Universitat Autònoma de Barcelona

escola d'enginyeria

4583-3: VEHÍCULOS-ROBOT DE GUIADO AUTOMÁTICO PARA ALMACENES Y PLANTAS DE MANUFACTURACIÓN

Memoria del Proyecto Final de Carrera
De Ingeniería en Informática
Realizado por
José Miguel Moreno Villafranca
y dirigido por
Lluís Ribas-Xirgo
Bellaterra, 7 de Junio de 2012

Universitat Autònoma de Barcelona

Escola Tècnica Superior d'Enginyeria

El sotasignat, Lluís Ribas-Xirgo

Professor de l'Escola Tècnica Superior d'Enginyeria de la UAB,

CERTIFICA:

Que el treball a què correspon aquesta memòria ha estat realitzat sota la seva direcció per en José Miguel Moreno Villafranca

I per tal que consti firma la present.

Signat: .....

Bellaterra, 14 de Juny de 2012

AGRADECIMIENTOS

Quiero agradecer a todas las personas que hayan colaborado con su tiempo y conocimiento a elaborar este proyecto de final de carrera y que, gracias a ellos, ha llegado a buen fin.

Muchas gracias a Cristina Martín y a todos aquellos que han colaborado.

Siempre agradecido:

José Miguel Moreno

ÍNDICE

CAPÍTULO 1. Introducción

1.1 Contexto

En la industria y sobre todo en los puertos y almacenes, poco a poco, es más habitual encontrarse sistemas automatizados para el transporte interno de los materiales. Los materiales se transportan en vehículos sin conductor que se guían de forma automática sin requerir un operario, ya sea a bordo o controlándolo a distancia, para guiarlos. Estos vehículos se conocen como *vehículos de guiado automático* o más conocidos por sus siglas en inglés AGV provenientes de Automated Guided Vehicle.

El tema del proyecto es la gestión de tráfico de AGV. Estos vehículos sin conductor son capaces de seguir caminos, detectar obstáculos y más funciones a pequeña escala sin necesidad de una supervisión humana constante. Algunos tienen una capacidad de proceso superior y son capaces de calcular rutas, comunicarse con otros para establecer una preferencia en un cruce, pero no pueden tomar decisiones de una forma óptima o calcular una ruta teniendo en cuenta el estado global del tráfico. Además, un cambio en el área de trabajo implica tener que reprogramar todos los AGV. Con un sistema central que les de las órdenes, calcule las trayectorias, de las preferencias en los cruces y controle el tráfico de una forma centralizada permite reducir los costes de los AGV, que necesitaran menos capacidad de proceso, y se podrá gestionar el tráfico desde un punto de vista global y más óptimo.

Para el proyecto se simulará un caso sencillo de un laboratorio de análisis clínico en el que los AGV transportan muestras de sangre hasta los diferentes analizadores encontrándose en el camino con cruces, bifurcaciones y actualizaciones de rutas. Las muestras de sangre han de ser cargadas en el AGV en un área destinada a ello, transportadas hasta las máquinas de análisis y seguir en la zona de análisis hasta que se confirme que el análisis ha sido satisfactorio y llevar las muestras para ser retiradas del AGV dejándolo listo para un nuevo transporte o recargar las baterías. Esto es lo que actualmente realizan las cintas y con estas restricciones en un laboratorio real.

1.2 Motivación

Los sistemas de transporte basados en vehículos AGV son más flexibles y permiten una reestructuración del área de trabajo rápida y con un bajo coste en materiales. Por la parte del sistema central, permite la traslación de un entorno de trabajo a otro con facilidad, simplemente adaptando las órdenes a los nuevos AGV y especificándole al software el nuevo entorno donde van a trabajar los AGV. Hoy día, en muchos sitios donde se usan AGV, solo hay 1 o 2 vehículos de este tipo circulando por una determinada zona. La gestión del tráfico, por tanto, es sencilla o inexistente, en algunos casos los AGV tienen limitado los espacios por donde se pueden mover y tienen pocas áreas comunes de paso. Por tanto se puede gestionar de una forma rápida y sin requerir una gran capacidad de cómputo (por ejemplo: pasa el que viene por la derecha). Pero en entornos donde decenas o centenas de AGV en un espacio

reducido se cruzan, esta norma podría acabar con el colapso de las zonas sin preferencia. De ahí que sea importante utilizar un sistemas de gestión y simulación para permitir el crecimiento en el uso de este tipo de vehículos en la industria y garantizar a su vez con las simulaciones el correcto funcionamiento de estos.

1.3 Problemas

Los problemas que se plantean resolver en el proyecto tienen que ver con el control del tráfico de los AGV que circulan por la planta de análisis. En esta planta encontramos bifurcaciones, uniones, zonas de adelantamiento, atascos y actualizaciones de ruta de los AGV en marcha.

-En las bifurcaciones no encontramos problemas complicados, ya que los AGV simplemente eligen un camino u otro dependiendo de la ruta que tengan fijada.

-En las uniones sí que encontramos problemas debido a que, en caso de haber un vehículo en cada carril de la unión, hay que especificar cuál de los dos tiene preferencia de paso.

-En las zonas de adelantamiento hay que comunicarse con el sistema central para especificar si se debe o no tomar la ruta. Además están compuestas por una bifurcación y una unión, pero el problema es el mismo que los casos anteriores y no plantearía nuevos casos.

-Los atascos en nuestro caso son difíciles de resolver debido a que no tenemos diferentes rutas para llegar a un mismo punto y por tanto no se pueden evitar las zonas conflictivas. Si se puede por otra parte dar más preferencia a la zona atascada en los semáforos para intentar deshacer el colapso.

-Las actualizaciones de ruta plantean el mismo problema que las zonas de adelantamiento en cuanto a la comunicación con el sistema central, solo que esta vez la respuesta es una lista de destino/s nueva.

1.4 Objetivos

Los objetivos que se pretenden conseguir con este proyecto son:

1. Obtener un software de gestión de AGV capaz de controlar el tráfico y parcialmente los AGV en nuestro laboratorio de análisis.
2. Obtener una simulación para verificar la utilidad de los algoritmos en diferentes situaciones de densidad de tráfico y velocidad de trabajo.
3. Crear una simulación completa con los algoritmos seleccionados y generar órdenes aleatorias para ver la simulación global.
4. Tratar casos puntuales de batería baja, cambios de ruta, análisis de urgencia, etc.

1.5 Tareas

Para conseguir los objetivos del proyecto se necesita realizar las siguientes tareas:

-Plantear los posibles problemas que se encontraran los AGV durante su trabajo.

-Buscar algoritmos de gestión de tráfico y su implementación.

-Planificar y programar las clases, diagrama UML o máquinas de estado para la gestión de los AGV.

-Aprender a usar el entorno de simulación AnyLogic como programa base para la simulación.

-Programar los algoritmos mediante una programación en espiral generando un caso sencillo al principio y rehaciendo las clases para que cada vez puedan albergar casos más complicados hasta lograr resolver los problemas objetivos que plantea el entorno de trabajo.

-Simular problemas reduciendo el área de trabajo para centrarnos y facilitar los eventos problemáticos.

-Simulación global de toda la planta

-Realizar la memoria

-Crear videos con la simulación para la presentación

1.6 Planificación

Tarea	Horas	Total
Buscar algoritmos de gestión de tráfico requeridos para el caso de estudio	5 h	5 h
Plantear los problemas del tráfico y asignar algoritmos	15 h	20 h
Aprender AnyLogic	90 h	110 h
Programar clase a clase e integrar en el sistema global	130 h	240 h
-Introducir código	65 h	65 h
-Testear	15 h	80 h
-Realizar simulación	20 h	100 h
-Realizar ajustes	15 h	115 h
-Sincronizar con otras clases	15 h	130 h
Simulación global	10 h	250 h
Realizar ajustes de las clases	10 h	260 h
Crear videos	5 h	265 h
Memoria	40 h	305 h
Presentación	40 h	345 h
Reuniones	30 h	375 h
Libre para imprevistos	75 h	450 h

1.7 Análisis de viabilidad

El proyecto es viable en el aspecto técnico ya que únicamente es necesario un PC con determinado software. La fluidez en la ejecución no supone un inconveniente (exceptuando el tiempo que se pierde) debido a que al ser una simulación no necesita una respuesta en tiempo real (un segundo de simulación puede ser calculado en más tiempo, pero el resultado sería el mismo que si se realiza en la realidad). Esto permite el uso de prácticamente cualquier PC y por tanto elimina la necesidad de inversión en nuevo hardware. En el aspecto de recursos

humanos la viabilidad es aparentemente aceptable ya que una única persona respetando los tiempos es capaz de llevar a cabo el proyecto.

1.8 Estado del arte

Hoy día podemos encontrar varias empresas que comercian con software de gestión de tráfico de AGV, pero al contrario que ocurre con otro tipo de software como el Microsoft Office o el PhotoShop que se puede empezar a usar tras la instalación, este no es un producto terminado, sino que facilita la gestión y los cálculos de rutas así como la creación de una interface de usuario pero se debe personalizar a las necesidades de la empresa y los AGV con los que trabaja.

Entre los diferentes proveedores de software podemos encontrar los productos de gestión de AGV (o que en el paquete de programas se encuentre un gestor de AGV) E'tricc (Egemin Transport Intelligent Control Center), Transbotics Movement Optimizer (TMO), Gottwald Management and Navigation Systems o Autostore WMS.

Por lo general no se suele encontrar únicamente AGV en las empresas que deciden una automatización de sus labores de transporte y almacenaje. Esto se debe a que muchos AGV no controlan el tamaño de la carga que transportan ni que su posicionado sobre el sea correcto. Esto conlleva a que en zonas estrechas la carga pueda impactar con algo produciendo serios daños. Entre las posibles soluciones está que la carga la realice también una grúa automática con sensores para verificar el correcto posicionamiento de la carga sobre el AGV y así maximizar el correcto funcionamiento del sistema de transporte automatizado.

CAPÍTULO 2. Especificaciones

2.1 Laboratorio de análisis clínico

La simulación se basara en un escenario real pero no se llevara a cabo en el. Este escenario es un laboratorio de análisis clínico que actualmente funciona con un sistema de cintas de transporte. Para este caso en concreto, el sistema de cintas da muy buen resultado y no compensa la inversión para cambiar el sistema de cintas a un sistema de AGV, pero como se trata de demostrar la adaptabilidad de los AGV a los entornos de trabajo, el laboratorio nos puede servir perfectamente como ejemplo de un posible uso de los mismos y como posible alternativa en caso de querer construir un laboratorio nuevo.

Se tratara de sustituir las cintas por líneas y marcas en el suelo que seguirán los AGV desde el punto de carga de las muestras de sangre hasta los distintos puntos donde las máquinas de diagnóstico realizan el análisis. Tras realizar los análisis el AGV seguirá dando vueltas por el circuito y repitiendo las pruebas hasta que todas den la confirmación de haberse realizado con éxito. En ese momento el AGV retornará hasta el área de carga para descargar las muestras de sangre y recoger nuevas muestras para analizar.

2.2 Modelo de AGV

El AGV es un vehículo pequeño, en nuestro caso, que se guía por una línea en el suelo y marcas que indican, por ejemplo, tramos del camino con las que debe decidir si tomar una bifurcación o si debe consultar si el semáforo de la intersección lo tiene en verde para continuar. Dispone de un emisor-receptor de bluetooth para comunicarse con el ordenador central. Dado que el fin del proyecto es la simulación de varios AGV no nos centraremos en los sistemas para el seguimiento de líneas ni detección de obstáculos ya que son procesos transparentes al sistema de gestión y muchos programas de simulación ya lo realizan de forma automática.

Para ejecutar el programa del sistema central se usara un PC con sistema operativo Windows y un software de simulación (AnyLogic). La simulación de los AGV se realiza por software con programas específicos ya que no se dispone del número suficiente de AGV para una simulación real.

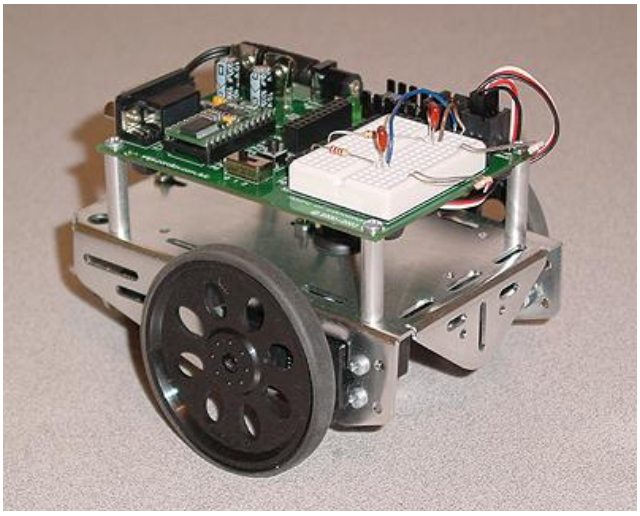


Imagen de un AGV. Modelo de Boe Bot

2.3 Área de trabajo

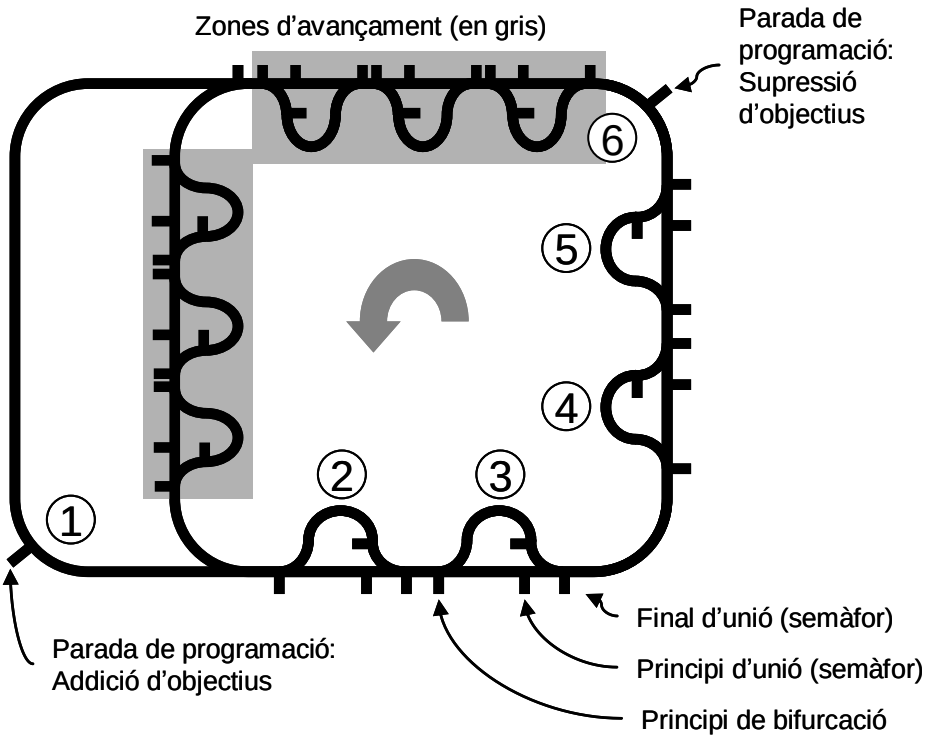


Figura 2.1

En este mapa (Figura 2.1) podemos ver los recorridos que podrán realizar los AGV. Los recorridos son de un único sentido y dispone de dos puntos donde recoger información de ruta (el punto 1 y el punto 6) aunque en cada intersección tiene que ver el estado de los semáforos.

Las paradas 2 y 3 realizan el mismo tipo de análisis y por lo tanto a nivel de resultado es indistinto si la prueba se realiza en la máquina 2 o 3. Para decidir que máquina realiza los análisis se mirará la carga de trabajo de cada una y se decide el destino más óptimo para balancear la carga a la hora de enviar las órdenes al AGV.

Las órdenes a ejecutar por el AGV se compondrán de una lista de paradas en las que se debe realizar un análisis. Debido a que el sistema está gestionado externamente, no se ofrecerán alternativas en la ruta ya que la decisión que deben tomar para elegir la máquina de destino ya estará tomada con el fin de balancear cargas de trabajo.

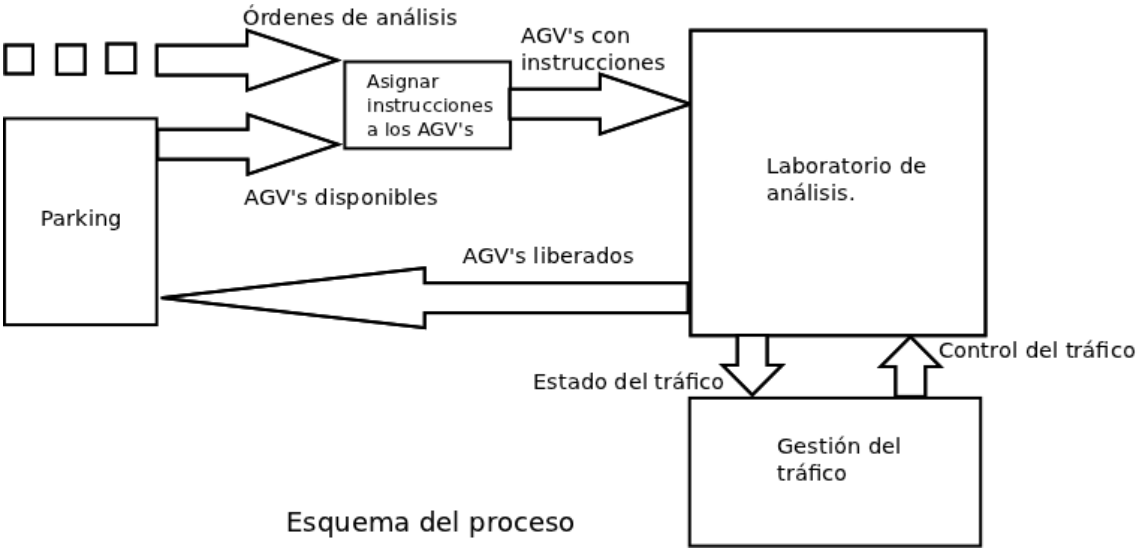
2.4 Problemas

Controlar el tráfico. Llevar el control de los vehículos que hay en cada tramo decisivo del circuito como por ejemplo entradas a máquinas de análisis.

Intersecciones. Control de quien pasa mediante semáforos adecuados al tipo de tráfico que circula por cada intersección. Controlando en cada caso la saturación del tráfico, la preferencia de los vehículos o el tiempo que llevan esperando.

Rutas. Controlar la carga de trabajo de las máquinas que realizan una misma función para que se asignen los destinos más óptimos.

2.5 Modelo del controlador



Esquema del proceso

Situándonos en el contexto de la simulación del laboratorio de análisis de sangre consideramos que, desde una fuente externa, nos llegan las muestras de sangre con la lista de análisis que se le debe realizar. A medida que disponemos de AGV disponibles en el parking (a la izquierda del esquema del proceso) se le carga la muestra de sangre y la lista de destinos que son las máquinas que realizan cada tipo de análisis. Una vez finalizado el proceso de carga y comunicación de los destinos al AGV, este entra en el laboratorio de análisis.

Dentro del laboratorio de análisis, la gestión del tráfico se organiza de forma concurrente por diversas máquinas de estado que operan de forma paralela y local a la zona que controlan. Estas miran el número de AGV que hay en los tramos que controlan, la prioridad, el tiempo que llevan esperando y/o la capacidad o disponibilidad de los tramos a los que desembocan decidiendo así si abrir un camino, otro o cortar ambos. En las bifurcaciones también es necesaria la comunicación con el AGV para verificar el estado del tramo al que desea acceder el AGV.

También se controla el número de vehículos que pasan de cada rama en las uniones para evitar que se creen casos de vehículos parados demasiado tiempo por no darse un caso de prioridad sobre el otro tramo. Esto hace que, además de con los AGV, la gestión de tráfico (identificado en la parte inferior derecha del esquema del proceso) tenga que comunicarse con el laboratorio para controlar el número de AGV que pasan por los puntos de unión.

Con toda esta información, desde la gestión de tráfico se retorna el estado de los semáforos que permiten o impiden el paso por determinados puntos a los AGV. Es en esta gestión donde se requiere mayor testeo debido a que cambios con el número de vehículos, velocidad y distancia de seguridad entre AGV pueden hacer más optima una gestión distinta.

Los AGV tras realizar el análisis se mantienen circulando por dentro del laboratorio hasta que las máquinas de análisis dan el visto bueno de todos los análisis que han realizado. En caso de necesidad, el AGV volvería a la máquina que ha fallado el análisis a repetirlo.

Una vez realizados los análisis y confirmado que se ha realizado de forma satisfactoria, el AGV vuelve al parking para descargar las muestras que transporta y volver a ingresar en la lista de AGV disponibles para realizar una nueva asignación de tarea. En esta simulación no se tienen en cuenta necesidades de recarga de batería ni mantenimiento de los AGV. En un sistema real los AGV liberados deberían pasar un chequeo de estado de la batería y total de horas de uso para saber si han de recargar, han de someterse a las labores de mantenimiento o ingresar en la lista de AGV disponibles.

CAPÍTULO 3. Gestión del tráfico

3.1 Los semáforos

3.1.1 Descripción

Para este proyecto utilizamos tres tipos de algoritmos para controlar los semáforos de las uniones dependiendo de la zona donde están y el tipo de tráfico que le llega.

Se pueden catalogar en tres tipos

- Reactivos a la saturación y control de paso
- Reactivos a la prioridad.
- Reactivos por tiempo máximo de espera

A estos semáforos se les denomina reactivos ya que reaccionan ante una situación y no ante una previsión. Esto conlleva a que en la programación tenga que preverse las situaciones en las que se puede encontrar el semáforo y eso no siempre es fácil. En este caso las intersecciones son sencillas y lo único que se debe preocupar es evitar que un vehículo esté siempre o demasiado tiempo esperando. Para evitar las esperas demasiado largas, en estos semáforos se usan contadores de tiempo, distancia o de paso de vehículos, como se detalla a continuación.

Todos los semáforos tienen un aspecto en común. Antes de las uniones de ambos carriles hay un pequeño tramo que va desde el semáforo hasta la unión física de las líneas. Esto es debido a que los vehículos tienen un sistema para detectar obstáculos delante de ellos, pero no por el lateral. Si el semáforo se situase justo en la intersección de las líneas, los vehículos podrían colisionar lateralmente haciendo que uno o ambos se saliesen de la línea en el mejor de los casos. Colocando esta distancia de seguridad se evita que colisionen en la intersección con los que vienen por el otro carril.

En las intersecciones posteriores a las máquinas de análisis se usa el primer tipo de semáforo.

El semáforo reactivo por saturación y control de paso se basa en controlar cuantos vehículos están esperando en cada tramo y abrir el paso dependiendo de la saturación. En un principio el algoritmo controlaba la saturación de ambos carriles de la intersección, pero no era recomendable debido a que la saturación en el carril de salida de la máquina de análisis impedía a esta seguir trabajando, con lo que independientemente del estado del otro carril, hay que descongestionar esté lo antes posible. El control de paso simplemente es un sistema para evitar esperas demasiado largas. Cuando pasa un cierto número de vehículos por un carril y hay al menos un vehículo esperando en el otro el semáforo cambia para darle paso a los vehículos que esperan. Este control de paso es el mismo que se usa en el siguiente semáforo.

En las intersecciones posteriores a las zonas de adelantamiento se usa el segundo tipo de semáforo.

El semáforo reactivo a la prioridad es un tipo de semáforo adecuado para intersecciones donde casi todo el tráfico viene por un carril y por el otro carril circulan vehículos con preferencia. El tráfico por el carril con preferencia por lo general es bajo y muy separado. Esto complica usar el tipo de semáforo anterior ya que antes de vaciarse el carril puede que entre otro vehículo que debe atravesar todo el carril para salir, alargando la espera. Para este se usa un control que verifica si hay vehículo en el carril y la distancia de este hasta la salida. Si la distancia es larga se abre el paso a los vehículos del otro carril y cuando llega a la intersección recupera la preferencia de paso.

Para asignar prioridad a un vehículo debe cumplir una de las dos condiciones:

- Tener que realizar al menos un análisis, fruto de haberlo realizado con anterioridad y que el resultado fuese erróneo, por lo que se debe repetir.
- Haber realizado todos los análisis con resultado satisfactorio.

Por tanto nos quedan sin prioridad los que están a la espera de resultados de los análisis. Dado que deben permanecer en el circuito hasta tener los resultados cogen el camino más largo y le dan siempre preferencia a los vehículos que si tienen.

Este tipo de semáforo tiene un problema aparente. No se hace control alguno para garantizar que el tramo en el que los vehículos no tienen preferencia, estos podrían estar esperando demasiado tiempo. Pero el tráfico habitual en estos tramos es que casi todos pasan por el tramo sin preferencia. Por el otro carril, el de los que tienen preferencia, hay un control de distancia hasta el final. Si hay algún vehículo en el tramo y está demasiado lejos de la intersección, se vuelve a dar preferencia al carril sin preferencia hasta que el vehículo llegue al final del tramo siempre y cuando en ese carril haya algún AGV esperando para pasar. Por lo tanto, excepto en algún caso puntual de tráfico saturado, la espera en un carril no es demasiado larga, pero se puede excusar ya que los de ese carril están esperando.

En la entrada del laboratorio se encuentra el semáforo reactivo por tiempo máximo de espera.

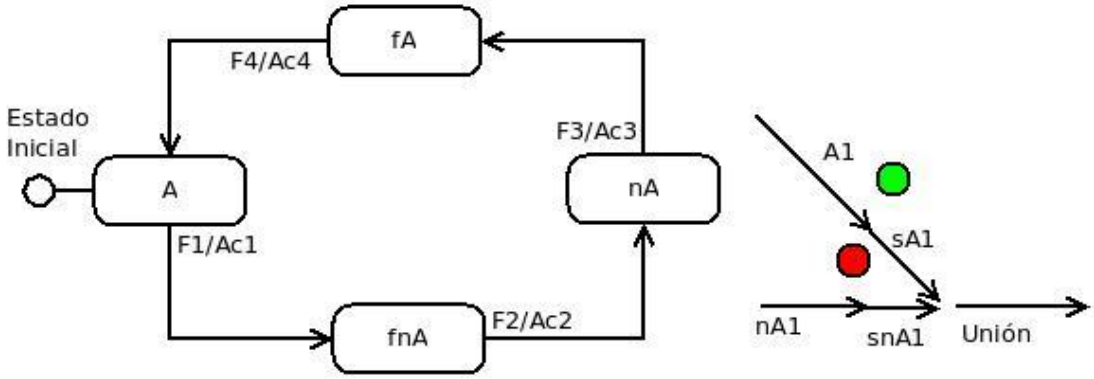
En este caso, el tráfico es tan variado que los dos anteriores no responden bien a este tipo de cruce. Aquí el tráfico puede llegar muy separado o muy compacto tanto en espacio como en tiempo. Para esto se usa un semáforo con control de tiempo máximo. Cuando llega un vehículo al carril sin preferencia se activa un nuevo estado. Si transcurrido un tiempo, el vehículo en espera no ha recibido la paso, se cambia para dar paso a este. Funciona así en ambos carriles de la intersección.

3.1.2 Funcionamiento interno (pseudocódigo)

En este apartado se muestran las figuras/esquemas de las máquinas de estado junto a una descripción de que realizan y que activa sus transiciones. Al lado se adjunta una captura de las intersecciones que controla y los nombres abreviados de los tramos para referirse a ellos en el pseudocódigo.

Dicho pseudocódigo se basa en dar una descripción de las acciones que realiza el gestor de tráfico para decidir a qué carril da paso. Por esa razón no se utiliza ningún lenguaje de programación como C o Java para hacerlo más entendible a los lectores que no conozcan dicho lenguaje y porque la acción que realiza es independiente del lenguaje de programación además de que en algunos puede requerir más acciones de las descritas para conseguir un correcto funcionamiento.

A continuación se muestran los diferentes tipos de semáforos.



Semáforo reactivo a la saturación y control de paso

En esta imagen podemos ver la máquina de estados que controla los semáforos que se sitúan en la unión que hay tras las máquinas de análisis con el carril por donde pasan los vehículos que no han realizado dicho análisis. A la derecha de la máquina de estados está una sección del circuito con dicho semáforo.

El estado inicial (A) en el que se encuentra el laboratorio al arrancar es que la salida del carril proveniente de la máquina de análisis (A1) está abierto y el otro (nA1) cerrado.

F1, F2, F3 y F4 son las funciones lógicas que validan el cambio de estado mientras que Ac1, Ac2, Ac3 y Ac4 son las acciones que realizan cuando se produce un cambio de estado.

En las funciones usamos una variable "contador" que se incrementa con cada vehículo que pasa de A1 a sA1 o de nA1 a snA1.

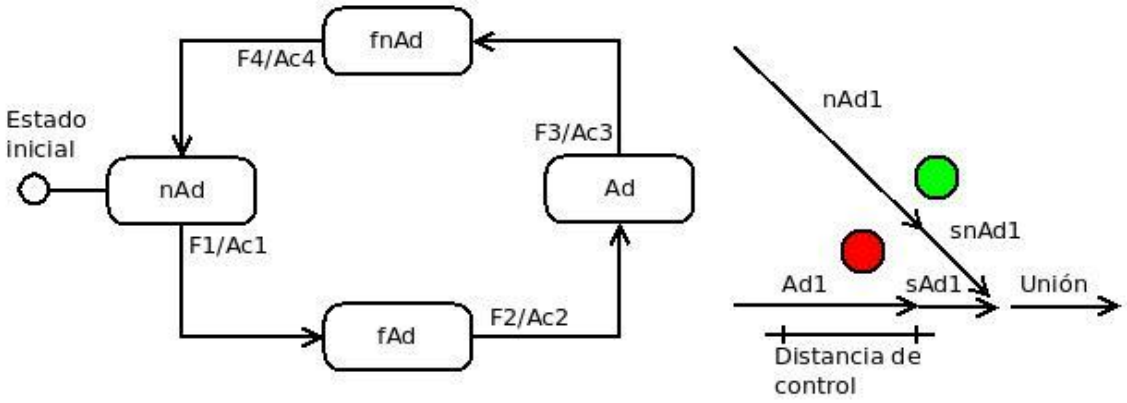
Para entender las siglas, A1 es la línea que hay tras la máquina de análisis. El 1 es simplemente para distinguirla del estado de la máquina de estados A.

nA1 es la línea que siguen los vehículos que no han pasado por esta máquina de análisis.

sA1 es la salida de la línea A1 y snA1 es la salida de la línea nA1. Son los tramos de seguridad que evitan las colisiones laterales entre los vehículos que se explica en el apartado 3.1.1.

Unión es la línea que sigue tras la unión de sA1 y snA1.

- F1 : Se ejecuta si :
    - Contador es mayor o igual a 5 y en el otro carril hay vehículos esperando.
    - En el carril A1 no hay vehículos y en nA1 si
- Ac1 : Cuando se cumple F1 se bloquea el semáforo de A1 y se pone contador a 0
- F2 : Se ejecuta si :
    - En sA1 no hay vehículos y en Unión se puede entrar
- Ac2 : Cuando se cumple F2 se abre el semáforo de nA1
- F3 : Se ejecuta si :
    - En A1 hay tres o más vehículos esperando
    - Si el contador es mayor o igual a 5 y en el carril A1 hay vehículos esperando
    - Si no hay vehículos en nA1 y si hay en A1
- Ac3 : Cuando se cumple F3 se bloquea el semáforo de nA1 y se pone el contador a 0
- F4 : Se ejecuta si :
    - En snA1 no hay vehículos y en Unión se puede entrar
- Ac4 : Cuando se cumple F4 se abre el semáforo de A1



Semáforo reactivo a la prioridad

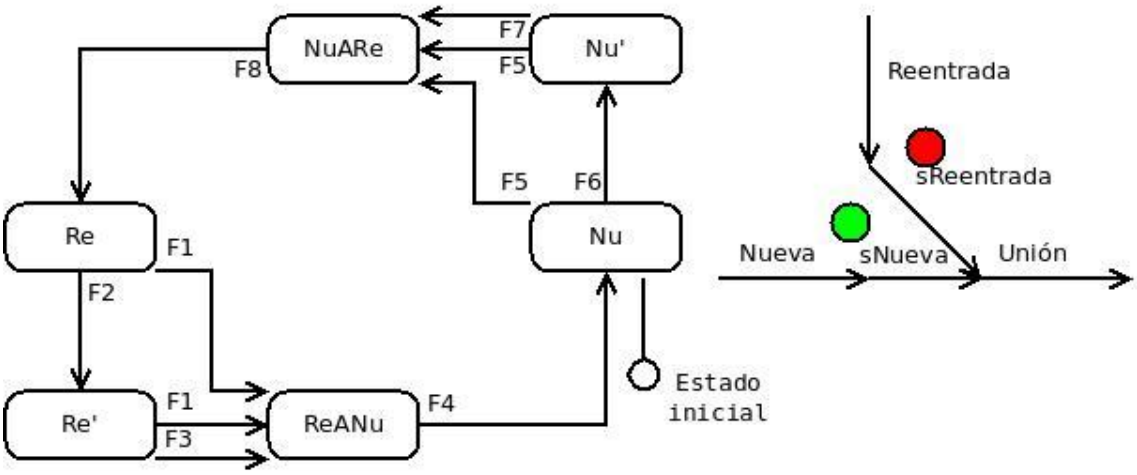
En la imagen situada justo arriba podemos ver la máquina de estados del semáforo que controla las uniones de los vehículos que circulan sin prioridad y los que sí.

Esta máquina de estados requiere una única variable, "salidaAdelantamiento", la cual solo admite los valores cierto/falso.

Al llegar al final del tramo con preferencia, Ad1, la variable adquiere el valor cierto y al salir del tramo vuelve a ponerla en falso.

El funcionamiento del esquema es como el anterior, F son las funciones que activan el cambio de estado y Ac las acciones que realizan en el cambio de estado.

- F1 : Se ejecuta si :
  - La variable salidaAdelantamiento está con valor true. Esta situación solo se da cuando hay un vehículo al final de Ad1 esperando a que el semáforo le dé preferencia.
- Ac1 : Cuando se cumple F1 se cierra el semáforo de nAd1
- F2 : Se ejecuta si :
  - No hay vehículos en snAd1 y hay espacio en Unión para entrar.
- Ac2 : Cuando se cumple F2 se abre el semáforo de Ad1
- F3 : Se ejecuta si :
  - Hay al menos un vehículo en nAd1 y en Ad1 el vehículo más cercano a la salida está a una distancia superior a la "Distancia de control"
  - Si en nAd1 hay algún vehículo y en Ad1 no hay vehículos
- Ac3 : Cuando se cumple F3 se cierra el semáforo de Ad1
- F4 : Se ejecuta si :
  - No hay vehículos en sAd1 y hay espacio en Unión para entrar.
- Ac4 : Cuando se cumple F4 se abre el semáforo de nAd1



Semáforo reactivo por tiempo máximo de espera

En esta máquina de estados tenemos dos estados nuevos y una diferencia con las máquinas anteriores. En estas las transacciones no realizan acción alguna cuando se disparan. En su lugar son los estados los que realizan una acción cuando entran. La mejora es que algunos estados llegan desde tres caminos posibles y en los tres habría que poner la acción a realizar. Hecho que conlleva a la duplicación de código y mayores problemas si se ha de cambiar. Al poner la acción en el estado, es indistinta la transición por donde se llega, simplemente importa al estado en el que se llega.

Las acciones que realizan los estados son

- Re : Abre el semáforo de Reentrada
- Re' : No realiza acción alguna
- ReANu : Cierra el semáforo de Reentrada
- Nu : Abre el semáforo de Nueva (nueva entrada de un vehículo al que se le acaba de asignar una nueva lista de análisis)
- Nu' : No realiza acción alguna
- NuARe : cierra el semáforo de Nueva

Las funciones para disparar las transiciones son:

- F1 : Si hay vehículos en Nueva y no hay vehículos en Reentrada
- F2 : Si hay vehículos en Nueva y en Reentrada
- F3 : Si ha pasado un tiempo máximo y aun no se le ha dado preferencia al carril de Nueva
- F4 : No hay vehículos en sReentrada y se puede entrar en Unión
- F5 : Si hay vehículos en Reentrada y no hay vehículos en Nueva
- F6 : Si hay vehículos en Reentrada y en Nueva
- F7 : Si ha pasado un tiempo máximo y aun no se le ha dado preferencia al carril de Reentrada
- F8 : No hay vehículos en sNueva y se puede entrar en Unión.

3.2 Otros controles

Para la simulación en Anylogic es necesaria una serie de semáforos adicionales para controlar el acceso a las bifurcaciones y a los objetos delay. Al ser un tema de la simulación se explican con más claridad en el capítulo dedicado a Anylogic.

CAPÍTULO 4. Modelo en Anylogic

4.1 Qué es

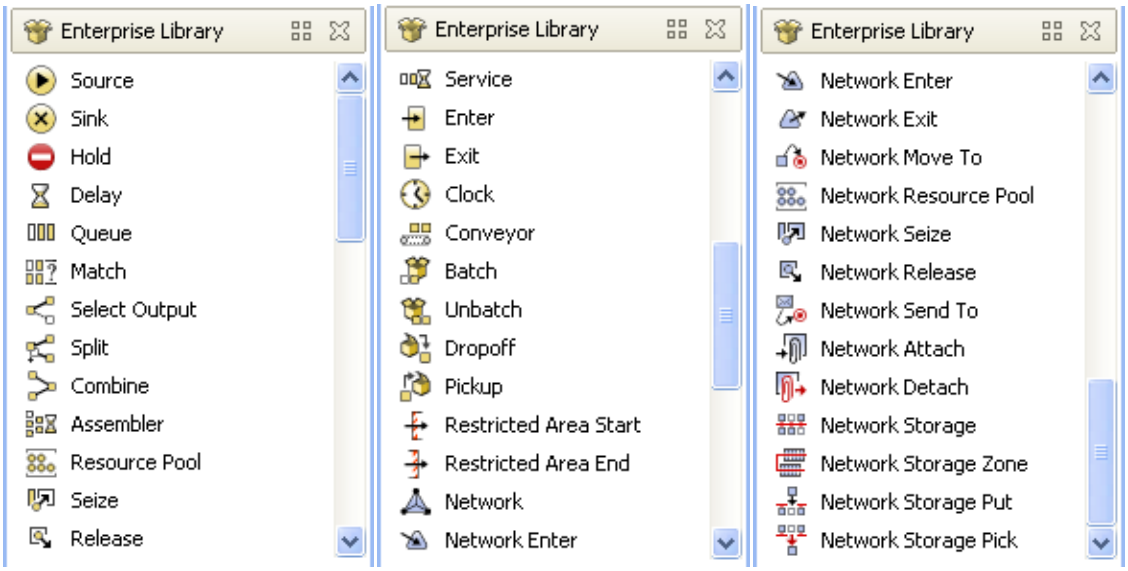
Anylogic es un software de simulación de XJ Technologies que permite, entre otras cosas, crear fácilmente una animación para ver el comportamiento de los procesos. Aunque se puede hacer en 3D, para esta simulación se ha implementado en 2D, mucho más fácil de crear y con un consumo de recursos por parte del PC muy inferior.

La programación se realiza mediante instrucciones en lenguaje java, dando la opción de crear clases ya sean nuevas o heredadas de las que ofrece Anylogic. Esto permite, en especial en este proyecto, agregar funciones a la clase que representa los vehículos para facilitar las consultas de ruta y decidir los caminos que deben seguir, modificar el estado de los análisis realizados o por realizar, asignar de forma específica la ruta a seguir o, para la simulación, generar rutas aleatorias, etc.

Anylogic viene con varios paquetes de clases que nos facilitan la implementación de la simulación. Entre ellos está el paquete Enterprise, que nos ofrece la mayoría de los objetos para realizar el control de tráfico en este tipo de simulación.

4.2 Objetos

Los objetos que usamos en la simulación, como se especifica antes, están casi todos en el paquete Enterprise, pero también se requieren los paquetes General, Analysis, Statechart y Presentation.



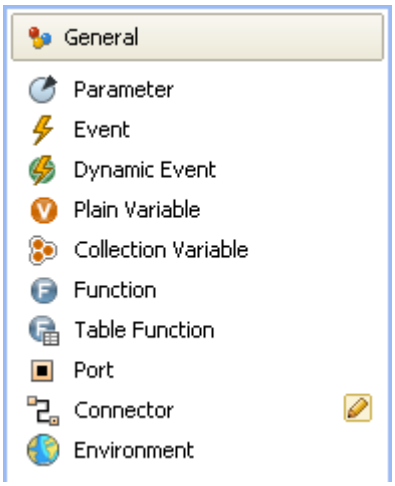
En Enterprise tenemos los objetos:

- Source: Genera objetos de la clase especificada (por defecto Entity, pero como necesitamos funciones adicionales, creamos la clase BloodCarManagement heredada de Entity que nos permite seguir interactuando con las clases de este paquete y

agregar funciones y variables internas.) Este objeto permite configurar la forma en que va creando los objetos, ya sea mediante la llamada a una función, siguiendo una tabla, un intervalo de tiempo, etc. En el caso de esta simulación se crean a intervalos de tiempo pero se almacenan en una cola de espera. Dicha cola de espera sería lo que representa "AGV con instrucciones" en la figura Esquema del proceso en el punto 2.4.

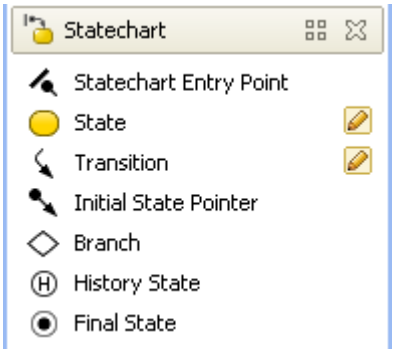
- Queue: Es la anteriormente mencionada cola. A modo de simulación representa un almacén para guardar componentes. Aquí se usa al principio para almacenar los objetos generados por source sin preocuparnos de los fallos que pueda generar por intentar introducir un objeto en un conveyor donde no hay espacio. Esta acción genera un error en la simulación y es por tanto un factor muy a tener en cuenta a la hora de gestionar el tráfico ya que durante el trayecto no se usan colas para almacenar AGV, igual que ocurriría en un caso real.
- Conveyor: Es el seguidor de líneas de nuestros AGV. Se le asignan las líneas que han de seguir, la velocidad y la distancia mínima que deben guardar los AGV entre ellos. En la realidad estos objetos son la programación de los AGV para seguir las líneas y unos sensores para detectar cuando entran, salen o hay espacio suficiente para que entre otro AGV en la línea ayudado de un pc externo para almacenar y modificar estas variables.

  Esta clase ofrece las funciones .size() que nos muestra el número de vehículos que hay dentro del tramo, la función .canEnter() que nos devuelve un booleano (cierto o falso) avisando si hay espacio suficiente en la entrada para que entre otro vehículo y .spaceAhead(0) que nos da la distancia del primer vehículo hasta el final del tramo. Además nos permite realizar acciones "on enter" al entrar un vehículo, "at exit" cuando un vehículo está preparado para salir y "on exit" al salir el vehículo. Dependiendo del tramo pueden ser necesarias unas acciones u otras.
- Hold: Barreras que permiten el paso o no. La correcta gestión de estas barreras es el objetivo de este proyecto. Para bloquearlas o permitir el paso tienen la función .setBlocked(true/false) y nos ofrece la posibilidad de realizar una acción "on exit".
- SelectOutput: Permiten enviar los vehículos por uno de los dos caminos que tienen a la salida. Representan las bifurcaciones del modelo real. La elección se hace mediante una respuesta de true/false a una condición (esta clase permite también la elección del camino mediante una función aleatoria con un porcentaje de envíos por uno u otro camino, pero en nuestra simulación se usa la opción de "por condición"). Estas bifurcaciones no permiten que un vehículo se quede en ellas a la espera de tener espacio para entrar al siguiente tramo, por lo que se les precede de un hold que impide el avance hasta que se garantiza que en el tramo de destino hay espacio (usando la función .canEnter()).
- Delay: Este objeto simula un tiempo de proceso. Simulamos que los AGV no siempre consiguen aparcar con exactitud a la primera y por eso algunos tardan más que otros. Este objeto nos permite retener un vehículo durante un tiempo. Dicho tiempo puede ser fijado o mediante una función normal que da un tiempo de media con un margen de error (en el caso del proyecto se ha puesto que tarda de media 2 segundos pero el tiempo que pasa va de entre 1 y 3 segundos).



En General tenemos los objetos:

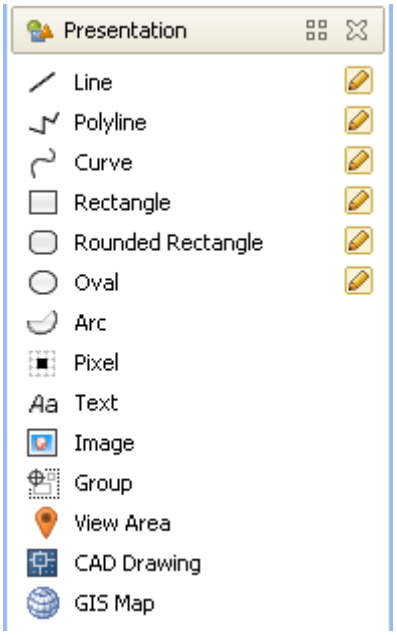
- Plain variable: Variable global. Si se declara pública, cualquier objeto o función la puede ver y modificar.
- Function: Función global. Igual que la variable, si es publica cualquier objeto u otra función la puede llamar. Se debe especificar el tipo de retorno (o void en caso de no retornar nada) y los parámetros con los que se llama.
- Dynamic event: Eventos que se crean en un momento y tras el tiempo que se le fija (primer parámetro cuando se crea) se ejecuta. Se pueden pasar parámetros adicionales, estos parámetros van detrás del tiempo en segundos que debe transcurrir para que se ejecute. En el proyecto se crean eventos dinámicos cuando se realiza un análisis y cuando se ejecutan simulan que la máquina ha terminado de realizar el análisis, comunicando al AGV si debe repetir el análisis o si ya está listo para devolver la carga.
- Collection variable: Es la ArrayList de Java. Nos permite almacenar, buscar y extraer objetos de un tipo determinado. Esta clase nos permite trabajar sin necesidad de saber el tamaño del vector como nos exigiría en el caso de estar trabajando en C o C++.



En Statechart tenemos los objetos

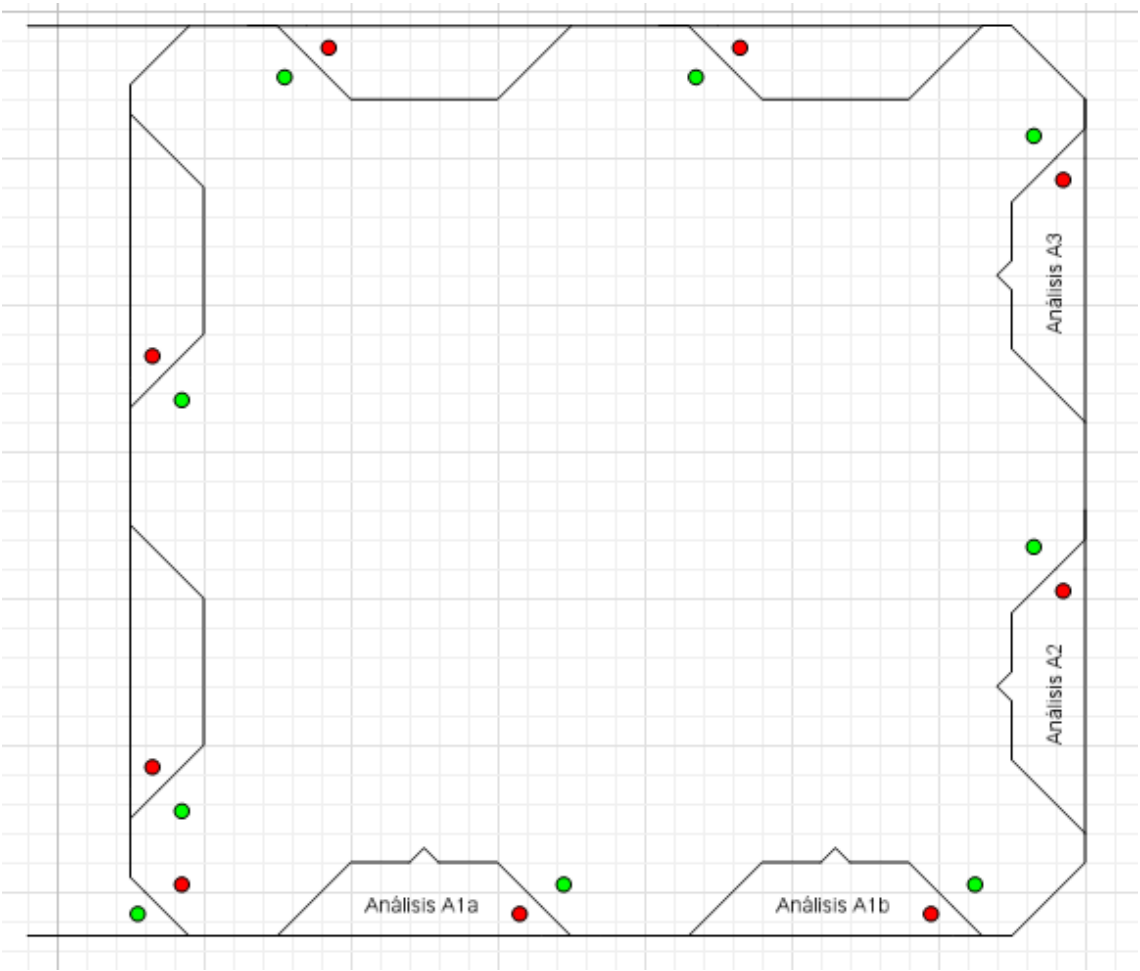
- Statechart Entry Point: Dentro de una máquina de estados nos indica el estado en el que se encontrara la máquina al iniciarse la simulación. Es imprescindible uno de estos en cada máquina o Anylogic mostrara un error.

- State: Uno de los estados de la máquina de estados. Nos permite realizar acciones al entrar y al salir del estado. En la simulación los estados no realizan acción alguna ya que en esta simulación, dichas acciones vienen determinadas en las transacciones de un estado a otro.
- Transition: Son las anteriormente nombradas transacciones. Tienen varias opciones para efectuar los cambios de un estado a otro. Las condiciones para activar la transición que se usan en este proyecto son:
    - Time out : se le especifica un tiempo y transcurrido ese tiempo (si sigue en el mismo estado) salta al estado de destino. En el proyecto se usan en dos casos. Uno es el de refresco. Cuando una máquina de estados cambia de un estado a otro (o al mismo) vuelve a evaluar todos los estados para ver si se deben actualizar así que una de estas transiciones se usa para ir refrescando las máquinas de estado. El otro caso que se usa es el de control de tiempo de espera. Cuando un vehículo lleva demasiado tiempo esperando a que le den preferencia en un semáforo salta esta transición y le da paso. De cara a la simulación, este caso es más útil ya que el otro simplemente es para controlar el refresco.
    - Condition : Salta cuando la condición es cierta. Puede ser una variable con valor cierto o falso que algún otro elemento controle su valor, puede ser un conjunto de condiciones (a<3 && b == 0) cuyo resultado de cierto o falso o puede ser una llamada a una función que retorne el valor adecuado. Hay que tener presente que los cambios en las funciones que evalúan la condición de salto no se reevalúan al hacer algún cambio, por lo tanto es necesario siempre una transición de tiempo para poder reevaluar en caso de cambio.

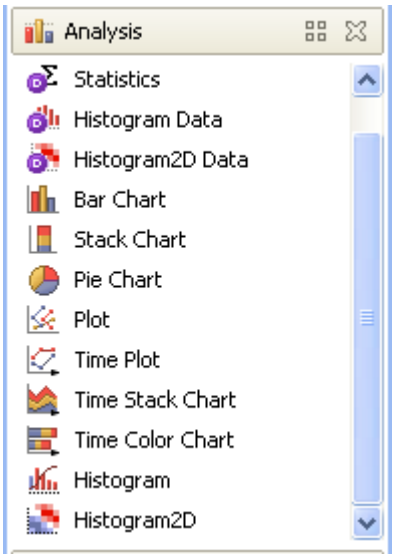


En Presentation tenemos los objetos

- Polyline: Este objeto nos permite realizar rectas enlazadas de tantos tramos como sea necesario. Aunque hay un objeto Line que permite realizar una recta de un solo tramo, no se usan en este proyecto debido a que la clase conveyor requiere un polyline para asignarle la guía del recorrido.
- Oval, rectangle: Son objetos con el único fin de crear dibujos en la simulación. Entre las configuraciones que ofrece están el color interno y el de la línea de borde. La función .setFillColor(Color color) permite cambiar el color de relleno de estos objetos. En el proyecto se usan con el único fin estético de cambiar los colores simulando un semáforo.
- Group: Seleccionando varios objetos de esta librería, podemos hacer luego un group. Este objeto junta en uno varios y permite usarlo como un elemento único. En este proyecto, el vehículo que se ve en la simulación es un conjunto de rectángulos y óvalos agrupados. Esta agrupación permite también escalar el conjunto en proporción.
- Text: Sirve para insertar texto. Permite elegir tipos de letra, tamaño y estilos. Una vez insertados se pueden girar como en los ejemplos de Análisis A2 y Análisis A3.



Mapa del laboratorio representado por objetos de Presentation



En Analysis encontramos

- Plot: Se puede configurar los valores de ambos ejes o dejarlo en modo automático donde el propio Anylogic se encarga de ajustarlo. El valor o función que muestra y cada cuanto quieres que se haga una lectura. El tamaño de las líneas y el formato. Tiene más funciones pero no se usan en este proyecto

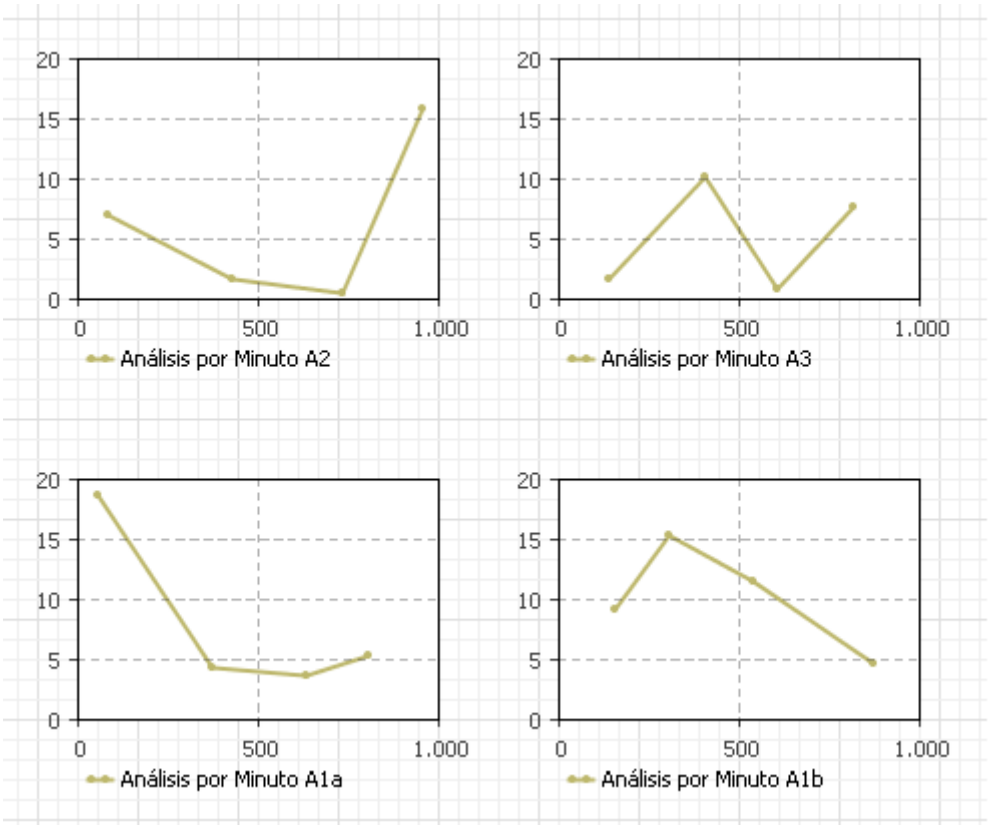


Imagen de varios Plot's

4.3 Controles

Previamente se explican los algoritmos y los objetos que se usan dentro de Anylogic. Aquí veremos algunos ejemplos de cómo se ensamblan para conseguir los fines que deseamos.

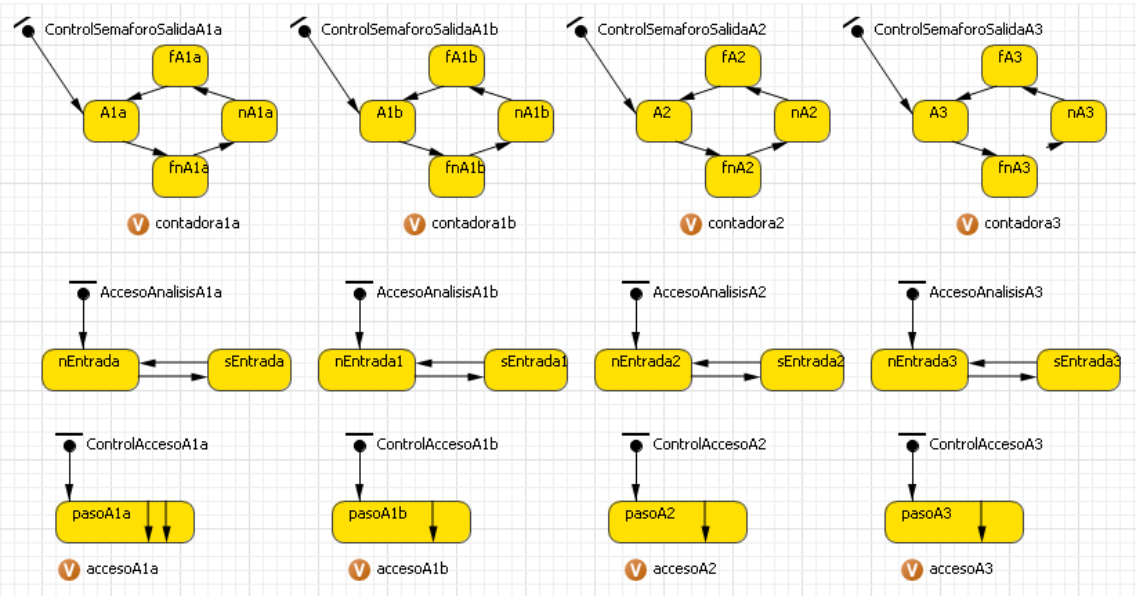


Figura de los controles de tráfico en zonas de análisis.

Esta imagen muestra las máquinas de estado que controlan las zonas de los análisis. Cada una de las 4 columnas de grupos de máquinas de estados controla un tramo. Externamente tienen el mismo aspecto (exceptuando la que está situada abajo a la izquierda) e internamente tienen un código muy parecido.

La máquina de abajo a la izquierda tiene dos transiciones, representadas por flechas, a diferencia de las otras similares que tienen una. La transición adicional es una transición que se dispara cada 0.1 segundos. Esta transición no efectúa acción alguna. Su función en cambio es más importante. Cuando una transición salta, se re-evalúan todas las demás transiciones. Sin esta transición, llegado el momento en que todas las máquinas lleguen a un punto estable, no se ejecutarían aunque las condiciones de las transiciones varíen de forma que si se deberían ejecutar.

La fila superior es la que se encarga de las bifurcaciones que hay a la entrada de los tramos de análisis. El funcionamiento está explicado en el capitulo anterior.

La segunda fila controla el acceso a la máquina de análisis. El tramo en forma piramidal que hay en la parte superior del tramo. Ésta máquina de estados es necesaria por la forma de funcionar de Anylogic. Los objetos delay no permiten el almacenamiento de objetos por lo que no se puede dejar entrar un objeto si no se garantiza que hay espacio detrás como para poder salir. Por poner un ejemplo, simularía un horno en el que los objetos han de permanecer un tiempo fijo. Si a la salida del horno no hay espacio para que el objeto salga sufriría daños. Este no es el caso que tratamos, pero como el funcionamiento es así y la recomendación que da Anylogic de poner un estante para almacenar no cuadraría con la realidad, ya que no tenemos donde almacenar los AGV si no es en el tramo de circulación.

La tercera fila también se debe a un objeto de Anylogic que no permite el almacenamiento en su interior. En este caso el selectOutput. Cuando un AGV termina un tramo e intenta entrar a otro donde no hay espacio, este se detiene en el anterior sin entrar en el siguiente. En las bifurcaciones tenemos el problema de que entre el tramo de donde sale y cualquiera de los dos donde debe entrar hay que poner un objeto selectOutput. En este caso el AGV sale del tramo, entra en el selectOutput y en ese momento es cuando se puede encontrar con que no puede entrar en el tramo. Esta situación produce un error en la simulación deteniéndola.

Para evitar este fallo, antes de salir del tramo el AGV indica que camino va a tomar y un semáforo impide el paso hasta que en el tramo de destino hay espacio para que pueda entrar el AGV. Tras la entrada en el tramo el semáforo vuelve a cerrarse para controlar el acceso del siguiente AGV.

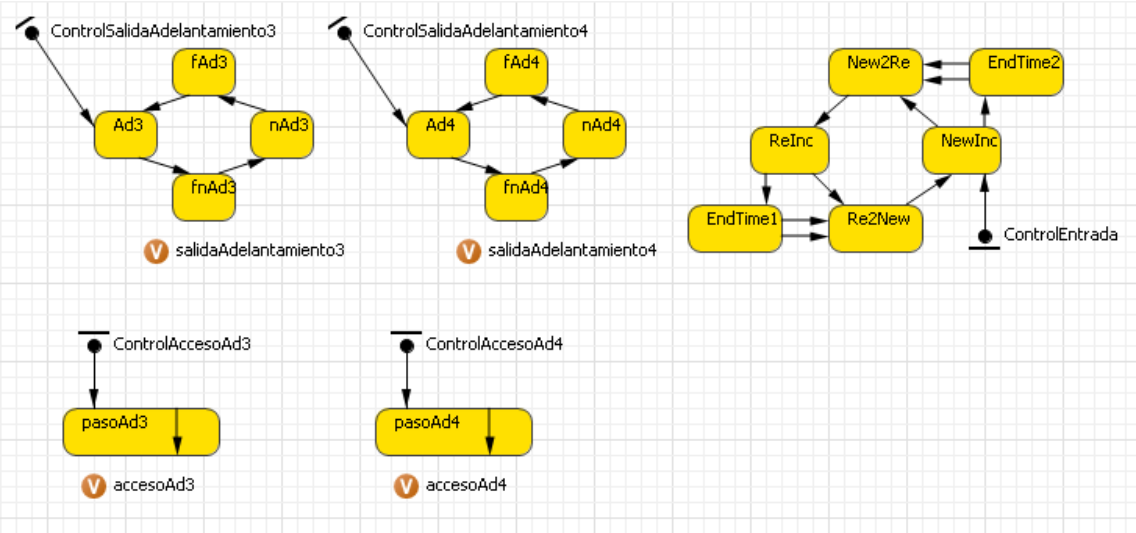


Figura de control de tráfico en zonas de adelantamiento y reincorporación

El control de los tramos de adelantamiento es similar al de los tramos de análisis. La única diferencia es que no tenemos un delay y por tanto no hay que controlar su acceso.

En la parte derecha de la imagen se ve la máquina de estados que controla el semáforo de la entrada y la reincorporación. El funcionamiento se explica en el capitulo anterior.

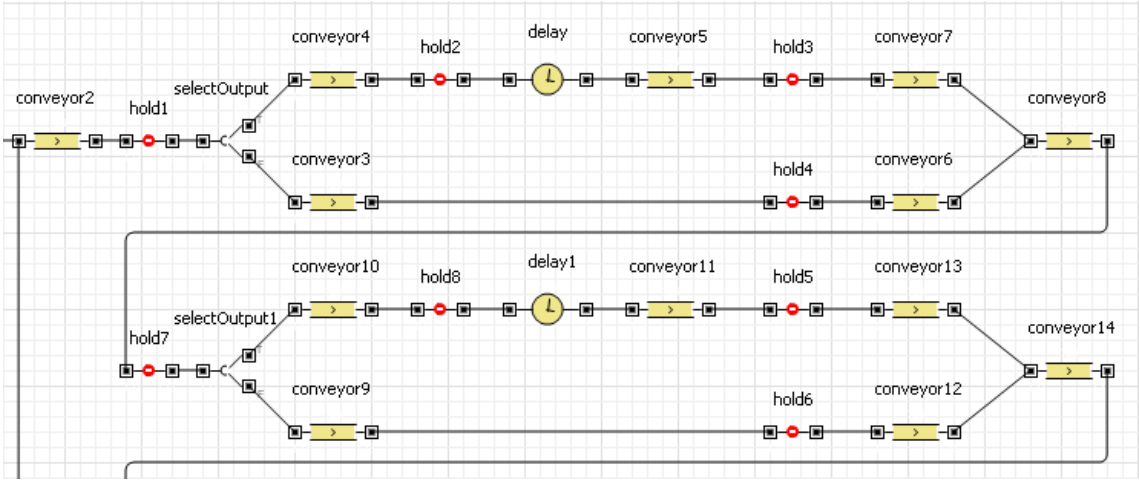


Figura con los actuadores de tráfico

En la imagen superior tenemos los componentes que controlan y actúan sobre el tráfico de dos tramos de las máquinas de análisis. Los AGV que han de realizar el análisis van por el tramo superior, donde está el objeto delay.

En la parte superior izquierda podemos ver conveyor2, hold1 selectOutput y de este dos salidas hacia conveyor4 y conveyor3.

Cuando un AGV llega al final del conveyor mira si puede salir y en tal caso abandona el conveyor hacia su siguiente destino. En este caso el siguiente destino es el hold1. Este es, como se indica anteriormente, un semáforo de bloqueo o paso. Si está abierto pasa al siguiente elemento, selectOutput. En este elemento tenemos un problema y es que ha de salir de dicho elemento si o si. Por lo tanto en caso de no haber espacio en el conveyor de destino, Anylogic devuelve un error. Para garantizar que hay espacio, tras pasar el AGV, este cierra el hold anterior y una máquina de estados lo vuelve a abrir cuando ve que el conveyor de destino al que va el AGV tiene espacio para albergar un nuevo vehículo. Esta función parece correcta a simple vista, pero durante la simulación de vez en cuando nos salta un error de que se ha intentado acceder a un destino donde no hay espacio mostrándonos que no funciona del todo correcto. Para encontrar la solución hay que analizar el funcionamiento interno del análisis. Al tener procesos concurrentes en alguna ocasión podemos tener

Simulación del AGV 1	Simulación del AGV 2
-AGV intenta acceder a un conveyor -La máquina de estados ve que el conveyor de destino está disponible para aceptar otro vehículo y por tanto abre el semáforo para que pase -El AGV pasa (cambio de simulación a otro AGV)	-espera -El AGV avanza hasta el final del conveyor e intenta acceder al siguiente -Como el vehículo anterior no ha cerrado el semáforo, este sigue abierto y por tanto pasa -El AGV se encuentra en el selectOutput y no tiene espacio para salir. -Envía mensaje de error y se detiene la simulación.

Para evitar este tipo de fallos hay que realizar la misma función pero de forma distinta eliminando la dependencia de actuación de los otros vehículos y forzando la ejecución en el orden correcto.

Simulación AGV
-El AGV se aproxima al final del conveyor -Cierra el semáforo -Al llegar al final, la máquina de estados valora el estado de destino del AGV y en caso de tener espacio para entrar abre el semáforo. -El AGV pasa.

Con este algoritmo prescindimos de las acciones de otro AGV realizando en esencia la misma función. La diferencia principal consiste en que el semáforo permanece abierto hasta que llega el AGV y el mismo se lo cierra, algo que visto externamente puede parecer ilógico si lo comparamos con una situación similar de tráfico. Por hacernos una idea, es como si al acercarnos a un peaje, este tenga el semáforo en verde y la barrera levantada dando la sensación de que se puede pasar y al llegar seamos nosotros los que pongamos el semáforo en rojo y bajemos la barrera hasta que alguien considere oportuno volver a levantarla y dejarnos paso. Pero para la simulación e incluso en una implementación real de la planta, el algoritmo es eficaz y cumple con su cometido.

Con el delay que hay por el camino de la salida superior del selectOutput ocurre algo similar. En este caso es una máquina de estados la que se encarga de bloquear o abrir el paso. El delay, al igual que el selectOutput envía un fallo si intenta salir y no hay espacio. La diferencia que lo hace más sencillo es que no hay que preocuparse de varias opciones de destino, siempre es la misma y por tanto solo hay que asegurarse de que esa tenga espacio. Aquí no hay problemas de que entren dos debido a que delay si que puede fijar una capacidad e impedir el paso al conveyor predecesor si dicha capacidad ha sido alcanzada (en nuestro caso es 1) pero es

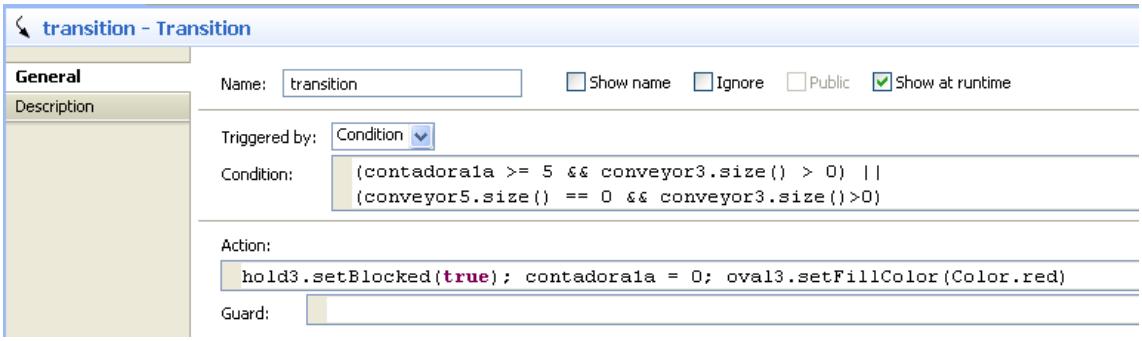
necesario un semáforo puesto que si puede aceptar una nueva entrada cuando no tiene espacio para que esta salga.

Antes de unirse los dos conveyors que hay al final del tramo encontramos un hold y un conveyor en cada camino. Estos holds son los que controla el semáforo de las intersecciones cuya explicación detallada del funcionamiento se encuentra descrita al principio de este apartado. Los conveyors que hay tras los holds se deben a la razón de que en las intersecciones los AGV no pueden detectar otros vehículos que se aproximen por el lateral y por tanto se necesita que un tramo del otro carril no contenga AGV para evitar las colisiones laterales ya que el detector de obstáculos solo funciona de forma frontal.

Las zonas de adelantamiento son muy similares. Simplemente no tienen la zona de hold y delay que tienen estas para las máquinas de análisis.

4.4 Capturas y código

Las capturas que se realizan en este apartado son, como en el resto del proyecto, de la versión 6.4.1 de Anylogic. Otras versiones pueden variar la distribución y el número de componentes configurables de cada elemento. De todas formas, los principales son prácticamente los mismos y como mucho cambian la presentación gráfica, pero mantienen el nombre y la utilidad.



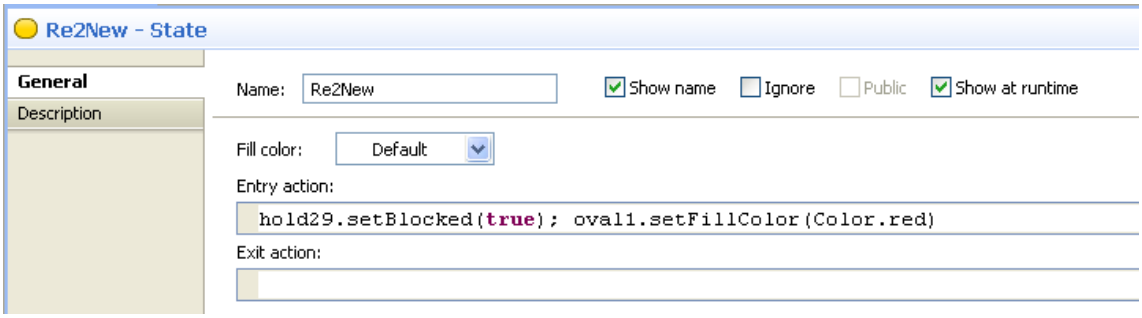
Captura de las propiedades de un transition

De una forma general, los objetos de Anylogic contienen una pestaña Description, debajo de la pestaña seleccionada en la imagen General. Esta pestaña es para todos igual y sirve para dejar un comentario del uso, explicación del código, la configuración o como muchas veces se usa, para guardar el código antes de hacer una modificación.

En la pestaña General vemos:

- Name: Es el nombre que recibe el elemento. Este tiene que ser único en todo el proyecto.

- Show name: En la pantalla del área de trabajo muestra el nombre del objeto o, como en este caso en el que no está seleccionado, no.
- Ignore: Si temporalmente no queremos que se dispare esta transición para poder realizar pruebas (como funciona el proyecto si se dispara con esta condición, como si se dispara con otra o que ocurre si dejamos las dos, por ejemplo)
- Show at runtime: si queremos que durante la reproducción de la simulación queremos que se vea o no.
- Public: En este caso nos aparece en gris y no nos deja seleccionarla. Por lo general esta opción es igual que en java, si queremos que se pueda acceder desde cualquier punto o no.
- Triggered by: En esta selección tenemos puesto condition. Esto es que la transición se dispara cuando ocurre una condición que especificamos a continuación. También tiene timeout, rate y message. Rate y message no se usan en este proyecto. Timeout se dispara transcurrido un cierto tiempo.
- Condition: En este recuadro tenemos la condición con la que se dispara la transición. Dicha condición debe dar como resultado un booleano true o false. Este puede venir como en el caso de la imagen de una secuencia de comparaciones y uniones lógicas agrupadas por paréntesis. También se puede realizar con una llamada a una función que retorne un true o false o directamente poner true o false.
- Action: Es la acción que realiza.
- Guard: Se evalúa y en caso de ser true se ejecuta el triggered by. No se usa en este proyecto.



Captura de las propiedades de un state

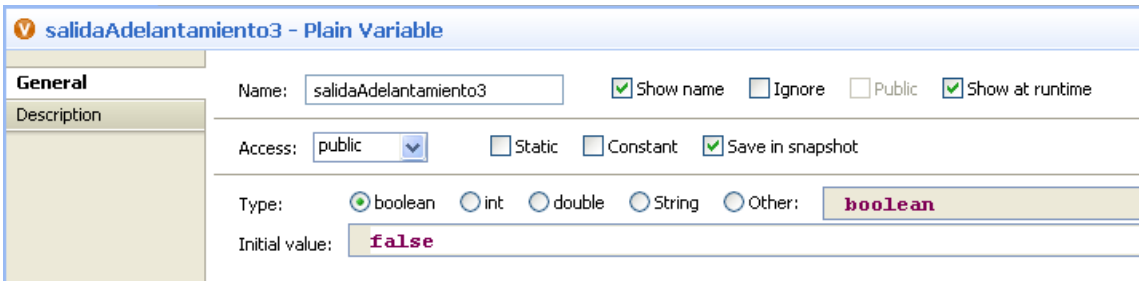
En esta captura vemos la configuración de un estado. En todo el proyecto los estados no suelen realizar nada exceptuando los de la máquina de estados que controla la reentrada y la entrada de nuevos AGV. La razón es, como se menciona antes, que a esta máquina se puede llegar de un estado a otro a través de varios caminos, por lo que es mejor actuar desde los estados que no repetir código. Pero se podría hacer como en las otras máquinas o cambiar las otras máquinas para que actuaran desde los estados y no desde las transiciones.

En esta pestaña podemos ver:

- Fill color: Solo tiene un efecto visual. Sirve para poner el color de fondo del estado al gusto. Por defecto es amarillo como se ve en capturas anteriores.
- Entry action: Acción que se realiza al entrar al estado. Como se ve en la captura, la ultima instrucción de cada bloque no lleva “;”. En algunas ocasiones, poner el “;” daba

un error ya que lo pone Anylogic por defecto y se encontraba con casos de “;;” y daba error. En otras ocasiones poner un “;” al final no produce fallo alguno.

- Exit action: Acción que se realiza al salir del estado.

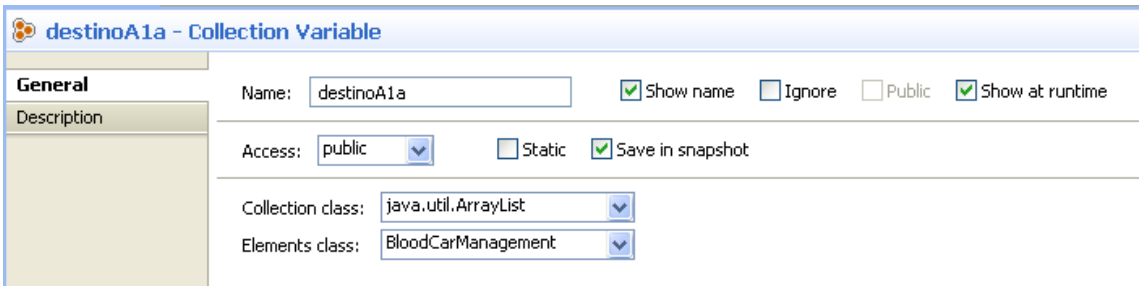


Captura de las propiedades de un Plain Variable

En esta captura vemos la configuración de una variable global simple.

En esta pestaña podemos ver:

- Access: Igual que en java, podemos seleccionar, public, prívate, protected y default.
- Static: Solo es útil si estamos tratando un objeto activo el cual se pueda duplicar en un proyecto. Con esto conseguiríamos que la variable tenga el mismo valor en todos los objetos, al igual que ocurre con las declaraciones static de java.
- Constant: En este proyecto se usan dos variables declaradas constant. Una de ellas es la velocidad de los AGV, lo que permite que cambiando el valor se modifiquen todos los elementos. Se podría hacer el caso anterior sin declarar constant, pero con esta selección aseguramos que no se modifica el valor durante la simulación.
- Save in snapshot: El estado de la simulación se puede grabar en un archivo para posteriormente recuperarla. Al marcar esta casilla indica que el valor de la variable es importante y por tanto se debe almacenar. En el proyecto no se hace nada de esto, pero viene activada por defecto.
- Type: indica el tipo de la variable.
- Initial value: valor inicial de la variable. Se puede dejar en blanco, pero si se inicia se tiene que dar un valor valido.

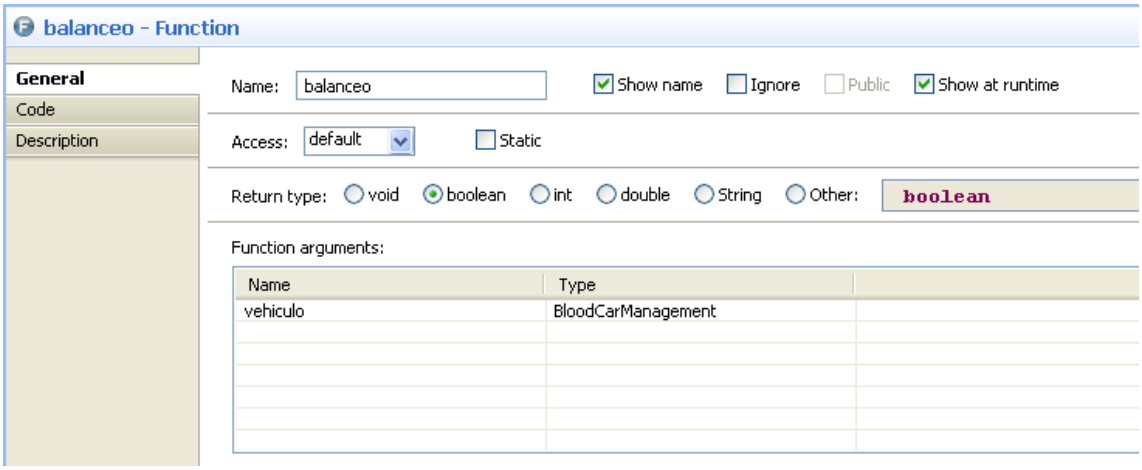


Captura de las propiedades de un Collection Variable

En esta captura vemos la configuración de una Collection Variable. Estas variables permiten almacenar una lista dinámica en la que podemos introducir o sacar elementos. En este ejemplo almacenamos variables del tipo BloodCarManagement. Esta clase es heredada de Entity y da la

funcionalidad a los AGV. Esta variable se usa para controlar que vehículos están destinados a la primera máquina de análisis A1a o a la segunda (la lista de vehículos que se destinan a la segunda máquina es otra lista idéntica exceptuando el nombre).

La clase de la collection es java.util.ArrayList. Esto nos ofrece unas funciones básicas como agregar, buscar si un elemento está en su lista, eliminar un elemento o eliminar todos los elementos de la lista. Con estas funciones podemos manejar la lista para el proyecto, pero si necesitáramos funciones adicionales se puede crear una clase java que herede de ArrayList a la cual le podamos hacer otro tipo de consultas. Un ejemplo sería mirar con una sola línea de código si dos elementos están en la lista. Se puede hacer con más líneas de código con las herramientas básicas que proporciona ArrayList, pero de este modo aumentamos la abstracción del código y reducimos su tamaño.

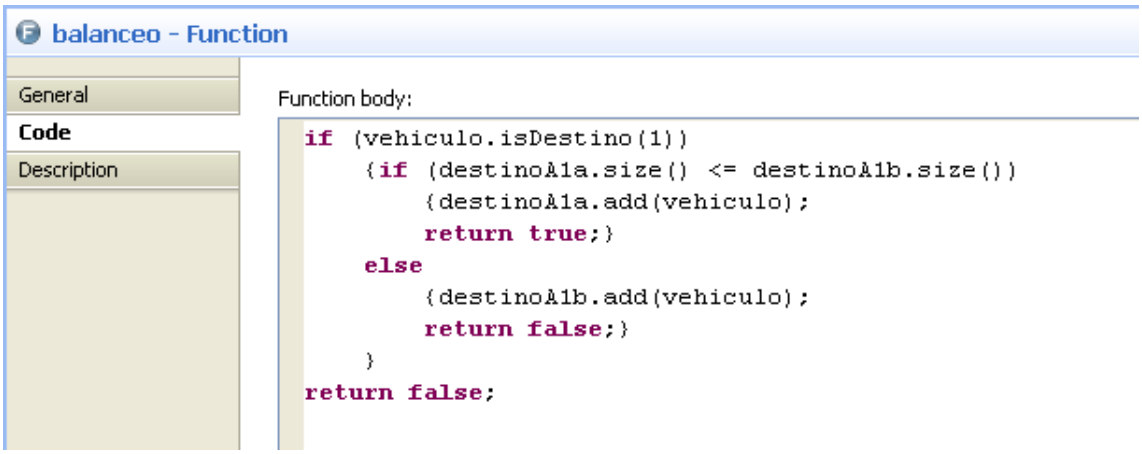


Captura de las propiedades de una Function

Estos objetos de Anylogic nos permiten crear funciones que pueden llamar y ejecutar el resto de los objetos. Esta en concreto es la que se usa para distribuir la carga de trabajo de las máquinas de análisis A1a y A1b.

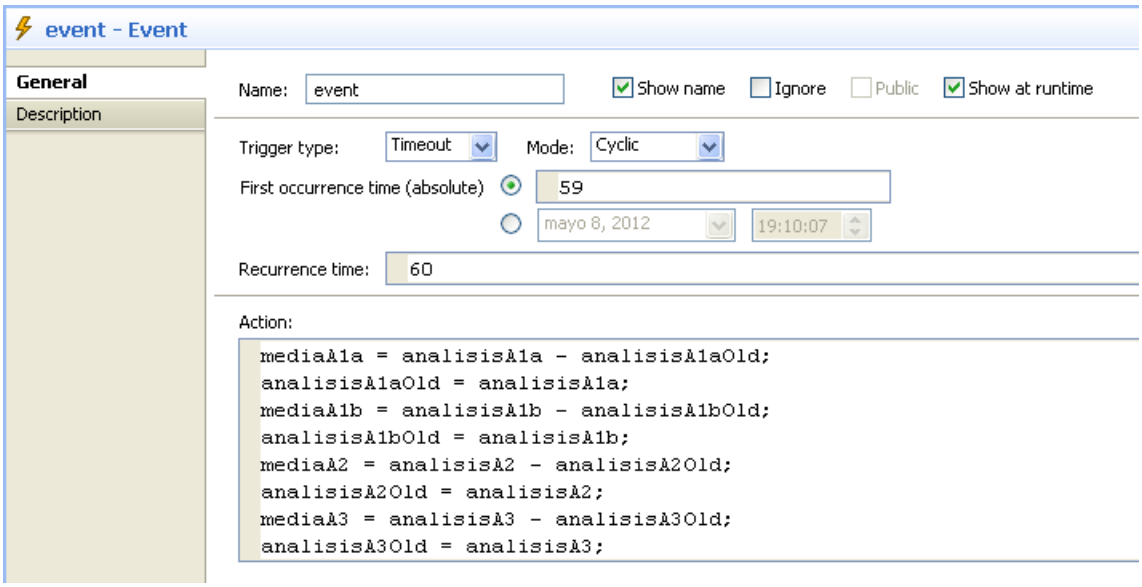
En esta pestaña podemos ver

- Return type: Indicamos el tipo de retorno que dará la función. En este caso un boolean.
- Function arguments: Indica los argumentos que se le deben pasar a la función para ser llamada. El primero es “Name”, que será el nombre que usaremos en el código para referirnos a ese argumento y el segundo es “Type” que definirá de que tipo es ese argumento.



Captura de el código de una Function

Los objetos Function tienen una pestaña más que los que se han visto hasta ahora. Code es donde se escribe el código de la función. Aquí podemos ver como se usa el argumento vehiculo y una de sus funciones ".isDestino(1)" para saber si el vehículo debe hacer el primer análisis. Tambien accedemos a una Collection Variable (explicada su función anteriormente) para mirar el número de elementos que contiene y agregar un elemento a una de ellas en caso de ser necesario. Como la función tiene Return type boolean tenemos que devolver en cualquier caso un valor true o false, que se almacena en una variable que usa un selectOutput para decidir qué camino ha de tomar el AGV. En una primera versión del código no se usaban las ArrayList y por tanto era imprescindible el retorno de true o false para indicar si debía ir o no a la primera máquina a realizar el análisis. Con las ArrayList esto se podría evitar en los SelectOutput, pero las máquinas de estado requieren una variable con el destino del AGV para los cálculos.

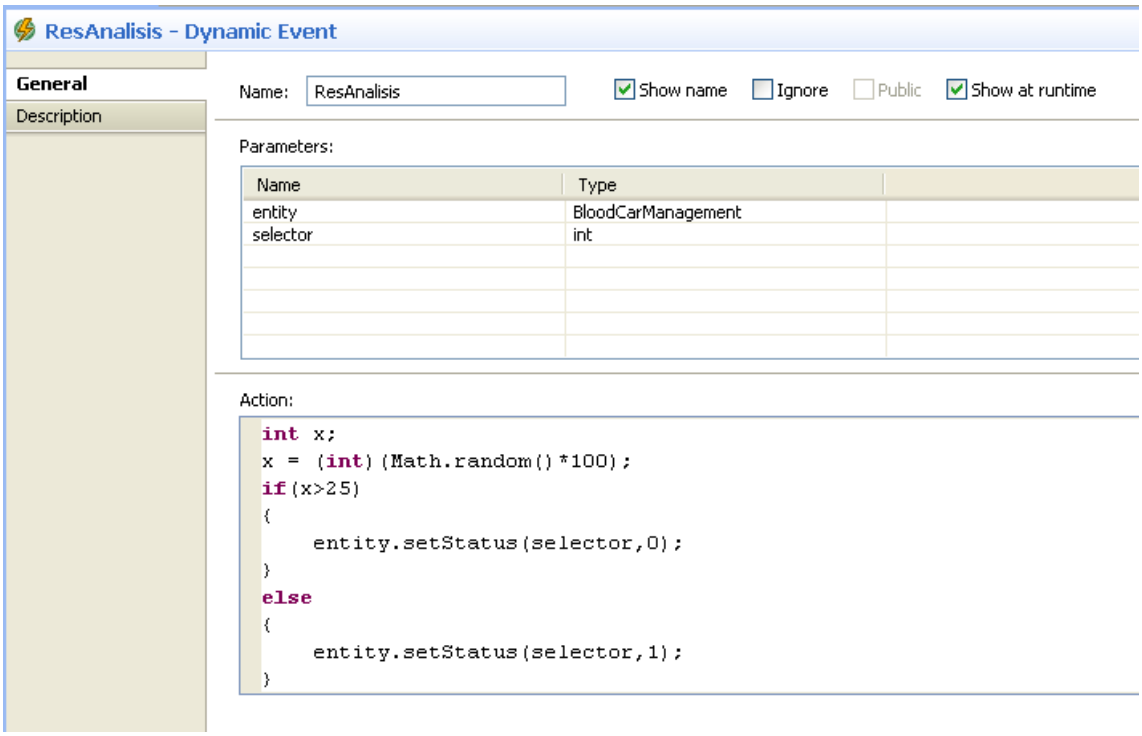


Captura de las propiedades de un Event

En la imagen superior podemos ver los ajustes de un Event. Este objeto permite ejecutar un código transcurrido un tiempo desde la activación de la simulación y que se repita cada cierto tiempo si así se lo configuramos.

En esta pestaña podemos ver

- Trigger type: Nos indica el modo en el que se ejecuta el evento. Timeout (el que se muestra en la imagen) se ejecuta una vez transcurrido un tiempo. También tiene la opción de Rate y condition.
- Mode: Esta opción aparece si el trigger type es “Timeout”. Nos muestra el modo en que se ejecuta la acción. En la imagen se ve la opción Cyclic, lo que permite ejecutarlo cada cierto tiempo después de la primera ejecución. También está la opción user control y occurs once que dejan, por este orden, ejecutarlo a petición del usuario o que solo se ejecute una vez.
- First ocurrence time (absolute): La primera opción que muestra son los segundos que transcurren desde que se inicia la ejecución hasta que se ejecuta por primera vez la acción. La segunda opción es para fijar una fecha de ejecución.
- Recurrence time: Esta opción solo aparece si tenemos seleccionado Mode con Cyclic. Nos indica el tiempo que transcurre entre ejecuciones.
- Action: Es el código que se ejecuta. En este caso ajusta los valores para saber cuántos vehículos han pasado por cada máquina de estados para poder realizar la gráfica. No realiza trabajo útil para el fin principal del proyecto de gestionar el tráfico.



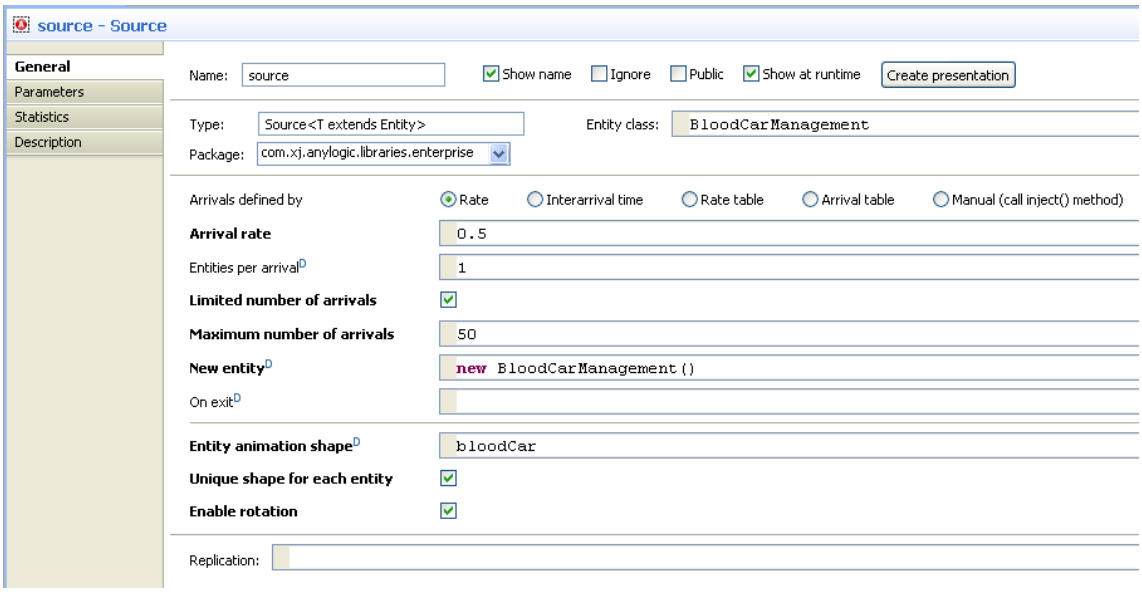
Captura de las propiedades de un Dynamic Event

Un Dynamic Event ocurre transcurrido un cierto tiempo tras ser invocado. En esta captura podemos ver la configuración. Es muy similar a un Function solo que los Function Arguments ahora son Parameters y no permite el retorno de un valor. También hay una importante diferencia, cuando se crea un Dynamic Event el primer parámetro que se le pasa es el tiempo que transcurrirá antes de ser ejecutado y detrás van los elementos que aparecen en la lista “Parameters”. En este caso simula el haber finalizado el análisis y haber obtenido una respuesta. Esta tiene un 25% de posibilidades de haber salido mal y tenerse que volver a realizar.

Para crear un Dynamic Event se debe crear y guardar en una variable como en el siguiente código.

```
"DynamicEvent e = create_ResAnalisis(60,entity,1);
entity.setEvent(e)"
```

La variable "e" es del tipo DynamicEvent con lo que podemos guardar dicho elemento en ella. La llamada a la función create_ResAnalisis(60, entity, 1) es la que nos crea un Dynamic Event del tipo ResAnalisis (el nombre que le asignamos al Dynamic Event). 60 es el tiempo que tardará en empezar la ejecución tras su creación. Los parámetros entity y 1 son los que pide el propio evento para su creación. El código "entity.setEvent(e) es propio de los objetos BloodCarManagement y almacena los Dynamic Events ya que como "e" es una variable que se pierde tras la ejecución del código debemos guardarlo en una variable que no se pierda y los objetos de la clase BloodCarManagement tienen un ArrayList para almacenarlos.



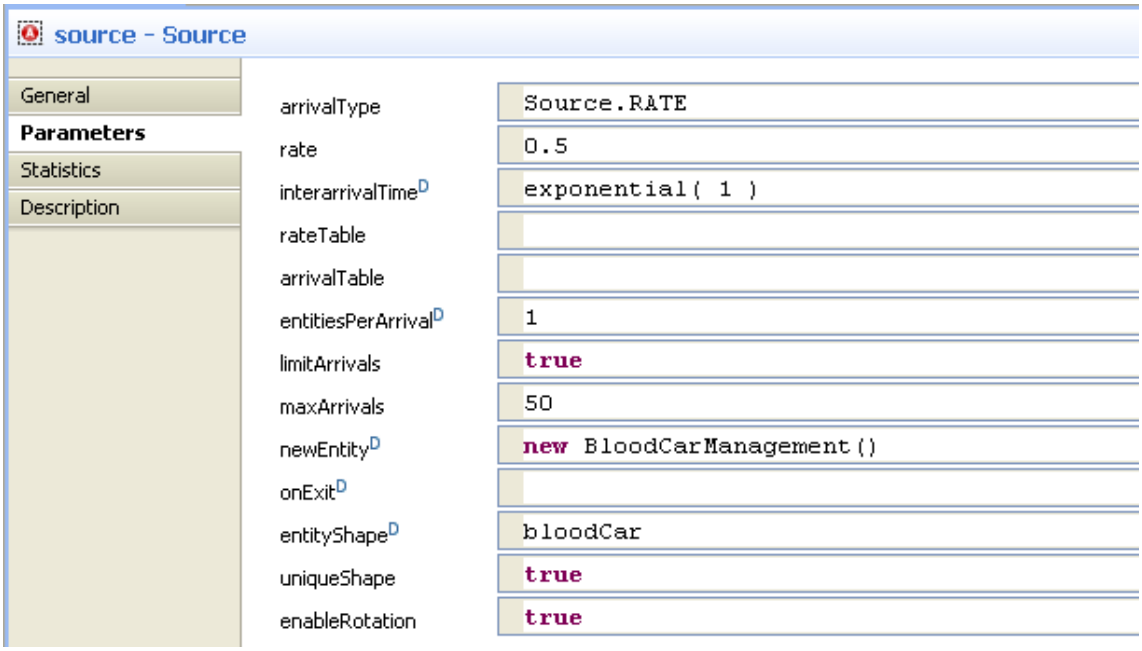
Captura de las propiedades de un Source

En la captura superior podemos ver la configuración de un objeto Source.

En esta pestaña podemos ver:

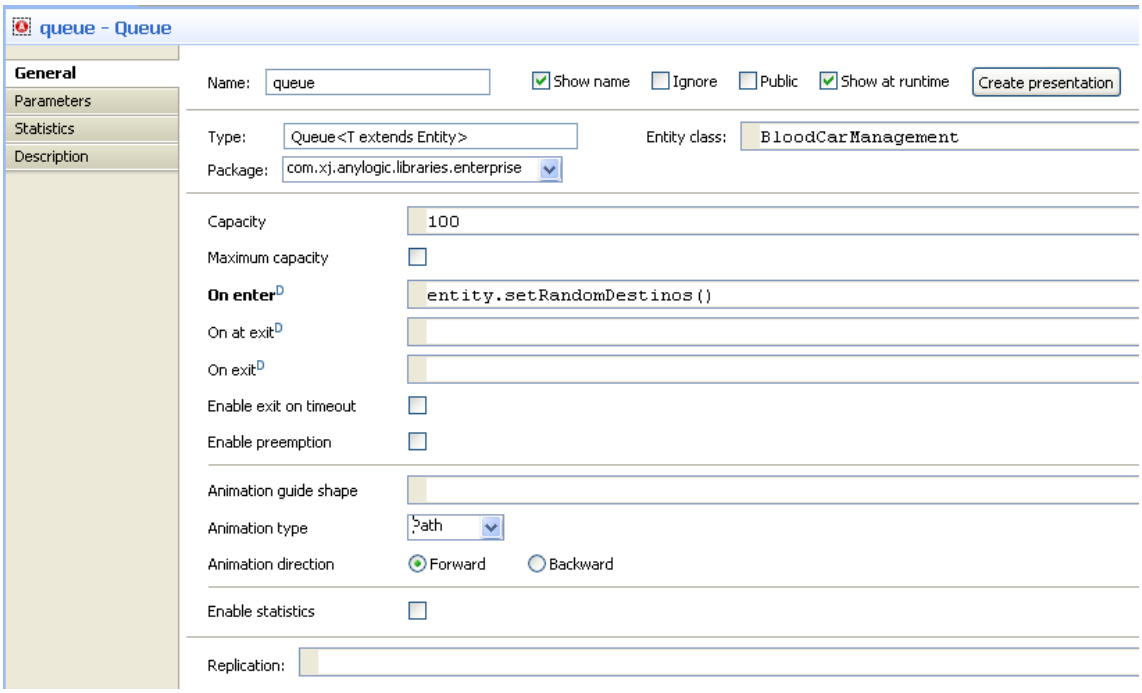
- Type y Package: Se dejan por defecto.
- Entity class: Por defecto viene Entity. Como en este proyecto se requieren funciones adicionales de las que proporciona la clase Entity hemos creado BloodCarManagement que hereda de Entity y nos proporciona dicha funcionalidad adicional.
- Arrival defined by: Diferentes formas de organizar las llegadas de elementos. En este caso tenemos seleccionada Rate que nos define una media de llegadas en torno al valor que le damos. Tanto Rate como Interarrival time (si se usa el valor que trae por defecto) es aleatória aunque se puede fijar. Las que se basan en tablas buscan en ellas los valores para saber los momentos en que se debe introducir un objeto y la opción Manual nos permite llamarlas de forma manual.

- Arrival rate: Aparece si tenemos Rate seleccionado. Con el valor 0.5 le indicamos que cada 2 segundos entre un vehículo (un BloodCarManagement).
- Entities per arrival: Nos permite definir el número de elementos que llegan cada vez.
- Limited number of arrivals: Si está chequeado limita a un número el total de elementos que se crean.
- Maximum number of arrivals: Aparece si la opción anterior esta activada. Indica el total de elementos que se crean.
- New enitity: Función que se debe ejecutar para crear un nuevo elemento.
- On exit: Código que se ejecuta al salir un elemento del objeto Source.
- Entity animation shape: Imagen que se usa para visualizar el objeto. En nuestro caso es un group que contiene rectángulos que forman la imagen de los vehículos que se ven en la animación.
- Unique shape for each entity: Con esto seleccionamos si se debe duplicar la imagen o se puede usar la misma para todos. Como no se modifica el color de los objetos se podría dejar sin seleccionar, pero por defecto viene activada y tampoco supone un gran consumo de recursos.
- Enable rotation: Activado hace que los objetos giren si la dirección de la línea que siguen varía. Solo tiene efectos estéticos.
- Replication: Esta vacío por defecto.



Captura de los parámetros de Source

En la pestaña Parameters podemos ver las variables que hemos configurado anteriormente. Es la misma configuración que en la anterior pero nos indica el nombre de las variables por si durante la ejecución queremos cambiarlas.

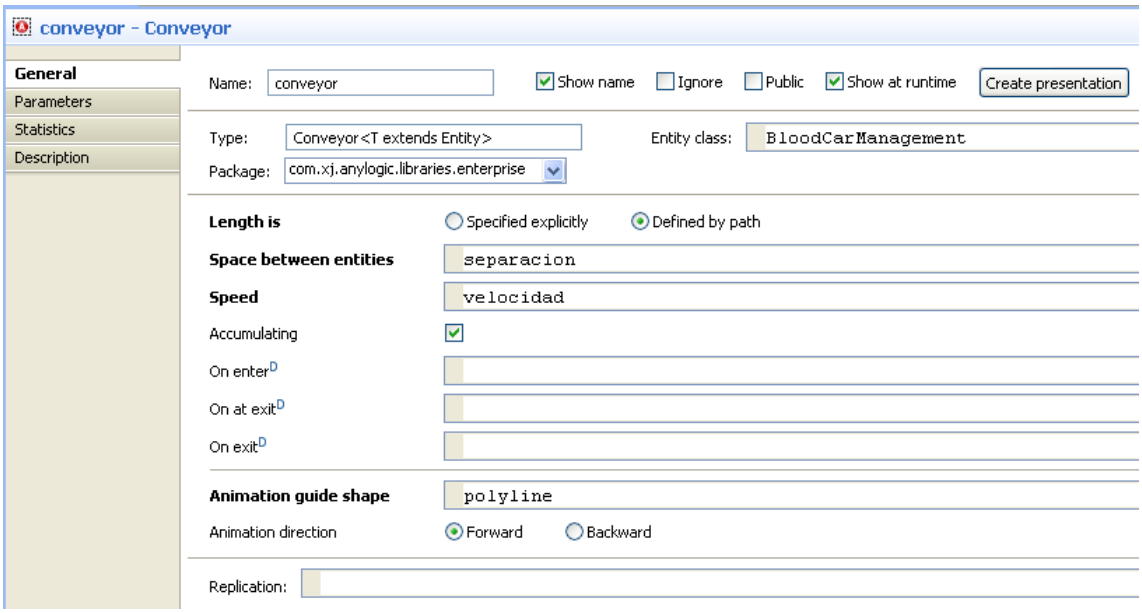


Captura de las propiedades de un Queue

El objeto Queue tiene al igual que Source, unas configuraciones que se dejan por defecto como Type o Package. Entity class se usa la misma durante todo el proyecto.

En esta pestaña podemos ver:

- Capacity: Indica la capacidad máxima de este almacen. En este caso hemos puesto 100.
- Maximum capacity: Al seleccionar esto cambia el tamaño máximo al valor más alto que pueden alcanzar las variables de tipo entero (Integer). No da el valor exacto debido a que se puede exportar a diferentes plataformas y estas pueden tener un número distinto como valor máximo de los enteros.
- On enter: Código que se ejecuta cuando un elemento entra en este objeto. En este caso asigna de forma aleatoria los destinos que debe visitar el AGV para realizar los análisis.
- On at exit: Código que se ejecuta cuando un elemento está listo para salir.
- On exit: Código que se ejecuta una vez ha salido el elemento de este objeto.
- Enable exit on timeout: Se puede fijar un tiempo máximo de permanencia en estos almacenes. Si se activa esta casilla se tiene que especificar el dicho tiempo y transcurrido ese tiempo si no ha salido por el puerto normal sale por el puerto de timeout.



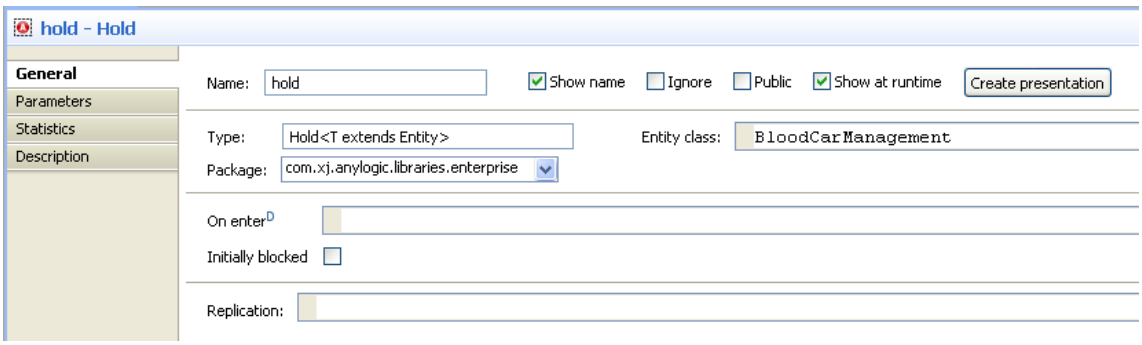
Captura de las propiedades de un Conveyor

En esta imagen vemos que el animation guide shape no está en blanco como en el caso anterior. Es por eso que se explicara en este.

En esta pestaña podemos ver

- Length is: nos permite definir si queremos que la longitud sea especificada por un valor o por un path (una línea dibujada). El primer caso sería seleccionando Specified explicitly mientras que si lo define el tamaño de una línea sería Defined by path.
- Space between entities: Es la separación que tienen los elementos entre sí. Aquí podemos encontrarnos un problema y es que este espacio es entre elementos dentro de un mismo conveyor pero se puede dar el caso de ver una colisión en la animación ya que entre distintos conveyors no guarda dicha distancia de seguridad. Por lo tanto si un vehículo está al final de un conveyor y otro al principio del siguiente sin poder avanzar, la animación mostraría las imágenes de los dos vehículos superpuestas. Esto en la realidad no sucedería ya que los AGV detectan siempre la distancia con el objeto que tienen en frente y no hacen distinción entre conveyors.
- Speed: Define la velocidad que recorre el AGV cada segundo. Tanto aquí como en el apartado anterior se usa una variable. La razón es que hay unos 50 conveyors en todo el proyecto y a la hora de cambiar la velocidad para hacer las estadísticas supondría un sobreesfuerzo.
- Accumulating: Al marcar esta opción permites que si un AGV no puede avanzar por alguna razón (como haber llegado al final de un conveyor y encontrarse un semáforo en rojo que le impide el paso) los que vienen detrás siguen avanzando hasta llegar a la distancia de seguridad entre AGV. Sin esta opción validada se comportarían como si de una cinta de transporte se tratase deteniendo todos los elementos de la cinta si uno no puede seguir avanzando.
- Animation guide shape: Objeto del tipo polyline que sirve de guía para avanzar al AGV.
- Animation direction: Indica la dirección de avance del AGV. Es forward por defecto pero si la línea se dibuja al revés (si bien dibujada fuese de izquierda a derecha, al

revés sería de derecha a izquierda) y se activa la opción backward no notaríamos diferencia en la animación.

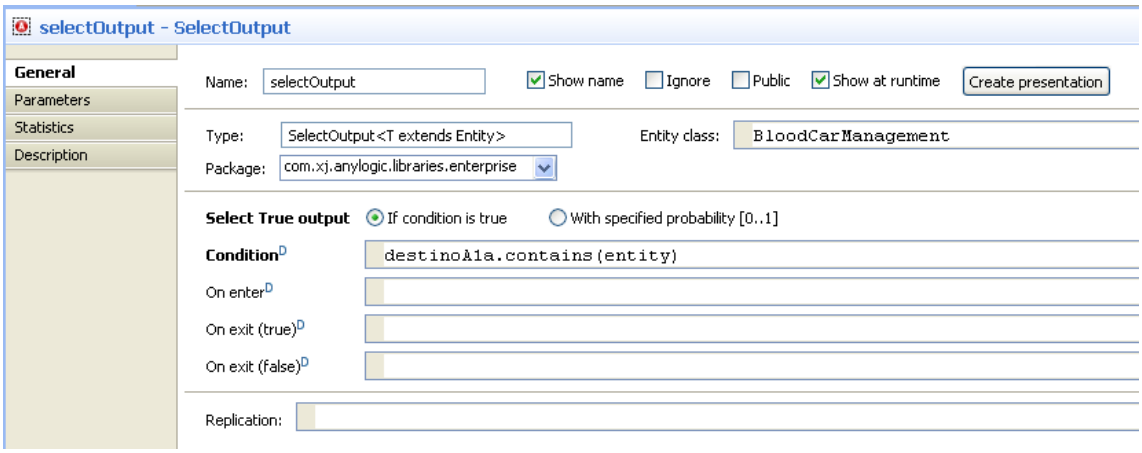


Captura de las propiedades del Hold

Los parámetros de comportamiento son similares a los anteriores. No ofrece opciones de "At on exit" ni "on exit". Tampoco son necesarios ya que los objetos pasan o no por el hold, así que estas dos opciones y "on enter" se ejecutarían de forma consecutiva.

En esta pestaña podemos ver

- Initially blocked: Muestra si se encuentra bloqueado desde el principio o no. Como en los casos anteriores esta opción se puede modificar desde código modificando el parámetro que lo controla.



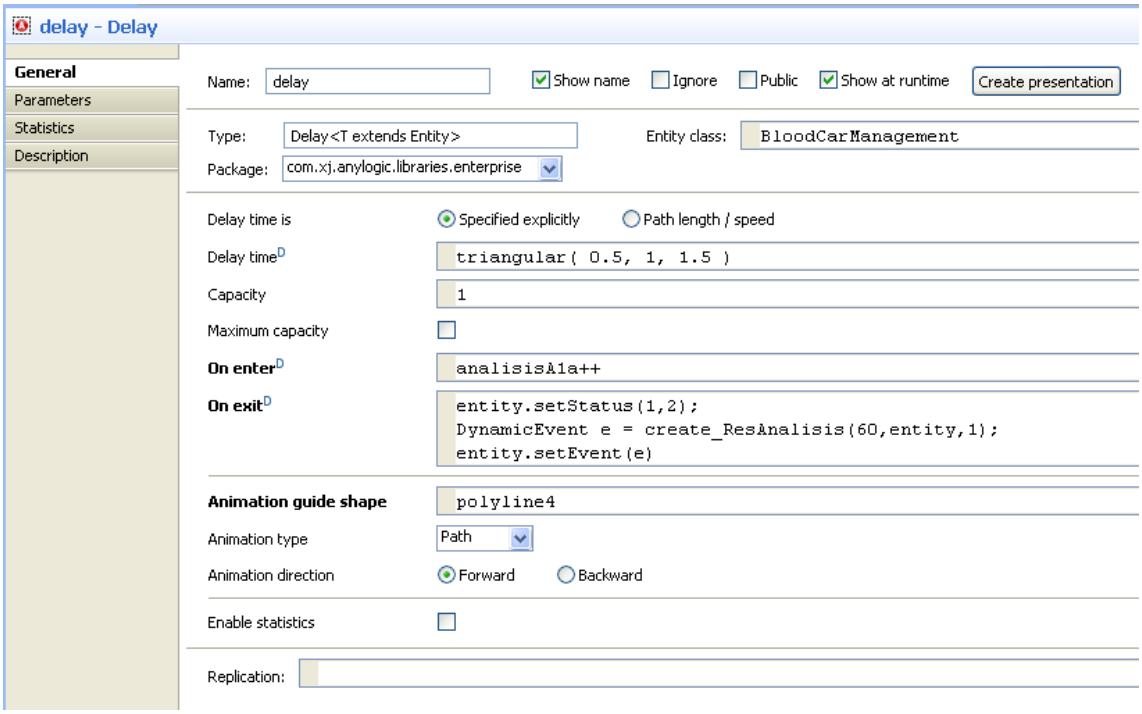
Captura de las propiedades del SelectOutput

En los objetos de tipo SelectOutput hay que tener especial cuidado de que siempre tenga sitio donde ir los objetos que salen.

En esta pestaña vemos:

- Select True output: permite elegir entre que una condición sea cierta o otorga true con un porcentaje de probabilidad seleccionando "If condition is true" o "with specified probability[0..1]" respectivamente.

- Condition: nos pide la condición para saber qué camino debe tomar. La función que se ve retorna cierto o falso en el caso de que el entity que tenemos este o no en la Collection Variable destinoA1a.
- On exit(true/false): a diferencia de los otros elementos que solo disponían de una salida, este tiene dos opciones de salida y nos permite realizar acciones distintas si el AGV sale por una salida u otra.



Captura de las propiedades de un Delay

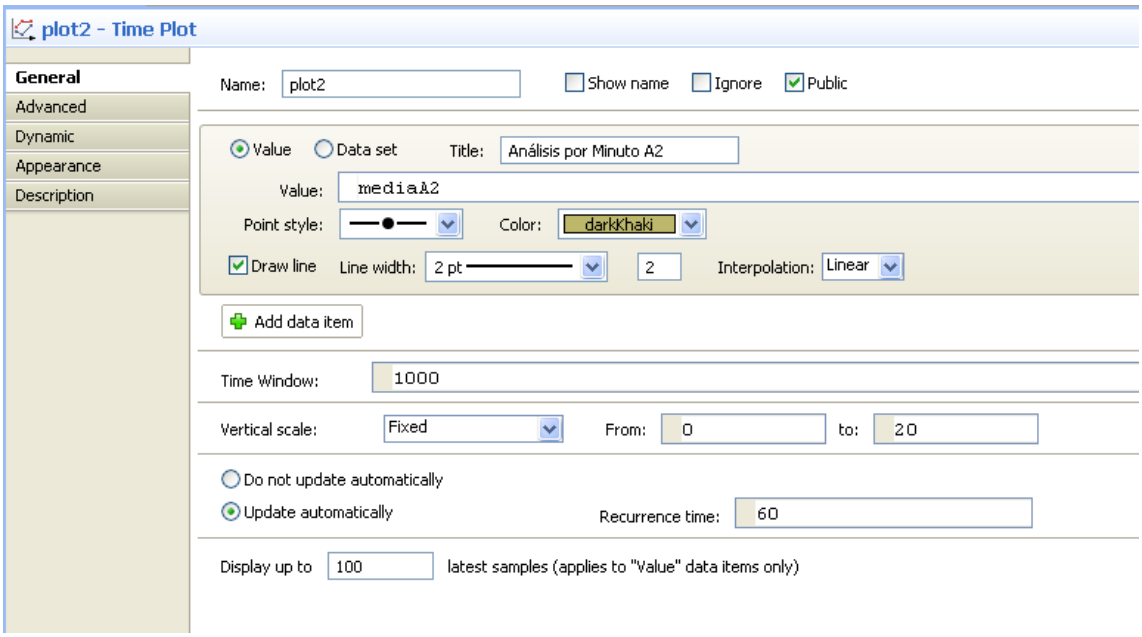
Delay, al igual que SelectOutput nos da un error si cuando un objeto debe salir no tiene espacio para hacerlo.

En esta pestaña vemos

- Delay time is: Nos permite elegir entre Specified explicitly o Path length/speed. O le especificamos el tiempo directamente o el tiempo será definido por la longitud de la pista dividida por la velocidad a la que avanza. En este caso la elección es un tiempo especificado, aunque como veremos a continuación no significa que sea un tiempo fijado.
- Delay time: Aquí le decimos el tiempo en segundos que debe tardar. Como esta tarea consiste en acercarse a la máquina de análisis, ajustarse para quedarse justo debajo de donde están las agujas que toman la muestra de sangre y volver a la pista, no se pone un tiempo determinado para realizar la operación. En su lugar se ha puesto una función aleatoria que devuelve un número entre 0.5 y 1.5 con un reparto triangular donde la gran mayoría de los AGV tardaran 1 segundo o muy aproximado y solo unos pocos tardaran 0.5 o 1.5 segundos o aproximado.
- Capacity: Capacidad del delay. En este caso se ha puesto que solo puede entrar 1 vehículo a la máquina de análisis pero en el que hay situado al final, en la salída del

laboratorio tiene capacidad para 50 ya que simula el tiempo en el que los AGV dejan la carga, cogen una nueva y están listos para volver a entrar.

- Enable statistics: No se usa en el proyecto.

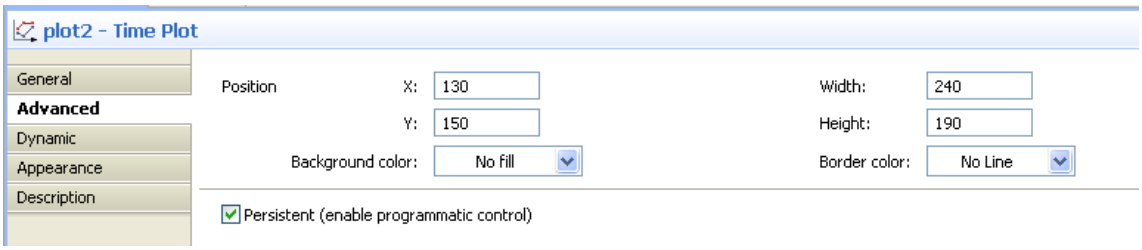


Captura de las propiedades del Plot

Plot nos permite realizar un seguimiento de las estadísticas en tiempo de simulación.

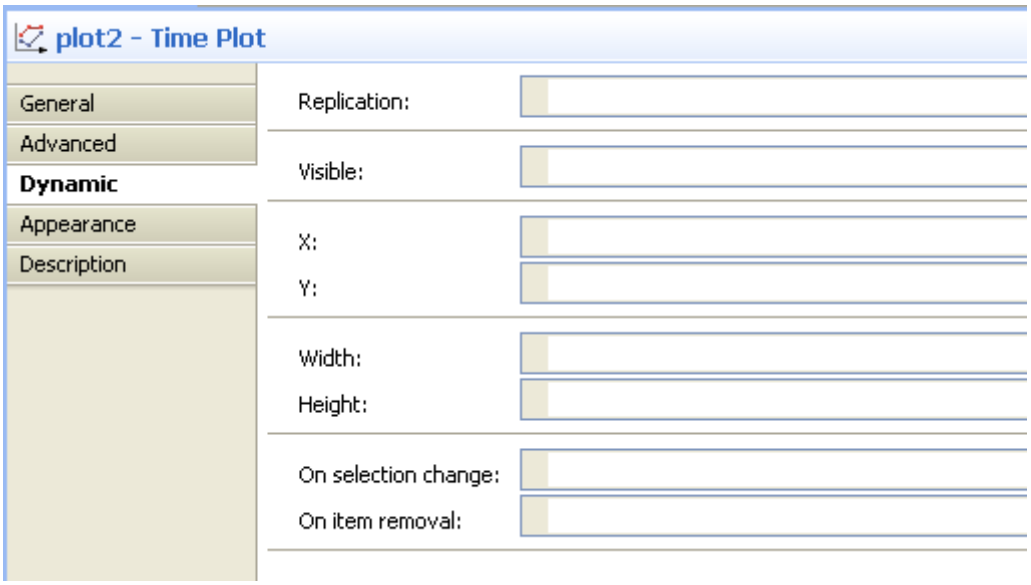
En esta pestaña podemos ver:

- Value: Indica que mostrara un valor y más abajo que valor es.
- Point style: Estilo del punto para cada vez que se hace un cambio o lectura.
- Color: el color de la línea
- Draw line: Si está seleccionado muestra una línea que une los puntos
- Add data ítem: Si queremos podemos mostrar más de una variable en la gráfica. Al darle a este botón nos agrega una ventana como la sombreada que hay justo encima del botón.
- Time Window: Valor del eje de las X. Al llegar a este valor agrega un 50% más compactando la gráfica. Si se supera este 50% se eliminan los 500 primeros segundos y añade otros al final.
- Vertical scale: nos permite tenerla fija o automática. Al quererla comparar con las otras es mejor hacerla fija para que a simple vista se vea con más facilidad que valores son más altos.
- From/to: Al determinar fija la configuración anterior nos pide unos valores de mínimo y máximo.
- Do not update automatically/Update automatically: Le indicamos si la actualización se realiza de forma automática o si por el contrario se le llamara desde una función para que se actualice. En este caso hemos elegido de forma automática y con un refresco de 60 segundos
- Display up to: Marca un límite de muestras que aparecen en la gráfica.



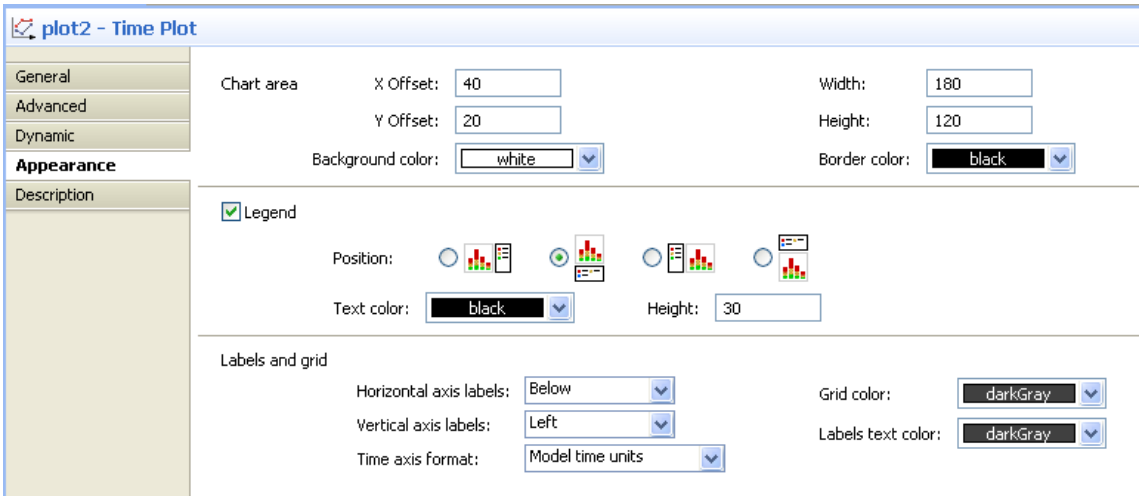
Captura del advanced del plot

Nos permite ajustar ciertas configuraciones, pero por lo general, exceptuando los colores, se hace de forma gráfica en la ventana de main.



Captura del Dynamic del plot

Variables que en este proyecto no se llegan a usar.



Captura del Appearance del plot

En esta imagen podemos ver entre otras opciones una que nos permite seleccionar la apariencia del plot. En este proyecto se han dejado por defecto con los datos debajo de la gráfica. Solo tiene efectos visuales. El resto de las opciones

Uno de los mayores problemas inconvenientes que encontramos al programar en Anylogic es la reutilización de código en el proyecto. En un entorno de programación habitual se pueden realizar funciones con parámetros que sirvan para todos los casos similares, como interesaría que ocurriera al programar los semáforos que realizan la misma función en todo el proyecto. Pero en Anylogic tenemos los objetos y los campos de configuración que hay que cambiar a mano. Se pueden copiar áreas de objetos (como los conveyors, delays, holds y sus conexiones) y pegarlos como objetos nuevos. En ese caso nos crea objetos nuevos y nos mantiene la configuración interna y las conexiones de los puertos. Pero las asociaciones de, por ejemplo, conveyors y polylines se deben modificar y al ser un gran número de campos que se deben cambiar es normal que al realizar muchas copias siempre aparezca un fallo del tipo un AGV que desaparece y aparece en otra zona del mapa o que la gestión del tráfico se realice con el estado del tráfico de otra intersección. Estos fallos son, sin duda, fallos del programador, que aunque se siga una metodología paso a paso para realizar los cambios de las configuraciones de cada objeto para adaptarlo a las nuevas, siempre se olvidan algunas o se comete algún error de escritura. En definitiva, demasiado trabajo manual para reutilizar código.

Por otra parte, la velocidad de los AGV se marcaba fija en las propiedades del conveyor en la primera versión que se uso para el aprendizaje. Al ver que para realizar la estadística esto supondría un aumento del trabajo de cambiar las velocidades y también supondría también una fuente de posibles errores si se olvidaba cambiar algún valor o se ponía uno erróneo. De cara a la versión completa del laboratorio de análisis se uso una variable estática para cambiar todos los valores a la vez y reduciendo la posibilidad de fallo. También estaba la posibilidad de usar un active object que no se usa en este proyecto. Un active object permite realizar cambios en los valores, en este caso la velocidad, mientras se ejecuta la simulación de forma manual.

CAPÍTULO 5. Resultados

5.1 Casos de prueba

Este sistema trabaja con una planta ya definida pero hay diversos parámetros que hay que ajustar para maximizar su rendimiento. En este caso el rendimiento se medirá por número de muestras procesadas por unidad de tiempo.

Los parámetros que pueden controlarse son el del número de AGV presentes en la planta, su velocidad media y la probabilidad de éxito en las primeras determinaciones en cada máquina.

Se han establecido diversos conjuntos de valores de dichos parámetros para analizar cómo varía el rendimiento en función de los mismos.

A continuación se presenta primero unos ejemplos simples de cómo se han obtenido los diversos conjuntos de valores. El primero es para una planta con vehículos con una misma velocidad y para una distribución de probabilidades determinada y variando el número de vehículos. El segundo, que se verá en el apartado 5.3, mantiene fijos todos los parámetros excepto el de la velocidad. Finalmente, para obtener unos datos relevantes, se muestran las gráficas obtenidas variando diversos parámetros al mismo tiempo.

5.2 Cantidad de AGV

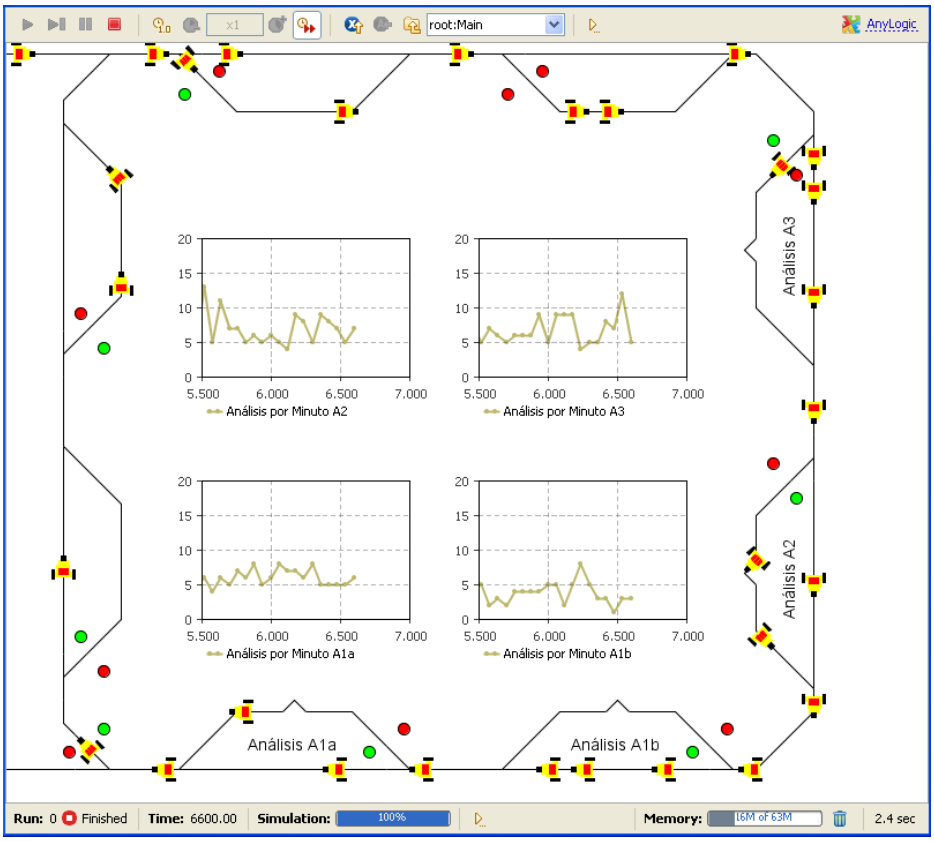
Es obvio que si hay pocos AGV no se podrá obtener un rendimiento elevado, pues muchas muestras quedarán en espera. Al contrario, si hay demasiados, las vías de circulación quedarán colapsadas y el rendimiento final decaerá.

En este apartado se tratará de este aspecto, con una planta con cuatro máquinas de análisis, situadas en la parte inferior y en la parte derecha de la imagen, y cuatro zonas de adelantamiento, situadas en la parte superior y en la parte izquierda, una entrada a la planta en la parte inferior izquierda por donde entraran los AGV a los que se les acaba de asignar una carga de muestras a analizar y una salida del laboratorio, donde los AGV que han finalizado su tarea vuelven a la zona de carga y descarga para una nueva tarea.

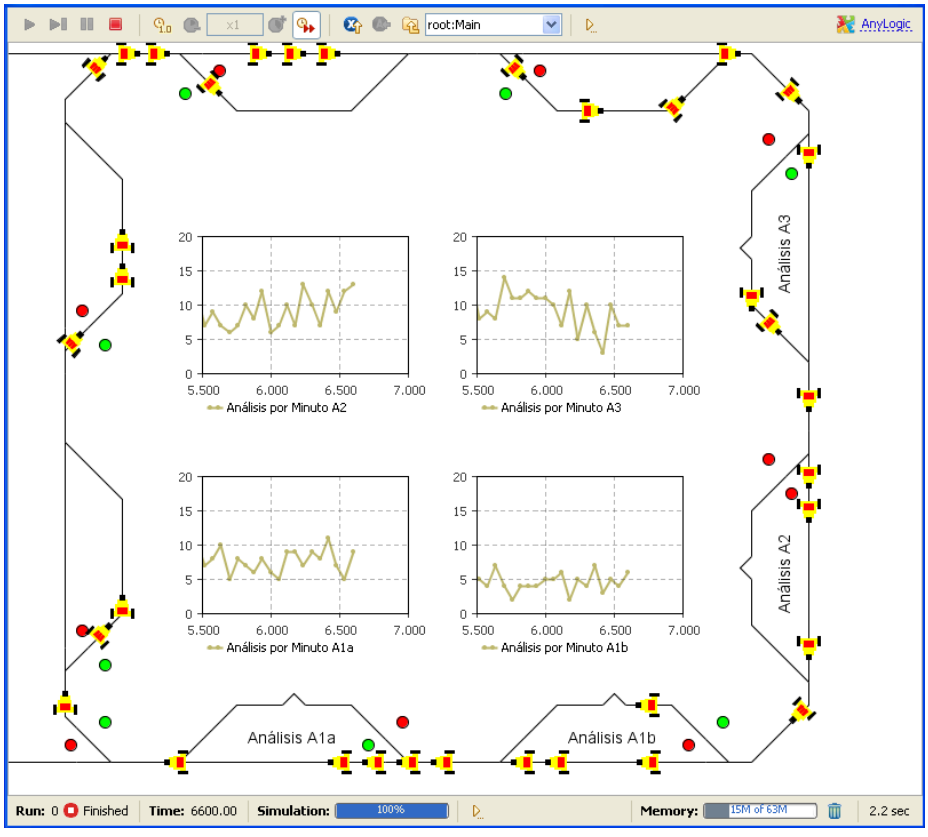
Se verá cómo varía el funcionamiento de la planta a medida que aumentamos el número de AGV disponibles para transportar las muestras de sangre hasta las máquinas de análisis. Esta variación se verá reflejada en las cuatro gráficas que hay en el centro de la planta y que muestran los análisis que se realizan cada minuto.

En las gráficas se aprecian picos y caídas pero tienden a estabilizarse en una franja de la gráfica. Esta franja será pues la medida del rendimiento que se ha obtenido al usar en la planta el número de AGV indicado en la simulación.

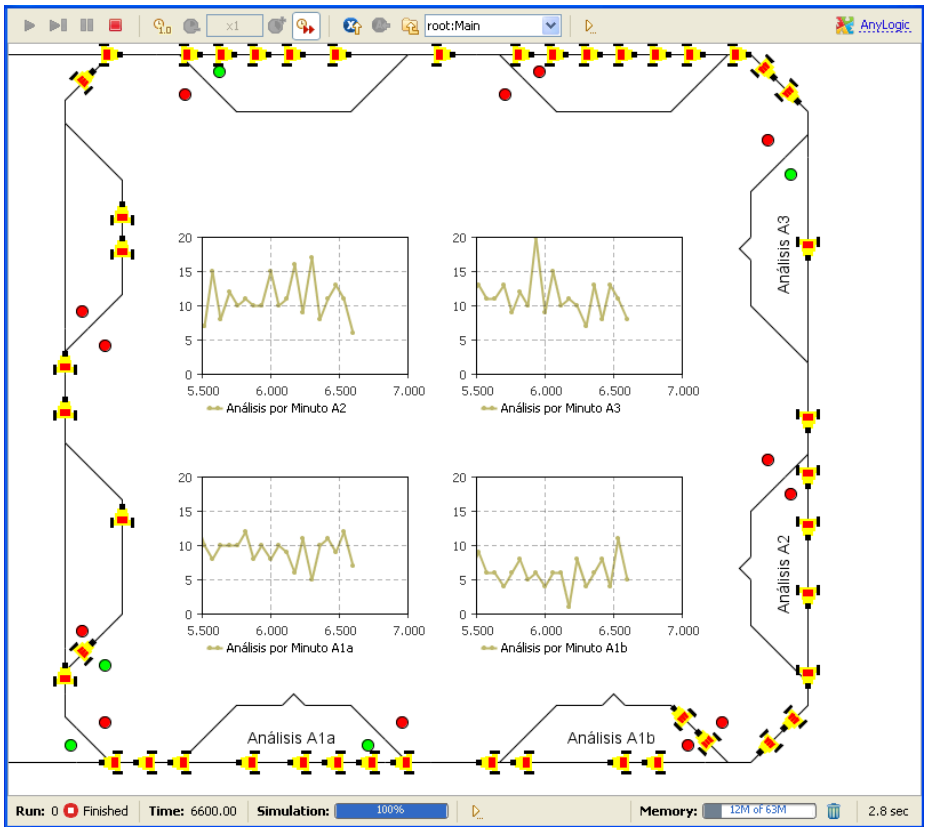
Es posible que en las capturas se aprecien menos vehículos AGV de los que se indican en la simulación. Esto es debido a que al terminar salen de la planta a la espera de nuevas instrucciones.



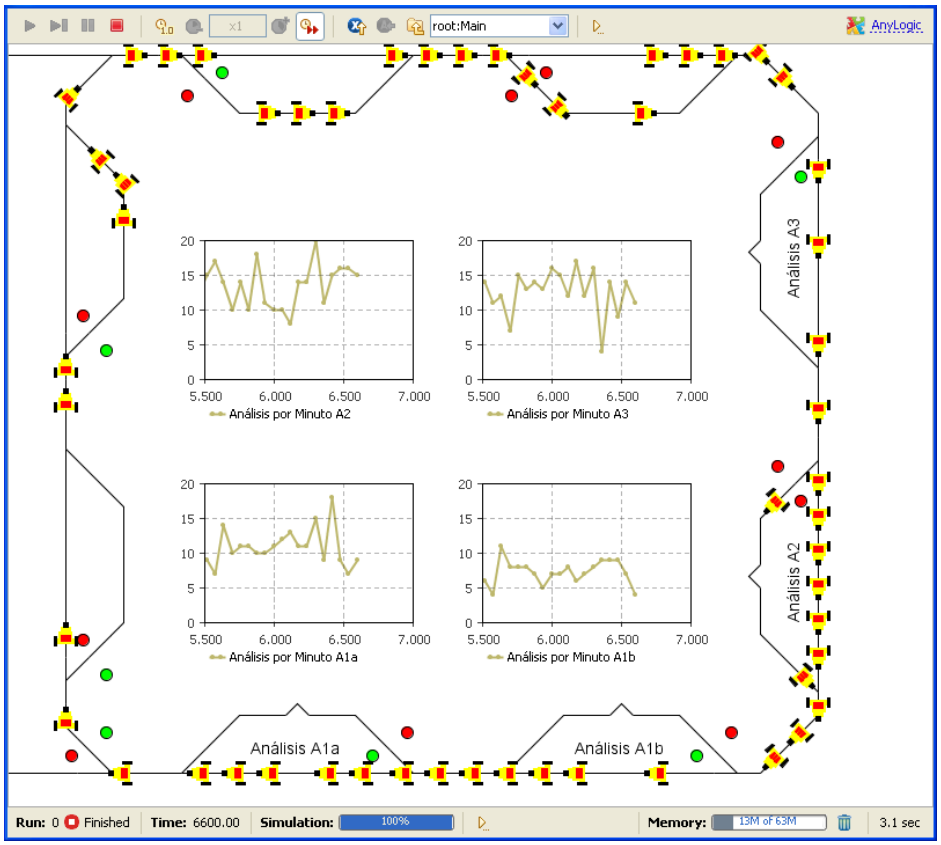
Simulación con 30 AGV



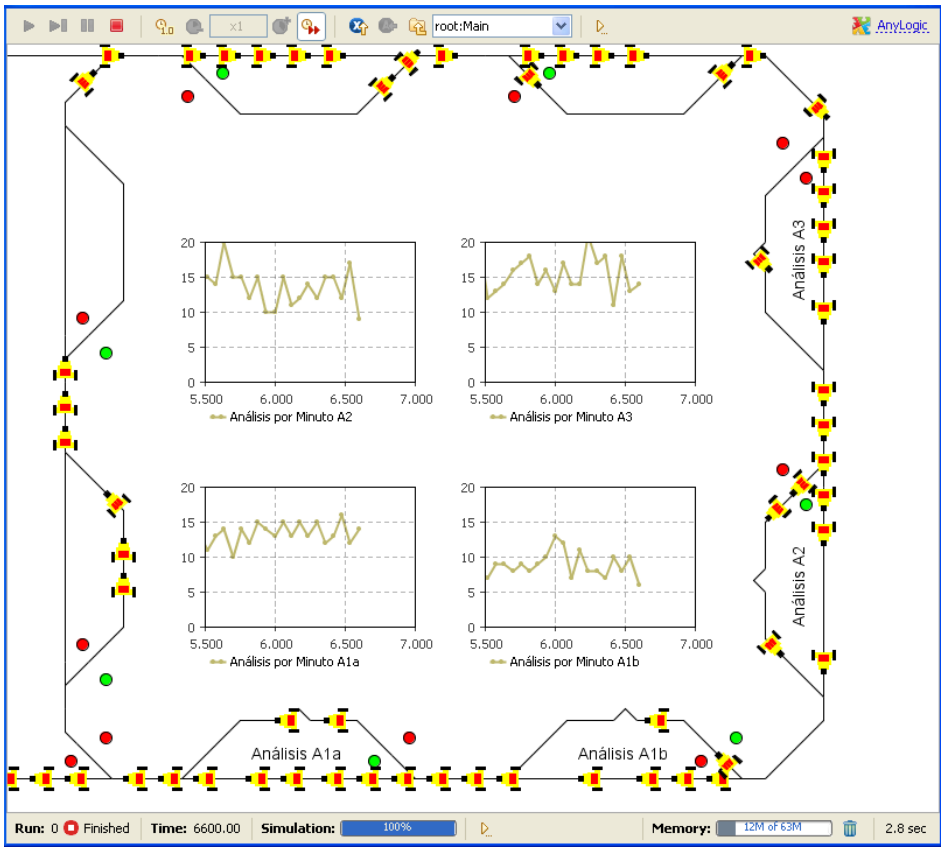
Simulación con 40 AGV



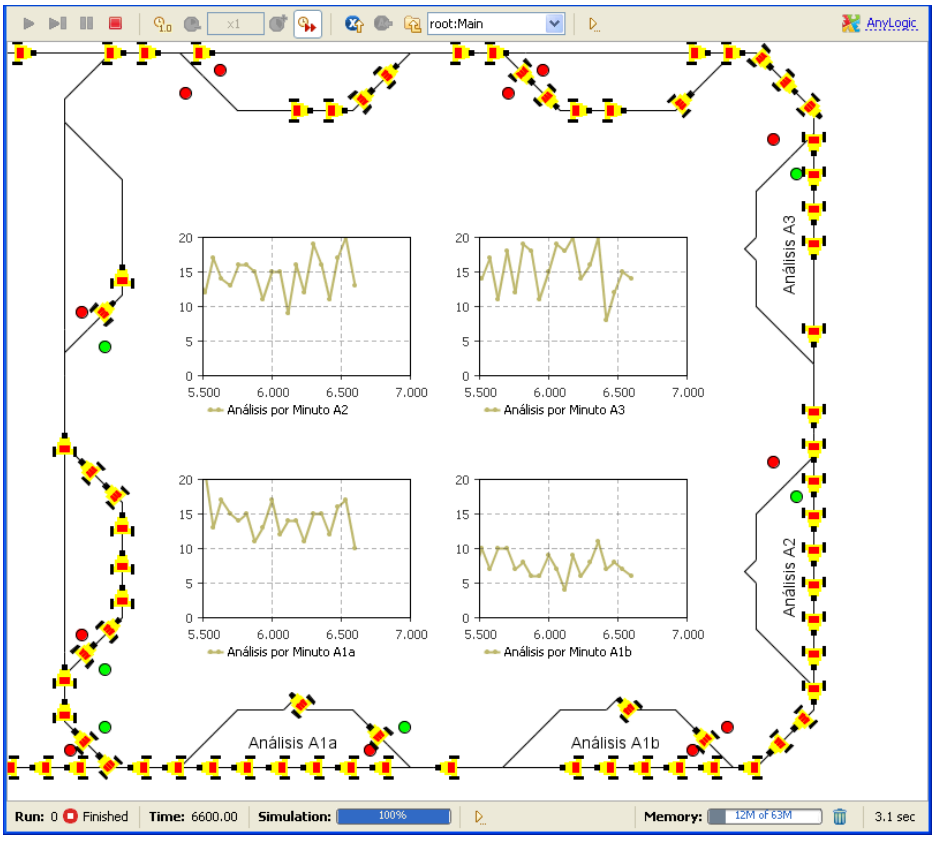
Simulación con 50 AGV



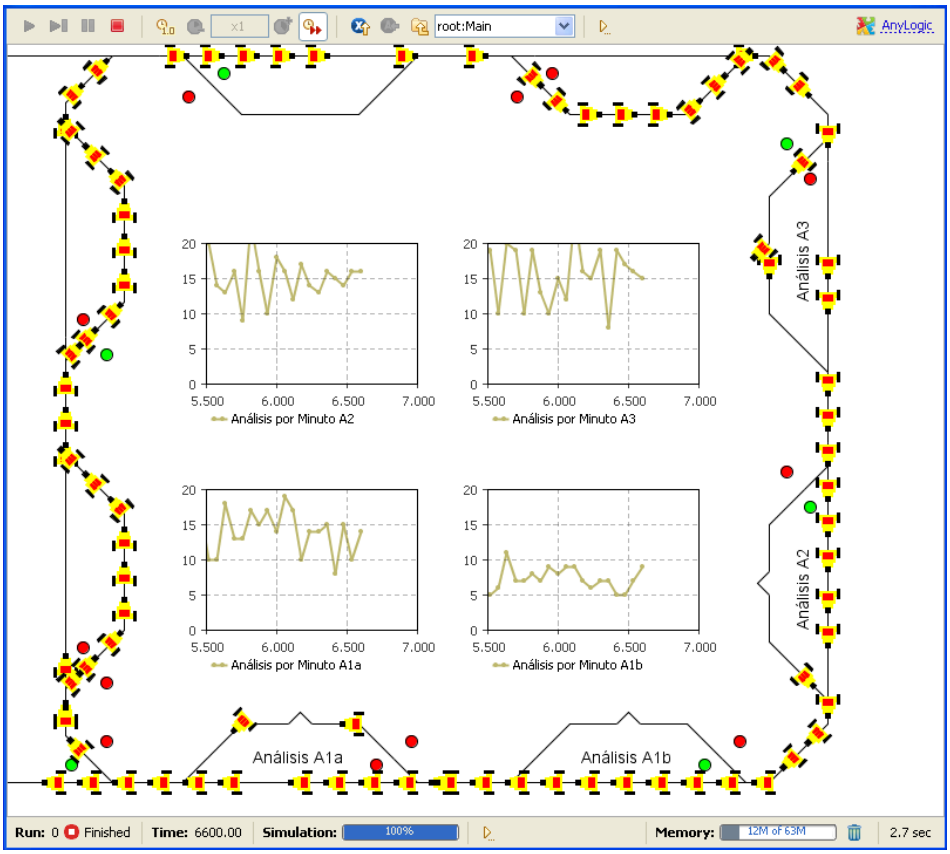
Simulación con 60 AGV



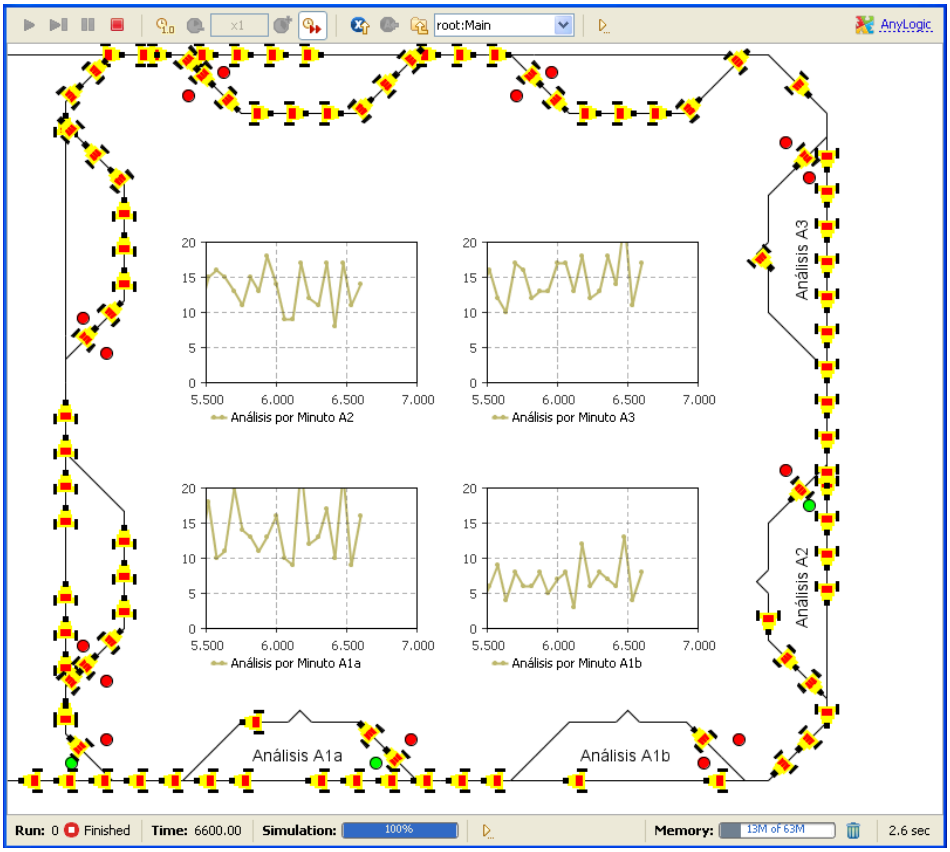
Simulación con 70 AGV



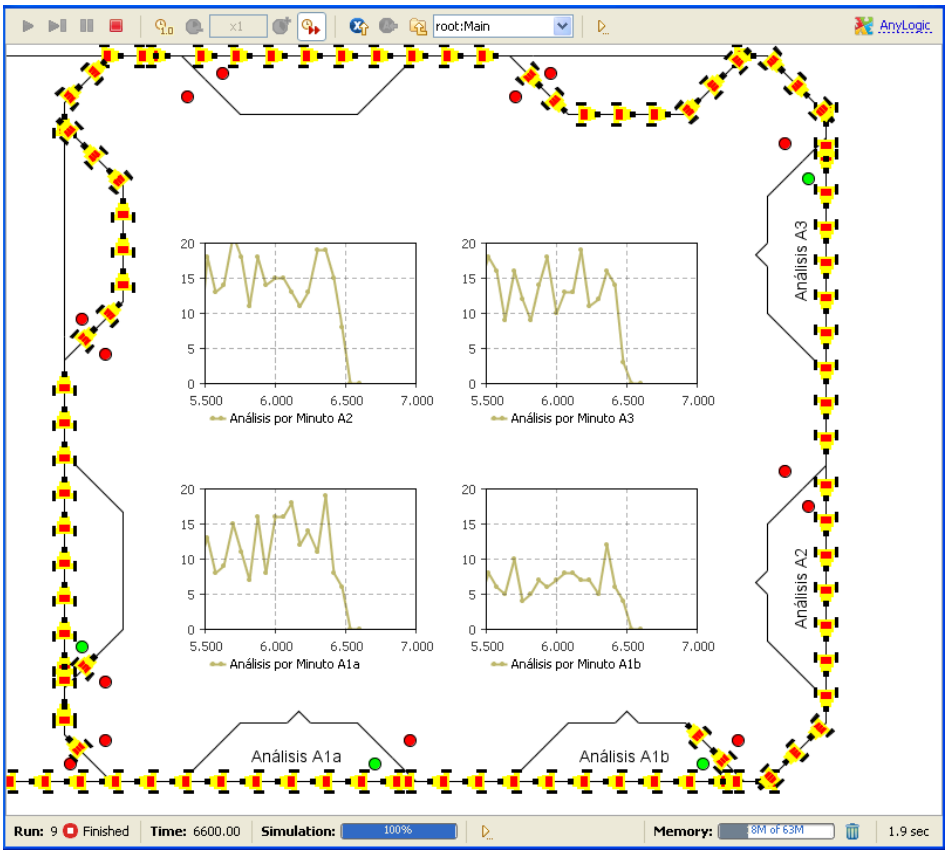
Simulación con 80 AGV



Simulación con 90 AGV



Simulación con 100 AGV

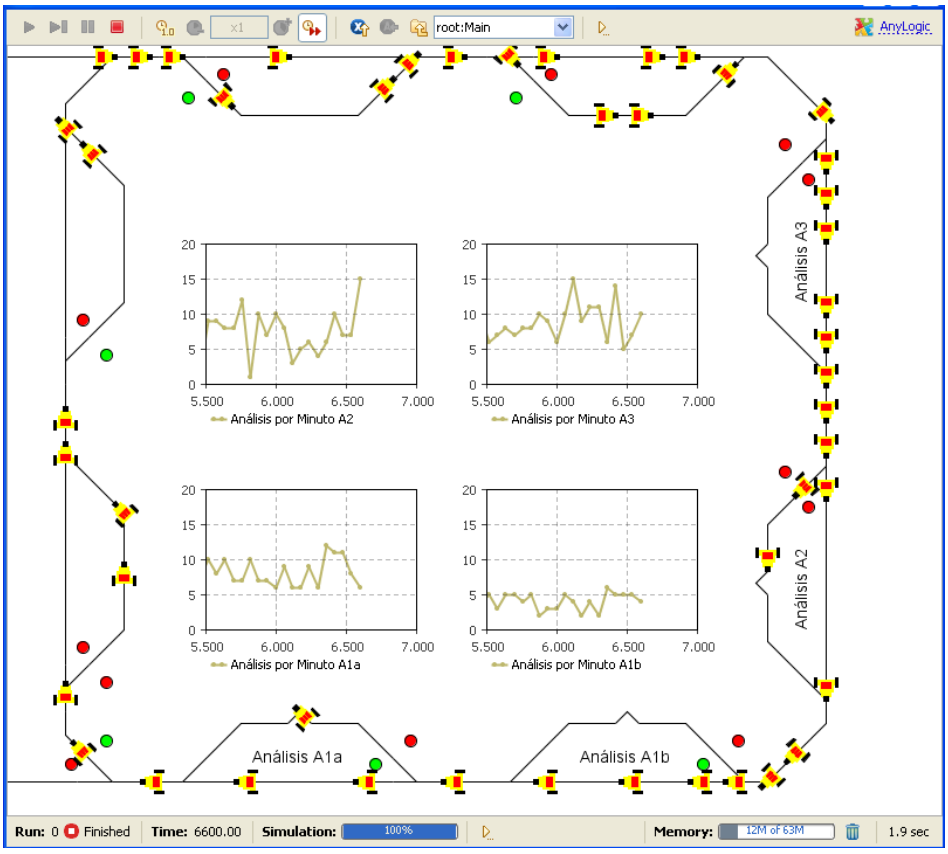


Simulación con 100 AGV colapsada

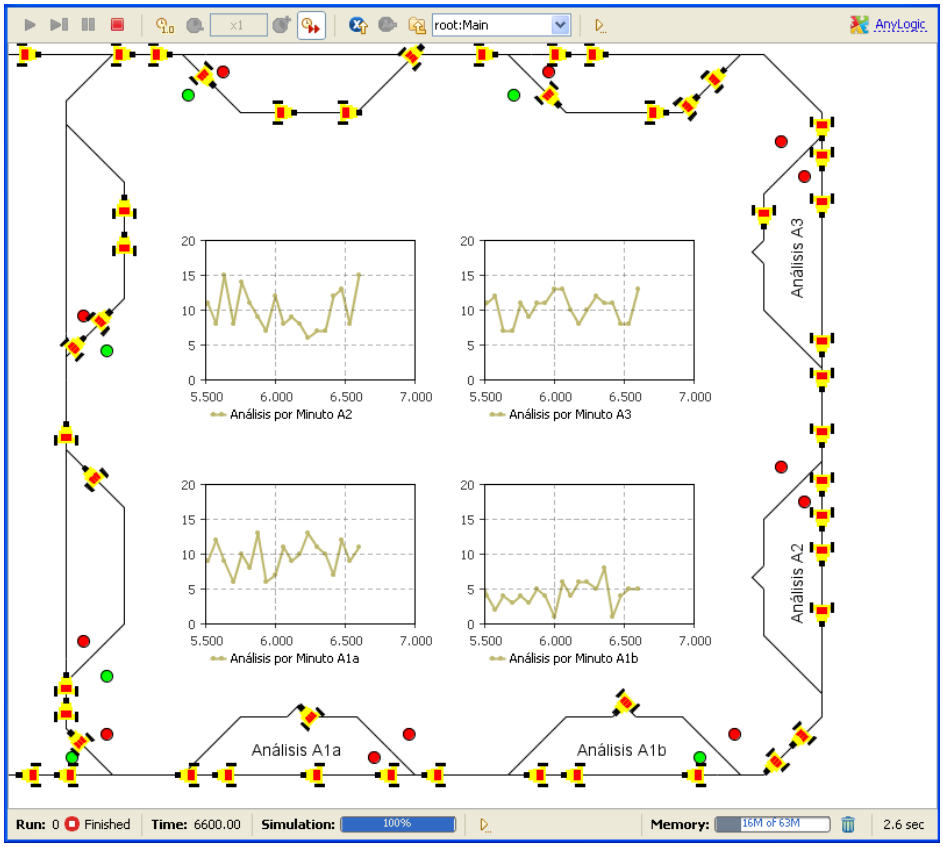
En estas gráficas vemos como el número de análisis por minuto que se realizan en cada máquina deja de crecer de forma tan clara a partir de los 60 AGV. Esto nos indica que por más que aumentemos el número de vehículos disponibles no conseguiremos un rendimiento mucho mayor ya que colapsamos las vías con vehículos que esperan los resultados de los análisis. De todas formas, hay que tener en cuenta que este valor, 60 AGV, es solo orientativo de cómo la curva de rendimiento llega a unos límites a medida que llegamos a la saturación de las vías. Como no se dispone de los datos reales de tamaño de un AGV capaz de transportar las muestras de sangre no podemos saber cuántos vehículos se requieren para colapsar las vías, pero el comportamiento de saturación sería similar al llegar a un cierto límite. Experimentando con 100 AGV en el laboratorio se llegan a dar casos de colapso total de modo que todos los AGV están parados y sin opción de avanzar como se ve en la última imagen de simulación.

5.3 Velocidad

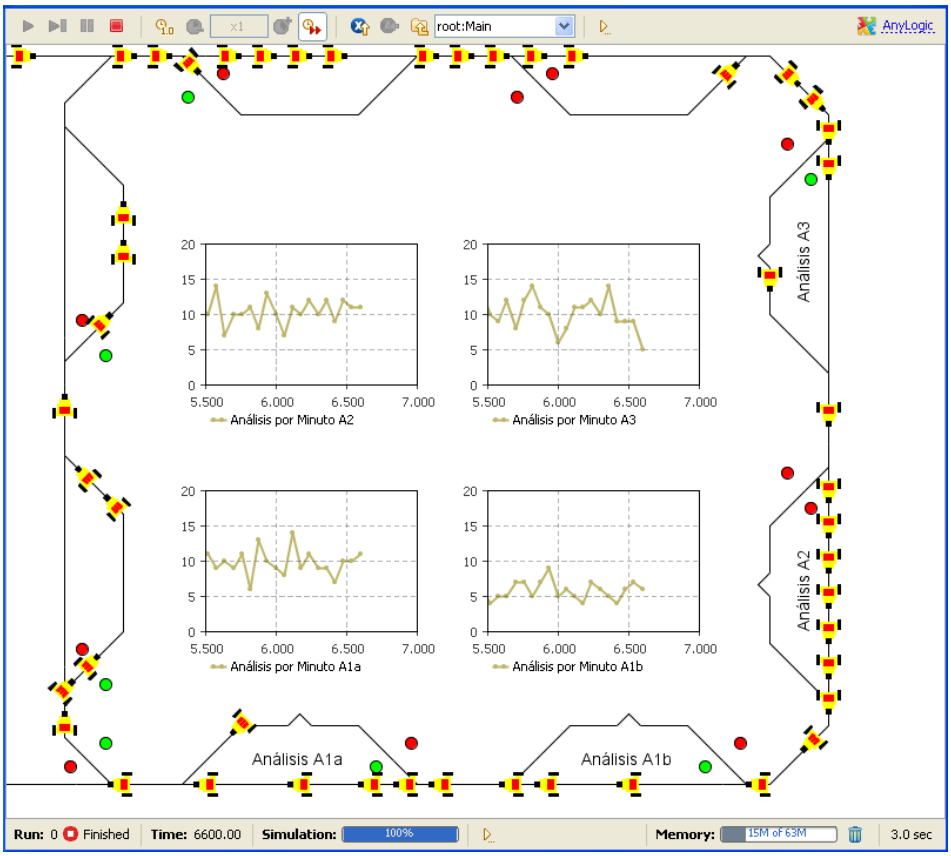
Para medir los cambios de rendimiento que genera aumentar o reducir la velocidad, testeamos velocidades que van desde los 30 a los 100 variando de 10 en 10. Las unidades de la velocidad no están determinadas. Para la simulación 10 es la longitud de un lado de los cuadrados que hay en la cuadricula donde se incorporan los componentes. Para hacer una aproximación, se podría decir que más o menos, la velocidad se está expresando en centímetros por segundo. El número de AGV que se simularán será 50, que como se puede ver en el apartado anterior, es un número que no llega al rendimiento máximo ni a un estado saturado. La planta es la misma.



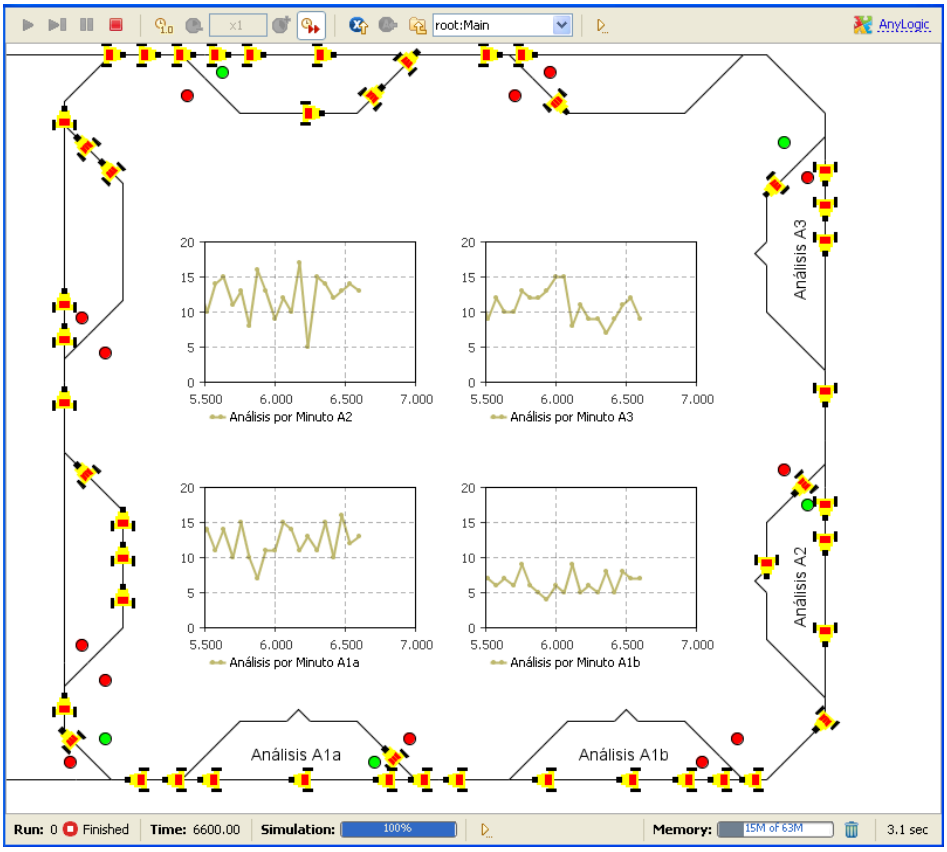
Simulación a velocidad 30



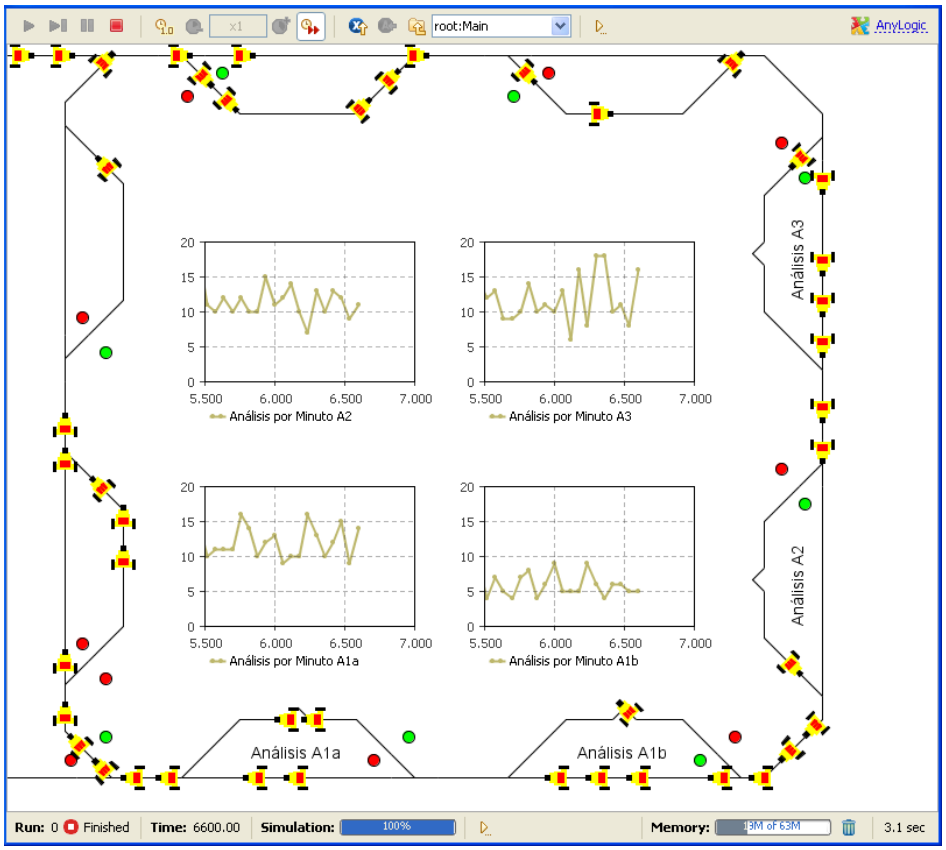
Simulación a velocidad 40



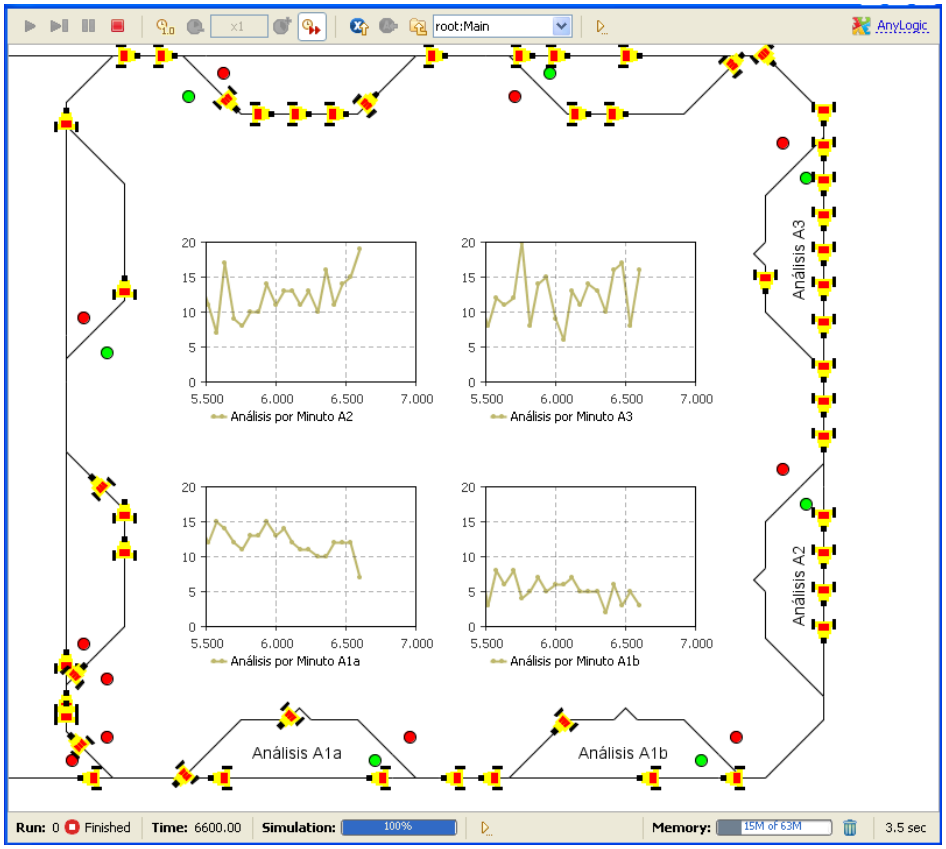
Simulación a velocidad 50



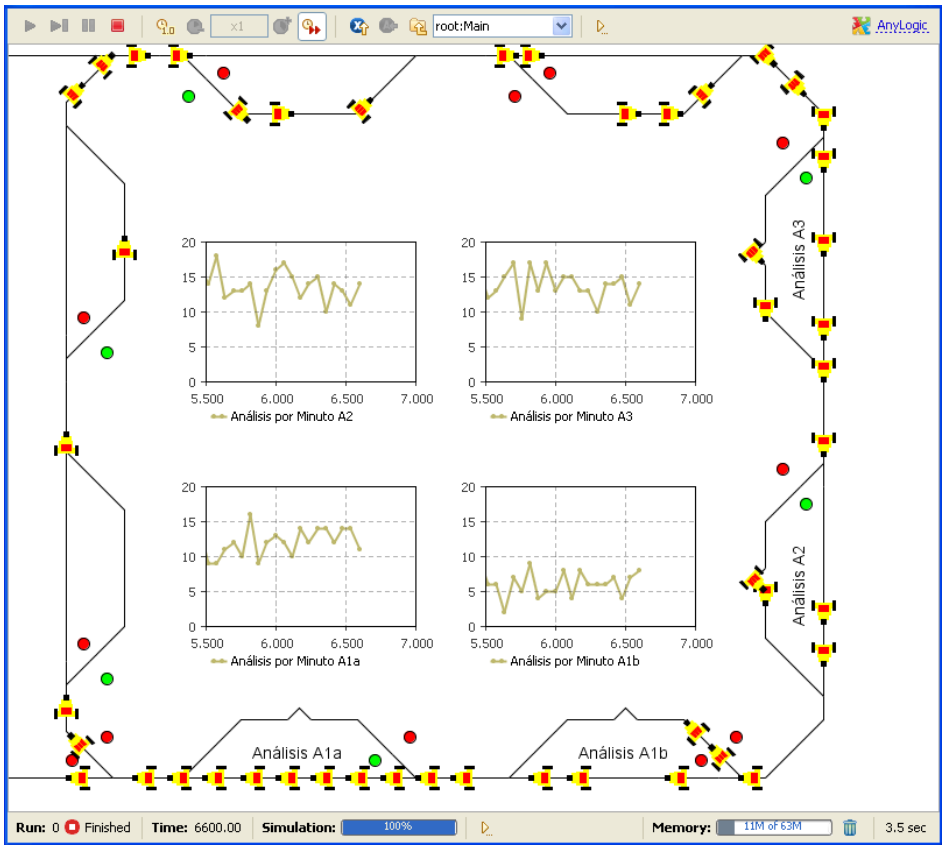
Simulación a velocidad 60



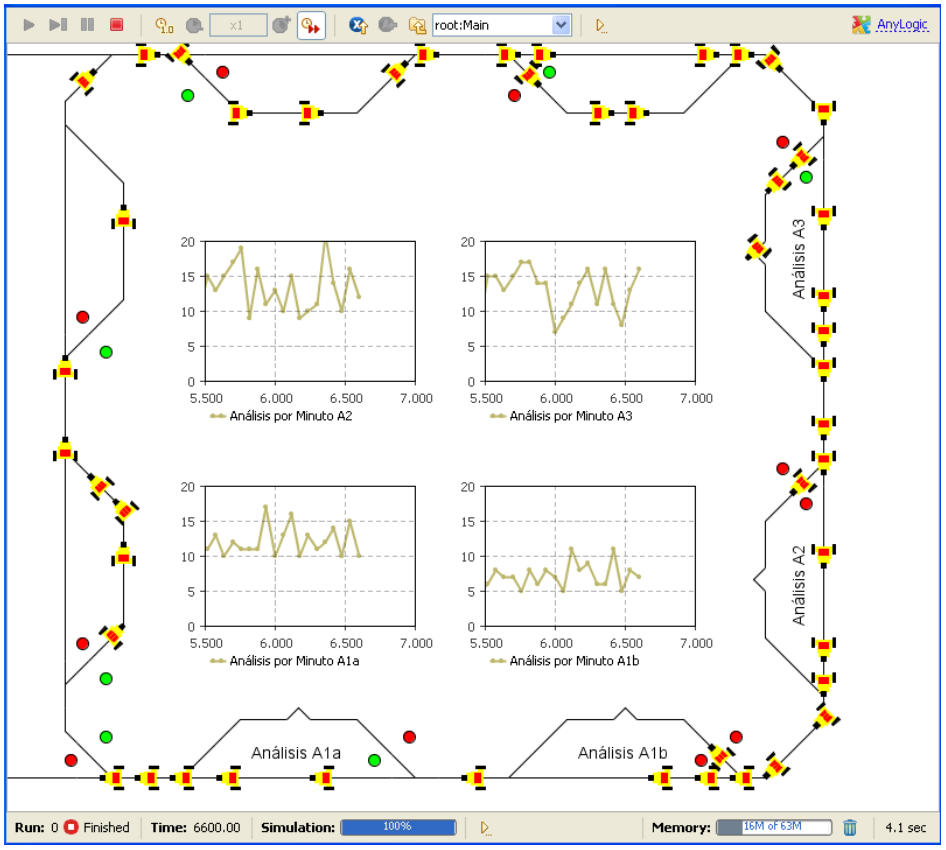
Simulación a velocidad 70



Simulación a velocidad 80



Simulación a velocidad 90



Simulación a velocidad 100

En estas gráficas se puede observar que hay un pequeño salto cuando cambiamos de 30 a 40 y a 50 pero las mejoras que ofrece el aumento de velocidad hasta llegar a 100 se van reduciendo incrementando cada vez menos. Hay que tener en cuenta que las entradas de datos son aleatorias con lo que se pueden ver picos y caídas durante la simulación que no tienen porqué estar relacionadas con el dato que se está analizando así que lo realmente importante es el área de la gráfica por donde se mueven los datos.

5.4 Balanceo de las cargas

Las dos primeras máquinas de análisis denominada A1a y A1b realizan el mismo tipo de análisis como ya se ha comentado anteriormente y por tanto se reparten la faena. Pese a repartirse la faena, se puede ver como la primera máquina siempre tiene un índice mayor de análisis que la segunda.

La explicación a este suceso es que a pesar de que estén balanceadas, el reparto se realiza a la entrada de la primera máquina. Esto hace que un vehículo asignado a la segunda máquina tarde más tiempo en llegar hasta ella y por tanto la segunda máquina mantiene su carga más tiempo, mientras que la primera al tener un recorrido menor realiza el análisis y reduce su carga situándose así en una posición preferente para que le asigne otro vehículo que tenga que realizar dicho análisis. También está el caso en que lleguen los análisis muy separados temporalmente y por tanto al llegar y no tener cargas ninguna de las máquinas, se asignan a la primera máquina y esta lo realiza antes de que llegue el siguiente. Esto conlleva a que el siguiente vehículo se encuentre en la misma situación que el anterior, ninguna de las dos máquinas tiene carga y por defecto se asigna a la primera mostrando en las gráficas de salida un mayor número de análisis en la primera máquina respecto a la segunda.

Se podría hacer un balance más óptimo usando el total de análisis que lleva cada máquina evitando el caso anterior pero primero no aumentaría el rendimiento de análisis por minuto y segundo, al requerir menos tiempo la primera máquina que la segunda, el desbalance en un determinado instante podría colapsar la entrada de la segunda máquina cortando el tráfico de los AGV.

5.5 Comparativa

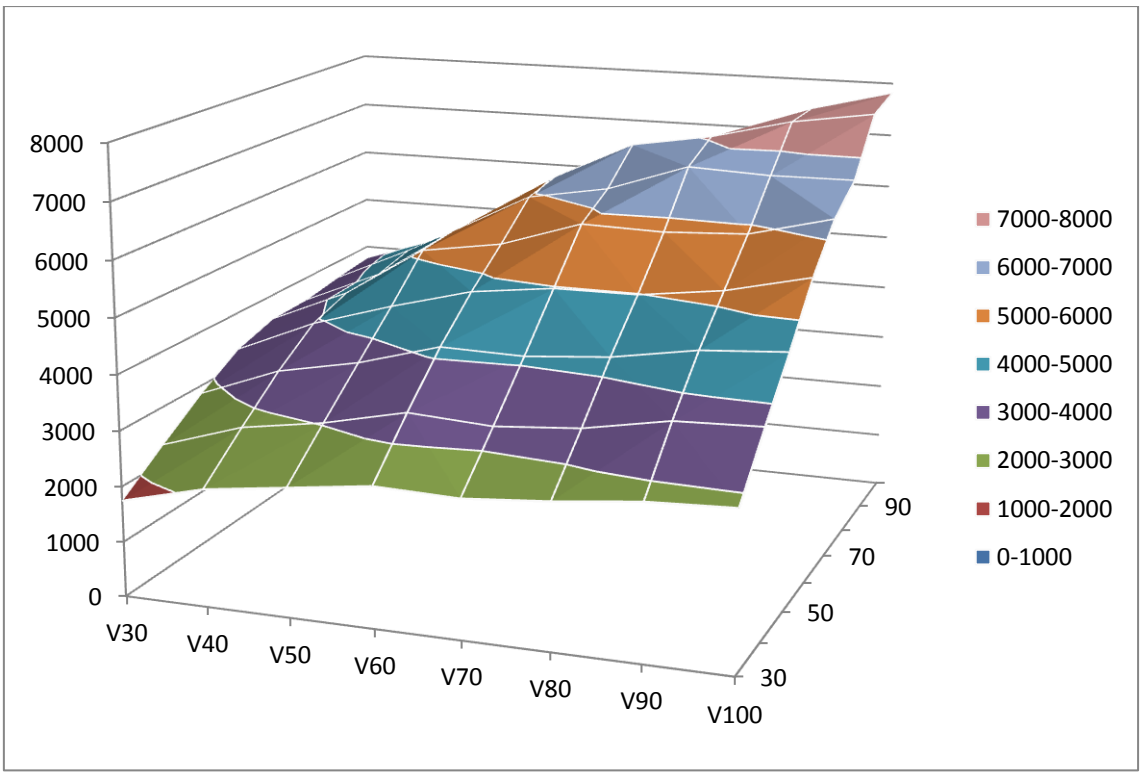
En la tabla que hay a continuación se muestra el número de muestras que se han realizado entre las 4 máquinas de análisis en 6000 segundos (100 minutos) variando el número de AGV que hay disponibles (representado por las columnas) y la velocidad a la que se movían (representado por las filas). Además se han eliminado los 600 segundos del principio para eliminar los pequeños fallos que inducen en las estadísticas la puesta en marcha de la simulación o también conocido como tiempo de calentamiento [1, 2]. Las simulaciones eran en total de 6600 segundos, con este tiempo los valores aleatorios se compensan más o menos y los 600 segundos del principio dan tiempo a que arranque la simulación independientemente del número de AGV que se utilizan (entre 30 y 100).

La simulación se ha repetido 10 veces para obtener una media ya que hay factores aleatorios, tanto en la espera de las máquinas de análisis como en la asignación de los análisis a realizar, y por tanto una única simulación puede dar un resultado poco orientativo.

Tanto en la tabla como en la gráfica se representan las velocidades mediante “V(número)” y el total de AGV usados por un número simple.

Vel\AGV	30	40	50	60	70	80	90	100
V30	1747,6	2311,4	2849,4	3300,6	3526,1	3549	3665,2	3753
V40	2123,4	2788,9	3416,6	3934,1	4168,8	4192,5	4350	4253,3
V50	2321,9	3026,5	3712,3	4388,6	5113,2	5167,8	5056,1	4784,1
V60	2520,9	3365,2	4142,5	4789,8	5348	5948,2	6063,3	5256,7
V70	2477,7	3277,4	4112,2	4975,9	5839,6	6208	6787,8	5756,5
V80	2602,5	3399,2	4239,1	4986,7	5802,3	6727,1	7036,6	6768,3
V90	2770,7	3681,7	4497,7	5186,1	5878,9	6661,3	7414,4	7421,5
V100	2838,8	3738,7	4606,7	5541,2	6262,6	6674,7	7643,5	7806,6

Tabla de comparativas de AGV



Grafica comparativa de velocidad y número de AGV

Se puede apreciar tanto en los datos como en la gráfica 3D que el aumento de AGV en pista no siempre conlleva un mayor rendimiento en el número de análisis realizados, poco a poco se va reduciendo el incremento hasta llegar un momento en que el colapso de las vías reduce la velocidad de análisis. Con la velocidad se pueden observar comportamientos extraños. Se puede ver como para ciertos valores de AGV el rendimiento tiene un comportamiento ondulado, perdiendo rendimiento para valores más altos y volviéndolos a recuperar al seguir subiendo de velocidad, pero por lo general aumentar la velocidad suele conllevar un aumento en el total de análisis realizados por unidad de tiempo.

También existe un límite que puede aparecer sin necesidad del colapso de las vías. Se puede alcanzar un máximo en el rendimiento por llegar a la saturación de las máquinas que realizan los análisis. En esta simulación no se llega a esos límites.

En la siguiente tabla vemos una comparación de cómo incrementa el total de análisis tras incrementar los AGV disponibles.

Vel\AGV	30	40	50	60	70	80	90	100
V30	100,00%	132,26%	123,28%	115,83%	106,83%	100,65%	103,27%	102,40%
V40	100,00%	131,34%	122,51%	115,15%	105,97%	100,57%	103,76%	97,78%
V50	100,00%	130,35%	122,66%	118,22%	116,51%	101,07%	97,84%	94,62%
V60	100,00%	133,49%	123,10%	115,63%	111,65%	111,22%	101,94%	86,70%
V70	100,00%	132,28%	125,47%	121,00%	117,36%	106,31%	109,34%	84,81%
V80	100,00%	130,61%	124,71%	117,64%	116,36%	115,94%	104,60%	96,19%
V90	100,00%	132,88%	122,16%	115,31%	113,36%	113,31%	111,31%	100,10%
V100	100,00%	131,70%	123,22%	120,29%	113,02%	106,58%	114,51%	102,13%

En la tabla anterior vemos una comparativa del porcentaje de análisis que se han realizado al incrementar en 10 el número de AGV disponibles. La primera columna no es un dato real debido a que no hay datos de cuantos análisis se realizan si se dispone de 20 vehículos, pero se usará como punto de partida para calcular las siguientes mediciones.

Vemos que siempre se obtiene una mejora pero que poco a poco va reduciéndose hasta llegar a los 100 donde la mayoría empeoran el rendimiento. Los rendimientos inferiores al 100% se han resaltado en rojo

En la siguiente tabla se realiza una comparación similar pero esta vez tratando la velocidad de circulación de los AGV.

Vel\AGV	30	40	50	60	70	80	90	100
V30	100,00%	100,00%	100,00%	100,00%	100,00%	100,00%	100,00%	100,00%
V40	121,50%	120,66%	119,91%	119,19%	118,23%	118,13%	118,68%	113,33%
V50	109,35%	108,52%	108,65%	111,55%	122,65%	123,26%	116,23%	112,48%
V60	108,57%	111,19%	111,59%	109,14%	104,59%	115,10%	119,92%	109,88%
V70	98,29%	97,39%	99,27%	103,89%	109,19%	104,37%	111,95%	109,51%
V80	105,04%	103,72%	103,09%	100,22%	99,36%	108,36%	103,67%	117,58%
V90	106,46%	108,31%	106,10%	104,00%	101,32%	99,02%	105,37%	109,65%
V100	102,46%	101,55%	102,42%	106,85%	106,53%	100,20%	103,09%	105,19%

En esta tabla hacemos lo mismo que en el caso anterior pero con las velocidades. Se puede ver que el comportamiento no sigue un patrón exacto dando bajas del rendimiento por un aumento en la velocidad en determinados casos (marcados en rojo) y luego vuelve a aumentar el rendimiento el aumentar la velocidad. Pero por lo general estas mejoras van reduciéndose a medida que aumentamos siendo bastante notables los saltos de 40 a 50 y de 50 a 60.

Todos estos datos hay que considerar que no se ajustan completamente a la realidad debido a que los análisis son generados de forma aleatoria y con solo 10 repeticiones del análisis por cada caso. Esto debería reducir el ruido introducido en los datos, pero los valores aleatorios y el sistema de gestión del tráfico pueden introducir estos cambios. Es por ello que antes de

realizar una inversión se debe realizar un estudio y simulación del comportamiento ya que no siempre más vehículos o más velocidad traen por consecuencia un mayor rendimiento sin olvidar que se pueda valorar de antemano como funcionará nuestra gestión del tráfico.

Con las dos comparativas anteriores podemos decidir el número de AGV y la velocidad de los mismos en función del incremento de precio que suponga aumentar el número de AGV o elegir AGV con prestaciones de velocidad superiores y de paso ver si merece la pena dicha inversión o si por el contrario perderemos prestaciones.

En la tabla que hay a continuación se muestra la desviación típica o también conocida como desviación estándar [3]. Este valor nos muestra la dispersión de los valores con los que se obtuvieron los valores medios mostrados en la tabla de comparativas de AGV. Un valor elevado significa que hay una gran diferencia entre los valores obtenidos para hacer la media. Hay que tener también en cuenta que a medida que las muestras son de mayor la dispersión crecen los valores de la desviación típica.

Vel\AGV	30	40	50	60	70	80	90	100
V30	29,8075	22,0666	35,7622	30,9918	30,4027	44,1990	30,0030	32,2318
V40	19,1729	18,5439	31,6621	34,3881	21,3531	27,8817	28,4605	120,7735
V50	26,6477	27,8418	29,4054	30,9451	31,5517	27,4785	53,6748	45,1035
V60	25,6578	21,0703	26,3702	22,2401	30,3425	48,4763	42,1480	35,8641
V70	23,0702	27,1792	35,2887	43,1392	34,8336	46,3873	54,3319	76,3737
V80	25,3476	38,2297	26,3837	33,3635	36,2248	41,2215	37,8453	576,8114
V90	40,1665	39,8917	47,3405	24,5377	41,6825	35,7400	50,3746	76,8711
V100	22,3199	26,7957	33,4167	46,5040	42,6203	28,9944	29,3229	38,6701

Tabla con las desviaciones típicas de las medias realizadas

En esta tabla podemos ver la desviación típica de las medias que se han realizado para obtener los datos. Se puede ver que exceptuando la columna que representa el uso de 100 AGV los valores son bajos y si además los comparamos con los valores medios obtenemos la siguiente tabla.

Vel\AGV	30	40	50	60	70	80	90	100
V30	1,71%	0,95%	1,26%	0,94%	0,86%	1,25%	0,82%	0,86%
V40	0,90%	0,66%	0,93%	0,87%	0,51%	0,67%	0,65%	2,84%
V50	1,15%	0,92%	0,79%	0,71%	0,62%	0,53%	1,06%	0,94%
V60	1,02%	0,63%	0,64%	0,46%	0,57%	0,81%	0,70%	0,68%
V70	0,93%	0,83%	0,86%	0,87%	0,60%	0,75%	0,80%	1,33%
V80	0,97%	1,12%	0,62%	0,67%	0,62%	0,61%	0,54%	8,52%
V90	1,45%	1,08%	1,05%	0,47%	0,71%	0,54%	0,68%	1,04%
V100	0,79%	0,72%	0,73%	0,84%	0,68%	0,43%	0,38%	0,50%

Tabla con el porcentaje del valor que representa la desviación típica respecto a la media

En esta tabla vemos como las diferencias por la generación de datos aleatoria ha sido prácticamente erradicada por el tiempo de simulación obteniendo en casi toda la tabla variaciones de los valores inferiores al 1% y unos pocos valores que se hallan entre el 1 y el 2%. Solamente en la columna de los 100 AGV podemos apreciar valores más altos. Esto es más

debido al colapso que sufrían las vías que a los valores aleatorios para realizar análisis que se les entregaba a los AGV.

La probabilidad para realizar un análisis en los casos anteriores es de 50% para el primer análisis, el que realizan las dos primeras máquinas, y un 35% para las otras dos. En una comparativa en la que se aumenta la probabilidad a 100% para el primer análisis y un 50% para los otros dos aumentamos el total de análisis realizados en una simulación. Para comparar se han usado los datos de 50 AGV a velocidad 50 y con los nuevos parámetros se pasa de una media de 3712,3 a 4565,9. Un aumento de 853,6 es equivalente a subir los 50 AGV a una velocidad de 100 o aumentar los AGV disponibles hasta 70. Esto deduce que el transportar una mayor carga y por tanto aumentar las posibilidades de ir a más máquinas de análisis aumenta el rendimiento. Esto abre una nueva vía a la hora de decidir la adquisición de AGV en función de la carga que puedan transportar.

CAPÍTULO 6: Conclusiones

6.1 Introducción

En este proyecto se pretendía conseguir una simulación de la gestión del tráfico de una planta de análisis donde un conjunto de AGV transportaba muestras desde un punto de carga hasta las máquinas de análisis, esperar que los resultados que daban las máquinas fuesen válidos y salir para descargar la carga para proceder a un nuevo ciclo de análisis. Con este proceso los AGV conseguían sustituir a las cintas transportadoras que actualmente realizan esta tarea.

Al sustituir la cinta por vehículos AGV se mejora la tolerancia a averías debido a que los AGV son independientes y no detiene todos los procesos una avería como pasa con las cintas. También permite, en caso de ampliación o modificación de la planta, adaptar el transporte a las nuevas necesidades de la planta con un coste muy inferior debido a que solo hay que modificar el software y añadir las líneas en el suelo.

6.2 Trabajo realizado

Para realizar el proyecto se han hecho una serie de actividades clasificables en tres grandes bloques: memoria, programación y reuniones. Aunque las tres están fuertemente relacionadas. La memoria incluye, aparte de este documento, la investigación de lo que se va a hacer, cómo se va a realizar y un estudio posterior de los resultados obtenidos. La programación pone en práctica lo investigado y verifica de forma experimental que se han tenido en cuenta todos los aspectos para el correcto funcionamiento. Las reuniones han servido como guía para ir marcando objetivos valorando los hitos conseguidos desde la última reunión y encarar el proyecto a buen fin tanto en el aspecto de programación como en el de la correcta realización de la memoria.

A continuación se resumen las fechas, temáticas de las reuniones y horas de reunión (apartado 6.2.1) y las actividades realizadas en los períodos existentes entre reunión y reunión con su duración y el tiempo total acumulado (apartado 6.2.2). Dicho tiempo acumulado no incluye las horas de reunión ya que estas se suman en el apartado 6.2.3 para obtener el tiempo total del proyecto con su desglose.

6.2.1 Reuniones

Por lo general se han realizado reuniones cada dos semanas de entre una y dos horas. Dichas reuniones tenían la finalidad de, en un principio, servir como guía para encarar el proyecto hacia buen camino cambiando en muchos aspectos la forma en que se suelen hacer trabajos más pequeños para las diferentes asignaturas cursadas en la carrera. Más adelante, las reuniones nos servían para exponer los avances y problemas encontrados durante las dos semanas y así consolidar los objetivos a conseguir durante el siguiente período hasta finalizar el proyecto.

Cabe recordar que las horas de la tabla que hay a continuación suponen un gasto superior (en caso de tenerse que pagar) debido a que se implica el tutor del proyecto y el alumno.

Fecha	Tipo	Horas
19/10/11	Presentación del tema	1.5
3/11/11	Reunión visión global	2
24/11/11	Reunión informe previo	2
15/12/11	Reunión informe previo	1.5
12/1/12	Reunión informe previo	2
18/1/12	Reunión informe previo	1
9/2/12	Reunión especificaciones del proyecto	1
23/2/12	Reunión especificaciones del proyecto	2
8/3/12	Reunión informar de avances	2
22/3/12	Reunión informar de avances	2
12/4/12	Reunión nuevos objetivos	1
26/4/12	Reunión informar de avances	1
10/5/12	Reunión informar de avances	2
24/5/12	Reunión cambios de la memoria	1
7/6/12	Reunión ajustes formato memoria	1
14/6/12	Reunión preparación de la presentación	2
21/6/12	Reunión ensayo de la presentación	2

6.2.2 Trabajo independiente

A continuación se detallan las horas dedicadas al proyecto de forma individual distribuidas entre los períodos de tiempo entre reunión y reunión. En cada descripción se acompaña de las horas dedicadas a dicha tarea y un total de horas dedicadas al proyecto hasta dicha tarea incluida en ese total.

	Horas	Total
19/10/11 a 3/11/11		
Buscar información del control de tráfico de AGV	6h	6h
3/11/11 a 17/11/11		
Plantear objetivos, tareas y planificación temporal y pasarlo al informe previo	12h	18h
Buscar tutoriales de Anylogic	10h	28h
17/11/11 a 24/11/11		
Programar con Anylogic ejercicios de ejemplo	10h	38h
24/11/11 a 15/12/11		
Aprender Anylogic	20h	58h
Modificar informe	8h	66h
15/12/11 a 12/1/12		
Ejemplos sencillos en Anylogic aplicables al PFC	20h	86h
Modificar informe	4h	90h
Resolver problemas Anylogic	12h	102h

12/1/12 a 18/1/12		
Modificar informe previo	8h	110h
18/1/12 a 9/2/12		
Programar ejemplos pequeños en Anylogic	15h	125h
Combinar ejemplos pequeños	6h	131h
Aprender Anylogic	10h	141h
9/2/12 a 23/2/12		
Avanzar memoria	8h	149h
Resolver dudas Anylogic	6h	155h
23/2/12 a 8/3/12		
Avanzar memoria	8h	163h
Programar Anylogic	6h	169h
8/3/12 a 22/3/12		
Modificar algoritmos para mejorar la simulación	8h	177h
Comenzar la simulación completa	5h	182h
22/3/12 a 12/4/12		
Simulación completa	36h	218h
12/4/12 a 26/4/12		
Corregir fallos de simulación completa	8h	226h
Avanzar memoria	6h	232h
26/4/12 a 10/5/12		
Memoria	18h	250h
10/5/12 a 24/5/12		
Memoria	16h	266h
24/5/12 a 7/6/12		
Memoria	24h	290h
Estadísticas	8h	298h
7/6/12 a final		
Memoria	4h	302h
Presentación	40h	342h

6.2.3 Tiempo total

En este apartado haremos un resumen de las horas dedicadas en total al proyecto en sus cuatro grandes bloques .El resumen de los tiempos totales del trabajo son:

- Memoria 130 horas
- Programación 172 horas
- Presentación 40 horas
- Reuniones 27 horas

Total 369 horas

En la descripción anterior se han agrupado tanto informe previo como buscar información o realizar estadísticas en la memoria. Aunque parte de la realización de estadísticas supuso un cambio en la simulación con el fin de mostrar y extraer los datos, se cuenta como tiempo de memoria ya que no se toca en ningún momento la gestión del tráfico.

La programación agrupa tanto la entrada de código como el aprendizaje de Anylogic y los pequeños programas de ejemplo que se encuentran en algunos tutoriales.

La presentación agrupa tanto la creación de la presentación en si como la creación de videos para la misma.

En general, las horas dedicadas al proyecto han sido similares a las que se planteaban en un principio exceptuando el tiempo para imprevistos. A la hora de la verdad el reparto no ha sido tan ajustado como en la planificación inicial. Las herramientas con las que cuenta Anylogic facilitaron la labor de gestionar el tráfico una vez se aprendía a usarlas. Por otro lado, la memoria ha llevado más horas de las previstas en un principio. La falta de experiencia y buenos hábitos a la hora de elaborar informes han hecho que elaborar la memoria se alargue considerablemente.

6.3 Resultados

Al final se ha conseguido realizar la gestión del tráfico y se ha podido contrastar su funcionamiento con una simulación. La simulación se ha realizado para que, con relativa facilidad, se puede modificar para variar el número de AGV y su velocidad de circulación. Por otra parte también se puede modificar la probabilidad con la que los AGV visitan las máquinas de análisis o, directamente, que le lleguen las peticiones de análisis desde un archivo eliminando de esa forma el factor aleatorio.

Los resultados, como se comenta en sus diferentes apartados, nos muestran un comportamiento orientativo a lo que debería ocurrir en la realidad. Con dicho comportamiento se puede apreciar las mejoras que supone un mayor número de AGV o usar AGV con una velocidad de circulación mayor. Esto nos puede ayudar a la hora de poner en práctica el cambio de cintas a AGV, en caso de que se fuese a hacer o para elegir que dispositivo funciona mejor si se pretende construir un laboratorio nuevo y decidir si es más rentable aumentar el número de AGV o elegir un modelo de AGV de mayor velocidad en caso de elegir la opción de los AGV. También ayuda a ver si una posible modificación en el área de

trabajo puede afectar de una forma inesperada al tráfico de los vehículos generando algún atasco y permitiendo realizar un cambio en la gestión del tráfico para evitar dicho problema antes de encontrarse con el problema real.

Como visión personal considero que ha ido bien el proyecto y los resultados han sido, en la medida de lo posible, bastante buenos mostrando una aproximación ajustable de cómo se comportarían los AGV en caso de sustituir las cintas transportadoras. Respecto a la planificación inicial, los tiempos requeridos para hacer la memoria han sobrepasado mucho lo planificado en un principio y en el apartado de aprendizaje y programación, en ocasiones era difícil separar el aprendizaje de Anylogic con la programación de clases ya que en muchas ocasiones se realizaban de forma simultánea.

6.4 Líneas de continuación

A partir de este proyecto se puede empezar a trabajar en alternativas a las cintas transportadoras en lugar de la sustitución. Esto conllevaría, en un primer lugar, crear una red de caminos, en lugar de una única ruta, y un sistema de gestión de tráfico adicional para calcular y/o controlar las rutas de los AGV para reducir el camino y la congestión de algunos tramos. Esto también requeriría un estado de las máquinas de estado, que ahora controlan los semáforos, para que se puedan controlar de forma externa y global el tráfico y poder deshacer estados críticos como los que se muestran en la Simulación con 100 AGV colapsada. También sería muy productivo crear una zona de estacionamiento dentro de la planta para los vehículos que están esperando resultados de los análisis. Esto reduciría el tráfico en las vías que llegan hasta las máquinas de análisis y alargaría la vida de las baterías de los AGV mejorando el rendimiento de análisis realizados por cada carga de batería.

Bibliografía

CONTROL DE TRÁFICO.

La bibliografía para ver la gestión del tráfico que se realiza ha sido complicada de encontrar ya que las empresas no liberan los algoritmos o la forma de planificar el tráfico. La poca información que se encuentra se basa en, que solo circule un AGV en un área de trabajo o que en caso de haber dos o más AGV cerca, se detengan y solo uno circule hasta alejarse de la zona conflictiva dando paso a que otro tenga permiso para moverse. Esta opción sería sencilla de implementar, pero no podría acercarse a los resultados de análisis por día que genera un sistema de transporte por cintas.

En la web de Savant Automation podemos encontrar información un poco más detallada aunque sin entrar en detalles técnicos o código de cómo se gestiona el tráfico de vehículos AGV. En la dirección http://www.agvsystems.com/basics/traffic.htm nos indica los 3 tipos de gestión del tráfico.

- El control de zona http://www.agvsystems.com/basics/zone.htm
- El control frontal http://www.agvsystems.com/basics/forward.htm
- El control combinado http://www.agvsystems.com/basics/comb.htm

El control de zona se basa en dividir el área de circulación de los AGV en pequeños segmentos con una entrada y una salida. Antes de entrar en cada sección el AGV se comunica con un sistema central que autoriza la entrada si la zona está despejada o se avisan entre ellos por radiofrecuencia que van a entrar en una zona y por tanto los demás que quieran entrar en dicha zona deberán esperar a que el que está dentro avise de que se puede entrar.

El control frontal es un sistema autónomo del AGV. Se basa en detectar mediante mecanismos como los ultrasonidos, cámaras o sensores de impacto obstáculos que impiden el paso. Este sistema es muy recomendable cuando el tráfico de los vehículos AGV se realiza en una dirección y durante un gran recorrido. Si el vehículo situado en frente se debe detener por la razón que sea, el AGV que le sigue se detendrá al ver que la distancia es inferior a la considerada como segura. Este sistema tiene un problema y es que en los cruces se pueden producir colisiones debido a que el sistema solo detecta obstáculos delante y no detectaría un vehículo aproximándose por los laterales

El control combinado es una combinación de los anteriores. Los vehículos AGV se aseguran de que no tienen obstáculos delante a la hora de avanzar y el control de zona se encarga de que en los puntos conflictivos, donde se podría producir un impacto lateral, se acceda de forma segura.

En el proyecto usamos un control combinado ya que los AGV controlan siempre la distancia con el vehículo predecesor y las intersecciones son controladas por semáforos externos a los AGV para garantizar que no se producen colisiones.

ANYLOGIC

En el campo de Anylogic si que se encuentra más información de cómo realizar un proyecto básico y que en gran parte ha ayudado mucho para empezar a usarlo. Los tutoriales que utilice para empezar el proyecto fueron los creados por Khoi Tran http://de.linkedin.com/pub/khoi-tran/14/b80/5a3 y que se pueden encontrar en el enlace de http://www.slideshare.net/simbean . Pero cuando el proyecto crece y se requieren casos más complejos se debe usar la ayuda de Anylogic.

REFERENCIAS

[1] —, *Simprocess Help*, Simprocess, [Disponible en: http://www.simprocess.net/docs/SP48/WebHelp/sphelp/simulate_run_settings.html]

[2] J. Merrick, *Steady-State Statistical Analysis, Course on*, Virginia Commonwealth University, August 20, 1998.[Disponible en: http://www.courses.vcu.edu/MATH-jrm/OPER641/Slides/623SteadyStateStatistics.ppt]

[3] Explicación y ejemplo de la variación típica o estándar. http://www.disfrutalasmatematicas.com/datos/desviacion-estandar.html

4583-3: VEHÍCULOS-ROBOT DE GUIADO AUTOMÁTICO PARA ALMACENES Y PLANTAS DE MANUFACTURACIÓN

Proyecto final de carrera

 En este proyecto de final de carrera se realiza la gestión del tráfico de AGV y la simulación de su comportamiento al circular por una planta de estudio. Con la simulación se puede ver como varía el comportamiento de la planta al modificar el número de AGV y la velocidad a la que circulan. La planta objeto de estudio es un laboratorio de análisis clínico en el que se ha sustituido el sistema de transporte interno basado en cintas por uno con AGV, con lo que se ha podido comprobar que dicha sustitución es factible.

 En aquet projecte de final de carrera s'ha fet la gestió del trànsit d'AGV i la simulació del seu comportament al circular per una planta d'estudi. Amb la simulació es pot veure com varia el comportament de la planta en modificar el nombre d'AGV i la velocitat a la que circulen. La planta objecte d'estudi és un laboratori d'anàlisis clíniques en el que s'ha substituït el sistema de transport intern basat en cintes per un amb AGV, amb el que s'ha pogut comprovar que aquesta substitució és factible.

 In this capstone project an AGV traffic management system has been done, as well as the simulation when they circulate in a case study plant. With those simulations it is possible to observe the behaviour of the plant under different values for the number of AGVs and their speed. The plant of case study has been an automated laboratory of clinical analyses where the conveyor-belt based internal transport system has been replaced by AGVs. The corresponding simulation data shows that this option is feasible.