

1 DISEÑO DEL SISTEMA EN TIEMPO REAL CON HRT-HOOD.

Vamos a utilizar la metodología HRT-HOOD para construir el sistema en tiempo real que se encarga del correcto funcionamiento de la maqueta de tren.

Este diseño se obtiene de desarrollar iterativa y concurrentemente tanto la arquitectura lógica como la arquitectura física. El diseño de la arquitectura lógica está destinado fundamentalmente a la satisfacción de los requisitos funcionales. En el diseño de la arquitectura física se relaciona la arquitectura lógica con los recursos de ejecución, se asignan atributos temporales a los objetos y se asegura el cumplimiento de los requisitos no funcionales (requisitos de tiempos de respuesta, seguridad, fiabilidad).

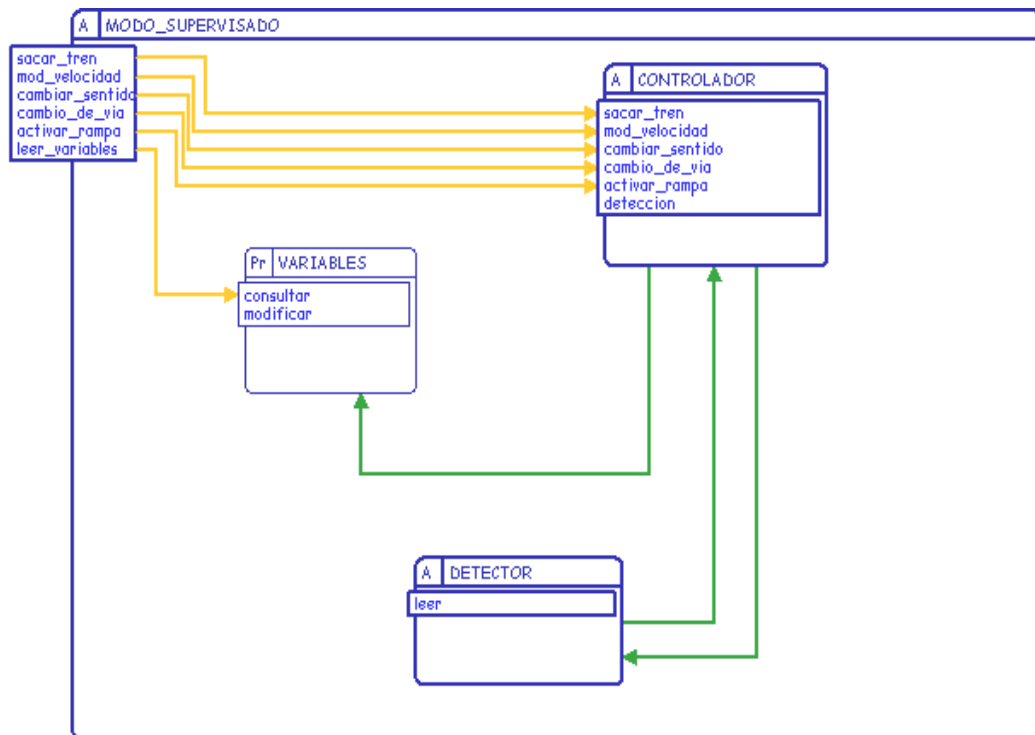
En este proyecto tiene mucha mayor importancia la arquitectura lógica que la arquitectura física. Ya que el sistema de control sólo tiene un procesador, el sistema no está distribuido y no debe haber problema con los plazos de respuesta es lenta comparándola con la capacidad de procesamiento de los computadores. Por esta razón no se ha realizado un diseño de la arquitectura física.

El funcionamiento del sistema, explicado muy por encima, consiste en detectar cíclicamente la posición de los trenes y en función de su situación el sistema de control decidirá si debe actuar o no. De forma parecida, cuando el controlador reciba una petición deberá decidir si puede ser atendida, o no.

Las siguientes figuras mostrarán los esquemas del diseño HRT_HOOD del sistema de control, que fue denominado MODO_SUPERVISADO.

El sistema que realizará el control de la maqueta y se encargará de la supervisión, permitirá realizar las siguientes operaciones:

- Sacar un tren.
- Modificar la velocidad de los trenes.
- Cambiar el sentido de una vía.
- Modificar un cambio de vía.
- Activar las rampas de desacople.
- Leer las variables del sistema.



Para realizar el control de la maqueta disponemos de dos objetos activos, DETECTOR y CONTROLADOR, y un objeto protegido VARIABLES.

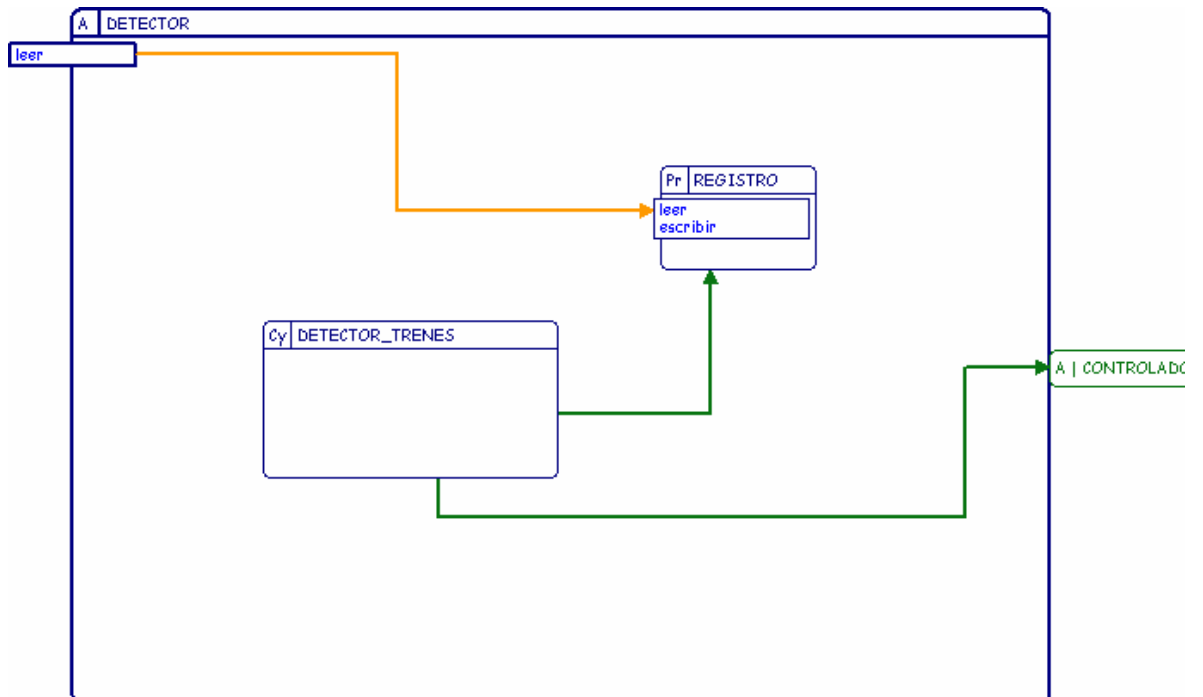
DETECTOR: Localiza la posición de los trenes en la maqueta.

CONTROLADOR: Realiza las operaciones de control sobre los trenes y diversos componentes de la maqueta.

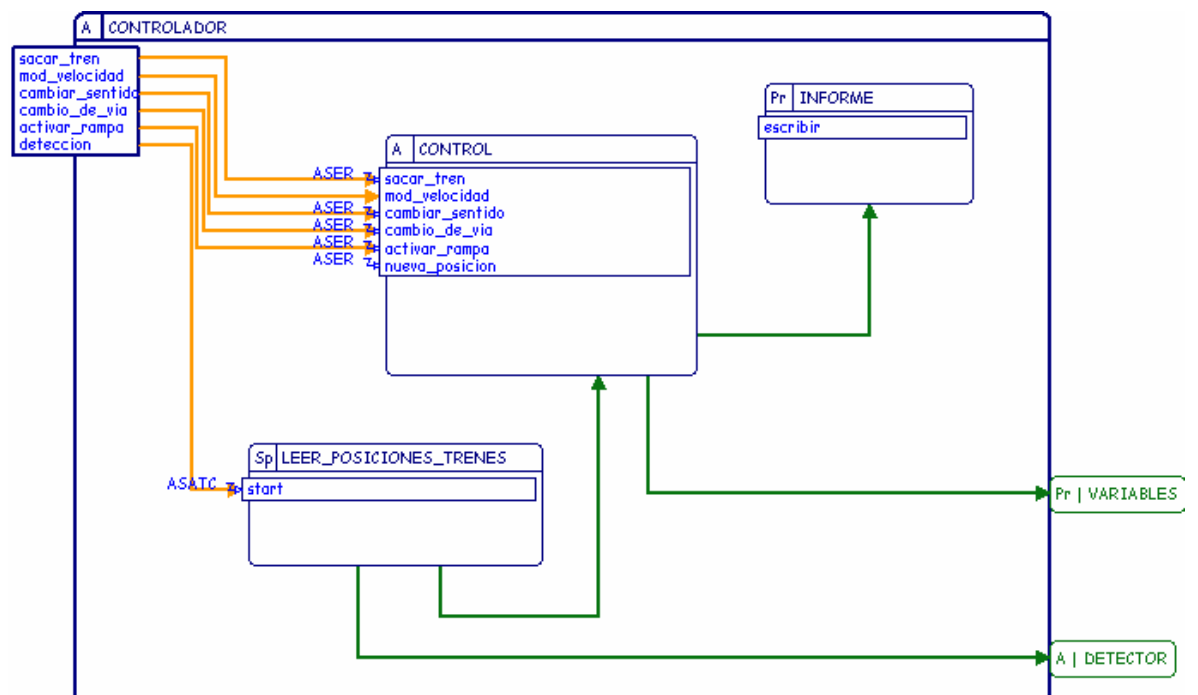
En el objeto protegido VARIABLES se almacena el estado de las variables del sistema, estado de los cambios de vía, sentido, velocidad y posición de los trenes, rampas de desenganche y semáforos. El estado de las variables puede ser consultado por el operador del sistema. El objeto CONTROLADOR es el encargado de modificar VARIABLES.

DETECTOR está formado por un objeto protegido REGISTRO y por el objeto cíclico DETECTOR_TRENES que localiza la posición de los trenes, y almacena los resultados en REGISTRO.

Cada vez que se haga una escritura en REGISTRO se advierte a CONTROLADOR, para que consulte las nuevas posiciones almacenadas en REGISTRO.



CONTROLADOR está formado por el objeto activo CONTROL, el objeto protegido INFORME y por el objeto esporádico LEER_POSICIONES_TRENES.



Cuando en DETECTOR se realiza una escritura en REGISTRO se activa el objeto esporádico LEER_POSICIONES_TRENES, que comprobará si los trenes han cambiado de posición, o si los valores leídos son los mismos que en mediciones anteriores.

Si las posiciones no son nuevas no hace nada, pero si algún tren ha cambiado de posición se avisa al objeto activo CONTROL y se le pasan las nuevas posiciones de los trenes.

El objeto CONTROL se encargará de estudiar las peticiones para sacar un tren, modificar los cambios de vía, cambiar el sentido de los trenes, activar las rampas de desacople de vagones, o modificar las velocidades de los trenes, siendo el encargado de admitir o rechazar las peticiones, en función del estado de la maqueta.

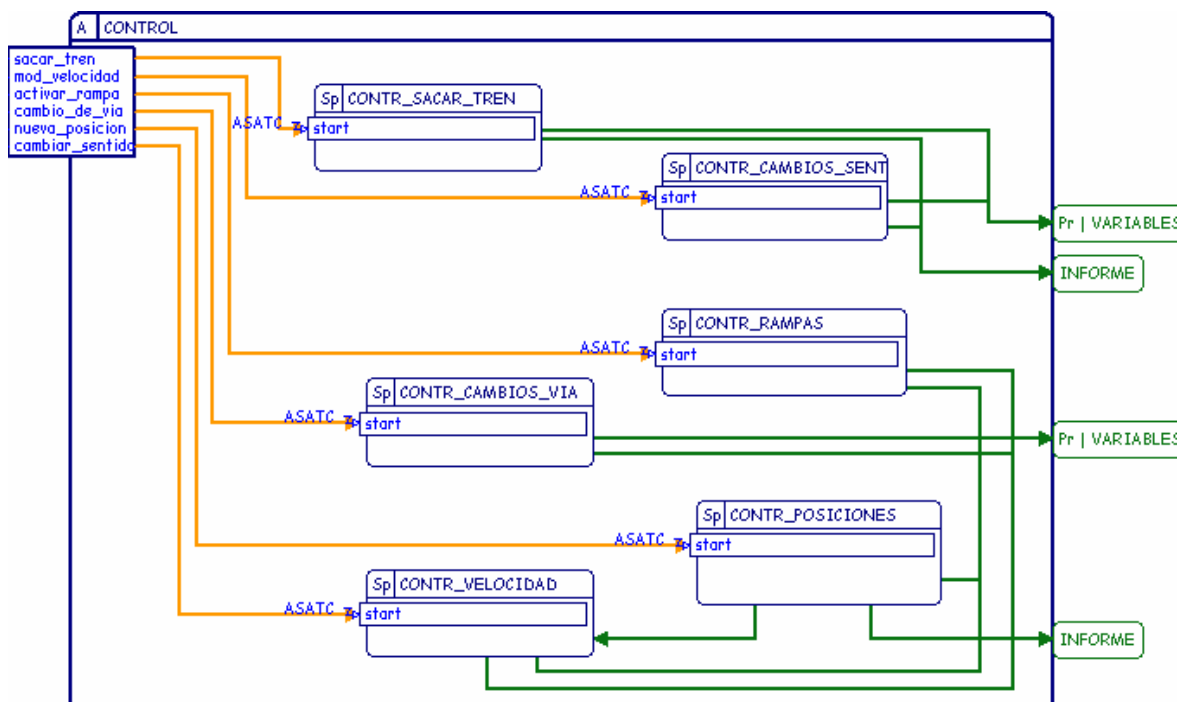
El objeto CONTROL utilizará el objeto protegido INFORME para documentar todos los eventos sucedidos en la maqueta y las acciones llevadas a cabo sobre los accionadores. Cada vez que se escriba algo en INFORME anotará también la hora.

Cada vez que se añada algo a INFORME se visualizará por pantalla y además podrá ser consultado posteriormente por el operador del sistema.

El objeto CONTROL está formado por varios objetos esporádicos que se encargan de controlar que se dan las condiciones necesarias para poder sacar un tren, cambiar el sentido de una vía, activar las rampas de desacople, modificar los cambios de vía, cambiar la velocidad de las vías, y controlar las medidas a tomar en función de las posiciones de los trenes.

Cada objeto esporádico al realizar sus operaciones accederá al objeto protegido VARIABLES. Si se dan las condiciones necesarias para satisfacer la petición recibida, se realizarán las modificaciones pertinentes y se escribirá en el objeto protegido INFORME lo referente a la petición recibida (si ha sido atendida, se anotará la hora y los cambios hechos en VARIABLES), si una petición no es atendida se anota en INFORME que la petición no ha sido atendida.

El objeto CONTR_ POSICIONES utiliza a CONTR_VELOCIDAD para realizar su misión, ya que necesita poder actuar sobre la velocidad de los trenes.



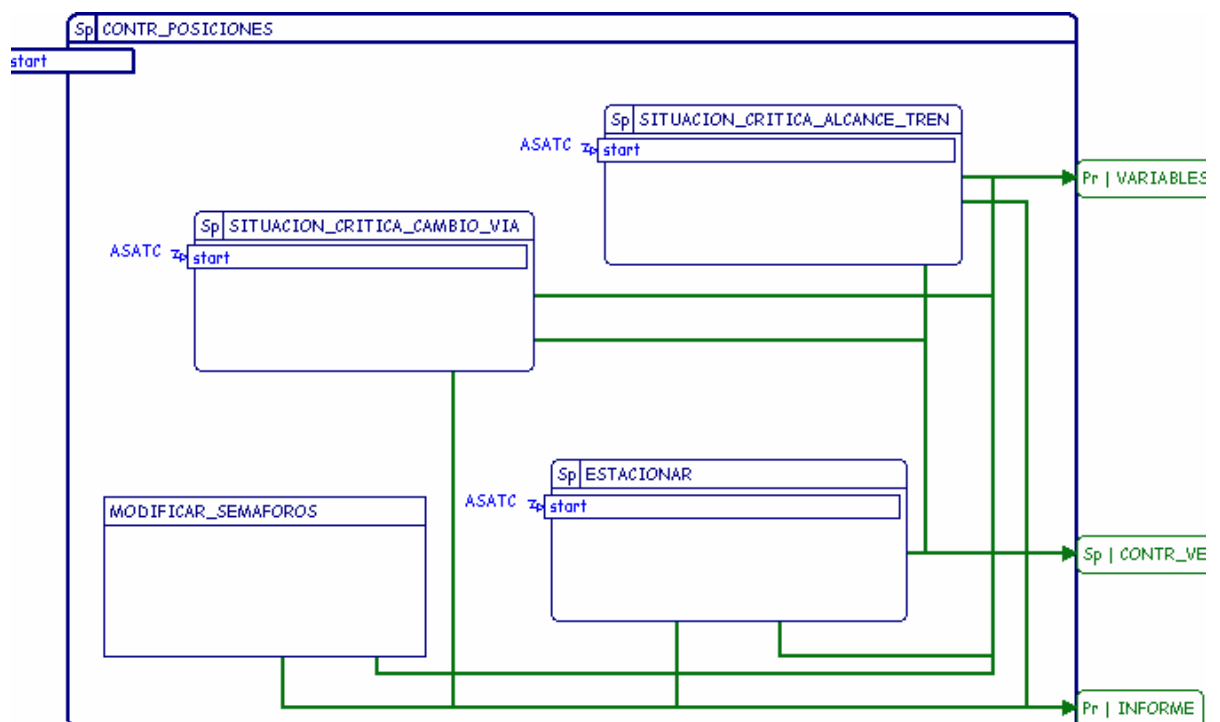
De todos los objetos esporádicos que constituyen el objeto control el más importante es CONTR_POSICIONES. Se encargará de detectar las situaciones que pueden hacer colisionar

los trenes, señalar la presencia de algún tren en la zona de intersección con la carretera, y detectar la entrada de un tren en terminales. Se descompone en tres objetos esporádicos, que se encargan de tratar las situaciones antes detectadas, y un objeto pasivo para modificar los semáforos.

El objeto esporádico `SITUACION_CRITICA_CAMBIO_VIA` se activará cuando exista riesgo de colisión entre los trenes, y realiza las acciones necesarias para superar la situación crítica, como hemos dicho anteriormente tendrá que modificar la velocidad de los trenes. El tratamiento consiste en detener el tren que cambia de vía para dejar que pase el otro, y una vez que haya pasado se vuelve a alimentar al tren para que cambie de vía. Para eso necesita al objeto `CONTR_VELOCIDAD`.

El objeto esporádico `SITUACION_CRITICA_ALCANCE_TREN` se activará cuando un tren esté alcanzando al otro, y se encargará de que los trenes se distancien, para lo cual detendrá al tren perseguidor, actuando sobre las salidas del autómata, inhibiendo la alimentación del tren perseguidor, para que el otro coja ventaja.

El otro objeto esporádico, `ESTACIONAR`, realizará las acciones necesarias durante las maniobras de un tren al entrar en terminales.



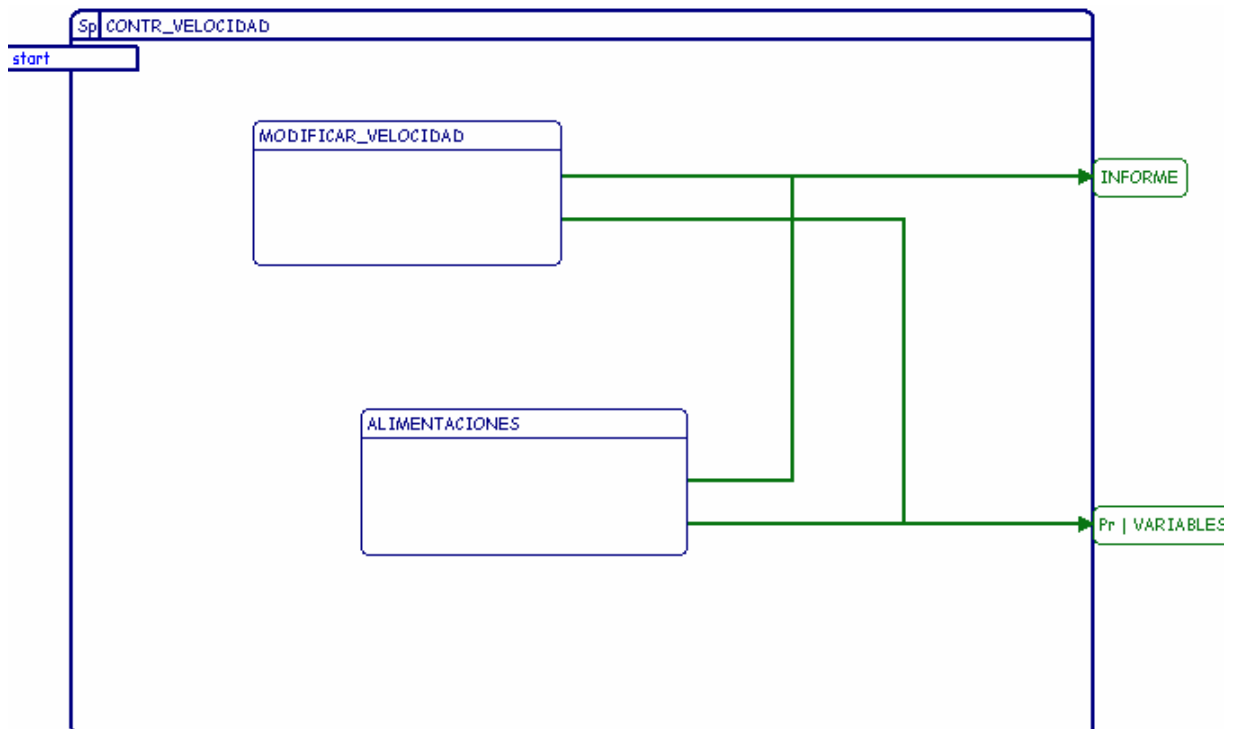
El objeto pasivo `MODIFICAR_SEMAFOROS` será utilizado para actualizar los semáforos.

`CONTR_VELOCIDAD` se encarga de comprobar si se dan las condiciones necesarias para permitir modificar la velocidad de los trenes. Para cambiar la velocidad de los trenes llama a los objetos pasivos `MODIFICAR_VELOCIDAD` y `ALIMENTACIONES`.

Las peticiones para modificar la velocidad deben indicar sobre qué vía se quieren efectuar los cambios y el nuevo valor de su velocidad.

Ante una petición CONTR_VELOCIDAD estudiará si se puede permitir el cambio de velocidad. Si no se modifica la velocidad se anotará en INFORME que la petición fue rechazada. Si se atiende la petición de modificar la velocidad del tren se usará el objeto pasivo MODIFICAR_VELOCIDAD, se guardarán los cambios en VARIABLES, y se anotará en INFORME el cambio de velocidad y la hora.

El objeto MODIFICAR_VELOCIDAD es el que hace que cambie la velocidad del tren, mientras que ALIMENTACIONES se encarga de que la alimentación llegue a los diversos tramos de vías, o bien sean inhibidos.

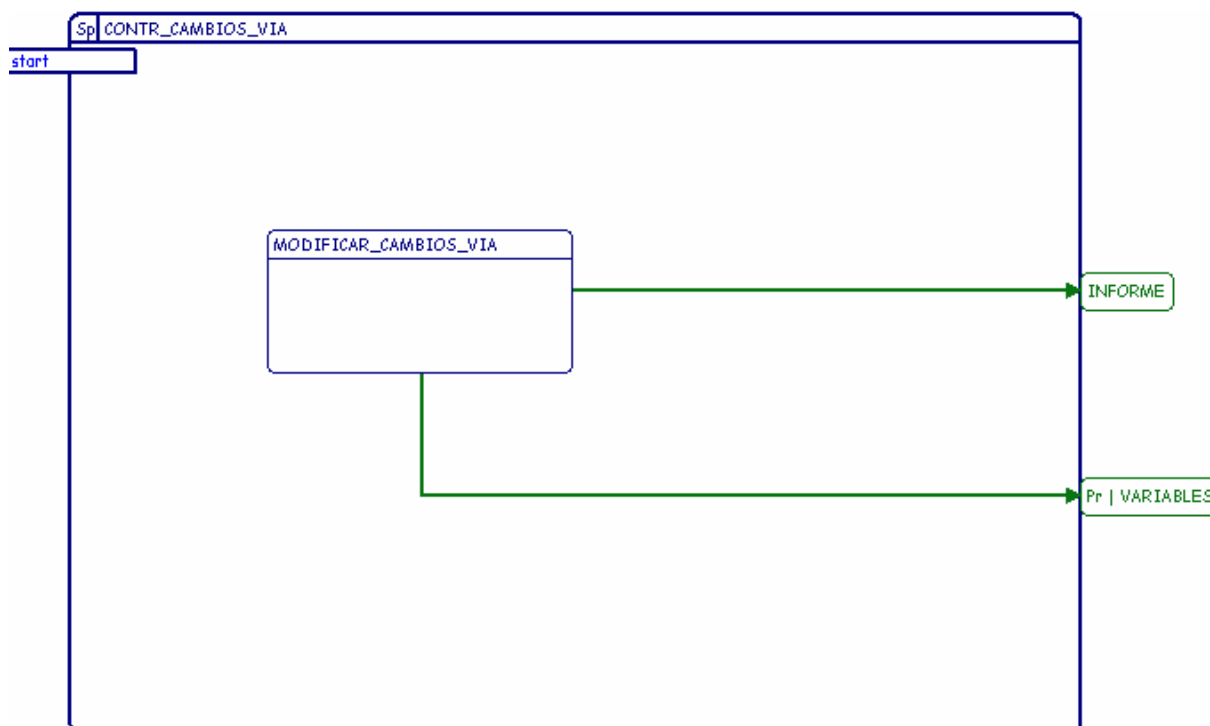


CONTR_CAMBIOS_VÍA se encarga de comprobar si se dan las condiciones necesarias para permitir modificar un cambio de vía. Para modificar los cambios de vía se llama al objeto pasivo MODIFICAR_CAMBIOS_VIA.

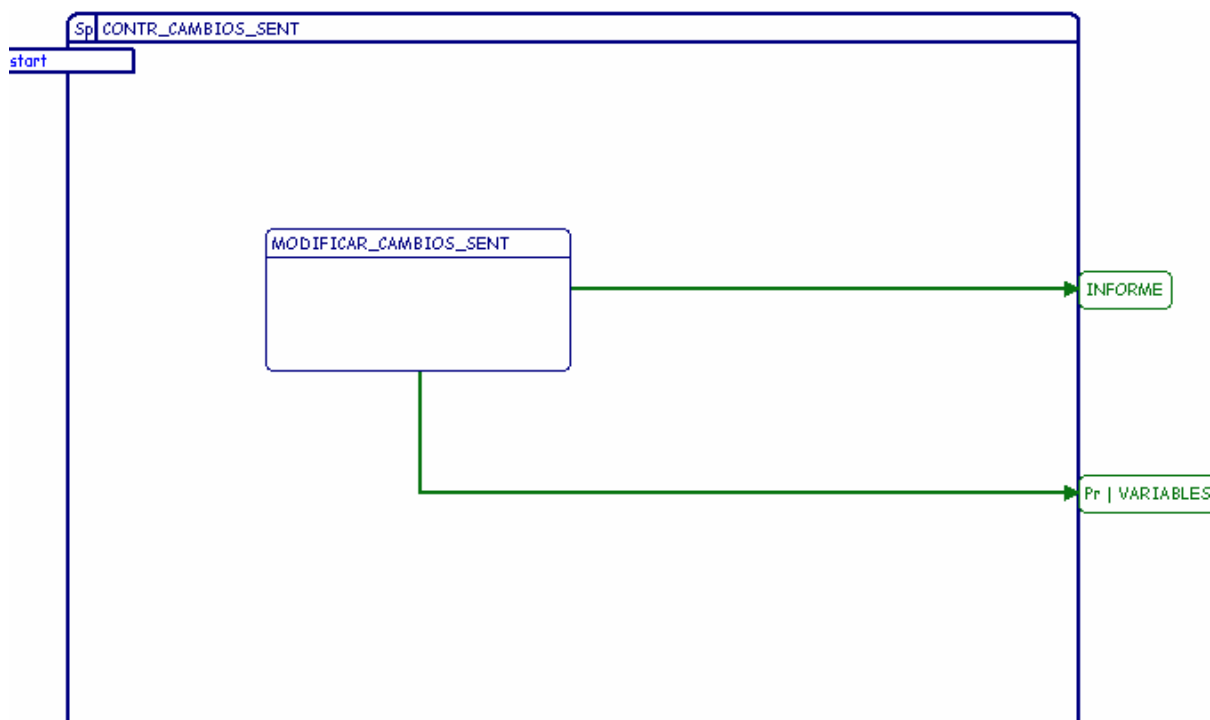
Las peticiones para modificar un cambio deben indicar qué cambio se quiere modificar y en qué posición se quiere colocar.

El objeto MODIFICAR_CAMBIOS_VIA es el encargado de efectuar las modificaciones sobre los cambios de vía.

Ante la petición de modificar un cambio de vía CONTR_CAMBIOS_VIA, estudiará si se dan las condiciones para llevarlo a cabo. Si se produce el cambio de vía usará MODIFICAR_CAMBIOS_VIA para efectuar el cambio, guardará la nueva posición en VARIABLES, y en INFORME se anotarán los cambios. Si no se produce se anotará en INFORME que la petición fue rechazada.



CONTR_CAMBIOS_SENT se encarga de comprobar si se dan las condiciones necesarias para permitir cambiar el sentido de una vía. Para modificar el sentido de la vía se llama al objeto pasivo MODIFICAR_CAMBIOS_SENT.



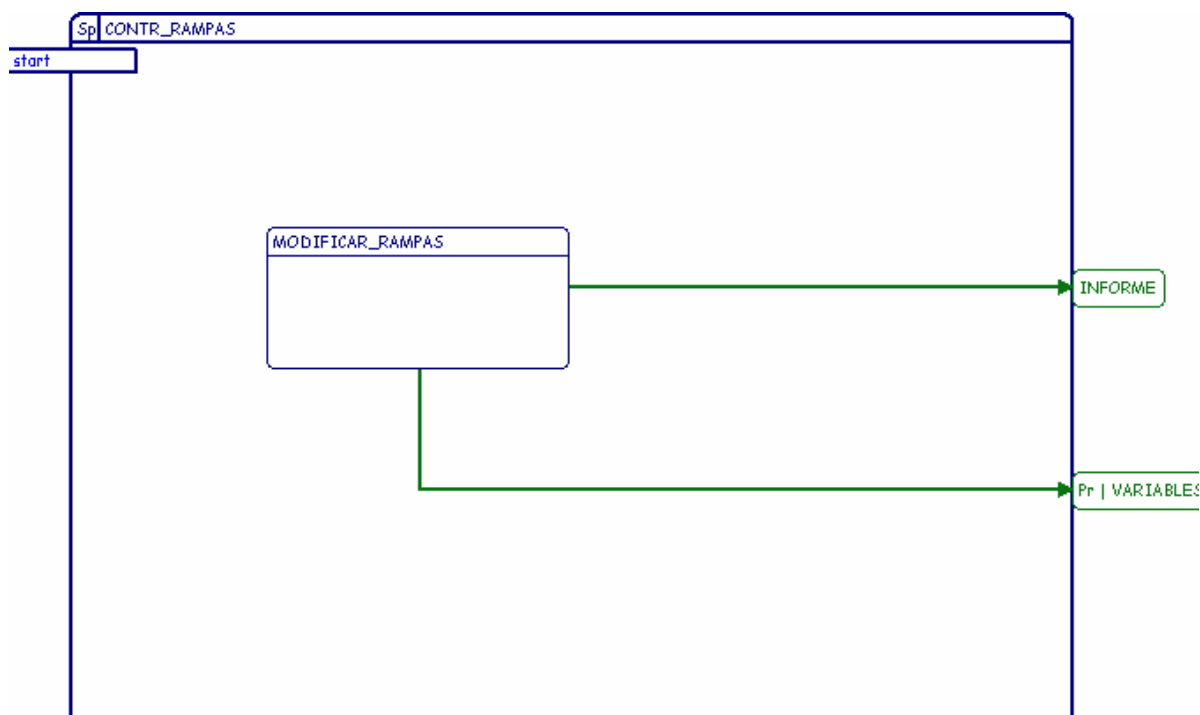
Las peticiones deben indicar a qué vía se quiere cambiar el sentido.

El objeto MODIFICAR_CAMBIOS_SENT es el encargado de modificar el sentido.

Ante la petición de cambiar el sentido de una vía CONTR_CAMBIOS_SENT estudiará si se dan las condiciones necesarias para permitirlo. Si se produce el cambio de sentido

llamará a MODIFICAR_CAMBIOS_SENT para efectuar el cambio, guardará el nuevo valor en VARIABLES, y en INFORME se anotarán los cambios. Si no se produce se anotará en INFORME que la petición fue rechazada.

CONTR_RAMPAS se encarga de comprobar si se dan las condiciones necesarias para permitir activar una rampa. Para activar las rampas se llama al objeto pasivo MODIFICAR_RAMPAS.



Ante la petición de activar una rampa CONTR_RAMPAS estudiará si se dan las condiciones para activarla. Si no se atiende la petición se anotará en INFORME que fue rechazada. Si se atiende la petición se anotará en INFORME que la petición para activar la rampa ha sido atendida y la hora en que se produjo, se guardará el nuevo valor en VARIABLES. Las peticiones deben indicar qué rampa se quiere activar.