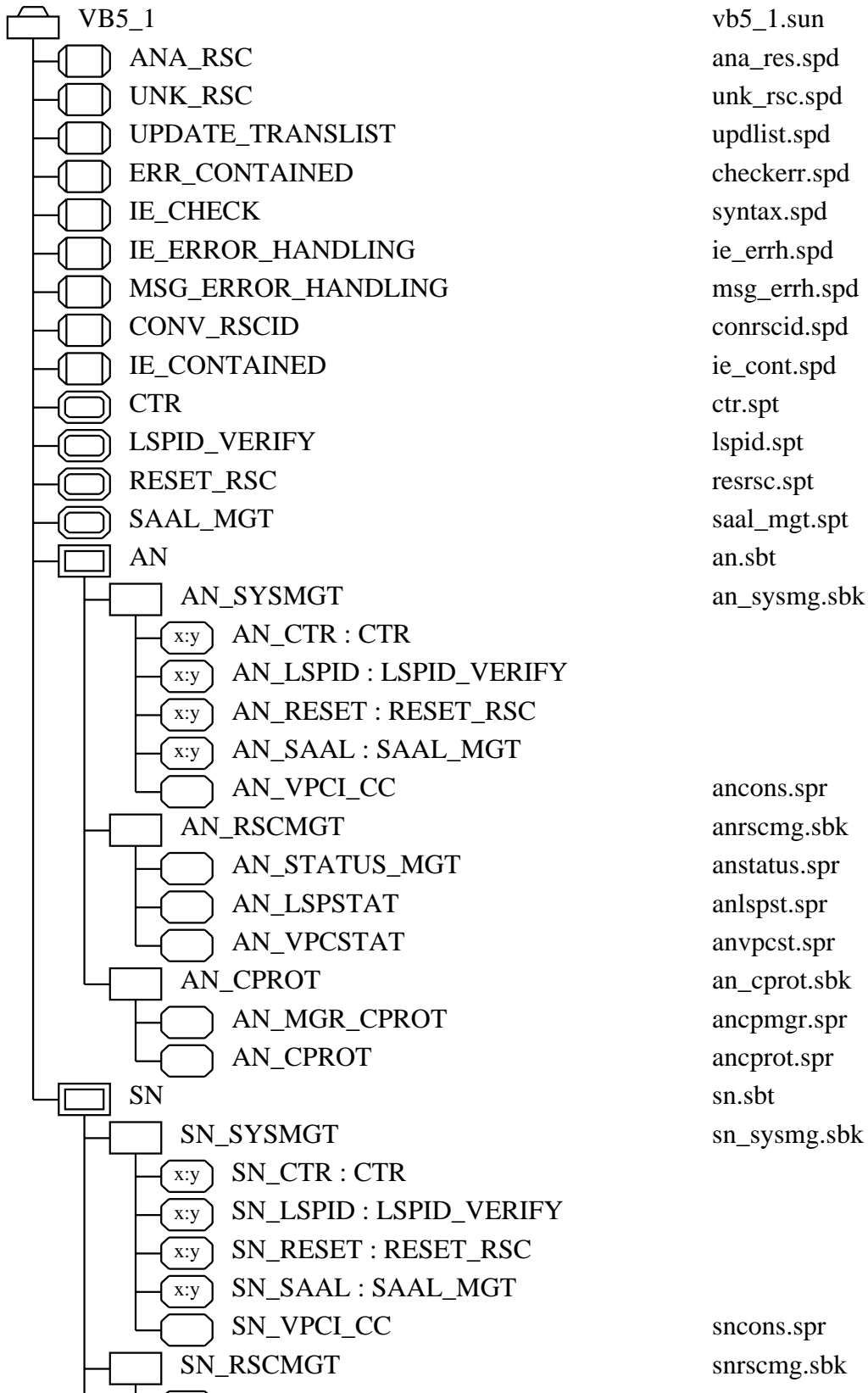


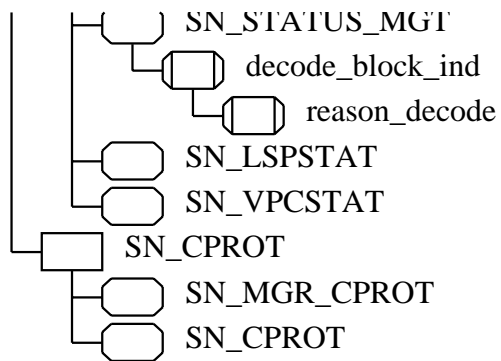
Table of Contents

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_LSPSTAT	118
Process AN_VPCSTAT	120
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_LSPSTAT	180
Process SN_VPCSTAT	181
Block SN_CPROT	183
Process SN_MGR_CPROT	184
Process SN_CPROT	194

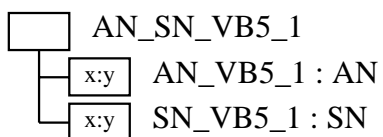
Diagram Structure





snstatus.spr
 dec_bloc.spd
 dec_reas.spd
 snlspst.spr
 snvpcst.spr
 sn_cprot.sbk
 sncpmgr.spr
 sncprot.spr

— Systems



ansn.ssy



an.ssy



sn.ssy

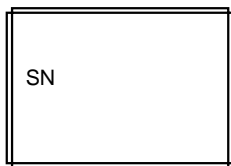
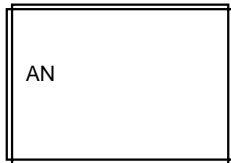


Package VB5_1

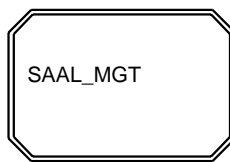
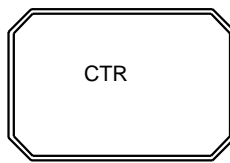
1(37)



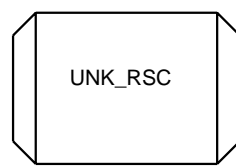
/* The block types
AN & SN. */

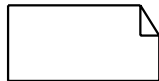


/* Process types that are
used both in AN and SN. */



/* Procedures that are used
both in AN and SN. */



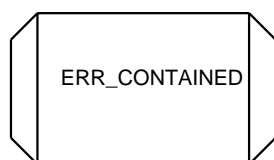


Package VB5_1

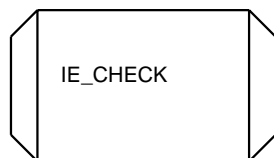
2(37)

/*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 errors 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 explicitly 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)
*/



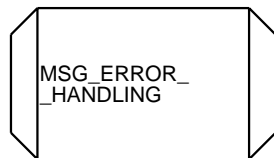
Procedure ERR_CONTAINED is used to control the error handling during simulation runs. It allows switching on and off of the error handling via an external synonym C_ErrContained. If C_ErrContained is set to FALSE, no error handling is activated. If C_ErrContained is set to TRUE, then you have to indicate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



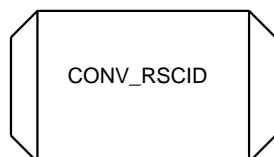
Procedure IE_CHECK is used to check the syntax of the IEs of a VB5-message. The common octets of each message (first 9 octets) are not considered for error handling. The syntax check is described in an informal manner and shall be used for simulation purposes only. Subclause 13.3.1.5 'General error handling' contains the normative part of the error handling procedures which shall be applied for real implementations.



Procedure IE_ERROR_HANDLING is used for handling of an IE syntax error (see table 36)



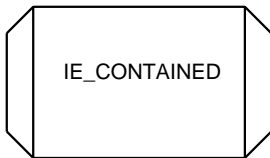
Procedure MSG_ERROR_HANDLING is used for handling of syntax errors in the common octets of each message (first 9 octets, see table 35)



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*/



Procedure IE_CONTAINED shall be used for analysis of Rsclds within a VB5 message (procedure ANA_RSC).

For simulation purposes this procedure evaluates the number of present Rsclds. The Rsclds are stored within an unlimited array A_RTMCRRscList. 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_EmptyRscld, i.e. C_EmptyRscld has to be set to (. 0,0,0,0 .)

To check whether a resource ID is present or not, this synonym C_EmptyRscld is used.

/*Resource identifier IE not present*/
SYNONYM C_EmptyRscld ST_Rscld = (. 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 Rscld of each VPC is given by a parameter of the primitive to the corresponding STATUS_MGT process, see signal definition.
The Rscld is only known by the STATUS_MGT process. This process maps the Rscld to the corresponding PID. I.e. the VPCSTAT process needs not to know his own Rscld.
For the indication of Rsclds within primitives, see definition of type ST_Rscld. 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 Rscld 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
*/



Package VB5_1

5(37)



/* Naming conventions */

```
/*Type definitions for signal parameters*/
/**/
/*In the following the SDL data types of signal parameters are defined. This is done in order
to allow for tool supported simulation and validation of the SDLs. This SDL data modelling
shall not supersede the message and parameter formats and codes as defined in clause 14. Only
those data are modelled which are needed for simulation and validation*/
/**/
/*Prefix Naming Conventions for Data Type Definitions
AT_XXX  ARRAY Data Type
IT_XXX  INTEGER Data Type
PT_XXX  PID Data Type
ST_XXX  STRUCT Data Type*/
/**/
/*Prefix Naming Conventions for Variables and Constants
A_XXX  ARRAY Variable
B_XXX  BOOLEAN Variable
C_XXX  Constant of any type
P_XXX  PID Variable
S_XXX  STRUCT Variable
V_XXX  INTEGER or NATURAL Variable
*/
```

```
/*Prefixes for primitives*/
/**/
/*VB5.1 primitives represent in an abstract way the exchange of information and control
between functional entities. They do not specify or constrain implementation. The primitives
are classified as follows*/
/*
AAL_    primitives between the system management and the ATM adaptation layer
MDU_    primitives between system management or resource management and RTMC protocol
mee_    primitives between system management, resource management or RTMC protocol
        and the environment in which the RTMC protocol functions are performed
MIE_    primitives internal to a SDL block
MPH_    primitives between system management and resource management
*/
```

/* 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_RTMCRsclnd = INTEGER
CONSTANTS 0:2
ENDSYNTYPE IT_RTMCRsclnd;
/*Possible Resource Indicator Values*/
SYNONYM C_CompLSP IT_RTMCRsclnd = 0; /*Complete LSP*/
SYNONYM C_ConnLSP IT_RTMCRsclnd = 1; /*Connections at LSP*/
SYNONYM C_ConnLUP IT_RTMCRsclnd = 2; /*Connections at LUP*/
/**/
/*Typedefinition for Unknown Resource Indicator Value*/
/*refer to table 58*/
SYNTYPE IT_RTMCUnkRsclnd = INTEGER
CONSTANTS 0:1
ENDSYNTYPE IT_RTMCUnkRsclnd;
/*Possible Unknown Resource Indicator Values*/
SYNONYM C_LogIdUnk IT_RTMCUnkRsclnd = 0; /*LSP Id or LUP Id unknown*/
SYNONYM C_VPCIUnk IT_RTMCUnkRsclnd = 1; /*VPCI unknown*/
/**/
/*Typedefinition for Resource Identifier Information Element*/
/*refer to figure 74*/
NEWTYPE ST_RsclId STRUCT
V_Rsclnd IT_RTMCRsclnd; /*Resource Indicator*/
V_UnkRsclnd IT_RTMCUnkRsclnd; /*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_RTMCBIRscld STRUCT
S_RTMCRscld ST_Rscld; /*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_Rscld)
ENDNEWTYPE;
/**/
/*Typedefinition for repeated Blocked Resource Identifier Information Elements in a RTMC message*/
NEWTYPE AT_RTMCBIRscldList ARRAY
(INTEGER,ST_RTMCBIRscld)
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_IEMContErr 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_RTMCTResult = INTEGER
CONSTANTS 0:4
ENDSYNTYPE IT_RTMCTResult;
/**/
/*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; /*Admin Full*/
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_BIRscId STRUCT
  S_RscId      ST_RscId;
  V_Reason     IT_Reason;
ENDNEWTYPE ST_BIRscId;
/**/
NEWTYPE AT_BIRscList ARRAY
  (INTEGER, ST_BIRscId)
ENDNEWTYPE AT_BIRscList;
```

/*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_RTMCBIRscList),
/*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*/

```

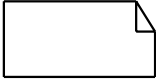
signal
meeVerifyLspIdReq,
/*Environment requests the verification of the LSP ID
*/
meeVerifyLspIdConf (IT_Result),
/* Environment is informed about the result of the verify LSP ID procedure
*/
meeResetRscReq (AT_RscList),
/*Environment requests the reset of resources (VPC/LSP), SN shall be informed
*/
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_BIRscList),
/*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_BIRscList),
/*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_Rscld),
/* Environment requests the invocation of the VPCI consistency check procedure, AN shall be informed
*/
meeConsCheckConf (IT_Result, ST_Rscld),
/*Environment is informed about the start of the VPCI consistency check procedure, ST_Rscld is optional
*/
meeConsCheckEndReq (ST_Rscld),
/*Environment requests the termination of the VPCI consistency check procedure, AN shall be informed
*/
meeConsCheckEndConf (IT_Result, ST_Rscld);
/*Environment is informed about the termination of the VPCI consistency check procedure*/
```

/*Declaration of common primitives
from/to CPROT entity*/

```

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*/

```



Package VB5_1

17(37)



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

```
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_RTMCBIRscList),
/*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_RTMResult, 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_RTMResult, ST_RscId);
/*CPROT entity is requested to send a CONSISTECNY_CHECK_END_ACK message to the SN
*/
```

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

```

signal
MDU_unblock_ind (PT_CprotPID, AT_RscList),
/*CPROT entity indicates the receipt of a UNBLOCK_RSC message
*/
MDU_unblock_res (PT_CprotPID, IT_RTMCRresult, AT_RscList),
/*CPROT entity is requested to send a UNBLOCK_RSC_ACK message
*/
MDU_block_ind (PT_CprotPID, AT_RTMCBIRscList),
/*CPROT entity indicates the receipt of a BLOCK_RSC message
*/
MDU_block_res (PT_CprotPID, IT_RTMCRresult, 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_RTMCRresult, AT_RscList),
/*CPROT entity is requested to send an AWAIT_CLEAR_ACK message for VPCs to the AN
*/
MDU_cons_check_req (ST_Rscld),
/*CPROT entity is requested to send a CONS_CHECK_REQ message to the AN
*/
MDU_cons_check_conf (IT_Result, ST_Rscld),
/*CPROT entity indicates the reception of a CONS_CHECK_ACK message from the AN
*/
MDU_cons_check_end_req (ST_Rscld),
/*CPROT entity is requested to send a CONS_CHECK_END message to the AN
*/
MDU_cons_check_end_conf (IT_Result, ST_Rscld);
/*CPROT entity indicates the reception of a CONS_CHECK_END_ACK message from the AN
*/

```



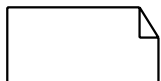
Package VB5_1

19(37)



/* 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;

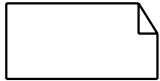
signallist SL_CTR_CPROT=
MDU_start_traffic;
MDU_stop_traffic;

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

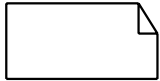
24(37)



/*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;

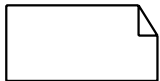
/*Declaration of SN Signallists
for primitives to the
environment*/

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_Rscld),
/*VPC STATFSM is requested to reset a blocking reason
*/
MIE_VPC_unblock_conf (ST_Rscld),
/*VPC STATFSM confirms the resetting of a blocking reason
*/
MIE_VPC_unblock_ind (ST_Rscld),
/*Indication to SN_VPCSTAT process that all blocking reason have been reset in the AN
*/
MIE_VPC_unblock_res (ST_Rscld),
/* SN_VPCSTAT process confirms the resetting of all blocking reasons
*/
MIE_LSP_unblock_req (ST_Rscld),
/*LSP STATFSM is requested to reset a blocking reason
*/
MIE_LSP_unblock_conf (ST_Rscld),
/*LSP STATFSM confirms the resetting of a blocking reason
*/
MIE_LSP_unblock_ind (ST_Rscld),
/*Indication to SN_LSPSTAT process that all blocking reason have been reset in the AN
*/
MIE_LSP_unblock_res (ST_Rscld),
/* 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_Rscld, IT_Reason),
/*VPC STATFSM is requested to set a blocking reason
*/
MIE_VPC_block_conf (ST_Rscld, ST_RTMCReason),
/*VPC STATFSM confirms the setting of a blocking reason
*/
MIE_VPC_await_clear_req (ST_Rscld),
/*VPC STATFSM is requested to enter the shutdown state
*/
MIE_VPC_await_clear_conf (ST_Rscld),
/*VPC STATFSM confirms the transition to the shutdown state
*/
MIE_LSP_block_req (ST_Rscld, IT_Reason),
/*LSP STATFSM is requested to set a blocking reason
*/
MIE_LSP_block_conf (ST_Rscld, ST_RTMCReason);
/*LSP STATFSM confirms the setting of a blocking reason*/

```



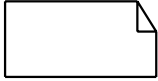
Package VB5_1

31(37)



/*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)



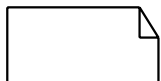
/*Declaration of AN Signallists
for status FSM primitives*/

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;

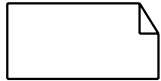
signallist SL_SNLSPSTAT_STATUS =
MIE_LSP_unblock_res,
MIE_LSP_unblock_conf,
MIE_LSP_block_res;

/* 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_RTMCBIRscList),
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)

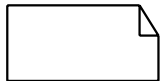


/* Declaration of primitives internal
to AN_CPROT */

```
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_RTMCBIRscList),
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_RTMCBIRscList),
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);
```



Package VB5_1

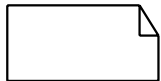
36(37)



/* 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;
```

```

: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 */

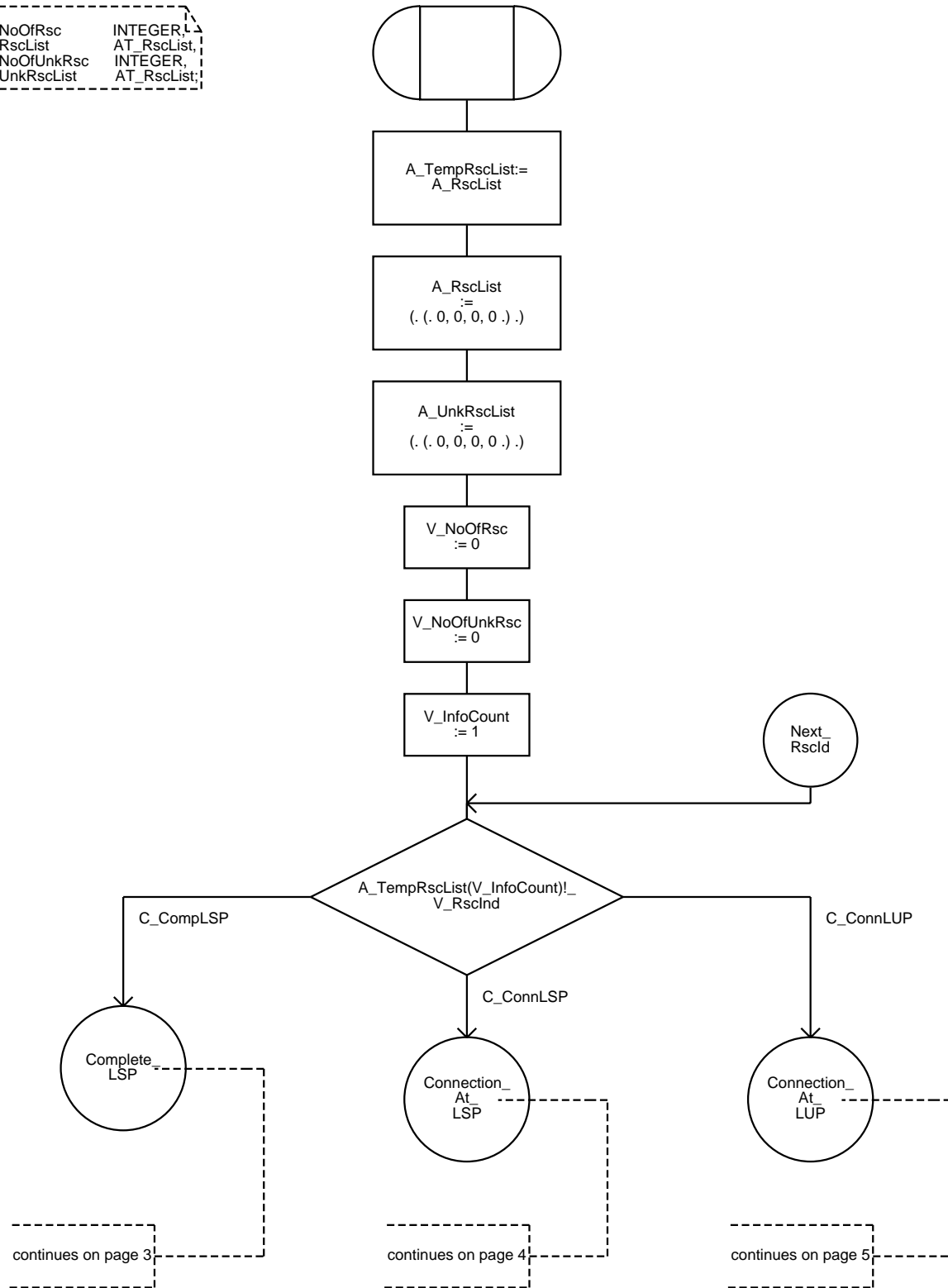
/*Definitions and Declarations*/

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

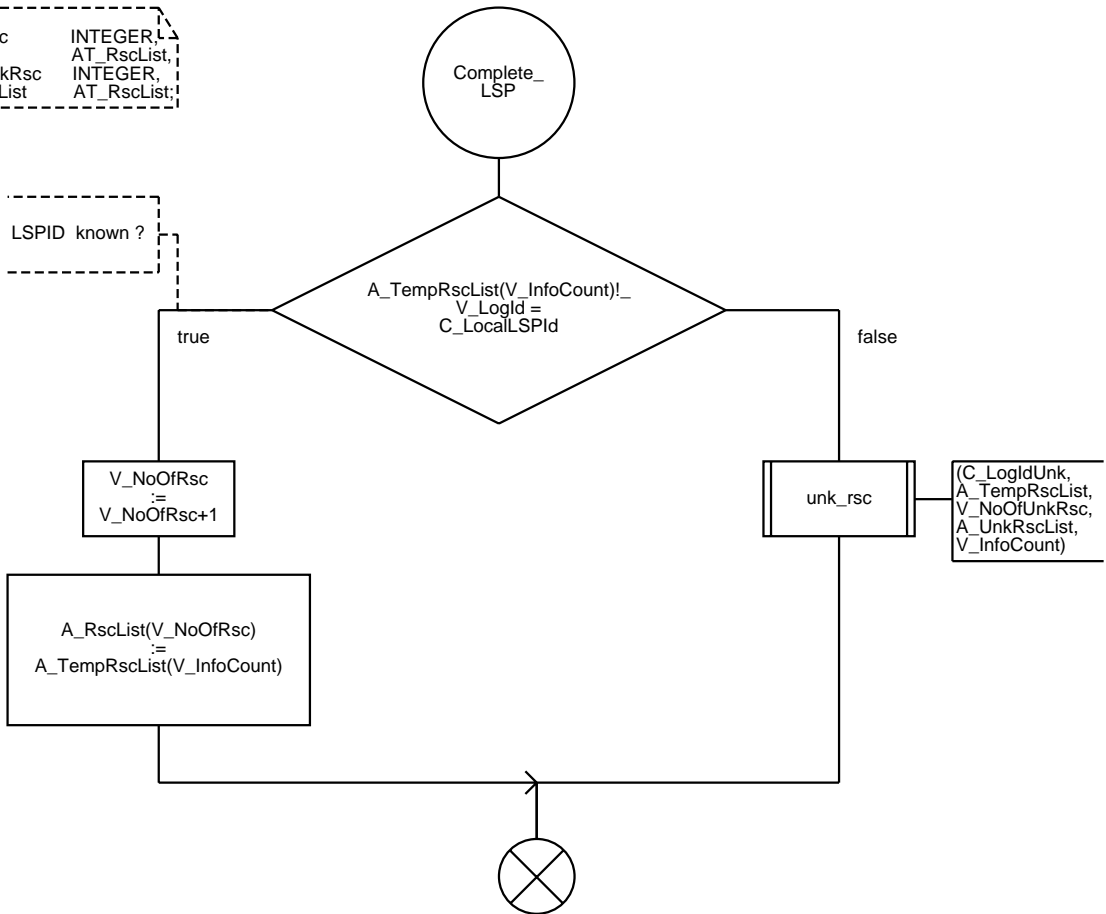
/*Data declarations for internal use*/
DCL
A_TempRscList AT_RscList ; /*Array for temporary storage of Resource Identifier Information Elements*/

```

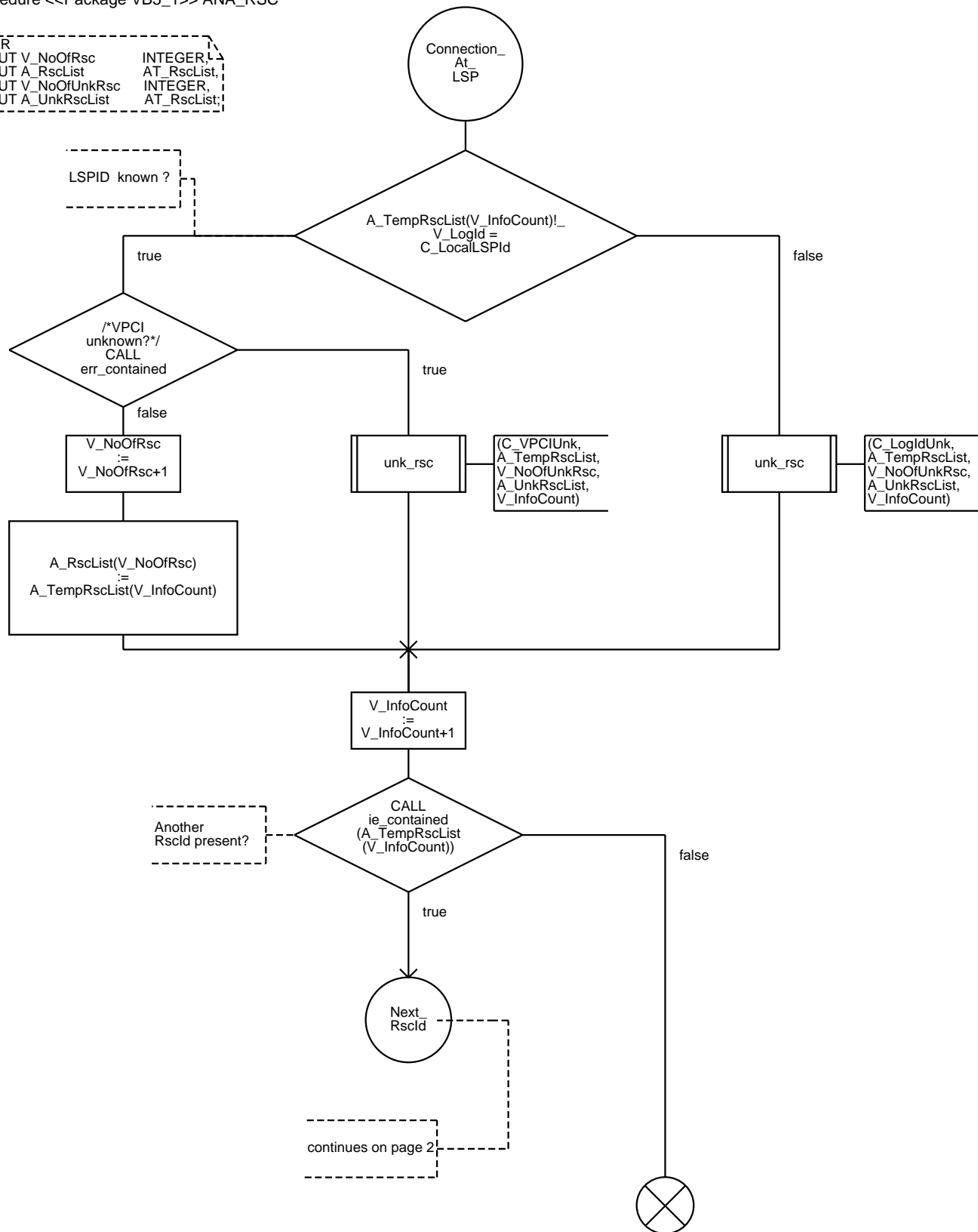
:FPAR
IN/OUT V_NoOfRsc      INTEGER
IN/OUT A_RscList      AT_RscList
IN/OUT V_NoOfUnkRsc   INTEGER
IN/OUT A_UnkRscList   AT_RscList
    
```



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

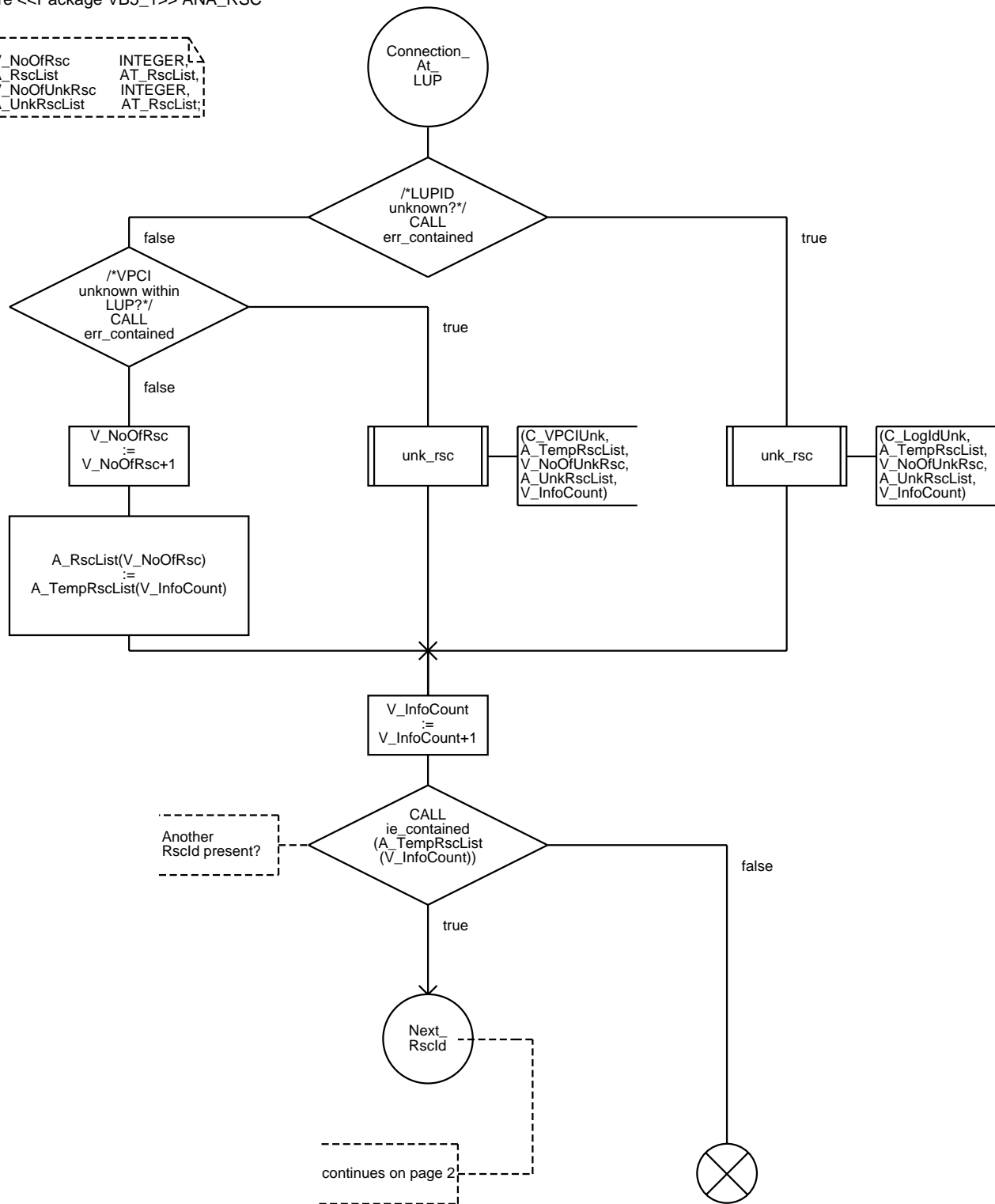


FPAR
 IN/OUT V_NoOfRsc INTEGER;
 IN/OUT A_RscList AT_RscList;
 IN/OUT V_NoOfUnkRsc INTEGER;
 IN/OUT A_UnkRscList AT_RscList;



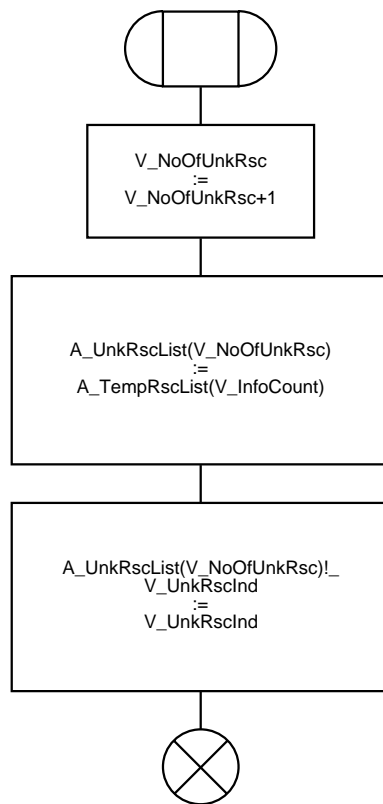
```

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

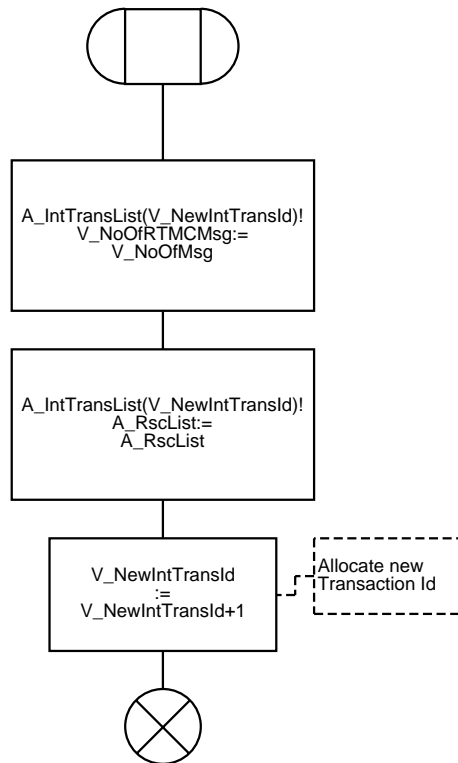


```

/*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;
  
```



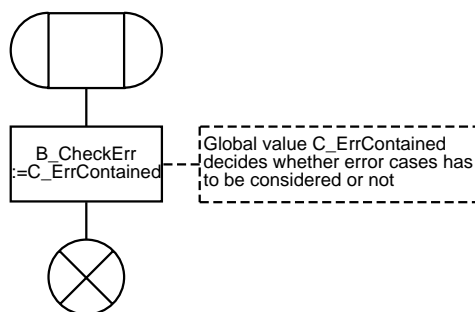
;FPAR
 IN/OUT A_IntTransList AT_IntTransList;
 V_NoOfMsg INTEGER;
 IN/OUT V_NewIntTransId IT_IntTransId,
 A_RscList AT_RscList;



Procedure ERR_CONTAINED

1(1)

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

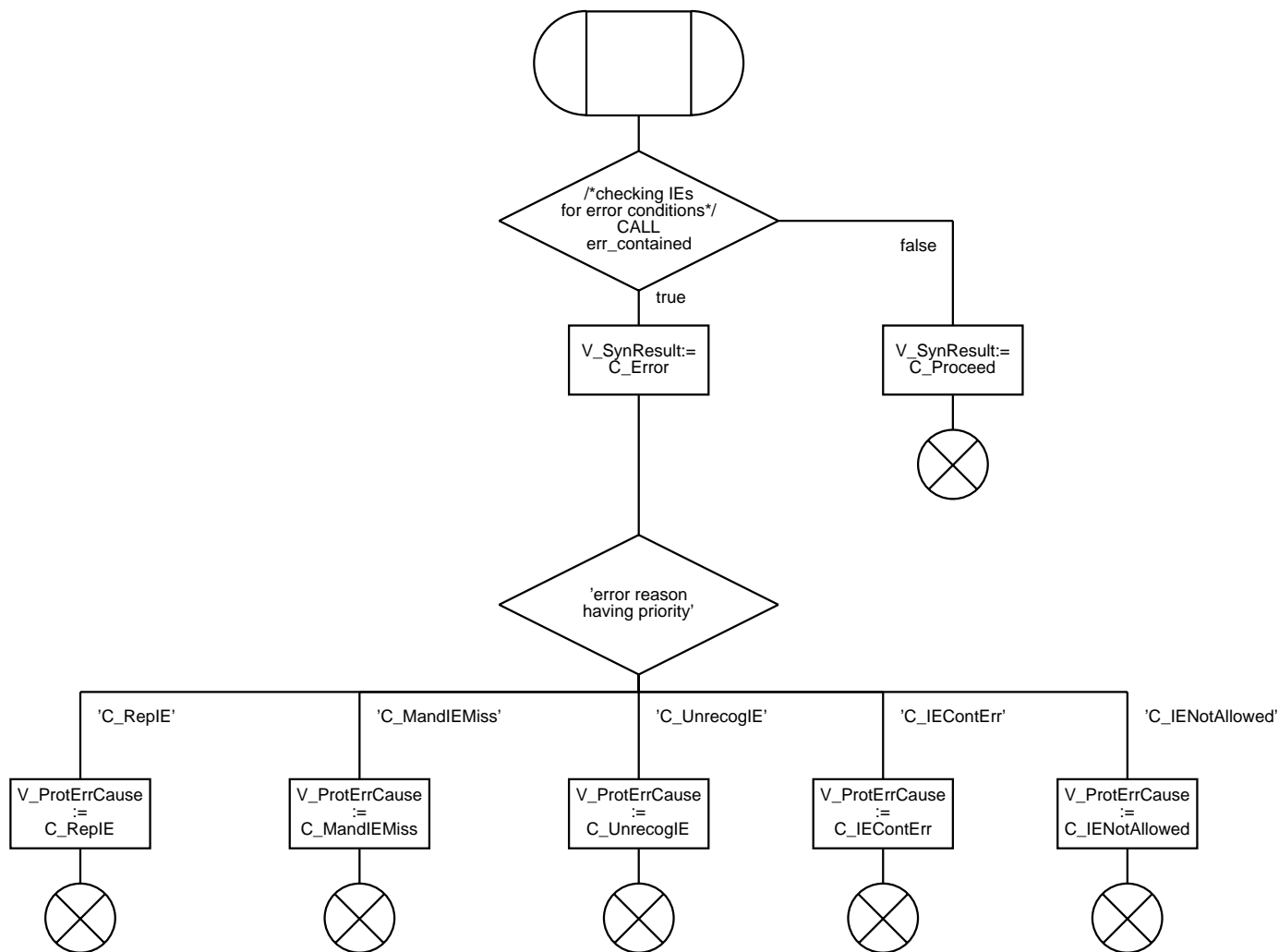


FPAR
IN/OUT V_ProtErrCause IT_RTMCProtErrCause;
RETURNS V_SynResult INTEGER;

/* Procedure used only for simulation purposes. Supports the syntax error checking of IEs. The procedure checks first whether error checking is activated at all via the external synonym C_ErrContained. If error checking is activated, then the possible protocol error cause (see table 55) can be set by direct input during simulation. If error checking is de-activated, the normal procedures apply. */

/*Definitions and
Declarations*/

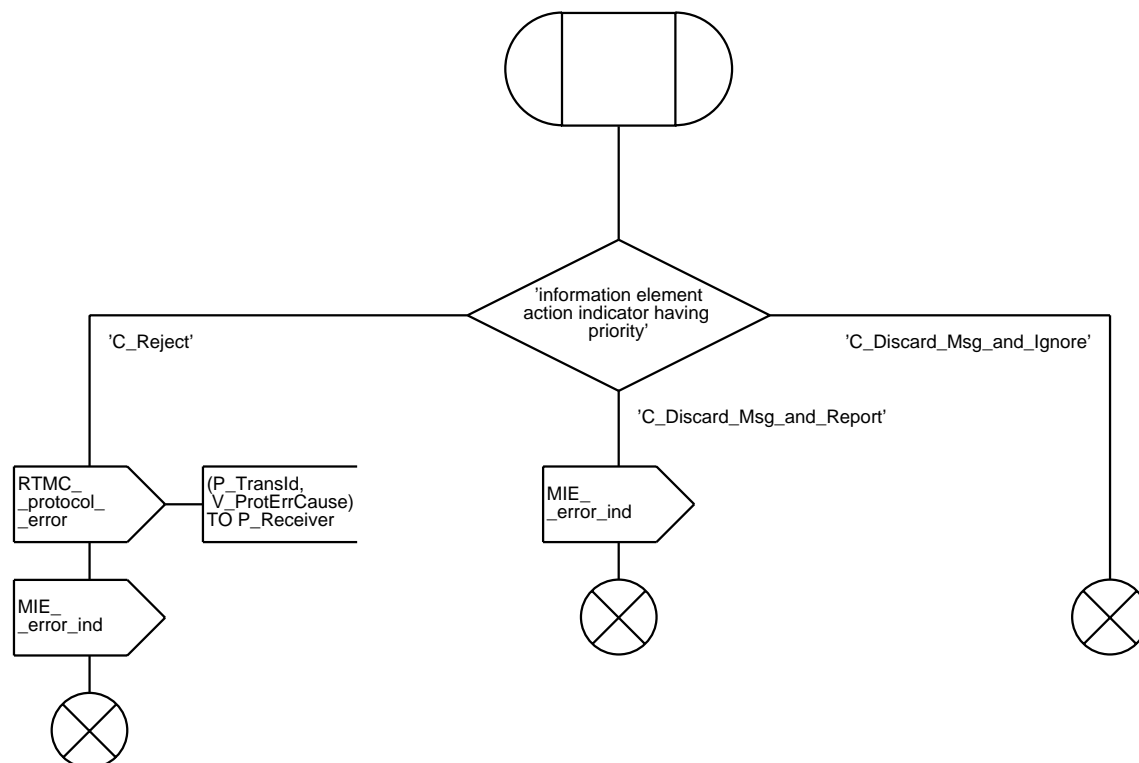
/*Possible values for V_SynResult */
SYNONYM C_Proceed INTEGER =1; /*possible result of syntax check: proceed with message processing*/
SYNONYM C_Error INTEGER =2; /*possible result of syntax check: error in message*/



FPAR
P_TransId PT_TransId,
V_ProtErrCause IT_RTMCProtErrCause,
P_Receiver PID;

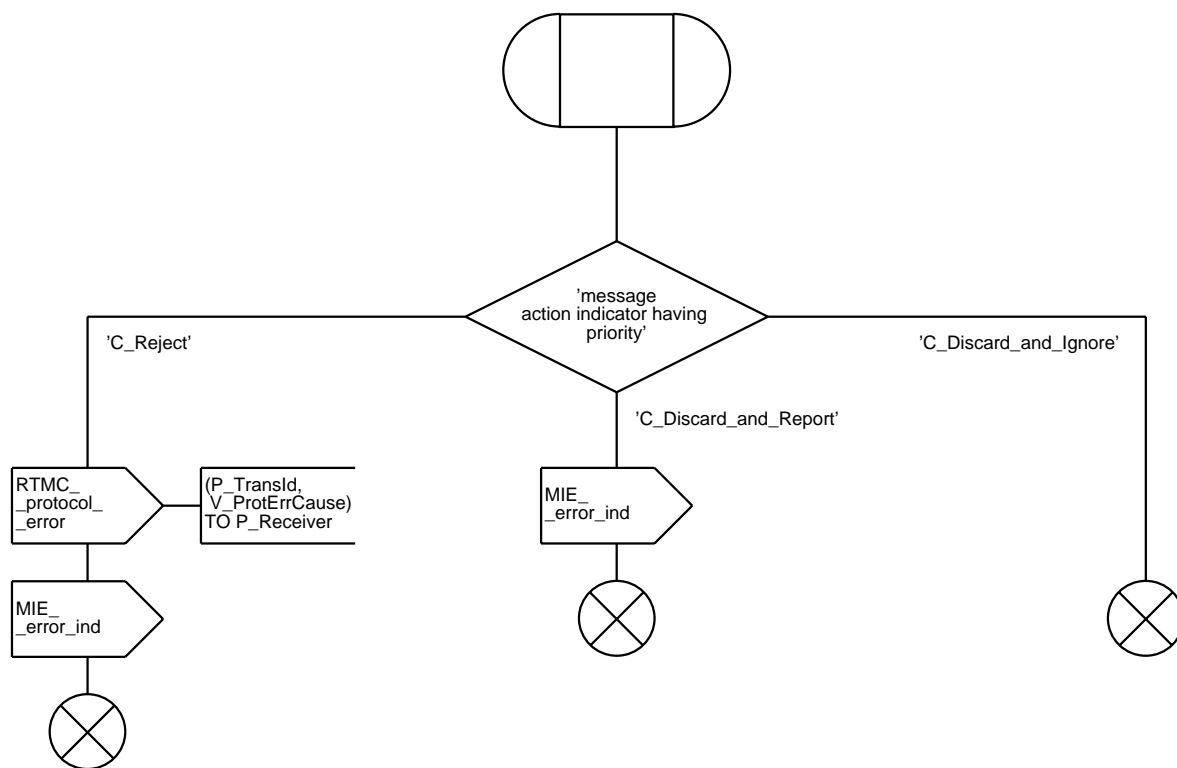
/* Procedure used only for simulation purposes. The possible actions for handling of an IE syntax error (see table 36) can be set by direct input during simulation. */

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



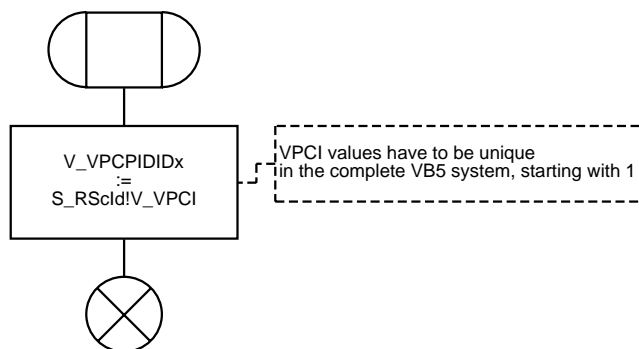
FPAR
 P_TransID PT_TransId,
 V_ProtErrCause IT_RTMCProtErrCause,
 P_Receiver PID;

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



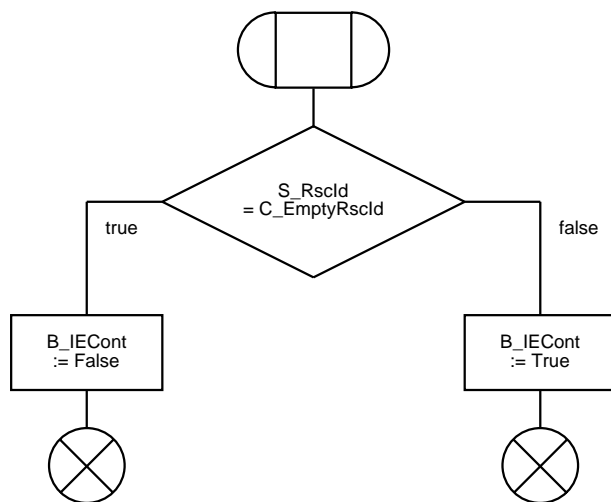
```

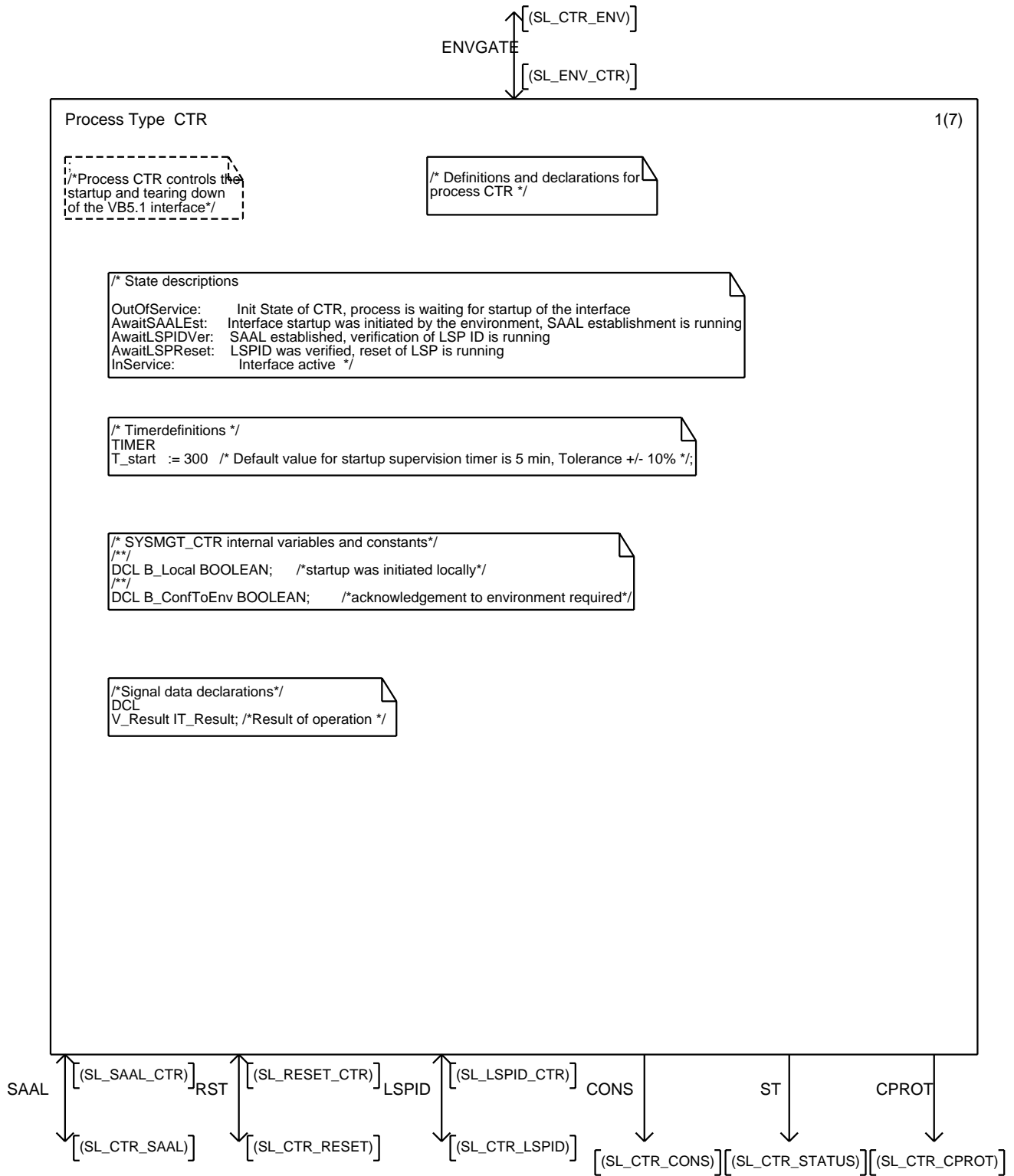
:
: FPAR S_RscId ST_RscId;
: RETURNS V_VPCPIDx INTEGER;
: /*Only for simulation use. Determines the index of a resource
: identifier in the PID table*/
    
```



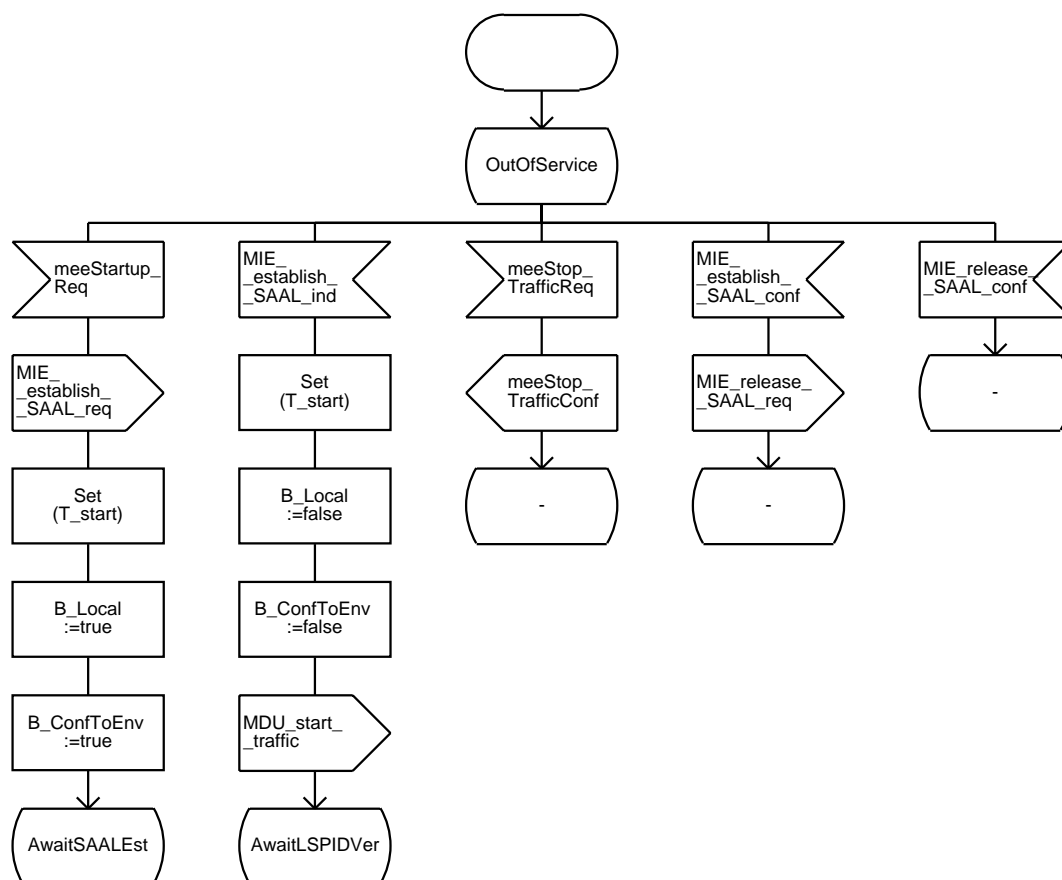
```

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

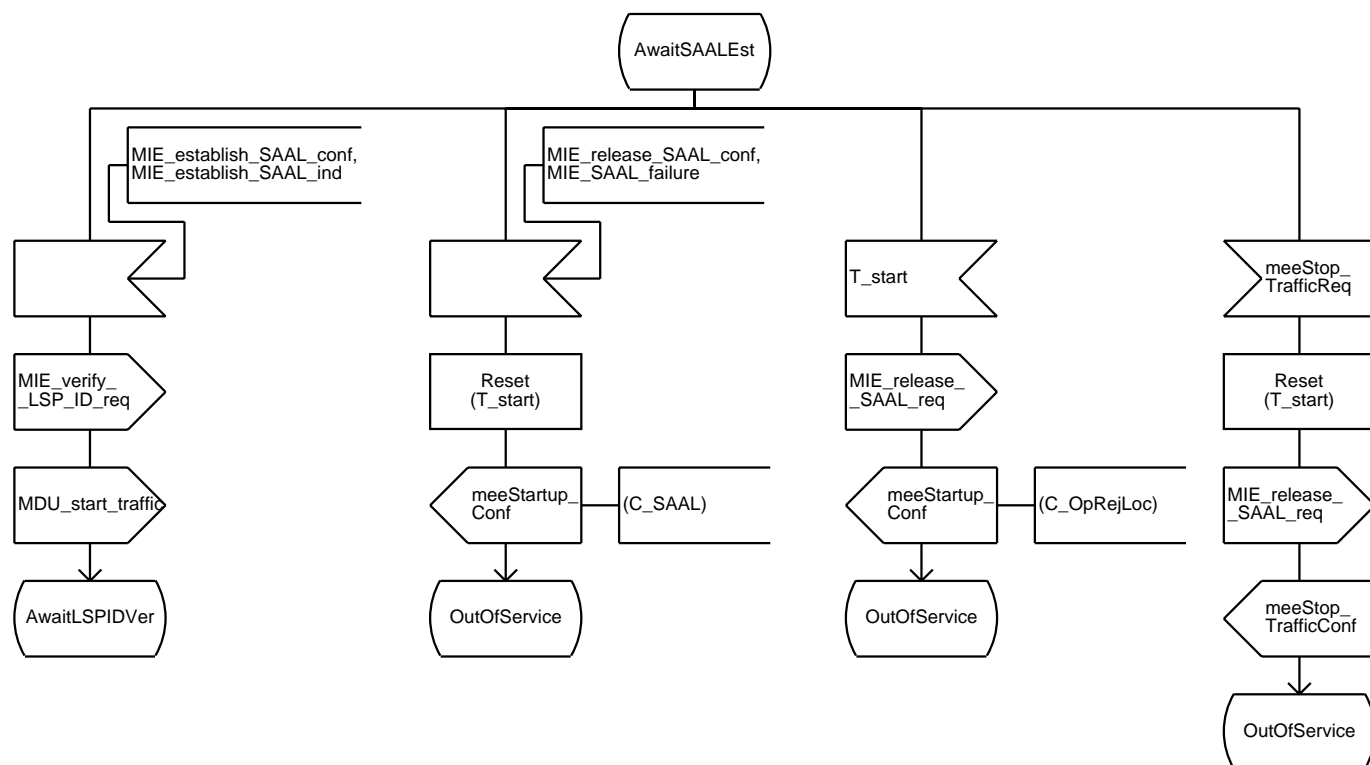




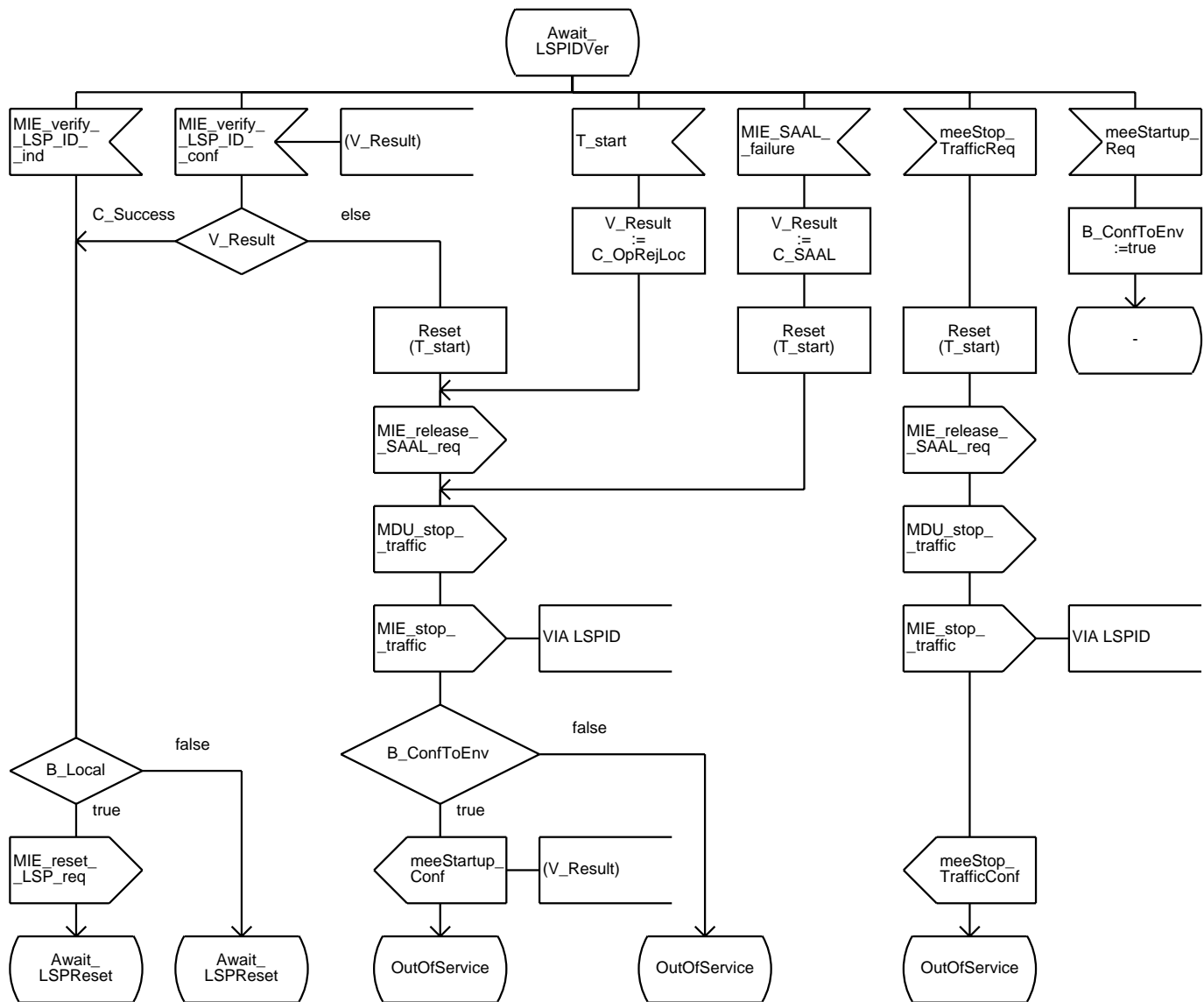
/*Process CTR controls the
startup and tearing down
of the VB5.1 interface*/



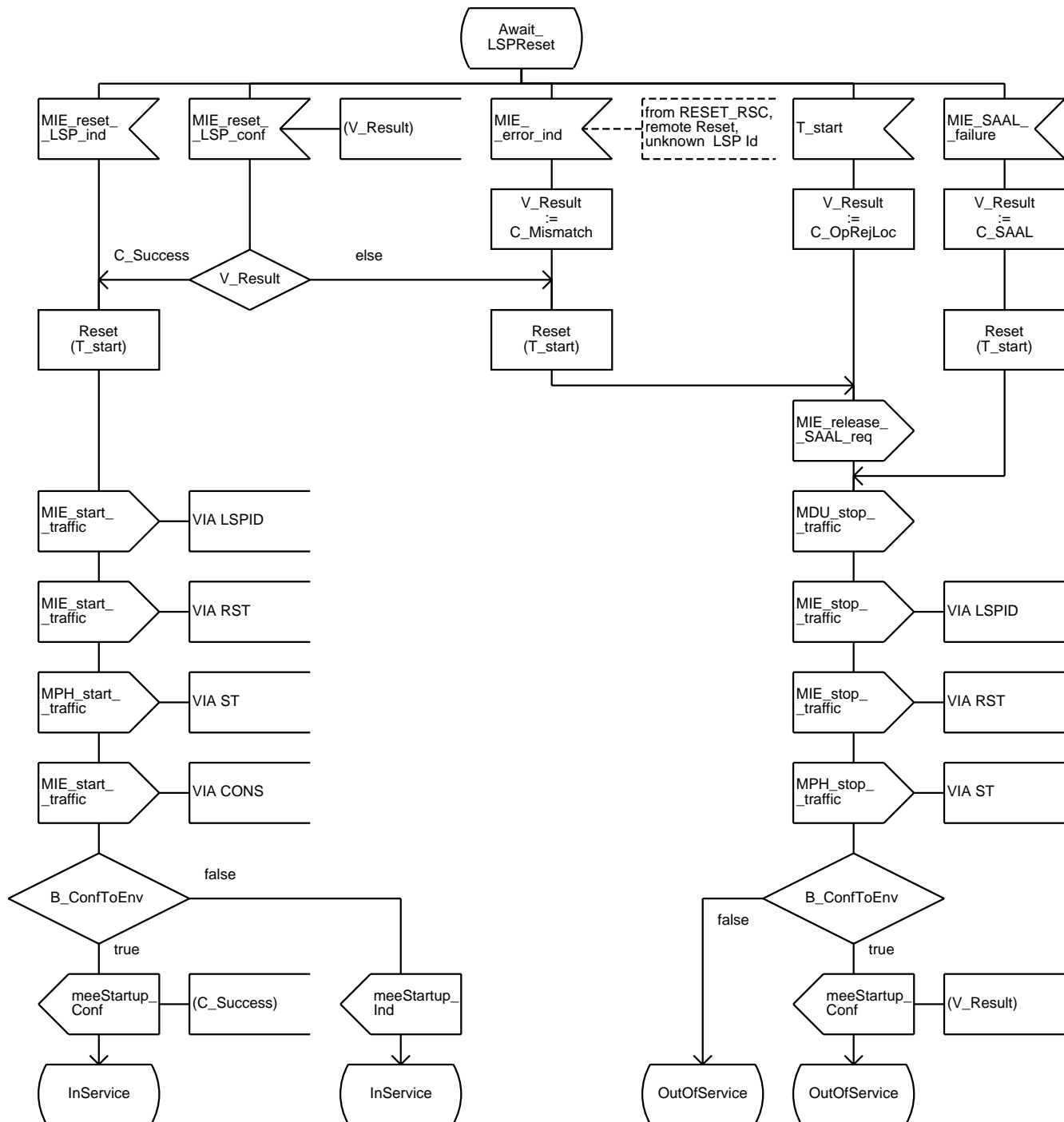
/*Process CTR controls the startup and tearing down of the VB5.1 interface*/



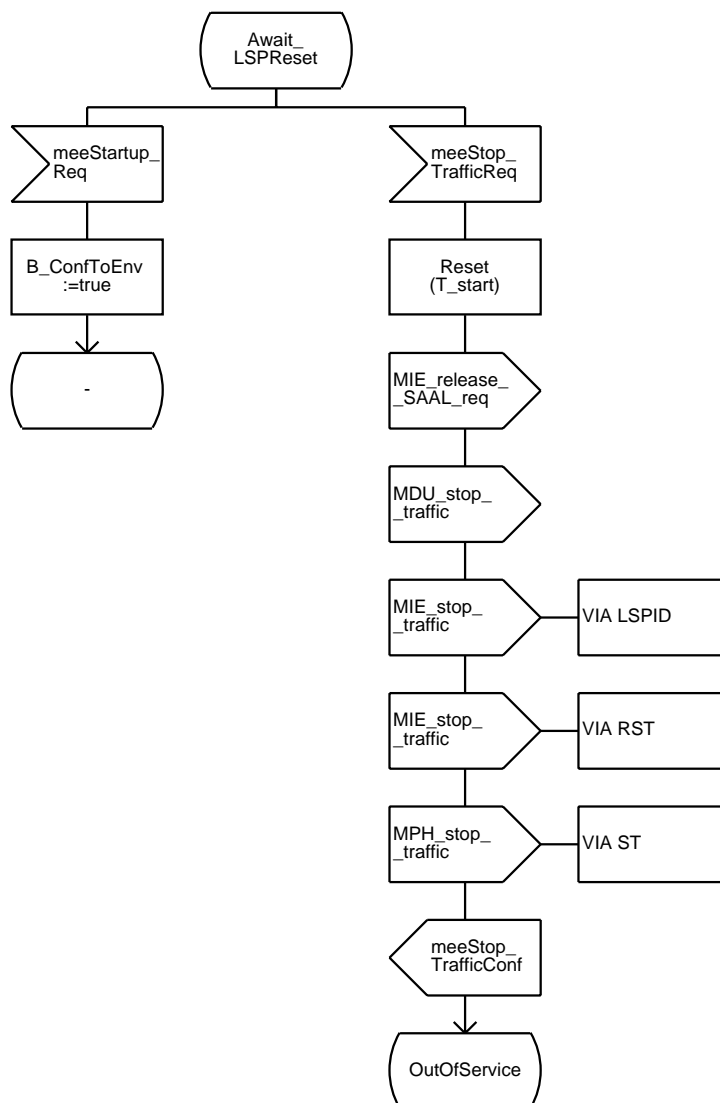
```
/*Process CTR controls the
startup and tearing down
of the VB5.1 interface*/
```



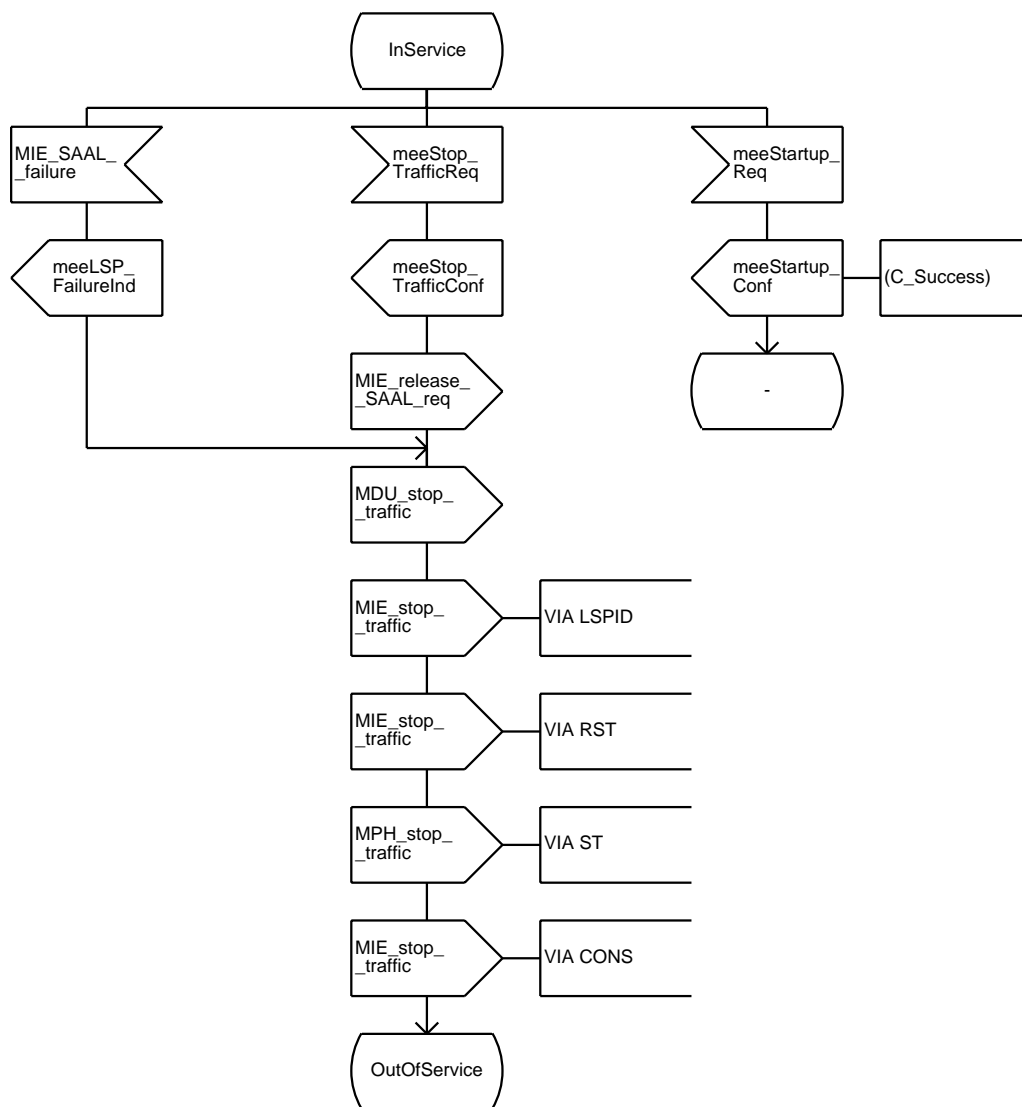
/*Process CTR controls the startup and tearing down of the VB5.1 interface*/

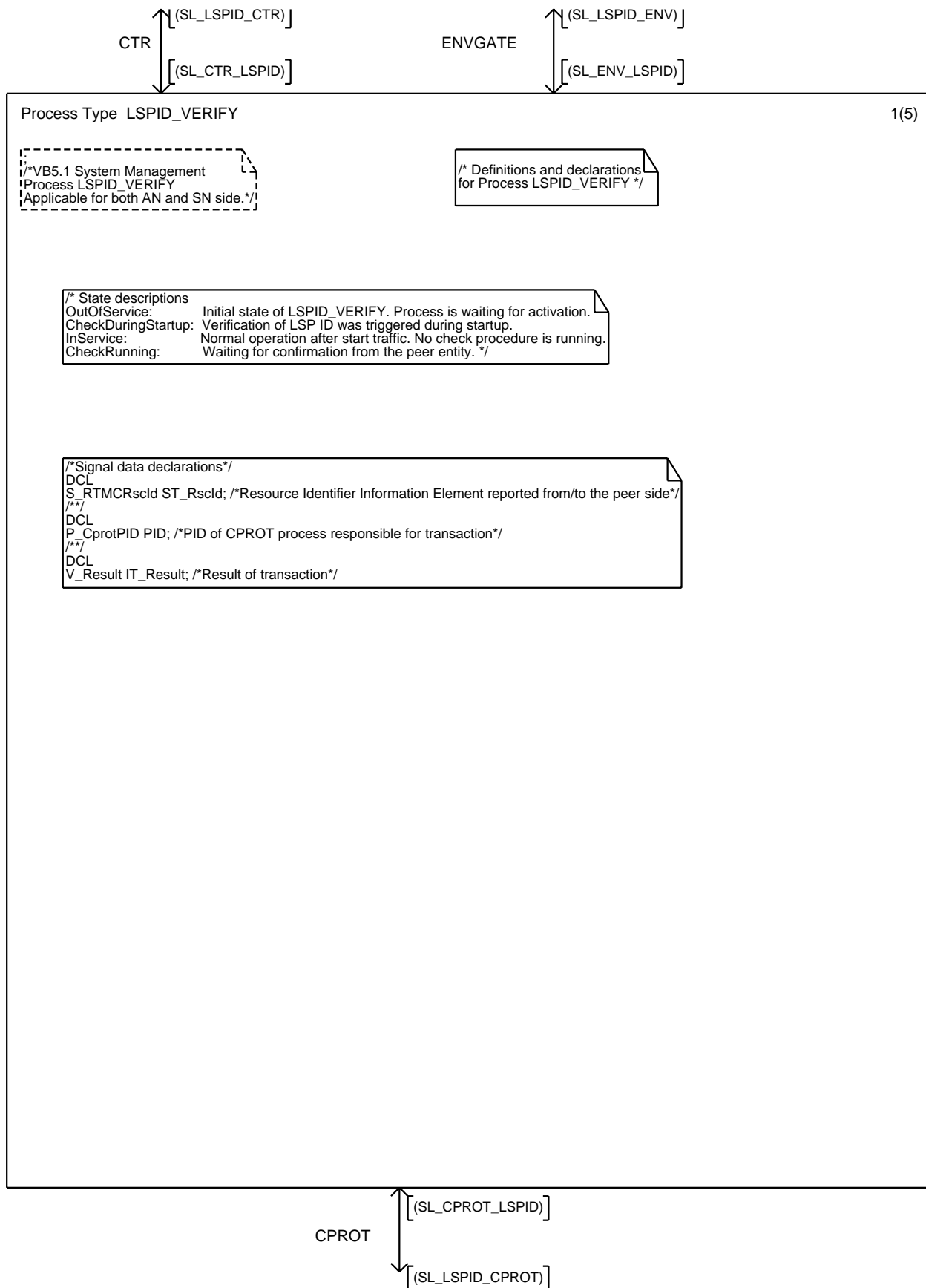


/*Process CTR controls the startup and tearing down of the VB5.1 interface*/

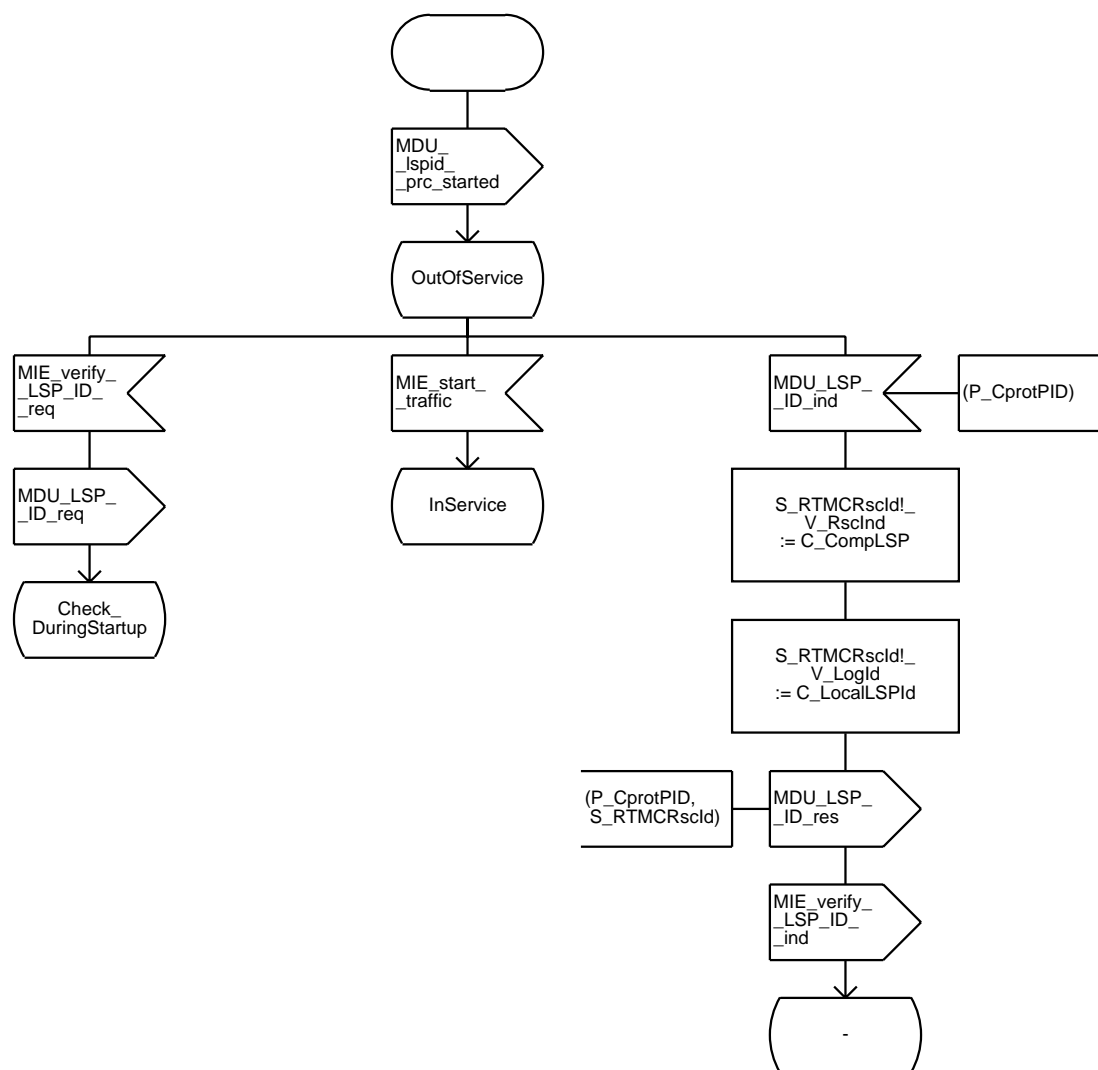


/*Process CTR controls the startup and tearing down of the VB5.1 interface*/

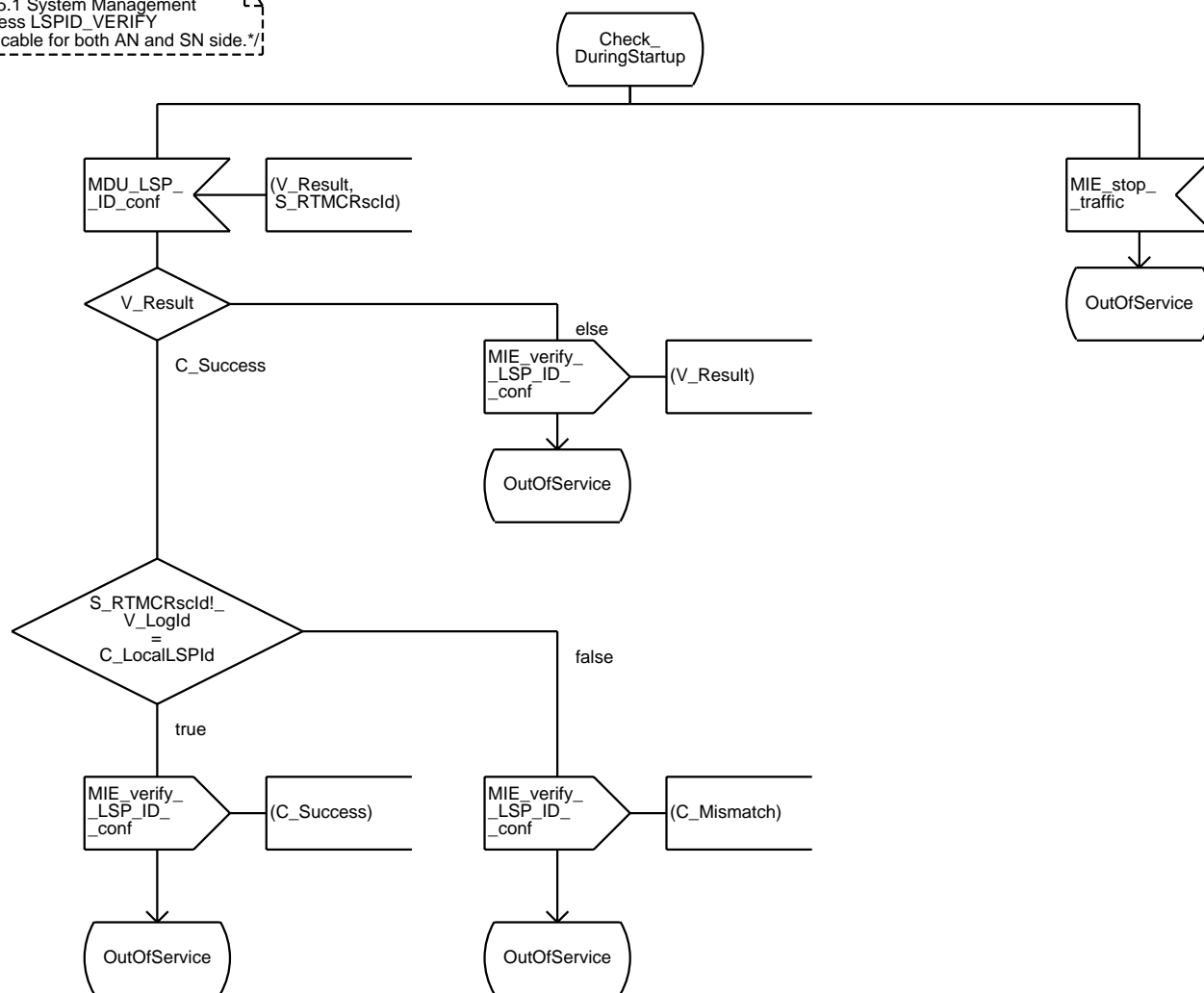




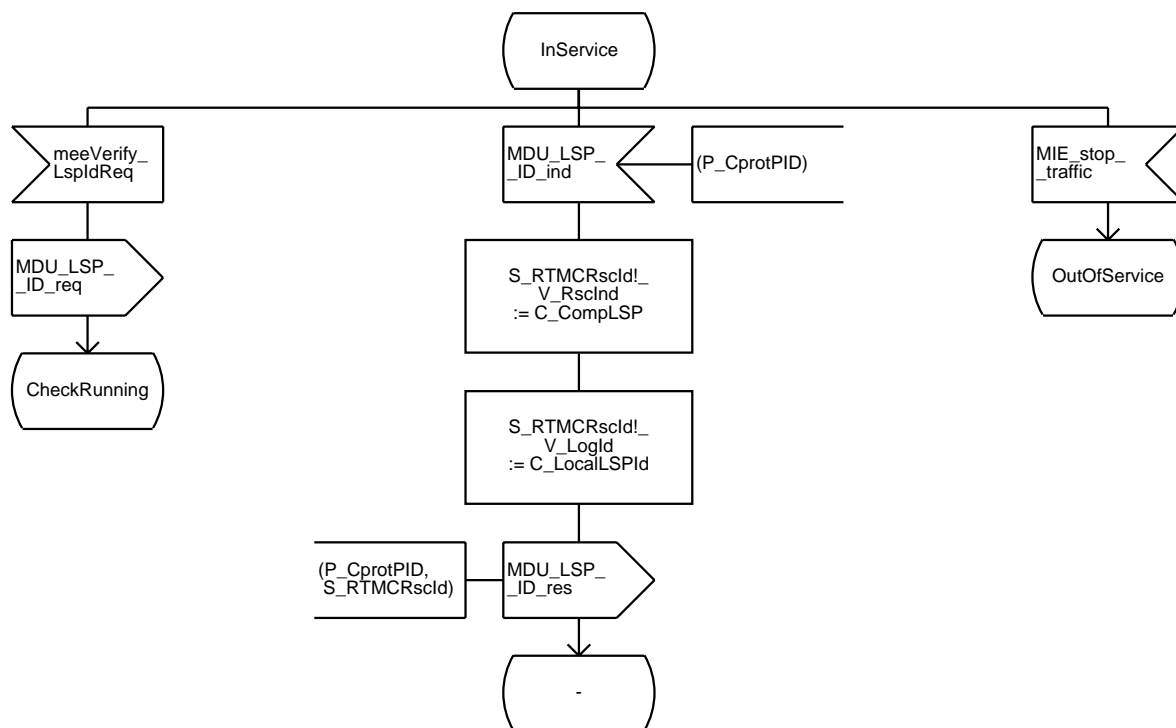
/*VB5.1 System Management
Process LSPID_VERIFY
Applicable for both AN and SN side.*/



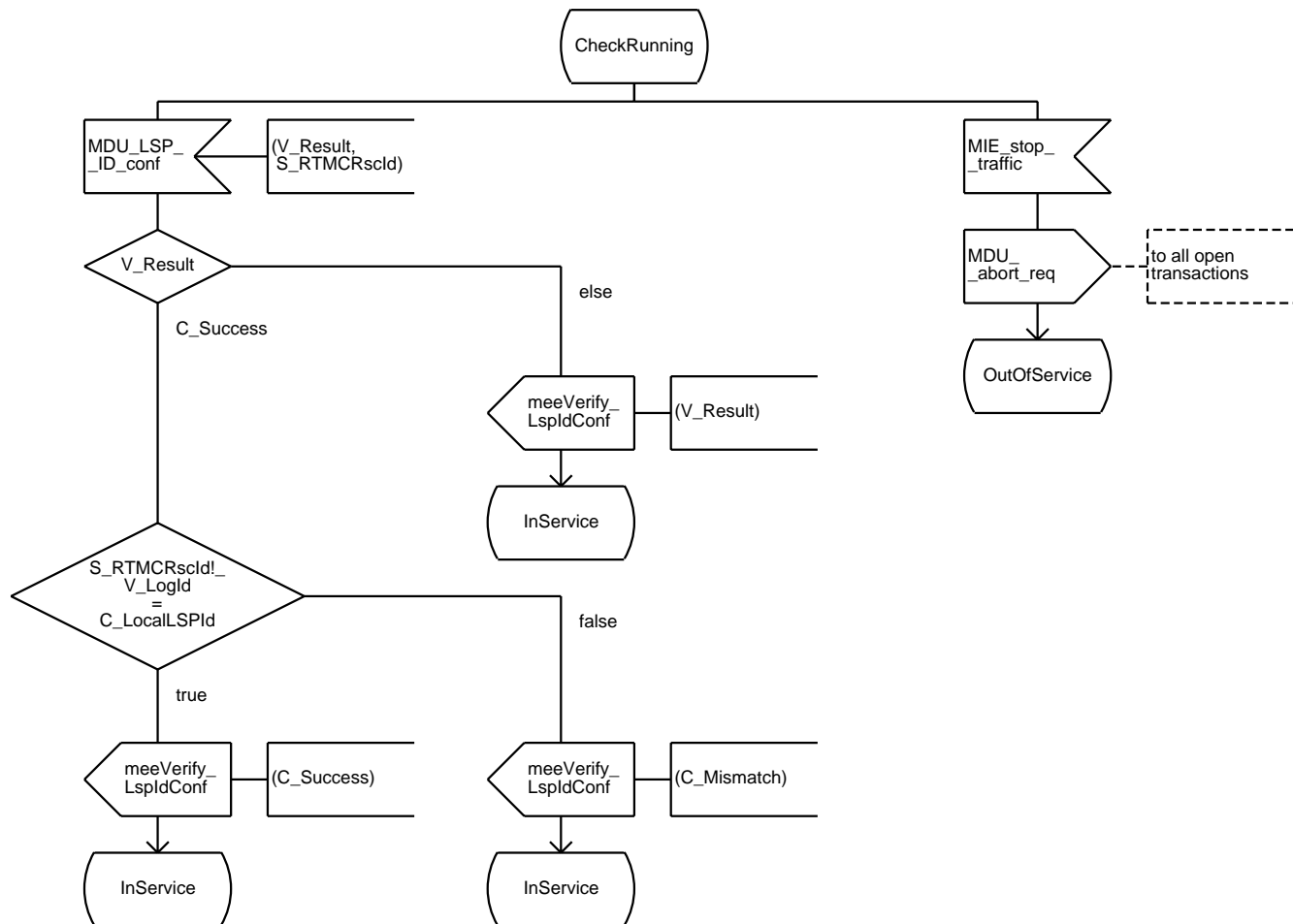
/*VB5.1 System Management
Process LSPID_VERIFY
Applicable for both AN and SN side.*/

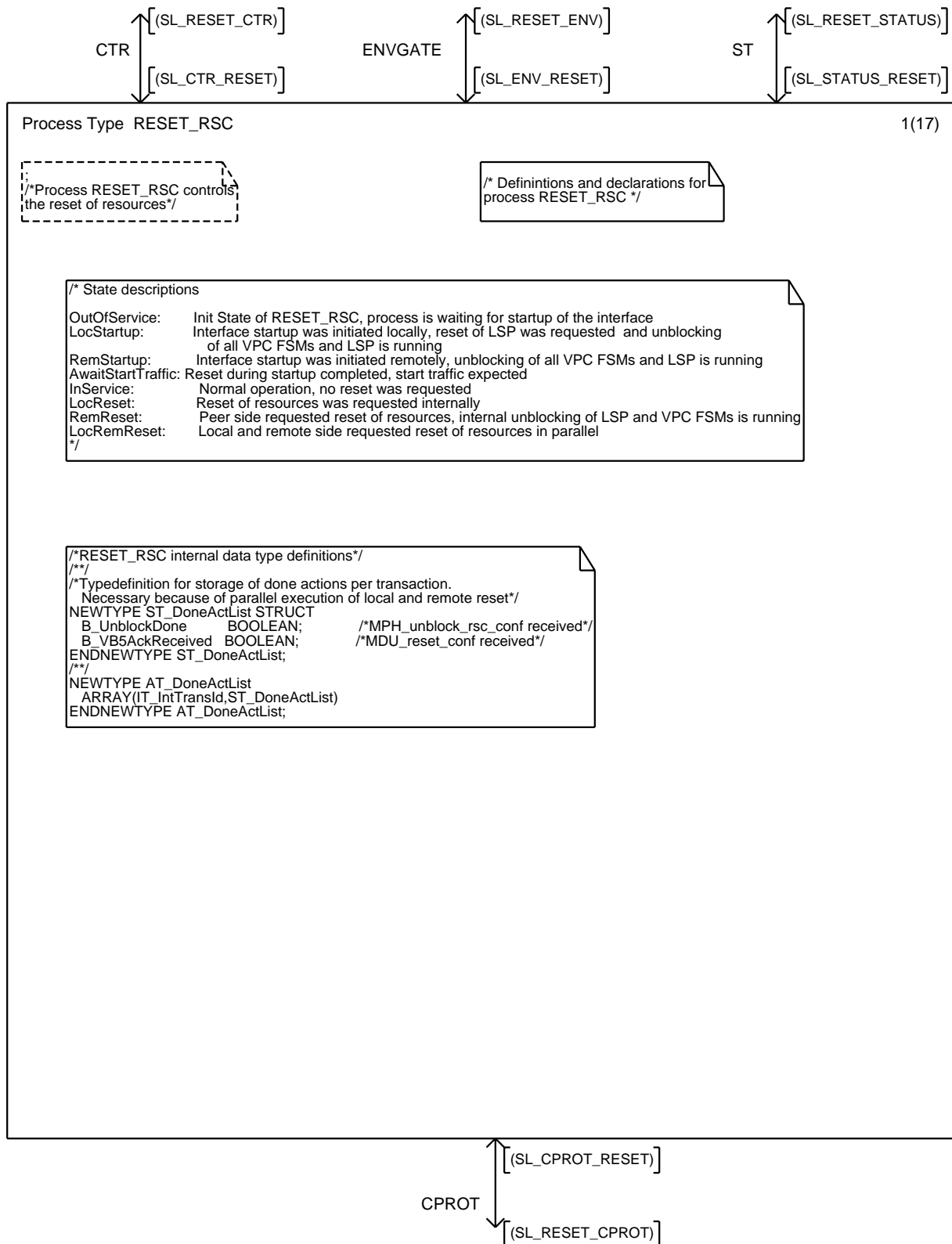


/*VB5.1 System Management
Process LSPID_VERIFY
Applicable for both AN and SN side.*/



/*VB5.1 System Management
Process LSPID_VERIFY
Applicable for both AN and SN side.*/





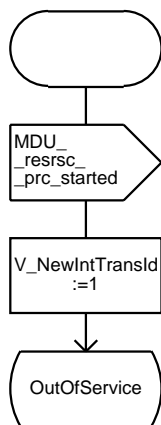
/*Process RESET_RSC controls
the reset of resources*/

/* Definintions and declarations for
process RESET_RSC */

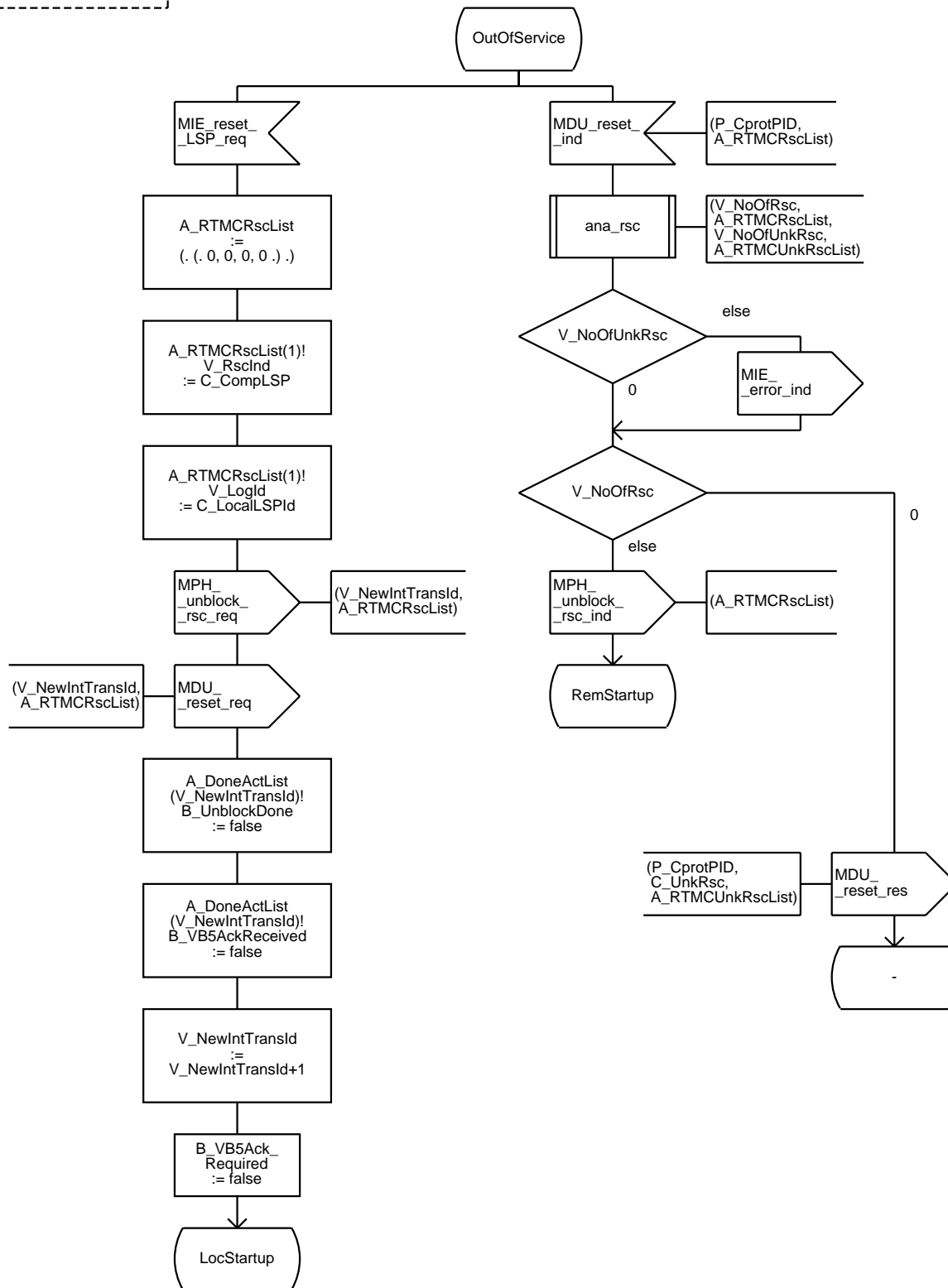
```
/* Internal variables and constants */
/**/
DCL
A_DoneActList AT_DoneActList; /*Done Actions per transaction*/
/**/
DCL
B_VB5AckRequired BOOLEAN; /*MDU_reset_res to be sent*/
/**/
DCL
V_NoOfRsc INTEGER; /*Number of resources*/
/**/
DCL
V_NoOfUnkRsc INTEGER; /*Number of unknown resources*/
/**/
DCL
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_RTMCRRscList 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*/
```

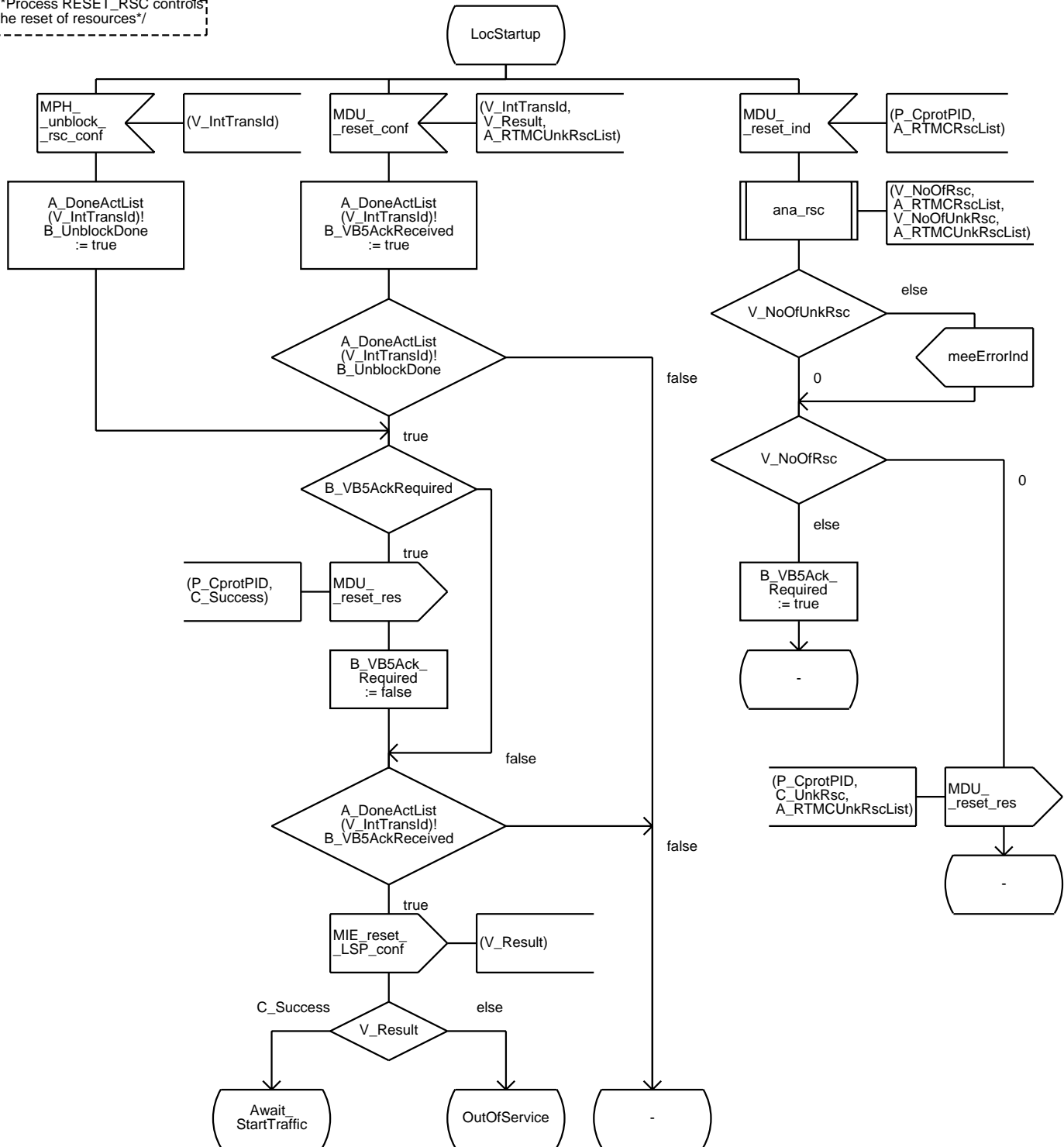
/*Process RESET_RSC controls
the reset of resources*/



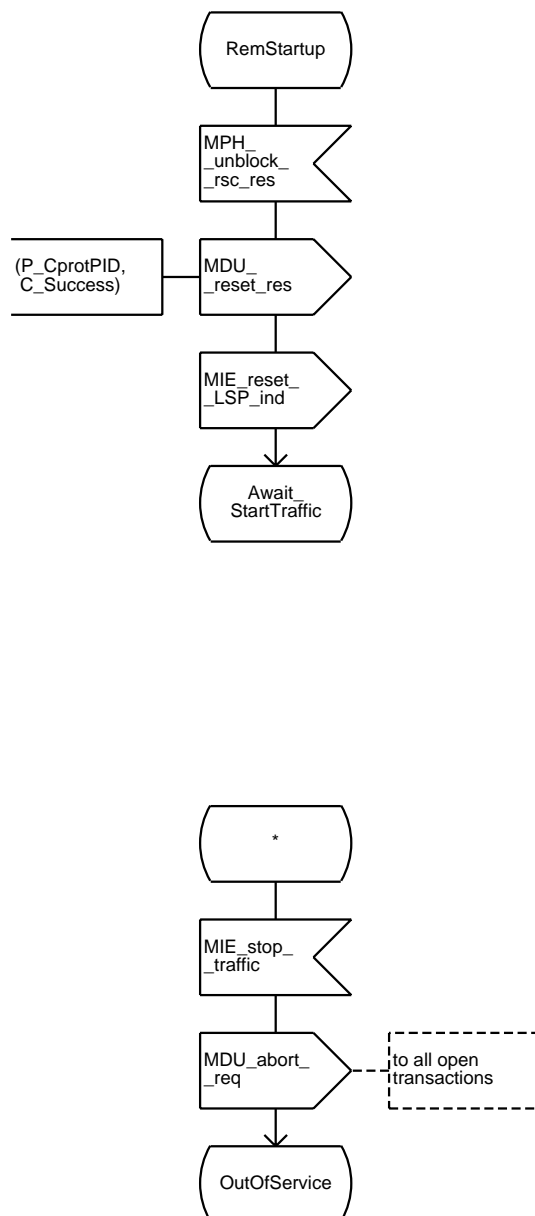
/*Process RESET_RSC controls
the reset of resources*/



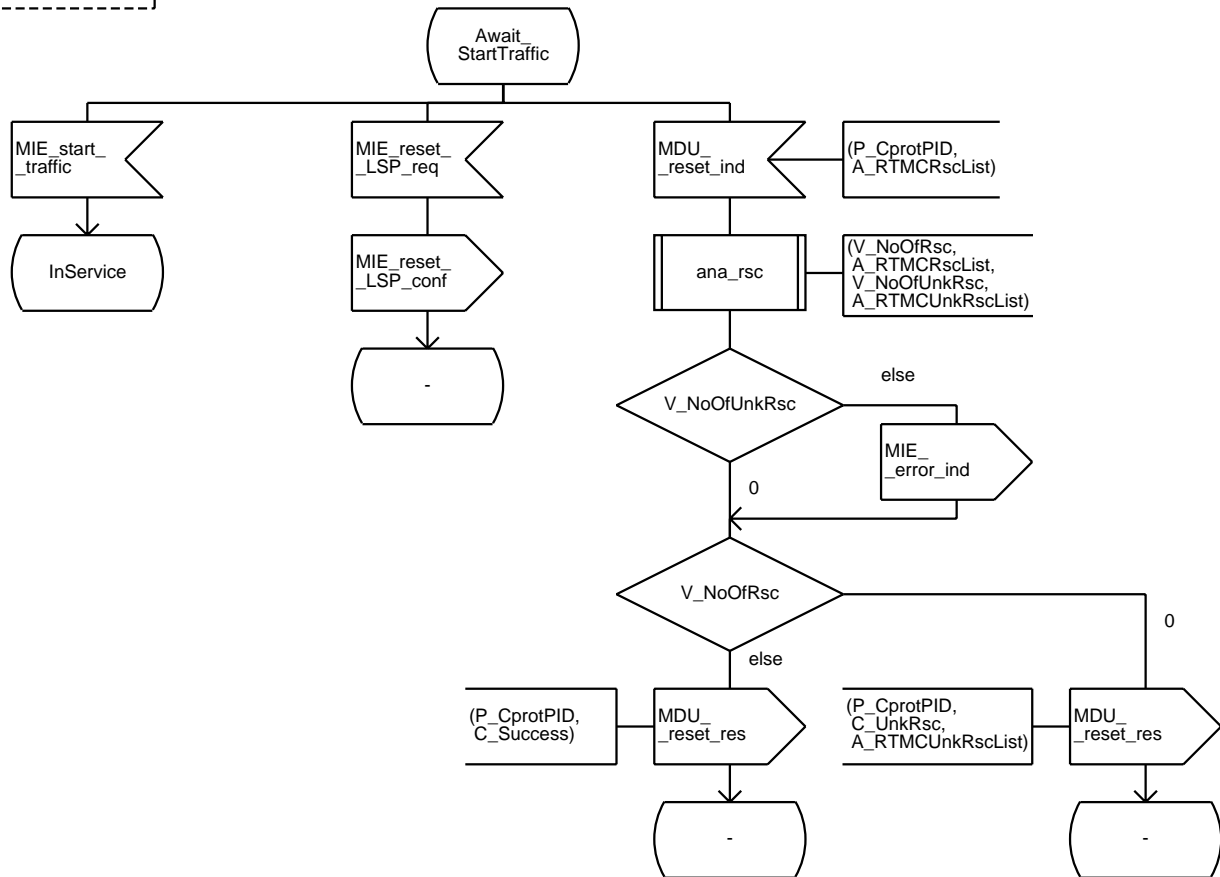
/*Process RESET_RSC controls
the reset of resources*/



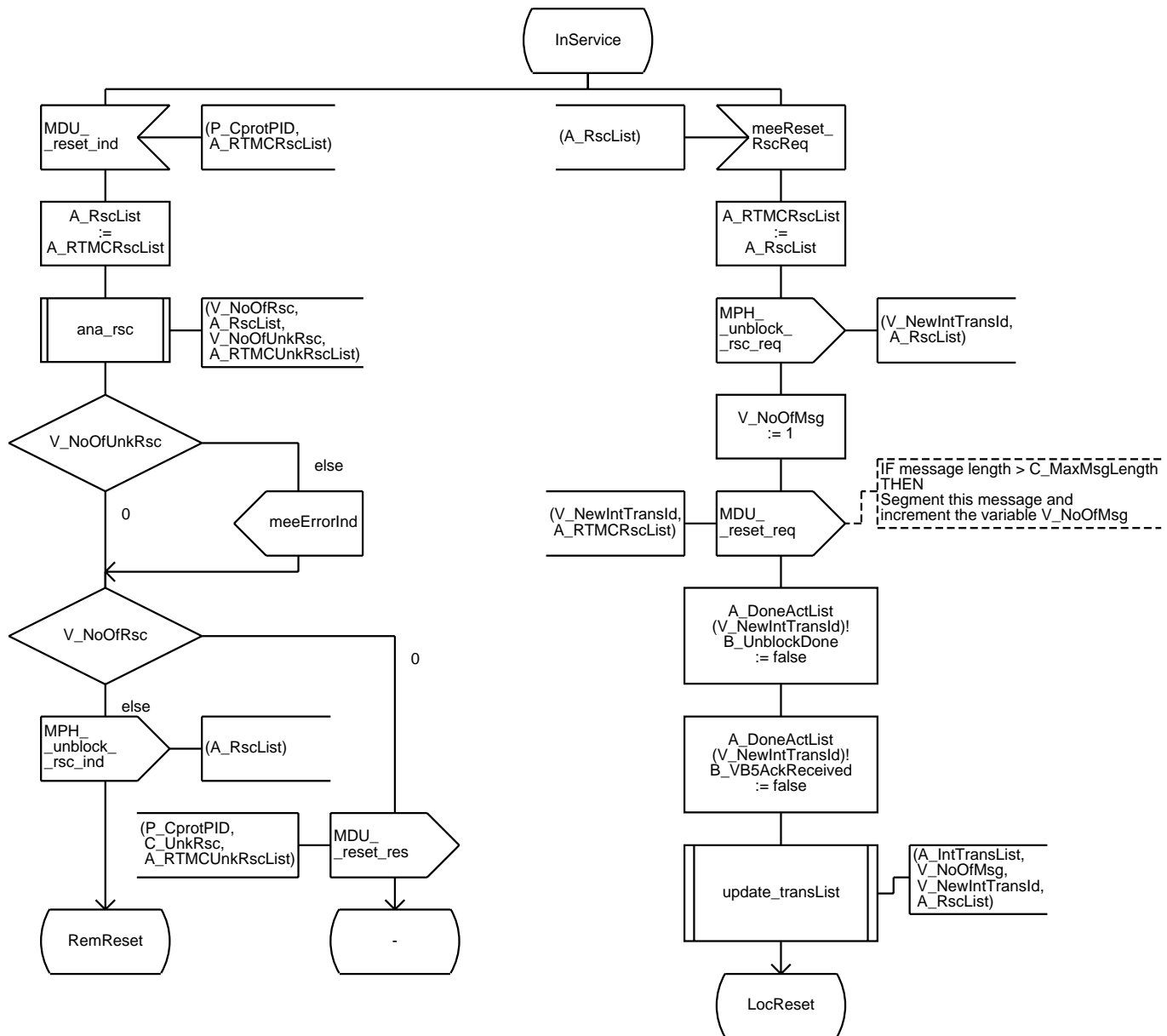
/*Process RESET_RSC controls
the reset of resources*/



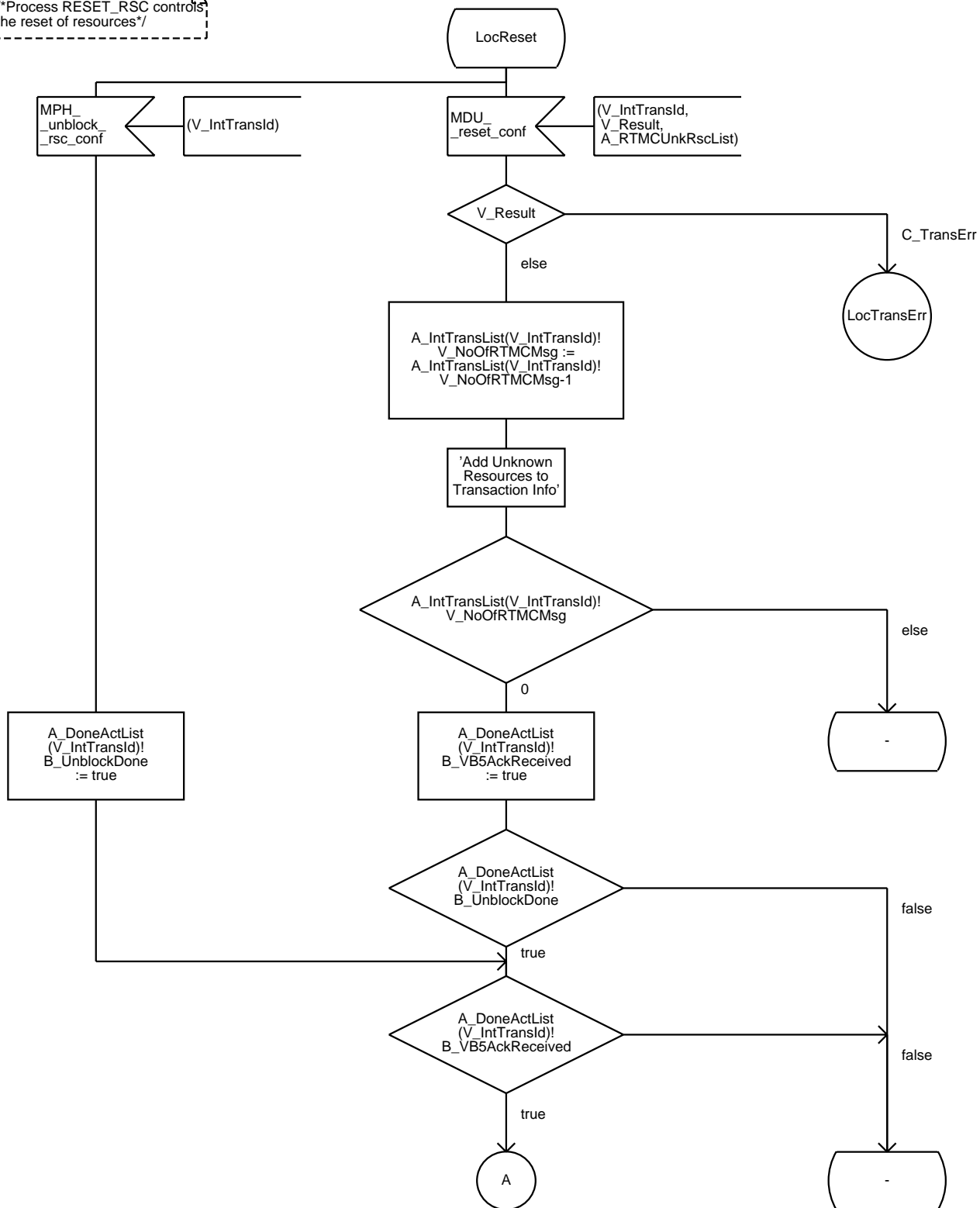
/*Process RESET_RSC controls
the reset of resources*/



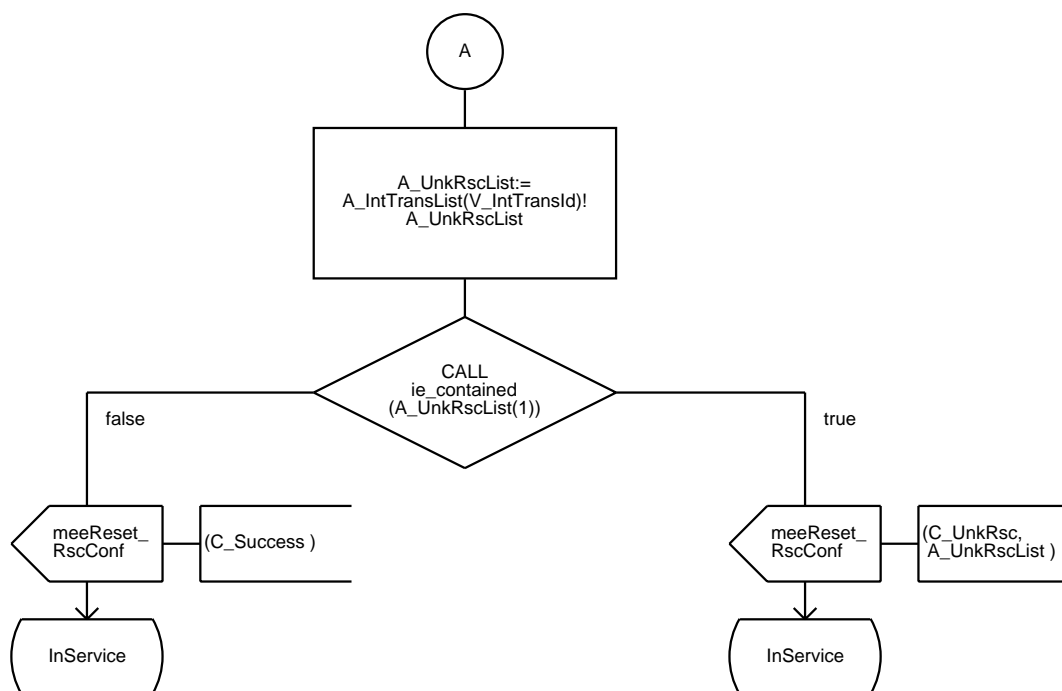
/*Process RESET_RSC controls
the reset of resources*/



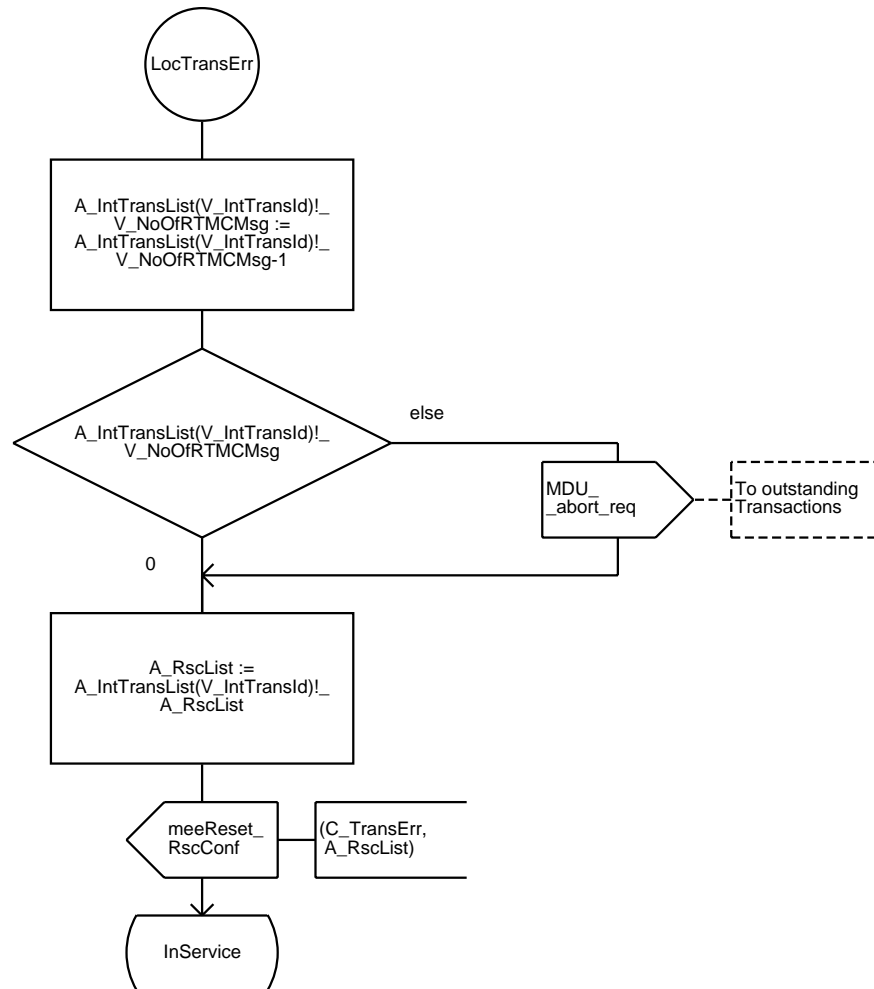
/*Process RESET_RSC controls
the reset of resources*/



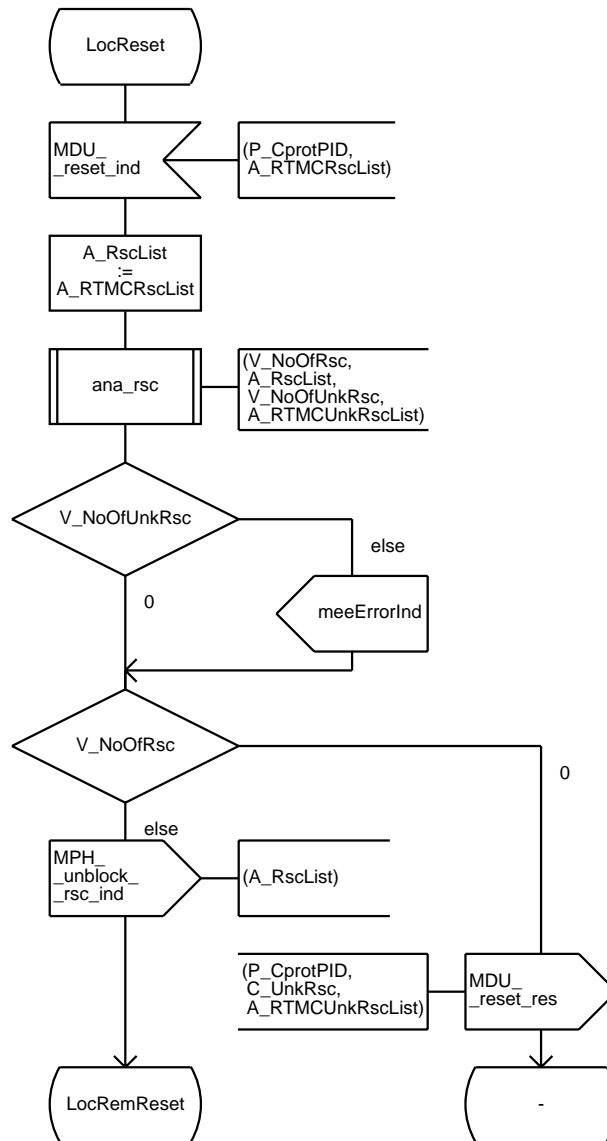
/*Process RESET_RSC controls
the reset of resources*/



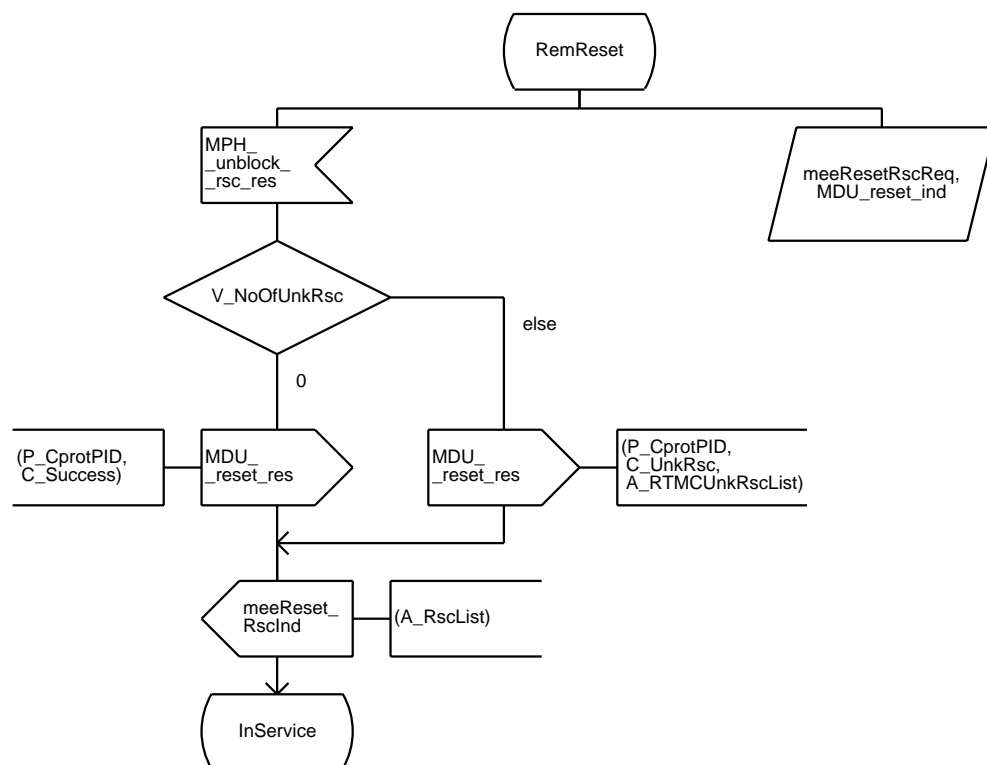
/*Process RESET_RSC controls
the reset of resources*/



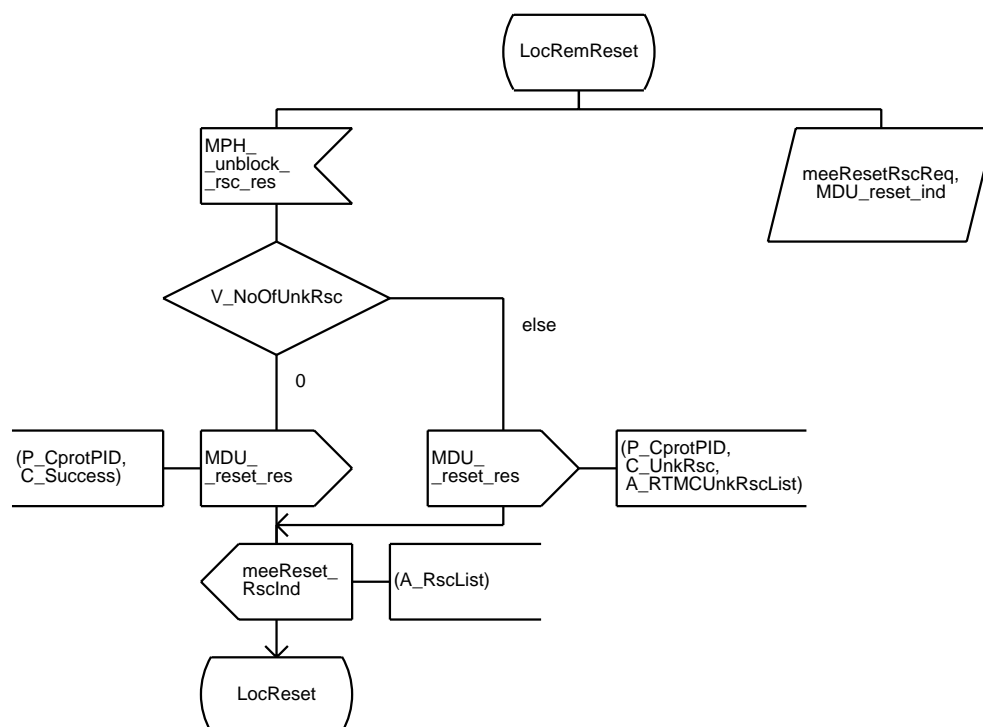
/*Process RESET_RSC controls
the reset of resources*/

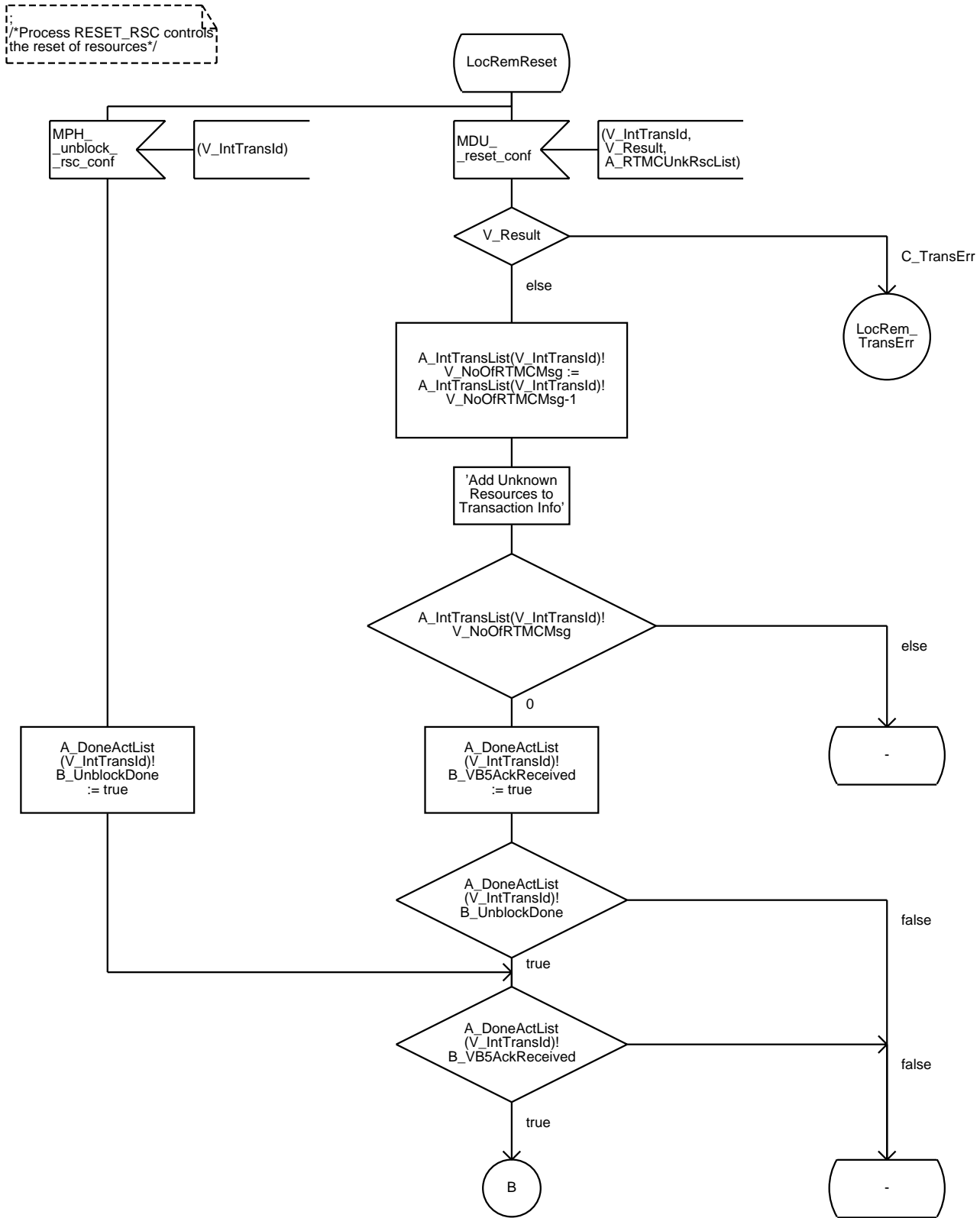


/*Process RESET_RSC controls
the reset of resources*/

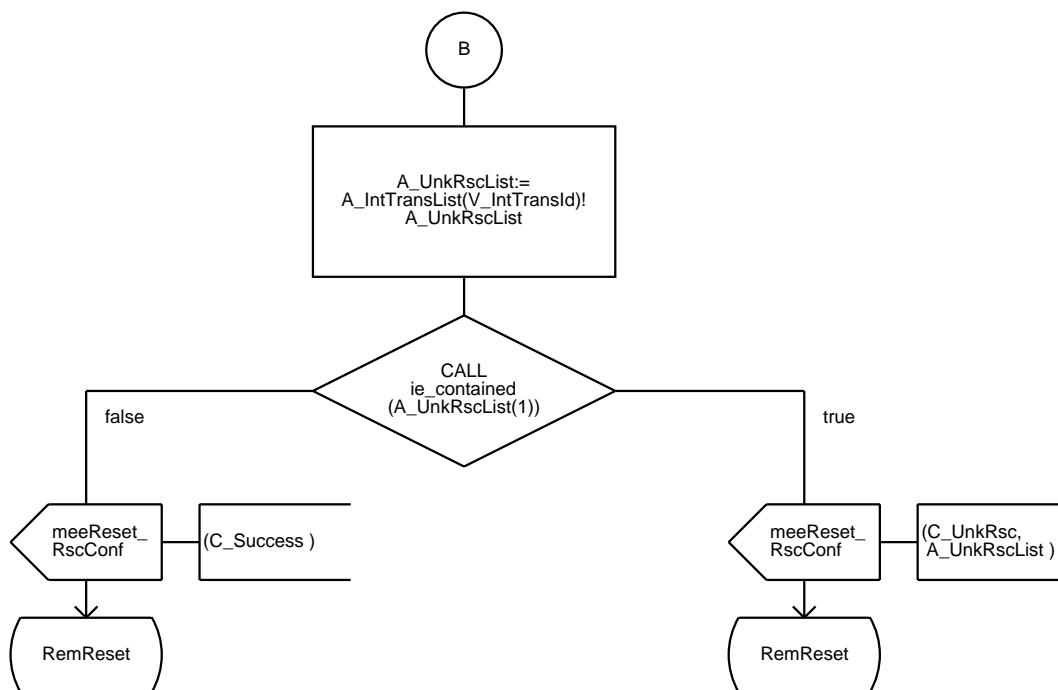


/*Process RESET_RSC controls
the reset of resources*/

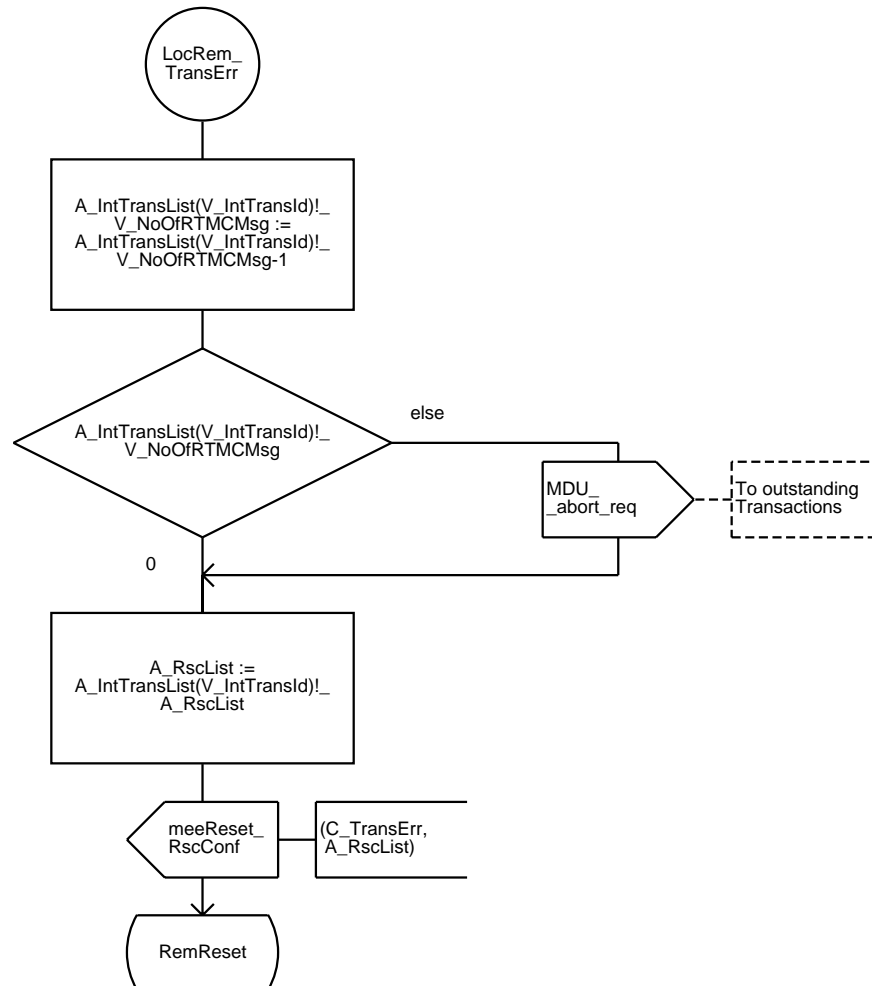


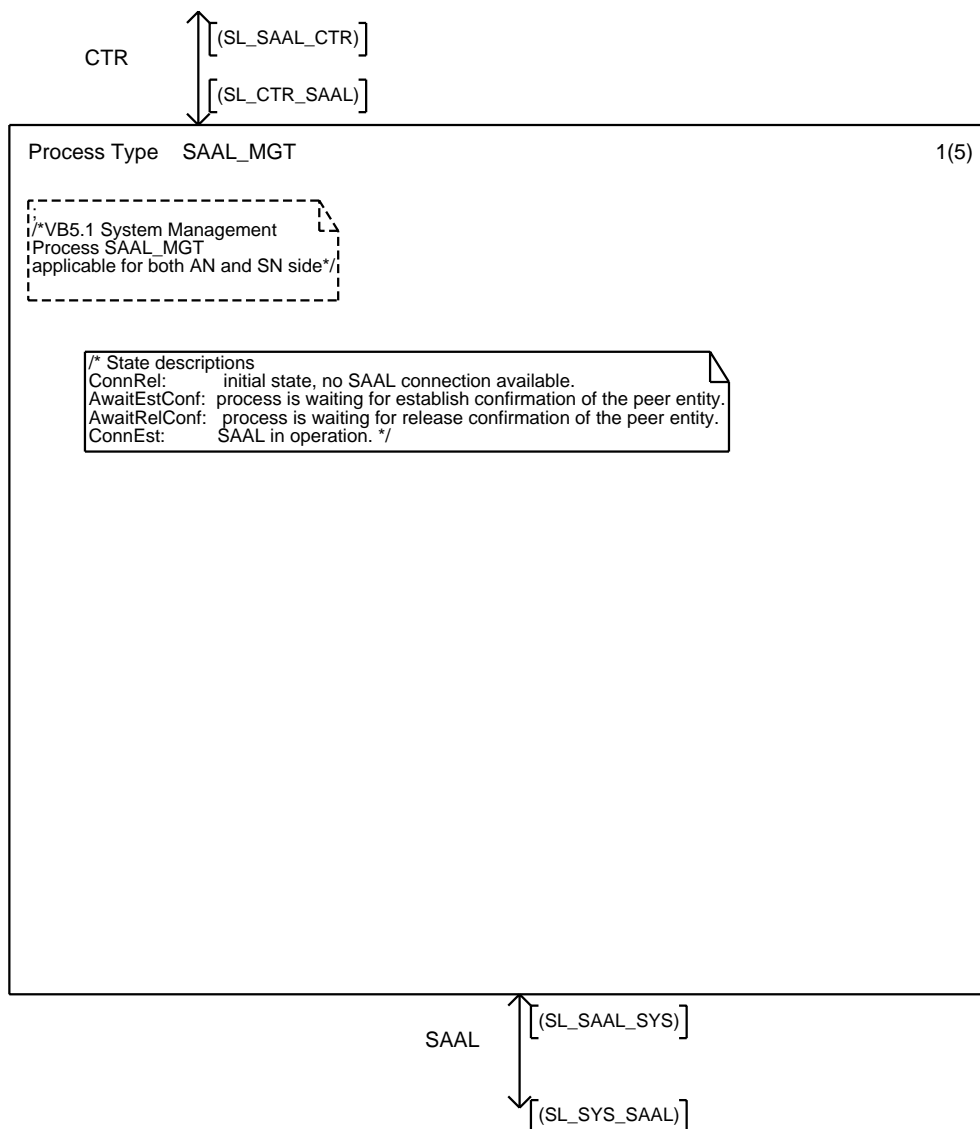


/*Process RESET_RSC controls
the reset of resources*/

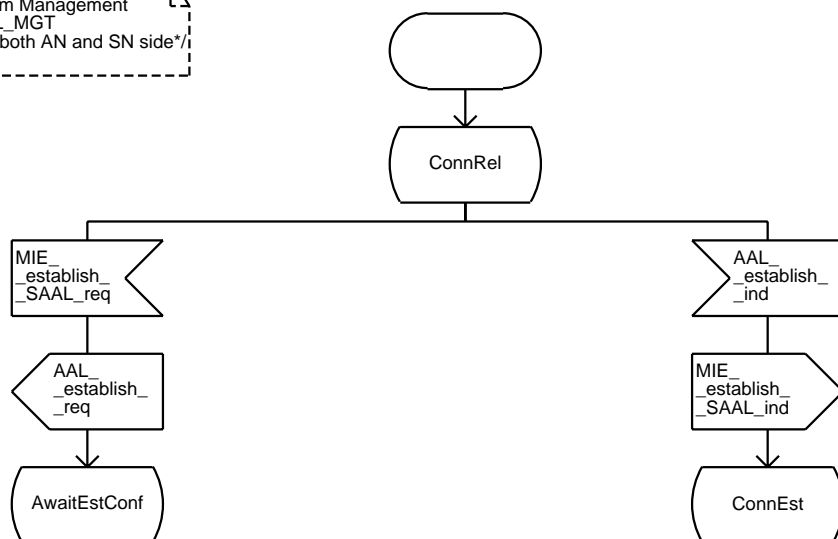


/*Process RESET_RSC controls
the reset of resources*/

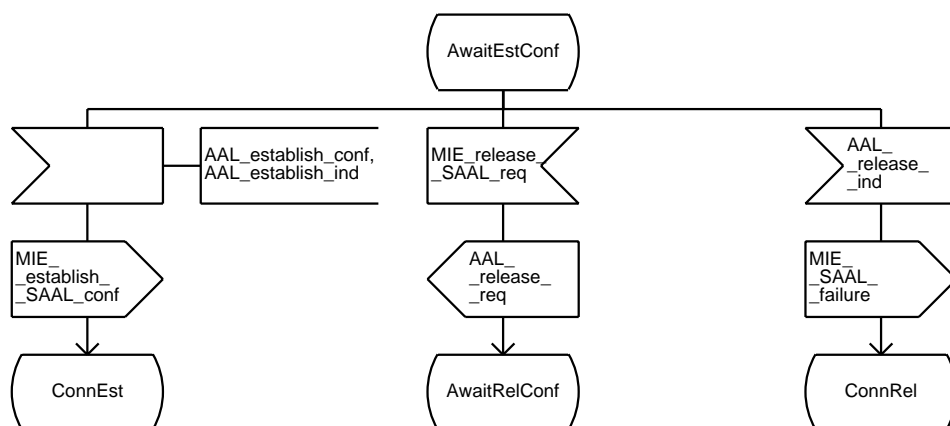




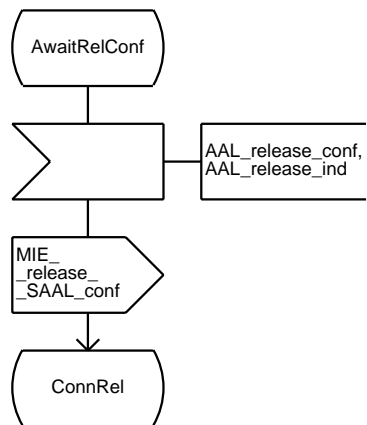
/*VB5.1 System Management
Process SAAL_MGT
applicable for both AN and SN side*/



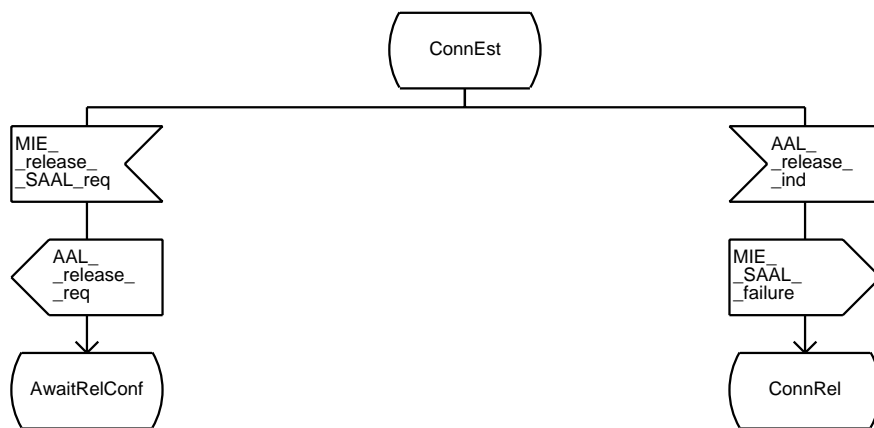
/*VB5.1 System Management
Process SAAL_MGT
applicable for both AN and SN side*/

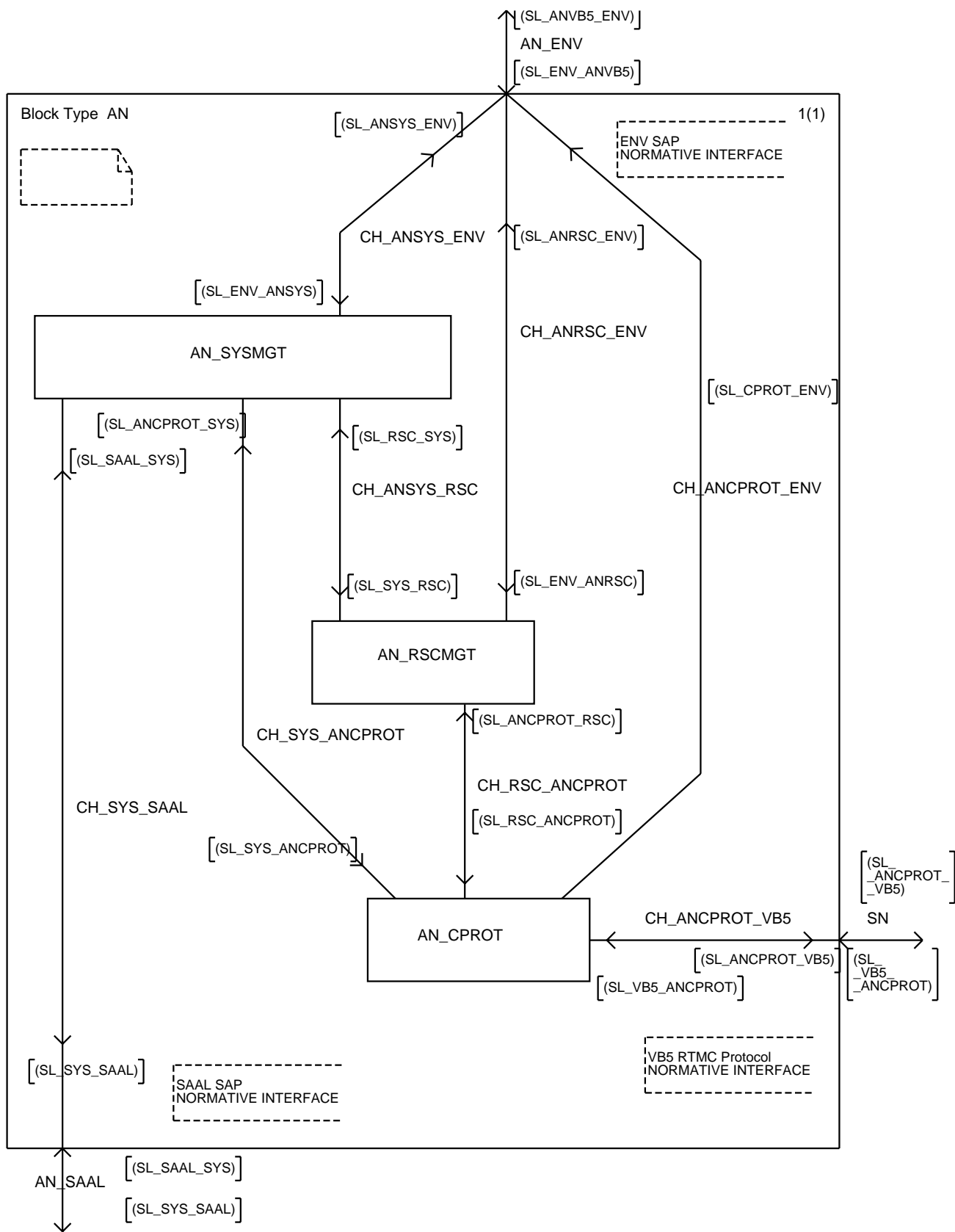


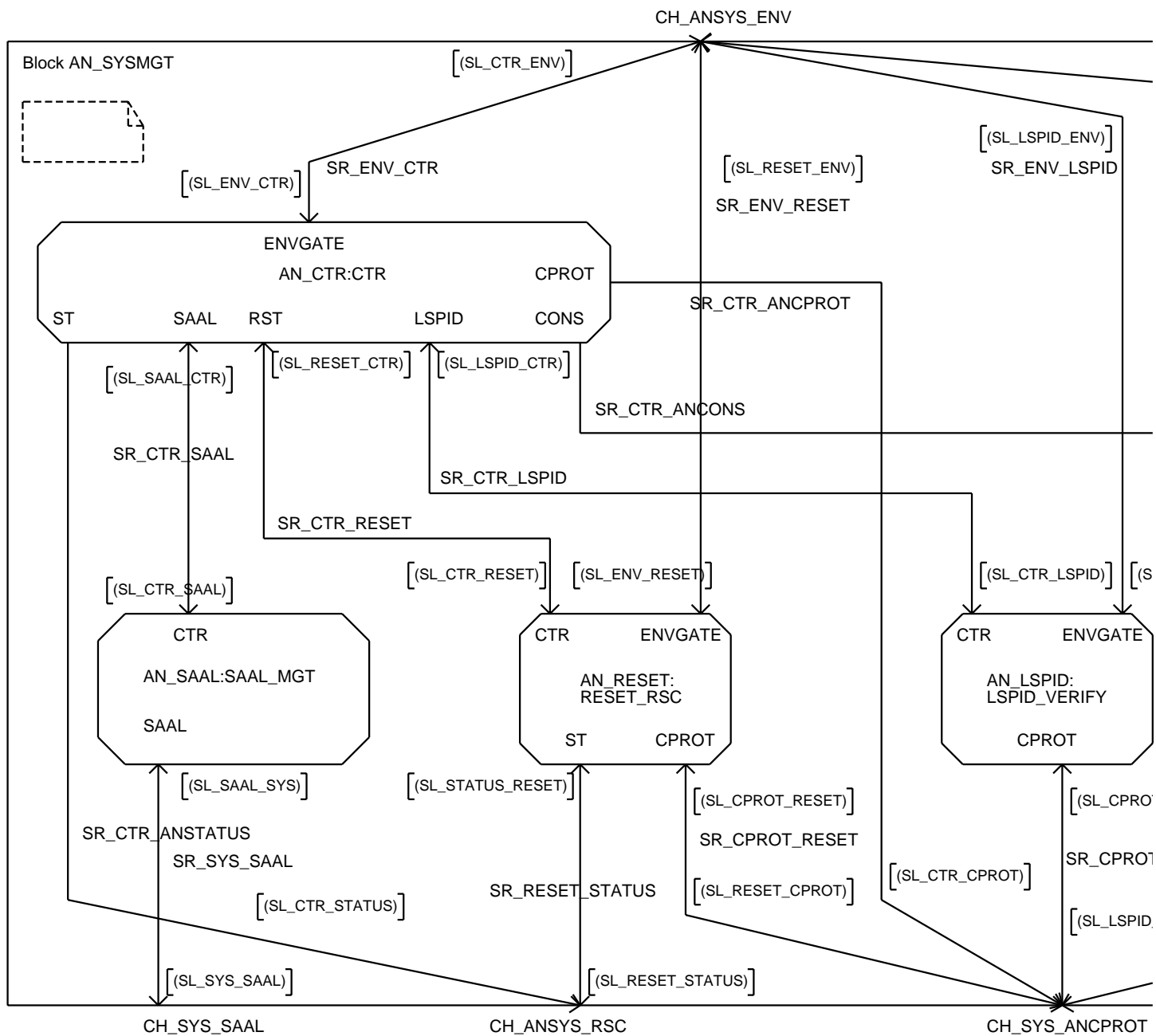
/*VB5.1 System Management
Process SAAL_MGT
applicable for both AN and SN side*/

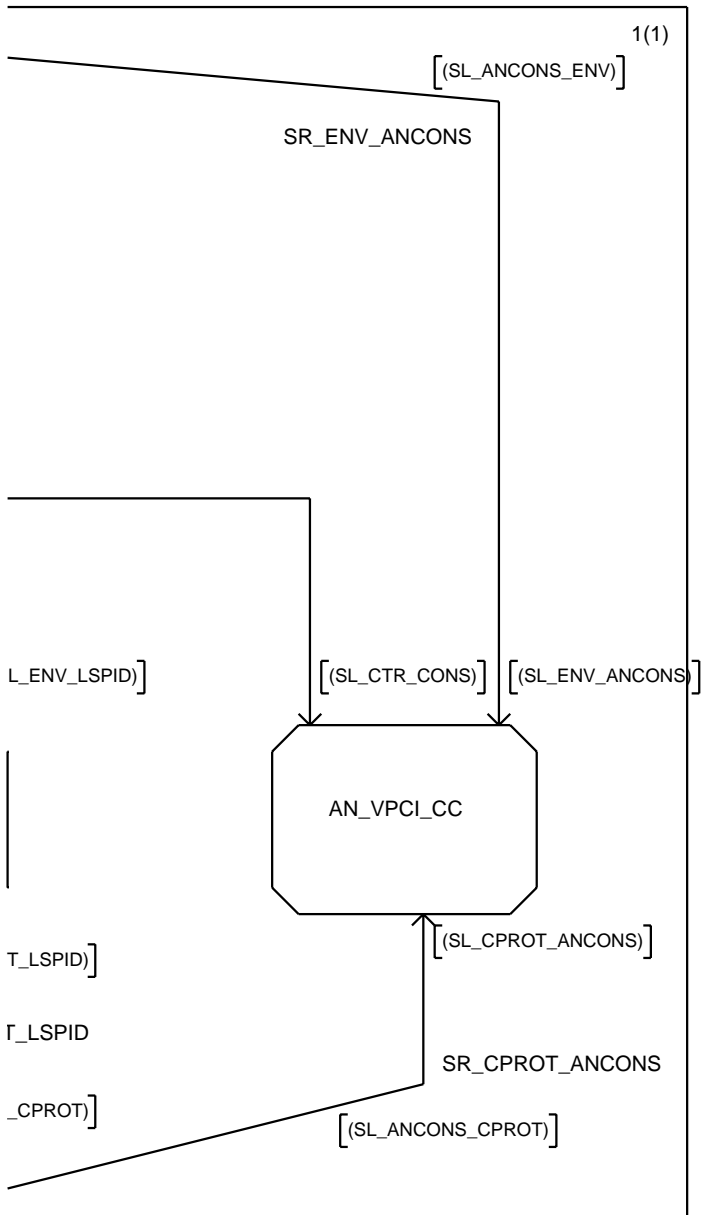


/*VB5.1 System Management
Process SAAL_MGT
applicable for both AN and SN side*/









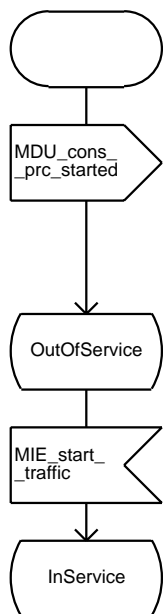
/*VB5.1 System Management
Process AN_VPCI_CC
Applicable to AN side*/

/* Definitions and declarations
for process AN_VPCI_CC */

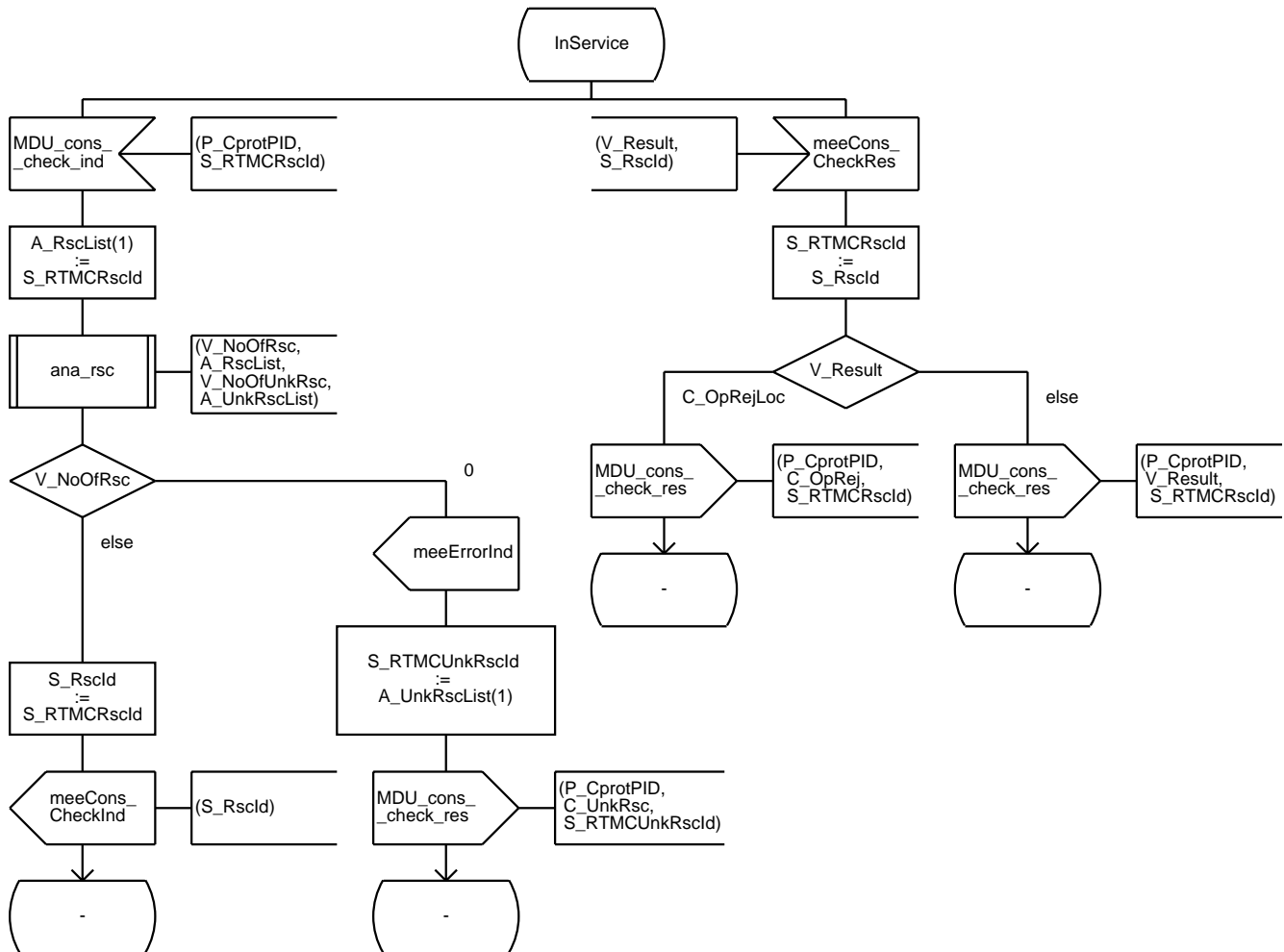
/* State descriptions
OutOfService: Initial state. Process is waiting for activation from CTR process.
InService: Process started, handling all events
*/

/*Signal data declarations*/
DCL
S_RscId ST_RscId;
S_RTMC RscId 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*/

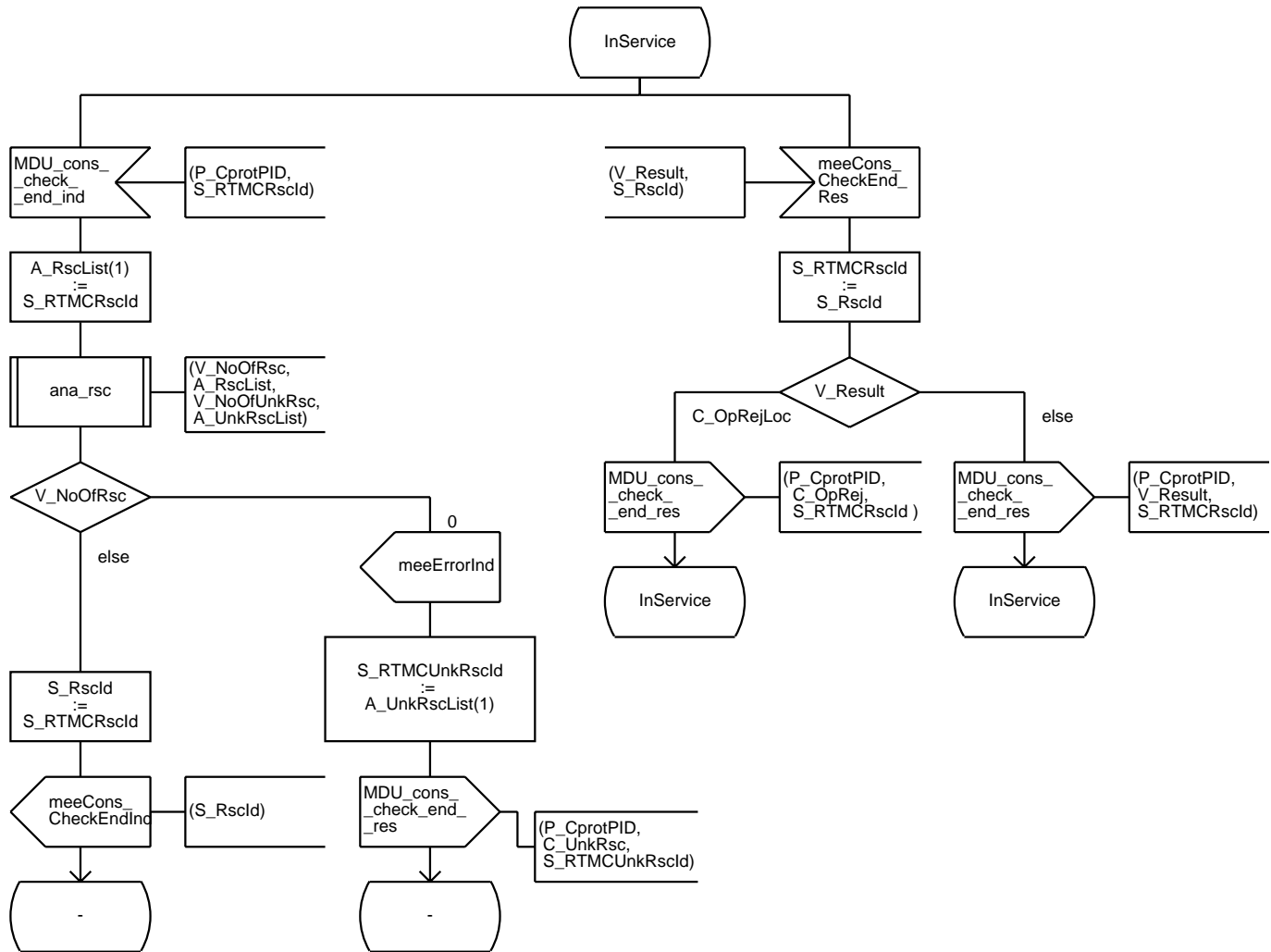
/*VB5.1 System Management
Process AN_VPCI_CC
Applicable to AN side*/



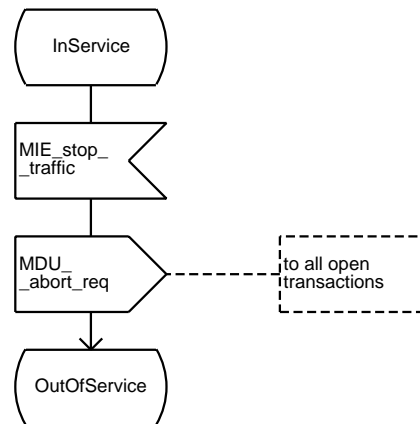
/*VB5.1 System Management
Process AN_VPCI_CC
Applicable to AN side*/

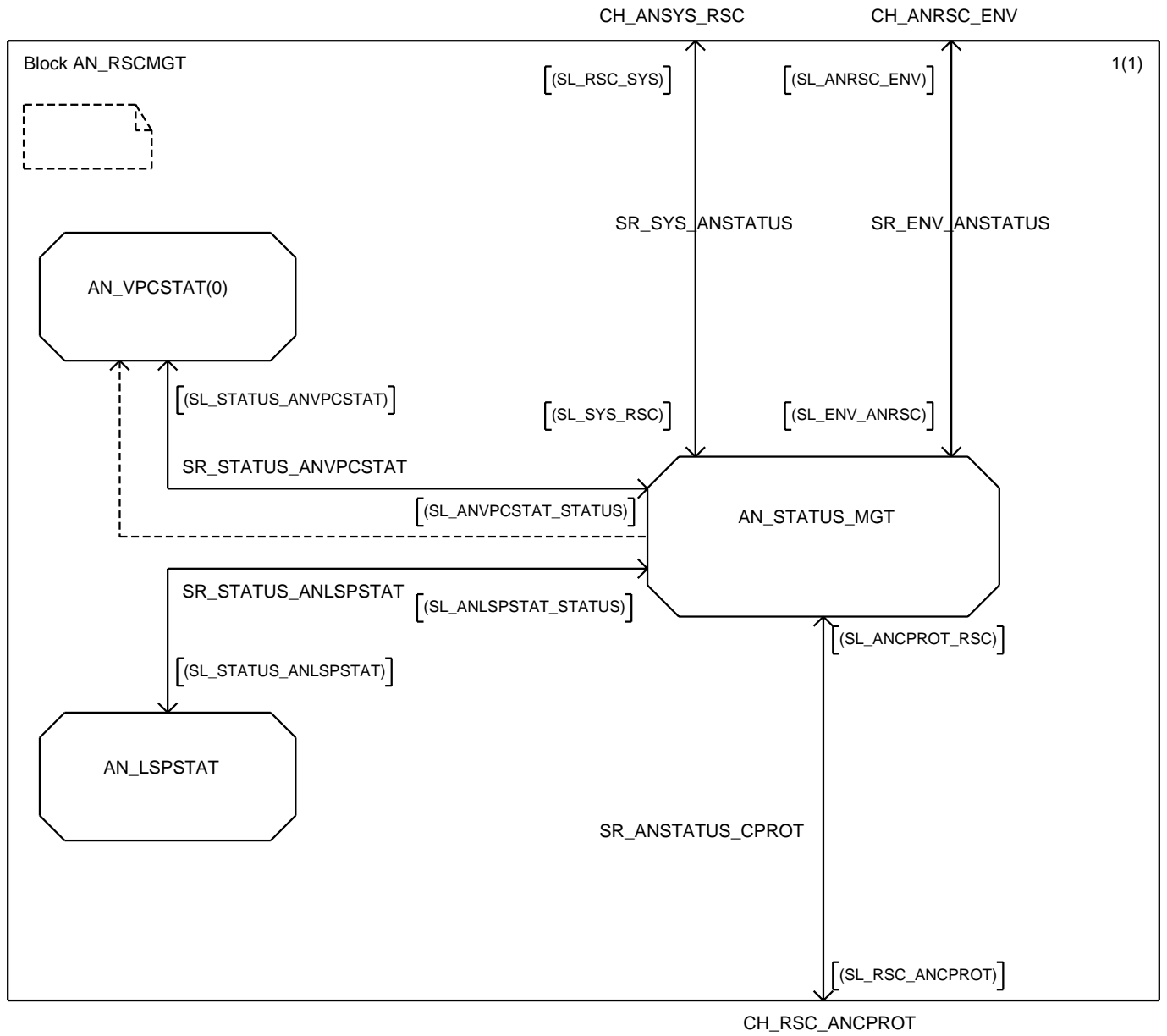


/*VB5.1 System Management
Process AN_VPCI_CC
Applicable to AN side*/



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






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

```
/* Definitions 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
AwaitBIFFSMAck:     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_Bi      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_RTMCRCscList AT_RscList;
  A_RTMCBIRscList AT_RTMCBIRscList;
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*/

/* Definitions and declarations for
AN_STATUS_MGT */

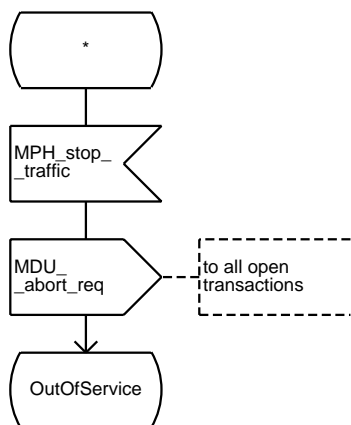
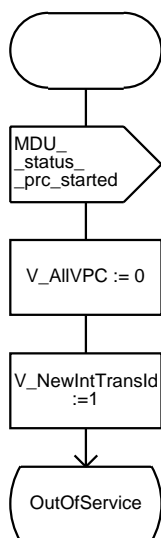
```
/*AN_STATUS_MGT internal variables and constants */
/**
/*Variables for co-ordinating "unlock_rsc" procedure (e.g. during interface startup or reset of LSP)*/
DCL
B_VPCUblDone, /*MIE_VPC_unlock_conf received from all requested VPC FSMs*/
B_LSPUblDone /*MIE_LSP_unlock_conf received*/
BOOLEAN;
/**
/*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*/

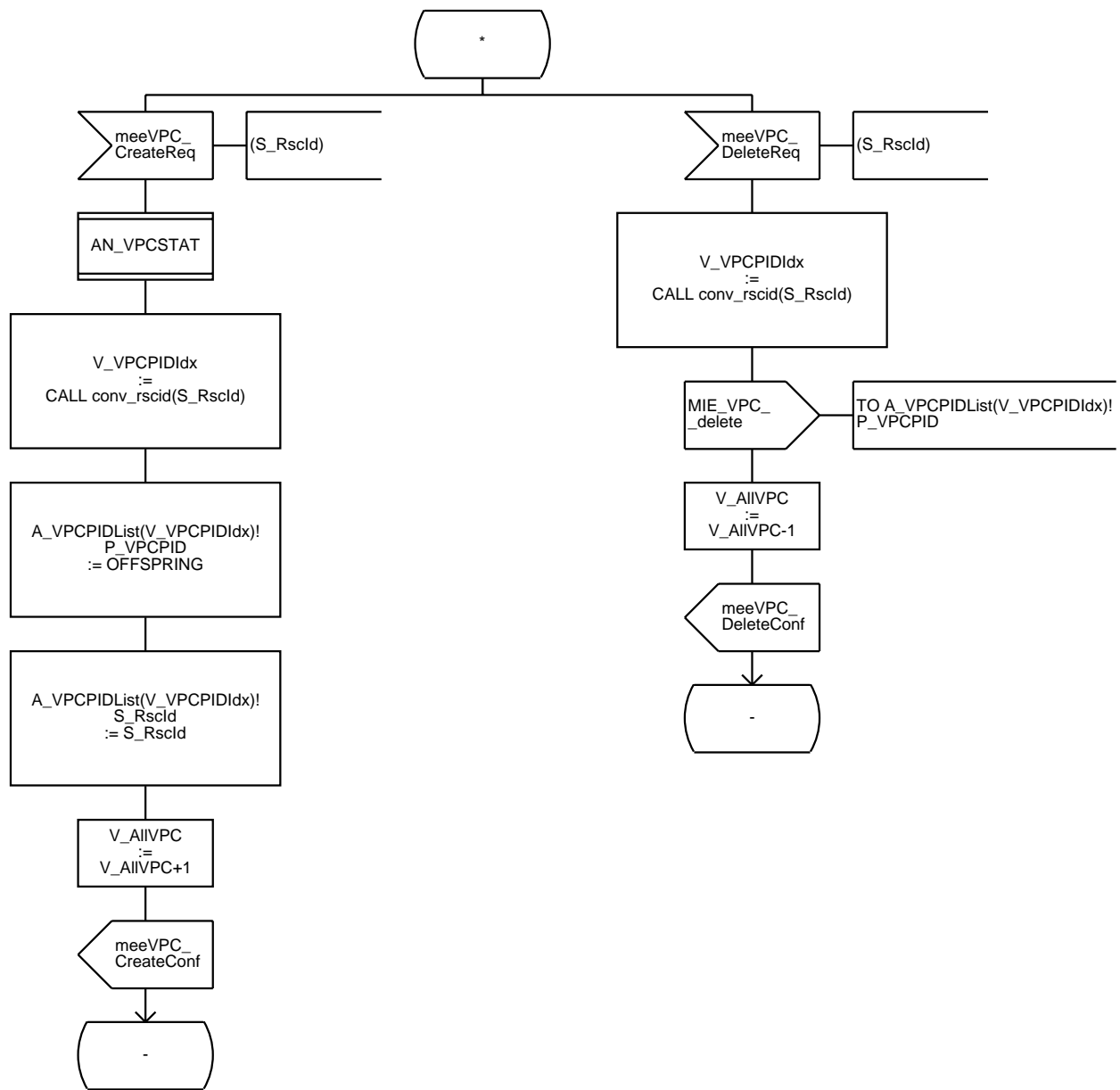
/* Definintions and declarations for
AN_STATUS_MGT */

```
/*Signal data declarations*/
/**/
/*Resource identifier, used for identifying VPCSTAT/LSPSTAT process*/
DCL S_RscId ST_RscId;
/**/
/*List of Resource identifiers*/
DCL A_RTMC RscList 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_BIRscList AT_BIRscList;
/**/
/*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;
```

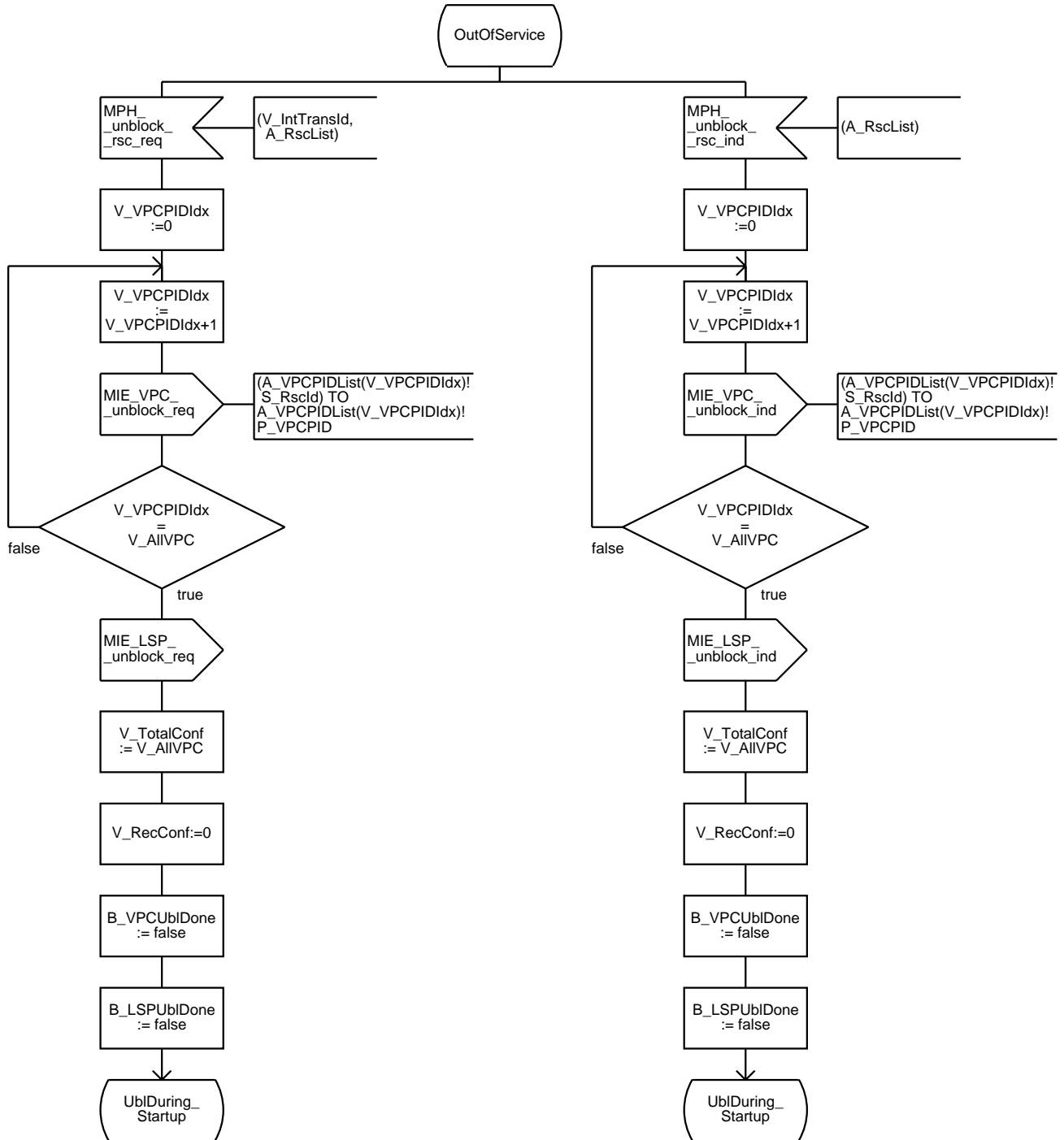
/*AN_STATUS_MGT co-ordinates
local status changes of VPCs and LSP*/



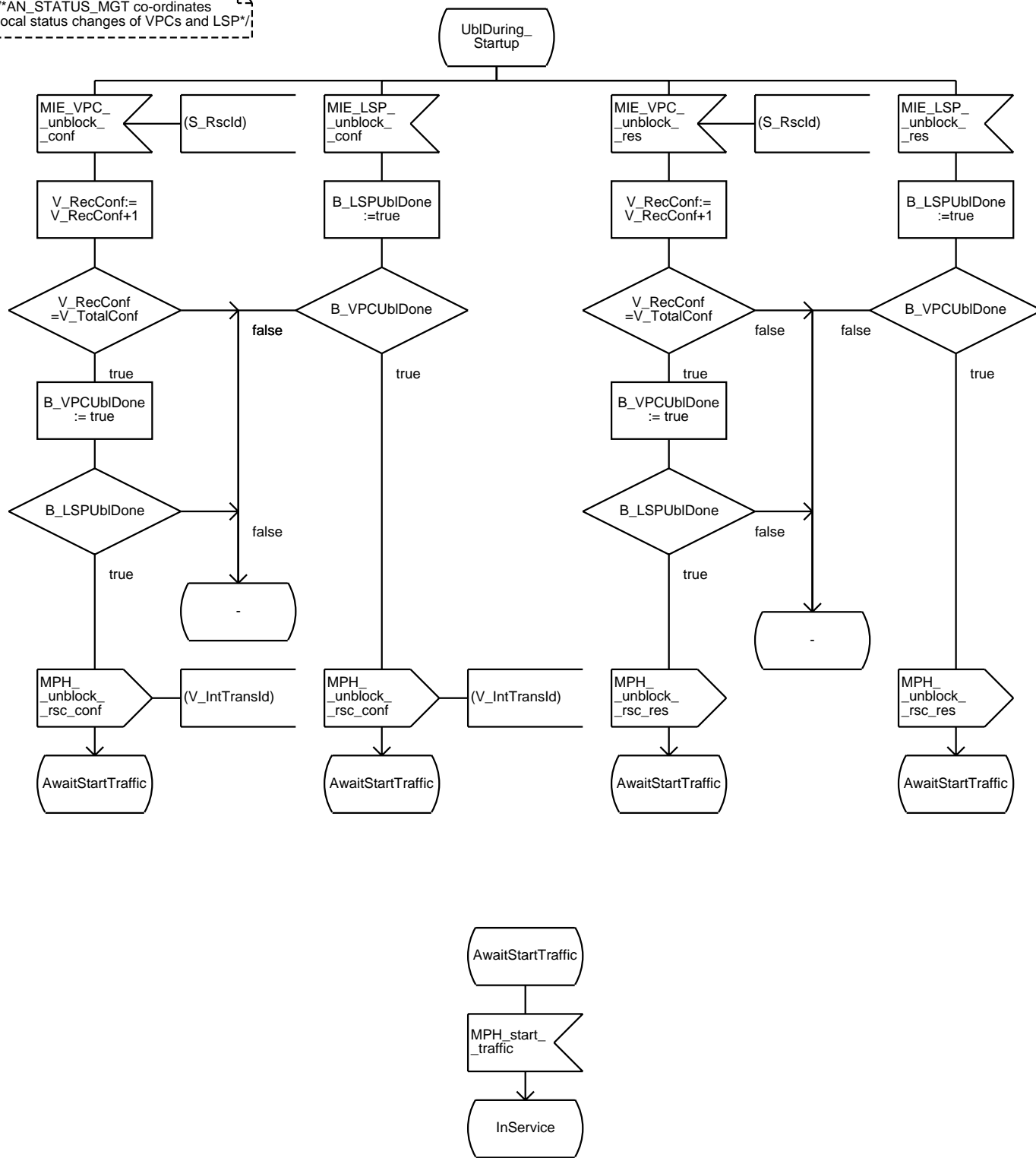
/*AN_STATUS_MGT co-ordinates
local status changes of VPCs and LSP*/



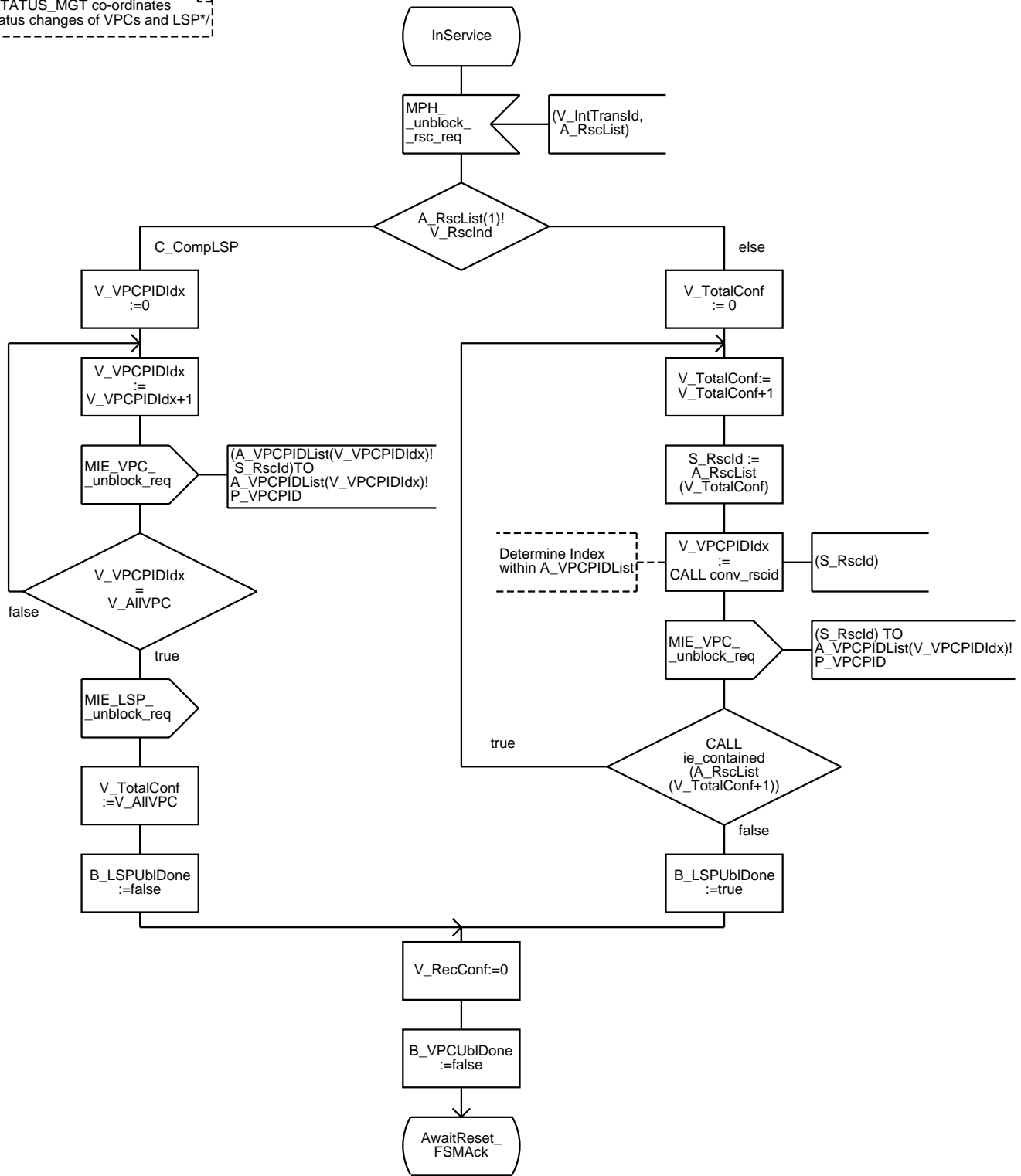
/*AN_STATUS_MGT co-ordinates
local status changes of VPCs and LSP*/



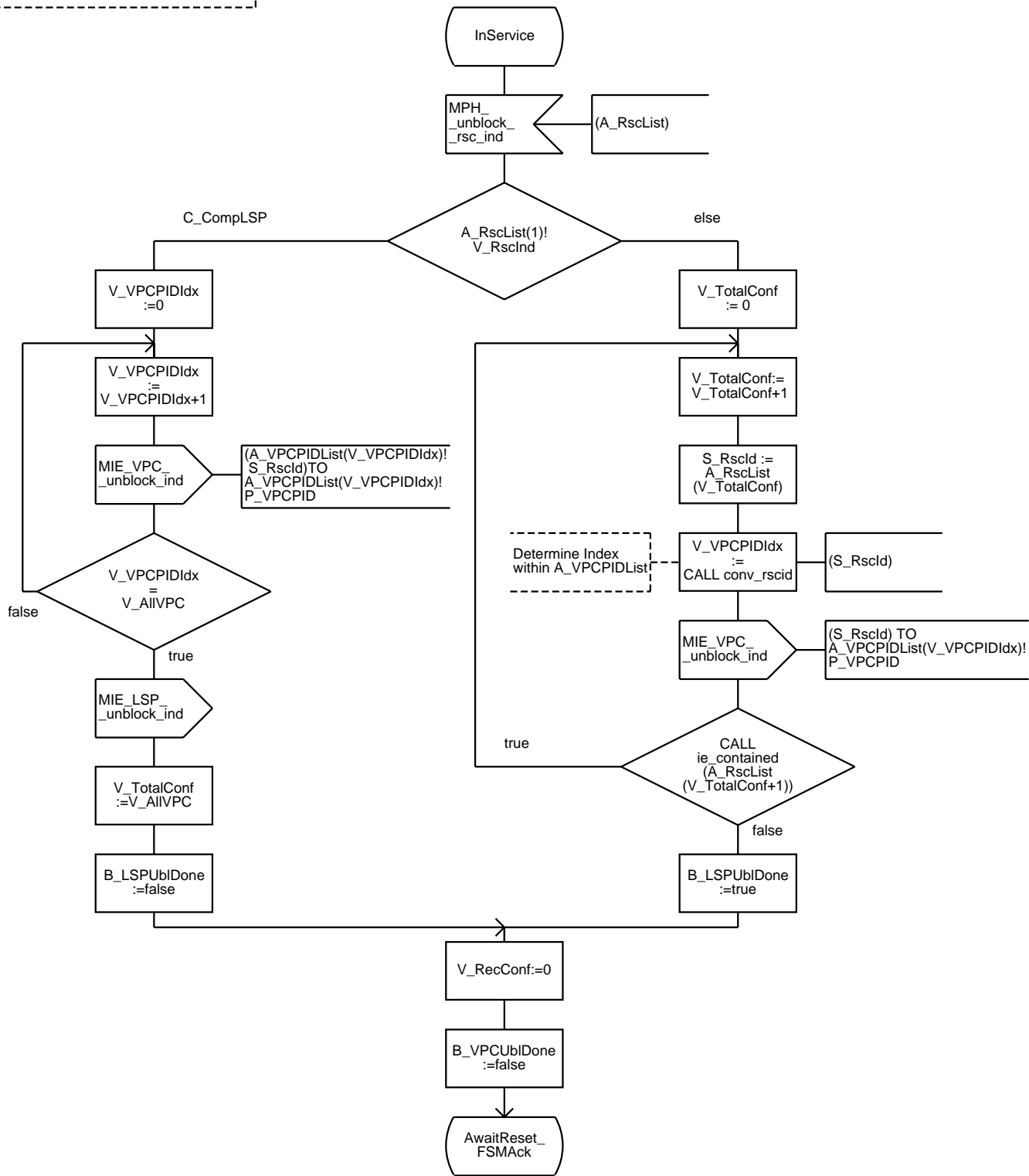
/*AN_STATUS_MGT co-ordinates
local status changes of VPCs and LSP*/



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



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

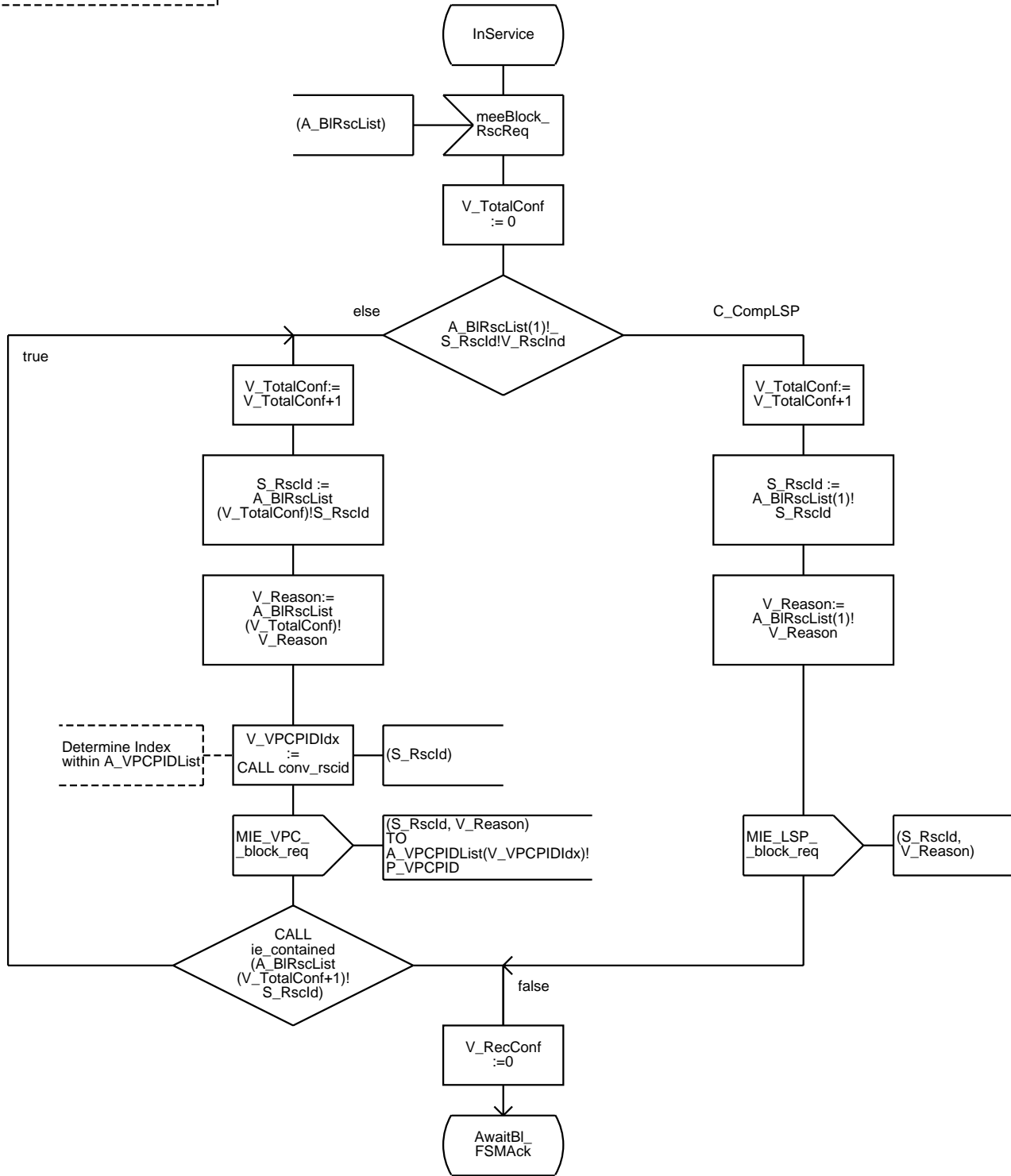


```

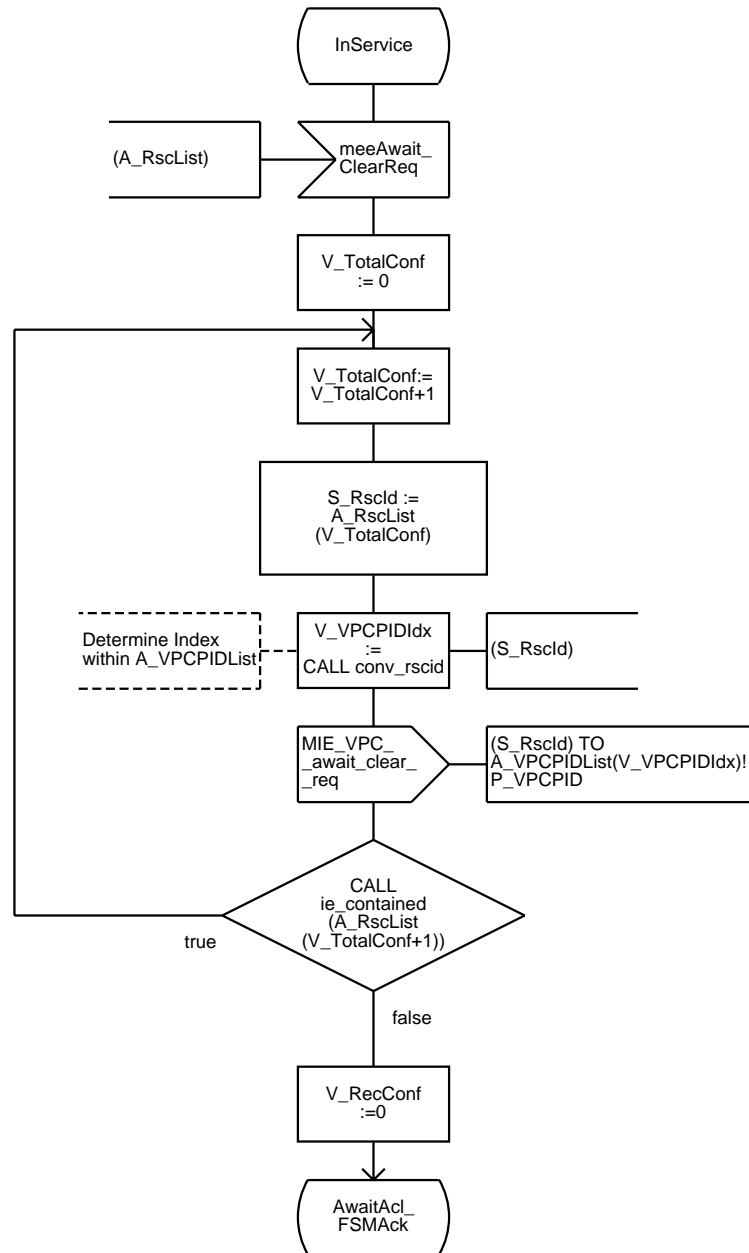
graph TD
    InService([InService]) --> UnblockReq{mee_Unblock_RscReq}
    A_RscList[A_RscList] --> UnblockReq
    UnblockReq --> V_TotalConf0[V_TotalConf := 0]
    V_TotalConf0 --> Decision1{A_RscList(1)!-V_RscInd}
    Decision1 -- "C_CompLSP" --> V_TotalConf1[V_TotalConf := V_TotalConf+1]
    Decision1 -- "else" --> V_TotalConf1
    V_TotalConf1 --> S_RscId[S_RscId := A_RscList(V_TotalConf)]
    S_RscId --> V_VPCPIDIdx[V_VPCPIDIdx := CALL conv_rscId]
    S_RscId --> V_VPCPIDIdx
    V_VPCPIDIdx --> MIE_VPC_UnblockReq{MIE_VPC_unblock_req}
    S_RscId --> MIE_VPC_UnblockReq
    MIE_VPC_UnblockReq --> Decision2{CALL ie_contained(A_RscList(V_TotalConf+1))}
    Decision2 -- "true" --> Decision1
    Decision2 -- "false" --> V_RecConf0[V_RecConf := 0]
    V_RecConf0 --> AwaitUblFSMAck([AwaitUbl_FSMAck])
    S_RscId --> MIE_LSP_UnblockReq{MIE_LSP_unblock_req}
    MIE_LSP_UnblockReq --> Decision1

```

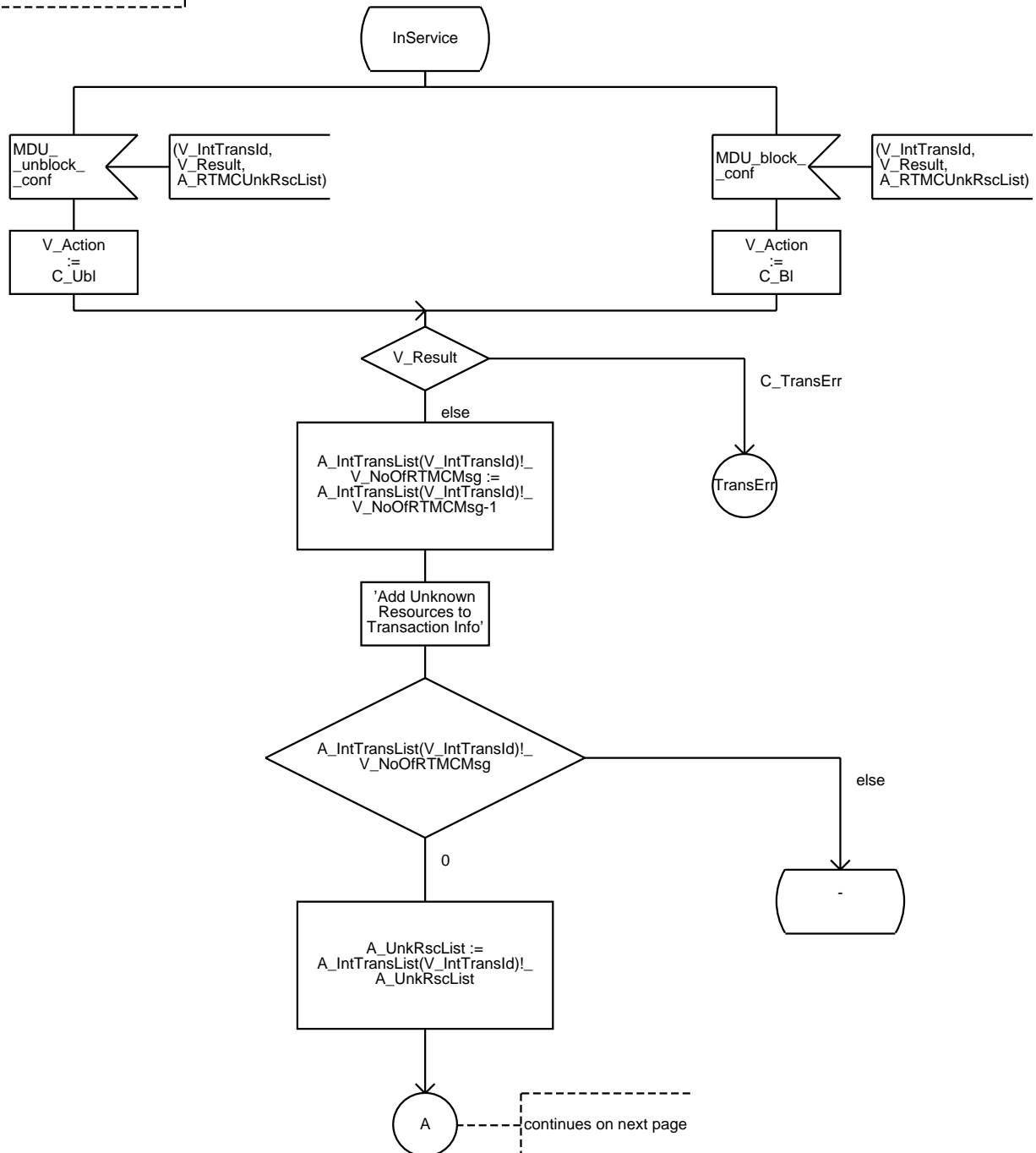
/*AN_STATUS_MGT co-ordinates
local status changes of VPCs and LSP*/



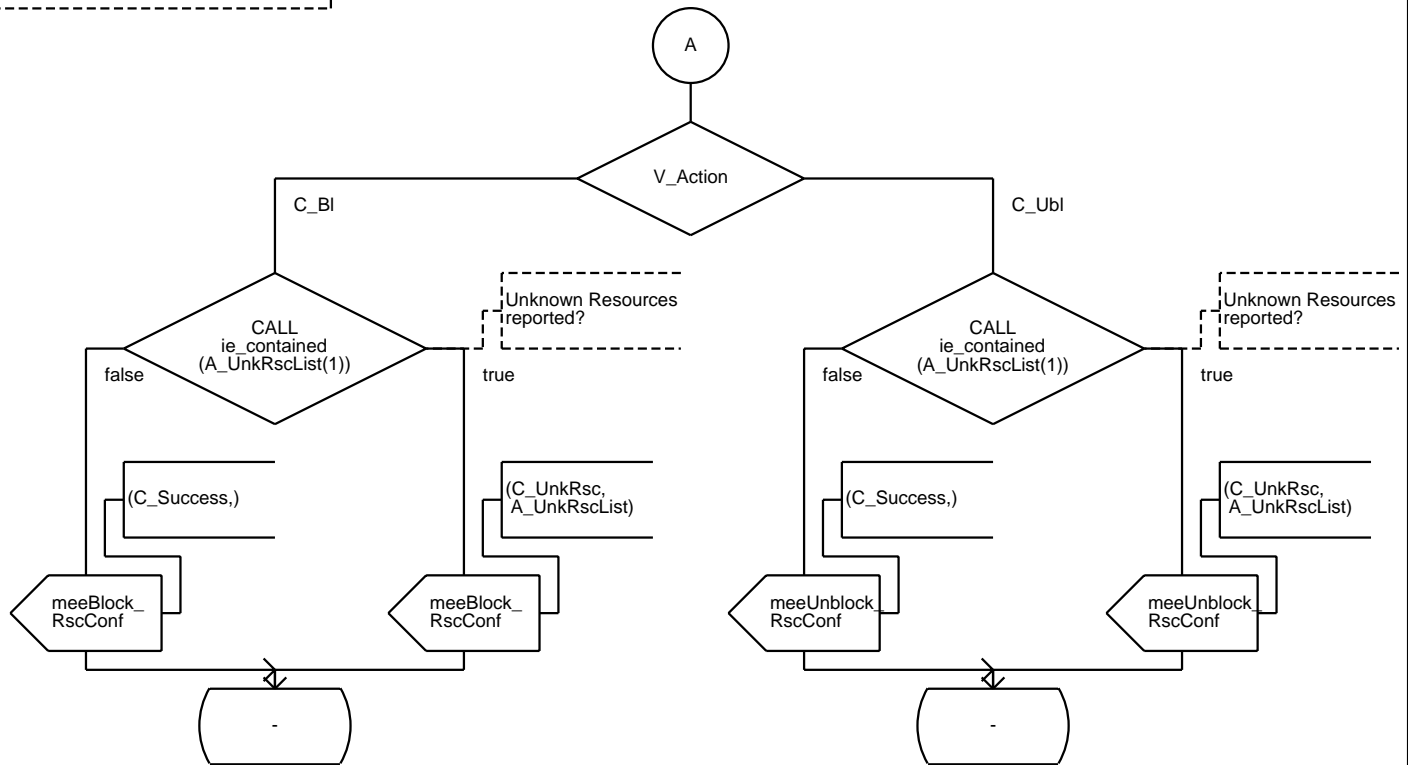
/*AN_STATUS_MGT co-ordinates
local status changes of VPCs and LSP*/



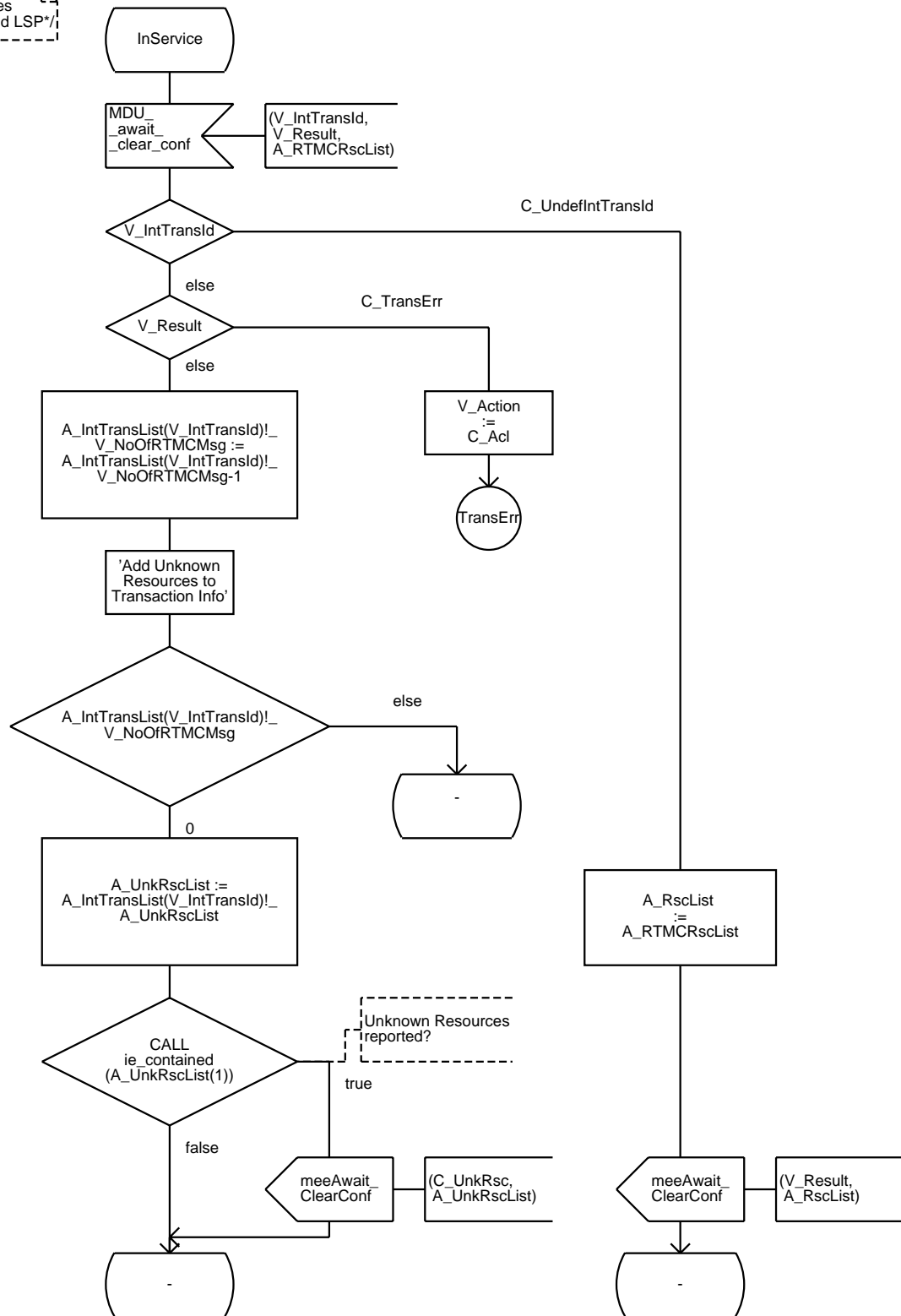
/*AN_STATUS_MGT co-ordinates
local status changes of VPCs and LSP*/



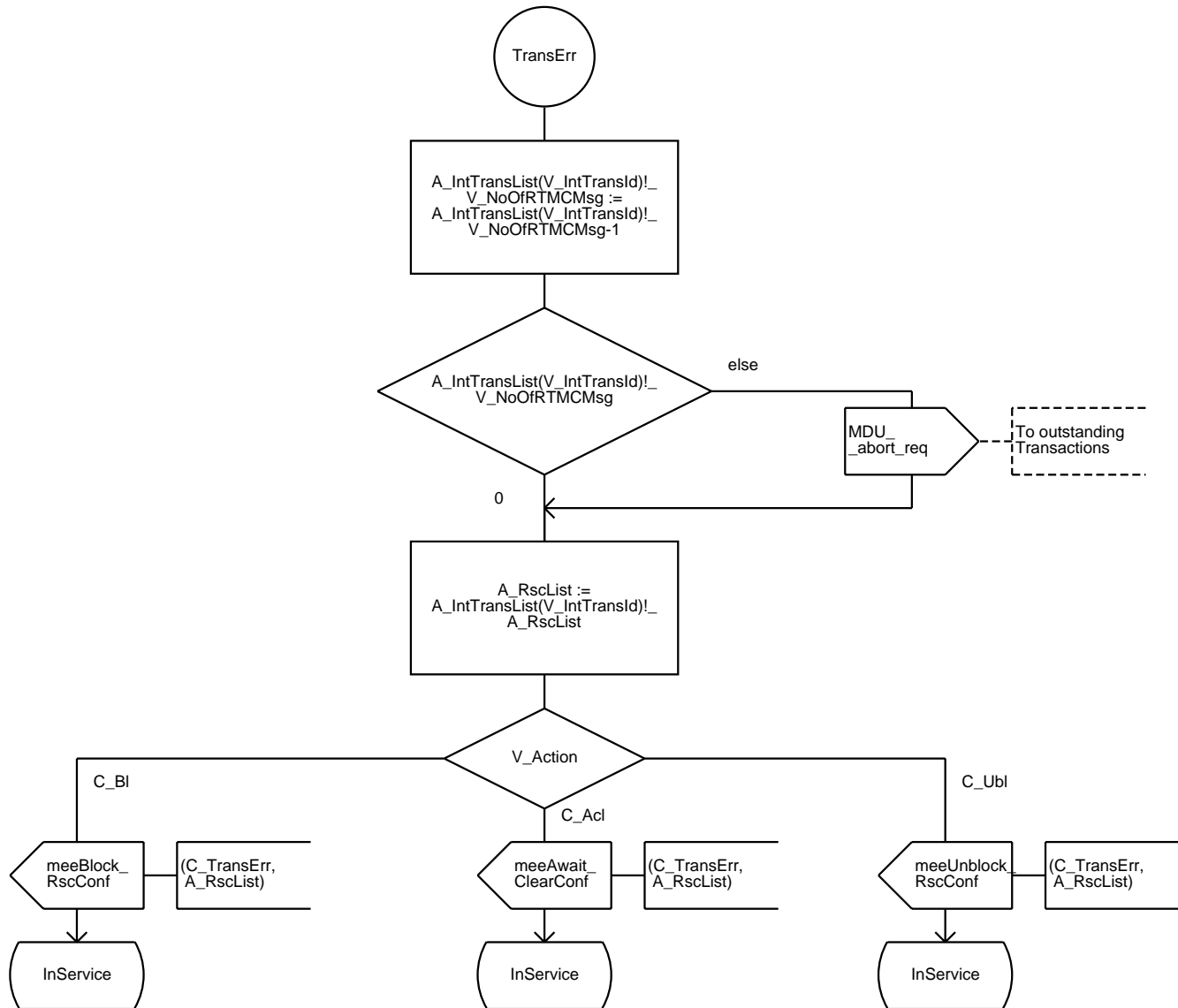
/*AN_STATUS_MGT co-ordinates
local status changes of VPCs and LSP*/



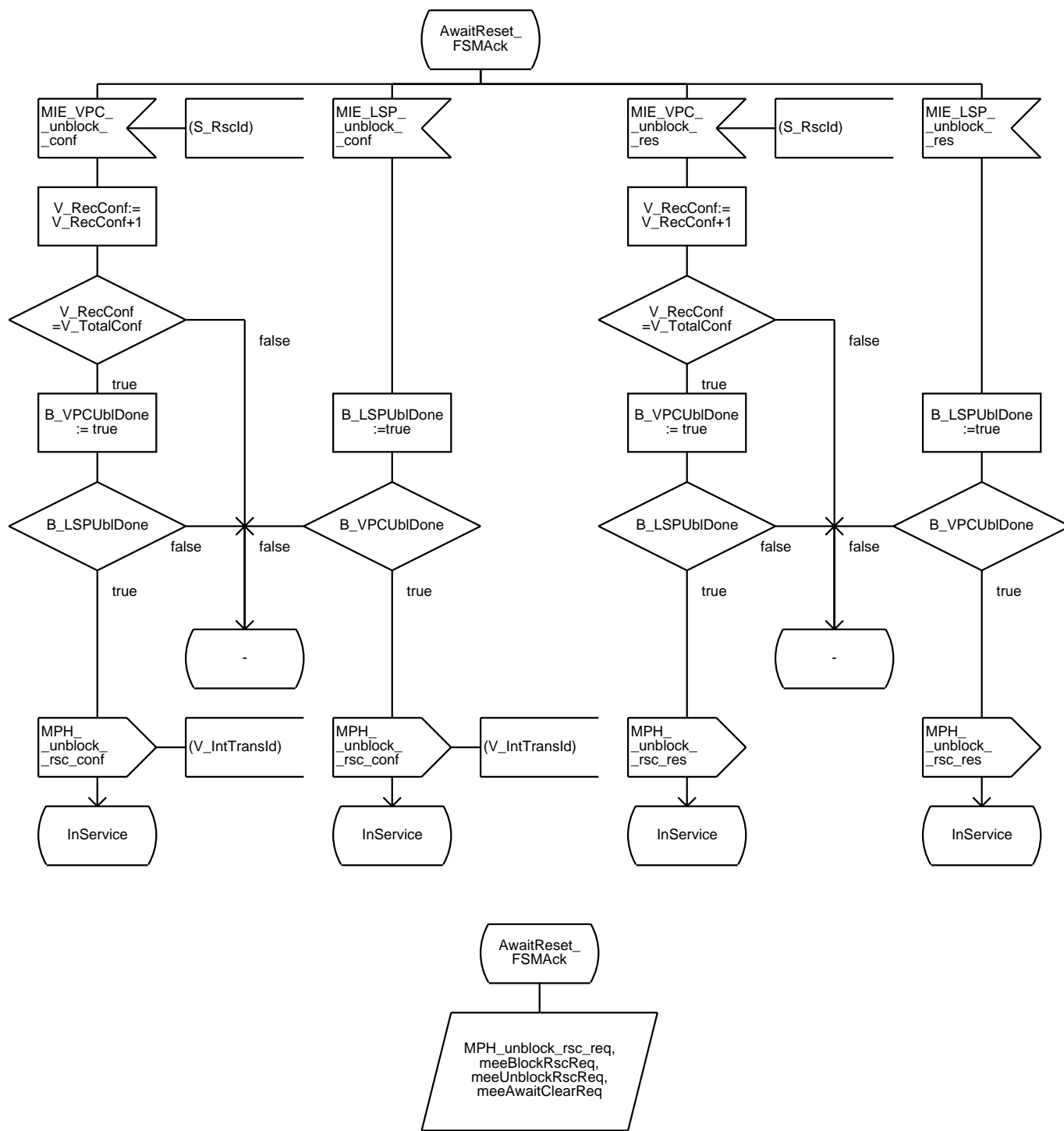
/*AN_STATUS_MGT co-ordinates
local status changes of VPCs and LSP*/



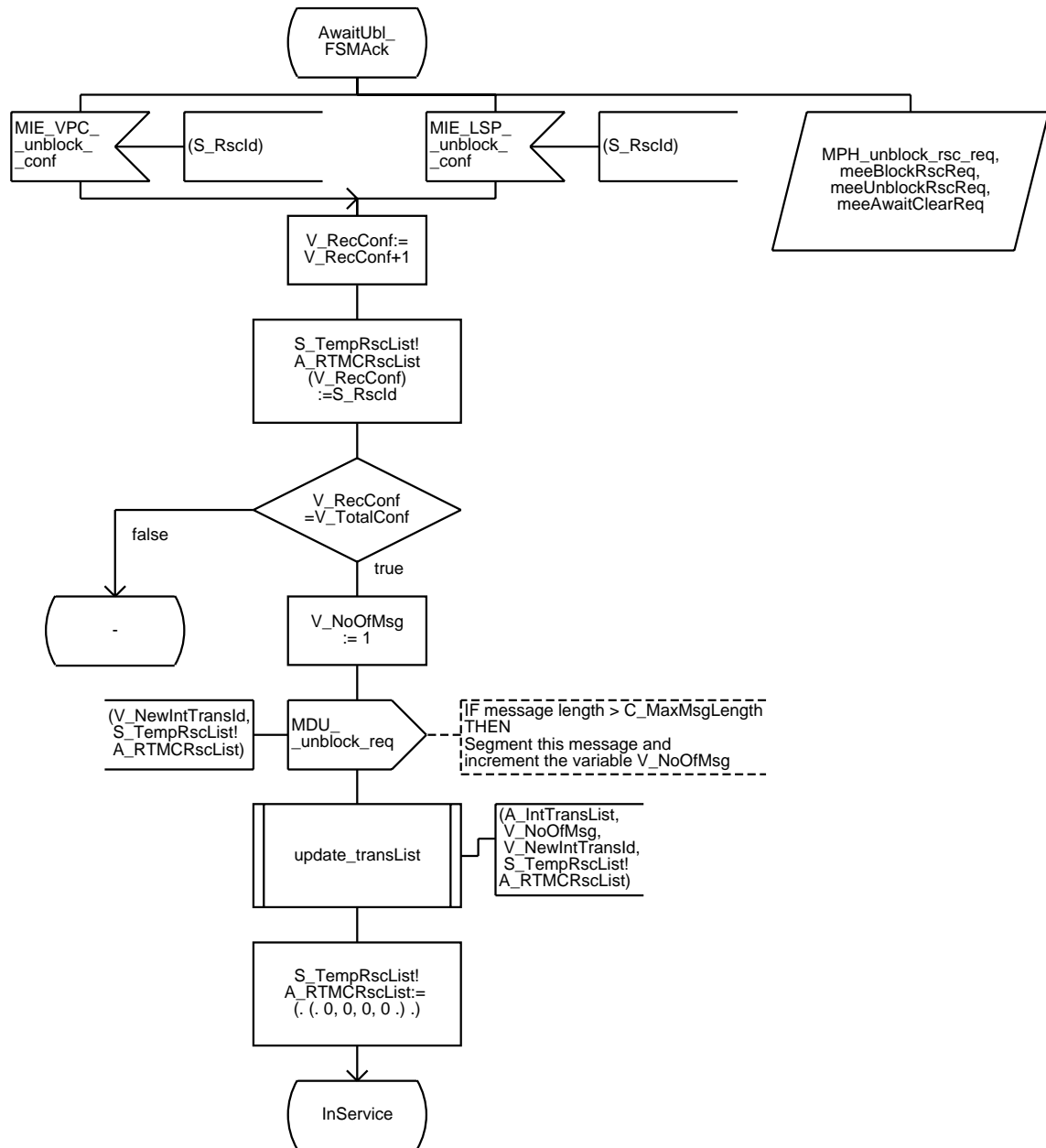
/*AN_STATUS_MGT co-ordinates
local status changes of VPCs and LSP*/



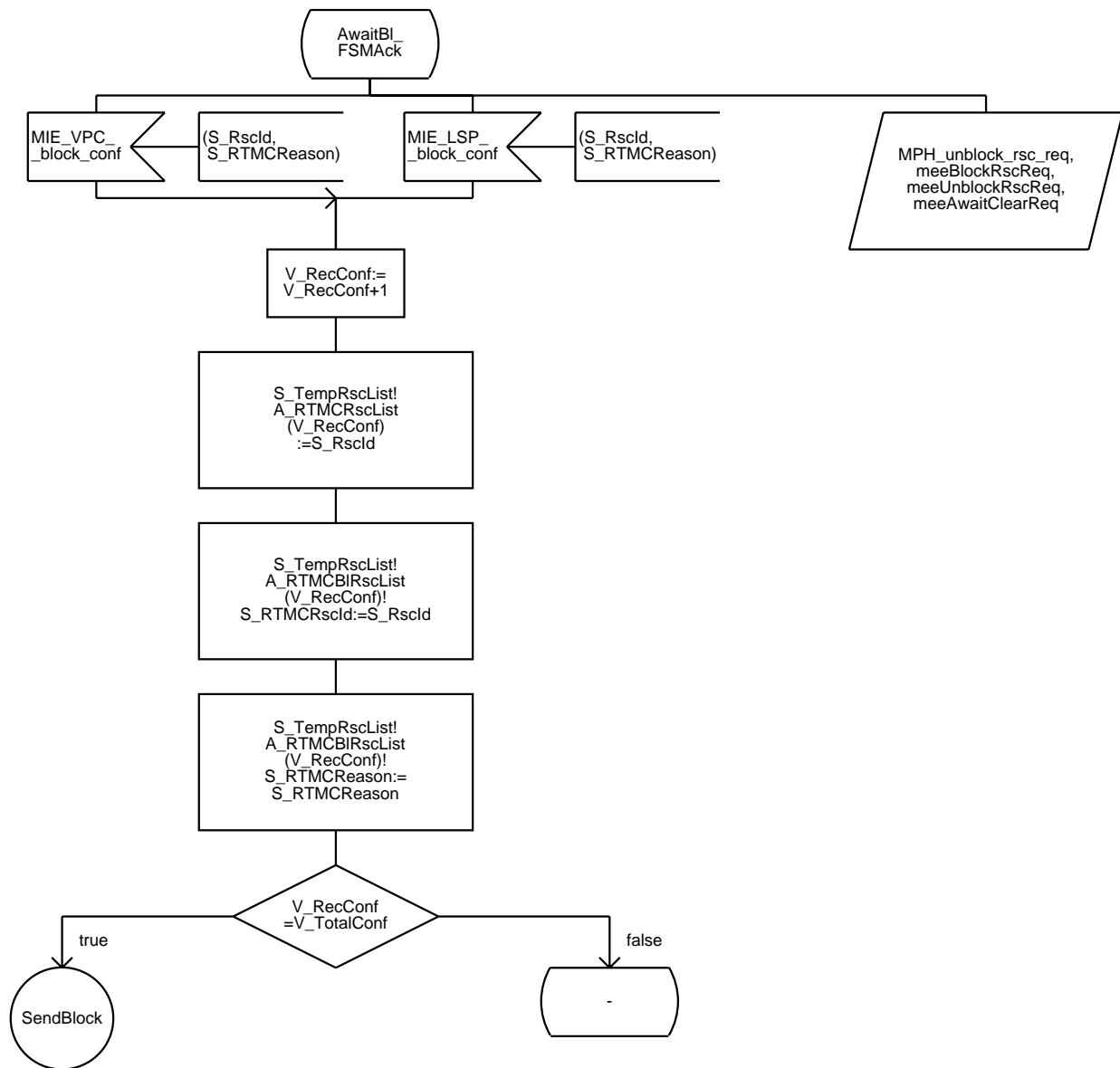
/*AN_STATUS_MGT co-ordinates
local status changes of VPCs and LSP*/



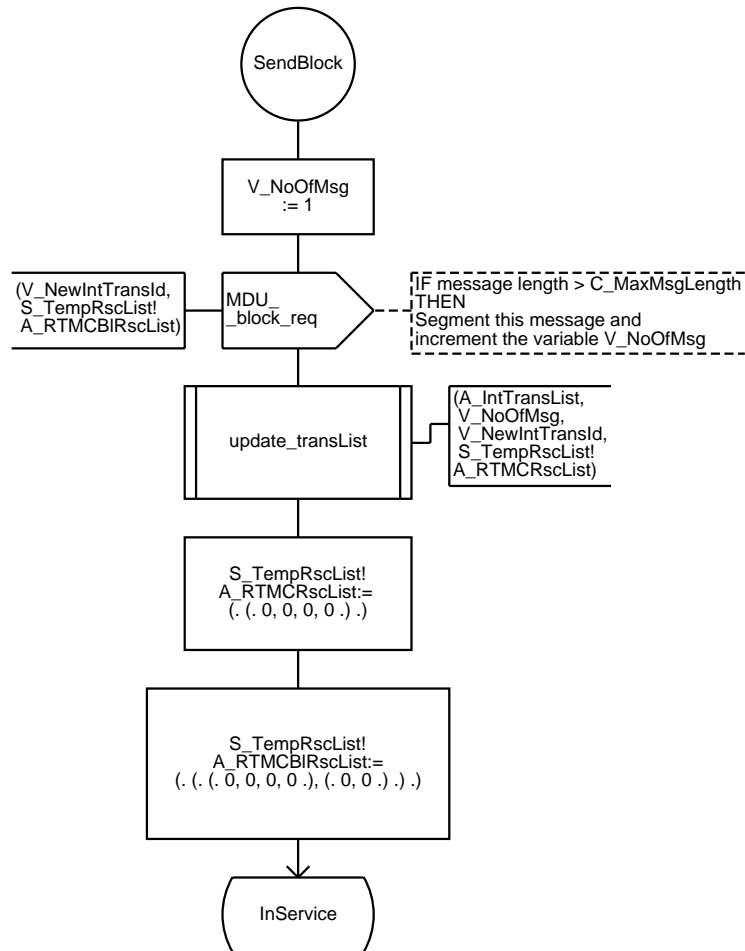
/*AN_STATUS_MGT co-ordinates
local status changes of VPCs and LSP*/



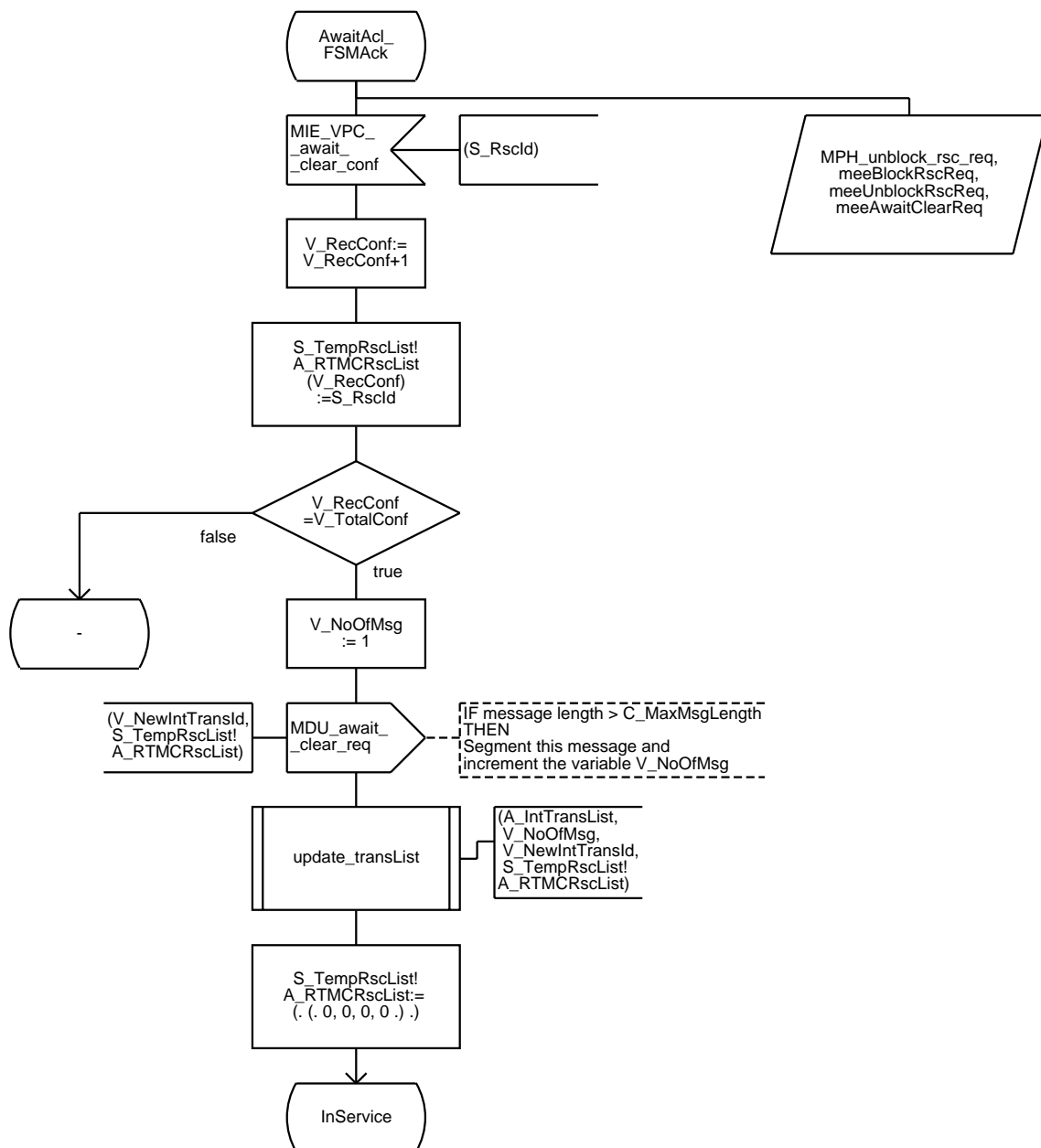
/*AN_STATUS_MGT co-ordinates
local status changes of VPCs and LSP*/



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



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



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

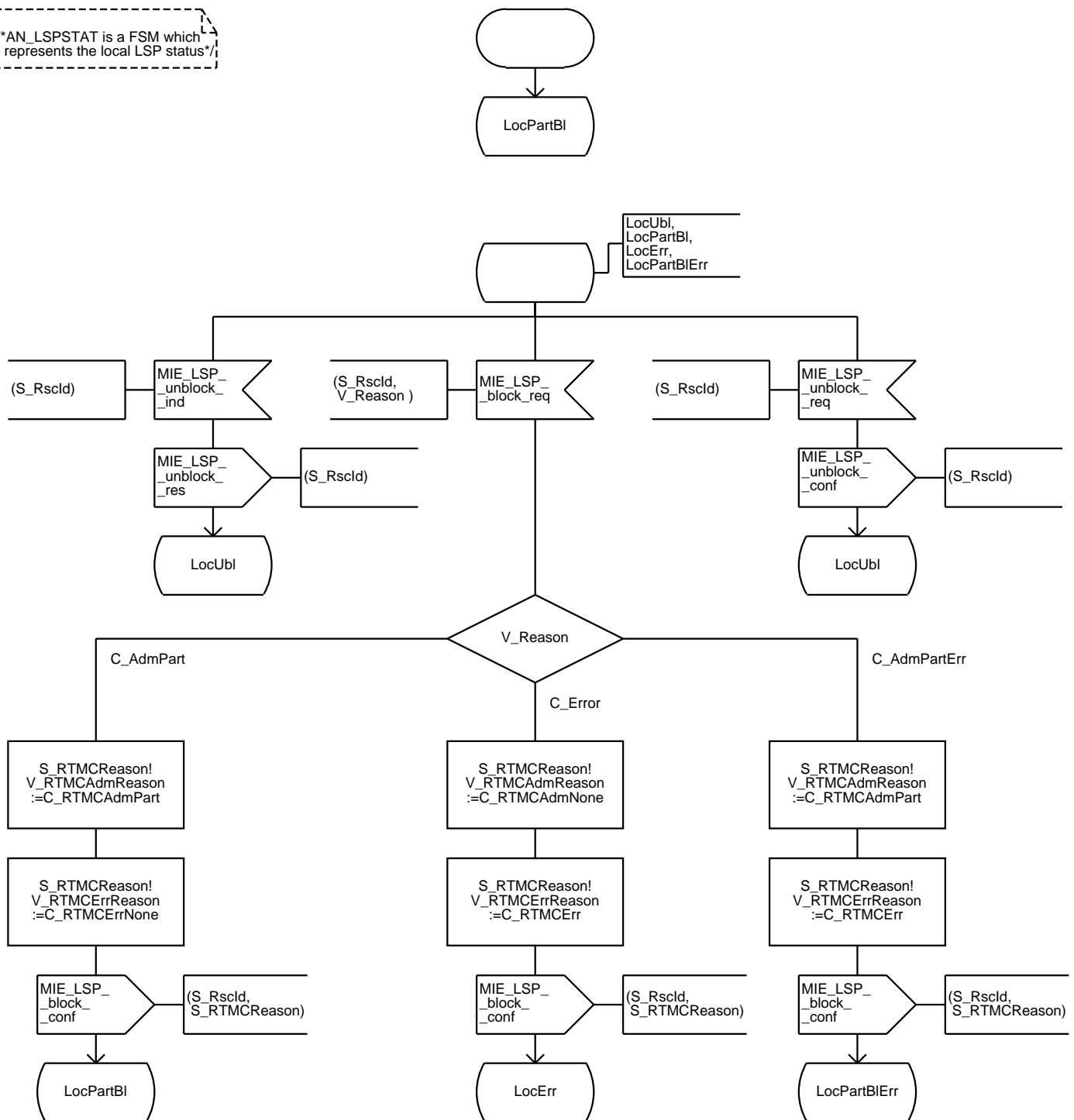
/* State descriptions

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*/

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



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

/*Definitions and declarations for AN_VPCSTAT*/

/* State descriptions

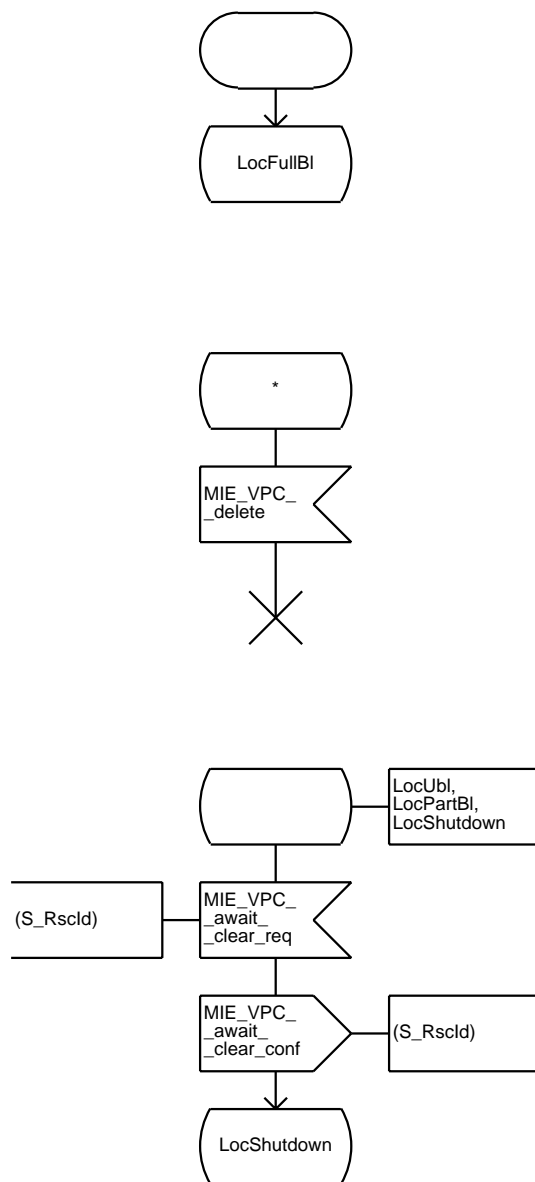
LocUbl: VPC available for Service
 LocErr: VPC unavailable due to fault conditions
 LocPartBl: VPC partially unavailable due to partial locking by AN operator
 LocPartBlErr: VPC unavailable due to partial locking by AN operator AND fault conditions
 LocShutDown: Shutting Down of VPC was requested by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

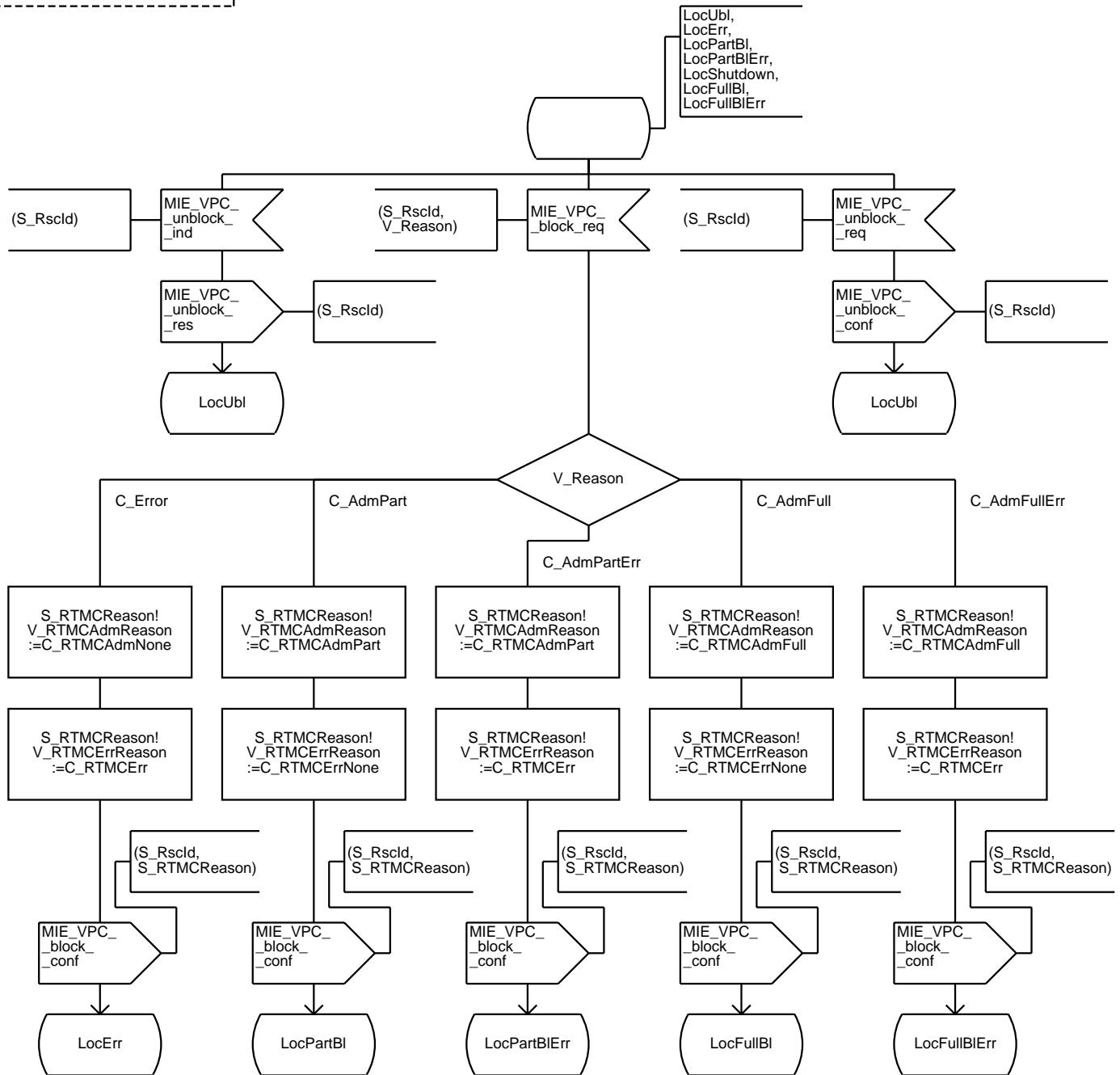
DCL
 V_Reason IT_Reason; /*Reason for status change given by the environment*/
 /**/
 DCL
 S_RTMCReason ST_RTMCReason; /*Reason for status change given to the STATUS_MGT*/
 /**/
 DCL
 S_RscId ST_RscId; /*Resource identification*/

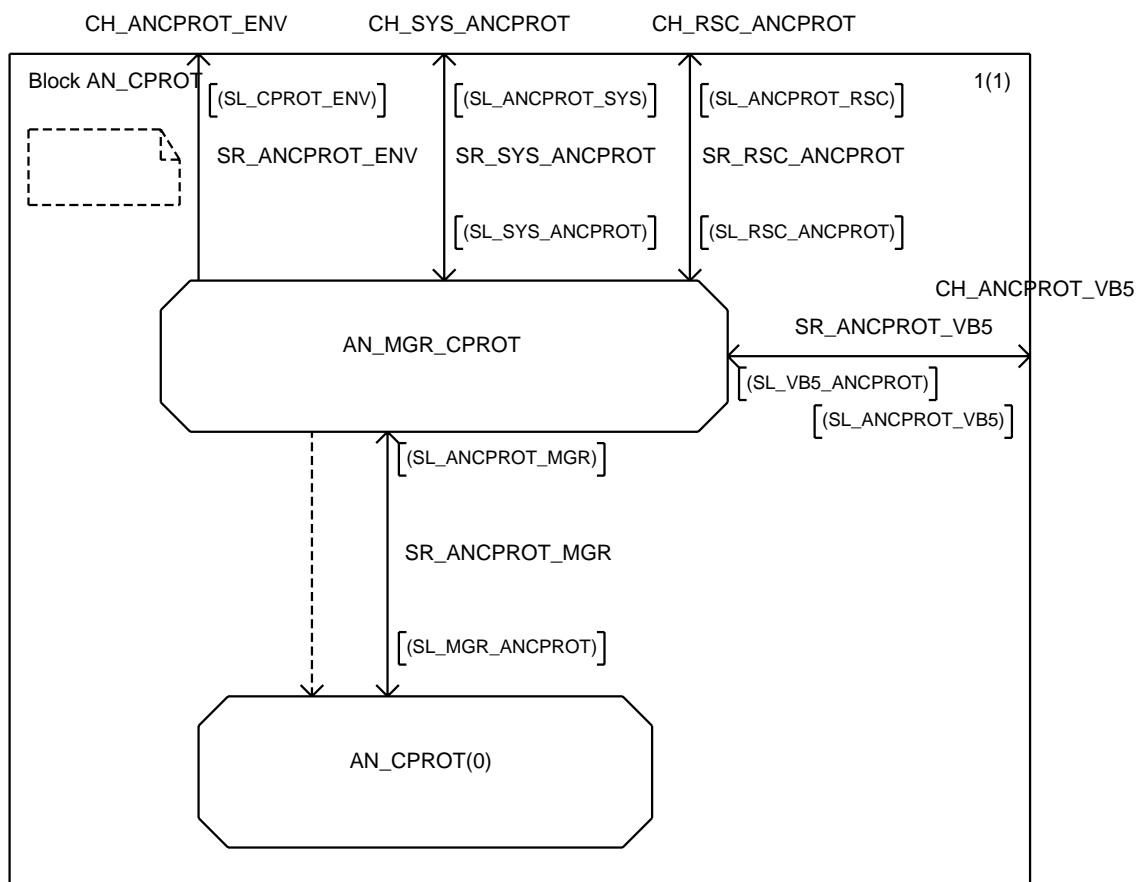
/* The state of the VC carrying the RTMC protocol is affected by the administrative state of the VPC it belongs to. I.e. it is not possible to send or receive RTMC messages if the administrative state of the VPC is set to full blocked or the VPC is disabled. This is not considered in the SDLs but has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

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



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





```

/*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;

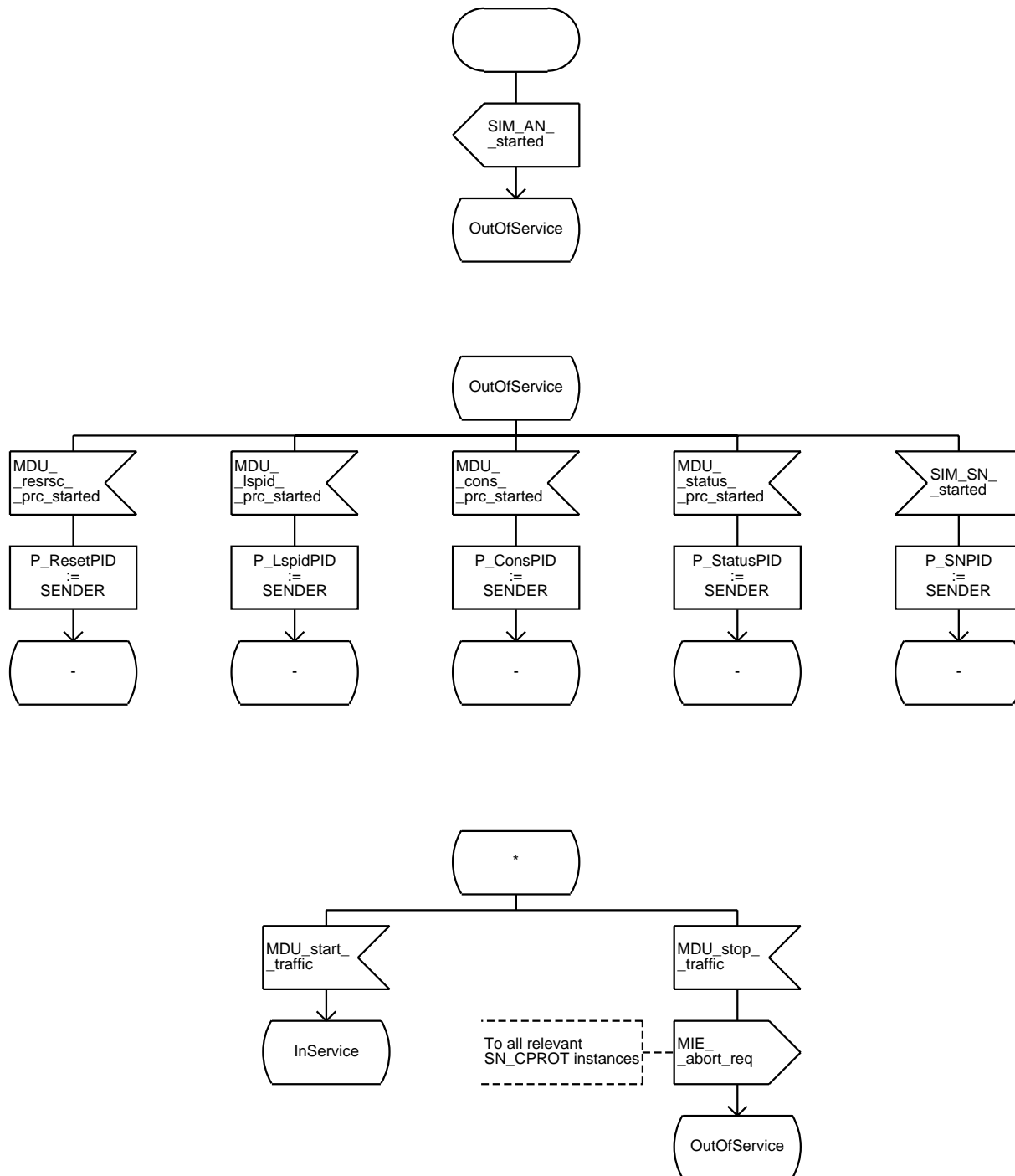
```

```

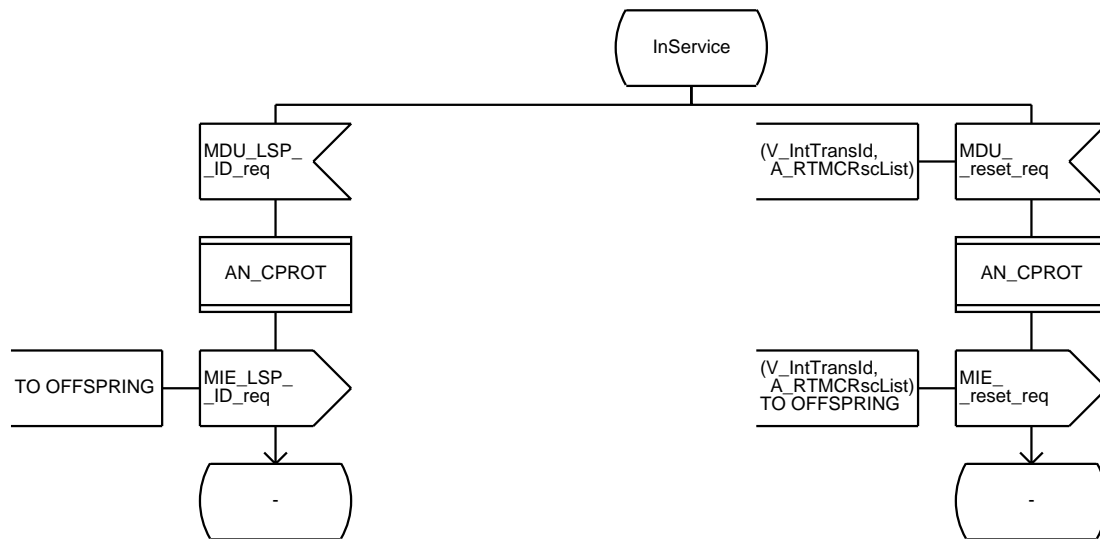
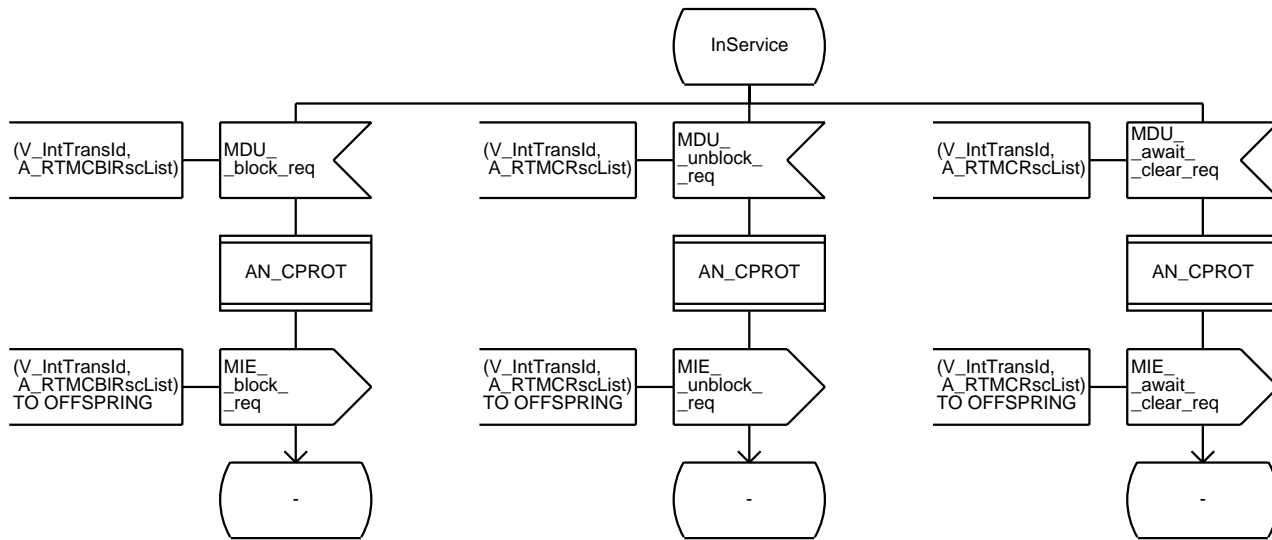
/*Primitive Data Declarations*/
DCL
A_RTMCBIRscList AT_RTMCBIRscList;
/**/
DCL
A_RTMCRscList AT_RscList;
/**/
DCL
A_RTMCUnkRscList AT_RscList;
/**/
DCL
S_RTMCRsclD ST_RsclD;
/**/
DCL
S_RTMCUnkRsclD ST_RsclD;
/**/
DCL
V_RTMCResult IT_RTMCResult;
/**/
DCL
V_Result IT_Result;
/**/
DCL
V_IntTranslD IT_IntTranslD;
/**/
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_TranslD, /*Transaction Identifier*/
P_SNPID /*PID of SN_CPROT_MGR*/
PID;

```

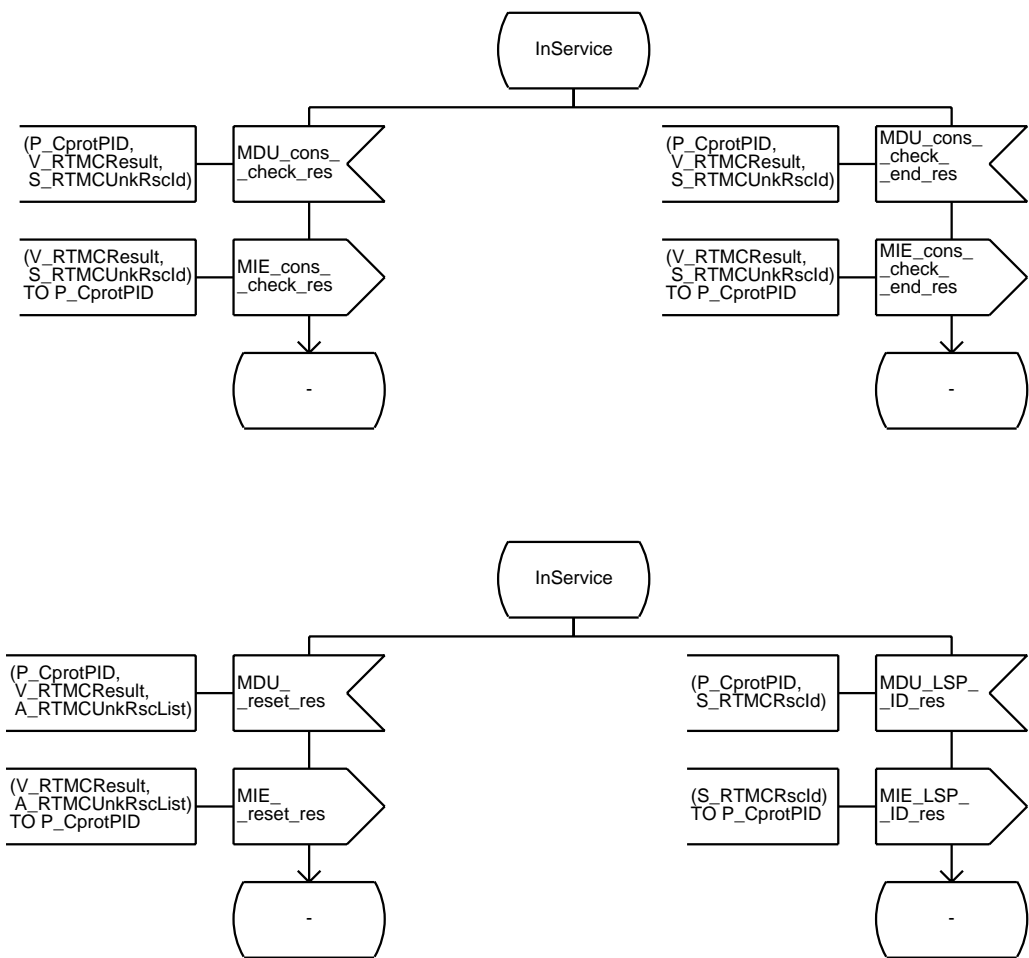
/*AN_CPROT Manager, controls
instantiation of AN_CPROT
processes*/



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



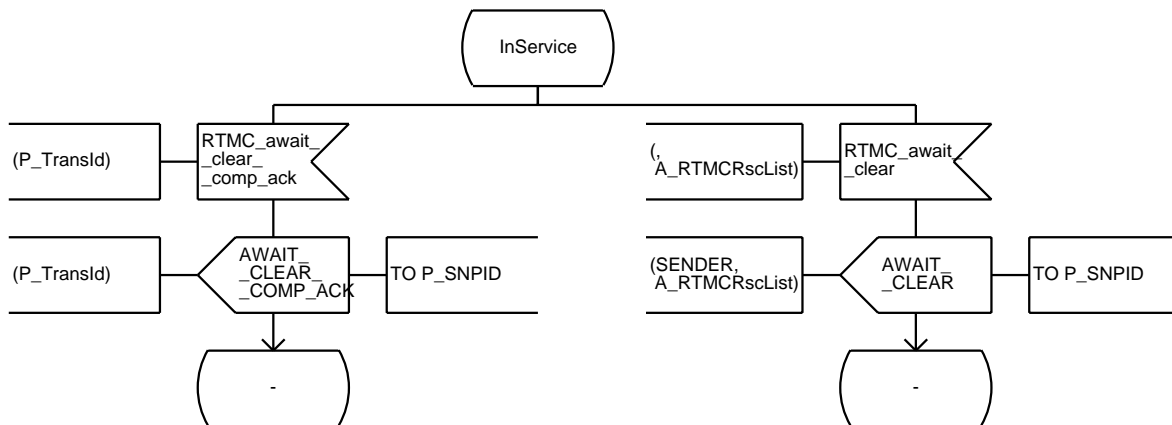
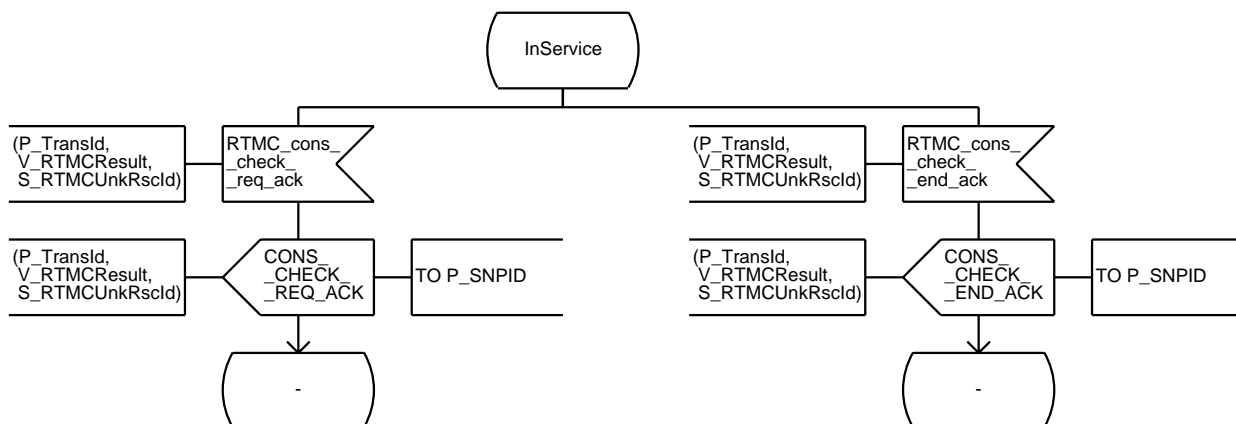
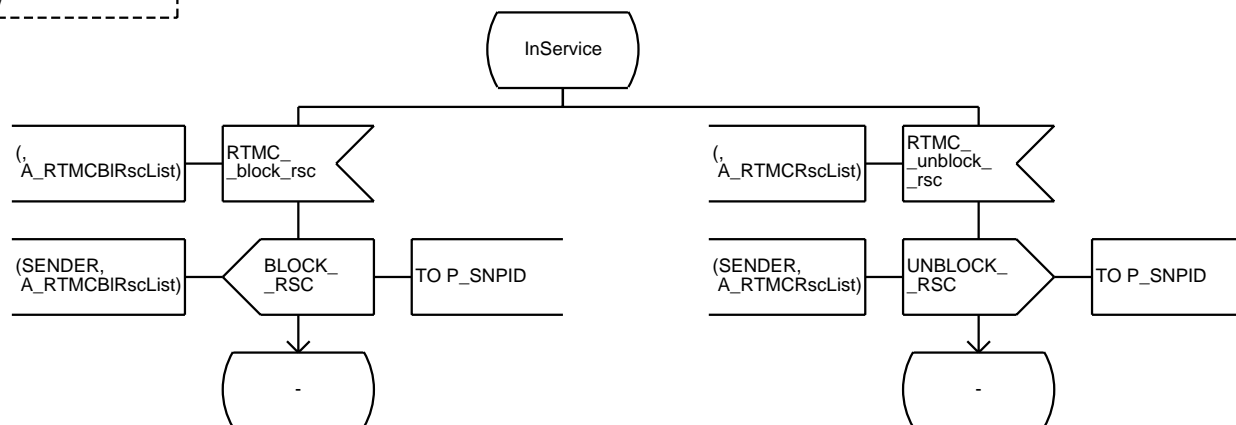
/*AN_CPROT Manager, controls
instantiation of AN_CPROT
processes*/



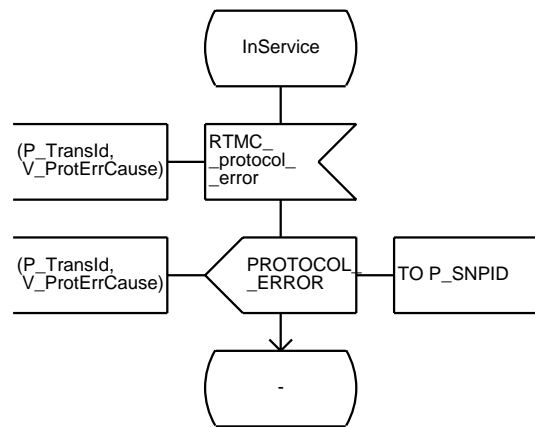
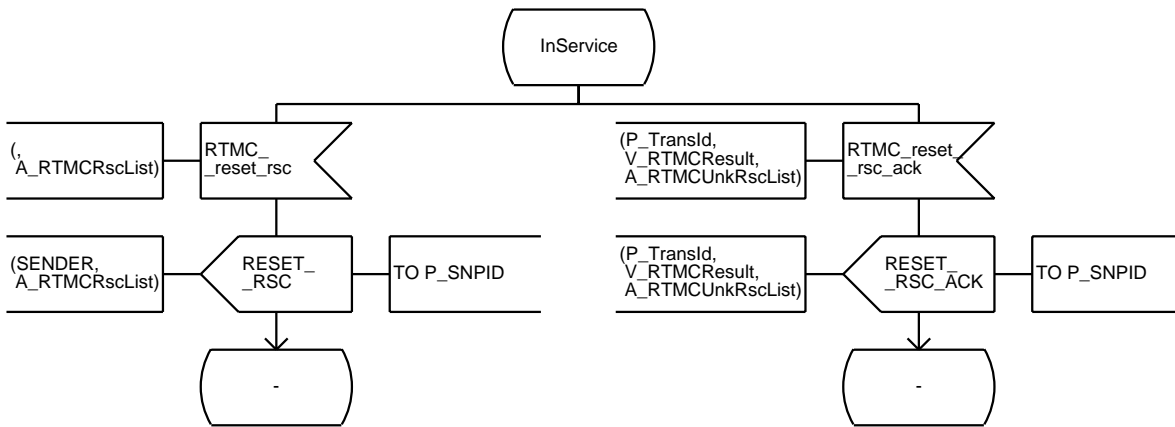
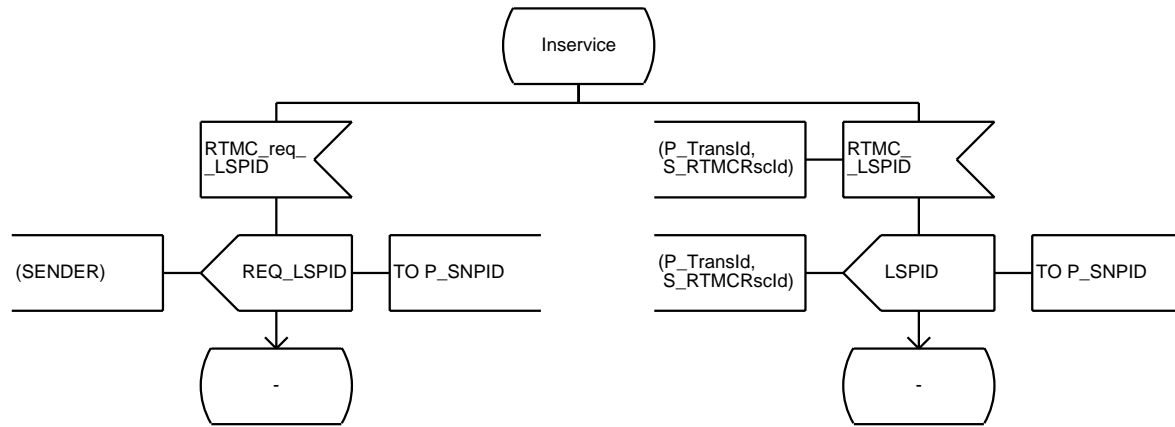
/*AN_CPROT Manager, controls
instantiation of AN_CPROT
processes*/



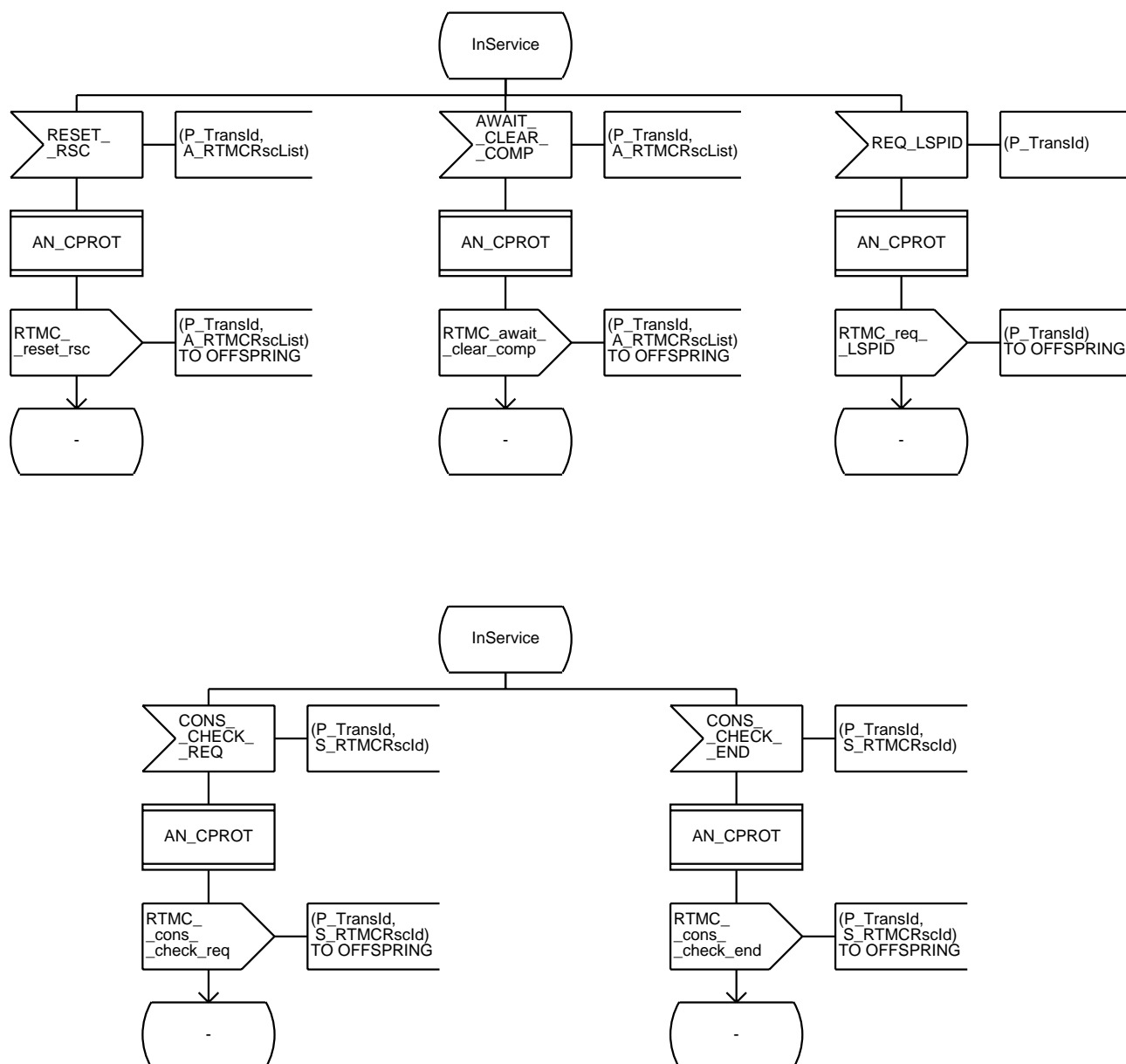
/*AN_CPROT Manager, controls
instantiation of AN_CPROT
processes*/



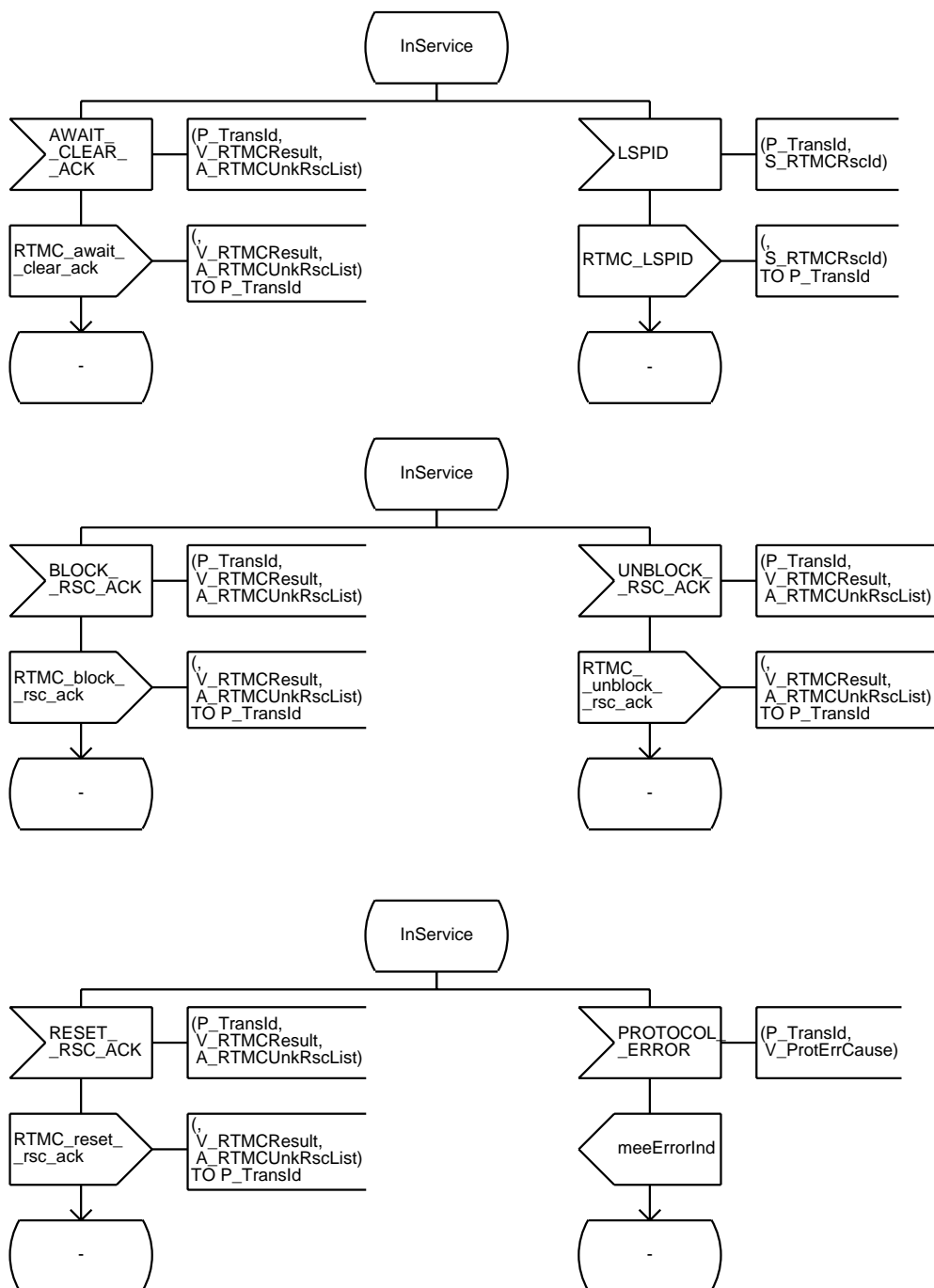
/*AN_CPROT Manager, controls
instantiation of AN_CPROT
processes*/



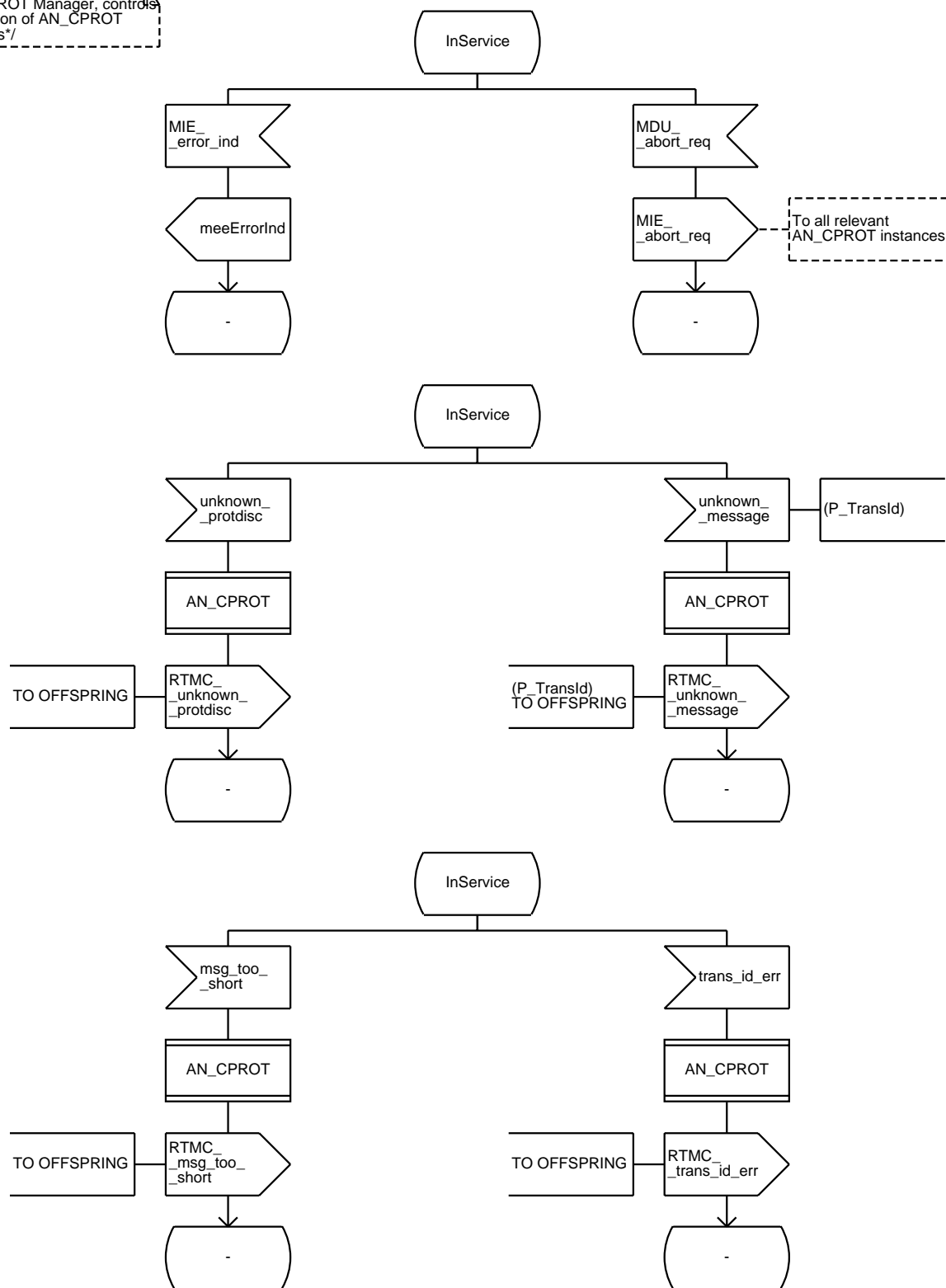
/*AN_CPROT Manager, controls
instantiation of AN_CPROT
processes*/



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



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



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

/* Definitions 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 */

/* Timer definitions */

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*/

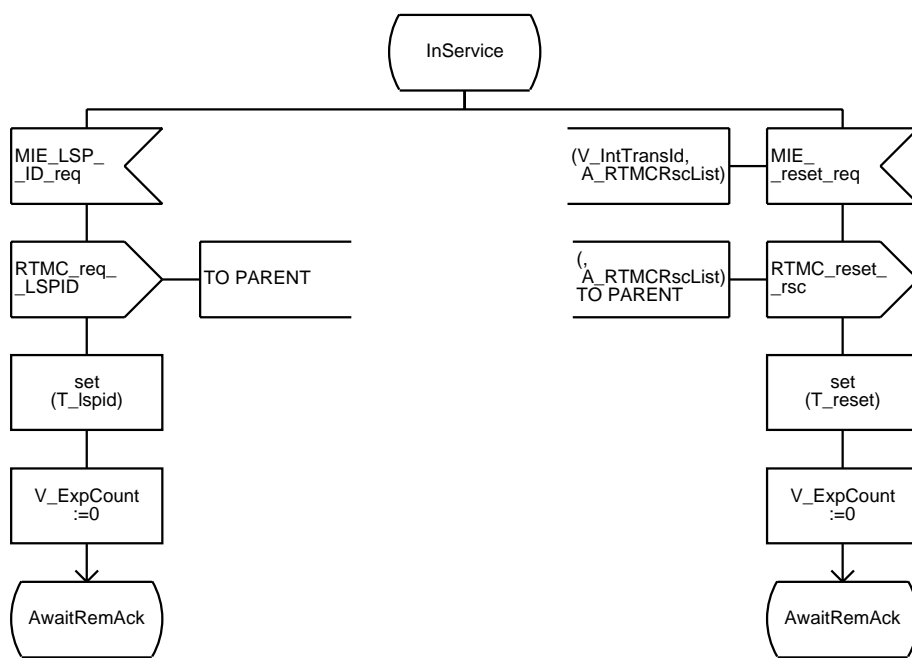
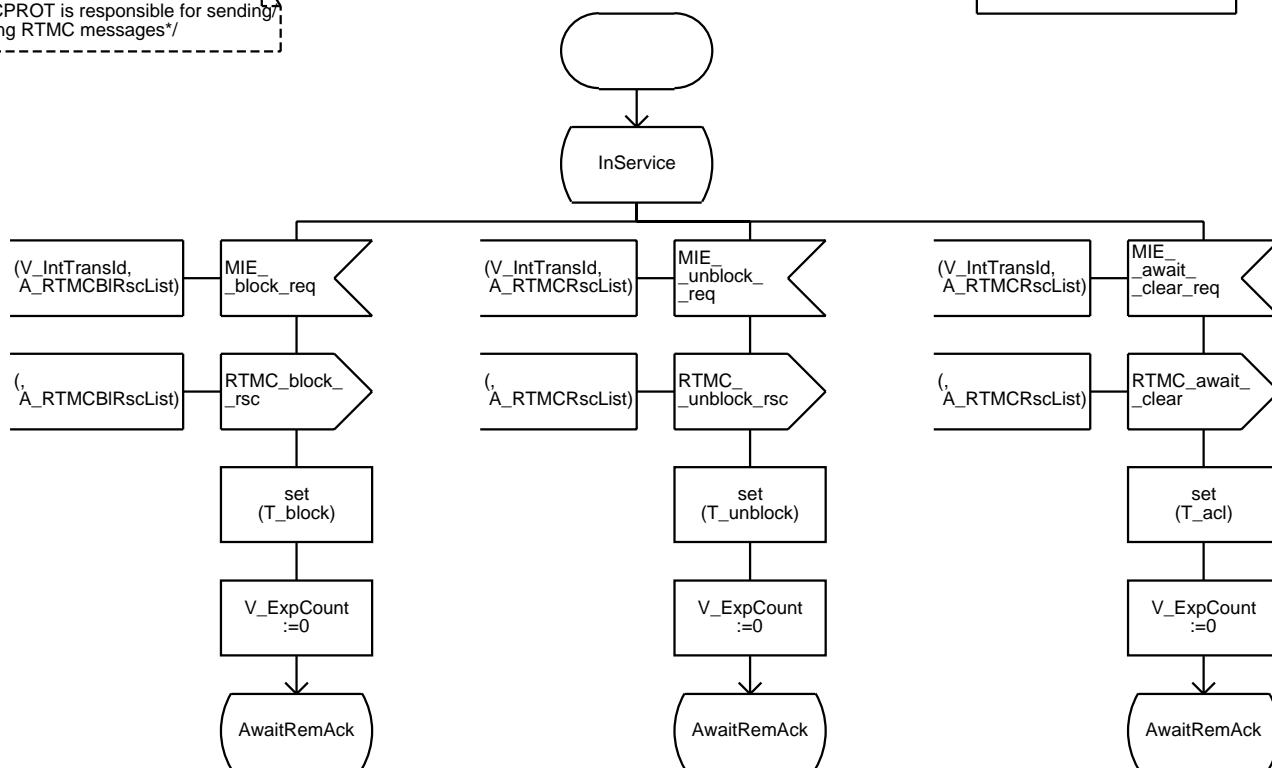
/**/
/* timer handling */
DCL V_ExpCount NATURAL; /*number of actual message repetitions*/
/*possible values*/
SYNONYM C_ExpMax INTEGER = 1; /*maximum number of message repetitions is 1*/
/**/
/*syntax check handling*/
DCL V_SynResult INTEGER; /*syntax check result, provided by procedure IE_CHECK*/
/*possible 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_RTMCResId ST_ResId; /*Resource Identifier Information Element*/
/**/
DCL
S_RTMCUnkResId ST_ResId; /*Resource Identifier Information Element unknown by the peer side*/
/**/
DCL
A_RTMCResList AT_ResList; /*Array of Resource Identifier Information Elements*/
/**/
DCL
A_RTMCBIRscList AT_RTMCBIRscList; /*Array of Blocked Resource Identifier Information Elements*/
/**/
DCL
A_RTMCUnkResList AT_ResList; /*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*/

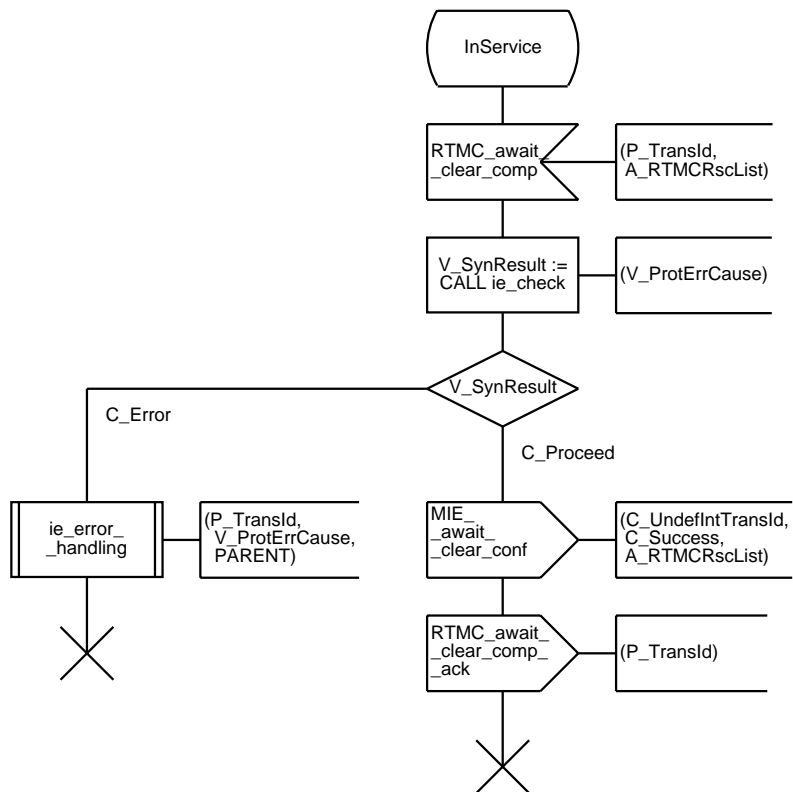
/*AN_CPROT is responsible for sending
receiving RTMC messages*/

/*Handling of Requests from
AN System Management */



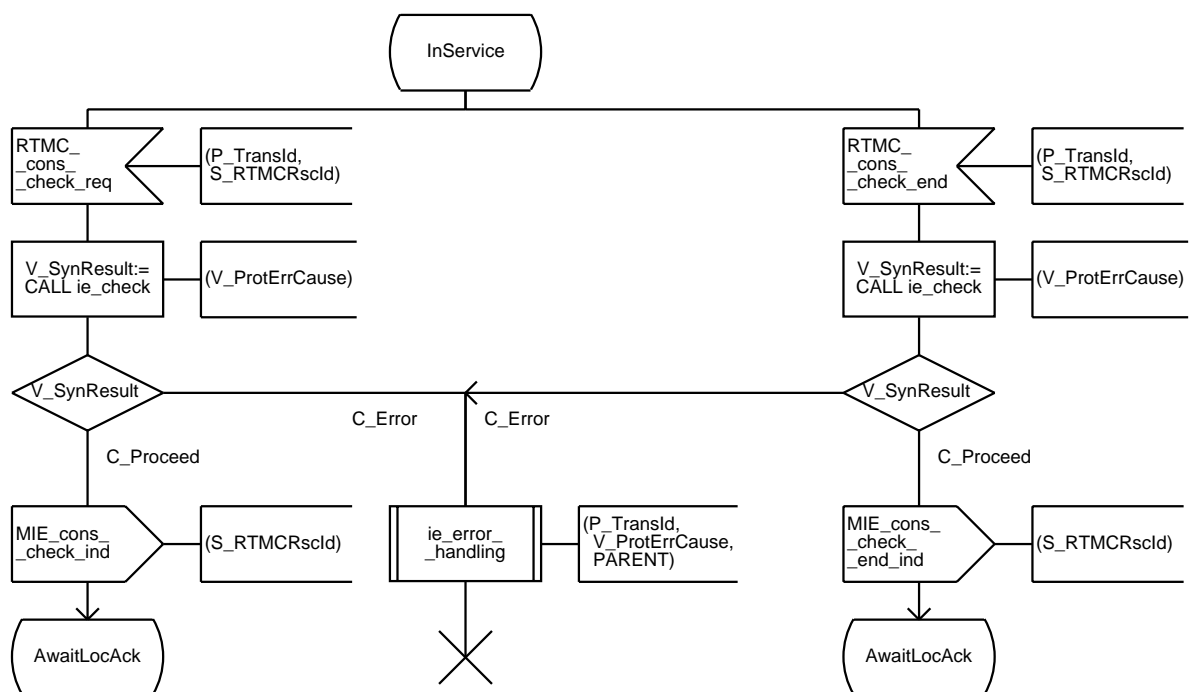
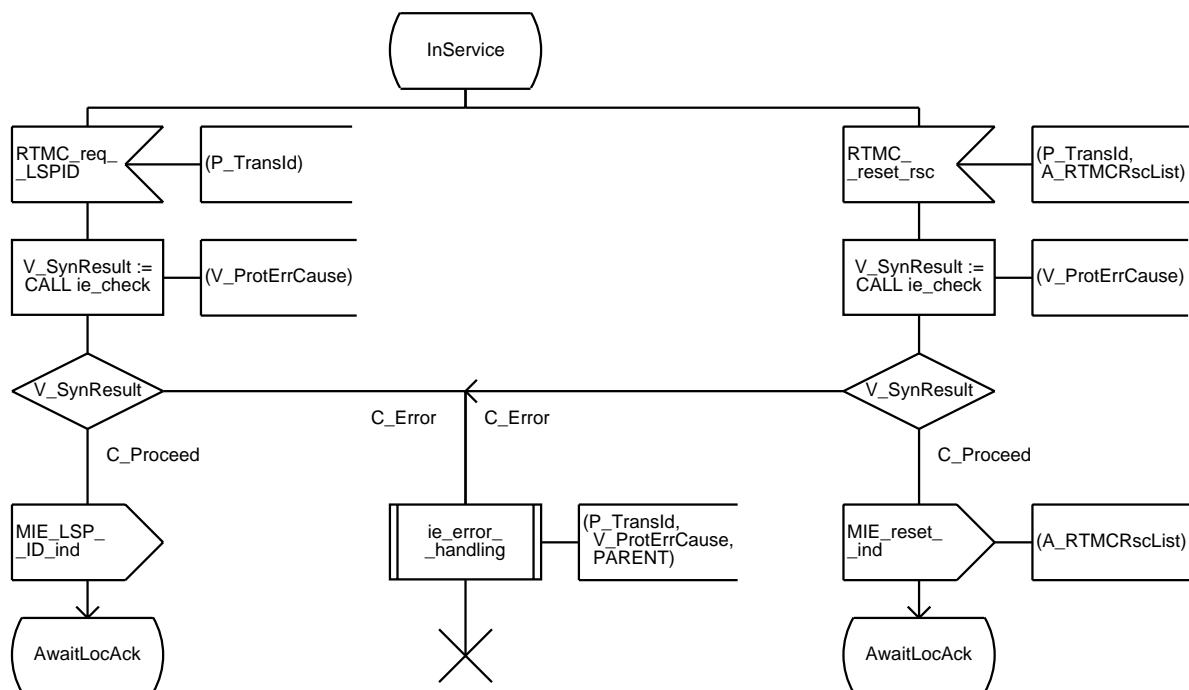
/*AN_CPROT is responsible for sending
receiving RTMC messages*/

/* Handling of
VB5 Messages from SN */



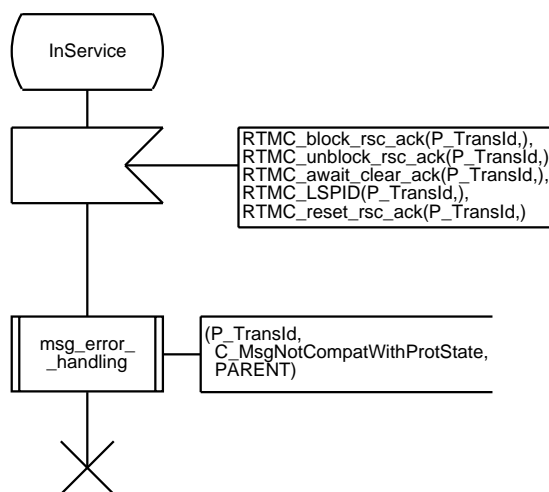
/*AN_CPROT is responsible for sending
receiving RTMC messages*/

/* Handling of
VB5 Messages from SN */



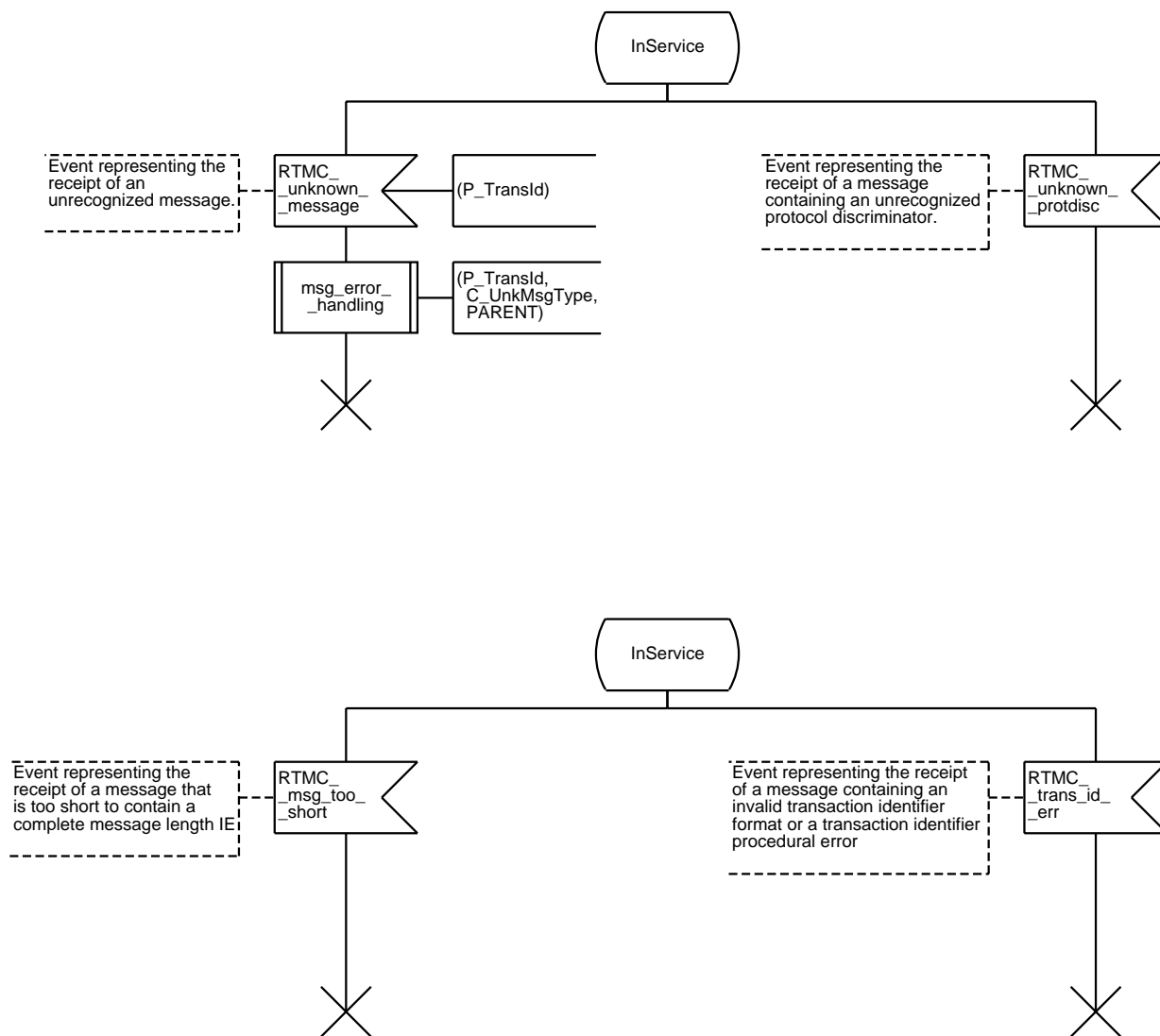
/*AN_CPROT is responsible for sending
receiving RTMC messages*/

/* Handling of
unexpected events */



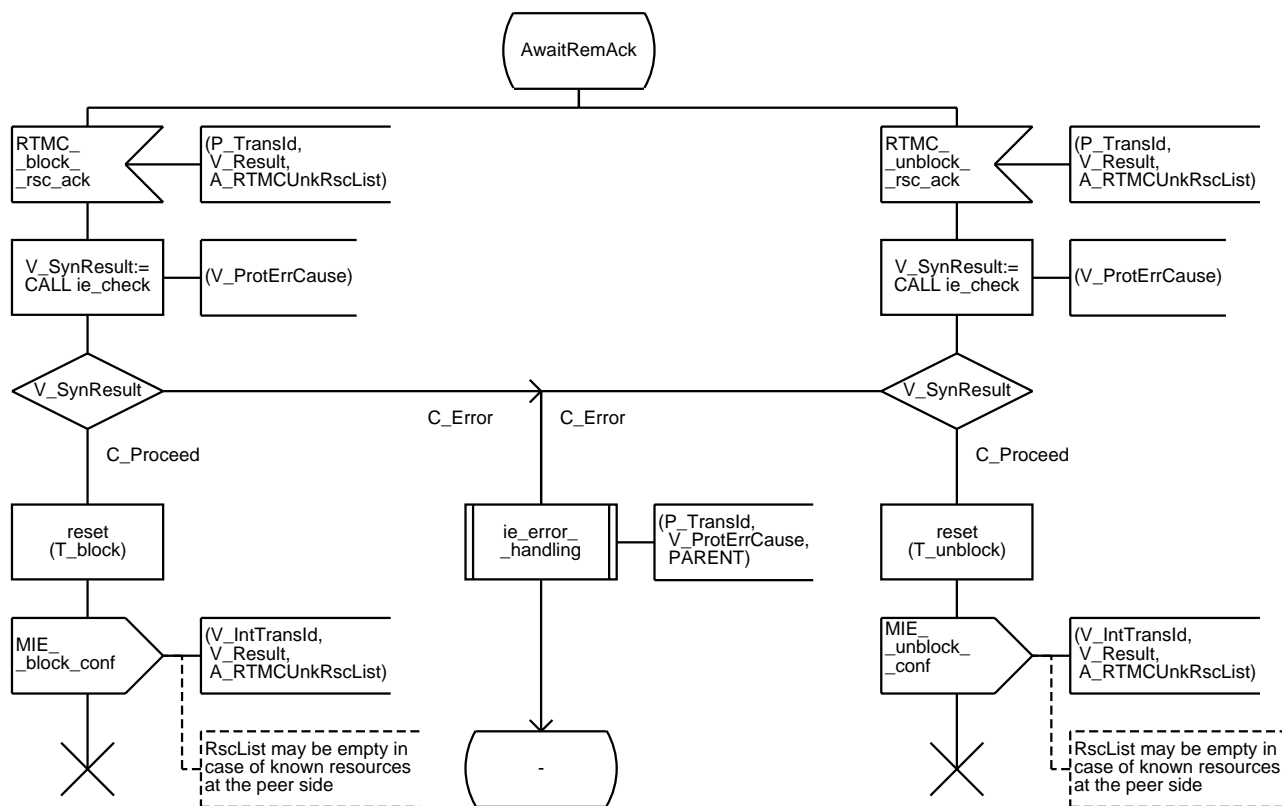
/*AN_CPROT is responsible for sending
receiving RTMC messages*/

/* Handling of
pseudo events for
error handling*/



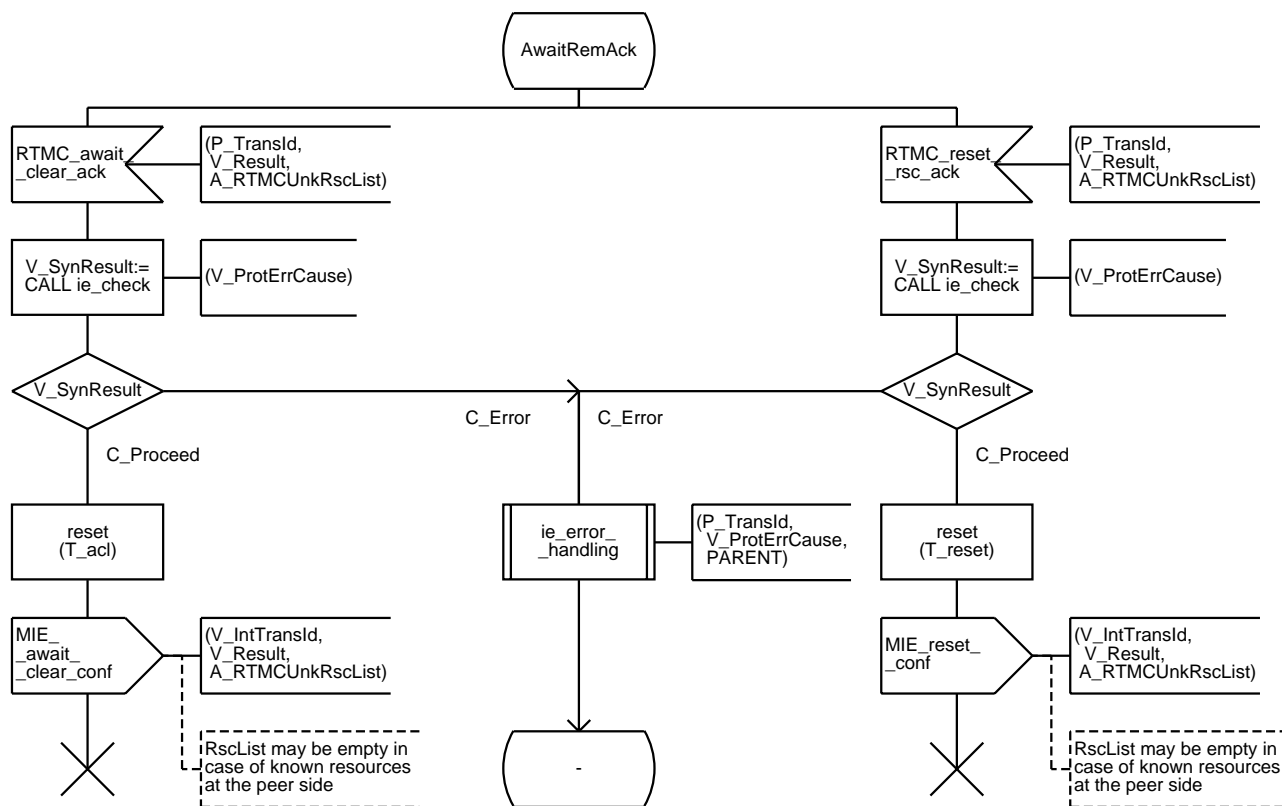
/*AN_CPROT is responsible for sending
receiving RTMC messages*/

/*Handling of SN
Acknowledgements */



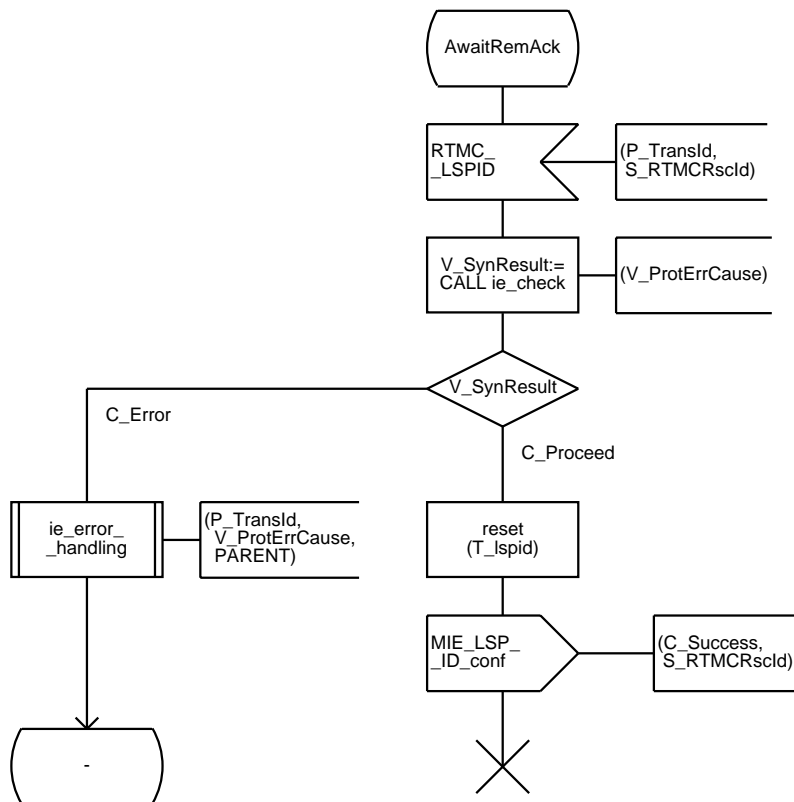
/*AN_CPROT is responsible for sending
receiving RTMC messages*/

/*Handling of SN
Acknowledgements */



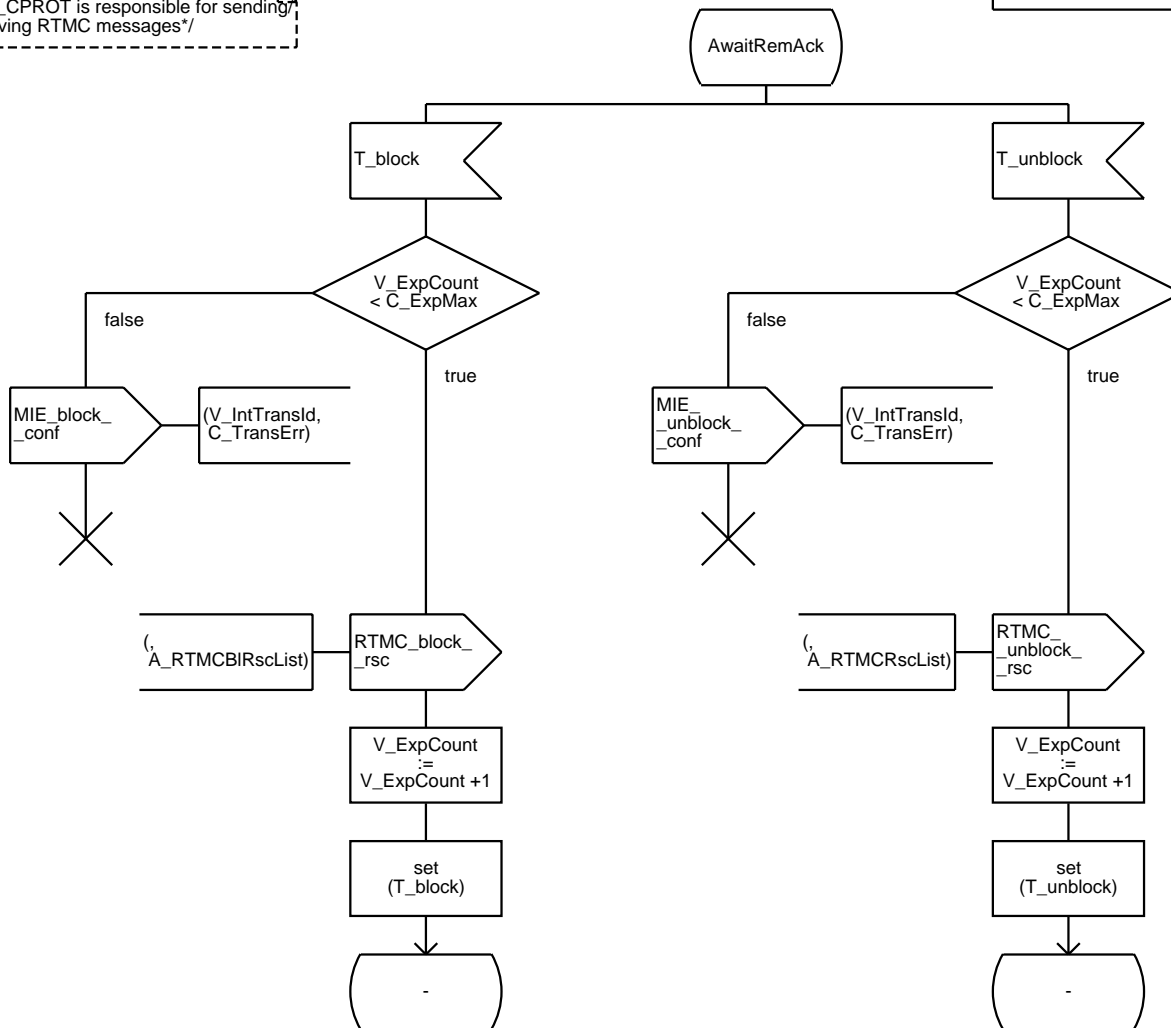
/*AN_CPROT is responsible for sending/
receiving RTMC messages*/

/*Handling of SN
Acknowledgements */



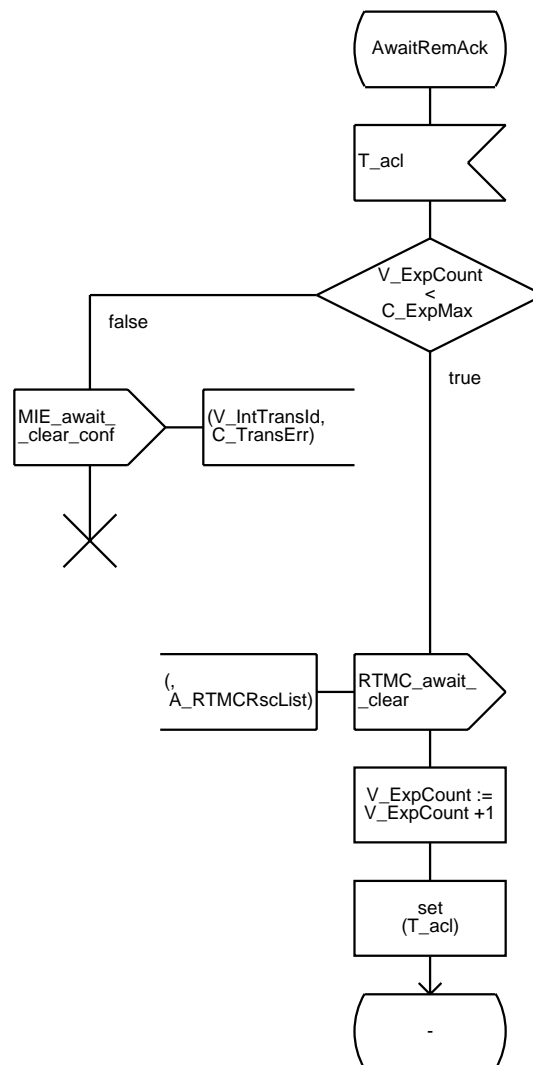
/*AN_CPROT is responsible for sending
receiving RTMC messages*/

/* Timer expirations */



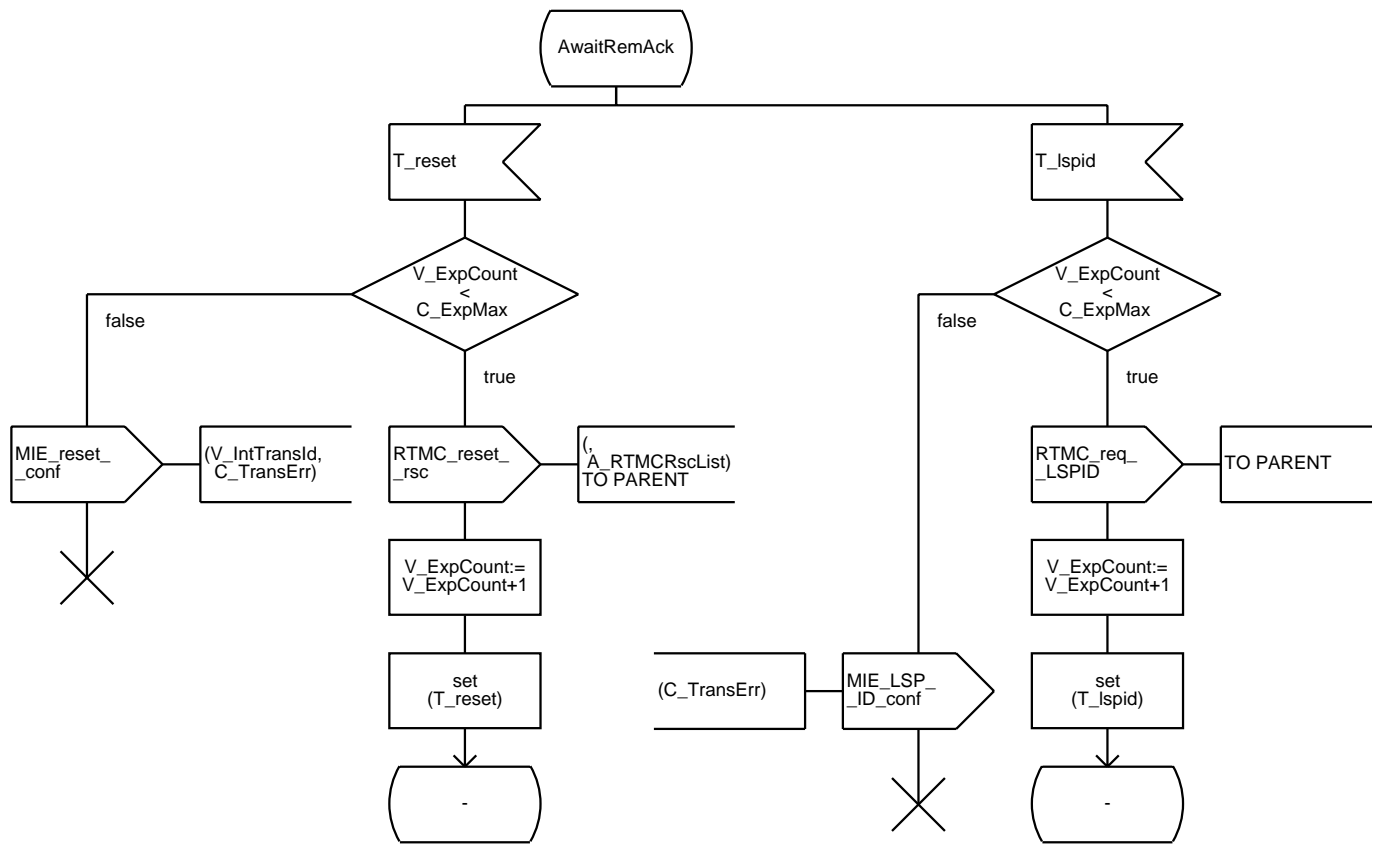
/*AN_CPROT is responsible for sending
receiving RTMC messages*/

/* Timer expirations */



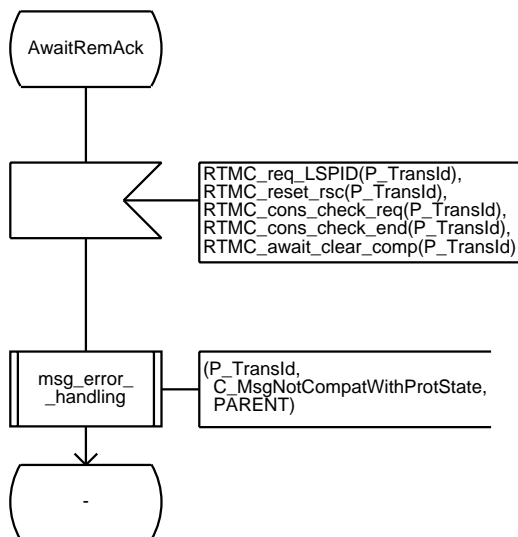
/*AN_CPROT is responsible for sending
receiving RTMC messages*/

/* Timer expirations */



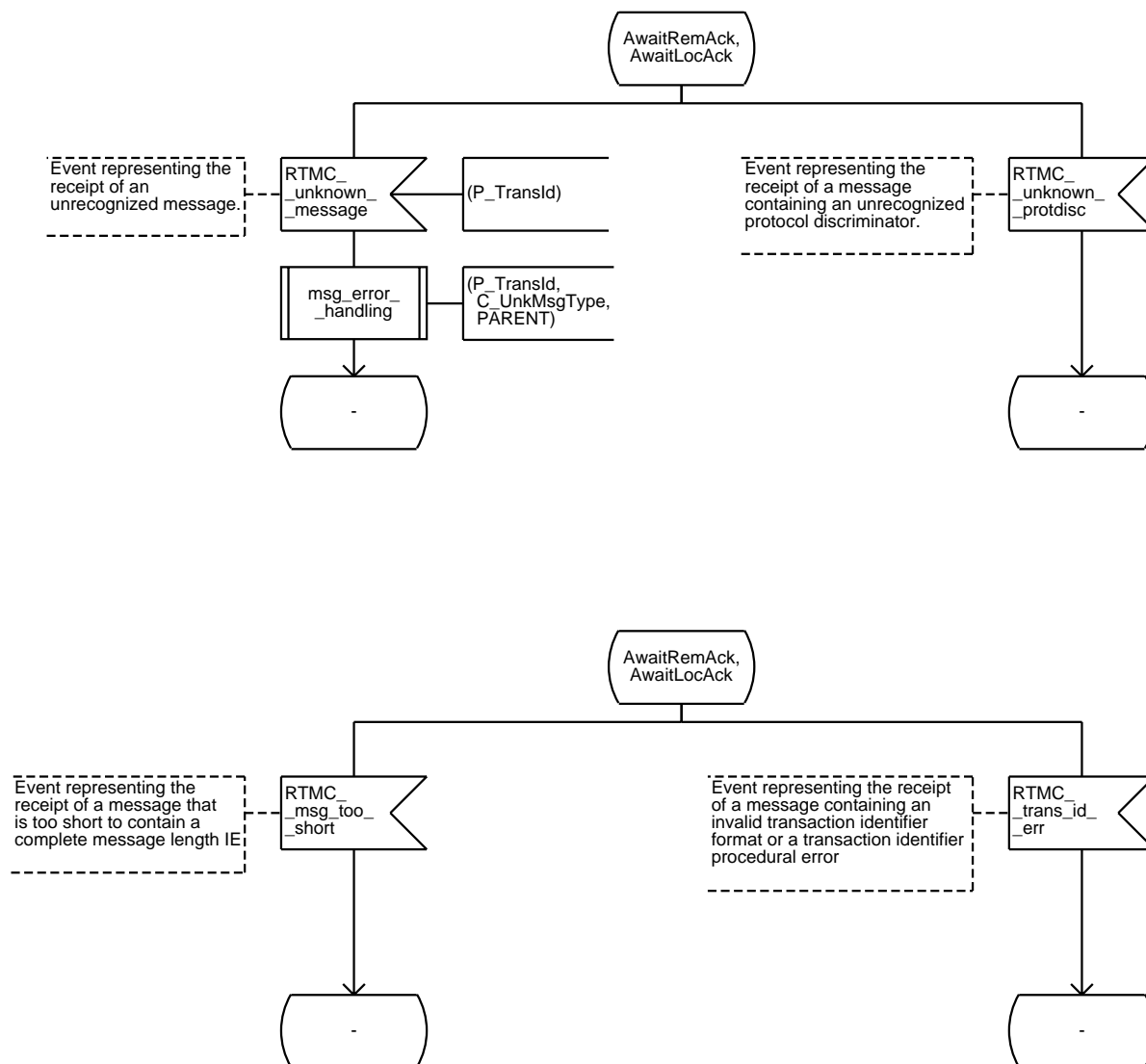
/*AN_CPROT is responsible for sending
receiving RTMC messages*/

/* Handling of
unexpected events */



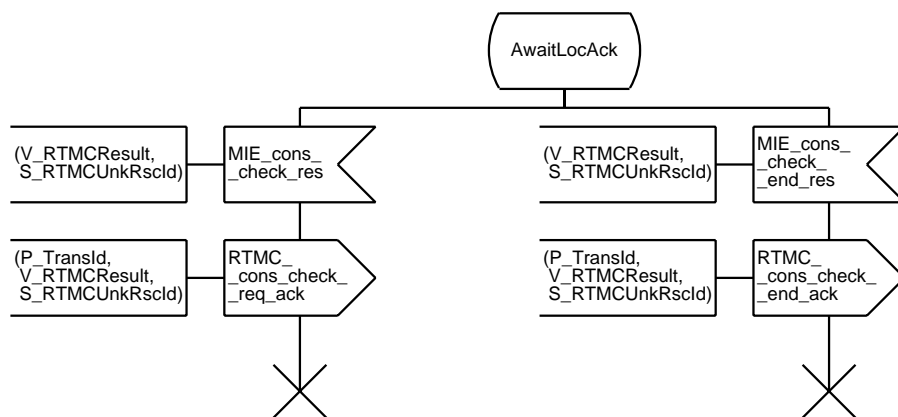
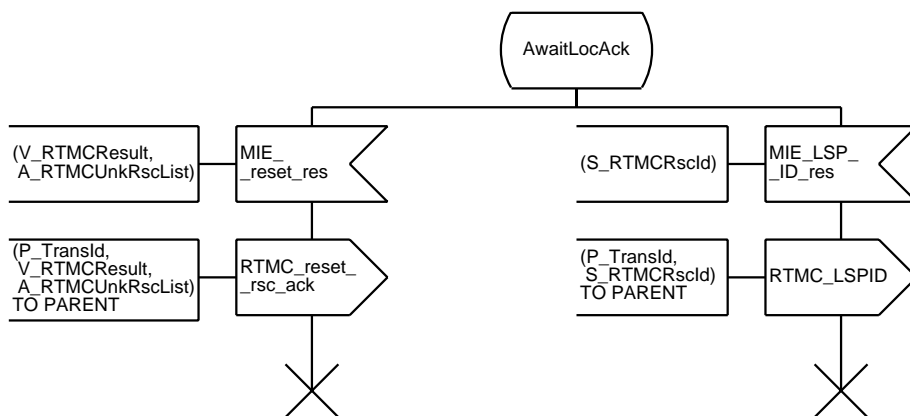
/*AN_CPROT is responsible for sending
receiving RTMC messages*/

/* Handling of
pseudo events for
error handling*/



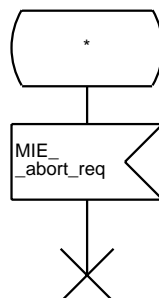
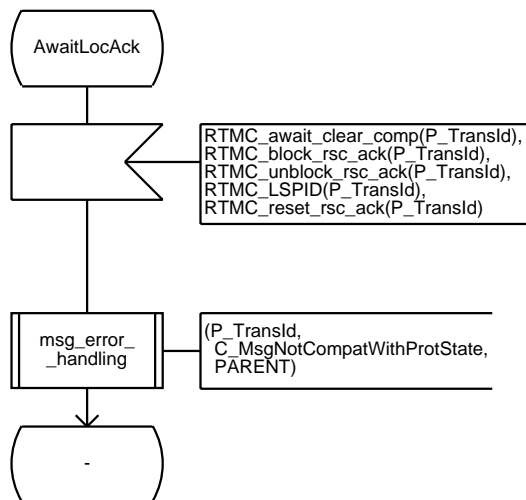
/*AN_CPROT is responsible for sending
receiving RTMC messages*/

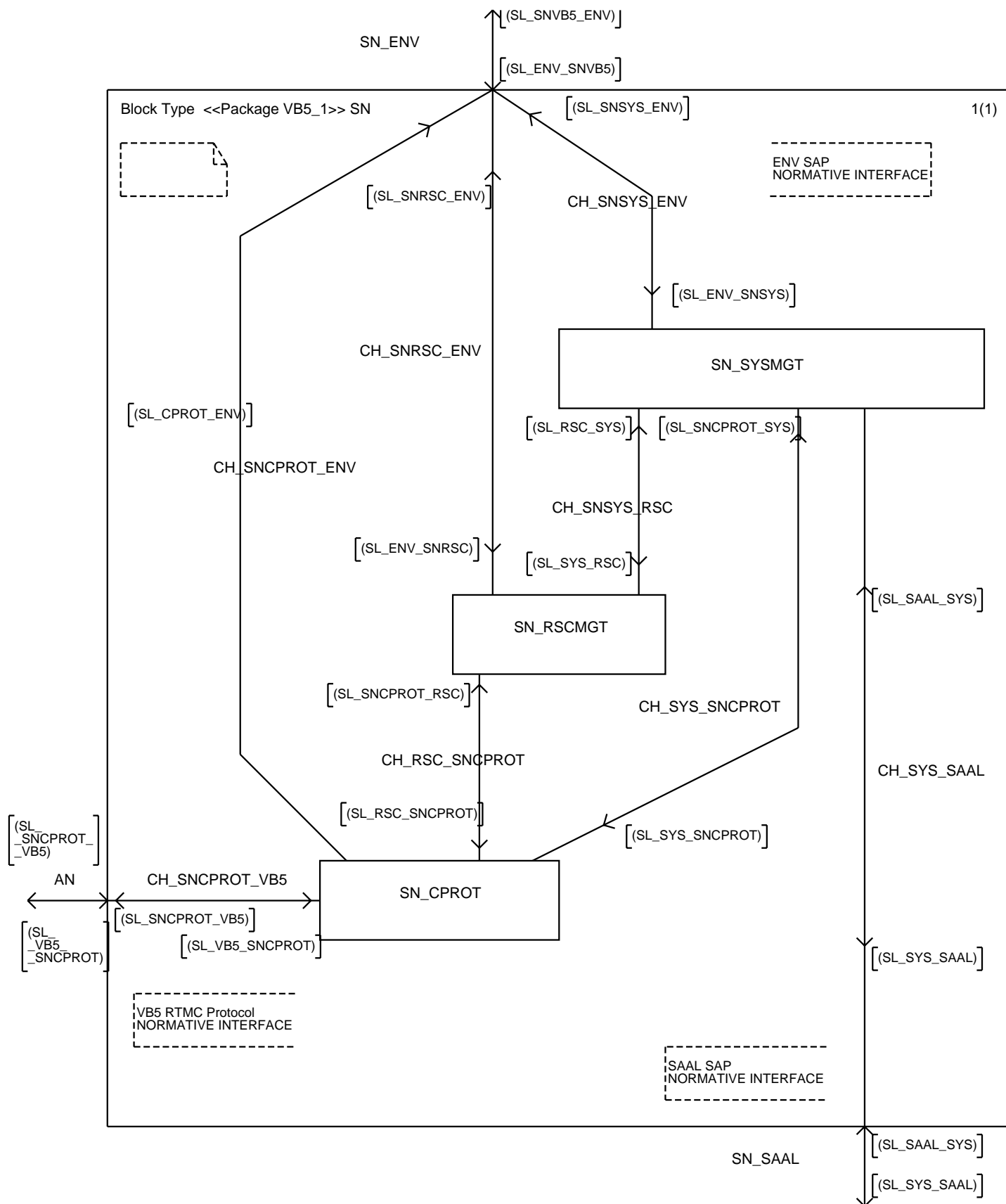
/*Handling of system
management acknowledgements */

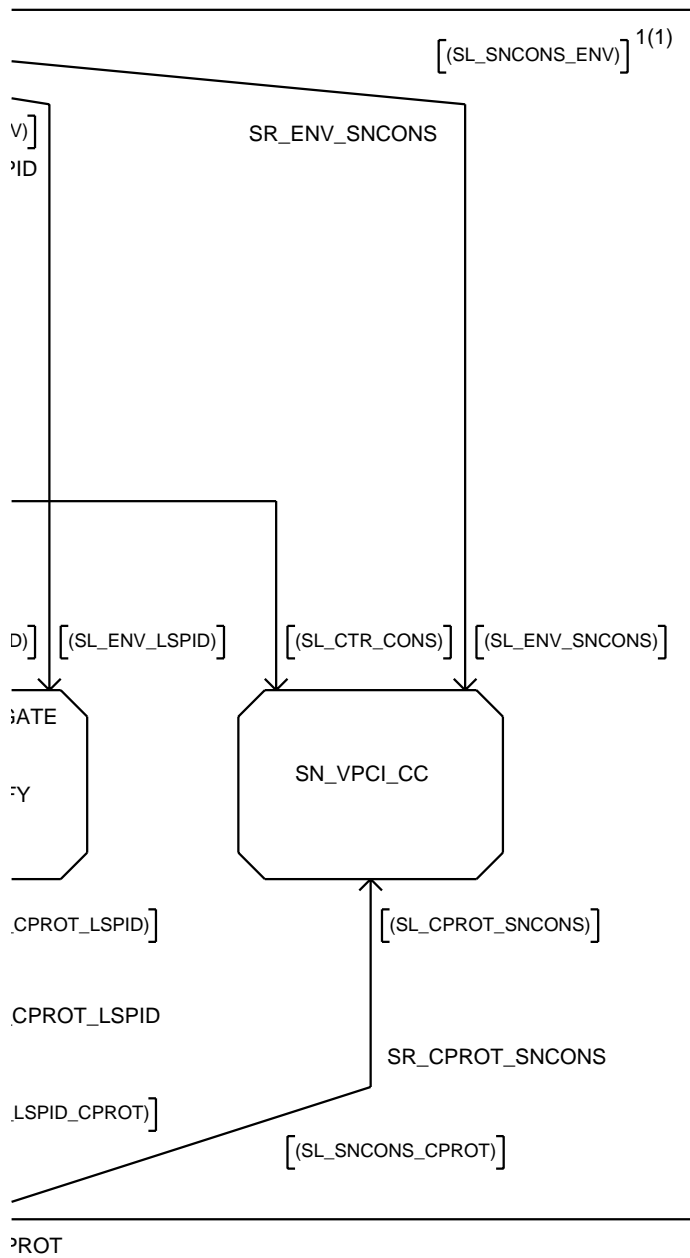


/*AN_CPROT is responsible for sending
receiving RTMC messages*/

/* Handling of
unexpected events */







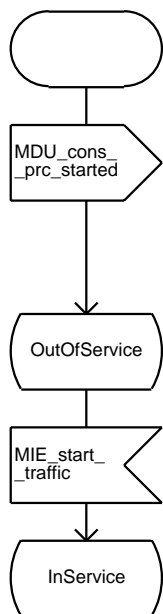
/*VB5.1 System Management
Process SN_VPCI_CC
Applicable for SN side*/

/* Definitions and declarations
for process SN_VPCI_CC */

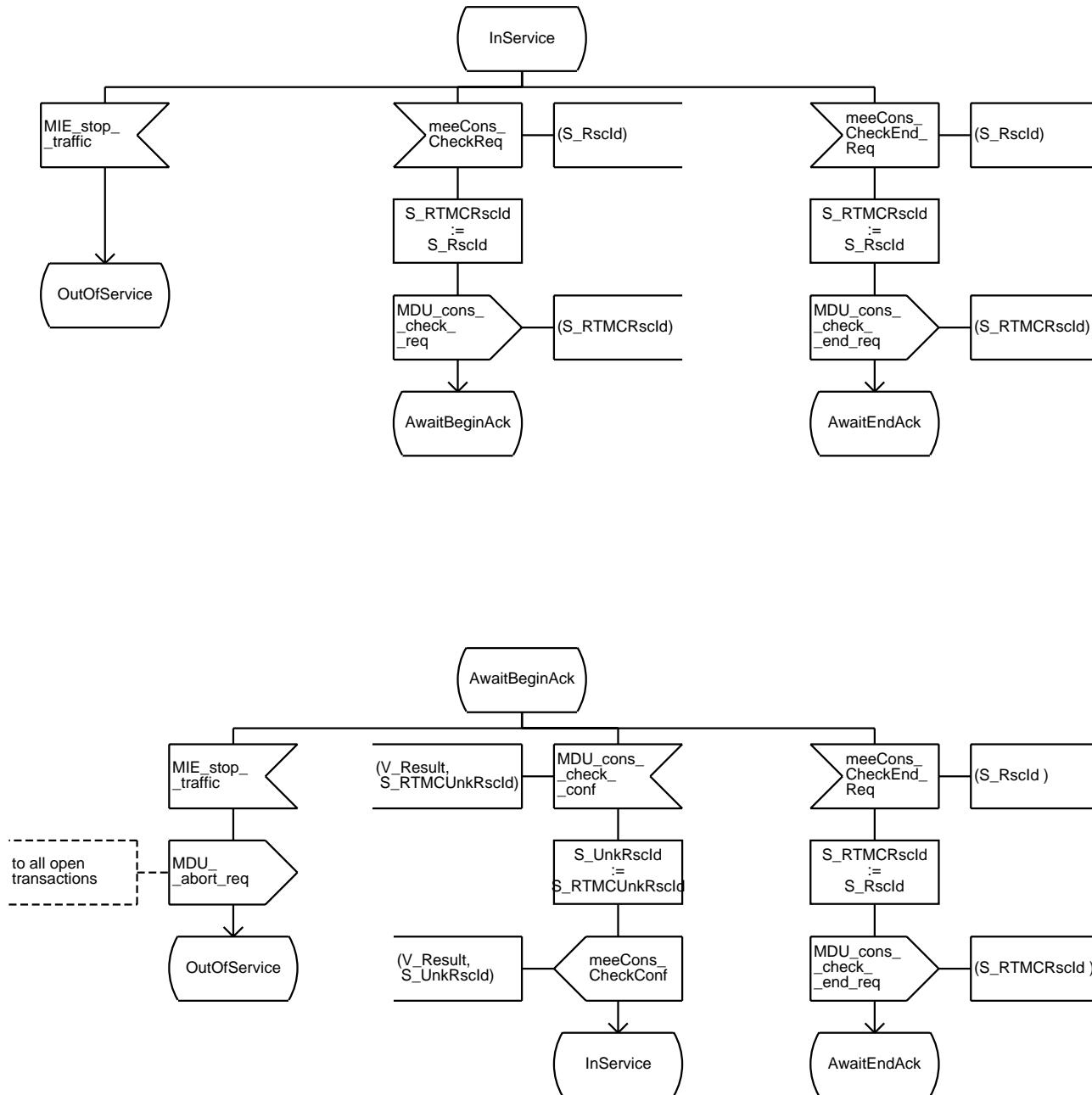
/*State descriptions
OutOfService: Initial state. Process is waiting for activation with start traffic.
InService: Process is waiting for primitive from environment
AwaitBeginAck: Process is waiting for confirmation of check begin from AN side.
AwaitEndAck: Process is waiting for confirmation of check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

/*Signal data declarations*/
DCL
S_Rscld ST_Rscld; /*Resource Identifier reported from/to local side*/
S_RTMC Rscld ST_Rscld; /*Resource Identifier reported from/to peer side*/
/**/
DCL
S_UnkRscld ST_Rscld; /* Unknown Resource Identifier reported from/to local side*/
S_RTMCUnkRscld ST_Rscld; /* Unknown Resource Identifier reported from/to peer side*/
/**/
DCL
V_Result IT_Result; /*Result*/

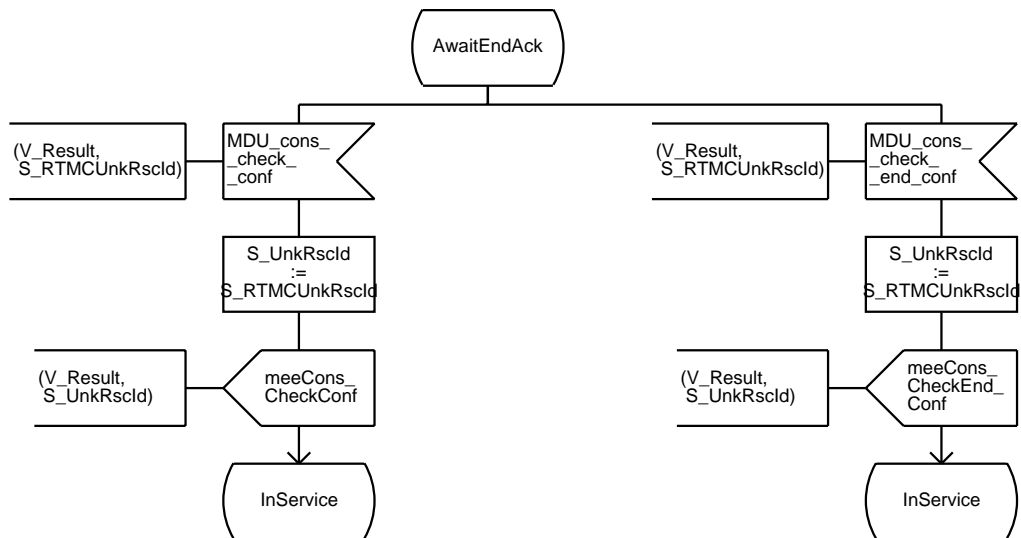
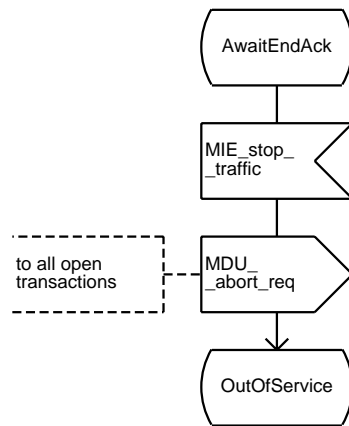
/*VB5.1 System Management
Process SN_VPCI_CC
Applicable for SN side*/

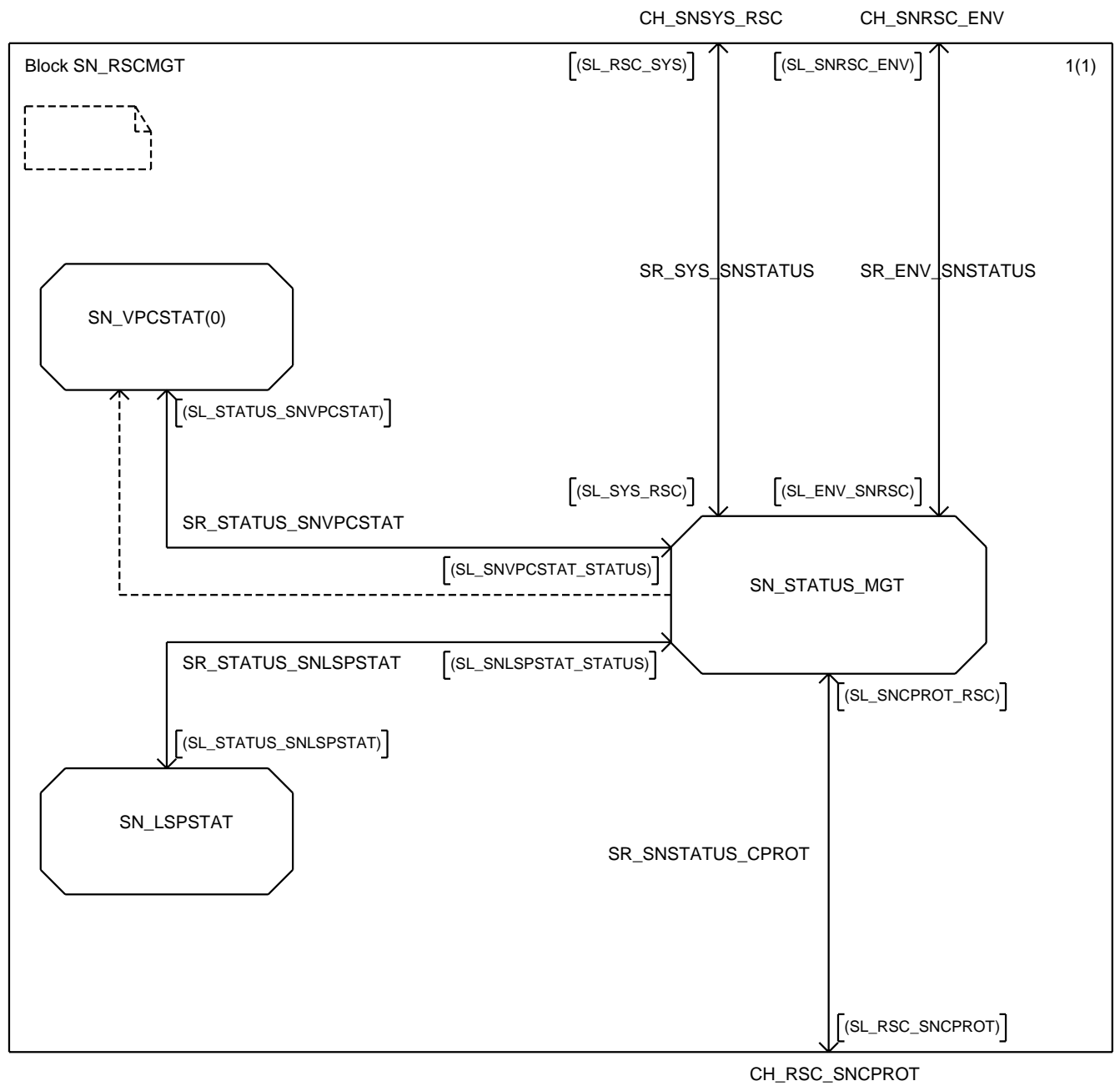


/*VB5.1 System Management
Process SN_VPCI_CC
Applicable for SN side*/



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






```
/*SN_STATUS_MGT controls L-
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
AwaitBIFSMack:     Status change was requested, blocking of single STATFSMs is running
AwaitAcIFSMack:    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_Rsclid  ST_Rsclid;
  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
```

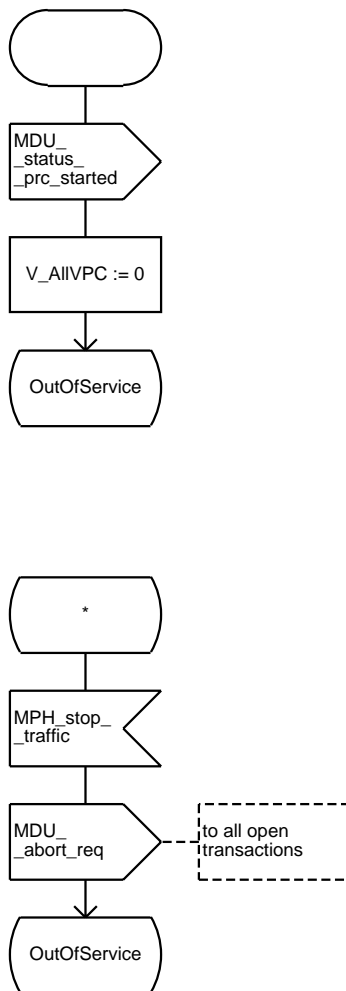
/*SN_STATUS_MGT controls state changes of VPCs and LSP reported by the AN*/

/* Definintions and declarations for SN_STATUS_MGT*/

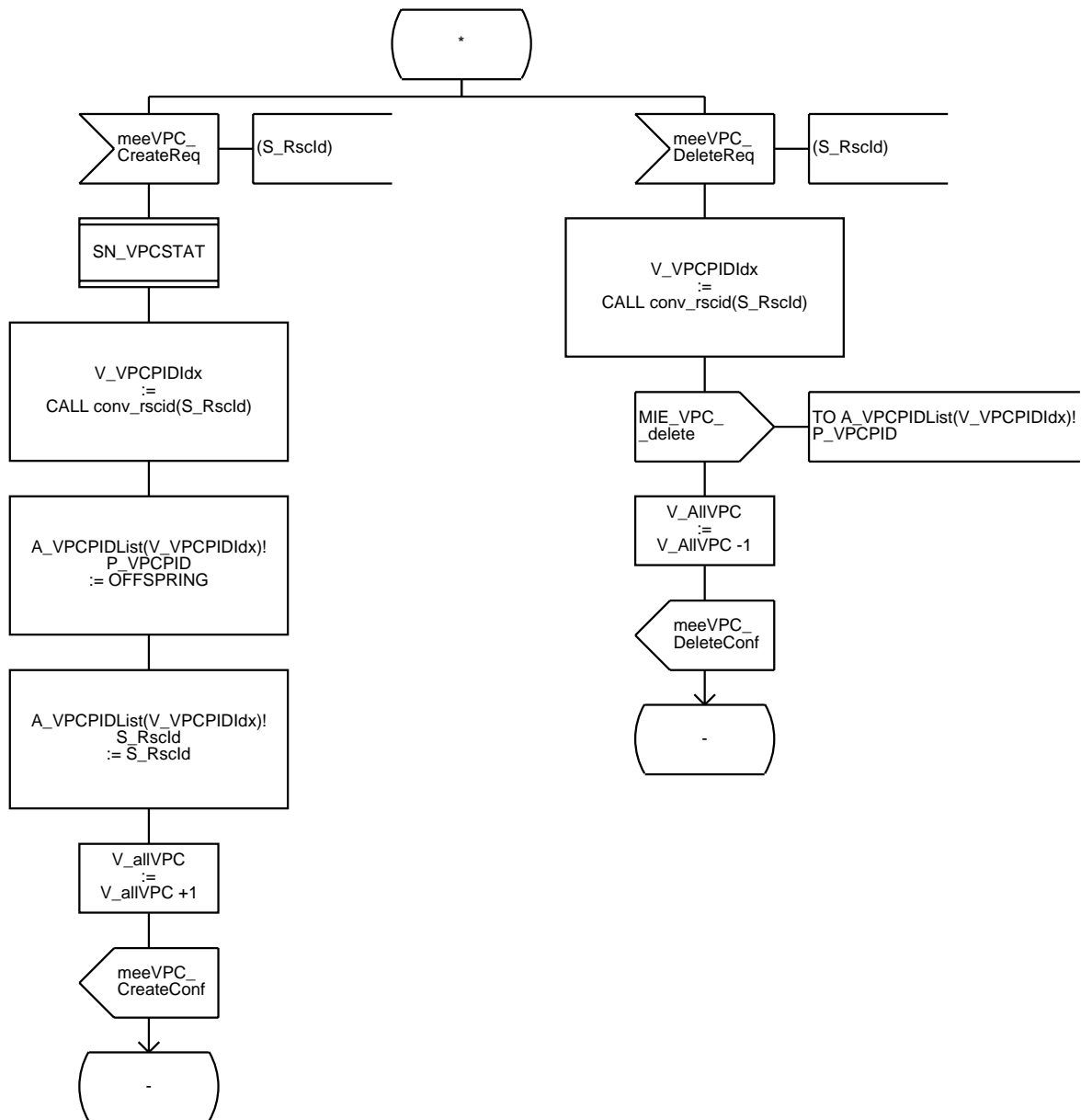
```
/*Signal data declarations*/
/**/
DCL
S_RscId      ST_RscId; /*Resource identifier*/
/**/
DCL
A_RscList     AT_RscList; /*Resource Identifier List reported from/to local side*/
A_RTMCRRscList AT_RscList; /*Resource identifier List reported from/to peer side*/
/**/
DCL
A_RTMCBIRscList AT_RTMCBIRscList; /*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_BIRscList AT_BIRscList;
/**/
/*Variable for storing AN_VPCSTAT PIDs*/
DCL A_VPCPIDList AT_VPCPIDList;
DCL V_VPCPIDIdx INTEGER; /*Variable for indexing PID Table*/
/**/
DCL V_AliVPC INTEGER; /*Total number of provisioned VPCs*/
```

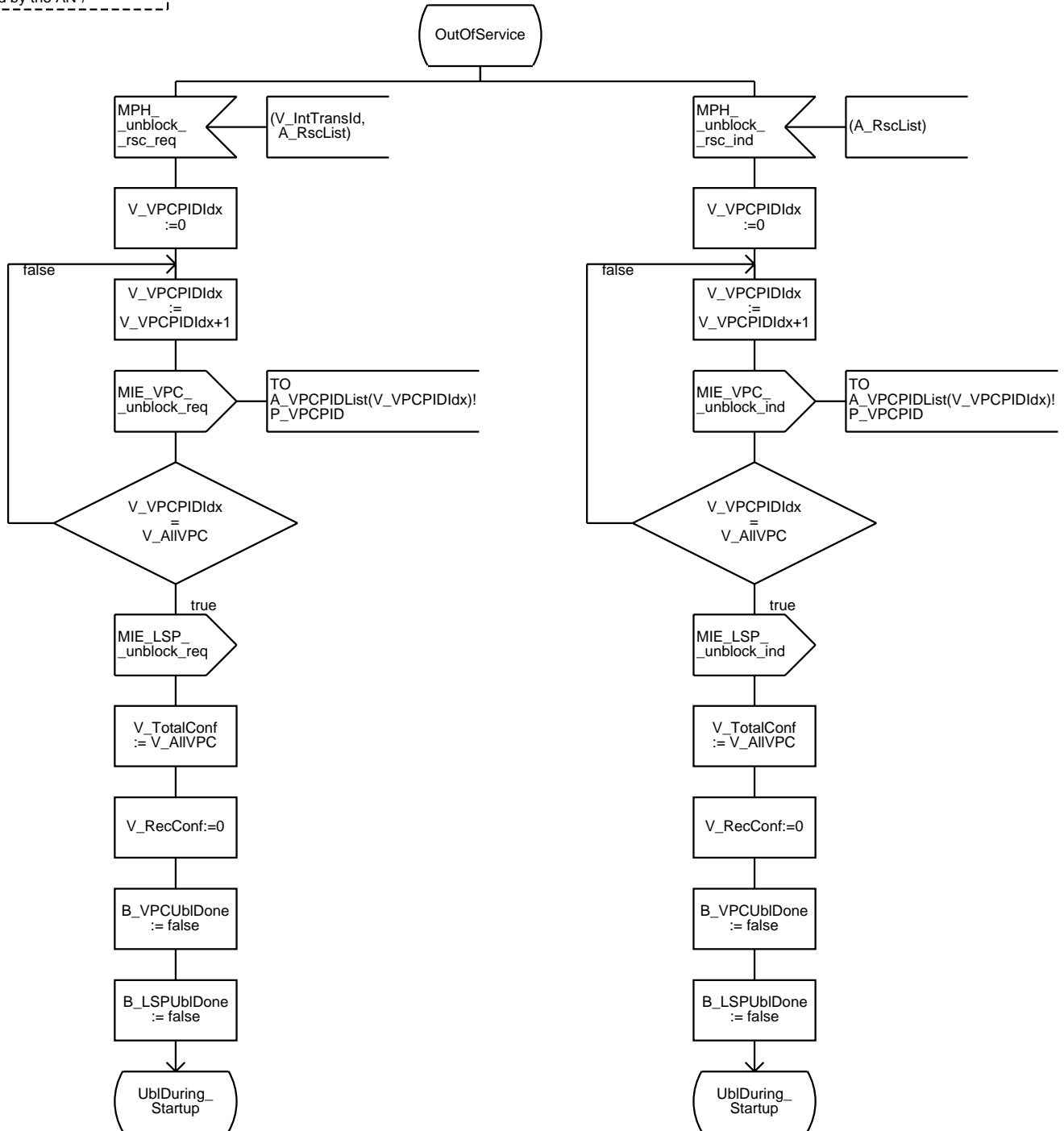
/*SN_STATUS_MGT controls
state changes of VPCs and LSP
reported by the AN*/



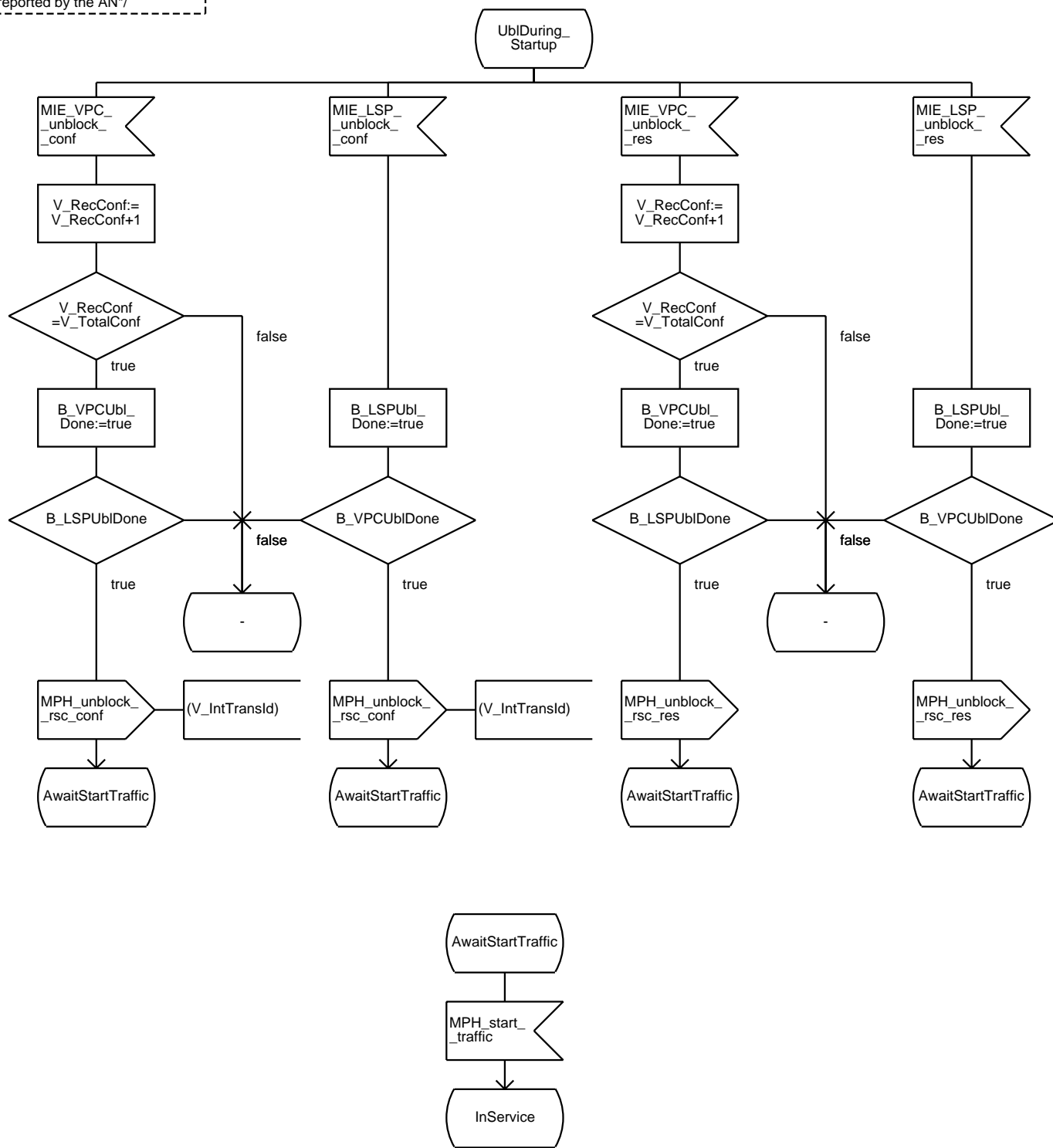
/*SN_STATUS_MGT controls
state changes of VPCs and LSP
reported by the AN*/



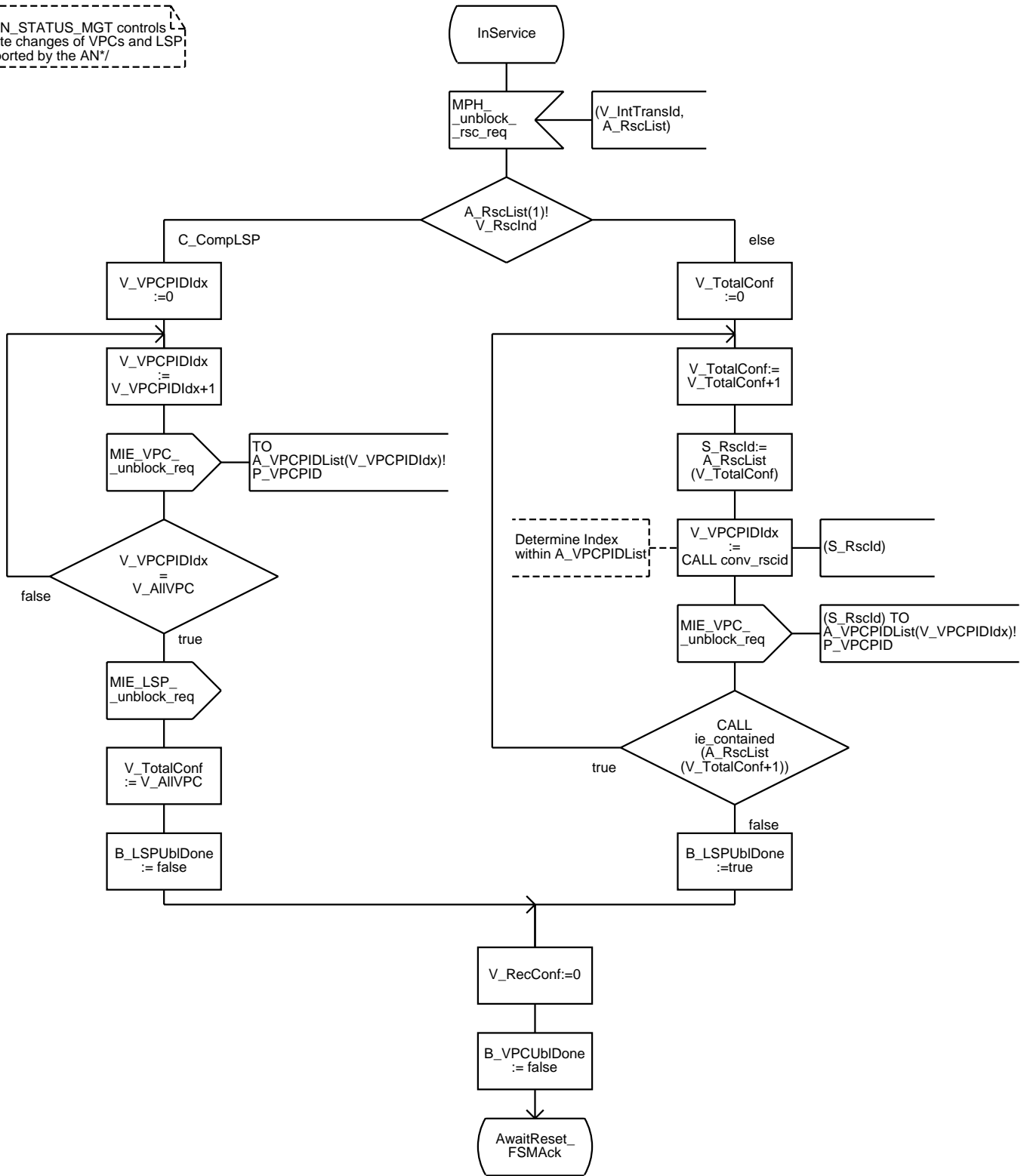
/*SN_STATUS_MGT controls the state changes of VPCs and LSP reported by the AN*/



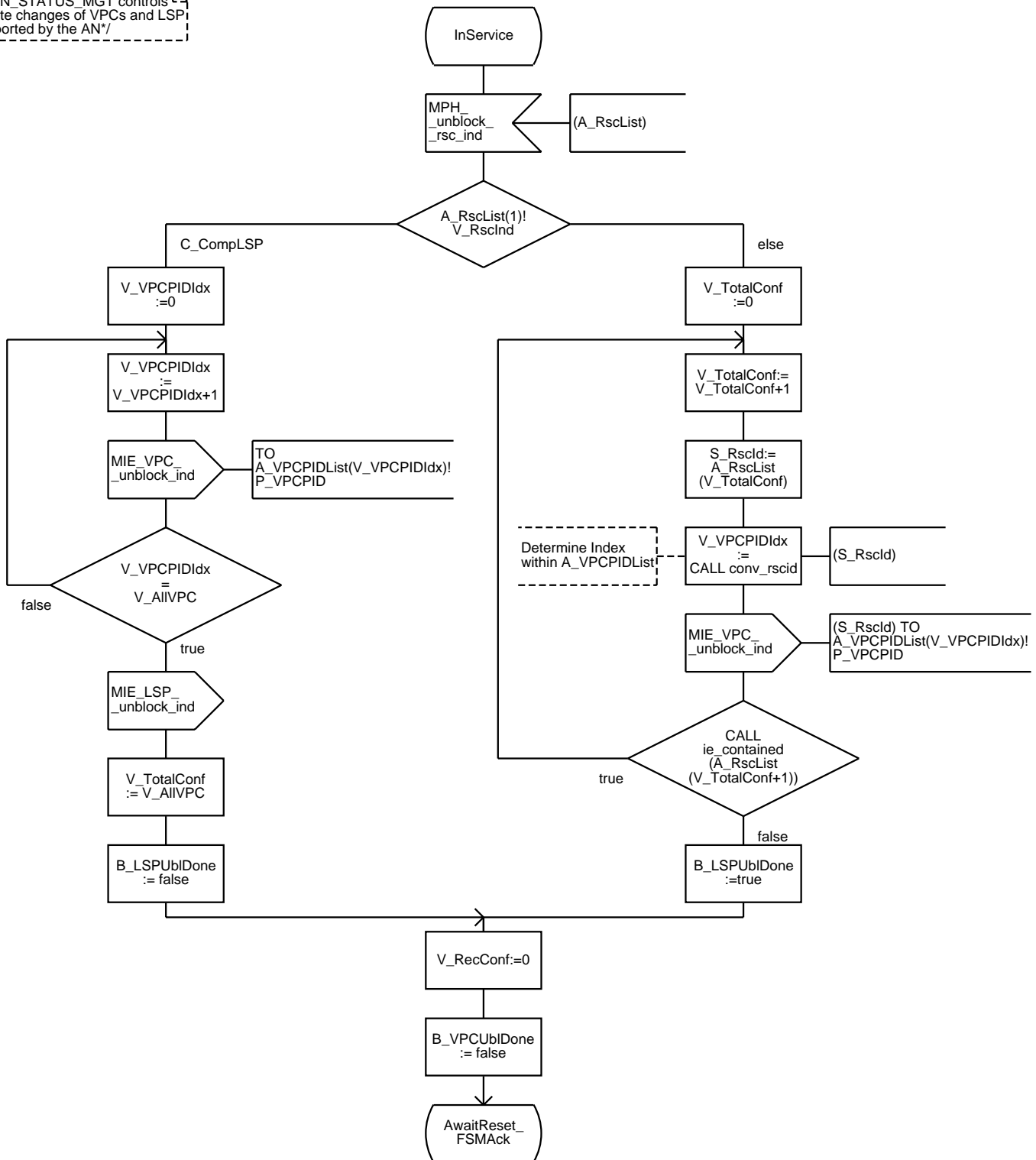
/*SN_STATUS_MGT controls the state changes of VPCs and LSP reported by the AN*/



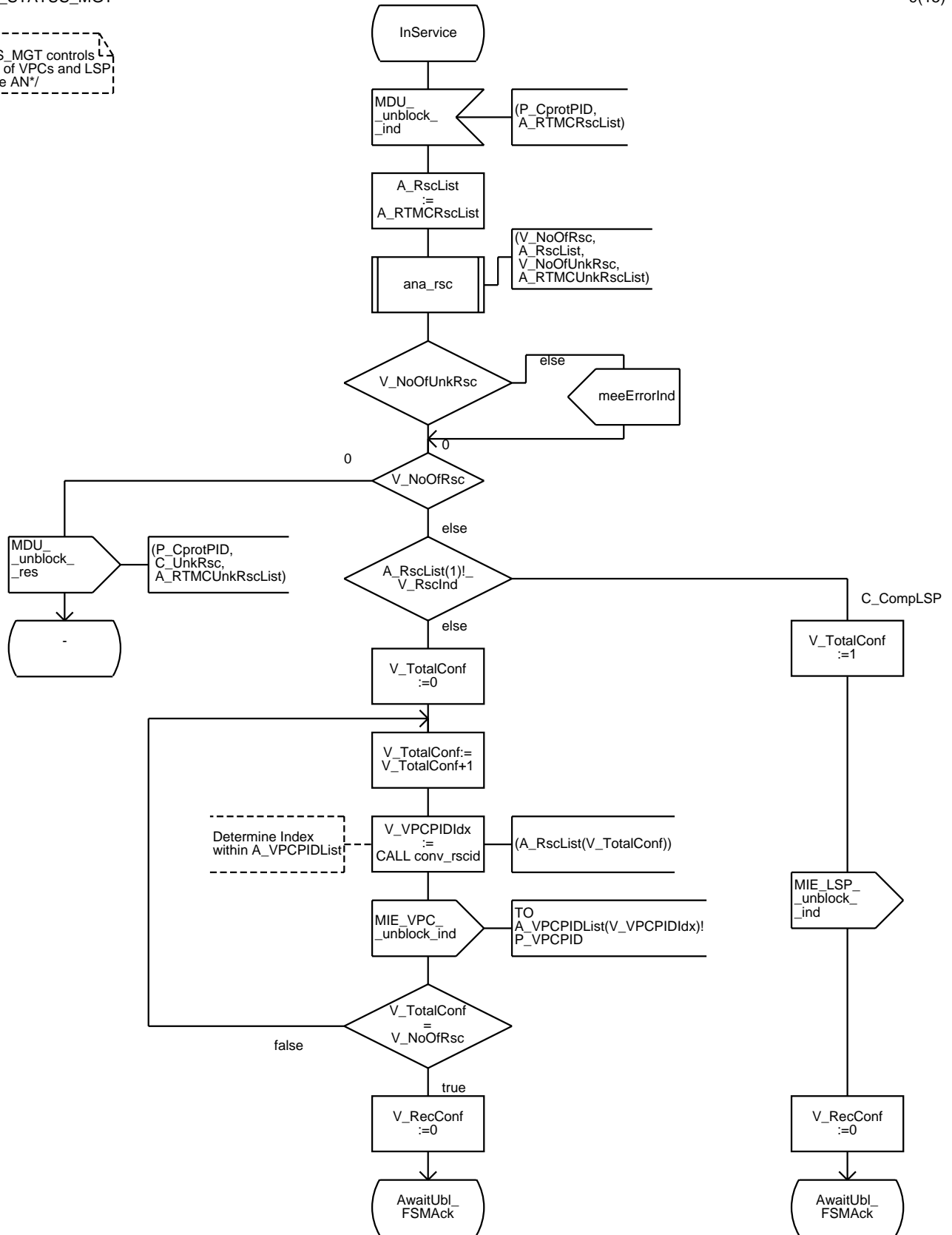
/*SN_STATUS_MGT controls the state changes of VPCs and LSP reported by the AN*/



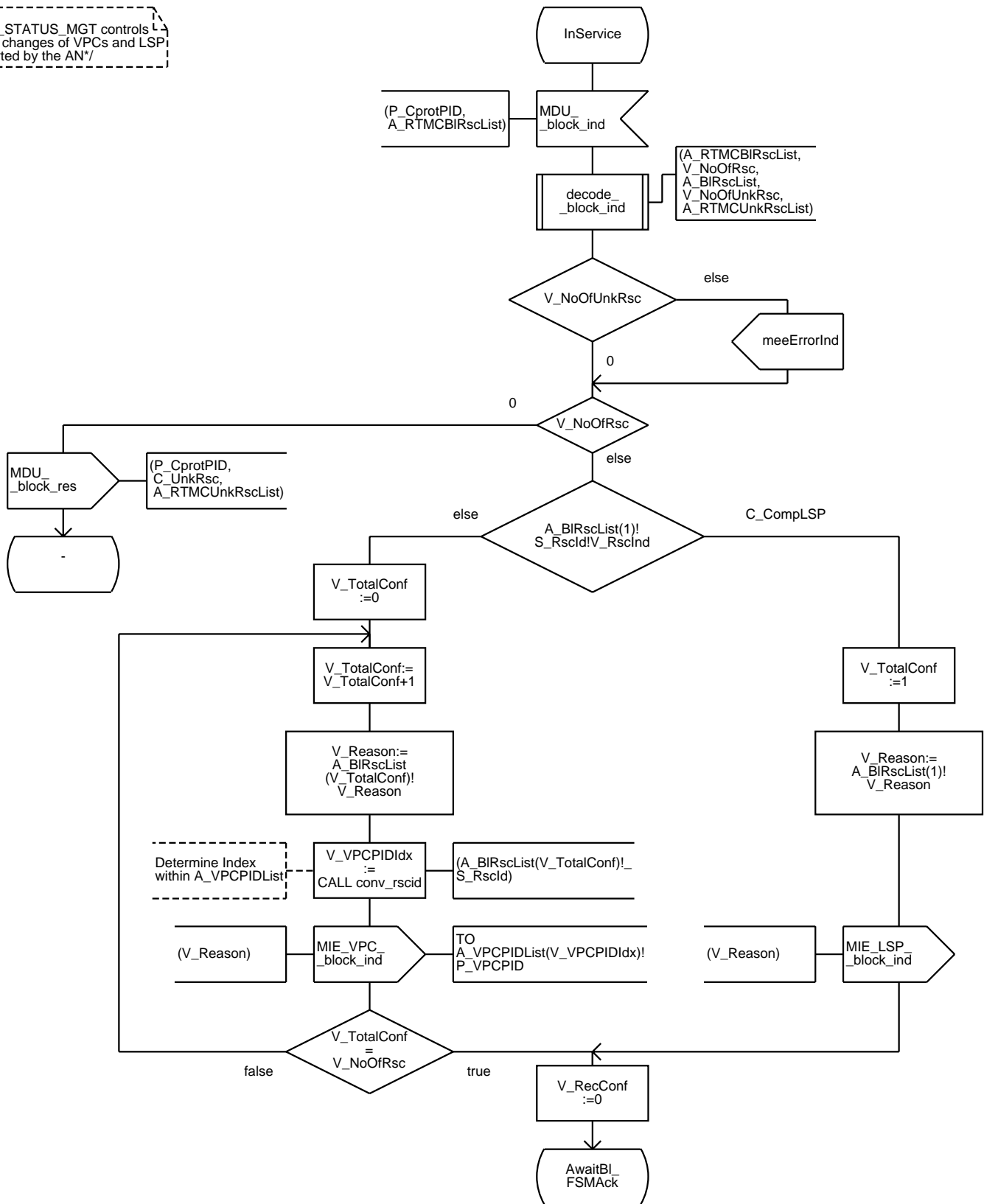
/*SN_STATUS_MGT controls
state changes of VPCs and LSP
reported by the AN*/



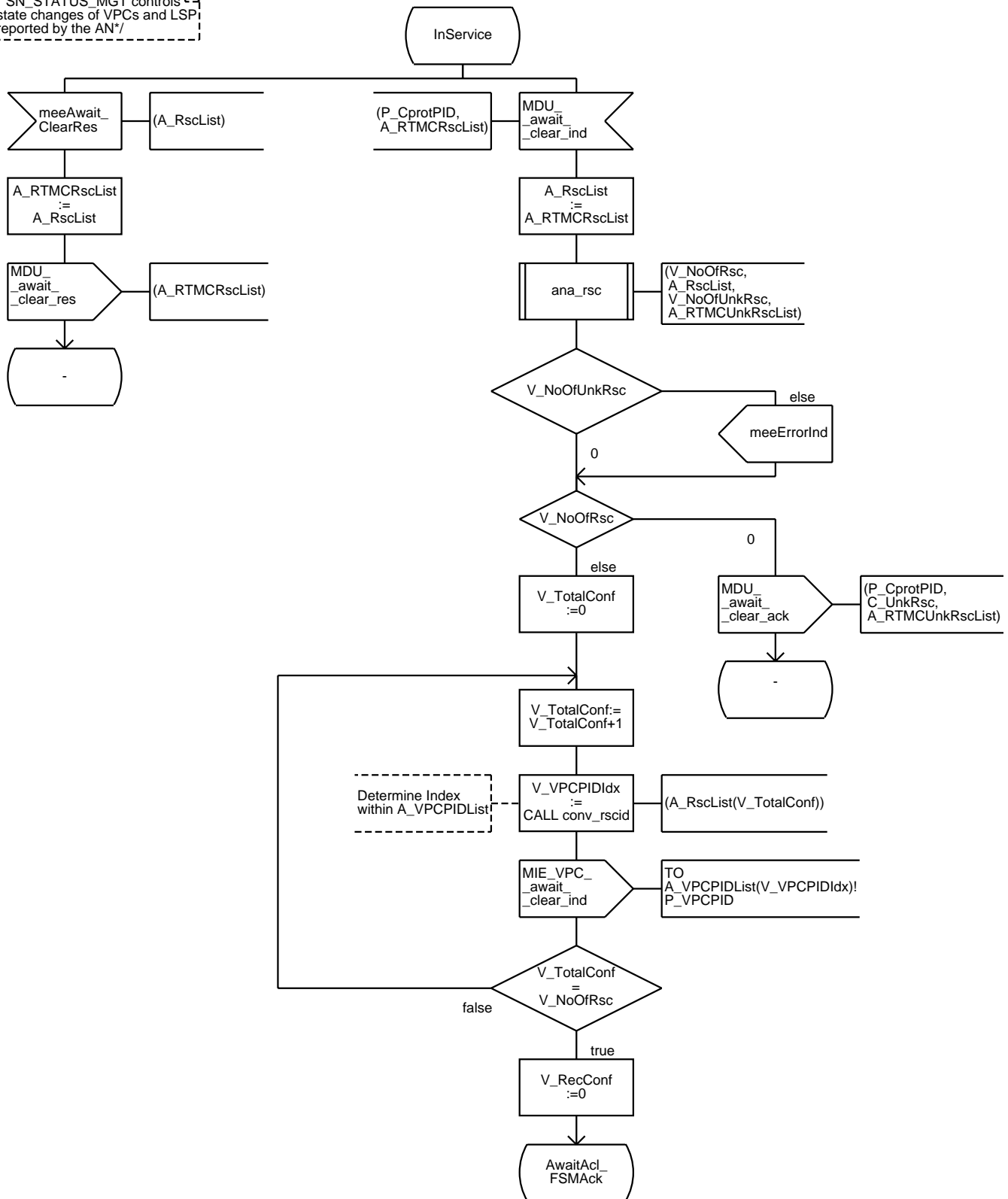
/*SN_STATUS_MGT controls
state changes of VPCs and LSP
reported by the AN*/



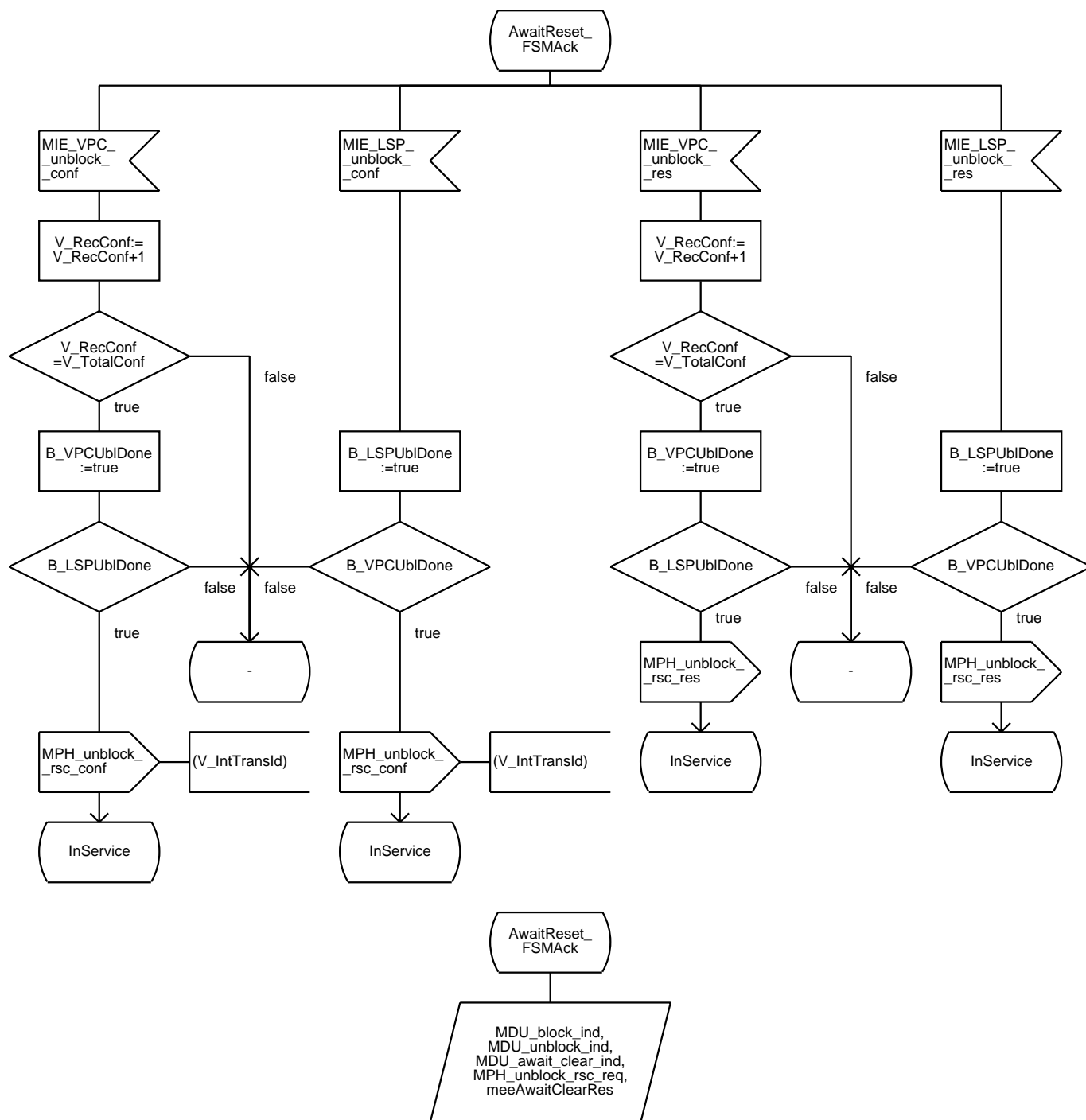
/*SN_STATUS_MGT controls
state changes of VPCs and LSP
reported by the AN*/



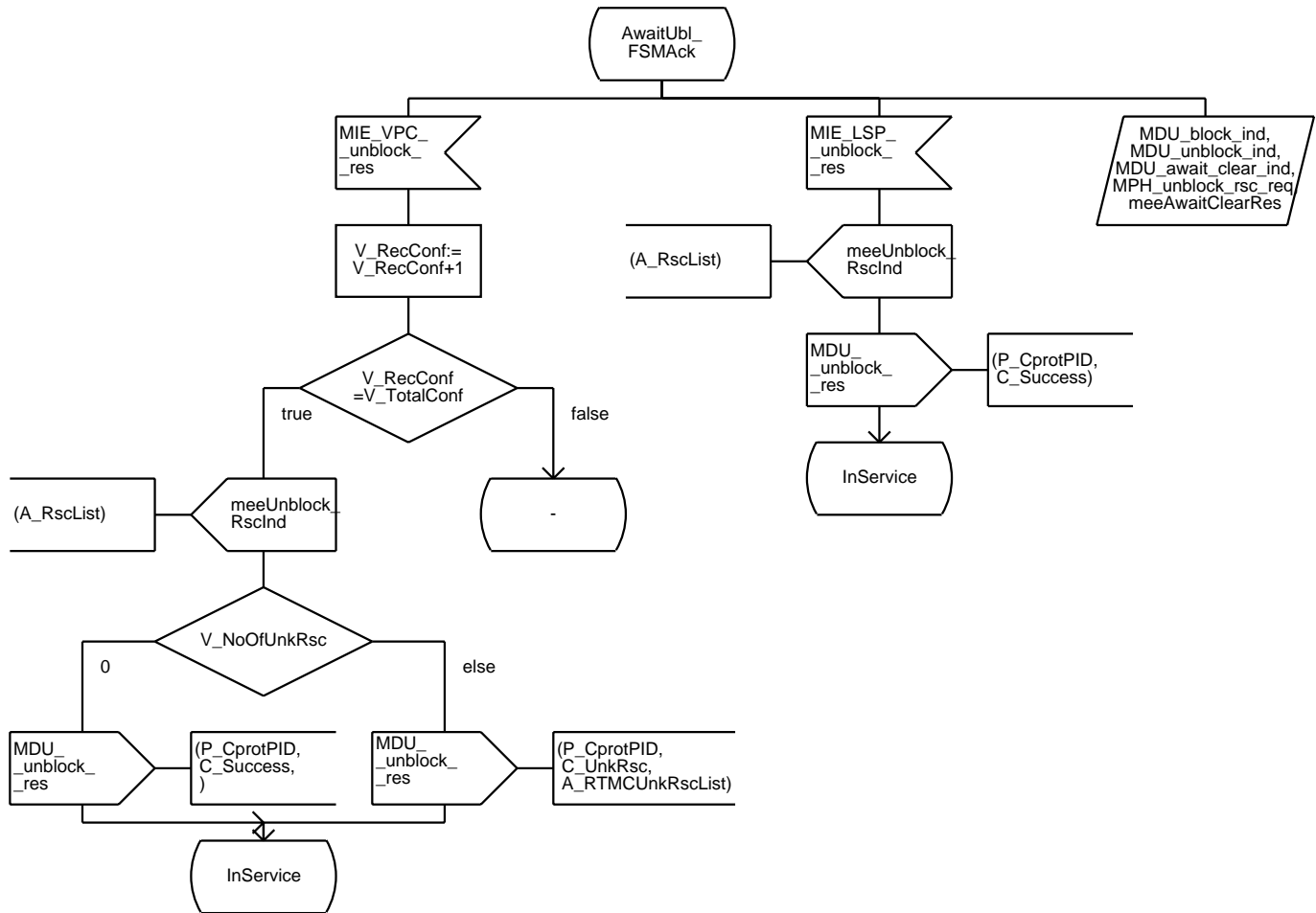
/*SN_STATUS_MGT controls
state changes of VPCs and LSP
reported by the AN*/



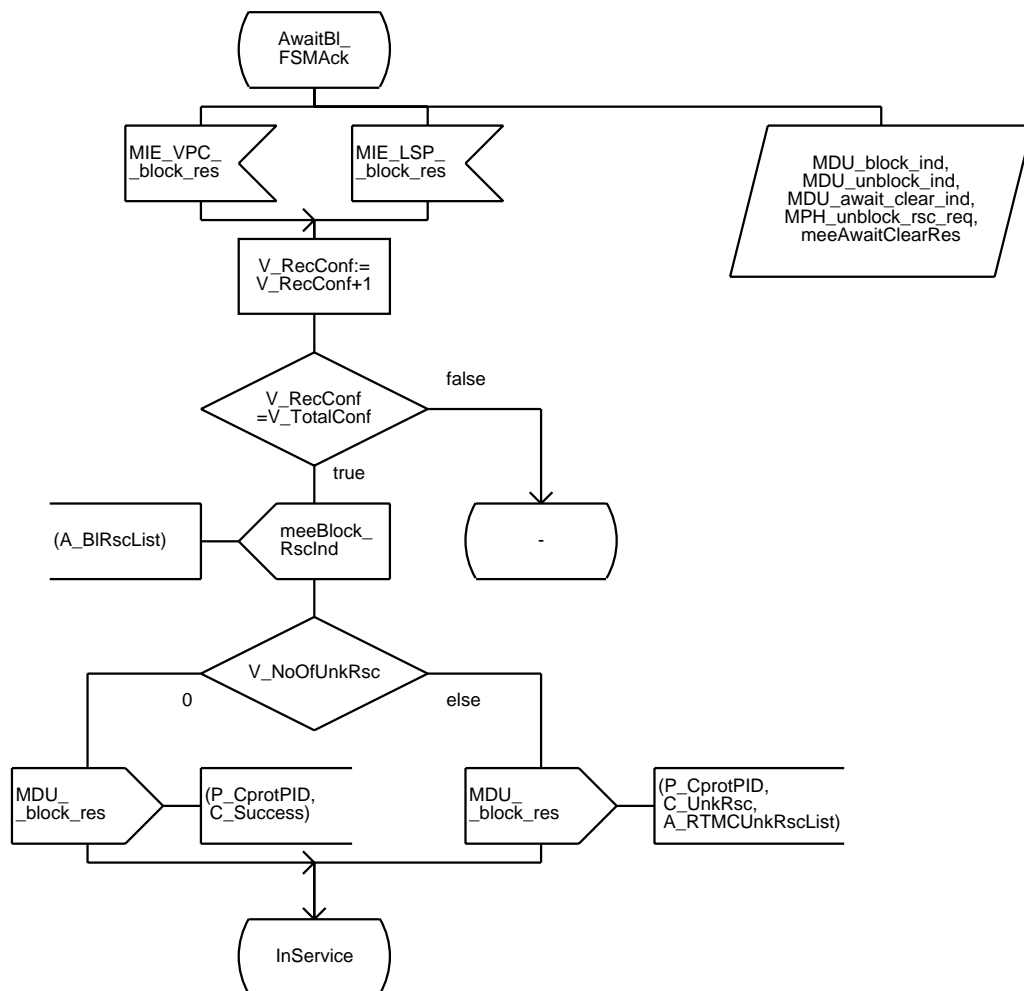
/*SN_STATUS_MGT controls the state changes of VPCs and LSP reported by the AN*/



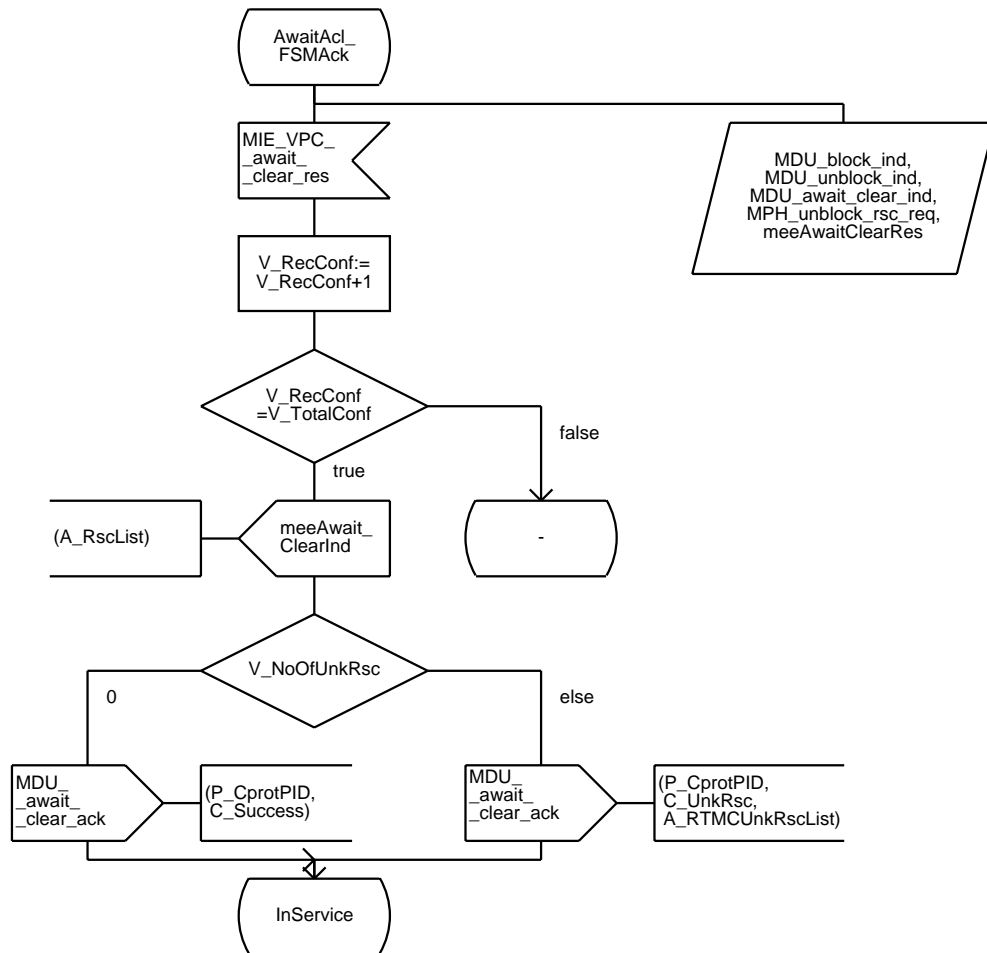
/*SN_STATUS_MGT controls
state changes of VPCs and LSP
reported by the AN*/



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



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



```

;FPAR
A RTMCBIRscList      AT RTMCBIRscList
IN/OUT V_NoOfRsc      INTEGER
IN/OUT A_BIRscList    AT_BIRscList
IN/OUT V_NoOfUnkRsc    INTEGER
IN/OUT A_RTMCUnkRscList AT_RscList;

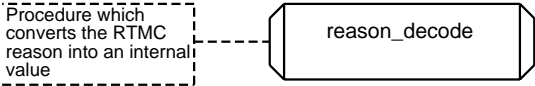
```

/*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*/

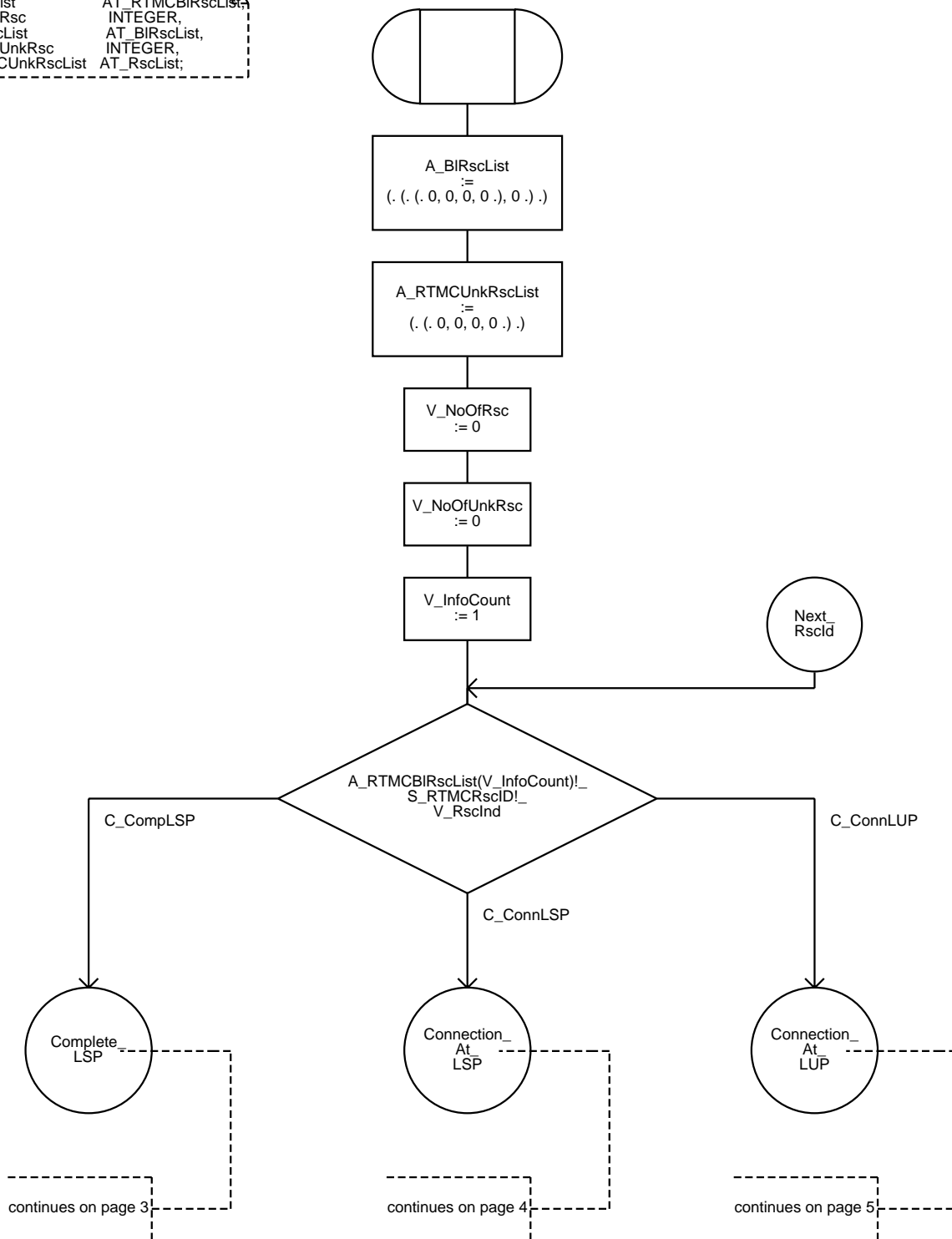
/*Data declarations for internal use*/
DCL
A_TempRscList AT_RscList ; /*Array of Resource Identifier Information Elements*/

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



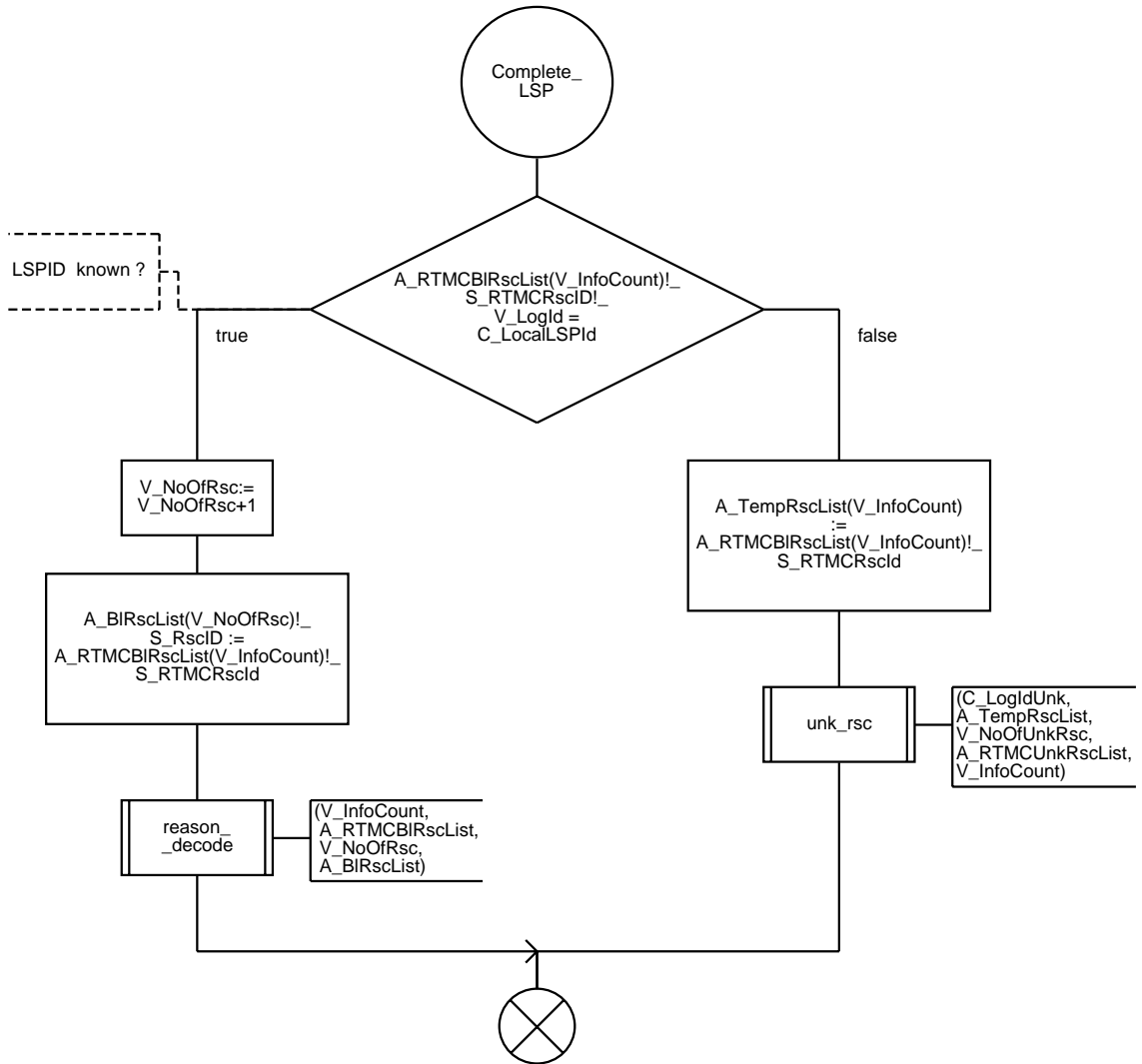

```

;FPAR
A_RTMCBIRscList  AT_RTMCBIRscList
IN/OUT V_NoOfRsc  INTEGER
IN/OUT A_BIRscList  AT_BIRscList
IN/OUT V_NoOfUnkRsc  INTEGER
IN/OUT A_RTMCUnkRscList  AT_RscList;
    
```



```

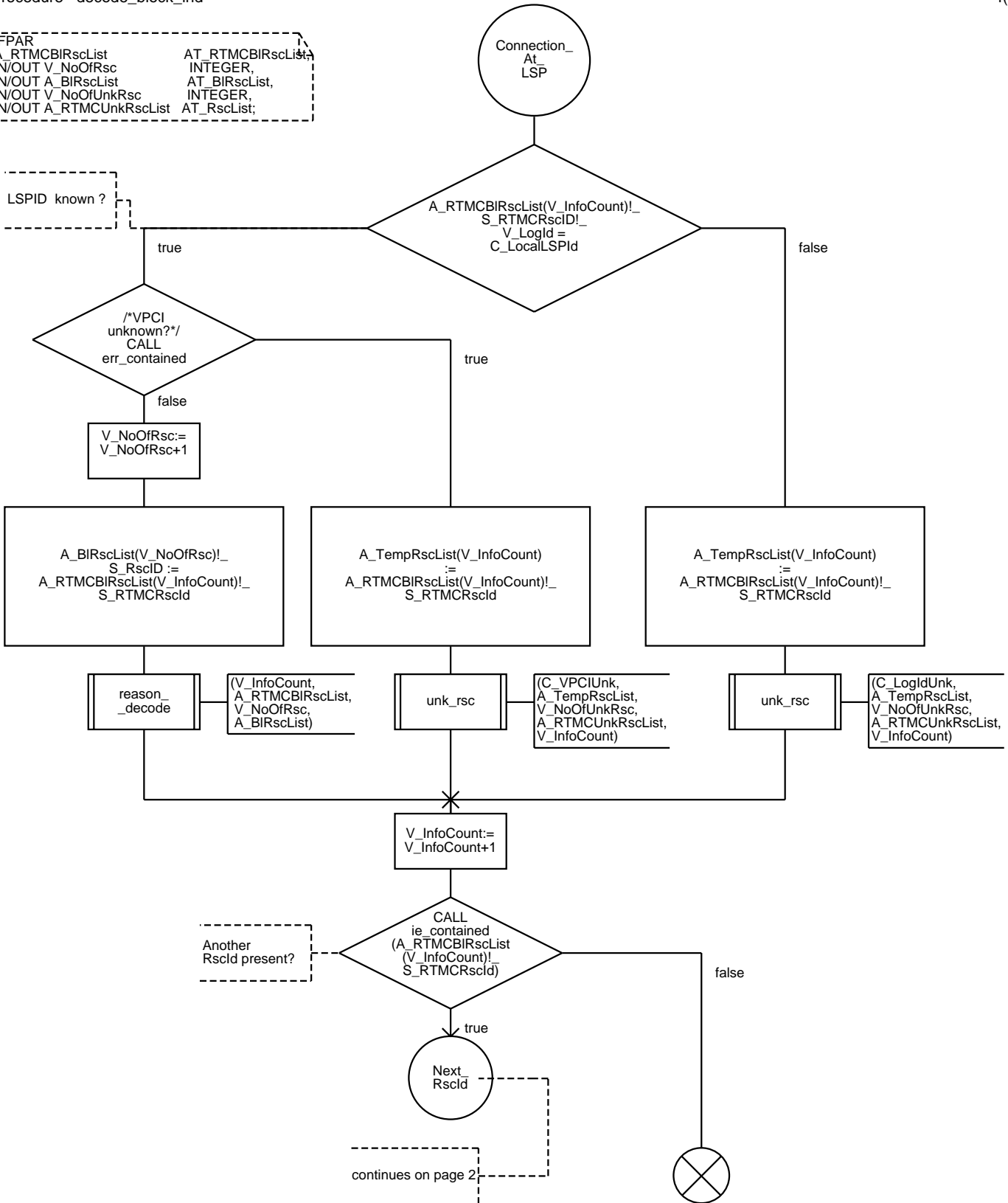
;FPAR
A_RTMCBIRscList      AT_RTMCBIRscList
IN/OUT V_NoOfRsc      INTEGER,
IN/OUT A_BIRscList    AT_BIRscList,
IN/OUT V_NoOfUnkRsc    INTEGER,
IN/OUT A_RTMCUnkRscList AT_RscList;
    
```



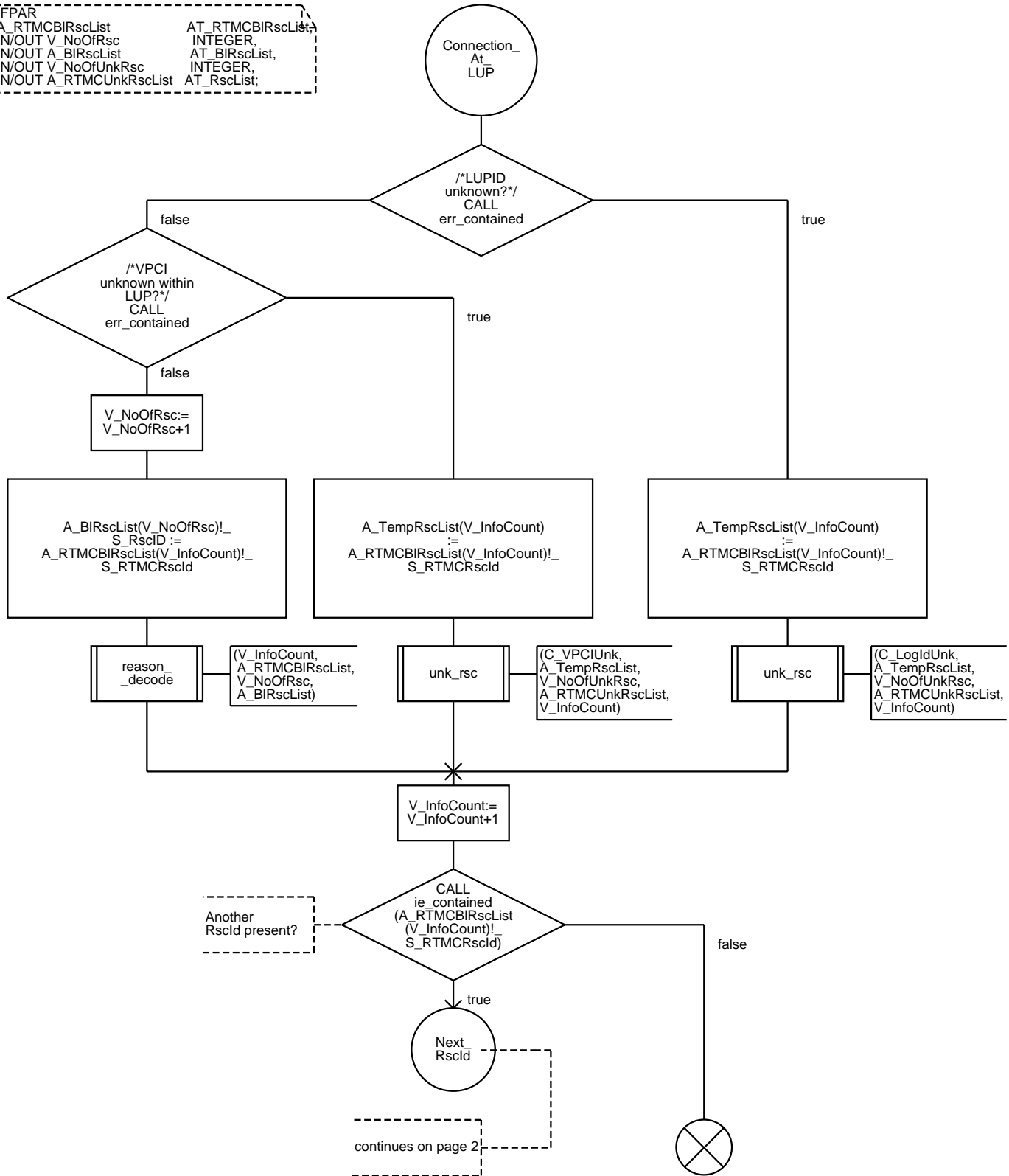
```

;FPAR
A_RTMCBIRscList  AT_RTMCBIRscList
IN/OUT V_NoOfRsc  INTEGER,
IN/OUT A_BIRscList AT_BIRscList,
IN/OUT V_NoOfUnkRsc INTEGER,
IN/OUT A_RTMUnkRscList AT_RscList;
    
```

LSPID known ?

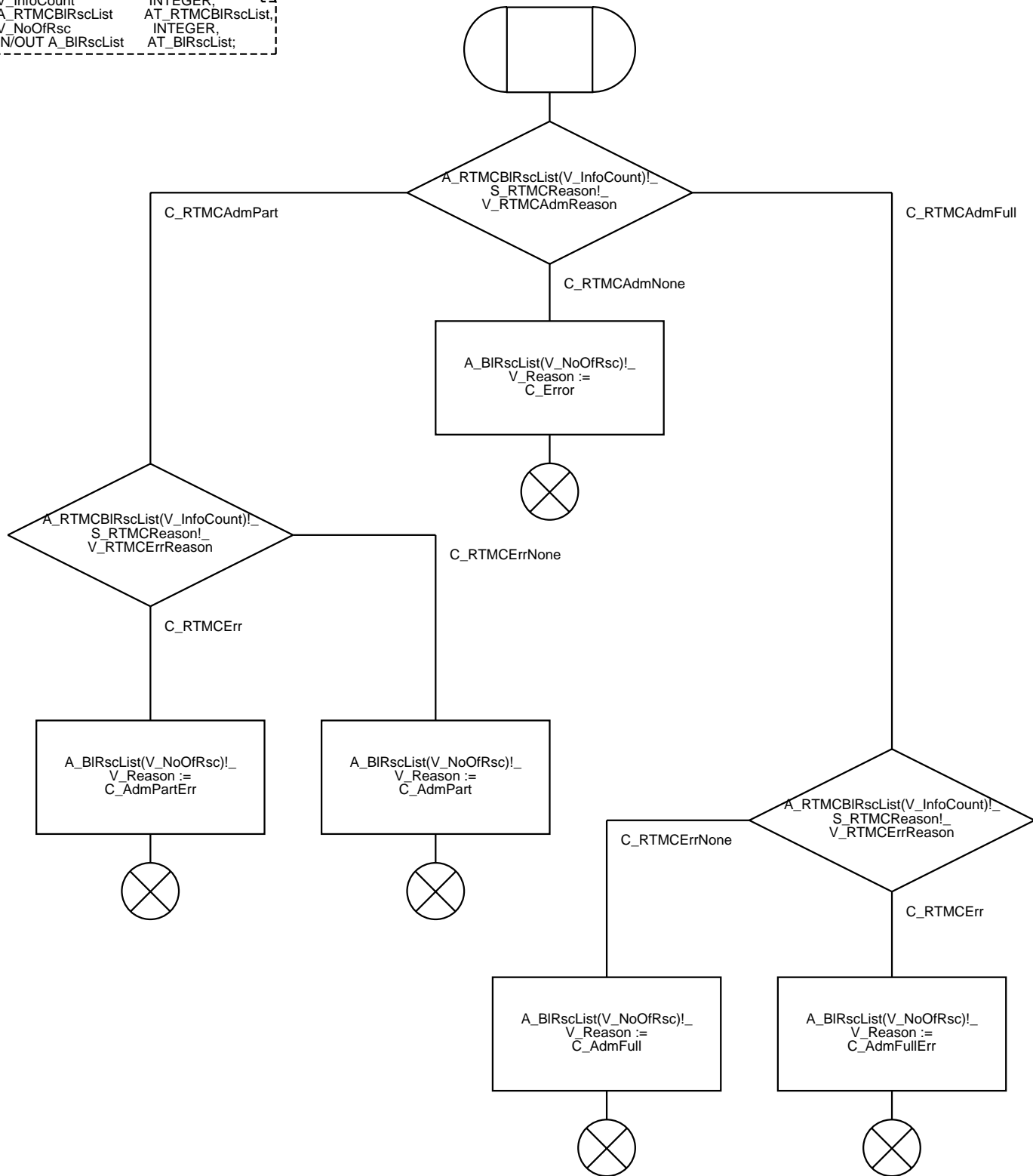


FPAR
 A_RTMCBIRscList AT_RTMCBIRscList
 IN/OUT V_NoOfRsc INTEGER
 IN/OUT A_BIRscList AT_BIRscList
 IN/OUT V_NoOfUnkRsc INTEGER
 IN/OUT A_RTMCUnkRscList AT_RscList;



```

;FPAR
V_InfoCount      INTEGER,
A_RTMCBIRscList  AT_RTMCBIRscList,
V_NoOfRsc        INTEGER,
IN/OUT A_BIRscList  AT_BIRscList;
    
```



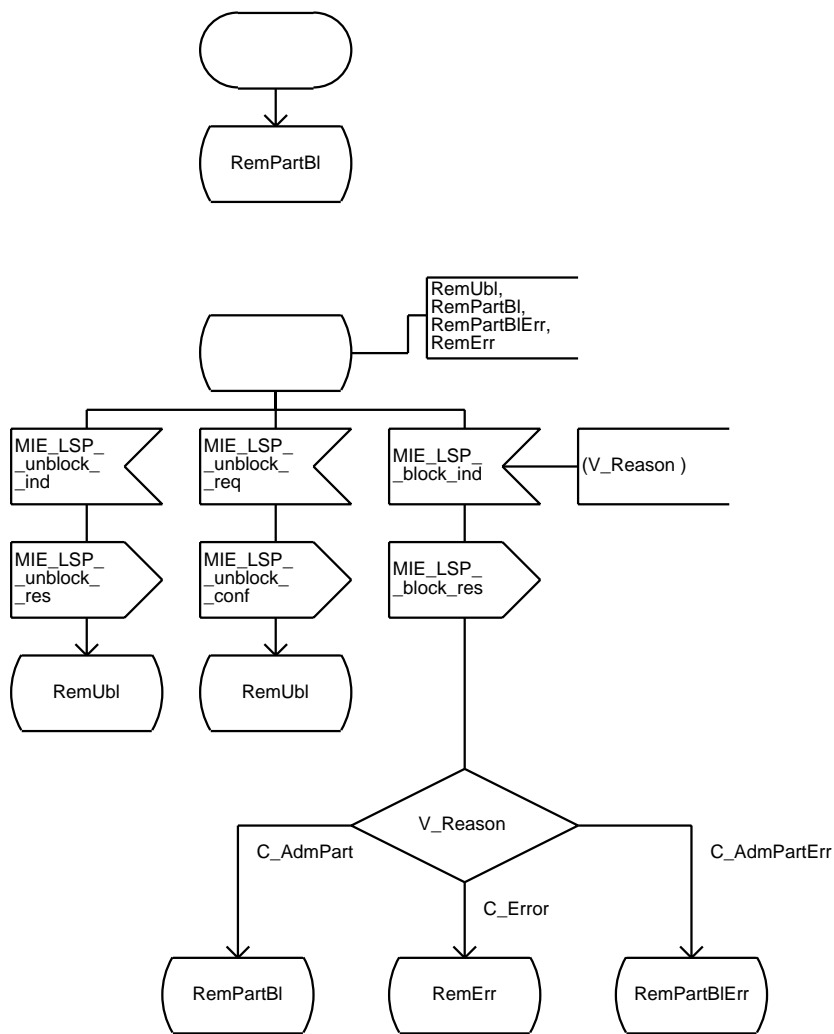
/* SN_LSPSTAT is a FSM which represents the remote LSP status */

/* 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*/



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

/* State descriptions

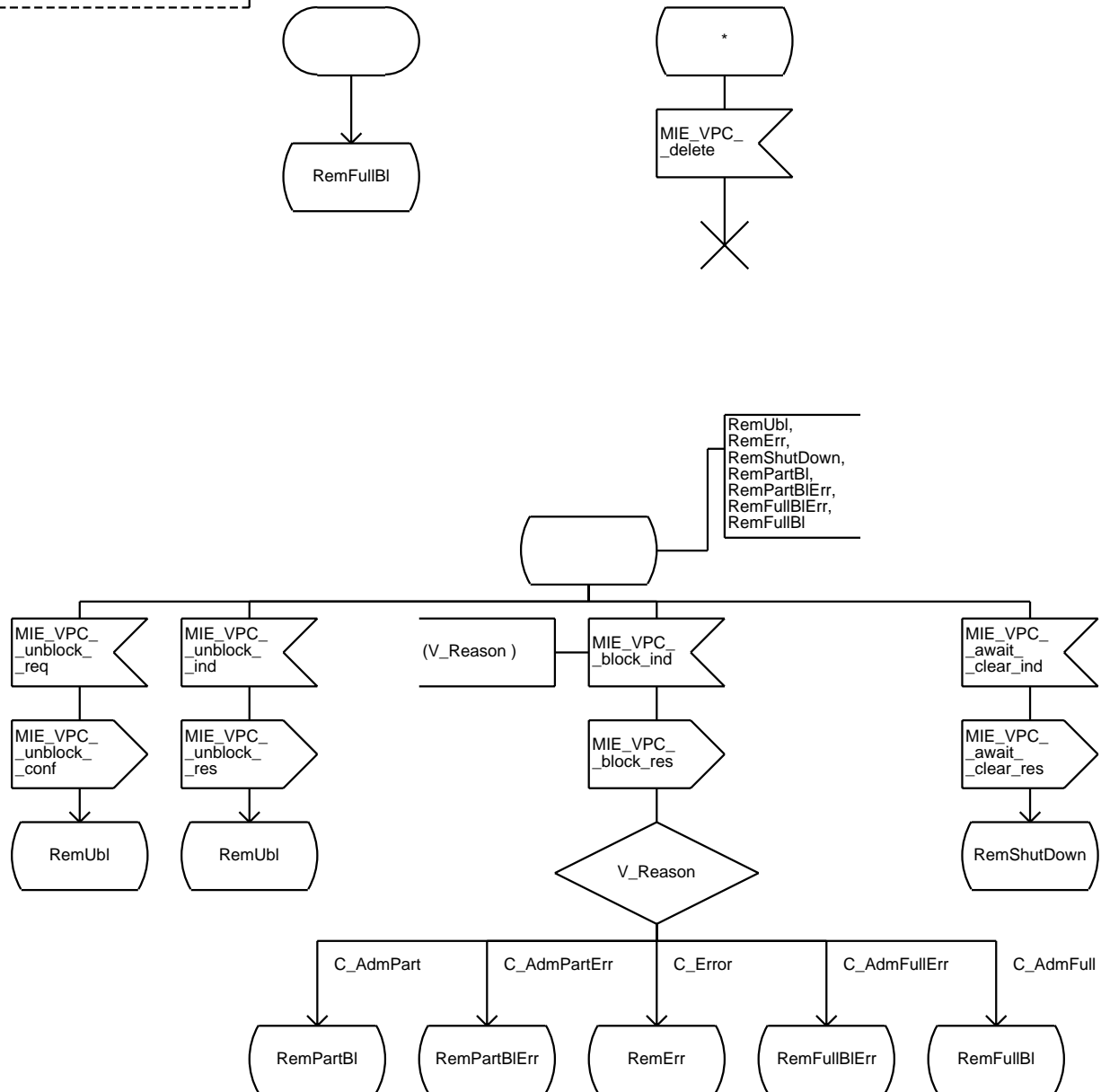
RemUbl: VPC available for Service
RemFullBl: VPC unavailable for service due to locking by AN operator
RemPartBl: VPC partially unavailable due to partial locking by AN operator
RemErr: VPC unavailable due to fault conditions
RemFullBlErr: VPC unavailable due to locking by AN operator AND fault conditions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

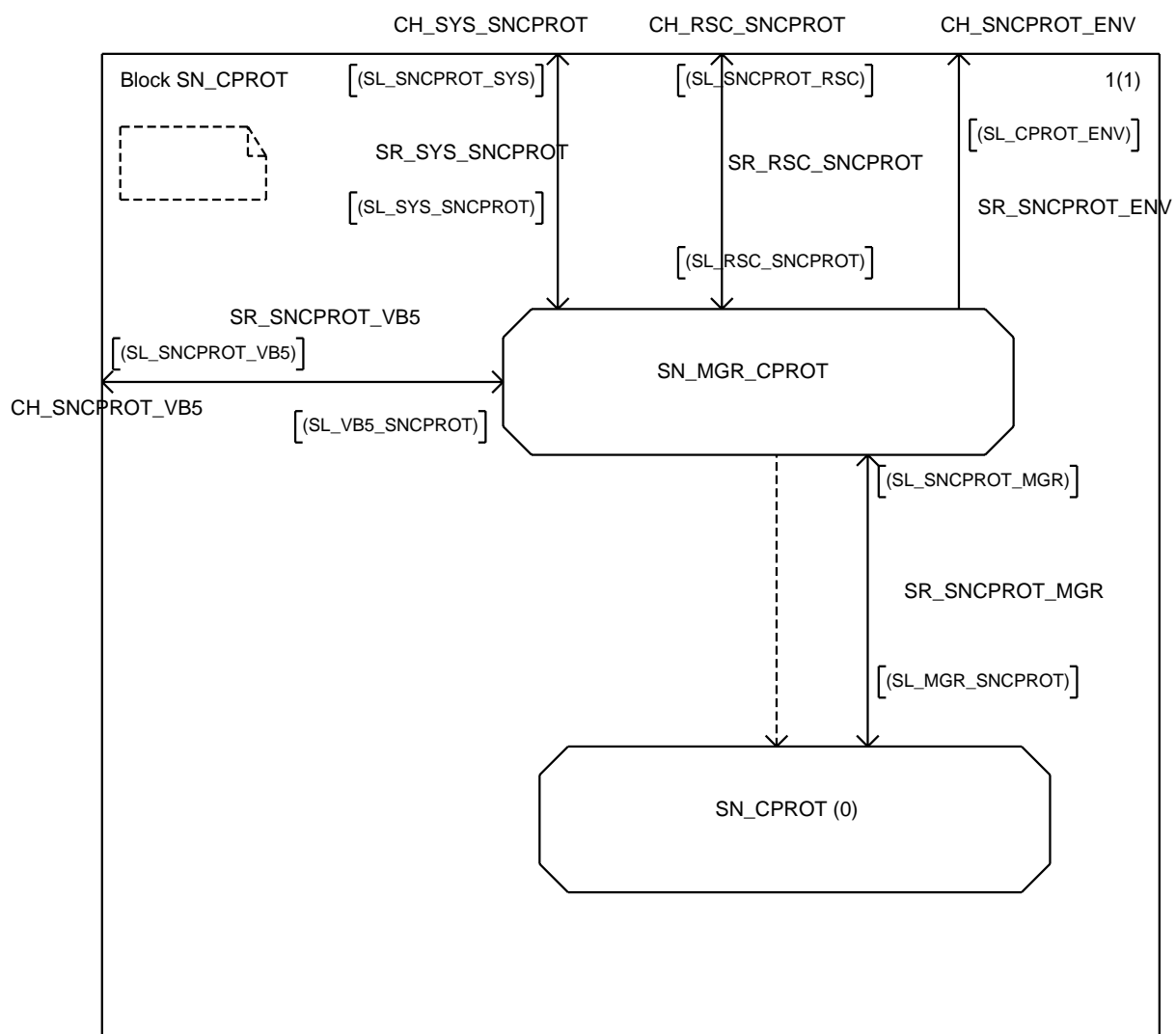
/*Signal data declarations*/

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.
*/

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





```

/*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;

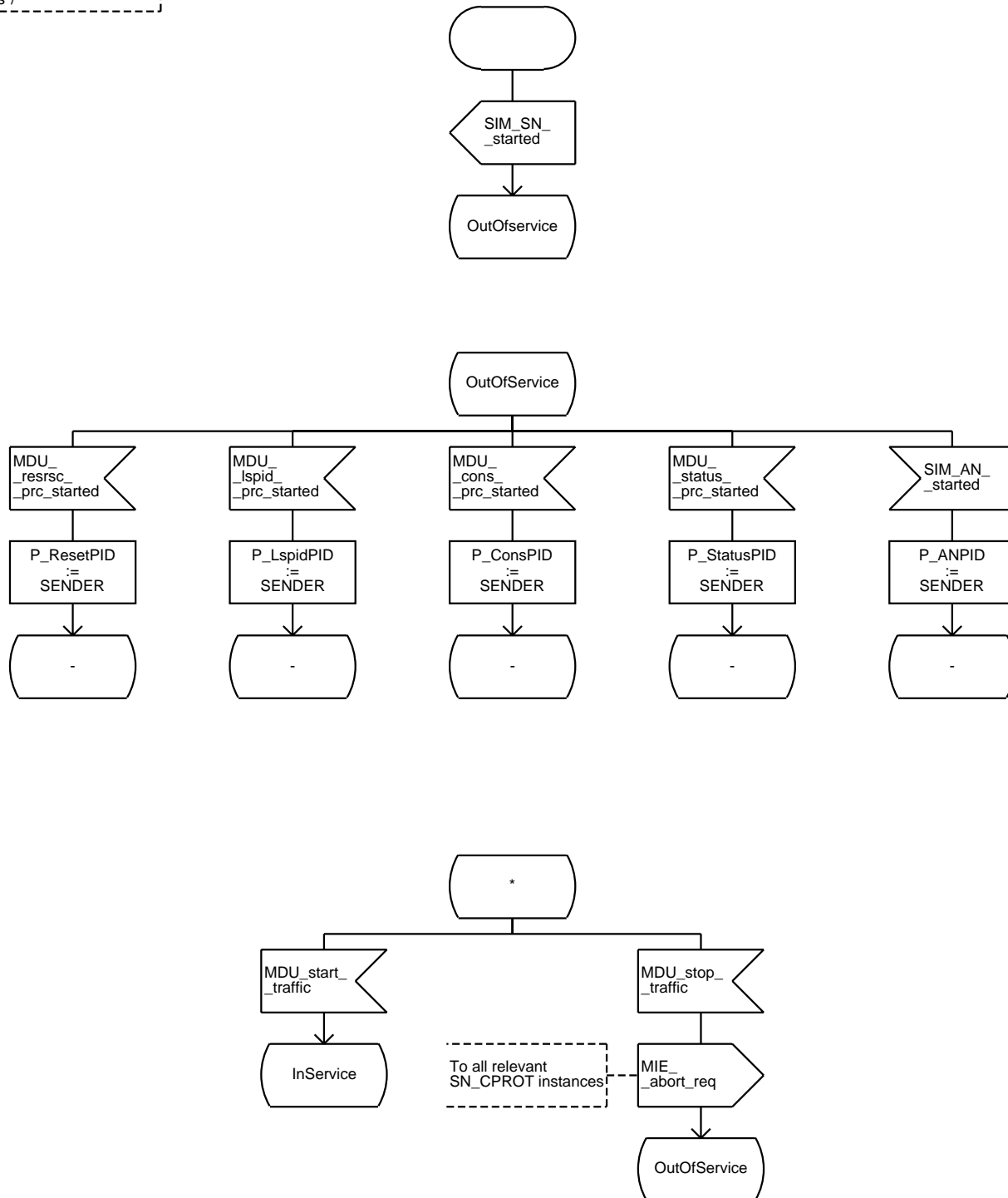
```

```

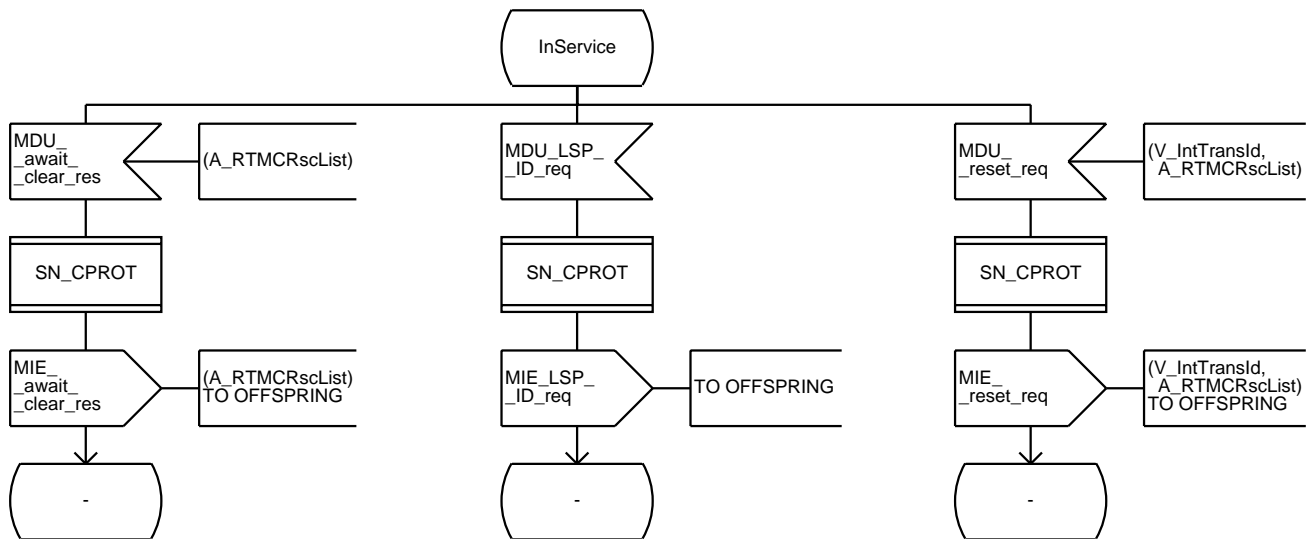
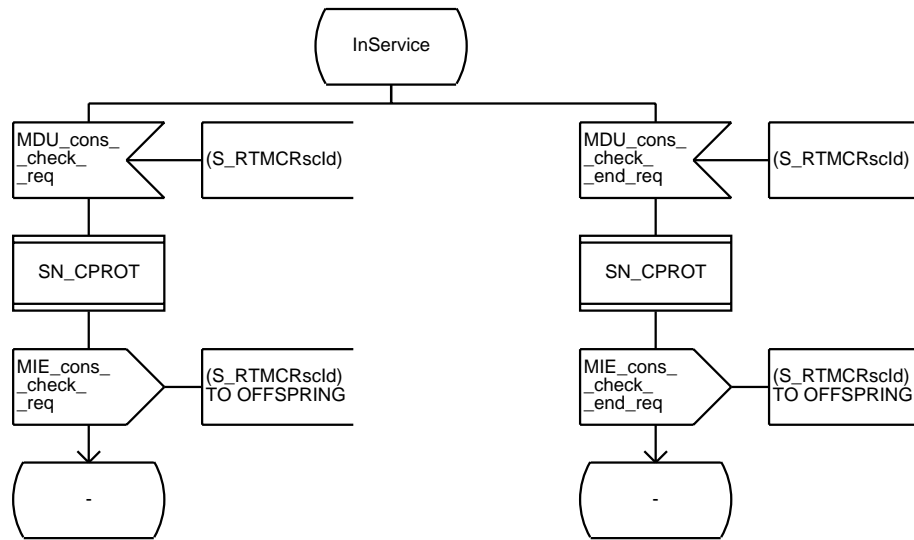
/*Primitive Data Declarations*/
DCL
A_RTMCBIRscList AT_RTMCBIRscList;
/**/
DCL
A_RTMCRscList AT_RscList;
/**/
DCL
A_RTMCUnkRscList AT_RscList;
/**/
DCL
S_RTMCRscld ST_Rscld;
/**/
DCL
S_RTMCUnkRscld ST_Rscld;
/**/
DCL
V_RTMCResult IT_RTMCResult;
/**/
DCL
V_IntTransId IT_IntTransId;
/**/
DCL
V_Result IT_Result;
/**/
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;

```

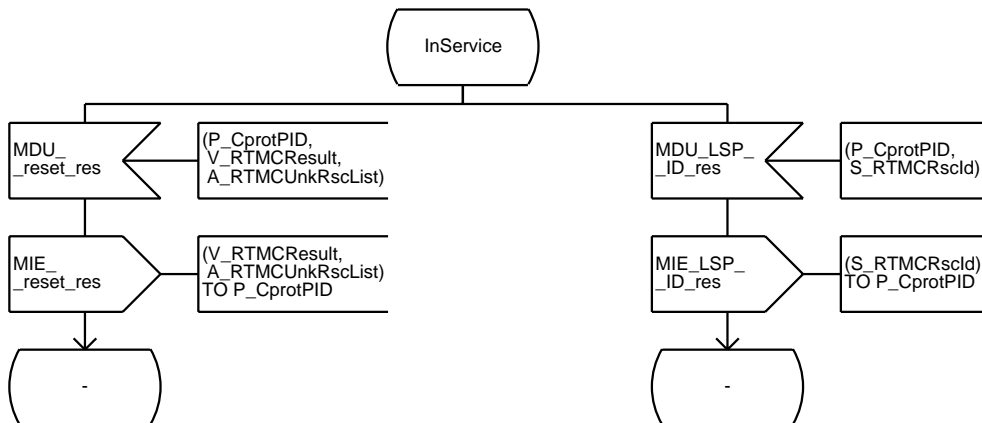
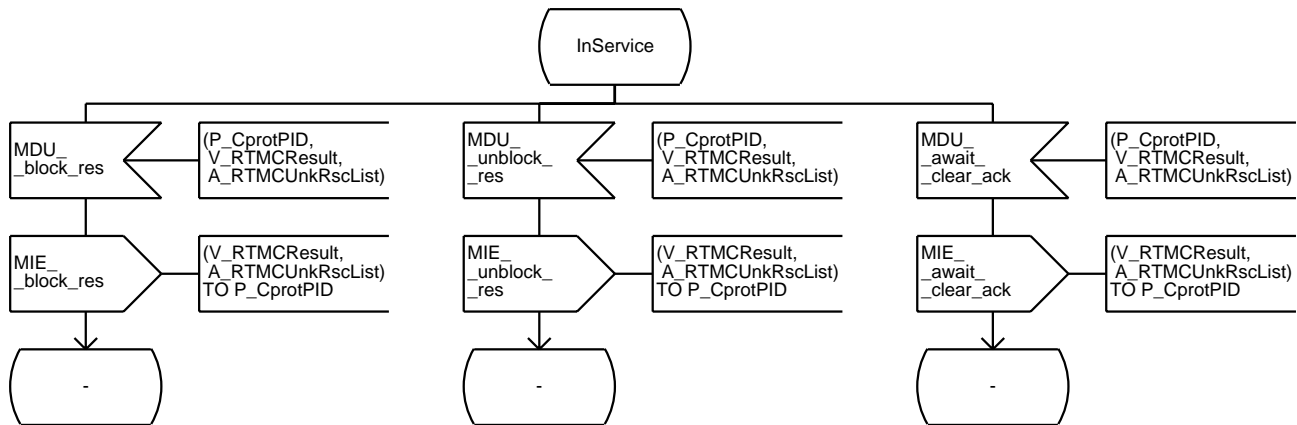
/*SN_CPROT Manager, controls
instantiation of SN_CPROT
processes*/



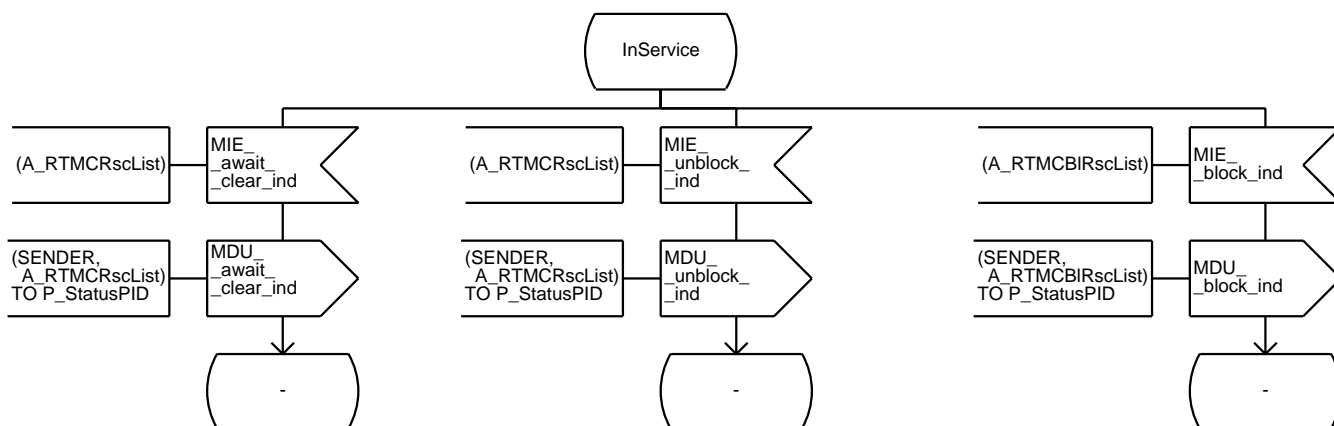
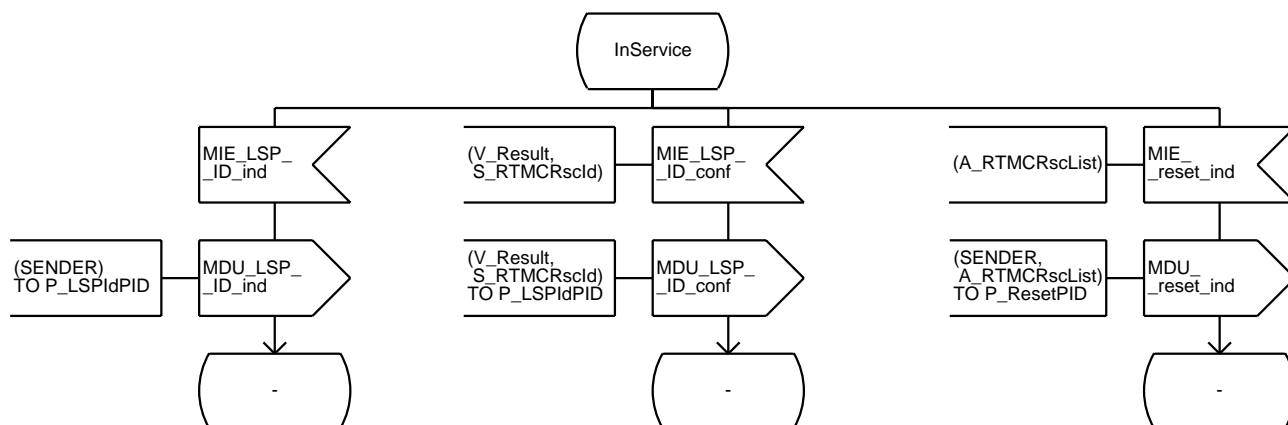
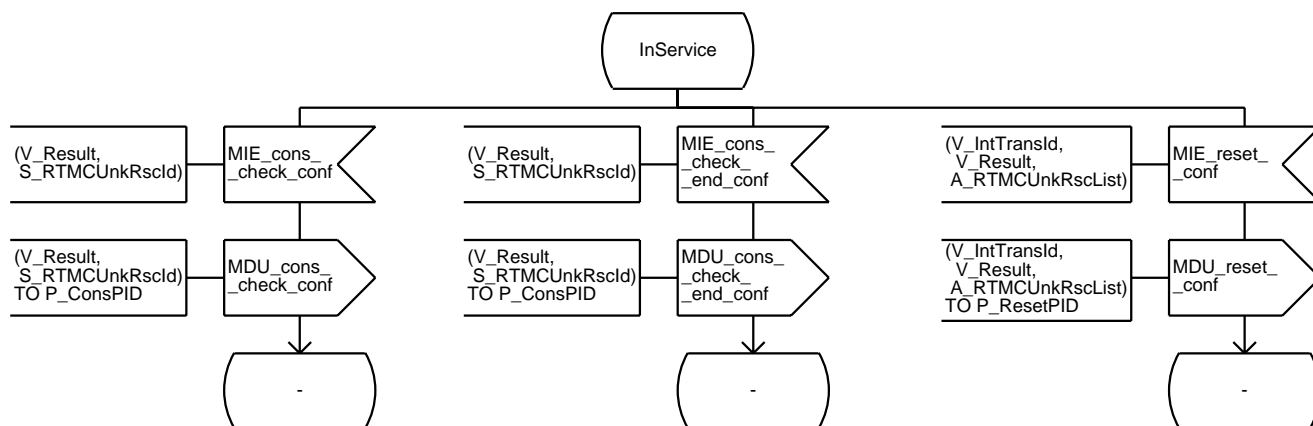
/*SN_CPROT Manager, controls
instantiation of SN_CPROT
processes*/



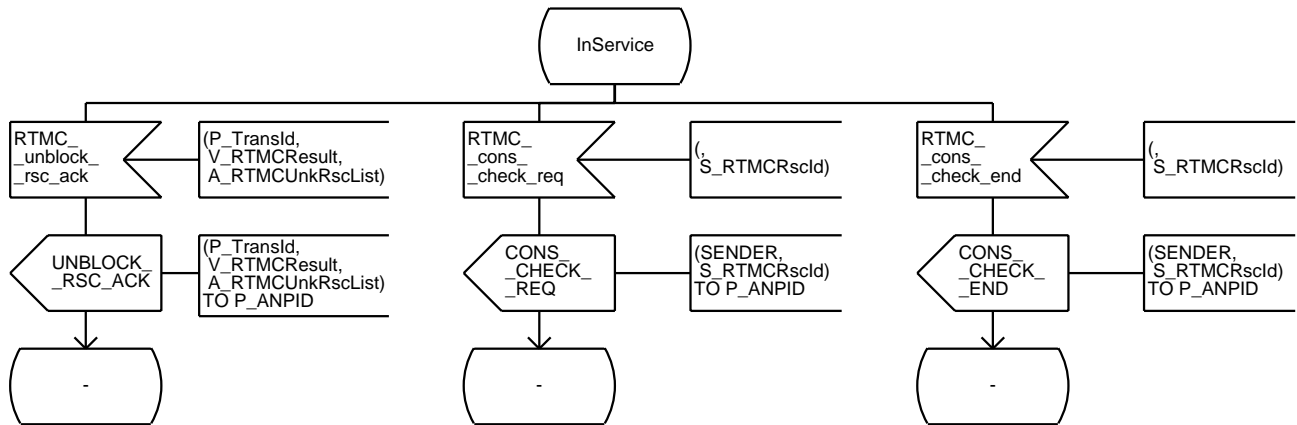
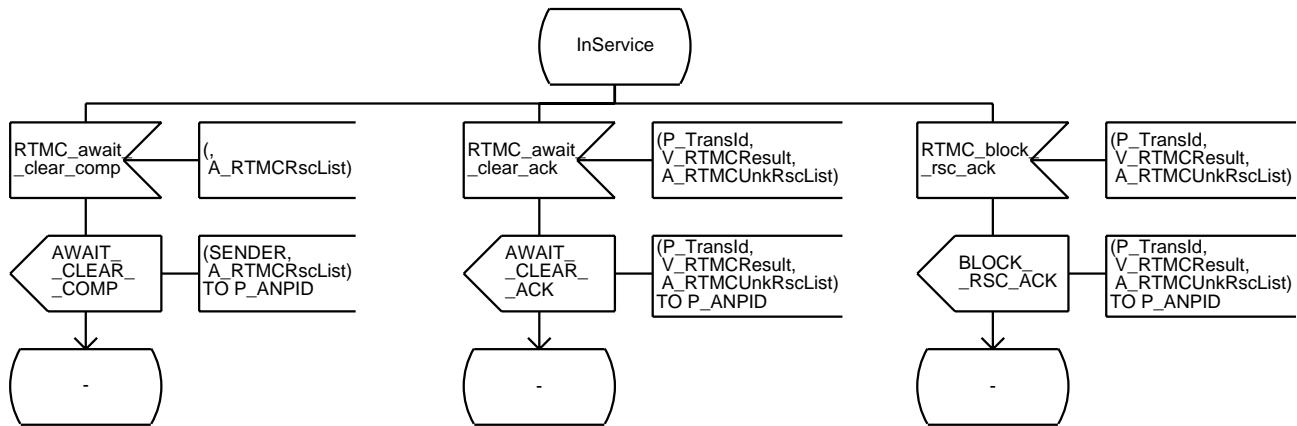
/*SN_CPROT Manager, controls
instantiation of SN_CPROT
processes*/



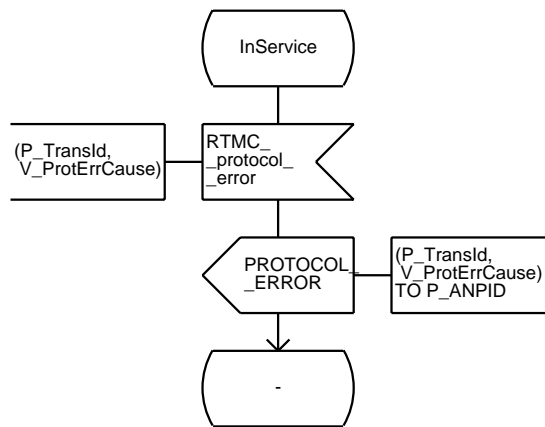
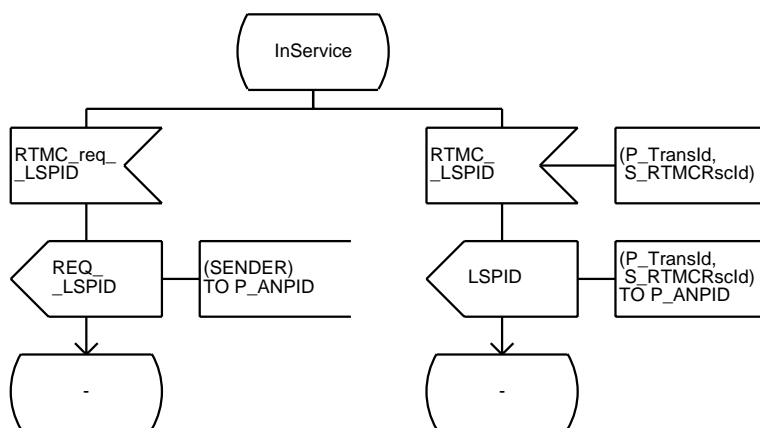
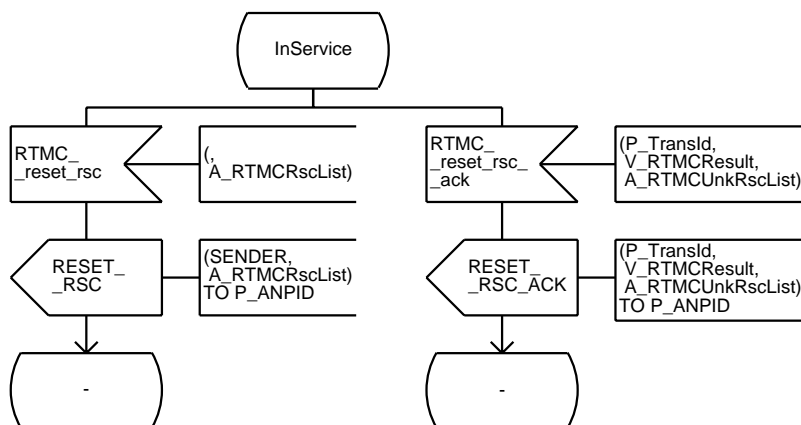
/*SN_CPROT Manager, controls
instantiation of SN_CPROT
processes*/



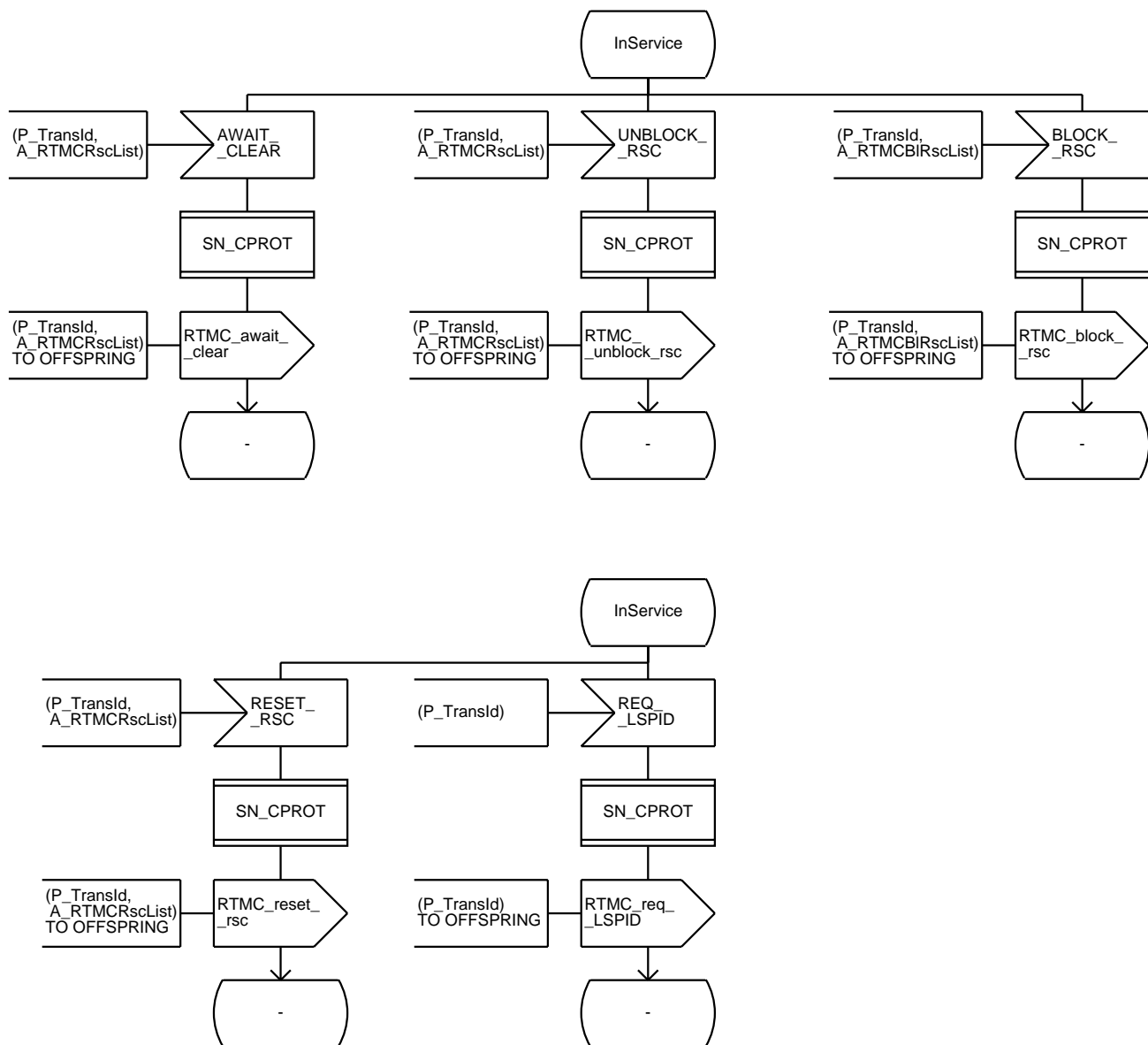
/*SN_CPROT Manager, controls
instantiation of SN_CPROT
processes*/



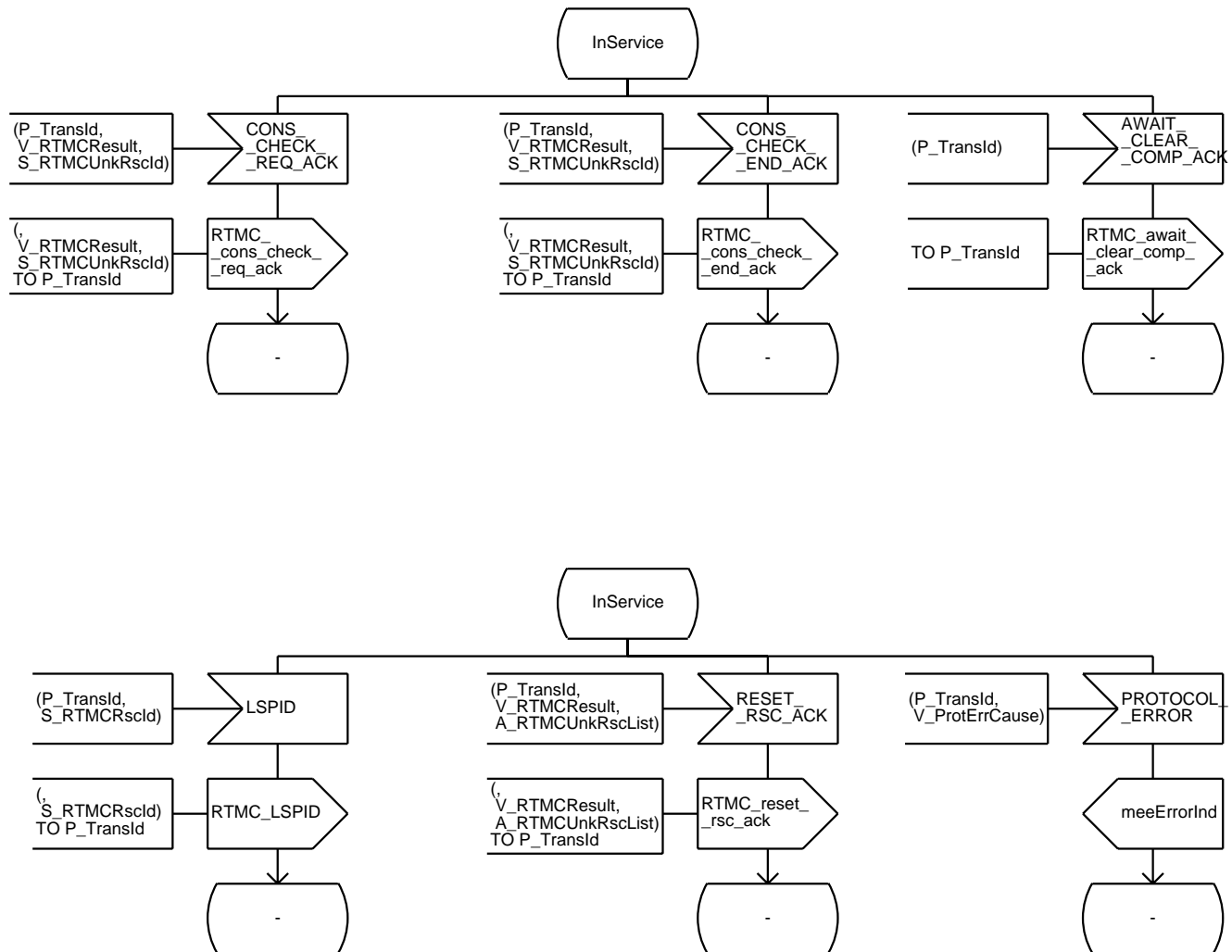
/*SN_CPROT Manager, controls
instantiation of SN_CPROT
processes*/



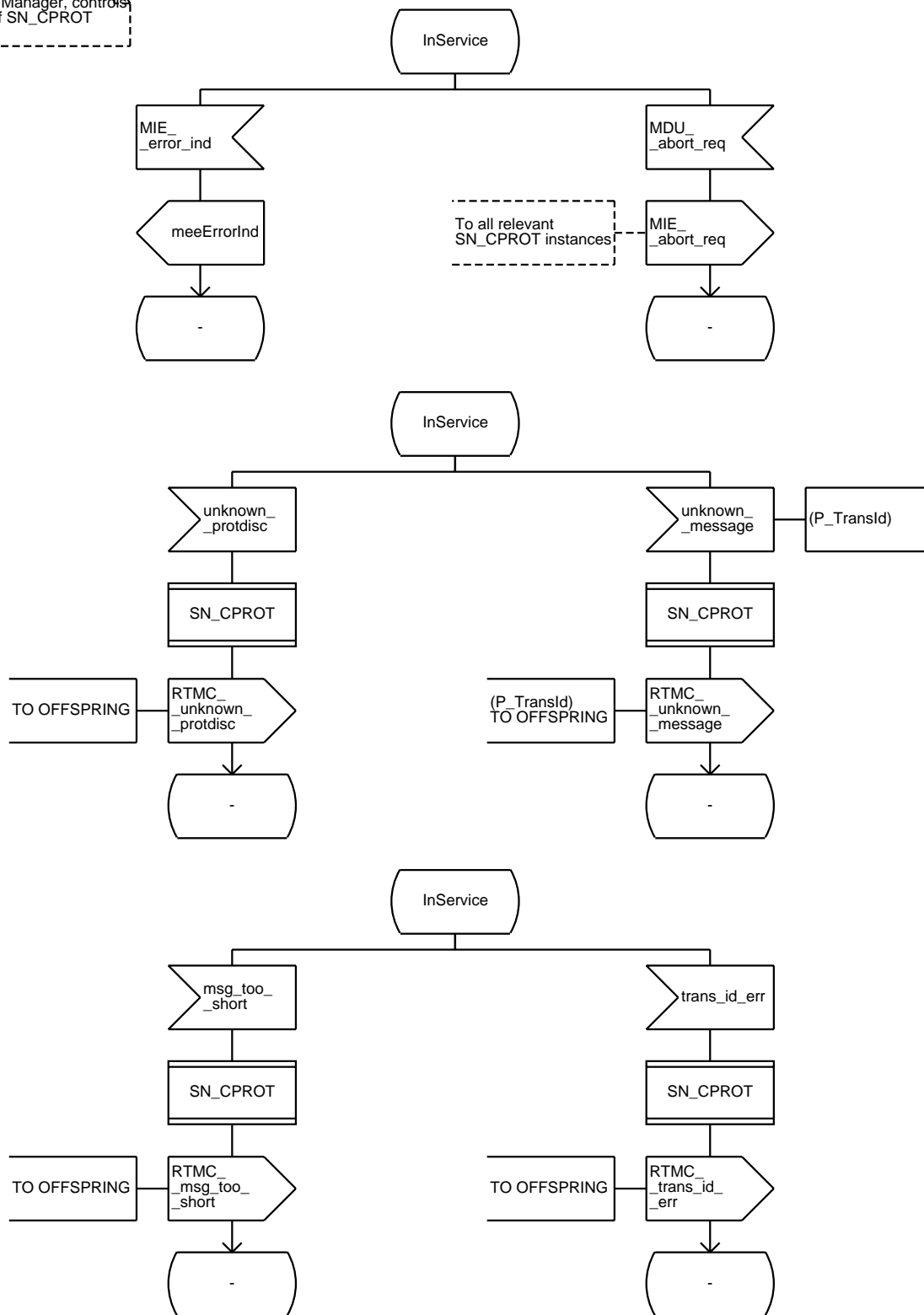
/*SN_CPROT Manager, controls
instantiation of SN_CPROT
processes*/



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



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



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

/* Definitions 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_consensd := 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*/

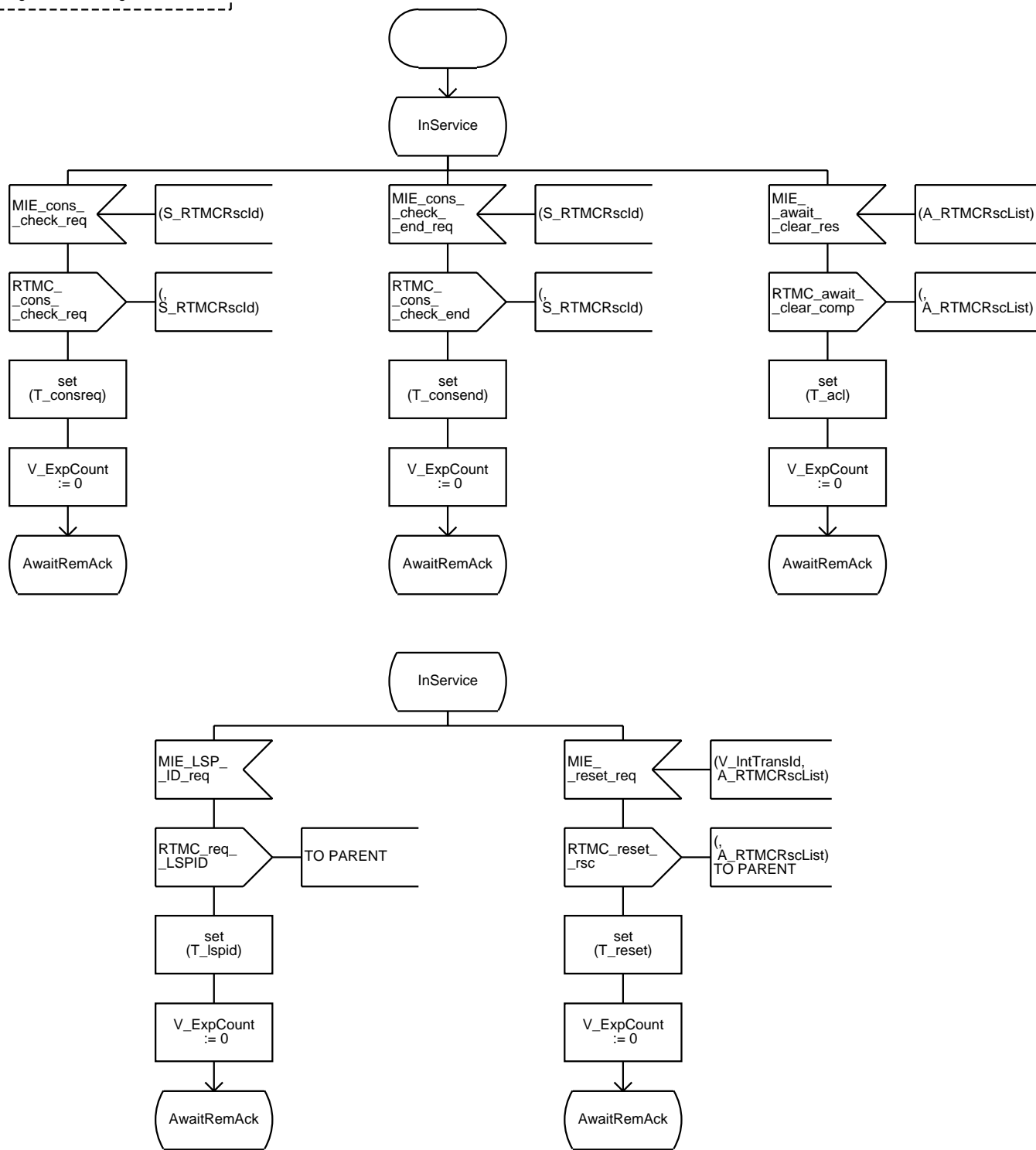
/**/
 /* timer handling */
 DCL V_ExpCount NATURAL; /*number of actual message repetitions*/
 /*possible values*/
 SYNONYM C_ExpMax INTEGER = 1; /*maximum number of message repetitions is 1*/
 /**/
 /*syntax check handling*/
 DCL V_SynResult INTEGER; /*syntax check result, provided by procedure IE_CHECK*/
 /*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_RTMCRsclId ST_RsclId; /*Resource Identifier Information Element*/
 /**/
 DCL
 S_RTMCUnkRsclId ST_RsclId; /*Resource Identifier Information Element unknown by the peer side*/
 /**/
 DCL
 A_RTMCRsclList AT_RsclList; /*Array of Resource Identifier Information Elements*/
 /**/
 DCL
 A_RTMCBIRsclList AT_RTMCBIRsclList; /*Array of Blocked Resource Identifier Information Elements*/
 /**/
 DCL
 A_RTMCUnkRsclList AT_RsclList; /*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*/

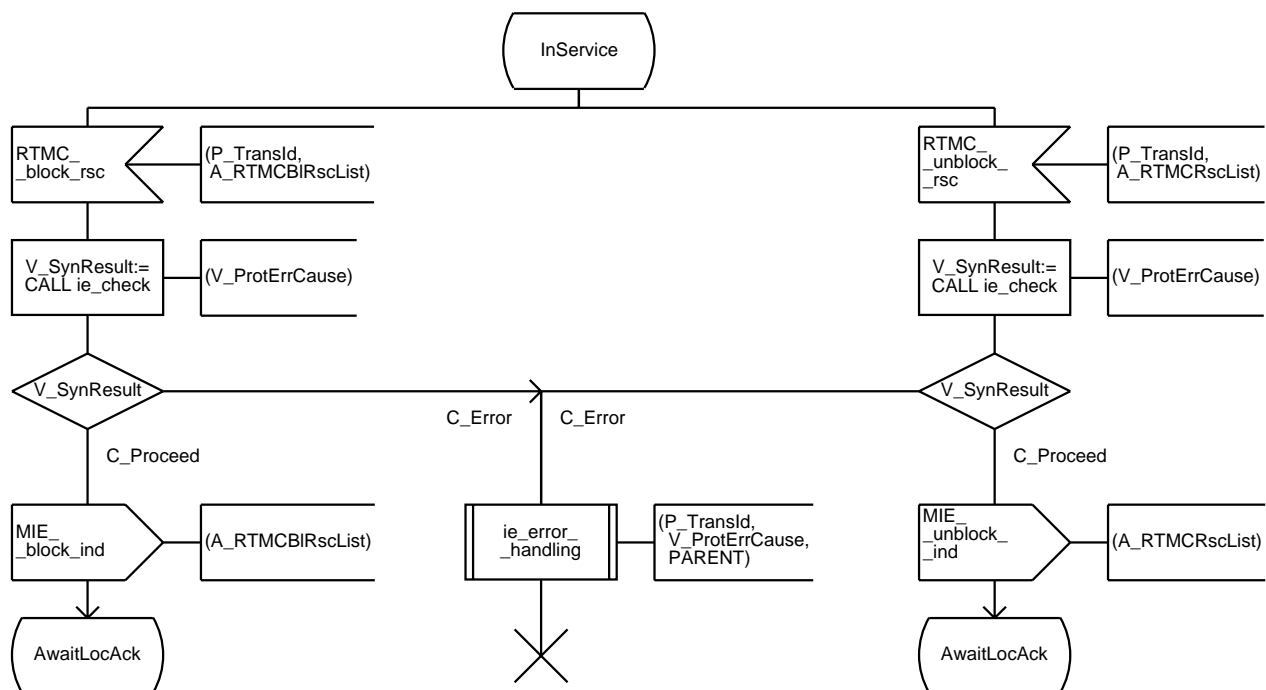
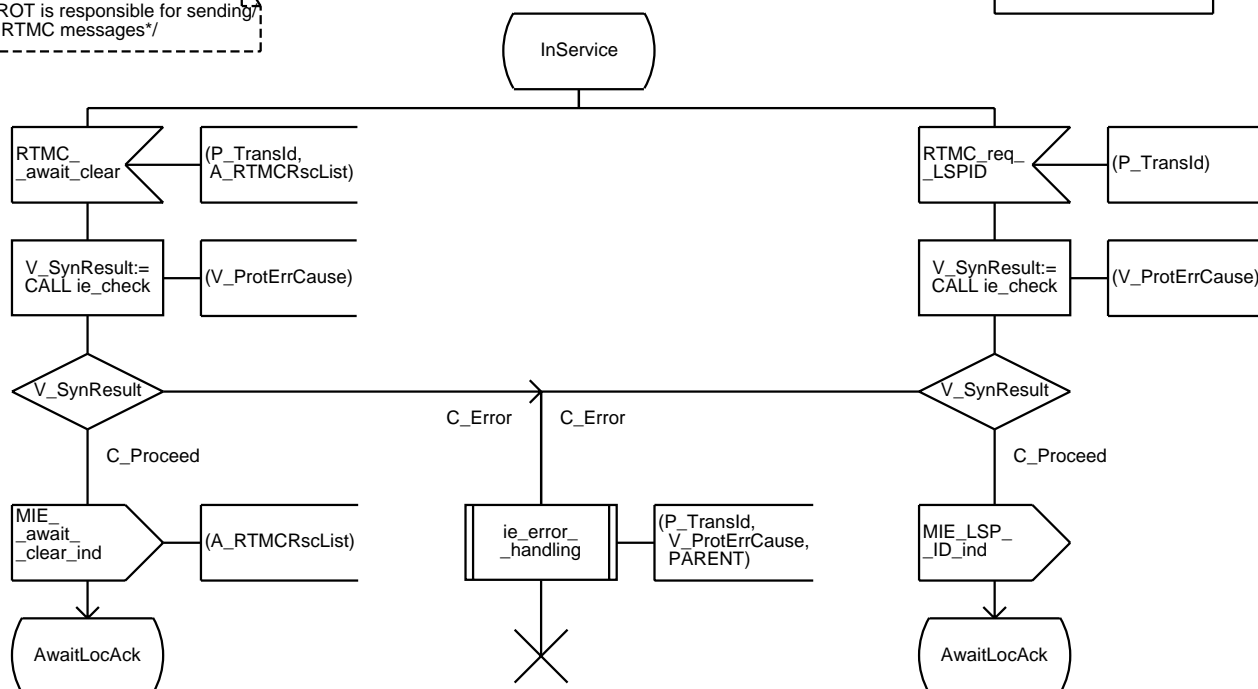
/*SN_CPROT is responsible for sending
receiving RTMC messages*/

/*Handling of
Requests from
SN System Management */



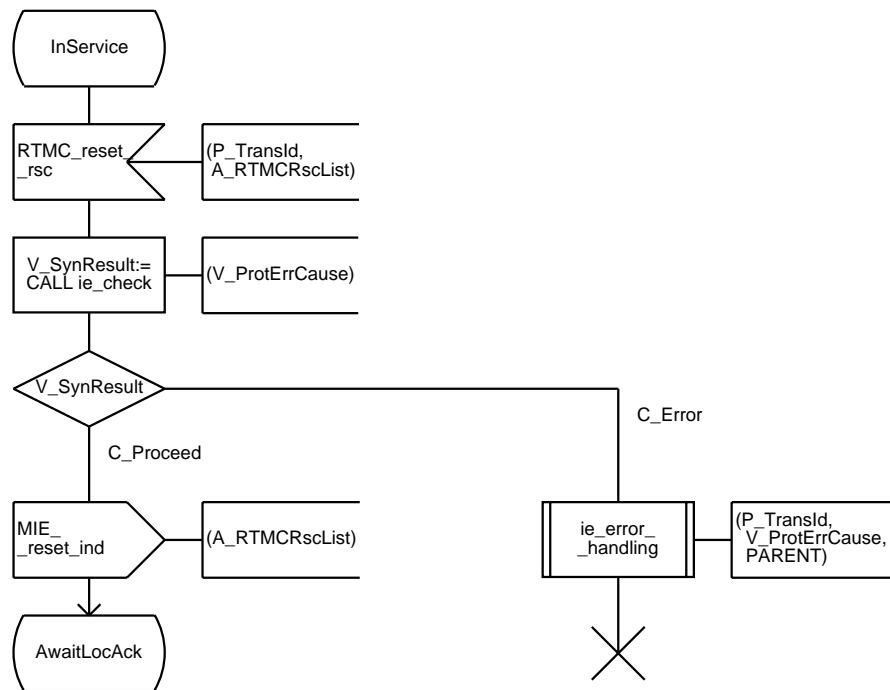
/*SN_CPROT is responsible for sending
receiving RTMC messages*/

/*Handling of
VB5 Messages from AN*/



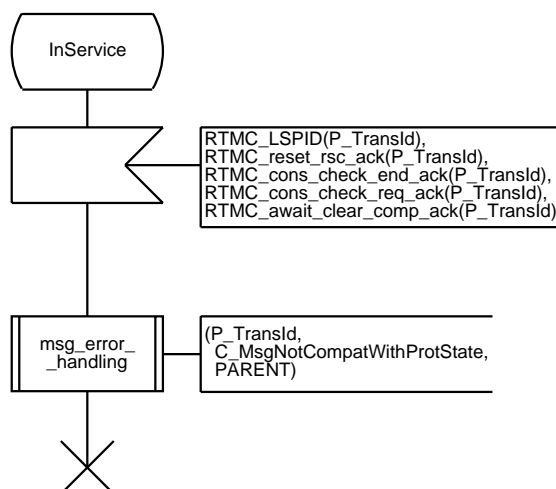
/*SN_CPROT is responsible for sending,
receiving RTMC messages*/

/*Handling of
VB5 Messages from AN*/



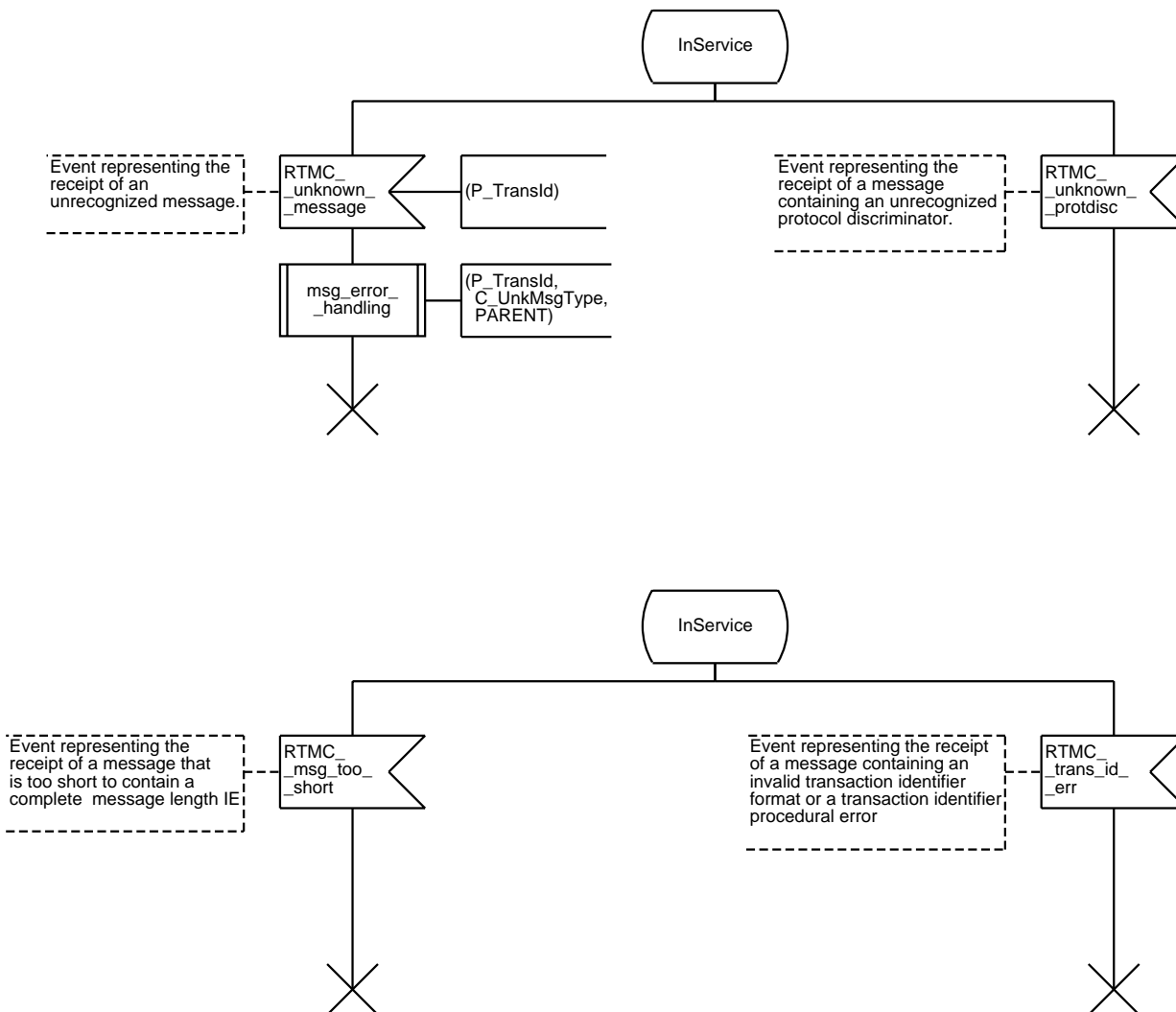
/*SN_CPROT is responsible for sending
receiving RTMC messages*/

/* Handling of
unexpected events */



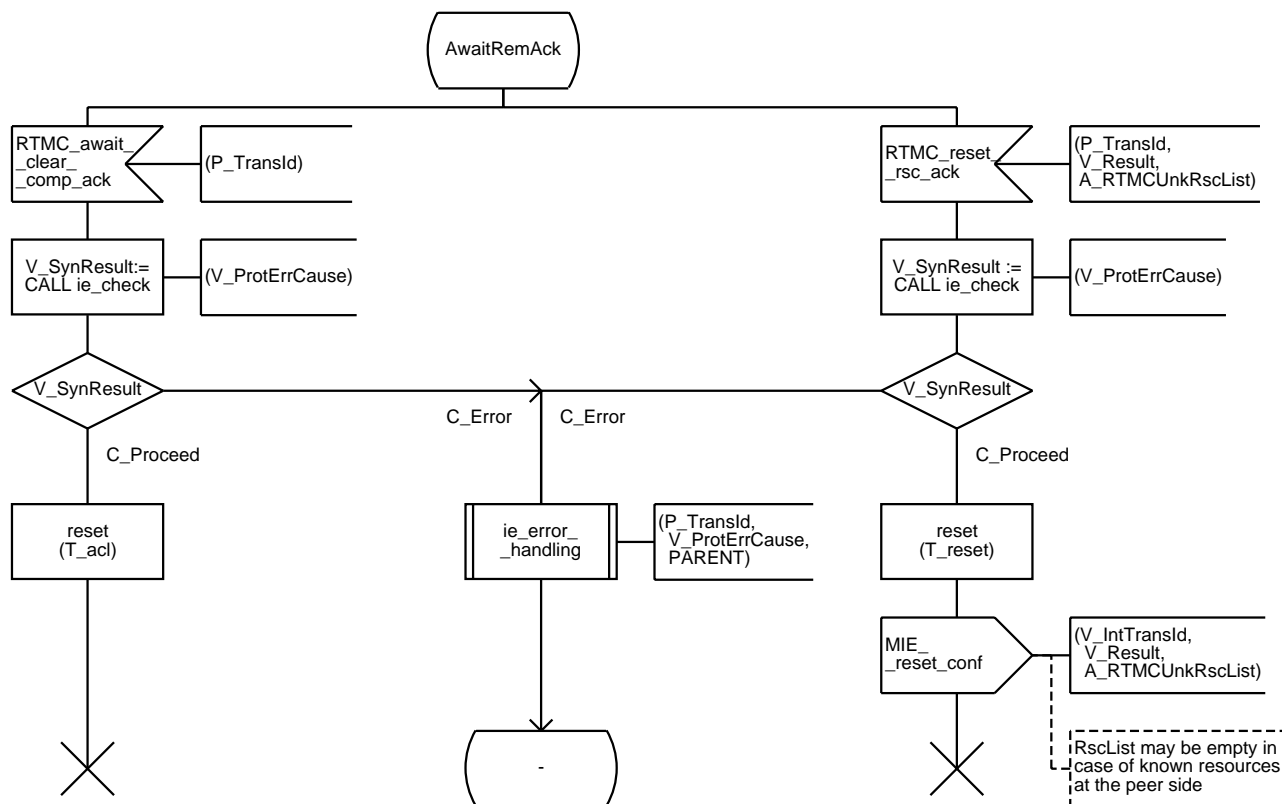
/*SN_CPROT is responsible for sending
receiving RTMC messages*/

/* Handling of
pseudo events for
error handling*/



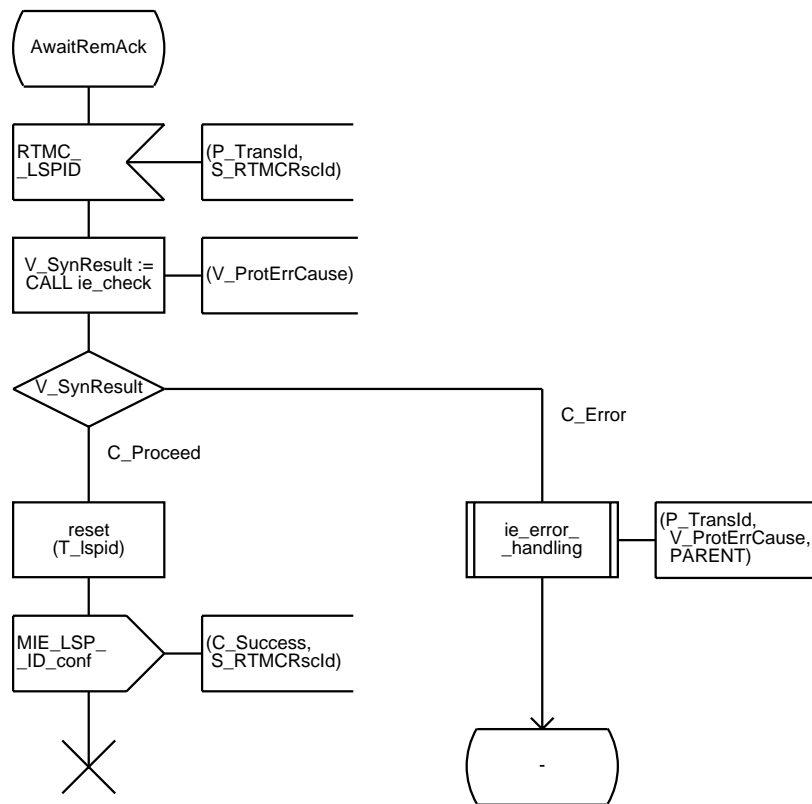
/*SN_CPROT is responsible for sending
receiving RTMC messages*/

/*Handling of AN
Acknowledgements */



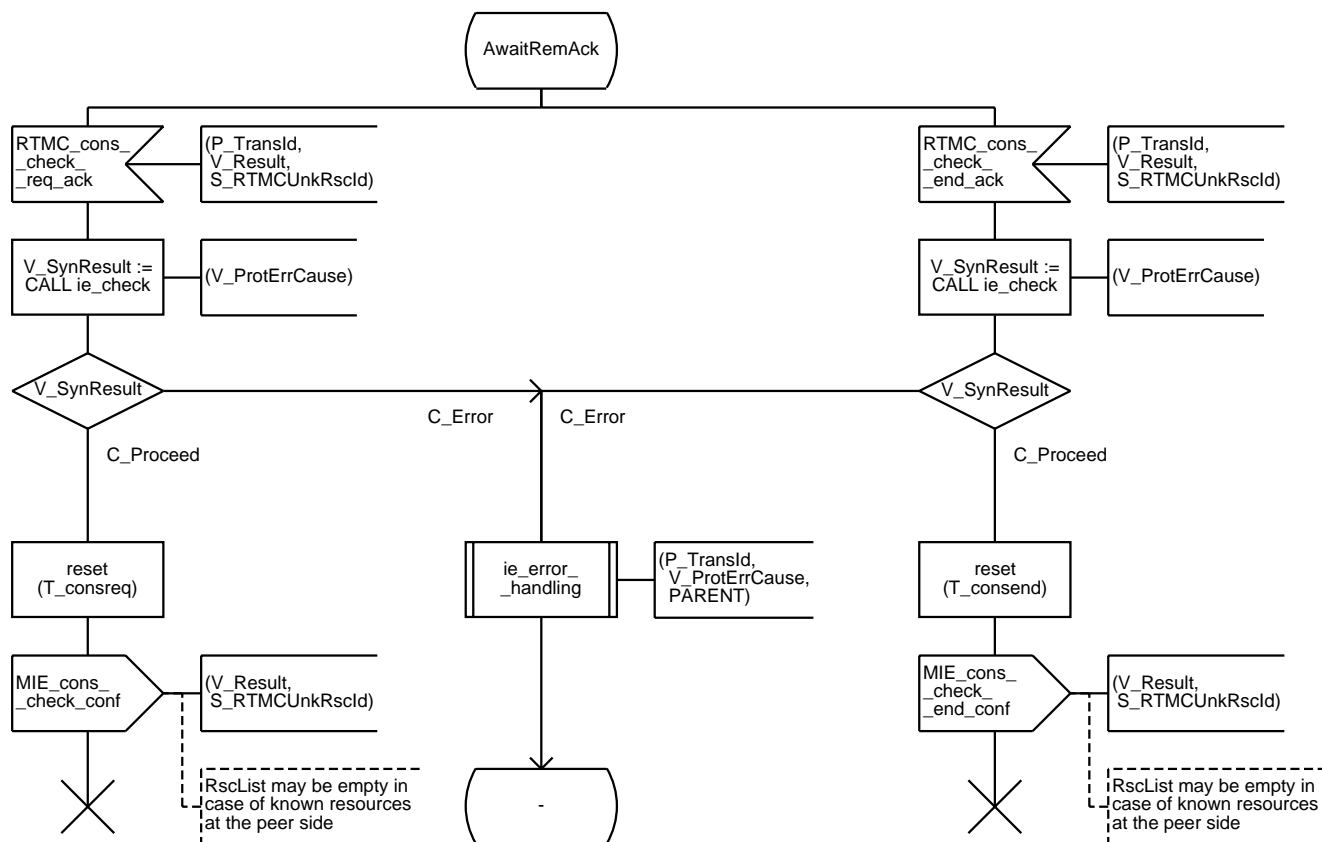
/*SN_CPROT is responsible for sending,
receiving RTMC messages*/

/*Handling of AN
Acknowledgements */



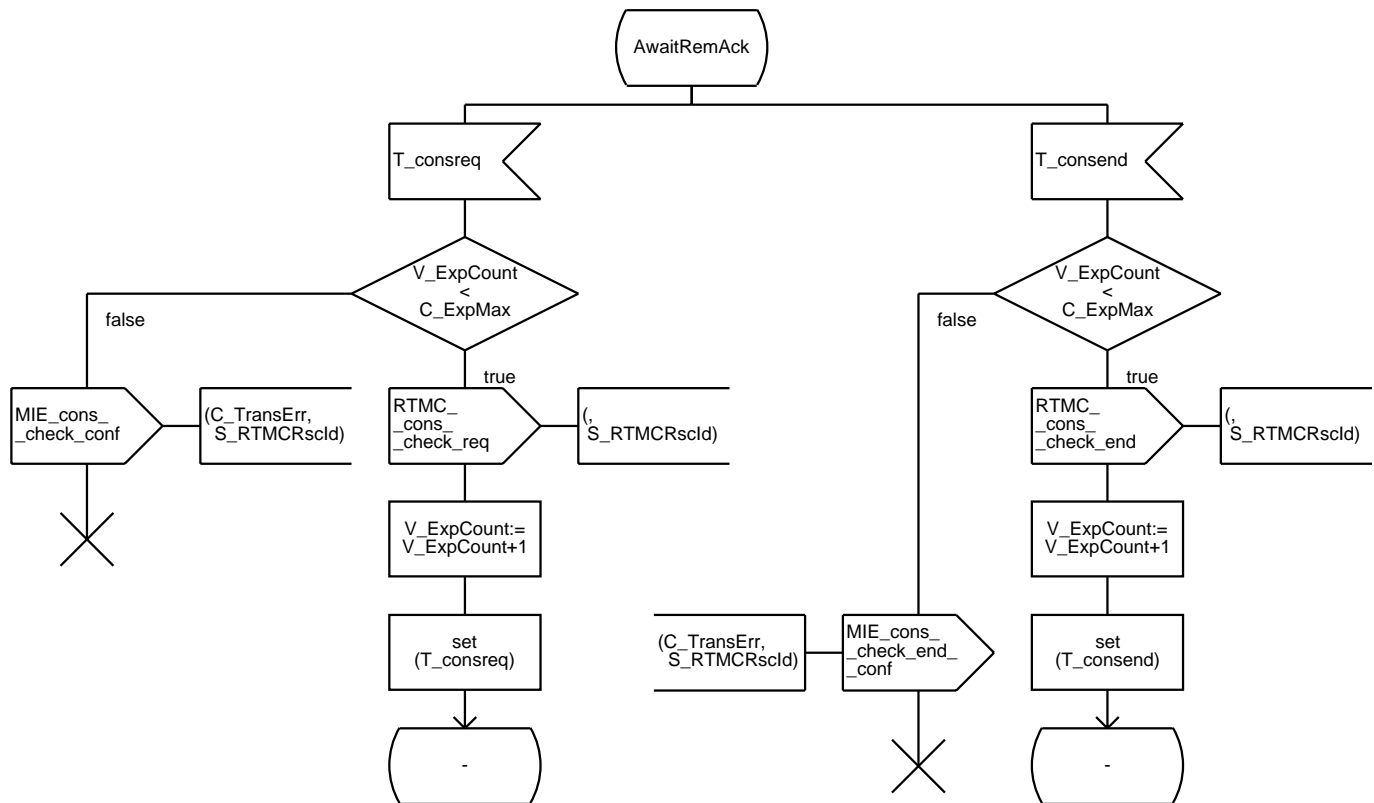
/*SN_CPROT is responsible for sending
receiving RTMC messages*/

/*Handling of AN
Acknowledgements*/



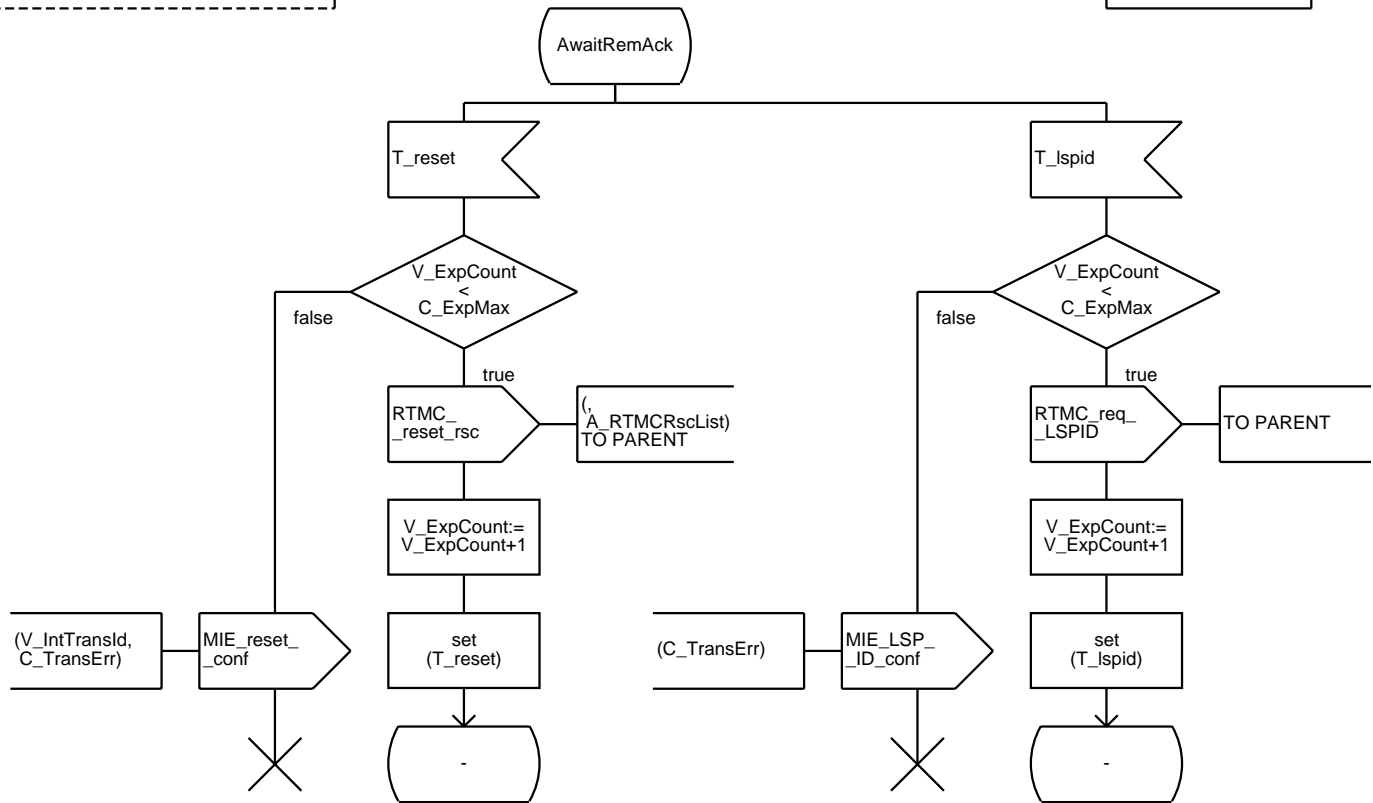
/*SN_CPROT is responsible for sending
receiving RTMC messages*/

/* Timer expirations */



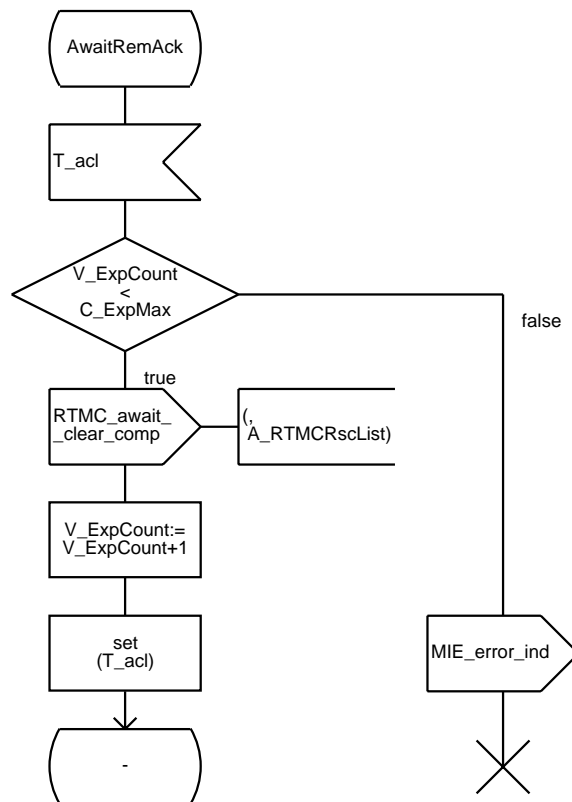
/*SN_CPROT is responsible for sending
receiving RTMC messages*/

/* Timer expirations */



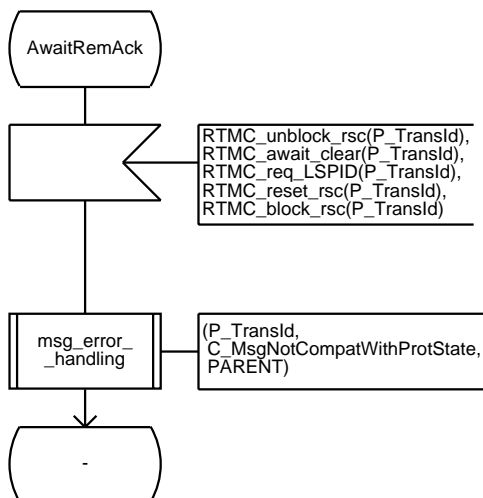
/*SN_CPROT is responsible for sending
receiving RTMC messages*/

/* Timer expirations */



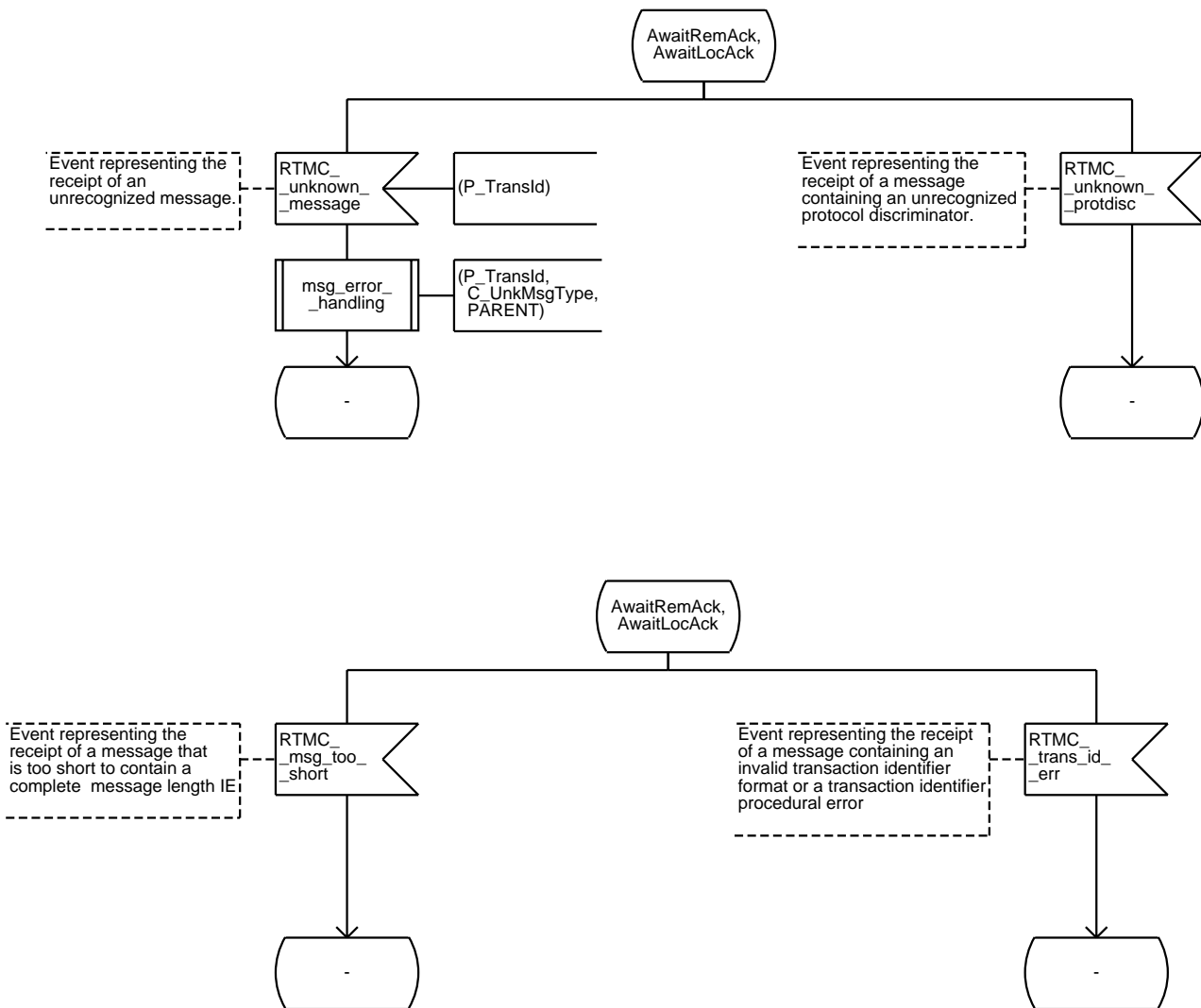
/*SN_CPROT is responsible for sending
receiving RTMC messages*/

/* Handling of
unexpected events */



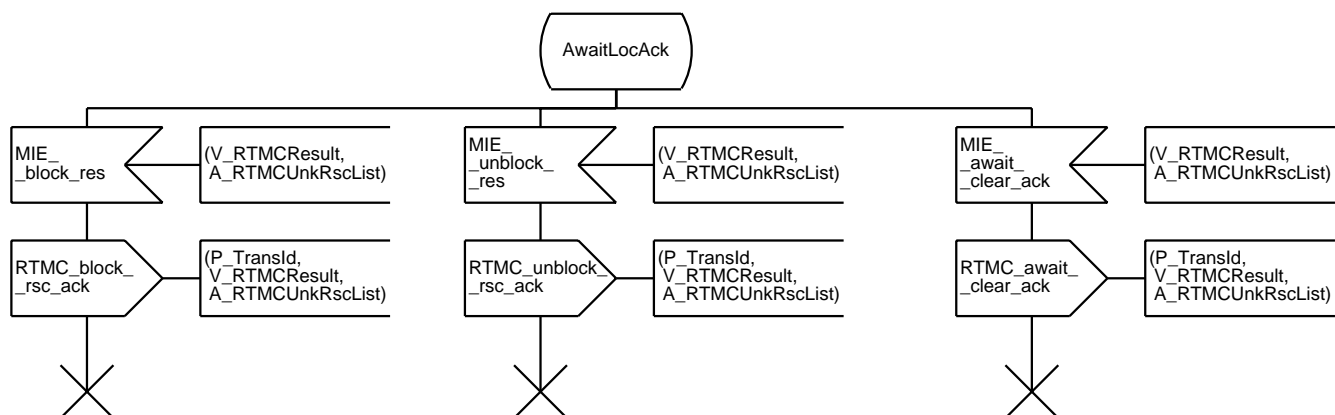
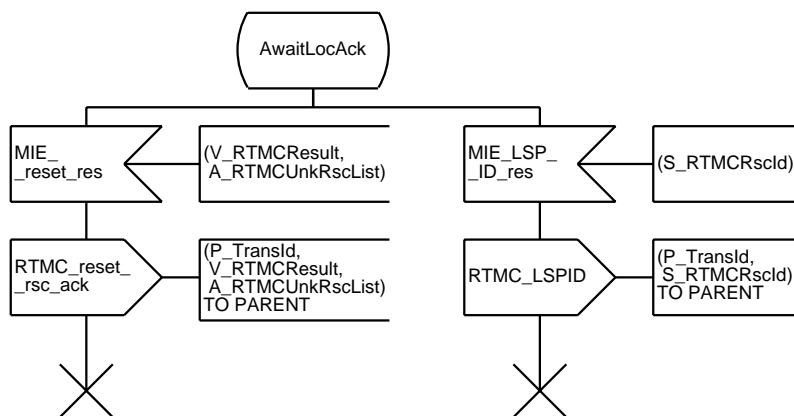
/*SN_CPROT is responsible for sending
receiving RTMC messages*/

/* Handling of
pseudo events for
error handling*/



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

/*Handling of system
management acknowledgements */



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

/* Handling of
unexpected events */

