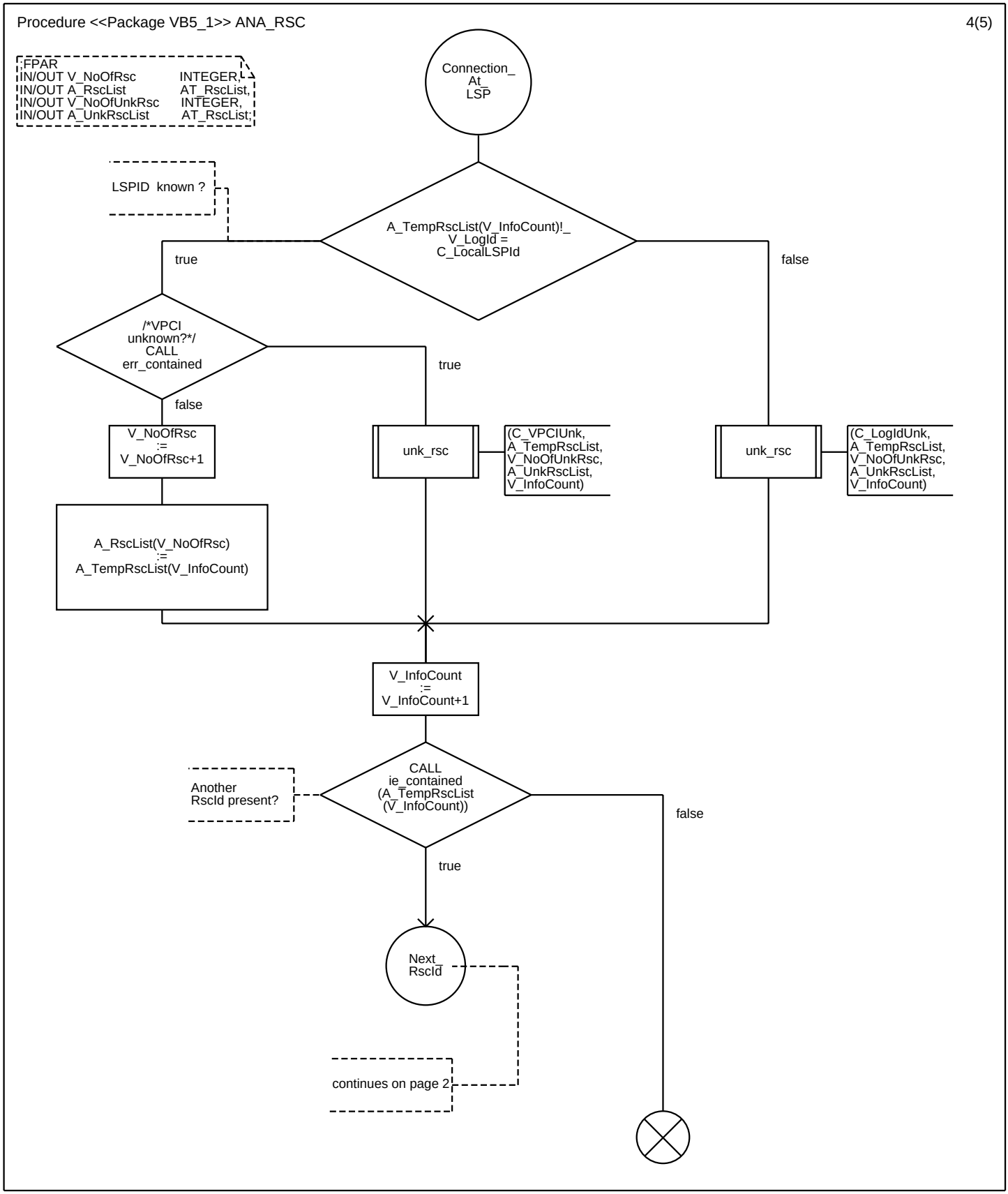
```
/*Definitions and Declarations*/
```

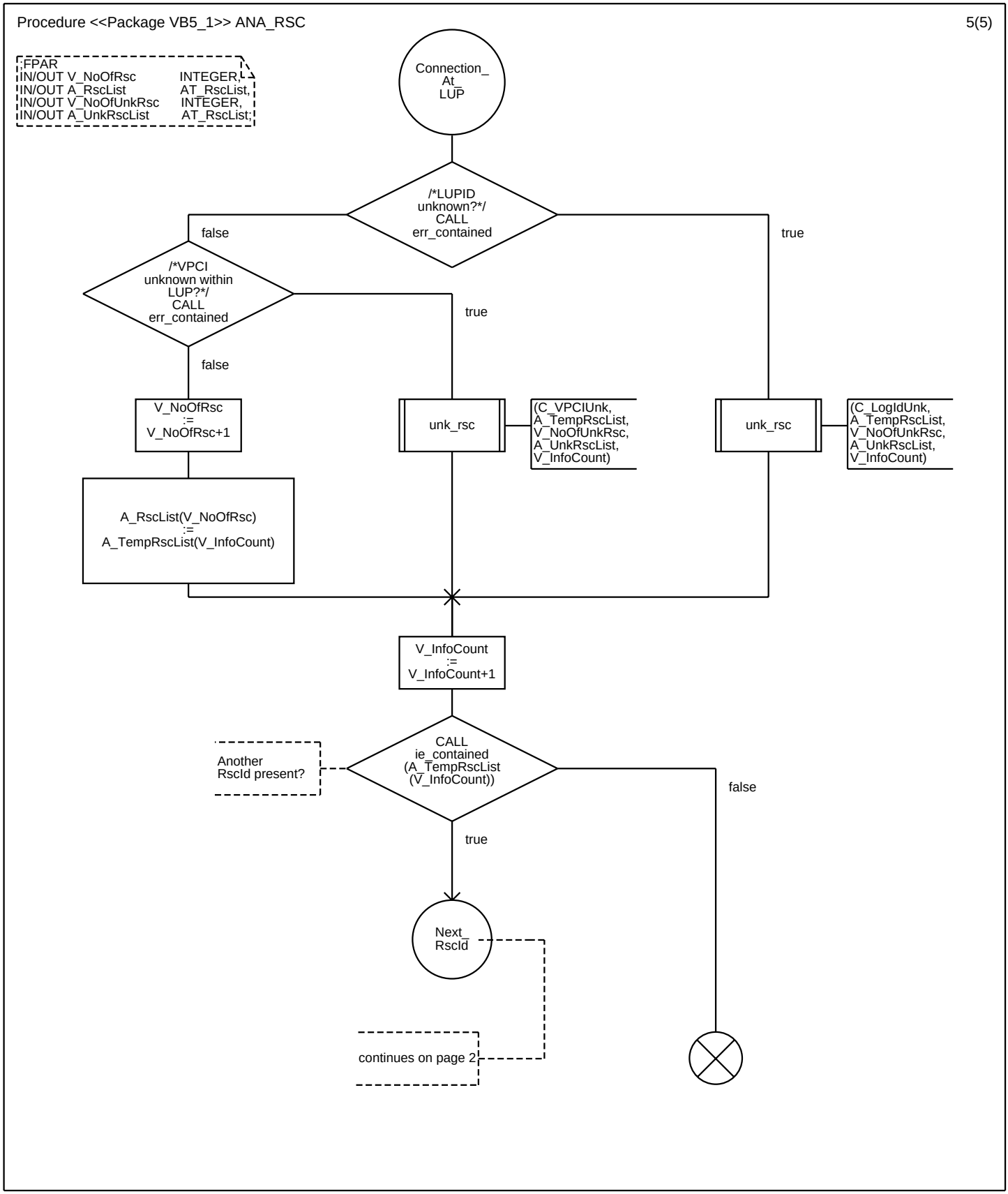
```
/*Internal variables of ANALYZE_RESOURCE*/
/**/
DCL
V_InfoCount
INTEGER;
```

```
/*Data declarations for internal use*/
DCL
A_TempRscList  AT_RscList   ; /*Array for temporary storage of Resource Identifier Information Elements*/
```


Procedure <<Package VB5_1>> ANA_RSC
2(5)
;FPAR
IN/OUT V_NoOfRsc INTEGER,
IN/OUT A_RscList AT_RscList,
IN/OUT V_NoOfUnkRsc INTEGER,
IN/OUT A_UnkRscList AT_RscList;
A_TempRscList:= A_RscList
A_RscList := (. (. 0, 0, 0, 0 .) .)
A_UnkRscList := (. (. 0, 0, 0, 0 .) .)
V_NoOfRsc := 0
V_NoOfUnkRsc := 0
V_InfoCount := 1
Next_ RscId
A_TempRscList(V_InfoCount)!_ V_RscInd
C_CompLSP
C_ConnLSP
C_ConnLUP
Complete_ LSP
Connection_ At_ LSP
Connection_ At_ LUP
continues on page 3
continues on page 4
continues on page 5

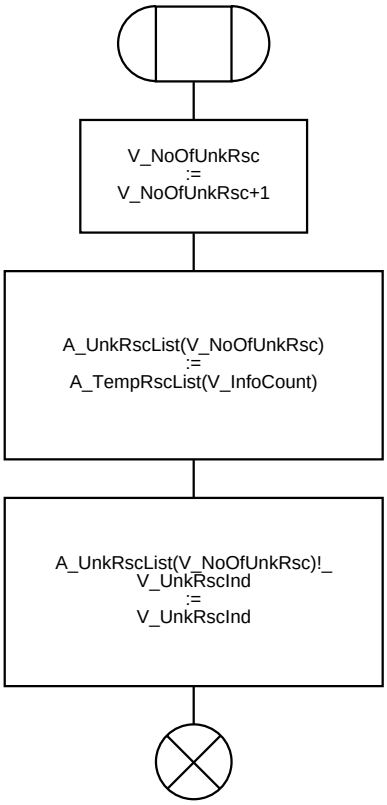

;FPAR
IN/OUT V_NoOfRsc INTEGER,
IN/OUT A_RscList AT_RscList,
IN/OUT V_NoOfUnkRsc INTEGER,
IN/OUT A_UnkRscList AT_RscList;
Complete_ LSP
LSPID known ?
A_TempRscList(V_InfoCount)!_ V_LogId = C_LocalLSPId
true
false
V_NoOfRsc := V_NoOfRsc+1
unk_rsc
(C_LogIdUnk, A_TempRscList, V_NoOfUnkRsc, A_UnkRscList, V_InfoCount)
A_RscList(V_NoOfRsc) := A_TempRscList(V_InfoCount)


Procedure <<Package VB5_1>> ANA_RSC
4(5)
FPAR
IN/OUT V_NoOfRsc INTEGER,
IN/OUT A_RscList AT_RscList,
IN/OUT V_NoOfUnkRsc INTEGER,
IN/OUT A_UnkRscList AT_RscList;
Connection_ At_ LSP
LSPID known ?
A_TempRscList(V_InfoCount)!_ V_LogId = C_LocalLSPId
true
false
/*VPCI unknown?*/ CALL err_contained
true
false
V_NoOfRsc := V_NoOfRsc+1
unk_rsc
(C_VPCIUnk, A_TempRscList, V_NoOfUnkRsc, A_UnkRscList, V_InfoCount)
unk_rsc
(C_LogIdUnk, A_TempRscList, V_NoOfUnkRsc, A_UnkRscList, V_InfoCount)
A_RscList(V_NoOfRsc) := A_TempRscList(V_InfoCount)
V_InfoCount := V_InfoCount+1
Another RscId present?
CALL ie_contained (A_TempRscList (V_InfoCount))
false
true
Next_ RscId
continues on page 2


Procedure <<Package VB5_1>> ANA_RSC
5(5)
;FPAR
IN/OUT V_NoOfRsc INTEGER,
IN/OUT A_RscList AT_RscList,
IN/OUT V_NoOfUnkRsc INTEGER,
IN/OUT A_UnkRscList AT_RscList;
Connection_ At_ LUP
/*LUPID unknown?*/ CALL err_contained
false
true
/*VPCI unknown within LUP?*/ CALL err_contained
true
false
V_NoOfRsc := V_NoOfRsc+1
unk_rsc
(C_VPCIUnk, A_TempRscList, V_NoOfUnkRsc, A_UnkRscList, V_InfoCount)
unk_rsc
(C_LogIdUnk, A_TempRscList, V_NoOfUnkRsc, A_UnkRscList, V_InfoCount)
A_RscList(V_NoOfRsc) := A_TempRscList(V_InfoCount)
V_InfoCount := V_InfoCount+1
Another RscId present?
CALL ie_contained (A_TempRscList (V_InfoCount))
false
true
Next_ RscId
continues on page 2

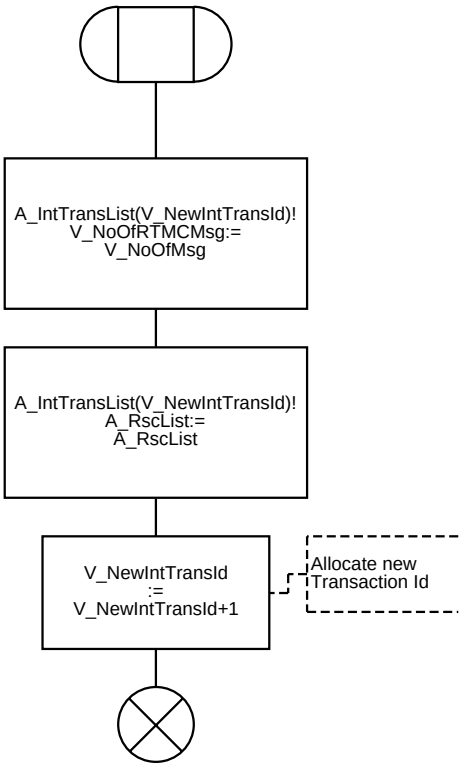
Procedure unk_rsc 1(1)

```
/*Handling of data arrays
in case of found unknown resources*/
;FPAR
   V_UnkRscInd                IT_RTMCUnkRscInd,
   A_TempRscList              AT_RscList,
IN/OUT V_NoOfUnkRsc           INTEGER,
IN/OUT A_UnkRscList           AT_RscList,
   V_InfoCount                INTEGER;
```

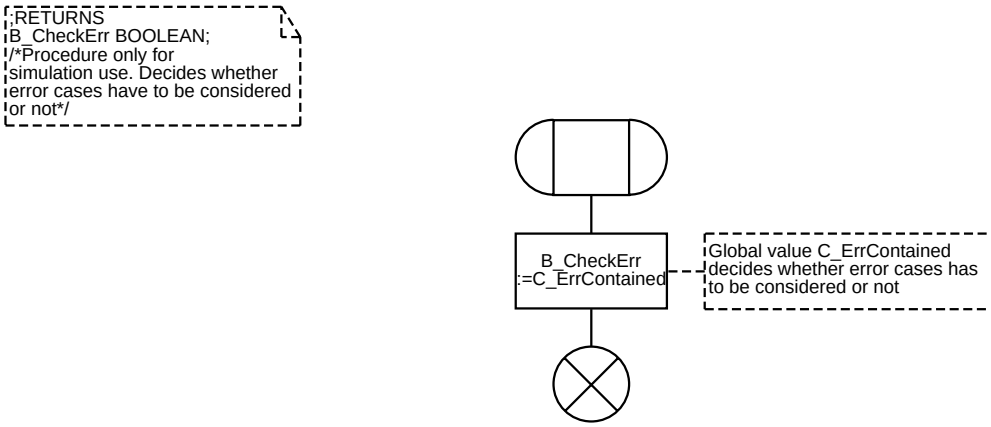


Procedure UPDATE_TRANSLIST 1(1)

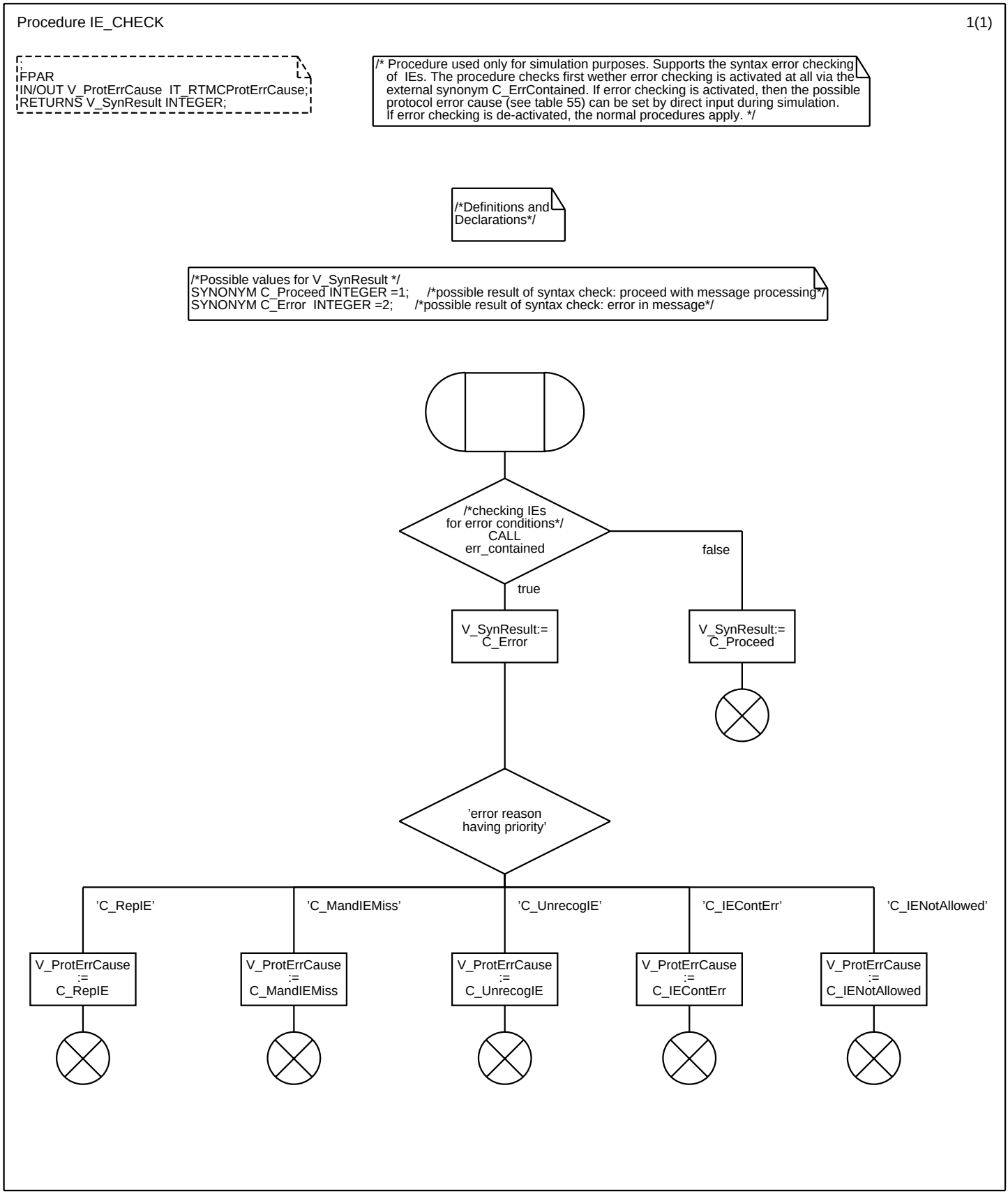
```
;FPAR
IN/OUT   A_IntTransList        AT_IntTransList,
         V_NoOfMsg             INTEGER,
IN/OUT   V_NewIntTransId       IT_IntTransId,
         A_RscList             AT_RscList;
```



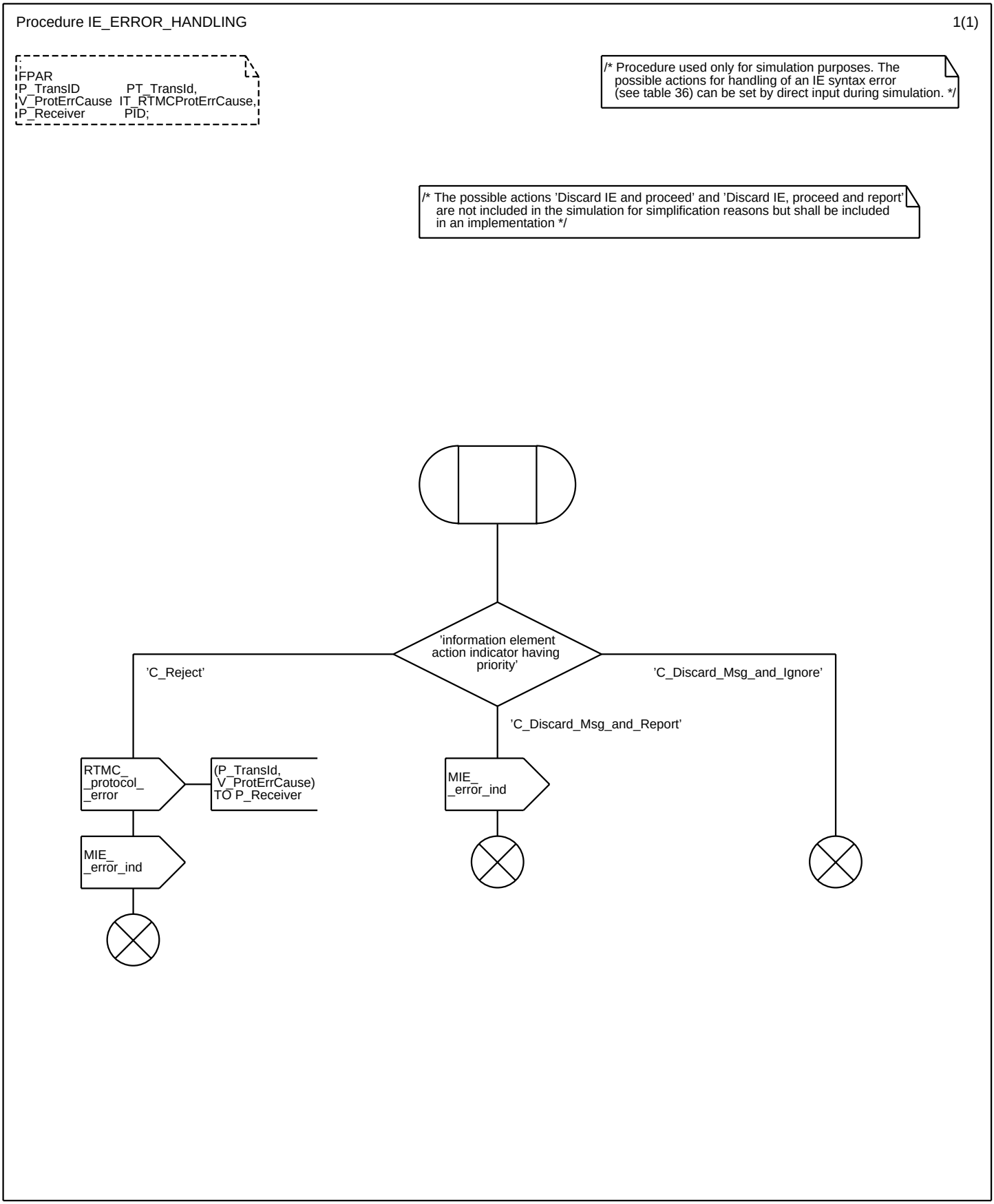
Procedure ERR_CONTAINED 1(1)



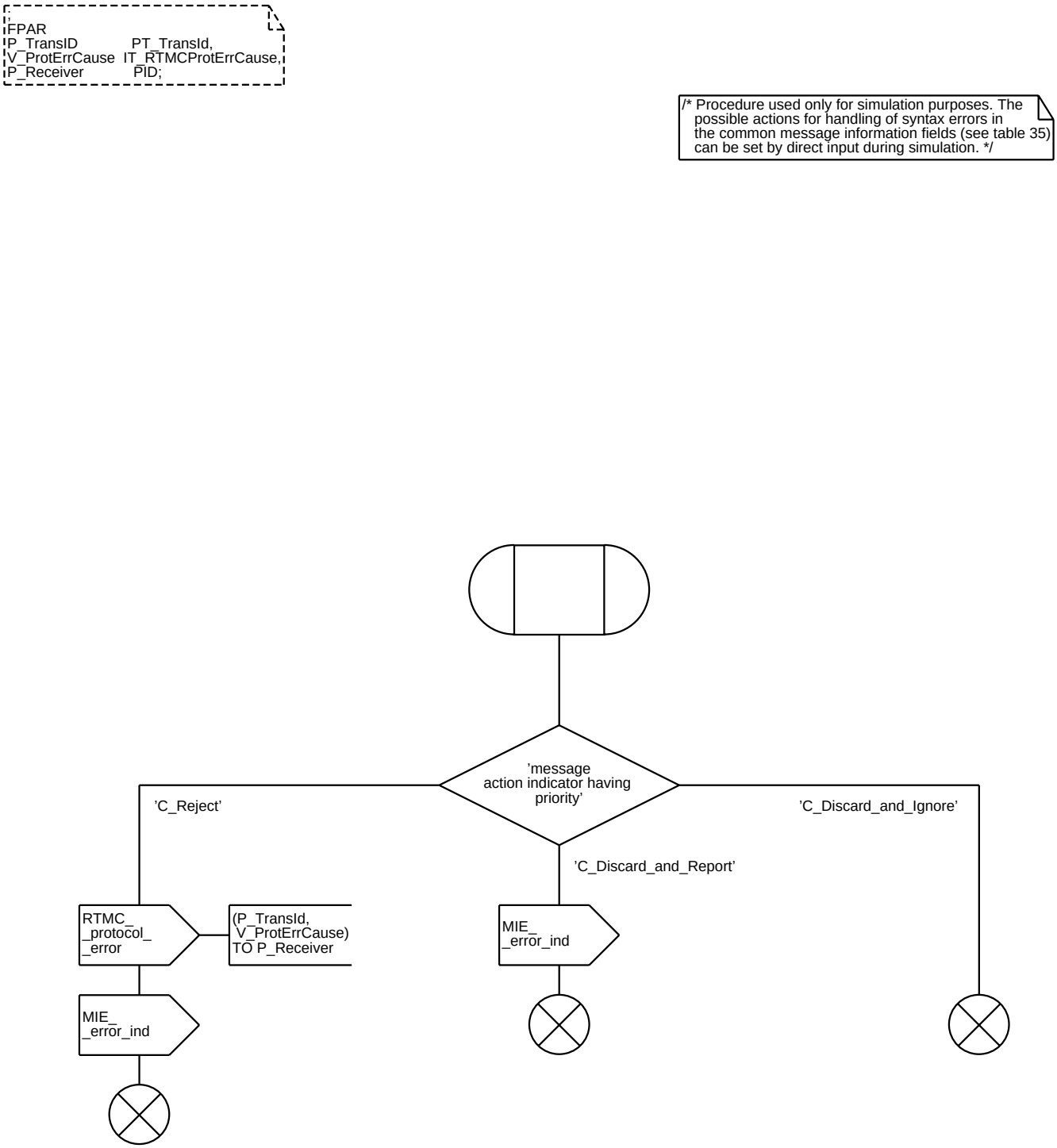
Procedure IE_CHECK 1(1)



Procedure IE_ERROR_HANDLING 1(1)

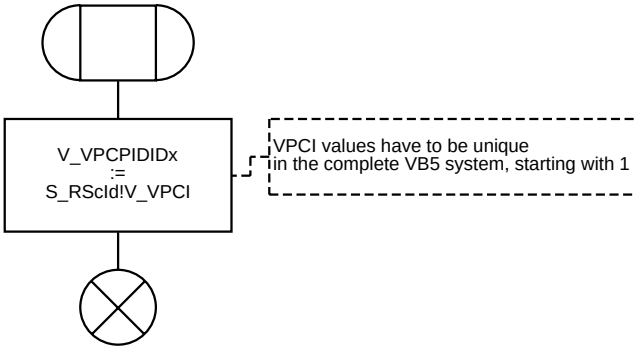


Procedure MSG_ERROR_HANDLING 1(1)



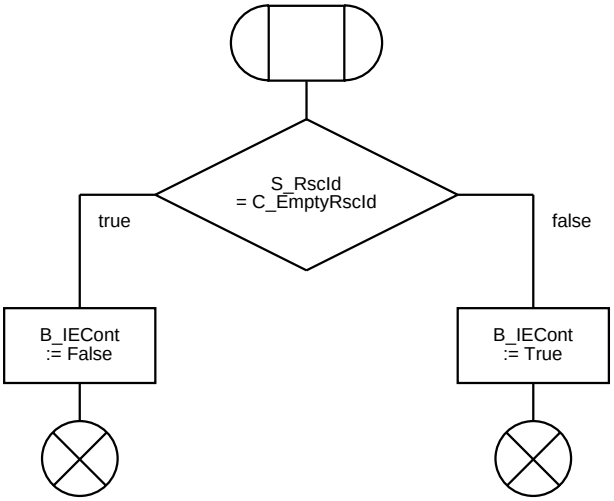
Procedure CONV_RSCID 1(1)

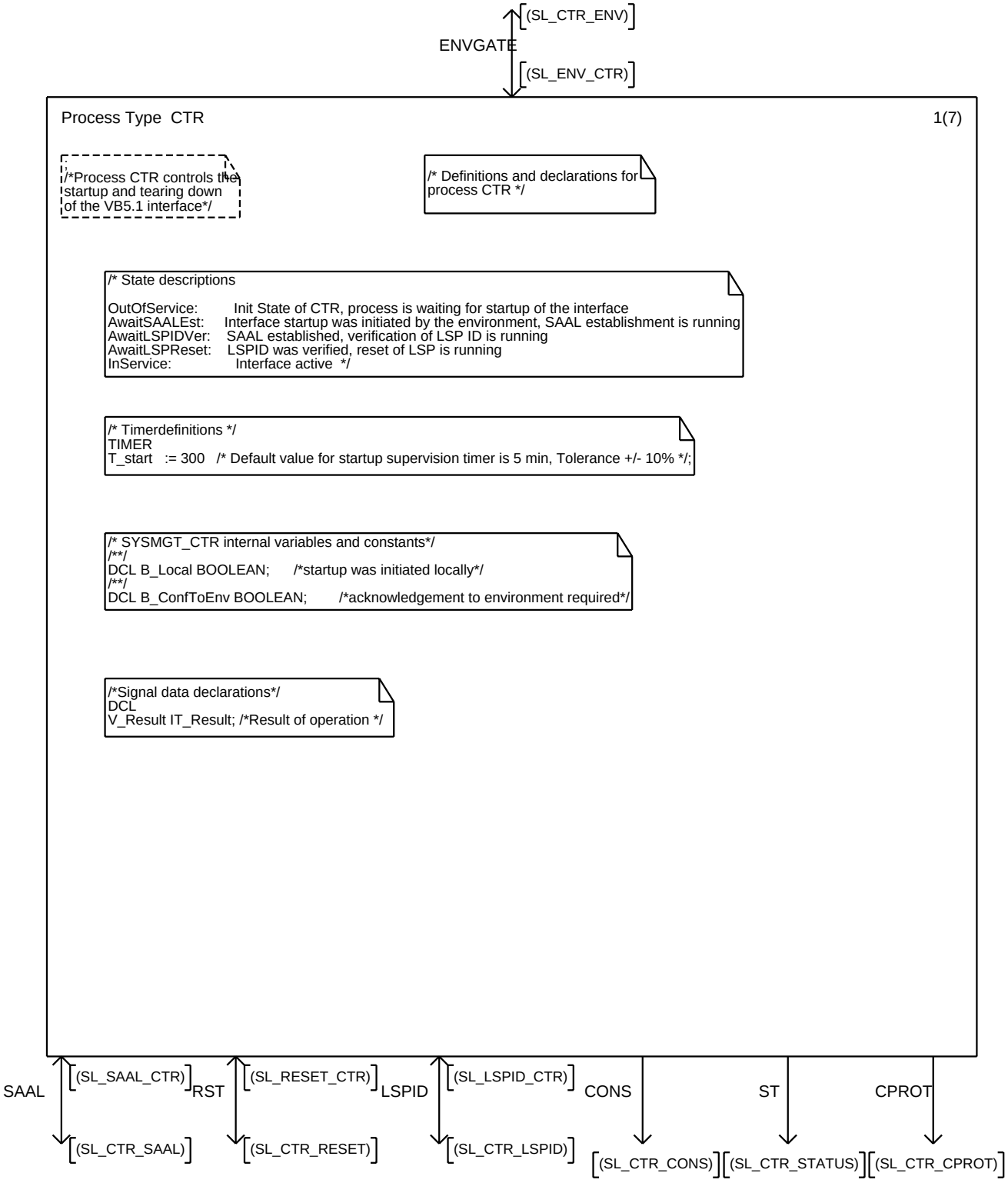
```
FPAR S_RscId ST_RscId;
RETURNS V_VPCPIDIdx INTEGER;
/*Only for simulation use. Determines the index of a resource
identifier in the PID table*/
```

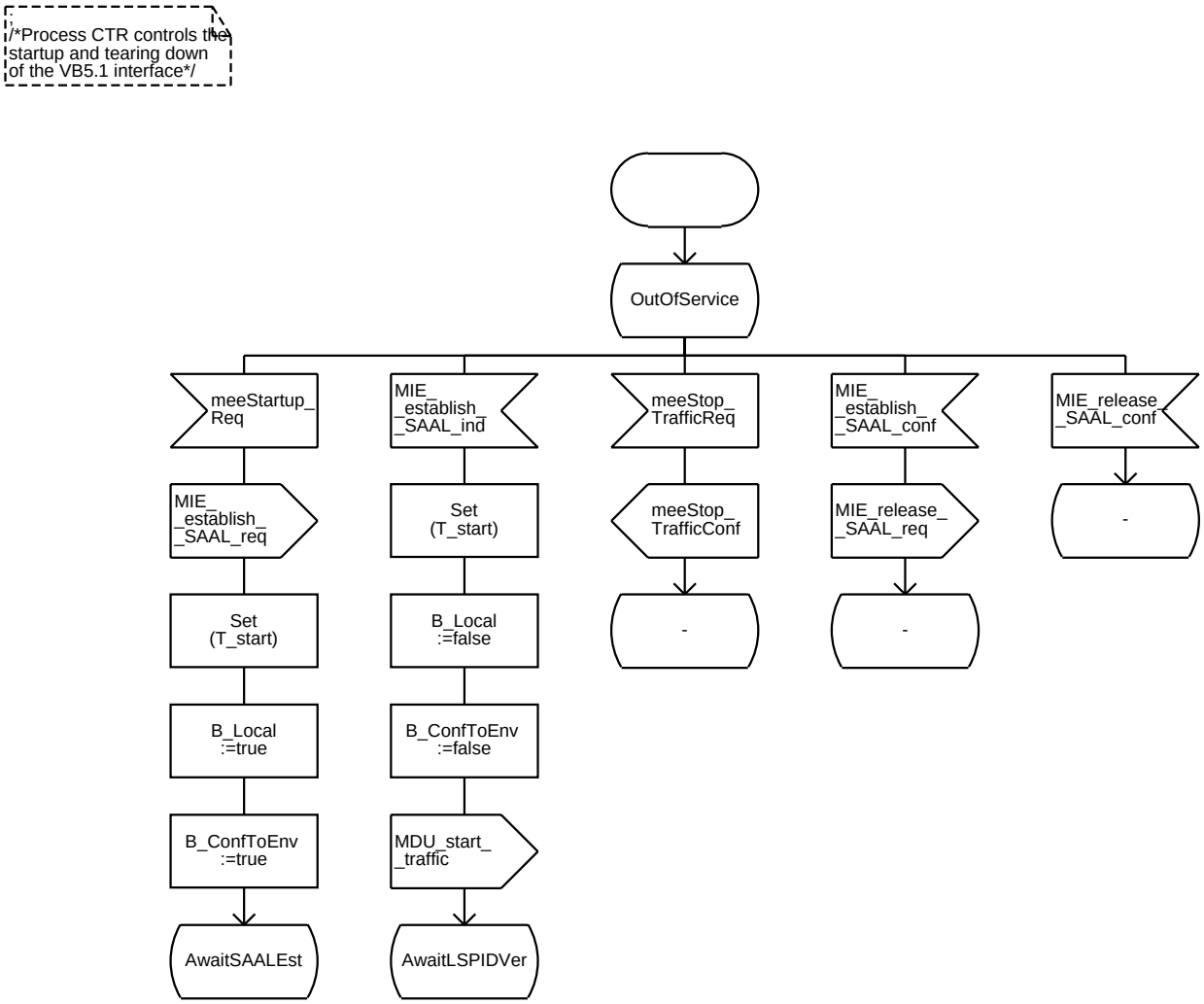


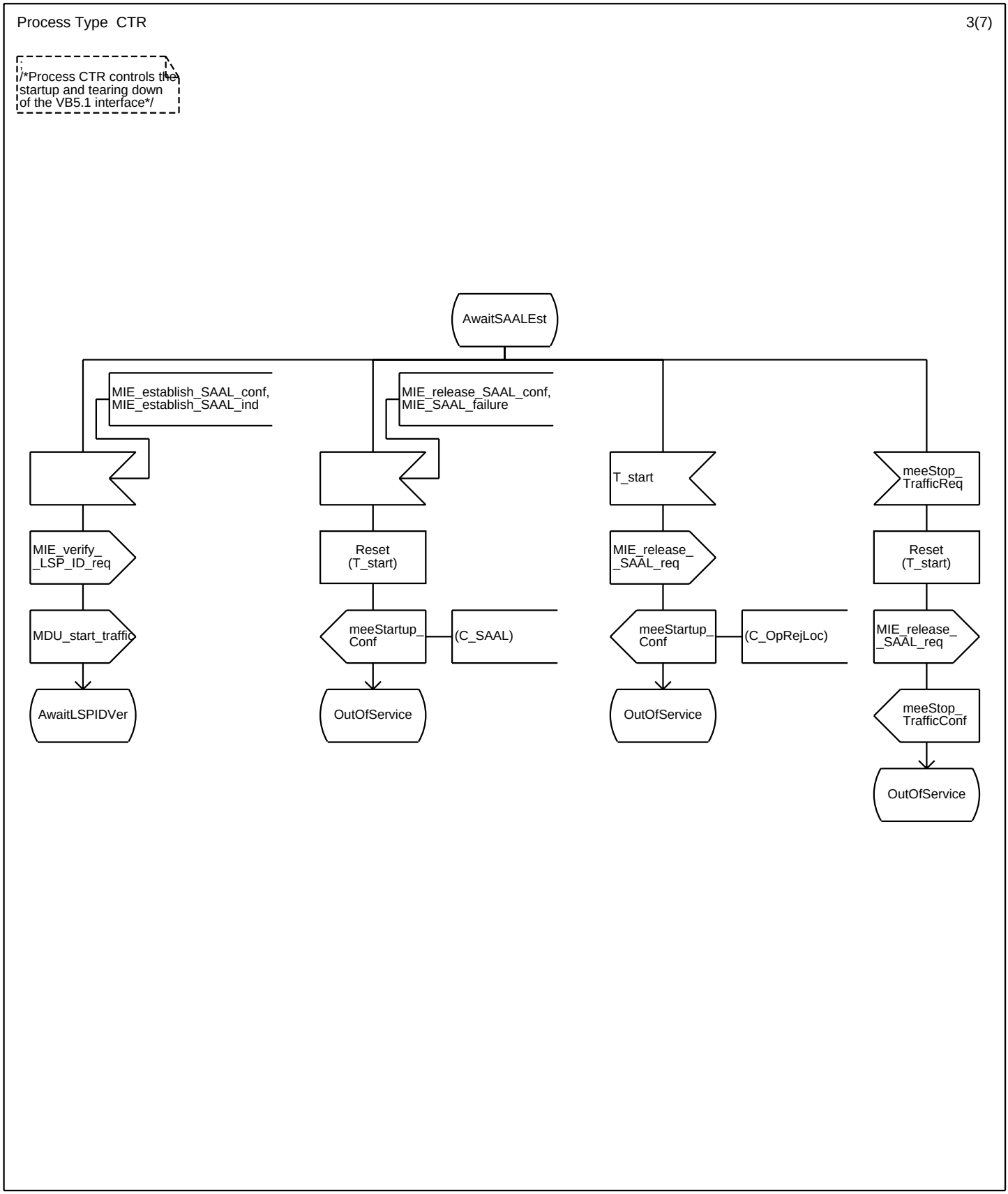
Procedure IE_CONTAINED 1(1)

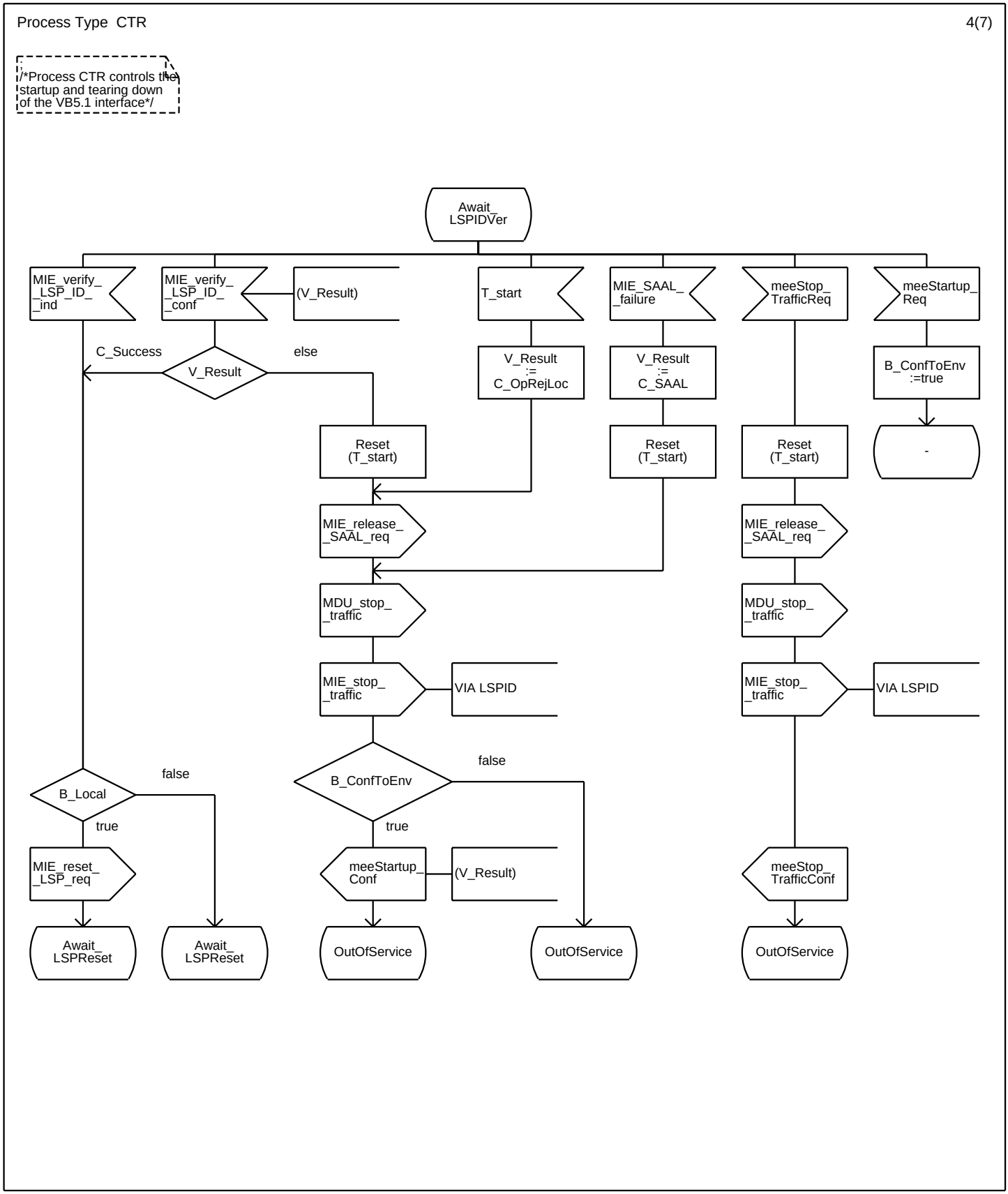
```
FPAR S_RscId ST_RscId;
RETURNS B_IECont BOOLEAN;
/*Only for simulation use. Determines whether
a resource identifier is present or not*/
```

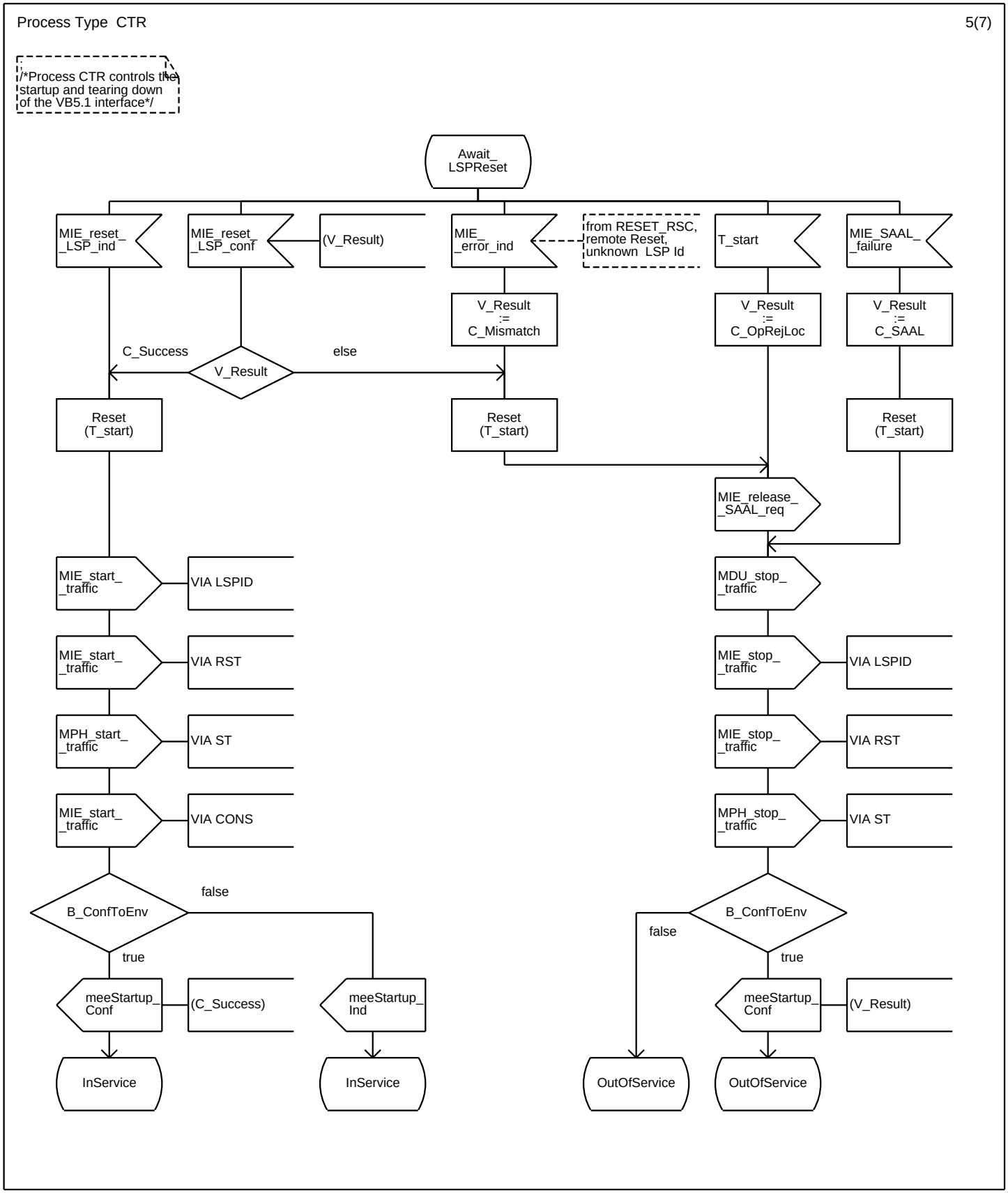


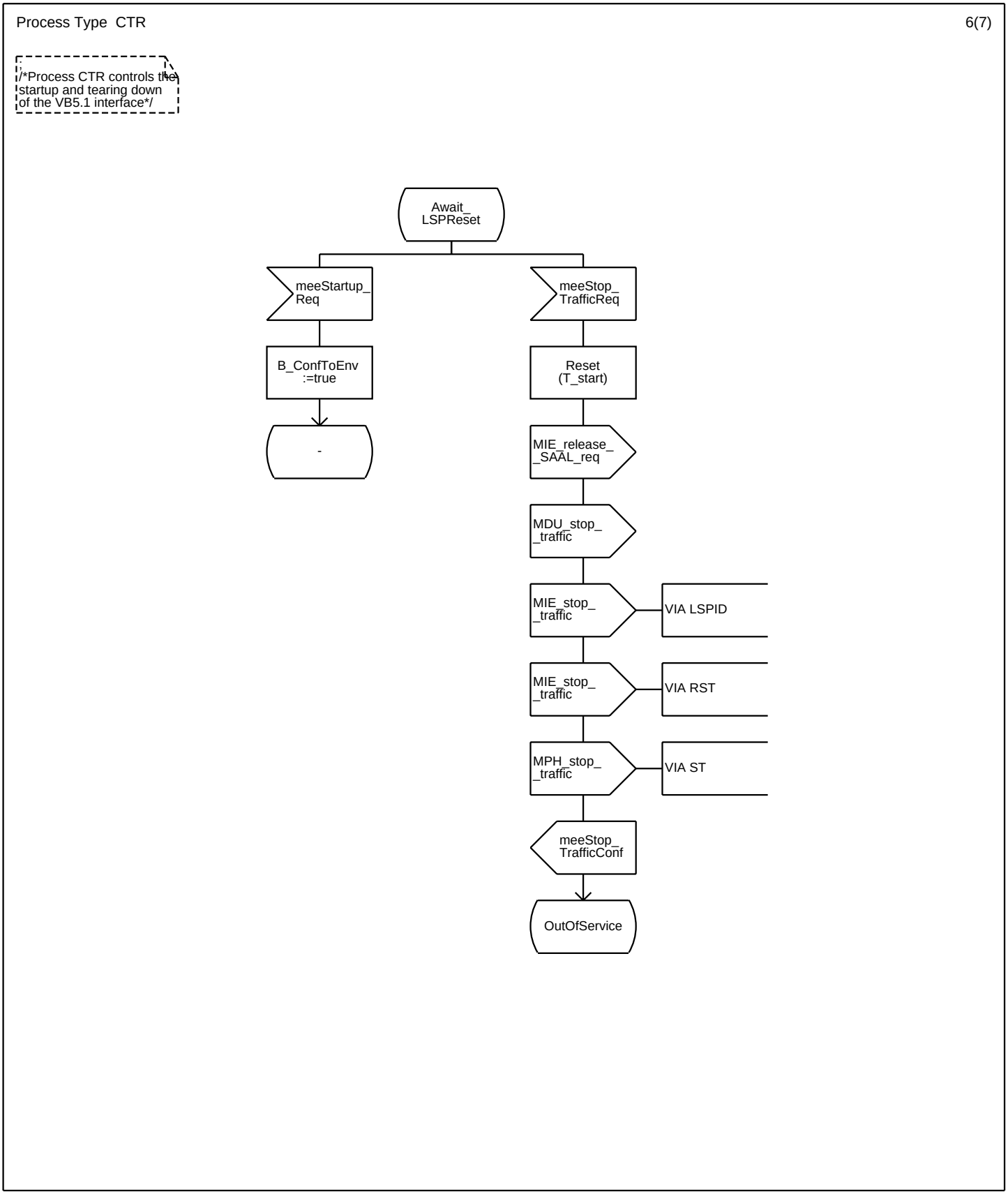

(SL_CTR_ENV)
ENVGATE
(SL_ENV_CTR)
Process Type CTR
1(7)
/*Process CTR controls the startup and tearing down of the VB5.1 interface*/
/* Definitions and declarations for process CTR */
/* State descriptions
OutOfService: Init State of CTR, process is waiting for startup of the interface
AwaitSAALEst: Interface startup was initiated by the environment, SAAL establishment is running
AwaitLSPIDVer: SAAL established, verification of LSP ID is running
AwaitLSPReset: LSPID was verified, reset of LSP is running
InService: Interface active */
/* Timerdefinitions */
TIMER
T_start := 300 /* Default value for startup supervision timer is 5 min, Tolerance +/- 10% */;
/* SYSMGT_CTR internal variables and constants*/
/**/
DCL B_Local BOOLEAN; /*startup was initiated locally*/
/**/
DCL B_ConfToEnv BOOLEAN; /*acknowledgement to environment required*/
/*Signal data declarations*/
DCL
V_Result IT_Result; /*Result of operation */
SAAL
(SL_SAAL_CTR)
(SL_CTR_SAAL)
RST
(SL_RESET_CTR)
(SL_CTR_RESET)
LSPID
(SL_LSPID_CTR)
(SL_CTR_LSPID)
CONS
(SL_CTR_CONS)
ST
(SL_CTR_STATUS)
CPROT
(SL_CTR_CPROT)

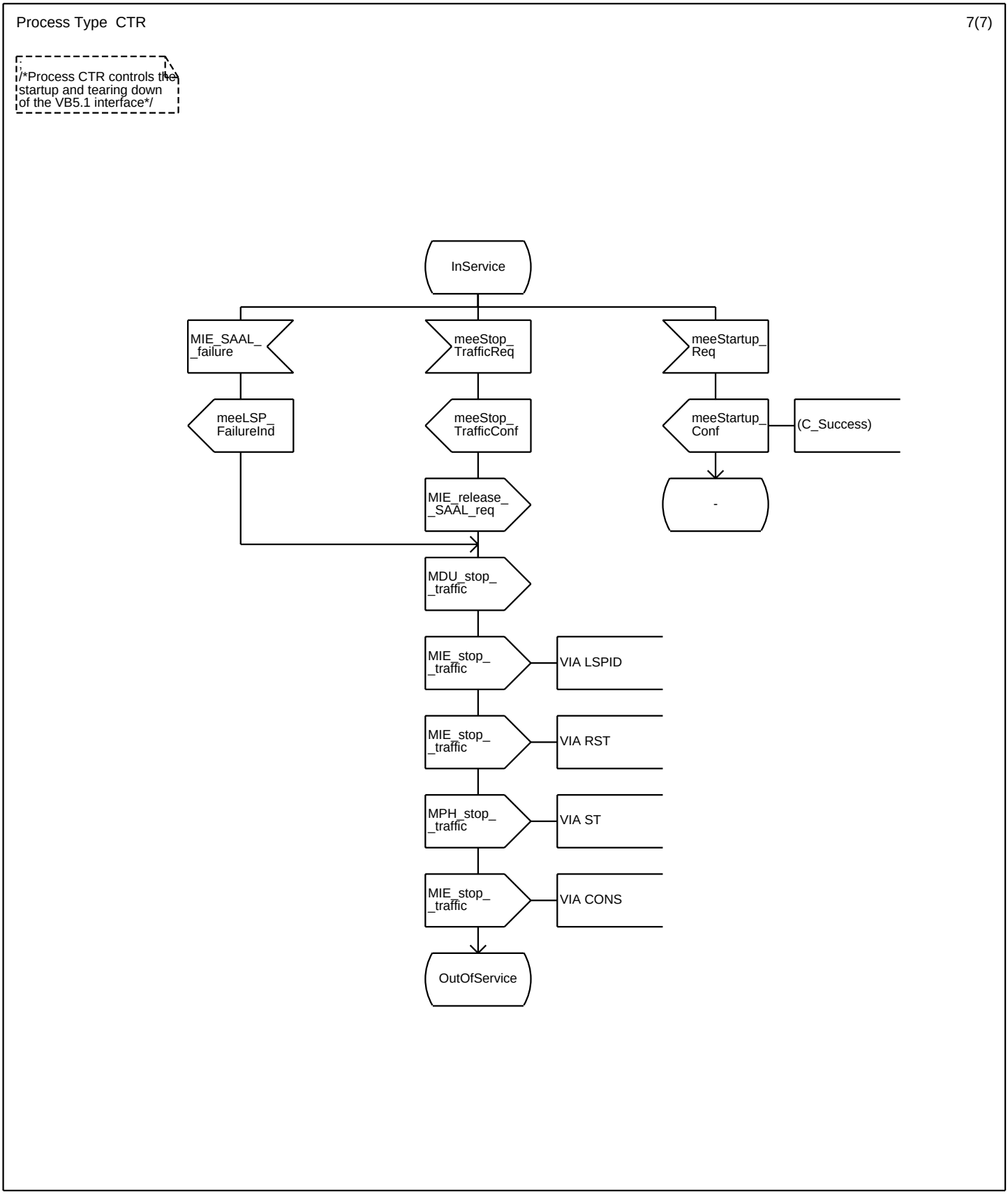

/*Process CTR controls the startup and tearing down of the VB5.1 interface*/
OutOfService
meeStartup_ Req
MIE_ _establish_ _SAAL_req
Set (T_start)
B_Local :=true
B_ConfToEnv :=true
AwaitSAALEst
MIE_ _establish_ _SAAL_ind
Set (T_start)
B_Local :=false
B_ConfToEnv :=false
MDU_start_ _traffic
AwaitLSPIDVer
meeStop_ TrafficReq
meeStop_ TrafficConf
-
MIE_ _establish_ _SAAL_conf
MIE_release_ _SAAL_req
-
MIE_release _SAAL_conf
-

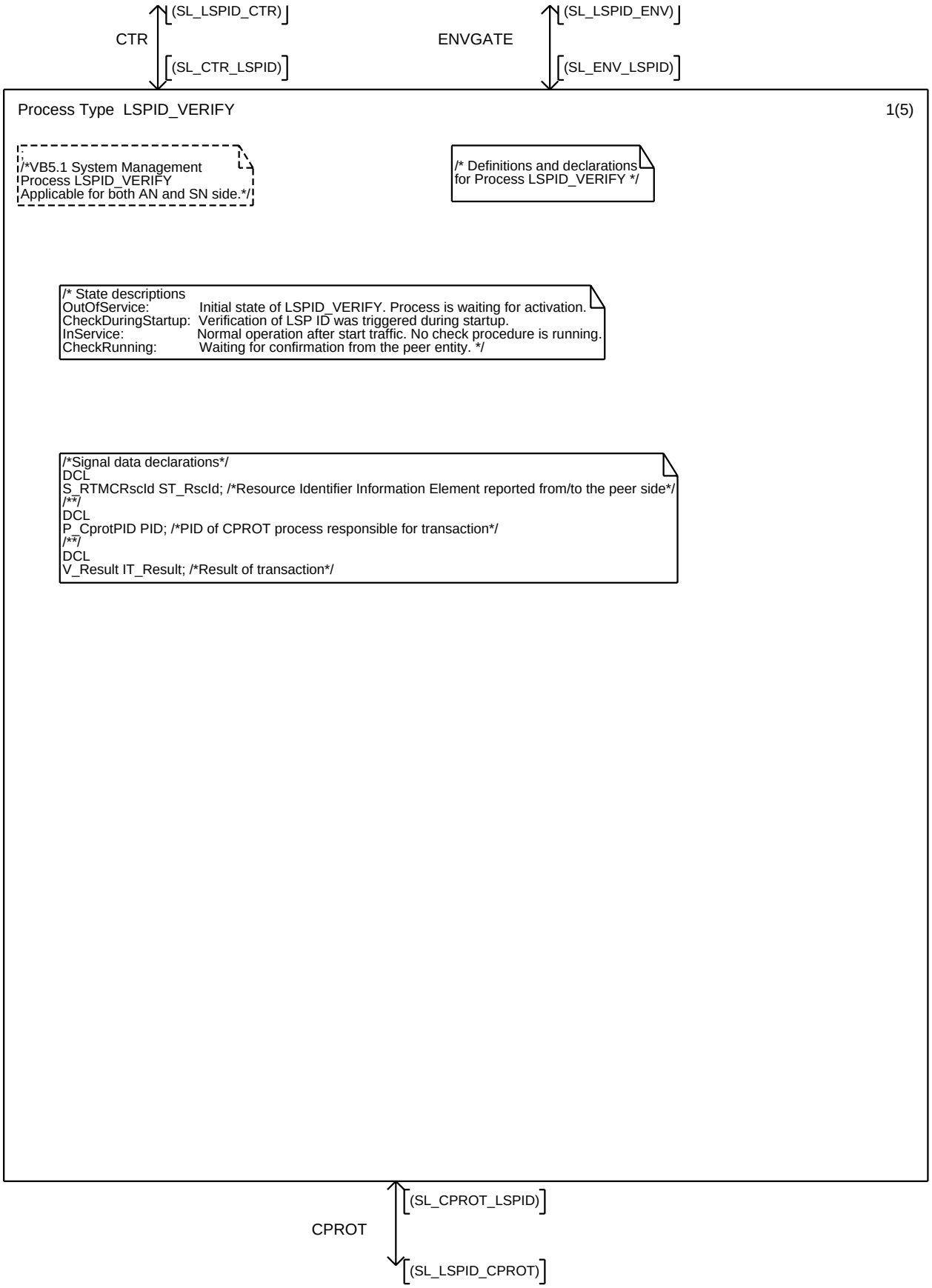

/*Process CTR controls the startup and tearing down of the VB5.1 interface*/
AwaitSAALEst
MIE_establish_SAAL_conf, MIE_establish_SAAL_ind
MIE_release_SAAL_conf, MIE_SAAL_failure
T_start
meeStop_ TrafficReq
MIE_verify_ _LSP_ID_req
Reset (T_start)
MIE_release_ _SAAL_req
Reset (T_start)
MDU_start_traffic
meeStartup_ Conf
(C_SAAL)
meeStartup_ Conf
(C_OpRejLoc)
MIE_release_ _SAAL_req
AwaitLSPIDVer
OutOfService
OutOfService
meeStop_ TrafficConf
OutOfService


Process Type CTR
4(7)
/*Process CTR controls the startup and tearing down of the VB5.1 interface*/
Await_ LSPIDVer
MIE_verify_ _LSP_ID_ _ind
MIE_verify_ _LSP_ID_ _conf
(V_Result)
T_start
MIE_SAAL_ _failure
meeStop_ TrafficReq
meeStartup_ Req
C_Success
V_Result
else
V_Result := C_OpRejLoc
V_Result := C_SAAL
B_ConfToEnv :=true
Reset (T_start)
Reset (T_start)
Reset (T_start)
-
MIE_release_ _SAAL_req
MIE_release_ _SAAL_req
MDU_stop_ _traffic
MDU_stop_ _traffic
MIE_stop_ _traffic
VIA LSPID
MIE_stop_ _traffic
VIA LSPID
false
B_Local
B_ConfToEnv
false
true
true
MIE_reset_ _LSP_req
meeStartup_ Conf
(V_Result)
meeStop_ TrafficConf
Await_ LSPReset
Await_ LSPReset
OutOfService
OutOfService
OutOfService

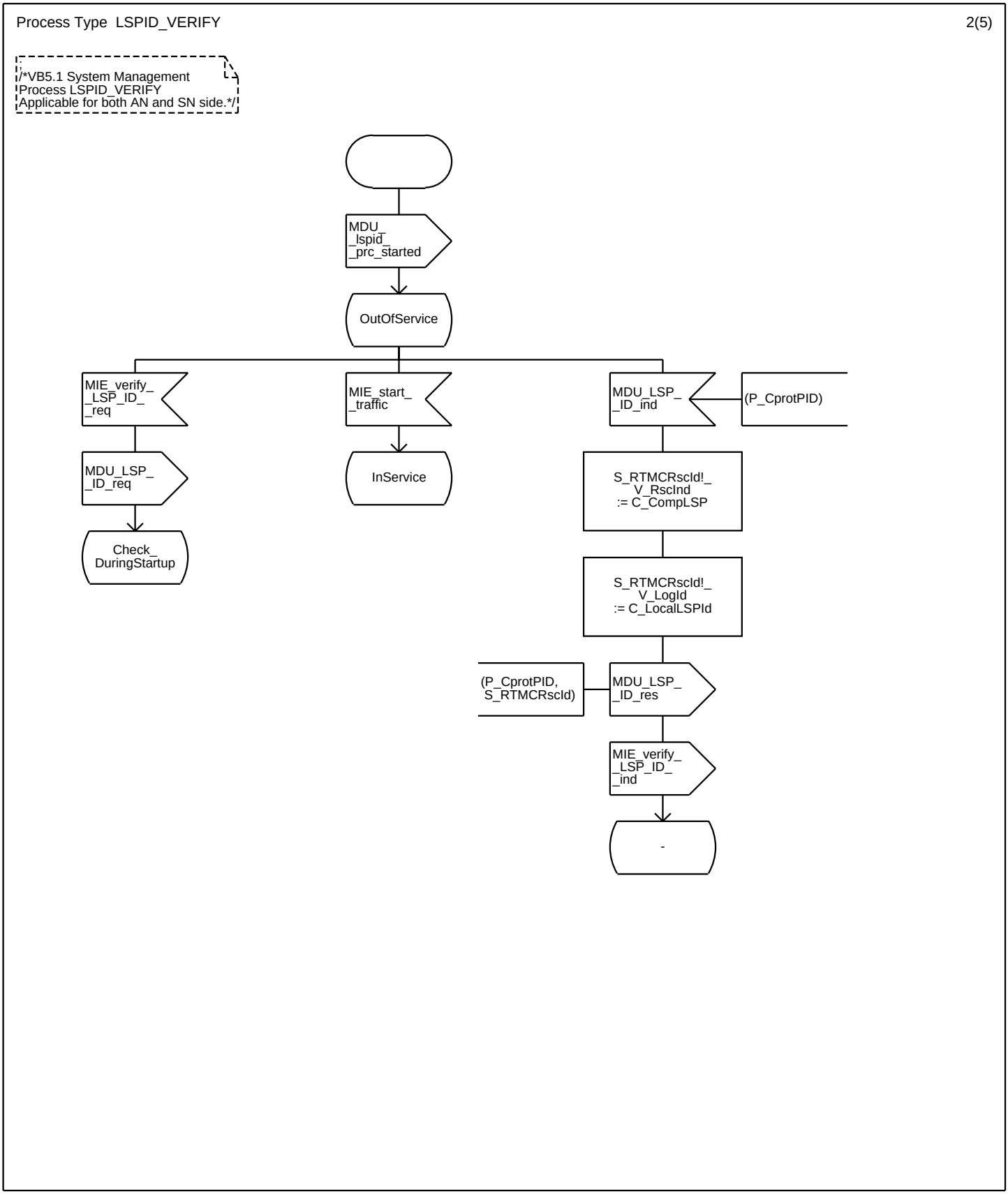

Process Type CTR
5(7)
/*Process CTR controls the startup and tearing down of the VB5.1 interface*/
Await_ LSPReset
MIE_reset_ _LSP_ind
MIE_reset_ _LSP_conf
(V_Result)
MIE_ _error_ind
from RESET_RSC, remote Reset, unknown LSP Id
T_start
MIE_SAAL_ _failure
V_Result := C_Mismatch
V_Result := C_OpRejLoc
V_Result := C_SAAL
C_Success
V_Result
else
Reset (T_start)
Reset (T_start)
Reset (T_start)
MIE_release_ _SAAL_req
MIE_start_ _traffic
VIA LSPID
MDU_stop_ _traffic
MIE_start_ _traffic
VIA RST
MIE_stop_ _traffic
VIA LSPID
MPH_start_ _traffic
VIA ST
MIE_stop_ _traffic
VIA RST
MIE_start_ _traffic
VIA CONS
MPH_stop_ _traffic
VIA ST
B_ConfToEnv
false
true
B_ConfToEnv
false
true
meeStartup_ Conf
(C_Success)
meeStartup_ Ind
meeStartup_ Conf
(V_Result)
InService
InService
OutOfService
OutOfService

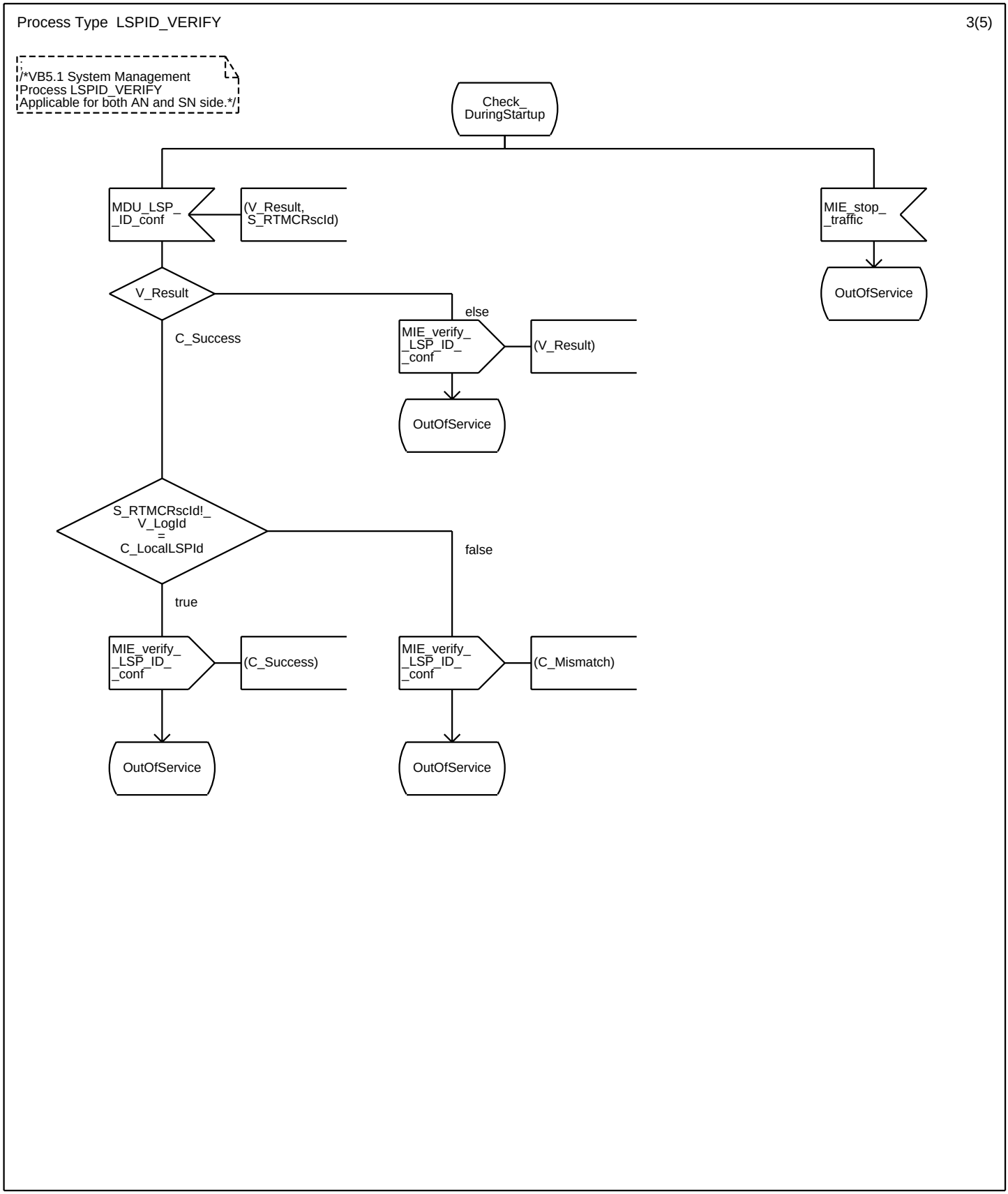

//*Process CTR controls the startup and tearing down of the VB5.1 interface*/
Await_ LSPReset
meeStartup_ Req
meeStop_ TrafficReq
B_ConfToEnv :=true
Reset (T_start)
-
MIE_release_ _SAAL_req
MDU_stop_ _traffic
MIE_stop_ _traffic
VIA LSPID
MIE_stop_ _traffic
VIA RST
MPH_stop_ _traffic
VIA ST
meeStop_ TrafficConf
OutOfService


/*Process CTR controls the startup and tearing down of the VB5.1 interface*/
InService
MIE_SAAL_ _failure
meeStop_ TrafficReq
meeStartup_ Req
meeLSP_ FailureInd
meeStop_ TrafficConf
meeStartup_ Conf
(C_Success)
MIE_release_ _SAAL_req
-
MDU_stop_ _traffic
MIE_stop_ _traffic
VIA LSPID
MIE_stop_ _traffic
VIA RST
MPH_stop_ _traffic
VIA ST
MIE_stop_ _traffic
VIA CONS
OutOfService

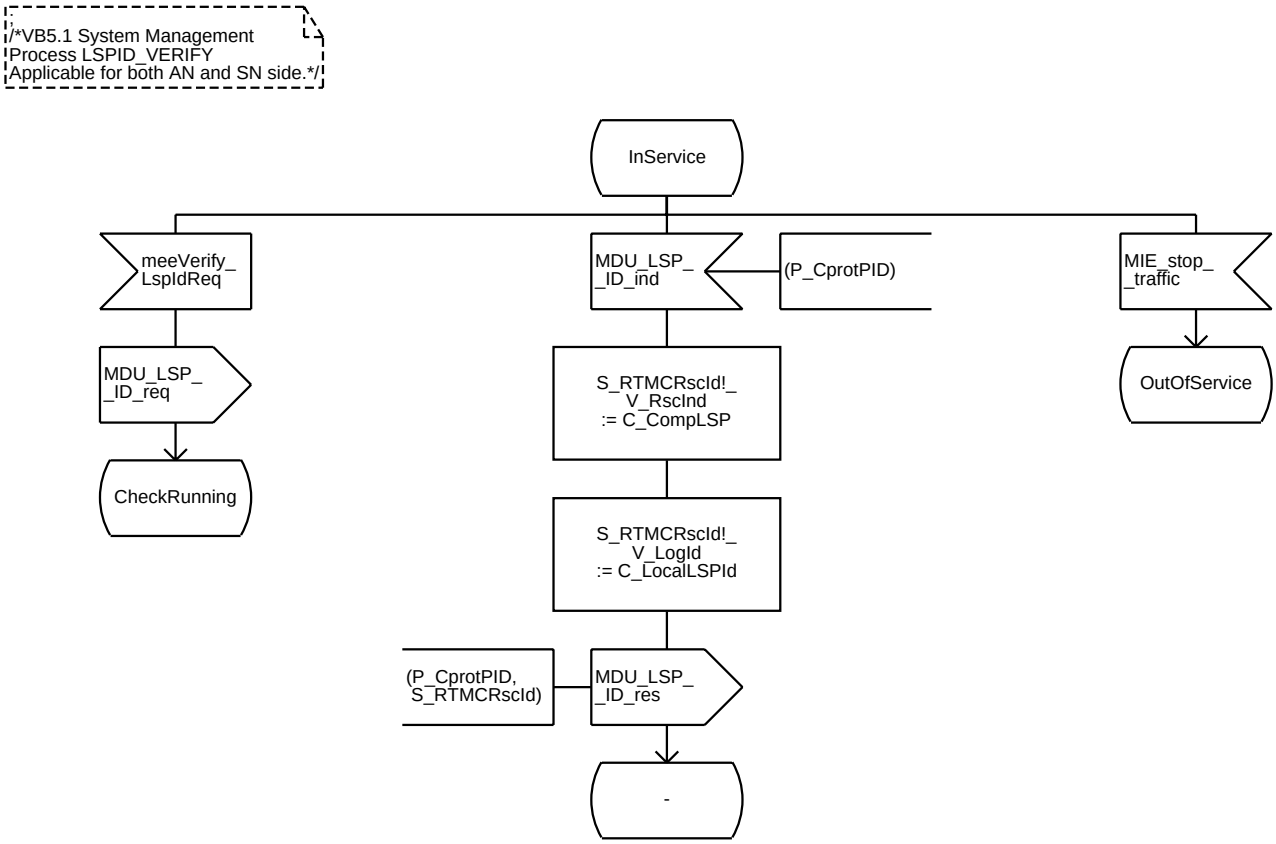

(SL_LSPID_CTR)
CTR
(SL_CTR_LSPID)
(SL_LSPID_ENV)
ENVGATE
(SL_ENV_LSPID)
Process Type LSPID_VERIFY
1(5)
/*VB5.1 System Management
Process LSPID_VERIFY
Applicable for both AN and SN side.*/
/* Definitions and declarations
for Process LSPID_VERIFY */
/* State descriptions
OutOfService: Initial state of LSPID_VERIFY. Process is waiting for activation.
CheckDuringStartup: Verification of LSP ID was triggered during startup.
InService: Normal operation after start traffic. No check procedure is running.
CheckRunning: Waiting for confirmation from the peer entity. */
/*Signal data declarations*/
DCL
S_RTMCRscId ST_RscId; /*Resource Identifier Information Element reported from/to the peer side*/
/**/
DCL
P_CprotPID PID; /*PID of CPROT process responsible for transaction*/
/**/
DCL
V_Result IT_Result; /*Result of transaction*/
(SL_CPROT_LSPID)
CPROT
(SL_LSPID_CPROT)

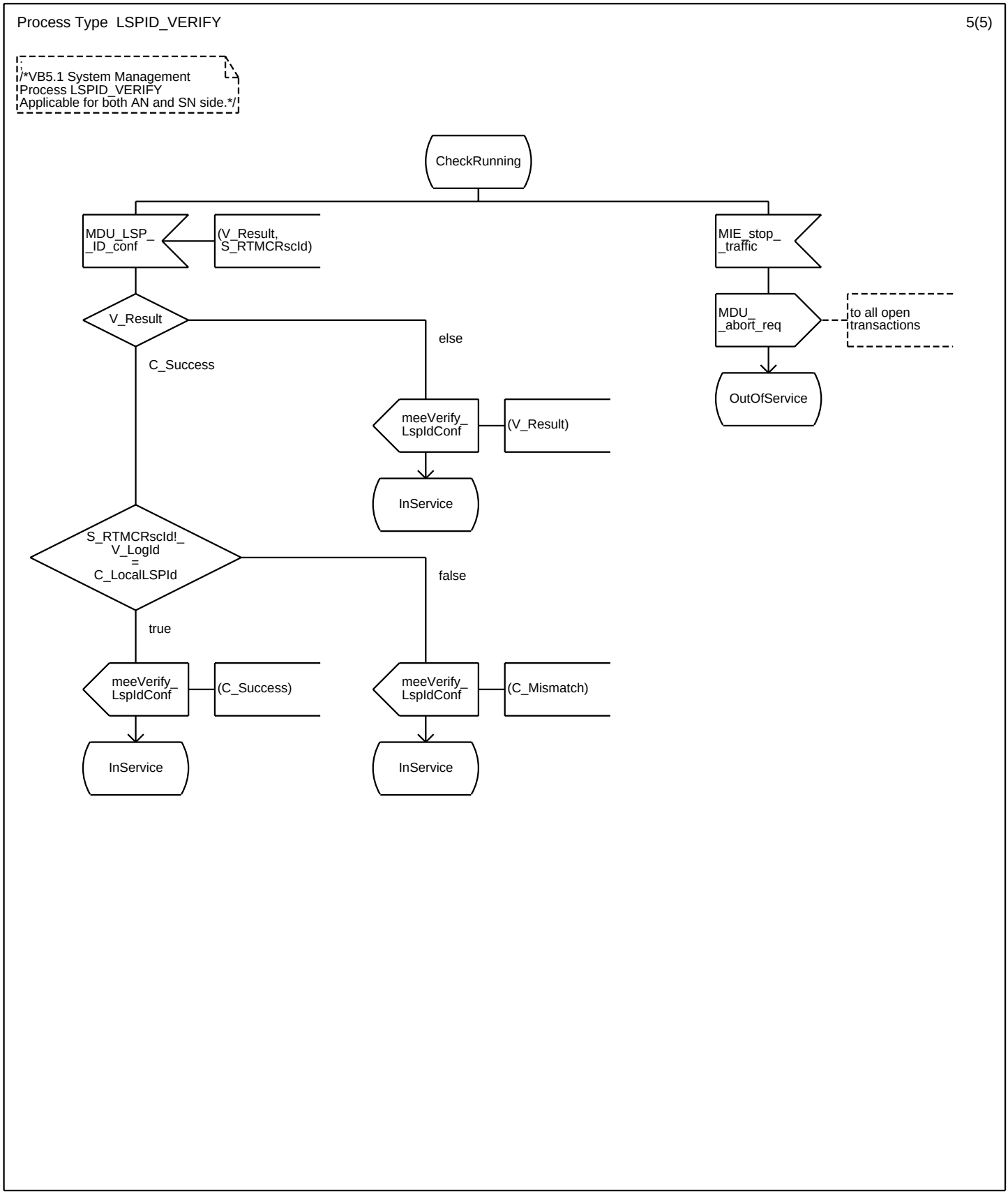
Process Type LSPID_VERIFY 2(5)

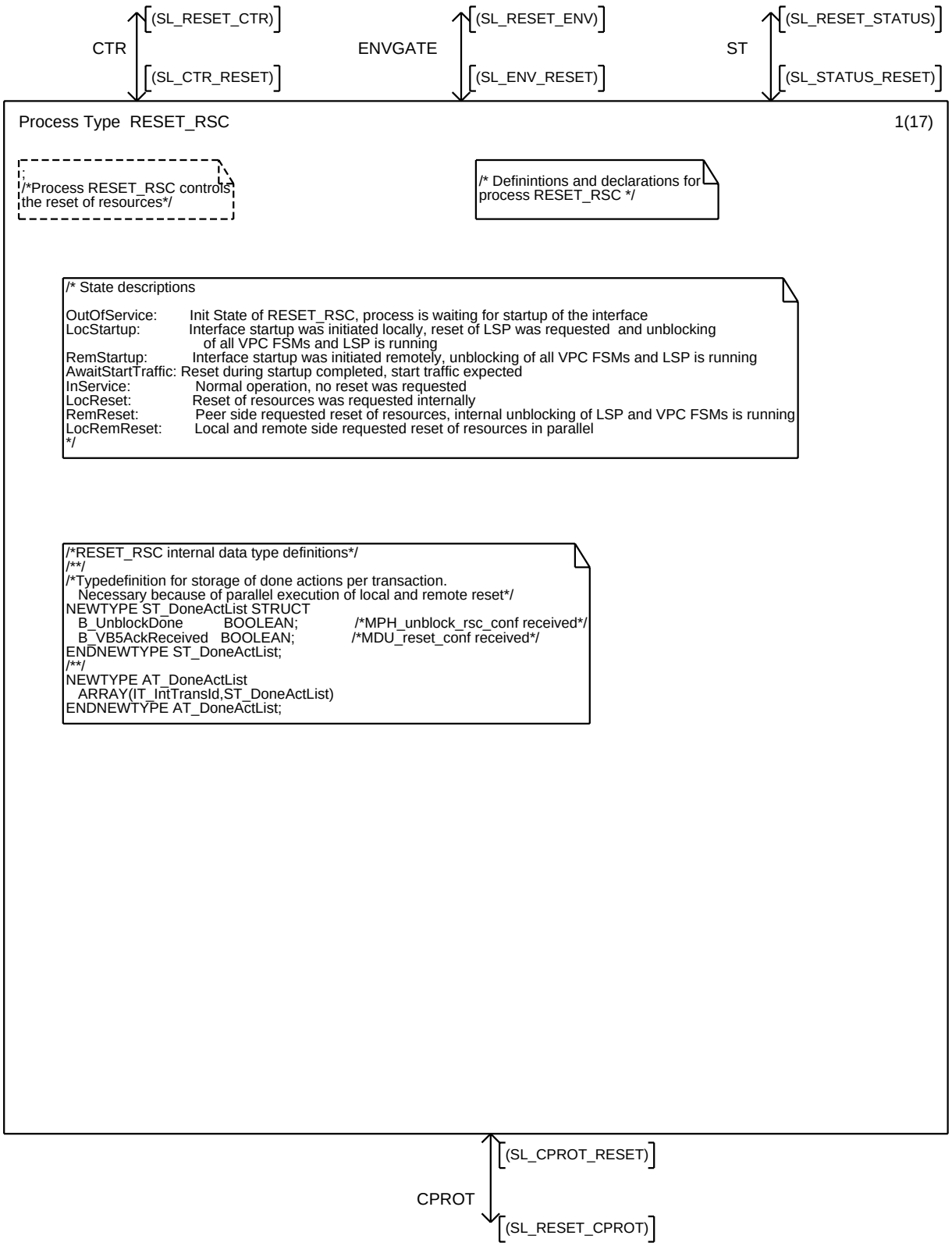



Process Type LSPID_VERIFY
3(5)
/*VB5.1 System Management
Process LSPID_VERIFY
Applicable for both AN and SN side.*/
Check_
DuringStartup
MDU_LSP_
_ID_conf
(V_Result,
S_RTMCRscId)
MIE_stop_
_traffic
OutOfService
V_Result
else
C_Success
MIE_verify_
_LSP_ID_
_conf
(V_Result)
OutOfService
S_RTMCRscId!_
V_LogId
=
C_LocalLSPId
false
true
MIE_verify_
_LSP_ID_
_conf
(C_Success)
MIE_verify_
_LSP_ID_
_conf
(C_Mismatch)
OutOfService
OutOfService

Process Type LSPID_VERIFY 4(5)




Process Type LSPID_VERIFY
5(5)
/*VB5.1 System Management
Process LSPID_VERIFY
Applicable for both AN and SN side.*/
CheckRunning
MDU_LSP_
_ID_conf
(V_Result,
S_RTMCRscId)
MIE_stop_
_traffic
V_Result
else
C_Success
MDU_
_abort_req
to all open
transactions
OutOfService
meeVerify_
LspIdConf
(V_Result)
InService
S_RTMCRscId!_
V_LogId
=
C_LocalLSPId
false
true
meeVerify_
LspIdConf
(C_Success)
meeVerify_
LspIdConf
(C_Mismatch)
InService
InService

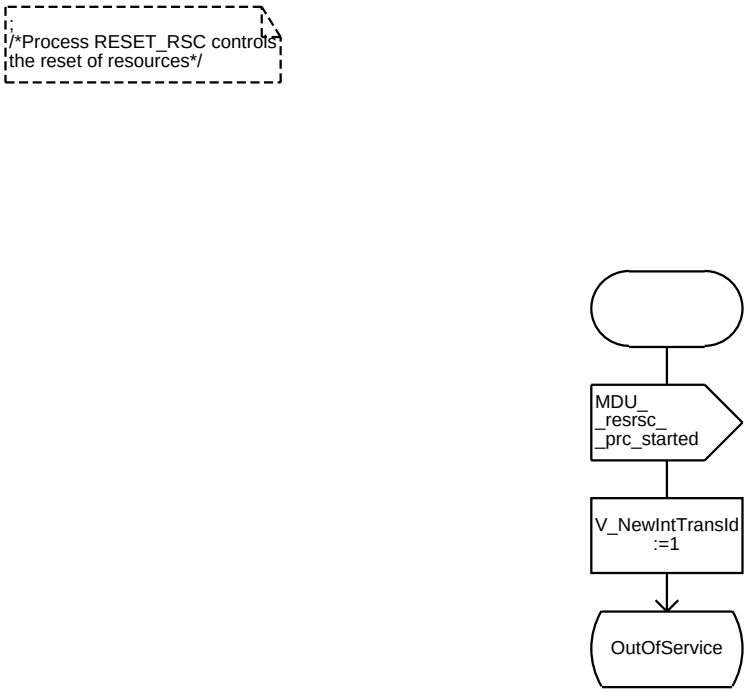

(SL_RESET_CTR)
CTR
(SL_CTR_RESET)
(SL_RESET_ENV)
ENVGATE
(SL_ENV_RESET)
(SL_RESET_STATUS)
ST
(SL_STATUS_RESET)
Process Type RESET_RSC
1(17)
/*Process RESET_RSC controls the reset of resources*/
/* Definintions and declarations for process RESET_RSC */
/* State descriptions
OutOfService: Init State of RESET_RSC, process is waiting for startup of the interface
LocStartup: Interface startup was initiated locally, reset of LSP was requested and unblocking of all VPC FSMs and LSP is running
RemStartup: Interface startup was initiated remotely, unblocking of all VPC FSMs and LSP is running
AwaitStartTraffic: Reset during startup completed, start traffic expected
InService: Normal operation, no reset was requested
LocReset: Reset of resources was requested internally
RemReset: Peer side requested reset of resources, internal unblocking of LSP and VPC FSMs is running
LocRemReset: Local and remote side requested reset of resources in parallel
*/
/*RESET_RSC internal data type definitions*/
/**/
/*Typedefinition for storage of done actions per transaction.
Necessary because of parallel execution of local and remote reset*/
NEWTYPE ST_DoneActList STRUCT
B_UnblockDone BOOLEAN; /*MPH_unblock_rsc_conf received*/
B_VB5AckReceived BOOLEAN; /*MDU_reset_conf received*/
ENDNEWTYPE ST_DoneActList;
/**/
NEWTYPE AT_DoneActList
ARRAY(IT_IntTransId,ST_DoneActList)
ENDNEWTYPE AT_DoneActList;
(SL_CPROT_RESET)
CPROT
(SL_RESET_CPROT)

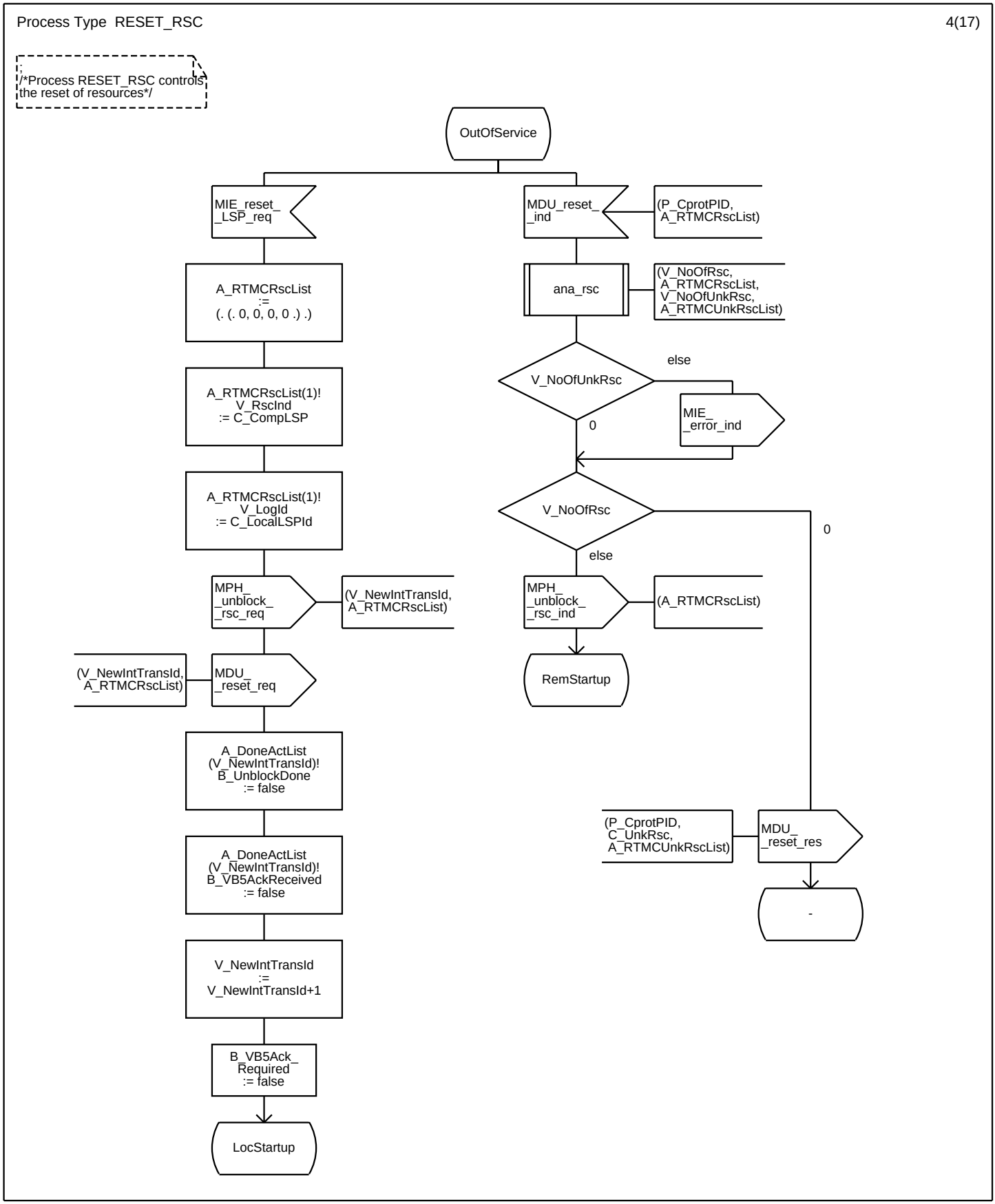
```
;
/*Process RESET_RSC controls
the reset of resources*/
```

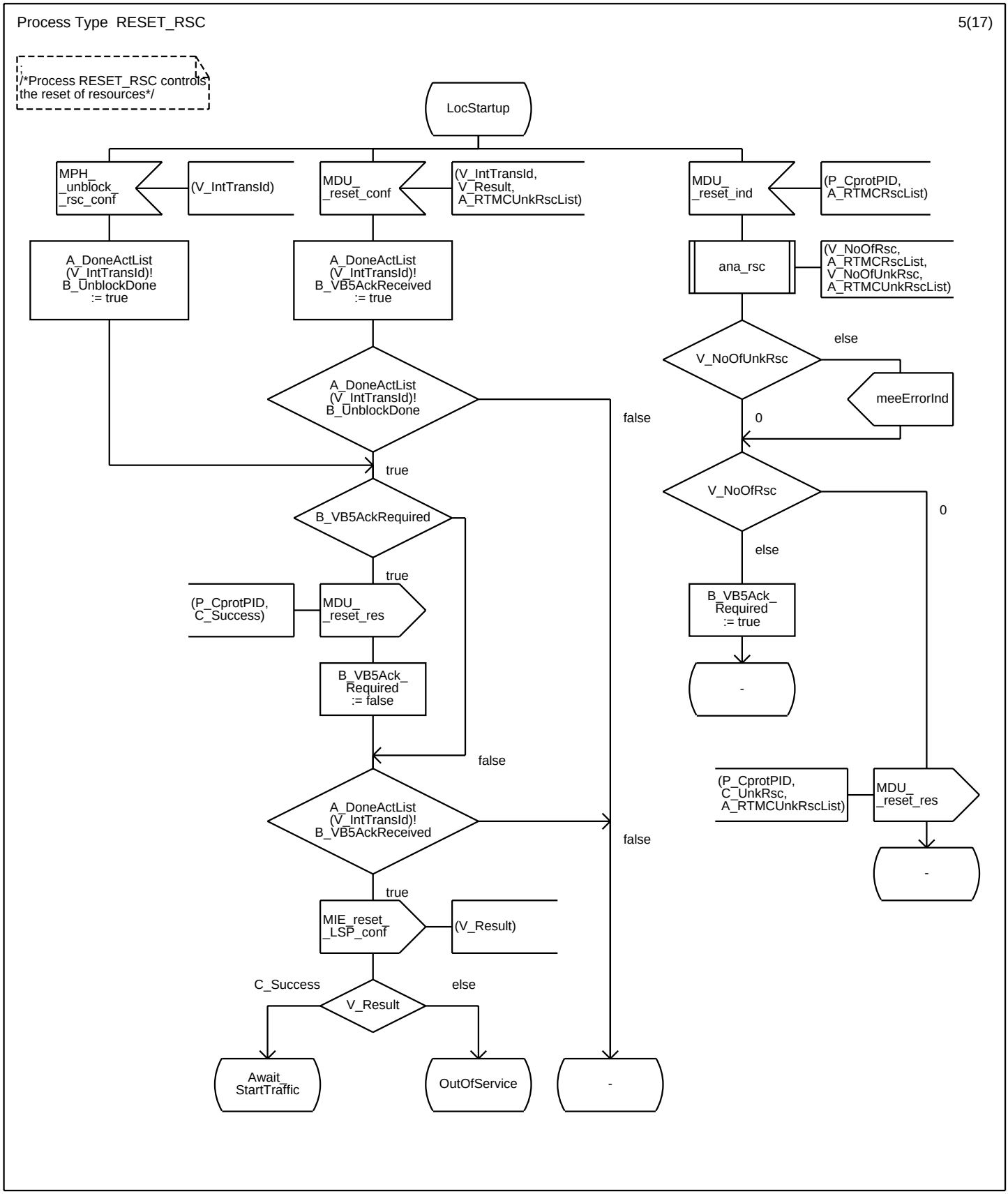
```
/* Definintions and declarations for
process RESET_RSC */
```

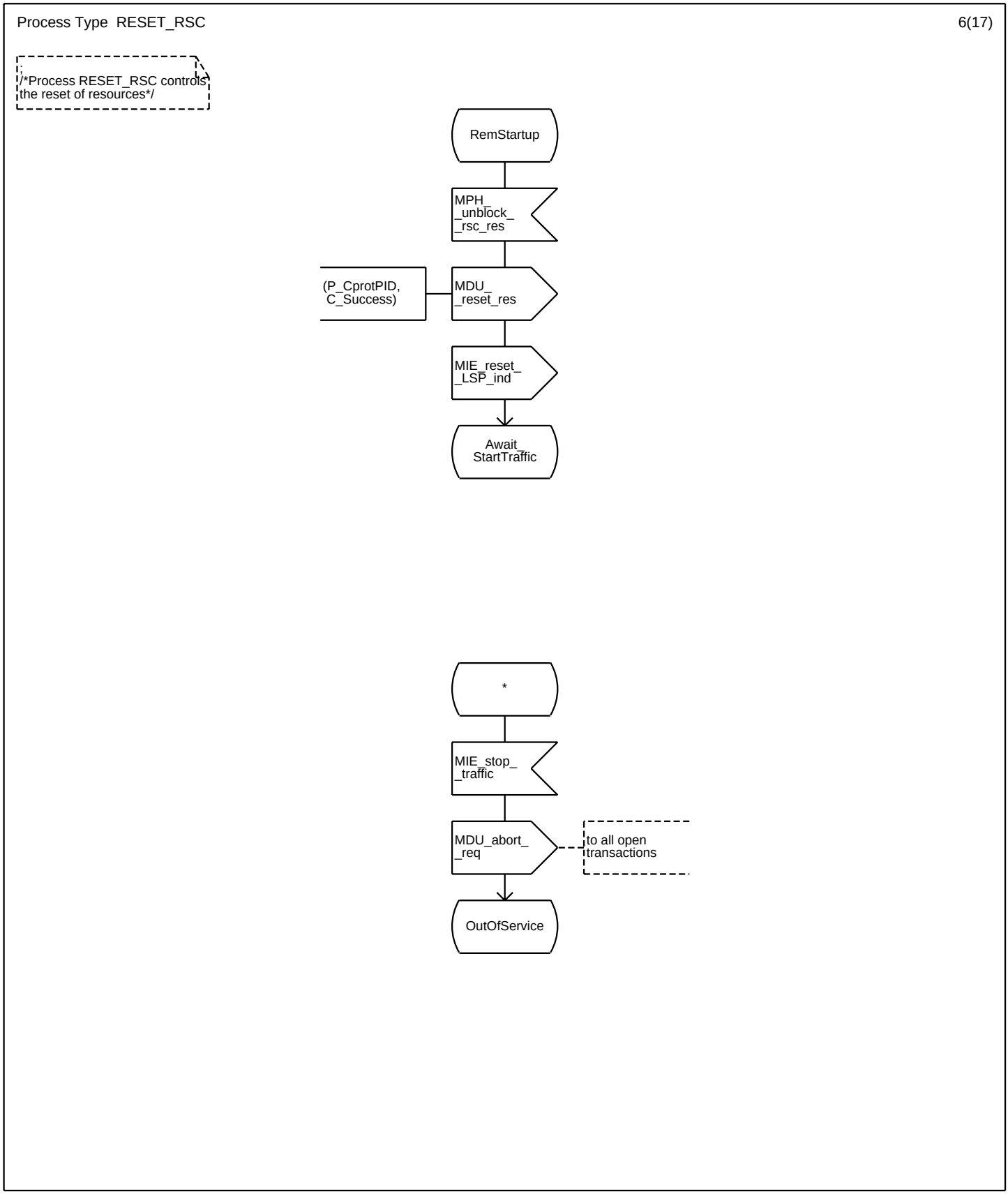
```
/* Internal variables and constants */
/**/
DCL
A_DoneActList   AT_DoneActList;    /*Done Actions per transaction*/
/**/
DCL
B_VB5AckRequired   BOOLEAN;     /*MDU_reset_res to be sent*/
/**/
DCL
V_NoOfRsc INTEGER; /*Number of resources*/
/**/
DCL
V_NoOfUnkRsc INTEGER; /*Number of unknown resources*/
/**/
DCL
V_NewIntTransId IT_IntTransId; /*Variable for indexing IntTransIdList*/
/**/
DCL
V_NoOfMsg INTEGER; /*Number of messages per transaction*/
/**/
DCL A_IntTransList AT_IntTransList; /*Variable for storage of transaction Id data*/
```

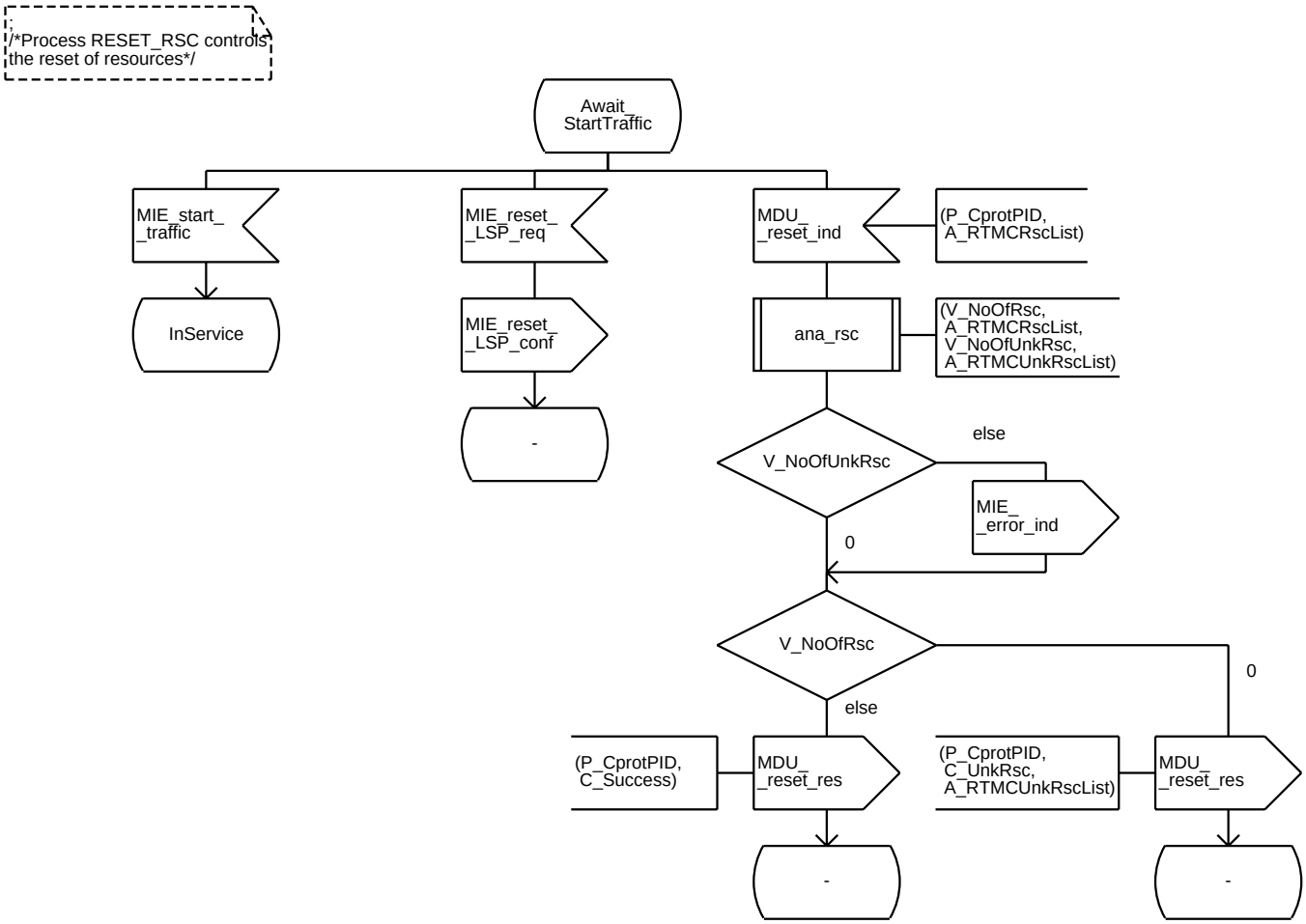
```
/*Signal data declarations*/
DCL
A_RTMCRscList AT_RscList, /*Resource Identifier Information Element List reported from/to the peer side*/
A_RscList            AT_RscList, /*Resource Identifier List reported from the local side*/
A_UnkRscList      AT_RscList, /*Resource Identifier List reported to the  local side*/
A_RTMCUnkRscList AT_RscList; /*Unknown Resource Identifier IE List reported from/to the peer side*/
/**/
DCL
P_CprotPID PID; /*PID of CPROT process responsible for transaction*/
/**/
DCL
V_Result       IT_Result; /*Result*/
/**/
DCL
V_IntTransId IT_IntTransId; /*Variable for indexing IntTransIdList*/
```

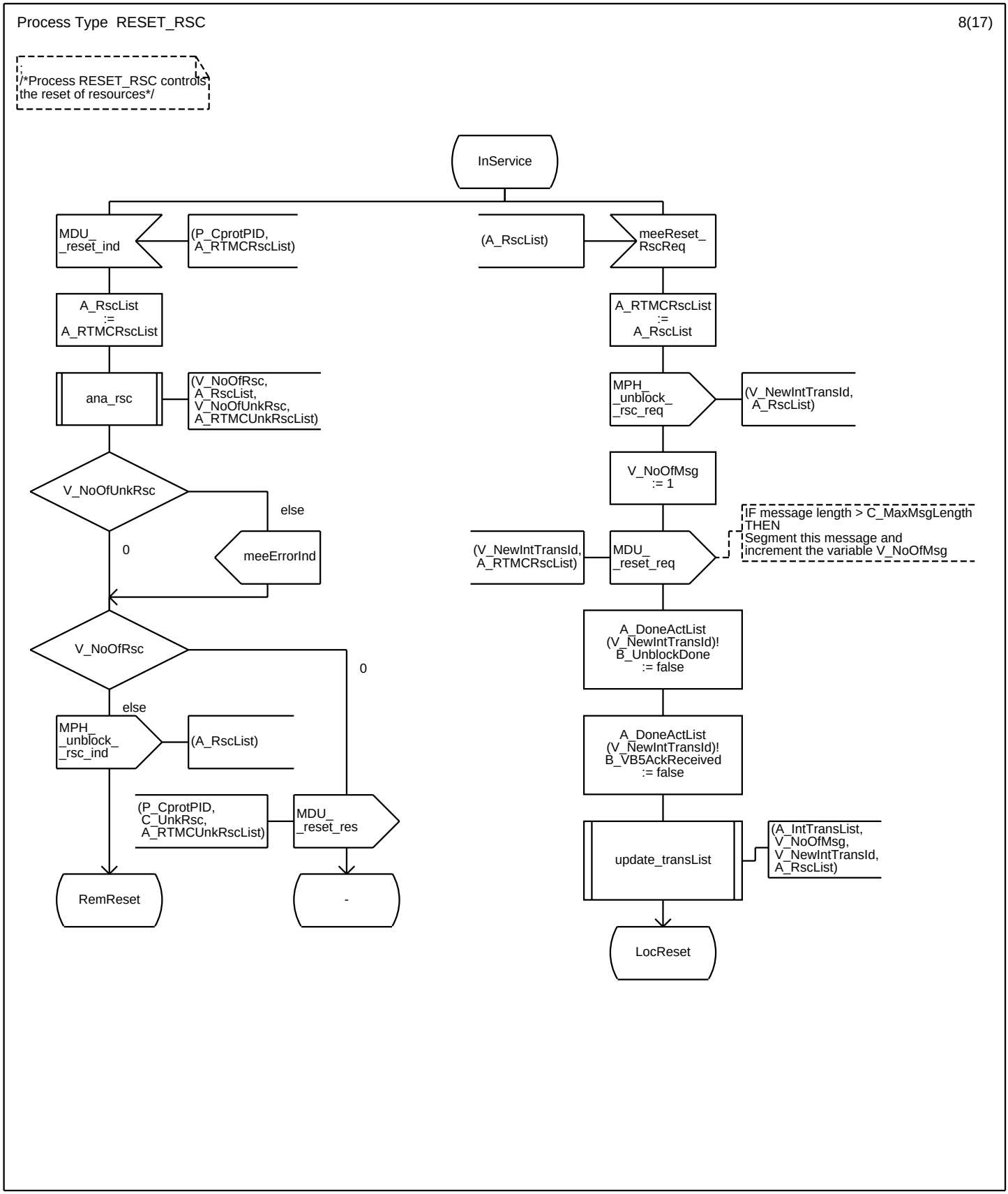

;
/*Process RESET_RSC controls the reset of resources*/
MDU_
resrsc
_prc_started
V_NewIntTransId
:=1
OutOfService

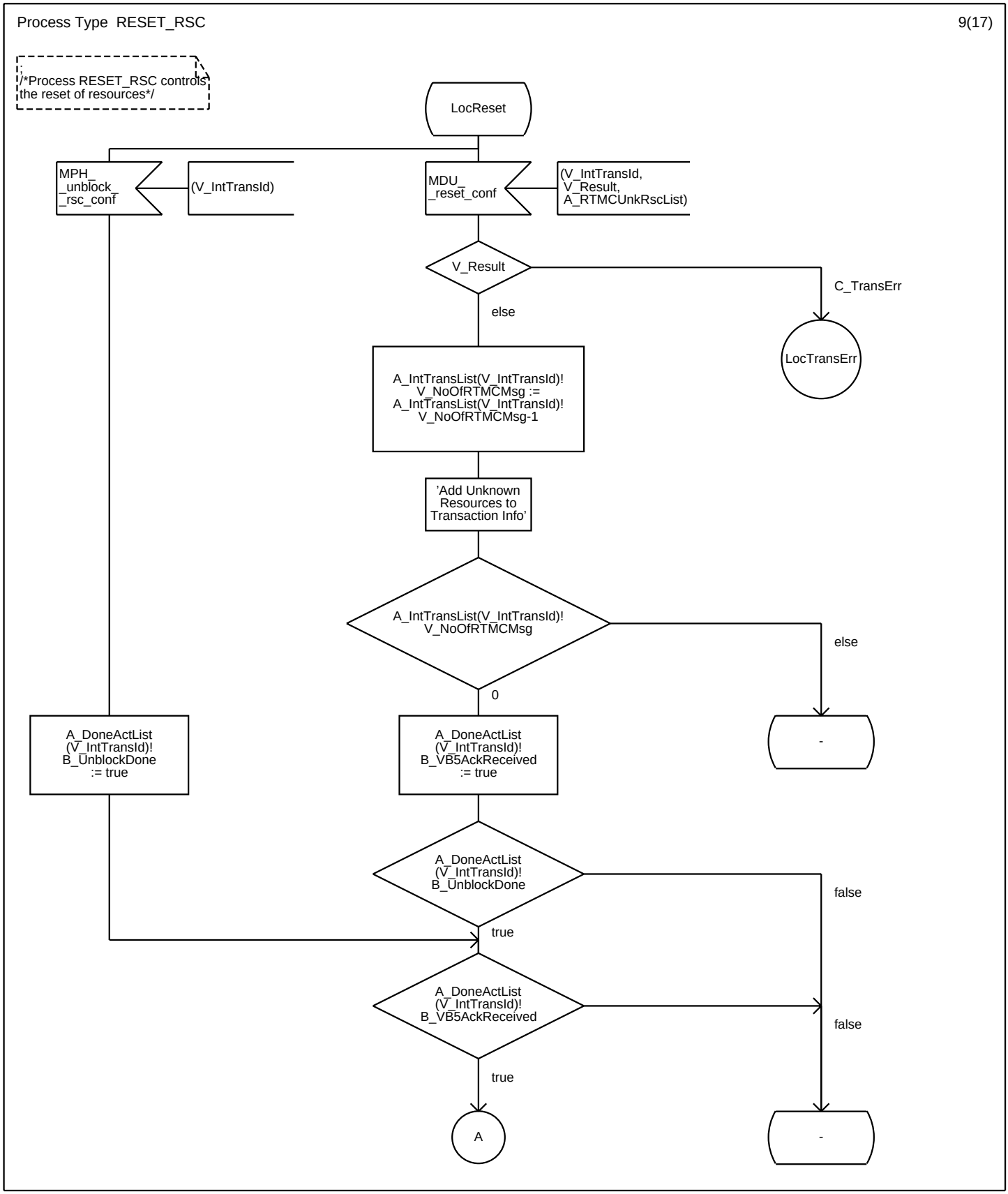

;
/*Process RESET_RSC controls the reset of resources*/
OutOfService
MIE_reset_ _LSP_req
MDU_reset_ _ind
(P_CprotPID, A_RTMCRscList)
A_RTMCRscList := (. (. 0, 0, 0, 0 .) .)
ana_rsc
(V_NoOfRsc, A_RTMCRscList, V_NoOfUnkRsc, A_RTMCUnkRscList)
V_NoOfUnkRsc
else
0
MIE_ _error_ind
A_RTMCRscList(1)! V_RscInd := C_CompLSP
A_RTMCRscList(1)! V_LogId := C_LocalLSPId
V_NoOfRsc
0
else
MPH_ _unblock_ _rsc_req
(V_NewIntTransId, A_RTMCRscList)
MPH_ _unblock_ _rsc_ind
(A_RTMCRscList)
(V_NewIntTransId, A_RTMCRscList)
MDU_ _reset_req
RemStartup
A_DoneActList (V_NewIntTransId)! B_UnblockDone := false
(P_CprotPID, C_UnkRsc, A_RTMCUnkRscList)
MDU_ _reset_res
A_DoneActList (V_NewIntTransId)! B_VB5AckReceived := false
-
V_NewIntTransId := V_NewIntTransId+1
B_VB5Ack_ Required := false
LocStartup

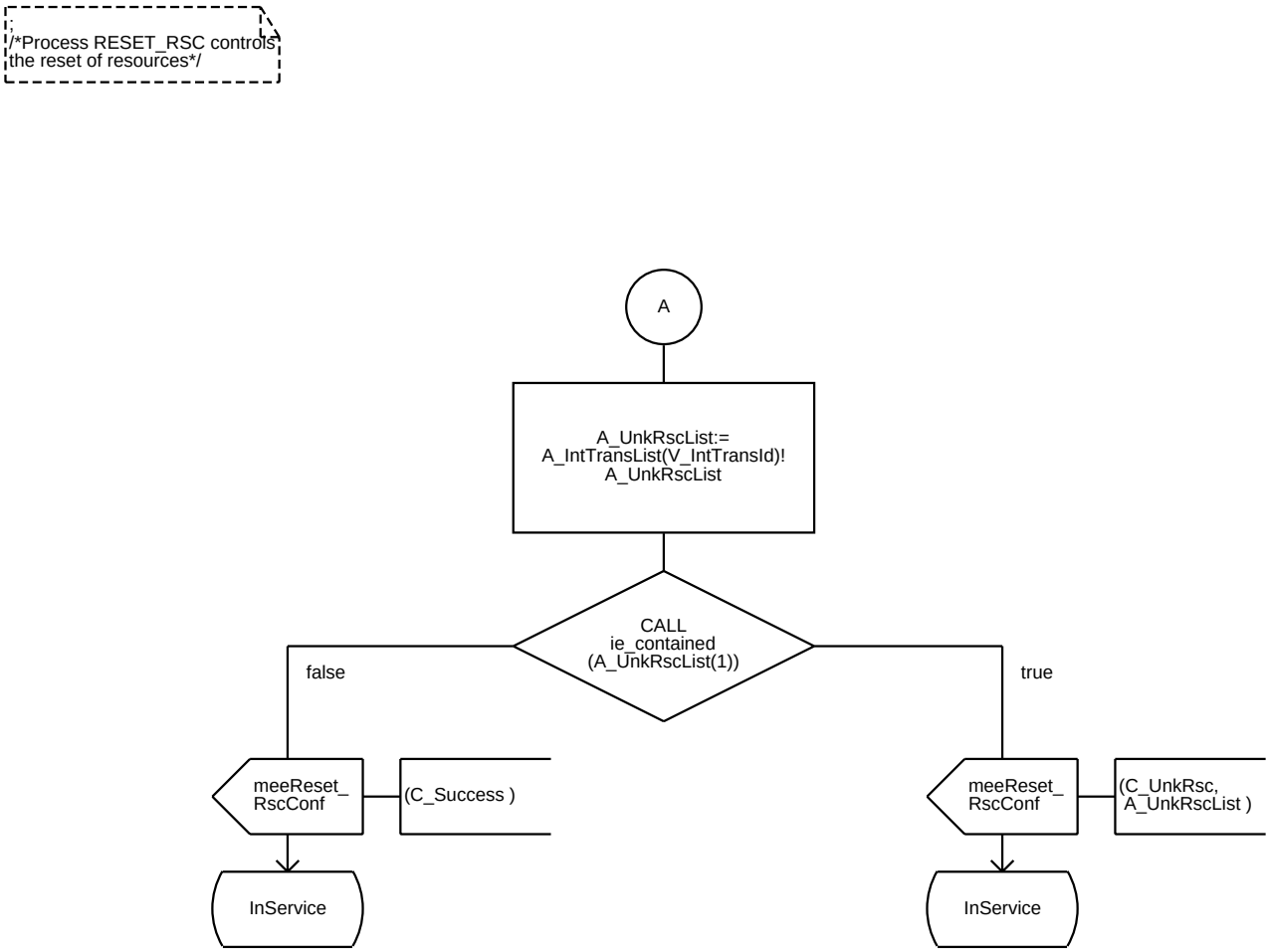

Process Type RESET_RSC
5(17)
;
/*Process RESET_RSC controls the reset of resources*/
LocStartup
MPH_ _unblock_ _rsc_conf
(V_IntTransId)
MDU_ _reset_conf
(V_IntTransId, V_Result, A_RTMCUnkRscList)
MDU_ _reset_ind
(P_CprotPID, A_RTMCRscList)
A_DoneActList (V_IntTransId)! B_UnblockDone := true
A_DoneActList (V_IntTransId)! B_VB5AckReceived := true
ana_rsc
(V_NoOfRsc, A_RTMCRscList, V_NoOfUnkRsc, A_RTMCUnkRscList)
V_NoOfUnkRsc
else
meeErrorInd
0
A_DoneActList (V_IntTransId)! B_UnblockDone
false
true
V_NoOfRsc
0
B_VB5AckRequired
else
true
(P_CprotPID, C_Success)
MDU_ _reset_res
B_VB5Ack_ Required := true
B_VB5Ack_ Required := false
-
false
(P_CprotPID, C_UnkRsc, A_RTMCUnkRscList)
MDU_ _reset_res
A_DoneActList (V_IntTransId)! B_VB5AckReceived
false
-
true
MIE_reset_ _LSP_conf
(V_Result)
C_Success
else
V_Result
Await_ StartTraffic
OutOfService
-

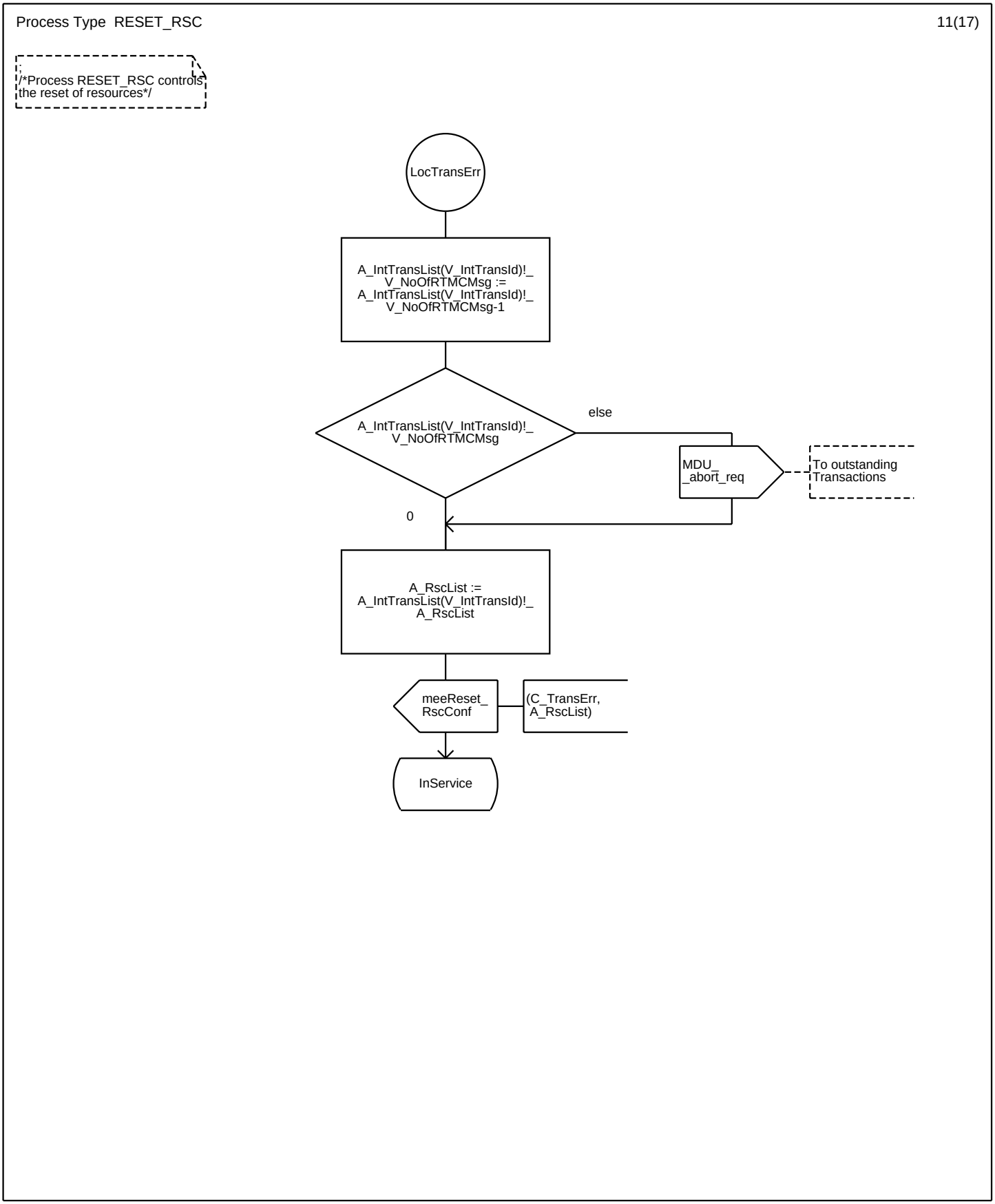

;
/*Process RESET_RSC controls the reset of resources*/
RemStartup
MPH_
unblock
_rsc_res
(P_CprotPID,
C_Success)
MDU_
_reset_res
MIE_reset_
_LSP_ind
Await_
StartTraffic
*
MIE_stop_
_traffic
MDU_abort_
_req
to all open
transactions
OutOfService


;
/*Process RESET_RSC controls the reset of resources*/
Await_ StartTraffic
MIE_start_ _traffic
MIE_reset_ _LSP_req
MDU_ _reset_ind
(P_CprotPID, A_RTMCRscList)
InService
MIE_reset_ _LSP_conf
ana_rsc
(V_NoOfRsc, A_RTMCRscList, V_NoOfUnkRsc, A_RTMCUnkRscList)
-
V_NoOfUnkRsc
else
MIE_ _error_ind
0
V_NoOfRsc
0
else
(P_CprotPID, C_Success)
MDU_ _reset_res
(P_CprotPID, C_UnkRsc, A_RTMCUnkRscList)
MDU_ _reset_res
-
-

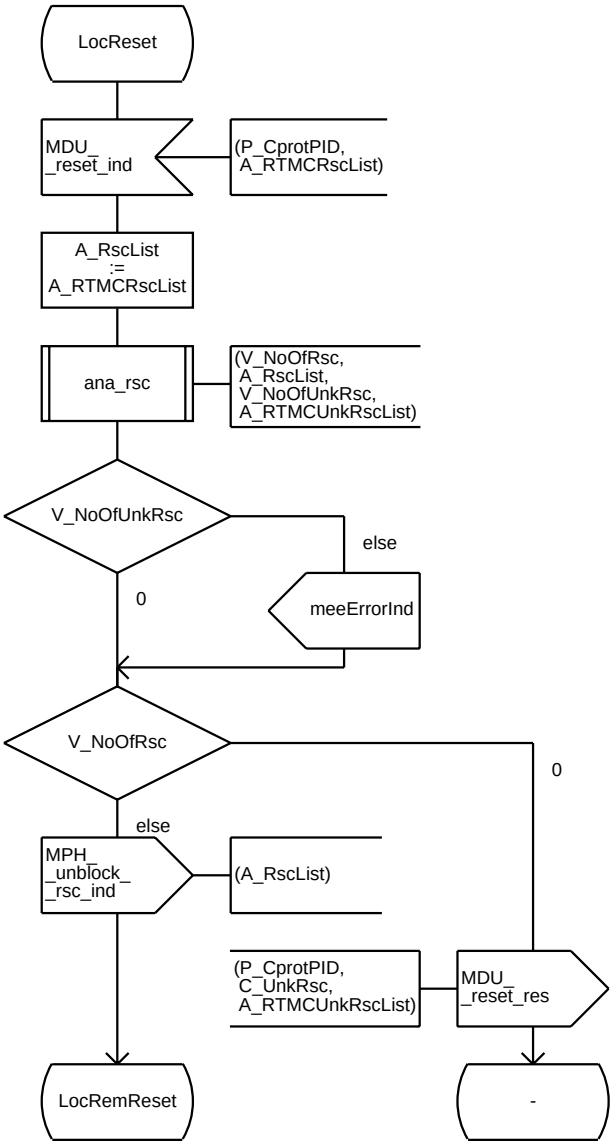

Process Type RESET_RSC
8(17)
/*Process RESET_RSC controls the reset of resources*/
InService
MDU_ _reset_ind
(P_CprotPID, A_RTMCRscList)
(A_RscList)
meeReset_ RscReq
A_RscList := A_RTMCRscList
A_RTMCRscList := A_RscList
ana_rsc
(V_NoOfRsc, A_RscList, V_NoOfUnkRsc, A_RTMCUnkRscList)
MPH_ _unblock_ _rsc_req
(V_NewIntTransId, A_RscList)
V_NoOfUnkRsc
else
0
meeErrorInd
V_NoOfMsg := 1
IF message length > C_MaxMsgLength THEN Segment this message and increment the variable V_NoOfMsg
(V_NewIntTransId, A_RTMCRscList)
MDU_ _reset_req
V_NoOfRsc
0
else
MPH_ _unblock_ _rsc_ind
(A_RscList)
A_DoneActList (V_NewIntTransId)! B_UnblockDone := false
A_DoneActList (V_NewIntTransId)! B_VB5AckReceived := false
(P_CprotPID, C_UnkRsc, A_RTMCUnkRscList)
MDU_ _reset_res
update_transList
(A_IntTransList, V_NoOfMsg, V_NewIntTransId, A_RscList)
RemReset
-
LocReset

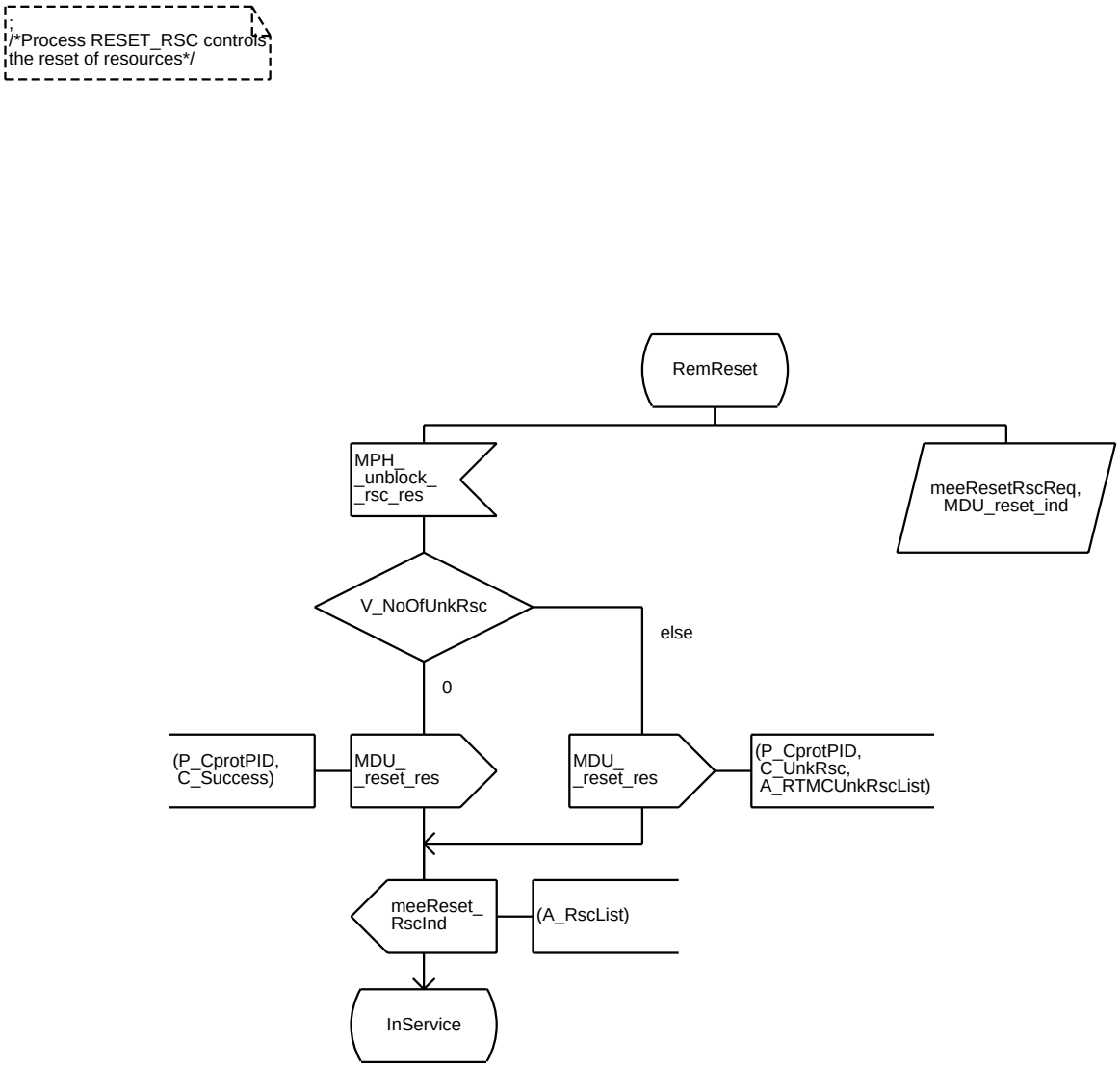

Process Type RESET_RSC
9(17)
;
/*Process RESET_RSC controls the reset of resources*/
LocReset
MPH_
unblock
_rsc_conf
(V_IntTransId)
MDU_
_reset_conf
(V_IntTransId,
V_Result,
A_RTMCUnkRscList)
V_Result
C_TransErr
else
LocTransErr
A_IntTransList(V_IntTransId)!
V_NoOfRTMCMsg :=
A_IntTransList(V_IntTransId)!
V_NoOfRTMCMsg-1
'Add Unknown
Resources to
Transaction Info'
A_IntTransList(V_IntTransId)!
V_NoOfRTMCMsg
else
0
-
A_DoneActList
(V_IntTransId)!
B_UnblockDone
:= true
A_DoneActList
(V_IntTransId)!
B_VB5AckReceived
:= true
A_DoneActList
(V_IntTransId)!
B_UnblockDone
false
true
A_DoneActList
(V_IntTransId)!
B_VB5AckReceived
false
true
A
-

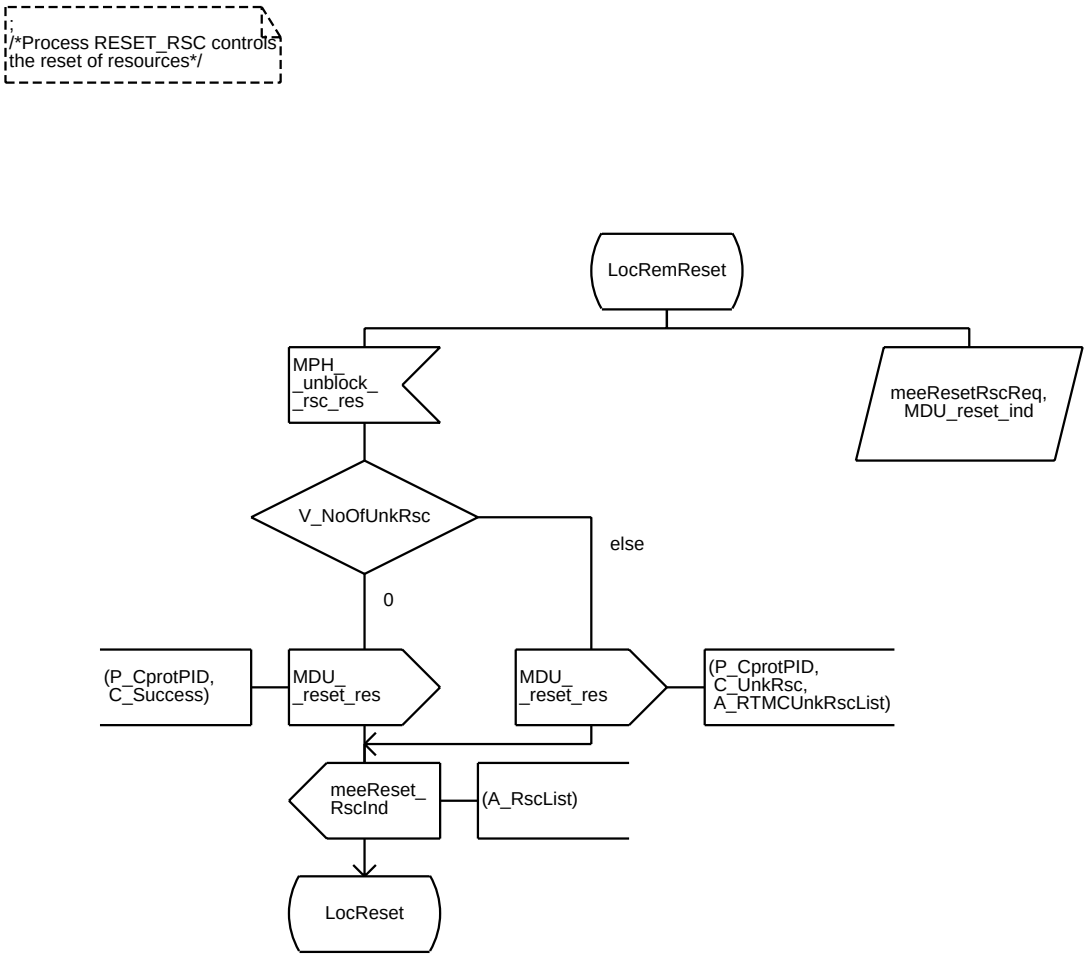

;
/*Process RESET_RSC controls the reset of resources*/
A
A_UnkRscList:=
A_IntTransList(V_IntTransId)!
A_UnkRscList
CALL
ie_contained
(A_UnkRscList(1))
false
true
meeReset_
RscConf
(C_Success)
meeReset_
RscConf
(C_UnkRsc,
A_UnkRscList)
InService
InService

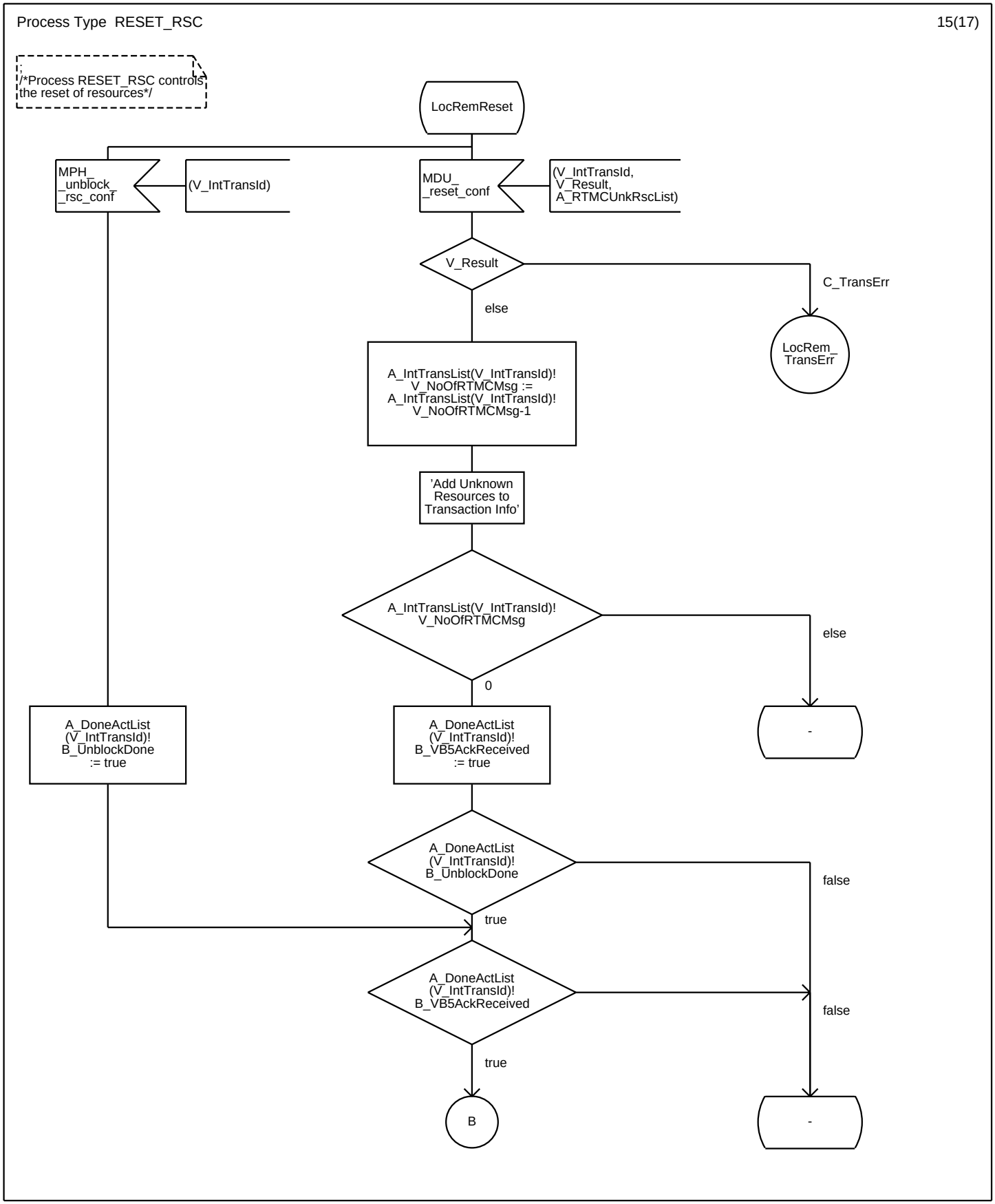

;
/*Process RESET_RSC controls the reset of resources*/
LocTransErr
A_IntTransList(V_IntTransId)!_ V_NoOfRTMCMsg := A_IntTransList(V_IntTransId)!_ V_NoOfRTMCMsg-1
A_IntTransList(V_IntTransId)!_ V_NoOfRTMCMsg
else
MDU_ _abort_req
To outstanding Transactions
0
A_RscList := A_IntTransList(V_IntTransId)!_ A_RscList
meeReset_ RscConf
(C_TransErr, A_RscList)
InService

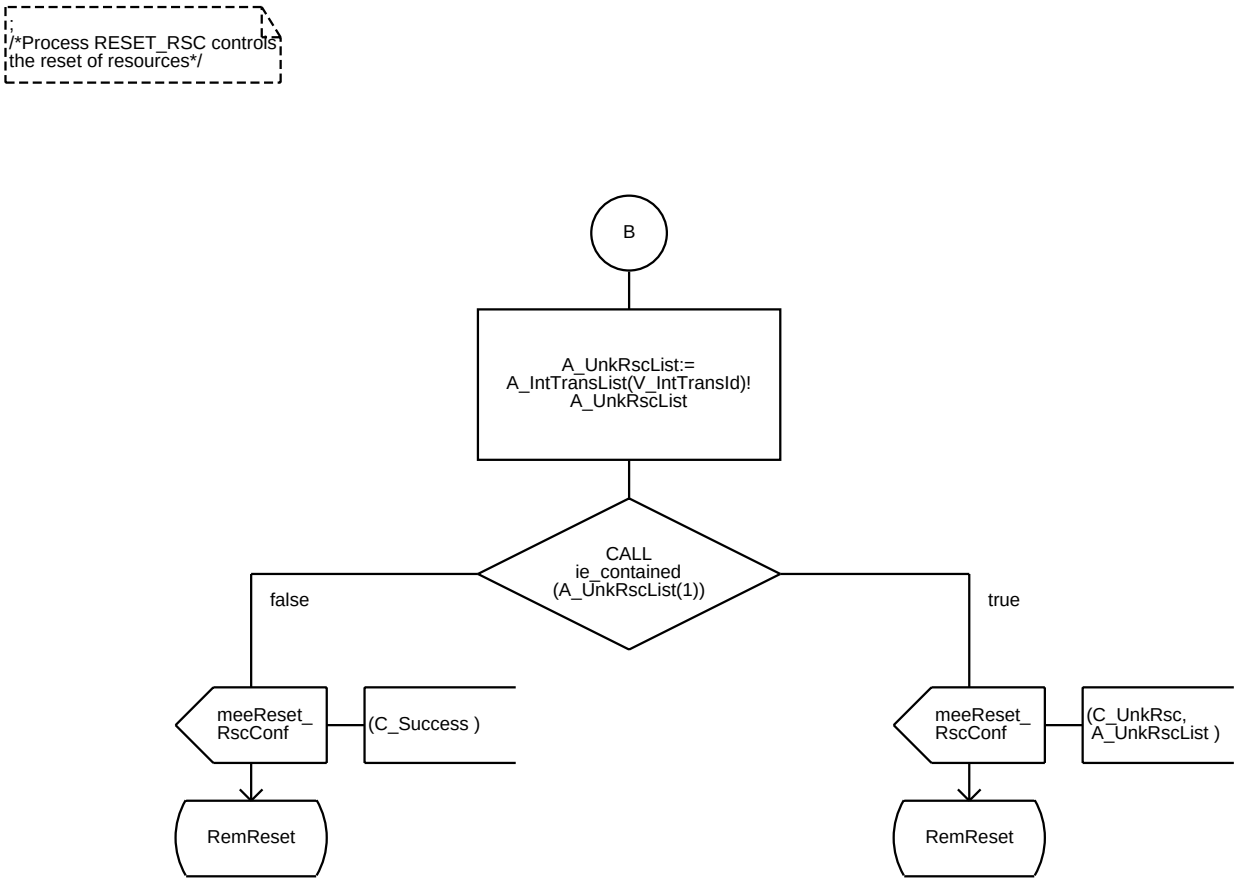
;
/*Process RESET_RSC controls the reset of resources*/

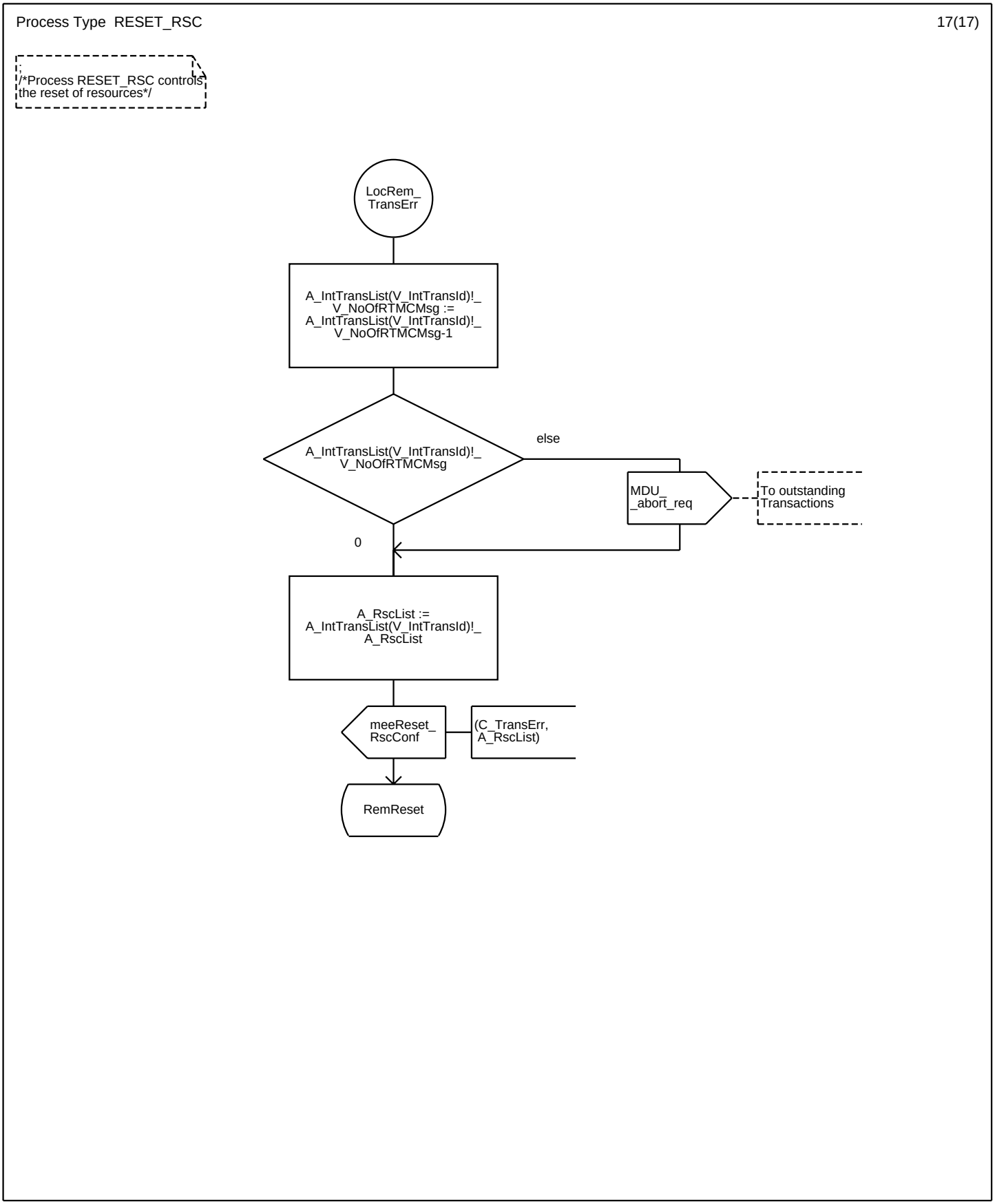


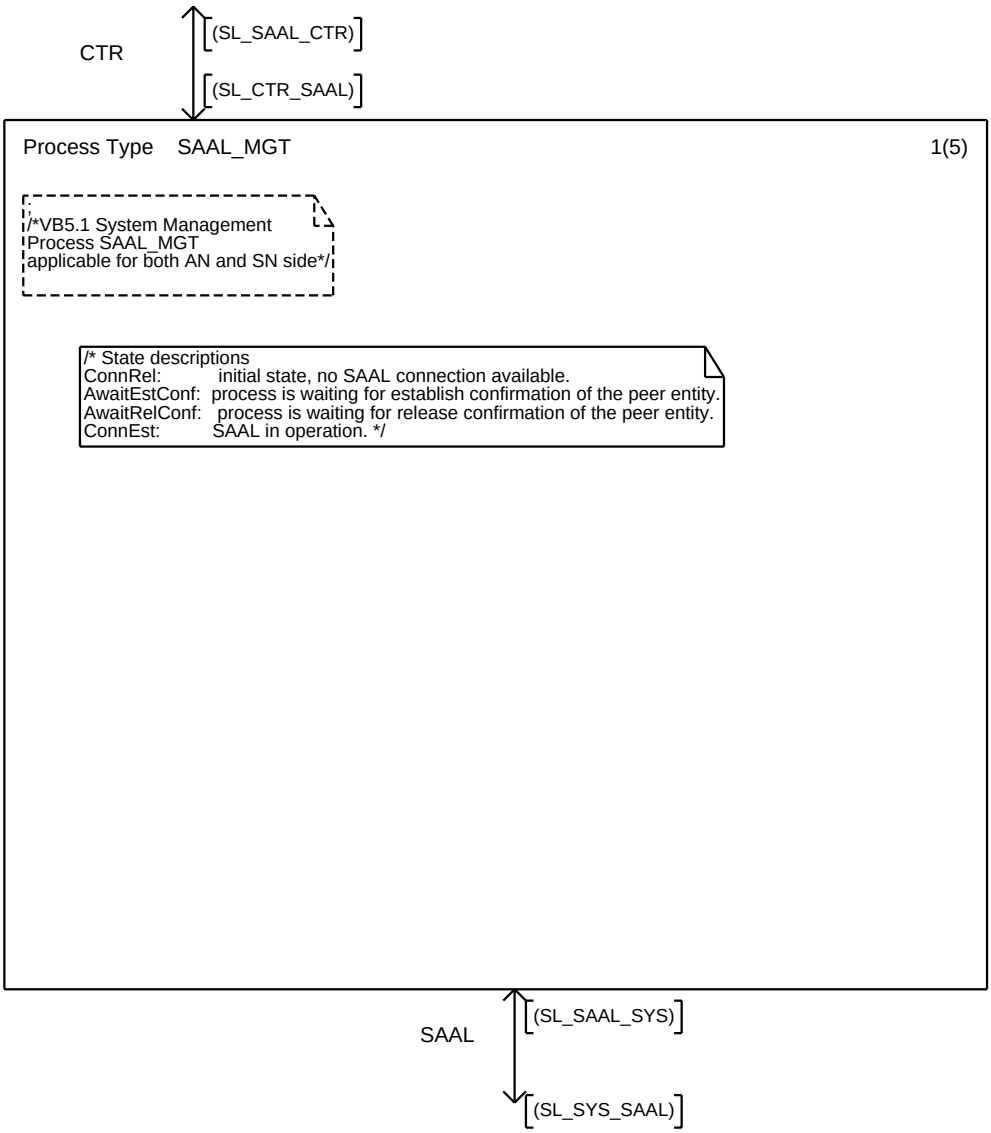

;
/*Process RESET_RSC controls the reset of resources*/
RemReset
MPH_ _unblock_ _rsc_res
meeResetRscReq, MDU_reset_ind
V_NoOfUnkRsc
else
0
(P_CprotPID, C_Success)
MDU_ _reset_res
MDU_ _reset_res
(P_CprotPID, C_UnkRsc, A_RTMCUnkRscList)
meeReset_ RscInd
(A_RscList)
InService


;
/*Process RESET_RSC controls the reset of resources*/
LocRemReset
MPH_ _unblock_ _rsc_res
meeResetRscReq, MDU_reset_ind
V_NoOfUnkRsc
else
0
(P_CprotPID, C_Success)
MDU_ _reset_res
MDU_ _reset_res
(P_CprotPID, C_UnkRsc, A_RTMCUnkRscList)
meeReset_ RscInd
(A_RscList)
LocReset

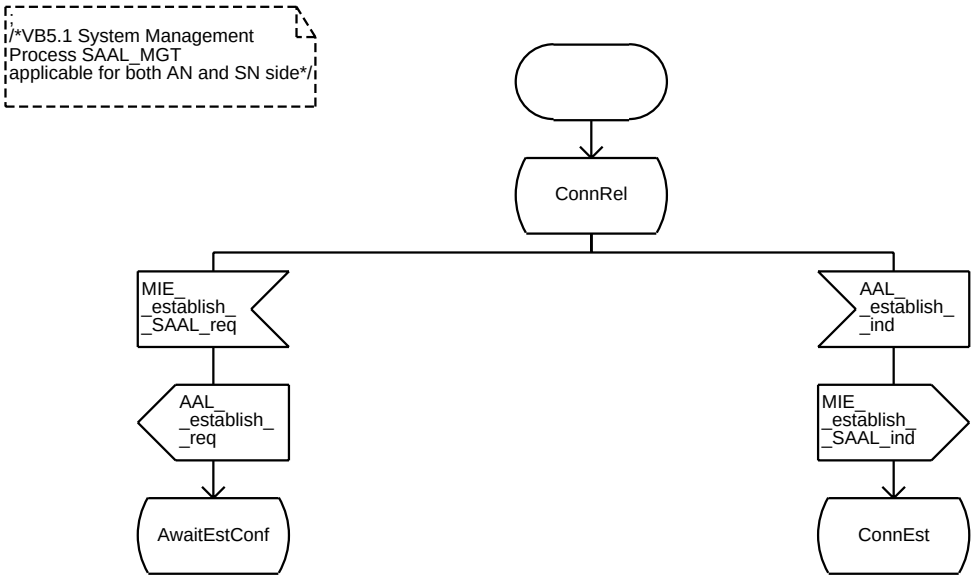

Process Type RESET_RSC
15(17)
/*Process RESET_RSC controls the reset of resources*/
LocRemReset
MPH_ _unblock_ _rsc_conf
(V_IntTransId)
MDU_ _reset_conf
(V_IntTransId, V_Result, A_RTMCUnkRscList)
V_Result
C_TransErr
else
LocRem_ TransErr
A_IntTransList(V_IntTransId)! V_NoOfRTMCMsg := A_IntTransList(V_IntTransId)! V_NoOfRTMCMsg-1
'Add Unknown Resources to Transaction Info'
A_IntTransList(V_IntTransId)! V_NoOfRTMCMsg
else
0
-
A_DoneActList (V_IntTransId)! B_UnblockDone := true
A_DoneActList (V_IntTransId)! B_VB5AckReceived := true
A_DoneActList (V_IntTransId)! B_UnblockDone
false
true
A_DoneActList (V_IntTransId)! B_VB5AckReceived
false
true
B
-


;
/*Process RESET_RSC controls
the reset of resources*/
B
A_UnkRscList:=
A_IntTransList(V_IntTransId)!
A_UnkRscList
CALL
ie_contained
(A_UnkRscList(1))
false
true
meeReset_
RscConf
(C_Success)
RemReset
meeReset_
RscConf
(C_UnkRsc,
A_UnkRscList)
RemReset

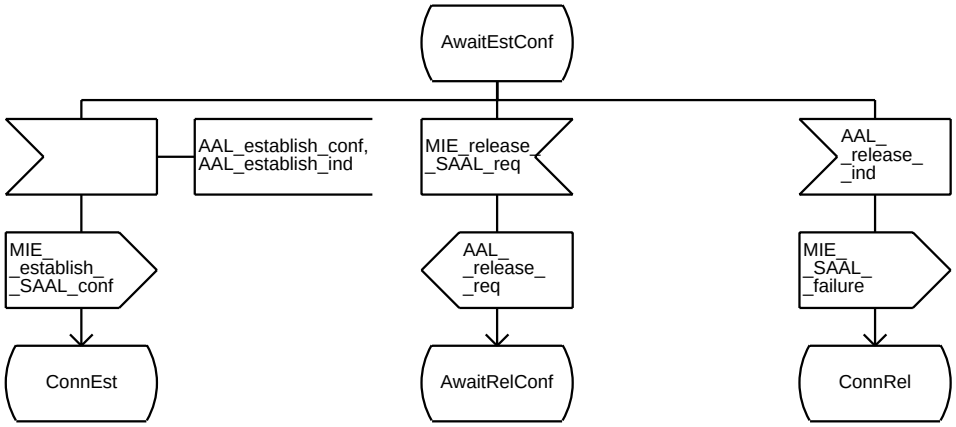

;
/*Process RESET_RSC controls the reset of resources*/
LocRem_ TransErr
A_IntTransList(V_IntTransId)!_ V_NoOfRTMCMsg := A_IntTransList(V_IntTransId)!_ V_NoOfRTMCMsg-1
A_IntTransList(V_IntTransId)!_ V_NoOfRTMCMsg
else
MDU_ _abort_req
To outstanding Transactions
0
A_RscList := A_IntTransList(V_IntTransId)!_ A_RscList
meeReset_ RscConf
(C_TransErr, A_RscList)
RemReset

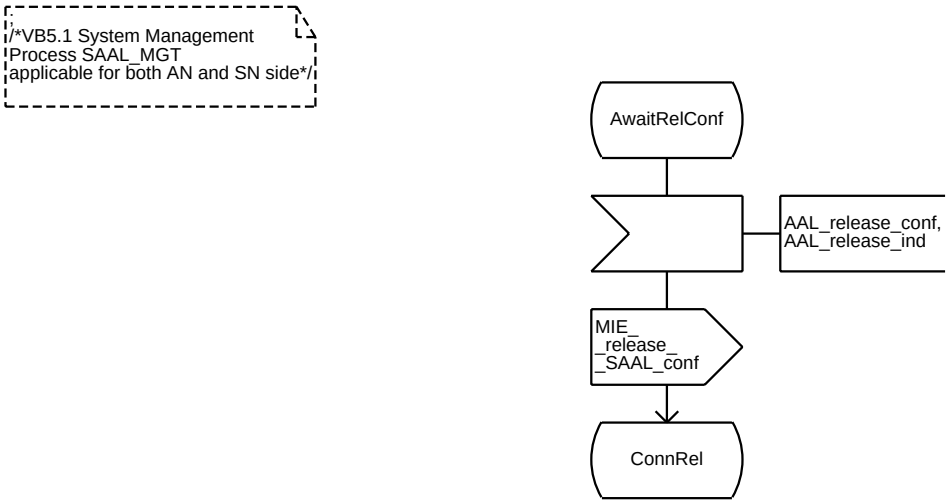

CTR
(SL_SAAL_CTR)
(SL_CTR_SAAL)
Process Type SAAL_MGT
1(5)
/*VB5.1 System Management
Process SAAL_MGT
applicable for both AN and SN side*/
/* State descriptions
ConnRel: initial state, no SAAL connection available.
AwaitEstConf: process is waiting for establish confirmation of the peer entity.
AwaitRelConf: process is waiting for release confirmation of the peer entity.
ConnEst: SAAL in operation. */
SAAL
(SL_SAAL_SYS)
(SL_SYS_SAAL)

Process Type SAAL_MGT 2(5)

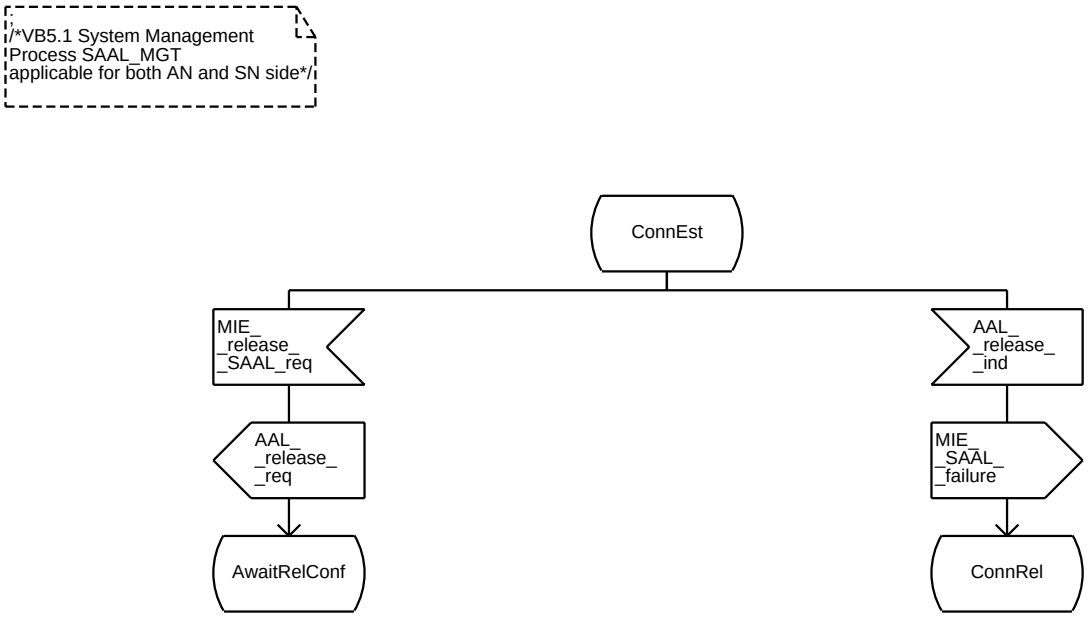


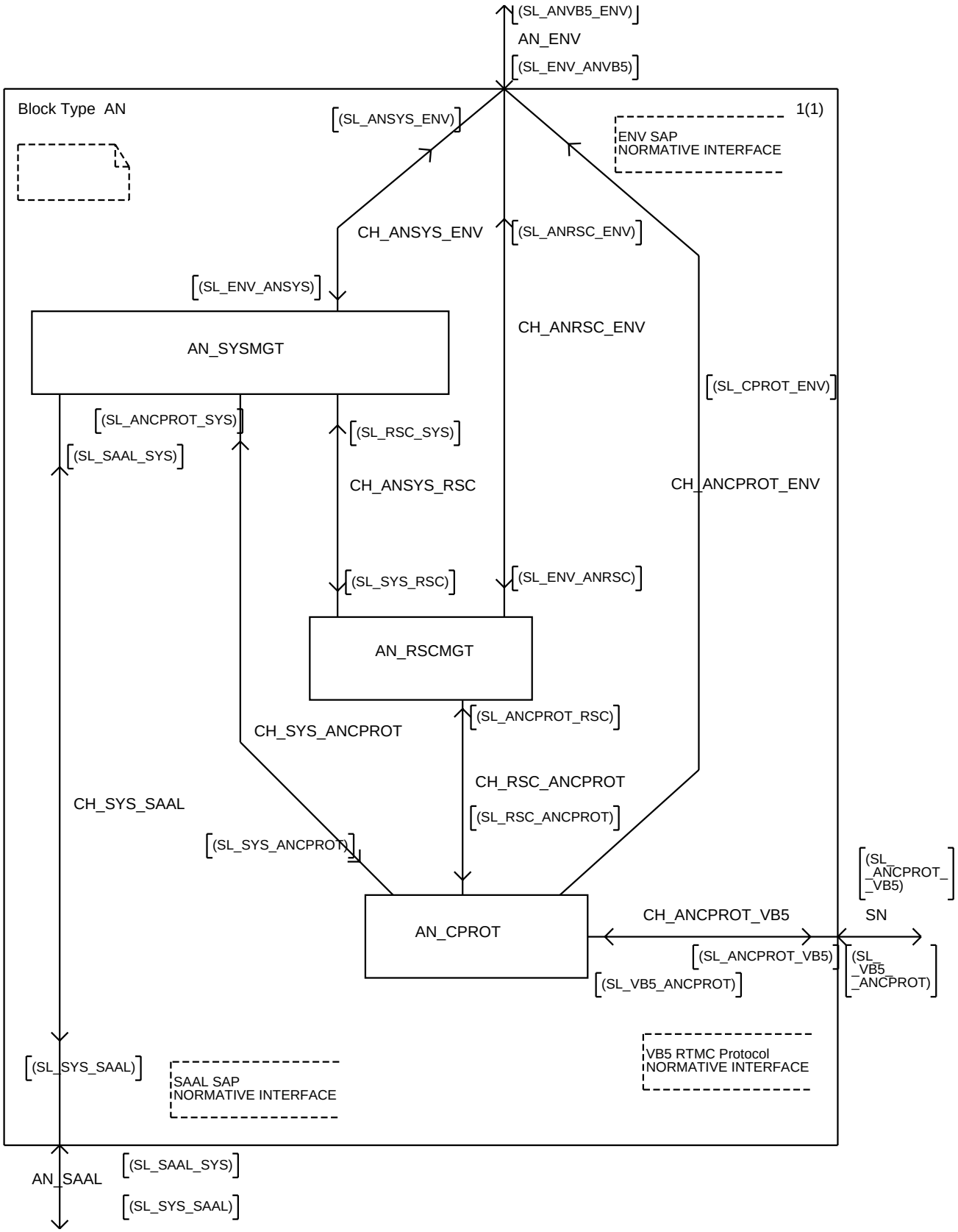
/*VB5.1 System Management
Process SAAL_MGT
applicable for both AN and SN side*/

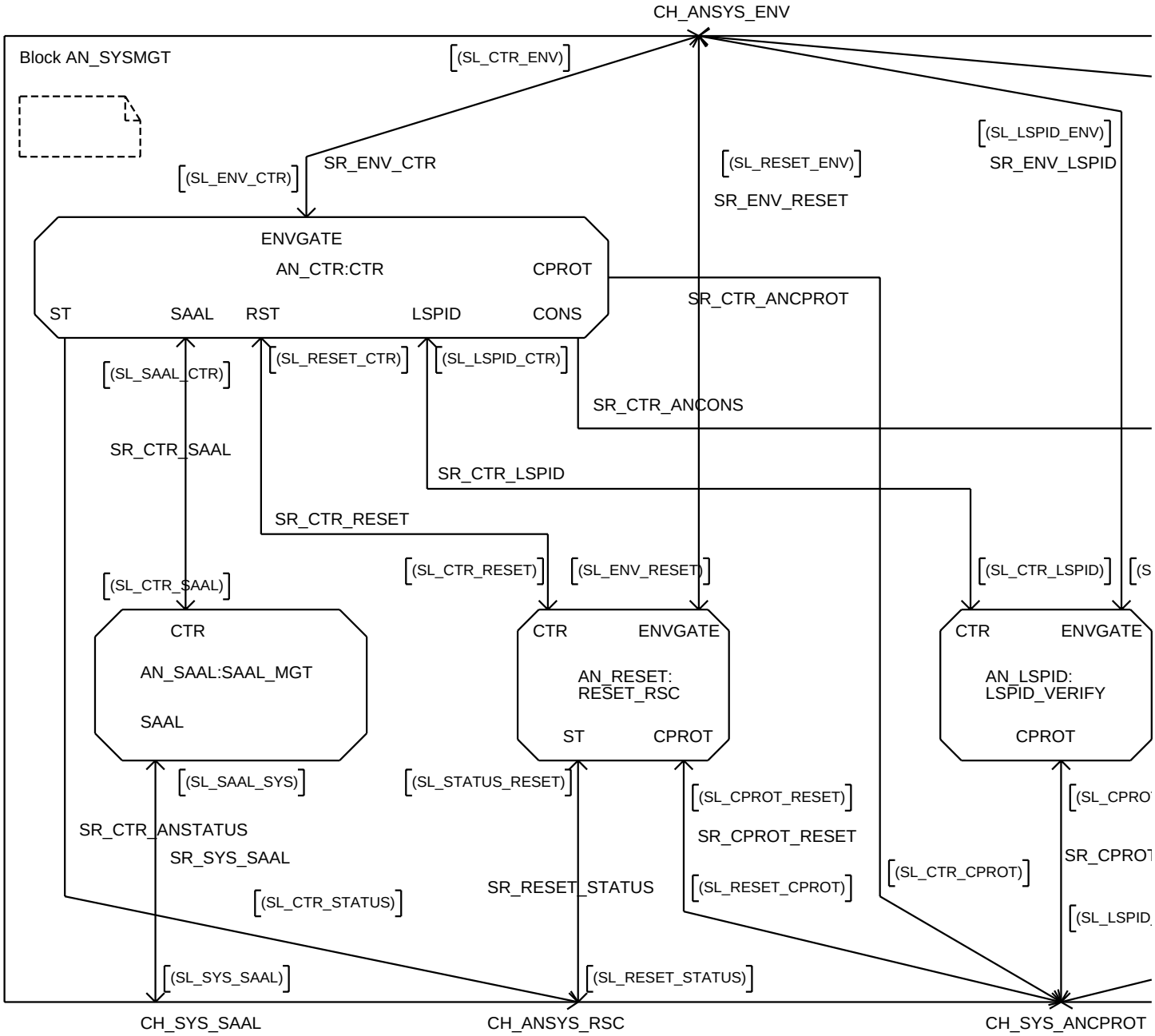


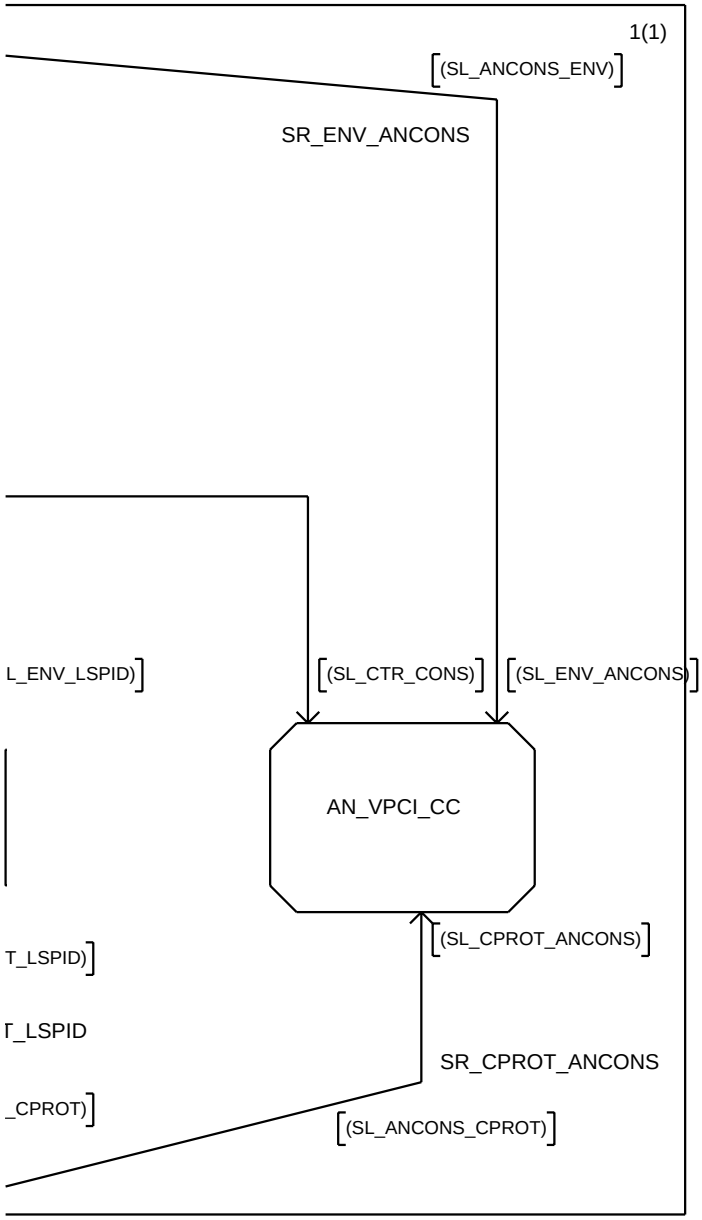

/*VB5.1 System Management
Process SAAL_MGT
applicable for both AN and SN side*/
AwaitRelConf
AAL_release_conf,
AAL_release_ind
MIE_
release
_SAAL_conf
ConnRel

Process Type SAAL_MGT 5(5)




(SL_ANVB5_ENV)
AN_ENV
(SL_ENV_ANVB5)
Block Type AN
1(1)
(SL_ANSYS_ENV)
ENV SAP
NORMATIVE INTERFACE
CH_ANSYS_ENV
(SL_ANRSC_ENV)
(SL_ENV_ANSYS)
CH_ANRSC_ENV
AN_SYSMGT
(SL_CPROT_ENV)
(SL_ANCPROT_SYS)
(SL_RSC_SYS)
(SL_SAAL_SYS)
CH_ANSYS_RSC
CH_ANCPROT_ENV
(SL_SYS_RSC)
(SL_ENV_ANRSC)
AN_RSCMGT
(SL_ANCPROT_RSC)
CH_SYS_ANCPROT
CH_RSC_ANCPROT
CH_SYS_SAAL
(SL_RSC_ANCPROT)
(SL_SYS_ANCPROT)
(SL_
ANCPROT
_VB5)
AN_CPROT
CH_ANCPROT_VB5
SN
(SL_ANCPROT_VB5)
(SL_
_VB5_
_ANCPROT)
(SL_VB5_ANCPROT)
(SL_SYS_SAAL)
SAAL SAP
NORMATIVE INTERFACE
VB5 RTMC Protocol
NORMATIVE INTERFACE
AN_SAAL
(SL_SAAL_SYS)
(SL_SYS_SAAL)


CH_ANSYS_ENV
Block AN_SYSMGT
(SL_CTR_ENV)
(SL_ENV_CTR)
SR_ENV_CTR
(SL_RESET_ENV)
SR_ENV_RESET
(SL_LSPID_ENV)
SR_ENV_LSPID
ENVGATE
AN_CTR:CTR
CPROT
ST
SAAL
RST
LSPID
CONS
SR_CTR_ANCPROT
(SL_SAAL_CTR)
(SL_RESET_CTR)
(SL_LSPID_CTR)
SR_CTR_ANCONS
SR_CTR_SAAL
SR_CTR_LSPID
SR_CTR_RESET
(SL_CTR_RESET)
(SL_ENV_RESET)
(SL_CTR_LSPID)
(SL_CTR_SAAL)
CTR
AN_SAAL:SAAL_MGT
SAAL
CTR
ENVGATE
AN_RESET:
RESET_RSC
ST
CPROT
CTR
ENVGATE
AN_LSPID:
LSPID_VERIFY
CPROT
(SL_SAAL_SYS)
(SL_STATUS_RESET)
(SL_CPROT_RESET)
SR_CTR_ANSTATUS
SR_SYS_SAAL
SR_CPROT_RESET
(SL_CTR_CPROT)
SR_RESET_STATUS
(SL_RESET_CPROT)
(SL_CTR_STATUS)
(SL_SYS_SAAL)
(SL_RESET_STATUS)
CH_SYS_SAAL
CH_ANSYS_RSC
CH_SYS_ANCPROT


1(1)
(SL_ANCONS_ENV)
SR_ENV_ANCONS
L_ENV_LSPID)
(SL_CTR_CONS)
(SL_ENV_ANCONS)
AN_VPCI_CC
(SL_CPROT_ANCONS)
T_LSPID)
T_LSPID
SR_CPROT_ANCONS
_CPROT)
(SL_ANCONS_CPROT)

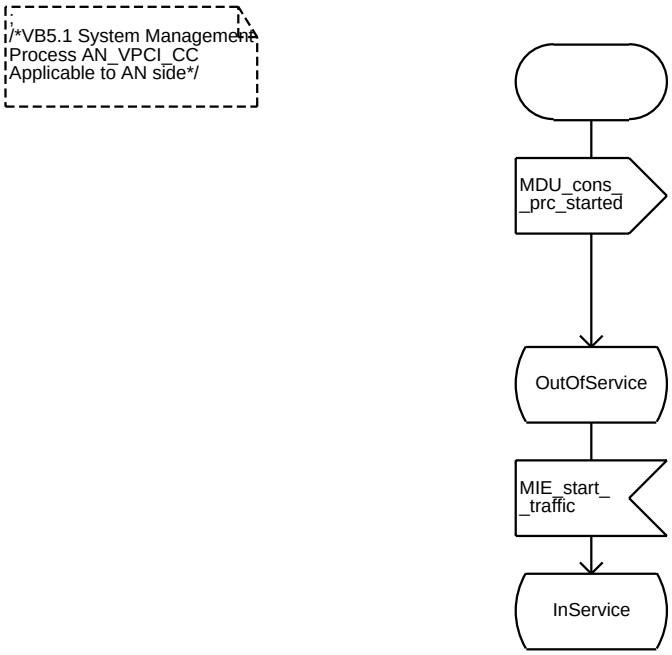
Process AN_VPCI_CC 1(5)

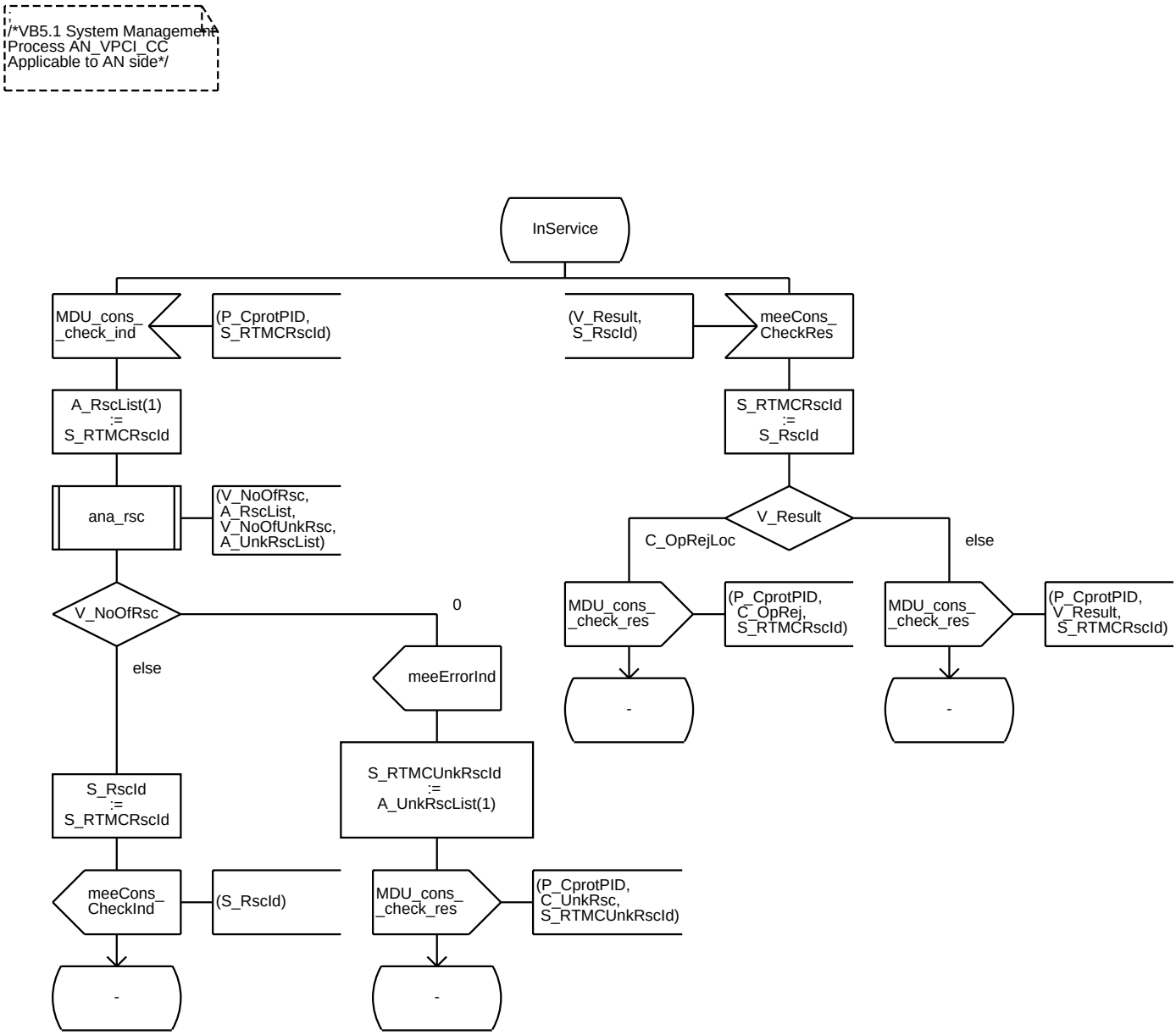
```
/*VB5.1 System Management
Process AN_VPCI_CC
Applicable to AN side*/
```

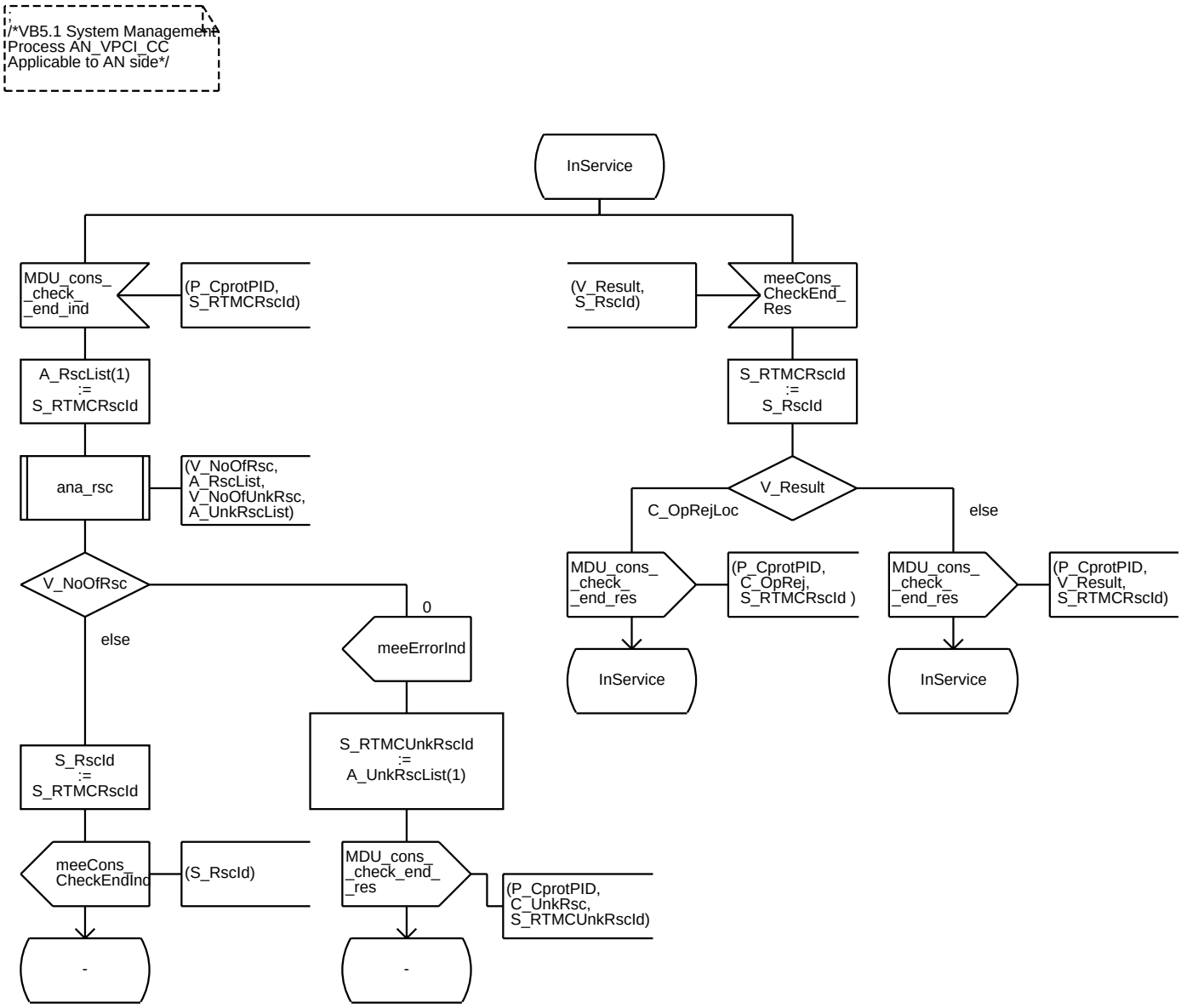
```
/* Definitions and declarations
for process AN_VPCI_CC */
```

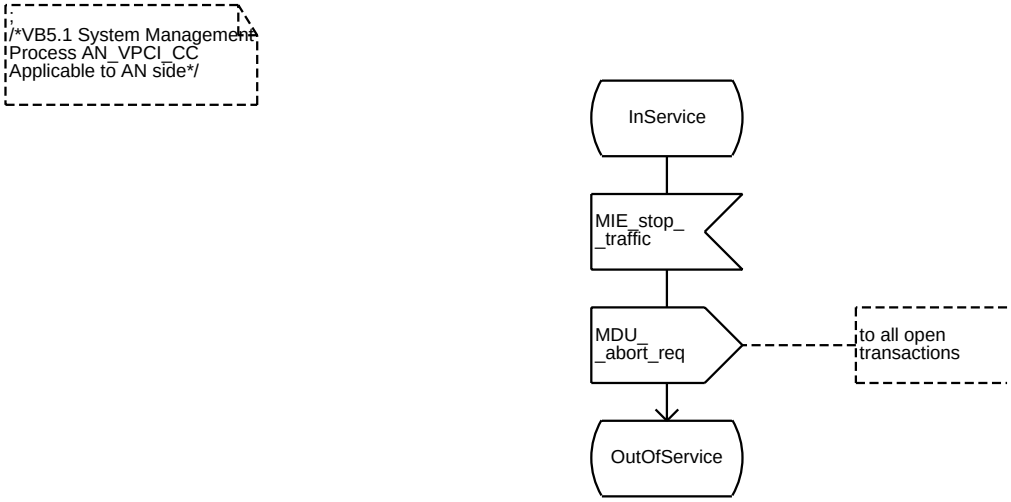
```
/* State descriptions
OutOfService: Initial state. Process is waiting for activation from CTR process.
InService:        Process started, handling all events
*/
```

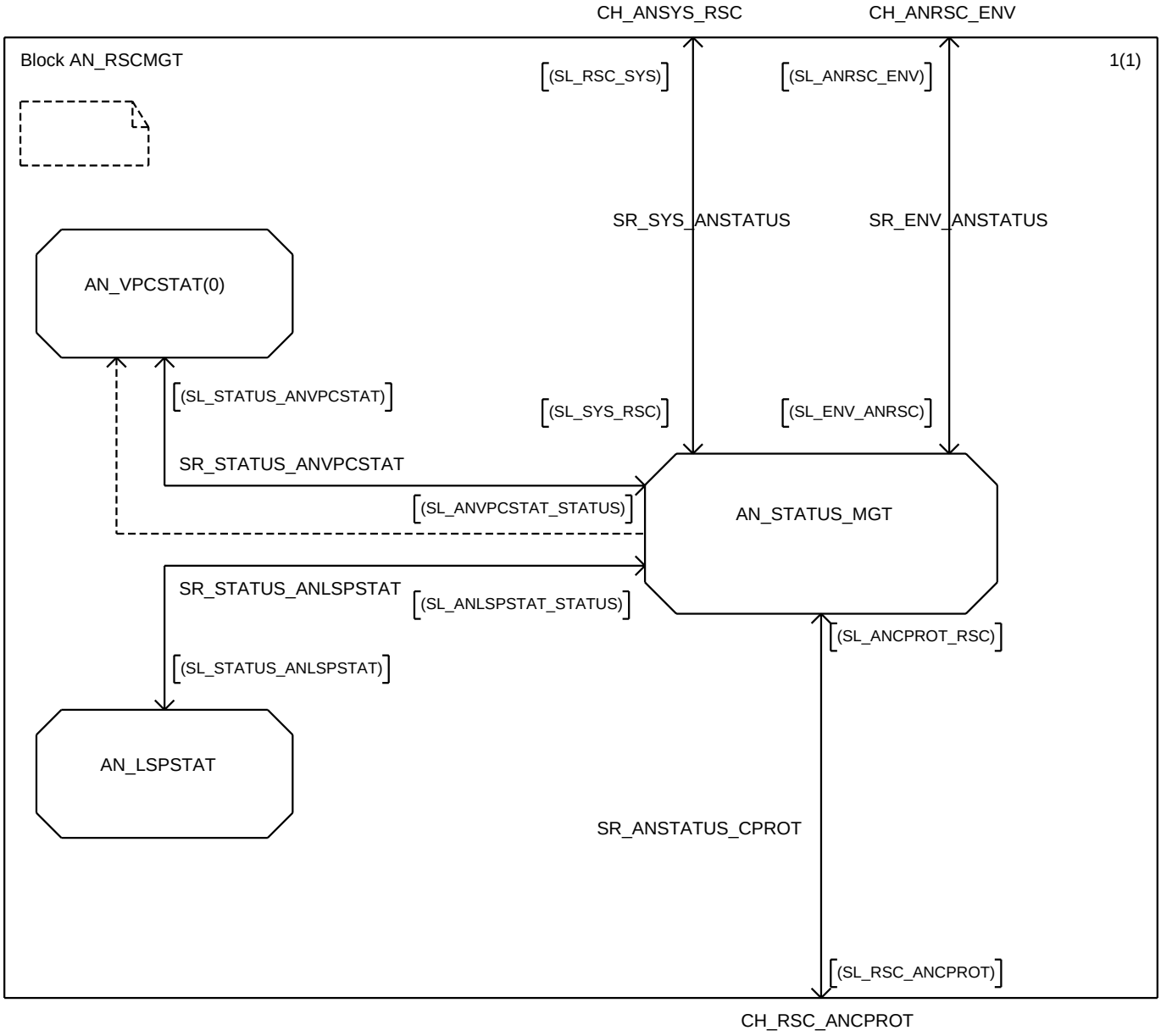
```
/*Signal data declarations*/
DCL
S_RscId            ST_RscId,
S_RTMCRscId ST_RscId; /*Resource Identifier Information Element*/
/**/
DCL
S_RTMCUnkRscId ST_RscId; /*Unknown Resource Identifier Information Element*/
/**/
DCL
A_RscList AT_RscList; /*Resource Identifier Information Element Table (needed for interface to ana_rsc procedure)*/
/**/
DCL
A_UnkRscList AT_RscList; /*Unknown Resource Identifier IE  Table (needed for interface to ana_rsc procedure)*/
/**/
DCL
V_Result IT_Result; /*Result*/
/**/
DCL
V_NoOfRsc INTEGER; /*Number of resources*/
/**/
DCL
V_NoOfUnkRsc INTEGER; /*Number of unknown resources*/
/**/
DCL
P_CprotPID PID; /*PID of CPROT instance responsible for transaction*/
```


/*VB5.1 System Management
Process AN_VPCI_CC
Applicable to AN side*/
MDU_cons_
_prc_started
OutOfService
MIE_start_
_traffic
InService


/*VB5.1 System Management
Process AN_VPCI_CC
Applicable to AN side*/
InService
MDU_cons_
_check_ind
(P_CprotPID,
S_RTMCRscId)
(V_Result,
S_RscId)
meeCons_
CheckRes
A_RscList(1)
:=
S_RTMCRscId
S_RTMCRscId
:=
S_RscId
ana_rsc
(V_NoOfRsc,
A_RscList,
V_NoOfUnkRsc,
A_UnkRscList)
V_Result
C_OpRejLoc
else
V_NoOfRsc
0
else
MDU_cons_
_check_res
(P_CprotPID,
C_OpRej,
S_RTMCRscId)
MDU_cons_
_check_res
(P_CprotPID,
V_Result,
S_RTMCRscId)
-
-
meeErrorInd
S_RscId
:=
S_RTMCRscId
S_RTMCUnkRscId
:=
A_UnkRscList(1)
meeCons_
CheckInd
(S_RscId)
MDU_cons_
_check_res
(P_CprotPID,
C_UnkRsc,
S_RTMCUnkRscId)
-
-


/*VB5.1 System Management
Process AN_VPCI_CC
Applicable to AN side*/
InService
MDU_cons_
check
_end_ind
(P_CprotPID,
S_RTMCRscId)
(V_Result,
S_RscId)
meeCons_
CheckEnd_
Res
A_RscList(1)
:=
S_RTMCRscId
S_RTMCRscId
:=
S_RscId
ana_rsc
(V_NoOfRsc,
A_RscList,
V_NoOfUnkRsc,
A_UnkRscList)
V_Result
C_OpRejLoc
else
MDU_cons_
check
_end_res
(P_CprotPID,
C_OpRej,
S_RTMCRscId)
MDU_cons_
check
_end_res
(P_CprotPID,
V_Result,
S_RTMCRscId)
V_NoOfRsc
0
else
meeErrorInd
InService
InService
S_RTMCUnkRscId
:=
A_UnkRscList(1)
S_RscId
:=
S_RTMCRscId
meeCons_
CheckEndInd
(S_RscId)
MDU_cons_
_check_end_
_res
(P_CprotPID,
C_UnkRsc,
S_RTMCUnkRscId)
-
-


/*VB5.1 System Management
Process AN_VPCI_CC
Applicable to AN side*/
InService
MIE_stop_
_traffic
MDU_
_abort_req
to all open
transactions
OutOfService


CH_ANSYS_RSC
CH_ANRSC_ENV
Block AN_RSCMGT
1(1)
(SL_RSC_SYS)
(SL_ANRSC_ENV)
SR_SYS_ANSTATUS
SR_ENV_ANSTATUS
AN_VPCSTAT(0)
(SL_STATUS_ANVPCSTAT)
(SL_SYS_RSC)
(SL_ENV_ANRSC)
SR_STATUS_ANVPCSTAT
(SL_ANVPCSTAT_STATUS)
AN_STATUS_MGT
SR_STATUS_ANLSPSTAT
(SL_ANLSPSTAT_STATUS)
(SL_ANCPROT_RSC)
(SL_STATUS_ANLSPSTAT)
AN_LSPSTAT
SR_ANSTATUS_CPROT
(SL_RSC_ANCPROT)
CH_RSC_ANCPROT

Process AN_STATUS_MGT 1(21)

```
;
/*AN_STATUS_MGT co-ordinates
local status changes of VPCs and LSP*/
```

```
/* Definintions and declarations for
AN_STATUS_MGT */
```

```
/* State descriptions

OutOfService:          Init State of AN_STATUS_MGT, process is waiting for startup of the interface
UblDuringStartup:      Interface startup was initiated and unblocking of all VPCFSMs and LSPFSM is running
AwaitStartTraffic:     Unblocking during startup completed, start traffic expected
InService:             Normal operation, no status change was requested
AwaitResetFSMAck:      Reset is running
AwaitUblFSMAck:        Unblocking of single FSMs is running
AwaitBlFSMAck:         Blocking of single FSMs is running
AwaitAclFSMAck:        Shutdown of single FSMs is running
*/
```

```
/*AN_STATUS_MGT internal data type definitions*/
/**/
/*Typedefinition for storing the requested action*/
SYNTYPE IT_Action = INTEGER
ENDSYNTYPE IT_Action;
/*Possible Values*/
SYNONYM C_Bl      IT_Action = 1;
SYNONYM C_Ubl     IT_Action = 2;
SYNONYM C_Acl     IT_Action = 3;
/**/
/*Typedefinition for temporary storage of data given by VPCSTAT/LSPSTAT processes*/
NEWTYPE ST_TempRscList STRUCT
   A_RTMCRscList     AT_RscList;
   A_RTMCBlRscList  AT_RTMCBlRscList;
ENDNEWTYPE ST_TempRscList;
/**/
/*Typedefinition for storing AN_VPCSTAT process addresses*/
NEWTYPE ST_VPCPID STRUCT
    S_RscId       ST_RscId;
    P_VPCPID  PID;
ENDNEWTYPE ST_VPCPID;
NEWTYPE AT_VPCPIDList ARRAY
    (INTEGER, ST_VPCPID)
ENDNEWTYPE AT_VPCPIDList;
```

```
;
/*AN_STATUS_MGT co-ordinates
local status changes of VPCs and LSP*/
```

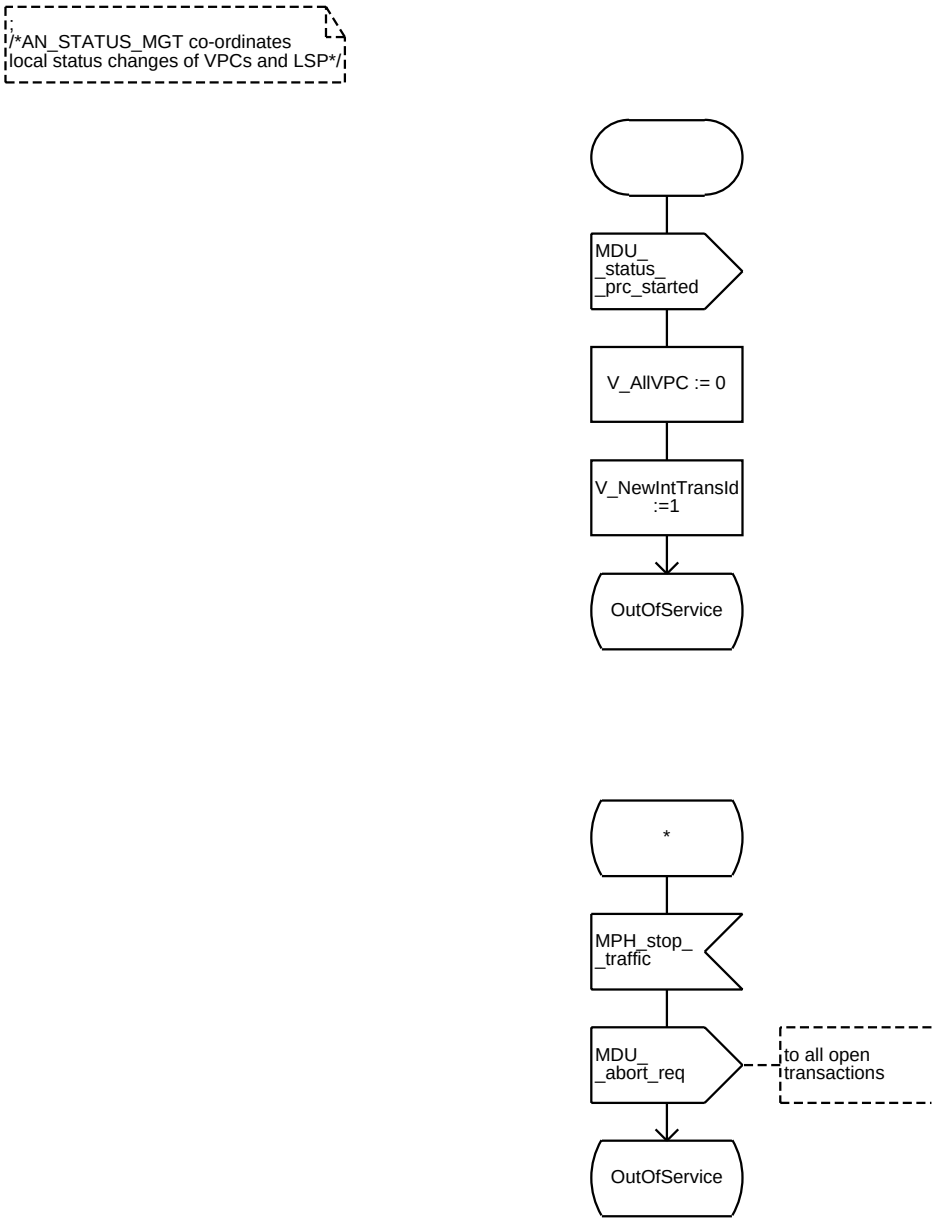
```
/* Definintions and declarations for
AN_STATUS_MGT */
```

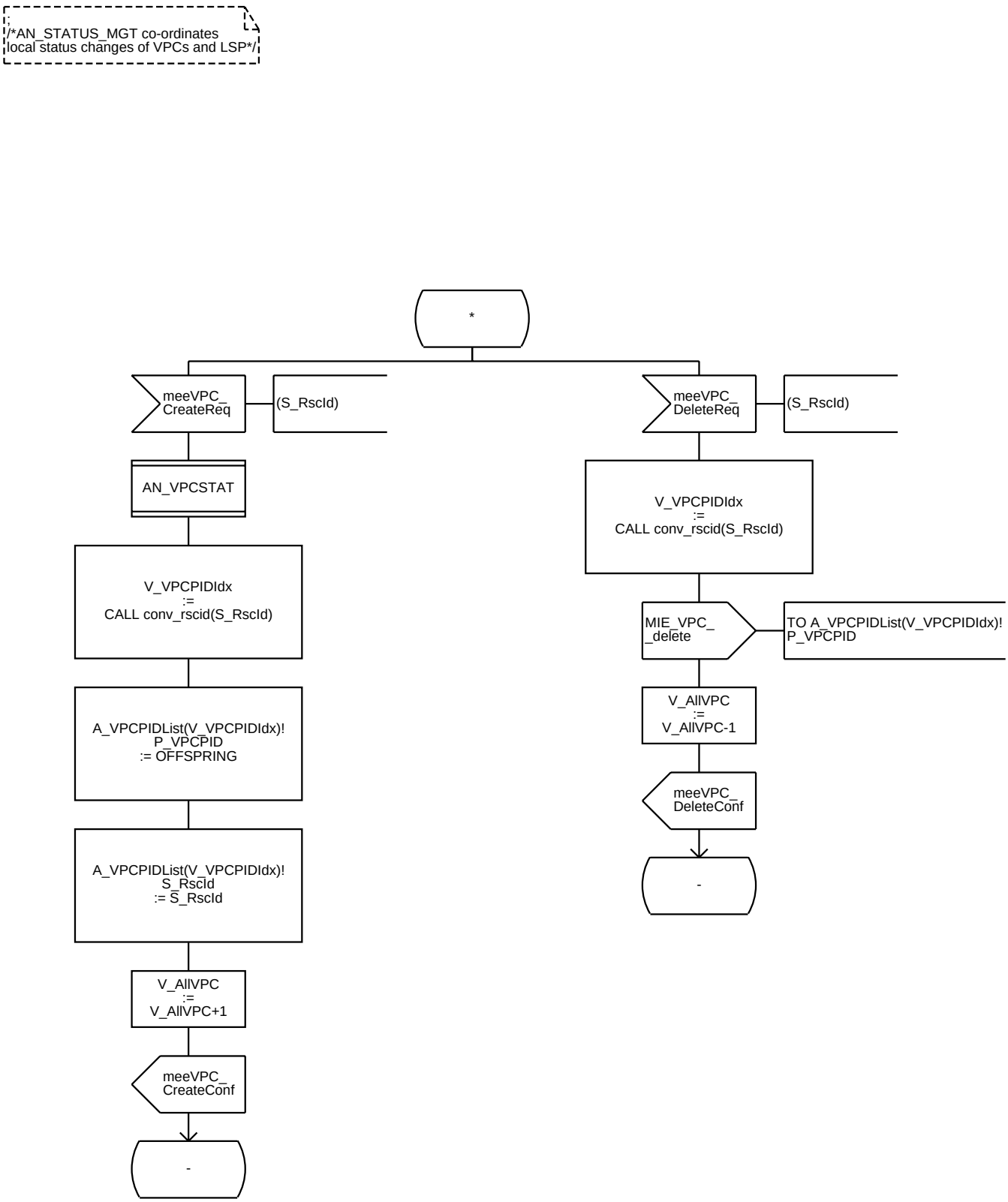
```
/*AN_STATUS_MGT internal variables and constants */
/**/
/*Variables for co-ordinating "unblock_rsc" procedure (e.g. during interface startup or reset of LSP)*/
DCL
B_VPCUblDone,    /*MIE_VPC_unblock_conf received from all requested VPC FSMs*/
B_LSPUblDone     /*MIE_LSP_unblock_conf received*/
BOOLEAN;
/**/
/*Variables for controlling VPCSTAT/LSPSTAT responses*/
DCL
V_TotalConf,    /*number of expected VPC state change confirmations*/
V_RecConf       /*number of received VPC state change confirmations*/
INTEGER;
/**/
/*Variable for temporary storage of data given by VPCSTAT/LSPSTAT processes*/
DCL S_TempRscList ST_TempRscList;
/**/
/*Variable for storage of transaction Id data*/
DCL A_IntTransList AT_IntTransList;
/**/
/*Variable for storing AN_VPCSTAT PIDs*/
DCL A_VPCPIDList AT_VPCPIDList;
/**/
/*Variable for indexing PID Table*/
DCL V_VPCPIDIdx INTEGER;
/**/
/*Variable for indexing IntTransIdList*/
DCL V_NewIntTransId IT_IntTransId;
/**/
/*Total number of provisioned VPCs*/
DCL V_AllVPC INTEGER;
/**/
/*Number of messages per transaction*/
DCL V_NoOfMsg INTEGER;
/**/
/*Requested action*/
DCL V_Action IT_Action;
```

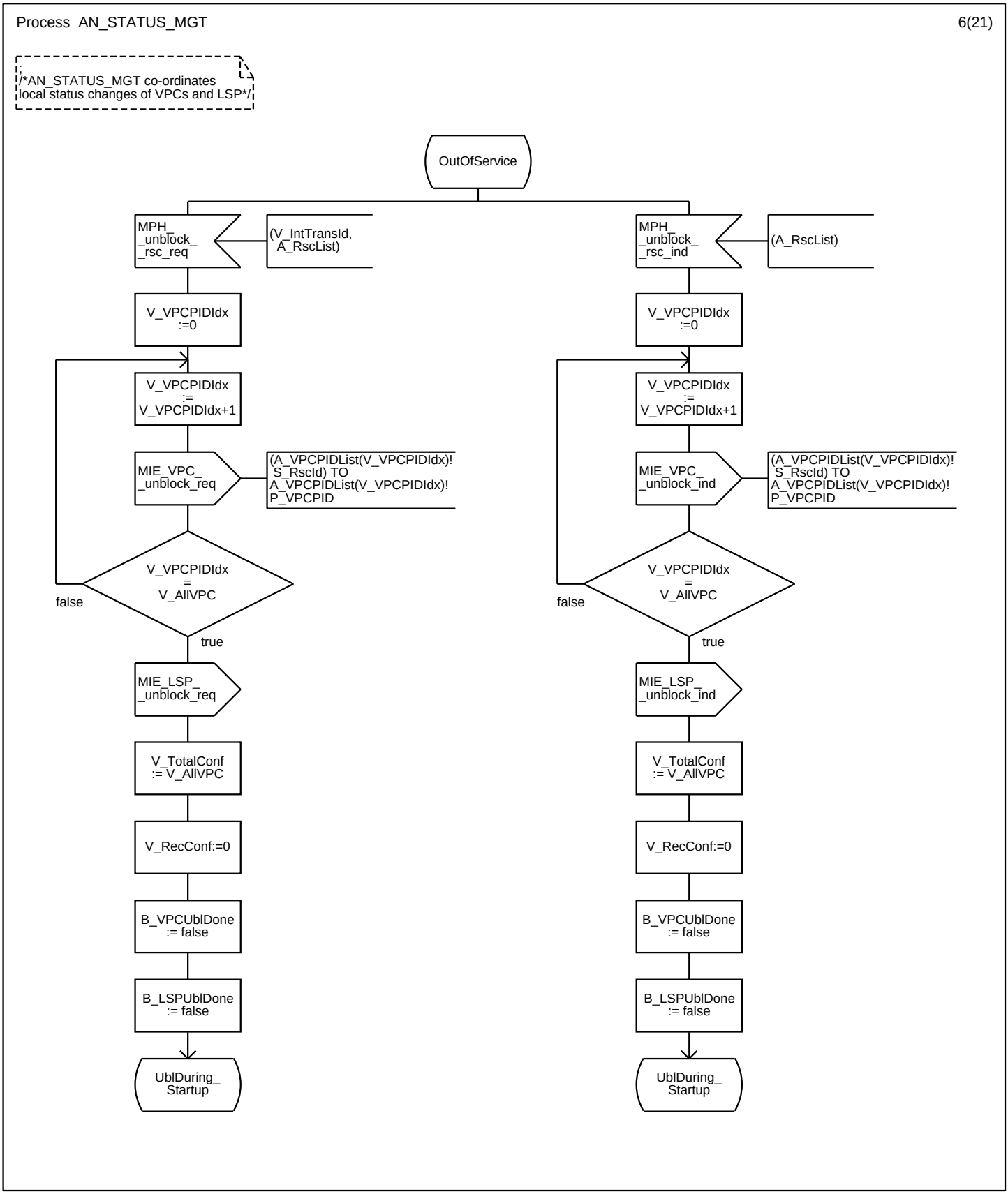
```
;
/*AN_STATUS_MGT co-ordinates
local status changes of VPCs and LSP*/
```

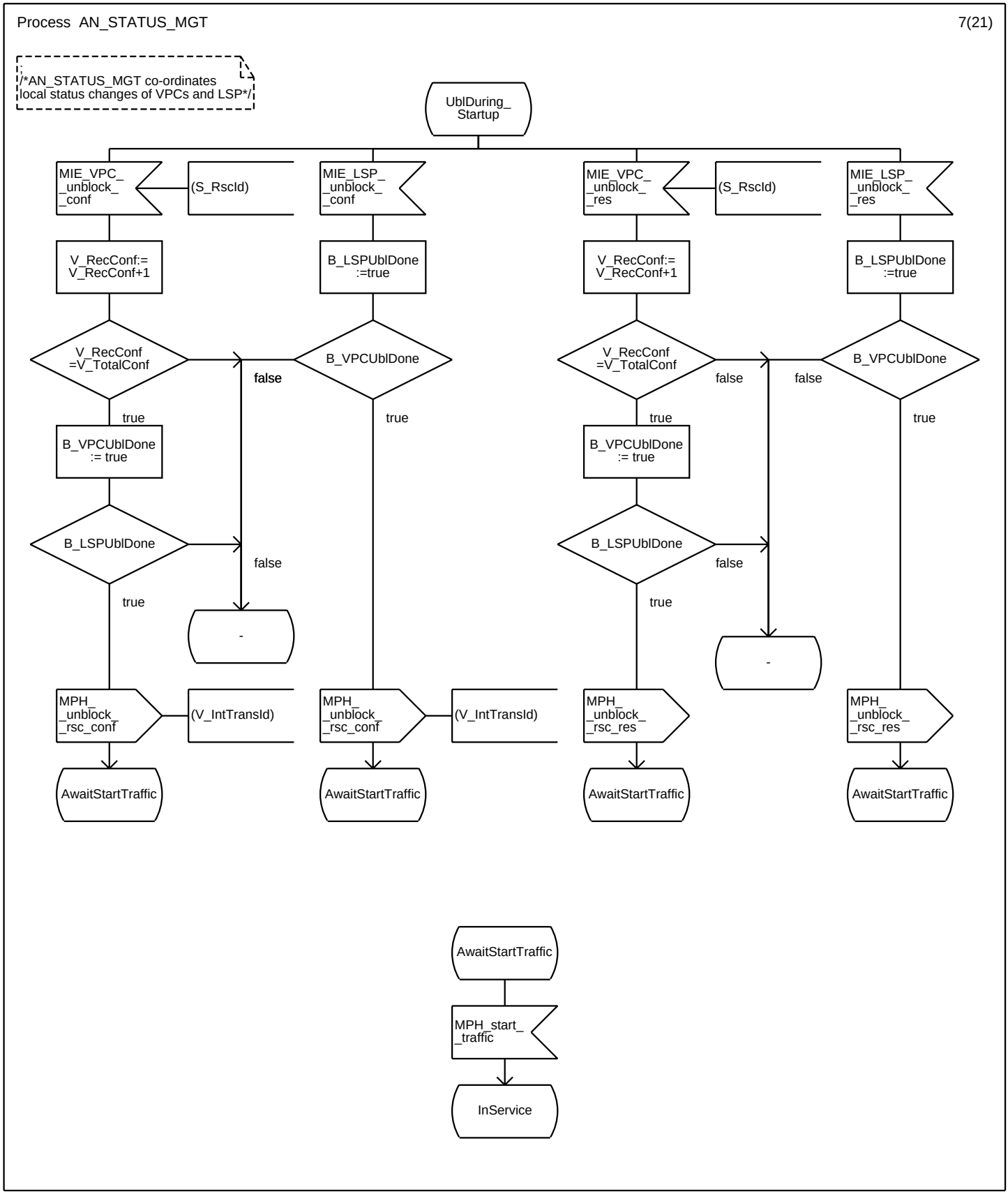
```
/* Definintions and declarations for
AN_STATUS_MGT */
```

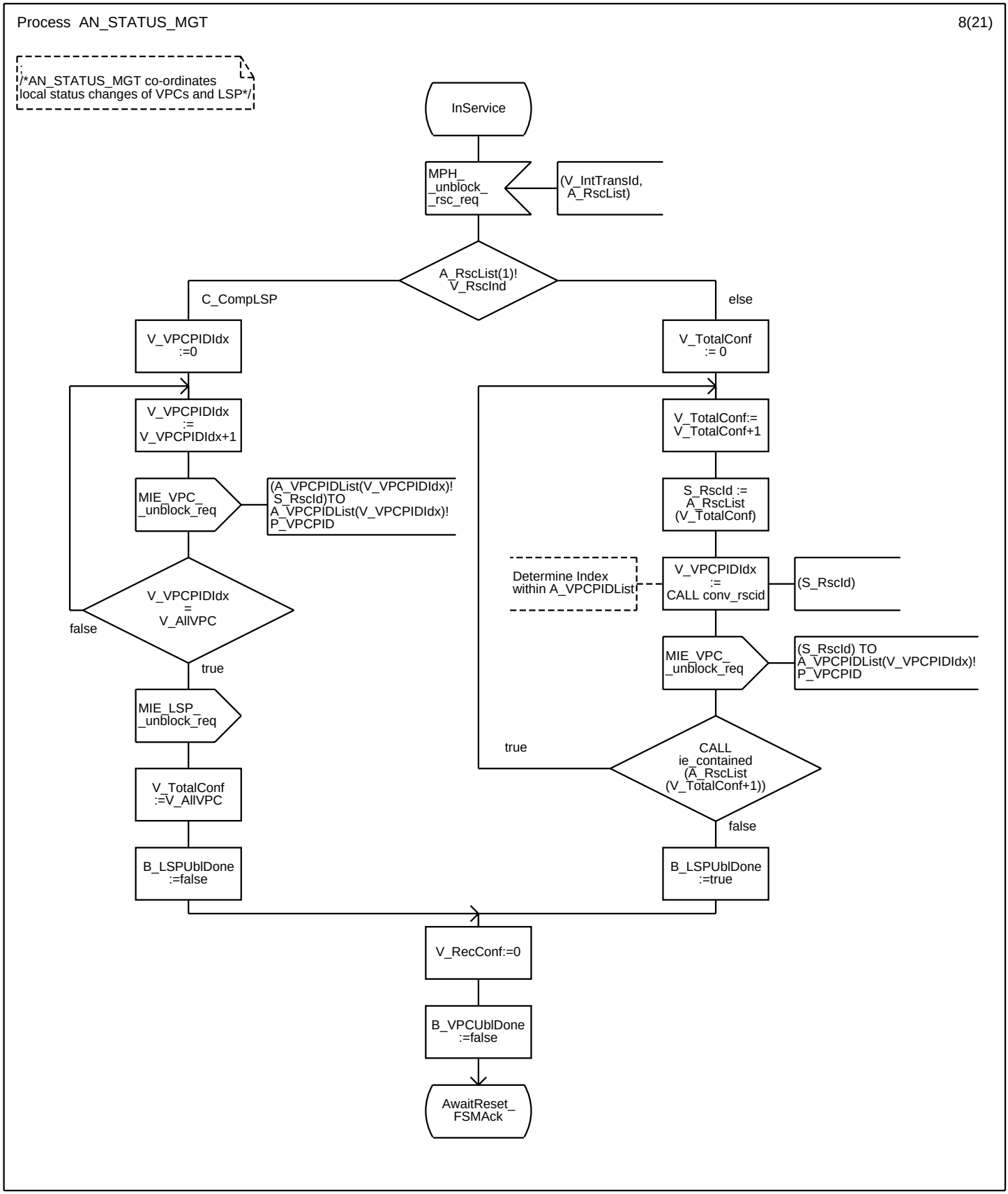
```
/*Signal data declarations*/
/**/
/*Resource identifier, used for identifying VPCSTAT/LSPSTAT process*/
DCL S_RscId ST_RscId;
/**/
/*List of Resource identifiers*/
DCL A_RTMCRscList AT_RscList ;
/**/
/*Status change reason given by the environment*/
DCL V_Reason IT_Reason;
/**/
/*Status change reason given from the FSMs*/
DCL S_RTMCReason ST_RTMCReason;
/**/
/*List of resources received from the environment*/
DCL A_RscList AT_RscList;
/**/
/*List of blocked resources received from the environment*/
DCL A_BlRscList AT_BlRscList;
/**/
/*Transaction identifier for communication with AN_CPROT process*/
DCL V_IntTransId IT_IntTransId;
/**/
/*List of resources unknown by the peer side, reported by AN_CPROT*/
DCL A_RTMCUnkRscList AT_RscList;
/**/
/*List of resources unknown by the peer side, reported to the environment*/
DCL A_UnkRscList AT_RscList;
/**/
/*Result of Operations*/
DCL V_Result      IT_Result;
```

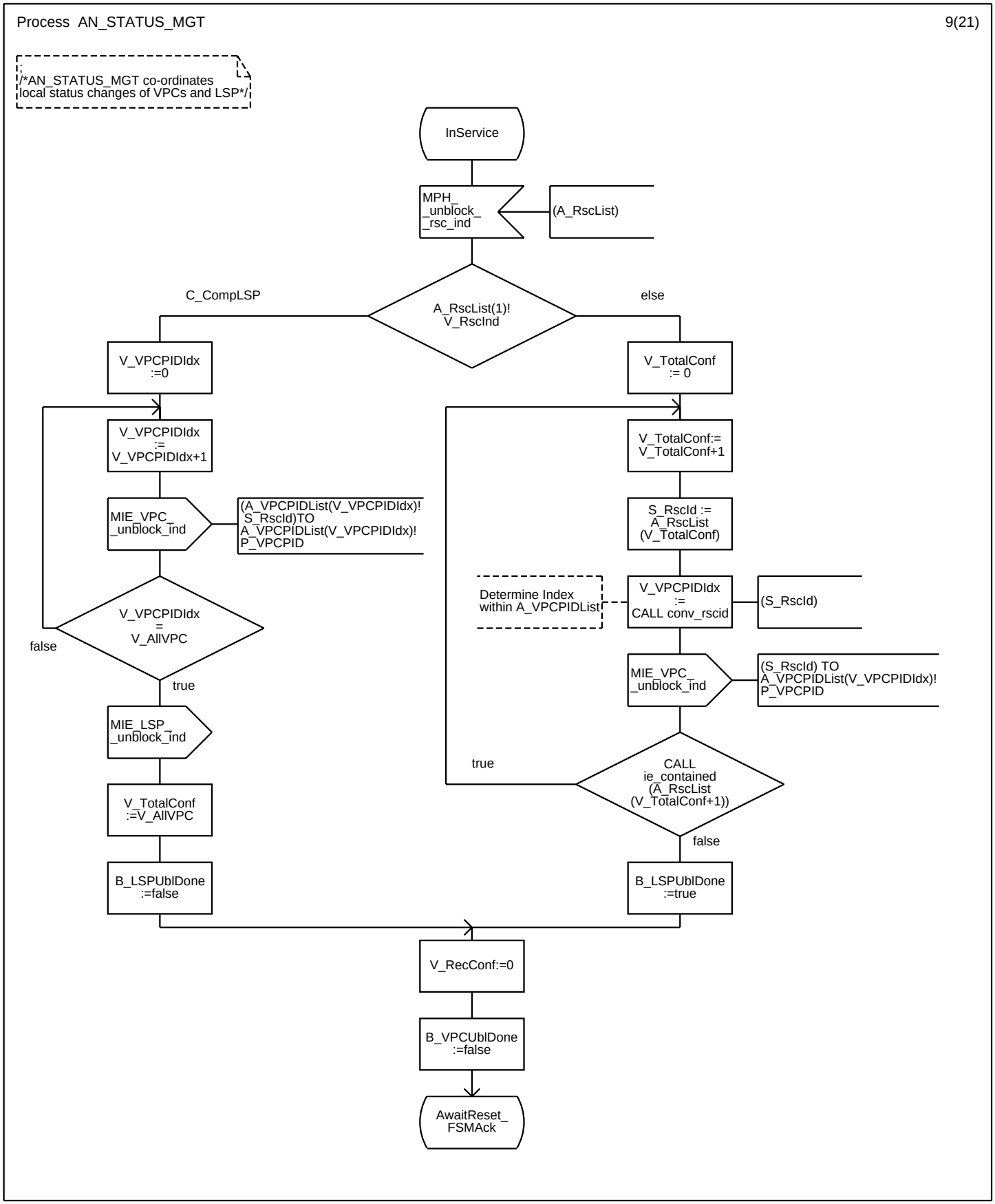

;
/*AN_STATUS_MGT co-ordinates
local status changes of VPCs and LSP*/
MDU_
status
_prc_started
V_AllVPC := 0
V_NewIntTransId
:=1
OutOfService
*
MPH_stop_
_traffic
MDU_
_abort_req
to all open
transactions
OutOfService

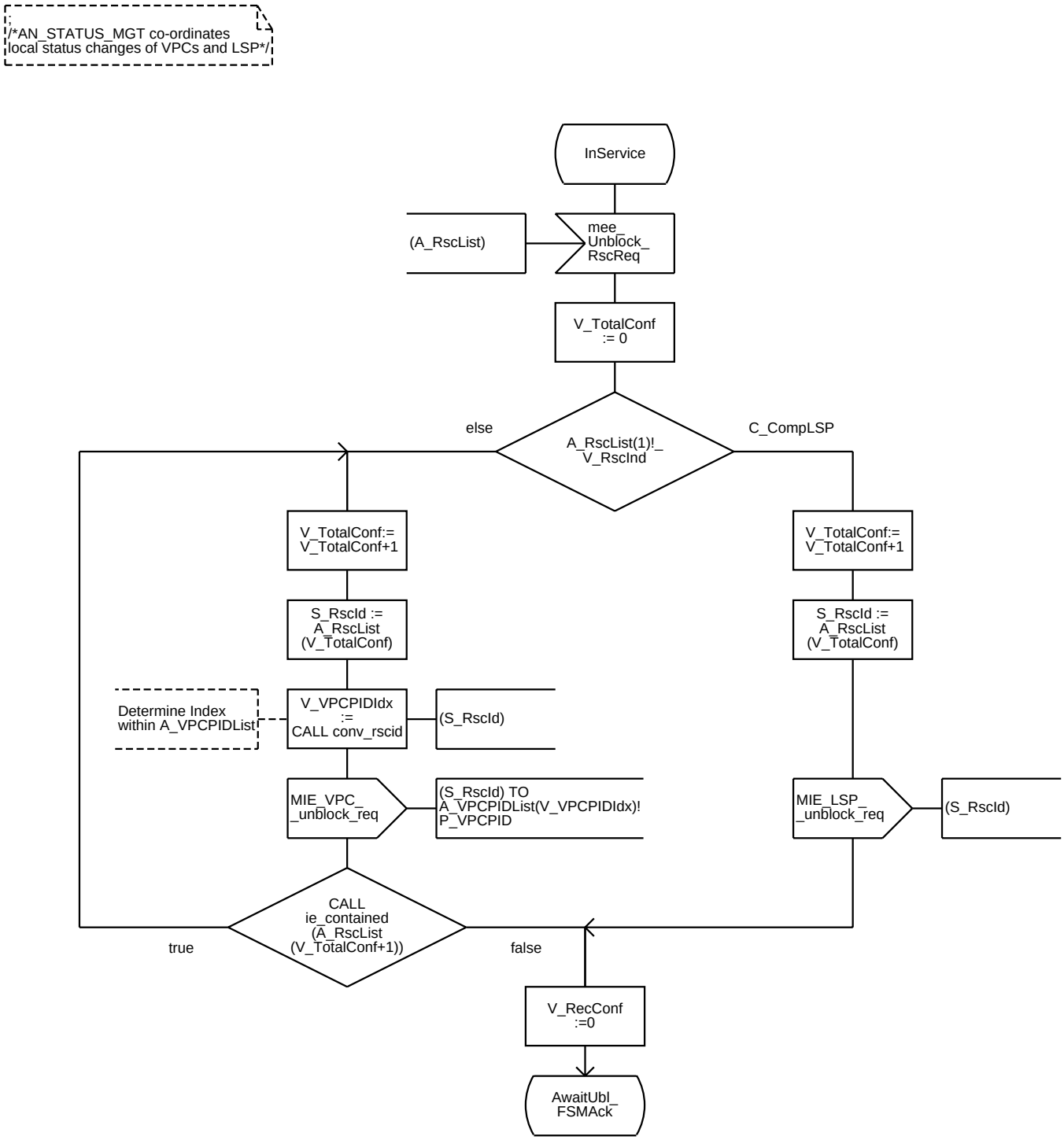

;
/*AN_STATUS_MGT co-ordinates
local status changes of VPCs and LSP*/
*
meeVPC_
CreateReq
(S_RscId)
meeVPC_
DeleteReq
(S_RscId)
AN_VPCSTAT
V_VPCPIDIdx
:=
CALL conv_rscid(S_RscId)
V_VPCPIDIdx
:=
CALL conv_rscid(S_RscId)
MIE_VPC_
_delete
TO A_VPCPIDList(V_VPCPIDIdx)!
P_VPCPID
A_VPCPIDList(V_VPCPIDIdx)!
P_VPCPID
:= OFFSPRING
V_AllVPC
:=
V_AllVPC-1
meeVPC_
DeleteConf
-
A_VPCPIDList(V_VPCPIDIdx)!
S_RscId
:= S_RscId
V_AllVPC
:=
V_AllVPC+1
meeVPC_
CreateConf
-

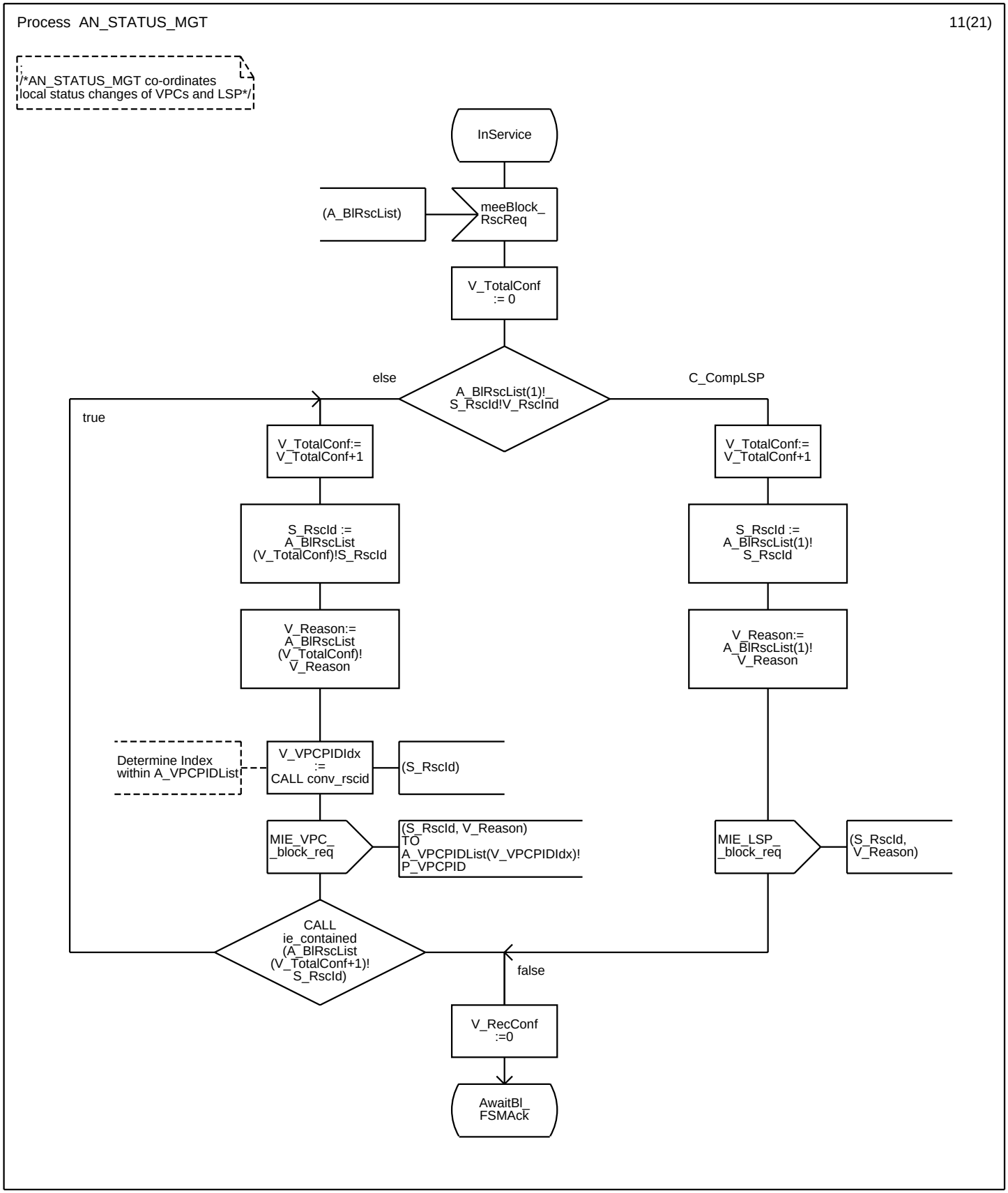

Process AN_STATUS_MGT
6(21)
/*AN_STATUS_MGT co-ordinates local status changes of VPCs and LSP*/
OutOfService
MPH_ _unblock_ _rsc_req
(V_IntTransId, A_RscList)
V_VPCPIDIdx :=0
V_VPCPIDIdx := V_VPCPIDIdx+1
MIE_VPC_ _unblock_req
(A_VPCPIDList(V_VPCPIDIdx)! S_RscId) TO A_VPCPIDList(V_VPCPIDIdx)! P_VPCPID
V_VPCPIDIdx = V_AllVPC
false
true
MIE_LSP_ _unblock_req
V_TotalConf := V_AllVPC
V_RecConf:=0
B_VPCUblDone := false
B_LSPUblDone := false
UblDuring_ Startup
MPH_ _unblock_ _rsc_ind
(A_RscList)
V_VPCPIDIdx :=0
V_VPCPIDIdx := V_VPCPIDIdx+1
MIE_VPC_ _unblock_ind
(A_VPCPIDList(V_VPCPIDIdx)! S_RscId) TO A_VPCPIDList(V_VPCPIDIdx)! P_VPCPID
V_VPCPIDIdx = V_AllVPC
false
true
MIE_LSP_ _unblock_ind
V_TotalConf := V_AllVPC
V_RecConf:=0
B_VPCUblDone := false
B_LSPUblDone := false
UblDuring_ Startup

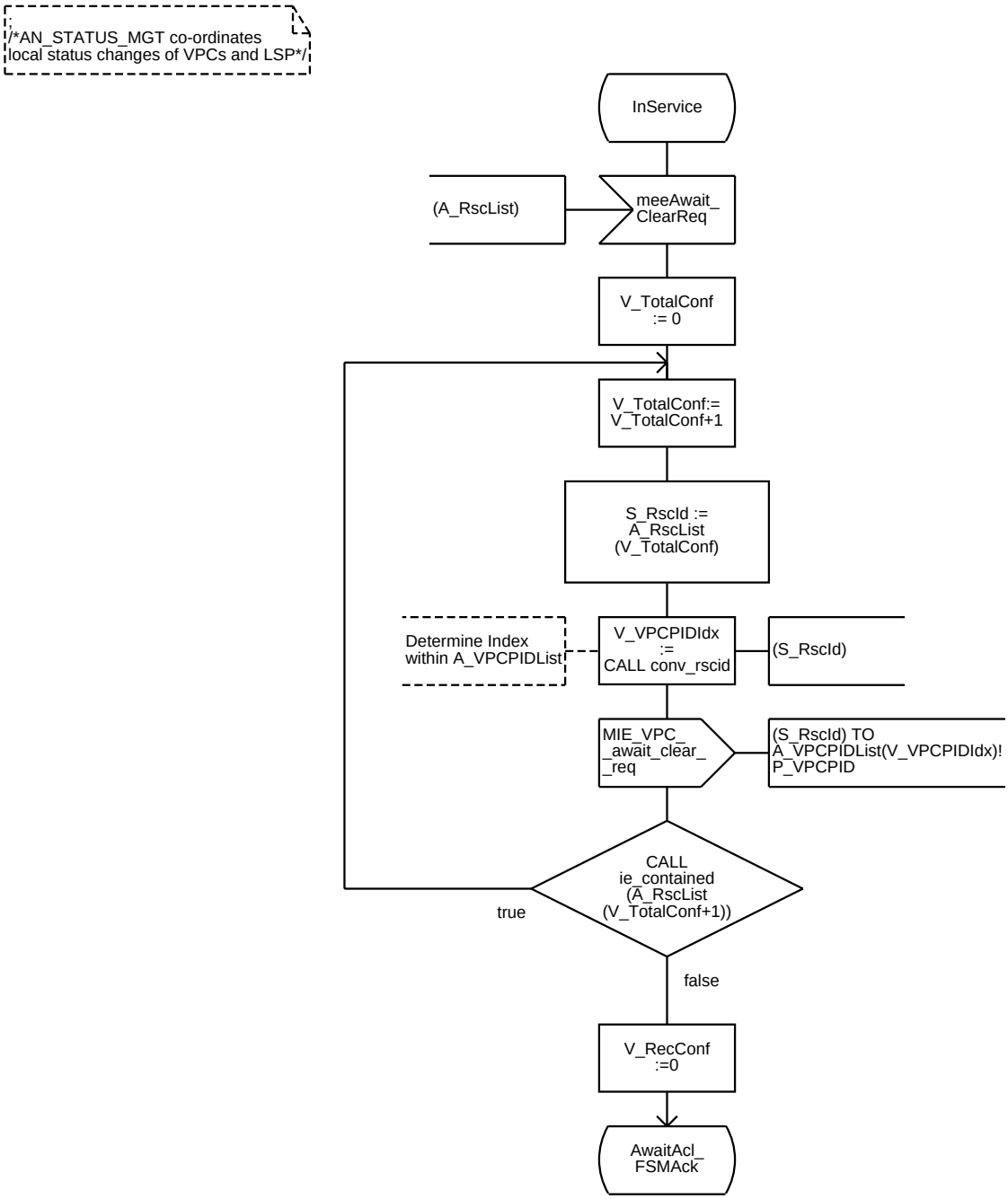

Process AN_STATUS_MGT
7(21)
/*AN_STATUS_MGT co-ordinates local status changes of VPCs and LSP*/
UblDuring_ Startup
MIE_VPC_ _unblock_ _conf
(S_RscId)
MIE_LSP_ _unblock_ _conf
MIE_VPC_ _unblock_ _res
(S_RscId)
MIE_LSP_ _unblock_ _res
V_RecConf:= V_RecConf+1
B_LSPUblDone :=true
V_RecConf:= V_RecConf+1
B_LSPUblDone :=true
V_RecConf =V_TotalConf
false
B_VPCUblDone
true
V_RecConf =V_TotalConf
false
false
B_VPCUblDone
true
true
true
B_VPCUblDone := true
B_VPCUblDone := true
B_LSPUblDone
false
true
B_LSPUblDone
false
true
-
-
MPH_ _unblock_ _rsc_conf
(V_IntTransId)
MPH_ _unblock_ _rsc_conf
(V_IntTransId)
MPH_ _unblock_ _rsc_res
MPH_ _unblock_ _rsc_res
AwaitStartTraffic
AwaitStartTraffic
AwaitStartTraffic
AwaitStartTraffic
AwaitStartTraffic
MPH_start_ _traffic
InService

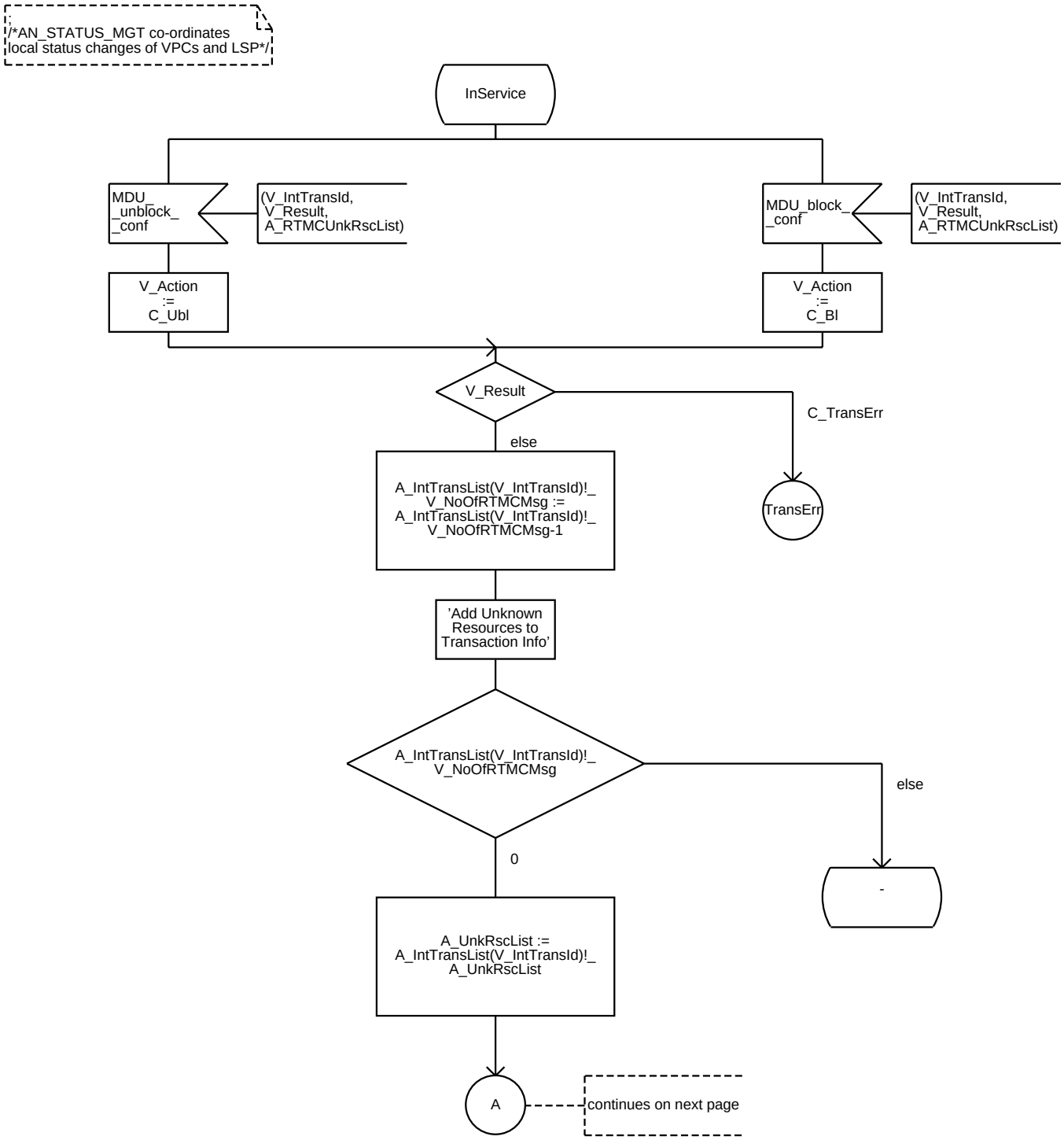

Process AN_STATUS_MGT
8(21)
;
/*AN_STATUS_MGT co-ordinates
local status changes of VPCs and LSP*/
InService
MPH_
unblock
_rsc_req
(V_IntTransId,
A_RscList)
A_RscList(1)!
V_RscInd
C_CompLSP
else
V_VPCPIDIdx
:=0
V_VPCPIDIdx
:=
V_VPCPIDIdx+1
MIE_VPC_
_unblock_req
(A_VPCPIDList(V_VPCPIDIdx)!
S_RscId)TO
A_VPCPIDList(V_VPCPIDIdx)!
P_VPCPID
V_VPCPIDIdx
=
V_AllVPC
false
true
MIE_LSP_
_unblock_req
V_TotalConf
:=V_AllVPC
B_LSPUblDone
:=false
V_TotalConf
:= 0
V_TotalConf:=
V_TotalConf+1
S_RscId :=
A_RscList
(V_TotalConf)
Determine Index
within A_VPCPIDList
V_VPCPIDIdx
:=
CALL conv_rscid
(S_RscId)
MIE_VPC_
_unblock_req
(S_RscId) TO
A_VPCPIDList(V_VPCPIDIdx)!
P_VPCPID
true
CALL
ie_contained
(A_RscList
(V_TotalConf+1))
false
B_LSPUblDone
:=true
V_RecConf:=0
B_VPCUblDone
:=false
AwaitReset_
FSMAck

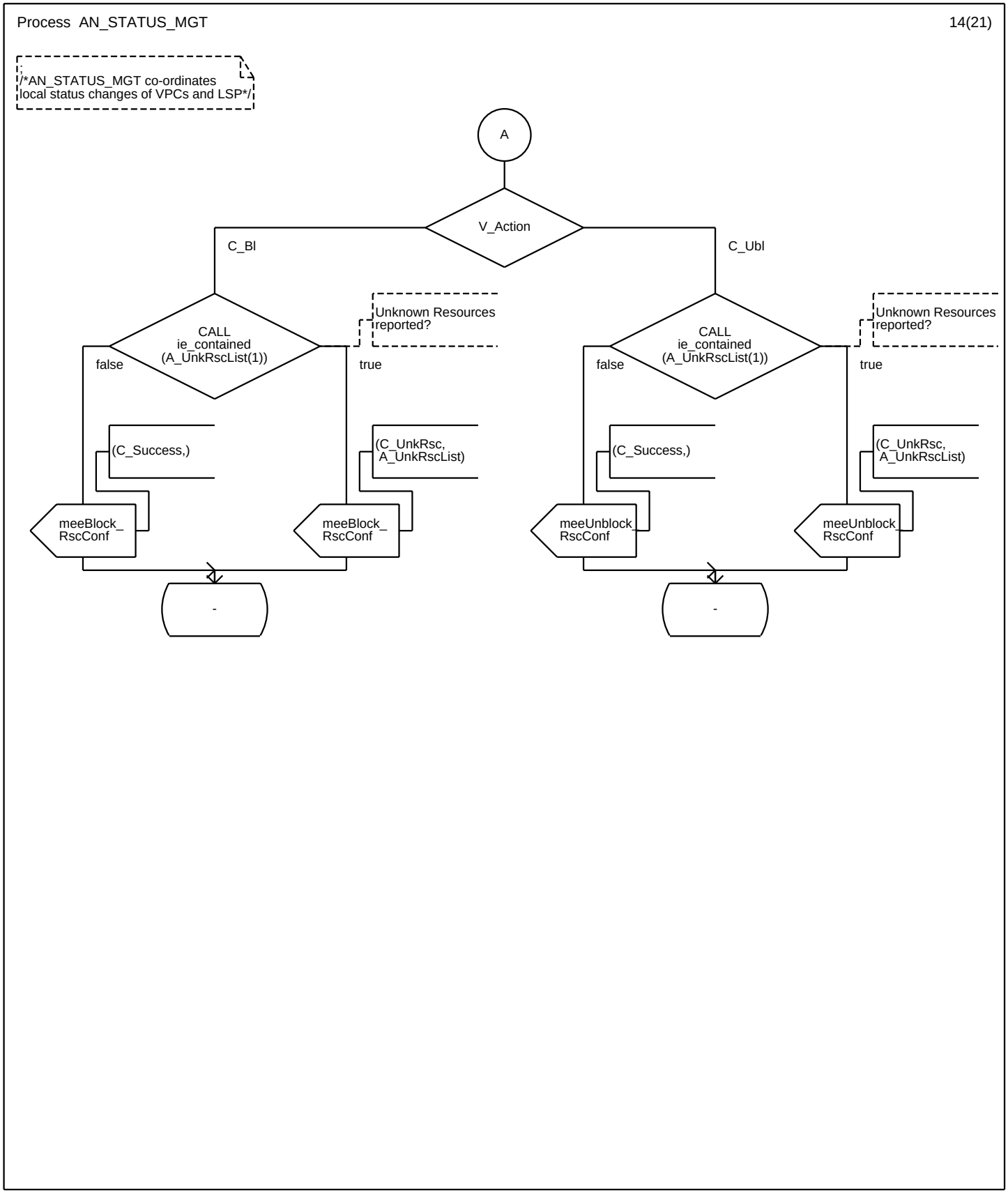

Process AN_STATUS_MGT
9(21)
;
/*AN_STATUS_MGT co-ordinates
local status changes of VPCs and LSP*/
InService
MPH_
unblock
_rsc_ind
(A_RscList)
A_RscList(1)!
V_RscInd
C_CompLSP
else
V_VPCPIDIdx
:=0
V_VPCPIDIdx
:=
V_VPCPIDIdx+1
MIE_VPC_
_unblock_ind
(A_VPCPIDList(V_VPCPIDIdx)!
S_RscId)TO
A_VPCPIDList(V_VPCPIDIdx)!
P_VPCPID
V_VPCPIDIdx
=
V_AllVPC
false
true
MIE_LSP_
_unblock_ind
V_TotalConf
:=V_AllVPC
B_LSPUblDone
:=false
V_TotalConf
:= 0
V_TotalConf:=
V_TotalConf+1
S_RscId :=
A_RscList
(V_TotalConf)
Determine Index
within A_VPCPIDList
V_VPCPIDIdx
:=
CALL conv_rscid
(S_RscId)
MIE_VPC_
_unblock_ind
(S_RscId) TO
A_VPCPIDList(V_VPCPIDIdx)!
P_VPCPID
true
CALL
ie_contained
(A_RscList
(V_TotalConf+1))
false
B_LSPUblDone
:=true
V_RecConf:=0
B_VPCUblDone
:=false
AwaitReset_
FSMAck

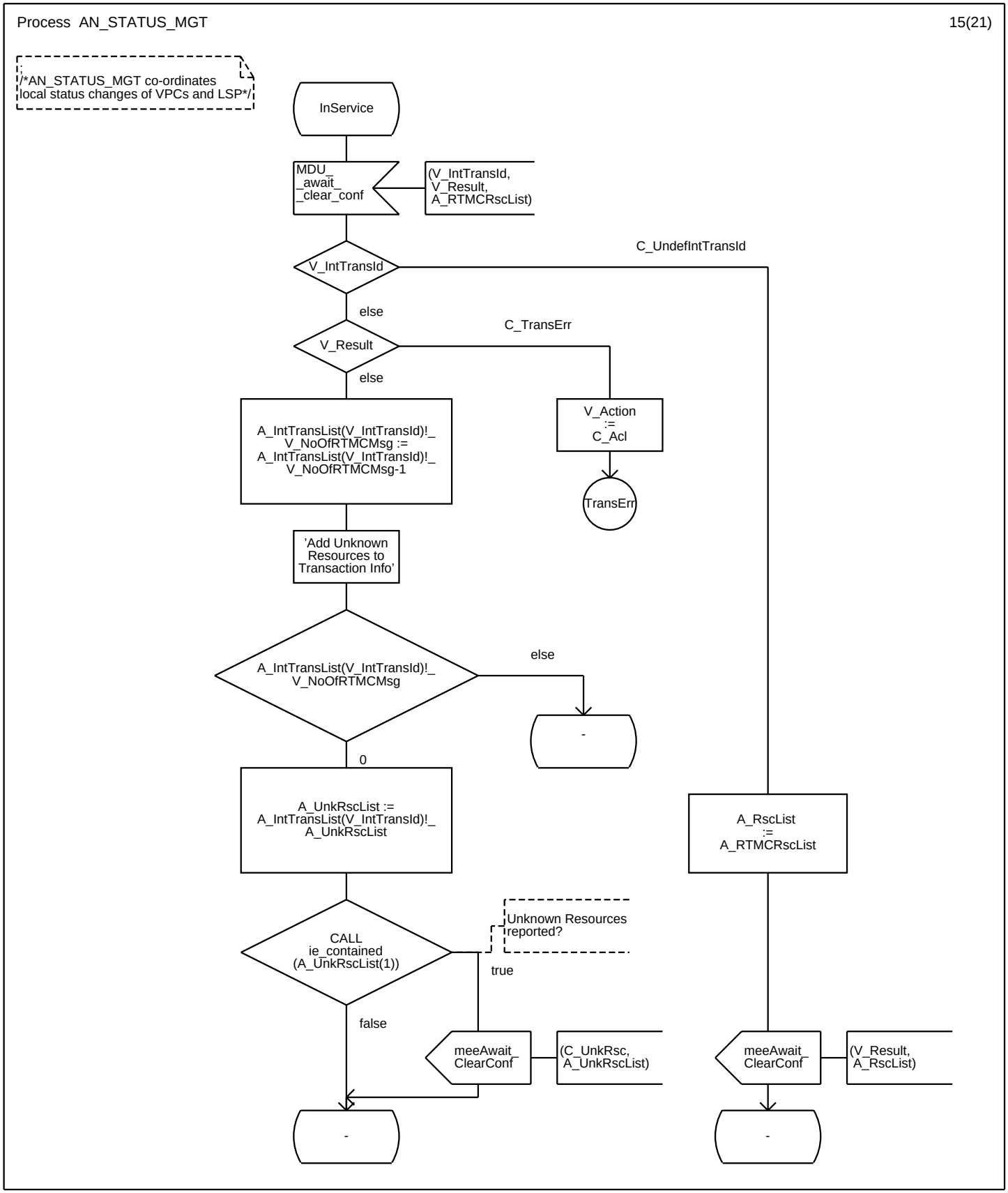

;
/*AN_STATUS_MGT co-ordinates
local status changes of VPCs and LSP*/
InService
(A_RscList)
mee_
Unblock_
RscReq
V_TotalConf
:= 0
else
A_RscList(1)!_
V_RscInd
C_CompLSP
V_TotalConf:=
V_TotalConf+1
S_RscId :=
A_RscList
(V_TotalConf)
Determine Index
within A_VPCPIDList
V_VPCPIDIdx
:=
CALL conv_rscid
(S_RscId)
MIE_VPC_
_unblock_req
(S_RscId) TO
A_VPCPIDList(V_VPCPIDIdx)!
P_VPCPID
V_TotalConf:=
V_TotalConf+1
S_RscId :=
A_RscList
(V_TotalConf)
MIE_LSP_
_unblock_req
(S_RscId)
CALL
ie_contained
(A_RscList
(V_TotalConf+1))
true
false
V_RecConf
:=0
AwaitUbl_
FSMAck

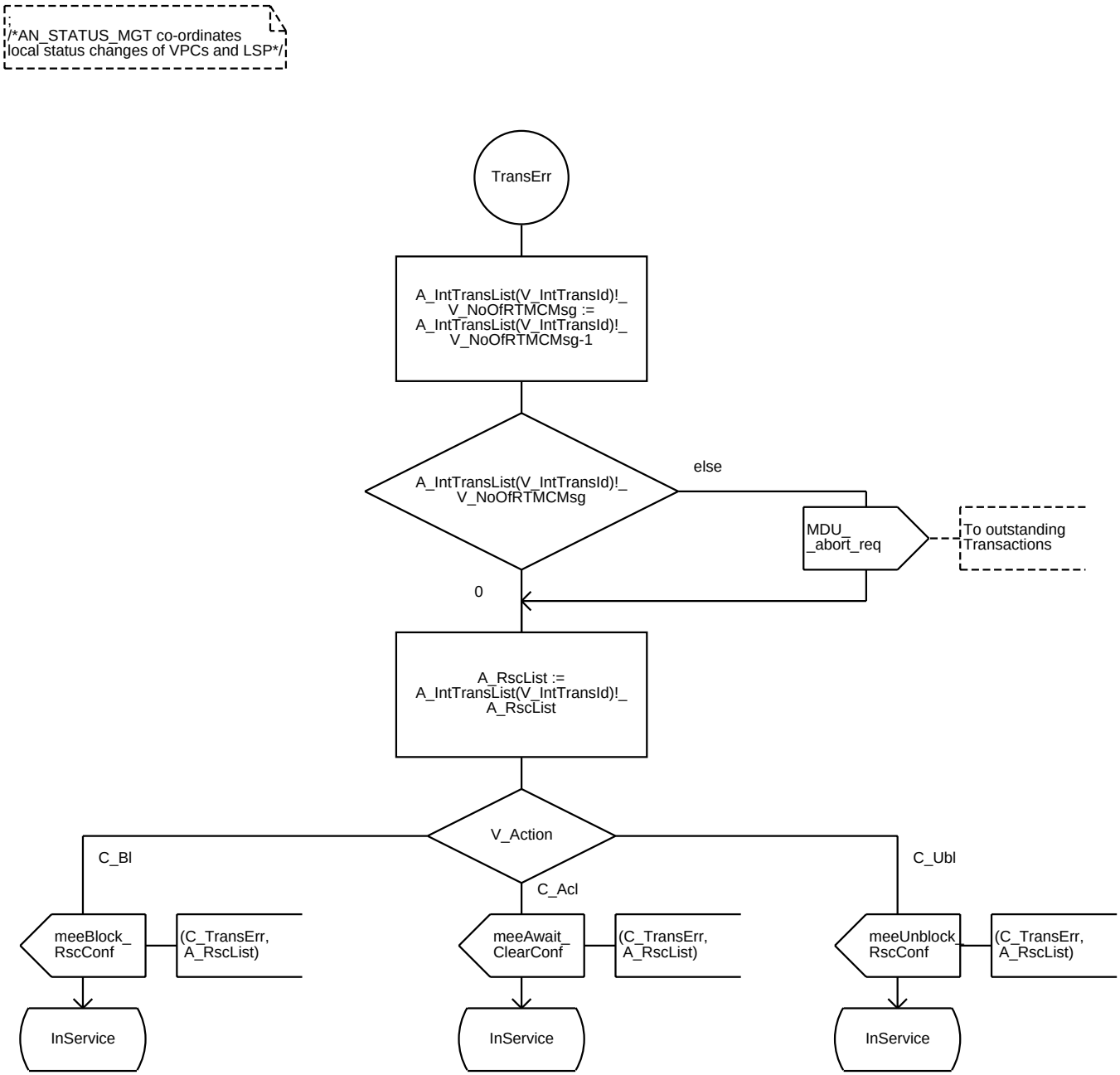

Process AN_STATUS_MGT
11(21)
;
/*AN_STATUS_MGT co-ordinates
local status changes of VPCs and LSP*/
InService
(A_BlRscList)
meeBlock_
RscReq
V_TotalConf
:= 0
else
A_BlRscList(1)!_
S_RscId!V_RscInd
C_CompLSP
true
V_TotalConf:=
V_TotalConf+1
V_TotalConf:=
V_TotalConf+1
S_RscId :=
A_BlRscList
(V_TotalConf)!S_RscId
S_RscId :=
A_BlRscList(1)!
S_RscId
V_Reason:=
A_BlRscList
(V_TotalConf)!
V_Reason
V_Reason:=
A_BlRscList(1)!
V_Reason
Determine Index
within A_VPCPIDList
V_VPCPIDIdx
:=
CALL conv_rscid
(S_RscId)
MIE_VPC_
_block_req
(S_RscId, V_Reason)
TO
A_VPCPIDList(V_VPCPIDIdx)!
P_VPCPID
MIE_LSP_
_block_req
(S_RscId,
V_Reason)
CALL
ie_contained
(A_BlRscList
(V_TotalConf+1)!
S_RscId)
false
V_RecConf
:=0
AwaitBl_
FSMAck

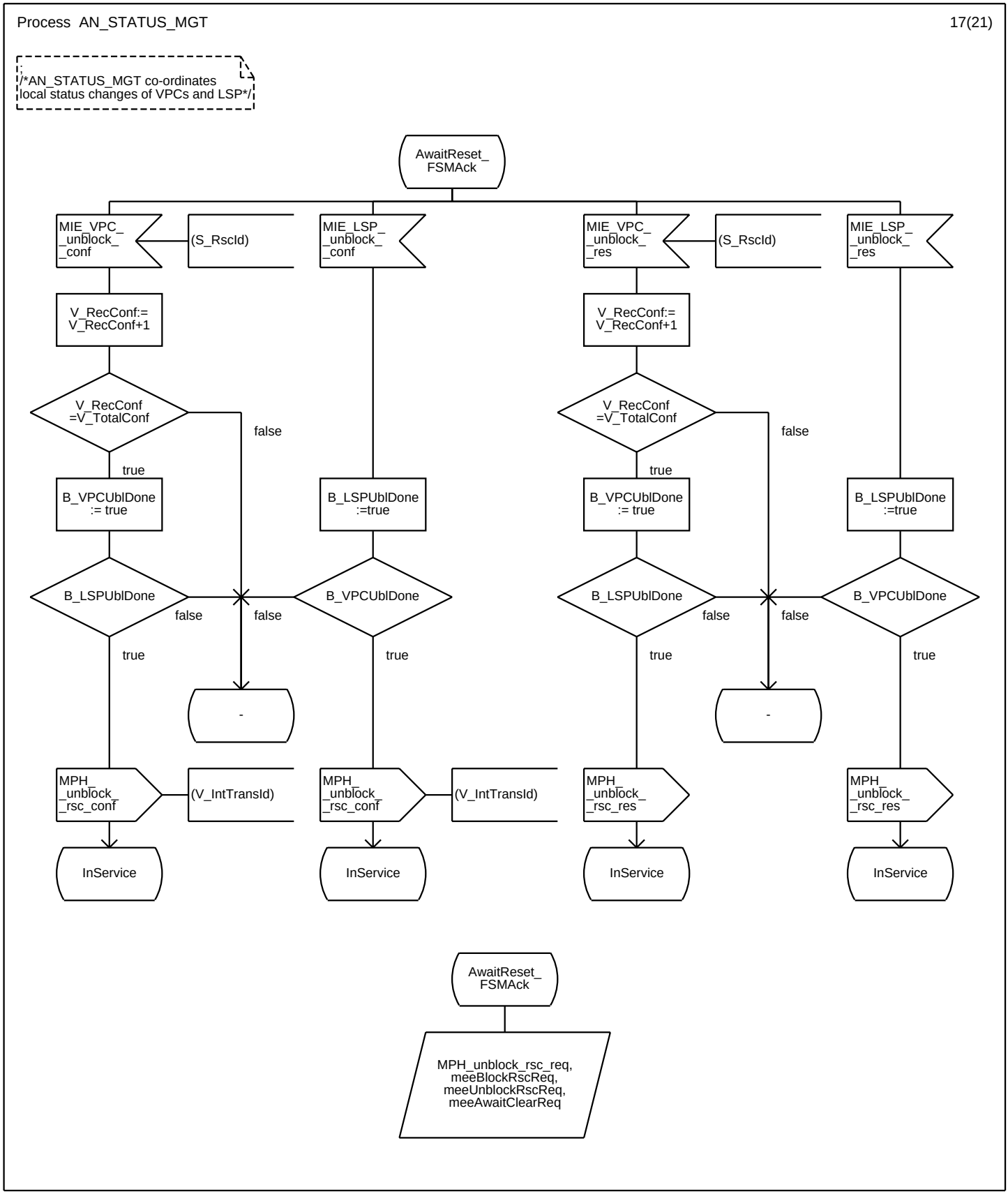

;
/*AN_STATUS_MGT co-ordinates
local status changes of VPCs and LSP*/
InService
(A_RscList)
meeAwait_
ClearReq
V_TotalConf
:= 0
V_TotalConf:=
V_TotalConf+1
S_RscId :=
A_RscList
(V_TotalConf)
Determine Index
within A_VPCPIDList
V_VPCPIDIdx
:=
CALL conv_rscid
(S_RscId)
MIE_VPC_
_await_clear_
_req
(S_RscId) TO
A_VPCPIDList(V_VPCPIDIdx)!
P_VPCPID
CALL
ie_contained
(A_RscList
(V_TotalConf+1))
true
false
V_RecConf
:=0
AwaitAcl_
FSMAck

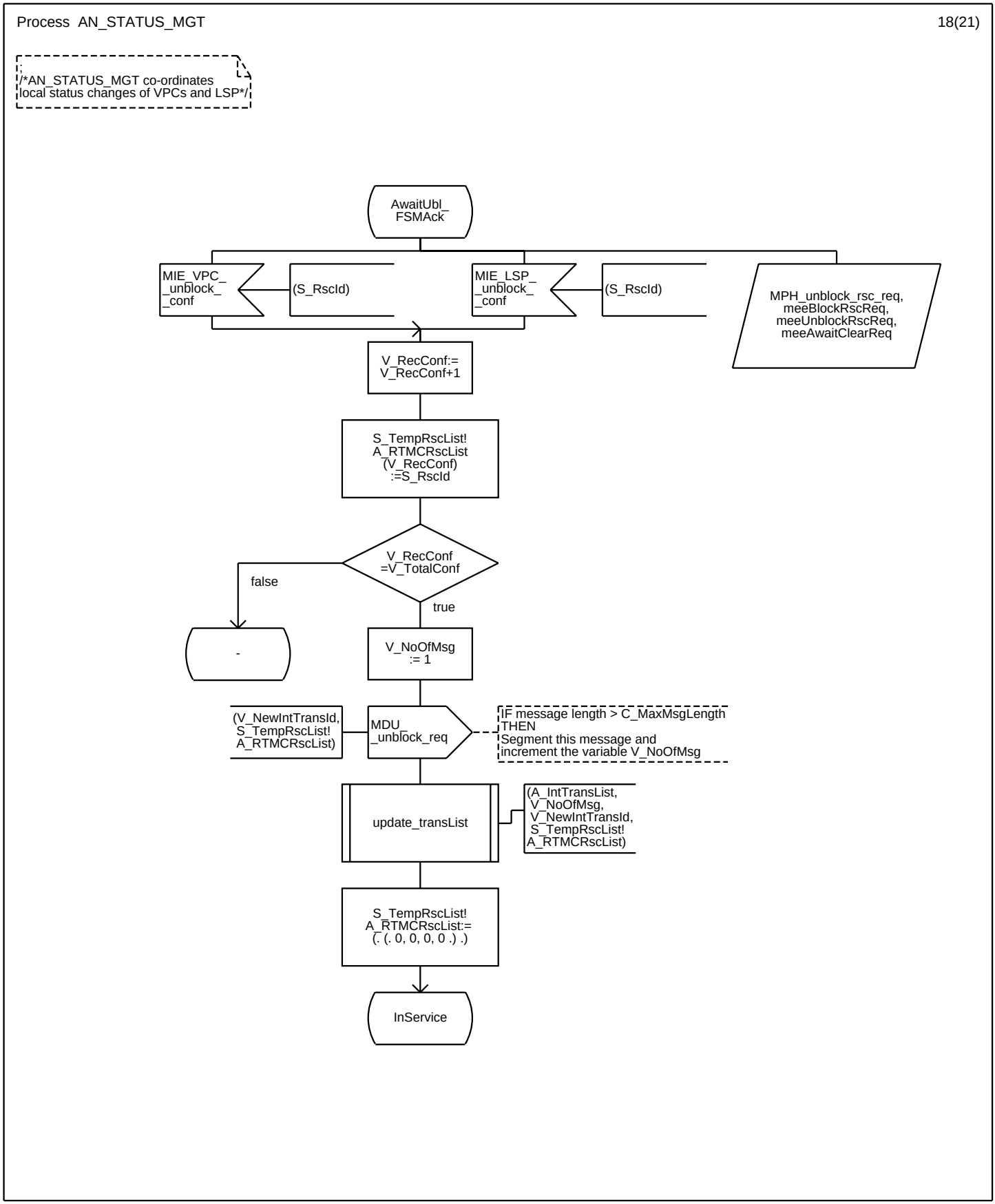

;
/*AN_STATUS_MGT co-ordinates
local status changes of VPCs and LSP*/
InService
MDU_
unblock
_conf
(V_IntTransId,
V_Result,
A_RTMCUnkRscList)
MDU_block_
_conf
(V_IntTransId,
V_Result,
A_RTMCUnkRscList)
V_Action
:=
C_Ubl
V_Action
:=
C_Bl
V_Result
C_TransErr
else
TransErr
A_IntTransList(V_IntTransId)!_
V_NoOfRTMCMsg :=
A_IntTransList(V_IntTransId)!_
V_NoOfRTMCMsg-1
'Add Unknown
Resources to
Transaction Info'
A_IntTransList(V_IntTransId)!_
V_NoOfRTMCMsg
else
0
-
A_UnkRscList :=
A_IntTransList(V_IntTransId)!_
A_UnkRscList
A
continues on next page

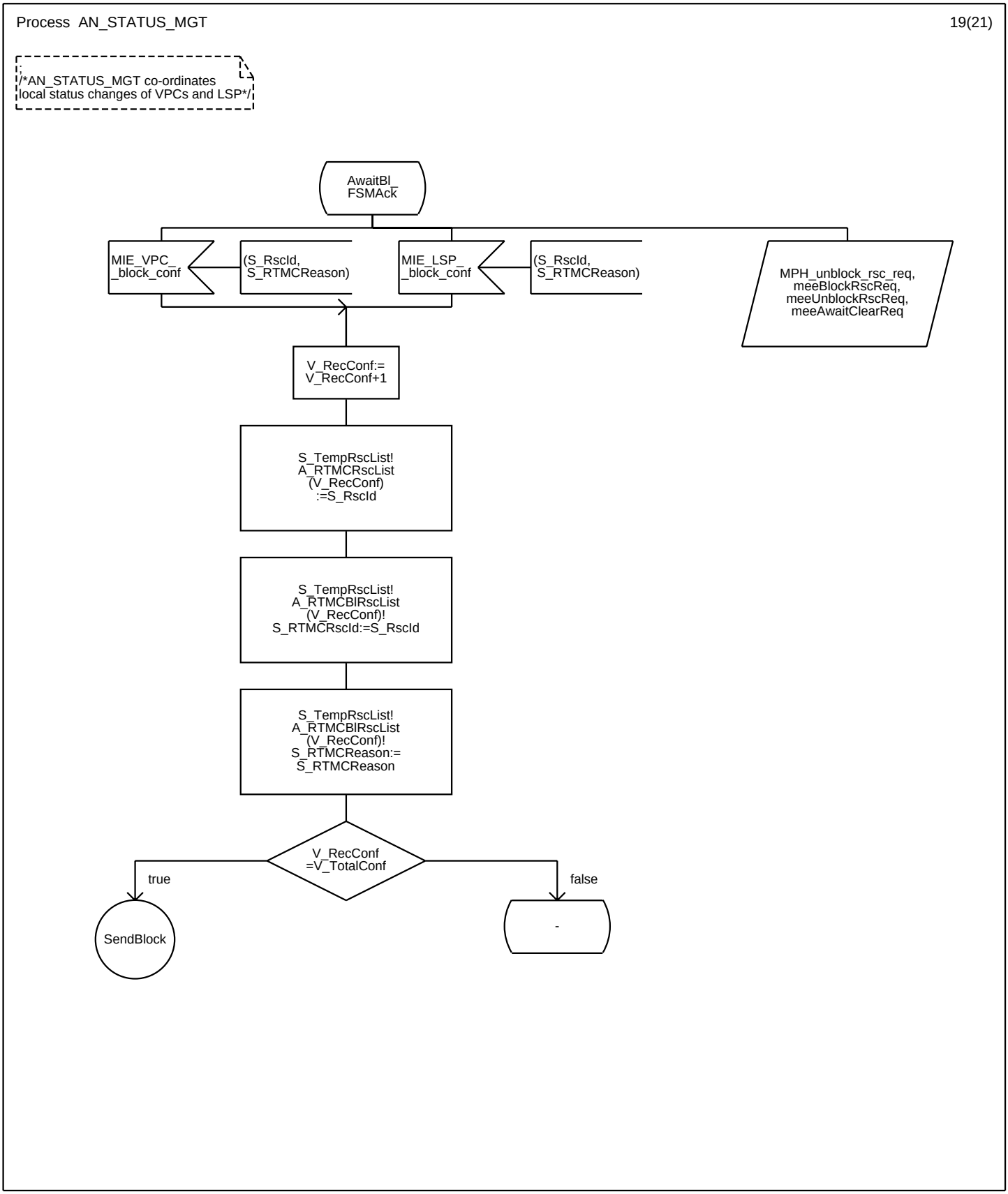

/*AN_STATUS_MGT co-ordinates
local status changes of VPCs and LSP*/
A
V_Action
C_Bl
C_Ubl
CALL
ie_contained
(A_UnkRscList(1))
Unknown Resources
reported?
false
true
(C_Success,)
(C_UnkRsc,
A_UnkRscList)
meeBlock_
RscConf
meeUnblock_
RscConf
-

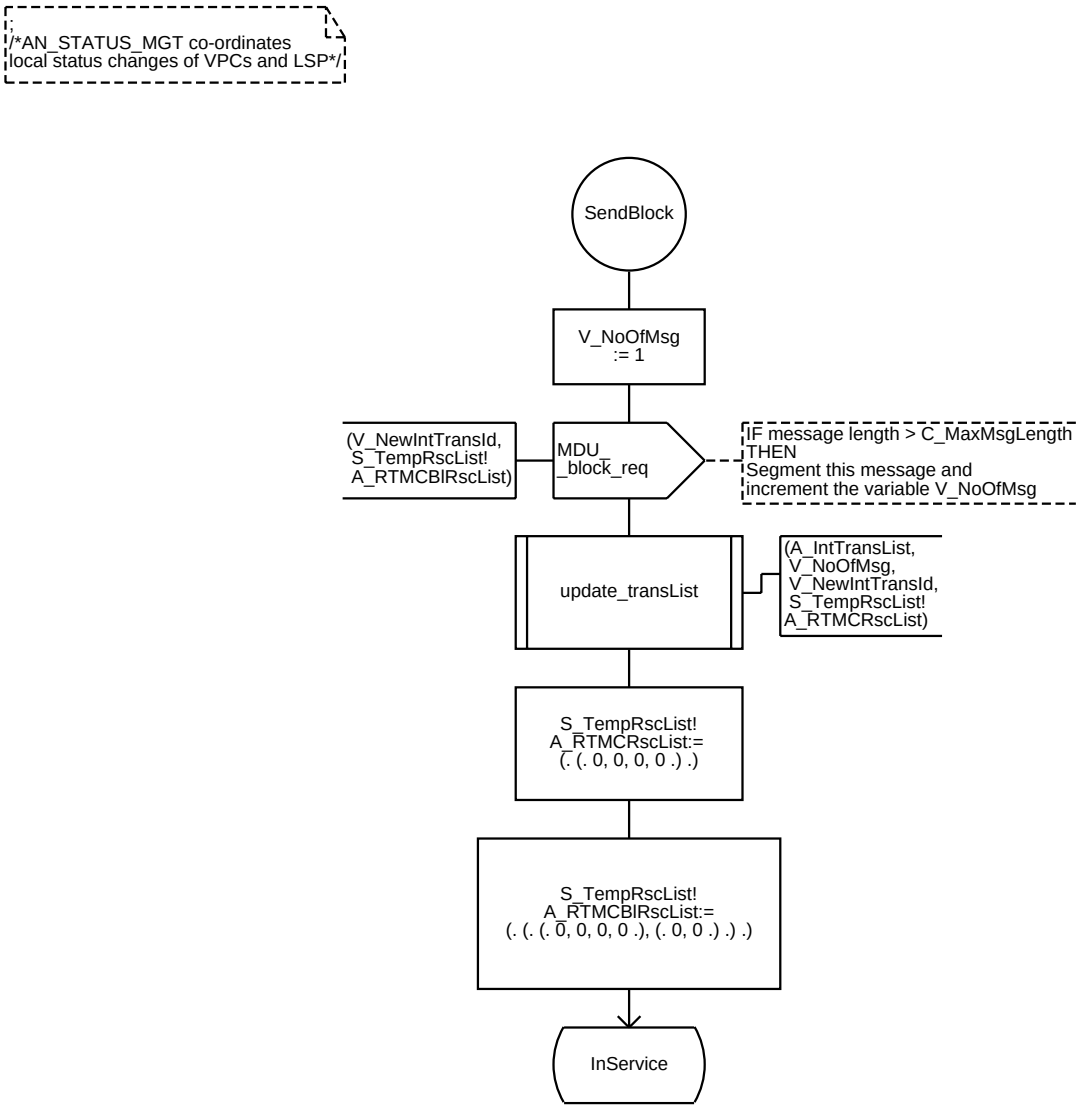

Process AN_STATUS_MGT
15(21)
;
/*AN_STATUS_MGT co-ordinates
local status changes of VPCs and LSP*/
InService
MDU_
await
_clear_conf
(V_IntTransId,
V_Result,
A_RTMCRscList)
V_IntTransId
C_UndefIntTransId
else
V_Result
C_TransErr
else
A_IntTransList(V_IntTransId)!_
V_NoOfRTMCMsg :=
A_IntTransList(V_IntTransId)!_
V_NoOfRTMCMsg-1
V_Action
:=
C_Acl
TransErr
'Add Unknown
Resources to
Transaction Info'
A_IntTransList(V_IntTransId)!_
V_NoOfRTMCMsg
else
-
0
A_UnkRscList :=
A_IntTransList(V_IntTransId)!_
A_UnkRscList
A_RscList
:=
A_RTMCRscList
CALL
ie_contained
(A_UnkRscList(1))
Unknown Resources
reported?
true
false
meeAwait_
ClearConf
(C_UnkRsc,
A_UnkRscList)
meeAwait_
ClearConf
(V_Result,
A_RscList)
-
-

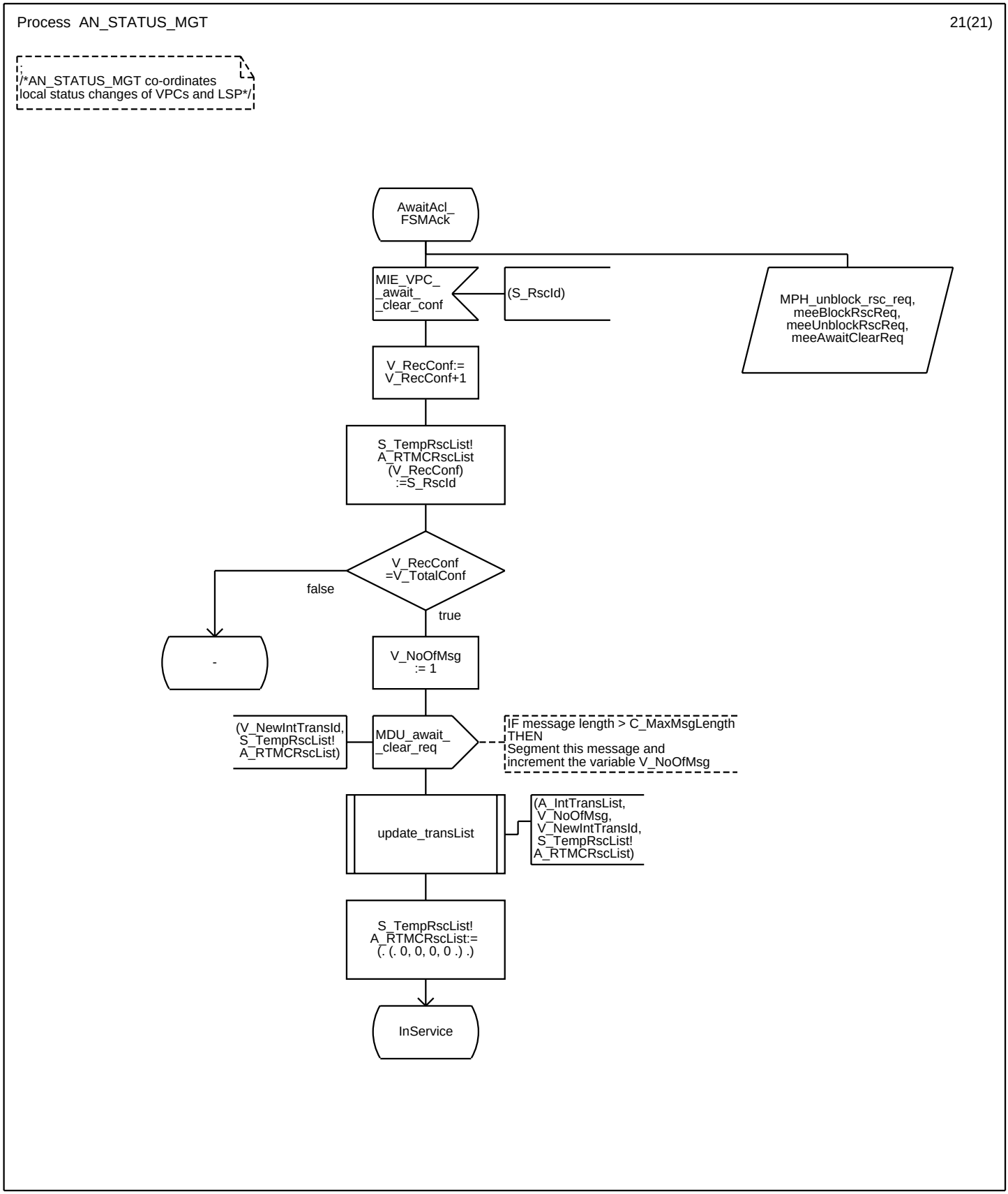

;
/*AN_STATUS_MGT co-ordinates
local status changes of VPCs and LSP*/
TransErr
A_IntTransList(V_IntTransId)!_
V_NoOfRTMCMsg :=
A_IntTransList(V_IntTransId)!_
V_NoOfRTMCMsg-1
A_IntTransList(V_IntTransId)!_
V_NoOfRTMCMsg
else
MDU_
_abort_req
To outstanding
Transactions
0
A_RscList :=
A_IntTransList(V_IntTransId)!_
A_RscList
V_Action
C_Bl
C_Acl
C_Ubl
meeBlock_
RscConf
(C_TransErr,
A_RscList)
meeAwait_
ClearConf
(C_TransErr,
A_RscList)
meeUnblock_
RscConf
(C_TransErr,
A_RscList)
InService
InService
InService


;
/*AN_STATUS_MGT co-ordinates
local status changes of VPCs and LSP*/
AwaitReset_
FSMAck
MIE_VPC_
unblock
_conf
(S_RscId)
MIE_LSP_
unblock
_conf
MIE_VPC_
unblock
_res
(S_RscId)
MIE_LSP_
unblock
_res
V_RecConf:=
V_RecConf+1
V_RecConf
=V_TotalConf
false
true
B_VPCUblDone
:= true
B_LSPUblDone
:=true
B_LSPUblDone
B_VPCUblDone
false
false
true
-
MPH_
unblock
_rsc_conf
(V_IntTransId)
MPH_
unblock
_rsc_res
InService
AwaitReset_
FSMAck
MPH_unblock_rsc_req,
meeBlockRscReq,
meeUnblockRscReq,
meeAwaitClearReq


;
/*AN_STATUS_MGT co-ordinates
local status changes of VPCs and LSP*/
AwaitUbl_
FSMAck
MIE_VPC_
unblock
_conf
(S_RscId)
MIE_LSP_
unblock
_conf
(S_RscId)
MPH_unblock_rsc_req,
meeBlockRscReq,
meeUnblockRscReq,
meeAwaitClearReq
V_RecConf:=
V_RecConf+1
S_TempRscList!
A_RTMCRscList
(V_RecConf)
:=S_RscId
V_RecConf
=V_TotalConf
false
true
-
V_NoOfMsg
:= 1
(V_NewIntTransId,
S_TempRscList!
A_RTMCRscList)
MDU_
_unblock_req
IF message length > C_MaxMsgLength
THEN
Segment this message and
increment the variable V_NoOfMsg
update_transList
(A_IntTransList,
V_NoOfMsg,
V_NewIntTransId,
S_TempRscList!
A_RTMCRscList)
S_TempRscList!
A_RTMCRscList:=
(. (. 0, 0, 0, 0 .) .)
InService


;
/*AN_STATUS_MGT co-ordinates
local status changes of VPCs and LSP*/
AwaitBl_
FSMAck
MIE_VPC_
_block_conf
(S_RscId,
S_RTMCReason)
MIE_LSP_
_block_conf
(S_RscId,
S_RTMCReason)
MPH_unblock_rsc_req,
meeBlockRscReq,
meeUnblockRscReq,
meeAwaitClearReq
V_RecConf:=
V_RecConf+1
S_TempRscList!
A_RTMCRscList
(V_RecConf)
:=S_RscId
S_TempRscList!
A_RTMCBlRscList
(V_RecConf)!
S_RTMCRscId:=S_RscId
S_TempRscList!
A_RTMCBlRscList
(V_RecConf)!
S_RTMCReason:=
S_RTMCReason
V_RecConf
=V_TotalConf
true
false
SendBlock
-


;
/*AN_STATUS_MGT co-ordinates
local status changes of VPCs and LSP*/
SendBlock
V_NoOfMsg
:= 1
(V_NewIntTransId,
S_TempRscList!
A_RTMCBlRscList)
MDU_
_block_req
IF message length > C_MaxMsgLength
THEN
Segment this message and
increment the variable V_NoOfMsg
update_transList
(A_IntTransList,
V_NoOfMsg,
V_NewIntTransId,
S_TempRscList!
A_RTMCRscList)
S_TempRscList!
A_RTMCRscList:=
(. (. 0, 0, 0, 0 .) .)
S_TempRscList!
A_RTMCBlRscList:=
(. (. (. 0, 0, 0, 0 .), (. 0, 0 .) .) .)
InService


;
/*AN_STATUS_MGT co-ordinates
local status changes of VPCs and LSP*/
AwaitAcl_
FSMAck
MIE_VPC_
await
_clear_conf
(S_RscId)
MPH_unblock_rsc_req,
meeBlockRscReq,
meeUnblockRscReq,
meeAwaitClearReq
V_RecConf:=
V_RecConf+1
S_TempRscList!
A_RTMCRscList
(V_RecConf)
:=S_RscId
V_RecConf
=V_TotalConf
false
true
-
V_NoOfMsg
:= 1
(V_NewIntTransId,
S_TempRscList!
A_RTMCRscList)
MDU_await_
_clear_req
IF message length > C_MaxMsgLength
THEN
Segment this message and
increment the variable V_NoOfMsg
update_transList
(A_IntTransList,
V_NoOfMsg,
V_NewIntTransId,
S_TempRscList!
A_RTMCRscList)
S_TempRscList!
A_RTMCRscList:=
(. (. 0, 0, 0, 0 .) .)
InService

Process AN_VPCSTAT 1(3)

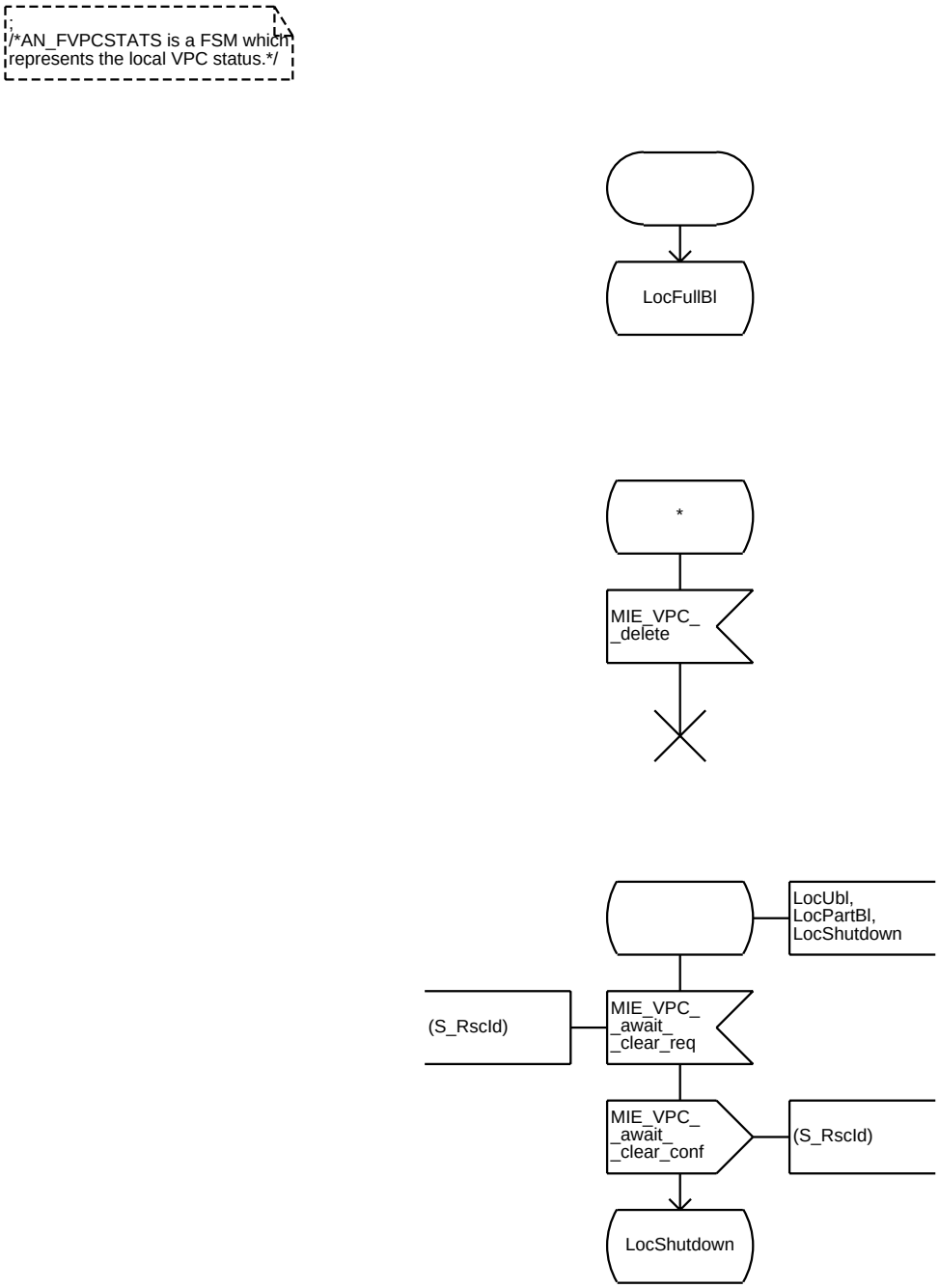
```
;
/*AN_FVPCSTATS is a FSM which
represents the local VPC status.*/
```

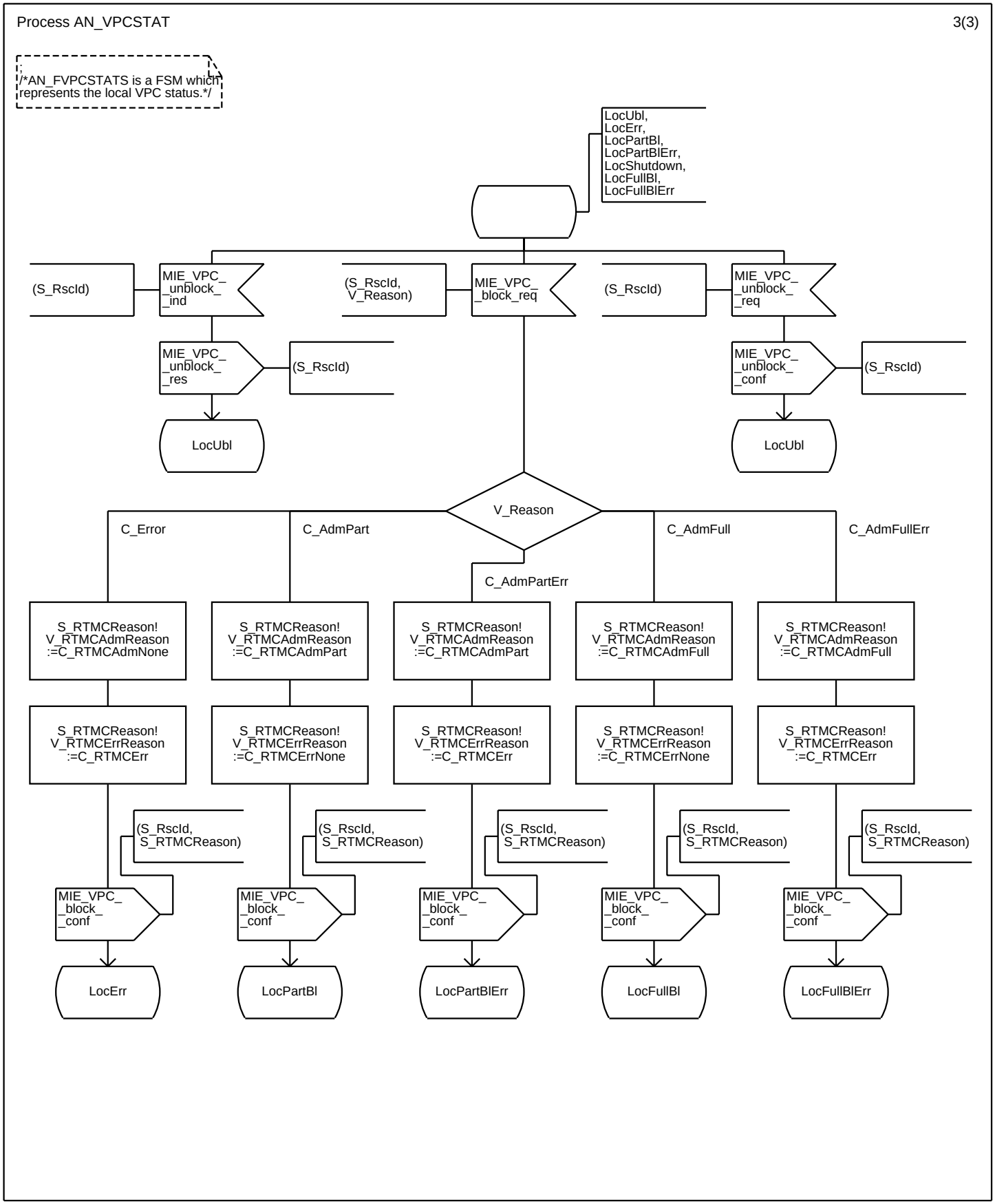
```
/*Definitions and
declarations for AN_VPCSTAT*/
```

```
/* State descriptions

LocUbl:          VPC available for Service
LocErr:          VPC unavailable due to fault conditions
LocPartBl:       VPC partially unavailable due to partial locking by AN operator
LocPartBlErr:    VPC unavailable due to partial locking by AN operator AND fault conditions
LocShutDown:     Shutting Down of VPC was requested by AN operator
LocFullBl:       VPC unavailable for service due to locking by AN operator
LocFullBlErr:    VPC unavailable due to locking by AN operator AND fault conditions
*/
```

```
/*Signal data declarations*/
DCL
V_Reason  IT_Reason; /*Reason for status change given by the environment*/
/**/
DCL
S_RTMCReason ST_RTMCReason; /*Reason for status change given to the STATUS_MGT*/
/**/
DCL
S_RscId ST_RscId;  /*Resource identification*/
```

```
/* The state of the VC carrying the RTMC protocol is affected by the
   administrative state of the VPC it belongs to. I.e. it is not possible
   to send or receive RTMC messages if the administrative state of the VPC
   is set to full blocked or the VPC is disabled. This is not considered
   in the SDLs but has to be taken into account in an implementation.
*/
```


/*AN_FVPCSTATS is a FSM which represents the local VPC status.*/
LocFullBl
*
MIE_VPC_
_delete
LocUbl,
LocPartBl,
LocShutdown
(S_RscId)
MIE_VPC_
await
_clear_req
MIE_VPC_
await
_clear_conf
(S_RscId)
LocShutdown

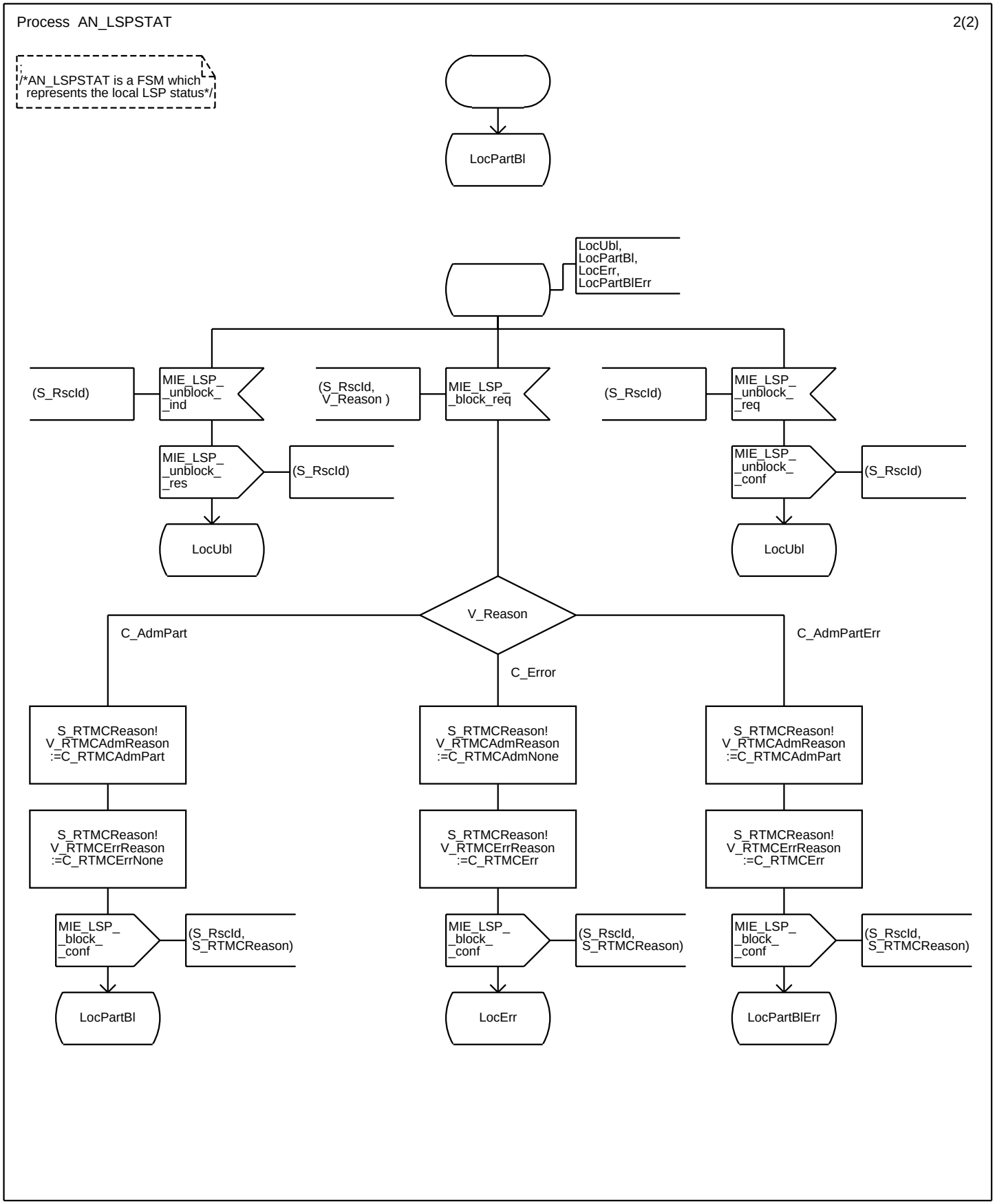

Process AN_VPCSTAT
3(3)
;
/*AN_FVPCSTATS is a FSM which represents the local VPC status.*/
LocUbl, LocErr, LocPartBl, LocPartBlErr, LocShutdown, LocFullBl, LocFullBlErr
(S_RscId)
MIE_VPC_ _unblock_ _ind
(S_RscId, V_Reason)
MIE_VPC_ _block_req
(S_RscId)
MIE_VPC_ _unblock_ _req
MIE_VPC_ _unblock_ _res
(S_RscId)
MIE_VPC_ _unblock_ _conf
(S_RscId)
LocUbl
LocUbl
V_Reason
C_Error
C_AdmPart
C_AdmPartErr
C_AdmFull
C_AdmFullErr
S_RTMCReason! V_RTMCAdmReason :=C_RTMCAdmNone
S_RTMCReason! V_RTMCAdmReason :=C_RTMCAdmPart
S_RTMCReason! V_RTMCAdmReason :=C_RTMCAdmPart
S_RTMCReason! V_RTMCAdmReason :=C_RTMCAdmFull
S_RTMCReason! V_RTMCAdmReason :=C_RTMCAdmFull
S_RTMCReason! V_RTMCErrReason :=C_RTMCErr
S_RTMCReason! V_RTMCErrReason :=C_RTMCErrNone
S_RTMCReason! V_RTMCErrReason :=C_RTMCErr
S_RTMCReason! V_RTMCErrReason :=C_RTMCErrNone
S_RTMCReason! V_RTMCErrReason :=C_RTMCErr
(S_RscId, S_RTMCReason)
(S_RscId, S_RTMCReason)
(S_RscId, S_RTMCReason)
(S_RscId, S_RTMCReason)
(S_RscId, S_RTMCReason)
MIE_VPC_ _block_ _conf
MIE_VPC_ _block_ _conf
MIE_VPC_ _block_ _conf
MIE_VPC_ _block_ _conf
MIE_VPC_ _block_ _conf
LocErr
LocPartBl
LocPartBlErr
LocFullBl
LocFullBlErr

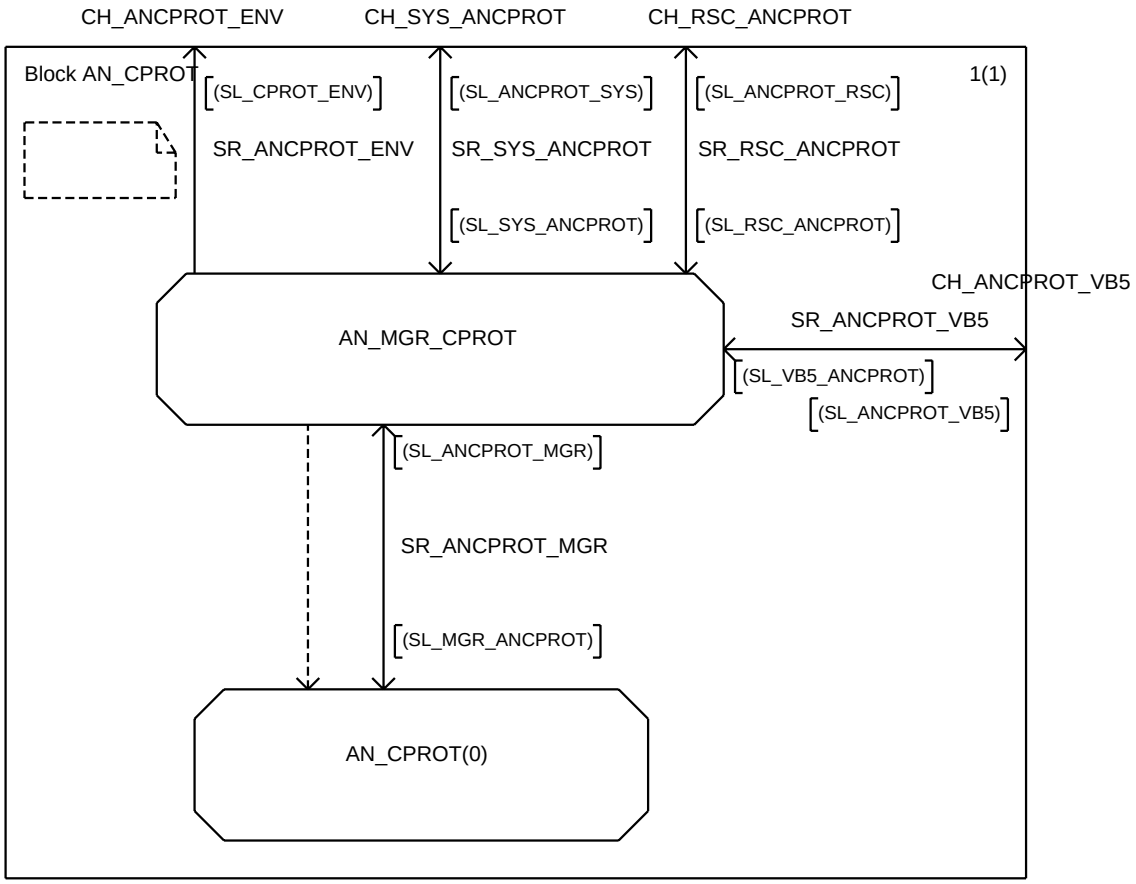
Process AN_LSPSTAT 1(2)

```
;
/*AN_LSPSTAT is a FSM which
 represents the local LSP status*/
```

```
/* State descriptions

LocUbl:        LSP available for Service
LocPartBl:     LSP partially unavailable due to partial locking by AN operator
LocErr:        LSP unavailable due to fault conditions
LocPartBlErr: LSP unavailable due to partial locking by AN operator AND fault conditions  */
```

```
/*Signal data declarations*/
DCL
V_Reason IT_Reason; /*Reason for status change given by the environment*/
/**/
DCL
S_RTMCReason ST_RTMCReason; /*Reason for status change given to the STATUS_MGT*/
/**/
DCL
S_RscId ST_RscId;  /*Resource identification*/
```


Process AN_LSPSTAT
2(2)
/*AN_LSPSTAT is a FSM which represents the local LSP status*/
LocPartBl
LocUbl, LocPartBl, LocErr, LocPartBlErr
(S_RscId)
MIE_LSP_ _unblock_ _ind
(S_RscId, V_Reason)
MIE_LSP_ _block_req
(S_RscId)
MIE_LSP_ _unblock_ _req
MIE_LSP_ _unblock_ _res
(S_RscId)
MIE_LSP_ _unblock_ _conf
(S_RscId)
LocUbl
LocUbl
V_Reason
C_AdmPart
C_Error
C_AdmPartErr
S_RTMCReason! V_RTMCAdmReason :=C_RTMCAdmPart
S_RTMCReason! V_RTMCAdmReason :=C_RTMCAdmNone
S_RTMCReason! V_RTMCAdmReason :=C_RTMCAdmPart
S_RTMCReason! V_RTMCErrReason :=C_RTMCErrNone
S_RTMCReason! V_RTMCErrReason :=C_RTMCErr
S_RTMCReason! V_RTMCErrReason :=C_RTMCErr
MIE_LSP_ _block_ _conf
(S_RscId, S_RTMCReason)
MIE_LSP_ _block_ _conf
(S_RscId, S_RTMCReason)
MIE_LSP_ _block_ _conf
(S_RscId, S_RTMCReason)
LocPartBl
LocErr
LocPartBlErr

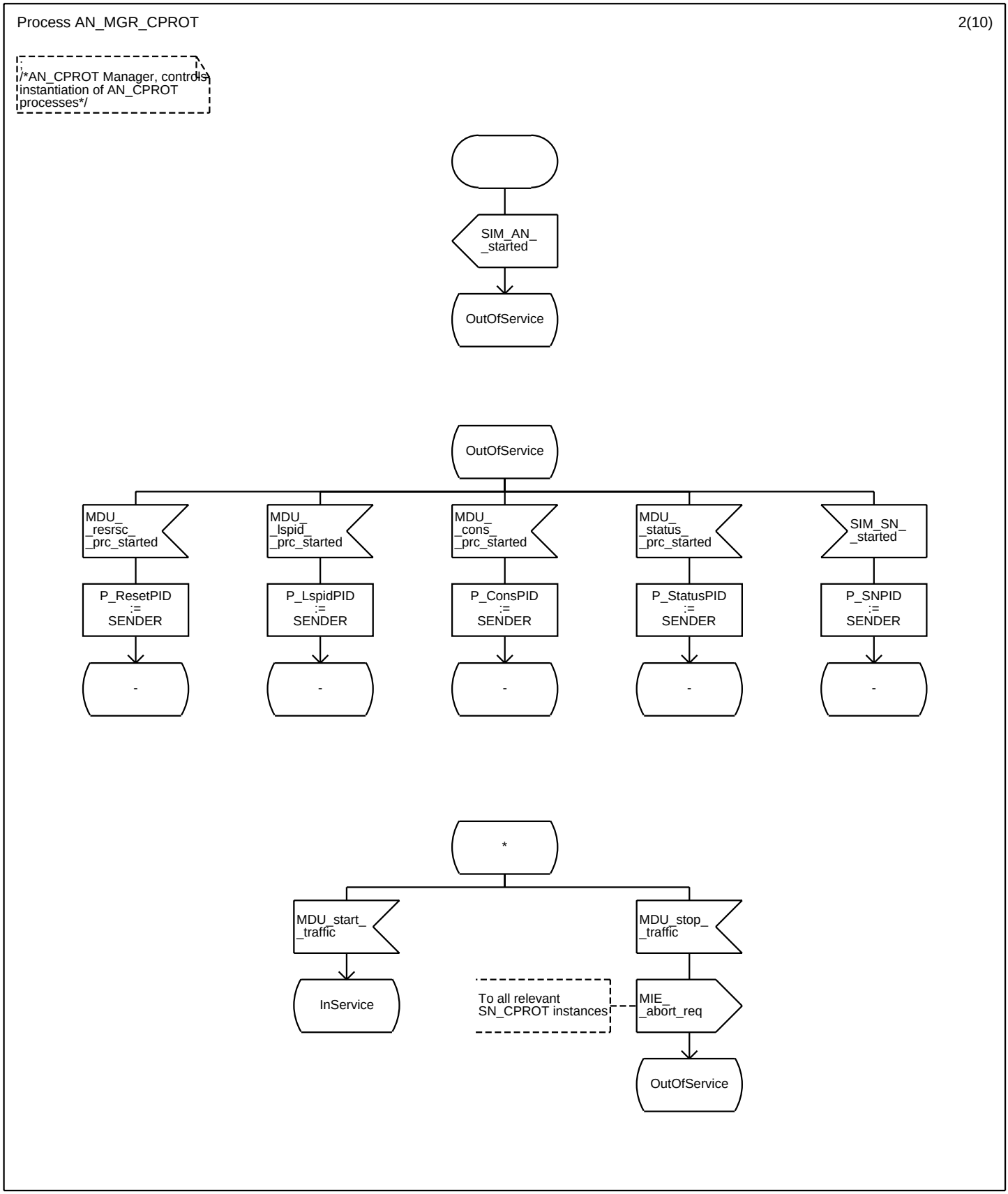

CH_ANCPROT_ENV
CH_SYS_ANCPROT
CH_RSC_ANCPROT
Block AN_CPROT
1(1)
(SL_CPROT_ENV)
(SL_ANCPROT_SYS)
(SL_ANCPROT_RSC)
SR_ANCPROT_ENV
SR_SYS_ANCPROT
SR_RSC_ANCPROT
(SL_SYS_ANCPROT)
(SL_RSC_ANCPROT)
CH_ANCPROT_VB5
SR_ANCPROT_VB5
AN_MGR_CPROT
(SL_VB5_ANCPROT)
(SL_ANCPROT_VB5)
(SL_ANCPROT_MGR)
SR_ANCPROT_MGR
(SL_MGR_ANCPROT)
AN_CPROT(0)

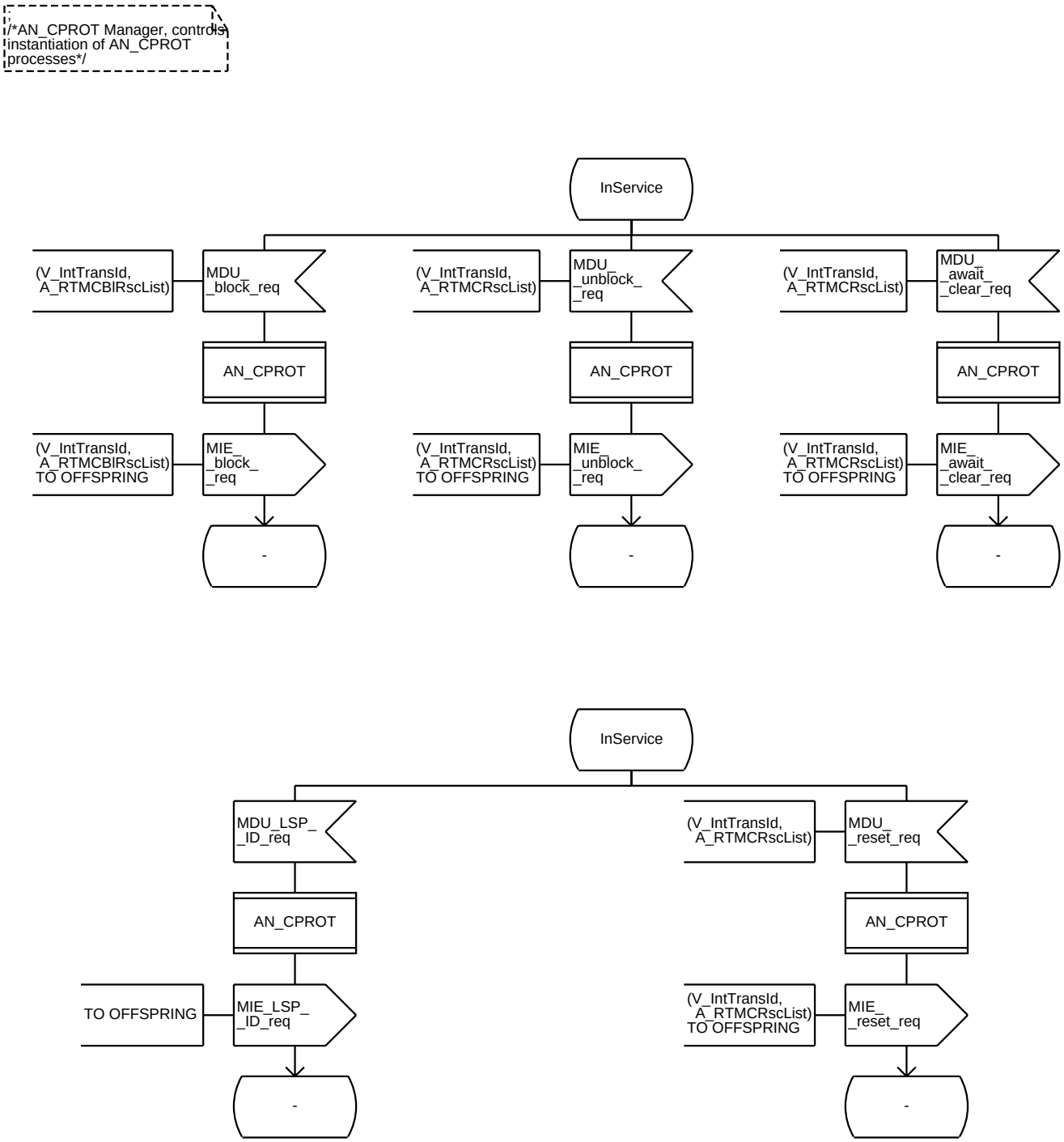
Process AN_MGR_CPROT 1(10)

```
/*AN_CPROT Manager, controls
instantiation of AN_CPROT
processes*/
```

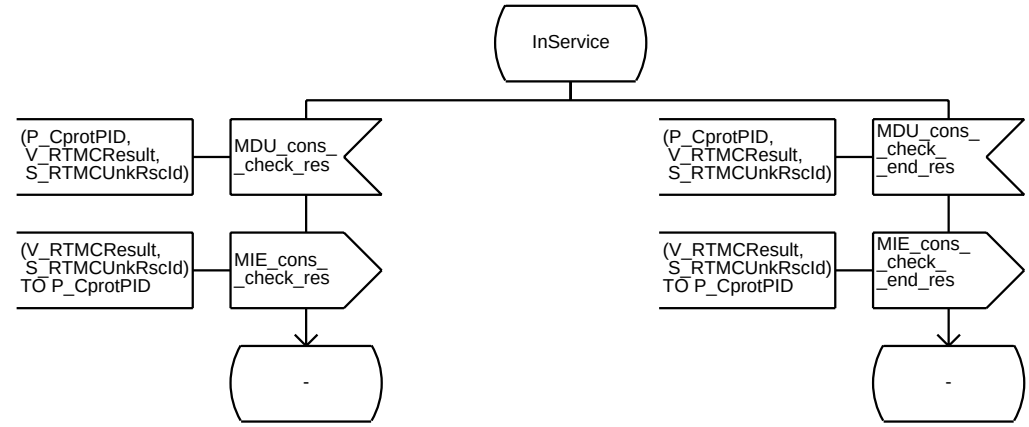
```
NEWTYPE AT_PIDList ARRAY /*This array maps the PID of the AN_CPROT instance on the PID of the requesting system management process*/
   (PID, PID)
ENDNEWTYPE AT_PIDList;
```

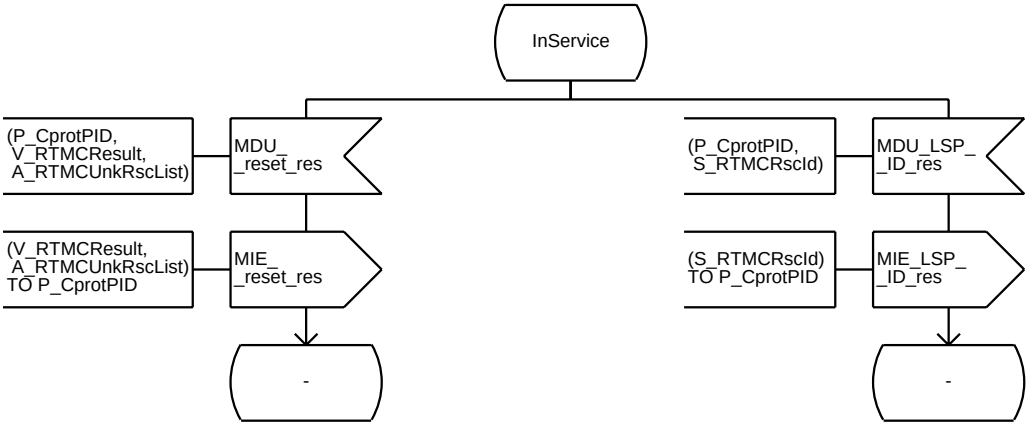
```
/*Primitive Data Declarations*/
DCL
A_RTMCBlRscList AT_RTMCBlRscList;
/**/
DCL
A_RTMCRscList AT_RscList;
/**/
DCL
A_RTMCUnkRscList AT_RscList;
/**/
DCL
S_RTMCRscId ST_RscId;
/**/
DCL
S_RTMCUnkRscId ST_RscId;
/**/
DCL
V_RTMCResult IT_RTMCResult;
/**/
DCL
V_Result IT_Result;
/**/
DCL
V_IntTransId IT_IntTransId;
/**/
DCL
V_ProtErrCause IT_RTMCProtErrCause;
/**/
DCL
P_StatusPID,    /*PID of AN_STATUS_MGT process*/
P_ResetPID,     /*PID of RESET_RSC process*/
P_ConsPID,      /*PID of AN_CONS process*/
P_LSPIdPID,     /*PID of CHECK_LSPID process*/
P_CprotPID,     /*PID of AN_CPROT process instance*/
P_TransId,      /*Transaction Identifier*/
P_SNPID         /*PID of SN_CPROT_MGR*/
PID;
```

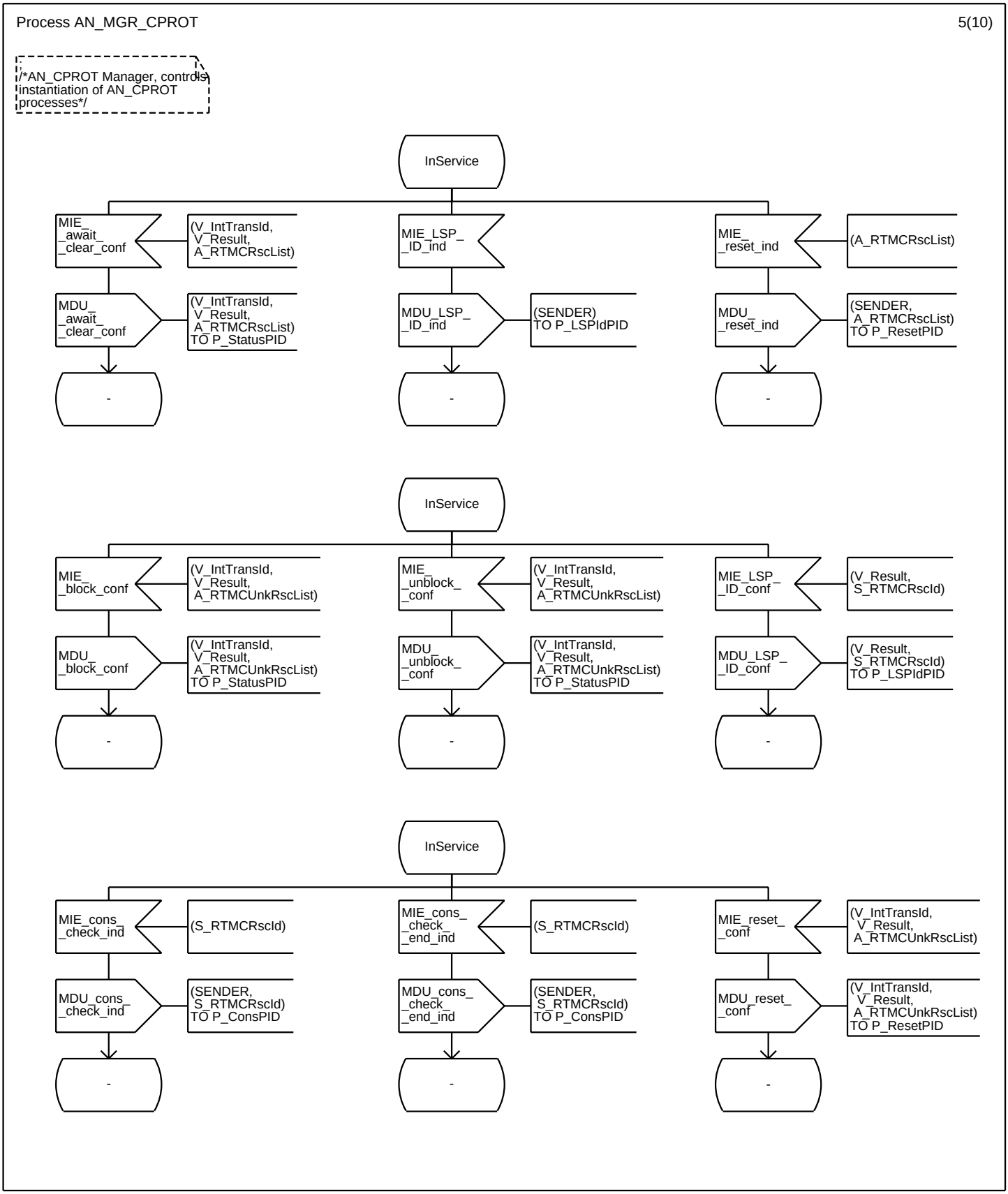

//*AN_CPROT Manager, controls instantiation of AN_CPROT processes*/
SIM_AN_ _started
OutOfService
OutOfService
MDU_ _resrsc_ _prc_started
MDU_ _lspid_ _prc_started
MDU_ _cons_ _prc_started
MDU_ _status_ _prc_started
SIM_SN_ _started
P_ResetPID := SENDER
P_LspidPID := SENDER
P_ConsPID := SENDER
P_StatusPID := SENDER
P_SNPID := SENDER
-
-
-
-
-
*
MDU_start_ _traffic
MDU_stop_ _traffic
InService
To all relevant SN_CPROT instances
MIE_ _abort_req
OutOfService

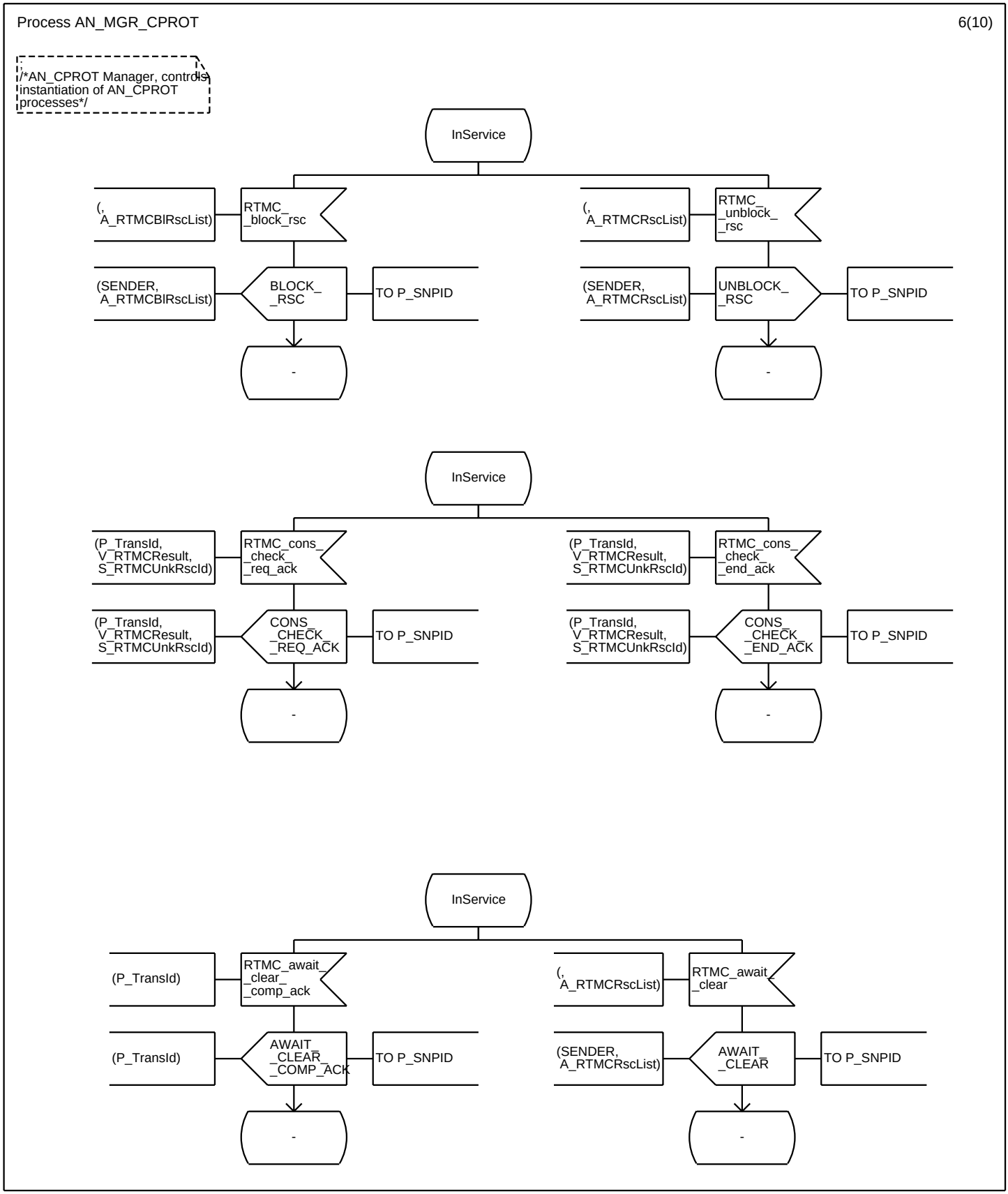

/*AN_CPROT Manager, controls instantiation of AN_CPROT processes*/
InService
(V_IntTransId, A_RTMCBlRscList)
MDU_ _block_req
AN_CPROT
(V_IntTransId, A_RTMCBlRscList) TO OFFSPRING
MIE_ _block_ _req
-
(V_IntTransId, A_RTMCRscList)
MDU_ _unblock_ _req
AN_CPROT
(V_IntTransId, A_RTMCRscList) TO OFFSPRING
MIE_ _unblock_ _req
-
(V_IntTransId, A_RTMCRscList)
MDU_ _await_ _clear_req
AN_CPROT
(V_IntTransId, A_RTMCRscList) TO OFFSPRING
MIE_ _await_ _clear_req
-
InService
MDU_LSP_ _ID_req
AN_CPROT
TO OFFSPRING
MIE_LSP_ _ID_req
-
(V_IntTransId, A_RTMCRscList)
MDU_ _reset_req
AN_CPROT
(V_IntTransId, A_RTMCRscList) TO OFFSPRING
MIE_ _reset_req
-

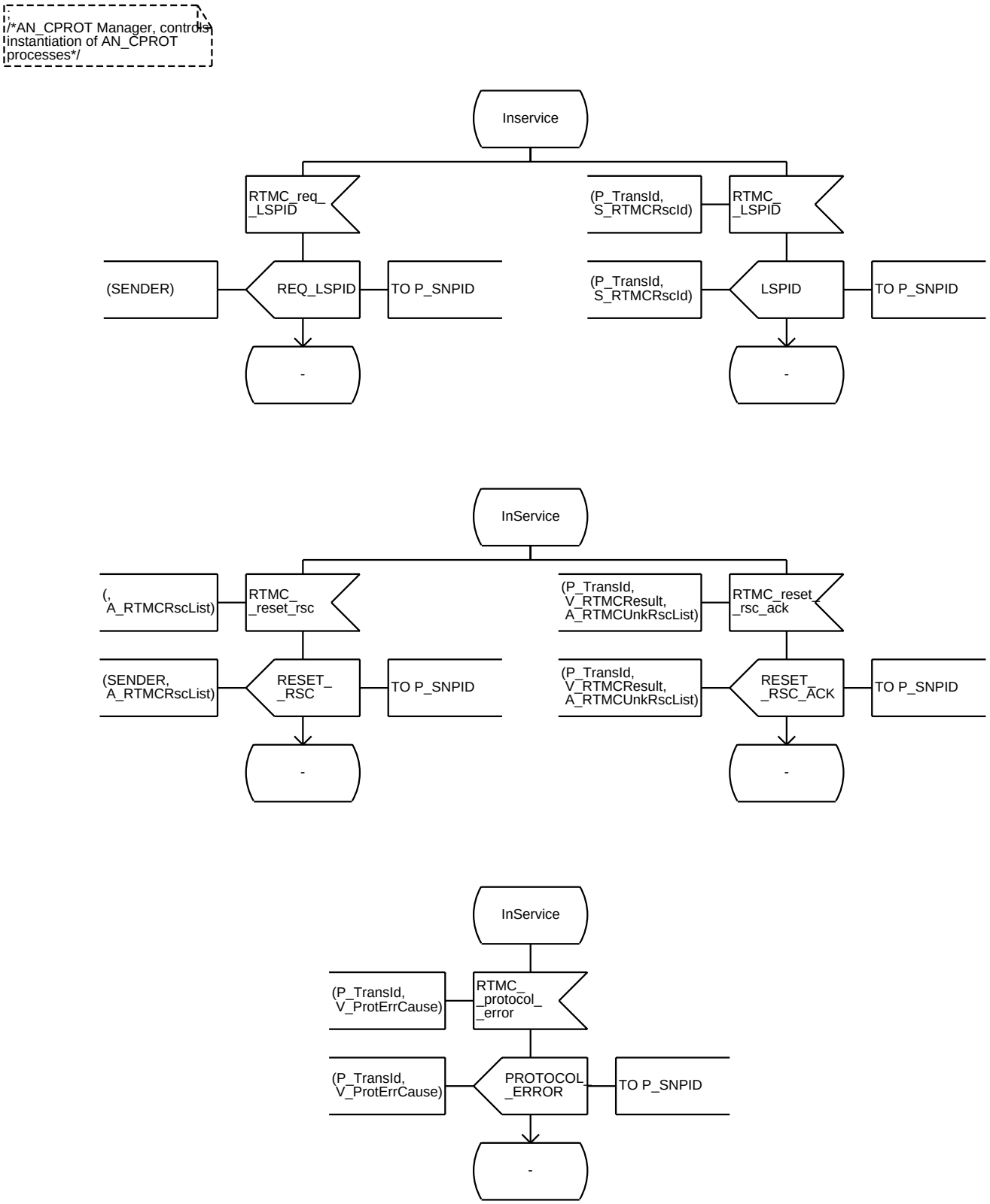
/*AN_CPROT Manager, controls instantiation of AN_CPROT processes*/

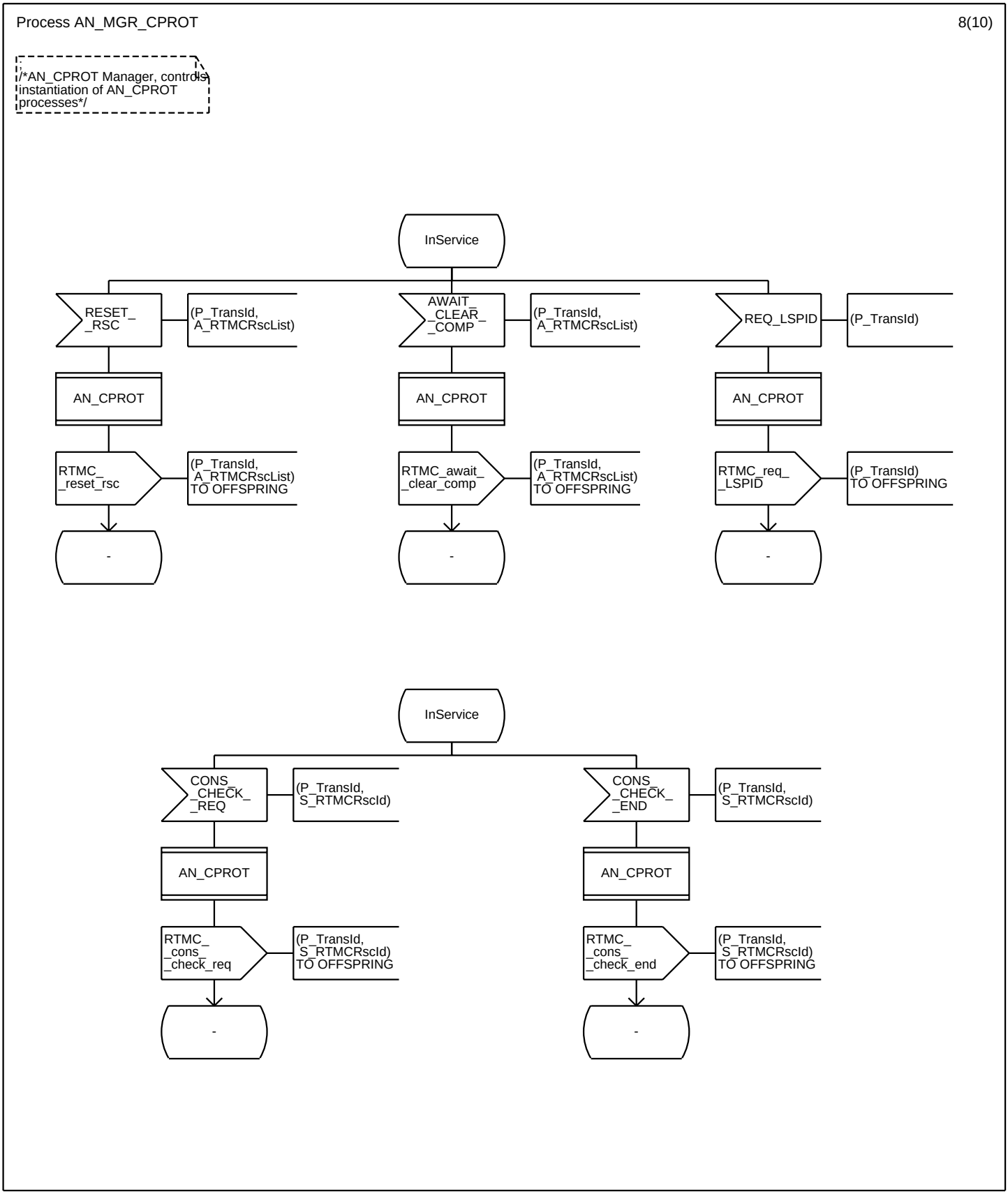


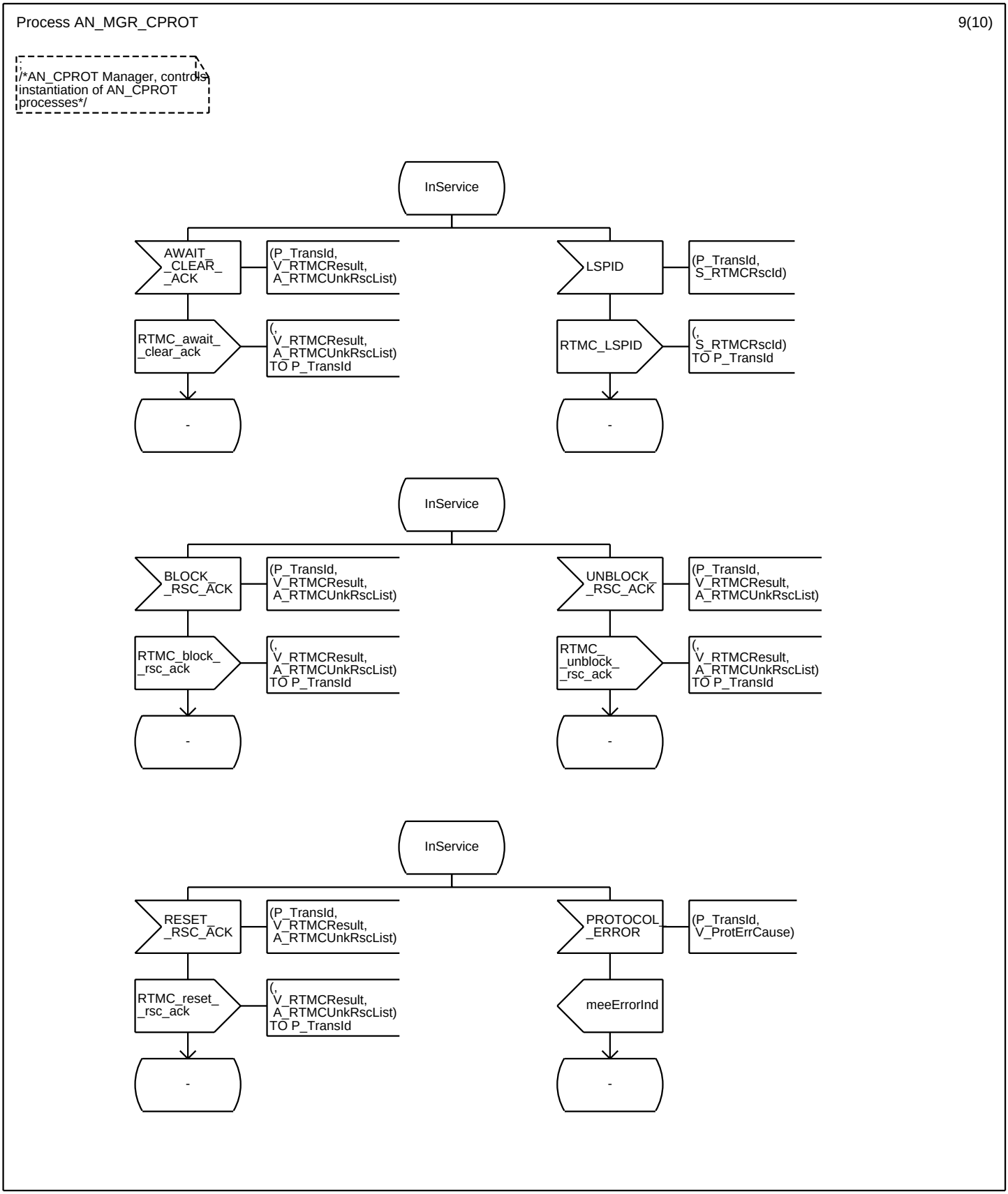


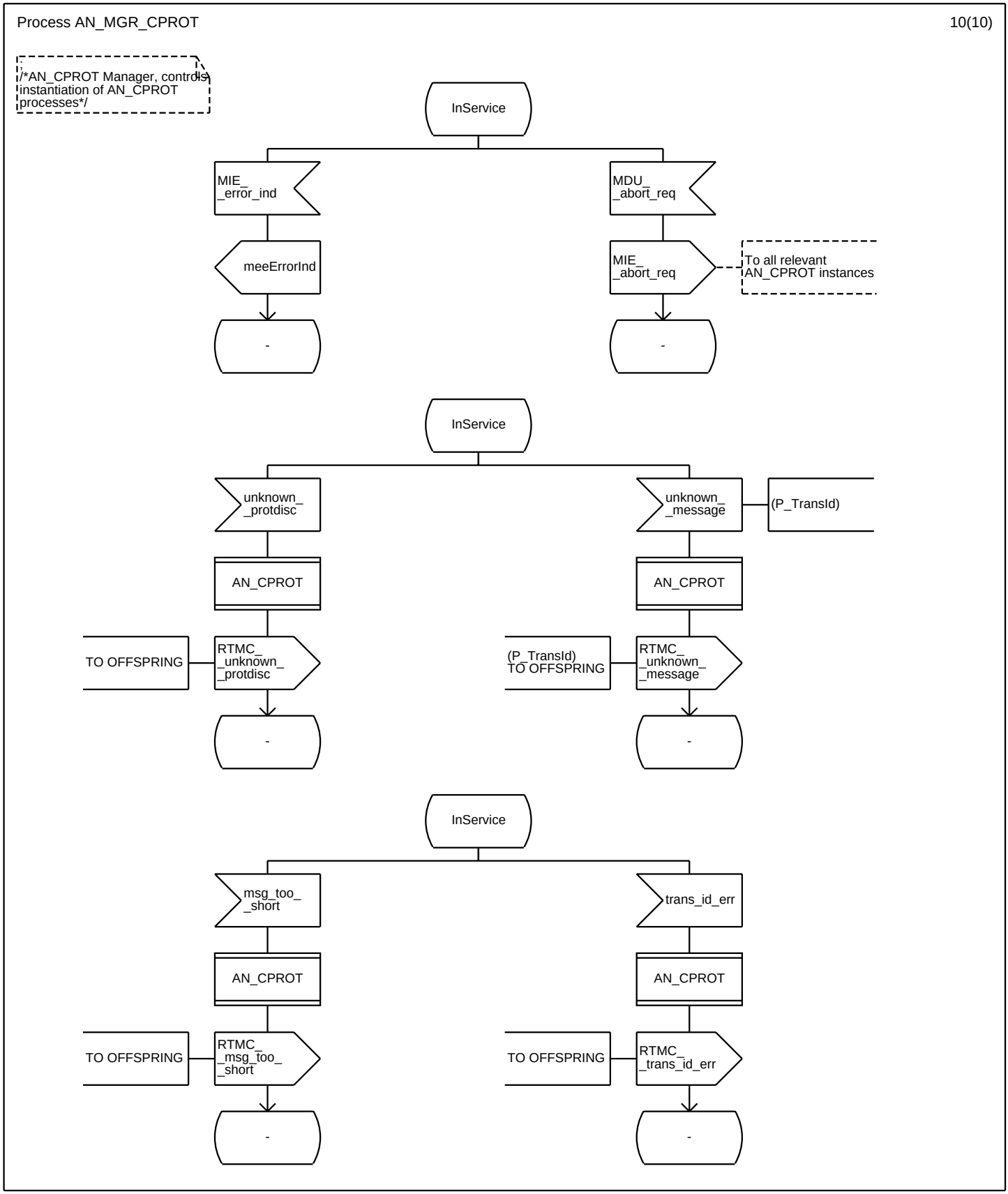

/*AN_CPROT Manager, controls instantiation of AN_CPROT processes*/
InService
MIE_ _await_ _clear_conf
(V_IntTransId, V_Result, A_RTMCRscList)
MDU_ _await_ _clear_conf
(V_IntTransId, V_Result, A_RTMCRscList) TO P_StatusPID
-
MIE_LSP_ _ID_ind
MDU_LSP_ _ID_ind
(SENDER) TO P_LSPIdPID
-
MIE_ _reset_ind
(A_RTMCRscList)
MDU_ _reset_ind
(SENDER, A_RTMCRscList) TO P_ResetPID
-
InService
MIE_ _block_conf
(V_IntTransId, V_Result, A_RTMCUnkRscList)
MDU_ _block_conf
(V_IntTransId, V_Result, A_RTMCUnkRscList) TO P_StatusPID
-
MIE_ _unblock_ _conf
(V_IntTransId, V_Result, A_RTMCUnkRscList)
MDU_ _unblock_ _conf
(V_IntTransId, V_Result, A_RTMCUnkRscList) TO P_StatusPID
-
MIE_LSP_ _ID_conf
(V_Result, S_RTMCRscId)
MDU_LSP_ _ID_conf
(V_Result, S_RTMCRscId) TO P_LSPIdPID
-
InService
MIE_cons_ _check_ind
(S_RTMCRscId)
MDU_cons_ _check_ind
(SENDER, S_RTMCRscId) TO P_ConsPID
-
MIE_cons_ _check_ _end_ind
(S_RTMCRscId)
MDU_cons_ _check_ _end_ind
(SENDER, S_RTMCRscId) TO P_ConsPID
-
MIE_reset_ _conf
(V_IntTransId, V_Result, A_RTMCUnkRscList)
MDU_reset_ _conf
(V_IntTransId, V_Result, A_RTMCUnkRscList) TO P_ResetPID
-


/*AN_CPROT Manager, controls instantiation of AN_CPROT processes*/
InService
(, A_RTMCBlRscList)
RTMC_ _block_rsc
(SENDER, A_RTMCBlRscList)
BLOCK_ _RSC
TO P_SNPID
-
(, A_RTMCRscList)
RTMC_ _unblock_ _rsc
(SENDER, A_RTMCRscList)
UNBLOCK_ _RSC
TO P_SNPID
-
InService
(P_TransId, V_RTMCResult, S_RTMCUnkRscId)
RTMC_cons_ _check_ _req_ack
(P_TransId, V_RTMCResult, S_RTMCUnkRscId)
CONS_ _CHECK_ _REQ_ACK
TO P_SNPID
-
(P_TransId, V_RTMCResult, S_RTMCUnkRscId)
RTMC_cons_ _check_ _end_ack
(P_TransId, V_RTMCResult, S_RTMCUnkRscId)
CONS_ _CHECK_ _END_ACK
TO P_SNPID
-
InService
(P_TransId)
RTMC_await_ _clear_ _comp_ack
(P_TransId)
AWAIT_ _CLEAR_ _COMP_ACK
TO P_SNPID
-
(, A_RTMCRscList)
RTMC_await_ _clear
(SENDER, A_RTMCRscList)
AWAIT_ _CLEAR
TO P_SNPID
-


/*AN_CPROT Manager, controls instantiation of AN_CPROT processes*/
Inservice
RTMC_req_
_LSPID
(SENDER)
REQ_LSPID
TO P_SNPID
-
(P_TransId,
S_RTMCRscId)
RTMC_
_LSPID
(P_TransId,
S_RTMCRscId)
LSPID
TO P_SNPID
-
InService
(,
A_RTMCRscList)
RTMC_
_reset_rsc
(SENDER,
A_RTMCRscList)
RESET_
_RSC
TO P_SNPID
-
(P_TransId,
V_RTMCResult,
A_RTMCUnkRscList)
RTMC_reset_
_rsc_ack
(P_TransId,
V_RTMCResult,
A_RTMCUnkRscList)
RESET_
_RSC_ACK
TO P_SNPID
-
InService
(P_TransId,
V_ProtErrCause)
RTMC_
protocol
_error
(P_TransId,
V_ProtErrCause)
PROTOCOL
_ERROR
TO P_SNPID
-


//*AN_CPROT Manager, controls instantiation of AN_CPROT processes*/
InService
RESET_ _RSC
(P_TransId, A_RTMCRscList)
AN_CPROT
RTMC_ _reset_rsc
(P_TransId, A_RTMCRscList) TO OFFSPRING
-
AWAIT_ _CLEAR_ _COMP
(P_TransId, A_RTMCRscList)
AN_CPROT
RTMC_await_ _clear_comp
(P_TransId, A_RTMCRscList) TO OFFSPRING
-
REQ_LSPID
(P_TransId)
AN_CPROT
RTMC_req_ _LSPID
(P_TransId) TO OFFSPRING
-
InService
CONS_ _CHECK_ _REQ
(P_TransId, S_RTMCRscId)
AN_CPROT
RTMC_ _cons_ _check_req
(P_TransId, S_RTMCRscId) TO OFFSPRING
-
CONS_ _CHECK_ _END
(P_TransId, S_RTMCRscId)
AN_CPROT
RTMC_ _cons_ _check_end
(P_TransId, S_RTMCRscId) TO OFFSPRING
-


/*AN_CPROT Manager, controls instantiation of AN_CPROT processes*/
InService
AWAIT_ _CLEAR_ _ACK
(P_TransId, V_RTMCResult, A_RTMCUnkRscList)
RTMC_await_ _clear_ack
(, V_RTMCResult, A_RTMCUnkRscList) TO P_TransId
-
LSPID
(P_TransId, S_RTMCRscId)
RTMC_LSPID
(, S_RTMCRscId) TO P_TransId
-
InService
BLOCK_ _RSC_ACK
(P_TransId, V_RTMCResult, A_RTMCUnkRscList)
RTMC_block_ _rsc_ack
(, V_RTMCResult, A_RTMCUnkRscList) TO P_TransId
-
UNBLOCK_ _RSC_ACK
(P_TransId, V_RTMCResult, A_RTMCUnkRscList)
RTMC_ _unblock_ _rsc_ack
(, V_RTMCResult, A_RTMCUnkRscList) TO P_TransId
-
InService
RESET_ _RSC_ACK
(P_TransId, V_RTMCResult, A_RTMCUnkRscList)
RTMC_reset_ _rsc_ack
(, V_RTMCResult, A_RTMCUnkRscList) TO P_TransId
-
PROTOCOL_ _ERROR
(P_TransId, V_ProtErrCause)
meeErrorInd
-


Process AN_MGR_CPROT
10(10)
/*AN_CPROT Manager, controls instantiation of AN_CPROT processes*/
InService
MIE_ _error_ind
MDU_ _abort_req
meeErrorInd
MIE_ _abort_req
To all relevant AN_CPROT instances
-
-
InService
unknown_ _protdisc
unknown_ _message
(P_TransId)
AN_CPROT
AN_CPROT
TO OFFSPRING
RTMC_ _unknown_ _protdisc
(P_TransId) TO OFFSPRING
RTMC_ _unknown_ _message
-
-
InService
msg_too_ _short
trans_id_err
AN_CPROT
AN_CPROT
TO OFFSPRING
RTMC_ _msg_too_ _short
TO OFFSPRING
RTMC_ _trans_id_err
-
-

Process AN_CPROT 1(16)

```
;
/*AN_CPROT is responsible for sending/
receiving RTMC messages*/
```

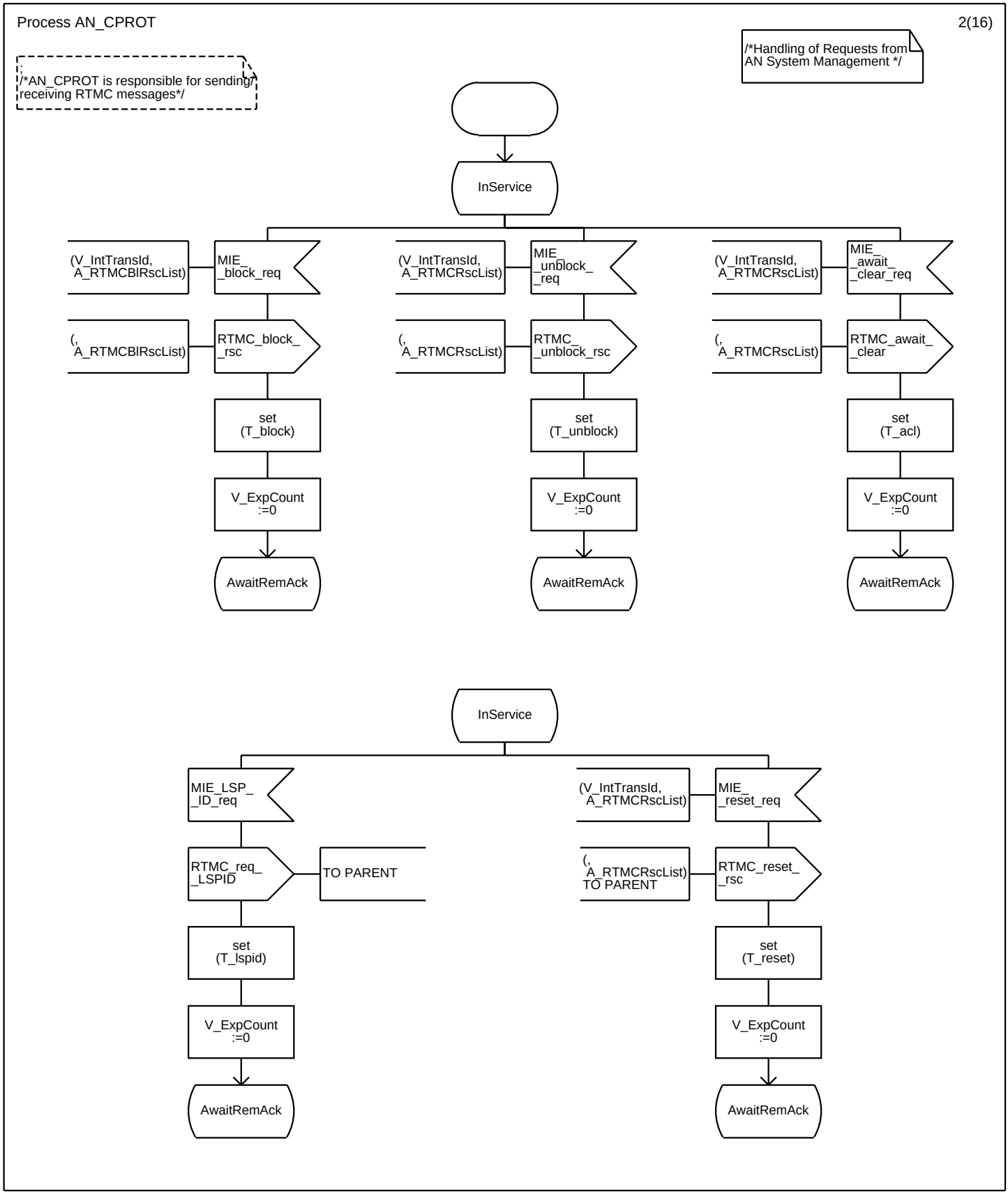
```
/* Definintions and declarations for
AN_CPROT */
```

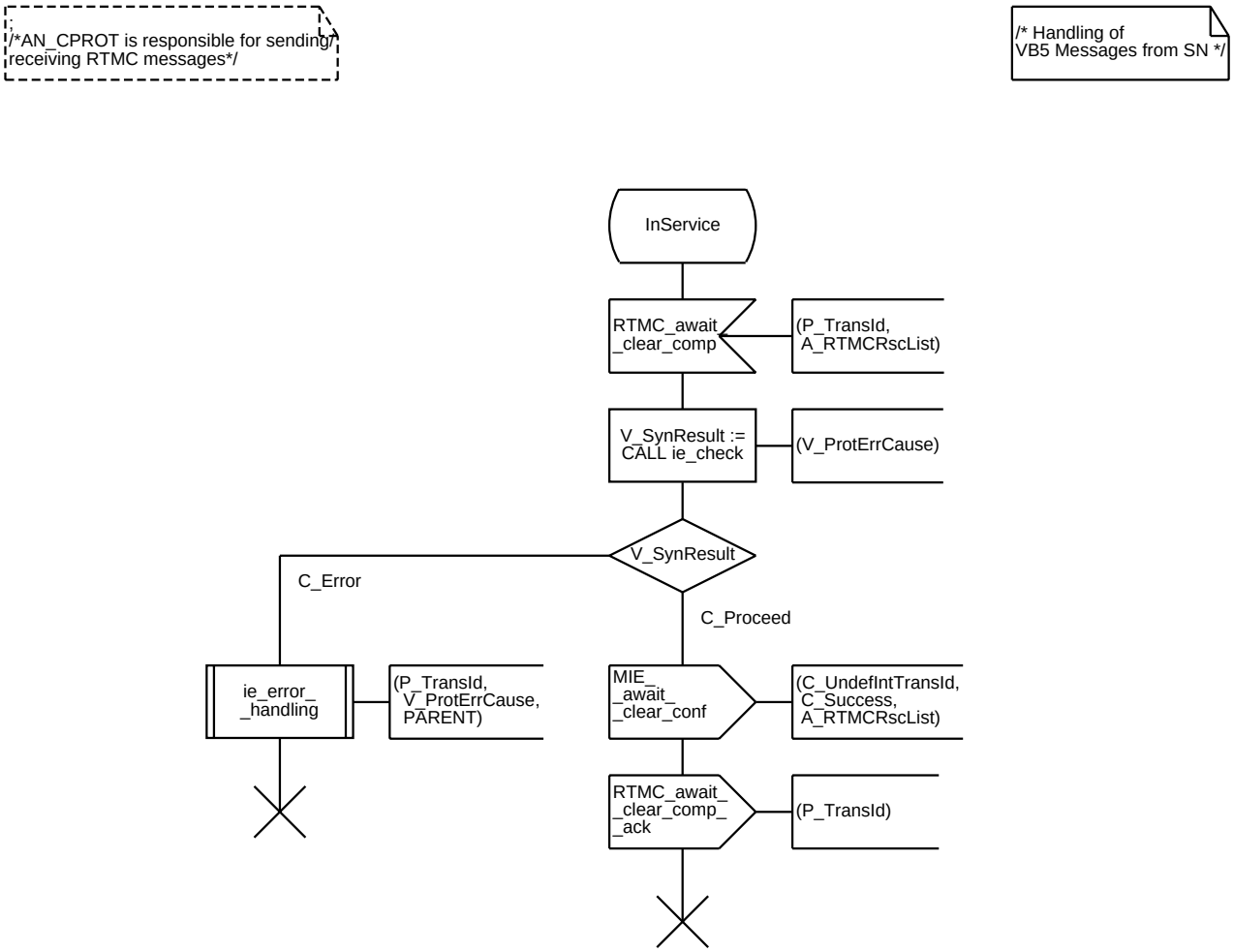
```
/* State descriptions

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

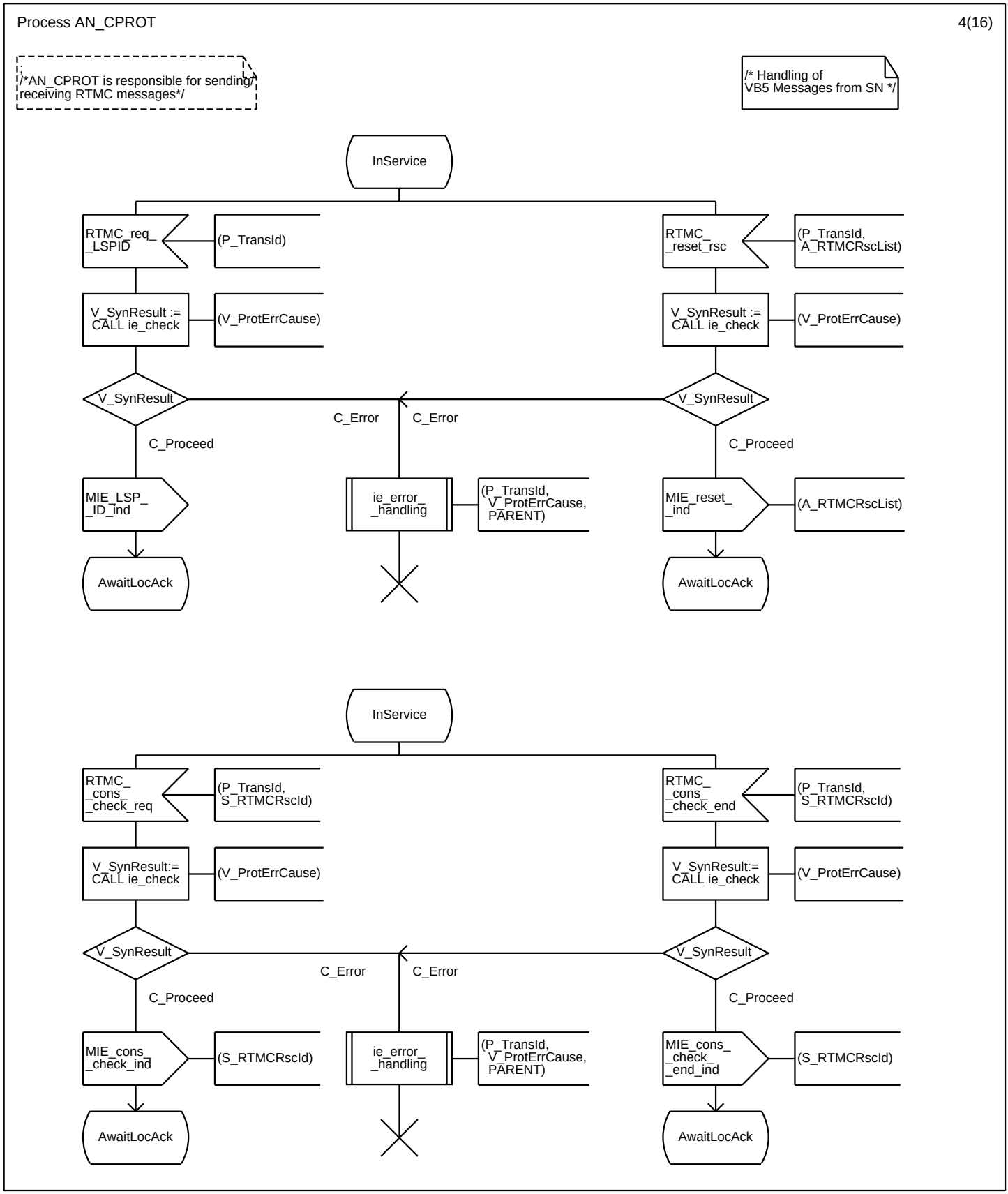
```
/* Timerdefinitions */
TIMER
T_block      := 1   /* Default value for BLOCK_RSC supervision timer is 1 sec, Tolerance +/- 10% */,
T_unblock   := 1   /* Default value for UNBLOCK_RSC supervision timer is 1 sec, Tolerance +/- 10% */,
T_acl         := 1   /* Default value for AWAIT_CLEAR supervision timer is 1 sec, Tolerance +/- 10% */,
T_lspid       := 1   /* Default value for REQ_LSPID supervision timer is 1 sec, Tolerance +/- 10% */,
T_reset      := 60 /* Default value for RESET_RSC supervision timer is 60 sec, Tolerance +/- 10% */;
```

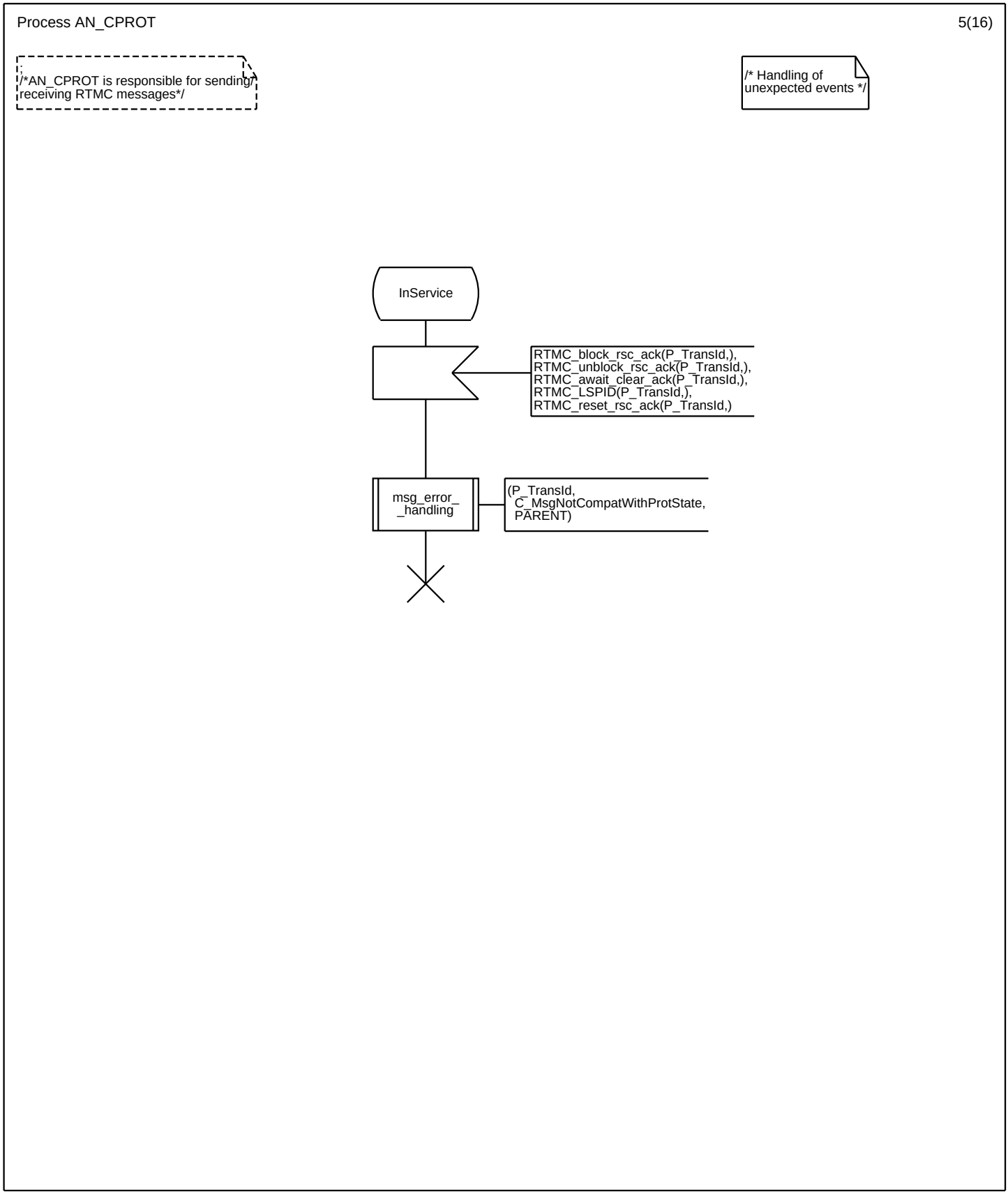
```
/* AN_CPROT internal variables and constants*/
/**/
/* timer handling */
DCL V_ExpCount NATURAL;          /*number of actual message repetitions*/
/*possible values*/
SYNONYM C_ExpMax INTEGER = 1;     /*maximum number of message repetitions is 1*/
/**/
/*syntax check handling*/
DCL V_SynResult INTEGER;          /*syntax check result, provided by procedure IE_CHECK*/
/*possible values*/
SYNONYM C_Proceed INTEGER =1;     /*possible result of syntax check: proceed with message processing*/
SYNONYM C_Error    INTEGER =2;     /*possible result of syntax check: message with error condition */
```

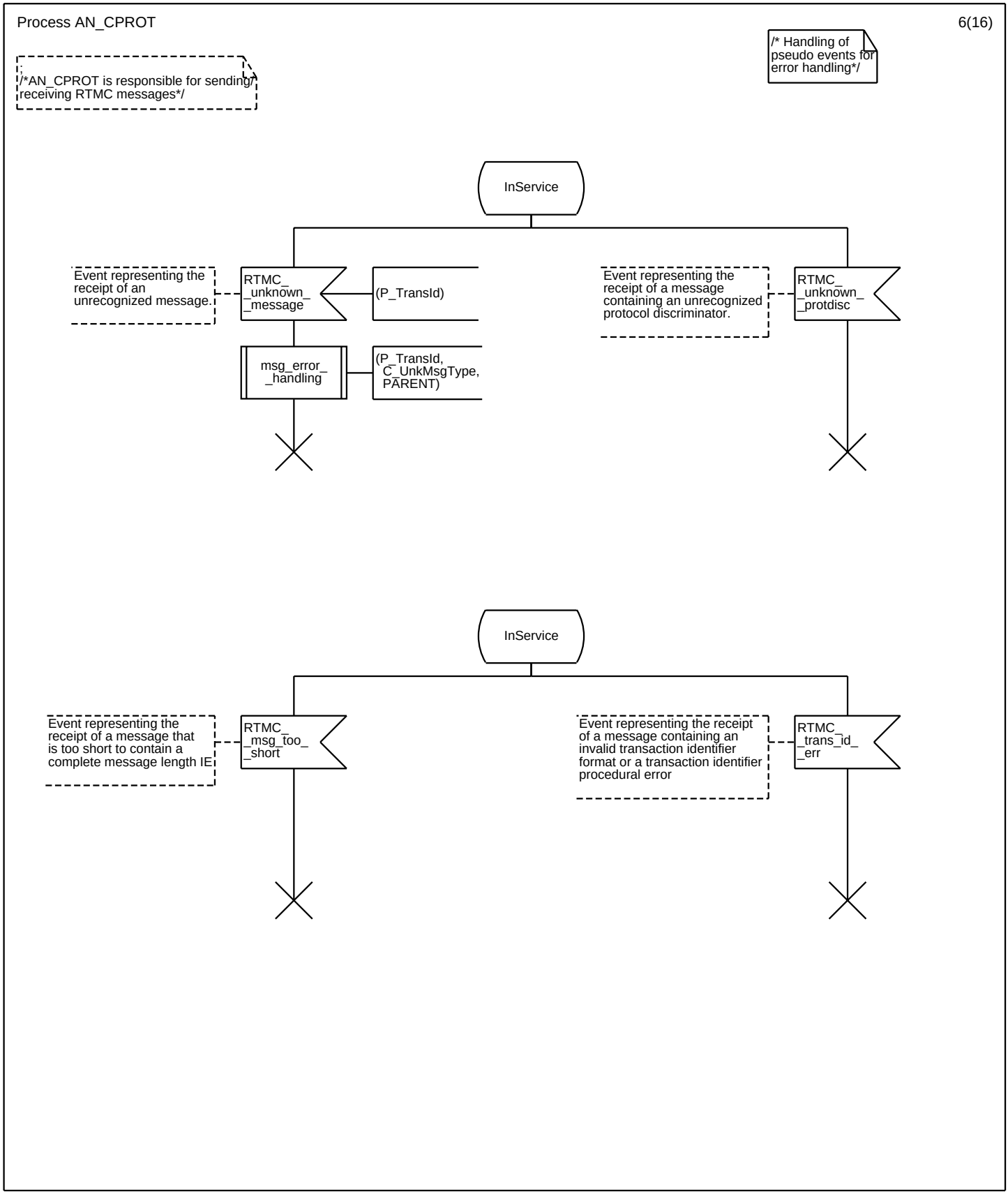
```
/*Signal data declarations*/
DCL
S_RTMCRscId ST_RscId; /*Resource Identifier Information Element*/
/**/
DCL
S_RTMCUnkRscId ST_RscId; /*Resource Identifier Information Element  unknown by the peer side*/
/**/
DCL
A_RTMCRscList AT_RscList;  /*Array of Resource Identifier Information Elements*/
/**/
DCL
A_RTMCBlRscList AT_RTMCBlRscList;  /*Array of Blocked Resource Identifier Information Elements*/
/**/
DCL
A_RTMCUnkRscList AT_RscList;  /*Array of Resource Identifier Information Elements unknown by the peer side*/
/**/
DCL
V_Result             IT_Result,          /* Result to local side */
V_RTMCResult     IT_RTMCResult; /* Result from/to peer side*/
/**/
DCL
V_IntTransId IT_IntTransId; /*identifies the transaction towards AN_STATUS_MGT process*/
/**/
DCL
P_TransId PID; /*transaction identifier towards SN*/
/**/
DCL
V_ProtErrCause   IT_RTMCProtErrCause; /*Error Cause in case of reject*/
```

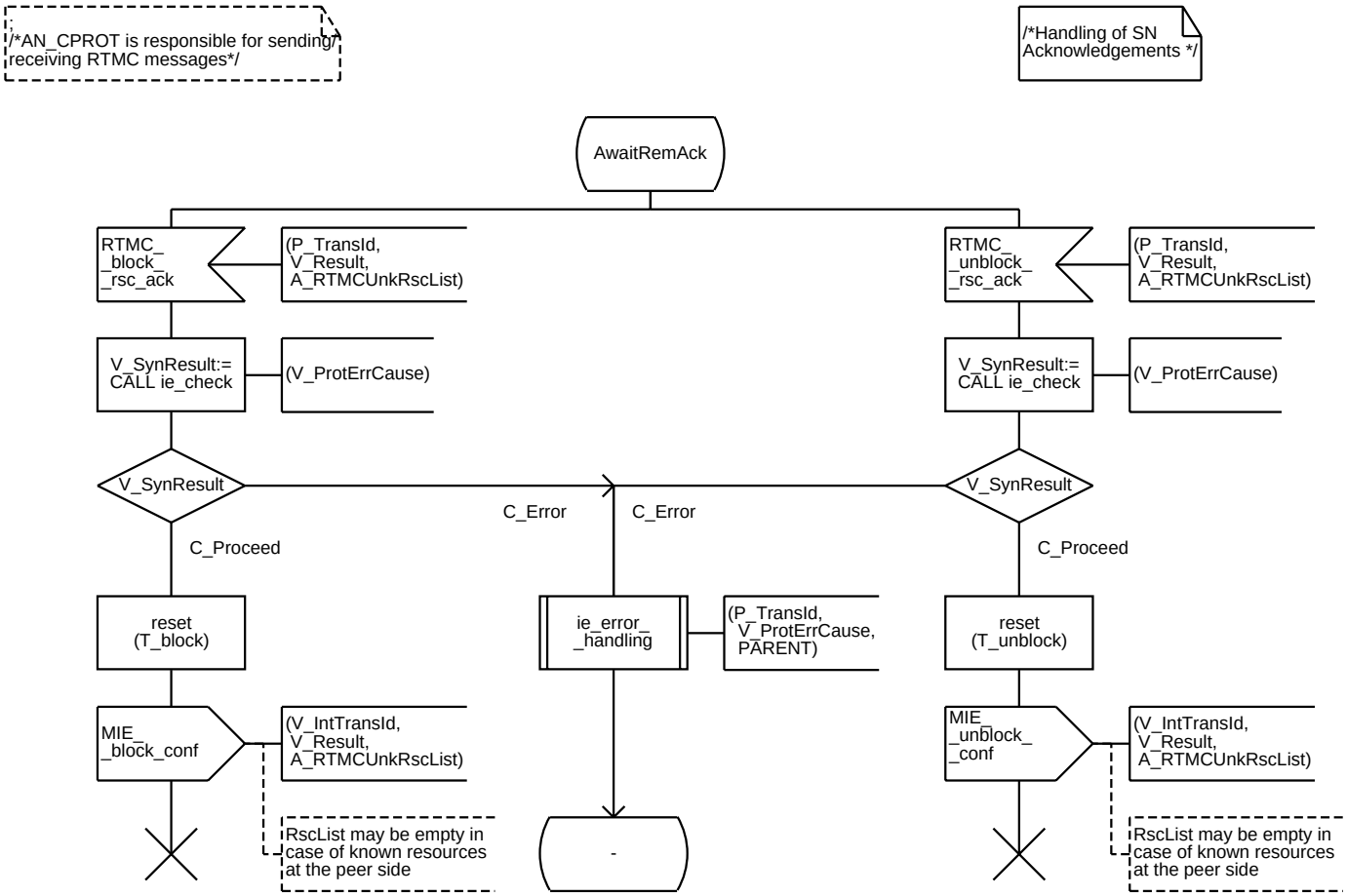

Process AN_CPROT
2(16)
/*Handling of Requests from AN System Management */
/*AN_CPROT is responsible for sending/receiving RTMC messages*/
InService
(V_IntTransId, A_RTMCBlRscList)
MIE_ _block_req
(, A_RTMCBlRscList)
RTMC_block_ _rsc
set (T_block)
V_ExpCount :=0
AwaitRemAck
(V_IntTransId, A_RTMCRscList)
MIE_ _unblock_ _req
(, A_RTMCRscList)
RTMC_ _unblock_rsc
set (T_unblock)
V_ExpCount :=0
AwaitRemAck
(V_IntTransId, A_RTMCRscList)
MIE_ _await_ _clear_req
(, A_RTMCRscList)
RTMC_await_ _clear
set (T_acl)
V_ExpCount :=0
AwaitRemAck
InService
MIE_LSP_ _ID_req
RTMC_req_ _LSPID
TO PARENT
set (T_lspid)
V_ExpCount :=0
AwaitRemAck
(V_IntTransId, A_RTMCRscList)
MIE_ _reset_req
(, A_RTMCRscList) TO PARENT
RTMC_reset_ _rsc
set (T_reset)
V_ExpCount :=0
AwaitRemAck

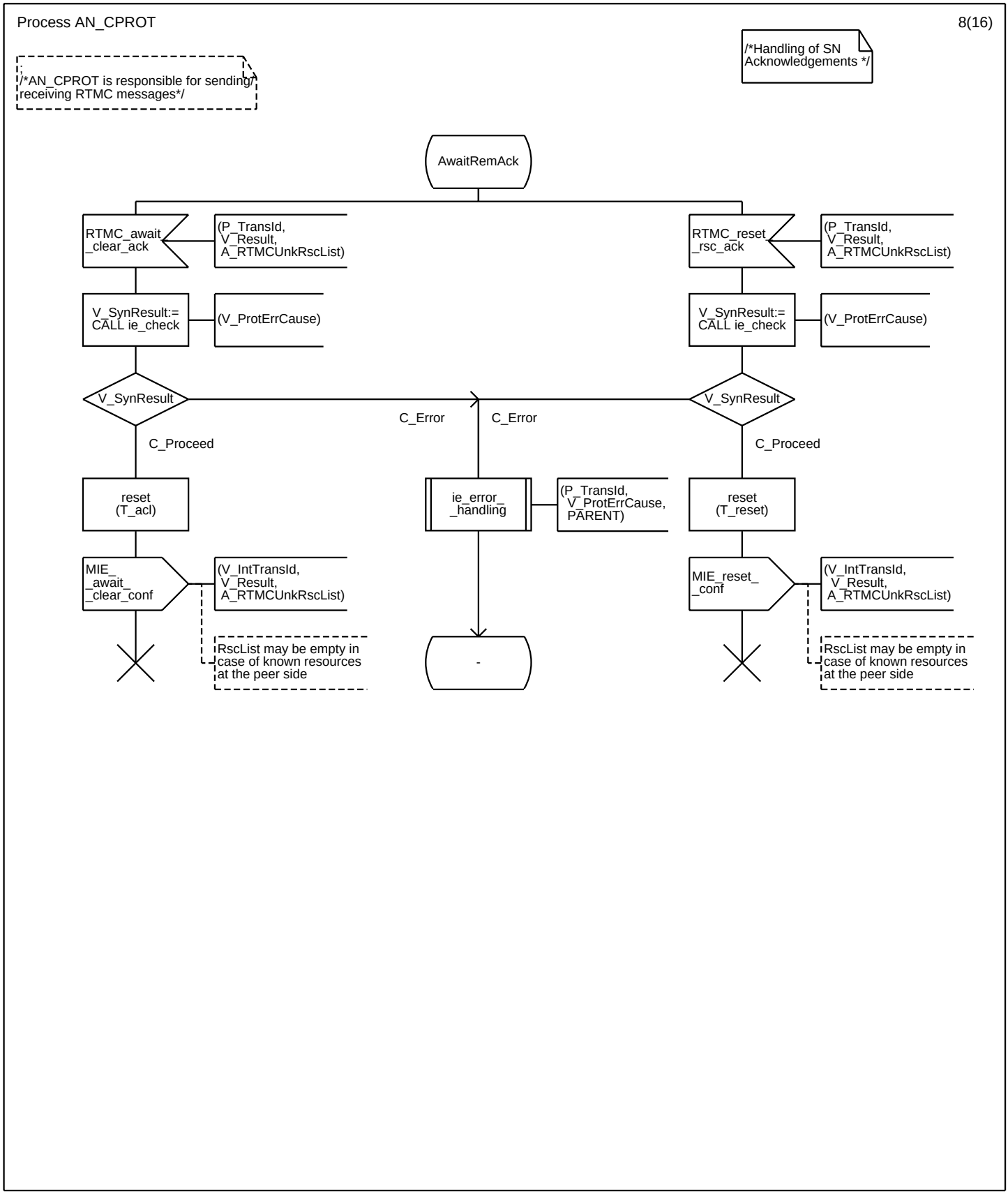

/*AN_CPROT is responsible for sending/receiving RTMC messages*/
/* Handling of VB5 Messages from SN */
InService
RTMC_await_clear_comp
(P_TransId, A_RTMCRscList)
V_SynResult := CALL ie_check
(V_ProtErrCause)
V_SynResult
C_Error
C_Proceed
ie_error_handling
(P_TransId, V_ProtErrCause, PARENT)
MIE_await_clear_conf
(C_UndefIntTransId, C_Success, A_RTMCRscList)
RTMC_await_clear_comp_ack
(P_TransId)

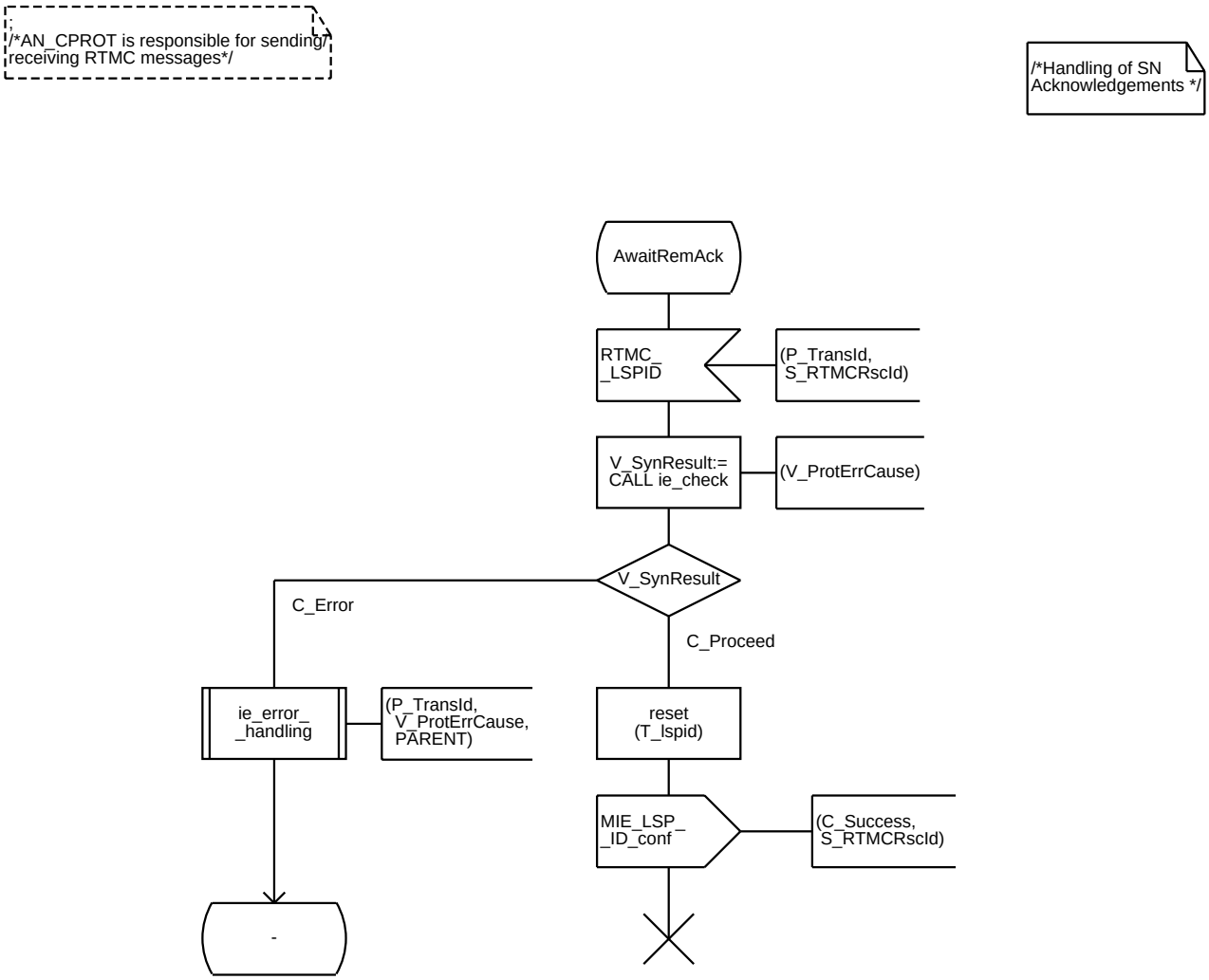

Process AN_CPROT
4(16)
/*AN_CPROT is responsible for sending/receiving RTMC messages*/
/* Handling of VB5 Messages from SN */
InService
RTMC_req_ _LSPID
(P_TransId)
V_SynResult := CALL ie_check
(V_ProtErrCause)
V_SynResult
C_Proceed
MIE_LSP_ _ID_ind
AwaitLocAck
C_Error
C_Error
ie_error_ _handling
(P_TransId, V_ProtErrCause, PARENT)
RTMC_ _reset_rsc
(P_TransId, A_RTMCRscList)
V_SynResult := CALL ie_check
(V_ProtErrCause)
V_SynResult
C_Proceed
MIE_reset_ _ind
(A_RTMCRscList)
AwaitLocAck
InService
RTMC_ _cons_ _check_req
(P_TransId, S_RTMCRscId)
V_SynResult:= CALL ie_check
(V_ProtErrCause)
V_SynResult
C_Proceed
MIE_cons_ _check_ind
(S_RTMCRscId)
AwaitLocAck
C_Error
C_Error
ie_error_ _handling
(P_TransId, V_ProtErrCause, PARENT)
RTMC_ _cons_ _check_end
(P_TransId, S_RTMCRscId)
V_SynResult:= CALL ie_check
(V_ProtErrCause)
V_SynResult
C_Proceed
MIE_cons_ _check_ _end_ind
(S_RTMCRscId)
AwaitLocAck

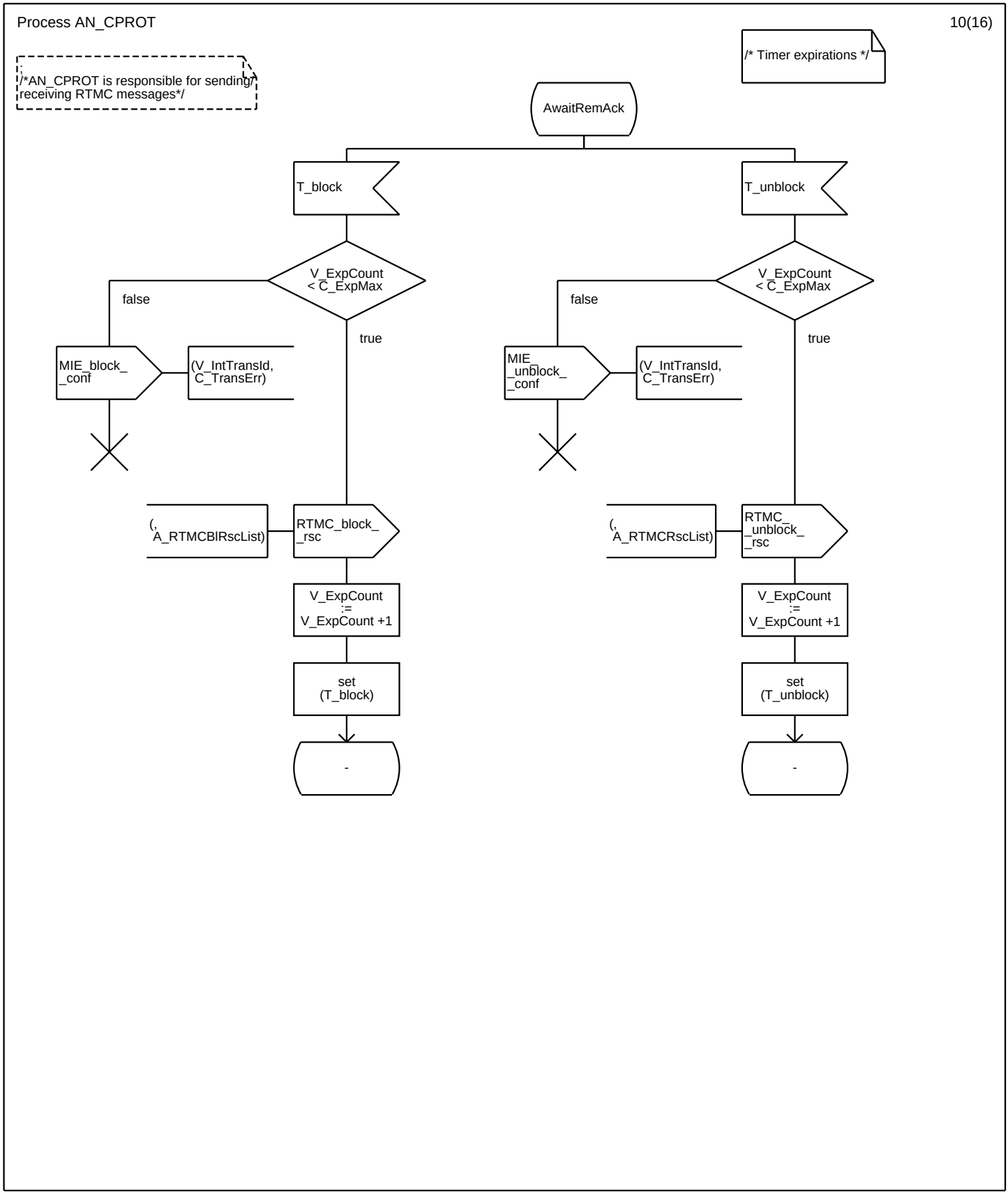

/*AN_CPROT is responsible for sending/receiving RTMC messages*/
/* Handling of unexpected events */
InService
RTMC_block_rsc_ack(P_TransId,),
RTMC_unblock_rsc_ack(P_TransId,),
RTMC_await_clear_ack(P_TransId,),
RTMC_LSPID(P_TransId,),
RTMC_reset_rsc_ack(P_TransId,)
msg_error_
_handling
(P_TransId,
C_MsgNotCompatWithProtState,
PARENT)

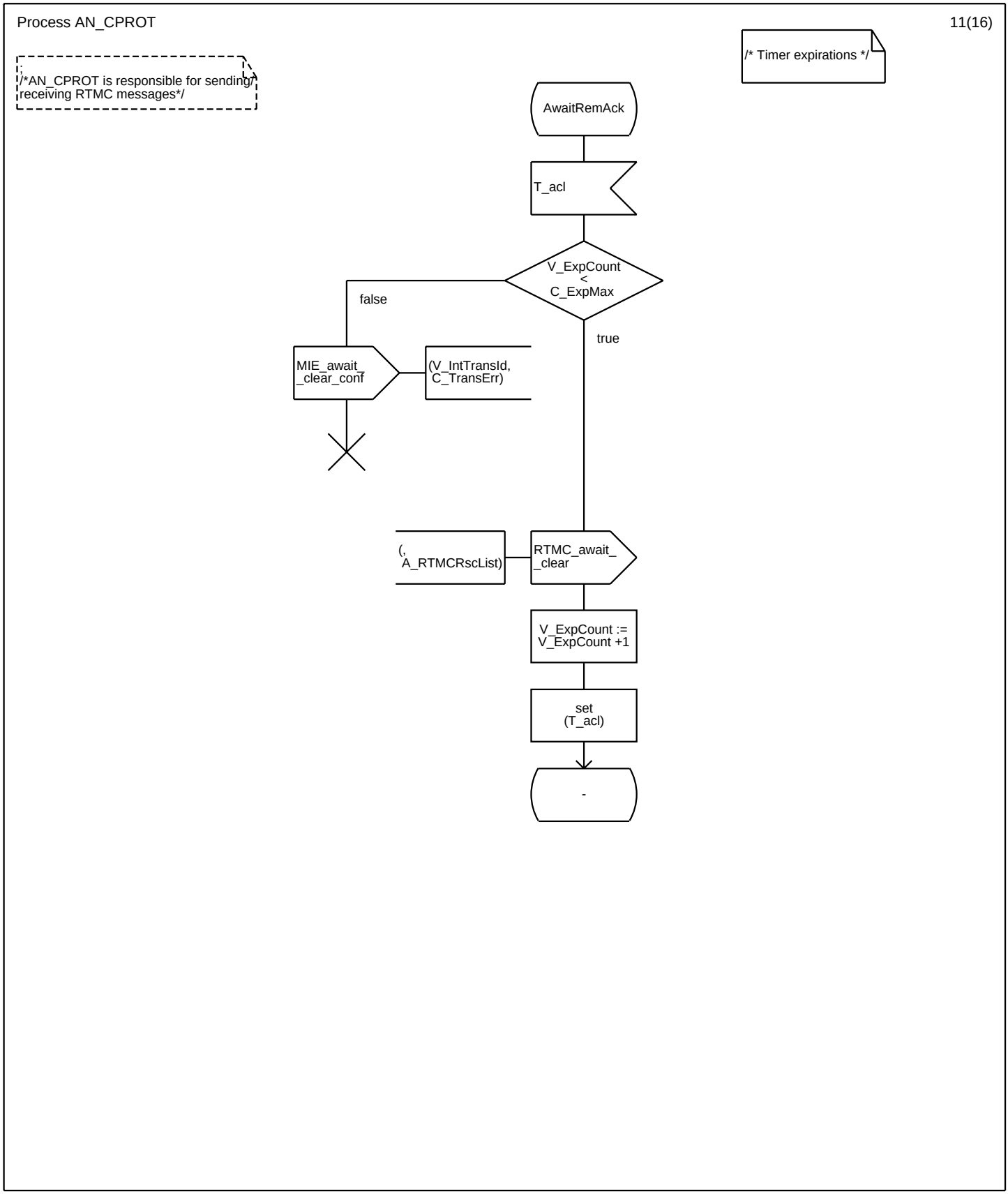

Process AN_CPROT
6(16)
/*AN_CPROT is responsible for sending/receiving RTMC messages*/
/* Handling of pseudo events for error handling*/
InService
Event representing the receipt of an unrecognized message.
RTMC_ _unknown_ _message
(P_TransId)
msg_error_ _handling
(P_TransId, C_UnkMsgType, PARENT)
Event representing the receipt of a message containing an unrecognized protocol discriminator.
RTMC_ _unknown_ _protdisc
InService
Event representing the receipt of a message that is too short to contain a complete message length IE
RTMC_ _msg_too_ _short
Event representing the receipt of a message containing an invalid transaction identifier format or a transaction identifier procedural error
RTMC_ _trans_id_ _err

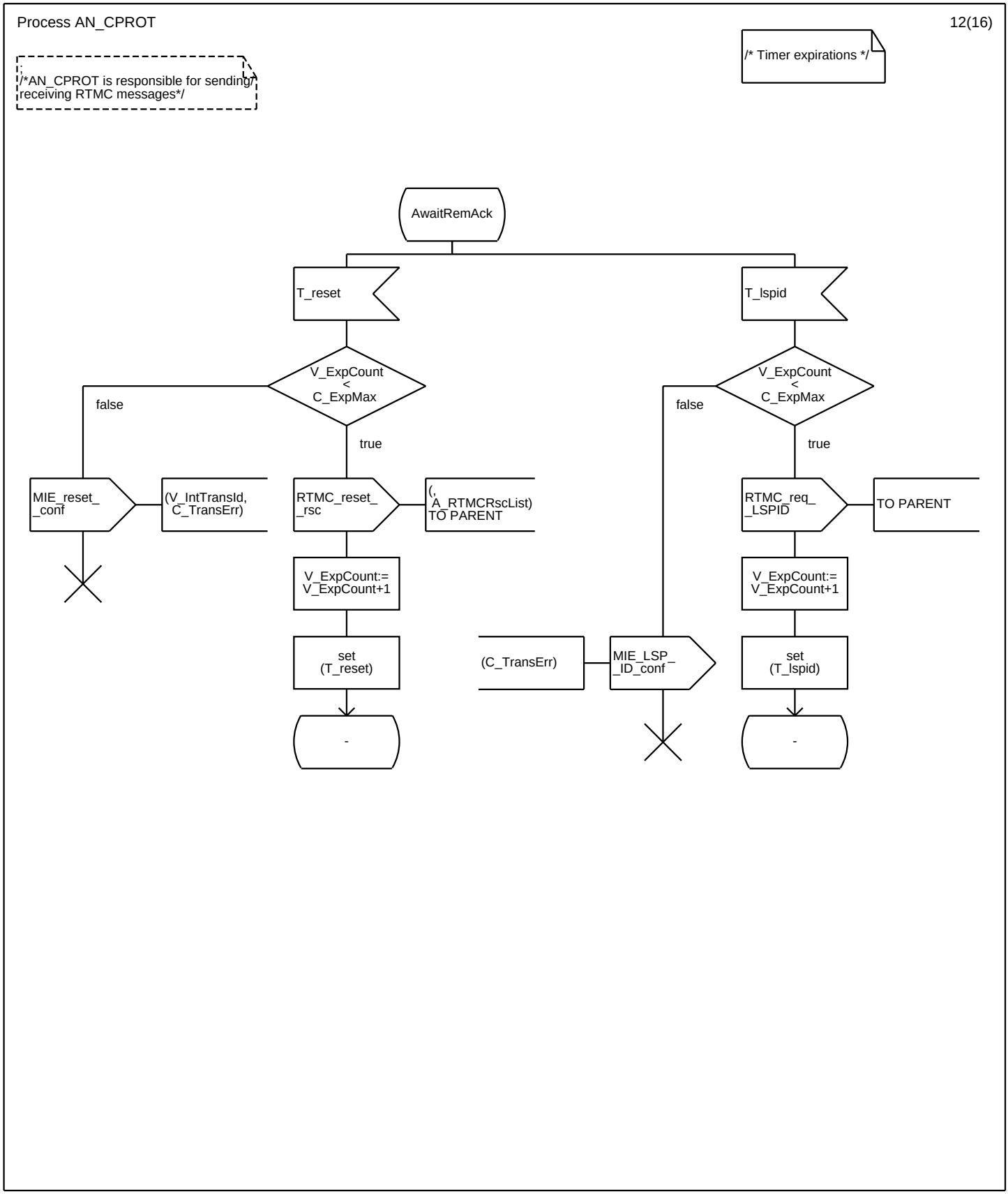

/*AN_CPROT is responsible for sending/receiving RTMC messages*/
/*Handling of SN Acknowledgements */
AwaitRemAck
RTMC_ _block_ _rsc_ack
(P_TransId, V_Result, A_RTMCUnkRscList)
V_SynResult:= CALL ie_check
(V_ProtErrCause)
V_SynResult
C_Error
C_Error
C_Proceed
reset (T_block)
ie_error_ _handling
(P_TransId, V_ProtErrCause, PARENT)
MIE_ _block_conf
(V_IntTransId, V_Result, A_RTMCUnkRscList)
RscList may be empty in case of known resources at the peer side
-
RTMC_ _unblock_ _rsc_ack
(P_TransId, V_Result, A_RTMCUnkRscList)
V_SynResult:= CALL ie_check
(V_ProtErrCause)
V_SynResult
C_Proceed
reset (T_unblock)
MIE_ _unblock_ _conf
(V_IntTransId, V_Result, A_RTMCUnkRscList)
RscList may be empty in case of known resources at the peer side

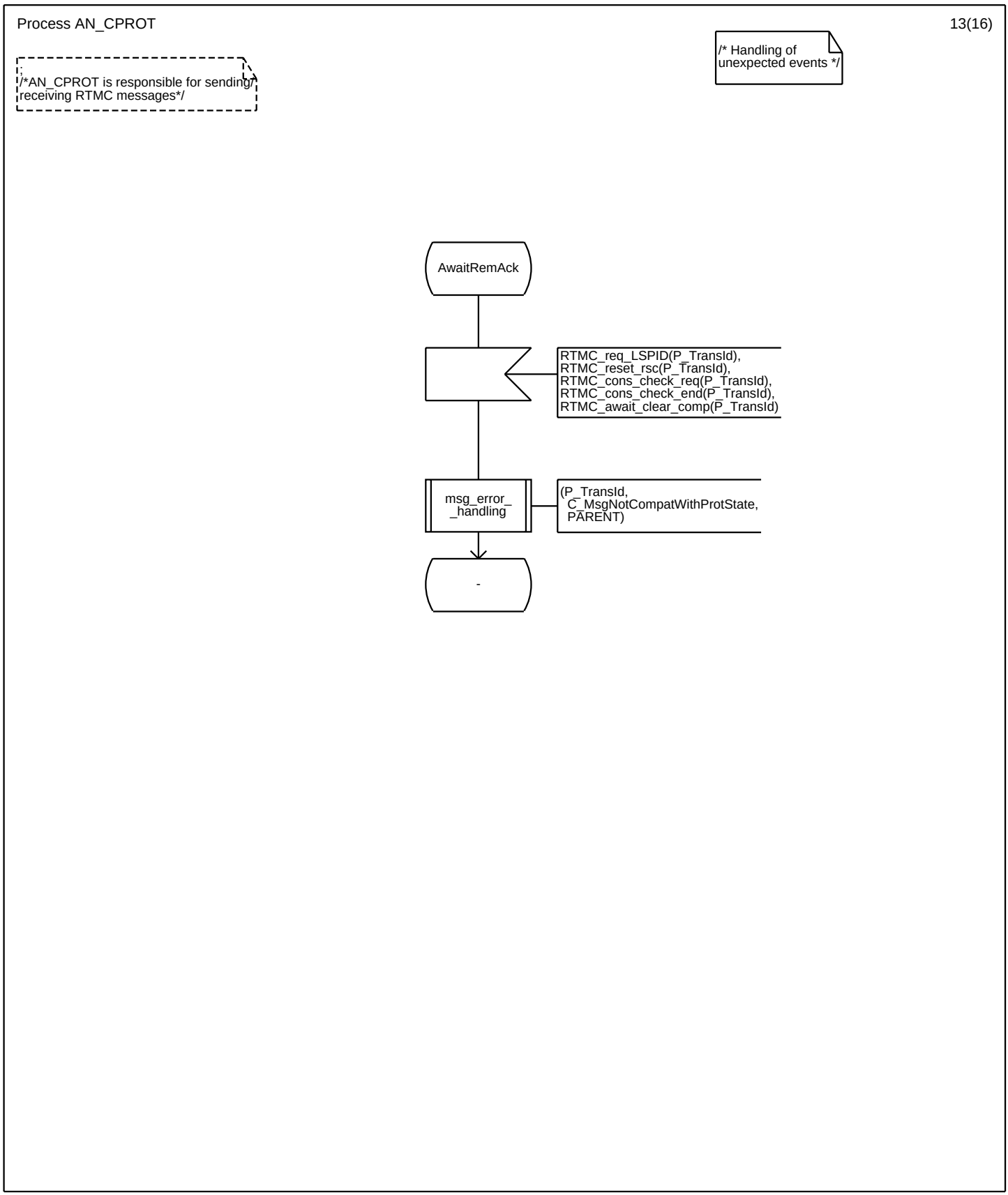

Process AN_CPROT
8(16)
/*Handling of SN Acknowledgements */
/*AN_CPROT is responsible for sending/receiving RTMC messages*/
AwaitRemAck
RTMC_await _clear_ack
(P_TransId, V_Result, A_RTMCUnkRscList)
RTMC_reset _rsc_ack
(P_TransId, V_Result, A_RTMCUnkRscList)
V_SynResult:= CALL ie_check
(V_ProtErrCause)
V_SynResult:= CALL ie_check
(V_ProtErrCause)
V_SynResult
C_Error
C_Error
V_SynResult
C_Proceed
C_Proceed
reset (T_acl)
ie_error_ _handling
(P_TransId, V_ProtErrCause, PARENT)
reset (T_reset)
MIE_ _await_ _clear_conf
(V_IntTransId, V_Result, A_RTMCUnkRscList)
MIE_reset_ _conf
(V_IntTransId, V_Result, A_RTMCUnkRscList)
RscList may be empty in case of known resources at the peer side
-
RscList may be empty in case of known resources at the peer side

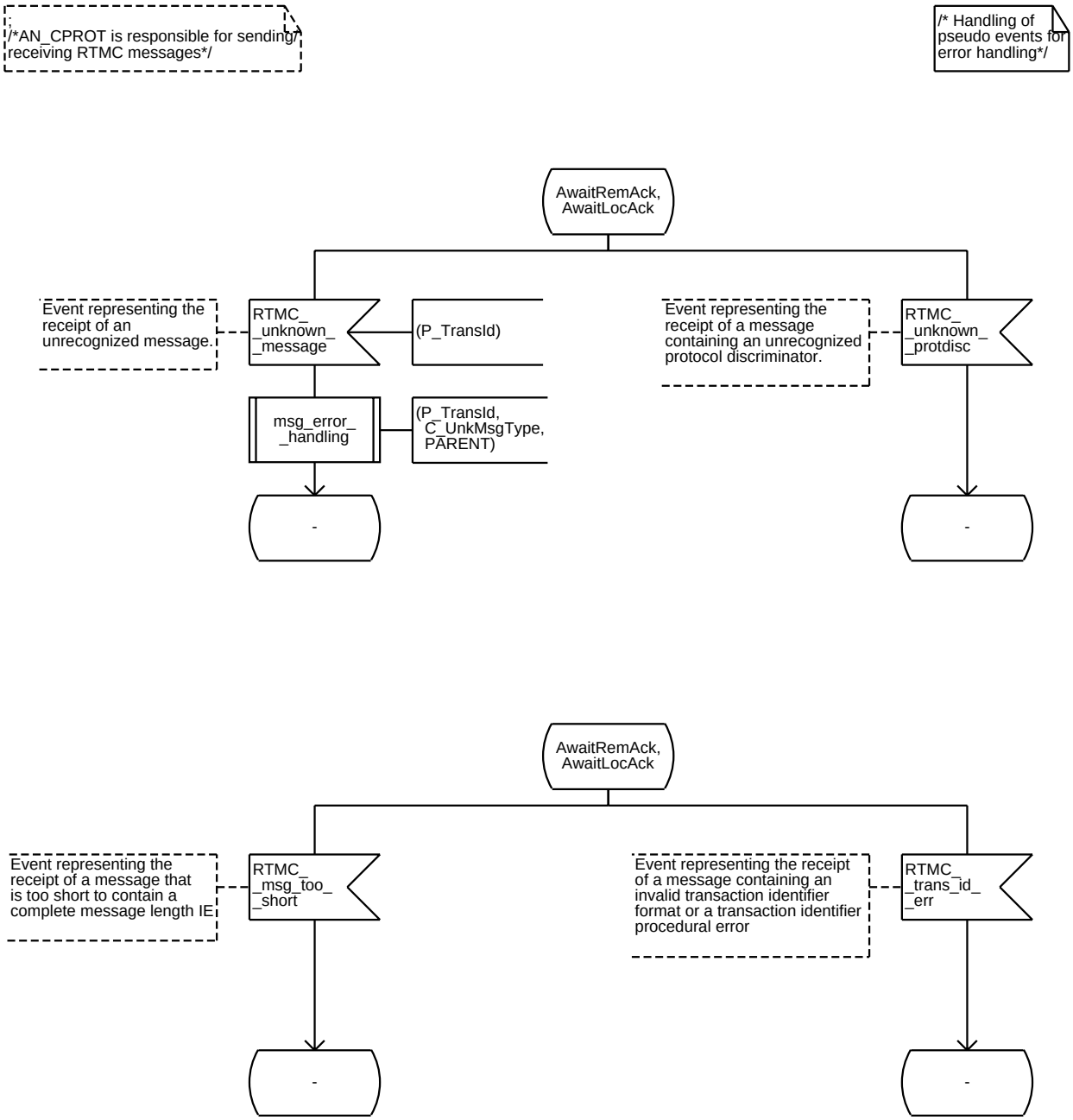

;
/*AN_CPROT is responsible for sending/
receiving RTMC messages*/
/*Handling of SN
Acknowledgements */
AwaitRemAck
RTMC_
_LSPID
(P_TransId,
S_RTMCRscId)
V_SynResult:=
CALL ie_check
(V_ProtErrCause)
V_SynResult
C_Error
C_Proceed
ie_error_
_handling
(P_TransId,
V_ProtErrCause,
PARENT)
reset
(T_lspid)
MIE_LSP_
_ID_conf
(C_Success,
S_RTMCRscId)
-

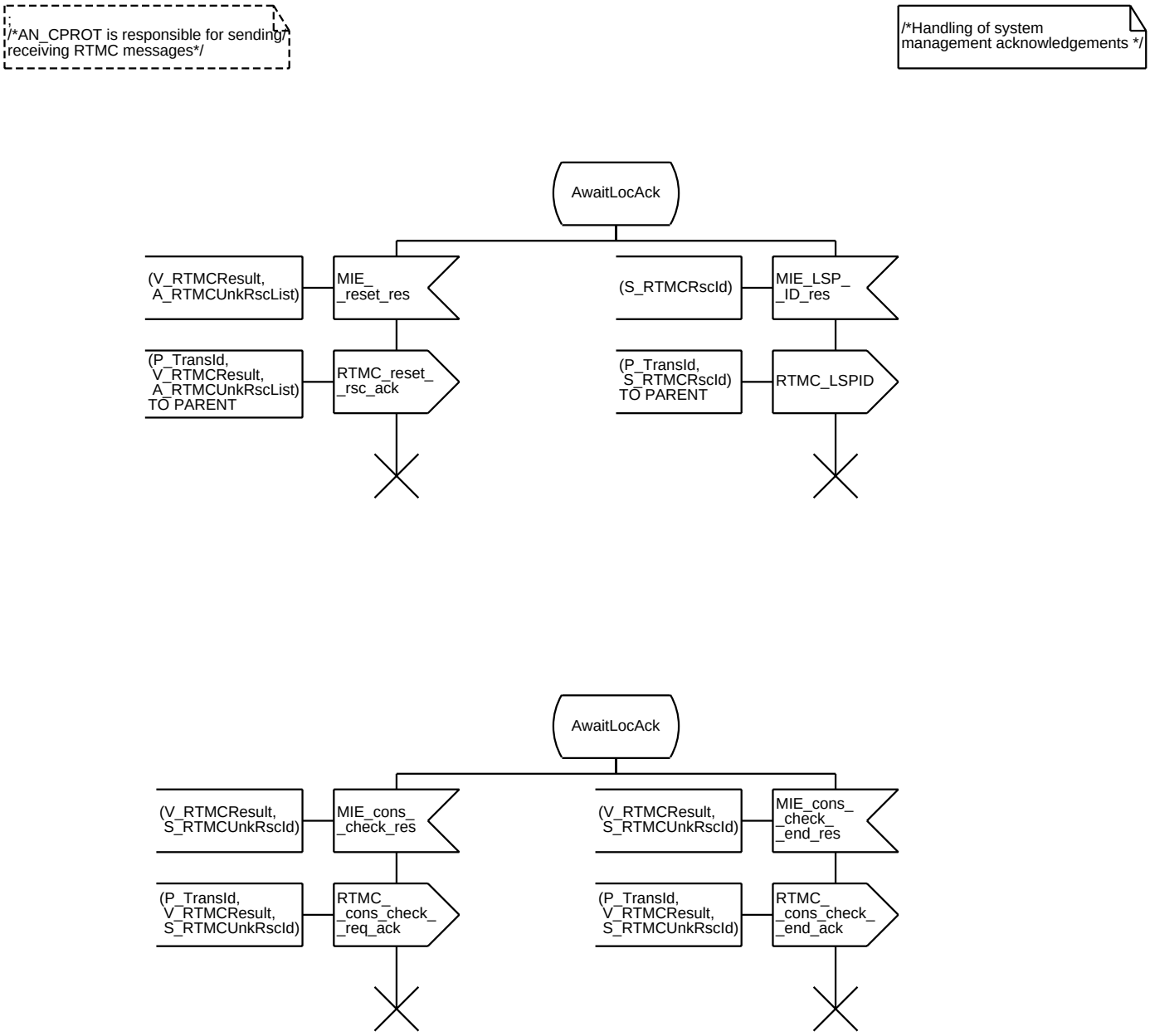

Process AN_CPROT
10(16)
/* Timer expirations */
/*AN_CPROT is responsible for sending/receiving RTMC messages*/
AwaitRemAck
T_block
T_unblock
V_ExpCount < C_ExpMax
false
true
MIE_block_ _conf
(V_IntTransId, C_TransErr)
(, A_RTMCBlRscList)
RTMC_block_ _rsc
V_ExpCount := V_ExpCount +1
set (T_block)
-
V_ExpCount < C_ExpMax
false
true
MIE_ _unblock_ _conf
(V_IntTransId, C_TransErr)
(, A_RTMCRscList)
RTMC_ _unblock_ _rsc
V_ExpCount := V_ExpCount +1
set (T_unblock)
-

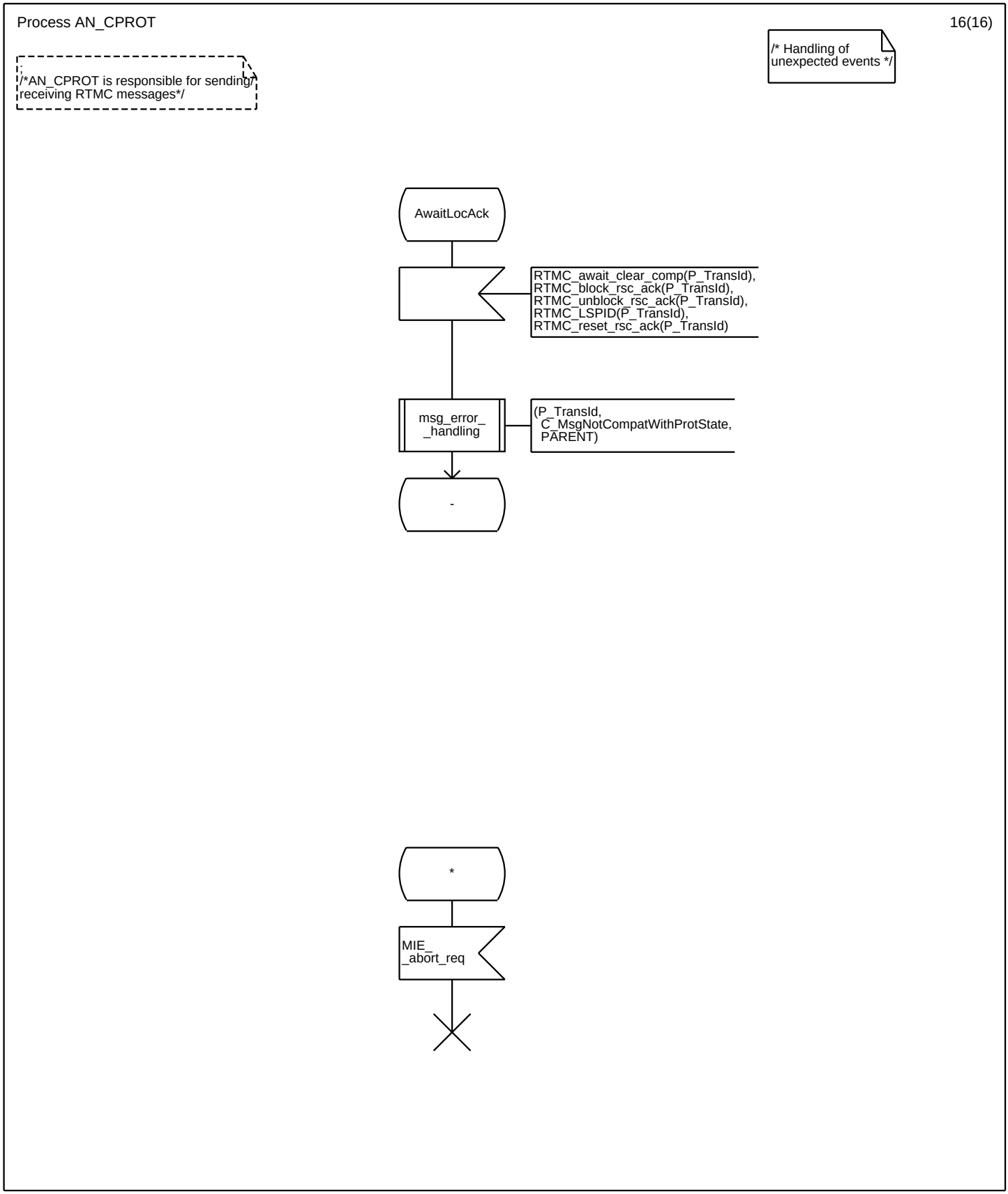

Process AN_CPROT
11(16)
/* Timer expirations */
/*AN_CPROT is responsible for sending/
receiving RTMC messages*/
AwaitRemAck
T_acl
V_ExpCount
<
C_ExpMax
false
true
MIE_await_
_clear_conf
(V_IntTransId,
C_TransErr)
(,
A_RTMCRscList)
RTMC_await_
_clear
V_ExpCount :=
V_ExpCount +1
set
(T_acl)
-

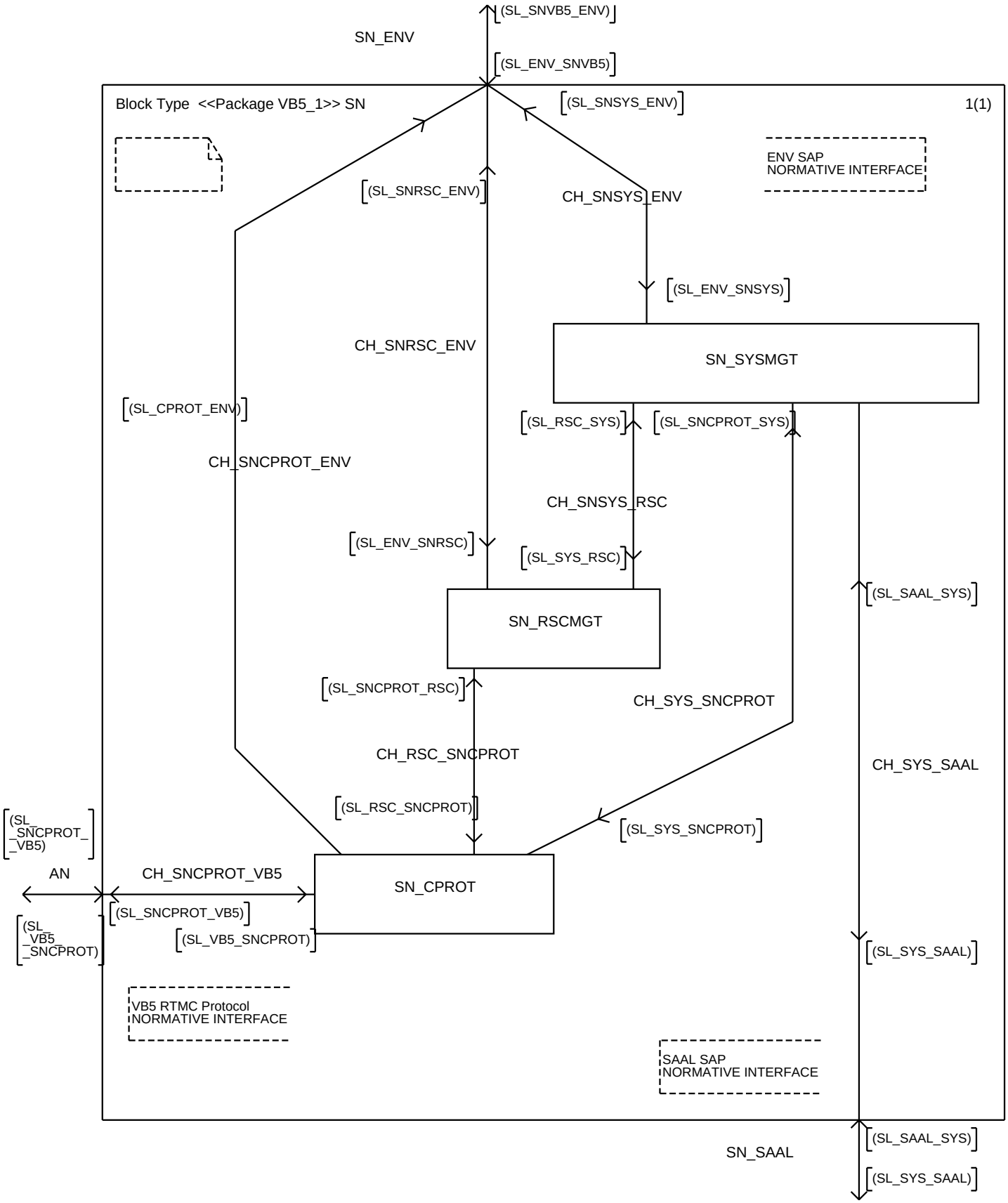

Process AN_CPROT
12(16)
/* Timer expirations */
/*AN_CPROT is responsible for sending/receiving RTMC messages*/
AwaitRemAck
T_reset
T_lspid
V_ExpCount < C_ExpMax
false
true
MIE_reset_ _conf
(V_IntTransId, C_TransErr)
RTMC_reset_ _rsc
(, A_RTMCRscList) TO PARENT
V_ExpCount:= V_ExpCount+1
set (T_reset)
-
(C_TransErr)
MIE_LSP_ _ID_conf
RTMC_req_ _LSPID
TO PARENT
set (T_lspid)

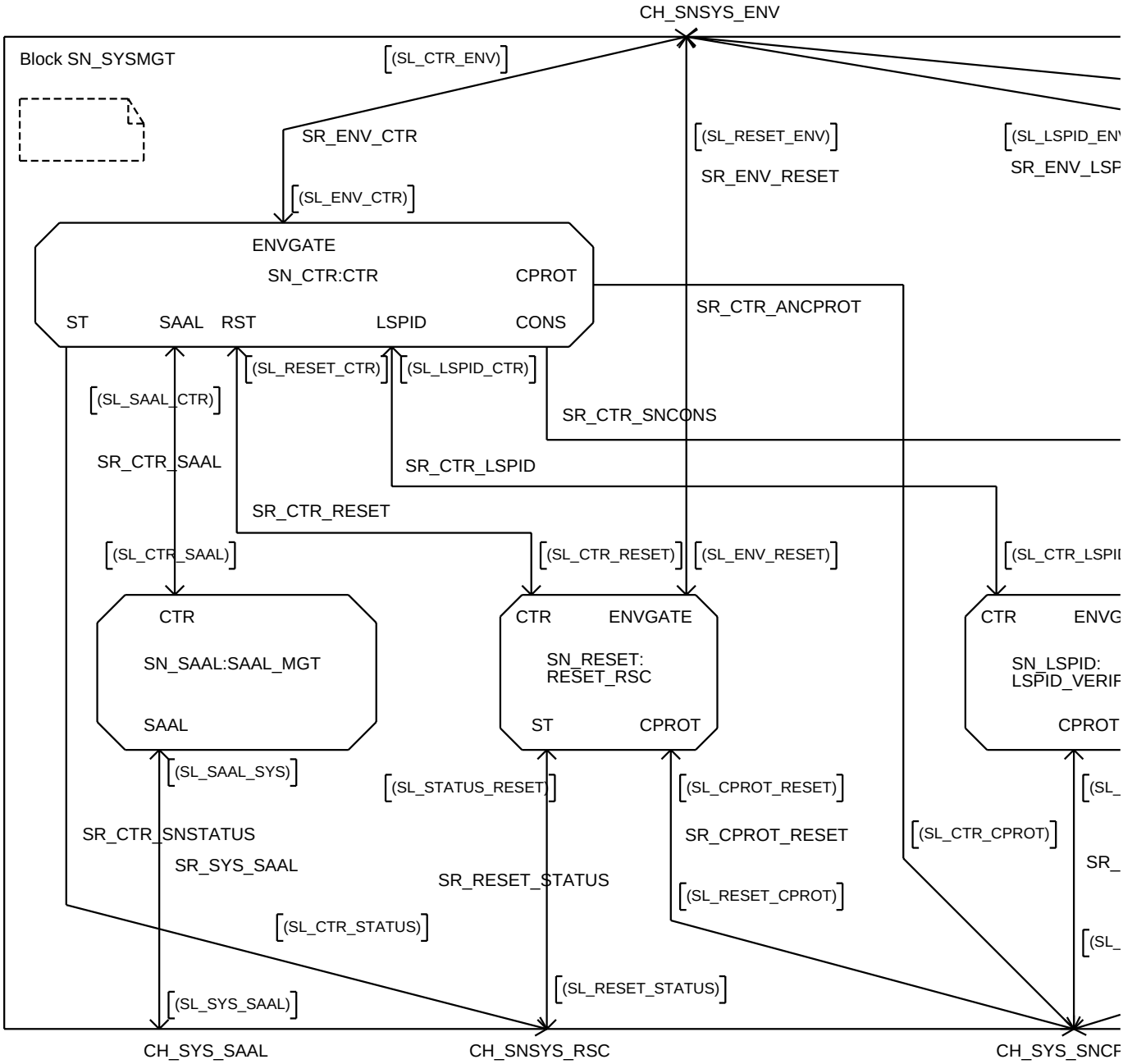

/*AN_CPROT is responsible for sending/receiving RTMC messages*/
/* Handling of unexpected events */
AwaitRemAck
RTMC_req_LSPID(P_TransId),
RTMC_reset_rsc(P_TransId),
RTMC_cons_check_req(P_TransId),
RTMC_cons_check_end(P_TransId),
RTMC_await_clear_comp(P_TransId)
msg_error_
_handling
(P_TransId,
C_MsgNotCompatWithProtState,
PARENT)
-

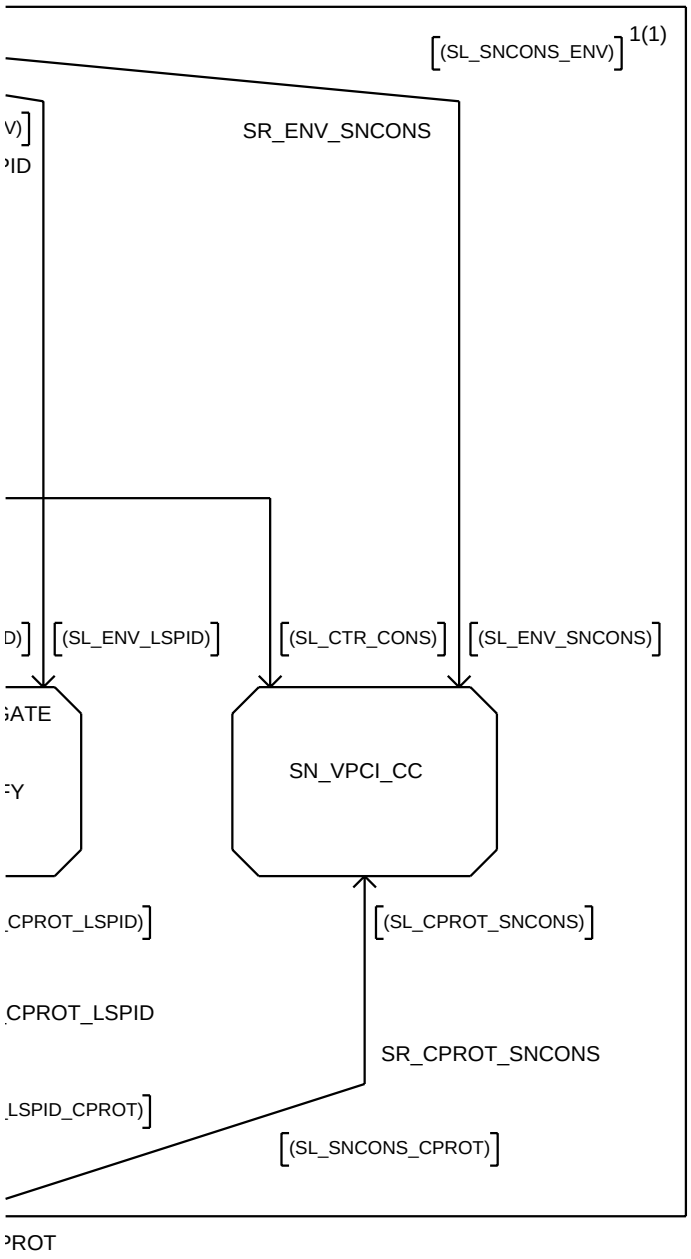

;
/*AN_CPROT is responsible for sending/receiving RTMC messages*/
/* Handling of pseudo events for error handling*/
AwaitRemAck, AwaitLocAck
Event representing the receipt of an unrecognized message.
RTMC_ _unknown_ _message
(P_TransId)
msg_error_ _handling
(P_TransId, C_UnkMsgType, PARENT)
-
Event representing the receipt of a message containing an unrecognized protocol discriminator.
RTMC_ _unknown_ _protdisc
-
AwaitRemAck, AwaitLocAck
Event representing the receipt of a message that is too short to contain a complete message length IE
RTMC_ _msg_too_ _short
-
Event representing the receipt of a message containing an invalid transaction identifier format or a transaction identifier procedural error
RTMC_ _trans_id_ _err
-


/*AN_CPROT is responsible for sending/receiving RTMC messages*/
/*Handling of system management acknowledgements */
AwaitLocAck
(V_RTMCResult, A_RTMCUnkRscList)
MIE_ _reset_res
(S_RTMCRscId)
MIE_LSP_ _ID_res
(P_TransId, V_RTMCResult, A_RTMCUnkRscList) TO PARENT
RTMC_reset_ _rsc_ack
(P_TransId, S_RTMCRscId) TO PARENT
RTMC_LSPID
AwaitLocAck
(V_RTMCResult, S_RTMCUnkRscId)
MIE_cons_ _check_res
(V_RTMCResult, S_RTMCUnkRscId)
MIE_cons_ _check_ _end_res
(P_TransId, V_RTMCResult, S_RTMCUnkRscId)
RTMC_ _cons_check_ _req_ack
(P_TransId, V_RTMCResult, S_RTMCUnkRscId)
RTMC_ _cons_check_ _end_ack


/*AN_CPROT is responsible for sending/receiving RTMC messages*/
/* Handling of unexpected events */
AwaitLocAck
RTMC_await_clear_comp(P_TransId),
RTMC_block_rsc_ack(P_TransId),
RTMC_unblock_rsc_ack(P_TransId),
RTMC_LSPID(P_TransId),
RTMC_reset_rsc_ack(P_TransId)
msg_error_
_handling
(P_TransId,
C_MsgNotCompatWithProtState,
PARENT)
-
*
MIE_
_abort_req


(SL_SNVB5_ENV)
SN_ENV
(SL_ENV_SNVB5)
Block Type <<Package VB5_1>> SN
(SL_SNSYS_ENV)
1(1)
ENV SAP
NORMATIVE INTERFACE
(SL_SNRSC_ENV)
CH_SNSYS_ENV
(SL_ENV_SNSYS)
CH_SNRSC_ENV
SN_SYSMGT
(SL_CPROT_ENV)
(SL_RSC_SYS)
(SL_SNCPROT_SYS)
CH_SNCPROT_ENV
CH_SNSYS_RSC
(SL_ENV_SNRSC)
(SL_SYS_RSC)
(SL_SAAL_SYS)
SN_RSCMGT
(SL_SNCPROT_RSC)
CH_SYS_SNCPROT
CH_RSC_SNCPROT
CH_SYS_SAAL
(SL_RSC_SNCPROT)
(SL_SYS_SNCPROT)
(SL_
SNCPROT
_VB5)
AN
CH_SNCPROT_VB5
SN_CPROT
(SL_SNCPROT_VB5)
(SL_
_VB5_
_SNCPROT)
(SL_VB5_SNCPROT)
(SL_SYS_SAAL)
VB5 RTMC Protocol
NORMATIVE INTERFACE
SAAL SAP
NORMATIVE INTERFACE
(SL_SAAL_SYS)
SN_SAAL
(SL_SYS_SAAL)


CH_SNSYS_ENV
Block SN_SYSMGT
(SL_CTR_ENV)
SR_ENV_CTR
(SL_ENV_CTR)
(SL_RESET_ENV)
SR_ENV_RESET
ENVGATE
SN_CTR:CTR
CPROT
ST
SAAL
RST
LSPID
CONS
SR_CTR_ANCPROT
(SL_RESET_CTR)
(SL_LSPID_CTR)
(SL_SAAL_CTR)
SR_CTR_SNCONS
SR_CTR_SAAL
SR_CTR_LSPID
SR_CTR_RESET
(SL_CTR_SAAL)
(SL_CTR_RESET)
(SL_ENV_RESET)
CTR
SN_SAAL:SAAL_MGT
SAAL
CTR
ENVGATE
SN_RESET:
RESET_RSC
ST
CPROT
(SL_SAAL_SYS)
(SL_STATUS_RESET)
(SL_CPROT_RESET)
SR_CTR_SNSTATUS
SR_CPROT_RESET
(SL_CTR_CPROT)
SR_SYS_SAAL
SR_RESET_STATUS
(SL_RESET_CPROT)
(SL_CTR_STATUS)
(SL_RESET_STATUS)
(SL_SYS_SAAL)
CH_SYS_SAAL
CH_SNSYS_RSC


1(1)
(SL_SNCONS_ENV)
SR_ENV_SNCONS
(SL_ENV_LSPID)
(SL_CTR_CONS)
(SL_ENV_SNCONS)
SN_VPCI_CC
(SL_CPROT_SNCONS)
SR_CPROT_SNCONS
(SL_SNCONS_CPROT)

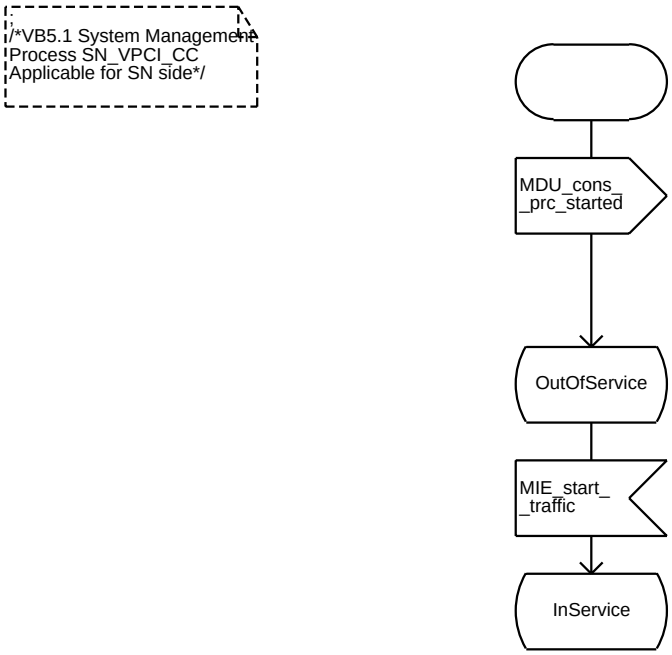
Process SN_VPCI_CC 1(4)

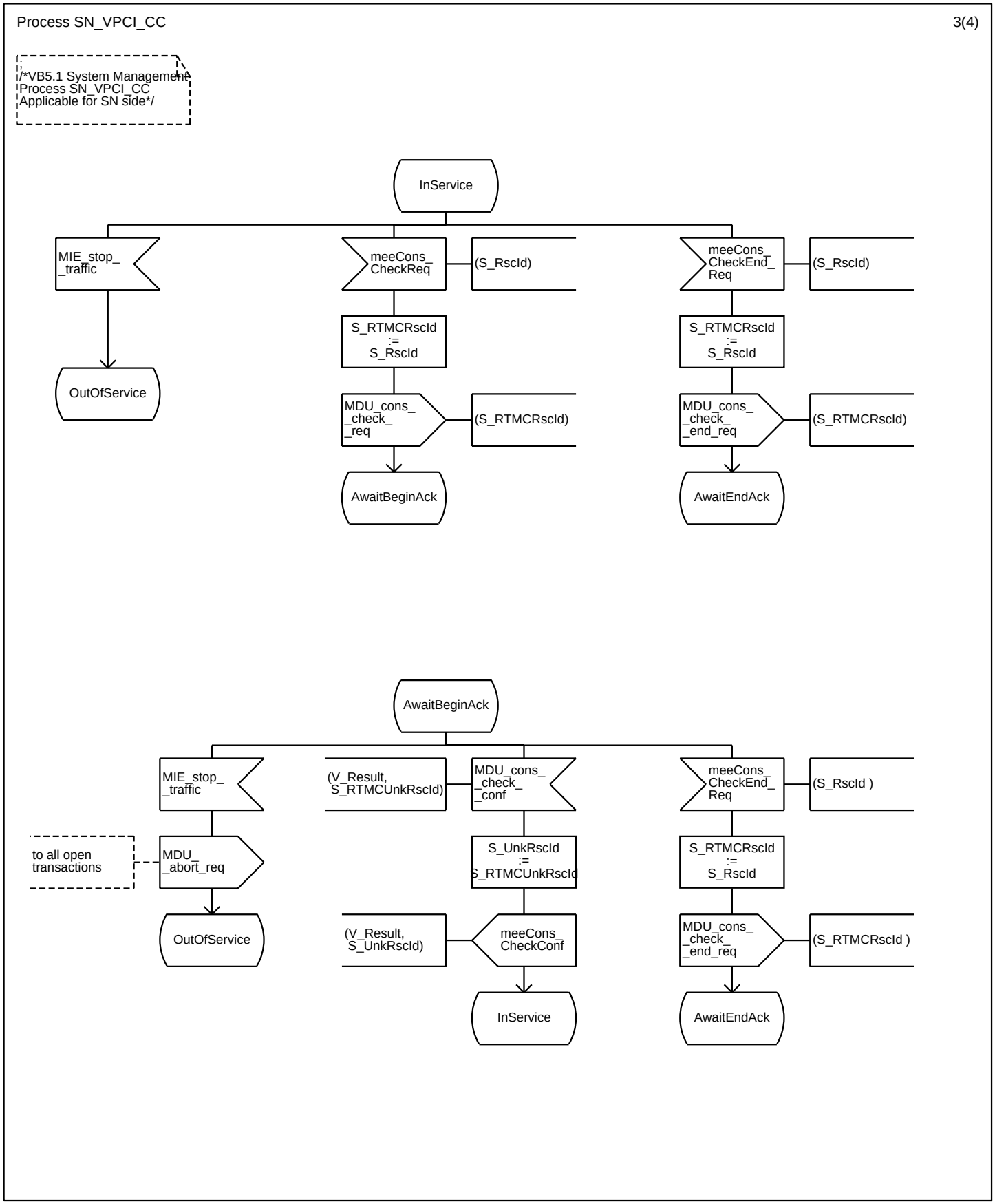
```
/*VB5.1 System Management
Process SN_VPCI_CC
Applicable for SN side*/
```

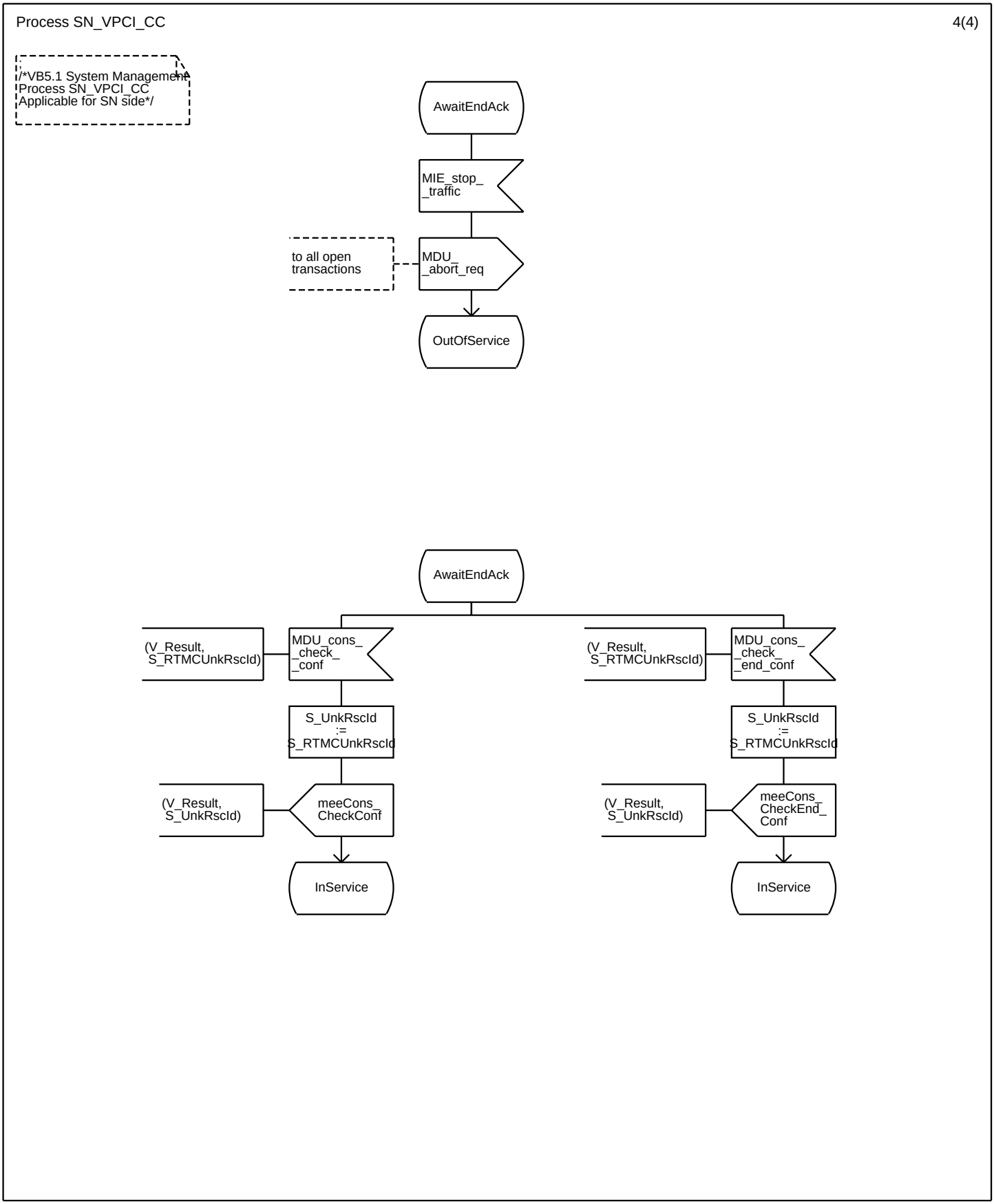
```
/* Definitions and declarations
for process SN_VPCI_CC */
```

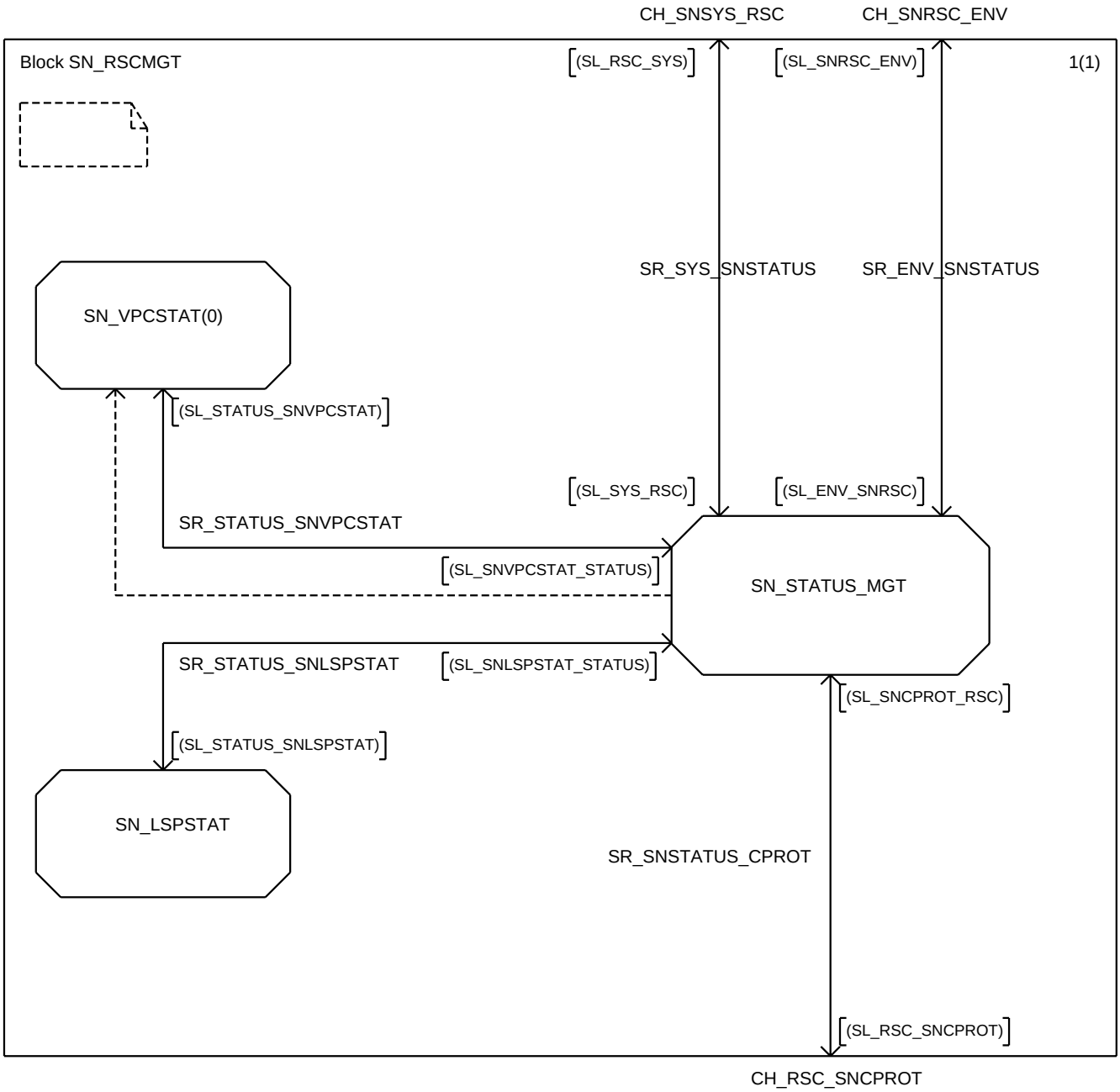
```
/*State descriptions
OutOfService:    Initial state. Process is waiting for activation with start traffic.
InService:       Process is waiting for primitive from environment
AwaitBeginAck:   Process is waiting for confirmation of check begin from AN side.
AwaitEndAck:     Process is waiting for confirmation of check end from AN side. */
```

```
/*Signal data declarations*/
DCL
S_RscId          ST_RscId,          /*Resource Identifier reported from/to local side*/
S_RTMCRscId ST_RscId;          /*Resource Identifier reported from/to peer side*/
/**/
DCL
S_UnkRscId          ST_RscId,     /* Unknown Resource Identifier reported from/to local side*/
S_RTMCUnkRscId ST_RscId;     /* Unknown Resource Identifier reported from/to peer side*/
/**/
DCL
V_Result     IT_Result;       /*Result*/
```


/*VB5.1 System Management
Process SN_VPCI_CC
Applicable for SN side*/
MDU_cons_
_prc_started
OutOfService
MIE_start_
_traffic
InService


Process SN_VPCI_CC
3(4)
/*VB5.1 System Management
Process SN_VPCI_CC
Applicable for SN side*/
InService
MIE_stop_
_traffic
meeCons_
CheckReq
(S_RscId)
meeCons_
CheckEnd_
Req
(S_RscId)
S_RTMCRscId
:=
S_RscId
S_RTMCRscId
:=
S_RscId
OutOfService
MDU_cons_
check
_req
(S_RTMCRscId)
MDU_cons_
check
_end_req
(S_RTMCRscId)
AwaitBeginAck
AwaitEndAck
AwaitBeginAck
MIE_stop_
_traffic
(V_Result,
S_RTMCUnkRscId)
MDU_cons_
check
_conf
meeCons_
CheckEnd_
Req
(S_RscId)
to all open
transactions
MDU_
_abort_req
S_UnkRscId
:=
S_RTMCUnkRscId
S_RTMCRscId
:=
S_RscId
OutOfService
(V_Result,
S_UnkRscId)
meeCons_
CheckConf
MDU_cons_
check
_end_req
(S_RTMCRscId)
InService
AwaitEndAck


Process SN_VPCI_CC
4(4)
/*VB5.1 System Management
Process SN_VPCI_CC
Applicable for SN side*/
AwaitEndAck
MIE_stop_
_traffic
to all open
transactions
MDU_
_abort_req
OutOfService
AwaitEndAck
(V_Result,
S_RTMCUnkRscId)
MDU_cons_
check
_conf
S_UnkRscId
:=
S_RTMCUnkRscId
(V_Result,
S_UnkRscId)
meeCons_
CheckConf
InService
(V_Result,
S_RTMCUnkRscId)
MDU_cons_
check
_end_conf
S_UnkRscId
:=
S_RTMCUnkRscId
(V_Result,
S_UnkRscId)
meeCons_
CheckEnd_
Conf
InService


CH_SNSYS_RSC
CH_SNRSC_ENV
Block SN_RSCMGT
(SL_RSC_SYS)
(SL_SNRSC_ENV)
1(1)
SR_SYS_SNSTATUS
SR_ENV_SNSTATUS
SN_VPCSTAT(0)
(SL_STATUS_SNVPCSTAT)
(SL_SYS_RSC)
(SL_ENV_SNRSC)
SR_STATUS_SNVPCSTAT
(SL_SNVPCSTAT_STATUS)
SN_STATUS_MGT
SR_STATUS_SNLSPSTAT
(SL_SNLSPSTAT_STATUS)
(SL_SNCPROT_RSC)
(SL_STATUS_SNLSPSTAT)
SN_LSPSTAT
SR_SNSTATUS_CPROT
(SL_RSC_SNCPROT)
CH_RSC_SNCPROT

Process SN_STATUS_MGT 1(15)

```
/*SN_STATUS_MGT controls
state changes of VPCs and LSP
reported by the AN*/
```

```
/* Definintions and declarations for
SN_STATUS_MGT*/
```

```
/* State descriptions

OutOfService:          Init State of SN_SYSMGT_STATUS, process is waiting for startup of the interface
UblDuringStrtup:       Interface startup was initiated and unblocking of all VPCFSMs and LSPFSM is running
AwaitStartTraffic:     Unblocking during startup completed, start traffic expected
InService:             Normal operation, no status change was requested
AwaitResetFSMAck:      Reset is running
AwaitUblFSMAck:        Status change was requested, unblocking of single STATFSMs is running
AwaitBlFSMAck:         Status change was requested, blocking of single STATFSMs is running
AwaitAclFSMAck:        Status change was requested, shutdown of single STATFSMs is running
*/
```

```
/*SN_STATUS_MGT internal data type definitions*/
/**/
/*Typedefinition for storing SN_VPCSTAT process addresses*/
NEWTYPE ST_VPCPID STRUCT
   S_RscId      ST_RscId;
   P_VPCPID  PID;
ENDNEWTYPE ST_VPCPID;
NEWTYPE AT_VPCPIDList ARRAY
   (INTEGER, ST_VPCPID)
ENDNEWTYPE AT_VPCPIDList;
```

```
Procedure for decoding
blocked resource identifiers
```

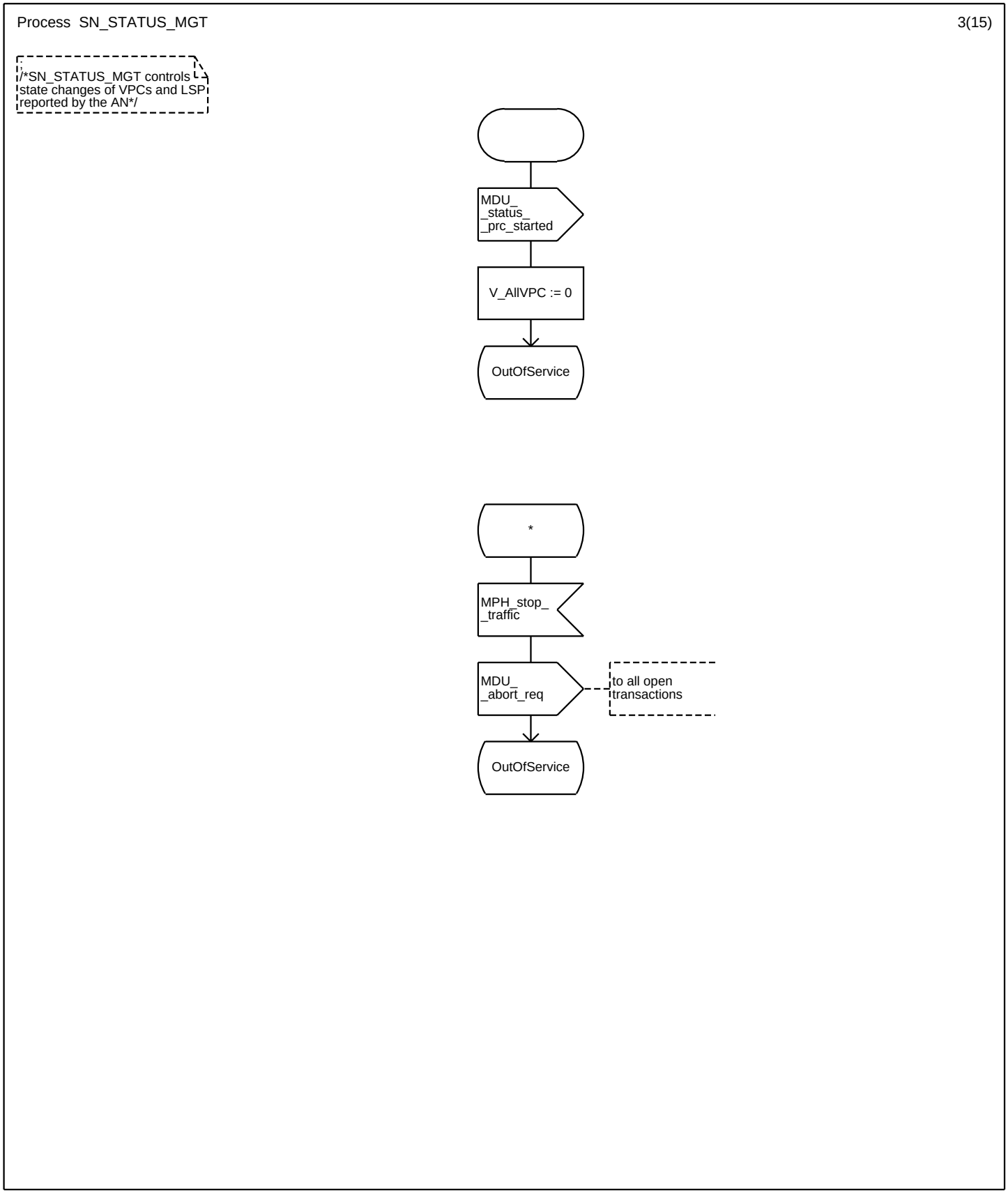
decode_block_ind

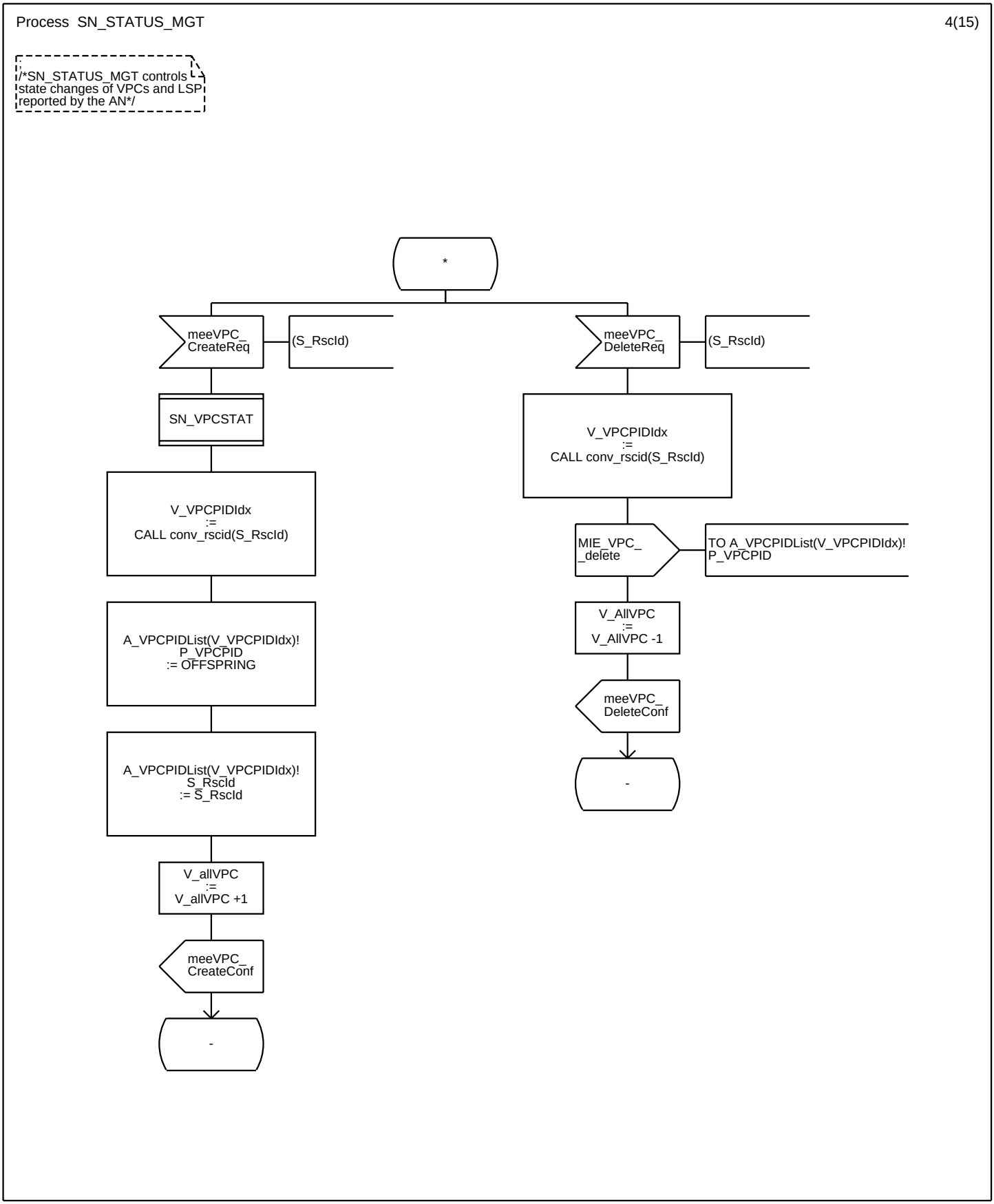
```
/*SN_STATUS_MGT controls
state changes of VPCs and LSP
reported by the AN*/
```

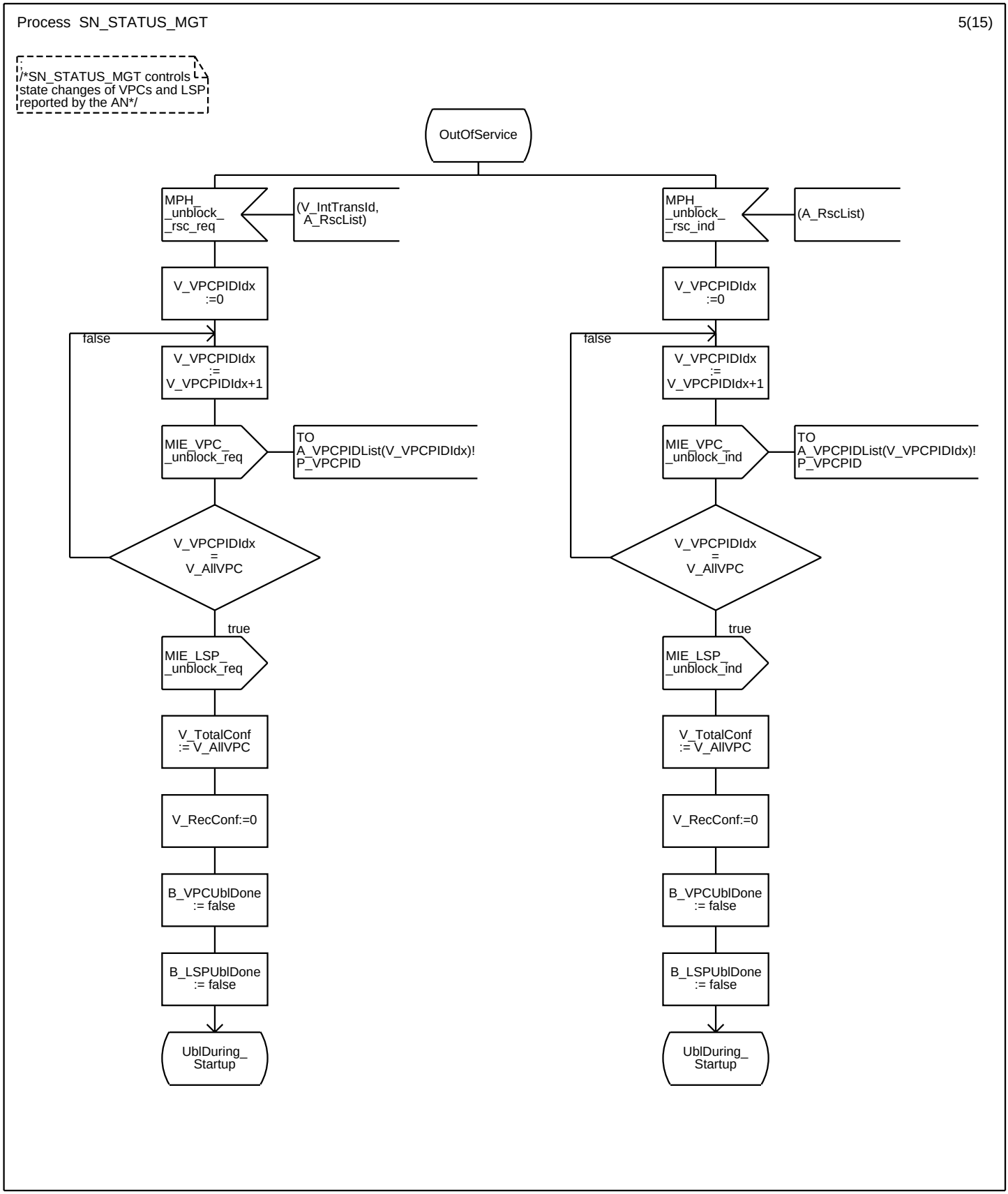
```
/* Definintions and declarations for
SN_STATUS_MGT*/
```

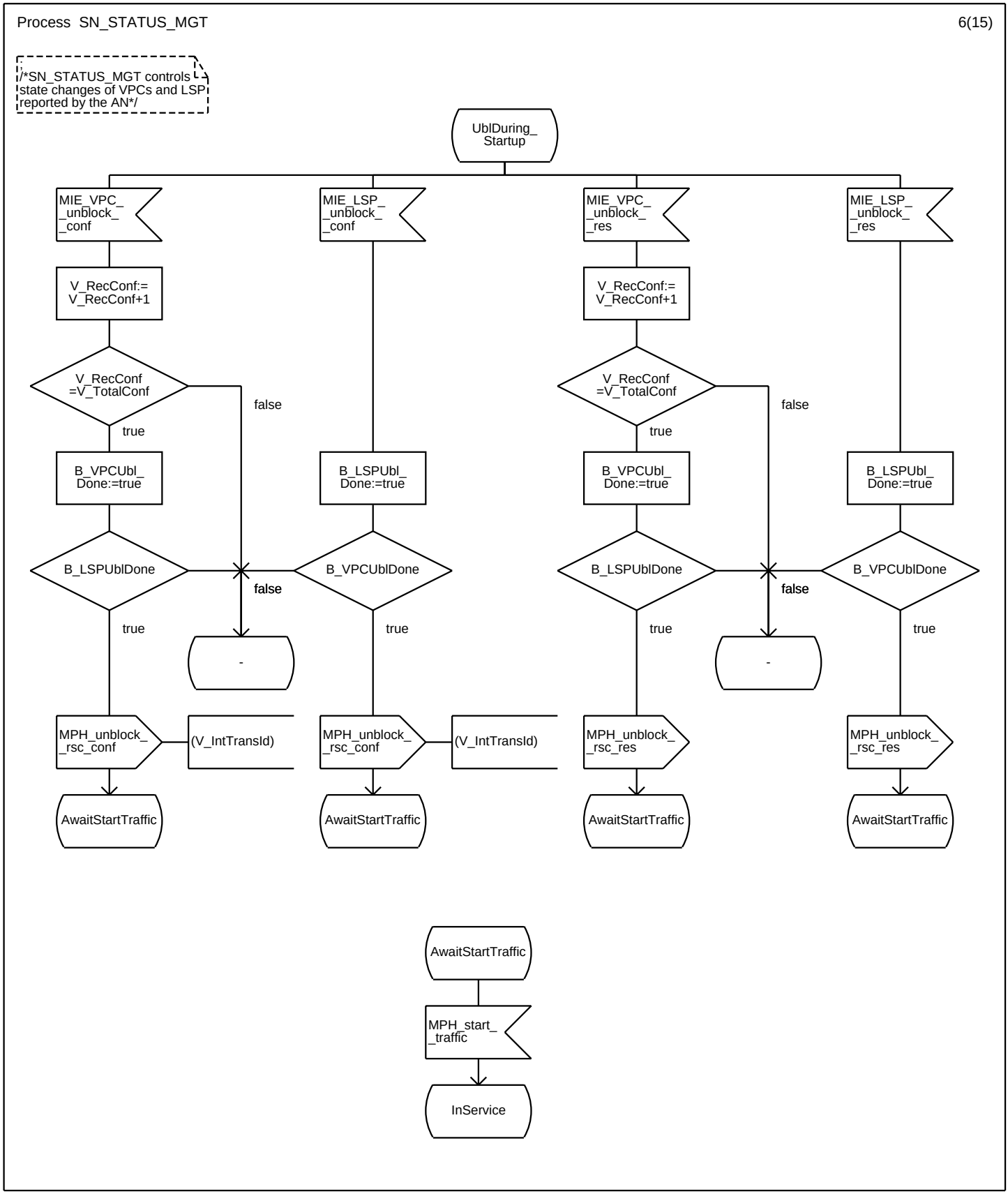
```
/*Signal data declarations*/
/**/
DCL
S_RscId            ST_RscId; /*Resource identifier*/
/**/
DCL
A_RscList           AT_RscList,  /*Resource Identifier List reported from/to local side*/
A_RTMCRscList AT_RscList;  /*Resource identifier List reported from/to peer side*/
/**/
DCL
A_RTMCBlRscList AT_RTMCBlRscList;  /*Array of Blocked Resource Identifier Information Elements*/
/**/
DCL
A_RTMCUnkRscList AT_RscList;  /*Array of Unknown Resource Identifier Information Elements*/
/**/
DCL
V_Reason IT_Reason;  /*Status change reason given to VPCSTAT/LSPSTAT process*/
/**/
DCL
V_IntTransId IT_IntTransId; /*Transaction identifier for communication with AN_CPROT process*/
/**/
DCL
V_NoOfRsc INTEGER; /*Number of resources*/
/**/
DCL
V_NoOfUnkRsc INTEGER; /*Number of unknown resources*/
/**/
DCL
P_CprotPID PID; /*PID of CPROT process responsible for transaction*/
```

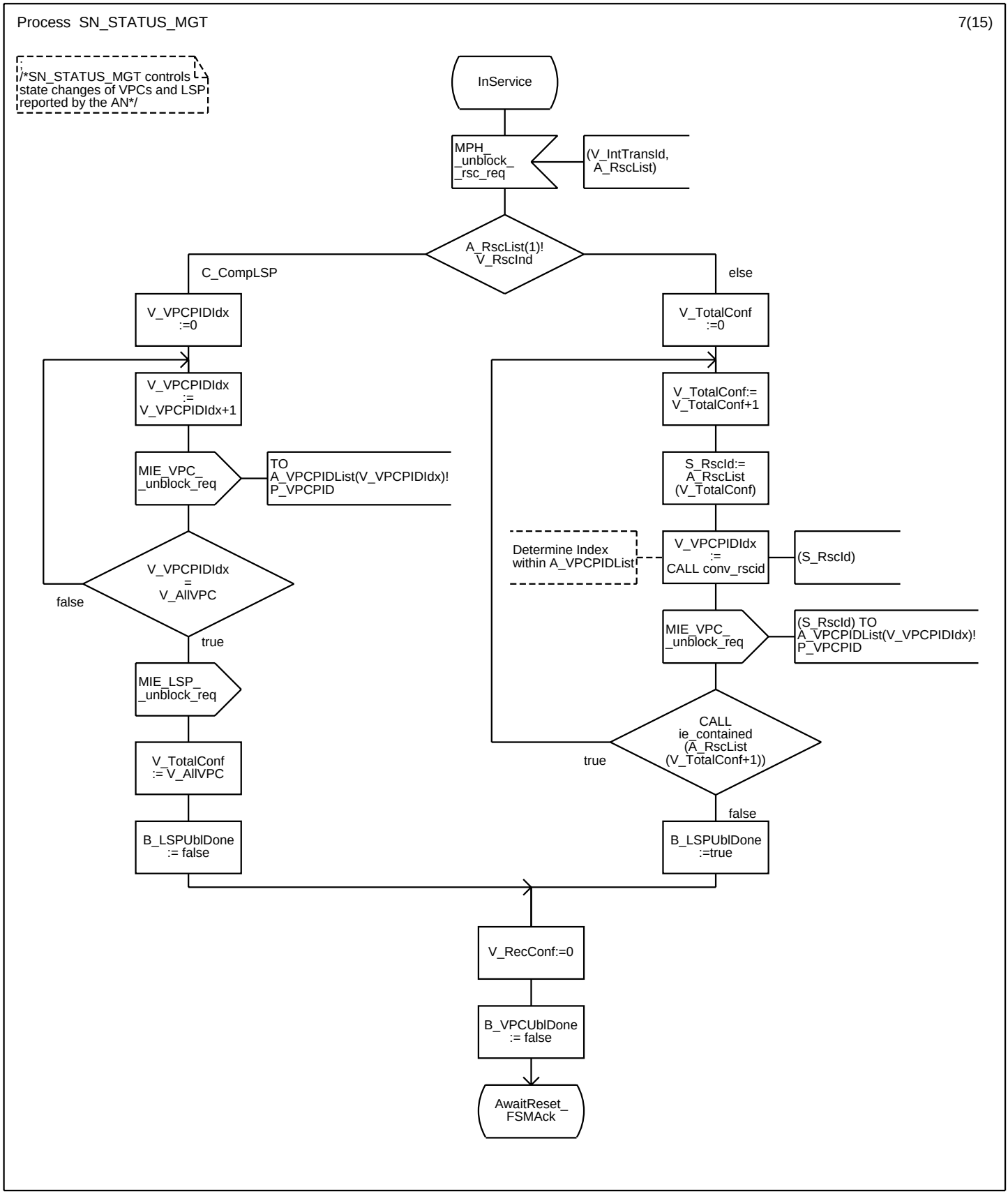
```
/* SN_STATUS_MGT internal variables and constants */
/**/
/*Variables for co-ordinating "unblock_rsc" procedure (e.g. during interface startup or reset of VPC/LSP)*/
/**/
DCL
B_VPCUblDone,    /*MIE_VPC_unblock_conf received from all requested VPC STATFSM*/
B_LSPUblDone    /*MIE_LSP_unblock_conf received or not necessary*/
BOOLEAN;
/**/
DCL
V_Totalconf,    /*number of expected VPC state change confirmations*/
V_RecConf    /*number of received VPC state change confirmations*/
INTEGER;
/**/
/*Variable for decoded blocked resources*/
DCL A_BlRscList AT_BlRscList;
/**/
/*Variable for storing AN_VPCSTAT PIDs*/
DCL A_VPCPIDList AT_VPCPIDList;
DCL V_VPCPIDIdx INTEGER; /*Variable for indexing PID Table*/
/**/
DCL V_AllVPC INTEGER; /*Total number of provisioned VPCs*/
```

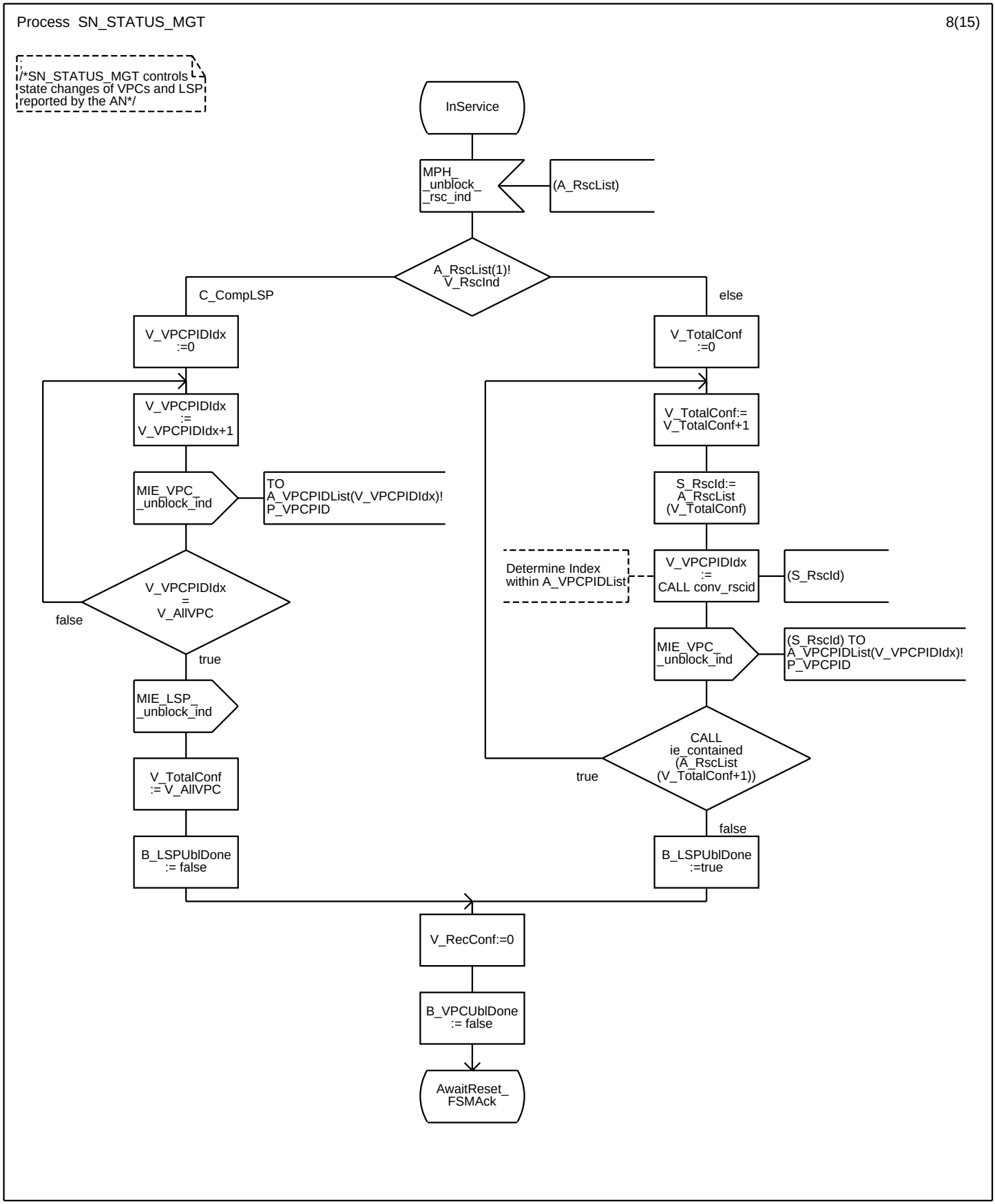

/*SN_STATUS_MGT controls state changes of VPCs and LSP reported by the AN*/
MDU_ _status_ _prc_started
V_AllVPC := 0
OutOfService
*
MPH_stop_ _traffic
MDU_ _abort_req
to all open transactions
OutOfService

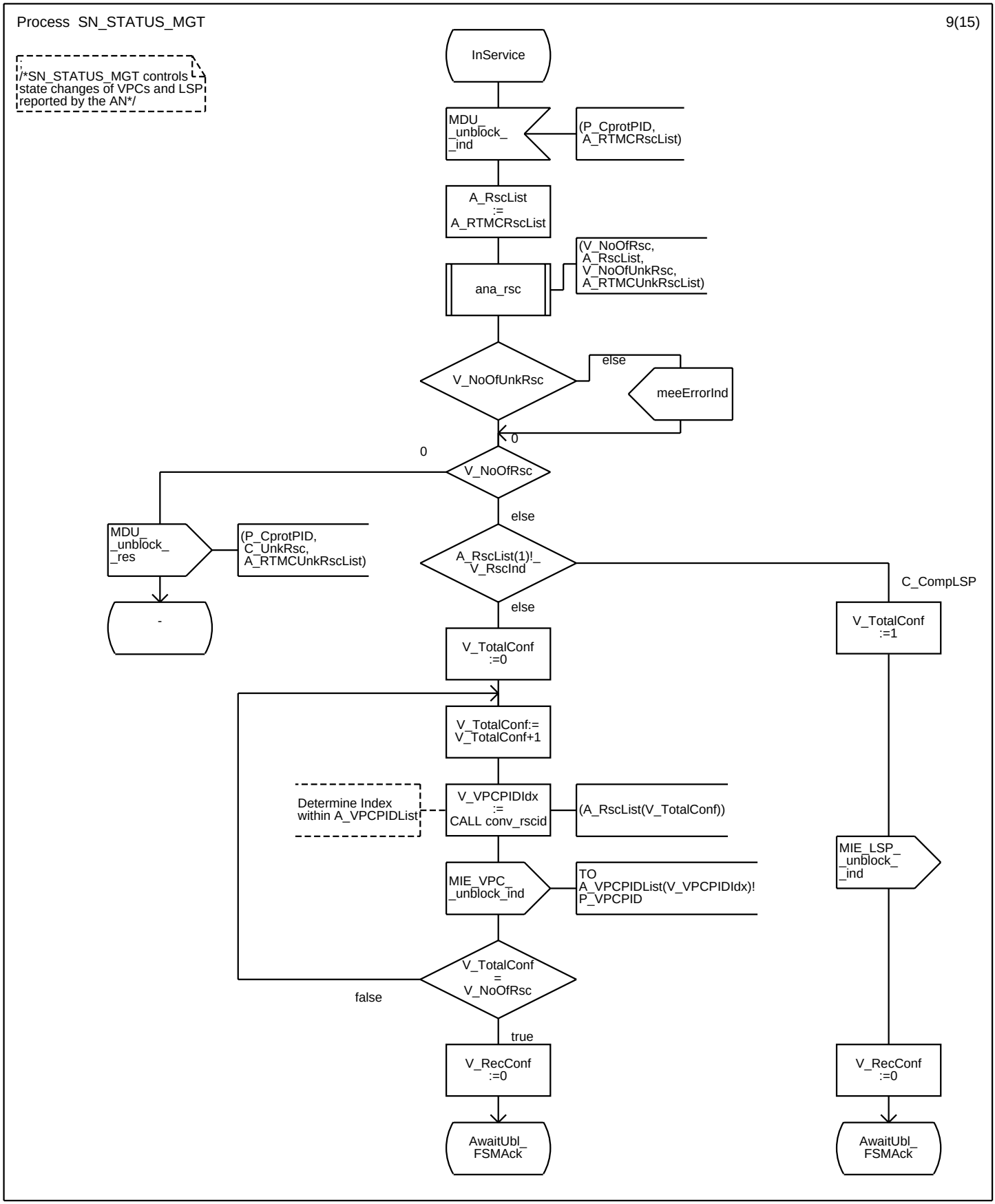

/*SN_STATUS_MGT controls state changes of VPCs and LSP reported by the AN*/
*
meeVPC_ CreateReq
(S_RscId)
meeVPC_ DeleteReq
(S_RscId)
SN_VPCSTAT
V_VPCPIDIdx := CALL conv_rscid(S_RscId)
V_VPCPIDIdx := CALL conv_rscid(S_RscId)
MIE_VPC_ _delete
TO A_VPCPIDList(V_VPCPIDIdx)! P_VPCPID
A_VPCPIDList(V_VPCPIDIdx)! P_VPCPID := OFFSPRING
V_AllVPC := V_AllVPC -1
meeVPC_ DeleteConf
-
A_VPCPIDList(V_VPCPIDIdx)! S_RscId := S_RscId
V_allVPC := V_allVPC +1
meeVPC_ CreateConf
-

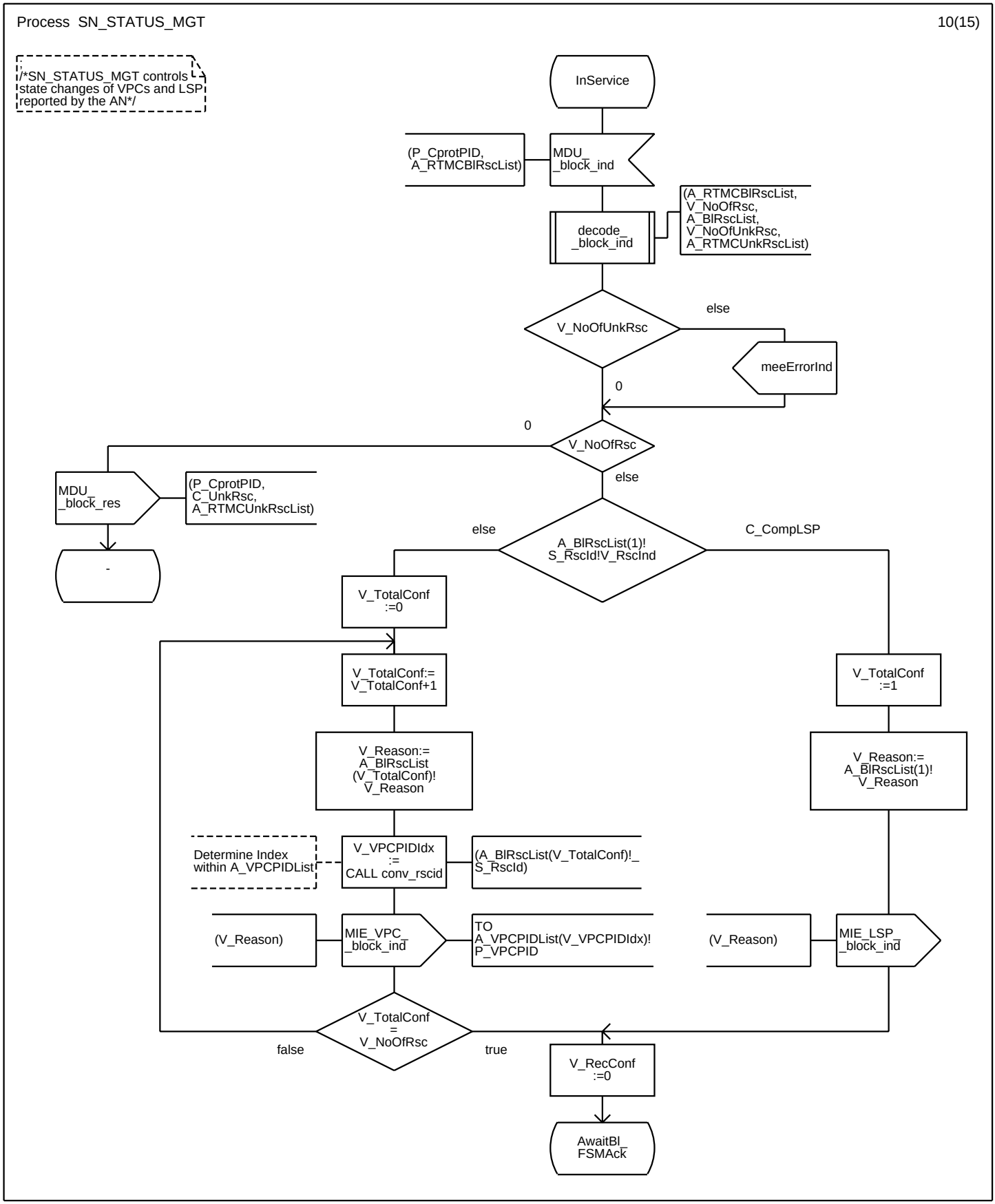

Process SN_STATUS_MGT
5(15)
/*SN_STATUS_MGT controls state changes of VPCs and LSP reported by the AN*/
OutOfService
MPH_ _unblock_ _rsc_req
(V_IntTransId, A_RscList)
V_VPCPIDIdx :=0
false
V_VPCPIDIdx := V_VPCPIDIdx+1
MIE_VPC_ _unblock_req
TO A_VPCPIDList(V_VPCPIDIdx)! P_VPCPID
V_VPCPIDIdx = V_AllVPC
true
MIE_LSP_ _unblock_req
V_TotalConf := V_AllVPC
V_RecConf:=0
B_VPCUblDone := false
B_LSPUblDone := false
UblDuring_ Startup
MPH_ _unblock_ _rsc_ind
(A_RscList)
V_VPCPIDIdx :=0
false
V_VPCPIDIdx := V_VPCPIDIdx+1
MIE_VPC_ _unblock_ind
TO A_VPCPIDList(V_VPCPIDIdx)! P_VPCPID
V_VPCPIDIdx = V_AllVPC
true
MIE_LSP_ _unblock_ind
V_TotalConf := V_AllVPC
V_RecConf:=0
B_VPCUblDone := false
B_LSPUblDone := false
UblDuring_ Startup

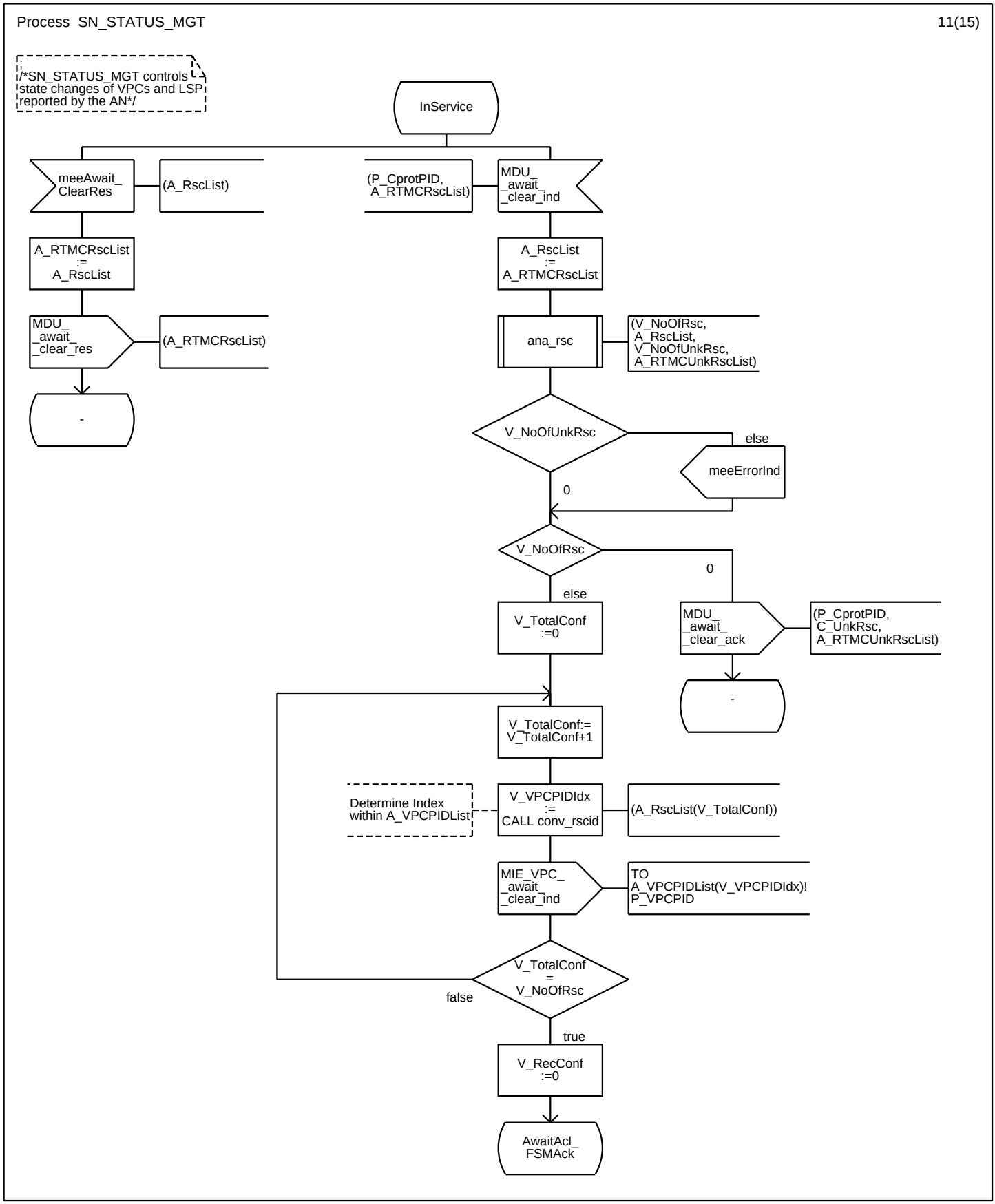

Process SN_STATUS_MGT
6(15)
/*SN_STATUS_MGT controls state changes of VPCs and LSP reported by the AN*/
UblDuring_ Startup
MIE_VPC_ _unblock_ _conf
MIE_LSP_ _unblock_ _conf
MIE_VPC_ _unblock_ _res
MIE_LSP_ _unblock_ _res
V_RecConf:= V_RecConf+1
V_RecConf =V_TotalConf
false
true
B_VPCUbl_ Done:=true
B_LSPUbl_ Done:=true
B_LSPUblDone
B_VPCUblDone
-
MPH_unblock_ _rsc_conf
(V_IntTransId)
MPH_unblock_ _rsc_res
AwaitStartTraffic
MPH_start_ _traffic
InService

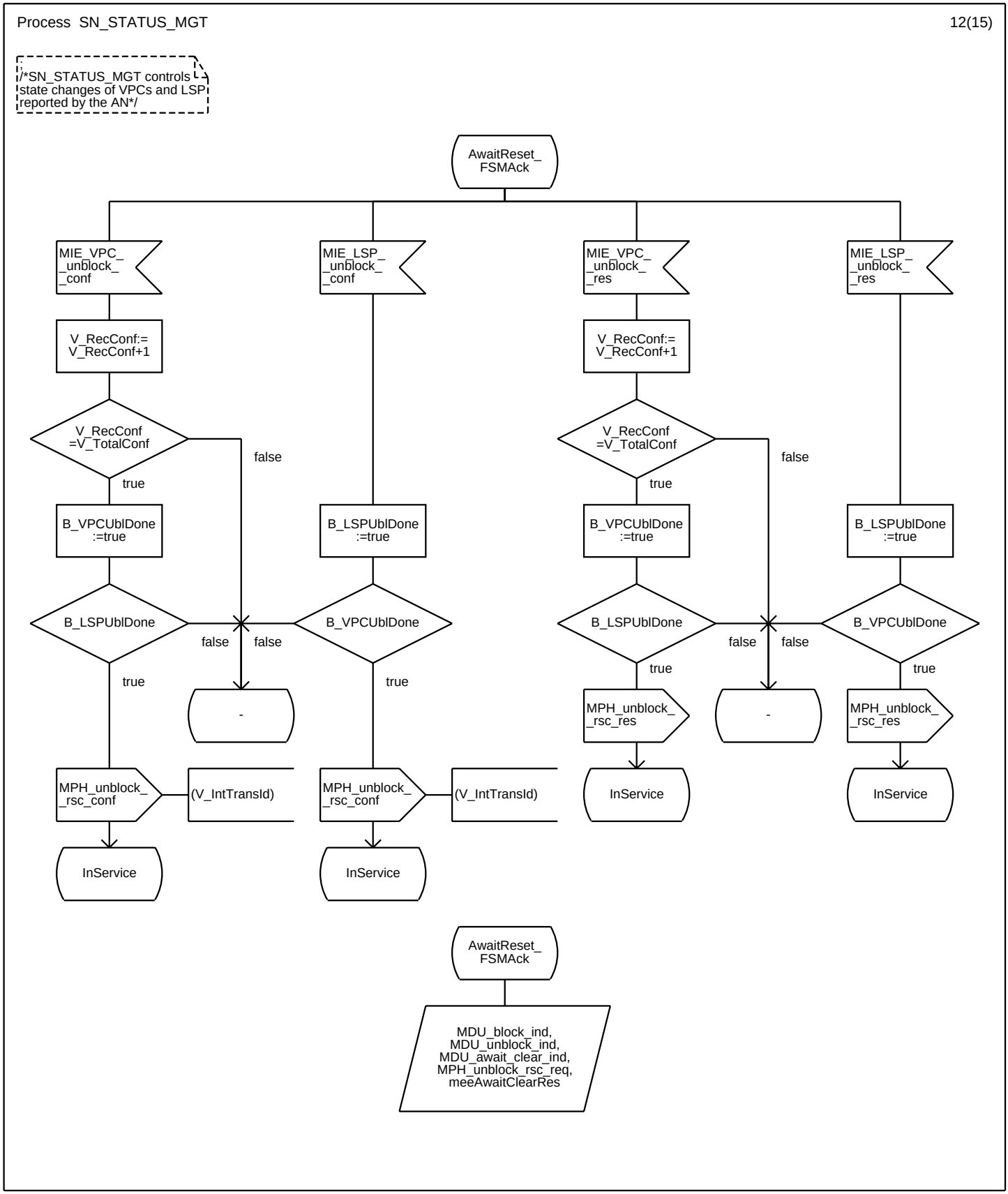

Process SN_STATUS_MGT
7(15)
/*SN_STATUS_MGT controls state changes of VPCs and LSP reported by the AN*/
InService
MPH_ _unblock_ _rsc_req
(V_IntTransId, A_RscList)
A_RscList(1)! V_RscInd
C_CompLSP
else
V_VPCPIDIdx :=0
V_VPCPIDIdx := V_VPCPIDIdx+1
MIE_VPC_ _unblock_req
TO A_VPCPIDList(V_VPCPIDIdx)! P_VPCPID
V_VPCPIDIdx = V_AllVPC
false
true
MIE_LSP_ _unblock_req
V_TotalConf := V_AllVPC
B_LSPUblDone := false
V_TotalConf :=0
V_TotalConf:= V_TotalConf+1
S_RscId:= A_RscList (V_TotalConf)
Determine Index within A_VPCPIDList
V_VPCPIDIdx := CALL conv_rscid
(S_RscId)
MIE_VPC_ _unblock_req
(S_RscId) TO A_VPCPIDList(V_VPCPIDIdx)! P_VPCPID
CALL ie_contained (A_RscList (V_TotalConf+1))
true
false
B_LSPUblDone :=true
V_RecConf:=0
B_VPCUblDone := false
AwaitReset_ FSMAck

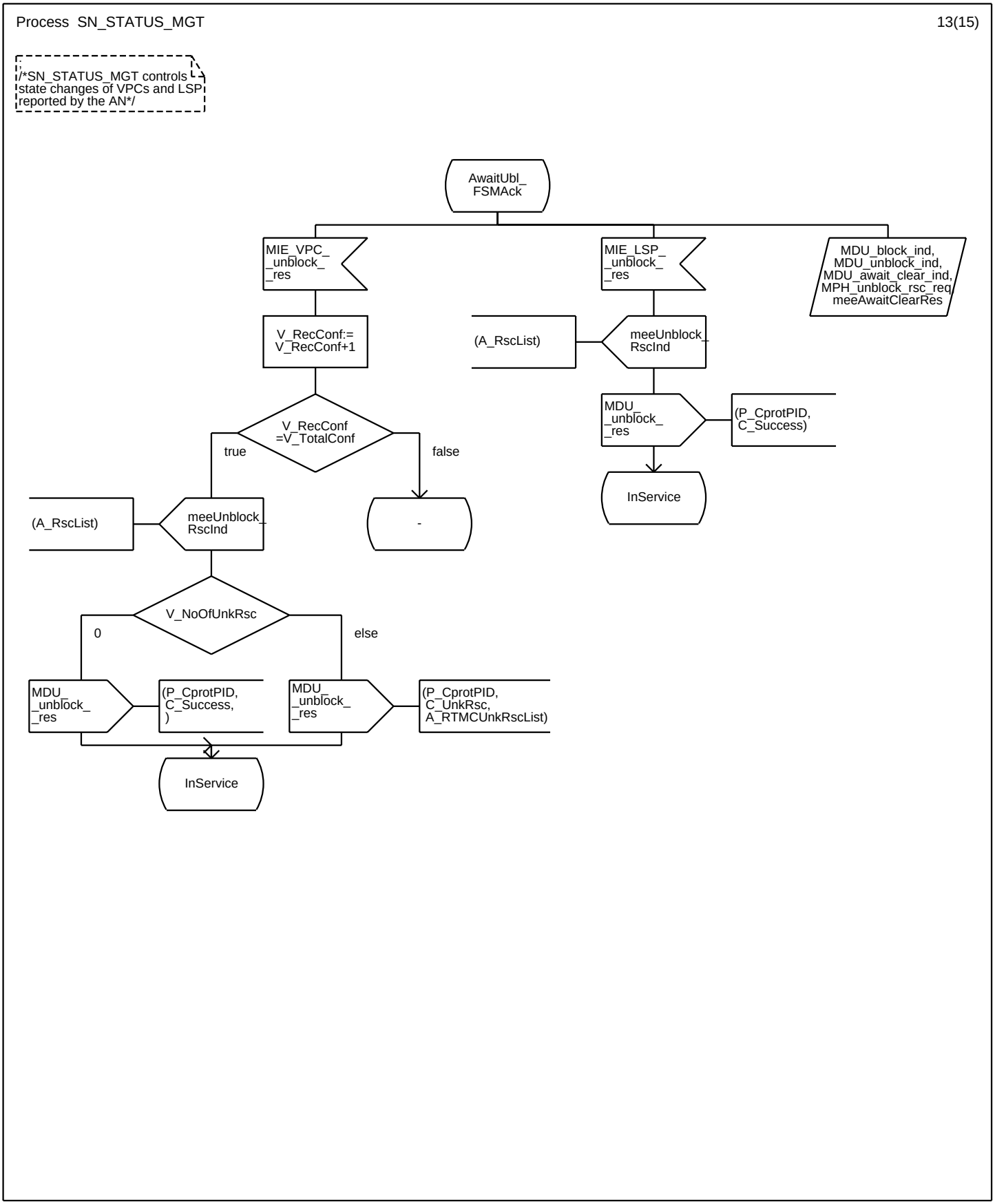

/*SN_STATUS_MGT controls state changes of VPCs and LSP reported by the AN*/
InService
MPH_ _unblock_ _rsc_ind
(A_RscList)
A_RscList(1)! V_RscInd
C_CompLSP
else
V_VPCPIDIdx :=0
V_VPCPIDIdx := V_VPCPIDIdx+1
MIE_VPC_ _unblock_ind
TO A_VPCPIDList(V_VPCPIDIdx)! P_VPCPID
V_VPCPIDIdx = V_AllVPC
false
true
MIE_LSP_ _unblock_ind
V_TotalConf := V_AllVPC
B_LSPUblDone := false
V_TotalConf :=0
V_TotalConf:= V_TotalConf+1
S_RscId:= A_RscList (V_TotalConf)
Determine Index within A_VPCPIDList
V_VPCPIDIdx := CALL conv_rscid
(S_RscId)
MIE_VPC_ _unblock_ind
(S_RscId) TO A_VPCPIDList(V_VPCPIDIdx)! P_VPCPID
CALL ie_contained (A_RscList (V_TotalConf+1))
true
false
B_LSPUblDone :=true
V_RecConf:=0
B_VPCUblDone := false
AwaitReset_ FSMAck

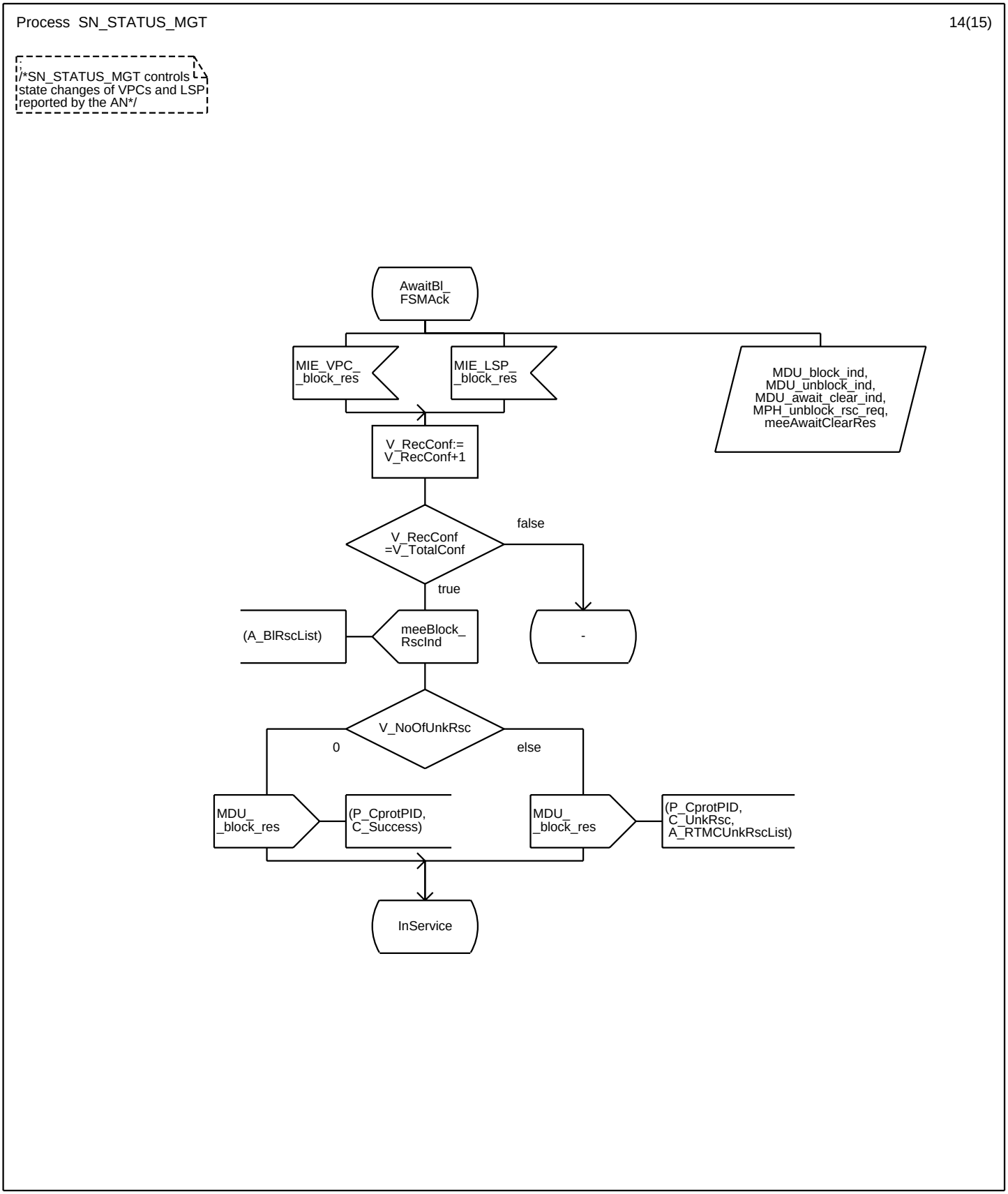

Process SN_STATUS_MGT
9(15)
/*SN_STATUS_MGT controls state changes of VPCs and LSP reported by the AN*/
InService
MDU_ _unblock_ _ind
(P_CprotPID, A_RTMCRscList)
A_RscList := A_RTMCRscList
ana_rsc
(V_NoOfRsc, A_RscList, V_NoOfUnkRsc, A_RTMCUnkRscList)
V_NoOfUnkRsc
else
meeErrorInd
0
0
V_NoOfRsc
else
MDU_ _unblock_ _res
(P_CprotPID, C_UnkRsc, A_RTMCUnkRscList)
A_RscList(1)!_ V_RscInd
C_CompLSP
-
else
V_TotalConf :=1
V_TotalConf :=0
V_TotalConf:= V_TotalConf+1
Determine Index within A_VPCPIDList
V_VPCPIDIdx := CALL conv_rscid
(A_RscList(V_TotalConf))
MIE_LSP_ _unblock_ _ind
MIE_VPC_ _unblock_ind
TO A_VPCPIDList(V_VPCPIDIdx)! P_VPCPID
V_TotalConf = V_NoOfRsc
false
true
V_RecConf :=0
V_RecConf :=0
AwaitUbl_ FSMAck
AwaitUbl_ FSMAck


Process SN_STATUS_MGT
10(15)
/*SN_STATUS_MGT controls state changes of VPCs and LSP reported by the AN*/
InService
(P_CprotPID, A_RTMCBlRscList)
MDU_ _block_ind
decode_ _block_ind
(A_RTMCBlRscList, V_NoOfRsc, A_BlRscList, V_NoOfUnkRsc, A_RTMCUnkRscList)
V_NoOfUnkRsc
else
meeErrorInd
0
0
V_NoOfRsc
else
MDU_ _block_res
(P_CprotPID, C_UnkRsc, A_RTMCUnkRscList)
-
else
A_BlRscList(1)! S_RscId!V_RscInd
C_CompLSP
V_TotalConf :=0
V_TotalConf:= V_TotalConf+1
V_TotalConf :=1
V_Reason:= A_BlRscList (V_TotalConf)! V_Reason
V_Reason:= A_BlRscList(1)! V_Reason
Determine Index within A_VPCPIDList
V_VPCPIDIdx := CALL conv_rscid
(A_BlRscList(V_TotalConf)!_ S_RscId)
(V_Reason)
MIE_VPC_ _block_ind
TO A_VPCPIDList(V_VPCPIDIdx)! P_VPCPID
(V_Reason)
MIE_LSP_ _block_ind
V_TotalConf = V_NoOfRsc
false
true
V_RecConf :=0
AwaitBl_ FSMAck

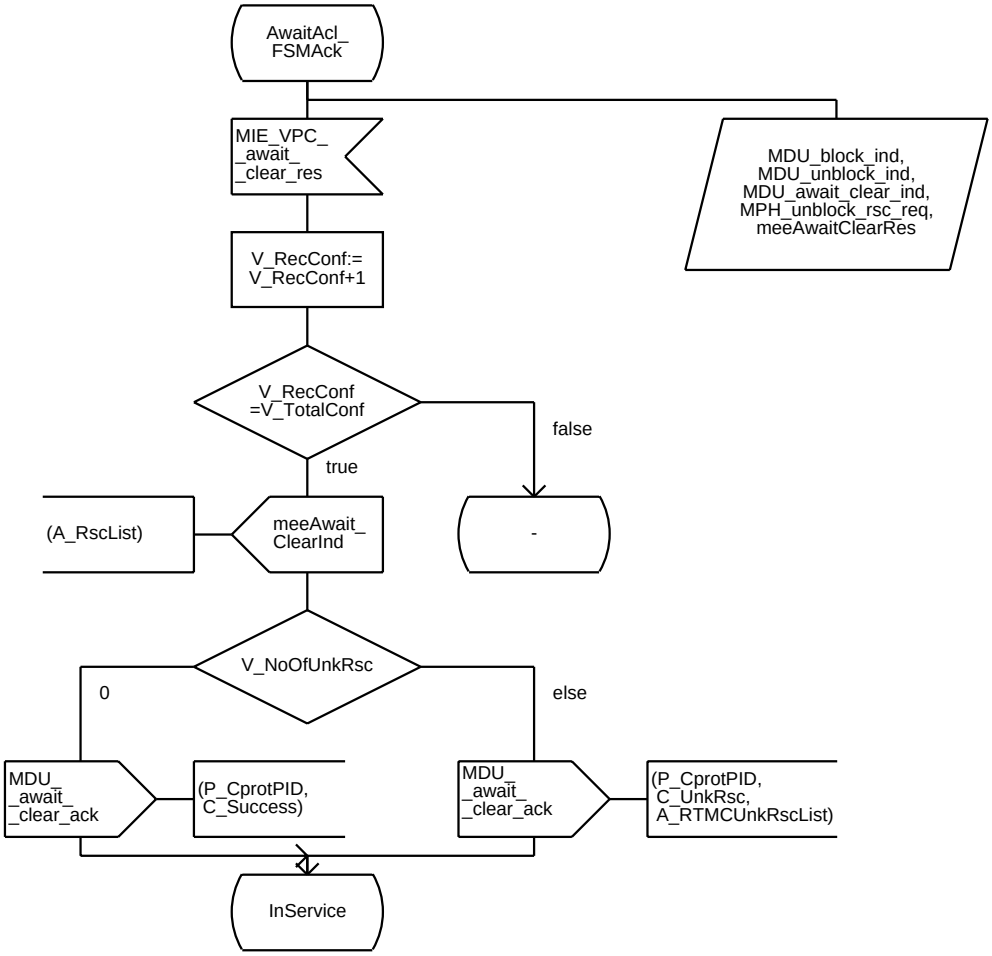

Process SN_STATUS_MGT
11(15)
/*SN_STATUS_MGT controls state changes of VPCs and LSP reported by the AN*/
InService
meeAwait_ ClearRes
(A_RscList)
(P_CprotPID, A_RTMCRscList)
MDU_ _await_ _clear_ind
A_RTMCRscList := A_RscList
A_RscList := A_RTMCRscList
MDU_ _await_ _clear_res
(A_RTMCRscList)
ana_rsc
(V_NoOfRsc, A_RscList, V_NoOfUnkRsc, A_RTMCUnkRscList)
-
V_NoOfUnkRsc
else
meeErrorInd
0
V_NoOfRsc
0
else
V_TotalConf :=0
MDU_ _await_ _clear_ack
(P_CprotPID, C_UnkRsc, A_RTMCUnkRscList)
-
V_TotalConf:= V_TotalConf+1
Determine Index within A_VPCPIDList
V_VPCPIDIdx := CALL conv_rscid
(A_RscList(V_TotalConf))
MIE_VPC_ _await_ _clear_ind
TO A_VPCPIDList(V_VPCPIDIdx)! P_VPCPID
V_TotalConf = V_NoOfRsc
false
true
V_RecConf :=0
AwaitAcl_ FSMAck


Process SN_STATUS_MGT
12(15)
/*SN_STATUS_MGT controls state changes of VPCs and LSP reported by the AN*/
AwaitReset_ FSMAck
MIE_VPC_ _unblock_ _conf
MIE_LSP_ _unblock_ _conf
MIE_VPC_ _unblock_ _res
MIE_LSP_ _unblock_ _res
V_RecConf:= V_RecConf+1
V_RecConf =V_TotalConf
false
true
B_VPCUblDone :=true
B_LSPUblDone :=true
B_LSPUblDone
B_VPCUblDone
false
false
true
true
-
MPH_unblock_ _rsc_conf
(V_IntTransId)
InService
MPH_unblock_ _rsc_res
AwaitReset_ FSMAck
MDU_block_ind, MDU_unblock_ind, MDU_await_clear_ind, MPH_unblock_rsc_req, meeAwaitClearRes


/*SN_STATUS_MGT controls state changes of VPCs and LSP reported by the AN*/
AwaitUbl_ FSMAck
MIE_VPC_ _unblock_ _res
MIE_LSP_ _unblock_ _res
MDU_block_ind, MDU_unblock_ind, MDU_await_clear_ind, MPH_unblock_rsc_req, meeAwaitClearRes
V_RecConf:= V_RecConf+1
(A_RscList)
meeUnblock_ RscInd
MDU_ _unblock_ _res
(P_CprotPID, C_Success)
InService
V_RecConf =V_TotalConf
true
false
(A_RscList)
meeUnblock_ RscInd
-
V_NoOfUnkRsc
0
else
MDU_ _unblock_ _res
(P_CprotPID, C_Success,)
MDU_ _unblock_ _res
(P_CprotPID, C_UnkRsc, A_RTMCUnkRscList)
InService


/*SN_STATUS_MGT controls state changes of VPCs and LSP reported by the AN*/
AwaitBl_ FSMAck
MIE_VPC_ _block_res
MIE_LSP_ _block_res
MDU_block_ind, MDU_unblock_ind, MDU_await_clear_ind, MPH_unblock_rsc_req, meeAwaitClearRes
V_RecConf:= V_RecConf+1
V_RecConf =V_TotalConf
false
true
(A_BlRscList)
meeBlock_ RscInd
-
V_NoOfUnkRsc
0
else
MDU_ _block_res
(P_CprotPID, C_Success)
MDU_ _block_res
(P_CprotPID, C_UnkRsc, A_RTMCUnkRscList)
InService

/*SN_STATUS_MGT controls state changes of VPCs and LSP reported by the AN*/



Procedure decode_block_ind 1(5)

```
;FPAR
A_RTMCBlRscList                AT_RTMCBlRscList,
IN/OUT V_NoOfRsc                INTEGER,
IN/OUT A_BlRscList              AT_BlRscList,
IN/OUT V_NoOfUnkRsc             INTEGER,
IN/OUT A_RTMCUnkRscList         AT_RscList;
```

```
/*This procedure decodes the list of blocked resources
reported by the AN: it checks whether the resource is known
and transforms the reason IE into an internal value (by
using the procedure reason_decode*/
```

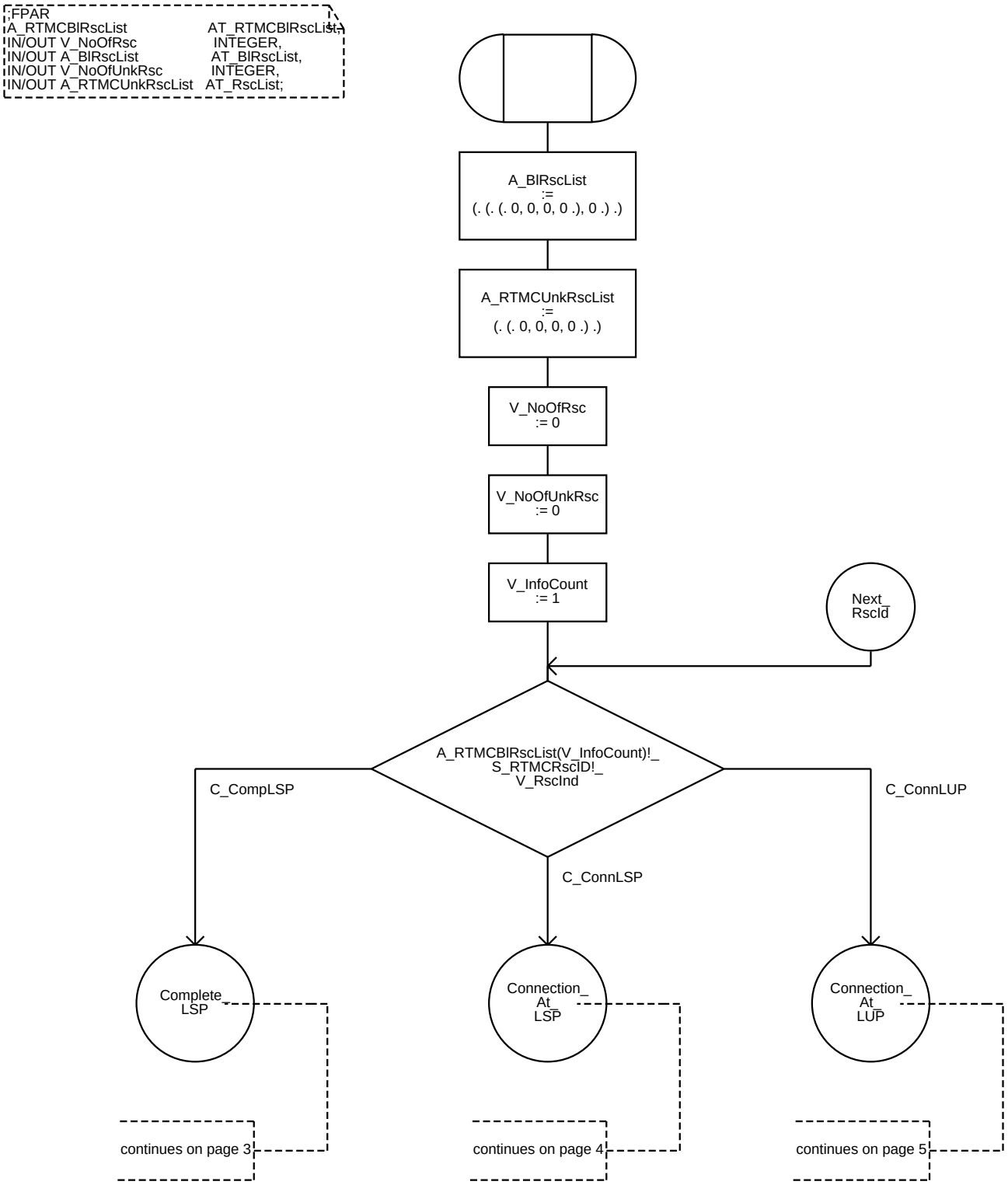
```
/*Definitions and Declarations*/
```

```
/*Data declarations for internal use*/
DCL
A_TempRscList  AT_RscList   ; /*Array of Resource Identifier Information Elements*/
```

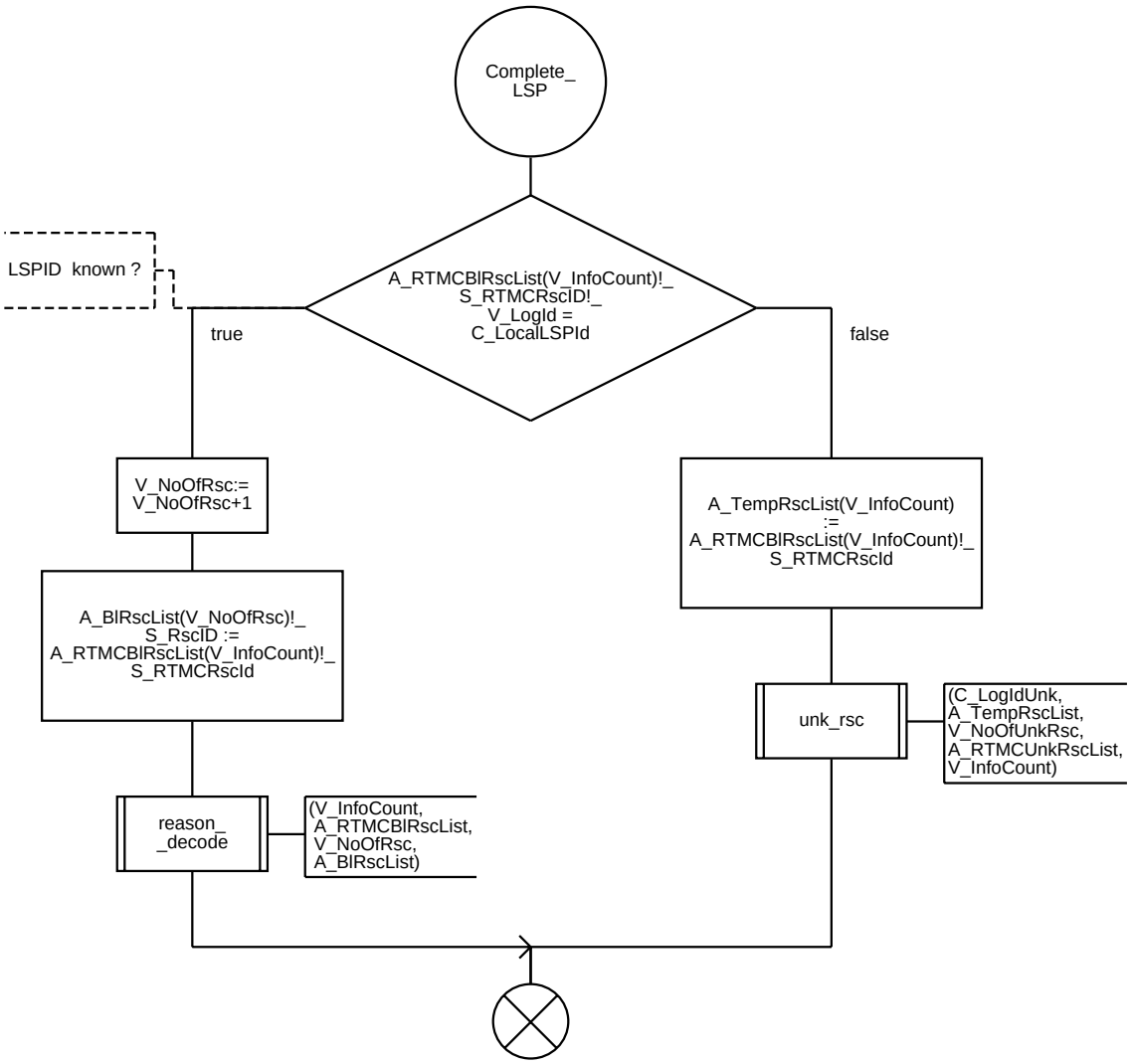
```
/*Internal variables of DECODE_BLOCK_IND*/
/**/
DCL V_InfoCount INTEGER;
```

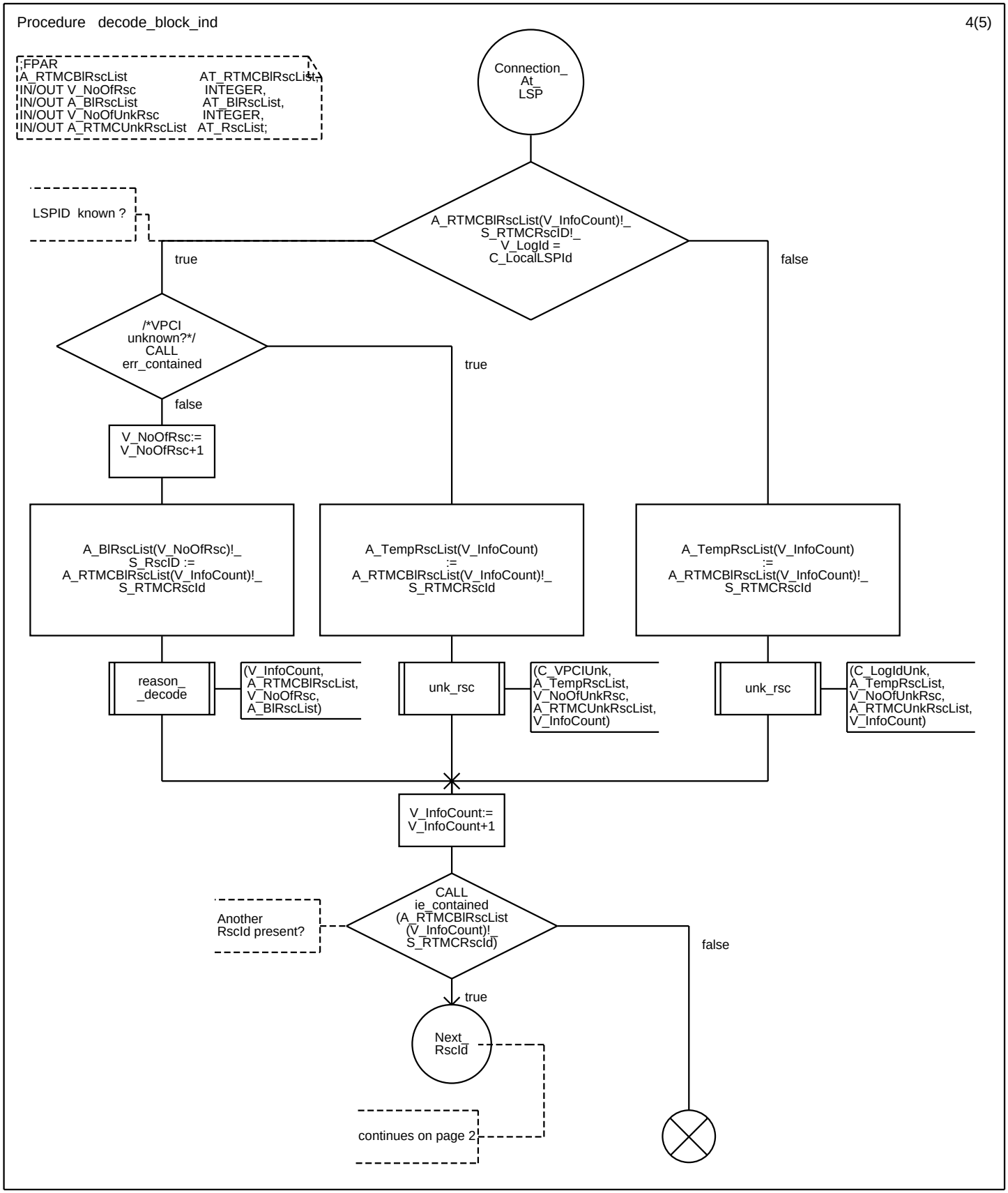
Procedure which converts the RTMC reason into an internal value

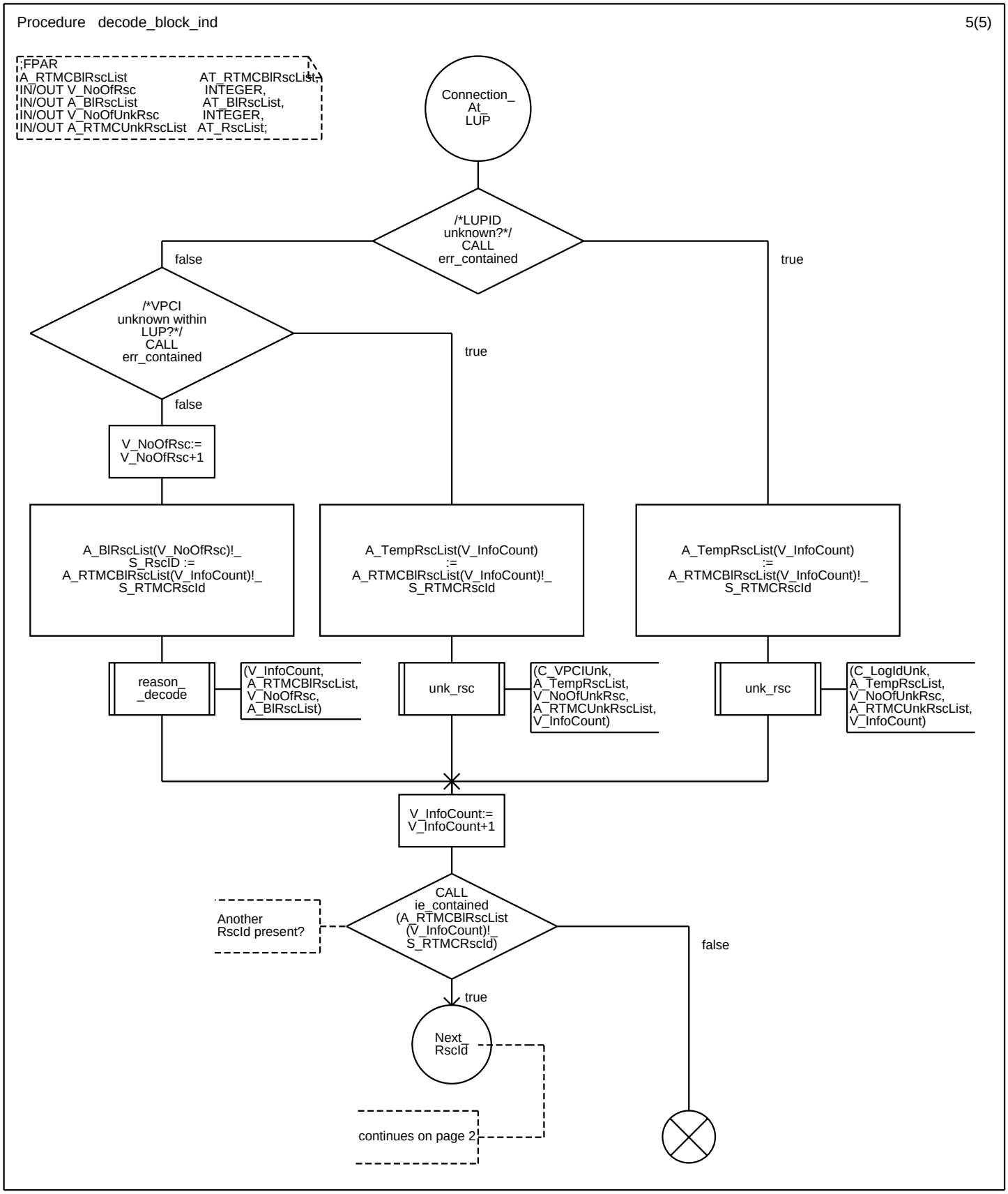
reason_decode

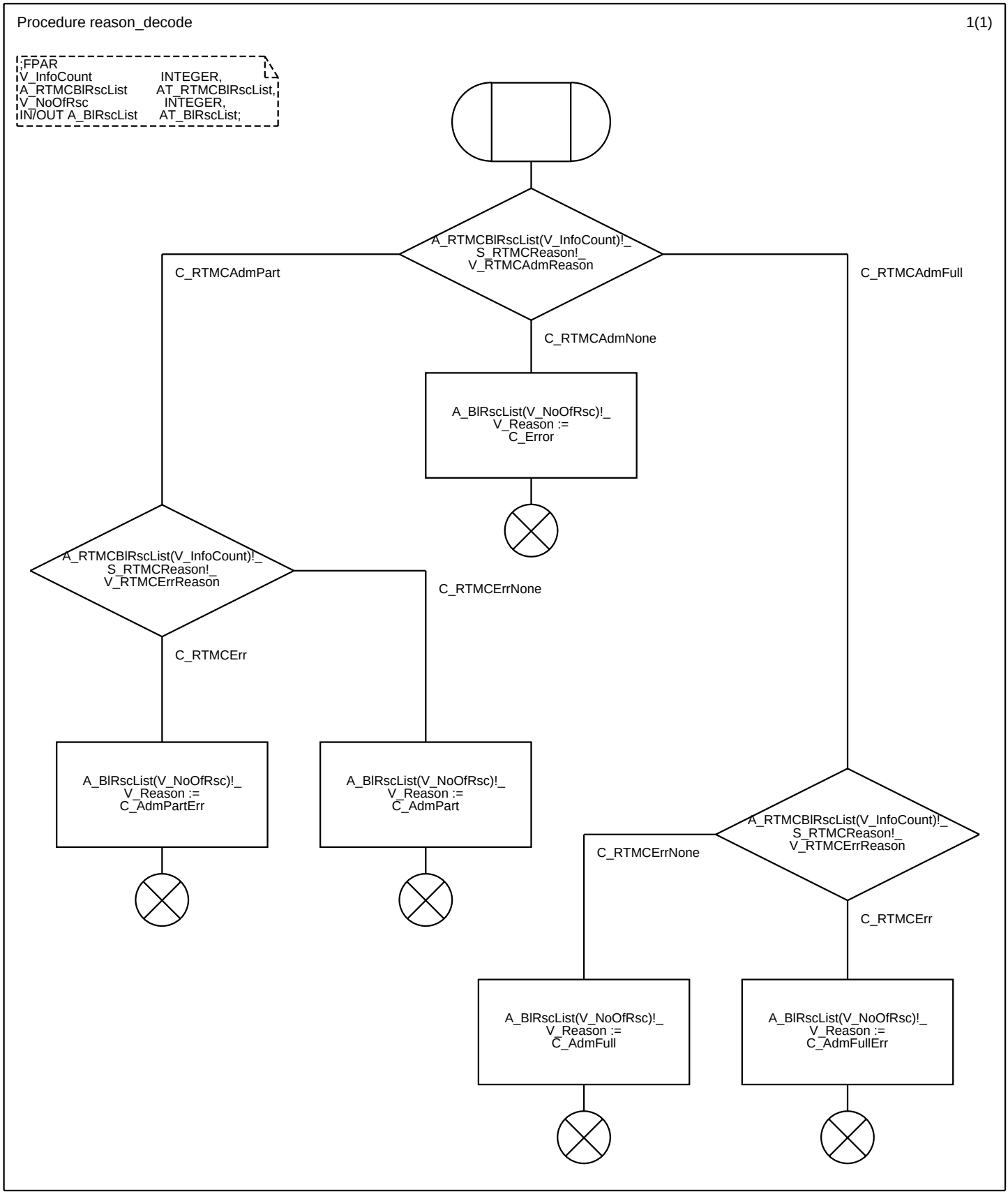

;FPAR
A_RTMCBlRscList AT_RTMCBlRscList,
IN/OUT V_NoOfRsc INTEGER,
IN/OUT A_BlRscList AT_BlRscList,
IN/OUT V_NoOfUnkRsc INTEGER,
IN/OUT A_RTMCUnkRscList AT_RscList;
A_BlRscList := (. (. (. 0, 0, 0, 0 .), 0 .) .)
A_RTMCUnkRscList := (. (. 0, 0, 0, 0 .) .)
V_NoOfRsc := 0
V_NoOfUnkRsc := 0
V_InfoCount := 1
Next_ RscId
A_RTMCBlRscList(V_InfoCount)!_ S_RTMCRscID!_ V_RscInd
C_CompLSP
C_ConnLSP
C_ConnLUP
Complete_ LSP
Connection_ At_ LSP
Connection_ At_ LUP
continues on page 3
continues on page 4
continues on page 5

```
;FPAR
A_RTMCBlRscList              AT_RTMCBlRscList,
IN/OUT V_NoOfRsc              INTEGER,
IN/OUT A_BlRscList            AT_BlRscList,
IN/OUT V_NoOfUnkRsc           INTEGER,
IN/OUT A_RTMCUnkRscList       AT_RscList;
```




Procedure decode_block_ind
4(5)
;FPAR
A_RTMCBlRscList AT_RTMCBlRscList;
IN/OUT V_NoOfRsc INTEGER,
IN/OUT A_BlRscList AT_BlRscList,
IN/OUT V_NoOfUnkRsc INTEGER,
IN/OUT A_RTMCUnkRscList AT_RscList;
Connection_ At_ LSP
LSPID known ?
A_RTMCBlRscList(V_InfoCount)!_ S_RTMCRscID!_ V_LogId = C_LocalLSPId
true
false
/*VPCI unknown?*/ CALL err_contained
true
false
V_NoOfRsc:= V_NoOfRsc+1
A_BlRscList(V_NoOfRsc)!_ S_RscID := A_RTMCBlRscList(V_InfoCount)!_ S_RTMCRscId
A_TempRscList(V_InfoCount) := A_RTMCBlRscList(V_InfoCount)!_ S_RTMCRscId
A_TempRscList(V_InfoCount) := A_RTMCBlRscList(V_InfoCount)!_ S_RTMCRscId
reason_ _decode
(V_InfoCount, A_RTMCBlRscList, V_NoOfRsc, A_BlRscList)
unk_rsc
(C_VPCIUnk, A_TempRscList, V_NoOfUnkRsc, A_RTMCUnkRscList, V_InfoCount)
unk_rsc
(C_LogIdUnk, A_TempRscList, V_NoOfUnkRsc, A_RTMCUnkRscList, V_InfoCount)
V_InfoCount:= V_InfoCount+1
Another RscId present?
CALL ie_contained (A_RTMCBlRscList (V_InfoCount)!_ S_RTMCRscId)
false
true
Next_ RscId
continues on page 2


Procedure decode_block_ind
5(5)
;FPAR
A_RTMCBlRscList AT_RTMCBlRscList;
IN/OUT V_NoOfRsc INTEGER,
IN/OUT A_BlRscList AT_BlRscList,
IN/OUT V_NoOfUnkRsc INTEGER,
IN/OUT A_RTMCUnkRscList AT_RscList;
Connection_ At_ LUP
/*LUPID unknown?*/ CALL err_contained
false
true
/*VPCI unknown within LUP?*/ CALL err_contained
true
false
V_NoOfRsc:= V_NoOfRsc+1
A_BlRscList(V_NoOfRsc)!_ S_RscID := A_RTMCBlRscList(V_InfoCount)!_ S_RTMCRscId
A_TempRscList(V_InfoCount) := A_RTMCBlRscList(V_InfoCount)!_ S_RTMCRscId
A_TempRscList(V_InfoCount) := A_RTMCBlRscList(V_InfoCount)!_ S_RTMCRscId
reason_ _decode
(V_InfoCount, A_RTMCBlRscList, V_NoOfRsc, A_BlRscList)
unk_rsc
(C_VPCIUnk, A_TempRscList, V_NoOfUnkRsc, A_RTMCUnkRscList, V_InfoCount)
unk_rsc
(C_LogIdUnk, A_TempRscList, V_NoOfUnkRsc, A_RTMCUnkRscList, V_InfoCount)
V_InfoCount:= V_InfoCount+1
Another RscId present?
CALL ie_contained (A_RTMCBlRscList (V_InfoCount)!_ S_RTMCRscId)
false
true
Next_ RscId
continues on page 2


Procedure reason_decode
1(1)
;FPAR
V_InfoCount INTEGER,
A_RTMCBlRscList AT_RTMCBlRscList,
V_NoOfRsc INTEGER,
IN/OUT A_BlRscList AT_BlRscList;
A_RTMCBlRscList(V_InfoCount)!_
S_RTMCReason!_
V_RTMCAdmReason
C_RTMCAdmPart
C_RTMCAdmFull
C_RTMCAdmNone
A_BlRscList(V_NoOfRsc)!_
V_Reason :=
C_Error
A_RTMCBlRscList(V_InfoCount)!_
S_RTMCReason!_
V_RTMCErrReason
C_RTMCErrNone
C_RTMCErr
A_BlRscList(V_NoOfRsc)!_
V_Reason :=
C_AdmPartErr
A_BlRscList(V_NoOfRsc)!_
V_Reason :=
C_AdmPart
A_RTMCBlRscList(V_InfoCount)!_
S_RTMCReason!_
V_RTMCErrReason
C_RTMCErrNone
C_RTMCErr
A_BlRscList(V_NoOfRsc)!_
V_Reason :=
C_AdmFull
A_BlRscList(V_NoOfRsc)!_
V_Reason :=
C_AdmFullErr

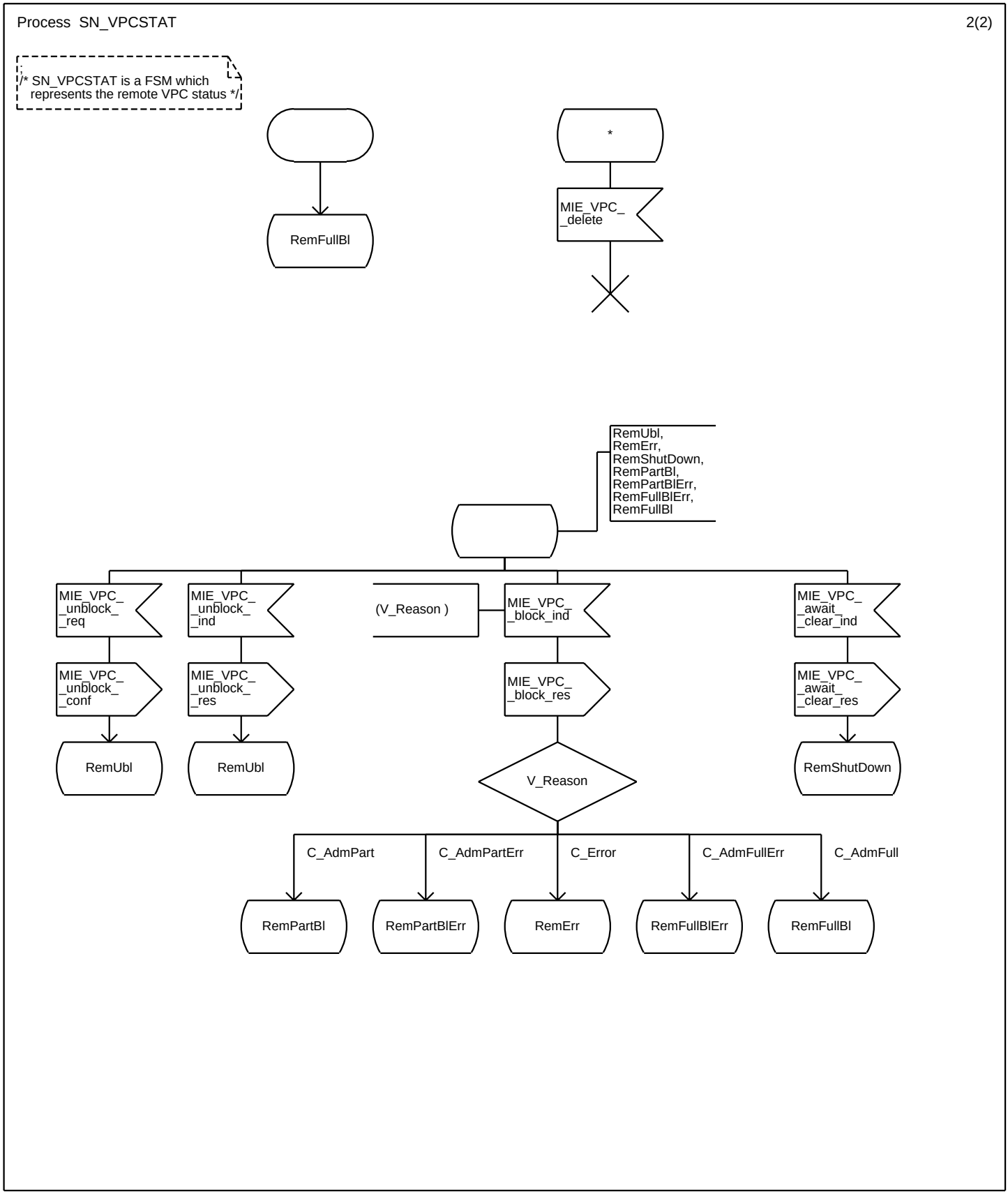
Process SN_VPCSTAT 1(2)

```
;
/* SN_VPCSTAT is a FSM which
   represents the remote VPC status */
```

```
/* State descriptions

RemUbl:          VPC available for Service
RemFullBl:        VPC unavailable for service due to locking by AN operator
RemPartBl:       VPC partially unavailable due to partial locking by AN operator
RemErr:          VPC unavailable due to fault conditions
RemFullBlErr:     VPC unavailable due to locking by AN operator AND fault conditions
RemPartBlErr:   VPC unavailable due to partial locking by AN operator AND fault conditions
RemShutDown: Shutting Down of VPC was requested by AN operator */
```

```
/*Signal data declarations*/
DCL
V_Reason IT_Reason; /*Reason for status change given bySN_STATUS_MGT*/
```

```
/* The state of the VC carrying the RTMC protocol is affected by the
   administrative state of the VPC it belongs to. I.e. it is not possible
   to send or receive RTMC messages if the administrative state of the VPC
   is set to full blocked or the VPC is disabled. This is not considered
   in the SDLs but has to be taken into account in an implementation.
*/
```

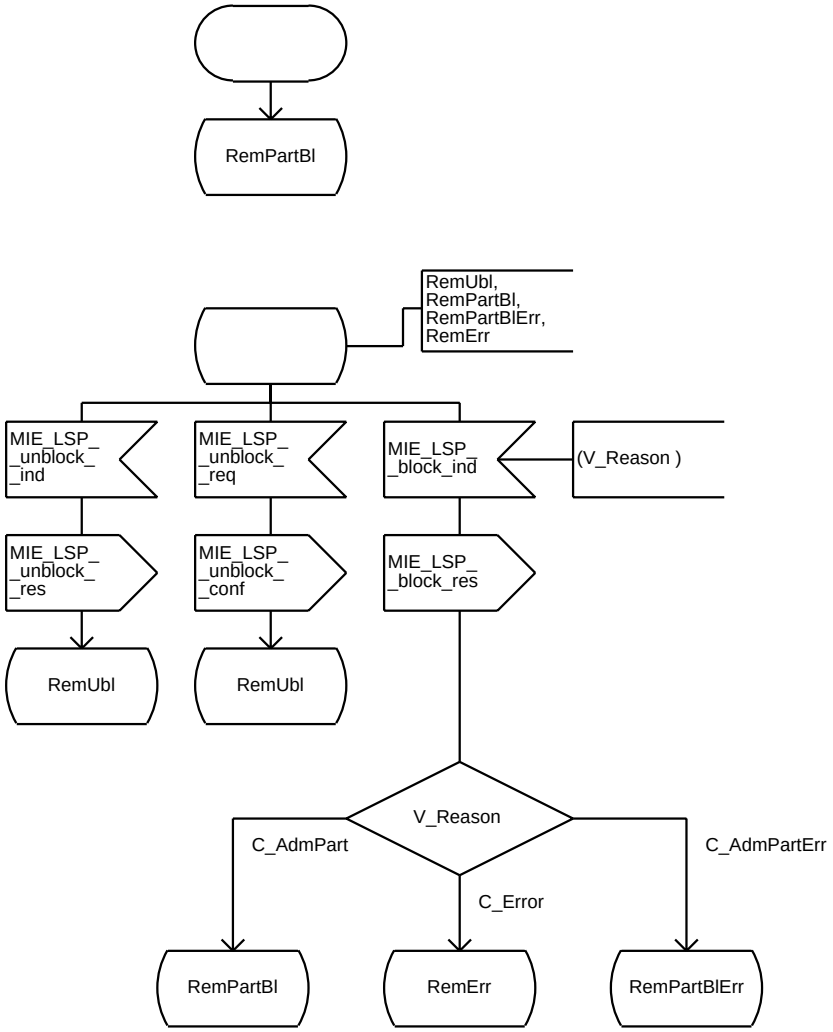

Process SN_VPCSTAT
2(2)
/* SN_VPCSTAT is a FSM which
represents the remote VPC status */
RemFullBl
*
MIE_VPC_
_delete
RemUbl,
RemErr,
RemShutDown,
RemPartBl,
RemPartBlErr,
RemFullBlErr,
RemFullBl
MIE_VPC_
unblock
_req
MIE_VPC_
unblock
_ind
(V_Reason)
MIE_VPC_
_block_ind
MIE_VPC_
await
_clear_ind
MIE_VPC_
unblock
_conf
MIE_VPC_
unblock
_res
MIE_VPC_
_block_res
MIE_VPC_
await
_clear_res
RemUbl
RemUbl
V_Reason
RemShutDown
C_AdmPart
C_AdmPartErr
C_Error
C_AdmFullErr
C_AdmFull
RemPartBl
RemPartBlErr
RemErr
RemFullBlErr
RemFullBl

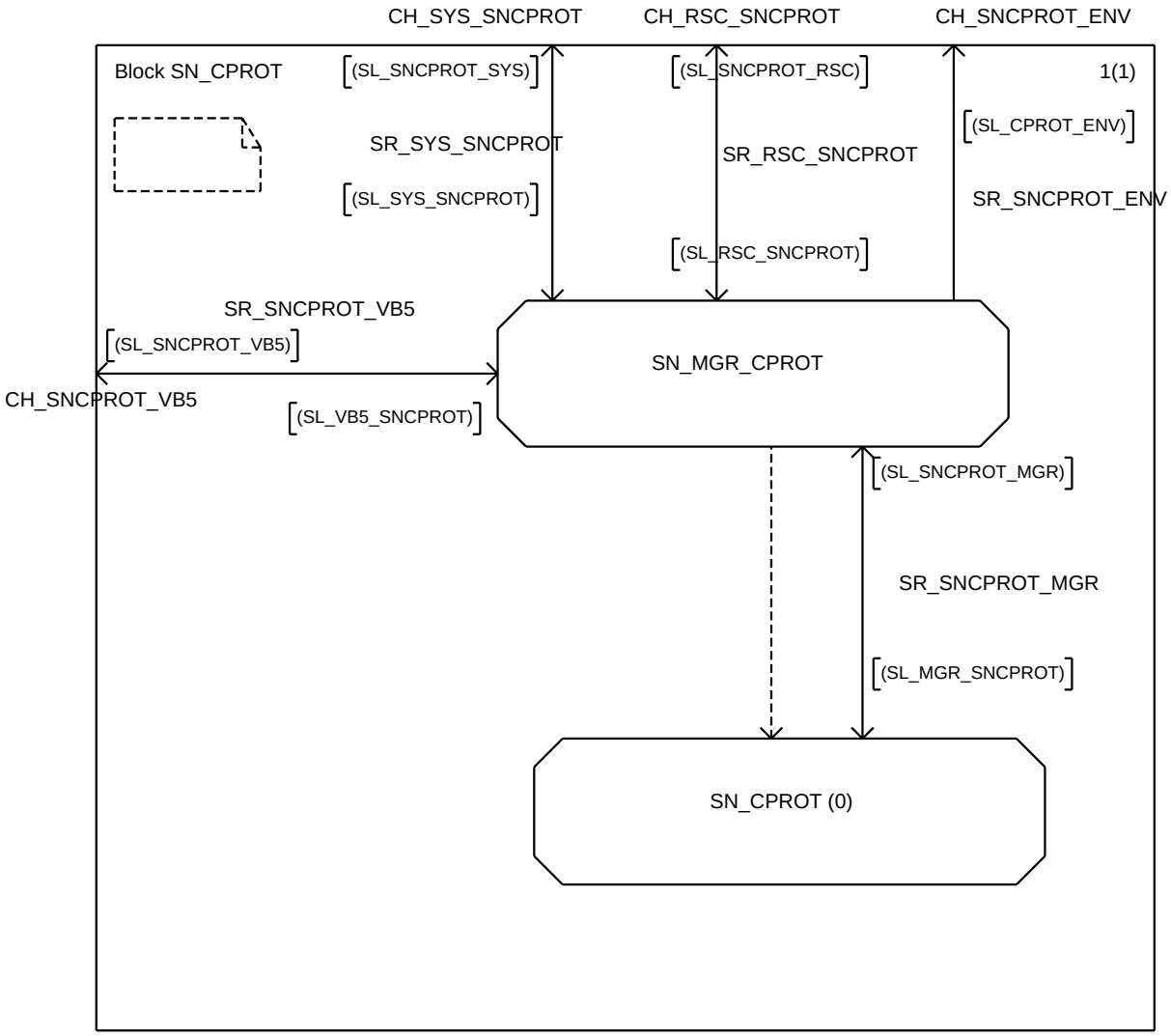
Process SN_LSPSTAT 1(1)

```
;
/* SN_LSPSTAT is a FSM which
   represents the remote LSP status */
```

```
/* State descriptions

RemUbl:          LSP available for Service
RemPartBl:       LSP partially unavailable due to partial locking by AN operator
RemErr:          LSP unavailable due to fault conditions
RemPartBlErr:    LSP unavailable due to partial locking by AN operator AND fault conditions */
```

```
/*Signal data declarations*/
DCL
V_Reason IT_Reason; /*Reason for status change given by SN_STATUS_MGT*/
```



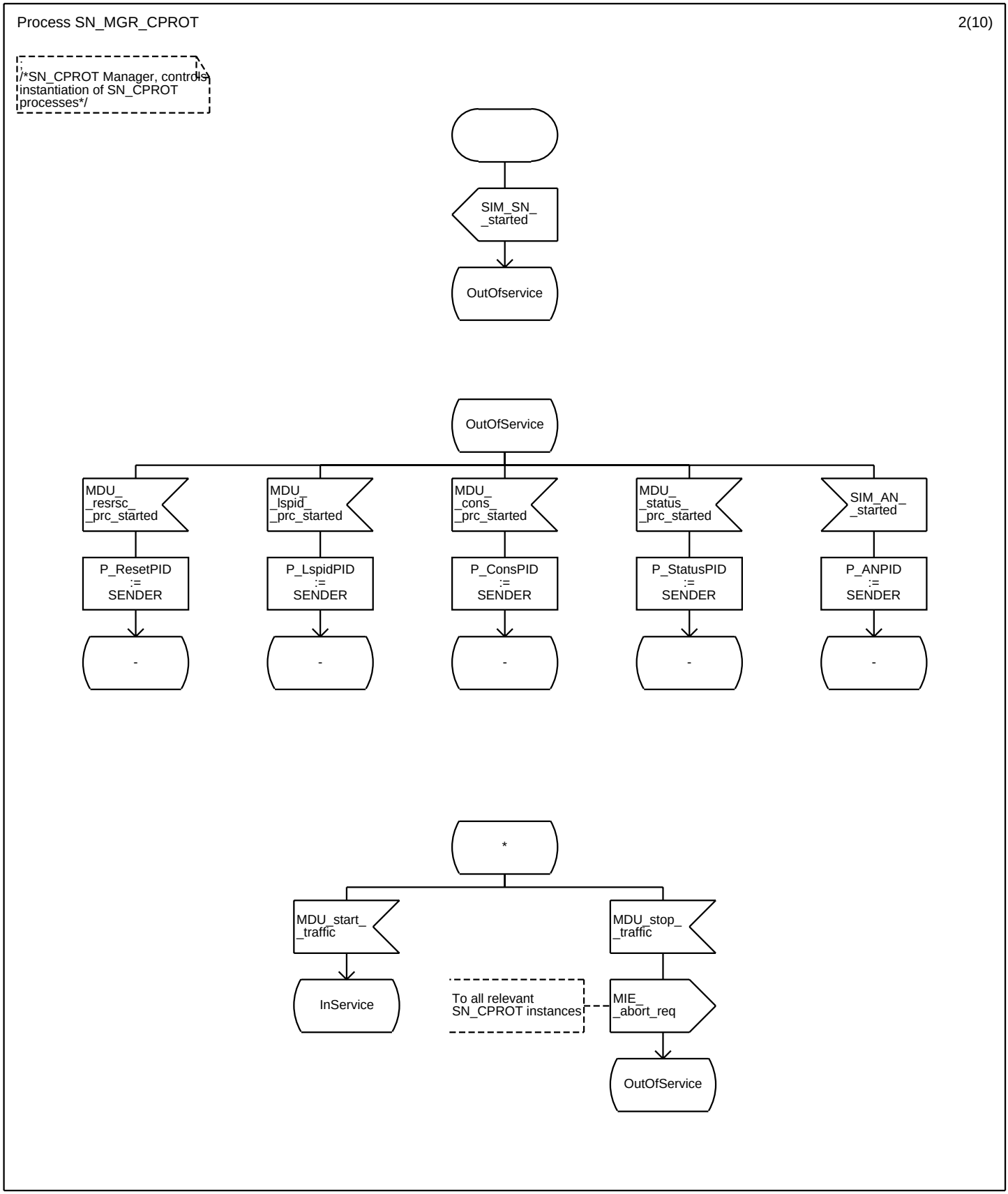

CH_SYS_SNCPROT
CH_RSC_SNCPROT
CH_SNCPROT_ENV
Block SN_CPROT
(SL_SNCPROT_SYS)
(SL_SNCPROT_RSC)
1(1)
(SL_CPROT_ENV)
SR_SYS_SNCPROT
SR_RSC_SNCPROT
(SL_SYS_SNCPROT)
SR_SNCPROT_ENV
(SL_RSC_SNCPROT)
SR_SNCPROT_VB5
(SL_SNCPROT_VB5)
SN_MGR_CPROT
CH_SNCPROT_VB5
(SL_VB5_SNCPROT)
(SL_SNCPROT_MGR)
SR_SNCPROT_MGR
(SL_MGR_SNCPROT)
SN_CPROT (0)

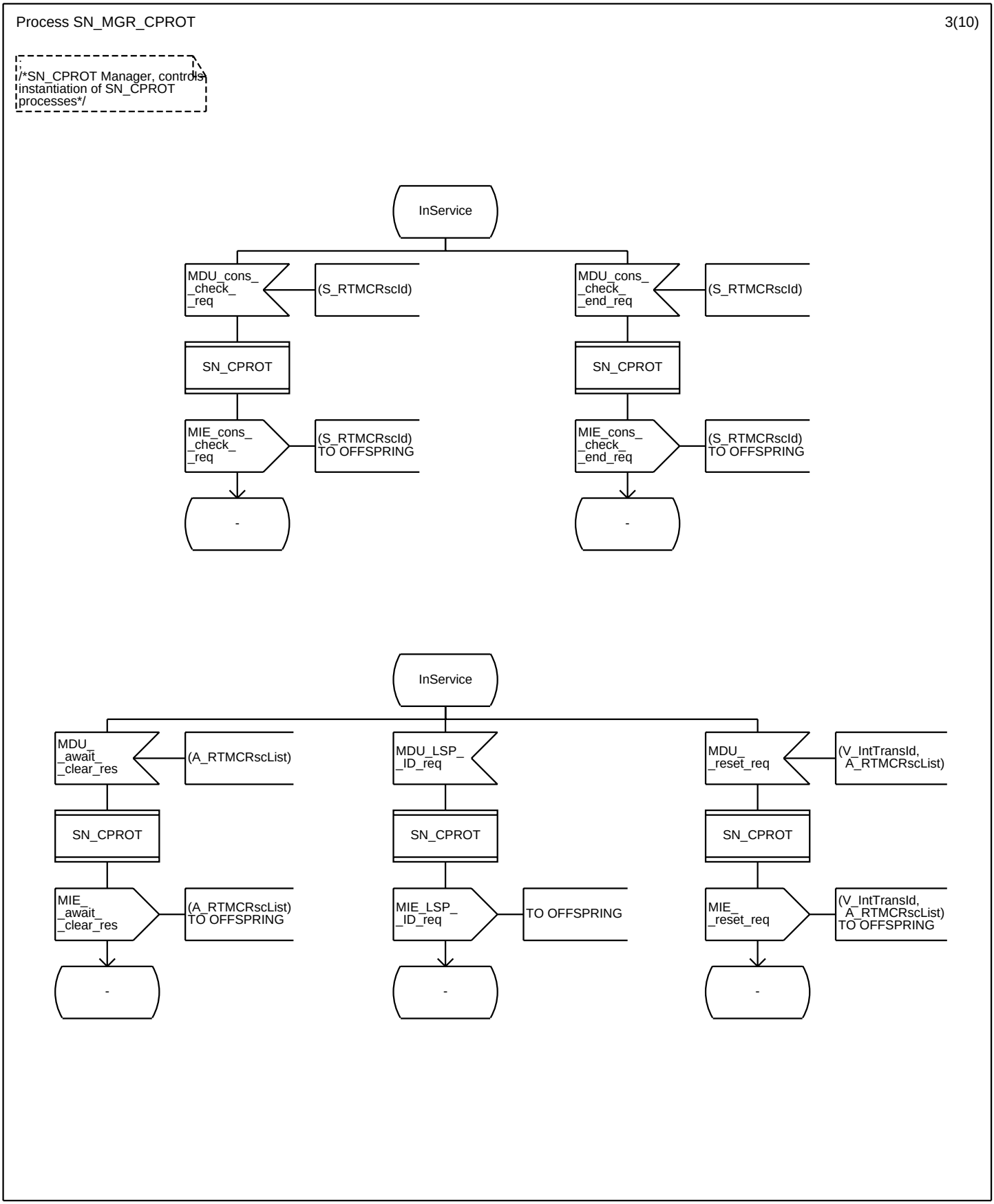
Process SN_MGR_CPROT 1(10)

```
/*SN_CPROT Manager, controls
instantiation of SN_CPROT
processes*/
```

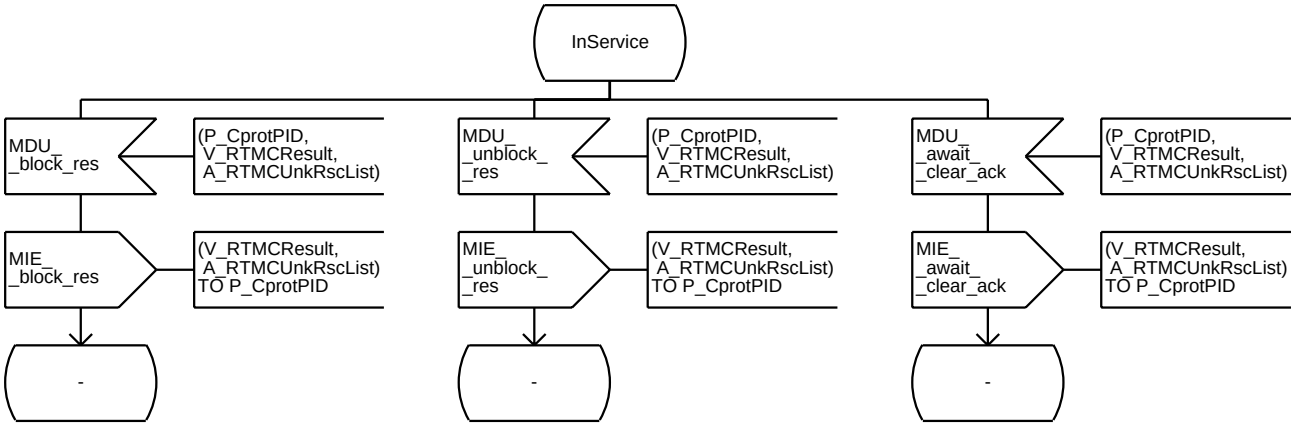
```
NEWTYPE AT_PIDList ARRAY /*This array maps the PID of the SN_CPROT instance on the PID of the requesting system management process*/
   (PID, PID)
ENDNEWTYPE AT_PIDList;
```

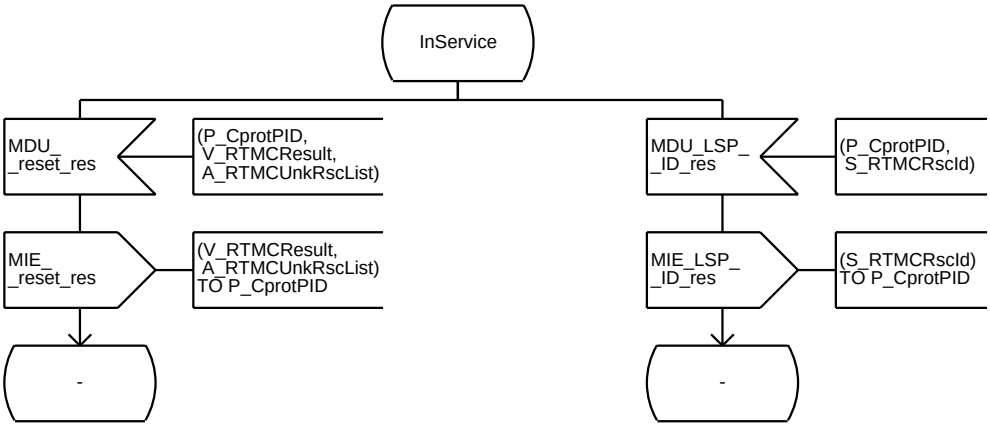
```
/*Primitive Data Declarations*/
DCL
A_RTMCBlRscList AT_RTMCBlRscList;
/**/
DCL
A_RTMCRscList AT_RscList;
/**/
DCL
A_RTMCUnkRscList AT_RscList;
/**/
DCL
S_RTMCRscId ST_RscId;
/**/
DCL
S_RTMCUnkRscId ST_RscId;
/**/
DCL
V_RTMCResult IT_RTMCResult;
/**/
DCL
V_IntTransId IT_IntTransId;
/**/
DCL
V_Result IT_Result;
/**/
DCL
V_ProtErrCause IT_RTMCProtErrCause;
/**/
DCL
P_StatusPID,    /*PID of AN_STATUS_MGT process*/
P_ResetPID,     /*PID of RESET_RSC process*/
P_ConsPID,      /*PID of AN_CONS process*/
P_LSPIdPID,     /*PID of CHECK_LSPID process*/
P_CprotPID,     /*PID of SN_CPROT process instance*/
P_TransId,      /*transaction identifier*/
P_ANPID         /*PID of AN_CPROT_MGR*/
PID;
```

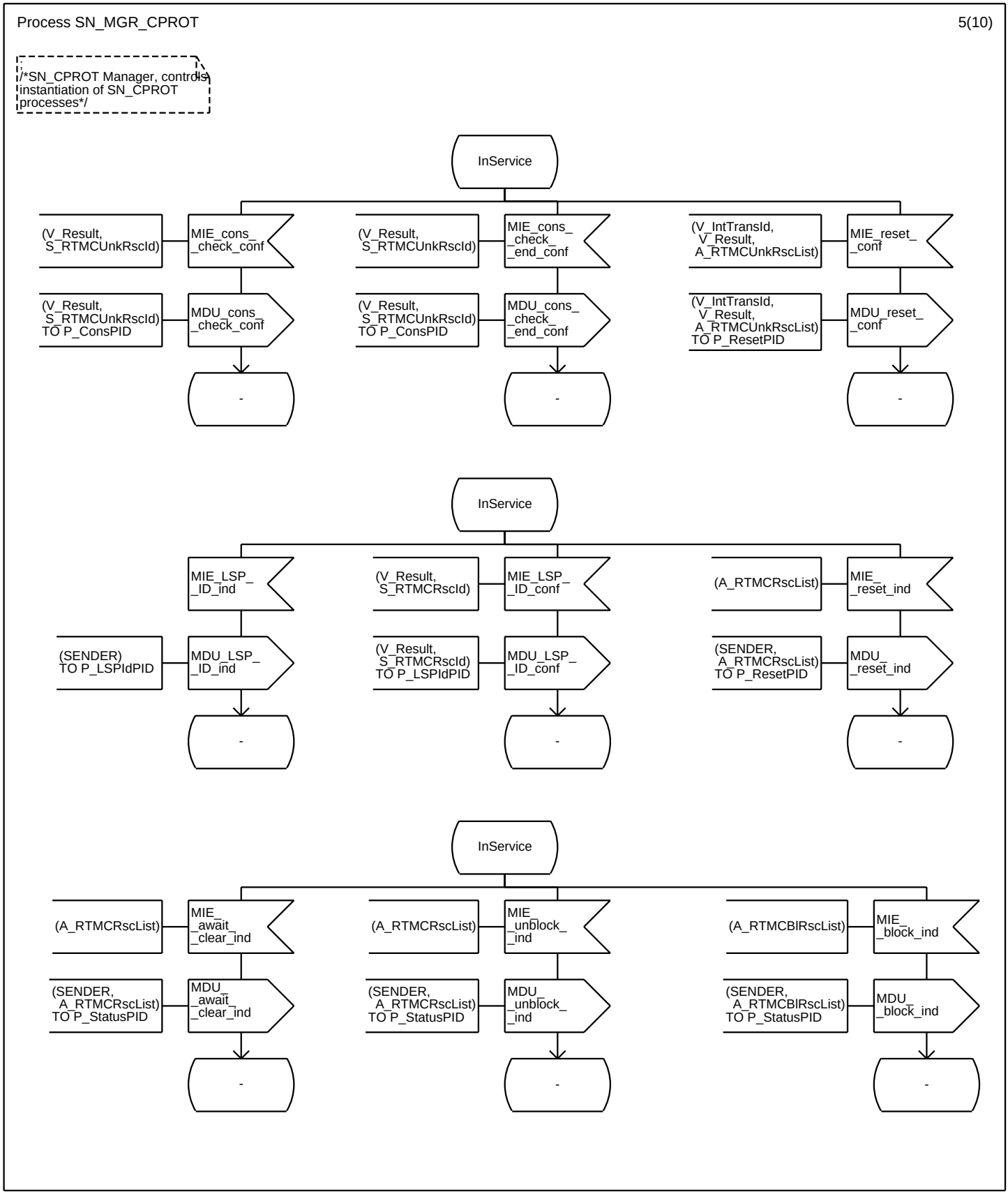

/*SN_CPROT Manager, controls instantiation of SN_CPROT processes*/
SIM_SN_ _started
OutOfservice
OutOfService
MDU_ _resrsc_ _prc_started
MDU_ _lspid_ _prc_started
MDU_ _cons_ _prc_started
MDU_ _status_ _prc_started
SIM_AN_ _started
P_ResetPID := SENDER
P_LspidPID := SENDER
P_ConsPID := SENDER
P_StatusPID := SENDER
P_ANPID := SENDER
-
-
-
-
-
*
MDU_start_ _traffic
MDU_stop_ _traffic
InService
To all relevant SN_CPROT instances
MIE_ _abort_req
OutOfService

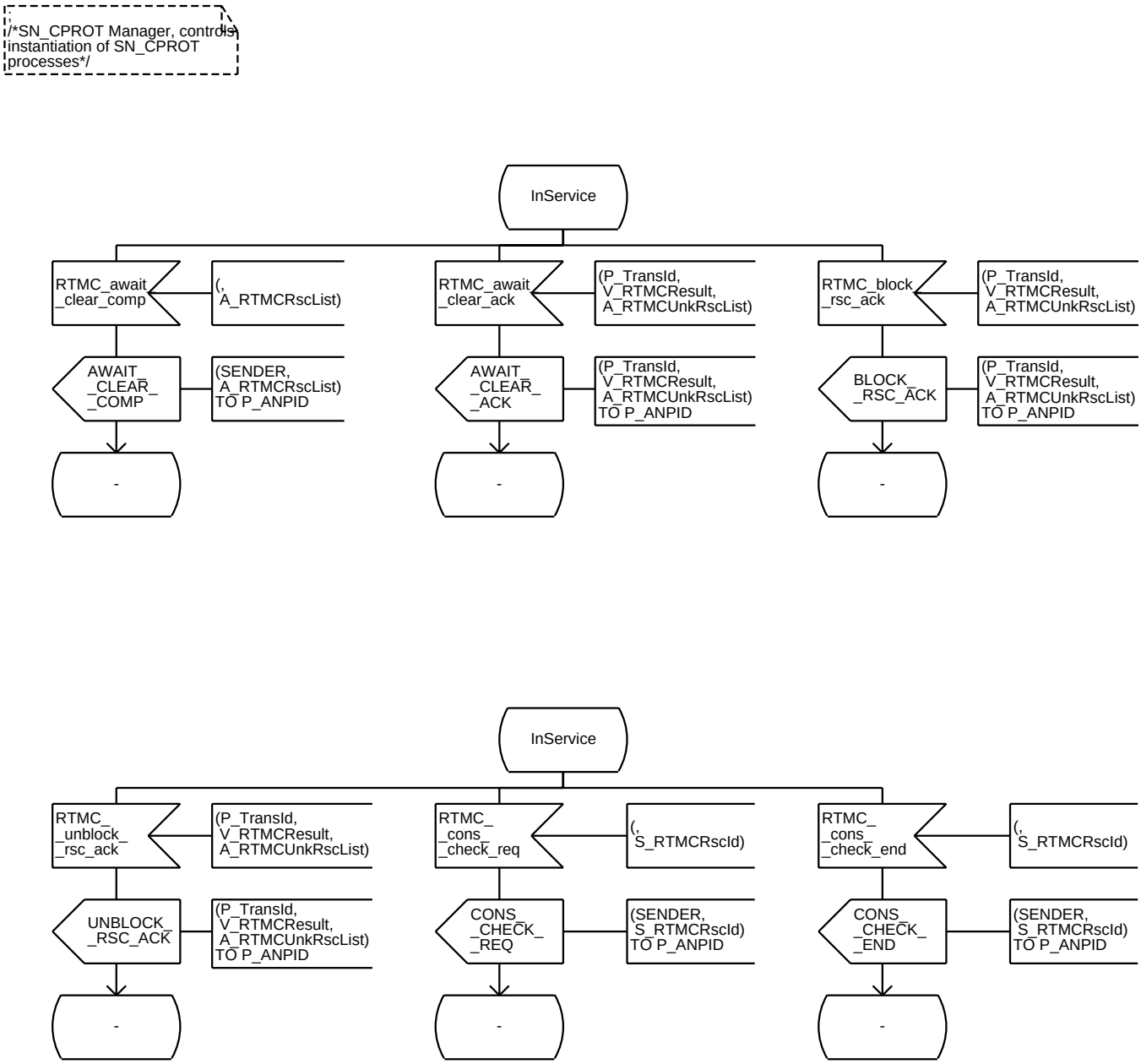

/*SN_CPROT Manager, controls instantiation of SN_CPROT processes*/
InService
MDU_cons_ _check_ _req
(S_RTMCRscId)
MDU_cons_ _check_ _end_req
(S_RTMCRscId)
SN_CPROT
SN_CPROT
MIE_cons_ _check_ _req
(S_RTMCRscId) TO OFFSPRING
MIE_cons_ _check_ _end_req
(S_RTMCRscId) TO OFFSPRING
-
-
InService
MDU_ _await_ _clear_res
(A_RTMCRscList)
MDU_LSP_ _ID_req
MDU_ _reset_req
(V_IntTransId, A_RTMCRscList)
SN_CPROT
SN_CPROT
SN_CPROT
MIE_ _await_ _clear_res
(A_RTMCRscList) TO OFFSPRING
MIE_LSP_ _ID_req
TO OFFSPRING
MIE_ _reset_req
(V_IntTransId, A_RTMCRscList) TO OFFSPRING
-
-
-

/*SN_CPROT Manager, controls
instantiation of SN_CPROT
processes*/

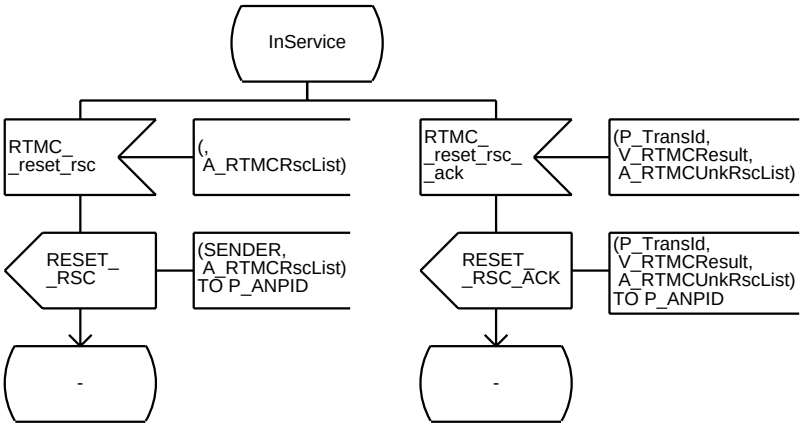

InService
MDU_
_block_res
(P_CprotPID,
V_RTMCResult,
A_RTMCUnkRscList)
MIE_
_block_res
(V_RTMCResult,
A_RTMCUnkRscList)
TO P_CprotPID
-
MDU_
unblock
_res
(P_CprotPID,
V_RTMCResult,
A_RTMCUnkRscList)
MIE_
unblock
_res
(V_RTMCResult,
A_RTMCUnkRscList)
TO P_CprotPID
-
MDU_
await
_clear_ack
(P_CprotPID,
V_RTMCResult,
A_RTMCUnkRscList)
MIE_
await
_clear_ack
(V_RTMCResult,
A_RTMCUnkRscList)
TO P_CprotPID
-

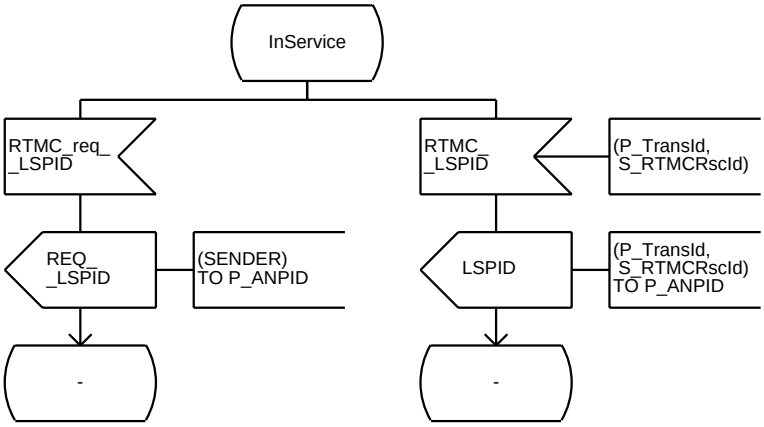

InService
MDU_
_reset_res
(P_CprotPID,
V_RTMCResult,
A_RTMCUnkRscList)
MIE_
_reset_res
(V_RTMCResult,
A_RTMCUnkRscList)
TO P_CprotPID
-
MDU_LSP_
_ID_res
(P_CprotPID,
S_RTMCRscId)
MIE_LSP_
_ID_res
(S_RTMCRscId)
TO P_CprotPID
-

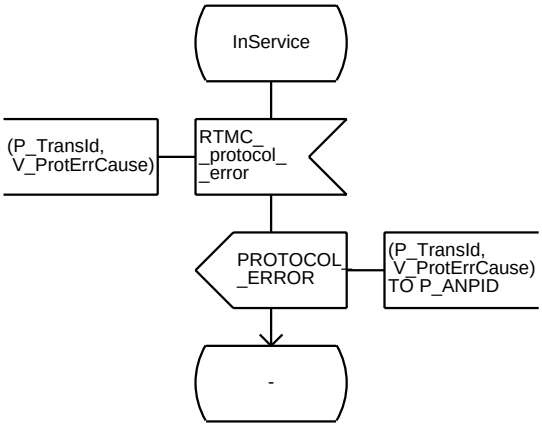

Process SN_MGR_CPROT
5(10)
/*SN_CPROT Manager, controls instantiation of SN_CPROT processes*/
InService
(V_Result, S_RTMCUnkRscId)
MIE_cons_ _check_conf
(V_Result, S_RTMCUnkRscId)
MIE_cons_ _check_ _end_conf
(V_IntTransId, V_Result, A_RTMCUnkRscList)
MIE_reset_ _conf
(V_Result, S_RTMCUnkRscId) TO P_ConsPID
MDU_cons_ _check_conf
(V_Result, S_RTMCUnkRscId) TO P_ConsPID
MDU_cons_ _check_ _end_conf
(V_IntTransId, V_Result, A_RTMCUnkRscList) TO P_ResetPID
MDU_reset_ _conf
-
-
-
InService
MIE_LSP_ _ID_ind
(V_Result, S_RTMCRscId)
MIE_LSP_ _ID_conf
(A_RTMCRscList)
MIE_ _reset_ind
(SENDER) TO P_LSPIdPID
MDU_LSP_ _ID_ind
(V_Result, S_RTMCRscId) TO P_LSPIdPID
MDU_LSP_ _ID_conf
(SENDER, A_RTMCRscList) TO P_ResetPID
MDU_ _reset_ind
-
-
-
InService
(A_RTMCRscList)
MIE_ _await_ _clear_ind
(A_RTMCRscList)
MIE_ _unblock_ _ind
(A_RTMCBlRscList)
MIE_ _block_ind
(SENDER, A_RTMCRscList) TO P_StatusPID
MDU_ _await_ _clear_ind
(SENDER, A_RTMCRscList) TO P_StatusPID
MDU_ _unblock_ _ind
(SENDER, A_RTMCBlRscList) TO P_StatusPID
MDU_ _block_ind
-
-
-

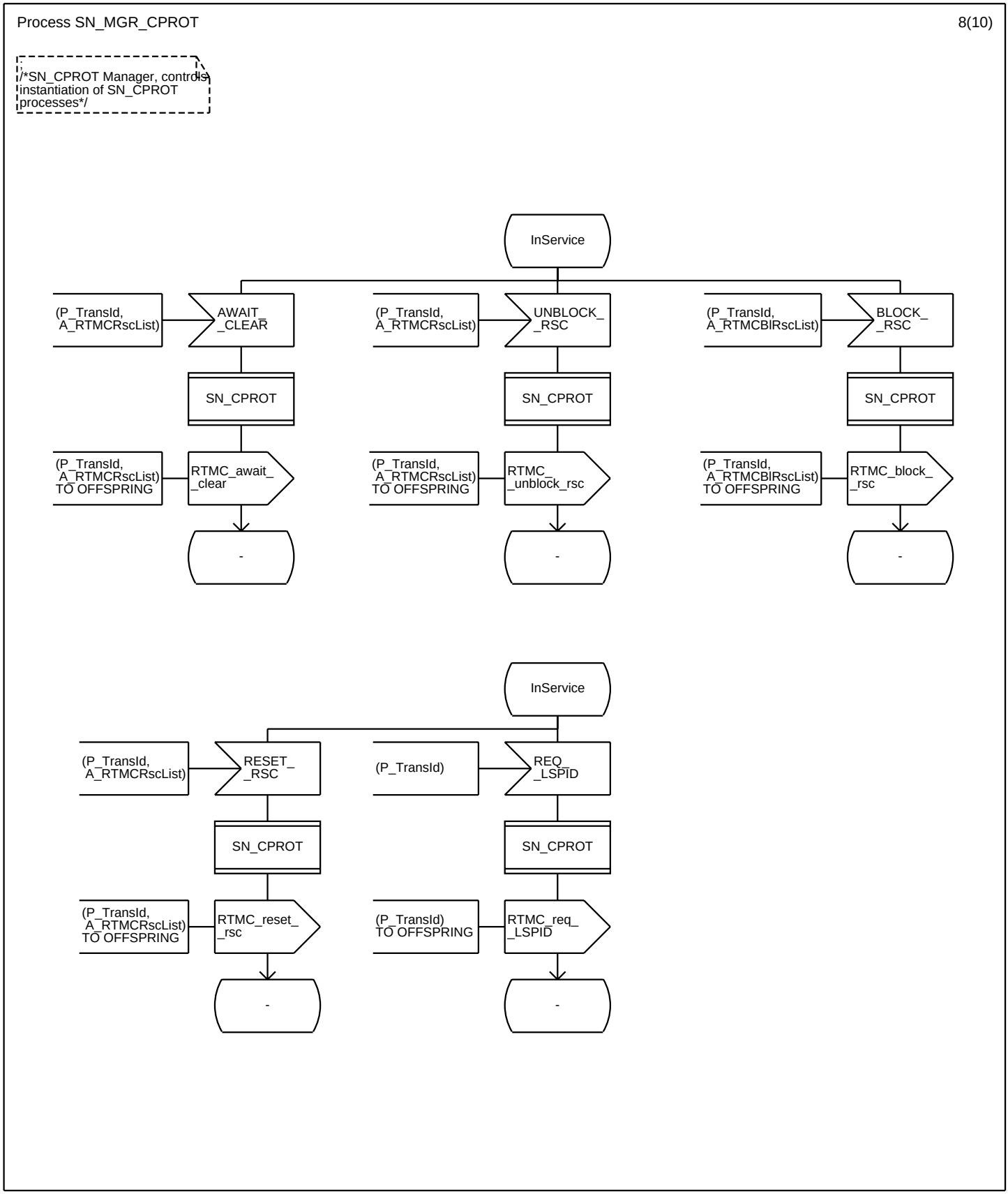

/*SN_CPROT Manager, controls instantiation of SN_CPROT processes*/
InService
RTMC_await _clear_comp
(, A_RTMCRscList)
AWAIT_ _CLEAR_ _COMP
(SENDER, A_RTMCRscList) TO P_ANPID
-
RTMC_await _clear_ack
(P_TransId, V_RTMCResult, A_RTMCUnkRscList)
AWAIT_ _CLEAR_ _ACK
(P_TransId, V_RTMCResult, A_RTMCUnkRscList) TO P_ANPID
-
RTMC_block _rsc_ack
(P_TransId, V_RTMCResult, A_RTMCUnkRscList)
BLOCK_ _RSC_ACK
(P_TransId, V_RTMCResult, A_RTMCUnkRscList) TO P_ANPID
-
InService
RTMC_ _unblock_ _rsc_ack
(P_TransId, V_RTMCResult, A_RTMCUnkRscList)
UNBLOCK_ _RSC_ACK
(P_TransId, V_RTMCResult, A_RTMCUnkRscList) TO P_ANPID
-
RTMC_ _cons_ _check_req
(, S_RTMCRscId)
CONS_ _CHECK_ _REQ
(SENDER, S_RTMCRscId) TO P_ANPID
-
RTMC_ _cons_ _check_end
(, S_RTMCRscId)
CONS_ _CHECK_ _END
(SENDER, S_RTMCRscId) TO P_ANPID
-

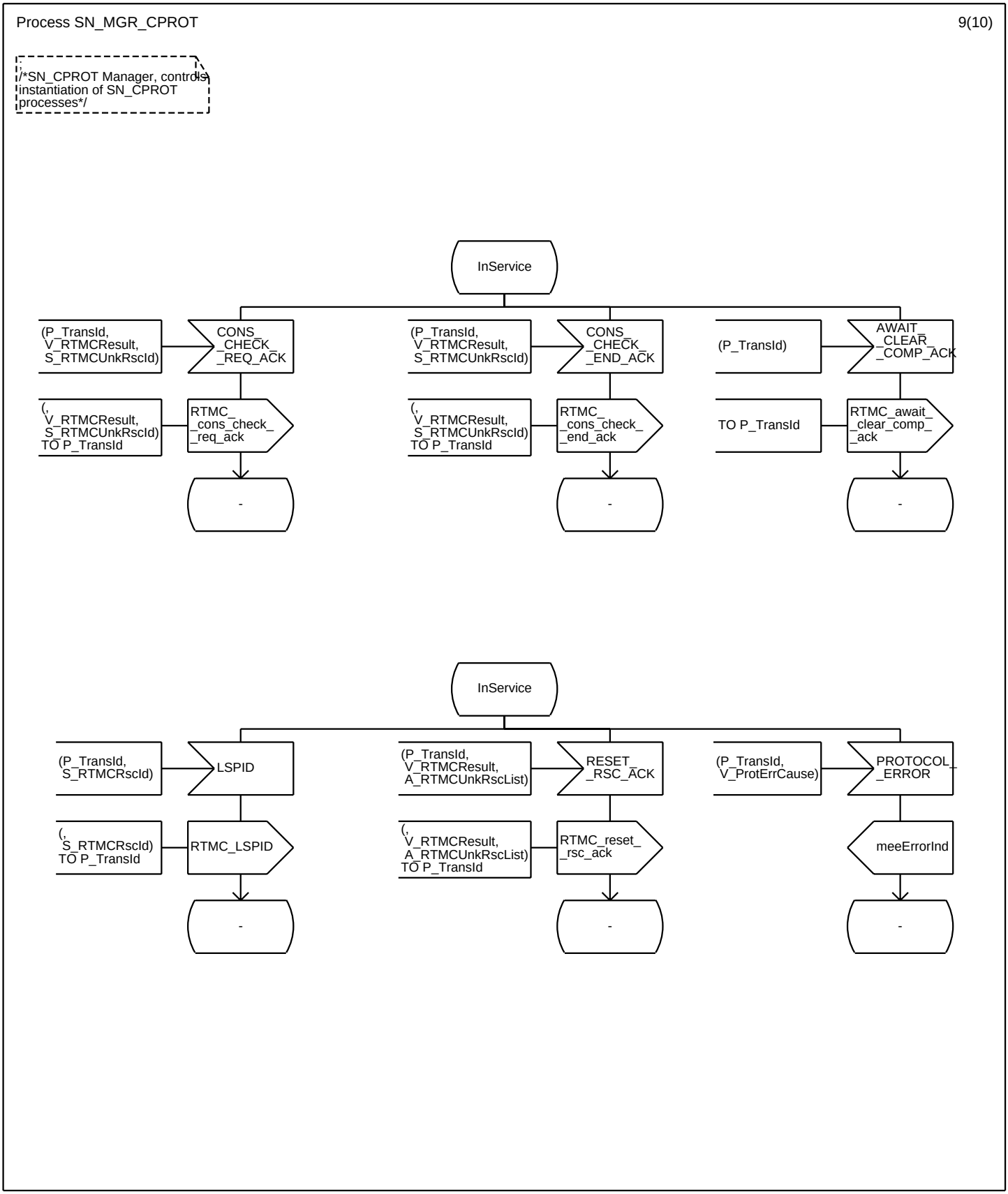
/*SN_CPROT Manager, controls instantiation of SN_CPROT processes*/

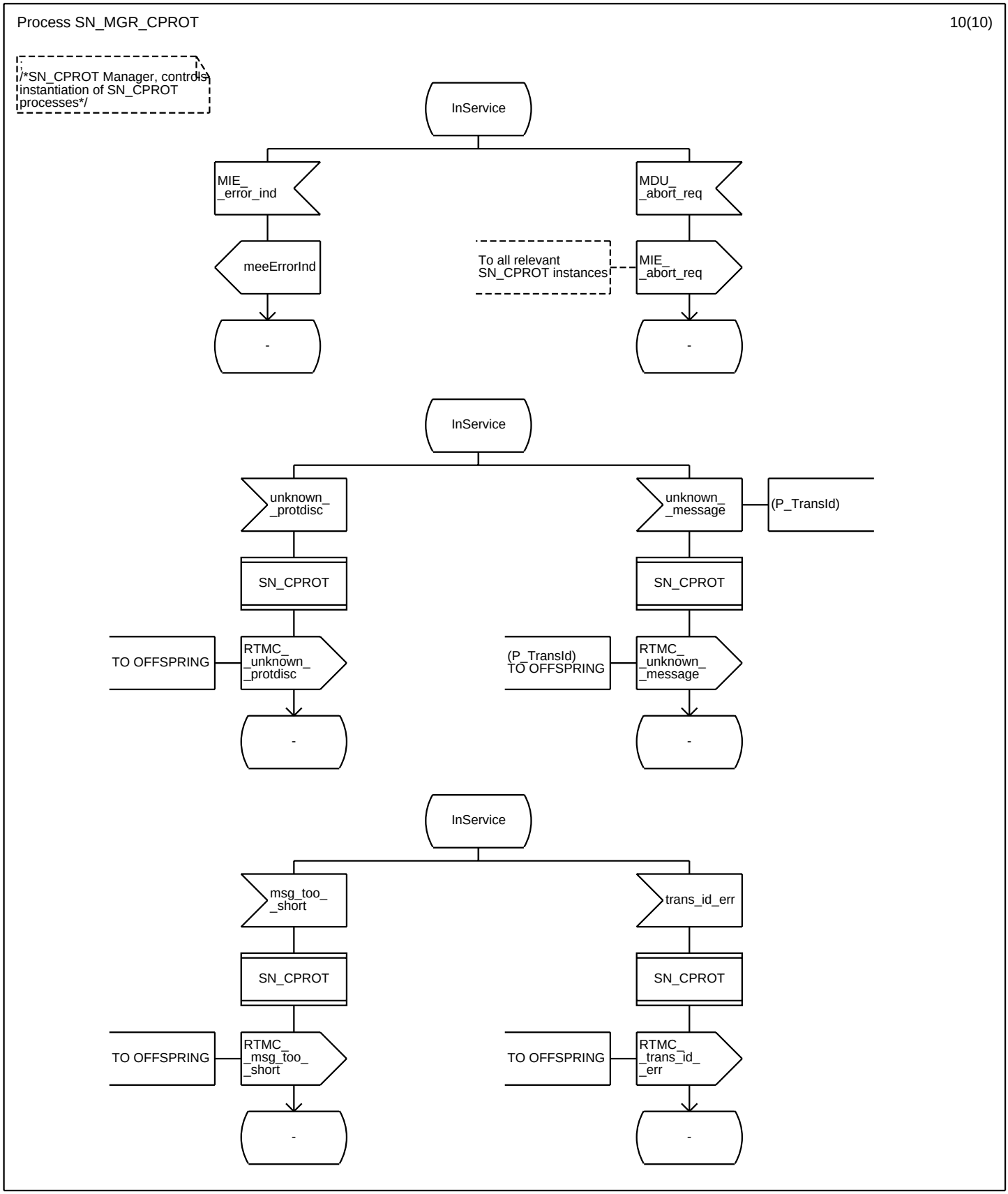







/*SN_CPROT Manager, controls instantiation of SN_CPROT processes*/
InService
(P_TransId, A_RTMCRscList)
AWAIT_ _CLEAR
SN_CPROT
(P_TransId, A_RTMCRscList) TO OFFSPRING
RTMC_await_ _clear
-
(P_TransId, A_RTMCRscList)
UNBLOCK_ _RSC
SN_CPROT
(P_TransId, A_RTMCRscList) TO OFFSPRING
RTMC_ _unblock_rsc
-
(P_TransId, A_RTMCBlRscList)
BLOCK_ _RSC
SN_CPROT
(P_TransId, A_RTMCBlRscList) TO OFFSPRING
RTMC_block_ _rsc
-
InService
(P_TransId, A_RTMCRscList)
RESET_ _RSC
SN_CPROT
(P_TransId, A_RTMCRscList) TO OFFSPRING
RTMC_reset_ _rsc
-
(P_TransId)
REQ_ _LSPID
SN_CPROT
(P_TransId) TO OFFSPRING
RTMC_req_ _LSPID
-


/*SN_CPROT Manager, controls instantiation of SN_CPROT processes*/
InService
(P_TransId, V_RTMCResult, S_RTMCUnkRscId)
CONS_ _CHECK_ _REQ_ACK
(, V_RTMCResult, S_RTMCUnkRscId) TO P_TransId
RTMC_ _cons_check_ _req_ack
-
(P_TransId, V_RTMCResult, S_RTMCUnkRscId)
CONS_ _CHECK_ _END_ACK
(, V_RTMCResult, S_RTMCUnkRscId) TO P_TransId
RTMC_ _cons_check_ _end_ack
-
(P_TransId)
AWAIT_ _CLEAR_ _COMP_ACK
TO P_TransId
RTMC_await_ _clear_comp_ _ack
-
InService
(P_TransId, S_RTMCRscId)
LSPID
(, S_RTMCRscId) TO P_TransId
RTMC_LSPID
-
(P_TransId, V_RTMCResult, A_RTMCUnkRscList)
RESET_ _RSC_ACK
(, V_RTMCResult, A_RTMCUnkRscList) TO P_TransId
RTMC_reset_ _rsc_ack
-
(P_TransId, V_ProtErrCause)
PROTOCOL_ _ERROR
meeErrorInd
-


Process SN_MGR_CPROT
10(10)
/*SN_CPROT Manager, controls instantiation of SN_CPROT processes*/
InService
MIE_ _error_ind
MDU_ _abort_req
meeErrorInd
To all relevant SN_CPROT instances
MIE_ _abort_req
-
-
InService
unknown_ _protdisc
unknown_ _message
(P_TransId)
SN_CPROT
SN_CPROT
TO OFFSPRING
RTMC_ _unknown_ _protdisc
(P_TransId) TO OFFSPRING
RTMC_ _unknown_ _message
-
-
InService
msg_too_ _short
trans_id_err
SN_CPROT
SN_CPROT
TO OFFSPRING
RTMC_ _msg_too_ _short
TO OFFSPRING
RTMC_ _trans_id_ _err
-
-

```
;
/*SN_CPROT is responsible for sending/
receiving RTMC messages*/
```

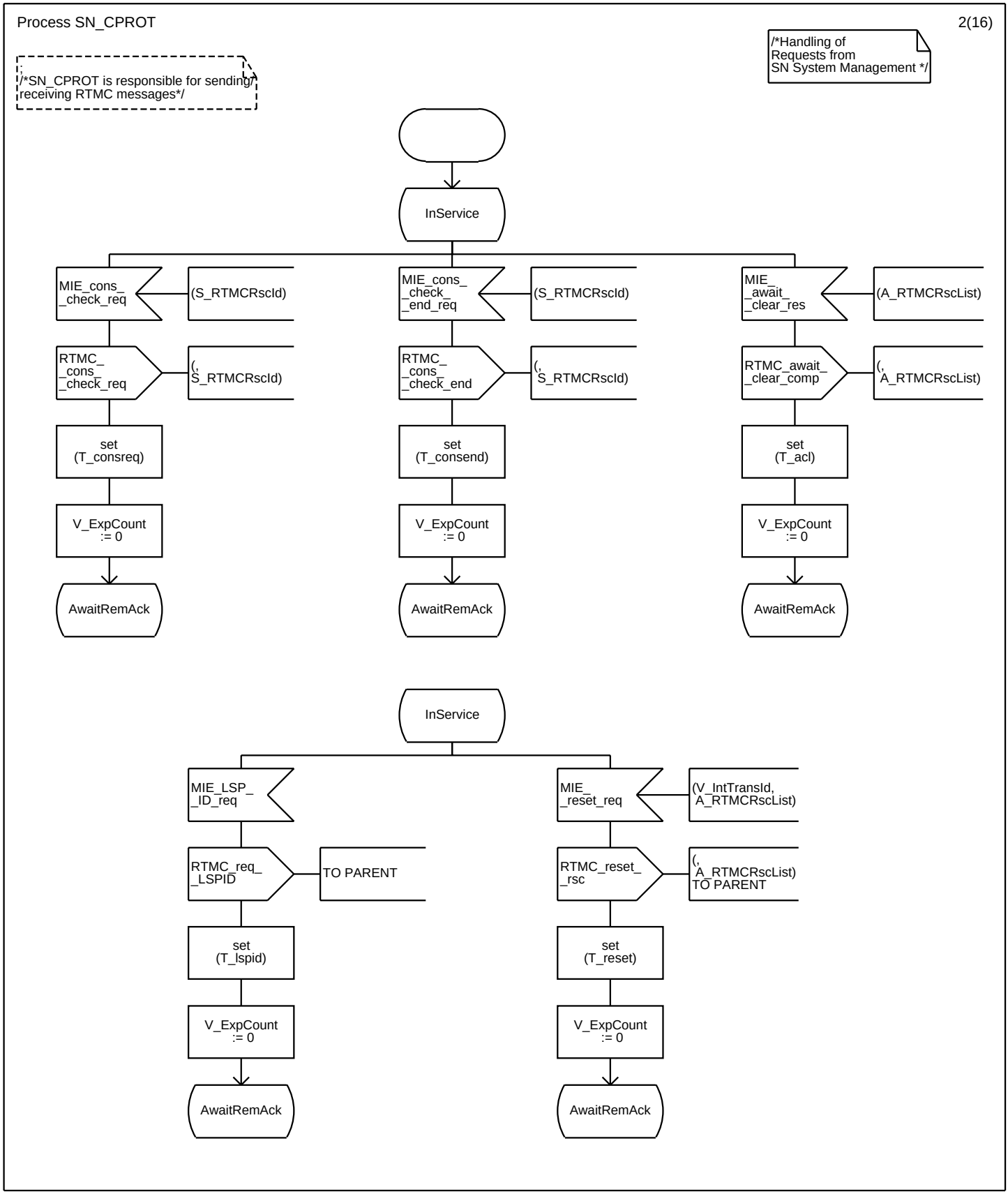
```
/* Definintions and declarations for
SN_CPROT */
```

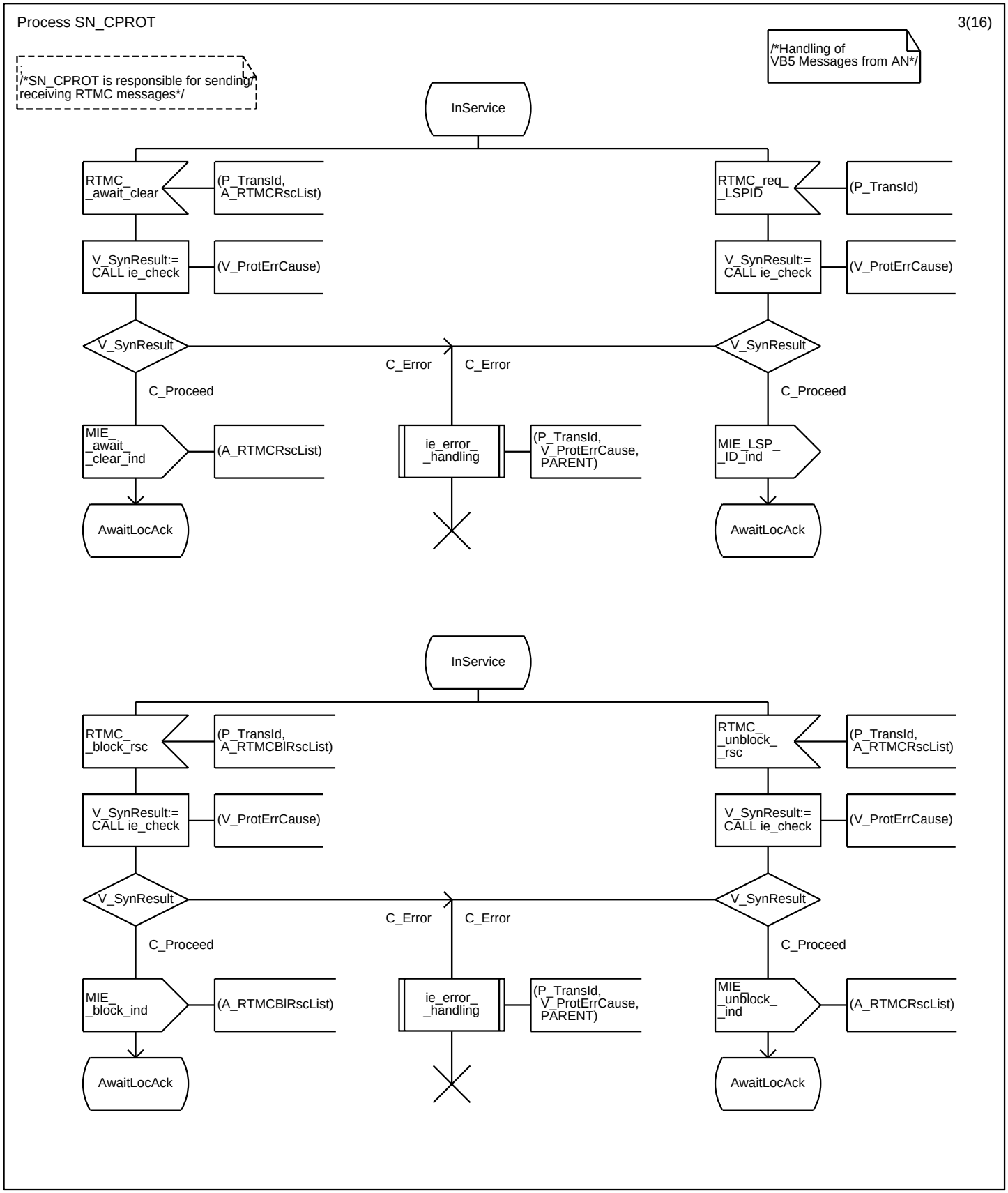
```
/* State descriptions

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

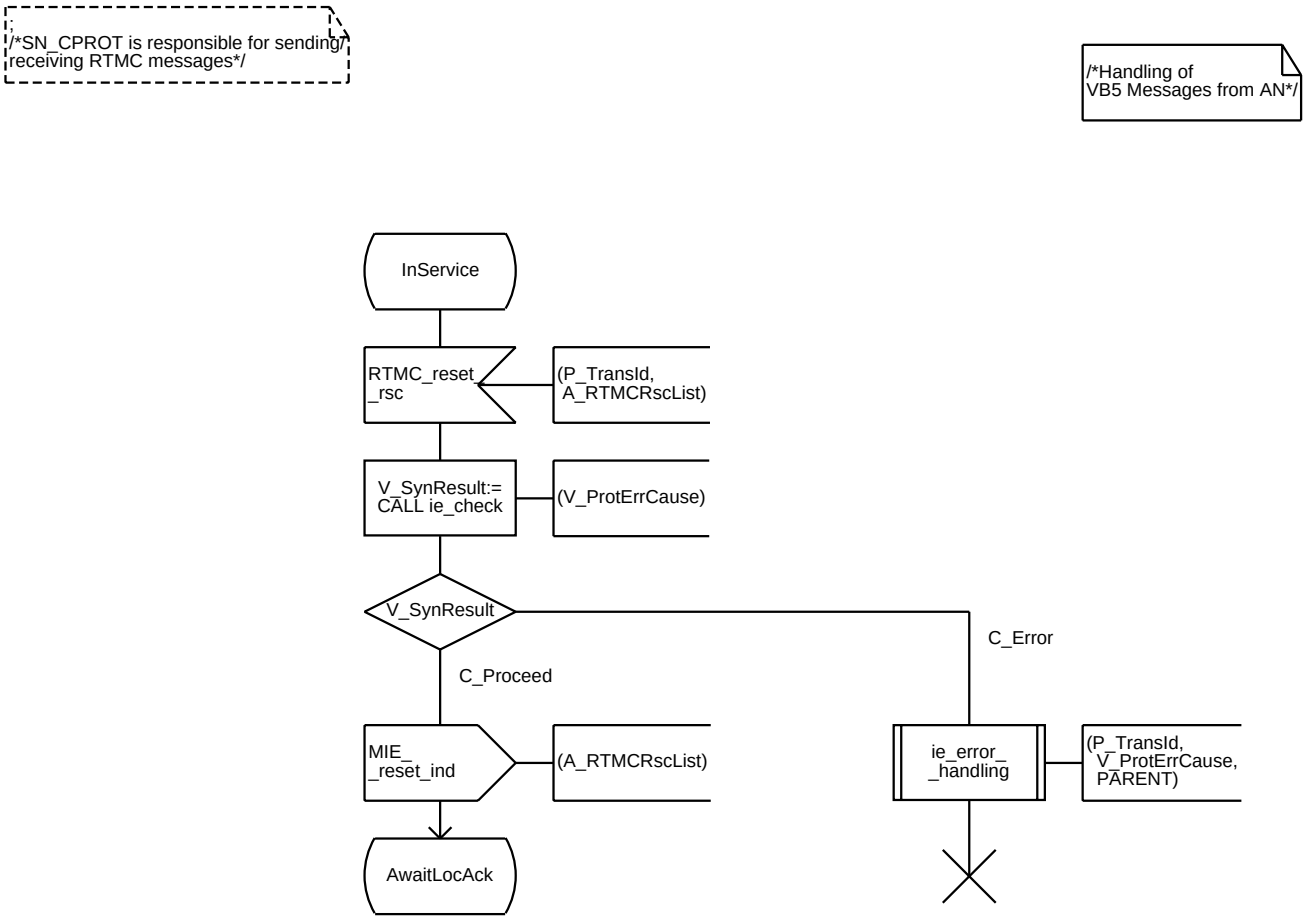
```
/* Timerdefinitions */
TIMER
T_acl       := 1   /* Default value for AWAIT_CLEAR_COMP supervision timer is 1 sec, Tolerance +/- 10%  */,
T_lspid     := 1   /* Default value for REQ_LSPID supervision timer is 1 sec, Tolerance +/- 10%   */,
T_consreq := 10 /* Default value for CONS_CHECK_REQ supervision timer is 10 sec, Tolerance +/- 10%   */,
T_consend := 10 /* Default value for CONS_CHECK_END supervision timer is 10 sec, Tolerance +/- 10%   */,
T_reset     := 60 /* Default value for RESET_RSC supervision timer is 60 sec, Tolerance +/- 10%   */;
```

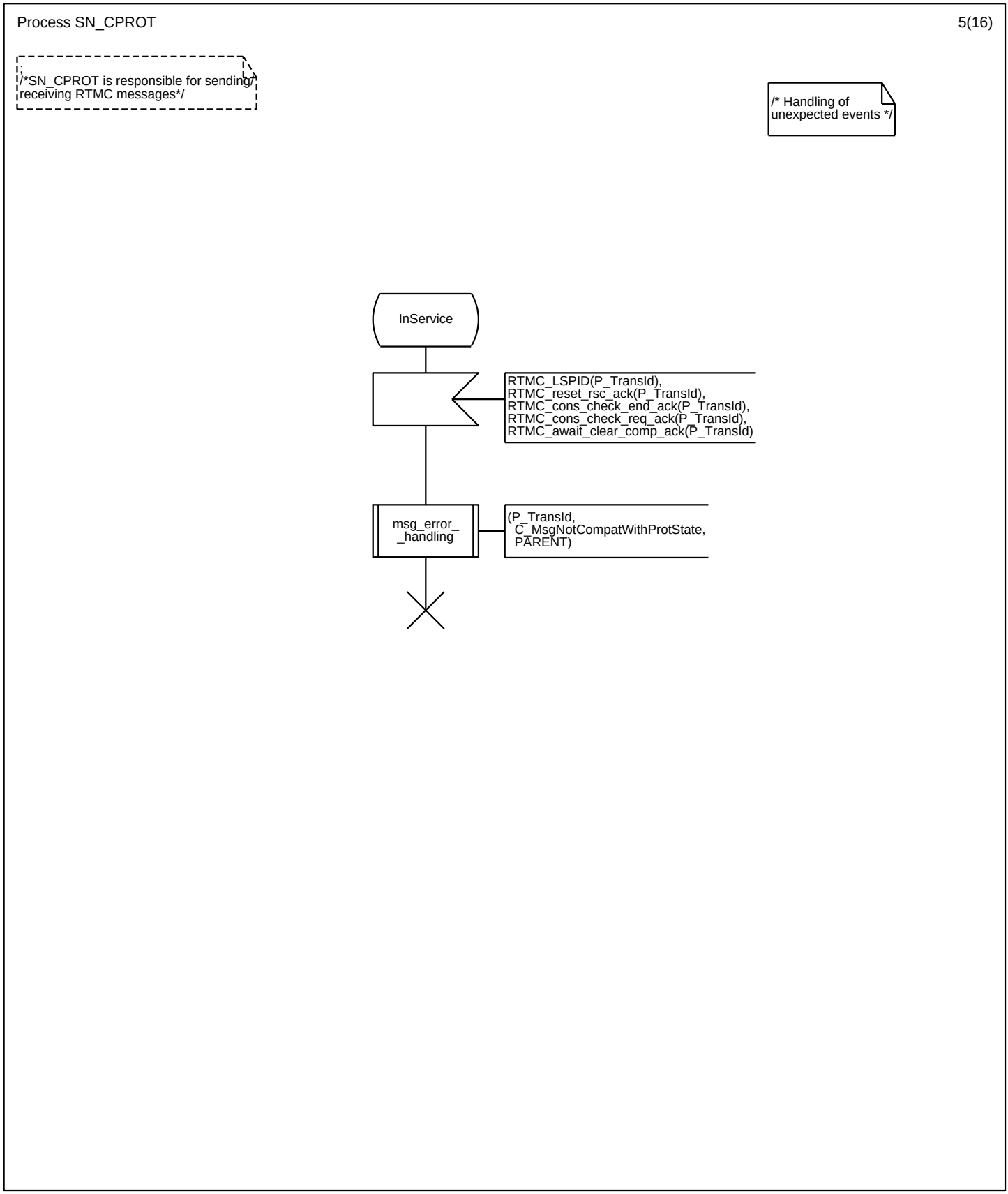
```
/* SN_CPROT internal variables and constants*/
/**/
/* timer handling */
DCL V_ExpCount NATURAL;  /*number of actual message repetitions*/
/*possible values*/
SYNONYM C_ExpMax INTEGER = 1;        /*maximum number of message repetitions is 1*/
/**/
/*syntax check handling*/
DCL V_SynResult INTEGER;  /*syntax check result, provided by procedure IE_CHECK*/
/*possible values*/
SYNONYM C_Proceed INTEGER =1;     /*possible result of syntax check: proceed with message processing*/
SYNONYM C_Error INTEGER =2;     /*possible result of syntax check: message with error condition/
```

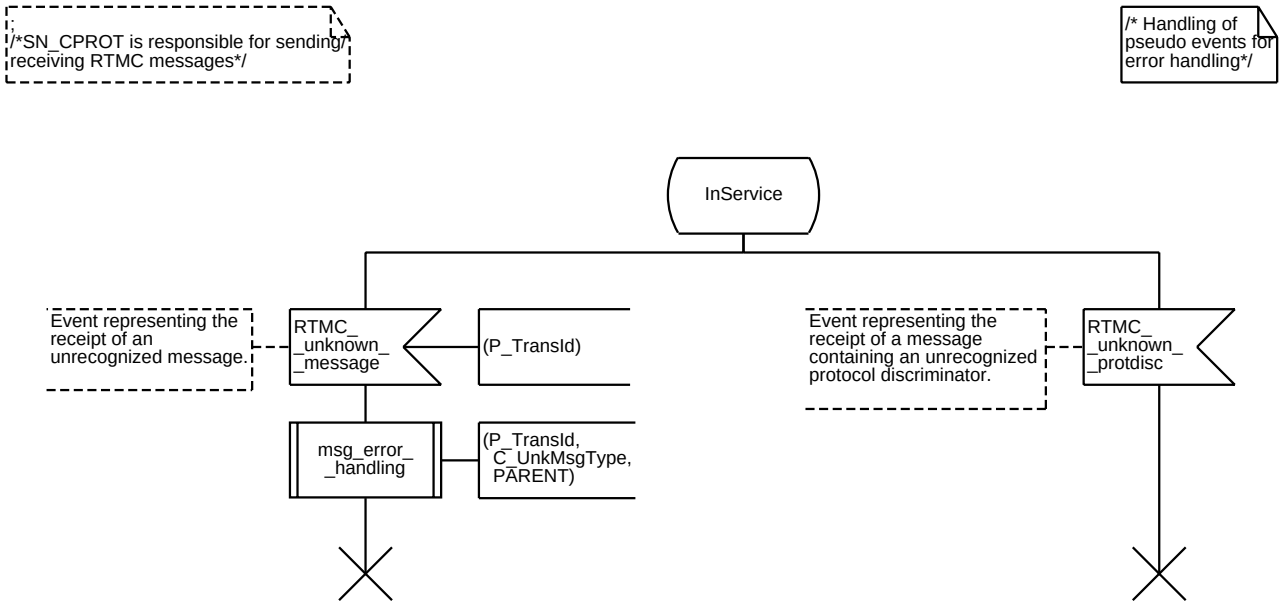
```
/*Signal data declarations*/
DCL
S_RTMCRscId ST_RscId; /*Resource Identifier Information Element*/
/**/
DCL
S_RTMCUnkRscId ST_RscId; /*Resource Identifier Information Element  unknown by the peer side*/
/**/
DCL
A_RTMCRscList AT_RscList;  /*Array of Resource Identifier Information Elements*/
/**/
DCL
A_RTMCBlRscList AT_RTMCBlRscList;  /*Array of Blocked Resource Identifier Information Elements*/
/**/
DCL
A_RTMCUnkRscList AT_RscList;  /*Array of Resource Identifier Information Elements unknown by the peer side*/
/**/
DCL
V_IntTransId IT_IntTransId;
/**/
DCL
V_RTMCResult IT_RTMCResult, /* Result from/to peer side */
V_Result         IT_Result;          /* Result to local side*/
/**/
DCL
P_TransId PID; /*transaction identifier towards AN*/
/**/
DCL
V_ProtErrCause   IT_RTMCProtErrCause;   /*Error Cause in case of reject*/
```

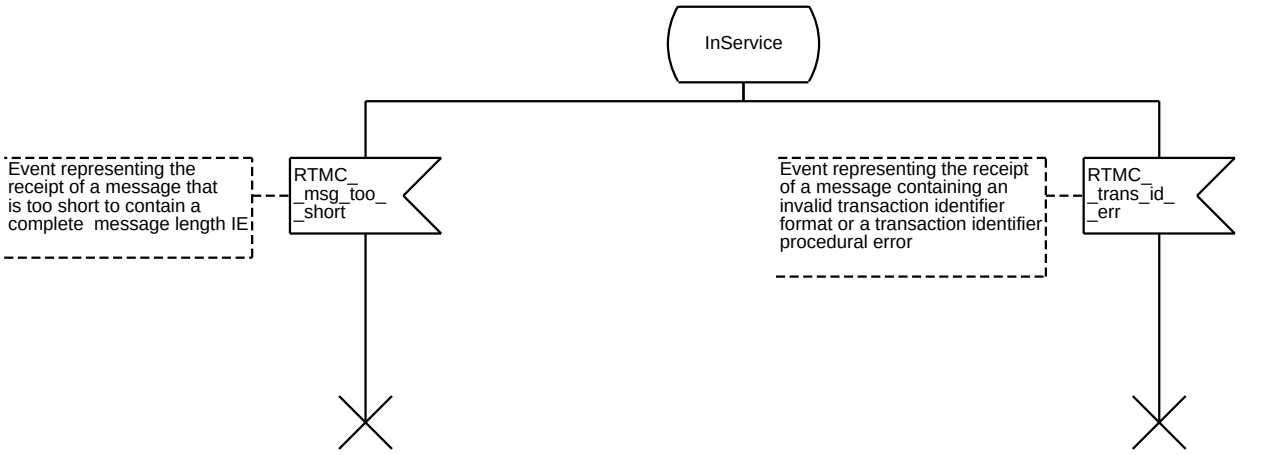

Process SN_CPROT
2(16)
/*Handling of Requests from SN System Management */
/*SN_CPROT is responsible for sending/receiving RTMC messages*/
InService
MIE_cons_ _check_req
(S_RTMCRscId)
RTMC_ _cons_ _check_req
(, S_RTMCRscId)
set (T_consreq)
V_ExpCount := 0
AwaitRemAck
MIE_cons_ _check_ _end_req
(S_RTMCRscId)
RTMC_ _cons_ _check_end
(, S_RTMCRscId)
set (T_consend)
V_ExpCount := 0
AwaitRemAck
MIE_ _await_ _clear_res
(A_RTMCRscList)
RTMC_await_ _clear_comp
(, A_RTMCRscList)
set (T_acl)
V_ExpCount := 0
AwaitRemAck
InService
MIE_LSP_ _ID_req
RTMC_req_ _LSPID
TO PARENT
set (T_lspid)
V_ExpCount := 0
AwaitRemAck
MIE_ _reset_req
(V_IntTransId, A_RTMCRscList)
RTMC_reset_ _rsc
(, A_RTMCRscList) TO PARENT
set (T_reset)
V_ExpCount := 0
AwaitRemAck

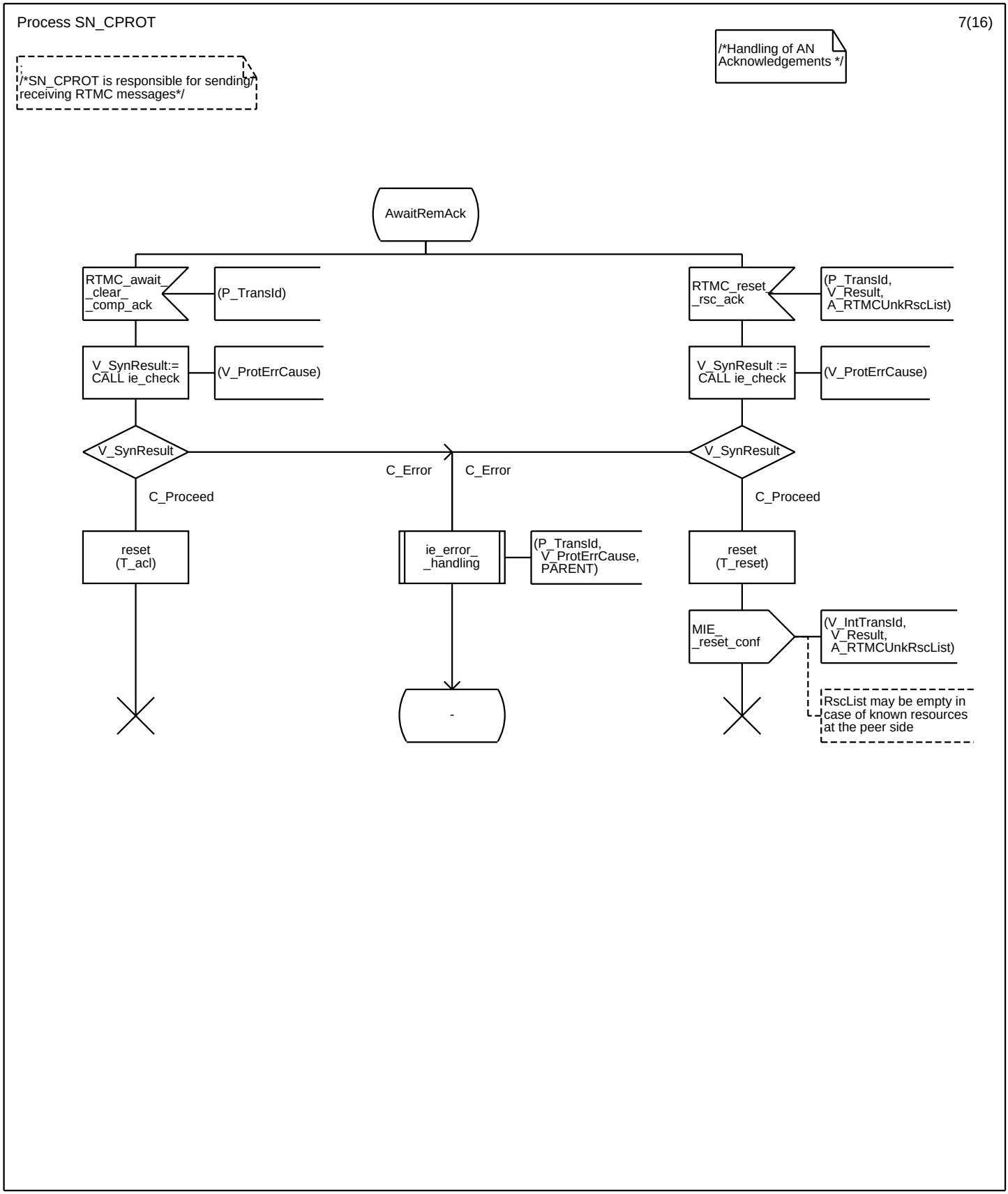

Process SN_CPROT
3(16)
/*Handling of
VB5 Messages from AN*/
/*SN_CPROT is responsible for sending/
receiving RTMC messages*/
InService
RTMC_
_await_clear
(P_TransId,
A_RTMCRscList)
RTMC_req_
_LSPID
(P_TransId)
V_SynResult:=
CALL ie_check
(V_ProtErrCause)
V_SynResult:=
CALL ie_check
(V_ProtErrCause)
V_SynResult
C_Error
C_Error
V_SynResult
C_Proceed
C_Proceed
MIE_
await
_clear_ind
(A_RTMCRscList)
ie_error_
_handling
(P_TransId,
V_ProtErrCause,
PARENT)
MIE_LSP_
_ID_ind
AwaitLocAck
AwaitLocAck
InService
RTMC_
_block_rsc
(P_TransId,
A_RTMCBlRscList)
RTMC_
unblock
_rsc
(P_TransId,
A_RTMCRscList)
V_SynResult:=
CALL ie_check
(V_ProtErrCause)
V_SynResult:=
CALL ie_check
(V_ProtErrCause)
V_SynResult
C_Error
C_Error
V_SynResult
C_Proceed
C_Proceed
MIE_
_block_ind
(A_RTMCBlRscList)
ie_error_
_handling
(P_TransId,
V_ProtErrCause,
PARENT)
MIE_
unblock
_ind
(A_RTMCRscList)
AwaitLocAck
AwaitLocAck

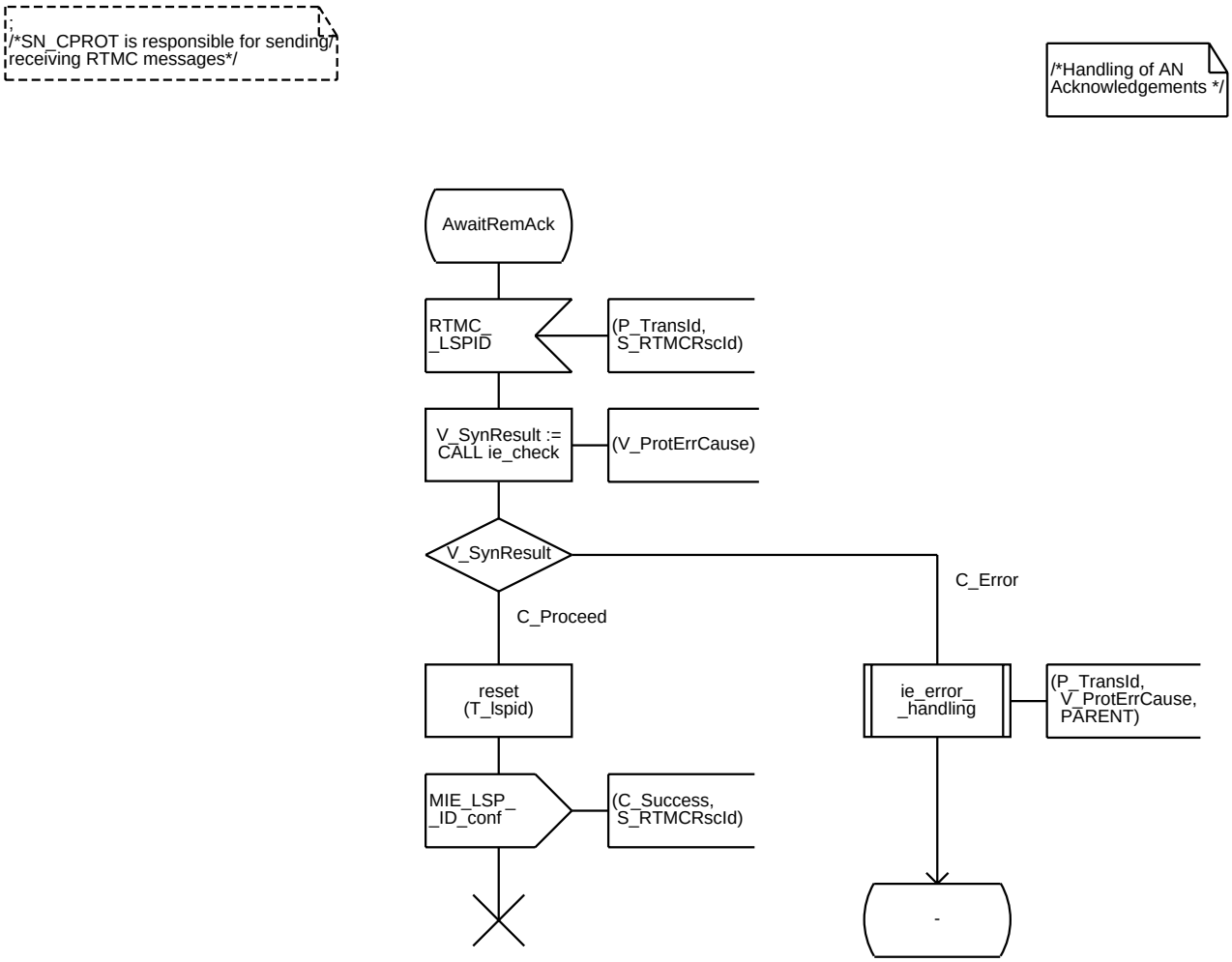

/*SN_CPROT is responsible for sending/receiving RTMC messages*/
/*Handling of VB5 Messages from AN*/
InService
RTMC_reset_rsc
(P_TransId, A_RTMCRscList)
V_SynResult:= CALL ie_check
(V_ProtErrCause)
V_SynResult
C_Error
C_Proceed
MIE_reset_ind
(A_RTMCRscList)
ie_error_handling
(P_TransId, V_ProtErrCause, PARENT)
AwaitLocAck

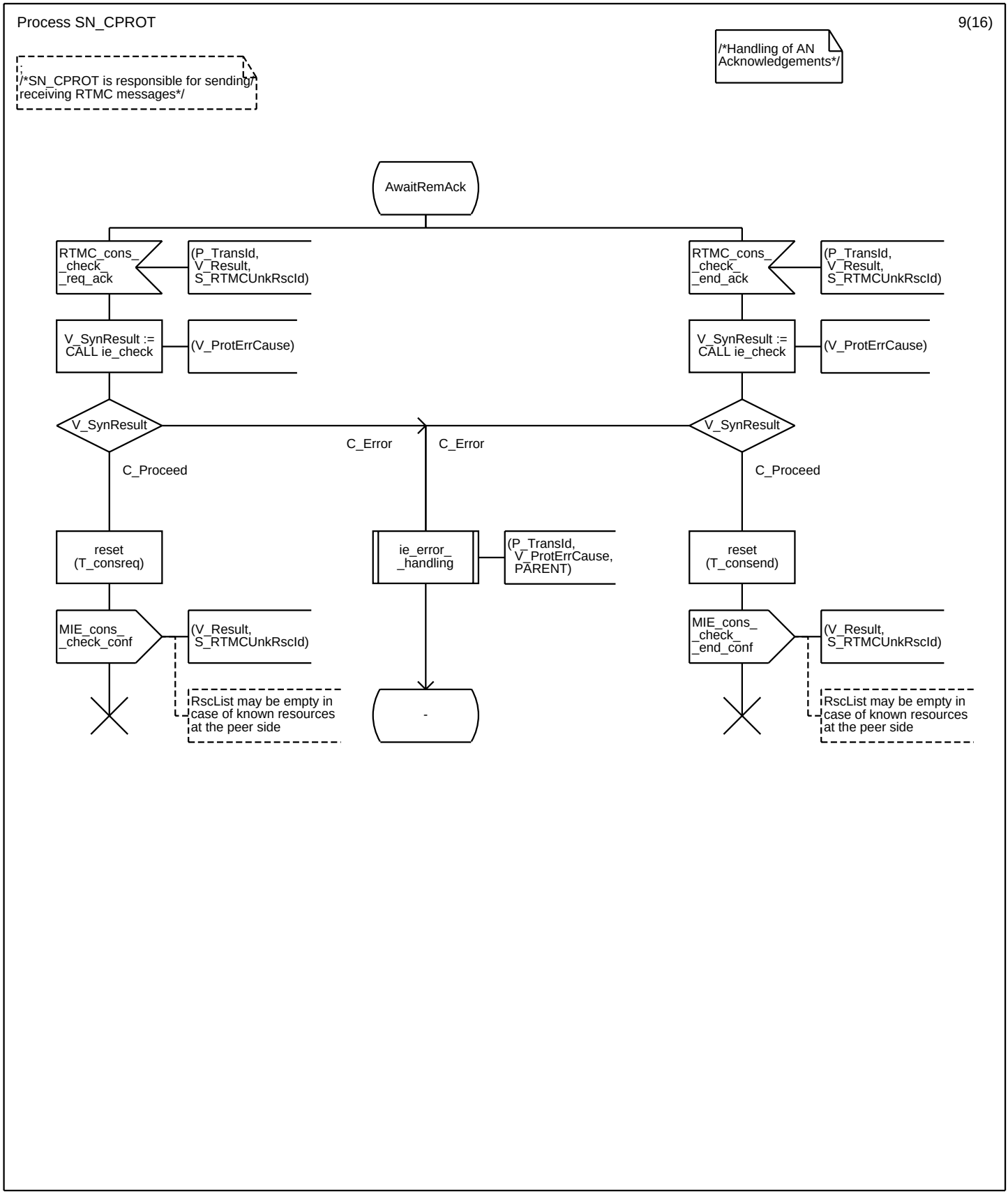

/*SN_CPROT is responsible for sending/receiving RTMC messages*/
/* Handling of unexpected events */
InService
RTMC_LSPID(P_TransId),
RTMC_reset_rsc_ack(P_TransId),
RTMC_cons_check_end_ack(P_TransId),
RTMC_cons_check_req_ack(P_TransId),
RTMC_await_clear_comp_ack(P_TransId)
msg_error_
_handling
(P_TransId,
C_MsgNotCompatWithProtState,
PARENT)

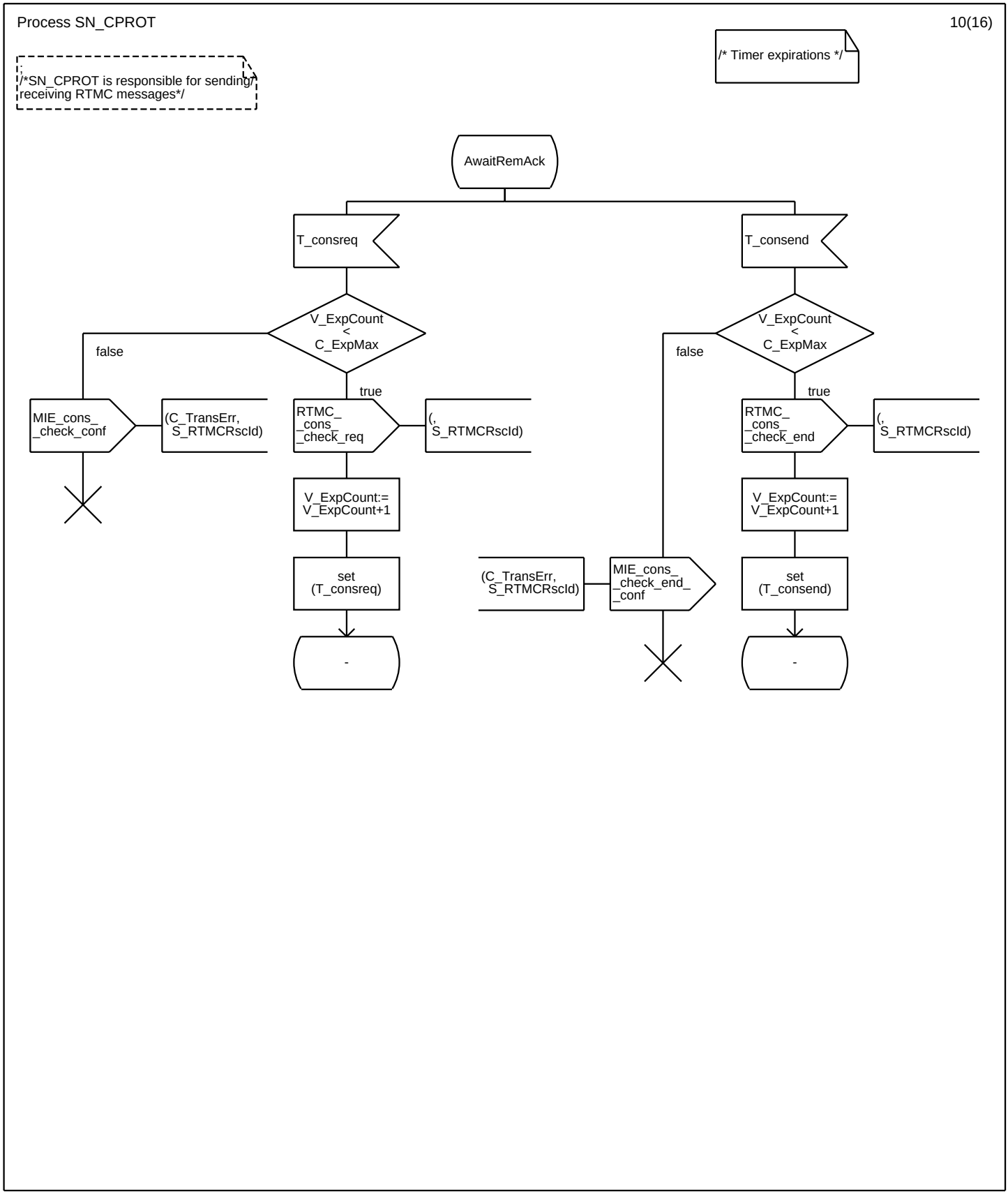

/*SN_CPROT is responsible for sending/receiving RTMC messages*/
/* Handling of pseudo events for error handling*/
InService
Event representing the receipt of an unrecognized message.
RTMC_ _unknown_ _message
(P_TransId)
msg_error_ _handling
(P_TransId, C_UnkMsgType, PARENT)
Event representing the receipt of a message containing an unrecognized protocol discriminator.
RTMC_ _unknown_ _protdisc

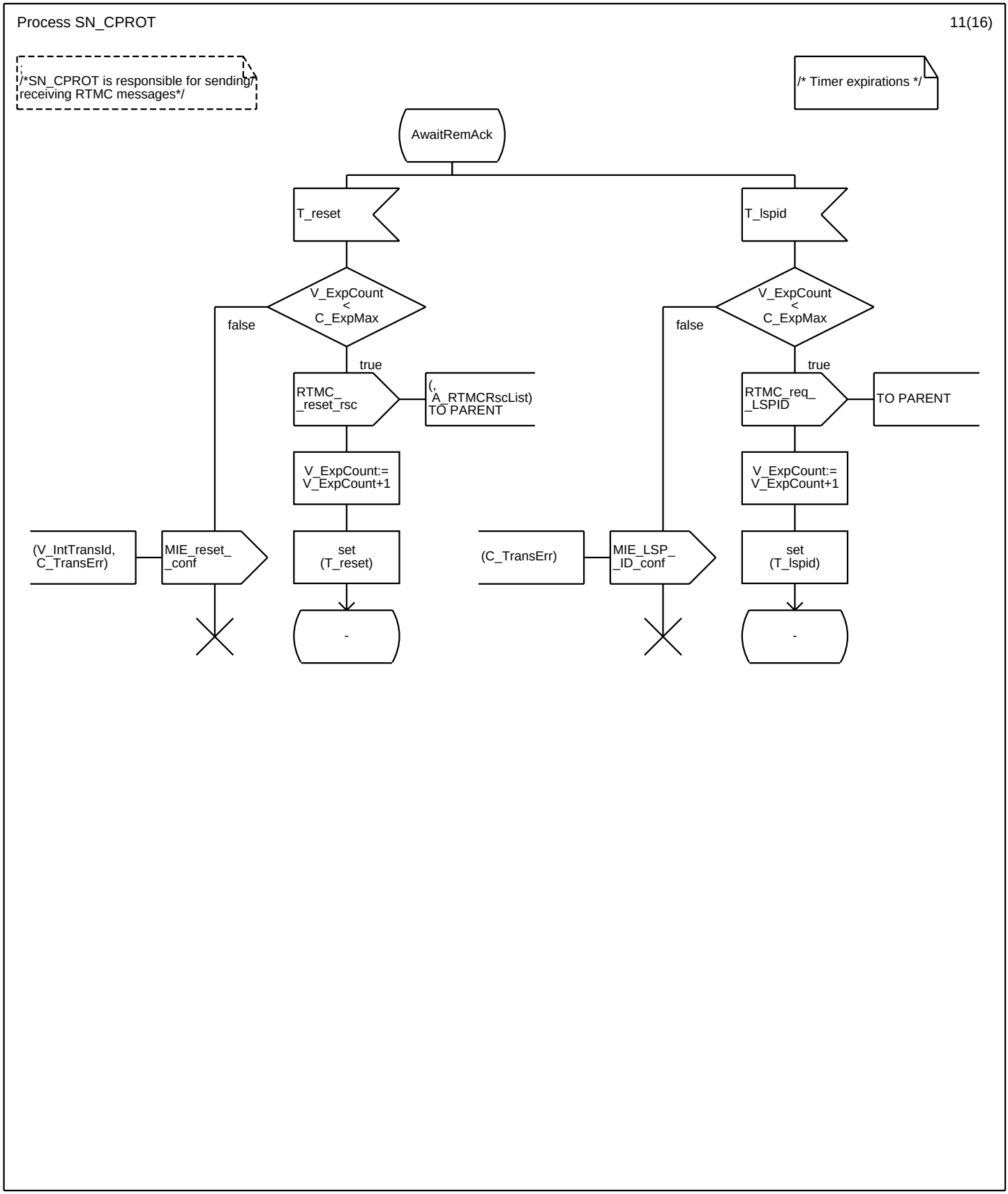

InService
Event representing the receipt of a message that is too short to contain a complete message length IE
RTMC_ _msg_too_ _short
Event representing the receipt of a message containing an invalid transaction identifier format or a transaction identifier procedural error
RTMC_ _trans_id_ _err

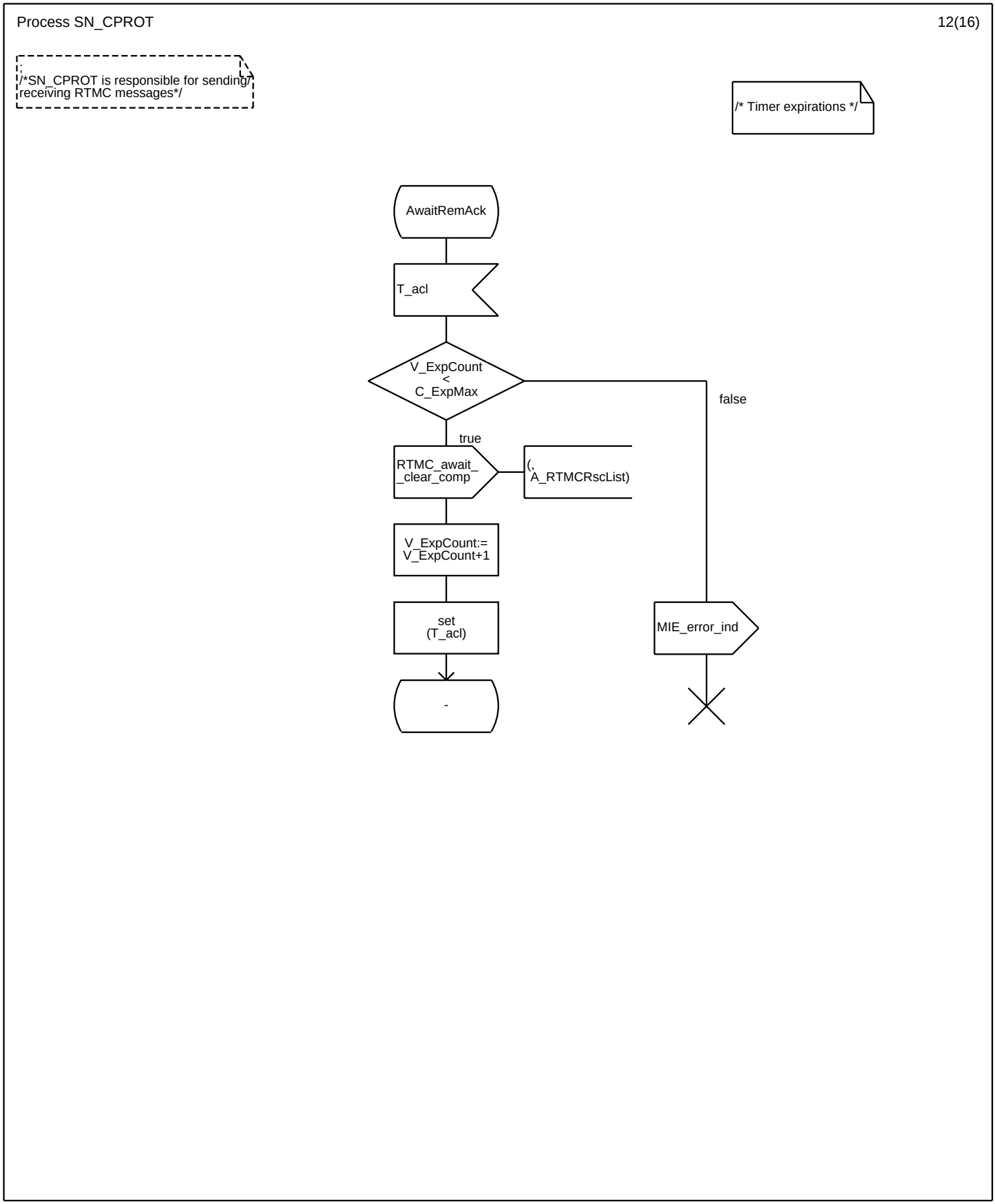

Process SN_CPROT
7(16)
/*SN_CPROT is responsible for sending/receiving RTMC messages*/
/*Handling of AN Acknowledgements */
AwaitRemAck
RTMC_await_clear_comp_ack
(P_TransId)
RTMC_reset_rsc_ack
(P_TransId, V_Result, A_RTMCUnkRscList)
V_SynResult:= CALL ie_check
(V_ProtErrCause)
V_SynResult := CALL ie_check
(V_ProtErrCause)
V_SynResult
C_Error
C_Error
V_SynResult
C_Proceed
C_Proceed
reset (T_acl)
ie_error_handling
(P_TransId, V_ProtErrCause, PARENT)
reset (T_reset)
MIE_reset_conf
(V_IntTransId, V_Result, A_RTMCUnkRscList)
RscList may be empty in case of known resources at the peer side
-

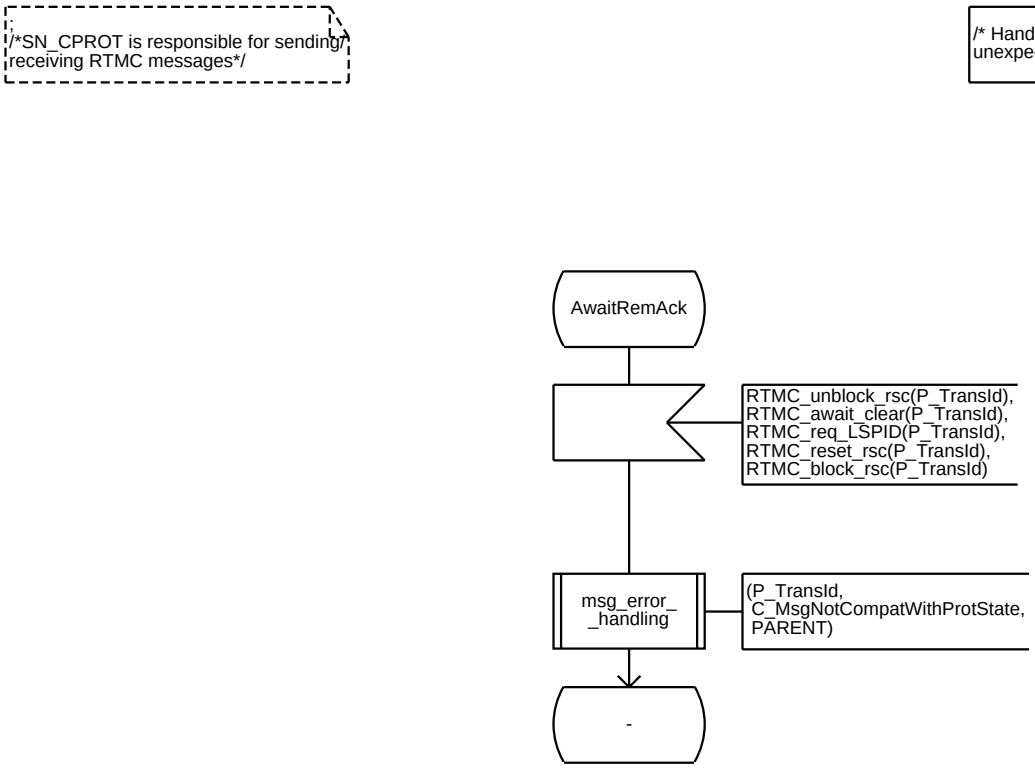

/*SN_CPROT is responsible for sending/receiving RTMC messages*/
/*Handling of AN Acknowledgements */
AwaitRemAck
RTMC_ _LSPID
(P_TransId, S_RTMCRscId)
V_SynResult := CALL ie_check
(V_ProtErrCause)
V_SynResult
C_Error
C_Proceed
reset (T_lspid)
ie_error_ _handling
(P_TransId, V_ProtErrCause, PARENT)
MIE_LSP_ _ID_conf
(C_Success, S_RTMCRscId)
-

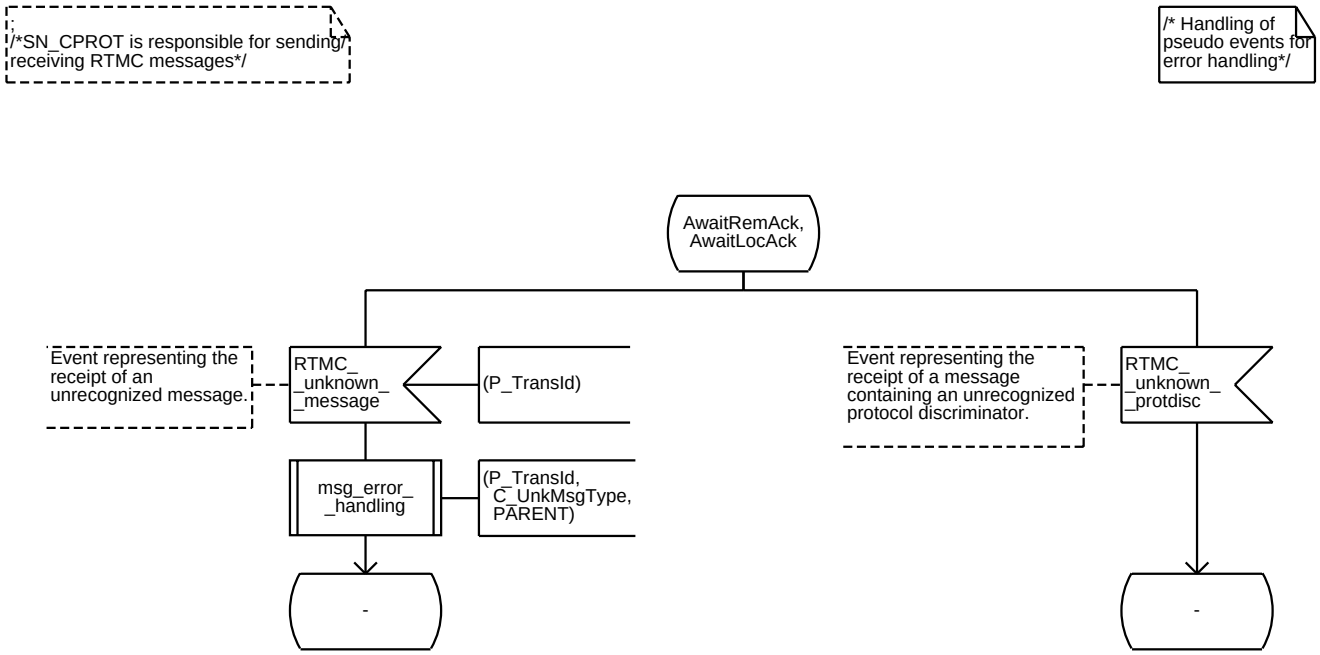

Process SN_CPROT
9(16)
/*SN_CPROT is responsible for sending/receiving RTMC messages*/
/*Handling of AN Acknowledgements*/
AwaitRemAck
RTMC_cons_ _check_ _req_ack
(P_TransId, V_Result, S_RTMCUnkRscId)
V_SynResult := CALL ie_check
(V_ProtErrCause)
V_SynResult
C_Proceed
reset (T_consreq)
MIE_cons_ _check_conf
(V_Result, S_RTMCUnkRscId)
RscList may be empty in case of known resources at the peer side
C_Error
C_Error
ie_error_ _handling
(P_TransId, V_ProtErrCause, PARENT)
-
RTMC_cons_ _check_ _end_ack
(P_TransId, V_Result, S_RTMCUnkRscId)
V_SynResult := CALL ie_check
(V_ProtErrCause)
V_SynResult
C_Proceed
reset (T_consend)
MIE_cons_ _check_ _end_conf
(V_Result, S_RTMCUnkRscId)
RscList may be empty in case of known resources at the peer side

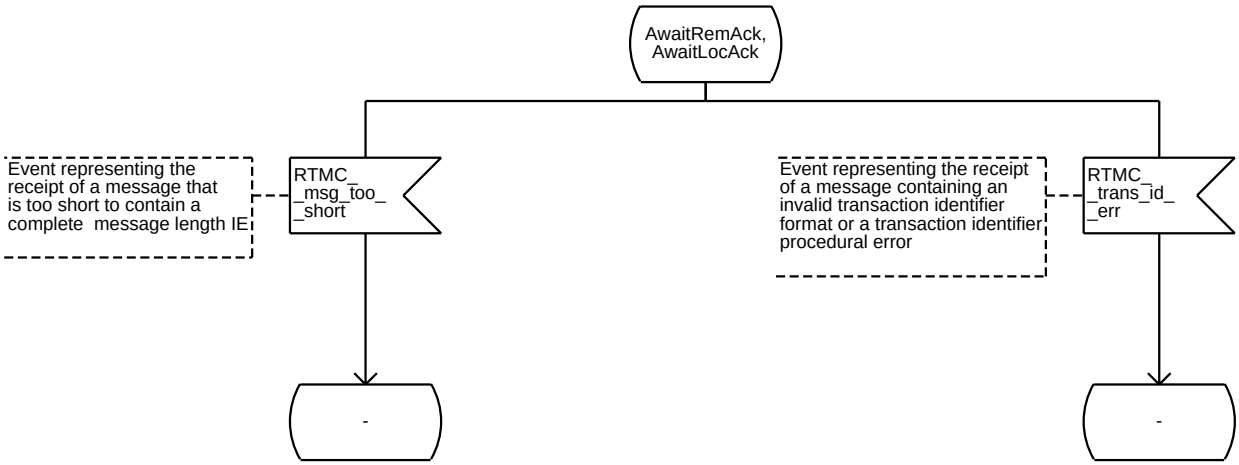

Process SN_CPROT
10(16)
/* Timer expirations */
/*SN_CPROT is responsible for sending/receiving RTMC messages*/
AwaitRemAck
T_consreq
T_consend
V_ExpCount < C_ExpMax
false
true
MIE_cons_ _check_conf
(C_TransErr, S_RTMCRscId)
RTMC_ _cons_ _check_req
(, S_RTMCRscId)
V_ExpCount:= V_ExpCount+1
set (T_consreq)
-
V_ExpCount < C_ExpMax
false
true
RTMC_ _cons_ _check_end
(, S_RTMCRscId)
V_ExpCount:= V_ExpCount+1
set (T_consend)
-
(C_TransErr, S_RTMCRscId)
MIE_cons_ _check_end_ _conf


;
/*SN_CPROT is responsible for sending/
receiving RTMC messages*/
/* Timer expirations */
AwaitRemAck
T_reset
T_lspid
V_ExpCount
<
C_ExpMax
false
true
RTMC_
_reset_rsc
(,
A_RTMCRscList)
TO PARENT
V_ExpCount:=
V_ExpCount+1
(V_IntTransId,
C_TransErr)
MIE_reset_
_conf
set
(T_reset)
-
V_ExpCount
<
C_ExpMax
false
true
RTMC_req_
_LSPID
TO PARENT
V_ExpCount:=
V_ExpCount+1
(C_TransErr)
MIE_LSP_
_ID_conf
set
(T_lspid)
-


;
/*SN_CPROT is responsible for sending/
receiving RTMC messages*/
/* Timer expirations */
AwaitRemAck
T_acl
V_ExpCount
<
C_ExpMax
false
true
RTMC_await_
_clear_comp
(,
A_RTMCRscList)
V_ExpCount:=
V_ExpCount+1
set
(T_acl)
MIE_error_ind
-

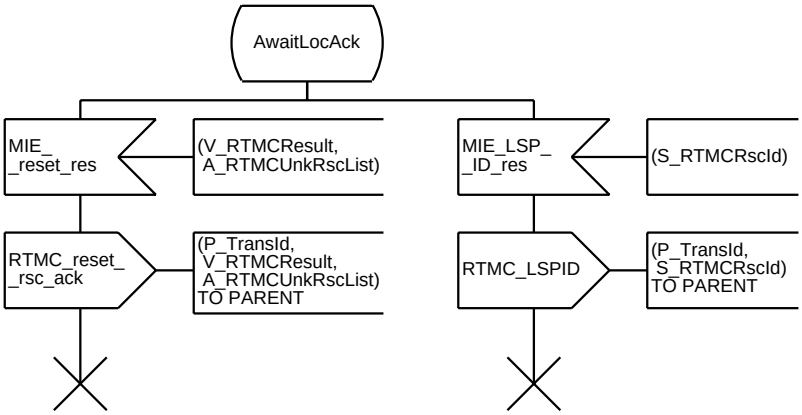

/*SN_CPROT is responsible for sending/receiving RTMC messages*/
/* Handling of unexpected events */
AwaitRemAck
RTMC_unblock_rsc(P_TransId),
RTMC_await_clear(P_TransId),
RTMC_req_LSPID(P_TransId),
RTMC_reset_rsc(P_TransId),
RTMC_block_rsc(P_TransId)
msg_error_
_handling
(P_TransId,
C_MsgNotCompatWithProtState,
PARENT)
-

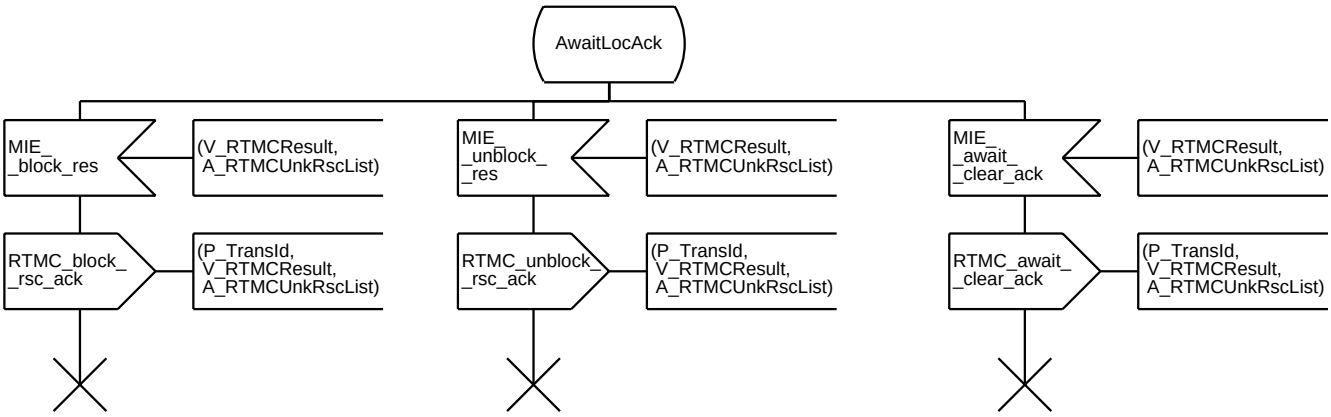

/*SN_CPROT is responsible for sending/receiving RTMC messages*/
/* Handling of pseudo events for error handling*/
AwaitRemAck, AwaitLocAck
Event representing the receipt of an unrecognized message.
RTMC_ _unknown_ _message
(P_TransId)
msg_error_ _handling
(P_TransId, C_UnkMsgType, PARENT)
-
Event representing the receipt of a message containing an unrecognized protocol discriminator.
RTMC_ _unknown_ _protdisc
-

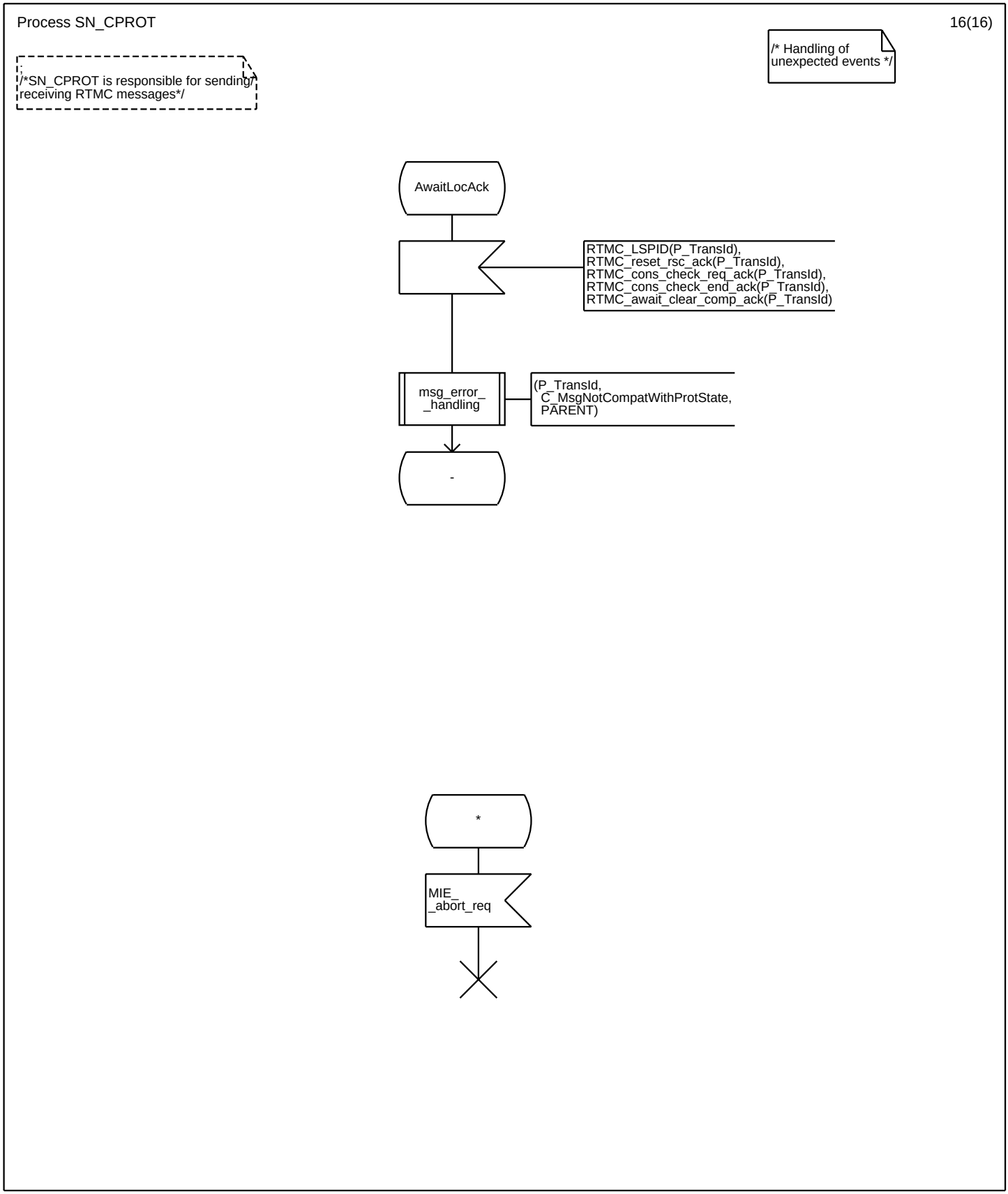

AwaitRemAck, AwaitLocAck
Event representing the receipt of a message that is too short to contain a complete message length IE
RTMC_ _msg_too_ _short
-
Event representing the receipt of a message containing an invalid transaction identifier format or a transaction identifier procedural error
RTMC_ _trans_id_ _err
-

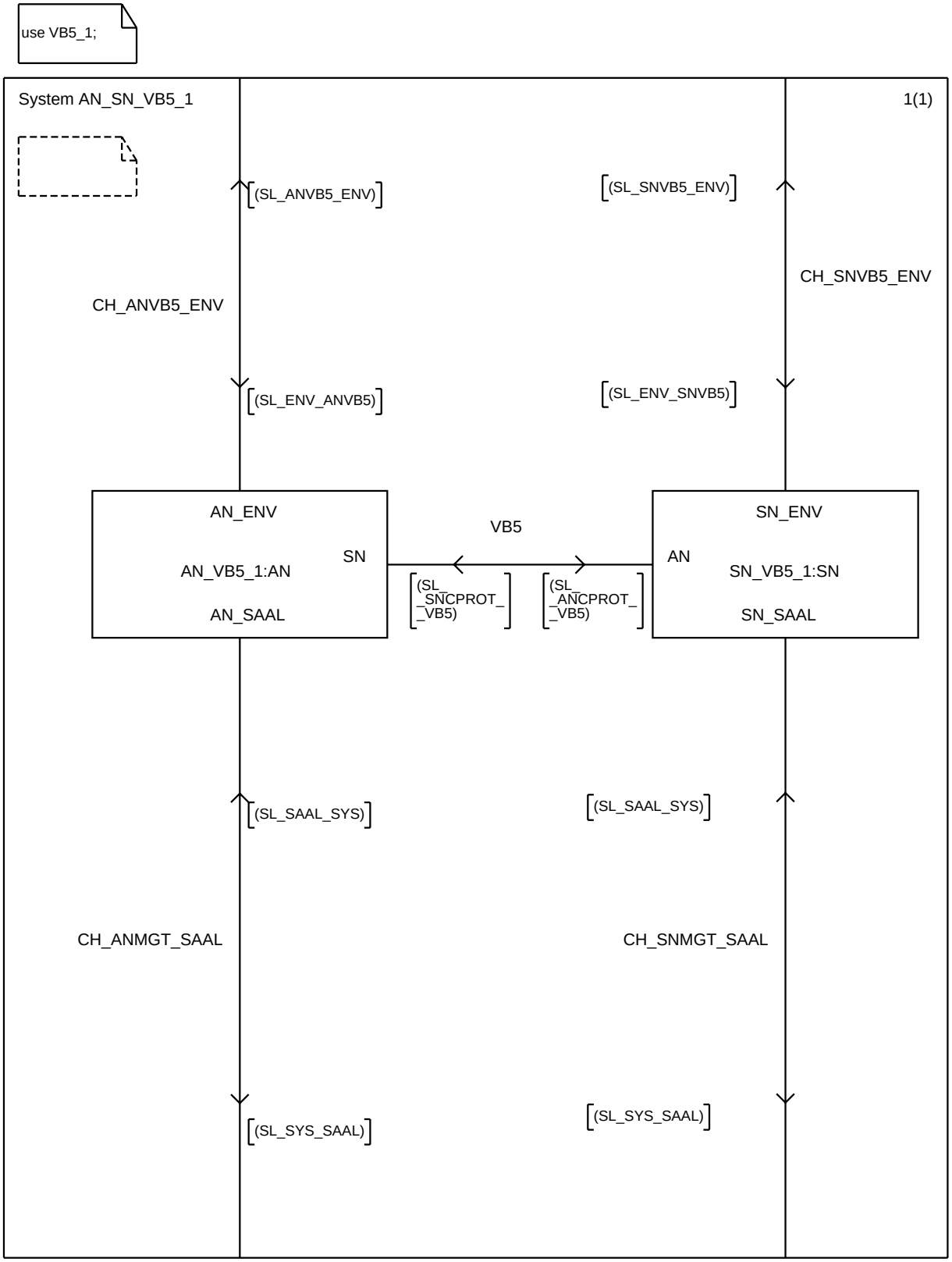
/*SN_CPROT is responsible for sending/receiving RTMC messages*/

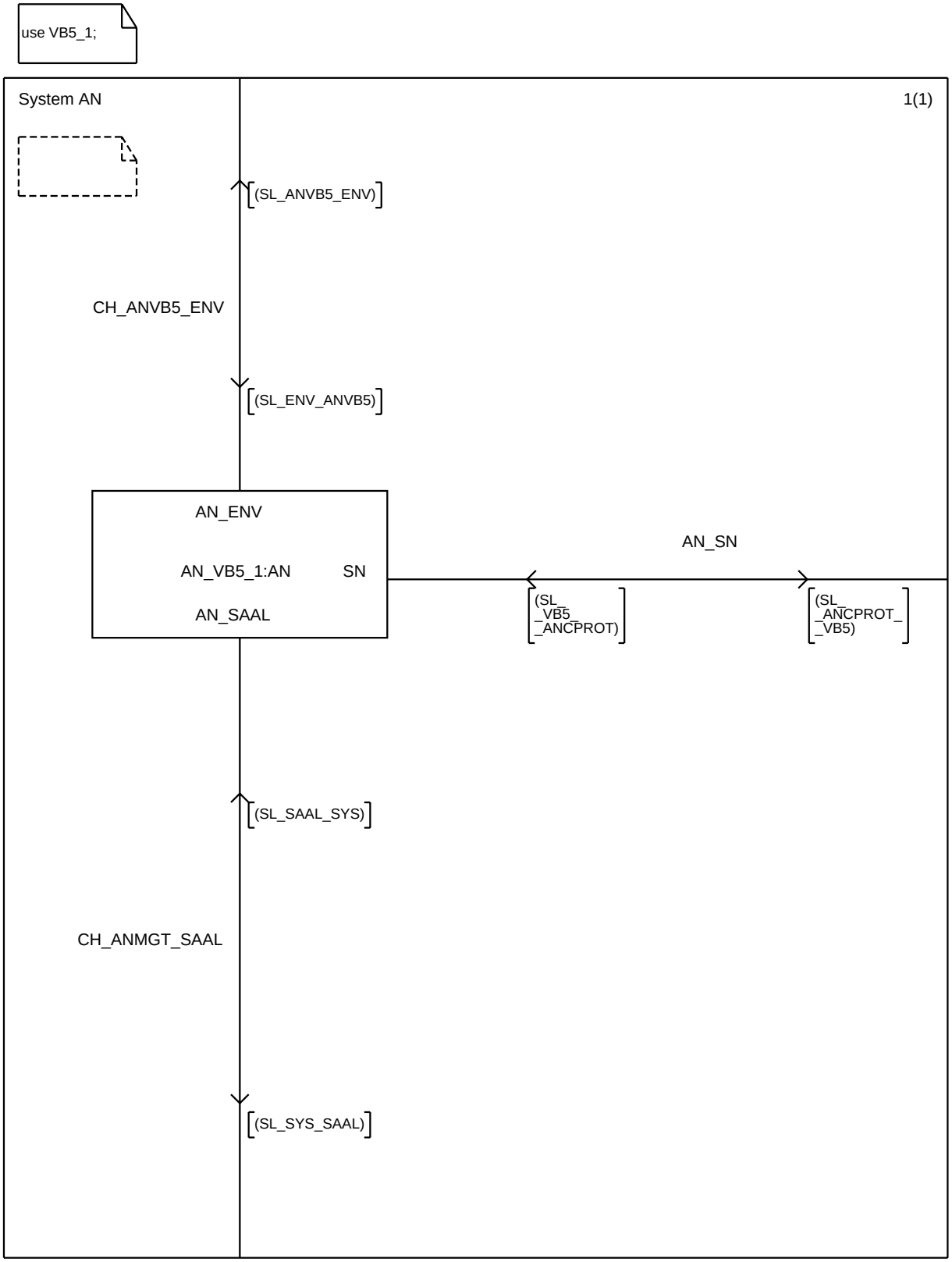
/*Handling of system management acknowledgements */

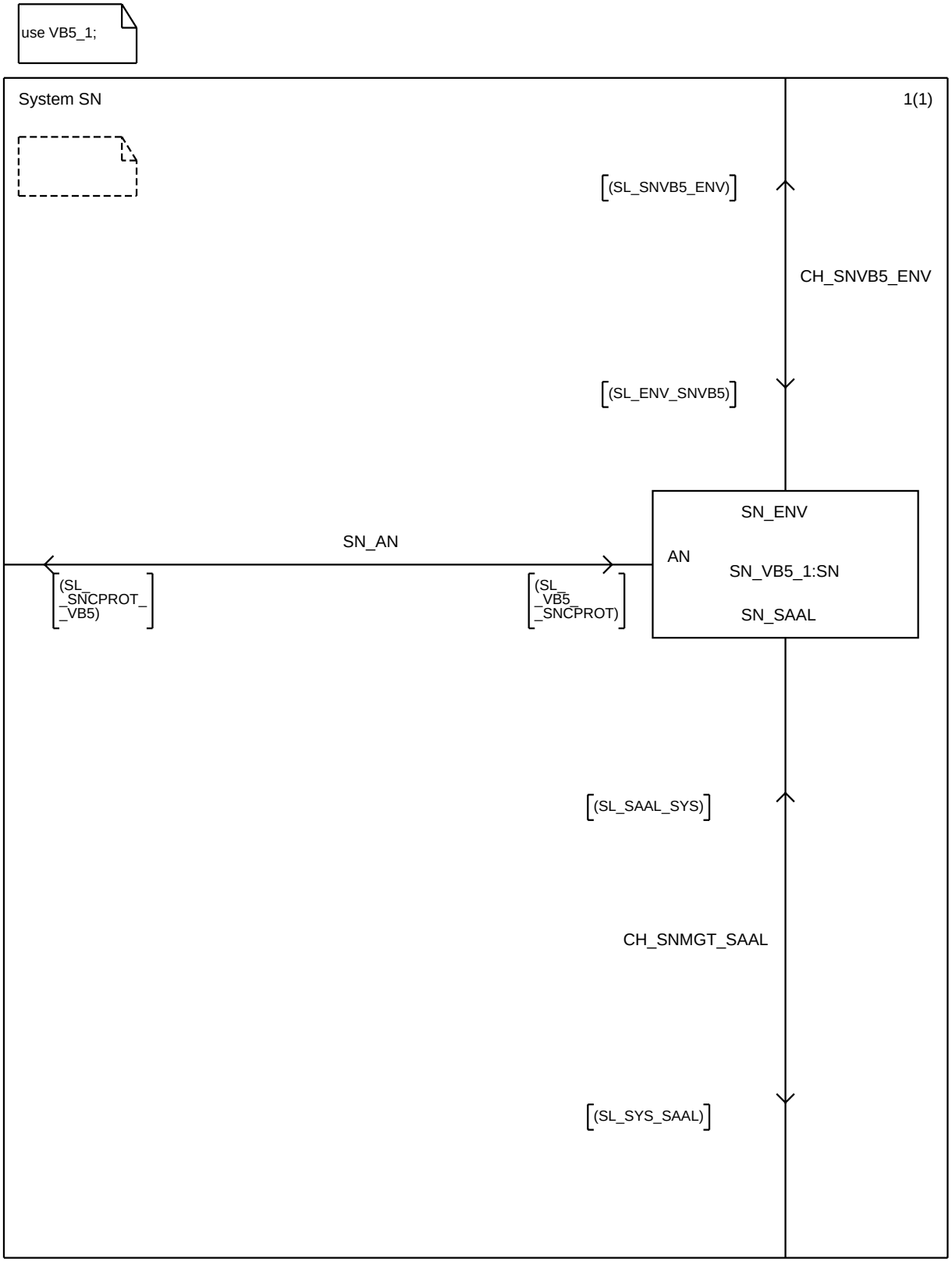





Process SN_CPROT
16(16)
/*SN_CPROT is responsible for sending/receiving RTMC messages*/
/* Handling of unexpected events */
AwaitLocAck
RTMC_LSPID(P_TransId),
RTMC_reset_rsc_ack(P_TransId),
RTMC_cons_check_req_ack(P_TransId),
RTMC_cons_check_end_ack(P_TransId),
RTMC_await_clear_comp_ack(P_TransId)
msg_error_
_handling
(P_TransId,
C_MsgNotCompatWithProtState,
PARENT)
-
*
MIE_
_abort_req


use VB5_1;
System AN_SN_VB5_1
1(1)
(SL_ANVB5_ENV)
(SL_SNVB5_ENV)
CH_SNVB5_ENV
CH_ANVB5_ENV
(SL_ENV_ANVB5)
(SL_ENV_SNVB5)
AN_ENV
SN
AN_VB5_1:AN
AN_SAAL
VB5
(SL_ _SNCPROT_ _VB5)
(SL_ _ANCPROT_ _VB5)
SN_ENV
AN
SN_VB5_1:SN
SN_SAAL
(SL_SAAL_SYS)
(SL_SAAL_SYS)
CH_ANMGT_SAAL
CH_SNMGT_SAAL
(SL_SYS_SAAL)
(SL_SYS_SAAL)


use VB5_1;
System AN
1(1)
(SL_ANVB5_ENV)
CH_ANVB5_ENV
(SL_ENV_ANVB5)
AN_ENV
AN_VB5_1:AN
SN
AN_SAAL
AN_SN
(SL_
_VB5_
_ANCPROT)
(SL_
ANCPROT
_VB5)
(SL_SAAL_SYS)
CH_ANMGT_SAAL
(SL_SYS_SAAL)


use VB5_1;
System SN
1(1)
(SL_SNVB5_ENV)
CH_SNVB5_ENV
(SL_ENV_SNVB5)
SN_ENV
SN_AN
AN
SN_VB5_1:SN
(SL_
SNCPROT
_VB5)
(SL_
_VB5_
_SNCPROT)
SN_SAAL
(SL_SAAL_SYS)
CH_SNMGT_SAAL
(SL_SYS_SAAL)