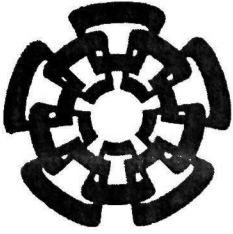


xx(122409.1)



CINVESTAV

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

Micro-Simulación de tráfico urbano basada en Sistemas de Agentes Móviles

**CINVESTAV
IPN
ADQUISICION
DE LIBROS**

Tesis que presenta:
Emmanuel López Neri

para obtener el grado de:
Maestro en Ciencias

en la especialidad de:
Ingeniería eléctrica

CINVESTAV I.P.N.
SECCION DE INFORMACION
Y

Directores de Tesis

Dr. Luis Ernesto López Mellado
Dr. Antonio Ramírez Treviño

Guadalajara, Jalisco, Agosto del 2005.

CLASIF.: TK165.58.167 2005
ADQUIS.: SSI-363
FECHA: 20-DIC-EM-2005
PROCED.: DON-2005

J.D. 121923-2001

Micro-Simulación de tráfico urbano basada en Sistemas de Agentes Móviles

**Tesis de Maestria en Ciencias
Ingeniería eléctrica**

Por:

Emmanuel López Neri

Ingeniero Electrónico en Computación

Centro de Enseñanza Técnica Industrial 1998-2003

Becario de CONACYT, expediente no. 180892

Directores de Tesis

Dr. Luis Ernesto López Mellado

Dr. Antonio Ramírez Treviño

CINVESTAV del IPN Unidad Guadalajara, Agosto del 2005.

Agradecimientos:

*A mis padres y hermanas por su apoyo incondicional,
Mis más sinceros agradecimientos a mis colegas y amigos.
A mis asesores,
Al **CONACYT** por su apoyo*

Resumen

Esta tesis aborda el problema de micro-simulación de sistemas de tráfico urbano (STU) siguiendo una metodología basada en agentes móviles. Se presenta una metodología para construir modelos jerárquicos y modulares de STU usando el formalismo basado en redes de Petri anidadas de 3 niveles, 3-LNS, donde el nivel más alto describe la red de tráfico; los modelos de nivel medio modelan al vehículo (agentes móviles) y otras entidades que se encuentran en la red (como los semáforos); el tercer nivel describe de manera detallada los procesos ejecutados por los agentes, por ejemplo los planes y los comportamientos del conductor. Los modelos jerárquicos y modulares nos llevan a obtener una arquitectura limpia orientada a agentes, la cual es implementada en JAVA. Se presenta además, el diseño de un simulador de tráfico urbano basado en sistemas multi-agente usando la metodología orientada a eventos, que nos permite obtener tiempos de ejecución de la simulación más rápidos.

Palabras clave: micro-simulación, eventos discretos, agentes móviles.

ÍNDICE GENERAL

ÍNDICE DE FIGURAS	IV
ÍNDICE DE TABLAS	V
PRESENTACIÓN	1
1 MODELADO Y SIMULACIÓN DE SISTEMAS DE TRÁFICO URBANO	3
1.1 Sistemas de tráfico urbano.....	4
1.2 Simulación de tráfico urbano.....	5
1.3 Características de los simuladores de tráfico urbano.....	6
1.4 Especificación formal y modelado de sistemas de tráfico urbano	17
1.4.1 Ecuaciones diferenciales	17
1.4.2 Teoría de partículas	19
1.4.3 Redes de colas	19
1.4.4 Autómata Celular	21
1.4.5 Redes de Petri.....	22
1.5 Paradigmas de modelos de computación para la simulación de tráfico urbano	29
1.5.1 Basada en comunicación.....	29
1.5.2 Uso de paradigma multi-agente en simulación tráfico urbano	31
2 DISEÑO DE UN SIMULADOR DE TRÁFICO URBANO	35
2.1 Requerimientos generales	36
2.1.1 Enunciado de Problema / Oportunidad	36
2.1.2 Enunciado de los Objetivos del Proyecto.....	36
2.1.3 Beneficios Esperados	37
2.2 Requerimientos Funcionales.....	37
2.2.1 Propuesta y justificación de soluciones a requerimientos funcionales.....	38
2.3 Descripción del sistema.....	40
2.3.1 Arquitectura	40
2.3.2 Modelado de sistemas de tráfico urbano	45
2.3.3 Productos de terceras partes	46
3 MODELADO DE SISTEMAS DE TRÁFICO URBANO CON REDES DE PETRI....	49
3.1 Un sistema de red a tres niveles: 3-LNS.....	50
3.1.1 Redes de Petri generalizadas	50
3.1.2 Evolución del marcado de redes de Petri	51
3.1.3 Regla de Disparo	51
3.1.4 Sistema de red a tres niveles	51
3.1.5 Evolución del sistema de red.....	54

3.2	Caso de estudio.....	56
3.2.1	Estrategia de Modelado.....	56
3.2.2	El modelo de la red de tráfico	57
3.2.3	Las redes agente	59
3.2.4	Redes de nivel 3: los objetos y los recursos	62
4	IMPLEMENTACIÓN DEL SIMULADOR DE TRÁFICO URBANO CON AGENTES MÓVILES.....	67
4.1	Metodología de implementación multi-agente.....	68
4.2	Implementación del modelo multi-agente.....	69
4.2.1	Sistema Objetivo	69
4.2.2	Conocimiento Macro-k	69
4.2.3	Conocimiento Micro-k	70
4.2.4	Modelo de dominio	72
4.2.5	Modelo de diseño	73
4.2.6	Modelo Operacional.....	74
4.2.7	Construcción	78
5	ANÁLISIS DE RESULTADOS.....	79
5.1	Pruebas y simulaciones.....	80
5.2	Interpretación de resultados.	81
6	CONCLUSIONES	85
	REFERENCIAS BIBLIOGRÁFICAS	87

ÍNDICE DE FIGURAS

FIGURA 1-2 DIAGRAMA FUNDAMENTAL DE FLUJO DE TRÁNSITO.	11
FIGURA 1-12 RED DE PETRI CONTINUA, REPRESENTANDO COLA DE ESPERA.	27
FIGURA 1-13 RED DE PETRI, REPRESENTANDO UNA INTERSECCIÓN.	27
FIGURA 1-14 MODELO CON RP ESTOCÁSTICA DE UNA INTERSECCIÓN DE 2 FASES.	28
FIGURA 2-1 RELACIÓN DE EVENTOS ARRIBO, PARO Y CAMBIO DE FASE.	42
FIGURA 2-2 RELACIÓN DE EVENTOS CAMBIO DE CARRIL Y AVANCE DESPUÉS DEL CAMBIO.	42
FIGURA 2-3 SISTEMA MULTI-AGENTE DE SISTEMA DE TRÁFICO URBANO.	43
FIGURA 2-4 DIAGRAMA DE CLASES DEL SIMULADOR.	44
FIGURA 2-5 COMUNICACIÓN ENTRE EL AGENTE INTERSECCIÓN Y EL CALLE.	44
FIGURA 2-6 COMUNICACIÓN ENTRE LOS AGENTES INTERSECCIÓN.	45
FIGURA 2-7 MODELO EN REDES DE PETRI DE LA RED DE TRÁFICO URBANO.	45
FIGURA 2-8 MODELO EN RP :3LNS, DE UN SISTEMA DE TRÁFICO URBANO.	46
FIGURA 3-1 SEGMENTO DE UNA RED DE TRES NIVELES.	50
FIGURA 3-2 RED OBJETO.	51
FIGURA 3-3 RED AGENTE.	52
FIGURA 3-4 RED AMBIENTE.	53
FIGURA 3-5 CASO DE ESTUDIO.	56
FIGURA 3-6 JERARQUÍA DEL MODELO 3-LNS DE LA RED DE TRÁFICO.	57
FIGURA 3-7 RED AMBIENTE DE NIVEL 1, TIPO <i>TYPEURBAN</i> _{1,1} .	58
FIGURA 3-8 RED AGENTE CALLE DE NIVEL 2, TIPO <i>TYPESTREET</i> _{2,1} .	59
FIGURA 3-9 RED AGENTE VEHÍCULO DE NIVEL 2, TIPO <i>TYPECAR</i> _{2,2}	61
FIGURA 3-10 RED OBJETO DE NIVEL 3, TIPO <i>TYPEAE</i> _{3,1}	62
FIGURA 3-11 RED OBJETO DE NIVEL 3, TIPO <i>TYPESE</i> _{3,2} Y <i>TYPECLEE</i> _{3,3} .	63
FIGURA 3-12 RED OBJETO DE NIVEL 3, TIPO <i>TYPESE</i> _{3,4} Y <i>TYPECLE</i> _{3,5} .	64
FIGURA 3-13 RED OBJETO DE NIVEL 3, TIPO <i>TYPECE</i> _{3,6} .	65
FIGURA 3-14 RED OBJETO DE NIVEL 3, TIPO <i>TYPEPLAN</i> _{3,8} .	65
FIGURA 4-1 METODOLOGÍA PARA DESARROLLO DE MABS.	68
FIGURA 4-2 SIMULACIÓN CON LA TÉCNICA DE EVENTO SIGUIENTE.	71
FIGURA 4-3 LAS DIFERENCIAS ENTRE SIMULACIÓN CONTINUA Y DISCRETA EN ESPACIO.	72
FIGURA 4-4 MODELO DEL DOMINIO DEL SISTEMA.	72
FIGURA 4-5 MODELO DE DISEÑO DEL SISTEMA.	73
FIGURA 4-6 ALGORITMO DE CONTROL DE LA SIMULACIÓN.	75
FIGURA 4-7 ALGORITMO DE FLUJO DEL EVENTO <i>AE</i> .	76
FIGURA 4-8 CLASES DEL SIMULADOR.	77
FIGURA 4-9 USO DE VECTORES PARA ORDENAR LOS VEHÍCULOS EN LA CALLE.	78
FIGURA 5-1 METODOLOGÍA PARA LA IMPLEMENTACIÓN DE MABS. PONIENDO ÉNFASIS EN LA ÚLTIMA ETAPA.	80
FIGURA 5-2 PANTALLA DEL SIMULADOR.	81
FIGURA 5-3 GRÁFICAS DE DENSIDAD CON A) 0.2 MIN. Y B) 8 MIN. DE RETARDO EN REACCIÓN DEL CONDUCTOR	82
FIGURA 5-4 GRÁFICAS DE FLUJO CON A) 0.2 MIN. Y B) 8 MIN. DE RETARDO EN REACCIÓN DEL CONDUCTOR	82

ÍNDICE DE TABLAS

TABLA 1-1 CLASIFICACIÓN GENERAL DE LOS MODELOS DE TRÁFICO URBANO	9
TABLA 2-1 LISTA DE REQUERIMIENTOS FUNCIONALES.	38
TABLA 2-2 TIEMPOS DE EJECUCIÓN	40
TABLA 2-3 ESPECIFICACIÓN DES DE UN SISTEMA DE TRÁFICO URBANO.	41
TABLA 3-1 DEFINICIÓN FORMAL DE LA RED TIPO <i>TYPEURBAN</i> _{1,1}	58
TABLA 3-2 DEFINICIÓN FORMAL DE LA RED TIPO <i>TYPESTREET</i> _{2,1}	60
TABLA 3-3 DEFINICIÓN FORMAL DE LA RED TIPO <i>TYPECAR</i> _{2,2}	62
TABLA 3-4 DEFINICIÓN FORMAL DE LA RED TIPO <i>TYPEAE</i> _{3,1}	63
TABLA 3-5 DEFINICIÓN FORMAL DE LA RED TIPO <i>TYPESE</i> _{3,2}	64
TABLA 3-6 DEFINICIÓN FORMAL DE LA RED TIPO <i>TYPESE</i> _{3,4}	64
TABLA 3-7 DEFINICIÓN FORMAL DE LA RED TIPO <i>TYPECE</i> _{3,6}	65
TABLA 3-8 DEFINICIÓN FORMAL DE LA RED TIPO <i>TYPEPLAN</i> _{3,8}	66

PRESENTACIÓN

La zona metropolitana de Guadalajara (ZMG) es la segunda de mayor tamaño en la República Mexicana, con cerca de tres millones y medio de habitantes, los cuales corresponden alrededor del 54% de la población total del Estado de Jalisco.

La superficie urbanizada de la ZMG es de aproximadamente 35,000 hectáreas. La población de la ZMG ha registrado un incremento significativo en las últimas décadas. Actualmente existen más de un millón de vehículos, transportando apenas una o dos personas y cerca de cuatro mil quinientos autobuses, transportando mil pasajeros diarios cada uno. En consecuencia, se experimentan los fenómenos de congestión, colas, etc., observados en los sistemas de tráfico urbano cuando se incrementa el número de usuarios y no se aplican estrategias adecuadas de control y optimización para reducir el impacto de éstos fenómenos sobre los conductores y el medio ambiente, tales como estrés y contaminación; entre otros.

Una de las alternativas para abordar este tipo de problemas, es la herramienta de simulación, que ha venido incrementando su uso en los últimos años. El uso de ésta herramienta permite a los ingenieros de tráfico tomar decisiones en base a los resultados lanzados por la simulación, además de evaluar diferentes estrategias de control y nuevas tecnologías usadas para la optimización de los sistemas de tráfico urbano.

Para desarrollar un simulador se requiere de un modelo matemático que capture el comportamiento del sistema de tráfico urbano, existen metodologías que proponen diversas formas de capturar el comportamiento de los sistemas de tráfico urbano en modelos que pueden ser analizados de manera analítica o simulación. Nuestra tesis hace uso de modelos jerárquicos, que reducen la complejidad del diseño del software de simulación, proponiendo el uso del formalismo 3-LNS de redes de Petri. Este modelado da como resultado una especificación limpia y orientada a agentes móviles.

El proceso de implementación del modelo es llevado a cabo a través de una metodología propuesta (*Drogoul, 2002*) que nos permite generar un simulador basado en multi-agente (MABS), que es perfectamente modelado por el formalismo de redes de Petri. En la revisión de la literatura, se concluye que todos los simuladores usan el mecanismo de división de tiempo para el avance de la simulación. Se propone el uso de la técnica de evento siguiente, que tiene entre otras ventajas, el incremento en eficiencia de tiempo de ejecución y que representa de una manera natural a los sistemas de tráfico urbano.

La descripción de nuestro trabajo esta organizado como sigue:

- a) Capítulo 1, titulado “Modelado y simulación de sistemas de tráfico urbano”, en éste apartado se muestra un panorama de las filosofías existentes para el modelado de sistemas de tráfico urbano tales como redes de petri, autómata celular, entre otros. Además se analizan algunos simuladores desarrollados con éstas.
- b) Capítulo 2, titulado “diseño de un simulador de tráfico urbano”, En éste capítulo se presenta la especificación inicial de un simulador de tráfico urbano. Se concibe como un micro-simulador orientado a eventos basado en el enfoque de los sistemas multi-agente, donde los componentes son modelados usando un formalismo de redes de Petri anidadas a tres niveles.
- c) Capítulo 3, titulado “Modelado con n-LNS”, trata sobre la construcción de modelos de sistemas de tráfico urbano (STU) utilizando 3-LNS, un formalismo basado en redes de Petri el cual permite representar en tres niveles los componentes de un sistema de agentes móviles. Un modelo de STU describe en el nivel 1 la red de tráfico, en el nivel 2 se describen a los agentes, tales como, vehículos, semáforos, etc. y en el nivel 3 se describen los componentes de los agentes, en este caso, los planes de trayecto, habilidades y actitudes de los conductores, etc.,
- d) Capítulo 4, titulado “Implementación del simulador de tráfico urbano con agentes móviles”, analiza la forma en la que se implementa el simulador con la técnica orientada a eventos, basada en la metodología multi-agente. Mostramos además la metodología de implementación basada en multi-agentes, así como los pasos que se siguieron para conseguir el desarrollo del sistema.
- e) Capítulo 5, titulado “Análisis de Resultados”, se muestra los resultados obtenidos del simulador desarrollado.
- f) Capítulo 6, “Conclusiones”, resume el trabajo actual y futuro, así como algunas recomendaciones.
- g) El último capítulo, lista las referencias utilizadas para el desarrollo de éste trabajo.

Capítulo 1

MODELADO Y SIMULACIÓN DE SISTEMAS DE TRÁFICO URBANO

Resumen. En éste capítulo se revisa la literatura existente de simulación de tráfico urbano, así como los diferentes formalismos para su modelado, técnicas de simulación y la implementación de simuladores de tráfico urbano. Se muestra que la técnica de avance de tiempo de simulación por división de tiempo es la más utilizada. La técnica de evento siguiente es la más eficaz, pero pocos son los trabajos que la utilizan. Se revisan los diferentes paradigmas de programación para la implementación de un simulador, el paradigma basado en comunicación, resulta ser el más utilizado (intercambio de mensaje y multi-agente) y el más eficaz para la implementación sistemas de tráfico urbano.

1.1 Sistemas de tráfico urbano

Un *sistema de tráfico urbano* está compuesto por los siguientes elementos (figura 1-1) :

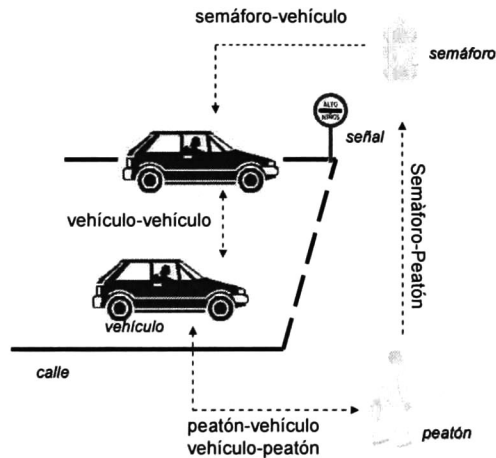


Figura 1-1 Elementos de un sistema de tráfico urbano.

- Calle:** es un recurso del sistema que es ocupado por los vehículos que pelean por ocupar un segmento de ésta. Por lo general, en redes urbanas se tienen de uno a dos carriles.
- Vehículos:** entidades básicas del sistema moviéndose en una dirección y con una posición dada, controlados por un conductor. Se puede realizar una descomposición más detallada nombrando su comportamiento, planes de viaje, etc. Podemos nombrar también a los autobuses, bicicletas, etc., pero por simplicidad de diseño definimos al vehículo como única entidad móvil del sistema.
- Semáforos:** son elementos de control de tráfico, categorizados por las estrategias de control que sea utilizado. Utilizados para asignar los recursos (calles). Éstos pueden contar con cierto grado de inteligencia, de manera que pueden tomar decisiones sobre que calle tiene más prioridad para dar el paso a los vehículos.
- Señalamientos:** son representaciones gráficas que informan al conductor sobre características especiales de la calle, el estado del tráfico de otras calles o alternativas de rutas, son de tipo ceda el paso, indicadores de máximos de velocidad, etc.

En la literatura el término *sistemas de transportación inteligente* (ITS: *intelligent*

transportation system), cubre la aplicación de una amplia variedad de tecnologías de computación, comunicación, posicionamiento, sensado, control y otras para la optimización de los sistemas de tráfico. Las principales categorías de los ITS (Hoogendoorn, 2001) incluyen a los *sistemas avanzados de manejo de tráfico* (ATMS: advanced traffic management systems) y a los *sistemas de información al viajero avanzado* (ATIS: advanced traveler information systems).

El estado del sistema se conoce a través del valor de las siguientes variables:

- a) Número de vehículos
- b) Estrategia de control de los semáforos.
- c) Número de accidentes de tráfico.
- d) Eventos de tráfico calendarizados (como calles en construcción, mítines o marchas, etc.).

Estas variables son modificadas por los siguientes *parámetros* del sistema.

- a) Velocidad de ingreso de los vehículos al sistema (tiempos de arribo).
- b) Capacidad de los segmentos de calle.
- c) Estrategia de control de los semáforos.
- d) Comportamientos del conductor.
- e) Situaciones climatológicas.

El sistema no cuenta con un objetivo global (como los sistemas de manufactura). Los objetivos son individuales (cada vehículo tiene su propio plan de viaje) que en conjunto presentan ciertos fenómenos conocidos en los sistemas de tráfico, tales como colas, congestión y olas; entre otros.

1.2 Simulación de tráfico urbano

La dinámica de un sistema de tráfico urbano (STU) tiene características que lo hacen muy semejante a la de los sistemas hidráulicos. Sin embargo, en un STU, las entidades que lo conforman tienen un comportamiento que pudiésemos llamar inteligente ya que la dinámica de las entidades (los vehículos) es controlada por el conductor, que por propia iniciativa puede cambiar la ruta de viaje del vehículo, incrementar su velocidad, etc.

El interés por estudiar estos sistemas se ha acrecentado debido a la necesidad de optimizar el control de tráfico, disminuir la congestión vial y la contaminación; entre otros fenómenos

característicos de éstos sistemas. De ahí la importancia de entender las relaciones entre sus componentes y poder predecir el funcionamiento del sistema bajo ciertas circunstancias con el cambio de algunos parámetros del sistema de tráfico urbano.

Existe la posibilidad de evaluar las diferentes alternativas de optimización directamente en los STU existentes y utilizar datos estadísticos para predecir los posibles cambios en el sistema, además de obtener las relaciones entre sus componentes para poder analizarlos. Sin embargo, éste proceso resulta poco viable por costoso y complejo para sistemas de gran tamaño. Para lograrlo existen varias alternativas. Una de ellas es la simulación, herramienta matemática que hace uso de la computadora.

Por medio de la simulación se pueden probar y analizar nuevas tecnologías, como ATIS, que requieren de predicción del comportamiento de los conductores frente a su posible implementación. Por estas razones, la simulación ha sido ampliamente usada como herramienta de evaluación de estrategias de control de tráfico y en consecuencia, varios simuladores han sido desarrollados para tratar de atacar diferentes problemas de tráfico.

El desarrollo de ATIS requiere del soporte de estudios complementarios para mostrar claramente la viabilidad de los sistemas y los beneficios que se pueden esperar de su operación. Las inversiones hechas que son muy altas, deben ser justificadas de una forma robusta, esto significa hacer estudios de viabilidad que validen los sistemas propuestos, asegurar los impactos esperados y proveer las bases para un análisis de costos bien sustentados. No hay duda que los ATIS representan el futuro de los STU y que el desarrollo de estos sistemas dominará la escena del transporte terrestre en los años venideros, sin embargo, todavía existen algunas áreas no exploradas en lo que concierne a la correcta implementación de éstas. Aún existen dudas de como los usuarios reaccionarán ante estos sistemas y las decisiones en como deben ser operados eficientemente, dudas que deberán ser clarificadas antes de realizar las instalaciones físicas de ITS.

1.3 Características de los simuladores de tráfico urbano

En la literatura existen diferentes simuladores de tráfico urbano, podemos usar las siguientes características para poder compararles (*Hoogendoorn, 2001*):

- a) Nivel de detalle.
- b) Escala de variables independientes.

- c) Representación de procesos.
- d) Escala de aplicación.
- e) Manejo de avance de tiempo de simulación.

A continuación describimos cada una de éstas características y describimos algunos simuladores que se distinguen por el uso de éstas.

a) Nivel de detalle

Los modelos de tráfico pueden ser clasificados de acuerdo al nivel de detalle con el cual es representado el sistema de tráfico. Esta clasificación puede ser llevada a cabo considerando las entidades de tráfico que son distinguidas y el nivel de descripción de éstas entidades en los modelos de flujo respectivos. Los simuladores se pueden categorizar por su nivel de detalle en tres tipos: macroscópica, mesoscópica y microscópica (*Hoogendoorn, 2001*).

o *Simulación macroscópica*

Los modelos de *simulación macroscópica* describen el tráfico a un nivel de agregación alto, como un flujo, sin distinguir las partes que le constituyen. Por ejemplo, el flujo de tráfico es representado de una manera agregada, usando características tales como nivel de flujo, densidad y velocidad. Las maniobras de los vehículos individuales, tales como los cambios de carril, no son por lo general representadas de manera explícita. En un modelo de simulación macroscópico se puede suponer que el flujo de tráfico está asignado de manera propia a los carriles de las avenidas y se emplea un método de aproximación para éste fin. Los modelos de flujo macroscópico pueden ser clasificados de acuerdo al número de ecuaciones parciales que forman parte del modelo y por el orden de las mismas ecuaciones.

o *Simulación mesoscópica*

Un modelo de *simulación mesoscópica* no distingue a los vehículos como individuales, pero especifica el comportamiento individual, por ejemplo, en términos probabilistas, para éste fin, el tráfico es representado por un pequeño grupo de entidades del tráfico. Las actividades y las interacciones son descritas a un nivel de detalle bajo, por ejemplo, una maniobra de cambio de carril pudiera ser representada como un evento instantáneo, donde la decisión para llevarla a cabo está basada en valores de densidad de carril relativas y en diferenciales de velocidad. Algunos modelos de simulación mesoscópica describen la dinámica de distribución de velocidad.

o *Simulación microscópica.*

Un modelo de *simulación microscópica* describe tanto el comportamiento espacio-tiempo de las entidades del sistema como las interacciones entre las entidades, con un alto nivel de detalle individual. Por ejemplo, para cada vehículo en el flujo de tráfico, un cambio de carril es descrito como una cadena detallada de decisiones del conductor, además de una descripción detallada del comportamiento del conductor, también es modelado en detalle el comportamiento de control del vehículo (por ejemplo cambiar velocidad, etc.) en correspondencia a las condiciones que prevalecen alrededor. Además, se describe el funcionamiento de algunas partes específicas (subunidades) del vehículo.

La simulación de tráfico microscópico ha sido probada como la mejor herramienta para cumplir con estos objetivos, esto no se debe sólo a su habilidad para capturar de manera completa la dinámica del fenómeno del tráfico dependiente del tiempo (congestionamientos, colas, etc.), sino que también de capturar el comportamiento de los conductores, cuando son expuestos a la implementación de sistemas ATIS (*Barceló, 2001*).

Según lo anterior podemos definir una clasificación de los elementos de los sistemas de tráfico que podemos simular usando cierto nivel de detalle (*Akcelik, 2004*). Esta estructura bidimensional (tabla 1-1) considera la naturaleza y el nivel de detalle ofrecido por el modelo de tráfico en el manejo de la geometría y los elementos del tráfico. La atención está puesta sobre el movimiento del vehículo en el flujo de tráfico.

Elementos del Tráfico	Vehículos Individuales			Micro-simulación
	Grupos de vehículos	Macro-simulación		Simulación Mesoscópica
	Ciclos de manejo ¹			Micro-analítico
	Flujo Tráfico	Macro-analítico	Mesoscópica-analítica	
	Modelos Flujo-Velocidad	Macro-analítico		
	Todos los carriles agregados	Grupos de Carriles ² (o "enlaces")	Carril o segmentos de carril	
	Elementos Físicos de las calles			

Tabla 1-1 Clasificación general de los modelos de tráfico urbano

b) Escala de variables independientes

Ya que la mayoría de los modelos de tráfico describen sistemas dinámicos, una clasificación natural es la escala de tiempo, se distinguen dos escalas de tiempo, continua y discreta. Un modelo continuo, describe la forma en la que el estado del sistema de tráfico cambia continuamente en el tiempo, en respuesta a estímulos continuos. Los modelos discretos asumen que los cambios en el estado del sistema ocurren discontinuamente en el tiempo a instantes de tiempo discreto. Además del tiempo, también se pueden describir como continuas o discretas otro tipo de variables, por ejemplo la posición, la velocidad, la velocidad deseada, etc..

o *Sistemas continuos*

Como ejemplo de un sistema continuo tenemos a la teoría cinética. En ésta teoría el tránsito se estudia como un gas donde las partículas que lo componen interactúan unas con otras y cada partícula representa un automóvil. En esta teoría el flujo de tránsito puede ser entendido al estudiar la función de distribución $f(x, v, t)$ que describe el número de autos en el intervalo de la carretera $(x, x + dx)$ con una velocidad $(v, v + dv)$ para un tiempo dado t . Esta función también se puede expresar en términos de dos variables fundamentales:

¹ Ciclos de manejo pueden ser definidos como las trazas tiempo-velocidad del vehículo que consiste de los elementos de aceleración, desaceleración cruce y ocupado

² Grupo de carriles es el conjunto de carriles con uno o 2 carriles compartidos (por ejemplo, carril 1: vuelta-izquierda y derecho, carril 2: derecho) o un conjunto de carriles de vuelta exclusivos (por ejemplo, un carril para las vueltas a la derecha).

concentración y flujo. Un buen entendimiento de la relación de concentración y flujo es la base para describir el tránsito. La curva flujo-concentración se denomina el *diagrama fundamental* de tránsito en una carretera o la *ecuación de estado* de la teoría de tránsito (ver figura 1-2). En bajas densidades, el flujo se incrementa linealmente junto con la densidad, los autos se mueven de manera independiente en el tránsito diluido con pequeñas fluctuaciones alrededor de una velocidad media, con el incremento en la concentración los autos se van estorbando unos a otros cada vez más, lo cual lleva a la reducción del promedio de velocidad. Los efectos de que los autos se estén estorbando se hacen dominantes en altas densidades. En una *densidad crítica*, el flujo exhibe un máximo, de tal manera que un incremento en la densidad resulta en un decremento del avance total y finalmente en una completa fase de congestión donde no hay auto que pueda moverse y el flujo es cero.

Por otro lado, la *varianza* en las velocidades de los autos propicia la formación de congestionamientos o atascamientos de tránsito. Los atascamientos de tránsito son fases localmente congestionadas en las cuales los autos viajan a baja velocidad o están detenidos. Entre los atascamientos uno encuentra autos en bajas concentraciones viajando a altas velocidades. Esas fases congestionadas son configuraciones bastante estables con su propia dinámica. Los autos conduciendo dentro de atascamientos se estabilizan y los autos saliendo desde el frente reducen la longitud del atascamiento. Desde éste punto de vista, los atascamientos se mueven lentamente en la dirección opuesta a la de los autos, su velocidad se puede estimar con un modelo muy simple donde uno asume que hay una velocidad pequeña constante v_1 y una densidad ρ_1 en el atascamiento y una velocidad alta v_2 y baja densidad ρ_2 fuera de él. Si los autos aceleran y disminuyen su velocidad en el acto, el atascamiento se mueve con una velocidad $v_{jam} = -\frac{(v_2 - v_1)\rho_1}{\rho_2 + v_1}$

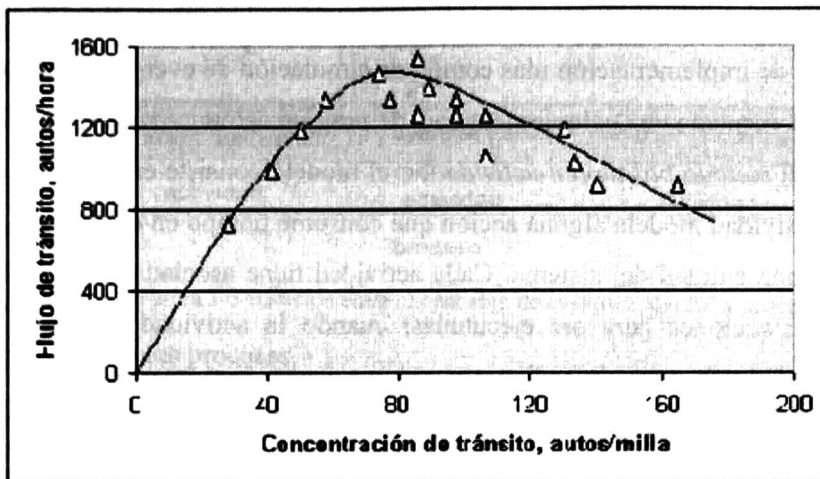


Figura 1-2 Diagrama fundamental de flujo de tránsito.

Puesto que el radio de densidad es pequeño, los atascamientos se mueven lentamente (Zamora, 2002).

o *Sistema de eventos discretos*

Un sistema de tráfico urbano puede ser especificado como un sistema de eventos discretos (DES: discrete-event system), para lograrlo tenemos que tener bien definidas las interacciones entre los componentes del sistema y que los cambios en el estado del sistema se lleven a cabo a intervalos de tiempo predefinido.

La técnica de modelado discreto, ha demostrado ser significativamente más eficiente computacionalmente; ya que implementa aspectos importantes de la dinámica de fluidos, los cuales también están presentes en el flujo de tráfico de una manera natural (Russell, 1996). Existen 3 enfoques utilizados para la simulación de DES (Helsgaun, 2000): el enfoque basado en eventos, el basado en actividades y el basado en procesos.

En el *método basado en eventos* el modelo consiste en una colección de eventos. Cada evento modela un cambio de estado del sistema y es responsable del calendarizado de otros eventos que dependen de éste evento. Cada evento tiene asociado un tiempo de ocurrencia y algunas acciones a ser ejecutadas cuando el evento ocurra. En el ejemplo de simulación de tráfico urbano, el arribo de un vehículo es un ejemplo de un evento, las acciones asociadas con éste evento son la inclusión del vehículo en el carril y el calendarizado del siguiente

arribo de un vehículo. La simulación con el método basado en eventos es el más simple y es el estilo de implementación más común de simulación de eventos discretos porque éste puede ser implementado en cualquier lenguaje de programación.

En el *método basado en actividades*, el modelo consiste en una colección de actividades. Cada actividad modela alguna acción que consume tiempo en el sistema, que es desempeñada por alguna entidad del sistema. Cada actividad tiene asociada: una condición de inicio, una serie de acciones para ser ejecutadas; cuando la actividad comience, la duración de la actividad y una serie de acciones a ser ejecutadas para cuando la actividad finalice. En el problema de tráfico urbano el cruce a través de una calle es un ejemplo de una actividad, la condición de inicio de esta actividad es que exista el espacio disponible y que no haya objetos a lo largo del camino. Cuando la actividad comienza, el recurso (tramo de calle) es tomado por el vehículo y lo libera conforme el vehículo va avanzando. La duración de la actividad dependerá de la velocidad del vehículo. El método basado en actividades es relativamente fácil de entender, sin embargo su eficiencia de ejecución es pobre comparada con el método basado en eventos.

En el *método basado en procesos* el modelo consiste en una colección de procesos. Cada proceso modela el ciclo de vida de una entidad y es una secuencia lógica de actividades relativas ordenadas en el tiempo. En el problema de tráfico urbano el vehículo es el actor principal de un proceso. Cada vehículo ejecuta la siguiente secuencia de actividades: arrancar, avanzar, cruzar, etc. ya que los procesos representan a los objetos en el mundo real, la simulación basada en procesos es de alguna manera fácil de entender. La implementación, sin embargo, no es fácil y la eficiencia de ejecución puede ser pobre si la implementación no se hace correctamente. En la figura 1-3 se ilustra una relación entre los conceptos evento, actividad y proceso.

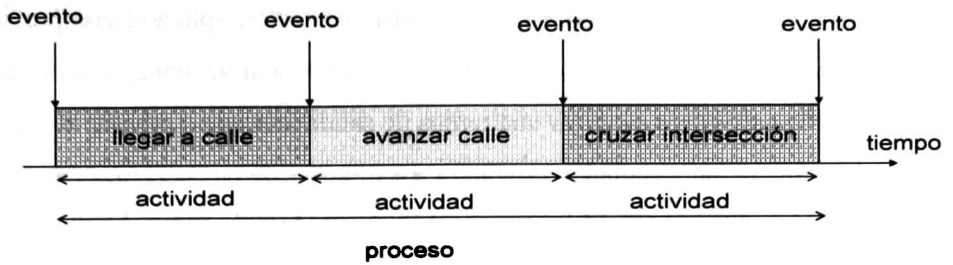


Figura 1-3 Relación entre el concepto de evento, actividad y proceso.

c) Representación de los procesos

A éste respecto, se distinguen los modelos determinísticos y estocásticos. El primero no contiene variables aleatorias. Los modelos estocásticos incorporan procesos que incluyen variaciones aleatorias. Por ejemplo, un modelo de siguiendo-al-vehículo puede ser formulado tanto como determinístico o como estocástico, definiendo el tiempo de reacción del conductor constante o de manera aleatoria, respectivamente.

d) Escala de aplicación

La escala de aplicación indica el área de aplicación del modelo, por ejemplo, el modelo puede describir la dinámica de las entidades para un simple tramo de avenida, una ciudad completa, sólo una calle, o solo una intersección; entre otras.

e) Manejo de avance del tiempo en simulación

Otra característica es la forma en la que se maneja el avance del tiempo de simulación y los podemos clasificar en dos tipos: división de tiempo (*time-slice*) y de evento siguiente.

o *División de tiempo: (time –slice)*

Probablemente la manera más sencilla de controlar el flujo de tiempo en la simulación es incrementarlo en intervalos iguales. Éste método es llamado “división de tiempo” e involucra actualizar y examinar el modelo a intervalos regulares de tiempo, así, para una división de tiempo de longitud Δt , el modelo es actualizado en el instante $(t+\Delta t)$ para cambios ocurridos en el intervalo $(t$ a $(t+\Delta t))$.

La simulación que utiliza ésta técnica, realiza pasos de tiempo discretos (iguales), en un ambiente continuo del espacio. En cada paso de tiempo, el vehículo mide la distancia a los coches

que se encuentran a su alrededor y de esta información aplica el comportamiento de conductor para decidir si acelera o desacelera. También, para cada vehículo se tomará la decisión si deberá cambiar de carril para alcanzar una rampa de salida, entrada o rebasar al vehículo que va delante.

JUDGE, es un simulador orientado a eventos (*Russell, 1996*), originalmente diseñado para trabajar con FPGA (Field-Programmable Gate Array) como hardware de apoyo adicional, haciéndolo un poco más económico ya que en cierta forma esto suplanta a la computación masiva o paralela, implementada por otros simuladores. En *JUDGE* se divide la calle en celdas, de tamaños predefinidos (6 metros generalmente) y cada vehículo usa una celda para simular su avance, mismo que dependerá del estado de las celdas vecinas (ocupadas o no). A diferencia del modelo de autómatas celulares, éste contiene celdas llamadas “celdas rápidas”, que pueden teletransportar (calendarizar a futuro) el vehículo hasta varias celdas adelante, en un sólo paso de simulación. La forma de avanzar el tiempo y el uso de celdas por avanzar, es muy semejante al usado por el *proyecto de control de tráfico urbano de Guadalajara*, liderado por la Dra. Helena Unger (*Unger, 2005*) de la UDG que utiliza un micro-simulador basado en la teoría de redes de colas.

CELL-DEVS. Los modelos DEVS (*LoTartaro, 2001*) se construyen en base a un conjunto de modelos de simulación básicos, que se combinan para formar modelos acoplados. El paradigma *CELL-DEVS* permite la descripción de modelos celulares a través de su definición como modelos atómicos DEVS. Los modelos pueden ser de comportamiento (atómicos) o estructurales (acoplados). Un modelo acoplado especifica cómo se conectan las entradas y salidas de los componentes. Los nuevos modelos también son modelos básicos modulares y pueden usarse para armar modelos de mayor nivel. El paradigma *CELL-DEVS* permite la descripción de modelos celulares a través de su definición como modelos atómicos DEVS con distintos tipos de demoras, cada celda será definida como un modelo atómico y podrán utilizarse distintas clases de demoras para su comportamiento, estos modelos utilizan la técnica de división de tiempo para realizar el avance de los vehículos en las celdas.

SUMO (*Krajzewicz, 2002*), es otra herramienta que utiliza la técnica de división de tiempo, para el avance del tiempo en la simulación. La simulación de flujo de tráfico urbano es basada en un modelo microscópico. Esto es, cada vehículo que se mueve en la red de tráfico simulada, es modelado individualmente teniendo un lugar y cierta velocidad. En cada paso de simulación, el cual, tiene una duración de 1 segundo, el estado del vehículo es actualizado dependiendo del

vehículo que esté adelante y de las características de la red en la que el vehículo se esté moviendo. Éste simulador puede ejecutar alrededor de 1 millón de movimientos de vehículos por segundo, dependiendo de la complejidad de la red. BABSIM (Erlemann, 2004), es muy semejante a SUMO, las divisiones de tiempo son ajustables desde 0.1 a 1 segundo.

- o *Técnica del evento siguiente*

Ya que muchos sistemas tienen ciertos periodos en el que no se ejecuta ningún proceso, la longitud de cambio del tiempo de simulación es variable, entonces es preferible utilizar una variable de tiempo que cambie de igual manera. En éste caso, 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 (ejecutivo), avanza el tiempo hasta la hora en la que va a ocurrir el evento significativo siguiente. Por lo tanto, si nada va a ocurrir en los próximos 3 minutos, el ejecutivo moverá el tiempo de simulación 3 minutos adelante. La naturaleza de saltar entre los puntos de tiempo más significativos, lo hace ser más eficiente y permite que los modelos sean evaluados más rápidamente. Invariablemente, el software de simulación que utilice esta técnica, tendrá la capacidad de graficar al usuario los cambios del estado del sistema (inicia máquina a trabajar, se descompone, termina trabajo, etc.) y el movimiento de las entidades. Como ya dijimos, la técnica salta entre los puntos de tiempo significativos, entonces estos saltos pueden ser desiguales. El efecto producido en las gráficas mostradas al usuario, pudiesen resultar engañosas ya que parecerá que el sistema es analizado por periodos de tiempo muy largos, cuando no es así. Éste efecto es análogo al observar una película, se adelanta rápidamente y luego continua su velocidad normal.

La técnica se muestra en la figura 1-4. El diagrama muestra dos elementos esenciales: el reloj y el ejecutivo de la simulación. Aquí el ejecutivo de la simulación utilizará una lista del eventos (una cadena de eventos cronológicamente establecidos).

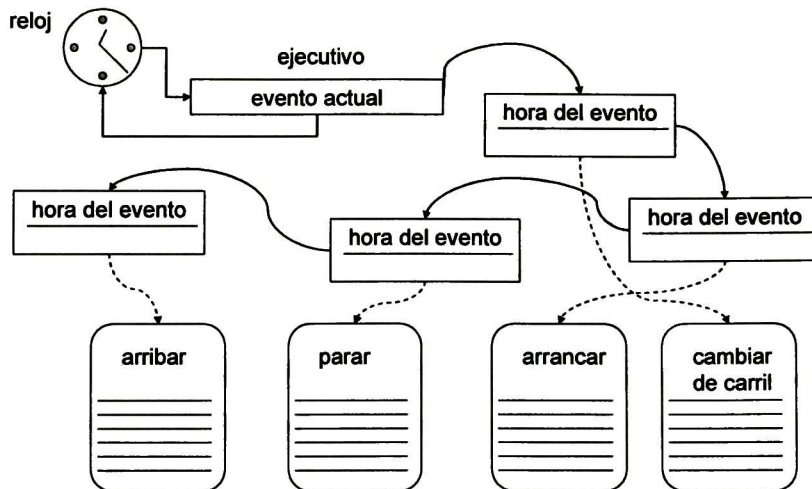


Figura 1-4 Detalle de la técnica evento siguiente.

El ejecutivo es responsable de tomar el evento siguiente a ejecutar. El ejecutivo quita el primer evento de la lista y ejecuta el evento de acuerdo a la lógica del modelo. Cualquier nuevo evento que ocurra, se inserta en la lista en el orden adecuado, el ciclo entonces se repite, cada evento en la lista de eventos, tiene dos elementos de información importantes. El primer elemento es el instante de tiempo en el que será ejecutado. El segundo elemento es la referencia a la lógica del modelo que necesita ser ejecutada (que acciones deberán ejecutarse). Esto permite que el ejecutivo lleve a cabo la lógica correcta en el instante de tiempo correcto. Observe que más de un evento puede referirse a la misma lógica, esto significa que puede ser utilizada muchas veces durante el funcionamiento de la simulación.

AIMSUN2 es un simulador que combina las dos técnicas de avance de tiempo de simulación (Barceló, 1998). En principio para actualizar los estados de cada vehículo usa división de tiempo ya que en cada paso de simulación, dependiendo del estado del vehículo y de sus vecinos, aplica algoritmos de siguiendo-al-vehículo y cambio-de-carril para definir su estado siguiente. Utiliza el método de división de tiempo para el cambio de estado de los semáforos, usando lo que se llama escaneo de actividades. El 94 % del tiempo consumido por la simulación es causado por el tiempo que se tarda el simulador en actualizar todas las entidades del ambiente (vehículos) (Barceló, 1998). Proponen el uso de la programación en paralelo, usando hilos Sin embargo el uso de hilos requiere varios procesadores para obtener una buena eficiencia de implementación, que a la vez incrementa el costo del hardware utilizado. El sistema tiene un editor orientado al

usuario para la creación de la red urbana y la simulación de diferentes tipos de vehículos, así como diferentes estrategias de control.

Conocer las características del simulador nos permite saber cuales son los alcances y limitaciones de la simulación. En la siguiente sección, abordaremos la problemática de capturar el comportamiento del sistema de tráfico urbano como parte del diseño de un simulador de tráfico urbano, es decir, el modelado y especificación formal de un sistema de tráfico urbano. Existen diferentes formalismos que permiten representar un sistema, algunos tienen ciertas ventajas sobre otros. La selección dependerá en que es lo que queremos simular y las herramientas que nos proporciona el formalismo.

1.4 Especificación formal y modelado de sistemas de tráfico urbano

Un modelo es definido como la representación de un sistema con el propósito de estudiarlo. Los modelos pueden ser clasificados como matemáticos o físicos. Un modelo matemático usa nociones simbólicas y ecuaciones matemáticas para representar el sistema. Un modelo de simulación es un tipo particular de modelo matemático de un sistema (*Banks, 2000*).

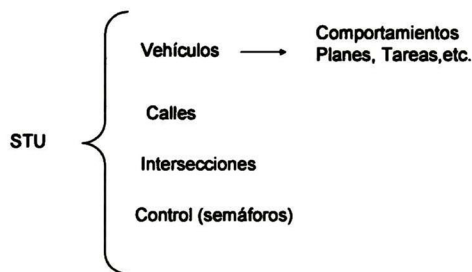


Figura 1-5 Entidades que conforman la STU.

En la figura 1-5 se muestran las entidades que conforman el STU. Existen diferentes formalismos para modelar cada uno de esos componentes, algunos tienen la capacidad de representar los comportamientos con gran eficiencia y otros tienen mucha mayor ventaja en el modelado de la infraestructura (calles, intersecciones, etc.).

A continuación analizamos las principales metodologías existentes para especificar y modelar sistemas de tráfico urbano.

1.4.1 Ecuaciones diferenciales

Son usadas principalmente para el modelado de comportamientos continuos, un caso son los

comportamientos del conductor en las calles de los sistemas de tráfico urbano.

a) *Modelo siguiendo al-vehículo (car-following) (Mussi, 2000)*

Seguindo-al-vehículo se enfoca principalmente en describir de una manera detallada cuando un vehículo sigue a otro. Esta descripción puede ser usada para aproximarse al comportamiento macroscópico del flujo de tráfico en un sólo carril. Los modelos siguiendo-al-vehículo asumen que existe una relación respuesta-estímulo que describe el proceso de control de una unidad vehículo-conductor:

$$Response(t + T) = \lambda * Stimulus(t); \text{ Donde } \lambda \text{ es alguna constante.}$$

El modelo de siguiendo al vehículo generalmente asume que un conductor intenta mantener al vehículo siempre a la delantera y evitar colisiones. El conductor para lograr estos objetivos dependerá de la experiencia y habilidad del mismo, en general, podemos decir que el estímulo es dado por las velocidades relativas. La respuesta de este modelo es la aceleración o desaceleración del vehículo seguidor.

En el modelo lineal, las respuestas no dependen del espacio entre los vehículos. Ésta suposición puede no ser muy realista. Para poder tomar en cuenta el espacio entre vehículos, podemos suponer que λ es inversamente proporcional al espacio entre vehículos, de una manera más general, todos los modelos pueden ser resumidos de la siguiente forma:

$$\begin{aligned} \bar{x}_f(t + T) &= \lambda * [\dot{x}_l(t) - \dot{x}_f(t)] \\ \lambda &= a_{l,m} * \dot{x}_f^m(t + T) / [x_l(t) - x_f(t)]^l \end{aligned} \quad (1-1)$$

Donde $a_{l,m}$ es una constante que será determinada experimentalmente.

b) *Modelo de cambio de carril (lane changing) (Mussi, 2000)*

Sin considerar la motivación (rebasar un carro que va a baja velocidad, tomar la posición correcta para la siguiente vuelta, evitar la colisión con algún obstáculo, etc.) que lleva a un vehículo a realizar un cambio de carril, éste consiste en moverse de un carril a otro en pocos segundos y con las restricciones de los vehículos en el carril destino y el carril origen. Esto puede parecer sencillo pero para modelar un cambio de carril de una forma realista no lo es tanto, la

mayoría de los esfuerzos de investigación en esta área se han enfocado en el comportamiento de aceptabilidad de espacio (gap acceptance).

Gipps (*Wilson, 2001*) desarrollo un modelo para una situación de manejo en un área urbana la cual intenta contestar las siguientes preguntas: ¿Es posible cambiar de carril?, ¿es necesario cambiar de carril?, ¿es deseable cambiar de carril?.

1.4.2 Teoría de partículas

Aunque éste tipo de modelado distingue los vehículos de manera individual, su comportamiento es descrito agregando ecuaciones de movimiento, por ejemplo un modelo de flujo de tráfico macroscópico. Estos modelos de partículas pueden ser considerados como un tipo específico de una aproximación de solución numérica (llamada métodos discretos de partículas), aplicadas a los modelos de flujo de tráfico continuo mesoscópico o macroscópico. Un ejemplo del modelo de partículas es INTEGRATION, (Brooks,1998) que es un modelo de simulación dinámica y microscópica. El flujo de tráfico en éste modelo es representado como una serie de vehículos individuales donde cada uno sigue una relación de flujo de tráfico macroscópico previamente especificado.

1.4.3 Redes de colas

Un modelo simple con teoría de colas, es mostrado en la figura 1-6, los clientes arriban uno tras otro y son ordenados en una cola o línea de espera y eventualmente son servidos y finalmente dejan el sistema.

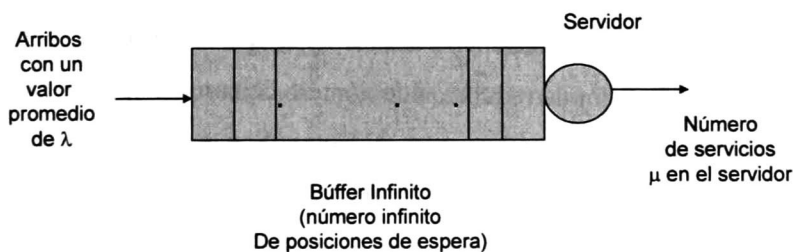


Figura 1-6 Modelo general de la teoría de colas.

Lo mismo puede ser utilizado para modelar un sistema de tráfico urbano (figura 1-7). Una cola es formada cuando la demanda (“la cantidad de arribos”) excede la capacidad (“La cantidad de servicios”). Se utiliza para determinar las longitudes de colas que se forman por los vehículos al querer dar vuelta o al cruzar alguna intersección o del número de los carriles requeridos en el diseño geométrico de una avenida.

Datos Básicos para análisis de colas:

- a) Promedio de cantidad de vehículos que arriban, l (veh/hr)
- b) Promedio de la cantidad de vehículos que son servidos (o que salen) , m (veh/hr).
- c) Número de canales de procesamiento.
- d) Tipo de cola:
 - FIFO – “first in, first out” (“primero en entrar, primero en salir”, es el más común por ejemplo, una línea de espera típica).
 - LIFO – “last in, first out” (“último en entrar, primero en salir”, menos común, por ejemplo los ferrocarriles o los sistemas de tren ligero).

Modelos de colas de tipos diferentes:

- a) Cola D/D/1 – Arribos Uniformes (determinísticos), salidas uniformes, un canal.
- b) Cola M/D1 – Arribos Poisson (proceso aleatorio), salidas uniformes, un canal.
- c) Cola M/M/1 – Arribos Poisson, salidas Poisson, un canal.
- d) Cola M/M/N – Arribos Poisson, salidas Poisson, canales múltiples (N)

Ejemplo del modelo Cola D/D/1 arribos uniformes, salidas uniformes un canal: Un embotellamiento en una avenida de alta velocidad, es una aplicación especial de un análisis de cola D/D/1. Los embotellamientos ocurren cuando una porción de la avenida tiene una capacidad menor que la parte de entrada de la avenida. Existen generalmente dos tipos de embotellamientos:

- a) Embotellamientos Recurrentes. Existen donde la avenida por sí misma limita su capacidad, como con la reducción de número de carriles.
- b) Embotellamientos inducidos por accidentes. Son resultado de un accidente de tráfico o una descompostura, donde el movimiento del tráfico es restringido, así que la capacidad es reducida en gran medida.

La teoría de redes de colas en su forma más simple asume que un servicio es proporcionado con una cierta velocidad y que las solicitudes a ese servicio llegan usualmente a diferentes velocidades. El tiempo entre el arribo del elemento n y el elemento $n-1$ de una cola se llama el tiempo de *interarribo* del elemento n y la longitud de tiempo en la cual el elemento n de la cola permanecerá en servicio, se llama tiempo de servicio.

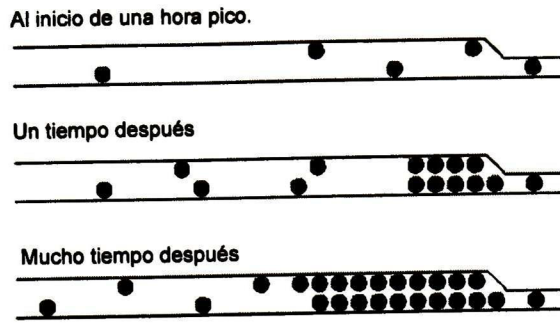


Figura 1-7 Fenómenos de tráfico usando teoría de colas.

Supongamos que el índice (velocidad) de arribos es λ y el índice de salidas desde una cola no vacía es μ ; entonces el promedio de la distribución de interarribos es $1/\lambda$ y el promedio de la distribución de tiempo de servicio es $1/\mu$. El índice de esas cantidades $\lambda/\mu = \rho$ es llamado la *intensidad de tránsito* de la cola.

1.4.4 Autómata Celular

Los modelos de autómata celular (CA: celular autómata) describen al sistema de tráfico como una rejilla (figura 1-8) de celdas de tamaño igual (típicamente cada una representando 7.5 m). Un modelo CA describe en una manera discreta el movimiento de los vehículos desde una celda a otra. El tamaño de las celdas esta seleccionado de tal forma que un vehículo yendo a una velocidad igual a 1 se mueve a la siguiente celda en el flujo correspondiente durante un paso de tiempo de simulación. La velocidad del vehículo sólo puede asumir un número limitado de valores discretos que van desde cero a v_{\max} . El proceso puede ser estructurado en tres pasos:

- Aceleración:** cada vehículo con una velocidad menor a la velocidad máxima, v_{\max} , acelera a su velocidad mayor, esto es, $v \leftarrow \min. (v_{\max}, v+1)$
- Desaceleración:** si la velocidad es menor que el espacio de seguridad al vehículo precedente, entonces el vehículo desacelerara: $v \leftarrow \max (v-1, 0)$.
- Alentarse:** con la probabilidad dada de p_{\max} , la velocidad de un vehículo sufre una disminución espontánea: $v \leftarrow \max (v-1, 0)$.

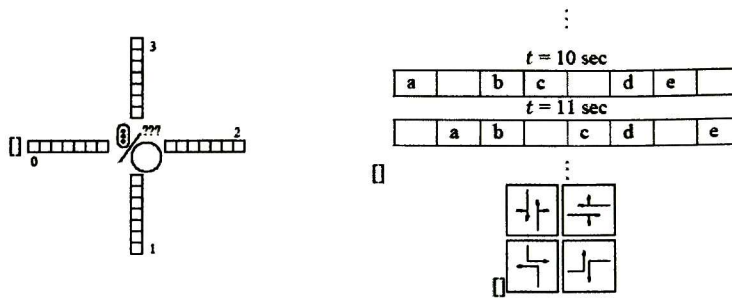


Figura 1-8 Lógica de manejo de un autómatas celular y Representación de una glorieta en CA.

Usando éste conjunto mínimo de reglas de manejo y la habilidad para aplicar la computación paralela, el modelo CA es muy rápido y puede consecuentemente ser usado tanto para operaciones de simulación de tráfico en redes de tráfico de alta escala y para propósitos de asignación y predicción de tráfico. El modelo inicial de un sólo carril de Kai Nagel (*Nagel, 2002*), (*Nagel, 1992*) ha sido generalizado para flujo de tráfico multi-carril y multi-clase (*Nagel, 1998*). Debido a la simplicidad de computación, los modelos CA se pueden aplicar a redes de tráfico muy grandes.

CA demuestra que no ocurre congestión a bajas densidades pero existen cambios espontáneos que generan congestión a altas densidades. Éste modelo fue exitoso en su habilidad para imitar patrones de tráfico actuales. Los modelos MÁS son usualmente formulados con líneas de una dimensión de CA, el cual proporciona comportamientos reales de cambio de carril. El movimiento, sin embargo, depende en el número de espacios vacíos alrededor del agente, velocidad y obstáculos

1.4.5 Redes de Petri

Comparado con otros métodos analíticos y de simulación, éste método tiene las siguientes ventajas:

- a) Las redes de Petri (RP) pueden fácilmente expresar concurrencia, competencia por recursos y sincronización, actividades entre el tráfico y el control del mismo, ayudando a lograr resultados más convincentes.
- b) El modelado con RP permite hacer cambios fáciles a la configuración de la red, la lógica

de control del tráfico, temporización y coordinación y las suposiciones del flujo de entrada.

- c) Un modelo grafico en RP es soportado matemáticamente, haciendo posible el verificar que la lógica de control asegure todas las exclusiones mutuas, que las capacidades no sean excedidas y un bloqueo del sistema no suceda debido a la lógica de control o temporización. Dada la versatilidad de RP existen en la literatura una gran cantidad de modelos para tráfico urbano (*Wang, 1999*), (*Tolba, 2001*), (*DiCesare, 1994*), (*Giglio, 2001*).

a) Modelado con RP temporizada coloreada

El formalismo RP es usado para modelar el flujo vehicular y el control del mismo. El modelo puede ser visto como en dos partes que interactúan entre si: el control de señales de tráfico y el flujo de tráfico en la red, buscando con esto dividir el sistema de por si complejo en RP menos complejas (*DiCesare, 1994*).

La primer parte, figura 1-9, describe las interacciones entre las calles, es decir las intersecciones y flujo del tráfico, modelando la ruta y la dinámica de los automóviles a través de las intersecciones y de los segmentos de calle. La otra parte, llamada RP de control (figura 1-10), describe la operación de cada una de las señales de tráfico, manejando la secuencia y lógica de control de los semáforos. Esta RP garantiza las propiedades de exclusión mutua, por ejemplo, dos direcciones en conflicto (flujo que puede provocar un accidente o choque entre carros) no pueden estar al mismo tiempo en verde.

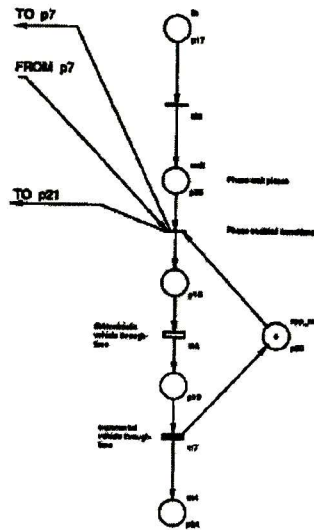


Figura 1-9 Red que representa el flujo de tráfico en calle e intersección.

Existen conexiones directas al módulo de control donde el movimiento de los carros es afectada por los semáforos en la intersección. Existe también ciertas conexiones de retroalimentación que entregan datos al módulo de control, enterándose de la presencia de los carros en diferentes ubicaciones en el sistema, modelando con esto sistemas de control de tráfico inteligente. Para lograrlo, el modelo tiene la capacidad de manejar cierto tipo de transiciones que ejecutan código, estas son llamadas *transiciones aumentadas*, estas son usadas para determinar que fase de semáforo deberá ser ejecutada y para lograrlo, ejecuta código que define la fase que se pondrá en función, dependiendo del número de vehículos (marcas³ de la red de Petri) en espera en cada una de las calles de la intersección. En conclusión, dos estrategias de control son probadas:

- Ciclo predefinido completo de cambio de fases
- Cambio de fases dependiendo de la selección hecha por la transición aumentada (control inteligente).

³ En la literatura de redes de Petri, se conocen como “tokens” por su traducción en ingles.

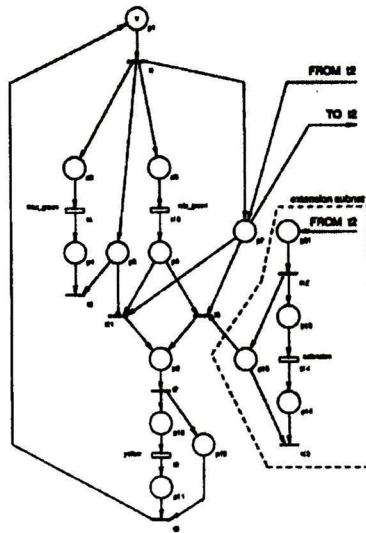


Figura 1-10 Red de control (semáforo)

Otro trabajo que utiliza RP coloreada para modelar flujo de tráfico urbano (*Kinata, 2001*), donde se aprovecha las características gráficas de RP, figura 1-11. Las marcas representan los vehículos que están usando la calle y los colores indican la ruta que seguirán estos vehículos. Usando la micro-simulación, representa sólo calles de un carril y cada lugar representa un espacio de 7.6m (para un vehículo aproximadamente), los lugares son temporizados, esto representa el tiempo de espera para cruzar éste espacio, existe cierto tipo de retroalimentación dependiendo en el flujo del carril. Desafortunadamente, la explosión de estados debido al uso de los lugares para representar los segmentos de red hacen que éste enfoque no sea el más adecuado para la representación de flujo de tráfico urbano.

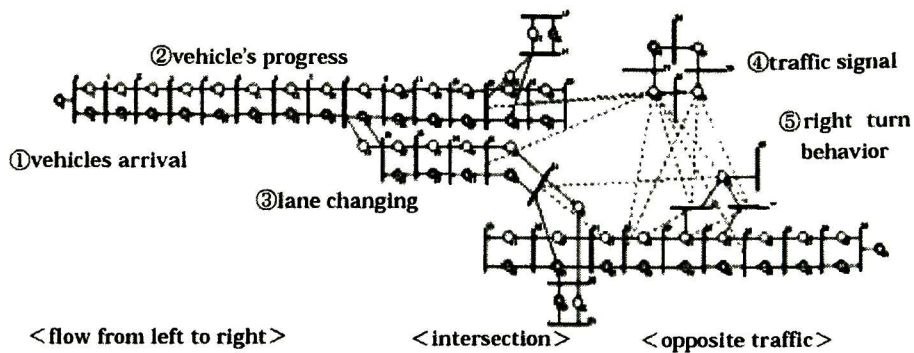


Figura 1-11 representación de Flujo de Tráfico Urbano, usando RP coloreadas.

b) Modelado con RP Continúas e Híbridas

El modelado y control del comportamiento de tráfico con RP continuas, ha sido muy difundido (*Julvez, 2004*), principalmente en el uso de RP híbridas, esto es, RP que tienen partes con comportamiento de RP continuas y partes con comportamiento de RP discretas. Éste trabajo se aborda el problema de la explosión de estados generada por la cantidad de vehículos que se encuentran en la red de tráfico (más de un millón para modelos de gran escala), adoptando un enfoque macroscópico usando variables como la densidad, la velocidad y el flujo. En éste reporte, se usa el mismo concepto de transiciones de velocidad variable o semántica de servidores infinitos (*Tolba, 2001*), permitiendo simular con gran exactitud el flujo de tráfico a través de una sección de calle, así dependiendo de la densidad de la sección de calle siguiente dependerá el de la sección anterior, así si se genera una congestión adelante esta se extenderá a las secciones anteriores. El problema que se observa, es que éste modelo, por ejemplo, puede ser usado para representar el comportamiento de descarga de una batería, pero no lo podemos usar para modelos como los de tráfico urbano (*Julvez, 2004*). Para explicar esta limitante, veamos el siguiente ejemplo, usando una línea de clientes en espera de ser atendidos por un cajero en un banco: los símbolos en el lugar p_1 de la figura 1-12, representan a los clientes que están en espera de ser atendidos por el cajero (transición t_1). Llegará un momento en que esa transición estará ocupada constantemente, hasta que no se vacié el lugar p_1 , pero llegara un momento en que los clientes empiecen a llegar esporádicamente, pero la transición se tendría que seguirá comportando exponencialmente, es decir seguirá atendiendo clientes de una manera exponencial, lo cual no es cierto el cajero no por tener 1000 clientes en la fila incrementara en 1000 su velocidad y así de

manera infinita, no es así ya que hay un límite de velocidad, es entonces que se propone (Julvez, 2004) una modificación al modelo propuesto originalmente (Tolba, 2001).

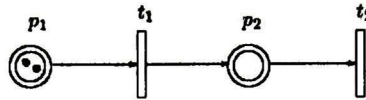


Figura 1-12 Red de Petri continua, representando cola de espera.

La manera de hacerlo es agregando 3 lugares, donde cada uno de ellos representa respectivamente la capacidad del carril, el máximo flujo posible y la capacidad de la siguiente sección, como se muestra en la figura 1-13:

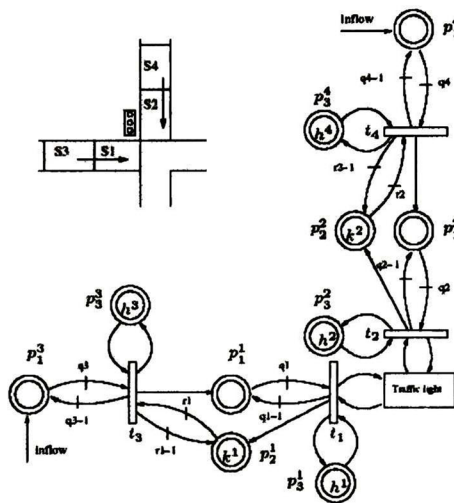


Figura 1-13 Red de Petri, representando una intersección.

Actualmente la mayoría de los modelos de RP que existen para sistemas de tráfico, representan al tráfico como flujo continuo, en lugar de manera discreta. En parte esto es debido a la búsqueda de lograr relajar el sistema y evitar en lo posible, la explosión de estados debida a lo complejo de la red y la gran cantidad de entidades de que está formado el sistema de tráfico urbano. Se observa, por ejemplo, que los parámetros de desempeño como el retardo, el grado de saturación y la longitud de colas son derivadas de funciones analíticas las cuales dependen ya sea de una representación idealizada del sistema o métodos empíricos derivados de datos de campo (Barclay, 1999). Conforme el nivel de diseño de los modelos de tráfico se incrementa, los modelos determinísticos se vuelven más problemáticos. En condiciones de congestión, las funciones retardo-volumen tienden a ser inestables y las interacciones entre las intersecciones y

los efectos de segmentos de calle muy pequeños introduce discontinuidades en las funciones. Aunque sea teóricamente posible incrementar el sistema que es analizado y adoptar funciones más complicadas, en práctica, el sistema fácilmente se vuelve inmanejable (Barclay, 1999). Así que podemos concluir, no es satisfactorio representar al tráfico como flujo continuo ya que por ejemplo, el número de arribos en el periodo de análisis no necesariamente iguala el número de salidas, precisamente causado por la discontinuidad de las funciones en redes muy grandes.

c) Modelado con RP estocásticas.

Usando RP estocásticas se puede modelar el flujo de tráfico urbano, así como el comportamiento de los vehículos en el cruce de una intersección (Figura 1-14). Usando un modelo de distribución aleatoria se describe el movimiento de los vehículos en un segmento de calle entre dos intersecciones consecutivas (Wang, 1999).

El método esta basado en el modelo estocástico de RP de intersecciones aisladas, las cuales consideran una sola dirección del tráfico a lo largo de las calles en todo momento y las interacciones entre el tráfico de diferentes direcciones son aproximadas parcialmente por modelos estadísticos. Haciendo esto, reducen dramáticamente la complejidad de cómputo, que otros métodos basados en RP no logran combatir, debido a la consideración tan detallada entre todas las direcciones de tráfico.

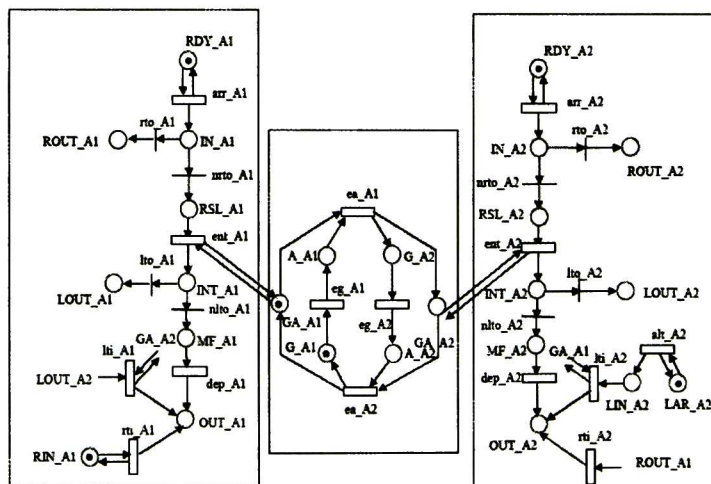


Figura 1-14 Modelo con RP estocástica de una intersección de 2 fases.

Una vez que contemos con el modelo del sistema de tráfico urbano, podemos usarlo para trasladarlo a una computadora y ejecutar la simulación. Existen diferentes paradigmas para hacer lograrlo.

1.5 Paradigmas de modelos de computación para la simulación de tráfico urbano

"Kuhn utilizó el término (paradigma)... para describir un conjunto de teorías, estándares y métodos que juntos representan una manera de organizar el conocimiento; es decir, una manera de ver el mundo. En la tesis de Kuhn las revoluciones en la ciencia ocurren cuando un paradigma más viejo se examina de nueva cuenta, se rechaza y sustituye por otro. Esto es en éste sentido, como un modelo o ejemplo y como acercamiento de organización, que Roberto Floyd utilizó el término en su conferencia del ACM Turing Award, de 1979 que fue titulada "los paradigmas de la programación" (Floyd, 1979]. Un paradigma de programación es una manera de conceptualizar lo que significa realizar una computación y cómo las tareas a ser realizadas en una computadora deben ser estructuradas y organizadas (*Budd, 1995*). Se distinguen los siguientes paradigmas de modelos de computación (*Schunemann,*).

- a) Requisiciones impersonales: actualización imperativa.
- b) Flujo de datos: aplicativo
 - a. Aplicativo puro (incl. Relacional).
 - b. Basado en procesos aplicativo.
- c) Basado en comunicación.
 - a. Paso de mensajes.
 - b. Multi-agente.
- d) Cibernético.

Estamos interesados en el modelo basado en comunicación ya que es el que presenta mayor flexibilidad para representar un sistema de tráfico urbano.

1.5.1 Basada en comunicación

Estos modelos son caracterizados por un tipo de interacción "señal-respuesta" : El cómputo se lleva a cabo por unidades de computo virtual que se comunican una con otra. Diversos estilos

de comunicación y unidades de cómputo conducen a variaciones del modelo, corresponde a una arquitectura de componentes interactuantes.

Modelos de paso de mensajes, modelos Cliente/Servidor o modelos Objeto ("modelos imperativos personales") están muy relacionados con la abstracción del objeto: la acción primitiva característica es que un "objeto" (unidad de cómputo) comunica la información enviando los "mensajes" a otros "objetos" muy específicos. El mensaje es una requisición servicio que identifica a un servicio (operación) y que proporciona los parámetros necesarios. Como reacción al recibir un mensaje, un objeto puede enviar mensajes a otros objetos; puede también haber otras clases de acciones que ocurren internamente en un objeto. Otra acción primitiva es la creación de un nuevo objeto. "El "cómputo es realizado por objetos comunicándose uno con otro, solicitando que otros objetos realizan acciones. Los objetos se comunican enviando y recibiendo mensajes. Un mensaje es un pedido a una acción y que contiene una serie de argumentos que pueden ser necesarios para completar la tarea. Cada objeto tiene su propia memoria, que consiste en otros objetos." (Budd, 1995)

En los modelos *manejados por eventos o multi-agente* las unidades de cómputo son llamadas agentes. La acción primitiva característica es que un agente comunica la información difundiendo "eventos" a todos los "agentes" (que puedan procesarlos o no hacer caso de ellos). Corresponde a una arquitectura manejada por eventos. El modelo multi-agente, estructura un sistema interactivo en una colección de agentes especializados que producen y reaccionan a los eventos. Un agente sistema completo de procesamiento de información: incluye receptores de eventos y transmisores de eventos , una memoria para mantener el estado y un procesador que cíclicamente procesa los eventos de entrada, actualiza su propio estado y puede producir eventos o cambiar su interés en clases de eventos que recibe en la entrada. Los agentes se comunican con otros agentes, incluyendo el operador (usuario de la computadora).

El enfoque basado en sistemas multi-agente (MÁS: multi-agent systems), permite una descripción mucho más detallada (microscópica) que el uso, por ejemplo, de simulaciones macroscópicas, donde el congestionamiento de tráfico, el desvío de tráfico y otros fenómenos son considerados como emergentes. En los modelos MÁS de tráfico vehicular los agentes reaccionan de acuerdo a los vehículos que los rodean, condiciones locales, estructura de camino local y parámetros globales. También la percepción de condiciones de tráfico por los agentes puede usualmente ser suficiente para simular la congestión. El resultado más interesante concluye que

para densidades intermedias, la congestión ocurre dependiendo de la historia más reciente del sistema (*Benenson, 2004*): Si existió congestión, entonces la congestión puede mantenerse, mientras que si fue pequeña, el tráfico tiende a fluir. Analizaremos algunos simuladores, que implementan el modelo del sistema de tráfico urbano usando el paradigma MÁS, para generar el simulador.

1.5.2 Uso de paradigma multi-agente en simulación tráfico urbano

a) Simulador MÁS GLD (*Green Light District*) basado en CA.

Éste simulador (*Wiering, 2003*) contiene un editor que permite definir estructuras urbanas (calles e intersecciones) basada en CA. Existen dos tipos de agentes que pueden ocupar la infraestructura: los vehículos y los semáforos, todos los agentes actúan de manera autónoma, siguiendo algunas reglas muy simples y actualizándose en cada paso de la simulación.

Los vehículos entran a la red en los nodos extremos, cada nodo⁴ extremo tiene una cierta probabilidad de generar un vehículo en cada paso de la simulación, cada vehículo que es generado se le asigna un destino, que puede ser alguno de los nodos de los extremos. La distribución de destinos para cada nodo de los extremos puede ser ajustada.

Existen diferentes tipos de vehículos, definidos por su velocidad máxima, longitud y número de pasajeros. Los carros se mueven a una velocidad de 2 unidades (o uno o cero si tienen que frenar) por paso de simulación, tienen una longitud de dos unidades y tienen dos pasajeros. El estado de cada vehículo es actualizado en cada paso de simulación. El vehículo se mueve cierta distancia de acuerdo a su velocidad, o se para si es que delante de él existe algún otro vehículo o una luz roja en el semáforo. En una intersección el vehículo decide que carril deberá tomar de acuerdo a su plan de viaje. Las intersecciones pueden ser ocupadas por los semáforos. Para cada intersección, existe un número posible de algoritmos (el algoritmo TC-1, que usa aprendizaje reforzado, TC-1 que no toma en cuenta los destinos, TC-1 Bucket, se le agrega el algoritmo bucket (*Wiering, 2003*), TC-1 Co-learning, co-aprendizaje, primero el mejor, ACGJ-3, cola más larga relativamente primero y de manera aleatoria) para hacer el cambio de luces de manera segura. En cada paso de simulación, el controlador decide cual de estos algoritmos es el mejor. Estos pueden usar información de los vehículos que están en espera de cruzar la calle, del destino de estos y de los otros semáforos, para poder tomar una decisión.

⁴ Nodo, se dice de la intersección de 1 o más calles, en una red de tráfico urbana.

b) Simulación de tráfico Multi-nivel

Para el modelado de sistemas de tráfico urbano de varios carriles usando múltiples niveles de abstracción en el contexto de autómatas celulares, el espacio es representado como una rejilla uniforme en el cual los vehículos toman la decisión de avanzar. Se supone que el comportamiento de un sistema de tráfico de varios carriles esta basado en decisiones hechas por los seres humanos, mismas que son tomadas en diferentes niveles de detalle, dependiendo de los perfiles de los conductores (*Claramunt, 2001*). Éste modelo, soporta en principio, diferentes niveles de detalle (microscópico, macroscópico) en el modelo de autómatas celular y razonamiento espacial cualitativo, esto es, diferentes niveles de relaciones espaciales entre los vehículos actuando en un ambiente multi-carril. Además el modelado de tal sistema, debe integrar variedad y aleatoriedad en el tipo de comportamiento individual. Con el fin de replicar esta variedad de comportamiento se propone el modelado multi-agente (MÁS), el cual es relativamente eficiente en la simulación de diferentes tipos de conocimiento a nivel microscópico. Éste modelo propone una simulación de un sistema multi-carril desde un punto de vista microscópico, los comportamientos individuales son modelados a diferentes niveles de abstracción y son usados diferentes tipos de conocimiento y acciones.

c) Herramienta de Simulación: ARCHISIM

En ésta herramienta de simulación de tráfico se hace uso de un submodelo de comportamiento para las decisiones del conductor (*Champion, 2001*). El modelo del conductor resulta de estudios profundos llevados a cabo por situaciones psicológicas reales en el manejo. Así es que los comportamientos no siguen alguna regla previamente definida. En ARCHISIM, el fenómeno de tráfico se genera a partir de acciones individuales y de interacciones de varios de los actores participantes en la red de tráfico. ARCHISIM es un modelo de simulación de comportamiento y su implementación se sigue de los principios multi-agentes (MÁS). Con ARCHISIM, los agentes son conductores simulados en vehículos virtuales y consiste de tres subsistemas: percepción, interpretación, toma de decisión y acción. Ellos se enfocan en la parte de interpretación y toma de decisión. Cada agente tiene un modelo de su ambiente y éste interactúa con otros agentes (carros, trailers, etc.), la infraestructura (semáforos) y el camino, cada agente tiene objetivos y ciertas habilidades. El comportamiento de los agentes no sigue

alguna regla previamente definida. Cada agente tiene su propia actitud. El objetivo no es construir un robot capaz de conducir automáticamente, pero si el estudiar el comportamiento del conductor y la forma en la que los fenómenos de tráfico ocurren, en ARCHISIM los agentes son autónomos y pueden potencialmente reaccionar a cualquier situación. El sistema de tráfico puede entonces mostrar una gran aptitud para organizarse y coordinarse por si mismo. La ventaja de los modelos multi-agentes es que ofrecen un sistema más interactivo y abierto a diferencia de los modelos clásicos, así que es posible modificar dinámicamente las condiciones de la simulación (preferencias de conductor virtuales, algoritmos de control de las luces de tráfico, etc.).

El formalismo tradicional de redes de Petri utilizado para modelar un sistema de red de tráfico urbano captura de manera adecuada el comportamiento del sistema. Sin embargo, no se encontró un mecanismo que nos permita convertir el modelo conceptual en un modelo computacional de manera eficiente. Los demás formalismos, de igual manera capturan las características del sistema, pero carecen de la característica jerárquica que reduce la complejidad del diseño de un simulador de tráfico urbano.

Los simuladores actuales utilizan la técnica de avance de tiempo de la simulación por división de tiempo, que resulta ser menos eficiente que la técnica del de evento siguiente. No existe simulador que utilice ésta técnica, el paradigma de programación más utilizado es el basado en multi-agente por representar de manera natural el comportamiento de las entidades en el sistema de tráfico urbano.

Capítulo 2

DISEÑO DE UN SIMULADOR DE TRÁFICO URBANO

Resumen. En éste capítulo se presenta la especificación inicial de un simulador de tráfico urbano. Se concibe como un micro-simulador orientado a eventos basado en el enfoque de los sistemas multi-agente, donde los componentes son modelados usando un formalismo de redes de Petri anidadas a tres niveles.

2.1 Requerimientos generales

En esta sección se presenta lo que se quiere obtener del simulador y cuales son los beneficios del desarrollo del simulador.

2.1.1 Enunciado de Problema / Oportunidad

Oportunidades:

- a) La reducción del tiempo-costo computacional permitirían incrementar la cantidad de nodos y enlaces que podamos simular, sin incrementar los costos con arquitecturas más complejas y costosas, usando la técnica de avance de tiempo del evento siguiente.
- b) Contar con una plataforma de simulación que soporte comportamientos complejos (multi-agente), que permita evaluar los sistemas ATM y probar el impacto en los conductores de nuevas tecnologías ITS así como su eficiencia.
- c) Contar con una plataforma que modele la mayor cantidad de clases de medios de transporte, que permita tener un control mayor y de gran exactitud en las pruebas de eficiencia de controles de tráfico y en la predicción de transporte urbano (área de planeación).

Por lo tanto para desarrollar un simulador de tráfico urbano que permita evaluar diferentes estrategias de control, diferentes comportamientos multi-agente y eficiente se mencionan a continuación los objetivos del proyecto

2.1.2 Enunciado de los Objetivos del Proyecto

Objetivos de la implementación del proyecto.

- a) Reducir Potencialmente el tiempo de simulación.
- b) Poder implementar diferentes tipos de comportamientos.
- c) Analizar diferentes estrategias de control.
- d) Probar tecnologías de ITS.
- e) Cumplir con los requerimientos necesarios para poder funcionar en la planeación urbana (tenga la mayor cantidad de clases de medios de transporte) (*Beimborn, 1996*).

Objetivos del Departamento (sistemas de eventos discretos).

- a) Generar una herramienta de simulación orientada a eventos.

- b) Involucrarse en el tema de tráfico urbano.
- c) Generar conocimiento para mejora de la sociedad.

2.1.3 Beneficios Esperados

El proyecto tiene como objetivo desarrollar un sistema que permitirá en primera instancia a los ingenieros de tráfico y al área de sistemas de eventos discretos:

Tener una plataforma para la implementación de nuevas metodologías de control de semáforos (tráfico urbano) y de aprendizaje (multi-agente).

2.2 Requerimientos Funcionales

La tabla 1-1 muestra los requerimientos funcionales para el sistema. El beneficio de una característica muestra que tan importante es esta para lograr las tareas principales del sistema.

Siendo de tres tipos:

Crítico: Característica es necesaria.

Importante: Característica es usable y realizable.

Útil: Característica que pudiese mejorar ciertos aspectos del sistema.

Cada característica tiene un estado, el cual puede tomar alguno de los siguientes valores:

Propuesta: Característica puede ser realizada en el futuro.

Aceptada: Característica que será implementada.

Realizada: Característica que ya ha sido implementada.

ID	Característica	Beneficio	Estado
Manejo de Eventos			
1	Eventos de manejo de Mouse (Clic, Mover, Seleccionar, Liberar,...)	Crítico	Aceptada
2	Manejo de eventos del teclado	Importante	Aceptada
Control de semáforos			
3	Herramienta interfaz que permita al simulador trabajar con cualquier tipo de control de señal adoptivo o de tiempo real, como TRANSY, SYNCHRO o el estándar Nema.	Crítico	Aceptada
Control de Simulación.			
4	Técnica del evento siguiente.	Crítico	Aceptada
Personalizado (Perfiles de usuario)			
5	Estado de la GUI puede ser salvado y cargado.	Útil	Propuesta
GUIs 2D			
6	Una interfaz gráfica que pueda usar mapas digitales ya existentes de las redes urbanas: dxf, jpg, bit map, etc.	Crítico	Aceptada
7	Una interfaz que permita editar al usuario cualquier tipo de instalaciones de tráfico urbano	Importante	Aceptada
8	Salida animada en 2D, para que el usuario pueda ver el avance de la simulación.	Útil	Aceptada
Modelado de Comportamientos			
9	Una interfaz que permita modelar el comportamiento individual de los vehículos.	Importante	Aceptada
GUIs 3D			
10	Una interfaz 3D que permita al usuario ver el avance de la simulación en 3D.	Útil	Propuesta

Tabla 2-1 Lista de requerimientos funcionales.

2.2.1 Propuesta y justificación de soluciones a requerimientos funcionales.

Uso de la técnica evento siguiente(requerimiento ID # 4).

Decidimos utilizar la técnica de evento siguiente porque tiene dos ventajas sobre el enfoque de división de tiempo. La primer ventaja es que el incremento de tiempo se ajusta de manera

automática a los periodos de actividad alta y baja, así se evita la comprobación inútil e innecesaria del modelo. La segunda ventaja es la claridad con la que representa un evento significativo que ha ocurrido en la simulación. La desventaja de esta técnica, es que se debe de tener mucho más información para el control de la simulación y que el tiempo de simulación no fluye suavemente. Por supuesto que hay algunos sistemas que sus acontecimientos ocurren a intervalos regulares. Por ejemplo, un supermercado puede comprobar los niveles de existencias a la misma hora cada día y para surtirla puede ser a una hora predeterminada. En tales casos, es definitivamente más adecuado utilizar el enfoque de división de tiempo, sin embargo, debe ser observado que la técnica de evento siguiente es más general, ya que si los eventos en un sistema ocurren en intervalos regulares (una vez por día, quizás), la técnica del evento siguiente actuará como si fuera un enfoque de división de tiempo. Sin embargo, al contrario no es cierto.

En la literatura existen investigaciones que demuestran que la técnica del evento siguiente resulta ser mucho más efectiva que el método de división de tiempo al ser usada en simulación de tráfico. Existen investigaciones para probar las diferencias existentes entre los métodos división de tiempo y evento siguiente usadas en un simulador (*Schulze, 1997*). El resultado demuestra que la técnica de evento siguiente es por mucho, la mejor ya que resulta más económica computacionalmente. El objetivo de la investigación fue determinar la influencia de los dos métodos de avance del tiempo, en los tiempos de ejecución de la simulación, ejecutándose en una PC con un procesador Pentium a 120 Mhz.

Los parámetros usados fueron los siguientes:

- Para división de tiempo, las divisiones o incrementos de tiempo fueron de 1.0, 0.5 y 0.2 segundos.
- La carga vehicular varía en 3 niveles. Comenzando con el nivel básico de carga (Nivel 1), la carga fue incrementada en un 50%. Esto significa :
 - Nivel 2 = 1.5* Nivel 1 y
 - Nivel 3 = 2.0 * Nivel 1

La tabla 2-1 contiene los tiempos de ejecución obtenidos para la simulación de un escenario (todos los tiempos están en segundos) .

	Carga de nivel 1	Carga de nivel 2	Carga de nivel 3
Técnica evento siguiente	94 ± 1.0	150 ± 1.7	507 ± 19.9
División = 1 seg.	146 ± 1.6	245 ± 3.8	908 ± 20.0
División = 0.5 seg.	253 ± 3.3	409 ± 6.0	1342 ± 47.6
División = 0.2 seg.	584 ± 7.3	934 ± 12.9	2782 ± 76.6

Tabla 2-2 Tiempos de ejecución

2.3 Descripción del sistema

Esta sección describe al sistema en forma más detallada. Primero se muestra la arquitectura del sistema, seguido por el modelado jerárquico. Al final de esta sección se describen algunos productos de terceras partes usadas en el desarrollo del sistema.

2.3.1 Arquitectura

El simulador es desarrollado usando lenguaje Java. El simulador esta planeado para ser una aplicación auto soportada para minimizar los problemas y limitaciones que el uso del applet pueda generar.

Descripción orientada a eventos del sistema

Para poder especificar nuestro sistema como DES, debemos encontrar las entidades que existen en el sistema y definir los principales eventos que se generan por las relaciones entre ellos. Usando los siguientes conceptos, definimos nuestros objetos y entidades (Banks, 2000) : el *sistema*, es una colección de entidades (por ejemplo, personas y maquinas) que interactúan entre ellas para completar uno o varios objetivos. El *modelo* es una representación abstracta de un sistema, usualmente contiene relaciones lógicas y/o matemáticas, las cuales describen un sistema en términos de estado, entidades y sus atributos, conjuntos, eventos, actividades y retardos, en esta parte nos apoyaremos en la metodología de modelado 3-LNS, que veremos en la siguiente sección. El *estado del Sistema* es una colección de variables que contienen toda la información necesaria para describir el sistema en cualquier instante de tiempo. Una *entidad* es cualquier objeto o componente en el sistema el cual requiere una representación explícita en el modelo (por ejemplo, un servidor, un cliente, una máquina). Los *atributos* son las propiedades de una entidad

dada (por ejemplo, la prioridad de espera de un cliente y la ruta de un trabajo a través de un servidor de trabajos). Un *conjunto* es una colección de entidades (permanente o temporal), ordenada en alguna forma lógica (como los clientes en una línea de espera ordenados por el primero en llegar, el primero en salir). Un *evento* es la ocurrencia instantánea que cambia el estado de un sistema (tal como el arribo de un cliente). La *actividad* es un intervalo de tiempo de duración específica (por ejemplo, la duración de algún servicio o la duración de interarribos⁵). El *retardo* es una cantidad de tiempo cuya duración no es conocida hasta que termina (por ejemplo el retardo de un cliente en una línea de espera de tipo “último en entrar, primero en salir”, el cual dependerá de futuros arribos de clientes). En la tabla 2-1 se muestra la especificación de un sistema de tráfico urbano como DES.

Estado del Sistema: L(t), LQ1(t), LQ2(t), MT(t)	Eventos:	Entidades:	Conjuntos:
L(t) = Núm. de vehículos en calle	(AE,t,car _i) car _i arriba a la calle en el instante t	Vehículo , (car _i ,t) vehículo i arriba en el instante t.	cars, todos los vehículos en la calle.
LQ1(t) = Núm. vehículos en carril 1.	(SE,t, car _i) , car _i para en el instante t.		carPosition, pos vehículos en calle.
LQ2(t) = Núm. vehículos en carril 2.	(CE,t, car _i) , car _i cruza la calle en el instante t.		carline_1, vehículos 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 Continúa el avance en el instante t.		
	(LCE,t,light _i) , light _i cambio el estado del semáforo en el inst t.		

Tabla 2-3 Especificación DES de un Sistema de Tráfico Urbano.

Los sistemas considerados son dinámicos, por lo que el estado del sistema, atributos de las entidades, el número de actividades activas, el contenido de los conjuntos y las actividades y los retardos actualmente en progreso son todas funciones del tiempo y su estado siguiente dependerá de su estado actual o el estado actual depende del estado anterior de las variables. El avance de

⁵ En el capítulo 1, teoría de colas, se describe éste término.

tiempo es representado por una variable llamada CLOCK.

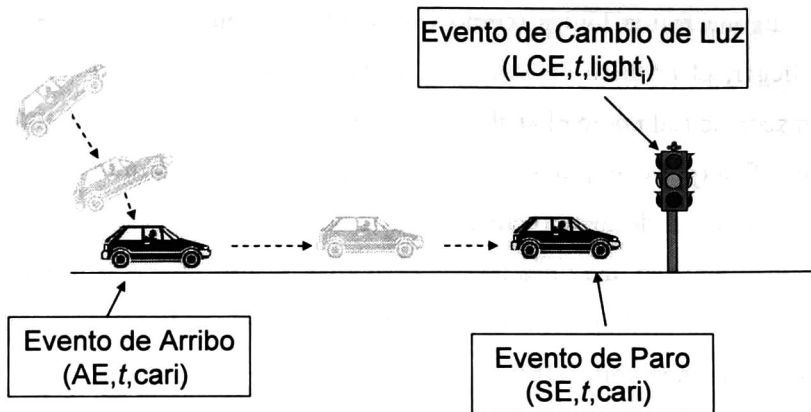


Figura 2-1 Relación de eventos arribo, paro y cambio de fase.

En la figura 2-1 se muestra la relación de los eventos de arribo, de paro y de cambio de luz en una calle. En la figura 2-2 se muestra los eventos de cambio de carril y de iniciar a avanzar después de un cambio de carril.

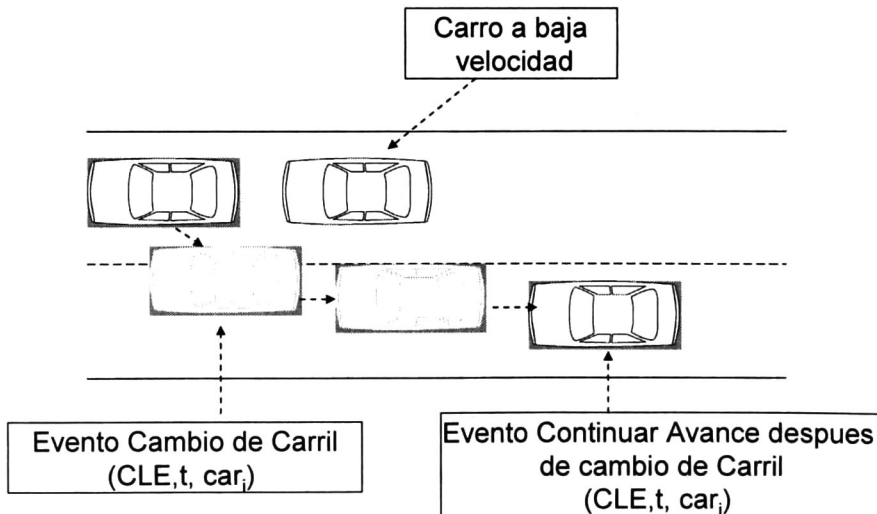


Figura 2-2 Relación de eventos cambio de carril y avance después del cambio.

Planteamiento del simulador basado en enfoque multi-agente

El uso del paradigma multi-agente se presenta en la figura 2-3, que permite observar los agentes involucrados en la simulación del sistema:

- a) *Agente intersección*: Encargado de coordinar las diferentes estrategias de control de semáforos, así como de sincronizar los diferentes eventos que ocurren en esta intersección de cada una de las calles que se conectan a esta intersección.
- b) *Agente calle*: Encargado de transmitir la información de las entidades que se encuentran en la vecindad de cada vehículo, así como el estado de los semáforos y de eventos posibles a ocurrir en la intersección.
- c) *Agente vehículo*: Es la entidad principal del sistema de tráfico urbano ya que de esta dependen los diferentes fenómenos que observamos, como la congestión, por ejemplo. Éste se comunica con el agente calle, para pedir información del estado de su ambiente.

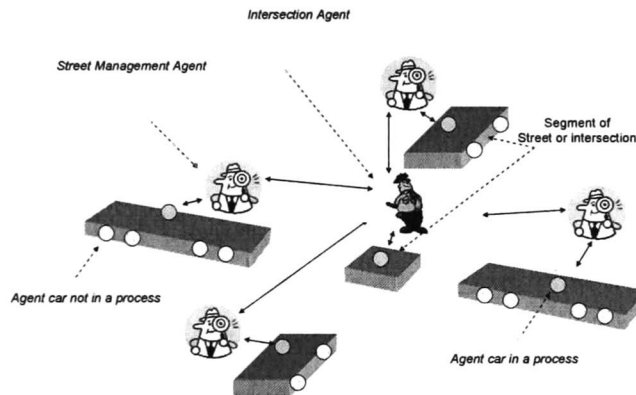


Figura 2-3 Sistema Multi-agente de sistema de tráfico urbano.

En la figura 2-4 se presentan un diagrama de las principales clases del sistema.

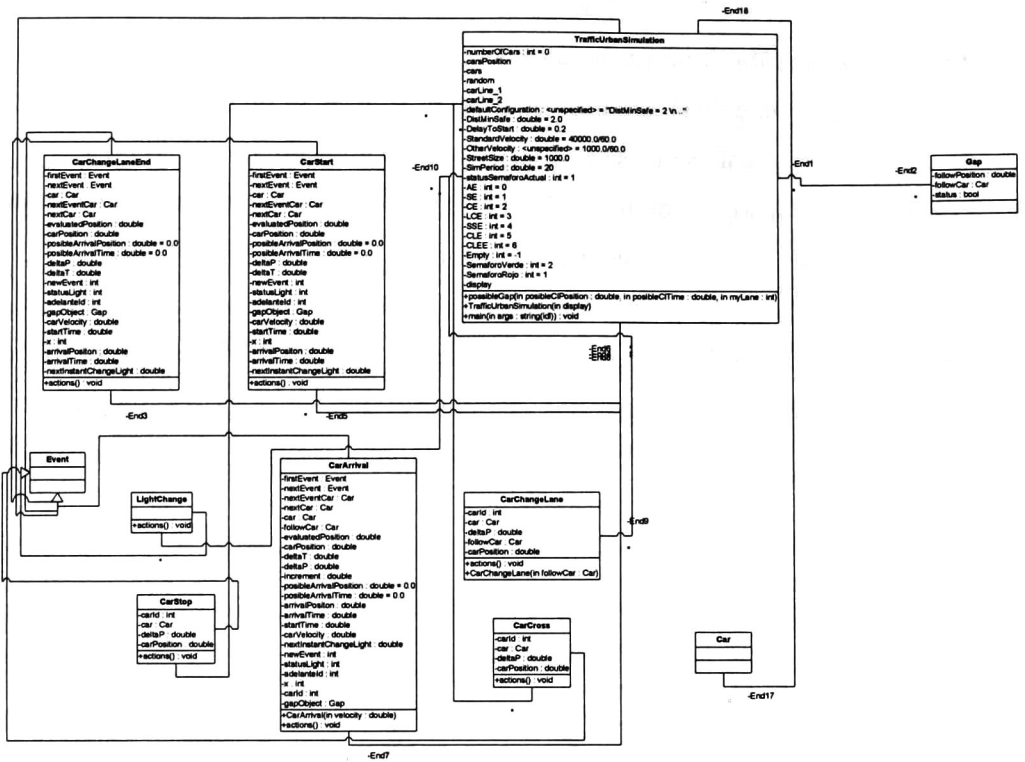


Figura 2-4 Diagrama de clases del simulador.

En la figura 2-5 y 2-6 se presentan las diferentes comunicaciones que se pueden implementar entre los agentes. En el capítulo 4, se analiza la implementación multi-agente del simulador un poco más detallada.

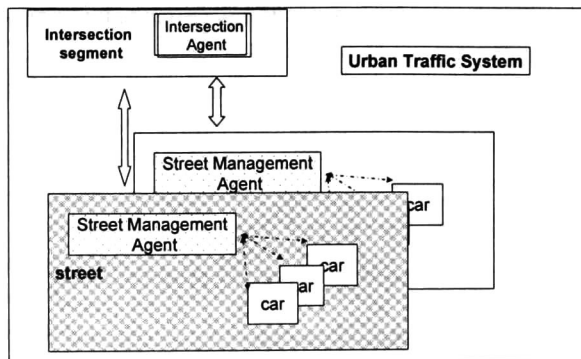


Figura 2-5 Comunicación entre el agente intersección y el calle.

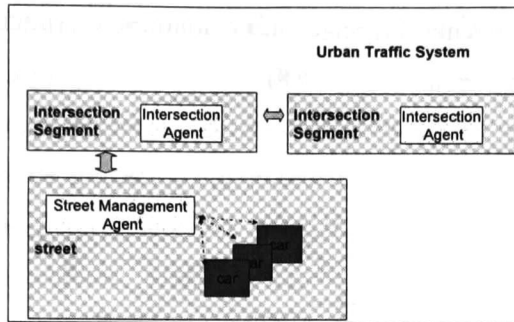


Figura 2-6 Comunicación entre los agentes intersección.

2.3.2 Modelado de sistemas de tráfico urbano

Como ya vimos en el capítulo 1, existen diferentes tipos de formalismos para el modelado de sistemas de tráfico urbano: autómatas celulares, redes de colas, teoría de partículas, etc. Con el fin de poder enfrentar la especificación de sistemas grandes y complejos, como las redes urbanas del tráfico, el formalismo de redes de Petri 3-LNS es el más conveniente, debido a sus características de naturaleza gráfica y el soporte de técnicas de análisis matemático: claridad y simplicidad para representar el comportamiento complejo, como causalidad, concurrencia, paralelismo, sincronización, toma de decisiones, e intercambio de información. Además, de la capacidad de representar de manera jerárquica y modular al sistema. En la figura 2-7, se muestra como se modela la red de tráfico con la red de Petri. La esfera (marca) representa el modelo del vehículo y los círculos (lugares) representan las calles e intersecciones.

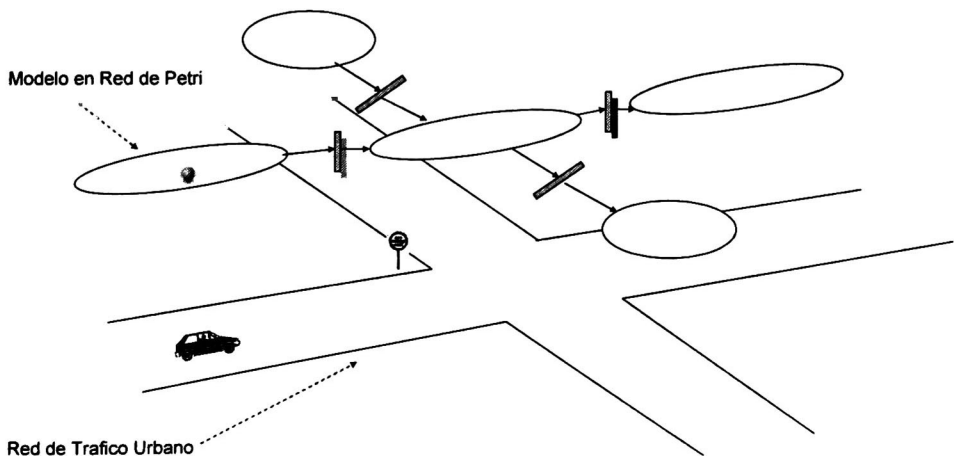


Figura 2-7 Modelo en redes de Petri de la red de tráfico urbano.

Además podemos representar las relaciones comunicación (interacciones) entre las entidades, así como el control de las mismas (figura 2-8):

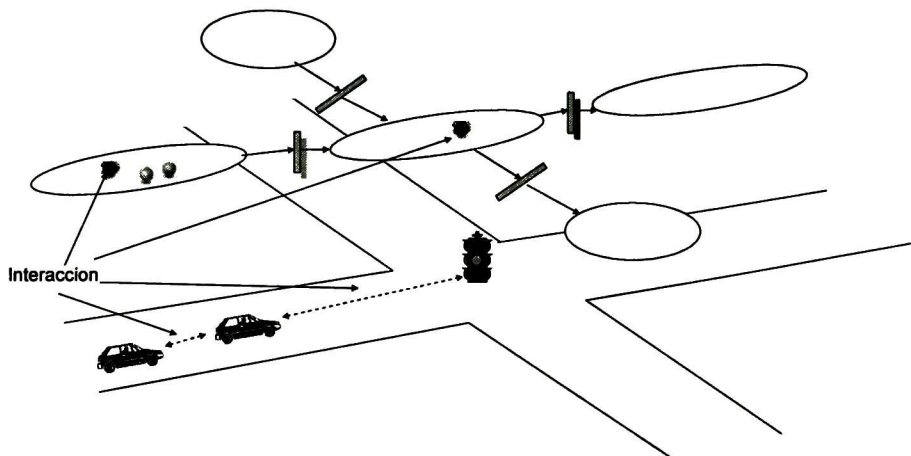


Figura 2-8 Modelo en RP :3LNS, de un sistema de tráfico urbano.

Modelando la red de tráfico urbano con el formalismo 3-LNS, nos da la capacidad de representar la movilidad de los vehículos de una manera intuitiva a través del movimiento de las marcas dentro de la red. Generamos un modelo libre de detalles de implementación, que nos conduce a una estrategia para modelar de manera modular y jerárquica y además de contar con una definición formal de nuestro sistema, todo es nos permite reforzar y tener una mejor implementación de la fase de implementación utilizando la metodología de multi-agentes.

2.3.3 Productos de terceras partes

El sistema usara los siguientes productos:

- JAVASIMULATION paquete para java TM.

Existen en la literatura diferentes herramientas orientados específicamente ala implementación de simulación de eventos discretos, como: Arena, AutoMod, Deneb/QUEST ,Extend, Micro Saint, ProModel, Taylor ED, WITNESS (PC-221),SLX y GPSS/H entre otros (PC-190).

La ventaja de la simulación de eventos discretos, como ya se menciona anteriormente, se puede implementarla en cualquier lenguaje de propósito general, c, c++, pascal, java, etc.(PC-221). Hemos decidido usar la herramienta java, por su capacidad de trabajar como distribuida, su

actualidad y valor actual que tiene como lenguaje de programación distribuida.

Sin embargo, aunque la plataforma de Java tiene varios paquetes estándar, no existe paquete para la simulación de eventos discretos, esto es desafortunado ya que la simulación de eventos discretos constituye un área de aplicación importante de la programación orientada a objetos, éste hecho está perfectamente demostrado por SIMULA, uno de los primeros lenguajes de programación orientado a objetos. SIMULA provee la clase estándar SIMULATION, una herramienta sumamente poderosa para la simulación de eventos discretos. JAVASIMULATION, es un paquete Java para simulación de eventos discretos basado en procesos. Éste paquete puede ser visto como una implementación de la clase SIMULATION en Java. Además de las funciones de simulación, el paquete incluye los medios para la manipulación de listas y la generación de números aleatorios tal y como lo encontramos en SIMULA.

La especificación del simulador que se propone cuenta con las siguientes características:

- a) Micro-simulador.
- b) Discreto.
- c) Estocástico.
- d) Basado en multi-agente.
- e) Usa la técnica de avance de tiempo de evento siguiente.

El modelo del simulador que captura el comportamiento del sistema de tráfico urbano, se realiza con el formalismo de redes de Petri anidadas a tres niveles: 3-LNS.

Capítulo 3

MODELADO DE SISTEMAS DE TRÁFICO URBANO CON REDES DE PETRI

Resumen. El presente capítulo aborda la construcción de modelos de sistemas de tráfico urbano (STU) usando 3-LNS, un formalismo basado en redes de Petri el cual permite representar en tres niveles los componentes de un sistema de agentes móviles. Un modelo de STU describe en el nivel 1 el ambiente del STU, en este caso la infraestructura de la red de tráfico, en el nivel 2, los agentes del STU, tales como, vehículos, semáforos, etc. y en el nivel 3 los componentes de los agentes, como planes de trayecto, habilidades y actitudes de los conductores, etc.

3.1 Un sistema de red a tres niveles: 3-LNS

3-LNS es un sistema que permite definir modelos de redes de Petri (RP) anidadas a tres niveles: el marcado de una RP pueden ser otras RP hasta un nivel de anidamiento de 2 por lo que se distinguen tres niveles. El marcado de las RP de nivel 3 consiste de símbolos como en las RP coloreadas; la figura 3-1 ilustra esta idea.

El sistema de red fue inicialmente definido para el modelado de sistemas multiagentes móviles (Almeyda 2002] y posteriormente fue utilizado para especificar sistemas de robots móviles (López 2005b) y de manufactura por lotes (Villanueva 2003)(López 2005a]. Incluiremos aquí la definición del formalismo para una mejor comprensión de los modelos presentados.

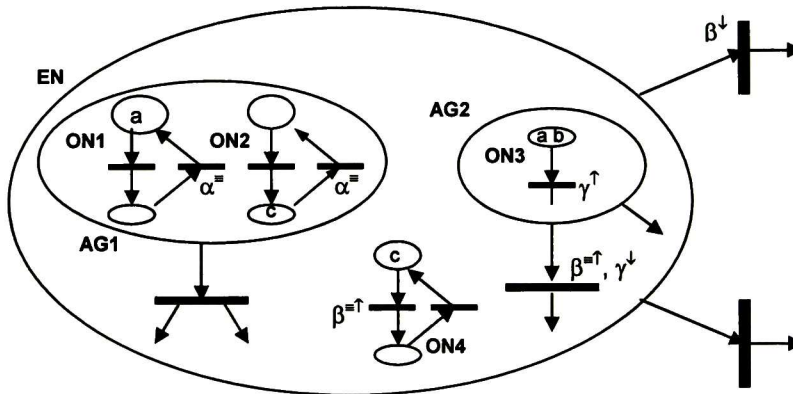


Figura 3-1 Segmento de una red de tres niveles.

3.1.1 Redes de Petri generalizadas

a) Conceptos básicos

Una red de Petri (RP) es un digrafo bipartido con un marcado. Los dos tipos de vértices son los lugares y las transiciones unidos por arcos. En un modelo gráfico, los lugares son representados por círculos y las transiciones por barras o cajas (Figura 3-1). Un marcado (función de marcado) asigna a cada lugar un entero no negativo. Un lugar p al que se le asigna el número entero no negativo k , se dice que es marcado con k marcas. Esto es representado por k puntos (o círculos), o por el número k , dentro del círculo representando a p .

Definición 3-1. Una RP generalizada es una tupla $N = (G, \pi, \mu)$ donde:

- $G = (P, T, F)$ es una estructura de red.
- P es un conjunto de lugares finito no vacío.
- T es un conjunto de transiciones finito no vacío.
- $P \cap T = \emptyset$.
- $F \subseteq P \times T \cup T \times P$ es una relación de flujo de la red.
- $\pi: F \rightarrow \mathbb{Z}^+$ es una función representando el peso de los arcos.
- $\mu: P \rightarrow \mathbb{Z}^+$ es una función representando el marcado (distribución de marcas) de la red.

3.1.2 Evolución del marcado de redes de Petri

El comportamiento dinámico de un sistema puede ser descrito por la evolución de las marcas en la red que lo representa. Las transiciones desempeñan un papel importante, cada transición es responsable del “movimiento” de las marcas de sus lugares de entrada (lugares que están conectados a la transición por alguno de los arcos que entran a ésta) a sus lugares de salida (aquellos lugares a los que se conectan los arcos que salen de la transición).

3.1.3 Regla de Disparo

Una transición t_j puede ser disparada ssi está habilitada; esto es, que todos sus lugares de entrada tengan marcas, tantas como el peso de los arcos de entrada. El disparo de una transición, que puede ser considerado instantáneo, causa un cambio en el marcado. Cada lugar de entrada reduce el número de marcas en la cantidad indicada por el peso asociado al arco de entrada y cada lugar de salida incrementa el número de marcas en la cantidad indicada por el peso asociado al arco de salida.

3.1.4 Sistema de red a tres niveles

En un sistema de red a tres niveles llamaremos a red de nivel más alto, la *red ambiente* o de nivel 1. A las redes de nivel 2 se les denomina *redes agente* y a las redes de nivel 3, *redes objeto*.

Definición 3-2 Una *red objeto* es una tupla $NETobj = (TYPEobj, \mu)$ donde:

- $TYPEobj = (G, SYMB, LABELS, \tau, \lambda, \pi)$ es un tipo-objeto.
- G es una estructura RP
- $SYMB$ es un conjunto finito de símbolos.
- $LABELS$ es un conjunto finito de etiquetas.
- $\tau: P \rightarrow 2^{SYMB} - \emptyset$ es una función de asignación de símbolos a los lugares.
- $\lambda: T \rightarrow 2^{LAB} - \emptyset$ es una función de asignación de etiquetas a las transiciones.
- $LAB = (LABELS \cup \{\epsilon\}) \times (LABELS \cup \{\epsilon\})$
- $\pi: F \times LABELS \rightarrow M_{SYMB}$ es una función que representa el peso de los arcos con respecto a las etiquetas de las transiciones. (M_{SYMB} es la familia de todos los multiconjuntos sobre $SYMB$).
- $\mu: P \rightarrow M_{SYMB}$ es una función de marcado.

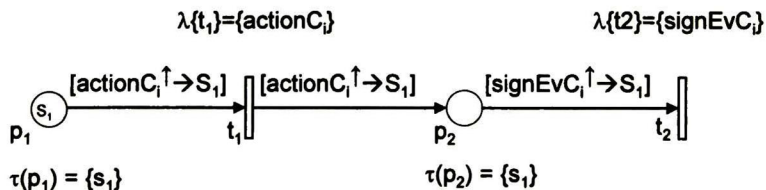


Figura 3-2 Red Objeto.

La noción de una red objeto (figura 3-2) es muy parecida a la de RP coloreada; se define, como un tipo de red con un marcado; el tipo ($TYPEobj$) es una estructura RP con información

adicional. *SYMB* y *LABELS* juegan un papel muy similar al de los colores asociados a los lugares y transiciones respectivamente; τ y λ representan esta asignación.

Además, los pares resultantes de la función $\lambda(t_j)$, establecen que la transición se puede sincronizar localmente (con transiciones de otras red objeto) y de manera externa (con otras transiciones de la red agente), excepto si este par es (ϵ, ϵ) . Para cada arco, la función π relaciona las etiquetas de las transiciones con los multiconjuntos de los símbolos de los lugares, estableciendo condiciones *PRE* y *POST* para el disparo de la transición.

Definición 3-3. Una red agente es una tupla $NETag = (TYPEag, \mu)$ donde:

- $TYPEag = (G, TYPESo, LABELS, VARS, \tau, \lambda, \pi)$ es un tipo-agente.
- G es una estructura RP.
- $TYPESo$ es un conjunto finito de redes tipo-objeto.
- $VARS$ es un conjunto finito de variables.
- $\tau: P \rightarrow 2^{TYPESo} - \emptyset$ es una función de asignación de tipos a los lugares.
- $\lambda: T \rightarrow 2^{LAB} - \emptyset$ es una función de asignación (de subconjuntos) de las etiquetas a las transiciones. $LAB = (LABELS \cup \{\epsilon\}) \times (LABELS \cup \{\epsilon\}) \times (LABELS \cup \{\epsilon\})$
- $\pi: F \times LABELS \rightarrow M_{VARS}$ es una función de asignación de las variables (con respecto a las etiquetas de las transiciones) a los arcos.
- $\mu: P \rightarrow M_{NETSobj}$ es una función de marcado. $NETS_{obj}$ es el conjunto de todas las redes objeto.

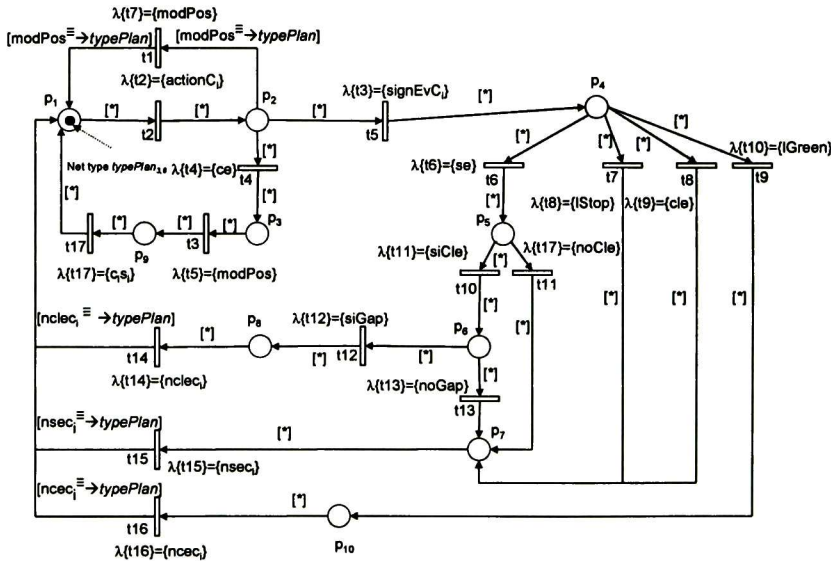


Figura 3-3 Red agente.

De manera muy similar a las redes objeto, una red agente (figura 3-3) es un tipo de red con un marcado; el tipo $TYPEag$ es una estructura RP con información adicional que define los datos que son manipulados ($TYPESo$, τ), el funcionamiento interno ($VARS$, π) y la interacción con otras redes : una tripleta en $\lambda(t_j)$ que no sea $(\epsilon, \epsilon, \epsilon)$ asocia respectivamente las etiquetas para sincronización local , interna (con otras transiciones de las redes objeto) y externa; μ proporciona la distribución actual de las redes objeto en los lugares.

Definición 3-4. Una red ambiente es una tupla $NETenv = (TYPEenv, \mu)$ donde :

- $TYPEenv = (G, TYPESoa, LABELS, VARS, \tau, \lambda, \pi)$ es un tipo-ambiente.
- G es una estructura RP.
- $TYPESoa$ es un conjunto finito de tipos-objeto y tipos-agente.
- $LABELS$ es un conjunto finito de etiquetas.
- $VARS$ es un conjunto finito de variables.
- $\tau: P \rightarrow 2^{TYPESoa} - \emptyset$ es una función de asignación de tipos.
- $\lambda: T \rightarrow 2^{LABELS \cup \{e\}} - \emptyset$ es una función de asignación de etiquetas.
- $\pi: F \times LABELS \rightarrow M_{VARS}$ es una función de asignación de variables (con respecto a las etiquetas de las transiciones) a los arcos.
- $\mu: P \rightarrow M_{NETS}$ es una función de marcado. $NETS$ es el conjunto de todas las redes objeto y agente del sistema de red.

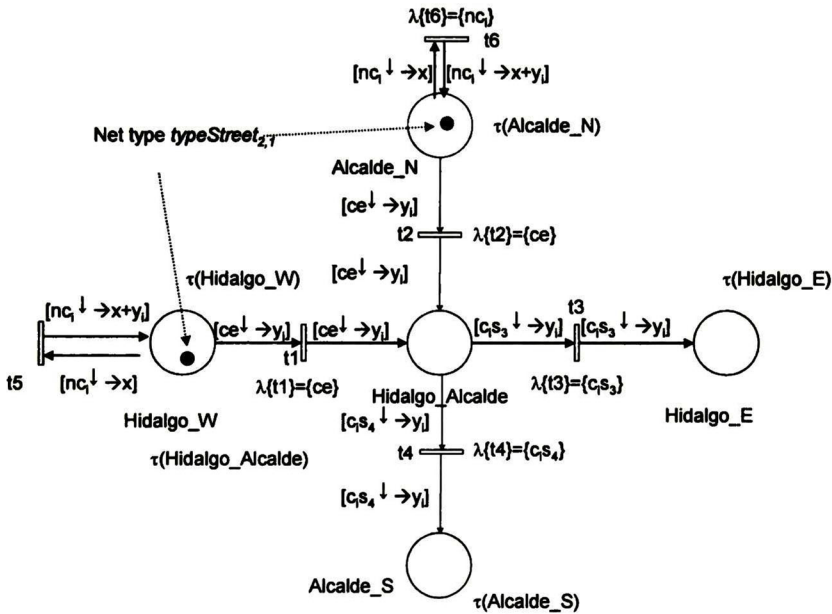


Figura 3-4 Red Ambiente.

Una red ambiente (figura 3-4) es una estructura RP con información adicional que define los datos que son manipulados ($TYPEs, \tau$), el funcionamiento interno ($VARS, \pi$) y la sincronización con las redes agente y objeto contenidas ($LABELS, \lambda$); μ proporciona la distribución actual de las redes objeto y agente dentro de los lugares.

Definición 3-5. Un sistema de red de tres niveles es una tripleta $NS = (EN, AG, OBJ)$, donde:

- EN es una red ambiente $NETenv$ (nivel 0).
- $AG = \{NETag_1, NETag_2, \dots, NETag_k\}$ es un conjunto de redes agente (nivel 1).
- $OBJ = \{NETobj_1, NETobj_2, \dots, NETobj_r\}$ es un conjunto de redes objeto (nivel 2).

La figura 3-1 ilustra estas definiciones; en esta figura se muestra una parte de un modelo 3-LNS, incluyendo la red ambiente (*EN*), dos redes agente (*AG1*, *AG2*) y cuatro redes objeto (*ON1*, *ON2*, *ON3*, *ON4*). En un modelo gráfico, las etiquetas asociadas a las transiciones establecen la sincronización local, interna y externa, usando los superíndices \equiv , \downarrow y \uparrow respectivamente.

3.1.5 Evolución del sistema de red

Un sistema de red es un modelo multicomponente modular donde la interacción entre los módulos es soportada por la sincronización entre las transiciones, que es definido por el etiquetado simbólico de las transiciones.

En todos los niveles las transiciones pueden dispararse de manera autónoma o sincronizada. Considerando esta característica, se describen las reglas de habilitación y disparo de las transiciones de la red.

Para expresar las condiciones de habilitación y de disparo para las redes agente y la red ambiente, es necesario introducir algunas definiciones sobre las *variables* y *ligado*, usadas en esta tesis.

Definición 3-6. Una *variable* $v \in VARS$ puede representar cualquier red objeto o agente. El tipo *Type* (v) de una *variable* puede ser *TYPEag* o *TYPEobj*. *Type* (Var) nos da el conjunto de variables de $Var \subseteq VARS$.

Definición 3-7. Sea *NETS* un conjunto que contiene redes objeto o redes agente. Una *ligado* b sobre un conjunto *variable* $Var = \{x, y, \dots\}$ es una función $b: Var \rightarrow NETS$ tal que para toda $v \in Var$, $b(v)$ es una red objeto o red agente que es de tipo *Type* (v). $m_{\leftarrow b}$ denota un multiconjunto de redes, resultado de hacer una instancia de un multiconjunto de variables m con ligado b .

Evolución de las redes objeto

Una transición t_j de una red objeto puede dispararse ssi está habilitada; t_j es habilitada con respecto a la etiqueta lab , si $\forall p, \mu(p) \supseteq \pi((p, t_j), lab)$ y las condiciones de alguno de los casos siguientes se satisfacen:

Caso 1. Si $lab = (\epsilon, \epsilon)$ entonces no es requerida una condición adicional. El disparo de t_j es ejecutada de manera autónoma.

Caso 2. Si $lab \neq (\epsilon, \epsilon)$, deberá ser considerada alguna de las siguientes situaciones :

- i) $lab = (l, \epsilon)$. Se requiere la habilitación simultánea de las transiciones etiquetadas con \bar{l} que pertenecen a otras redes objetos en el mismo lugar de una red agente o red ambiente. El disparo de estas transiciones es ejecutado de manera simultánea y todas las redes (localmente) sincronizadas se mantienen en el mismo lugar.
- ii) $lab = (\epsilon, l)$. Se requiere la habilitación simultánea de las transiciones etiquetadas con lab^\uparrow que aparecen en la salida del lugar (de una red agente o red ambiente) donde la red objeto esta incluida. El disparo de estas transiciones provocan la transferencia de redes.
- iii) $lab = (l, l)$, denotada por \bar{l}^\uparrow , representa que ambos tipos de sincronización son requeridos, entonces las situaciones i) y ii) son aplicadas simultáneamente.

Evolución de las redes agente

Una transición t_j de una red agente $NETag_i$ está habilitada con respecto a la etiqueta lab si :
Existe un ligado $b_{tj}: VARS_{tj} \rightarrow OBJ$, donde $VARS_{tj}$ es el conjunto de las variables que aparecen en todos los $\pi((p, t_j), lab)$.

Debe satisfacerse que $\forall p, \mu(p) \supseteq \pi((p, t_j), lab)_{<b_{tj}>}$, y las condiciones de alguno de los siguientes casos :

Caso 1. Si $lab = (\varepsilon, \varepsilon, \varepsilon)$ entonces no se requiere de una condición adicional. El disparo de t_j es ejecutado de manera autónoma.

Caso 2. Si $lab \neq (\varepsilon, \varepsilon, \varepsilon)$ debemos considerar uno o varias de las situaciones siguientes:

- a) $lab = (l, \varepsilon, \varepsilon)$. Se requiere la habilitación simultánea de las transiciones etiquetadas con el símbolo Γ que pertenecen a otras redes dentro del mismo lugar p' de la red ambiente y las redes objeto marcando los lugares de entrada de t_j . El disparo de estas transiciones es ejecutado simultáneamente y todas las redes (localmente) sincronizadas se mantienen dentro de p' .
- b) $lab = (\varepsilon, l, \varepsilon)$. También es requerida la habilitación de las transiciones etiquetadas con el símbolo Γ^\uparrow de las redes objeto, contenidas dentro de $\bullet t_j$. El disparo de t_j provoca la transferencia de las redes objeto (aquellas cuyas transiciones también son disparadas) declaradas dentro de $\pi((p, t_j), \Gamma^\uparrow)_{<b_{tj}>}$.
- c) $lab = (\varepsilon, \varepsilon, l)$. También es requerida la habilitación de al menos una de las $t' \in p' \bullet$, etiquetadas con Γ^\downarrow , de la red ambiente. El disparo de t' provoca la transferencia de la red $NETag_j$ (y otras $NETobj$) declaradas dentro de $\pi((p', t'), \Gamma^\downarrow)_{<b_{t'}>}$.

Evolución de la red ambiente

Una transición t_j de la red ambiente EN es habilitada con respecto a una etiqueta lab si :

- Existe un ligado $b_{tj}: VARS_{tj} \rightarrow OBJ \cup AG$, donde $VARS_{tj}$ es el conjunto de variables que aparecen en todos los $\pi((p, t_j), lab)$ y
debe satisfacer $\forall p, \mu(p) \supseteq \pi((p, t_j), lab)_{<b_{tj}>}$ y las condiciones para alguno de los siguientes casos:

Caso 1. Si $lab = \varepsilon$ entonces no se requiere de una condición adicional. El disparo de t_j es ejecutado de manera autónoma.

Caso 2. Si $lab = \Gamma^\downarrow$ entonces se requiere de la habilitación simultanea de las transiciones etiquetadas con el símbolo Γ^\uparrow que pertenecen a otras redes objeto o agente dentro de $\bullet t_j$. Estas transiciones se disparan simultáneamente y las redes objeto o agente declaradas por $\pi((p, t_j), lab)_{<b_{tj}>}$ son desplazadas .

Disparo de Transiciones

El disparo de transiciones en todos los niveles de red modifica el marcado quitando $\pi((p, t_j), lab) < b_{ij} >$ en todos los lugares de entrada y agregando $\pi((t_j, p), lab) < b_{ij} >$ a todos los lugares de salida.

En la figura 3-1 , *ON1* y *ON2* están sincronizadas a través de las transiciones etiquetadas con el símbolo $\alpha^=$; ambas transiciones deberán estar habilitadas para dispararse.. *AG2* y *ON3* están sincronizadas a través de las transiciones etiquetadas con los símbolos γ^\downarrow y γ^\uparrow respectivamente; el disparo de la transición de *AG2* remueve *ON3* del lugar de entrada y cambia su marcado; al mismo tiempo β establece la interacción entre *ON4* (β^{\uparrow}), *AG2* (β^{\uparrow}) y *EN* (β^\downarrow).

3.2 Caso de estudio

Nuestro caso de estudio es una componente de un sistema de tráfico urbano ilustrada en la figura 3-5; este componente está formado por los siguiente elementos:

- Una interseccion : segmento que es punto de union de 4 segmentos de calle. En este punto suponemos la existencia del control de flujo de tráfico de cada calle (los semáforos).
- Calles: en este caso de estudio son 4 los segmentos de calle, cada calle es unidireccional. La calle Hidalgo_W y Alcalde_N, son fuente de vehículos. Y las calles Hidalgo_E y Alcalde_S son los nodos pozo o donde salen los vehículos del sistema.
- Semáforos: tomamos el control efectuado por estos, mas no como entidades físicas del sistema.
- Vehículos: Son las entidades principales del sistema. Tienen comportamientos de toma de decisión individual. Cada uno cuenta con un plan de viaje, es decir , cuando llegue a la intersección, de acuerdo a su plan seleccionará la calle siguiente a seguir.

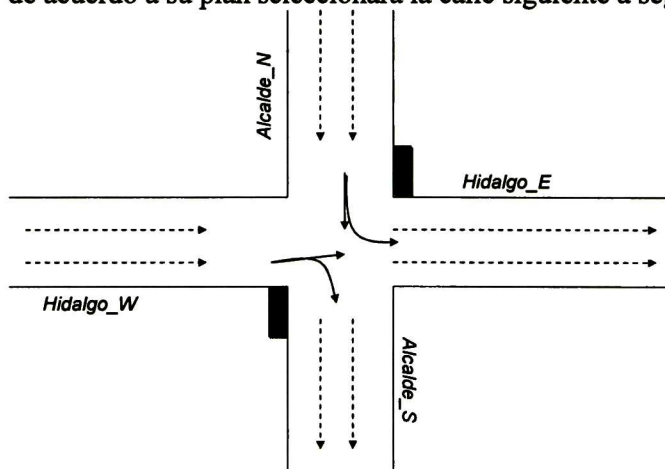


Figura 3-5 Caso de estudio.

3.2.1 Estrategia de Modelado

Para utilizar el formalismo 3-LNS para modelar el sistema previamente descrito, proponemos describir la infraestructura de la red de tráfico mediante la red ambiente. Las redes agente

describen la dinámica de los vehículos y el control de la simulación utilizando la técnica de avance evento siguiente. Las redes objeto (nivel 3) representan el comportamiento del conductor, los planes de viaje y los procesos del sistema. Una red agente incluida en una calle coordina la ocurrencia de eventos, el comportamiento del conductor, cada proceso (red objeto) ejecuta sus acciones y genera nuevos eventos. El agente (móvil) vehículo se desplaza por las calles de acuerdo a su plan de viaje (una red objeto incluida como marca) hasta que alcanza su lugar destino. La figura 3-6 ilustra la jerarquía de las componentes del modelo en tres niveles, la cual surge intuitivamente del sistema de tráfico descrito. Los módulos resultantes estarán relacionados por el etiquetado de las transiciones y/o la inclusión dentro de otras redes como marcas. Esta estrategia, nos permite obtener de manera sistemática y simple, modelos claros de sistemas complejos; compuestos de sub-modelos compactos.

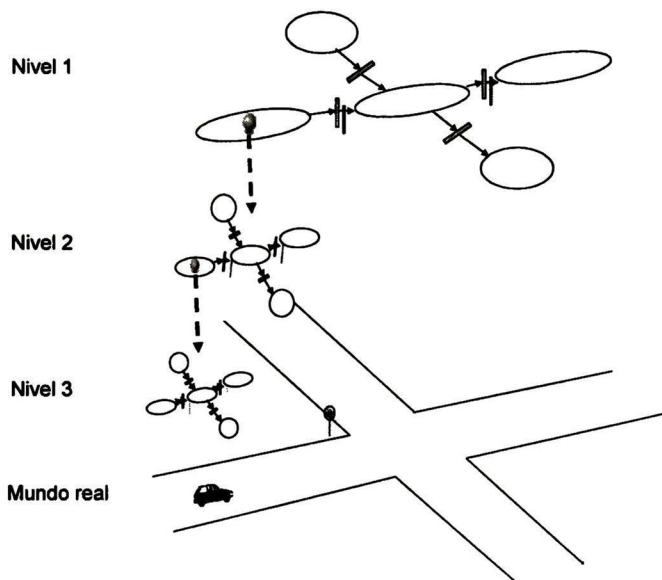


Figura 3-6 Jerarquía del modelo 3-LNS de la red de tráfico.

3.2.2 El modelo de la red de tráfico

Una red de nivel 1 o red ambiente describe la red de tráfico urbano; en este nivel se representan las componentes físicas de la red de tráfico tales como las calles, las intersecciones, los vehículos y los semáforos. También están incluidas en la red ambiente otras entidades no físicas que intervienen en la simulación tales como los agentes de gestión de la calle que regulan la población de vehículos. Cada lugar representa un segmento de calle o un espacio de intersección; cada uno de estos lugares pueden contener agentes vehículos y agentes de gestión de la calle, que son una representación abstracta de la interacción de todos los carros y la aplicación conceptual de la técnica de avance de tiempo de evento siguiente. La red ambiente, denominada *typeUrban_{1,1}*, es presentada en la figura 3-7 y la definición formal se incluye en la tabla 3-1.

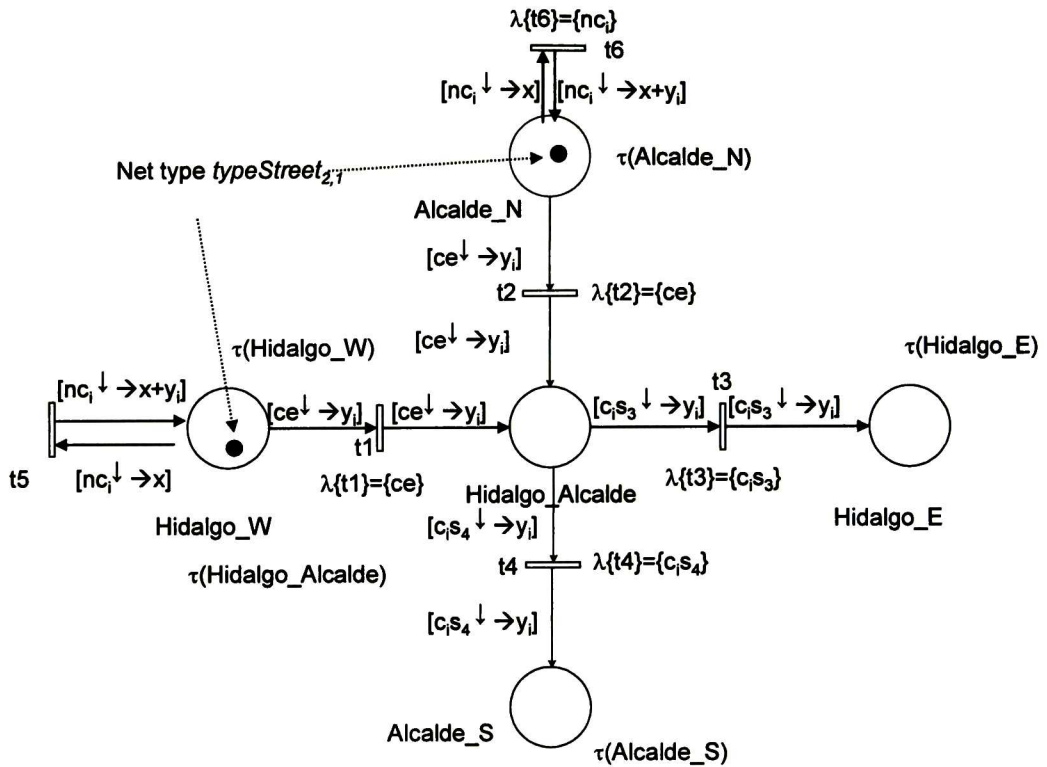


Figura 3-7 Red ambiente de nivel 1, tipo $typeUrban_{1,1}$.

Conjuntos	Etiquetado	Asignación de tipos
$TOKEN_{1,1} = \{typeStreet_{2,1}, typeCar_{2,1}\}$ $LABEL_{1,1} = \{nc_i, ce, c_i s_3, c_i s_4\}$. $VAR_{1,1} = \{x, y\}$.	$\lambda\{t1\} = \lambda\{t2\} = \{ce\}$ $\lambda\{t3\} = \{c_i s_3\}$ $\lambda\{t4\} = \{c_i s_4\}$ $\lambda\{t5\} = \lambda\{t6\} = \{nc_i\}$	$\tau(Hidalgo_W) =$ $\tau(Alcalde_N) =$ $\tau(Hidalgo_E) =$ $\tau(Hidalgo_Alcalde) =$ $\tau(Alcalde_S) = TOKEN_{1,1}$
Pesos		
$\pi((Hidalgo_W, t1), ce^\downarrow) = \pi((Alcalde_N, t2), ce^\downarrow) =$ $\pi((Alcalde_S, t4), c_i s_4^\downarrow) = \pi((Alcalde_S, t3), c_i s_3^\downarrow) =$ $\pi((t1, Alcalde_S), ce^\downarrow) = \pi((t2, Alcalde_S), ce^\downarrow) =$ $\pi((t3, Hidalgo_E), c_i s_3^\downarrow) = \pi((t4, Hidalgo_Alcalde), c_i s_4^\downarrow) = y_i$	$\pi((t5, Hidalgo_W), nc_i^\downarrow) =$ $\pi((t6, Alcalde_N), nc_i^\downarrow) = x + y_i$ $\pi((Hidalgo_W, t5), nc_i^\downarrow) =$ $\pi((Alcalde_N, t6), nc_i^\downarrow) = x$	

Tabla 3-1 Definición formal de la red tipo $typeUrban_{1,1}$

Como puede observarse en la figura 3-7 el modelo es muy intuitivo; los lugares denominados Alcalde_N, Alcalde_S, Hidalgo_E, e Hidalgo_W representan los segmentos de calle del cruce, mientras que el lugar Hidalgo_Alcalde representa el espacio correspondiente a la intersección. Las transiciones t_1, t_2, t_3, t_4 y sus respectivos arcos que conectan los lugares representan el flujo permitido de vehículos. Las transiciones t_5, t_6 son usadas para ingresar nuevos vehículos a los lugares Hidalgo_W y Alcalde_N respectivamente.

En el marcado inicial se tiene una marca del tipo $typeStreet_{2,1}$ en cada lugar. Con este

marcado las transiciones t_5 y t_6 están habilitadas. El disparo de éstas, sincronizado con una transición de la red $typeStreet_{2,1}$, hace aparecer marcas del tipo $typecar_{2,2}$ en los lugares Hidalgo_W y Alcalde_N respectivamente. Posteriormente, para representar el avance de los vehículos, el disparo de t_1 o t_2 transfiere las marcas del tipo $typecar_{2,2}$ al lugar que representa la intersección; ahí éstas pueden avanzar por t_3 o t_4 de acuerdo a su plan de viaje (descrito por $typePlan_{3,8}$).

3.2.3 Las redes agente

En el nivel 2 consideramos en este ejemplo dos tipos de agentes: uno estacionario, denominado $typeStreet_{2,1}$ y otro móvil denominado $typeCar_{2,2}$. El agente móvil representa el comportamiento general de un vehículo (y su conductor) y el agente estacionario se encarga de la gestión de la calle interactuando con el agente móvil y con la red ambiente. La interacción se efectúa sincronizando algunas de las transiciones del modelo $typeStreet_{2,1}$ con las del agente móvil $typeCar_{2,2}$ y las de la red ambiente $typeUrban_{1,1}$.

a) Agente de gestión de la calle

La figura 3-8 muestra el modelo correspondiente al agente de calle denominado $typeStreet_{2,1}$. La definición formal está dada en la tabla 3-2. Esta red describe la administración de los acontecimientos del simulador, considerado como un DES. Su ejecución proporciona la lógica necesaria para permitir la interacción entre vehículos.

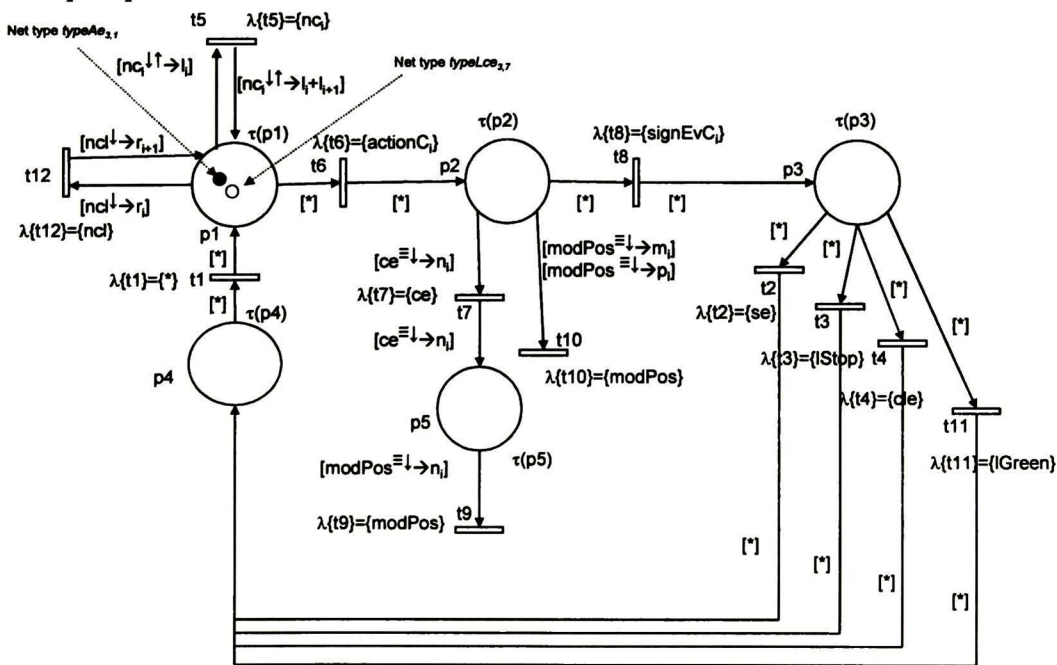


Figura 3-8 Red agente calle de nivel 2, tipo $typeStreet_{2,1}$.

El marcado inicial es un proceso que describe el arribo de un vehículo ($typeAe_i$) y un evento

de cambio de fase en el semáforo (*typeLce_i*); el proceso de arribo generará un vehículo nuevo (*typeCar_{2,2}*) al sincronizarse con la red ambiente. Una vez generado el vehículo, éste se sincronizará con la red agente calle (*typeStreet_{2,1}*) a través de la transición t_6 y de acuerdo con el tipo de proceso, dispara la búsqueda para el siguiente proceso eminente para el vehículo o solo modifica la posición del vehículo. El instante en el que la red ambiente (*typeUrban_{1,1}*) dispare alguna transición de movilidad (t_1, t_2, t_3, t_4), el agente vehículo (*typeCar_{2,2}*) se mueve a otra calle o intersección.

Conjuntos	
TOKEN _{2,1} = { <i>typeAe_{3,1}, typeCle_{3,2}, typeSe_{3,3}, typeSse_{3,4}, typeClee_{3,5}, typeCe_{3,6}, typeLce_{3,7}</i> }. LABEL _{2,1} = { <i>ncl, nc_i, actionC_i, signEvC_i, ce, modPos, se, lStop, cle, lGreen, nsec_i, ncec_i, nclec_i</i> }. VAR _{2,1} = { <i>l: typeAe_{3,1}, m: typeSe_{3,2}, n: typeCe_{3,3}, o: typeSse_{3,4}, p: typeCle_{3,5}, q: typeClee_{3,6}, r: typeLce_{3,7}</i> }	
Asignación de tipos	
$\tau(p1) = \tau(p2) = \{typeAe_{3,1}, typeSe_{3,2}, typeCe_{3,3}, typeSse_{3,4}, typeCle_{3,5}, typeClee_{3,6}, typeLce_{3,7}\}$ $\tau(p3) = \tau(p4) = \{typeAe_{3,1}, typeSse_{3,4}, typeClee_{3,6}\}$ $\tau(p5) = \{typeCe_{3,3}\}$	
Etiquetado	Pesos
$\lambda\{t1\} = \{nsec_i, ncec_i, nclec_i\}$ $\lambda\{t2\} = \{se\}$ $\lambda\{t3\} = \{lStop\}$ $\lambda\{t4\} = \{cle\}$ $\lambda\{t5\} = \{nc_i\}$ $\lambda\{t12\} = \{ncl\}$ $\lambda\{t6\} = \{actionC_i\}$ $\lambda\{t7\} = \{ce\}$ $\lambda\{t8\} = \{signEvC_i\}$ $\lambda\{t9\} = \lambda\{t10\} = \{modPos\}$ $\lambda\{t11\} = \{lGreen\}$	$\pi((t_8, p_3), signEvC_i^{\equiv 1}) = \pi((p_2, t_8), signEvC_i^{\equiv 1}) = \pi((p_3, t_2), se^{\equiv}) = \pi((p_3, t_3), lStop^{\equiv}) = \pi((p_3, t_4), cle^{\equiv}) = \pi((p_3, t_{11}), lGreen^{\equiv}) = \pi((p_3, t_2), se^{\equiv}) = \pi((t_2, p_4), se^{\equiv}) = \pi((t_3, p_4), lStop^{\equiv}) = \pi((t_{11}, p_4), lGreen^{\equiv}) = \pi((t_4, p_4), cle^{\equiv}) = \pi((p_4, t_1), nsec_i^{\equiv}) = \pi((p_4, t_1), ncec_i^{\equiv}) = \pi((p_4, t_1), nclec_i^{\equiv}) = l_i, o_i \text{ ó } p_i$ $\pi((p_1, t_5), nc_i^{\uparrow 1}) = l_i$ $\pi((p_1, t_{12}), ncl^{\uparrow 1}) = r_i$ $\pi((p_2, t_{10}), modPos^{\equiv 1}) = m_i \text{ ó } p_i$ $\pi((p_1, t_6), actionC_i^{\equiv 1}) = \pi((t_6, p_2), actionC_i^{\equiv 1}) = l_i, m_i, n_i, o_i, p_i \text{ ó } q_i$ $\pi((t_5, p_1), nc_i^{\uparrow 1}) = r_{i+1}$ $\pi((p_5, t_9), modPos^{\equiv 1}) = \pi((t_7, p_5), ce_i^{\equiv 1}) = \pi((p_2, t_7), ce_i^{\equiv 1}) = n_i$ $\pi((t_1, p_1), nsec_i^{\equiv}) = m_i + o_i$ $\pi((t_1, p_1), ncec_i^{\equiv}) = n_i$ $\pi((t_1, p_1), nclec_i^{\equiv}) = p_i + q_i$

Tabla 3-2 Definición formal de la red tipo *typeStreet_{2,1}*

b) El agente vehículo

A diferencia del agente de gestión de a calle, el agente vehículo se desplaza dentro de la red ambiente representando el movimiento del agente físico en las calles. El modelo de este agente es una red de nivel 2, llamada *typeCar_{2,2}*, la cual representa el comportamiento del conductor y la manera en la que este sigue el plan de viaje. La red es mostrada en la figura 3-9 y la definición formal de este modelo es detallada en la tabla 3-3. El funcionamiento de esta red es descrito a continuación.

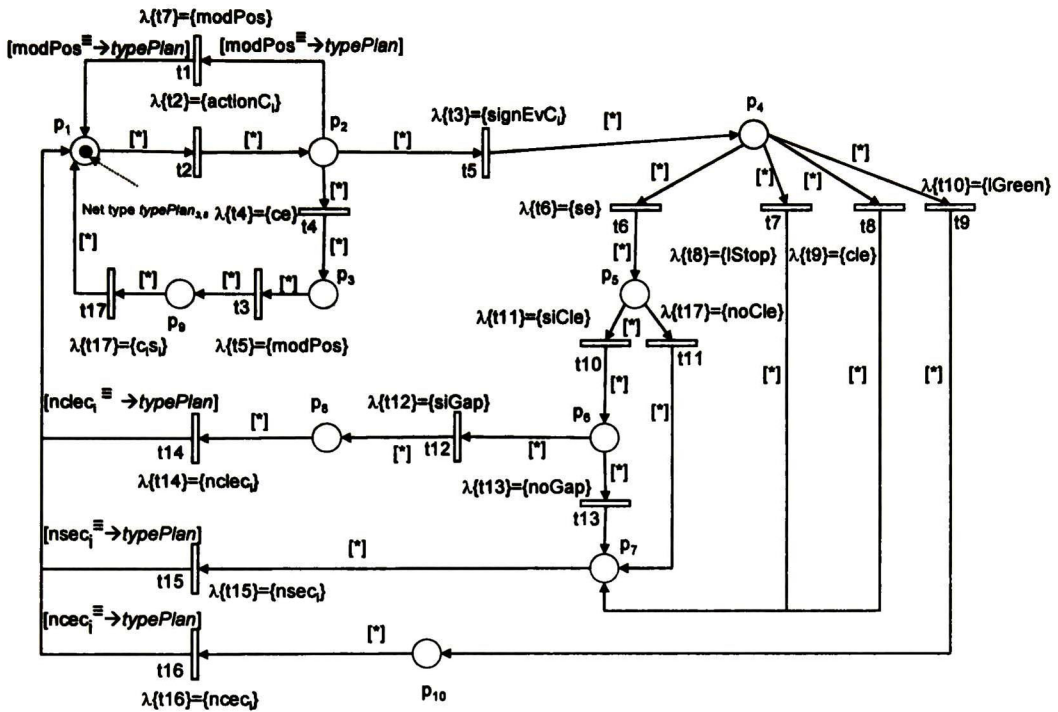


Figura 3-9 Red agente vehiculo de nivel 2, tipo $typeCar_{2,2}$

En el marcado inicial el lugar p_1 contiene una marca tipo red que describe el plan de viaje ($typePlan_{3,8}$). La transición t_2 está sincronizada con la transición t_6 de la red agente calle ($typeStreet_{2,1}$). Cuando alguna de las marcas tipo red (procesos) de la red agente calle cambia su estado a disponible (éstas son redes temporizadas) se habilita la transición t_6 de la red agente calle y si éste proceso se relaciona con éste vehículo (a través de la etiqueta $actionC_i$, donde C_i es el identificador de éste vehículo), t_2 se dispara. Si el proceso es de tipo “cruce de calle” ($typeCe$), la transición t_4 se dispara, se modifica la posición del vehículo y se sincroniza la transición t_{17} con la red que describe el plan de viaje; avanzando el plan y sincronizando ésta transición con las transiciones de movilidad de la red ambiente para hacer el cambio correspondiente de calle de acuerdo al plan de viaje. Si es un proceso de “paro” ($typeSe_i$) o de “cambio de carril” ($typeCle_i$) se modifica la posición del vehículo, disparando la transición t_2 . Si es un proceso de tipo “arribo” ($typeAe_i$), “iniciar-después-de-un-evento-de-alto” ($typeSse_i$) o “iniciar-después-de-un-cambio-de-carril” ($typeClee_i$), se dispara la transición t_5 , sincronizándose con el agente calle, el cual busca el siguiente proceso significativo para éste vehículo; que deberá tomar en cuenta para continuar con el avance. Una vez que el agente calle conoce el siguiente proceso significativo (marcando alguno de los lugares de entrada de las transiciones t_6 a t_9 del agente calle), el vehículo deberá tomar la decisión en base a este resultado; cambiar de carril (si la transición t_6 esta habilitada) o parar (si el semáforo se encontrará en rojo, transición t_7 habilitada), cambiar de carril, si algún vehículo va a cambiar de carril y obstruirá el paso de éste vehículo (transición t_8 , habilitada), o generar un evento de cruce (si será encontrado el semáforo con luz verde, transición t_9). Ésta decisión es de tipo estocástica ya que las transiciones t_6 a t_9 de la red $typeCar_{2,2}$ son éste tipo, simulando la toma de decisión del conductor. Entonces, se genera un nuevo evento en sincronización con el agente calle ($typeStreet_{2,1}$).

Conjuntos	
TOKEN _{2,2} = {typePlan _{3,1} }. LABEL _{2,2} = {modPos, actionC _i , ce, signEvC _i , se, lStop, cle, lGreen, gap, siGap, noGap, nclec _i , nsec _i , ncec _i }. VAR _{2,2} = {ϕ}	
Asignación de tipos	
$\tau(p_1) = \tau(p_2) = \tau(p_3) = \tau(p_4) = \tau(p_5) = \tau(p_6) = \tau(p_7) = \tau(p_8) = \{typePlan_{3,1}\}.$	
Etiquetado	Pesos
$\lambda\{t1\} = \lambda\{t3\} = \{\text{modPos}\}$ $\lambda\{t8\} = \{\text{cle}\}$ $\lambda\{t2\} = \{\text{actionC}_i\}$ $\lambda\{t9\} = \{\text{lGreen}\}$ $\lambda\{t4\} = \{\text{ce}\}$ $\lambda\{t10\} = \{\text{gap}\}$ $\lambda\{t5\} = \{\text{signEvC}_i\}$ $\lambda\{t11\} = \{\emptyset\}$ $\lambda\{t6\} = \{\text{se}\}$ $\lambda\{t12\} = \{\text{siGap}\}$ $\lambda\{t7\} = \{\text{lStop}\}$ $\lambda\{t13\} = \{\emptyset\}$ $\lambda\{t14\} = \{\text{nclec}_i\}$ $\lambda\{t15\} = \{\text{nsec}_i\}$ $\lambda\{t16\} = \{\text{ncec}_i\}$	$\pi((p_1, t_2), \text{actionC}_i^{\bar{=}}) = \pi((t_2, p_2), \text{actionC}_i^{\bar{=}}) = \pi((p_2, t_4), \text{ce}_i^{\downarrow}) = \pi((t_4, p_3), \text{ce}_i^{\downarrow}) = \pi((t_3, p_9), \text{modPos}^{\bar{=}}) = \pi((p_2, t_7), \text{modPos}^{\bar{=}}) = \pi((t_7, p_1), \text{modPos}^{\bar{=}}) = \pi((p_3, t_3), \text{modPos}^{\bar{=}}) = \pi((p_9, t_{17}), c_i s_i^{\downarrow}) = \pi((t_{17}, p_1), c_i s_i^{\downarrow}) = \pi((p_2, t_5), \text{signEvC}_i^{\bar{=}}) = \pi((t_5, p_4), \text{signEvC}_i^{\bar{=}}) = \pi((p_4, t_6), \text{se}^{\bar{=}}) = \pi((t_6, p_5), \text{se}^{\bar{=}}) = \pi((p_4, t_7), \text{lStop}^{\bar{=}}) = \pi((t_7, p_7), \text{lStop}^{\bar{=}}) = \pi((p_4, t_9), \text{lGreen}^{\bar{=}}) = \pi((t_9, p_{10}), \text{lGreen}^{\bar{=}}) = \pi((p_4, t_8), \text{cle}^{\bar{=}}) = \pi((t_8, p_7), \text{cle}^{\bar{=}}) = \pi((p_{10}, t_{16}), \text{ncec}_i^{\bar{=}}) = \pi((t_{16}, p_1), \text{ncec}_i^{\bar{=}}) = \pi((p_7, t_{15}), \text{nsec}_i^{\bar{=}}) = \pi((t_{15}, p_1), \text{nsec}_i^{\bar{=}}) = \pi((p_8, t_{14}), \text{nclec}_i^{\bar{=}}) = \pi((t_{14}, p_1), \text{nclec}_i^{\bar{=}}) = \pi((p_5, t_{10}), \text{siCle}^{\bar{=}}) = \pi((t_{10}, p_6), \text{siCle}^{\bar{=}}) = \pi((t_{14}, p_1), \text{nclec}_i^{\bar{=}}) = \pi((p_5, t_1), \text{noCle}^{\bar{=}}) = \pi((t_{11}, p_7), \text{noCle}^{\bar{=}}) = \pi((p_6, t_{13}), \text{noGap}^{\bar{=}}) = \pi((t_{13}, p_7), \text{noGap}^{\bar{=}}) = \pi((p_6, t_{12}), \text{siGap}^{\bar{=}}) = \pi((t_{12}, p_8), \text{siGap}^{\bar{=}}) = typePlan_3.$

Tabla 3-3 Definición formal de la red tipo *typeCar_{2,2}*

3.2.4 Redes de nivel 3: los objetos y los recursos

La manera en la que el vehículo debe avanzar y los procesos usados para realizar la simulación manejada por procesos, son representados por redes objeto. Describiremos brevemente el funcionamiento de estos modelos los cuales complementan la descripción de las redes agente y la red ambiente de nuestro modelo.

a) Proceso de arribo de vehículos

Esta red representa el proceso de arribo de los vehículos a la red ambiente, la calendarización de otro vehículo a la red y la generación del primer proceso para la red agente vehículo *typeCar_{2,2}* que la contiene. El modelo de este proceso es una red de nivel 3, llamada *typeAe_{3,1}*, contenida en una red agente vehículo como marca inicial. La red es mostrada en la figura 3-10 y la definición formal de este modelo es detallada en la tabla 3-4. El funcionamiento de esta red es descrito a continuación.

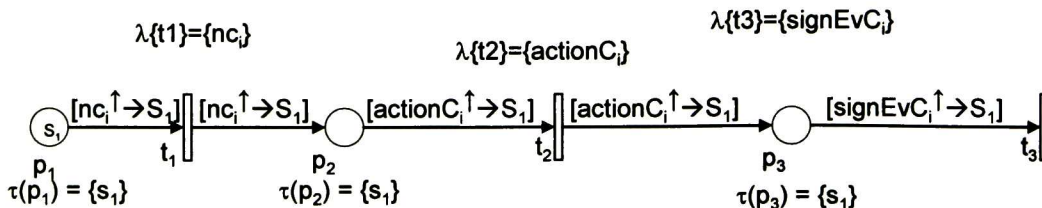


Figura 3-10 Red objeto de nivel 3, tipo *typeAe_{3,1}*

La red tiene el lugar p_1 temporizado, así que el símbolo que está en éste lugar no está disponible hasta transcurrido cierto tiempo. Una vez haya transcurrido el tiempo, t_1 es habilitada y se sincroniza con la transición t_1 de la red agente calle, entonces la red agente calle se sincronizada externamente con la transición t_5 de la red *typeUrban*_{1,1} y se genera una nueva marca (*typeCar*_{2,2}) de tipo i . Ahora se dispara la transición t_2 , que indica a la red agente vehículo, a través de la red agente calle, que busque el primer proceso siguiente significativo para el vehículo en cuestión.

<i>Conjuntos</i>	
$TOKEN_{3,1} = \{s_1\}$. $LABEL_{3,1} = \{nc_i, actionC_i, signEvC_i\}$. $VAR_{3,1} = \{\phi\}$	
<i>Asignación de tipos</i>	
$\tau(p_1) = \tau(p_2) = \tau(p_3) = \{s_1\}$	
Etiquetado	Pesos
$\lambda\{t_1\} = \{nc_i\}$ $\lambda\{t_3\} = \{signEvC_i\}$ $\lambda\{t_2\} = \{actionC_i\}$	$\pi((p_2, t_2), actionC_i^\uparrow) = \pi((t_2, p_3), actionC_i^\uparrow) = \pi((p_1, t_1), nc_i^\uparrow) = \pi((t_1, p_2), nc_i^\uparrow) = \pi((p_3, t_3), signEvC_i^\uparrow) = s_1$

Tabla 3-4 Definición formal de la red tipo *typeAe*_{3,1}

b) Procesos generadores de procesos

Estas redes representan el proceso de generación de nuevos procesos. El modelo de éste proceso es una red de nivel 3, llamada *typeSe*_{3,4} o *typeCle*_{3,5} contenida en una red agente calle. La red es mostrada en la figura 3-11 y la definición formal de este modelo es detallada en la tabla 3-5. El funcionamiento de esta red es descrito a continuación.

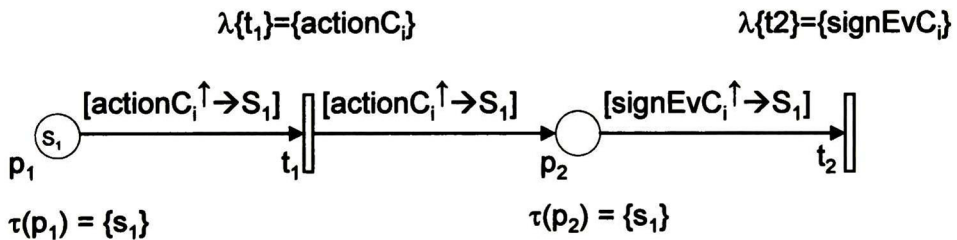


Figura 3-11 Red objeto de nivel 3, tipo *typeSse*_{3,2} y *typeCle*_{3,3}.

Estas redes son generadas al mismo tiempo que las redes *typeSe*_{3,4} y *typeCle*_{3,5}. Estos procesos buscan el siguiente proceso significativo, *signEvC_i*, para el vehículo de tipo i , después de haber hecho un paro total, o un cambio de carril respectivamente, la transición t_2 , indica al agente calle (*typeStreet*_{2,1}) que informe al vehículo tipo i (*typeCar*_{2,2}) el proceso eminente que pudiese afectarle y entonces tomar una decisión (comportamiento del conductor).

Conjuntos	
$TOKEN_{3,2} = TOKEN_{3,2} = \{s_1\}.$ $LABEL_{3,2} = LABEL_{3,2} = \{actionC_i, signEvC_i\}.$ $VAR_{3,2} = VAR_{3,2} = \{\emptyset\}$	
Asignación de tipos	
$\tau(p_1) = \tau(p_2) = \{s_1\}.$	
Etiquetado	Pesos
$\lambda\{t_1\} = \{actionC_i\}$ $\lambda\{t_2\} = \{signEvC_i\}$	$\pi((p_1, t_1), actionC_i \uparrow) = \pi((t_1, p_2), actionC_i \uparrow) = \pi((p_2, t_2), signEvC_i \uparrow) = s_1$

Tabla 3-5 Definición formal de la red tipo *typeSse_{3,2}*

c) Procesos de paro y de cambio de carril

Esta red representa el proceso de paro o de cambio de carril del vehículo y su correspondiente modificación de posición. El modelo de este proceso es una red de nivel 3, llamada *typeSe_{3,4}* o *typeCle_{3,5}*, contenidas en una red agente calle. La red es mostrada en la figura 3-12 y la definición formal de este modelo es detallada en la tabla 3-6. El funcionamiento de esta red es descrito a continuación.

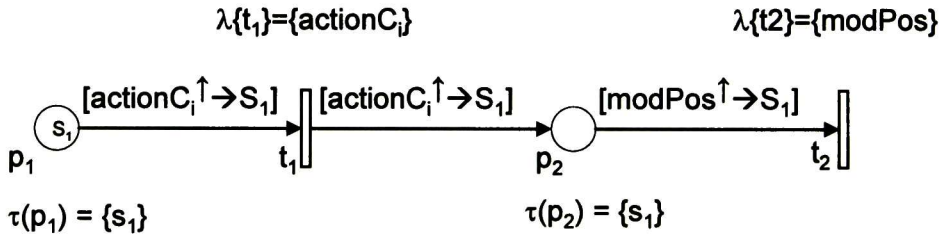


Figura 3-12 Red objeto de nivel 3, tipo *typeSe_{3,4}* y *typeCle_{3,5}*.

Estas redes representan los procesos de hacer un alto (*typeSe_{3,4}*) o un cambio de carril (*typeCle_{3,5}*), sólo modifican la posición del vehículo, cuando estos eventos son generados las redes *typeSse_{3,2}* y *typeClee_{3,3}* también son generadas, porque una vez que el vehículo haya hecho el alto, necesita generar un nuevo evento para decidir cual será su siguiente acción después de que ocurra el evento de paro o cambio de carril.

Conjuntos	
$TOKEN_{3,4} = TOKEN_{3,5} = \{s_1\}.$ $LABEL_{3,4} = LABEL_{3,5} = \{actionC_i, modPos\}.$ $VAR_{3,4} = VAR_{3,5} = \{\emptyset\}$	
Asignación de tipos	
$\tau(p_1) = \tau(p_2) = \{s_1\}.$	
Etiquetado	Pesos
$\lambda\{t_1\} = \{actionC_i\}$ $\lambda\{t_2\} = \{modPos\}$	$\pi((p_1, t_1), actionC_i \uparrow) = \pi((t_1, p_2), actionC_i \uparrow) = \pi((p_2, t_2), modPos \uparrow) = s_1$

Tabla 3-6 Definición Formal de la red tipo *typeSe_{3,4}*

d) Proceso de arribo al fin de la calle

Esta red representa el proceso de arribo al final de la calle. El modelo de este proceso es una red de nivel 3, llamada *typeCe_{3,6}*, contenida en una red agente calle. La red es mostrada en la figura 3-13 y la definición formal de este modelo es detallada en la tabla 3-7. El funcionamiento de esta red es descrito a continuación.

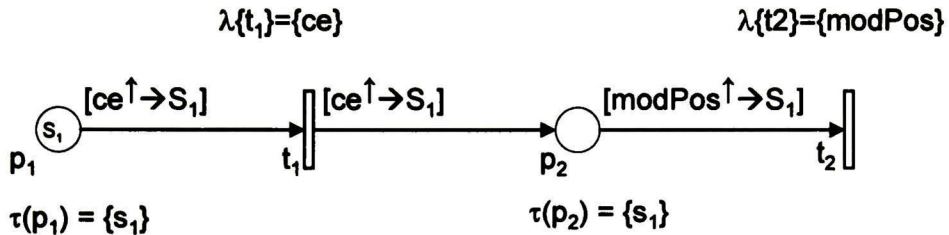


Figura 3-13 Red objeto de nivel 3, tipo *typeCe_{3,6}*.

Esta red describe el alcance de fin de la calle y el respectivo cruce de la intersección de la red. Modifica su posición al valor final de la calle, porque ha alcanzado el final de la calle y esta listo para seleccionar la siguiente calle de acuerdo a su plan de viaje.

Conjuntos	
TOKEN _{3,6} = {s ₁ }. LABEL _{3,6} = {ce, c _i s ₄ }. VAR _{3,6} = {ϕ}	
Asignación de tipos	
τ(p ₁) = τ(p ₂) = τ(p ₃) = {s ₁ }.	
Etiquetado	Pesos
λ{t ₁ } = {ce} λ{t ₂ } = {modPos}	π((p ₁ , t ₁), ce [↑]) = π((t ₁ , p ₂), ce [↑]) = π((p ₂ , t ₂), modPos [↑]) = s ₁

Tabla 3-7 Definición formal de la red tipo *typeCe_{3,6}*

e) Plan de viaje

La ruta que el vehículo debe seguir para llegar a su destino, está descrita por la red *typePlan_{3,8}* la cual es mostrada en la figura 3-14. Ésta red está contenida en una red agente vehículo como marca inicial. La red es mostrada en la figura 3-14 y la definición formal de este modelo es detallada en la tabla 3-8. El funcionamiento de esta red es descrito a continuación.

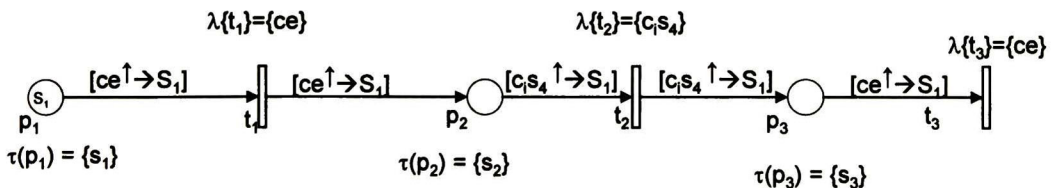


Figura 3-14 Red objeto de nivel 3, tipo *typePlan_{3,8}*.

La ruta que el vehículo debe seguir para llegar a su destino, esta descrita por al red *typePlan_{3,8}* la red tiene una transición etiquetada con *ce*, esta es disparada una vez que el vehículo cruce la calle y arribe a la intersección, y la transición *c_is₄* sea disparada cuando es

seleccionada la calle a la que el vehículo debe ir.

Conjuntos	
$TOKEN_{3,8} = \{s_j\}$. $LABEL_{3,8} = \{ce, c, s_4\}$. $VAR_{3,8} = \{\emptyset\}$	
Asignación de tipos	
$\tau(p_1) = \tau(p_2) = \tau(p_3) = \{s_j\}$.	
Etiquetado	Pesos
$\lambda\{t_1\} = \lambda\{t_3\} = \{ce\}$ $\lambda\{t_2\} = \{c, s_4\}$	$\pi((p_1, t_1), ce^\dagger) = \pi((t_1, p_2), ce^\dagger) = \pi((p_2, t_2), c, s_4^\dagger) = \pi((t_2, p_3), c, s_4^\dagger) =$ $\pi((p_3, t_3), ce^\dagger) = s_j$

Tabla 3-8 Definición formal de la red tipo *typePlan_{3,8}*

La construcción de modelos de sistemas de tráfico urbano (STU) usando 3-LNS, permite representar de manera jerárquica los componentes que le conforman, permitiendo un modelo claro y orientado a multi-agentes.

Se presenta un modelo de STU que describe en el nivel 1 el ambiente del STU, en este caso la infraestructura de la red de tráfico, en el nivel 2, los agentes del STU, tales como, vehículos, semáforos, etc. y en el nivel 3 los componentes de los agentes, como planes de trayecto, habilidades y actitudes de los conductores, etc.

Capítulo 4

IMPLEMENTACIÓN DEL SIMULADOR DE TRÁFICO URBANO CON AGENTES MÓVILES.

Resumen. El presente capítulo describe la metodología con la que se implementa el simulador de tráfico urbano, usando la técnica de evento siguiente para el control de avance de tiempo. Cada etapa de la metodología es explicada en base a la experiencia de implementación de un simulador para el caso de estudio de una calle y una intersección. Se muestran los componentes principales del sistema y su objetivo.

4.1 Metodología de implementación multi-agente

En la literatura existen varias metodologías que describen el proceso de diseño de un simulador basado en multi-agente (MABS: multi-agent based simulation) (Fishwick, 1996), (Gilbert, 1999). Estas metodologías no cuentan con alguna etapa en la que se traduzcan los agentes reales en agentes computacionales y por lo tanto, los simuladores no resultan ser realmente MABS.

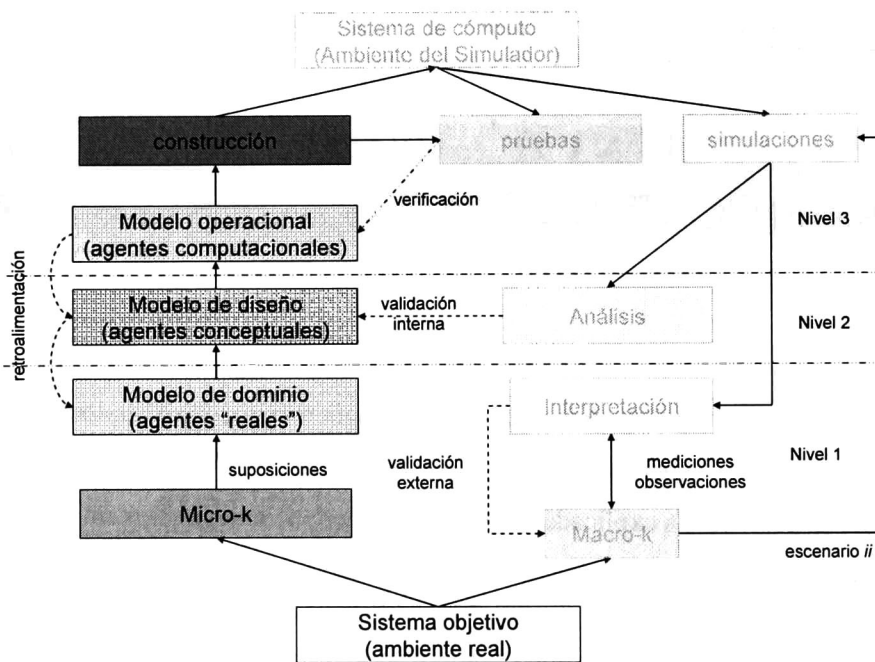


Figura 4-1 Metodología para desarrollo de MABS.

(Drogoul, 2002) analiza éste problema y propone una nueva metodología (figura 4-1), en la que se toma en cuenta la conversión: agente real \rightarrow agente computacional. Se describen 3 papeles principales en el desarrollo de MABS: El primero es ejecutado por los llamados temáticos, si dominan y conocen el funcionamiento, fenómenos y la teoría relacionada con el comportamiento del sistema. Por otra parte, puesto que la simulación se pondrá a funcionar en una computadora, necesitamos a expertos en informática que puedan construir los programas, a los cuales llamamos informáticos. Todavía y muy importante, es tomar en cuenta la brecha que existe entre estas dos comunidades (temáticos y los informáticos). Entonces una tercera comunidad que es involucrada

en el proceso de construir la simulación (diseño, construcción, ejecución y análisis) son los modeladores.

La metodología propuesta se explicará más adelante junto con la implementación (sección 4.2) de un simulador propio. En la figura 4-1, se pueden apreciar las capas 1 a la 3, éstas representan las etapas del proceso, en las que se involucra el temático, el modelador y el informático, respectivamente.

4.2 Implementación del modelo multi-agente

4.2.1 Sistema Objetivo

El sistema del que se quiere desarrollar un simulador, es de tráfico urbano, con tráfico urbano nos referimos a las calles e intersecciones de un área urbana. De éste sistema, la mayoría de las personas tienen un ligero entendimiento de cómo se comporta, cuáles son los principales fenómenos que presenta, qué entidades lo componen, etc. Toda esta información se puede catalogar en dos tipos: macro-k y micro-k.

4.2.2 Conocimiento Macro-k

El conocimiento Macro-k (macro-conocimiento) es un conjunto de conocimiento “global” acerca de una situación, un fenómeno o una estructura (llamada el sistema objetivo) del cual podemos obtener mediciones, observaciones, etc..

El flujo, por ejemplo, es una variable tipo macro-k ya que permite medir la eficiencia de un sistema de transporte como un todo, tomando el número de vehículos que pasan por algún punto en específico en un periodo de tiempo. Para lograr hacer ésta medición en el simulador, se divide el flujo por el número de carriles (Nagel, 2004). Si por ejemplo, durante el tiempo T se mide N_T vehículos, el flujo es entonces:

$$q_T = \frac{N_T}{TN_{\text{carriles}}} \quad (4-1)$$

Una unidad típica del flujo son “Número de vehículos por hora y carril” En nuestro simulador se implementa esta medición. Existe una variable que almacena el número de vehículos que pasan por la intersección cada vez que transcurre un T predefinido. Una vez

transcurrido T, se divide la variable entre el número de carriles multiplicada por T y se envía el resultado a un archivo de texto y la variable en cuestión se reinicia. La densidad ρ_L es el número de vehículos en cierto tramo de la calle, dividida por la longitud de éste tramo L. Con el fin de permitir una comparación es muy útil dividir por el número de carriles:

$$\rho_L = \frac{N_{veh}}{LN_{carriles}} \quad (4-2)$$

La densidad resultante es dada en “ (número de) vehículos por km y carril” Para medir la densidad en nuestro simulador, se calcula el número de vehículos que están en la calle en el mismo instante en el que se calcula el flujo, para poder representar de manera adecuada la relación densidad-flujo.

4.2.3 Conocimiento Micro-k

Es un micro-conocimiento acerca de los comportamientos e interacciones de los agentes, sin el cual, el fenómeno de congestión, por ejemplo, no existiría. Se utilizan los modelos presentados en el capítulo 1, de seguimiento-vehículo y cambio-de-carril, de éste conocimiento se generan suposiciones, como en este caso, que existen dos velocidades que puede tener un vehículo, esto por simplicidad de diseño. Para cálculo de avance del vehículo hacemos uso de las funciones analíticas, mostradas en la figura 4-2. Una característica importante de estas funciones es la habilidad para lograr soluciones exactas. Su principal ventaja reside en su habilidad para representar fielmente el mundo real, a diferencia de otros métodos, como el uso del cálculo de elementos finitos (enteros o discretos), que por naturaleza, cuando el modelo es ejecutado existe una pérdida de precisión. Una fuente de error es debida a la longitud arbitraria del paso de simulación (*Russell, 1996*).

$$v = v_0 + at \quad (4-3)$$

$$x = x_0 + v_0t + \frac{1}{2}at^2 \quad (4-4)$$

$$v^2 = v_0^2 + 2a(x - x_0) \quad (4-5)$$

$$x - x_0 = \left(\frac{v_0 + v}{2} \right)t \quad (4-6)$$

En la figura 4-3, se muestra en ejemplo, con sólo un carril y el estado del semáforo en alto, es generada una cola, el primer vehículo calcula su tiempo de arribo a la intersección (1.5 min.) y genera el evento de alto (SE) a ser ejecutado dentro de 1.5 min. El siguiente carro que llega, por ejemplo 3 segundos después del primero, calcula su tiempo de arribo a la intersección, pero ahora toma en cuenta que hay un vehículo en frente de él, éste vehículo permanecerá ahí, así que tendrá que generar un evento de alto total una posición detrás del primer vehículo, usando la ecuación 4-3 , el vehículo calcula el tiempo exacto en el que llegara hasta el primer vehículo, pero con una distancia de seguridad (2 m) con $\Delta t = \frac{\Delta x}{v}$, donde Δt es el tiempo necesario para avanzar Δx metros con una velocidad de v km/hr.

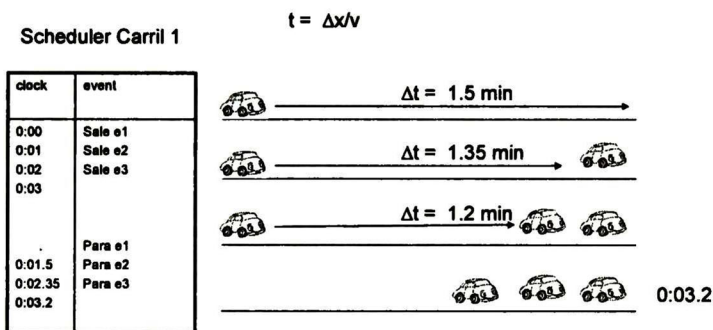


Figura 4-2 Simulación con la técnica de evento siguiente.

Para lograr lo anterior se utiliza simulación continua en espacio, cada vehículo tiene cierta posición descrita por un número de punto-flotante. En contraste con las simulaciones discretas en espacio, que son un tipo especial de autómatas celulares, éstas usan celdas y los vehículos en las calles simuladas “brincan” de una celda a otra. En la figura 4-3 se muestran estas diferencias.

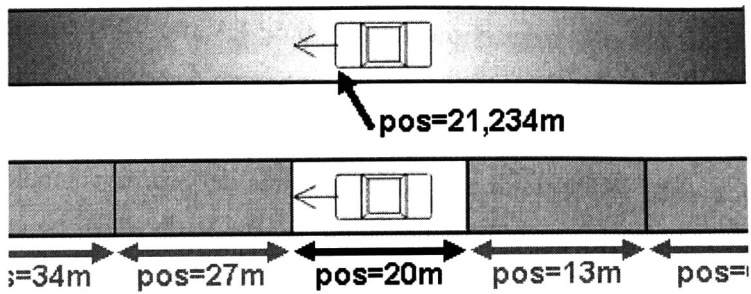


Figura 4-3 Las diferencias entre simulación continua y discreta en espacio.

4.2.4 Modelo de dominio

En esta etapa, el temático pretende entender y explicar las contribuciones de los comportamientos de los agentes al comportamiento global del sistema y cómo es que se relacionan unas con otras. En esta etapa se definen los límites del sistema.

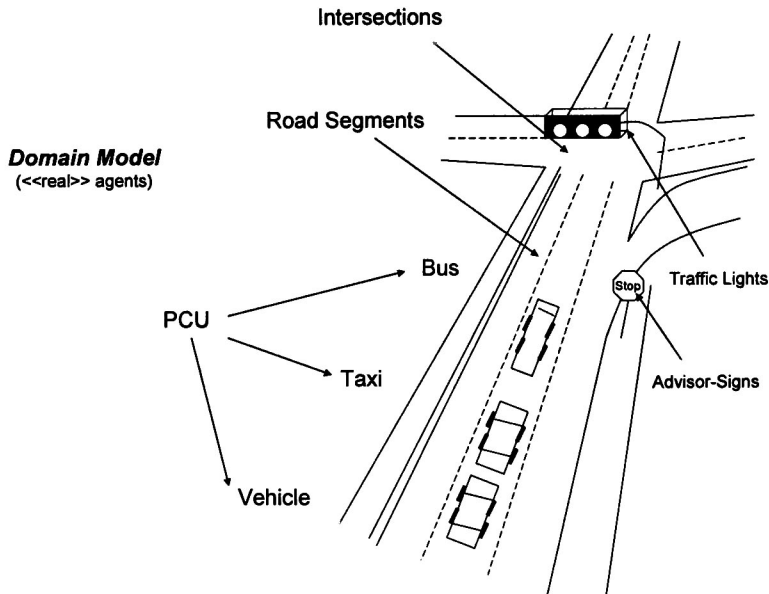


Figura 4-4 Modelo del dominio del sistema.

En la figura 4-4, se puede apreciar el resultado de éste modelo, donde se define quién se comunica con quién y cuáles son las entidades a tomar en cuenta del sistema de tráfico urbano:

- Semáforos
- Calles
- Vehículos
- Intersecciones

4.2.5 Modelo de diseño

En esta etapa el modelador traslada el modelo del dominio en algo más formal, que eventualmente será implementado. Identifica los conceptos principales (para interacción, comunicación, etc.) y selecciona el formalismo que mejor represente el comportamiento multi-agente. En esta etapa se trabaja con agentes conceptuales aún no implementados, pero si descritos a un nivel abstracto mayor que el anterior.

Por lo anterior que se decide utilizar la extensión 3-LNS, una extensión del formalismo RP, de acuerdo al análisis que se presenta en el capítulo 3 (Figura 4-5).

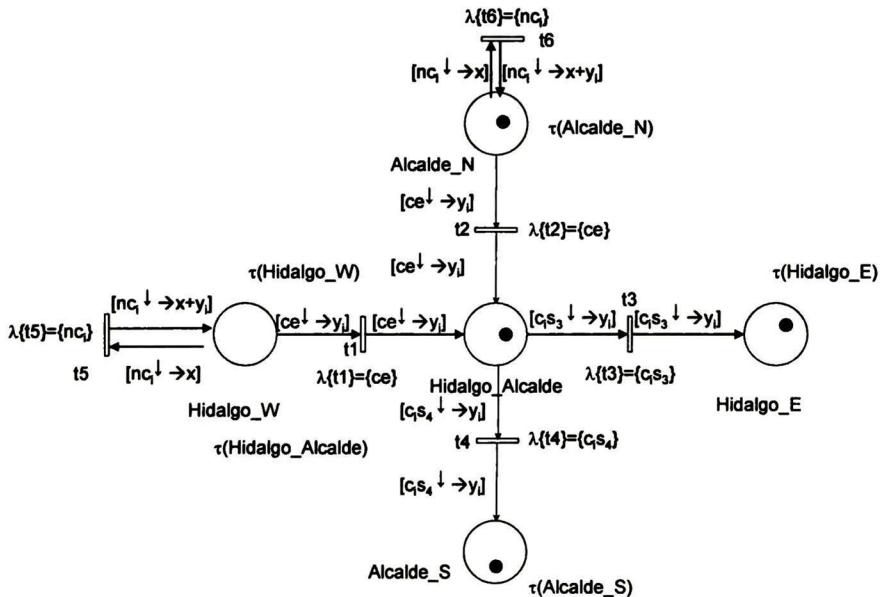


Figura 4-5 Modelo de diseño del sistema.

En el modelo se definen 7 eventos que modifican el estado del sistema en el tiempo. El evento *AE* (arrival event: evento de arribo): la primer actividad ejecutada después de disparado el evento, es generar un vehículo nuevo en el segmento calle. Con éste evento el vehículo obtiene

(del agente calle) la información de posibles obstáculos que estarán en su camino antes de llegar a la intersección. El primer obstáculo posible es un vehículo que puede:

- 1) ir delante de éste e ir más lento,
- 2) ir delante de éste y estar parado (a causa del semáforo o por descompostura) e
- 3) ir un vehículo de otro carril realizando un cambio de carril para colocarse frente a éste.

Para el primer caso, el vehículo puede tomar la decisión de cambiar de carril o simplemente disminuir su velocidad, en el segundo caso, tendrá que pararse después del vehículo o cambiar de carril y en el último caso, hacer un paro total o disminuir su velocidad. El segundo obstáculo es el semáforo, si está en verde cruza la intersección de lo contrario hace un paro total. Éste evento además, genera nuevos eventos tipo *AE* de acuerdo a una distribución de Poisson.

El evento *CLE* (change lane event: cambio de carril): éste evento es generado cuando el vehículo encuentra un obstáculo (vehículo) en su camino. Primero debe verificar que haya espacio suficiente para hacer éste cambio de carril y que ningún otro vehículo llegue a ocupar el mismo espacio en el mismo instante, éste evento genera además un *CLEE* (change lane event end), que ejecutará las mismas actividades que el evento *AE*, pero sin generar nuevos vehículos.

El evento *SE* (stop event: evento de paro): éste evento avanza al vehículo hasta la posición de alto total calculada y genera un nuevo evento llamado *SEE* (start from stop event), que busca los siguientes obstáculos en el instante que el vehículo inicie de nueva cuenta su avance.

El evento *CE* (cross event: evento de cruce): generado cuando el vehículo ya puede cruzar la calle.

El evento *LCE* (light change event: evento de cambio de luz): generado para hacer los cambios en el estado de los semáforos.

4.2.6 Modelo Operacional

El objetivo del informático es escribir un programa para computadora que tome el modelo de diseño como entrada y pueda ejecutarlo en un ambiente de simulación. En esta etapa es donde se define la distribución, el calendarizado del tiempo y las comunicaciones, entre otros aspectos.

Lo primero que se define es el algoritmo que controla la simulación. Éste es el encargado de

tomar el control y calendarizado de los eventos del sistema.

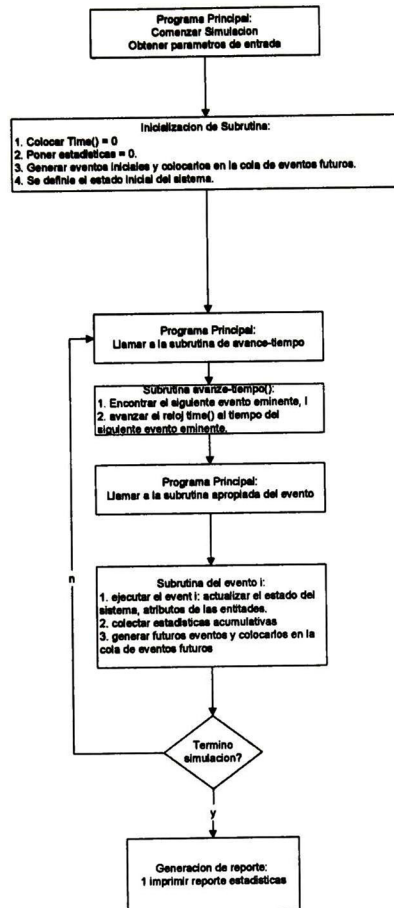


Figura 4-6 Algoritmo de control de la simulación.

El algoritmo de ejecución de los eventos son definidos de igual manera, su descripción es tal y como se hizo en la sección anterior.

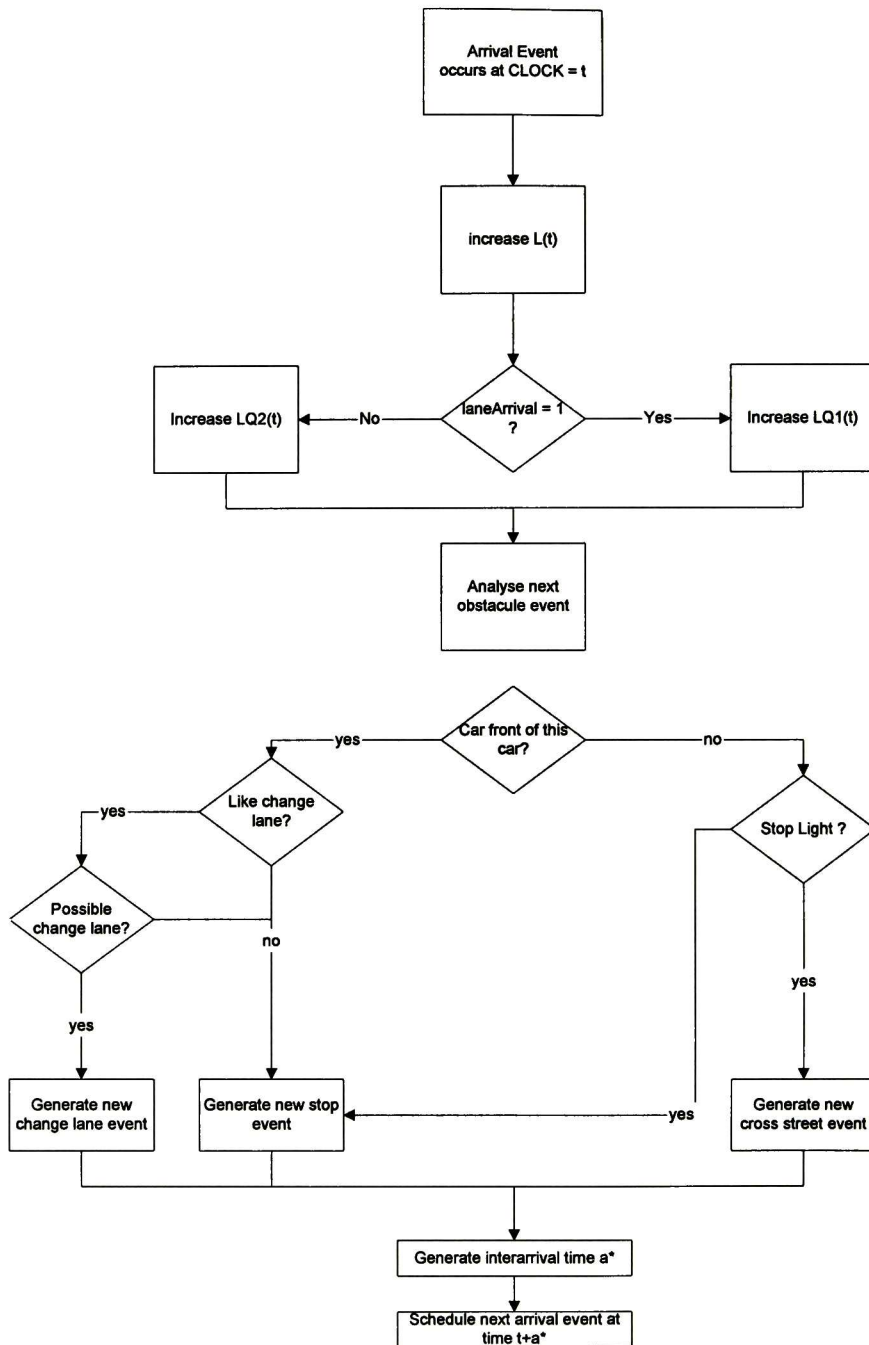


Figura 4-7 Algoritmo de Flujo del evento *AE*.

Este algoritmo es muy parecido a los de los eventos *SEE*, *CLEE* pero sin la generación de un nuevo *AE*. En la figura siguiente se muestra el diagrama de clases del sistema para representar

las relaciones que existen entre los componentes del sistema así como los atributos que lo conforman.

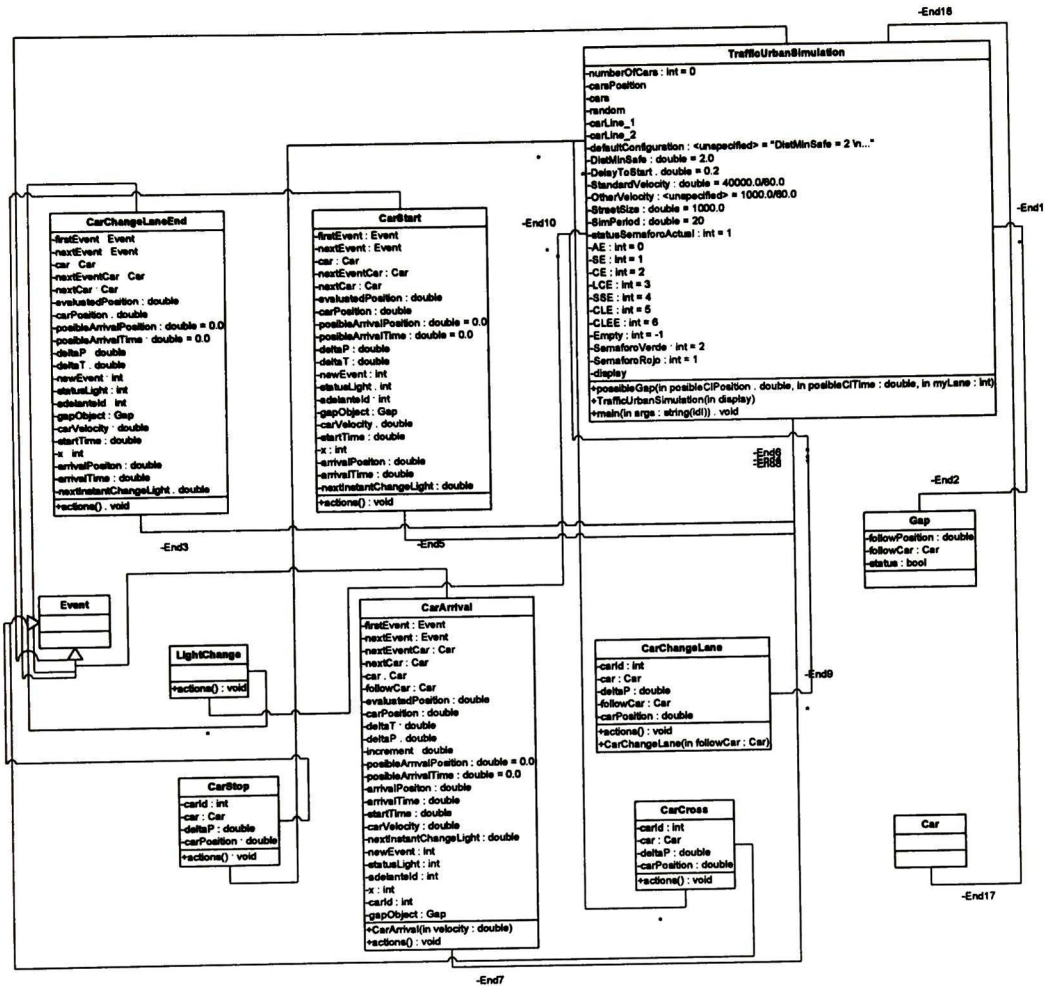


Figura 4-8 Clases del simulador.

El agente calle, contiene una relación de los vehículos que están usando el segmento de calle coordinado por éste. En la figura 4-9 se muestra esa relación.

Agente Calle

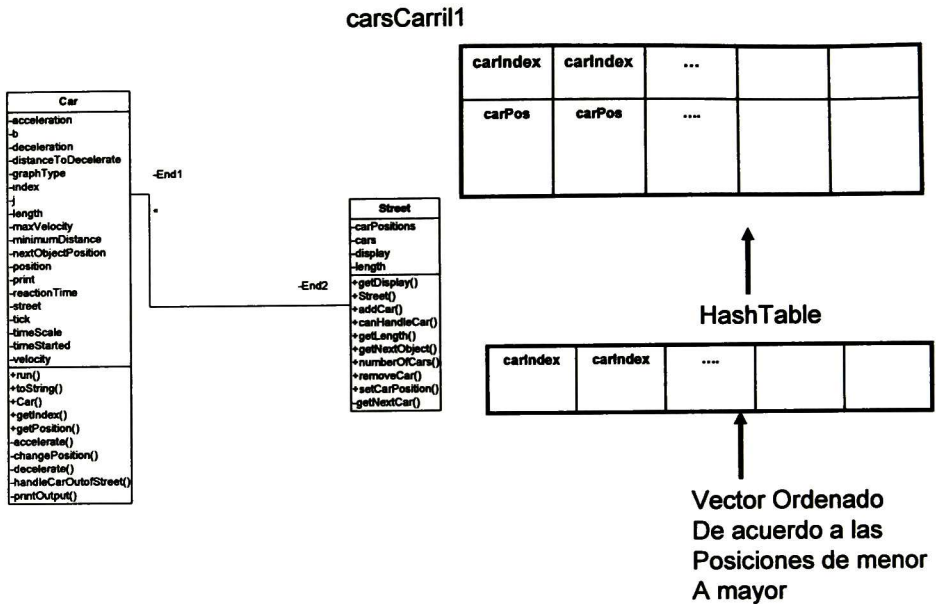


Figura 4-9 Uso de vectores para ordenar los vehículos en la calle.

4.2.7 Construcción

Se usa lenguaje JAVA para la construcción del simulador. Se usa un applet para cargar todas las clases de la simulación. En el siguiente capítulo siguiente se analizan los pasos restantes de la metodología.

La división de los roles en el diseño de un simulador basado en multi-agentes, propuesto por (Drogoul, 2002) nos permite dividir el problema en partes menos complejos y que además tienen conceptos diferentes de lo que es un “agente”, pero se pone énfasis en los puntos donde se hace el cambio de “agente conceptual” a “agente computacional”, con el apoyo del modelo realizado con el formalismo de redes de Petri, nos permite ir de manera muy natural implementado el simulador.

Capítulo 5

ANÁLISIS DE RESULTADOS

Resumen. En el presente capítulo continúa con la descripción de la metodología para el desarrollo de simuladores basados en multi-agente, presentado en el capítulo anterior. Se presentan los resultados de la simulaciones y el análisis respectivo. Se observan los fenómenos de congestión de tráfico.

5.1 Pruebas y simulaciones

Continuando con la descripción de la metodología para la implementación de MABS (Drogoul, 2002), describimos la última etapa que consiste en las pruebas (verificación), simulaciones, análisis (validación interna) e interpretación (figura 5-1).

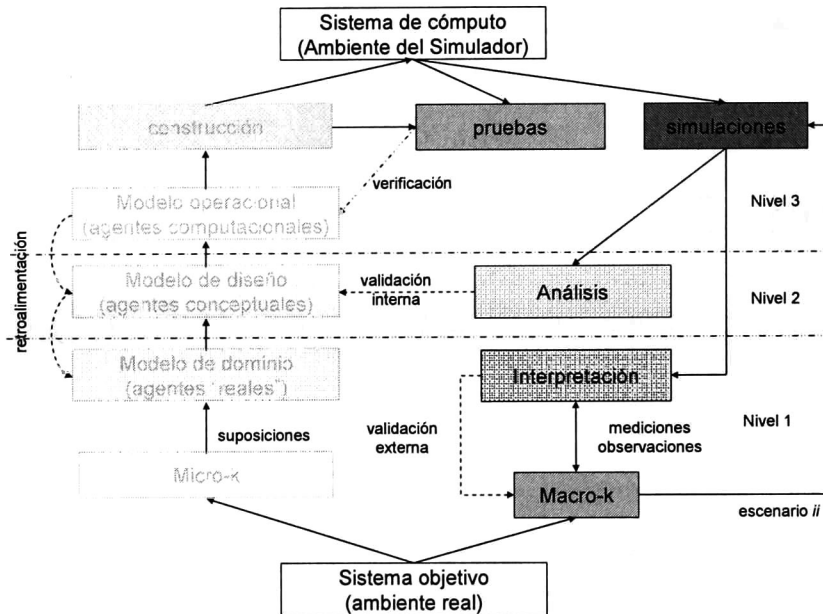


Figura 5-1 Metodología para la implementación de MABS. Poniendo énfasis en la última etapa.

En ésta etapa se realizan las pruebas de la correcta implementación del modelo operacional en la construcción del simulador, eliminar todo tipo de errores lógicos o de incorrecta implementación de los algoritmos. Las simulaciones forman parte de esta etapa y son los resultados obtenidos de las corridas del programa resultado de la implementación del modelo de diseño, se ejecutaron simulaciones de una calle de 1 kilómetro de longitud, con una intersección en su extremo. Modificando los parámetros de velocidad, la distancia mínima posible de seguridad entre vehículos, la distribución aleatoria de arribo de vehículos entre otros. En la figura 5-2 se muestra una de estas corridas.

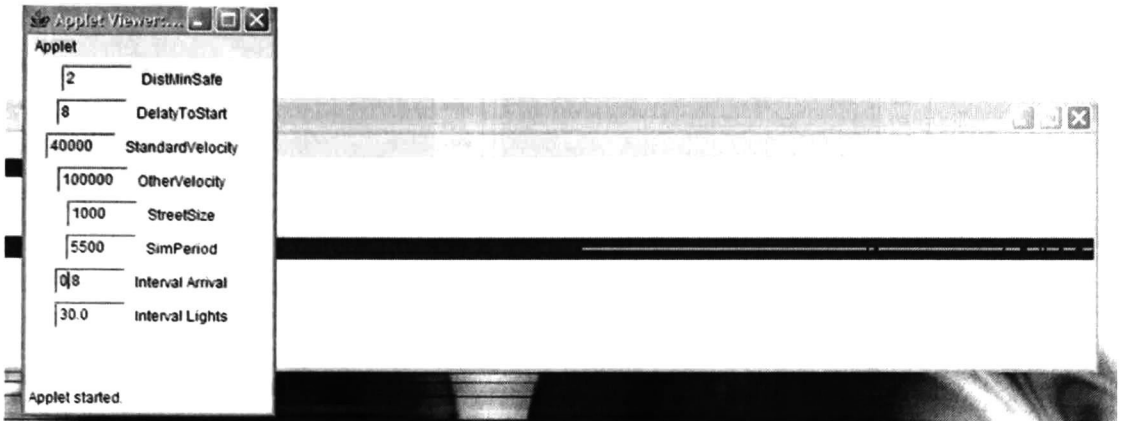


Figura 5-2 Pantalla del simulador.

Los parámetros que se pueden modificar al simulador son:

- a) Distancia de seguridad mínima.
- b) Retardo de reacción del conductor.
- c) Velocidad estándar.
- d) Otra velocidad.
- e) Tamaño de la calle.
- f) Periodo de simulación.
- g) Intervalo de arribo.
- h) Intervalo de cambio de fases.

Si modificamos los parámetros podemos apreciar el comportamiento esperado, como por ejemplo una cola de vehículos, mostrado en la figura 5-2. Si colocamos los siguientes valores a los parámetros $distminsafe = 2$ mtrs, $delayToStart=0.2$ min., el tiempo de simulación en 91.66 hrs., el resultado lo muestra en 19 segundos.

5.2 Interpretación de resultados.

En éstas etapas se toman los resultados de las simulaciones y se analizan los resultados ya sea de manera gráfica o analítica. En la figura 5-3 se muestran los resultados obtenidos para dos corridas con parámetros diferentes.

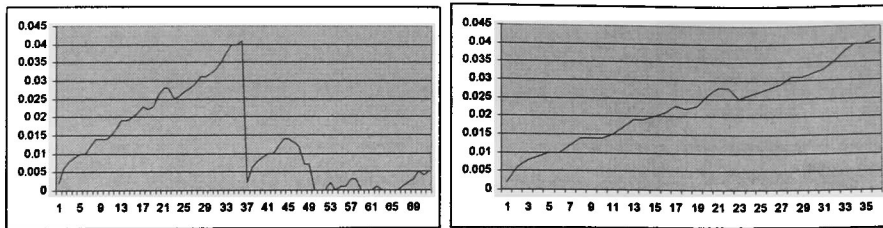


Figura 5-3 Gráficas de densidad con a) 0.2 min. y b) 8 min. de retardo en reacción del conductor

La interpretación de estas gráficas es como sigue: en el capítulo anterior se definió la densidad como la relación de número de carros por unidad de espacio, entonces en la imagen de la izquierda de la figura 5-4, que es resultado de una simulación con una reacción del conductor igual a 0.2 min, la densidad tiende a bajar, al inicio se incrementa en razón del estado en rojo del semáforo, así que se comienzan a acumular los vehículos, pero una vez transcurridos los primeros cambios de semáforo la densidad tiende a ser baja. Por el contrario, si la reacción del conductor es muy larga (8 min), esto provocara que se empiecen a acumular los vehículos (generar colas), esto debido a que además del obstáculo del semáforo, aún cuando el conductor puede avanzar tarda en reaccionar mucho tiempo, provocando el acumulamiento.

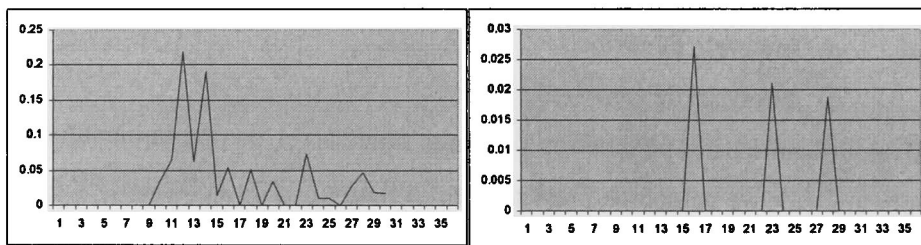


Figura 5-4 Gráficas de flujo con a) 0.2 min. y b) 8 min. de retardo en reacción del conductor

Con respecto al flujo, la gráfica de la izquierda de la figura 5-4 muestra que los picos están más juntos, dado que el flujo es constante y la altura de los picos representa el valor de flujo. La gráfica de la derecha es lo contrario, los picos están muy separados, es decir, es muy lento el tráfico y sólo se incrementa el flujo en el cambio de semáforos, que es máximo. Llegando a concluir que se pueden generar los fenómenos de congestión y colas, provocados por el incremento de vehículos en el sistema de tráfico urbano y la eliminación de estos fenómenos en la reducción del incremento de los vehículos al sistema. En este capítulo se finaliza la

descripción de la metodología para el desarrollo de simuladores basados en multi-agente presentado en el capítulo anterior.

Capítulo 6

CONCLUSIONES

Resumen. Se presentan las conclusiones generadas de la investigación del estado del arte y la implementación de la propuesta del simulador de tráfico urbano, hecha en el capítulo 2. Se muestran los avances y trabajo futuro en relación a ésta tesis..

El formalismo tradicional de redes de Petri utilizado para modelar un sistema de red de tráfico urbano captura de manera adecuada el comportamiento del sistema. Sin embargo, no se encontró un mecanismo que nos permita convertir el modelo conceptual en un modelo computacional de manera eficiente, los demás formalismos, de igual manera capturan las características del sistema, pero carecen de la característica jerárquica, que reduce la complejidad del diseño de un simulador de tráfico urbano.

Los simuladores actuales utilizan la técnica de avance de tiempo de la simulación por división de tiempo, que resulta ser menos eficiente que la de evento siguiente. No existe simulador que utilice ésta técnica. El paradigma de programación más utilizado es el orientado agente por ser el que de manera natural representa el comportamiento de las entidades en el sistema de tráfico urbano. La ventaja de los modelos multi-agentes es que ofrecen un sistema más interactivo y abierto a diferencia de los modelos clásicos. Así que es posible modificar dinámicamente las condiciones de la simulación (preferencias de conductor virtuales, algoritmos de control de las luces de tráfico, etc.).

Cuando se presentó el diseño de un simulador basado en agentes para sistemas de tráfico urbano, basado en la metodología orientada a eventos y el modelado de sistemas de tráfico urbano usando una extensión al formalismo de redes de Petri (anidadas): 3-LNS, así como un prototipo de simulador desarrollado en el lenguaje JAVA.

Se ha mostrado que los modelos jerárquicos y modulares (en 3-LNS) nos llevan a un diseño claro y bien estructurado de software basado en agentes dado que la simulación orientada a eventos se adapta y es eficiente en los sistemas multi-agentes.

Se continua incrementando la complejidad de las capacidades de los agentes: móviles, agente administrador de la calle, para poder manejar problemas mas realistas. Es decir, poder incrementar el numero de intersecciones y enfrentar los problemas de control de eventos en cada una de las calles e intersecciones. Pretendiendo completar el prototipo de simulación que permita agregar componentes de ITS y seguir trabajando en el control y optimización de los semáforos.

REFERENCIAS BIBLIOGRÁFICAS

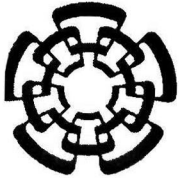
- (Akcelik, 2004) Akcelik R., Besley M. and Thompson D., **“Traffic Models”**, 2004, <http://www.aatraffic.com/TrafficModels.htm>.
- (Banks, 2000) Banks J., Carson J. S., and Nelson B. L., et al, **“Discrete-Event System Simulation”**, 2000, Prentice Hall International Series in industrial and systems engineering, 3rd edition.
- (Barceló, 1998) Barceló J., Casas J., and Ferrer J. L., et al, **“Modelling advanced transport telematic applications with microscopic simulators: The case of AIMSUN2”**, 1998, TSS-Transport Simulation Systems, Barcelona.
- (Barceló, 1998a) Barceló J., Ferrer J.L., and García D., et al, **“Microscopic traffic simulation for att systems analysis. a parallel computing version”**, 1998, Contribution to the 25th Aniversary of CRT, University of Montreal, www.tss-bcn.com/documents.html.
- (Barceló, 2001) Barceló J., **“Microscopic traffic simulation: A tool for the analysis and assessment of ITS systems”**, 2001, En Highway Capacity Committee, Half Year Meeting, Lake Tahoe.
- (Barclay, 1999) Barclay W.J., **“Point-to-point microscopic simulation: a discussion of issues”**, 1999, Paper presentado en el 1st Western Pacific y 3rd Australia-Japan Workshop on Stochastic Models.
- (Beimborn, 1996) Beimborn E., and Kennedy R., **“Inside the Blackbox: Making Transportation Models Work for Livable Communities”**, 1996, Citizens for a Better Environment, Milwaukee, Wisconsin, and the Environmental Defense Fund, Washington, D.C..
- (Benenson, 2004) Benenson I., **“Geosimulation: Automata-based modeling of urban phenomena”**, 2004, John Wiley and Sons, pp. 312.
- (Budd, 1995) Budd, T., **“Multiparadigm Programming in Leda”**, 1995, Addison Wesley, Reading, Massachusetts, pp. 394.
- (Champion, 2001) Champion A., Espie E., and Auberlet J., **“Behavioral Road Traffic Simulation with ARCHISIM”**, 2001, Summer Computer Simulation Conference.

- (Claramunt, 2001) Claramunt C., and Jiang B., “**A Qualitative Model for the Simulation of Traffic Behaviours in a Multi-lane Environment**”, 2001, Journal of Geographical Sciences, Vol. 11, special issue on geo-visualization, pp. 29-42.
- (DiCesare, 1994) DiCesare F., Kulp P., and Gile M., et al, “**The application of Petri nets to the modeling, analysis and control of intelligent urban traffic networks,**” 1994, En Proc. 15th Int. Conf. Application Theory Petri Nets, R. Valette, Ed. Zaragoza, Spain, pp. 2-15
- (DiFebbraro, 2001) Di Febbraro A., Giglio D., and Sacco N., “**Modular representation of urban traffic systems based on hybrid Petri nets**”, 2001, En Proc. IEEE Int. Conf. Intelligent Transportation Systems, pp. 866–871.
- (Drogoul, 2002) Drogoul A., Vanbergue D., and Meurisse T., “**Multi-Agent Based Simulation: Where are the Agents ?**”, 2002, Proceedings of MABS'02, LNCS, Springer-Verlag.
- (Erlemann, 2004) Erlemann K., “**BABSIM , An Object-oriented Software Framework for Microscopic Simulation of Freeway Traffic**”, 2004, En the Xth International Conference on Computing in Civil and Building Engineering, pp. 316.
- (Fishwick, 1996) Fishwick P., “**Computer simulation : growth through extension**”, 1996, IEEE Potential, February/March, pp. 24-27.
- (Floyd, 1979) Floyd R., “**The paradigms of programming**”, 1979, En Communications of the ACM archive, Volume 22 , Issue 8, pp. 455 – 460.
- (Gilbert, 1999) Nigel G., and Troitzsch K.G., “**Simulation for the Social Scientist**”, 1999, Open.University Press.
- (Helsgaun, 2000) Helsgaun K., “**Discrete Event Simulation in Java**”, 2000, Writings on Computer Science, Roskilde University.
www.akira.ruc.dk/~keld/research/JAVASIMULATION/JAVASIMULATION-1.1/docs/Report.pdf
- (Hoogendoorn, 2001) Hoogendoorn S., and Bovy P., “**State-of-the-art of Vehicular Traffic Flow Modelling**”, 2001, En Special Issue on Road Traffic Modelling and Control of the Journal of Systems and Control Eng. Proc. of the IME I.

- (Julvez, 2004) Julvez J., and Boel R., "**Modelling and controlling traffic behaviour with continuous Petri nets**", 2004, IFAC World Congress.
<http://www.supelec.fr/lss/CTS/WWW/paper-julvez-01.pdf>
- (Kimata, 2001) Kimata N., Kisino K., and Siromizu Y., "**Development of Practical Software for Micro Traffic Flow Petri Net Simulator**", 2001, En Journal of Civil Engineering Information Processing System.
<http://www.jsce.or.jp/committee/cceips/Proceed02/fullpaper/2001-02.PDF>
- (Krajzewicz, 2002) Krajzewicz D., Hertkorn G., and Rössel C., et al, "**SUMO (Simulation of Urban MObility) An open-source traffic simulation**" 2002, En Proceedings of the 4th Middle East Symposium on Simulation and Modelling, SCS European Publishing House, P. 183-187, 4th Middle East Symposium on Simulation and Modelling.
- (LoTartaro, 2001) Lo Tártaro M., Torres C., and Wainer G., "**Defining models of urban traffic using the TSC tool**", 2001, En Proceedings of the Winter Simulation Conference, Washington, DC. U.S.A..
<http://www.sce.carleton.ca/faculty/wainer/celldevs/thesis/tsc/tesis.zip>
- (Mussi, 2000) Mussi P., Jiménez T., and Siegel G., "**A road traffic simulator : car-following and lane-changing**", 2000, En *European Simulation Multiconference 2000*, Gent, Belgium, pp. 241--245.
- (Nagel, 1992) Nagel K., Schreckenberg M., "**A cellular automaton model for freeway traffic**", 1992, Journal of Physique I, vol. 2 , pp. 2221-2229.
- (Nagel, 1998) Nagel K., et al., "**Two-lane traffic rules for cellular automata: A systematic approach**", 1998, Physical Review E, Vol. 58 No. 2.
<http://www.sim.inf.ethz.ch/papers/nagel-etc-2lane/nagel-etc-2lane.pdf>
- (Nagel, 2002) Nagel K., "**Traffic networks**". 2002, En Handbook on networks, edited by S. Bornholdt and H.G Schuster.
<http://www.sim.inf.ethz.ch/papers/networks-book/networks-book.pdf>
- (Nagel, 2004) Nagel K., "**Multi-Agent Transportation Simulation**", 2004, Draft of forthcoming book.

- (Ramos, 2003) Ramos G. M., and Gonzalez V. J., “¿Un transporte público digno para los tapatíos?”, 2003, En www.claves.udg.mx, 7 de marzo del 2003, CUCSH-UDG.
- (Russell, 1996) Russell G., Shaw P., and Ferguson N., “**Accurate Rapid Simulation of Urban Traffic using Discrete Modelling**”, 1996, Unpublished Report, Napier University, Computer Studies, Craiglockhart, 216 Colinton Road, Edinburgh, EH14 1DJ. <http://grussell.org/me/pubs/abs-9.pdf>
- (Schriber, 1997) T. J., and Brunner D. T., “**Inside discrete-event simulation software: how it Works and why it matters**”, 1997, Proceedings of the Winter Simulation Conference, Vol. 1, pag. 72-80.
- (Schulze, 1997) Schulze T., Fliess T., “**Urban Traffic Simulation with psycho-physical vehicle-following models**”, 1997, En Proceedings of the 1997 Winter Simulation Conference ed. S. Andradottir, K. J. Healy, D. H. Withers, and B. L. Nelson
- (Severance, 2001) Severance F. L., “**System Modeling and Simulation**”, 2001, John Wiley and Sons, pp. 518.
- (Unger, 2005) Unger H., “**The Guadalajara Urban Traffic Control Project An Overview about features and needs for tomorrow's mobile city communities**”, 2005, En I2CS 2005 Innovative Internet Community systems, June 20-22 , Paris, France.
- (Tolba, 2001) Tolba C., “**Continuous Petri Nets Models for the analysis of traffic urban networks**”, 2001, En: Proceedings of IEEE Systems, Man, and Cybernetics Conference, Arizona, USA, pages 1323-1328.
- (Wang, 1999) Wang J., Jin C., and Deng Y., “**Performance analysis of traffic networks based on stochastic timed Petri net models**”, 1999, En Proceedings of the 5th IEEE International Conference on Engineering of Complex Computer Systems, Las Vegas, 77-85.
- (Wiering, 2003) Wiering M., “**Intelligent Traffic Control**”, 2003, En ERCIM News, No. 53.
http://www.ercim.org/publication/Ercim_News/enw53/wiering.html

- (Wilson, 2001) Wilson, R. E., “**An analysis of Gipp’s car-following model of highway traffic**”, 2001, En IMA Journal of Applied Mathematics, p. 316.
- (Zamora, 2002) Zamora R., and McIntosh H., “**Modelación de flujo de tránsito de autos utilizando autómatas celulares**”, 2002, tesis de doctorado, Cinvestav Puebla.
<http://delta.cs.cinvestav.mx/~mcintosh/comun/tesismaestria/rene/tesisReneHtml/Tesis.html>



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

Micro-Simulación de tráfico urbano basada en Sistemas de Agentes
Móviles

del (la) C.

Emmanuel LÓPEZ NERI

el día 31 de Agosto de 2005.

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

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

Dr. Antonio Ramírez Treviño
Investigador CINVESTAV 2A
CINVESTAV Unidad Guadalajara

Dr. Helena Unger
Profesor investigador
Universidad de Guadalajara



CINVESTAV
BIBLIOTECA CENTRAL



SS1T000008084