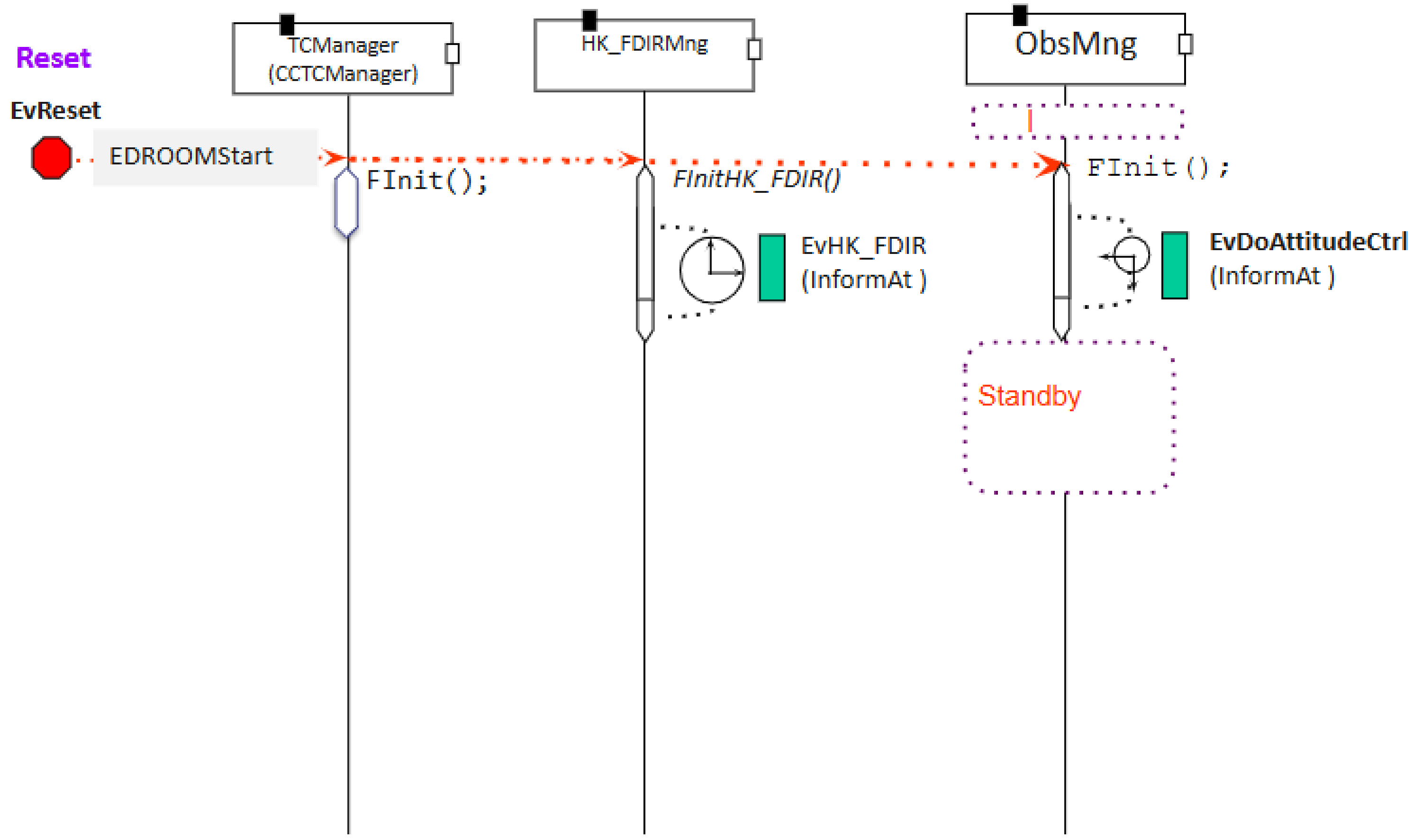
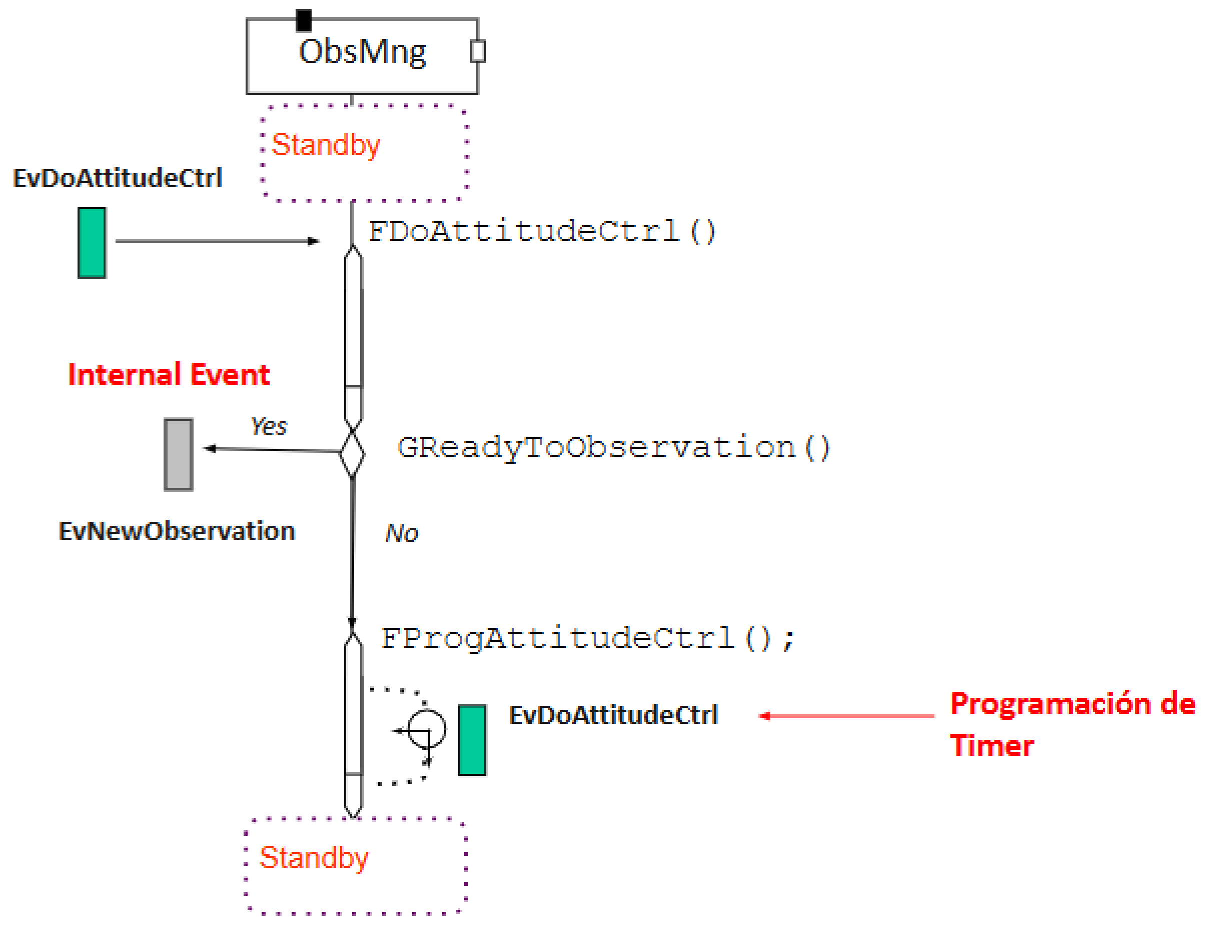
Entregables 1,2,3,4

Adrián Pérez Antón

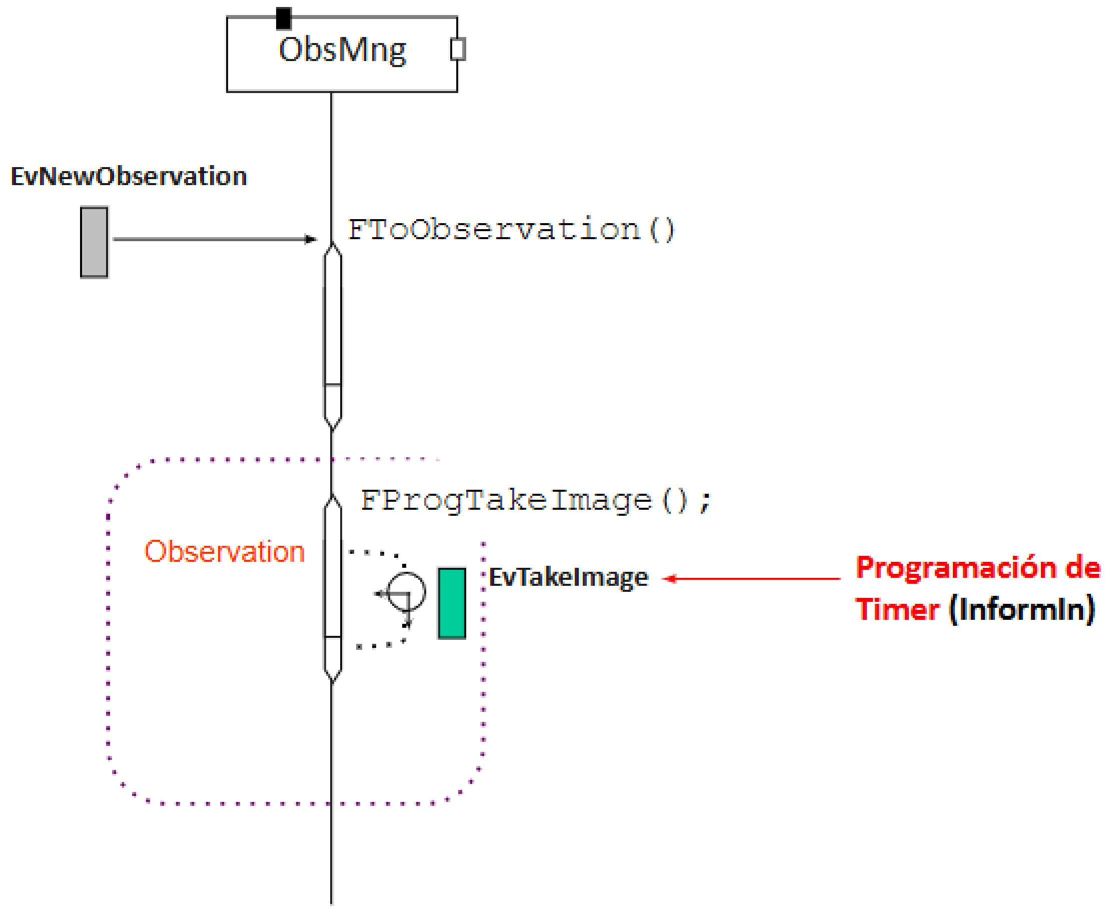
ESCENARIOS CCObsMng



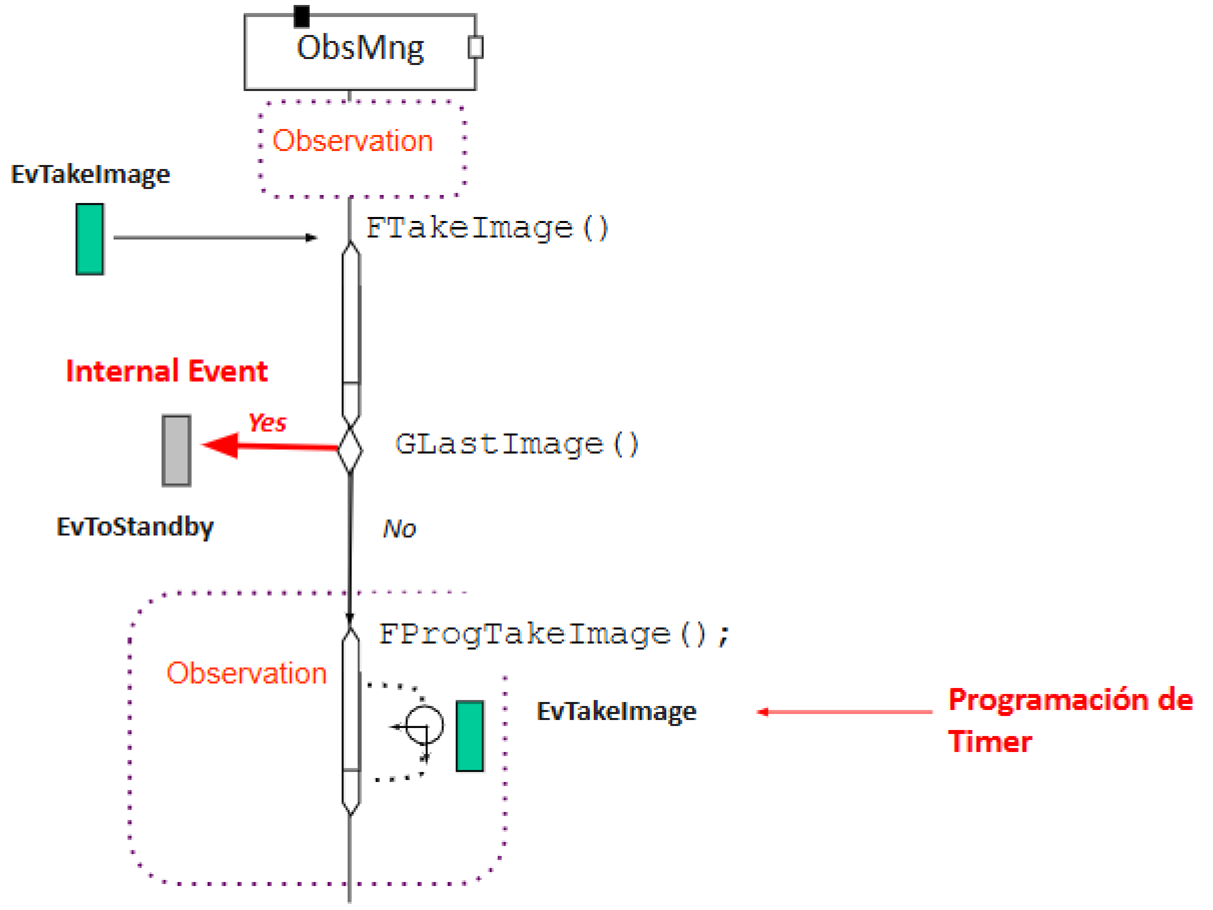
ESCENARIOS CCObsMng



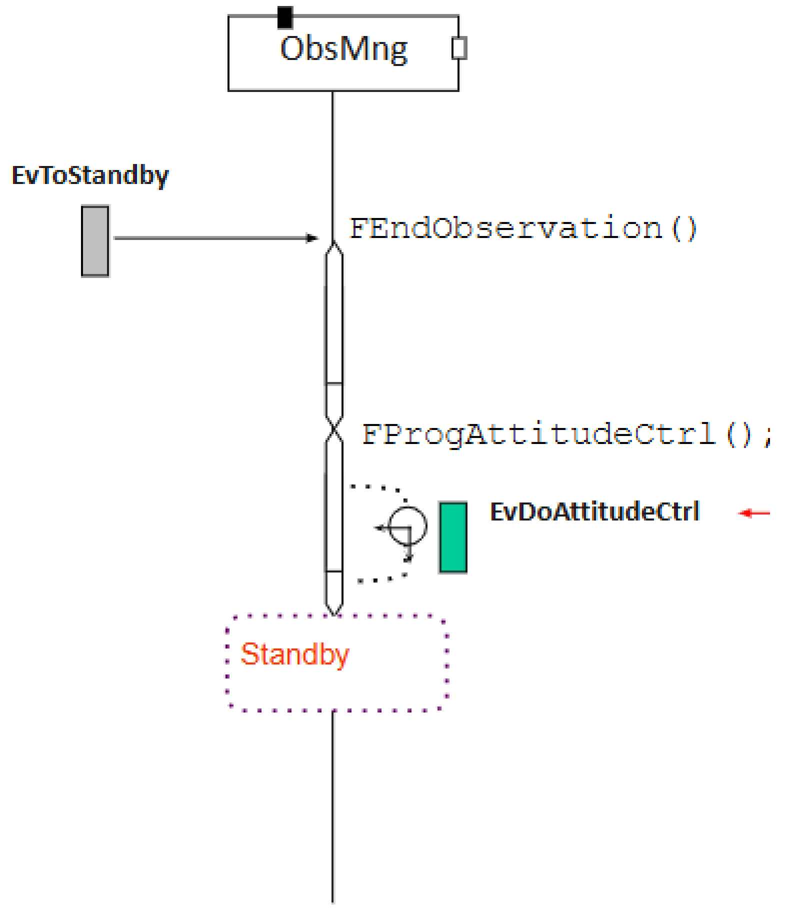
ESCENARIOS CCObsMng



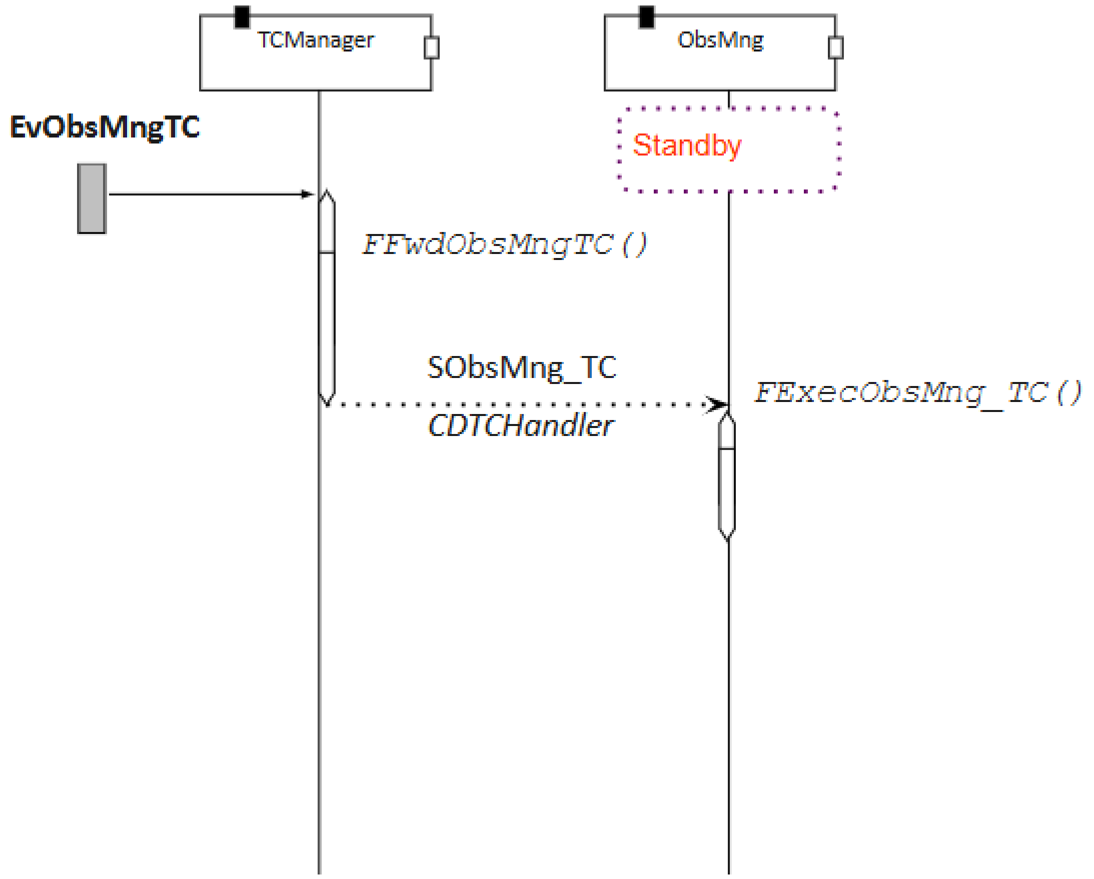
ESCENARIOS CCObsMng



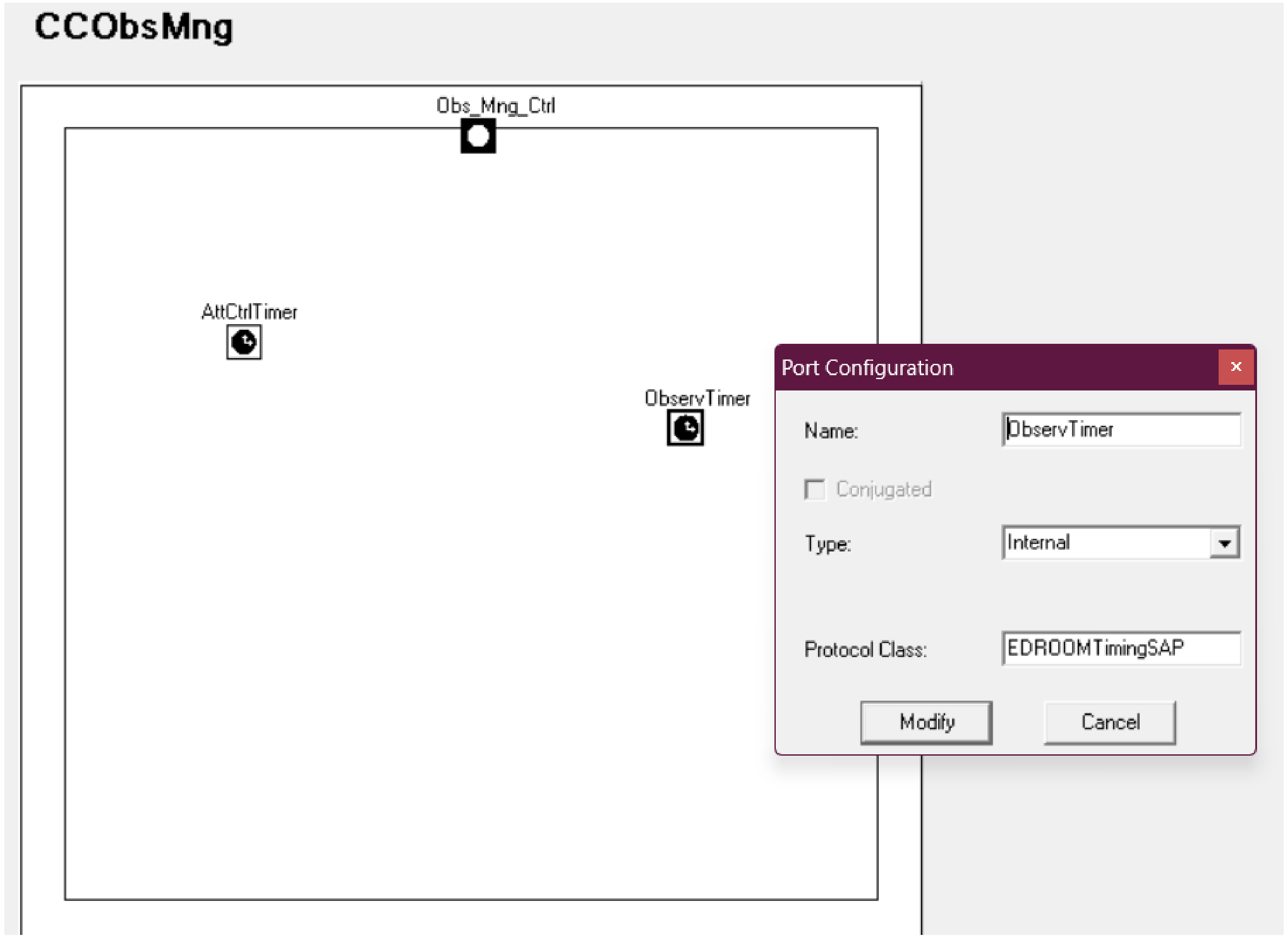
ESCENARIOS CCObsMng



ESCENARIOS CCObsMng



DEFINICIÓN CLASE PROTOCOLO



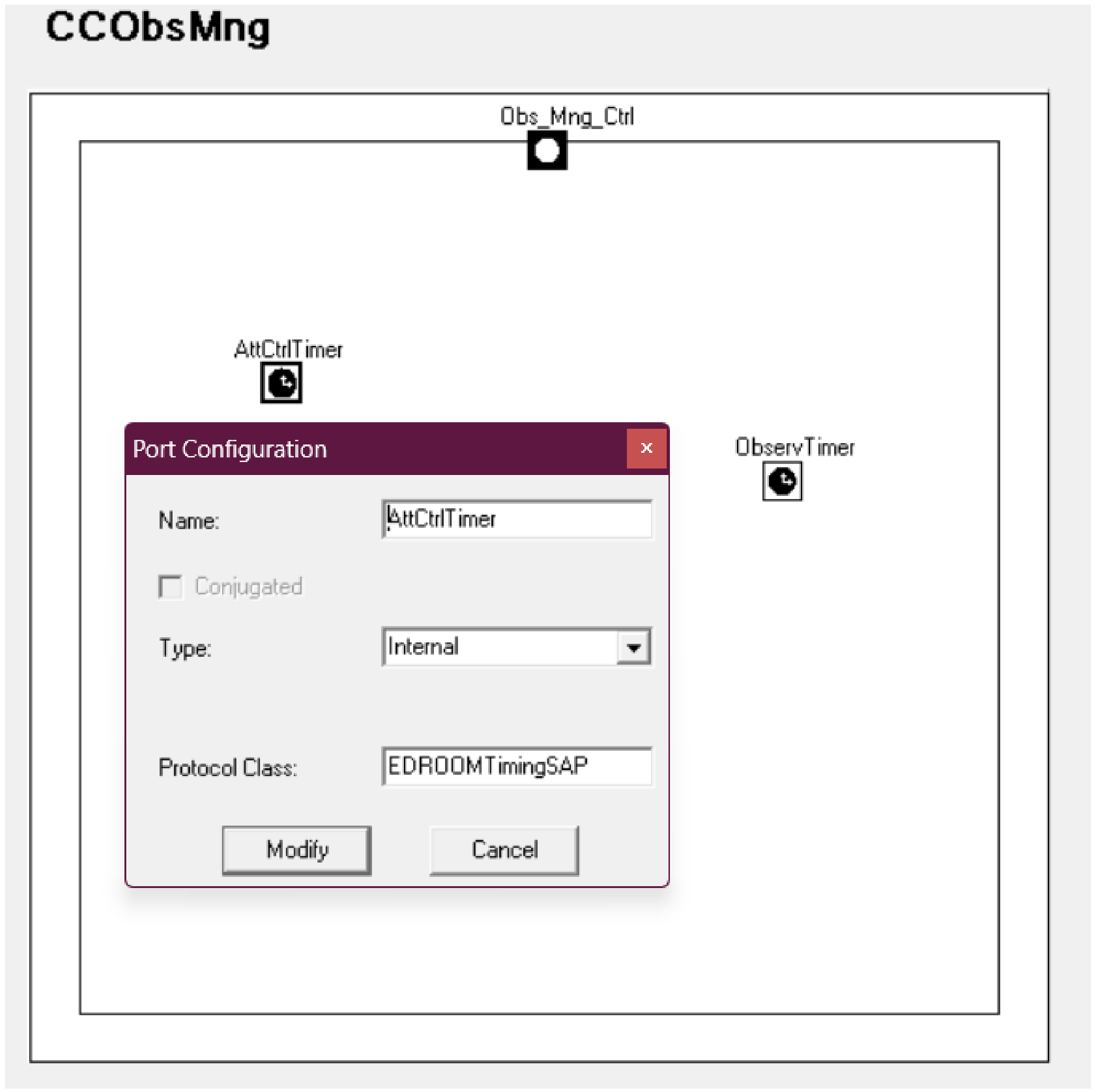
Name: ObservTimer

[ ] Conjugated

Type: Internal

Protocol Class: EDROOMTimingSAP

DEFINICIÓN CLASE PROTOCOLO



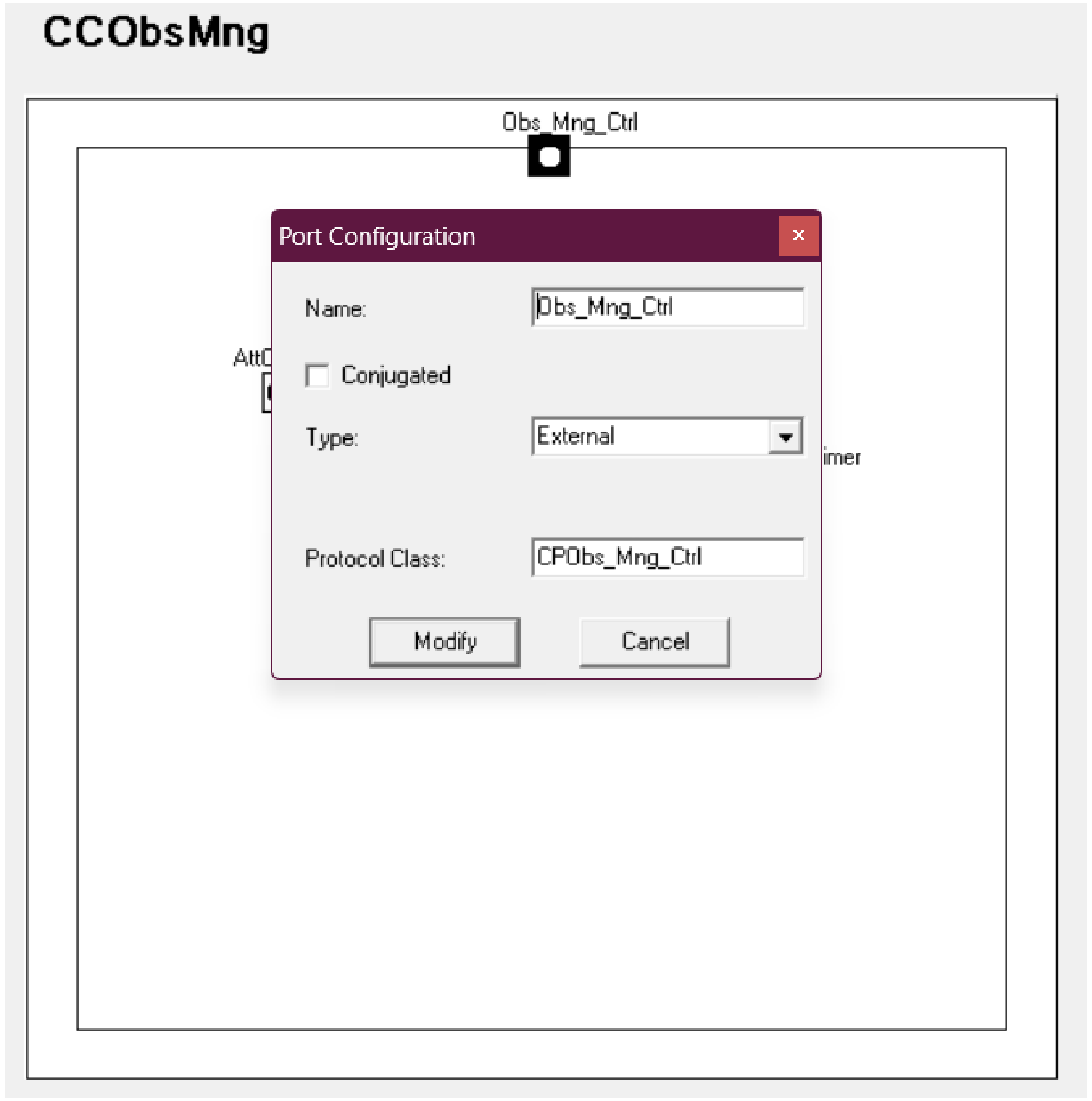
Name: AttCtrlTimer

Conjugated

Type: Internal

Protocol Class: EDROOMTimingSAP

DEFINICIÓN CLASE PROTOCOLO



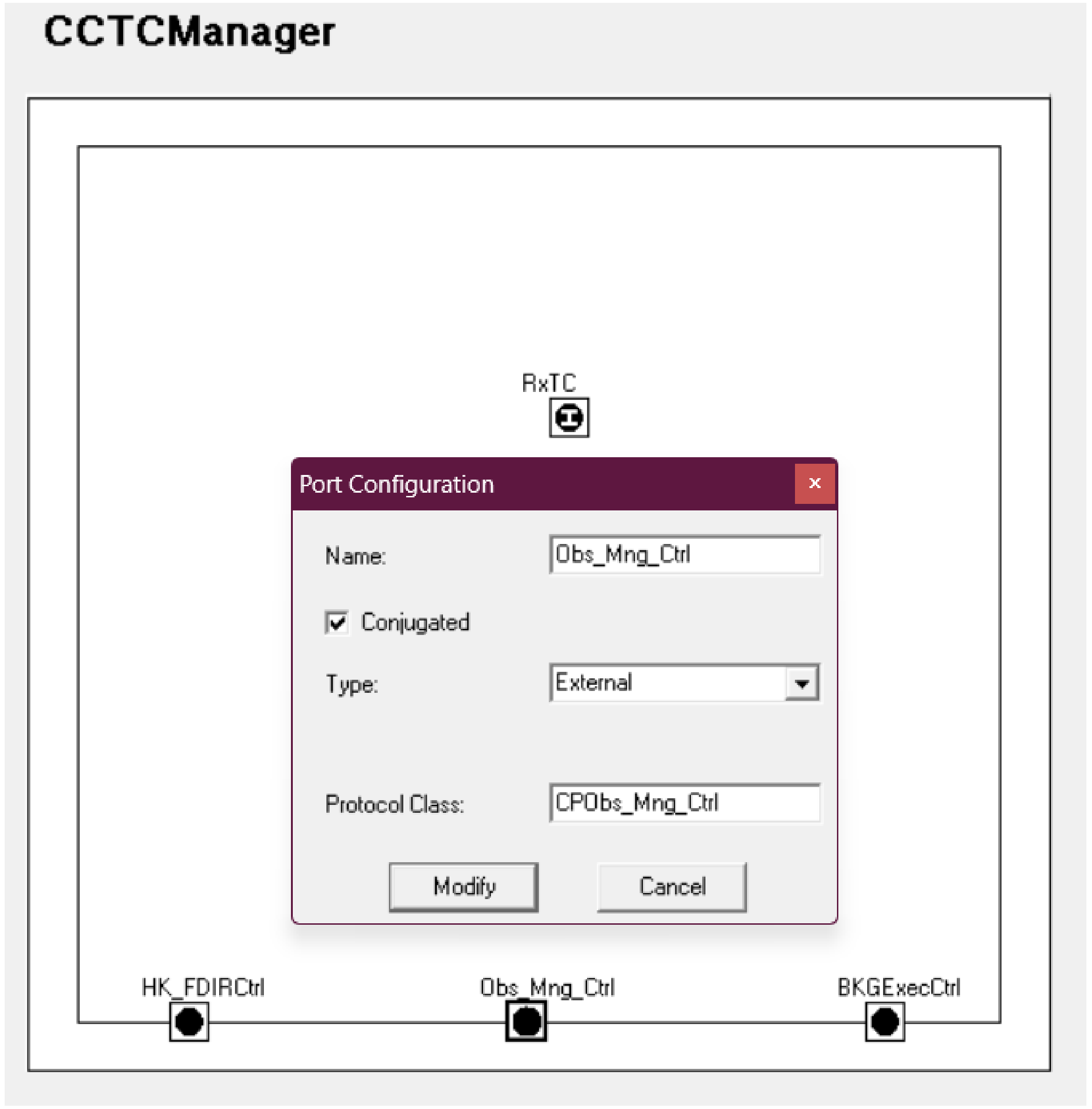
Name: Obs_Mng_Ctrl

Conjugated [ ]

Type: External

Protocol Class: CPObs_Mng_Ctrl

DEFINICIÓN CLASE PROTOCOLO



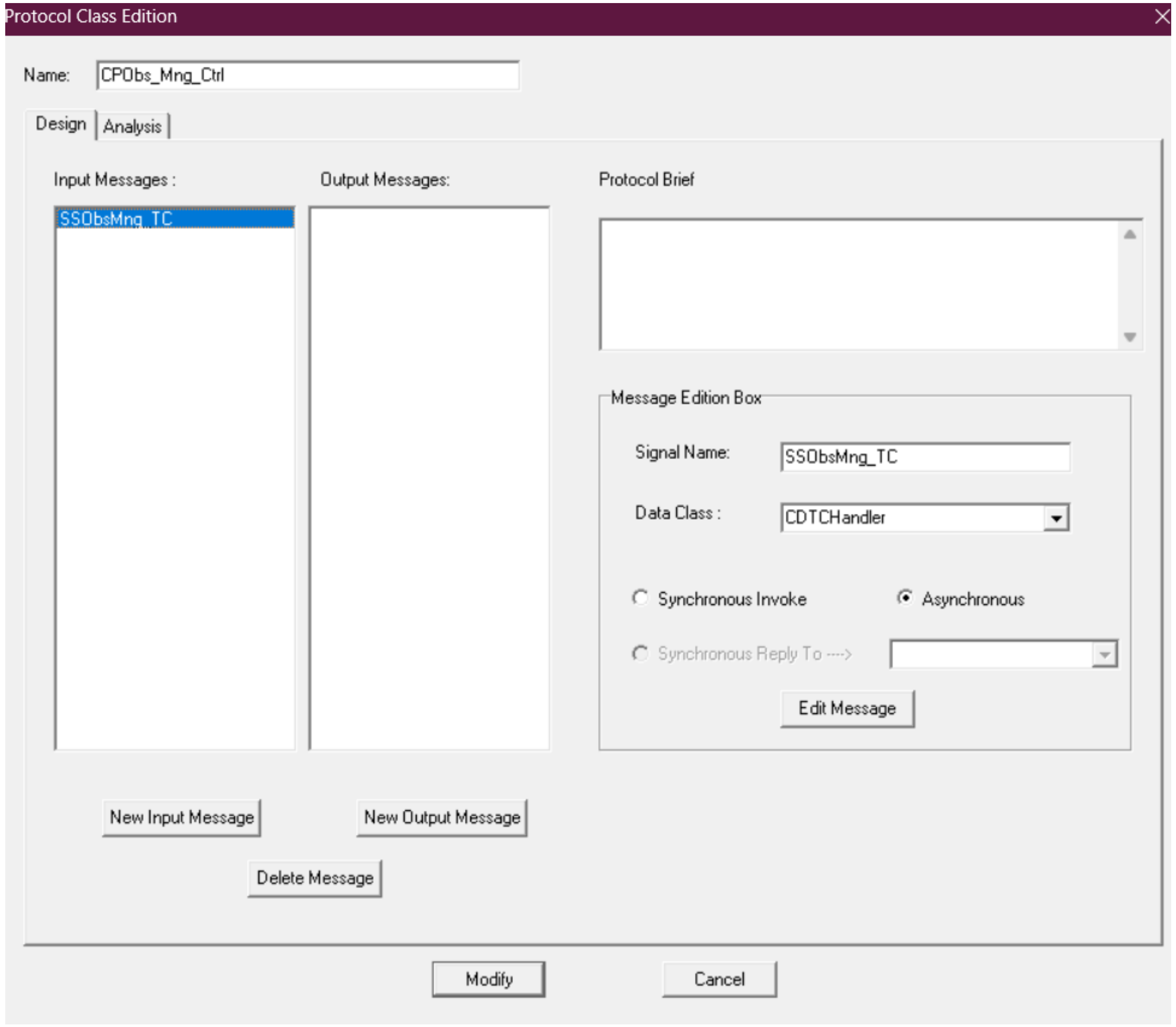
Name: Obs_Mng_Ctrl

☑ Conjugated

Type: External

Protocol Class: CPObs_Mng_Ctrl

DEFINICIÓN CLASE PROTOCOLO



Protocol Class Edition

Name: CPObs_Mng_Ctrl

Design | Analysis

Input Messages: SSObsMng_TC

Output Messages:

Protocol Brief

Message Edition Box

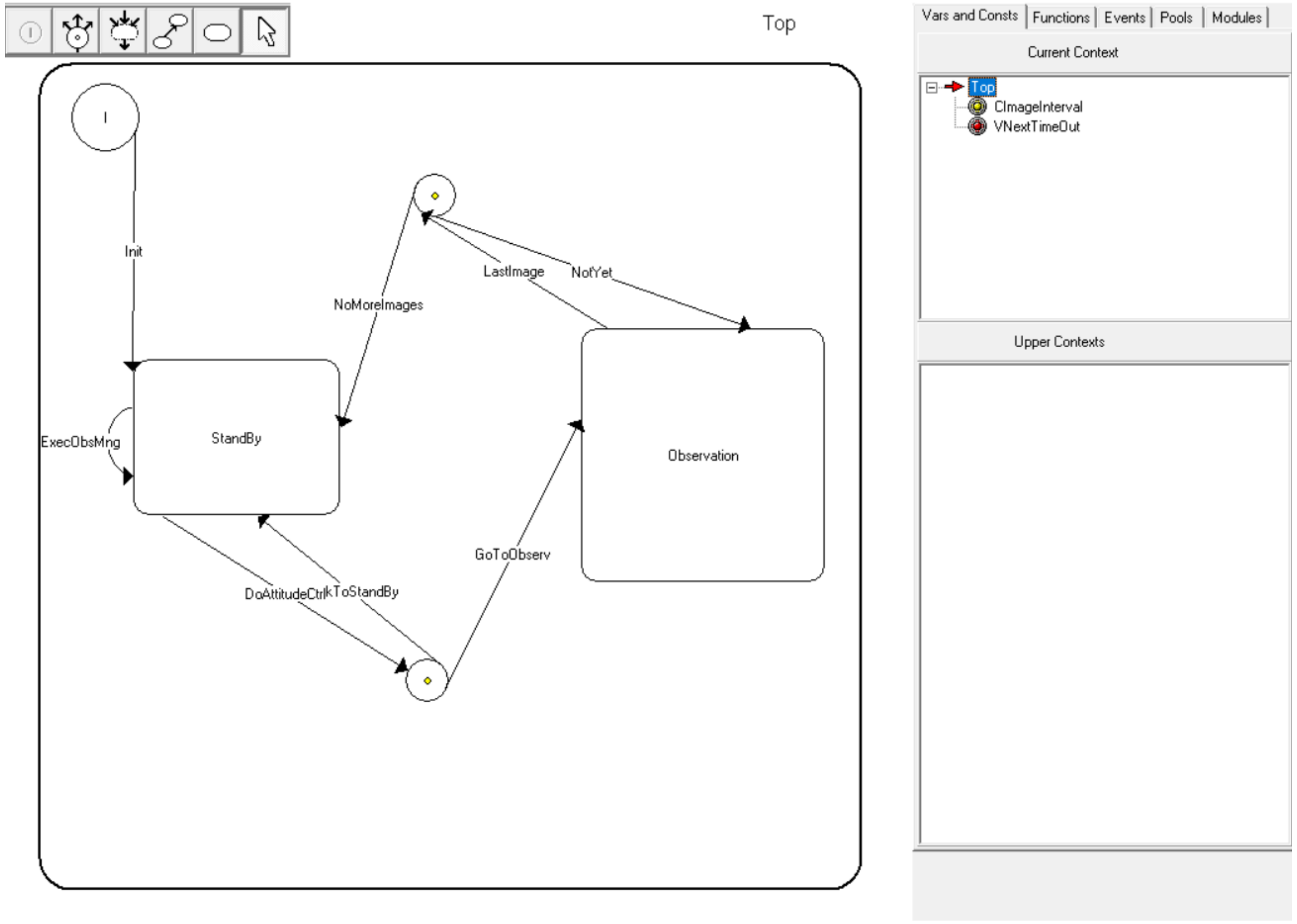
Signal Name: SSObsMng_TC

Data Class: CDTCHandler

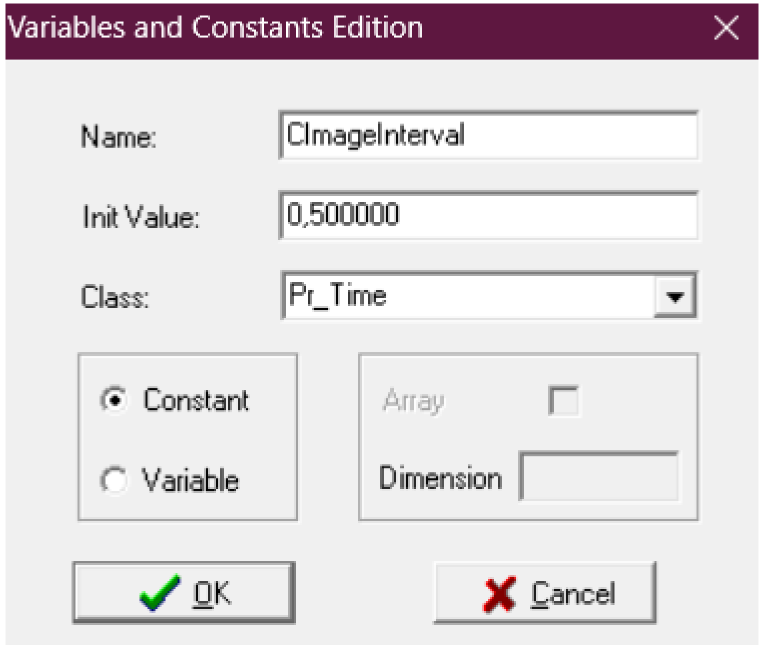
Synchronous Invoke [✓ Asynchronous]

Synchronous Reply To ---->

INTERFAZ CCObsMng



INTERFAZ CCObsMng



Variables and Constants Edition

Name: CImageInterval

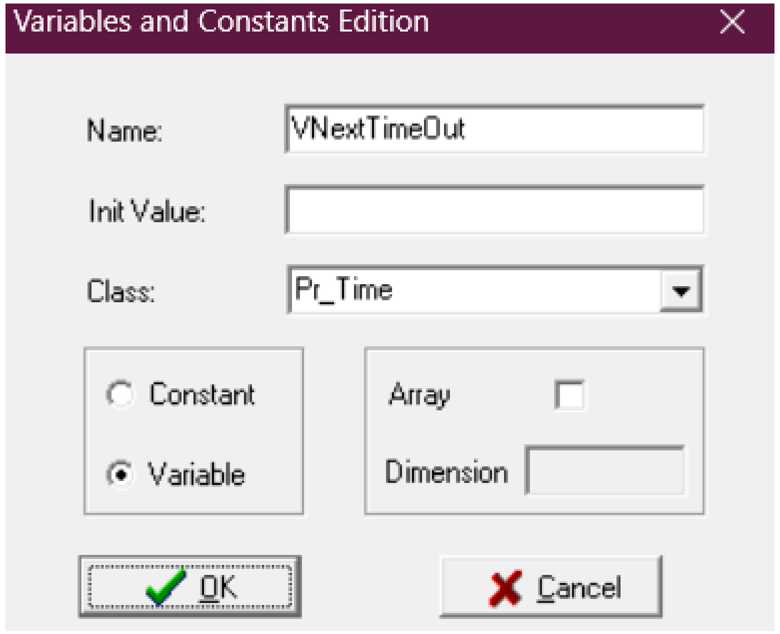
Init Value: 0,500000

Class: Pr_Time

[✓ Constant] Variable

Array ☐

Dimension



Variables and Constants Edition

Name: VNextTimeOut

Init Value:

Class: Pr_Time

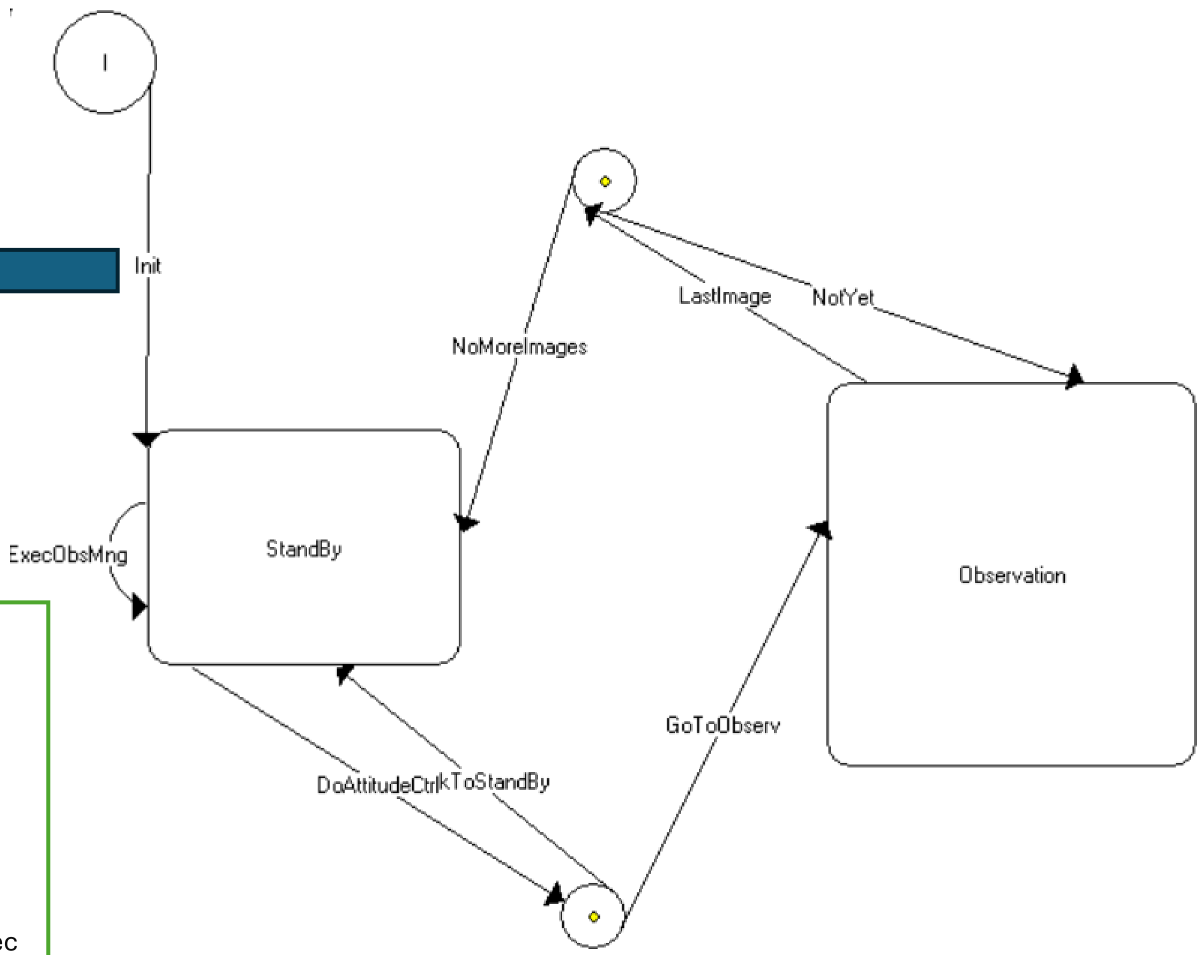
Constant [✓ Variable]

Array ☐

Dimension

INTERFAZ CCObsMng (Init)

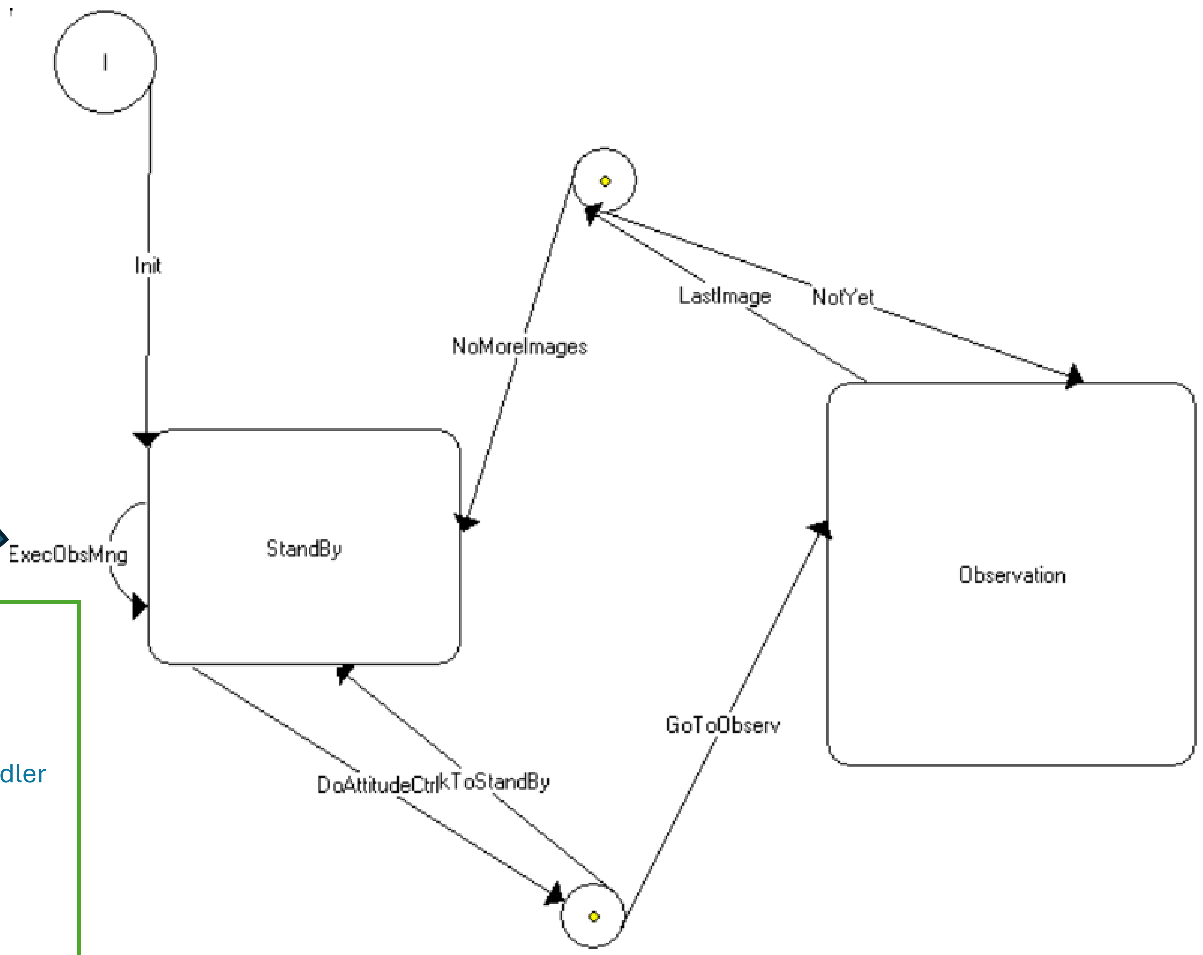
Action Inform At: Finit();



```
{
Pr_Time time;

//Timing Service useful methods

//time.GetTime(); // Get current monotonic time
//time.Add(X,Y); // Add X sec + Y microsec
time.GetTime(); // Get current monotonic time
time+=Pr_Time(0,100000); // Add X sec + Y microsec

AttCtrlTimer.InformAt ( time);
}
```

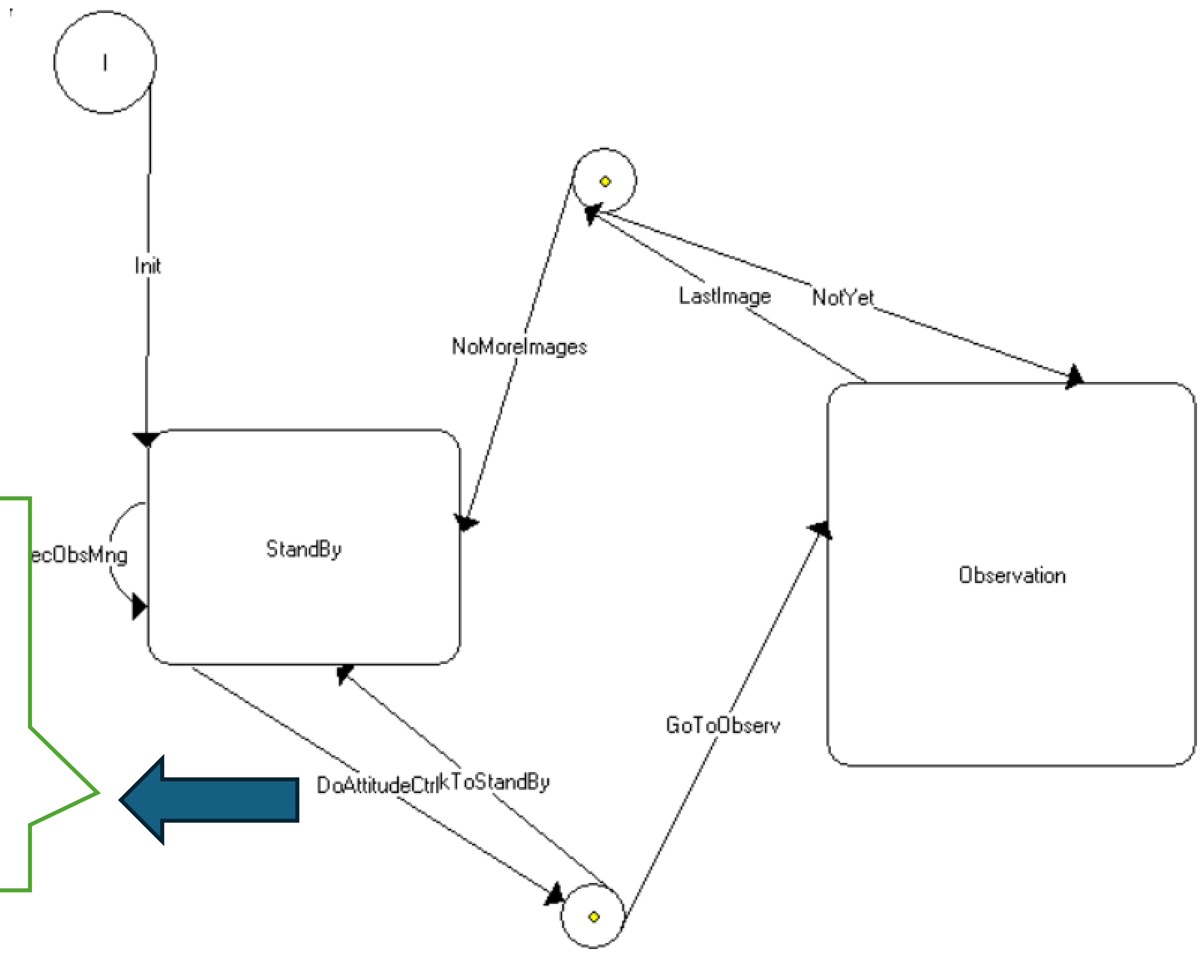
INTERFAZ CCObsMng (ExecObsMng)

Msg->Data H: FExecObsMng_TC;
Trigger:
P:Obs_Mng_Ctrl
S:SSObsMng_TC
G:true



```
{
CDTCHandler & VarSSObsMng_TC=*(CDTCHandler
*)Msg->data;

varSSObsMng_TC.ExecTC();

AttCtrlTimer.InformAt ( time);
}
```

INTERFAZ CCObsMng (DoAttitudeCtrl)



Action: FDoAttitudeCtrl();

Trigger:
P:AttCtrlTimer
S:EDROOMSignalTimeout
G:true

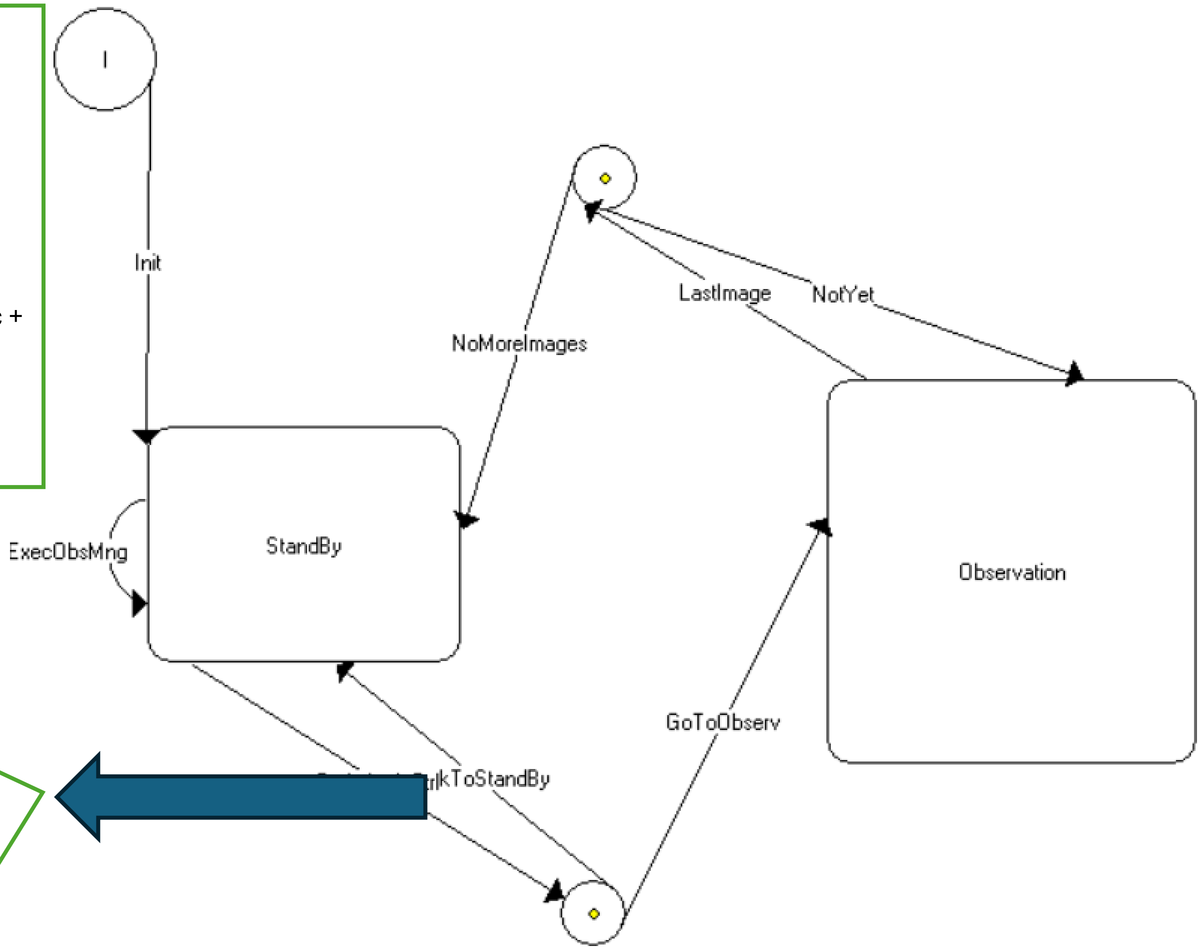
pus_service129_do_attitude_ctrl();

INTERFAZ CCObsMng (BackToStandBy)

```
{
Pr_Time time;


//Timing Service useful methods

//time.GetTime(); // Get current monotonic time
//time.Add(X,Y); // Add X sec + Y microse

VNextTimeOut+= Pr_Time(0,10000); // Add X sec +
Y microsec
time=VNextTimeOut

AttCtrlTimer.InformAt ( time);
}
```

Action: FProgAttitudeCtrl();

Trigger:
P:AttCtrlTimer
S:EDROOMSignalTimeout
G:true

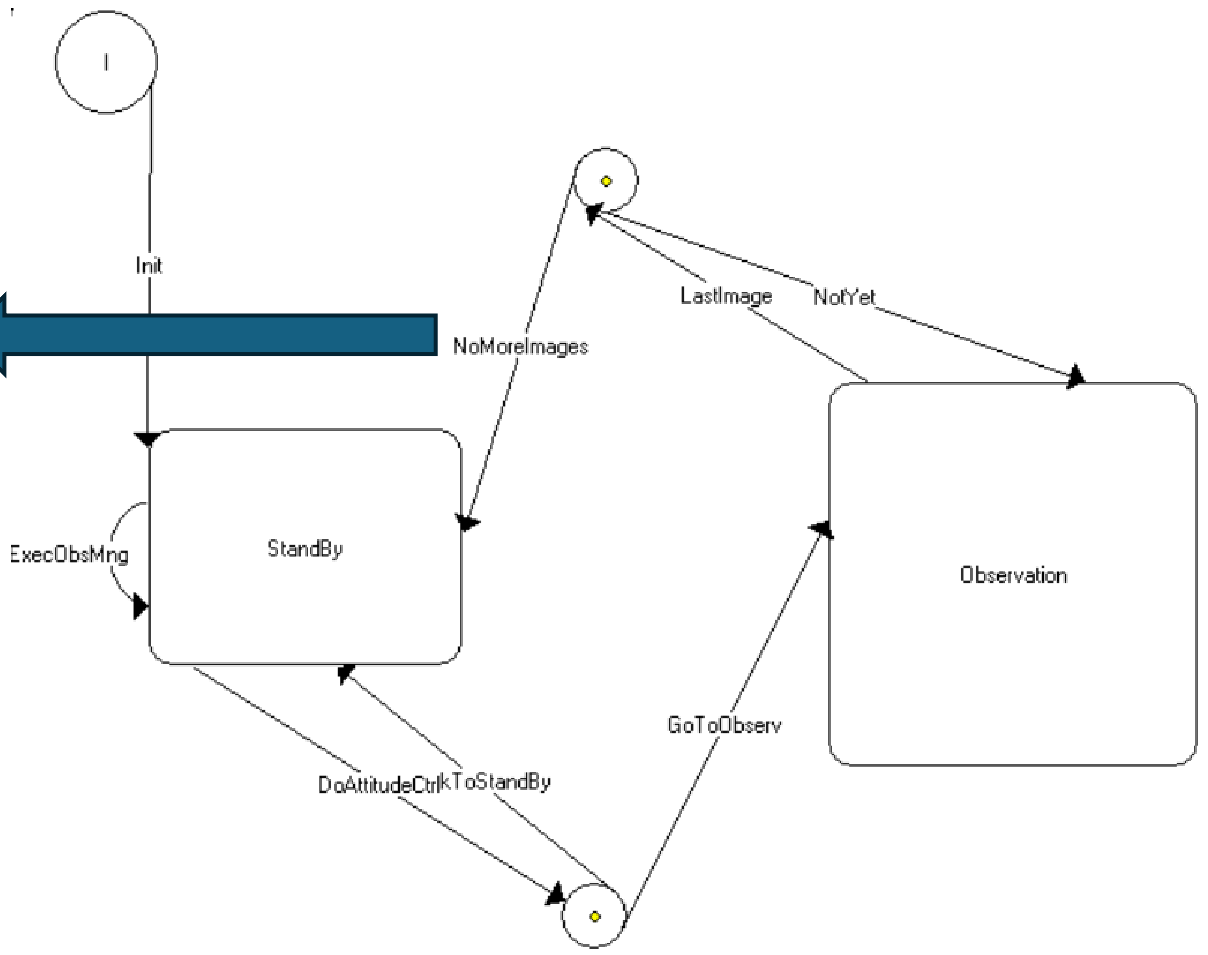


INTERFAZ CCObsMng (NoMoreImages)

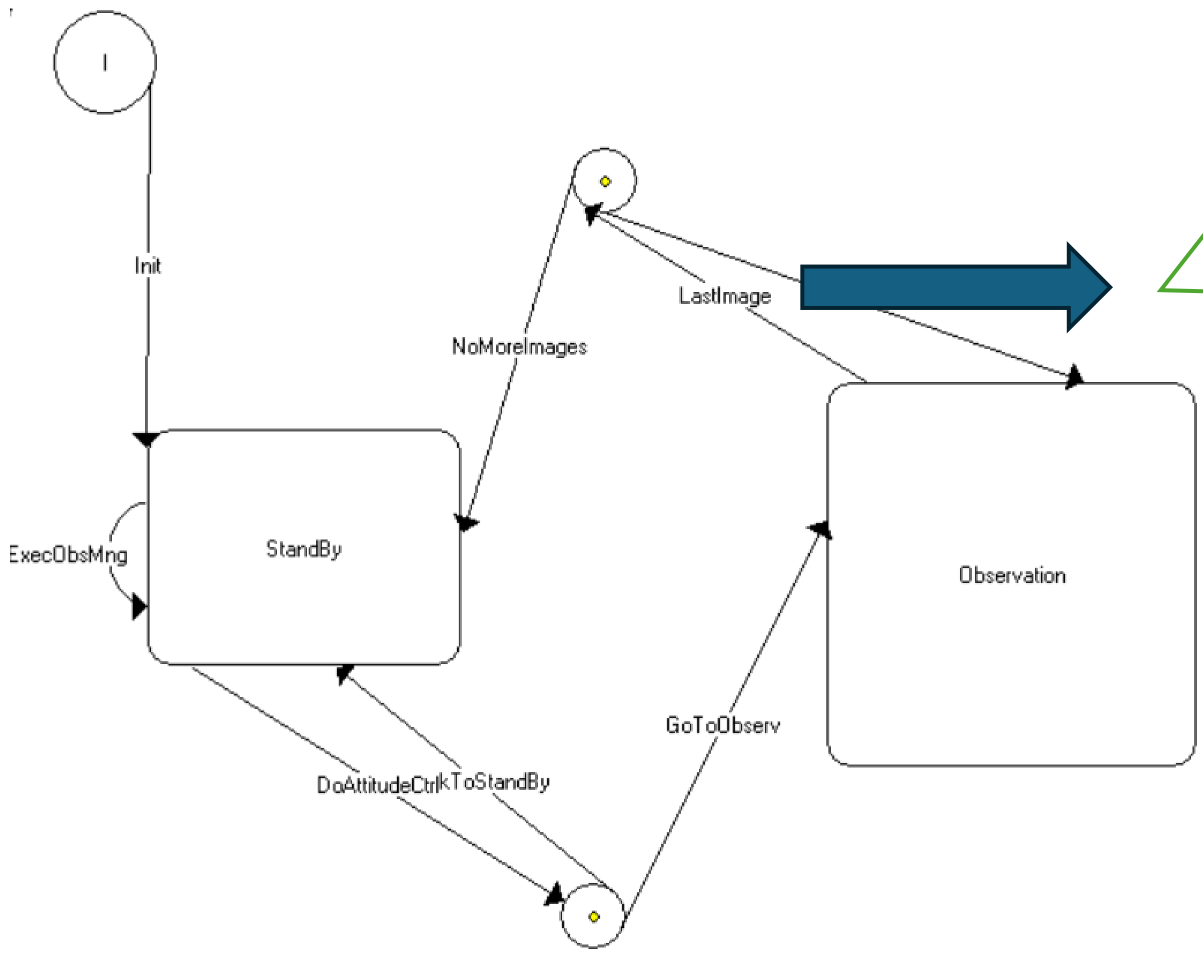
Composite Action:
FCEndObs();
G:GLastImage();
Action:FEndObservation();
Action:FProgAttitudeCtrl();

VNextTimeOut.GetTime();

```
{
Pr_Time time;

//Timing Service useful methods

//time.GetTime(); // Get current monotonic time
//time.Add(X,Y); // Add X sec + Y microse

VNextTimeOut+= Pr_Time(0,10000); // Add X sec + Y microsec
time=VNextTimeOut

AttCtrlTimer.InformAt ( time);
}
```



INTERFAZ CCObsMng (LastImage)

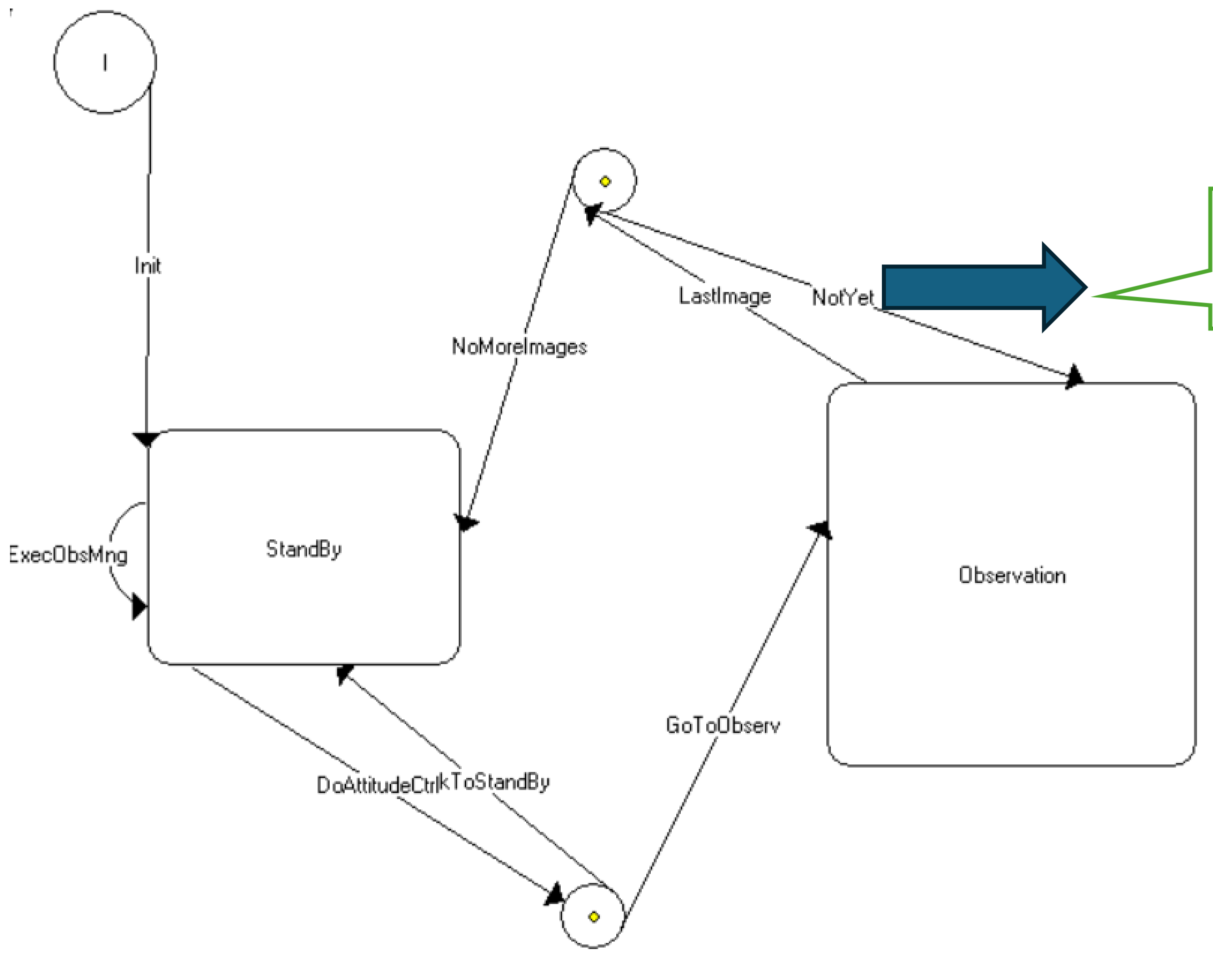


Action: FTakeImage();

Trigger:
P:Observtimer
S:EDROOMSignalTimeout
G:true

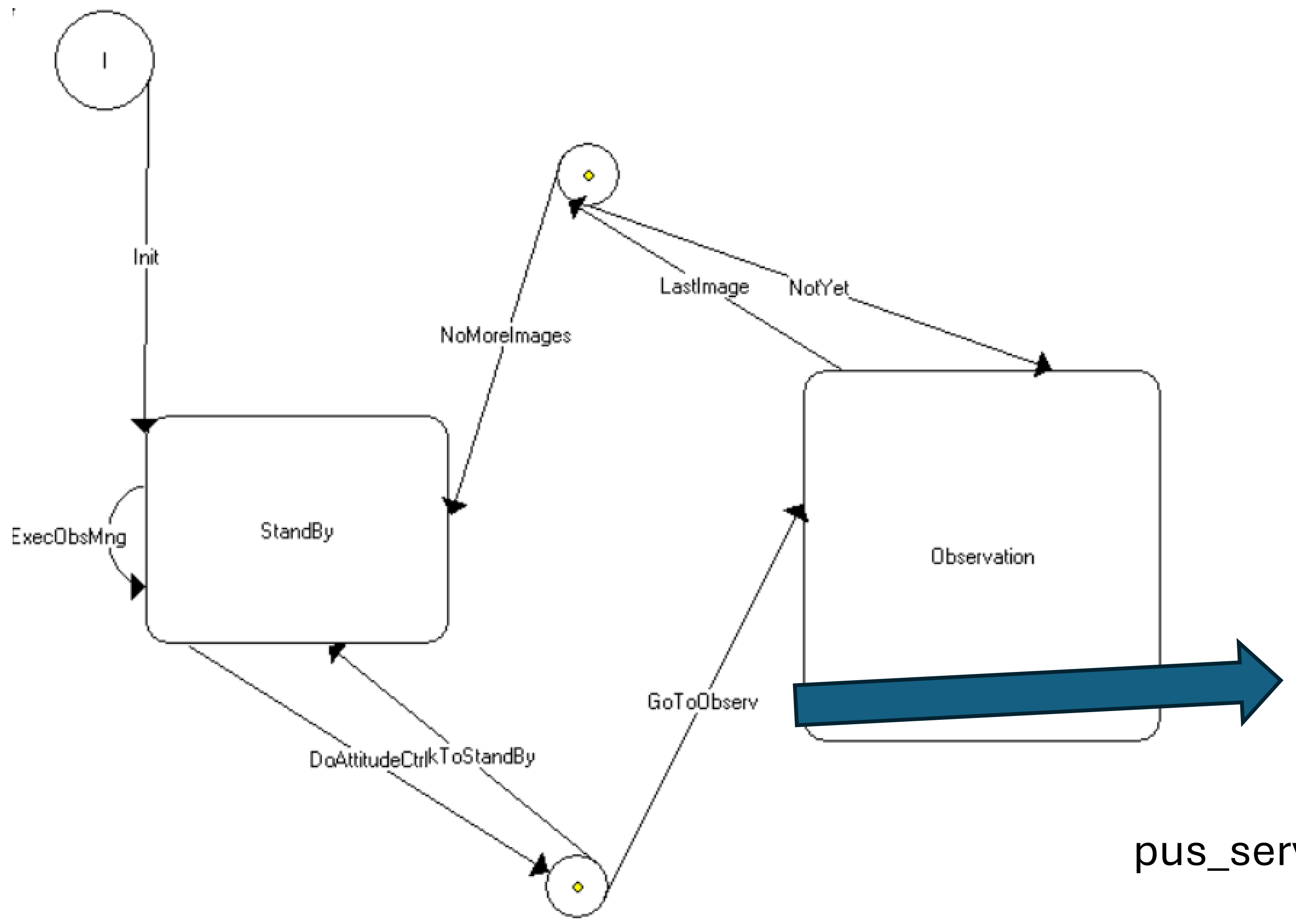
pus_service129_take_image();

INTERFAZ CCObsMng (NotYet)



G:true

INTERFAZ CCObsMng (GoToObserv)



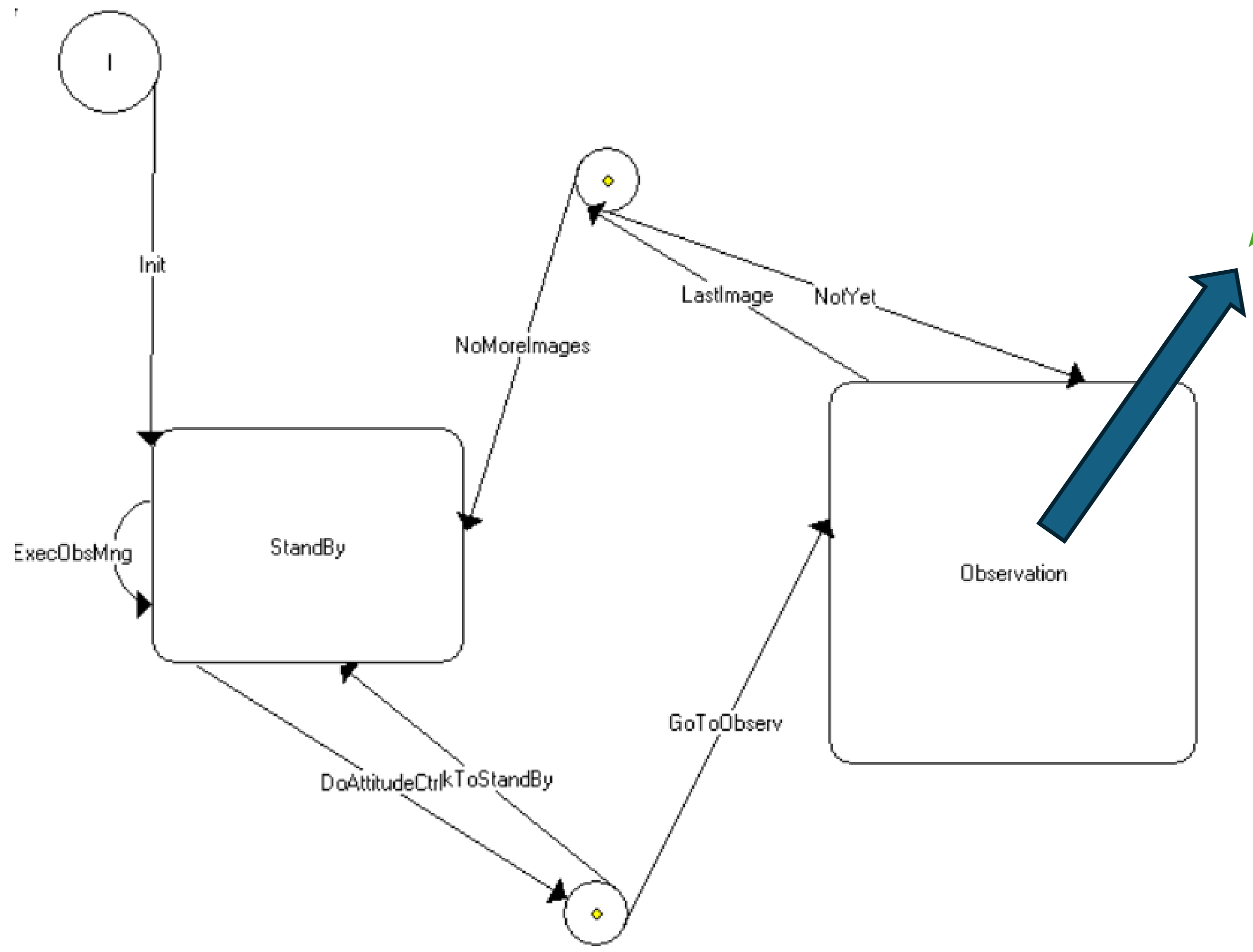
Action:
FToObservation();

Trigger:
G:GreadyToObservation()

pus_service129_start_observation();

return pus_service129_is_observation_ready();

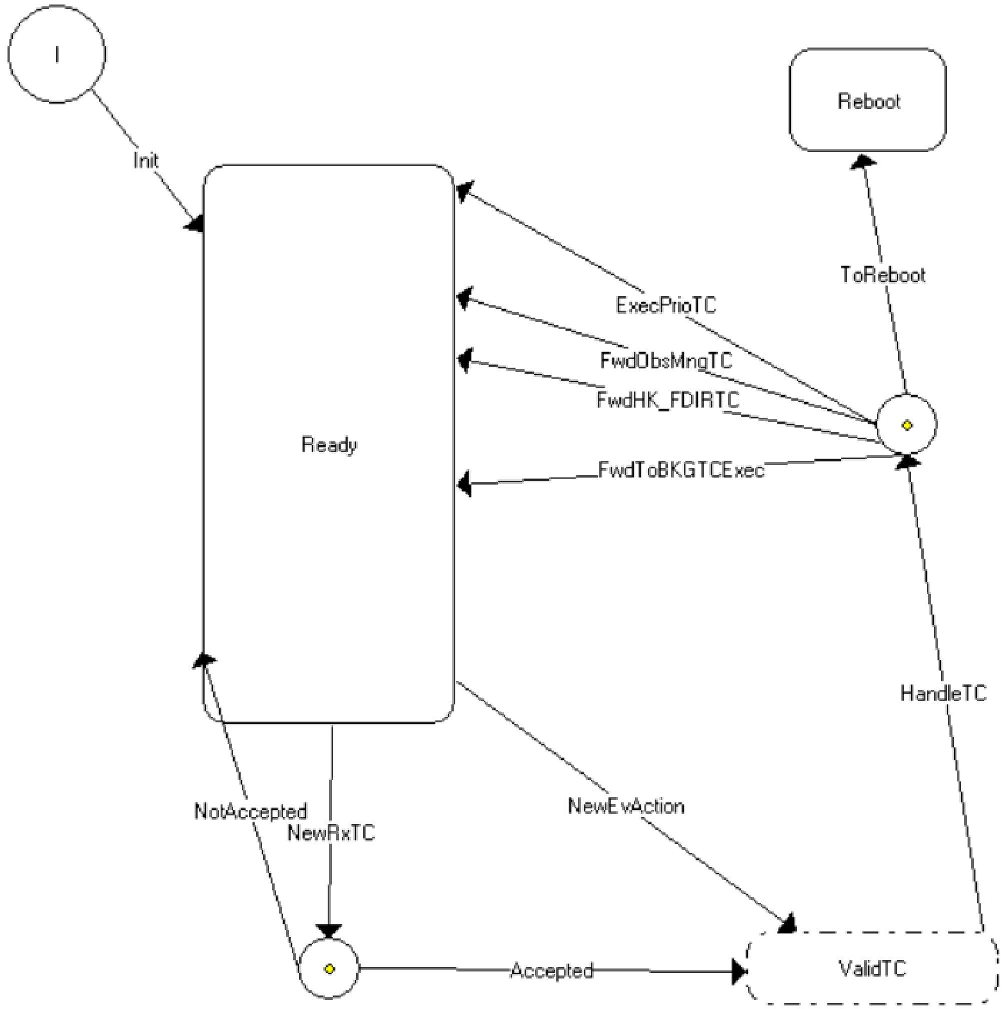
INTERFAZ CCObsMng (Observation)



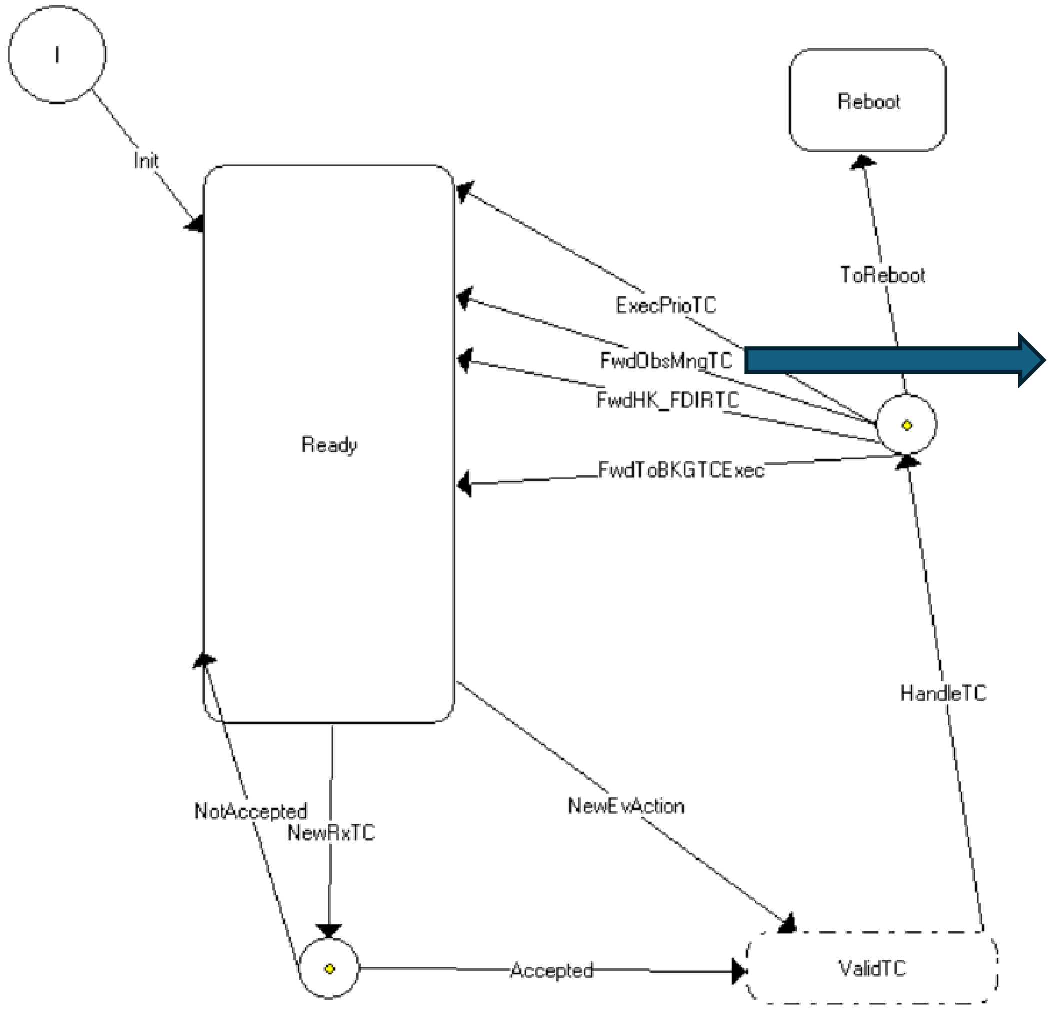
Inform In: FProgTakeImage;
Trigger:
P:Obs_Mng_Ctrl
S:SSObsMng_TC
G:true

```
{
Pr_Time interval;

//Timing Service useful methods

//time.GetTime(); // Get current monotonic time
//time.Add(X,Y); // Add X sec + Y microsec
interval=CImageInterval;

        OBservTimer.InformIn( interval);
}
```

INTERFAZ CCTCManager



INTERFAZ CCTCManager (FwdObsMngTC)



Send: FFwdObsMngTC;
Trigger:
P:Obs_Mng_Ctrl
S:SSObsMng_TC
G:GFwdObsMngTC()

return VTCExecCtrl.IsObsMng();

```
{
CDTCHandler * pSSObsMng_TC_Data =
EDROOMPoolCDTCHandler.AllocData();

// Complete Data

*pSSObsMng_TC_Data=VCurrentTC;

ObsMng.send(SSObsMng_TC,pSSO
bsMng_TC_Data,&EDROOMPoolCD TCHandler);
}
```