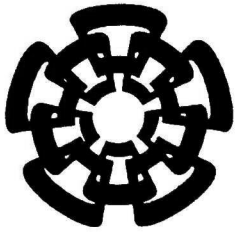


xx (131339.1)



CINVESTAV

Centro de Investigación y de Estudios Avanzados del I.P.N.
Unidad Guadalajara

Control de tráfico urbano basado en sistemas multiagentes

CINVESTAV
IPN
ADQUISICION
DE LIBROS

Tesis que presenta:
Joel Antonio Trejo Sanchez

para obtener el grado de:
Maestro en Ciencias

en la especialidad de:
Ingeniería Eléctrica

CINVESTAV I.P.N.
SECCION DE INFORMACION
Y FUNDAMENTACION

Directores de Tesis
Dr. Luis Ernesto López Mellado
Dr. Antonio Ramírez Treviño

Guadalajara, Jalisco, Septiembre de 2006.

GLASIF.: I K16S.G8 .775 2006
ADQUIS.: SSI-416
FECHA: 16-V-2007
PROCED.: Don.-2007
\$ _____

ID: 130771-1001

Control de tráfico urbano basado en sistemas multiagentes

**Tesis de Maestría en Ciencias
Ingeniería Eléctrica**

Por:

Joel Antonio Trejo Sanchez

Licenciado en Ciencias de la Computación
Universidad Autónoma de Yucatán 1998-2003

Becario de CONACYT, expediente no. 191501

Directores de Tesis

Dr. Luis Ernesto López Mellado

Dr. Antonio Ramírez Treviño

CINVESTAV del IPN Unidad Guadalajara, Septiembre de 2006.

Agradecimientos:

*A mis padres y a mis hermanos por su apoyo incondicional,
A mí amada esposa quien siempre fue mi inspiración,
A mis asesores por el tiempo y profesionalismo dedicado,
A mis compañeros por sus sabios consejos,
Agradezco a Dios quien siempre me sostiene,
Al **CONACYT** por su apoyo.*

Introducción

Capítulo 1. Especificación de sistemas de control de tráfico urbano.

1.1. Sistemas de control de tráfico urbano.....	1
1.1.1. Control de tráfico urbano.....	1
1.1.2. Semáforos y coordinación.....	1
1.1.3. Conceptos sobre sistemas de control de tráfico urbano	2
1.1.4. Sistemas de eventos discretos.....	4
1.1.5. Sistemas multiagentes.	5
1.2. Enfoques de modelado de sistemas de control de tráfico urbano.....	6
1.2.1. Modelo continuo.	7
1.2.2. Modelo discreto.....	8
1.2.3. Modelo híbrido.....	9
1.3. Metodologías de control de tráfico urbano.....	10
1.3.1. Sistemas multiagentes.....	10
1.3.2. Redes de Petri.....	14
1.3.3. Lógica difusa.....	21
1.3.4. Otros enfoques.....	23
1.4. Conclusiones.....	27

Capítulo 2. Requerimientos para el desarrollo de un sistema de control de tráfico.....

2.1. Simulador.	28
2.1.1. Integración del sistema de control de tráfico urbano.	29
2.2. Especificación de requerimientos.	31
2.2.1. Descripción del problema.....	31
2.2.2. Evolución del sistema.....	31
2.2.3. Definición de requerimientos.	31
2.3. Conclusiones.	35

Capítulo 3. Esquema multiagente para el control de tráfico urbano..

3.1. Introducción.....	36
3.2. Agente de control.....	37
3.2.1. Escenario del agente de control.....	37
3.2.2. Interacción del agente de control con la máquina de simulación.....	39
3.2.3. Arquitectura del agente de control.....	40
3.3. Coordinación de agentes de control.....	59
3.3.1. Escenario.....	59
3.3.2. Comunicación entre agentes.....	60
3.3.3. Sincronización de políticas de control.....	61

3.4. Conclusiones.	63
<i>Capítulo 4. Implementación del sistema de control de tráfico urbano basado en sistemas multiagentes.....</i>	64
4.1. Introducción.....	64
4.2. Estrategia general.....	65
4.2.1. Agente de control.....	65
4.2.2. Interacción con la máquina de simulación.....	79
4.2.3. Comunicación entre agentes de control.....	81
4.3. Conclusiones.....	83
<i>Capítulo 5. Análisis de resultados y pruebas.....</i>	84
5.1. Introducción.....	84
5.2. Pruebas y simulaciones realizadas.....	88
5.3. Resultados obtenidos.....	90
5.4. Conclusiones.....	91
<i>Conclusiones.....</i>	92

Introducción

Es común experimentar congestión de tráfico urbano, en las principales ciudades del mundo; esto se debe principalmente al incremento que se presenta año con año en el número de vehículos que circulan en la zona metropolitana de las grandes ciudades.

La congestión de tráfico urbano causa considerables costos debido a pérdidas de tiempo, incrementa la posibilidad de accidentes, y los problemas de contaminación en las principales ciudades por lo que tiene un impacto negativo en el ambiente. También es responsable de problemas de salud tales como estrés, ruido y complicaciones similares. La solución de aumentar la dimensión de la red de tráfico urbano no siempre resulta ser la mejor opción además es muy difícil y muy costosa, especialmente en áreas urbanas.

Una de las principales respuestas al problema del control de tráfico urbano es reducir el tiempo de espera de los usuarios en la red de tráfico. Se puede reducir el tiempo de espera de los usuarios en la red de tráfico por medio del cambio dinámico de las señales desplegadas en los semáforos, y que este cambio se realice de acuerdo a la demanda de tráfico y a la coordinación con intersecciones adyacentes.

Algunas de las metodologías para el control de tráfico urbano se basan en inteligencia artificial para lograr una adaptabilidad del control de acuerdo a la demanda. De esta forma existen técnicas basadas en lógica difusa, redes neuronales, sistemas multiagentes, teoría de grafos, programación lineal, entre otras.

En esta tesis se propone un esquema basado en agentes para control de tráfico urbano; debido a las facilidades que este enfoque representa para interactuar con su ambiente y llevar a cabo una eficiente coordinación con otros agentes. Cada intersección es controlada por un agente y de esta forma cada agente se puede comunicar con intersecciones adyacentes, las cuales a su vez son controladas por agentes. Asimismo, cada intersección es modelada por una red de Petri, la cual se encarga de enviar eventos de cambio de luz cuando se disparan sus transiciones.

Nos apoyamos de un simulador de tráfico urbano orientado a eventos para realizar las pruebas del controlador. De esta forma, la metodología de control de tráfico urbano envía eventos de cambio de luz a las intersecciones pertenecientes a la red de tráfico urbano representado en el simulador.

La descripción de nuestro trabajo esta organizado como sigue:

En el capítulo 1, se presentan las nociones básicas de todo sistema de control de tráfico urbano; se define formalmente como está conformado un sistema de control de tráfico urbano y cómo se realiza el modelado de los mismos. Asimismo presentamos el estado del arte del control de tráfico urbano y las metodologías existentes.

En el capítulo 2 se presenta la especificación inicial del sistema de control de tráfico urbano. En este capítulo presentamos el diseño de nuestro sistema de control, incluyendo desde los requerimientos generales hasta los específicos.

En el capítulo 3 se propuso la arquitectura del agente de control así como la descripción y funcionalidad de cada uno de sus componentes. Establecemos la forma en que el agente de control interactúa con otros agentes asignados a intersecciones aledañas.

En el capítulo 4 se describe la implementación del sistema de control. Se define como es construido el agente de control y que herramientas son utilizadas para la programación de cada uno de sus componentes. De igual manera presentamos los protocolos empleados para realizar la comunicación entre los agentes y la interacción con la máquina de simulación.

En el capítulo 5 se presenta el caso de estudio de la simulación de una red de tráfico. En este capítulo se muestra la forma en que nuestro sistema de control es capaz de adaptarse a la demanda de tráfico por medio de diferentes pruebas realizadas con diferentes condiciones de tráfico.

Capítulo 1

Especificación de sistemas de control de tráfico urbano

Resumen: En este capítulo se presentan los conceptos básicos referentes al control de tráfico urbano. Se analizan los principales enfoques para control de tráfico urbano así como sus características, ventajas y desventajas.

1.1. Sistemas de control de tráfico urbano

El control de tráfico se encarga de administrar el movimiento de los vehículos en la estructura de las calles. Principalmente los objetivos del control de tráfico urbano son el de reducir la congestión en las calles, y disminuir el tiempo que los usuarios de la red de tráfico tienen que esperar.

1.1.1. Control de tráfico urbano

Un sistema de control de tráfico urbano puede ser tan sencillo como poner una señal de alto en una intersección, puede ser un proceso tan complejo, que requiera de la instalación de un conjunto de dispositivos de control en continua predicción y comunicación con otras intersecciones.

Los sistemas de control están encargados de proponer las estrategias de control en la red de tráfico. La determinación de aplicar una estrategia de control depende de las condiciones de tráfico; incluidas la velocidad y tiempo de viaje de los vehículos, densidad, flujo de tráfico y retraso [K.J. Button, 2001]. Las condiciones de tráfico son monitoreadas a través de sensores, cámaras, vehículos de prueba, y otras.

1.1.2. Semáforos y coordinación

Los semáforos son una herramienta efectiva para reducir la congestión en las calles involucradas en una intersección. Se han definido tres formas de operación en el control de

semáforos [K.J. Button, 2001]: tiempo fijo, semi-adaptable y adaptable. Con el control de semáforos en tiempo fijo, el tiempo de ciclo, las fases, y el cambio de fases están predeterminados. En el control adaptable los parámetros de la estrategia de control cambian dinámicamente conforme cambian las condiciones de tráfico; se utilizan generalmente sensores que están puestos estratégicamente en la intersección para monitorear la demanda de tráfico. El control de semáforos semi-adaptable es cuando solo una parte de la intersección tiene control dinámico de acuerdo a las condiciones de tráfico. El control semi-adaptable es apropiado cuando solo una calle en la intersección tiene el mayor flujo de autos y las otras calles tienen flujo esporádico.

El control de las intersecciones puede hacerse localmente o en coordinación con otras intersecciones. Cuando la coordinación es aplicada de forma óptima el flujo de tráfico en un área urbana puede suavizarse de tal forma que se mantiene constante. Una coordinación eficiente requiere de sensores que monitoreen las condiciones de tráfico, así como de dispositivos de comunicación y de algoritmos para optimizar las señales de control.

1.1.3. Elementos de un sistema de control de tráfico urbano

Los sistemas de control de tráfico deben tomar en cuenta los siguientes elementos de una intersección en un sistema de tráfico urbano (figura 1.1).

- Una *calle* es un recurso de la red de tráfico donde los vehículos ocupan un segmento de la misma. Una calle puede tener uno o más carriles donde cada carril puede tener su propia cola de espera y sus propias métricas de control.
- Un *vehículo* es una entidad básica del sistema que se mueve en una dirección y es controlado por un conductor. Un vehículo puede ser un auto, un autobús, una ambulancia, una motocicleta, etcétera.
- Una *intersección* consiste de un conjunto de calles relacionadas entre sí y en el área de cruce.

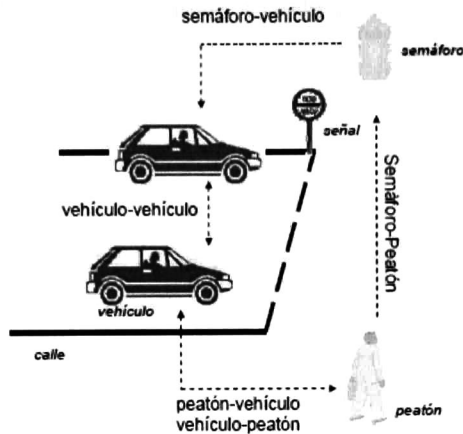


Figura 1.1. Intersección en una red de tráfico urbano.

- Un *Semáforo* es el elemento de la intersección responsable de desplegar las estrategias de control establecidas en la intersección.
- Un *ciclo* es la repetición de una secuencia de señales en una intersección.
- Una *fase* es una parte del ciclo, durante el cual un subconjunto de calles tienen derecho a paso simultáneamente.
- El *estado del sistema de tráfico urbano* indica la situación en la que se encuentra el sistema en un instante de tiempo dado, y se puede conocer a través de las siguientes variables: número de vehículos en las calles, flujo de los segmentos de calle, densidad de los segmentos de calle y número de accidentes.
- El *flujo* se define para el caso de control de tráfico urbano, como el número de vehículos que pasan por algún punto, específicamente una intersección en un período de tiempo. El flujo se divide entre el número de calles involucradas en la intersección. Para obtener el flujo tenemos la siguiente fórmula:

$$q_j = N_j/T$$

Donde:

q_j es el flujo,

N_j es el número de vehículos que cruzan la calle j .

T es el tiempo durante el cual se calcula el flujo.

- La *densidad* se define como el número de vehículos en un tramo de la calle dividida entre la longitud de la calle:

$$\rho_j = N_j / L$$

Donde:

ρ_j es la densidad,

N_j es el número de vehículos que cruzan la calle j .

L es la longitud de la calle.

1.1.4. Sistemas de eventos discretos

Los sistemas de eventos discretos [P. Loborg, 1994] son aquellos que están compuestos por elementos que manejan entidades discretas, es decir numerables y diferenciables entre sí. Su funcionamiento está caracterizado por una sucesión finita o infinita de estados, delimitados por eventos. Cuando ocurre un evento se genera un cambio de estado, generalmente estos cambios se presentan de manera asíncrona.

El control de tráfico urbano es un sistema complejo que puede ser modelado como un sistema de eventos discretos.

El modelado de un sistema es la abstracción que describe el comportamiento del sistema. En el modelado es necesario identificar las variables de entrada y las variables de salida, es necesario también identificar las variables de estado, y establecer las relaciones matemáticas que involucran la entrada $u(t)$, la salida $y(t)$ y el estado $x(t)$ del sistema. Una noción importante en Sistemas de Eventos Discretos es el concepto de estado.

El estado $x(t_0)$ [P. Loborg, 1994] de un sistema en el instante t_0 es la información en t_0 que junto con la entrada $u(t)$, para $t \geq t_0$, determina la salida $y(t)$, para todo $t \geq t_0$.

El proceso de modelado consiste en determinar las relaciones matemáticas convenientes que involucran la entrada, la salida y el estado, estas relaciones se conocen como la dinámica de un sistema. En particular, nos interesa obtener expresiones para $x(t)$ proporcionando $x(t_0)$ y la función de entrada $u(t)$, $t \geq t_0$. El conjunto de ecuaciones requeridas para especificar el estado $x(t)$ para todo $t \geq t_0$, dado $x(t_0)$ y la función $u(t)$, $t \geq t_0$, se llaman ecuaciones de estado.

En un SED las señales de los sensores y de los actuadores generalmente son modeladas como alfabetos de entrada Σ_{in} y alfabetos de salida Σ_{out} del sistema. A continuación presentamos la definición formal de un sistema de eventos discretos.

1.1.5. Sistemas multiagentes

Antes de poder dar una definición clara de los sistemas multiagentes, debemos conocer la definición de agente. Dar una definición de agente es una tarea difícil, dado que, en la actualidad no existe una definición universalmente aceptada de lo que es un agente; pero todas las definiciones concuerdan en que la autonomía es parte esencial de un agente. Nosotros presentamos a continuación la definición dada por Ferber.

Ferber [Ferber, 99] presenta la siguiente definición: Un agente es una entidad física o virtual que

- es capaz de actuar en su ambiente,
- interactúa con otros agentes,
- es guiado por un conjunto de tendencias,
- posee sus propios recursos,
- es capaz de percibir su ambiente
- tiene una representación parcial de su ambiente (quizá ninguna),
- posee habilidades y ofrece servicios,
- puede replicarse,
- su comportamiento tiende a satisfacer sus objetivos, tomando en cuenta los recursos y habilidades disponibles, y dependiendo de su percepción, representación y los mensajes que recibe.

Cuando hablamos de sistemas multiagentes nos referimos al campo de la inteligencia artificial que provee los principios para la construcción de sistemas complejos que involucran múltiples agentes y mecanismos para coordinación de agentes independientes.

Los sistemas multiagente están formados por agentes o sistemas autónomos capaces de trabajar conjuntamente para la resolución de problemas, con la particularidad que cada

uno de ellos dispone de información o de capacidad incompleta para solucionar dicho problema.

Los agentes pueden ser de tres tipos principalmente: reactivos, proactivos y cooperativos.

- **Reactivos.** Los agentes inteligentes son capaces de percibir su ambiente y responden rápidamente a los cambios que ocurren en el, con el fin de satisfacer sus objetivos.
- **Proactivos.** Los agentes toman la iniciativa con el fin de satisfacer sus objetivos.
- **Cooperativos.** También llamados de interacción social, son capaces de interactuar con otros agentes (posiblemente humanos) con el fin de satisfacer sus objetivos.

Los sistemas multiagentes tienen las siguientes características.

- Cada agente tiene insuficiente capacidad para resolver el problema por si solo.
- Cada agente tiene un comportamiento concurrente.
- No hay control global del sistema.
- Los datos son descentralizados.

1.2. Modelado de sistemas de control de tráfico urbano

Con el fin de estudiar las propiedades y comportamiento de un sistema un modelo debe ser definido, el cual representa una abstracción del sistema. El uso de modelos de tráfico, nos da la oportunidad de analizar, predecir y controlar el comportamiento de los sistemas de tráfico. En un modelo es necesario identificar los parámetros de entrada y de salida del sistema, las variables de estado del sistema, y las relaciones que existen entre ellas. En un sistema de control de tráfico urbano los parámetros para establecer el comportamiento del sistema son:

- **Parámetros de entrada.**
 - Las intersecciones involucradas en la red de tráfico.
 - Las calles involucradas en las intersecciones.
 - El flujo de los segmentos de calle.
 - La densidad de los segmentos de calle.

- Parámetros de salida.
 - La secuencia de las fases involucradas en las intersecciones.
 - El tiempo asignado a cada fase involucrada en las intersecciones.

Presentamos a continuación los principales enfoques de modelado de sistemas de control de tráfico urbano.

1.2.1. Modelos continuos

Los modelos continuos consideran principalmente tres variables para el modelado: densidad, velocidad y flujo.

Método de Webster

Con el objetivo de optimizar las variables de control de tráfico en este método se necesita una función objetivo. Generalmente se escoge la función objetivo que se relaciona con el tiempo total de retraso D en una intersección. Para minimizar D se requiere una fórmula que estime retraso promedio d_i de los vehículos que esperan en una calle debido a las señales de tráfico. Si a una calle se le da un tiempo de rojo muy largo, se puede formar una cola muy larga de vehículos en esa calle. Es fácil deducir que el tiempo de verde debe depender en el volumen de tráfico en esa calle.

Webster [J. Button, 2001] desarrolló una fórmula para determinar el óptimo tiempo de ciclo y tiempos de verde minimizando el tiempo total de retraso. El método de Webster ha sido diseñado para estimar el retraso promedio por vehículo como una función de (a) la entrada de tráfico promedio en la calle i (x_i vehículos por segundos), (b) la proporción de tiempo que se le da luz verde a la calle i (y_i), (c) El flujo de saturación de la calle (s_i vehículos por segundo), y (d) El tiempo de ciclo (τ segundos). La fórmula es:

Modelo de partículas

Cada vehículo es representado por una partícula [Rodrigues L., 2004] en una retícula dimensional con L sitios. Cada sitio puede ser ocupado o no por una partícula. La velocidad y la posición de la partícula j se denota por x_j y v_j respectivamente. La distancia d_j entre dos sitios consecutivos se representa por a partículas.

El estado del sistema se define de acuerdo a las posiciones $\{x_j\}$ y velocidades $\{v_j\}$ de las N partículas. La densidad de partículas $\rho = N/(aL)$ es el número de partículas por unidad de longitud. El flujo está dado por:

$$\phi = \rho v.$$

Donde v es la velocidad promedio y está dada por:

$$v = \frac{1}{N} \sum_j \langle v_j \rangle.$$

Redes de Petri continuas

Las redes de Petri continuas pueden ser usadas para modelar un sistema continuo de tráfico urbano. La red de tráfico se divide en secciones [Júlvez, 2001]. Cada sección es modelada por una red de Petri continua. El modelo completo se obtiene uniendo todas las secciones; el disparo de la transición que conecta las secciones representa el flujo (Φ) de autos de una sección a otra.

La cantidad que se dispara en una transición, es un número real no negativo. El estado de una RP continua en el instante de tiempo τ es un vector $m(\tau)$ de números reales no negativos. El flujo de tráfico que pasa por la transición t en el tiempo τ se define como $f[t](\tau) = \lambda(\tau) \text{enab}(t, m(\tau))$, donde $\lambda(\tau) > 0$, representa la velocidad de la transición y $\text{enab}(t, m(\tau))$ es el grado en el que están habilitadas las transiciones. La evolución de un marcado está dada por: $m(\tau) = C \cdot f(\tau)$, donde C representa la matriz de incidencia de la red.

1.2.2. Modelo discreto

Cuando hablamos de control de tráfico urbano, debemos tomar en cuenta los elementos implicados en las intersecciones. En una intersección, cuando esta contiene semáforos, los desplazamientos (movements) son combinados para formar fases, las cuales tienen asignado un intervalo de tiempo durante el cual los desplazamientos pertenecientes a la fase tienen derecho a cruzar la intersección. En el modelo discreto se toma en cuenta el arribo de tráfico como una secuencia de llegadas de vehículos individuales en lugar de tomarse como variables de flujo y densidad.

Redes de Petri

En un sistema de control de tráfico urbano; el controlador debe asegurar la operación eficiente de la intersección, el controlador no debe permitir desplazamientos conflictivos que tengan derecho de paso simultáneamente. Además, el controlador debe permitir el cambio de una fase a otra cuando las condiciones de tráfico sean dadas.

Las redes de Petri son un formalismo efectivo para el modelado de sistemas de control de tráfico urbano ya que permite tomar en cuenta los desplazamientos permitidos en las fases y los tiempos asignados a ellas.

1.2.3. Modelos híbridos

Un modelo híbrido para control de tráfico urbano se propone cuando una parte del sistema se modela con un modelo continuo, y otra parte puede ser modelada por un modelo discreto. Una red de tráfico urbano puede ser modelada eficientemente por medio de un modelo híbrido; en el cual, el comportamiento de los vehículos en la estructura de tráfico es modelado por medio de variables del estado del sistema, tales como el flujo y densidad, es decir, se tiene un modelo continuo para el comportamiento de los vehículos; y el cambio de las luces son representados por medio de un modelo discreto. En [Di Febbraro A., 2004], son usadas redes de Petri continuas para modelar las calles, mientras que para modelar las intersecciones son usadas redes de Petri discretas.

TRANSYT

Es el modelo más famoso para establecer los tiempos de los semáforos en una red de tráfico urbano. TRANSYT [Button K., 2001] es un modelo de control de tráfico urbano basado en tiempo fijo e incluye algoritmos para optimizar los tiempos asignados a las fases. TRANSYT requiere de un modelo detallado de la red de tráfico, el cual incluye la estructura de la red, flujos de saturación, tiempos de viaje, y los principales orígenes de vehículos y destinos de los mismos. En el modelo las entradas de los flujos son constantes sobre ciertos períodos de tiempo.

Los principales elementos del modelo son los relacionados con el flujo de tráfico. Esencialmente corresponden a histogramas con un valor correspondiente al flujo de vehículos por hora (eje y) contra el tiempo (eje x). El modelo calcula los flujos, retrasos, y

altos durante el periodo de tiempo estudiado. Para optimizar los tiempos asignados, el modelo cambia los tiempos sistemáticamente y en cada cambio que realiza evalúa el rendimiento del modelo tomando en cuenta el tiempo de espera de los vehículos.

SCOOT

La forma en que el modelo SCOOT [Button K., 2001] funciona es muy similar a la del TRANSYT, pero el tiempo de verde es tomado en cuenta de tal forma que minimice el “grado de saturación” en cada intersección. El tiempo de ciclo es calculado con el fin de reducir el máximo “grado de saturación” abajo del 90% si es posible. El modelo de optimización es establecido al principio y los cambios en las señales son implementados dinámicamente, con pequeños cambios frecuentemente aplicados a los tiempos asignados a las fases.

SCATS

SCATS [Button K., 2001] propone tres formas de operación: adaptativa, fija y aislada. En la estrategia fija, las señales no responden a cambios en los flujos; las señales siguen planes de tiempo predeterminados y de esta forma se establece la coordinación. En la estrategia aislada todas las señales relevantes funcionan en su propia intersección de acuerdo a una estrategia fija o una estrategia adaptativa.

1.3. Metodologías de control de tráfico urbano

En esta sección presentamos los principales enfoques para establecer un modelo de control de tráfico urbano. Las metodologías presentadas aquí son analizadas y se obtienen los beneficios y complicaciones que implican la implementación de estos enfoques.

1.3.1. Sistemas Multiagentes

Los sistemas multiagentes (SMA) son un paradigma de la inteligencia artificial distribuida, en el que los agentes tienen la capacidad de interactuar con su ambiente y con otros agentes para tomar las acciones más adecuadas para alcanzar sus objetivos. Los agentes tienen la capacidad de cooperar para obtener una coordinación eficiente.

Control de tráfico urbano usando Sistemas multiagentes.

- En [Roosmond, 2000] se clasifican tres tipos de agentes en control de semáforos.

- Las carreteras se representan por agentes especiales de segmentos de carreteras (RSA).
- Las intersecciones controladas son representadas por agentes de intersecciones (ITSA).
- Para áreas o rutas específicas, hay un agente de área (alto nivel).

En esencia el diseño consiste en muchos agentes ITSA, algunos agentes de autoridad (de ruta y de área) y opcionalmente RSA's. Los agentes ITSA (figura 1.2) toman decisiones acerca de como controlar sus intersecciones basado en sus objetivos, capacidades, conocimiento, percepción y datos.

El modelo de predicción el cual es alimentado por muchas entradas, da las predicciones con tráfico actual y ajusta los cambios necesarios para hacer esas predicciones más efectivas. La estrategia de control consiste en utilizar el modelo de predicción para calcular la estrategia de control óptima. El ITSA verifica con otros agentes adjuntos el esquema de control de tráfico y por último planea la estrategia de control de semáforos.

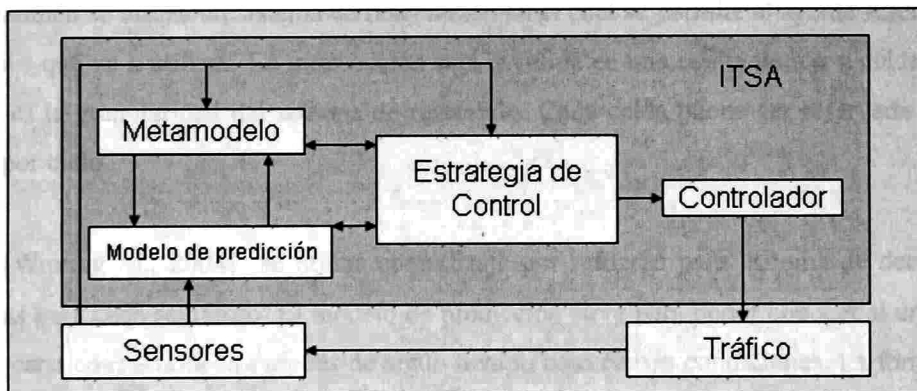


Figura 1.2. Arquitectura de un agente ITSA.

- Las funciones objetivo en control de tráfico urbano están relacionadas con minimizar el tiempo de espera promedio de los usuarios y maximizar la cantidad de usuarios que entran en el sistema. En [Dresner, 2004] se define como C el conjunto de todos los vehículos v_i que pasan a través de una intersección en un periodo de tiempo. El sistema calcula que un vehículo debería hacer el viaje de un punto A a un punto B en un tiempo $t_0(i)$, pero el

vehículo en realidad realiza el viaje en un tiempo $t(i)$. Por lo tanto se calcula el retraso promedio como.

$$\frac{1}{|C|} \sum_{v_i \in C} t(i) - t_0(i)$$

En este trabajo los conductores y los agentes semáforo interactúan directamente.

- El agente vehículo calcula cuando va a alcanzar el semáforo dada su velocidad actual.
- El agente manda un mensaje a la intersección informándole al semáforo el tiempo en el que espera llegar.
- La intersección responde con el tiempo durante el cual estará en verde.
- El agente puede hacer ajustes a sus parámetros para asegurar que alcanzará el verde.

También se utiliza un sistema de reservación en el cual se permite al agente reservar los espacios que va a utilizar. La intersección está dividida en una rejilla de $n \times n$ celdas en el que n es la granularidad del sistema de reservado. Cada celda puede ser reservada por un carro por ciclo.

- En [Wiering M., 2004] se utiliza aprendizaje por refuerzo para la toma de decisiones óptimas para cada semáforo. El modelo de predicción sirve para poder conocer si una calle puede estar congestionada después de algún tiempo bajo ciertas condiciones. La forma más simple de predicción de tráfico en una intersección es medir el tráfico en un periodo determinado, y suponer que las condiciones serán las mismas para el próximo periodo.
- Los sistemas multiagentes pueden ser utilizados para la predicción. Cada agente puede tener dos capas, una para las decisiones para el control del vehículo y otra para las decisiones de que ruta seguir. En [Moriarty y Langley 1998] se utilizó aprendizaje por refuerzo para distribuir control de tráfico.

El aprendizaje por refuerzo permite al agente interactuar con su ambiente y aprender por medio del algoritmo establecido; ha sido aplicado para encontrar buenas soluciones a problemas del paradigma sistemas multiagentes, tales como el control de semáforos.

El uso de agentes permite distribuir las tareas en la red, lo que permitiría la construcción de sistemas inteligentes de bajo costo. Actualmente la ejecución de todas las operaciones implicadas en un sistema de tráfico urbano, se realiza en forma integral, pero los SMA permiten asignar tareas específicas a determinados agentes.

- En [Wang, 2005] se definen agentes por defecto que aseguran la operación básica y rendimiento del controlador de tráfico. Se pueden construir estrategias de control fijas para construir los agentes por defecto. El host tiene agentes de control en ejecución los cuales contienen un módulo encargado de tomar las decisiones (figura 1.3).



Figura 1.3. Módulo de decisión del agente de control.

Se distinguen tres niveles en la estructura del sistema propuesto por Wang:

- Nivel de Organización.
 - Se planea que va a hacer cada agente, se le proporcionan bases de datos, protocolos, algoritmos para la toma de decisiones.
- Nivel de Coordinación.

- Es la interfase entre el nivel de organización y el nivel de ejecución.
- Generalmente consiste de un despachador y coordinadores; el despachador recibe los agentes de control del nivel de organización y asigna los agentes de control al coordinador apropiado a través de una WAN.

Nivel de organización → Dispatcher → Coordinador.

Nivel de Ejecución.

- Es el nivel encargado de desplegar, guardar y ejecutar los agentes de control.

Podemos concluir que el enfoque SMA, parece ser un esquema muy bueno para controlar y coordinar el tráfico urbano. Las principales ventajas que ofrece trabajar con sistemas multiagentes, son:

- i. Facilidad de comunicación entre agentes situados en diferentes host.
- ii. Repartición de las tareas entre los diferentes agentes.
- iii. Interacción de los agentes con su ambiente.
- iv. Cooperación entre los agentes para obtener una solución factible.
- v. Coordinación entre un grupo de agentes involucrados y no entre todos los que estén en el sistema en un momento dado.

Entre las limitaciones que encontramos al trabajar con sistemas multiagentes, encontramos que.

- i. Es un paradigma relativamente nuevo.
- ii. Una gran cantidad de agentes en el sistema reduce el rendimiento del mismo, debido al número de mensajes que se envían y a los recursos que consume cada agente.

1.3.2. Redes de Petri

Las redes de Petri son un formalismo eficiente para el modelado de sistemas de eventos discretos (SED). El tráfico urbano puede ser modelado como un SED, por lo cual

las redes de Petri son una herramienta promisorio para describir sistemas de control de tráfico urbano.

Control de tráfico urbano usando redes de Petri

- En [Cetin, 2004] se propone un modelo basado en redes de Petri para el control de semáforos. El modelo de control de señales puede separarse en dos categorías: 1) determinar que secuencia deben seguir las señales para optimizar el rendimiento del sistema y 2) como implementar la lógica de control de señales.

En la figura 1.4 se describe la lógica para el despliegado de rojo, verde y ámbar del semáforo para un desplazamiento dado. Si suponemos que una marca ha sido depositada en el lugar GG, se habilita la transición (t1); el disparo de la transición t1 deposita marcas en los lugares verde inicial (lugar 1), despliega verde (DG), y en desplazamiento (M). La marca en el lugar M permite que los vehículos se muevan, y la marca en DG finalizará la luz verde una vez que una marca sea depositada en el lugar activa rojo (GR). Cuando se dispara la transición (Min) se deposita una marca en el lugar verde extendido (EG). Esta marca se mantiene en ese lugar hasta que una de tres condiciones sea alcanzada: la transición (Act) se habilita de acuerdo a la demanda si se opera en un esquema adaptable; se habilita la transición FO cuando se opera en un modo coordinado; o se habilita la transición (Max) si se alcanza el tiempo máximo de verde para esa fase. Cuando alguno de estos tres casos ocurre la marca en EG es removida y una marca es depositada en termina verde (RG).

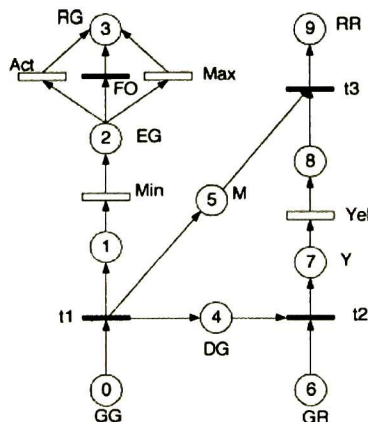


Figura 1.4. Red de Petri temporizada.

- En [DiFebrero, 2004] se proponen redes de Petri Híbridas (HPN) para modelar una red de tráfico urbano. Modelar las redes de tráfico urbano como HPN nos permite tomar ventaja del modelado del flujo de tráfico urbano como fluidos, tomando en cuenta el dinamismo de las luces de tráfico (event- driven) y su influencia en el dinamismo del flujo, los siguientes problemas de control pueden ser resueltos.

1. Optimización del plan de los semáforos.
2. Rutas dinámicas.
3. Control de vehículos especiales.

La figura 1.5 consiste de una intersección particular I_h con tres calles de entrada R1, R2 y R3, tres calles de salida R4, R5 y R6, y las calles adyacentes. Donde f_j^{in} representa el flujo de vehículos que se aproximan a la intersección, f_j representa el flujo de vehículos que cruzan la intersección, y f_j^{out} representa el flujo de vehículos que dejan la intersección; q_j denota la cola de vehículos en espera de cruzar la intersección.

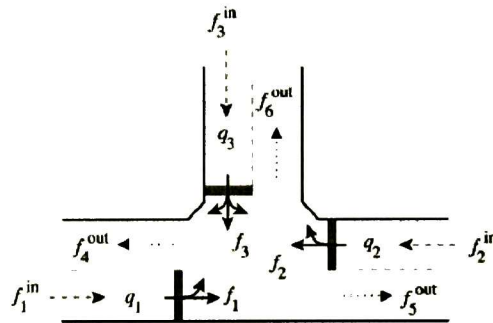


Figura 1.5. Intersección de la red de tráfico urbano.

El modelo de la intersección en la figura 1.5 es representado en la figura 1.6. Está red está compuesta por tres redes de Petri continuas que modelan el flujo de vehículos que entran a la intersección y las colas, y tres redes de Petri que modelan el flujo de vehículos que salen de la intersección. Además una red de Petri discreta representa los semáforos. El marcado en la parte continua de la red representa el número de vehículos que hay en cada lugar; los lugares p_e^i y p_d^i , representan respectivamente que los vehículos pueden o no pueden fluir por la calle R_i . En el modelo discreto, el lugar p_k representa que la fase k está

habilitada, el lugar p_k^B representa el período de verde para la fase k y el lugar p_k^A representa el período de ámbar para la fase k .

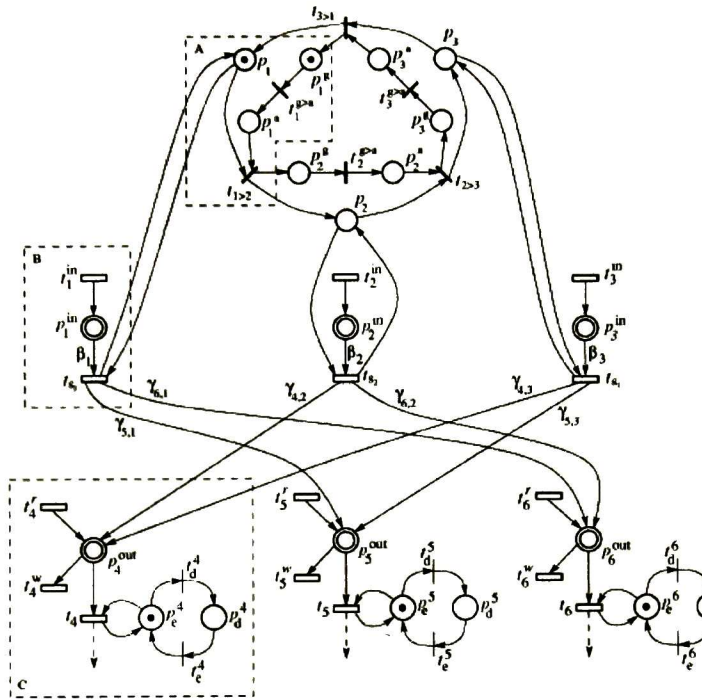


Figura 1.6. Modelo de sistema de tráfico con RP híbridas.

- En [Mancinelli ,2005] se propone la construcción de una red de Petri que describe la sincronización entre todas las intersecciones de una ciudad pequeña; el modelo se basa en la velocidad de los autos, y una descripción simple de los flujos de los carros.

En la figura 1.7 se ilustra una intersección y el modelo de red de Petri asociado a la intersección en la red de tráfico modelada en este trabajo. Se denota por $x_0(t)$ y $x_1(t)$ el número total de luces verdes que han ocurrido en cada una de las fases hasta el tiempo t . La duración de cada una de las dos fases verdes se denota por τ y v respectivamente. Se define como α el número de autos que doblan en la dirección disponible en la intersección. Se denota como $u_0(t)$ y $u_1(t)$ el total de carros que llegan a la intersección hasta el tiempo t , y se denota $y_0(t)$ y $y_1(t)$ el total de carros que han dejado la intersección hasta el tiempo t .

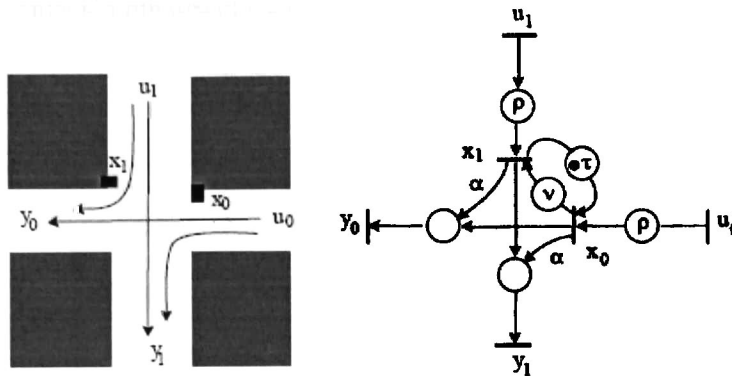


Figura 1.7. Modelo para una intersección en una red de tráfico.

- En [Szyrka M., 2004] se utilizan RTCP-nets (Real Time Coloured Petri Nets) para control de semáforos. Las RTCP-nets pueden ser vistas como una subclase de las redes de Petri coloreadas temporizadas.

Esta nueva subclase fue definida para alcanzar los siguientes objetivos.

- Acelerar y facilitar el dibujar los modelos de redes de Petri.
- Equipar las redes de Petri con la capacidad del modelado directo de elementos, tales como prioridades y tiempo que son comunes en programación concurrente.
- Facilitar el análisis formal de las propiedades de las redes de Petri temporizadas de alto nivel – Grafos de alcanzabilidad finitos en la mayoría de los casos.

El conjunto de lugares P esta dividido en dos subconjuntos: PM , el conjunto de lugares principales y PA , el conjunto de lugares auxiliares. Los lugares principales representan las partes distinguibles (elementos) del sistema modelado.

El conjunto T de todas las transiciones también se divide en dos subconjuntos: TM , (transiciones principales) y TA (transiciones auxiliares). Las transiciones principales representan acciones del sistema modelado.

- En [Carbone, 2004] se propone una aproximación modular, en la cual las áreas urbanas a ser controladas son modeladas como redes de Petri temporizadas interconectadas. En ese

caso una representación microscópica de tráfico urbano puede ser usada por algunas partes de la red.

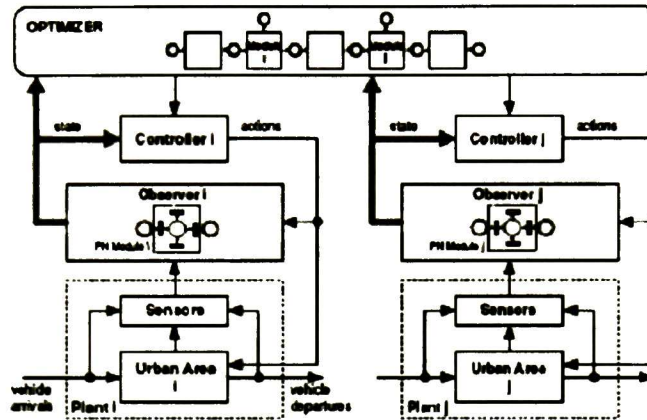


Figura 1.8. Esquema general de un sistema de control de tráfico urbano.

Cada área urbana (figura 1.8) a ser controlada junto con sus sensores puede ser vista como la planta. Las señales de sensor acerca de las llegadas y salidas de los vehículos mas la información intermedia son considerados como eventos de entrada para el subsistema observador. El observador contiene un modelo detallado de redes de Petri temporizadas del área controlada.

El modelo propuesto puede cambiar sus parámetros de acuerdo a la planta, esto es, puede cambiar el tiempo de sus transiciones temporizadas de acuerdo a las condiciones de tráfico. La modularidad está hecha para poder considerar como un modulo cada componente elemental del área urbana y se pueden interconectar por medio de lugares.

- En [Júlvez, 2001] se introduce un modelo de tráfico urbano basado en redes de Petri continuas/híbridas. La calle está dividida en secciones, cada sección es modelada como una red de Petri continua. Estas secciones son unidas por medio de una transición que interconecta dos secciones. Los semáforos son modelados por lugares y transiciones discretos.

La figura 1.9 modela cuatro secciones y una intersección en la cual el tráfico es regulado por un semáforo (figura 1.10). La red es una red de Petri híbrida dado que incluye lugares y transiciones discretos y continuos.

El modelo del semáforo es representado en la figura 1.10. Tiene seis fases representado por cada uno de los lugares de la red. Una fase está activa cuando su lugar correspondiente está marcado. Como solo una fase puede estar activa solo existe una marca en la red. El significado de las fases es: ggg: los carros de L1 cruzan; ggr: para el tráfico de L1; grr: los carros de L2 empiezan a cruzar; rrr: los carros de L2 cruzan; rrg: para el tráfico de L2; rgg: los carros de L1 empiezan a cruzar.

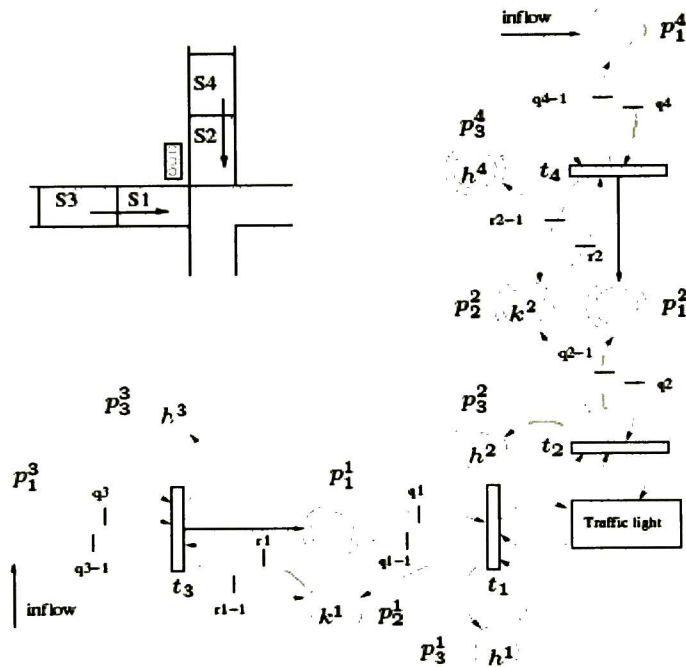


Figura 1.9. Modelo continuo para el sistema de tráfico.

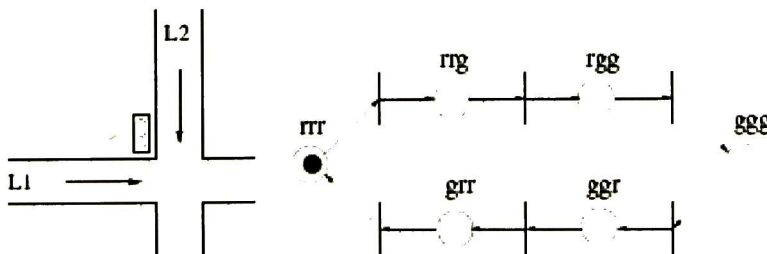


Figura 1.10 Modelo discreto para el control de semáforos.

1.3.3. Lógica difusa

En sistemas de control de tráfico urbano se ha visto que la lógica difusa tiene un rendimiento eficiente para controlar las luces y las fases de los semáforos. En esta sección se abordan algunas de las técnicas empleadas para control de tráfico urbano utilizando el paradigma de la lógica difusa, así como también, las ventajas y las desventajas de utilizar esta herramienta.

La lógica difusa [Kaehler, 1993] es una herramienta matemática que se usa para establecer una metodología de control cuya implementación puede ser simple o pequeña pero también puede ser grande y distribuida. Puede ser implementada en hardware, software o una combinación de ambos.

La lógica difusa provee una forma fácil de obtener una conclusión basada en información imprecisa, información vaga, información incompleta, información con ruido o información ambigua. La lógica difusa trabaja en la forma en que una persona toma las decisiones.

- En [Sanal, 1999] se considera una simple intersección con dos calles. El sistema tiene los siguientes subsistemas(figura 1.11).
 - Unidades de colección de información: Los sensores obtienen los parámetros de control.
 - Sistema de control. Propone las estrategias de control en la intersección.
 - Sistema de información local: Estructura de la red de tráfico, dispositivos electrónicos que evalúa las mediciones del detector y las luces de los semáforos.
 - Sistema de desplegado de información: Semáforos controlables.

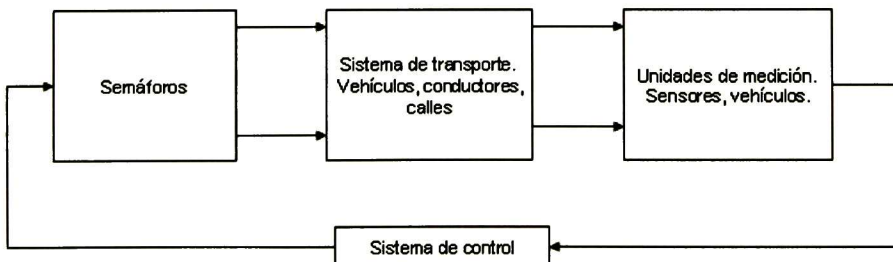


Figura 1.11. Diagrama de un sistema de control de tráfico urbano con lógica difusa.

- En [Sazi, 2002] se presenta un modelo de control el cual es desarrollado para controladores multifases en intersecciones aisladas. Contiene un controlador de lógica difusa (figura 1.12) encargado de asignar los tiempos y otro encargado de secuenciar las fases.



Figura 1.12 Controlador de tráfico urbano.

Parámetros de entrada del controlador de tiempo.

- Cola más larga en la señal roja. {very few, few, medium, many}.
- Número de vehículos que pasan en señal verde. {very few, few, medium, many}.
- Tiempo ocioso de la señal verde {very few, few, medium, many}.

Parámetro de salida del controlador de tiempo.

- Decisión de cambiar el tiempo asignado a las luces. {more decrease, decrease, no change, increase, more increase}

Parámetros de entrada del controlador de fases.

- Cola más larga en la señal roja. {very few, few, medium, many}.
- Cola más larga próxima fase. { few, medium, many}.
- Tiempo de cambio de la cola más larga en señal roja {few,medium,many}

Parámetro de salida del controlador de fases.

- Cambiar la decisión de la próxima fase. {No change, change}.

Hemos observado que esta metodología resulta muy eficaz para el control de tráfico urbano de una intersección aislada. Dada la naturaleza de la lógica difusa, esta metodología de control para tráfico urbano permite tomar decisiones de acuerdo a como lo haría un operador humano ante una situación similar, y tomando datos que pudieran en un momento dado tener cierto ruido o datos incompletos.

A pesar de que el enfoque de lógica difusa para control de tráfico urbano resulta ser muy eficiente, en todos los artículos que se han revisado, solo se controlan intersecciones aisladas. Esto es bueno si quiere controlarse una sola intersección, pero para poder aplicar coordinación, se necesitaría alguna otra herramienta como los sistemas multiagentes para poder lograr sincronizar varias intersecciones.

1.3.4. Otros enfoques

Se han analizado tres enfoques principales de control de tráfico urbano. Existen algunas aproximaciones para control de tráfico urbano que pueden ser muy buenas y de las cuales no existe mucha literatura.

En este apartado analizamos tres de estos enfoques. Primeramente se analiza control de tráfico urbano por medio de teoría de grafos, en este enfoque se utiliza programación dinámica para obtener una ruta optima. También analizamos control de tráfico urbano por medio de control adaptativo, en este tópico se analiza aprendizaje por refuerzo y se ve como la duración de las luces se va adaptando de acuerdo a la demanda. Por último analizamos control de semáforos utilizando programación lineal para alcanzar el objetivo de controlar el tráfico urbano.

Control de tráfico urbano usando teoría de grafos

En [Riedel, 1993] se presenta un enfoque para controlar el semáforo de una intersección de tráfico urbano utilizando teoría de grafos. Para cada semáforo se distinguen tres fases. Green (G), Red (R) y Intermedia (Ámbar). A diferencia de las fases G y R la intermedia tiene una duración fija.

Para obtener una estrategia de control óptima, se utiliza programación dinámica. Cada estado es un nodo K en el árbol de programación dinámica (figura 1.13). El nodo inicial es que todas las luces estén en rojo y que las colas estén vacías.

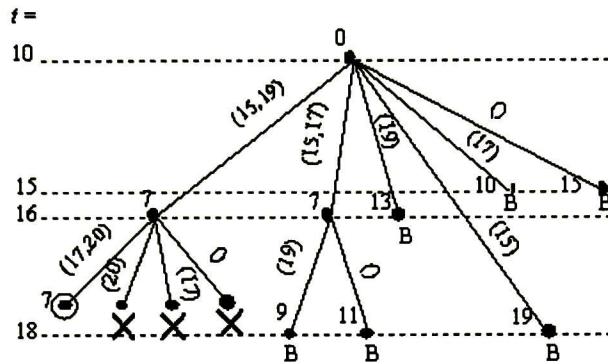


Figura 1.13. Árbol de programación dinámica

Se deben cumplir las siguientes condiciones.

- Tiene que ser definida una regla de generación $g(K)$. La función g determina todos los nodos los cuales pueden seguir directamente después de K . $g(K)=\{g_1(K),\dots,g_n(K)\}$.

- Debe existir una estrategia G de generación que indique en que punto de tiempo $g(K)$ debe ser aplicado, esto es cuando los siguientes nodos deben ser generados.

Debe haber una función de costo $c(K)$ que describe el costo para el nodo K . La función de costo debe ser aditiva, los costos de un nodo siguiente resulta de la adición del costo del ultimo nodo con algún incremento. $c(g_i(K)) = c(K) + \Delta K \rightarrow g_i(K)c$.

Debe haber una condición de comparación V que determina si dos nodos K_A y K_B son comparables o no $(K_A, K_B) \in V$ o no.

Finalmente debe haber una regla de comparación " $=$ " para estados comparables K_A y K_B , como usuales: es $c(K_A)=c(K_B)$?

El algoritmo es implementado como búsqueda en profundidad siguiendo un camino. Una ventaja de la búsqueda en profundidad es que podemos obtener un límite muy rápido para el objetivo, y aplicar luego divide y vencerás además de la búsqueda en profundidad.

El enfoque de teoría de grafos nos permite reducir la complejidad computacional al utilizar programación dinámica combinada con divide y vencerás. De esta manera se logra un control eficiente de tráfico urbano.

Sin embargo, este enfoque cae en la posibilidad de la explosión de estados lo cual implica que la complejidad computacional se aumenta y ya no se tendría un control eficiente.

Control de tráfico urbano usando control adaptativo

En este enfoque [Sivasankaran, 2000] se incrustan microcontroladores en las señales de tráfico que ayudan a analizar el tráfico en una intersección. Sin embargo, es insuficiente con esta técnica debido a que los microcontroladores tienen memoria limitada y solo puede hacerse un análisis de pocos datos.

Este enfoque nos ayuda a encontrar una solución factible donde los microcontroladores de una región están conectados a una computadora principal que puede almacenar grandes cantidad de datos y realizar un análisis mas sofisticado a través de técnicas de modelado de procesos estocásticos. El plan se divide en dos fases.

Fase1:

- Puede tomarse en cuenta una pequeña área (5 – 10 regiones)
- Seleccionar una buena técnica de análisis para construir un modelo predictivo.
Diseñar un software portable que pueda enviar una subtask a algún host en Internet que este dispuesto a participar.
Varias herramientas de simulación pueden ser utilizadas para probar el rendimiento del sistema.

Fase2:

- Extender el sistema a múltiples regiones.

Control de tráfico urbano usando programación lineal

Se distinguen cinco formas básicas de control de intersecciones:

- i. Control implícito o ausencia de control.
- ii. Intersección con señal de prioridad.
- iii. Intersección semaforizada.
- iv. Intersección giratoria o rotonda.
- v. Intersección a distinto nivel.

La selección del tipo de control más adecuado puede hacerse de dos maneras básicas: realizar un diseño preliminar de dos o tres alternativas de control; calcular los costos y beneficios de operación de cada alternativa, y elegir la alternativa más eficiente a partir de un análisis costo beneficio. Utilizar normas que justifiquen cada tipo de control en base a un análisis aproximado beneficio costo en el cual se sustentan.

Los objetivos tras la instalación de semáforos son reducir y prevenir accidentes, reducir demoras a vehículos y peatones y reducir consumos de combustibles y emisión de contaminantes.

La programación de semáforos consiste en determinar, de acuerdo a la demanda por cada rama de la intersección, los siguientes aspectos:

- Diseño de fases: conjunto de movimientos no conflictivos que tienen derecho a vía simultáneamente.
- Diseño de entreverdes: la determinación de tiempos de amarillo rojo - rojo y verdes mínimos para cada fase.

Determinación del tiempo de ciclo: duración mínima, máxima y óptima de la secuencia completa de fases.

Determinación de repartos: definición de los tiempos de verde a cada fase.

- Determinación de desfases: diferencia entre los inicios de tiempos de verde para fases homólogas de semáforos adyacentes en la red.

Se aborda la programación de semáforos aislados y se aborda el problema de determinar el ciclo y repartos de un semáforo aislado usando programación lineal. Para resolver el problema se pueden especificar dos funciones objetivo básicas:

- Minimizar la demora total en la intersección.
- Maximizar la capacidad de la intersección.

1.4. Conclusiones

En este capítulo se definió un sistema de control de tráfico urbano; se describe la forma en que los sistemas de control son modelados. Se presentaron también diferentes enfoques para el control de tráfico urbano.

Primeramente describimos los sistemas multiagentes como enfoque de control de tráfico urbano. Se mencionaron las ventajas y desventajas de utilizar este enfoque así como algunas metodologías existentes.

El control de tráfico puede verse como un sistema de eventos discretos. Las redes de Petri nos permiten analizar un modelo de acuerdo a sus propiedades, tal es el caso de el análisis de invariantes, bloqueos, vivacidad, acotamiento entre otras. Las redes de Petri temporizadas dan la posibilidad de poner tiempos a cada una de las diferentes fases de una política de control.

El modelado de redes de Petri continuas resulta ser una buena opción para el control de tráfico urbano pues permite modelar bien el comportamiento del sistema de tráfico. Se propone redes de Petri discretas para modelar los semáforos en una intersección.

Se analizó la lógica difusa como herramienta para el control de tráfico urbano; se mencionaron los puntos fuertes de este enfoque para control de tráfico urbano; sin embargo, las desventajas que presenta la hacen una pobre herramienta para coordinación de tráfico. También se mencionaron otros enfoques de control como la teoría de grafos, la programación lineal y sistemas adaptativos. En todos ellos vimos que son eficientes en algunos casos para control, pero en otros casos resultan poco eficientes.

Por tanto podemos concluir que para nuestro sistema lo más adecuado es emplear una arquitectura multiagente cuyo modelo de intersección este representado por redes de Petri temporizadas, dado que es lo que mejor se adaptan a nuestra propuesta y al analizar los diferentes enfoques hemos concluido que tomando esta metodología podemos lograr obtener los resultados esperados.

Capítulo 2

Requerimientos para el desarrollo de un sistema de control de tráfico.

Resumen: En este capítulo se presenta una descripción general del simulador de tráfico urbano y se analiza la necesidad de integrar un sistema de control de tráfico. Se plantean los requerimientos específicos para el desarrollo del modelo de control de tráfico urbano.

2.1. Simulador

El simulador actualmente en desarrollo es un modelo de simulación microscópico y utiliza la técnica de evento siguiente en la cual se tiene una lista de eventos futuros donde se almacenan los eventos que habrán de ocurrir en la simulación. El modelo se examina y actualiza solamente cuando se sabe que va a ocurrir un cambio en el estado del sistema. Estos cambios son llamados generalmente “eventos” ya que el tiempo se mueve de evento a evento, el enfoque es llamado la técnica del evento siguiente. Con esta técnica, el algoritmo encargado de avanzar el tiempo de simulación, avanza el tiempo hasta la hora en la que va a ocurrir el evento significativo siguiente. La implementación del simulador es realizada en el lenguaje de programación Java.

Descripción orientada a eventos del simulador

El simulador de tráfico urbano es especificado como un sistema de eventos discretos (DES: discrete-event system); se deben tener bien definidas las interacciones entre los componentes del sistema y los cambios en el estado del sistema deben llevarse a cabo a intervalos de tiempo predefinidos.

Cada evento modela un cambio del estado del sistema y es responsable de generar otros eventos que dependen de este evento. Cada evento tiene asignado un tiempo de ocurrencia y algunas acciones a ser ejecutadas cuando el evento ocurra. Los eventos generados son inscritos en la lista de eventos futuros (FEL).

En la figura 2.1 [Neri ,2005] se muestra la especificación de un sistema de tráfico urbano como un sistema de eventos discretos.

Estado del Sistema: L(t), LQ1(t), LQ2(t), MT(t)	Eventos:	Entidades:	Conjuntos:
L(t) = Núm. de vehiculos en calle	(AE,t,car _i) car _i arriba a la calle en el instante t	Vehiculo , (car _i ,t) vehiculo i arriba en el instante t.	cars, todos los vehiculos en la calle.
LQ1(t) = Núm. vehiculos en carril 1.	(SE,t, car _i) , car _i para en el instante t.		carPosition, pos vehiculos en calle.
LQ2(t) = Núm. vehiculos en carril 2.	(CE,t, car _i) , car _i cruza la calle en el instante t.		carline_1, vehiculos en el carril 1.
MT(t) = Prom. tiempo cruzar la calle.	(CLE,t, car _i) , car _i cambia de carril en el instante t.		
	(SSE,t, car _i) , car _i inicia avance en el instante t.		
	(CLEE,t, car _i) ,car _i Continua el avance en el instante t.		
	(LCE,t,light) , light, cambio el estado del semáforo en el inst t.		

Figura. 2.1. Elementos del simulador de tráfico urbano

2.1.1. Integración del sistema de control de tráfico urbano

El simulador se complementa al integrarle un sistema de control de tráfico urbano; de esta manera se puede comprobar mediante la realización de pruebas la adaptabilidad del controlador a las condiciones de tráfico. En la siguiente figura se ilustra la integración del módulo de control al simulador. El controlador debe proponer cambios de luz a la máquina de simulación en forma de eventos que puedan ser entendidos por la FEL.

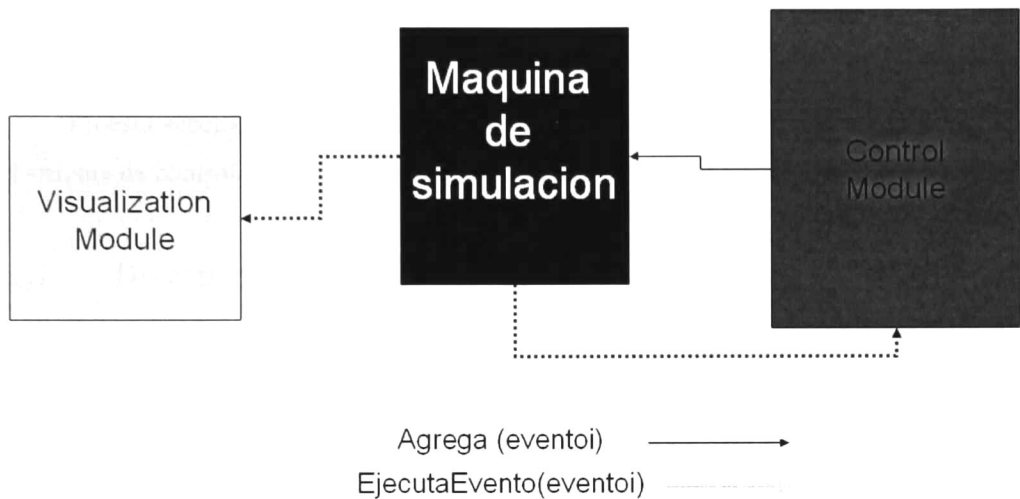


Figura 2.2 Integración del controlador al simulador

Como resultado de la implementación del sistema de control se espera obtener los siguientes resultados.

- Reducir demoras a vehículos y peatones.
 - i. Al existir un control de semáforos se espera reducir el tiempo de espera de los usuarios.
- Incrementar la capacidad de la infraestructura de red de tráfico.
 - i. El hecho de que la red de tráfico se aproveche a su máxima capacidad, quiere decir que el mayor número de vehículos utilicen la red sin que exista congestión.
- Reducir y prevenir accidentes.
 - i. La personalidad del usuario tiene mucho que ver con este objetivo, dado que un usuario con personalidad agresiva puede no respetar la señal de tráfico.
 - ii. Se debe tener presente que si se producen demoras excesivas y no justificadas por la instalación de un semáforo, estos se tienden a dejar de respetar.
- Reducir consumos de combustibles y emisión de contaminantes.

- Generar una herramienta de control de tráfico urbano que interactúe con el simulador de tráfico urbano.

2.2. Especificación de requerimientos

En esta sección presentamos los requerimientos funcionales para la implementación del sistema de control de tráfico urbano y la adaptación del mismo al simulador.

2.2.1. Descripción del problema.

Debemos demostrar que el sistema de control se adapta a la demanda de tráfico, para ello se debe desarrollar una cama de pruebas de los diferentes grupos de control. Las pruebas deben realizarse en diferentes escenarios, y es necesario efectuar las simulaciones con los diferentes esquemas de control para observar el comportamiento del controlador.

2.2.2. Evolución del sistema.

En esta sección definimos las suposiciones fundamentales en las cuales se basa el sistema y describe los cambios a los que debemos anticiparnos.

- La máquina de simulación será capaz de simular el comportamiento real de una red de tráfico, incluyendo las políticas de control establecidas por el agente de control. Actualmente nuestro modelo esta compuesto por una red de carreteras con cuatro intersecciones.
- Existe un módulo de visualización encargado de proveer un ambiente gráfico a lo que está ocurriendo en el sistema de tráfico urbano: llegada de vehículos, cruce de automóviles, cambios de luz.
- Existe un medio que brinda una eficiente conectividad entre los diferentes módulos involucrados en la simulación.

2.2.3. Definición de requerimientos.

En este apartado nos encargamos de definir en términos comprensibles los requerimientos del sistema de control de tráfico urbano.

Requerimientos de Software.

Se han establecido ciertos requerimientos de Software enfocados principalmente a la capacidad que debe tener el sistema de ser portable y tomando en cuenta la conectividad entre los módulos.

- Se ha establecido como requerimiento que el sistema de control tráfico urbano trabaje en cualquier plataforma, este requerimiento implica que las herramientas para desarrollar la metodología puedan de igual manera funcionar bajo diferentes sistemas operativos. Específicamente este requerimiento sugiere que el lenguaje de programación soporte varias plataformas, esto es, que funcione bajo cualquier sistema operativo.
- Dado que los módulos implicados en la simulación deben ser distribuidos, esto es que trabajen desde diferentes plataformas que pueden ser homogéneas o heterogéneas, se ha definido como requerimiento que se provea un medio que permita la conectividad entre los diferentes módulos implicados en la simulación.
- Debido a que en el desarrollo del sistema de control de tráfico urbano es necesario definir una base de datos la cual sirva para definir los recursos de la calle, tales como el estado de la calle, número de intersecciones en la red, fases en la intersección, por mencionar solo algunas; se hace necesario el requerimiento de establecer una conexión entre la base de datos y el sistema de control, por lo cual se hace uso de un sistema manejador de bases de datos.

Requerimientos de la máquina de simulación.

Deben tomarse en cuenta para el desarrollo del módulo de control los siguientes requerimientos para el buen desempeño de la simulación.

- Los cambios de luces de la red de tráfico deben ser provistos por el módulo de control hacia la máquina de simulación en forma de eventos.
- El agente de control asigna tiempos para dar preferencia de luz verde a algunas fases, esta metodología funciona bien cuando se trata de un esquema de control fijo; sin embargo, al realizar los cambios dinámicamente existe la posibilidad de que el controlador programe un evento de cambio de luz en un tiempo t , siendo que ya se tengan programados en la FEL eventos a realizarse en un tiempo t' donde

$t' > t$. En la figura 2.3 podemos observar una situación de conflicto en la que el controlador asigna un cambio de luz para el tiempo t , siendo que la FEL tienen eventos programados en instantes de tiempo mayores a t . Debido a esto el agente de control debe tomar en cuenta que los eventos de cambio de luz que sean programados respeten el orden de los eventos establecidos en la FEL.

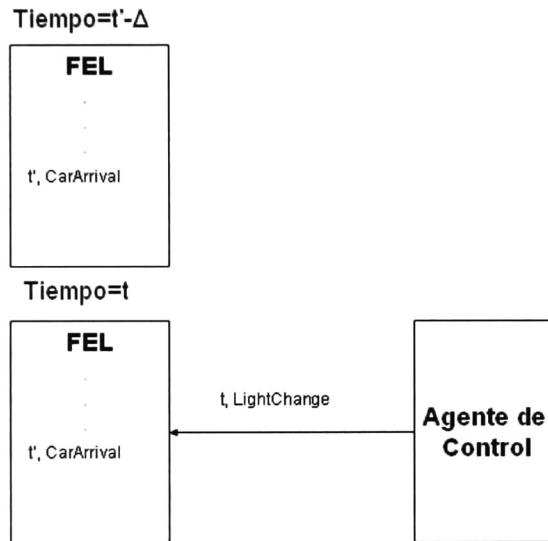


Figura 2.3. Conflicto al enviar evento de cambio de luz para el tiempo $t < t'$

Requerimientos de conectividad.

En un sistema real de tráfico urbano es común que exista un flujo variable de vehículos y existen muchos recursos a ser tomados en cuenta en el control y administración de dicho sistema. El simulador tiene como objetivo representar un sistema de tráfico de tal forma que pueda simular el comportamiento del tráfico en forma real. Dado que un solo procesador resulta insuficiente para representar tanto el flujo de tráfico como el control, es necesario asignar a uno o más procesadores la tarea de simular el tráfico de una red y a otro procesador o más procesadores la tarea de controlar el flujo de vehículos en la red. Con la finalidad de que los módulos puedan comunicarse entre sí y las distintas tareas se realicen en paralelo se ha establecido el requerimiento de que se utilice un Middleware (ver figura 2.4) para la comunicación entre los módulos implicados en la simulación.

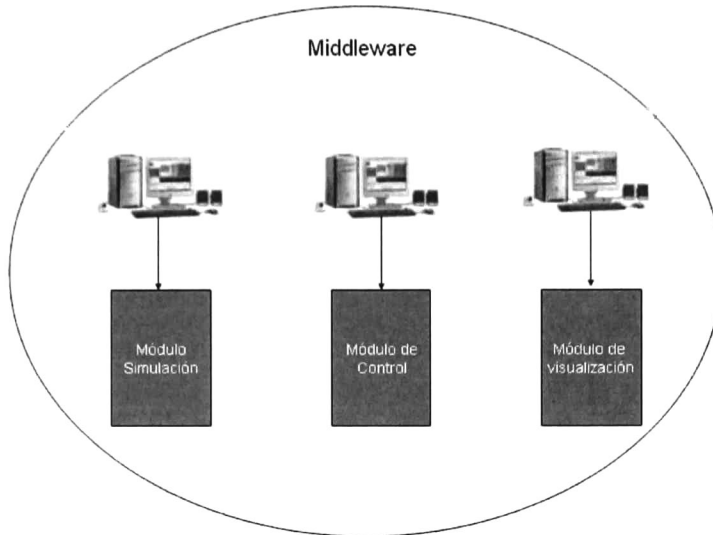


Figura 2.4. Ambiente distribuido del simulador.

Requerimientos de la metodología de control.

Se han definido los requerimientos que establecen la forma en que el módulo de control proporciona los eventos de cambios de luz al simulador.

- El módulo de control debe definir una metodología de control. Esta metodología de control debe proporcionar cualquiera de las siguientes estrategias de control de acuerdo a las condiciones de tráfico observadas.
 - Estrategia de control fija, también conocida como pretimed.
 - Estrategia de control adaptativa, es decir, cambia dinámicamente de acuerdo a la demanda de tráfico.
 - Reemplazo de políticas de control de acuerdo a la demanda de tráfico.
 - Coordinación entre controladores.

2.3. Conclusiones.

En este capítulo se presentaron los requerimientos para el desarrollo del sistema de control de tráfico urbano. La primera parte de este capítulo estuvo dedicada a presentar el trabajo ya realizado en cuanto al tema de tráfico urbano. Presentamos las características del simulador con el que contamos así como de la metodología en la que está basado. Se establecieron los requerimientos generales del sistema, en los que describimos el problema de control de tráfico urbano y se realizó un trabajo de investigación, con la finalidad de obtener todos los requerimientos funcionales para el control de tráfico. Se realizó una especificación de requerimientos en la que entre otras cosas definimos los límites de la implementación y la definición de cada uno de los requerimientos necesarios para el buen funcionamiento del controlador.

En el siguiente capítulo se define una arquitectura de agentes para el control de tráfico urbano. Se describe la manera en que el agente de control trabaja para obtener las variables de estado necesarias que definen las condiciones de tráfico y poder realizar los ajustes necesarios.

Capítulo 3

Esquema multiagente para el control de tráfico urbano.

Resumen: En este capítulo presentamos nuestro esquema multiagente para el control de tráfico urbano. En nuestro esquema cada intersección es controlada por un agente de control; se describe en detalle cada componente del agente de control y la forma en que los agentes interactúan con el ambiente y entre ellos mismos.

3.1. Introducción

En este capítulo se propone una arquitectura de agentes para el control de tráfico urbano. En nuestro enfoque el área urbana, está compuesta por intersecciones y calles; cada intersección es controlada por un agente de control y el modelo de la misma es representado como una red de Petri temporizada. Cada agente de control posee las siguientes características:

Interactúa con el ambiente. El agente puede conocer en cualquier instante de tiempo el estado en el que se encuentra el ambiente, y tomar las acciones que considere necesarias para alcanzar sus objetivos.

- *Toma decisiones:* El agente de control decide que estrategia de control aplicar de acuerdo a las condiciones de tráfico observadas.
- *Proactivo.* Toma la iniciativa de aplicar una estrategia de control de acuerdo a las condiciones de tráfico dadas.
- *Comunicación con otros agentes.* El agente de control puede comunicarse con los demás agentes en la estructura de la red de tráfico.

Cada agente tiene capacidad insuficiente para controlar la red de tráfico por si solo, por ese motivo es necesario que los agentes interactúen unos con otros para alcanzar sus objetivos.

- Los agentes proveen un control descentralizado ya que cada agente está encargado de realizar el control sobre la intersección a la cual está asignado.
- Actualmente el agente de control contiene una base de conocimientos, la cual está diseñada para aprender por medio de la actualización de sus reglas de decisión.

3.2. Agente de Control

El agente de control debe tomar en cuenta el problema de la saturación de congestión. Una calle saturada impide que el tráfico de vehículos pueda fluir de una calle a otra aunque esta tenga derecho de paso (figura 3.1). El sistema que proponemos toma en cuenta esta situación y evita que ocurra la congestión primeramente por medio de un control local y si esto no resulta, se coordina con agentes asignados en intersecciones adyacentes

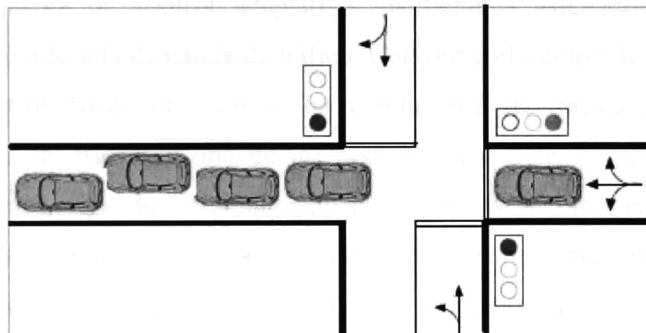


Figura 3.1. Saturación en una calle.

3.2.1. Escenario del agente de control

El agente de control debe proponer las políticas de control de acuerdo a las condiciones de tráfico observadas en la intersección; por lo tanto el ambiente en el que un agente de control se desenvuelve es en la intersección a la cual está asignado. La manera en que el agente de control se adapta a la demanda de tráfico responde a tres posibles esquemas.

- *Esquema de control fijo.*

En el esquema de control fijo el agente asigna los tiempos asignados a las fases de acuerdo a tiempos predeterminados. El intervalo de tiempo asignado a cada fase en el esquema de control fijo se basa en patrones históricos de tráfico. Inicialmente el agente de control adopta un esquema de control fijo con la finalidad de que la simulación empiece su ejecución. Una vez que el simulador ha iniciado su ejecución el agente de control decide si continua con un esquema de control fijo o bien decide cambiar la metodología de control.

Cuando las condiciones de tráfico observadas son predecibles durante un periodo de tiempo significativo, el agente de control considera que la mejor política es la que asigna a las fases tiempos fijos.

- *Esquema de control adaptativo.*

En el esquema de control adaptativo los tiempos asignados a las fases son establecidos de acuerdo a la demanda de tráfico. Conforme el tiempo de simulación avanza el agente de control puede observar si las condiciones de tráfico son adversas para determinada calle; en este caso una de las opciones que tiene el agente de control es modificar los tiempos asignados originalmente a las fases involucradas en esa intersección.

Cuando las condiciones de tráfico observadas son impredecibles o tienen un comportamiento inconstante, el agente de control puede deducir que una política de control adecuada para este tipo de comportamiento de tráfico es un control adaptativo en donde los tiempos asignados a las fases se modifiquen de acuerdo a la demanda de tráfico.

Reemplazo de políticas de control.

En esta metodología de control el agente decide cambiar la secuencia de las fases responsables del control en una intersección. Cuando durante el tiempo de simulación se obtienen condiciones de tráfico adversas, tales como una saturación excesiva de la congestión, y el agente de control considera que la modificación de los tiempos asignados a las fases es insuficiente para preservar las condiciones optimas para el flujo de tráfico en

esa intersección; entonces el agente puede realizar modificaciones a la secuencia de las fases, tales como incluir o eliminar vueltas a la izquierda, incrementar o disminuir el número de calles que tienen derecho de paso simultáneamente, entre otras. Dar paso a peatones cuando estos lo soliciten requiere modificar la secuencia de las fases, asimismo dar paso a vehículos de prioridad previa solicitud de los mismos requiere modificar la secuencia de las fases.

3.2.2. Interacción del agente de control con la máquina de simulación

El agente de control y la máquina de simulación interactúan a través de una interfaz. Los eventos de cambio de luz son provistos por el agente de control; este se conecta a la máquina de simulación por medio de la interfaz, y programa eventos de cambio de luz a la FEL (Future Event List). Por medio de esta misma conexión a través de la interfaz el agente de control obtiene los datos correspondientes al estado de las calles y de las intersecciones. En la figura 3.2 observamos como se realiza la comunicación entre el agente de control y la máquina de simulación. Estos eventos se rigen por un esquema de control fijo para el cambio de luces. Durante el transcurso del tiempo de simulación, la máquina de simulación envía el historial de los datos que van ocurriendo. De esta manera el agente de control puede tomar la decisión de cambiar los tiempos asignados a cada fase de acuerdo a la demanda de tráfico, esto es al flujo y densidad observados en las calles, o bien, puede cambiar la política de control cambiando las fases implicadas en la red de tráfico. La arquitectura de control que proponemos posee la característica de que una intersección se coordina con otras intersecciones. Esto significa que si se observa congestión o algún incidente en alguna calle perteneciente a una intersección, el agente encargado de controlar el tráfico en esa intersección puede tomar la decisión de comunicarse con los agentes en intersecciones adyacentes para informarles de la situación y requerirles que cambien su política de control. El sistema de control de tráfico urbano que proponemos permite sincronizar un subconjunto de las intersecciones; es decir, solicitar coordinación solo en un grupo de intersecciones y que el resto permanezca de acuerdo a su estrategia actual.

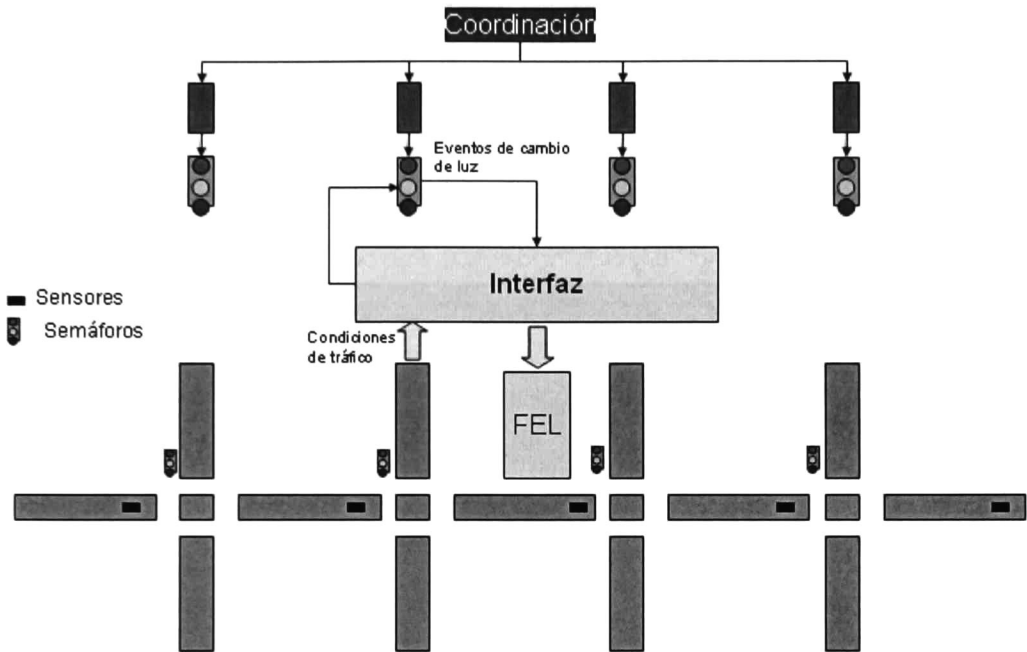


Figura 3.2 Interacción entre el agente de control y la máquina de simulación.

3.2.3. Arquitectura del agente de control

El agente de control es un agente proactivo debido a que toma las decisiones adelantándose a lo que ocurre en su ambiente. En la figura 3.3 se ilustra la arquitectura del agente de control. Describimos a continuación el funcionamiento del agente y de cada uno de sus componentes.

3.2.3.1. Funciones del agente de control

Inicialmente el agente de control genera eventos de cambio de luz que siguen un esquema de control fijo; estos eventos son agregados en la FEL del simulador, lo cual le indica al simulador que puede iniciar la simulación. Durante el transcurso de la simulación el simulador ejecuta los eventos contenidos en la FEL; la ejecución de estos eventos representa el estado de las calles en un tiempo determinado. El agente de control recibe del simulador el estado de las calles involucradas en la intersección en la cual está asignado.

Una vez que la simulación ha iniciado, el agente de control decide que políticas de control establecer de acuerdo a las condiciones de tráfico representado por el estado de las

calles provisto por el simulador. La decisión respecto a la modificación de una política de control genera eventos de cambio de luz que deben ser agregados a la FEL para su próxima ejecución. El agente de control se comunica con otros agentes por medio del envío de mensajes referentes al establecimiento de una política de control. Enseguida describimos el comportamiento de los componentes del agente de control.

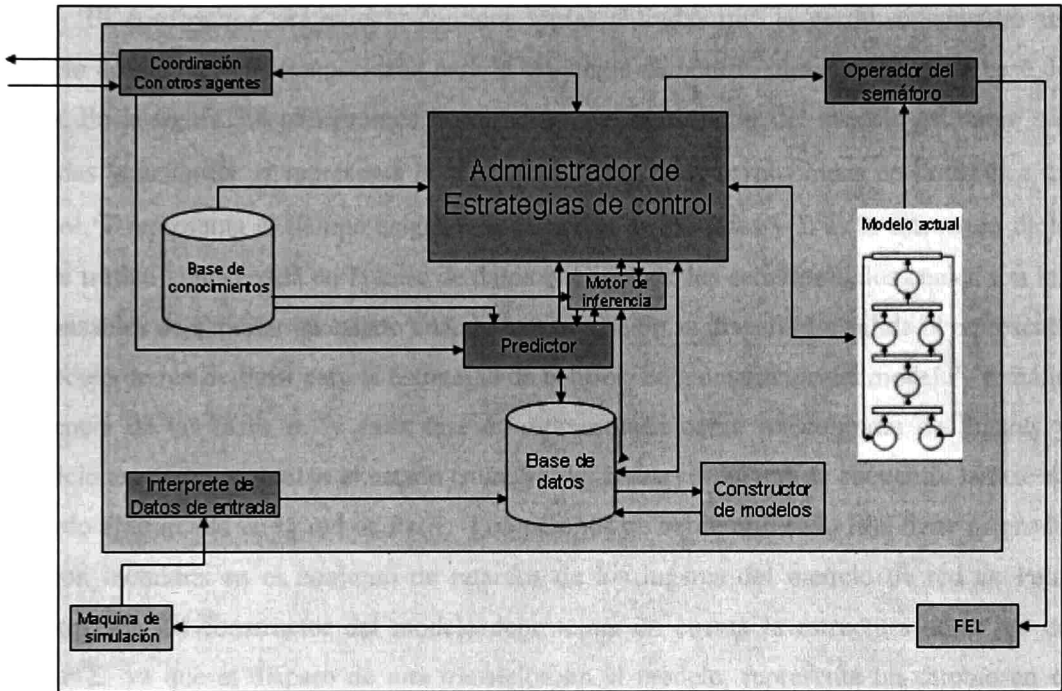


Figura 3.3 Arquitectura del agente de control.

3.2.3.2. Componentes del agente

Constructor de modelos de políticas de control

La tarea del agente de control, consiste en tomar la decisión de que estrategia de control aplicar de acuerdo al estado de tráfico en las calles. El conjunto de estrategias de control disponibles para la administración de las políticas de control, se encuentran almacenadas en la base de datos. Estas estrategias de control deben ser presentadas en forma de modelos de redes de Petri para poder ser interpretadas por el agente de control.

El módulo “*constructor de modelos de políticas de control*” se encarga de construir fuera de línea los modelos de red de Petri temporizadas correspondientes a las estrategias de control almacenadas en la base de datos, de tal forma que antes de que la simulación empiece, las políticas de control ya se encuentran disponibles en forma de modelos de RP para la aplicación de la estrategia de control correspondiente a dicho modelo.

El *constructor del modelo* contiene varias entradas que le permiten construir un modelo de red de Petri temporizada para la estrategia de control almacenada en la base de datos. En la figura 3.4 observamos la estructura del constructor del modelo así como sus entradas y su salida: σ representa la secuencia de las fases involucradas en la política de control, T representa el tiempo asignado a cada una de las fases y Σ es la estructura de la red de tráfico almacenada en la base de datos que incluye los semáforos, los cuales son los responsables de cambiar su estado cuando una transición se dispara. La salida N representa el modelo de red de Petri para la estrategia de control. El “constructor del modelo” toma la secuencia de las fases σ y cada fase es representada como un conjunto de lugares y transiciones que representan el estado (rojo, verde, ámbar) en el que se encuentra la fase de acuerdo al marcado en la red de Petri. Los tiempos de verde que cada fase tiene asignado (T) son incluidos en el conjunto de retardos de los lugares del modelo de red de Petri temporizada. El constructor del modelo debe tomar en cuenta la estructura de la red de tráfico Σ , ya que el disparo de una transición en el modelo, representa un cambio en el estado de algún semáforo en la red de tráfico. Finalmente los modelos de red de Petri generados por el “constructor de modelos de políticas de control” son almacenados en la base de datos para que otros componentes del agente de control puedan acceder a ellos.

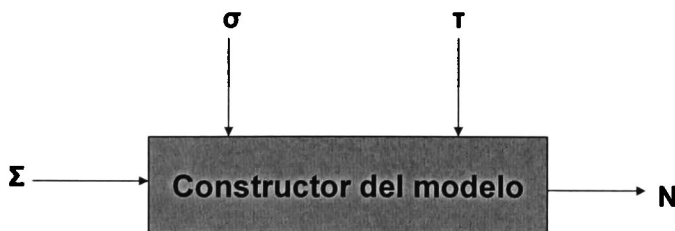


Figura 3.4 Módulo constructor del modelo.

Interprete de datos de entrada

Durante la simulación, mientras los eventos están siendo ejecutados por la máquina de simulación, es posible obtener el estado de las calles. El *intérprete de datos de entrada* se encarga de analizar toda esta información, y obtiene el flujo y densidad de tráfico de las calles involucradas en la intersección. El intérprete de datos de entrada envía esta información a la base de datos para ser almacenada.

La ejecución de los eventos inscritos en la FEL representa el estado en el que se encuentra la red de tráfico durante un periodo de tiempo determinado. El agente de control está interesado en conocer algunos de estos valores que pueden influir en las condiciones de tráfico de las intersecciones. Cada determinado tiempo el simulador le envía al agente de control la información referente a los eventos que ha ejecutado. El intérprete de datos de entrada analiza toda la información recibida y selecciona solo aquellos eventos que correspondan a eventos ocurridos en las calles pertenecientes a la intersección controlada por el agente (llegada de autos, salida de autos, cambio de carril); el intérprete de datos de entrada calcula de acuerdo a la información seleccionada, los valores de flujo de tráfico y densidad en esas calles.

Los valores correspondientes al flujo y la densidad deben ser almacenados en la base de datos para tener un registro del historial que representa las condiciones de tráfico durante el tiempo de simulación. Estos valores son tomados en cuenta por otros componentes del agente de control para la toma de decisiones.

Base de datos.

La *base de datos* contiene información sobre la estructura de la red de tráfico, tales como el número de calles involucradas, el número de carriles, los semáforos involucrados en la intersección, entre otras. La base de datos también contiene información sobre el estado de las calles en un instante de tiempo determinado. En la base de datos se encuentra almacenada la colección de hechos, la cual es requerida principalmente por el administrador de estrategias de control para la toma de decisiones. A continuación detallamos estos componentes.

Estructura de la red de tráfico.

• Estructura de un carril. La figura 3.5 ilustra la estructura de un carril. Una calle puede estar compuesta por más de un carril. Cada carril perteneciente a la red de tráfico, cuenta con un identificador único que lo distingue de los demás carriles en la estructura. Como puede observarse, ambos extremos del carril están etiquetados. El extremo A indica el inicio del carril, es decir, el segmento donde los vehículos procedentes de otra calle se posicionaran. El extremo B, representa el final del carril, en otras palabras, el segmento de desde donde los vehículos cruzan hacia otra calle. En la figura 3.6 ilustramos el esquema del carril; como puede notarse la dirección del carril está implícitamente representada en la estructura del mismo. Los extremos A y B pueden ser conectados a un nodo de la red de tráfico. Algunos carriles pueden contener un semáforo el cual es representado lógicamente por medio de la referencia del identificador del semáforo en el campo semáforo del carril. Los valores correspondientes al flujo y densidad se actualizan frecuentemente durante la simulación.



Figura 3.5 Estructura de un carril

CARRIL	
Identificador	
A	
B	
Semáforo	
Densidad	
Flujo	

Figura 3.6 Esquema del carril

- **Estructura de un nodo.** En la figura 3.7 observamos la estructura de un nodo. Esta estructura permite que dos o más carriles se unan. Cada nodo posee un identificador único que lo distingue de otros nodos. Su estructura es similar a la de un carril, solo que éste posee cuatro lados. En la figura 3.8 ilustramos el esquema de un nodo; los extremos del nodo pueden ser conectados a un carril o a otros nodos en la red de tráfico; cuando un extremo de un nodo es conectado a otro nodo forman una intersección. Al igual que en el carril los valores correspondientes al flujo y densidad se actualizan durante la simulación.

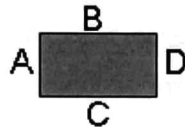


Figura 3.7 Estructura de un nodo.

Nodo	
Identificador	
A	
B	
D	
E	
Densidad	
Flujo	

Figura 3.8 Esquema del nodo

- **Semáforos.** Algunos carriles incluyen un elemento llamado semáforo, el cual les indica a los vehículos posicionados en un carril el derecho de cruzar. Los semáforos forman parte de la estructura de la red de tráfico. Cada semáforo contiene un identificador único que lo distingue de otros semáforos en la red de tráfico. En la siguiente figura ilustramos el esquema de un semáforo. El campo carril representa donde está lógicamente ubicado el semáforo, el estado representa la luz que despliega un semáforo en un instante de tiempo dado.

Semáforo	
Identificador	
Carril	
Estado	

Figura 3.8 Esquema de un semáforo

Construcción de la red de tráfico.

Cada extremo de un carril debe estar conectado a un nodo. El extremo A está conectado a la intersección desde la cual entran los vehículos al carril. El extremo B está conectado a la intersección hacia donde los vehículos pertenecientes al carril cruzan la intersección con la finalidad de llegar a otro carril.

La unión con otros elementos de la red de tráfico no es tan sencilla en el caso de los nodos. Los extremos A y D del nodo deben estar conectados a elementos carril. Los extremos B y C del nodo pueden conectarse a un carril, o bien, a otro nodo para formar una intersección. En la figura 3.9 se ilustra la construcción de un segmento de la estructura de la red de tráfico. Esta manera en que se construye la red, permite que un tiempo futuro la red de tráfico pueda incrementar el número de elementos que pertenecen a ella.

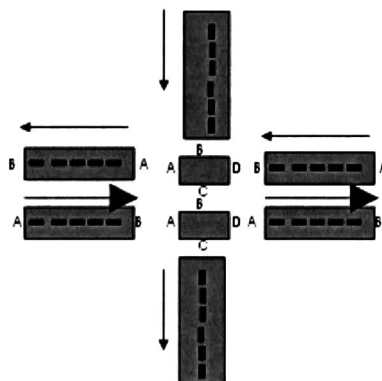


Figura 3.9 Construcción de una intersección.

Cada carril tiene almacenados en la base de datos sus valores históricos de flujo y densidad. El administrador de estrategias del agente de control, puede requerir en un

momento específico de la simulación los valores observados en el flujo y densidad del carril para la toma de decisiones.

Modelos de las estrategias de control

Cada intersección contiene un conjunto predeterminado de estrategias de control, disponibles para ser aplicadas. La base de datos almacena la estructura de los modelos de red de Petri que especifican las estrategias de control aplicables en dicha intersección. La siguiente figura ilustra el esquema de los modelos de RP que representan las estrategias de control. El campo “habilitada” indica si la estrategia de control representada es la establecida actualmente en la intersección que está siendo controlada. Los modelos son requeridos por el administrador de estrategias cuando decide modificar una política de control.

Estrategias de control	
Matriz de incidencia	
Conjunto de retardos	
Habilitada	

Figura 3.10 Esquema de las estrategias de control

Colección de hechos.

La colección de hechos alberga los datos correspondientes a la aplicación de determinada estrategia de control cuando se presentan ciertas condiciones en el tráfico observado. La colección de hechos puede desempeñar el papel de memoria auxiliar en la cual se registran los razonamientos llevados a cabo.

Inicialmente, la colección de hechos dispone solo de los datos que le hayan sido introducidos; pero a medida que el motor de inferencia vaya realizando razonamientos, que deduzcan nuevos conocimientos referentes al sistema, la colección de hechos se va actualizando. En la figura 3.11 se ilustra el esquema para la representación de los hechos en la base de datos. A continuación presentamos algunos hechos inicialmente incluidos en la colección de hechos.

1. Se aplica control fijo cuando el flujo y la densidad son predecibles y constantes.

2. Cuando el agente modifica sus políticas de control se debe avisar a los agentes en intersecciones vecinas.
3. Cuando un peatón solicita cruzar una calle, se le debe dar preferencia.

Colección de hechos	
Identificador	
Hecho	

Figura 3.11 Esquema de la colección de hechos

Base de conocimientos

La base de conocimientos contiene las reglas de decisión utilizadas por el predictor y el administrador de estrategias de control para realizar decisiones sobre el comportamiento del controlador. La base de reglas está organizada en subconjuntos de acuerdo al tipo de decisiones. El acceso a un determinado grupo de reglas se realiza a través de metareglas. A continuación se describen los diversos tipos de reglas.

- *Reglas de decisión para la modificación de la temporización de políticas.* Este subconjunto de reglas determina que el tiempo asignado a alguna fase involucrada en la intersección aumente o disminuya. Las condiciones en las reglas toman en cuenta los valores de flujo y densidad observados en las calles involucradas en la intersección. La variación en el tiempo asignado a alguna de las fases se calcula de acuerdo a las variables de flujo y densidad. Enseguida presentamos algunas reglas de decisión para la modificación de la temporización de políticas. Cabe mencionar que los valores “flujo requerido” y “densidad deseada” son constantes previamente definidas.

1. SI la densidad en el carril es menor que la densidad deseada ENTONCES disminuye el tiempo asignado a las fases que involucran ese carril.
2. SI el flujo en el carril es mayor que el flujo requerido ENTONCES disminuye el tiempo asignado a las fases que involucran ese carril.
3. Si la densidad en el carril es mayor que la densidad deseada ENTONCES aumenta el tiempo asignado a las fases que involucran ese carril.

4. SI el flujo en el carril es menor que el flujo requerido ENTONCES aumenta el tiempo asignado a las fases que involucran ese carril.

● *Reglas de decisión para el reemplazo de la política de control.* Este subconjunto de reglas determina si se debe continuar con la política de control actual, o si debe aplicarse una nueva política de control más adecuada a las condiciones dadas. En estas reglas el conjunto de condiciones establecen la política más adecuada como en el caso, de cuando debe activarse flecha a la izquierda, o cuando debe darse preferencia simultánea a carriles opuestos (Norte-Sur, Sur-Norte), o cuando hay que dar paso a peatones, entre otras. Enseguida presentamos algunas reglas de decisión para el reemplazo de la política de control.

1. SI la densidad en el carril es menor a la densidad en el carril opuesto ENTONCES habilitar política de control que da preferencia simultánea a carriles opuestos.
2. Si el flujo en el carril opuesto es cero ENTONCES continúe con la regla 5.
3. SI la densidad en el carril es mucho mayor que en el carril posterior ENTONCES habilitar política de control que permita flecha a la izquierda en el carril.
4. SI un peatón solicita cruzar una calle ENTONCES habilitar política de control que le de paso a los peatones.
5. SI la densidad en el carril opuesto es cero ENTONCES habilitar política de control que no le de preferencia a la fase que incluye el carril opuesto.

● *Reglas de decisión para establecer coordinación con otros controladores.* Este subconjunto de reglas determina cuando es necesario solicitar coordinación con controladores de intersecciones vecinas. Las reglas se activan en caso de que exista alguna situación de tráfico extrema en alguna calle adyacente a la intersección y no sea posible resolverla localmente, entonces es necesario solicitar coordinación con alguna intersección. Enseguida presentamos algunas reglas de decisión para establecer coordinación con otros controladores.

1. SI el flujo en el carril es mucho menor al flujo requerido ENTONCES continúe con la regla 3

2. SI la densidad es mucho mayor a la densidad deseada ENTONCES solicitar coordinación con la intersección aledaña que contiene al carril desde donde llegan los vehículos.
3. Si la densidad es mucho mayor a la densidad deseada ENTONCES solicitar coordinación con intersecciones aledañas.

● *Reglas de decisión para la predicción de condiciones de tráfico futuros.* Estas reglas toman en cuenta las condiciones actuales de tráfico en los diferentes carriles involucrados en la intersección y en intersecciones vecinas; su ejecución conduce a predicciones sobre el comportamiento de tráfico en carriles relacionados con la intersección. La predicción se realiza proyectando el comportamiento a 100 unidades de tiempo. El resultado de estas reglas sirve para que el administrador de estrategias se apoye en la toma de decisión de establecer una política de control. Enseguida presentamos algunas reglas de decisión para el reemplazo de la política de control.

1. SI la densidad observada en el carril posterior es mayor que la densidad deseada ENTONCES el flujo en el carril actual será menor.
2. SI la densidad observada en el carril anterior es mucho mayor a la densidad deseada ENTONCES la densidad en el carril actual será mayor.
3. SI el flujo en el carril es mayor al flujo requerido ENTONCES la densidad será menor.

Estas reglas deben validarse para demostrar que son correctas. En el capítulo cinco se presentan las pruebas y se analizan los resultados de aplicar la metodología de control para comprobar la validez de las reglas antes presentadas.

Predictor

El *Predictor* es un módulo importante para el desarrollo de un sistema de control de tráfico pro-activo. El agente de control puede estimar como estarán las condiciones de tráfico en un futuro cercano por medio del predictor, y de esta forma tomar en cuenta el estado futuro de la red de tráfico al proponer una política de control. El predictor compara la precisión de las predicciones conforme la máquina de simulación va proporcionando las

condiciones de tráfico actuales y realiza los ajustes necesarios si la predicción no es suficientemente precisa. El predictor es alimentado por el estado actual de las calles, las condiciones de tráfico relevantes (accidentes, lluvias, congestiones), las reglas de decisión provistas por la base de conocimientos, y el estado de las calles en intersecciones vecinas. El predictor puede calcular las condiciones de tráfico futuras, tomando en cuenta las reglas de decisión y por medio del módulo de vigilancia. Cuando el simulador envía el estado actual de la red de tráfico, la base de datos se actualiza con los valores correspondientes de flujo y densidad; es entonces cuando el predictor hace una llamada al motor de inferencia para deducir por medio de las reglas de decisión y de la base de hechos el comportamiento del tráfico en un futuro cercano; estas predicciones son almacenadas en la base de datos para ser utilizados por el administrador de estrategias en la toma de decisiones. El predictor también hace una llamada al módulo de vigilancia para corroborar que las predicciones realizadas sean precisas. A continuación describimos el comportamiento del módulo de vigilancia.

Módulo de vigilancia.

El predictor contiene un módulo especializado que hemos denominado *módulo de vigilancia*, el cual se encarga de asegurar que las predicciones realizadas por el predictor sean precisas. Las tareas que realiza el *módulo de vigilancia* incluyen:

- Comprobar que las predicciones realizadas estén de acuerdo con las condiciones actuales de tráfico.
- Asegurar que las condiciones de tráfico sean las deseadas.
- Informar al predictor sobre condiciones de tráfico adversas: accidentes, calles saturadas, vehículos de prioridad, peatones.

Administrador de estrategias

El agente de control es un agente proactivo, pues tiene la habilidad de adelantarse a las condiciones de tráfico que se presentaran en el futuro. El módulo encargado de realizar el cambio de políticas de control es el *administrador de estrategias*. La forma en que el administrador de estrategias realiza este cambio puede ser de dos maneras: a) actualizando los tiempos asignados a las fases en la política actual o b) reemplazando la política de

control actual por otra más adecuada. El *administrador de estrategias* incluye varias funciones especializadas en tareas específicas para proponer las políticas de control adecuadas. Estas funciones se describen a continuación.

Obtención de fases involucradas

Cuando se incrementa o disminuye la demanda de tráfico en alguna calle perteneciente a la intersección, es posible en algunos casos actualizar los tiempos asignados en las fases que toman en cuenta esa calle. La función de “obtener fases involucradas”, se encarga de determinar las fases que deben actualizar el tiempo de luz verde asignado debido al cambio en la demanda de tráfico de un carril.

Administración de la coordinación

Cuando se propone una nueva política de control es necesario informar a las intersecciones vecinas que se ha establecido una nueva estrategia. El administrador de estrategias envía la información referente a la nueva política de control establecida al “módulo de coordinación con otros agentes”, el cual se encarga de comunicarse con los agentes interesados en conocer los cambios realizados. El administrador de estrategias recibe la información correspondiente a la actualización de las políticas de control en intersecciones adyacentes a través del “módulo de coordinación con otros agentes”; esta información es procesada por el procedimiento de *administración de la coordinación* para calcular las fases que pueden verse afectadas debido a este cambio. Cuando un agente solicita coordinación con otros agentes, en este procedimiento se analiza la situación y se decide si se establece una política de control sincronizada. El resultado de este procedimiento es requerido por otros módulos del administrador de estrategias para la actualización de las políticas de control.

Actualización de políticas de control.

La adaptación del agente a nuevas condiciones de tráfico es realizado modificando las políticas de control actuales. Este cambio puede ser realizado en dos formas: actualizando la temporización a la política actual o bien reemplazando la política actual de control por otra.

- La *función de actualización de temporización* interactúa con la base de conocimientos a través del motor de inferencias para decidir si el tiempo actual asignado a las fases en la política de control debe ser modificado. La actualización se realiza en algunos o todos los retardos asociados a los lugares en el modelo de red de Petri de la política de control. Esta función recibe como parámetro el conjunto actual de retardos del modelo de red de Petri, y las condiciones de tráfico actuales de la red. Esta función regresa al modelo un conjunto nuevo de retardos asignados a los lugares.
- La *función de reemplazo de estrategias* interactúa con la base de conocimientos a través del motor de inferencias para determinar si la política de control actual debe ser reemplazada. Cuando el cambio de las condiciones de tráfico es significativo el motor de inferencia puede decidir sustituir la política actual con otra más adecuada al comportamiento actual de tráfico; incluso esta función permite adaptarse a la configuración definida por el usuario, como establecer una política a determinada hora; esta función recibe como parámetros la política actual de control, las condiciones de tráfico, y la configuración del usuario. Esta función regresa un nuevo modelo de red de Petri correspondiente a la nueva estrategia de control.

Activación de una nueva política de control.

Le indica al operador de tráfico que el modelo actual ha sido modificado; de esta forma el módulo “operador del semáforo” debe detener la ejecución de su actual política y empezar la ejecución de la nueva estrategia.

Motor de inferencia

El motor de inferencia es el encargado de controlar que reglas son activadas de acuerdo al comportamiento de tráfico observado. El motor de inferencia realiza un ciclo en el que de acuerdo a las condiciones dadas propone las modificaciones necesarias para establecer una nueva estrategia de control; la forma en que realiza este ciclo el motor de inferencias es como sigue:

- Obtiene como entrada las reglas de decisión en la base de conocimientos, las condiciones dadas de tráfico en forma de colección de hechos y por medio del predictor, el comportamiento que tendrá la red de tráfico en un futuro cercano.
- Si existe mas de una regla de decisión que se pueda activar, selecciona una (resuelve conflictos).
- Aplica las reglas de decisión que modifican la actual política de control.
- Mientras avanza el tiempo de simulación, el motor de inferencia provee nuevos razonamientos, los cuales son almacenados en la base de datos en forma de colecciones de hechos.

Modelo de la política de control

Debido a que las políticas de control pueden representarse como un sistema de eventos discreto, el modelo de las mismas puede ser representado por redes de Petri. Presentamos una definición de este formalismo para el modelado de sistemas de eventos discretos.

Definición de redes de Petri

Una red de Petri generalizada $N=(G,\mu)$ [López, 2002] es un dígrafo bipartito G con una función de marcado μ . $G=(P,T,Pre,Post)$, donde P es un conjunto finito de lugares ($|P|=n \geq 0$), T es un conjunto de transiciones ($|T|=m \geq 0$), $P \cap T = \{\}$; $P \cup T \neq \{\}$; Pre es una función de incidencia previa la cual representa el peso de los arcos de entrada a las transiciones, $Pre: P \times T \rightarrow \mathbb{N}^{\geq 0}$; $Post$ es una función de incidencia posterior (peso de los arcos de salida), $Post: P \times T \rightarrow \mathbb{N}^{\geq 0}$. Pre y $Post$ son representadas usualmente por matrices. Gráficamente los lugares se representan por círculos y las barras como transiciones.

Los lugares pueden contener tokens representados como puntos dentro de los círculos; la distribución de tokens dentro de la red, llamada marcado, es una función $\mu: P \rightarrow \mathbb{N}^{\geq 0}$, usualmente representado por un vector $[\mu(p_1) \dots \mu(p_n)]^T$, donde $\mu(p_i)$ es el número de tokens en p_i ; μ_0 describe el marcado inicial.

El comportamiento dinámico de un sistema modelado puede ser descrito por la evolución de los tokens a lo largo de los lugares en la red. La evolución del marcado está controlada por una regla de dos partes: a) una transición $t_j \in T$ está habilitada sii para toda

$p_i \in P$, $\mu(p_i) \geq \text{Pre}(p_i, t_j)$; b) el disparo de una transición t_j puede ser realizado si la transición está habilitada. Esto provoca modificación en el marcado: cada lugar de entrada de t_j pierde $\text{Pre}(p_i, t_j)$ tokens, y cada lugar de salida de t_j gana $\text{Post}(p_i, t_j)$ tokens. Esto es descrito por la siguiente expresión.

$$\mu'(p_i) = \mu(p_i) - \text{Pre}(p_i, t_j) + \text{Post}(p_i, t_j) \quad \text{para todo } p_i \in P$$

Redes de Petri Temporizadas.

Una *RPTT* (*Red de Petri temporizada en transiciones*) está definida [López, 2002] como: (N, D, Γ) , donde N es la RP generalizada, D es un conjunto de retrasos; $D = \{d_i \mid i=0, \dots, m\}$, donde cada $d_i \in \mathbb{R}^+ \cup \{0\}$ representan el tiempo que una t_i toma para realizar la ejecución que representa la transición y Γ es un conjunto ordenado cuyos elementos $(t_j, t_k) \in T \times \mathbb{R}^+ \cup \{0\}$ representa el instante de disparo t_k de una transición t_i . El disparo que se realiza en una transición t_i se realiza como en cualquier otra RP con una pequeña variación: Las marcas que se agregan a los lugares de salida de t_i aparecen después de d_j unidades de tiempo.

Una *RPTL* (*Red de Petri temporizada en lugares*) es definida [E.López, 2002] como: (N, D, Γ) , donde D y Γ son definidas como en las RPTT. Γ representa el instante de disparo t_k de una transición t_i . $D = \{d_i\}$ para $i=1, \dots, n$, cada $d_i \in \mathbb{R}^+ \cup \{0\}$ especifica el tiempo en el que las marcas que llegan al lugar p_i no están disponibles. Una marca está disponible después de transcurridos d_i unidades de tiempo desde el instante de su llegada a p_i . Solo las marcas disponibles habilitan las transiciones.

Descripción del modelo

El modelo que describe la política de control actual la cual determina la operación de los semáforos en la intersección, está representado por medio de una red de Petri temporizada que especifica todos los movimientos que tienen derecho de paso en cada fase de la política de control, y la duración de cada fase.

La figura 3.12 muestra un modelo de red de Petri temporizada que representa una política de control de tres fases. Las marcas en los lugares representan el estado en el que se encuentra el sistema de control en un instante de tiempo determinado. El retraso asignado a cada uno de los lugares p_1 , p_3 y p_5 representa el tiempo durante el cual los movimientos

pertenecientes a la fase respectiva tendrán luz verde. El retraso asignado a los lugares p2, p4 y p6, representa un tiempo fijo de luz ámbar.

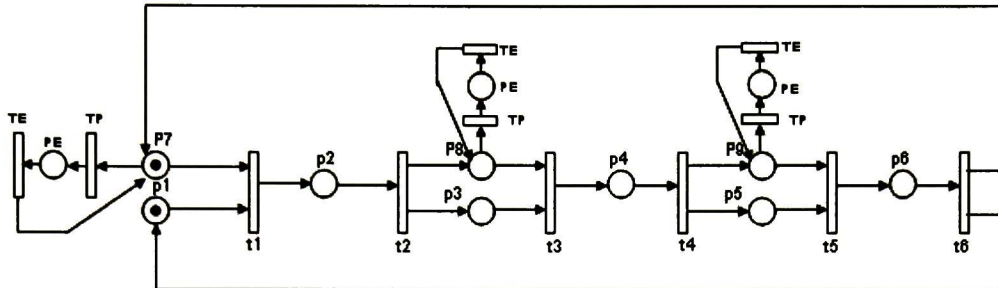


Figura 3.12 Modelo de RP temporizada para una política de control de 3 fases.

La evolución del marcado determina el cambio de una fase a otra. El marcado inicial con marcas en los lugares p1 y p7 representa que el semáforo tiene luz verde en la primera fase; eventualmente un peatón podría solicitar permiso para cruzar la intersección; en ese caso se habilita un evento de cruce de peatón y la transición TP se dispara, lo que provoca que se marque el lugar PE. El lugar PE tiene asignado un retardo, el cual representa el tiempo durante el cual los peatones tendrán derecho de pasar. El disparo de la transición TE regresa la marca al lugar P7 y de esta forma el semáforo tiene luz verde para la primera fase. Cuando se dispara la transición t1, se marca el lugar p2 que representa luz ámbar para la fase 1. Cuando se dispara la transición t2, los lugares p3 y p8 se marcan lo que indica que la fase 2 tiene ahora luz verde. La evolución del marcado a partir de estos lugares puede realizarse de forma análoga a la descrita en el marcado inicial.

En la siguiente figura se presenta una política de control más compleja; el modelo representa dos posibles secuencias de fases (las cuales pueden ser consideradas como dos políticas de control). El operador del semáforo puede disparar la secuencia $\sigma_1 = (t_1 t_2 t_3 t_4 t_5 t_6)^*$ mientras el evento externo asociado a t'1 no se active; cuando ocurre tal evento entonces t'1 se dispara y la secuencia $\sigma_2 = (t'_1 t'_2 t'_3 t'_4 t'_5 t'_6)^*$ es disparada. La primera secuencia de fases es disparada de nuevo cuando t'2 (de acuerdo a un evento externo) es disparado.

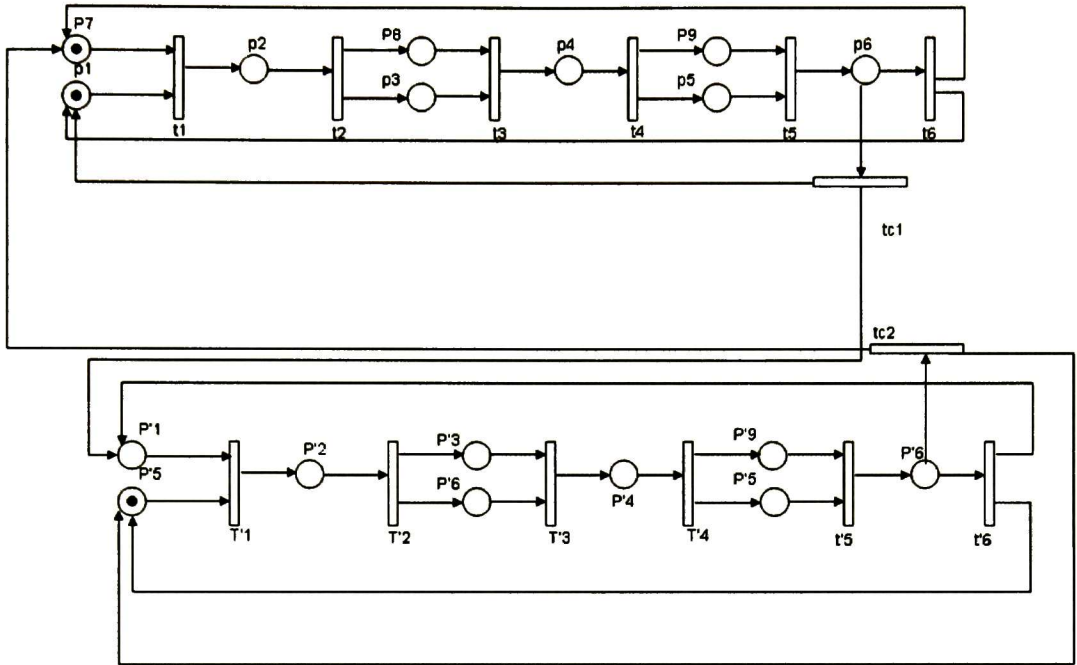


Figura 3.13 Reemplazo de políticas de control.

Como podemos darnos cuenta existen numerosas posibilidades para establecer una estrategia de control. Es posible según vimos en este último ejemplo combinar secuencias de fases y de esta manera habilitar diferentes políticas en momentos determinados durante el tiempo de simulación; el agente puede proponer políticas de control que respondan a peticiones de los peatones o de los vehículos de prioridad.

Operador del semáforo

El operador del semáforo ejecuta el modelo de la política de control actual. La secuencia de eventos (disparo de transiciones) y los instantes de tiempo en que son disparados son determinados por la ecuación de estados de la red de Petri. Estos eventos son traducidos en eventos significativos para el simulador, los cuales son agregados a la lista de eventos futuros respetando el orden de los eventos que ya han sido agregados a la lista

Cuando el administrador de estrategias le indica al operador de tráfico que la actual política de control ha cambiado, el operador de tráfico debe terminar la ejecución del modelo actual. Cuando se detiene una política de control, el operador de tráfico espera a

que termine la ejecución del modelo actual y posteriormente inicia la ejecución del modelo de la nueva política de control. La nueva política puede ser tan simple como modificar los tiempos asignados a las fases, o bien, puede ser tan compleja como modificar completamente la secuencia de las fases o el orden de las mismas.

Cuando una transición es disparada, el disparo de esta transición representa nuevos eventos de cambio de luz que deben ser inscritos en la lista de eventos futuros. El intérprete de transiciones analiza la estructura del modelo de red de Petri, y de acuerdo a la información obtenida, determina que eventos deben agregarse en la lista de eventos futuros y el instante de tiempo en el que deben ejecutarse, respetando el orden en que son inscritos.

Módulo de coordinación con otros agentes

El agente de control tiene la capacidad de comunicarse con otros agentes asignados en otras intersecciones; para ello módulo de coordinación está encargado de proveer un mecanismo de comunicación; este módulo es el módulo de coordinación realizando las siguientes funciones:

- Se encarga de informar a los agentes asignados en otras intersecciones, sobre los cambios realizados a las políticas de control.
- Solicita la modificación de las políticas de control en intersecciones vecinas. Cuando se presentan ciertas condiciones de tráfico, tales como la entrada de un vehículo de prioridad a la intersección; el módulo de coordinación de agentes le informa a las intersecciones vecinas sobre esta situación; entonces el agente de control asignado en esas intersecciones puede tomar la decisión de modificar sus estrategias de control para permitir que el vehículo de prioridad fluya por estas intersecciones sin ningún retraso
- El módulo de coordinación determina a que intersecciones es necesario informarles sobre las modificaciones de las políticas de control. Debido a que en una red de tráfico existen muchas intersecciones, no es necesario informarle a todas las intersecciones pertenecientes a la estructura de la red sobre los cambios realizados en las políticas de control, pues esto ocasionaría un uso excesivo de los recursos del sistema; sin embargo, el módulo de coordinación determina de acuerdo a ciertas

condiciones, cuales intersecciones deben conocer los cambios realizados por el agente.

3.3. Coordinación de agentes de control

En la sección anterior describimos las características del agente de control, así como de sus componentes. Como en todo sistema multiagente, los agentes involucrados en el sistema pueden comunicarse unos con otros. En esta sección describimos la manera en que los agentes de control se comunican, y la sincronización realizada para establecer una política de control.

3.3.1. Escenario

La finalidad de usar varios agentes en un ambiente de simulación, es que el sistema multiagente pueda funcionar para una simulación distribuida y a su vez pueda aplicarse a una red de tráfico real. La estructura básica del sistema de control propuesto consiste de un conjunto de agentes de control. Un agente de control es asignado en cada intersección, y realiza las funciones y cálculos requeridos en esa intersección. Los agentes interactúan con la máquina de simulación y con otros agentes asignados en otras intersecciones, con el objetivo de permanecer sincronizados.

Cada vez que un agente realiza un cambio en las políticas de control de la intersección a la cual está asignado, este les avisa a los agentes que deben conocer los cambios que se realicen en las estrategias de control de la intersección. Los agentes que reciben la información pueden tener dos posibles respuestas:

- *Ignorar la información.* Cuando el agente considera que la información recibida no afectará las condiciones de tráfico en alguna calle perteneciente a la intersección que está siendo controlada, entonces el agente de control no toma en cuenta la información recibida para posibles modificaciones en sus políticas de control.
- *Analizar la información.* En dado caso que un agente de control considere que la información enviada por otro agente, afectará significativamente las condiciones de tráfico en alguna calle perteneciente a la intersección involucrada, el agente analiza la información recibida y decide si aplica un cambio a su actual política de control.

3.3.2. Comunicación entre agentes

Para asegurar un buen desempeño del controlador y un mecanismo de comunicación entre los agentes de control es necesario integrar y coordinar las actividades de cada agente con el ambiente. Para lograr esto, proponemos un esquema de comunicación entre los agentes. En la figura 3.14 se ilustra la manera en que están distribuidos los agentes en la red de tráfico. Se tiene una red de tráfico la cual contiene una avenida principal con dos carriles, y cuatro calles paralelas que intersectan la avenida principal.

Cada agente de control provee protocolos, algoritmos y bases de conocimiento para el aprendizaje del agente y la toma de decisiones para proponer una política de control óptima y de esta forma asegurar el buen comportamiento en la intersección para la cual está asignado; esto se refleja en los tiempos asignados en las fases involucradas en la intersección correspondiente.

Se supone que cada segmento de calle perteneciente a alguna intersección contiene sensores, los cuales proporcionan al agente de control el estado de las calles en un instante determinado. Cuando el agente de control proporciona los eventos de cambio de luz de acuerdo a la política de control establecida; las fases cambian.

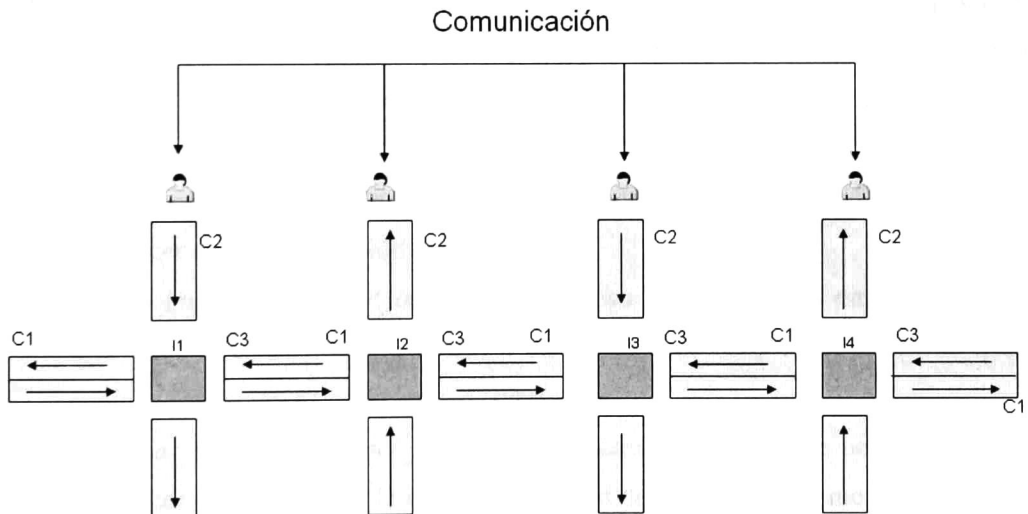


Figura 3.14 Coordinación entre los agentes de la red de tráfico

Como se puede observar, los agentes están conectados unos con otros, esto indica que existe comunicación entre ellos; los agentes están distribuidos en la red de tráfico. La

comunicación representa la interacción que tienen los agentes cuando cambian sus políticas de control.

3.3.3. Sincronización de políticas de control

Es posible que un subconjunto de los agentes involucrados en la simulación, se coordinen para proponer una estrategia de control común a ellos. En las avenidas principales es deseable que los tiempos asignados a las calles aledañas estén desfasados mediante un delta pequeño; esto se logra sincronizando los tiempos asignados a las fases en cada una de las intersecciones.

Cuando un agente cambia de política de control o requiere la sincronización con otros, este agente emisor contacta a otros agentes para iniciar una conversación; los agentes interesados en comunicarse con él le responden; enseguida el agente de control les envía la información relevante y de acuerdo a esa información los agentes interesados deciden que acciones deben seguir en el establecimiento de sus políticas de control. Los agentes receptores a su vez deben responder al agente emisor que los mensajes enviados han sido recibidos. Los métodos definidos para la comunicación entre agentes son:

- *Buscar agentes (lookup_agent)*: devuelve la localización de un agente específico o de un subconjunto de los agentes que pueden estar interesados en conocer los cambios realizados en la política actual del agente que llama al método lookup_agent.

Iniciar comunicación (start_dialog): Contacta a los agentes interesados en establecer una comunicación con el agente.

Acepta invitación (accept_invitation): En caso que un agente este interesado en establecer la comunicación, le envía un mensaje de aceptación al agente que le envió la invitación para iniciar la conversación.

- *Rechazar invitación (reject_invitation)*: En caso que el agente no se interese en establecer comunicación, le envía un mensaje de rechazo al agente que le envió la invitación para iniciar la conversación.

Envía información (send information): El agente de control envía información referente a los cambios que ha realizado en sus políticas de control a los agentes que han aceptado la invitación de iniciar la conversación.

Recibe información (get_information): Los agentes a los que se les envía la información acerca de los cambios realizados en las políticas de control, le indican por medio de este método al agente de control, que han recibido satisfactoriamente la información.

En la siguiente figura observamos la sincronización de una política de control para evitar que un vehículo de control tenga que hacer alto en alguna intersección. En (a) observamos que una ambulancia entra a una calle perteneciente a la intersección 1 en el instante 8:00 el agente asignado a esa intersección le envía una solicitud a los agentes asignados en otras intersecciones para que establezcan una política sincronizada para permitir que el vehículo de prioridad fluya libremente a través de la red de tráfico; los agentes asignados a las otras intersecciones responden satisfactoriamente a la petición realizada por el agente emisor lo cual permite que en el instante 8:20 (b) se aplique una estrategia de control favorable al vehículo de prioridad en la ultima intersección observada.

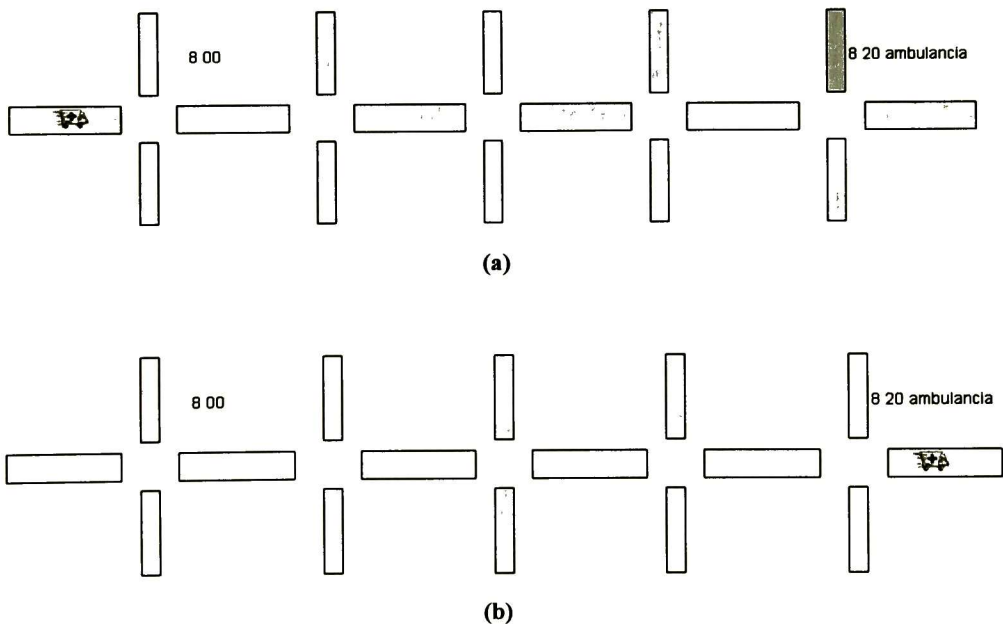


Figura 3.15 Sincronización de políticas de control.

3.4. Conclusiones.

En este capítulo presentamos un esquema basado en sistemas multiagentes para el control de tráfico urbano. En nuestro esquema el semáforo de cada intersección es controlado por un agente; este agente propone una estrategia de control de acuerdo a la demanda de tráfico. Nuestra metodología permite que un agente proponga diferentes políticas de control en instantes de tiempo diferentes durante la simulación. El agente de control interactúa con otros agentes en intersecciones vecinas para que las estrategias de control estén sincronizadas en la estructura de la red de tráfico; esta interacción puede ser total o parcial; en otras palabras, puede requerirse que toda la red de tráfico este completamente sincronizada, o que solo un subconjunto de intersecciones se coordinen.

En el próximo capítulo se describen las herramientas utilizadas para la implementación de los agentes de control, y se describen los medios para establecer la interacción entre los agentes.

Capítulo 4

Implementación del sistema de control de tráfico urbano basado en sistemas multiagentes

Resumen: En este capítulo presentamos una biblioteca de clases y métodos para la implementación de un sistema multiagente para el control y administración de un sistema de tráfico urbano. Las clases y métodos están definidos en el lenguaje de programación Java. Además, definimos la implementación de los componentes del agente de control.

4.1. Introducción

En nuestro trabajo se presenta la implementación de un sistema multiagente para el control de tráfico urbano, en donde cada agente está encargado de realizar el control de la intersección en donde está asignado. El sistema no solo provee el control individual de cada intersección, sino que permite que los agentes se coordinen parcial o totalmente para el establecimiento de una estrategia de control sincronizada. Se proponen las clases y métodos necesarios para construir el agente de control y los protocolos de comunicación entre ellos.

Se utiliza Java como lenguaje de programación, ya que es un lenguaje de programación multiplataforma que permite la programación concurrente y distribuida a través de hilos indispensable para la comunicación entre los agentes; además su distribución gratuita ha ayudado a su difusión.

Por otra parte, nuestro sistema de control de tráfico permite que en un momento inesperado de la simulación, de acuerdo a la demanda de tráfico, se reemplace la política de control actual; el modelo de la estrategia de control es representado por medio de redes de Petri

4.2. Estrategia general

En esta sección presentamos la estrategia para el desarrollo de nuestro sistema de control. Se utiliza el Middleware ProActive (figura 4.1) para establecer la comunicación entre los módulos involucrados en la simulación. ProActive también proporciona los protocolos necesarios para la comunicación entre los agentes de control.

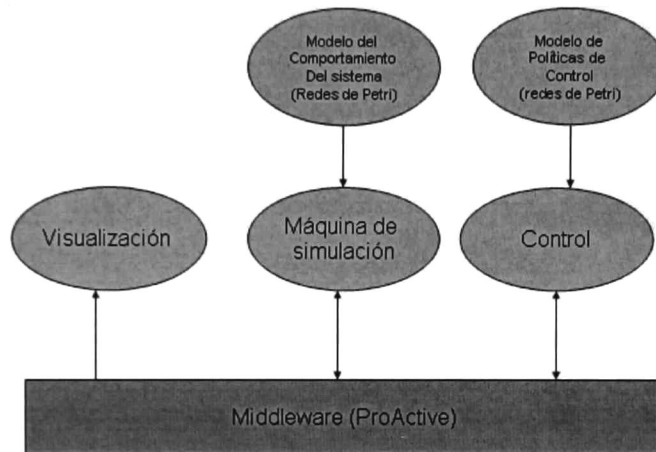


Figura 4.1 Interacción por medio de Middleware ProActive

Presentamos un conjunto de diagramas y una biblioteca de clases e interfaces para la implementación de los componentes del agente de control. Se presenta una descripción general del Middleware ProActive y las razones que nos motivaron a utilizar este Middleware. Por último describimos la interacción que el módulo de control tiene con la máquina de simulación, así como los protocolos para la comunicación entre los agentes de control.

4.2.1. El agente de control

Describiremos brevemente algunas características importantes de los módulos que componen el agente de control. La siguiente figura muestra el diagrama de clases en notación UML del sistema de control. Este diagrama contiene las clases más importantes de la biblioteca propuesta. En las siguientes subsecciones se describen los atributos y métodos más importantes de las clases.

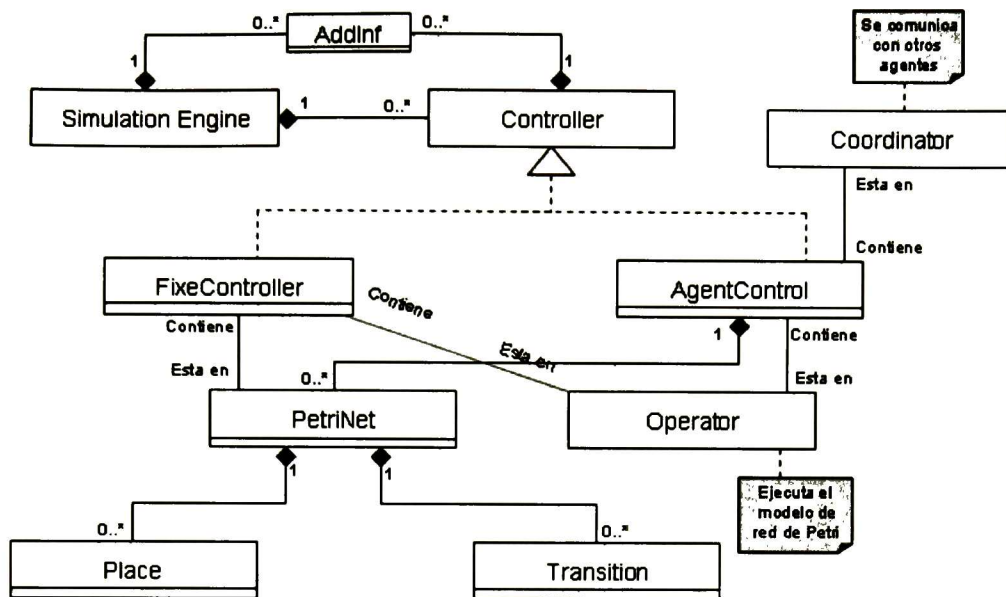


Figura 4.2 Diagrama UML de las principales clases para implementar el sistema de control.

El siguiente código Java muestra el esqueleto de la clase agente de control con sus principales atributos y métodos. Dentro de esta clase se encuentran implementados cada uno de los componentes contenidos en el agente de control.

```

public class ControlAgent{
    final double distance;
    final double AverageVel;
    final double desiredDensity;
    final double desiredFlow;
    private double MinimunGreenTime;
    private double InitialInterval;
    final double MaximunGreenTime;
    final double NumberOfCarPerSecond;
    final double DesiredFlow;
    final double DesiredDensity;
    public String IntersectionValue;
    public String SecuenciaValue;
    public double TiempoInicial;
    public int NumberOfFases;
    public PN[] model;
    private double Delta[];
    public double Ciclo;
    private double[][] FasesDinamicas;
    ////Constructor
    public ControlAgent(String _intersection,String _secuencia,double _tiempoinicial,double _ciclo)

    //Metodos publicos
}

```

```

public void startController(String _interseccion,String _secuencia,double _tiempoinicial)
public int[] getSemaphoresValues(String value)
public int getDirectionSemaphore(int LinkValue,int Direction)
public int getLinkValue(String Node)
public void DynamicChange(int _link,double _flujo,double _densidad,int NumPN)
public void changeDelta(int pos,double newdelta,int NumPN)

//Metodos privados
private void createNet(String seqPhases,int NumPN)
private void setMarked(String _label,int tokens,int NumPN)
private void startPetriNet(int NumPN)
private void setConnection()
private void predictor(double flow, double density)
private void management_strategy()
private double[ ][ ] asignarInterseccion(String curInterseccion,String curSecuencia)
private void copyFases(String seqPhases)
private void calculateGreenTimes()
private int getLinkIntersection(int _link)
private int[ ] getIntersectionLink(int _interseccion)
private double[ ] getLinkPhases(int Sec1,int Sec2)
}

```

El método `startController` es el que inicia la ejecución del agente de control. El código siguiente describe el comportamiento de este método. Primeramente el agente se conecta a la base de datos (`setConnection`) para obtener los valores iniciales referentes a las estrategias de control; para crear los modelos correspondientes a las políticas de control disponibles se hace una llamada a los procedimientos `createNet()`; y finalmente el operador de semáforos (`startPetriNet`) genera los primeros eventos de cambio de luz, los cuales son enviados al simulador.

```

public void startController(String _interseccion,String _secuencia,double _tiempoinicial){
    Intersection Value=_interseccion;
    SecuenciaValue=_secuencia;
    TiempoInicial=_tiempoinicial;
    setConnection();
    createNet("Fase 1",0);
    createNet("Fase 2",1);
    createNet("Fase 3",2);
    /*We assign the initial times, Initial Interval, minimum interval, and so on*/
    calculateGreenTimes();
    startPetriNet(0);
}

```

Hemos descrito la fase inicial del agente de control, el cual genera los primeros eventos de cambio de luz. Estos eventos son establecidos de acuerdo a un esquema de control fijo; pero a medida que avanza la simulación el agente de control se adapta a las

condiciones de tráfico. A continuación presentamos la implementación de los principales componentes del agente de control que permiten que el agente de control tenga un comportamiento adaptativo.

Base de datos

La base de datos contiene información referente a la estructura de la red de tráfico, también contiene la definición de las estrategias de control disponibles para la intersección, y se encuentra almacenada también la colección de hechos, la cual es requerida principalmente por el administrador de estrategias de control para la toma de decisiones.

La información almacenada en la base de datos debe ser administrada por un controlador de base de datos. Debido a que la implementación del sistema de control es realizada bajo la plataforma Windows, se seleccionó Microsoft Access como el manejador de base de datos por default; sin embargo, la migración a cualquier otro controlador de base de datos es posible gracias a la manera en que Java interactúa con los manejadores de bases de datos. Presentamos a continuación la descripción de algunas tablas contenidas en la base de datos.

- La tabla *Red* representa la estructura de la red de tráfico. En esta tabla se almacena la información referente a la manera en que están relacionados los componentes de la red de tráfico. La figura 4.3 nos muestra la estructura de la tabla red.

Tabla Red

Integer ID
Integer Type
Integer A
Integer B
Integer C
Integer D
Integer Type_Ctrl
Integer Id_Ctrl

Figura 4.3 Tabla Red

La red de tráfico puede contener dos tipos de elementos: calles e intersecciones. El campo ID es el identificador único de cada uno de los elementos de la red de tráfico; Type indica si el elemento es de tipo calle o de tipo intersección; los campos A,B,C, y

D relacionan cada elemento de la red de tráfico, con otros elementos de la red; Type_Ctrl e Id_Ctrl se refieren respectivamente al tipo de control (semáforo, ceda el paso, señal de alto) y al identificador del elemento de control. A manera de ilustración presentamos en la figura 4.4 el proceso para la construcción de un segmento de la red de tráfico.

Id	Type	A	B	C	D	Type_Ctrl	Id_Ctrl
Int 1	0	Calle 1	Int 2	Calle 6	Calle 2	0	0
Calle 1	1	0	Int 1	0	0	1	1
Calle 2	1	Int 1	0	0	0	0	0
Int 2	0	Calle 4	Calle 5	Int 1	Calle 3	0	0
Calle 3	1	0	Int 2	0	0	1	2
Calle 4	1	Int 2	0	0	0	0	0
Calle 5	1	0	Int 2	0	0	1	3
Calle 6	1	Int 1	0	0	0	0	0

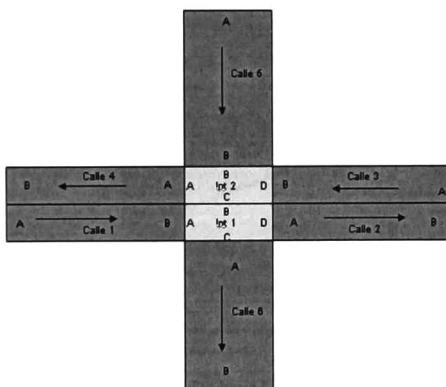


Figura 4.4. Construcción de una red de tráfico.

Intérprete de datos de entrada

Una vez que ha iniciado la simulación, el agente de control recibe información relacionada al estado de las calles. El módulo “interprete de datos de entrada” está encargado de interpretar esta información enviada por la máquina de simulación. El diagrama de flujo de la figura 4.5 describe el comportamiento general del “intérprete de datos de entrada”.

Como puede notarse en el algoritmo, el intérprete de datos de entrada realiza el análisis de los datos durante todo el tiempo que dura la simulación. En el segundo ciclo “mientras que”, se hace énfasis a que deben existir al menos tres cambios de luz programados en la lista de eventos futuros, antes de que se calculen nuevos valores de flujo y densidad; esta condición asegura que todos los cambios de luz que se programen respetaran el orden de los eventos programados en la FEL. Los valores obtenidos de flujo y

densidad son almacenados en la base de datos, para luego ser utilizados por otros componentes del agente de control. El siguiente código es un fragmento de la implementación de este módulo del agente de control.

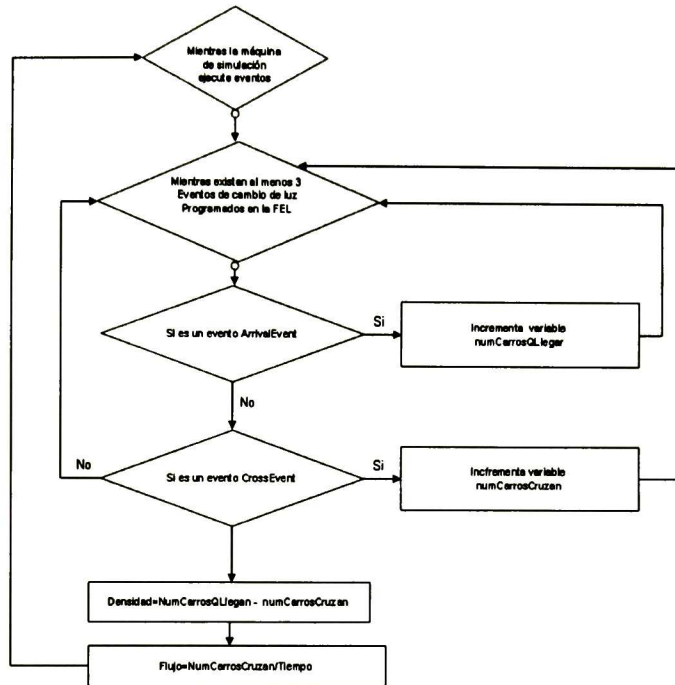


Figura 4.5 Comportamiento del intérprete de datos de entrada.

```

public void getFlowDensity(String s_message)
{
  ...
  ...
  While (NumberOfCLESEnded-NumberOfCLEExecuted>3)
  {
    if (_link>0)
    {
      if (_type==10 || _type==7)//Es un arrival link event
      {
        NumberOfCarsArrive[_link]=NumberOfCarsArrive[_link]+1;
      }
      if (_type==2)//Cross Event
      {
        NumberOfCarsCross[_link]=NumberOfCarsCross[_link]+1;
      }
    }
  } //END WHILE

  for (int h=1;h<27;h++){
    Densidad[h]=Densidad[h]+NumberOfCarsArrive[h]-NumberOfCarsCross[h];
    flujoi=(double)NumberOfCarsCross[h]/100;
    updateLinkTable(h, Densidad[h],flujoi);
    n_message=h+" "+flujoi+" "+Densidad[h]+" "+_time;
    /*Links involved in this intersection controlled by this agent*/
    for(this.userBag.newIterator(); this.userBag.hasNext().)
    {
  
```



```

    this.userBag.next();
    ///Para que solo le envíe cross events, cambio de luz, y arribos
    {
        numberOfEventsSent++;
        this.userBag.currentUser().ExecutedEvent(n_message);
    }
}
}
cycleFlow++;
//Should start the PN
//NumFlujos++;
for (int h=0;h<27;h++){
    NumberOfCarsArrive[h]=0;
    NumberOfCarsCross[h]=0;
}
}
}
}

```

Cuando se han calculado los valores de flujo y densidad, estos son almacenados en la base de datos (updateLinkTable()) para que puedan ser consultados por otros componentes del agente de control.

Administrador de estrategias y predictor

Conforme avanza la simulación el simulador envía al agente de control información referente al estado de las calles. Con la información recibida el administrador de estrategias junto con el predictor se encargan de proponer los modelos de las políticas de control que mejor se adapten a la demanda de tráfico. En la figura 4.6 se ilustra el diagrama de flujo que describe el comportamiento de estos módulos.

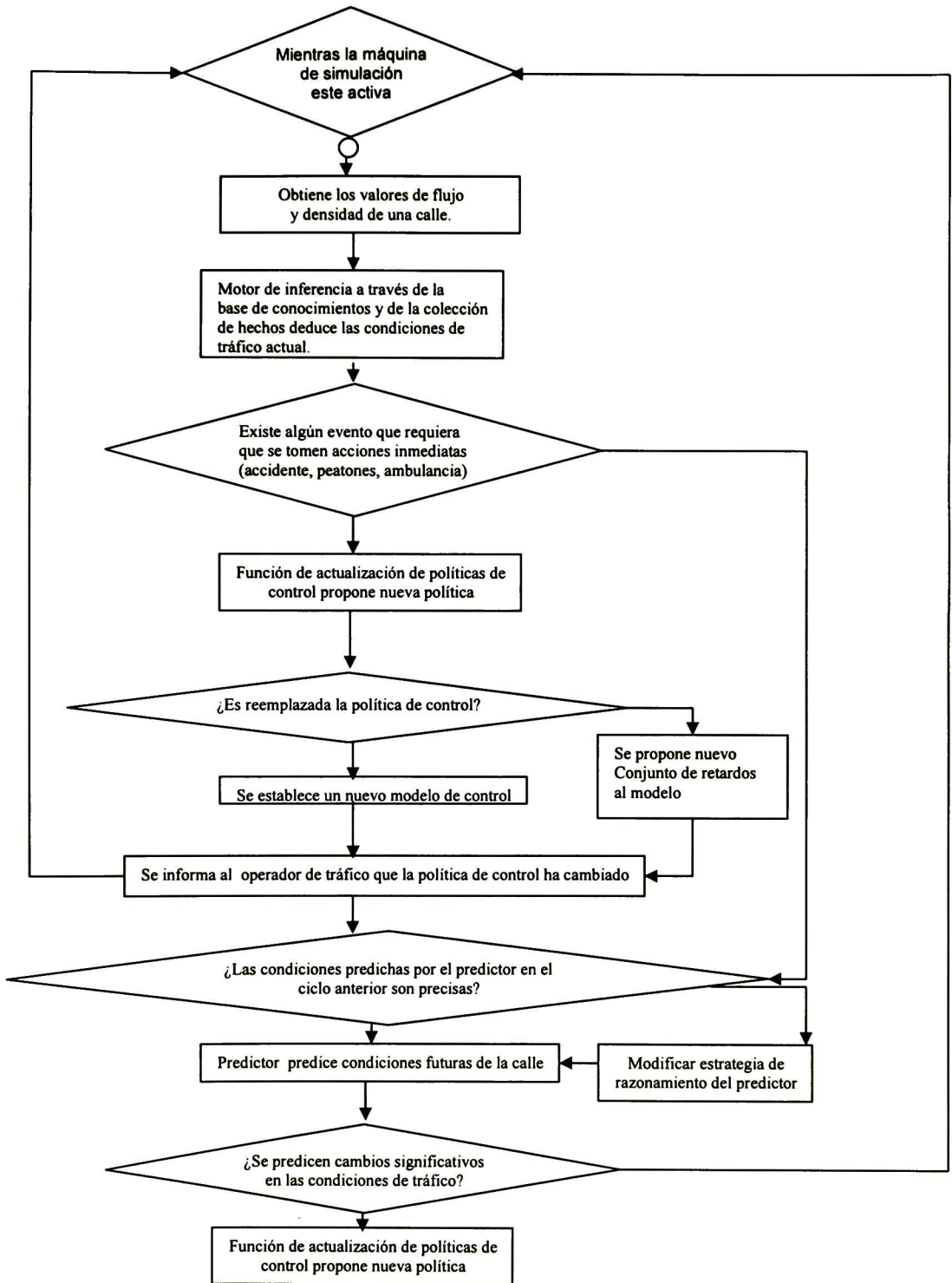


Figura 4.6 Diagrama de flujo del administrador de estrategias

Como pudimos darnos en cuenta en el algoritmo, tanto el administrador de estrategias como el predictor llaman al motor de inferencias para deducir el comportamiento que se debe seguir. En el siguiente código se realiza la llamada al motor de inferencias.

```

public void Administrador_Estrategias(int _link,double _flujo,int[] _densidad,int NumPN)
{
    String _label="";
    double delta=0; //Incremento o decremento
    double[] involvedphases={0,0,0,0,0,0,0,0,0,0}; //Hasta 10 fases involucradas.
    int _intersectionsides[]={0,0,0,0};
    int _intersection=getLinkIntersection(_link,true);
    int opposed_link=0;
    if (_intersection!=0)//Existe un semaforo en la interseccion
    {
        //Obtiene las fases involucradas
        GetInvolvedPhases(_link);
        delta=0;
        //Motor de inferencia
        MotorDeInferencia("ActualizaTiempos",_flujo,_densidad);
        MotorDeInferencia("ReemplazaPolitica",_flujo,_densidad);
        MotorDeInferencia("Predice",_flujo,_densidad);
    }
}

```

Motor de inferencias

El motor de inferencias contiene las metareglas que seleccionan el subconjunto de reglas de decisión que deben dispararse para la toma de decisiones. A continuación presentamos un fragmento del código del motor de inferencias.

```

public void MotorDeInferencia(String llamada,double _flujo, int _densidad)
{
    if (llamada.equals(new String("ActualizaTiempos")))
    {
        ParserRulesDecision1(_flujo,_densidad);
    }
    if (llamada.equals(new String("ReemplazaPolitica")))
    {
        ParserRulesDecision2(_flujo,_densidad);
    }
    if (llamada.equals(new String("Coordina")))
    {
        ParserRulesDecision3(_flujo,_densidad);
    }
    if (llamada.equals(new String("Predice")))
    {
        ParserRulesDecision4(_flujo,_densidad);
    }
}

```

Base de conocimientos

Las reglas de decisión están almacenadas en la base de conocimientos; la base de conocimientos está diseñada para que pueda ser actualizada con nuevas reglas de decisión. El siguiente código describe algunas reglas de decisión.

```
private void ParserRulesDecision1(double _flujo,int _densidad)
{
    double delta=0; //Incremento o decremento
    if (_flujo<RequiredFlow-0.7)
    {
        delta=(double)(_densidad-desiredDensity)/4)*InitialInterval;
    }
    if (_densidad>DesiredDensity+20)
    {
        delta=(double)(_densidad-desiredDensity)/6)*InitialInterval;
    }
    if (_densidad<DesiredDensity-30)
    {
        delta=(double)(_densidad-desiredDensity)/12)*InitialInterval;
    }
    if (_flujo>RequiredFlow+1)
    {
        delta=(double)(desiredFlow-_flujo)*InitialInterval;
    }
    if (_flujo<=RequiredFlow-1.5)//Aumentar el tiempo ya que pocos carros estan pasando
    {
        delta=(double)(desiredFlow-_flujo)*InitialInterval;
    }
    if (_flujo<RequiredFlow-0.5)
    {
        delta=(double)(_densidad-desiredDensity)/3)*InitialInterval;
    }
    if (_densidad>DesiredDensity+25)
    {
        delta=(double)(_densidad-desiredDensity)/5)*InitialInterval;
    }
    if (_densidad<DesiredDensity-40)
    {
        delta=(double)(_densidad-desiredDensity)/16)*InitialInterval;
    }
    if (_flujo>RequiredFlow+1.2)
    {
        delta=(double)(desiredFlow-_flujo)*InitialInterval;
    }
    if (_flujo<=RequiredFlow-1.8)//Aumentar el tiempo ya que pocos carros estan pasando
    {
        delta=(double)(desiredFlow-_flujo)*InitialInterval;
    }
}

private void ParserRulesDecision2(double _flujo,int _densidad)
{
    int NewPolicy=CurrentPolicy;
    if (oppositeDensity>_densidad)
    {
        NewPolicy=1;
    }
    if (oppositeDensity==0 && oppositeFlow==0)
    {
        NewPolicy=3;
    }
    if (oppositeDensity==0 && oppositeFlow!=0)
    {
        NewPolicy=2;
    }
}
```

```

    }
    if (oppositeDensity<=_densidad)
    {
        NewPolicy=2;
    }
    if (oppositeDensity!=0 && oppositeFlow==0)
    {
        NewPolicy=1;
    }
    if (oppositeDensity!=0 && oppositeFlow!=0)
    {
        NewPolicy=0;
    }
}

private void ParserRulesDecision3(double _flujo,int _densidad)
{
    String _label;
    double _delta;
    String s_message=getMessage(_flujo,_densidad);
    if (_label=="O") //link es lado oeste
    {
        LookupAgent(currentAgent+getNeighbors("O"));
        Start_Dialog();
        SendInformation(s_message);
    }
    if (_label=="N") //link es lado Norte
    {
        LookupAgent(currentAgent+getNeighbors("N"));
        Start_Dialog();
        SendInformation(s_message);
    }
    if (_label=="S") //link es lado Sur
    {
        LookupAgent(currentAgent+getNeighbors("S"));
        Start_Dialog();
        SendInformation(s_message);
    }
    if (_label=="E") //link es lado Este
    {
        LookupAgent(currentAgent+getNeighbors("E"));
        Start_Dialog();
        SendInformation(s_message);
    }
}

private void ParserRulesDecision4(double _flujo,int _densidad)
{
    if (Flow<RequiredFlow && OppositeFlow>RequiredFlow)
    {
        UpdateFutureDensity(IncreaseDensity(_density-DesiredDensity));
        UpdateFutureFlow(DecreaseFlow(_flujo-RequiredFlow));
    }
    if (Flow>RequiredFlow && OppositeFlow<RequiredFlow)
    {
        UpdateFutureDensity(DecreaseDensity(_density-DesiredDensity));
        UpdateFutureFlow(IncreaseFlow(_flujo-RequiredFlow));
    }
    if (Flow>RequiredFlow && OppositeFlow>RequiredFlow)
    {
        UpdateFutureDensity(IncreaseDensity(_density+DesiredDensity));
        UpdateFutureFlow(DecreaseFlow(_flujo+RequiredFlow));
    }
    if (Flow<RequiredFlow && OppositeFlow<RequiredFlow)
    {
        UpdateFutureDensity(DecreaseDensity(_density-DesiredDensity));
        UpdateFutureFlow(IncreaseFlow(_flujo-RequiredFlow));
    }
    if (oppositeDensity>DesiredDensity && oppositeFlow>RequiredFlow)

```



```

    {
        UpdateFutureDensity(IncreaseDensity(0));
        UpdateFutureFlow(DecreaseFlow(0));
    }
    if (oppositeDensity>DesiredDensity && oppositeFlow<RequiredFlow)
    {
        UpdateFutureDensity(IncreaseDensity(_density-DesiredDensity));
        UpdateFutureFlow(DecreaseFlow(_flujo-RequiredFlow));
    }
    if (oppositeDensity<DesiredDensity && oppositeFlow>RequiredFlow)
    {
        UpdateFutureDensity(DecreaseDensity(_density+DesiredDensity));
        UpdateFutureFlow(IncreaseFlow(_flujo+RequiredFlow));
    }
    if (oppositeDensity<DesiredDensity && oppositeFlow<RequiredFlow)
    {
        UpdateFutureDensity(DecreaseDensity(_density-DesiredDensity));
        UpdateFutureFlow(IncreaseFlow(_flujo-RequiredFlow));
    }
    if (oppositeDensity<=_densidad)
    {
        UpdateFutureDensity(DecreaseDensity(_density-DesiredDensity));
        UpdateFutureFlow(IncreaseFlow(_flujo-RequiredFlow));
    }
    if (oppositeDensity!=0 && oppositeFlow==0)
    {
        UpdateFutureDensity(DecreaseDensity(_density-DesiredDensity));
        UpdateFutureFlow(IncreaseFlow(_flujo+RequiredFlow));
    }
    if (oppositeDensity!=0 && oppositeFlow!=0)
    {
        UpdateFutureDensity(IncreaseDensity(_density-DesiredDensity));
        UpdateFutureFlow(DecreaseFlow(_flujo-RequiredFlow));
    }
}

```

Modelo

La política de control es representada por medio de un modelo de red de Petri temporizada. La estrategia de control establecida incluye la secuencia de las fases involucradas en la intersección y el tiempo asignado a cada fase. El marcado en la red de Petri indica la fase que se encuentra activa en ese instante y el retardo asignado a los lugares representa el tiempo que desplegara luz verde el semáforo para esa fase. El siguiente es el código utilizado para la implementación de la red de Petri; en el podemos distinguir los elementos que constituyen la red.

```

public class PN{
    public int NP;//Numero de lugares
    public int NT;//Numero de transiciones
    public double time; //Tiempo de la red de Petri.(Control)
    public Transition transarray[];//Arreglo de las transiciones pertenecientes a la red de Petri
    public Place placearray[];//Array of places
    public double pretimed;

    public PN()
    {
        NP=0;
        NT=0;
        transarray=new Transition[130]; //Si se pasa de 30 lo redimensionamos
    }
}

```

```

        placearray=new Place[150]; //If placearray lenght is greater than 50 then we resize it
    }

    public void setTime(double _time)

    public void addPlace(String namePlace,double _time,int _tokens)

    public void addTransition(String nameTransition,double _time)

    public int getPlace(String _label)

    public int getTransition(String _label)

    public boolean isEnabled(Transition T)

    public int[] getMarcado()

    public int[][] getIncidenceMatrix(){
        int PRE[][]=new int[NT][NP];
        int POST[][]=new int[NT][NP];
        int Matrix[][]=new int[NT][NP];
        /**We get the PRE*/
        for (int i=0; i<NT;i++) //De la transicion 1 a la ultima
            for (int j=0;j<NP;j++) //Del lugar 1 al ultimo
                {
                    PRE[i][j]=transarray[i].getInputArc(placearray[j]);
                    System.out.println("Transicion "+transarray[i].label+" Lugar"+placearray[j].label+" "+PRE[i][j]);
                }
        /**Now we get the POST*/
        for (int i=0; i<NT;i++) //De la transicion 1 a la ultima
            for (int j=0;j<NP;j++) //Del lugar 1 al ultimo
                {
                    POST[i][j]=transarray[i].getOutputArc(placearray[j]);
                }
        /**Now we get the Incidence Matrix*/
        for (int i=0; i<NT;i++) //De la transicion 1 a la ultima
            for (int j=0;j<NP;j++) //Del lugar 1 al ultimo
                {
                    Matrix[i][j]=POST[i][j]-PRE[i][j];
                }
        return Matrix;
    }

    public int[] getFireVector()
    {
        //Fire transitions enabled in transarray are 1 an others are 0
        int vector[]=new int[NT];
        for (int i=0;i<NT;i++)
            {
                if (isEnabled(transarray[i]))
                    {
                        vector[i]=1;
                        transarray[i].enableTransition=true;
                    }
                else
                    {
                        vector[i]=0;
                        transarray[i].enableTransition=false;
                    }
            }
        return vector;
    }
    /**We get the new Marked when T is fired*/
    public int[] FireT()
}

```

Operador del semáforo

El operador del semáforo es el módulo que se encarga de ejecutar la política de control establecida, representada por una red de Petri temporizada. Dada la naturaleza concurrente de los agentes de control, la operación cada agente está asignada a un hilo, dado que todos los agentes de control ejecutan simultáneamente sus políticas de control. El operador de tráfico tiene la función de traducir el disparo de las transiciones en la red de Petri a eventos de cambio de luz significativos para la FEL. El siguiente código representa la implementación del operador de tráfico.

```
private void startPetriNet(boolean isFirst,int NumPN){
    int marcado[]=new int[controller[NumPN].NP];
    //Boolean variable isFirst is to assign 1 more trigger to transitions

    AddInf ObjEv[]; //Mensajes enviados al simulador.
    ObjEv=new AddInf[1000000];
    int evnumber=0;
    int NumberOfTriggers;
    int NumberOfCiclos;//Cuando se dispara por primera vez la RP se dispara dos veces
    currentPN=NumPN;
    //while (controller.time<Ciclo)
    if (isFirst)
    {
        NumberOfTriggers=controller[NumPN].NT+1;
        NumberOfCiclos=2;
        for (int h=0;h<8;h++)
        { /*first transition*/
            if (controller[NumPN].transarray[0].semaphore[h][0]!=0)
            {
                mainSemaphore=controller[NumPN].transarray[0].semaphore[h][0];
                h=8;
            }
        }
    }
    else
    {
        NumberOfTriggers=controller[NumPN].NT;
        NumberOfCiclos=1;
    }
    for (int t=0;t<NumberOfCiclos;t++)
    for (int k=0;k<NumberOfTriggers;k++)//Number of transitions
    {
        for (int i=0;i<controller[NumPN].NT;i++)
        {
            if (controller[NumPN].transarray[i].enableTransition)
            {
                for (int j=0;j<8;j++) //Los eventos que dispara
                {
                    if (controller[NumPN].transarray[i].semaphore[j][0]!=0 )
                    {
                        /* An event is programmed at FEL and is not an ambar*/
                        if (controller[NumPN].transarray[i].semaphore[j][0]==mainSemaphore &&
controller[NumPN].transarray[i].semaphore[j][1]!=3)
                        {
                            NumberOfCLEsended++;
                        }
                        ObjEv[evnumber]=new
AddInf(controller[NumPN].transarray[i].semaphore[j][0],3,controller[NumPN].preimed,controller[NumPN].transarray[i].semaphore[j][1
],0,1);
                        c3ddispatcher.addEvent(ObjEv[evnumber]);
                    }
                }
            }
        }
    }
}
```

```

        System.out.println(ObjEv[evnumber].EventTime+"
"+ObjEv[evnumber].ObjectId+" "+ObjEv[evnumber].EventType+" "+ObjEv[evnumber].DesiredStatus);
        evnumber++;
    }
}
controller[NumPN].FireT();
}
}

```

Middleware ProActive

La integración del sistema de control al simulador de tráfico urbano implica un incremento en la complejidad del simulador. El sistema actual permite que la simulación esté distribuida en diferentes host; por esta razón el sistema de control puede estar asignado a un host diferente al de los host asignados al simulador. La distribución de los módulos puede realizarse a través de un middleware que permita una comunicación eficiente entre los diferentes módulos involucrados en la simulación.

El middleware ProActive [INRIA, 2006] es un conjunto de bibliotecas en Java para computación paralela, distribuida y concurrente; también se caracteriza por proveer movilidad y seguridad en un marco uniforme. Con un conjunto reducido de primitivas, ProActive permite programar aplicaciones que son distribuidas en una red de área local, en un cluster de ordenadores o en grids de Internet. ProActive está hecho solo con clases estándar de Java, y no requiere cambios en la máquina virtual. Debido a que ProActive ofrece muchas facilidades y ventajas con respecto a los demás middlewares analizados, se ha decidido implementar la interacción entre los módulos del simulador de tráfico urbano a través del middleware ProActive.

4.2.2. Interacción con la máquina de simulación

El agente de control y la máquina de simulación interactúan durante todo el tiempo de la simulación; esto indica que la información fluye constantemente de un módulo a otro. La máquina de simulación envía a los agentes de control información referente a las condiciones de tráfico, y los agentes de control envían información referente a los cambios de luz que deben ser programados en la FEL. Debido a que generalmente la máquina de simulación y los agentes de control están ubicados en host diferentes es necesario definir la manera en que esta información fluye de un host a otro.

El simulador está diseñado de tal forma que la máquina de simulación es un despachador (dispatcher) al que se conectan los agentes de control, los cuales pueden estar en diferentes hosts. Inicialmente la máquina de simulación espera a que los agentes de control se conecten a ella (figura 4.7); una vez que los agentes de control se han conectado, estos envían los primeros eventos de cambio de luz para que la simulación inicie. Durante el transcurso de la simulación, la máquina de simulación envía las condiciones de tráfico actuales a los agentes involucrados en la simulación; con la información recibida los agentes de control analizan la información y envían eventos de cambio de luz en forma de mensajes hacia la máquina de simulación, para ser programados en la FEL.

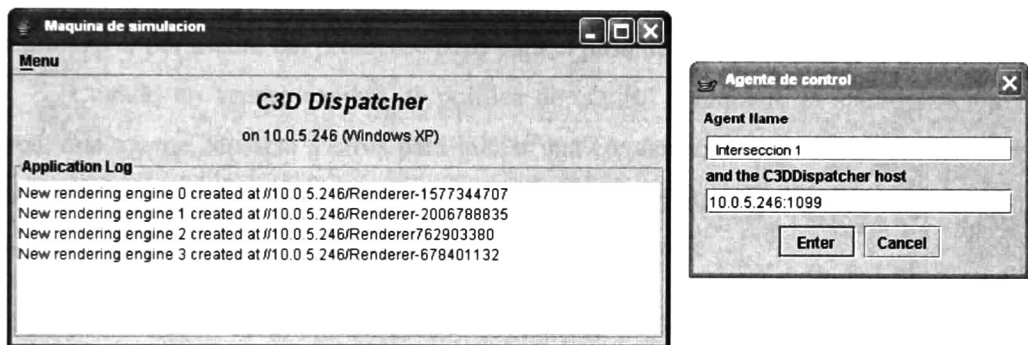


Figura 4.7 Interacción con la máquina de simulación

Los mensajes que envía el agente de control hacia la máquina de simulación son enviados a través de la primitiva `addEvent()` propia de ProActive que se basa en el protocolo RMI (Remote Method Invocation). Los mensajes enviados por el agente de control contienen la estructura del evento de cambio de luz. El siguiente código es la implementación de la clase evento en la que pueden distinguirse las propiedades de los eventos que son enviados por los agentes de control.

```
public class Evento implements Serializable
{
    public int ObjectId;//Id del semaforo
    public int EventType;//Tipo de evento 3 es cambio de luces
    public double EventTime;//Tiempo del evento
    public int DesiredStatus;//0 rojo, 1 verde derecho, 2 verde flecha
    public int EventPositionX;
    public int EventPositionY;

    public Evento(int Obj,int Tipo,double Tiempo,int Estado,int X,int Y){
        ObjectId=Obj;
    }
}
```



```

EventType=Tipo;
EventTime=Tiempo;
DesiredStatus=Estado;
EventPositionX=X;
EventPositionY=Y;

```

```

}
}

```

4.2.3. Comunicación entre los agentes de control

Un agente debe informar a sus agentes vecinos sobre cualquier cambio realizado en sus estrategias de control; además es posible que un agente solicite a un subconjunto dado de agentes que se sincronicen para establecer determinada política de control. Todo este flujo de información entre los agentes involucrados en la simulación es realizado a través de ProActive por medio del protocolo RMI para el paso de mensajes.

Cuando un agente cambia de política de control o requiere la sincronización con otros, este agente contacta a otros para iniciar una conversación; en el capítulo anterior se describieron los métodos para la comunicación entre agentes. El siguiente código es un fragmento de la implementación de algunos de estos métodos.

```

private void setCoordination(String s_message)
{
    String _label;
    double _delta;
    double[] involvedphases={0,0,0,0,0,0,0,0,0,0}; //Hasta 10 fases involucradas que afectan a esta interseccion.
    double backtime;
    String[] result=s_message.split(",");
    _label=result[0];
    _delta=Double.valueOf(result[1]).doubleValue();
    if (_label=="O") //link es lado oeste
    {
        involvedphases=getLinkPhases(3,7);
        //3 es fase OE, 7 es fase WN
    }
    else
    {
        if (_label=="N") //link es lado Norte
        {
            involvedphases=getLinkPhases(0,4);
            //NS,NE
        }
        else
        {
            if (_label=="S") //link es lado Sur
            {
                involvedphases=getLinkPhases(1,5);
                //SN,SO
            }
            else //link es lado Este _label=="E"
            {
                involvedphases=getLinkPhases(2,6);
                //EO,ES
            }
        }
    }
}

```

```

    }
}
for (int i=0;i<NumberOfFases; i++)
{
    if (involvedphases[i]!=0)//Quiere decir que el link involucra esta fase
    {
        backtime=involvedphases[i];
        if (involvedphases[i]+_delta>MinimumGreenTime && involvedphases[i]+_delta<MaximunGreenTime)
        {
            involvedphases[i]=involvedphases[i]+_delta;
        }
        else
        {
            if (involvedphases[i]+_delta<=MinimumGreenTime)
                involvedphases[i]=MinimumGreenTime;
            else
                involvedphases[i]=MaximunGreenTime;
        }
    }
}
}

private void CoordinarAgentes(int _link,int _opplink,String xlabel,double _flujo,double _densidad)
{
    //Esta es la interseccion uno por lo cual puede coordinarse con la interseccion 2.
    /*Si entra en esta funcion implica que hay mas de 130 carros en la carretera
    * luego necesita coordinarse aumentar el tiempo de la fase en la interseccion (i+1)
    * y disminuir el tiempo de fase en la interseccion (i-1)
    */
    int nextIntersection;
    int antIntersection;
    double deltanext=0;
    double deltaant=0;
    String _Antmessage="";
    String _Nextmessage="";
    if (xlabel=="O" || xlabel=="E")
    {
        nextIntersection=convertToIntersection(getLinkIntersection(_opplink,true));
        antIntersection=convertToIntersection(getLinkIntersection(_link,false));
        //Debemos tomar en cuenta que antIntersection y nextIntersection
        //No son Intersection 1, Intersection 2...Debemos hacer una funcion
        //Que los convierta por ejemplo 19= intersection 1
        deltanext=(double)((_densidad-desiredDensity)+(_flujo-desiredFlow)*10)/7;//Unidades de tiempo
        deltaant=deltanext*-1;
        _Nextmessage=xlabel+" "+deltanext;//Debe tomar en cuenta la coordenada(N,S,E,O) y la delta
        _Antmessage=xlabel+" "+deltaant;
        if (nextIntersection!=0) //Existe la interseccion
        {
            /*coordinarse para la siguiente interseccion.*/
            Integer talkId1 = (Integer) h_users.get("Intersection "+nextIntersection);
            gui.writeMessage("<to " + "Intersection "+nextIntersection + "> " + _Nextmessage + "\n");
            c3ddispatcher.userWriteMessage(talkId1.intValue(),_Nextmessage);
        }
        if (antIntersection!=0)
        {
            /*coordinarse para la Interseccion anterior.*/
            Integer talkId2 = (Integer) h_users.get("Intersection "+antIntersection);
            gui.writeMessage("<to " + "Intersection "+antIntersection + "> " + _Antmessage + "\n");
            c3ddispatcher.userWriteMessage(talkId2.intValue(),_Antmessage);
        }
    }
    else
    {
        log("There is not Intersection to set coordination");
    }
}
}

```

4.3. Conclusiones

En este capítulo hemos presentado un conjunto de clases y métodos que permiten la construcción de un sistema multiagente para el control de tráfico urbano. El agente de control es implementado en el lenguaje de programación Java; para la construcción del agente de control presentamos la implementación de sus componentes principales. También definimos la forma en que el agente de control interactúa con el simulador. Se definió el middleware ProActive como el mecanismo de comunicación entre los agentes.

En el siguiente capítulo presentamos los resultados de las pruebas realizadas en el laboratorio, que demuestran que nuestro sistema de control está trabajando eficientemente de acuerdo a los requerimientos previos.

Capítulo 5

Análisis de resultados y pruebas

Resumen: En este capítulo presentamos las pruebas realizadas al integrar el sistema de control al simulador. Se realiza un análisis de los datos obtenidos y de esta manera se comprueba la eficiencia del sistema de control.

5.1. Introducción

Las pruebas en un sistema de control de tráfico urbano consisten en observar el comportamiento del sistema de control de tráfico urbano dadas ciertas condiciones de tráfico en una red de tráfico.

En este trabajo se integra al simulador el módulo de control de tráfico urbano. Se realizan varias corridas del simulador con diferentes condiciones de tráfico, y a través del análisis de los resultados puede notarse como el controlador de tráfico se va adaptando de acuerdo a la demanda.

5.2. Pruebas y simulaciones realizadas

Las simulaciones forman parte de esta etapa y de acuerdo a las diferentes corridas podemos analizar los resultados del sistema de control. Se realiza la simulación incorporando el sistema de control de tráfico urbano en una red de tráfico que contiene cuatro intersecciones, cada intersección esta compuesta por cuatro segmentos de calle de 1 Km. de longitud cada segmento (ver figura 5.1). Los parámetros correspondientes a la velocidad de los vehículos, la distribución aleatoria de llegada de vehículos, el tiempo inicial asignado a las fases, entre otros son modificados para observar el comportamiento del sistema de control de acuerdo a las condiciones de tráfico. Los siguientes pasos son realizados al realizar una corrida del sistema.

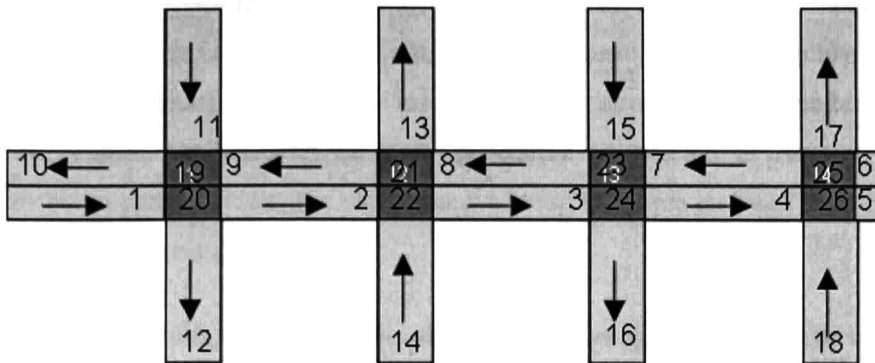


Figura 5.1 Estructura de la red de tráfico.

- Conexión de los agentes a la máquina de simulación.** Debido a que los agentes trabajan de manera concurrente, es necesario que estos le indiquen a la máquina de simulación cuando debe iniciar su ejecución. Los parámetros referentes a la frecuencia de llegada de los vehículos, y la velocidad de los mismos son provistos por la máquina de simulación y pueden ser modificados en el código fuente de la misma. En la figura 5.2 se ilustra una corrida del simulador con el sistema de control integrado.

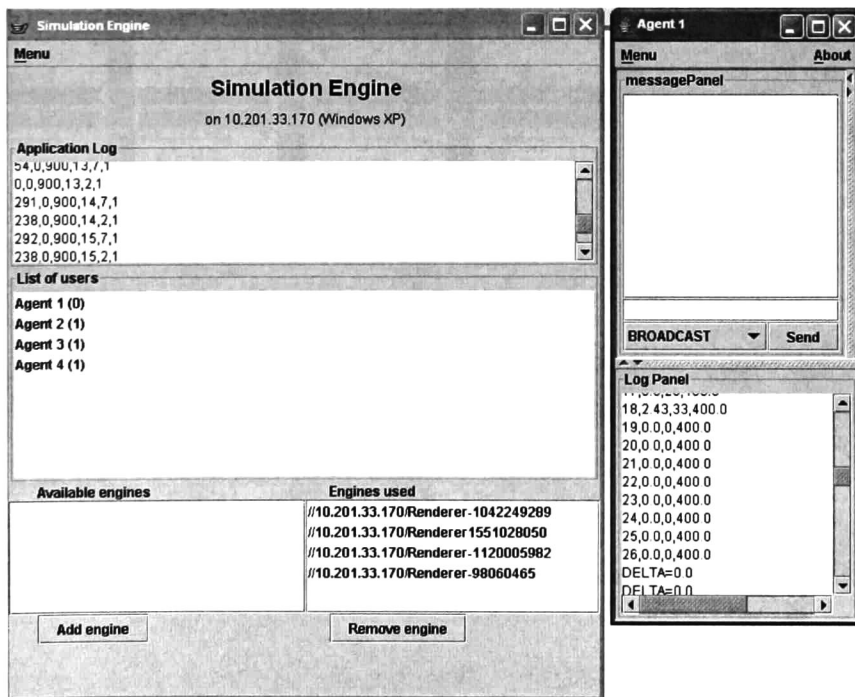


Figura 5.2 Corrida del simulador

La secuencia de las fases y el tiempo asignado a las mismas cambia dinámicamente durante el transcurso de la simulación. Cada agente de control genera dos archivos de texto los cuales son requeridos para analizar los resultados y observar el comportamiento del sistema de control; los archivos de texto generados se refieren al flujo y densidad observados en un periodo de tiempo específico y a la forma en que las fases se adaptan a las condiciones de tráfico observadas.

• **Visualización del comportamiento del sistema.** Una vez que los agentes de control han generado los archivos correspondientes a la simulación, es posible presentar gráficamente el comportamiento del sistema. En la figura 5.3 podemos observar como se forman las colas en las calles involucradas en la red de tráfico y a medida que avanza la simulación, las fases se van adaptando, lo que se refleja en la reducción de las colas y el incremento en el flujo de tráfico.

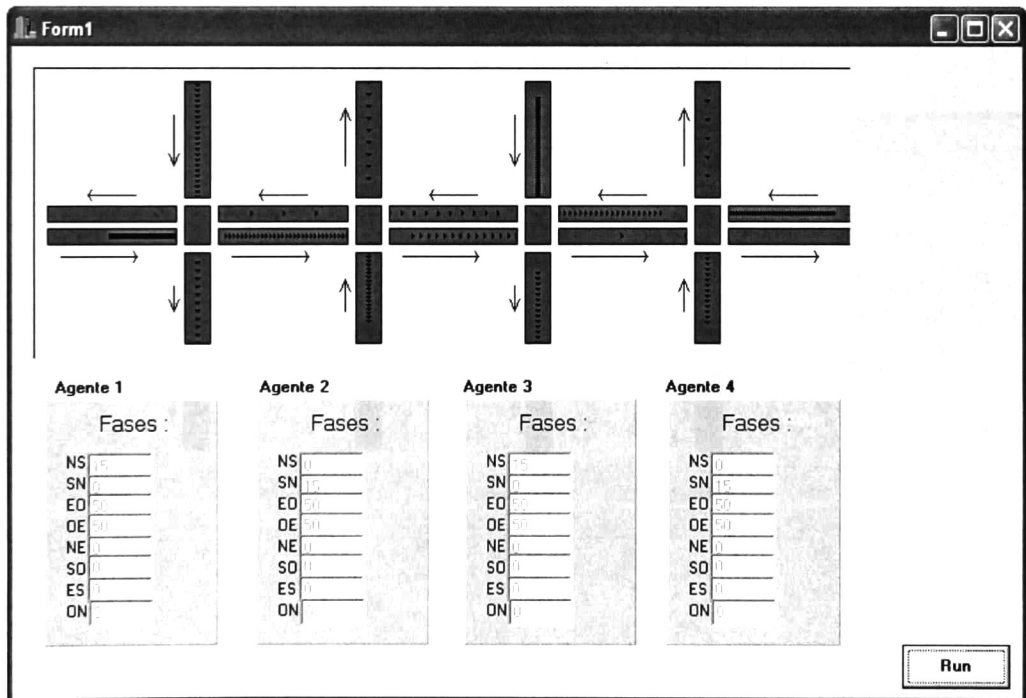


Figura 5.3 Inicio de la simulación

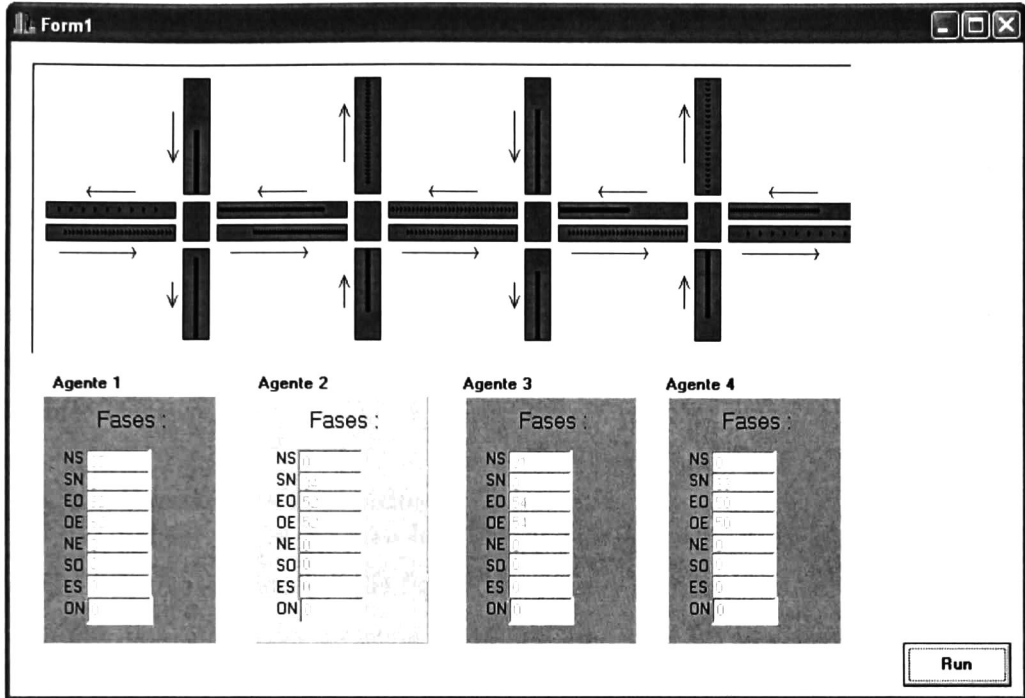


Figura 5.4 Actualización de los tiempos asignados a las fases

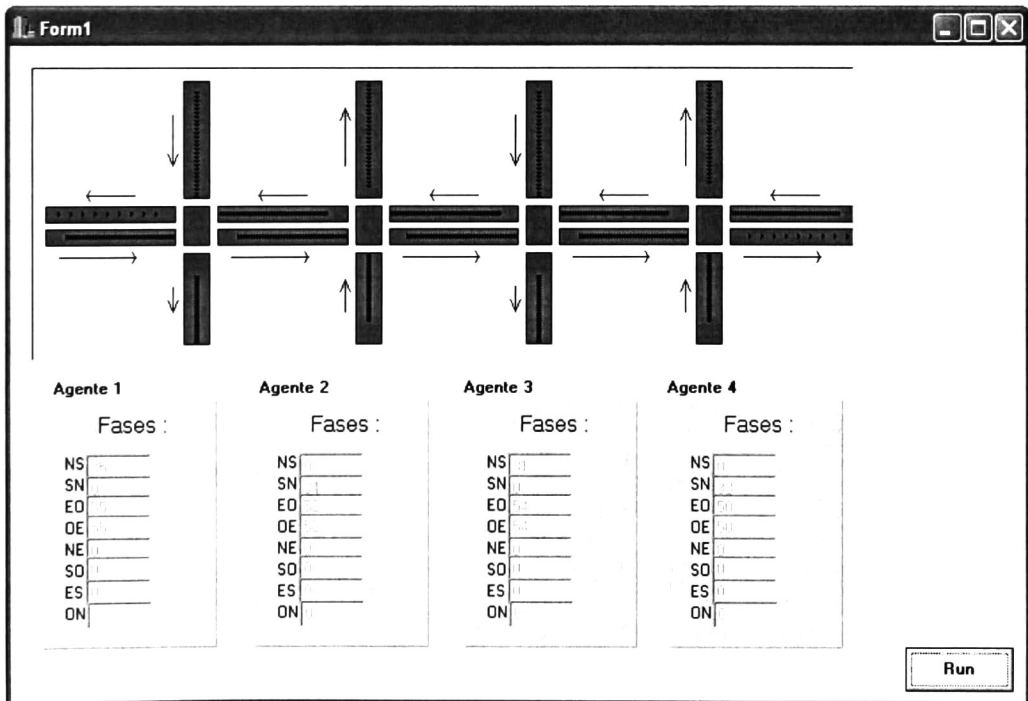


Figura 5.5 Fin de la simulación

Puede notarse al inicio de la simulación (figura 5.3) que en algunas calles se forman colas más largas y en otras casi no hay vehículos en las colas; esto es debido a que en primera instancia los tiempos asignados a las fases son fijos y que apenas empiezan a fluir los vehículos en la red de tráfico. En la figura 5.4 observamos que el tiempo asignado a las fases ha cambiado, lo que implica que el flujo y la densidad sean casi constantes en todas las calles. En la figura 5.5 podemos notar de acuerdo a la longitud de las colas, que cuando la simulación ha finalizado, el flujo y la densidad en todas las calles se comporta de forma constante, lo cual nos indica que el sistema de control está funcionando de manera eficiente.

5.3. Resultados obtenidos

En esta sección se analizan los resultados de aplicar al simulador una política de control adaptativa (actuated). En la figura 5.6 se ilustra el comportamiento del simulador de tráfico que implementa un esquema de control fijo; en estas gráficas podemos notar el tráfico observado para los carriles 1 y 11 respectivamente. Como puede notarse para el esquema de control fijo los valores de flujo y densidad tienen un comportamiento inconstante, ya que hay tiempos en donde se reduce la densidad y aumenta el flujo pero hay otros tiempos en donde la densidad aumenta considerablemente y el flujo disminuye. En la figura 5.7 vemos las gráficas de flujo y densidad para la calle 1 y la calle 11 respectivamente. En estas gráficas se sigue una política de control en donde los tiempos asignados a las fases cambian dinámicamente de acuerdo a las condiciones de tráfico observadas; las líneas rojas indican que en ese instante de tiempo se modificaron los tiempos asignados a las fases involucradas en esas calles. En las gráficas podemos notar varios estados: al principio la densidad es alta y el flujo es bajo; entonces conforme el tiempo de simulación avanza, la densidad se reduce y tiende a ser constante y el flujo se incrementa y también tiende a ser constante.

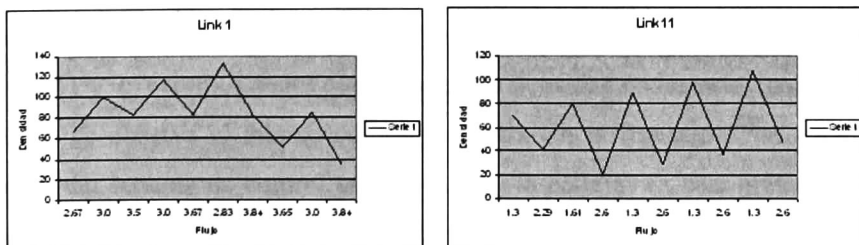


Figura 5.6. Esquema de control fijo para los carriles 1 y 11

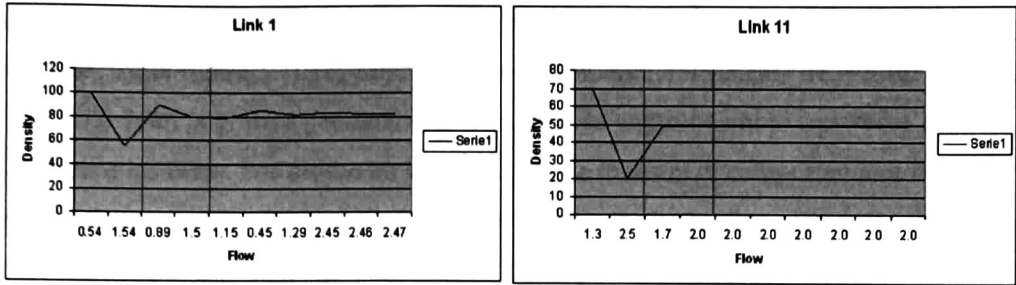


Figura 5.7 Actualización de tiempos para la calle 1 y la calle 11

En la figura 5.8 se muestra una política de control establecida para las calles 1 y 11, en la cual además de que los tiempos asignados a las fases pueden cambiar dinámicamente, el modelo de control puede ser reemplazado por un modelo más adecuado a las condiciones de tráfico dadas. La línea verde en la gráfica denota que se ha reemplazado el modelo de control actual por un modelo que se adapte mejor a las condiciones de tráfico dadas; las líneas rojas representan que los tiempos asignados a las fases involucradas en el modelo de control han cambiado. Como podemos observar en las gráficas después de reemplazar las políticas de control (línea verde) la densidad disminuye y el flujo se incrementa, pero el flujo y la densidad son variables, de ahí el comportamiento en pico de la gráfica; sin embargo, al cambiarse los tiempos asignados a las fases después de un tiempo transcurrido el flujo de tráfico y la densidad tienden a ser constantes.

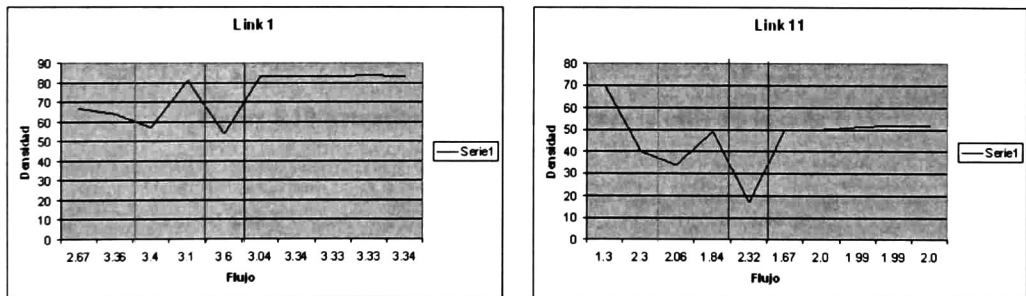


Figura 5.8 Reemplazo de política de control para la calle 1 y la calle 11

Hemos visto el comportamiento del sistema de control en la intersección compuesta por las calles 1, 11 y 9. Veremos ahora que el flujo y la densidad en las principales calles de la red de tráfico se observa el mismo comportamiento. En la figura 5.9 se muestra el comportamiento del sistema de control en la calle 2 y la calle 8; en esta gráfica podemos observar que el sistema de control modificó tres veces el tiempo asignado a las fases que

involucran estas calles y al final, el flujo de vehículos y la densidad de tráfico se mantuvo constante. En la figura 5.10 se ilustra el comportamiento del sistema de control en la calle 3 y la calle 7; en la gráfica notamos que el sistema de control cambió tres veces los tiempos asignados a las fases. Por último en la figura 5.11 se muestra el comportamiento del sistema de control para la calle 4 y la calle 18; como vemos, en esta gráfica solo se modificaron dos veces los tiempos de las fases. En todos los casos podemos notar que el sistema de control realizó los ajustes necesarios para que el flujo y la densidad sean equilibrados.

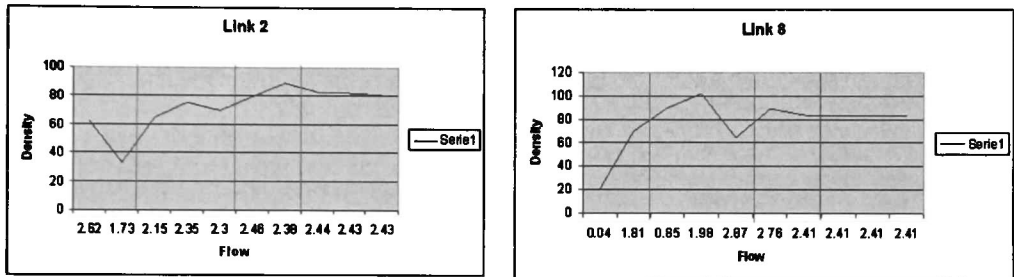


Figura 5.9 Actualización de tiempos para la calle 2 y la calle 8

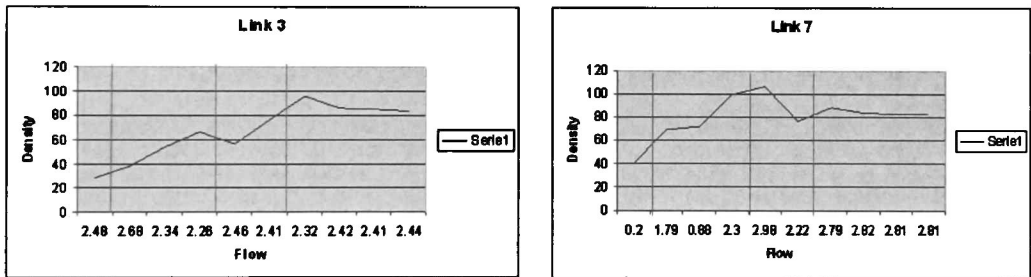


Figura 5.10 Actualización de tiempos para la calle 3 y la calle 7

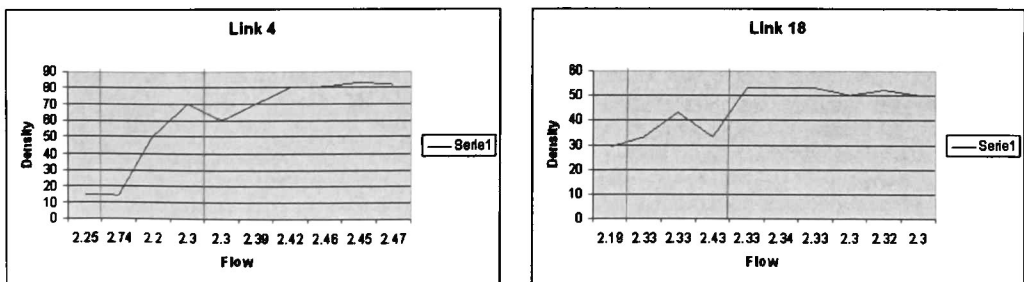


Figura 5.11 Actualización de tiempos para la calle 4 y la calle 18

5.4. Conclusiones

En este capítulo se han presentado las pruebas realizadas al integrar el sistema de control al simulador de tráfico urbano. Se ha demostrado que los agentes de control proveen un mecanismo para la administración del tráfico urbano en una red de tráfico urbano.

Asimismo pudimos corroborar que el sistema de control está trabajando eficientemente ya que en las gráficas presentadas puede observarse que el sistema de control se adapta a la demanda de tráfico.

Conclusiones

En esta tesis se presentó un esquema de control de tráfico urbano basado en sistemas multiagentes. En nuestro trabajo se propuso una arquitectura de agentes en donde el sistema establece una estrategia de control en una red de tráfico urbana. Cada cruceo es controlado por un agente de control especializado el cual está encargado de seleccionar una estrategia de control de acuerdo a las condiciones de tráfico en la intersección.

El controlador es un agente de control proactivo que permite predecir cuales serán las condiciones de tráfico en un instante de tiempo futuro y seleccionar la estrategia de control adecuada a las condiciones de tráfico observadas. El agente de control aplica una política de control pertinente y le envía a la máquina de simulación el comportamiento de los semáforos en forma de eventos de cambio de luz. La adaptación del agente a nuevas condiciones de tráfico es realizado por el cambio en las políticas de control. Este cambio de las políticas de control puede ser establecido de dos maneras: a) actualizando el tiempo asignado a la política actual, o b) reemplazando la política actual por una más adecuada. Es posible definir políticas de control complejas que incluyen más de una secuencia de fases y que alteran la operación de las mismas por el disparo de eventos externos que representan, petición de peatones, diferentes condiciones de tráfico, vehículos de prioridad, entre otras. Nuestra arquitectura de agente permite la coordinación con otros agentes de control asignados a otras intersecciones. Esto significa que si un agente percibe saturación en alguna calle que pertenece a la intersección, este agente puede decidir interactuar con algunos agentes asignados en intersecciones vecinas para sincronizar alguna estrategia de control.

El agente de control ha sido desarrollado de acuerdo a la arquitectura propuesta usando Java. La interacción con otros componentes del simulador es realizada a través de objetos activos en ProActive. Como puede notarse, durante las simulaciones realizadas los índices de rendimiento de la red de tráfico han sido mejorados considerablemente cuando las estrategias de control adaptativo se han establecido en las políticas de control.

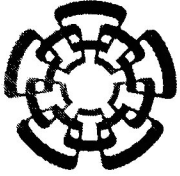
Actualmente se está trabajando en a) proponer políticas de control más complejas que respondan a eventos externos (peatones, vehículos de prioridad), y b) que los agentes puedan proporcionar en tiempo real la definición de las políticas de control.

BIBLIOGRAFIA

- [Arámburo, 2005] J. Aramburo. "Diagnóstico Distribuidos de Faltas en Sistemas de Eventos Discretos utilizando Modelos Reducidos" Tesis de maestria. CINVESTAV Unidad Guadalajara. 2005.
- [Buton, 2001] K.J. Button and D.A. Hensher. "Handbook of transport systems and traffic control". October 2001.
- [Carbone, 2004] Carbone Ciro "Control Oriented Discrete Event Models for Manufacturing and Transport Systems" Phd Thesis, Facoltà di Ingegneria Elettrica. 2003-2004.
- [Cetin, 2004] G.F. Cetin, M. "Modeling traffic signal control using Petri nets". Intelligent Transportation Systems, IEEE Transactions on, Sept 2004, pp 177-187.
- [Chantaraskul,2005] Chantaraskul Soamsiri. "An intelligent-agent approach for Managing congestion in W-CDMA Networks". Phd Thesis. Departament of Electronic Engineering Queen Mary, University of London, August 2005.
- [DiFebbraro, 2004] DiFebbraro Angela, Giglio D, Sacco. "Urban Traffic Control Structure Based on Hybrid Petri Nets". Intelligent Transportation Systems, IEEE transactions on, Dec 2004, pp 224-237.
- [Dresner, 2004] Kurt Dresner, Peter Stone. "Multiagent Traffic Management: A Reservation-Based Intersection Control Mechanism". AAMAS'04, July 19-23, 2004, New York, New York, USA.
- [Dresner, 2005] Dresner Kurt, Stone Peter. "Turning the corner: Improved Intersection Control for Autonomous Vehicles". *The 2005 IEEE Intelligent Vehicles Symposium*, June 2005.
- [Dresner, 2005] Dresner Kurt, Stone Peter. "Multiagent Traffic Management: An improved Intersection Control Mechanism" AAMAS 05, July 25-29, 2005.
- [Fujimoto, 2002] Fujimoto Richard, Leonard II John. "Grand Challenges in Modelling and Simulating Urban Transportation Systems" .Technical Report. Georgia Tech Modelling & Simulation Research & Education Center (MSREC) and a college of Computing Georgia Institute of technology Atlanta, Georgia 30332. 2002.
- [Groot, 1997] R.M. De Groot. "Dynamic Traffic Control of Free-Navigating Automatic Guided Vehicles". University of Twente. The Netherlands Department of Computer Science. Master's Thesis . July 1997.
- [Helsgaun,2000] Helsgaun Keld. "Discrete Event Simulation in Java". Departament of computer Science. Roskilde University. DK-4000 Roskilde, Denmark.
- [Hewage, 2004] Hewage Kasun N.,Ruwanpura Janaka Y. "Optimization of Traffic Signal Light Timing Using Simulation" Preceedings of the 2004 Winter Simulation Conference.
- [Hoogendoorn, 2001] Hoogendoorn,S., and P.Bovy, "State of the art of vehicular traffic flow modelling", in special issue on road traffic modelling and control of the journal of systems and control eng. Proc. Of the IME I, 2001.
- [INRIA,2001] INRIA."ProActive Manual Version 3.0.1." <http://www-sop.inria.fr/oasis/proactive/>
- [Jones, 2004] Steven L. Jones , Jr and Andrew Sullivan. "Comparison of Three Traffic Simulation Packages for Analysis of access Management Techniques". 6th Access Management Conference. August 2004.
- [Julvez, 2001] Julvez Jorge, Boel René. "Modelling and Controlling Traffic Behaviour with Continuous Petri Nets" D.G.A. ref B106/2001 and a European Community Marie Curie Fellowship,CTS, contract number: HPMT-CT-2001-00278
- [Kaehler, 1993] Steven D. Kaehler Fuzzy. Logic Tutorial 1993. "http://www.seattlerobotics.org/encoder/mar98/fuz/flindex.html"

- [Loborg, 1994] P. Loborg. "Error Recovery in Automation – An overview" AAAI-94 Symposium on Detecting and Resolving Errors in Manufacturing Systems. 1994.
- [López, 1997] E. López. "Introducción a las Redes de Petri". Facultad de Ciencias Físico-Matemáticas. Universidad de Nuevo León. 1997.
- [López, 2002] E.López. "Analysis of discrete event systems by Simulation of Timed Petri net models" Mathematics and Computers in Simulation 61 (2002) 53-59.
- [Mancinelli, 2005] Mancinelli Elina, Cohen guy, Gaubert Stéphane, Quadrat Jean Pierre, Rofman Edmundo. "On Traffic Light Control of Regular Towns". INRIA, Sept 2005.
- [Murat, 2002] Sazi Murat Yetis, Gedizlioglu Ergum. "A New Approach for Fuzzy Traffic Signal Control". The Scientific and Technical Research Council of Turkey with a number of INTAG 915, 13th Mini-Euro Conference, 2002.
- [Murata, 1989] Murata Tadao. "Petri Nets: Properties, Analysis and Applications". Proceedings of the IEEE, vol 77, NO. 4. April 1989.
- [Nagel, 1992] K. Nagel. "A Cellular automaton model for freeway traffic". J. Phys. I France 2 (1992) 2221-2229.
- [Nagel, 2003] K. Nagel. "Multi-agent transportation simulation". 2003, <http://www.sim.inf.ethz.ch/teach/traffic/>
- [Neri, 2005] Lopez Neri. "Micro-Simulación de Trafico Urbano Basada en Sistemas de Agentes Móviles". Master Degree Thesis 2005.
- [Patrick, 2001] Ehlert Patrick A.M. and Rothkrantz Leon J.M.. "Microscopic traffic simulation with reactive driving agents". 2001 IEEE Intelligent Transportation Systems Conference Proceedings, Oakland(CA), USA August 25-29 2001.
- [Riedel, 1993] Riedel Th., U. Brunner. "Traffic Control Using Graph Theory" 12th IFAC World Congress, 18.-23. Juli 1993, Sydney (Australien).
- [Rishiyur, 2003] Rishiyur S. Nikhil. "A (Midly) Intelligent Traffic Light System (An introductiony Bluespec Tutorial)" Sandsbur Corporation. January 13, 2003.
- [Rodrigues, 2004] Rodrigues L. "Continuous Time Sthocastic Models for Vehicular Traffic on Highways". Brazilian journal of Physics, vol 34, no 2A. June, 2004.
- [Roozmond, 2000] Danko A. Roozmond , Jan L.H. Rogier. "Agent controlled traffic lights". ESIT 2000, 14-15 September 2000, Aachen, Germany.
- [Sanal, 1999] Sanal R.P. "Intelligent Traffic Signals Using Fuzzy Logia Control" 11th Kerala Science Congress, Idukki, Kerala, India 1999.
- [Schmidt, 1994] C. Schmidt Douglas, Tatsuya Suda. "A Framework for Experimenting with High-Performance Transport System Process Architectures" Technical Report. Departament of Information and Computer Science University of California. 1994.
- [Sivasankaran] Sivasankaran Vinod. "An Adaptative Network for Traffic Signal Synchronization Using Distributed Computing". University Of Rochester. NY – 14627.
- [Skabardonis, 1998] Skabardonis Alexander, Bertini Robert L., Gallager Brian R.: "Development and Application of Control Strategies for Signalized Intersections in Coordinated Systems". For publication Trasportation Research Record, March 1998.
- [Szpirka, 2004] Szpyrka Marcin, Szmuc Tomasz. "New Time model and Design Method for RTCP-nets" WRIP 2004, september 6-8 2004, Istanbul Trukey.

- [TRAFFICWARE] Trafficware Corporation. "Synchro Plus SimTraffic 6". www.trafficware.com.
- [Wang, 2005] Fei-Yue Wang. "Agent-Based Control for Networked Traffic Management Systems". IEEE Computer Society September 2005. Vol 20, No. 5. pp 92-96.
- [Wang, 2005] Fei-Yue Wang. "Rhodes to Intelligent Transportation Systems". IEEE Computer Society Enero/Febrero 2005. Vol 20, No. 1. pp 10-15.
- [Wiering, 2004] Marco Wiering, Jelle Van Veenen, Jilles Vreeken, Arne Koopman. "Intelligent Traffic Light Control". Institute of information and computing sciences, Utrecht University Technical Report. UU-CS-2004-029
- [Wooldridge, 1999] Wooldridge Michael. "Intelligent Agents ". In G.Weiss, Editor: Multiagent Systems (MIT Press 1999).
- U.S. Department of transportation. "Developing Traffic Signal Control Systems Using the National ITS Architecture". Febraury 1998.
- National Transportation Planning and Research Center Thiruvananthapuram. "Intelligent Traffic Signals Using Fuzzy Logic Control". Preceedings oh the 11th Kerala Science Congress, Idukki, Kerala, India 1999.
- "http://www.ct.gov/dot/lib/dot/Documents/dpublications/Capacity_Analysis_&_Signal_Timing.pdf"



CENTRO DE INVESTIGACIÓN Y DE ESTUDIOS AVANZADOS DEL I.P.N. UNIDAD GUADALAJARA

El Jurado designado por la Unidad Guadalajara del Centro de Investigación y de Estudios Avanzados del Instituto Politécnico Nacional aprobó la tesis

Control de tráfico urbano basado en sistemas multiagentes

del (la) C.

Joel Antonio TREJO SANCHEZ

el día 29 de Septiembre de 2006.

Dr. Luis Ernesto López Mellado
Investigador CINVESTAV 3A
CINVESTAV Unidad Guadalajara

Dr. Félix Francisco Ramos Gorchado
Investigador CINVESTAV 2B
CINVESTAV Unidad Guadalajara

Dra. María Elena Meda Campaña
Profesor Titular
Universidad de Guadalajara CUCEA

Dr. Mario Angel Siller González
Pico
Investigador CINVESTAV
CINVESTAV Unidad Guadalajara



CINVESTAV
BIBLIOTECA CENTRAL



SSIT000008749