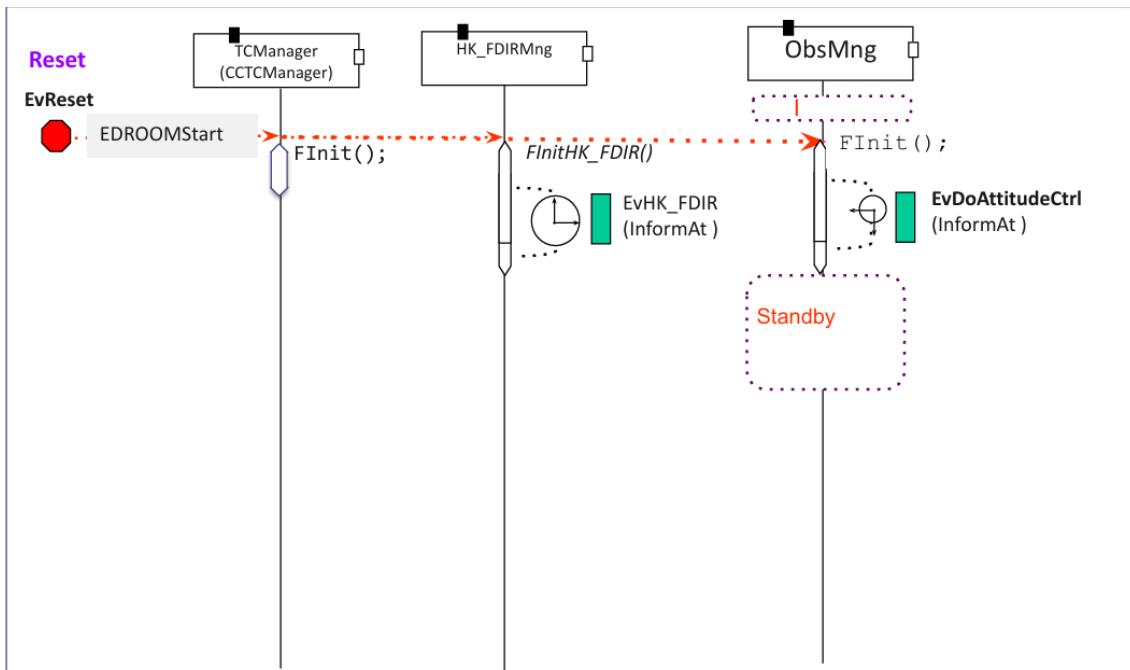
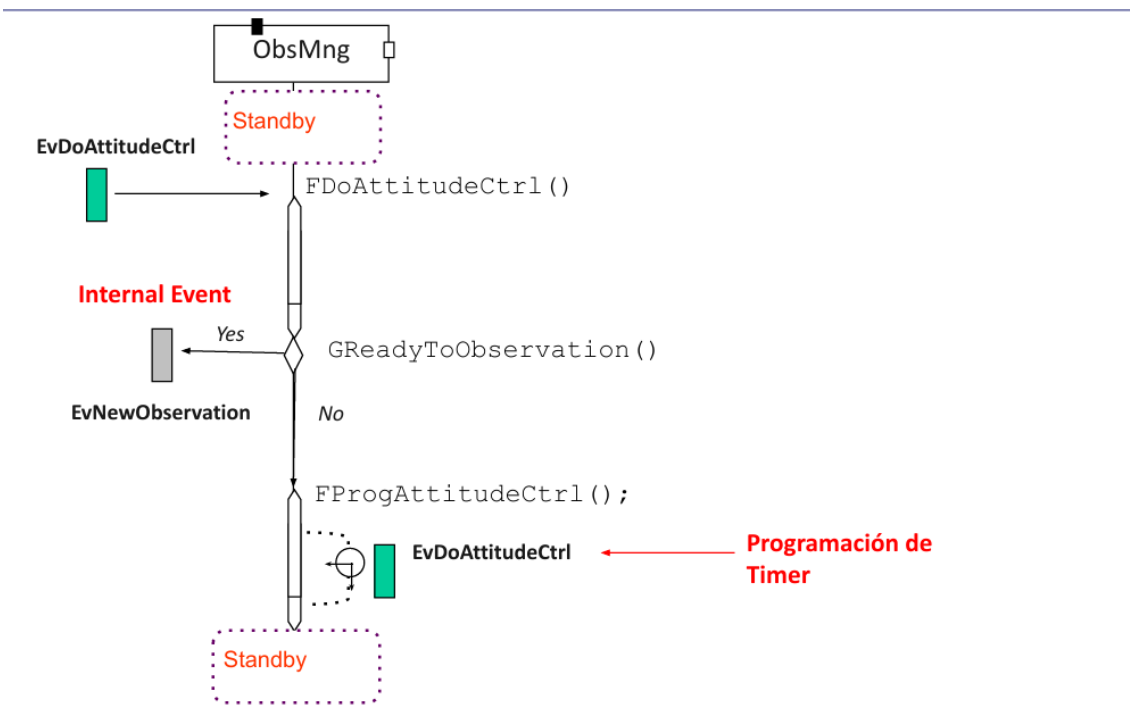


ENTREGABLE 1: ESCENARIOS INVOLUCRADOS DEL NUEVO COMPONENTE OBSMNG

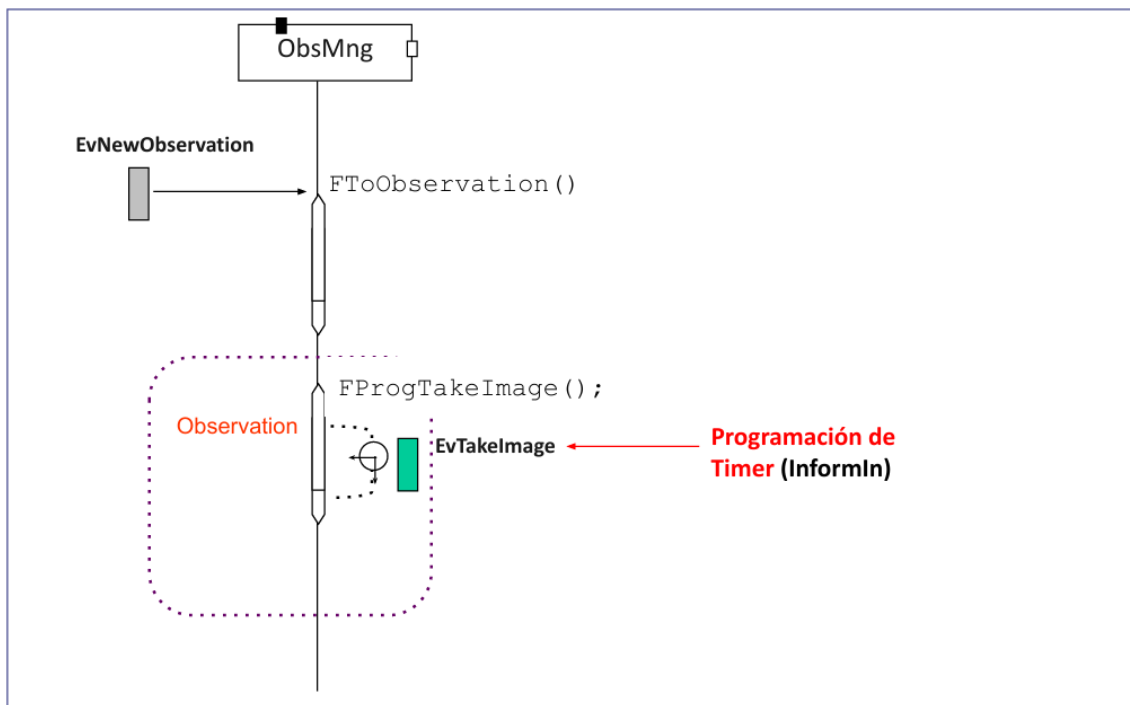
PRIMER ESCENARIO: EVRESET



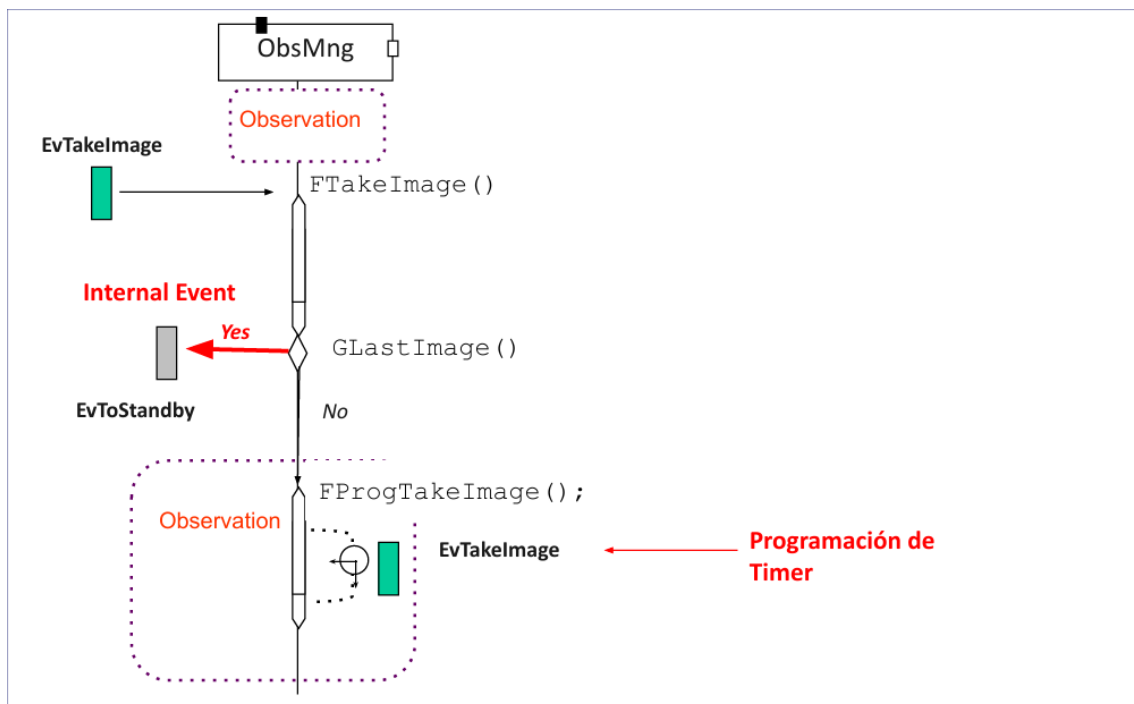
SEGUNDO ESCENARIO: EVDOATTITUDECTRL



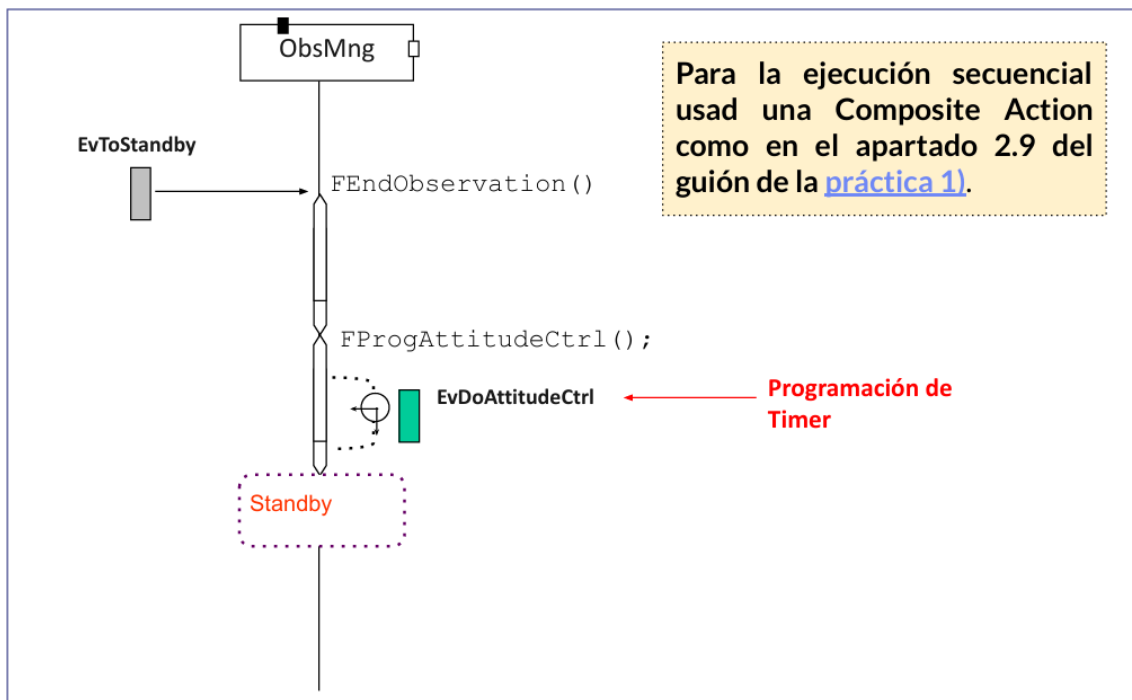
TERCER ESCENARIO: EVNEWOBSERVATION



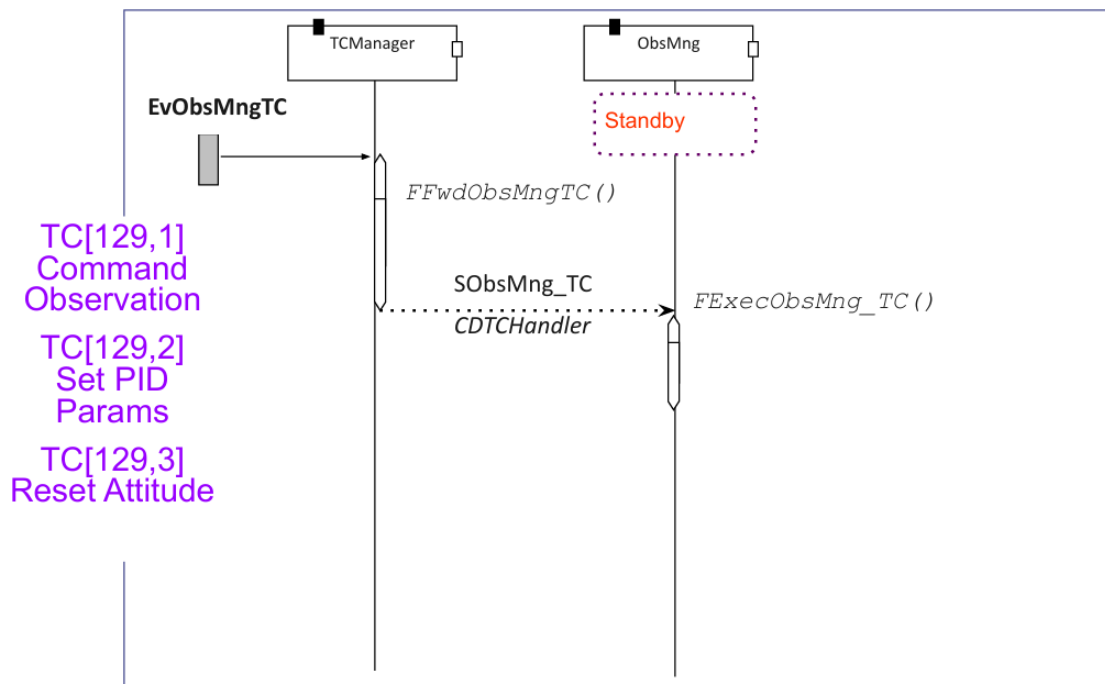
CUARTO ESCENARIO: EVTAKEIMAGE



QUINTO ESCENARIO: EVTOSTANDBY

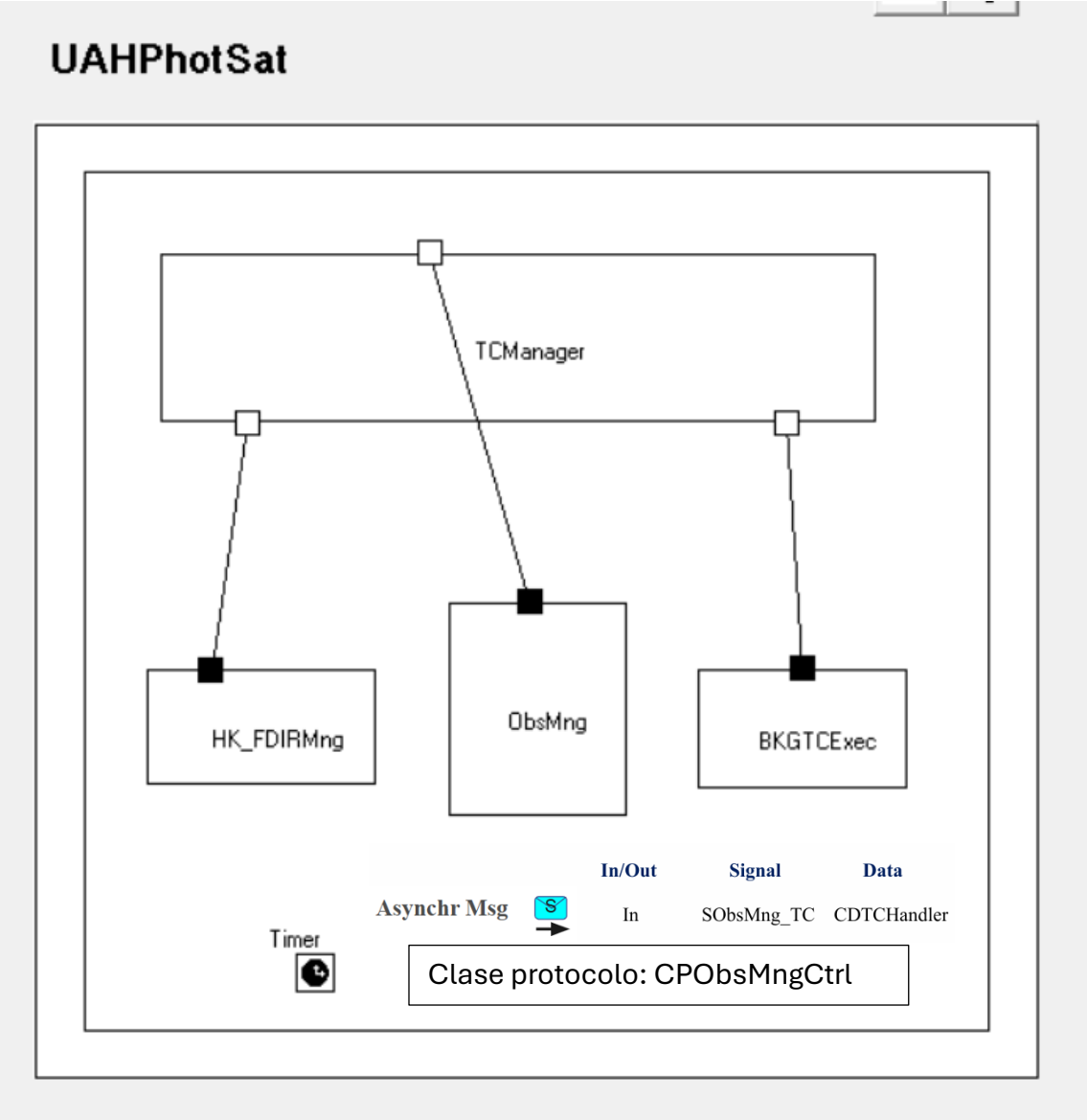


ÚLTIMO ESCENARIO: EVOBSMNGTC

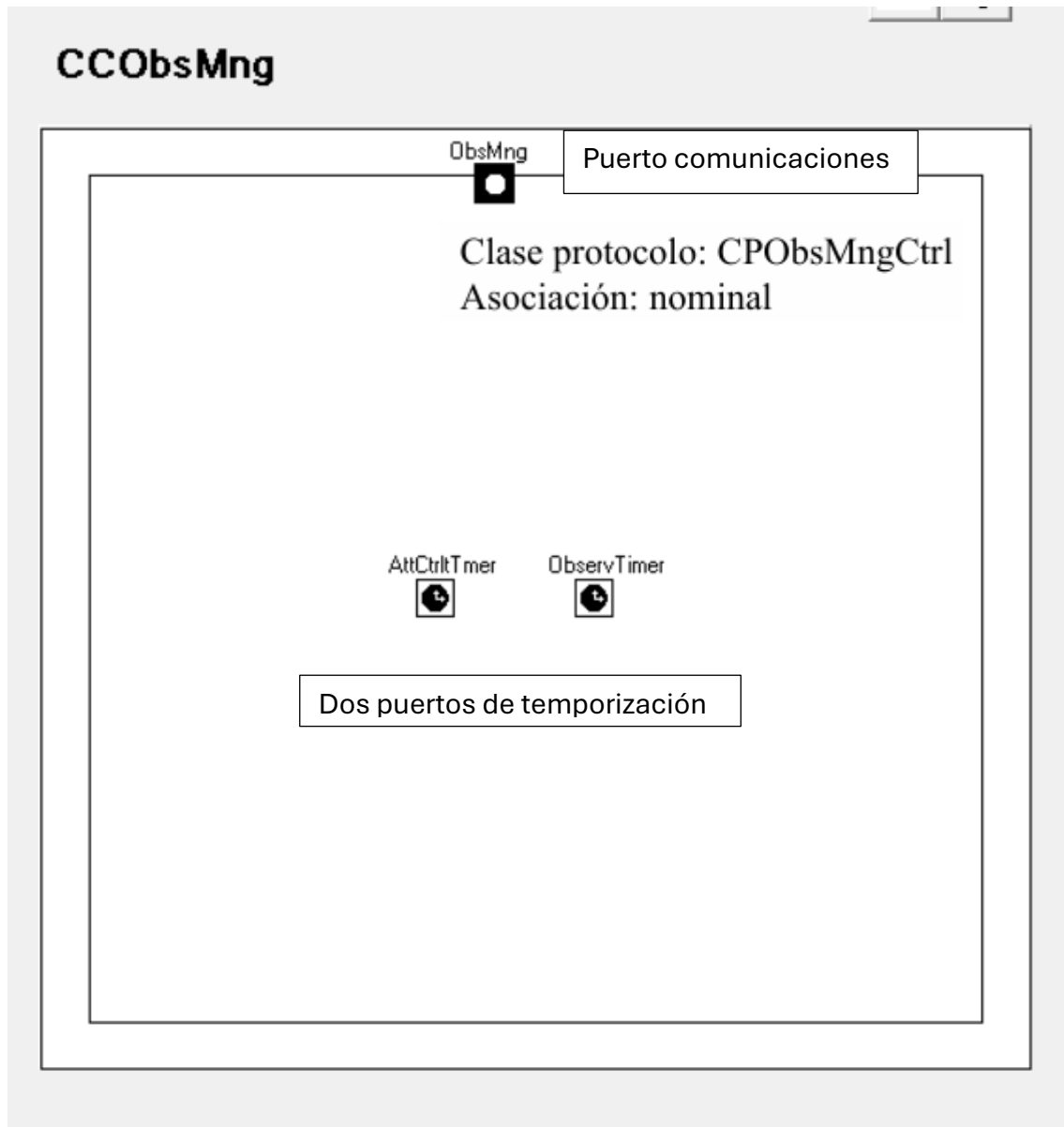


El nuevo componente creado está presente en 6 escenarios distintos.

ENTREGABLE 2: DEFINICIÓN DE LA CLASE PROTOCOLO

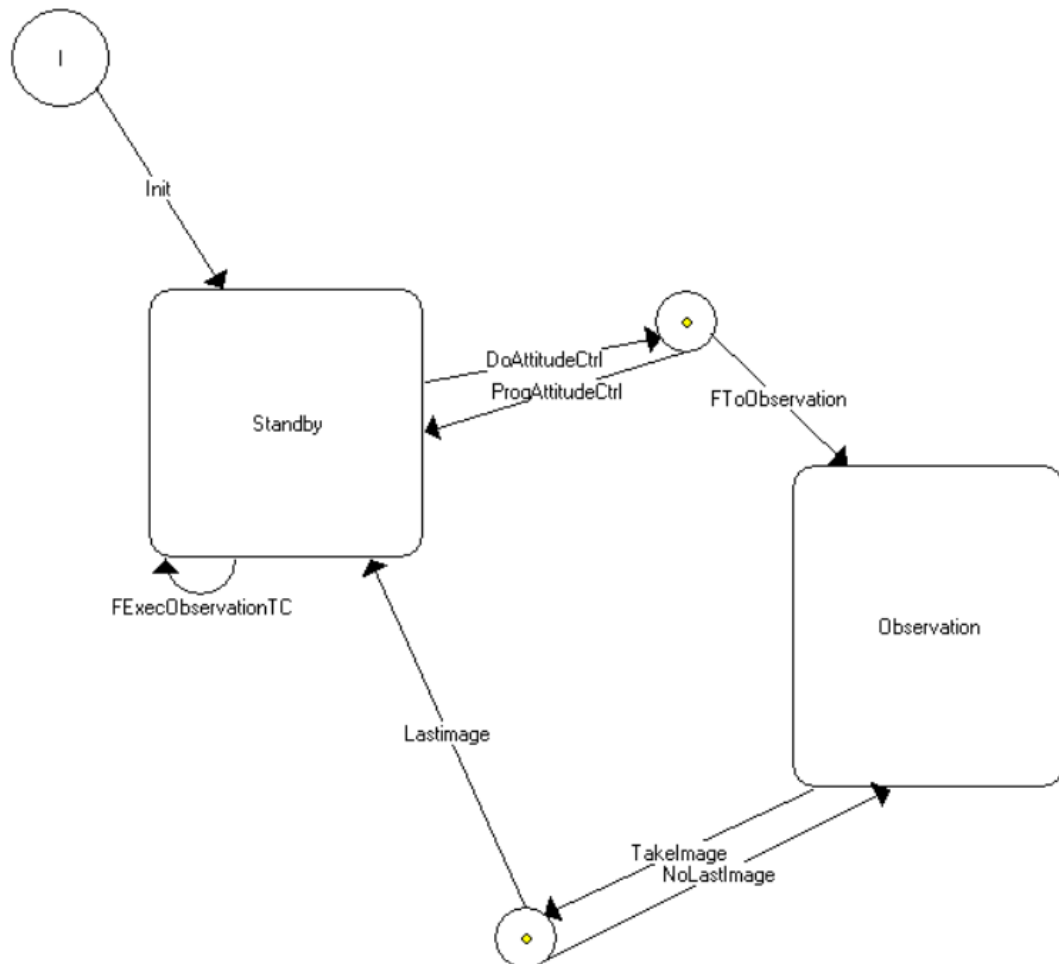


ENTREGABLE 3: DEFINICIÓN DE PUERTOS Y TIPOS



Se definen los 3 puertos pedidos, dos con requisitos de temporización y un puerto de comunicaciones, en este caso nominal, como vemos en la figura.

ENTREGABLE 4: MÁQUINA DE ESTADOS, TRIGGERS Y CÓDIGOS



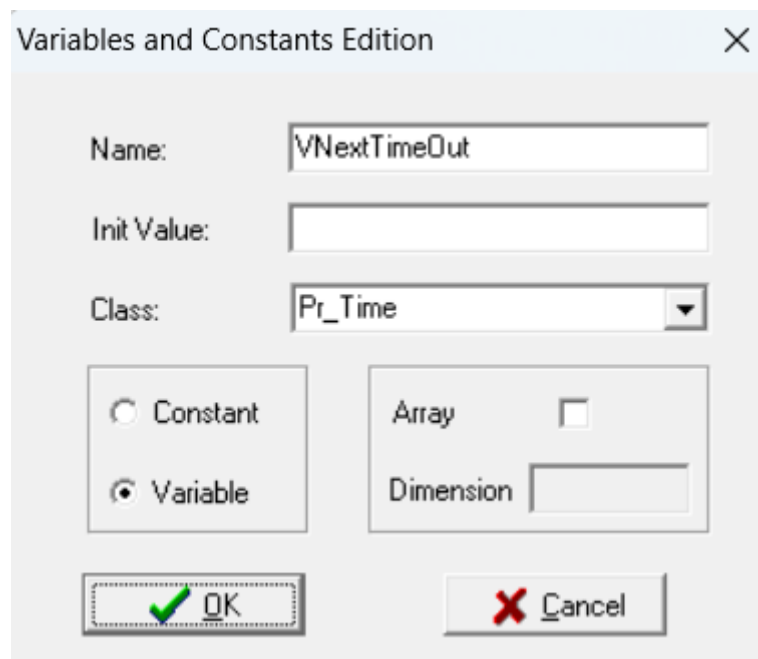
En esta primera imagen vemos la máquina de estados, a continuación, vamos a ir desglosando cada transición mediante capturas de pantalla del propio edroom y cuadros de textos con las condiciones de disparo.

TRANSICIÓN FINIT: Todas las funciones comienzan, por void (nombre de la transición) {código}. Para abreviar y por facilidad solo haré captura del código de dentro de la propia función.

```
Pr_Time time;  
  
//Timing Service useful methods  
  
time.GetTime(); // Get current monotonic time  
time +=Pr_Time(0,100000);  
  
AttCtrlTimer.InformAt ( time );|
```

Inform At: Finit()
P: EDROOMsl
S: EDROOMStart
G:true

Esta función, como podemos intuir por el código, lleva una variable asociada.



The image shows a dialog box titled "Variables and Constants Edition" with a close button (X) in the top right corner. The dialog contains the following fields and controls:

- Name:** A text box containing "VNextTimeOut".
- Init Value:** An empty text box.
- Class:** A dropdown menu showing "Pr_Time".
- Constant/Variable:** Two radio buttons. "Constant" is unselected, and "Variable" is selected.
- Array:** A checkbox that is unselected.
- Dimension:** An empty text box.
- Buttons:** At the bottom, there are two buttons: "OK" (with a green checkmark icon) and "Cancel" (with a red X icon).

AUTOTRANSICIÓN EXECOBSERVATIONTC

Msg->Data Handler: FExecObsMng_TC()

P: ObsMngCtrl

S: SObsMng_TC

G:true

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

```
    // Data access  
    varSObsMng_TC.ExecTC();
```

TRANSICIÓN DOATTITUDECTRL

Action: FDoAttitudeCtrl()

P: AttCtrlTimer

S: EDROOMSignalTimeout

G:true

```
pus_servicel29_do_attitude_ctrl();
```

TRANSICIÓN PROGATTITUDECTRL: LI

Action: FProgAttitudeCtrl()

G:true

```
Pr_Time time;
```

```
//Timing Service useful methods
```

```
time.GetTime(); // Get current monotonic time  
time +=Pr_Time(0,100000);
```

```
AttCtrltTmer.InformAt ( time );|
```


TRANSICIÓN FTOOBSERVATION: Primera guarda creada con código

Action: FToObservation()
G:GReadyToObservation()

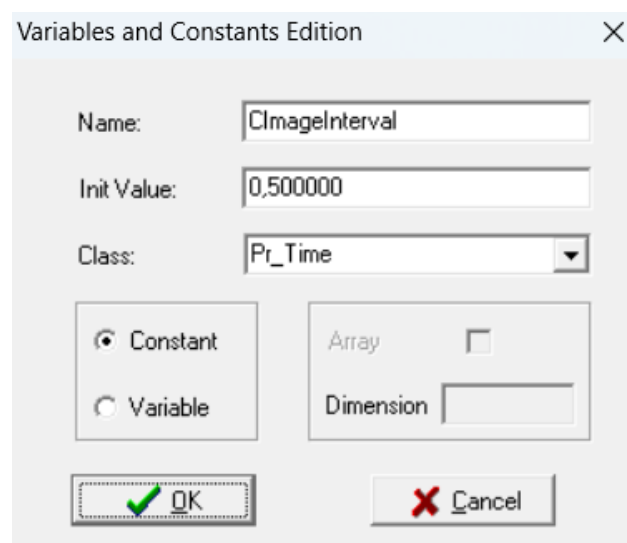
```
pus_servicel29_start_observation();  
  
return pus_servicel29_is_observation_ready();
```

COMPONENTE OBSERVATION: Función dentro del propio componente

Entry Inform In: FProgTakeImage()
P: ObservTimer
S: EDROOMSignalTimeout

```
Pr_Time interval;  
  
//Timing Service useful methods  
  
interval = CImageInterval; // interval of X sec + Y microsec  
  
//timers absolutos infor at e inform in relativos  
ObserverTimer.InformIn ( interval );|
```

Además, tiene una constante asociada:



The image shows a 'Variables and Constants Edition' dialog box. It has a title bar with a close button (X). The dialog contains the following fields and controls:

- Name:** A text box containing 'CImageInterval'.
- Init Value:** A text box containing '0,500000'.
- Class:** A dropdown menu showing 'Pr_Time'.
- Constant/Variable:** Two radio buttons. 'Constant' is selected (indicated by a filled circle), and 'Variable' is unselected (indicated by an empty circle).
- Array:** A checkbox that is unselected.
- Dimension:** A text box, currently empty.
- Buttons:** At the bottom, there are two buttons: 'OK' (with a green checkmark icon) and 'Cancel' (with a red X icon).

TRANSICIÓN TAKEIMAGE

Action: FTakeImage()

P: ObservTimer

S: EDROOMSignalTimeout

G:true

```
pus_servicel29_take_image();
```

TRANSICIÓN NOLASTIMAGE

G:true

TRANSICIÓN LASTIMAGE: Esta es una acción compuesta y además con la guarda que también hemos tenido que crear.

```
FEndObservation();
```

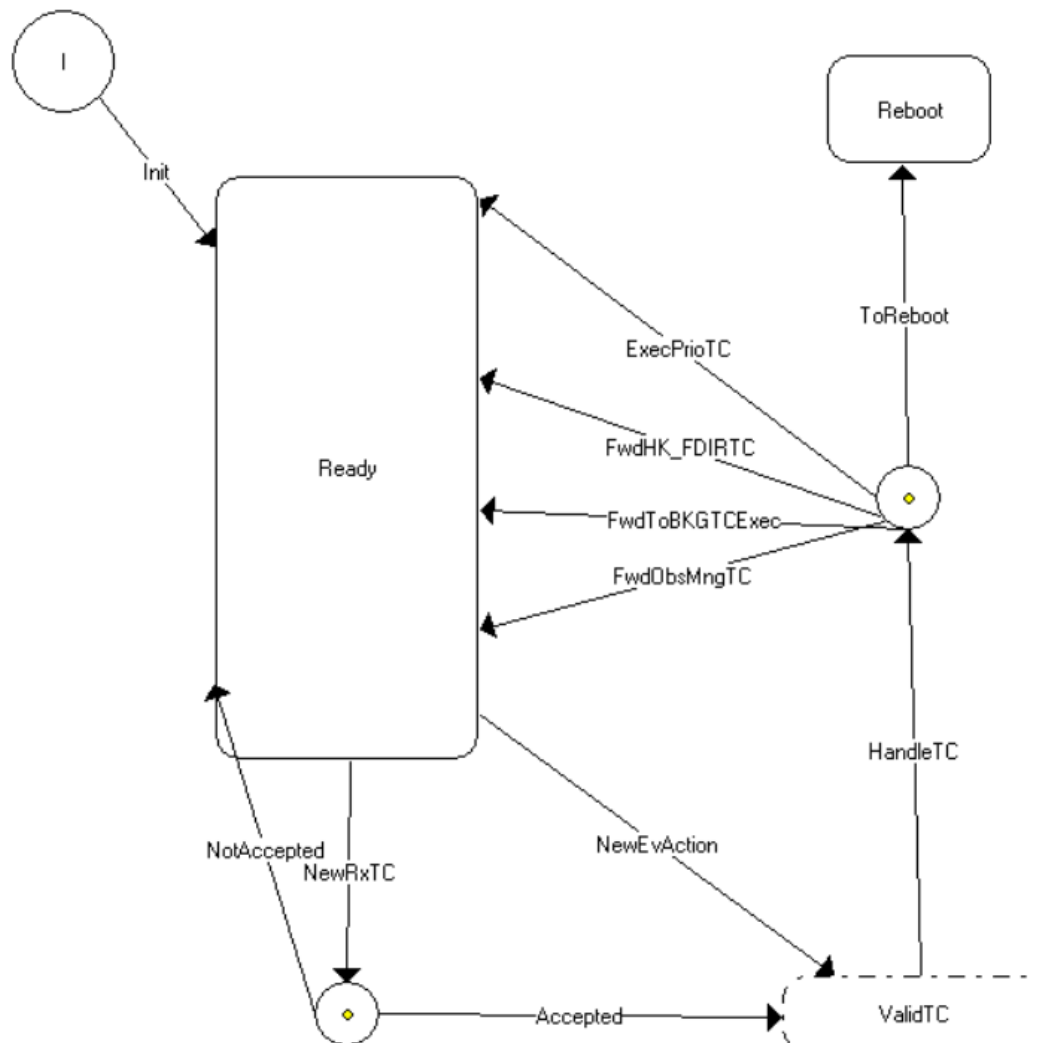
```
FProgAttitudeCtrl();
```

```
VNextTimeOut.GetTime();  
pus_servicel29_end_observation();
```

```
return pus_servicel29_is_last_image();
```

No adjunto el código de la segunda función de la composite, porque ya lo hemos indicado más arriba.

Ahora había que añadir otra transición más fuera de esta máquina de estados, se mostrará ahora (la de forward, además es de tipo send).



TRANSICIÓN FWDOBSMNGTC

EDROOM Service	
send	▼
Port	ObsMng ▼
Signal	SObsMng_TC ▼
Data Class	CDTCHandler ▼

Send: FFwdObsMngTC()
P: ObsMngCtrl
S: SObsMng_TC
Data Class: CDTCHandler
G: GFwdToObsMng()

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

```
        // Complete Data
```

```
        *pSObsMng_TC_Data=VCurrentTC;
```

CÓDIGO DE LA GUARDA

```
        return VTCExecCtrl.IsObsMngTC();
```

```
    ObsMng.send(SObsMng_TC,pSObsMng_TC_Data,&EDROOMPoolCDTCHandler);  
}
```