- wand
  - Constant_declarations
    - PR Declaration
  - general type_declarations
    - PR Declaration
  - type_declarations
    - PR Declaration
  - General PR declaration
    - PR Declaration
    - PR Declaration
  - PR declarations AP
    - PR Declaration
    - PR Declaration
    - PR Declaration
  - DefaultPartition
    - PR Declaration
    - PR Declaration
    - PR Declaration
    - PR Declaration
    - AP_Steady_State_Ctr
      - PR Declaration
      - PR Declaration
      - PR Declaration
      - ASG
        - PR Declaration
      - AMAIA
        - DefaultPartition
          - PR Declaration
        - Steady
          - PR Declaration
          - PR Declaration
        - Steady
          - IAA_Process
            - PR Declaration
      - ABA
        - PR Declaration
      - ATI
        - PR Declaration
      - ALS
        - PR Declaration

system wand

```
/* CONSTANTS */

/*SYNONYM Max_DLX_Buffer = 1000;*/  /* Size of DLX buffer */
/* SYNONYM TT_size = 1000; */

/*SYNONYM Windowsize =16;*/ /* See Martin's reflexion : no more than
                         15 cells can be sent without acknowledge */
SYNONYM MT_MAX_actif NATURAL = 5;       /* 256 (theoretical value) */
SYNONYM BROADCAST_MAC_ADDR NATURAL = 1;
/*SYNONYM CONTROL_MAC_ADDR NATURAL = 0;*/
/*SYNONYM NULL_MAC_ADDR NATURAL = 6;*/



/*SYNONYM CELL_DURATION = 1;*/ /* In real WAND spec : 21,6. 10^-6 s */

/* ----------- channels -------------------- */
/*SYNONYM VC_max = 10;*/ /* Number max of Virtual Channels associated
                       to each MT,  Theoretical equal to 256. */
/*SYNONYM Control_MVC_Id MVC_id_type = 0;*/
/*SYNONYM M_control MVC_Id_type = 1;*/
/*SYNONYM UNI_Control MVC_Id_type = 2;*/
/*SYNONYM Broadcast_MVC_Id MVC_Id_type = VC_max-1;*/
/*SYNONYM UP Direction=0;*/
/*SYNONYM DOWN Direction = 1;*/
/*SYNONYM the_END Direction = -1;*/
/*SYNONYM MAX_AP_index = 5;*/ /* Number max of AP infos stored in MTC */
SYNONYM MAX_TIME_PERIODS = 10;
SYNONYM IAA_MAX = 3;
SYNONYM STEP = 3;
```

system wand

```
/* general type declarations */
/* ------------------------- */

/*NEWTYPE return_type
  LITERALS Success, Connec_Refused, No_resp, Not_associated,
          Already_associated, Timedout, Failed, Hunt_success,
        channel_exhausted;
ENDNEWTYPE return_type;*/

/*NEWTYPE susp_type
   LITERALS later;
ENDNEWTYPE;*/

NEWTYPE tip_type
 LITERALS start_tip, end_tip;
ENDNEWTYPE tip_type;

/*NEWTYPE MPDU_sig_type
   LITERALS MPDU_MT_AP_ASSOCIATION, MPDU_MT_AP_MAC_ADDR_RECEIVED,
  MPDU_MT_AP_DEASSOCIATION, MPDU_AP_MT_ASSOCIATION_ACK,
  MPDU_AP_MT_MAC_ADDR_ALLOCATE, MPDU_AP_MT_DEASSOCIATION,
  MPDU_MT_AP_CONN_ACTIVATE, MPDU_AP_MT_CONN_ACTIVE,
  MPDU_AP_MT_CONN_REFUSED, MPDU_AP_MT_CONN_PENDING,
  MPDU_TIP, MPDU_TIP_ACK, MPDU_BEACON;
ENDNEWTYPE;*/

/*NEWTYPE TED_type
   LITERALS TX_SOF, TX_FrameHeader, TX_DOWNLINK_MPDU, TX_UPLINK_MPDU,
            RX_SOF, RX_FrameHeader, RX,   RX_OFF;
ENDNEWTYPE;*/

/* MT type declaration */
/* ------------------------------ */

/* NEWTYPE HO_type
  LITERALS poweron_ho, FORWARD_HO, BACKWARD_HO, no_HO;
 ENDNEWTYPE;*/

NEWTYPE ATM_ad_type
   LITERALS BROADCAST_ADDR;
 ENDNEWTYPE ATM_ad_type;

SYNTYPE MAC_ad_type = NATURAL
  DEFAULT BROADCAST_MAC_ADDR
  CONSTANTS MT_MAX_actif
ENDSYNTYPE MAC_ad_type;

/* NEWTYPE Assoc_type
    LITERALS Non_Associated, Associated;
 ENDNEWTYPE Assoc_type;*/

NEWTYPE RC_Type
 LITERALS NO_ACTION, INVITATION, DEASSOC;
ENDNEWTYPE RC_Type;

NEWTYPE MT_STATUS_RecType STRUCT
 MT_ATM_Ad ATM_ad_type;
 ValidAddrFlag Boolean;
 WakeUpFlag Boolean;
 Counter Integer;
 Counter1 Integer;
ENDNEWTYPE;
```

system wand

```
 /* --------------------- connections -------------------- */
/*SYNTYPE MVC_Id_type =Integer   CONSTANTS 0:VC_max-1
ENDSYNTYPE MVC_Id_type;*/

/*NEWTYPE service_type   LITERALS delay_sensitive, loss_sensitive;
ENDNEWTYPE service_type;*/

/*NEWTYPE ArrayMvcDlrPid ARRAY(MVC_id_type, Pid);
ENDNEWTYPE ArrayMvcDlrPid;*/

/*NEWTYPE MVC_table   ARRAY(MVC_Id_type,boolean);
ENDNEWTYPE MVC_table;*/

/*NEWTYPE MVC_nb STRUCT
  MVC_Id MVC_id_Type;
  nb Integer;
ENDNEWTYPE MVC_nb;*/

/*NEWTYPE MVC_nb_string   STRING(MVC_nb,Void);
ENDNEWTYPE MVC_nb_string;*/

/*NEWTYPE MPDU_type   LITERALS data_type, FH_type, SOF_type;
ENDNEWTYPE MPDU_type;*/

/* ------------ Traffic Table --------------------- */

/*SYNTYPE TT_id_type = Integer CONSTANTS 0:Max_DLX_buffer
ENDSYNTYPE TT_id_type;*/

/*NEWTYPE ATM_cell STRUCT
  Cell_Number Integer;
  SN Integer;
  RN Integer;
  MVC_Id MVC_id_type;
  CRC boolean;
ENDNEWTYPE ATM_cell;*/

/*NEWTYPE ArrayATMcells  ARRAY(TT_id_type,ATM_cell);
ENDNEWTYPE ArrayATMcells;*/

/*NEWTYPE Cell_list  STRING(ATM_cell,Void);
ENDNEWTYPE Cell_list;*/

/*NEWTYPE Traffic_Table  ARRAY(MVC_Id_type,ArrayATMcells);
ENDNEWTYPE Traffic_Table;*/

  /* ------------- Slot Map ----------------- */

/*SYNTYPE Direction = Integer  CONSTANTS -1:1
ENDSYNTYPE Direction;*/

/*NEWTYPE Time_slot_entry STRUCT
 slot TIME;           + MVC_id, MAC_ad, length, offset
 dir Direction;
ENDNEWTYPE Time_slot_entry;*/

/*NEWTYPE TSlot_string             type of the slot map
   STRING(Time_slot_entry, Void);
ENDNEWTYPE TSlot_string;*/

SYNTYPE MT_MAC_AddrType=Integer CONSTANTS 0:MT_MAX_actif
ENdSYNTYPE;

NEWTYPE IAA_TableType ARRAY(MT_MAC_AddrType, MT_Status_RecType);
ENDNEWTYPE;

NEWTYPE Toggle_ARRAY Array(MT_MAC_AddrType, Boolean);
ENDNEWTYPE;
```

system wand

```
/* Upper layer signals */
/* ------------------- */
/* SIGNAL RELEASEAAL0_req,
  OPENAAL5_req, CLOSEAAL5_req, XMITAAL5_req, RELEASEAAL5_req,
  RECVAAL5_ind(signal_type),*/
/*SIGNAL ALLOCATEAAL0_req, ALLOCATEAAL0_cnf,
      XMITAAL0_req(Integer), RECVAAL0_ind(Integer),
      OPENAAL0_req, CLOSEAAL0_req;*/

/* Upper layer signallists */
/* ----------------------- */
/* SIGNALLIST Aatm2MDLX = RECVAAL0_ind, ALLOCATEAAL0_cnf;*/
/* SIGNALLIST MDLX2Aatm = RELEASEAAL0_req, ALLOCATEAAL0_req;*/
/* SIGNALLIST MLCP2Aatm = OPENAAL5_req, CLOSEAAL5_req, XMITAAL5_req,
                      RELEASEAAL5_req; */
/*SIGNALLIST MDLR2Aatm = XMITAAL0_req;*/
/*SIGNALLIST MGDL2Aatm = OPENAAL0_req, CLOSEAAL0_req;*/
```

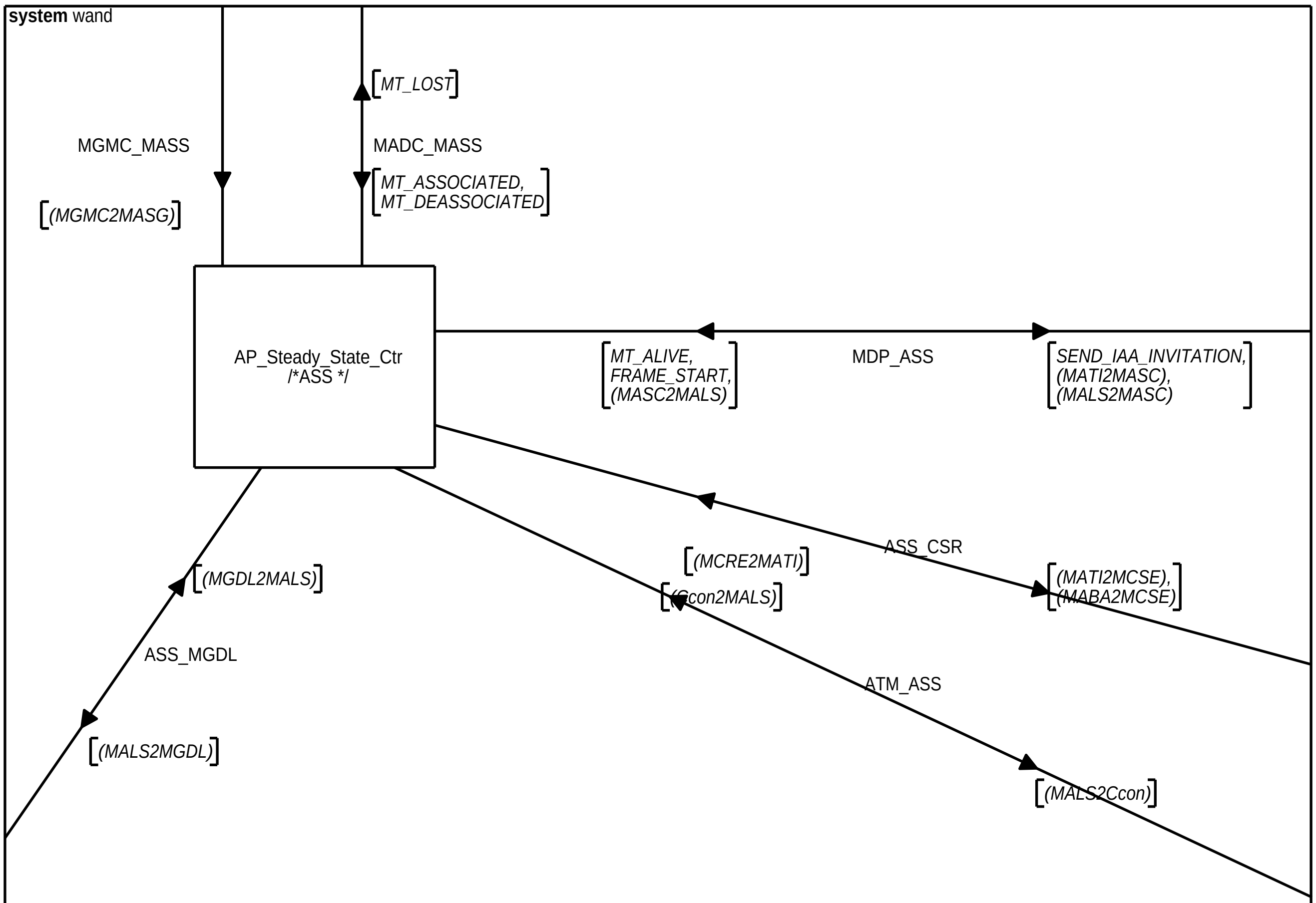
```
/* PEER TO PEER CONTROL SIGNALS (between AP and MT)  */
/* ------------------------------------------------ */

SIGNAL /* MPDU_MT_AP_ASSOCIATION, MPDU_MT_AP_MAC_ADDR_RECEIVED,
 MPDU_MT_AP_DEASSOCIATION(MAC_ad_type, return_type),
 MPDU_AP_MT_ASSOCIATION_ACK(return_type),
 MPDU_AP_MT_MAC_ADDR_ALLOCATE(ATM_ad_type),
 MPDU_AP_MT_DEASSOCIATION(MAC_ad_type, return_type),
 MPDU_MT_AP_CONN_ACTIVATE, MPDU_AP_MT_CONN_ACTIVE,
 MPDU_AP_MT_CONN_REFUSED, MPDU_AP_MT_CONN_PENDING, */
 MPDU_TIP(MAC_ad_type,tip_type), MPDU_TIP_ACK(MAC_ad_type),  MPDU_BEACON;

/*SIGNAL RX_FH;*/

/* PEER TO PEER CONTROL SIGNALS (between AP and MT) */
/* ------------------------------------------------ */

/*SIGNALLIST MMAA2MAAA_INITIAL_MPDU = MPDU_MT_AP_ASSOCIATION;*/
/*SIGNALLIST MMAA2MAAA_MPDU = MPDU_MT_AP_MAC_ADDR_RECEIVED,
                           MPDU_MT_AP_DEASSOCIATION;*/
/*SIGNALLIST MAAA2MMAA_MPDU = MPDU_AP_MT_ASSOCIATION_ACK,
                           MPDU_AP_MT_MAC_ADDR_ALLOCATE,
                           MPDU_AP_MT_DEASSOCIATION;*/
/*SIGNALLIST MMMA2MAMA_MPDU = MPDU_MT_AP_CONN_ACTIVATE;*/
/*SIGNALLIST MAMA2MMMA_MPDU = MPDU_AP_MT_CONN_ACTIVE, MPDU_AP_MT_CONN_REFUSED,
                           MPDU_AP_MT_CONN_PENDING;*/
SIGNALLIST MMTI2MATI_MPDU = MPDU_TIP;
SIGNALLIST MATI2MMTI_MPDU = MPDU_TIP_ACK;
SIGNALLIST MABA2MMBA_MPDU = MPDU_BEACON;


/*SIGNAL From_MT_to_AP(MAC_ad_type,MPDU_type,Cell_list, MVC_Id_type, Integer,Integer),
      From_AP_to_MT(MPDU_type,TSlot_string,Cell_list,MVC_Id_type,Integer);*/
/*SIGNAL PID_REGISTERED(MAC_ad_type, MVC_Id_type, Pid),
      PID_DEREGISTERED(MAC_ad_type,MVC_Id_type,Pid);*/
/*SIGNAL SERVICE_PARAMETER(service_type),
      SET_TRAFFIC_PARM(MAC_ad_type,MVC_Id_type, service_type),
      DESTROY(MVC_Id_type);*/
```

system wand

```
/* AP MAC Layer signals */
/* -------------------- */

/* Peer to peer : Mascara Control to Signalling */

SIGNAL ALARM_ind, GET_INFO_cnf, RR_STATUS_cnf/*, ASSOCIATION_ind,
 AP_DEASSOCIATION_ind, AP_VC_RELEASE_cnf(return_type),
 AP_DEASSOCIATION_cnf(return_type),
 AP_ASSOCIATION_LOST_ind, VC_RESERVED_cnf(return_type),
 VC_SWITCHED_cnf(return_type),
 CLOSE_MAC_cnf, OPEN_AP_MAC_cnf(return_type)*/, FRAME_START;

/* Peer to peer : Signalling to Mascara Control */

SIGNAL /*AP_VC_RELEASE_req, AP_DEASSOCIATION_req, VC_RESERVED_req,
VC_SWITCHED_req,*/ GET_INFO_req, RR_STATUS_req/*, CLOSE_MAC_req,
OPEN_AP_MAC_req*/;
```

```
/* AP MAC Layer signallists */
/* ------------------------ */

/* AP Peer to peer : Mascara Control to Signalling */
/* ----------------------------------------------- */
SIGNALLIST MALS2Ccon = ALARM_ind, GET_INFO_cnf, RR_STATUS_cnf;
/*SIGNALLIST MAAA2Ccon = ASSOCIATION_ind , AP_DEASSOCIATION_ind,
                      AP_VC_RELEASE_cnf;*/
/*SIGNALLIST MADG2Ccon = AP_DEASSOCIATION_cnf, AP_ASSOCIATION_LOST_ind,
                      AP_VC_RELEASE_cnf;*/
/*SIGNALLIST MAMA2Ccon = VC_RESERVED_cnf, VC_SWITCHED_cnf;*/
/*SIGNALLIST MGMCA2Ccon = CLOSE_MAC_cnf, OPEN_AP_MAC_cnf;*/

/* AP Peer to peer : Signalling to Mascara Control */
/* ----------------------------------------------- */
/*SIGNALLIST Ccon2MAAA = AP_VC_RELEASE_req;*/
/*SIGNALLIST Ccon2MADG = AP_DEASSOCIATION_req;*/
/*SIGNALLIST Ccon2MAMA = VC_RESERVED_req, VC_SWITCHED_req;*/
SIGNALLIST Ccon2MALS = GET_INFO_req, RR_STATUS_req;
/*SIGNALLIST Ccon2MGMCA = CLOSE_MAC_req, OPEN_AP_MAC_req;*/
```

/* The 2 signallists Ccon2MGMC have been renamed by Ccon2MGMCA and
Ccon2MGMCM – Idem for MGMC2Ccon */



/* The channel MT_AP replaces the physical layer of both AP and MT, as well as
the wireless link between them. */

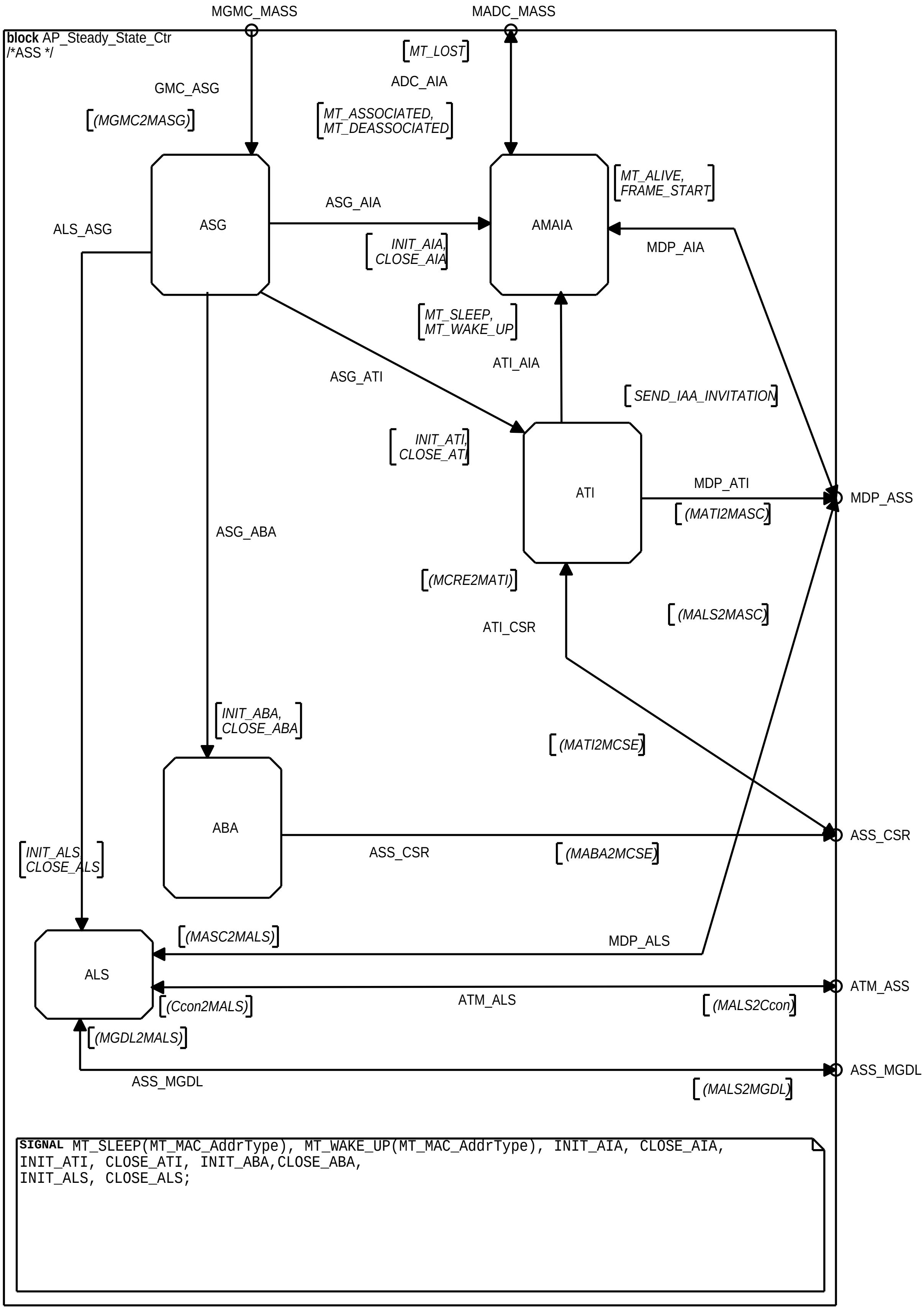
/* The environment is composed of :
- the upper layers of the MT (accessed through channel user)
- the upper layers of AP and the wired ATM network (accessed through channel
AP_to_CS_or_ATM) */

/* We will probably need to specify another MT, representing the broadcast
MT with MT_MAC_Addr=BROADCAST_MT_MAC_ADDR and MVC_Id = BROADCAST_MVC_Id_ADDR.
All broadcast signalling is done in the AP through the CSR/WDLC of that virtual
connection*/

```
/* AP MAC Layer */

SIGNAL WDLC_COLLECT_DATA_cnf,
/*INSTANCIATE_DLC(MAC_ad_type,MVC_id_type,service_type), */
 ASC_COLLECT_DATA_cnf;

/* ICC - Control */
/* DELETED FROM THIS MODEL ---> Abstraction of ICC !!!!

SIGNALLIST MMCN2MADG = (Ccon2MAAA), (Ccon2MAMA), (Ccon2MADG);
SIGNALLIST MMCN2MALS = (Ccon2MALS);
SIGNALLIST MMCN2MGMC = (Ccon2MGMCA);

SIGNALLIST MALS2MMCN = (MALS2Ccon);
SIGNALLIST MAAA2MMCN = (MAAA2Ccon), (MAMA2Ccon);
SIGNALLIST MAMA2MMCN = (MAMA2Ccon);
SIGNALLIST MADG2MMCN = (MADG2Ccon), (MAMA2Ccon);
SIGNALLIST MGMC2MMCN = (MGMCA2Ccon); */

/* Control - CSR */
/*SIGNALLIST MAAA2MCSE = (MAAA2MMAA_MPDU);*/
/*SIGNALLIST MAMA2MCSE = (MAMA2MMMA_MPDU);*/
SIGNALLIST MABA2MCSE = (MABA2MMBA_MPDU);
SIGNALLIST MATI2MCSE = (MATI2MMTI_MPDU);
/*SIGNALLIST MAAA2MGSR = INSTANCIATE_CSR, DESTROY;*/
/*SIGNALLIST MCRE2MAAA = (MMAA2MAAA_MPDU), (MMMA2MAMA_MPDU);*/
/*SIGNALLIST MCRE2MAMA = (MMMA2MAMA_MPDU);*/
SIGNALLIST MCRE2MATI = (MMTI2MATI_MPDU);
/*SIGNALLIST MGSR2MAAA = PID_REGISTERED;*/
/*SIGNALLIST MGSR2MGMC = PID_REGISTERED;*/  /* to be verified ! */
/*SIGNALLIST MGMC2MGSR = INSTANCIATE_CSR, DESTROY ;*/ /* to be verified ! */

/* Control - WDLC */
/*SIGNALLIST MAMA2MGDL = INSTANCIATE_DLC, DESTROY;*/
SIGNALLIST MALS2MGDL = WDLC_COLLECT_DATA_req;
SIGNALLIST MGDL2MALS = WDLC_COLLECT_DATA_cnf, WDLC_ALARM_ind;
/*SIGNALLIST MGDL2MAMA = PID_REGISTERED;*/

/* added on the base of error messages */
/* SIGNAL asc_alarm_ind, close_mdp_done, tip_mt_ind(MT_MAC_AddrType), init_mdp;
SIGNAL close_mdp, traffic_released, send_iaa_invitation, init_mdp_done; */

/* Control - Data Pump */
/*SIGNALLIST MGMC2MGDP = INIT_MDP, CLOSE_MDP, SET_TRAFFIC_PARM,
                       TRAFFIC_RELEASED;*/
/*SIGNALLIST MAMA2MGDP = SET_TRAFFIC_PARM, TRAFFIC_RELEASED;*/
/*SIGNALLIST MAIA2MASC = SEND_IAA_INVITATION;*/
SIGNALLIST MALS2MASC = ASC_COLLECT_DATA_req;
SIGNALLIST MATI2MASC = TIP_MT_ind;
/*SIGNALLIST MGDP2MGMC = INIT_MDP_DONE, CLOSE_MDP_DONE;*/
/*SIGNALLIST MMPR2MADG = (MMAA2MAAA_INITIAL_MPDU);*/
SIGNALLIST MASC2MALS = ASC_COLLECT_DATA_cnf, ASC_ALARM_ind;

/* CSR - WDLC */
/*SIGNALLIST MGSR2MGDL = INSTANCIATE_DLC, DESTROY;*/
/*SIGNALLIST MGDL2MGSR = PID_REGISTERED;*/

/* WDLC - Data Pump */
/*SIGNALLIST MGDL2MMPR = PID_REGISTERED, PID_DEREGISTERED;*/
/* SIGNALLIST MDLX2MASC = GEN_WDLC_FB;*/
```

```
/* AP Mascara Control */

/* It is responsible for the general Control of Mascara */

SIGNAL /*AP_DEASSOCIATE_DONE, AP_DEASSOCIATE,INIT_ADC, */
 MT_ASSOCIATED(ATM_ad_type, MAC_ad_type), MT_DEASSOCIATED(MAC_ad_type),
 INIT_ASS, CLOSE_ASS/*, INIT_RCL, CLOSE_RCL,
 INIT_RCL_OK, INIT_RCL_KO, CLOSE_RCL_DONE*/;

/* Generic - Dynamic */
/*SIGNALLIST MGMC2MADG = AP_DEASSOCIATE, INIT_ADC;*/
/*SIGNALLIST MADG2MGMC = AP_DEASSOCIATE_DONE;*/

/* Generic - Steady State */
SIGNALLIST MGMC2MASG = INIT_ASS, CLOSE_ASS;

/* Generic - Radio */
/*SIGNALLIST MGMC2MRCM = INIT_RCL, CLOSE_RCL;*/
/*SIGNALLIST MRCM2MGMC = INIT_RCL_OK, INIT_RCL_KO, CLOSE_RCL_DONE;*/

/* Steady State - Dynamic */
SIGNAL MT_LOST(MAC_ad_type);
/*SIGNALLIST MADG2MAIA = MT_ASSOCIATED, MT_DEASSOCIATED;*/
```

```
/* Physical layer - Control */
/* SIGNAL PHY_MEASURE_RSSI, PHY_SELECT_CHANNEL, PHY_MEASURED_RSSI;*/
/* These signals have been encapsulated in the
Radio Control process */

/* CSR - WDLC */
/*SIGNAL CTRL_CELL_RCVD, XMIT_CTRL_CELL(MPDU_sig_type);*/

/* WDLC - Data Pump */
/*SIGNAL TRAFFIC_UPDATED(Integer, Integer, MAC_ad_type, MVC_id_type,
   Cell_list, Integer),
   GEN_WDLC_FB ,
   MPDU_BODY_RCVD(Cell_list, Integer, Integer);*/

/* Data Pump - Control */
SIGNAL MT_ALIVE(MAC_ad_type);

/* Control - CSR */
/*SIGNAL INSTANCIATE_CSR(MAC_ad_type, MVC_Id_type);*/

/* Control - WDLC */
SIGNAL WDLC_ALARM_ind, WDLC_COLLECT_DATA_req;

/* Control - Data Pump */
SIGNAL /*INIT_MDP, CLOSE_MDP,*/
  ASC_ALARM_ind, /*INIT_MDP_DONE, CLOSE_MDP_DONE,*/ TIP_MT_ind(MT_MAC_AddrType)/*,
  TRAFFIC_RELEASED(MVC_id_type)*/;

SIGNAL SEND_IAA_INVITATION(MAC_ad_type),
      ASC_COLLECT_DATA_req;
```

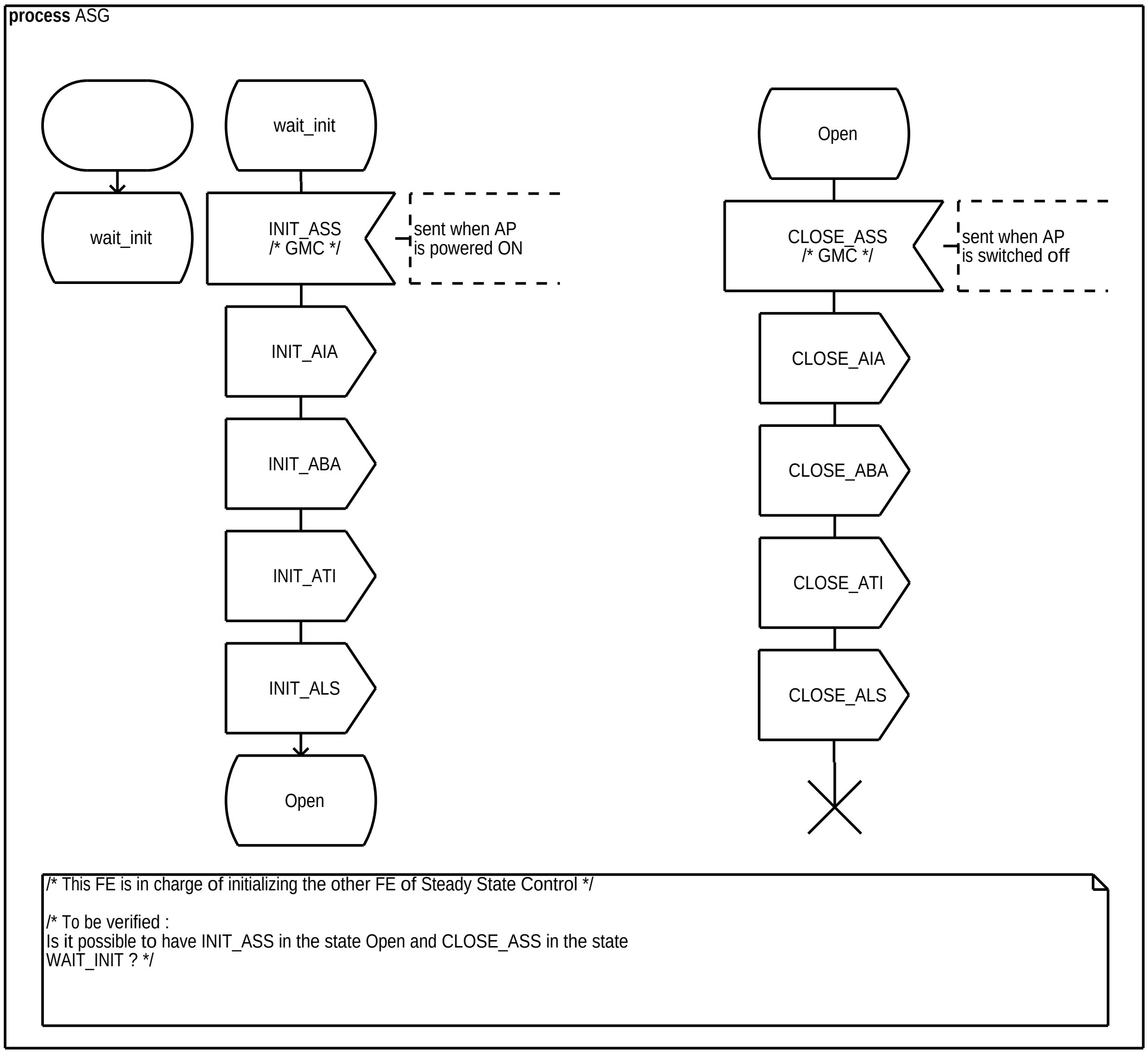


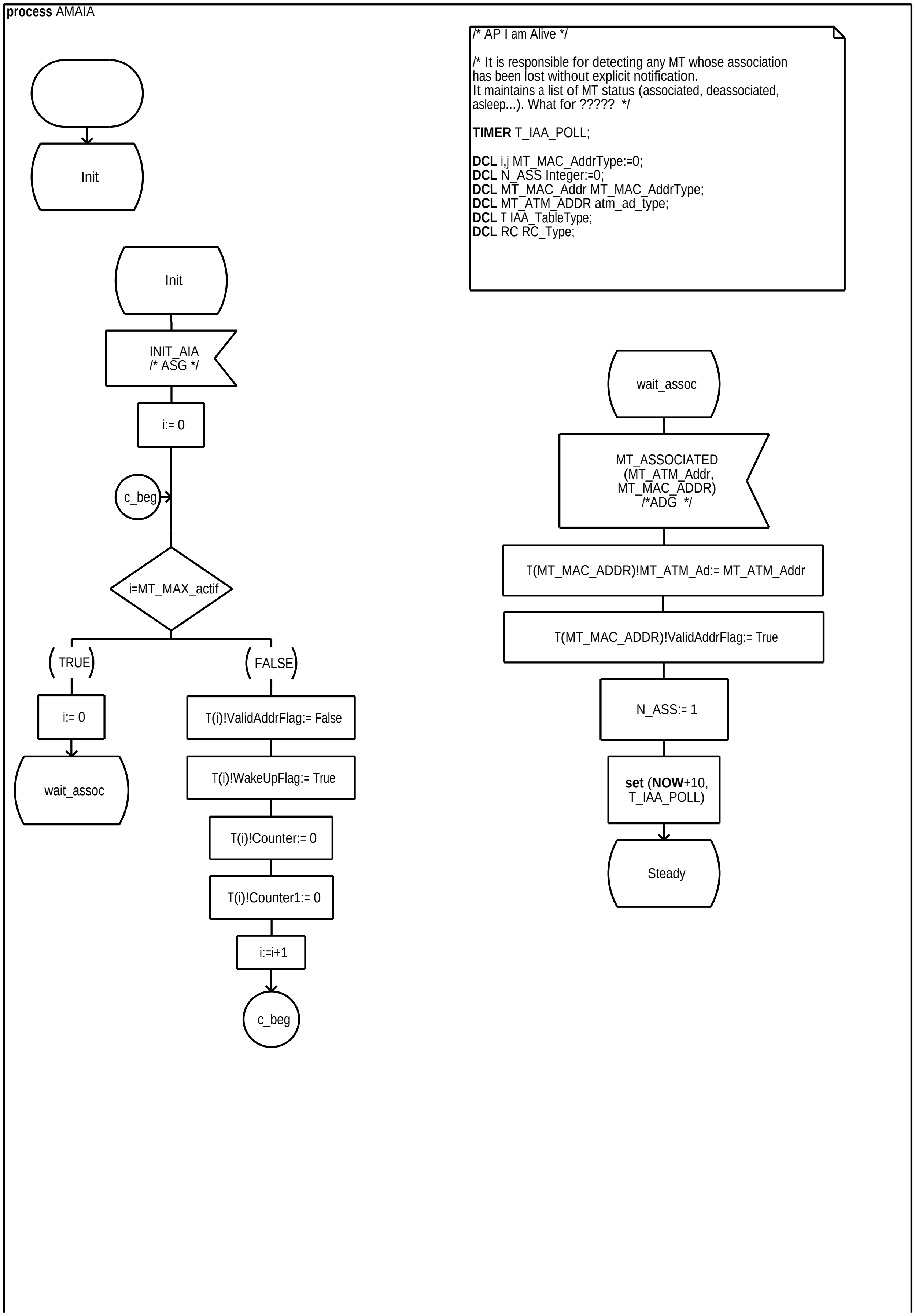
MGMC_MASS
MADC_MASS
block AP_Steady_State_Ctr
/*ASS */
[MT_LOST]
GMC_ASG
ADC_AIA
[(MGMC2MASG)]
[MT_ASSOCIATED,
MT_DEASSOCIATED]
ASG
ASG_AIA
AMAIA
[MT_ALIVE,
FRAME_START]
ALS_ASG
[INIT_AIA,
CLOSE_AIA]
MDP_AIA
[MT_SLEEP,
MT_WAKE_UP]
ATI_AIA
ASG_ATI
[SEND_IAA_INVITATION]
[INIT_ATI,
CLOSE_ATI]
ATI
MDP_ATI
MDP_ASS
[(MATI2MASC)]
ASG_ABA
[(MCRE2MATI)]
[(MALS2MASC)]
ATI_CSR
[INIT_ABA,
CLOSE_ABA]
[(MATI2MCSE)]
ABA
ASS_CSR
[INIT_ALS
CLOSE_ALS]
ASS_CSR
[(MABA2MCSE)]
[(MASC2MALS)]
MDP_ALS
ALS
ATM_ASS
[(Ccon2MALS)]
ATM_ALS
[(MALS2Ccon)]
[(MGDL2MALS)]
ASS_MGDL
ASS_MGDL
[(MALS2MGDL)]
SIGNAL MT_SLEEP(MT_MAC_AddrType), MT_WAKE_UP(MT_MAC_AddrType), INIT_AIA, CLOSE_AIA,
INIT_ATI, CLOSE_ATI, INIT_ABA,CLOSE_ABA,
INIT_ALS, CLOSE_ALS;

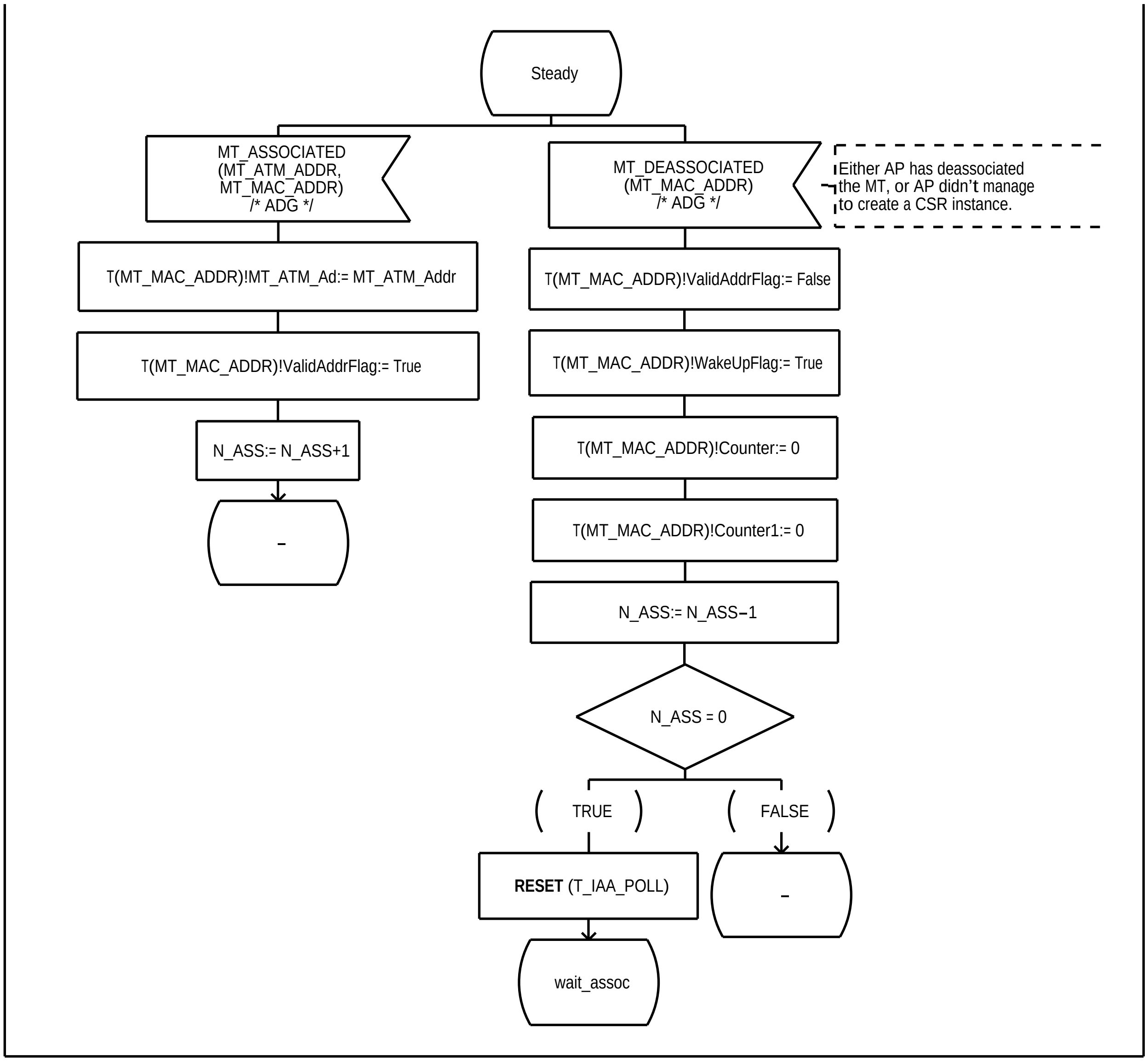
/* These various functional entities correspond to static control
functions. SOME OF THEM ARE NOT DESCRIBED */

/* ASG: initialises the other FE's */
/* AMAIA (I Am Alive) : Detects any MT whose association has been lost */
/* ABA (Beacon) : Broadcasts at regular intervals the beacon */
/* ATI (TIP) : Runs the TIP protocol (identifies incommunicado MTs */
/* ALS (Link Status Recorder) : Compiles the status of the channels */

/* The signals INIT_ASS, CLOSE_ASS from Gen Masc Control are missing */

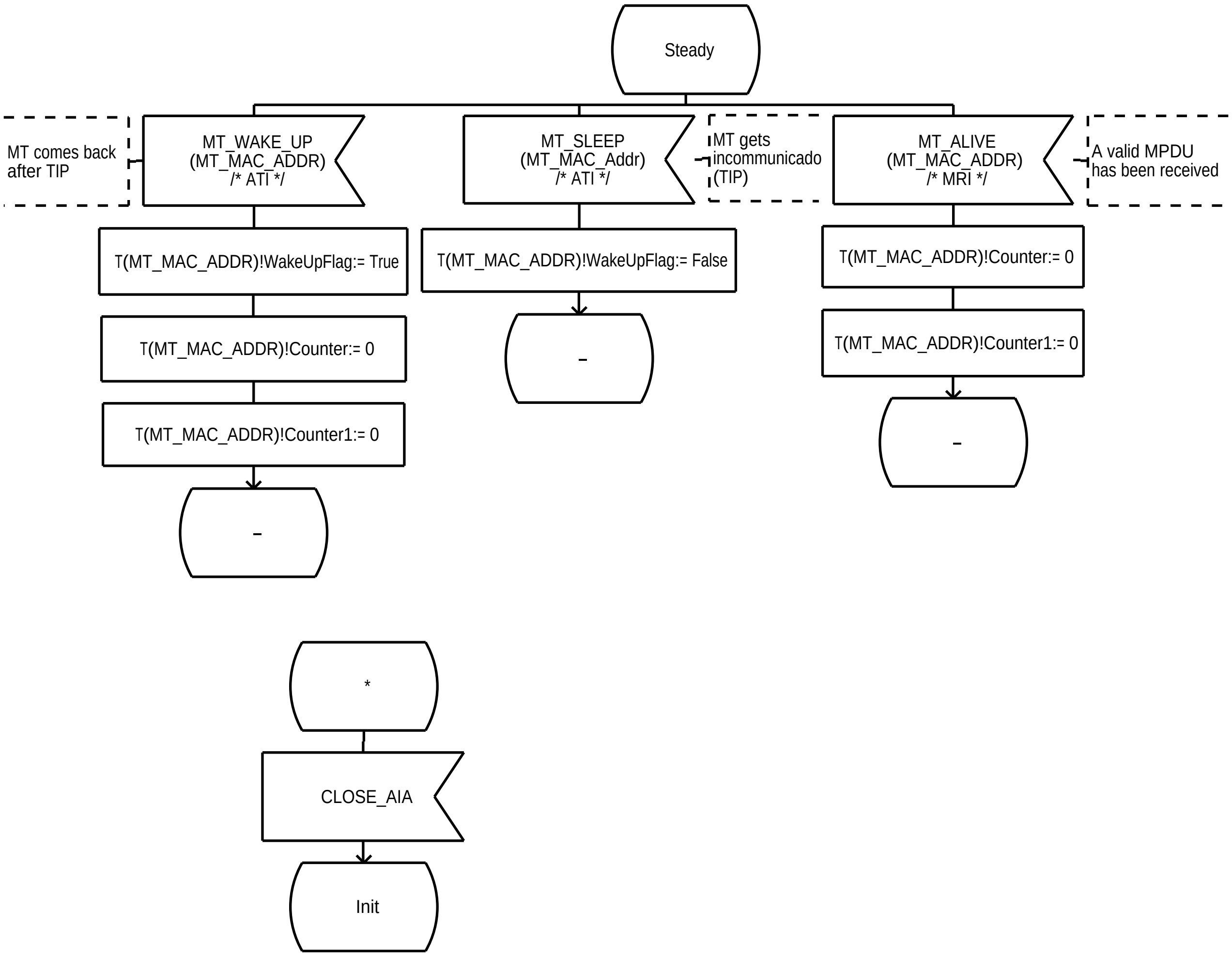

process ASG
wait_init
wait_init
INIT_ASS
/* GMC */
sent when AP
is powered ON
INIT_AIA
INIT_ABA
INIT_ATI
INIT_ALS
Open
Open
CLOSE_ASS
/* GMC */
sent when AP
is switched off
CLOSE_AIA
CLOSE_ABA
CLOSE_ATI
CLOSE_ALS
/* This FE is in charge of initializing the other FE of Steady State Control */
/* To be verified :
Is it possible to have INIT_ASS in the state Open and CLOSE_ASS in the state
WAIT_INIT ? */


process AMAIA
Init
/* AP I am Alive */
/* It is responsible for detecting any MT whose association has been lost without explicit notification.
It maintains a list of MT status (associated, deassociated, asleep...). What for ????? */
TIMER T_IAA_POLL;
DCL i,j MT_MAC_AddrType:=0;
DCL N_ASS Integer:=0;
DCL MT_MAC_Addr MT_MAC_AddrType;
DCL MT_ATM_ADDR atm_ad_type;
DCL T IAA_TableType;
DCL RC RC_Type;
Init
INIT_AIA /* ASG */
i:= 0
c_beg
i=MT_MAX_actif
TRUE
FALSE
i:= 0
wait_assoc
T(i)!ValidAddrFlag:= False
T(i)!WakeUpFlag:= True
T(i)!Counter:= 0
T(i)!Counter1:= 0
i:=i+1
c_beg
wait_assoc
MT_ASSOCIATED (MT_ATM_Addr, MT_MAC_ADDR) /*ADG */
T(MT_MAC_ADDR)!MT_ATM_Ad:= MT_ATM_Addr
T(MT_MAC_ADDR)!ValidAddrFlag:= True
N_ASS:= 1
set (NOW+10, T_IAA_POLL)
Steady



Steady
MT_ASSOCIATED (MT_ATM_ADDR, MT_MAC_ADDR) /* ADG */
T(MT_MAC_ADDR)!MT_ATM_Ad:= MT_ATM_Addr
T(MT_MAC_ADDR)!ValidAddrFlag:= True
N_ASS:= N_ASS+1
-
MT_DEASSOCIATED (MT_MAC_ADDR) /* ADG */
Either AP has deassociated the MT, or AP didn't manage to create a CSR instance.
T(MT_MAC_ADDR)!ValidAddrFlag:= False
T(MT_MAC_ADDR)!WakeUpFlag:= True
T(MT_MAC_ADDR)!Counter:= 0
T(MT_MAC_ADDR)!Counter1:= 0
N_ASS:= N_ASS−1
N_ASS = 0
TRUE
FALSE
RESET (T_IAA_POLL)
-
wait_assoc

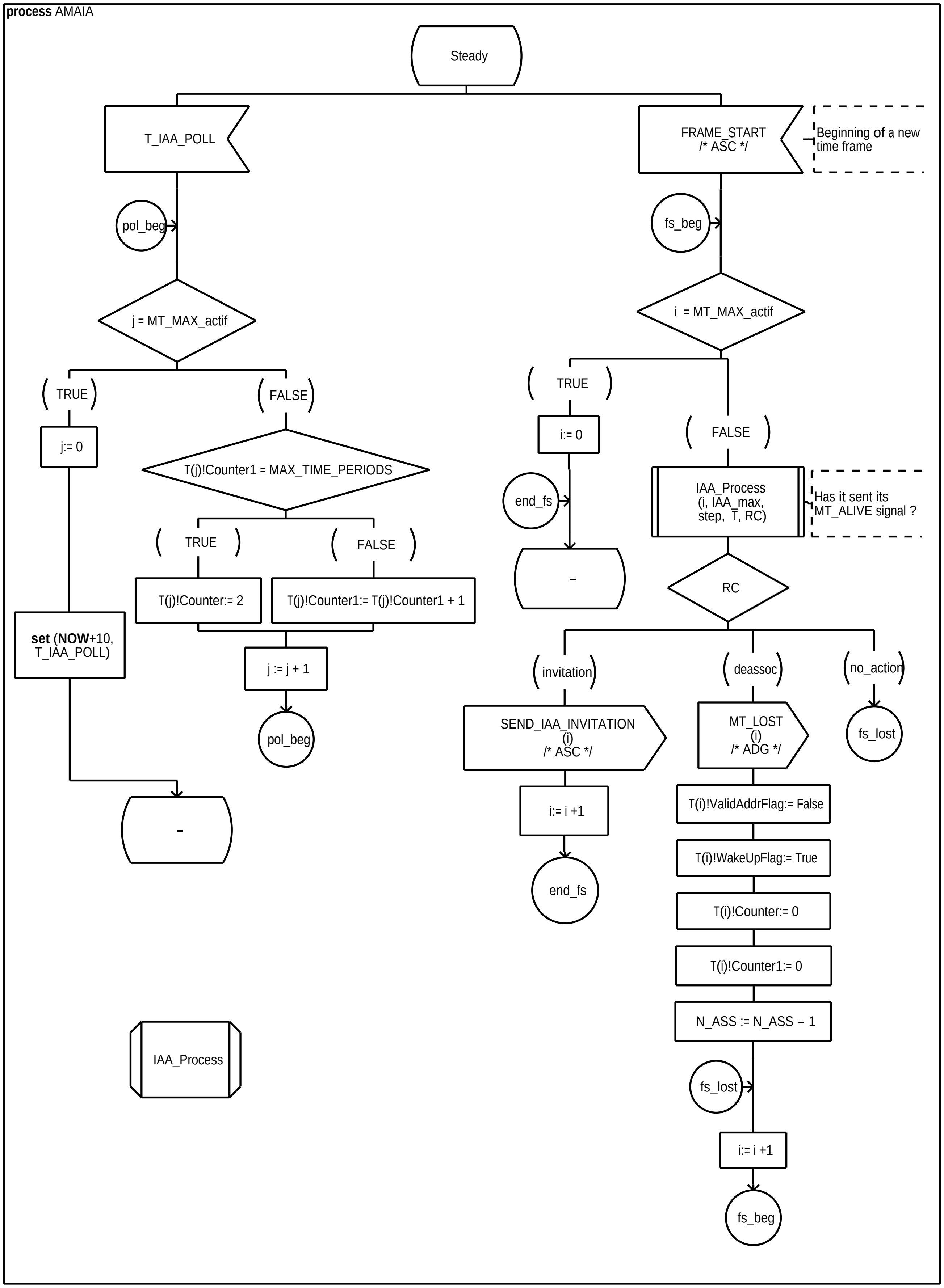
process AMAIA



/* AIA is in the state Steady when there is at least one MT associated,
otherwise it is in the state wait_assoc */

/* AIA checks all associated MT for verifying if they have sent
their I_AM_ALIVE signals when expected. If not, send an invitation,
which forces the MT to respond by sending a MPDU. There is at most
one IAA_invitation sent per time frame.

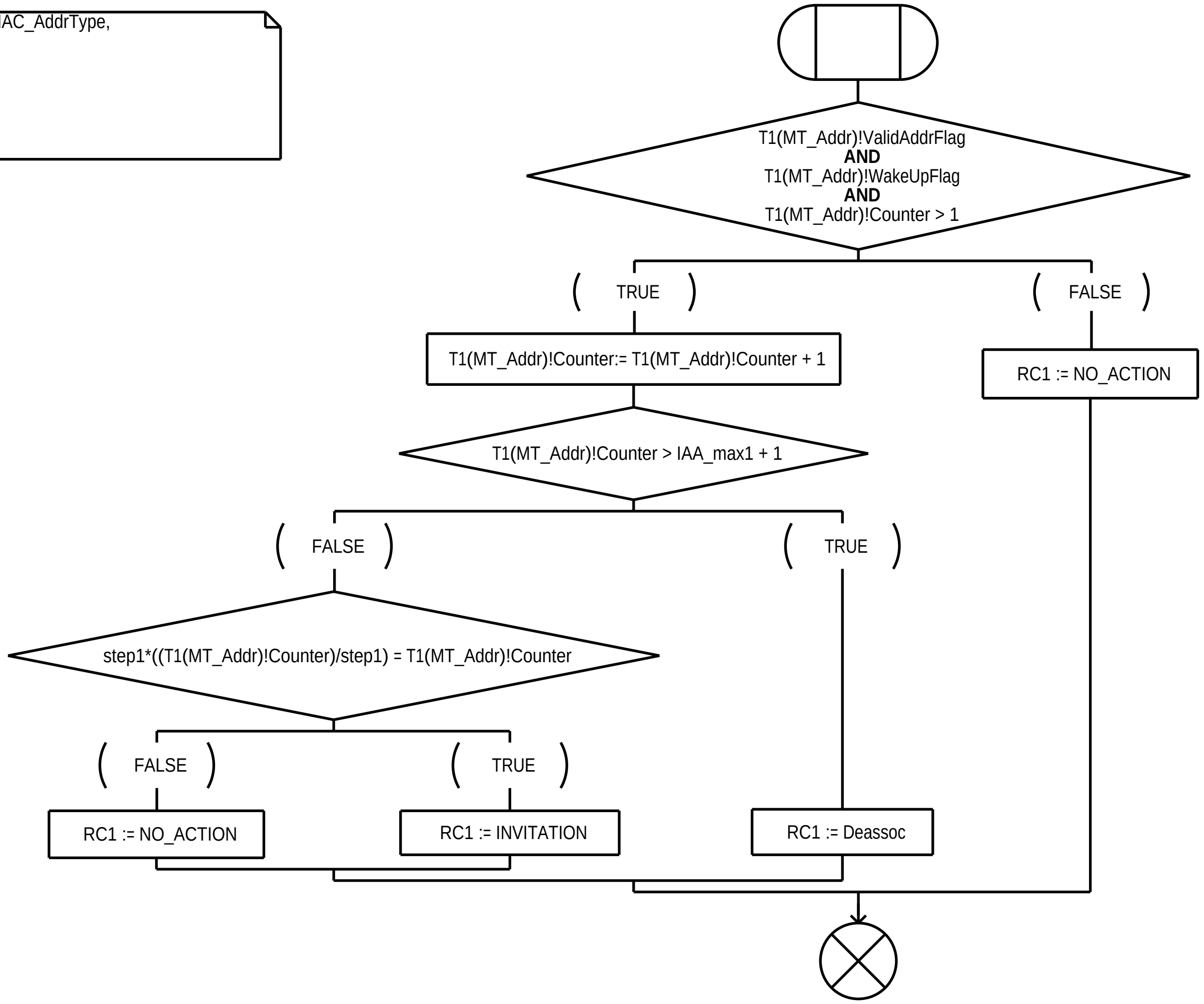
If MT_ALIVE is received, that means that a valid MPDU has been
received by the MPR. If after several invitations, there is still
no response, the MT is considered as lost. */



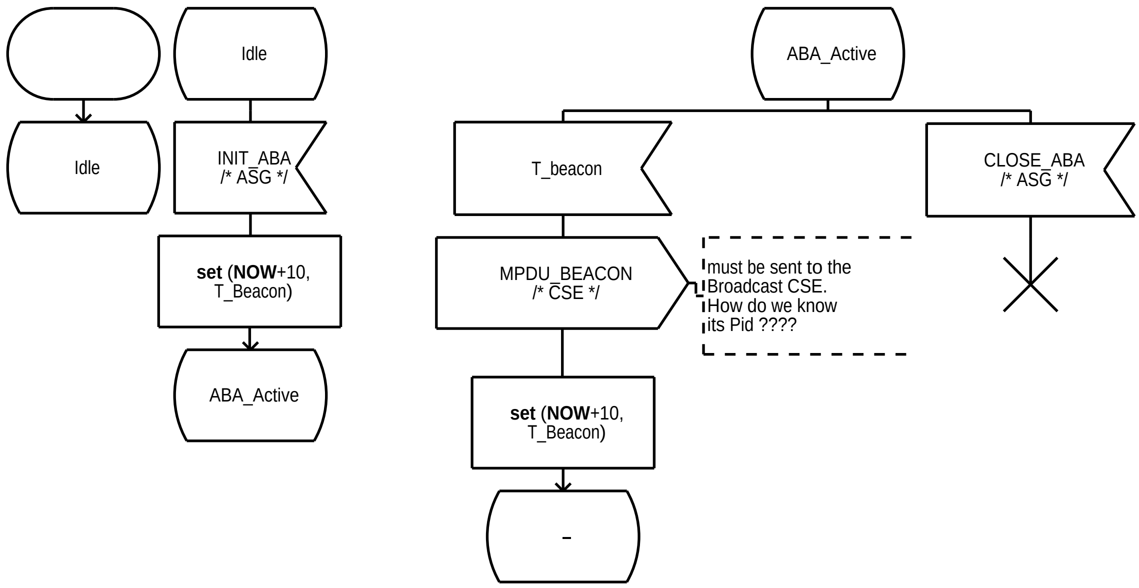
process AMAIA
Steady
T_IAA_POLL
pol_beg
j = MT_MAX_actif
TRUE
FALSE
j:= 0
T(j)!Counter1 = MAX_TIME_PERIODS
TRUE
FALSE
T(j)!Counter:= 2
T(j)!Counter1:= T(j)!Counter1 + 1
j := j + 1
pol_beg
set (NOW+10, T_IAA_POLL)
-
IAA_Process
FRAME_START /* ASC */
Beginning of a new time frame
fs_beg
i = MT_MAX_actif
TRUE
FALSE
i:= 0
end_fs
-
IAA_Process (i, IAA_max, step, T, RC)
Has it sent its MT_ALIVE signal ?
RC
invitation
deassoc
no_action
SEND_IAA_INVITATION (i) /* ASC */
i:= i +1
end_fs
MT_LOST (i) /* ADG */
fs_lost
T(i)!ValidAddrFlag:= False
T(i)!WakeUpFlag:= True
T(i)!Counter:= 0
T(i)!Counter1:= 0
N_ASS := N_ASS – 1
fs_lost
i:= i +1
fs_beg

procedure IAA_Process

FPAR IN/OUT MT_Addr MT_MAC_AddrType,
IN IAA_max1 Integer,
IN step1 Integer,
IN/OUT T1 IAA_TableType,
IN/OUT RC1 RC_Type;



process ABA

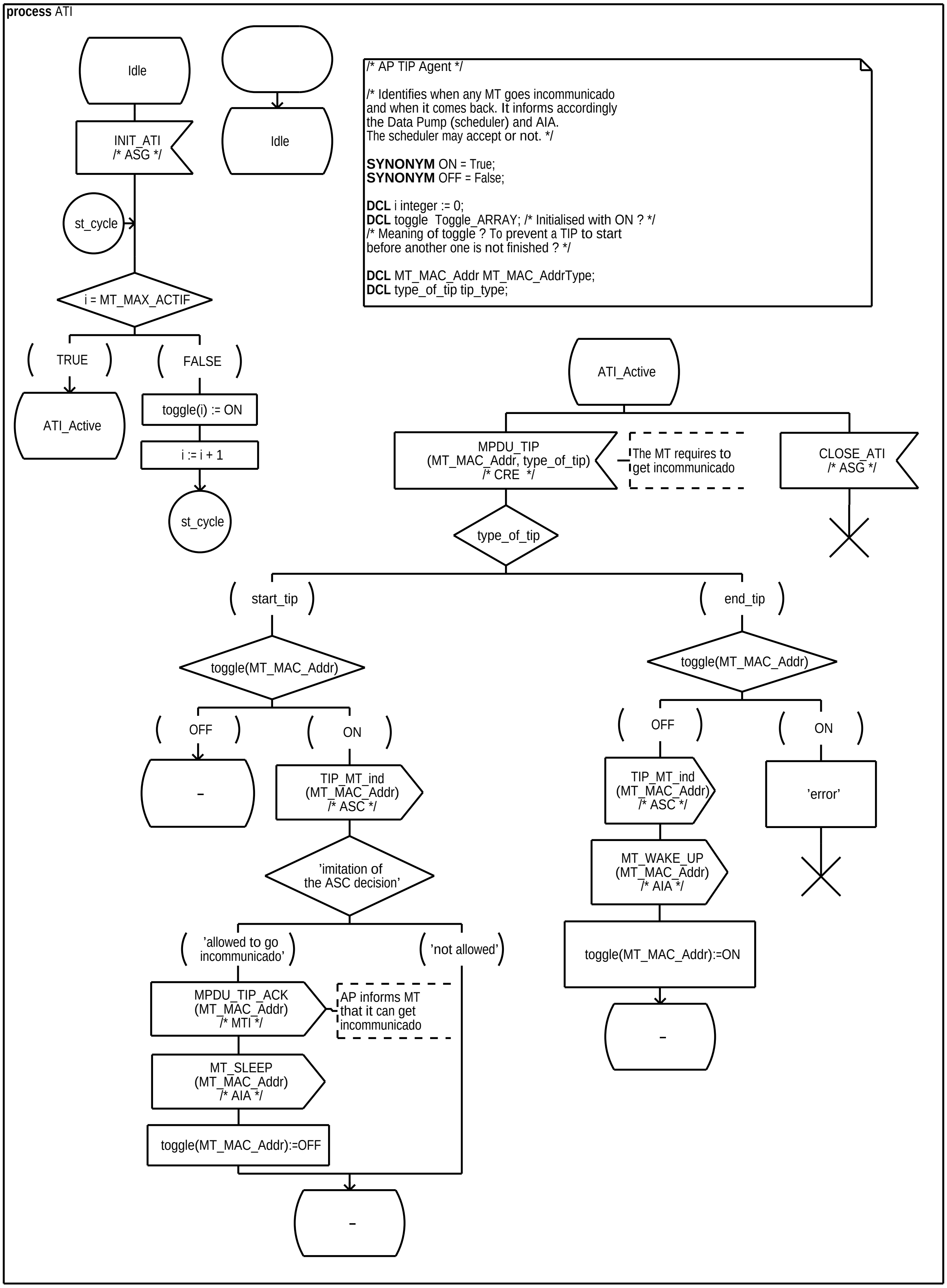


/* AP Beacon Agent */

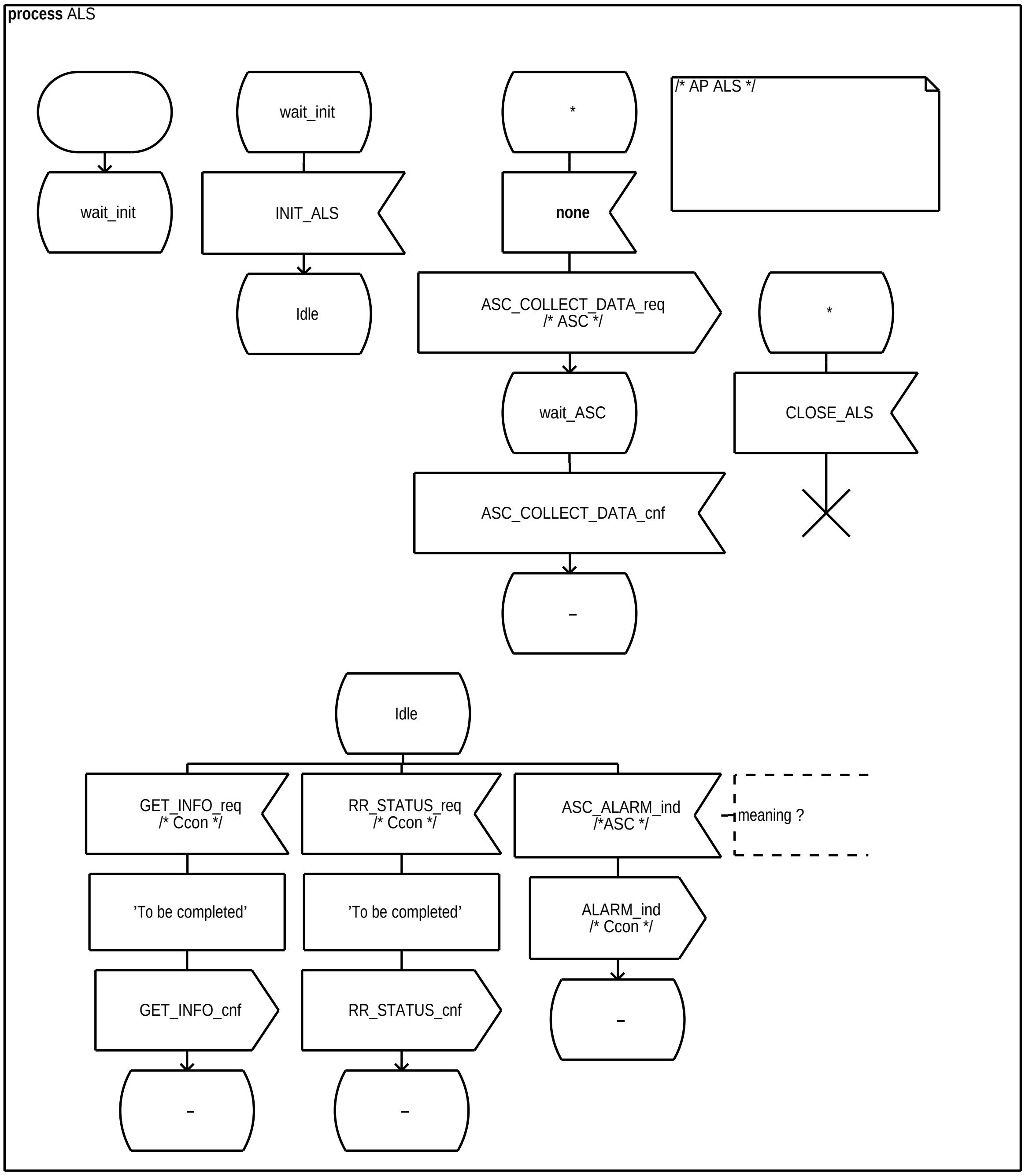
/* Is responsible for the broadcasting at regular intervals the Beacon
control MPDU used to assist the radio environment procedure of a new MT,
as well as the handover procedure of an associated MT.

It receives information related to neighbouring AP (HOW ?) */

TIMER T_Beacon; /* What is the period ? */



process ATI
Idle
INIT_ATI
/* ASG */
st_cycle
i = MT_MAX_ACTIF
TRUE
FALSE
ATI_Active
toggle(i) := ON
i := i + 1
st_cycle
Idle
/* AP TIP Agent */
/* Identifies when any MT goes incommunicado
and when it comes back. It informs accordingly
the Data Pump (scheduler) and AIA.
The scheduler may accept or not. */
SYNONYM ON = True;
SYNONYM OFF = False;
DCL i integer := 0;
DCL toggle Toggle_ARRAY; /* Initialised with ON ? */
/* Meaning of toggle ? To prevent a TIP to start
before another one is not finished ? */
DCL MT_MAC_Addr MT_MAC_AddrType;
DCL type_of_tip tip_type;
ATI_Active
MPDU_TIP
(MT_MAC_Addr, type_of_tip)
/* CRE */
The MT requires to
get incommunicado
CLOSE_ATI
/* ASG */
type_of_tip
start_tip
end_tip
toggle(MT_MAC_Addr)
OFF
ON
-
TIP_MT_ind
(MT_MAC_Addr)
/* ASC */
'imitation of
the ASC decision'
'allowed to go
incommunicado'
'not allowed'
MPDU_TIP_ACK
(MT_MAC_Addr)
/* MTI */
AP informs MT
that it can get
incommunicado
MT_SLEEP
(MT_MAC_Addr)
/* AIA */
toggle(MT_MAC_Addr):=OFF
-
toggle(MT_MAC_Addr)
OFF
ON
TIP_MT_ind
(MT_MAC_Addr)
/* ASC */
'error'
MT_WAKE_UP
(MT_MAC_Addr)
/* AIA */
toggle(MT_MAC_Addr):=ON
-


process ALS
wait_init
wait_init
INIT_ALS
Idle
*
none
/* AP ALS */
ASC_COLLECT_DATA_req
/* ASC */
*
wait_ASC
CLOSE_ALS
ASC_COLLECT_DATA_cnf
-
Idle
GET_INFO_req
/* Ccon */
RR_STATUS_req
/* Ccon */
ASC_ALARM_ind
/*ASC */
meaning ?
'To be completed'
'To be completed'
ALARM_ind
/* Ccon */
GET_INFO_cnf
RR_STATUS_cnf
-
-
-