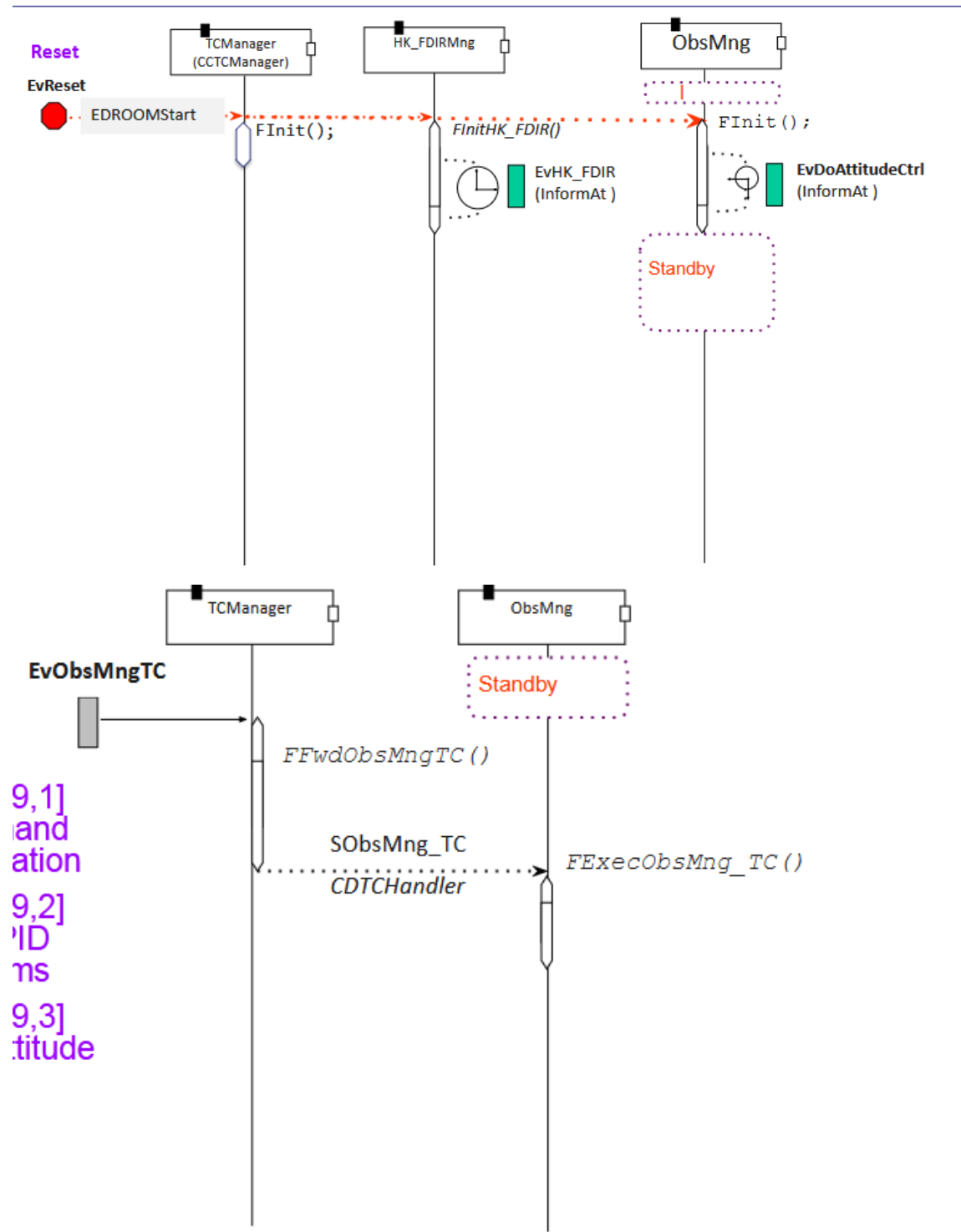


Entregable 1: Escenarios en los que está involucrado el nuevo componente ObsMng que controla la observación.



Entregable 2: Definición de la clase Procolo a añadir al Modelo EDROOM, aportando toda la información de cada mensaje.

Protocol Class Edition

Name:

Design | Analysis

Input Messages :

Output Messages:

Protocol Brief

Message Edition Box

Signal Name:

Data Class :

☐ Synchronous Invoke ☐ Asynchronous

☐ Synchronous Reply To ---->

Edit Message

New Input Message New Output Message

Delete Message

Modify Cancel

Message Edition Box

Signal Name:

Data Class :

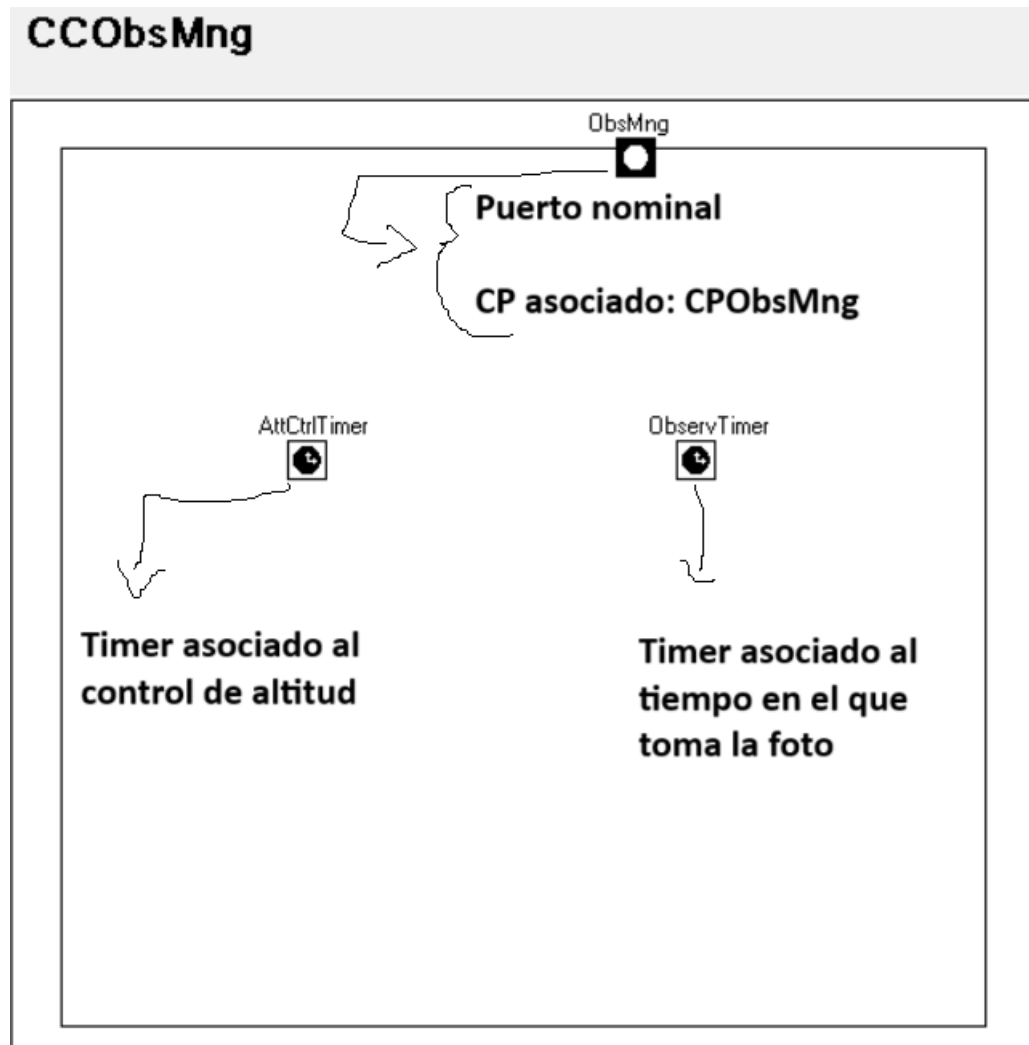
☐ Synchronous Invoke ☒ Asynchronous

☐ Synchronous Reply To ---->

Edit Message

Esto está definido desde la perspectiva de ObsMng. Para el TcMng sería el mensaje de salida.

Entregable 3: Diseño de la interfaz de la clase componente CCObsMng empleando la misma notación gráfica que se ha proporcionado durante las prácticas



Entregable 4: Diseño del comportamiento de la clase componente CCObsMng empleando la misma notación gráfica que se ha proporcionado durante las prácticas



## FDoAttitudeCtrl()

Function Edition

Declaration: FDoAttitudeCtrl()

Brief

Standard Library Includes

```
pus_servicel29_do_attitude_ctrl();
```

## FEndObservation()

Function Edition

Declaration: FEndObservation()

Brief

Standard Library Includes

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

## FExecObsMng\_TC()

Function Edition

Declaration: FExecObsMng\_TC()

Brief

Standard Library Includes

```
{  
    CDTCHandler & varSObsMng_TC = *(CDTCHandler *)Msg->data;  
  
    // Data access  
    varSObsMng_TC.ExecTC();  
  
}
```

**EDROOM Service**  
Msg->data  
Port  
ObsMng  
Signal  
SObsMng\_TC  
Data Class  
CDTCHandler  
Service Request

## FInit()

Function Edition

Declaration: FInit()

Brief

Standard Library Includes

```
{  
    Pr_Time time;  
  
    //Timing Service useful methods  
  
    time.GetTime(); // Get current monotonic time  
    time+=Pr_Time(0,100000); // Add X sec + Y microsec  
  
    AttCtrlTimer.InformAt( time );  
}
```

**EDROOM Service**  
InformAt  
Port  
AttCtrlTimer  
Signal  
EDROOMSignalTimeout  
Data Class  
CDTCHandler  
Service Request

## FProgAttitudeCtrl()

Function Edition

Declaration:

Brief

Standard Library Includes

```
{
    Pr_Time time;

    //Timing Service useful methods

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

    AttCtrlTimer.InformAt( time );
}
```

**EDROOM Service**

InformAt

Port

Signal

Data Class

## FProgTakeImage()

Function Edition

Declaration:

Brief

Standard Library Includes

```
{
    Pr_Time interval;

    interval=CImageInterval;

    ObservTimer.InformIn( interval );
}
```

**EDROOM Service**

InformIn

Port

Signal

Data Class

# FTakeImage()

Function Edition

Declaration:FTakeImage()

Brief

Standard Library Includes

pus\_servicel29\_take\_image();

# FToObservation()

Function Edition

Declaration:FToObservation()

Brief

Standard Library Includes

pus\_servicel29\_start\_observation();



## GLastImage()

Function Edition

Declaration:

Brief

Standard Library Includes

```
return pus_servicel29_is_last_image();
```

## GReadyToObservation()

Function Edition

Declaration:

Brief

Standard Library Includes

```
return pus_servicel29_is_observation_ready();
```

## DECLARACIÓN DE VARIABLES Y CONSTANTES:

### CImageInterval

Variables and Constants Edition ✕

Name:

Init Value:

Class:

☒ Constant ☐ Variable

Array ☐

Dimension

☒ OK ☒ Cancel

### VNextTimeOut

Variables and Constants Edition ✕

Name:

Init Value:

Class:

☐ Constant ☒ Variable

Array ☐

Dimension

☒ OK ☒ Cancel