

ENTREGABLES 1,2,3,4

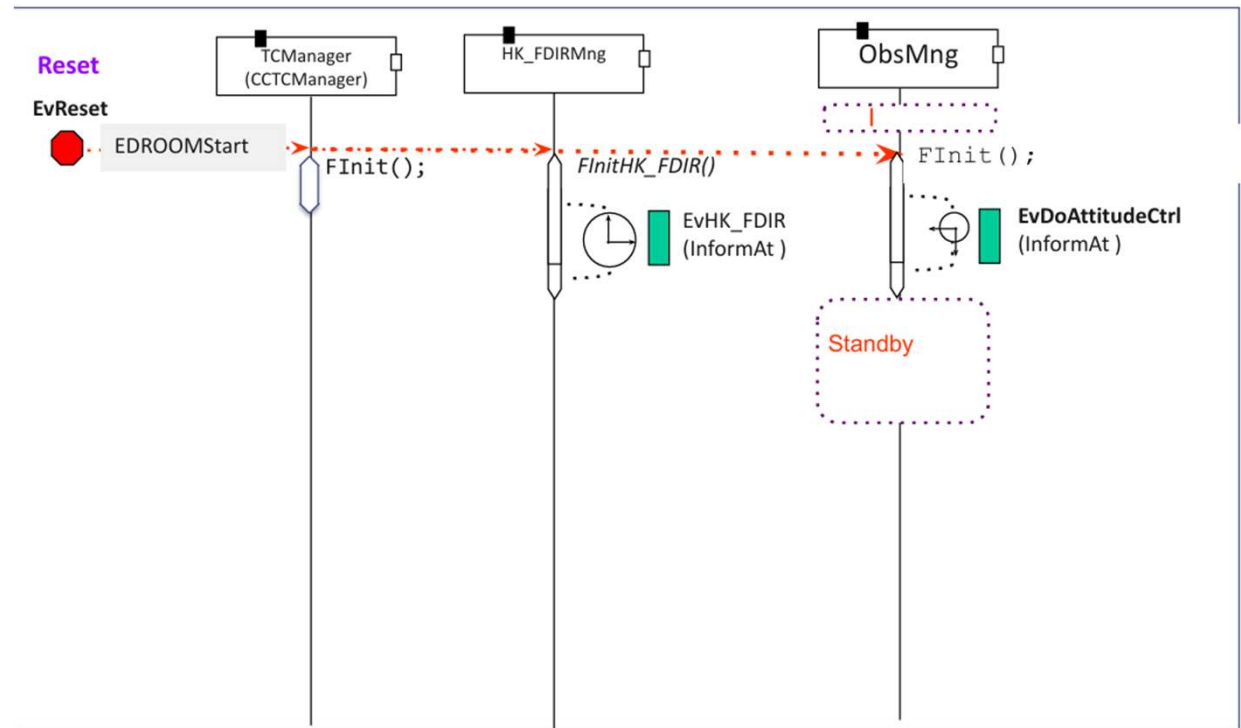
Juan Pedro López Fiestas

OBDH

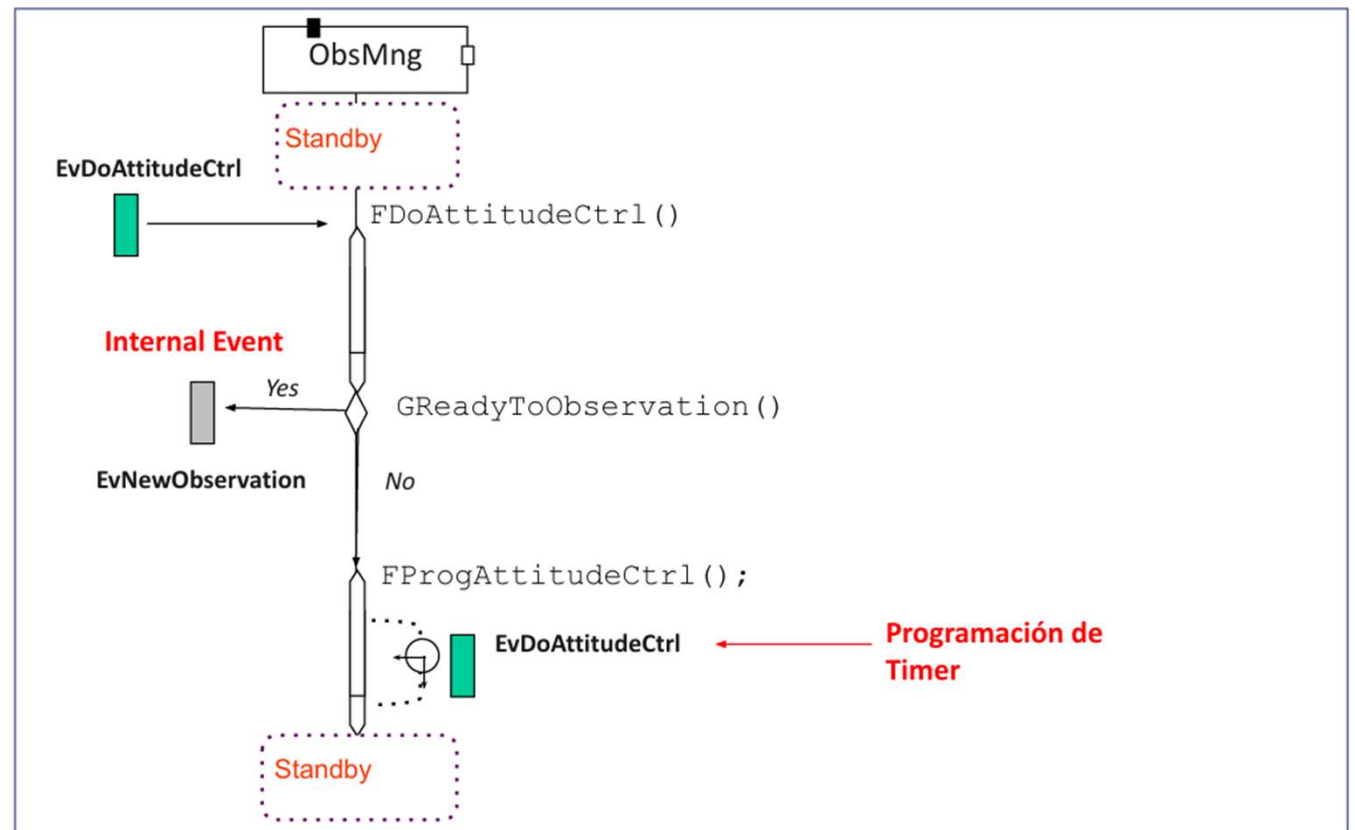
ENTREGABLE 1

Escenarios en los que está presente ObsMng

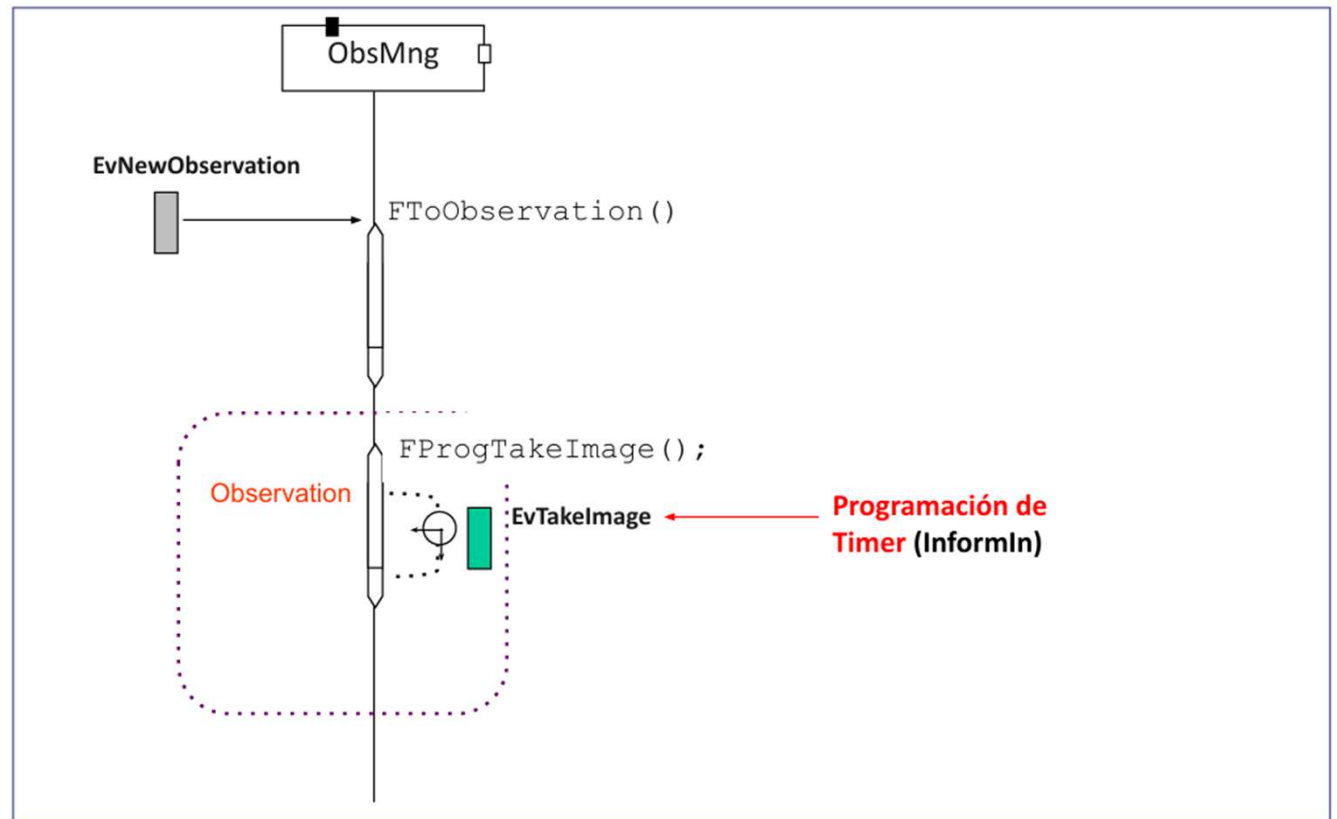
Escenario donde vemos la Finit() (inicio del edroom)



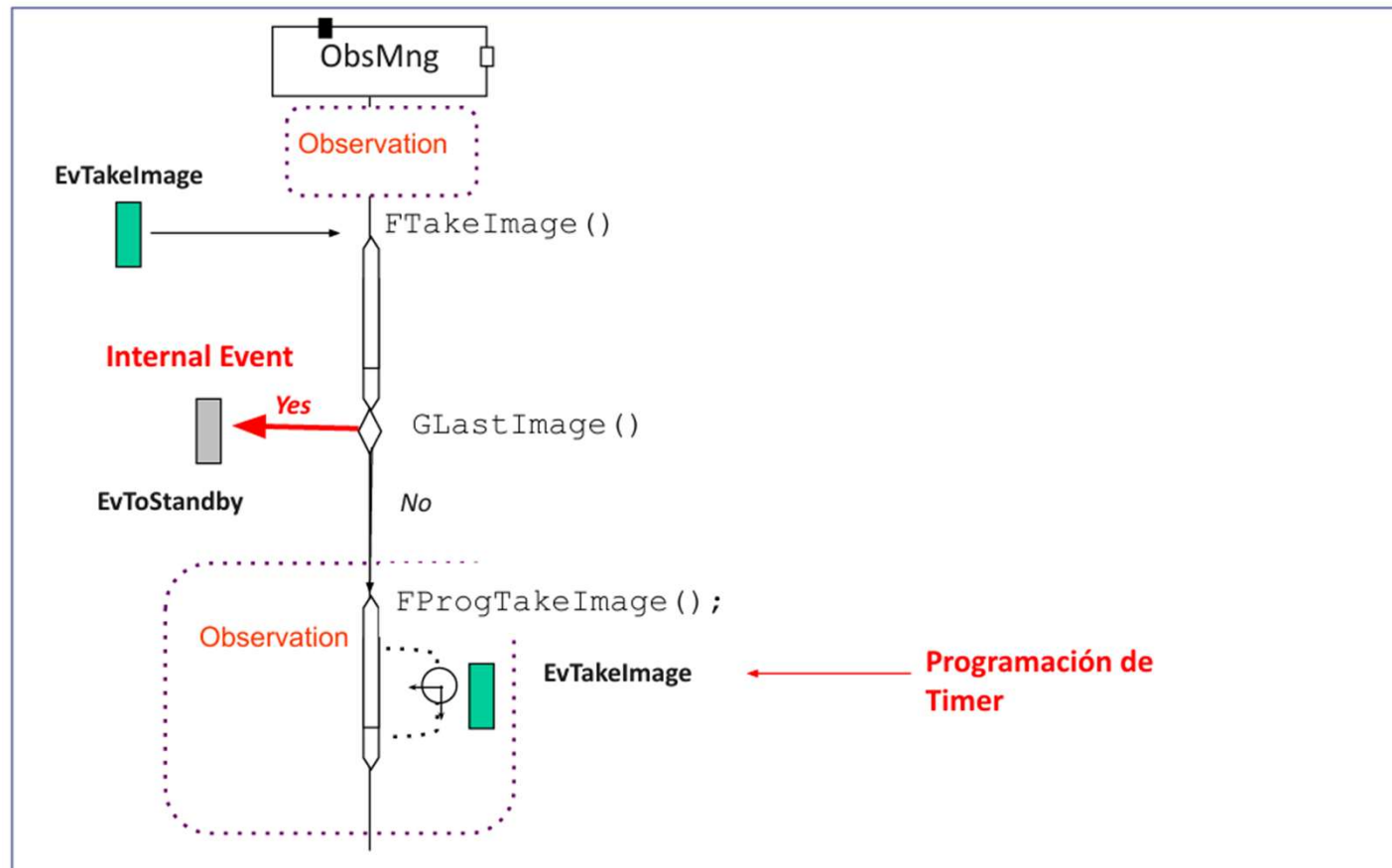
Primer evento EvDoAttitudeCtrl



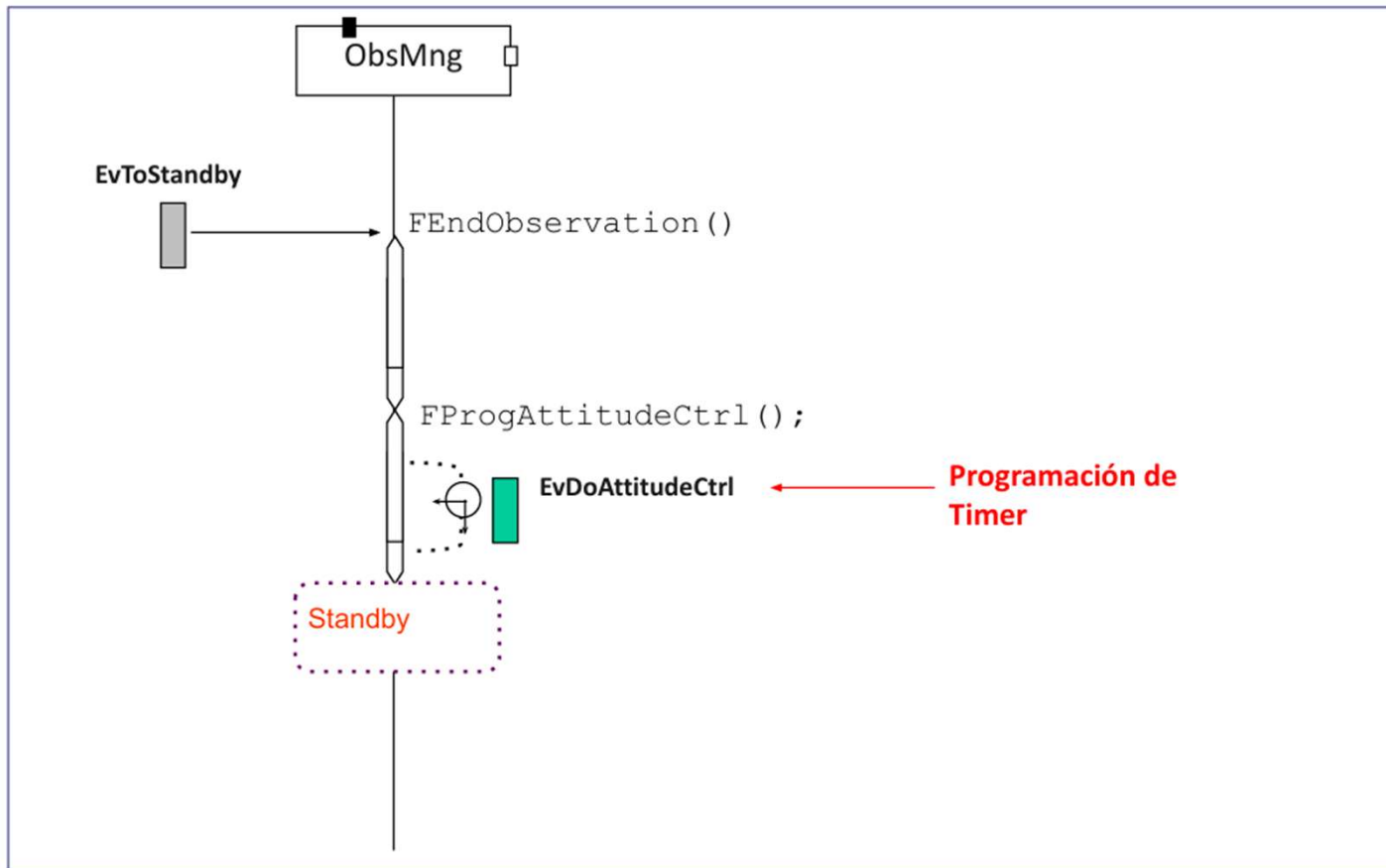
Evento interno EvNewObservation



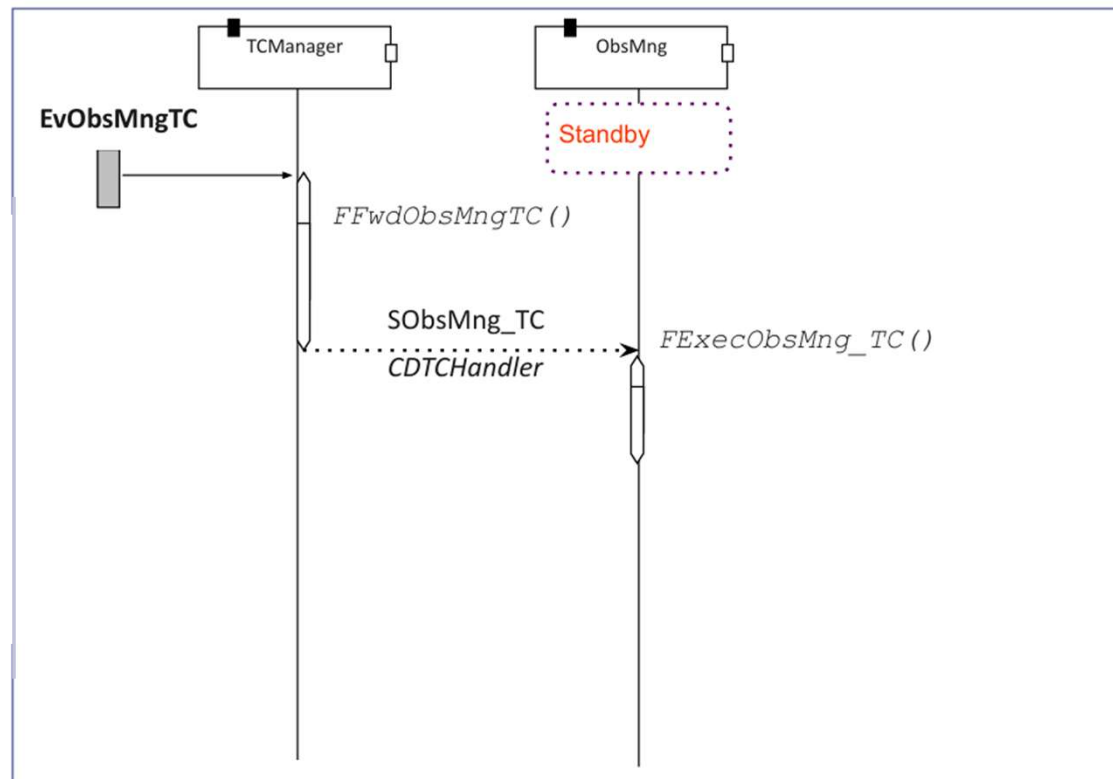
Evento en observation de EvTakeImage



Evento interno EvToStandby



Último escenario con la función FexecObsMng_TC



ENTREGABLE 2

Definición de la clase protocolo

Protocol Class Edition

Name:

Design | Analysis

Input Messages :

SSObsMng_TC

Output Messages:

Protocol Brief

Message Edition Box

Signal Name:

Data Class :

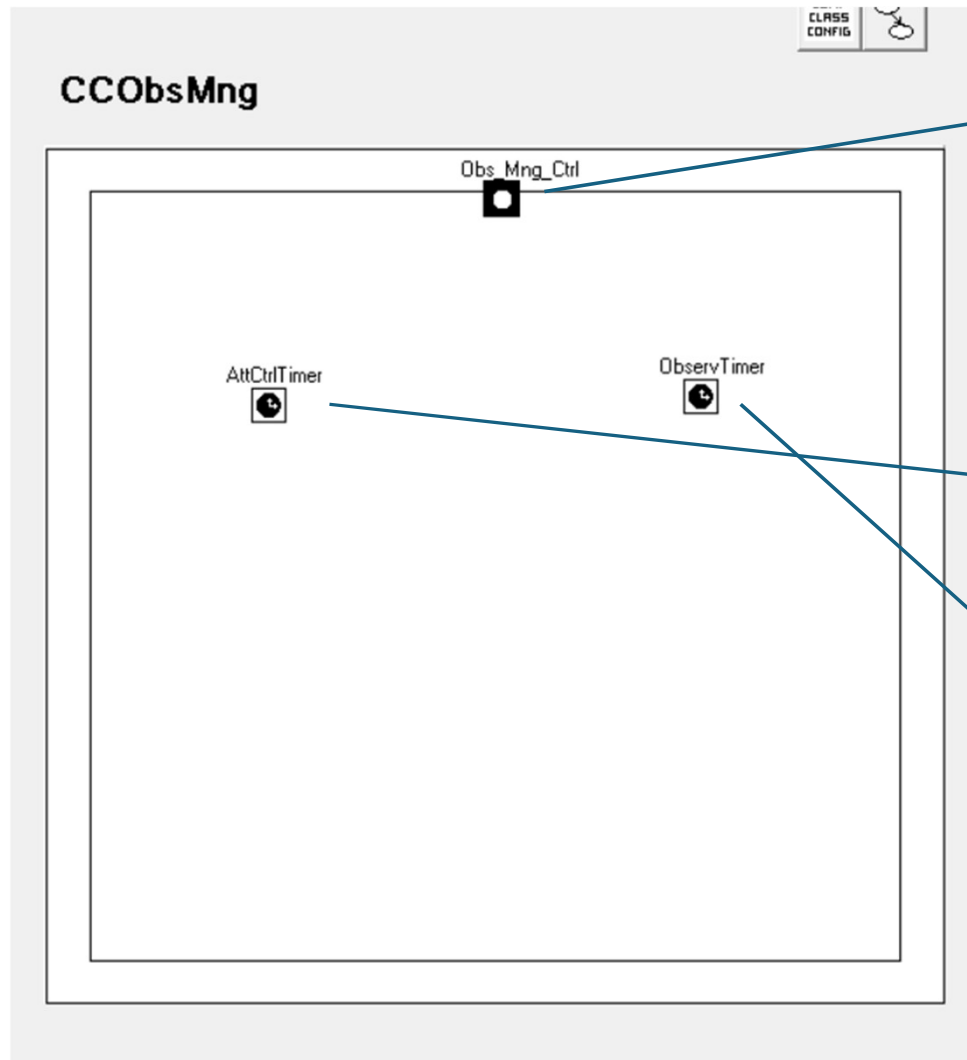
☐ Synchronous Invoke
 ☒ Asynchronous

☐ Synchronous Reply To ---->

	In/Out	Signal	Data
Msg 	In	SSObsMng_TC	CDTCHandler

ENTREGABLE 3

Diseño de la interfaz de la CCObsMng



ObsMngCtrl

Tipo externo no conjugado
Puerto de comunicaciones
CPObs_Mng_Ctrl

ObservTimer

Tipo interno temporal
EDROOMTimingSAP

AttCtrlTimer

Tipo interno temporal
EDROOMTimingSAP

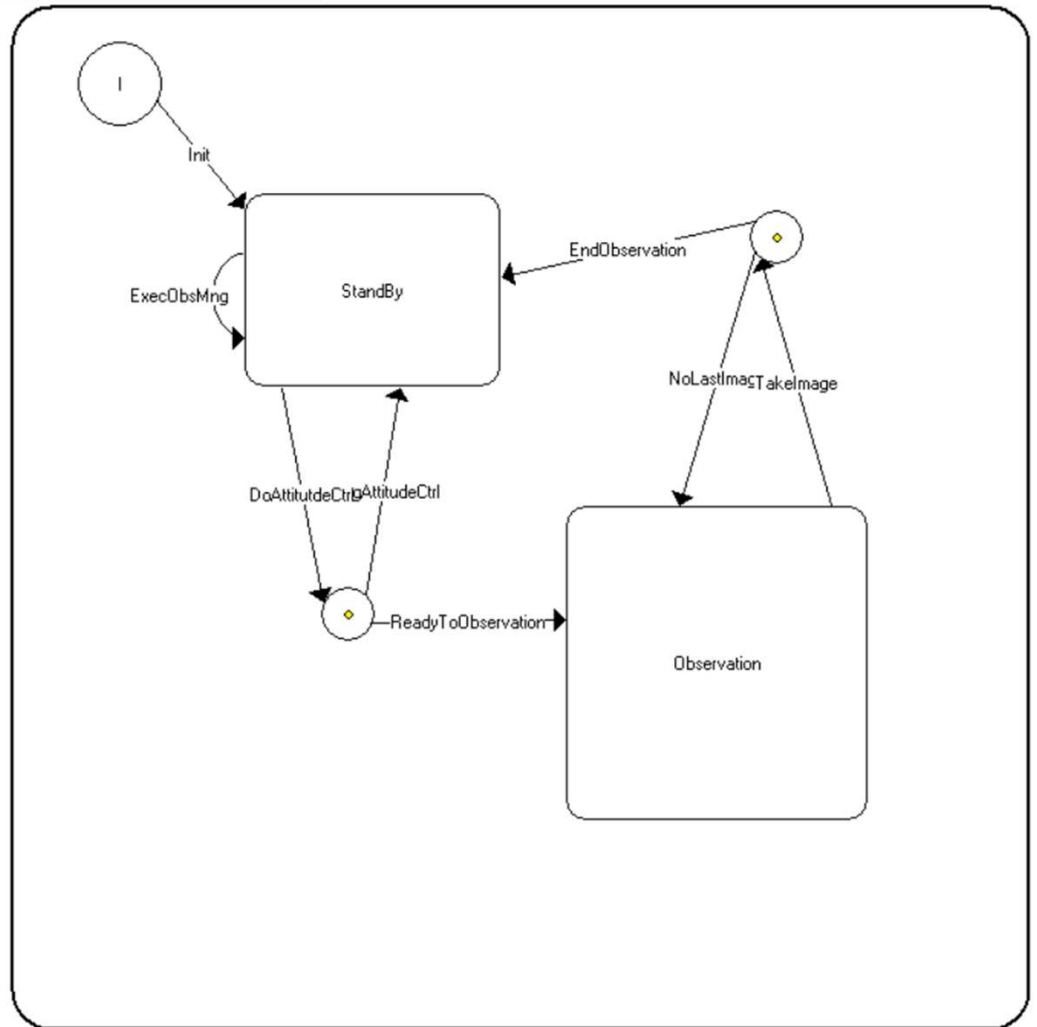
ENTREGABLE 4

Diseño del comportamiento de CCObsMng

Máquina de estados



Top



Variables y constantes

Variables and Constants Edition

Name:

Init Value:

Class:

☒ Constant

☐ Variable

Array ☐

Dimension

☒ OK ☐ Cancel

Variables and Constants Edition

Name:

Init Value:

Class:

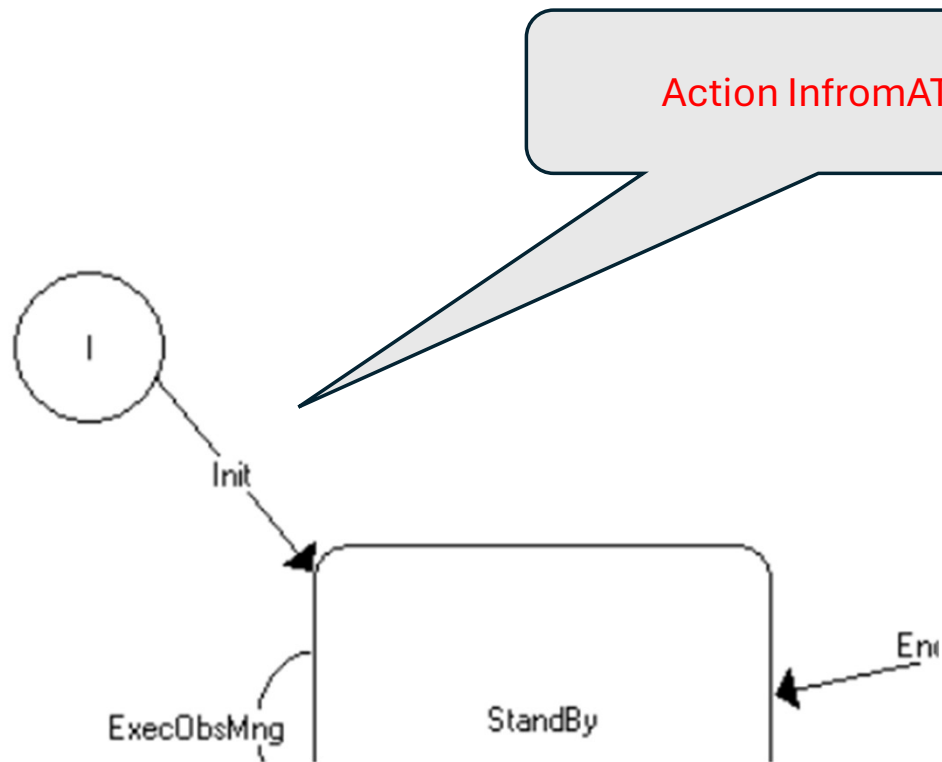
☐ Constant

☒ Variable

Array ☐

Dimension

☐ OK ☒ Cancel



Action InfromAT: **Finit();**

```
//Timing Service useful methods

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


Declaration: FExecObsMng_TC()

Brief

```

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

    varSSObsMng_TC.ExecTC();
}

```

Standard Library Includes

EDROOM Service

Msg->data

Port

Obs_Mng_Ctrl

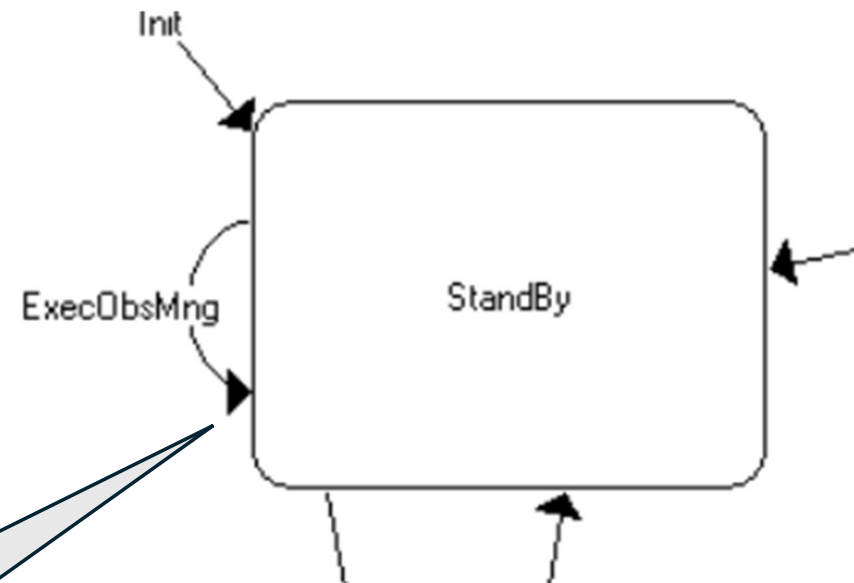
Signal

SSObsMng_TC

Data Class

CDTCHandler

Service Request



Msg->Data H : FExecObsMng_TC();

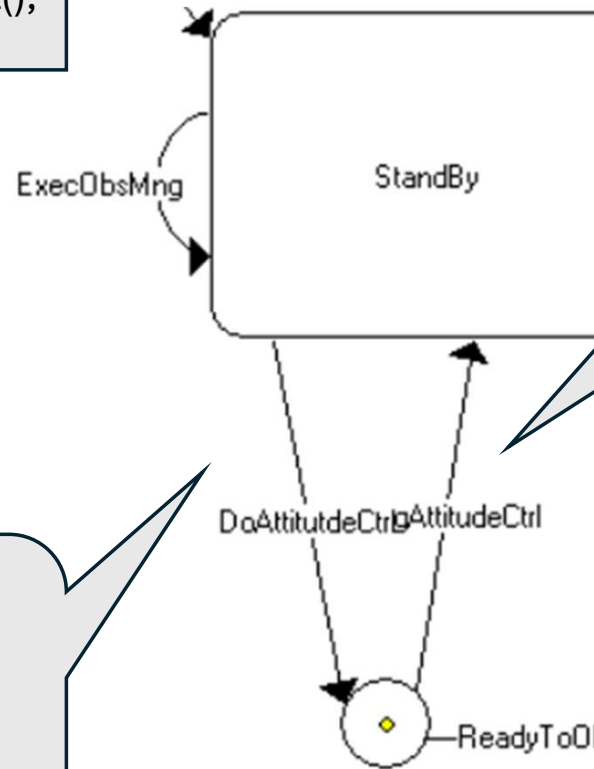
Trigger:

P: Obs_Mng_Ctrl

S: SSObsMng_TC

G: true

pus_service129_do_attitude_ctrl();



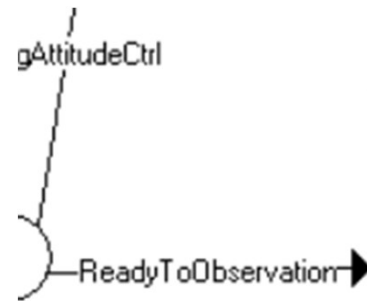
Action: FDoAttitudeCtrl();
Trigger:
P: AttCtrlTimer
S: EDROOMSignalTimeout
G: true

Action InformAt: FProgAttitudeCtrl();
Trigger:
P: AttCtrlTimer
S: EDROOMSignalTimeout
G: true

//Timing Service useful methods

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

VNextTimeOut+=Pr_Time(0,100000);
time=VNextTimeOut;



Action: FToObservation();
G: GReadyToObservation()

pus_service129_start_observation();

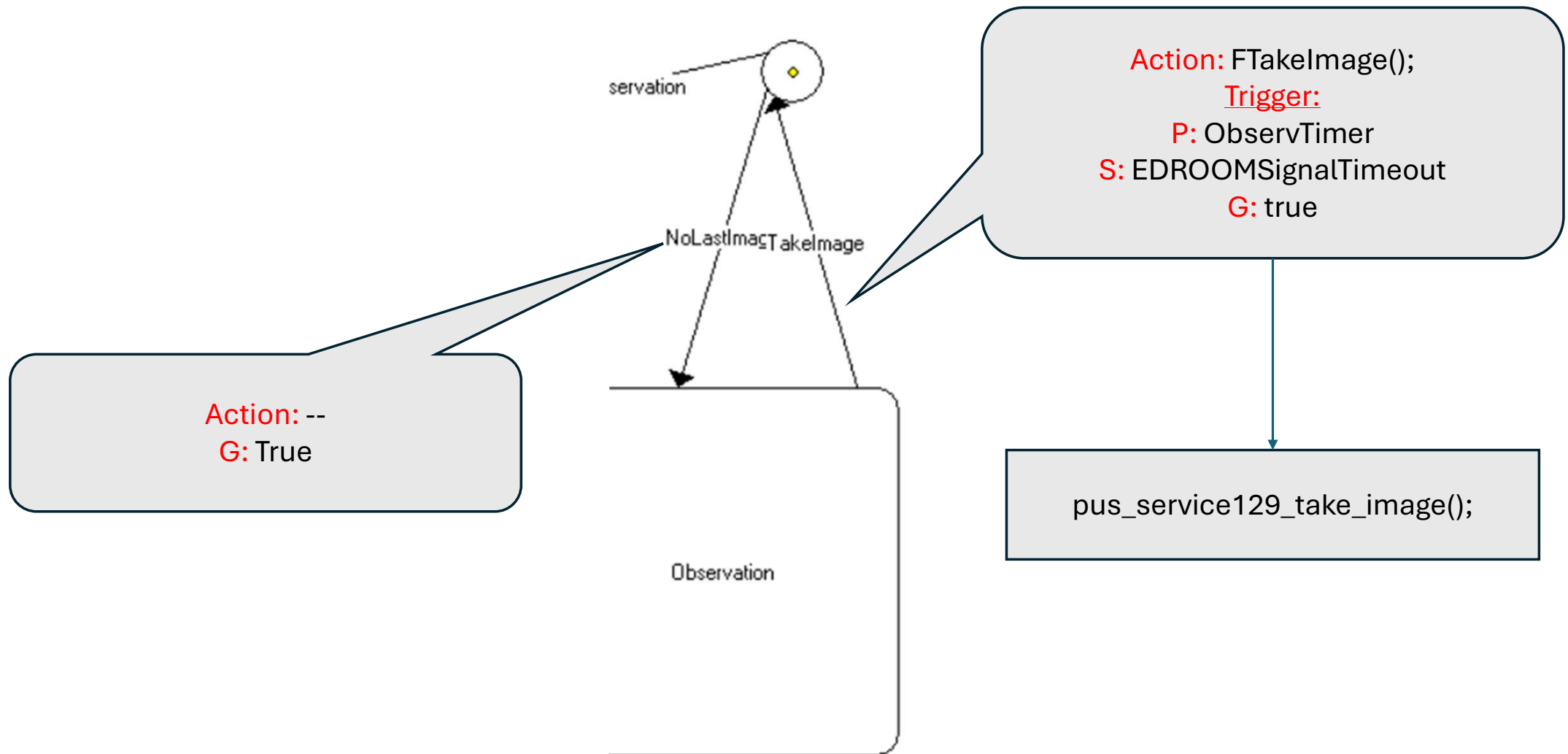
Entry Action: FProgTakeImage();

//Timing Service useful methods

//time.GetTime(); // Get current monotonic time

//time.Add(X,Y); // Add X sec + Y microsec

interval=CImageInterval;



Action: FCEndObservation();
G: GLastImage()

FCEndObservation();

FEndObservation();

FProgAttitudeCtrl();

VNextTimeOut.GetTime();

